Panasonic®

PROGRAMMABLE CONTROLLER

FP Series

Programming Manual

ARCT1F313E-18

2011.08 | panasonic-electric-works.net/sunx

Table of Contents

Chapter 1 Relays, Memory Areas and Constants

Chapter 2 Basic Instructions

Chapter 3 High−level Instructions

Chapter 4 Precautions Concerning Programs

Chapter 5 Appendix

Basic Instructions

Special setting instructions

Data compare instructions

High–level Instructions

Data transfer instructions

Control instruction

Binary arithmetic instructions

BCD arithmetic instructions

Data compare instructions

Logic operation instructions

Data conversion instructions

Data shift instructions

FIFO instructions

Basic function instructions

Data rotate instructions

Bit manipulation instructions

Basic function instruction

Special instructions

BIN arithmetic instructions

Special instructions (High−speed counter instructions)

Character string instructions

Integer type data processing instructions

Integer type non-linear function instructions

BCD type real number operation instructions

Floating point type real number operation instructions (for FP2/FP2SH/FP10SH)

Floating point type real number data processing instructions

Time series processing instruction

Compare instructions

Index register bank processing instructions

File register bank processing instructions

Chapter 1

Relays, Memory Areas and Constants

1.1 Table of Relays, Memory Areas and Constants

1.1.1 FP0/FP−e

FP0

Item		Numbering			Function
		C10/C14/C16	C32/SL1	T32C	
Relay	External input relay (X)	208 points (X0 to X12F)			Turns on/off based on external input.
	External output relay (Y)	208 points (Y0 to Y12F)			Externally outputs on/off state.
	Internal relay (R) (* Note 2)	1,008 points (R0 to R62F)			Relay which turns on/off only within program.
	Timer (T) (* Note 2)	144 points (T0 to T99/C100 to C143) (* Note 1)			If a **TM** instruction has timed out, the contact with the same number turns on.
	Counter (C) (* Note 2)				If a **CT** instruction has counted up, the contact with the same number turns on.
	Special internal relay (R)	64 points (R9000 to R903F)			Relay which turns on/off based on specific conditions and is used as a flag.
Memory area	External input relay (WX)	13 words (WX0 to WX12)			Code for specifying 16 external input points as one word (16 bits) of data.
	External output relay (WY)	13 words (WY0 to WY12)			Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (WR) (* Note 2)	63 words (WR0 to WR62)			Code for specifying 16 internal relay points as one word (16 bits) of data.
	Data register (DT) (* Note 2)	1,660 words (DT0 to DT1659)	6,144 words (DT0 to DT6143)	16,384 words (DT0 to DT16383)	Data memory used in program. Data is handled in 16−bit units (one word).
	Timer/Counter set value area (SV) (* Note 2)	144 words (SV0 to SV143)			Data memory for storing a target value of a timer and an initial value of a counter. Stores by timer/counter number.
	Timer/Counter elapsed value area (EV) (* Note 2)	144 words (EV0 to EV143)			Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/counter number.
	Special data register (DT)	112 words (DT9000 to DT9111)		112 words (DT90000 to DT90111)	Data memory for storing specific data. Various settings and error codes are stored.
	Index register (I)	2 words (IX, IY)			Register can be used as an address of memory area and constants modifier.
Control instruction point	Master control relay points (MCR)	32 points			
	Number of labels (JP and LOOP)	64 labels		255 labels	
	Number of step ladders	128 stages		704 stages (* Note 1)	
	Number of subroutines	16 subroutines		100 subroutines	
	Number of interrupt programs	7 programs (external 6 points, internal 1 point) SL1: 1 program (internal 1 point)			

Item		Numbering: C10/C14/C16	Numbering: C32/SL1	Numbering: T32C	Function
Constant	**Decimal constants (K)**	K−32768 to K32767 (for 16-bit operation)			
		K−2147483648 to K2147483647 (for 32-bit operation)			
	Hexadecimal constants (H)	H0 to HFFFF (for 16-bit operation)			
		H0 to HFFFFFFFF (for 32-bit operation)			
	Floating point type (F)	F$-1.175494 \times 10^{-38}$ to F-3.402823×10^{38}			
		F1.175494×10^{-38} to F3.402823×10^{38}			

Notes

1) **The points for the timer and counter can be changed by the setting of system register 5. The numbers given in the table are the numbers when system register 5 is at its default setting.**

2) **There are two unit types, the hold type that saves the conditions that exist just before turning the power off or changing form the RUN mode to PROG. mode, and the non−hold type that resets them. For the FP0 T32C, the selection of hold type and non−hold type can be changed by the setting of system register. These areas can be specified as hold type or non−hold type by setting system register. For the FP0 C10/C14/C16/C32/SL1, that area is fixed and allotted the numbers as shown below.**

Hold type and Non-hold type areas

Item		C10/C14/C16	C32/SL1
Timer		Non−hold type: All points	
Counter	**Non-hold type**	From the set value to C139	From the set value to C127
	Hold type	4 points (elapsed values) (C140 to C143)	16 points (elapsed values) C128 to C143
Internal relay	**Non-hold type**	976 points (R0 to R60F) 61 words (WR0 to WR60)	880 points (R0 to R54F) 55 words (WR0 to WR54)
	Hold type	32 points (R610 to R62F) 2 words (WR61 to WR62)	128 points (R550 to R62F) 8 words (WR55 to WR62)
Data register	**Non-hold type**	1652 words (DT0 to DT1651)	6112 words (DT0 to DT6111)
	Hold type	8 words (DT1652 to DT1659)	32 words (DT6112 to DT6143)

FP–e

	Item	Number of points	Memory area available for use		Function
			Matsushita	IEC	
Relay	**External input relay (see note 3)**	208	**X**0–**X**12F	%**IX**0.0– %**IX**12.15	Turns on or off based on external input.
	External output relay (see note 3)	208	**Y**0–**Y**12F	%**QX**0.0– %**QX**12.15	Outputs on or off state externally.
	Internal relay (see note 2)	1008	**R**0–**R**62F	%**MX0**.0– %**MX0**.62.15	Turns on or off only within a program.
	Timer (see notes 1 and 2)	100	**T**0–**T**99/ **C**100–**C**143	%**MX1**.0– %**MX1**. 99/ %**MX2**.100– %**MX2**.143	Turns on when the timer reaches the specified time. Corresponds to the timer number.
	Counter (see notes 1 and 2)	44	**C**100–**C**143/ **T**0–**T**99	%**MX2**.100– %**MX2**.143/ %**MX1**.0– %**MX1**.99	Turns on when the counter increments. Corresponds to the counter number.
	Special internal relay	64	**R**9000–**R**903F	%**MX0**.900.0– %**MX0**.903.15	Turns on or off based on specific conditions. Used as a flag.
Memory area (words)	**External input relay (see note 3)**	13 words	**WX**0–**WX**12	%**IW**0– %**IW**12	Code for specifying 16 external input points as one word (16 bits) of data.
	External output relay (see note 3)	13 words	**WY**0–**WY**12	%**QW**0– %**QW**12	Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (see note 2)	63 words	**WR**0–**WR**62	%**MW0**.0– %**MW0**.62	Code for specifying 16 internal relay points as one word (16 bits) of data.
	Data register (see note 2)	1660 words	**DT**0–**DT**1659	%**MW5**.0– %**MW5**.1659	Data memory used in a program. Data is handled in 16-bit units (one word).
	Timer/counter set value area	144 words	**SV**0–**SV**143	%**MW3**.0– %**MW3**.143	Data memory for storing a target value of a timer and an initial value of a counter. Stores by timer/counter number.
	Timer/counter elapsed value area (see note 2)	144 words	**EV**0–**EV**143	%**MW4**.0– %**MW4**.143	Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/counter number.
	Special data register	112 words	**DT**9000– **DT**9111	%**MW5**.9000– %**MW5**.9111	Data memory for storing specific data. Various settings and error codes are stored.
	Index register	2 words	IX–IY	%**MW6**.0– %**MW6**.1	Used as an address of memory area and constants modifier.

Item		Number of points	Memory area available for use		Function
			Matsushita	IEC	
Memory area (double word) (see note 4)	**External input relay (see note 3)**	6 double words	**DWX**0–**DWX**11	%**ID**0–%**ID**11	Code for specifying 32 external input points as a double word (32 bits) of data.
	External output relay (see note 3)	6 double words	**DWY**0–**DWY**11	%**QD**0–%**QD**11	Code for specifying 32 external output points as double word (32 bits) of data.
	Internal relay (see note 2)	31 double words	**DWR**0–**DWR**61	%**MD0**.0–%**MD0**.61	Code for specifying 32 internal relay points as double word (32 bits) of data.
	Data register (see note 2)	830 double words	**DDT**0–**DDT**1658	%**MD5**.0–%**MD5**.1658	Data memory used in a program. Data is handled in 32-bit units (double word).
	Timer/counter set value area	72 double words	**DSV**0–**DSV**142	%**MD3**.0–%**MD3**.142	Data memory for storing a target value of a timer and an initial value of a counter. Stores by timer/counter number.
	Timer/counter elapsed value area (see note 2)	72 double words	**DEV**0–**DEV**142	%**MD4**.0–%**MD4**.142	Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/counter number.
	Special data register	56 double words	**DDT**9000–**DDT**9110	%**MD5**.9000–%**MD5**.9110	Data memory for storing specific data. Various settings and error codes are stored.
	Index register	1 double words	**DI**0	%**MD6**.0	Used as an address of memory area and constants modifier.

Item		Number of points
Control instruction point	**Master control relay points (MCR)**	32 points
	Number of labels (JP and LOOP)	64 labels
	Number of step ladders	128 stages
	Number of subroutines	16 subroutines
	Number of interrupt programs	7 programs (external: 6, internal: 1)

Item		Range available for use	
		Matsushita	IEC
Constant	**Decimal constants**	K–32768 to K32767 (for 16-bit operation)	–32768 to 32767 (for 16-bit operation)
		K–2147483648 to K2147483647 (for 32-bit operation)	–2147483648 to 2147483647 (for 32-bit operation)
	Hexadecimal constants	H0 to HFFFF (for 16-bit operation)	16#0 to 16#FFFF (for 16-bit operation)
		H0 to HFFFFFFFF (for 32-bit operation)	16#0 to 16#FFFFFFFF (for 32-bit operation)
	Floating point type	F–1.175494 $\times$ 10^{-38} to F–3.402823 $\times$ 10^{38}	–1.17549410E–38 to –3.402823E38
		F1.175494 $\times$ 10^{-38} to F3.402823 $\times$ 10^{38}	1.17549410E–38 to 3.402823E38

Notes

1) **The points for the timer and counter can be changed by the setting of System register No. 5. The number given in the table above are the numbers when System register No. 5 is at its default setting.**
2) **There are two unit types;
the hold type that saves the conditions that exist just before turning the power off or changing from the RUN mode to PROG. mode, and the non-hold type that resets them.
These areas can be specified as hold type or non-hold type by setting system register.
For the FP-e, that area is fixed and allotted the numbers as shown in the table below. For the FP-e with clock/calendar function type, the selection of hold type and non-hold type can be changed by the setting of system register.**
3) **The number of points noted above is the number reserved in the system. For the actual number of points available for use, refer to "I/O Allocation" in Appendix A.**
4) **Double words cannot be specified with FPWIN GR.**

Hold type and non-hold type areas*1

Model		AFPE224300 (Standard type)	AFPE224305 (Calendar timer type)	AFPE214325 (Thermocouple input type)
Timer		Non-hold type: all points		
Counter	Non-hold type	From the set value to C139		
	Hold type	C140 to C143, EV140 to EV143 (elapsed value)		
		SV: non-hold *2	SV: hold	
Internal relay	Non-hold type	976 points (R0 to R60F) 61 words (WR0 to WR60)		
	Hold type	32 points (R610 to R62F) 2 words (WR61 to WR62)		
Data register	Non-hold type	1652 words (DT0 to DT1651)		
	Hold type	8 words (DT1652 to DT1659)		

***1 When a battery is installed in a calendar timer type FP-e, the areas above can be changed using the system register. If a battery is not installed, the data cannot be stored even when the settings are changed using the system register.**

***2 Use the following methods for holding the SV data:**

- **Set the transfer instruction for the special data register (DT) to hold the data. Then, perform the setting so that the data can be transferred from DT to SV after the RUN mode starts.**
- **Use the FP-e model with a battery.**

1.1.2 FP0R

Item		Number of points and range of memory area available for use		Function
		C10, C14, C16	C32, T32, F32	
Relay	External input (X) Note1)	1760 points (X0 to X109F)		Turns on or off based on external input.
	External output (Y) Note1)	1760 points (Y0 to Y109F)		Externally outputs on or off state
	Internal relay (R) Note2)	4096 points (R0 to R255F)		Relay which turns on or off only within program.
	Link relay (L) Note2)	2048 points (L0 to L127F)		This relay is a shared relay used for PLC link.
	Timer (T) Note2)	1024 points (T0 to T1007/C1008 to C1023) Note3)		This goes on when the timer reaches the specified time. It corresponds to the timer number.
	Counter (C) Note2)			This goes on when the counter increments. It corresponds to the counter number.
	Special internal relay (R)	224 points (from R9000)		Relay which turns on or off based on specific conditions and is used as a flag.
Memory area	External input (WX) Note1)	110 words (WX0 to WX109)		Code for specifying 16 external input points as one word (16 bits) of data.
	External output (WY) Note1)	110 words (WY0 to WY109)		Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (WR) Note2)	256 words (WR0 to WR255)		Code for specifying 16 internal relay points as one word (16 bits) of data.
	Link relay (WL)	128 words (WL0 to WL127)		Code for specifying 16 link relay points as one word (16 bits) of data.
	Data register (DT) Note2)	12315 words (DT0 to DT12314)	32765 words (DT0 to DT32764)	Data memory used in program. Data is handled in 16-bit units (one word).
	Link register (LD) Note2)	256 words (LD0 to LD255)		This is a shared data memory which is used within the PLC link. Data is handled in 16-bit units (one word).
	Timer/Counter set value area (SV) Note2)	1024 words (SV0 to SV1023)		Data memory for storing a target value of a timer and setting value of a counter. Stores by timer/counter number.
	Timer/Counter elapsed value area (EV) Note2)	1024 words (EV0 to EV1023)		Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/counter number.
	Special data register (DT)	440 words (DT90000 to DT90439)		Data memory for storing specific data. Various settings and error codes are stored.
	Index register (I)	14 words (I0 to ID)		Register can be used as an address of memory area and constants modifier.

Item		Number of points and range of memory area available for use		Function
		C10, C14, C16	C32, T32, F32	
Control instruc-tion point	Master control relay points (MCR)	256 points		
	Number of labels (JP and LOOP)	256 points		
	Number of step ladders	1000 stages		
	Number of subroutines	500 subroutines		
	Number of interrupt programs	C10: 11 programs (6 external input points, 1 periodical interrupt point, 4−pulse match points) Other than C10: 13 programs (8 external input points, 1 periodical interrupt point, 4−pulse match points)		
Constant	Decimal constants (K)	K−32, 768 to K32, 767 (for 16−bit operation)		
		K−2, 147, 483, 648 to K2, 147, 483, 647 (for 32−bit operation)		
	Hexadecimal constants (H)	H0 to HFFFF (for 16−bit operation)		
		H0 to HFFFFFFFF (for 32−bit operation)		
	Floating point type (F)	F−1.175494 $\times$ 10^{-38} to F−3.402823 $\times$ 10^{38}		
		F1.175494 $\times$ 10^{-38} to F3.402823 $\times$ 10^{38}		

Notes

1) **The number of points noted above is the number reserved as the calculation memory. The actual number of points available for use is determined by the hardware configuration.**

2) **There are two types, one is the hold type that the last state is stored even if the power supply turns off or the mode is changed to PROG. mode from RUN mode, and the other is the non−hold type that the state is reset.**
   For C10/C14/C16/C32: The hold type areas and non−hold type areas are fixed. For information on the sections of each area, refer to the performance specifications.
   For T32/F32: The settings of the hold type areas and non−hold type areas can be changed using the system registers.
   On T32, if the battery has run out, the data in the hold area may be indefinite (Not cleared to 0)

3) **The points for the timer and counter can be changed by the setting of system register 5. The number given in the table are the numbers when system register 5 is at its default setting.**

1.1.3 FPΣ

12k type

Item		Number of points	Memory area available for use	Function
Relay	**External input relay (see note 1) FPG–C32T/C32TTM**	512	**X**0–**X**31F	Turns on or off based on external input.
	External input relay (see note 1) FPG–C32T2/C32T2TM FPG–C24R2/C24R2TM FPG–C28P2	1184	**X**0–**X**73F	
	External output relay (see note 1) FPG–C32T/C32TTM	512	**Y**0–**Y**31F	Externally outputs on or off state.
	External output relay (see note 1) FPG–C32T2/C32T2TM FPG–C24R2/C24R2TM FPG–C28P2	1184	**Y**0–**Y**73F	
	Internal relay (see note 2)	1568	**R**0–**R**97F	Turns on or off only within a program.
	Link relay (see note 2)	1024	**L**0–**L**63F	Shared relay used for PLC link.
	Timer (see notes 2 and 3)	1024	**T**0–**T**1007/**C**1008–**C**1023	Goes on when the timer reaches the specified time. Corresponds to the timer number.
	Counter (see notes 2 and 3)	1024	**C**1008–**C**1023/**T**0–**T**1007	Goes on when the counter increments. Corresponds to the counter number.
	Special internal relay	176	**R**9000–**R**910F	Turns on or off based on specific conditions. Used as a flag.
Memory area (words)	**External input relay (see note 1) FPG–C32T/C32TTM**	32 words	**WX**0–**WX**31	Code for specifying 16 external input points as one word (16 bits) of data.
	External input relay (see note 1) FPG–C32T2/C32T2TM FPG–C24R2/C24R2TM FPG–C28P2	74 words	**WX**0–**WX**73	
	External output relay (see note 1) FPG–C32T/C32TTM	32 words	**WY**0–**WY**31	Code for specifying 16 external output points as one word (16 bits) of data.
	External output relay (see note 1) FPG–C32T2/C32T2TM FPG–C24R2/C24R2TM FPG–C28P2	74 words	**WY**0–**WY**73	
	Internal relay (see note 2)	98 words	**WR**0–**WR**97	Code for specifying 16 internal relay points as one word (16 bits) of data.
	Link relay	64 words	**WL**0–**WL**63	Code for specifying 16 link relay points as one word (16 bits) of data.

Item		Number of points	Memory area available for use	Function
Memory area (words)	**Data register (see note 2)**	32765 words	**DT**0−**DT**32764	Data memory used in a program. Data is handled in 16-bit units (one word).
	Link data register (see note 2)	128 words	**LD**0−**LD**127	A shared data memory which is used within the PLC link. Data is handled in 16-bit units (one word).
	Timer/counter set value area (see note 2)	1024 words	**SV**0−**SV**1023	Data memory for storing a target value of a timer and an initial value of a counter. Stores by timer/counter number.
	Timer/counter elapsed value area (see note 2)	1024 words	**EV**0−**EV**1023	Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/counter number.
	Special data register	260 words	**DT**90000−**DT**90259	Data memory for storing specific data. Various settings and error codes are stored.
	Index register	14 words	**I**0−**I**D	Can be used as an address of memory area and constants modifier.

Item		Number of points
Control instruction point	**Master control relay points (MCR)**	256
	Number of labels (JP and LOOP)	256
	Number of step ladders	1,000 stages
	Number of subroutines	100 subroutines
	Number of interrupt programs	9 programs (8 external input points “X0 to X7”, 1 periodical interrupt point “0.5 ms to 30s”)

Item		Range available for use
Constant	**Decimal constants (integer type)**	K−32768 to K32767 (for 16-bit operation)
		K−2147483648 to K2147483647 (for 32-bit operation)
	Hexadecimal constants	H0 to HFFFF (for 16-bit operation)
		H0 to HFFFFFFFF (for 32-bit operation)
	Floating point type	F−1.175494 $\times$ 10^{-38} to F−3.402823 $\times$ 10^{38}
		F1.175494 $\times$ 10^{-38} to F3.402823 $\times$ 10^{38}

Notes

1) **The number of points noted above is the number reserved as the calculation memory. The actual number of points available for use is determined by the hardware configuration.**
2) **If no battery is used, only the fixed area is backed up (counters 16 points: C1008 to C1023, internal relays 128 points: R900 to R97F, data registers: DT32710 to DT32764). When the optional battery is used, data can be backed up. Areas to be held and not held can be specified using the system registers.**
3) **The points for the timer and counter can be changed by the setting of system register 5. The number given in the table are the numbers when system register 5 is at its default setting.**

32k type

Item		Number of points and range of memory area available for use 32TH/C32THTM C32T2H/C32T2HTM C24R2H/C24R2HTM C28P2H/C28P2HTM	Function
Relay	**External input (see note 1) (X)**	1184 points (X0 to X73F)	Turns on or off based on external input.
	External output (see note 1) (Y)	1184 points (Y0 to Y73F)	Externally outputs on or off state.
	Internal relay (see note 2) (R)	4096 points (R0 to R255F)	Relay which turns on or off only within program.
	Link relay (see note 2) (L)	2048 points (L0 to R127F)	This relay is a shared relay used for PLC link.
	Timer (see note 2) (T)	1024 points (T0 to T1007/C1008 to C1023) (see note 3)	This goes on when the timer reaches the specified time. It corresponds to the timer number.
	Counter (see note 2) (C)		This goes on when the counter increments. It corresponds to the counter number.
	Special internal relay (R)	176 points (R9000 to R910F)	Relay which turns on or off based on specific conditions and is used as a flag.
Memory area	**External input (see note 1) (WX)**	74 words (WX0 to WX73)	Code for specifying 16 external input points as one word (16 bits) of data.
	External output (see note 1) (WY)	74 words (WY0 to WY73)	Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (see note 2) (WR)	256 words (WR0 to WR255)	Code for specifying 16 internal relay points as one word (16 bits) of data.
	Link relay (WL)	128 words (WL0 to WL127)	Code for specifying 16 link relay points as one word (16 bits) of data.
	Data register (see note 2) (DT)	32765 words (DT0 to DT32764)	Data memory used in program. Data is handled in 16-bit units (one word).
	Link register (see note 2) (LD)	256 words (LD0 to LD255)	This is a shared data memory which is used within the PLC link. Data is handled in 16-bit units (one word).
	Timer/Counter set value area (see note 2) (SV)	1024 words (SV0 to SV1023)	Data memory for storing a target value of a timer and setting value of a counter. Stores by timer/counter number.
	Timer/Counter elapsed value area (see note 2) (EV)	1024 words (EV0 to EV1023)	Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/counter number.
	Special data register (DT)	260 words (DT90000 to DT90259)	Data memory for storing specific data. Various settings and error codes are stored.
	Index register (I)	14 words (I0 to ID)	Register can be used as an address of memory area and constants modifier.

Item		Number of points and range of memory area available for use 32TH/C32THTM C32T2H/C32T2HTM C24R2H/C24R2HTM C28P2H/C28P2HTM	Function
Control instruction point	Master control relay points (MCR)	256	
	Number of labels (JP and LOOP)	256	
	Number of step ladders	1,000 stages	
	Number of subroutines	100 subroutines	
	Number of interrupt programs	9 programs (8 external input points “X0 to X7”, 1 periodical interrupt point “0.5 ms to 30s”)	
Constant	Decimal constants (integer type) (K)	K−32768 to K32767 (for 16-bit operation)	
		K−2147483648 to K2147483647 (for 32-bit operation)	
	Hexadecimal constants (H)	H0 to HFFFF (for 16-bit operation)	
		H0 to HFFFFFFFF (for 32-bit operation)	
	Floating point type (F)	F$-1.175494 \times 10^{-38}$ to F-3.402823×10^{38}	
		F1.175494×10^{-38} to F3.402823×10^{38}	

Notes

1) **The number of points noted above is the number reserved as the calculation memory. The actual number of points available for use is determined by the hardware configuration.**
2) **If no battery is used, only the fixed area is backed up. (counters 16 points: C1008 to C1023, internal relays 128 points: R2480 to R255F, data registers 55 words: DT32710 to DT32764).**
   Writing is available up to 10000 times. Then the optional battery is used, all area can be backed up. Areas to be held and not held can be specified using the system registers. If an area is held when the battery is not installed, the value of data may be indefinite as it is not cleared to 0 when the power is turned on. When the battery ran out of the power, the data at the hold area will be indefinite.
3) **Note3)The points for the timer and counter can be changed by the setting of system register 5. The number given in the table are the numbers when system register 5 is at its default setting.**

1.1.4 FP−X

Item		Number of points and range of memory area available for use		Function
		C14	C30, C60	
Relay	External input (X) Note1)	1760 points (X0 to X109F)		Turns on or off based on external input.
	External output (Y) Note1)	1760 points (Y0 to Y109F)		Externally outputs on or off state
	Internal relay (R) Note2)	4096 points (R0 to R255F)		Relay which turns on or off only within program.
	Link relay (L) Note2)	2048 points (L0 to L127F)		This relay is a shared relay used for PLC link.
	Timer (T) Note2)	1024 points (T0 to T1007/C1008 to C1023) Note3)		This goes on when the timer reaches the specified time. It corresponds to the timer number.
	Counter (C) Note2)			This goes on when the counter increments. It corresponds to the counter number.
	Special internal relay (R)	192 points (R9000 to R911F)		Relay which turns on or off based on specific conditions and is used as a flag.
Memory area	External input (WX) Note1)	110 words (WX0 to WX109)		Code for specifying 16 external input points as one word (16 bits) of data.
	External output (WY) Note1)	110 words (WY0 to WY109)		Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (WR) Note2)	256 words (WR0 to WR255)		Code for specifying 16 internal relay points as one word (16 bits) of data.
	Link relay (WL)	128 words (WL0 to WL127)		Code for specifying 16 link relay points as one word (16 bits) of data.
	Data register (DT) Note2)	12285 words (DT0 to DT12284)	32765 words (DT0 to DT32764)	Data memory used in program. Data is handled in 16−bit units (one word).
	Link register (LD) Note2)	256 words (LD0 to LD255)		This is a shared data memory which is used within the PLC link. Data is handled in 16−bit units (one word).
	Timer/Counter set value area (SV) Note2)	1024 words (SV0 to SV1023)		Data memory for storing a target value of a timer and setting value of a counter. Stores by timer/counter number.
	Timer/Counter elapsed value area (EV) Note2)	1024 words (EV0 to EV1023)		Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/counter number.
	Special data register (DT)	374 words (DT90000 to DT90373)		Data memory for storing specific data. Various settings and error codes are stored.
	Index register (I)	14 words (I0 to ID)		Register can be used as an address of memory area and constants modifier.

Item		Number of points and range of memory area available for use		Function
		C14	C30, C60	
Control instruction point	Differential points	Unlimited points		
	Master control relay points (MCR)	256 points		
	Number of labels (JP and LOOP)	256 points		
	Number of step ladders	1000 stages		
	Number of subroutines	500 subroutines		
	Number of interrupt programs	Input 14 programs, periodical interrupt 1 program		
Constant	Decimal constants (K)	K−32, 768 to K32, 767 (for 16−bit operation)		
		K−2, 147, 483, 648 to K2, 147, 483, 647 (for 32−bit operation)		
	Hexadecimal constants (H)	H0 to HFFFF (for 16−bit operation)		
		H0 to HFFFFFFFF (for 32−bit operation)		
	Floating point type (F)	F$-1.175494 \times 10^{-38}$ to F-3.402823×10^{38}		
		F1.175494×10^{-38} to F3.402823×10^{38}		

Notes

1) **The number of points noted above is the number reserved as the calculation memory. The actual number of points available for use is determined by the hardware configuration.**
2) **If no battery is used, only the fixed area is backed up. (counters 16 points: C1008 to C1023, internal relays 128 points: R2470 to R255F, data registers 55 words, C14: DT12230 to DT12284, C30/C60: DT32710 to DT32764). Writing is available up to 10000 times. Then the optional battery is used, all area can be backed up.**
   Areas to be held and not held can be specified using the system registers. If an area is held when the battery is not installed, the value of data may be indefinite as it is not cleared to 0 when the power is turned on. When the battery ran out of the power, the data at the hold area will be indefinite.
3) **The points for the timer and counter can be changed by the setting of system register 5. The number given in the table are the numbers when system register 5 is at its default setting.**

1.1.5 FP2

Item		Numbering	Function
Relay	**External input relay (X)**	2,048 points (X0 to X127F)	Turn on or off based on external input.
	External output relay (Y)	2,048 points (Y0 to Y127F)	Externally outputs on or off state.
	Internal relay (R) (* Note 1)	4,048 points (R0 to R252F)	Relay which turns on or off only within program.
	Link relay (L) (* Note 1)	2,048 points (L0 to L127F)	This relay is a shared relay used for MEWNET link system.
	Timer (T) (* Notes 1 and 2)	1,024 points (T0 to T999/ C1000 to C1023)	If a **TM** instruction has timed out, the contact with the same number turns on.
	Counter (C) (* Notes 1 and 2)		If a **CT** instruction has counted up, the contact with the same number turns on.
	Pulse relay (P)	1,024 points (P0 to P63F)	This relay is used to turn on only for one scan duration programmed with the **OT**" and **OT**# instructions.
	Special internal relay (R)	176 points (R9000 to R910F)	Relay which turns on or off based on specific conditions and is used as a flag.
Memory area	**External input relay (WX)**	128 words (WX0 to WX127)	Code for specifying 16 external input points as one word (16 bits) of data.
	External output relay (WY)	128 words (WY0 to WY127)	Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (WR)	253 words (WR0 to WR252)	Code for specifying 16 internal relay points as one word (16 bits) of data.
	Link relay (WL)	128 words (WL0 to WL127)	Code for specifying 16 link relay points as one word (16 bits) of data.
	Data register (DT) (* Note 1)	6,000 words (DT0 to DT5999)	Data memory used in program. Data is handled in 16-bit units (one word).
	Link data register (LD) (* Note 1)	256 words (LD0 to LD255)	This is a shared data memory which is used within the MEWNET link system. Data is handled in 16-bit units (one word).
	Timer/Counter set value area (SV) (* Note 1)	1,024 words (SV0 to SV1023)	Data memory for storing a target value of a timer and an initial value of a counter. Stores by timer/counter number.
	Timer/Counter elapsed value area (EV) (* Note 1)	1,024 words (EV0 to EV1023)	Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/ counter number.
	File register (FL) (* Notes 1 and 3)	FP2 (16 K): 0 to 14,333 words (FL0 to FL14332) FP2 (32 K) (when expanded): 0 to 30,717 words (FL0 to FL30716)	Data memory used in program. Data is handled in 16-bit units (one word).
	Special data register (DT)	256 words (DT90000 to DT90255)	Data memory for storing specific data. Various settings and error codes are stored.
	Index register (I)	14 words (I0 to ID)	Register can be used as an address of memory area and constants modifier.

Item		Numbering
Control instruction point	Master control relay points (MCR)	256 points
	Number of labels (JP and LOOP)	Total: 256 points
	Number of step ladder (* Note 4)	1,000 steps
	Number of subroutine	100 subroutines
	Number of interrupt program	1 program (periodical interrupt: allows setting of the time interval within the range from 0.5ms to 1.5s)
Constant	Decimal constants (K)	K–32768 to K32767 (for 16-bit operation)
		K–2147483648 to K2147483647 (for 32-bit operation)
	Hexadecimal constants (H)	H0 to HFFFF (for 16-bit operation)
		H0 to HFFFFFFFF (for 32-bit operation)
	Floating point type (f)	f$-1.175494 \times 10^{-38}$ to f-3.402823×10^{38}
		f1.175494×10^{-38} to f3.402823×10^{38}

Notes

1) **There are two unit types, the hold type that saves the conditions that exist just before turning the power off or changing from the RUN mode to PROG. mode, and the non-hold type that resets them. The selection of hold type and non-hold type can be changed by the setting of system register.**

2) **The points for the timer and counter can be changed by the setting of system register 5. The numbers given in the table are numbers when system register 5 is at its default setting.**

3) **The size of the file register varies depending on the settings of system registers 0, 1 and 2.**

4) **Hold or non-hold type can be set using the system registers.**

1.1.6 FP2SH

Item		Numbering	Function
Relay	**External input relay (X)**	8,192 points (X0 to X511F)	Turn on or off based on external input.
	External output relay (Y)	8,192 points (Y0 to Y511F)	Externally outputs on or off state.
	Internal relay (R) **(* Note 1)**	14.192 points (R0 to R886F)	Relay which turns on or off only within program.
	Link relay (L) **(* Note 1)**	10,240 points (L0 to L639F)	This relay is a shared relay used for MEWNET link system.
	Timer (T) **(* Notes 1 and 2)**	3,072 points (T0 to T2999/ C3000 to C3071)	If a **TM** instruction has timed out, the contact with the same number turns on.
	Counter (C) **(* Notes 1 and 2)**		If a **CT** instruction has counted up, the contact with the same number turns on
	Pulse relay (P)	2,048 points (P0 to P127F)	This relay is used to turn on only for one scan duration programmed with the **OT**" and **OT**# instructions.
	Error alarm relay (E)	2,048 points (E0 to E2047)	If turned on while the unit is running, this relay stores the history in a dedicated buffer. Program this relay so that it is turned on at the time of abnormality.
	Special internal relay (R)	176 points (R9000 to R910F)	Relay which turns on or off based on specific conditions and is used as a flag.
Memory area	**External input relay (WX)**	512 words (WX0 to WX511)	Code for specifying 16 external input points as one word (16 bits) of data.
	External output relay (WY)	512 words (WY0 to WY511)	Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (WR)	887 words (WR0 to WR886)	Code for specifying 16 internal relay points as one word (16 bits) of data.
	Link relay (WL)	640 words (WL0 to WL639)	Code for specifying 16 link relay points as one word (16 bits) of data.
	Data register (DT) **(* Note 1)**	10,240 words (DT0 to DT10239)	Data memory used in program. Data is handled in 16-bit units (one word).
	Link data register (LD) **(* Note 1)**	8,448 words (LD0 to LD8447)	This is a shared data memory which is used within the MEWNET link system. Data is handled in 16-bit units (one word).
	Timer/Counter set value area (SV) **(* Note 1)**	3,072 words (SV0 to SV3071)	Data memory for storing a target value of a timer and an initial value of a counter. Stores by timer/counter number.
	Timer/Counter elapsed value area (EV) **(* Note 1)**	3,072 words (EV0 to EV3071)	Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/ counter number.
	File register (FL) **(* Note 1)**	98,295 words (32,765 words ×3 banks)	Data memory used in program. Data is handled in 16-bit units (one word).

Item		Numbering	Function
Memory area	**Special data register (DT)**	512 words (DT90000 to DT90511)	Data memory for storing specific data. Various settings and error codes are stored.
	Index register (I)	14 words × 16 banks (I0 to ID)	Register can be used as an address of memory area and constants modifier.
Control instruction point	**Master control relay points (MCR)**	256 points (For FP2−C3P: 1st program: 256 points/2nd program: 256 points)	
	Number of labels (JP and LOOP)	256 points (For FP2−C3P: 1st program: 256 points/2nd program: 256 points)	
	Number of step ladder (* Note 3)	1,000 steps (For FP2−C3P: 1st program only)	
	Number of subroutine	100 subroutines	
	Number of interrupt program	1 program (periodical interrupt: allows setting of the time interval within the range from 0.5ms to 1.5s) (For FP2−C3P: 1st program only)	
Constant	**Decimal constants (K)**	K−32768 to K32767 (for 16-bit operation)	
		K−2147483648 to K2147483647 (for 32-bit operation)	
	Hexadecimal constants (H)	H0 to HFFFF (for 16-bit operation)	
		H0 to HFFFFFFFF (for 32-bit operation)	
	Floating point type (f)	f−1.175494 × 10^{-38} to f−3.402823 × 10^{38}	
		f1.175494 × 10^{-38} to f3.402823 × 10^{38}	

Notes

1) **There are two unit types, the hold type that saves the conditions that exist just before turning the power off or changing from the RUN mode to PROG. mode, and the non-hold type that resets them. The selection of hold type and non-hold type can be changed by the setting of system register.**

2) **The points for the timer and counter can be changed by the setting of system register 5. The numbers given in the table are numbers when system register 5 is at its default setting.**

3) **Hold or non-hold type can be set using the system registers.**

1.1.7 FP10SH

Item		Numbering	Function
Relay	**External input relay (X)**	8,192 points (X0 to X511F)	Turn on or off based on external input.
	External output relay (Y)	8,192 points (Y0 to Y511F)	Externally outputs on or off state.
	Internal relay (R) **(* Note 1)**	14,192 points (R0 to R886F)	Relay which turns on or off only within program.
	Link relay (L) **(* Note 1)**	10,240 points (L0 to L639F)	This relay is a shared relay used for MEWNET link system.
	Timer (T) **(* Notes 1 and 2)**	3,072 points (T0 to T2999/ C3000 to C3071)	If a **TM** instruction has timed out, the contact with the same number turns on.
	Counter (C) **(* Notes 1 and 2)**		If a **CT** instruction has counted up, the contact with the same number turns on.
	Pulse relay (P)	2,048 points (P0 to P127F)	This relay is used to turn on only for one scan duration programmed with the **OT**" and **OT**# instructions.
	Error alarm relay (E)	2,048 points (E0 to E2047)	If turned on while the unit is running, this relay stores the history in a dedicated buffer. Program this relay so that it is turned on at the time of abnormality.
	Special internal relay (R)	176 points (R9000 to R910F)	Relay which turns on or off based on specific conditions and is used as a flag.
Memory area	**External input relay (WX)**	512 words (WX0 to WX511)	Code for specifying 16 external input points as one word (16 bits) of data.
	External output relay (WY)	512 words (WY0 to WY511)	Code for specifying 16 external output points as one word (16 bits) of data.
	Internal relay (WR)	887 words (WR0 to WR886)	Code for specifying 16 internal relay points as one word (16 bits) of data.
	Link relay (WL)	640 words (WL0 to WL639)	Code for specifying 16 link relay points as one word (16 bits) of data.
	Data register (DT) **(* Note 1)**	10,240 words (DT0 to DT10239)	Data memory used in program. Data is handled in 16-bit units (one word).
	Link data register (LD) **(* Note 1)**	8,448 words (LD0 to LD8447)	This is a shared data memory which is used within the MEWNET link system. Data is handled in 16-bit units (one word).
	Timer/Counter set value area (SV) **(* Note 1)**	3,072 words (SV0 to SV3071)	Data memory for storing a target value of a timer and an initial value of a counter. Stores by timer/counter number.
	Timer/Counter elapsed value area (EV) **(* Note 1)**	3,072 words (EV0 to EV3071)	Data memory for storing the elapsed value during operation of a timer/counter. Stores by timer/ counter number.
	File register (FL) **(* Note 1)**	32,765 words (FL0 to FL32764)	Data memory used in program. Data is handled in 16-bit units (one word).

Item		Numbering	Function
Memory area	Special data register (DT)	512 words (DT90000 to DT90511)	Data memory for storing specific data. Various settings and error codes are stored.
	Index register (I)	14 words × 16 banks (I0 to ID)	Register can be used as an address of memory area and constants modifier.
Control instruction point	Master control relay points (MCR)	256 points (when using the 90k step expansion memory, up to a total of 512 points can be used for the 1st and 2nd programs)	
	Number of labels (JP and LOOP)	256 points (when using the 90k step expansion memory, up to a total of 512 points can be used for the 1st and 2nd programs)	
	Number of step ladder (* Note 3)	1,000 steps (can only be used for the 1st program)	
	Number of subroutine	100 subroutines (can only be used for the 1st program)	
	Number of interrupt program	25 program (can only be used for the 1st program)	
Constant	Decimal constants (K)	K–32768 to K32767 (for 16-bit operation)	
		K–2147483648 to K2147483647 (for 32-bit operation)	
	Hexadecimal constants (H)	H0 to HFFFF (for 16-bit operation)	
		H0 to HFFFFFFFF (for 32-bit operation)	
	Floating point type (f)	f–1.175494 $\times$ 10^{-38} to f–3.402823 $\times$ 10^{38} f1.175494 $\times$ 10^{-38} to f3.402823 $\times$ 10^{38}	

Notes

1) **There are two unit types, the hold type that saves the conditions that exist just before turning the power off or changing from the RUN mode to PROG. mode, and the non-hold type that resets them. The selection of hold type and non-hold type can be changed by the setting of system register.**

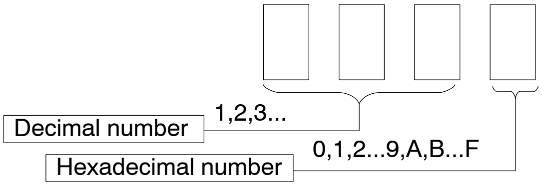
2) **The points for the timer and counter can be changed by the setting of system register 5. The numbers given in the table are numbers when system register 5 is at its default setting.**

3) **Hold or non-hold type can be set.**

1.1.8 Relay Numbers

External input relays (X), External output relays (Y), Internal relays (R), Link relays (L) and Pulse relays (P)

Since these relays are handled in units of 16 points, they are expressed as a combination of decimal and hexadecimal numbers as shown below.



The maximum value that can be selected varies with each relay.

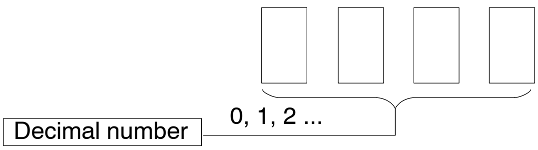
Example: External input relay (X)

X0, X1 .... XF
X10, X11 .... X1F
X20, X21 .... X2F
....

Timers (T) and Counters (C)

The addresses for timer contacts (T) and counter contacts (C) are correspond to the timer and counter instruction numbers and expressed in decimals as shown below.

Example: FP2



T0, T1 .... T999
C1000, C1001 .... C1023

Note

Counters and timers share the same area. The division of the area can be changed with system register 5. (The table and example are when settings are the default values.)

Error alarm relays (E) (FP2SH/FP10SH only)

The addresses for error alarm relays (E) are represented in only decimals.

E0, E1 .... E2047

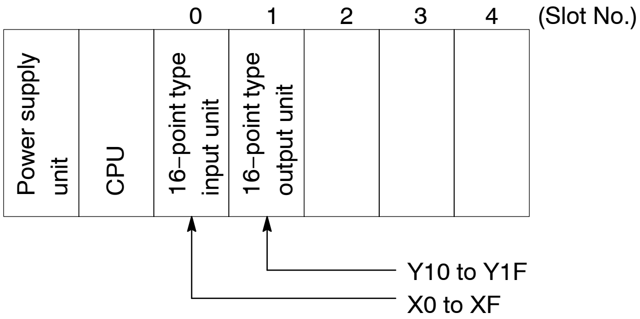
External input relay (X) and External output relay (Y)

Only relays with numbers actually allocated to input contacts can be used as external input relay (X).

Only relays with numbers actually allocated to output contacts can output as external output relay (Y). The external output relays (Y) which are not allocated can be used as internal relays.

Allocation of numbers is determined by the combination of units and boards used. For details about the I/O allocation, refer to “Hardware Manual” of each PLC.

Example: **FP2**



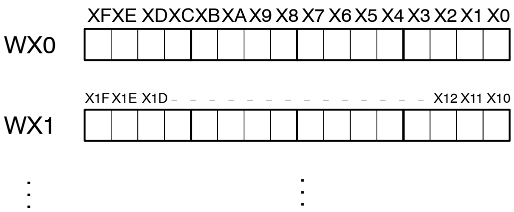
The 16 points external input relays X0 through XF are allotted for the 16-point type input unit for slot 0, and the 16 points external output relays Y10 through Y1F are allotted for the 16-point type output unit for slot 1.

The 16 points X10 through X1F cannot be used in this such combination.

Relation of WX, WY, WR and WL to X, Y, R and L

WX, WY WR and WL correspond respectively to groups of 16 external input relay (X) points, 16 external output relay (Y) points, 16 internal relay (R) points and 16 link relay (L) points.

Example: **Word external input relay (WX)**
Each relay is composed of 16 external input relay (X) points as shown below.



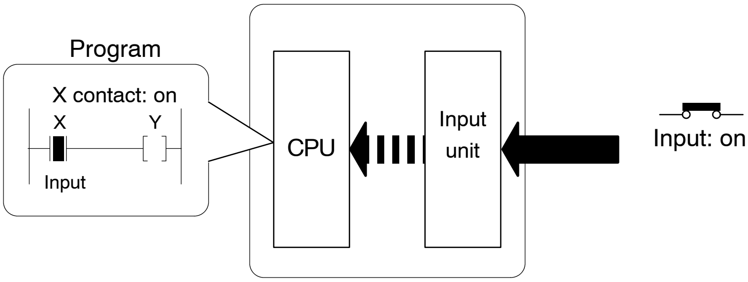
When the state of an external input relay (X) changes, the content of WX also changes.

1.2 Explanation of Relays

1.2.1 External Input Relays (X)

Function of external input relays (X)

This relay feeds signals to the programmable controller from an external device such as a limit switch or a photoelectric sensor.



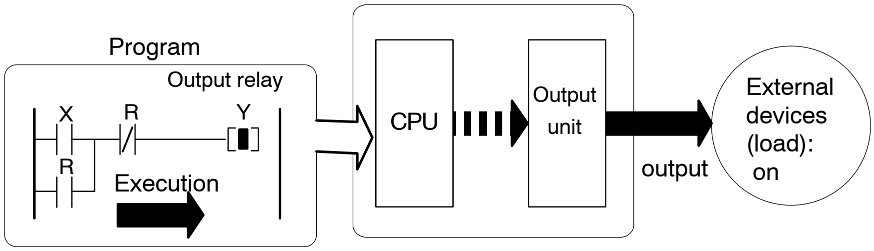
Usage restrictions

The addresses for inputs which do not actually exist cannot be used.
The on or off status of the external input relays cannot be changed by the program in the programmable controller.
There are no restrictions on the number of times one external input relay is programmed.

1.2.2 External Output Relays (Y)

Function of external output relays (Y)

This relay outputs the program execution result of the programmable controller and activates an external device (load) such as a solenoid, operating panel or intelligent unit.
The on or off status of the external output relay is output as a control signal.

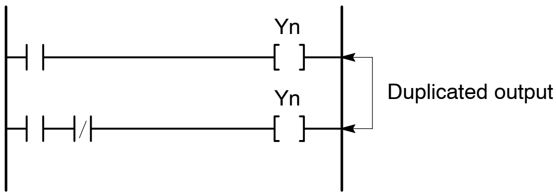


Usage restrictions

External output relays which are not actually allocated can be used in the same way as internal relays. However, they cannot be specified as hold types.
When used as contacts, there are no restrictions on the number of times that can be used.
As a rule, when specified as the output destination for operation results of **OT** and **KP** instructions, use is limited to once in a program (to inhibit double output).

Note

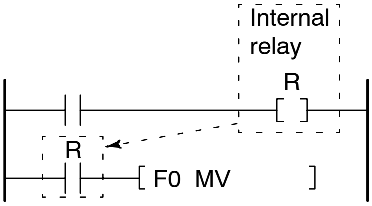
You can permit duplicated use of an output by changing the system register 20 setting. Even if the same relay is used as an operand for instructions such as SET and RST, it is not regarded as duplicated use of outputs.



1.2.3 Internal Relays (R)

Function of internal relays (R)

This relay can be used only within program and on or off status does not provide an external output. When the coil of the relay is energized, its contacts turn on.



Usage restrictions

When used as contacts, there are no restrictions on the number of times that can be used.
As a rule, when specified as the output destination for operation results of **OT** and **KP** instructions, use is limited to once in a program (to inhibit double output).

Note

You can permit duplicated use of an output by changing the system register 20 setting. Even if the same relay is used as an operand for instructions such as SET and RST, it is not regarded as duplicated use of outputs.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, all of the internal relays (R) go off. If a hold type has been specified (see next page), the internal relays (R) go off as well.

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the relays are not cleared even if the Initialize/Test switch is set to the upper side.

Non−hold type relay and hold type relay

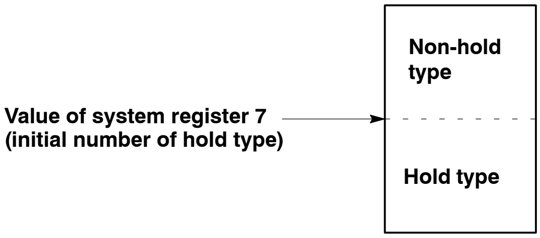
There are two types of internal relays: hold type relays and non−hold type relays. When the power is turned off or the mode changed from RUN to PROG.,

- Hold type relays hold their on or off status and resume operation in that status when the system is restarted.
- Non−hold type relays reset.

For the FP0 C10/C14/C16/C32, and FP−e without clock/calendar function, non−hold type and hold type relay numbers are as follows:

Item	Non-hold type	Hold type
FP0 C10, C14, C16 **FP−e**	R0 to R60F (976 points)	R610 to R62F (32 points)
FP0 C32	R0 to R54F (880 points)	R550 to R62F (128 points)

For the FP0 T32C/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH, and FP−e with clock/calendar function, system register 7 can be used to specify whether a hold type or a non−hold type is used. If the beginning of a hold type relay is specified using a word number, relays before that point will be non−hold types, and subsequent relays will be hold types.



Default settings for hold types and non−hold types

Type	Non−hold type	Hold type
FP10SH/FP2SH	R0 to R499F (8000 points)	R5000 to R886F (6192 points)
FP2	R0 to R199F (3200 points)	R2000 to R252F (848 points)
FP0 T32C	R0 to R9F (160 points)	R100 to R62F (848 points)
FPΣ	R0 to R89F (1440 points)	R900 to R97F (128 points)
FP−e	R0 to R60F (976 points)	R610 to R62F (32 points)
FP−X/FP0R	R0 to R247F (3968 points)	R2480 to R255F (128 points)

Note

For FP0R, FPΣ, FP−X and FP−e, in case of not using back−up battery, please keep the default value. Otherwise we cannot guarantee the function of hold/non−hold value.

1.2.4 Special Internal Relays

Function of special internal relays

The special internal relays turn on or off under specific conditions. The on or off state is not externally output and only functions within the program.
The principal special internal relays are as follows:

- **Operation status flags:**

  Operation status is indicated by on or off.
  - Operation (RUN mode) in progress (R9020)
  - Forced input/output in progress (R9029)
  - Link station operation (R9060 to R909F)
  - Turns on and off at each scan (R9012)
  - Result of comparison instruction (R900A to R900C)
  - High−speed counter control flag (R903A to R903D) and others
- **Error flags:**

  Turns on when an error occurs.
  - Operation error (R9007, R9008)
  - Shared memory access error (R9031) and others
- **Relays which turn on and off under special conditions:**

  The required conditions can be selected in the program and the relays used accordingly.
  - Always on relay (R9010)
  - Clock pulse relay (R9018 to R901E) and others

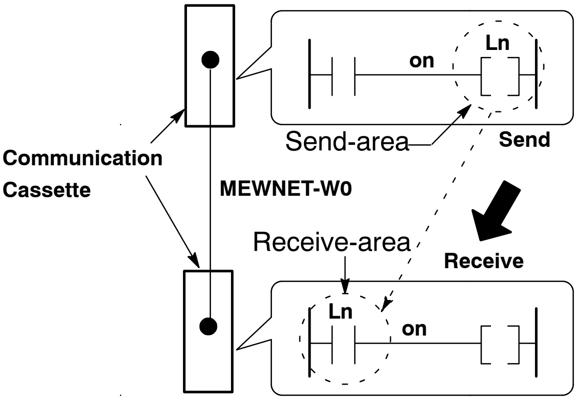
For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, special internal relays R9000 to R910F go off. If self−diagnostic error 44 or an error with a lower number has occurred, however, R9000 to R9008 are not cleared.

1.2.5 Link Relays (L) for FPΣ, FP−X, FP0R

Function of link relays (L)

Link relays are relays used for the PC Link, that can be shared between multiple programmable controllers when they are connected using a PLC link.

If calculation results are output to the link relay (coil) of a certain PLC, the results are also sent to other PLC connected with MEWNET, and will be reflected in link relay (contact) that have the same number.



When link relays are used, bit information can be exchanged in this way between PLCs.

Available range of link relays

The available range of link relays varies depending on the type of network and the combination of units. The available range and number of points must be specified separately for each network.

For MEWNET−W0:
A maximum of 1,024 points are available with one control unit. The available range is from L0 to L63F

Specifying hold type and non−hold type relays

There are two types of link relays, which can be switched when the power is turned off and the mode is switched from RUN to PROG and operation is stopped.

Hold type relays, which hold the on or off status in effect immediately prior to stopping, during the period between stopping and resuming operation

Non−hold type relays, which are reset when operation stops

In case of using back−up battery, System register 10 can be used to specify whether the link relays are the hold or non−hold type.

Range	System register no.
L0 to L63F	10

If the beginning of a hold type relay is specified using a word number, relays before that point will be non−hold types, and subsequent relays will be hold types.

For example, if “10” is set for system register 10, L0 to L9F will be non−hold types, and L100 to L63F will be hold types.

For the default value, all link relays are hold types.

If used as link relays for reception, be aware that no holding operation is carried out, even if the link relays are specified as hold types using the system registers.

Usage restrictions

When used as contacts, there are no restrictions on the number of times that can be used.
As a rule, when specified as the output destination for operation results of **OT** instruction and **KP** instruction, use is limited to once in a program (to inhibit double output).

Notes

- **System register 20 can be used to permit double output. Also, double output does not result if the SET and RST instructions are used.**
- **Link relays must be allocated when the network is configured, before programming is done. The method by which allocations are made varies depending on the type of network. Refer to the manual for the pertinent link unit.**

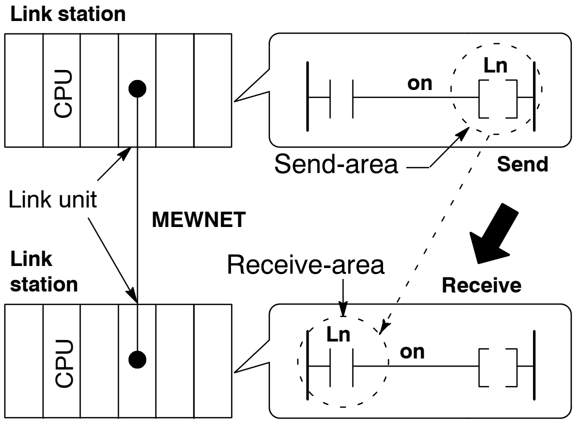
1.2.6 Link Relays (L) for FP2/FP2SH/FP10SH

Function of link relays (L)

Link relays are relays used for the PC Link, that can be shared between multiple programmable controllers when they are connected using a MEWNET link.
The following types of MEWNET links are available.

- MEWNET-H link system for FP10SH (for coaxial cables)
- MEWNET-W link system for FP2, FP2SH and FP10SH (for wire cables)
- MEWNET-P link system for FP10SH (for fiber−optic cables)

If calculation results are output to the link relay (coil) of a certain PLC, the results are also sent to other PLC connected with MEWNET, and will be reflected in link relay (contact) that have the same number.



When link relays are used, bit information can be exchanged in this way between PLCs.

Available range of link relays

The available range of link relays varies depending on the type of network and the combination of units. The available range and number of points must be specified separately for each network.

For MEWNET−W and MEWNET−P:
A maximum of 1,024 points are available with one link unit. The available range is from L0 to L63F for the first unit (PC Link 0), and from L640 to L127F to the second unit (PC Link 1).

For MEWNET−W2
A maximum of 4,096 points can be used per link unit. Please set the range of use at the MEWNET−W2 settings menu.

With the FP2SH, the range between L0 and L639F can be specified. When used with MEWNET−W the range between L0 and L127F cannot be used.

With the FP2, the range between L0 and L127F can be specified. Also, the internal relay can be used in place of the link relay by setting the MEWNET−W2 setting menu. However, when used with MEWNET−W the range between L0 and L127F cannot be used with MEWNET−W2.

For MEWNET−H:
A maximum of 10,240 points can be used. Please set the range to be used with the MEWNET−H link setting software.

With the FP10SH, the range from L0 to L639F can be used.
If used in conjunction with a MEWNET−W or MEWNET−P link unit, be aware that the range from L0 to L127F cannot be used.

Specifying hold type and non−hold type relays

There are two types of link relays, which can be switched when the power is turned off and the mode is switched from RUN to PROG and operation is stopped.

Hold type relays, which hold the on or off status in effect immediately prior to stopping, during the period between stopping and resuming operation

Non−hold type relays, which are reset when operation stops

System register 10, 11, and 16 can be used to specify whether the link relays are the hold or non−hold type.

Range	System register no.
L0 to L63F	10
L640 to L127F	11
L1280 to L639F	16

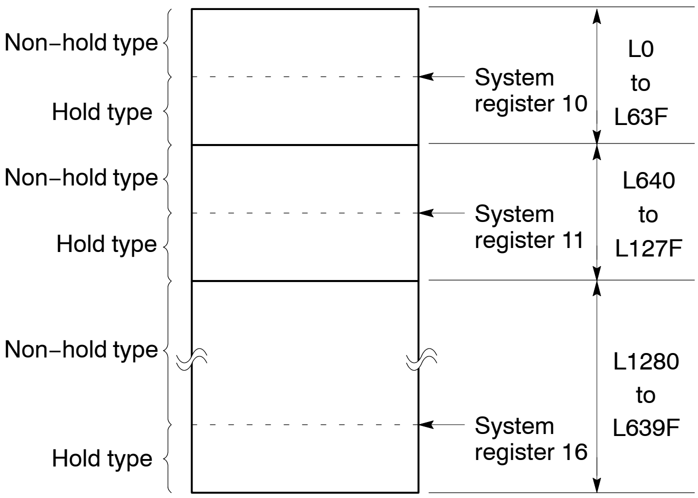
If the beginning of a hold type relay is specified using a word number, relays before that point will be non−hold types, and subsequent relays will be hold types.

For example, if “10” is set for system register 10, L0 to L9F will be non−hold types, and L100 to L63F will be hold types.

For the default value, all link relays are hold types.

If used as link relays for reception, be aware that no holding operation is carried out, even if the link relays are specified as hold types using the system registers.

Example:



Usage restrictions

When used as contacts, there are no restrictions on the number of times that can be used.
As a rule, when specified as the output destination for operation results of **OT** instruction and **KP** instruction, use is limited to once in a program (to inhibit double output).

Notes

- **System register 20 can be used to permit double output. Also, double output does not result if the SET and RST instructions are used.**
- **Link relays must be allocated when the network is configured, before programming is done. The method by which allocations are made varies depending on the type of network. Refer to the manual for the pertinent link unit.**

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, all of the link relays (L) go off. If a hold type has been specified (see next page), these relays go off as well.

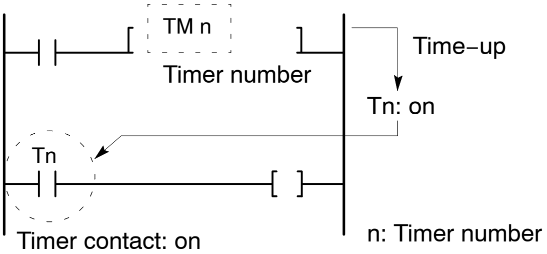
Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the relays are not cleared even if the Initialize/Test switch is set to the upper side.

1.2.7 Timer (T)

Function of timers (T)

When a timer is activated and the set time elapses, the timer contact with the same number as the timer turns on.
When the timer is in the time−up state and the timer execution condition turns off, the timer contact turns off.



Usage restrictions

When used as contacts, there are no restrictions on the number of times that can be used.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, the timer contact goes off. If a hold type has been specified, it goes off as well.

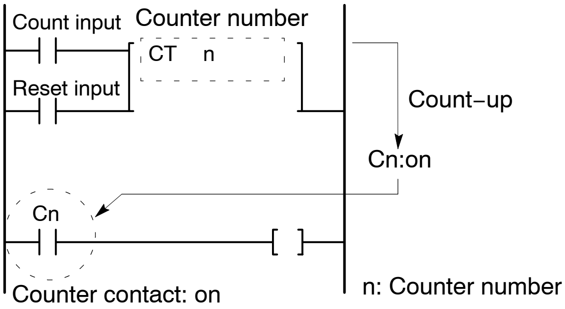
Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the timer contact is not cleared even if the Initialize/Test switch is set to the upper side.

1.2.8 Counter (C)

Function of counters (C)

When the decrement−type preset counter is activated and the elapsed value reaches zero, the counter contact with the same number as the counter turns on.
When the counter's reset input is turned on, the counter contact turns off.



Usage restrictions

When used as contacts, there are no resrictions on the number of times that can be used.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, the counter contact goes off. If a hold type has been specified, it goes off as well.

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the counter contact is not cleared even if the Initialize/ Test switch is set to the upper side.

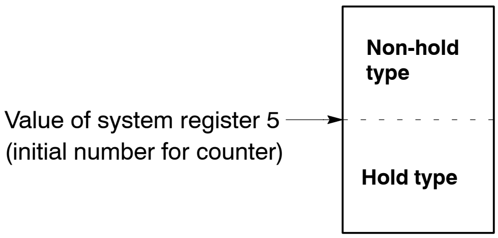
1.2.9 Items Shared by the Timer and Counter

Timer and counter partitioning

Timers and counters share the same area. The partitioning of the area can be changed to obtain the number of timers or counters needed.

Partition the area by setting system register 5. If the initial number of the counter is specified, those prior to that point will be timers, and those subsequent to that point will be counters.

If the same value is set for system register 5 and 6, timers are non−hold types, and counters are hold types. Normally, the same value should be set for both system registers.



Default settings for timer and counter

Type	Timer	Counter
FP2SH/FP10SH	T0 to T2999 (3000 points)	C3000 to C3071 (72 points)
FP2	T0 to T999 (1000 points)	C1000 to C1023 (24 points)
FPΣ, FP−X, FP0R	T0 to T1007 (1008 points)	C1008 to C1023 (16 points)
FP0, FP−e	T0 to T99 (100 points)	C100 to C143 (44 points)

Hold type and non−hold type partitioning

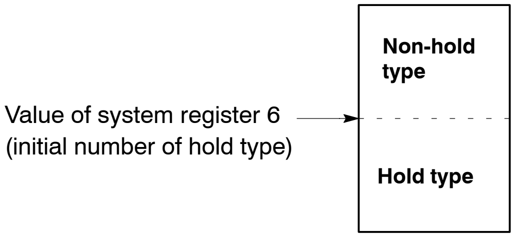
The contents of timer contacts, counter contacts, set value areas and elapsed value areas can be held when the power is turned off or the mode switched from RUN to PROG., and operation later resumed based on those contents.

In the case of the FP0 C10/C14/C16/C32, and FP−e without clock/calendar function, the areas which hold their contents when the power is turned off are fixed as shown below. System register settings 6 to 8 as well as 14 become invalid.

Timer	Non−hold type: all points
Counter	Non−hold type FP0 C10, C14, C16 FP−e: From set value to C139 FP0 C32: From set value to C127
	Hold type FP0 C10, C14, C16 FP−e: C140 to C143 FP0 C32: C128 to C143

For the FP0 T32C/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH, and FP−e with clock/calendar function, system register 6 can be used to specify whether a hold type or a non−hold type is used. If the beginning of a hold type is specified using a word number, the contents of timer/counter contacts and set value/elapsed value areas before that point will be non−hold types, and subsequent the contents of timer/counter contacts and set value/elapsed value areas will be hold types.

Even if specifying for the unit without batteries, the data will be indefinite.



Default settings for hold types and non−hold types

Type	Non−hold type	Hold type
FP2SH/FP10SH	0 to 2999 (3000 points)	3000 to 3071 (72 points)
FP2	0 to 999 (1000 points)	1000 to 1023 (24 points)
FPΣ, FP−X, FP0R	0 to 1007 (1008 points)	1008 to 1023 (16 points)
FP−e	0 to 139 (140 points)	140 to 143 (4 points)
	SV: non−hold *1	SV: hold
FP0 T32C	0 to 99 (100 points)	100 to 143 (44 points)

Note

For FP0R, FPΣ, FP−X and FP−e, in case of not using back−up battery, please keep the default value. Otherwise we cannot guarantee the function of hold/non−hold value.

***1 Use the following methods for holding the SV data:**
- **Set the transfer instruction for the data register (DT) to hold the data. Then, perform the setting so that the data can be transferred from DT to SV after the RUN mode starts.**
- **Use the FP−e model with a battery.**

For the FP2/FP2SH/FP10SH, contacts of timers and counters specified as hold types, as well as setting value areas and elapsed value areas, are cleared to 0 when the Initialize/Test switch is set to the upper side (the Initialize side).

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the counter contact is not cleared even if the Initialize/Test switch is set to the upper side.

1.2.10 Pulse Relays (P)

Note

Pulse relays (P) can only be used with the FP2/FP2SH/FP10SH.

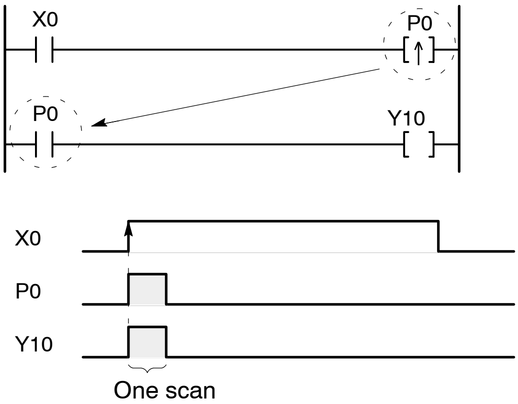
Function of pulse relays (P)

A pulse relay (P) goes on for one scan only. The on or off state is not externally output and only operates in the program.

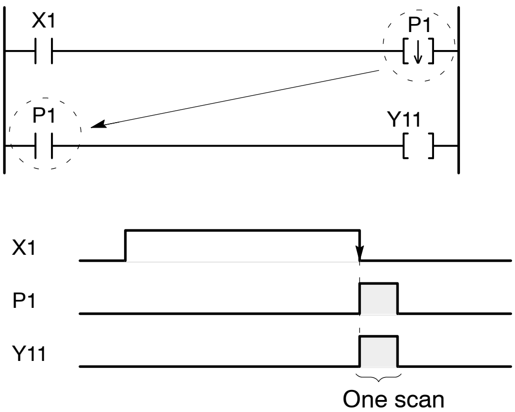
A pulse relay only goes on when a leading edge start instruction (**OT**↑) or a trailing edge start instruction (**OT**↓) is executed.

When used as the trigger, a pulse relay only operates during one scan when leading edge or trailing edge is detected.

Example 1: Differential execution when input X0 rises



Example 2: Differential execution when input X1 falls



Usage restrictions

Pulse relays are cleared when the power is turned off.

A pulse relay can only be used once in a program as an output destination for an **OT**↑ or **OT**↓ instruction (double output is prohibited).

There is no limitation to the number of times a pulse relay can used as a contact.

A pulse relay cannot be specified as an output destination for an **OT**, **KP**, **SET**, **RST** or **ALT** instruction.

A word unit pulse relay (WP) cannot be specified as a storage location for a high-level instruction.

1.2.11 Error Alarm Relays (E)

Note

Error alarm relays can only be used with the FP2SH/FP10SH.

Function of error alarm relays (E)

Error alarm relays are used to feed back error conditions freely assigned by the user to internal relays, and to store them in memory.

Error alarm relays are turned on and off using the **SET** and **RST** instructions in the user program.

When an error alarm relay goes on, the number of error alarm relays which are on, the relay numbers, and the data of the calendar timer which went on first are stored in a memory area in the CPU unit.

DT90400	No. of relays which are on	
DT90401 to DT90419	Relay numbers which are on	
DT90420	Min./sec. data	Data of calendar timer for first relay to go on
DT90421	Day/time data	
DT90422	Year/month data	

Information for up to 500 error alarm relays can be stored in the memory area. Those which can be monitored or operated by the user, however, are those in the range from DT90401 to DT90419 only.

Usage restrictions and precautions

Error alarm relay (E) cannot be specified as the output destination for the **OT**, **KP**, or **ALT** instructions.

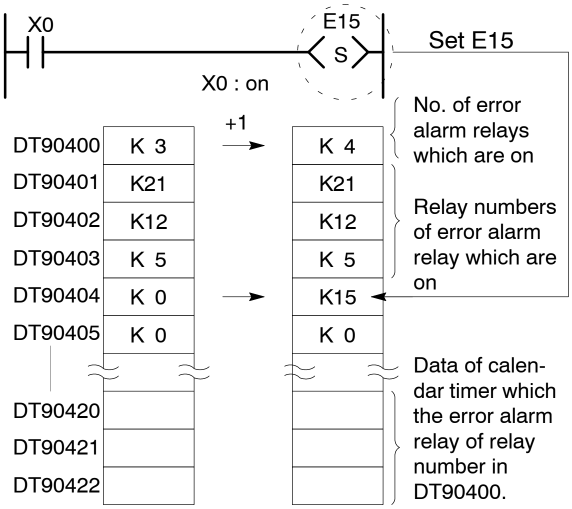
Error alarm relay (E) can be turned on and off in multiple locations in the program, using the **SET** and **RST** instructions. However, no check is carried out for overlapping use.

Program for setting (turning on) an error alarm relays

The **SET** instruction should be used to turn on error alarm relays in the error alarm conditions.

Error alarm relays are held even if the error condition goes off.

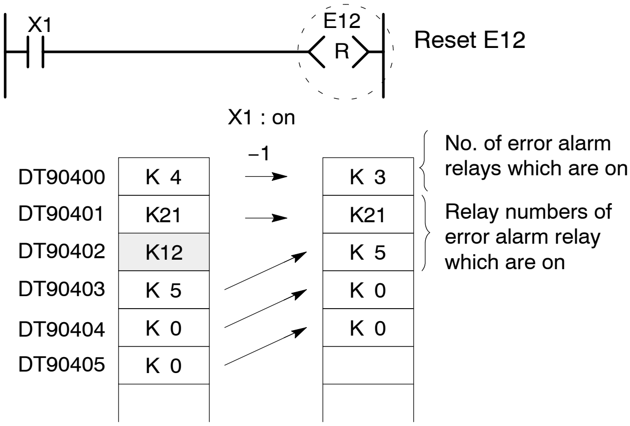
Example: If X0 goes on when an error occurs



Program for resetting (turning off) an error alarm relay

When an error has been corrected, the **RST** instruction should be used to turn off the error alarm relay.

Example: If X1 goes on when an error is corrected



Clearing all buffer areas

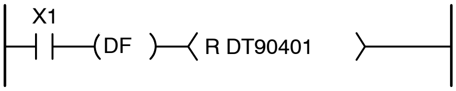
Either of the following methods may be used.

- To reset all of the error alarm relays, use the **RST** instruction in the same way as that described on the next page, and specify special data register DT90400.
- If the Initialize/Test switch is set to the Initialize side in the PROG mode, all error alarm relays (E) go off, and the storage buffer is cleared.
  (To avoid clearing the buffer with the Initialize switch, change the setting of system register 4.)

Clearing buffer areas and initial data

Of the areas in which relay numbers are stored, only DT90400 and DT90401 can be cleared by directly specifying the special data register with the **RST** instruction.
If DT90400 is specified, all error information in the buffer is cleared, and if DT90401 is specified, the initial relay number in the buffer area is cleared. Buffers fill up as shown in the example below.

Example: When the contents of DT90401 are deleted using the RST instruction



Range visible to user	Storage buffer 1	K 1	DT90401
	Storage buffer 2	K 2	DT90402
	Storage buffer 3	K 3	DT90403
	Storage buffer 4	K 4	DT90404
	Storage buffer 17	K17	DT90417
	Storage buffer 18	K18	DT90418
	Storage buffer 19	K19	DT90419
Range not visible to user	Storage buffer 20	K20	
	Storage buffer 21	K21	
	Storage buffer 499	K499	
	Storage buffer 500	K500	

Range visible to user	Storage buffer 1	K 2	DT90401
	Storage buffer 2	K 3	DT90402
	Storage buffer 3	K 4	DT90403
	Storage buffer 4	K 5	DT90404
	Storage buffer 17	K18	DT90417
	Storage buffer 18	K19	DT90418
	Storage buffer 19	K20	DT90419
Range not visible to user	Storage buffer 20	K21	
	Storage buffer 21	K22	
	Storage buffer 499	K500	
	Storage buffer 500		

1.3 Explanation of Memory Areas

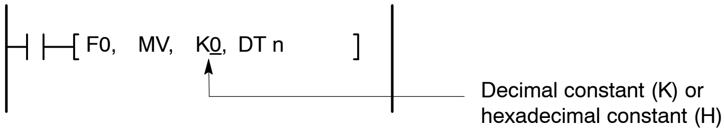
1.3.1 Data Register (DT)

Function of data registers (DT)

Data registers are memory areas which are handled in word (16−bit) units, and are used to store data such as numerical data configured of 16 bits.

Bit position	15	.	.	12	11	.	.	8	7	.	.	4	3	.	.	0
DTn	0	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0

Example of a program which writes a numeric value to DTn.



When 32-bit (double word) data is handled in data registers, use two data registers as a set. The number of the data register for the lower 16 bits is specified.

DTn+1																DTn															
0	0	0	1	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0

higher 16-bit area　　　　lower 16-bit area

Non–hold type data and hold–type data

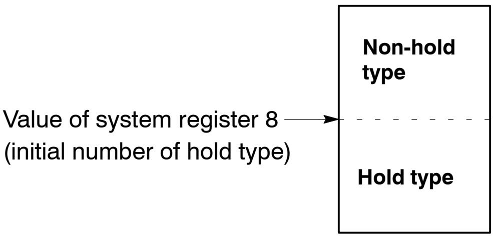
There are two types of data registers which handle data differently when the power is turned off or the mode is changed from RUN to PROG.:

- Hold type data registers hold their contents while operation stops and allow operation to be restarted with the contents still effective.
- Non–hold type data registers reset when operation stops.

For the FP0 C10/C14/C16/C32, and FP–e without clock/calendar function, non–hold type and hold type data register numbers are as shown in the following table.

Item		FP0 C10/C14/C16 and FP–e	FP0 C32
Data register	**Non–hold type**	1652 words (DT0 to DT1651)	6112 words (DT0 to DT6111)
	Hold type	8 words (DT1652 to DT1659)	32 words (DT6112 to DT6143)

For the FP0 T32C/FPΣ/FP0R/FP–X/FP2/FP2SH/FP10SH, and FP–e with clock/calendar function, system register 8 can be used to specify whether hold types or non–hold types are to be used. If the beginning of a hold type data register is specified using a word number, data registers before that point will be non–hold types, and subsequent data registers will be hold types.



Default settings for hold types and non–hold types

Type	Non–hold type	Hold type
FPΣ, FP–X (C30, C60)	DT0 to DT32709 (32710 words)	DT32710 to DT32765 (55 words)
FP–X (C14)	DT0 to DT12229 (12230 words)	DT12230 to DT12284 (55 words)
FP–e	DT0 to DT1651 (1652 words)	DT1652 to DT1659 (8 words)
FP0R C10, C14, C16	DT0 to DT11999 (12315 words)	DT12000 to DT12314 (315 words)
FP0R C32, T32, F32	DT0 to DT32449 (32451 words)	DT32450 to DT32764 (315 words)

Note

For FPΣ, FP–X and FP–e, in case of not using back–up battery, please keep the default value. Otherwise we cannot guarantee the function of hold/non–hold value.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, all data registers (DT) are cleared to 0. Even if a hold type has been specified, these are cleared to 0.

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the data registers are not cleared even if the Initialize/ Test switch is set to the upper side.

1.3.2 Special Data Registers (DT)

Function of the special data registers

These data registers have specific applications.
Data cannot be written to most of them using instructions such as **F0 (MV)**.
With the FP0 T32C, FP0R, FPΣ, FP−X, FP2, FP2SH, FP10SH and the FP0 C10/C14/C16/ C32, FP−e, the special data registers have different numbers, but the last three digits of the numbers are the same.

Example:

With the FP0 C10/C14/ C16/C32, FP−e: **D T 9 0 5 5**

With the FP0 T32C, FP0R FPΣ, FP−X, FP2, FP2SH and FP10SH: **D T 9 0 0 5 5**

The last three digits are the same.

If a “0” is added, this becomes the five−digit number used with the FP0 T32C, FPΣ, FP−X, FP0R, FP2, FP2SH and FP10SH.

The main functions of special data registers are:

Environmental settings and operation statuses

The operation statuses of the programmable controller specified with the system registers and the various types of instructions are stored.

- Link communication status (DT9140 to DT9254/DT90140 to DT90254)
- High−speed counter control flag (DT9052/DT90052) and others

Error contents

The unit in which the error occurred, and other information, is stored.

- Self−diagnostic error code (DT9000/DT90000)
- The slot number of the unit where the error occurred (DT9002, DT9003, etc.)
- Remote input/output error slave station numbers (DT9131 to DT9135)
- The address where the operation error occurred (DT9017, DT9018/DT90017, DT90018)

Clock/calendar

(can be used with all types of the FP0 T32C, FP0R, FP−e, FPΣ, FP−X, FP2, FP2SH and FP10SH)

The year, month, day, hour, minute, second, and day of the week tracked by the calendar timer are stored here (DT9053 to DT9057/DT90053 to DT90057).

Note

The values stored for the clock/calendar can be overwritten (to calibrate the date and time). Values should be written to DT9054 to DT9057/DT90054 to DT90057 either using the F0 (MV) instruction or directly, using programming tools.

High−performance counter

These registers are used for reading and writing the target value and elapsed value of the high−performance counters.

- High−performance counter elapsed/target value area (DT9044 to DT9051/DT90044 to DT90051 and DT9104 to DT9111/DT90104 to DT90111)

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, all special data registers are cleared to 0. If self−diagnosis error 44 or an error with a lower number occurs, however, DT9000 (DT90000 with the FP2, FP2SH and FP10SH) is not cleared.

1.3.3 File Registers (FL)

Function of file registers (FL)

File registers are memory areas which are handled in word (16−bit) units, and are used to store data such as numerical data configured of 16 bits.
They can be used in exactly the same way as data registers.

Bit position	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0
FLn	0	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0

Double−word specifications can also be used in the same way as with data registers. 32−bit data can be handled.

The number of file registers varies depending on the type and the system register settings.

Type	No. of file register words
FP10SH	32,765 words
FP2 (32 K)	Max. 30,717 words (see note)
FP2 (16 K)	Max. 14,333 words (see note)
FP2SH	32,765 words ×3 banks

Note

The number of words varies depending on the type and the system register settings.

Non−hold type data and hold type data

System register 9 can be used to specify whether hold types or non−hold types of file registers are to be used. For the default setting, all file registers are hold types.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, all file registers are cleared to 0. Even if a hold type has been specified, these are cleared to 0.

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the file registers are not cleared to 0 even if the Initialize/Test switch is set to the upper side.

1.3.4 WX, WY, WR and WL

Function of WX, WY, WR and WL

Relays (X, Y, R, L) can be handled as blocks of 16 points.

These are one−word (16−bit) memory areas, thus they can be treated as data memory.

The composition of the one−word memory areas is as follows.
The numbers correspond to the words as shown.

WR0																
	RF	RE	RD	RC	RB	RA	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
WR1																
	R1F	R1E	.	.	.	.	.	.	.	.	.	.	.	R12	R11	R10
WR2																
	R2F	R2E	.	.	.	.	.	.	.	.	.	.	.	R22	R21	R20

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, WX, WY, WR, and WL are cleared to 0. Even if a hold type has been specified, these are cleared to 0.

Pulse relays (P) and error alarm relays (E) cannot be handled in word units.

Examples of using WX, WY, WR and WL

WX can be used to read in digital switch and keyboard inputs, and WY can be used for output to 7−segment displays.

WR can also be used as a shift register.

All of the relays can be used to monitor 16−bit words.

Precautions concerning usage

If an on or off status of one of the relays composing the memory area changes, the memory area value will also change.

WR0																
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	(H9)

When R1 is turned on

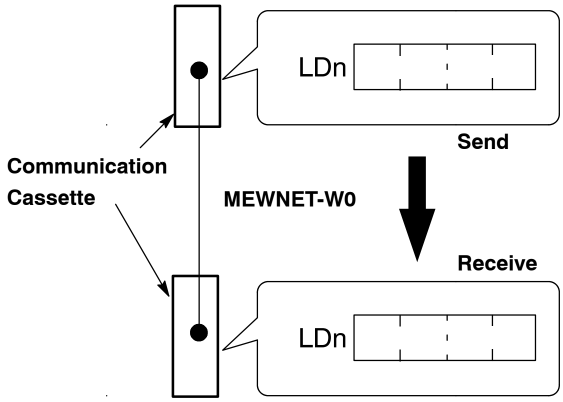
WR0																
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	(HB)

1.3.5 Link Data Registers (LD) for FPΣ/FP−X/FP0R

Function of link data registers (LD)

Link data registers are data memories for “PC links”, which are shared between multiple programmable controllers which are connected through the same network link.

When data is written to a link data register of one PLC, the contents are stored in the link data registers that have the same numbers, in other PLCs connected through the network.



When link data registers are used, data can be exchanged between PLCs simply by writing the data, as shown here.

Available range of link data registers

The available range of link data registers varies depending on the type of network and the combination of units. The available range and number of points must be specified separately for each network

For MEWNET−W0

A maximum of 128 words can be used with one control unit. The available range is from LD0 to LD127

Specifying hold type and non−hold type registers

There are two types of link data registers, which can be switched when the power is turned off and the mode is switched from RUN to PROG and operation is stopped.

- Hold type registers, which hold the on or off status in effect immediately prior to stopping, during the period between stopping and resuming operation
- Non−hold type registers, which are reset when operation stops

In case of using back−up battery, System registers 12 can be used to specify whether the link data registers are the hold type or non-hold type.

Range	System register no.
LD0 to LD127	12

If the beginning of a hold type register is specified using a word number, registers before that point will be non−hold types, and subsequent registers will be hold types. For example, if “64” is set for system register 12, LD0 to LD63 will be non−hold types, and LD64 to LD127 will be hold types.

For the default value, all link data registers are hold types.

If used as link data registers for reception, be aware that no holding operation is carried out, even if the link data registers are specified as hold types using the system registers.

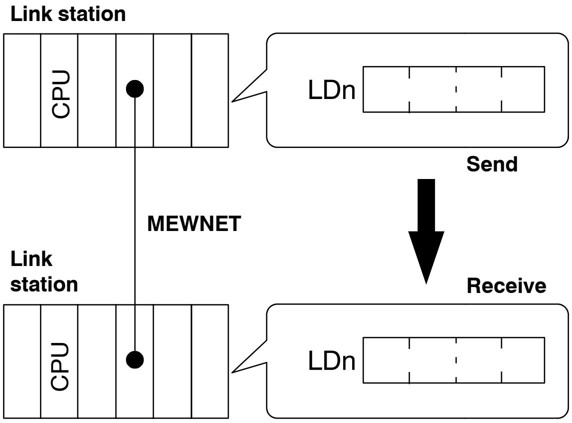
1.3.6 Link Data Registers (LD) for FP2/FP2SH/FP10SH

Function of link data registers (LD)

Link data registers are data memories for “PC links”, which are shared between multiple programmable controllers which are connected through the same MEWNET link.
The following types of MEWNET links are available.

- MEWNET-H link system for FP10SH (for coaxial cables)
- MEWNET-W link system for FP2, FP2SH and FP10SH (for wire cables)
- MEWNET-P link system for FP10SH (for fiber–optic cables)

When data is written to a link data register of one PLC, the contents are stored in the link data registers that have the same numbers, in other PLCs connected through the MEWNET.



When link data registers are used, data can be exchanged between PLCs simply by writing the data, as shown here.

Available range of link data registers

The available range of link data registers varies depending on the type of network and the combination of units. The available range and number of points must be specified separately for each network.

For MEWNET−W and MEWNET−P:

A maximum of 128 words can be used with one link unit. The available range is from LD0 to LD127 for the first unit (PC Link 0), and from LD128 to LD255 for the second unit (PC Link 1).

For MEWNET−W2:
A maximum of 4,096 words can be used per link unit. Please set the range of use at the MEWNET−W2 settings menu.

With the FP2SH, the range between LD0 and LD8447 can be specified. When used with MEWNET−W the range between LD0 and LD255 cannot be used.

With the FP2, the range between LD0 and LD255 can be specified. Also, the data register can be used in place of the link relay by setting the MEWNET−W2 setting menu. However, when used with MEWNET−W the range between LD0 and LD255 cannot be used with MEWNET−W2.

For MEWNET−H:
A maximum of 8,192 words can be used. Please set the range to be used with the MEWNET−H link setting software.

With the FP10SH, the range from LD0 to LD8447 can be used.
If used in conjunction with a MEWNET−W or MEWNET−P link unit, be aware that the range from LD0 to LD255 cannot be used.

Specifying hold type and non-hold type registers

There are two types of link data registers, which can be switched when the power is turned off and the mode is switched from RUN to PROG and operation is stopped.

- Hold type registers, which hold the on or off status in effect immediately prior to stopping, during the period between stopping and resuming operation
- Non-hold type registers, which are reset when operation stops

System registers 12, 13 and 17 can be used to specify whether the link data registers are the hold type or non-hold type.

Range	System register no.
LD0 to LD127	12
LD128 to LD255	13
LD256 to LD8447	17

If the beginning of a hold type register is specified using a word number, registers before that point will be non-hold types, and subsequent registers will be hold types. For example, if “64” is set for system register 12, LD0 to LD63 will be non-hold types, and LD64 to LD127 will be hold types.

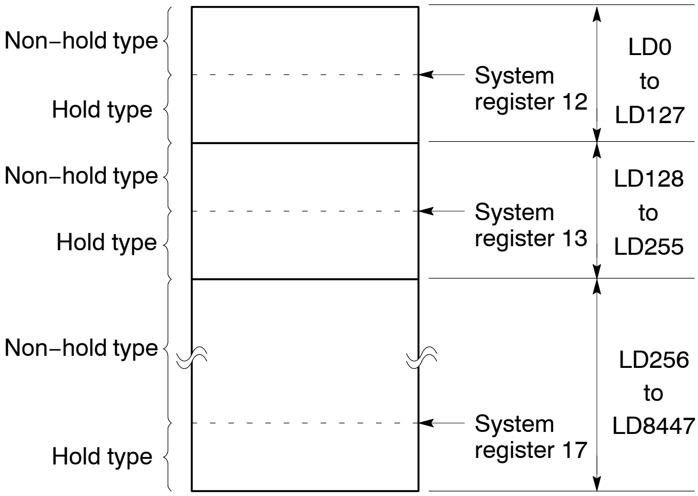
For the default value, all link data registers are hold types.

If used as link data registers for reception, be aware that no holding operation is carried out, even if the link data registers are specified as hold types using the system registers.

Note

This is determined based on the settings of system register 0 and 1.

Example:



Note

Link data registers must be allocated when the network is configured, before programming is done. The method by which allocations are made varies depending on the type of network. Refer to the manual for the pertinent link unit.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, all of the link data registers (LD) are cleared to 0. Even if a hold type has been specified, these link data registers are cleared to 0.

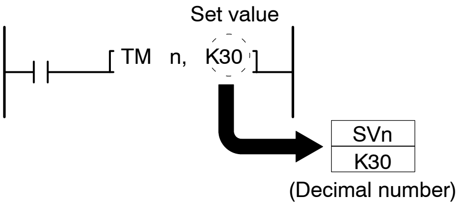
Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the link data registers are not cleared to 0 even if the Initialize/Test switch is set to the upper side.

1.3.7 Set Value Area for Timer/Counter (SV)

Function of set value areas (SV)

A set value for a timer or counter is stored in the set value area (SV) with the same number as the timer or counter.



A decimal number or SV area number is specified for the set value when the **TM** or **CT** instruction is entered in the program.

An SV is a one–word, 16–bit memory area which stores a decimal number from K0 to K32767.

Using set value area (SV)

During RUN mode, a set value for a timer or counter can be changed by rewriting the corresponding set value area.

The value in a set value area can be read and changed from the program by specifying the destination and other information in **F0 (MV)** data transfer instruction.

The set value area can be read and rewritten using a programming tool.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, all timer/counter setting value areas (SV) are cleared to 0. Even if a hold type has been specified, these are cleared to 0.

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that these areas not cleared even if the Initialize/Test switch is set to the upper side.

Example:

SV and EV areas are in a one–to–one correspondence with timers and counters.

Timer/Counter number	Set value area (SV)	Elapsed value area (EV)
T0 T1 : T99	SV0 SV1 : SV99	EV0 EV1 : EV99
C100 :	SV100 :	EV100 :

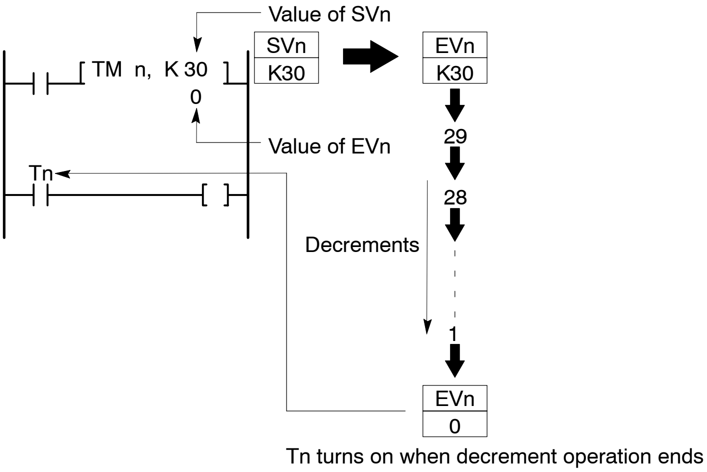
1.3.8 Elapsed Value Area for Timer/Counter (EV)

Function of elapsed value areas (EV)

While a timer or counter is operating, the elapsed value is stored in the elapsed value area (EV) with the same number as the timer or counter.

When the EV reaches zero, the timer or counter contact with the same number turns on.

An EV is a one–word, 16–bit memory area which stores a decimal number from K0 to K32767.



Using elapsed value area (EV)

The elapsed value of a timer or counter in operation can be changed to prolong or shorten the operation.

The value in elapsed value area can be read and changed from the program by specifying the **F0 (MV)** data transfer instruction.

The elapsed value area can be read and rewritten using a programming tool.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, the timer/counter elapsed value areas (EV) are cleared to 0. Even if a hold type has been specified, these values are cleared to 0.

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that the timer/counter elapsed value areas are not cleared to 0 even if the Initialize/Test switch is set to the upper side.

1.3.9 Index Registers (IX, IY) (for FP0, FP−e)

Function of index registers (IX, IY)

Index registers are used to indirectly specify constants and memory area addresses. Two 16−bit registers are available, IX and IY. Changing addresses and constants using a value in an index register is called “index modification”.

With the FP0, FP−e, index modification is possible only with regard to operands of high−level instructions.

Modifying an address

Address = Base address + Value in IX or IY (K constant)

Example: Modifying DT11

IXDT11

Base address		IX value		Target address
11	+	K0	=	DT11
11	+	K10	=	DT21
11	+	K−10	=	DT1

Modifying a constant

Constant = Base value + Value in IX or IY

Example 1: Modifying K100

IXK100

Base value		IX value		Constant
K100	+	K0	=	K100
K100	+	K10	=	K110
K100	+	K−10	=	K90

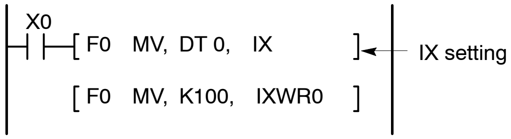
Example 2: Modifying H10

IXH10

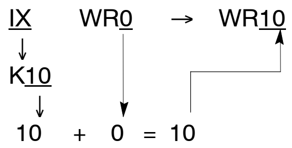
Base value		IX value		Constant
H10	+	HA	=	H1A
H10	+	H10	=	H20

Index modification method

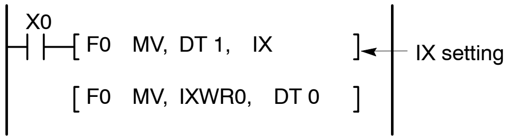
Example 1: Modifying a destination address



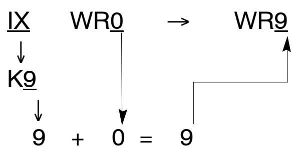
The value of DT0 determines the WR address where K100 is written.
When the DT0 value is K10, K100 is written to WR10.



Example 2: Modifying a source address



The value of DT1 determines the WR address for transferring a value to DT0.
When the DT1 value is K9, the value in WR9 is transferred to DT0.



Cautions when using index registers

An index register can not be modified with an index register.
IXIX, IXIY

If the result of address modification overflows the memory area, an operation error will result.

When the address resulting from modification is negative or a large number.

When modifying 32−bit constants, IX is specified. At this point, IX and IY in combination are handled as 32−bit data.

Higher 16−bit area	Lower 16−bit area
Contents of IY	Contents of IX

The results of modification will be 32−bit data.

Note

For detailed information about the procedures for using index registers ☛ section 4.5

1.3.10 Index Registers (I0 to ID) (for FPΣ/FP−X/FP0R)

Function of index registers (I0 to ID)

Index registers are used for indirect specification of values to addresses and operands in relays and memory areas.

There are a total of 14 index registers which can be used with the FPΣ, consisting of I0 to I9 and IA to ID.

Cautions when using index registers

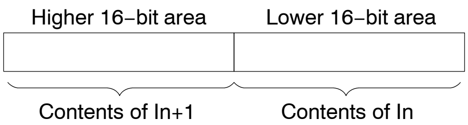
An index register can not be modified with an index register.
I0I0, I1I1

An index register can be modified using a different index register.
Available: I0IA, Not available: I0I0

If the result of address modification overflows the memory area, an operation error will result.

When the address resulting from modification is negative or a large number.

When a 32−bit constant is modified, the specified index register number and the following index register number are used in combination to handle the data as a 32−bit data.



The results of modification will be 32−bit data.

Note

When 32−bit constants are being modified, ID should not be specified.

The following index modifications are possible

Memory area numbers used with high-level instructions

K constants (16-bit and 32-bit) and H constants (16-bit and 32-bit) specified with high-level instructions

Note

There are some cases in which index modification cannot be specified, depending on the instruction. Confirm the table of "Operands" on the page describing the various instructions.

1.3.11 Index Registers (I0 to ID) (for FP2, FP2SH and FP10SH)

Function of index registers (I0 to ID)

Index registers are used for indirect specification of values to addresses and operands in relays and memory areas.

Changing an address or a constant using an index register value is called “index modification”.

There are a total of 14 index registers which can be used with the FP2, FP2SH and FP10SH, consisting of I0 to I9 and IA to ID.

With the FP2SH/FP10SH, because there are bank areas for index registers, changing the bank enables 14 points x 16 banks = 224 points of index registers available for use.

Cautions when using index registers

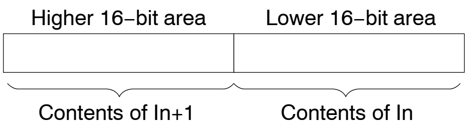
An index register can not be modified with an index register.
I0I0, I1I1

An index register can be modified using a different index register.
Available: I0IA, Not available: I0I0

If the result of address modification overflows the memory area, an operation error will result.

When the address resulting from modification is negative or a large number.

When a 32–bit constant is modified, the specified index register number and the following index register number are used in combination to handle the data as a 32–bit data.



The results of modification will be 32–bit data.

Note

When 32–bit constants are being modified, ID should not be specified.

For the FP2/FP2SH/FP10SH, if the Initialize/Test switch is set to the upper side (the Initialize side) in the PROG mode, index registers I0 to ID are cleared to 0.

Note

With the FP2SH/FP10SH, system register 4 can be set in such a way that these are not cleared to 0 even if the Initialize/Test switch is set to the upper side.

The bank switching function for index registers can be used on the FP2SH/FP10SH. This function is not provided in the FP2.

The following index modifications are possible

Memory area numbers used with high-level instructions

K constants (16-bit and 32-bit) and H constants (16-bit and 32-bit) specified with high-level instructions

Relay numbers used with the following basic instructions: **ST**, **ST**/, **AN**, **AN**/, **OR**, **OR**/, **OT**, **KP**, **SET**, **RST**, **OT**↑, **OT**↓

Instruction numbers specified with the following basic instructions: **TM**, **CT**, **MC**, **MCE**, **JP**, **LOOP**, **CALL**, **FCAL** (**FCAL** instruction can be used with the FP2SH/FP10SH.)

Memory areas used with the following basic instructions: **TM**, **CT**, **SR**

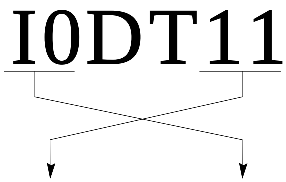
Note

There are some cases in which index modification cannot be specified, depending on the instruction. Confirm the table of "Operands" on the page describing the various instructions.

Modification of memory area numbers specified by high-level instructions

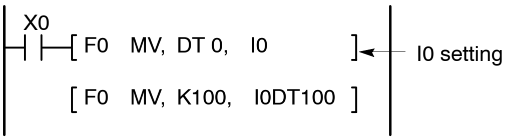
Address = Base address + value in I0 through ID (K constant)

Example: Modifying DT11

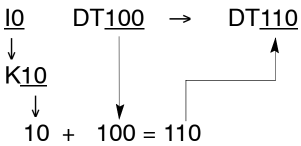


Base address		I0 value		Target address
11	+	K0	=	DT11
11	+	K10	=	DT21
11	+	K−10	=	DT1

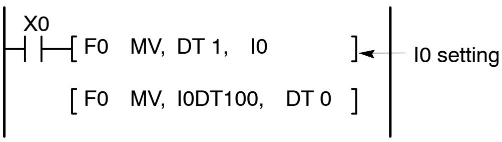
Example 1: Modifying a destination address



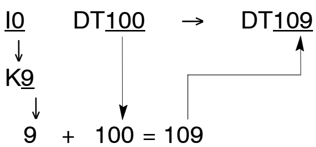
The value of DT0 determines the DT address where K100 is written.
When the DT0 value is K10, K100 is written to DT110.



Example 2: Modifying a source address



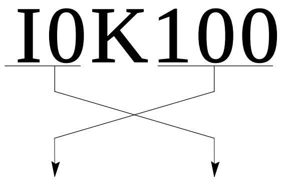
The value of DT1 determines the DT address for transferring a value to DT0.
When the DT1 value is K9, the value in DT109 is transferred to DT0.



Modification of values of constants specified by high−level instructions

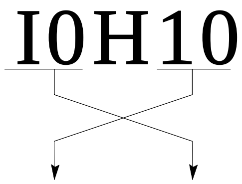
Constant = Base value + value in I0 through ID

Example 1: Modifying 16-bit constant K100



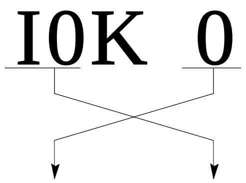
Base value		I0 value		16-bit constant
K100	+	K0	=	K100
K100	+	K10	=	K110
K100	+	K−10	=	K90

Example 2: Modifying 16-bit constant H10



Base value		I0 value		16-bit constant
H10	+	HA	=	H1A
H10	+	H10	=	H20

Example 3: Modifying 32-bit constant K0

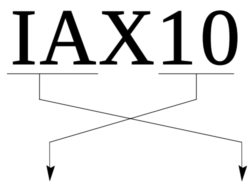


Base value		I1 and I0 value		32-bit constant
K0	+	K10000	=	K10000
K0	+	K60000	=	K60000
K0	+	K999999	=	K999999

Modification of relay numbers specified by basic instructions

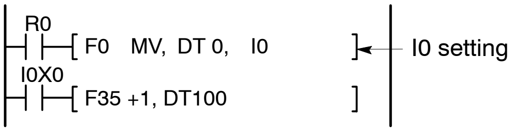
Number = Base number + value in I0 through ID (K constant / H constant)

Example: Modifying X10

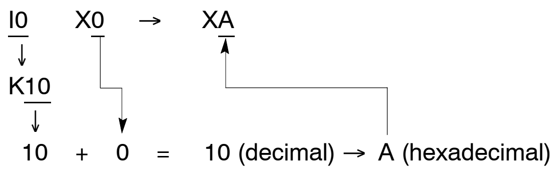


Base number		IA value		Target number
10	+	H0	=	X10
10	+	HF	=	X1F
10	+	H−10	=	X0
19	+	K7	=	X20
19	+	K−11	=	XE

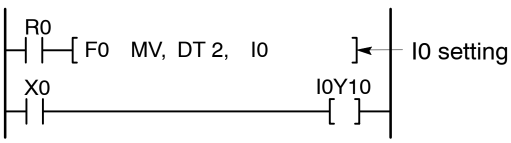
Example 1: Modifying a trigger



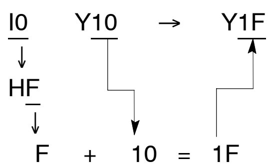
The trigger of the F35 (+1) instruction is determined by the DT0 value.
When the value of DT0 is K10, the F35 (+1) instruction is executed when XA goes on.



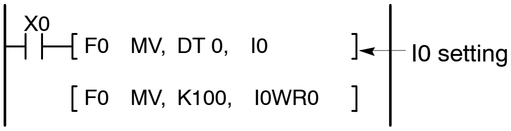
Example 2: Modifying an output destination



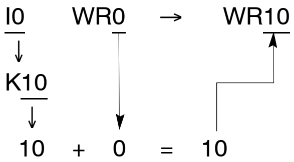
The value of DT2 determines the output destination when X0 goes on.
When the value of DT0 is HF and X0 goes on, Y1F goes on.



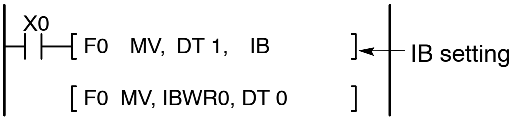
Example 3: Modifying a destination address



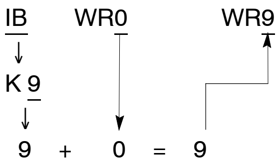
The value of DT0 determines the address of WR where K100 is written.
When the value of DT0 is K10, K100 is written to WR10.



Example 4: Modifying a source address



The value of DT1 determines the address of WR for transferring a value to DT0.
When the value of DT1 is K9, the value in WR9 is transferred to DT0.



Items requiring particular attention

For the external input relay (X), external output relay (Y), and internal relay (R), when using index modification on relay numbers, be aware that the last digit of the relay number is hexadecimal and the first digits are decimal.

Example: **For external input relay (X)**

X □□ □

Decimal
1, 2, 3 12

Hexadecimal
0, 1, 2, 3 .... 9 A,B .... F

X 0, X 1 . . . . X F
X 10, X 11 . . . . X 1F
X 20, X 21 . . . . X 2F
⋮ ⋮

Example using I0X0

Value of I0		Target address
K	H	
0	0	X0
1	1	X1
:	:	:
9	9	X9
10	A	XA
:	:	:
15	F	XF
16	10	X10
:	:	:
31	1F	X1F
:	:	:
159	9F	X9F
160	A0	X100
161	A1	X101
:	:	:
255	FF	X15F
256	100	X160
257	101	X161
:	:	:
265	10A	X169
267	10B	X16A
:	:	:

Modifying instruction numbers of basic instructions

Timer numbers

Modifying TML20 −−− TML I020

Counter numbers

Modifying CT3000 −−− CT I03000

Shift register numbers

Modifying SRWR0 −−− SR I0WR0

Master control numbers

Modifying MCE1 −−− MCE I01

Label number specification with the Jump instruction

Modifying JP1 −−− JP I01

Label number specification with the Loop instruction

Modifying LOOP5 −−− LOOP I05

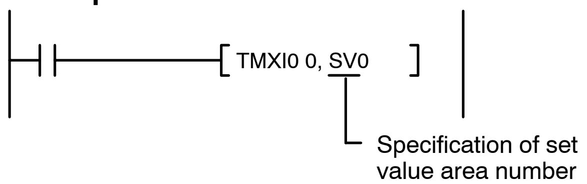
Subroutine program numbers

Modifying CALL10 −−− CALL I010

Note

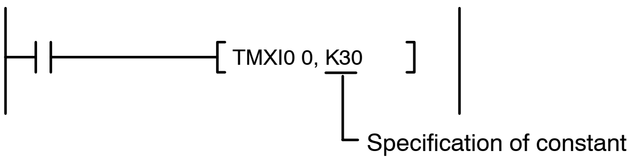
Timer numbers and counter numbers can be modified only when a memory area is specified for the set value.

Correct



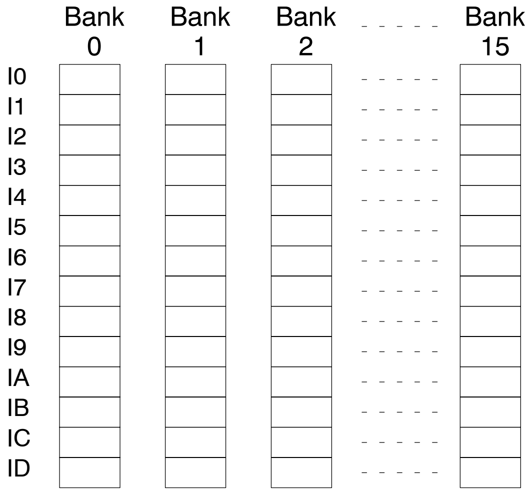
Modification cannot be done if the set value is specified with a constant.

Incorrect



Changing index register banks (for FP2SH/FP10SH only)

The banks of the index registers of the FP2SH/FP10SH can be changed to allow use of up to 224 points (14 points × 16 banks) in a program.

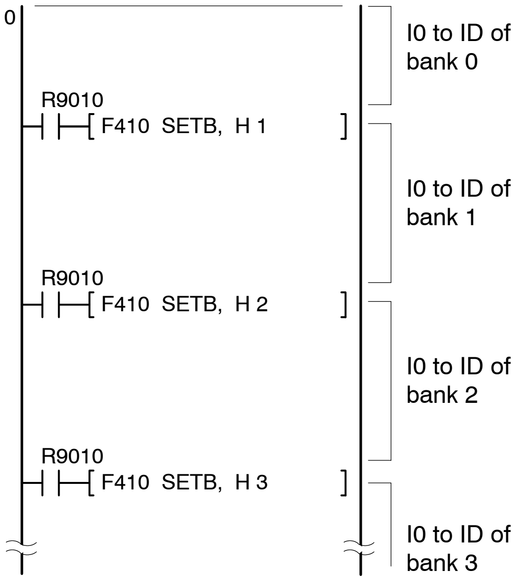


When the register bank setting instruction **F410 (SETB)** or the register bank changing instruction **F411 (CHGB)** is used to specify a bank number, index registers I0 to ID used after that point can be used as separate index registers from the I0 to ID index registers used prior to changing the bank.

The bank is automatically set to bank 0 before execution of the leading address of the program. The bank is also automatically set to bank 0 before execution of the leading address of a second program.

The bank numbers of index registers used in interrupt programs, subroutines, and other sub programs should be specified in such a way that the **F411 (CHGB)** instruction is executed at the beginning of the sub program, and the **F412 (POPB)** instruction is executed at the end of the sub program.

Example 1: Changing banks using a register bank setting instruction F410 (SETB)

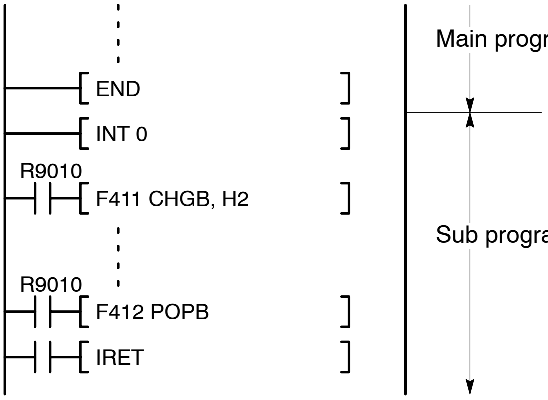


Different values can be set for I0 in bank 0, bank 1 and bank 2. The set values are only effective within their respective ranges.

Note

For details on changing bank instruction, refer to the explanations of F410 (SETB), F411 (CHGB) and F412 (POPB) instructions.

Example 2: Changing banks within an interrupt program



1.4 Explanation of Constants

1.4.1 Integer Type Decimal Constants (K)

Function of decimal constants (K)

This is binary data that has been converted to the decimal format.

When entering and reading a decimal constant, specify the value by entering a K at the beginning.

Decimal constants are primarily used to specify data sizes and quantities such as set values for timer.

In the PLC, the decimal constant (K) is processed as binary (BIN) data in units of 16 bits, as shown below.

The sign is determined by the MSB “Most Significant Bit” (bit position 15). [A “0” indicates a positive sign (+), and a “1” indicates a negative sign (–).] The MSB (Most Significant Bit) is called the “sign bit”.

Example: Decimal number “+32” (K32)

Bit position	15	.	.	12	11	.	.	8	7	.	.	4	3	.	.	0
Binary data	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

“+”

Example: Decimal number “–32” (K–32)

Bit position	15	.	.	12	11	.	.	8	7	.	.	4	3	.	.	0
Binary data	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

“–”

Data is normally handled in units of one word (16 bits), however, it is also occasionally handled in units of two words (32 bits). In this case, as well, the MSB serves as the sign bit.

The available range of a decimal constant is:

16-bit equivalent data: K-32768 to K32767

32-bit equivalent data: K-2147483648 to K2147483647

1.4.2 Hexadecimal Constants (H)

Function of hexadecimal constants (H)

Hexadecimal constants are values which have been converted from binary into hexadecimal. When entering and reading a hexadecimal constant, specify the value by entering an H at the beginning.

Hexadecimal constants are primarily used to specify an ordering of 1's and 0's in 16-bit data, such as system register settings and specification of control data for high-level instructions. Hexadecimal constants are also used to specify BCD data.

In the PLC, the hexadecimal constant (H) is processed as binary (BIN) data in units of 16 bits, as shown below.

Example: Hexadecimal number "2A" (H2A)

Bit position	15 . . 12				11 . . 8				7 . . 4				3 . . 0			
Hexadecimal	0				0				2				A			
Binary data	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0

Data is normally handled in units of one word (16 bits), however, it is also occasionally handled in units of two words (32 bits).

The available range of a hexadecimal constant is:

16-bit equivalent data: H0 to HFFFF

32-bit equivalent data: H0 to HFFFFFFFF

1.4.3 Floating Point Type Real Numbers (f)

Available PLC

FP0, FP0R, FP−e, FPΣ, FP−X, FP2, FP2SH and FP10SH

Range of floating point type real numbers that can be used in operations

The range of floating point type real numbers that can be stored in the memory area is as noted below.
Range of negative numbers: -3.402823×10^{38} to $-1.175494 \times 10^{-38}$
Range of positive numbers: 1.175494×10^{-38} to 3.402823×10^{38}

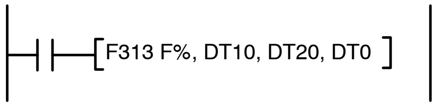
Even if the results of the real−number operation involve multiple digits, the actual processing is effective for a mantissa of up to 7 digits.

Example: If the actual operation result were 0.33333333 ..., the stored data would consist of the value 0.3333333.

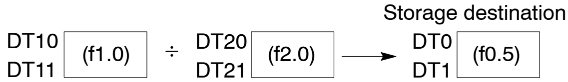
Area in which floating point type real numbers are stored

With floating point type real number operation instructions, the area in which data converted to a real number is stored consists of two words (32 bits) per data element. As a result, in transmission instructions such as that used to send real−number data to a storage area and in other operations, data should be moved in units of two words (32 bits).

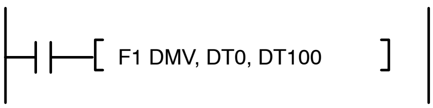
Example 1: If DT0 is specified as the area in which floating point type real number data is to be stored, the data will be written to DT0 and DT1.

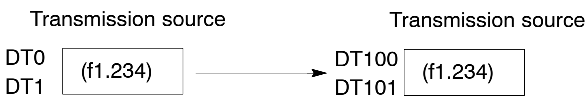


The operation results will be stored in DT0 and DT1.



Example 2: When floating point type real number data stored in DT0 and DT1 is being sent to destination, the 32−bit data sending instruction F1 (DMV) instruction should be used.





Processing of floating point type real number operations

1) Processing by specifying an integer device

Instructions can be used to store data in a specific location. Adding the symbol % or # to either S (source: the area from which the data is loaded) or D (destination: the area in which the result is stored) determines how the data is processed. If added to S (source), integer data is automatically converted to real–number data and the operation is carried out. If added to D (destination), the real–number data resulting from the operation is automatically converted into integer data and stored in the destination.

When the integer area consists of 16–bit data ... It is specified using the % symbol.
When the integer area consists of 32–bit data ... It is specified using the # symbol.

Example 1: Specifying the target operation data S for an integer device

The contents of “DT10” and “DT20” are converted to real numbers, and the operation is executed. The results are stored in “DT30 and DT31” as real–number data.

```
|—| |—[ F310 F+, %DT10, %DT20, DT30 ]|
```

Example 2: Specifying stored results D for an integer device

The target operation data stored in “DT40 and DT41” and “DT50 and DT51” are loaded, and the operation is executed. The results of the operation are converted to an integer and stored in DT60.

```
|—| |—[ F310 F+, DT40, DT50, %DT60 ]|
```

Example 3: When the integer data S targeted by the operation is stored as two words

The contents of “DT70 and DT71” and “DT80 and DT81” are converted to real numbers and the operation is executed. The results of the operation are stored in “DT90 and DT91” as real–number data.

```
|—| |—[ F310 F+, #DT70, #DT80, DT90 ]|
```

In processing involving an integer device specification and real numbers being converted to integers, the processing is the same as that of the **F327 (INT)** instruction.

If the real–number data is a positive number, the number is rounded off, and any digits to the right of the decimal point are discarded.

If the real–number data is a negative number, the value 0.4999 ... is subtracted from the target real–number data, and the value is rounded off to the decimal point.

Example 1: If the operation result is f1.234, the value will be stored as integer data “K1”.

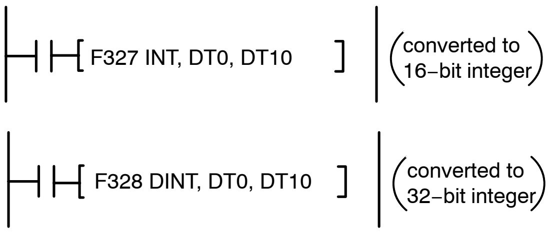
Example 2: If the operation result is f–1.234, the value will be stored as integer data “K–2”.

Integer device specification can be used for the following instructions.
F309 (FMV) to **F324 (FSQR)** / **F336 (FABS)** to **F338 (DEG)** / **F345 (FCMP)** to **F349 (FZONE)**

2) Using the integer → real number and real number → integer conversion instructions to convert values

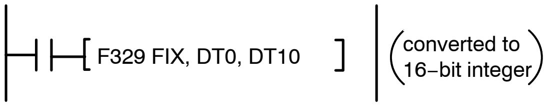
With this method, a conversion instruction is used to convert integer data to real numbers.
When the integer data is 16–bit data, **F325 (FLT)** is used.
When the integer data is 32–bit data, **F326 (DFLT)** is used.
Real–number data that has undergone real–number operation processing is converted from real–number data to integer data using the **F327 (INT)** to **F332 (DROFF)** conversion instructions.

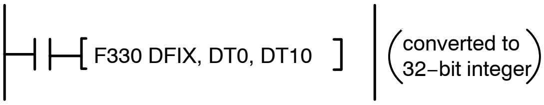
Example 1: When conversion is carried out using the maximum value that does not exceed the allowable range



When the value is a positive number, the result is rounded off to the decimal point.
When the value is a negative number, the value 0.4999 ... is subtracted from the data, and the result is rounded off.
If the real–number data is 1.5, it is converted as integer data K1.
If the real–number data is –1.5, it is converted as integer data K–2.

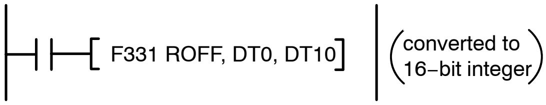
Example 2: When conversion is carried out by rounding down the digits to the right of the decimal point

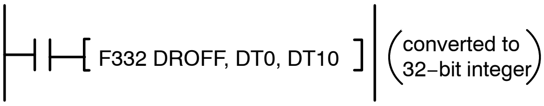




Digits to the right of the decimal point are rounded down.
If the real–number data is 1.5, it is converted as integer data K1.
If the real–number data is –1.5, it is converted as integer data K–1.

Example 3: When conversion is carried out by rounding off the digits to the right of the decimal point.





Digits to the right of the decimal point are rounded off.
If the real–number data is 1.5, it is converted as integer data K2.
If the real–number data is –1.5, it is converted as integer data K–2.

3) Direct specification of the real−number constant data

When operations are being carried out on real−number constants as real−number data, the values can be directly input by using a programming tool in which “f” is added either to the target data “S” or the destination “D” defined by the instruction.
The range that can be specified by these instructions is 0.0000001 to 9999999 (the effective value consists of seven digits).

Example: **Specifying the target data “S” with a real−number constant**

The real−number data stored in DT10 and DT11 is multiplied by the real−number constant 0.5, and the result of the operation stored in DT20 and DT21 as real−number data.

```
|—| |—[ F312 f*, DT10, f0.5, DT20 ]|
```

4) Specifying a K constant for conversion

The K constant (32−bit data) is an integer data element, so it is automatically converted to real−number data and the operation is executed.

```
|—| |—[ F310 F+, K10, DT50, DT60 ]|
```

Automatic conversion to real number

5) Specifying an H constant for conversion

With an H constant (32−bit data), the operation is carried out using the H constant as floating point data.

Operation if an overflow occurs

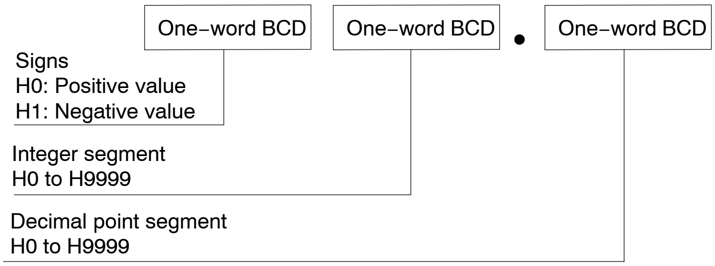
If the operation result exceeds the real−number range, an overflow flag (R9009) is set.
If this occurs, one of the values noted below is set for R9009 as a result.
Positive infinite value: H7F800000
Negative infinite value: HFF800000

1.4.4 BCD Type Real Numbers (H) (for FP2, FP2SH and FP10SH)

Range of BCD type real numbers that can be used in operations

The range of real–number data that can be stored in the memory area is as noted below.
–9999.9999 to +9999.9999

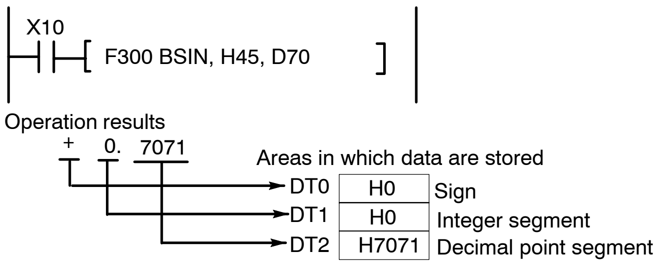
Data stored in the memory area in one–word units, with the positive/negative sign coming first, followed by the integer segment and then by the decimal point and any subsequent digits.



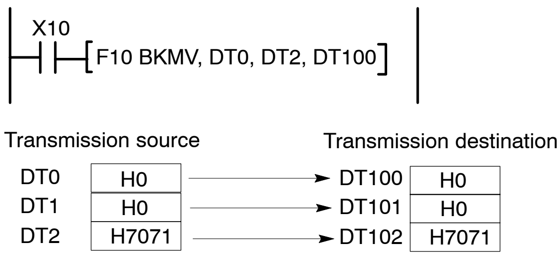
Area in which the BCD type real number is stored

In the BCD type real number operation instructions, the area in which data converted to real numbers is stored consists of a three–word area for each data element. As a result, in instructions such as that used to send real–number data to a storage area and in other operations, data should be moved in units of three words.

Example 1: If DT0 is specified as the area in which BCD type real–number data is to be stored, the data will be written to "DT0 to DT2".



Example 2: When sending BCD type real–number data stored in "DT0 to DT2", the F10 (BKMV) block transmission instruction or a similar instruction should be used, and the data sent in three–word units.



1.4.5 Character Constants (M)

Function of character constants (M)

The character constant is used to express ASCII code in binary.

The character constant is expressed by adding the prefix M to the data.

There are only two instructions in which character constants can be specified, **F95 (ASC)** instruction, **F257** to **F265 (SYS1)** instruction and **F149 (MSG)** instruction.

The character constant M is stored in a specified memory area in the PLC as BIN data, as shown below.

Example: When character constant "MEWNET" is input

T	E	N	W	E	M	Character constant
54	45	4E	57	45	4D	ASCII Hex code
One word		One word		One word		

1.5 Data Ranges Which can be Handled in the PLC

1.5.1 Data Ranges Which can be Handled in the PLC

16–bit data

Data which can be handled in the PLC (16–bit binary data)	Decimal constants	Hexadecimal constants
0111111111111111	K 32767	H7FFF
. . .	. . .	. . .
0000000000000001	K 1	H0001
0000000000000000	K 0	H0000
1111111111111111	K –1	HFFFF
. . .	. . .	. . .
1000000000000000	K –32768	H8000

32-bit data

Data which can be handled in the PLC (32–bit binary data)	Decimal constants	Hexadecimal constants
01111111111111111111111111111111	K 2147483647	H7FFFFFFF
. . .	. . .	. . .
00000000000000000000000000000001	K 1	H00000001
00000000000000000000000000000000	K 0	H00000000
11111111111111111111111111111111	K –1	HFFFFFFFF
. . .	. . .	. . .
10000000000000000000000000000000	K–2147483648	H80000000

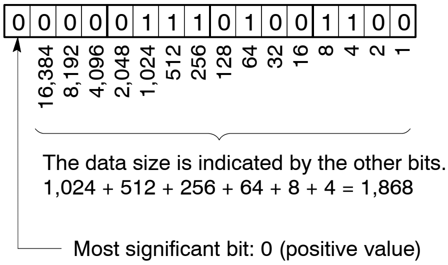
Expression of decimal numbers in PLC

Decimal number is basically processed in 16-bit or 32-bit binary.

The most significant bit (MSB) expresses negative or positive sign of the data. When the MSB is “0”, data is regarded as having a zero or positive value and when the MSB is “1”, data is regarded as having a negative value.

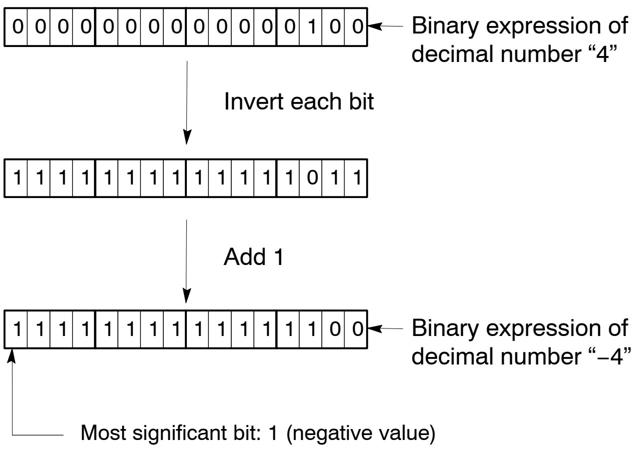
In the case of positive numbers, the bits following the most significant bit express the size of the data.

Example 1: Expressing the decimal number “1868”



A negative number is expressed as a two’s complement (the bits of the 16−bit binary data of the positive number are inverted and 1 is added to the result).

Example 2: Expressing the decimal number “-4”



Data ranges which can be handled in the PLC

Binary data which can be handled by programmable controllers are:
16-bit binary data: K-32768 to K32767
32-bit binary data: K-2147483648 to K2147483647

BCD code which can be handled by programmable controllers are:
16-bit (4-digit BCD H code): H0 to H9999
32-bit (8-digit BCD H code): H0 to H99999999

If any of the above ranges are exceeded when processing the corresponding data, overflow or underflow will result.

BCD is an acronym for binary coded decimal and refers to expressing each digit of a decimal number by four binary digits.

Example: When the decimal number is expressed in BCD

Decimal number	**6**	**4**	**5**	Each digit is converted to binary
	↓	↓	↓	
BCD code (Binary coded decimal)	0110	0100	0101	→ H645

1.5.2 Overflow and Underflow

Operation instructions occasionally produce a value which is outside of the allowed range. This is called overflow if the value exceeds the maximum value and underflow if the value falls short of the minimum value. When an overflow or underflow occurs, the carry flag R9009 turns on.

Overflow and underflow during binary operation

If any of the following values are exceeded, overflow or underflow will result.

16−bit binary operation
(Overflow results if over the maximum value.)

Max. value	K 32767	H 7FFF
	⋮	⋮
	K 1	H 0001
	K 0	H 0000
	K −1	H FFFF
	⋮	⋮
Min. value	K-32768	H 8000

(Underflow results if under the minimum value.)

32−bit binary operation
(Overflow results if over the maximum value.)

Max. value	K 2147483647	H 7FFFFFFF
	⋮	⋮
	K 1	H 00000001
	K 0	H 00000000
	K −1	H FFFFFFFF
	⋮	⋮
Min. value	K-2147483648	H 80000000

(Underflow results if under the minimum value.)

Overflow and underflow during BCD operation

If any of the following values are exceeded, overflow or underflow will result.
Only positive values can be handled.

4−digit BCD code operation
(Overflow results if over the maximum value.)

Max. value	H 9999
	⋮
Min. value	H 0

(Underflow results if under the minimum value.)

8−digit BCD code operation
(Overflow results if over the maximum value.)

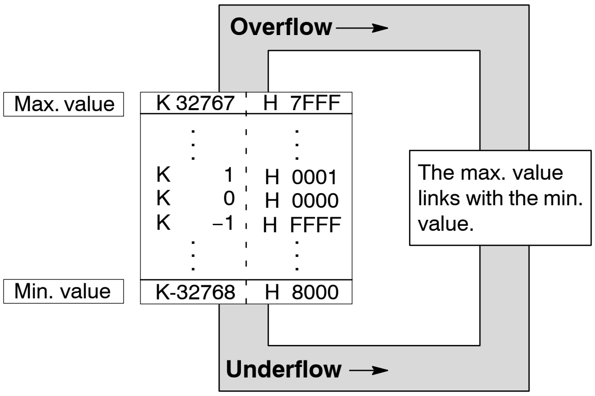
Max. value	H 99999999
	⋮
Min. value	H 0

(Underflow results if under the minimum value.)

Values when overflow or underflow occurs

Numerical value handled by the FP series programmable controller all form a loop joined at the maximum value and the minimum value as shown below.

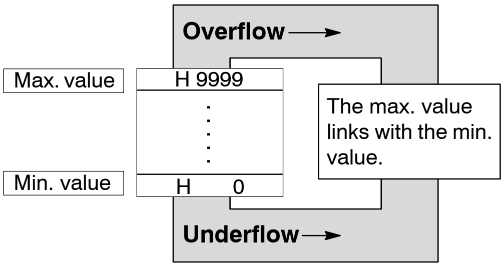
16–bit binary operation



Example 1: For K32767 + K1 (overflow)
The operation result is K–32768 and the carry flag turns on.

Example 2: For K–32768 – K1 (underflow)
The operation result is K32767 and the carry flag turns on.

4–digit BCD code operation



Example 1: For H9999 + H1 (overflow)
The operation result is H0 and the carry flag turns on.

Example 2: For H0 – H1 (underflow)
The operation result is H9999 and the carry flag turns on.

Chapter 2

Basic Instructions

2.1 Composition of Basic Instructions

2.1.1 Sequence Basic Instructions

These basic instructions perform bit unit logic operations and are the basis of the relay sequence circuit.

As shown in the illustration below, this is expressed by the combination of the relay coil and contact.

There are several relay types which are explained in “Section 1.2”, and the relay which can be specified depends on the instruction. Refer to the explanation of each instruction.

Example:

Start (ST) instruction

Read the on or off status of the specified contact.

ST X0

Read the status of external input (X0).

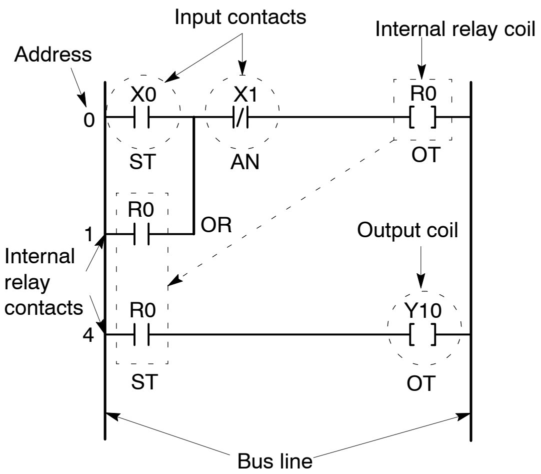
Out (OT) instruction

Output the operation result to the specified coil.

OT Y10

Outputs the operated result (on and off) to the external output (Y10).

Example:



<Ladder diagram>

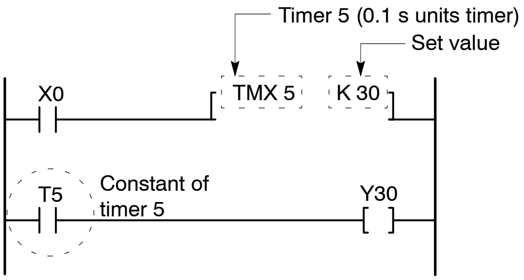
2.1.2 Basic Function Instructions

These are the timer, counter and shift register instructions.

To specify set values, the instructions are composed of several steps.

Example:

Example of setting 3.0 seconds in the 0.1 second timer (timer 5)



Timing begins when X0 turns on, and T5 turns on when 3.0 seconds elapses.

2.1.3 Control Instructions

These instructions determine the order and flow of program execution.

It is possible to change the sections to be executed, or to execute only the necessary segments, depending on the conditions.

Specify the section which will execute. This is composed of several steps.

Master control relay

A certain part of the program (specified with **MC** or **MCE**) is only executed when the appropriate condition is met.

Jump

Skips execution of part of the program (specified with **JP** or **LBL**) when the appropriate condition is met. This shortens program execution time.

Step ladder control

Part of the program (specified with **SSTP** or **STPE**) is treated as an independent “process”, and sequential and branch execution is carried out.

Subroutine program

A program which is repeatedly executed for a particular operation is called as a subroutine (specified with **SUB** or **RET**) and executed when needed.

Interrupt program

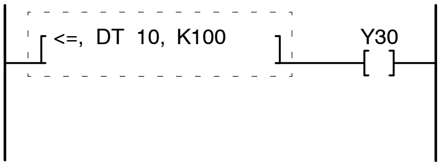
In addition to the normal program, enter an interrupt program (specified with **INT** or **IRET**) if you need a program which will execute immediately when a certain condition is met. When an interrupt is received, the normal program is interrupted and the interrupt program is executed.

2.1.4 Data Compare Instructions

This is a group of instructions which compare two data. A contact is turned on or off based on the result of the comparison. Each comparison instruction is composed of several steps.

Example:

Example of comparing the value of DT10 to K100.



If the value of DT10 is less than K100, Y30 is turned on.
If the value of DT10 is greater than K100, Y30 is turned off.

2.2 Number of Steps in the FP2, FP2SH and FP10SH

Number of steps in basic instructions

Of the basic instructions used with the FP2, FP2SH and FP10SH, the number of steps in the following instructions changes depending on the number specified.

Sequence basic instructions

With **Start (ST)**, **Out (OT)**, **And (AN)**, **Or (OR)**, and **Keep (KP)**, the number of steps making up the instruction changes depending on the relay number which has been specified.

Type of relay		Relay number	Steps	
			Normal	With index modification
Input	**X**	0 to 127F	1	2
Output	**Y**	1280 or more	2	2
Internal relay	**R**	R0 to R111F	1	2
		R1120 or more	2	2
Special internal relay	**R**	R9000 to R910F	2	2
Link relay	**L**	L0 to L127F	1	2
		L1280 to L639F	2	2
Timer	**T**	0 to 255	1	2
Counter	**C**	256 or more	2	2

Note

Index modification is possible only with the FP2, FP2SH and FP10SH.

Basic function instructions

Type of instruction		Specified number	Steps	
			Normal	With index modification
0.001 s units timer	**TML**	0 to 255	3	4
0.01 s units timer	**TMR**	256 or more	4	4
0.1 s units timer	**TMX**			
1 s units timer	**TMY**	0 to 255	4	5
		256 or more	5	5
Counter	**CT**	0 to 255	3	4
		256 or more	4	4
Shift register	**SR**	WR0 to WR239	1	2
		WR240 or more	2	2

Note

Index modification is possible only with the FP2, FP2SH and FP10SH.

Control and subroutine instructions

Instructions	Steps	
	Normal specification	With index modification
JP	2	3
LOOP	4	5
CALL	2	3
FCAL	4	5

Note

Index modification is possible only with the FP2, FP2SH and FP10SH.Table of Basic Instructions

ST
ST/
OT

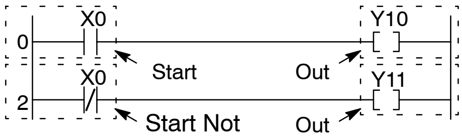
Start

Start Not

Out

Outline **ST, ST/:** Begins a logic operation.
OT: Outputs the operation result.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	0	ST X 0
	1	OT Y 10
	2	ST/ X 0
	3	OT Y 11

Operands

Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L (*1)	P (*2)	E (*3)	T	C	(*4)
ST, ST/	A	A	A	A	A	A	A	A	A
OT	N/A	A	A	A	N/A	N/A	N/A	N/A	

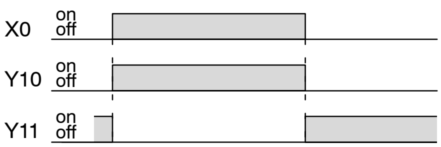
A: Available
N/A: Not Available

(*1) This cannot be used with the FP0/FP−e.
(*2) This can be used only with the FP2/FP2SH/FP10SH.
(*3) This can be used only with the FP2SH/FP10SH.
(*4) This can be used only with the FP0R/FP2/FP2SH/FP10SH.

Explanation of example

Y10 goes on when X0 turns on.

Y11 goes on when X0 turns off.



Description

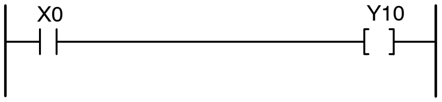
The **ST** instruction starts logic operations and regards the input contact specified at the start as a Form A (normally open) contact.

The **ST**/ instruction starts logic operations and regards the input contact specified at the start as a Form B (normally closed) contact.

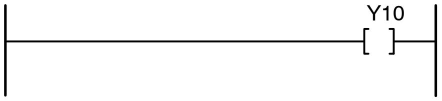
The **OT** instruction outputs the operation result to a specified coil.

Precautions during programming

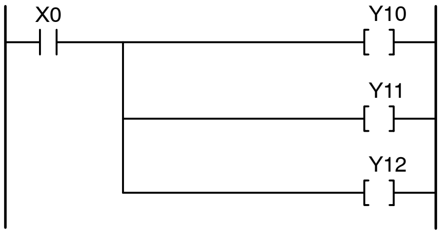
The **ST** and **ST**/ instructions start from the bus line.



The **OT** instruction cannot start directly from the bus line.



The **OT** instruction can be used consecutively.

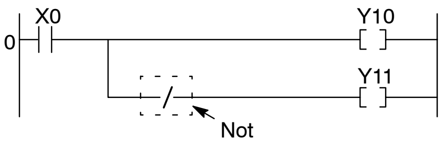


Some input devices, such as emergency stop switches, usually have a Form B (normally closed) contact. When an emergency stop switch with a Form B contact is programmed, be sure to use the **ST** instruction.

/ Not

Outline Inverts the operation result up to this instruction.

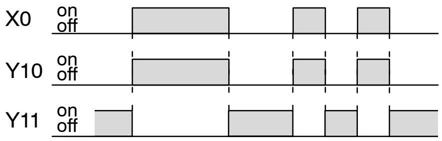
Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST X 0
	1	OT Y 10
	2	/
	3	OT Y 11

Explanation of example

Y10 goes on and Y11 goes off when X0 turns on.

Y10 goes off and Y11 goes on when X0 turns off.



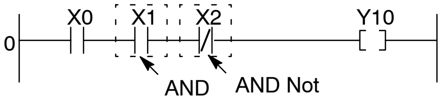
Description

The / instruction inverts the operation result up to this instruction.

AN — AND
AN/ — AND Not

Outline **AN:** Connects Form A (normally open) contacts in series.
AN/: Connects Form B (normally closed) contacts in series.

Program example

Ladder Diagram	Boolean			
	Address	Instruction		
	0	ST	X	0
	1	AN	X	1
	2	AN/	X	2
	3	OT	Y	10

Operands

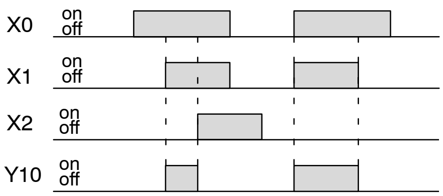
Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L (*1)	P (*2)	E (*3)	T	C	(*4)
AN, AN/	A	A	A	A	A	A	A	A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0/FP−e.
(*2) This can be used only with the FP2/FP2SH/FP10SH.
(*3) This can be used only with the FP2SH/FP10SH.
(*4) This can be used only with the FP0R/FP2/FP2SH/FP10SH.

Explanation of example

Y10 goes on when both X0 and X1 turn on and also X2 turns off.



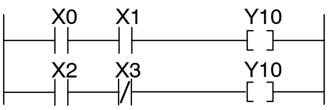
Description

Performs a logical AND operation with the results of the immediately preceding serially connected operation.

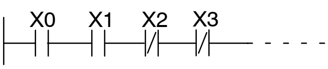
Precautions during programming

Use the **AN** instruction when normally open contacts (Form A contacts) are serially connected.

Use the **AN/** instruction when normally closed contacts (Form B contacts) are serially connected.



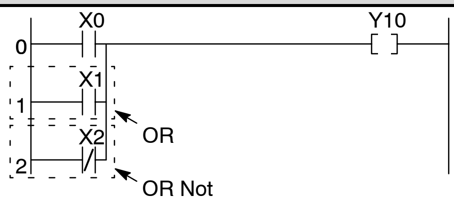
The **AN** and **AN/** instructions can be used consecutively.



OR — OR
OR/ — OR Not

Outline **OR:** Connects Form A (normally open) contacts in parallel.
OR/: Connects Form B (normally closed) contacts in parallel.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST X 0
	1	OR X 1
	2	OR/ X 2
	3	OT Y 10

Operands

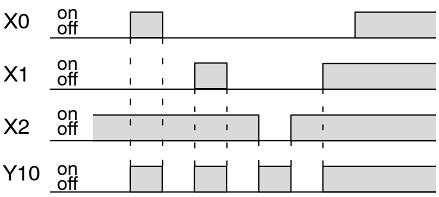
Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L (*1)	P (*2)	E (*3)	T	C	(*4)
OR, OR/	A	A	A	A	A	A	A	A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0/FP-e.
(*2) This can be used only with the FP2/FP2SH/FP10SH.
(*3) This can be used only with the FP2SH/FP10SH.
(*4) This can be used only with the FP0R/FP2/FP2SH/FP10SH.

Explanation of example

Y10 goes on when either X0 or X1 turns on or X2 turns off.



Description

Performs a logical OR operation with the results of the immediately preceding operation connected in parallel.

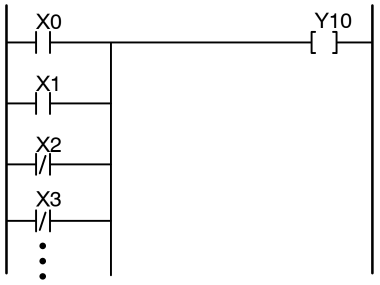
Precautions during programming

Use the **OR** instruction when normally open contacts (Form A contacts) are connected in parallel.

Use the **OR**/ instruction when normally closed contacts (Form B contacts) are connected in parallel.

The **OR** instruction starts from the bus line.

The **OR** and **OR**/ instructions can be used consecutively.



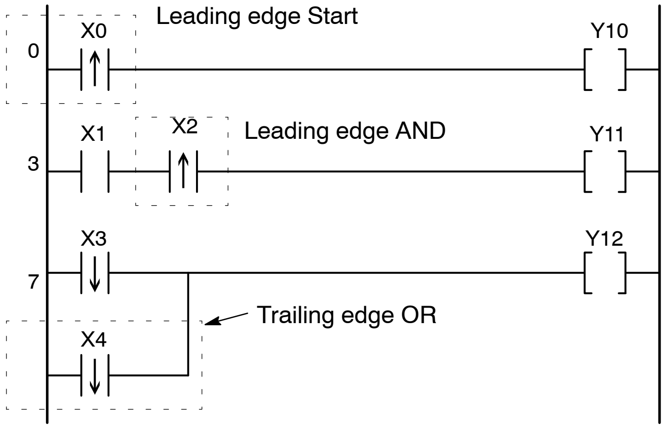
ST↑	**Leading edge Start**
ST↓	**Trailing edge Start**
AN↑	**Leading edge AND**
AN↓	**Trailing edge AND**
OR↑	**Leading edge OR**
OR↓	**Trailing edge OR**

Availability
FP2/FP2SH/FP10SH FP–X (V2.00 or more) FPΣ (V3.10 or more) FP0R

Outline Contact instructions for leading edge detection and trailing edge detection
Logic processing is only carried out during the scan following detection of a leading edge or trailing edge in the signal.

Program example

Ladder Diagram



Boolean

Address	Instruction		
0	ST↑	X	0
2	OT	Y	10
3	ST	X	1
4	AN↑	Y	2
6	OT	X	11
7	ST↓	Y	3
9	OR↓	X	4
11	OT	Y	12

Operands

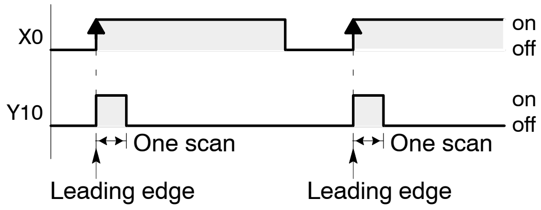
Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L	P	E	T	C	
ST↑, ST↓	A	A	A	A	A	N/A	A	A	N/A
AN↑, AN↓	A	A	A	A	A	N/A	A	A	N/A
OR↑, OR↓	A	A	A	A	A	N/A	A	A	N/A

A: Available
N/A: Not Available

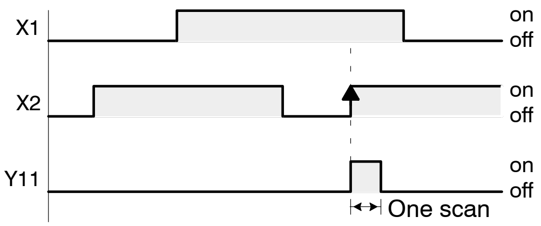
Explanation of example

ST↑, AN↑ and OR↑ instructions

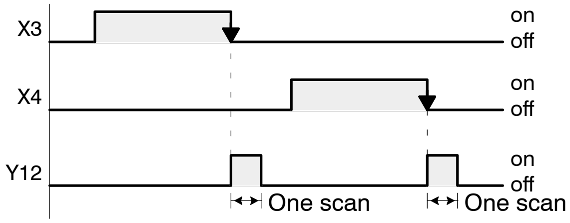
Output to Y10 takes place for one scan only following a change in X0 from off to on.



Output to Y11 takes place for one scan only following a change in X2 from off to on when X1 is on.



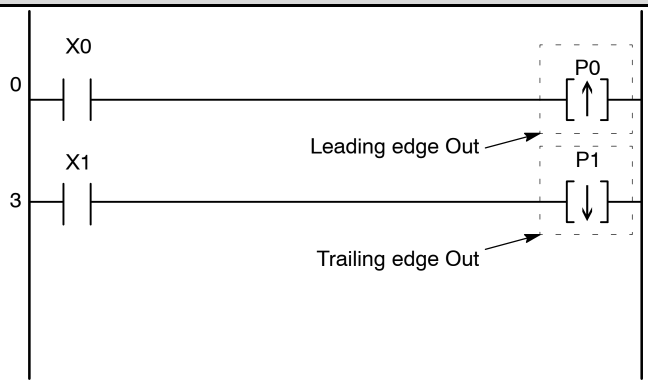
Output to Y12 takes place for one scan only following a change in X3 or X4 from on to off.



OT↑ Leading edge Out
OT↓ Trailing edge Out

Outline Leading edge detection and trailing edge detection output
The result of processing is output to the pulse relay for one scan only.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	0	ST X 0
	1	OT↑ P 0
	3	ST X 1
	4	OT↓ P 1

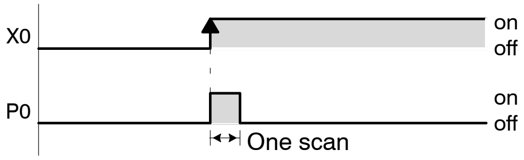
Operands

Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L	P	E	T	C	
OT↑	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
OT↓	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A

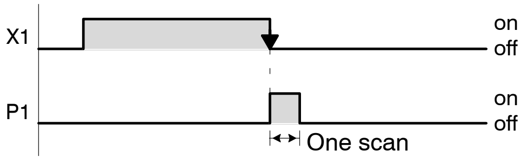
A: Available
N/A: Not Available

Explanation of example

Output to pulse relay “P0” takes place for one scan only following a change in X0 from off to on.



Output to pulse relay “P1” takes place for one scan only following a change in X1 from on to off.



Description

OT↑ instructions

Output to the pulse relay takes place for one scan only following a change in the immediately previous processing result from off to on. The pulse relay goes on for one scan only.

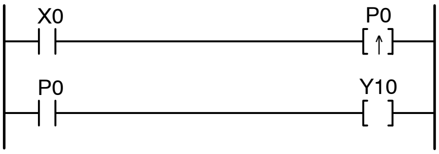
OT↓ instructions

Output to the pulse relay takes place for one scan only following a change in the immediately previous processing result from on to off. The pulse relay goes on for one scan only.

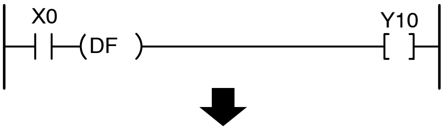
Precautions during programming

When the pulse relay (P) (which goes on for one scan only due to execution of a **OT↑** or **OT↓** instruction) is used with a logic instruction (**ST**, **AN** or **OR**), operation is the same as a normal contact followed by **DF** instruction.

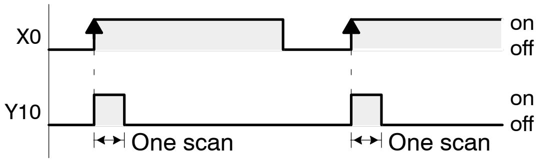
Example using an **OT↑** instruction and the pulse relay (P)



Example using a **DF** instruction



Both example are executed as shown below.



ALT Alternative out

Outline Inverts the output condition each time the leading edge of the signal is detected.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST X 0
	1	ALT Y 10

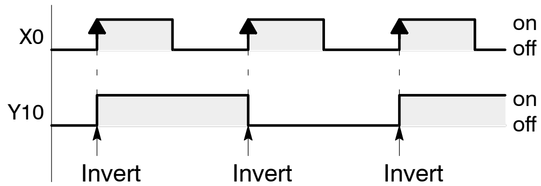
Operands

Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L	P	E	T	C	
ALT	N/A	A	A	A	N/A	N/A	N/A	N/A	N/A

A: Available
N/A: Not Available

Explanation of example

Each time X0 changes from off to on, the on/off state of output Y10 toggles.



Description

When the immediately previous processing result changes from off to on, the on/off state of the specified coil toggles.

The on/off state of the specified coil is held until an **ALT** instruction specifying that coil rises. (Flip-flop control)

Precautions during programming

During the interval that the input remains on, the output only toggles when the rise occurs, not after that.

When the mode is changed to RUN or the power is turned on in RUN mode such that the input is initially on, toggling does not occur at the first scan.

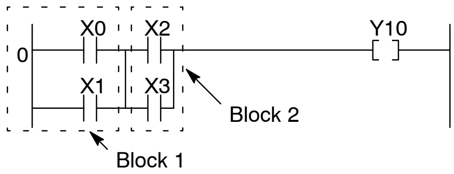
When used with instructions which change the order of execution such as **MC** to **MCE** and **JP** to **LBL** (see below), take care because the operation of instructions may change depending on the timing of instruction execution and input.

- **MC** to **MCE** instructions
- **JP** to **LBL** instructions
- **F19 (SJP)** to **LBL** instructions
- **LOOP** to **LBL** instructions
- **CNDE** instruction
- Step ladder instructions
- Subroutine instructions

ANS AND stack

Outline Multiple blocks are connected in series.

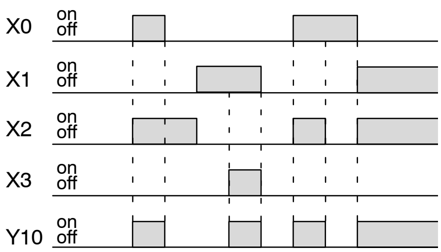
Program example

Ladder Diagram	Boolean			
	Address	Instruction		
	0	ST	X	0
	1	OR	X	1
	2	ST	X	2
	3	OR	X	3
	4	ANS		
	5	OT	Y	10

Explanation of example

Y10 goes on when X0 or X1 and X2 or X3 turn on.

$\underbrace{(\text{X0 OR X1})}_{\text{block 1}}$ AND $\underbrace{(\text{X2 OR X3})}_{\text{block 2}} \rightarrow$ Y10



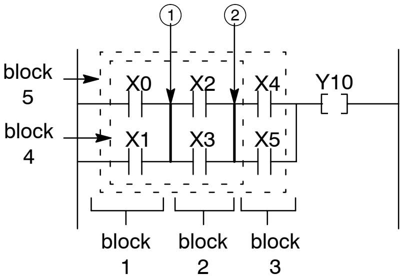
Description

Blocks connected in parallel are connected in series.

A block begins with the **ST** instruction.

When blocks are consecutive

When blocks are consecutive, a division of the blocks should be considered, such as that shown below.

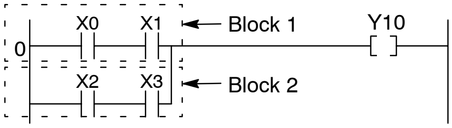


block 5	block 4	block 1	ST	X	0
			OR	X	1
		block 2	ST	X	2
			OR	X	3
			ANS 1		
		block 3	ST	X	4
			OR	X	5
			ANS 2		
			OUT	Y	10

ORS OR stack

Outline Multiple blocks are connected in parallel.

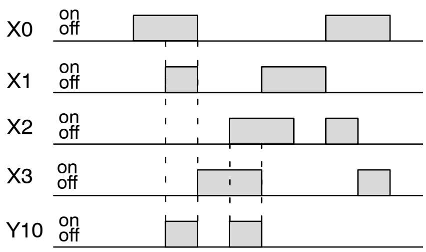
Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST X 0
	1	AN X 1
	2	ST X 2
	3	AN X 3
	4	ORS
	5	OT Y 10

Explanation of example

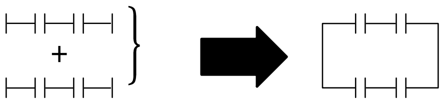
Y10 goes on when both X0 and X1 or both X2 and X3 turn on.

$$\underbrace{(X0\ AND\ X1)}_{block\ 1}\ OR\ \underbrace{(X2\ AND\ X3)}_{block\ 2} \rightarrow Y10$$



Description

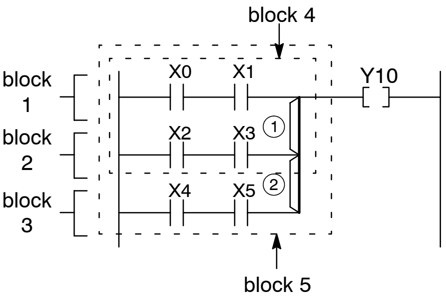
Blocks connected in series are connected in parallel.



A block begins with the **ST** instruction.

When blocks are consecutive

When blocks are consecutive, a division of the blocks should be considered, such as that shown below.



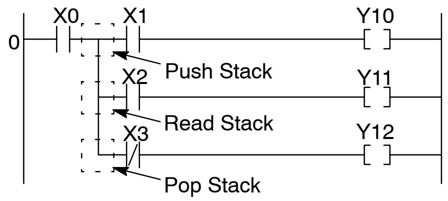
block 5	block 4	block 1	ST	X	0
			AN	X	1
		block 2	ST	X	2
			AN	X	3
			ORS .... ①		
		block 3	ST	X	4
			AN	X	5
			ORS .... ②		
			OUT	Y	10

PSHS Push stack
RDS Read stack
POPS Pop stack

Outline

PSHS: Stores the operation result up to this instruction.
RDS: Reads the operation result stored by the **PSHS** instruction.
POPS: Reads and clears the operation result stored by the **PSHS** instruction.

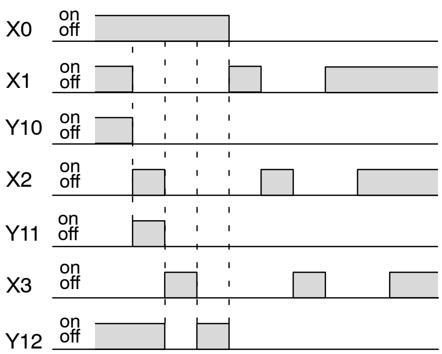
Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST X 0
	1	PSHS
	2	AN X 1
	3	OT Y 10
	4	RDS
	5	AN X 2
	6	OT Y 11
	7	POPS
	8	AN/ X 3
	9	OT Y 12

Explanation of example

When X0 turns on:

- Stores the operation result up to the **PSHS** instruction and Y10 goes on when X1 turns on.
- Reads the stored result using the **RDS** instruction and Y11 goes on when X2 turns on.
- Reads the stored result using the **POPS** instruction and Y12 goes on when X3 turns off. Also clears the result stored by the **PSHS** instruction.



Description

One operation result can be stored in memory and read, and multiple processes performed.

PSHS (stores operation result):

Stores the operation result up to this instruction and continues execution from the next step.

RDS (reads operation result):

Reads the operation result stored using the **PSHS** instruction and, using this result, continues operation from the next step.

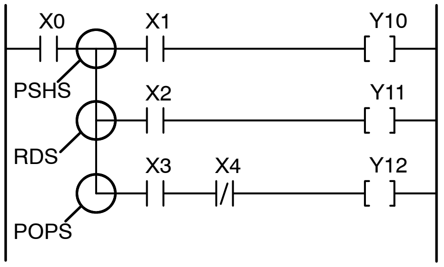
POPS (resets operation contents):

Reads the operation result stored using the **PSHS** instruction and, using this result, continues operation from the next step. Also clears the operation result stored by the **PSHS** instruction.

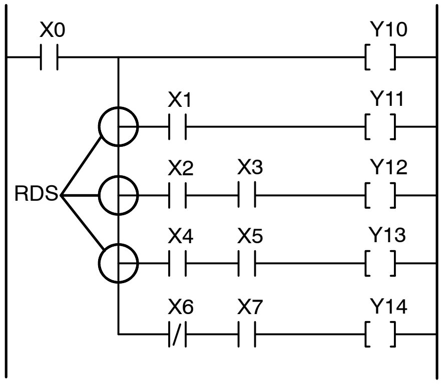
These instructions are used if there is branching from a single contact, followed by another contact or contacts.

Precautions during programming

You can continue to use the same operation result several times by repeatedly using the **RDS** instruction. When you are finished, be sure to issue the **POPS** instruction.



An **RDS** instruction can be used repeatedly any number of times.

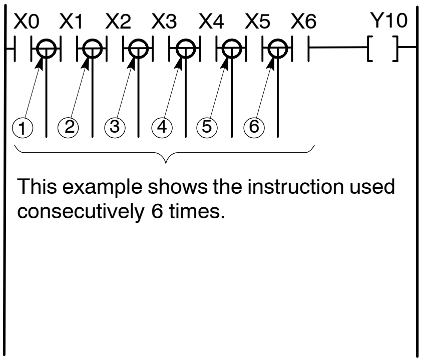


Caution regarding repeated use of a PSHS instruction

The **PSHS** instruction is limited in the number of times that it can be used consecutively. The number of times that the instruction can be used consecutively before the next **POPS** instruction is as shown below.

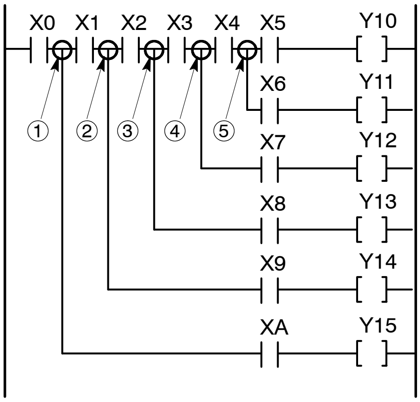
Type	No. of consecutive times
FP0, FP−e, FPΣ, FP−X , FP0R	Up to 8 times maximum
FP2, FP2SH, FP10SH	Up to 7 times maximum

If the instruction is used consecutively more than the allowable number of times, be aware that the program will not run correctly.




```
ST X 0
PSHS ....①
AN X 1
PSHS ....②
AN X 2
PSHS ....③
AN X 3
PSHS ....④
AN X 4
PSHS ....⑤
AN X 5
PSHS ....⑥
AN X 6
OT Y10
```

If a **POPS** instruction is used during repeated use of a **PSHS** instruction, reading will take place in order beginning from the last data stored by the **PSHS** instruction.




```
POPS ..... ⑤Reading
AN   X  6
OT   Y  11
POPS ..... ④Reading
AN   X  7
OT   Y  12
POPS ..... ③Reading
AN   X  8
OT   Y  13
  :
```

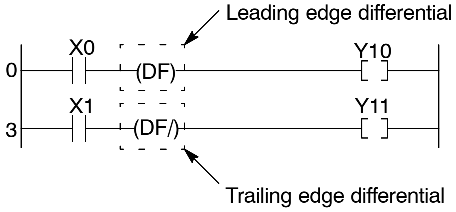
DF Leading edge differential
DF/ Trailing edge differential

Outline

DF: Turns on the contact for only one scan when the leading edge of the trigger is detected.

DF/: Turns on the contact for only one scan when the trailing edge of the trigger is detected.

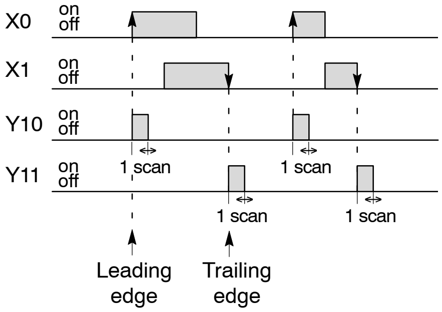
Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	0	ST X 0
	1	DF
	2	OT Y 10
	3	ST X 1
	4	DF/
	5	OT Y 11

Explanation of example

Y10 goes on for only one scan when the leading edge (off → on) of X0 is detected.

Y11 goes on for only one scan when the trailing edge (on → off) of X1 is detected.



Related instructions

With the FPΣ, FP−X, FP0R, FP2, FP2SH and FP10SH, the **DFI** instruction can be used. It is executed only for the first scan.

Description

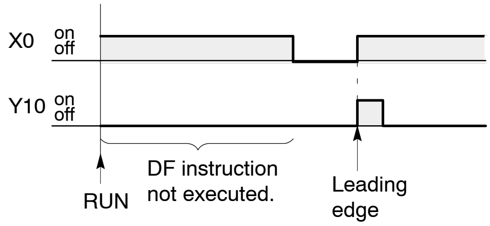
The **DF** instruction executes and turns on output for only one scan duration when the trigger changes from an off to an on state.

The **DF**/ instruction executes and turns on output for only one scan duration when the trigger changes from an on to an off state.

There is no limit on the number of times the **DF** and **DF**/ instructions can be used.

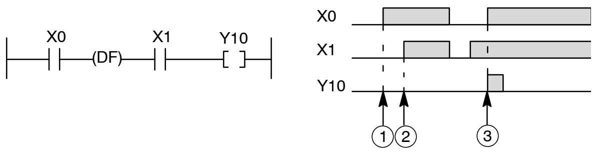
With the **DF** and **DF**/ differential instructions, only a change in the on and off status of the contact is detected. Thus, if the execution condition is initially on such as when the mode is changed to RUN or the power turned on in RUN mode, output will not be obtained.

Example: Leading edge differential (DF) instruction



Precautions during programming

With a program such as the one in the figure below, operation will be as follows.

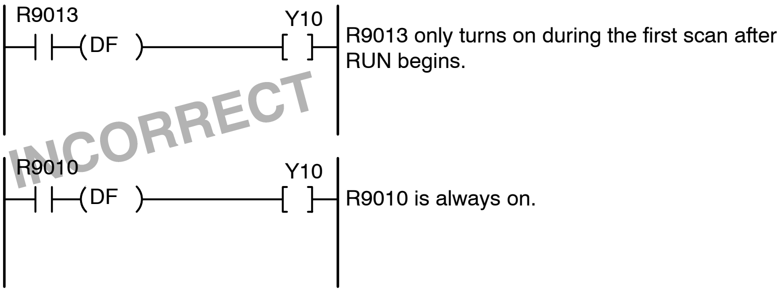


① When X1 is off, even if X0 rises, Y10 remains off.

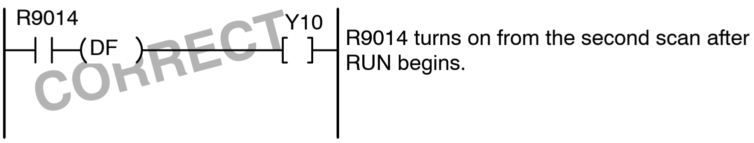
② Even if X1 rises when X0 is on, Y10 remains off.

③ If X0 rises when X1 is on, then Y10 will go on for one scan.

In the following program the execution condition is initially on, therefore output is not obtained.



With the following program, output is obtained.

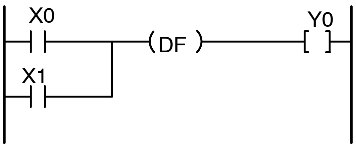


Caution is required when using a differential instruction with instructions which change the order of instruction execution such as **MC** and **MCE** or **JP** and **LBL** (below instructions).

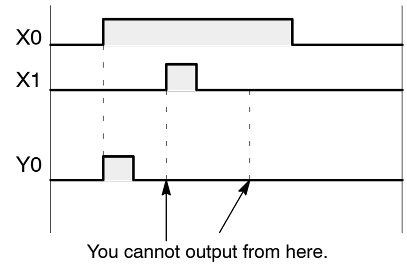
- **MC** to **MCE** instructions
- **JP** to **LBL** instructions
- **F19 (SJP)** to **LBL** instructions
- **LOOP** to **LBL** instructions
- **CNDE** instruction
- Step ladder instructions
- Subroutine instructions

When combining a differential instruction with an AND stack or pop stack instruction, take care that the syntax is correct.

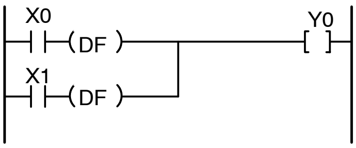
Operation is as follows with a circuit like the one shown below.



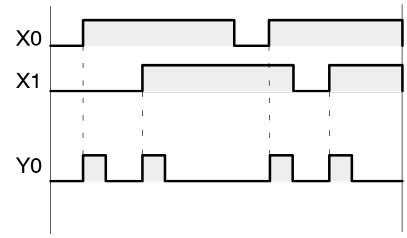
Time chart



Please use a program as follows when Y0 is turned on at the rise of either X0 or X1.



Time chart

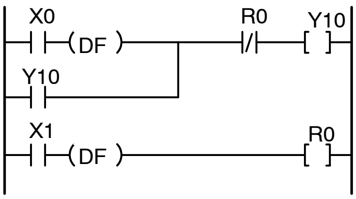


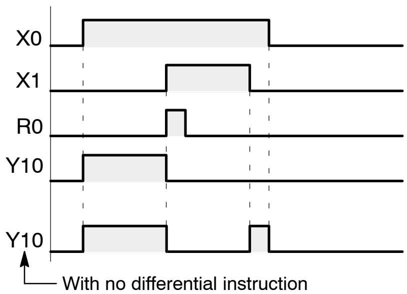
Example of a differential instruction application

Using a differential instruction makes it easier to adjust a program.

Application example for self−hold circuit

Using a differential instruction makes it possible to handle long input signals.

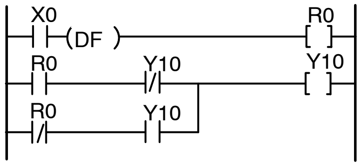




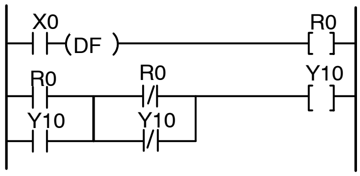
Application example for alternating circuit

A differential instruction can also be applied to an alternating circuit to hold and release the circuit using a single signal.

Example 1:



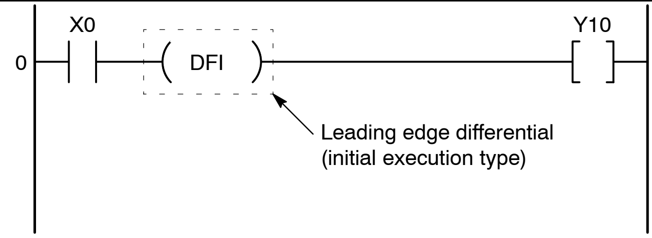
Example 2:



DFI Leading edge differential (initial execution type)

Outline When a leading edge of signal is detected, the contact goes on during that scan only. Leading edge detection is possible at the first scan.

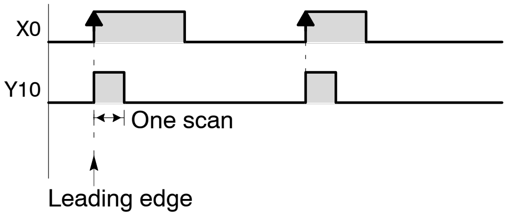
Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	0	ST X 0
	1	DFI
	3	OT Y 10

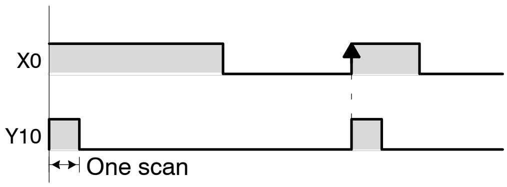
Explanation of example

Output to Y10 takes place for one scan only following a change in X0 from off to on.

When the trigger X0 is met after RUN is begun



When the trigger X0 is met before RUN



Description

When the trigger (execution condition) changes from off to on, the **DFI** instruction outputs (differential output) during the following scan only.

When the trigger (execution condition) is met before RUN is begun, output (differential output) takes place at the first scan.

There is no limit to the number of times the **DFI** instruction can be used.

When the mode is changed to RUN or the power is turned on in RUN mode and the trigger (execution condition) is already met, a **DF** instruction will not obtain output at the first scan. For this reason, use a **DFI** instruction.

Precautions during programming

When used with instructions which change the order of execution such as **MC** to **MCE** and **JP** to **LBL** (see below), caution must be exercised.

- **MC** to **MCE** instructions
- **JP** to **LBL** instructions
- **F19 (SJP)** to **LBL** instructions
- **LOOP** to **LBL** instructions
- **CNDE** instruction
- Step ladder instructions
- Subroutine instructions

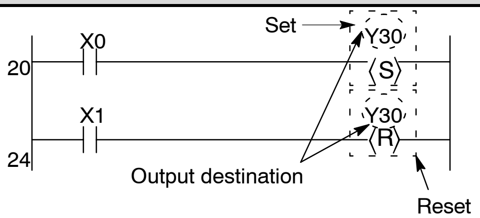
Take care that the syntax is correct when combining a differential instruction with an **ANS** or **POPS** instruction.

SET Set
RST Reset

Outline **SET:** When the execution conditions have been satisfied, the output is turned on, and the on status is retained.
RST: When the execution conditions have been satisfied, the output is turned off, and the off status is retained.

Program example

Ladder Diagram



Boolean				
Address	**Instruction**			
20	ST	X	0	
21	SET	Y	30	
24	ST	X	1	
25	RST	Y	30	

Operands

Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L (*1)	P	E (*2)	T	C	(*3)
SET, RST	N/A	A	A	A	N/A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0/FP−e.
(*2) This can be used only with the FP2SH/FP10SH.
(*3) This can be used only with the FP0R/FP2/FP2SH/FP10SH.

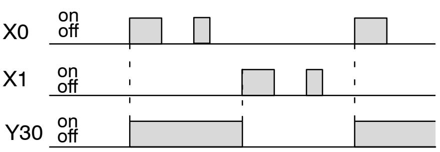
Operand	Relay				Timer/Counter		Register		Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	FL	I	K	H	M	
RST	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

* A word device can be used only by FP2/FP2SH.

Explanation of example

When X0 turns on, Y30 goes on and holds on.

When X1 turns on, Y30 goes off and stays off.



Description

The **SET** instruction executes when the trigger is turned on. Output turns on and holds on even if the trigger's state changes.

The **RST** instruction executes when the trigger is turned on. Output coil turns off and stays off even if the trigger's state changes.

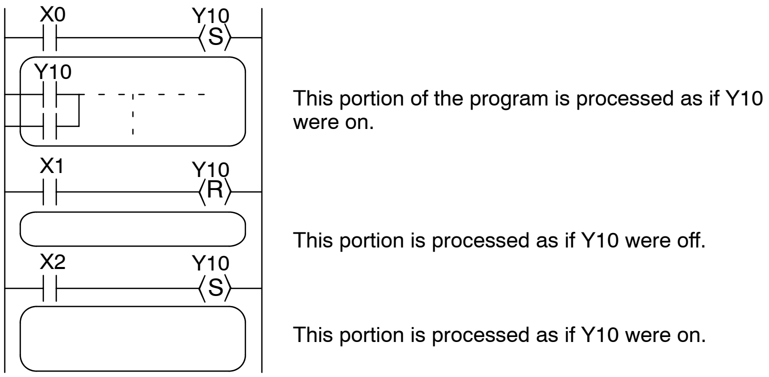
You can use relays with the same number as many times as you like with the **SET** and **RST** instructions. (Even if a total check is run, this is not handled as a syntax error.)

When SET and RST instructions are used

When the **SET** and **RST** instructions are used, the output changes with each step during processing of the operation.

Example:

When X0, X1, and X2 are turned on



I/O update is performed when an **ED** instruction is executed, therefore the data actually output is determined by the final operation result. In the above example, the Y10 output is on.

To output a result while operation is still in progress, use a partial I/O update instruction (**F143**).

Precautions during programming

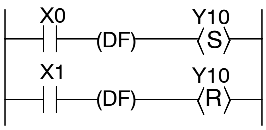
The output destination of a **SET** instruction is held even during the operation of an **MC** instruction.

The output destination of a **SET** instruction is reset when the mode is changed from RUN to PROG. or when the power is turned off, except when a hold type internal relay is specified as the output destination.

SET and RST instructions and differential instructions

Be sure to place a **DF** instruction before the **SET** and **RST** instructions to make program development and refinement easier.

This is particularly effective when the same output destination is used in several places in the program.



Precautions when using the FP2SH and FP10SH

It is not possible to specify a pulse relay (P) as the output destination for a **SET** or **RST** instruction.

All error alarm buffers can be cleared using the **RST** DT90400 instruction.
The head of the error alarm buffers can be cleared using the **RST** DT90401 instruction.

How relays are handled with SET and RST instructions

Relays can be turned off using the **RST** instruction.

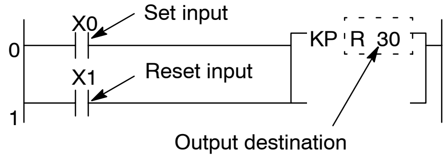
Using the various relays with the **SET** and **RST** instructions does not result in double output.

It is not possible to specify a pulse relay (P) as the output destination for a **SET** or **RST** instruction.

KP Keep

Outline This is output which is accompanied by set or reset input, and which is retained.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0 1 2	ST X 0 ST X 1 KP R 30

Operands

Instruction	Relay						Timer/Counter Contact		Index modifier
	X	Y	R	L (*1)	P	E	T	C	(*2)
KP	N/A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

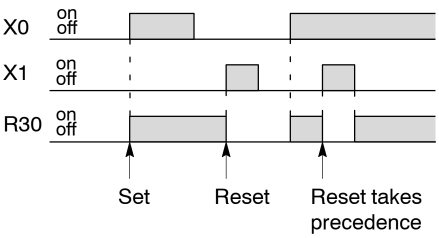
(*1) This cannot be used with the FP0/FP−e.

(*2) This can be used only with the FP2/FP2SH/FP10SH.

Explanation of example

When X0 turns on, output relay R30 goes on and stays on.

R30 goes off when X1 turns on.



Description

When the set input turns on, output of the specified relay goes on and stays on.

Output relay goes off when the reset input turns on.

The output relay's on state is maintained until a reset input turns on, regardless of the on or off states of the set input.

If the set input and reset input turn on simultaneously, the reset input has priority.

Precautions during programming

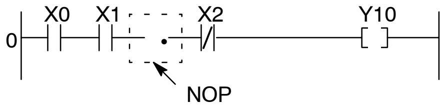
When the **KP** instruction is programmed between the **MC** and **MCE** instructions, the status of output destination (relay) specified by the **KP** instruction is maintained.

If a internal relay (R) specified by the **KP** instruction is set as the non-hold type, it is reset when the mode of operation is changed from RUN to PROG or when the power is turned off.
(If an internal relay set as a hold type is specified as the output destination, a reset does not take place.)

NOP No operation

Outline No operation

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST X 0
	1	AN X 1
	2	NOP
	3	AN/ X 2
	4	OT Y 10

Description

This instruction has no effect on the operation result to that point.

The same operation takes place without a **NOP** instruction.

The **NOP** instruction can be used to make the program easier to read when checking or correcting.

When you want to delete an instruction without changing addresses, write a **NOP** instruction (overwrite the previous instruction).

When you want to move the addresses of one part of a program without changing the program, insert a **NOP** instruction.

This is a convenient means of breaking a long program into several blocks.

Example:

To move the starting point of a program block from address 39 to address 40, insert a NOP instruction at address 39.

Address	Instruction
36	ST X0
•	OR X1
•	OT Y10
39	ST X2
40	AN X3
•	OT R20
•	ST R2
•	DF
44	ST X3

This moves the starting point to address 40.

Address	Instruction
36	ST X0
•	OR X1
•	OT Y10
39	NOP
40	ST X2
41	AN X3
•	OT R20
•	ST R2
•	DF
45	ST X3

Insert a **NOP** instruction.

Deleting a NOP instruction

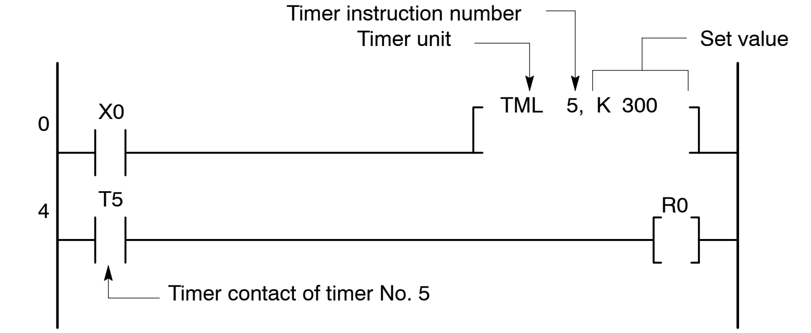
To delete the **NOP** instruction after editing in the PROG. mode, use the programming tools.

TML Timer (0.001s units)

Outline Sets the on-delay timer for 0.001s units

Program example

Ladder Diagram



Boolean

Address	Instruction		
0	ST	X	0
1	TM	L	5
	K		300
4	ST	T	5
5	OT	R	0

Operands

Instruction	Relay				Timer/ Counter		Register			Index register		Constant		Index modifier
	WX (*1)	WY (*1)	WR (*1)	WL (*1)	SV	EV (*1)	DT (*1)	LD (*1)	FL (*3)	IX	IY	K	H	(*2)
Set value	A	A	A	A	A	A	A	A	A	N/A	N/A	A	N/A	A

A: Available
N/A: Not Available

(*1) This can be used with the FP2SH/FP10SH/FP−X (V2.0 or more)/FPΣ (V3.10 or more)/FP0R.
(*2) This can be used with the FP2/FP2SH/FP10SH.
(*3) This can be used with the FP2SH/FP10SH.

Description

The timer is reset and does not retain its data when the power is turned off or the mode is changed from RUN to PROG. (If you need to retain the operating state , set system register 6. In that case, a battery must be used.)
Note) The FP0 T32 is the type with a built−in secondary battery.

When the trigger (execution condition) is on, the set time decrements until the elapsed value reaches zero, and at this point the timer contact Tn (n represents the timer contact number) goes on.

If the trigger (execution condition) goes off during decrement operation, operation stops and the elapsed value is reset to zero (cleared).

An **OT** instruction can appear immediately after a timer coil.

Setting the time in the timer

The time setting is equal to the time increment multiplied by the value set in the timer.

The value set in the timer can be a decimal value within the range K1 to K32767. The time increment is 0.001 seconds, producing a time range of 0.001 to 32.767 seconds.

Example:

When the value K43 is set, the time will be 0.001 × 43 = 0.043 seconds.

When the K500 is set, the set time will be 0.001 × 500 = 0.5 seconds.

Precautions during programming

The timer value decrements during processing, therefore, create the program so that one decrement occurs during one scan. (A correct result will not be obtained if no processing operations or multiple processing operations take place during one scan due to an interrupt program or **JP**/**LOOP** instruction.) If multiple processing operations are needed during one scan, set system register 4.

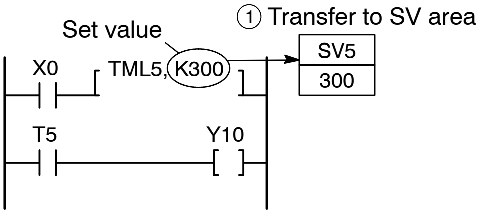
Take care that the syntax is correct when combining a timer instruction with an **ANS** or **POPS** instruction.

1) Specifying the timer setting with a decimal constant K

Timer operation when a decimal constant K is specified

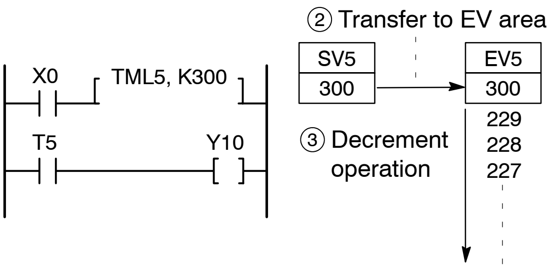
When a K constant is specified for the timer setting, the memory area SV with the same number as the timer number is used as the setting value area.

① When the mode is changed to RUN or the power is turned on in RUN mode, the timer setting will be transferred to the setting value area SV with the same number as the timer.

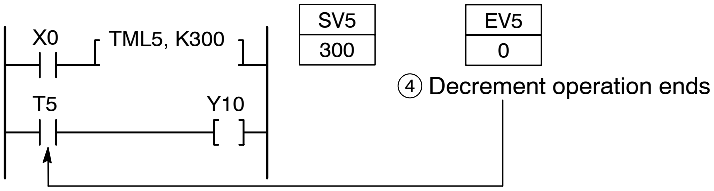


② When the trigger X0 (timer execution condition) rises from off to on, the setting is transferred from the setting value area SV to the elapsed value area EV with the same number.
(The same operation takes place if the mode is changed to RUN while the trigger (execution condition) is on.)

③ With each scan, the value in the elapsed value area EV decrements if the trigger (execution condition) is on.



④ When the value in the elapsed value area EV reaches zero, the timer contact T with the same number goes on.

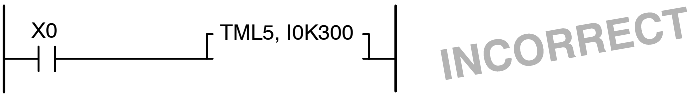


Important points when specifying constant (K)

The constant (K) can be changed during RUN.

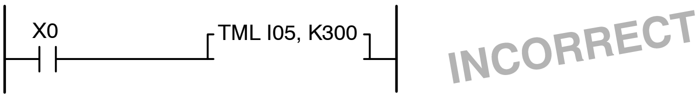
A specified constant (K) cannot be modified by index modification.

This program cannot be executed.



When the constant (K) is specified, the timer number cannot be modified by index modification.

This program cannot be executed.

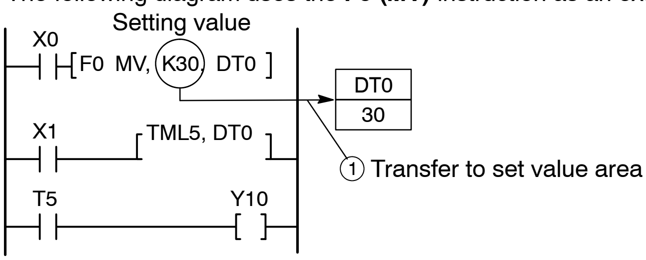


2) Using a word memory area for a timer setting

Timer operation when a word memory area is specified

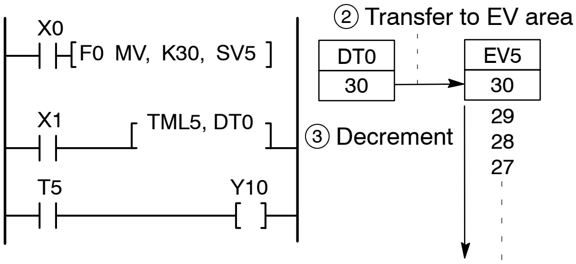
A word memory area specified as a set value is used as a setting value area

① When the execution condition (X0) for a high-level instruction goes on, the setting value is set in the specified area (this explanation uses DT0 as an example).
The following diagram uses the **F0 (MV)** instruction as an example.

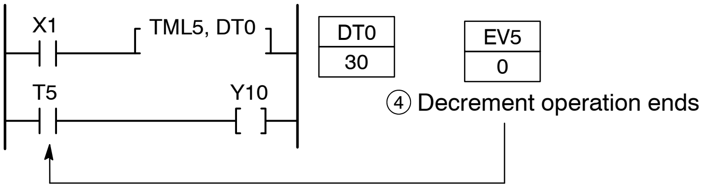


② When the timer execution condition rises from off to on, the value is transferred from the setting value area (DT0 in this example) to the elapsed value area EV with the same number as the timer.
(The same operation takes place if the mode is changed to RUN while the trigger (execution condition) is on.)

③ With each scan, the value in the elapsed value area EV decrements if the trigger (execution condition) is on.



④ When the value in the elapsed value area EV reaches zero, the timer contact T with the same number goes on.



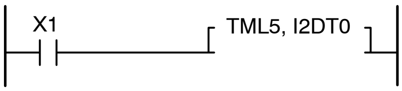
Important points when specifying a word memory area

Even if the value of the specified word memory area is changed during decrement operation, decrement operation will continue using the value prior to the change. Timer operation using the new value will not begin until the next time the execution condition changes from off to on.

There are both word memory area which reset (non-hold type) and do not reset (hold type) when the power is turned off or the mode changed from RUN to PROG. If you need to retain the value written to a word memory area when the power is turned on a second time, or after the mode is changed from RUN to PROG., use a memory area which has been set for hold type with the system register.

When a word memory area is used for a set value, the memory area address and timer number can be modified by index modification.

Example: Modifying a memory area address



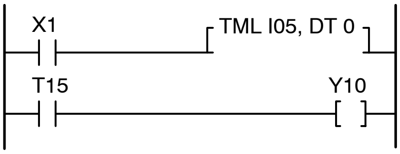
When I2 = K10, DT10 is used as the setting value area.

– Setting value area: DT10

– Elapsed value area: EV5

– Timer contact: T5

Example: Modifying a timer number



When I0 = K10, the timer operates as TML15.

– Setting value area: DT0

– Elapsed value area: EV15

– Timer contact: T15

The timer contact can also be modified by index modification.

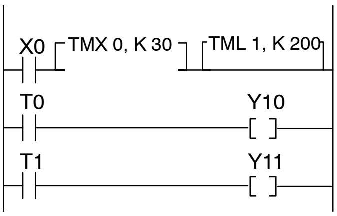
Notes

- **When a timer number is modified, the number of steps is 4 regardless of the value in the index register.**
- **When both the memory area address and timer number are modified, different index registers can be used for each.**

Examples of timer instruction applications

Serial connection of timer

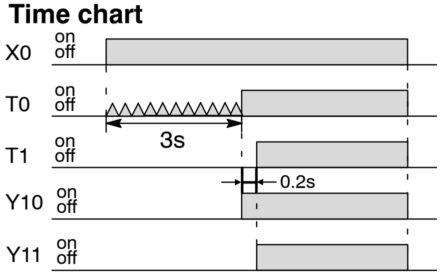
Ladder diagram



Boolean

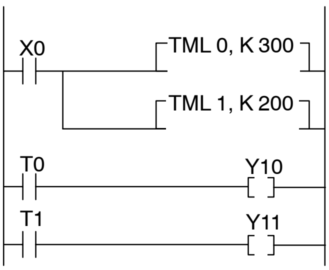
ST	X	0
TMX		0
K		30
TML		1
K		200
ST	T	0
OT	Y	10
ST	T	1
OT	Y	11

Time chart



Parallel connection of timer

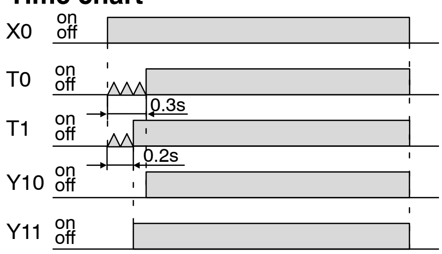
Ladder diagram



Boolean

ST	X	0
PSHS		
TML		0
K		300
POPS		
TML		1
K		200
ST	T	0
OT	Y	10
ST	T	1
OT	Y	11

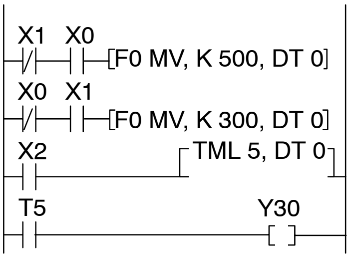
Time chart



Changing set values based on specified conditions

The set value is K50 when X0 is on and K30 when X1 is on.

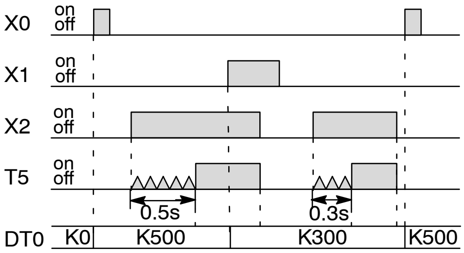
Ladder diagram



Boolean

ST/	X	1
AN	X	0
F0		(MV)
	K	500
	DT	0
ST/	X	0
AN	X	1
F0		(MV)
	K	300
	DT	0
ST	X	2
TML		5
	DT	0
ST	T	5
OT	Y	30

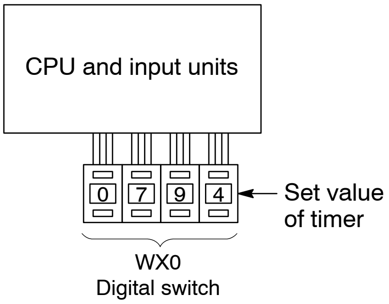
Time chart



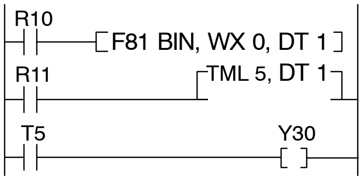
Example of setting a set value from external digital switches

The BCD data of the digital switches connected to X0 through XF is converted and becomes the set value

Connection example



Ladder diagram



Boolean

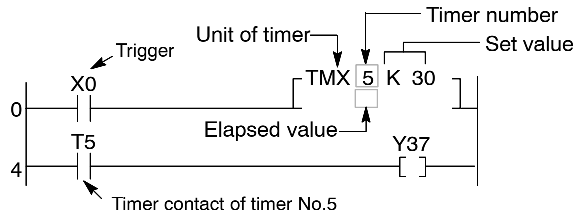
ST	R	10
F81		(BIN)
	WX	0
	DT	1
ST	R	11
TML		5
	DT	1
ST	T	5
OT	Y	30

TMR Timer (0.01s units)
TMX Timer (0.1s units)
TMY Timer (1.0s units)

Outline **TMR:** Sets the on-delay timer for 0.01 s units
TMX: Sets the on-delay timer for 0.1 s units
TMY: Sets the on-delay timer for 1.0 s units

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
0	ST	X	0
1	TMX		5
	K		30
4	ST	T	5
5	OT	Y	37

Operands

Instruction	Relay				Timer/ Counter		Register			Index register		Constant		Index modifier
	WX (*1)	WY (*1)	WR (*1)	WL (*1)	SV	EV (*1)	DT (*1)	LD (*1)	FL (*3)	IX	IY	K	H	(*2)
Set value	A	A	A	A	A	A	A	A	A	N/A	N/A	A	N/A	A

A: Available
N/A: Not Available

(*1) This can be used with the FP2SH/FP10SH/FP−X (V2.0 or more)/FPΣ (V3.10 or more)/FP0R.
(*2) This can be used with the FP2/FP2SH/FP10SH.
(*3) This can be used with the FP2SH/FP10SH.

Description

The timer is a non−hold type that is reset if the power is turned off, or if the mode is changed from the RUN to the PROG mode. (If it is necessary to hold the operation state, set system register 6. In that case, a battery must be used.)
Note) The FP0 T32 is the type with a built−in secondary battery.

When the trigger is on, the set time [n] decrements, and when the elapsed value reaches zero, timer contact Tn (n is the timer contact number) turns on.

If the trigger turns off during operation, operation stops and the elapsed value is reset (cleared to 0).

An **OT** instruction can be entered immediately after a timer coil.

Timer set time

The formula of the timer set time is [the time unit] × [set value]

The timer setting [n] must be a decimal constant from K1 to K32767.

– **TMR** is from 0.01 to 327.67 seconds in increments of 0.01 seconds.

– **TMX** is from 0.1 to 3276.7 seconds in increments of 0.1 seconds.

– **TMY** is from 1 to 32767 seconds in increments of 1 second.

Example:

When K43 is set in TMX, the set time is 0.1 × 43 = 4.3 seconds.

When K500 is set in TMR, the set time is 0.01 × 500 = 5 seconds.

Precautions during programming

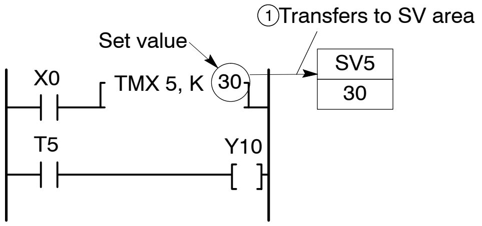
In order to ensure correct timer operation, the **TM** instruction should be executed in every scan. Be aware of this when using instructions like **INT**, **JP** and **LOOP**.

When a timer instruction is combined with an **ANS** or **POPS** instruction, take care that the syntax is correct.

Timer operation

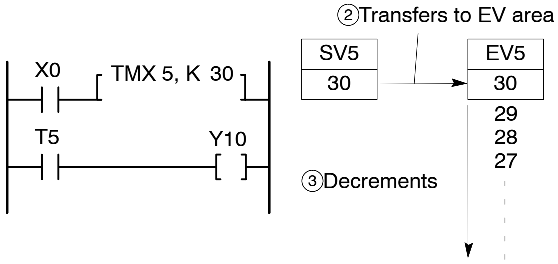
The following is an example of setting the set value with a K constant. For an explanation of operation when an set value area (SV) is specified, see the following pages.

① When the mode is changed to RUN or when the power is turned on with the mode set to RUN, the timer set value is transferred to the set value area (SV) with the same number.

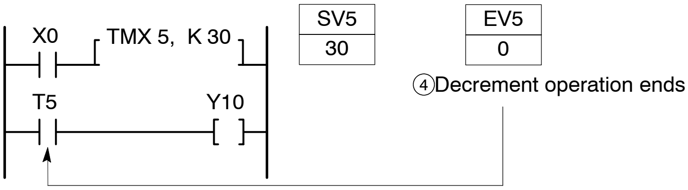


② When the timer trigger rises from off to on, the setting is transferred from the set value area (SV) to the elapsed value area (EV) with the same number. (The same operation takes place if the mode is changed to RUN when the trigger is on.)

③ The value in the elapsed value area (EV) decrements every scan if the trigger stays on.



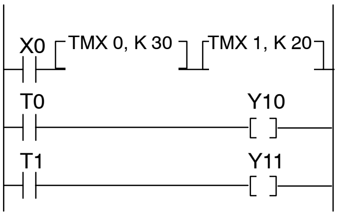
④ When the value in the elapsed value area (EV) reaches zero, the timer contact (T) with same number turns on.



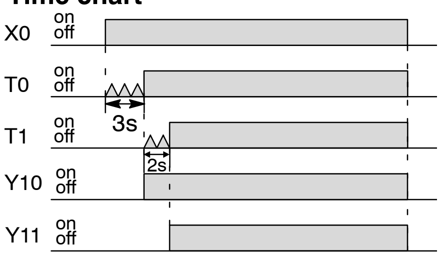
Examples of timer instruction applications

Serial connection of timer

Ladder diagram	Boolean	Time chart

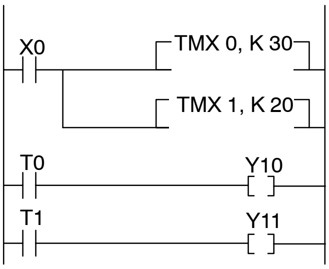


ST	X	0
TMX		0
K		30
TMX		1
K		20
ST	T	0
OT	Y	10
ST	T	1
OT	Y	11

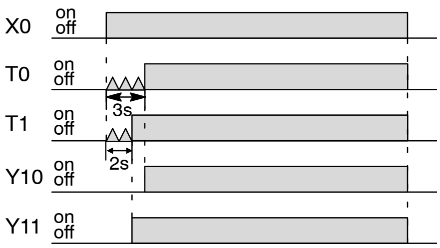


Parallel connection of timer

Ladder diagram	Boolean	Time chart

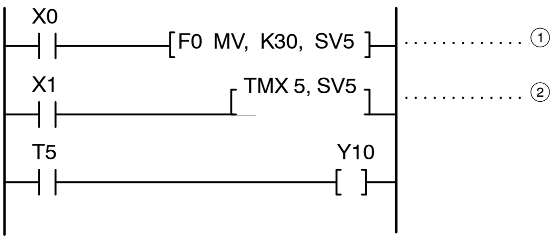


ST	X	0
PSHS		
TMX		0
K		30
POPS		
TMX		1
K		20
ST	T	0
OT	Y	10
ST	T	1
OT	Y	11



Directly specifying a set value area number as a timer setting value

With FP0/FP−e/FPΣ/FP−X/FP2/FP2SH/FP10SH with a CPU Ver. 4.4 or later with a CPU Ver. 2.7 or later, the set value area number can be specified directly as the set value n.

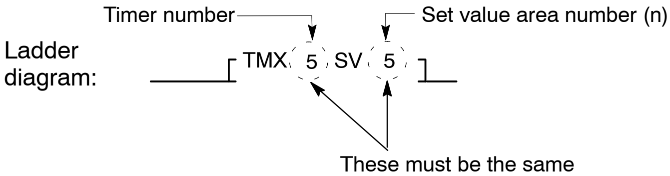


The above program operates as follows:

① When trigger X0 is on the data transfer instruction **F0 (MV)** is executed, set the K30 in SV5.

② When trigger X1 turns on, decrement operation begins from the set value 30.

Specify n (the number of the set value area SV) to be the same number as the timer.

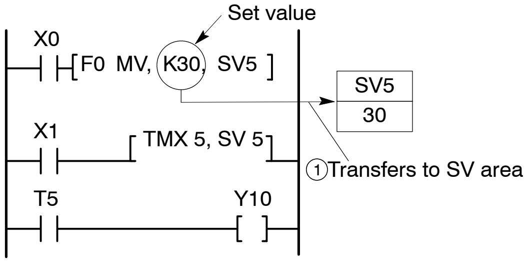


Even if the value of the set value area (SV) is changed during decrement operation, the decrement operation will continue from the value before the change. Timer operation from the new value will not begin until decrement operation has ended or is interrupted and the trigger subsequently changes from off to on.

The set value area (SV) is normally a non−hold type which resets if the power is turned off or the mode is changed from RUN to PROG.
If the SV value was changed while in the RUN mode, and that value is to be used as the set value, without being reset, the next time that the power supply is turned on, or when the mode is changed from RUN to PROG, system register 6 should be used to specify the value as a hold type.

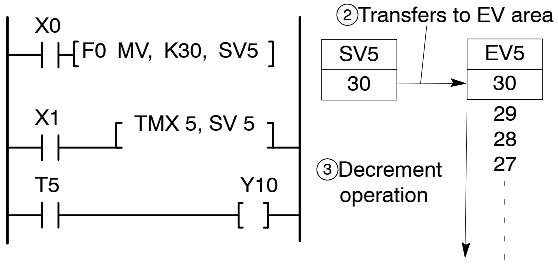
Timer operation when a set value area number is directly specified

① When the trigger for a high−level instruction is on, the value is set in the set value area (SV). The following diagram shows an example of using the high−level instruction **F0(MV)**.

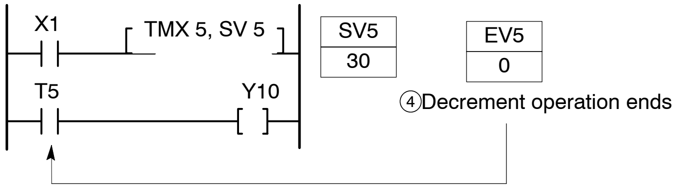


② When the timer trigger rises from off to on, the setting is transferred from the set value area (SV) to the elapsed value area (EV) with the same number. (The same operation takes place if the mode is changed to RUN when the trigger is on.)

③ The value in the elapsed value area (EV) decrements if the trigger stays on every scan.



④ When the value in the elapsed value area (EV) reaches zero, the timer contact (T) with same number turns on.

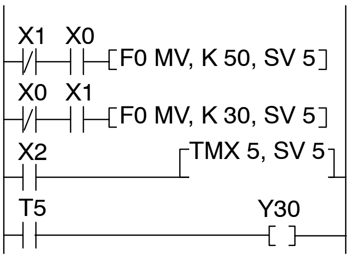


Examples of applying direct specification of set value area numbers

Changing set values based on specified conditions

The set value is K50 when X0 is on and K30 when X1 is on.

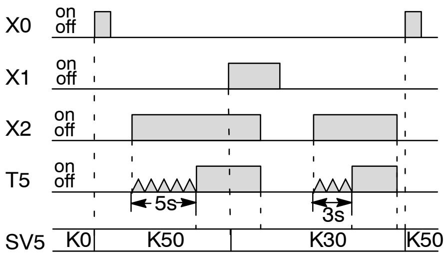
Ladder diagram



Boolean

```
ST/   X     1
AN    X     0
F0       (MV)
      K    50
      SV    5
ST/   X     0
AN    X     1
F0       (MV)
      K    30
      SV    5
ST    X     2
TMX         5
      SV    5
ST    T     5
OT    Y    30
```

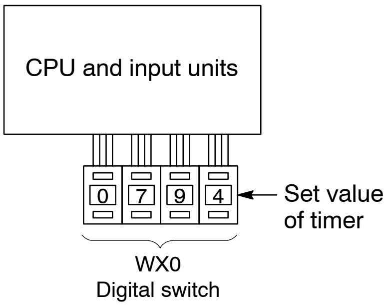
Time chart



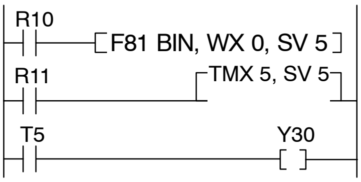
Example of setting a set value from external digital switches

The BCD data of the digital switches connected to X0 through XF is converted and becomes the set value

Connection example



Ladder diagram



Boolean

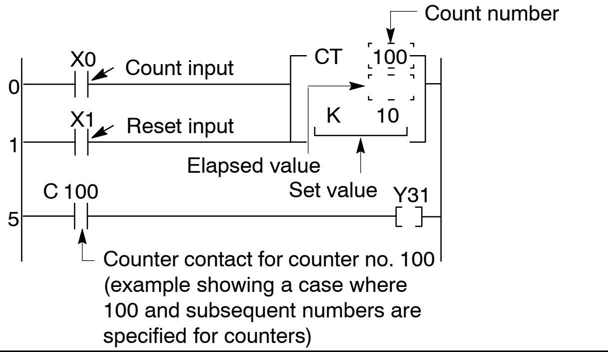
```
ST    R    10
F81      (BIN)
      WX    0
      SV    5
ST    R    11
TMX         5
      SV    5
ST    T     5
OT    Y    30
```

With the FP2SH/FP10SH, FP−X (Ver 2.0 or later), a memory area such as a data register DT can be specified as the set value. Regarding the operation, refer to the operation when specifying the SV.

CT Counter

Outline Decrements a preset counter.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	0 1 2 5 6	ST X 0 ST X 1 CT 100 K 10 ST C 100 OT Y 31

Operands

Instruction	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX (*1)	WY (*1)	WR (*1)	WL (*1)	SV	EV (*1)	DT (*1)	LD (*1)	FL (*3)	IX	IY	K	H	(*2)
Set value	A	A	A	A	A	A	A	A	A	N/A	N/A	A	N/A	A

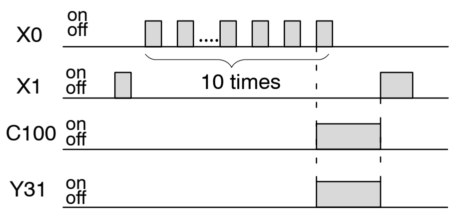
A: Available
N/A: Not Available

(*1) This can be used with the FP2SH/FP10SH/FP−X (V2.0 or more)/FPΣ (V3.10 or more)/FP0R.
(*2) This can be used with the FP2/FP2SH/FP10SH.
(*3) This can be used with the FP2SH/FP10SH.

Explanation of example

When the leading edge of X0 is detected ten times, counter contact C100 turns on and then Y31 goes on.

The elapsed value is reset when X1 turns on.



Description

The counter is a decremental preset counter.

At the fall time when the reset input goes from on to off, the value of the set value area (SV) is preset in the elapsed value area (EV).

When the reset input is on, the elapsed value is reset to 0.

When the count input changes from off to on, the set value begins to decrement, and when the elapsed value reaches 0, the counter contact Cn (n is the counter number) turns on.

If the count input and reset input both turn on at the same time, the reset input is given priority.

If the count input rises and the reset input falls at the same time, the count input is ignored and preset is executed.

An **OT** instruction can be entered immediately after a counter instruction.

Setting the counting value

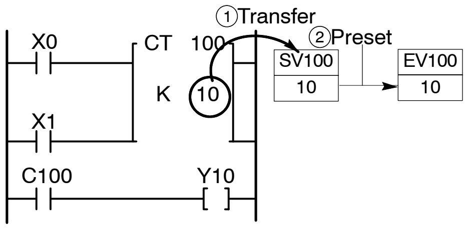
The counting value can be set to a decimal constant (K constant) from K0 to K32767.

Counter operation

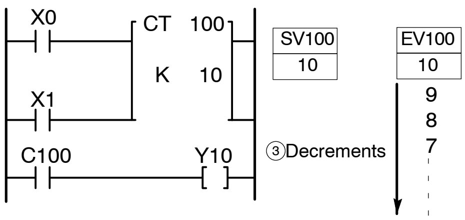
The following are examples of specifying a K constant as the set value. For an explanation of operation when a set value area number is specified, see following pages.
(This example shows a case in which “100” is specified for the counter.)

① When the mode is changed to RUN or the power is turned on with the mode set to RUN, the counter set value is transferred to the set value area (SV) with the same number.

② When the reset input falls, the value in the set value area (SV) is preset in the elapsed value area (EV).

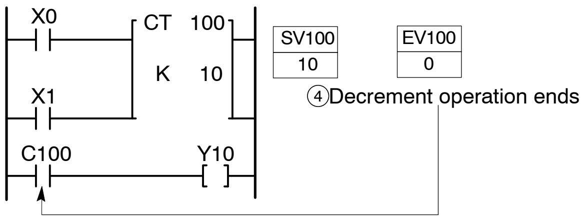


③ Each time the count input X0 turns on, the value in the elapsed value area (EV) decrements.



☛ next page

④ When the value in the elapsed value area (EV) reaches zero, the counter contact (C) with the same number turns on.



Precaution during programming

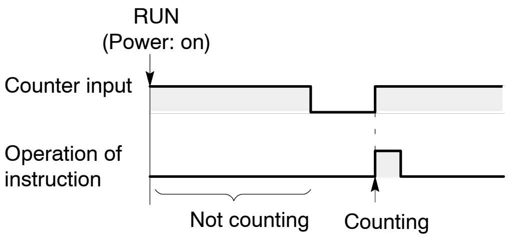
When combining a counter instruction with an **AND** stack instruction or pop stack instruction, take care that the syntax is correct.

Precautions of counting input detection

In a counter instruction, the decrement takes place when the rise of the count input from off to on is detected.

If the count input remains continuously on, since a decrement will only take place at the rise, no further subtraction will take place.

In cases where the count input is initially on such as when the mode is changed to RUN or the power is turned on with the mode set to RUN, decrement operation will not take place at the first scan.



When used in combination with instructions which change the order of instruction execution such as **MC** and **MCE** or **JP** and **LBL** (see below), the operation of the instruction may change depending on the timing of instruction execution and the count input. Exercise caution in these cases.

- **MC** to **MCE** instructions
- **JP** to **LBL** instructions
- **F19 (SJP)** to **LBL** instructions
- **LOOP** to **LBL** instructions
- **CNDE** instructions
- Step ladder instructions
- Subroutine instructions

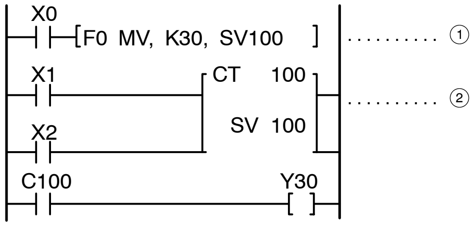
Related instructions

Counter instructions also include an up/down counter instruction (**F118**).

An increment instruction (**F35**) can be used to provide the same type of function.

Directly specifying a set value area number as a counter set value

With FP0/FP−e/FPΣ/FP−X/FP2/FP2SH/FP10SH with a CPU of Ver. 4.4 or later, the set value area number can be specified directly as the set value n.



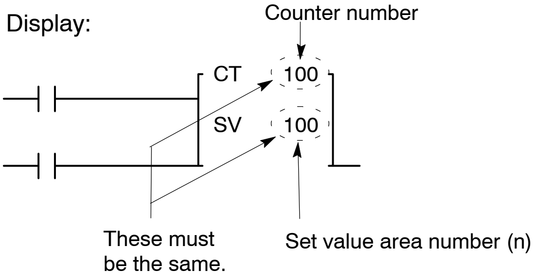
The above program operates as follows:

① When trigger X0 is on the data transfer instruction [**F0 (MV)**] is executed, set the K30 in SV100.

② When the count input X1 turns on, decrement operation begins from the set value 30.

Specify [n] (the number of the set value area SV) to be the same number as the counter.

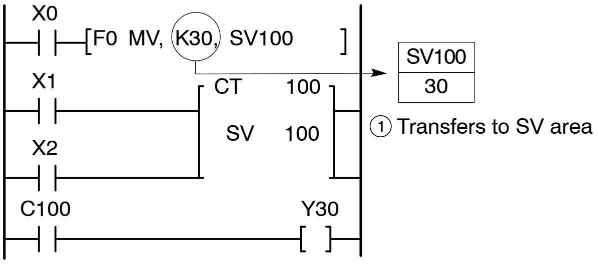
Display:



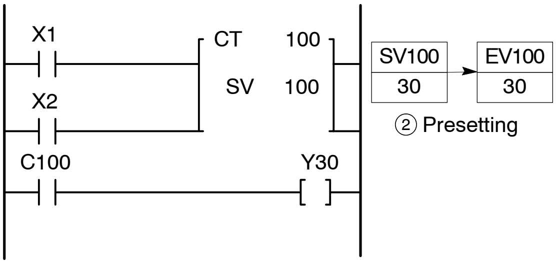
Even if the value in the set value area (SV) is changed during decrement operation, the decrement operation will continue from the value before the change. Counter operation from the new value will not begin until the counter is reset and the count input subsequently changes from off to on.

Counter operation when a set value area number is directly specified

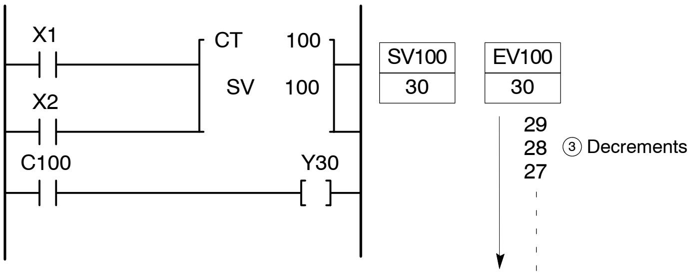
① When the trigger for a high−level instruction is on, the value is set in the set value area (SV). The following diagram shows an example of using the high−level instruction **F0 (MV)**.



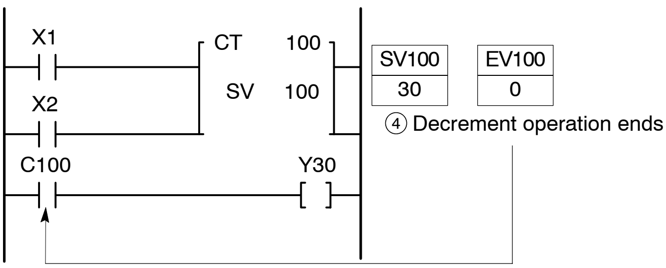
② When the reset input is off, the value in the set value area (SV) is preset in the elapsed value area (EV).



③ Each time the count input X1 turns on, the value in the elapsed value area (EV) decrements.



④ When the elapsed value area (EV) reaches zero, the counter contact C with the same number turns on.

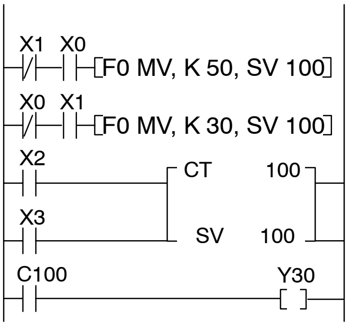


Examples of applying direct specification of set value area numbers

Changing set values based on specified conditions

The set value is K50 when X0 is on and K30 when X1 is on.

Ladder diagram

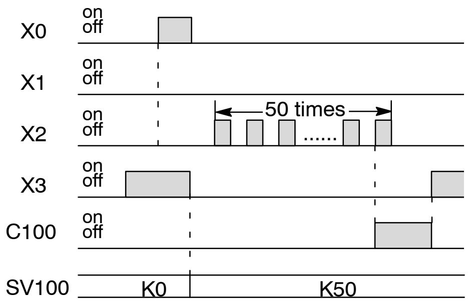


Boolean

```
ST/  X     1
AN   X     0
F0      (MV)
     K    50
     SV  100
ST/  X     0
AN   X     1
F0      (MV)
     K    30
     SV  100
ST   X     2
ST   X     3
CT       100
     SV  100
ST   C   100
OT   Y    30
```

Time chart

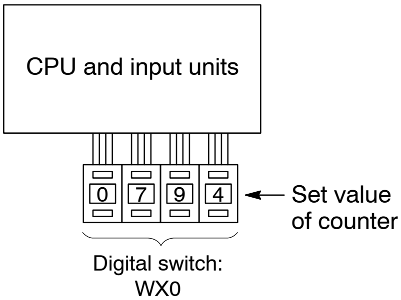
Example when X0 turns on



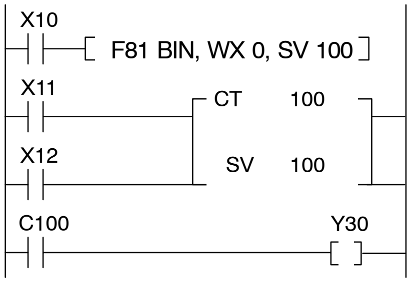
Setting a set value from external digital switches

The BCD data of the digital switches connected to X0 through XF is converted and becomes the set value

Connection diagram



Ladder diagram



Boolean

```
ST   X    10
F81     (BIN)
     WX    0
     SV  100
ST   X    11
ST   X    12
CT       100
     SV  100
ST   C   100
OT    Y   30
```

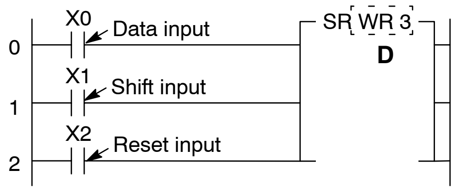
With the FP2SH/FP10SH, FP−X (Ver 2.0 or later), a memory area such as a data register DT can be specified as the set value. Regarding the operation, refer to the operation when specifying the SV.

SR Shift register

Outline One bit shift of 16-bit [word internal relay (WR)] data to the left.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
0	ST	X	0
1	ST	X	1
2	ST	X	2
3	SR	WR	3

Operands

Instruction	Relay				Timer/Counter		Register			Index register		Constant		Index modifier (*)
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
D: Data area	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A

(*) This can be used only with the FP2/FP2SH/FP10SH.

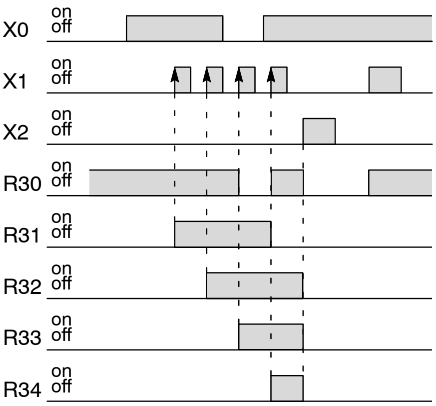
A: Available
N/A: Not Available

Explanation of example

If the X1 turns on when X2 is in the off state, the contents of the internal relay WR3 (internal relays R30 to R3F) are shifted one bit to the left.

“1” is shifted in R30 if X0 is on, and “0” is shifted in R30 if X0 is off.

If the X2 turns on, the contents of WR3 are reset to 0.



Description

Shifts the specified data area (WR) one bit to the left.

When the shift input turns on (rises), the contents of WR are shifted one bit to the left.

During the shift, 1 is set in the empty bit (least significant bit) if the data input is on, or 0 if the data input is off.

When shift input is turned on:

WR3	3F · · · · · · · · · · · · · · · 30			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	0 0 0 0	1 0 0 0	1 0 0 0	1 1 0 0

Shifts one bit to the left.

Binary data	0 0 0 1	0 0 0 1	0 0 0 1	1 0 0 0

Data input (X0) on: set bit to 1.
Data input (X0) off: set bit to 0.

When the reset input turns on, the contents of WR are cleared.

When reset input is turned on:

WR (Binary data)	0 0 1 1	0 1 0 0	0 0 0 1	1 0 0 1

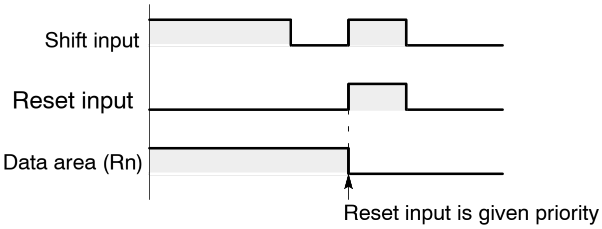
Contents of WR3 are cleared to 0.

WR (Binary data)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Precautions during programming

The **SR** instruction needs data input, a shift input, and a reset input.

When the reset input and the shift input are detected simultaneously, the reset input has priority.



If the internal relay area is specified as a hold type, take care that the data in the area is not reset to “0” when the power turns on.

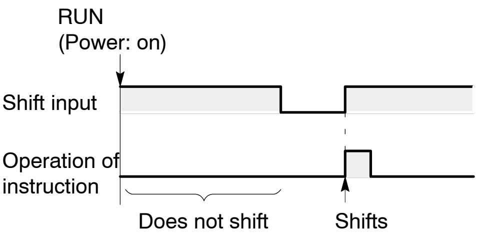
When combining a shift register instruction with an **ANS** or **POPS** instruction, take care that the syntax is correct.

Cautions on shift input detection

With **SR** instructions, shift operation takes place when the off–on rise of the shift input is detected.

If the shift input remains continuously on, a shift will only take place at the rise. No further shifts will take place.

In cases where the shift input is initially on such as when the mode is changed to RUN or when the power is turned on with the mode set to RUN, a shift operation will not take place at the first scan.



When used in combination with instructions which change the order of instruction execution such as **MC** and **MCE** or **JP** and **LBL** (see below), the operation of the instruction may change depending on the timing of instruction execution and the shift input.
Exercise caution in these cases.

- **MC** to **MCE** instructions
- **JP** to **LBL** instructions
- **F19 (SJP)** to **LBL** instructions
- **LOOP** to **LBL** instructions
- **CNDE** instruction
- Step ladder instructions
- Subroutine instructions

Related instructions

In addition to the shift register instruction, there is also a left/right shift register (**F119**).

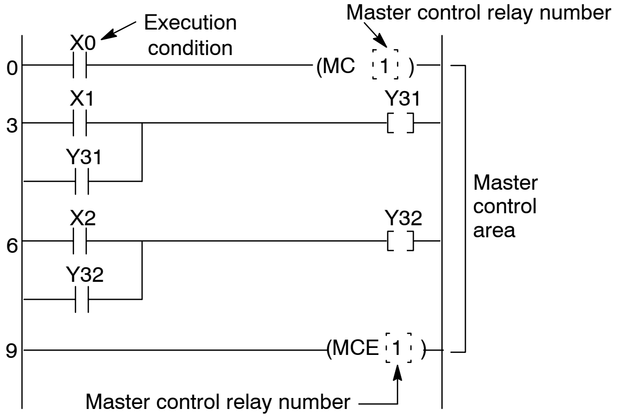
The same type of operation can also be implemented using a data shift instructions (**F100** to **F113**) or a data rotate instructions (**F120** to **F123**).

MC Master control relay
MCE Master control relay end

Outline Executes the program between the **MC** and **MCE** when the execution condition turns on.
When the execution condition is off, output between the **MC** and **MCE** is turned off.

Program example

Ladder Diagram

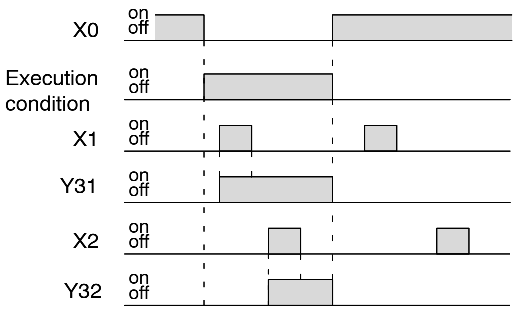


Boolean			
Address	Instruction		
0	ST/	X	0
1	MC		1
3	ST	X	1
4	OR	Y	31
5	OT	Y	31
6	ST	X	2
7	OR	Y	32
8	OT	Y	32
9	MCE		1

Explanation of example

Executes the program from the **MC1** instruction to the **MCE1** instruction when the execution condition X0 turns on.

If the execution condition is off, output is turned off without processing being carried out between the **MC1** and **MCE1** instructions.



Description

Executes program between the **MC** and **MCE** instructions when the execution condition turns on.

When the execution condition is in the off state, the instructions operate as follows.

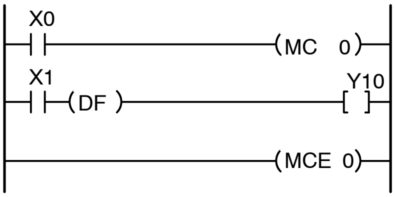
Instruction	Condition of input and output
OT	All off
KP	Holds the state.
SET	Holds the state.
RST	Holds the state.
TM	Reset
CT	Holds the value.
SR	Holds the value.
Differential	See next page.
Other instructions	Not executed

You must be careful when using one of the instructions below, which are executed by detecting the leading edge of execution condition such as the differential instruction.

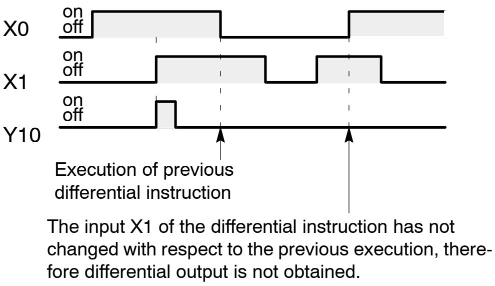
- **DF** instruction
- Count input for **CT** instruction
- Count input for **F118 (UDC)** instruction
- Shift input for **SR** instruction
- Shift input for **F119 (LRSR)** instruction
- **NSTP** instruction
- Differential execution type high−level instruction (this instruction is specified by P and a number)

Operation of differential instructions between MC and MCE

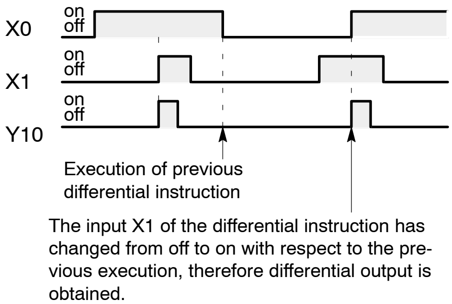
If a differential instruction is used between **MC** and **MCE**, the output will vary as follows depending on the timing of the **MC** execution condition and the input of differential instruction.



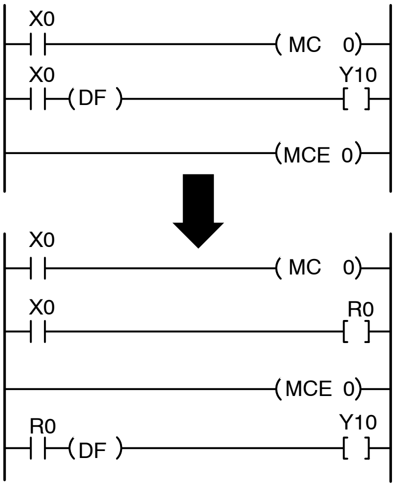
Time chart 1



Time chart 2

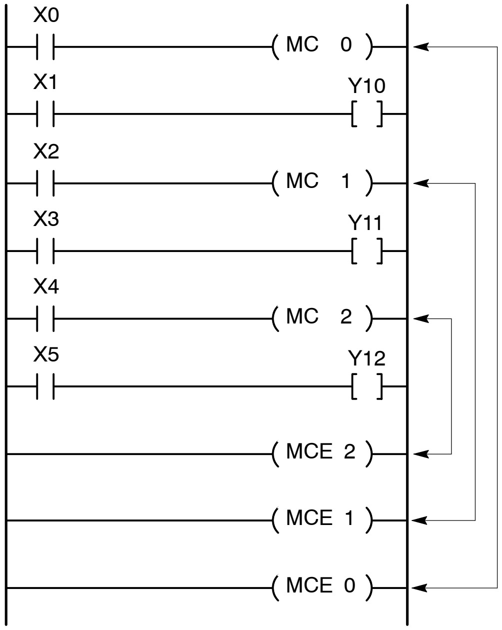


Output will not be obtained if the same execution condition is specified for an **MC** instruction and a differential instruction. If output is needed, enter the differential instruction outside of the **MC** to **MCE** instruction sequence.



Precautions during programming

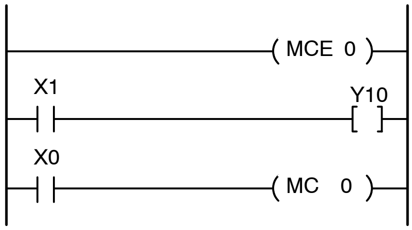
A second **MC**–**MCE** instruction pair can be entered (nested) between an initial **MC**–**MCE** instruction pair. (There is no limit to the number of nestings.)



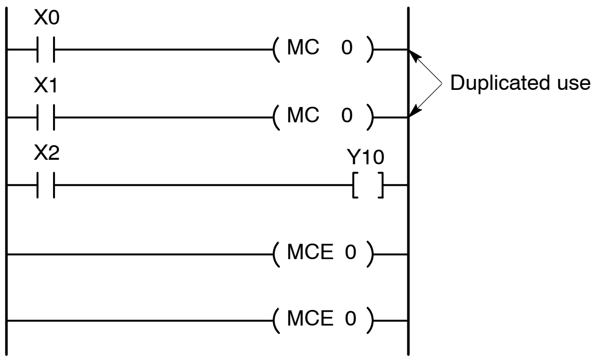
The program cannot be executed if:

If either **MC** or **MCE** is missing

The order of the **MC** and **MCE** instructions is reversed.



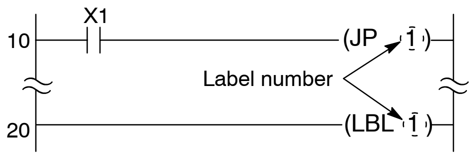
There are two or more master control instruction sets with the same number.



JP Jump
LBL Label

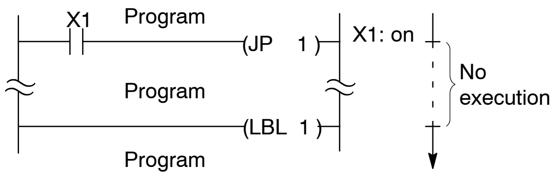
Outline Skips to the **LBL** instruction with the same number as the **JP** instruction.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11 ⋮ 20	ST X 1 JP 1 ⋮ LBL 1

Explanation of example

When the execution condition X1 turns on, the program skips from **JP1** to **LBL1**.



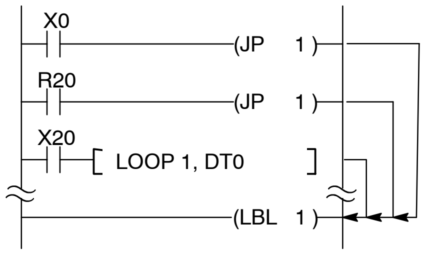
Description

When the execution condition turns on, the program jumps to the label (**LBL**) instruction that has the same number as the specified jump number.

The program then continues with the instructions starting from the address of the label that is the jump destination.

The same label is used for the **JP**, **LOOP** and **F19(SJP)** instructions. Any of these instructions can be used as the starting point for the jump destination.

Two or more **JP** instructions with the same number can be used in a program.



Two or more **LBL** instructions with same number cannot be specified in the same program.

If a label for the jump destination is not programmed, a syntax error will occur.

You must be careful when using one of the instructions below, which are executed by detecting the rise of a execution condition such as the differential instruction.

– **DF** (leading edge differential)
– Count input with **CT** (counter)
– Count input with **F118** (up/down counter)
– Shift input with **SR** (shift register)
– Shift input with **F119** (left/right shift register)
– **NSTP** (next step)
– Differential execution type high−level instruction (this instruction is specified by P and a number)

Precautions during programming

If the address of the **LBL** instruction precedes the address of the **JP** instruction, the scan will not terminate and an operation bottleneck error may occur.

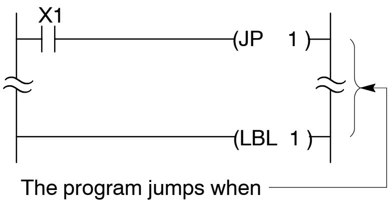
The **JP** instruction and **LBL** instruction cannot be used in the step ladder area (the area between **SSTP** and **STPE**).

You cannot perform a jump from a main program to a sub program (a subroutine program or interrupt program after the **ED** instruction), from a sub program to a main program, or from a sub program to another sub program.

TM, CT, and SR instruction operation between JP and LBL instructions

When the **LBL** instruction is located after the **JP** instruction:

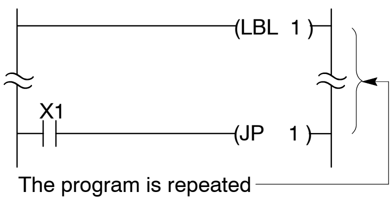
– **TM** instruction: The **TM** instruction is not executed. If it is not executed once during a single scan the correct time cannot be guaranteed.
– **CT** instruction: Even if the count input is on, counting is not performed. The elapsed value is preserved.
– **SR** instruction: Even if the shift input is on, no shift is performed. The contents of the specified register are preserved.



The program jumps when the execution condition turns on.

When the **LBL** instruction is located before the **JP** instruction:

– **TM** instruction: Because the **TM** will run several times during a single scan, the correct time cannot be guaranteed.
– **CT** instruction: If the state of the count input does not change during the scan, it will operate in the usual way.
– **SR** instruction: If the state of the shift input does not change during the scan, it will operate in the usual way.



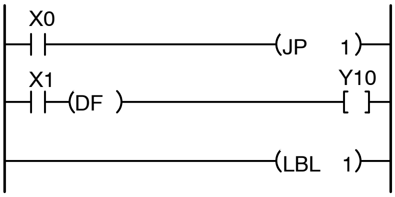
The program is repeated when the execution condition turns on.

☞ **Note**

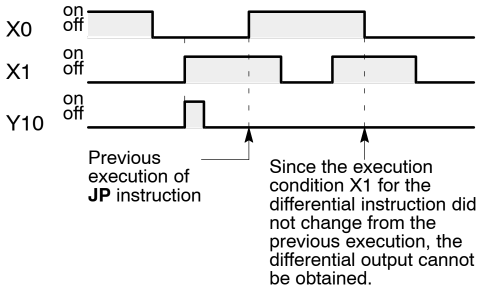
With the FP2SH and FP10SH, the time can be kept accurately even if these are executed multiple times during a single scan. To use these together with the JP instruction, change the setting of system register 4.

Differential instruction operation between JP and LBL instructions

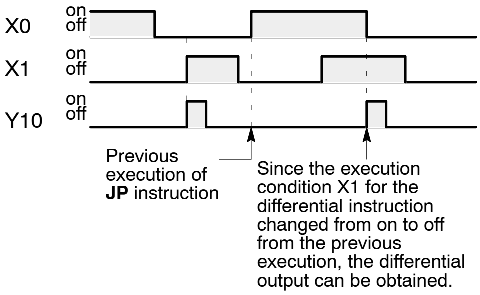
If a differential instruction is used in the area between a **JP** and **LBL** instruction, be aware that the output will differ as shown below depending on the execution condition of the **JP** and the input timing of differential instruction.



Time chart 1

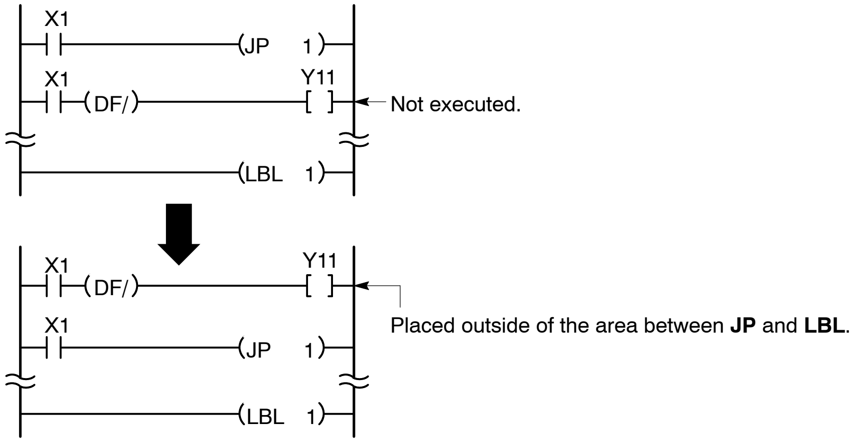


Time chart 2



When the execution condition for the **JP** instruction equals the execution condition for the differential instruction, the leading edge (or trailing edge) of the execution condition for the differential instruction will not be detected.

If the differential output is required, do not write the differential instruction between the **JP** and **LBL** instructions.

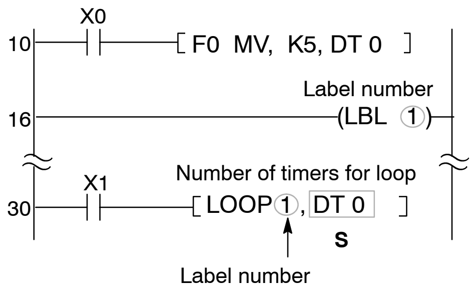


LOOP Loop
LBL Label

Outline Skips to the **LBL** instruction that has the same number as the **LOOP** instruction and executes what follows, repeatedly, until the data of a specified operand becomes “0”.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	X	0
11	F0		(MV)
	K		5
		DT	0
16	LBL		1
		⋮	
30	ST	X	1
31	LOOP		1
	DT		0

S	16-bit area for setting number of times for loop operation

Operands

Instruction	Relay				Timer/Counter		Register			Index register		Constant		Index modifier (*5)
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
Set value	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

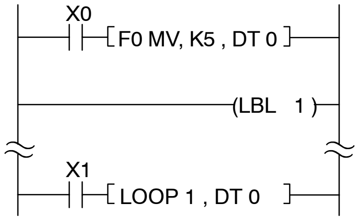
(*1) This cannot be used with the FP0/FP−e.
(*2) This cannot be used with the FP0/FP−e/FP0R/FPΣ/FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH, this is ID.
(*5) Only FP2, FP2SH and FP10SH label numbers can be used.

Description

When the execution condition (trigger) turns on, 1 is subtracted from the contents of S and if the result is other than 0, the program jumps to the label (**LBL** instruction) that has the same number as the specified number.

The program then continues with the instructions starting from the address of the label that is the loop destination.

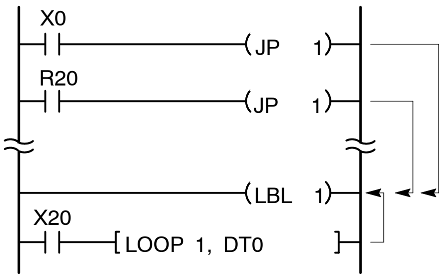
Set the number of times to execute the program with the **LOOP** instruction. When the number of times set in S (K constant) reaches 0, the jump will not occur even if the execution condition (trigger) turns on.



If the value of DT0 is K5, then after performing the jump 5 times, even if X1 is set to on, the jump operation is not executed.

If the contents of memory area specified by S is 0 from the beginning, the jump operation is not executed (it is ignored).

A label is common for the **JP** instruction, the **LOOP** instruction and the **F19 (SJP)** instruction. One can be used as the destination for all instructions as many times as required.



Two or more **LBL** instructions with the same number cannot be specified in a program.

With the FP2, FP2SH and FP10SH, index modification of the number specified by the **LOOP** instruction is possible.

If a label for the loop destination is not programmed, a syntax error will occur.

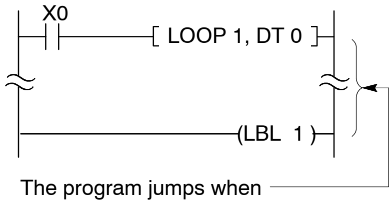
Flag conditions

- Error flag (R9007): Turns on and remains on when the specified value in the data area “S” becomes less than “0” (when the most significant bit (bit position 15) of the specified data area becomes “1”).
- Error flag (R9008): Turns on for an instant when the specified value in the data area becomes less than “0” (when the most significant bit (bit position 15) of the specified data area becomes “1”).

TM, CT, and SR instruction operation between the LOOP and LBL instructions

When the **LBL** instruction is located after the **LOOP** instruction:

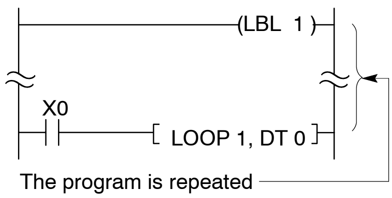
- **TM** instruction: The **TM** instruction is not executed. If it is not executed once during a single scan, the correct time cannot be guaranteed.
- **CT** instruction: Even if the count input is on, counting is not performed. The elapsed value is preserved.
- **SR** instruction: Even if the shift input is on, no shift is performed. The contents of the specified register are preserved.



The program jumps when the execution condition (trigger) turns on.

When the **LBL** instruction is located before the **LOOP** instruction:

- **TM** instruction: Because the timer will run several times during a single scan, the correct time cannot be guaranteed (see note).
- **CT** instruction: If the state of the count input does not change during the scan, it will operate in the usual way.
- **SR** instruction: If the state of the shift input does not change during the scan, it will operate in the usual way.



The program is repeated when the execution condition (trigger) turns on.

☞ **Note**

With the FP2SH and FP10SH, the time can be kept accurately even if these are executed multiple times during a single scan. To use these together with the LOOP instruction, change the setting of system register 4.

Precautions during programming

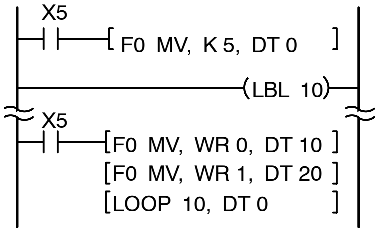
When the label is written in an address before the **LOOP** instruction, be careful of the following points.

Be sure to have the instruction that sets the number of loop cycles before the area between the **LBL** and **LOOP** instructions.

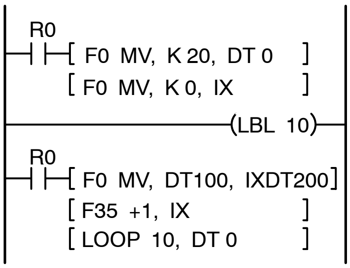
Set the instructions that will be repeated between **LBL** and **LOOP** so that they have the same trigger as the **LOOP** instruction.

During repeating, it is possible that one scan can exceed the surveillance time for operation jams and cause a operation bottleneck error.

Example 1: Execute 5 times of both F0 (MV) instructions when X5 is on.



Example 2: Send the value of DT100 to DT200 through DT219.



The **LOOP** instruction and **LBL** instruction cannot be used in the step ladder area (the area between **SSTP** and **STPE**).

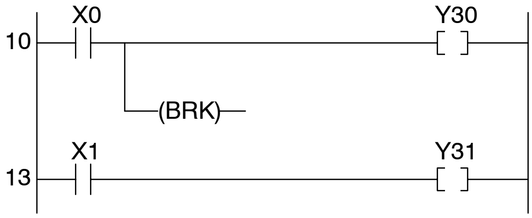
You cannot perform a jump from a main program to a sub-program (a subroutine program or interrupt program after the **ED** instruction), from a sub-program to a main program, or from a sub-program to another sub-program.

You must be careful when using one of the instructions below, which are executed by detecting the leading edge of execution condition (trigger) such as the differential instruction.
- **DF** (leading edge differential)
- Count input of **CT** (counter)
- Count input of **F118** (up/down counter)
- Shift input of **SR** (shift register)
- Shift input of **F119** (left/right shift register)
- **NSTP** (next step)
- Differential execution type high-level instruction (this instruction is specified by P and a number)

BRK Break

Outline Stops execution in TEST/RUN mode.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST X 0
	11	OT Y 30
	12	BRK
	13	ST X 1
	14	OT Y 31

Description

The **BRK** instruction is effective only in the TEST/RUN mode. In the normal RUN condition, this instruction is not executed.

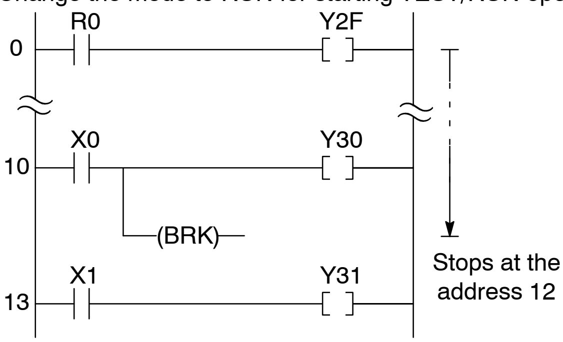
In the TEST/RUN mode, program execution is temporarily stopped with the address containing this **BRK** instruction.

The **BRK** instruction is used for checking the program by executing a part of the program.

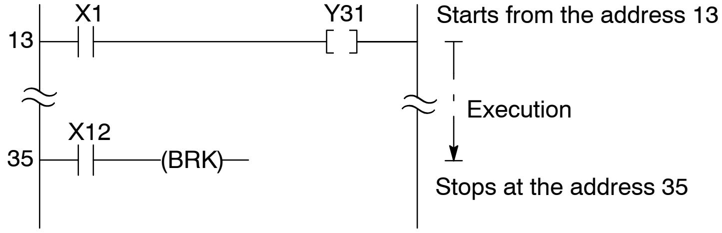
How to use the BRK instruction

Procedure:

1. Set the INITIALIZE/TEST switch of the CPU to the TEST side.
2. Select the modes for TEST/RUN operation by using programming tool software, as follows:
   – Output: DISABLE or ENABLE (select one according to your requirements)
   – **BRK**: VALID (in the **BRK** instruction valid mode)
   – Test mode: ConTI (in the continuous run mode)
3. Change the mode to RUN for starting TEST/RUN operation.



4. When X0 is in the on state, the **BRK** instruction is executed and program execution stops.
5. Press the “F3” key while holding down the “Shift” key in the MONITOR & TEST RUN window of the programming tool software to continue the program execution.
   If a **BRK** instruction is executed, program execution stops.



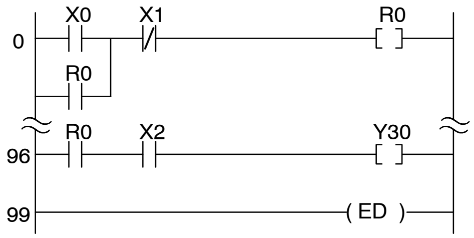
6. Up to the end of the program, proceed according to the operation in steps 4 and 5 above.
   If you want, change to the step operation mode, in which the program stops execution after execution of each instruction.

ED End

Outline Indicates the end of the ordinary program.

Program example

Ladder Diagram

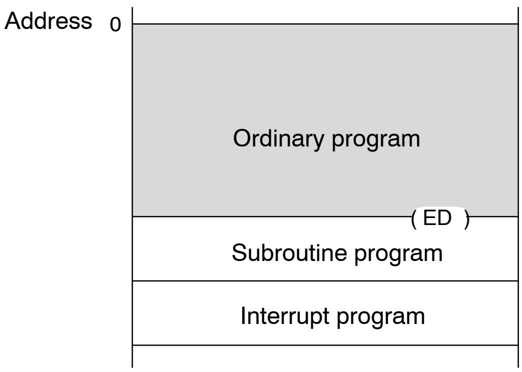


Boolean			
Address	Instruction		
0	ST	X	0
1	OR	R	0
2	AN/	X	1
3	OT	R	0
⋮	⋮		
96	ST	R	0
97	AN	X	2
98	OT	Y	30
99	ED		

Description

Indicates the end of the ordinary program.

Program area



Program areas are divided into an ordinary program area (main program) and “subroutine” and “interrupt program” areas (sub–programs) using this instruction.

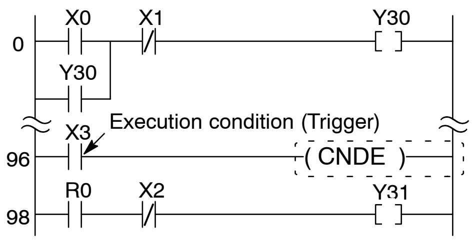
Enter subroutine programs and interrupt programs after the **ED** instruction.

CNDE Conditional end

Outline Ends one scan of the program when the execution condition (trigger) turns on.

Program example

Ladder Diagram



Boolean			
Address	**Instruction**		
0	ST	X	0
1	OR	Y	30
2	AN/	X	1
3	OT	Y	30
⋮	⋮		
96	ST	X	3
97	CNDE		
98	ST	R	0
99	AN/	X	2
100	OT	Y	31

Description

The **CNDE** instruction enables you to end one scan of the program.

When the execution condition (trigger) turns on, the program finishes and the input, output, and other such operations are performed. When the operations are completed, the program then returns to the starting address.

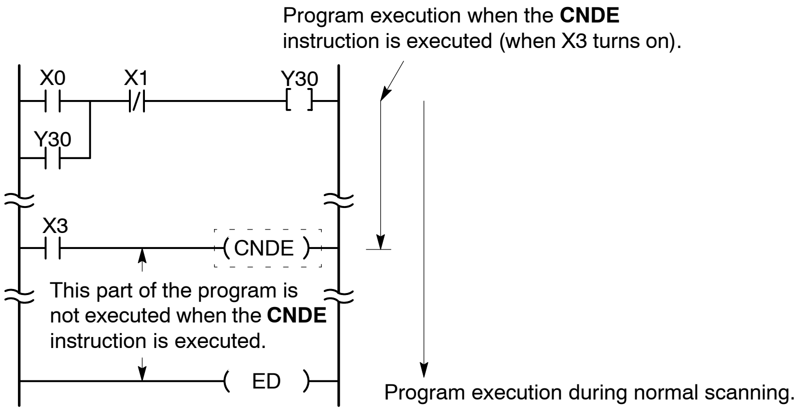
You can adjust the timing that operations are performed by performing the operations only after a required number of program scans are completed.

The **CNDE** instruction cannot be performed in sub−programs such as subroutine programs or interrupt programs. Use the **CNDE** instruction in the main program area only.

Two or more **CNDE** instructions can be used within the main program.

You must be careful when using one of the instructions below, which are executed by detecting the leading edge of a execution condition (trigger) such as the differential instruction.

- **DF** (leading edge differential)
- Count input for **CT** (counter)
- Count input for **F118 (UDC)** (up/down counter)
- Shift input for **SR** (shift register)
- Shift input for **F119 (LRSR)** (left/right shift register)
- **NSTP** (next step)
- Differential execution type high−level instruction (this instruction is specified by P and a number)

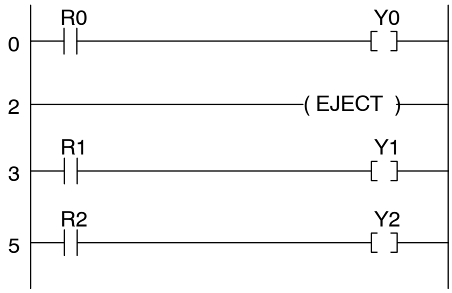

Program execution when the CNDE instruction is executed (when X3 turns on).
X0
X1
Y30
Y30
X3
CNDE
This part of the program is not executed when the CNDE instruction is executed.
ED
Program execution during normal scanning.

EJECT Eject

Outline Adds page break for use when printing.

Program example

Ladder Diagram



Boolean			
Address	Instruction		
0	ST	R	0
1	OT	Y	0
2	EJECT		
3	ST	R	1
4	OT	Y	1
5	ST	R	2
6	OT	Y	2

Explanation of example

Insert the EJECT instruction in the address where you want the page to break when printing out the program you created. In the above, the page will break at address 2.

Description

When printing out the program created with the software tool, the page will break at the position where this instruction is inserted.
As with the NOP instruction, processes in the program will not be affected.

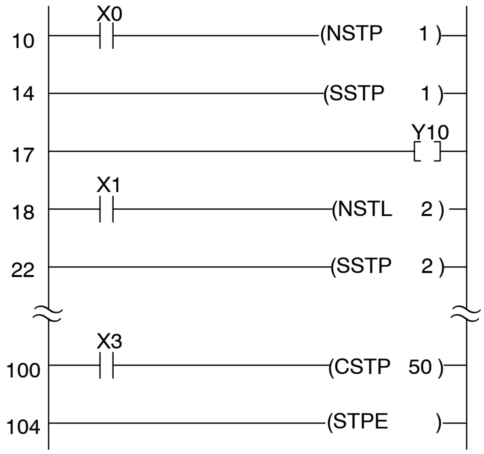
SSTP Start step
NSTL Next step (scan execution type)
NSTP Next step (pulse execution type)
CSTP Clear step
STPE Step end

Outline

SSTP: Indicates the start of a step ladder process.
NSTL: Opens a step ladder process.
NSTL is executed every scan if its trigger is on.
NSTP: Opens a step ladder process.
NSTP is executed when the leading edge of its trigger is detected.
CSTP: Resets the specified process.
STPE: Indicates the end of step ladder area.

Program example

Ladder Diagram



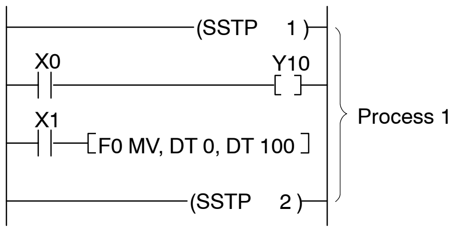
Boolean			
Address	**Instruction**		
10	ST	X	0
11	NSTP		1
14	SSTP		1
17	OT	Y	10
18	ST	X	1
19	NSTL		2
22	SSTP		2
⋮	⋮		
100	ST	X	3
101	CSTP		50
104	STPE		

Description

When the **NSTL** instruction or the **NSTP** instruction is executed, the process starting with the **SSTP** instruction of the specified number is started and executed.

In a step ladder program, a process is identified as being from one **SSTP** instruction to the next **SSTP** or **STPE** instruction.

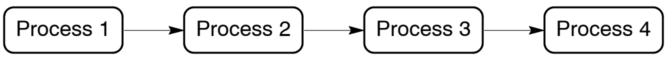
Example:



Operations such as the sequence control, selection branch control, parallel branch control are easily executed.

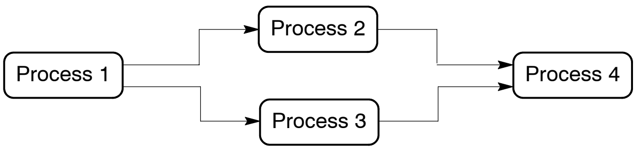
– Sequence control

Only the necessary processes are switched and executed in order.



– Selection branch control

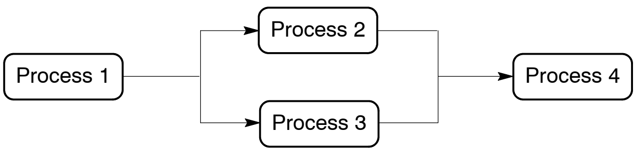
The processes are selected and executed according to conditions.



– Parallel branch merge control

Multiple processes are executed simultaneously.

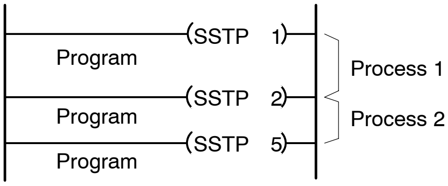
After each process is completed, the next process is executed.



Syntax of step ladder instruction

SSTP (start step) instruction:

This instruction indicates the start of a process n.



In a step ladder program, a process n is identified as being from one **SSTPn** instruction to the next **SSTP** or **STPE** instruction.

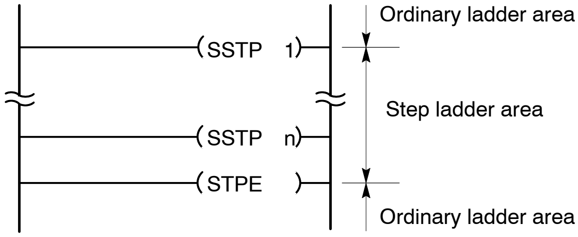
No two processes can have the same process number.

The **OT** instruction can be programmed at the address just after the **SSTP** instruction.

The **SSTP** instruction cannot be programmed in sub–program (subroutine or interrupt program area).

The area starting from the first **SSTP** instruction to the **STPE** instruction is referred to as the step ladder area. The programs in this area are all controlled as processes.

Other areas are referred to as ordinary ladder areas.

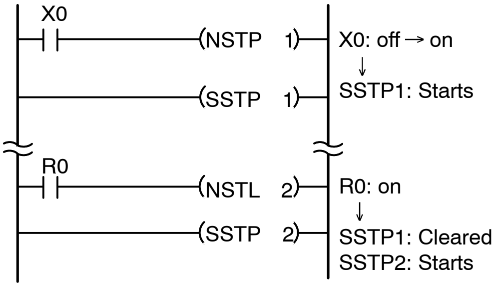


There is a special internal relay that turns on for one scan only when a process on the step ladder starts. (R9015: step ladder initial pulse relay.) This relay is used to perform operations for only one scan for counter reset or other such process starts.

NSTL (Next step, scan execution type) instruction:
NSTP (Next step, differential (pulse) execution type) instruction:

When an **NSTPn** or **NSTLn** instruction is executed, the process with the same process number “n” as the **NSTP** or **NSTL** instruction is opened.

The execution condition (trigger) for the next step instruction means the execution condition (trigger) to start the process.

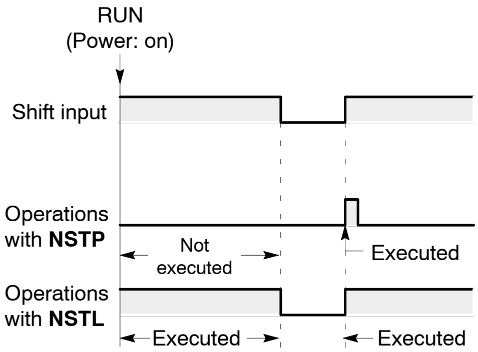


Declare the first process to start in the next step instruction in the ordinary ladder area.

A process can be started from the ordinary ladder area or from an already started process.

However, when you start a process with a next step instruction from within a process, the process that is operating and contains the next step instruction is automatically cleared and the specified process starts. Be aware that the outputs and other processes are actually turned off by the clear operation during the next scan.

The **NSTP** instruction is a differential (pulse) execution type instruction, so it is executed for only one time when the execution condition (trigger) turns on. Also, since it only detects if the execution condition (trigger) has changed between on and off, when switched to the RUN mode or when the power supply is turned on while in the RUN mode and the execution condition (trigger) is already on, the instruction is not executed.



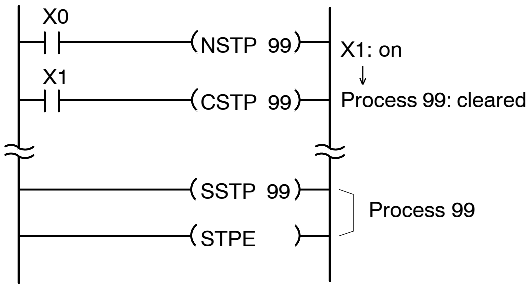
When you use the **NSTP** instruction with one of the following instructions that changes the order of the execution of instructions, be aware that the operation of the instructions will differ depending on the timing of their execution and their triggers.

- **MC** to **MCE** instructions
- **JP** to **LBL** instructions
- **F19 (SJP)** –**LBL** instructions
- **LOOP** to **LBL** instructions
- **CNDE** instruction
- Step ladder instructions
- Subroutine instruction

When combining the **NSTP** instruction with an **ANS** or **POPS** instruction, be careful that the programming is correct.

CSTP (clear step) instruction:

When a **CSTP** instruction is executed, the process “n” with the same process number “n” is cleared. This instruction can be used to clear the final process or to clear the processes when the parallel branch merge control is executed.



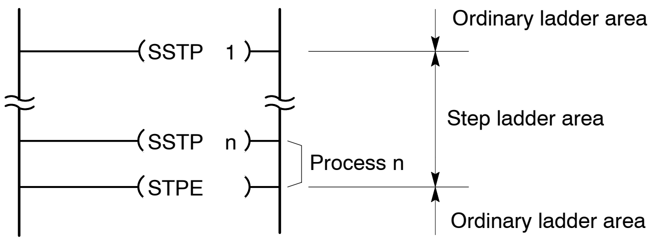
A process can also be cleared from the ordinary ladder area or from a process that is already started.

Note

With the FPΣ, FP−X, FP0R, FP2, FP2SH and FP10SH, the SCLR (Block Clear) instruction used to clear multiple processes of specified range at one time. Refer to “SCLR” instruction.

STPE (step end) instruction:

The **STPE** instruction indicates the end of the step ladder area. Be sure to write this instruction at the end of the last process. Thus, the final process of the step ladder is from **SSTP** to **STPE**.



In the above situation, process n is the last process.

The **STPE** instruction is used only once in the main program. (This instruction cannot be programmed in sub−program such as a subroutine program or interrupt program.)

Precautions during programming

You do not have to program processes in the order of process numbers.

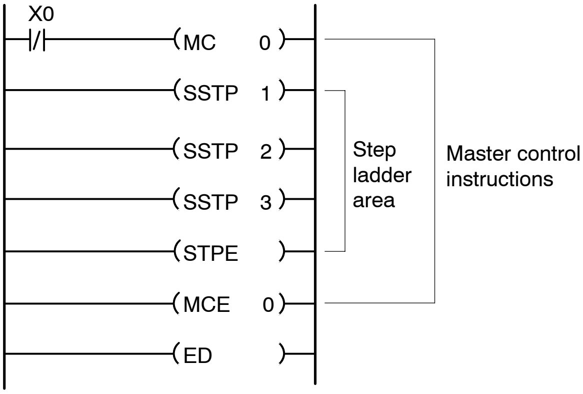
In the step ladder area, you cannot use the following instructions:

– Jump instructions (**JP** and **LBL**)
– Loop instructions (**LOOP** and **LBL**)
– Master control instructions (**MC** and **MCE**)
– Subroutine instructions (**SUB** and **RET**) (*)
– Interrupt instructions (**INT** and **IRET**)
– **ED** instruction
– **CNDE** instruction

(*): The **CALL** instruction can be used within the step ladder area.

When you need to clear an entire processes in step ladder program, use the master control (**MC** and **MCE**) instructions as shown below.

Example: All processes are cleared when X0 becomes on.

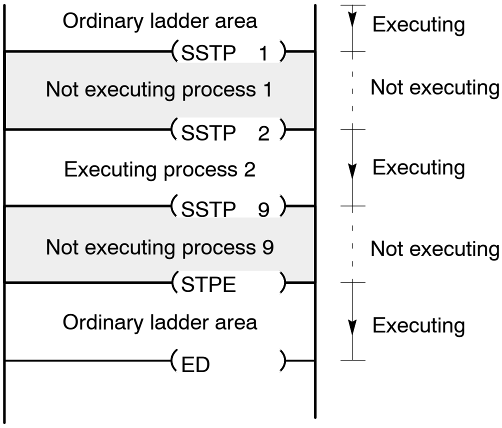


It is not necessary to execute processes in order of process numbers. You can execute two or more processes at the same time.

Once you force on or off an output that is programmed in a process not yet executed, the output condition is maintained until the process starts even if the forced on and off operation is canceled.

Step ladder operations

When the step ladder processes are programmed, program execution proceeds in the ordinary ladder area and in the processes triggered by the next step instructions (**NSTL** or **NSTP**). Processes that have not been triggered are ignored.



In the diagram, program execution occurs in the ordinary ladder areas and in process 2.

The moment the step ladder process is opened, step ladder internal pulse relay R9015 turns on for an instant only in the first scan of the process. You can use R9015 to reset a counter or shift register used in the opened process.

The execution state (start/stop) for processes are stored in special data registers:

Type	Special data register
FP0 C10, C14, C16, C32/ FP−e	DT9060 to DT9067
FP0 T32/FP0R	DT90060 to DT90067
FPΣ/FP−X/FP2/FP2SH/ FP10SH	DT90060 to DT90122

Example: The start−up conditions for processes No. 16 through No. 31

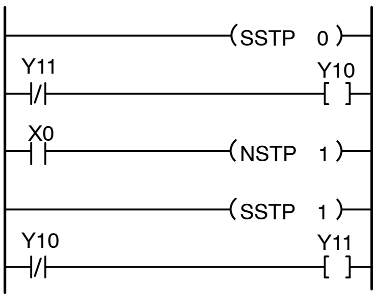
Bit position	15 · · 12				11 · · 8				7 · · 4				3 · · 0			
Process number	31 · · 28				27 · · 24				23 · · 20				19 · · 16			
DT9061/DT90061	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

When bit position 8 of DT9061/DT90061 is "1", the process no. 24 is starting.

Notes on process clear

If the next step instruction is executed in an active process, that process is automatically cleared. However, the actual clear operation does not occur until the next scan. Therefore, for one scan during the process transition, there will be two simultaneously active processes. If you do not want them to be on at the same time, program with an interlock circuit. If there is a possibility of processes being simultaneously on because of hardware response delays, adapt hardware processing to take the response delay into account.

Example:



If the process is cleared, the instructions in that process operate as follows.

Instruction	Operation status
OT	All off
KP	Holds the state.
SET	Holds the state.
RST	Holds the state.
TM	Reset the elapsed value and timer contact output.
CT	Holds the state at the time just before the trigger turns off.
SR	Holds the state at the time just before the trigger turns off.
DF and DF/ (*)	Remembers the state of execution condition (trigger).
Other instructions	Not executed.

(*): Same operation as when the execution condition (trigger) for the **MC** instruction turns off. Refer to the explanation of the **MC** and **MCE** instructions.

You must be careful when using one of the instructions below, which are executed by detecting the leading edge of execution condition (trigger) such as the differential instruction.

- – **DF** (leading edge differential)
- – Count input of **CT** (counter)
- – Count input of **F118 (UDC)** (up/down counter)
- – Shift input of **SR** (shift register)
- – Shift input of **F119 (LRSR)** (left/right shift register)
- – **NSTP** (next step)
- – Differential execution type high−level instruction (this instruction is specified by P and a number)

Examples of step ladder instructions

① Sequence control of a process

This program repeats the same process until the work in a particular process is completed, then switches to be the next process as soon as the work is completed.

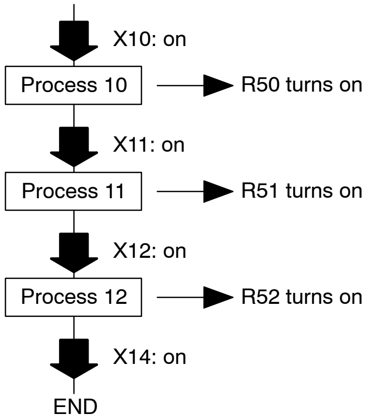
Program an **NSTL** instruction to trigger the next process in each process. When the **NSTL** instruction is executed, the next process is activated, and the currently executing process is cleared.

It is not necessary to execute in order of process number. You can also program the **NSTL** instruction to trigger a previous process in response to current conditions.

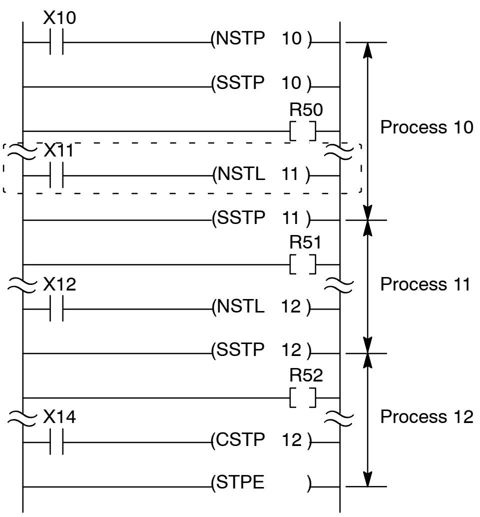
Program example

1) When the X10 turns on, process 10 is executed.
2) When X11 turns on, process 10 is cleared and process 11 is executed.
3) When X12 turns on, process 11 is cleared and process 12 is executed.
4) When X14 turns on, process 12 is cleared and step ladder operation finishes.

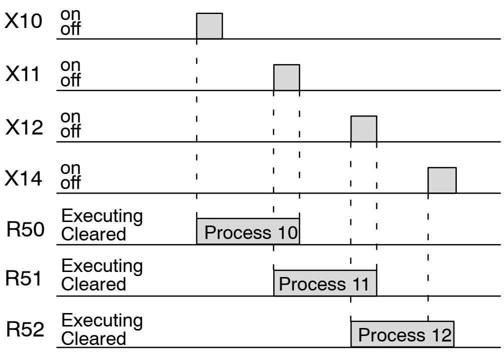
Process flowchart



Program



Time chart



② Selection branch control of a process

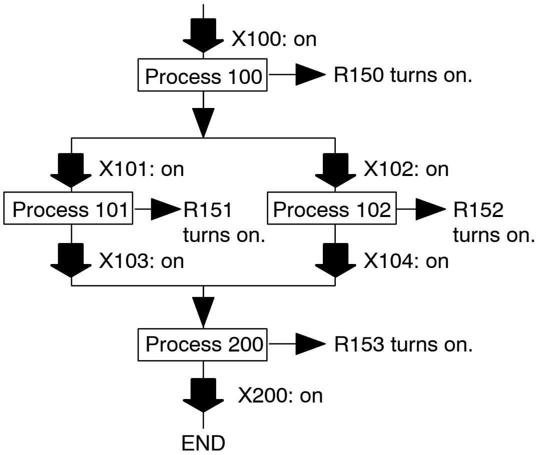
This program selects and switches to the next process according to the actions and results of a particular process. Each process loops until its work is completed.

Program two or more **NSTL** instructions to trigger the next process in a process. Depending on the execution conditions, the next process is selected, triggered and program execution is transferred.

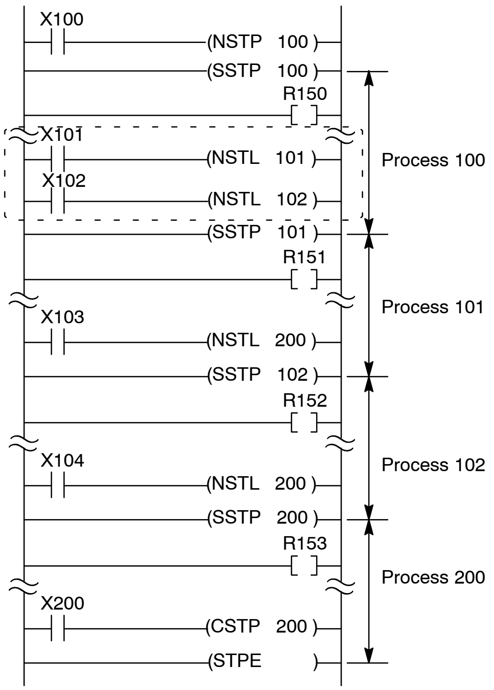
Program example

1) When X100 turns on, process 100 is executed.
2) When X101 turns on in process 100, process 101 is executed. Or when X102 turns on in process 100, process 102 is executed.
3) When X103 turns on in process 101, process 101 is cleared and process 200 is executed. When X104 turns on in process 102, process 102 is cleared and process 200 is executed.
4) When X200 turns on, process 200 is cleared and step ladder operation finishes.

Process flowchart

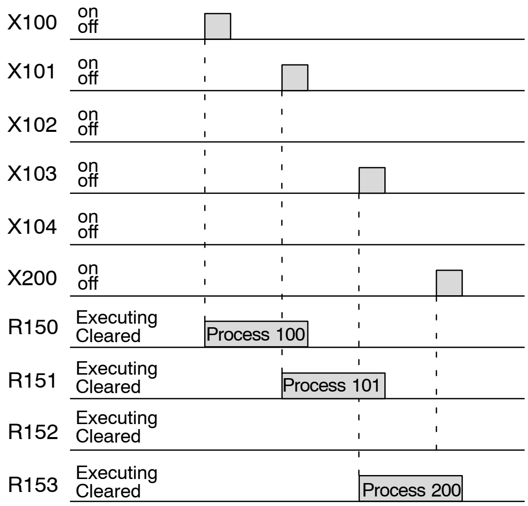


Program



Time chart

When X101 turns on



③ Parallel branch merge control of a process

This program triggers multiple processes simultaneously. After each of the branch processes has completed its work, they merge again before transferring execution to the next process.

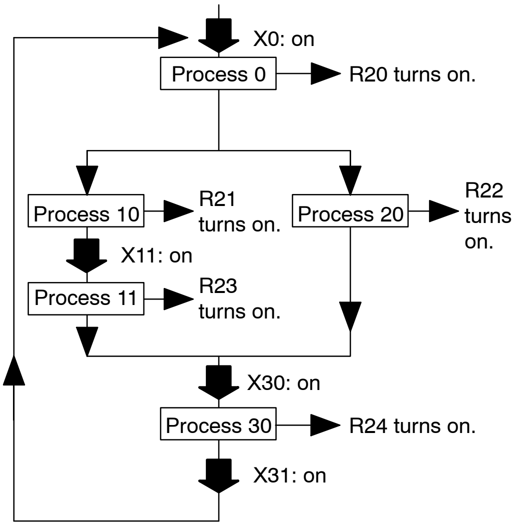
Program multiple **NSTL** instructions for one trigger in a process.

To merge processes, include a flag indicating the state of the other processes in the transfer condition for the next process.
When they merge and execute the next process, clear all uncleared processes at the same time.

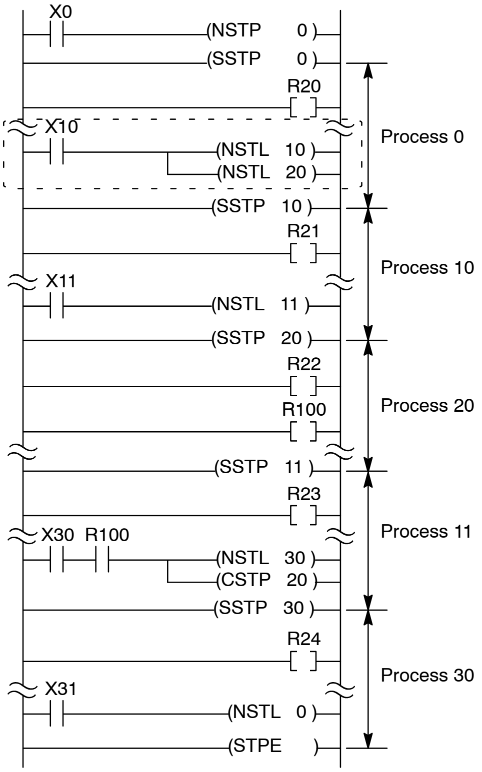
Program example

1) When X0 turns on, process 0 is executed.
2) When X10 turns on, process 0 is cleared and process 10 and process 20 are executed simultaneously (Parallel branch control).
3) When X11 turns on, process 10 is cleared and process 11 is executed.
4) When X30 turns on, process 11 and process 20 are cleared and process 30 is activated. (Merge control)
   - Clear process 20 with the clear instruction.
   - Clear process 11 and execute process 30.
5) When X31 turns on, process 30 is cleared and initial process 0 is executed again.

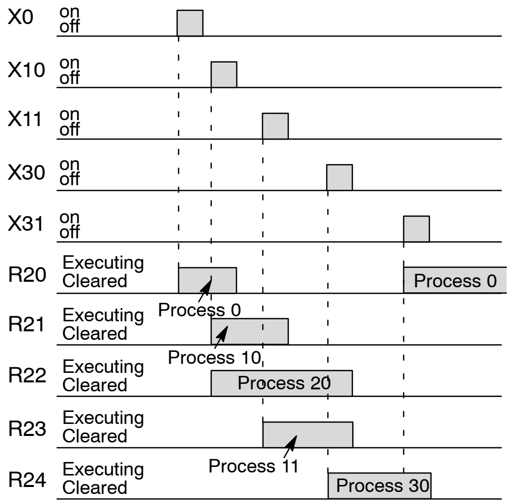
Process flowchart



Program



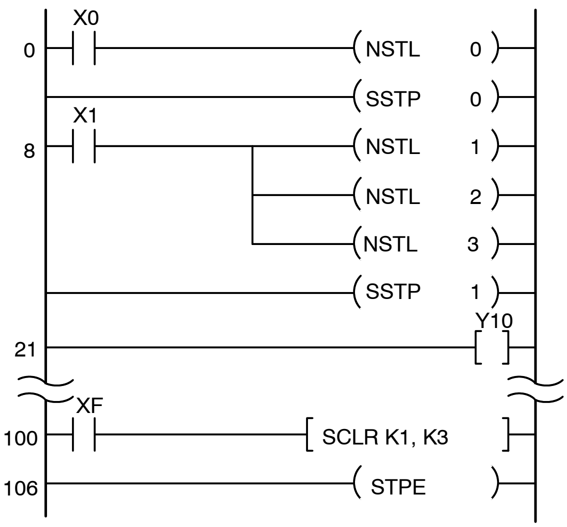
Time chart



SCLR Clear multiple processes

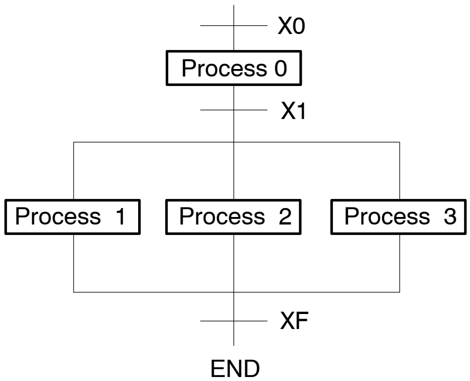
Outline Reset multiple processes specified by n1 and n2.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	0	ST X 0
	1	NSTL 0
	4	SSTP 0
	8	ST X 1
	9	NSTL 1
	12	NSTL 2
	15	NSTL 3
	18	SSTP 1
	21	OT Y 10
	:	:
	100	ST X F
	101	SCLR
		K 1
		K 3
	106	STPE

Explanation of example

When input XF goes on, processes in operation from 1 through 3 are cleared.



Description

When an **SCLR** instruction is executed, all processes in operation from process n1 through process n2 are cleared.

Precautions during programming

Set so that n1 is greater than or equal to n2 (n1 ≧ n2).

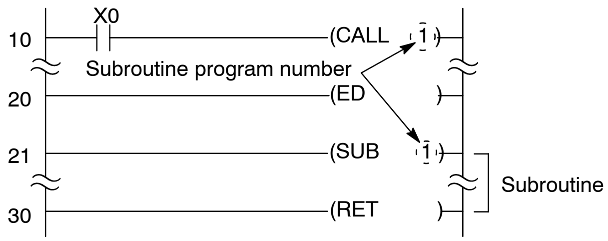
The **SCLR** instruction can be executed from both normal ladder areas and operating processes.

CALL Subroutine call
SUB Subroutine entry
RET Subroutine return

Outline

CALL: Executes the specified subroutine program.
SUB: Indicates the start of the subroutine program.
RET: Indicates the end of the subroutine program.

Program example

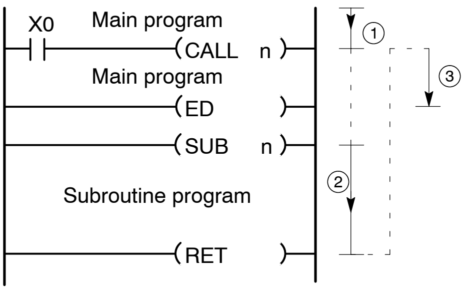


Ladder Diagram	Boolean Address	Boolean Instruction		
	10	ST	X	0
	11	CALL		1
	⋮		⋮	
	20	ED		
	21	SUB		1
	⋮		⋮	
	30	RET		

Description

When the execution condition (trigger) turns on, the **CALL** instruction is executed and the subroutine program of the specified number is executed starting with the **SUB** instruction.

When the subroutine reaches the **RET** instruction, the program returns to the address after the **CALL** instruction of the main program and the execution of the main program resumes.



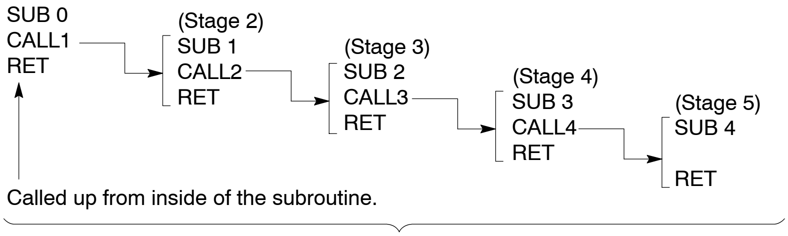
When **CALLn** is executed, the program is executed in the order ①, ②, ③ shown above.

Subroutine Program Syntax

The subroutine program n is the program from the **SUBn** instruction to the **RET** instruction. Always place the address (subroutine) after the **ED** instruction.

The **CALL** instruction can be programmed in the main program area, interrupt program area, or subroutine program area.
Two or more **CALL** instructions with the same program number can be specified in a program.

Nesting of subroutines is possible until the 5th nesting.



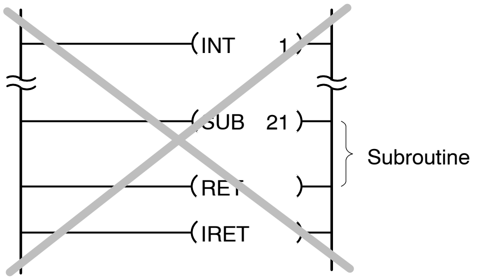
5th nesting example

Flag conditions

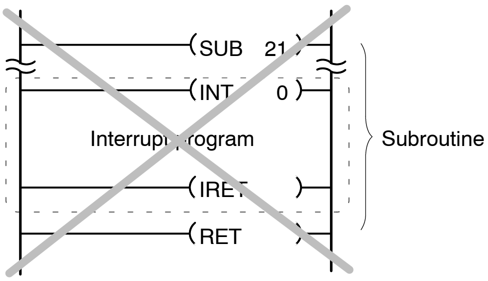
- Error flag (R9007): Turns on and stays on when performing five nestings and executing the **CALL** instruction for the subroutine of the 5th nesting.
- Error flag (R9008): Turns on for an instant when performing five nestings and executing the **CALL** instruction for the subroutine of the 5th nesting.

Precautions during programming

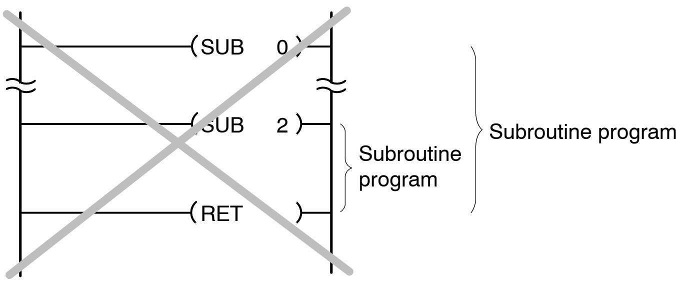
In the interrupt program, a subroutine program cannot be used.



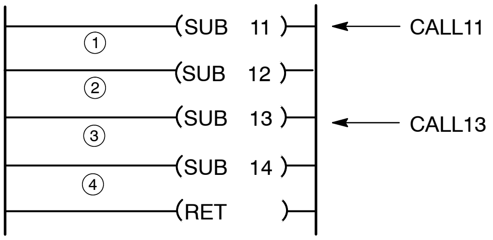
In the subroutine program, an interrupt program cannot be used.



For the FP0/FP−e, subroutine program cannot be written inside another subroutine program.



For the FP2/FP2SH/FP10SH, subroutine programs may be constructed with multiple entrances and only one exit.



When “CALL 11” is executed, ① to ④ are executed.
When “CALL 13” is executed, ③ and ④ are executed.

You must be careful when you use, in a subroutine, one of the instructions below that is executed by detecting the leading of execution condition (trigger) such as the differential instruction.

– **DF** (leading edge differential)
– Count input of **CT** (counter)
– Count input of **F118** (up/down counter)
– Shift input of **SR** (shift register)
– Shift input of **F119** (left/right shift register)
– **NSTP** (next step)
– Differential execution type high–level instruction (this instruction is specified by P and a number)

When the CALL instruction execution condition (trigger) is off

If the execution condition (trigger) for the **CALL** instruction is in the off state, the subroutine program is not executed. (This is the same for **CALL** instructions within master controls or step ladders.) When the execution condition (trigger) for the **CALL** instruction is in the off state, the instructions in the subroutine operate as follows.

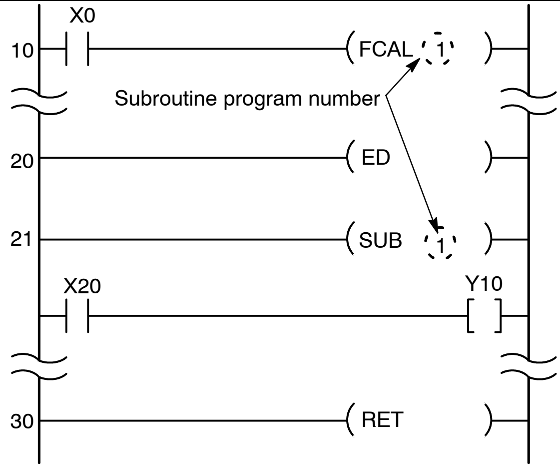
Instruction	Operation status
OT	Holds the state.
KP	Holds the state.
SET	Holds the state.
RST	Holds the state.
TM	Does not perform any timing. If timing is not performed once per scan, the correct time cannot be guaranteed.
CT	Holds the elapsed value.
SR	Holds the elapsed value.
DF and DF/	Same as when a differential instruction is used between **MC** and **MCE** instructions. See page 2–57.
Other instructions	Not executed.

FCAL Output off type subroutine call

Outline Executes the specified subroutine. When returning to the main program, all outputs in the subroutine program are set to off.

Program example

Ladder Diagram



Boolean			
Address	**Instruction**		
10	ST	X	0
11	FCAL		1
:		:	
20	ED		
21	SUB		1
23	ST	X	20
24	OT	Y	10
:		:	
30	RET	Y	10

Description

Operation and syntax are the same as normal subroutine call instructions. However, the following points are different.

If the trigger for the **CALL** instruction is in the off state, the subroutine program is not executed. When the trigger for the **CALL** instruction is in the off state, the instructions in the subroutine program operate as follows.

Instruction	Operation status
OT	All off (differs from a normal subroutine instruction)
KP	Holds the state.
SET	Holds the state.
RST	Holds the state.
TM	Resets (differs from a normal subroutine instruction)
CT	Holds the state at the time just before the trigger turns off.
SR	Holds the state at the time just before the trigger turns off.
DF and DF/	Same operation as when the execution condition (trigger) for the **MC** instruction turns off. Refer to the explanation of the **MC** and **MCE** instructions.
Other instructions	Not executed.

Precautions during programming

Like a **CALL** instruction, up to five nesting levels are possible. However, it will not be possible to use certain MC numbers depending on the number of nesting levels as shown below.

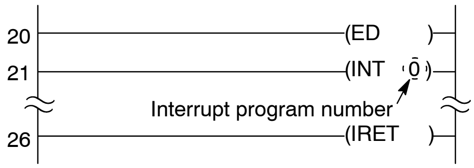
Calls from other than subroutines	MC255
2nd	MC255 to 254
3rd	MC255 to 253
4th	MC255 to 252
5th	MC255 to 251

Availability
FP0/FP0R/FP−e/ FPΣ/FP−X

INT Interrupt
IRET Interrupt return

Outline **INT:** Indicates the start of the interrupt program.
IRET: Indicates the end of the interrupt program.

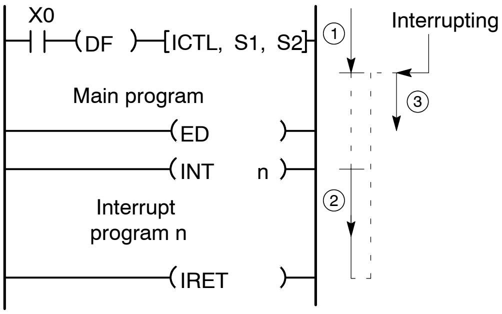
Program example

Ladder Diagram	Boolean	
	Address	Instruction
	20 21 ⋮ 26	ED INT 0 ⋮ IRET

Description

When an interrupt is input, the interrupt program of the number specified is executed starting from the **INT** instruction.

When the interrupt program reaches the **IRET** instruction, the program returns to the address where the interrupt occurred and the main program resumes.



When the interrupt occurs, the program is executed in the order ①–②–③ shown above.

Syntax of interrupt program

An interrupt program n is the program between the **INTn** instruction and the **IRET** instruction. The interrupt program must always be placed after the **ED** instruction.

The number of the interrupt program is decided by the type of the interrupt.

Interrupt Program No.	Interrupt input			
	FP0/FP−e	FPΣ/FP0R	FP−X Ry	FP−X Tr
INT0	X0	X0	X0	X0
INT1	X1	X1	X1	X1
INT2	X2	X2	X2	X2
INT3	X3	X3	X3	X3
INT4	X4	X4	X4	X4
INT5	X5	X5	X5	X5
INT6	—	X6	X6	X6
INT7	—	X7	X7	X7
INT8	—	—	X100	—
INT9	—	—	X101	—
INT10	—	—	X102	—
INT11	—	—	X200	—
INT12	—	—	X201	—
INT13	—	—	X202	—
INT24	Periodical interrupt			

Interrupt Program No.	High−speed counter−initiated interrupt				
	FP0/FP−e	FP0R	FPΣ	FP−X Ry	FP−X Tr
INT0	ch0	ch0	ch0	ch0	ch0
INT1	ch1	ch1	ch1	ch1	ch1
INT2	—	—	—	ch2	ch2
INT3	ch2	ch2	ch2	ch3	ch3
INT4	ch3	ch3	ch3	ch4	ch4
INT5	—	—	—	ch5	ch5
INT6	—	ch4	—	ch6	ch6
INT7	—	ch5	—	ch7	ch7
INT8	—	PLS−ch0	—	ch8	—
INT9	—	PLS−ch1	—	ch9	—
INT10	—	PLS−ch2	—	—	—
INT11	—	PLS−ch3	—	chA	—
INT12	—	—	—	chB	—
INT13	—	—	—	—	—

Note) When using the high−speed counter−initiated interrupt program, the counting performance of the high−speed counter may be decrement at the moment of the start−up of the interrupt program.

Note) Only for the PLS−ch* of FP0R, it is the target value match interrupt of pulse output.

Before inputting an interrupt program

① Declare the contact point to be used as the interrupt input (trigger).
Select the contact point to be used as the interrupt input (trigger) and indicate it at system register 403.

Notes

- **If the high−speed counter/pulse catch is set, that contact cannot be used as the interrupt input (trigger).**
- **For the high speed counter−initiated interrupts and periodical interrupts, it is not necessary to indicate the input (trigger) contact.**

② Enable the execution of interrupt programs.
The default conditions are set with interrupt programs disabled. Enable the execution of interrupt programs with the **ICTL** instruction.

Precaution when rewriting in RUN mode (for FP0/FP0R/FP−e/FPΣ/FP−X)

If the program is rewritten in the RUN mode, execution will be inhibited for all interrupt programs, and will have to be enabled again after the rewriting has been completed in the RUN mode.

Use the R9034 (rewriting done flag in RUN) to enable the interrupt programs again automatically using the ladder program. The R9034 is a special relay that turns on for only I scan after the completion of the rewriting in the RUN mode.

Interrupt program execution

There are three types of interrupt.

① **Interrupt from the input contact**

The interrupt occurs when the input specified at system register 403.

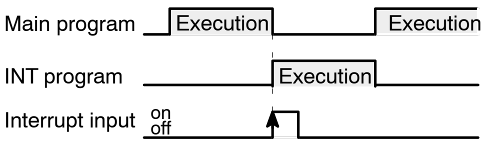
② **High−speed counter−initiated interrupt**

When executing the high−speed counter instruction, the interrupt occurs when the high−speed counter elapsed value equals the set target value.

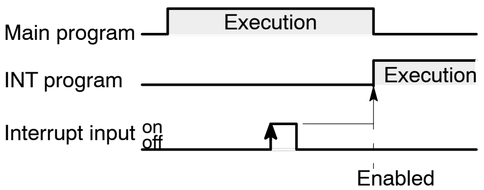
③ **Periodical interrupt (INT24)**

The interrupt occurs in fixed time intervals. Set the time interval with the **ICTL** instruction.

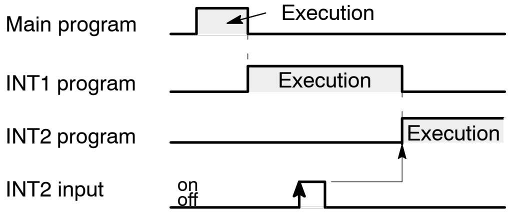
When the interrupt occurs, the interrupt program with the corresponding number is executed.



If interrupts are disabled, an interrupt will occur only at the point when interrupts are enabled with the **ICTL** instruction.



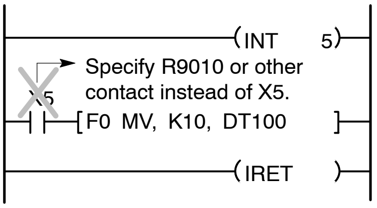
When another interrupt program is being executed, an interrupt will occur after the current program is completed.



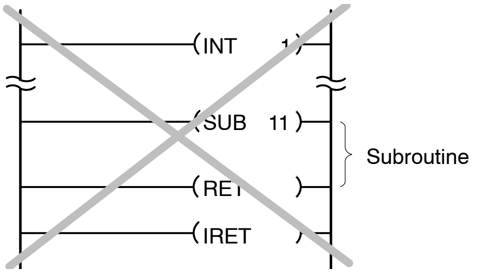
Precautions during programming for all types

If either the **INT** instruction or **IRET** instruction is missing, a syntax error will result.

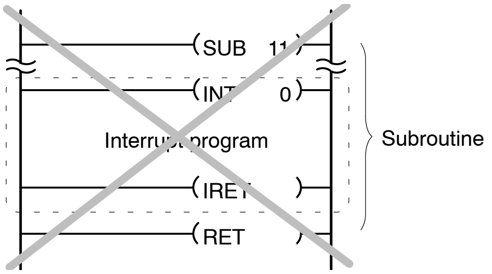
When an interrupt is issued, the operation memory corresponding to the interrupt input contact does not undergo I/O refreshing. Therefore, contacts other than the interrupt input contact, such as the constantly−on relay R9010, should be specified by the input conditions in the interrupt program.



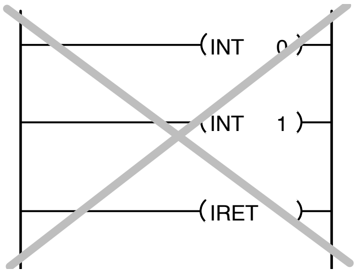
A subroutine program cannot be used in an interrupt program.



An interrupt program cannot be used in a subroutine program.



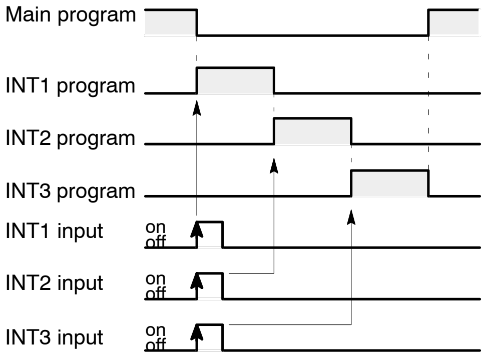
Interrupt program cannot be programmed into another interrupt program.



Control when more than one interrupt occurs simultaneously.

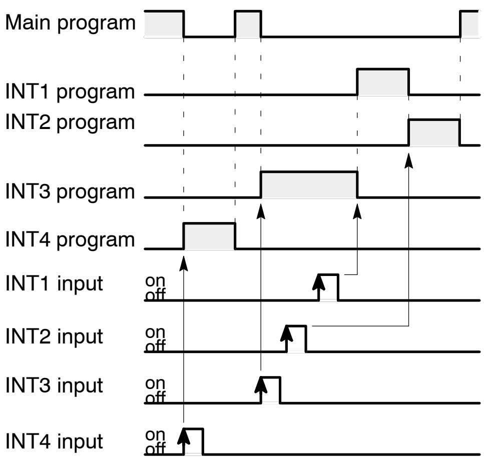
When more than one interrupt occurs simultaneously, the interrupt program with the smaller number is executed first. The other interrupt programs are then placed in the execution waiting state. After the first interrupt program is completed, the other programs will be executed in order from the smallest number to the greatest.

Example:



When more than one interrupt occurs while a interrupt program is being executed, the other programs will be executed in order from the smallest number to the greatest after the program currently being executed is finished.

Example:

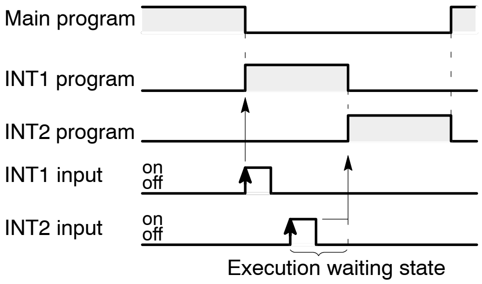


During execution of the INT3 program in the example above, the INT2 input occurred before INT1. But, after INT3 program is done, the INT1 program is executed first and then INT2 is executed.

Interrupt program execution waiting state and clearing

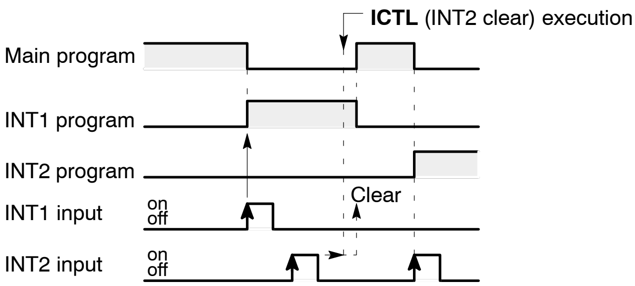
When multiple interrupt programs occur simultaneously or new interrupt programs occur during the execution of another interrupt program, the interrupt programs of lower preference are placed in the execution waiting state. They are then executed in order of preference when the other interrupt programs are completed.

Example:



If placed in the execution waiting state, there is a time difference between the time of when the interrupt occurs and when the interrupt program is actually executed. If you do not want to execute the interrupt program placed in the execution waiting state because of this, it can be cleared with the **ICTL** instruction. An interrupt program that is cleared will not be executed.

Example:



When the execution of interrupt programs is disabled with the **ICTL** instruction, interrupts that occur are still placed in the execution waiting state. When the execution is enabled with the **ICTL** instruction, the waiting interrupt programs will then be executed. Programs in the execution waiting state can be cleared with the **ICTL** instruction.

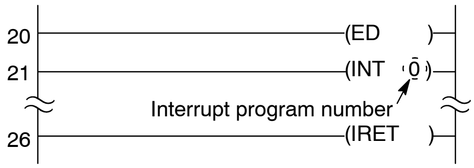
Availability
FP2/FP2SH/FP10SH

INT Interrupt
IRET Interrupt return

Outline

INT: Indicates the start of the interrupt program.
IRET: Indicates the end of the interrupt program.

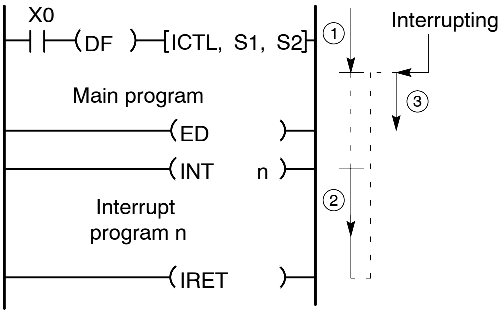
Program example

Ladder Diagram	Boolean	
	Address	Instruction
	20 21 ⋮ 26	ED INT 0 ⋮ IRET

Description

When an interrupt is input, the interrupt program of the number specified is executed starting from the **INT** instruction.

When the interrupt program reaches the **IRET** instruction, the program returns to the address where the interrupt occurred and the main program resumes.



When the interrupt occurs, the program is executed in the order ①–②–③ shown above.
In the default mode, all interrupt programs are disabled and cannot be executed. The **ICTL** instruction should be used to enable execution of an interrupt program.

Syntax of interrupt program

An interrupt program n (n: 0 to 24) is the program between the **INTn** instruction and the **IRET** instruction. The interrupt program must always be placed after the **ED** instruction. Up to 25 programs can be written.

The number of the interrupt program is decided by the type of the interrupt.
INT0 to INT15: Interrupts from the interrupt unit
INT16 to INT23: Interrupts from an intelligent unit that issues interrupts
INT24: Periodic interrupt

Interrupt program execution

There are three types of interrupt.

① **Interrupts from a interrupt unit (corresponding to INT0 to INT15)**

Interrupts are issued in response to the rise or fall of the interrupt unit input (whether rising or falling is specified on the unit side).

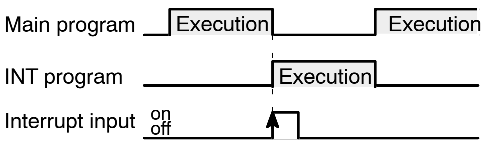
② **Interrupts from an intelligent unit that issues interrupts (corresponding to INT16 to INT23)**

Interrupts are issued in response to the operation status of an intelligent unit with an interrupt issuing function.

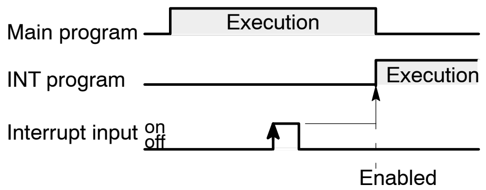
③ **Periodical interrupt (INT24)**

The interrupt occurs in fixed time intervals. Set the time interval with the **ICTL** instruction.

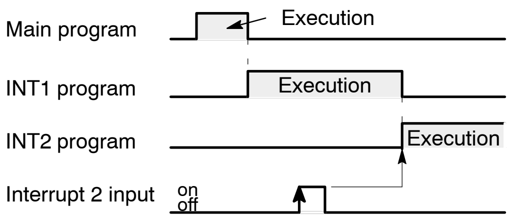
When the interrupt occurs, the interrupt program with the corresponding number is executed.



If interrupts are disabled, an interrupt will occur only at the point when interrupts are enabled with the **ICTL** instruction.



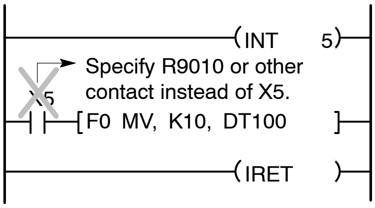
When another interrupt program is being executed, an interrupt will occur after the current program is completed.



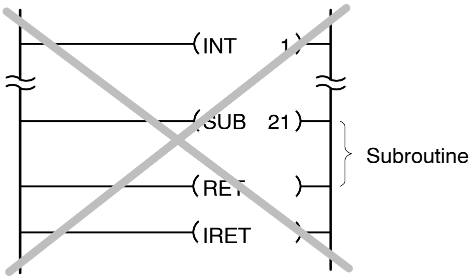
Precautions during programming for all types

If either the **INT** instruction or **IRET** instruction is missing, a syntax error will result.

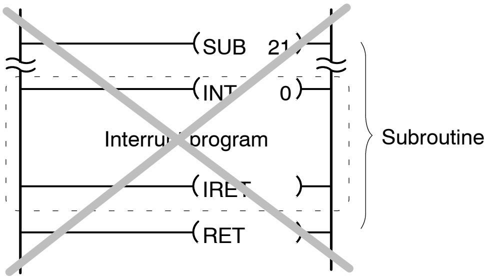
When an interrupt is issued, the operation memory corresponding to the interrupt input contact does not undergo I/O refreshing. Therefore, contacts other than the interrupt input contact, such as the constantly−on relay R9010, should be specified by the input conditions in the interrupt program.



A subroutine program cannot be used in an interrupt program.



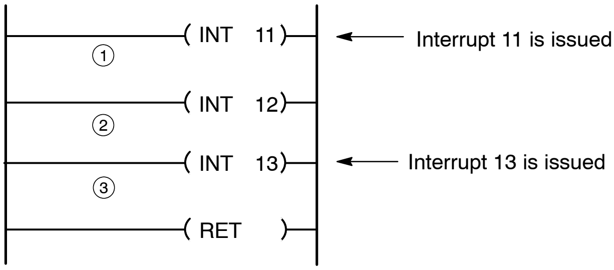
An interrupt program cannot be used in a subroutine program.



Interrupt programs with multiple entrances and one exit can be written.

Example:

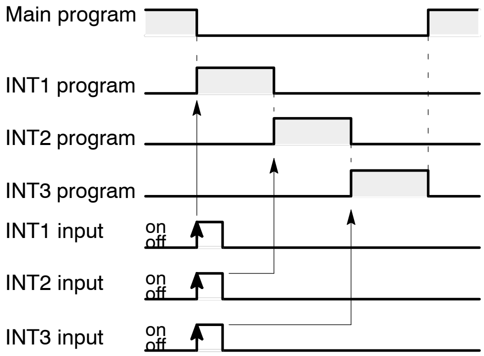
When the interrupt of the interrupt program 11 is issued, ① to ③ are executed. When the interrupt of No. 13 is issued, ③ is executed.



Control when more than one interrupt occurs simultaneously.

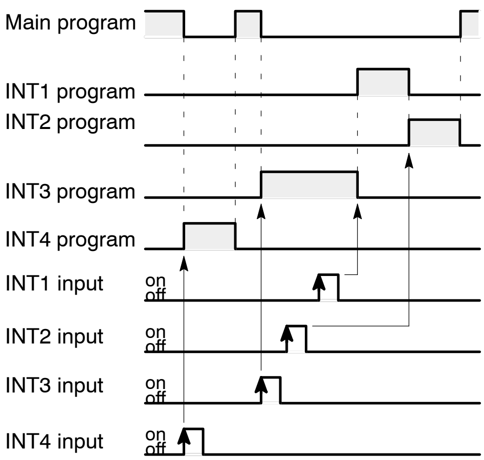
When more than one interrupt occurs simultaneously, the interrupt program with the smaller number is executed first. The other interrupt programs are then placed in the execution waiting state. After the first interrupt program is completed, the other programs will be executed in order from the smallest number to the greatest.

Example:



When more than one interrupt occurs while a interrupt program is being executed, the other programs will be executed in order from the smallest number to the greatest after the program currently being executed is finished.

Example:

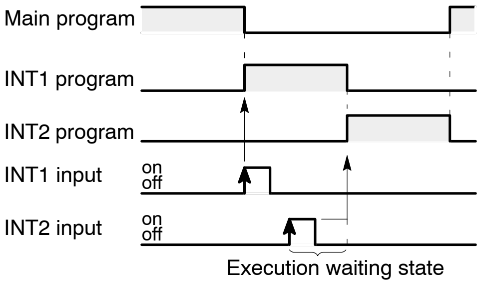


During execution of the INT3 program in the example above, the INT2 input occurred before INT1. But, after INT3 program is done, the INT1 program is executed first and then INT2 is executed.

Interrupt program execution waiting state and clearing

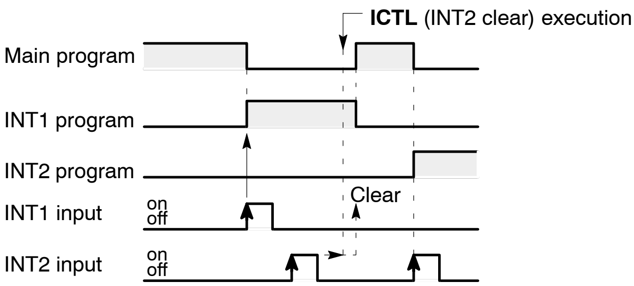
When multiple interrupt programs occur simultaneously or new interrupt programs occur during the execution of another interrupt program, the interrupt programs of lower preference are placed in the execution waiting state. They are then executed in order of preference when the other interrupt programs are completed.

Example:



If placed in the execution waiting state, there is a time difference between the time of when the interrupt occurs and when the interrupt program is actually executed. If you do not want to execute the interrupt program placed in the execution waiting state because of this, it can be cleared with the **ICTL** instruction. An interrupt program that is cleared will not be executed.

Example:



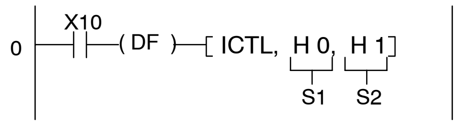
When the execution of interrupt programs is disabled with the **ICTL** instruction, interrupts that occur are still placed in the execution waiting state. When the execution is enabled with the **ICTL** instruction, the waiting interrupt programs will then be executed. Programs in the execution waiting state can be cleared with the **ICTL** instruction.

Availability
FP0/FP−e/FPΣ/FP−X/ FP0R

ICTL Interrupt control

Outline Performs the interrupt enable or disable and the interrupt clear.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	0	ST X 10
	1	DF
	2	ICTL
		H 0
		H 1

S1	16-bit equivalent constant or 16-bit area for interrupt control data setting
S2	16-bit equivalent constant or 16-bit area for interrupt condition setting

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A: Available

Description

When the **ICTL** instruction is executed, the interrupt program enable/disable and interrupt clear are set according to the settings in S1 and S2.

Be sure to use **ICTL** instructions so that they are executed once at the leading edge of the execution condition (trigger) using the **DF** instruction.

Two or more **ICTL** instructions can have the same execution condition (trigger).

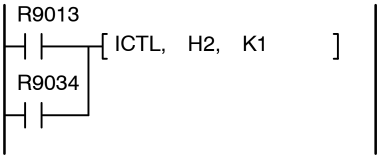
☞ Note

Before executing an interrupt program, be sure to execute the ICTL instruction and enable the execution of the interrupt program.

Precaution if rewriting during a RUN operation (for FP0/FP0R/FP−e/FPΣ)

If rewriting is done during a RUN operation while the interrupt function is being used, execution of the interrupt function is inhibited. The ICTL instruction has to be used once again to enable the interrupt program to be executed.

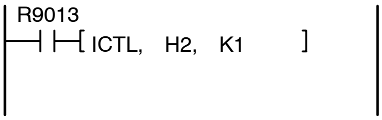
Example: A periodic interrupt is set every 10 ms when the operation is begun. (After rewriting during a RUN operation, interrupts are enabled again.)



Executes INT24 every 10 ms.

Input examples

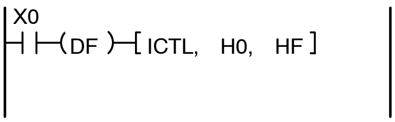
Example 1: Setting a periodical interrupt every 10ms from the start of operations



Executes INT24 every 10ms

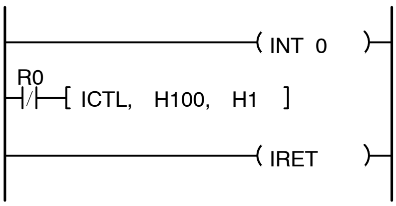
The R9013 (initial pulse relay) turns on only for the first scan after operations begin.

Example 2: Enable INT0 through INT3 when X0 rises.



X0: Enables INT0 to INT3 when on

Example 3: Clear interrupts other than INT0 after the INT0 program is completed.



Interrupts other than INT0 are cleared when the program of INT0 is finished

Specifying control data

S1: Specifying the control functions and interrupt types

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
S1				

Interrupt type selection
H00: INT 0 to INT 7
H02: INT 24 (10ms units)
H03: INT 24 (0.5ms units)

Selection of control function
H00: Interrupt “enabled/disabled” control
H01: Interrupt trigger reset control

Set S1 = H0 to specify enable or disable for the execution of INT0 through INT7.
Set S1 = H100 to clear interrupts INT0 through INT7.
Set S1 = H2 (for units of 10ms) to set the time interval for INT24.
Set S1 = H3 (for units of 0.5ms) to set the time interval for INT24.

Precautions during programming

The inputs that can be actually used as an interrupt input are different depending on the models. (Refer to the table below.)

Interrupt Program No.	Interrupt input			
	FP0/FP−e	FPΣ/FP0R	FP−X Ry	FP−X Tr
INT0	X0	X0	X0	X0
INT1	X1	X1	X1	X1
INT2	X2	X2	X2	X2
INT3	X3	X3	X3	X3
INT4	X4	X4	X4	X4
INT5	X5	X5	X5	X5
INT6	—	X6	X6	X6
INT7	—	X7	X7	X7
INT8	—	—	X100	—
INT9	—	—	X101	—
INT10	—	—	X102	—
INT11	—	—	X200	—
INT12	—	—	X201	—
INT13	—	—	X202	—
INT24	Periodical interrupt			

Note) When using the high−speed counter−initiated interrupt program, the counting performance of the high−speed counter may be decrement at the moment of the start−up of the interrupt program.

S2: Specifying the control of interrupts

① Enabling or disabling interrupt programs (when S1 = H0 or S1 = H1).

Set the control data in the bit corresponding to the number of the interrupt program that you want to control.

Set the bit corresponding to the number of the program you want to enable to “1.” (INT program disabled.)

Set the bit corresponding to the number of the program you want to disable to “0.” (INT program enabled.)

Example: When specified so that the interrupt programs INT1 and INT2 are enabled, and INT0 and INT3 to 13 are inhibited

Bit position	15 · ·12	11 · · 8	7 · · 4	3 · · 0
INT program number	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
S2 (Enabled/ disabled)	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0

② Clearing interrupt programs (when S1 = H100 or S1 = H101)

Set the control data in the bit corresponding to the number of the interrupt program that you want to control.

Set the bit corresponding to the number of the program you want to clear to “0.” (INT program disabled.)

Set the bit corresponding to the number of the program you want to not clear to “1.” (INT program enabled.)

Example: When specified so that the interrupt programs INT0 to INT2 are cleared, and INT3 to INT13 are not cleared

Bit position	15 · ·12	11 · · 8	7 · · 4	3 · · 0
INT program number	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
S2 (Enabled/ disabled)	0 0 0 0	0 0 0 0	1 1 1 1	1 0 0 0

③ Specifying periodical interrupt (when S1 = H2)

Specify the setting with decimal number. The time interval = value of S2 × 10 (ms).

Bit position	15 · ·12	11 · · 8	7 · · 4	3 · · 0
S2				

K0 to K3000

Time interval setting: K1 to K3000 (10ms to 30s)

INT24 disabled: K0

④ Specifying periodical interrupt programs (when S1 = H3)

The time interval = value of S2 × 0.5 (ms).

Bit position	15 · ·12	11 · · 8	7 · · 4	3 · · 0
S2				

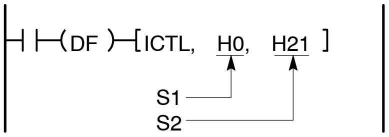
K0 to K3000

Time interval setting: K1 to K3000 (0.5ms to 1.5s)

INT24 disabled: K0

Example of enabling the execution of interrupt programs

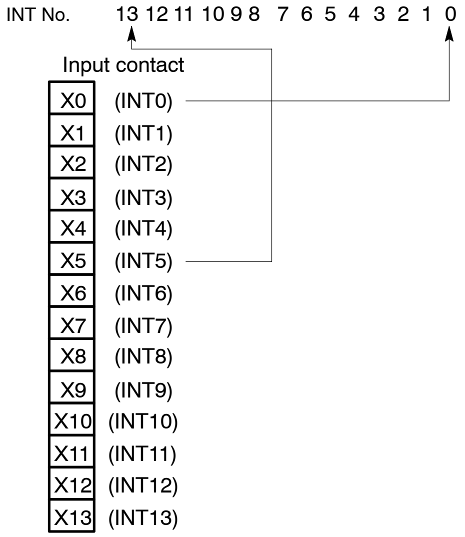
Example:



S1: H0000
Specifies enabling or disabling of interrupt programs that correspond to interrupts at specified input contact or to target value match interrupts.

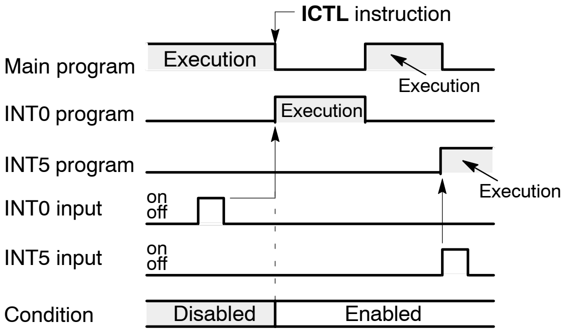
S2: H0021
Enable INT0 and INT5 (set bits 0 and 5 to “1”) and disable all others.

Bit position	15 · ·12	11 · · 8	7 · · 4	3 · · 0
S2	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 1



Set the bits to “1” that correspond to the interrupts to be enabled.

When this ICTL instruction is executed, interrupt programs No. 0 and No. 5 will be executed when their corresponding interrupt inputs occur.



How to start the interrupt program when executing the high−speed counter match ON/match OFF instruction.

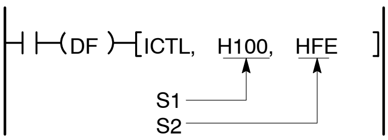
1. Set the counter by the system register. (It is not necessary to set the external interrupt.)
2. Describe the interrupt program on the program. The high−speed counter corresponds to the interrupt program as below.

Interrupt Program No.	High−speed counter−initiated interrupt			
	FP0/FP−e	FPΣ/FP0R	FP−X Ry	FP−X Tr
INT0	ch0	ch0	ch0	ch0
INT1	ch1	ch1	ch1	ch1
INT2	—	—	ch2	ch2
INT3	ch2	ch2	ch3	ch3
INT4	ch3	ch3	ch4	ch4
INT5	—	—	ch5	ch5
INT6	—	—	ch6	ch6
INT7	—	—	ch7	ch7
INT8	—	—	ch8	—
INT9	—	—	ch9	—
INT10	—	—	—	—
INT11	—	—	chA	—
INT12	—	—	chB	—
INT13	—	—	—	—

3. Enable the setting by the ICTLinstruction.
   Enable ICTL, H0, H9 −INT0 and INT7.
4. Start the match ON/match OFF instruction.
5. The program is executed when the conditions for the match ON/match OFF instruction are met.

Example for clearing interrupt programs

Example:

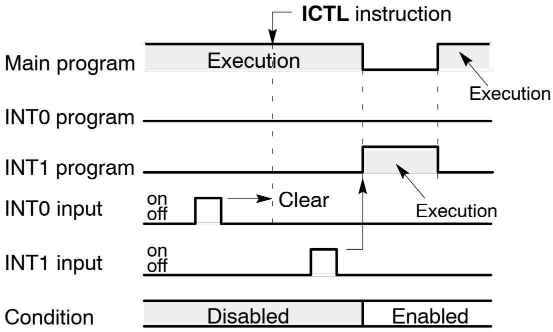


S1: H100
Clears interrupts from specified input contact or target value match interrupts.

S2: HFE
Clears interrupt INT0 (bit 0 is “0”) and does not clear the other interrupts.

For the relationship between the set value and the interrupt input contact, refer to page 2 – 106.

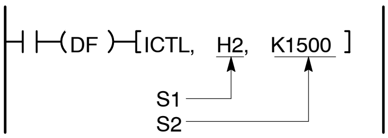
Even though the INT0 interrupt input occurred, when the interrupt program is disabled, the ICTL instruction can still be used to clear the INT0 interrupt.



Since INT0 is cleared, the INT0 program will not be executed even after execution is enabled. Since INT1 is not cleared, the INT1 program will be executed after execution is enabled.

Example for setting periodical interrupt

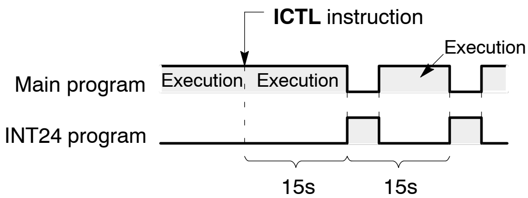
Example:



[S1]: H0002
Specifies periodical interrupt

[S2]: K1500
Specifies the time interval for the periodical interrupt. With K1500, the time interval is K1500 x 10ms = 15000ms (15s)

After this ICTL instruction is executed, the periodical interrupt will occur every 15 seconds. At these times, the INT24 interrupt program will be executed.



To stop the periodical interrupt program, execute the following program.

```
|--| |--(DF)--[ICTL,   H2,   K0      ]  |
```

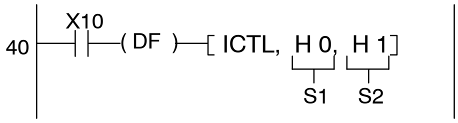
Availability
FP2/FP2SH/FP10SH

ICTL Interrupt control

Outline Performs the interrupt enable or disable and the interrupt clear.

Program example

Ladder Diagram



Boolean

Address	Instruction		
40	ST	X	10
41	DF		
42	ICTL		
	H		0
	H		1

S1	16-bit equivalent constant or 16-bit area for interrupt control data setting
S2	16-bit equivalent constant or 16-bit area for interrupt condition setting

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX (*1)	IY (*2)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A: Available

(*1) With the FP2, FP2SH and FP10SH, this is I0 to IC.

(*2) With the FP2, FP2SH and FP10SH, this is ID.

Description

When the **ICTL** instruction is executed, the interrupt program enable/disable and interrupt clear are set according to the settings in S1 and S2.

Be sure to use **ICTL** instructions so that they are executed once at the leading edge of the execution condition (trigger) using the **DF** instruction.

Two or more **ICTL** instructions can have the same execution condition (trigger).

☞ **Note**

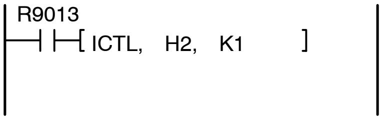
Before executing an interrupt program, be sure to execute the ICTL instruction and enable the execution of the interrupt program.

Flag conditions

- Error flag (R9007, R9008):
  - It turns on, when the specified address using the index modifier exceeds a limit.
  - It turns on, when the value outside of the range is specified for the interruption type and control function of [S1].
  - It turns on, when the value outside of the range is specified for [S2].

Input examples

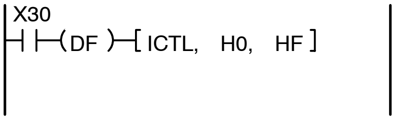
Example 1: Setting a periodical interrupt every 10ms from the start of operations



Executes INT24 every 10ms

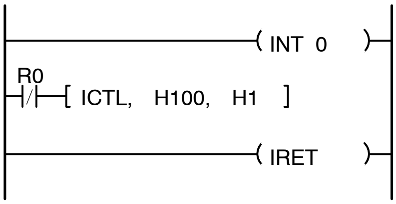
The R9013 (initial pulse relay) turns on only for the first scan after operations begin.

Example 2: Enable INT0 through INT3 when X30 rises.



X30: Enables INT0 to INT3 when on

Example 3: Clear interrupts other than INT0 after the INT0 program is completed.



Interrupts other than INT0 are cleared when the program of INT0 is finished

Specifying control data

S1: Specifying the control functions and interrupt types

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
S1				

Interrupt type selection (bits 7 to 0)
H00: INT 0 to INT15
H01: INT16 to INT23 (*1)
H02: INT24 (10ms units)
H03: INT24 (0.5ms units) (*2)
H04: INT24 (in increments of 10 ms) (*3)
H05: INT24 (in increments of 0.5 ms) (*3)

Selection of control function (bits 15 to 8)
H00: Interrupt execution “enabled/disabled”
H01: Interrupt clear

(*1) The intelligent unit which issues interrupts has a high−speed counter unit, a pulse output unit, and other units.

(*2) FP2/FP2SH/FP10SH only

(*3) Available from FP2/FP2SH Ver. 1.50 or later

If execution has been specified as enabled or disabled for INT0 to INT15, [S1] = H0.
If an interrupt clear has been specified for INT0 to INT15, [S1] = H100.
If execution has been specified as enabled or disabled for INT16 to INT23, [S1] = H1.
If an interrupt clear has been specified for INT16 to INT23, [S1] = H101.
Set [S1] = H2 to set the time intervals for INT24.
Set [S1] = H3 to set the time intervals for INT24 (for the FP2, FP2SH and FP10SH only)

S2: Specifying the control of interrupts

① Enabling or disabling interrupt programs (when S1 = H0 or S1 = H1).

Set the control data in the bit corresponding to the number of the interrupt program that you want to control.

Set the bit corresponding to the number of the program you want to enable to “1.” (INT program disabled.)

Set the bit corresponding to the number of the program you want to disable to “0.” (INT program enabled.)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
INT program number	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
S2 (Enabled/ disabled)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
INT program number	——	——	23 22 21 20	19 18 17 16
S2 (Enabled/ disabled)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

② Clearing interrupt programs (when S1 = H100 or S1 = H101)

Set the control data in the bit corresponding to the number of the interrupt program that you want to control.

Set the bit corresponding to the number of the program you want to clear to “0.” (INT program disabled.)

Set the bit corresponding to the number of the program you want to not clear to “1.” (INT program enabled.)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
INT program number	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
S2 (Enabled/ disabled)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
INT program number	——	——	23 22 21 20	19 18 17 16
S2 (Enabled/ disabled)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

③ Specifying periodical interrupt (when S1 = H2 or S1=H4)

Specify the setting with decimal number. The time interval = value of S2 × 10 (ms).

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
S2				

K0 to K3000

Time interval setting: K1 to K3000 (10ms to 30s)

INT24 disabled: K0

Note

For the difference in the operation of H2 and H4, refer to ”Example 2 for setting periodical interrupt”.
(H4 can be specified with the PLC FP2/FP2SH Ver. 1.50 or later.)

④ Specifying periodical interrupt programs (when S1 = H3 or S1=H5) for FP0/FP2/FP2SH/FP10SH only

Specify the setting with decimal number.

The time interval = value of S2 × 0.5 (ms).

Bit position	15 · ·12	11 · · 8	7 · · 4	3 · · 0
S2				

$\underbrace{\qquad\qquad\qquad\qquad}_{\text{K0 to K3000}}$

Time interval setting: K1 to K3000 (0.5ms to 1.5s)

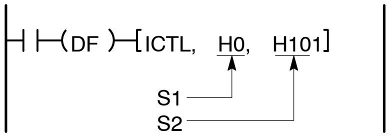
INT24 disabled: K0

☞ Note

For the difference in the operation of H3 and H5, refer to "Example 2 for setting periodical interrupt".
(H5 can be specified with the PLC FP2/FP2SH Ver. 1.50 or later.)

Example of enabling the execution of interrupt programs

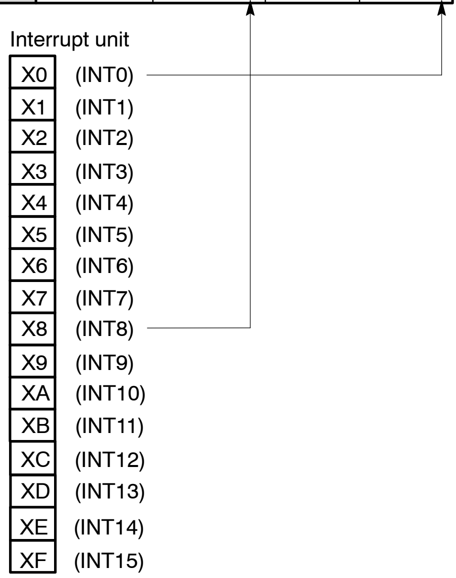
Example:



[S1]: H0000
This specifies whether execution of the interrupt program corresponding to the interrupt from the interrupt unit (INT0 to INT15) is enabled or disabled.

[S2]: H0101
Enable INT0 and INT8 (set bits 0 and 8 to “1”) and disable all others.

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
S2	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1
INT number	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0

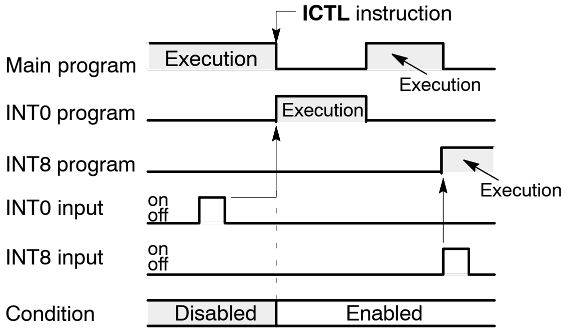


Set the bits to “1” that correspond to the interrupts to be enabled.

The I/O number is an example showing the interrupt unit mounted in Slot 0.

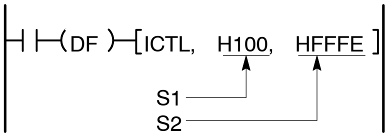
To enable all interrupts INT0 to INT15, set S2 = HFFFF.

When this **ICTL** instruction is executed, interrupt programs INT0 and INT8 will be executed when their corresponding interrupt inputs occur.



Example for clearing interrupt programs

Example:



[S1]: H0100

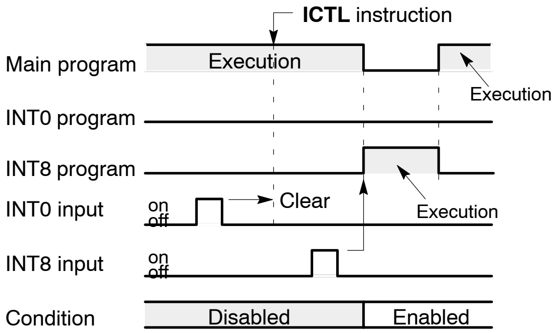
Clears interrupts from the interrupt unit (INT 0 to INT15).

[S2]: HFFFE

Clears interrupt INT0 (bit 0 is “0”) and does not clear the other interrupts.

For the relationship between the set value and the interrupt unit, refer to page 2 – 114 “Example of enabling the execution of interrupt programs.”

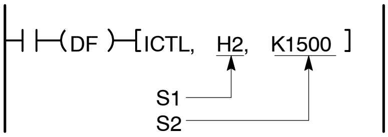
Even though the INT0 interrupt input occurred, when the interrupt program is disabled, the ICTL instruction can still be used to clear the INT0 interrupt.



Since INT0 is cleared, the INT0 program will not be executed even after execution is enabled. Since INT8 is not cleared, the INT8 program will be executed after execution is enabled.

Example 1 for setting periodical interrupt

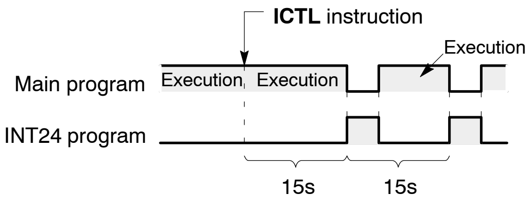
Example:



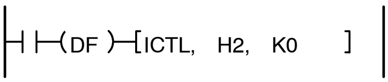
[S1]: H0002
Specifies periodical interrupt (units: 10ms)

[S2]: K1500
Specifies the time interval for the periodical interrupt. With K1500, the time interval is K1500 x 10ms = 15000ms (15s)

After this ICTL instruction is executed, the periodical interrupt will occur every 15 seconds. At these times, the INT24 interrupt program will be executed.

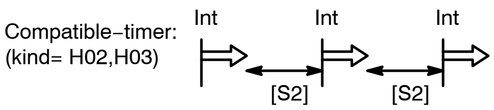


To stop the periodical interrupt program, execute the following program.

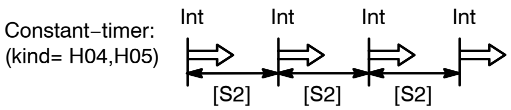


Example 2 for setting periodical interrupt

When H4 or H5 is designated, the periodical interrupt occurs at the specified interval regardless of interrupt processing time.

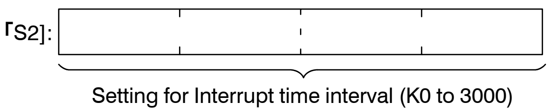


After the periodical interrupt program completed, the next interrupt timing is counted.
When the execution time of the periodical interrupt program is less than 500 μs, the interrupt is carried out at every interval specified by [S2]. However, when the execution time is 500 μs or longer, the interval is automatically shifted in increments of 500 μs.



Regardless of the execution time of the interrupt program, it is executed for the fixed interval.

The specified time interval for the periodical interrupt must be longer than the time taken for the interrupt processing.



When the specified time for the periodical interrupt is longer than the execution time of the interrupt program, the operation cannot be carried out for the specified interval, and the CPU may give an alarm.

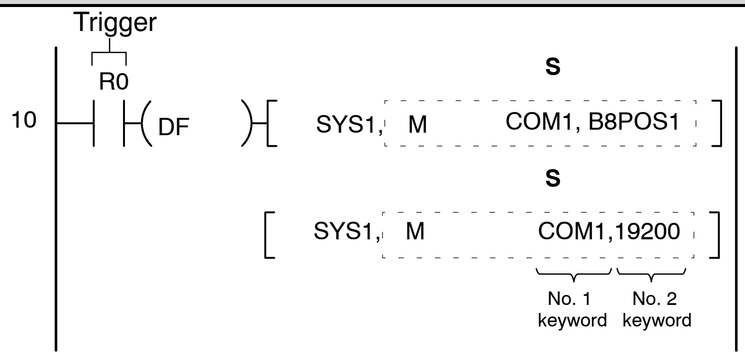
Availability
FPΣ/FP−X/FP0R

SYS1 Communication conditions setting

Outline This changes the communication conditions for the COM port or Tool port based on the contents specified by the character constant.

Program example

Ladder Diagram



Boolean Non-ladder	
Address	**Instruction**
10	ST R 0
11	DF
12	SYS1
	M COM1, B8POS1
25	SYS1
	M COM1,19200

S	Character constant “M”

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When R0 turns on, the transmission format and baud rate for the COM. 1 port are set as follows.

Character bit: 8, Parity: Odd
Stop bit: 1
Baud rate: 19,200 bps

Description

The communication conditions for the port specified by No. 1 keyword are changed to the contents specified by No. 2 keyword.

Contents that can be changed include the following:

1) Communication format
2) Baud rate
3) Unit No. For FP0R, indirect settings are available.
4) Header and Terminator
5) RS (Request to Send) control

Keyword setting

1) **Communication format** (Shared by the Tool, COM. 1 and COM. 2 ports)

SYS1, M TOOL,B7 PN S1

Port used
TOOL: Tool port
COM1: COM. 1 port
COM2: COM. 2 port

Character bit
B7: 7bits B8: 8bits

Parity
PN: None PO: Odd PE: Even

Stop bit
S1: 1 S2: 2

2) **Baud rate** (Shared by the Tool, COM. 1 and COM. 2 ports)

SYS1, M TOOL,19200

Port used
TOOL: Tool port
COM1: COM. 1 port
COM2: COM. 2 port

Baud rate

300: 300 bps	600: 600 bps
1200:1200 bps	2400: 2,400 bps
4800: 4,800 bps	9600: 9,600 bps
19200: 19,200 bps	38400: 38,400 bps
57600: 57,600 bps	115200: 115,200 bps

The baud rates of 300, 600 or 1200 bps can be specified only with the FP0R, FP−X ver 2.0 or later and FPΣ ver3.10 or later.
Also, those baud rates cannot be specified by the system register.

3) **Unit No**. (Shared by the Tool, COM. 1 and COM. 2 ports)

SYS1, M TOOL,No1

Port used
TOOL: Tool port
COM1: COM. 1 port
COM2: COM. 2 port

Unit No.
No1 to No99: No. 1 to No. 99

For FP0R, the indirect settings of unit number is available.

SYS1, M COM1, D000D

Port used
TOOL: Tool port
COM1: COM. port

DT number that Unit No. is stored.

D000D = DT0
D9999 = DT9999 Always specify with a number of D+4 digits.

4) **Header and Terminator** (Shared by the TOOL, the COM. 1 and COM. 2 ports)

SYS1, M COM1,STX

Port used
TOOL: Tool port (FPΣ 32k/FP−X/FP0R)
COM1: COM. 1 port
COM2: COM. 2 port

Header
STX: STX
NOSTX: STX not exist

Terminator
ETX: ETX
CR: CR
CRLF: CR + LF
NOTERM: None

5) **RS (Request to Send) control** (COM. 1 port only)

SYS1, M COM1,RTS1

Port used
COM1: COM. 1 port

RS control for the 1−channel RS232C type communication cassette
RTS1: Disables communication
(Sets the RS terminal to “on”)
RTS0: Enables communication
(Sets the RS terminal to “off”)

* Not available for FP0R.

Precautions during programming

- Executing this instruction does not rewrite the contents of the system ROM in the control unit. As a result, turning the power supply off and then on again rewrites the contents of the system registers specified by the tool software.
- We recommend using differential execution with this instruction.
- Because the system register settings are changed, a verification error may occur in some cases if verification is carried out with the tools.
- For No. 1 and No. 2 keywords, input 12 letters after “M” aligning to the right. Separate No.1 and No.2 keywords with a comma ”,” and do not use spaces. An operation error will occur.
  [Example] If inputting (SYS1, M COM1, WAIT2)
  Input => M ␣ ␣ C O M 1 , W A I T 2
  Input a space after “M” to be 12 letter aligning to the right.
- For FP0R, specify COM1 for COM port.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - Any character other than a keyword is specified
  - There is no comma between No. 1 and No. 2 keywords
  - The small letter of the alphabet is used to specify the keyword (except for numbers used to specify unit No.)
  - No communication cassette has been installed when COM1 or COM2 has been set
  - The setting of the unit No. setting switch is anything other than 0 when COM1 or COM2 has been set and the unit No. is being changed
  - The unit No. set using this instruction is anything other than a value between 1 and 99
  - The baud rate or transmission format for COM1 has been changed when the PLC link mode is specified for COM1
  - The baud rate or transmission format is changed while the Tool port, COM. port 1, or COM. port 2 is being initialized using MODEM
  - The communication mode is set to anything other than the general communication mode when header and terminator have been set
  - Any communication cassette other than the 1−channel RS232C type communication cassette is installed when using RS control
  - The specified unit No. is larger than the largest unit No. specified by the system register when the COM. 1 port is in the PLC link mode

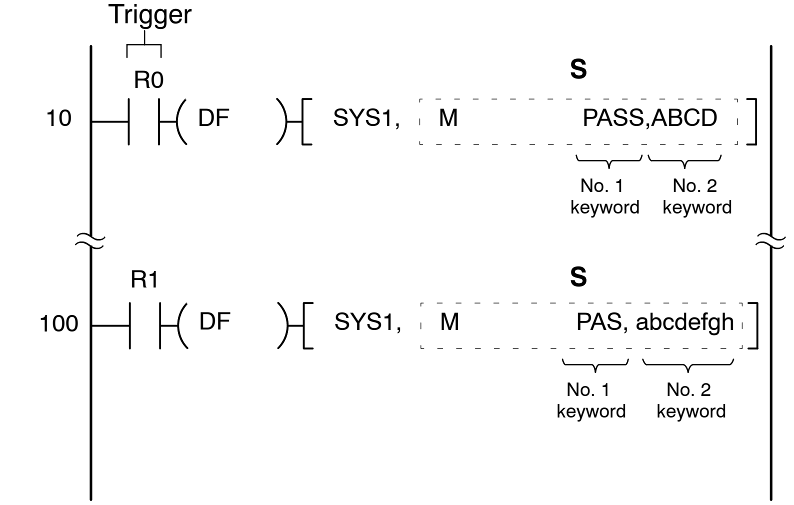
Availability
FPΣ/FP−X/FP0R

SYS1 Password setting

Outline This changes the password specified by the controller, based on the contents specified by the character constant.

Program example

Ladder Diagram



Boolean Non-ladder

Address	Instruction
10	ST R 0
11	DF
12	SYS1
	M PASS,ABCD
100	ST R 1
101	DF
102	SYS1
	M PAS,abcdefgh

S	Character constant “M”

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When R0 turns on, the controller password is changed to “ABCD”.

Description

This changes the password specified by the controller to the contents specified by No. 2 keyword.

Keyword setting

For the 4−digit password

SYS1, M PASS,ABCD

PASS: Fixed

Password (Example: To set the password to “ABCD”)

For the 8−digit password (It is available for FPΣ 32k/FP−X/FP0R.)

SYS1, M PAS, abcdefgh

PAS: Fixed

Password (Example: When “abcdefgh” is specified for the password.)

If the specified characters are less than 8, spaces are added at the end of the characters to be 8−digit password.

Precautions during programming

- When this instruction is executed, writing to the internal F−ROM takes approximately 100 ms.
- If the specified password is the same as the password that has already been written, the password is not written to the F−ROM.
- We recommend using differential execution with this instruction.
- For No. 1 and No. 2 keywords, input 12 letters after “M” aligning to the right. Separate No.1 and No.2 keywords with a comma ”,” and do not use spaces. An operation error will occur.
  [Example] If inputting (SYS1, M COM1, WAIT2)
  Input => M ␣ ␣ C O M 1 , W A I T 2
  Input a space after “M” to be 12 letter aligning to the right.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - Any character other than a keyword is specified
  - There is no comma between No. 1 and No. 2 keywords
  - The small letter of the alphabet is used to specify the keyword (For the 4−digit password)
  - The data specified for the password setting is any character other than 0 to 9 or A to F, or the specified data consists of other than four digits. (For the 4−digit password)

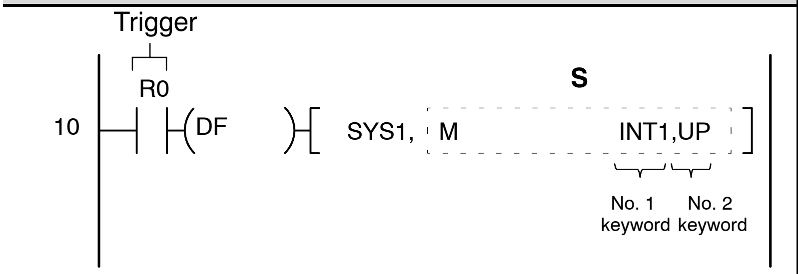
Availability
FPΣ/FP−X/FP0R

SYS1 Interrupt setting

Outline This sets the interrupt input based on the contents specified by the character constant.

Program example

Ladder Diagram



Boolean Non-ladder	
Address	**Instruction**
10	ST R 0
11	DF
12	SYS1
	M INT1,UP

S	Character constant "M"

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When R0 turns on, input X1 is set to the interrupt that becomes valid at the rising edge.

Description

This sets the input specified by No. 1 keyword as the interrupt input, and changes the input conditions to the contents specified by No. 2 keyword.

Keyword setting

SYS1, M INT2,UP

Interrupt input

INT0: X0 INT1: X1 INT8: X0 INT9: X1
INT2: X2 INT3: X3 INT10: X2 INT11: X3
INT4: X4 INT5: X5 INT12: X4 INT13: X5
INT6: X6 INT7: X7
INT8 − INT10 → Pulse I/O cassette 1
INT11 − INT12 → Pulse I/O cassette 2

Effective edges

UP: Rising edge
DOWN: Falling edge
BOTH: Rising and falling edges

Precautions during programming

- Executing this instruction does not rewrite the contents of the system ROM in the control unit. As a result, turning the power supply off and then on again rewrites the contents of the system registers specified by the tool software.
- We recommend using differential execution with this instruction.
- When UP or DOWN has been specified, the contents of the system registers change in accordance with the specification, so a verification error may occur in some cases, when the program is verified. When BOTH has been specified, the contents of the system registers do not change.
- For No. 1 and No. 2 keywords, input 12 letters after “M” aligning to the right. Separate No.1 and No.2 keywords with a comma ”,” and do not use spaces. An operation error will occur.
  [Example] If inputting (SYS1, M COM1, WAIT2)
  Input => M ␣ ␣ C O M 1 , W A I T 2
  Input a space after “M” to be 12 letter aligning to the right.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - Any character other than a keyword is specified
  - There is no comma between No. 1 and No. 2 keywords
  - The small letter of the alphabet is used to specify the keyword

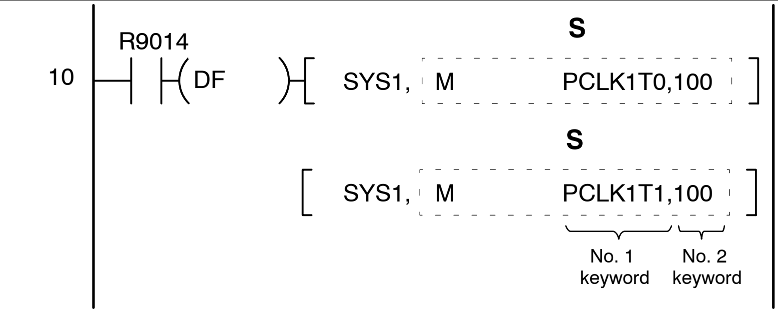
Availability
FPΣ/FP-X/FP0R

SYS1 PLC link time setting

Outline This sets the system setting time when a PLC link is used, based on the contents specified by the character constant.

Program example

Ladder Diagram



Boolean Non-ladder	
Address	**Instruction**
10	ST R 90141
11	DF
12	SYS1
	M PCLK1T0,100
25	SYS1
	M PCLK1T1,100

S	Character constant “M”

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When R9014 turns on when a PLC link is being used, the link entry wait time and the error detection times for transmission assurance relay are set as follows.

Link entry wait time: 100 ms

Error detection time for transmission assurance relay: 100 ms

Description

The conditions specified by No. 1 keyword are set as the time specified by No. 2 keyword.

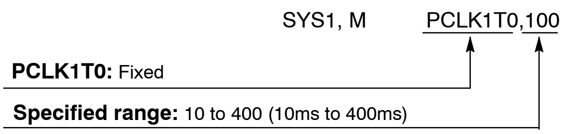
The setting for the link entry waiting time is set if the transmission cycle time is shortened when there are stations that have not joined the link (*).

* Stations that have not joined the link: Stations that have not been connected between the No. 1 station and the station with the largest number, or stations for which the power supply has not been turned on

The error detection time setting for the transmission assurance relay is set if the time between the power supply being turned off at one station and the transmission assurance relay being turned off at a different station is to be shortened.

Keyword setting

1) **Link entry wait time**



2) **Error detection time for transmission assurance relay**

SYS1, M PCLK1T1,100

PCLK1T1: Fixed

Specified range: 100 to 6400 (100ms to 6400ms)

Precautions during programming

- The program should be placed at the beginning of all PLCs being linked, and the same values specified.
- This instruction should be specified in order to set special internal relay R9014 as the differential execution condition.
- The setting contents of the system registers are not affected by this instruction being executed.
- For No. 1 and No. 2 keywords, input 12 letters after "M" aligning to the right. Separate No.1 and No.2 keywords with a comma "," and do not use spaces. An operation error will occur.
  [Example] If inputting (SYS1, M COM1, WAIT2)
  Input => M ␣ ␣ C O M 1 , W A I T 2
  Input a space after "M" to be 12 letter aligning to the right.

Precautions when setting the link entry wait time

- This should be specified such that the value is at least twice that of the largest scan time of all the PLCs that are linked.
- If a short value has been specified, there may be some PLCs that are not able to join the link even though the power supply for that PLC has been turned on.
- If there are any stations that have not joined the link, the setting should not be changed, even if the link transmission cycle time is longer as a result. (The default value is 400 ms.)

Precautions when setting the error detection time for the transmission assurance relay

- This should be specified such that the value is at least twice that of the largest transmission cycle time of all the PLCs that are linked.
- If a short value has been specified, there is a possibility that the transmission assurance relay will malfunction.
- The setting should not be changed, even if the detection time for the transmission assurance relay is longer as a result. (The default value is 6400 ms.)

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - Any character other than a keyword is specified
  - There is no comma between No. 1 and No. 2 keywords
  - The small letter of the alphabet is used to specify the keyword
  - The specified value is outside the specified range

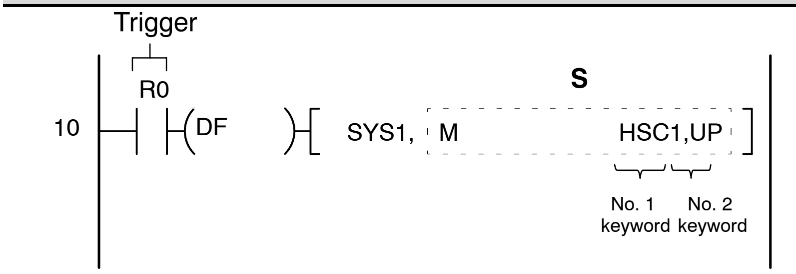
Availability
FPΣ 32k FP-X Ver 1.10 or more FP0R

SYS1 Change high-speed counter operation mode

Outline This changes the operation mode of the high-speed counter based on the contents specified by the character constant.

Program example

Ladder Diagram



Boolean Non-ladder	
Address	**Instruction**
10	ST R 0
11	DF
12	SYS1
	M HSC1,UP

S	Character constant "M"

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When R0 turns on, the operation mode of the high-speed counter CH0 is set to the addition mode.

Description

This changes the operation mode of the high-speed counter specified by No.1 keyword to the operation mode specified by No.2 keyword.

Keyword setting

SYS1, M HSC1,UP

High-speed counter setting

HSCn n: 0 to 9, A, B with FP-X Ry type
n: 0 to 7 with FP-X Tr type
n: 0, 1, 2, 3 with FPΣ
n: 0, 1, 2, 3, 4, 5 with FP0R

UP: Addition input setting
DOWN: Subtraction input setting

Precautions during programming

- If the system register is not set to the addition input or subtraction input for this instruction, an operation error occurs. Set the system register to the addition or subtraction input in advance. When the addition/subtraction input setting is specified even if the setting has been already done, an operation error does not occur.
- Executing this instruction does not rewrite the contents of the system ROM in the control unit. As a result, turning the power supply off and then on again rewrites the contents of the system registers specified by the tool software.
- We recommend using differential execution with this instruction.
- When UP or DOWN has been specified, the contents of the system registers change in accordance with the specification, so a verification error may occur in some cases, when the program is verified. When BOTH has been specified, the contents of the system registers do not change.
- For No. 1 and No. 2 keywords, input 12 letters after "M" aligning to the right. Separate No.1 and No.2 keywords with a comma "," and do not use spaces. An operation error will occur.
  [Example] If inputting (SYS1, M COM1, WAIT2)
  Input => M ␣ ␣ C O M 1 , W A I T 2
  Input a space after "M" to be 12 letter aligning to the right.

Flag conditions

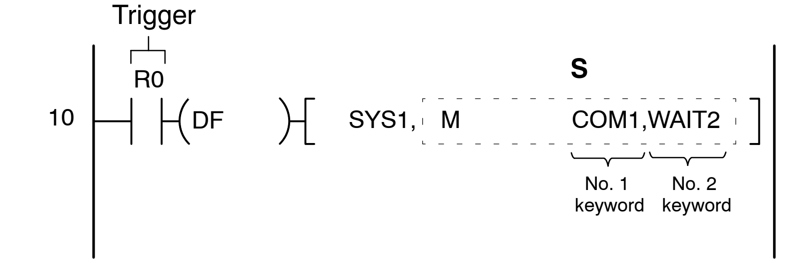
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - Any character other than a keyword is specified
  - There is no comma between No. 1 and No. 2 keywords
  - The small letter of the alphabet is used to specify the keyword
  - The system register is set to items other than the addition input or subtraction input.

Availability
FPΣ/FP-X/FP0R

SYS1 MEWTOCOL-COM response control

Outline This specifies the response waiting time based on the MEWTOCOL-COM of the COM port or Tool port, in response to the contents specified by the character constant.

Program example

Ladder Diagram	Boolean Non-ladder Address	Boolean Non-ladder Instruction
	10	ST R 0
	11	DF
	12	SYS1
		M COM1,WAIT2

S	Character constant "M"

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

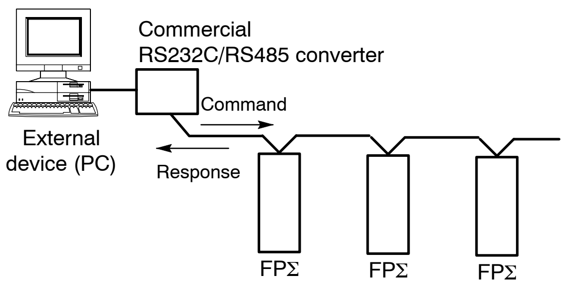
A: Available
N/A: Not Available

Description

The port MEWTOCOL-COM response time specified by No. 1 keyword is delayed based on the contents specified by No. 2 keyword.

This instruction is used to delay the response time on the PLC side until the state is reached in which commands can be sent by an external device and responses can be received from the PLC.

<Usage example>
When a commercial RS232C/RS485 converter is being used to carry out communication between a personal computer and the FPΣ, this instruction is used to return the PLC response after switching of the enable signal has been completed on the converter side.



Keyword setting

SYS1, M TOOL,WAITn

Port used
TOOL: Tool port
COM1: COM. 1 port
COM2: COM. 2 port

Response time
WAIT0 to WAIT999 (n: 0 to 999)

If the communication mode or the MOD BUS RTV mode has been set to the computer link mode, the set time is the scan time x n (n: 0 to 999).

If the communication mode has been set to the PLC link mode, the set time is n μs (n: 0 to 999).

If n = 0, the delay time set by this instruction will be set to “None”.

Precautions during programming

- The settings should not be changed as long as there is no trouble, to prevent the PLC link from getting unstable.
- This instruction is valid only if the setting on the controller side has been set to the computer link mode or the PLC link mode.
- The instruction should be executed at the beginning of the program, at the rise of R9014. The same value should be set for all linked PLCs.
- Executing this instruction does not change the settings in the system registers.
- If changing the settings, a value of at least twice should be set.
- We recommend using differential execution with this instruction.
- When the power supply to the PLC is off, the settings set by this instruction are cleared. (The set value will become 0.) If the mode is switched to the PROG. mode after the instruction has been executed, however, the settings will be retained.
- If a commercial RS232C/RS485 converter is being used in the PLC link mode, this instruction should be programmed in all of the stations (PLCs) connected to the link.
- For No. 1 and No. 2 keywords, input 12 letters after “M” aligning to the right. Separate No.1 and No.2 keywords with a comma ”,” and do not use spaces. An operation error will occur.
  [Example] If inputting (SYS1, M COM1, WAIT2)
  Input => M ␣ ␣ C O M 1 , W A I T 2
  Input a space after “M” to be 12 letter aligning to the right.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – Any character other than a keyword is specified
  – There is no comma between No. 1 and No. 2 keywords
  – The small letter of the alphabet is used to specify the keyword
  – No communication cassette has been installed when COM1 or COM2 has been set

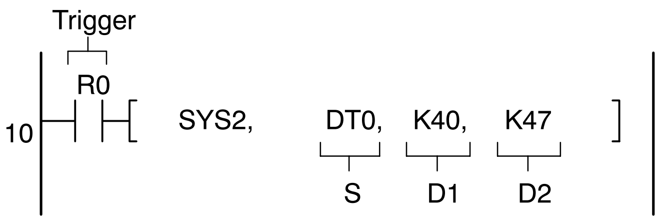
SYS2 Change system registers (No. 40 to No. 47, No. 50 to No. 57)

Availability
FPΣ/FP−X/FP0R

Outline This changes the settings entered for the system registers of the PLC link function, in accordance with the specified data.

Program example

Ladder Diagram



Boolean Address	Instruction		
10	ST	R	0
11	SYS2		
	DT		0
	K		40
	K		47

S	Starting number of the area in which 16−bit data is stored
D1	Starting number of the system registers being specified (K40 to K47)
D2	Ending number of the system registers being specified (K40 to K47)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A
D1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
D2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A

A: Available
N/A: Not Available

Description

The contents of system registers No. 40 to No. 47 are changed to the contents of the data registers starting with the number specified by [S].

Note) With the FP0R, the FPΣ 32k and the FP−X, the contents of system registers No. 50 to No. 57 are also changed.

System registers

	No.	Name	Setting value and range
PLC WO−0	40	Range of link relays used	0 to 64 words
	41	Range of link data registers used	0 to 128 words
	42	Starting number for link relay transmission	0 to 63
	43	Link relay transmission size	0 to 64 words
	44	Starting number for link data register transmission	0 to 127
	45	Link data register transmission size	0 to 127 words
	46	PC (PLC) Link switch flag	Normal/reverse
	47	Maximum unit number setting for MEWNET−W0 PLC link	1 to 16
PLC WO−1	50	Range of link relays used	0 to 64 words
	51	Range of link data registers used	0 to 128 words
	52	Starting number for link relay transmission	64 to 127
	53	Link relay transmission size	0 to 64 words
	54	Starting number for link data register transmission	128 to 255
	55	Link data register transmission size	0 to 127 words
	57	Maximum unit number setting for MEWNET−W0 PLC link	1 to 16

Program example

```
─┤ ├─[F0 MV , K  64, DT0 ]   Set value of system register 40
     [F0 MV , K 128, DT1 ]   Set value of system register 41
     [F0 MV , K   0, DT2 ]   Set value of system register 42
     [F0 MV , K  10, DT3 ]   Set value of system register 43
     [F0 MV , K   0, DT4 ]   Set value of system register 44
     [F0 MV , K  10, DT5 ]   Set value of system register 45
     [F0 MV , K   0, DT6 ]   Set value of system register 46
     [F0 MV , K   5, DT7 ]   Set value of system register 47
     [SYS2 , DT0, K40, K47 ] Sets the values stored in DT0 to DT7 in system
                             registers 40 to 47
```

Decimals of the average value are rounded off so that the average value is an integer.

Precaution during programming

- Executing this instruction does not rewrite the contents of the system ROM in the control unit. As a result, turning the power supply off and then on again rewrites the contents of the system registers specified by the tool software.
- A value between K40 and K47 should be specified for “D1” or “D2”. Also, the values should always be specified in such a way that D1 $\leqq$ D2.
- The values of the system registers change, so a verification error may occur when the program is verified.

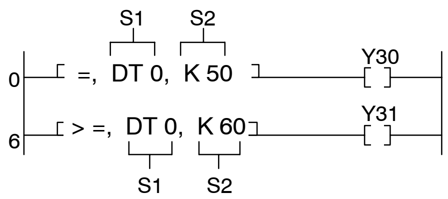
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - D1 > D2
  - The specified value is outside the ranges specified for the various system registers setting values

ST =	**16–bit data comparison:**	**Start equal**
ST <>	**16–bit data comparison:**	**Start equal not**
ST >	**16–bit data comparison:**	**Start larger**
ST >=	**16–bit data comparison:**	**Start equal or larger**
ST <	**16–bit data comparison:**	**Start smaller**
ST <=	**16–bit data comparison:**	**Start equal or smaller**

Outline Performs start operation by comparing two word data items with the comparison condition. The contact goes on or off depending on the result of the comparison.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction		
	0	ST =		
		DT		0
		K		50
	5	OT	Y	30
	6	ST > =		
		DT		0
		K		60
	11	OT	Y	31

S1	16-bit equivalent constant or 16-bit area to be compared
S2	16-bit equivalent constant or 16-bit area to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A: Available

(*1) This cannot be used with the FP0/FP−e.

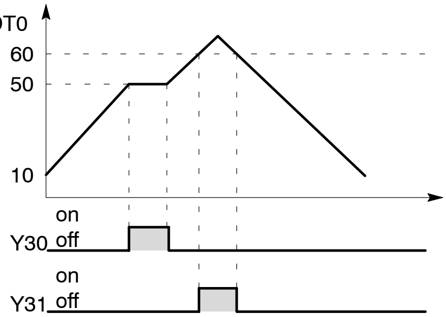
(*2) This cannot be used with the FP0/FP−e/FPΣ/FP−X/FP0R.

(*3) With the FPΣ, FP−X, FP0R, FP2, FP2SH and FP10SH, this is I0 to IC.

(*4) With the FPΣ, FP−X, FP0R, FP2, FP2SH and FP10SH, this is ID.

Explanation of example

Compares the contents of data register DT0 with the constant K50 and K60. If DT0 = K50, the external output relay Y30 goes on and if DT0 ≧ K60, the external output relay Y31 turns on.



Description

Compares the word data specified by S1 with the word data specified by S2 according to the comparison condition.

The **ST** instruction initiates a logical operation as the liaison contact when the comparison result is a specified status (=, <, >, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	S1 < S2	**S1 = S2**	**S1 > S2**
ST=	off	on	off
ST < >	on	off	on
ST>	off	off	on
ST> =	off	on	on
ST<	on	off	off
ST< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

Precautions concerning usage

The start comparison instructions **ST =, ST<>, ST >, ST >=, ST <**, and **ST <=** are programmed from the bus line.

If mixed with BCD or other type of data, the value will be regarded as negative when the most significant bit is 1 and a correct comparison may not be obtained. In this case, use an **F81 (BIN)** instruction or similar instruction to change the data to binary data before making the comparison.

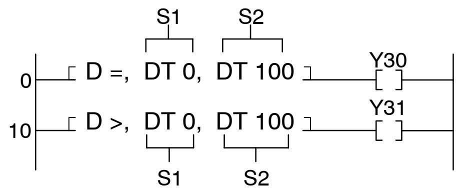
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

STD = 32–bit data comparison: Start equal
STD <> 32–bit data comparison: Start equal not
STD > 32–bit data comparison: Start larger
STD >= 32–bit data comparison: Start equal or larger
STD < 32–bit data comparison: Start smaller
STD <= 32–bit data comparison: Start equal or smaller

Outline Performs start operation by comparing two double word data items with the comparison condition. The contact goes on or off depending on the result of the comparison.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0 9 10 19	STD = DT 0 DT 100 OT Y 30 STD > DT 0 DT 100 OT Y 31

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0/FP−e.
(*2) This cannot be used with the FP0/FP−e/FPΣ/FP−X/FP0R.
(*3) With the FPΣ, FP−X, FP0R, FP2, FP2SH and FP10SH, this is I0 to IC.
(*4) With the FPΣ, FP−X, FP0R, FP2, FP2SH and FP10SH, this is ID.

Explanation of example

Compares the contents of data registers (DT1, DT0) with the data registers (DT101, DT100).
If (DT1, DT0) = (DT101, DT100), the external output relay Y30 goes on and if (DT1, DT0) > (DT101, DT100), the external output relay Y31 goes on.

Description

Compares the double word data specified by S1 and S1+1 with the double word data specified by S2 and S2+1 according to the comparison condition.

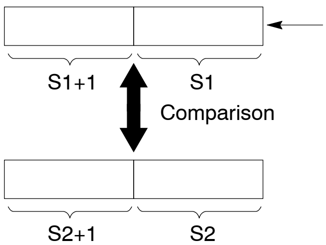
The **STD** instruction initiates a logical operation as the liaison contact when the comparison result is a specified status (=, <, >, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	(S1+1, S1) < (S2+2, S2)	**(S1+1, S1) = (S2+2, S2)**	**(S1+1, S1) > (S2+2, S2)**
STD=	off	on	off
STD< >	on	off	on
STD>	off	off	on
STD> =	off	on	on
STD<	on	off	off
STD< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.



The specified data area and the following data area are handled together as 32–bit data.

Precautions concerning usage

The start comparison instructions **STD =, STD <>, STD >, STD >=, STD <, and STD <=** are programmed from the bus line.

If mixed with BCD or other type of data, the value will be regarded as negative when the most significant bit is 1 and a correct comparison may not be obtained. In this case, use an **F83 (DBIN)** instruction or similar instruction to change the data to binary data before making the comparison.

Flag conditions

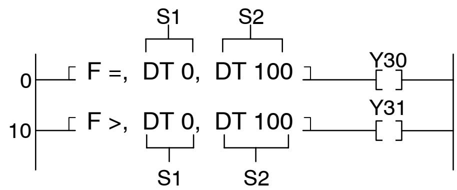
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

STF = Floating point real number data comparison: Start equal
STF <> Floating point real number data comparison: Start equal not
STF > Floating point real number data comparison: Start larger
STF >= Floating point real number data comparison: Start equal or larger
STF < Floating point real number data comparison: Start smaller
STF <= Floating point real number data comparison: Start equal or smaller

Availability
FP0R FP−X Ver 1.10 or more FPΣ 32k

Outline Performs start operation by comparing two single precision real number data items with the comparison condition. The contact goes on or off depending on the result of the comparison.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	STF =
		DT 0
		DT 100
	9	OT Y 30
	10	STF >
		DT 0
		DT 100
	19	OT Y 31

S1	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared
S2	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A

(*1) This cannot be used with the FPΣ/FP−X.

(*2) I0 to ID.

A: Available

* Index modification of a real number is not possible.

Explanation of example

Compares the real number value of data registers (DT0, DT1) with the real number value of data registers (DT100, DT101). If (DT0, DT1) = (DT100, DT101), the external output relay Y30 goes on and if (DT0, DT1) > (DT100, DT101), the external output relay Y31 goes on.

Description

Compares the real number data specified by S1 and S1+1 with the real number data specified by S2 and S2+1 according to the comparison condition.

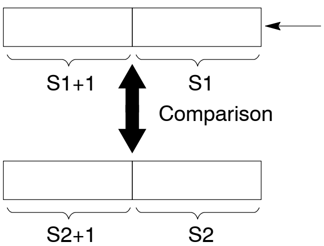
The **STF** instruction initiates a logical operation as the liaison contact when the comparison result is a specified status (=, <, >, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	(S1+1, S1) < (S2+2, S2)	(S1+1, S1) = (S2+2, S2)	(S1+1, S1) > (S2+2, S2)
STF=	off	on	off
STF< >	on	off	on
STF>	off	off	on
STF> =	off	on	on
STF<	on	off	off
STF< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.



The specified data area and the following data area are handled together as single precision real number data.

Precautions concerning usage

The start comparison instructions **STF =, STF <>, STF >, STF >=, STF <, and STF <=** are programmed from the bus line.

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

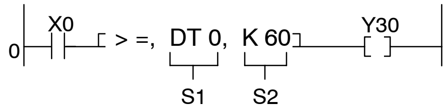
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data are specified in (S1+1, S1) and (S2+1, S2).

AN = 16–bit data comparison: AND equal
AN <> 16–bit data comparison: AND equal not
AN > 16–bit data comparison: AND larger
AN >= 16–bit data comparison: AND equal or larger
AN < 16–bit data comparison: AND smaller
AN <= 16–bit data comparison: AND equal or smaller

Outline Performs AND operation by comparing two word data items with the comparison condition. The contact goes on or off depending on the result of the comparison. The contacts are connected in series.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0 1 6	ST X 0 AN > = DT 0 K 60 OT Y 30

S1	16-bit equivalent constant or 16-bit area to be compared
S2	16-bit equivalent constant or 16-bit area to be compared

Operands

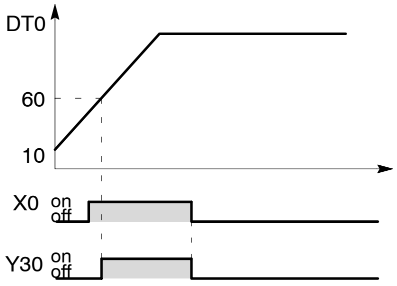
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A: Available

(*1) This cannot be used with the FP0/FP−e.
(*2) This cannot be used with the FP0/FP0R/FP−e/FPΣ/FP−X.
(*3) With the FPΣ, FP−X, FP0R, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FPΣ, FP−X, FP0R, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Compares the contents of data register DT0 with the constant K60 when X0 turns on. If DT0 ≧ K60 in the X0 on state, external output relay Y30 goes on. If DT0 < K60 or if X0 is in the off state, external output relay Y30 goes off.



Description

Compares the word data specified by S1 with the word data specified by S2 according to the comparison condition.

The **AN** instruction results in serial connection as the liaison contact when the comparison result is a specified status (=, <, >, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	S1 < S2	S1 = S2	S1 > S2
AN=	off	on	off
AN< >	on	off	on
AN>	off	off	on
AN> =	off	on	on
AN<	on	off	off
AN< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

Precautions concerning usage

Multiple **AND** comparison instructions **AN =, AN <>, AN >, AN >=, AN <, and AN <=** can be used consecutively.

If mixed with BCD or other type of data, the value will be regarded as negative when the most significant bit is 1 and a correct comparison may not be obtained. In this case, use an **F81 (BIN)** instruction or similar instruction to change the data to binary data before making the comparison.

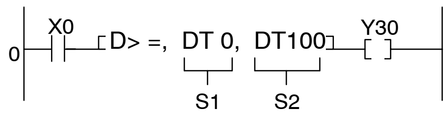
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

AND = 32–bit data comparison: AND equal
AND <> 32–bit data comparison: AND equal not
AND > 32–bit data comparison: AND larger
AND >= 32–bit data comparison: AND equal or larger
AND < 32–bit data comparison: AND smaller
AND <= 32–bit data comparison: AND equal or smaller

Outline Performs AND operation by comparing two double word data items with the comparison condition. The contact goes on or off depending on the result of the comparison. The contacts are connected in series.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST X 0
	1	AND >=
		DT 0
		DT 100
	10	OT Y 30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0/FP–e.
(*2) This cannot be used with the FP0/FP0R/FP–e/FPΣ/FP–X.
(*3) With the FPΣ, FP–X, FP0R, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FPΣ, FP–X, FP0R, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Compares the contents of data registers (DT1, DT0) with the data registers (DT101, DT100) when X0 turns on. If (DT1, DT0) $\geqq$ (DT101, DT100) in the X0 on state, the external output relay Y30 goes on.
If (DT1, DT0) < (DT101, DT100) or if X0 is in the off state, the external output relay Y30 goes off.

Description

Compares the double word data specified by S1 and S1+1 with the double word data specified by S2 and S2+1 according to the comparison condition.

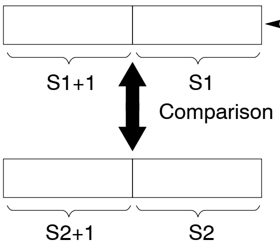
The **AND** instruction results in serial connection as the liaison contact when the comparison result is a specified status (=, <, >, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	(S1+1, S1) < (S2+1, S2)	**(S1+1, S1) = (S2+1, S2)**	**(S1+1, S1) > (S2+1, S2)**
AND=	off	on	off
AND< >	on	off	on
AND>	off	off	on
AND> =	off	on	on
AND<	on	off	off
AND< =	on	on	off

< > indicates $\neq$
>= indicates $\geqq$
<= indicates $\leqq$

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.



The specified data area and the following data area are handled together as 32–bit data.

Precautions concerning usage

Multiple **AND** comparisons instructions **AND =, AND <>, AND >, AND >=, AND <, and AND <=** can be used consecutively.

If mixed with BCD or other type of data, the value will be regarded as negative when the most significant bit is 1 and a correct comparison may not be obtained. In this case, use an **F83 (DBIN)** instruction or similar instruction to change the data to binary data before making the comparison.

Flag conditions

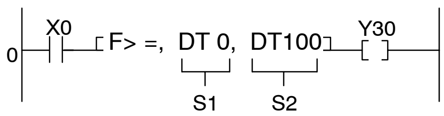
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

ANF = Floating point real number data comparison: AND equal
ANF <> Floating point real number data comparison: AND equal not
ANF > Floating point real number data comparison: AND larger
ANF >= Floating point real number data comparison: AND equal or larger
ANF < Floating point real number data comparison: AND smaller
ANF <= Floating point real number data comparison: AND equal or smaller

Availability
FP0R FP−X Ver 1.10 or more FPΣ 32k

Outline Performs AND operation by comparing two single precision real number data items with the comparison condition. The contact goes on or off depending on the result of the comparison. The contacts are connected in series.

Program example



Ladder Diagram	Boolean Address	Boolean Instruction
	0	ST X 0
	1	ANF>=
		DT 0
		DT 100
	10	OT Y 30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared
S2	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A2	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A

(*1) This cannot be used with the FPΣ/FP−X.
(*2) I0 to ID.

A: Available

* Index modification of a real number is not possible.

Explanation of example

Compares the real number value of data registers (DT0, DT1) with the real number value of data registers (DT100, DT101) when X0 turns on. If (DT0, DT1) ≧ (DT100, DT101) in the X0 on state, the external output relay Y30 goes on. If (DT0, DT1) < (DT100, DT101) or if X0 is in the off state, the external output relay Y30 goes off.

Description

Compares the real number data specified by S1 and S1 +1 with the real number data specified by S2 and S2+1 according to the comparison condition.

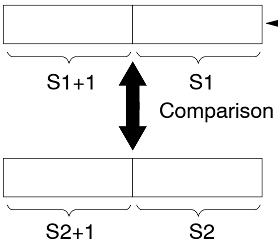
The **ANF** instruction results in serial connection as the liaison contact when the comparison result is a specified status (=, <, >, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	(S1+1, S1) < (S2+1, S2)	(S1+1, S1) = (S2+1, S2)	(S1+1, S1) > (S2+1, S2)
ANF=	off	on	off
ANF< >	on	off	on
ANF>	off	off	on
ANF> =	off	on	on
ANF<	on	off	off
ANF< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.



The specified data area and the following data area are handled together as single precision real number data.

Precautions concerning usage

Multiple **ANF** comparisons instructions **ANF =, ANF <>, ANF >, ANF >=, ANF <, and ANF** <= can be used consecutively.

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Flag conditions

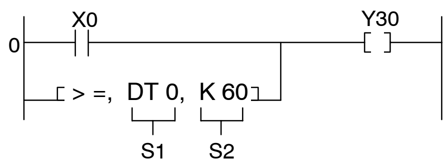
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The area specified using the index modifier exceeds the limit.
  - – If data other than real number data are specified in (S1+1, S1) and (S2+1, S2).

OR = 16–bit data comparison: OR equal
OR <> 16–bit data comparison: OR equal not
OR > 16–bit data comparison: OR larger
OR >= 16–bit data comparison: OR equal or larger
OR < 16–bit data comparison: OR smaller
OR <= 16–bit data comparison: OR equal or smaller

Outline Performs OR operation by comparing two word data items with the comparison condition. The contact goes on or off depending on the result of the comparison. The contacts are connected in parallel.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
0	ST	X	0
1	OR > =		
	DT		0
	K		60
6	OT	Y	30

S1	16-bit equivalent constant or 16-bit area to be compared
S2	16-bit equivalent constant or 16-bit area to be compared

Operands

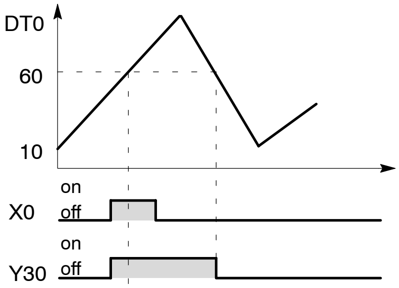
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A: Available

(*1) This cannot be used with the FP0/FP–e.
(*2) This cannot be used with the FP0/FP0R/FP–e/FPΣ/FP–X.
(*3) With the FPΣ, FP–X, FP0R, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FPΣ, FP–X, FP0R, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Y30 goes on when X0 is in the on state, or when DT0 ≧ K60. If DT0 < K60 and if X0 is in the off state, then Y30 goes off.



Description

Compares the word data specified by S1 with the word data specified by S2 according to the comparison condition.

The **OR** instruction results in parallel connection as the liaison contact when the comparison result is a specified status (=, <, >, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	S1 < S2	**S1 = S2**	**S1 > S2**
OR=	off	on	off
OR< >	on	off	on
OR>	off	off	on
OR> =	off	on	on
OR<	on	off	off
OR< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

Precautions concerning usage

The **OR** comparison instructions **OR =, OR <>, OR >, OR >=, OR <,** and **OR<=** are programmed from the bus line.

Multiple **OR** comparison instructions **OR =, OR <>, OR >, OR >=, OR <,** and **OR <=** can be used consecutively.

If mixed with BCD or other type of data, the value will be regarded as negative when the most significant bit is 1 and a correct comparison may not be obtained. In this case, use an **F81 (BIN)** instruction or similar instruction to change the data to binary data before making the comparison.

Flag conditions

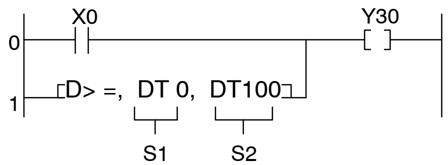
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

ORD = — 32–bit data comparison: OR equal
ORD <> — 32–bit data comparison: OR equal not
ORD > — 32–bit data comparison: OR larger
ORD >= — 32–bit data comparison: OR equal or larger
ORD < — 32–bit data comparison: OR smaller
ORD <= — 32–bit data comparison: OR equal or smaller

Outline Performs OR operation by comparing two double word data items with the comparison condition. The contact goes on or off depending on the result of the comparison. The contacts are connected in parallel.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
0	ST	X	0
1	ORD> =		
	DT		0
	DT		100
10	OT	Y	30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0/FP−e.
(*2) This cannot be used with the FP0/FP0R/FP−e/FPΣ/FP−X.
(*3) With the FPΣ, FP−X, FP0R, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FPΣ, FP−X, FP0R, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Compares the contents of data registers (DT1, DT0) with the data registers (DT101, DT100). When X0 turns on or if (DT1, DT0) ≧ (DT101, DT100), the external output relay Y30 goes on.
If (DT1, DT0) < (DT101, DT100) and if X0 is in the off state, the external output relay Y30 goes off.

Description

Compares the double word data specified by S1 and S1+1 with the double word data specified by S2 and S2+1 according to the comparison condition.

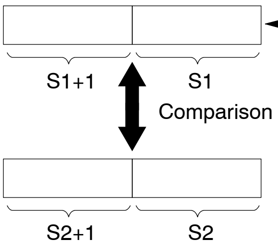
The **ORD** instruction results in parallel connection as the liaison contact when the comparison result is a specified status (=, >, <, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	(S1+1, S1)< (S2+1, S2)	(S1+1, S1) = (S2+1, S2)	(S1+1, S1) > (S2+1, S2)
ORD=	off	on	off
ORD< >	on	off	on
ORD>	off	off	on
ORD> =	off	on	on
ORD<	on	off	off
ORD< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.



The specified data area and the following data area are handled together as 32–bit data.

Precautions concerning usage

The **OR** comparison instructions **ORD =, ORD <>, ORD >, ORD >=, ORD <,** and **ORD <=** are programmed from the bus line.

Multiple **OR** comparison instructions **ORD =, ORD <>, ORD >, ORD >=, ORD<,** and **ORD <=** can be used consecutively.

If mixed with BCD or other type of data, the value will be regarded as negative when the most significant bit is 1 and a correct comparison may not be obtained. In this case, use an **F83 (DBIN)** instruction or similar instruction to change the data to binary data before making the comparison.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

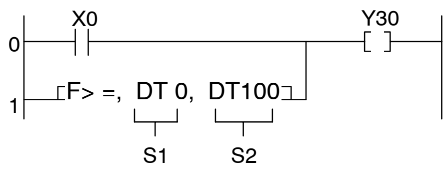
ORF = Floating point real number data comparison: OR equal
ORF <> Floating point real number data comparison: OR equal not
ORF > Floating point real number data comparison: OR larger
ORF >= Floating point real number data comparison: OR equal or larger
ORF < Floating point real number data comparison: OR smaller
ORF <= Floating point real number data comparison: OR equal or smaller

Availability
FP0R FP−X Ver 1.10 or more FPΣ 32k

Outline Performs OR operation by comparing two single precision real number data items with the comparison condition. The contact goes on or off depending on the result of the comparison. The contacts are connected in parallel.

Program example

Ladder Diagram



Address	Instruction		
0	ST	X	0
1	ORF> =		
	DT		0
	DT		100
10	OT	Y	30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared
S2	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A2	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A

(*1) This cannot be used with the FPΣ/FP−X.

(*2) I0 to ID.

A: Available

* Index modification of a real number is not possible.

Explanation of example

When X0 turns on or if (DT0, DT1) ≧ (DT100, DT101) by comparing the real number value of data registers (DT0, DT1) with the real number value of data registers (DT100, DT101), the external output relay Y30 goes on. If (DT0, DT1) < (DT100, DT101) and if X0 is in the off state, the external output relay Y30 goes off.

Description

Compares the real number data specified by S1 and S1 +1 with the real number data specified by S2 and S2+1 according to the comparison condition.

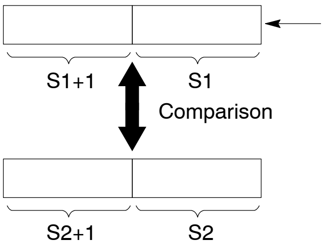
The **ORF** instruction results in parallel connection as the liaison contact when the comparison result is a specified status (=, >, <, etc.).

The result of the comparison operation is as follows:

Comparison instruction	Condition		
	(S1+1, S1)< (S2+1, S2)	(S1+1, S1) = (S2+1, S2)	(S1+1, S1) > (S2+1, S2)
ORF=	off	on	off
ORF< >	on	off	on
ORF>	off	off	on
ORF> =	off	on	on
ORF<	on	off	off
ORF< =	on	on	off

< > indicates ≠
>= indicates ≧
<= indicates ≦

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.



The specified data area and the following data area are handled together as single precision real number data.

Precautions concerning usage

The **OR** comparison instructions **ORF =, ORF <>, ORF >, ORF >=, ORF <,** and **ORF <=** are programmed from the bus line.

Multiple **OR** comparison instructions **ORF =, ORF <>, ORF >, ORF >=, ORF<,** and **ORF <=** can be used consecutively.

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data are specified in (S1+1, S1) and (S2+1, S2).

Chapter 3

High−level Instructions

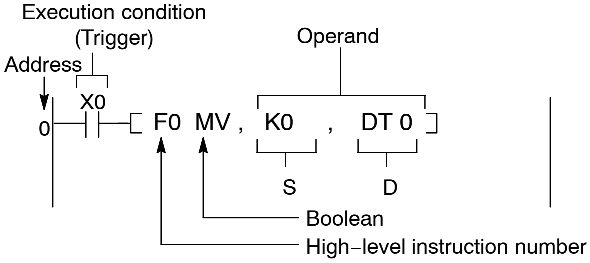
3.1 Composition of High-level Instructions

3.1.1 Composition

Each high-level instruction is composed of a high-level instruction number, boolean and operands.

Example: F0 (MV) instruction

The K0 (S) is copied to DT0 (D)



High−level instruction number
High−level instruction numbers are used for inputting the high-level instructions.

Boolean
Boolean indicate the processing content of each instruction.

Operand
Operands are used to specify the processing method and the storage area for processed data, etc. Operands are classified into three types: S (source), D (destination) and n (number).
The number of operands differ depending on the instruction.

Operand types

S (Source): Data which is to be processed or data which sets the processing method.

D (Destination): Location where result of processing is stored.

n (number): Numeric data which is to be processed or which sets the processing method.

Operands are specified using constants or memory areas (registers) as explained in section 1.3 and 1.4.
Refer to the explanations of the instructions as only certain memory areas (registers) and constants can be used with each instruction.

3.1.2 High-level Instruction Numbers and Program Input

High-level instruction numbers are assigned to high-level instructions. For example, the number assigned to the **MV** instruction (16-bit data transfer instruction) is 0 (**F0** or **P0**).

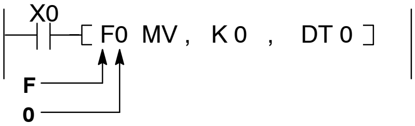
A high−level instruction is entered by entering its high-level instruction number.

A high−level instruction with the prefix “F” is executed in every scan while its execution condition (trigger) is in the on state.

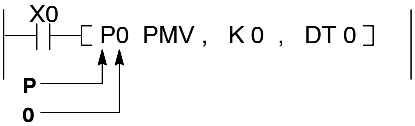
A high−level instruction with the prefix “P” is executed only when the leading edge of its execution condition (trigger) is detected.

For details about “F” and “P” type high−level instructions ☛ section 3.1.4

Input of “F” type high−level instruction



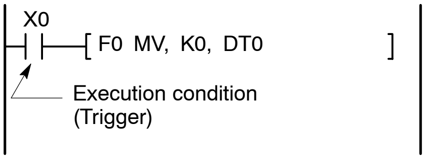
Input of “P” type high−level instruction



3.1.3 High-level Instruction and Execution Condition (Trigger)

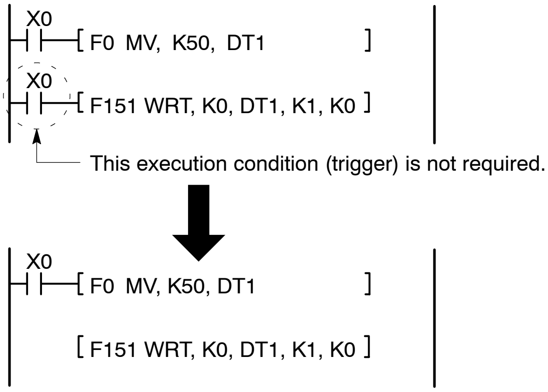
A high-level instruction is always used in a pair with its execution condition (trigger). When the operation result of the relay sequence circuit specified as the execution condition (trigger) is on, the high-level instruction is executed.

Example: When the execution condition (trigger) X0 is on, the F0 (MV) instruction is executed and K0 is transferred to DT0.



There is no need to program the same execution condition (trigger) many times when two or more high-level instructions are programmed consecutively with the same execution condition (trigger).

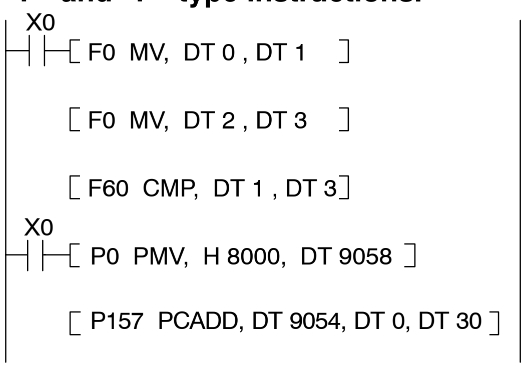
Example:



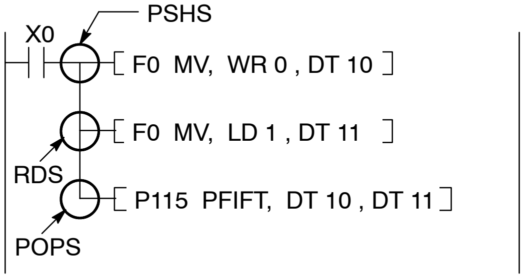
Precautions if omitting execution conditions (triggers)

If you need to program both “F” and “P” type high-level instructions using the same execution condition (trigger), proceed according to one of the following two examples.

Example 1: The same execution condition (trigger) is programmed twice for “F” and “P” type instructions.



Example 2: The execution condition (trigger) is programmed once using the PSHS, RDS and POPS instructions.

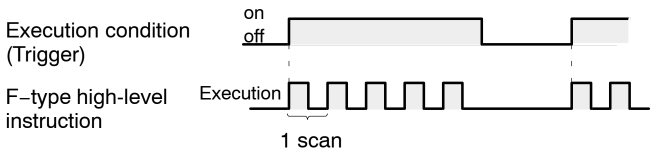


3.1.4 “F” and “P” Type High-level Instructions

For more high−level instructions, “F” and “P” types are available.

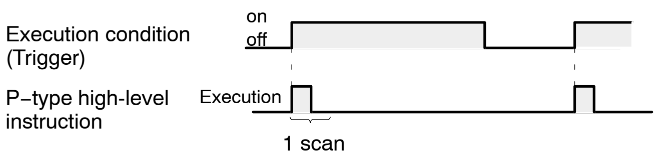
“F” type high-level instruction

While the execution condition (trigger) is on, the instruction is executed at each scan repeatedly.



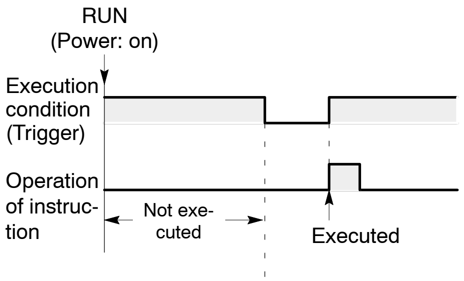
“P” type high-level instruction

The leading edge of the execution condition (trigger) is detected, and a single scan is executed.



As long as the execution condition (trigger) for the “P” type instruction continues to be on, the instruction is executed only at the rise of the condition, and is not subsequently executed.

If the mode is switched to the RUN mode, or the power supply is turned on in the RUN mode, the instruction is not executed in the first scan if the execution condition (trigger) for the “P” type instruction has been in effect from the beginning.



When you use the “P” type instruction with one of the following instructions that changes the order of the execution of instructions, be aware that the operation of the instructions will differ depending on the timing of their execution and their execution conditions (triggers).

– **MC** to **MCE** instructions
– **JP** to **LBL** instructions
– **F19 (SJP)** to **LBL** instructions
– **LOOP** to **LBL** instructions
– **CNDE** instruction
– Step ladder instructions
– Subroutine instruction

For detailed information ☛ section 4.3

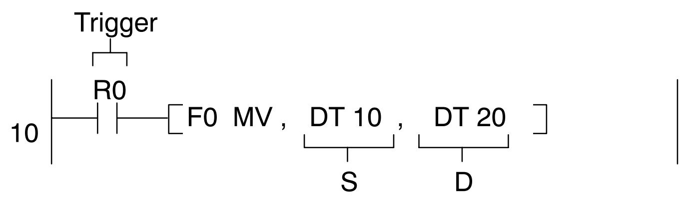
When combining the “P” type high–level instruction with an AND stack instruction or pop stack instruction, be careful that the programming is correct. For detailed information ☛ section 4.7

F0 (MV) P0 (PMV) 16-bit data move

Outline Copies 16-bit data to the specified 16-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P0 (PMV)" is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F 0 (MV)
	DT 10
	DT 20

S	16-bit equivalent constant or 16-bit area (source)
D	16-bit area (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

The contents of data register DT10 are copied to data register DT20 when trigger R0 turns on.

Description

The 16-bit data or 16-bit equivalent constant specified by S is copied to the area specified by D.

Reference

When using an FP0/FP−e/FP0R/FPΣ/FP−X high−speed counter: F0 (MV) page 3 − 437
When using an FP0/FP−e/FP0R/FPΣ/FP−X pulse output: F0 (MV) page 3 − 443

Application example

Example 1: Transfer K30 to timer set value area SV0 when R1 turns on.

```
    R1
 |--| |----[F0  MV,  K 30,  SV 0]          |
```

Example 2: Transfer the timer elapsed value EV0 to data register DT0 when R2 turns on.

```
    R2
 |--| |----[F0  MV,  EV 0,  DT 0]          |
```

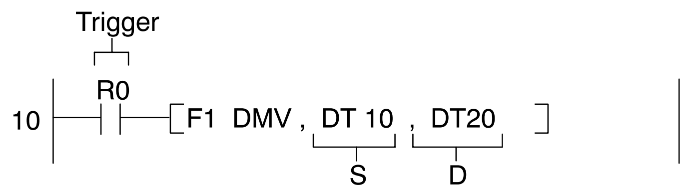
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F1 (DMV) / P1 (PDMV) 32-bit data move

Outline Copies 32-bit data to the specified 32-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P1 (PDMV)" is not available.

Program example

Ladder Diagram	Boolean: Address	Boolean: Instruction
	10 11	ST R 0 F 1 (DMV) DT 10 DT 20

S	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area for 32-bit area (destination)

Operands

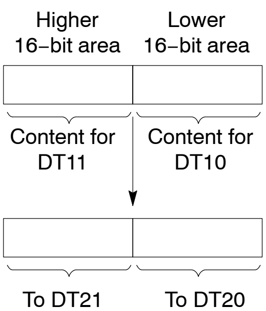
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

The contents of data registers DT11 and DT10 are copied to data registers DT21 and DT20 when trigger R0 turns on.



.... The content (16 bits) of DT10 and the content (16 bits) of DT11 are joined and treated as 32−bit data.

.... 32−bit data is separated into the higher 16−bit data and the lower 16−bit data, and stored in DT21 and DT20.

Description

The 32-bit data or 32-bit equivalent constant specified by S is copied to the 32-bit area specified by D.

When processing 32-bit data, the higher 16-bit areas (S+1, D+1) are automatically determined once the lower 16-bit areas (S, D) are specified.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

Reference

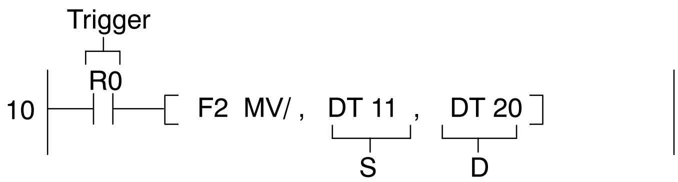
FP1/FP−M high−speed counter elapsed value: F1 (DMV) page ****
FP0/FP−e/FPΣ/FP−X high−speed counter pulse output elapsed value: F1 (DMV) page 3 − 449

F2 (MV/)
P2 (PMV/)

16-bit data invert and move

Outline Inverts 16-bit data and transfers it to the specified 16-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P2 (PMV/)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10 11	ST R 0 F 2 (MV/) DT 11 DT 20

S	16-bit equivalent constant or 16-bit area to be inverted (source)
D	16-bit area (destination)

Operands

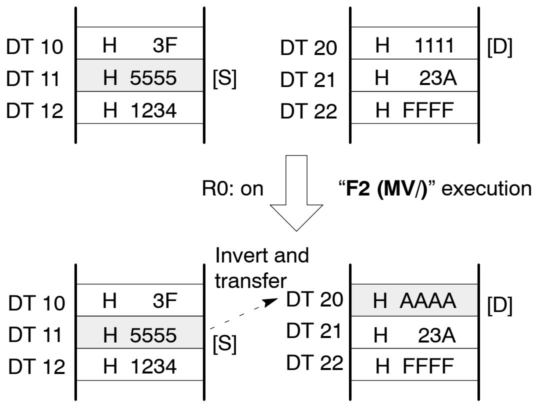
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

The contents of data register DT11 are inverted and transferred to data register DT20 when trigger R0 turns on.



Source [S]: H5555

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT11	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1

Invert and transfer

Destination [D]: HAAAA

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0

Description

The 16-bit data or 16-bit equivalent constant specified by S is inverted and transferred to the 16-bit area specified by D.

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	0 0 0 0	0 1 0 0	1 1 0 1	0 0 1 0
Hexadecimal	0	4	D	2

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	1 1 1 1	1 0 1 1	0 0 1 0	1 1 0 1
Hexadecimal	F	B	2	D

Flag conditions

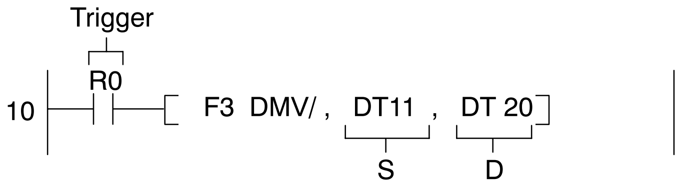
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F3 (DMV/) P3 (PDMV/) 32-bit data invert and move

Outline Inverts 32-bit data and transfers it to the specified 32-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P3 (PDMV/)" is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F 3 (DMV/)
	DT 11
	DT 20

S	32-bit equivalent constant or lower 16-bit area of 32-bit data to be inverted (source)
D	Lower 16-bit area of 32-bit data (destination)

Operands

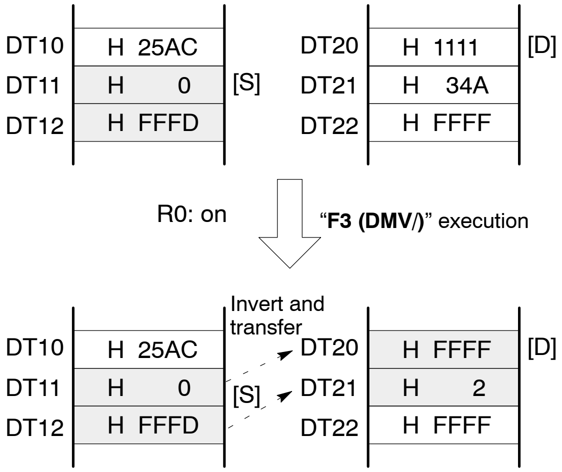
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

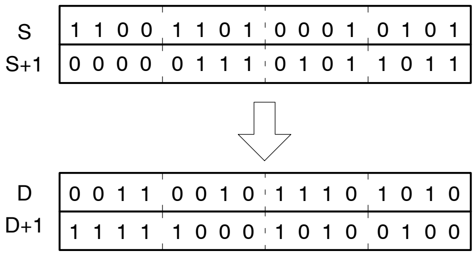
Explanation of example

The contents of data registers DT12 and DT11 are inverted and transferred to data registers DT21 and DT20 when trigger R0 turns on.



Description

The 32-bit data or 32-bit equivalent constant specified by S is inverted and transferred to the 32-bit area specified by D.



When processing 32-bit data, the higher 16-bit areas (S+1, D+1) are automatically determined once the lower 16-bit areas (S, D) are specified.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F4 (GETS)
P4 (PGETS)

Reading of head word No. of the specified slot.

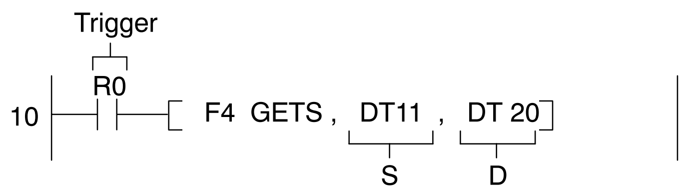
Outline

The head word No. of the specified slot is read.

This function is available from FP2/FP2SH Ver. 1.50 or later.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 4 (GETS)
	DT 11
	DT 20

S	Specification of slot numbers
D	Area in which the WX and WY numbers stored (32 bits)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

Explanation of example

The number of WX and WY for the slot specified by S is read, and set in [D, D+1].

D	Head number of WX of specified slot
D+1	Head number of WY of specified slot

When the unit is with X only, the same value is stored for the head number of WY.
When the unit is with Y only, the same value is stored for the head number of WX.
When the unit without input/output is specified, the same value is stored in D and D+1.

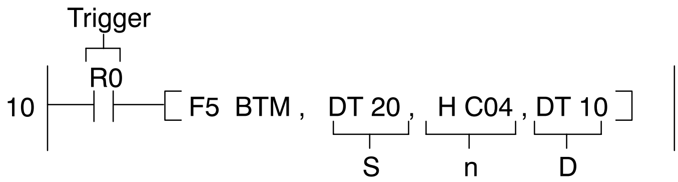
Flag conditions

- Error flag (R9007)(R9008):
  - It turns on, when the specified address using the index modifier exceeds a limit.
  - It turns on, when the number other than 0 to 31 is specified for the slot number.

F5 (BTM) / P5 (PBTM) Bit data move

Outline Copies bit data of one 16-bit area to the specified bit of another 16-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P5 (PBTM)” is not available.

Program example



Boolean	
Address	Instruction
10	ST R 0
11	F 5 (BTM)
	DT 20
	H C04
	DT 10

S	16-bit equivalent constant or 16-bit area (source)
n	16-bit equivalent constant or 16-bit area (specifies source and destination bit positions)
D	16-bit area (destination)

Operands

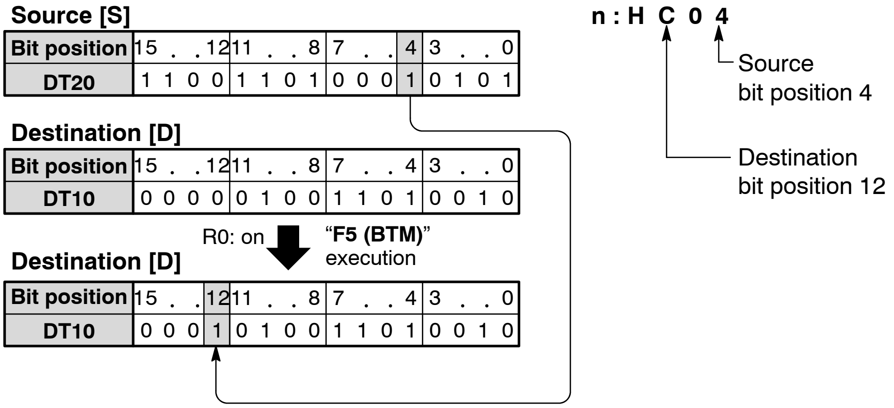
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

The data at bit position 4 in data register DT20 is copied to bit position 12 in data register DT10 when trigger R0 turns on.



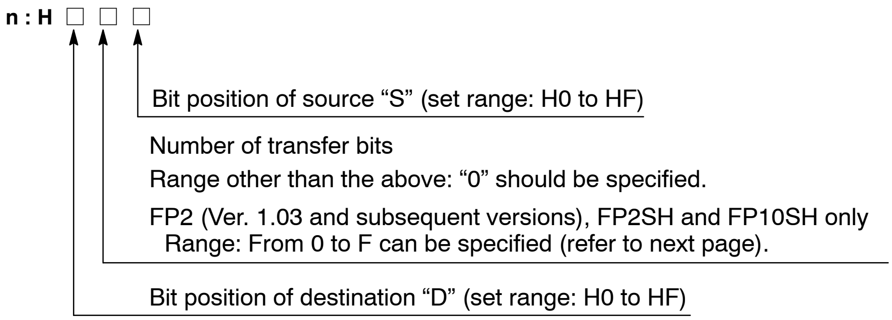
Description

A single bit in the 16-bit data or the 16-bit equivalent constant specified by S is copied to a bit of the 16-bit area specified by D, as specified by n.

With the FP2SH and FP10SH, it is possible to transfer the contents of multiple bits as a single transfer.

How to specify n

The "n" specifies the source and destination bit positions using hexadecimal data as follows:



Bit position specification for S and D

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Set value	HF	HE	HD	HC	HB	HA	H9	H8	H7	H6	H5	H4	H3	H2	H1	H0

For example, when bit position 10 is specified, "HA" should be specified.

If bit position 4 of S is being transferred to bit position 12 of D, n = HC04.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

Transferring multiple bits [this can only be executed with FP0R, FPΣ, FP−X, FP2 (Ver. 1.03 and subsequent versions), FP2SH, and FP10SH]

With the FP2, FP2SH and FP10SH, if the number of bits to be transferred is specified for n, the specified number of bits is transferred in sequential order, starting from the position specified by S, to destination, starting from the position specified by D.

Up to 16 bits can be transferred. The number of bits to be transferred should be specified as a hexadecimal value. The range is from 0 to F (1 bit to 16 bits).

No. of bits transferred	Setting (n)
1 bit	H□0□
2 bits	H□1□
3 bits	H□2□
4 bits	H□3□
5 bits	H□4□
6 bits	H□5□
7 bits	H□6□
8 bits	H□7□
9 bits	H□8□
10 bits	H□9□
11 bits	H□A□
12 bits	H□B□
13 bits	H□C□
14 bits	H□D□
15 bits	H□E□
16 bits	H□F□

Example: **When two bits are being transferred (n = H□1□)**
Two bits sent, starting from bit position 5 of S to bit position 10 of D........n = HA15

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
S	0 0 1 0	0 1 0 1	1 1 0 0	1 1 0 1

Two bits, starting from bit position 5

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
D	1 1 1 0	1 1 0 1	0 0 0 1	1 0 0 1

“**F5 (BTM)**” execution

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
D	1 1 1 0	1 0 0 1	0 0 0 1	1 0 0 1

Bit positions 5 and 6 of S are transferred to bit position 10 and 11 of D

If “0” is specified as the number of bits to be transferred, the specified one bit is transferred.

If the specified range extends beyond the area of S, the contents of the part extending beyond the area are transferred as “0”.

Example: When four bits starting from bit position 14 of S are transferred to bit position 2 of D...n = H23E

Bit position	. .	15 . . 12	11 . . 8	7 . . 4	3 . . 0
S	0 0	1 1 0 0	1 1 0 1	0 0 0 1	1 0 1 0

Four bits, starting from bit position 14

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
D	0 0 1 0	0 0 0 1	1 0 1 1	1 0 0 1

“**F5 (BTM)**” execution

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
D	0 0 1 0	0 0 0 1	1 0 0 0	1 1 0 1

Bit positions 14 and 15 of S are transferred to bit positions 2 and 3 of D. “0” is stored in bit positions 4 and 5 of D.

If the specified range extends beyond the area of D, the contents of the part extending beyond the area are not transferred. Data is not written to the next address.

Example: Six bits starting from bit position 6 of S are transferred to bit position 12 of D...n = HC56

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
S	0 0 0 0	0 1 0 1	1 1 0 1	1 1 1 1

Six bits, starting from bit position 6

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
D	0 0 0 0	1 0 0 1	1 1 1 1	0 0 1 1

“**F5 (BTM)**” execution

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
D	0 1 1 1	1 0 0 1	1 1 1 1	0 0 1 1

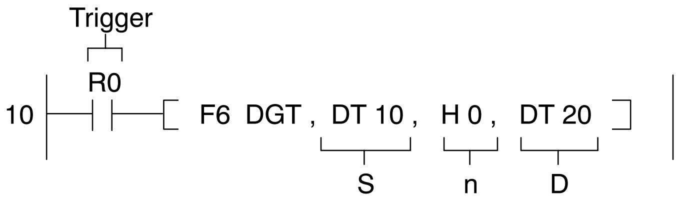
From among bit positions 6 to 11 of S, bit positions 6 to 9 are transferred to bit positions 12 to 15 of D (the contents of bit positions 10 and 11 of S are ignored).

F6 (DGT) P6 (PDGT) Hexadecimal digit data move

Outline Copies hexadecimal digits at one 16-bit area to the specified digit position in another 16-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P6 (PDGT)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 6 (DGT)
	DT 10
	H 0
	DT 20

S	16-bit equivalent constant or 16-bit area (source)
n	16-bit equivalent constant or 16-bit area (specifies source and destination hexadecimal digit position and number of hexadecimal digits)
D	16-bit area (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Hexadecimal digit position	3	2	1	0
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 0 0 0	0 0 0 1	0 1 0 0	1 0 0 1

The lower four bits in the data register DT10 is copied.

Hexadecimal digit position	3	2	1	0
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	1 0 0 0	1 0 1 0	1 0 0 1

In this example, the upper 12 bits of DT20 do not change.

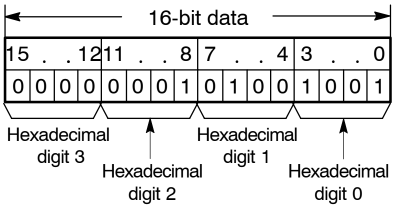
Description

The hexadecimal digits in the 16-bit data or in the 16-bit equivalent constant specified by S are copied to the 16-bit area specified by D, as specified by n.

Digits

Digits are units of 4 bits used when handling data.

With this instruction, 16–bit data is separated into four digits. The digits are called in order hexadecimal digit 0, digit 1, digit 2 and digit 3, beginning from the least significant four bits



Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

How to specify n

n specifies the ③ source hexadecimal digit position, the ② number of digits and the ① destination hexadecimal digit position to be copied using hexadecimal data as follows:

n : H □ □ □

③ Source: Starting hexadecimal digit position
H0: Hexadecimal digit 0
H1: Hexadecimal digit 1
H2: Hexadecimal digit 2
H3: Hexadecimal digit 3

② Number of hexadecimal digits to be copied
H0: Copies 1 hexadecimal digits (4 bits)
H1: Copies 2 hexadecimal digits (8 bits)
H2: Copies 3 hexadecimal digits (12 bits)
H3: Copies 4 hexadecimal digits (16 bits)

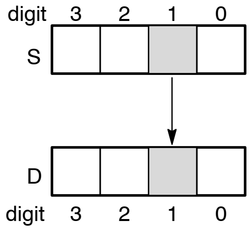
① Destination: Starting hexadecimal digit position
H0: Hexadecimal digit 0
H1: Hexadecimal digit 1
H2: Hexadecimal digit 2
H3: Hexadecimal digit 3

If the value for ①, ② and ③ is 0, such as “H000” in the example program on the previous page, use the short form, “H0.”

Examples of hexadecimal digit copy

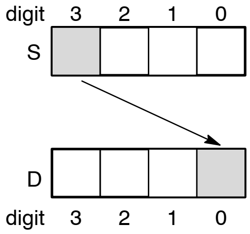
The following patterns of digit transfer are possible based on the specification of n.

(1) When hexadecimal digit 1 of the source is copied to hexadecimal digit 1 of the destination:



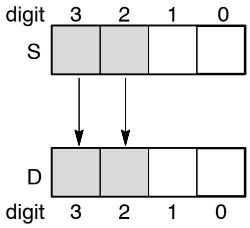
Specify n: H 1 0 1

(2) When hexadecimal digit 3 of the source is copied to hexadecimal digit 0 of the destination:



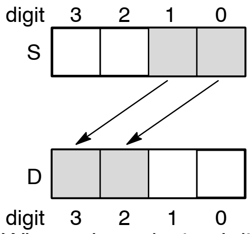
Specify n: H 0 0 3 (Short form: H3)

(3) When multiple hexadecimal digits (hexadecimal digits 2 and 3) of the source are copied to multiple hexadecimal digits (hexadecimal digits 2 and 3) of the destination:



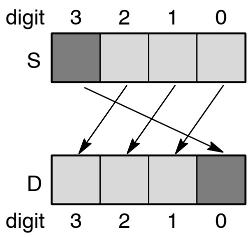
Specify n: H 2 1 2

(4) When multiple hexadecimal digits (hexadecimal digits 0 and 1) of the source are copied to multiple hexadecimal digits (hexadecimal digits 2 and 3) of the destination:



Specify n: H 2 1 0

(5) When 4 hexadecimal digits (hexadecimal digits 0 to 3) of the source are copied to 4 hexadecimal digits (hexadecimal digits 0 to 3) of the destination:



Specify n: H 1 3 0

F7 (MV2) / P7 (PMV2) Two 16-bit data move

Outline Copies two 16-bit data to the specified 32-bit area.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P7 (PMV2)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10	ST R 0
	11	F7 (MV2)
		DT 10
		DT 20
		DT 30

S1	16-bit equivalent constant or 16-bit area (source)
S2	16-bit equivalent constant or 16-bit area (source)
D	Lower 16-bit area for 32-bit area (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f	
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

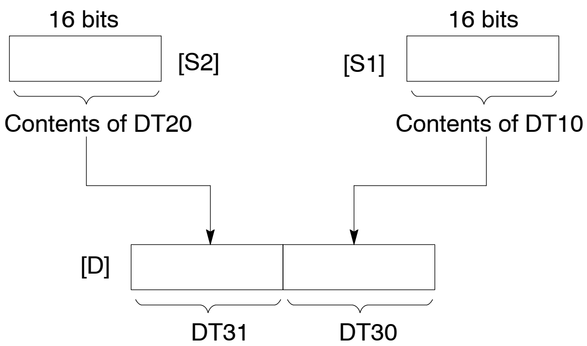
(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

The contents of data register DT10 are copied to data register DT30 when trigger R0 turns on.

The contents of data register DT20 are copied to data register DT31 when trigger R0 turns on.



Description

The two 16-bit data or two 16-bit equivalent constant specified by S1 and S2 is copied to the 32-bit area specified by D when the trigger turns on.

Related instruction

To copy three 16-bit data, use the **F190 (MV3)** instruction.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

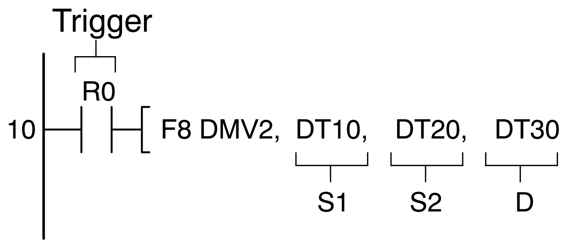
F8 (DMV2)
P8 (PDMV2)

Two 32-bit data move

Outline Copies two 32-bit data to the specified 64-bit area.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P8 (PDMV2)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F8	(DMV2)	
	DT		10
	DT		20
	DT		30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 64-bit area (destination)

Operands

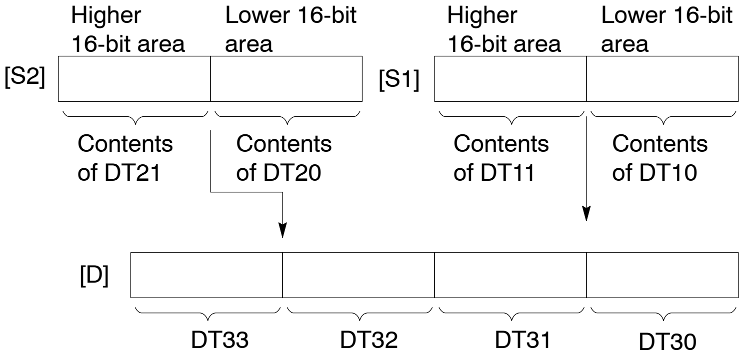
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

The contents of data register DT11 and DT10 and the contents of data register DT21 and DT20 are copied to data registers DT33, DT32, DT31 and DT30 when trigger R0 turns on.



Description

The two 32-bit data or two 32-bit equivalent constant specified by S1 and S2 is copied to the 64-bit area (D+3, D+2, D+1 and D) specified by D when the trigger turns on.

Related instruction

To copy three 32-bit data, use the **F191 (DMV3)** instruction.

Flag conditions

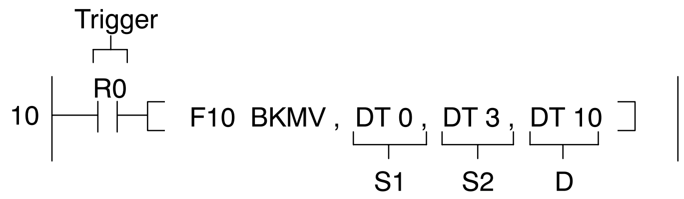
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F10 (BKMV) P10 (PBKMV) Block move

Outline Copies block data to the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 10 (BKMV)
	DT 0
	DT 3
	DT 10

S1	Starting 16-bit area (source)
S2	Ending 16-bit area (source)
D	Starting 16-bit area (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	I	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

A: Available
N/A: Not Available

Description

The data block specified by S1 and S2 is copied to the block starting from the 16-bit area specified by D.

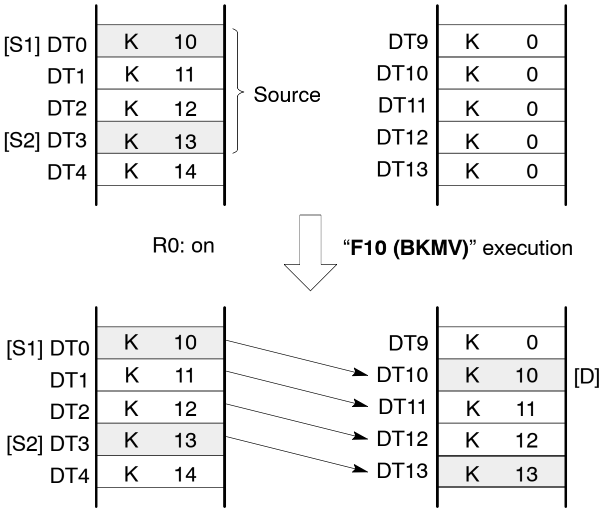
Precautions during programming

The starting area S1 and ending area S2 should be the same type of operand.

The number of the lower address should be specified by S1, and the number of the higher address should be specified by S2. If S1 is specified as higher than S2 and the instruction is executed, an operation error will occur.

Explanation of example

The data of data register “DT0 to DT3” is copied to the data registers “DT10 to DT13” when trigger R0 turns on.

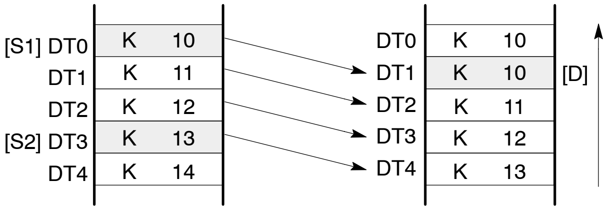


Precautions if the same type of memory area is specified for S1, S2, and D

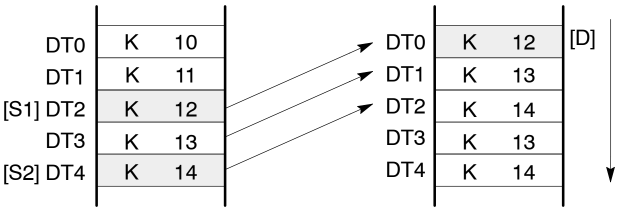
The instruction is not executed if the address and type of memory area is the same for S1 and D.

If the block being transferred overlaps the transfer destination, the transfer results will be overwritten.

If S1 < D, the source data is copied starting from the higher address to the lower address in order (DT4 → DT3 → DT2 → DT1).



If S1 > D, the source data is copied starting from the lower address to the higher address in order (DT0 → DT1 → DT2).



Flag conditions

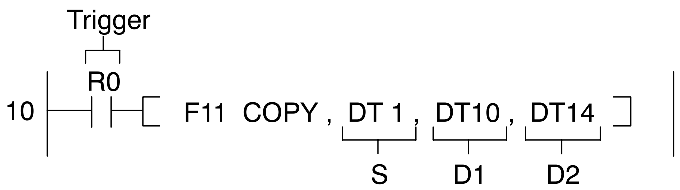
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2
  - The data block to be copied exceeds the limit of the destination area.

F11 (COPY) P11 (PCOPY) Block copy

Outline Copies the specified 16-bit data to a block with one or more 16-bit areas.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 11 (COPY)
	DT 1
	DT 10
	DT 14

S	16-bit equivalent constant or 16-bit area (source)
D1	Starting 16-bit area (destination)
D2	Ending 16-bit area (destination)

Operands

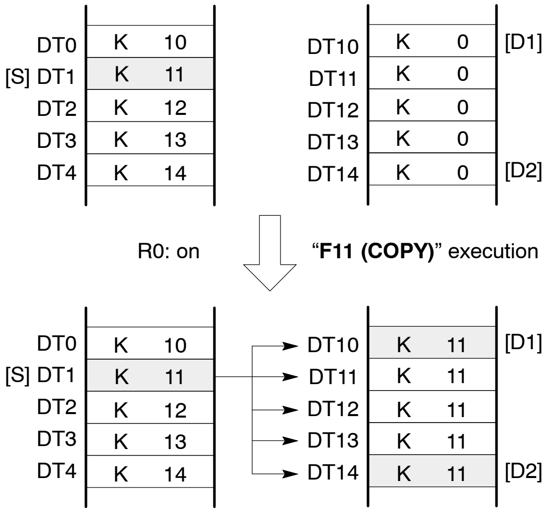
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

The contents of data register DT0 are copied to the block ranging from data register DT10 to DT14 when trigger R0 turns on.



Description

The 16-bit equivalent constant or 16-bit area specified by S is copied to all 16-bit areas of the block specified by D1 and D2.

Precautions during programming

The starting area D1 and ending area D2 should be the same type of operand.

The area of the lower address for the block being copied should be specified by D1, and the higher address should be specified by D2. If D1 is specified as higher than D2 and the instruction is executed, an operation error will occur.

When the same number as D1 and D2 is specified, the 16−bit data will be copied to the 16−bit area of that number.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - D1 > D2.

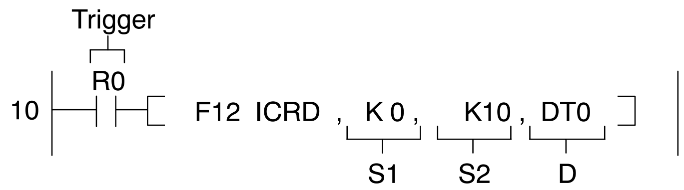
F12 (ICRD) Data read from EEPROM

Availability
FP0/FP−e

Outline Reads data from the EEPROM area.

Program example

Ladder Diagram



Boolean Non-ladder Address	Instruction		
10	ST	R	0
11	F 12		(ICRD)
	K		0
	K		10
	DT		0

S1	Constant for specifying the starting address of EEPROM (for source data)
S2	32−bit equivalent constant or lower 16−bit area of 32−bit data for specifying number of words to be read
D	Starting 16−bit area for storing data read from EEPROM (for destination)

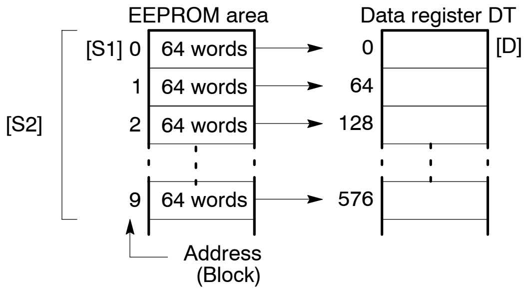
Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
S1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
S2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
D	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

A: Available
N/A: Not Available

Explanation of example

10 blocks of data stored in blocks 0 to 9 of the EEPROM are transferred to data registers DT0 to DT639 when execution condition (trigger) R0 turns on.



Description

S2 blocks of data stored in the EEPROM starting from S1 are transferred into the data register specified by D. At this time, the transferred data is handled in units of 1 block/64 words.

Precautions during programming

Values that can be specified by S1, S2 and D

Type	Memory area		
	S1	S2	D
FP0 C10, C14, C16, FP−e	K0 to K9	K1 to K10	DT0 to DT1595
FP0 C32, SL1	K0 to K95	K1 to K96	DT0 to DT6080
FP0 T32	K0 to K255	K1 to K256	DT0 to DT16320

Volume of data held in the EEPROM

Type	Volume that can be read
FP0 C10, C14, C16, FP−e	640 words
FP0 C32, SL1	6,144 words
FP0 T32	16,384 words

Because the initial data in the EEPROM is not fixed, caution is required when reading data that has not been written to the EEPROM.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The address specified by S1 does not exist in the EEPROM area.
  - The area specified by S2 exceeds the limit of the EEPROM area.
  - The area is exceeded when blocks specified by D and subsequent parameters are transferred.

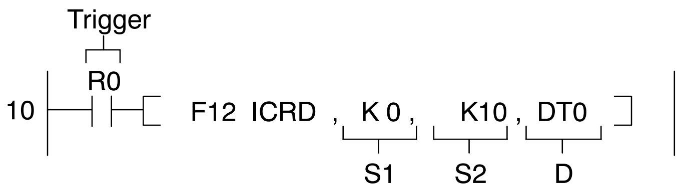
Availability
FPΣ/FP−X/FP0R

F12 (ICRD) Data read from F–ROM

Outline Reads data from the F–ROM area.

Program example

Ladder Diagram



Boolean Non-ladder Address	Instruction		
10	ST	R	0
11	F 12		(ICRD)
	K		0
	K		10
	DT		0

S1	Constant for specifying the starting address of F–ROM (for source data)
S2	32−bit equivalent constant or lower 16−bit area of 32−bit data for specifying number of words to be read
D	Starting 16−bit area for storing data read from F–ROM (for destination)

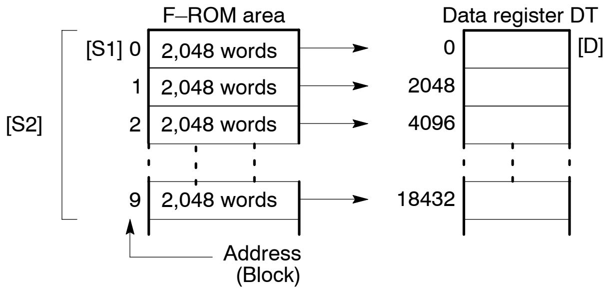
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	
S1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
S2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
D	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A

A: Available
N/A: Not Available

Explanation of example

10 blocks of data stored in blocks 0 to 9 of the F–ROM are transferred to data registers DT0 to DT20479 when execution condition (trigger) R0 turns on.



Description

S2 blocks of data stored in the F–ROM starting from S1 are transferred into the data register specified by D. At this time, the transferred data is handled in units of 1 block (2,048 words).

Precautions during programming

Values that can be specified by S1, S2 and D

Type	Memory area		
	S1	S2	D
FPΣ, FP−X, FP0R	K0 to K15	K1 to K16	DT0 to DT30720 (FP−X C14, FP0R C10, 14, 16: DT0 to DT12284)

Volume of data held in the F–ROM

Type	Volume that can be read
FPΣ, FP−X, FP0R	32,765 words (FP−X C14, FP0R C10, 14, 16: 12285 words)

Because the initial data in the F–ROM is not fixed, caution is required when reading data that has not been written to the F–ROM.

The initial value of F−ROM of FP0R can be cleared to 0 when the programs are all deleted with a tool software.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The address specified by S1 does not exist in the F–ROM area.
  - The area specified by S2 exceeds the limit of the F–ROM area.
  - The area is exceeded when blocks specified by D and subsequent parameters are transferred.

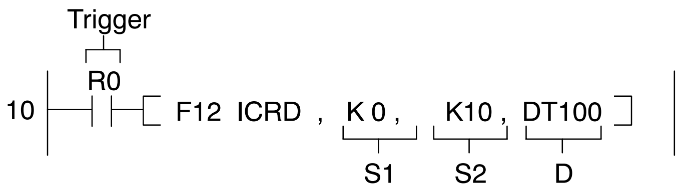
F12 (ICRD) P12 (PICRD) Data read from IC card

Availability
FP2SH/FP10SH

Outline Reads data from the expansion memory area of the IC card.

Program example

Ladder Diagram



Boolean Non-ladder Address	Instruction	
10	ST R	0
11	F 12	(ICRD)
	K	0
	K	10
	DT	100

S1	Constant for specifying the starting address of IC card expansion memory (for source data)
S2	32−bit equivalent constant or lower 16−bit area of 32−bit data for specifying number of words to be read
D	Starting 16−bit area for storing data read from IC card (for destination)

Operands

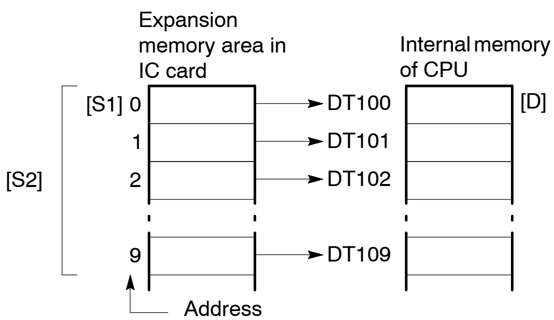
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX (*1)	IY (*2)	K	H	
S1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) This is I0 to IC.
(*2) This is ID.

A: Available
N/A: Not Available

Explanation of example

10 words of data stored in addresses 0 to 9 of the IC card expansion memory area are transferred to data registers DT100 to DT109 when trigger R0 turns on.



Description

S2 words of data stored in the IC card expansion memory area starting from S1 are transferred into the CPU memory location specified by D.

Precautions during programming

The values available for S1 and S2 vary depending on the size of the IC card expansion memory area.

When using an nkB IC card

Value that can be specified for S2: 1 to $\left(\frac{n \times 1024}{2} - 1\right)$

Value that can be specified for S1: 0 to ([S2]−1)

n	S1	S2
256 k	K131070	K131071 (H1FFFF)
512 k	K262142	K262143 (H3FFFF)
1 M	K524286	K524287 (H7FFFF)
2 M	K1048574	K1048575 (HFFFFF)

Note: When using as remaining DOS formatted mkB expansion memory:

S2: 1 to $\left(\frac{m \times 1024}{2}\right)$

S1: 0 to [S2]

Flag conditions

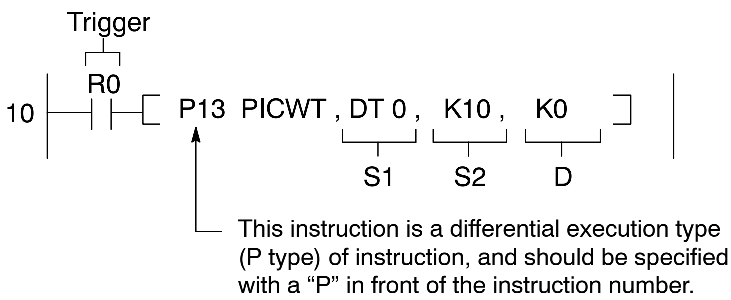
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - No IC card is installed in the CPU.
  - The IC card access enable switch is set to off (disabled).
  - No expansion memory area is found on the IC card.
  - The address specified by S1 does not exist in the expansion memory area of the IC card.
  - The area specified by S2 exceeds the limit of the expansion memory area of the IC card.
  - The area is exceeded when blocks specified by D and subsequent parameters are transferred.

Step	Availability
11	FP0 V2.0 or more/FP−e

P13 (PICWT) Data write to EEPROM

Outline Writes data to the EEPROM area.

Program example

Ladder Diagram	Boolean Non-ladder Address	Boolean Non-ladder Instruction
	10 11	ST R 0 P 13 (PICWT) DT 0 K 10 K 0

S1	Starting 16-bit area for storing source data
S2	32−bit equivalent constant or lower 16−bit area of 32−bit data for specifying number of words to be write
D	Starting address (constant) of EEPROM area for storing received data (for destination)

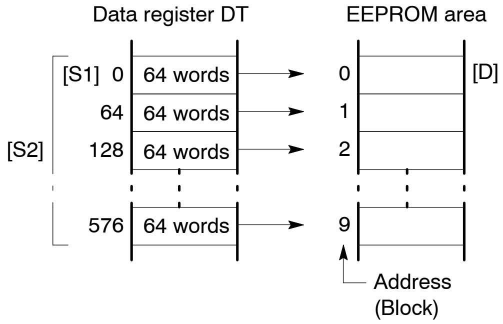
Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
S1	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
S2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A

A: Available
N/A: Not Available

Explanation of example

10 blocks (640 words) of data stored in data registers DT0 to DT576 are transferred to blocks 0 to 9 in the EEPROM area when execution condition (trigger) R0 turns on.



Description

S2 blocks of data stored in the data register starting from S1 are transferred into the EEPROM area specified by D. At this time, the transferred data is handled in units of 1 block/64 words.

Precautions during programming

Values that can be specified by S1, S2 and D

Type	Memory area		
	S1	S2	D
FP0 C10, C14, C16, FP−e	DT0 to DT1595	K1 to K10	K0 to K9
FP0 C32, SL1	DT0 to DT6080	K1 to K96	K0 to K95
FP0 T32	DT0 to DT16320	K1 to K256	K0 to K255

Volume of data that can be held in the EEPROM

Type	Volume that can be read
FP0 C10, C14, C16, FP−e	640 words
FP0 C32, SL1	6,144 words
FP0 T32	16,384 words

Data can be written to the EEPROM up to 10,000 times.

In order to prevent this instruction from being written to the EEPROM numerous times through erroneous programming, it has been set up as a differential execution type of instruction (P13). When setting up the program, however, please make sure that this instruction is not written to the EEPROM numerous times.

When the instruction is executed, the operation execution time will be approximately 5 ms longer for block (64 words).

This instruction should not be used in interrupt programs.

If the FP0R is used as the FP0, the execution time will be longer. (FP0 compatibility mode)

Comparative chart of execution time of FP0R in FP0 mode and FP0

No. of specified blocks	Execution time of FP0 (Unit: ms)	Execution time of FP0R in FP0 compatibility mode (Unit: ms)
1	5	100
2	10	100
4	20	100
8	40	100
16	80	100
32	160	100
33	165	200
40	205	200

Note that the execution time of the FP0R in FP0 mode is longer as shown in the above chart.

Flag conditions

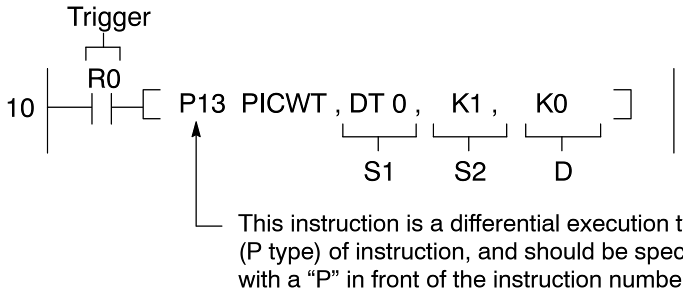
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The number specified by S1 does not exist in the memory area.
  - The area specified by S2 exceeds the limit of the memory area.
  - The area is exceeded when blocks specified by D and subsequent parameters are transferred.

Availability
FPΣ/FP-X/FP0R

P13 (PICWT) Data write to F–ROM

Outline Writes data to the F–ROM area.

Program example

Ladder Diagram	Boolean Non-ladder Address	Boolean Non-ladder Instruction
	10	ST R 0
	11	P 13 (PICWT)
		DT 0
		K 1
		K 0

S1	Starting 16-bit area for storing source data
S2	32−bit equivalent constant or lower 16−bit area of 32−bit data for specifying number of words to be write
D	Starting address of F–ROM area for storing received data (for destination)

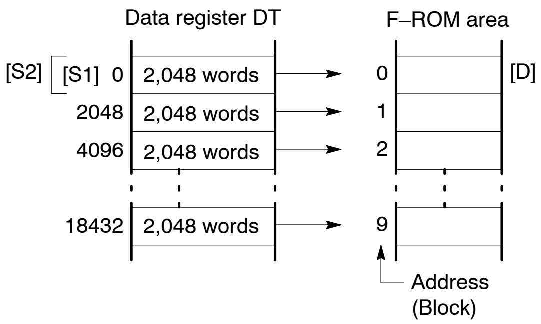
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	
S1	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A
S2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A

A: Available
N/A: Not Available

Explanation of example

1 block (2,048 words) of data stored in data registers DT0 is transferred to block 0 in the F–ROM area when execution condition (trigger) R0 turns on.



Description

S2 block of data stored in the data register starting from S1 is transferred into the F–ROM area specified by D. At this time, the transferred data is handled in units of 1 block (2,048 words).

Precautions during programming

Values that can be specified by S1, S2 and D

Type	Memory area		
	S1	S2	D
FPΣ, FP−X, FP0R	DT0 to DT30720 (FP−X C14, FP0R C10, 14, 16: DT0 to DT12284)	K1	K0 to K15

Volume of data that can be held in the F–ROM

Type	Volume that can be read
FPΣ, FP−X, FP0R	32,765 words (FP−X C14, FP0R C10, 14, 16: 12285 words)

Data can be written to the F–ROM up to 10,000 times.

In order to prevent this instruction from being written to the F–ROM numerous times through erroneous programming, it has been set up as a differential execution type of instruction (P13). When setting up the program, however, please make sure that this instruction is not written to the F–ROM numerous times.

The number of blocks that can be written to is only one. Also, a maximum time of approximately 100 ms is required for instruction execution. To write to multiple blocks, first divide into multiple scans.

This instruction should not be used in interrupt programs.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The number specified by S1 does not exist in the memory area.
  - The area specified by S2 exceeds the limit of the memory area.
  - The area is exceeded when blocks specified by D and subsequent parameters are transferred.

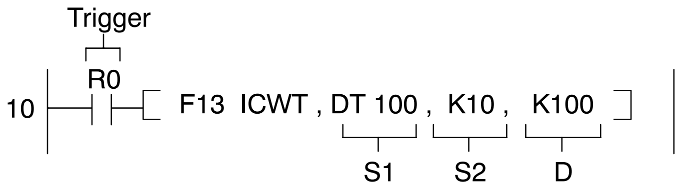
F13 (ICWT) P13 (PICWT)

Data write to IC card

Availability
FP2SH/FP10SH

Outline Writes data to the expansion memory area in the IC card.

Program example

Ladder Diagram	Boolean Non-ladder Address	Boolean Non-ladder Instruction
	10 11	ST R 0 F 13 (ICWT) DT 100 K 10 K 100

S1	Starting 16-bit area for storing source data
S2	32−bit equivalent constant or lower 16−bit area of 32−bit data for specifying number of words to be write
D	Starting address (constant) of IC card expansion memory area for storing received data (for destination)

Operands

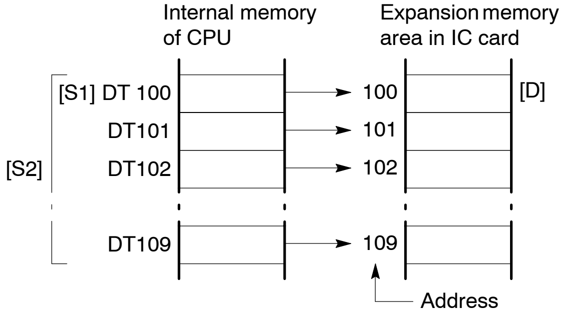
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX (*1)	IY (*2)	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A

(*1) This is I0 to IC.
(*2) This is ID.

A: Available
N/A: Not Available

Explanation of example

10 words of data stored in data registers DT100 to DT109 are transferred to addresses 100 to 109 in the expansion memory area in the IC card when trigger R0 turns on.



Description

S2 words of data stored in the CPU starting from S1 are transferred into the expansion memory area in the IC card specified by D.

The **F13 (ICWT)**/**P13 (PICWT)** instruction can be executed only in the expansion memory area of an SRAM−type IC card.

Precautions during programming

The values available for D vary depending on the size of expansion memory area in the IC card.

When using an nkB IC card

Value that can be specified for S2: 1 to $\left(\frac{n \times 1024}{2} - 1\right)$

Value that can be specified for S1: 0 to ([S2]−1)

n	S1	S2
256 k	K131070	K131071 (H1FFFF)
512 k	K262142	K262143 (H3FFFF)
1 M	K524286	K524287 (H7FFFF)
2 M	K1048574	K1048575 (HFFFFF)

Note: When using as remaining DOS formatted mkB expansion memory:

S2: 1 to $\left(\frac{m \times 1024}{2}\right)$

S1: 0 to [S2]

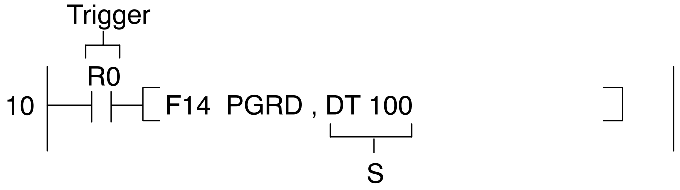
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - No IC card is installed in the CPU.
  - The IC card access enable switch is set to off (disabled).
  - Write protect is in effect on the card side.
  - The card is a FLASH−EEPROM type.
  - No expansion memory area is found on the IC card.
  - The area specified by S2 exceeds the limit of the expansion memory area of IC card.
  - The area is exceeded when blocks specified by D and subsequent parameters are transferred.

F14 (PGRD) P14 (PPGRD) Program read from IC card

Outline Reads a program from the IC card and executes it.

Program example

Ladder Diagram	Boolean Non-ladder	
	Address	Instruction
	10 11	ST R 0 F 14 (PGRD) DT 100

S	Starting 16-bit area (max. 4 words of data) for storing file name (max. 8 letters) in the ASCII format.

Operands

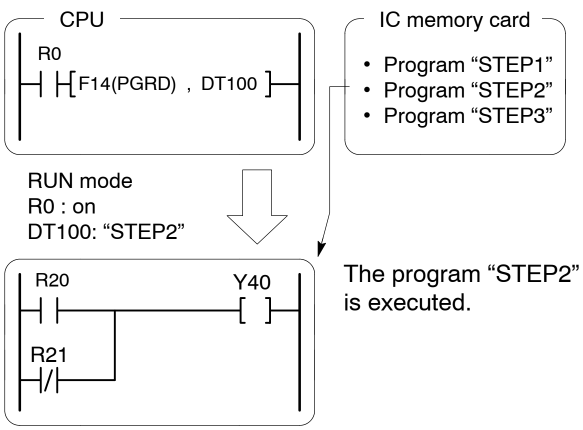
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX (*1)	IY (*2)	K	H	
S	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) With the FP2SH/FP10SH, this is I0 to IC.
(*2) With the FP2SH/FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

When the execution condition R0 is on, the programs for file names written to data register DT100 and subsequent data registers are read from the IC memory card, and are substituted for the program currently being executed.
If "STEP2" is written for data register DT100 or a subsequent register, the program with the file name "STEP2" stored on the IC memory card is read.



Description

The program for the file name stored in the area specified by S is read from the IC memory card, and is substituted for the program currently being executed.
Subsequent operation is carried out based on the program which was read.

Precautions when changing programs

Programs are changed when the **ED** instruction is executed. At that point, the mode changes automatically from the RUN mode to the PROG. mode.

All output goes off.

The contents of memory areas not specified as hold−type are cleared.

When a program is read, the system registers are rewritten at the same time. The same system register settings as those of the specified program should always be used, including the I/O map, remote I/O map, and others.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - No IC card is installed in the CPU.
  - The IC card access enable switch is set to off (disabled).
  - No DOS formatted area is found on the IC card.
  - The specified file name does not exist on the IC card.
  - The specified file is not a program file for the FP2SH/FP10SH.
  - The file is damaged.
  - A file name which cannot be used is specified.

Specifying file names

The program file name should be replaced with a character code, and written to the memory area that has S as the first address.

ASCII codes can be used.

No extension should be attached.

A single−byte numerical value H00 is the final code. If "H00" is written at the end of the file name (the MSB), the characters up to that point area treated as the file name.

If all 8 characters are specified for the file name, no final code is necessary. A code (H20) should be specified for any blank spaces.

All 8 characters specified

	(Higher)	(Lower)	
DT100	"B"	"A"	
DT101	"D"	"C"	
DT102	H20	H20	Space codes
DT103	H20	H20	Space codes

Only some of the characters specified

	(Higher)	(Lower)
DT100	"B"	"A"
DT101	"D"	"C"
DT102		H0 ← Final code

Specific example of specifying a file name

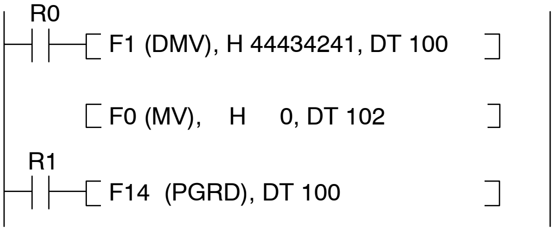
There are two ways to write a character code to a memory area specified by the **F14 (PGRD)** instruction.
The character code can be written directly, using a data move instruction (**F0** or **F1**).
The character code can be converted to the file name written when the program was created, using the ASCII conversion instruction **(F95)**.

Writing a character code directly

Example: **When only some of the characters are specified**

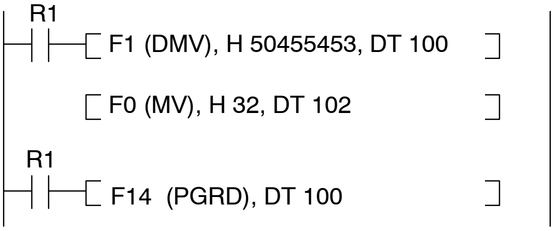
Specifying a file name of "ABCD"

File name	A	B	C	D
ASCII code	41	42	43	44



Specifying a file name of "STEP2"

File name	S	T	E	P	2
ASCII code	53	54	45	50	32



Specifying a file name with the ASCII conversion instruction, and converting it

The file name is converted to a character code using the ASCII conversion instruction **“F95 (ASC)”**, and is written to a specified memory area.

− Programming can only be done with the programming tool software.

− When the ASCII conversion instruction is executed, the results are stored in a 6−word (12−character) memory area. The specification should be made as follows.

Operand of F95 M □□□□□□□□␣␣␣␣

The file name (8 characters) should be entered with characters filling the spaces starting from the left. Spaces should be entered where characters are not specified.

Example: **(“␣” indicates a space)**

Specifying a file name of “ABCD”

```
    R0
|--| |--[F95 (ASC), M ABCD ␣␣␣␣␣␣␣␣,DT 100]        |
    R1
|--| |--[ F14  (PGRD), DT 100 ]                     |
|                                                    |
```

Specifying a file name of “STEP2”

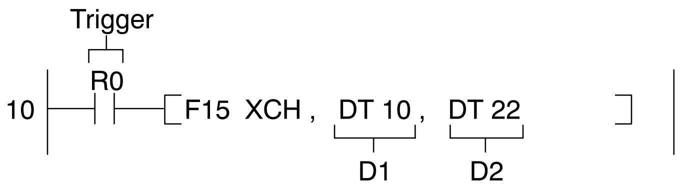
```
    R0
|--| |--[F95 (ASC), M STEP2 ␣␣␣␣␣␣␣,DT 100]        |
    R1
|--| |--[ F14  (PGRD), DT 100            ]          |
|                                                    |
```

F15 (XCH) P15 (PXCH) 16-bit data exchange

Outline Exchanges two 16-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P15 (PXCH)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 15 (XCH)
	DT 10
	DT 22

D1	16-bit area to be exchanged
D2	16-bit area to be exchanged

Operands

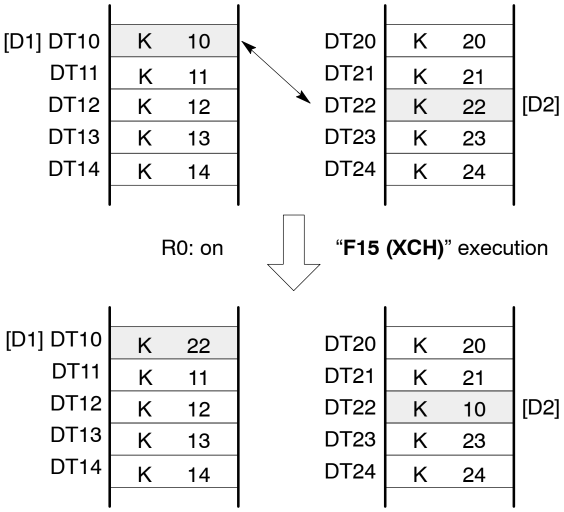
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D1	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

The contents of data register DT10 and data register DT22 are exchanged when trigger R0 turns on.



Description

The contents in the 16-bit areas specified by D1 and D2 are exchanged.

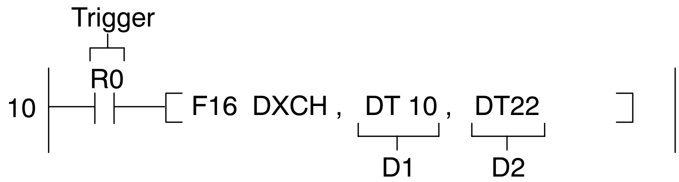
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F16 (DXCH) P16 (PDXCH) 32-bit data exchange

Outline Exchanges two 32-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F 16 (DXCH)
		DT 10
		DT 22

D1	Lower 16-bit area of 32-bit data to be exchanged
D2	Lower 16-bit area of 32-bit data to be exchanged

Operands

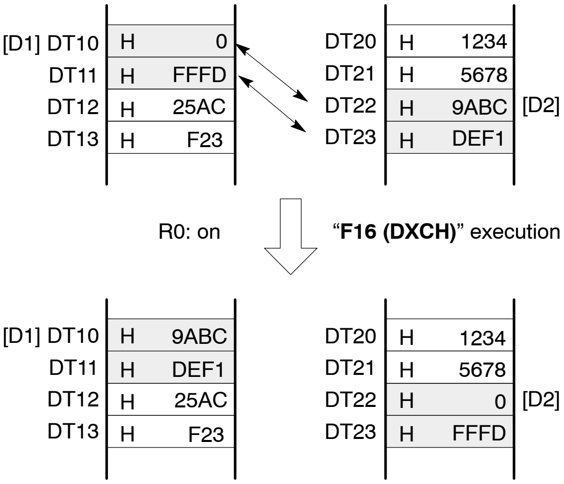
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

The contents of data registers DT11 and DT10 and data registers DT23 and DT22 are exchanged when trigger R0 turns on.



Description

The contents in the 32-bit areas specified by D1 and D2 are exchanged.

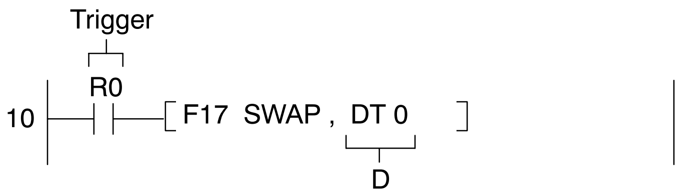
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F17 (SWAP) P17 (PSWAP) Higher/lower byte in 16-bit data exchange

Outline Exchanges higher and lower order bytes of the specified 16-bit data. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F 17 (SWAP)
		DT 0

D	16-bit area to be exchanged higher and lower bytes

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

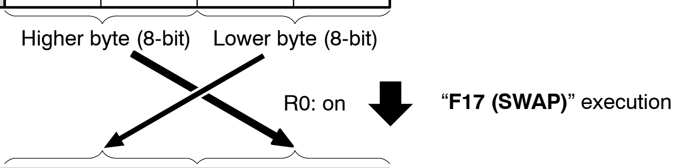
A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

The higher and lower bytes of data register DT0 are exchanged when trigger R0 turns on.

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	0 0 0 0	0 1 0 0	1 1 0 1	0 0 1 0
Hexadecimal	0	4	D	2



Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	1 1 0 1	0 0 1 0	0 0 0 0	0 1 0 0
Hexadecimal	D	2	0	4

Description

The higher order byte (higher 8-bit) and lower order byte (lower 8-bit) of the 16-bit area specified by D are exchanged.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

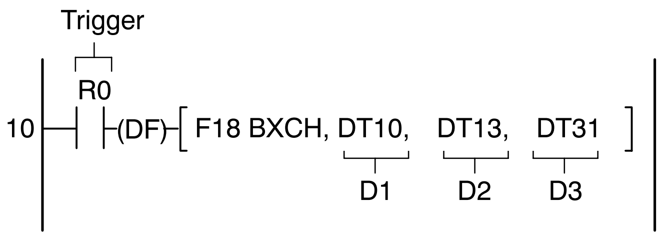
F18(BXCH)
P18(PBXCH)

16-bit blocked data exchange

Outline Exchanges the 16-bit blocked data.
For the FP0R/FPΣ/FP–X, the P type high–level instruction "P18 (PBXCH)" is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	0
11	F18		(BXCH)
12	DT		10
	DT		13
	DT		31

D1	Starting 16-bit area of block data 1
D2	Ending 16-bit area of block data 1
D3	Starting 16-bit area of block data 2

Operands

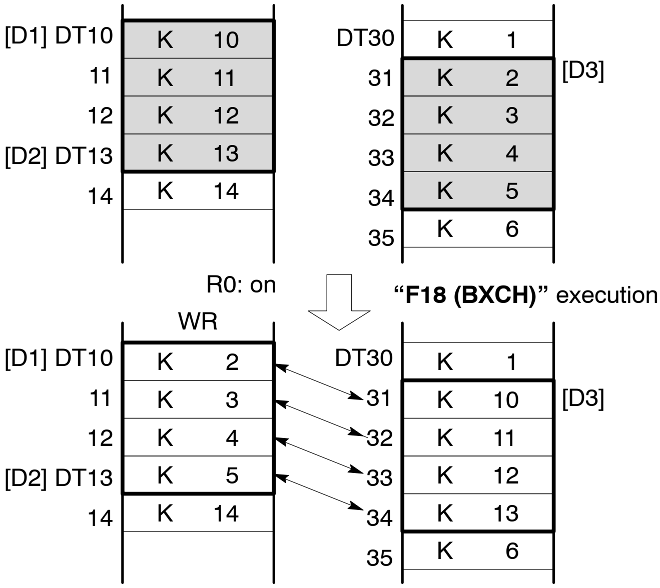
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D3	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP–X.

A: Available
N/A: Not Available

Explanation of example

The data block from data register DT10 to data register DT13 and the data block (DT31 to DT34) starting from data register DT31 are exchanged when trigger R0 turns on.



Description

The data block specified by D1 and D2 and the block starting from the 16-bit area specified by D3 are exchanged when the trigger turns on.

Precautions during programming

The starting area D1 and ending area D2 should:

Be the same type of operand.

Satisfy D1 ≦ D2. If area D1 > D2, an operation error occurs.

If the areas of blocks to be exchanged overlap, correct exchange will not be possible. Note, however, that an error will not occur (the error flag will not turn on).

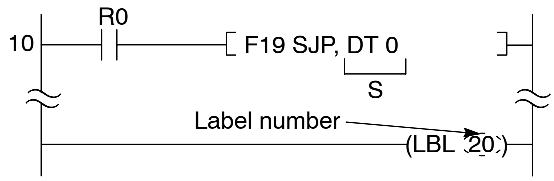
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - D1 > D2
  - The data block to be exchanged exceeds the limit of the destination area.

F19 (SJP) Auxiliary jump
LBL Label

Outline Skips to the **LBL** instruction with the same number as the data area specified by the **F19 (SJP)** instruction.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F19 (SJP) DT 0

S	16-bit area for storing the label number [0 to 255 (256 points)]

Operands

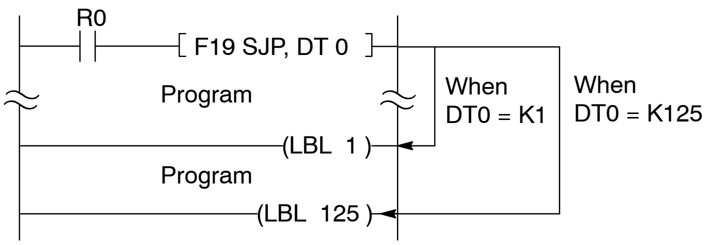
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX (*1)	IY (*2)	K	H	
S	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A

(*1) With the FP2, FP2SH, and FP10SH, this is I0 to IC.
(*2) With the FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Skips to the label number with the same number as the value in data register DT0 when trigger R0 turns on.



When DT0 is K1, the program skips from **F19 (SJP)** to **LBL1**.

When DT0 is K125, the program skips from **F19 (SJP)** to **LBL125**.

Description

The **F19 (SJP)** instruction skips the program between the **F19 (SJP)** and the **LBL** with the number specified by S when the trigger turns on.
Program execution continues from the next instruction after the jump destination label.

Up to 256 jump destinations can be specified (the range of values in which S can be stored is from K0 to K255).

LBL instructions are specified as destinations of **JP**, **LOOP** and **F19 (SJP)** instructions. Any instruction may be used as the starting point for the jump destination.

Two or more **LBL** instructions with the same number cannot be used in the same program.

If there is no label with the same number as the value of S, or if the value stored is outside of the range, the **F19 (SJP)** instruction will not be executed.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The content of S is smaller than K0.
  − The content of S is larger than K255.

Precautions during programming

If the label is written to an address prior to the **F19 (SJP)** instruction, be aware that there is a possibility that the scan cannot be completed, and an operation bottleneck error will occur.

The **F19 (SJP)** instruction cannot be used in a stepladder area (the range from **SSTP** to **CSTP**), in a subroutine, or in an interrupt program.

A jump cannot be made from a main program to a sub−program (subroutines or interrupt programs written subsequent to the **ED** instruction).

Using differential type instructions between F19 (SJP) and LBL instructions

This is the same as when programming is done between the **JP** and **LBL** instructions. Refer to the explanation of the **JP** and **LBL** instructions.

You must be careful when using one of the instructions below, which are executed by detecting the leading edge of execution condition (trigger) such as the differential instruction.

DF (leading edge differential)
Count input of **CT** (counter)
Count input of **F118** (up/down counter)
Shift input of **SR** (shift register)
Shift input of **F119** (left/right shift register)
NSTP (next step)
Differential execution type high−level instruction (this instruction is specified by P and a number)

F20 (+)
P20 (P+)

16-bit data addition
[D+S → D]

Outline Adds two 16-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P20 (P+)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
Trigger R0; 10 ─┤ ├─[F20 +, DT 1 , DT 10] (DT 1 = S, DT 10 = D)	10 11	ST R 0 F 20 (+) DT 1 DT 10

S	16-bit equivalent constant or 16-bit area (for addend)
D	16-bit area (for augend and result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

The contents of data register DT10 and data register DT1 are added together when trigger R0 turns on.
When the decimal number 4 is in DT1 and the decimal number 8 is in DT10, as shown below.

Augend [D]: K8

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0

+

Addend [S]: K4

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT1	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0

Result [D]: K12

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0

Description

The 16-bit equivalent constant or 16-bit area specified by S and the 16-bit area specified by D are added together.

Augend data		Addend data	Trigger turns on	Result
(D)	+	(S)	⟶	(D)

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If the calculated result accidentally overflows or underflows, use of the **F21 (D+)** instruction (32-bit data addition) is recommended.

When you use the **F21 (D+)** instruction instead of **F20 (+)**, be sure to convert the 16-bit addend and augend into 32-bit data using the **F89 (EXT)** instruction.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 16-bit data (overflows or underflows).

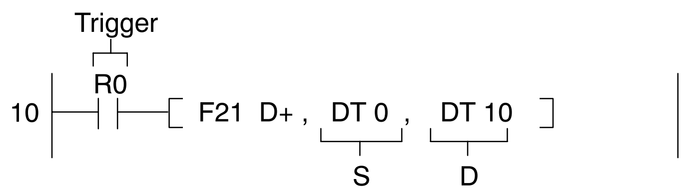
F21 (D+) P21 (PD+)

32-bit data addition
[(D+1, D) + (S+1, S) → (D+1, D)]

Outline Adds two 32-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P21 (PD+)” is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F 21 (D+)
	DT 0
	DT 10

S	32-bit equivalent constant or lower 16-bit area of 32-bit area (for addend)
D	Lower 16-bit area of 32-bit data (for augend and result)

Operands

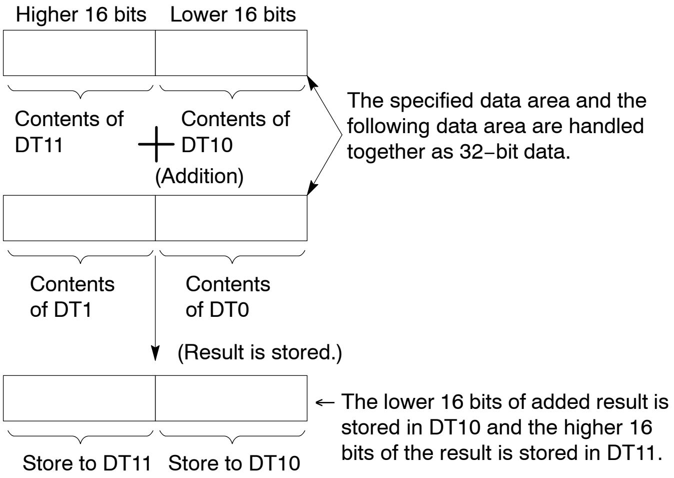
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

The contents (32 bits) of data registers DT11 and DT10 and the contents (32 bits) of data registers DT1 and DT0 are added together when trigger R0 turns on.



Description

The 32-bit equivalent constant or the 32-bit area specified by S and the 32-bit data specified by D are added together.

Augend data		**Addend data**		**Result**
(D+1, D)	+	**(S+1, S)**	⟶	**(D+1, D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 32-bit data (overflows or underflows).

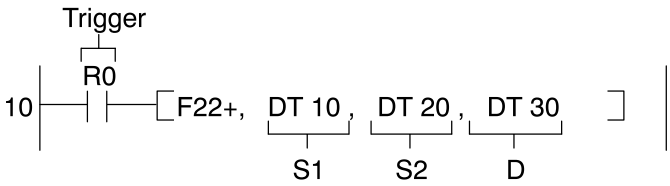
F22 (+) P22 (P+)

16-bit data addition [S1 + S2 → D]

Outline Adds two 16-bit data items and stores the result in the specified area. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P22 (P+)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 22 (+)
	DT 10
	DT 20
	DT 30

S1	16-bit equivalent constant or 16-bit area (for augend)
S2	16-bit equivalent constant or 16-bit area (for addend)
D	16-bit area (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

The contents of data registers DT10 and DT20 are added when trigger R0 turns on. The added result is stored in data register DT30.
when the decimal number 8 is in DT10 and the decimal number 4 is in DT20, as shown below.

Augend [S1]: K8

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0

Addend [S2]: K4 + **(Addition)**

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0

Result [D]: K12

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT30	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0

Description

The 16-bit data or 16-bit equivalent constant specified by S1 and S2 are added together. The added result is stored in D.

Augend data		**Addend data**		**Result**
(S1)	+	**(S2)**	⟶	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If the calculated result accidentally overflows or underflows, use of the **F23 (D+)** instruction (32-bit data addition) is recommended.

When you use the **F23 (D+)** instruction instead of **F22 (+)**, be sure to convert the 16-bit addend and augend into 32-bit data using the **F89 (EXT)** instruction.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 16-bit data (overflows or underflows).

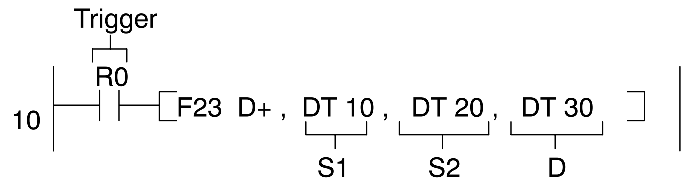
F23 (D+)
P23 (PD+)

32-bit data addition
[(S1+1, S1) + (S2+1, S2) → (D+1, D)]

Outline Adds two 32-bit data items and stores the result in the specified area. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P23 (PD+)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 23		(D+)
	DT		10
	DT		20
	DT		30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data (for augend)
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data (for addend)
D	Lower 16-bit area of 32-bit data (for result)

Operands

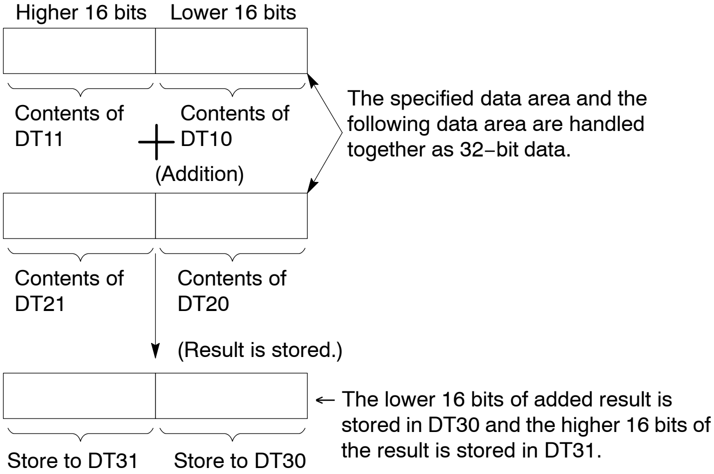
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

The contents of data registers DT11 and DT10 and the contents of data registers DT21 and DT20 are added when trigger R0 turns on. The added result is stored in data registers DT31 and DT30.



Description

The 32-bit data or 32-bit equivalent constant specified by S1 and S2 are added together. The added result is stored in D+1 and D.

Augend data		Addend data		Result
(S1+1, S1)	+	(S2+1, S2)	⟶	(D+1, D)

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1, D+1) are automatically determined once the lower 16-bit areas (S1, S2, D) are specified.

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

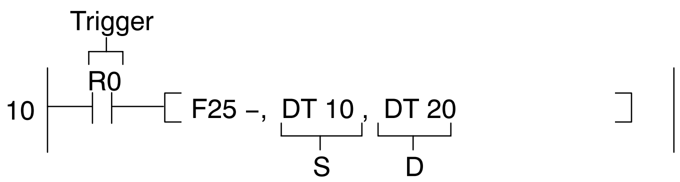
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 32-bit data (overflows or underflows).

F25 (−) P25 (P−)

16-bit data subtraction [D − S → D]

Outline Subtracts 16-bit data from the minuend.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P25 (P−)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F 25 (−)
		DT 10
		DT 20

S	16-bit equivalent constant or 16-bit area (for subtrahend)
D	16-bit area (for minuend and result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

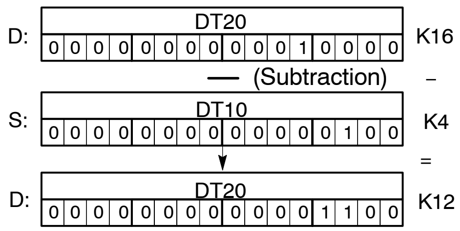
A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

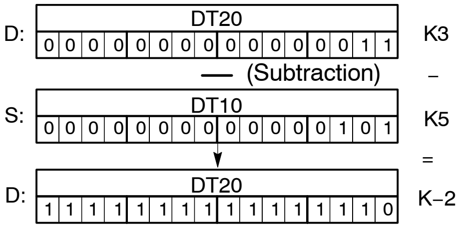
Explanation of example

Subtracts the contents of data register DT10 from the contents of data register DT20 when trigger R0 turns on.

Example 1: When the decimal number 16 is in DT20 and the decimal number 4 is in DT10.



Example 2: When the decimal number 3 is in DT20 and the decimal number 5 is in DT10.



Description

Subtracts the 16-bit equivalent constant or 16-bit area specified by S from the 16-bit area specified by D.

Minuend data		Subtrahend data		Result
(D)	–	**(S)**	⟶	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If the calculated result accidentally overflows or underflows, use of the **F26 (D−)** instruction (32-bit data subtraction) is recommended.

When you use the **F26 (D−)** instruction instead of **F25 (−)**, be sure to convert the 16-bit subtrahend and minuend into 32-bit data using the **F89 (EXT)** instruction.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

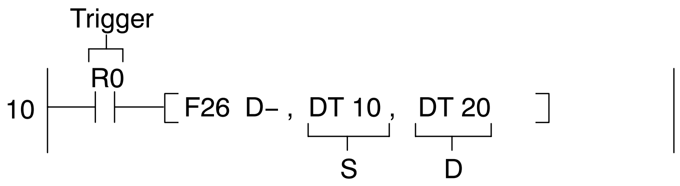
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 16-bit data (overflows or underflows).

F26 (D−)
P26 (PD−)

32-bit data subtraction
[(D+1, D) − (S+1, S) → (D+1, D)]

Outline Subtracts 32-bit data from the minuend.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P26 (PD−)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10 11	ST R 0 F 26 (D−) DT 10 DT 20

S	32-bit equivalent constant or lower 16-bit area of 32-bit area (for subtrahend)
D	Lower 16-bit area of 32-bit data (for minuend and result)

Operands

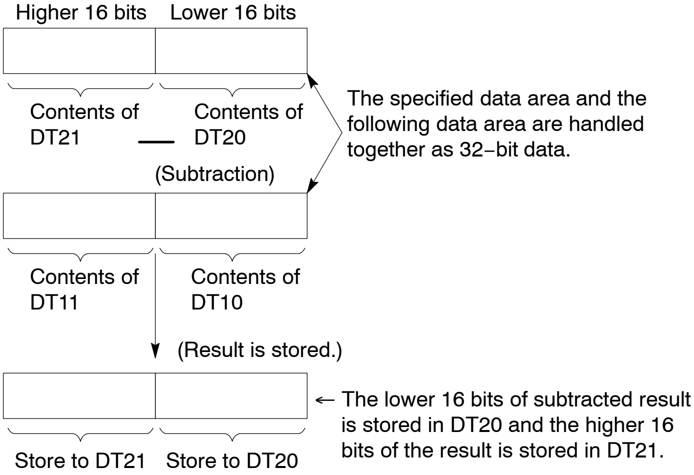
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Subtracts the contents (32 bits) of data registers DT11 and DT10 from the contents (32 bits) of data registers DT21 and DT20 when trigger R0 turns on.



Description

Subtracts the 32-bit equivalent constant or the 32-bit data specified by S from the 32-bit data specified by D.

Minuend data		Subtrahend data		Result
(D+1, D)	—	**(S+1, S)**	⟶	**(D+1, D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

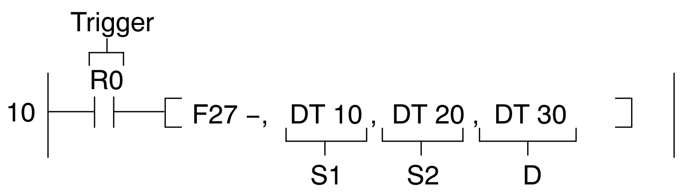
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 32-bit data (overflows or underflows).

F27 (−) P27 (P−)

16-bit data subtraction
[S1 − S2 → D]

Outline Subtracts 16-bit data from the minuend and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P27 (P−)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 27 (−) DT 10 DT 20 DT 30

S1	16-bit equivalent constant or 16-bit area (for minuend)
S2	16-bit equivalent constant or 16-bit area (for subtrahend)
D	16-bit area (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Description

Subtracts the 16-bit data or 16-bit equivalent constant specified by S2 from the 16-bit data or 16-bit equivalent constant specified by S1. The subtracted result is stored in D.

Minuend data		Subtrahend data		Result
(S1)	−	**(S2)**	⟶	**(D)**

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 16-bit data (overflows or underflows).

Explanation of example

Subtracts the contents of data register DT20 from the contents of data register DT10 when trigger R0 turns on. The subtracted result is stored in data register DT30.

Example 1: When the decimal number 16 is in DT10 and the decimal number 4 is in DT20.

Minuend [S1]: K16

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0

− (Subtraction)

Subtrahend [S2]: K4

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0

Result [D]: K12

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT30	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0

Example 2: When the decimal number 3 is in DT10 and the decimal number 5 is in DT20.

Minuend [S1]: K3

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1

− (Subtraction)

Subtrahend [S2]: K5

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 1

Result [D]: K−2

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT30	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If the calculated result accidentally overflows or underflows, use of the **F28 (D−)** instruction (32-bit data subtraction) is recommended.

When you use the **F28 (D−)** instruction instead of **F27 (−)** be sure to convert the 16-bit subtrahend and minuend into 32-bit data using the **F89 (EXT)** instruction.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

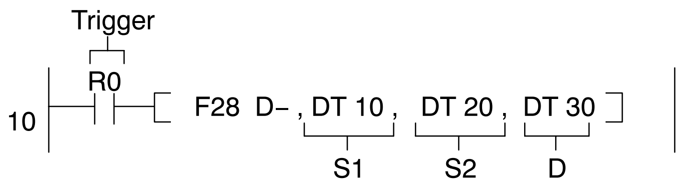
F28 (D−) P28 (PD−)

32-bit data subtraction
[(S1+1, S1) − (S2+1, S2) → (D+1, D)]

Outline Subtracts 32-bit data from the minuend and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P28 (PD−)” is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F 28 (D−)
	DT 10
	DT 20
	DT 30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data (for minuend)
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data (for subtrahend)
D	Lower 16-bit area of 32-bit data (for result)

Operands

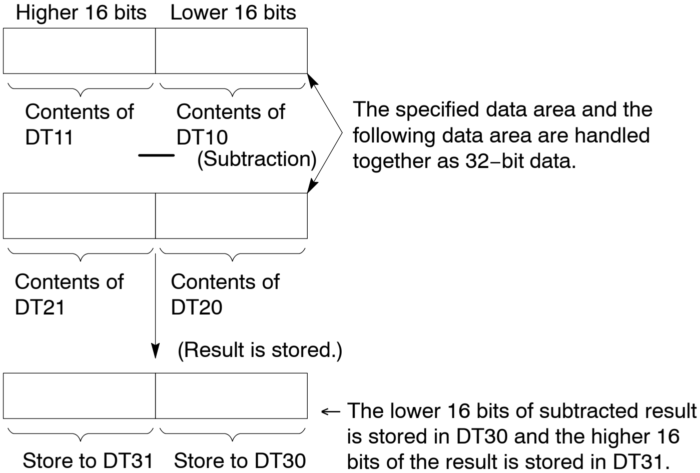
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Subtracts the contents of data registers DT21 and DT20 from the contents of data registers DT11 and DT10 when trigger R0 turns on. The subtracted result is stored in data registers DT31 and DT30.



Description

Subtracts the 32-bit data or 32-bit equivalent constant specified by S2 from the 32-bit data or 32-bit equivalent constant specified by S1. The subtracted result is stored in D+1 and D.

Minuend data		**Subtrahend data**		**Result**
(S1+1, S1)	—	**(S2+1, S2)**	⟶	**(D+1, D)**

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1 D+1) are automatically determined once the lower 16-bit areas (S1, S2, D) are specified.

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow or underflow will result.

Under normal circumstances, do not allow an overflow or underflow to occur.

If an overflow or underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

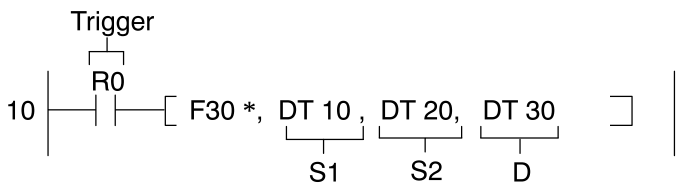
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 32-bit data (overflows or underflows).

F30 (*) P30 (P*) 16-bit data multiplication [S1 × S2 → (D+1, D)]

Outline Multiplies two 16-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 30		(*)
	DT		10
	DT		20
	DT		30

S1	16-bit equivalent constant or 16-bit area (for multiplicand)
S2	16-bit equivalent constant or 16-bit area (for multiplier)
D	Lower 16-bit area of 32-bit data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Multiplies the contents of data register DT10 and DT20 when trigger R0 turns on.
The result is stored in data registers DT 31 and DT 30.
When the decimal number 8 is in DT10 and the decimal number 2 is in DT20.

Multiplicand [S1]: K8

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0

Multiplier [S2]: K2 ✕

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0

Result [D+1, D]: K16

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT31	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Higher 16-bit area

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT30	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0

Lower 16-bit area

The lower 16 bits of the 32−bit multiplication result are stored in the specified memory area (DT30), and the higher16 bits are stored in the area following the specified area (DT31).

Description

Multiplies the 16-bit data or 16-bit equivalent constant specified by S1 and the 16-bit data or 16-bit equivalent constant specified by S2. The multiplied result is stored in D+1 and D (32-bit area).

Multiplicand data		**Multiplier data**		**Result**
(S1)	×	**(S2)**	⟶	**(D+1, D)**

The multiplied result is stored in the 32-bit area.

The higher 16-bit area (D+1) is automatically determined once the lower 16-bit area (D) is specified.

Flag conditions

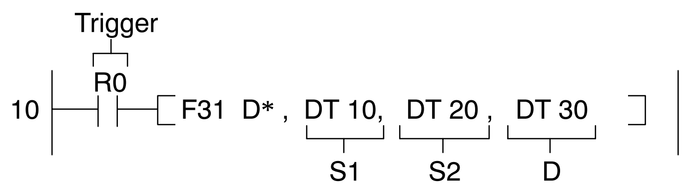
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F31 (D*) P31 (PD*)

32-bit data multiplication [(S1+1, S1) × (S2+1, S2) → (D+3, D+2, D+1, D)]

Outline Multiplies two 32-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10 11	ST R 0 F 31 (D*) DT 10 DT 20 DT 30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data (for multiplicand)
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data (for multiplier)
D	Lower 16-bit area of 64-bit data (for result)

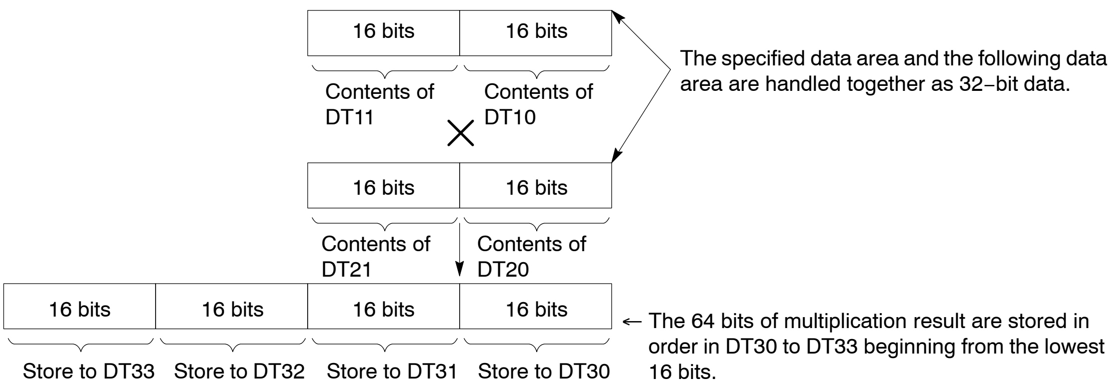
Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example



Description

Multiplies the 32-bit data or 32-bit equivalent constant specified by S1 and the one specified by S2.
The multiplied result is stored in D+3, D+2, D+1 and D.

Multiplicand data		Multiplier data		Result
(S1+1, S1)	×	**(S2+1, S2)**	→	**(D+3, D+2, D+1, D)**

The multiplied result is stored in the 64-bit area.

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.
The areas (D+3, D+2, D+1) other than the lowest 16-bit area (D) are automatically determined once the lowest 16-bit area is specified.

Flag conditions

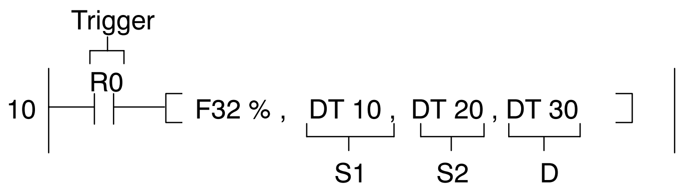
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F32 (%) P32 (P%) 16-bit data division [S1/S2 → D… (DT9015/DT90015)]

Outline Divides 16-bit data by the divisor.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram



Boolean

Address	Instruction
10	ST R 0
11	F 32 (%)
	DT 10
	DT 20
	DT 30

S1	16-bit equivalent constant or 16-bit area (for dividend)
S2	16-bit equivalent constant or 16-bit area (for divisor)
D	16-bit area (for quotient), (Remainder is stored in special data register DT9015/DT90015.)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Divides the contents of data register DT10 by decimal constant DT20 when trigger R0 turns on. The quotient is stored in data register DT30 and the remainder is stored in special data register DT9015/DT90015. When the decimal number 15 is in DT10 and the decimal number 4 is in DT20, as shown below.

Dividend [S1]: K15

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1

÷

Divisor [S2]: K4

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0

Quotient [D]: K3

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT30	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1

. . .

Remainder: K3

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT9015/ DT90015	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1

Description

The 16-bit data or 16-bit equivalent constant specified by S1 is divided by the 16-bit data or 16-bit equivalent constant specified by S2.
The quotient is stored in D and the remainder is stored in the special data register DT9015 (DT90015 for FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH).

Dividend data		Divisor		Quotient		Remainder
(S1)	÷	**(S2)**	⟶	**(D)**	· · · ·	**(DT9015/DT90015)**

With the FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH and FP0 C10, C14, C16, C32/FP−e, the numbers of the special data registers are different.

Type	Special data register
FP0 C10, C14, C16, C32/ FP−e	DT9015
FP0 T32/FP0R/FPΣ/FP−X/ FP2/FP2SH/FP10SH	DT90015

Flag conditions

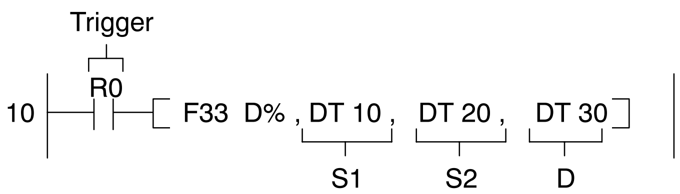
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the negative minimum value K-32768 (H8000) is divided by K-1 (HFFFF).

F33 (D%) P33 (PD%)

32-bit data division [(S1+1, S1)/(S2+1, S2) → (D+1, D)…(DT9016, DT9015)/ (DT90016, DT90015)]

Outline Divides 32-bit data by the divisor.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P33 (PD%)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F 33 (D%)
		DT 10
		DT 20
		DT 30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data (for dividend)
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data (for divisor)
D	Lower 16-bit area of 32-bit data (for quotient) (Remainder is stored in special data registers DT9016 and DT9015/DT90016 and DT90015.)

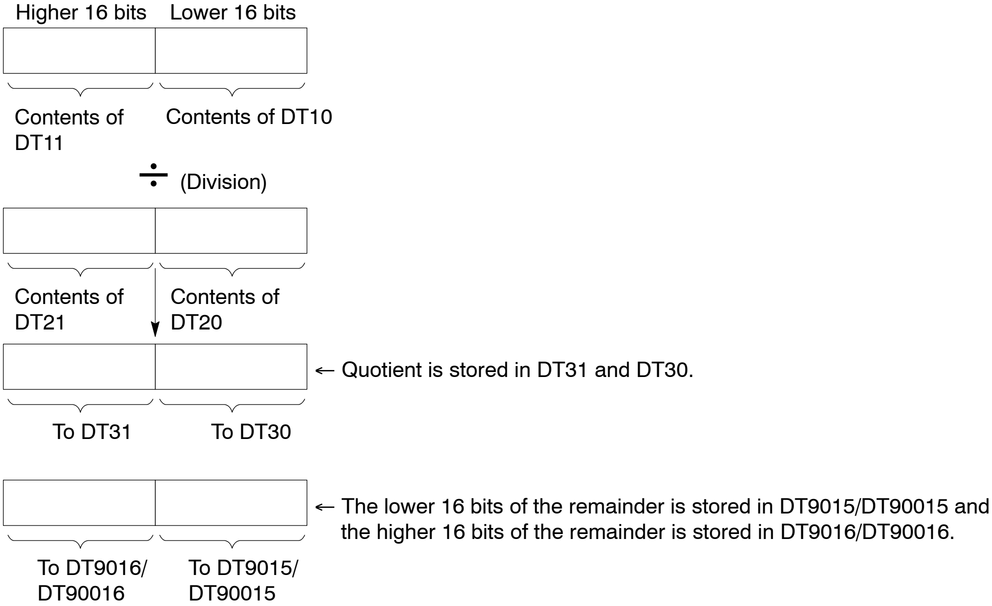
Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example



Description

The 32-bit data or 32-bit equivalent constant specified by S1 is divided by the 32-bit data or 32-bit equivalent constant specified by S2. The quotient is stored in D+1 and D and the remainder is stored in the special data registers DT9016 and DT9015 (DT90016 and DT90015 for FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH).

Dividend data		Divisor		Quotient	Remainder
S1: lower 16-bit **S1+1: higher 16-bit**	÷	**S2: lower 16-bit** **S2+1: higher 16-bit**	⟶	**D:** lower 16-bit **D+1:** higher 16-bit	**DT9015/DT90015** **DT9016/DT90016**

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1, D+1) are automatically determined once the lower 16-bit areas (S1, S2, D) are specified.

With the FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH and FP0 C10, C14, C16, C32/FP−e, the numbers of the special data registers are different.

Type	Special data register
FP0 C10, C14, C16, C32/ FP−e	DT9016, DT9015
FP0 T32/FP0R/FPΣ/FP−X/ FP2/FP2SH/FP10SH	DT90016, DT90015

Flag conditions

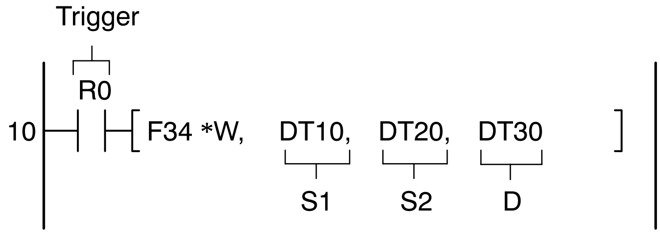
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when negative minimum value K-2147483648 (H80000000) is divided by K-1 (HFFFFFFFF).

F34 (*W) P34 (P*W) 16-bit data multiplication (result in 16 bits)

Outline Multiplies two 16-bit data items and stores the result in the specified 16-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P34 (P*W)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F34 (*W)
	DT 10
	DT 20
	DT 30

S1	16-bit equivalent constant or 16-bit area (for multiplicand)
S2	16-bit equivalent constant or 16-bit area (for multiplier)
D	16-bit area for storing multiplied result

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

Multiplies the contents of data register DT10 and data register DT20 when trigger R0 turns on. The multiplied result is stored in data register DT30.

Multiplicand [S1]: K8

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0

×

Multiplier [S2]: K2

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0

↓ R0: on

Result [D]: K16

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0

Description

Multiplies the 16-bit data or 16-bit equivalent constant specified by S1 and the 16-bit data or 16-bit equivalent constant specified by S2 when the trigger turns on. The multiplied result is stored in D (16-bit area).

Multiplicand data		Multiplier data	Trigger turns on	Result
S1	×	**S2**	⟶	**D**

The multiplied result is stored in the 16-bit area.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – The calculated result exceeds the 16-bit area specified by D.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

Precautions during programming

Keep the calculated result D within the range K−32768 to K32767.

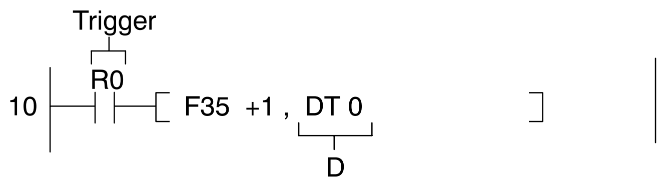
F35 (+1) P35 (P+1)

16-bit data increment [D + 1 → D]

Outline

Adds 1 to 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P35 (P+1)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 35 (+1) DT 0

D	16-bit area to be increased by 1

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Adds 1 to the contents of data register DT0 when trigger R0 turns on.

Original data [D]: K9

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1

+1

Result [D]: K10

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 0

Description

Adds 1 to the 16-bit data specified by D. The result is stored in D.

Original data				Result
(D)	**+**	**1**	→	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If the operation result accidentally overflows, use of the **F36 (D+1)** instruction (32-bit data increment) is recommended.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

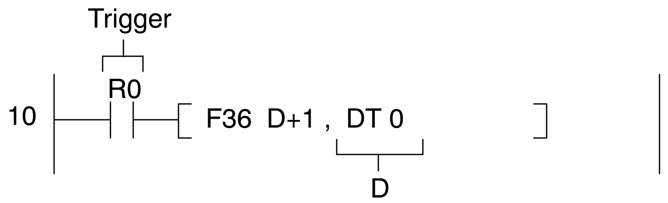
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 16-bit data (overflows).

F36 (D+1)
P36 (PD+1)

32-bit data increment
[(D + 1, D) + 1 → (D + 1, D)]

Outline Adds 1 to 32-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P36 (PD+1)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F 36 (D+1)
		DT 0

D	Lower 16-bit area of 32-bit data to be increased by 1

Operands

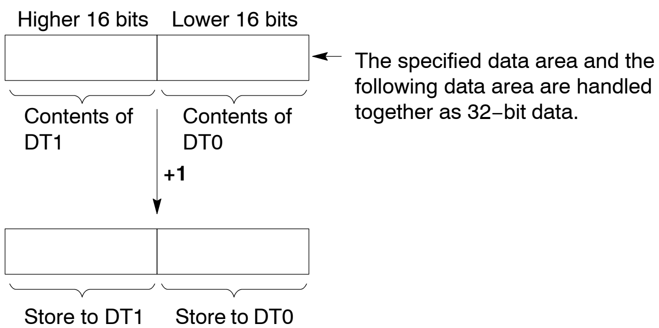
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Adds 1 to the content of data registers DT1 and DT0 when trigger R0 turns on.



Description

Adds 1 to the 32-bit data specified by D. The result is stored in D+1 and D.

Original data				**Result**
(D+1, D)	**+**	**1**	→	**(D+1, D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

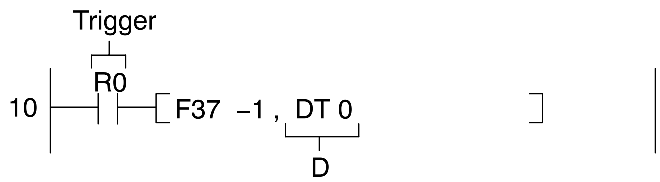
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 32-bit data (overflows).

F37 (−1)
P37 (P−1)

16-bit data decrement
[D − 1 → D]

Outline Subtracts 1 from 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example



Boolean Address	Boolean Instruction
10	ST R 0
11	F 37 (−1)
	DT 0

D	16-bit area to be decreased by 1

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Subtracts 1 from the contents of data register DT0 when trigger R0 turns on.

Original data [D]: K10

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 0

−1

Result [D]: K9

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1

Description

Subtracts 1 from the 16-bit data specified by D. The result is stored in D.

Original data **Result**

(D) **−** **1** → **(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If the operation result accidentally underflows, use of the **F38 (D−1)** instruction (32-bit data decrement) is recommended.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

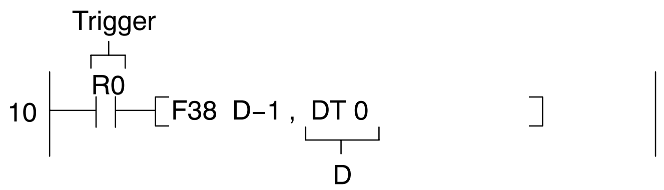
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 16-bit data (underflows).

F38 (D−1) P38 (PD−1)

32-bit data decrement
[(D+1, D) − 1 → (D+1, D)]

Outline Subtracts 1 from 32-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P38 (PD−1)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10	ST R 0
	11	F 38 (D−1)
		DT 0

D	Lower 16-bit area of 32-bit data to be decreased by 1

Operands

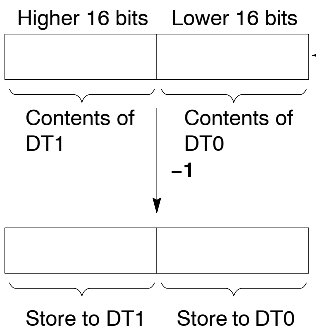
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

Explanation of example

Subtracts 1 from the content of data registers DT1 and DT0 when trigger R0 turns on.



The specified data area and the following data area are handled together as 32−bit data.

Description

Subtracts 1 from the 32-bit data specified by D. The result is stored in D+1 and D.

Original data				**Result**
(D+1, D)	**–**	**1**	→	**(D+1, D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

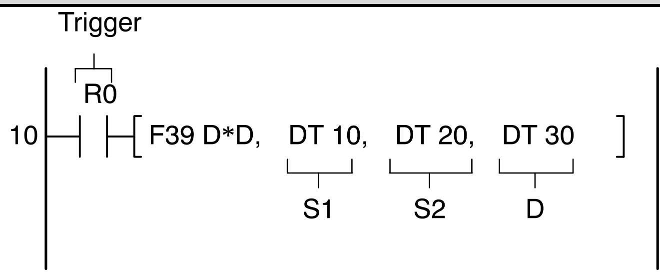
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 32-bit data (underflows).

F39 (D*D) P39 (PD*D) 32-bit data multiplication (result in 32 bits)

Outline Multiplies two 32-bit data items and stores the result in the specified 32-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P39 (PD*D)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F39		(D*D)
	DT		10
	DT		20
	DT		30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data (for multiplicand)
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data (for multiplier)
D	Lower 16-bit area of 32-bit data (for result)

Operands

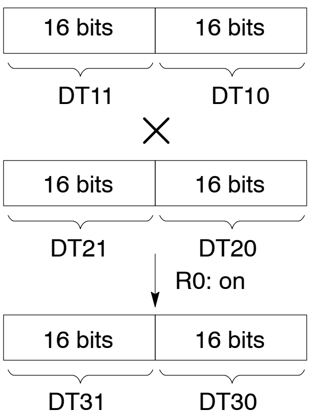
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

Multiplies the contents of data registers DT11 and DT10 and the contents of data registers DT21 and DT20 when trigger R0 turns on. The multiplied result is stored in data registers DT31 and DT30.



Description

Multiplies the 32-bit data or 32-bit equivalent constant specified by S1 and the one specified by S2 when the trigger turns on.
The multiplied result is stored in D+1 and D (32-bit area).

Multiplicand data		Multiplier data	Trigger turns on	Result (32-bit)
(S1: lower 16-bit S1+1: higher 16-bit)	×	(S2: lower 16-bit S2+1: higher 16-bit)	⟶	(D D+1)

The multiplied result is stored in the 32-bit area (2 words).

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The calculated result exceeds the 32-bit area specified by D.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

Precautions during programming

Keep the calculated result D within the range K−2147483648 to K2147483647.

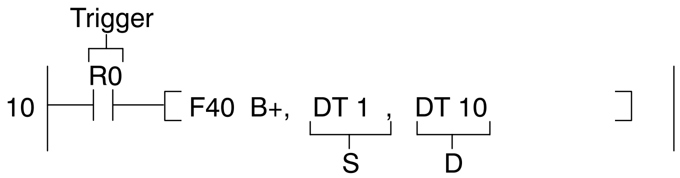
F40 (B+) P40 (PB+)

4-digit BCD data addition [D + S → D]

Outline Adds two BCD data items that express 8-digit decimal numbers (8-digit BCD H codes).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P40 (PB+)” is not available.

Program example

Ladder Diagram



Address	Instruction (Boolean)
10	ST R 0
11	F 40 (B+)
	DT 1
	DT 10

S	4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data (for addend)
D	16-bit area for 4-digit BCD data (for augend and result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

The contents of data register DT10 and data register DT1 are added together when trigger R0 turns on. When H4 (BCD)is in DT1 and H8 (BCD) is in DT10, as shown below.

Augend [D]: H8 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
BCD H code	0	0	0	8

Addend [S]: H4 (BCD) + (Addition)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT1	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0
BCD H code	0	0	0	4

Result [D]: H12 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0
BCD H code	0	0	1	2

Description

The 4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data specified by S and the 16-bit area for 4-digit BCD data specified by D are added together.

Augend data		Addend data		Result
(D)	+	**(S)**	⟶	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If the calculated result accidentally overflows, use of the **F41 (DB+)** instruction (8-digit BCD data addition) is recommended.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 4-digit BCD data (overflows).

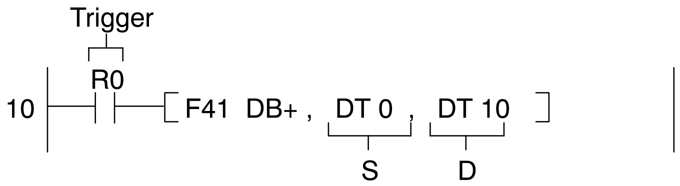
F41 (DB+) P41 (PDB+)

8-digit BCD data addition [(D+1, D) + (S+1, S) → (D+1, D)]

Outline Adds two BCD data items that express 8-digit decimal numbers (8-digit BCD H codes).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P41 (PDB+)” is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	0
11	F 41		(DB+)
	DT		0
	DT		10

S	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for addend)
D	Lower 16-bit area for 8-digit BCD data (for augend and result)

Operands

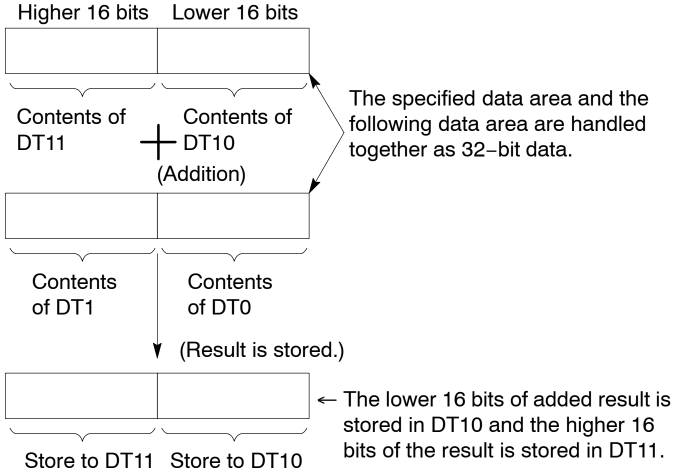
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

The contents of data registers DT11 and DT10 and the contents of data registers DT1 and DT0 are added together when trigger R0 turns on.



Description

The 8-digit BCD equivalent constant or 8-digit BCD data specified by S and the 8-digit BCD data specified by D are added together.

Augend data		Addend data		Result
D: lower 4-digit **D+1:** higher 4-digit	+	**S:** lower 4-digit **S+1:** higher 4-digit	→	**D:** lower 4-digit **D+1:** higher 4-digit

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The area specified using the index modifier exceeds the limit.
  − The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 8-digit BCD data (overflows).

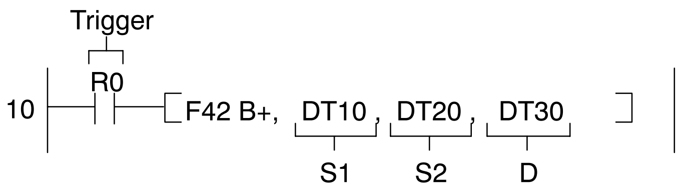
F42 (B+)
P42 (PB+)

4-digit BCD data addition
[S1 + S2 → D]

Outline Adds two BCD data items that express 4-digit decimal numbers (4-digit BCD H codes) and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P42 (PB+)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 42		(B+)
	DT		10
	DT		20
	DT		30

S1	4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data (for augend)
S2	4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data (for addend)
D	16-bit area for 4-digit BCD data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	K	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

The contents of data register DT10 and data register DT20 are added together when trigger R0 turns on. The added result is stored in data register DT30.
When H (BCD) 8 is in DT10 and H (BCD) 4 is in DT20, as shown below.

Augend [S1]: H8 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
BCD H code	0	0	0	8

Addend [S2]: H4 (BCD) + (Addition)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0
BCD H code	0	0	0	4

Result [D]: H12 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0
BCD H code	0	0	1	2

Description

The 4-digit BCD equivalent constants or 16-bit areas for 4-digit BCD data specified by S1 and S2 are added together. The added result is stored in D.

Augend data		**Addend data**		**Result**
(S1)	+	**(S2)**	⟶	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If the calculated result accidentally overflows, use of the **F43 (DB+)** instruction (8-digit BCD data addition) is recommended.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

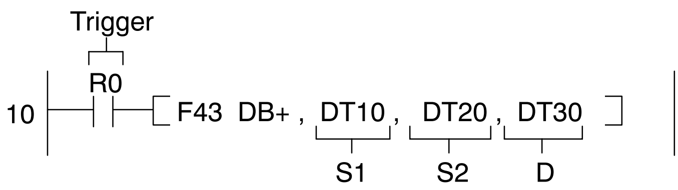
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The area specified using the index modifier exceeds the limit.
  − The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as "0".
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 4-digit BCD data (overflows).

F43 (DB+) P43 (PDB+)

8-digit BCD data addition
[(S1+1, S1) + (S2+1, S2) → (D+1, D)]

Outline Adds two BCD data items that express 8-digit decimal numbers (8-digit BCD H codes) and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P43 (PDB+)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 43 (DB+) DT 10 DT 20 DT 30

S1	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for augend)
S2	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for addend)
D	Lower 16-bit area for 8-digit BCD data (for result)

Operands

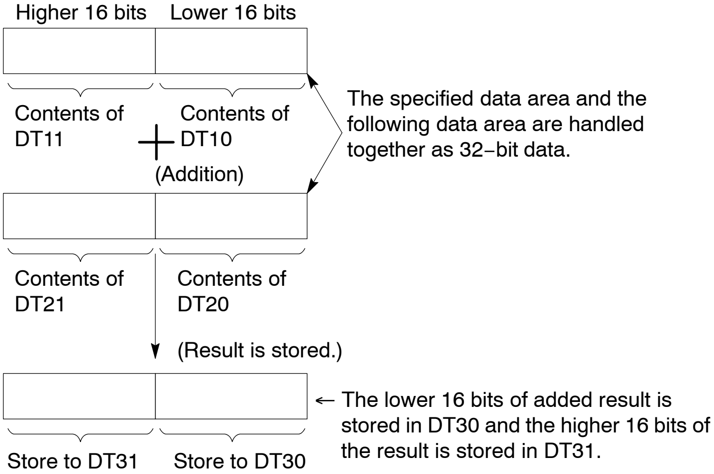
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	K	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

The contents of data registers DT11 and DT10 and the contents of data registers DT21 and DT20 are added together when trigger R0 turns on. The added result is stored in data registers DT31 and DT30.



Description

The 8-digit BCD equivalent constants or 8-digit BCD data specified by S1 and S2 are added together. The added result is stored in D+1 and D.

Augend data		Addend data		Result
(**S1:** lower 4-digit **S1+1:** higher 4-digit)	+	(**S2:** lower 4-digit **S2+1:** higher 4-digit)	→	(**D:** lower 4-digit **D+1:** higher 4-digit)

When processing 8-digit BCD data, the higher 16-bit areas for 8-digit BCD data (S+1, D+1) are automatically determined once the lower 16-bit areas (S, D) are specified.

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

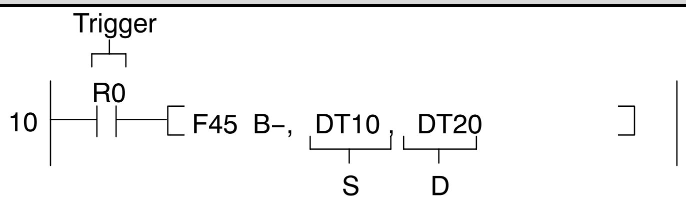
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 8-digit BCD data (overflows).

F45 (B−) P45 (PB−)

4-digit BCD data subtraction [D − S → D]

Outline Subtracts one BCD data item that expresses a 4-digit decimal number (4-digit BCD H codes) from another (minuend).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P45 (PB−)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 45 (B−) DT 10 DT 20

S	4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data (for subtrahend)
D	16-bit area for 4-digit BCD data (for minuend and result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

Subtracts the contents of data register DT10 from the contents of data register DT20 when trigger R0 turns on. When H (BCD) 16 is in DT20 and H (BCD) 4 is in DT10, as shown below.

Minuend [D]: H16 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	0 0 0 0	0 0 0 0	0 0 0 1	0 1 1 0
BCD H code	0	0	1	6

− (Subtraction)

Subtrahend [S]: H4 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0
BCD H code	0	0	0	4

Result [D]: H12 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0
BCD H code	0	0	1	2

Description

Subtracts the 4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data specified by S from the 16-bit area for 4-digit BCD data specified by D.

Minuend data		Subtrahend data		Result
(D)	–	(S)	⟶	(D)

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If the calculated result accidentally underflows, use of the **F46 (DB−)** instruction (8−digit BCD data subtraction) is recommended.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

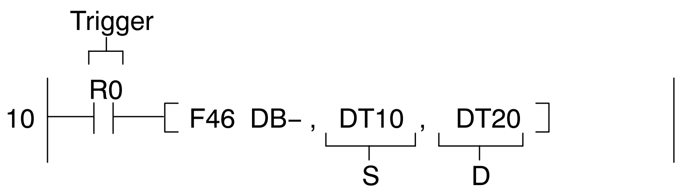
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The area specified using the index modifier exceeds the limit.
  − The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 4-digit BCD data (underflow).

F46 (DB−) P46 (PDB−)

8-digit BCD data subtraction [(D+1, D) − (S+1, S) → (D+1, D)]

Outline Subtracts one BCD data item that expresses an 8-digit decimal number (8-digit BCD H code) from another (minuend).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P46 (PDB−)” is not available.

Program example



Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F 46 (DB−)
		DT 10
		DT 20

S	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for subtrahend)
D	Lower 16-bit area for 8-digit BCD data (for minuend and result)

Operands

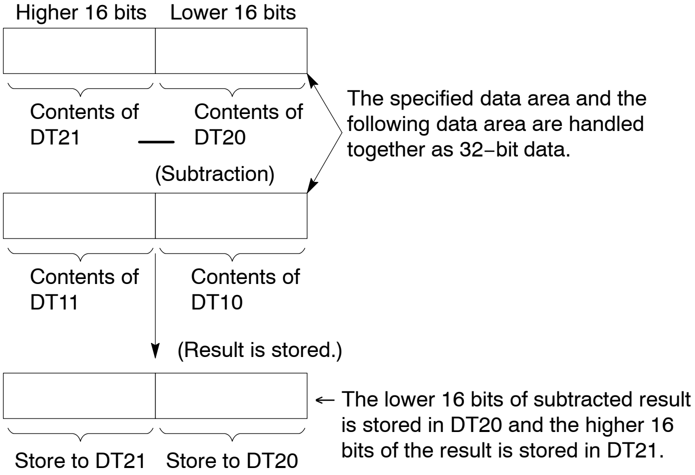
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

Subtracts the contents of data registers DT11 and DT10 from the contents of data registers DT21 and DT20 when trigger R0 turns on.



Description

Subtracts the 8-digit BCD equivalent constant or 8-digit BCD data specified by S from the 8-digit BCD data specified by D.

Augend data		Addend data		Result
D: lower 4-digit **D+1:** higher 4-digit	−	**S:** lower 4-digit **S+1:** higher 4-digit	→	**D:** lower 4-digit **D+1:** higher 4-digit

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The area specified using the index modifier exceeds the limit.
  − The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 8-digit BCD data (underflows).

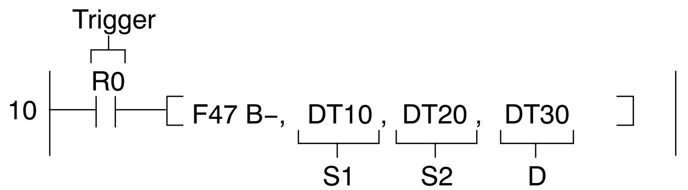
F47 (B−) P47 (PB−)

4-digit BCD data subtraction [S1 − S2 → D]

Outline Subtracts one BCD data item that expresses a 4-digit decimal number (4-digit BCD H code) from another (minuend) and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P47 (PB−)” is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F 47 (B−)
	DT 10
	DT 20
	DT 30

S1	4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data (for minuend)
S2	4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data (for subtrahend)
D	16-bit area for 4-digit BCD data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	K	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

Subtracts the contents of data register DT20 from the contents of data register DT10 when trigger R0 turns on. The subtracted result is stored in data register DT30.
When H (BCD) 16 is in DT10 and H (BCD) 4 is in DT20, as shown below.

Minuend [S1]: H16 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 1	0 1 1 0
BCD H code	0	0	1	6

− (Subtraction)

Subtrahend [S2]: H4 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0
BCD H code	0	0	0	4

Result [D]: H12 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0
BCD H code	0	0	1	2

Description

Subtracts the 4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data specified by S2 from the 4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data specified by S1.
The subtracted result is stored in D.

Minuend data		**Subtrahend data**		**Result**
(S1)	–	**(S2)**	→	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If the calculated result accidentally underflows, use of the **F48 (DB−)** instruction (8−digit BCD data subtraction) is recommended.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The area specified using the index modifier exceeds the limit.
  − The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 4-digit BCD data (underflows).

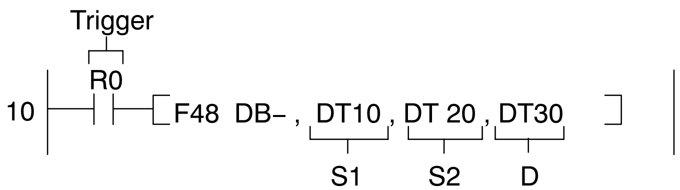
F48 (DB−) P48 (PDB−)

8-digit BCD data subtraction
[(S1+1, S1) − (S2+1, S2) → (D+1, D)]

Outline Subtracts one BCD data item that expresses an 8-digit decimal number (8-digit BCD H code) from another (minuend) and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P48 (PDB−)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 48 (DB−)
	DT 10
	DT 20
	DT 30

S1	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for minuend)
S2	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for subtrahend)
D	Lower 16-bit area for 8-digit BCD data (for result)

Operands

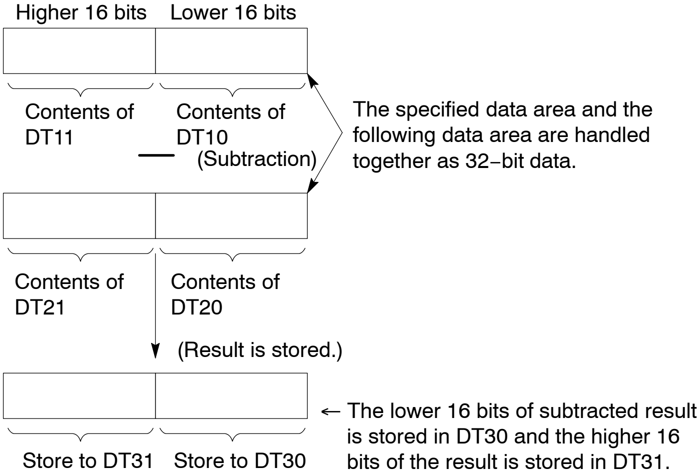
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	K	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available
K: K0 to K9

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Subtracts the contents of data registers DT21 and DT20 from the contents of data registers DT11 and DT10 when trigger R0 turns on. The subtracted result is stored in data registers DT31 and DT30.



Description

Subtracts the 8-digit BCD equivalent constant or 8-digit BCD data specified by S2 from the 8-digit BCD equivalent constant or the 8-digit BCD data specified by S1. The subtracted result is stored in D+1 and D.

Minuend data		Subtrahend data		Result
S1: lower 4-digit **S1+1:** higher 4-digit	−	**S2:** lower 4-digit **S2+1:** higher 4-digit	→	**D:** lower 4-digit **D+1:** higher 4-digit

When processing 8-digit BCD data, the higher 16-bit areas for 8-digit BCD data (S+1, D+1) are automatically determined once the lower 16-bit areas (S, D) are specified.

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 8-digit BCD data (underflows).

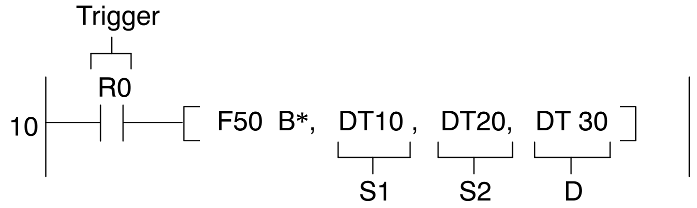
F50 (B*) P50 (PB*)

4-digit BCD data multiplication [S1 × S2 → (D+1, D)]

Outline Multiplies two BCD data items that express 4-digit decimal numbers (4-digit BCD H codes).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P50 (PB*)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 50 (B*)
	DT 10
	DT 20
	DT 30

S1	4-digit BCD equivalent constant or 16-bit area for BCD data (for multiplicand)
S2	4-digit BCD equivalent constant or 16-bit area for BCD data (for multiplier)
D	Lower 16-bit area for 8-digit BCD data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

When H (BCD) 8 is in DT10 and H (BCD) 2 is in DT20, as shown below.

Multiplicand [S1]: H8 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
BCD H code	0	0	0	8

×

Multiplier [S2]: H2 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0
BCD H code	0	0	0	2

R0: on

Result [D+1, D]: H16 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT31	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
BCD H code	0	0	0	0

Higher 4-digit area

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 0 0 0	0 0 0 0	0 0 0 1	0 1 1 0
BCD H code	0	0	1	6

Lower 4-digit area

The lower 16 bits of the 32−bit multiplication result are stored in the specified memory area (DT30), and the higher16 bits are stored in the area following the specified area (DT31).

Description

Multiplies the 4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data specified by S1 and S2. The multiplied result is stored in D+1 and D.

Multiplicand data (S1) × **Multiplier data (S2)** ⟶ **Multiplied result (D+1, D)**

The multiplied result is stored in the 8-digit area (32−bit area).

The higher 16-bit area (D+1) is automatically determined once the lower 16-bit area (D) is specified.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

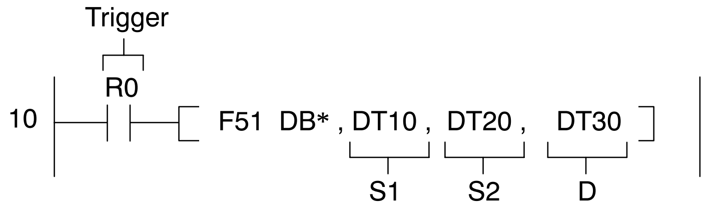
F51 (DB*) P51 (PDB*)

8-digit BCD data multiplication [(S1+1, S1) × (S2+1, S2) → (D+3, D+2, D+1, D)]

Outline Multiplies two BCD data items that express 8-digit decimal numbers (8-digit BCD H codes).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P51 (PDB*)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 51 (DB*)
	DT 10
	DT 20
	DT 30

S1	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for multiplicand)
S2	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for multiplier)
D	Lowest 16-bit area for 16-digit BCD data (for result)

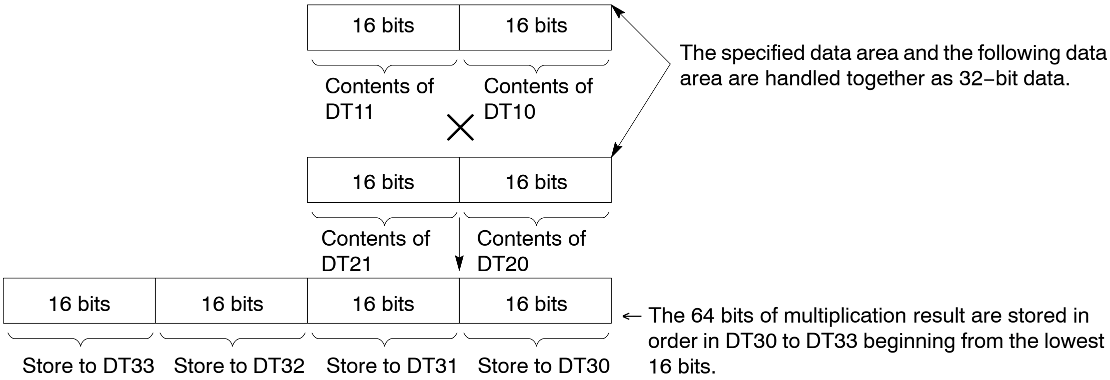
Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

Explanation of example



Description

Multiplies the 8-digit BCD equivalent constant or 8-digit BCD data specified by S1 and the one specified by S2. The multiplied result is stored in D+3, D+2, D+1, and D.

Multiplicand data		**Multiplier data**		**Result**
(S1+1, S1)	×	**(S2+1, S2)**	⟶	**(D+3, D+2, D+1, D)**

The multiplied result is stored in the 64-bit area (16−digit BCD).

When processing 8-digit BCD data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.The areas (D+3, D+2, D+1) other than the lowest 16-bit area (D) are automatically determined when the lowest 16-bit area is specified.

Flag conditions

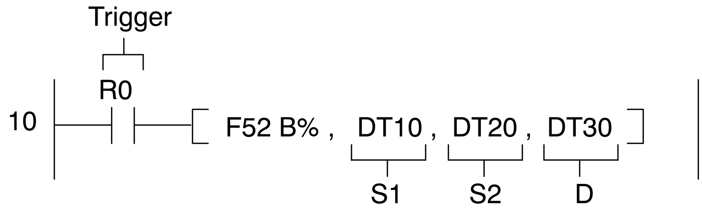
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F52 (B%) P52 (PB%)

4-digit BCD data division
[S1/S2 → D… (DT9015) or (DT90015)]

Outline Divides one BCD data item that expresses a 4-digit decimal number (4-digit BCD H code) by another (divisor).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P52 (PB%)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 52 (B%) DT 10 DT 20 DT 30

S1	4-digit BCD equivalent constant or 16-bit area for BCD data (for dividend)
S2	4-digit BCD equivalent constant or 16-bit area for BCD data (for divisor)
D	16-bit area for BCD data (for quotient) (Remainder is stored in special data register DT9015 or DT90015.)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	K	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available
K: K0 to K9

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Divides the contents of data register DT10 by the contents of data register DT20 when trigger R0 turns on. The quotient is stored in data register DT30 and the remainder is stored in special data register DT9015 (DT90015 for FP2/FP2SH/FP10SH).
When H (BCD) 15 is in DT10 and H (BCD) 4 is in DT20, as shown below.

Dividend [S1]: H15 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 0 0 0	0 0 0 0	0 0 0 1	0 1 0 1
BCD H code	0	0	1	5

Divisor [S2]: H4 (BCD) ÷

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0
BCD H code	0	0	0	4

X0: on

Quotient [D]: H3 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
BCD H code	0	0	0	3

Remainder: H3 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT9015/DT90015	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
BCD H code	0	0	0	3

Description

The 4-digit BCD equivalent constant or the 16-bit area for 4-digit BCD data specified by S1 is divided by the 4-digit BCD equivalent constant or the 16-bit area for 4-digit BCD data specified by S2. The quotient is stored in the area specified by D and the remainder is stored in a special data register DT9015 (DT90015 for FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH).

Dividend data		Divisor		Quotient		Remainder
(S1)	÷	**(S2)**	→	**(D)**	····	**(DT9015/ DT90015)**

With the FP0 T32,FP0R, FPΣ, FP−X, FP2, FP2SH, FP10SH and FP0 C10, C14, C16, C32/ FP−e, the numbers of the special data registers are different.

Type	Special data register
FP0 C10, C14, C16, C32/FP−e	DT9015
FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH	DT90015

Flag conditions

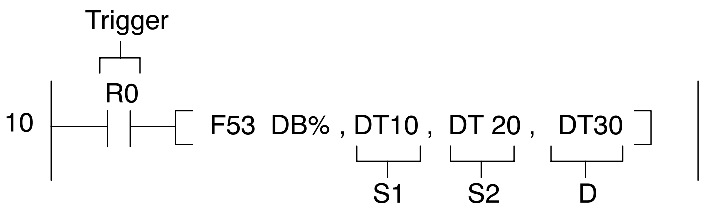
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
  - The 4-digit BCD equivalent constant or 4-digit BCD data for the divisor (specified by S2) is 0.
- = flag (R900B): Turns on for an instant when the calculated result (quotient) is recognized as “0”.

F53 (DB%) P53 (PDB%)

8-digit BCD data division [(S1+1, S1)/(S2+1, S2) → (D+1, D)… (DT9016, DT9015) or (DT90016, DT90015)]

Outline Divides one BCD data item that expresses an 8-digit decimal number (8-digit BCD H code) by another (divisor).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P53 (PDB%)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 53 (DB%) DT 10 DT 20 DT 30

S1	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for dividend)
S2	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD data (for divisor)
D	Lower 16-bit area for 8-digit BCD data (for quotient) (Remainder is stored in special data registers DT9016 and DT9015 or DT90016 and DT90015)

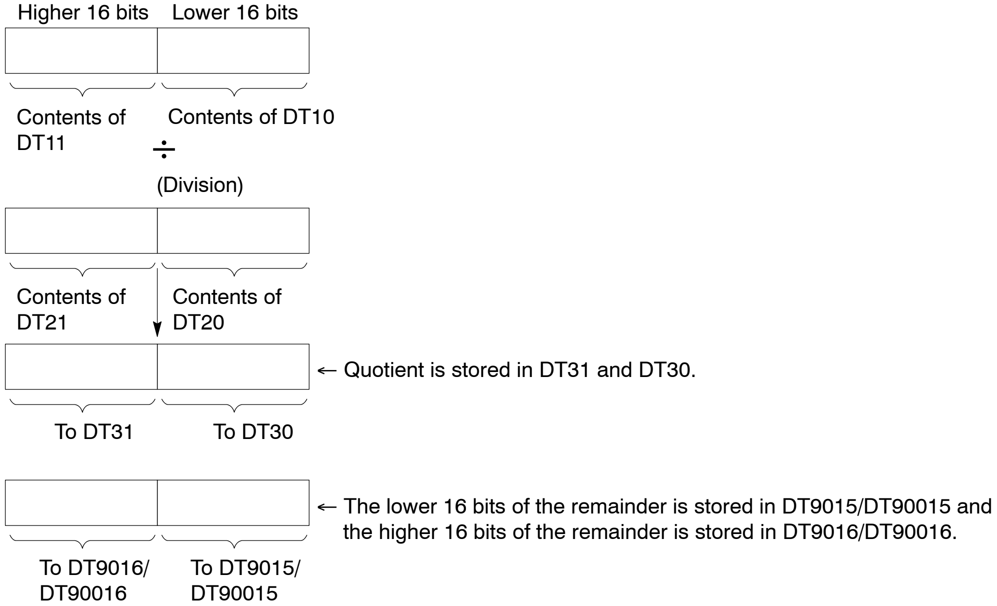
Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example



Description

The 8-digit BCD equivalent constant or the 8-digit BCD data specified by S1 is divided by the 8-digit BCD equivalent constant or the 8-digit BCD data specified by S2. The quotient is stored in the areas specified by D+1 and D, and the remainder is stored in special data registers DT9016 and DT9015 (DT90016 and DT90015 for FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH).

Dividend data		Divisor		Quotient		Remainder
(**S1:** lower 4-digit **S1+1:** higher 4-digit)	÷	(**S2:** lower 4-digit **S2+1:** higher 4-digit)	⟶	(**D:** lower 4-digit **D+1:** higher 4-digit)	....	DT9015/DT90015 DT9016/DT90016

When processing 8-digit BCD data, the higher 16-bit areas (S1+1, S2+1, D+1) are automatically determined once the lower 16-bit areas (S1, S2, D) are specified.

With the FP0 T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH and FP0 C10, C14, C16, C32/FP−e, the numbers of the special data registers are different.

Type	Special data register
FP0 C10, C14, C16, C32/FP−e	DT9015
FP0T32/FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH	DT90015

Flag conditions

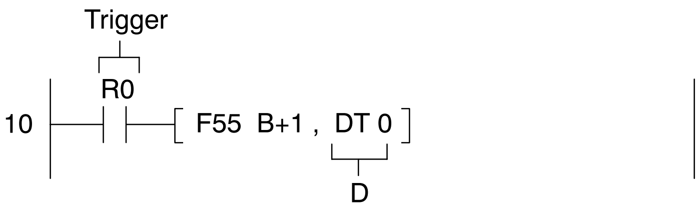
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
  - The 8-digit BCD equivalent constant or the 8-digit BCD data for the divisor (specified by S2) is 0.
- = flag (R900B): Turns on for an instant when the calculated result (quotient) is recognized as "0".

F55 (B+1)
P55 (PB+1)

4-digit BCD data increment
[D + 1 → D]

Outline Adds 1 to BCD data that expresses a 4-digit decimal number (4-digit BCD H code).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P55 (PB+1)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 55 (B+1) DT 0

D	16-bit area for 4-digit BCD data to be increased by 1

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Adds 1 to the contents of data register DT0 when trigger R0 turns on.

Original data [D]: H9 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1
BCD H code	0	0	0	9

+1

Result [D]: H10 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0
BCD H code	0	0	1	0

Description

Adds 1 to the 4-digit BCD data specified by D. The result is stored in D.

Original data				**Result**
(D)	**+**	**1**	⟶	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If the calculated result accidentally overflows, use of the **F56 (DB+1)** instruction (8-digit BCD data increment) is recommended.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

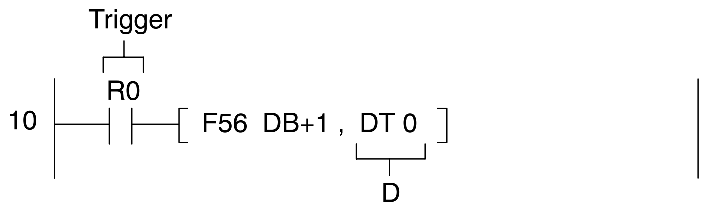
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 4-digit BCD data (overflows).

F56 (DB+1) P56 (PDB+1)

8-digit BCD data increment [(D+1, D) + 1 → (D+1, D)]

Outline Adds 1 to BCD data that expresses an 8-digit decimal number (8-digit BCD H code).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P56 (PDB+1)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 56 (DB+1) DT 0

D	Lower 16-bit area for 8-digit BCD data to be increased by 1

Operands

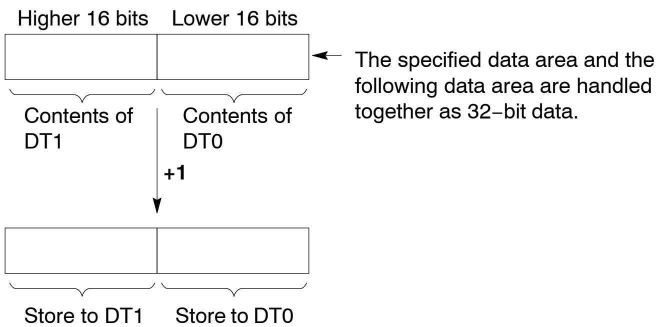
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Adds 1 to the contents (8−digit BCD data) of data registers DT1 and DT0 when trigger R0 turns on.



Description

Adds 1 to the 8-digit BCD data specified by D. The result is stored in D+1 and D.

Original data				**Result**
(D+1, D)	**+**	**1**	→	**(D+1, D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an overflow will result.

Under normal circumstances, do not allow an overflow to occur.

If an overflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 8-digit BCD data (overflows).

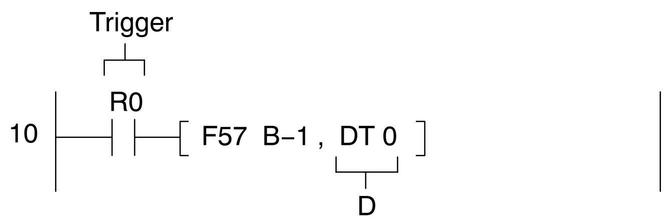
F57 (B−1) P57 (PB−1)

4-digit BCD data decrement [D − 1 → D]

Outline Subtracts 1 from BCD data that expresses a 4-digit decimal number (4-digit BCD H code).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P57 (PB−1)” is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction
10	ST R 0
11	F 57 (B−1)
	DT 0

D	16-bit area for BCD data to be decreased by 1

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Subtracts 1 from the contents of data register DT0 when trigger R0 turns on.

Original data [D]: H10 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0
BCD H code	0	0	1	0

⬇ −1

Result [D]: H9 (BCD)

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1
BCD H code	0	0	0	9

Description

Subtracts 1 from the 4-digit BCD data specified by D. The result is stored in D.

Original data				Result
(D)	**–**	**1**	→	**(D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If the calculated result accidentally underflow, use of the **F58 (DB−1)** instruction (8-digit BCD data decrement) is recommended.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 4-digit BCD data (underflows).

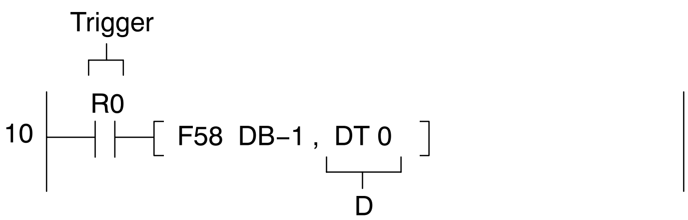
F58 (DB−1) P58 (PDB−1)

8-digit BCD data decrement
[(D+1, D) – 1 → (D+1, D)]

Outline

Subtracts 1 from BCD data that expresses an 8-digit decimal number (8-digit BCD H code).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P58 (PDB−1)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10 11	ST R 0 F 58 (DB−1) DT 0

D	Lower 16-bit area for 8-digit BCD data to be decreased by 1

Operands

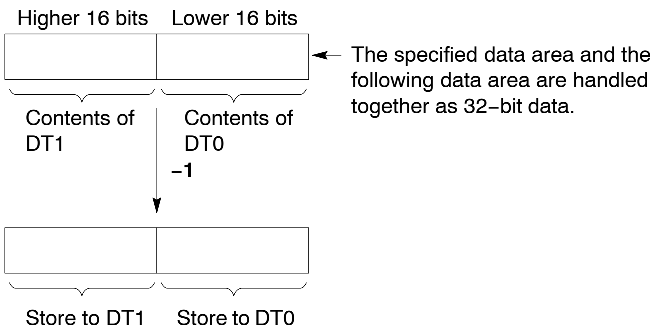
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

Explanation of example

Subtracts 1 from the contents (8−digit BCD data) of data registers DT1 and DT0 when trigger R0 turns on.



Description

Subtracts 1 from the 8-digit BCD data specified by D. The result is stored in D+1 and D.

Original data				**Result**
(D+1, D)	**–**	**1**	⟶	**(D+1, D)**

Precautions during programming

If the result of an arithmetic operation instruction does not fall within the range of values which can be handled, an underflow will result.

Under normal circumstances, do not allow an underflow to occur.

If an underflow occurs, the carry flag (special internal relay R9009) will turn on.

Flag conditions

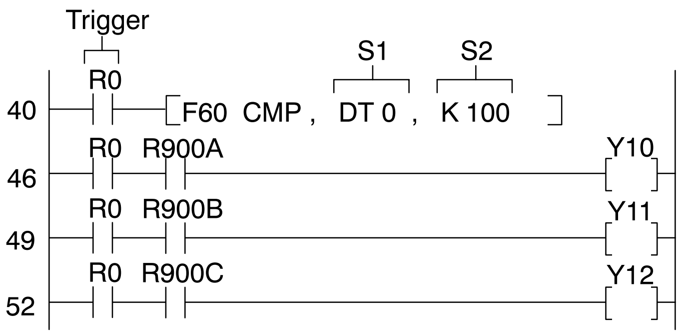
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data is not BCD data.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.
- Carry flag (R9009): Turns on for an instant when the calculated result exceeds the range of 8-digit BCD data (underflows).

F60 (CMP) P60 (PCMP) 16-bit data comparison

Outline The two specified 16−bit data are compared and the result is output to the special internal relay.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P60 (PCMP)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
40	ST R 0
41	F 60 (CMP)
	DT 0
	K 100
46	ST R 0
47	AN R 900A
48	OT Y 10
49	ST R 0
50	AN R 900B
51	OT Y 11
52	ST R 0
53	AN R 900C
54	OT Y 12

S1	16-bit equivalent constant or 16-bit area to be compared
S2	16-bit equivalent constant or 16-bit area to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A: Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Compares decimal constant K100 with the contents of data register DT0 when trigger R0 turns on.

When DT0 > K100, R900A turns on and external output relay Y10 turns on.

When DT0 = K100, R900B turns on and external output relay Y11 turns on.

When DT0 < K100, R900C turns on and external output relay Y12 turns on.

Description

Compares the 16-bit data specified by S1 with that specified by S2. The comparison result is output to the special internal relays R9009, R900A, R900B and R900C.

The following table lists the states of the carry flag (R9009), > flag (R900A), = flag (R900B), and < flag (R900C), depending on the relative sizes of S1 and S2.

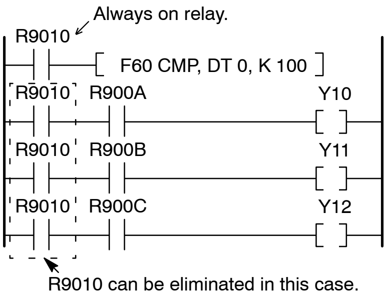
Comparison between S1 and S2	Flag			
	R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (carry flag)
S1 < S2	off	off	on	↕
S1 = S2	off	on	off	off
S1 > S2	on	off	off	↕

"↕": turns on or off according to the conditions

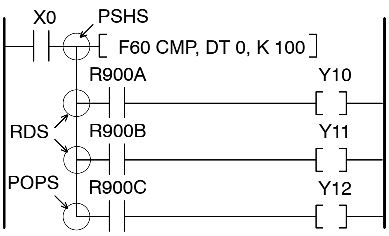
Execution condition (Trigger)

In this program example, the comparison will be performed only when R0 turns on.

If ongoing comparison is necessary, relay R9010, which is always on, should be used in the execution conditions (trigger).



You can also program the above using the **PSHS**, **RDS**, and **POPS** instructions.



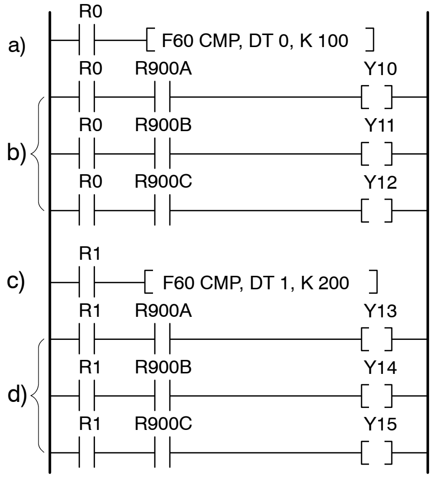
This is a program in which operation is the same as the above program example.

Precautions when using two or more comparison instructions

The comparison instruction flags R900A to R900C are updated with each execution of the comparison instruction.

If you use two or more comparison instructions in your program, be sure to use the flags immediately after each comparison instruction, by employing output relays or internal relays.

Example: Compares DT0 with K100, and DT1 with K200.

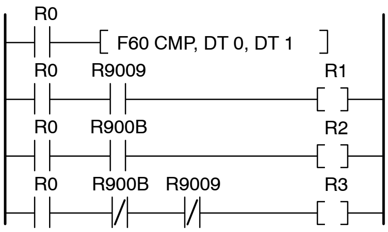


The comparison result for a) is output to the output relays (Y10, Y11, and Y12) of program b).
The comparison result for c) is output to the output relays (Y13, Y14, and Y15) of program d).

Precautions when comparing BCD or external data

When comparing special data, such as BCD or unsigned binary (0 to FFFF), construct your program as shown in the program example below, using special internal relays R900B and R9009.

Example: Compares BCD data in DT0 and DT1.



①....When DT0 < DT1, internal relay R1 turns on
②....When DT0 = DT1, internal relay R2 turns on
③....When DT0 > DT1, internal relay R3 turns on

Flag operation when comparing BCD data or unsigned 16−bit data (0 to FFFF)

Comparison between S1 and S2	Flag			
	R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (carry flag)
S1 < S2	↕	off	↕	on
S1 = S2	off	on	off	off
S1 > S2	↕	off	↕	off

"↕": turns on or off according to the conditions

For example, when S1 = H8000 and S2 = H1000, R900A will turn off and R900C will turn on. For this reason, the correct comparison result will not be obtained in a program which uses R900A and R900C.

S1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BCD	8				0				0				0			

S2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
BCD	1				0				0				0			

Flag conditions

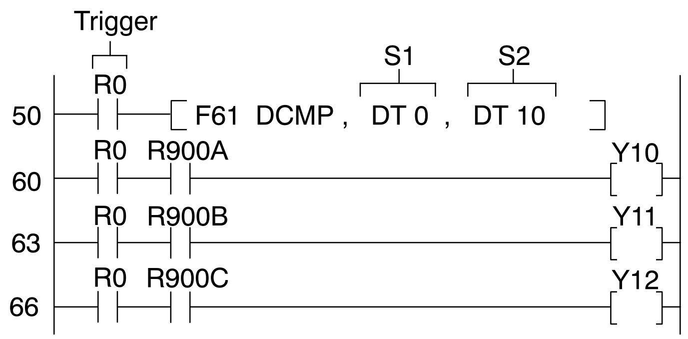
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F61 (DCMP) P61 (PDCMP) 32-bit data comparison

Outline The two specified 32−bit data are compared and the result is output to the special internal relay.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P61 (PDCMP)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
50	ST R 0
51	F 61 (DCMP)
	DT 0
	DT 10
60	ST R 0
61	AN R 900A
62	OT Y 10
63	ST R 0
64	AN R 900B
65	OT Y 11
66	ST R 0
67	AN R 900C
68	OT Y 12

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Compares the content (32−bit data) of data registers DT11 and DT10 with the content (32−bit data) of data registers DT1 and DT0 when trigger R0 turns on.

When (DT1 and DT0) > (DT11 and DT10), R900A turns on and external output relay Y10 turns on.

When (DT1 and DT0) = (DT11 and DT10), R900B turns on and external output relay Y11 turns on.

When (DT1 and DT0) < (DT11 and DT10), R900C turns on and external output relay Y12 turns on.

Description

Compares the 32-bit data or 32-bit equivalent constant specified by S1 with that specified by S2. The comparison result is output to special internal relays R9009, R900A, 900B, and R900C.

The following table lists the states of the carry flag (R9009), > flag (R900A), = flag (R900B), and < flag (R900C), depending on the relative sizes of (S1+1, S1) and (S2+1, S2).

Comparison between (S1+1, S1) and (S2+1, S2)	Flag			
	R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (carry flag)
(S1+1, S1) < (S2+1, S2)	off	off	on	↕
(S1+1, S1) = (S2+1, S2)	off	on	off	off
(S1+1, S1) > (S2+1, S2)	on	off	off	↕

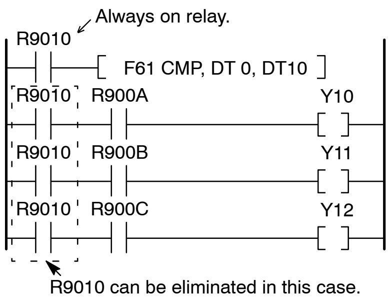
"↕": turns on or off according to the conditions

When processing 32-bit data, the higher 16-bit areas (S1+1, S2+1) are automatically determined once the lower 16-bit areas (S1, S2) are specified.

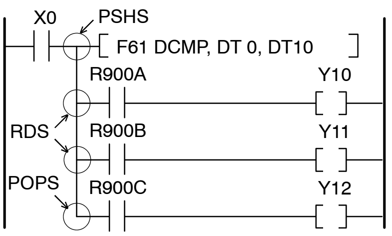
Execution condition (Trigger)

In this program example, the comparison will be performed only when R0 turns on.

If ongoing comparison is necessary, relay R9010, which is always on, should be used in the execution conditions.



You can also program the above using the **PSHS**, **RDS**, and **POPS** instructions.



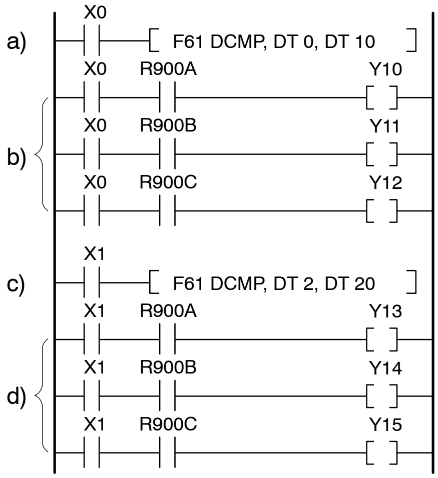
This is a program in which operation is the same as the above program.

Precautions when using two or more comparison instructions

The comparison instruction flags R900A to R900C are updated with each execution of the comparison instruction.

If you use two or more comparison instructions in your program, be sure to use the flags immediately after each comparison instruction, by employing output relays or internal relays.

Example: Compares DT1 and DT0 with DT11 and DT10, and DT3 and DT2 with DT21 and DT20.

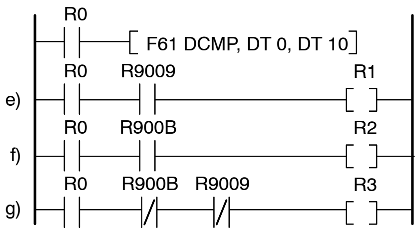


The comparison result for a) is output to the output relays (Y10, Y11, and Y12) of program b).
The comparison result for c) is output to the output relays (Y13, Y14, and Y15) of program d).

Precautions when comparing BCD or external data

When comparing special data, such as BCD or unsigned binary (0 to FFFFFFFF), flags R9009, R900A, R900B, and R900C work as shown in the table below. In this case, construct your program as shown in the program example below, using special internal relays R900B and R9009.

Example: Compares BCD data in (DT1, DT0) and (DT11, DT10).



e)....When (DT1, DT0) < (DT11, DT10), internal relay R1 turns on
f)....When (DT1, DT0) = (DT11, DT10), internal relay R2 turns on
g)....When (DT1, DT0) > (DT11, DT10), internal relay R3 turns on

Flag operation when comparing BCD data or unsigned 32−bit data (0 to FFFFFFFF)

Comparison between (S1+1, S1) and (S2+1, S2)	Flag			
	R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (carry flag)
(S1+1, S1) < (S2+1, S2)	↕	off	↕	on
(S1+1, S1) = (S2+1, S2)	off	on	off	off
(S1+1, S1) > (S2+1, S2)	↕	off	↕	off

"↕": turns on or off according to the conditions

For example, if an **F61 (DCMP)** instruction is executed when S1 = H80000000 (K−2147483648) and S2 = H10000001 (K+268435457), the result will be S1<S2. Thus R900A will turn off and R900C will turn on. In a program which uses R900A and R900C, the correct comparison result will not be obtained.

Flag conditions

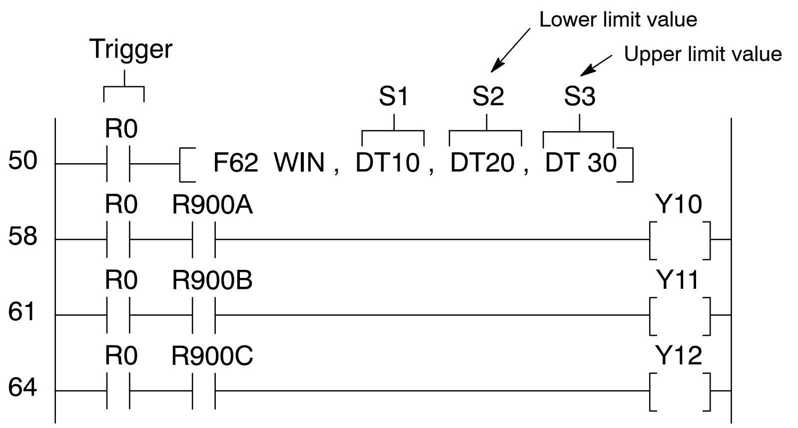
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F62 (WIN) P62 (PWIN) 16-bit data band comparison

Outline Compares one 16-bit data item with the data band specified by two other 16-bit data items and the comparison result is output to the special internal relay.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P62 (PWIN)” is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
50	ST	R	0
51	F 62		(WIN)
	DT		10
	DT		20
	DT		30
58	ST	R	0
59	AN	R	900A
60	OT	Y	10
61	ST	R	0
62	AN	R	900B
63	OT	Y	11
64	ST	R	0
65	AN	R	900C
66	OT	Y	12

S1	16-bit equivalent constant or 16-bit area to be compared
S2	16-bit equivalent constant or 16-bit area for lower limit
S3	16-bit equivalent constant or 16-bit area for upper limit

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	A

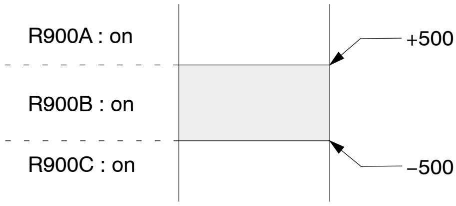
A: Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Compares the contents of data register DT10 with the contents of data register DT20 (lower limit of the data band) and data register DT30 (upper limit of the data band) when trigger R0 turns on.

Example: When K−500 is in DT20 and K500 is in DT30, as shown below.



When DT10 is K−680, R900C turns on and external output relay Y12 goes on.
When DT10 is K−500, R900B turns on and external output relay Y11 goes on.
When DT10 is K256, R900B turns on and external output relay Y11 goes on.
When DT10 is K680, R900A turns on and external output relay Y10 goes on.

Description

Compares the 16-bit equivalent constant or 16-bit data specified by S1 with the data band specified by S2 and S3. This instruction checks whether S1 is in the data band between S2 (lower limit) and S3 (upper limit), larger than S3, or smaller than S2. The comparison result is output to special internal relays R9009, R900A, R900B, and R900C.

The following table lists the states of the R9009, R900A, R900B and R900C.

Comparison between S1, S2 and S3	Flag			
	R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (carry flag)
S1 < S2	off	off	on	———
S2 ≦ S1 ≦ S3	off	on	off	———
S3 < S1	on	off	off	———

Precaution during programming

Set it so that the value of the lower limit is less than the value of the upper limit (S2 ≦ S3).

Flag conditions

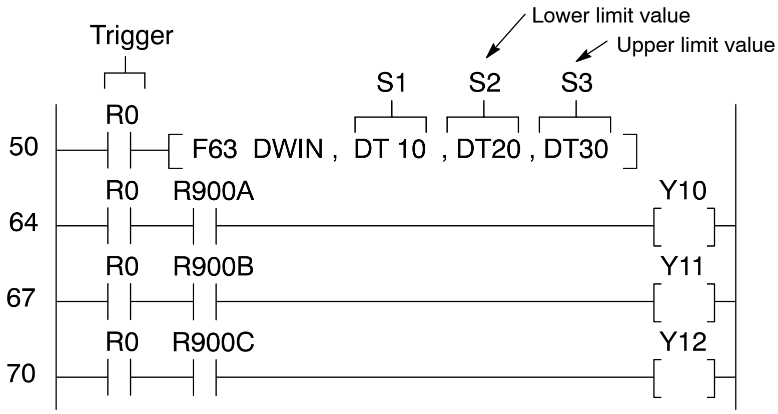
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S2 > S3.

F63 (DWIN) P63 (PDWIN) 32-bit data band comparison

Outline Compares one 32-bit data item with the data band specified by two other 32-bit data items and the comparison result is output to the special internal relay.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P63 (PDWIN)” is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
50	ST	R	0
51	F 63		(DWIN)
	DT		10
	DT		20
	DT		30
64	ST	R	0
65	AN	R	900A
66	OT	Y	10
67	ST	R	0
68	AN	R	900B
69	OT	Y	11
70	ST	R	0
71	AN	R	900C
72	OT	Y	12

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data to be compared
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data for lower limit
S3	32-bit equivalent constant or lower 16-bit area of 32-bit data for upper limit

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S3	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A

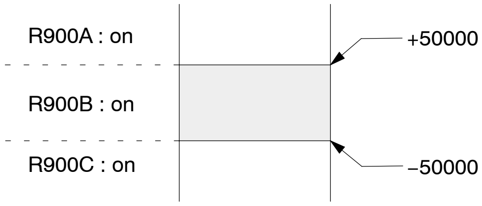
A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Compares the contents of data registers DT11 and DT10 with the contents of data registers DT21 and DT20 (lower limit of the data band) and data registers DT31 and DT30 (upper limit of the data band), when trigger R0 turns on.

Example: When K−50000 is in DT21 and DT20 and K50000 is in DT31 and DT30, as shown below.



When (DT11, DT10) is K−68000, R900C turns on and external output relay Y12 goes on.
When (DT11, DT10) is K−50000, R900B turns on and external output relay Y11 goes on.
When (DT11, DT10) is K25600, R900B turns on and external output relay Y11 goes on.
When (DT11, DT10) is K68000, R900A turns on and external output relay Y10 goes on.

Description

Compares the 32-bit equivalent constant or 32-bit data specified by S1 with the data band specified by S2 and S3. This instruction checks whether S1 is in the data band between S2 (lower limit) and S3 (upper limit), larger than S3, or smaller than S2. The comparison result is output to the special internal relays R9009, R900A, R900B, and R900C.

The following table lists the states of the R9009, R900A, R900B and R900C.

Comparison between (S1+1, S1), (S2+1, S2) and (S3+1, S3)	Flag			
	R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (carry flag)
(S1+1, S1) < (S2+1, S2)	off	off	on	———
(S2+1, S2) ≦ (S1+1, S1) ≦ (S3+1, S3)	off	on	off	———
(S3+1, S3) < (S1+1, S1)	on	off	off	———

Precaution during programming

Set it so that the value of the lower limit (S2 + 1, S2) is less than the value of the upper limit (S3 + 1, S3) [(S2+1, S2) ≦ (S3+1, S3)].

Flag conditions

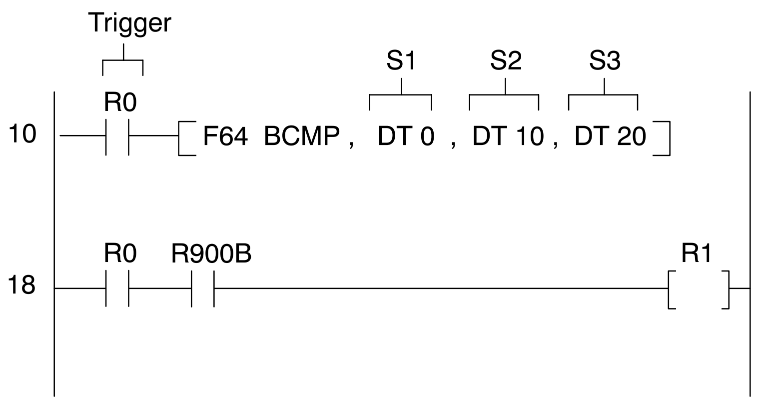
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - (S2+1, S2) > (S3+1, S3).

F64 (BCMP) P64 (PBCMP) Block data comparison

Outline Compares one specified data block with another in byte units.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P64 (PBCMP)" is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	0
11	F 64		(BCMP)
	DT		0
	DT		10
	DT		20
18	ST	R	0
19	AN	R	900B
20	OT	R	1

S1	16-bit equivalent constant or 16-bit area (specifies starting byte positions and number of bytes to be compared)
S2	Starting 16-bit area to be compared
S3	Starting 16-bit area to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S3	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

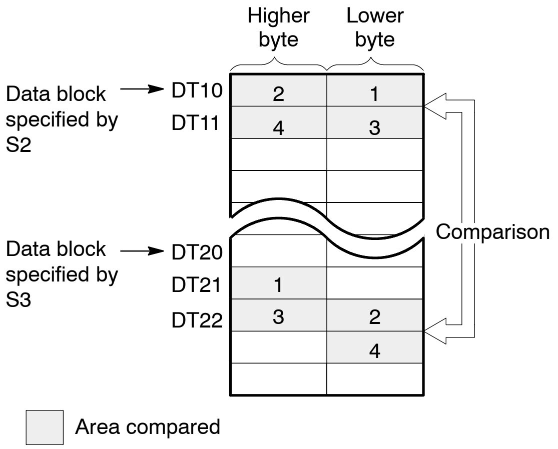
A: Available
N/A: Not Available

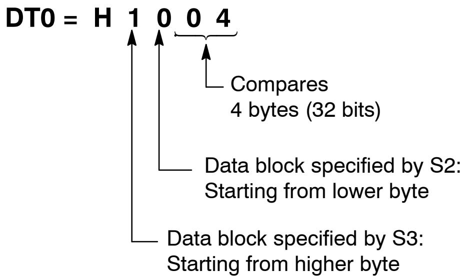
(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Compares the data block of data register DT10 (4 bytes from DT10 lower order byte) with data register DT20 (4 bytes from DT20 higher order byte) according to the comparison condition in data register DT0 when trigger R0 turns on. When the contents of the two data blocks are the same, internal relay R1 turns on.

If H1004 is entered in DT0, the two blocks are as follows.





Description

Compares the contents of the data block specified by S2 with the contents of the data block specified by S3 according to content specified by S1.

When the comparison result is S2 = S3, special internal relay R900B (=flag) turns on.

S1 is the control data that determines factors such as the size of the comparison.

How to specify control data “S1”

S1 specifies the starting byte position and the number of bytes to be compared using 4-digit BCD data as follows:

S1 = H □ □ □ □

Number of bytes to be compared
range: H01 to H99 (BCD)

Starting byte position for data block specified by S2
1: Starting from higher order byte
0: Starting from lower order byte

Starting byte position for data block specified by S3
1: Starting from higher order byte
0: Starting from lower order byte

Setting example:

To specify the 4 bytes beginning with the lower byte of the data block specified by S2 and the 4 bytes beginning with the upper byte of the data block specified by S3, set S1 to H1004.

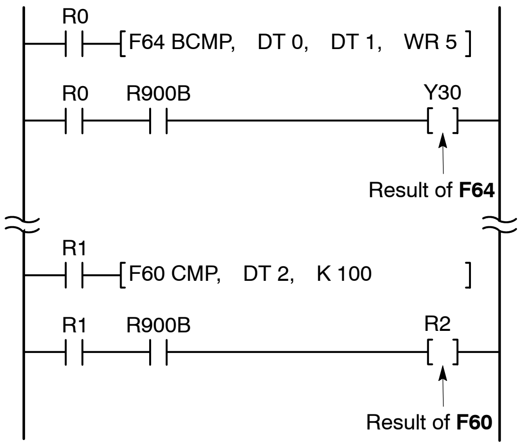
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data specified by S1 is not BCD data.
  - The specified data block area exceeds the limit.
    * For FP2SH and EP10SH, the error flag (R9007) turns on only when these operation errors occurs.

Precautions during programming

The flag R900B used for the compare instruction is renewed each time a compare instruction is executed. Accordingly:

- The program that uses R900B should be just after the **F64 (BCMP)** instruction.
- Output to an output relay or internal relay and save the result.



☞ Note

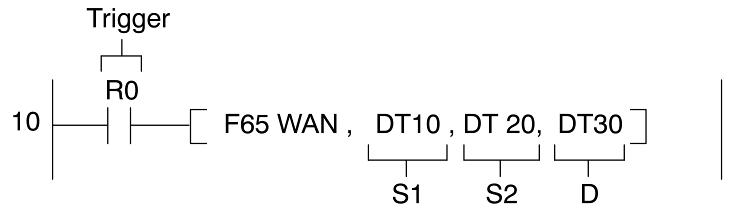
As shown in the above program, be sure to have the comparison internal relay before flag R900B. However, if you are using R9010 (on all the time), then it is unnecessary to have the comparison internal relay before R900B.

F65 (WAN) P65 (PWAN) 16-bit data AND

Outline Performs bit-wise AND operation on two 16-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P65 (PWAN)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 65		(WAN)
	DT		10
	DT		20
	DT		30

S1	16-bit equivalent constant or 16-bit area
S2	16-bit equivalent constant or 16-bit area
D	16-bit area for storing AND operation result

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Performs AND operation on each bit in data registers DT10 and DT20 when trigger R0 turns on.
The AND operation result is stored in data register DT30.

[S1]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 1 0 0	1 1 0 1	1 0 1 1	1 0 0 1

[S2]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1

R0: on

[D]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 0 0 0	0 0 0 0	1 0 1 1	1 0 0 1

Description

Performs AND operation on each bit in the 16-bit equivalent constant or 16-bit data specified by S1 and S2. The AND operation result is stored in the 16-bit area specified by D.

(S1) ∧ (S2) → (D)

You can use this instruction to turn off certain bits of the 16-bit data.

AND operation

The AND operation is shown below.

S1	S2	D
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Flag conditions

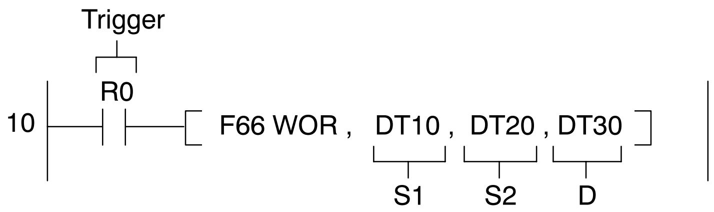
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F66 (WOR) P66 (PWOR) 16-bit data OR

Outline Performs bit-wise OR operation on two 16-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P66 (PWOR)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 66 (WOR)
	DT 10
	DT 20
	DT 30

S1	16-bit equivalent constant or 16-bit area
S2	16-bit equivalent constant or 16-bit area
D	16-bit area for storing OR operation result

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Performs OR operation on each bit in data registers DT10 and DT20 when trigger R0 turns on.
The OR operation result is stored in data register DT30.

[S1]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 1 0 0	1 1 0 1	1 0 1 1	1 0 0 1

[S2]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1

R0: on

[D]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 1 0 0	1 1 0 1	1 1 1 1	1 1 1 1

Description

Performs OR operation on each bit in the 16-bit equivalent constant or 16-bit data specified by S1 and S2. The OR operation result is stored in the 16-bit area specified by D.

(S1) ∨ (S2) → (D)

You can use this instruction to turn on certain bits of the 16-bit data.

OR operation

The OR operation is shown below.

S1	S2	D
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

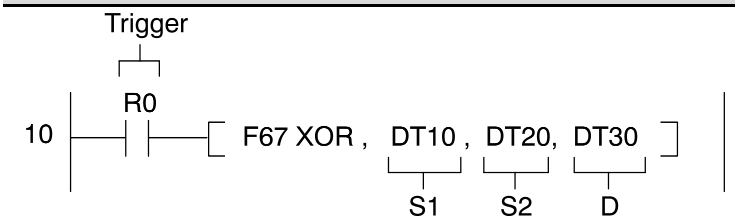
F67 (XOR)
P67 (PXOR)

16-bit data exclusive OR

Outline Performs bit-wise exclusive OR operation on two 16-bit data items.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P67 (PXOR)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 67		(XOR)
	DT		10
	DT		20
	DT		30

S1	16-bit equivalent constant or 16-bit area
S2	16-bit equivalent constant or 16-bit area
D	16-bit area for storing exclusive OR operation result

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Performs exclusive OR operation on each bit in data registers DT10 and DT20 when trigger R0 turns on. The exclusive OR operation result is stored in data register DT30.

[S1]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 1 0 0	1 1 0 1	1 0 1 1	1 0 0 1

[S2]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1

R0: on

[D]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	0 1 0 0	1 1 0 1	0 1 0 0	0 1 1 0

Description

Performs exclusive OR operation on each bit in the 16-bit equivalent constant or 16-bit data specified by S1 and S2. The exclusive OR operation result is stored in the 16-bit area specified by D.

$\{(S1) \wedge (\overline{S2})\} \vee \{(\overline{S1}) \wedge (S2)\} \rightarrow (D)$

Detects the bits whose on and off states do not match.

If the values of S1 and S2 are equal, all the bits of the data specified by D become 0.

Exclusive OR operation

The exclusive OR operation is shown below.

S1	S2	D
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

Flag conditions

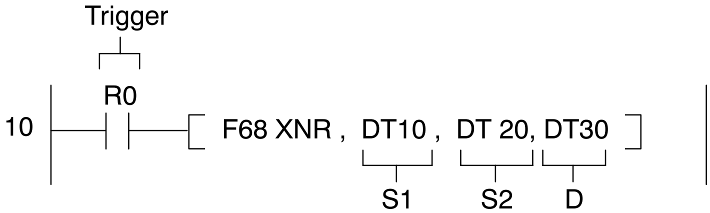
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F68 (XNR) P68 (PXNR)

16-bit data exclusive NOR

Outline Performs bit-wise exclusive NOR operation on two 16-bit data items. For the FP0R/FPΣ/FP-X/FP0/FP−e, the P type high−level instruction “P68 (PXNR)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F 68 (XNR) DT 10 DT 20 DT 30

S1	16-bit equivalent constant or 16-bit area
S2	16-bit equivalent constant or 16-bit area
D	16-bit area for storing exclusive NOR operation result

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

When the trigger R0 is on, if the values of the bits for the same positions with regard to the contents of data registers DT10 and DT20 are equal, the bit for that position is turned on (1) for data register DT30. If the values are not equal, the bit is turned off (0).

[S1]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 1 0 0	1 1 0 1	1 0 1 1	1 0 0 1

[S2]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1

R0: on

[D]

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT30	1 0 1 1	0 0 1 0	1 0 1 1	1 0 0 1

Description

Performs exclusive NOR operation on each bit in the 16-bit equivalent constant or 16-bit data specified by S1 and S2. The exclusive NOR operation result is stored in the 16-bit area specified by D.

$\{(S1) \wedge (S2)\} \vee \{(\overline{S1}) \wedge (\overline{S2})\} \rightarrow (D)$

Detects the bits whose on and off states match.

If the values of S1 and S2 are equal, all the bits of the data specified by D become 1.

Exclusive NOR operation

The exclusive NOR operation is shown below.

S1	S2	D
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

Flag conditions

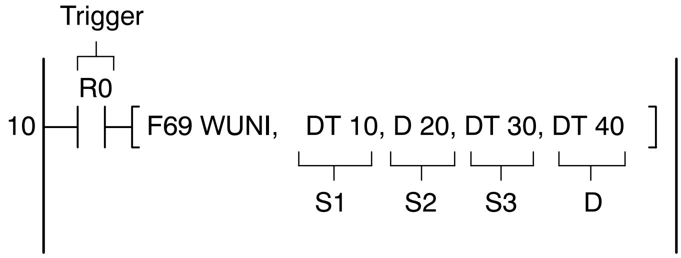
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F69 (WUNI) P69 (PWUNI) 16-bit data unite

Outline Unites two 16-bit data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P69 (PWUNI)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	10
11	F69	(WUNI)
	DT	10
	DT	20
	DT	30
	DT	40

S1	16-bit equivalent constant or 16-bit area
S2	16-bit equivalent constant or 16-bit area
S3	16-bit area which stores mask data for combination or 16-bit equivalent constant data.
D	16-bit area for storing calculated result

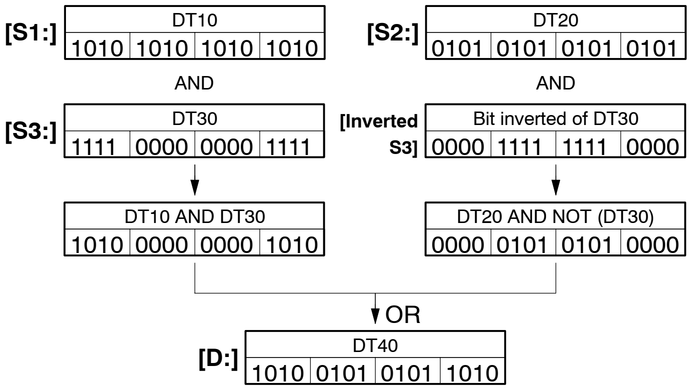
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example



Description

The two groups of word data specified by S1 and S2 are combined by bit unit processing using the mask data specified by S3 and stored in the area specified by D.

(S1 $\wedge$ S3) $\vee$ (S2 $\wedge$ S3) $\rightarrow$ (D)

When S3 is H0, the contents of S2 stored in the D.

When S3 is HFFFF, the contents of S1 stored in the D.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

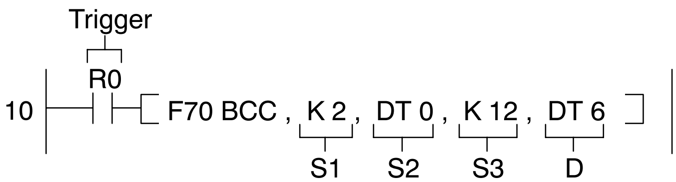
F70 (BCC)
P70 (PBCC)

Block check code calculation

Outline Calculates Block Check Code (BCC).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P70 (PBCC)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 70		(BCC)
	K		2
	DT		0
	K		12
	DT		6

S1	16-bit equivalent constant or 16-bit area (specifies BCC calculation method)
S2	Starting 16-bit area to calculate BCC
S3	16-bit equivalent constant or 16-bit area (specifies number of bytes for BCC calculation)
D	16-bit area for storing BCC

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Calculates the Block Check Code (BCC) of 12 bytes of ASCII data starting from data register DT0, via an exclusive OR operation, when trigger R0 turns on. The Block Check Code (BCC) is stored in the lower byte of data register DT6.

Description

Creates Block Check Code (BCC) from the starting position for the calculation specified by “S1” and “S2” using the calculation method specified by “S1”, and stores the result at the position specified by “D” and “S1” according to the conversion method specified by “S1”.

H □ □ □ □

- BCC calculation method
  - 0: Addition
  - 1: Subtraction
  - 2: Exclusive OR operation
  - A: CRC−16
- Starting byte position for calculation (No. of bytes from “S2”) Note 2)
  - 0 to F
- Starting byte position for storing results (No. of bytes from “D”) Note 2)
  - 0 to F
- Conversion data Note 2)
  - 0: Binary data (CRC : 2 bytes / Not CRC : 1 bytes)
  - 1: ASCII code (2 bytes)

Note 1) If CRC−16 is specified as the calculation method, ASCII code cannot be specified for the conversion data.

How to specify control data “S1”

Note 2) This can be used with the FP0R, FP−X (V2.00 or more) and FPΣ (V3.10 or more).

How to calculate the Block Check Code (BCC)

If BCC calculation method specified by “S1” is CRC,
The following generation polynomial is used and calculated (The same as MODBUS−RTU).

The generation polynomial : X^15+X^13+1 (H A001)

Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

Σ Error flag (R9008): Turns on for an instant when:

- BCC calculation method specified by “S1” is outside the specification range
- Conversion data specified by “S1” is outside the specification range

Application example 1

In this example, the block check code of the message being sent, ”%01#RCSX0000”, is calculated and is added after the message.

Transmission is done using ASCII codes.

BCC is calculated as an exclusive logical OR.

The message should be stored in the memory area as shown below.

Data register	DT6	DT5	DT4	DT3	DT2	DT1	DT0
ASCII HEX code		3 0 3 0	3 0 3 0	5 8 5 3	4 3 5 2	2 3 3 1	3 0 2 5
ASCII character		0 0	0 0	X S	C R	# 1	0 %

12 bytes

Block Check Code (BCC)

The **F70 (BCC)** instruction is as shown below.

[F70 BCC, K 2 , DT 0 , K 12 , DT 6]

S1: Exclusive logical OR

S2: Start of target data

S3: Length of target data

D: Calculation results

When this is executed, BCC (H 1D) is stored in the last byte of DT6 of D.

How to calculate the Block Check Code (BCC)

Exclusive ORing calculates the Block Check Code (BCC) with each ASCII character.

Character	ASCII HEX code	ASCII BIN code	
%	2 5	0 0 1 0 0 1 0 1	
0	3 0	0 0 1 1 0 0 0 0	Exclusive ORing
1	3 1	0 0 1 1 0 0 0 1	Exclusive ORing
#	2 3	0 0 1 0 0 0 1 1	Exclusive ORing
R	5 2	0 1 0 1 0 0 1 0	Exclusive ORing
C	4 3	0 1 0 0 0 0 1 1	Exclusive ORing
S	5 3	0 1 0 1 0 0 1 1	Exclusive ORing
X	5 8	0 1 0 1 1 0 0 0	Exclusive ORing
0	3 0	0 0 1 1 0 0 0 0	Exclusive ORing
0	3 0	0 0 1 1 0 0 0 0	Exclusive ORing
0	3 0	0 0 1 1 0 0 0 0	Exclusive ORing
0	3 0	0 0 1 1 0 0 0 0	Exclusive ORing

Exclusive OR operation

S1	S2	D
0	0	0
0	1	1
1	0	1
1	1	0

↓ calculation

Block Check Code (BCC)

ASCII HEX code	1	D
ASCII BIN code	0 0 0 1	1 1 0 1

→ This calculation result (H1D) is stored in the lower byte of DT6.

Application example 2

In this example, the block check code of the message being sent, ”%01#RCSX0000”, is calculated and is added at the end of the message.

Calculation method: Addition, conversion data: Binary data

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0000	0000	0000	3030	3030	5853	4352	2331	3025
				00	00	XS	CR	#1	0%

DT10 = H 0C00

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0000	0000	00A9	3030	3030	5853	4352	2331	3025
				00	00	XS	CR	#1	0%

Calculation method: Addition, conversion data: ASCII codes

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0000	0000	0000	3030	3030	5853	4352	2331	3025
				00	00	XS	CR	#1	0%

DT10 = H 1C00

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0000	0000	3941	3030	3030	5853	4352	2331	3025
			9A	00	00	XS	CR	#1	0%

Calculation method: Addition, conversion data: ASCII codes

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0000	0030	3030	3058	5343	5223	3130	2500	0000
		0	00	0X	SC	R#	10	%	

DT10 = H 1F30

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0039	4130	3030	3058	5343	5223	3130	2500	0000
	9	A0	00	0X	SC	R#	10	%	

Calculation method: CRC, conversion data: Binary data

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0000	0000	0000	3030	3030	5853	4352	2331	3025
				00	00	XS	CR	#1	0%

DT10 = H 0C0A

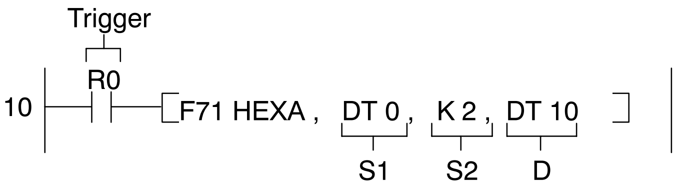
DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0000	0000	0000	2E0A	3030	3030	5853	4352	2331	3025
				00	00	XS	CR	#1	0%

F71 (HEXA) P71 (PHEXA) Hexadecimal data → ASCII code

Outline Converts 16-bit data to ASCII code that expresses the equivalent hexadecimals.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P71 (PHEXA)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 71 (HEXA)
	DT 0
	K 2
	DT 10

S1	Starting 16-bit area for hexadecimal number (source)
S2	16-bit equivalent constant or 16-bit area to specify number of source data bytes to be converted
D	Starting 16-bit area for storing ASCII code (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Converts 2 bytes of data stored in data register DT0 to ASCII codes that express the equivalent hexadecimals when trigger R0 turns on. The converted data is stored in data registers DT11 and DT10.

	DT0			
Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
Binary data	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1
Hexadecimal	A	B	C	D

⬇

	DT11				DT10			
ASCII HEX code	4	2	4	1	4	4	4	3
ASCII character	B		A		D		C	

Description

Converts the data starting from the 16-bit area specified by S1 to ASCII codes that express the equivalent hexadecimals.

The converted result is stored in the area starting from the 16-bit area specified by D.

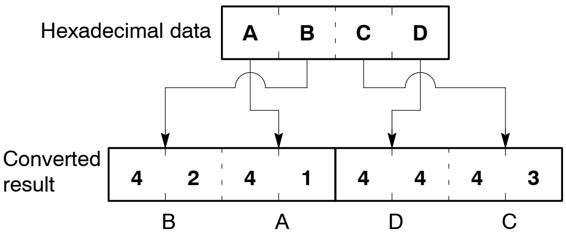
S2 specifies the number of source data bytes to be converted.

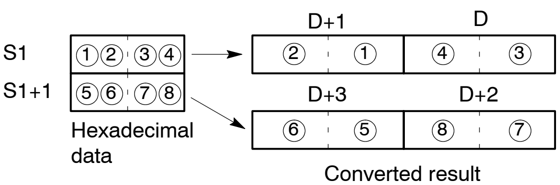
Since ASCII code requires eight bits (one byte) to express one hexadecimal character, the data length when converted to ASCII code becomes double the source data.

Precautions during programming

The two characters that make up one byte are interchanged when stored.

Two bytes are converted as one segment of data.





Flag conditions

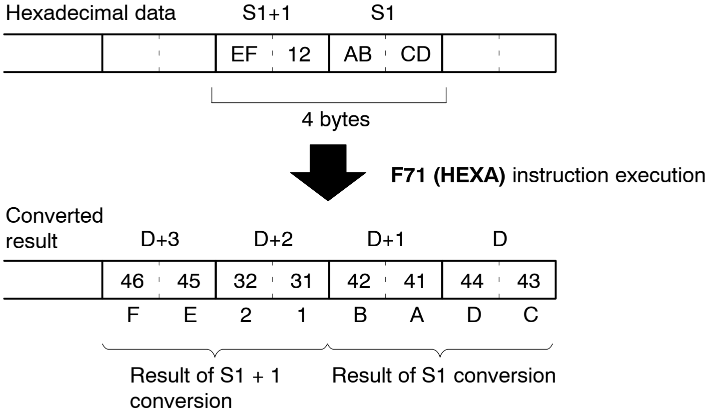
Σ Error flag (R9007): Turns on and stays on when:

Σ Error flag (R9008): Turns on for an instant when:
- The area specified using the index modifier exceeds the limit.
- The number of bytes specified by S2 exceeds the area specified by S1.
- The converted result exceeds the area specified by D.
- The data specified by S2 is recognized as “0”.

Conversion example

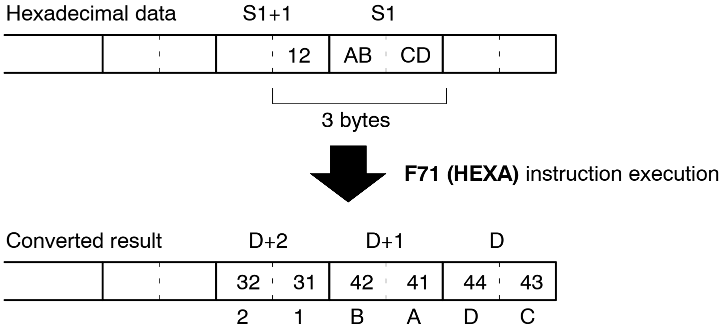
The following shows conversion of hexadecimal data to ASCII codes.

Conversion of four bytes of data (S2 = K4)



Conversion of three bytes of data (S2 = K3)

Since “byte” is specified as the unit, it is possible to convert only the last byte of the data for one word if desired.



ASCII HEX codes to express hexadecimal characters

Hexadecimal number	ASCII HEX code
0	H30
1	H31
2	H32
3	H33
4	H34
5	H35
6	H36
7	H37
8	H38
9	H39
A	H41
B	H42
C	H43
D	H44
E	H45
F	H46

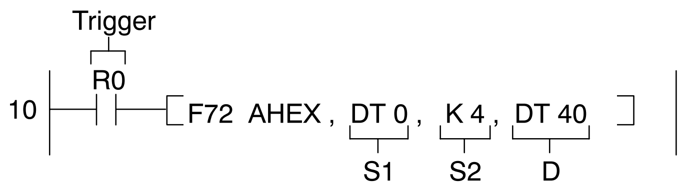
F72 (AHEX)
P72 (PAHEX)

ASCII code → Hexadecimal data

Outline Converts ASCII code that expresses hexadecimal characters to hexadecimal data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P72 (PAHEX)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 72 (AHEX)
	DT 0
	K 4
	DT 40

S1	Starting 16-bit area for ASCII code (source)
S2	16-bit equivalent constant or 16-bit area to specify number of source data bytes to be converted
D	Starting 16-bit area for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Converts 4 ASCII codes stored in data registers DT0 and DT1 to hexadecimal numbers when trigger R0 turns on. The converted data is stored in data register DT40.

	DT1				DT0			
ASCII HEX code	4	4	4	3	4	2	4	1
ASCII character	D		C		B		A	

	DT40			
Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
Binary data	1 1 0 0	1 1 0 1	1 0 1 0	1 0 1 1
Hexadecimal	C	D	A	B

Description

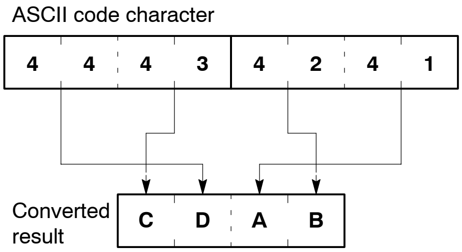
Converts ASCII code that expresses hexadecimal characters, starting from the 16-bit area specified by S1, to hexadecimal numbers as specified by S2. The converted result is stored in the area starting from the 16-bit area specified by D.

The volume of the results (hexadecimal numeric data) is half that of the converted ASCII code.

Precautions during programming

The data for two ASCII code characters is converted to two numeric digits for one word. When this takes place, the characters of the upper and lower bytes are interchanged.

Four characters are converted as one segment of data.



Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

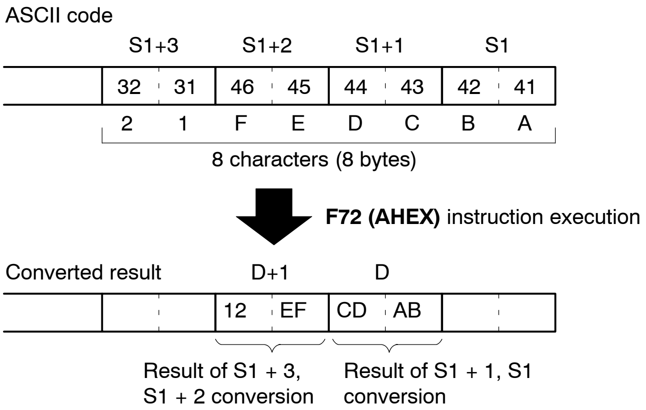
Σ Error flag (R9008): Turns on for an instant when:

- The area specified using the index modifier exceeds the limit.
- The number of bytes specified by S2 exceeds the area specified by S1.
- The converted result exceeds the area specified by D.
- The data specified by S2 is recognized as “0”.
- ASCII code, not a hexadecimal number (0 to F), is specified.

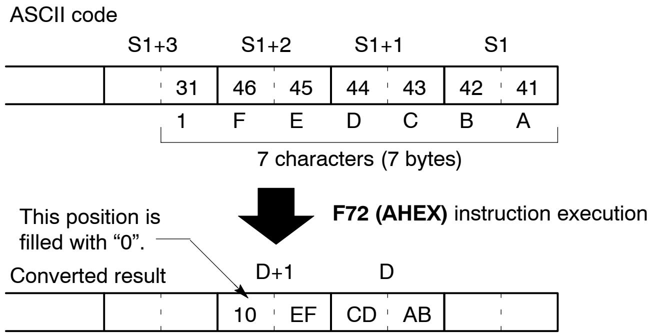
Conversion Example

The following shows conversion of ASCII codes to hexadecimal data.

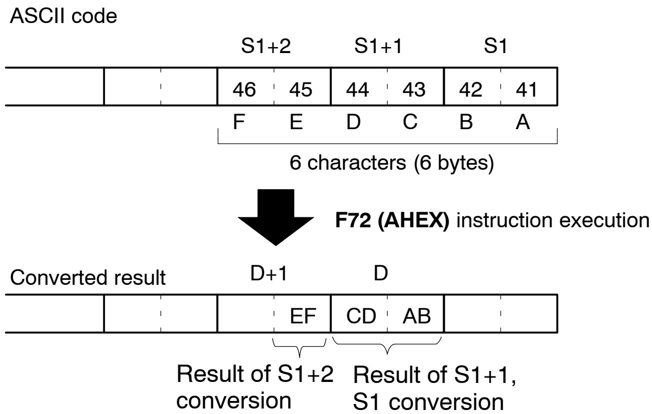
Conversion of eight characters (S2 = K8)



Conversion of 7 characters (S2 = K7)



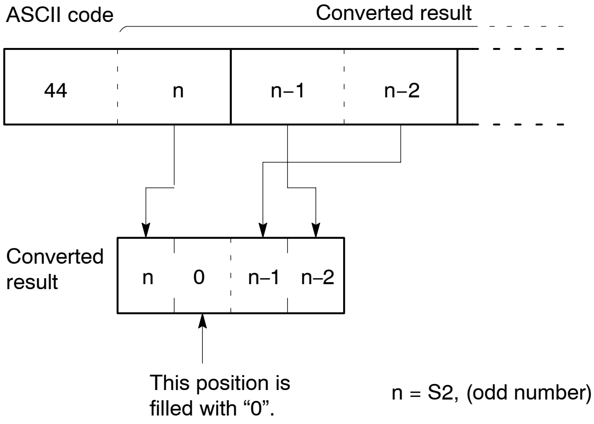
Conversion of 6 characters (S2 = K6)



☞ Note

In the conversion results, only the data for the lower byte is stored in D + 1 word. The data for the higher byte is left as it is, and does not change.

The converted results are stored in byte units.
If an odd number of characters is being converted, “0” will be entered for bit position 0 to 3 of the final data (byte) of the converted results.



ASCII HEX code/Hexadecimal characters

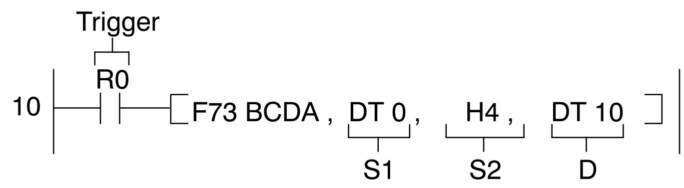
ASCII HEX code	Hexadecimal characters
H30	0
H31	1
H32	2
H33	3
H34	4
H35	5
H36	6
H37	7
H38	8
H39	9
H41	A
H42	B
H43	C
H44	D
H45	E
H46	F

F73 (BCDA) P73 (PBCDA) BCD data → ASCII code

Outline Converts BCD code to ASCII code that expresses the equivalent decimals.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P73 (PBCDA)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 73		(BCDA)
	DT		0
	H		4
	DT		10

S1	Starting 16-bit area for BCD data (source)
S2	16-bit equivalent constant or 16-bit area to specify number of source data bytes to be converted and to arrange the converted data
D	Starting 16-bit area for storing conversion result (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Converts BCD code that express a 4-digit decimal number (4-digit BCD H code) stored in data register DT0 to ASCII code when trigger R0 turns on. The converted data is stored in data registers DT10 and DT11.

When S2 = H2 (normal direction, 2 bytes conversion)

	DT0			
BCD H code	1	2	3	4

⬇

	DT11				DT10			
ASCII HEX code	3	2	3	1	3	4	3	3
ASCII character	2		1		4		3	

When S2 = H1002 (reverse direction, 2 bytes conversion)

	DT0			
BCD data	1	2	3	4

⬇

	DT11				DT10			
ASCII HEX code	3	4	3	3	3	2	3	1
ASCII character	4		3		2		1	

Description

Converts the BCD code starting from the 16-bit area specified by S1 to ASCII code that expresses the equivalent decimals as specified by S2. The converted result is stored in the area starting from the 16-bit area specified by D.

A maximum of four bytes (8−figure of data) can be converted.

S2 specifies the number of source data bytes and the direction of converted data (normal/reverse).

The data length when converted to ASCII code becomes double the BCD source data.

How to specify S2

S2 = H□ 0 0□

Number of bytes for BCD data

H1: 1 byte (BCD code that expresses a 2-digit decimal)
H2: 2 bytes (BCD code that expresses a 4-digit decimal)
H3: 3 bytes (BCD code that expresses a 6-digit decimal)
H4: 4 bytes (BCD code that expresses a 8-digit decimal)

Direction of converted data

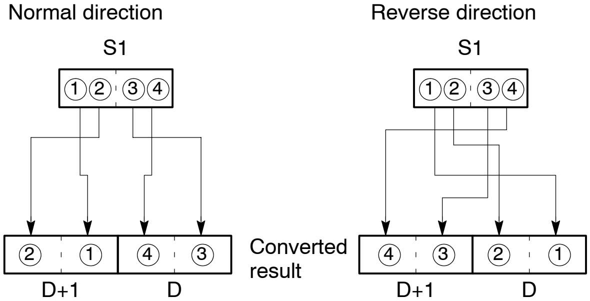
H0: Normal direction
H1: Reverse direction

Since you can specify source data in byte units, it is possible to convert only the lower byte of S1 to ASCII code.

Precautions during programming

The two characters that make up one byte are interchanged when stored.

Two bytes are converted as one segment of data.



Flag conditions

Σ Error flag (R9007):	Turns on and stays on when:
Σ Error flag (R9008):	Turns on for an instant when:

– The area specified using the index modifier exceeds the limit.
– The data specified by S1 is not BCD data.
– The number of bytes specified by S2 exceeds the area specified by S1.
– The converted result exceeds the area specified by D.
– The data specified by S2 is recognized as “0”.
– The number of bytes specified by S2 is more than H4.

Conversion Example

The following shows conversion from BCD data to ASCII codes.

Normal direction conversion of 4 bytes (S2 = H0004)

BCD data

			S1+1		S1			
BCD H code			12	34	56	78		

4 bytes

F73 (BCDA) instruction execution

Converted result	D+3		D+2		D+1		D	
ASCII HEX code	32	31	34	33	36	35	38	37
ASCII character	2	1	4	3	6	5	8	7
	Result of S1 + 1 conversion				Result of S1 conversion			

Reverse direction conversion of 4 bytes (S2 = H1004)

BCD data

			S1+1		S1			
BCD H code			12	34	56	78		

4 bytes

F73 (BCDA) instruction execution

Converted result	D+3		D+2		D+1		D	
ASCII HEX code	38	37	36	35	34	33	32	31
ASCII character	8	7	6	5	4	3	2	1
	Result of S1 conversion				Result of S1 + 1 conversion			

ASCII HEX code to express BCD character

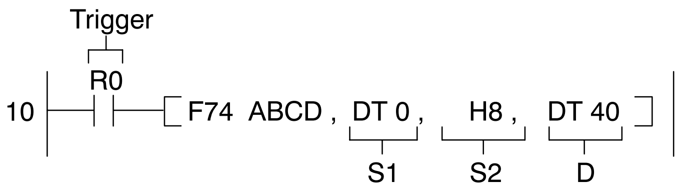
BCD character	ASCII HEX code
0	H30
1	H31
2	H32
3	H33
4	H34
5	H35
6	H36
7	H37
8	H38
9	H39

F74 (ABCD) P74 (PABCD) ASCII code → BCD data

Outline Converts ASCII code that expresses decimal characters to BCD code. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P74 (PABCD)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 74 (ABCD)
	DT 0
	H 8
	DT 40

S1	Starting 16-bit area for storing ASCII code (source)
S2	16-bit equivalent constant or 16-bit area to specify number of source data bytes to be converted and to arrange converted data
D	Starting 16-bit area for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Converts ASCII codes stored in data registers DT1 and DT0 to BCD data when trigger R0 turns on. The converted data is stored in data register DT40.

When S2 = H4 (normal direction, 4 bytes)

	DT1		DT0	
ASCII HEX code	3 4	3 3	3 2	3 1
ASCII character	4	3	2	1

	DT40			
BCD H code	3	4	1	2

When S2 = H1004 (reverse direction, 4 bytes)

	DT1		DT0	
ASCII HEX code	3 4	3 3	3 2	3 1
ASCII character	4	3	2	1

	DT40			
BCD H code	1	2	3	4

Description

Converts ASCII codes that express decimal characters, starting from the 16-bit area specified by S1, to BCD data as specified by S2. The converted result is stored in the area starting from the 16-bit area specified by D. A maximum of eight characters of data can be converted.

S2 specifies the number of source data bytes and the direction of converted data (normal/reverse).

The data length when converted to a BCD number becomes half the ASCII code source data.

How to specify S2

S2 = H □ 0 0 □

Number of bytes for ASCII character

H1: 1 byte (1 ASCII character)
H2: 2 bytes (2 ASCII characters)
H3: 3 bytes (3 ASCII characters)
H4: 4 bytes (4 ASCII characters)
H5: 5 bytes (5 ASCII characters)
H6: 6 bytes (6 ASCII characters)
H7: 7 bytes (7 ASCII characters)
H8: 8 bytes (8 ASCII characters)

Direction converted data

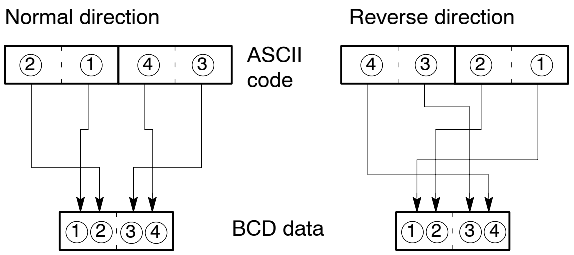
H0: Normal direction
H1: Reverse direction

Precautions during programming

The data for two ASCII code characters is converted to two numeric digits for one byte. When this takes place, the characters of the upper and lower bytes are interchanged.

Four characters are converted as one segment of data.

The converted results are stored in byte units.
If an odd number of characters is being converted, “0” will be entered for bit position 0 to 3 of the final data (byte) of the converted results if data is sequenced in the normal direction, and “0” will be entered for bit position 4 to 7 if data is being sequenced in the reverse direction.



Conversion Example

The following shows conversion from ASCII codes to BCD data.

8 ASCII characters conversion (S2=H0008)

ASCII code

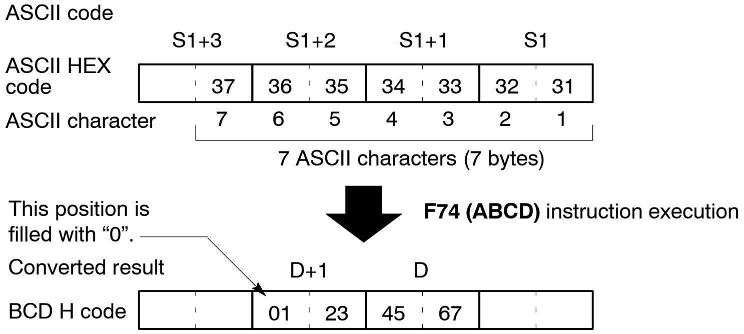
	S1+3		S1+2		S1+1		S1	
ASCII HEX code	38	37	36	35	34	33	32	31
ASCII character	8	7	6	5	4	3	2	1

8 ASCII characters (8 bytes)

F74 (ABCD) instruction execution

Converted result			D+1		D			
BCD H code			78	56	34	12		

7 ASCII characters (S2=H1007)



BCD character to express ASCII HEX code

ASCII HEX code	BCD character
H30	0
H31	1
H32	2
H33	3
H34	4
H35	5
H36	6
H37	7
H38	8
H39	9

Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

Σ Error flag (R9008): Turns on for an instant when:

− The area specified using the index modifier exceeds the limit.
− ASCII code not corresponding to decimal numbers (0 to 9) is specified.
− The number of bytes specified by S2 exceeds the area specified by S1.
− The converted result exceeds the area specified by D.
− The data specified by S2 is recognized as “0”.
− The number of bytes for ASCII characters in S2 is more than H8.

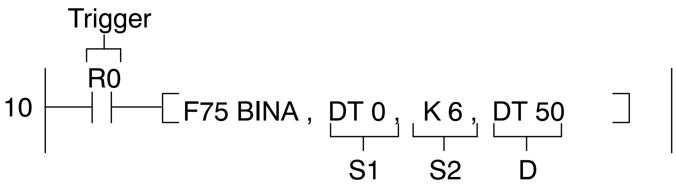
F75 (BINA) P75 (PBINA)

16-bit binary data → ASCII code

Outline Converts 16-bit data to ASCII code that expresses the equivalent decimals.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P75 (PBINA)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F 75 (BINA)
	DT 0
	K 6
	DT 50

S1	16-bit equivalent constant or 16-bit area to be converted (source)
S2	16-bit equivalent constant or 16-bit area to specify number of bytes used to express destination data (ASCII codes)
D	Starting 16-bit area for storing ASCII codes (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Converts the 16-bit data stored in data register DT0 to ASCII codes that express the equivalent decimals when trigger R0 turns on. The converted data is stored in data registers DT52 to DT50.

Source

	DT0			
Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
Decimal data	K−100			

Destination

	DT52				DT51				DT50			
ASCII HEX code	3	0	3	0	3	1	2	D	2	0	2	0
ASCII character	0		0		1		–					

6 bytes

SPACE

Description

Converts the 16-bit data specified by S1 to ASCII codes that express the equivalent decimals. The converted result is stored in the area starting from the 16-bit area specified by D as specified by S2.

Specify the number of bytes in decimal number in the S2. (This specification cannot be made with BCD data.)

Precautions during programming

If a positive number is converted, the "+" sign is not converted.

When a negative number is converted, the "–" sign is also converted to ASCII code (ASCII HEX code: H2D).

If the area specified by S2 is more than that required by the converted data, the ASCII code for "SPACE" (ASCII HEX code: H20) is stored in the extra area.

Data is stored in the direction towards the final address, so the position of the ASCII code may change, depending on the size of the data storage area.

When S2=K8 (8 bytes)

D+3		D+2		D+1		D			
30	30	31	2D	20	20	20	20		
0	0	1	–	(Space)	(Space)	(Space)	(Space)		

ASCII code — Extra bytes

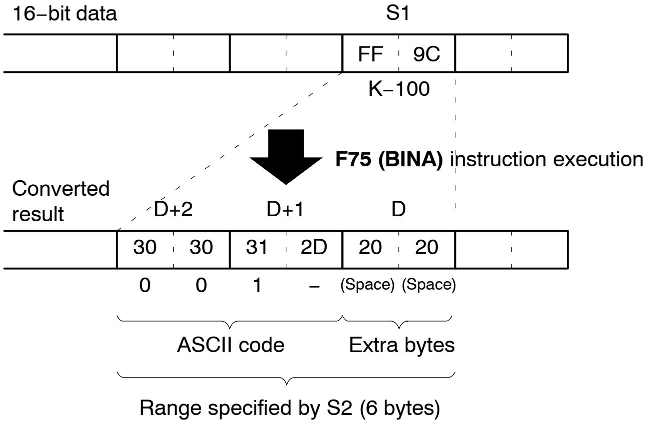
Range specified by S2

If the number of bytes of ASCII codes following conversion (including the minus sign) is larger than the number of bytes specified by the S2, an operation error occurs. Make sure the sign is taken into consideration when specifying the object of conversion for the S2.

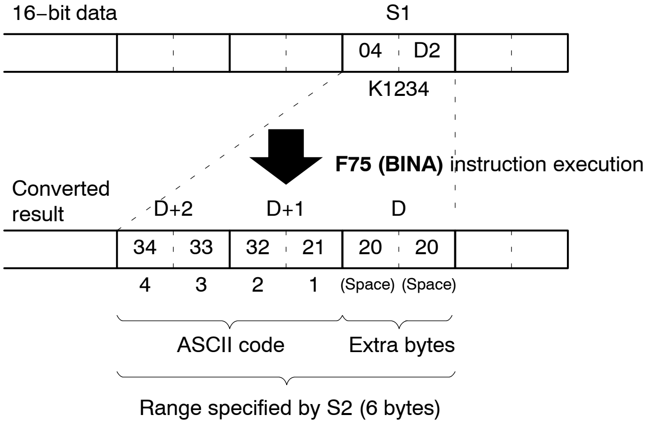
Conversion Example

The following shows conversion from 16−bit decimal data to ASCII codes.

When a negative number is converted



When a positive number is converted



Decimal characters to express ASCII HEX code

Decimal characters	ASCII HEX code
SPACE	H20
–	H2D
0	H30
1	H31
2	H32
3	H33
4	H34
5	H35
6	H36
7	H37
8	H38
9	H39

Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

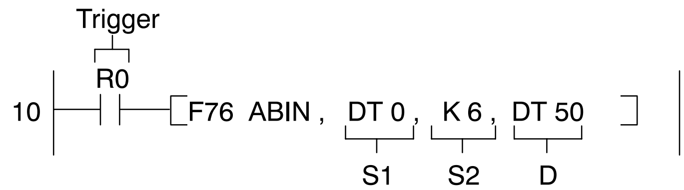
Σ Error flag (R9008): Turns on for an instant when:
- The area specified using the index modifier exceeds the limit.
- The number of bytes specified by S2 exceeds the area specified by D.
- The data specified by S2 is recognized as “0”.
- The converted result exceeds the area specified by D.
- The number of bytes of converted result exceeds the number of bytes specified by S2.

F76 (ABIN) P76 (PABIN) ASCII code → 16-bit binary data

Outline Converts ASCII code that expresses decimal digits to 16-bit data that expresses the equivalent number.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P76 (PABIN)” is not available.

Program example

Ladder Diagram



Boolean	
Address	Instruction
10	ST R 0
11	F 76 (ABIN)
	DT 0
	K 6
	DT 50

S1	Starting 16-bit area for ASCII code (source)
S2	16-bit equivalent constant or 16-bit area to specify number of source data bytes to be converted
D	16-bit area for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Converts ASCII codes (6 bytes) that express decimal digits in data registers DT2 to DT0 to 16-bit data when trigger R0 turns on. The converted data is stored in data register DT50.

Source

	DT2				DT1				DT0			
ASCII HEX code	3	0	3	0	3	1	2	D	3	0	3	0
ASCII character	0		0		1		—					

6 bytes

In this case, this position should be set to “SPACE” or “0”.

Destination

	DT50			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Decimal data	K−100			

Description

Converts the ASCII codes that express the decimal digits, starting from the 16-bit area specified by S1 to 16-bit data as specified by S2. The converted result is stored in the area specified by D.

S2 specifies the number of source data bytes to be converted using decimal number. (This specification cannot be made with BCD data.)

Precautions during programming

The ASCII codes being converted should be stored in the direction of the last address in the specified area.

If the area specified by S1 and S2 is more than that required for the data you want to convert, place “0” (ASCII HEX code: H30) or “SPACE” (ASCII HEX code: H20) into the extra bytes.

ASCII codes with signs (such as +: H2B and −: H2D) are also converted. The + codes can be omitted.

Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

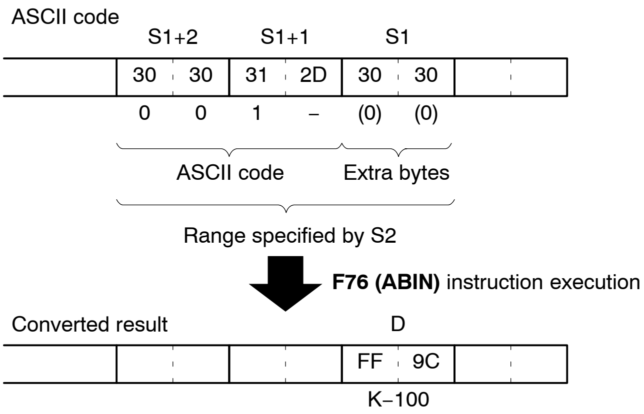
Σ Error flag (R9008): Turns on for an instant when:

- The area specified using the index modifier exceeds the limit.
- The number of bytes specified by S2 exceeds the area specified by S1.
- The data specified by S2 is recognized as “0”.
- The converted result exceeds the 16-bit area specified by D.
- The converted result exceeds the 16-bit data.
- ASCII code not corresponding to decimal numbers (0 to 9) or ASCII characters (+, −, and SPACE) is specified.

Conversion Example

The following shows conversion from ASCII codes to decimal data in a 16−bit configuration.

Example of converting an ASCII code indicating a negative number



Example of converting an ASCII code indicating a positive number

Example 1:

ASCII code

	S1+2		S1+1		S1			
	30	30	31	2B	20	20		
	0	0	1	+	(Space)	(Space)		

$\underbrace{\text{30 30 31 2B}}_{\text{ASCII code}}\ \underbrace{\text{20 20}}_{\text{Extra bytes}}$

Range specified by S2

Example 2:

ASCII code

	S1+2		S1+1		S1			
	30	30	31	20	20	20		
	0	0	1	(Space)	(Space)	(Space)		

$\underbrace{\text{30 30 31}}_{\text{ASCII code}}\ \underbrace{\text{20 20 20}}_{\text{Extra bytes}}$

Range specified by S2

F76 (ABIN) instruction execution

Converted result of example 1 or 2

					D			
					00	64		
					K100			

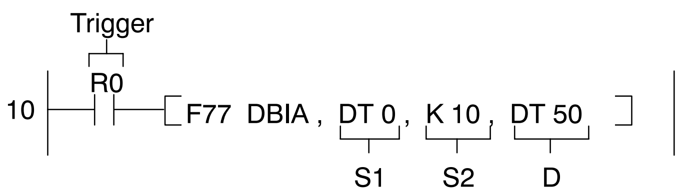
ASCII HEX code to express decimal characters

ASCII HEX code	Decimal characters
H20	SPACE
H2B	+
H2D	−
H30	0
H31	1
H32	2
H33	3
H34	4
H35	5
H36	6
H37	7
H38	8
H39	9

F77 (DBIA) / P77 (PDBIA) 32-bit binary data → ASCII code

Outline Converts 32-bit data to ASCII code that expresses the equivalent decimals.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P77 (PDBIA)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F 77 (DBIA)
		DT 0
		K 10
		DT 50

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data to be converted (source)
S2	16-bit equivalent constant or 16-bit area to specify number of bytes used to express destination data (ASCII codes)
D	Starting 16-bit area for storing ASCII codes (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Converts the 32-bit data stored in data registers DT1 and DT0 to ASCII code that expresses the equivalent decimals when trigger R0 turns on. The converted data is stored in data registers DT54 to DT50 (10 bytes).

Source

	DT1				DT0			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Decimal	K12345678							

Destination

	DT54				DT53				DT52				DT51				DT50			
ASCII HEX code	3	8	3	7	3	6	3	5	3	4	3	3	3	2	3	1	2	0	2	0
ASCII character	8		7		6		5		4		3		2		1					

$\underbrace{\hspace{10em}}_{\text{Number of bytes specified by S2 (10 bytes)}}$

SPACE

Description

Converts the 32-bit data specified by S1 to ASCII code that expresses the equivalent decimals. The converted result is stored in the area starting from the 16-bit area specified by D as specified by S2.

S2 specifies the number of bytes used to express the destination data using decimal.

Precautions during programming

When a positive number is converted, the “+” sign is not converted.

When a negative number is converted, the “−” sign is also converted to ASCII code (ASCII HEX code: H2D).

If the area specified by S2 is more than that required by the converted data, the ASCII code for “SPACE” (ASCII HEX code: H20) is stored in the extra area.

Data is stored in the direction of the last address, so the position of the ASCII code may change depending on the size of the data storage area.

If the number of bytes of ASCII codes following conversion (including the minus sign) is larger than the number of bytes specified by the S2, an operation error occurs. Make sure the sign is taken into consideration when specifying the object of conversion for the S2.

Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

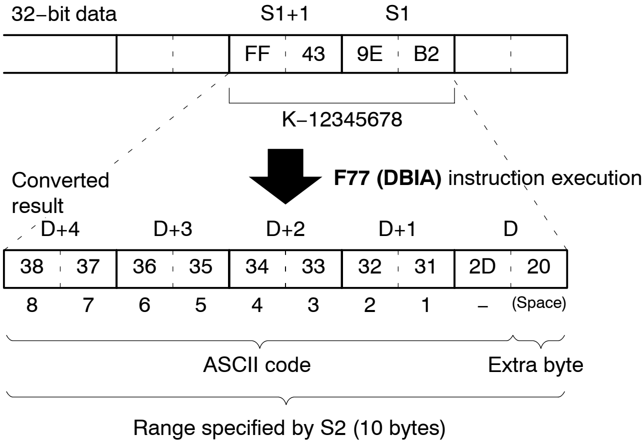
Σ Error flag (R9008): Turns on for an instant when:

- The area specified using the index modifier exceeds the limit.
- The number of bytes specified by S2 exceeds the area specified by D.
- The data specified by S2 is recognized as “0”.
- The converted result exceeds the area specified by D.
- The number of bytes of converted result exceeds the number of bytes specified by S2.
- For FP2SH and FP10SH, the error flag (R9007) turns on only when these operation errors occurs.

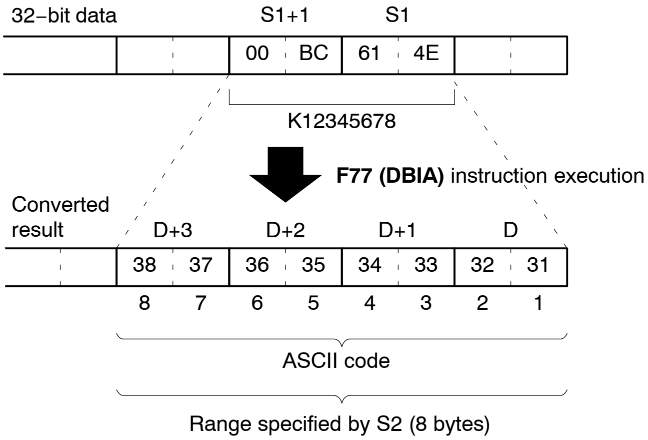
Conversion Example

The following shows conversion from 32–bit decimal format data to ASCII codes.

Example of converting a negative number



Example of converting a positive number



Decimal characters to express ASCII HEX code

Decimal characters	ASCII HEX code
SPACE	H20
+	H2B
–	H2D
0	H30
1	H31
2	H32
3	H33
4	H34
5	H35
6	H36
7	H37
8	H38
9	H39

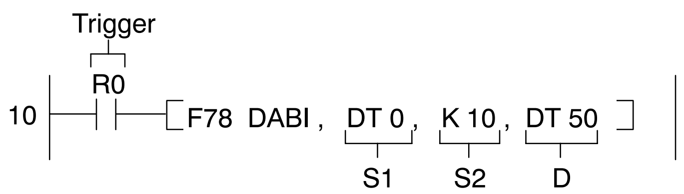
F78 (DABI) P78 (PDABI)

ASCII code → 32-bit binary data

Outline Converts ASCII code that expresses decimal digits to 32-bit data that expresses the equivalent number.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P78 (PDABI)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 78		(DABI)
	DT		0
	K		10
	DT		50

S1	Starting 16-bit area for ASCII code (source)
S2	16-bit equivalent constant or 16-bit area to specify number of source data bytes to be converted
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Converts ASCII codes (10 bytes) that express decimal digits in data registers DT4, DT3, DT2, DT1 and DT0 to 32-bit data when trigger R0 turns on. The converted data is stored in data registers DT51 and DT50.

Source

	DT4				DT3				DT2				DT1				DT0			
ASCII HEX code	3	8	3	7	3	6	3	5	3	4	3	3	3	2	3	1	2	0	2	0
ASCII character	8		7		6		5		4		3		2		1					

ASCII codes (10 bytes) specified by S2

SPACE

⬇

Destination

	DT51				DT50			
Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0	15 . : 12	11 . . 8	7 . . 4	3 . . 0
Decimal	K 1 2 3 4 5 6 7 8							

Description

Converts ASCII code that expresses the decimal digits, starting from the 16-bit area specified by S1 to 32-bit data as specified by S2. The converted result is stored in the area starting from the 16-bit area specified by D.

S2 specifies the number of bytes used to express the destination data using decimals.

Precautions during programming

The ASCII codes being converted should be stored in the direction of the last address in the specified area.

If the area specified by S1 and S2 is more than that required by the data you want to convert, place “0” (ASCII HEX code: H30) or “SPACE” (ASCII HEX code: H20) in the extra bytes.

ASCII codes with signs (such as +: H2B and −: H2D) are also converted. The + codes can be omitted.

Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

Σ Error flag (R9008): Turns on for an instant when:

- The area specified using the index modifier exceeds the limit.
- The number of bytes specified by S2 exceeds the area specified by S1.
- The data specified by S2 is recognized as “0”.
- The converted result exceeds the area specified by D.
- The converted result exceeds the 32-bit data.
- ASCII code not corresponding to decimal numbers (0 to 9) or ASCII characters (+, −, and SPACE) is specified.

Conversion Example

The following shows conversion from ASCII codes to decimal data in a 32−bit configuration.

Example of converting an ASCII code indicating a negative number

ASCII code

S1+4		S1+3		S1+2		S1+1		S1	
38	37	36	35	34	33	32	31	2D	20
8	7	6	5	4	3	2	1	–	(Space)

ASCII code — Extra byte

Range specified by S2 (10 bytes)

F78 (DABI) instruction execution

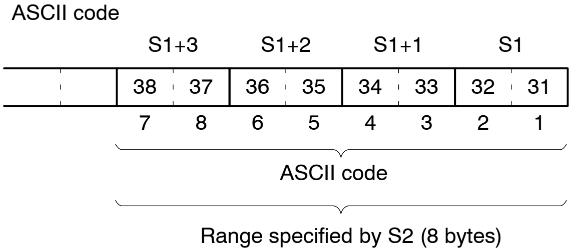
Converted result

			D+1		D			
			FF	43	9E	B2		

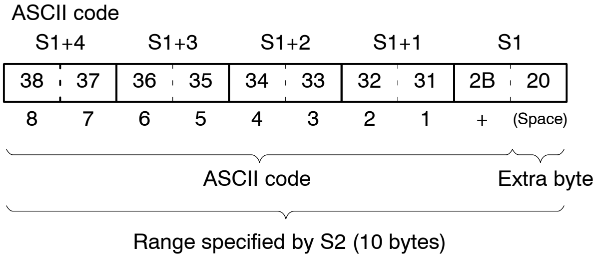
K−12345678

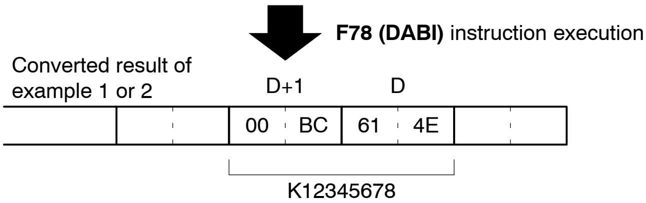
Example of converting an ASCII code indicating a positive number

Example 1:



Example 2:





ASCII HEX code to express decimal characters

ASCII HEX code	Decimal characters
H20	SPACE
H2B	+
H2D	−
H30	0
H31	1
H32	2
H33	3
H34	4
H35	5
H36	6
H37	7
H38	8
H39	9

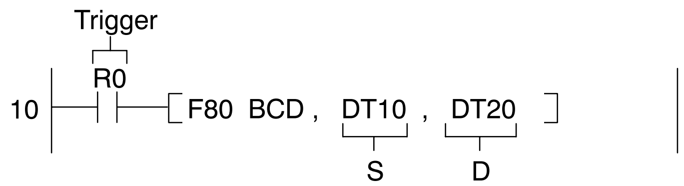
F80 (BCD) P80 (PBCD)

16-bit binary data → 4-digit BCD data

Outline Converts 16-bit binary data to BCD code the expresses a 4-digit decimal.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P80 (PBCD)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F 80		(BCD)
	DT		10
	DT		20

S	16-bit equivalent constant or 16-bit area for storing 16−bit binary data (source)
D	16-bit area for 4-digit BCD data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Converts the contents of data register DT10 to BCD code that expresses a 4-digit decimal when trigger R0 turns on. The converted data is stored in data register DT20.

If DT10 is 16 using decimal number conversion, the following will be stored in DT20.

Source [S]: K16

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0
Decimal	K16			

Conversion (to BCD code)

Destination [D]: H16 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	0 0 0 0	0 0 0 0	0 0 0 1	0 1 1 0
BCD H code	0	0	1	6

Description

Converts the 16-bit binary data specified by S to BCD code that expresses a 4-digit decimal. The converted data is stored in D.

Precautions during programming

The maximum value of 16−bit binary data that can be converted to BCD code is K9999 (H270F).

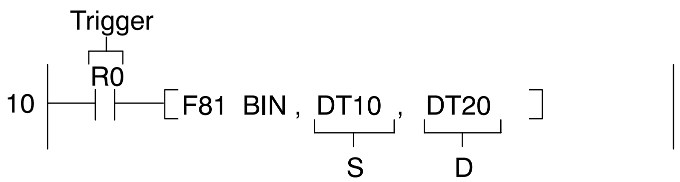
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The range that binary data can be BCD converted is exceeded. (When minus or when K9999 is exceeded)

F81 (BIN) P81 (PBIN) 4-digit BCD data → 16-bit binary data

Outline Converts BCD code that expresses a 4-digit decimal to 16-bit binary data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P81 (PBIN)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F 81 (BIN)
		DT 10
		DT 20

S	4-digit BCD equivalent constant or 16-bit area for 4-digit BCD data (source)
D	16-bit area for storing 16-bit binary data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

Converts the contents of data register DT10 to 16-bit binary data when trigger R0 turns on. The converted data is stored in data register DT20.

If DT10 is BCD data consisting of H15, the following will be stored in DT20.

Source [S]: H15 (BCD)

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 0	0 0 0 1	0 1 0 1
BCD H code	0	0	1	5

Conversion (to binary data)

Destination [D]: K15

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1
Decimal	K15			

Description

Converts BCD code that expresses a 4-digit decimal specified by S to 16-bit binary data. The converted data is stored in D.

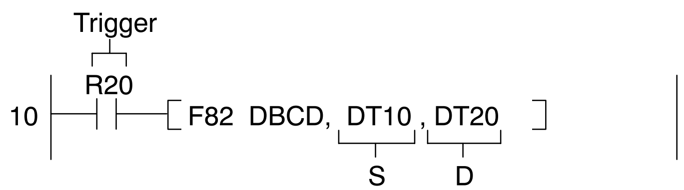
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data specified by S is not BCD data.

F82 (DBCD) P82 (PDBCD) 32-bit binary data → 8-digit BCD data

Outline Converts 32-bit binary data to BCD code that expresses an 8-digit decimal.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P82 (PDBCD)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 20 F 82 (DBCD) DT 10 DT 20

S	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area for 8-digit BCD code (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Converts the contents of data registers DT11 and DT10 to BCD code that expresses an 8-digit decimal when trigger R20 turns on. The converted data is stored in data registers DT21 and DT20.

Description

Converts the 32-bit data specified by S to BCD code that expresses an 8-digit decimal. The converted data is stored in D+1 and D.

Precaution during programming

The maximum value of binary data that can be converted to BCD code is K99999999 (H5F5E0FF).

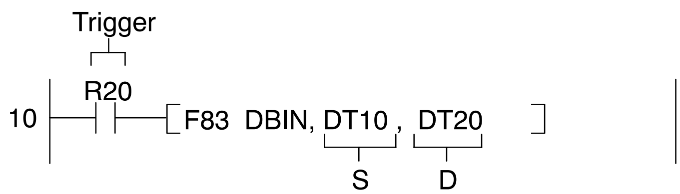
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - When the range that binary data can be BCD converted is exceeded. (When minus or when K99999999 is exceeded)

F83 (DBIN) P83 (PDBIN) 8-digit BCD data → 32-bit binary data

Outline Converts BCD code that expresses an 8-digit decimal to 32-bit binary data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P83 (PDBIN)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 20
	11	F 83 (DBIN)
		DT 10
		DT 20

S	8-digit BCD equivalent constant or lower 16-bit area for 8-digit BCD code (source)
D	Lower 16-bit area for 32-bit data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	K	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available
K: K0 to K9

Explanation of example

Converts BCD code that expresses an 8−digit decimal of data registers DT11 and DT10 to 32-bit binary data when trigger R20 turns on. The converted data is stored in data registers DT21 and DT20.

Description

Converts BCD code that expresses an 8-digit decimal specified by S to 32-bit binary data. The converted data is stored in D+1 and D.

Flag conditions

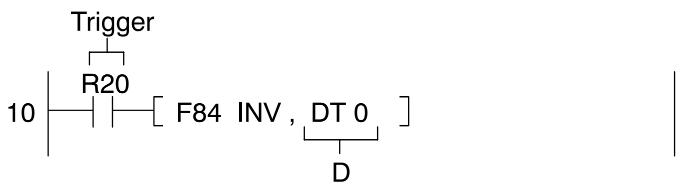
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The area specified using the index modifier exceeds the limit.
  − The data specified by S is not BCD data.

F84 (INV) P84 (PINV)

16-bit data invert

Outline Inverts all bits in the 16-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P84 (PINV)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10 11	ST R 20 F 84 (INV) DT 0

D	16-bit area to be inverted

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Inverts the contents of data register DT0 when trigger R20 turns on.

Destination

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	0 1 0 1	1 1 1 0	1 0 1 1	1 1 0 1

R20: on (inversion)

Destination

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	1 0 1 0	0 0 0 1	0 1 0 0	0 0 1 0

Description

Inverts each bit (0 or 1) of the 16-bit data specified by D.

This instruction is useful for controlling an external device (7−segment display) that uses negative logic operation.

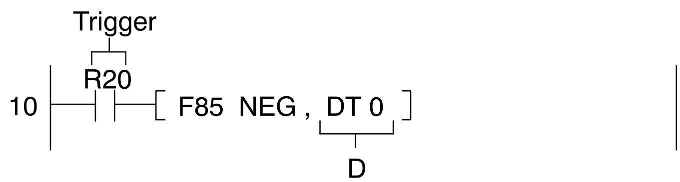
Flag conditions

- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.

F85 (NEG) P85 (PNEG) 16-bit data complement of 2

Outline Takes complement of 2 in 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P85 (PNEG)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 20 F 85 (NEG) DT 0

D	16-bit area for storing original data and its two's complement

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Takes two's complement of data register DT0 when trigger R20 turns on.

Destination

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
Decimal data	K3			

R20: on

Destination

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT0	1 1 1 1	1 1 1 1	1 1 1 1	1 1 0 1
Decimal data	K−3			

Description

Takes two’s complement of 16-bit data specified by D.

The two’s complement is obtained by inverting all bits and adding 1 to the inverted result.

This instruction is useful for changing the sign of 16-bit data from positive to negative or from negative to positive.

Flag conditions

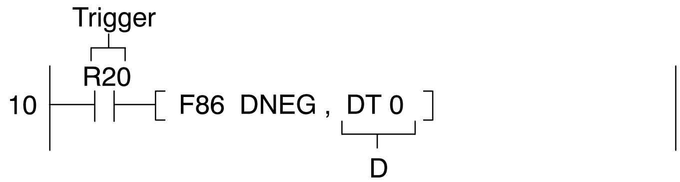
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F86 (DNEG)
P86 (PDNEG)

32-bit data complement of 2

Outline Takes complement of 2 in 32-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P86 (PDNEG)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 20 F 86 (DNEG) DT 0

D	Lower 16-bit area of 32-bit data for storing original data and its two's complement

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Takes two's complement of data registers DT1 and DT0 when trigger R20 turns on.

Destination	DT1				DT0			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 0 1
Decimal data	K−3							

Higher 16-bit area Lower 16-bit area

R20: on

Destination	DT1				DT0			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
Decimal data	K3							

Higher 16-bit area Lower 16-bit area

Description

Takes two's complement of 32-bit data specified by D.
The two's complement is obtained by inverting all bits and adding 1 to the inverted result.

This instruction is useful for changing the sign of 32-bit data from positive to negative or from negative to positive.

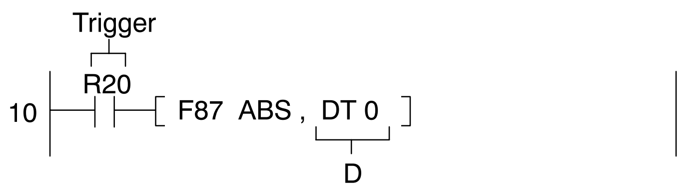
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F87 (ABS) P87 (PABS) 16-bit data absolute value

Outline Takes absolute value of signed 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P87 (PABS)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 20 F 87 (ABS) DT 0

D	16-bit area for storing original data and its absolute value

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Takes absolute value of data register DT0 when trigger R20 turns on.

For instance, regardless of whether the value of DT0 is K1 or K−1, it will be K1 when the instruction is executed.

Description

Takes absolute value of signed 16-bit data specified by D. The absolute value of the signed 16-bit data is stored in D.

This is effective for processing data in which the polarity (+ or −) changes.

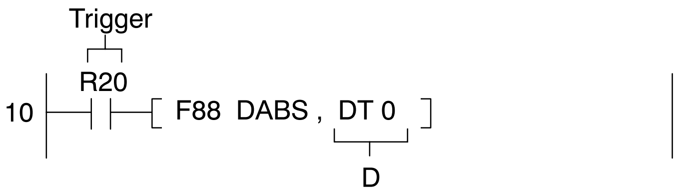
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  − The area specified using the index modifier exceeds the limit.
  − The 16-bit data is the negative minimum value "K-32768 (H8000)."
- Carry flag (R9009): Turns on for an instant when 16-bit data is the negative value "range: K-1 to K-32767 (HFFFF to H8001)."

F88 (DABS) P88 (PDABS) 32-bit data absolute value

Outline Takes absolute value of signed 32-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P88 (PDABS)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 20 F 88 (DABS) DT 0

D	Lower 16-bit area of 32-bit data for storing original data and its absolute value

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Takes absolute value of data registers DT1 and DT0 when trigger R20 turns on. The absolute value of DT1 and DT0 is stored in data registers DT1 and DT0.

Description

Takes the absolute value of signed 32-bit data specified by D. The absolute value of the 32-bit data is stored in D+1 and D.

This is effective for processing data in which the polarity (+ or −) changes.

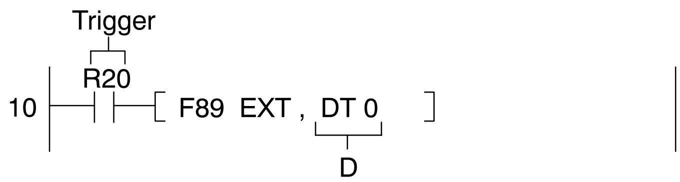
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The 32-bit data is the negative minimum value “K-2147483648 (H80000000).”
- Carry flag (R9009): Turns on for an instant when 32-bit data is negative value “range K-1 to K-2147483647 (HFFFFFFFF to H80000001).”

F89 (EXT) P89 (PEXT) 16-bit data sign extension

Outline Copies the sign bit of the specified 16-bit data to all the bits of the higher 16-bit area (extended 16-bit area).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P89 (PEXT)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 20 F 89 (EXT) DT 0

D	16-bit area for storing original 16-bit binary data

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Copies the sign bit of data register DT0 to all the bits of data register DT1 when trigger R20 turns on.

If K−2 is stored in DT0, the data will be as follows.

Sign bit (0: positive, 1: negative)

	DT0			
Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
Binary data	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0
Decimal data	K−2			

16-bit data

R20: on

	DT1				DT0			
Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0	15 . . 12	11 . . 8	7 . . 4	3 . . 0
Binary data	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0
Decimal data	K−2							

Higher 16-bit area (extended 16-bit area) | Lower 16-bit area

32-bit data

Description

16−bit data is converted to 32−bit data, without signs and values being changed.

If the sign bit (bit position 15) of the 16−bit data specified by D is 0, all 16 bits of the next area of D will be set to 0. If the sign bit is 1, all 16 bits will be set to 1.
By doing this, the 16−bit data is converted to 32−bit data, without the sign or the values changing.

Double word data with D as the first data can be used as the operand of 32−bit operation instructions following execution of the **F89 (EXT)** instruction.

Flag conditions

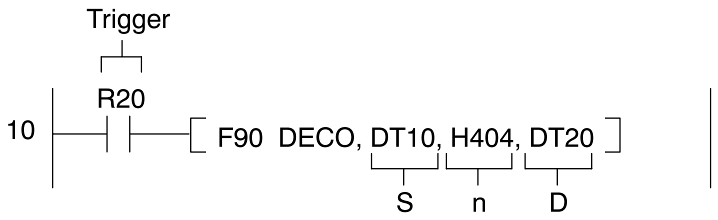
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F90 (DECO) P90 (PDECO) Decode

Outline Decodes the specified data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P90 (PDECO)" is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	20
11	F 90		(DECO)
	DT		10
	H		404
	DT		20

S	16-bit equivalent constant or 16-bit area to be decoded (source)
n	16-bit area equivalent constant or 16-bit area to specify starting bit position and number of bits to be decoded
D	Starting 16-bit area for storing decoded data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Decodes data register DT10 according to the n = H404 when trigger R20 turns on. The decoded result is stored in data register DT20.

Example: When n: H404

Source

Bit position 4

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 1 0 0	0 1 1 0	0 1 1 1	1 0 0 1

4 bits

4 bits from bit position 4 to 7 of DT10 are decoded.
[Binary: 0111 (decimal: K7)]

R20: on

Destination

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0

The decoded data "1 (on)" is stored in bit position 7 of DT20. The bits except bit position 7 become 0.

Description

Decodes the contents of 16-bit data specified by S according to the contents of n. The decoded result is stored in the area starting from 16-bit area specified by D.

The length of the area required to store decoding results changes depending on the length of the data being decoded.

How to specify control data “n”

n specifies the starting bit position and the number of bits to be decoded using hexadecimal data.

16-bit data

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
n	— — — —	0 0 0 0	— — — —	0 0 0 0

The bits of “–” mark are invalid.

Starting bit position to be decoded
(set range: H0 to HF)

Set value	Starting bit position
H0	0
H1	1
H2	2
H3	3
H4	4
H5	5
H6	6
H7	7
H8	8
H9	9
HA	10
HB	11
HC	12
HD	13
HE	14
HF	15

Number of bits to be decoded
(set range: H0 to H8)

Set value	Number of bits
H0	0
H1	1
H2	2
H3	3
H4	4
H5	5
H6	6
H7	7
H8	8

Relationship between number of bits and occupied data area for decoded result

Number of bits to be decoded	**Data area required for the result**	**Valid bits in the area for the result**
1	1-word	2-bit*
2	1-word	4-bit*
3	1-word	8-bit*
4	1-word	16-bit
5	2-word	32-bit
6	4-word	64-bit
7	8-word	128-bit
8	16-word	256-bit

* Invalid bits in the data area required for the result are set to “0”.

Decoded example

When decoding 4-bit data, 16-bit data for the decoded result is shown below.

Decoding conditions (n)

Starting bit position: H0 (bit position 0)

Number of bits to be decoded: H4 (4 bits)

Data to be decoded [Binary (decimal)]	**Decoded result**															
	15	**·**	**·**	**12**	**11**	**·**	**·**	**8**	**7**	**·**	**·**	**4**	**3**	**·**	**·**	**0**
0000 (K0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
0001 (K1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0
0010 (K2)	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0
0011 (K3)	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0
0100 (K4)	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0
0101 (K5)	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0
0110 (K6)	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0
0111 (K7)	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0
1000 (K8)	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
1001 (K9)	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
1010 (K10)	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0
1011 (K11)	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0
1100 (K12)	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0
1101 (K13)	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
1110 (K14)	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1111 (K15)	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

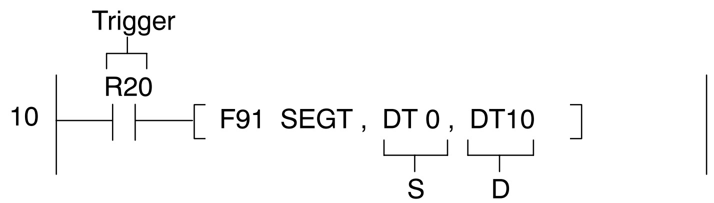
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The number of bits to be decoded is outside the range of 1 to 8.
  - The sum of the number of bits to be decoded and the starting bit position to be decoded is outside the range of 1 to 16.
  - The last data area for the decoded result exceeds the limit.

F91 (SEGT) P91 (PSEGT) 7-segment decode

Outline Converts 16-bit data to 4-digit data for 7-segment indication.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P91 (PSEGT)" is not available.

Program example

Ladder Diagram	Boolean: Address	Boolean: Instruction
	10 11	ST R 20 F 91 (SEGT) DT 0 DT 10

S	16-bit equivalent constant or 16-bit area to be converted to the 7-segment indication (source)
D	Starting 16-bit area for storing 4-digit data for 7-segment indication (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Converts the contents of data register DT0 to 4-digit data for 7-segment indication when trigger R20 turns on.
The converted data is stored in word internal relays DT11 and DT10.
For example, to display "ABCD", the following would be entered.

Source

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1
Hexadecimal	A	B	C	D

Destination

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT11	0 1 1 1	0 1 1 1	0 1 1 1	1 1 0 0
7-segment indication	A		B	

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 1 1	1 0 0 1	0 1 0 1	1 1 1 0
7-segment indication	C		D	

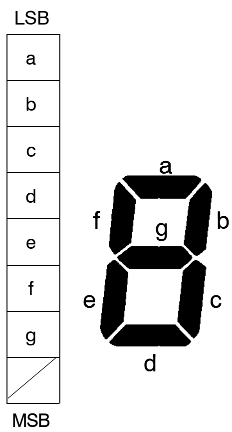
Description

Converts the 16-bit equivalent constant or 16-bit data specified by S to 4-digit data for 7-segment indication. The converted data is stored in the area starting from the 16-bit area specified by D.

The relationship between the displayed contents and the contents specified for S, and the data of the 7−segment display is shown below.

One digit data to be converted		8-bit data for 7-segment indication								7-segment indication
Hexadecimal	Binary		g	f	e	d	c	b	a	
H0	0 0 0 0	0	0	1	1	1	1	1	1	0
H1	0 0 0 1	0	0	0	0	0	1	1	0	1
H2	0 0 1 0	0	1	0	1	1	0	1	1	2
H3	0 0 1 1	0	1	0	0	1	1	1	1	3
H4	0 1 0 0	0	1	1	0	0	1	1	0	4
H5	0 1 0 1	0	1	1	0	1	1	0	1	5
H6	0 1 1 0	0	1	1	1	1	1	0	1	6
H7	0 1 1 1	0	0	1	0	0	1	1	1	7
H8	1 0 0 0	0	1	1	1	1	1	1	1	8
H9	1 0 0 1	0	1	1	0	1	1	1	1	9
HA	1 0 1 0	0	1	1	1	0	1	1	1	A
HB	1 0 1 1	0	1	1	1	1	1	0	0	b
HC	1 1 0 0	0	0	1	1	1	0	0	1	C
HD	1 1 0 1	0	1	0	1	1	1	1	0	d
HE	1 1 1 0	0	1	1	1	1	0	0	1	E
HF	1 1 1 1	0	1	1	1	0	0	0	1	F

Organization of 7-segment indication



Flag conditions

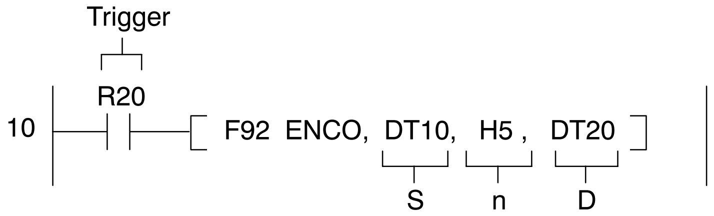
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The last data area for the converted result exceeds the limit.

F92 (ENCO) P92 (PENCO) Encode

Outline Encodes the specified data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P92 (PENCO)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 20
11	F 92 (ENCO)
	DT 10
	H 5
	DT 20

S	Starting 16-bit area to be encoded (source)
n	16-bit equivalent constant or 16-bit area to specify starting bit position and number of bits to be encoded
D	16-bit area for storing encoded data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Encodes contents of data register DT11 and DT10 according to the n: H5 when trigger R20 turns on. The encoded result is stored in 8 bits of data register DT20 starting from bit position 0.

When n: H0005

Number of bits to be encoded: 2^5 = 32 bits

Starting bit position to be encoded for destination data: bit position 0

The 8th bit of 32-bit data is in the on state.

Source

32 bits specified by H5

[S]					
	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
	DT11	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT10	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 0

Destination

Starting bit position

[D]					
	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
	DT20	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
	Decimal	K8			

The encoded result K8 (decimal) is stored in DT20.

Bit position 8 to 15 are filled with 0.

Encoded result:K8

Description

Encodes the contents of data specified by S according to the contents of n. The encoded result is stored in the 16-bit area specified by D starting from the specified bit position.

If more than one bit is on in a segment being decoded, the uppermost bit is effective.

The contents of the 2nL segment at the beginning of the area specified by the S are encoded. The encoded results are stored as decimal data, in the eight bits starting from the bit specified as the nH bit.

Invalid bits in the specified area for the result are set to 0.

How to specify control data “n”

n specifies the starting bit position of destination data and the number of bits to be decoded using hexadecimal data.

16-bit data

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
n	— — — —	0 0 0 0	— — — —	0 0 0 0

The bits of “–” mark are invalid.

Number of bits to be encoded
(set range: H1 to H8)

Set value	Number of bits
H1	2
H2	4
H3	8 (1 byte)
H4	16 (1 word)
H5	32 (2 words)
H6	64 (4 words)
H7	128 (8 words)
H8	256 (16 words)

Starting bit position of destination data to be encoded
(set range: H0 to HF)

Set value	Starting bit position
H0	0
H1	1
H2	2
H3	3
H4	4
H5	5
H6	6
H7	7
H8	8
H9	9
HA	10
HB	11
HC	12
HD	13
HE	14
HF	15

Encoded example

When encoding 16-bit data (nL=4), the encoded results are shown below.

Data to be encoded				Encoded result [Binary (decimal)]
15 · · 12	11 · · 8	7 · · 4	3 · · 0	
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 **1**	0 0 0 0 (K0)
0 0 0 0	0 0 0 0	0 0 0 0	0 0 **1** 0	0 0 0 1 (K1)
0 0 0 0	0 0 0 0	0 0 0 0	0 **1** 0 0	0 0 1 0 (K2)
0 0 0 0	0 0 0 0	0 0 0 0	**1** 0 0 0	0 0 1 1 (K3)
0 0 0 0	0 0 0 0	0 0 0 **1**	0 0 0 0	0 1 0 0 (K4)
0 0 0 0	0 0 0 0	0 0 **1** 0	0 0 0 0	0 1 0 1 (K5)
0 0 0 0	0 0 0 0	0 **1** 0 0	0 0 0 0	0 1 1 0 (K6)
0 0 0 0	0 0 0 0	**1** 0 0 0	0 0 0 0	0 1 1 1 (K7)
0 0 0 0	0 0 0 **1**	0 0 0 0	0 0 0 0	1 0 0 0 (K8)
0 0 0 0	0 0 **1** 0	0 0 0 0	0 0 0 0	1 0 0 1 (K9)
0 0 0 0	0 **1** 0 0	0 0 0 0	0 0 0 0	1 0 1 0 (K10)
0 0 0 0	**1** 0 0 0	0 0 0 0	0 0 0 0	1 0 1 1 (K11)
0 0 0 **1**	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0 (K12)
0 0 **1** 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 1 (K13)
0 **1** 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0 (K14)
1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1 (K15)

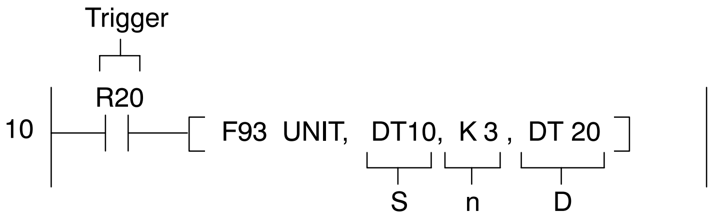
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The number of bits to be encoded is outside the range of 1 to 8.
  - The sum of the number of bits to be encoded and the starting bit position to be encoded is outside the range of 1 to 16.
  - The data to be encoded is 0.

F93 (UNIT) P93 (PUNIT) 16-bit data combine

Outline Extracts the lower 4 bits (bit positions 0 to 3) of the specified 16-bit areas and combines them into one word.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P93 (PUNIT)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 20 F 93 (UNIT) DT 10 K 3 DT 20

S	Starting 16-bit area to be extracted (source)
n	16-bit equivalent constant or 16-bit area to specify number of data to be extracted
D	16-bit area for storing combined data (destination)

Operands

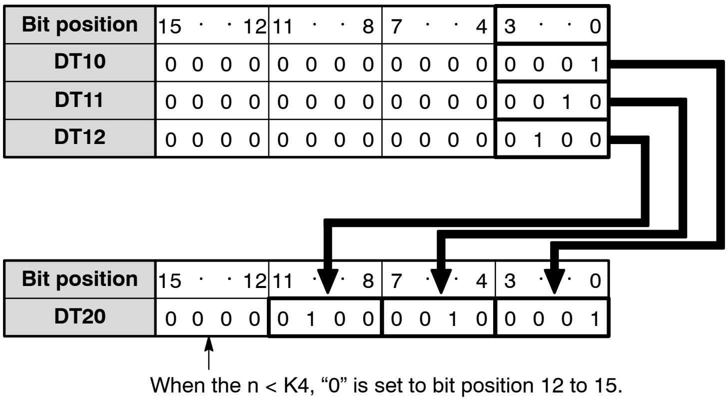
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Extracts lower 4 bits of data registers DT12 to DT10, combines the extracted data, and stores it in data register DT20 when trigger R20 turns on.



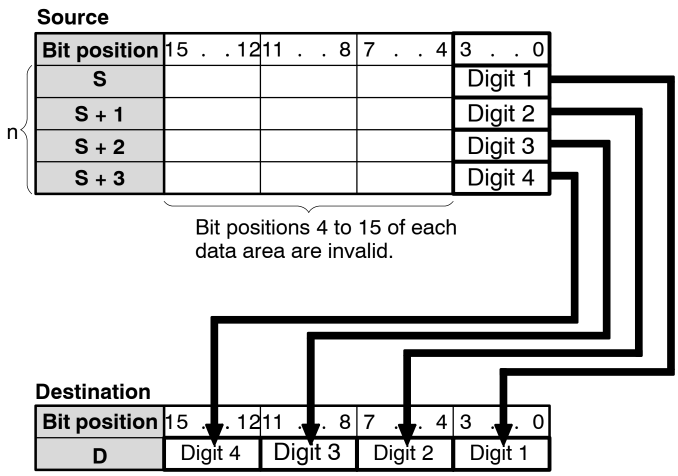
Description

Extracts the lower 4 bits (bit positions 0 to 3) of each specified area, starting from the 16-bit area specified by S and combines the extracted data into one word. The result is stored in the 16-bit area specified by D.

n specifies the number of data areas to be extracted.
(range of n: K0 to K4)

When K0 is specified for n, this instruction is not executed.

When n < K4, “0” is automatically set to positions at D where the corresponding 16-bit source data does not exist.



Flag conditions

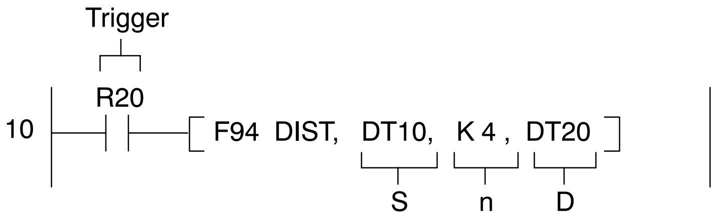
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – The number of data areas to be combined “n” is K5 or more.

F94 (DIST) P94 (PDIST) 16-bit data distribute

Outline Divides the specified 16-bit data into four 4-bit units and distributes the divided data into the lower 4 bits of the specified 16-bit areas.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P94 (PDIST)" is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 20
11	F 94 (DIST)
	DT 10
	K 4
	DT 20

S	16-bit equivalent constant or 16-bit area to be divided (source)
n	16-bit equivalent constant or 16-bit area to specify number of data to be divided
D	Starting 16-bit area for storing divided data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Divides the 16-bit data of data register DT10 into 4-bit units and the divided data is stored in the lower 4 bits (bit positions 0 to 3) of data registers DT20 to DT23 when trigger R20 turns on.

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT10	0 1 1 1	0 0 1 1	0 0 0 1	0 0 0 0

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
DT20	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
DT21	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1
DT22	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
DT23	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 1

Description

Divides the 16-bit data specified by S into 4-bit units and distributes the divided data into the lower 4 bits (bit positions 0 to 3) of 16-bit areas starting from D.

n specifies the number of data divisions.(range of n: K0 to K4)

When K0 is specified for n, this instruction is not executed.

Source

n

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
S	Digit 4	Digit 3	Digit 2	Digit 1

Destination

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
D				Digit 1
D + 1				Digit 2
D + 2				Digit 3
D + 3				Digit 4

Bit positions 4 to 15 are filled with 0s.

Flag conditions

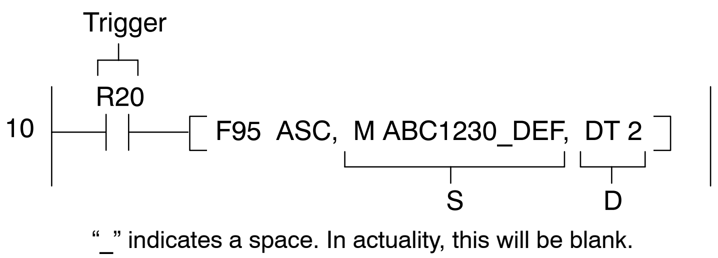
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The number n is 5 or more.
  - Transferring address specified by D to n data, and when area is exceeded.

F95 (ASC) P95 (PASC) Character → ASCII code

Outline Converts character constants to ASCII code.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P95 (PASC)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 20
11	F 95 (ASC)
	M ABC1230_DEF
	DT 2

S	Character constants (max. 12 letters) (source)
D	Starting 16-bit area for storing 6-word ASCII code (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant			Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Converts the character constants “ABC1230_DEF” to ASCII code when trigger R20 turns on. The ASCII code is stored in data registers DT2 to DT7.

Character constant

[S]	**Character constants**	M A B C 1 2 3 0 D E F

R20: on

[D]	**Data register**	DT7				DT6				DT5				DT4				DT3				DT2			
	ASCII HEX code	2	0	4	6	4	5	4	4	2	0	3	0	3	3	3	2	3	1	4	3	4	2	4	1
	ASCII character			F		E		D				0		3		2		1		C		B		A	

SPACE

When the number of character constants specified by S is less than 12, the ASCII code H20 (SPACE) is stored in the extra destination area.

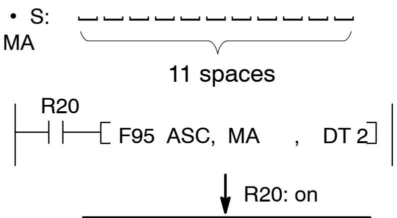
Description

Converts the character constants specified by S to ASCII code. The converted ASCII code is stored in 6 words starting from the 16-bit area specified by D.

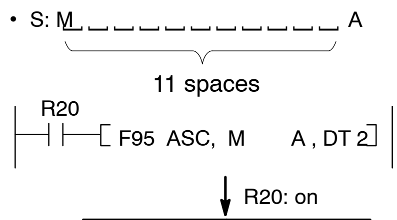
Precautions during programming

The character constant M can be input with the programming tool software.

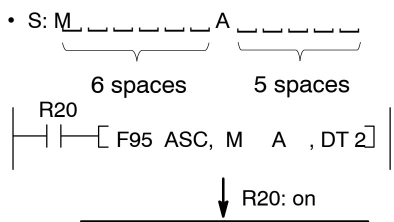
Conversion example of one character constant “A”



	Higher byte		Lower byte	
DT2:	H20	(␣)	H41	(A)
DT3:	H20	(␣)	H20	(␣)
DT4:	H20	(␣)	H20	(␣)
DT5:	H20	(␣)	H20	(␣)
DT6:	H20	(␣)	H20	(␣)
DT7:	H20	(␣)	H20	(␣)



	Higher byte		Lower byte	
DT2:	H20	(␣)	H20	(␣)
DT3:	H20	(␣)	H20	(␣)
DT4:	H20	(␣)	H20	(␣)
DT5:	H20	(␣)	H20	(␣)
DT6:	H20	(␣)	H20	(␣)
DT7:	H41	(A)	H20	(␣)



	Higher byte		Lower byte	
DT2:	H20	(␣)	H20	(␣)
DT3:	H20	(␣)	H20	(␣)
DT4:	H20	(␣)	H20	(␣)
DT5:	H20	(␣)	H41	(A)
DT6:	H20	(␣)	H20	(␣)
DT7:	H20	(␣)	H20	(␣)

Flag conditions

- Error flag (R9007): Turns on and stays on when the last area for ASCII code exceeds the limit (6 words: six 16-bit areas).
- Error flag (R9008): Turns on for an instant when the last area for ASCII code exceeds the limit (6 words: six 16-bit areas).

ASCII HEX code

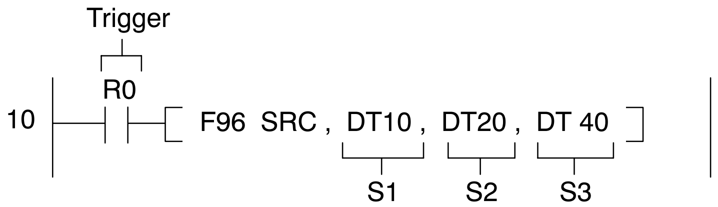
									b7								
									b6	0	0	0	0	1	1	1	1
									b5	0	0	1	1	0	0	1	1
									b4	0	1	0	1	0	1	0	1
										Most significant digit							
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0		ASCII HEX code	0	1	2	3	4	5	6	7
				0	0	0	0	Least significant digit	0	NUL	DLE	SPACE	0	@	P		p
				0	0	0	1		1	SOH	DC1	!	1	A	Q	a	q
				0	0	1	0		2	STX	DC2	”	2	B	R	b	r
				0	0	1	1		3	ETX	DC3	#	3	C	S	c	s
				0	1	0	0		4	EOT	DC4	$	4	D	T	d	t
				0	1	0	1		5	ENQ	NAK	%	5	E	U	e	u
				0	1	1	0		6	ACK	SYN	&	6	F	V	f	v
				0	1	1	1		7	BEL	ETB	’	7	G	W	g	w
				1	0	0	0		8	BS	CAN	(	8	H	X	h	x
				1	0	0	1		9	HT	EM	)	9	I	Y	i	y
				1	0	1	0		A	LF	SUB	*	:	J	Z	j	z
				1	0	1	1		B	VT	ESC	+	;	K	[	k	{
				1	1	0	0		C	FF	FS	,	<	L	\	l	|
				1	1	0	1		D	CR	GS	–	=	M	]	m	}
				1	1	1	0		E	SO	RS	.	>	N	^	n	~
				1	1	1	1		F	SI	US	/	?	O	_	o	DEL

F96 (SRC) P96 (PSRC) 16−bit data search

Outline Searches for a specified value in a block of 16-bit areas.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P96 (PSRC)" is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction
10	ST R 0
11	F 96 (SRC)
	DT 10
	DT 20
	DT 40

S1	16-bit equivalent constant or 16-bit area to store the searched value
S2	Starting 16-bit area of the block
S3	Ending 16-bit area of the block

Operands

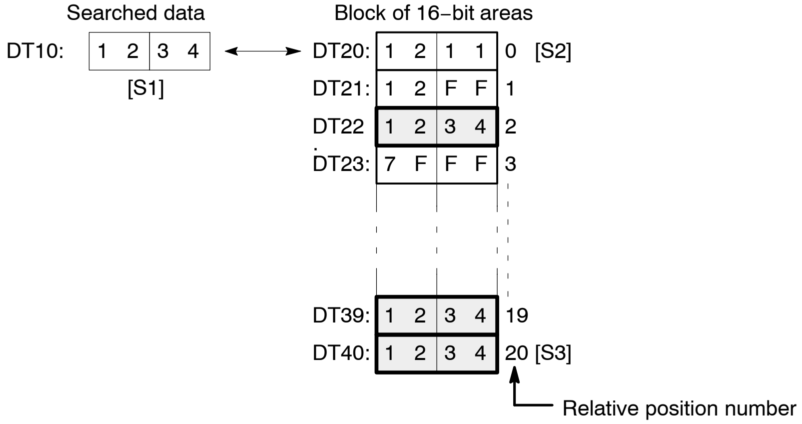
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S3	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Searches for the value given in data register DT10 in the block of data register DT20 through DT40 when trigger R0 turns on.
For example, to search the area of the value called H1234, “H1234” would be written to DT10.



If DT22, DT39, and DT40 match the searched data, the following occurs.

– If the number of registers matching the searched data = 3
  “K3” is stored in DT9037 (with the FP0 T32, FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH: DT90037).
– If the position of the first matching data (the relative position number) = 2
  “K2” is stored in DT9038 (with the FP0 T32, FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH: DT90038).

Description

Searches for values matching S1 in the block of 16-bit areas specified by S2 (starting area) through S3 (ending area).
When the search operation is performed, the search results are stored as follows.

– The number of data items that match S1 is stored in special data register DT9037 (with the FP0 T32, FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH: DT90037).
– The position of the first matching data item, counting from the starting 16-bit area S2, is stored in special data register DT9038 (with the FP0 T32, FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH: DT90038).

Starting area S2 and ending area S3 should:

– Be the same type of operand.
– Satisfy S2 ≦ S3.

Data is searched from S2 to S3.

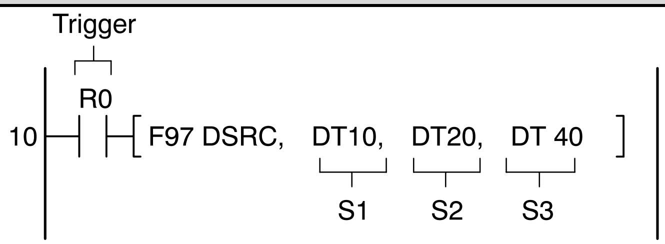
Flag conditions

• Error flag (R9007): Turns on and stays on when:
• Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – S2 > S3.

F97 (DSRC) P97 (PDSRC) 32-bit data search

Outline Searches for a specified value in a block of 32-bit areas.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P97 (PDSRC)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F97 (DSRC) DT 10 DT 20 DT 40

S1	32-bit equivalent constant or 32-bit area to store the searched value in
S2	Starting 32-bit area of the block
S3	Ending 32-bit area of the block

Operands

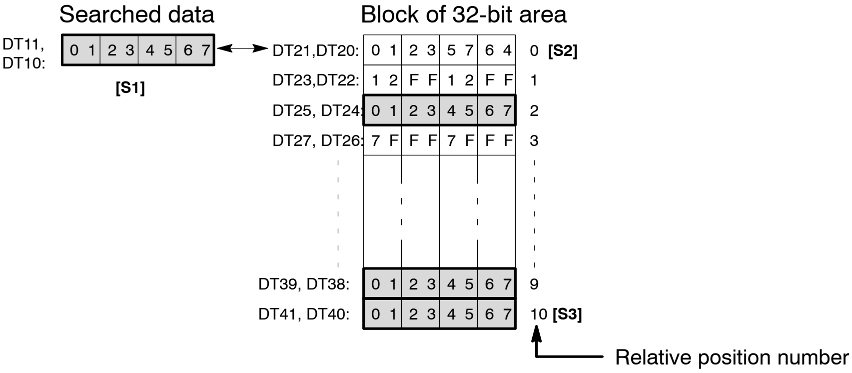
Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	N/A
S2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S3	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

Searches for the value given in data registers DT10 and DT11 in the block of data register DT20 through DT40 when trigger R0 turns on.
For example, to search the area of the value called H01234567, “H01234567” would be written to DT10 and DT11.



If “DT24 and DT25”, “DT38 and DT39”, and “DT40 and DT41” match the searched data, the following occurs.

– The number “K3” of data items that match the searched data (DT10 and DT11) is stored in special data register DT90037.
– The position “K2” of the first matching data item, counting from data register DT20, is stored in special data register DT90038.

Description

Searches for values matching S1 in the block of 32-bit areas specified by S2 (starting area) through S3 (ending area) when the trigger turns on.

When the search operation is performed, the search results are stored as follows.

– The number of data items that match S1 is stored in special data register DT90037.
– The position of the first matching data item, counting from the starting 32-bit area S2, is stored in special data register DT90038.

The starting area S2 and ending area S3 should:

– Be the same type of operand.
– Satisfy S2 $\leqq$ S3.

Data S1 is searched from S2 to S3.

Precautions during programming

If “0” or an even number is specified in S2, specify an even number in S3 as well.

If an odd number is specified in S2, specify an odd number in S3 as well.

Flag conditions

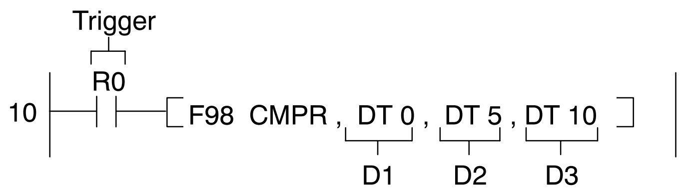
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – S2 > S3.

F98 (CMPR) P98 (PCMPR) Data table shift−out and compress

Outline Shifts out non-zero data stored at the highest address of the table to the specified area and compresses the data in the table to the higher address.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P98 (PCMPR)” is not available.

Program example

Ladder Diagram



Boolean: Address	Boolean: Instruction
10	ST R 0
11	F 98 (CMPR)
	DT 0
	DT 5
	DT 10

D1	Starting 16-bit area of the data table
D2	Ending 16−bit area of the data table
D3	16-bit area for storing the shift−out data

Operands

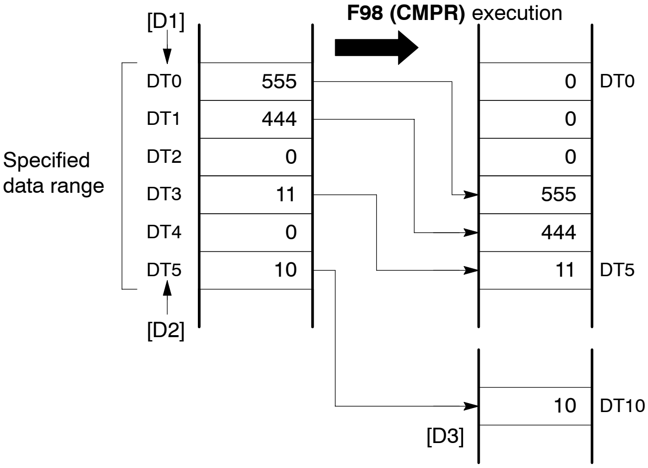
Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D3	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

If the execution condition (trigger) R0 is on, the contents of data register DT5 are sent to data register DT10. Also, in the range from DT0 to DT5, non−zero contents are stored in sequential order, starting from DT5. The "0 (zero)" is set in the other areas of the data table.

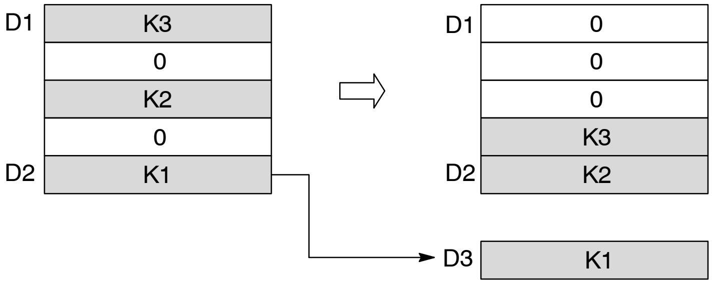


Description

The data in the table specified by D1 and D2 is rearranged as follows:

– Contents of D2 (highest address) is shifted out to the area specified by D3.

– Non−zero data is shifted (compressed) in sequential order, in the direction of the higher address in the specified range.



Starting area D1 and ending area D2 should be the same type of operand.

Be sure to specify D1 and D2 with "D1 ≦ D2".

If all data in the data table specified by D1 and D2 is 0, 0 is set in D3.

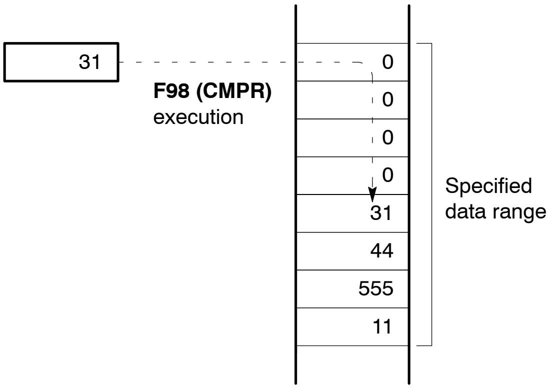
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - D1 > D2
  - D1 and D2 are not in the same memory area.

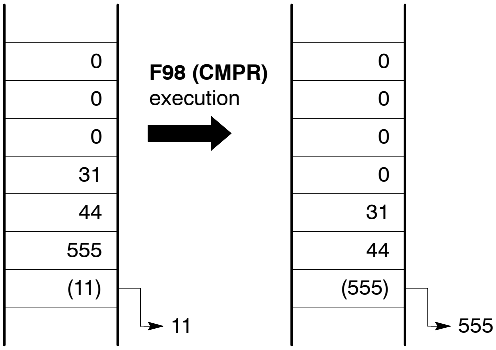
Application example

In combination with the **F99 (CMPW)**/**P99 (PCMPW)** instruction, this can be used to construct an optional buffer.

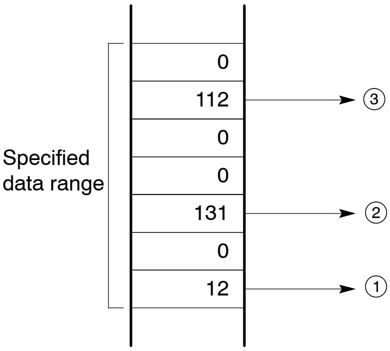
(1) Executing the **F99 (CMPW)**/**P99 (PCMPW)** instruction
When data items are written to the first address of the buffer (the area of the specified range), they are stored and accumulated in the buffer in sequential order. The oldest data will be stored in the last address of the buffer.



(2) Executing the **F98 (CMPR)**/**P98 (PCMPR)** instruction
When the data in the last address of the buffer (the area of the specified range) has been read, data can be extracted in sequential order, starting from the oldest data.
The rest of the data in the buffer is shifted in the direction of the first address, so normally, the oldest data at that point is stored in the last address of the buffer.



This can be used to extract valid non−zero data from the data written in random order.



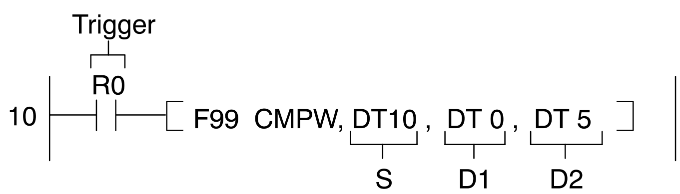
Each time the **F98 (CMPR)** instruction is executed, data is extracted in sequential order, from 1 to 3.

F99 (CMPW) P99 (PCMPW) Data table shift−in and compress

Outline Shifts in data to the smallest address of the specified data table and compresses the data in the table toward the higher address.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P99 (PCMPW)” is not available.

Program example

Ladder Diagram



Boolean				
Address	**Instruction**			
10	ST		R	0
11	F	99	(CMPW)	
	DT			10
	DT			0
	DT			5

S	16-bit equivalent constant or 16-bit area for storing shift−in data
D1	Starting 16-bit area of the data table
D2	Ending 16-bit area of the data table

Operands

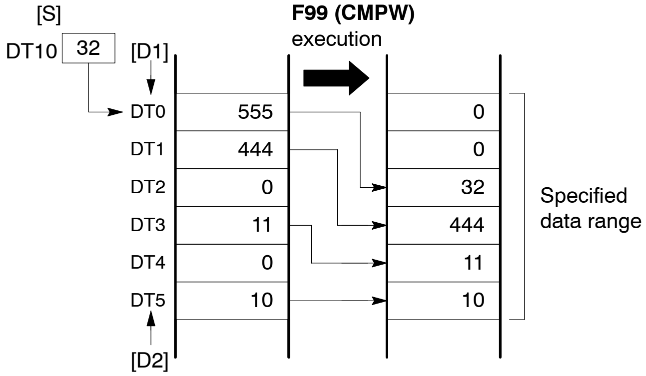
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	IX (*2)	IY (*3)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0R, FPΣ and FP−X.
(*2) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

If the execution condition (trigger) R0 is on, the contents of data register DT10 are sent to data register DT0. Also, in the range from DT0 to DT5, non−zero contents are stored in sequential order, starting from DT5. The "0 (zero)" is set in the other areas of the data table.



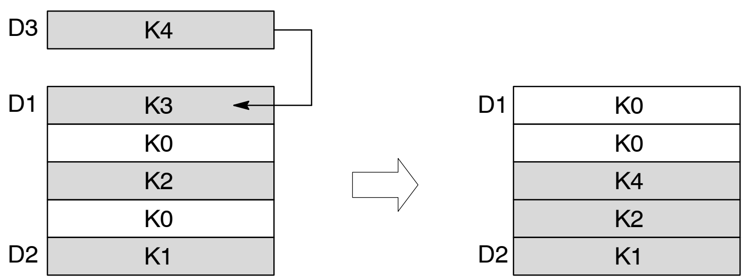
Note

Because the contents of S are written to DT0, the original contents of DT0 (for example, "555") are overwritten.

Description

The data in the table specified by D1 and D2 is rearranged as follows:

– Data specified by S is shifted in to the area specified by D1 (starting address).
– Non−zero data is shifted (compressed) in sequential order, in the direction of the higher address in the specified range.



Starting area D1 and ending area D2 should be the same type of operand.

Be sure to specify D1 and D2 with "D1 ≦ D2".

If the content of S is "0", only a compressed shift is carried out.

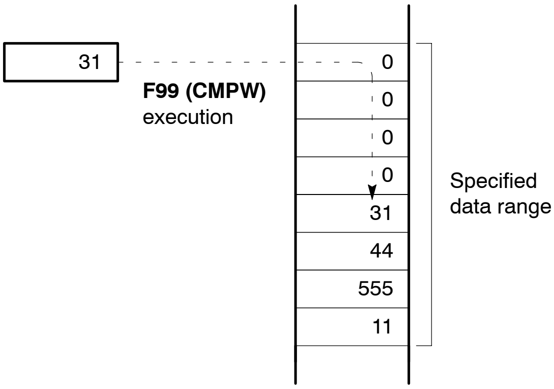
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – D1 > D2
  – Starting area D1 and ending area D2 are not in the same memory area.

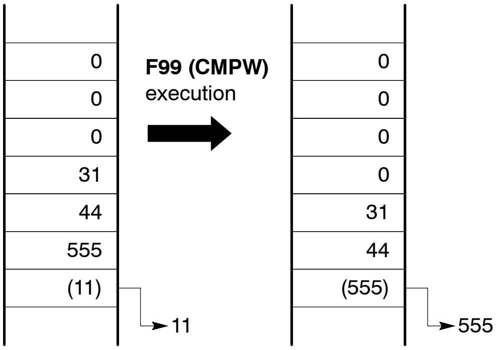
Application example

In combination with the **F98 (CMPR)**/**P98 (PCMPR)** instruction, this can be used to construct an optional buffer.

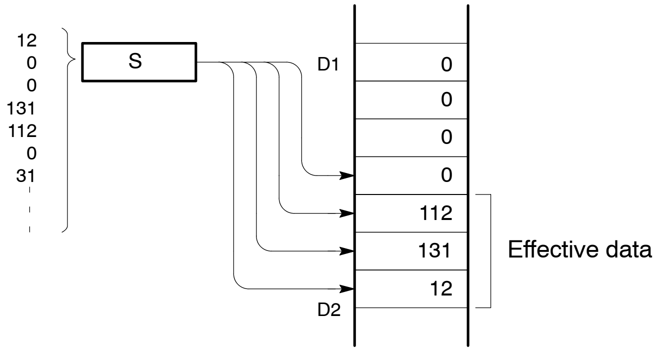
(1) Executing the **F99 (CMPW)**/**P99 (PCMPW)** instruction
When data items are written to the first address of the buffer (the area of the specified range), they are stored and accumulated in the buffer in sequential order. The oldest data will be stored in the last address of the buffer.



(2) Executing the **F98 (CMPR)**/**P98 (PCMPR)** instruction
When the data in the last address of the buffer (the area of the specified range) has been read, data can be extracted in sequential order, starting from the oldest data.
The rest of the data in the buffer is shifted in the direction of the first address, so normally, the oldest data at that point is stored in the last address of the buffer.



This can be used to extract valid non−zero data from the data written in random order.



Executing the **F99 (CMPW)** instruction causes only the valid data to be stored.

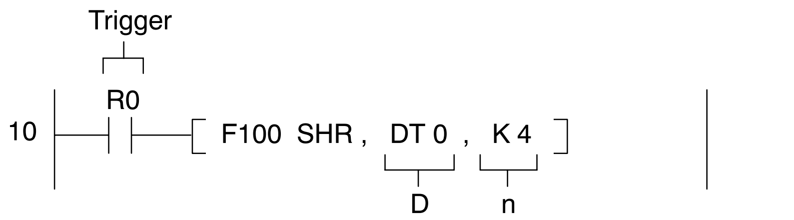
F100 (SHR) P100 (PSHR)

Right shift of multiple bits (n bits) in a 16−bit data

Outline Shifts a specified number of bits to the right in bit units.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P100 (PSHR)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F100 (SHR)
	DT 0
	K 4

D	16-bit area to be shifted to the right
n	16-bit equivalent constant or 16-bit area (specifies number of shifted bits)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Shifts 4 bits in data register DT0 to the right when trigger R0 turns on.

The data in bit position 3 is transferred to special internal relay R9009 (carry flag).

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	1 0 1 0	0 1 1 0	1 1 0 0	1 0 1 1

R0: on

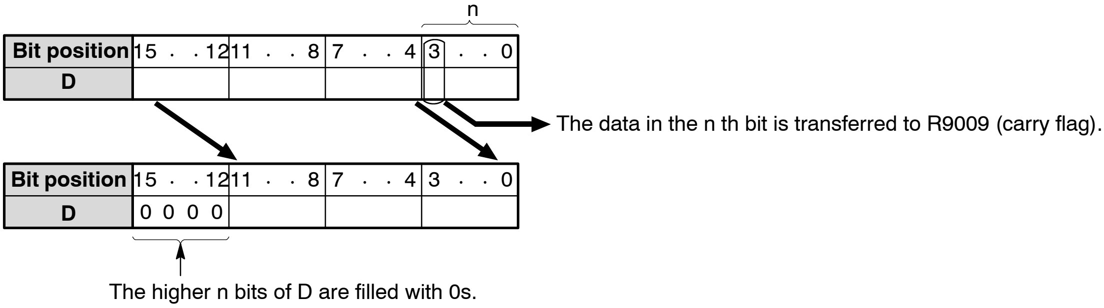
The data in bit position 3 is transferred to R9009 (carry flag).

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	0 0 0 0	1 0 1 0	0 1 1 0	1 1 0 0

In this case, the higher 4 bits of DT0 are filled with 0s.

Description

Shifts n bits of the 16-bit data area specified by D to the right (to the lower bit position).

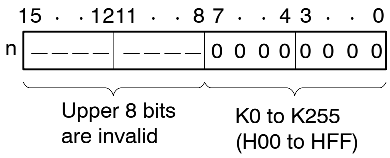


When n bits are shifted to the right,

– The higher n bits of the 16-bit data area are filled with 0s.

– The data in the n th bit is transferred to special internal relay R9009 (carry flag).

The n is effective only for the lower 8 bits of the 16−bit data. The amount of the shift can be specified within a range of 1 bit to 255 bits.



Flag conditions

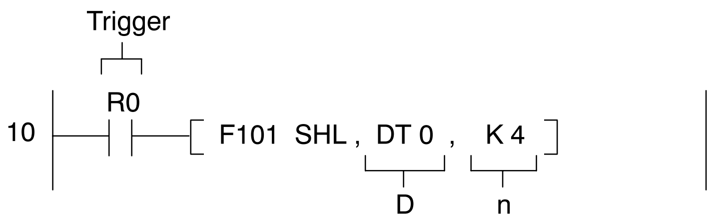
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the content of bit transferred to R9009 (n th bit) is recognized as 1.

F101 (SHL) P101 (PSHL) Left shift of multiple bits (n bits) in a 16−bit data

Outline Shifts a specified number of bits to the left in bit units.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P101 (PSHL)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F101		(SHL)
	DT		0
	K		4

D	16-bit area to be shifted to the left
n	16-bit equivalent constant or 16-bit area (specifies number of shifted bits)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

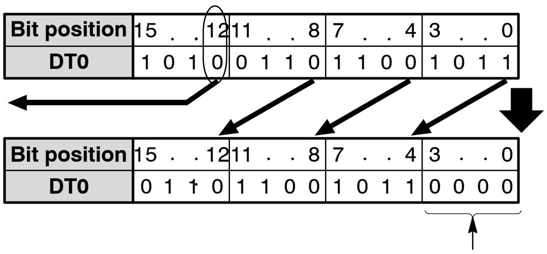
A: Available
N/A: Not Available

Explanation of example

Shifts 4 bits in data register DT0 to the left when trigger R0 turns on.

The data in bit position 12 is transferred to special internal relay R9009 (carry flag).

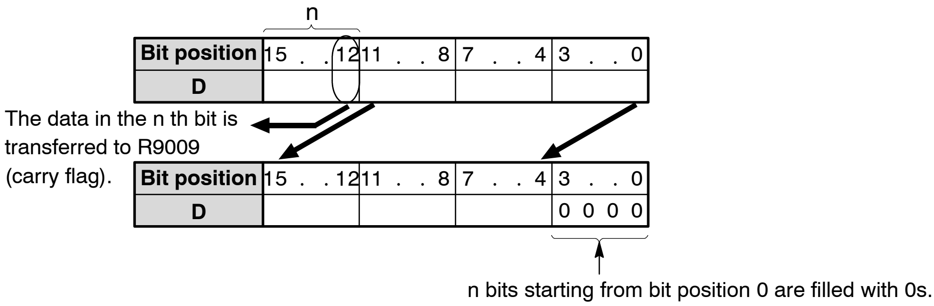
The data in bit position 12 is transferred to R9009 (carry flag).



In this case, the lower 4 bits of DT0 are filled with 0s.

Description

Shifts n bits of the 16-bit area specified by D to the left (to the higher bit position).

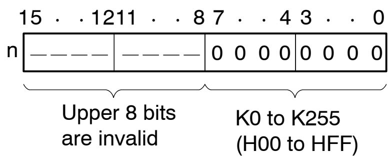


When the n bits are shifted to the left,

– The n bits starting from bit position 0 are filled with 0s.

– The data in the n th bit is transferred to special internal relay R9009 (carry flag).

The n is effective only for the lower 8 bits of the 16−bit data. The amount of the shift can be specified within a range of 1 bit to 255 bits.



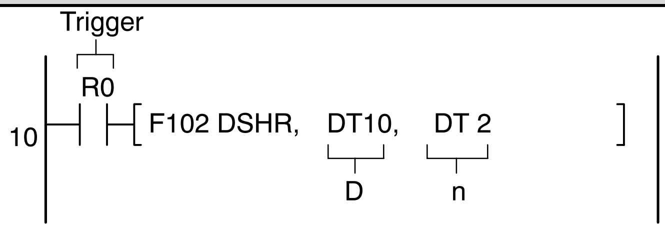
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the content of bit transferred to R9009 (n th bit) is recognized as 1.

F102 (DSHR) P102 (PDSHR) Right shift of n bits in a 32-bit data

Outline Shifts a specified number of bits to the right in bit units.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P102 (PDSHR)" is not available.

Program example

<table>
<tr><th rowspan="2">Ladder Diagram</th><th colspan="2">Boolean</th></tr>
<tr><th>Address</th><th>Instruction</th></tr>
<tr><td>
</td><td>10
11</td><td>ST R 0
F102 (DSHR)
DT 10
DT 2</td></tr>
</table>

D	Lower 16-bit area of 32-bit data area to be shifted to the right
n	16-bit equivalent constant or 16-bit area (specifies number of shifted bits) Range of n: K0 to K255 (H0 to HFF)

Operands

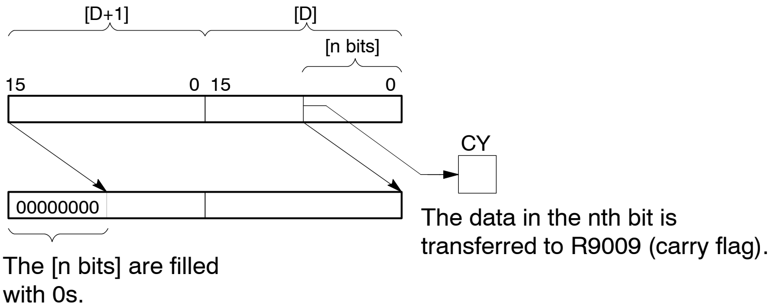
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Description

Shifts n bits of the 32-bit data area specified by D to the right (to the lower bit position) when the trigger turns on.

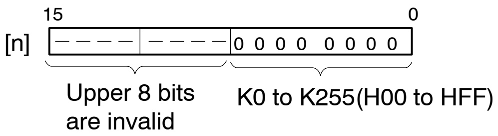


When n bits are shifted to the right,

– The higher n bits of the 16-bit data area specified by D are filled with 0s.

– The data in the nth bit is transferred to special internal relay R9009 (carry flag).

Only the lower eight bits of the 16-bit data [n] are effective. Select the amount of the shift within the range 1 to 255 bits.



When [n] is specified using K0, the contents of D and D+1 and the special internal relay R9009 (carry flag) do not change.

When [n] is specified using K32 or higher, the contents of D and D+1 change to 0.

Flag conditions

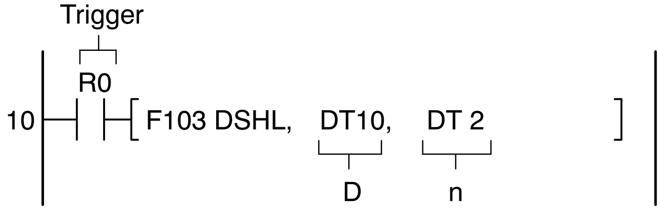
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Content in the nth bit from LBS (least significant bit) turns on for an instant when the bit transferred to R9009 is recognized as 1.

F103 (DSHL) P103 (PDSHL) Left shift of n bits in a 32-bit data

Outline Shifts a specified number of bits to the left in bit units.
For the FP0R/FPΣ/FP-X/FP0/FP-e, the P type high-level instruction "P103 (PDSHL)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F103 (DSHL)
	DT 10
	DT 2

D	Lower 16-bit area of 32-bit data to be shifted to the left
n	16-bit equivalent constant or 16-bit area (specifies number of shifted bits) Range of n: K0 to K255 (H0 to HFF)

Operands

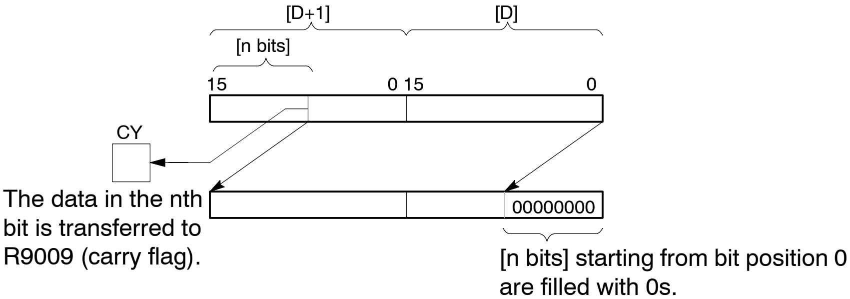
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP-X.

A: Available
N/A: Not Available

Description

Shifts n bits of the 32-bit area specified by D to the left (to the higher bit position) when the trigger turns on.



When the n bits are shifted to the left,
- The n bits starting from bit position 0 are filled with 0s.
- The data in the nth bit is transferred to special internal relay R9009 (carry flag).

Only the lower eight bits of the 16-bit data [n] are effective. Select the amount of the shift within the range 1 to 255 bits.

$$[n]\quad \overset{15}{}\underbrace{- - - -\ - - - -}_{\text{Upper 8 bits are invalid}}\ \underbrace{0\,0\,0\,0\ \ 0\,0\,0\,0}_{\text{K0 to K255(H00 to HFF)}}\overset{0}{}$$

When [n] is specified using K0, the contents of D and D+1 and the carry flag do not change.

When [n] is specified using K32 or higher, the contents of D and D+1 change to 0.

Flag conditions

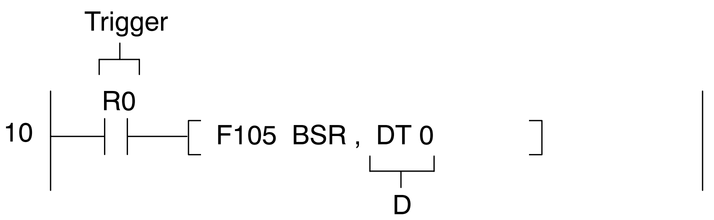
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Content in the nth from LSB (least significant bit) turns on for an instant when the bit transferred to R9009 is recognized as 1.

F105 (BSR) P105 (PBSR)

Right shift of one hexadecimal digit (4 bits)

Outline Shifts one hexadecimal digit (4 bits) of the specified 16-bit data to the right.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P105 (PBSR)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F105 (BSR) DT 0

D	16-bit area to be shifted to the right

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Shifts one hexadecimal digit (4 bits) in data register DT0 to the right when trigger R0 turns on.

The data in hexadecimal digit position 1 (bit positions 0 to 3) is shifted out and transferred to the lower digit position (bit positions 0 to 3) of special data register DT9014 (with the FP0 T32, FPΣ, FP−X, FP2, FP2SH and FP10SH: DT90014).

DT0	Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0	
	Binary	1 0 0 1	1 0 0 1	1 0 0 1	1 0 0 1	
	Hexadecimal	9	9	9	9	(H9999)

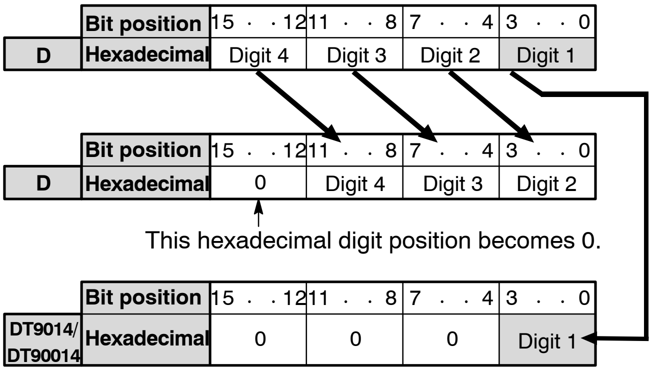
R0: on

DT0	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0	
	Binary	0 0 0 0	1 0 0 1	1 0 0 1	1 0 0 1	
	Hexadecimal	0	9	9	9	(H999)

DT9014/ DT90014	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0	
	Binary	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1	
	Hexadecimal	0	0	0	9	(H9)

Description

Shifts one hexadecimal digit (4 bits) of the 16-bit area specified by D to the right (to the lower digit position).



When one hexadecimal digit (4 bits) is shifted to the right,

– The data in hexadecimal digit position 1 (bit positions 0 to 3) of the 16−bit area specified by D is shifted out and is transferred to the lower digit (bit positions 0 to 3) of special data register DT9014 (with the FP0 T32, FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH: DT90014).

– The hexadecimal digit 4 (bit positions 12 to 15) of the 16-bit area specified by D becomes 0.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

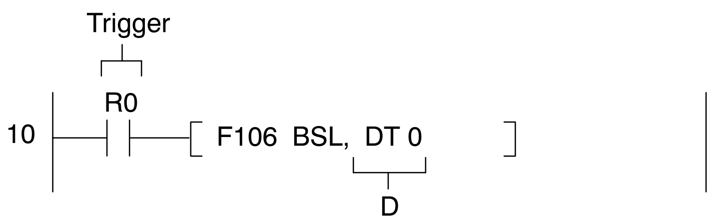
F106 (BSL) P106 (PBSL)

Left shift of one hexadecimal digit (4 bits)

Outline Shifts one hexadecimal digit (4 bits) of the specified 16-bit data to the left.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P106 (PBSL)" is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	0
11	F106		(BSL)
	DT		0

D	16-bit area to be shifted to the left

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Shifts one hexadecimal digit (4 bits) in data register DT0 to the left when trigger R0 turns on.

The data in hexadecimal digit position 4 (bit positions 12 to 15) is shifted out and transferred to the lower digit position (bit positions 0 to 3) of special data register DT9014 (with the FP0 T32, FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH: DT90014).

DT0	Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
	Binary	1 0 0 0	0 0 0 1	0 0 0 0	0 0 0 0
	Hexadecimal	8	1	0	0

(H8100)

R0: on

DT0	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
	Binary	0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0
	Hexadecimal	1	0	0	0

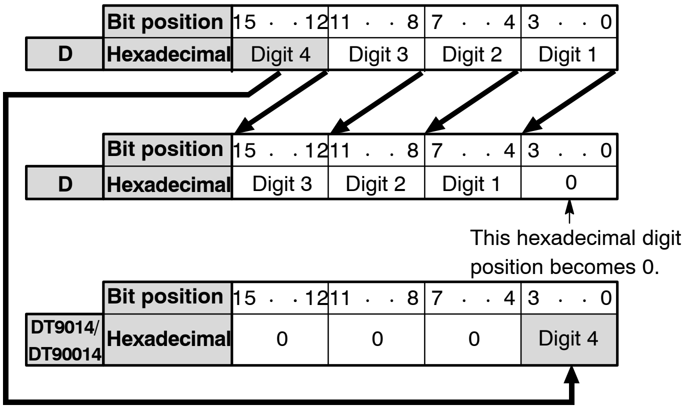
(H1000)

DT9014/ DT90014	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
	Binary	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
	Hexadecimal	0	0	0	8

(H8)

Description

Shifts one hexadecimal digit (4 bits) of the 16-bit area specified by D to the left (to the higher digit position).



When one hexadecimal digit (4 bits) is shifted to the left,

– The data in hexadecimal digit position 4 (bit positions 12 to 15) of the 16−bit data specified by D is shifted out and is transferred to the lower digit (bit positions 0 to 3) of special data register DT9014 (with the FP0 T32, FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH: DT90014).

– The hexadecimal digit position 1 (bit positions 0 to 3) of the 16-bit data specified by D becomes 0.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

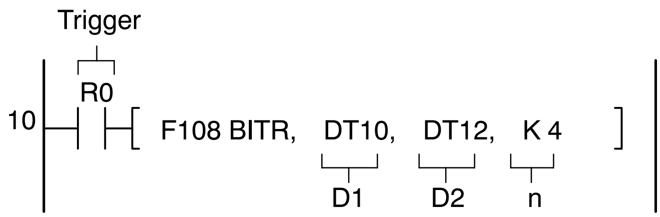
F108 (BITR) P108 (PBITR)

Right shift of multiple bits of 16-bit data range

Outline

Shifts multiple bits of a specified 16-bit data range to the right.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P108 (PBITR)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F108 (BITR) DT 10 DT 12 K 4

D1	Starting 16-bit area
D2	Ending 16-bit area
n	16-bit equivalent constant or 16-bit area to specify number of shifted bits

Operands

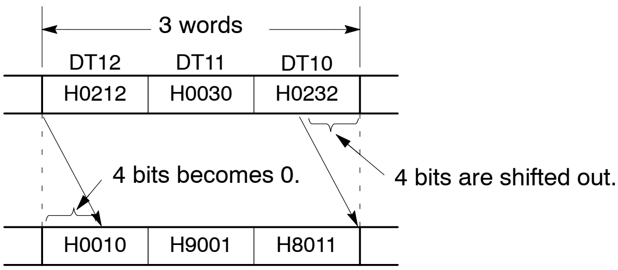
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

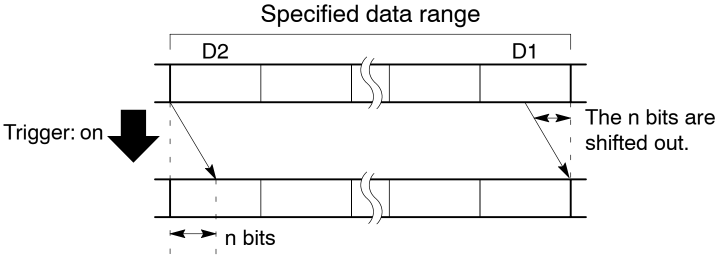
Explanation of example

Shifts 4 bits in data range (3 words) from DT10 through DT12 to the right when trigger R0 turns on.



Description

Shifts n bits of the data range specified by D1 (starting) and D2 (ending) to the right (to the lower bit position) when the trigger turns on.



D1 and D2 should be:

– The same type of operand.

– D1 $\leqq$ D2.

When n bits are shifted to the right,

– The n bits of starting 16-bit area D1 are shifted out.

– The n bits in the ending 16-bit area D2 becomes 0.

0 to 15 can be specified for n. When 0 is specified, no operation takes place.

Flag conditions

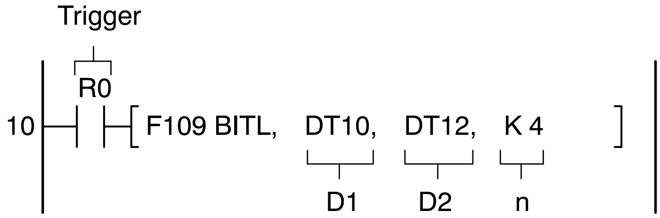
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - D1 > D2.
  - The number n is 16 or more.

F109 (BITL) P109 (PBITL) Left shift of multiple bits of 16-bit data range

Outline Shifts multiple bits of a specified 16-bit data range to the left.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P109 (PBITL)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 10
11	F109 (BITL)
	DT 10
	DT 12
	K 4

D1	Starting 16-bit area
D2	Ending 16-bit area
n	16-bit equivalent constant or 16-bit area to specify number of shifted bits

Operands

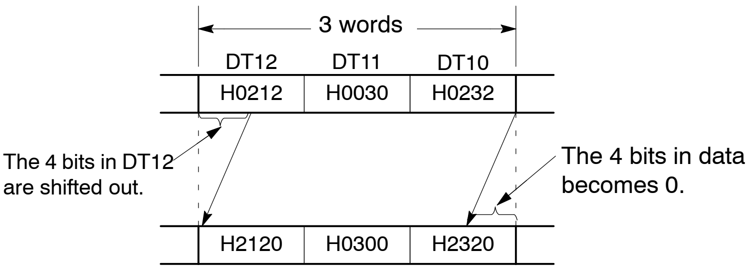
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

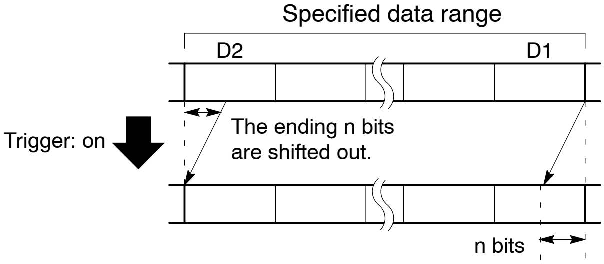
Explanation of example

Shifts 4 bits in the data range (3 words) from DT10 through DT12 to the left when trigger R0 turns on.



Description

Shifts n bits of the data range specified by D1 (starting) and D2 (ending) to the left (to the higher bit position) when the trigger turns on.



D1 and D2 should be:

– The same type of operand.

– D1 ≦ D2.

When n bits are shifted to the left,

– The n bits of ending 16-bit area D2 is shifted out.

– The n bits in the starting 16-bit area D1 becomes 0.

0 to 15 can be specified for n. When 0 is specified, no operation takes place.

Flag conditions

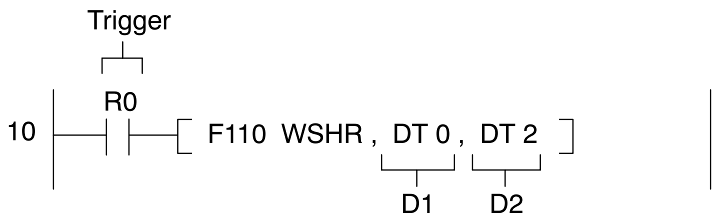
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - D1 > D2.
  - The number n is 16 or more.

F110 (WSHR) P110 (PWSHR) Right shift of one word (16 bits) of 16-bit data range

Outline Shifts one word (16 bits) of a specified 16-bit data range to the right.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P110 (PWSHR)" is not available.

Program example

Ladder Diagram



Address	Instruction		
10	ST	R	0
11	F110	(WSHR)	
	DT		0
	DT		2

D1	Starting 16-bit area
D2	Ending 16-bit area

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	I	K	H	
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

A: Available
N/A: Not Available

Explanation of example

Shifts one word (16 bits) of the data range (3 words) from DT0 through DT2 to the right when trigger R0 turns on.

Specified data range (3 words)

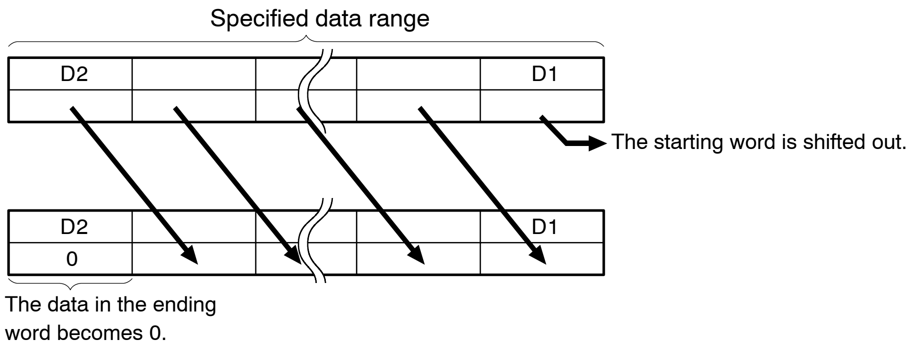
Data register	DT2	DT1	DT0
Hexadecimal	0 2 1 2	0 0 3 0	0 2 3 2

The data in DT0 is shifted out.

Data register	DT2	DT1	DT0
Hexadecimal	0 0 0 0	0 2 1 2	0 0 3 0

Description

Shifts one word (16 bits) of the data range specified by D1 (starting) and D2 (ending) to the right (to the lower word address).



Starting area D1 and ending area D2 should be:

- The same type of operand.
- D1 ≦ D2.

When one word (16 bits) is shifted to the right,

- The starting word (D1) is shifted out.
- The data in the ending word (D2) becomes 0.

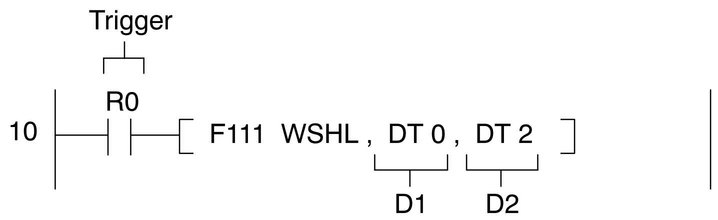
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - D1 > D2.

F111 (WSHL) P111 (PWSHL) Left shift of one word (16 bits) of 16-bit data range

Outline Shifts one word (16 bits) of a specified 16-bit data range to the left.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction (PWSHL)” is not available.

Program example



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F111	(WSHL)	
	DT		0
	DT		2

D1	Starting 16-bit area
D2	Ending 16-bit area

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	I	K	H	
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

A: Available
N/A: Not Available

Explanation of example

Shifts one word (16 bits) of the data range (3 words) from DT0 through DT2 to the left when trigger R0 turns on.

Specified data range (3 words)

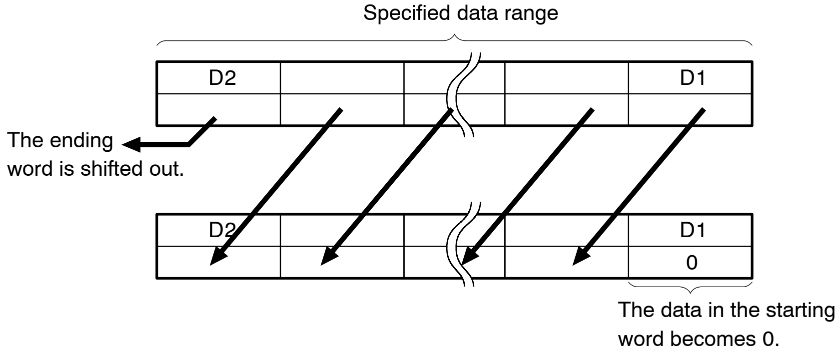
Data register	DT2	DT1	DT0
Hexadecimal	0 2 1 2	0 0 3 0	0 2 3 2

The data in DT2 is shifted out.

Data register	DT2	DT1	DT0
Hexadecimal	0 0 3 0	0 2 3 2	0 0 0 0

Description

Shifts one word (16 bits) of the data range specified by D1 (starting) and D2 (ending) to the left (to the higher word address).



Starting area D1 and ending area D2 should be:

- The same type of operand.
- D1 ≦ D2.

When one word (16 bits) is shifted to the left,

- The ending word (D2) is shifted out.
- The data in the starting word (D1) becomes 0.

Flag conditions

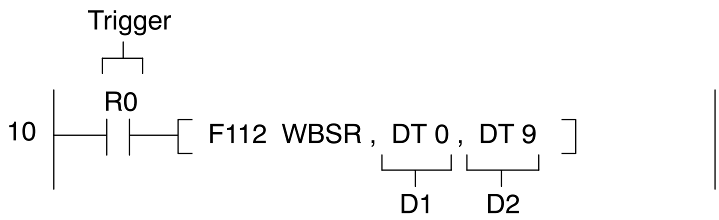
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - D1 > D2.

F112 (WBSR) P112 (PWBSR) Right shift of one hexadecimal digit (4−bit) of 16-bit data range

Outline Shifts one hexadecimal digit (4 bits) of a specified 16-bit data range to the right.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F112 (WBSR)
	DT 0
	DT 9

D1	Starting 16-bit area
D2	Ending 16-bit area

Operands

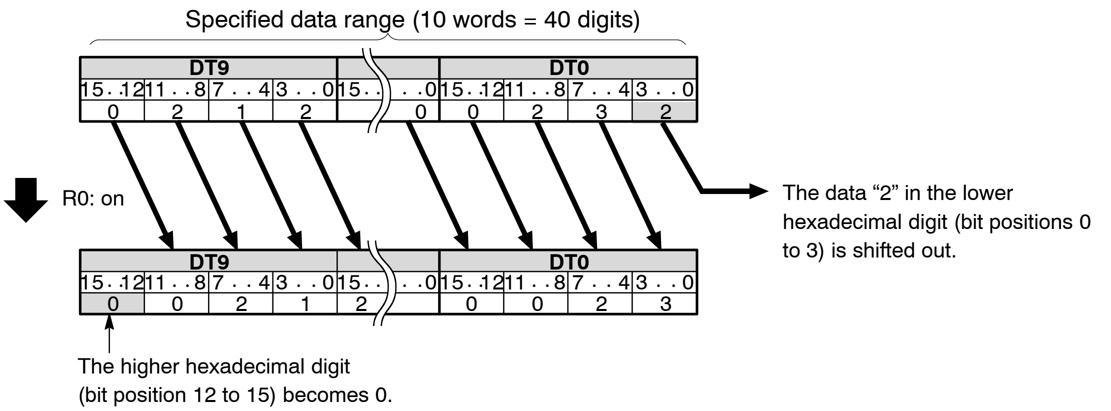
Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	I	K	H	
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

A: Available
N/A: Not Available

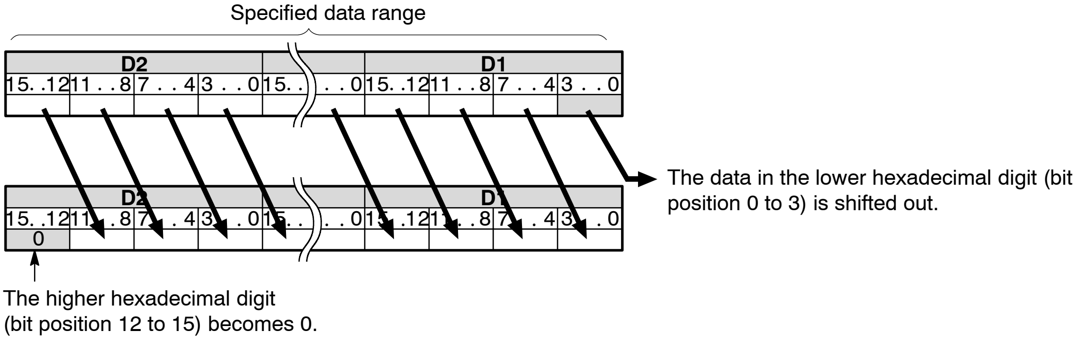
Explanation of example

Shifts one hexadecimal digit (4 bits) of the data range (10 words) from DT0 through DT9 to the right when trigger R0 turns on.



Description

Shifts one hexadecimal digit (4 bits) of the data range specified by D1 (starting) and D2 (ending) to the right (to the lower digit position).



Starting area D1 and ending area D2 should be:

– The same type of operand.

– D1 $\leqq$ D2.

When the hexadecimal digit (4 bits) is shifted to the right,

– The data at the lower hexadecimal digit (bit positions 0 to 3) of the 16-bit data specified by D1 is shifted out.

– The data at the higher hexadecimal digit (bit positions 12 to 15) in the 16-bit data specified by D2 becomes 0.

Flag conditions

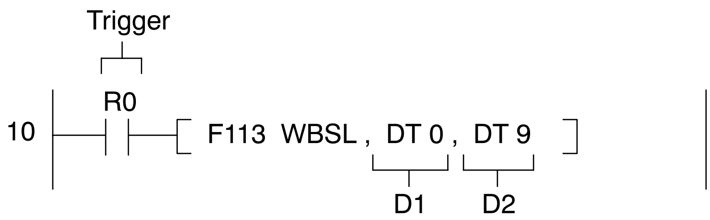
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – D1 > D2.

F113 (WBSL) P113 (PWBSL)

Left shift of one hexadecimal digit (4−bit) of 16-bit data range

Outline Shifts one hexadecimal digit (4 bits) of a specified 16-bit data range to the left.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P113 (PWBSL)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F113 (WBSL)
		DT 0
		DT 9

D1	Starting 16-bit area
D2	Ending 16-bit area

Operands

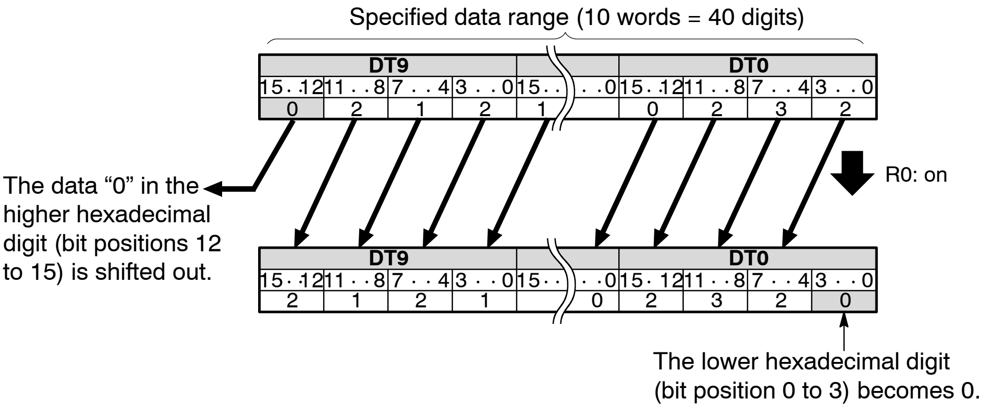
Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	I	K	H	
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

A: Available
N/A: Not Available

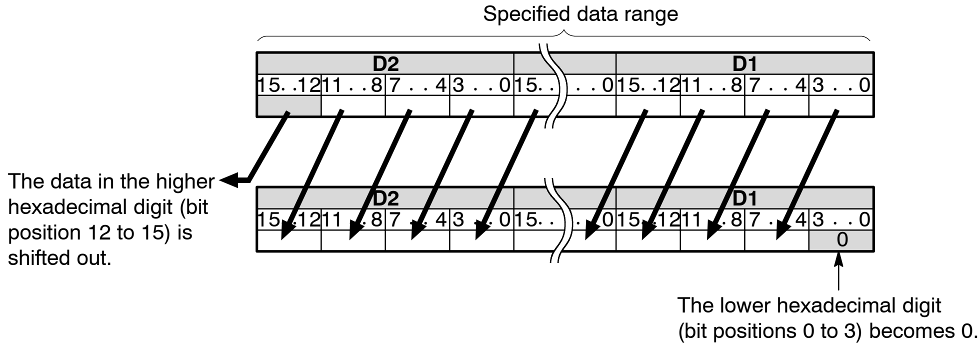
Explanation of example

Shifts one hexadecimal digit (4 bits) of the data range (10 words) from DT0 through DT9 to the left when trigger R0 turns on.



Description

Shifts one hexadecimal digit (4 bits) of the data range specified by D1 (starting) and D2 (ending) to the left (to the higher digit position).



Starting area D1 and ending area D2 should be:

– The same type of operand.

– D1 ≦ D2.

When the hexadecimal digit (4 bits) is shifted to the left,

– The data at the higher hexadecimal digit (bit positions 12 to 15) of the 16-bit data specified by D2 is shifted out.

– The data at the lower hexadecimal digit (bit positions 0 to 3) in the 16-bit data specified by D1 becomes 0.

Flag conditions

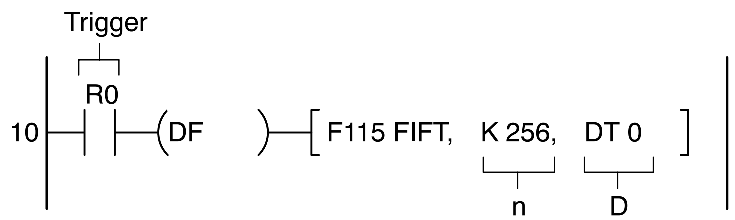
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – D1 > D2.

F115 (FIFT) P115 (PFIFT) FIFO buffer definition

Outline Defines the FIFO buffer conditions.
For the FP0R/FPΣ/FP–X/FP0/FP–e, the P type high–level instruction "P115 (PFIFT)" is not available.

Program example

Ladder Diagram



Address	Instruction		
10	ST	R	0
11	DF		
12	F115		(FIFT)
	K		256
	DT		0

n	16-bit equivalent constant or 16–bit area for specifying the memory size of FIFO buffer
D	Starting 16-bit area of FIFO buffer

Operands

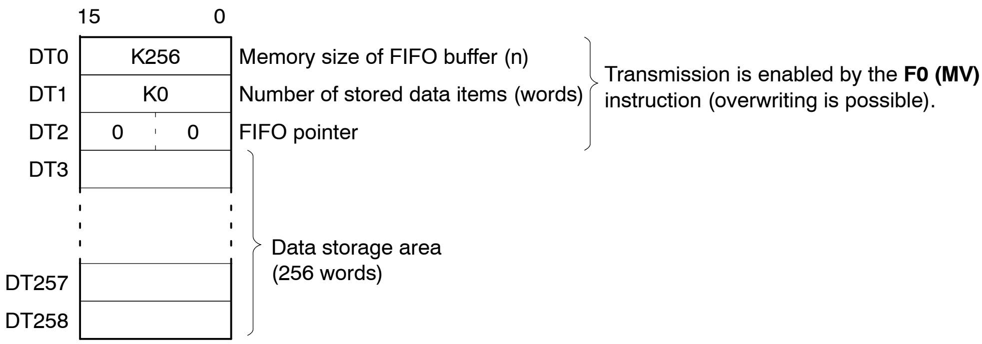
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD (*1)	FL	IX (*2)	IY (*3)	K	H	
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0R, FPΣ and FP–X.
(*2) With the FP0R, FPΣ, FP–X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*3) With the FP0R, FPΣ, FP–X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

When the execution condition (trigger) R0 is on, the area headed by DT0 is defined in the FIFO buffer area. The size of the FIFO buffer (K256) is stored in DT0, the number of data items stored is stored in DT1 (with a default value of K0), and the FIFO pointer (with a default value of H0000) is stored in DT2.
When n = K256, the 256 words from DT3 to DT258 are defined as the data storage area.

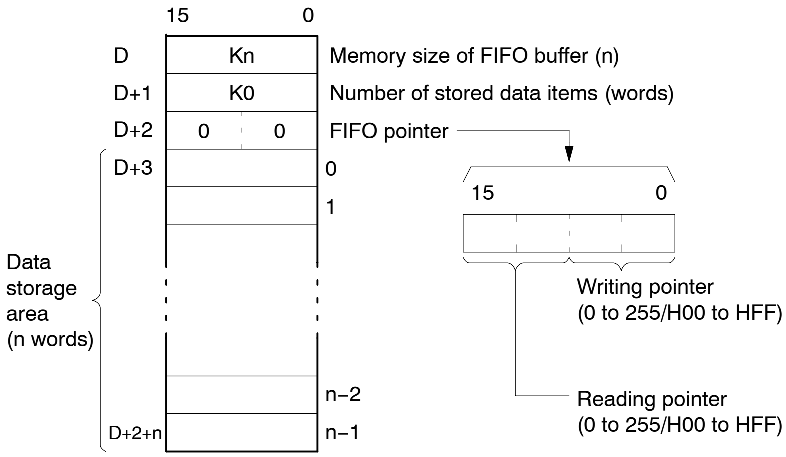


Description

This defines the area used as the FIFO buffer. A data storage area of n words (n = K1 to K256) is defined for the area specified by D.

Definition of the area using the **F115 (FIFT)** instruction should be carried out only once, before writing to or reading from the FIFO buffer. Normally, reading and writing are disabled while this instruction is being executed.

When the **F115 (FIFT)** instruction is executed, the FIFO buffer area is defined as follows.



When the **F115 (FIFT)** instruction is executed, the following are stored as default values: D = n (the value specified by the **F115 (FIFT)** instruction), D+1 = K0, and D+2 = H0000.

Flag conditions

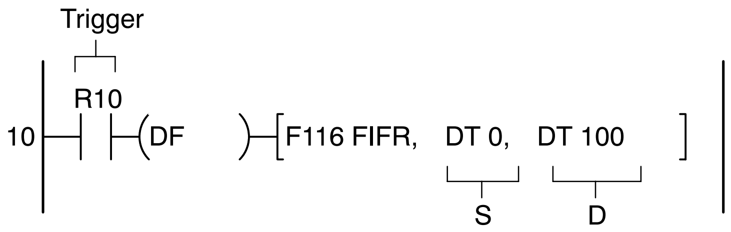
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - n = 0.
  - n > 256.
  - The area specified by n exceeds the limit.

F116 (FIFR) P116 (PFIFR) Data read from FIFO buffer

Outline Reads data from the FIFO buffer.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P116 (PFIFR)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 10
11	DF
12	F116 (FIFR)
	DT 0
	DT 100

S	Starting 16-bit area of FIFO buffer
D	16-bit area for storing data read from FIFO buffer

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	IX (*2)	IY (*3)	K	H	
S	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

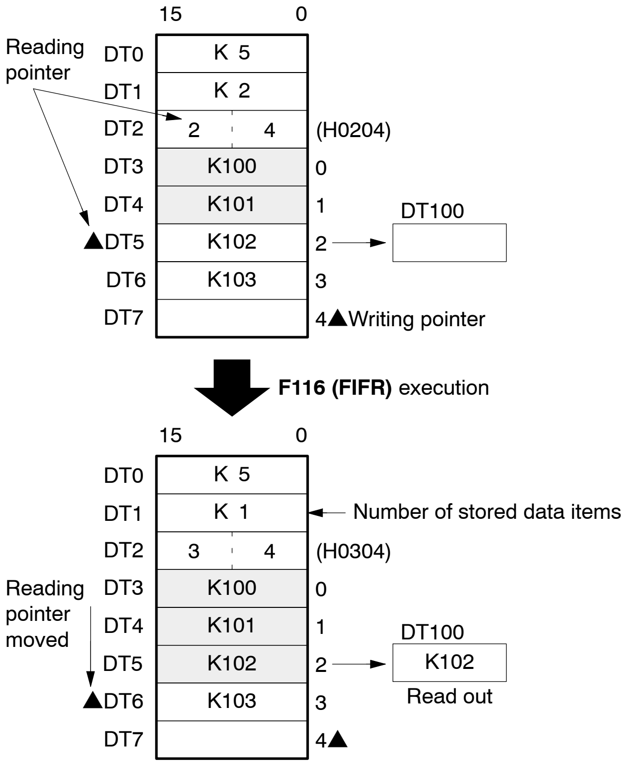
(*1) This cannot be used with the FP0R, FPΣ and FP−X.
(*2) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

When the execution condition (trigger) R10 is on, data is read from the FIFO buffer area headed by DT0, and is stored in DT100.

When the reading pointer is 2



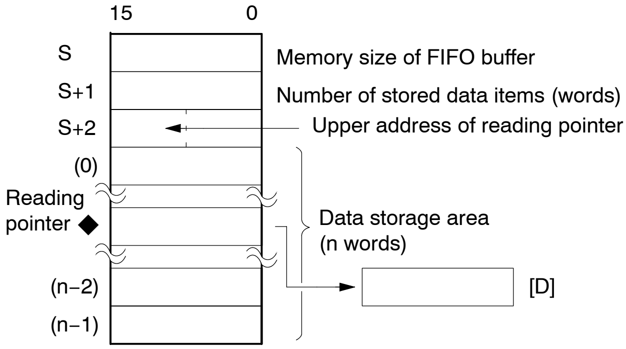
The contents of DT5, which is indicated by the reading pointer 2 are sent to DT100.

After the data has been read, 1 is subtracted to the contents of DT1 (the number of stored data items), and the reading pointer moves to 3.
(The next time that reading is carried out, the contents of DT100 are sent to DT6, indicated by the 3.)

Description

These instructions read data from the FIFO buffer headed by the area specified by S, and store it in the area specified by D.
S should specify the beginning of the FIFO buffer defined by the **F115 (FIFT)** instruction.

Reading of data is done starting from the address specified by the reading pointer when the instruction is executed.



- (0), (n−2) and (n−1) are addresses assigned to the data storage area.
- n is the value specified by the **F115 (FIFT)** instruction.

The reading pointer is stored in the upper eight bits of the third word of the FIFO buffer area, and is indicated by an address in the data storage area.
The actual address is the value of the leading address in the FIFO buffer area specified by S, plus 3, plus the value of reading pointer (the value of which only the first byte is a decimal value).

When the reading is executed, 1 is subtracted from the number of stored data items, and the reading pointer is incremented by 1.

☞ Notes

- **An error occurs if this is executed when the number of stored data items is 0.
No data is set for D.**
- **Reading is only carried out when the reading pointer is not equal to the writing pointer.**
- **If this is executed when the reading pointer is indicating the final address in the FIFO buffer (the n defined by the FIFO instruction minus 1), the reading pointer is set to 0.**

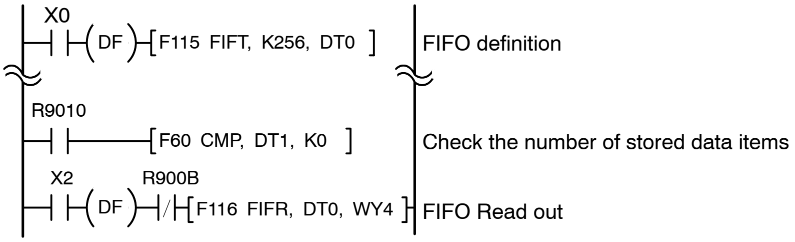
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The size (n) of the FIFO specified by S is n = 0, or when n > 256.
  - The number of stored data items of the FIFO = 0.
  - The number of stored data items of the FIFO >FIFO size (n).
  - The final address of the FIFO based on the FIFO size (n) exceeds the area.
  - The FIFO reading pointer > FIFO size (n).
  - The FIFO reading pointer is K256 (H100) or higher after the data has been read.

Precautions during programming

An error occurs if the **F116 (FIFR)** instruction is executed when the number of stored data items (S+1) is 0.

In the program noted below, the **F116 (FIFR)** instruction is not executed if the number of stored data items is 0.



How the FIFO buffer is used

The FIFO buffer is a buffer area in which data is stored in the order in which it is written to the buffer, and from which it is then read out in the stored order, starting from the first data item stored. It is convenient for storing objects on carrier lines and buffer lines in sequential order.

Usage procedure

The area to be used is defined as the FIFO buffer using the **F115 (FIFT)** instruction. (This should be done only once, before reading or writing is done.)

Data should be written using the **F117 (FIFW)** instruction, and read using the **F116 (FIFR)** instruction.

Writing data

When data is written, the data items are stored in sequential order, starting from the first data storage area. The writing pointer indicates the next area to which data is to be written.

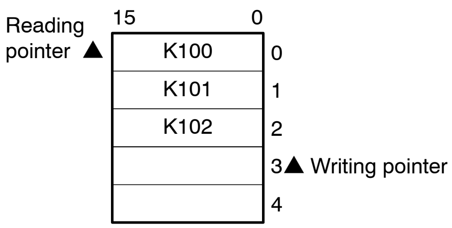
If the data storage area becomes full, further data writing is inhibited.

Reading data

When data is read, data is transferred starting from the first data item stored. The reading pointer indicates the next area from which data is to be read.

An error occurs if an attempt is made to read data when no data has been written to the data storage area.

Example of data storage area



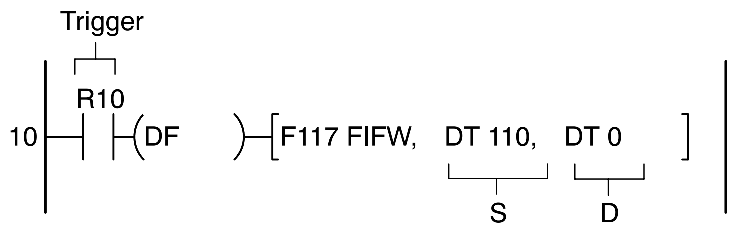
If data is written in the status shown above, the data will be stored in the area indicated by 3. The writing pointer moves to 4 (the next data item will be written to 4).
If data is read, it will be read from the area indicated by 0. The reading pointer moves to 1 (the next data item will be read from 1).

F117 (FIFW) P117 (PFIFW) Data write to FIFO buffer

Outline Writes data to the FIFO buffer.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P117 (PFIFW)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	DF		
12	F117		(FIFW)
	DT		110
	DT		0

S	16-bit equivalent constant or 16−bit area for storing data to write in FIFO buffer
D	Starting 16-bit area of FIFO buffer

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	IX (*2)	IY (*3)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

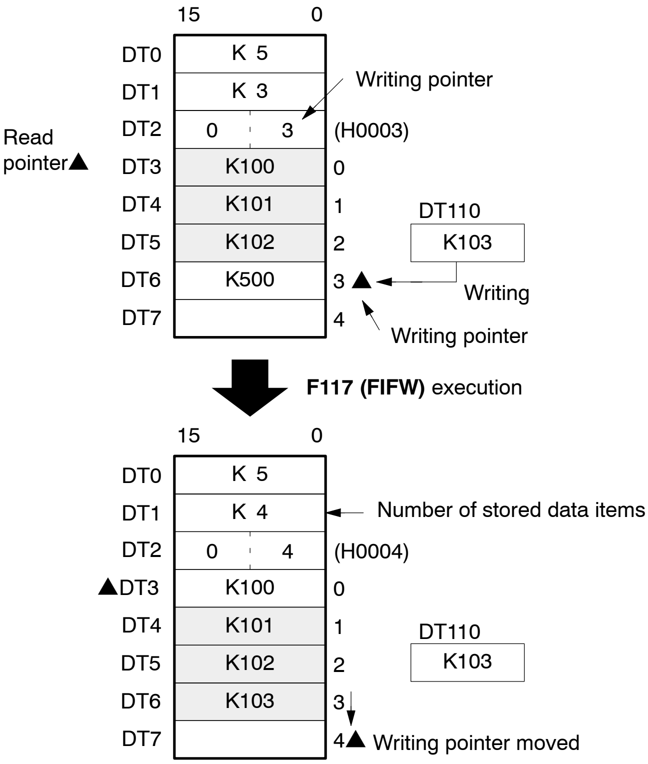
(*1) This cannot be used with the FP0R, FPΣ and FP−X.
(*2) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

When the execution condition (trigger) R10 is on, the contents of DT110 are written to the FIFO buffer area headed by DT0.

When the writing pointer is 3



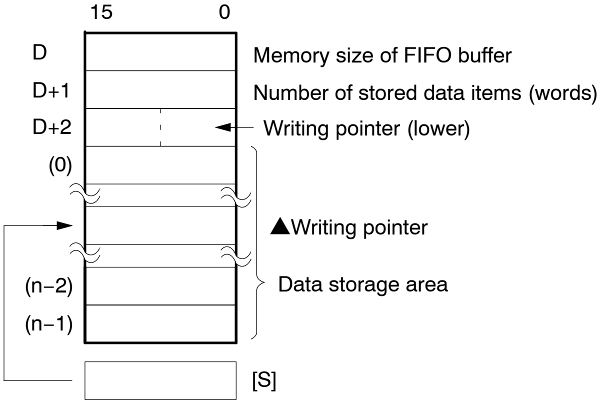
The contents “103” of DT110 are sent to DT6, which is indicated by the pointer 3.

After the data has been written, 1 is added to the contents of DT1 (the number of stored data items), and the writing pointer moves to 4.
(The next time that writing is carried out, the contents of DT110 are written to DT7, indicated by the 4.)

Description

The 16−bit data specified by S will be stored in the FIFO buffer headed by the area specified by D.
D should specify the beginning of the FIFO buffer defined by the **F115 (FIFT)** instruction.

The specified data is written to the address indicated by the writing pointer when the instruction is executed.



- (0), (n−2) and (n−1) are addresses assigned to the data storage areas.
- n is the value specified by the **F115 (FIFT)** instruction.

The writing pointer is stored in the lower eight bits of the third word of the FIFO buffer area, and is indicated by a relative position in the data storage area.
The actual address is the value of the leading address in the FIFO buffer area specified by D, plus 3, plus the value of writing pointer (the value of which only the lower byte is a decimal value).

When the writing is executed, 1 is added to the number of stored data items, and the writing pointer is incremented by 1.

Notes

- **An error occurs if this is executed when the FIFO buffer is full (the number of stored data items = the size n of the FIFO defined by the FIFT instruction). Writing is inhibited.**
- **If this is executed when the writing pointer is indicating the final address in the FIFO buffer (the ”n” value defined by the FIFT instruction), the writing pointer will be set to 0.**

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The size (n) of the FIFO specified by D is n = 0, or when n > 256.
  - The number of stored data items of the FIFO > FIFO size (n).
  - The final address of the FIFO based on the FIFO size (n) exceeds the area.
  - The writing pointer of the FIFO > FIFO size (n).
  - The FIFO writing pointer is K256 (H100) or higher after the data has been written.

How the FIFO buffer is used

The FIFO buffer is a buffer area in which data is stored in the order in which it is written to the buffer, and from which it is then read out in the stored order, starting from the first data item stored. It is convenient for storing objects on carrier lines and buffer lines in sequential order.

Usage procedure

The area to be used is defined as the FIFO buffer using the **F115 (FIFT)** instruction. (This should be done only once, before reading or writing is done.)

Data should be written using the **F117 (FIFW)** instruction, and read using the **F116 (FIFR)** instruction.

Writing data

When data is written, the data items are stored in sequential order, starting from the first data storage area. The writing pointer indicates the next area to which data is to be written.

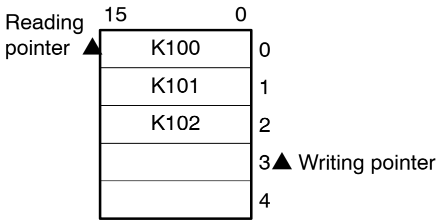
If the data storage area becomes full, further data writing is inhibited.

Reading data

When data is read, data is transferred starting from the first data item stored. The reading pointer indicates the next area from which data is to be read.

An error occurs if an attempt is made to read data when no data has been written to the data storage area.

Example of data storage area

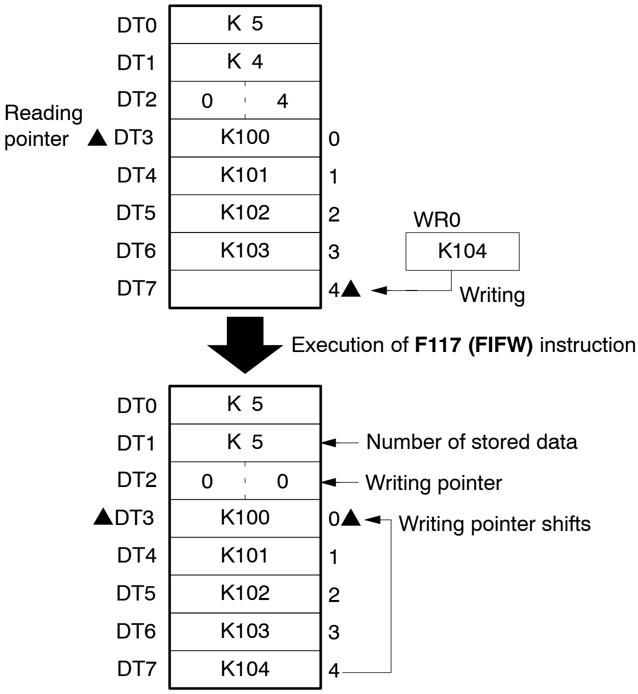


If data is written in the status shown above, the data will be stored in the area indicated by 3. The writing pointer moves to 4 (the next data item will be written to 4).
If data is read, it will be read from the area indicated by 0. The reading pointer moves to 1 (the next data item will be read from 1).

Precautions when using this instruction

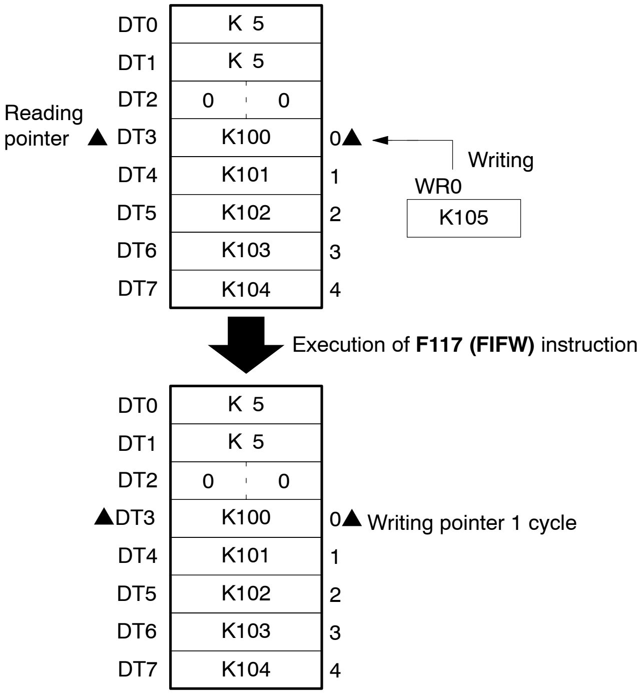
If data is received which exceeds the capacity of the buffer, an operation error will occur.

Example: If the writing pointer is at the end of the FIFO buffer



When the F117 (FIFW) instruction is executed, after data is written to the final address (4) in the buffer, the writing pointer becomes the first address (0).

Example: When the writing pointer has made one complete cycle

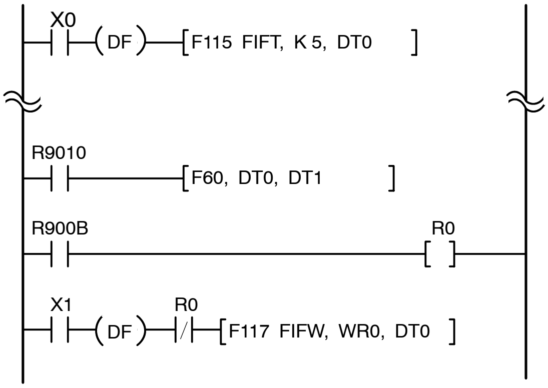


An error occurs, and processing is not carried out.

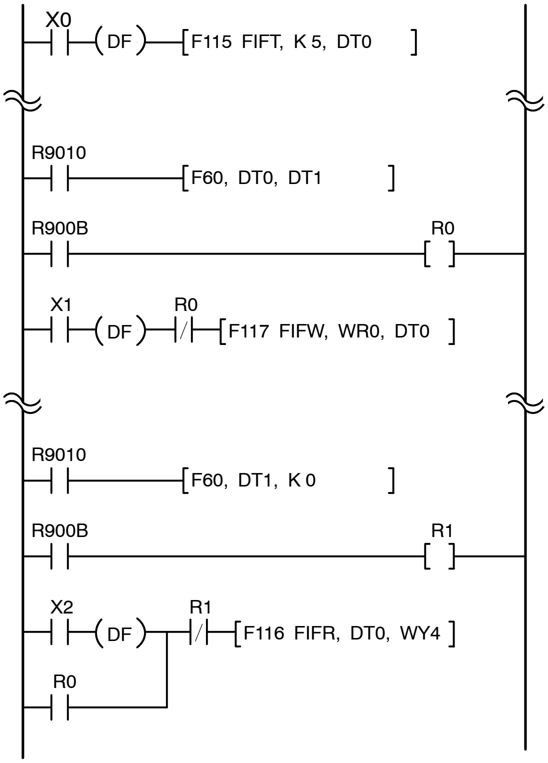
Because the number of data items stored in the FIFO buffer (DT1=5) exceeds the size of the FIFO buffer (DT0=5), the operation is not executed, and an operation error occurs.

Measures to avoid operation errors

Do not execute the **F117 (FIFW)** instruction using the comparison instruction. Avoid executing the **F117 (FIFW)** instruction when the size of the FIFO buffer (DT0) is equal to the number of data items stored in the buffer (DT1).



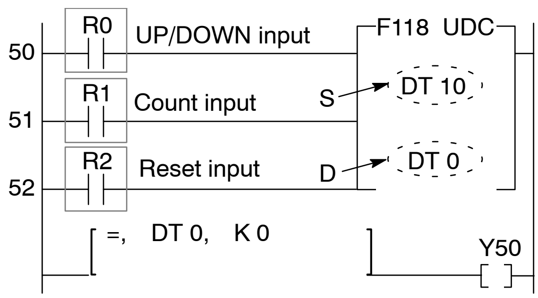
Execute the **F117 (FIFW)** instruction after executing the **F116 (FIFR)** instruction.



F118 (UDC) UP/DOWN counter

Outline Sets the UP/DOWN counter.

Program example



Address	Instruction		
50	ST	R	0
51	ST	R	1
52	ST	R	2
53	F118		(UDC)
	DT		10
	DT		0
58	ST=		
63	DT		0
	K		0
	OT	Y	50

S	16-bit equivalent constant or 16-bit area for counter preset value
D	16-bit area for counter elapsed value

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	I	K	H	
S	A	A	A	A	A	A	A	A	A	N/A	A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A

(*1) This cannot be used with the FP0 and FP−e.

(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

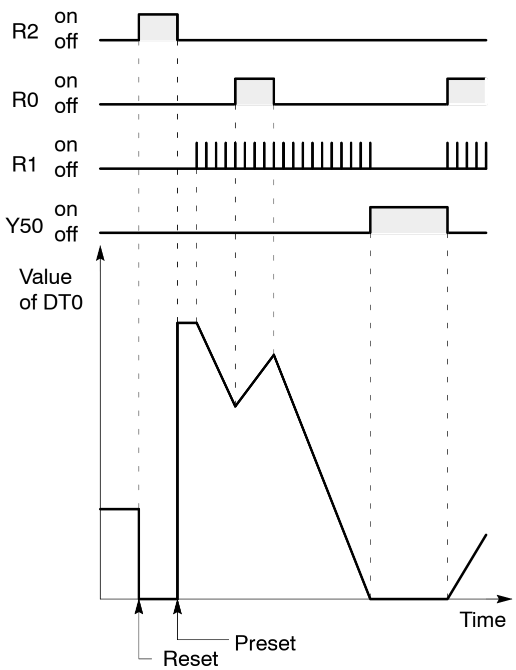
A: Available
N/A: Not Available

Explanation of example

The program on the preceding page shows an example in which initial values are set, and when the target value is 0, external output Y50 goes on.

This can be used, for example, in programs such as those that cause a display lamp to light when the work being added or subtracted has reached a certain quantity.

1) When the trailing edge of rest input X2 is detected (on → off), data (target value) in data register DT10 is transferred to DT0.
2) One is subtracted from value of DT0 when the count input R1 turns on while R0 is in the off state. (DOWN counter operation)
   One is added to DT0 when the count input R1 turns off while the UP/DOWN input R0 is in the on state. (Up counter operation)
3) If the counter elapsed value area DT0 = K0, external output Y50 turns on.



Description

The counter is switched between an incremental count (addition) or decremental count (subtraction) by turning the relay specified for up/down input on or off.

When the up/down input is on, the incremental counter (+1) is effective, and when it is off, the decremental counter (−1) is effective. The elapsed value is stored in the area specified by the D.

The preset value in S is transferred to D when the trailing edge of the reset input is detected (on → off). Set value range K-32768 to K32767 (H8000 to H7FFF)

When the count input is switched from off to on (the reset input is in “off” state), the value specified for the D is initialized, and the counting operation begins.

The elapsed value area of D is cleared when the reset input turns on.

The results of the counting operation can be determined by comparing the elapsed value of D with the specified value, using the data comparison instruction.

The data comparison instruction should be executed immediately following execution of **F118 (UDC)** instruction.

Precautions during programming

If the elapsed value area has been specified as a hold type memory area, the elapsed value acts in accordance with the contents being retained.

Be aware that, when an operation is begun, the set values are not automatically preset to the elapsed value area. To preset these values, the reset input must be switched from the “on” to the “off” state.

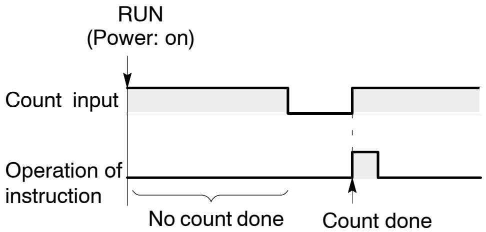
When combining the **F118 (UDC)** instruction with an **AND** stack instruction or **POP** stack instruction, be careful that the programming is correct.

Cautions on count input detection

In a **F118 (UDC)** instruction, the increment or decrement takes place when the rise of the count input from off to on is detected.

If the count input remains continuously on, since counting will only take place at the rise, no further counting will take place.

In cases where the count input is initially on such as when the mode is changed to RUN or the power is turned on with the mode set to RUN, increment or decrement operation will not take place at the first scan.



When you use the **F118 (UDC)** instruction with one of the following instructions that changes the order of the execution of instructions, be aware that the operation of the instructions will differ depending on the timing of their execution and their count input.

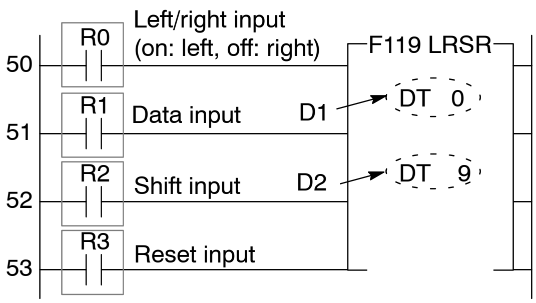
– **MC** to **MCE** instructions
– **JP** to **LBL** instructions
– **F19 (SJP)** to **LBL** instructions
– **LOOP** to **LBL** instructions
– **CNDE** instruction
– Step ladder instructions
– Subroutine instructions

F119 (LRSR) Left/right shift register

Outline Shifts one bit of the 16-bit data range to the left or right.

Program example

Ladder Diagram



Boolean

Address	Instruction		
50	ST	R	0
51	ST	R	1
52	ST	R	2
53	ST	R	3
54	F119		(LRSR)
	DT		0
	DT		9

D1	Starting 16-bit area whose one bit is shifted to the left or to the right
D2	Ending 16-bit area whose one bit is shifted to the left or to the right

Operands

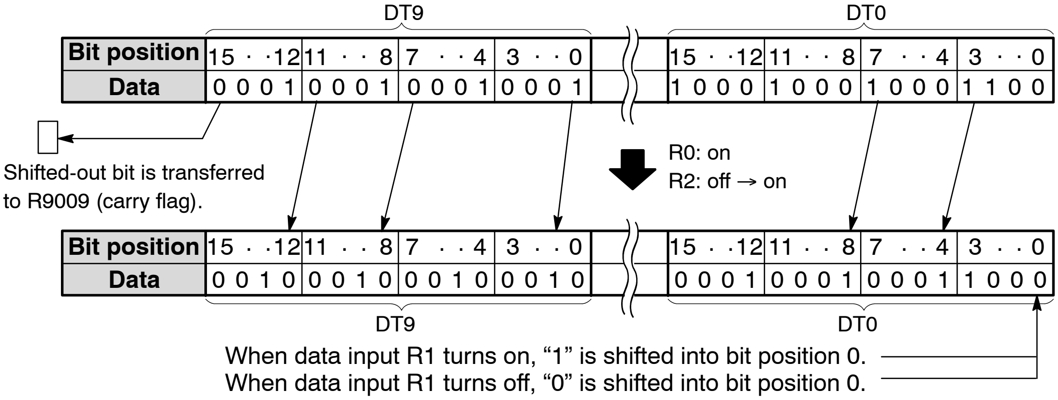
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX	IY	K	H	
D1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A
D2	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

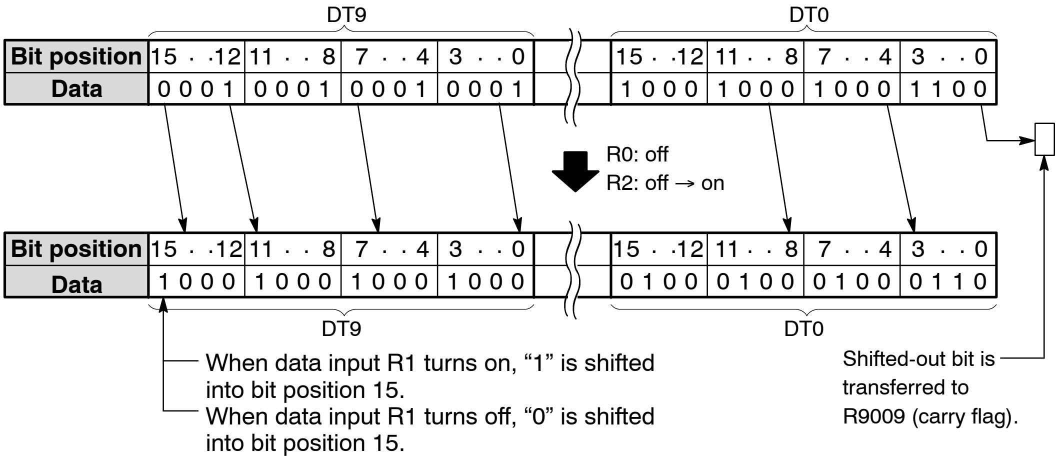
A: Available
N/A: Not Available

Explanation of example

Left shift operation



Right shift operation



Description

- This shift register changes direction, either left (direction of MSB) or right (direction of LSB), in which a shift of one bit is made, based on the on/off status of the relay specified by the left/right shift input.
- The shift operation is made to the left when the left/right shift input is on, and to the right when off.

Specify D1 and D2 so they are in the same type data area and be sure to set the data area addresses so that D1 $\leqq$ D2.

When the shift input changes from off to on (the reset input is off), the contents of the area specified by D1 and D2 are shifted one bit to the left or right.

When the data is shifted, 1 will be set in the empty bit left by the shift (the uppermost or lowermost bit) if the data input is on, and 0 if the data input is off. Also, the bit extracted by the shift (the uppermost bit for a shift to the left, and the lowermost bit for a shift to the right) will be set in the special internal relay R9009 (carry flag).

If the reset input is on, the contents of the specified area are cleared to 0.

Flag conditions

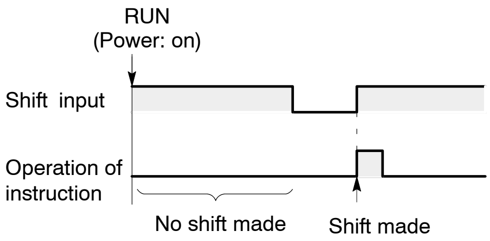
- Error flag (R9007): Turns on and stays on when the area specified using the starting 16-bit area (D1) is larger than the area specified by the ending 16-bit area (D2) (when D1 > D2).
- Error flag (R9008): Turns on for an instant when the area specified using the starting 16-bit area (D1) is larger than the area specified by the ending 16-bit area (D2) (when D1 > D2).
- Carry flag (R9009): Turns on for an instant when the bit shifted-out is “1”.

Cautions on shift input detection

In a **F119 (LRSR)** instruction, shift takes place when the off−on rise of the shift input is detected.

If the shift input remains continuously on, a shift will only take place at the rise. No further shifts will take place.

In cases where the shift input is initially on such as when the mode is changed to RUN or when the power is turned on with the mode set to RUN, a shift will not take place at the first scan.



When you use the **F119 (LRSR)** instruction with one of the following instructions that changes the order of the execution of instructions, be aware that the operation of the instructions will differ depending on the timing of their execution and their shift input.

– **MC** to **MCE** instructions
– **JP** to **LBL** instructions
– **F19 (SJP)** to **LBL** instructions
– **LOOP** to **LBL** instructions
– **CNDE** instruction
– Step ladder instructions
– Subroutine instructions

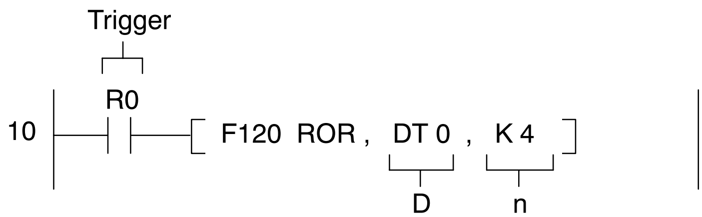
Precautions during programming

When combining the **F119 (LRSR)** instruction with an **AND** stack instruction or **POP** stack instruction, be careful that the programming is correct.

F120 (ROR) P120 (PROR) 16-bit data right rotation

Outline Rotates a specified number of bits in specified 16-bit data to the right. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P120 (PROR)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F120 (ROR) DT 0 K 4

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated

Operands

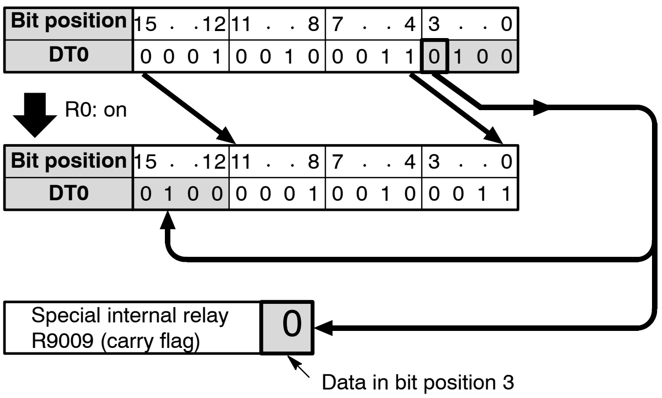
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

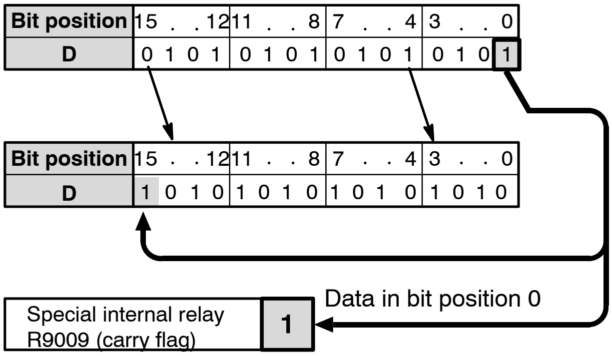
Rotates 4 bits in data register DT0 to the right when trigger R0 turns on.



Description

Rotates “n” bits of the 16-bit data specified by D to the right.

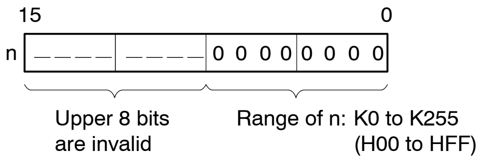
Example: Rotates 1 bit to the right



When “n” bits are rotated to the right,

- The data in bit position n-1 (n th bit starting from bit position 0) is transferred to special internal relay R9009 (carry flag).
- “n” bits starting from bit position 0 are shifted out to the right and then shifted into the higher bit positions of the 16-bit data specified by D.

For “n”, only the lower 8 bits in the 16 bit data are valid.



Precaution during programming

If the specified n is a multiple of 16 bits, the data will be the same as that before the operation.
e.g.,

n = K16: same operation as n = K0 (The carry flag does not change, either.)
n = K17: same operation as n = K1
⋮
n = K32: same operation as n = K0 (The carry flag does not change, either.)
n = K33: same operation as n = K1

Flag conditions

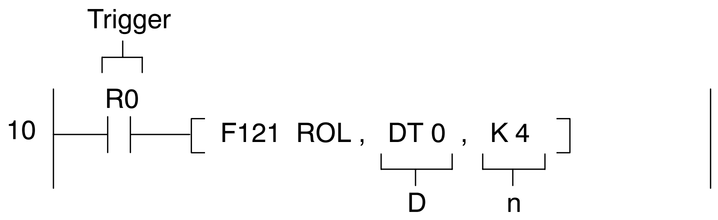
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the data in bit position n-1 is recognized as 1.

F121 (ROL) P121 (PROL) 16-bit data left rotation

Outline

Rotates a specified number of bits in specified 16-bit data to the left.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P121 (PROL)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F121 (ROL) DT 0 K 4

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated

Operands

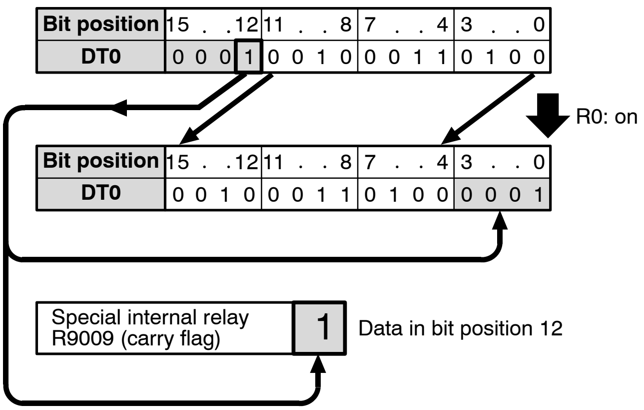
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

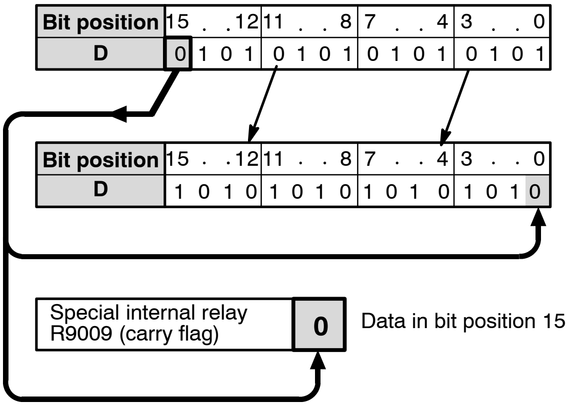
Rotates 4 bits in data register DT0 to the left when trigger R0 turns on.



Description

Rotates “n” bits of the 16-bit data specified by D to the left.

Example: Rotates 1 bit to the left

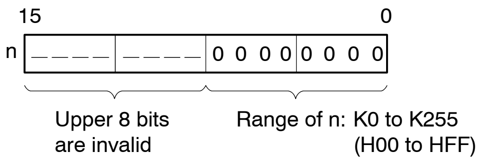


When “n” bits are rotated to the left,

– The data in bit position 16-n (n th bit starting from bit position 15) is transferred to special internal relay R9009 (carry flag).

– “n” bits starting from bit position 15 are shifted out to the left and then shifted into the lower bit positions of the 16-bit data specified by D.

For “n”, only the lower 8 bits in the 16 bit data are valid.



Precaution during programming

If the specified “n” is a multiple of 16 bits, the data will be the same as that before the operation.
e.g.,

n = K16: same operation as n = K0 (The carry flag does not change, either.)
n = K17: same operation as n = K1
⋮ ⋮
n = K32: same operation as n = K0 (The carry flag does not change, either.)
n = K33: same operation as n = K1

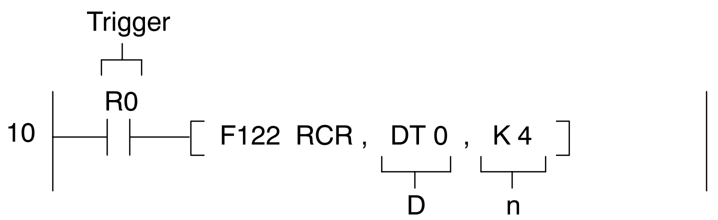
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the data in bit position 16-n is recognized as 1.

F122 (RCR) P122 (PRCR) 16-bit data right rotation with carry flag data

Outline Rotates a specified number of bits in the specified 16-bit data to the right together with carry flag data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P122 (PRCR)" is not available.

Program example



Ladder Diagram	Boolean	
	Address	Instruction
(see diagram above)	10	ST R 0
	11	F122 (RCR)
		DT 0
		K 4

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated

Operands

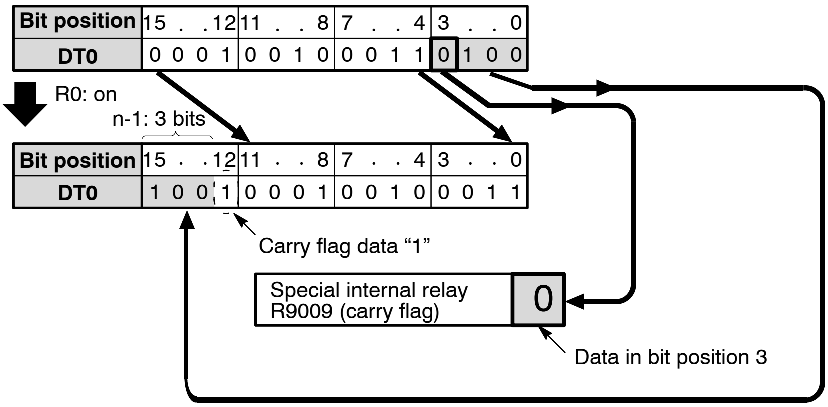
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

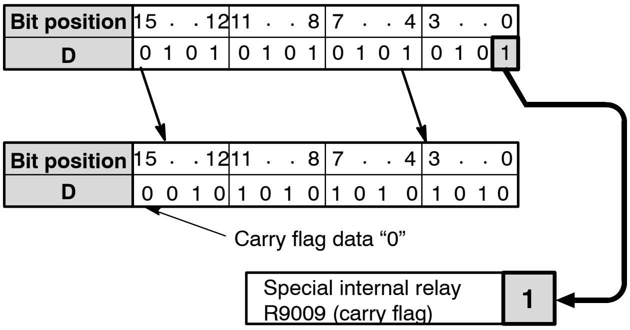
Rotates 4 bits in data register DT0 together with carry flag data "1" to the right when trigger R0 turns on.



Description

Rotates “n” bits of the 16-bit data specified by D, including carry flag data, to the right.

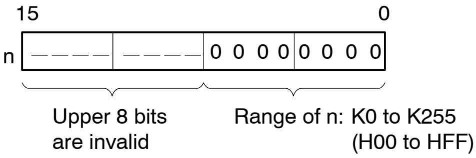
Example: Rotates 1 bit to the right



When “n” bits with carry flag data are rotated to the right,

- The data in bit position n-1 (n th bit starting from bit position 0) is transferred to special internal relay R9009 (carry flag).
- “n” bits starting from bit position 0 are shifted out to the right and then carry flag data and the n-1 bits starting from bit position 0 are shifted into the higher bit positions of the 16-bit data specified by D.

For n, only the lower 8 bits in the 16 bit data are valid.



Precaution during programming

If the specified “n” is a multiple of 17 bits, the data will be the same as that before the operation.

e.g., n = K17: same operation as n = K0

n = K18: same operation as n = K1

⋮ ⋮

n = K34: same operation as n = K0

n = K35: same operation as n = K1

Flag conditions

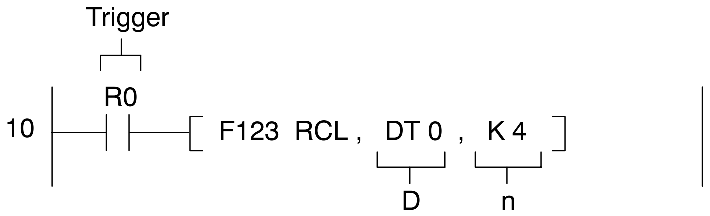
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the data in bit position n-1 is recognized as 1.

F123 (RCL) P123 (PRCL) 16-bit data left rotation with carry flag data

Outline Rotates a specified number of bits in the specified 16-bit data to the left together with carry flag data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P123 (PRCL)” is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	0
11	F123		(RCL)
	DT		0
	K		4

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated

Operands

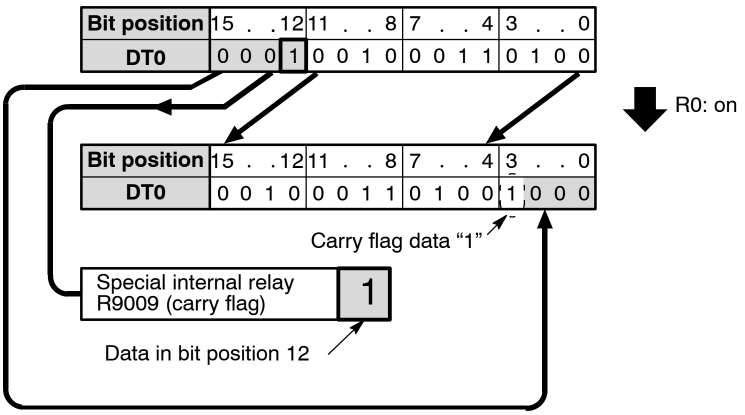
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

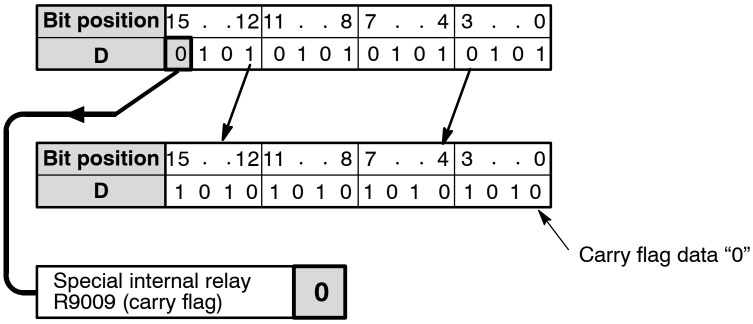
Rotates 4 bits in data register DT0 together with carry flag data “1” to the left when trigger R0 turns on.



Description

Rotates “n” bits of the 16-bit data specified by D, including carry flag data, to the left.

Example: Rotates 1 bit to the left

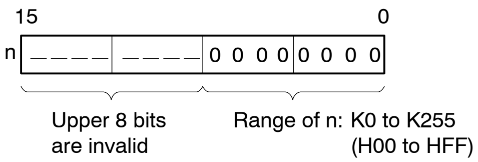


When “n” bits with carry flag data are rotated to the left,

– The data in bit position 16-n (n th bit starting from bit position 15) is transferred to special internal relay R9009 (carry flag).

– “n” bits starting from bit position 15 are shifted out to the left and then carry flag data and the n-1 bits starting from bit position 15 are shifted into the lower bit positions of the 16-bit data specified by D.

For n, only the lower 8 bits in the 16 bit data are valid.



Precaution during programming

If the specified “n” is a multiple of 17 bits, the data will be the same as that before the operation.

e.g., n = K17: same operation as n = K0

n = K18: same operation as n = K1

⋮

n = K34: same operation as n = K0

n = K35: same operation as n = K1

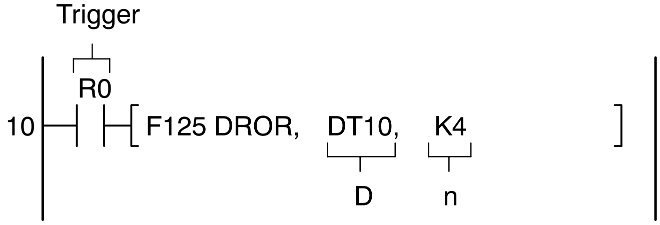
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the data in bit position 16-n is recognized as 1.

F125 (DROR) P125 (PDROR) 32-bit data right rotation

Outline Rotates a specified number of bits in specified 32-bit data to the right. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P125 (PDROR)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F125 (DROR) DT 10 K 4

D	Lower 16-bit area of 32-bit data
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated Range of n: K0 to K255 (H0 to HFF)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

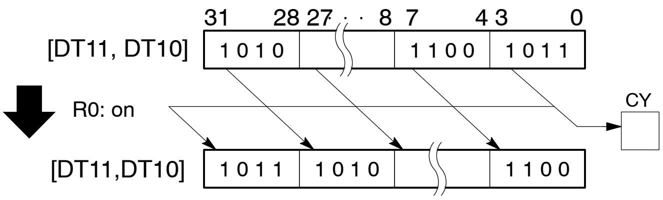
(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

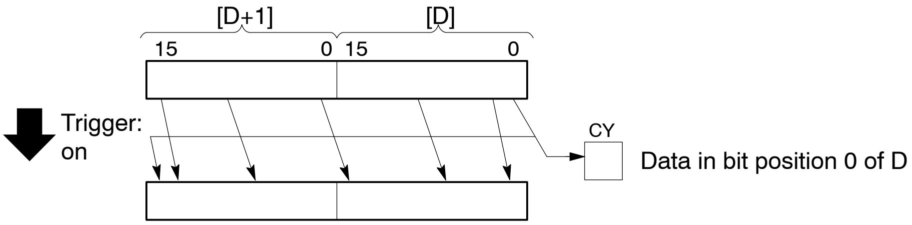
Rotates 4 bits in data registers DT11 and DT10 to the right when trigger R0 turns on.

When 4 bits are rotated to the right, the data in bit position 3 is transferred to special internal relay R9009 (carry flag).



Description

Rotates “n” bits of the 32-bit data specified by D to the right when the trigger turns on.

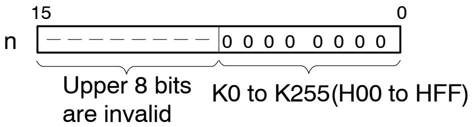


When “n” bits are rotated to the right,

– The data in bit position n-1 (nth bit starting from bit position 0) is transferred to special internal relay R9009 (carry flag).

– “n” bits starting from bit position 0 are shifted out to the right and then shifted into the higher bit positions of the 32-bit data specified by D.

Only the lower eight bits of the 16-bit data “n” are effective.



When “n” is specified using K0, the contents of “D+1, D” and the special internal relay R9009 (carry flag) do not change.

Precautions during programming

If the specified “n” is a multiple of 32 bits, the data will be the same as that before the operation.

e.g., n = K32: same operation as n = K0

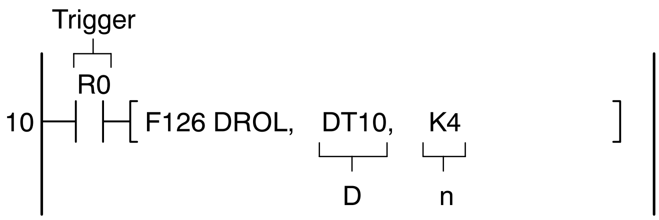
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Content in the nth bit from LSB (least significant bit) turns on for an instant when the data in bit position n-1 is recognized as 1.

F126 (DROL) P126 (PDROL) 32-bit data left rotation

Outline Rotates a specified number of bits in specified 32-bit data to the left. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P126 (PDROL)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F126 (DROL)
		DT 10
		K 4

D	Lower 16-bit area of 32-bit data
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated Range of n: K0 to K255 (H0 to HFF)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

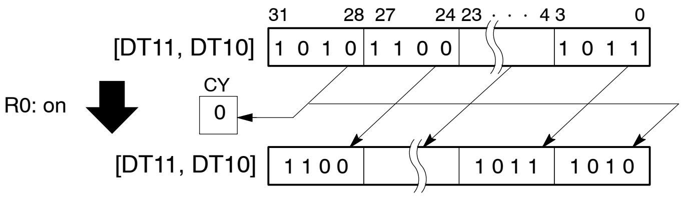
(*1) This cannot be used with the FP0R, FPΣ and FP−X.

A: Available
N/A: Not Available

Explanation of example

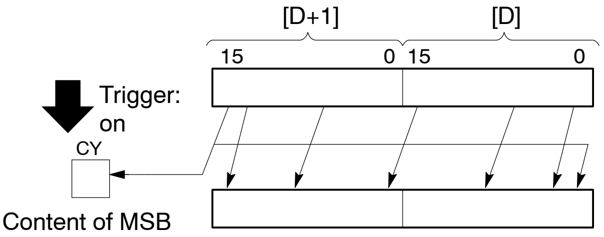
Rotates 4 bits in data registers DT11 and DT10 to the left when trigger R0 turns on.

The data in bit position 28 is transferred to special internal relay R9009 (carry flag).



Description

Rotates “n” bits of the 32-bit data specified by D to the left when the trigger turns on.

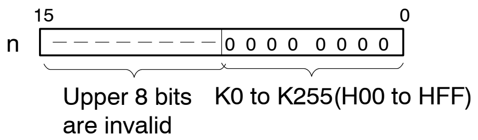


When “n” bits are rotated to the left,

– The data in bit position 32-n (nth bit starting from bit position 31) is transferred to special internal relay R9009 (carry flag).

– “n” bits starting from bit position 31 are shifted out to the left and then shifted into the lower bit positions of the 16-bit data specified by D.

Only the lower eight bits of the 16-bit data “n” are effective.



When “n” is specified using K0, the contents of “D+1, D” and the special internal relay R9009 (carry flag) do not change.

Precautions during programming

If the specified “n” is a multiple of 32 bits, the data will be the same as that before the operation.

e.g., n = K32: same operation as n = K0

Flag conditions

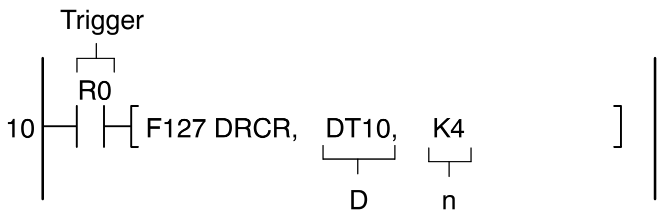
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Content in the nth bit from MBS (most significant bit) turns on for an instant when the data in bit position 32-n is recognized as 1.

F127 (DRCR) P127 (PDRCR) 32-bit data right rotation with carry flag data

Outline Rotates a specified number of bits in the specified 32-bit data to the right together with carry flag data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P127 (PDRCR)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F127		(DRCR)
	DT		10
	K		4

D	Lower 16-bit area of 32-bit data
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated Range of n: K0 to K255 (H0 to HFF)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

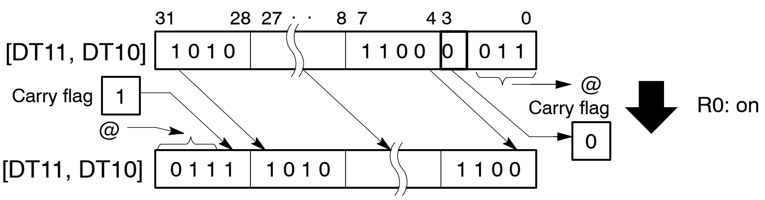
A: Available
N/A: Not Available

Explanation of example

Rotates 4 bits in data registers DT11 and DT10 together with carry flag data to the right when trigger R0 turns on.

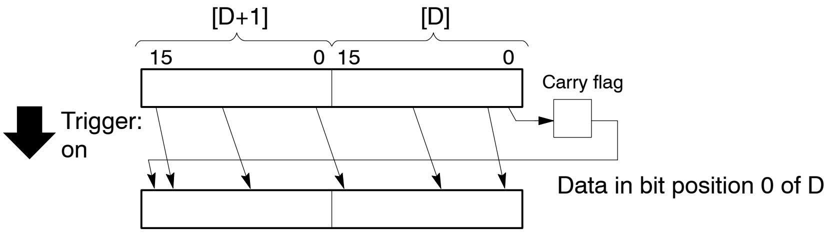
The data in bit position 3 is transferred to the carry flag (special internal relay R9009).

The data of the carry flag (special internal relay R9009) is transferred to the bit position 28.



Description

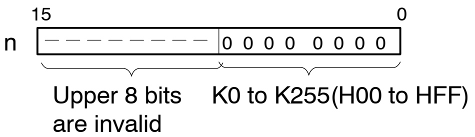
Rotates “n” bits of the 32-bit data specified by D, including carry flag data, to the right when the trigger turns on.



When “n” bits with carry flag data are rotated to the right,

– The data in bit position n-1 (nth bit starting from bit position 0) is transferred to special internal relay R9009 (carry flag).
– “n” bits starting from bit position 0 are shifted out to the right and then carry flag data and the n-1 bits starting from bit position 0 are shifted into the higher bit positions of the 32-bit data specified by D.

Only the lower eight bits of the 16-bit data “n” are effective.



When “n” is specified using K0, the contents of “D+1 and D” and the carry flag do not change.

Precautions during programming

If the specified “n” is a multiple of 33 bits, the data will be the same as that before the operation.

e.g., n = K33: same operation as n = K0

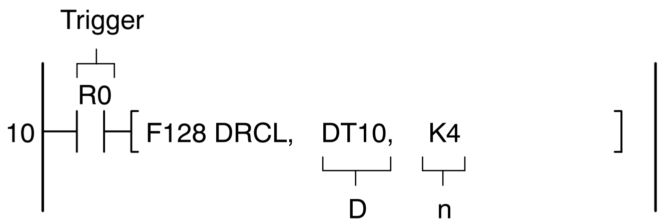
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Content in the nth bit from LSB (least significant bit) turns on for an instant when the data in bit position n-1 (nth bit starting from bit position 0) is recognized as 1.

F128 (DRCL) P128 (PDRCL) 32-bit data left rotation with carry flag data

Outline Rotates a specified number of bits in the specified 32-bit data to the left together with carry flag data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P128 (PDRCL)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F128 (DRCL) DT 10 K 4

D	Lower 16-bit area of 32-bit data
n	16-bit equivalent constant or 16-bit area to specify number of bits to be rotated Range of n: K0 to K255 (H0 to HFF)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	f		
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.

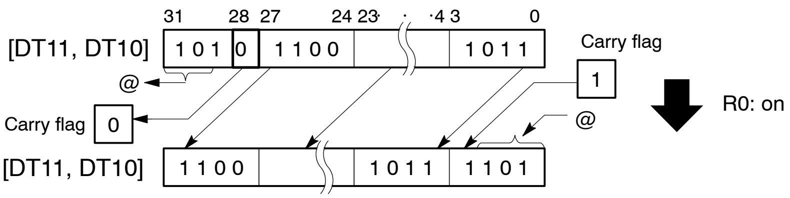
A: Available
N/A: Not Available

Explanation of example

Rotates 4 bits in data registers DT11 and DT10 together with carry flag data to the left when trigger R0 turns on.

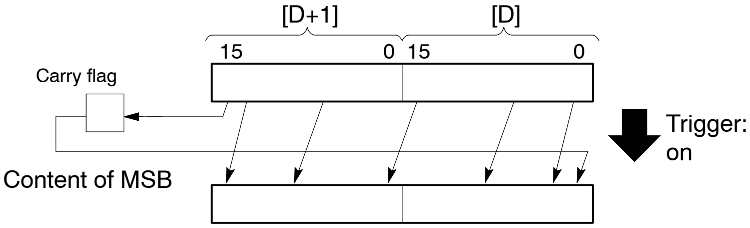
The data in bit position 28 is transferred to carry flag (special internal relay R9009).

The data of the carry flag is transferred to the bit position 3.



Description

Rotates “n” bits of the 32-bit data specified by D, including carry flag data, to the left when the trigger turns on.

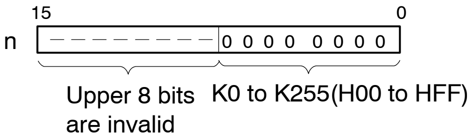


When “n” bits with carry flag data are rotated to the left,

– The data in bit position 32-n (nth bit starting from bit position 31) is transferred to special internal relay R9009 (carry flag).

– “n” bits starting from bit position 31 are shifted out to the left and then carry flag data and the n-1 bits starting from bit position 31 are shifted into the lower bit positions of the 32-bit data specified by D.

Only the lower eight bits of the 16-bit data “n” are effective.



When “n” is specified using K0, the contents of “D+1 and D” and the carry flag do not change.

Precautions during programming

If the specified “n” is a multiple of 33 bits, the data will be the same as that before the operation.

e.g., n = K33: same operation as n = K0

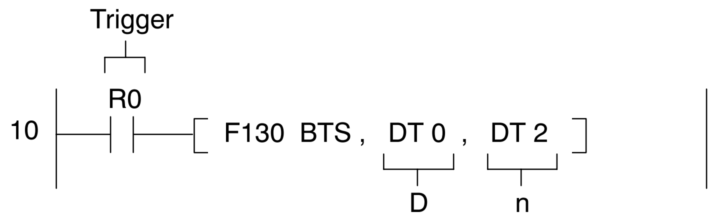
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Content in the nth bit from MSB (most significant bit) turns on for an instant when the data in bit position 31-n is recognized as 1.

F130 (BTS) P130 (PBTS) 16-bit data bit set

Outline Turns on a specified bit of 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P130 (PBTS)" is not available.

Program example



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F130		(BTS)
	DT		0
	DT		2

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify bit position

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Turns on bit position specified by DT2 in data register DT0 when trigger R0 turns on.

When the DT2 = K7, as shown below.

[n] DT2: K7

		15 . . 12	11 . . 8	7 . . 4	3 . . 0
	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
[D]	**DT0**	0 1 0 0	0 0 1 1	0 0 1 0	0 0 0 1

R0: on

	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
[D]	**DT0**	0 1 0 0	0 0 1 1	1 0 1 0	0 0 0 1

Bit position 7 is turned on (1) when R0 turns on.
(Bits other than the specified bit do not change.)

Description

Turns on the bit of 16-bit data specified by D and n.
Bits other than the specified bit do not change.

The “n” is decimal data specifying the bit position to be turned on. Range of “n”: K0 to K15

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
n	— — — —	— — — —	— — — —	

The data in bit positions 4 through 15 are invalid.

Range of “n”: K0 to K15 (H0 to HF)

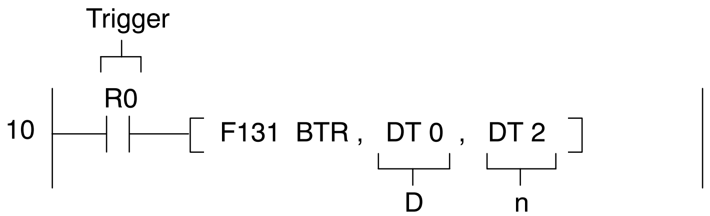
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F131 (BTR) P131 (PBTR) 16-bit data bit reset

Outline Turns off a specified bit of 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P131 (PBTR)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F131 (BTR) DT 0 DT 2

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify bit position

Operands

Operand	Relay WX	Relay WY	Relay WR	Relay WL (*1)	Timer/Counter SV	Timer/Counter EV	Register DT	Register LD (*1)	Register FL (*2)	Index register IX (*3)	Index register IY (*4)	Constant K	Constant H	Index modifier
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Turns off the bit specified by DT2 of data register DT0 when trigger R0 turns on.

When the DT2 = K7, as shown below.

[n] DT2: K7

	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
[D]	DT0	1 1 0 1	1 1 0 0	1 0 1 1	1 0 1 0

R0: on

	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
[D]	DT0	1 1 0 1	1 1 0 0	0 0 1 1	1 0 1 0

Bit position 7 is turned off (0) when R0 turns on.
(Bits other than the specified bit do not change.)

Description

Turns off the bit of 16-bit data specified by D and n.
Bits other than the specified bit do not change.

The “n” is decimal data specifying the bit position to be turned off. Range of “n”: K0 to K15

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
n	————	————	————	

The data in bit positions 4 through 15 are invalid.

Range of “n”: K0 to K15 (H0 to HF)

Flag conditions

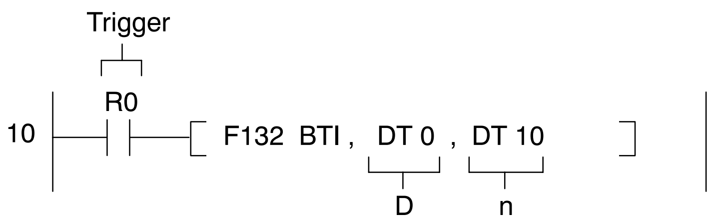
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F132 (BTI) P132 (PBTI) 16-bit data bit invert

Outline

Inverts a specified bit in 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P132 (PBTI)” is are not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F132 (BTI)
		DT 0
		DT 10

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify bit position

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Inverts the state of bit specified by DT10 in data register DT0 when trigger R0 turns on.

When the DT10 = K7, as shown below.

[n] DT10: K7

[D]

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	0 0 0 0	0 0 0 1	0 0 1 1	0 0 1 0

R0: on

[D]

Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
DT0	0 0 0 0	0 0 0 1	1 0 1 1	0 0 1 0

Condition of bit position 7 is inverted [off (0) → on (1)] when R0 turns on.
(Bits other than the specified bit are not changed.)

Description

Inverts [off (0) → on (1) or on (1) → off (1)] the state at bit position specified by “n” in the 16-bit area specified by D.

Bits other than the specified bit are not inverted.

The “n” is decimal data specifying the bit position to be inverted. Range of “n”: K0 to K15

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
n	– – – –	– – – –	– – – –	0 0 0 0

The data in bit positions 4 through 15 are invalid.

Range of “n”: K0 to K15 (H0 to HF)

Flag conditions

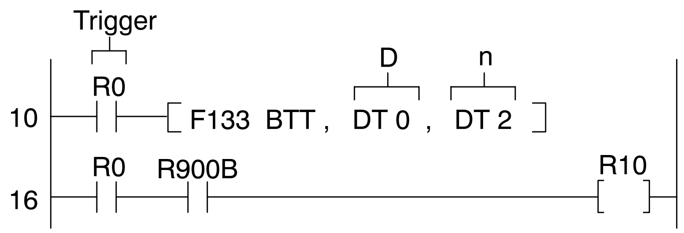
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F133 (BTT) P133 (PBTT) 16-bit data bit test

Outline Checks the state [on (1) or off (0)] of the specified bit in 16-bit data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P133 (PBTT)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F133		(BTT)
	DT		0
	DT		2
16	ST	R	0
17	AN	R	900B
18	OT	R	10

D	16-bit area
n	16-bit equivalent constant or 16-bit area to specify bit position

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Checks the state [on (1) or off (0)] of bit specified by DT2 in data register DT0 when trigger R0 turns on. If bit specified by DT2 is in the off (0) state, internal relay R10 goes on.

When the DT2 = K7, as shown below.

[n] DT2:K7

	Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
[D]	**DT0**	0 1 0 0	0 0 1 1	0 0 1 0	0 0 0 1

The state of bit position 7 is checked.

If bit position 7 is in the off state (0), R900B turns on and internal relay R10 goes on.

Description

Checks the state [on (1) or off (0)] of bit position specified by n in the 16-bit data specified by D. The judgment result is output to special internal relay R900B (=flag).

The specified bit is checked by special internal relay R900B.

– When the specified bit is on (1), special internal relay R900B (= flag) turns off.

– When the specified bit is off (0), special internal relay R900B (= flag) turns on.

The “n” is decimal data specifying the bit position to be checked. Range of “n”: K0 to K15

Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0
n	————	————	————	0 0 0 0

The data in bit positions 4 through 15 are invalid.

Range of “n”: K0 to K15 (H0 to HF)

Precaution when the judgement flag R900B is used two or more times

The judgment flag R900B is updated each time an operation instruction or comparison instruction is executed.

If the judgment flag is used two or more times,

– the program in which the judgment flag is used should be input immediately following the instruction which executes the judgment.

– the flag should be output to output relays or internal relays for each separate instruction.

Flag conditions

• Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.

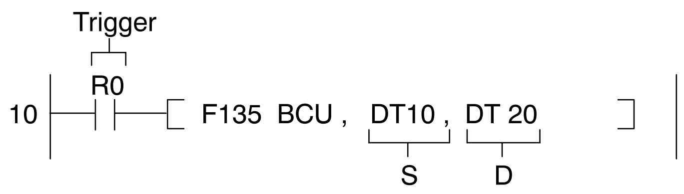
• Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

• = flag (R900B): Turns on for an instant when the specified bit to be checked is in the off (0) state.

F135 (BCU) P135 (PBCU) Number of on (1) bits in 16-bit data

Outline Counts the number of bits in the on (1) state in the specified 16-bit data. For the FP0R/FPΣ/FP–X/FP0/FP–e, the P type high–level instruction “P135 (PBCU)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F135 (BCU) DT 10 DT 20

S	16-bit equivalent constant or 16-bit area (source)
D	16-bit area (destination) for storing the number of bits in the on (1) state

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) This cannot be used with the FP0 and FP–e.
(*2) This cannot be used with the FP0, FP–e, FP0R, FPΣ, FP–X.
(*3) With the FP0R, FPΣ, FP–X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP–X, FP2, FP2SH, and FP10SH, this is ID.

Explanation of example

Counts the number of bits in the on (1) state in data register DT10 when trigger R0 turns on. The number of on (1) bits is stored in data register DT20.

	DT10			
Bit position	15 . . 12	11 . . 8	7 . . 4	3 . . 0
Binary data	0 0 0 0	0 0 0 1	0 0 1 1	0 1 0 1

The number of on (1) bits is “5”.

The K5 is stored in data register DT20 when R0 turns on.

Description

Counts the number of bits in the on (1) state in the 16-bit data specified by S. The counted result (number of on (1) bits) is stored in the 16-bit area specified by D.

The results are stored in decimal number.

Flag conditions

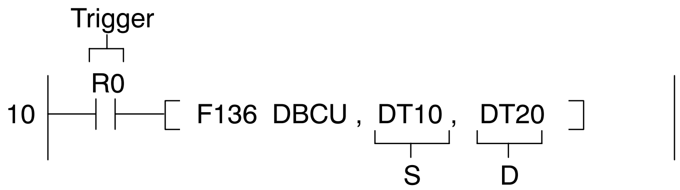
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F136 (DBCU) P136 (PDBCU) Number of on (1) bits in 32-bit data

Outline Counts the number of bits in the on (1) state in specified 32-bit data. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P136 (PDBCU)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F136		(DBCU)
	DT		10
	DT		20

S	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
D	16-bit area (destination) for storing the number of bits in the on (1) state

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Counts the number of bits in the on (1) state in data register DT11 and DT10 when trigger R0 turns on. The number of on (1) bits is stored in data register DT20.

	DT11				DT10			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	0 0 1 1	0 0 1 0	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1	0 0 1 1	0 1 1 0

The number of on (1) bits is “9”.

The K9 is stored in data register DT20 when R0 turns on.

Description

Counts the number of bits in the on (1) state in the 32-bit data specified by S. The counted result (number of on (1) bits) is stored in the 16-bit area specified by D.

The results are stored in decimal number.

Flag conditions

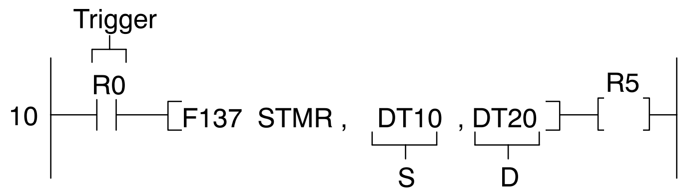
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F137 (STMR) Auxiliary timer (16−bit)

Outline Sets the 16−bit on−delay timer for 0.01 s units (0.01 to 327.67 s)

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F137		(STMR)
	DT		10
	DT		20
16	OT	R	5

S	16-bit equivalent constant or 16-bit area for timer set value
D	16-bit area for timer elapsed value

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY (*4)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.
(*4) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

When the execution condition (trigger) has been fulfilled, the auxiliary timer is activated, and when the value stored in DT10 x 0.01 seconds has elapsed, R5 goes on.

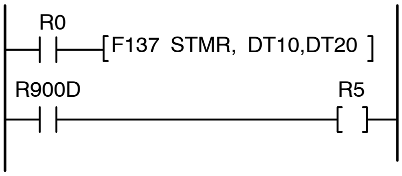
Description

This functions as a 0.01−second unit on delay timer. When the execution condition (trigger) is on, subtraction is carried out for the specified time, and when the elapsed value D reaches 0, the special internal relay R900D turns on. (The special internal relay R900D turns off when the execution condition (trigger) is off, and while subtraction is being carried out.)

With FP2/FP2SH/FP10SH, the **OT** instruction can be stated immediately following the auxiliary timer. When the execution condition (trigger) goes on, the set time is subtracted, and when the elapsed value D reaches 0, the relay being used with the **OT** instruction goes on at the same time that the special internal relay R900D goes on.

When the execution condition (trigger) is off, the elapsed value area is cleared to 0, and relays being used are turned off by the **OT** instruction.

When the time set for the special internal relay R900D has elapsed, the relay is turned on.
R900D can also be used as a timer contact. (The relay is off when the execution condition (trigger) is off, and while subtraction is being carried out.)



Operation is the same as that in the program example.

Timer set time

The timer setting is entered as a value of 0.01 x (timer set value).

The timer set value is specified as a K constant within the range of K1 to K32767.

STMR is set between 0.01 and 327.67 seconds, in units of 0.01 seconds.

If the set value is K500, the set time will be 0.01 x 500 = 5 seconds.

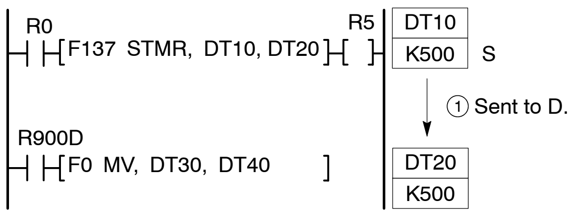
Precautions during programming

The area in which the set value is stored must be set so that the area specified for the elapsed value does not overlap any areas reserved for other timer or counter instructions, or memory areas used for high−level instruction operations.

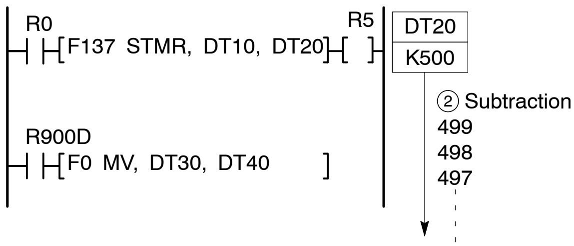
Because subtraction is carried out when operations are carried out, the program should be set up so that operations are carried out every scan.
(In cases such as programs where interrupt operation is carried out, or for jump or loop instructions, where several operations are carried out during one scan, or where it was not possible to carry out any operation during the scan, correct results cannot be obtained.)

How the Auxiliary Timer Works

① When the execution condition (trigger) R0 changes from off to on, the set value specified by the S is sent to the elapsed value area D.

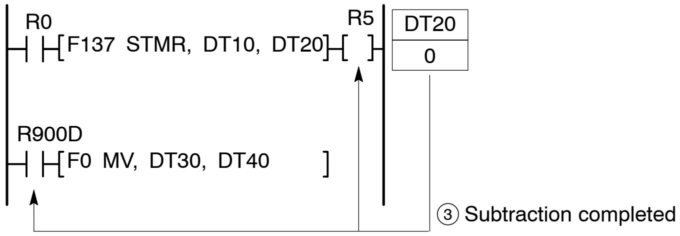


② If the execution condition (trigger) R0 stays on, every scan, the value in the elapsed value area D is subtracted.



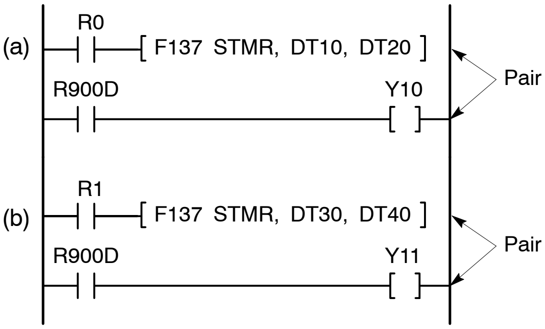
☛ next page

③ If the value in the elapsed value area D reaches 0, relay being used is turned on by the **OT** instruction which comes next in the program. The special internal relay R900D also goes on at this point.



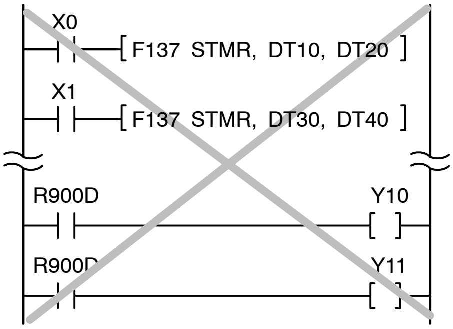
Precautions When Using R900D

If R900D is used and multiple auxiliary timers are being used, always use R900D in the line following the auxiliary timer instruction.



When timer (a), which is activated by R0 turns on, expires, Y10 goes on. When timer (b), which is activated by R1 turns on, expires, Y11 goes on.

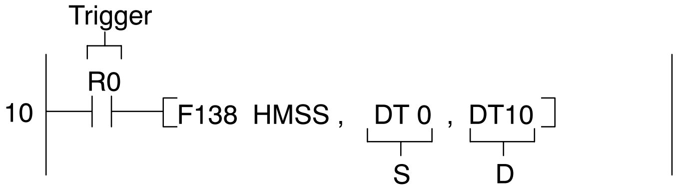
Describe the program as shown below will result in incorrect operation.



F138 (HMSS) P138 (PHMSS) Hours, minutes, and seconds data to seconds data

Outline Converts hour, minute, and second data to seconds data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P138 (PHMSS)” is not available.

Program example



Ladder Diagram	Boolean	
	Address	Instruction
(see diagram above)	10	ST R 0
	11	F138 (HMSS)
		DT 0
		DT 10

S	Starting 16-bit area for storing hours, minutes, and seconds data (source)
D	Starting 16-bit area for storing converted seconds data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Converts the hour, minute, and second data stored in data registers DT1 and DT0 to seconds data when trigger R0 turns on. The converted seconds data is stored in data registers DT11 and DT10.

7: 45’ 30” [H00074530 (BCD) (DT1 = H7, DT0 = H4530)]

DT1				DT0			
0	0	0	7	4	5	3	0

Hours data (DT1) — Minutes data (DT0 upper two digits) — Seconds data (DT0 lower two digits)

⬇ **F138 (HMSS)** instruction execution

27930” [H00027930 (BCD) (DT11 = H2, DT10 = H7930)]

DT11				DT10			
0	0	0	2	7	9	3	0

Seconds data

Description

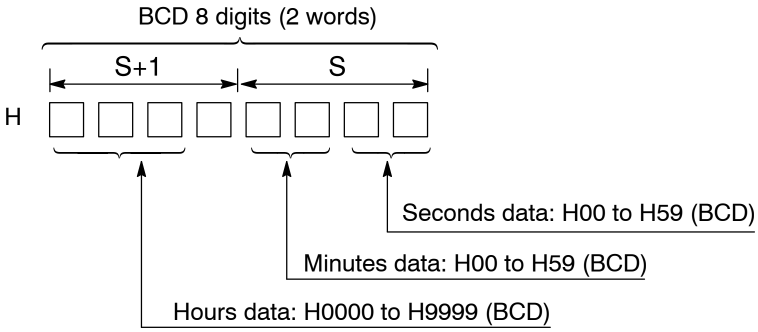
Converts the hour, minute, and second data stored in the 32-bit area specified by S to seconds data. The converted seconds data is stored in the 32-bit area specified by D.

Composition of data

Format of S+1 and S

32 bits (2 words) “S+1 and S” are allocated to express hour, minute, and second data. The data is expressed in BCD format.

The BCD H data should be used for setting the hour (4 digits), minute (2 digits), and second (2 digits) data as follows. (The max. time data that can be specified is 9,999 hours, 59 minutes and 59 seconds.)

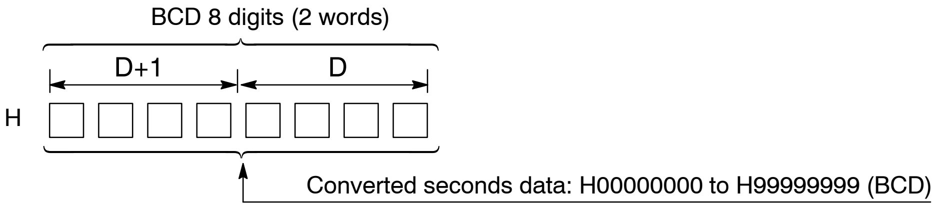


Example: 3:45’19” (S+1: H0003, S: H4519)

Format of D+1 and D

32 bits (2 words) are allocated to express the converted seconds data.

The converted seconds data is expressed in BCD format as follows:



Example: 35,999,999” (D+1: 3599, D: H9999)

Note

The maximum time data that can be specified is 9,999 hours, 59 minutes and 59 seconds, so the maximum value of the time data

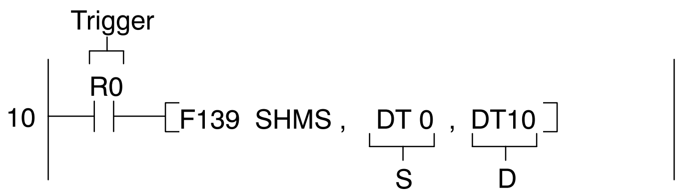
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data specified by S is not BCD data.
  - The minutes and seconds data specified by S exceeds the set range (00 to 59).

F139 (SHMS) P139 (PSHMS) Seconds data to hours, minutes, and seconds data

Outline Converts seconds data to hour, minute, and second data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P139 (PSHMS)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F139 (SHMS) DT 0 DT 10

S	Starting 16-bit area for storing seconds data (source)
D	Starting 16-bit area for storing converted hours, minutes, and seconds data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	IX (*3)	IY	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0 and FP−e.
(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*3) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Converts the seconds data stored in data registers DT1 and DT0 to hour, minute, and second data when trigger R0 turns on. The converted hour, minute, and second data is stored in data registers DT11 and DT10.

4000” [H00004000 (BCD) (DT1 = H0, DT0 = H4000)]

DT1				DT0			
0	0	0	0	4	0	0	0

Seconds data

F139 (SHMS) instruction execution

1: 6’ 40” [H00010640 (BCD) (DT11 = H0001, DT10 = H0640)]

DT11				DT10			
0	0	0	1	0	6	4	0

Hours data (DT11) — Minutes data (upper two digits of DT10) — Seconds data (lower two digits of DT10)

Description

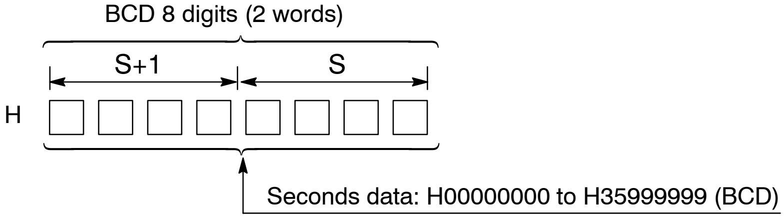
Converts the seconds data stored in the 32-bit area specified by S to hour, minute, and second data. The converted hour, minute, and second data is stored in the 32-bit area specified by D.

Composition of data

Format of S+1 and S

32 bits (2 words) “S+1 and S” are allocated to express the seconds data. The data is expressed in BCD format.

The BCD H data (8 digits) should be used for setting seconds data as follows:



Example: 35,999,999” (S+1: H3599, S: H9999)

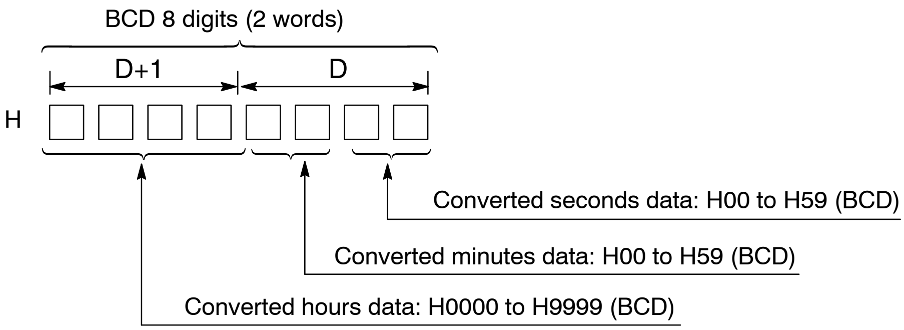
Note

The maximum value that can be stored in D is 9,999 hours, 59 minutes and 59 seconds, so the maximum value that can be specified for the time data for the seconds unit is 35,999,999 seconds.

Format of D+1 and D

32 bits (2 words) “D+1 and D” are allocated to express the converted hours, minutes and seconds data.

The converted hours (4 digits), minutes (2 digits) and seconds (2 digits) data is expressed in BCD format as follows:



Example: 3:45’19” (D+1: H0003, D: H4519)

Flag conditions

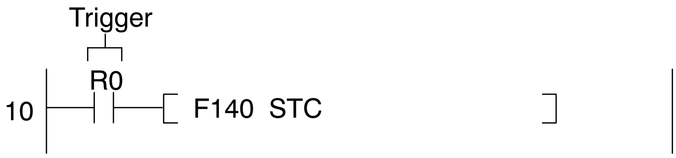
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data specified by S is not BCD data.
  - The data specified by S exceeds the set range (35,999,999).

F140 (STC)
P140 (PSTC)

Carry flag (R9009) set

Outline Turns on special internal relay R9009 (carry flag).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P140 (PSTC)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F140 (STC)

Description

Special internal relay R9009 (carry flag) goes on.

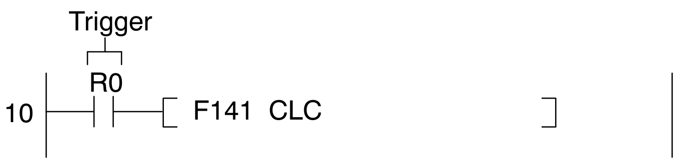
Flag condition

Carry flag (R9009): Turns on when this instruction is executed.

F141 (CLC)
P141 (PCLC) Carry flag (R9009) reset

Outline Turns off special internal relay R9009 (carry flag).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P141 (PCLC)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F141 (CLC)

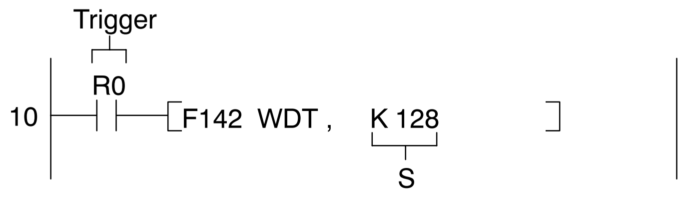
Description
Special internal relay R9009 (carry flag) goes off.

Flag condition
Carry flag (R9009): Turns off when this instruction is executed.

F142 (WDT) P142 (PWDT) Watching dog timer update

Outline Updates the time−out time of watching dog timer.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F142 (WDT) K 128

S	Constant for specifying the watching dog timer value

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Explanation of example

The watching dog timer value is changed to K128 (12.8 ms) when R0 turns on.

Description

This is a set value specified by S, and presets the time−out time for the operation delay watching dog timer.

Operation processing blocks which occur after a preset specified by this instruction will be monitored at the time−out time specified here.

S can be specified within the ranges given below.
K4 to K6400
The time−out time is S × 0.1(ms)

Example: If S is K100, the time−out time is 10ms

The time−out time of operation delay watching dog timer is updated at the start of each scan by referring to system register 30.

If you need to change the time−out time of watching dog timer for all scans, change the value in system register 30.

Using the **F142 (WDT)/P142 (PWDT)** instruction, you can change the time−out time (watching dog timer value) only for that scan.

Precautions during programming

The **F142 (WDT)** instruction may be used any number of times.

To change the time−out time through operation, use the process described below.

1) Execute the **F142 (WDT)** instruction immediately prior to the block to be processed, and specify the preset.
2) When the processing has been completed, execute the **F142 (WDT)** instruction again, and enter a preset with a new value.

If the time required for one scan exceeds 640 ms, the system watching dog timer will be activated, regardless of the time−out time set with the **F142 (WDT)** instruction, and operation will be interrupted and output turned off.

If you want to reset an erroneous condition caused by the system watching dog timer, clear that condition using one of the following methods:
− Using programming tool software.
− Turning the TEST/INITIALIZE switch to the INITIALIZE side.

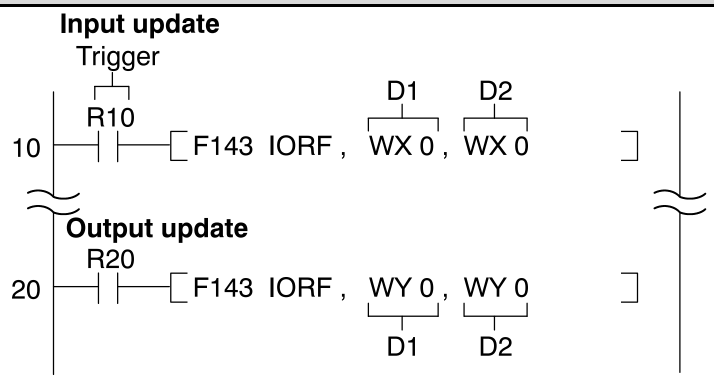
Availability
FP0/FP0R/FP−e/ FPΣ/FP−X

F143 (IORF) Partial I/O update

Outline Updates specified partial I/O points.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F143		(IORF)
	WX		0
	WX		0
	⋮		
20	ST	R	20
21	F143		(IORF)
	WY		0
	WY		0

D1	Starting word address
D2	Ending word address

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
D1	A	A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A
D2	A	A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

Explanation of example

Updates external input relays WX0 (X0 to XF) immediately when the execution condition (trigger) R10 turns on and updates external output relays WY0 (Y0 to YF) immediately when the execution condition (trigger) R20 turns on.

Description

Updates the external inputs X and external outputs Y specified by D1 and D2 immediately even in the program execution stage.

Refreshing (Updating) initiated by the **F143 (IORF)** instruction is done only for the control unit.

With input refreshing (updating), WX0 should be specified for [D1] and [D2].

With output refreshing (updating), WY0 should be specified for [D1] and [D2].

The allowable I/O range for the partial update varies depending on the models.

Availability of the partial I/O update for various models

	Control unit	FP0 Expansion	FPΣ Expansion	Add−on cassette	FP−X Expansion	FP0 Adapter
FP0	A	N/A	—	—	—	—
FP0R	A	A*	—	—	—	—
FP−e	A	—	—	—	—	—
FPΣ 12k	A	N/A	A	—	—	—
FPΣ 32k	A	A*)	A	—	—	—
FP−X	A	—	—	A	N/A	N/A

A: Available, N/A: Not Available

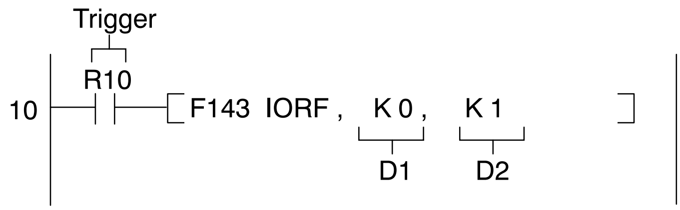
*) For FPΣ 32k type and FP0R, partial I/O update is possible with FP0 expansion units, however, it takes approx. 1 ms for 1 unit.

F143 (IORF) P143 (PIORF) Partial I/O update

Availability
FP2/FP2SH/FP10SH

Outline Updates specified partial I/O points.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 10 F143 (IORF) K 0 K 1

D1	Starting word address
D2	Ending word address

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
D1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
D2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A

A: Available
N/A: Not Available

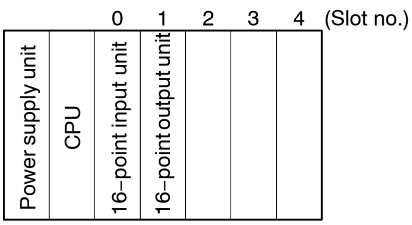
Explanation of example

Updates the input and output relay of word no. 0 to 1 (2 words, 32 points) immediately when the trigger R10 turns on.

When the configuration shown below is being used:

When the instruction is executed, the WX0 (X0 to XF) input processing and the WY1 (Y10 to Y1F) output processing are carried out.

Outline Updates specified partial I/O points.



Description

Updates the input and output relays (X and Y) specified by D1 and D2 immediately even in the program execution stage.
Refreshing (updating) initiated by the **F143 (IORF)** instruction is done only for the unit on the master and expansion backplanes. No update is performed for the input/output relay of the MEWNET-F (remote I/O) system slave station.

How to specify D1 and D2:

– Set the starting address D1 and the ending address D2 (D1 $\leqq$ D2).

– Specify the word address with K0 $\leqq$ D1 $\leqq$ D2 $\leqq$ K255.

– Set the same word address in both D1 and D2 to update only one word.

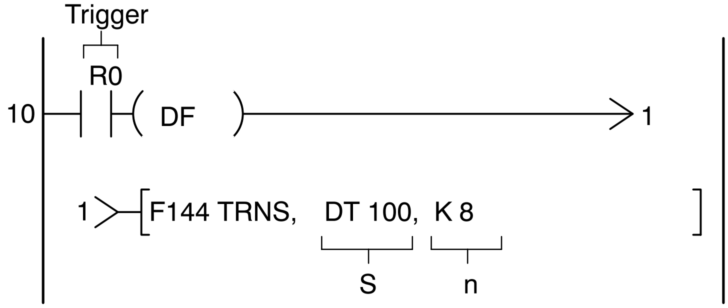
Availability
FP0/FP−e

F144 (TRNS) Serial data communication

Outline Communicates with an external device using the RS232C port.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	DF
12	F144 (TRNS)
	DT 100
	K 8

S	Starting 16-bit area for storing data to be sent
n	16-bit equivalent constant or 16-bit area to specify number of bytes to be sent – When the value is positive, an end code is added. – When the value is negative, an end code is not added. – When the value is H8000, the transmission mode of the RS232C port is changed.

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A

A: Available
N/A: Not Available

Description
Use this instruction for communication (transmission and reception) of command and data when an external device (personal computer, measuring instrument, bar code reader, etc.) is connected to the RS232C port.

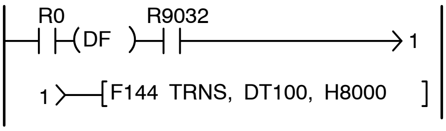
Transmission
The "n" bytes of the data stored in the data table with the starting area specified by S are transmitted from the RS232C port to an external device by serial transmission.
A start code and end code can be automatically added before transmission.

Reception
Reception is controlled by the reception completed flag (R9038) being turned on and off.

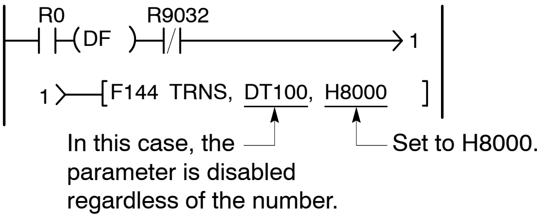
When the reception completed flag (R9038) is off, the data sent to the RS232C port is stored in the reception buffer specified in system registers 417 and 418.
When an **F144 (TRNS)** instruction is executed, the reception completed flag (R9038) goes off.

Switching the use of RS232C port
To switch between "computer link communication" and "serial data communication" (general purpose port), execute an **F144 (TRNS)** instruction.
Set "n" (the number of transmission bytes) to H8000, and then execute the instruction.

When executed when "general purpose port" is selected, the setting will change to "computer link."



When executed when "computer link" is selected, the setting will change to "general purpose port."



R9032: COM. (RS232C) port selection flag
This flag turns on when "General purpose port" is selected.

Note
When the power is turned on, the port use will revert to the setting of system register 412.

Flag conditions
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – The number of bytes specified by "n" exceeds the source data area range.

Program and operation during transmission

To transmit, write the transmission data to the data table, select it with an **F144 (TRNS)** instruction, and execute.

Data table for transmission

Data register areas beginning with the area selected by S are used as the data table for transmission.

[S]		
[S+1]	②	①
[S+2]	④	③
...	...	...
[S+n]	2n	2n−1

The number of bytes not yet transmitted is stored here.

Storage area for transmission data (the circled numbers indicate the order of transmission).

Write the transmission data to the transmission data storage area selected with S (from the second word on) using an **F0 (MV)** or **F95 (ASC)** instruction.

- Do not include an end code in the transmission data as it will be added automatically.
- If the start code is set to “Yes”, do not include a start code in the transmission data as it will be added automatically.
- There is no restriction on the number of bytes [n] that can be transmitted. Following the initial area of the data [S], transmission is possible up to the data range that can be used by the data register. However, if the FP0R is used as the FP0 (FP0 compatibility mode), the maximum number is 2048 bytes.

When the **F144 (TRNS)** instruction is executed, the number of data bytes not yet transmitted is stored in the starting area of the data table. *1

Note

Take care that the transmission data table and reception buffer areas (set in system registers 417 and 418) do not overlap.

Example:

Transmitting the eight characters A, B, C , D, E, F, G and H (8 bytes of data)
In this example, the data table is DT100 to DT104.

DT100	K8	
DT101	H42(B)	H41(A)
DT102	H44(D)	H43(C)
DT103	H46(F)	H45(E)
DT104	H48(H)	H47(G)

The number of bytes not yet transmitted is stored at each transmission. *1

Data is transmitted in order from the lower byte.

Data table before transmission

Program

Select the starting address of the transmission data table with S and the number of transmission data bytes with “n”.

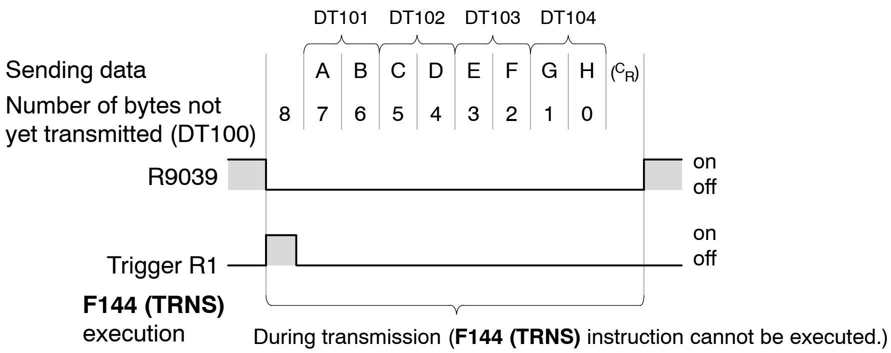
```
 R0
─┤ ├─[ F1 DMV, H44434241, DT101 ]   Write the transmission
     [ F1 DMV, H48474645, DT103 ]   data to the data table
 R1
─┤ ├─(DF )──────────────────> 1

1>─[ F144 TRNS, DT100, K 8    ]     Transmit the data in
                                    the data table
```

Operation

If the execution condition (trigger) for the **F144 (TRNS)** instruction is on when sending completed flag (R9039) goes on, operation will proceed as follows:

1) The “n” is preset in S (the number of bytes not yet transmitted). Furthermore, reception completed flag (R9038) is turned off and the reception data number is cleared to zero.
2) The data in the data table is transmitted in order from the lower byte in S+1.
   – As each byte is transmitted, the value in S (the number of bytes not yet transmitted) decrements by 1. *1
   – During transmission, the sending completed flag (R9039) goes off.
   – If the start code STX is set to “Yes” using system register 413, the start code will be automatically added to the beginning of the data.
   – The end code selected is automatically added to the end of the data.



3) When the specified quantity of data has been transmitted, the value in S (the number of bytes not yet transmitted) will be zero and the sending completed flag (R9039) will go on.

*1 When the FP0R is used as the FP0 (FP0 compatibility mode), the number of transmitted bytes will be set when the transmission starts, and it will not decrease until the completion. It will be cleared to 0 when the transmission completes.

Transmission without an end code

Specify the number of transmission bytes with a negative number.
Set the end code to “Note” for transmission and reception.

Example:

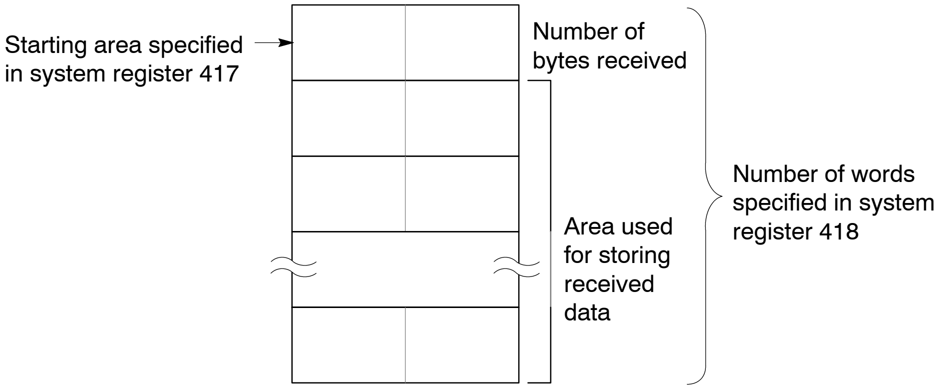
Program for transmitting 8 bytes of data without an end code

```
 R0
─┤ ├─(DF )──────────────────> 1

1>──[F144 TRNS, DT100, K−8    ]
                        ↑
                        └─ Set to K−8.
```

Setting the reception buffer: System register 417 and 418

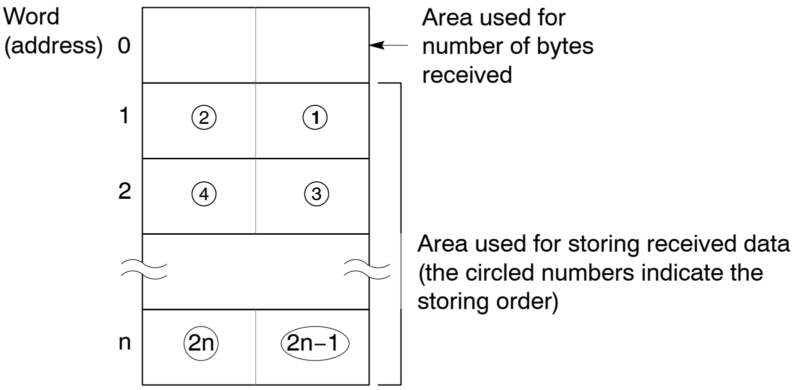
All areas of the data register are initially set for use as the reception buffer. To change the reception buffer, set the starting area number in system register 417 and the size (number of words, Max. 1,024 words) in system register 418.
The reception buffer will be as follows:



Program and operation during reception

Data sent from the external device connected to the RS232C port will be stored in the data register areas set as the reception buffer.

Reception buffer



Each time data is received, the amount of data received (number of bytes) is stored as a count in the leading address of the reception buffer. The initial value is zero.

The data received is stored in order in the reception data storage area beginning from the lower byte of the second word of the area.

Example:

Reception of the eight characters A, B, C, D, E, F, G, and H (8 bytes of data) from an external device

The reception buffer is DT200 to DT204 in this example.
System register settings are as follows:
– System register 417: K200
– System register 418: K5

DT200	K8	
DT201	H42 (B)	H41 (A)
DT202	H44 (D)	H43 (C)
DT203	H46 (F)	H45 (E)
DT204	H48 (H)	H47 (G)

Each time data is received, the number of bytes received is stored.

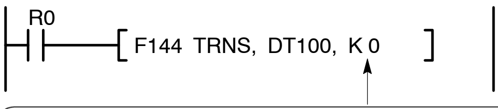
The reception data is stored in order from the lower byte.

Reception buffer when reception is completed

Program

When reception of data from an external device has been completed, the reception completed flag (R9038) goes on and further reception of data is not allowed.

To receive more data, an **F144 (TRNS)** instruction must be executed to turn off the reception completed flag (R9038) and clear the byte number to zero.



To repeat reception only, set to K0.

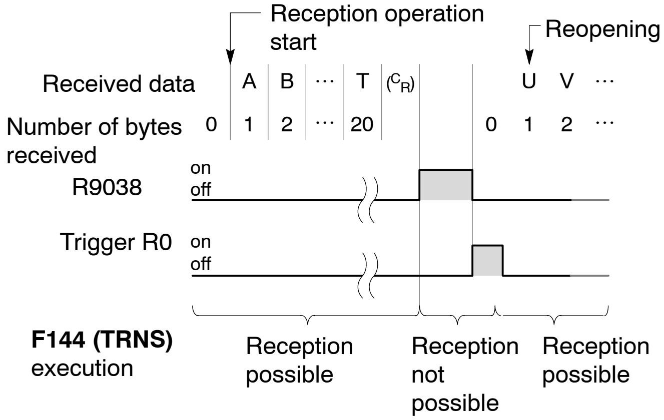
R9038 will also go off when the number of transmission bytes is set and transmission is carried out.

Operation

When the reception completed flag (9038) is off and data is sent from an external device, operation will proceed as follows. (After RUN, R9038 is off during the first scan.)

1) The data received is stored in order in the reception data storage area of reception buffer beginning from the lower byte of the second word of the area.

   Start and end codes will not be stored.
   With each one byte received, the value in the leading address of the reception buffer is incremented by 1.



2) When an end code is received, the reception completed flag (R9038) goes on. After this, no further reception of data is allowed.
3) When an **F144 (TRNS)** instruction is executed, the reception completed flag (R9038) goes off and the number of received data bytes is cleared to zero. Further data received is stored in order in the reception data storage area beginning from the lower byte of the second word of the area.

Note

For repeated reception of data, refer to the following procedure 1) to 5).
1) Receive data
2) Reception completed (R9038: on, Reception: not allowed)
3) Process received data
4) Execute F144 (TRNS) instruction (R9038: off, Reception: enable)
5) Receive further data

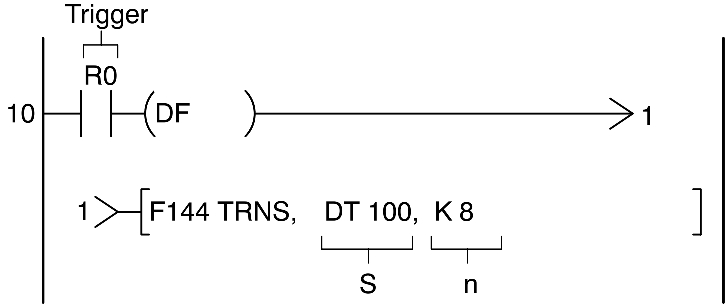
Availability
FP2/FP2SH/FP10SH

F144 (TRNS) Serial data communication

Outline Communicates with an external device using the COM. port of CPU.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	DF		
12	F144		(TRNS)
	DT		100
	K		8

S	Starting 16-bit area for storing data to be sent
n	16-bit equivalent constant or 16-bit area to specify number of bytes to be sent – When the value is positive, an end code is added. – When the value is negative, an end code is not added. – When the value is H8000, the transmission mode of the RS232C port is changed.

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A

A: Available
N/A: Not Available

Explanation of example

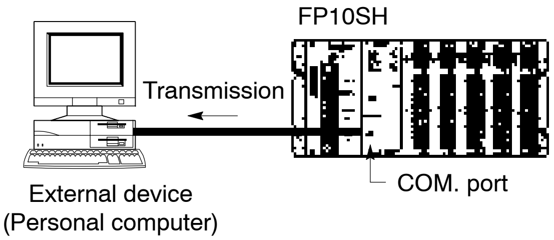
When the trigger R0 turns on, 8 bytes of the data stored in data registers DT101 through DT104 are transmitted from the COM. port.

Description

Use this instruction for communication (transmission and reception) of command and data when an external device (personal computer, measuring instrument, bar code reader, etc.) is connected to the COM. port of CPU.

Transmission

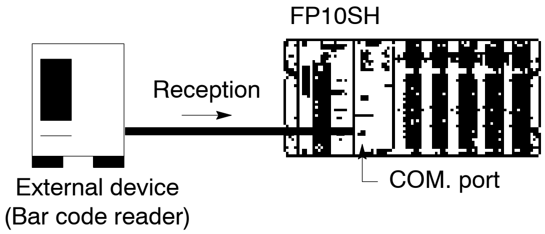
The “n” bytes of the data stored in the data table with the starting area specified by S are transmitted from the COM. port to an external device by serial transmission.
A start code and end code can be automatically added before transmission.



Reception

Reception is controlled by the reception completed flag (R9038) being turned on and off.

When reception completed flag (R9038) is off, the data sent to the COM. port stored in the reception buffer selected in system registers 417 and 418. When an **F144 (TRNS)** instruction is executed, reception completed flag (R9038) goes off.



Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The number of bytes specified by “n” exceeds the source data area range.

Preparation of transmission

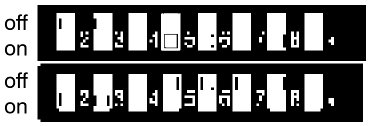
1) Setting the transmission format

For FP10SH

The initial settings for the transmission format are as follows:
– Data length: 8 bits
– Parity check: Yes, odd
– Stop bits: 1 bit
– End code: C_R
– Start code: No STX

To change the transmission format to match the external device connected to the COM. port, set the parameters with the upper row of operation mode switches.

Operation mode switches (using upper dip switch)



Upper dip switches

Lower dip switches

Functions		Settings							
		SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
MODEM control	Disabled	**off**							
	Enabled	on							
Start code	STX (H02) invalid		**off**						
	STX (H02) valid		on						
End code	None			off	off				
	C_R (H0D) and LF (H0A)			on	off				
	C_R (H0D)			**off**	**on**				
	EXT (H03)			on	on				
Stop bit	2 bits					off			
	1 bit					**on**			
Parity check	Invalid						off	off	
	Even parity						on	off	
	Odd parity						**on**	**on**	
Data length (character bit)	7 bits								off
	8 bits								**on**

For FP2/FP2SH

Set the transmission format parameter so that the ”Transmission Format Setting” of system register 413 matches the external device connected to the COM. port. The default setting is the same as that of the FP10SH.

The selected end code is automatically added during transmission. To disable the end code, set the number of transmission bytes using a negative value before the **F144 (TRNS)** instruction.
If the start code is set to “STX valid”, STX will be automatically added.

2) Setting the baud rate

For FP10SH

The baud rate (transmission speed) for serial transmission is initially set to 9600 bps.

To change the baud rate to match the external device connected to the COM. port, set the lower row of operation mode switches as shown below.

Operation mode switches (using lower dip switch)

Functions		Settings							
		SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
Transmission speed	115,200 bps	—	—	—	—	—	off	off	off
	57,600 bps	—	—	—	—	—	on	off	off
	38,400 bps	—	—	—	—	—	off	on	off
	19,200 bps	—	—	—	—	—	on	on	off
	9,600 bps	—	—	—	—	—	**off**	**off**	**on**
	4,800 bps	—	—	—	—	—	on	off	on
	2,400 bps	—	—	—	—	—	off	on	on
	1,200 bps	—	—	—	—	—	on	on	on

For FP2/FP2SH

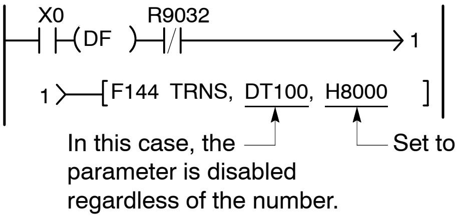
Set the transmission speed so that the “COM Port Baud Rate Setting” of system register 414 matches the external device connected to the COM. port. The default setting is “19200 bps”.

3) Setting the use of the COM. port

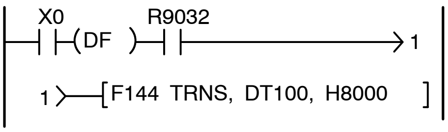
Set system register 412 for serial transmission (general purpose port).

To switch between “computer link communication” and “serial data communication” (general purpose port), execute an **F144 (TRNS)** instruction. Set “n” (the number of transmission bytes) to H8000, and then execute the instruction.

When executed when “computer link” is selected, the setting will change to “general purpose port.”



When executed when “general purpose port” is selected, the setting will change to “computer link.”



R9032: COM. port selection flag, this flag turns on when “General purpose port” is selected.

Note

When the power is turned on, the port use will revert to the setting of system register 412.

Program and operation during transmission

To transmit, write the transmission data to the data table, select it with an **F144 (TRNS)** instruction, and execute.

Data table for transmission

Data register areas beginning with the area selected by S are used as the data table for transmission.

[S]		
[S+1]	2	1
[S+2]	4	3
⋮		
[S+n]	2n	2n−1

← The number of bytes not yet transmitted is stored here.

Storage area for transmission data (the circled numbers indicate the order of transmission).

Write the transmission data to the transmission data storage area selected with S (from the second word on) using an **F0 (MV)** or **F95 (ASC)** instruction.

– Do not include an end code in the transmission data as it will be added automatically.

– If the start code is set to “Yes”, do not include a start code in the transmission data as it will be added automatically.

When the **F144 (TRNS)** instruction is executed, the number of data bytes not yet transmitted is stored in the starting area of the data table.

Note

Take care that the transmission data table and reception buffer areas (set in system registers 417 and 418) do not overlap.

Example:

Transmitting the eight characters A, B, C , D, E, F, G and H (8 bytes of data)
In this example, the data table is DT100 to DT104.

DT100	K8	
DT101	H42 (B)	H41 (A)
DT102	H44 (D)	H43 (C)
DT103	H46 (F)	H45 (E)
DT104	H48 (H)	H47 (G)

← The number of bytes not yet transmitted is stored at each transmission.

Data is transmitted in order from the lower byte.

Data table before transmission

Program

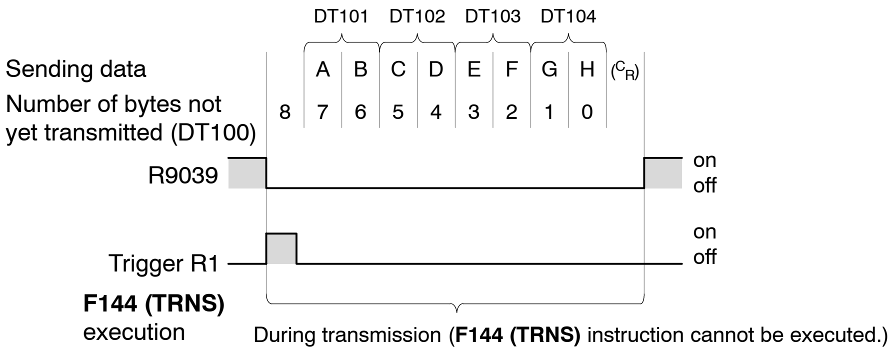
Select the starting address of the transmission data table with S and the number of transmission data bytes with “n”.

```
 R0
─┤ ├─[ F1 DMV, H44434241, DT101 ]     Write the transmission
      [ F1 DMV, H48474645, DT103 ]     data to the data table
 R1
─┤ ├─[  F144 TRNS, DT100, K 8    ]     Transmit the data in
                                       the data table
```

Operation

If the execution condition (trigger) for the **F144 (TRNS)** instruction is on when sending completed flag (R9039) goes on, operation will proceed as follows:

1) The “n” is preset in S (the number of bytes not yet transmitted). Furthermore, reception completed flag (R9038) is turned off and the reception data number is cleared to zero.
2) The data in the data table is transmitted in order from the lower byte in S+1.
   – As each byte is transmitted, the value in S (the number of bytes not yet transmitted) decrements by 1.
   – During transmission, the sending completed flag (R9039) goes off.
   – If the start code STX is set to “Yes” using system register 413, the start code will be automatically added to the beginning of the data.
   – The end code selected is automatically added to the end of the data.



3) When the specified quantity of data has been transmitted, the value in S (the number of bytes not yet transmitted) will be zero and the sending completed flag (R9039) will go on.

The **F144 (TRNS)** instruction cannot be executed and the R9039 is not turned on unless pin number 5 of RS232C port is turned on.

Transmission without an end code

Specify the number of transmission bytes with a negative number.
Set the end code to “Note” for transmission and reception.

Example:

Program for transmitting 8 bytes of data without an end code

```
 X0
─┤ ├─(DF )──────────────────> 1

1 >──[F144 TRNS, DT100, K–8   ]
                        ↑
                        └── Set to K–8.
```

Preparation of reception

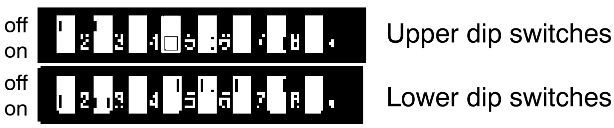
1) Setting the transmission format

For FP10SH

The initial settings for the transmission format are as follows:
– Data length: 8 bits
– Parity check: Yes, odd
– Stop bits: 1 bit
– End code: C_R
– Start code: No STX

To change the transmission format to match the external device connected to the COM. port, set the parameters with the upper row of operation mode switches.

Operation mode switches (using upper dip switch)



Functions		Settings							
		SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
MODEM control	Disabled	**off**							
	Enabled	on							
Start code	STX (H02) invalid		**off**						
	STX (H02) valid		on						
End code	None			off	off				
	C_R (H0D) and LF (H0A)			on	off				
	C_R (H0D)			**off**	**on**				
	EXT (H03)			on	on				
Stop bit	2 bits					off			
	1 bit					**on**			
Parity check	Invalid						off	off	
	Even parity						on	off	
	Odd parity						**on**	**on**	
Data length (character bit)	7 bits								off
	8 bits								**on**

For FP2/FP2SH

Set the transmission format parameter so that the “Transmission Format Setting” of system register 413 matches the external device connected to the COM. port. The default setting is the same as that of the FP10SH.

When the start code is valid, the data from the reception of STX to the reception of the selected end code is considered to be one frame.

2) Setting the baud rate

For FP10SH

The baud rate (transmission speed) for serial transmission is initially set to 9600 bps.

To change the baud rate to match the external device connected to the COM. port, set the lower row of operation mode switches as shown below.

Operation mode switches (using lower dip switch)

Functions		Settings							
		SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
Transmission speed	**115,200 bps**	—	—	—	—	—	off	off	off
	57,600 bps	—	—	—	—	—	on	off	off
	38,400 bps	—	—	—	—	—	off	on	off
	19,200 bps	—	—	—	—	—	on	on	off
	9,600 bps	—	—	—	—	—	**off**	**off**	**on**
	4,800 bps	—	—	—	—	—	on	off	on
	2,400 bps	—	—	—	—	—	off	on	on
	1,200 bps	—	—	—	—	—	on	on	on

For FP2/FP2SH

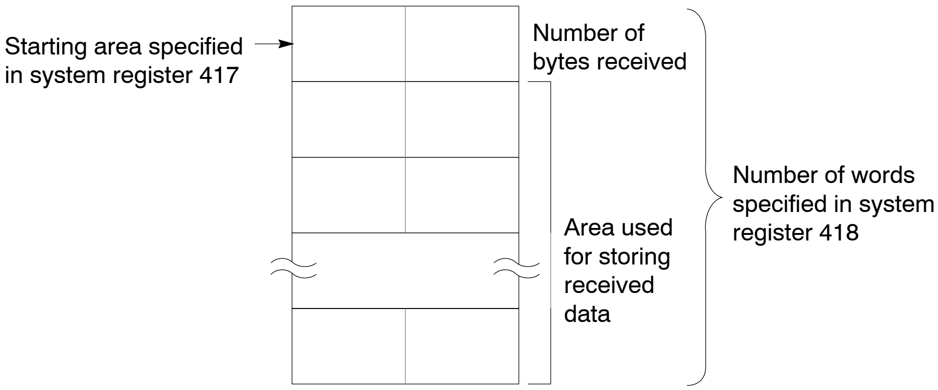
Set the transmission speed so that the “COM Port Baud Rate Setting” of system register 414 matches the external device connected to the COM. port. The default setting is “19200 bps”.

3) Setting the use of the COM. port

Set system register 412 for serial transmission (general purpose port).
The use of the COM. port can be changed by executing an F144 (TRNS) instruction.

4) Setting the reception buffer: System register 417 and 418

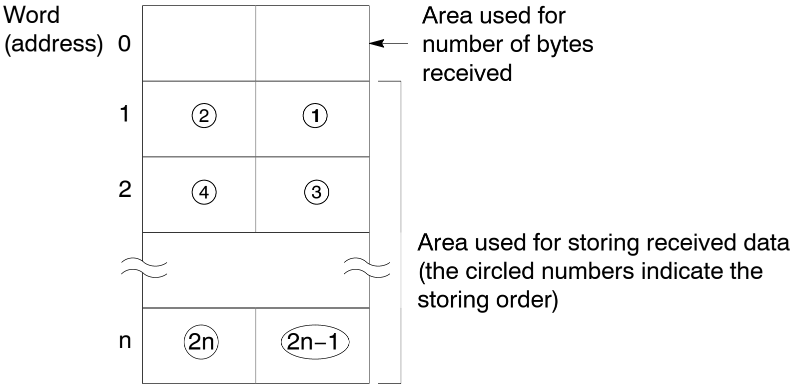
All areas of the data register are initially set for use as the reception buffer. To change the reception buffer, set the starting area number in system register 417 and the size (number of words, Max. 1,024 words) in system register 418.
The reception buffer will be as follows:



Program and operation during reception

Data sent from the external device connected to the COM port will be stored in the data register areas set as the reception buffer.

Reception buffer



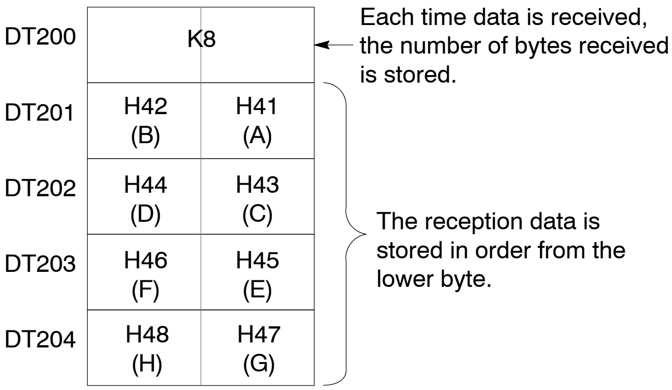
Each time data is received, the amount of data received (number of bytes) is stored as a count in the leading address of the reception buffer. The initial value is zero.

The data received is stored in order in the reception data storage area beginning from the lower byte of the second word of the area.

Example:

Reception of the eight characters A, B, C, D, E, F, G, and H (8 bytes of data) from an external device

The reception buffer is DT200 to DT204 in this example.
System register settings are as follows:
– System register 417: K200
– System register 418: K5



Reception buffer when reception is completed

Program

When reception of data from an external device has been completed, the reception completed flag (R9038) goes on and further reception of data is not allowed.

To receive more data, an **F144 (TRNS)** instruction must be executed to turn off the reception completed flag (R9038) and clear the byte number to zero.

```
  R0
──┤ ├──( DF )──[ F144  TRNS,  DT100,  K 0  ]
```

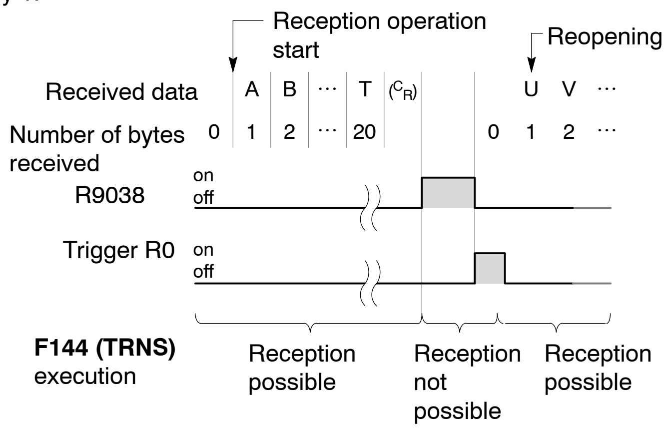
To repeat reception only, set to K0.

R9038 will also go off when the number of transmission bytes is set and transmission is carried out.

Operation

When the reception completed flag (9038) is off and data is sent from an external device, operation will proceed as follows. (After RUN, R9038 is off during the first scan.)

1) The data received is stored in order in the reception data storage area of reception buffer beginning from the lower byte of the second word of the area.
   Start and end codes will not be stored.
   With each one byte received, the value in the leading address of the reception buffer is incremented by 1.



2) When an end code is received, the reception completed flag (R9038) goes on. After this, no further reception of data is allowed.
3) When an **F144 (TRNS)** instruction is executed, the reception completed flag (R9038) goes off and the number of received data bytes is cleared to zero. Further data received is stored in order in the reception data storage area beginning from the lower byte of the second word of the area.

Note

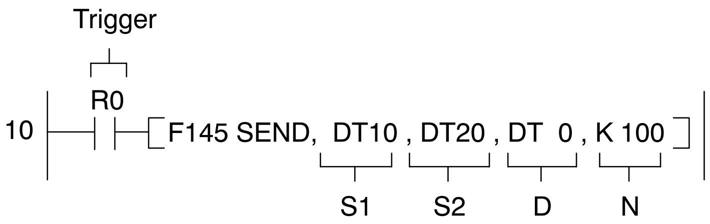
For repeated reception of data, refer to the following procedure 1) to 5).
1) Receive data
2) Reception completed (R9038: on, Reception: not allowed)
3) Process received data
4) Execute F144 (TRNS) instruction (R9038: off, Reception: enable)
5) Receive further data

F145 (SEND) P145 (PSEND) Data send (For MEWTOCOL master mode)

Availability
FP0R FP−X: Ver 1.2 or more FPΣ: 32k

Outline Sends specified data to another PLC or computer from the serial port of the unit.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F145 (SEND)
		DT 10
		DT 20
		DT 0
		K 100

S1	Starting 16−bit area for storing control data
S2	Starting 16−bit area for storing source data (data area at the local unit)
D	16−bit area of destination to send (The device No. is fixed at 0).
N	Starting 16−bit address of the destination to send.

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	N/A	A	A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	N/A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A	N/A
N	N/A	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	N/A	N/A	N/A	A	A	A

A: Available
N/A: Not Available

(*1) I0 to ID

(*2) It can be specified only for the FP0R, FPΣ V3.20 or later, FP−X V2.50 or later, however, an operation error will occur not a syntax error if K or H constant is specified as S1 and D.

Operation

- It is used to send commands to the serial port (COM1 or COM2) of the specified unit in the MEWTOCOL−COM mode connecting the unit that enables to receive the computer link command. Specify the computer link for the operation mode (system register setting) of the COM port to be used.
- The data of the local area specified by [S2] is written in the area of the remote unit specified by [D] and [N], according to the specification for the 2−word data stored in the control data with starting area specified by [S1].

Specifications for each item

- The control data specified by [S1][S1+1] is specified as follows.

[S1]: Specifying transmission unit and transmission method

[S1]:

H0			
Word unit transmission	Specifies No. of transmission words H001 to H1FB (1 to 507): when transmitting to Group A H001 to H18 (1 to 24)*: when transmitting to Group B H1FB is 507 words. H18 is 24 words.		

Group A	FPΣ, FP−X, FP0R, FP2, FP2SH, FP10SH
Group B	FP0, FP−e

* However, if either SV or EV is specified for the remote unit's device No. of [D], up to H19 (25) words is available.

[S1]:

H8		H0 fixed	
Bit unit transmission	Bit No. of remote unit (H0 to HF)		Bit No. of local unit (H0 to HF)

[S1+1]: Specifying the remote unit

[S1+1]:

	H0 fixed		
Selects COM port (H1 or H2)		Unit No. (H00 to H63) (0 to 99)	

(1) Specifying the transmission unit and transmission method [S1]
If data is to be sent in word units, specify the data volume, and if it is to be sent in bit units, specify the position of the target bit.

(2) Specifying the remote unit [S1+1]
Specify the remote unit with the unit number. When H00, it is global transmission. (No response)
Specify either the COM1 or COM2 port from which data is transmitted to the remote unit.
Specify H1 if only one COM port is available.

(3) Specify the area of the local unit by [S2] in which the data to be transmitted is stored
Specify the memory area of the local unit in which the data to be transmitted is stored.

(4) Specify the area of the remote unit for storing by [D] and [N].
Specify 0 for the device No. of [D].
Specify the memory area of the remote unit in which the data to be transmitted is stored, specifying the type D and the address N in combination.

Example) [D]:DT0, [N]:K100
↓
DT100

For the FP0R, FPΣ V3.20 or later, FP−X V2.50 or later, transmission can be performed without checking when DT0 or LD0 is specified for D, and H constant is specified for N.
(Example) In case of DT0 and HFFFF, it is possible to access DT63353. It is convenient to access the data registers of an eco−power meter KW8M.

- The MEWTOCOL−COM command is created according to the operands specified by [S1],[S1+1], [S2], [D], and [N].

Flag conditions

Σ Error flag (R9007) : Turns on and stays on when:

Σ Error flag (R9008) : Turns on for an instant when:

- The control data of [S1] and [S1+1] is a value outside of the specified range.
- The number of words specified by S1 causes the area of S2 or D to be exceeded when word unit transmission is being used.
- [D]+[N] exceeds the area of [D].
- The operation mode for the target COM port is other than compute link.
- Word unit
  If [D] is DT/LD, it turns on when [N] is not 0 to 32767.
  If [D] is WY/WR/WL/SV/EV, it turns on when [N] is not 0 to 9999.
- Bit unit
  It turns on when [D] is not WY/WR/WL.
  It turns on when [N] is not 0 to 999.
- The device No. of [D] is not 0.
- The communication cassette has not been installed for the target COM port.

Precautions during programming

- Specify the computer link for the operation mode (system register setting) of the COM port to be used.
- It is not possible to execute multiple F145 (SEND) instructions and F146 (RECV) instructions for the same communication port simultaneously.
  The program should be set up so that these instructions are executed when the SEND/RECV execution enabled flag (R9044: COM1/R904A: COM2) is on.

R9044 (COM1)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled
R904A (COM2)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled

- The SEND instruction only requests that the data be sent, but the actual processing takes place when the ED instruction is executed.
  The SEND/RECV execution end flag (R9045: COM1/R904B: COM2) can be used to check whether or not the transmission has been completed.

R9045 (COM1)	0: Completed normally 1: Completed with error (The error code is stored in DT90124.)
DT90124 (COM1)	If the transmission has been completed with an error (R9045 is on), the contents of the error (error code) are stored.
R904B (COM2)	0: Completed normally 1: Completed with error (The error code is stored in DT90125.)
DT90125 (COM2)	If the transmission has been completed with an error (R904B is on), the contents of the error (error code) are stored.

For information on the contents of error codes, refer to the manual. If the error code is H73, a communication time−out error has occurred.
The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 2.5 ms), using the setting of system register 32.
The default value is set to 10 seconds.

Error code (HEX)	Description
73	Time−out: Waiting for response

- For global transmission (the transmission performed by specifying H00 for the unit No.), the program should be set up so that the transmission is executed after a time of at least the maximum scan time elapsed.
- The F145 or F146 instruction cannot be executed if the target is a special internal relay (from R9000) or a special data register (from DT9000) or file register FL.

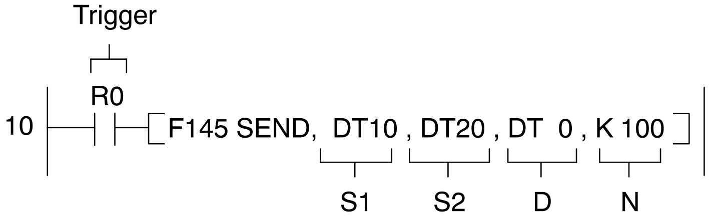
F145 (SEND) P145 (PSEND) Data send (For MODBUS master mode)

Availability
FP0R/FP−X FPΣ: 32k

Outline Sends specified data to another PLC or computer from the serial port of the unit.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	0
11	F145	(SEND)
	DT	10
	DT	20
	DT	0
	K	100

S1	Starting 16−bit area for storing control data
S2	Starting 16−bit area for storing source data (data area at the local unit)
D	16−bit area of destination to send (The device No. is fixed at 0).
N	Starting 16−bit address of the destination to send.

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	N/A	A	A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	N/A	A	A	N/A	N/A	A
D	N/A	A	A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
N	N/A	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	N/A	N/A	N/A	A	A	A

A: Available
N/A: Not Available

(*1) I0 to ID

(*2) It can be specified only for the FP0R, FPΣ V3.20 or later, FP−X V2.50 or later, however, an operation error will occur not a syntax error if K or H constant is specified as S1 and D.

Operation

- It is used to send commands to the serial port (COM1 or COM2) of the specified unit in the MODBUS mode connecting the unit that enables to receive the MODBUS command. (MODBUS command 05, 06, 15 and 16)
- The data of the local area specified by [S2] is written in the area of the remote unit specified by [D] and [N], according to the specification for the 2−word data stored in the control data with starting area specified by [S1].

Specifications for each item

- The control data specified by [S1][S1+1] is specified as follows.

[S1]: Specifying transmission unit and transmission method

[S1]:	H0			
	Word unit transmission	Specifies No. of transmission words (H001 to H07F) *According to the restrictions on the MODBUS protocol.		

[S1]:	H8		H0 fixed	
	Bit unit transmission	Bit No. of remote unit (H0 to HF)		Bit No. of local unit (H0 to HF)

[S1+1]: Specifying the remote unit

[S1+1]:		H0 fixed		
	Selects COM port (H1 or H2)		Unit No. (H00 to H63) (0 to 99)	

(1) Specifying the transmission unit and transmission method [S1]
If data is to be sent in word units, specify the data volume, and if it is to be sent in bit units, specify the position of the target bit.
*In word units, the maximum of 127 (7Fh) words can be transmitted as the transmission range is up to 254 bytes.

(2) Specifying the remote unit [S1+1]
Specify the remote unit with the unit number. When H00, it is global transmission. (No response)
Specify either the COM1 or COM2 port from which data is transmitted to the remote unit.
Specify H1 if only one COM port is available.

(3) Specify the area of the local unit by [S2] in which the data to be transmitted is stored
Specify the memory area of the local unit in which the data to be transmitted is stored.

(4) Specify the area of the remote unit for storing by [D] and [N].
Specify 0 for the device No. of [D].
Specify the memory area of the remote unit in which the data to be transmitted is stored, specifying the type D and the address N in combination.

Example) [D]:DT0, [N]:K100
↓
DT100

- The MODBUS command is created according to the operands specified by [S1],[S1+1], [S2], [D], and [N].
  When being transmitted in word units: The command 06 (DT1 word write), command 15 (Y, R multi−points write) and command 16 (DT multi−words write) can be transmitted.
  When being transmitted in bit units: The command 05 (Y, R single point write) can be transmitted.
  The transmission is executed adding the 2 bytes of CRC at the end after the MODBUS command has been created.

Explanation of command

Command 05 (Y, R single write) send

Example) When the value of the bit 0 of WR3 is transmitted to the 1st bit of WY1 of the unit No. 7 in the remote unit from the COM1.

[F145 (SEND), DT10, WR3, WY0, K1]

[S1]:	H8	H1	H0 fixed	H0	DT10
	Bit unit transmission	Bit No. of remote unit (H0 to HF)		Bit No. of local unit (H0 to HF)	

*Bit units (H8) should be specified for the transmission method of the [S1] to send the command 05.

[S1+1]:	H1	H0 fixed	H0	H7	DT11
	Selects COM port (H1 or H2)		Unit No. (H00 to H63) (0 to 99)		

[S1]: DT10 (DT10=8100H, DT11=1007H)
[S2]: WR3 (WR3=0007H)
[D]: WY0
[N]: K1

Command conversion

MODBUS commands

1	Slave address	07
2	Command (05H)	05
3	Coil No. (H)	00
4	Coil No. (L)	11
5	Setting status (H)	FF
6	Setting status (L)	00
7	CRC16 (H)	DC
8	CRC16 (L)	59

* Reads a value of the bit 0 of WR3 and sets the condition by selecting ON or OFF.
  Specify ON=FF00, OFF=0000.

Command 06 (DT1 word write) send

Example) When the 1-word data of WR3 is transmitted to the DT1000 of the unit No. 7 in the remote unit from the COM1.

[F145 (SEND), DT10, WR3, DT0, K1000]

[S1]:	H0	H0	H0	H1	DT10
	Word unit transmission	Specifies No. of transmission words (H1 to H7F)			

*Word units (H0) for the transmission method of [S1] and (H1) for No. of transmission words should be specified to send the command 06.

[S1+1]:	H1	H0 fixed	H0	H7	DT11
	Selects COM port (H1 or H2)		Unit No. (H00 to H63) (0 to 99)		

[S1]:DT10(DT10=0001H、DT11=1007H)
[S2]:WR3(WR3=1234H)
[D] :DT0
[N] :K1000

Command conversion

MODBUS commands

1	Slave address	07
2	Command (06H)	06
3	Starting No. of write (H)	03
4	Starting No. of write (L)	E8
5	Write data (H)	12
6	Write data (L)	34
7	CRC16 (H)	04
8	CRC16 (L)	AB

* Reads the word data of WR3 and sets in the write data.

Command 15 (Y, R multi−points write) send

Example) When the 64−bit data from the bit 0 of the WR3 to the bit F of the WR6 is transmitted to the W0 to Y3F of the unit No. 7 in the remote unit from the COM1.

[F145 (SEND), DT10, WR3, WY0, K0]

[S1]:

H0	H0	H0	H4	DT10
Word unit transmission	Specifies No. of transmission words (H1 to H7F)			

*Word unit (H0) should be specified for the transmission method of [S1] to send the command 15.

[S1+1]:

H1	H0 fixed	H0	H7	DT11
Selects COM port (H1 or H2)		Unit No. (H00 to H63) (0 to 99)		

[S1]:DT10(DT10=0004H、DT11=1007H)
[S2]:WR3(WR3=3210H
WR4=7654H
WR5=BA98H
WR6=FEDCH)
[D] :WY0
[N] :K0

Command conversion

*Specify the coil No. of the destination for the starting No. of status change. (Remote unit)
The quantity of changed coils is that the No. of write bits is changed to HEX.
Max. quantity of changed coils is 2032 (07F0H).
(due to the restrictions on the MODBUS protocol)
No. of data (No. of bytes) is calculated regarded 8 coils as 1 data (1 byte). (Max. 254 (FEH) bytes)

MODBUS commands

1	Slave address	07
2	Command (0FH)	0F
3	Starting No. of status change (H)	00
4	Starting No. of status change (L)	00
5	Quantity of changed coils (H)	00
6	Quantity of changed coils (L)	40
7	No. of data (No. of bytes)	08
8	Setting data 1	10
9	Setting data 2	32
10	Setting data 3	54
11	Setting data 4	76
12	Setting data 5	98
13	Setting data 6	BA
14	Setting data 7	DC
15	Setting data 8	FE
16	CRC16 (H)	6C
17	CRC16 (L)	B3

Command 16 (DT multi−words write) send

Example) When the 3−word data from WR3 to WR5 is transmitted to DT500 to DT502 of the unit No. 7 of the remote unit.

[F145 (SEND), DT10, WR3, DT0, K500]

[S1]:

H0	H0	H0	H3
Word unit transmission	Specifies No. of transmission words (H1 to H7F)		

DT10

*Word units (H0) should be specified for the transmission method of [S1] to send the command 16.

[S1+1]:

H1	H0 fixed	H0	H7
Selects COM port (H1 or H2)		Unit No. (H00 to H63) (0 to 99)	

DT11

[S1]:DT10(DT10=0003H, DT11=1007H)
[S2]:WR3(WR3=0011H
WR4=2233H
WR5=4455H)
[D] :DT0
[N] :K500

Command conversion

MODBUS commands

1	Slave address	07
2	Command (10H)	10
3	Starting No. of write (H)	01
4	Starting No. of write (L)	F4
5	No. of write registers (H)	00
6	No. of write registers (L)	03
7	No. of data (No. of bytes)	06
8	Write data 1 (H)	00
9	Write data 1 (L)	11
10	Write data 2 (H)	22
11	Write data 2 (L)	33
12	Write data 3 (H)	44
13	Write data 3 (L)	55
14	CRC16 (H)	5A
15	CRC16 (L)	E7

*Max. quantity of write registers is 127 (7FH) (due to the restrictions on the MODBUS protocol).
No. of data (No. of bytes) is calculated regarded No. of write registers as 2 bytes. (Max. 254 (FEH) bytes)

Flag conditions

Σ Error flag (R9007) : Turns on and stays on when:

Σ Error flag (R9008) : Turns on for an instant when:

- The control data of [S1] and [S1+1] is a value outside of the specified range.
- The number of words specified by S1 causes the area of S2 or D to be exceeded when word unit transmission is being used.
- [D]+[N] exceeds the area of [D].
- The MODBUS mode has not been specified for the COM port of the control data specified by [S1+1].
- The area of [D] is DT in bit unit transmission.
- The device No. of [D] is not 0.

Precautions during programming

- It is not possible to execute multiple F145 (SEND) instructions and F146 (RECV) instructions for the same communication port simultaneously.
  The program should be set up so that these instructions are executed when the SEND/RECV execution enabled flag (R9044: COM1/R904A: COM2) is on.

R9044 (COM1)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled
R904A (COM2)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled

- The SEND instruction only requests that the data be sent, but the actual processing takes place when the ED instruction is executed.
  The SEND/RECV execution end flag (R9045: COM1/R904B: COM2) can be used to check whether or not the transmission has been completed.

R9045 (COM1)	0: Completed normally 1: Completed with error (The error code is stored in DT90124.)
DT90124 (COM1)	If the transmission has been completed with an error (R9045 is on), the contents of the error (error code) are stored.
R904B (COM2)	0: Completed normally 1: Completed with error (The error code is stored in DT90125.)
DT90125 (COM2)	If the transmission has been completed with an error (R904B is on), the contents of the error (error code) are stored.

For information on the contents of error codes, refer to the manual. If the error code is H73, a communication time−out error has occurred.
The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 2.5 ms), using the setting of system register 32.
The default value is set to 10 seconds.

Error code (HEX)	Description
73	Time−out: Waiting for response

- For global transmission (the transmission performed by specifying H00 for the unit No.), the program should be set up so that the transmission is executed after a time of at least the maximum scan time elapsed.
- The F145 or F146 instruction cannot be executed if the target is a special internal relay (from R9000) or a special data register (from DT9000).

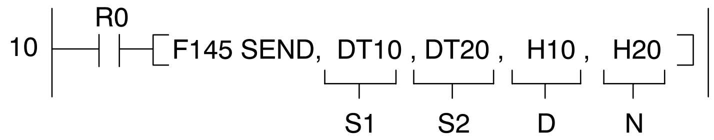
Availability
FP0R FP−X: Ver. 2.50 FPΣ: Ver. 3.20

F145 (SEND) Data send (MODBUS master II: Type directly specifying MODBUS address)

Outline Sends specified data to another PLC or computer from the serial port of the unit.
Feature: Data can be transmitted with this instruction only.

Program example

Ladder Diagram



Boolean: Address	Boolean: Instruction		
10	ST	R	0
11	F145		(SEND)
	DT		10
	DT		20
	H		10
	H		20

S1	Specification of transmission port, transmission command and destination unit No.
S2	Starting 16−bit area for storing source data
D	Specification of MODBUS address
N	Specification of No. of data to be sent (No. of words or bits)

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	N/A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	N/A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A	N/A
N	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A	A

A: Available
N/A: Not Available

(*1) I0 to ID

(*2) This instruction is available only for FP0R/FPΣ V3.20 or later/FP−X V2.50 or later.

Description

The send data specified by [S2] is sent to the MODBUS address specified by [D] with the MODBUS command by specifying the transmission port, transmission command (5 or 6) and destination unit number. (MODBUS commands 05, 06, 15 and 16)

The feature is that data can be transmitted with this instruction only.

Specifications for each item

- [S1]: Specifying port number, transmission command and destination unit number

[S1]:

H1 or H2	H5 or H6		
Selects COM port	Specifies transmission command	Unit No. (H00 to HFF)	

(1) Specifying COM port
Specify H1 for COM 1 port, and H2 for COM2 port.
If only one COM port is available, specify H1.

(2) Specifying transmission command.
H5 (HD) = Bit data/H6 (HE) = Word data
When specifying multiple points by [N], the command is automatically converted to the command 15 or 16 for sending multiple bits or multiple words.

On V1.06 or later version of FP0R, the command for writing multiple points (0F or 10) can be issued by specifying HD or HE even if 1 bit or 1 word is transferred.

(3) Specifying destination unit number

- Specify the area by [S2] in which the data to be transmitted is stored.
  Specify the starting number of the operation memory of the local unit in which the data to be transmitted is stored.
  This instruction can be executed even if the transmission command specified by [S1] and the device type in the remote unit differs.
  That means the contents of DT can be transmitted by specifying the bit data, or the contents of WR can be transmitted by specifying the word data.
  When the bit data is specified by [S1], data is always transmitted from the bit 0.
- Specifying the MODBUS address of the remote unit where data is transmitted by [D].
  Settable address: H0 to HFFFF
- Specifying the number of data transmitted by [N]
  Settable number of data
  For bit data: Max. 2040 (07F8H)
  For word data: Max. 127 (7FH)

Explanation of command

Command 05 (Coil single−point forcing)

Example) When the value of the bit 0 of WR3 is transmitted to the bit of the bit address H7788 of the unit No. 7 from the COM1.

[F145(SEND), H1507, WR3, H7788, K1]

[S1]:

H1	H5	H0	H7
Selects COM port		Unit No. (H00 to HFF)	

[S1]: H1507
[S2]: WR3 (WR3=0007H)
[D]: H7788
[N]: K1

Command conversion

MODBUS commands

1	Slave address	07
2	Command (05H)	05
3	Coil No. (H)	77
4	Coil No. (L)	88
5	Setting status (H)	FF
6	Setting status (L)	00
7	CRC16 (H)	17
8	CRC16 (L)	C2

* Reads a value of the bit 0 of WR3 and sets the condition by selecting ON or OFF.
Specify ON=FF00, OFF=0000.

Command 06 (Register single preset)

Example) When the 1−word data of WR3 is transmitted to the address H7788 of the unit No. 7 from the COM1.

[F145(SEND), H1607, WR3, H7788, K1]

[S1]:

H1	H6	H0	H7
Selects COM port		Unit No. (H00 to HFF)	

[S1]:H1607
[S2]:WR3(WR3=1234H)
[D] :H7788
[N] :K1

Command conversion

MODBUS commands

1	Slave address	07
2	Command (06H)	06
3	Starting No. of write (H)	77
4	Starting No. of write (L)	88
5	Write data (H)	12
6	Write data (L)	34
7	CRC16 (H)	53
8	CRC16 (L)	C2

* Reads the word data of WR3 and sets in the write data.

Command 15 (Multi−point coil forcing)

Example) When the 64−bit data from the bit 0 of the WR3 to the bit F of the WR6 is transmitted to the bit address H7788 of the unit No. 7 from the COM1.

[F145(SEND), H1507, WR3, H7788, K64]

[S1]:

H1	H5	H0	H7
Selects COM port		Unit No. (H00 to HFF)	

[S1]:H1507
[S2]:WR3 (WR3=3210H
WR4=7654H
WR5=BA98H
WR6=FEDCH)
[D] :H7788
[N] :K64

Command conversion

MODBUS commands

1	Slave address	07
2	Command (0FH)	0F
3	Starting No. of status change (H)	77
4	Starting No. of status change (L)	88
5	Quantity of changed coils (H)	00
6	Quantity of changed coils (L)	40
7	No. of data (No. of bytes)	08
8	Setting data 1	10
9	Setting data 2	32
10	Setting data 3	54
11	Setting data 4	76
12	Setting data 5	98
13	Setting data 6	BA
14	Setting data 7	DC
15	Setting data 8	FE
16	CRC16 (H)	3B
17	CRC16 (L)	65

* When specifying multiple points by [N], the command is automatically corrected.
H5 bit single write => H15 bit multiple write
The starting No. of status change is H7788. (Remote unit)
The quantity of changed coils is that the No. of write bits is changed to HEX.
Max. quantity of changed coils is 2040 (07F8H).
(due to the restrictions on the MODBUS protocol)
No. of data (No. of bytes) is calculated regarded 8 coils as 1 data (1 byte). (Max. 255 (FFH) bytes)

Command 16 (Multi-point register preset) send

Example) When the 3-word data from DT3 to DT5 is transmitted to the address H7788 of the unit No. 7 from the COM1 port.

[F145(SEND), H1607, DT3, H7788, K3]

[S1]:

H1	H6	H0	H7

Selects COM port (H1, H6) Unit No. (H00 to HFF) (H0, H7)

[S1] : H1607
[S2] : DT3 (DT3=0011H
DT4=2233H
DT5=4455H)
[D] : H7788
[N] : K3

Command conversion

MODBUS commands

1	Slave address	07
2	Command (10H)	10
3	Starting No. of write (H)	77
4	Starting No. of write (L)	88
5	No. of write registers (H)	00
6	No. of write registers (L)	03
7	No. of data (No. of bytes)	06
8	Write data 1 (H)	00
9	Write data 1 (L)	11
10	Write data 2 (H)	22
11	Write data 2 (L)	33
12	Write data 3 (H)	44
13	Write data 3 (L)	55
14	CRC16 (H)	2C
15	CRC16 (L)	BA

* When specifying multiple points by [N], the command is automatically corrected.

* Max. quantity of write registers is 127 (7FH).
(due to the restrictions on the MODBUS protocol)
No. of data (No. of bytes) is calculated regarded No. of write registers as 2 bytes. (Max. 254 (FEH) bytes)

Flag conditions

Σ Error flag (R9007) :
Σ Error flag (R9008) :

- Turns on when the control data of [S1] is a value outside of the specified range.
- Turns on when the MODBUS mode has not been specified for the COM port of the control data specified by [S1].
- Turns on when the number of transmission data is 0.
- Turns on when the number of transmission data is negative.
- Turns on when the number of transmission data [N] exceeds the operation memory area specified by [S2].
- Turns on when the number of transmission data [N] exceeds the limit of the MODBUS specifications.

Precautions during programming

- It is not possible to execute multiple F145 (SEND) instructions and F146 (RECV) instructions for the same communication port simultaneously.
  The program should be set up so that these instructions are executed when the SEND/RECV execution enabled flag (R9044: COM1/R904A: COM2) is on.

R9044 (COM1)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled
R904A (COM2)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled

- The SEND instruction only requests that the data be sent, but the actual processing takes place when the ED instruction is executed.
  The SEND/RECV execution end flag (R9045: COM1/R904B: COM2) can be used to check whether or not the transmission has been completed.

R9045 (COM1)	0: Completed normally 1: Completed with error (The error code is stored in DT90124.)
DT90124 (COM1)	If the transmission has been completed with an error (R9045 is on), the contents of the error (error code) are stored.
R904B (COM2)	0: Completed normally 1: Completed with error (The error code is stored in DT90125.)
DT90125 (COM2)	If the transmission has been completed with an error (R904B is on), the contents of the error (error code) are stored.

If the error code is H73, a communication time−out error has occurred.
The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 10 ms), using the setting of system register 32.
The default value is set to 10 seconds.

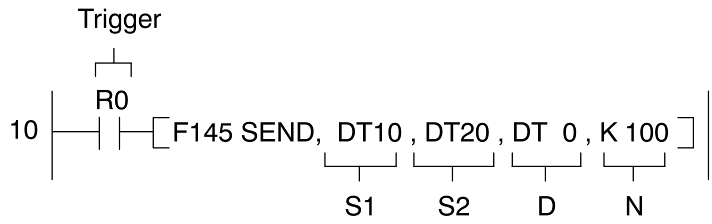
Error code (HEX)	Description
73	Time−out: Waiting for response

- For global transmission (the transmission performed by specifying H00 for the unit No.), the program should be set up so that the transmission is executed after a time of at least the maximum scan time elapsed.
- The F145 or F146 instruction cannot be executed if the target is a special internal relay (from R9000) on a special data register (DT90000).

F145 (SEND) P145 (PSEND) Data send (MEWNET link)

Outline Sends data to another station through link modules in the network.

Program example



Address	Instruction		
10	ST	R	0
11	F145		(SEND)
	DT		10
	DT		20
	DT		0
	K		100

S1	Starting 16-bit area for storing control data
S2	Starting 16-bit area for storing source data (data area at the local station)
D	Type of destination operands for storing data in the remote station. Be sure to select the area by setting address 0 (destination data area at another station).
N	Starting 16-bit area address for the destination operand specified in D above (destination data area in another station).

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A
N	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A

A: Available
N/A: Not Available

Explanation of example

① Example of word unit transmission

When the control data is as follows:

DT10(S1)=H0005 (=K5)

Word unit — 5 words

DT11(S1+1)=H010A

Unit No.10

Route No.1

the 5 words of data from DT20 to DT24 are sent to DT100 to DT104 of unit No. 10, which is connected to route No. 1, when the execution condition (trigger) R0 turns on.

② Example of bit unit transmission

When the control data is as follows:

DT10(S1)=H850D

Bit unit

Bit No. 13 of local station

Bit No. 5 of remote station

DT11(S1+1)=H010A

Unit No.10

Route No.1

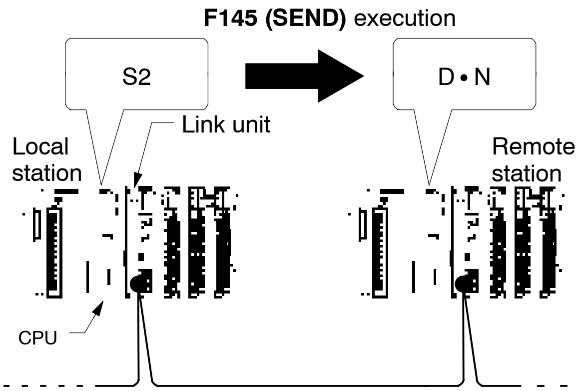
the on and off information of Bit No. 13 of DT20 is sent to Bit No. 5 of DT100 of Unit No. 10, which is connected to route No. 1, when the execution condition (trigger) R0 turns on.

If the network is configured only of the FP2, FP2SH, and FP10SH, specifying [FF] (HFF) for the unit number sends the same contents to all of the link stations on the same network.

Description

This sends the local station data for the area specified by S2 to the areas specified by the D and N of the remote stations connected with the MEWNET−W, MEWNET−P, and MEWNET−H.

The remote stations (routes and unit numbers), the transmission unit (bit unit or word unit), the transmission method, and other parameters are specified by the control data S1.



The remote station is specified by S1.

If general−purpose communication through the COM. port of the FP2, FP2SH, and FP10SH is being used, **F144 (TRNS)** instruction is used instead of this instruction.
Refer to the section describing the **F144 (TRNS)** instruction.

Specifying the various items

Control data (S1)

Specifying the remote station
Specify the remote station by means of a route number and unit number.
The setting is entered differently depending on whether the remote station is a PLC in the same network, or a PLC in a network on a different hierarchical level.

Specifying the transmission unit and transmission method
If data is to be sent in word units, specify the data volume, and if it is to be sent in bit units, specify the position of the target bit.

Specifying the memory area of the local station (S2)

Specify the memory area of the local station in which the data to be transmitted is stored.

Specifying the memory area of the remote station (D) and (N)

Specify the memory area of the remote station in which the data to be transmitted is stored, specifying the type D and the address N in combination.

Example: D: DT0, N: K100

↓

DT100

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The control data is a value outside of the specified range.
  - The remote station does not exist.
  - The number of words specified by S1 causes the area of S2 or D to be exceeded when word unit transmission is being used.

Transmitting to a PLC within the same network

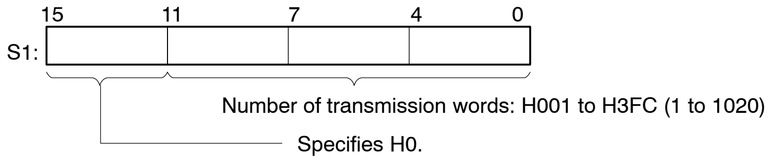
Specifying the control data (S1+1, S1)

The control data should be specified as an H constant. The transmission unit, transmission method and other parameters are specified with S1, and the remote station is specified with S1+1.

S1+1	S1
Specifying the remote station	Specifying the transmission unit and method

(1) Specifying word unit transmission

If word unit transmission is being used, the data for the specified number of words is sent from the memory area of the local station specified by S2, and is stored at the beginning of the memory area of the remote station specified by D and N.
If only the MEWNET−H network is being used, up to 1,020 words can be sent at one time, and if the network is using the MEWNET−P or MEWNET−W, up to 16 words can be sent at one time.

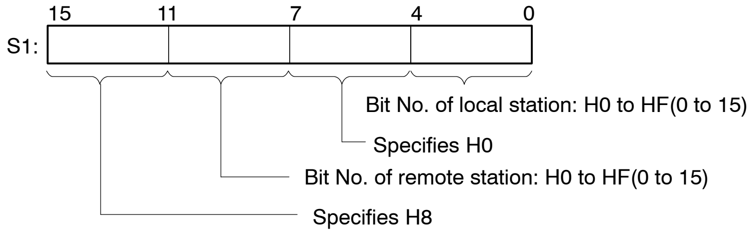


Example:

If 10 words of data are being sent, K10(H000A) should be specified in S1.

(2) Specifying bit unit transmission

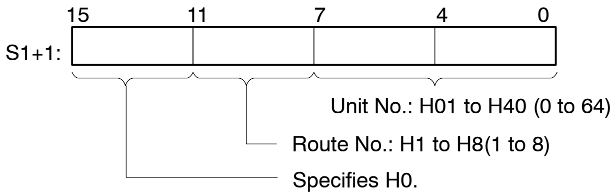
If bit unit transmission is being used, the information of the specified bit in the memory area of the local station specified by S2 is sent to the specified bit of the memory area of the remote station specified by D and N.



Example:

If the data of Bit No. 15 of the local station memory area is being sent to Bit No. 0 of the memory area in the remote station, H800F should be specified in S1.

(3) Specifying the remote station (common to both word/bit transmission)



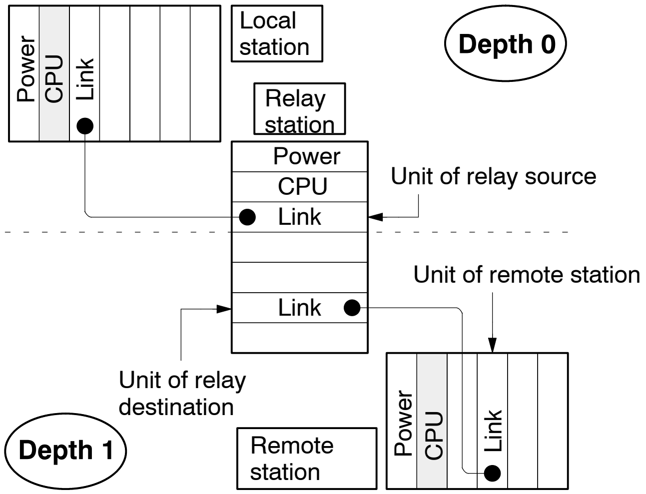
The unit number should be converted to a hexadecimal number and specified.
– For MEWNET−W: H01 to H20 (1 to 32)
– For MEWNET−P: H01 to H3F (1 to 63)
– For MEWNET−H: H01 to H40 (1 to 64)

Sending data to a PLC on a different hierarchical level

What is a hierarchical link?

A hierarchical link functions as a relay station between two link units installed on the same backplane, enabling communication between CPUs belonging to different networks.

Example: Communicating with a CPU at depth 1

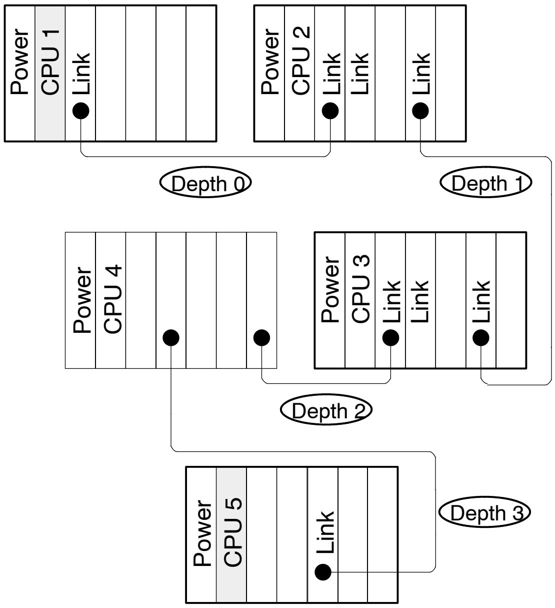


In this way, by passing data through a relay station, communication is possible to a depth of 3.

Note

When using the MEWNET−P and MEWNET−W, data can only be relayed one network deeper in the hierarchy.

Example: **Communicating with a CPU at depth 3 (sending data from CPU1 to CPU5)**



The numbers CPU1 to CPU5 have been temporarily assigned, for the purpose of indicating the relay order of the hierarchical links.

Specifying the control data (S1)

The control data should be specified as an H constant.
The transmission unit, transmission method and other parameters are specified with S1, and the remote station is specified with S1+1 and subsequent parameters (the relay source unit, relay destination unit, and unit targeted for communication). (depth + 3) words are required.

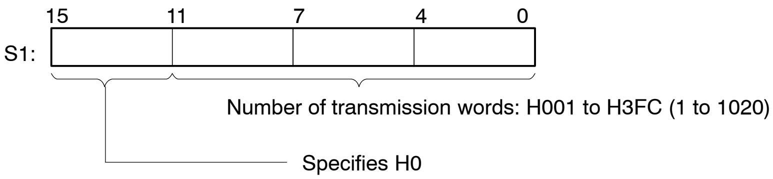
Example: **Control data when specifying a remote station which is at depth 3**

S1	**Specifying the transmission unit and method**			
[S1+1]	Local station	Depth (H03)	CPU1	Specifying the remote station
[S1+2]	Relay source	Relay destination	CPU2	
[S1+3]	Relay source	Relay destination	CPU3	
[S1+4]	Relay source	Relay destination	CPU4	
[S1+5]	Remote station	H00	CPU5	

————: Same network
−−−−−−−: Same backplane
The relay source is specified by a unit No. in the network, and the relay destination is specified by a route number on the backplane.

(1) Specifying word unit transmission

If word unit transmission is being used, the data for the specified number of words is sent from the memory area of the local station specified by S2, and is stored starting from the beginning of the memory area of the remote station specified by D and N.
If only the MEWNET−H network is being used, up to 1,020 words can be sent at one time, and if the network is using the MEWNET−P and MEWNET−W, up to 16 words can be sent at one time.

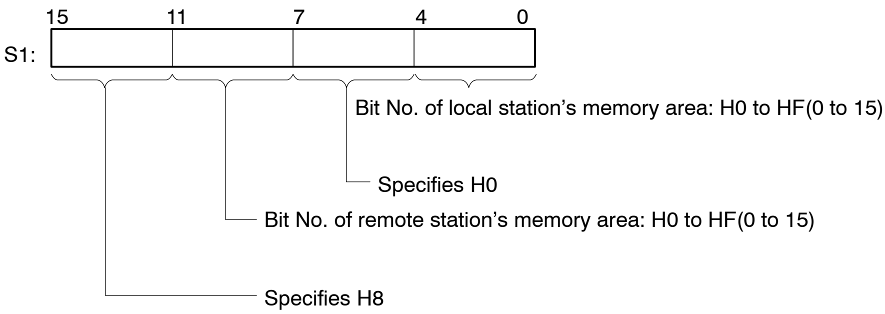


Example:

If 10 words of data are being sent, K10(H000A) should be specified in S1.

(2) Specifying bit unit transmission

If bit unit transmission is being used, the information of the specified bit in the memory area of the local station specified by S2 is sent to the specified bit of the memory area of the remote station specified by D and N.

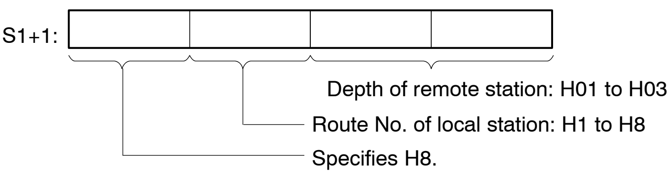


Example:

If the data of Bit No. 15 of the local station memory area is being sent to Bit No. 0 of the memory area in the remote station, H800F should be specified in S1.

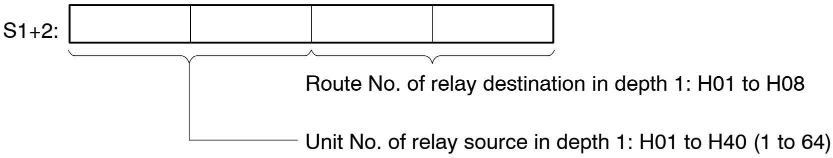
(3) Specifying the remote station (common to both word/bit transmission)

① Specifying the route No. and depth



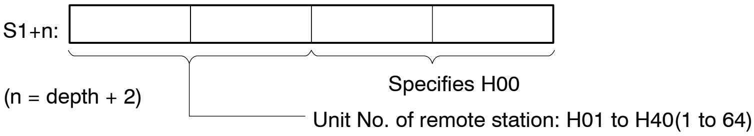
② Specifying the relay station

S+1 should be used to specify only the specified amount of depth, while (S1+3) is used to specify depth 2 for the same item, and (S1+4) is used to specify depth 3.



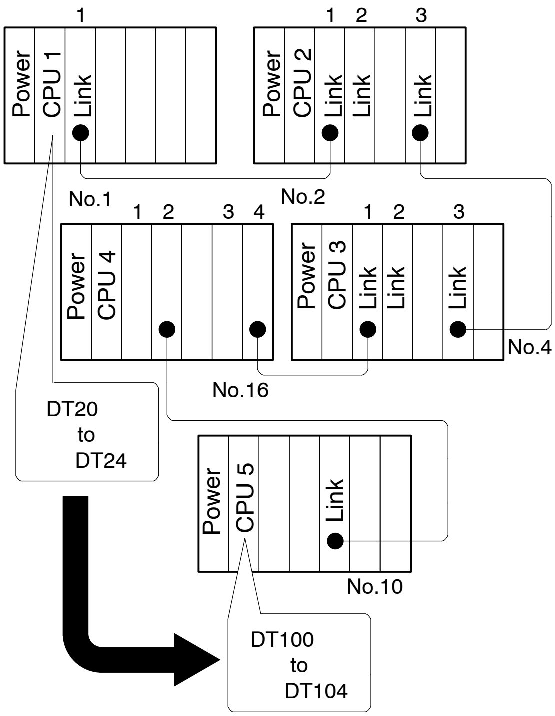
③ Specifying the remote station

This should be specified right after the specification of the relay station.

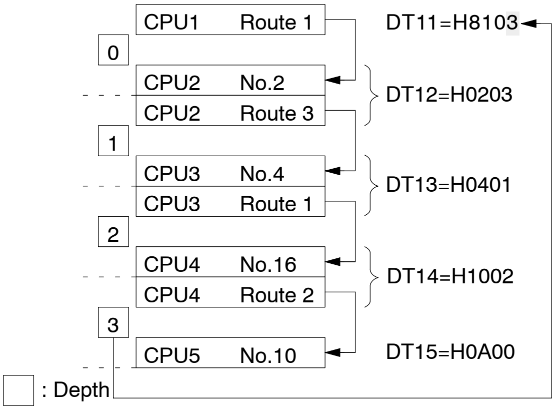


Example: When using the program example shown on page 3 – 339
In this example, the 5 words of data from DT20 to DT24 of the local station (CPU1) are sent to DT100 and subsequent addresses of the CPU (CPU5) shown below.

Connection diagram



In this example, the control data beginning with DT10 (depth 3 → 6 words) should be specified as shown below. To send the 5 words of data → DT10 = H0005



Precautions during programming

It is not possible to execute multiple **F145 (SEND)** instructions and **F146 (RECV)** instructions at the same time.

The program should be set up so that these instructions are executed when the MEWNET send/receive execution enabled flag (R9030) is on.

R9030	0: Execution inhibited (**F145 (SEND)**/**F146 (RECV)** instruction being executed) 1: Execution enabled

The **F145 (SEND)** instruction only requests that the data be sent, but the actual processing takes place when the **ED** instruction is executed. The MEWNET send/receive completed flag (R9031) can be used to check whether or not the transmission has been completed.

R9031	0: Completed normally 1: Completed with error (The error code is stored in DT9039.)
DT9039 (DT90039)	If the transmission has been completed with an error (R9031 is on), the contents of the error (error code) are stored.

For information on the contents of error codes, refer to the manual for that particular link unit. If the error code is H71 to H73, a communication time−out error has occurred. The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 10 ms), using the setting of system register 32. The default value is set to 10 seconds for FP2/FP2SH/FP10SH.

Error code (HEX)	Description
H71	Time out: Waiting for transmission answer
H72	Time−out: Waiting for transmission buffer to be emptied
H73	Time−out: Waiting for response

If there is any CPU other than the FP2SH and FP10SH connected to the network, global transmission (sending data using the HFF specification for the unit No.) should never be used.

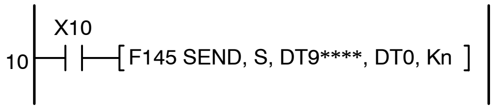
The **F145 (SEND)** instruction cannot be executed if the target is a special internal relay (from R9000) or a special data register (from DT9000/DT90000).

Additional information concerning the F145 (SEND) instruction

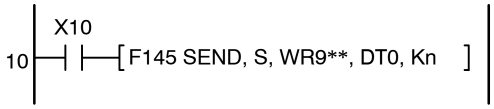
Sending the special data registers and special internal relays using the data transfer instruction

Special data registers and special internal relays cannot be sent using the **F145 (SEND)** instruction. Use a program like that shown below to send these types of data.

Sending FP2, FP2SH or FP10SH special data registers (source issuing the command: FP2/FP2SH/FP10SH)



Sending special internal relays (source issuing the command: FP2/FP2SH/FP10SH)



How to send FL (How to specify FL banks)

- How to specify the FL bank for destination units
  The FL to communicate is specified like FL0 + H10. Specify FL1 + H10 to specify the FL of the bank 1 (FL2 + H10 for the bank 2)
- How to specify the FL bank for a local unit
  Normally, FLn is specified for the FL for the local unit. If specifying, the FL will be sent to the bank which has been selected in the execution of this instruction.

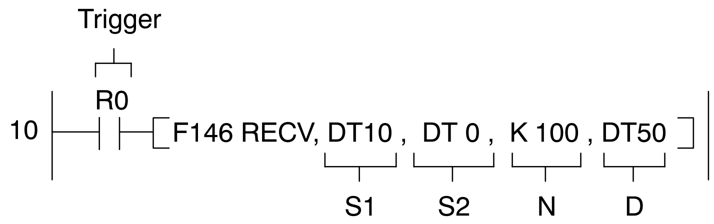
F146 (RECV) P146 (PRECV) Data receive (For MEWTOCOL master mode)

Availability
FP0R FP−X: Ver 1.2 or more FPΣ: 32k

Outline Receives specified data from the serial port of another PLC or computer to the unit.

Program example

Ladder Diagram



Boolean		
Address	**Instruction**	
10	ST	R 0
11	F146	(RECV)
	DT	10
	DT	0
	K	100
	DT	50

S1	Starting 16-bit area for storing control data
S2	16−bit area of destination to receive (The device No. is fixed at 0).
N	Starting address of the destination to receive.
D	Starting 16−bit area address for storing data received (destination data area at local unit).

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	N/A	A	A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A	N/A
N	N/A	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	N/A	N/A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) I0 to ID

(*2) It can be specified only for the FP0R, FPΣ V3.20 or later, FP−X V2.50 or later, however, an operation error will occur not a syntax error if K or H constant is specified as S1 and S2.

Operation

- It is used to send commands to the serial port (COM1 or COM2) of the specified unit in the MEWTOCOL−COM mode connecting the unit that enables to receive the computer link command.
- The data is sent from the area of the remote unit specified by [S2] and [N], and is stored in the area of the local unit that starts with [D], according to the specification for the 2−word data stored in the control data that starts with the area specified by [S1].

Specifications for each item

The control data specified by [S1][S1+1] is specified as follows.

[S1]: Specifying transmission unit and transmission method

[S1]:

H0			
Word unit transmission	Specifies No. of transmission words H001 to H1FD (1 to 509 words): when transmitting to Group A H001 to H1B (1 to 27 words): when transmitting to Group B		

Group A	FPΣ, FP−X, FP0R, FP2, FP2SH, FP10SH
Group B	FP0, FP−e

[S1]:

H8		H0 fixed	
Bit unit transmission	Bit No. of local unit (H0 to HF)		Bit No. of remote unit (H0 to HF)

[S1+1]: Specifying the remote unit

[S1+1]:

	H0 fixed		
Selects COM port (H1 or H2)		Unit No. (H01 to H63) (1 to 99)	

(1) Specifying the transmission unit and transmission method [S1]
If data is to be sent in word units, specify the data volume, and if it is to be sent in bit units, specify the position of the target bit.

(2) Specifying the remote unit [S1+1]
Specify the remote unit with the unit number.
Specify either the COM1 or COM2 port from which data is transmitted to the remote unit.
Specify H1 if only one COM port is available.

(3) Specifying the area of the remote unit which is received by [S2] and [N].
Specify 0 for the device No. of [S2].
Specify the memory area of the remote unit in which the data to be transmitted is stored, specifying the type S2 and the address N in combination.

Example) [S2]:DT0, [N]:K100
↓
DT100

For the FP0R, FPΣ V3.20 or later, FP−X V2.50 or later, transmission can be performed without checking when DT0 or LD0 is specified for D, and H constant is specified for N.
(Example) In case of DT0 and HFFFF, it is possible to access DT63353. It is convenient to access the data registers of an eco−power meter KW8M.

(4) Specifying the area of the local unit by [D] in which the data to be received is stored
Specify the memory area of the local unit in which the data to be received is stored.

- The MEWTOCOL−COM command is created according to the operands specified by [S1],[S1+1], [S2], [D], and [N].

Flag conditions

Σ Error flag (R9007) : Turns on and stays on when

Σ Error flag (R9008) : Turns on for an instant when

- The control data of [S1] and [S1+1] is a value outside of the specified range.
- The number of words specified by S1 causes the area of S2 or D to be exceeded when word unit transmission is being used.
- [S2]+[N] exceeds the area of [S2].
- The operation mode for the target COM port is other than compute link.
- Word unit
  If [S2] is DT/LD, it turns on when [N] is not 0 to 32767.
  If [S2] is WX/WY/WR/WL/SV/EV, it turns on when [N] is not 0 to 9999.
- Bit unit
  It turns on when [S2] is not WX/WY/WR/WL.
  It turns on when [N] is not 0 to 999.
- The device No. of [S2] is not 0.
- The communication cassette has not been installed for the target COM port.

Precautions during programming

- Specify the computer link for the operation mode (system register setting) of the COM port to be used.
- It is not possible to execute multiple F145 (SEND) instructions and F146 (RECV) instructions for the same communication port simultaneously.
  The program should be set up so that these instructions are executed when the SEND/RECV execution enabled flag (R9044: COM1/R904A: COM2) is on.

R9044 (COM1)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled
R904A (COM2)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled

- The SEND instruction only requests that the data be sent, but the actual processing takes place when the ED instruction is executed.
  The SEND/RECV execution end flag (R9045: COM1/R904B: COM2) can be used to check whether or not the transmission has been completed.

R9045 (COM1)	0: Completed normally 1: Completed with error (The error code is stored in DT90124.)
DT90124 (COM1)	If the transmission has been completed with an error (R9045 is on), the contents of the error (error code) are stored.
R904B (COM2)	0: Completed normally 1: Completed with error (The error code is stored in DT90125.)
DT90125 (COM2)	If the transmission has been completed with an error (R904B is on), the contents of the error (error code) are stored.

For information on the contents of error codes, refer to the manual. If the error code is H73, a communication time−out error has occurred.
The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 2.5 ms), using the setting of system register 32.
The default value is set to 10 seconds.

Error code (HEX)	Description
73	Time−out: Waiting for response

- The F145 or F146 instruction cannot be executed if the target is a special internal relay (from R9000) or a special data register (from DT9000) or file register FL.

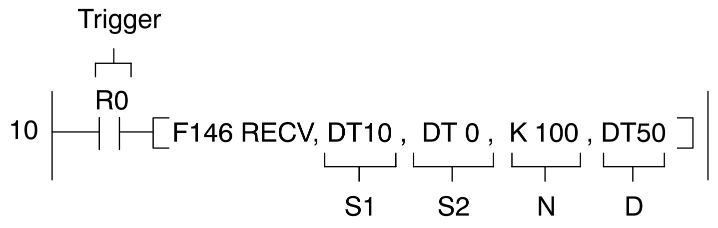
F146 (RECV) P146 (PRECV) Data receive (For MODBUS master mode)

Availability
FP0R/FP−X FPΣ: 32k

Outline Receives specified data from the serial port of another PLC or computer to the unit.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F146		(RECV)
	DT		10
	DT		0
	K		100
	DT		50

S1	Starting 16-bit area for storing control data
S2	16−bit area of destination to receive (The device No. is fixed at 0).
N	Starting address of the destination to receive.
D	Starting 16−bit area address for storing data received (destination data area at local unit).

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	N/A	A	A	N/A	N/A	A
S2	A	A	A	A	N/A	N/A	A	A	N/A	N/A	N/A	N/A	N/A	N/A
N	N/A	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	A (*2)	N/A	N/A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) I0 to ID

(*2) It can be specified only for the FP0R, FPΣ V3.20 or later, FP−X V2.50 or later, however, an operation error will occur not a syntax error if K or H constant is specified as S1 and S2.

Operation

- It is used to send commands to the serial port (COM1 or COM2) of the specified unit in the MODBUS mode connecting the unit that enables to receive the MODBUS command. (MODBUS command 01, 02, 03 and 04)
- The data is sent from the area of the remote unit specified by [S2] and [N], and is stored in the area of the local unit that starts with [D], according to the specification for the 2−word data stored in the control data that starts with the area specified by [S1].

Specifications for each item

The control data specified by [S1][S1+1] is specified as follows.

[S1]: Specifying transmission unit and transmission method

[S1]:	H0			
	Word unit transmission	Specifies No. of transmission words (H001 to H07F) *According to the restrictions on the MODBUS protocol.		

[S1]:	H8		H0 fixed	
	Bit unit transmission	Bit No. of local unit (H0 to HF)		Bit No. of remote unit (H0 to HF)

[S1+1]: Specifying the remote unit

[S1+1]:		H0 fixed		
	Selects COM port (H1 or H2)		Unit No. (H01 to H63) (1 to 99)	

(1) Specifying the transmission unit and transmission method [S1]
If data is to be sent in word units, specify the data volume, and if it is to be sent in bit units, specify the position of the target bit.
* For word units, the maximum of 127 (7Fh) words can be transmitted as the transmission range is up to 254 bytes.

(2) Specifying the remote unit [S1+1]
Specify the remote unit with the unit number.
Specify either the COM1 or COM2 port from which data is transmitted to the remote unit.
Specify H1 if only one COM port is available.

(3) Specifying the area of the remote unit which is received by [S2] and [N].
Specify 0 for the device No. of [S2].
Specify the memory area of the remote unit in which the data to be transmitted is stored, specifying the type S2 and the address N in combination.

Example) [S2]:DT0, [N]:K100
↓
DT100

(4) Specifying the area of the local unit by [D] in which the data to be received is stored
Specify the memory area of the local unit in which the data to be received is stored.

- The MODBUS command is created according to the operands specified by [S1],[S1+1], [S2], [D], and [N].
  When being transmitted in word units: The command 01 (Y, R coil read), command 02 (WL, LD read), command 03 (DT read) and command 04 (WL, LD read) can be transmitted.
  When being transmitted in bit units: The command 01 (Y, R coil read) and command 02 (X contact read) can be transmitted.
- The transmission is executed adding the 2 bytes of CRC at the end after the MODBUS command has been created.

Explanation of command

Command 01 (Y, R coil read) send

Example) When the 1 bit of Y17 is readed from the unit No. 17 of the remote unit, and a command to transmit the readed bit data to the 5th bit of the DT100 in the local unit is sent from the COM1.

[F146 (RECV), DT10, WY0, K1, DT100]

[S1]:	H8	H5	H0 fixed	H7	DT10
	Bit unit transmission	Bit No. of local unit (H0 to HF)		Bit No. of remote unit (H0 to HF)	

*Bit units (H8) should be specified for the transmission method of [S1] to read only 1bit of data by the command 01.

[S1+1]:	H1	H0 fixed	H1	H1	DT11
	Selects COM port (H1 or H2)		Unit No. (H01 to H63) (1 to 99)		

[S1]: DT10(DT10=8507H, DT11=1011H)
[S2]: WY0
[N] : K1
[D] : DT100

Command conversion

*Specify the coil No. of the destination for the starting No. of read. (Remote unit: Y17)
The quantity to read should be 1.

MODBUS commands

1	Slave address	11
2	Command (01H)	01
3	Starting No. of read (H)	00
4	Starting No. of read (L)	17
5	Quantity to read (H)	00
6	Quantity to read (L)	01
7	CRC16 (H)	DC
8	CRC16 (L)	59

Example) When the 64 bits (4 words) of data from Y10 to Y4F is readed from the unit No. 17 of the remote unit, and a command data to the area starting with DT100 in the local unit is sent from the COM1.

[F146 (RECV), DT10, WY0, K1, DT100]

[S1]:	H0	H0	H0	H4	DT10
	Word unit transmission	Specifies No. of transmission words (H001 to H07F)			

*Bit units (H0) should be specified for the transmission method of [S1] to read in word units by the command 01.

[S1+1]:	H1	H0 fixed	H1	H1	DT11
	Selects COM port (H1 or H2)		Unit No. (H01 to H63) (1 to 99)		

[S1]: DT10 (DT10=0004H, DT11=1011H)
[S2]: WY0
[N]: K1
[D]: DT100

Command conversion

* Specify the coil No. of the destination for the starting No. of read. (Remote unit: Y10)
The quantity to read should be the value of “No. of specified words X 16”. (64−bit read)

MODBUS commands

1	Slave address	11
2	Command (01H)	01
3	Starting No. of read (H)	00
4	Starting No. of read (L)	10
5	Quantity to read (H)	00
6	Quantity to read (L)	40
7	CRC16 (H)	3E
8	CRC16 (L)	AF

Command 02 (X contact read) send

Example) When the 1 bit of X17 is readed from the unit No. 17 of the remote unit, and a command to transmit the readed bit data to the 5th bit of DT100 in the local unit is sent.

[F146 (RECV), DT10, WX0, K1, DT100]

[S1]:	H8	H5	H0 fixed	H7	DT10
	Bit unit transmission	Bit No. of local unit (H0 to HF)		Bit No. of remote unit (H0 to HF)	

*Bit units (H8) should be specified for the transmission method of the [S1] to read only 1bit of data by the command 02.

[S1+1]:	H1	H0 fixed	H1	H1	DT11
	Selects COM port (H1 or H2)		Unit No. (H01 to H63) (1 to 99)		

[S1]: DT10(DT10=8507H, DT11=1011H)
[S2]: WX0
[N] : K1
[D] : DT100

Command conversion

*Specify the coil No. of the destination for the starting No. of read. (Remote unit: X17)
The quantity to read should be 1.

MODBUS commands

1	Slave address	11
2	Command (02H)	02
3	Starting No. of read (H)	00
4	Starting No. of read (L)	17
5	Quantity to read (H)	00
6	Quantity to read (L)	01
7	CRC16 (H)	0B
8	CRC16 (L)	5E

Example) When the 64 bits (4 words) of data from X10 to X4F is readed from the unit No. 17 of the remote unit, and a command data to the area starting with DT100 in the local unit is sent from the COM1.

[F146 (RECV), DT10, WX0, K1, DT100]

[S1]:	H0	H0	H0	H4	DT10
	Word unit transmission	Specifies No. of transmission words (H001 to H07F)			

*Bit units (H0) should be specified for the transmission method of [S1] to read in word units by the command 02.

[S1+1]:	H1	H0 fixed	H1	H1	DT11
	Selects COM port (H1 or H2)		Unit No. (H01 to H63) (1 to 99)		

[S1]: DT10 (DT10=0004H, DT11=1011H)
[S2]: WX0
[N]: K1
[D]: DT100

Command conversion

*Specify the coil No. of the destination for the starting No. of read. (Remote unit: X10)
The quantity to read should be the value of "No. of specified words X 16". (64-bit read)

MODBUS commands

1	Slave address	11
2	Command (02H)	02
3	Starting No. of read (H)	00
4	Starting No. of read (L)	10
5	Quantity to read (H)	00
6	Quantity to read (L)	40
7	CRC16 (H)	7A
8	CRC16 (L)	A0

Command 03 (DT read) send

Example) When the 6 words of data from DT500 to DT505 is readed from the unit No. 17 of the remote unit, and a command data to the area starting with DT100 in the local unit is sent from the COM1.

[F146 (RECV), DT10, DT0, K500, DT100]

[S1]:	H0	H0	H0	H6	DT10

Word unit transmission — Specifies No. of transmission words (H001 to H07F)

*Word units (H0) should be specified for the transmission method of [S1] to read in word units by the command 03.

[S1+1]:	H1	H0 fixed	H1	H1	DT11

Selects COM port (H1 or H2) — Unit No. (H01 to H63) (1 to 99)

[S1]: DT10 (DT10=0006H, DT11=1011H)
[S2]: DT0
[N]: K500
[D]: DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (03H)	03
3	Starting No. of read (H)	01
4	Starting No. of read (L)	F4
5	Quantity to read (H)	00
6	Quantity to read (L)	06
7	CRC16 (H)	87
8	CRC16 (L)	56

* Specify the data No. of the destination for the starting No. of read. (Remote unit: DT500)
The quantity to read should be the No. of specified words. (6−word read)

Command 04 (WL, LD read) send

Example) When the 6 words of data from WL20 to WL25 is readed from the unit No. 17 of the remote unit, and a command data to the area starting with DT100 in the local unit is sent from the COM1.

[F146 (RECV), DT10, WL0, K20, DT100]

[S1]:	H0	H0	H0	H6	DT10

Word unit transmission — Specifies No. of transmission words (H001 to H07F)

*Word units (H0) should be specified for the transmission method of [S1] to read in word units by the command 04.

[S1+1]:	H1	H0 fixed	H1	H1	DT11

Selects COM port (H1 or H2) — Unit No. (H01 to H63) (1 to 99)

[S1]: DT10 (DT10=0006H, DT11=1011H)
[S2]: WL0
[N]: K20
[D]: DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (04H)	04
3	Starting No. of read (H)	00
4	Starting No. of read (L)	14
5	Quantity to read (H)	00
6	Quantity to read (L)	06
7	CRC16 (H)	32
8	CRC16 (L)	9C

* Specify the data No. of the destination for the starting No. of read. (Remote unit: WL20)
The quantity to read should be the No. of specified words. (6−word read)

Example) When the 6 words of data from LD100 to LD105 is readed from the unit No. 17 of the remote unit, and a command data to the area starting with DT100 in the local unit is sent from the COM1.

[F146 (RECV), DT10, LD0, K100, DT100]

[S1]:

H0	H0	H0	H6	DT10

$\underbrace{\text{H0}}_{\text{Word unit transmission}}$ $\underbrace{\text{H0 H0 H6}}_{\text{Specifies No. of transmission words (H001 to H07F)}}$

*Word units (H0) should be specified for the transmission method of [S1] to read in word units by the command 04.

[S1+1]:

H1	H0 fixed	H1	H1	DT11

$\underbrace{\text{H1}}_{\text{Selects COM port (H1 or H2)}}$ $\underbrace{\text{H1 H1}}_{\text{Unit No. (H01 to H63) (1 to 99)}}$

[S1]: DT10 (DT10=0006H, DT11=1011H)
[S2]: LD0
[N]: K100
[D]: DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (04H)	04
3	Starting No. of read (H)	08
4	Starting No. of read (L)	34
5	Quantity to read (H)	00
6	Quantity to read (L)	06
7	CRC16 (H)	31
8	CRC16 (L)	36

* Specify the data No. of the destination for the starting No. of read. (Remote unit: LD100)
The quantity to read should be the No. of specified words. (6−word read)

*For specifying LD, it should be from 07D0H (LD0).

Flag conditions

Σ Error flag (R9007) : Turns on and stays on when

Σ Error flag (R9008) : Turns on for an instant when

- The control data of [S1] and [S1+1] is a value outside of the specified range.
- The number of words specified by S1 causes the area of S2 or D to be exceeded when word unit transmission is being used.
- [S2]+[N] exceeds the area of [S2].
- The MODBUS mode has not been specified for the COM port of the control data specified by [S1+1].
- The area of [S2] is DT, WL and LD in the bit unit transmission.
- The device No. of [S2] is not 0.

Precautions during programming

- It is not possible to execute multiple F145 (SEND) instructions and F146 (RECV) instructions for the same communication port simultaneously.
  The program should be set up so that these instructions are executed when the SEND/RECV execution enabled flag (R9044: COM1/R904A: COM2) is on.

R9044 (COM1)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled
R904A (COM2)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled

- The SEND instruction only requests that the data be sent, but the actual processing takes place when the ED instruction is executed.
  The SEND/RECV execution end flag (R9045: COM1/R904B: COM2) can be used to check whether or not the transmission has been completed.

R9045 (COM1)	0: Completed normally 1: Completed with error (The error code is stored in DT90124.)
DT90124 (COM1)	If the transmission has been completed with an error (R9045 is on), the contents of the error (error code) are stored.
R904B (COM2)	0: Completed normally 1: Completed with error (The error code is stored in DT90125.)
DT90125 (COM2)	If the transmission has been completed with an error (R904B is on), the contents of the error (error code) are stored.

For information on the contents of error codes, refer to the manual. If the error code is H73, a communication time−out error has occurred.
The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 2.5 ms), using the setting of system register 32.
The default value is set to 10 seconds.

Error code (HEX)	Description
73	Time−out: Waiting for response

- The F145 or F146 instruction cannot be executed if the target is a special internal relay (from R9000) or a special data register (from DT9000).

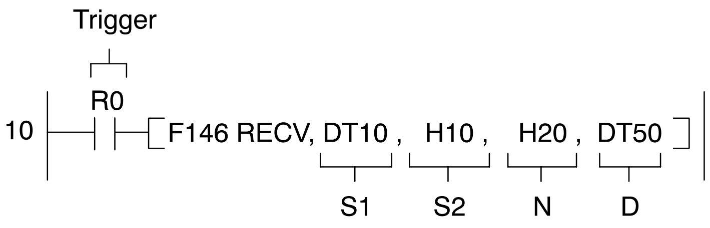
F146 (RECV) Data receive (MODBUS master mode II: Type directly specifying MODBUS address)

Availability
FP0R FP−X: Ver. 2.50 FPΣ: Ver. 3.20

Outline Receives specified data from the serial port of another PLC or computer to the unit.
Feature: Data can be transmitted with this instruction only.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F146		(RECV)
	DT		10
	H		10
	H		20
	DT		50

S1	Starting 16−bit area for storing control data
S2	Specification of MODBUS address
N	No. of received data
D	Starting 16−bit area address for storing data received

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	N/A	A	A	A	A	A
S2	A	A	A	A	N/A	N/A	A	A	N/A	N/A	N/A	A	A	N/A
N	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) I0 to ID

(*2) This instruction is available only for FP0R/FPΣ V3.20 or later/FP−X V.250 or later.

Description

The data of the volume specified by [N] is received form the MODBUS address specified by [S2] with the specification of the transmission port, transmission command (1 or 2) and destination unit number, and stored in the operation memory specified by [D].
MODBUS commands are transmitted. (MODBUS commands 01, 02, 03 and 04)

The feature is that data can be transmitted with this instruction only.

Specifications for each item

[S1]: Specifying port number, transmission command and destination unit number

[S1]:

H1 or H2	H1 H2 H3 H4		

Selects COM port | Specifies transmission command | Unit No. (H01 to HFF)

(1) Specifying COM port
Specify H1 for COM 1 port, and H2 for COM2 port.
If only one COM port is available, specify H1.

(2) Specifying transmission command.
Any one of H1, H2, H3 and H4 can be specified.

(3) Specifying destination unit number
Numbers in the range of H1 to HFF can be specified.

- Specifying the MODBUS address of the destination unit where data is transmitted by [S2].
  Settable address: H0 to HFFFF
- Specifying the number of data received by [N]
  Settable number of data
  For bit data: Max. 2040 (07F8H)
  For word data: Max. 127 (7FH)
- Specifying the area by [D] in which the data to be received is stored.
  Specify the starting number of the operation memory of the destination unit in which the data to be received is stored.
  This instruction can be executed even if the transmission command specified by [S1] and the device type in the destination unit differs.
  That means the contents of DT can be transmitted by specifying the bit data, or the contents of WR can be transmitted by specifying the word data.
  When the command 1 or 2 is specified, data is always stored from the bit 0 of [D].

Explanation of command

Command 01 (Coil status read)

Example) When 1 bit is read from the bit address H7788 of the unit No. 17 connected to the COM1 and written in the bit of DT100 of the local unit.

[F146(RECV), H1111,H7788, K1, DT100]

[S1]:

H1	H1	H1	H1

Selects COM port | Unit No. (H01 to HFF)

[S1]:H1111
[S2]:H7788
[N] :K1
[D] :DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (01H)	01
3	Starting No. of read (H)	77
4	Starting No. of read (L)	88
5	Quantity to read (H)	00
6	Quantity to read (L)	01
7	CRC16 (H)	64
8	CRC16 (L)	C4

Example) When 64 bits (4 words) are read from the bit address H7788 of the unit No. 17 connected to the COM1 and written in the bit 0 of DT100 of the local unit.

[F146(RECV), H1111,H7788, K64, DT100]

[S1]:	H1	H1	H1	H1
	Selects COM port		Unit No. (H01 to HFF)	

[S1]:H1111
[S2]:H7788
[N] :K64
[D] :DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (01H)	01
3	Starting No. of read (H)	77
4	Starting No. of read (L)	88
5	Quantity to read (H)	00
6	Quantity to read (L)	40
7	CRC16 (H)	A4
8	CRC16 (L)	F4

Command 02 (Input status read)

Example) When 1 bit is read from the bit address H7788 of the unit No. 17 connected to the COM1 and written in the bit of DT100 of the local unit.

[F146(RECV), H1211,H7788, K1, DT100]

[S1]:	H1	H1	H1	H1
	Selects COM port		Unit No. (H01 to HFF)	

[S1]:H1211
[S2]:H7788
[N] :K1
[D] :DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (02H)	02
3	Starting No. of read (H)	77
4	Starting No. of read (L)	88
5	Quantity to read (H)	00
6	Quantity to read (L)	01
7	CRC16 (H)	20
8	CRC16 (L)	C4

Example) When 64 bits (4 words) are read from the bit address H7788 of the unit No. 17 connected to the COM1 and written in the bit 0 of DT100 of the local unit.

[F146(RECV), H1211,H7788, K64, DT100]

[S1]:	H1	H1	H1	H1
	Selects COM port		Unit No. (H01 to HFF)	

[S1]:H1211
[S2]:H7788
[N] :K64
[D] :DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (02H)	02
3	Starting No. of read (H)	77
4	Starting No. of read (L)	88
5	Quantity to read (H)	00
6	Quantity to read (L)	40
7	CRC16 (H)	E0
8	CRC16 (L)	F4

Command 03 (Holding register read)

Example) When 6 words are read from the address H7788 of the unit No. 17 connected to the COM1 and written in the area starting with DT100 in the local unit.

[F146(RECV), H1311,H7788, K6, DT100]

[S1]:	H1	H3	H1	H1
	Selects COM port		Unit No. (H01 to HFF)	

[S1]:H1311
[S2]:H7788
[N] :K6
[D] :DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (03H)	03
3	Starting No. of read (H)	77
4	Starting No. of read (L)	88
5	Quantity to read (H)	00
6	Quantity to read (L)	06
7	CRC16 (H)	5C
8	CRC16 (L)	C6

Command 04 (Input register read)

Example) When 6 words are read from the address H7788 of the unit No. 17 connected to the COM1 and written in the area starting with DT100 in the local unit.

[F146(RECV), H1411,H7788, K6, DT100]

[S1]:H1411
[S2]:H7788
[N] :K6
[D] :DT100

Command conversion

MODBUS commands

1	Slave address	11
2	Command (04H)	04
3	Starting No. of read (H)	77
4	Starting No. of read (L)	88
5	Quantity to read (H)	00
6	Quantity to read (L)	06
7	CRC16 (H)	E9
8	CRC16 (L)	06

Flag conditions

Σ Error flag (R9007) :

Σ Error flag (R9008) :

- Turns on when the control data of [S1] is a value outside of the specified range.
- Turns on when the MODBUS mode has not been specified for the COM port of the control data specified by [S1].
- Turns on when the number of received data N is 0.
- Turns on when the number of received data is negative.
- Turns on when the number of received data [N] exceeds the limit of the MODBUS specifications.
- Turns on when the number of received data [N] exceeds the operation memory area specified by [D].

Precautions during programming

- It is not possible to execute multiple F145 (SEND) instructions and F146 (RECV) instructions for the same communication port simultaneously.
  The program should be set up so that these instructions are executed when the SEND/RECV execution flag (R9044: COM1/R904A: COM2) is on.

R9044 (COM1)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled
R904A (COM2)	0: Execution inhibited (SEND/RECV instruction being executed) 1: Execution enabled

- The SEND instruction only requests that the data be sent, but the actual processing takes place when the ED instruction is executed.
  The SEND/RECV execution end flag (R9045: COM1/R904B: COM2) can be used to check whether or not the transmission has been completed.

R9045 (COM1)	0: Completed normally 1: Completed with error (The error code is stored in DT90124.)
DT90124 (COM1)	If the transmission has been completed with an error (R9045 is on), the contents of the error (error code) are stored.
R904B (COM2)	0: Completed normally 1: Completed with error (The error code is stored in DT90125.)
DT90125 (COM2)	If the transmission has been completed with an error (R904B is on), the contents of the error (error code) are stored.

If the error code is H73, a communication time−out error has occurred.
The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 10 ms), using the setting of system register 32.
The default value is set to 10 seconds.

Error code (HEX)	Description
73	Time−out: Waiting for response

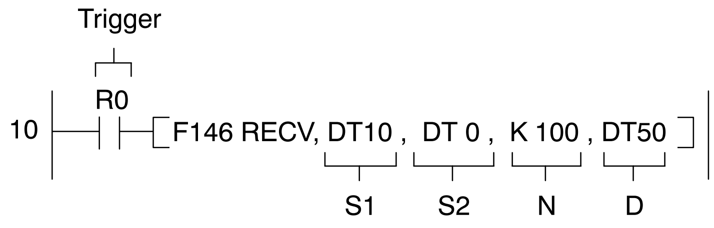
- For global transmission (the transmission performed by specifying H00 for the unit No.), the program should be set up so that the transmission is executed after a time of at least the maximum scan time elapsed.
- The F145 or F146 instruction cannot be executed if the target is a special internal relay (from R9000) on a special data register (DT90000).

F146 (RECV) P146 (PRECV)

Data receive (MEWNET link)

Outline Receives data from another station through link units in the network.

Program example



Boolean Address	Boolean Instruction
10	ST R 0
11	F146 (RECV)
	DT 10
	DT 0
	K 100
	DT 50

S1	Starting 16-bit area for storing control data
S2	Type of source operands for storing data in the remote station. Be sure to select the area by setting address 0 (source data area at another station).
N	Starting 16-bit area address for the source operand specified in S2 above (source data area at another station).
D	Starting 16-bit area address for storing data received (destination data area at local station).

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A
N	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

Explanation of example

1 Example of word unit reception

When the control data is as follows:

DT10(S1)=H0005 (=K5)

Word unit 5 words

DT11(S1+1)=H010A

Unit No.10
Route No.1

the data from DT100 to DT104 of the unit No. 10 connected to route No. 1 is sent to DT50 to DT54 of the local station when the execution condition (trigger) R0 turns on.

② Example of bit unit reception

When the control data is as follows:

DT10(S1)=H850D

Bit unit

Bit No. 13 of remote station's memory area

Bit No. 5 of local station's memory area

DT11(S1+1)=H010A

Unit No.10

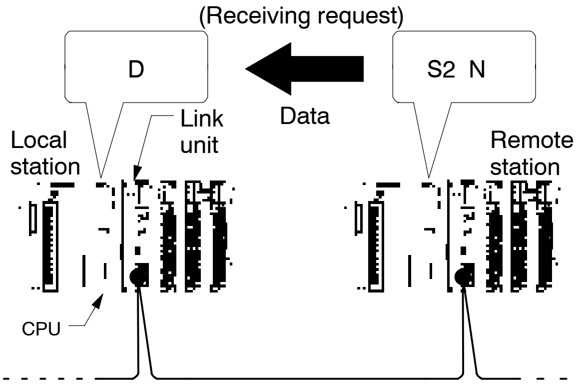
Route No.1

the on and off information of Bit No. 13 of DT100 of the unit No. 10 connected to route No. 1 is sent to Bit No. 5 of DT50 when the execution condition (trigger) R0 turns on.

Description

This reads the data in the area specified by S2 and N of a remote station connection with the MEWNET−W, MEWNET−P, MEWNET−H, and stores it in the area specified by D of the local station.

The remote stations (routes and unit numbers), the transmission unit (bit unit or word unit), the transmission method, and other parameters are specified by the control data S1.



The remote station is specified by S1.

If general−purpose communication through the COM. port of the FP2, FP2SH, and FP10SH is being used, **F144 (TRNS)** instruction is used instead of this instruction.

Refer to the section describing the **F144 (TRNS)** instruction.

Specifying the various items

Control data (S1)

Specifying the remote station
Specify the remote station by means of a route number and unit number.
The setting is entered differently depending on whether the remote station is a PLC in the same network, or a PLC in a network on a different hierarchical level.

Specifying the transmission unit and transmission method
If data is to be received in word units, specify the data volume, and if it is to be received in bit units, specify the position of the target bit.

Specifying the memory area of the remote station (S2) and (N)

Specify the memory area of the remote station in which the data being received is to be stored, specifying the type S2 and the address N in combination.

Example: S2: DT10, N: K100

↓

DT100

Specifying the memory area of the local station (D)

Specify the memory area of the local station in which the data received from the remote station is to be stored.

Flag conditions

Σ Error flag (R9007): Turns on and stays on when:

Σ Error flag (R9008): Turns on for an instant when:

- The control data is a value outside of the specified range.
- Tthe remote station does not exist.
- The number of words specified by S1 causes the area of S2 or D to be exceeded when word unit reception is being used.

Receiving from a PLC within the same network

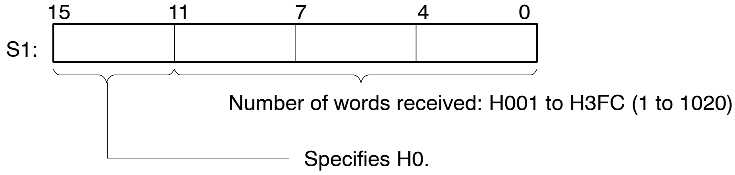
Specifying the control data (S1+1, S1)

The control data should be specified as an H constant. The transmission unit, transmission method and other parameters are specified with S1, and the remote station is specified with S1+1.

S1+1	S1
Specifying the remote station	Specifying the transmission unit and method

(1) Specifying word unit reception

If word unit reception is being used, the data for the specified number of words is sent from the memory area of the remote station specified by S2 and N, and is stored in the memory area of the local station that starts with D. If only the MEWNET−H network is being used, up to 1,020 words can be received at one time, and if the network is using the MEWNET−P/W, up to 16 words can be received at one time.

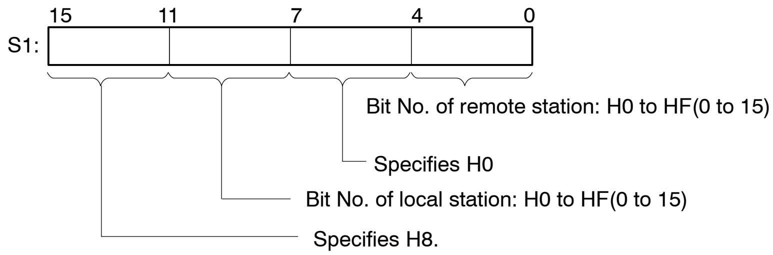


Example:

If 10 words of data are being received, K10(H000A) should be specified in S1.

(2) Specifying bit unit reception

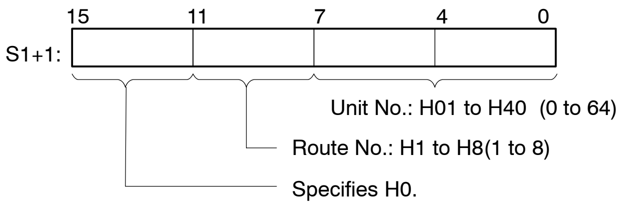
When data is being sent in bit units, the information for the specified bit of the memory area of the remote station specified by S2 and N is stored in the specified bit of the memory area of the local station specified by D.



Example:

If the data from Bit No. 0 of the memory area in the remote station is being sent to Bit No. 15 of the local station memory area, H8F00 should be specified in S1.

(3) Specifying the remote station (common to both word/bit transmission)



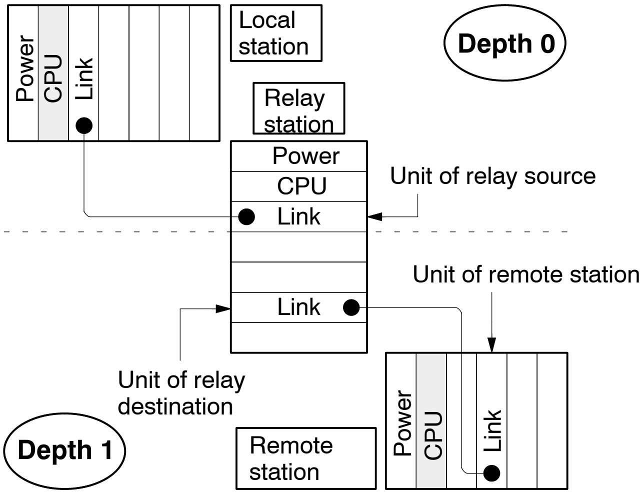
The unit number should be converted to a hexadecimal number and specified.
– For MEWNET−W: H01 to H20 (1 to 32)
– For MEWNET−P: H01 to H3F (1 to 63)
– For MEWNET−H: H01 to H40 (1 to 64)

Sending data from a PLC on a different hierarchical level

What is a hierarchical link?

A hierarchical link functions as a relay station between two link units installed on the same backplane, enabling communication between CPUs belonging to different networks.

Example: Communicating with a CPU at depth 1

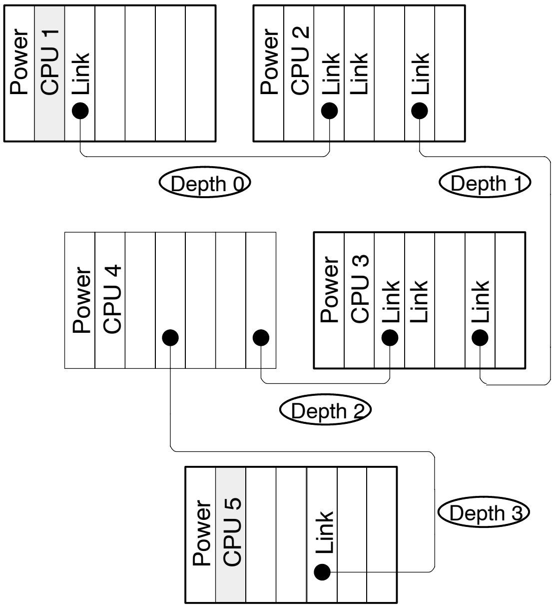


In this way, by passing data through a relay station, communication is possible to a depth of 3.

Note

When using the MEWNET−P and MEWNET−W, data can only be relayed one network deeper in the hierarchy.

Example: **Communicating with a CPU at depth 3 (reception from CPU5 to CPU1)**



The numbers CPU1 to CPU5 have been temporarily assigned, for the purpose of indicating the relay order of the hierarchical links.

Specifying the control data (S1)

The control data should be specified as an H constant.

The transmission unit, transmission method and other parameters are specified with S1, and the remote station is specified with S1+1 and subsequent parameters (the relay source unit, relay destination unit, and unit targeted for communication). (depth + 3) words are required.

Example: **Control data when specifying a remote station which is at depth 3**

S1	Specifying the transmission unit and method			
[S1+1]	Local station	Depth (H03)	CPU1	Specifying the remote station
[S1+2]	Relay source	Relay destination	CPU2	
[S1+3]	Relay source	Relay destination	CPU3	
[S1+4]	Relay source	Relay destination	CPU4	
[S1+5]	Remote station	H00	CPU5	

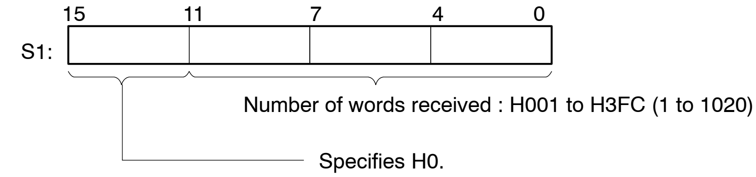
————: Same network

−−−−−−−: Same backplane

The relay source is specified by a unit No. in the network, and the relay destination is specified by a route number on the backplane.

(1) Specifying word unit reception

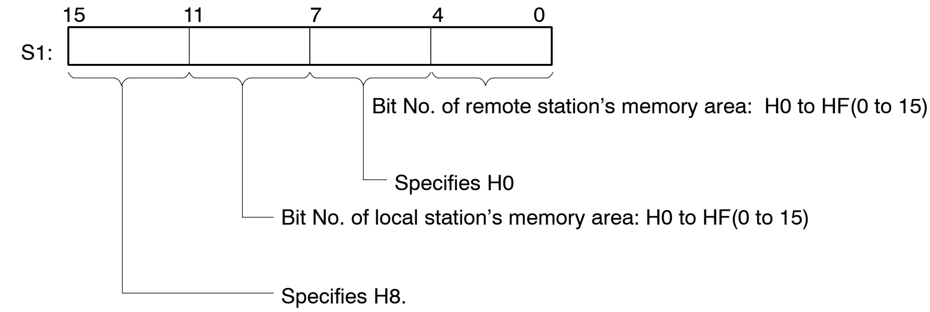
If word unit reception is being used, the data for the specified number of words is sent from the memory area of the remote station specified by S2 and N, and is stored in the memory area of the local station beginning with D. If only the MEWNET−H network is being used, up to 1,020 words can be received at one time, and if the network is using the MEWNET−P and MEWNET−W, up to 16 words can be received at one time.



Example: If 10 words of data are being received, K10(H000A) should be specified in S1.

(2) Specifying bit unit reception

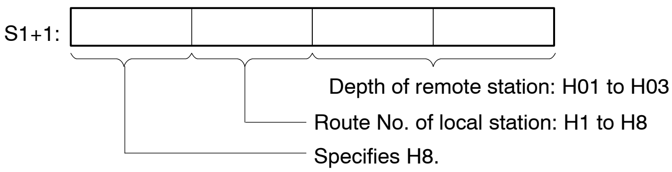
When data is being sent in bit units, the information for the specified bit of the memory area of the remote station specified by S2 and N is stored in the specified bit of the memory area of the local station specified by D.



Example: If the data from Bit No. 0 of the memory area in the remote station is being sent to Bit 15 of the local station memory area, H8F00 should be specified in S1.

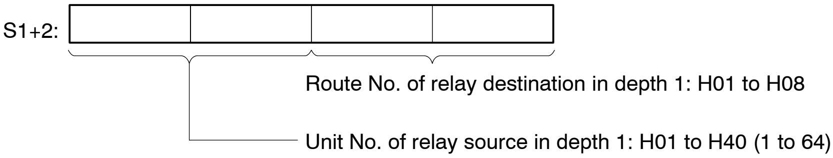
(3) Specifying the remote station (common to both word/bit transmission)

① Specifying the route No. and depth



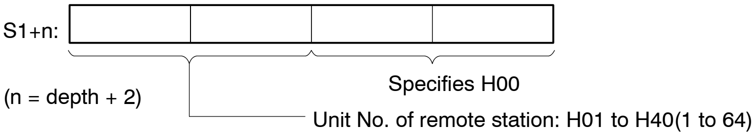
② Specifying the relay station

S+1 should be used to specify only the specified amount of depth, while (S1+3) is used to specify depth 2 for the same item, and (S1+4) is used to specify depth 3.



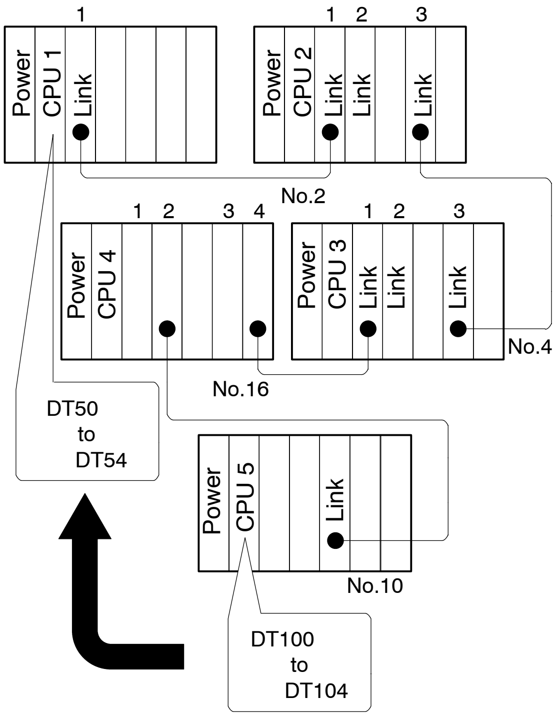
③ Specifying the remote station

This should be specified right after the specification of the relay station.

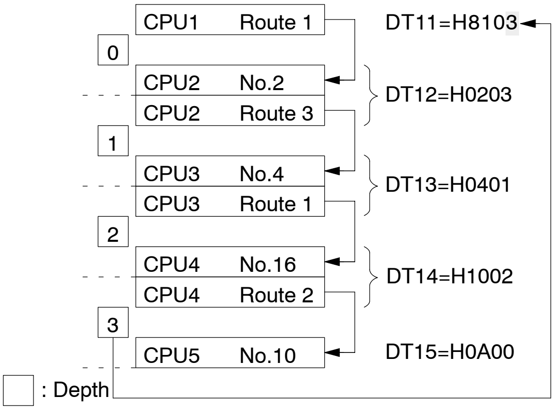


Example: When using the program example shown on page 3 – 376. In this example, the data from DT100 to DT104 of the CPU (CPU5) is received in DT50 to DT54 of the local station (CPU1) shown below.

Connection diagram



In this example, the control data beginning with DT10 (depth 3 → 6 words) should be specified as shown below. To receive the 5 words of data → DT10 = H0005



Precautions during programming

It is not possible to execute multiple **F145 (SEND)** instructions and **F146 (RECV)** instructions at the same time.

The program should be set up so that these instructions are executed when the MEWNET send/receive execution enabled flag (R9030) is on.

R9030	0: Execution inhibited **F145 (SEND)**/**F146 (RECV)** instruction being executed) 1: Execution enabled

The **F146 (RECV)** instruction only requests that the data be received, but the actual processing takes place when the **ED** instruction is executed. The MEWNET send/receive completed flag (R9031) can be used to check whether or not the reception has been completed.

R9031	0: Completed normally 1: Completed with error (The error code is stored in DT9039.)
DT9039 (DT90039)	If the transmission has been completed with an error (R9031 is on), the contents of the error (error code) are stored.

For information on the contents of error codes, refer to the manual for that particular link unit. If the error code is H71 to H73, a communication time−out error has occurred. The time−out time can be changed within a range of 10.0 ms to 81.9 seconds (in units of 10 ms), using the setting of system register 32. The default value is set to 10 seconds for FP2/FP2SH/FP10SH.

Error code (HEX)	Description
H71	Time out: Waiting for transmission answer
H72	Time−out: Waiting for transmission buffer to be emptied
H73	Time−out: Waiting for response

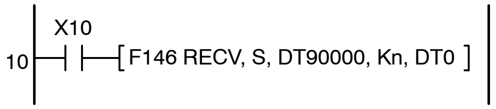
The **F146 (RECV)** instruction cannot be executed if the target is a special internal relay (from R9000) or a special data register (from DT9000/DT90000).

Additional information concerning the F146 (RECV) instruction

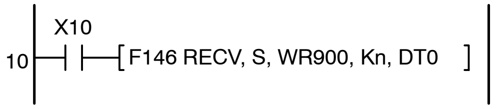
Receiving the special data registers and special internal relays using the data transfer instruction

Special data registers and special internal relays cannot be transferred using the **F146 (RECV)** instruction. Use a program like that shown below to receive these types of data.

Receiving special data registers in the FP2 , FP2SH or FP10SH (source issuing the command: FP2/FP2SH/FP10SH)



Receiving special internal relays (source issuing the command: FP2/FP2SH/FP10SH)



How to receive FL (How to specify FL banks)

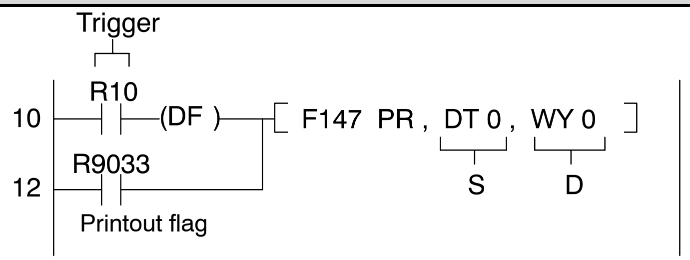
- How to specify the FL bank for destination units
  The FL to communicate is specified like FL0 + H10. Specify FL1 + H10 to specify the FL of the bank 1 (FL2 + H10 for the bank 2)
- How to specify the FL bank for a local unit
  Normally, FLn is specified for the FL for the local unit. If specifying, the FL will be received at the FL bank which has been selected in the execution of this instruction.

F147 (PR) Printout

Outline Outputs ASCII codes to the printer (for transistor output type only).

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	DF		
12	OR	R	9033
14	F147		(PR)
	DT		0
	WY		0

S	Starting 16-bit area for storing 12 bytes (6 words) of ASCII codes (source)
D	Word external output relay used for output of ASCII codes (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL (*1)	SV	EV	DT	LD (*1)	FL (*2)	I	K	H	
S	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A
D	N/A	A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

(*1) This cannot be used with the FP0 and FP−e.

(*2) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.

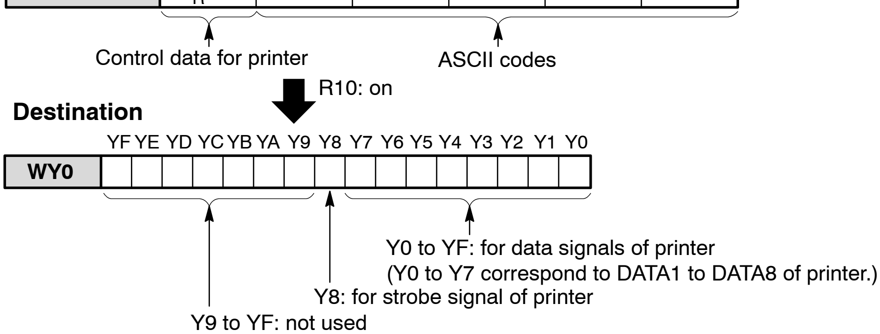
A: Available
N/A: Not Available

Explanation of example

The ASCII codes stored in data registers DT0 to DT5 are output through word external output relay WY0 when trigger R10 turns on.

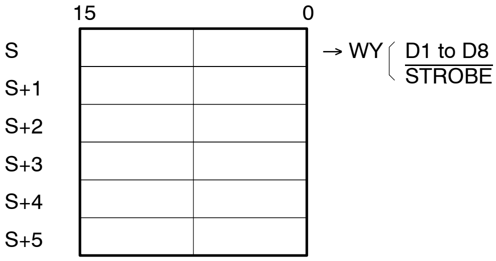
Source: ASCII code for 12 character A, B, C, D, E, F, G, H, I and J

Data register	DT5		DT4		DT3		DT2		DT1		DT0	
ASCII HEX code	0D	0A	4A	49	48	47	46	45	44	43	42	41
ASCII character	C_R	LF	J	I	H	G	F	E	D	C	B	A



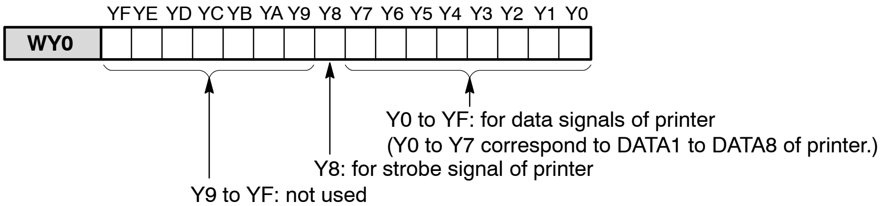
Description

Outputs the ASCII codes for 12 characters stored in the 6-word area specified by S through the word external output relay WY specified by D.



If the specified output is connected to a commercial printer, the characters corresponding to the output ASCII code are printed.

Only bit positions 0 to 8 of WY are used in the actual printout.



ASCII code is output in order starting from the lower byte of the starting area.

Be sure to set the control code (LF and C_R) for the printer as the final word of the data.

Three scans are required for 1 character constant output. Therefore, 37 scans are required until 12 character constants are output. (See “Time chart”)

Precautions during programming

Multiple **F147 (PR)** instructions cannot be executed at the same time. The program should be set up so that the printout flag (R9033) is used during execution of **F147 (PR)** instruction to inhibit simultaneous execution.

The ASCII code conversion instruction [**F95 (ASC)**] can be used to convert character constants (M) to ASCII codes.

Character constants (M) can be input only with programming tool software.

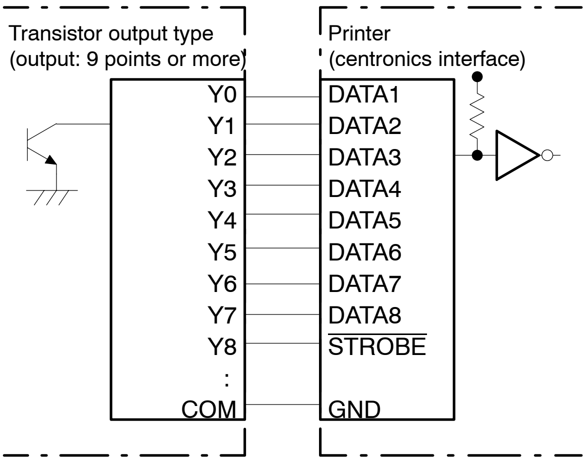
A transistor−type output unit/board is necessary.

When this instruction is executed, of the WY area specified by D, zero <off> is set for Y_9 to Y_F.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The ending area for storing ASCII codes exceeds the limit.
  - The trigger of another **F147 (PR)** instruction turns on while one **F147 (PR)** instruction is being executed.
- Printout flag (R9033): Turns on and stays on while a **F147 (PR)** instruction is being executed.

Connection example



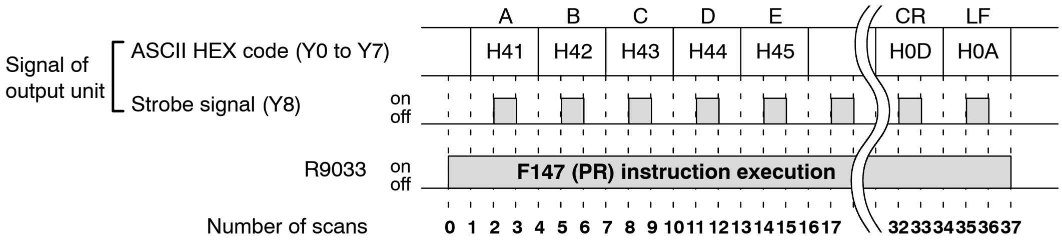
Data setting

Example: ASCII code for 10 character A, B, C, D, E, F, G, H, I and J

Data register	DT5		DT4		DT3		DT2		DT1		DT0	
ASCII HEX code	0D	0A	4A	49	48	47	46	45	44	43	42	41
ASCII character	CR	LF	J	I	H	G	F	E	D	C	B	A

Control data for printer (CR LF) — ASCII codes (J to A)

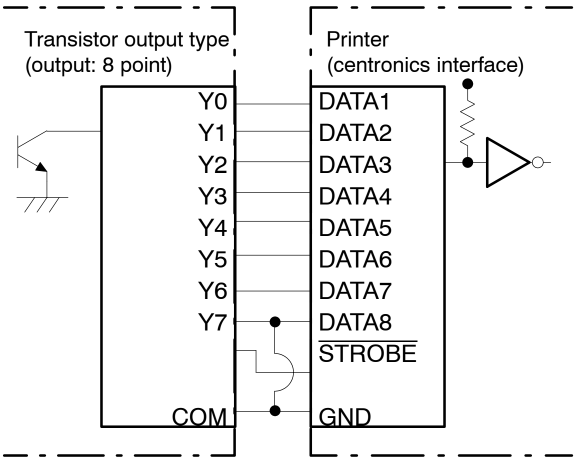
Time chart



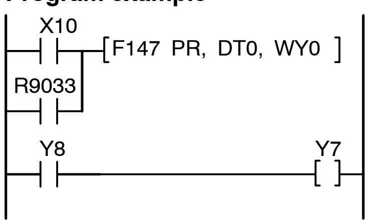
Using printer output during 8−point output

When only eight output points are being used, connections should be made as shown below, and the program should be set up so that the strobe signal is output from Y7.

Connection example



Program example



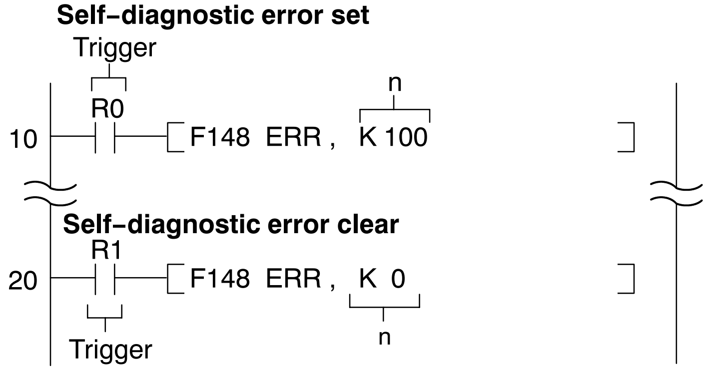
Strobe signal is output from Y7.

F148 (ERR) P148 (PERR) Self-diagnostic error set

Outline Sets the specified condition as a self-diagnostic error.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P148 (PERR)" is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction
10	ST R 0
11	F148 (ERR)
	K 100
	⋮
20	ST R 1
21	F148 (ERR)
	K 0

n	Self-diagnostic error code number Range: 0 and 100 to 299

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Explanation of example

The self−diagnosis error 100 is set when the execution condition (trigger) R0 turns on. For FP0/FP−e/FP0R/FPΣ/FP−X, the ERROR (ERROR/ALARM) LED on the control unit blinks and for FP2/FP2SH/FP10SH, ERROR LED on CPU lights, and operation stops.
(If a situation occurs in which you wish to set the self−diagnosis error 100, set up the program so that input R0 turns on.)

When the execution condition (trigger) R1 turns on, self−diagnostic errors of error codes 43 and higher are cleared.

Description

Along with self−diagnostic error codes specified by n being stored in the special data register DT9000 on DT90000, the self−diagnostic error flag (R9000) is turned on. Also, for FP0/FP−e/FP0R/FPΣ/FP−X, the ERROR/ALARM on the control unit blinks and for FP2/FP2SH/FP10SH, ERROR LED on the CPU lights.

The specified value “n” is what determines whether operation stops or continues when the instruction is executed.

“n” setting	Operation when error occurs
K100 to K199	Operation stops
K200 to K299	Operation continues

If “n” is set to a value between K200 and K299, if several **F148 (ERR)** instructions are processed at one time, codes are received in sequential order, starting with the lowest number.

If “n” is set to 0 and the **F148 (ERR)** instruction is executed, self−diagnostic errors with error codes of 43 and higher are cleared.

– For FP0/FP−e/FP0R/FPΣ/FP−X, ERROR/ALARM LED: turned off
– For FP2/FP2SH/FP10SH, ERROR LED: turned off
– R9000, R9005, R9006, R9007, R9008: off
– DT9000, DT9017, DT9018: Cleared to 0
  DT90000, DT90017, DT90018: Cleared to 0

F148 (ERR) instructions which specify the same error code can be notated in duplicate in the program.

Confirmation of self−diagnostic error

Self−diagnostic errors are checked in the normal way.

FP0 C10, C14, C16,C32/FP−e	FP0 T32/FP0R/FPΣ/FP−X/ FP2/FP2SH/FP10SH
DT9000	DT90000
DT9017	DT90017
DT9018	DT90018

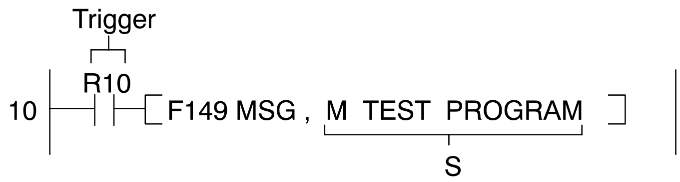
Flag conditions

- Error flag (R9007): Turns on and stays on when the value of n exceeds the limit of specified range “K0, or K100 to K299.”
- Error flag (R9008): Turns on and stays on when the value of n exceeds the limit of specified range “K0, or K100 to K299.”

F149 (MSG) P149 (PMSG) Message display

Outline Displays the message “specified character constant” on the programming tool.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P149 (PMSG)” is not available.

Program example



Boolean Address	Boolean Instruction
10	ST R 10
11	F149 (MSG)
	M TEST PROGRAM

S	Character constant for message

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX	IY	K	H	M	
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

Displays the message “TEST PROGRAM” on the programming tool when trigger R10 turns on.

Description

This instruction is used for displaying message specified by S on the programming tool.

The character constants (M) can be input only with programming tool software.

When the **F149 (MSG)** instruction is executed, the message flag (R9026) turns on and the message specified by S is set in special data registers DT9030 to DT9035/DT90030 to DT90035.

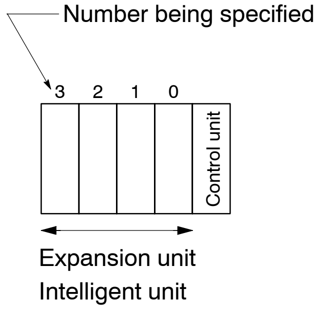
Type	Special data register
FP0 C10, C14, C16, C32/FP−e	DT9030 to DT9035
FP0 T32/FP0R/FPΣ/FP−X/ FP2/FP2SH/FP10SH	DT90030 to DT90035

Once the message is set in the special data registers, the message cannot be changed even if the **F149 (MSG)** instruction is executed again.
To clear the message in the special data registers, click on the “Cancel” button on “Display PLC Message” screen using the programming tool software.

Specifying Slot Numbers

With the FPΣ

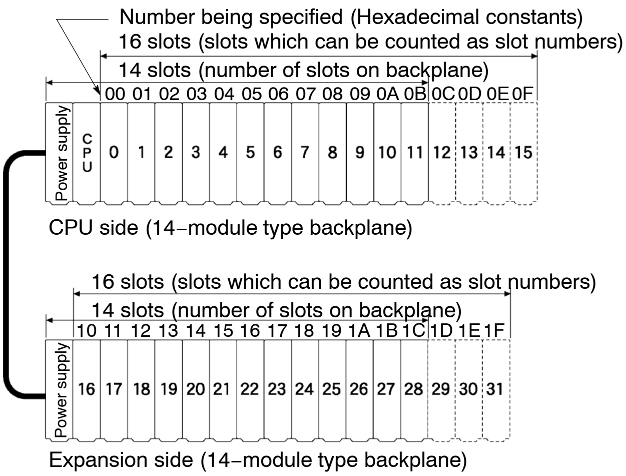
The slot numbers of target intelligent unit are allocated automatically, based on the installation position.



With the FP2 and FP2SH

The slot numbers of the target intelligent unit are allocated automatically, based on the installation position.

Slot numbers are allocated in the order of the board number.
With 7−, 9−, and 12−module type boards, slot numbers are specified in the same way as with the 14−module type.

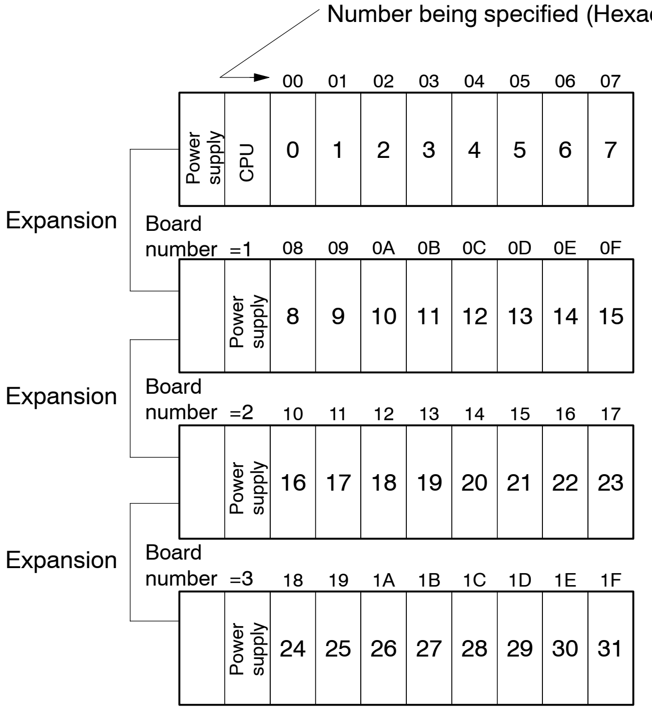


With the FP3 and FP10SH

The slot numbers of the target intelligent unit are allocated automatically, based on the installation position.

Slot numbers are allocated in the order of the board number.

With 3−slot and 5−slot boards, slot numbers are specified in the same way as with 8−slot boards.

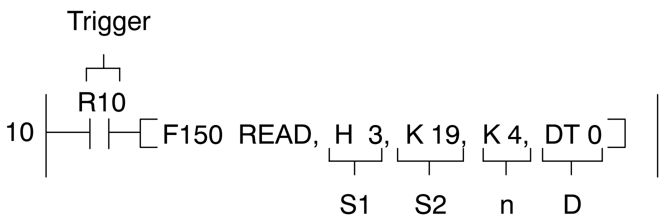


F150 (READ) P150 (PREAD) Data read from intelligent unit

Outline Reads data from the shared memory in an intelligent unit.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F150		(READ)
	H		3
	K		19
	K		4
	DT		0

S1	16-bit equivalent constant for specifying the bank number in the shared memory of the intelligent unit.
S2	16-bit equivalent constant for specifying the starting address in the shared memory of the intelligent unit (source data address).
n	16-bit equivalent constant for specifying the number of words to be read.
D	Starting 16-bit area address for storing read data (destination data address).

Operands

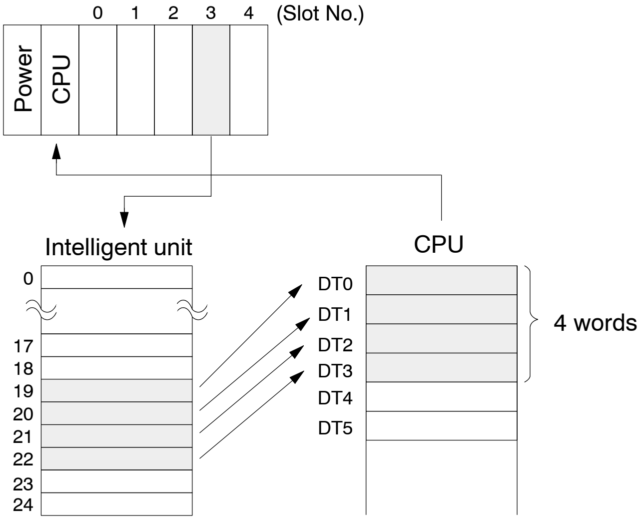
Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	
S1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
S2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FPΣ.

A: Available
N/A: Not Available

Explanation of example

Reads four words of data stored in the addresses starting from K19 to K22 of the intelligent unit shared memory (located in slot 3) and stores them in data registers DT0 to DT3 of CPU when trigger R10 turns on.



Description

The n words of the data stored in the shared memory of the intelligent unit/board specified by S1 is read from the address specified by S2, and is stored in the area specified by D of the CPU.

Specifying the various items

Specifying the slot number and bank number (S1)

Specify the slot in which the intelligent unit has been installed. If the memory has a bank, enter a specification that matches the bank number.

Initial readout address of the shared memory for the intelligent unit (S2)

Specify this referring to the shared memory tables for the various intelligent units.
To specify address 2, specify “K2”.

Number of words to be read (n)

Specify this using a K constant.
To read 10 words of data, specify “K10”.

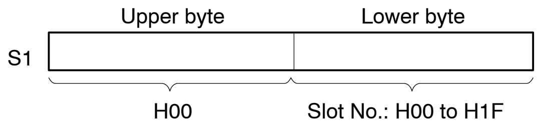
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The value of S1 exceeds the limit of specified range.
  - The area specified using the index modifier exceeds the limit.
  - The read data exceeds the area of D.

Specifying S1

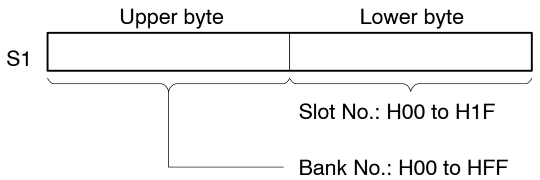
Intelligent unit without bank

Specify the slot number in which the target intelligent unit has been installed.



Intelligent unit with bank

Specify the slot number (H constant) in which the target intelligent unit has been installed, and the bank number (H constant).



Reference: Intelligent unit with bank

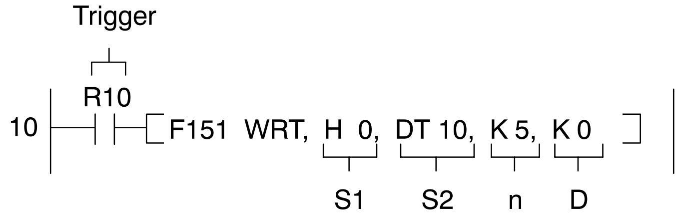
Name	Order No.
FP3 expansion data memory unit	AFP32091 AFP32092
FPΣ expansion data memory unit	AFPG201

F151 (WRT) P151 (PWRT) Data write into intelligent unit

Outline Writes data into the shared memory in an intelligent unit.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F151		(WRT)
	H		0
	DT		10
	K		5
	K		0

S1	16-bit equivalent constant for specifying the bank number in the shared memory of the intelligent unit.
S2	Starting 16-bit area address for storing data written in the shared memory.
n	16-bit equivalent constant for specifying the number of words written in the shared memory.
D	Starting 16-bit area address for storing data written (destination data address).

Operands

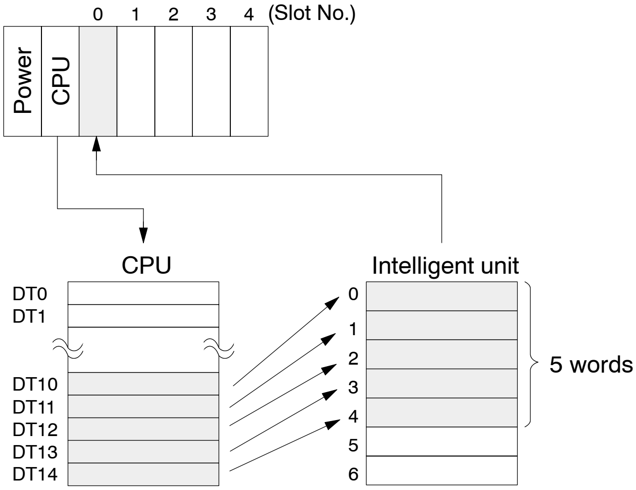
Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I	K	H	
S1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	A

(*1) This cannot be used with the FPΣ.

A: Available
N/A: Not Available

Explanation of example

Five words of data stored in data registers DT10 to DT14 of CPU are written into the addresses starting from K0 to K4 of the intelligent unit shared memory (located in slot 0) when trigger R10 turns on.



Description

Writes n words of the initial data from the area specified by S2 of the CPU to the address specified by D of the shared memory of the intelligent unit specified by S1.

Specifying the various items

Specifying the slot number and bank number (S1)

Specify the slot in which the intelligent unit has been installed. If the memory has a bank, enter a specification that matches the bank number.

Number of words to be written (n)

Specify this using a K constant.
To write 10 words of data, specify “K10”.

Initial address written to the shared memory of the intelligent unit (D)

Specify this referring to the table of shared memories for the various intelligent units.
To specify address 2, specify “K2”.

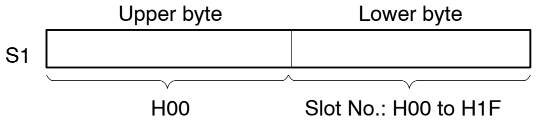
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The value of S1 exceeds the limit of specified range.
  - The area specified using the index modifier exceeds the limit.
  - The range of writing data exceeds the area specified using S2.

Specifying S1

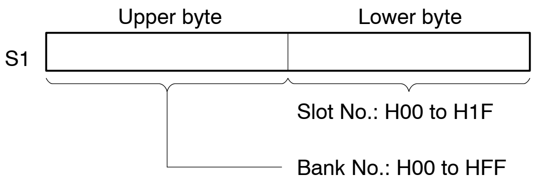
Intelligent unit without bank

Specify the slot number in which the target intelligent unit has been installed.



Intelligent unit with bank

Specify the slot number (H constant) in which the target intelligent unit has been installed, and the bank number (H constant).



Reference: Intelligent unit with bank

Name	Order No.
FP3 expansion data memory unit	AFP32091 AFP32092
FPΣ expansion data memory unit	AFPG201

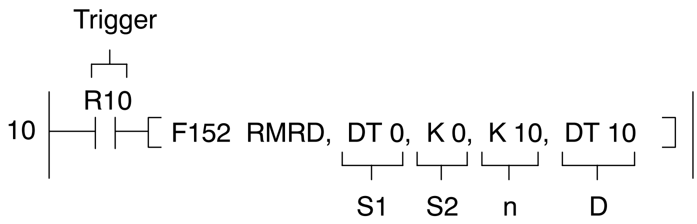
F152 (RMRD) P152 (PRMRD)

Data read from MEWNET-F slave station

Outline Reads data from the specified intelligent unit of the MEWNET-F slave station

Program example

Ladder Diagram



Boolean		
Address	**Instruction**	
10	ST R	10
11	F152	(RMRD)
	DT	0
	K	0
	K	10
	DT	10

S1	Lower 16-bit area of two 16-bit areas for storing control data for **F152 (RMRD)/P152 (PRMRD)**
S2	16-bit equivalent constant or 16-bit area for specifying starting shared memory address in the intelligent unit
n	16-bit equivalent constant or 16-bit area for specifying number of read data words
D	Starting 16-bit area for storing the read data

Operands

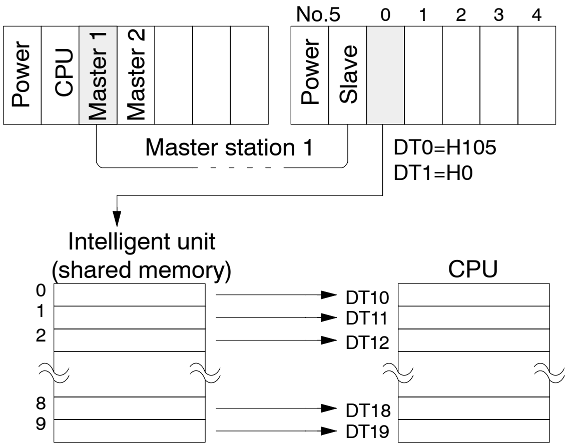
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX (*1)	IY (*2)	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) With the FP2, FP2SH, and FP10SH, this is I0 to IC.
(*2) With the FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Ten words of data stored at address 0 to 9 in the shared memory of the intelligent unit of the slave station specified by DT0 and DT1 are read and the read data stored in data registers DT10 to DT19 of the master station “CPU” when R10 turns on.



Description

This reads n words of the data stored in the shared memory of the intelligent unit of the slave station on the MEWNET−F (remote I/O system) specified by S1 and S1+1 from the address specified by S2, and stores it in the area of the master station CPU specified by D.

Specifying the various items

Control data (S1)

Specify the master station number, slave station number, and slot number (and the bank number, if there is a bank), and specify the memory of the intelligent unit (for detailed information, refer to the following page).

Initial readout address of the shared memory for the intelligent unit (S2)

Enter the specification, referring to the shared memory tables for the various intelligent units.
To specify address 2, specify “K2”.

Number of words to be read (n)

Specify this using a K constant.
To read 10 words of data, specify “K10”.

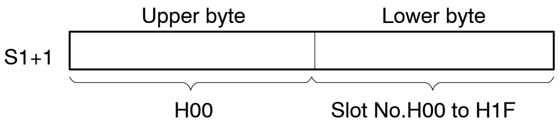
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The control data S1 exceeds the limit of specified range.
  - No MEWNET-F master unit is found.
  - The area specified using the index modifier exceeds the limit.
  - The read data exceeds the area of D.

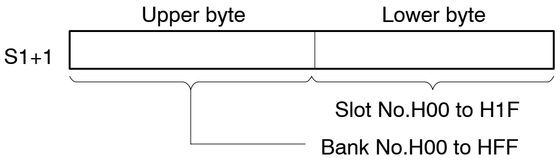
Specifying control data (S1+1 and S1)

Specify the master station number and the slave station number with S1, and the slot number of the target intelligent unit with S1+1.

Intelligent unit without bank



Intelligent unit with bank



Reference: Intelligent unit with bank

Name	Order No.
FP3 expansion data memory unit	AFP32091 AFP32092

Example of setting

When specifying the intelligent unit installed in slot number 0 of the No. 5 slave station on the path of the No. 1 master station, using the program example on page 3 − 401, the program will be structured as follows.

```
  R10
|—| |——[ F0  MV,  H0105,  DT0 ]
       [ F0  MV,  H    0 ,  DT1 ]
       [ F152  RMRD,  DT0,  K0,  K10,  DT10    ]
```

The control data is specified as shown below.
DT0=H0105 (master station No.1 and slave station No.5)
DT1=H0 (slot 0)

Precautions during programming

It is not possible to execute multiple **F152 (RMRD)** instructions and **F153 (RMWT)** instructions at the same time.

The program should be set up so that these instructions are executed when the **F152 (RMRD)/F153 (RMWT)** instruction execution enabled flag (R9035) is on.

R9035	0: Execution inhibited (**RMRD**/**RMWT** instruction being executed) 1: Execution enabled

The **F152 (RMRD)** instruction only enables a request to be accepted. The actual processing is carried out with the **ED** instruction. The **F152 (RMRD)/F153 (RMWT)** instruction completed flag (R9036) can be used to confirm whether or not the instruction has been executed.

R9036	0: Completed normally 1: Completed with error (The error code is stored in DT9036/DT90036)
DT9036 (DT90036)	If the transmission has been completed with an error (R9036 is on), the contents of the error (error code) are stored.

Reference: The error codes stored in the DT9036/DT90036

Error code (HEX)	Description
H5B	Time–out error (no intelligent unit found at the specified location.)
H68	No memory error (no memory exists at the specified address.)
H71	Send answer time–out error
H72	Send buffer full time–out error
H73	Response time–out error

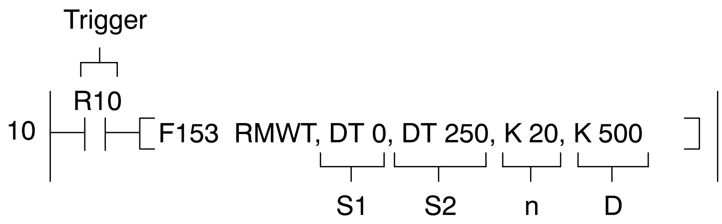
If the error code is H71 to H73, a communication time–out error has occurred. The time–out time can be changed within a range of 10.0 ms to 81.9 s (in units of 10 ms), using the setting of system register 32. The default value is set to 2 seconds for FP3 and 10 seconds for FP2/FP2SH/FP10SH.

F153 (RMWT) P153 (PRMWT) Data write into MEWNET-F slave station

Outline Writes data into the specified intelligent unit of the MEWNET-F slave station.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F153	(RMWT)	
	DT		0
	DT		250
	K		20
	K		500

S1	Lower 16-bit area of two 16-bit areas for storing control data of **F153 (RMWT)/P153 (PRMWT)**
S2	Starting 16-bit area for storing data transferred to the shared memory
n	16-bit equivalent constant or 16-bit area for specifying number of data words written
D	16-bit equivalent constant or 16-bit area for storing the starting address of the shared memory in the intelligent unit

Operands

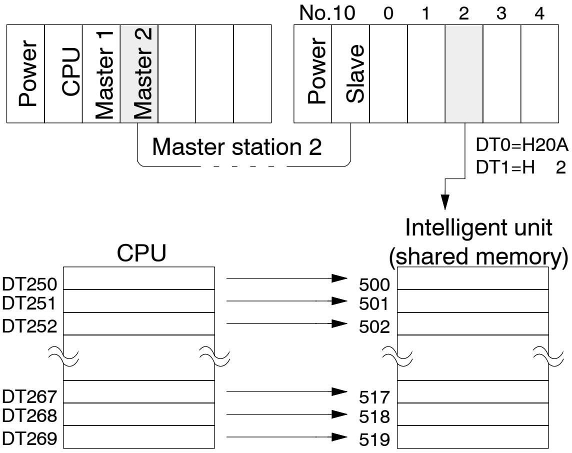
Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	IX (*1)	IY (*2)	K	H	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	A	A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) With the FP2, FP2SH, and FP10SH, this is I0 to IC.
(*2) With the FP2, FP2SH, and FP10SH, this is ID.

A: Available
N/A: Not Available

Explanation of example

Twenty words of data stored in data registers DT250 to DT269 of the master station “CPU” are written into the shared memory of the intelligent unit of slave station starting from address 500 to 519 specified by DT0 and DT1 when R10 turns on.



Description

This writes the initial n words of the data from the area specified by S2 of the CPU to the address specified by D of the shared memory of the intelligent unit of the slave station on the MEWNET−F (remote I/O system) specified by S1 and S1+1.

Specifying the various items

Control data (S1)

Specify the master station number, slave station number, and slot number (and the bank number, if there is a bank), and specify the memory of the intelligent unit (for detailed information, refer to the following page).

Specifying the address of the shared memory (S2)

Enter the specification, referring to the shared memory tables for the various intelligent units.
To specify address 2, specify “K2”.

Number of words to be write (n)

Specify this using a K constant.
To write 10 words of data, specify “K10”.

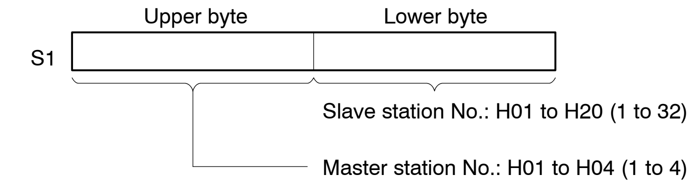
Flag conditions

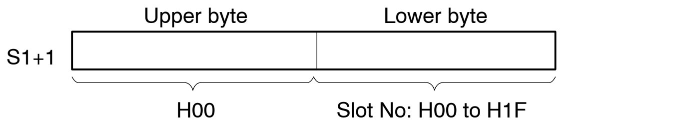
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The control data S1 exceeds the limit of specified range.
  - No MEWNET-F master unit is found.
  - The area specified using the index modifier exceeds the limit.
  - The range of writing data exceeds the area of S2.

Specifying control data (S1+1 and S1)

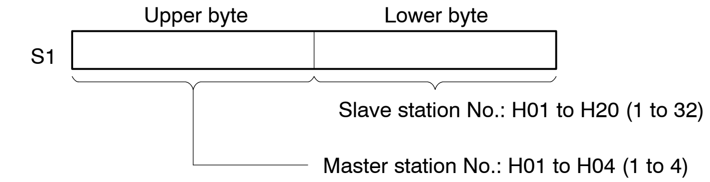
Specify the master station number and the slave station number with S1, and the memory of the target intelligent unit with S1+1.

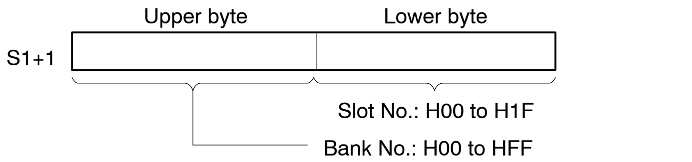
Intelligent unit without bank





Intelligent unit with bank





Reference: Intelligent unit with bank

Name	Order No.
FP3 expansion data memory unit	AFP32091 AFP32092

Example of setting

When specifying the intelligent unit installed in slot number 2 of the No. 10 slave station on the path of the No. 2 master station, using the program example on page 3 − 405, the program will be structured as follows.

```
   R10
──┤ ├──[ F0  MV,  H020A,  DT0 ]
       [ F0  MV,  H   2,  DT1   ]
       [ F153  RMWT,  DT0,  DT250,  K20,  K500 ]
```

The control data is specified as shown below.
DT0=H020A (master station No.2, slave station No.10)
DT1=H2 (slot 2)

Precautions during programming

It is not possible to execute multiple **F152 (RMRD)** instructions and **F153 (RMWT)** instructions at one time. The program should be set up so that these instructions are executed when the **F152 (RMRD)**/**F153 (RMWT)** instruction execution enabled flag (R9035) is on.

R9035	0: Execution inhibited (**RMRD**/**RMWT** instruction being executed) 1: Execution enabled

The **F152 (RMRD)** instruction only enables a request to be sent. The actual processing is carried out with the **ED** instruction. The **F152 (RMRD)**/**F153 (RMWT)** instruction completed flag (R9036) can be used to confirm whether or not the instruction has been executed.

R9036	0: Completed normally 1: Completed with error (The error code is stored in DT9036/DT90036)
DT9036 (DT90036)	If the transmission has been completed with an error (R9036 is on), the contents of the error (error code) are stored.

Reference: The error codes stored in the DT9036/DT90036

Error code (HEX)	Description
H5B	Time−out error (no intelligent unit found at the specified location.)
H68	No memory error (no memory exists at the specified address.)
H71	Send answer time−out error
H72	Send buffer full time−out error
H73	Response time−out error

If the error code is H71 to H73, a communication time−out error has occurred. The time−out time can be changed within a range of 10.0 ms to 81.9 s (in units of 10 ms), using the setting of system register 32. The default value is set to 2 seconds for FP3 and 10 seconds for FP2/FP2SH/FP10SH.

F155 (SMPL) P155 (PSMPL) Sampling start

Availability
FP2/FP2SH/FP10SH FP−X (V2.00 or more) FPΣ (V3.10 or more)/FP0R

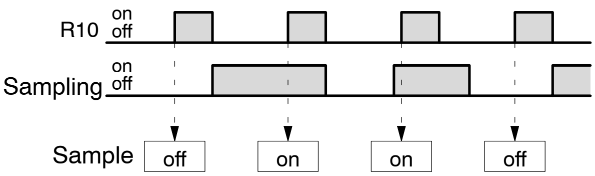
Outline Starts sampling data which is preset in trace memory.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 10 F155 (SMPL)

Explanation of example

When the execution condition (trigger) R10 turns on, sampling of a relay (contact) and register registered in advance is carried out.



Registration of the data to be sampled, specification of the sampling method (such as the cable and the time interval), and specification of the sampling trace can be done using only the programming tool software.

Description

During a sampling trace, sampling of the specified data (relay contacts and registers) is carried out, and the data contents at the time of sampling are stored in the sampling trace memory.

If the sampling trace settings and the startup have not been specified using the programming tool software, processing will not be carried out, even if the execution condition (trigger) is fulfilled.

Sampling traces

This is a function which samples the on/off status of the registered relay and the data stored in the register, either periodically or when the appropriate conditions have been fulfilled, and stores the results in memory. It can be used to confirm changes in the data.

16 relays points and 3 words of registers can be set.

Procedure for executing a sampling trace

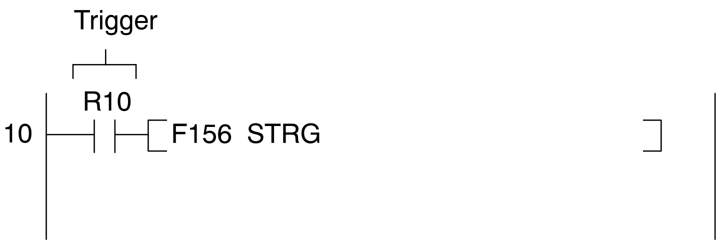
1. Specify registration of the data to be sampled and the sampling method (such as the number of times or the time interval).
2. Indicate that the sampling trace is to begin.
3. Sampling is carried out.
   Sampling can be carried out using with periodic sampling or sampling based on the **F155 (SMPL)** instruction.
4. Stops a sampling trace
   Programming tool software online operation or executing the **F156 (STRG)** instruction applies a stop command trigger. (When a trigger is applied, sampling of the specified delay is carried out, and then sampling stops.) (The programming tool software can also be used to initiate a forced stop.)
5. The programming tool software can be used to read the sampling results from the CPU, and to monitor and confirm them.

F156 (STRG) P156 (PSTRG) Sampling stop

Availability
FP2/FP2SH/FP10SH FP−X (V2.00 or more) FPΣ (V3.10 or more)/FP0R

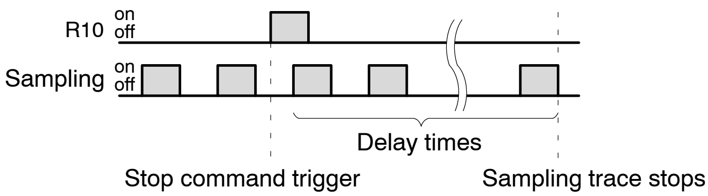
Outline Stops sampling data.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 10
	11	F156 (STRG)

Explanation of example

When the execution condition (trigger) R10 turns on, a sampling trace stop command trigger is applied.



Registration of the data to be sampled, specification of the sampling method (such as the cable and the time interval), and specification of the sampling trace can be done using only the programming tool software.

Description

This instruction applies a sampling trace stop command trigger. When a trigger is applied, sampling of the specified delay is carried out, and then sampling trace stops.

If the sampling trace settings and the startup have not been specified using the programming tool software, processing will not be carried out, even if the execution condition (trigger) is fulfilled.

Sampling traces

This is a function which samples the on/off status of the registered relay and the data stored in the register, either periodically or when the appropriate conditions have been fulfilled, and stores the results in memory. It can be used to confirm changes in the data.

16 relays points and 3 words of registers can be set.

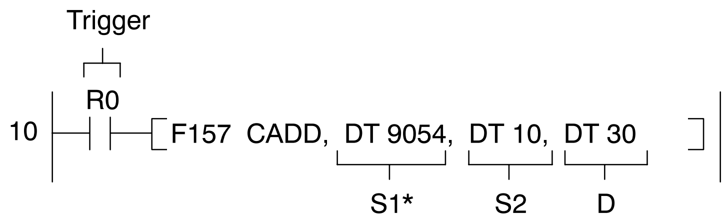
Procedure for executing a sampling trace

1. Specify registration of the data to be sampled and the sampling method (such as the number of times or the time interval).
2. Indicate that the sampling trace is to begin.
3. Sampling is carried out.
   Sampling can be carried out using with periodic sampling or sampling based on the **F155 (SMPL)** instruction.
4. Stops a sampling trace
   Programming tool software online operation or executing the **F156 (STRG)** instruction applies a stop command trigger. (When a trigger is applied, sampling of the specified delay is carried out, and then sampling stops.) (The programming tool software can also be used to initiate a forced stop.)
5. The programming tool software can be used to read the sampling results from the CPU, and to monitor and confirm them.

F157 (CADD) P157 (PCADD) Time addition

Outline Adds specified time data (hours, minutes, and seconds) to date (years, months, and days) and clock (hours, minutes, and seconds) data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P157 (PCADD)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
 (* When FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH, S1 = DT90054)	10 11	ST R 0 F157 (CADD) DT 9054 DT 10 DT 30

S1	Starting 16-bit area for storing date/clock data (3 words are occupied in form of BCD).
S2	Starting 16-bit area for storing time data (2 words are occupied in form of BCD).
D	Starting 16-bit area for storing result (3 words are occupied in form of BCD).

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*****)	IX (*****)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0, FP−e, FP0R, FPΣ, FP−X.
(*2) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Adds the time data stored in data registers DT11 and DT10 to the clock/calendar data stored in special data registers DT9054 to DT9056 (DT90054 to DT90056) when trigger R0 turns on. The result is stored in data registers DT32, DT31, and DT30.

June 17, 1992 10: 30’ 24”: H920617103024 (BCD)

DT9056 (DT90056)				DT9055 (DT90055)				DT9054 (DT90054)			
9	2	0	6	1	7	1	0	3	0	2	4
Years		Months		Days		Hours		Minutes		Seconds	

+ (Addition)

20: 45’ 00”: H00204500 (BCD)

DT11				DT10			
0	0	2	0	4	5	0	0
Hours				Minutes		Seconds	

June 18, 1992 7: 15’ 24”: H920618071524 (BCD)

DT32				DT31				DT30			
9	2	0	6	1	8	0	7	1	5	2	4
Years		Months		Days		Hours		Minutes		Seconds	

Description

The date/clock data (3 words) specified by S1 and the time data (2 words) specified by S2 are added together. The result (time of elapsed value) is stored in the area (3 words) specified by D.

Date/clock data

	S1+2		S1+1		S1	
BCD H code	H00 to H99	H01 to H12	H01 to H31	H00 to H23	H00 to H59	H00 to H59
	Years	Months	Days	Hours	Minutes	Seconds

+ (Addition)

Time data

	S2+1	S2	
BCD H code	H0000 to H9999	H00 to H59	H00 to H59
	Hours	Minutes	Seconds

Date/clock data

	D+2		D+1		D	
BCD H code	H00 to H99	H01 to H12	H01 to H31	H00 to H23	H00 to H59	H00 to H59
	Years	Months	Days	Hours	Minutes	Seconds

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data specified by S1 and S2 is not BCD data.
  - The data specified by S1 is not the date/clock data.
  - The data specified by S2 is not the time data.
  - The specified data exceeds the area.

Data configuration for the internal calendar timer

FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH

DT90056		DT90055		DT90054	
Years	Months	Days	Hours	Minutes	Seconds

Precautions during programming

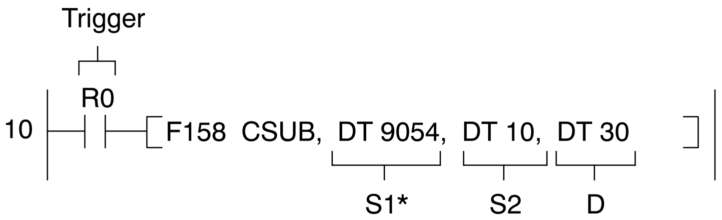
The special data registers DT9054 to DT9056/ DT90054 to DT90056, in which the values of the internal calendar timer are stored, cannot be specified directly for D. To change the value of the internal calendar timer, store the addition results in a separate memory area, and then use the **F0 (MV)** instruction to transfer the value to DT9054 to DT9056/DT90054 to DT90056.

F158 (CSUB) P158 (PCSUB) Time substruction

Outline Subtracts specified time data (hours, minutes, and seconds) from date (years, months, and days) and clock (hours, minutes, and seconds) data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P158 (PCSUB)” is not available.

Program example

Ladder Diagram



(* When FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH, S1 = DT90054)

Boolean Address	Boolean Instruction		
10	ST	R	0
11	F158		(CSUB)
	DT		9054
	DT		10
	DT		30

S1	Starting 16-bit area for storing date/clock data (3 words are used in form of BCD).
S2	Starting 16-bit area for storing time data (2 words are used in form of BCD).
D	Starting 16-bit area for storing result (3 words are used in form of BCD).

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	IX (*2)	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0, FP−e, FP0R, FPΣ and FP−X.
(*2) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

Subtracts the time data stored in data registers DT11 and DT10 from the date/clock data stored in data registers DT9054 to DT9056/ DT90054 to DT90056) when trigger R0 turns on. The result is stored in data registers DT32, DT31, and DT30.

June 17, 1992 10: 30’ 24”: H920617103024 (BCD)

DT9056 (DT90056)				DT9055 (DT90055)				DT9054 (DT90054)			
9	2	0	6	1	7	1	0	3	0	2	4
Years		Months		Days		Hours		Minutes		Seconds	

− (Subtraction)

3: 30’ 30”: H00033030 (BCD)

DT11				DT10			
0	0	0	3	3	0	3	0
Hours				Minutes		Seconds	

⬇

June 17, 1992 6: 59’ 54”: H920617065954 (BCD)

DT32				DT31				DT30			
9	2	0	6	1	7	0	6	5	9	5	4
Years		Months		Days		Hours		Minutes		Seconds	

Description

Subtracts time data (2 words) specified by S2 from the date/clock data (3 words) specified by S1. The result is stored in the area (3 words) specified by D.

Date/clock data

	S1+2		S1+1		S1	
BCD H code	H00 to H99	H01 to H12	H01 to H31	H00 to H23	H00 to H59	H00 to H59
	Years	Months	Days	Hours	Minutes	Seconds

− (Subtraction)

Time data

	S2+1	S2	
BCD H code	H0000 to H9999	H00 to H59	H00 to H59
	Hours	Minutes	Seconds

⬇

Date/clock data

	D+2		D+1		D	
BCD H code	H00 to H99	H01 to H12	H01 to H31	H00 to H23	H00 to H59	H00 to H59
	Years	Months	Days	Hours	Minutes	Seconds

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data specified by S1 and S2 is not BCD data.
  - The data specified by S1 is not the date/clock data.
  - The data specified by S2 is not the time data.
  - The specified data exceeds tha area.

Precautions during programming

The special data registers DT9054 to DT9056/DT90054 to DT90056 in which the values of the internal calendar timer are stored, cannot be specified directly for D. To change the value of the internal calendar timer, store the subtraction results in a separate memory area, and then use the **F0 (MV)** instruction to transfer the value to DT9054 to DT9056/DT90054 to DT90056.

Data configuration for the internal calendar timer

FP2/FP2SH/FP10SH

DT90056		DT90055		DT90054	
Years	Months	Days	Hours	Minutes	Seconds

Usage example: Computing the elapsed time

The elapsed time can be computed using the **F158 (CSUB)** instruction.

Using the calendar timer, store the starting time and ending time in the data memory, and compute the elapsed time between the two values. An example in which operation was stopped at 08:02:15 and resumed at 10:30:25 will be used to show how the time that operation was stopped is computed.

The computation can be thought of as subtracting 08:02:15 from 10:30:25.

Starting time: December 23, 1994 8: 2’ 15”

	S1+2				S1+1				S1			
BCD H code	9	4	1	2	2	3	0	8	0	2	1	5
	Years		Months		Days		Hours		Minutes		Seconds	

Ending time: December 23, 1994 10: 30’ 25”

	S2+2				S2+1				S2			
BCD H code	9	4	1	2	2	3	1	0	3	0	2	5
	Years		Months		Days		Hours		Minutes		Seconds	

The data to be subtracted is taken from the starting time data, as shown below.

(8: 02’ 15”)

BCD H code	0	0	0	8	0	2	1	5
	Hours				Minutes		Seconds	

The section indicating the “Day” is set to “0”.

F158 (CSUB) execution

The results will be as follows.

Result: December 23, 1994 2: 28’ 10”

	DT32				DT31				DT30			
BCD H code	9	4	1	2	2	3	0	2	2	8	1	0
	Years		Months		Days		Hours		Minutes		Seconds	

The section indicating the hour, minutes and seconds is read as ”2 hours, 28 minutes, 10 seconds”, and this is the elapsed time.

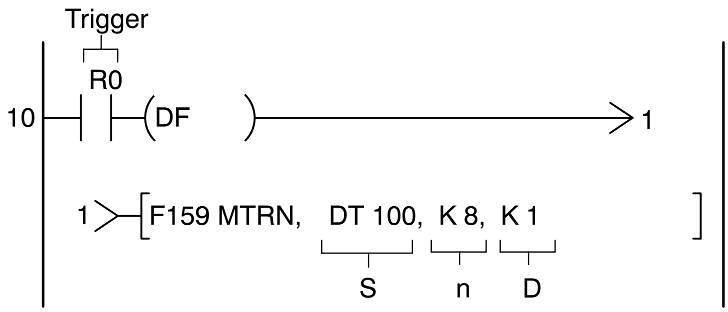
Availability
FPΣ/FP−X/FP0R

F159 (MTRN) Serial data communication

Outline This is used to send data to or receive data from an external device through the specified RS232C port.

Program example

Ladder Diagram



Boolean			
Address	Instruction		
10	ST	R	0
11	DF		
12	F159		(MTRN)
	DT		100
	K		8
	K		1

S	Starting area of data table (data register)
n	Area for storing the number of bytes of data to be transmitted, or constant data. When the value is positive, an end code is added. When the value is negative, an end code is not added. When the value is H8000, the transmission mode of the RS232C port is changed.
D	Port for transmitting data (K0, K1, K2) K0: TOOL port (FPΣ 32k, FP−X) K1: COM1 Port (FP0R: COM Port) K2: COM2 Port

Operands

Operand	Relay				Timer/Counter		Register		Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	IX (*1)	IY	K	H	
S	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	A

(*1) I0 to ID.

A: Available
N/A: Not Available

Description

This instruction is used to send and receive instructions and data when an external device (computer, measuring instrument, bar code reader, etc.) has been connected to the specified RS232C port.

1) Transmission

Transmits “n” bytes of the data stored in the data table that begins from the starting area specified in “S” through the communication port specified in “D” to an external device. A start code and end code can be automatically added to the transmission. The maximum number of bytes that can be transmitted is 2048.

2) Reception

Reception is controlled by the reception done flag (R9038/R9048) turning on and off. When the reception done flag is off, reception can take place at any time and data coming into the RS232C port is stored in the data register specified in system registers 416 to 419.
The F159(MTRN) instruction is used to turn off (enable reception) the reception done flag (R9038/R9048). The maximum number of bytes that can be received is 4094.

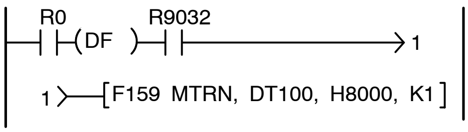
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when there is an index modifier.
  - The data table exceeds the area because of the number of bytes specified in “n”.

3) Changing the transmission mode of the RS232C port

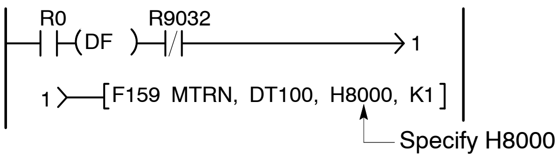
An F159(MTRN) instruction can be executed to change between “general transmission mode” and “computer link mode”. To do so, specify “H8000” in “n” (the number of transmission bytes) and execute the instruction.

Changing from “general port” to “computer link”



RS232C port selection flag in R9032 or R9042.
Turns on when “general port” is selected.

Changing from “computer link” to “general port”



Note

When the power is turned on, the mode of use selected in system register 412 takes effect.
The FP0R, FPΣ 32k, FP−X tool port is always set to the computer link mode in the PROG. mode.

Programming and operation during transmission

To execute transmission, write the data to be transmitted to the data table and specify with an **F158(MTRN)** instruction.

Use an **F0(MV)** or **F95(ASO)** instruction to write the data to be transmitted to the transmission data storage area specified in “S”.

- Do not include an end code in the transmission data. An end code is added automatically.
- When “yes” is specified for the start code in system register 413 or 414, do not add a start code to the transmission data. A start code is added automatically.
- The maximum number of transmission bytes “n” is 2048.

Data table for transmission

This is used as a data table for transmission, starting at the data register specified in “S”.

[S]		
[S+1]	2	1
[S+2]	4	3
~		
[S+n]	2n	2n−1

[S]: At the beginning of transmission, nothing is set. At the end of transmission, “0” is set.

[S+1] to [S+n]: Transmission data storage area (The circled numbers indicate the order of transmission.)

Example:

Transmitting the eight characters A, B, C, D, E, F, G, H (8 bytes of data)
This example uses DT100 to DT104 as the data table.

DT100	K8	
DT101	H42 (B)	H41 (A)
DT102	H44 (D)	H43 (C)
DT103	H46 (F)	H45 (E)
DT104	H48 (H)	H47 (G)

DT100: When transmission begins: Nothing is set. When transmission ends: Cleared to 0.

DT101 to DT104: Data is transmitted in order from the lower−order byte.

Data table before transmission

Notes

- **When using a RS232C × 1 ch type communication cassette, transmission does not take place until CS (Clear to Send) turns on. If you are not going to connect to the other device, connect to RS (Request to Send). (FPΣ, FP−X C14)**
  As for the FP−X C30/C60, it depends on the settings. Refer to the FP−X Manual.

***1. With the FP0R, FPΣ V3.10 or later, FP−X V2.50 or later, the number of transmitted data is set.**

Program

Specify the starting address of the transmission data table in “S”, and the number of data bytes to be transmitted in “n”.

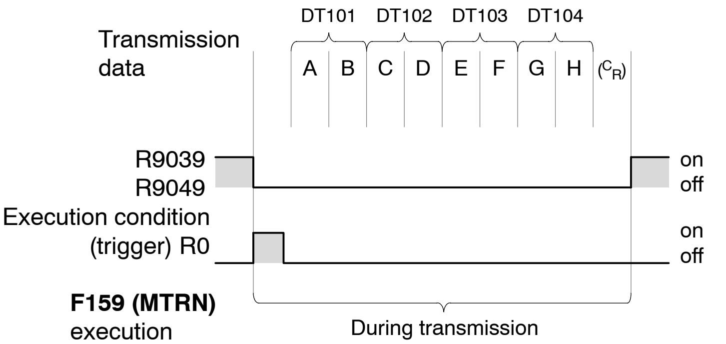
```
 R0
─┤ ├─[ F1 DMV, H44434241, DT101 ]     Write the transmission
     [ F1 DMV, H48474645, DT103 ]     data to the data table.
 R1
─┤ ├─( DF )──────────────────────> 1

1 >─[ F159 MTRN, DT100, K 8, K 1 ]    Transmit the data in
                                      the data table.
```

Operation

When the execution condition of the **F159(MTRN)** instruction turns on, operation is as follows when the transmission done flag (R9039/R9049) is on:

1) “n” is preset in “S”. The reception done flag (R9038/R9048) is turned off, and the reception data number is cleared to “0”.
2) The set data is transmitted in order from the lower−order byte in “S+1” of the table.
   - During transmission, the transmission done flag (R9039/R9049) turns off.
   - If system register 413 or 414 is set to start code with STX, a start code is automatically added to the beginning of the data.
   - The end code specified in system register 413 or 414 is automatically added to the end of the data.



During this interval the **F159(MTRN)** instruction cannot be executed.

3) When all of the specified quantity of data has been transmitted, the “S” value is cleared to “0” and the transmission done flag (R9039/R9049) turns on.

When you do not wish to add an end code to transmissions, use one of the following methods:

Specify the number of bytes to be transmitted using a negative number.
If you also do not wish to add an end code to receptions, set system register 413 or 414 to “no end code”.

Example:

Program for transmitting 8 bytes of data without adding an end code

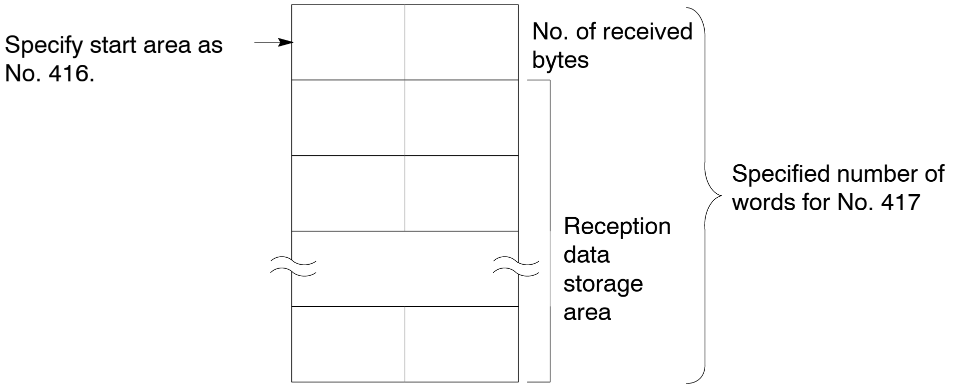
```
 R0
─┤ ├─( DF )──────────────────────> 1

1 >──[F159 MTRN, DT100, K−8, K1 ]
                        ↑
                        └─ Specify “K−8”.
```

Preparation for reception

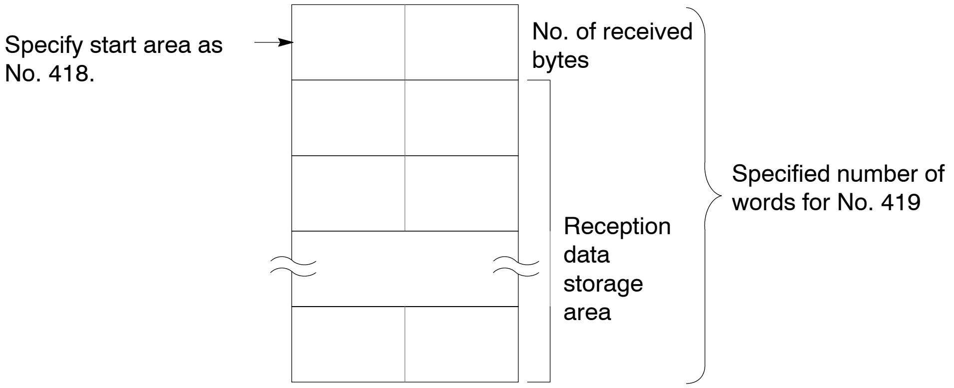
Setting of COM 1 port reception buffer No. 416 and No. 417

The area of data registers DT0 up to DT2047 is the default reception buffer.
The maximum number of bytes that can be received is 4094 bytes.



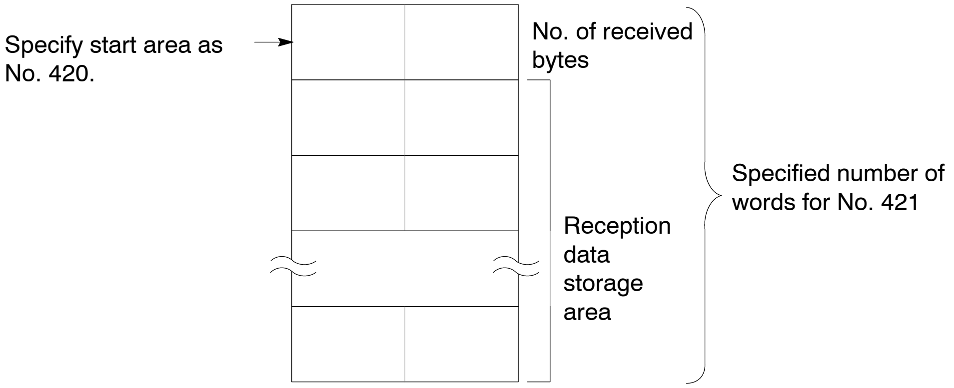
Setting of COM 2 port reception buffer No. 418 and No. 419 (This setting is not available for the FP0R.)

The area of data registers DT2048 up to DT4095 is the default reception buffer.
The maximum number of bytes that can be received is 4094 bytes.



Setting of Tool port reception buffer No. 420 and No. 421

The area of data registers DT4096 up to DT6143 is the default reception buffer.
The maximum number of bytes that can be received is 4094 bytes.



Programming and operation during reception

Data sent from an external device connected to the RS232C port is stored in the data registers that have been set as the reception buffer.

Data registers are used for the reception buffer. Specify the data registers in system registers 416 to 419.

The number of bytes of data received is stored in the starting address of the reception buffer. The initial value is “0”.

Received data is stored in the received data storage area in order from the lower−order byte.

Reception buffer

Using a “reception buffer” data register

Word (address)		
0		
1	2	1
2	4	3
≈		
n	2n	2n−1

The number of bytes received is stored in this area. (word 0)

Reception buffer storage area (The circled numbers indicate the order of storage.) (words 1 to n)

Example:

Receiving eight bytes of data, A, B, C, D, E, F, G, H, from an external device through the COM1 port

DT200 to DT204 are used as the reception buffer.
System register settings are as follows:
– System register 416: K200
– System register 417: K5

DT200	K8	
DT201	H42 (B)	H41 (A)
DT202	H44 (D)	H43 (C)
DT203	H46 (F)	H45 (E)
DT204	H48 (H)	H47 (G)

The received number of bytes is stored as data is stored. (DT200)

Received data is stored in order from the lower−order byte. (DT201 to DT204)

Reception buffer when reception is completed

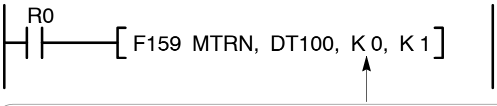
Table of related flags and system registers

Item	For COM1	For COM2	For Tool
Transmission mode flag	R9032	R9042	R9040
Reception done flag	R9038	R9048	R903E
Transmission done flag	R9039	R9049	R903F
Beginning of reception buffer	Specified in 416	Specified in 418	Specified in 420
Reception buffer capacity	Specified in 417	Specified in 419	Specified in 421

Program

The reception done flag (R9038/9048) turns on when data reception from the external device is completed. Reception of any further data is prohibited.

To receive subsequent data, you must execute an **F159(MTRN)** instruction to turn off the reception done flag (R9038/R9048) and clear the byte number to “0”.



To repeatedly perform only reception, specify K0.

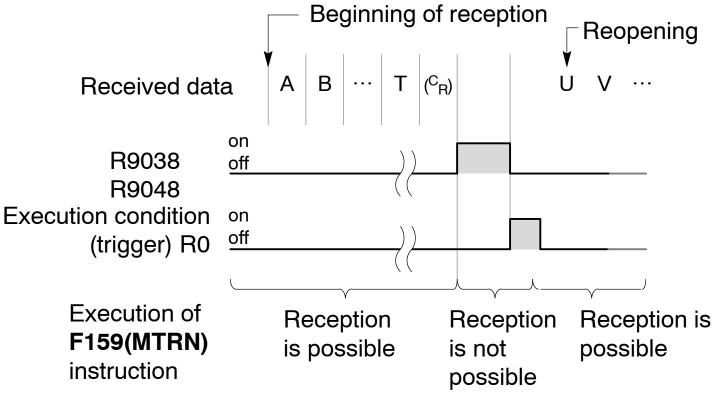
R9038/R9048 also turn off when transmission is performed with a byte number specification.

Operation

When the reception done flag (R9038/R9048) is off, operation takes place as follows when data is sent from an external device.
(R9038/R9048 are off during the first scan after RUN. “0” is set in the starting area of the reception buffer specified in the system registers.)

1) Incoming data is stored in order from the lower−order byte of the 2nd−word area of the reception buffer.

   Start and end codes are not stored.



2) When the end code is received, the reception done flag (R9038/9048) turns on. Reception of any further data is prohibited.
3) When an **F159(MTRN)** instruction is executed, the reception done flag (R9038/9048) turns off, the number of received bytes is cleared, and subsequent data is stored in order from the lower−order byte.

☞ Notes

- **To perform repeated reception of data, refer to the following steps.**
  1) Receive data
  2) Reception done (R9038/R9048: on, reception prohibited)
  3) Process received data
  4) Execute F159(MTRN) instruction (R9038/R9048: off, reception possible)
  5) Receive subsequent data
- **The reception done flag (R9038/R9048) also changes during scanning.**

F159 (MTRN) P159 (PMTRN) Serial data communication (for MCU COM port)

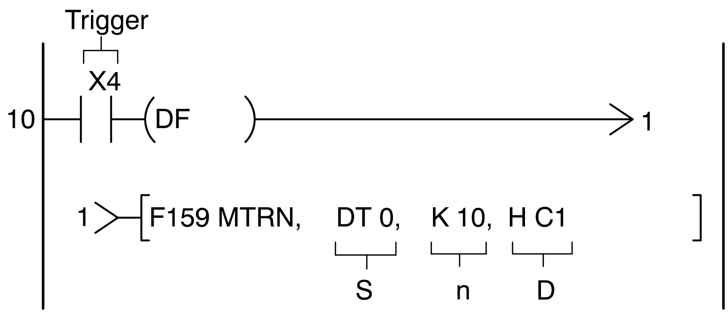
Availability
FP2/FP2SH

Outline Data is transmitted to external equipment via the COM port of the specified MCU.

This function is available from FP2/FP2SH Ver. 1.50 or later.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	X	4
11	DF		
12	F159		(MTRN)
	DT		0
	K		10
	H		C1

S	Head area of the data table
n	Area or constant data in which the byte number of the transmitted data stored – When it is positive value, the terminal code is added in transmission. – When it is negative value, the terminal code is not added in transmission. – In case of H8000, the application of the MCU port specified in transmission is changed.
D	Specification of the slot number and port number of the MCU unit which the data is transmitted.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I (*1)	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	A	A

(*1) I0 to ID.

A: Available
N/A: Not Available

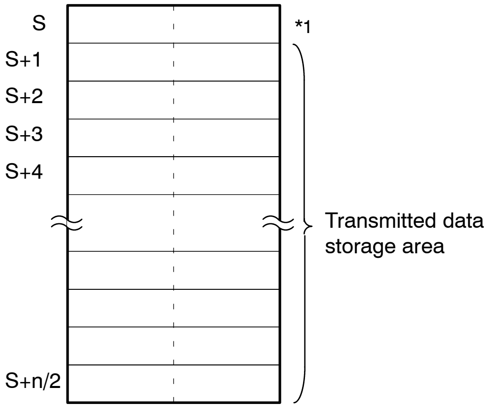
Reference: FP2 Multi Communication Unit Manual

Description

1) It is used to transmit commands or data to the COM port (COM1 or COM2) of the specified MCU unit connecting with external equipment (such as PC, measuring instrument, barcode reader).
Note: The operation mode of the communication port of the MCU should be set to the general−purpose serial communication mode.

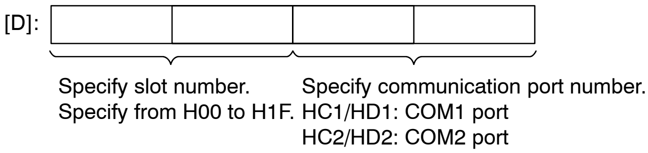
2) [n] bytes of the data stored in the data table which is headed with the area specified by [S] is transmitted to external equipment from the communication port of the CPU or MCU unit specified by [D].

Data table (transmitted buffer)



*1: Nothing is specified for the initial address of the transmitted buffer.

3) The slot number and the communication port number specified by [D] is set as below.



* Caution:

1. Specify to K1 (H1) for the COM port of the CPU.
2. When specifying [D] with the K constant,
   ex.) if the slot number is set to 3, and the COM2 (2) is selected for the communication port of the MCU,
   set as follows.
   H03C2 to K962 (*convert the content specified in hexadecimal to decimal)

4) The starting code and the terminal code can be added automatically in transmission.

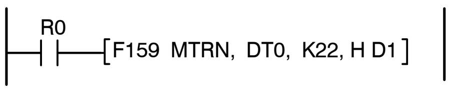
5) The transmitted byte number is maximum of 2048 bytes (including starting code and terminal code).

6) When a negative value is specified for the transmitted byte number, the data will be transmitted without the terminal code.

7) When 8000H is specified for the transmitted byte number, the operation mode of the specified communication port can be switched between the computer link and the general−purpose communication mode.

8) The communication parameter for the communication port can be set by specifying the communicating port number to HD1 or HD2.
When HD1 is designated: the communication parameter is registered for the COM 1 port.
When HD2 is designated: the communication parameter is registered for the COM 2 port.

Example



9) The communication parameter data consists of 11 words.

1) Unit number setting value (K1 to K99)
2) Baud rate setting value (K0 to K10) *2

*2. Baud rate setting value

Storage value	Baud rate
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	19200
7	38400
8	57600
9	115K
10	230K

3) Data length setting value (K0=7 bits, K1=8 bits)
4) Parity setting value (K0=no parity, K1=parity 0, K2=Odd, K3=Even)
5) Stop bit length setting value (K0: 1 bit, K1: 2 bits)
6) RS/CS setting (K0=disable, K1=able)
7) Waiting time for starting transmission (K: 0=Time for about three characters/effective time=Kn*0.01 ms (0 to 100 ms))
8) Starting code STX setting value (K0=disable, K1=able)
9) Terminator setting value (K0=cR, K1=cR+Lf, K2=time (24 bits), K3=EXT)
10) Reception done judgment time (K:0=immediate/effective time=Kn*0.01 ms (0 to 100 ms))
11) Modem initialization (K0=Not initialized when the power turns on, K1=Initialized when the power turns on)

☞ **Note**

The execution for switching the operation mode of the communication (between the computer link and the general−purpose serial communication mode) or the setting for the communication parameter should be carried out when no communication is performed.
If these operations are executed in communicating, the data which is being transmitted will be cancelled, and the reception error will occur for the data which is being received and this data may not be received properly.
When the communication parameter is specified, the received byte number should be specified to the even data of 22 bytes or smaller. If it is specified to the value larger than 22 bytes or odd byte, an error occurs in the parameter settings of the MCU.

Flag conditions

- Error flag (R9007) (R9008):
  - It turns on, when the specified address using the index modifier exceeds a limit.
  - It turns on, when the MCU unit does not exist in the slot No. specified by [D].
  - It turns on, when the MCU unit does not exist in the slot No. specified by [D]
  - It turns on, when the data device specified by [S] exceeds the area.
  - It turns on, when the transmitted byte number specified by [n] is outside of the specified area.
  - It turns on, when the transmitted byte number specified by [n] exceeds the area of the data table.
  - It turns on, when H8000 is designated in the PC link mode.
  - It turns on, when an additional parameter is registered in executing the parameter registration.
  - It turns on, when H8000 is designated in the parameter registration.
  - It turns on, when a negative value is designated in the parameter registration.

F161 (MRCV) P161 (PMRCV) Serial data reception (for MCU COM port)

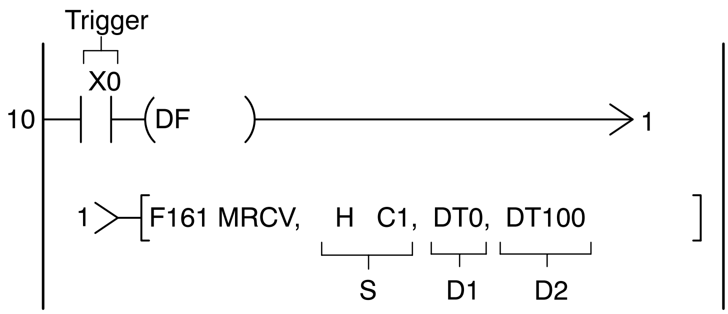
Availability
FP2/FP2SH

Outline Data is received from external equipment via the COM port of the specified MCU.

This function is available from FP2/FP2SH Ver. 1.50 or later.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	X	0
11	DF		
12	F161		(MRCV)
	H		C1
	DT		0
	DT		100

S	Specification of the slot number and port number of the MCU unit which the data is received.
D1	Initial address in which the received data stored.
D2	Ending address in which the received data stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I (*1)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A	A	A
D1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

(*1) I0 to ID.

A: Available
N/A: Not Available

Explanation of example

When the reception done signal X0 of the COM 1 port is on, the received data is readout, and stored in DT0 to DT100.

Flag conditions

- Error flag (R9007) (R9008):
  - It turns on, when the specified address using the index modifier exceeds a limit.
  - It turns on, when the MCU unit does not exist in the slot No. specified by [S].
  - It turns on, when the communication port specified by [S] does not exist.
  - It turns on, when the data device specified by [D1] exceeds the area.
  - It turns on, when the data device specified by [D2] exceeds the area.
  - It turns on, when [D1] > [D2].

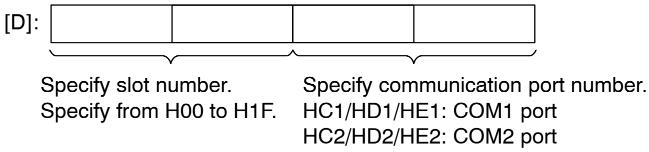
Reference: FP2 Multi Communication Unit Manual

Description

1) It is used to receive commands or data for the COM port (COM1 or COM2) of the specified MCU unit connecting with external equipment (such as PC, measuring instrument, barcode reader).
Note: The operation mode of the communication port of the MCU should be set to the general−purpose communication mode.

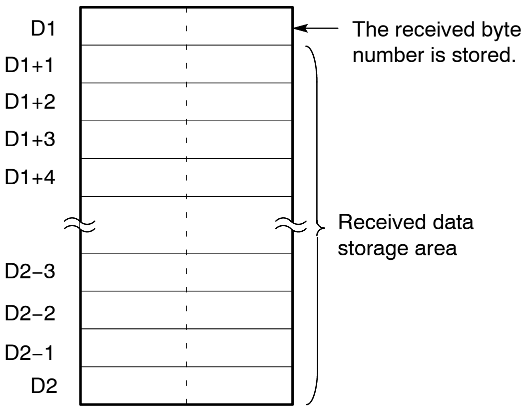
2) The received data is readout to the communication port of the MCU unit in the slot No. specified by [S], and stored in the specified data area of [D1] to [D2].

3) The slot number and the communication port number specified by [S] is set as below.



4) The received byte number is set for the initial address of the data area specified by [D1].
* If the received data exceeds the ending address specified by [D2], the operation error is detected. At that time, the data which has been received up to the area of [D2] is stored.

Data table (received buffer)



<Reading of communication parameter and condition>

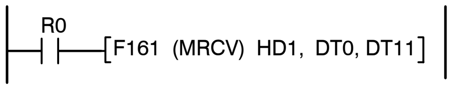
5) When the communication port numbers specified by [S] is HD1 or HD2, HE1 or HE2, the registered communication parameter and the monitoring data are read.

HD1:The communication parameter data in the COM 1 port is read.
HD2:The communication parameter data in the COM 2 port is read.

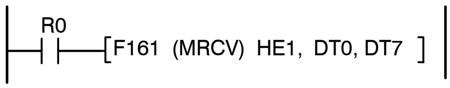
When HE1 or HE2 is designated, the operation mode of each communication port and the information on the communication cassette detection is read.

HE1:The monitoring data in the COM 1 port is read.
HE2:The monitoring data in the COM 2 port is read.

example:



example:



6) There are eight 2048−byte buffers in the received buffer of the MCU unit, and eight data can be received sequentially.

If nine or more data should be received, the MCU unit detects the received buffer full error.
If the received buffer FULL error is detected, the MCU unit prohibits the reception of data in that channel and inform about the error.
The byte number which can be received in one buffer is maximum of 2048 bytes (including terminal code).
However, the data which can be received with the MRCV do not include terminal code.

<Configuration of communication parameter>

7) The communication parameter data consists of 11 words.

1) Unit number setting value (K1 to K99)
2) Baud rate setting value (K0 to K10) *2

*2. Baud rate setting value

Storage value	Baud rate
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	19200
7	38400
8	57600
9	115K
10	230K

3) Data length setting value (K0=7 bits, K1=8 bits)
4) Parity setting value (K0=no parity, K1=parity 0, K2=Odd, K3=Even)
5) Stop bit length setting value (K0: 1 bit, K1: 2 bits)
6) RS/CS setting (K0=disable, K1=able)
7) Waiting time for starting transmission (K: 0=Time for about three characters/effective time=Kn*0.01 ms (0 to 100 ms))
8) Starting code STX setting value (K0=disable, K1=able)
9) Terminator setting value (K0=cR, K1=cR+Lf, K2=time (24 bits), K3=EXT)
10) Reception done judgment time (K:0=immediate/effective time=Kn*0.01 ms (0 to 100 ms)
11) Modem initialization (K0=Not initialized, K1=Initialized)

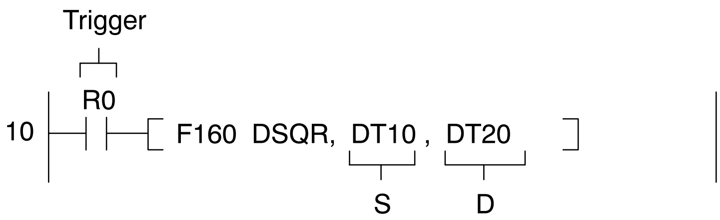
<Configuration of monitor data>

1) Operation mode (K0 to K7)
(K0=computer link, K1=general–purpose serial, K2=PC link, K7=modem initialization)
2) Communication cassette detection (from K0)
(No communication cassette=0, RS232C=K232, RS422=K422, RS485=K485)
3) Reception error code
(Lower byte: bit 0=received buffer overrun, bit 1=stop bit not detected, bit 2=parity unmatched)
(Higher byte: bit 0=received buffer overflow, bit 1=received buffer full)
4) Number of times reception errors (number of times which the reception errors to be stored in the above lower byte are detected)
5) Setting error code
(Lower byte: bit 0=error in the dip switch setting of the operation mode, bit 1=operation mode setting which exceeds the usable limit of the unit)
(Higher byte: bit 0=error in the communication parameter setting, bit 1=error in the number of transmitted data)
6) Error parameter No. (K0 to K11)
7) Modem initialization
(h0000=deinitialized h0100=now initializing h0200=initialization completed. h02FF=initialization failed.)

F160 (DSQR) P160 (PDSQR) 32-bit data square root

Outline Takes the square root of the specified 32-bit data.
For the FP0R, FPΣ and FP−X, the P type high−level instruction “P160 (PDSQR)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F160 (DSQR) DT 10 DT 20

S	32-bit equivalent constant or lower 16-bit area of 32-bit area for storing data to be calculated
D	Lower 16-bit area of 32-bit area for storing the calculated result

Operands

Operand	Relay				Timer/Counter		Register			Index register		Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	IX (*2)	IY	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) This cannot be used with the FP0R, FPΣ and FP−X.
(*2) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

A: Available
N/A: Not Available

Explanation of example

The square root of 32−bit data stored in DT11 and DT10 is calculated and stored in DT21 and DT20 when R0 turns on.
When K64 is stored in DT11 and DT10, the following occurs.

Source data [S+1, S]: K64

	DT11				DT10			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0
Decimal data	K64							

Higher 16-bit area Lower 16-bit area

R0: on

Destination [D+1, D]: K8

	DT21				DT20			
Bit position	15 · · 12	11 · · 8	7 · · 4	3 · · 0	15 · · 12	11 · · 8	7 · · 4	3 · · 0
Binary data	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
Decimal data	K8							

Higher 16-bit area Lower 16-bit area

Description

The square root of 32-bit data specified by S1 is calculated and stored in the 32-bit area specified by D.

In the result, the digits beyond the decimal point are disregarded.

$\sqrt{(S+1, S)} \rightarrow (D+1, D)$

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The data specified by S is a negative value.

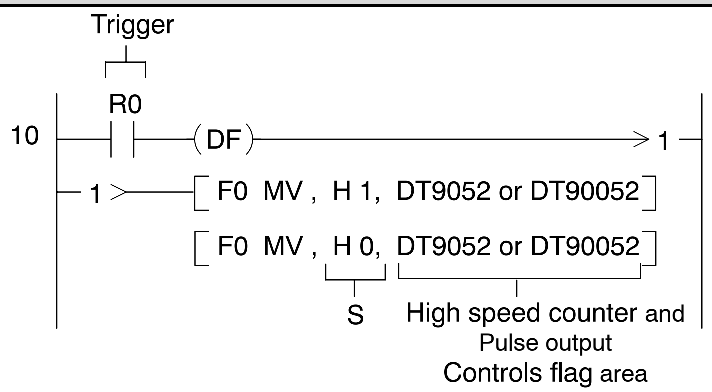
Availability
FP0/FP0R/FPΣ/FP−X

F0 (MV) High−speed counter control

Outline This instruction is used to perform control such as software reset, counter disabling, and high−speed counter instruction clearing.

Program example

Ladder Diagram



* The high−speed counter and pulse output controls flag area varies depending on the PLC type.

Boolean Address	Boolean Instruction		
10	ST	R	0
11	DF		
12	F0		(MV)
	H		1
	DT		9052
17	F0		(MV)
	H		0
	DT		9052

S	Area for storing high−speed counter control code or constant data

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX (*1)	IY (*2)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A

A: Available

(*1) I0 to IC on FPΣ/FP−X/FP0R

(*2) ID on FPΣ/FP−X/FP0R

Description

Performs high−speed counter control according to the control code specified in “S”.

This instruction is used to perform the following operations when using a high−speed counter:

<Function>

1) Performing a software reset
2) Disabling the count
3) Temporarily disables reset input setting using external inputs
4) Clearing control executed with high−speed counter and pulse output instructions **F166** or **F167**.

Once written, a control code is retained until the next write operation.

Precautions during programming

Hardware resets can only be disabled if a reset input is used.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used
  - The “S” is outside specification range

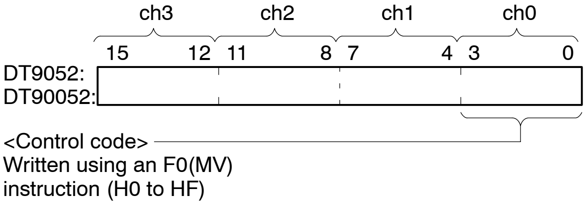
Using the FP0/FP−e

High−speed counter and Pulse output controls flag area

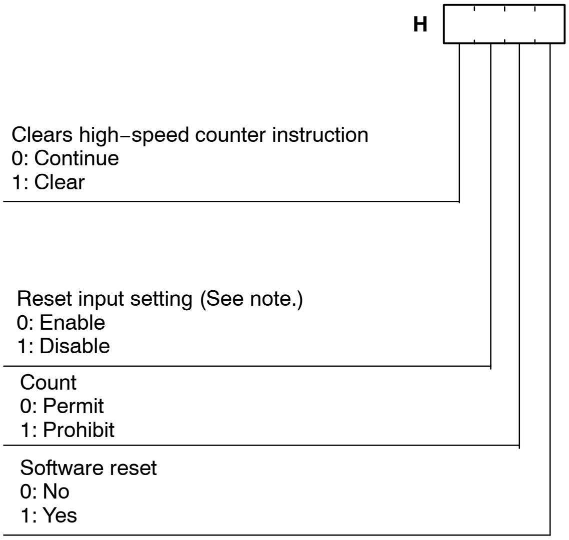
Four bits are allocated to each high−speed counter channel for use as the control code write area DT9052 (DT90052 on the FP0 T32)

A control code written using an **F0(MV)** instruction is stored in special data register DT9052 (DT90052 on the FP0 T32).

High−speed counter and Pulse output controls flag area of FP0



Select control codes in units of one bit and specify with H.



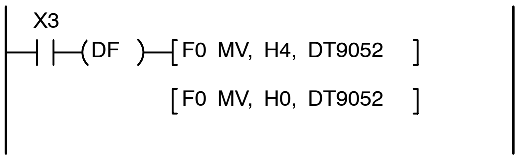
Note:
At the reset input setting, you set whether the reset input (X2 or X5), which was assigned by the system register high−speed counter setting, will be enabled or disabled.

Example:

- **Perform software reset .... H1(0001)**
- **Prohibit count .... H2(0010)**
- **Clear high−speed counter instruction .... H8(1000)**
- **Clear high−speed counter instruction and reset elapsed value .... H9(1001)**

Program example

Example: Software reset of channel 0 of high−speed counter.



Using the FPΣ

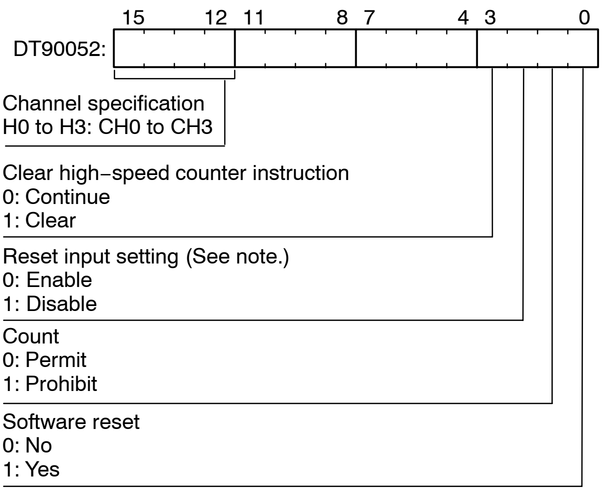
High−speed counter and Pulse output controls flag area

The area DT90052 for writing channels and control codes is allocated as shown below.

The control code written by the **F0 (MV)** instruction is stored in the control code monitor area while it is written in the special register DT90052. (Refer to the table below.)

The written data is the data for lower 8 bits only.

High−speed counter and Pulse output controls flag area of FPΣ



Note:
At the reset input setting, you set whether the reset input (X2 or X5), which was assigned by the system register high−speed counter setting, will be enabled or disabled.

Program example

Example 1: Software reset of channel 0 of high−speed counter

```
  R0
--| |--(DF )--[F0 MV, H1, DT90052 ]
              [F0 MV, H0, DT90052 ]
```

Example 2: Software reset of channel 2 of high−speed counter

```
  R0
--| |--(DF )--[F0 MV, H2001, DT90052]
              [F0 MV, H2000, DT90052]
```

Using the FP−X

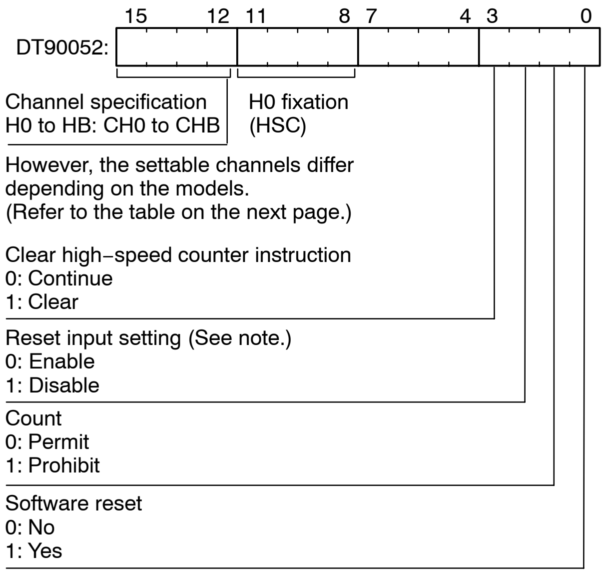
High−speed counter and Pulse output controls flag area

The area DT90052 for writing channels and control codes is allocated as shown below.

The control code written by the **F0 (MV)** instruction is stored in the control code monitor area while it is written in the special register DT90052. (Refer to the table below.)

The written data is the data for lower 8 bits only.

High−speed counter and Pulse output controls flag area of FP−X



Program example

Example 1: Software reset of channel 0 of high−speed counter

```
   R0
|--| |--(DF )--[F0 MV, H1, DT90052 ]
                [F0 MV, H0, DT90052 ]
```

Example 2: Software reset of channel 1 of high−speed counter

```
   R0
|--| |--(DF )--[F0 MV, H1001, DT90052 ]
                [F0 MV, H1000, DT90052 ]
```

Note: FP−X Ry type
At the reset input setting, you set whether the reset input (X2 or X5) of the pulse I/O cassette, which was assigned by the system register high−speed counter setting, will be enabled or disabled.

Note: FP−X Tr type
At the reset input setting, you set whether the reset input (X6 or X7) assigned for the controller input will be enabled or disabled.

Using the FP0R

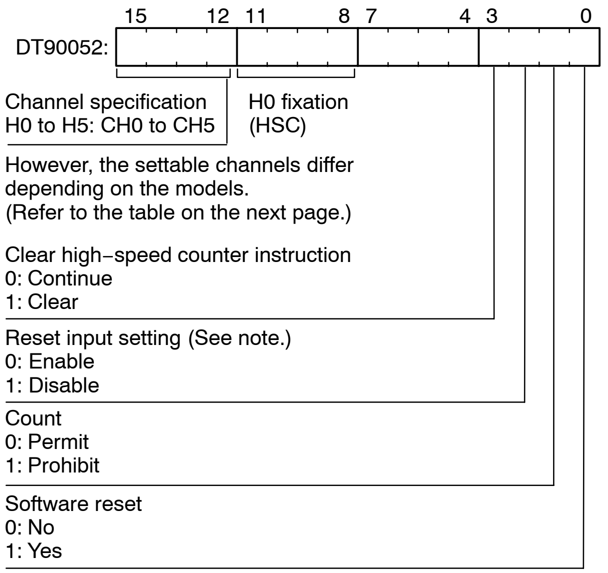
High−speed counter and Pulse output controls flag area

The area DT90052 for writing channels and control codes is allocated as shown below.

The control code written by the **F0 (MV)** instruction is stored in the control code monitor area while it is written in the special register DT90052. (Refer to the table below.)

The written data is the data for lower 8 bits only.

High−speed counter and Pulse output controls flag area of FP0R



High−speed counter control for FP0R, FPΣ and FP−X

Channel No.	Control code monitor area			
	FPΣ	FP−X Ry type	FP−X Tr type	FP0R
ch0	DT90190	DT90360	DT90370	DT90370
ch1	DT90191	DT90361	DT90371	DT90371
ch2	DT90192	DT90362	DT90372	DT90372
ch3	DT90193	DT90363	DT90373	DT90373
ch4	—	DT90364	DT90374	DT90374
ch5	—	DT90365	DT90375	DT90375
ch6	—	DT90366	DT90376	—
ch7	—	DT90367	DT90377	—
ch8	—	DT90368	—	—
ch9	—	DT90369	—	—
chA	—	DT90370	—	—
chB	—	DT90371	—	—

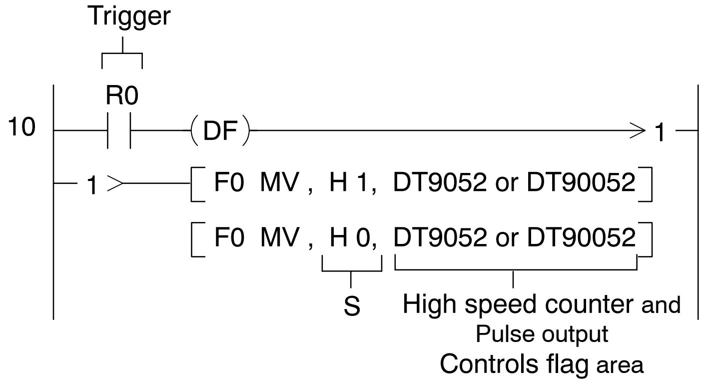
Availability
FP0/FP0R/FP-e/ FPΣ/FP-X

F0 (MV) Pulse output control

Outline This instruction is used to perform control such as software reset, counter disabling, and stopping pulse output.

Program example

Ladder Diagram



* The high-speed counter and pulse output controls flag area varies depending on the PLC type.

Boolean Address	Instruction		
10	ST	R	0
11	DF		
12	F0		(MV)
	H		1
	DT		9052
17	F0		(MV)
	H		0
	DT		9052

S	Area for storing pulse output control code or constant data

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX (*1)	IY (*2)	K	H	
S	A	A	A	A	A	A	A	A	A	A	A

A: Available

(*1) I0 to IC on FPΣ/FP-X/FP0R

(*2) ID on FPΣ/FP-X/FP0R

Description

Performs Pulse output control according to the control code specified in “S”.

This instruction is used to perform the following operations when using a Pulse output:

<Function>

1) Performing a software reset
2) Disabling the count
3) Preemptively stopping positioning/pulse output
4) Clearing control executed with pulse output-related instructions **F171** or **F176.**
5) Setting near home input when returning to home position and changing to deceleration.

Once written, a control code is retained until the next write operation.

Precautions during programming

The near home processing is not possible when the count is prohibited during a return to home position, or when a software reset is performed.

The near home bit is retained; however, each time you wish to perform near home processing during a return to home position, “1” must be written to the respective bit.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used
  - The “S” is outside specification range

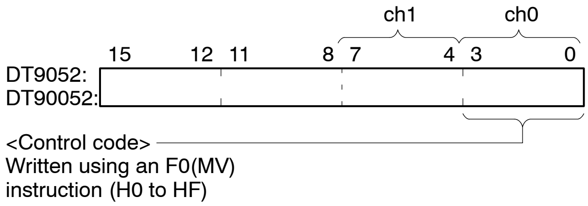
Using the FP0/FP−e

High−speed counter and Pulse output controls flag area

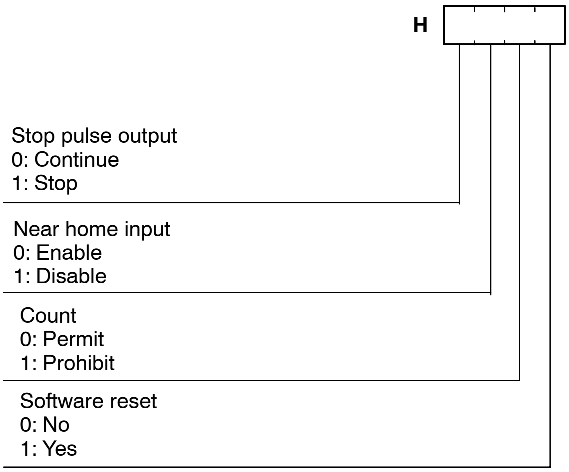
Four bits are allocated to each Pulse output channel for use as the control code write area DT9052 (DT90052 on the FP0 T32)

A control code written using an **F0(MV)** instruction is stored in special data register DT9052 (DT90052 on the FP0 T32).

High−speed counter and Pulse output controls flag area of FP0



Select control codes in units of one bit and specify with H.

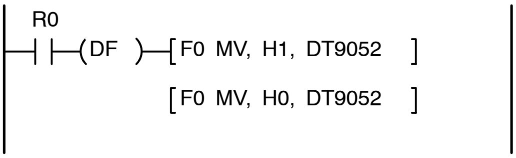


Example:

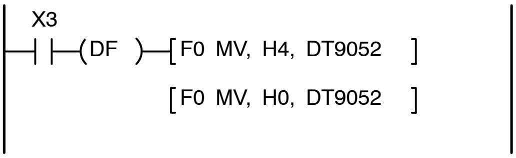
- **Perform software reset . . . . H1(0001)**
- **Prohibit count . . . . H2(0010)**
- **Stop pulse output . . . . H8(1000)**
- **Turn off pulse output and reset elapsed value . . H9(1001)**

Program example

Example 1: Software reset of channel 0 of Pulse output.



Example 2: Enable near home input during pulse output control and change to deceleration.



Using the FPΣ

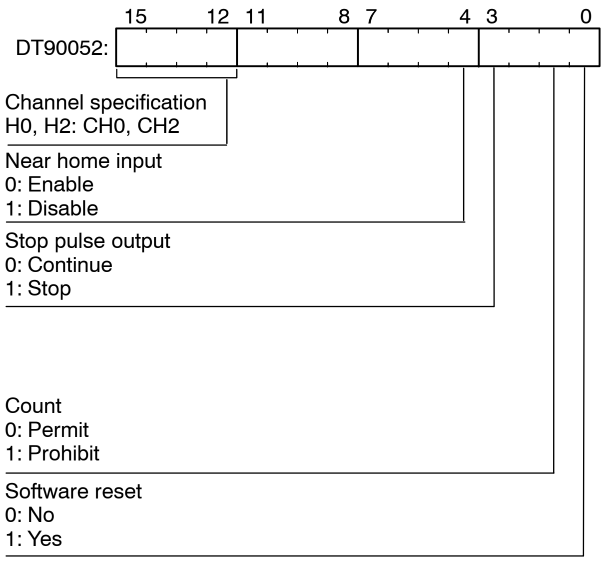
High−speed counter and Pulse output controls flag area

The area DT90052 for writing channels and control codes is allocated as shown below.

The control code written by the **F0 (MV)** instruction is stored in the control code monitor area while it is written in the special register DT90052. (Refer to the table below.)

The written data is the data for lower 8 bits only.

High−speed counter and Pulse output controls flag area of FPΣ



Program example

Example 1: Software reset of Pulse output

(ch0)

```
 R0
─┤ ├─(DF )─[F0 MV, H1, DT90052 ]
           [F0 MV, H0, DT90052 ]
```

(ch2)

```
 R0
─┤ ├─(DF )─[F0 MV, H2001, DT90052 ]
           [F0 MV, H2000, DT90052 ]
```

Example 2: Enable near home input during pulse output control and change to deceleration.

(ch0)

```
 X3
─┤ ├─(DF )─[F0 MV, H10, DT90052 ]
           [F0 MV, H0, DT90052 ]
```

(ch2)

```
 X3
─┤ ├─(DF )─[F0 MV, H2010, DT90052 ]
           [F0 MV, H2000, DT90052 ]
```

Using the FP-X

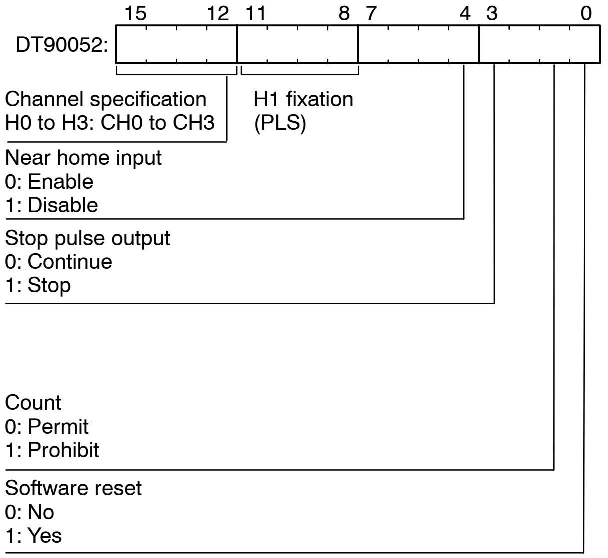
High-speed counter and Pulse output controls flag area

The area DT90052 for writing channels and control codes is allocated as shown below.

The control code written by the **F0 (MV)** instruction is stored in the control code monitor area while it is written in the special register DT90052. (Refer to the table below.)

The written data is the data for lower 8 bits only.

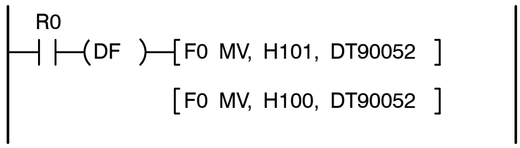
High-speed counter and Pulse output controls flag area of FP-X



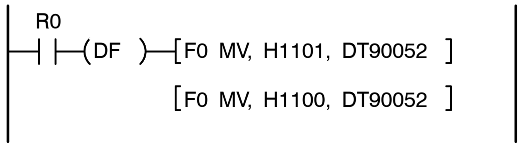
Program example

Example 1: Software reset of Pulse output

(ch0)

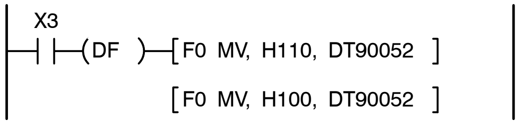


(ch1)

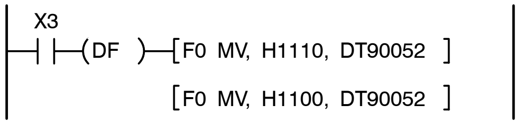


Example 2: Enable near home input during pulse output control and change to deceleration.

(ch0)



(ch1)



Using the FP0R

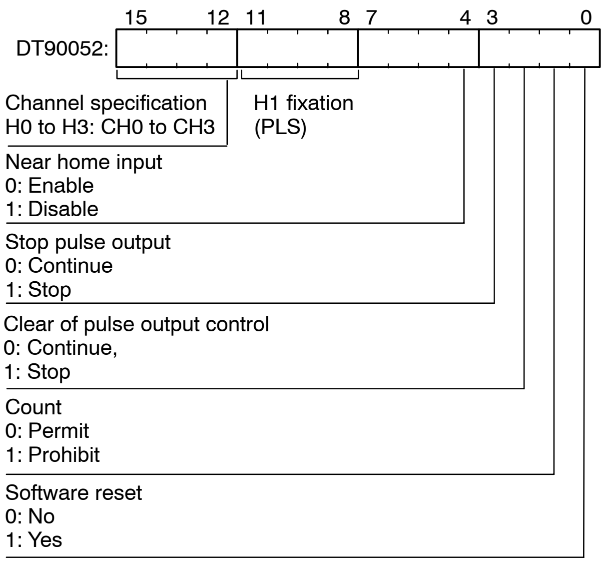
High−speed counter and Pulse output controls flag area

The area DT90052 for writing channels and control codes is allocated as shown below.

The control code written by the **F0 (MV)** instruction is stored in the control code monitor area while it is written in the special register DT90052. (Refer to the table below.)

The written data is the data for lower 8 bits only.

High−speed counter and Pulse output controls flag area of FP0R



* The pulse output control is available when controlling the pulse output ch with F166(HC1S) or F167(HC1R) instruction.

Program example

Refer to the program example of FP−X.

Pulse output control for FPΣ and FP−X

Channel No.	Control code monitor area			
	FPΣ	FP−X Ry type	FP−X Tr type	FP0R
ch0	DT90190	DT90372	DT90380	DT90380
ch1	—	DT90373	DT90381	DT90381
ch2	DT90192	—	DT90382	DT90382
ch3	—	—	DT90383	DT90383

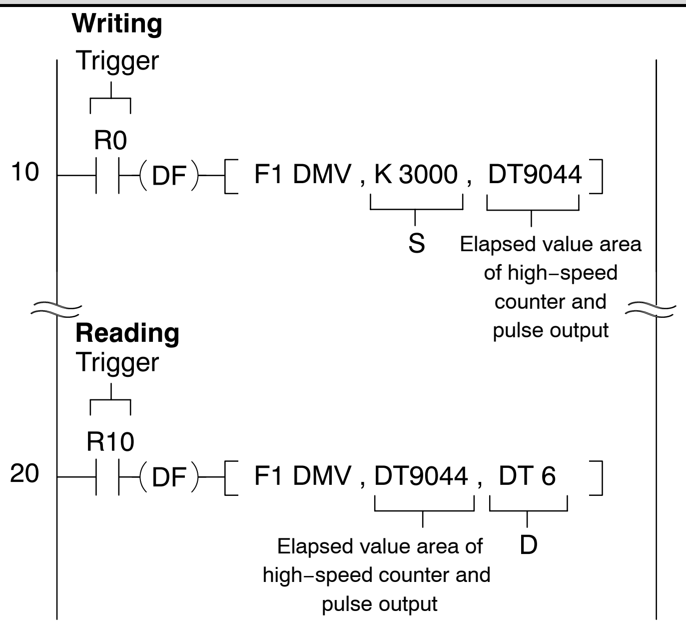
F1 (DMV) Writing and reading the high-speed counter and pulse output elapsed value

Availability
FP0/FP0R/FP-e/ FPΣ/FP-X

Outline This instruction is used to write and read the elapsed value of the high-speed counter/pulse output.

Program example

Ladder Diagram



* The high-speed counter and pulse output elapsed value area varies depending on the PLC type.

Boolean

Address	Instruction		
10	ST	R	0
11	DF		
12	F1		(DMV)
	K		3000
	DT		9044
:	:		
:	:		
20	ST	R	10
21	DF		
22	F1		(DMV)
	DT		9044
	DT		6

S	Writing Area for storing the elapsed value (32 bits) write in the high-speed counter/pulse output, or constant data
D	Reading Area for reading the elapsed value of the high-speed counter/pulse output

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX (*1)	IY (*2)	K	H	
S	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

(*1) I0 to IC on FPΣ/FP-X/FP0R

(*2) ID on FPΣ/FP-X/FP0R

Writing the elapsed value

This instruction writes the 32−bit data specified in “S” to the elapsed value area of the high−speed counter and pulse output channel being used, and simultaneously sets the data in the elapsed value area of the high−speed counter used inside the system.

Make sure the 32−bit data value that is written to the elapsed value is within the following range.

Type	Allowed setting range
FP0/FP−e	K −8,388,608 to K 8,388,607
FPΣ/FP−X/FP0R	K −2,147,483,648 to K 2,147,483,647

Writing is only possible using an **F1(DMV)** instruction. Writing is not possible using other applied instructions such as the transfer instruction **F0(MV)** or arithmetic instructions.

When specifying the memory area in “S” or “D” (when reading), specify only the lower−order 16 bits of the memory area number.

Explanation of example

When the execution condition R0 is on, K3000 is written to the elapsed value area of ch0 of the high−speed counter and pulse output.

Reading the elapsed value

The contents of the special data register that stores the elapsed value of the high−speed counter and pulse output is written to the area specified in “D”.

Explanation of example

When the execution condition R10 is on, the elapsed value of the high−speed counter and pulse output is transferred to data registers DT6 and DT7.

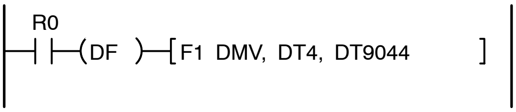
Flag conditions

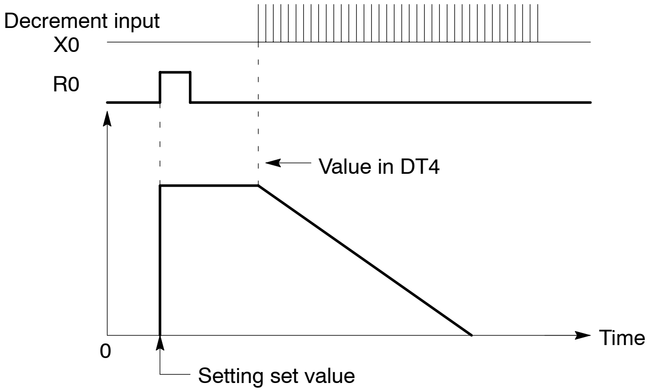
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used
  - The “S” is outside specification range

Program examples

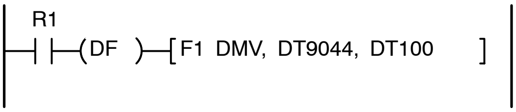
The elapsed value area varies depending on the model and channel number.

Example 1: On R0 input, the value in data register DT4 is set in the ch0 elapsed value area as the set value.





Example 2: On R1 input, the elapsed value of the ch0 is stored in data register DT100.



Example 3: When the elapsed value of the ch0 is greater than K10000, the internal relay R0 turns on.

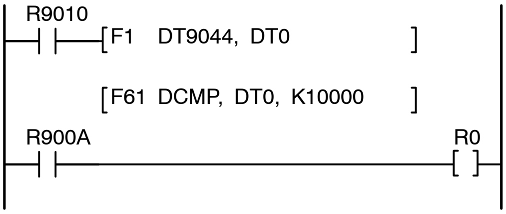


Table of channel number and elapsed value area

For FP0/FP−e

High−speed counter channel no.	Pulse output channel no.	Elapsed value area
ch0	ch0	DT9044 to DT9045
ch1	ch1	DT9048 to DT9049
ch2	–	DT9104 to DT9105
ch3	–	DT9108 to DT9109

For FP0(T32)

High−speed counter channel no.	Pulse output channel no.	Elapsed value area
ch0	ch0	DT90044 to DT90045
ch1	ch1	DT90048 to DT90049
ch2	–	DT90104 to DT90105
ch3	–	DT90108 to DT90109

For FPΣ

High−speed counter channel no.	Pulse output channel no.	Elapsed value area
ch0	ch0	DT90044 to DT90045
ch1	–	DT90048 to DT90049
ch2	ch2	DT90200 to DT90201
ch3	–	DT90204 to DT90205

For FP−X Ry type

High−speed counter channel no.	Pulse output channel no.	Elapsed value area
ch0	–	DT90300 to DT90301
ch1	–	DT90304 to DT90305
ch2	–	DT90308 to DT90309
ch3	–	DT90312 to DT90313
ch4	–	DT90316 to DT90317
ch5	–	DT90320 to DT90321
ch6	–	DT90324 to DT90325
ch7	–	DT90328 to DT90329
ch8	–	DT90332 to DT90333
ch9	–	DT90336 to DT90337
chA	–	DT90340 to DT90341
chB	–	DT90344 to DT90345
–	ch0	DT90348 to DT90349
–	ch1	DT90352 to DT90353

For FP−X Tr type

High−speed counter channel no.	Pulse output channel no.	Elapsed value area
ch0	−	DT90300 to DT90301
ch1	−	DT90304 to DT90305
ch2	−	DT90308 to DT90309
ch3	−	DT90312 to DT90313
ch4	−	DT90316 to DT90317
ch5	−	DT90320 to DT90321
ch6	−	DT90324 to DT90325
ch7	−	DT90328 to DT90329
−	ch0	DT90348 to DT90349
−	ch1	DT90352 to DT90353
−	ch2	DT90356 to DT90357
−	ch3	DT90360 to DT90361

For FP0R

High−speed counter channel no.	Pulse output channel no.	Elapsed value area
ch0	−	DT90300 to DT90301
ch1	−	DT90304 to DT90305
ch2	−	DT90308 to DT90309
ch3	−	DT90312 to DT90313
ch4	−	DT90316 to DT90317
ch5	−	DT90320 to DT90323
−	ch0	DT90400 to DT90401
−	ch1	DT90410 to DT90411
−	ch2	DT90420 to DT90421
−	ch3	DT90430 to DT90431

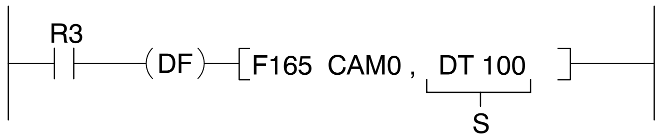
F165(CAM0) Cam control (High−speed counter control)

Availability
FP0R

Outline This instruction enables the control according to the maximum of 31−point target values for the high−speed counter.
[Feature] An interrupt program can be also executed whenever the elapsed value reaches each target value.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	3
11	DF		
12	F165		(CAM0)
	DT		100

S	Starting 16−bit area

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
S	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

It notifies that the elapsed value has reached a maximum of 31 target values in the pattern specified with the data table starting with the address specified by [S]. The internal relays corresponding to the target positions are turned on.
Also, the interrupt program INTn can be executed at the target position.
*It is necessary to allow the execution of the interrupt program with ICTL instruction.

1. Check that the same value is not used for the target values of the control table and they are arranged in ascending order.
2. Judge the position of the current value of high−speed counter, set 1 to the corresponding bit in the target position notification area of the internal relay, and clear the others to 0.
3. After that, in case of addition, the target position notification internal relay changes every time the elapsed value matches the target values.
   However, in case of subtraction, (Target value − 1) is used as the target position data.

[When the maximum value control is not performed]

When the maximum target value is 0, and the reset input is not permitted,

4. When the elapsed value matches the maximum target position ”m” in add operation, the next target position will be the negative minimum value.
5. When the elapsed value matches the minimum (target position 1−1) in subtraction operation, the target position will be the positive maximum value.

The control with the maximum target value is available as well as the above control operation.
The maximum target value can be specified at the end of the table. When the elapsed value matches the maximum target value, the elapsed value is cleared to 0, and The beginning of the internal relay in the position notification area is turned on.

To perform the control with the maximum target value, positive integer numbers must be specified for all the target position data.

[When the maximum value control is performed]

Using the maximum target value of data table or hardware/software reset signal enables the value to return to the starting address of data table. (V1.06 or later)

In add operation, the elapsed value will be cleared to 0 when it reaches the maximum target value (when reset signal is detected), and the starting bit of position notification relay will be turned on.
In subtraction operation, when the elapsed value reaches −1, the maximum target value will be set as the elapsed value, and the bit corresponding to the target position "m" will be turned on for position output.

Note: Hardware reset is CH0: X2, CH1: X2, CH2: X5, CH3: X5.

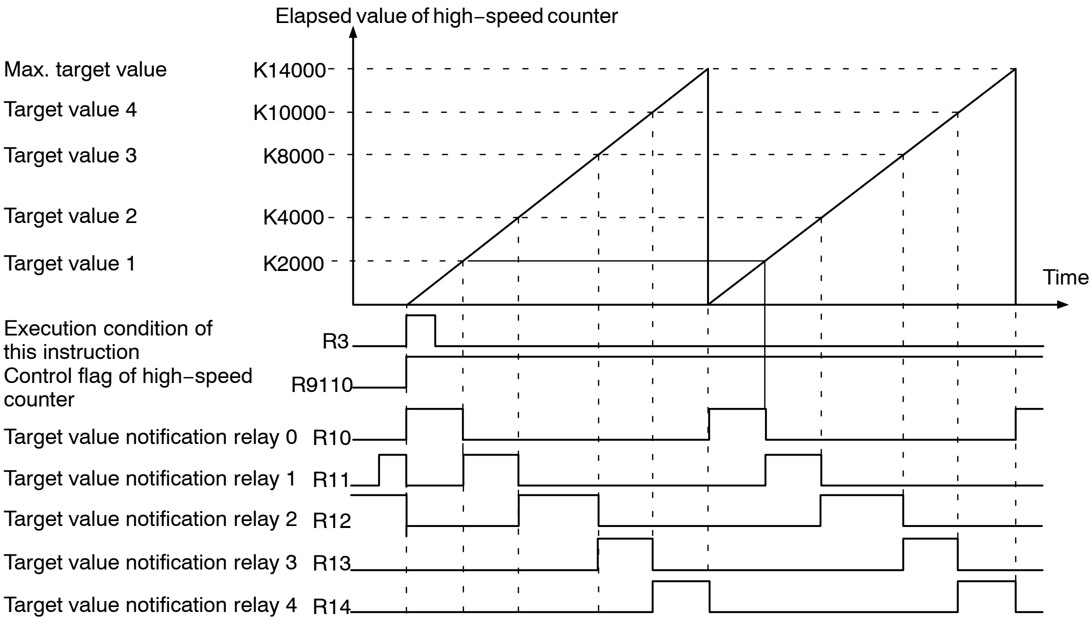
Description of hardware reset signal operation

	When the maximum value control is not performed	When the maximum value control is not performed
V1.06 or later	The elapsed value is cleared to 0, and the table pointer moves to the beginning.	Only the elapsed value is cleared to 0.
V1.05 or older	Only the elapsed value is cleared to 0.	Only the elapsed value is cleared to 0.

When the maximum value control has been specified, set the maximum target value to a large value which cannot be reached for returning the value to the starting address of data table using the hardware reset signal.

Sample operation: When controlling the high−speed counter CH0 with the maximum target value

Example: If the instruction is executed on the leading edge of the trigger R3, and the elapsed value when the execution is started is smaller than the target position 1
The target position notification area is specified from R10.



Precautions during programming

To use this instruction, the high−speed counter function must be used.

The high−speed counter control flag (R9110 to R9115) corresponding to the specified channel turns on when the execution condition of F165(CAM0) instruction turns on until the cam control is cleared.

When the high−speed counter control flag (R9110 to R9115) is on, the high−speed counter control instructions (F166(HC1S), F167(HC1R), F178(PLSM)) to the high−speed counter of the same channel cannot be executed.

To stop the control with this instruction, execute “Clears high−speed counter instruction”.

If the elapsed value to be controlled with this instruction is rewritten, an unexpected operation might be performed.

When controlling with the main program, set the target value to be “Minimum travelling time between each target value” greater than ”1 scan time”.

When controlling with a interrupt program, set the target value to be “Minimum travelling time between each target value” greater than “Maximum execution time of interrupt program”.

Set the interval between adjacent target values within 1msec not to match them.

This instruction can be simultaneously executed for 2 channels only.

When the maximum value control and the hardware/software reset is used at the same time, do not operate them intensively in a short time.

When hardware/software reset is used, set the first target value to an integer value that is 1 or more.

Channels of high−speed counter and areas used

High−speed counter channel No.	Control flag	Elapsed value area	Target value area	Interrupt program
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303	INT0
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307	INT1
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311	INT3
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315	INT4
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319	INT6
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323	INT7

Setting of data table

S S+1	Specification of high−speed counter channel	(*1)	
S+2 S+3	Word No. of internal relay for position notification	(*2)	
S+4 S+5	Specification of No. of target position m	(*3)	A maximum of 31 target positions can be specified.
S+6 S+7	Target position 1	(*4)	
S+8 S+9	Target position 2	(*4)	
S+10 S+11	Target position 3	(*4)	
	Target position m−1	(*4)	
	Target position m	(*4)	
	Max. target value	(*5)	(Note) Arrange the target values of the target positions 1 to m in ascending order. The same value cannot be specified.

*1: Specification of high−speed counter channel

Specify the channel of the high−speed counter/pulse output with H constant in the starting area (2 words) of the data table.

H 0 0 0 $\boxed{0}$ 0 0 0 $\boxed{1}$

<Specification of the channel of high−speed counter/pulse output>
Allowable range for specifying the high−speed counter: 0 to 5

Specification of maximum value control: 0 = Not control with the maximum target value
1 = Control with the maximum target value

<Specification of high−speed counter/pulse output>
0: Specification of high−speed counter

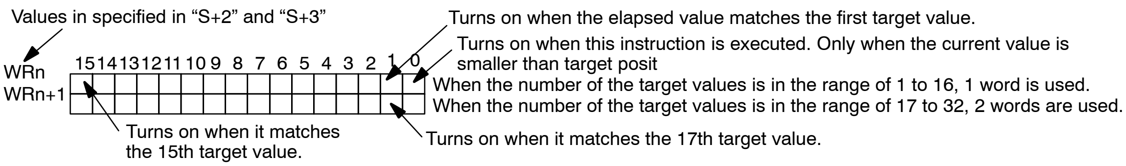
*2: Specify the word number of the internal relay where the target position is output. (Note1)

In the 3rd word and 4th word areas, specify the word number of the internal relay where the target position is output.

*3: Specification of the number of target positions (Note1)

Specify the number of target positions. Settable range: K1 to K31

<Method of target position notification>



*4: Specification of target position: Specify the target position after the 5th word.

Settable range: K−2147483648 to K2147483647
(H80000000 to H7FFFFFFF)

*5: Specification of maximum target value: Specify the maximum target value in the next address of the target value at the final target position.

Settable range: K−2147483648 to K2147483647
(H80000000 to H7FFFFFFF)

The interrupt program INTn corresponding to the specified high−speed counter channel can be executed at the target position.
The interrupt program corresponding to the channel to be controlled is programmed.
After this instruction is executed, interrupt will be permitted with ICTL instruction.

Note1) Specify numbers so that the total of them does not exceed the maximum area of the internal relay.

Example of setting 1

[Condition]

(1) Target values: 4 points   Position output from R10

(2) Each target value is as the table below.

Position output	Target value
1 (R11)	2000
2 (R12)	4000
3 (R13)	8000
4 (R14)	10000

(3) The maximum value is 14000 pulses.

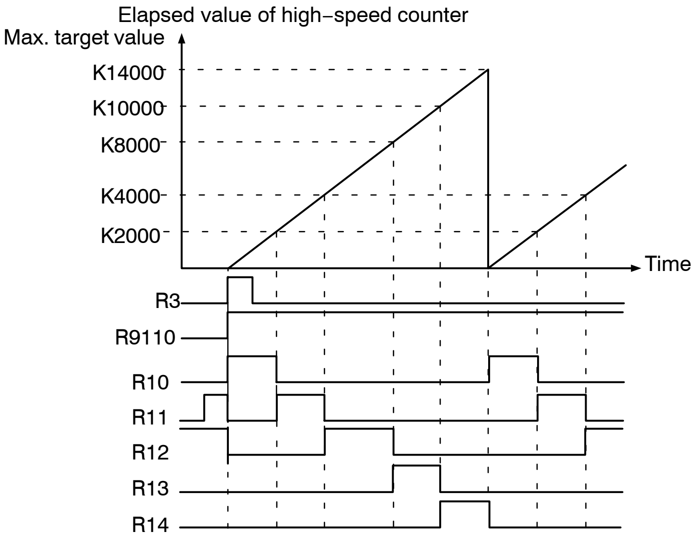
(4) The elapsed value of the high−speed counter is cleared to 0 before starting the position output.

Program

```
 R9013
─┤ ├──[ F1 DMV , H      10,  DT100  ]   Specification of high−speed counter channel 0
      [ F1 DMV , H       1,  DT102  ]   Specification of internal relay word No.
      [ F1 DMV , H       4,  DT104  ]   4−point output
      [ F1 DMV , K    2000,  DT106  ]   Target value 1
      [ F1 DMV , K    4000,  DT108  ]   Target value 2
      [ F1 DMV , K    8000,  DT110  ]   Target value 3
      [ F1 DMV , K   10000,  DT112  ]   Target value 4
      [ F1 DMV , K   14000,  DT114  ]   Max. target value
      [ F1 DMV , K       0,  DT90300]   Reset of elapsed value
 R3
─┤ ├──( DF )──[ F165 CAM0, DT100 ]      Start of cam control
```

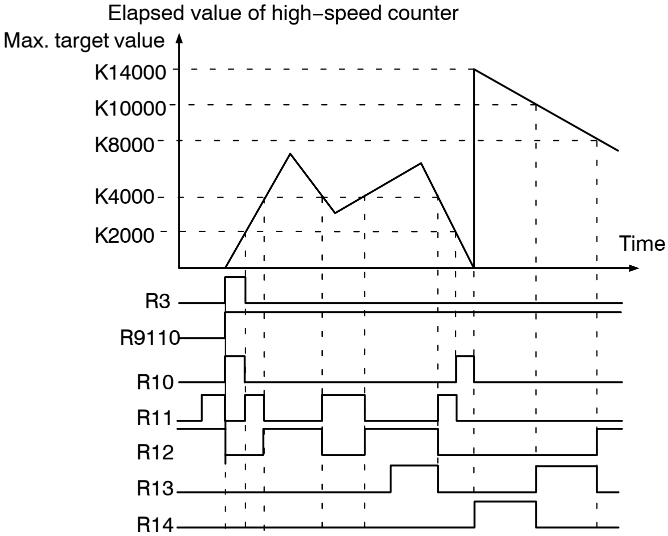
Explanation of program operation

When adding elapsed values with the maximum target value
When the internal relay R3 is on, the operation is as follows.



Explanation of program operation

When adding + subtracting elapsed values with the maximum target value
When the internal relay R3 is on, the operation is as follows.



Example of setting 2

[Condition]

(1) Cam output: 4 points Output from R10 to R13

(2) The target values for each cam are as the table below.

Cam output	Target value
1 (R11)	−10000
2 (R12)	−4000
3 (R13)	4000
4 (R14)	8000

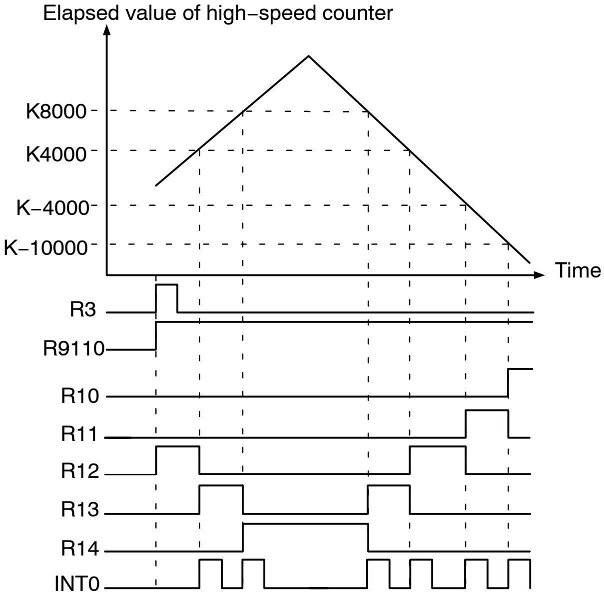
Program

```
R9013
─┤ ├──[ ICTL,     H        0,  H 1     ]   INT0 interrupt for CH0 is permitted.
R9013
─┤ ├──[ F1 DMV ,  H        0,  DT100   ]   Specification of high−speed counter channel 0
      [ F1 DMV ,  H        1,  DT102   ]   Specification of internal relay word No.
      [ F1 DMV ,  H        4,  DT104   ]   4−point output
      [ F1 DMV ,  K   −10000,  DT106   ]   Target value 1
      [ F1 DMV ,  K    −4000,  DT108   ]   Target value 2
      [ F1 DMV ,  K     4000,  DT110   ]   Target value 3
      [ F1 DMV ,  K     8000,  DT112   ]   Target value 4
      [ F1 DMV ,  K        0,  DT114   ]   Max. target value
      [ F1 DMV ,  K        0,  DT90300 ]   Reset of elapsed value
R3
─┤ ├──( DF )──[ F165 CAM0,  DT100 ]         Start of cam control
```

Explanation of program operation

When adding + subtracting elapsed values with the maximum target value and interrupt control, the operation will be performed as below if the following conditions are met; Elapsed value when the instruction is executed: K−4000 < Elapsed value < K4000   Internal relay R3: ON

*The operation can be started once the interrupt program is permitted to be started with ICTL instruction.



* The execution time in the interrupt program should be shorter than the travelling time between control positions.

Flag conditions

- Error flag (R9007):
- Error flag (R9008):
  - Turns on when the area specified using the index modifier exceeds the limit.
  - Turns on when the specified channel is out of the setting range.
  - Turns on when any other setting than the high−speed counter/pulse output is specified.
  - Turns on when the word number of the internal relay where the target value is output is out of the setting range.
  - Turns on when the specification of the number of target values exceeds the limit. (Up to 31)
  - Turns on when the garge value is greater than the maximum target value.
  - Turns on when the target value is 0.
  - Turns on when the targe values are not arranged in asceding order.
  - Turns on ending 16−bito area used for this instruction exceeds the limit of data table.
  - Turns on when the high−speed counter has not been set for the specified channel by the system register.

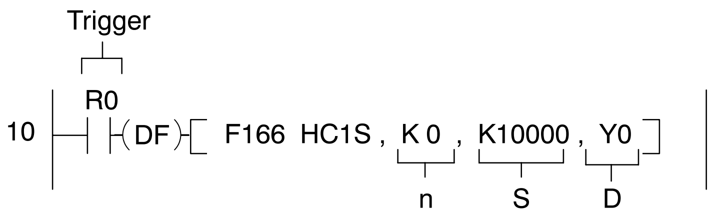
F166 (HC1S) Target value match on (with channel specification)

Availability
FP0/FP0R/FP−e/ FPΣ/FP−X

Outline When the elapsed value of the specified channel of the high−speed counter matches the target value, the specified output is turned on.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	DF
12	F166 (HC1S)
	K 0
	K 10000
	Y 0

n	The channel number of the high−speed counter that corresponds to the match output (FP0/FPΣ: H0 to H3, FP−X: H0 to HB).
S	The high−speed counter target value data or the starting address of the area that contains the data.
D	The output coil that is turned on when the values match (Yn).

Operands

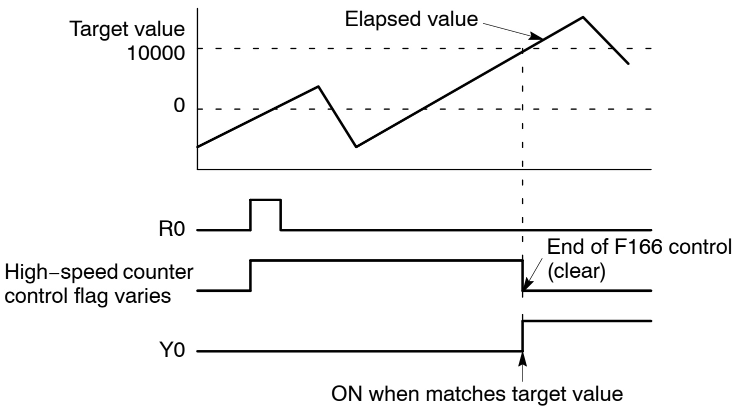
Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX (*1)	IY (*2)	K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A
S	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

A: Available
N/A: Not Available

(*1) I0 to IC on FPΣ/FP−X/FP0R

(*2) ID on FPΣ/FP−X/FP0R

Explanation of example



High−speed counter control flag varies

FP0, FPΣ, FP−e	R903A
FP−X, FP0R	R9110

(Refer to next page)

The number of the high−speed counter control flag varies depending on the channel used.

Regarding the channel number and control flag for each model, refer to the table on the next page.

Description

The number specified in “S” is set as the target value of the high−speed counter, and when the elapsed value matches the target value, the specified output “Yn” turns on (by interrupt processing).

The target value setting and target value match output control are cleared when the elapsed value matches the target value.

Specify a 32−bit data value for the target value “S” within the following range:
FP0/FP−e K−8,388,608 to K8,388,607
FPΣ/FP−X−FP0R K−2,147,483,648 to K2,147,483,647

The “S” value is stored in the target value area when the instruction is executed.

Possible specification range for “Yn”:

Devices specified for the match ON/OFF output

Type	Device area
FP0/FP−e	Y0 to Y7
FPΣ	Y0 to Y7
FPΣ (V3.10 or more)	Y0 to Y1F
FP−X	Y0 to Y29F

However, for the device that is not implemented, only the memory turns ON/OFF.

However, when the output that is not implemented is specified, only the WY memory is set/reset.

Precautions during programming

Set the high−speed counter by the system register before using this instruction.

The high−speed counter control flag turns on when the execution condition of the **F166(HC1S)** instruction turns on and remains on until the target value match output turns on. During this time, an instruction to the high−speed counter of the same channel (**F166** through **F176**) cannot be executed.

Before the elapsed value matches the target value, the target value and target value match output setting are not cleared even if a hardware reset is performed (the elapsed value is cleared to “0”).

A check for double output with **OT** instructions, **KP** instructions, and other applied instructions is not performed on the output Y that is specified for target value match output.

To turn off the target value match output that was turned on with this instruction, reset using an **RST** instruction or **F0(MV)** instruction, or use as a pair with an **F167(HC1R)** instruction.

If both the normal program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

The high−speed counter control flag also changes during scanning.

The interrupt program is able to be executed, when the high−speed counter elapsed value equals the set target value.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used.
  - The “n” is outside specification range.
  - The “S” is outside specification range.
  - The “D” is outside specification range.
  - The high−speed counter has not been set for the specified channel by the system register.

FP0, FP−e

Channel No.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT9044 to DT9045	DT9046 to DT9047
ch1	R903B	DT9048 to DT9049	DT9050 to DT9051
ch2	R903C	DT9104 to DT9105	DT9106 to DT9107
ch3	R903D	DT9108 to DT9109	DT9110 to DT9111

FPΣ

Channel No.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT90044 to DT90045	DT90046 to DT90047
ch1	R903B	DT90048 to DT90049	DT90050 to DT90051
ch2	R903C	DT91200 to DT91201	DT91202 to DT91203
ch3	R903D	DT91204 to DT91205	DT91206 to DT91207

FP−X Ry type:ch0 to chB T type : ch0 to ch7 FP0R: ch0 to ch5

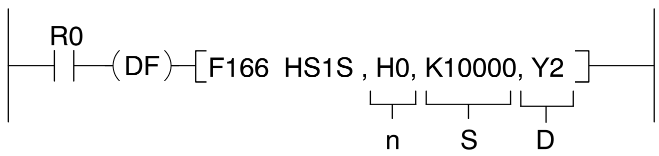
Channel No.	Control flag	Elapsed value area	Target value area
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323
ch6	R9116	DT90324 to DT90325	DT90326 to DT90327
ch7	R9117	DT90328 to DT90329	DT90330 to DT90331
ch8	R9118	DT90332 to DT90333	DT90334 to DT90335
ch9	R9119	DT90336 to DT90337	DT90338 to DT90339
chA	R911A	DT90340 to DT90341	DT90342 to DT90343
chB	R911B	DT90344 to DT90345	DT90346 to DT90347

Availability
FP0R

F166(HC1S) Target value match on (High−speed counter control)

Outline When the elapsed value of the specified channel of the high−speed counter (HSC) matches the target value, the specified output is turned on.

Program example



Address	Instruction		
10	ST	R	0
11	DF		
12	F166		(HC1S)
	H		0
	K		10000
	Y		2

n	The channel number of the high−speed counter that corresponds to the match output.
S	The high−speed counter target value data or the starting address of the area that contains the data.
D	The output coil that is turned on when the values match. (Yn)

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A
S	A	A	A	A	A	A	A	A	N/A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

The number specified in “S” is set as the target value of the high−speed counter, and when the elapsed value matches the target value, the specified output “Yn” turns on (by interrupt processing).

Specify the channel number of the high−speed counter in “n”.

H 0 $\boxed{0}$ 0 $\boxed{0}$

<Specification of the channel of high−speed counter/pulse output>
Allowable range for specifying the high−speed counter: 0 to 5

<Specification of high−speed counter/pulse output>
0: Specification of high−speed counter

The target value setting and target value match output control are cleared when the elapsed value matches the target value.

Specify a 32−bit data value for the target value “S” within the following range:
FP0R K−2,147,483,648 to K2,147,483,647

The “S” value is stored in the target value area when the instruction is executed.

Possible specification range for “Yn”: Devices specified for the match ON/OFF output

Type	Device area
FP0R	Y0 to Y1F

However, for the device that is not implemented, only the memory turns ON/OFF.

Example of target value match on setting

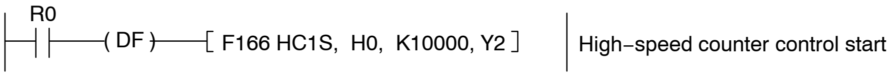
When specifying the high−speed counter

Condition

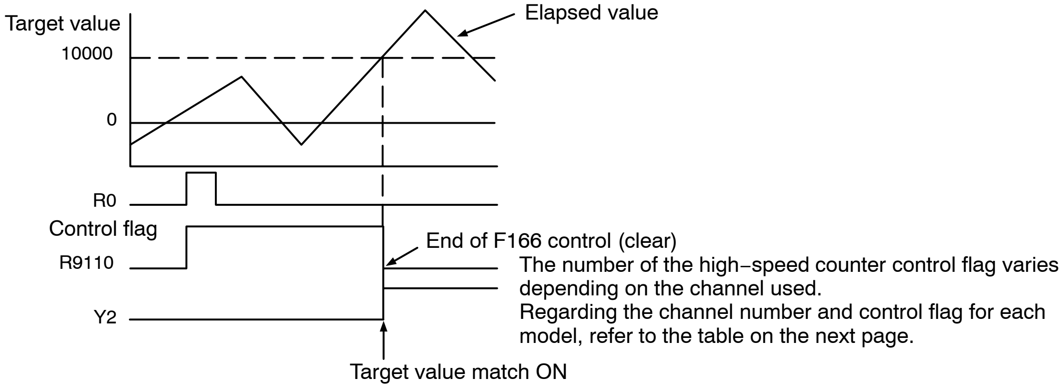
(1) Specify the high−speed counter channel number 0.

(2) Set the target value to 10000.

(3) Set the output coil to be turned off when the values match to Y2.



Execution of program



FP0R <In case of high−speed counter>

Channel No.	Control flag	Elapsed value area	Target value area	Interrupt program
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303	INT0
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307	INT1
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311	INT3
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315	INT4
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319	INT6
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323	INT7

Precautions during programming

Set the high−speed counter channel by the system register before using this instruction.
Without the setting, an operation error occurs.

The high−speed counter control flag (R9110 to R9115) turns on when the execution condition of this instruction turns on and remains on until the target value matches.

During this time, the high−speed counter control instructions (F165(CAM0), F166(HC1S), F167(HC1R), F178(PLSM)) to the high−speed counter of the same channel cannot be executed.

Before the elapsed value matches the target value, the target value and target value match output setting are not cleared even if a hardware reset is performed (the elapsed value is cleared to ”0”).

A check for double output with OT instructions, KP instructions and other applied instructions is not performed on the output Y that is specified for target value match output.

If the control is cleared with F0(MV) S, DT90052 instruction, the control of this instruction is cancelled and the high−speed control flag turns off.

If both the normal program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

The interrupt program is able to be executed, when the high−speed counter elapsed value equals the set target value. The INT program description and the permission using ICTL instruction is required.

Flag conditions

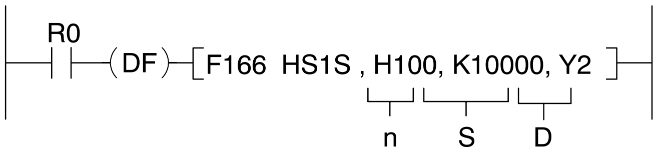
- Error flag (R9007):
- Error flag (R9008):
  - Turns on when the area specified using the index modifier exceeds the limit.
  - Turns on when n is out of the specified range.
  - Turns on when [S]n is out of the specified range.
  - Turns on when [D]n is out of the specified range.
  - Turns on when the high−speed counter has not been set for the specified channel by the system register.

F166(HC1S) Target value match on (Pulse output control)

Availability
FP0R

Outline When the elapsed value of the specified pulse output channel matches the target value, the specified output is turned on.

Program example



Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	DF
	12	F166 (HC1S)
		H 100
		K 10000
		Y 2

n	The channel number of the pulse output that corresponds to the match output.
S	The pulse output target value data or the starting address of the area that contains the data.
D	The output coil that is turned on when the values match. (Yn)

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A
S	A	A	A	A	A	A	A	A	N/A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

The number specified in “S” is set as the target value of the pulse output channel, and when the elapsed value matches the target value, the specified output “Yn” turns on (by interrupt processing).

Specify the channel number of the pulse output in “n”.

H 0 $\boxed{1}$ 0 $\boxed{0}$

<Specification of the channel of high−speed counter/pulse output>
Allowable range for specifying the pulse output: 0 to 3

<Specification of high−speed counter/pulse output>
1: Specification of pulse output

The target value setting and target value match output control are cleared when the elapsed value matches the target value. The control flag also turns off.

Specify a 32−bit data value for the target value “S” within the following range:
FP0R K−2,147,483,648 to K2,147,483,647

The “S” value is stored in the target value area when the instruction is executed.

Possible specification range for “Yn”: Devices specified for the match ON/OFF output

Type	Device area
FP0R	Y0 to Y1F

However, for the device that is not implemented, only the memory turns ON/OFF.

Example of target value match on setting

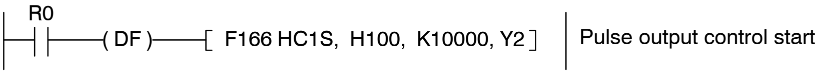
When specifying the pulse output

Condition

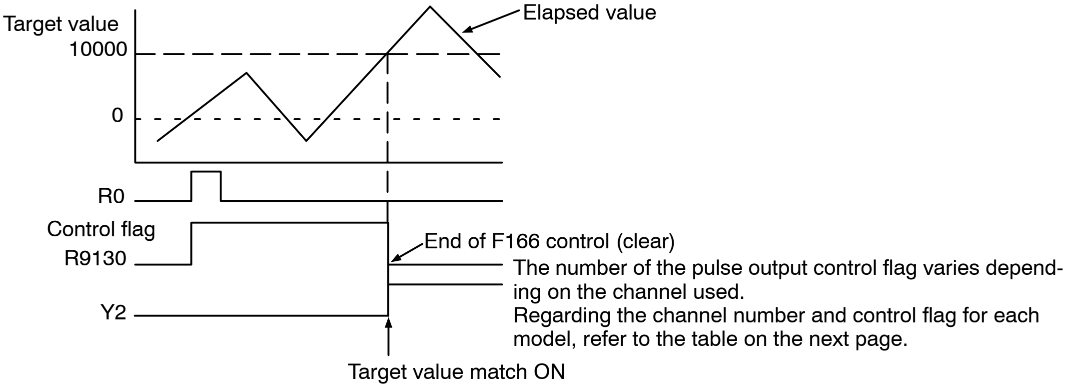
(1) Specify the pulse output channel number 0.

(2) Set the target value to 10000.

(3) Set the output coil to be turned off when the values match to Y2.



Execution of program



FP0R <In case of pulse output>

Channel No.	For pulse output			For pulse output control		
	Pulse output instruction flag	Elapsed value area	Target value area	Control flag	Target value area	Interrupt program
ch0	R9120	DT90400 to DT90401	DT90402 to DT90403	R9130	DT90404 to DT90405	INT8
ch1	R9121	DT90410 to DT90411	DT90412 to DT90413	R9131	DT90414 to DT90415	INT9
ch2	R9122	DT90420 to DT90421	DT90422 to DT90423	R9132	DT90424 to DT90425	INT10
ch3	R9123	DT90430 to DT90431	DT90432 to DT90433	R9133	DT90434 to DT90435	INT11

Precautions during programming

Set the pulse output channel by the system register before using this instruction.
Without the setting, an operation error occurs.

The pulse control flag (R9130 to R9133) turns on when the execution condition of this instruction turns on and remains on until the target value matches.
During this time, the pulse output control instructions (F165(CAM0), F166(HC1S), F167(HC1R)) to the pulse output of the same channel cannot be executed.

This instruction is available for all the pulse output instructions except F173(PWMH) instruction.
This instruction can be executed before or after the execution of the above pulse output instruction.

A check for double output with OT instructions, KP instructions and other applied instructions is not performed on the output Y that is specified for target value match output.

If the control is cleared with F0(MV) S, DT90052 instruction, the control of this instruction is cancelled and the high−speed control flag turns off, however, the pulse output continues.

If both the normal program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

The interrupt program is able to be executed, when the high−speed counter elapsed value equals the set target value. The INT program description and the permission using ICTL instruction is required.

Flag conditions

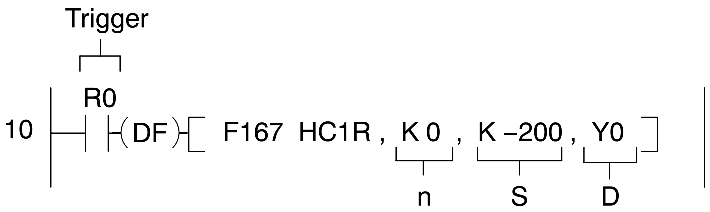
- Error flag (R9007):
- Error flag (R9008):
  - Turns on when the area specified using the index modifier exceeds the limit.
  - Turns on when n is out of the specified range.
  - Turns on when [S]n is out of the specified range.
  - Turns on when [D]n is out of the specified range.
  - Turns on when the pulse output has not been set for the specified channel by the system register.

Availability
FP0/FP0R/FP−e/ FPΣ/FP−X

F167 (HC1R) Target value match off (with channel specification)

Outline When the elapsed value of the specified channel of the high−speed counter matches the target value, the specified output is turned off.

Program example



Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	DF
	12	F167 (HC1R)
		K 0
		K − 200
		Y 0

n	The channel number of the high−speed counter that corresponds to the match output (FP0/FP−e/FPΣ: H0 to H3, FP−X: H0 to HB).
S	The high−speed counter target value data or the starting address of the area that contains the data.
D	The output coil that is turned off when the values match (Yn n: 0 to 7)

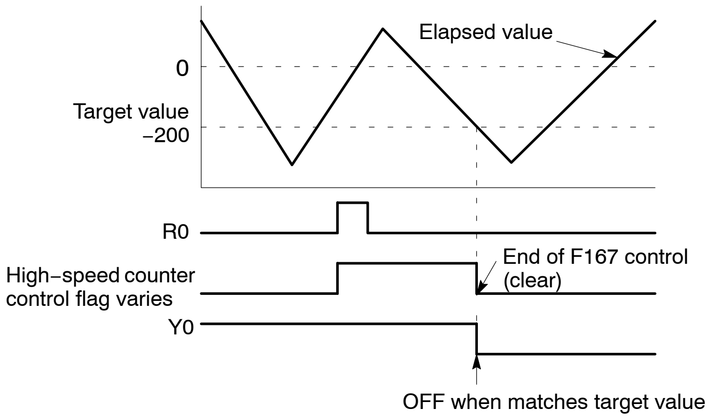
Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX (*1)	IY (*2)	K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A
S	A	A	A	A	A	A	A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

A: Available
N/A: Not Available

(*1) I0 to IC on FPΣ/FP−X/FP0R
(*2) ID on FPΣ/FP−X/FP0R

Explanation of example



High−speed counter control flag varies

FP0, FPΣ, FP−e	R903A
FP−X, FP0R	R9110

(Refer to next page)

The number of the high−speed counter control flag varies depending on the channel used.

Regarding the channel number and control flag for each model, refer to the table on the next page.

Description

The number specified in “S” is set as the target value of the high−speed counter, and when the elapsed value matches the target value, the specified output “Yn” turns off (by interrupt processing).

The target value setting and target value match output control are cleared when the elapsed value matches the target value.

Specify a 32−bit data value for the target value “S” within the following range:
FP0/FP−e K−8,388,608 to K8,388,607
FPΣ/FP−X/FP0R K−2,147,483,648 to K2,147,483,647

The “S” value is stored in the target value area when the instruction is executed.

Possible specification range for “Yn”:

Devices specified for the match ON/OFF output

Type	Device area
FP0/FP−e	Y0 to Y7
FPΣ	Y0 to Y7
FPΣ (V3.10 or more)/FP0R	Y0 to Y1F
FP−X	Y0 to Y29F

However, for the device that is not implemented, only the memory turns ON/OFF.

However, when the output that is not implemented is specified, only the WY memory is set/reset.

Precautions during programming

Set the high−speed counter by the system register before using this instruction.

The high−speed counter control flag turns on when the execution condition of the **F167(HC1S)** instruction turns on and remains on until the target value match output turns off. During this time, an instruction to the high−speed counter of the same channel (F166 through F173) cannot be executed.

Before the elapsed value matches the target value, the target value and target value match output setting are not cleared even if a hardware reset is performed (the elapsed value is cleared to “0”).

A check for double output with **OT** instructions, **KP** instructions, and other applied instructions is not performed on the output Y that is specified for target value match output.

To turn on the target value match output that was turned off with this instruction, reset using an **SET** instruction or **F0(MV)** instruction, or use as a pair with an **F166(HC1S)** instruction.

If both the normal program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

The high−speed counter control flag also changes during scanning.

The interrupt program is able to be executed, when the high−speed counter elapsed value equals the set target value.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used.
  - The “n” is outside specification range.
  - The “S” is outside specification range.
  - The “D” is outside specification range.
  - The high−speed counter has not been set for the specified channel by the system register.

FP0, FP−e

Channel No.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT9044 to DT9045	DT9046 to DT9047
ch1	R903B	DT9048 to DT9049	DT9050 to DT9051
ch2	R903C	DT9104 to DT9105	DT9106 to DT9107
ch3	R903D	DT9108 to DT9109	DT9110 to DT9111

FPΣ

Channel No.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT90044 to DT90045	DT90046 to DT90047
ch1	R903B	DT90048 to DT90049	DT90050 to DT90051
ch2	R903C	DT91200 to DT91201	DT91202 to DT91203
ch3	R903D	DT91204 to DT91205	DT91206 to DT91207

FP−X Ry type:ch0 to chB T type : ch0 to ch7 FP0R: ch0 to ch5

Channel No.	Control flag	Elapsed value area	Target value area
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323
ch6	R9116	DT90324 to DT90325	DT90326 to DT90327
ch7	R9117	DT90328 to DT90329	DT90330 to DT90331
ch8	R9118	DT90332 to DT90333	DT90334 to DT90335
ch9	R9119	DT90336 to DT90337	DT90338 to DT90339
chA	R911A	DT90340 to DT90341	DT90342 to DT90343
chB	R911B	DT90344 to DT90345	DT90346 to DT90347

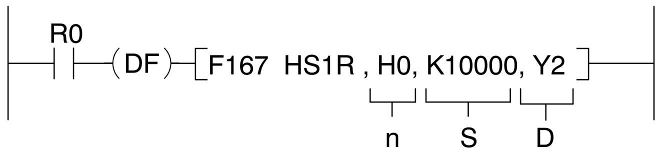
Availability
FP0R

F167(HC1R) Target value match off (High−speed counter control)

Outline When the elapsed value of the specified channel of the high−speed counter (HSC) matches the target value, the specified output is turned off.

Program example

Ladder Diagram



Boolean Address	Instruction		
10	ST	R	0
11	DF		
12	F167		(HC1R)
	H		0
	K		10000
	Y		2

n	The channel number of the high−speed counter that corresponds to the match output.
S	The high−speed counter target value data or the starting address of the area that contains the data.
D	The output coil that is turned on when the values match. (Yn)

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A
S	A	A	A	A	A	A	A	A	N/A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

The number specified in “S” is set as the target value of the high−speed counter, and when the elapsed value matches the target value, the specified output “Yn” turns off (by interrupt processing).

Specify the channel number of the high−speed counter in “n”.

H 0 $\boxed{0}$ 0 $\boxed{0}$

<Specification of the channel of high−speed counter/pulse output>
Allowable range for specifying the high−speed counter: 0 to 5

<Specification of high−speed counter/pulse output>
0: Specification of high−speed counter

The target value setting and target value match output control are cleared when the elapsed value matches the target value.

Specify a 32−bit data value for the target value “S” within the following range:
FP0R K−2,147,483,648 to K2,147,483,647

The “S” value is stored in the target value area when the instruction is executed.

Possible specification range for “Yn”: Devices specified for the match ON/OFF output

Type	Device area
FP0R	Y0 to Y1F

However, for the device that is not implemented, only the memory turns ON/OFF.

Example of target value match OFF setting

When specifying the high−speed counter

Condition

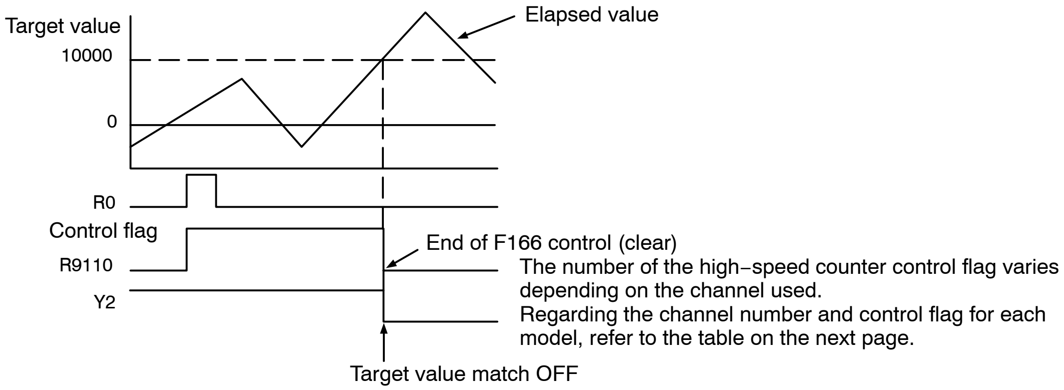
(1) Specify the high−speed counter channel number 0.

(2) Set the target value to 10000.

(3) Set the output coil to be turned off when the values match to Y2.

```
 R0
|-| |---( DF )---[ F167 HC1R,  H0,  K10000, Y2 ]   | High−speed counter control start
```

Execution of program



FP0R <In case of high−speed counter>

Channel No.	Control flag	Elapsed value area	Target value area	Interrupt program
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303	INT0
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307	INT1
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311	INT3
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315	INT4
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319	INT6
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323	INT7

Precautions during programming

Set the high−speed counter channel by the system register before using this instruction.
Without the setting, an operation error occurs.

The high−speed counter control flag (R9110 to R9115) turns on when the execution condition of this instruction turns on and remains on until the target value matches.
During this time, the high−speed counter control instructions (F165(CAM0), F166(HC1S), F167(HC1R), F178(PLSM)) to the high−speed counter of the same channel cannot be executed.

Before the elapsed value matches the target value, the target value and target value match output setting are not cleared even if a hardware reset is performed (the elapsed value is cleared to ”0”).

A check for double output with OT instructions, KP instructions and other applied instructions is not performed on the output Y that is specified for target value match output.

If the control is cleared with F0(MV) S, DT90052 instruction, the control of this instruction is cancelled and the high−speed control flag turns off.

If both the normal program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

The interrupt program is able to be executed, when the high−speed counter elapsed value equals the set target value. The INT program description and the permission using ICTL instruction is required.

Flag conditions

- Error flag (R9007):
- Error flag (R9008):

  – Turns on when the area specified using the index modifier exceeds the limit.
  – Turns on when n is out of the specified range.
  – Turns on when [S]n is out of the specified range.
  – Turns on when [D]n is out of the specified range.
  – Turns on when the high−speed counter has not been set for the specified channel by the system register.

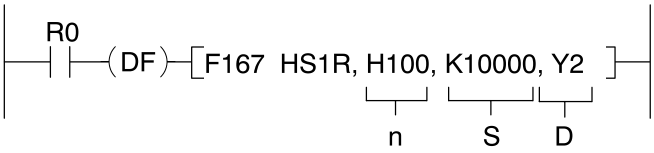
F167 (HC1R) Target value match off (Pulse output control)

Availability
FP0R

Outline When the elapsed value of the specified pulse output channel matches the target value, the specified output is turned off.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	DF
12	F167 (HC1R)
	H 100
	K 10000
	Y 2

n	The channel number of the pulse output that corresponds to the match output.
S	The pulse output target value data or the starting address of the area that contains the data.
D	The output coil that is turned on when the values match. (Yn)

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A
S	A	A	A	A	A	A	A	A	N/A	N/A	A	A	A
D	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

The number specified in “S” is set as the target value of the pulse output channel, and when the elapsed value matches the target value, the specified output “Yn” turns on (by interrupt processing).

Specify the channel number of the pulse output in “n”.

H 0 1 0 0

<Specification of the channel of high−speed counter/pulse output>
Allowable range for specifying the pulse output: 0 to 3

<Specification of high−speed counter/pulse output>
1: Specification of pulse output

The target value setting and target value match output control are cleared when the elapsed value matches the target value. The control flag also turns off.

Specify a 32−bit data value for the target value “S” within the following range:
FP0R K−2,147,483,648 to K2,147,483,647

The “S” value is stored in the target value area when the instruction is executed.

Possible specification range for “Yn”: Devices specified for the match ON/OFF output

Type	Device area
FP0R	Y0 to Y1F

However, for the device that is not implemented, only the memory turns ON/OFF.

Example of target value match OFF setting

When specifying the pulse output

Condition

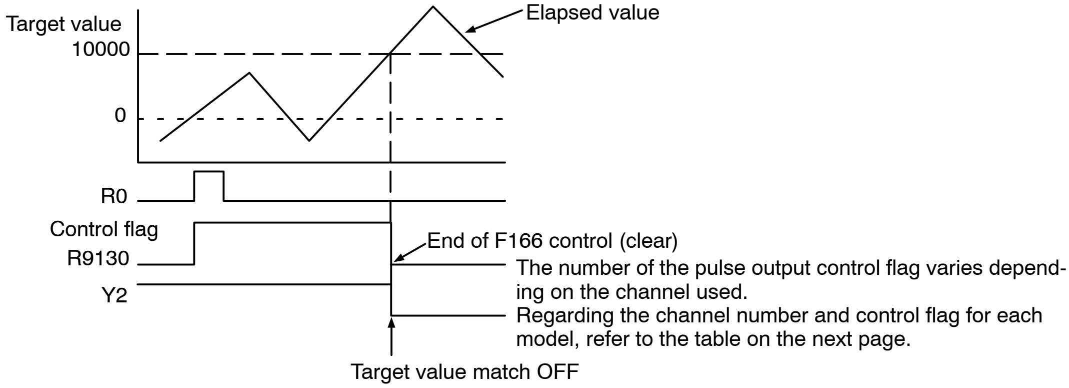
(1) Specify the pulse output channel number 0.

(2) Set the target value to 10000.

(3) Set the output coil to be turned off when the values match to Y2.

```
   R0
|--| |----( DF )----[ F167 HC1R,  H100,  K10000, Y2 ]  | Pulse output control start
```

Execution of program



FP0R <In case of pulse output>

Channel No.	For pulse output			For pulse output control		
	Pulse instruction in execution	Elapsed value area	Target value area	Control flag	Target value area	Interrupt program
ch0	R9120	DT90400 to DT90401	DT90402 to DT90403	R9130	DT90404 to DT90405	INT8
ch1	R9121	DT90410 to DT90411	DT90412 to DT90413	R9131	DT90414 to DT90415	INT9
ch2	R9122	DT90420 to DT90421	DT90422 to DT90423	R9132	DT90424 to DT90425	INT10
ch3	R9123	DT90430 to DT90431	DT90432 to DT90433	R9133	DT90434 to DT90435	INT11

Precautions during programming

Set the pulse output channel by the system register before using this instruction.
Without the setting, an operation error occurs.

The pulse control flag (R9130 to R9133) turns on when the execution condition of this instruction turns on and remains on until the target value matches.
During this time, the pulse output control instructions (F165(CAM0), F166(HC1S), F167(HC1R)) to the pulse output of the same channel cannot be executed.

This instruction is available for all the pulse output instructions except F173(PWMH) instruction.
This instruction can be executed before or after the execution of the above pulse output instruction.

A check for double output with OT instructions, KP instructions and other applied instructions is not performed on the output Y that is specified for target value match output.

If the control is cleared with F0(MV) S, DT90052 instruction, the control of this instruction is cancelled and the high−speed control flag turns off, however, the pulse output continues.

If both the normal program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

The interrupt program is able to be executed, when the high−speed counter elapsed value equals the set target value. The INT program description and the permission using ICTL instruction is required.

Flag conditions

- Error flag (R9007):
- Error flag (R9008):

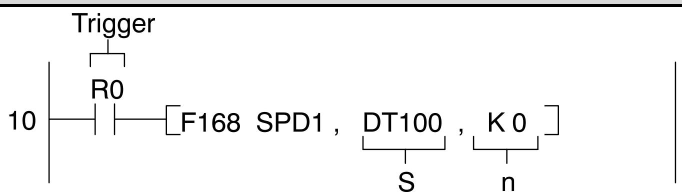
– Turns on when the area specified using the index modifier exceeds the limit.
– Turns on when n is out of the specified range.
– Turns on when [S]n is out of the specified range.
– Turns on when [D]n is out of the specified range.
– Turns on when the pulse output has not been set for the specified channel by the system register.

Availability
FP0/FP−e

F168 (SPD1) Positioning control (trapezoidal control)

Outline Outputs a pulse from the specified output (Y0 or Y1) according to the specified parameter.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F168 (SPD1) DT 100 K 0

S	Starting address for the area that contains the data table.
n	Output Yn that corresponds to the pulse output (n: K0 or K1).

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and the execution condition (trigger) is in the on state, a pulse is output from the specified output (Y0 or Y1).

The control code, initial speed, maximum speed, acceleration/deceleration time, and target value, are specified by a user program with a data table as shown on the following page.

The frequency is switched by the acceleration/deceleration time specified for changing from the initial speed to the maximum speed. During deceleration (normally 30 steps), the frequency is changed based on the same slope as during acceleration.

Table of areas used

Channel no.	Control flag	Elapsed value area	Target value area	Directional output
ch0	R903A	DT9044, DT9045 (For FP0 T32, DT90044, DT90045)	DT9046, DT9047 (For FP0 T32, DT90046, DT90047)	Y2
ch1	R903B	DT9048, DT9049 (For FP0 T32, DT90048, DT90049)	DT9050, DT9051 (For FP0 T32, DT90050, DT90051)	Y3

Notes

- **When this instruction is used, the setting for the channel corresponding to system register 400 should be set to “High−speed counter not used”.**
- **By performing rewrite during RUN during pulse output, more than the set number of pulses may be output.**

Description of operating mode

Incremental <relative value control>

Outputs the pulses set with the target value.

Target value \ Operation mode	Control code: H02 Forward off/Reverse on	Control code: H03 Forward on/Reverse off	Elapsed value
Positive	Pulse output on direction output off	Pulse output on direction output on	Addition
Negative	Pulse output on direction output on	Pulse output on direction output off	Subtraction

Absolute <absolute value control>

Outputs a number of pulses equal to the difference between the set target value and the current value.

Target value \ Operation mode	Control code: H12 Forward off/Reverse on	Control code: H13 Forward on/Reverse off	Elapsed value
Target value greater than current value	Pulse output on direction output off	Pulse output on direction output on	Addition
Target value less than current value	Pulse output on direction output on	Pulse output on direction output off	Subtraction

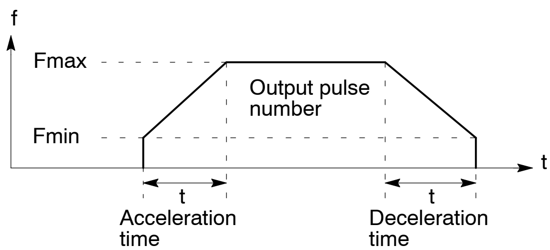
Precautions during programming

If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

Run the program referring to page 3 − 487, when controlling the motor in one direction using the pulse output function.

Data table settings

S	Control code	(*1)
S+1	Initial speed Fmin (Hz)	K40 to K5000 (Hz)
S+2	Maximum speed Fmax (Hz)	K40 to K9500 (Hz) (*2)
S+3	Acceleration/deceleration time t (ms)	K30 to K32767 (ms)
S+4 S+5	Target value (pulse number)	K−8,388,608 to K+8,388,607
S+6	K0	Specify “K0”



☛ next page

(*1): Specify the control code by setting the constant H.

H □ □ □

Pulse width specification
0: Duty 50%
1: Fixed pulse width (approx. 80μs)

Note: A specification of 2 or higher will result in 0.
The pulse width is the output value from the IC and the actual pulse width varies due to the delay in the response of photocoupler.

Operation mode and directional output theory

00:	Incremental	Does not use directional output
02:	Incremental	forward off/reverse on
03:	Incremental	forward on/reverse off
10:	Absolute	Does not use directional output
12:	Absolute	forward off/reverse on
13:	Absolute	forward on/reverse off

(*2): When the pulse width is set to duty 50%, the maximum is 6kHz.
When the pulse width is set to fixed pulse width (approx. 80μs), the maximum is 9.5kHz.
(Thermocouple input type of FP−e is removed.)

Notes for using FP0 compatiblity mode of FP0R

① The elapsed value and target value of high−speed counter and pulse output is signed 32−bit values.

② The high−speed counter continues counting even if data exceeds the FP0 range (signed 24−bit).

③ The pulse output continues outputting even if data exceeds the FP0 range (signed 24−bit).

④ The waveforms of pulse output are a duty cycle of 25% regardless of the designation of instructions.

⑤ Even if the no count setting is specified with a pulse output instruction, it counts in the addition mode.

⑥ The maximum frequency of pulse output is 10000Hz.

⑦ When using the pulse output instruction, it is not used for the pulse ouput and normal output.

Supplement to the operation in the case with the direction output

1: FP0

1: When specifying a duty of 50%: The pulse output will start approx. “Initial speed period/2” hours later after the direction output.
When the initial speed is 500Hz, it is approx. 1ms.

2: When specifying 80us fixedly: The pulse output will start approx. “Initial speed period − 25us” hours later after the direction output.
When the initial speed is 500Hz, it is approx. 1.98ms.

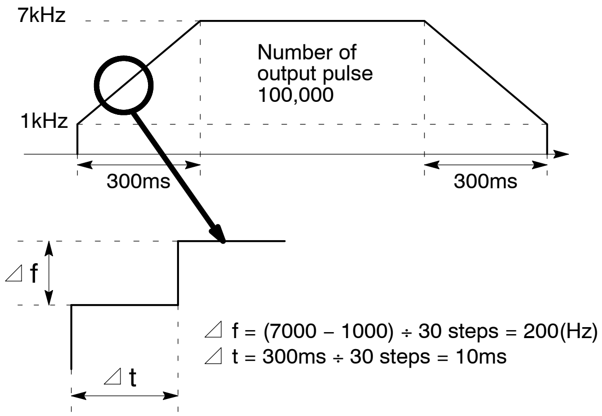
2: For FP0 compatibility mode

Pulses are output using a duty of 25% fixedly. (The setting is invalid.)
The pulse output will start approx. 300us later after the direction output. (The characteristics of a motor driver is considered.)

Application example

```
   R0
 ──┤ ├───[ F0  MV,  H 2,  DT 0            ]
         [ F0  MV,  K1000,  DT 1          ]
         [ F0  MV,  K7000,  DT 2          ]
         [ F0  MV,  K300,  DT 3           ]
         [ F1  DMV,  K100000,  DT 4       ]
         [ F0  MV,  K 0,  DT 6            ]
   R1
 ──┤ ├───[ F168  SPD1,  DT 0,  K 0        ]
```

DT 0	H 2
DT 1	K 1000
DT 2	K 7000
DT 3	K 300
DT 4 DT 5	K100000
DT 6	0



Flag conditions

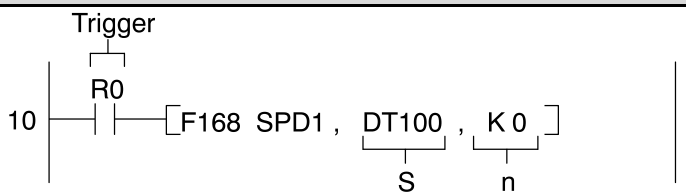
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - n is number except 0 and 1.
  - The value of S exceeds the limit of specified range.
  - S+1 is less than K40
  - S+1 > S+2
  - The value of “S+5, S+4” exceeds the limit of specified range.

Availability
FP0/FP−e

F168 (SPD1) Positioning control (home position return)

Outline Outputs a pulse from the specified output (Y0 or Y1) according to the specified parameter.

Program example

Ladder Diagram	Boolean: Address	Boolean: Instruction
	10 11	ST R 0 F168 (SPD1) DT 100 K 0

S	Starting address for the area that contains the data table.
n	Output Yn that corresponds to the pulse output (n: K0 or K1).

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and the execution condition (trigger) is in the on state, a pulse is output from the specified output (Y0 or Y1).

The control code, initial speed, maximum speed, and acceleration/deceleration time are specified by a user program with a data table as shown on the following page.

The frequency is switched by the acceleration/deceleration time specified for changing from the initial speed to the maximum speed. During deceleration (normally 30 steps), the frequency is changed based on the same slope as during acceleration.

Table of areas used

Channel no.	Control flag	Elapsed value area	Target value area	Directional output	Near home input	Home input
ch0	R903A	DT9044, DT9045 (For FP0 T32, DT90044, DT90045)	DT9046, DT9047 (For FP0 T32, DT90046, DT90047)	Y2	DT9052 bit2 (For FP0 T32, DT90052)	X0
ch1	R903B	DT9048, DT9049 (For FP0 T32, DT90048, DT90049)	DT9050, DT9051 (For FP0 T32, DT90050, DT90051)	Y3	DT9052 (For FP0 T32, DT90052)	X1

Notes

- **When this instruction is used, the setting for the channel corresponding to system register 400 should be set to “High−speed counter not used”.**
- **By performing rewrite during RUN during pulse output, more than the set number of pulses may be output.**

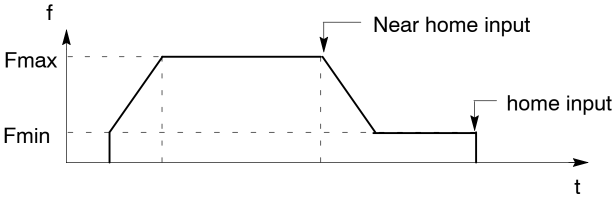
Description of operating mode

Until the home input (X0 or X1) is entered, the pulse is continuously output. To decelerate the movement when near the home, set the bit corresponding to DT9052 to off → on → off → with the near home input. During operation, the elapsed value area and set value area will become insufficient. At the completion of operations, the elapsed value will become 0.

Home position return mode II

Home position return by means of near home input and home input

Deceleration occurs when near home input occurs, and pulse output stops after home input. The control code (lower order) on the next page should be set to H24 to H27.



Home position return mode I

Home position return using only home input

Pulse output stops when home input occurs. Use a control code (lower order) setting on the following page from H20 to H23.

Precautions during programming

When the control code (lower order) is H20 to H23, the home input is enabled after near home input regardless of whether deceleration has ended or is still in progress.

When the control code (lower order) is H24 to H27, the home input is only enabled following near home input after deceleration to the initial speed has been completed.

Even when home input has occurred, executing this instruction causes pulse output to begin.

If the near home input is enabled while acceleration is in progress, deceleration begins.

If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

Run the program referring to page 3 − 487, when controlling the motor in one direction using the pulse output function.

Data table settings

S	Control code	(*1)
S+1	Initial speed Fmin (Hz)	K40 to K5000 (Hz)
S+2	Maximum speed Fmax (Hz)	K40 to K9500 (Hz) (*2)
S+3	Acceleration/deceleration time t (ms)	K30 to K32767 (ms)

☛ next page

(*1): Specify the control code by setting the constant H.

H □ □ □

Pulse width specification
0: Duty 50%
1: Fixed pulse width (approx. 80μs)

Note: A specification of 2 or higher will result in 0.
The pulse width is the output value from the IC and the actual pulse width varies due to the delay in the response of photocoupler.

Operation mode and directional output theory

20: Home position return mode I	No directional output
22: Home position return mode I	directional output off
23: Home position return mode I	directional output on
24: Home position return mode II	No directional output
26: Home position return mode II	output off
27: Home position return mode II	output on

24, 26, and 27 are supported by CPU Ver. 2.0 and subsequent versions.

(*2): When the pulse width is set to duty 50%, the maximum is 6kHz. When the pulse width is set to fixed pulse width (approx. 80μs), the maximum is 9.5kHz.
(Thermocouple input type of FP−e is removed.)

Notes for using FP0 compatiblity mode of FP0R

① The elapsed value and target value of high−speed counter and pulse output is signed 32−bit values.

② The high−speed counter continues counting even if data exceeds the FP0 range (signed 24−bit).

③ The pulse output continues outputting even if data exceeds the FP0 range (signed 24−bit).

④ The waveforms of pulse output are a duty cycle of 25% regardless of the designation of instructions.

⑤ Even if the no count setting is specified with a pulse output instruction, it counts in the addition mode.

⑥ The maximum frequency of pulse output is 10000Hz.

⑦ When using the pulse output instruction, it is not used for the pulse ouput and normal output.

Supplement to the operation in the case with the direction output

1: FP0

1: When specifying a duty of 50%: The pulse output will start approx. “Initial speed period/2” hours later after the direction output.
When the initial speed is 500Hz, it is approx. 1ms.

2: When specifying 80us fixedly: The pulse output will start approx. “Initial speed period − 25us” hours later after the direction output.
When the initial speed is 500Hz, it is approx. 1.98ms.

2: For FP0 compatibility mode

Pulses are output using a duty of 25% fixedly. (The setting is invalid.)
The pulse output will start approx. 300us later after the direction output. (The characteristics of a motor driver is considered.)

Application example

```
 R0
─┤ ├──[ F0 MV, H 22, DT 0          ]
      [ F0 MV, K1000, DT 1         ]
      [ F0 MV, K7000, DT 2         ]
      [ F0 MV, K300, DT 3          ]
 R1
─┤ ├──[ F168 SPD1, DT 0, K 0       ]
```

DT 0	H 22
DT 1	K 1000
DT 2	K 7000
DT 3	K 300

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - n is number except 0 and 1.
  - The value of S exceeds the limit of specified range.
  - S+1 is less than K40
  - S+1 > S+2

Caution regarding pulse output function (F168 and F169)

Use a program such as the following when performing continuous motor rotation in one direction.

```
 X0                                    R0
─┤ ├─(DF  )───────────────────────────[ ]─
 R0
─┤ ├─[ F1  DMV,  K0,        DT9044]        Clear the elapsed value
     [ F0  MV,   H2,        DT0   ]        Incremental control
     [ F0  MV,   K1000,     DT1   ]        Initial speed
     [ F0  MV,   K7000,     DT2   ]        Maximum speed
     [ F0  MV,   K300,      DT3   ]        Acceleration/deceleration time
     [ F1  DMV,  K100000,   DT4   ]        Target value (Movement amount)
     [ F0  MV,   K0,        DT6   ]
     [ F168 SPD1, DT0, K0         ]        Positioning start
```

Pulse output stops when the upper limit of the internal elapse value is exceeded if rotation is in one direction only.
As a countermeasure, reset the elapsed value (zero clear) before executing **F168 (SPD1)** or **F169 (PLS)** instructions, as with the program, above.
The pulse output does not stop when the FP0R is used as the FP0 (FP0 compatibility mode).
The elapsed value is signed 32−bit value.

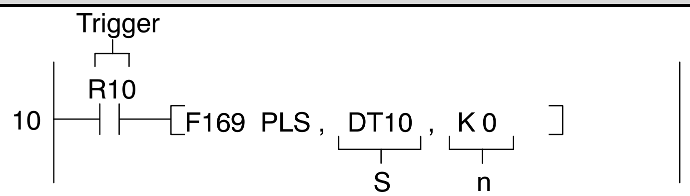
F169 (PLS) Pulse output (with channel specification) (JOG operation)

Availability
FP0/FP−e

Outline Outputs the pulse of the specified parameter from the specified output (Y0 or Y1).

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F169		(PLS)
	DT		10
	K		0

S	Starting address for the area that contains the data table.
n	Output Yn that corresponds to the pulse output (n: K0 or K1).

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and the execution condition (trigger) is in the on state, a pulse is output from the specified channel. The pulse is output while the execution condition (trigger) is in the on state.

By specifying either incremental counting or decremental counting in the control code, this instruction can be used as an instruction for JOG operations. For that situation, set the control code with combinations such as H12 (incremental, directional output off) and H22 (decremental, directional output on).

The frequency and duty can be changed each scan. (This becomes effective with the next pulse output after this instruction is executed.)

See below for the corresponding areas.

Channel no.	Control flag	Data register for elapsed value
ch0	R903A	DT9044, DT9045 (For FP0 T32, DT90044, DT90045)
ch1	R903B	DT9048, DT9049 (For FP0 T32, DT90048, DT90049)

When using the incremental counting mode, the pulse stops when the elapsed value exceeds H7FFFFF.

When using the decremental counting mode, the pulse stops when the elapsed value exceeds HFF800000.

Notes

- **When this instruction is used, the setting for the channel corresponding to system register 400 should be set to “High−speed counter not used”.**
- **By performing a rewrite during RUN while operating, the pulse output will stop during rewriting.**

Data table settings

S	Control code	(*1)
S+1	Frequency (Hz)	K40 to K10000 (Hz) (*2)

(*1): Specify the control code by setting the constant H.

H □ □ □

Pulse width specification
- 0: Fixed pulse width (approx. 80μs) (CPU ver. 2.1 or later)
- 1 to 9: Duty ration approx. 10 to 90% (10% increments)

Operation mode and directional output
- 00: No counting mode
- 10: Incremental counting mode with no directional output
- 12: Incremental counting mode with directional output off
- 13: Incremental counting mode with directional output on
- 20: Decremental counting mode with no directional output
- 22: Decremental counting mode with directional output on
- 23: Decremental counting mode with directional output off

(*2): When the pulse width is set to duty 50%, the maximum is 6kHz. When the pulse width is set to fixed pulse width (approx. 80μs), the maximum is 9.5kHz.
(Thermocouple input type of FP−e is removed.)

Notes for using FP0 compatiblity mode of FP0R

1 The elapsed value and target value of high−speed counter and pulse output is signed 32−bit values.

2 The high−speed counter continues counting even if data exceeds the FP0 range (signed 24−bit).

3 The pulse output continues outputting even if data exceeds the FP0 range (signed 24−bit).

4 The waveforms of pulse output are a duty cycle of 25% regardless of the designation of instructions.

5 Even if the no count setting is specified with a pulse output instruction, it counts in the addition mode.

6 The maximum frequency of pulse output is 10000Hz.

7 When using the pulse output instruction, it is not used for the pulse ouput and normal output.

Supplement to the operation in the case with the direction output

1: FP0

1: When specifying a duty of 50%: The pulse output will start approx. “Initial speed period/2” hours later after the direction output.
When the initial speed is 500Hz, it is approx. 1ms.

2: When specifying 80us fixedly: The pulse output will start approx. “Initial speed period − 25us” hours later after the direction output.
When the initial speed is 500Hz, it is approx. 1.98ms.

2: For FP0 compatibility mode

Pulses are output using a duty of 25% fixedly. (The setting is invalid.)
The pulse output will start approx. 300us later after the direction output. (The characteristics of a motor driver is considered.)

Precautions during programming

If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

Run the program referring to page 3 − 487, when controlling the motor in one direction using the pulse output function.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - n is number except 0 and 1.

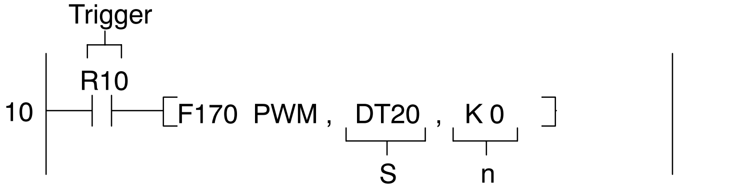
Availability
FP0/FP−e

F170 (PWM) PWM output (with channel specification)

Outline Outputs the PWM of the specified parameter from the specified output (Y0 or Y1).

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	10
11	F170		(PWM)
	DT		20
	K		0

S	Starting address for the area that contains the data table.
n	Output Yn that corresponds to the PWM output (n: K0 or K1).

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and execution condition (trigger) is in the on state, a PWM is output from the specified channel. The PWM is output while the execution condition (trigger) is in the on state.

The frequency and duty are specified with the data table on the right made by a user program.

Since the output is delayed near the maximum and minimum levels, the set duty ratio will differ.

The duty can be changed each scan. The frequency settings is only effective at the start of the execution of the instruction (becomes effective after the next pulse output).

See below for the corresponding areas.

Channel no.	Control flag
ch0	R903A
ch1	R903B

☞ Notes

- **When this instruction is used, the setting for the channel corresponding to system register 400 should be set to “High−speed counter not used”.**
- **By performing a rewrite during RUN while operating, the pulse output will stop during rewriting.**
- **If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.**

Data table settings

S	Control code	H0 to H16 (*1)
S+1	Duty (%)	K1 to K999 (0.1% to 99.9%)

(*1): Control code contents (frequency settings)

	FP0		FP0 compatibility mode of FP0R	
Setting	**Frequency (Hz)**	**Period (ms)**	**Frequency (Hz)**	**Period (ms)**
H11	1000	1.0	1000	1.0
H12	714	1.4	750	1.3
H13	500	2.0	500	2.0
H14	400	2.5	400	2.5
H15	200	5.0	200	5.0
H16	100	10.0	100	10.0
H0	38	26.3	40	25.0
H1	19	52.6	20	50.0
H2	9.5	105.3	10	100.0
H3	4.8	208.3	6	166.7
H4	2.4	416.7	Cannot be specified	
H5	1.2	833.3	Cannot be specified	
H6	0.6	1666.7	Cannot be specified	
H7	0.3	3333.3	Cannot be specified	
H8	0.15	6666.7	Cannot be specified	

H11 to H16 are supported by CPU Ver. 2.0 and subsequent versions.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - n is number except 0 and 1.
  - The frequency setting value set with (S) is outside the specification range.
  - 100% or higher is set with (S + 1)

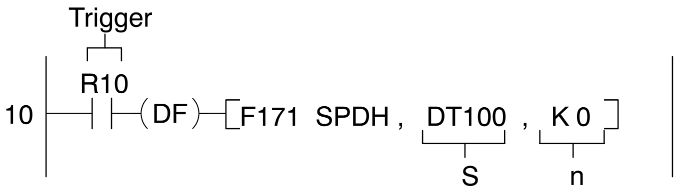
F171 (SPDH) Pulse output (with channel specification) (trapezoidal control)

Availability
FPΣ/FP−X

Outline This instruction outputs pulses from the specified channel for the pulse output according to the specified parameters.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	DF		
12	F171	(SPDH)	
	DT		100
	K		0

S	Starting address of area containing the data table.
n	Channel for pulse output.

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

Pulses are output from the specified channel when the corresponding control flag turns off and the execution condition is in on state.

For FPΣ

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch2	Y3	CW	PLS
	Y4	CCW	SIGN

For FP−X Ry type (AFPX−PLS)

Channel no.	Output	Output method	
ch0 Casette mounting part 1	Y100	CW	PLS
	Y101	CCW	SIGN
ch1 Casette mounting part 2	Y200	CW	PLS
	Y201	CCW	SIGN

For FP−X Tr type

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch1	Y2	CW	PLS
	Y3	CCW	SIGN
ch2	Y4	CW	PLS
	Y5	CCW	SIGN
ch3	Y6	CW	PLS
	Y7	CCW	SIGN

Note) There is no ch3 for C14T and C14TD.
Note) The pulse I/O cassette (AFPX−PLS) cannot be installed on the FP−X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

The control code, initial speed, maximum speed, acceleration/deceleration time, and target value are specified by creating the data table "S" to "S+11" on the following page using the user program.

The frequency is changed using the specified acceleration/deceleration time from the initial speed to the maximum speed. During deceleration, the frequency is changed based on the same slope as during acceleration.

If the frequency is set to 50 kHz or more, specify a duty of 1/4 (25%).

If the frequency for ch2 or ch3 of FP−X Tr type is set to 10kHz or more, specify a duty of 1/4 (25%).

Table of areas used

For FPΣ

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT90044, DT90045	DT90046, DT90047
ch2	R903C	DT90200, DT90201	DT90202, DT90203

For FP−X Ry type

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R911C	DT90348, DT90349	DT90350, DT90351
ch1	R911D	DT90352, DT90353	DT90354, DT90355

Note) Ch1 cannot be used for C14R.

For FP−X Tr type

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R911C	DT90348	DT90350
		DT90349	DT90351
ch1	R911D	DT90352	DT90354
		DT90353	DT90355
ch2	R911E	DT90356	DT90358
		DT90357	DT90359
ch3	R911F	DT90360	DT90362
		DT90361	DT90363

Note) There is no ch3 for C14T and C14TD.
Note) The pulse I/O cassette (AFPX−PLS) cannot be installed on the FP−X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

Operation modes

Incremental <relative value control>

Outputs the pulses set with the target value.

Target value \ Selected mode	CW/CCW	PLS + SIGN Forward off Reverse on	PLS + SIGN Forward on Reverse off	Elapsed value
Positive	Pulse output from CW	Pulse output on direction output off	Pulse output on direction output on	Addition
Negative	Pulse output from CCW	Pulse output on direction output on	Pulse output on direction output off	Subtraction

Absolute <absolute value control>

Outputs a number of pulses equal to the difference between the set target value and the current value.

Target value \ Selected mode	CW/CCW	PLS + SIGN Forward off Reverse on	PLS + SIGN Forward on Reverse off	Elapsed value
Target value greater than current value	Pulse output from CW	Pulse output on direction output off	Pulse output on direction output on	Addition
Target value less than current value	Pulse output from CCW	Pulse output on direction output on	Pulse output on direction output off	Subtraction

Precautions during programming

If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

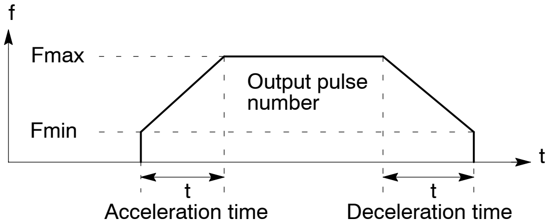
During the time that the circular interpolation control flag R904E is on, the pulse output instructions F166 to F176 cannot be executed.

When using this instruction for FPΣ, set the channels corresponding to system registers 400 and 401 to “Not set as high−speed counter”.

If you perform a rewrite during RUN when pulse output is taking place, more pulses than the setting may be output.

When using this instruction for FP−X, set the pulse output by the system register.

Setting the data table



S S+1	Control code	(*1)
S+2 S+3	Initial speed Fmin (Hz)	(*2)
S+4 S+5	Maximum speed Fmax (Hz)	(*2)
S+6 S+7	Acceleration/deceleration time t (ms)	(*3)
S+8 S+9	Target value (pulse number)	(*4)
S+10 S+11	K0	

(*1): Specification of control code (specify with H constant)

H □ □ □ □ □ □ □ □

0: Fixed

Number of acceleration/deceleration steps
0: 30 steps
1: 60 steps (Can be specified for only FPΣ V1.4 or more and FP−X.)

Duty (on width)
0: Duty 1/2 (50%)
1: Duty 1/4 (25%)

Frequency range
0: 1.5 Hz to 9.8 kHz
1: 48 Hz to 100 kHz
2: 191 Hz to 100 kHz

Operation mode and output method

00: Incremental	CW/CCW	
02: Incremental	PLS + SIGN	(forward off / reverse on)
03: Incremental	PLS + SIGN	(forward on / reverse off)
10: Absolute	CW/CCW	
12: Absolute	PLS + SIGN	(forward off / reverse on)
13: Absolute	PLS + SIGN	(forward on / reverse off)

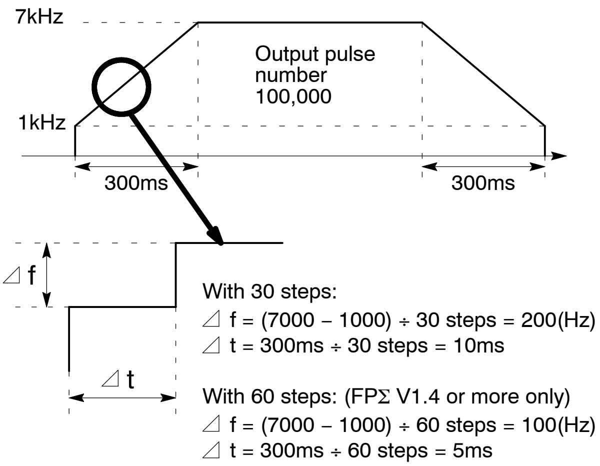
(*2): Frequency (Hz) “K constant”
Frequency range
0: 1.5 Hz to 9.8 kHz [K1 to K9800 (units: Hz)] (Max. error near 9.8 kHz: approx. −0.9 kHz)
  * Set “1” to specify 1.5 Hz.
1: 48 Hz to 100 kHz [K48 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. −3 kHz)
2: 191 Hz to 100 kHz [K191 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. −0.8 kHz)
Initial speed: Set to 30 kHz or lower.

(*3): Acceleration/deceleration time (ms) “K constant”
With 30 steps: K30 to K32760 (Set in units of 30 ms.)
With 60 steps: K60 to K32760 (C32T2 and C28P2 only) (Set in units of 60 ms.)

(*4): Target value
K−2147483648 to K2147483647

Application example

```
 R0
─┤ ├──[ F1  DMV, H1100, DT 0     ]
      [ F1  DMV, K1000, DT 2     ]
      [ F1  DMV, K7000, DT 4     ]
      [ F1  DMV, K300, DT 6      ]
      [ F1  DMV, K100000, DT 8   ]
      [ F1  DMV, K 0, DT 10      ]
 R1
─┤ ├(DF)[ F171 SPDH, DT 0, K 0   ]
```



Acceleration/deceleration time setting

When setting the acceleration/deceleration time, number of steps and initial speed, please use values that satisfy the following formula. When the acceleration/deceleration time has 30 steps please use 30 ms units. When it has 60 steps, please use 60 ms units. *5

Acceleration/deceleration time: t [ms] ≧ (no. of steps x 1000) / initial speed f0 [Hz]

(*5): If they are set without using 30 ms units or 60 ms units, the values will be automatically corrected to the multiple values of 30 ms or 60 ms (larger value).

Flag conditions

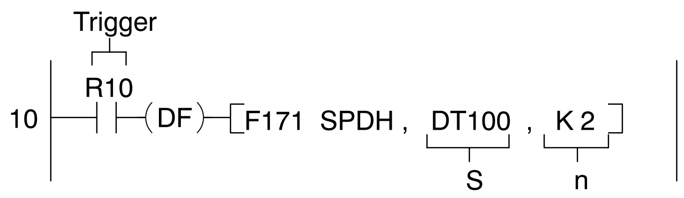
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used.
  - The “n” is outside specification range.
  - The data of “S, S+1” to “S+4, S+5” are outside specification range.
  - The “S+2, S+3” > “S+4, S+5”.
  - The “S+8, S+9” is outside specification range.
  - With the FP−X, the pulse output has not been set by the system register.

F171 (SPDH) Pulse output (with channel specification) (home position return)

Availability
FPΣ/FP–X

Outline This instruction outputs pulses from the specified channel for the pulse output according to the specified parameters.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11 12	ST R 10 DF F171 (SPDH) DT 100 K 2

S	Starting address of area containing the data table.
n	Channel for pulse output.

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

Pulses are output from the specified channel when the corresponding control flag turns off and the execution condition is in on state.

For FPΣ

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
	Y2	Deviation counter clear	
ch2	Y3	CW	PLS
	Y4	CCW	SIGN
	Y5	Deviation counter clear	

For FP–X Ry type (AFPX–PLS)

Channel no.	Output	Output method	
ch0 Casette mounting part 1	Y100	CW	PLS
	Y101	CCW	SIGN
	Y102	Deviation counter clear	
ch1 Casette mounting part 2	Y200	CW	PLS
	Y201	CCW	SIGN
	Y202	Deviation counter clear	

For FP−X Tr type

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
	Y4 or Y8	Deviation counter clear	
ch1	Y2	CW	PLS
	Y3	CCW	SIGN
	Y5 or Y9	Deviation counter clear	
ch2	Y4	CW	PLS
	Y5	CCW	SIGN
	No deviation counter clear control		
ch3	Y6	CW	PLS
	Y7	CCW	SIGN
	No deviation counter clear control		

Note) There is no ch3 for C14T and C14TD.
Note) C14T and C14TD is Y4 or Y5. C30T, C30TD, C60T and C60TD is Y8 or Y9.
Note) The deviation counter clear control is not available for the ch2 and ch3.
Note) Use the ch2 and ch3 at up to 20 kHz.

The control code, initial speed, maximum speed, acceleration/deceleration time, and deviation counter clear signal are specified by creating a data table as described on the following page using the user program.

The frequency is changed using the specified acceleration/deceleration time from the initial speed to the maximum speed. During deceleration, the frequency is changed based on the same slope as during acceleration.

If the frequency is set to 50 kHz or more, specify a duty of 1/4 (25%).

If the frequency for ch2 or ch3 of FP−X Tr type is set to 10kHz or more, specify a duty of 1/4 (25%).

Table of areas used

For FPΣ

Channel no.	Control flag	Elapsed value area	Target value area	Near home	Home input
ch0	R903A	DT90044, DT90045	DT90046, DT90047	DT90052 bit2	X2
ch2	R903C	DT90200, DT90201	DT90202, DT90203	DT90052 bit4	X5

For FP−X Ry type

Channel no.	Control flag	Elapsed value area	Target value area	Near home	Home input
ch0	R911C	DT90348, DT90349	DT90350, DT90351	DT90052 bit4	X4
ch1	R911D	DT90352, DT90353	DT90354, DT90355	DT90052 bit4	X5

For FP−X Tr type

Channel no.	Control flag	Elapsed value area	Target value area	Near home	home input
ch0	R911C	DT90348	DT90350	DT90052 <bit4>	X4
		DT90349	DT90351		
ch1	R911D	DT90352	DT90354		X5
		DT90353	DT90355		
ch2	R911E	DT90356	DT90358		X6
		DT90357	DT90359		
ch3	R911F	DT90360	DT90362		X7
		DT90361	DT90363		

Note) There is no ch3 for C14T and C14TD.

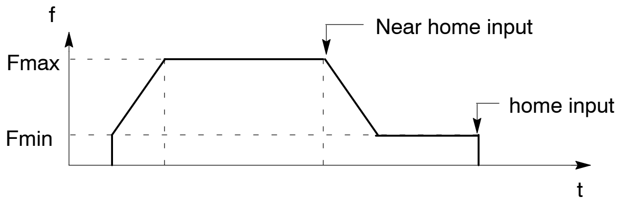
Operation modes

Return to home position

Pulses are output continually until home input (X2 or X5) occurs. To decelerate at near home, set the corresponding bit of special data register DT90052 off → on → off when near home input occurs.
The value in the elapsed value area during a home position return differs from the current value. When the return is completed, the elapsed value changes to 0.

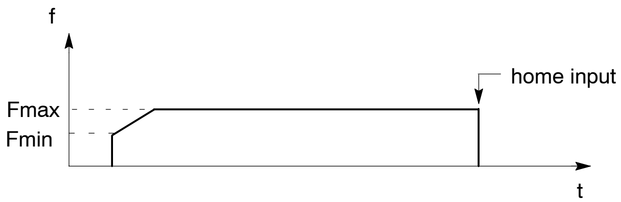
Home position return by means of near home input and home input

Deceleration occurs when near home input occurs, and pulse output stops after home input. Operation varies depending on the control code (lower order) settings described on the following page.



Home position return using only home input

Pulse output stops when home input occurs. Use a control code (lower order) setting on the following page from H20 to H27.



Setting the data table

S S+1	Control code	(*1)
S+2 S+3	Initial speed Fmin (Hz)	(*2)
S+4 S+5	Maximum speed Fmax (Hz)	(*2)
S+6 S+7	Acceleration/deceleration time t (ms)	(*3)
S+8 S+9	Deviation counter clear signal output time tr(ms)	(*4)

(*1): Control code specification (specify with an H constant)

H □ □ □ □ □ □ □ □

0: Fixed

Number of acceleration/deceleration steps
0: 30 steps
1: 60 steps (Can be specified for only FPΣ V1.4 or more and FP−X.)

Duty (on width)
0: Duty 1/2 (50%)
1: Duty 1/4 (25%)

Frequency range
0: 1.5 Hz to 9.8 kHz
1: 48 Hz to 100 kHz
2: 191 Hz to 100 kHz

Operation mode and output method

20: Home position return mode I	CW
21: Home position return mode I	CCW
22: Home position return mode I	Direction output off
23: Home position return mode I	Direction output on
24: Home position return mode I	CW + deviation counter clear
25: Home position return mode I	CCW + deviation counter clear
26: Home position return mode I	Direction output off + deviation counter clear
27: Home position return mode I	Direction output on + deviation counter clear
30: Home position return mode II	CW
31: Home position return mode II	CCW
32: Home position return mode II	Direction output off
33: Home position return mode II	Direction output on
34: Home position return mode II	CW + deviation counter clear
35: Home position return mode II	CCW + deviation counter clear
36: Home position return mode II	Direction output off + deviation counter clear
37: Home position return mode II	Direction output on + deviation counter clear

(*2): Frequency (Hz) “K constant”
Frequency range
0: 1.5 Hz to 9.8 kHz [K1 to K9800 (units: Hz)] (Max. error near 9.8 kHz: approx. −0.9 kHz)
* Set “1” to specify 1.5 Hz.
1: 48 Hz to 100 kHz [K48 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. −3 kHz)
For this range we recommend a duty of 1/4.
2: 191 Hz to 100 kHz [K191 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. −0.8 kHz)
For this range we recommend a duty of 1/4.
Initial speed: Set to 30 kHz or lower.

(*3): Acceleration/deceleration time (ms) “K constant”
With 30 steps: K30 to K32760
With 60 steps: K60 to K32760 (FPΣ V1.4 or more and FP−X only)

(*4): Deviation counter clear signal output time
Set the deviation counter clear signal output time.
0.5 ms to 100 ms [K0 to K100] Set value and margin of error (0.5 ms or less)
Specify K0 when not using this signal or when specifying 0.5 ms

Application example

```
  R0
──┤ ├────[ F1  DMV,  H1125,  DT 0       ]
         [ F1  DMV,  K1000,  DT 2       ]
         [ F1  DMV,  K7000,  DT 4       ]
         [ F1  DMV,  K100,  DT 6        ]
         [ F1  DMV,  K10,  DT 8         ]
  R1
──┤ ├(DF)─[ F171  SPDH,  DT 0,  K 2     ]
```

Acceleration/deceleration time setting

When setting the acceleration/deceleration time, number of steps and initial speed, please use values that satisfy the following formula. When the acceleration/deceleration time has 30 steps please use 30 ms units. When it has 60 steps, please use 60 ms units. *5

Acceleration/deceleration time: t [ms] ≧ (no. of steps x 1000) / initial speed f0 [Hz]

(*5): If they are set without using 30 ms units or 60 ms units, the values will be automatically corrected to the multiple values of 30 ms or 60 ms (larger value).

Precautions during programming

When the control code (lower order) is H20 to H27 (Home return mode type I), the home input is enabled after near home input regardless of whether deceleration has ended or is still in progress.

When the control code (lower order) is H30 to H37 (Home return mode type II), the home input is only enabled following near home input after deceleration to the initial speed has been completed.

Even when home input has occurred, executing this instruction causes pulse output to begin.

If the near home input is enabled while acceleration is in progress, deceleration begins.

If both the normal program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

When using this instruction for FPΣ, set the channels corresponding to system registers 400 and 401 to “Not set as high−speed counter”.

When using this instruction for FP−X, set the pulse output by the system register.

If you perform a rewrite during RUN when pulse output is taking place, more pulses than the setting may be output.

During the time that the circular interpolation control flag R904E is on, the pulse output instructions F166 to F176 cannot be executed.

Please refer to ”F0 (MV) instruction pulse output control” when doing a soft reset, count disable, pulse output stop, or near home process.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used.
  - The “n” is outside specification range.
  - The data of “S, S+1” to “S+4, S+5” are outside specification range.
  - The “S+2, S+3” > “S+4, S+5”.
  - With the FP−X, the pulse output has not been set by the system register.

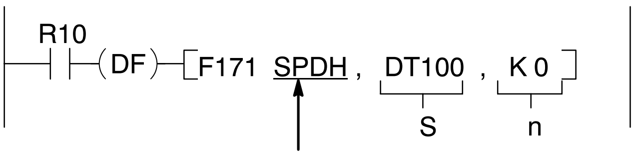
F171 (SPDH) Pulse output (Trapezoidal control)

Availability
FP0R

Outline Outputs pulses from the specified pulse output channels according to the specified parameters.
[Feature] An acceleration time and deceleration time can be set respectively. Also, the deceleration stop is available.
The target speed can be changed.

Program example

Ladder Diagram



To change the speed, do not insert the DF instruction as the trigger should be kept being ON.

Boolean

Address	Instruction		
10	ST	R	10
11	DF		
12	F171		(SPDH)
	DT		100
	K		0

S	Starting 16–bit area for registering data tables
n	Channels intended for pulse output (n=0 to 3)

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	A
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

When the trigger is on, pulses are output from the specified channels and the trapezoidal control can be performed.

The control code, initial speed, target speed, acceleration time, deceleration time and the target value is specified by creating data tables [S] to [S+11] using user programs.

When accelerating, the frequency is changed in the acceleration time specified from the initial speed to the target speed.

When decelerating, the frequency is changed in the deceleration time specified from the target speed.

The deceleration stop request is available by the control data (bit5) of DT90052. (e.g.) F0(MV) H120, DT90052

When using the same condition as the table used at the previous startup, the operation can be started at high speed without calculation.

Method of acceleration/deceleration and initial speed
During the pulse output, the pulse output instruction flag corresponding to the channel turns on.

When the deceleration stop is requested during acceleration, deceleration is performed with the same slope of the deceleration time from the target speed.

In this instruction, the operation is processed giving the acceleration/deceleration time priority.

The pulse output frequency can be changed by rewriting the target speed during the pulse output.

Two control methods are available, which are type 0 and type 1. Using the type 0, the speed can be changed within the range of the target speed specified first.
Using the type 1, the speed can be accelerated/decelerated up to the range of the maximum frequency, regardless of the target speed specified fist.

Image of operation 1: When the target speed is not changed

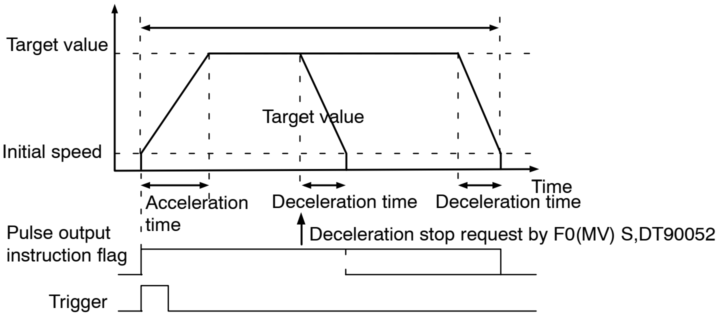
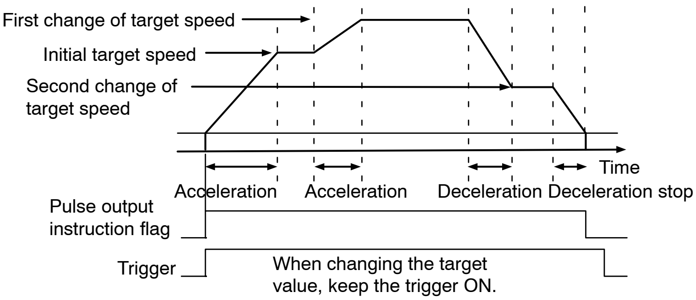


Image of operation 2: When the target value is changed



Precautions during programming

When the same channel is described in a normal program and interrupt program both, do not execute them at the same time.

This instruction cannot be executed when the corresponding control flag to the channel started is on.

When rewriting during RUN is performed, the pulse output will stop.

The instruction cannot be started when the deceleration stop request flag is on.

To restart after stopping the operation, turn off the trigger once, and then turn it on again.

When the instruction is started during the interrupt program, specify the execution in the interrupt program with the control code. Speed cannot be changed when the instruction is executed in the interrupt program.

Pules output channels and areas used

Channel No.	Output	Output type		Pulse output instruction flag	Elapsed value area	Target value area	Correction speed of initial speed	Deceleration minimum speed	Acceleration forbidden area starting position
ch0	Y0	CW	PLS	R9120	DT90400 DT90401	DT90402 DT90403	R90406	DT90407	DT90408 DT90409
	Y1	CCW	SIGN						
ch1	Y2	CW	PLS	R9121	DT90410 DT90411	DT90412 DT90413	R90416	DT90417	DT90418 DT90419
	Y3	CCW	SIGN						
ch2	Y4	CW	PLS	R9122	DT90420 DT90421	DT90422 DT90423	R90426	DT90427	DT90428 DT90429
	Y5	CCW	SIGN						
ch3	Y6	CW	PLS	R9123	DT90430 DT90431	DT90432 DT90433	R90436	DT90437	DT90438 DT90439
	Y7	CCW	SIGN						

Settings of data table

Table of type 0

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Velocity range (Frequency) (Hz) 1Hz to 50kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time up to the target speed: Acceleration time range (ms) K1 to K32760 (Unit: ms)
S+8 S+9	Deceleration time (ms)	Deceleration time from the target speed: Deceleration time range (ms) K1 to K32760 (Unit: ms)
S+10 S+11	Target value (No. of pulses)	Target value range K−2,147,483,648 to K2,147,483,647

Table of type 1

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Velocity range (Frequency) (Hz) 1Hz to 50kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time up to the max. speed 50kHz: Acceleration time range (ms) K1 to K32760 (Unit: ms)
S+8 S+9	Deceleration time (ms)	Deceleration time from the max. speed 50kHz: Deceleration time range (ms) K1 to K32760 (Unit: ms)
S+10 S+11	Target value (No. of pulses)	Target value range K−2,147,483,648 to K2,147,483,647

Note: If the speed is changed to a value over 50kHz during the operation, it will be corrected to 50kHz.

Note the following characteristics according to the specified initial speed.

(1) When the initial speed is 1 or higher, and lower than 46Hz, the control up to the maximum frequency to the degree of 10kHz can be performed. If the frequency is higher than that, the speed error will be larger.

(2) When the initial speed is 46 or higher, and lower than 184Hz, the control up to 50kHz can be performed.

(3) When the initial speed is 184 or higher, the control up to 50kHz can be performed. The speed error around 50kHz will be smallest.

Change of speed during pulse output

(1)With the type 0, if a value larger than the target speed at start−up is specified, it will be corrected to the target speed at start−up. With the type 1, if the target value is set to a value larger than 50kHz, it will be corrected to 50kHz.

(2)If the elapsed value crosses over the acceleration forbidden area starting position during accelerating, acceleration cannot be performed. For information on the acceleration forbidden area starting position, refer to the special registers.

(3)For deceleration, the speed cannot be lower than the deceleration minimum speed.
For information on the deceleration minimum speed, refer to the special registers.

Assignment of control code (Specify with H constant)

H □ □ □ □ □ □ □ □

10: Fixed

Control assignment
0: Trapezoidal

Control assignment type
0: Type 0 1: Type 1

Interrupt execution assignment
0: Execute in main program.
1: Execute in interrupt program.
(The trigger is the level type.)

Output assignment
0: Pulse output
1: Calculation only

Operation mode assignment
0: Incremental
1: Absolute

Output type assignment
0: CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

*As for the output assignment
When starting the instruction with the setting of “1: Calculation only”, the pulse output is not performed.
When starting the instruction with the assignment of the same channel and the same parameter after executing this instruction once for a channel, it can be started at high speed. (It is the same for the both cases of “Pulse output” and “Calculation only”.)
However, if a parameter other than the parameter used for the previous execution is specified, the high−speed startup cannot be performed.

Note) The same parameter means that all the parameters other than the output assignment are the same.

Output type

Incremental <Relative control>

The pulses specified for the target value are output.

Mode selection / Target value	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When positive	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When negative	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

Absolute <Absolute value control>

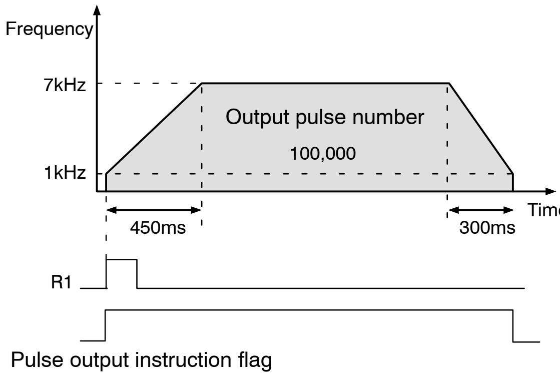
The pulse that is the difference between the specified target value and the current value is output.

Mode selection / Target value	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When target value is larger than current value	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When target value is smaller than current value	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

[Explanation of pulse output operation]
Pulses are output using a duty of 25% fixedly.
When using the PLS +SIGN method, pulses will be output approx. 300 us later after the output of direction signal. (The characteristics of a motor driver is considered.)

Sample program1: Trapezoidal control type 0, No deceleration stop request, No change of speed

```
 R0
─┤ ├──( DF )──[ F1  DMV, H10000000, DT0  ]
              [ F1  DMV, K1000, DT2      ]
              [ F1  DMV, K7000, DT4      ]
              [ F1  DMV, K450, DT6       ]
              [ F1  DMV, K300, DT8       ]
              [ F1  DMV, K100000, DT10   ]
 R1
─┤ ├──( DF )──[ F171  SPDH,  DT0,  K0    ]
```

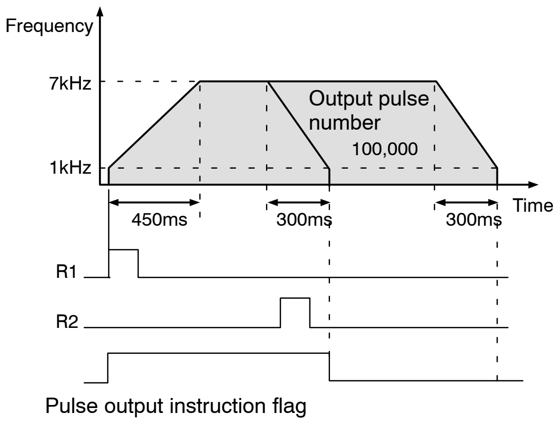


Data table

DT0	Control information	Trapezoidal control Incremental CW/CCW
DT2	Initial speed (Hz)	1000Hz
DT4	Target speed (Hz)	7000Hz
DT6	Acceleration time (ms)	450ms
DT8	Deceleration time (ms)	300ms
DT10	Target value (No. of pulses)	100,000 pulses

Sample program2: Trapezoidal control type 0, With deceleration stop request, With change of speed

```
 R0
─┤ ├──( DF )──[ F1 DMV, H10000000, DT0  ]
              [ F1 DMV, K1000, DT2      ]
              [ F1 DMV, K7000, DT4      ]
              [ F1 DMV, K450, DT6       ]
              [ F1 DMV, K300, DT8       ]
              [ F1 DMV, K100000, DT10   ]
 R1
─┤ ├──( DF )──[ F171  SPDH,  DT0,  K0   ]
 R2
─┤ ├──( DF )──[ F0  MV  H120   DT90052  ]
              [ F0  MV  H100,  DT90052  ]
```



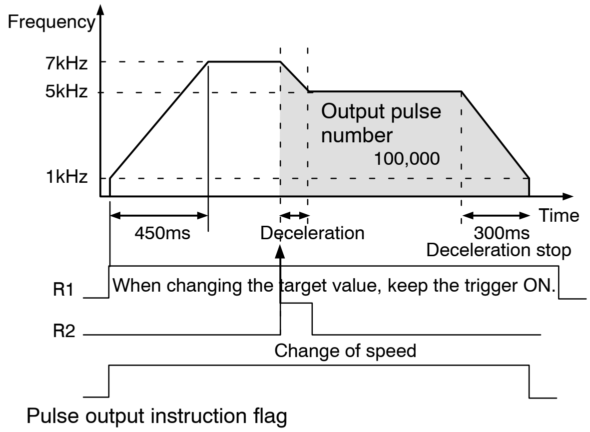
The deceleration stop is performed according to the deceleration time after the detection of deceleration stop request.

Data table

Refer to the Sample program1.

Sample program3: Trapezoidal control type 0, with change of speed

```
R0
─┤ ├──( DF )──[ F1 DMV, H10000000, DT0 ]
              [ F1 DMV, K1000, DT2 ]
              [ F1 DMV, K7000, DT4 ]
              [ F1 DMV, K450, DT8 ]
              [ F1 DMV, K300, DT6 ]
              [ F1 DMV, K100000, DT10 ]
R1
─┤ ├──────────[ F171 SPDH, DT0, K0 ]
R2
─┤ ├──( DF )──[ F1 DMV, K5000, DT4 ]
```

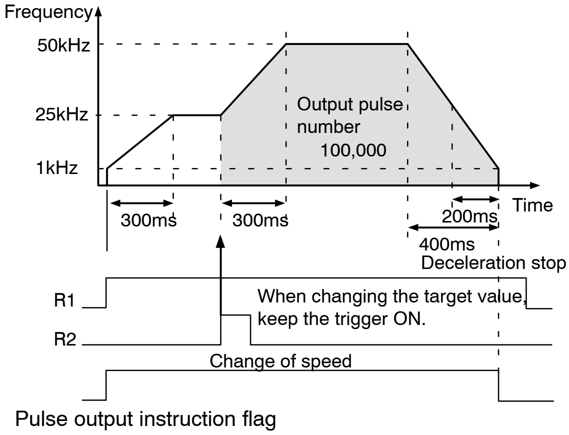


Data table

DT0	Control information	Trapezoidal control Incremental CW/CCW
DT2	Initial speed (Hz)	1000Hz
DT4	Target speed (Hz)	7000Hz
DT6	Acceleration time (ms)	450ms
DT8	Deceleration time (ms)	300ms
DT10	Target value (No. of pulses)	100,000 pulses

Sample program4: Trapezoidal control type 1, with change of speed

```
R0
─┤├──( DF )──[ F1 DMV, H10010000, DT0 ]
             [ F1 DMV, K1000, DT2 ]
             [ F1 DMV, K25000, DT4 ]
             [ F1 DMV, K600, DT8 ]
             [ F1 DMV, K400, DT6 ]
             [ F1 DMV, K100000, DT10 ]
R1
─┤├──────────[ F171 SPDH, DT0, K0 ]
R2
─┤├──( DF )──[ F1 DMV, K20000, DT4 ]
```



Data table

DT0	Control information	Trapezoidal control Incremental CW/CCW	
DT2	Initial speed (Hz)	1000Hz	
DT4	Target speed (Hz)	25000Hz	
DT6	Acceleration time (ms)	600ms	Acceleration time up to 50kHz
DT8	Deceleration time (ms)	400ms	Deceleration time from 50kHz
DT10	Target value (No. of pulses)	100,000 pulses	

Flag conditions

- Error flag (R9007):
- Error flag (R9008):

  – Turns on when the area specified using the index modifier exceeds the limit.
  – Turns on when n is out of the specified range.
  – Turns on when each data of [S,S+1] to [S+4,S+5] is out of the specified range.
  – Turns on when [S+2,S+3]>[S+4,S+5].
  – Turns on when [S10,S+11] is out of the specified range.
  – Turns on when the pulse output has not been set by the system register.
  – Turns on when the interrupt execution has been specified for executing the instruction in the main program.

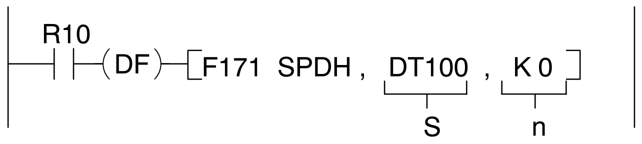
F171 (SPDH) Pulse output (JOG positioning type 0)

Availability
FP0R

Outline Outputs the specified number of pulses and performs the deceleration stops after the position control starting input during the pulse output.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	10
11	DF	
12	F171	(SPDH)
	DT	100
	K	0

S	Starting 16−bit area for registering data tables
n	Channels intended for pulse output (n=0 to 3)

Operands

Operand	Relay			Timer/ Counter	Register	Index register	SWR	Constant		Index modifier
	WX	WY	WR	EV	DT	In (*1)		K	H	
S	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

(*1) I0 to ID

Description

When the trigger is on, pulses are output from the specified channels.

The control code, initial speed, target speed, acceleration time, deceleration time and the target value after the position control starting input is specified by creating data tables [S] to [S+11] using user programs.

When accelerating, the frequency is changed in the acceleration time specified from the initial speed to the target speed.

The pulse output continues until the position control starting input turns on after reaching the target speed.

After the position control starting input turned on, the pulse output continues up to the target value, and then decelerates and stops.

For using the position control staring input (X0, X1, X2, X3), set the system register 402.

When decelerating, the frequency is changed in the deceleration time specified from the target speed.

The positioning control can be also started by the control data (bit6) of DT90052.
(e.g.) F0(MV) H140, DT90052

The deceleration stop can be requested by the control data (bit5) of DT90052.
(e.g.) F0(MV) H120, DT90052

When using the same condition as the table used at the previous startup, the operation can be started at high speed without calculation.

During the pulse output, the pulse output instruction flag corresponding to channel turns on.

When the deceleration stop is requested during acceleration, deceleration is performed with the same slope of the deceleration time from the target speed.

In this instruction, the operation is processed giving the acceleration/deceleration time priority.

The initial speed may be corrected to enable accelerating/decelerating within the specified time.

Image of operation: When the target speed does not change

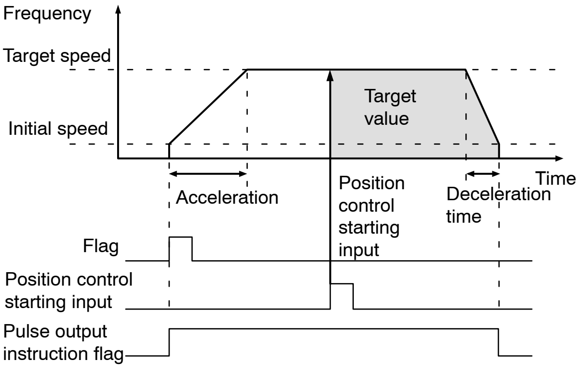
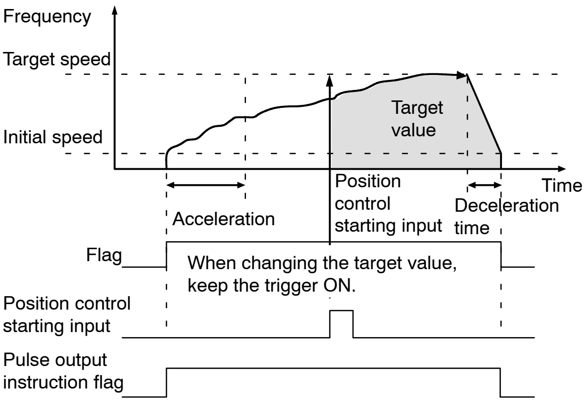


Image of operation: When the target speed changes



Precautions during programming

When the same channel is described in a normal program and interrupt program both, do not execute them at the same time.

This instruction cannot be executed when the corresponding pulse output instruction flag to the channel started is on.

When rewriting during RUN is performed, the pulse output will stop.

As for the position control starting input, only the rising edge (ON) is detected.

The instruction cannot be started when the deceleration stop request flag is on.

Note that the methods to stop the pulse output in this instruction are only any of the following operations: Turning on the position control starting input (position control starting flag), requesting the deceleration stop and executing an emergency stop.

To restart after stopping the operation, turn off the trigger once, and then turn it on again.

When the instruction is started during the interrupt program, specify the execution in the interrupt program with the control code. Speed cannot be changed when the instruction is executed in the interrupt program.

Pulse output channels and areas used

Channel No.	Output	Output type		Position control starting input	Pulse output instruction flag	Elapsed value area	Target value area	Correction speed of initial speed	Deceleration minimum speed	Acceleration forbidden area starting position
ch0	Y0	CW	PLS	X0 DT90052 bit6	R9120	DT90400 DT90401	DT90402 DT90403	DT90406	DT90407	DT90408 DT90409
	Y1	CCW	SIGN							
ch1	Y2	CW	PLS	X1 DT90052 bit6	R9121	DT90410 DT90411	DT90412 DT90413	DT90416	DT90417	DT90418 DT90419
	Y3	CCW	SIGN							
ch2	Y4	CW	PLS	X2 DT90052 bit6	R9122	DT90420 DT90421	DT90422 DT90423	DT90426	DT90427	DT90428 DT90429
	Y5	CCW	SIGN							
ch3	Y6	CW	PLS	X3 DT90052 bit6	R9123	DT90430 DT90431	DT90432 DT90433	DT90436	DT90437	DT90438 DT90439
	Y7	CCW	SIGN							

Setting the data table

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Velocity range (Frequency) (Hz) 1Hz to 50kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time up to the max. speed 50kHz. Acceleration time range (ms) K1 to K32760 (Unit: ms)
S+8 S+9	Deceleration time (ms)	Deceleration time from the max. speed 50kHz. Deceleration time range (ms) K1 to K32760 (Unit: ms)
S+10 S+11	Target value (No. of pulses)	Target value range K−2,147,483,648 to K2,147,483,647

Note the following characteristics according to the specified initial speed.

(1) When the initial speed is 1 or higher, and lower than 46Hz, the control up to the maximum frequency to the degree of 10kHz can be performed.
If the frequency is higher than that, the speed error will be larger.

(2) When the initial speed is 46 or higher, and lower than 184Hz, the control up to 50kHz can be performed.

(3) When the initial speed is 184 or higher, the control up to 50kHz can be performed.
The speed error around 50kHz will be smallest.

Change of speed during pulse output

(1) If the target value is set to a value larger than 50kHz, it will be corrected to 50kHz.

(2) If the elapsed value crosses over the acceleration forbidden area starting position during accelerating, acceleration cannot be performed.
For information on the acceleration forbidden area starting position, refer to the special registers.

(3) For deceleration, the speed cannot be lower than the deceleration minimum speed.
For information on the deceleration minimum speed, refer to the special registers.

Assignment of control code (Specify with H constant)

H □ □ □ □ □ □ □ □

10: Fixed

Control assignment
1: JOG positioning

Control assignment 2
0: Type 0

Interrupt execution assignment
0: Execute in main program.
1: Execute in interrupt program.
(The trigger is the level type.)

Output assignment
0: Pulse output
1: Calculation only

Operation mode assignment
0: Incremental

Output type assignment
0: CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

When the target value has been set to 0, it will stop when the position control starting input turns on.
(Only V1.06 or later)
For reversing the output when the target value has been set to 0, set the output type of control code to 4, 5, 6 instead of 0, 1, 2.

*As for the output assignment

When starting the instruction with the setting of “1: Calculation only”, the pulse output is not performed.
When starting the instruction with the assignment of the same channel and the same parameter after executing this instruction once for a channel, it can be started at high speed. (It is the same for the both cases of “Pulse output” and “Calculation only”.)
However, if a parameter other than the parameter used for the previous execution is specified, the high−speed startup cannot be performed.

Note) The same parameter means that all the parameters other than the output assignment are the same.

Output type

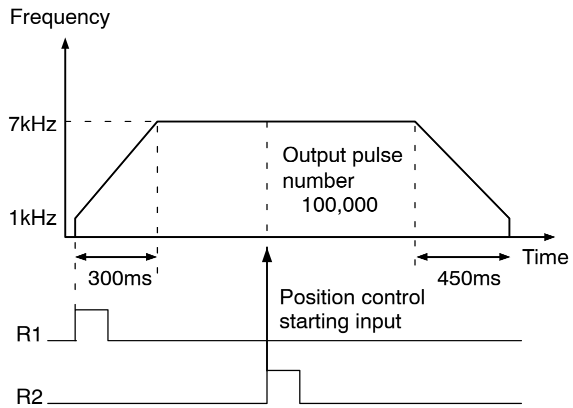
Mode selection / Target value	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When positive	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When negative	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

When the target value has been set to 0, the output will be the forward mode when the output type is set to 0, 1, 2. For performing the reverse output, set the type to 4, 5, 6 instead of 0, 1, 2. (V1.06 or later)

[Explanation of pulse output operation]
Pulses are output using a duty of 25% fixedly.
When using the PLS +SIGN method, pulses will be output approx. 300 us later after the output of direction signal.
(The characteristics of a motor driver is considered.)

Sample program

```
R0
─┤├──( DF )──[ F1  DMV, H10100000, DT0  ]
             [ F1  DMV, K1000, DT2      ]
             [ F1  DMV, K7000, DT4      ]
             [ F1  DMV, K300, DT6       ]
             [ F1  DMV, K450, DT8       ]
             [ F1  DMV, H100000, DT10   ]
R1
─┤├──( DF )──[ F171  SPDH,  DT0,  K0    ]
R2
─┤├──( DF )──[ F0  MV  H140,  DT90052   ]
             [ F0  MV  H100,  DT90052   ]
```



Flag conditions

- Error flag (R9007):
- Error flag (R9008):
  - Turns on when the area specified using the index modifier exceeds the limit.
  - Turns on when n is out of the specified range.
  - Turns on when each data of [S,S+1] to [S+4,S+5] is out of the specified range.
  - Turns on when [S+2,S+3]>[S+4,S+5].
  - Turns on when [S+10,S+11] is out of the specified range.
  - Turns on when the pulse output has not been set by the system register.
  - Turns on when the interrupt execution has been specified for executing the instruction in the main program.

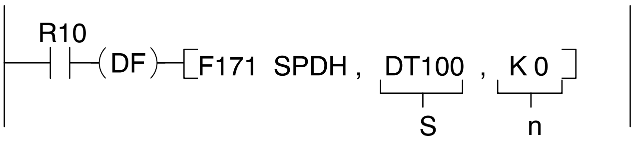
F171 (SPDH) Pulse output (JOG positioning type 1)

Availability
FP0R

Outline Outputs the specified number of pulses changing the target speed again and performs the deceleration stop after the position control starting input during the pulse output.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	DF		
12	F171		(SPDH)
	DT		100
	K		0

S	Starting 16−bit area for registering data tables
n	Channels intended for pulse output (n=0 to 3)

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	A
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and the trigger is on, pulses are output from the specified channels.

The control code, initial speed, target speed 1, acceleration time, target speed 2 after position control starting input, change time, deceleration time and the target value is specified by creating data tables [S] to [S+15] using user programs.

When accelerating, the frequency is changed in the acceleration time specified from the initial speed to the target speed.

After the position control starting input turned on, the pulse output continues up to the target value, and then decelerates and stops.

For using the position control staring input (X0, X1, X2, X3), set the system register 402.

When decelerating, the frequency is changed in the deceleration time specified from the target speed.

The positioning control can be also started by the control data (bit6) of DT90052.
(e.g.) F0(MV) H140, DT90052

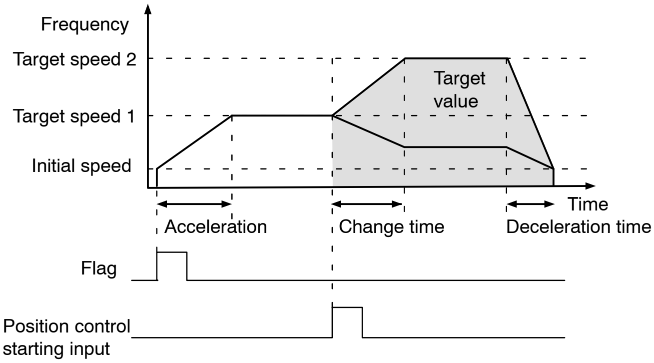
The deceleration stop can be requested by the control data (bit5) of DT90052.
(e.g.) F0(MV) H120, DT90052

When using the same condition as the table used at the previous startup, the operation can be started at high speed without calculation.

Method of acceleration/deceleration and initial speed
When the deceleration stop is requested during acceleration, deceleration is performed with the same slope of the deceleration time from the target speed.

In this instruction, the operation is processed giving the acceleration/deceleration time priority.

The initial speed may be corrected to enable accelerating/decelerating within the specified time.



Note) Note that the position control starting input will be disregarded even if it is turned on during acceleration.

Precautions during programming

When the same channel is described in a normal program and interrupt program both, do not execute them at the same time.

This instruction cannot be executed when the corresponding control flag to the channel started is on.

If rewriting during RUN is performed during pulse output, pulses more than the setting may be output.

As for the position control starting input, only the rising edge (ON) is detected.

The instruction cannot be started when the deceleration stop request flag is on.

Note that the methods to stop the pulse output in this instruction are only any of the following operations: Turning on the position control starting input (position control starting flag), requesting the deceleration stop and executing an emergency stop.

The target speed cannot be changed with this instruction.

When the instruction is started during the interrupt program, specify the execution in the interrupt program with the control code.

Pules output channels and areas used

Channel No.	Output	Output type		Position control starting input	Pulse output instruction flag	Elapsed value area	Target value area
ch0	Y0	CW	PLS	X0 DT90052 bit6	R9120	DT90400 DT90401	DT90402 DT90403
	Y1	CCW	SIGN				
ch1	Y2	CW	PLS	X1 DT90052 bit6	R9121	DT90410 DT90411	DT90412 DT90413
	Y3	CCW	SIGN				
ch2	Y4	CW	PLS	X2 DT90052 bit6	R9122	DT90420 DT90421	DT90422 DT90423
	Y5	CCW	SIGN				
ch3	Y6	CW	PLS	X3 DT90052 bit6	R9123	DT90430 DT90431	DT90432 DT90433
	Y7	CCW	SIGN				

Setting the data table

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Velocity range (Frequency) (Hz) 1Hz to 50kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed 1 (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration/deceleration time range (ms) K1 to K32760 (Unit: ms)
S+8 S+9	Target speed 2 (Hz)	Velocity range (Frequency) (Hz) 1Hz to 50kHz [K1 to K50000 (Unit: Hz)]
S+10 S+11	Change time (ms)	K1 to K32760 (Unit: ms)
S+12 S+13	Deceleration time (ms)	K1 to K32760 (Unit: ms)
S+14 S+15	Target value (No. of pulses)	Target value range K−2,147,483,648 to K2,147,483,647

Note the following characteristics according to the specified initial speed.

(1) When the initial speed is 1 or higher, and lower than 46Hz, the control up to the maximum frequency to the degree of 10kHz can be performed.
If the frequency is higher than that, the speed error will be larger.

(2) When the initial speed is 46 or higher, and lower than 184Hz, the control up to 50kHz can be performed.

(3) When the initial speed is 184 or higher, the control up to 50kHz can be performed.
The speed error around 50kHz will be smallest.

Assignment of control code (Specify with H constant)

H □ □ □ □ □ □ □ □

- 10: Fixed
- Control assignment
  1: JOG positioning
- Control assignment 2
  1: Type 1
- Interrupt execution assignment
  0: Execute in main program.
  1: Execute in interrupt program.
  (The trigger is the level type.)
- Output assignment
  0: Pulse output
  1: Calculation only
- Operation mode assignment
  0: Incremental
- Output type assignment
  0: CW/CCW
  1: PLS+SIGN (Forward OFF/Reverse ON)
  2: PLS+SIGN (Forward ON/Reverse OFF)

*As for the output assignment

When starting the instruction with the setting of “1: Calculation only”, the pulse output is not performed.
When starting the instruction with the assignment of the same channel and the same parameter after executing this instruction once for a channel, it can be started at high speed. (It is the same for the both cases of “Pulse output” and “Calculation only”.)
However, if a parameter other than the parameter used for the previous execution is specified, the high−speed startup cannot be performed.

Note) The same parameter means that all the parameters other than the output assignment are the same.

Output type

Incremental <Relative control>

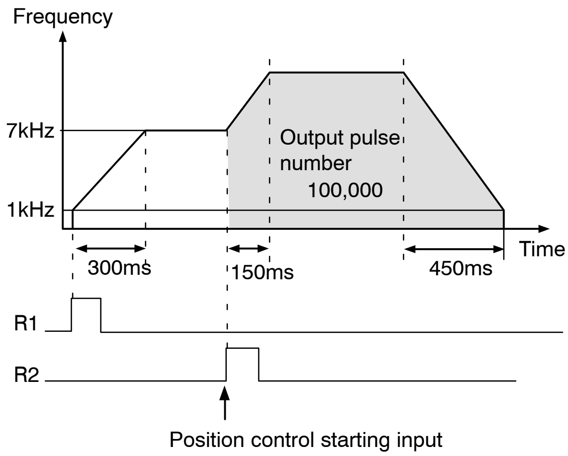
The pulse specified for the target value are output.

Target value \ Mode selection	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When positive	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When negative	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

[Explanation of pulse output operation]
Pulses are output using a duty of 25% fixedly.
When using the PLS +SIGN method, pulses will be output approx. 300 us later after the output of direction signal.
(The characteristics of a motor driver is considered.)

Sample program

```
R0
─┤ ├──( DF )──[ F1 DMV, H10110000, DT0   ]
              [ F1 DMV, K1000, DT2       ]
              [ F1 DMV, K7000, DT4       ]
              [ F1 DMV, K300, DT6        ]
              [ F1 DMV, K10000, DT8      ]
              [ F1 DMV, K150, DT10       ]
              [ F1 DMV, K450, DT12       ]
              [ F1 DMV, K100000, DT14    ]
R1
─┤ ├──( DF )──[ F171  SPDH,  DT0,  K0    ]
R2
─┤ ├──( DF )──[ F0  MV H140,  DT90052    ]
              [ F0  MV H140,  DT90052    ]
```



Flag conditions

- Error flag (R9007):
- Error flag (R9008):
  - Turns on when the area specified using the index modifier exceeds the limit.
  - Turns on when n is out of the specified range.
  - Turns on when each data of [S,S+1] to [S+4,S+5] is out of the specified range.
  - Turns on when [S+8,S+9] is out of the specified range.
  - Turns on when [S+2,S+3]>[S+4,S+5].
  - Turns on when [S+2,S+3]>[S+8,S+9].
  - Turns on when [S+14,S+15] is out of the specified range.
  - Turns on when the pulse output has not been set by the system register.

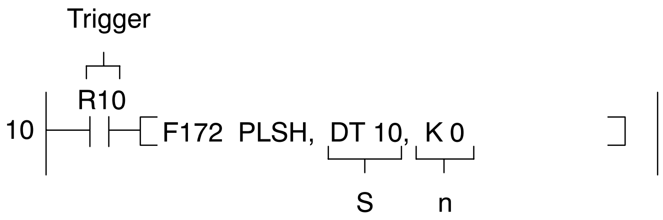
F172 (PLSH) Pulse output (with channel specification) (JOG operation)

Availability
FPΣ/FP−X

Outline Outputs the pulses of the specified parameter from the specified channel for the pulse output.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F172	(PLSH)	
	DT		10
	K		0

S	Starting number for the area that contains the data table
n	Channel that corresponds to the pulse output

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and the execution condition is in the on state, pulses are output from the specified channel. The pulses are output while the execution condition is on.

For FPΣ

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch2	Y3	CW	PLS
	Y4	CCW	SIGN

For FP−X Ry type (AFPX−PLS)

Channel no.	Output	Output method	
ch0 Cassete mounting part 1	Y100	CW	PLS
	Y101	CCW	SIGN
ch1 Cassete mounting part 2	Y200	CW	PLS
	Y201	CCW	SIGN

For FP−X Tr type

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch1	Y2	CW	PLS
	Y3	CCW	SIGN
ch2	Y4	CW	PLS
	Y5	CCW	SIGN
ch3	Y6	CW	PLS
	Y7	CCW	SIGN

Note) There is no ch3 for C14T and C14TD.
Note) The pulse I/O cassette (AFPX−PLS) cannot be installed on the FP−X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

By specifying either addition counting or subtraction counting in the control code, this instruction can be used as an instruction for JOG operations.

Frequency can be changed in each scan, and the target value can be changed asynchronously. However, the control code cannot be changed during instruction execution.

If a frequency of 50 kHz or higher is specified, a duty of 1/4 (25%) should be specified.

If the frequency for ch2 or ch3 of FP−X Tr type is set to 10kHz or more, specify a duty of 1/4 (25%).

Table of areas used

For FPΣ

Channel no.	Control flag	Elapsed value
ch0	R903A	DT90044, DT90045
ch2	R903C	DT90200, DT90201

For FP−X Ry type

Channel no.	Control flag	Elapsed value
ch0	R911C	DT90348, DT90349
ch1	R911D	DT90352, DT90353

Note) Ch1 cannot be used for C14R.

For FP−X Tr type

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R911C	DT90348	DT90350
		DT90349	DT90351
ch1	R911D	DT90352	DT90354
		DT90353	DT90355
ch2	R911E	DT90356	DT90358
		DT90357	DT90359
ch3	R911F	DT90360	DT90362
		DT90361	DT90363

Note) There is no ch3 for C14T and C14TD.
Note) The pulse I/O cassette (AFPX−PLS) cannot be installed on the FP−X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

Precautions during programming

During the time that the circular interpolation control flag R904E is on, the pulse output instructions F166 to F176 cannot be executed.

When using this instruction for FPΣ, the setting for the channels corresponding to system registers no. 400 and no. 401 should be set to “High−speed counter not used”.

When using this instruction for FP−X, set the pulse output by the system register.

If a rewrite is executed during RUN while the system is operating, pulse output stops while the program is being rewritten.

If the same notation is being used for both the ordinary program and the interrupt program, make sure they are not both executed at the same time.

Target value setting can be used in FPΣ V1.4 or more and FP−X only.

If a value outside of the specified range is written for the frequency area while the instruction is being executed, the frequency output will be adjusted to either to the minimum or the maximum. And when starting execution of the instruction, an operation error occurs.

If the control code is changed after the instruction startup, it will be invalid.
If the frequency is changed to a value outside the specification range after the instruction startup, an operation error will not occur and the program will run at the minimum or maximum value in the specification range.

Flag conditions

- Error flag (R9007): Turns on when:
- Error flag (R9008): Turns on when:
  - The control code or frequency is outside the specification range. (During instruction startup)
  - The specified area is exceeded when an index is modified.
  - The “n” is outside specification range.
  - With the FP−X, the pulse output has not been set by the system register.

Data table settings

Mode with no target value

S S+1	Control code	(*1)
S+2 S+3	Frequency	(*2)

Target value match stop mode

S S+1	Control code	(*1)
S+2 S+3	Frequency	(*2)
S+4 S+5	Target value	(*3)

(*1): Control code specification (specify with an H constant)

H□□□□□□□□

0: Fixed

Target value setting
0: Mode with no target value
1: Target value match stop mode
(Can be specified for only FPΣ V1.4 or more and FP–X.)

Duty (on width)
0: Duty 1/2 (50%)
1: Duty 1/4 (25%)

Frequency range
0: 1.5 Hz to 9.8 kHz
1: 48 Hz to 100 kHz
2: 191 Hz to 100 kHz

Output method

00: No counting	CW
01: No counting	CCW
10: Addition counting	CW
12: Addition counting	Directional output off
13: Addition counting	Directional output on
21: Subtraction counting	CCW
22: Subtraction counting	Directional output off
23: Subtraction counting	Directional output on

(*2): Frequency (Hz) "K constant"
Frequency range
0: 1.5 Hz to 9.8 kHz [K1 to K9800 (units: Hz)] (Max. error near 9.8 kHz: approx. –0.9 kHz)
* Set "1" to specify 1.5 Hz.
1: 48 Hz to 100 kHz [K48 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. –3 kHz)
2: 191 Hz to 100 kHz [K191 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. –0.8 kHz)
For counting method, set the initial instruction execution frequency to 30 kHz or lower.

(*3): Target value (Absolute value) (FPΣ V1.4 or more and FP–X only)
This is used when setting the target value match stop mode.(Absolute only)

Designate the target value setting in the range indicated below. If an out of range value is designated, the number of pulses output will be different than the designated value. The target value setting is ignored in the no count mode.

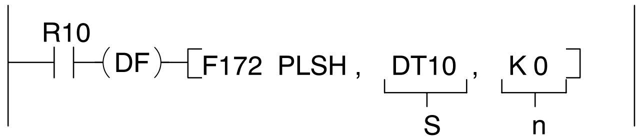
Output method	Range of target values which can be designated
Addition counting	Designate a value larger than the current value
Subtraction counting	Designate a value smaller than the current value

F172(PLSH) Double word compare: Start equalPulse output (JOG operation type 0 and 1)

Availability
FP0R

Outline Performs the pulse output from the specified pulse output channels according to the specified parameters.
[Feature] Acceleration time and deceleration time can be set individually. The deceleration stop is also available.
The target speed can be changed.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 10 F172 (PLSH) DT 10 K 0

S	Starting 16−bit area for registering data tables
n	Channels intended for pulse output (n=0 to 3)

Operands

Operand	Relay				Timer/Counter		Register	Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	In (*1)			K	H	
n	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A	A
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

When the corresponding pulse output instruction flag is off and the trigger is on, pulses are output from the specified channels.

With the JOG operation type , the control code, initial speed, target speed, acceleration time and deceleration time is specified by creating data tables [S] to [S+9] using user programs.

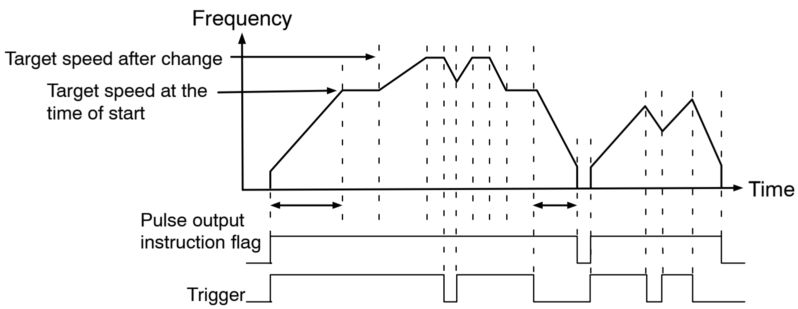
With the JOG operation type 1 (with target values), the target value is specified in a range of [S+10] to [S+11] as well as the above items.

When accelerating, the frequency is changed in the acceleration time specified from the initial speed to the target speed.

When the trigger is turned off after starting the instruction, the deceleration stop is performed.

When decelerating, the frequency is changed from the target speed in the specified deceleration time.

When the trigger is turned on during deceleration, acceleration is performed again from deceleration.



Precautions during programming

When the same channel is described in a normal program and interrupt program both, do not execute them at the same time.

This instruction cannot be executed when the corresponding pulse output instruction flag to the channel started is on.

If rewriting during RUN is performed during the pulse output, the pulse output stops during the program is being rewritten.

It is not effective if the control code is changed after starting the instruction. It has no effect on the operation.

Pules output channels and areas used

Channel No.	Output	Output type		Pulse output instruction flag	Elapsed value area	Target value area	Correction speed of initial speed	Deceleration minimum speed	Acceleration forbidden area starting position
ch0	Y0	CW	PLS	R9120	DT90400	DT90402	DT90406	DT90407	DT90408
	Y1	CCW	SIGN		DT90401	DT90403			DT90409
ch1	Y2	CW	PLS	R9121	DT90410	DT90412	DT90416	DT90417	DT90418
	Y3	CCW	SIGN		DT90411	DT90413			DT90419
ch2	Y4	CW	PLS	R9122	DT90420	DT90422	DT90426	DT90427	DT90428
	Y5	CCW	SIGN		DT90421	DT90423			DT90429
ch3	Y6	CW	PLS	R9123	DT90430	DT90432	DT90436	DT90437	DT90438
	Y7	CCW	SIGN		DT90431	DT90433			DT90439

Setting the data table

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Velocity range (Frequency) (Hz) 1Hz to 50kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed 1 (Hz)	Acceleration time up to the max. speed 50kHz. Acceleration time range (ms) K1 to K32760 (Unit: ms)
S+6 S+7	Acceleration time (ms)	Deceleration time from the max. speed 50kHz. Deceleration time range (ms) K1 to K32760 (Unit: ms)
S+8 S+9	Deceleration time (ms)	Target value range Note: Available for JOG type 1 (with target values) only. K−2,147,483,648 to K2,147,483,647
	Target value	Available for JOG type 1 only

Note the following characteristics according to the specified initial speed.

(1) When the initial speed is 1 or higher, and lower than 46Hz, the control up to the maximum frequency to the degree of 10kHz can be performed.
If the frequency is higher than that, the speed error will be larger.

(2) When the initial speed is 46 or higher, and lower than 184Hz, the control up to 50kHz can be performed.

(3) When the initial speed is 184 or higher, the control up to 50kHz can be performed.
The speed error around 50kHz will be smallest.

Change of speed during pulse output

(1) With the type 0, if a value larger than the target speed at start−up is specified, it will be corrected to the target speed at start−up.
With the type 1, if the target value is set to a value larger than 50kHz, it will be corrected to 50kHz.

(2) If the elapsed value crosses over the acceleration forbidden area starting position during accelerating, acceleration cannot be performed.
For information on the acceleration forbidden area starting position, refer to the special registers.

(3) For deceleration, the speed cannot be lower than the deceleration minimum speed.
For information on the deceleration minimum speed, refer to the special registers.

Assignment of control code (Specify with H constant)

H □ □ □ □ □ □ □ □

10: Fixed

Control assignment
0: JOG

Control assignment 2
0: Type 0 (Without target values)
1: Type 1 (With target values)

0: Fixed

Output assignment
0: Pulse output
1: Calculation only

Movement direction
Type 0 (Without target values)
0: Forward
1: Reverse
Type 1 (With target values)
0: Incremental
1: Absolute

Output type assignment
0: CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

*As for the output assignment

When starting the instruction with the setting of “1: Calculation only”, the pulse output is not performed.
When starting the instruction with the assignment of the same channel and the same parameter after executing this instruction once for a channel, it can be started at high speed. (It is the same for the both cases of “Pulse output” and “Calculation only”.)
However, if a parameter other than the parameter used for the previous execution is specified, the high−speed startup cannot be performed.

Note) The same parameter means that all the parameters other than the output assignment are the same.

Output type 0 (without target values)

Mode selection / Operation mode	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
Forward	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
Reverse	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

Output type 1 (with target values)

Incremental <Relative control>

The pulses specified for the target value are output.

Mode selection / Target value	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When positive	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When negative	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

Absolute <Absolute value control>

The pulse that is the difference between the specified target value and the current value is output.

Mode selection / Target value	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When target value is larger than current value	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When target value is smaller than current value	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

[Explanation of pulse output operation]
Pulses are output using a duty of 25% fixedly.
When using the PLS +SIGN method, pulses will be output approx. 300 us later after the output of direction signal. (The characteristics of a motor driver is considered.)

Method of acceleration/deceleration and initial speed

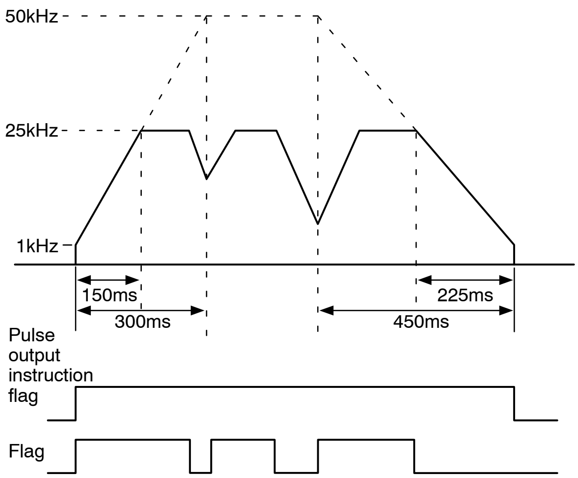
When the deceleration stop is requested during acceleration, deceleration is performed with the same slope of the deceleration time from the target speed.

In this instruction, the operation is processed giving the acceleration/deceleration time priority.

The initial speed may be corrected to enable accelerating/decelerating within the specified time.

Sample program

```
R0
--| |-----------[ F1 DMV, H10000000, DT0 ]
                [ F1 DMV, K1000, DT2     ]
                [ F1 DMV, K25000, DT4    ]
                [ F1 DMV, K300, DT6      ]
                [ F1 DMV, K450, DT8      ]
R1
--| |-----------[ F172 PLSH, DT0, K0     ]
```



Flag conditions

- Error flag (R9007):
- Error flag (R9008):
  - – Turns on when the control code or frequency is out of the settable range (when the instruction is started).
  - – Turns on when the area specified using the index modifier exceeds the limit.
  - – Turns on when n is out of the specified range.
  - – Turns on when the pulse output of the specified channel is not set by the system register.

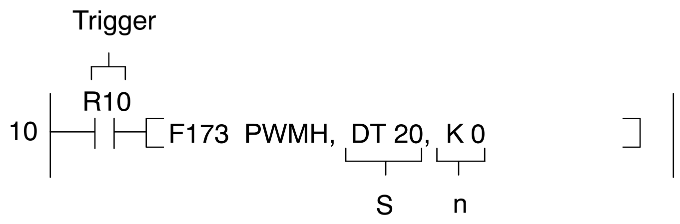
F173 (PWMH) PWM output (with channel specification)

Availability
FPΣ/FP−X/FP0R

Outline Outputs the PWM of the specified parameter from the specified channel for the PWM output.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 10
11	F173 (PWMH)
	DT 20
	K 0

S	Starting number for the area that contains the data table
n	Channel targeted by the PWM output

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and the execution condition is in the on state, a PWM is output from the specified channel for the PWM output. The PWM is output while the execution condition is on.

The data table shown at below, indicating the frequency and duty, is created and the values are specified by the user program.

The duty, particularly when it is close to the minimum or maximum value, may be off from the specified ratio, depending on the load voltage and load current.

The duty can be changed for each separate scan. Control codes, however, cannot be changed while an instruction is being executed.

Table of areas used

For FPΣ

Channel no.	Output	Control flag
ch0	Y0	R903A
ch2	Y3	R903C

For FP−X Ry type (AFPX−PLS)

Channel no.	Output	Control flag
ch0 Casette mounting part 1	Y100	R911C
ch1 Casette mounting part 2	Y200	R911D

For FP−X Tr

Channel no.	Output	Control flag
ch0	Y0	R911C
ch1	Y2	R911D
ch2	Y4	R911E
ch3	Y6	R911F

Note) There is no ch3 for FPX−C14T.
Note) The pulse I/O cassette (AFPX−PLS) cannot be installed on the FP−X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

For FP0R

Channel no.	Output	Pulse I/O instruction flag
ch0	Y0	R9120
ch1	Y2	R9121
ch2	Y4	R9122
ch3	Y6	R9123

Precautions during programming

During the time that the circular interpolation control flag R904E is on, the pulse output instructions F166 to F176 cannot be executed.

When using this instruction for FPΣ, the setting for the channels corresponding to system registers no. 400 and no. 401 should be set to “High−speed counter not used”.

When using this instruction for FP−X, set the PWM output by the system register.

If a rewrite is executed during RUN while the system is operating, pulse output stops while the program is being rewritten.

If the same notation is being used for both the ordinary program and the interrupt program, make sure they are not both executed at the same time.

If a value over the specified range is written for the duty area while the instruction is being executed, the duty output will be adjusted to the maximum. And when starting execution of the instruction, an operation error occurs.

If the control code is changed after the instruction startup, it does not affect the frequency but the resolution of the duty.

If the frequency is changed to a value outside the specification range after the instruction startup, an operation error will not occur and the program will run with the duty of 100 resolution.

If the duty is changed to 100% or higher after the instruction startup, an operation error will not occur and the program will run at the maximum value of the specified resolution.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified area is exceeded when an index is modified.
  - The n is any value other than 0 or 2.
  - The control code is outside specification range. (During instruction startup)
  - The duty is higher than 100%. (During instruction startup)
  - With the FP−X, the PWM output has not been set by the system register.

Data table settings

S	Control code	(*1)
S+1	Duty	(*2)

(*1): Control code specification (specify using K constant)

For FPΣ and FP−X

Resolution of 1000

K	Frequency (Hz)	Timing (ms)
K0	1.5	666.67
K1	2.0	502.51
K2	4.1	245.70
K3	6.1	163.93
K4	8.1	122.85
K5	9.8	102.35
K6	19.5	51.20
K7	48.8	20.48
K8	97.7	10.24
K9	201.6	4.96
K10	403.2	2.48
K11	500.0	2.00
K12	694.4	1.44
K13	1.0 k	0.96
K14	1.3 k	0.80
K15	1.6 k	0.64
K16	2.1 k	0.48
K17	3.1 k	0.32
K18	6.3 k	0.16
K19	12.5 k	0.08

Resolution of 100

K	Frequency (Hz)	Timing (ms)
K20	15.6 k	0.06
K21	20.8 k	0.05
K22	25.0 k	0.04
K23	31.3 k	0.03
K24	41.7 k	0.02

Note: When using ch2 or ch3 on FP−X Tr type, use the control codes up to K20.

(*2): Specification of duty (specify using K constant)
If the control code is K0 to K19, the duty is K0 to K999 (0.0% to 99.9%).
If the control code is K20 to K24, the duty is K0 to K99 (0% to 99%).
Set values are specified in units of 1% (K10) (digits below the decimal point are rounded off).

For FP0R

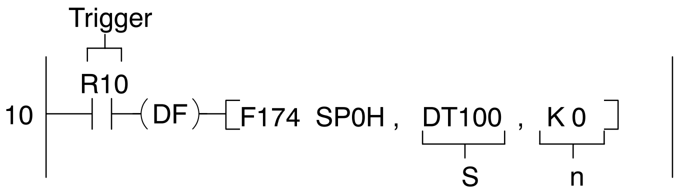
K	Frequency (Hz)	Period (ms)
K3	6	166.67
K4	7.5	133.33
K5	12.5	80.00
K6	25	40.00
K7	50	20.00
K8	100	10.00
K9	200	5.00
K10	400	2.50
K11	600	1.67
K12	800	1.25
K13	1000	1.00
K14	1200	0.83
K15	1600	0.63
K16	2000	0.50
K17	3000	0.33
K18	4800	0.21
Other than the above	Cannot be specified	

F174(SP0H) Pulse output (with channel specification) (Selectable data table control operation)

Availability
FPΣ/FP−X

Outline Outputs the pulses from the specified channel for the pulse output according to the specified data table.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11 12	ST R 10 DF F174 (SP0H) DT 100 K 0

S	Starting address of area containing the data table.
n	Channel for pulse output.

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding control flag is off and the execution condition is in the on state, pulses are output from the specified channel (ch0 or ch2) based on the contents set for the data table in which the first address is that specified by “S”.

For FPΣ

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch2	Y3	CW	PLS
	Y4	CCW	SIGN

For FP−X Ry type (AFPX−PLS)

Channel no.	Output	Output method	
ch0 Cassete mounting part 1	Y100	CW	PLS
	Y101	CCW	SIGN
ch1 Cassete mounting part 2	Y200	CW	PLS
	Y201	CCW	SIGN

For FP–X Tr type

Channel no.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch1	Y2	CW	PLS
	Y3	CCW	SIGN
ch2	Y4	CW	PLS
	Y5	CCW	SIGN
ch3	Y6	CW	PLS
	Y7	CCW	SIGN

Note) There is no ch3 for C14T and C14TD.
Note) The pulse I/O cassette (AFPX–PLS) cannot be installed on the FP–X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

When the elapsed value of the high–speed counter reaches the target value specified in the data table, the pulse frequency is switched (interrupt processing is carried out).

When the elapsed value agrees with the last target value, the pulse output operation finishes.

Use the **F0 (MV)** instruction to control the high–speed counter to force the pulse output control to stop.

If the frequency is set to 50 kHz or more, specify a duty of 1/4 (25%).

If the frequency for ch2 or ch3 of FP–X Tr type is set to 10kHz or more, specify a duty of 1/4 (25%).

Table of areas used

For FPΣ

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT90044, DT90045	DT90046, DT90047
ch2	R903C	DT90200, DT90201	DT90202, DT90203

For FP–X Ry type

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R911C	DT90348, DT90349	DT90350, DT90351
ch1	R911D	DT90352, DT90353	DT90354, DT90355

For FP–X Tr type

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R911C	DT90348	DT90350
		DT90349	DT90351
ch1	R911D	DT90352	DT90354
		DT90353	DT90355
ch2	R911E	DT90356	DT90358
		DT90357	DT90359
ch3	R911F	DT90360	DT90362
		DT90361	DT90363

Note) There is no ch3 for C14T and C14TD.
Note) The pulse I/O cassette (AFPX–PLS) cannot be installed on the FP–X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

Precautions during programming

The high−speed counter control flag R903A (R903C) is on from the time that the execution condition for the F174 (SP0H) instruction has gone on until the pulse output stops.

During the time that the high−speed counter control flag R903A (R903C) is on, the high−speed counter and pulse output instructions F166 to F176, which use the same control flag, cannot be executed.

During the time that the circular interpolation control flag R904E is on, the pulse output instructions F166 to F176 cannot be executed.

An operation error occurs if a value that is not within the allowable range is specified for the control code or for frequency 1. (If the data for frequency 1 is 0, the operation is terminated without anything being executed.)

Pulse output is stopped if the frequency of the second or a subsequent stage is specified as 0 or as a value outside the allowable range.

If the table pointer exceeds the data register DT area during pulse output, pulse output control stops and the high−speed counter control flag R903A (R903C) goes off.

Always make sure that the target values are specified within the ranges indicated on the following page. If a value outside the allowable range is specified, the number of pulses output will be different from the specified value.

If a periodic interrupt or high−speed counter value interrupt program is run, or the PLC link function is used at the same time, a frequency of 80 kHz or less should be used.

Note: With FP−X, refer to the table of areas used for the internal relay equivalent to R903A (R903C).

When using this instruction for FP−X, set the pulse output by the system register.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used.
  - The “n” is other than 0 or 2.
  - The control code or frequency 1 is outside setting range.
  - With the FP−X, the pulse output has not been set by the system register.

Setting the data tabler

Address	Contents	Note
[S]	Control code	(*1)
[S+2]	Frequency 1	(*2)
[S+4]	Target value 1 (Number of pulses)	(*3)
[S+6]	Frequency 2	
[S+8]	Target value 2 (Number of pulses)	
	≈	
[S+2n]	Frequency n	
[S+2(n+1)]	Target value n (Number of pulses)	
[S+2(n+2)]	K0	End of table (Pulse output stops.)

(*1): Specification of control code “H constant”

H □□□ □□□ □□

Upper word
0: Fixed

Duty (on width)
0: Duty 1/2 (50%)
1: Duty 1/4 (25%)

Frequency range
0: 1.5 Hz to 9.8 kHz
1: 48 Hz to 100 kHz
2: 191 Hz to 100 kHz

Operation mode
0: Incremental Specifies the amount of travel (number of pulses)
1: Absolute Specifies the target value (absolute value)

Output method
0: CW (addition counting)
1: CCW (subtraction counting)
2: PLS+SIGN (forward off) (addition counting)
3: PLS+SIGN (reverse on) (subtraction counting)
4: PLS+SIGN (forward on) (addition counting)
5: PLS+SIGN (reverse off) (subtraction counting)

(*2): Frequency (Hz) “K constant”
Frequency range
0: 1.5 Hz to 9.8 kHz [K1 to K9800 (units: Hz)] (Max. error near 9.8 kHz: approx. −0.9 kHz)
 * Set “1” to specify 1.5 Hz.
1: 48 Hz to 100 kHz [K48 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. −3 kHz)
2: 191 Hz to 100 kHz [K191 to K100000 (units: Hz)] (Max. error near 100 kHz: approx. −0.8 kHz)
Initial speed: Set ”Frequency 1” to 30 kHz or less.

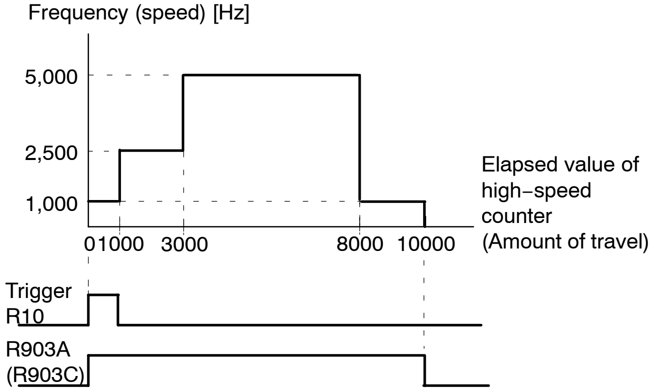
(*3): Target value (K−2147483648 to K2147483647)
The value of the 32−bit data specified for the target value should be within the range indicated in the table below.

Specification of control code		Range of allowable target values
Operation mode	**Output method**	
Incremental	**Addition counting**	Specifies a positive value.
	Subtraction counting	Specifies a negative value.
Absolute	**Addition counting**	Specifies a value larger than the current value
	Subtraction counting	Specifies a value smaller than the current value

Program example

[Operation content]

1. Pulse output from the specified channel ch0 begins at 1,000 Hz when the **F174 (SP0H)** instruction execution condition (trigger) R10 goes on.
2. At the point when 1,000 pulses have been counted at a frequency of 1,000 Hz, the frequency switches to 2,500 Hz.
3. At the point when 3,000 pulses have been counted at a frequency of 2,500 Hz, the frequency switches to 5,000 Hz.
4. At the point when 8,000 pulses have been counted at a frequency of 5,000 Hz, the frequency switches to 1,000 Hz.
5. At the point when 10,000 pulses have been counted, pulse output stops.



When the execution condition (trigger) R10 of the **F174 (SP0H)** instruction goes on, the high−speed counter control flag R903A (R903C) goes on. When the elapsed value reaches 10,000 and pulse output stops, R903A (R903C) goes off.

Note: With FP−X, refer to the table of areas used for the internal relay equivalent to R903A (R903C).

[Settings and program]

The frequency range is from 191 Hz to 100 kHz, the duty 1/4 (25%), the operation mode is Incremental, and the output method is CW.

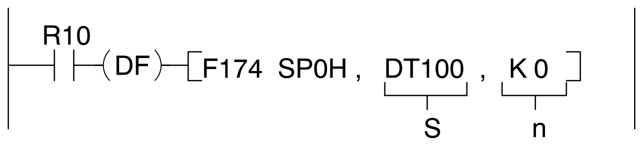
```
 R0
─┤ ├─[F1 DMV , H 1200, DT100]     Control code: "H1200"
      [F1 DMV , K 1000, DT102]     Frequency 1: 1,000Hz
      [F1 DMV , K 1000, DT104]     Target value 1: 1,000 pulses
      [F1 DMV , K 2500, DT106]     Frequency 2: 2,500Hz
      [F1 DMV , K 2000, DT108]     Target value 2: 2,000 pulses
      [F1 DMV , K 5000, DT110]     Frequency 3: 5,000Hz
      [F1 DMV , K 5000, DT112]     Target value 3: 5,000 pulses
      [F1 DMV , K 1000, DT114]     Frequency 4: 1,000Hz
      [F1 DMV , K 2000, DT116]     Target value 4: 2,000 pulses
      [F1 DMV , K    0, DT118]     Output pulse stops
 R10
─┤ ├─(DF)─[F174 SP0H,DT100,K0]     Pulse output control
```

F174(SP0H) Pulse output (Arbitrary data table control operation)

Availability
FP0R

Outline Outputs pulses from the specified pulse output channels according to the specified data table.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11 12	ST R 10 DF F174 (SP0H) DT 100 K 0

S	Starting 16−bit area for registering data tables
n	Channels intended for pulse output (n=0 to 3)

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

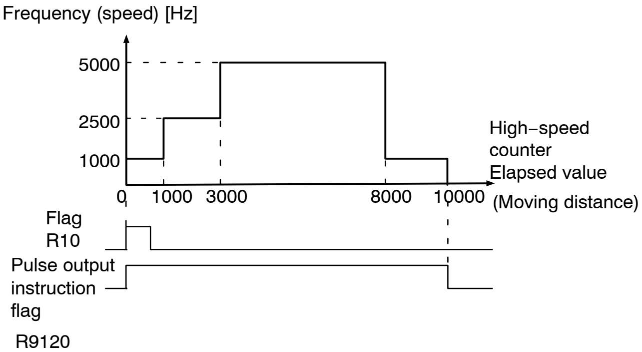
Description

When the corresponding pulse output instruction flag is off and the trigger is on, pulses are output from the specified channels according to the conditions specified in the data table which starts with the address specified by [S].

If the elapsed value of the pulse output reaches the target value specified in the data table, the pulse frequency is changed (by the interrupt operation).

When the elapsed value reaches the final target value, the pulse output stops.

Image of operation



Pules output channels and areas used

Channel No.	Output	Output type		Pulse output instruction flag	Elapsed value area	Target value area
ch0	Y0	CW	PLS	R9120	DT90400 DT90401	DT90402 DT90403
	Y1	CCW	SIGN			
ch1	Y2	CW	PLS	R9121	DT90410 DT90411	DT90412 DT90413
	Y3	CCW	SIGN			
ch2	Y4	CW	PLS	R9122	DT90420 DT90421	DT90422 DT90423
	Y5	CCW	SIGN			
ch3	Y6	CW	PLS	R9123	DT90430 DT90431	DT90432 DT90433
	Y7	CCW	SIGN			

Setting the data table

S S+1	Control code *1
S+2 S+3	Frequency 1 (Hz) *2
S+4 S+5	Target value 1 (No. of pulses) *3
S+2 S+3	Frequency 2 (Hz)
S+4 S+5	Target value 2 (No. of pulses)
≈	≈
S+2n S+2n+1	Frequency n (Hz)
S+2(n+1) S+2(n+1)+1	Target value n (No. of pulses)
S+2(n+2) S+2(n+2)+1	K0: fixed: Table end

Note: If the frequency “n” is set to a value larger than 50kHz, it will be corrected to 50kHz.

Note the following characteristics applied according to the value of frequency 1.

(1) When the frequency 1 is 1 or higher, and lower than 46Hz, the control up to the maximum frequency to the degree of 10kHz can be performed.
If the frequency “n” is set to a value below 6Hz, it will be corrected to 6Hz.

(2) When the frequency 1 is 46 or higher, and lower than 184Hz, the control up to 50kHz can be performed.
If the frequency “n” is set to a value below 46Hz, it will be corrected to 46Hz.

(3) When the frequency 1 is 184 or higher, the control up to 50kHz can be performed.
If the frequency “n” is set to a value below 184Hz, it will be corrected to 184Hz.

*1: Assignment of control code (Specify with H constant)

H □ □ □ □ □ □ □ □

10: Fixed

Control assignment
0: Arbitrary table control

000: Fixed

Operation mode assignment
0: Incremental
1: Absolute

Output type assignment
0: CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
1: PLS+SIGN (Forward ON/Reverse OFF)

*2: Velocity range (Frequency) (Hz) <K constant>
1Hz to 50kHz [K1 to K50000 (Unit: Hz)]

*3: Target value range (ms) <K constant>
K−2,147,483,648 to K2,147,483,647

32−bit data value specified for the target value should be within the following range.

Assignment of control code		Range of settable target values
Operation mode	**Output type**	
Incremental	**Count: Addition**	Specify a positive value.
	Count: Subtraction	Specify a negative value.
Absolute	**Count: Addition**	Specify a value larger than the current value.
	Count: Subtraction	Specify a value smaller than the current value.

Sample program

Description of operation

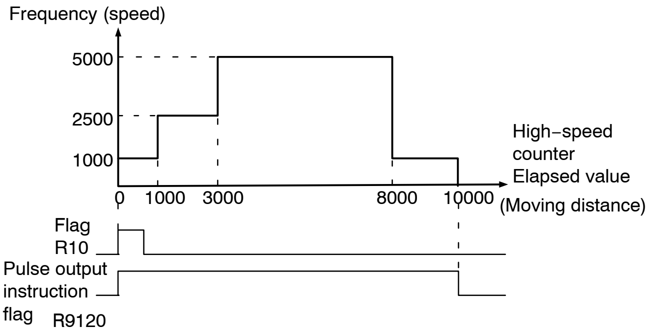
(1) When the trigger R10 in F174 (SP0H) instruction is on, the pulses are output at 1000 Hz from the specified channel.

(2) When 1000 pulses are counted at 1000Hz, the frequency changes to 2500Hz.

(3) When 3000 pulses are counted at 2500Hz, the frequency changes to 5000Hz.

(4) When 8000 pulses are counted at 5000Hz, the frequency changes to 1000Hz.

(5) When 1000 pulses are counted, the pulse output stops.



Note) When the trigger R10 in F174(SP0H) instruction turns on, the pulse output instruction flag will be on. Once the pulse output stops when the elapsed value reached 10000, the high−speed counter control flag will be off.

Setting and program

Control assignment: Arbitrary table control, operation mode: Incremental, the output type is CW/CCW.

```
 R0
─┤├──[ F1 DMV,H  10000000,DT100 ]    Control code: "H10000000"
     [ F1  DMV, K   1000, DT102  ]    Frequency 1: 1000 Hz
     [ F1  DMV, K   1000, DT104  ]    Target value 1: 1000 pulses
     [ F1  DMV, K   2500, DT106  ]    Frequency 2: 2500 Hz
     [ F1  DMV, K   3000, DT108  ]    Target value 2: 3000 pulses
     [ F1  DMV, K   5000, DT110  ]    Frequency 3: 5000 Hz
     [ F1  DMV, K   5000, DT112  ]    Target value 3: 5000 pulses
     [ F1  DMV, K   1000, DT114  ]    Frequency 4: 1000 Hz
     [ F1  DMV, K   2000, DT116  ]    Target value 4: 2000 pulses
     [ F1  DMV, K      0, DT118  ]    Stop of pulse output
 R10
─┤├──( DF )─[ F174 SP0H, DT100, K0 ]  Start of pulse output
```

Precautions during programming

The pulse output instruction flag is on until the pulse output stops after turning on the trigger in F174(SP0H) instruction.

When the frequency 1 is out the settable range, the operation error occurs. (If the data of the frequency 1 is 0, the operation ends without processing anything.)

When the second frequency or later is 0 or out of the specified range, the pulse output stops.

When the direction is reversed by executing the instruction with the specified target value, the pulse output stops.

Do not execute this instruction in the normal program and the interrupt program at the same time.

When the table point exceeds the data register (DT) area during the pulse output, the pulse output stops and the high−speed counter control flag turns off.

The target value must be specified within the range. If an outlying value is specified, the number of pulses that is different from the specified condition is output.

Flag conditions

- Error flag (R9007):
- Error flag (R9008):

  – Turns on when the area specified using the index modifier exceeds the limit.

  – Turns on when n is out of the specified range.

  – Turns on when the control code or frequency 1 is out of the specified range.

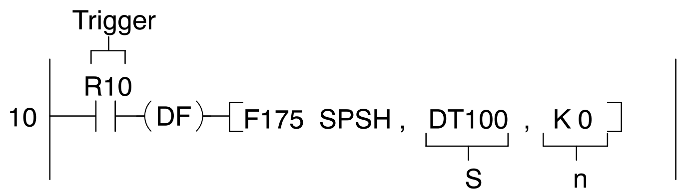
F175(SPSH) Pulse output (Linear interpolation)

Availability
FPΣ C32T2, C32T2H C28P2, C28P2H/FP−X

Outline Pulses are output from channel for 2 pulse output, in accordance with the parameters in the designated data table, so that the path to the target position forms a straight line.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	DF		
12	F175		(SPSH)
	DT		100
	K		0

S	Starting address of area containing the data table.
n	0: Fixed (FPΣ, FP−X Ry type) 0 or 2 (FP−X Tr type)

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

Pulses are output from channel ch0 (X−axis) and ch2 (Y−axis) (FP−X: ch1) when the corresponding control flag is off and the execution conditions are on.

For FPΣ

Channel no.	Output	Output method	
ch0 (for X−axis)	Y0	CW	PLS
	Y1	CCW	SIGN
ch2 (for Y−axis)	Y3	CW	PLS
	Y4	CCW	SIGN

For FP−X Ry type (AFPX−PLS)

Channel no.	Output	Output method	
ch0 (for X−axis) Cassete mounting part 1	Y100	CW	PLS
	Y101	CCW	SIGN
ch1 (for Y−axis) Cassete mounting part 2	Y200	CW	PLS
	Y201	CCW	SIGN

For FP−X Tr type

Channel no.	Output	Output method	
ch0 (for X−axis)	Y0	CW	PLS
	Y1	CCW	SIGN
ch1 (for Y−axis)	Y2	CW	PLS
	Y3	CCW	SIGN
ch2 (for X−axis)	Y4	CW	PLS
	Y5	CCW	SIGN
ch3 (for Y−axis)	Y6	CW	PLS
	Y7	CCW	SIGN

Note) For the FP−X Tr type, the combinations of ch0 (X axis) and ch1 (Y axis), and ch2 (X axis) and ch3 (Y axis) can be used.
Note) As there is no ch3 for C14T and C14TD, only the combination of ch0 and ch1 can execute the linear interpolation.

The control code, initial speed, maximum speed, acceleration/deceleration time, and target value are specified by creating the data table “S” to “S+11” on the following page using the user program.

If the frequency is set to 40 kHz or more, specify a duty of 1/4 (25%).

If the frequency for ch2 or ch3 of FP−X Tr type is set to 10kHz or more, specify a duty of 1/4 (25%).

Table of areas used

For FPΣ

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT90044, DT90045	DT90046, DT90047
ch2	R903C	DT90200, DT90201	DT90202, DT90203

For FP−X Ry type

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R911C	DT90348, DT90349	DT90350, DT90351
ch1	R911D	DT90352, DT90353	DT90354, DT90355

For FP−X Tr type

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R911C	DT90348	DT90350
		DT90349	DT90351
ch1	R911D	DT90352	DT90354
		DT90353	DT90355
ch2	R911E	DT90356	DT90358
		DT90357	DT90359
ch3	R911F	DT90360	DT90362
		DT90361	DT90363

Note) There is no ch3 for C14T and C14TD.
Note) The pulse I/O cassette (AFPX−PLS) cannot be installed on the FP−X Tr type.
Note) Use the ch2 and ch3 at up to 20 kHz.

Precautions during programming

Designate settings for the target value and movement distance so they are within the following range.
−8,388,608 to +8,388,607

When using in combination with other positioning instructions like **F171**, designate so the target value is within the above range, even in those instructions.

When using in application requiring precision, check with the actual machine.

If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

During the time that the circular interpolation control flag R904E is on, the pulse output instructions F166 to F176 cannot be executed.

When using this instruction for FPΣ, set the channels corresponding to system registers 400 and 401 to "Not set as high−speed counter".

If you perform a rewrite during RUN when pulse output is taking place, more pulses than the setting may be output.

When using this instruction for FP−X, set the pulse output by the system register.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used.
  - The "n" is other than 0.
  - The data "S, S+1 to S+10, S+11" of data table are outside specification range.
  - The composite speed designation satisfies:
    Initial speed "S+2, S+3" > Maximum speed "S+4, S+5"
  - The composite speed designation satisfies:
    Maximum speed "S+4, S+5" > 100kHz
  - The composite speed designation satisfies:
    Maximum speed "S+4, S+5" > 20kHz when outputting ch2 or ch3 with FP−X Tr type.
  - Incremental mode is designated and the value of "current value + movement distance" is outside the range −8388608 to +8388607.
  - Absolute mode is designated and the target value is outside the range −8388608 to +8388607.
  - With the FP−X, the pulse output has not been set by the system register.

Setting the data table

[S]	Control code	(*1)
[S+2]	Composite speed Initial speed Fmin(Hz)	(*2)
[S+4]	Composite speed Maximum speed Fmax(Hz)	(*2)
[S+6]	Acceleration/Deceleration time T (ms)	(*3)
[S+8]	X−axis (CH0) Target value (Movement distance)	(*4)
[S+10]	Y−axis (FPΣ: CH2, FP−X: CH1) Target value (Movement distance)	(*4)
[S+12]	X−axis (CH0) component speed Initial speed Fxmin	(*5)
[S+14]	X−axis (CH0) component speed Maximum speed Fxmax	(*5)
[S+16]	Y−axis (FPΣ: CH2, FP−X: CH1) component speed Initial speed Fymin	(*5)
[S+18]	Y−axis (FPΣ: CH2, FP−X: CH1) component speed Maximum speed Fymax	(*5)
[S+20]	X−axis (CH0) frequency range	(*6)
[S+21]	Y−axis (FPΣ: CH2, FP−X: CH1) frequency range	(*6)
[S+22]	X−axis (CH0) number of acceleration/deceleration steps	(*7)
[S+23]	Y−axis (FPΣ: CH2, FP−X: CH1) number of acceleration/deceleration steps	(*7)

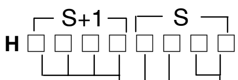
Setting area ([S] to [S+10])

Designated with user program

Operation result storage area ([S+12] to [S+23])

Parameters for each axis component, calculated due to instruction execution, are stored here.

(*1): Specification of control code (specify with H constant)



0: Fixed

Duty (on width)
0: Duty 1/2 (50%)
1: Duty 1/4 (25%)

0: Fixed

Operation mode and output method

00: Incremental	CW/CCW	
02: Incremental	PLS + SIGN	(forward off / reverse on)
03: Incremental	PLS + SIGN	(forward on / reverse off)
10: Absolute	CW/CCW	
12: Absolute	PLS + SIGN	(forward off / reverse on)
13: Absolute	PLS + SIGN	(forward on / reverse off)

(*2): Composite speed (Initial speed, Maximum speed) (Hz) <K constant>

1.5Hz to 100kHz [K1 to K100000]
However, 1.5Hz is for an angle of 0deg or 90deg only.
Also, specify K1 when specifying 1.5 Hz.

If the component speed drops lower than the minimum speed for each frequency range, then the speed will become the corrected component speed, so be careful. (See *6)
When simultaneously using a high−speed counter, periodical interrupt or PLC link, do not set to 60kHz or higher.
If initial speed is set equal to maximum speed, pulses will be output with no acceleration/deceleration.
Set the composite speed so that component speed of each axis is 1.5 Hz or greater.
Composite speed (initial speed): 30 kHz or lower

Note:
Cautions regarding specification of composite speed (initial speed)
The trajectory might not be linear if the initial composite speeds for CH0 and CH2 are not 1.5 Hz or higher in the formula below (when the formula below can't be worked out).

$$f \geqq \frac{1.5 \sqrt{(\Delta x2+\Delta y2)}}{\Delta x}$$

Δx: Short CH of distance between target and current value
Δy: Long CH of distance between target and current value
When using ch2, ch3 of FP−X Tr type, 1.5Hz to 20kHz [K1 to K20000]

(*3): Acceleration/deceleration time (ms) "K constant"
K0 to K32767
If this is 0, pulses will be output for the initial speed (composite speed) as is, with no acceleration/deceleration.

(*4): Target value
K−8388608 to K8388607
When operating only one axis,
a) In incremental mode, set the target value for the axis which will not be operated to 0.
b) In absolute mode, set the target value for the axis which will not be operated the same as the current value.
Note: Infinite feed is not possible during linear interpolation.

(*5): Component speed (Initial speed and maximum speed of each axis)
This is stored as 2 words in real numbers type.

$$\text{X−axis component speed} = \frac{(\text{Composite speed}) \times (\text{X−axis movement distance})}{\sqrt{((\text{X−axis movement distance})^2 + (\text{Y−axis movement distance})^2)}}$$

$$\text{Y−axis component speed} = \frac{(\text{Composite speed}) \times (\text{Y−axis movement distance})}{\sqrt{((\text{X−axis movement distance})^2 + (\text{Y−axis movement distance})^2)}}$$

Example:
Even if the initial speed is corrected (See *6), the calculation value will be stored as is in the operation result storage area.

(*6): Frequency range
The system automatically selects the frequency range for each component of each axis.
Range 0: 1.5Hz to 9.8kHz
Range 1: 48Hz to 100kHz
Range 2: 191Hz to 100kHz
a) If maximum speed ≦ 9800Hz
If initial speed < 1.5Hz, initial speed is corrected to 1.5Hz, and range 0 is selected.
If initial speed ≧ 1.5Hz, range 0 is selected.
b) If 9800Hz < maximum speed ≦ 100000Hz,
If initial speed < 48Hz, initial speed is corrected to 48Hz, and range 0 is selected.
If 48Hz ≦ initial speed < 191Hz, range 1 is selected.
If initial speed ≧ 191Hz, range 2 is selected.

(*7): Number of acceleration/deceleration steps
The system automatically calculates the number of acceleration/deceleration steps in the range 0 to 60 steps.
If the operation result is 0, pulses are output for the initial speed (composite speed) as is, with no acceleration/deceleration.
The number of acceleration/deceleration steps is found using the formula:
acceleration/deceleration time (ms) x component initial speed (Hz).

Example:

With incremental, initial speed 300Hz, maximum speed 5kHz, acceleration/deceleration time 0.5s, CH0 target value 1000, CH2 target value 50

$$\text{CH0 component initial speed} = \frac{300 \times 1000}{\sqrt{(1000^2 + 50^2)}} = 299.626\text{Hz}$$

$$\text{CH2 component initial speed} = \frac{300 \times 50}{\sqrt{(1000^2 + 50^2)}} = 14.981\text{Hz}$$

CH0 number of acceleration/deceleration steps = $500 \times 10^{-3} \times 299.626 \fallingdotseq 147.8 \Rightarrow 60$ steps

CH2 number of acceleration/deceleration steps = $500 \times 10^{-3} \times 14.981 \fallingdotseq 7.4 \Rightarrow 7$ steps

Note: With FP−X, CH2 is CH1.

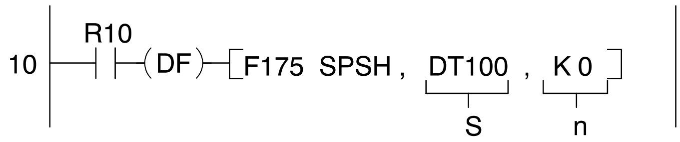
F175 (SPSH) Pulse output (Linear interpolation)

Availability
FP0R

Outline Pulses area output from channel for 2 pulse output, in accordance with the parameters in the designated data table, so that the path to the target position forms a straight line.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	10
11	DF		
12	F175		(SPSH)
	DT		100
	K		0

S	Starting 16−bit area for registering data tables
n	Channels intended for pulse output (n=0 to 3)

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant	Index modifier
	WX	WY	WR	SV	EV	DT	I	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding pulse output instruction flag is off and the trigger is on, pulses are output from channel ch0 (X−axis) and ch2 (Y−axis), or ch3 (X−axis) and ch4 (Y−axis).

Channel no.	Output	Output method	
ch0 (for X−axis)	Y0	CW	PLS
	Y1	CCW	SIGN
ch2 (for Y−axis)	Y3	CW	PLS
	Y4	CCW	SIGN
ch3 (for X−axis)	Y5	CW	PLS
	Y6	CCW	SIGN
ch4 (for Y−axis)	Y7	CW	PLS
	Y8	CCW	SIGN

The control code, initial speed, maximum speed, acceleration/deceleration time, and target value are specified by creating the data table ”S” to ”S+11” on the following page using the user program.

When the elapsed value reaches the final target value, the pulse output stops.

Table of areas used

FP0R

Pulse output channel No.	Pulse output instruction flag	Elapsed value area	Target value area	Target value area for match ON/OFF	Correction speed of initial speed
ch0	R9120	DT90400 to DT90401	DT90402 to DT90403	DT90404 to DT90405	DT90406
ch1	R9121	DT90410 to DT90411	DT90412 to DT90413	DT90414 to DT90415	DT90416
ch2	R9122	DT90420 to DT90421	DT90422 to DT90423	DT90424 to DT90425	DT90426
ch3	R9123	DT90430 to DT90431	DT90432 to DT90433	DT90434 to DT90435	DT90436

Precautions during programming

Designate settings for the target value and movement distance so they are within the following range.

K−8,388,608 to +8,388,607

When using in combination with other positioning instructions like F171, designate so the target value is within the above range, even in those instructions.

When using in application requiring precision, check with the actual machine.

If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

Setting of data table

The linear interpolation can be performed with either combination of (CH0 and CH1) or (CH2 and CH3). The acceleration time and deceleration time can be specified individually.

S S+1	Control code	(*1)	Setting area Designated with user program
S+2 S+3	Composite speed: Initial speed (Hz)	(*2)	
S+4 S+5	Composite speed: Target speed (Hz)	(*2)	
S+6 S+7	Acceleration time (ms)	(*3)	
S+8 S+9	Deceleration time (ms)	(*3)	
S+10 S+11	X−axis target value	(*4)	
S+12 S+13	Y−axis target value	(*4)	
S+14 S+15	X−axis component speed: Initial speed (Hz)	(*5)	Operation result storage area Parameters for each axis component, calculated due to instruction execution, are stored here.
S+16 S+17	X−axis component speed: Target speed (Hz)		
S+18 S+19	Y−axis component speed: Initial speed (Hz)		
S+20 S+21	Y−axis component speed: Target speed (Hz)		

***1: Assignment of control code (Specify with H constant)**

H □ □ □ □ □ □ □ □

10: Fixed

Control assignment
0: Interpolation

Type
0: Linear interpolation

0: Fixed

Output assignment
0: Pulse output
1: Calculation only

Operation mode assignment
0: Incremental
1: Absolute

Output type assignment
0: CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

*As for the output assignment

When starting the instruction with the setting of “1: Calculation only”, the pulse output is not performed.
When starting the instruction with the assignment of the same channel and the same parameter after executing this instruction once for a channel, it can be started at high speed. (It is the same for the both cases of “Pulse output” and “Calculation only”.)
However, if a parameter other than the parameter used for the previous execution is specified, the high−speed startup cannot be performed.

Note) The same parameter means that all the parameters other than the output assignment are the same.

Output type

Incremental <Relative control>

The pulses specified for the target value are output.

Mode selection / Target value	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When positive	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When negative	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

Absolute <Absolute value control>

The pulse that is the difference between the specified target value and the current value is output.

Mode selection / Target value	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
When target value is larger than current value	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
When target value is smaller than current value	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

[Explanation of pulse output operation]
Pulses are output using a duty of 25% fixedly.
When using the PLS +SIGN method, pulses will be output approx. 300 us later after the output of direction signal. (The characteristics of a motor driver is considered.)

***2: Composite speed range (Initial speed, Maximum speed) (Hz) <K constant>**

6.0Hz to 50kHz [K6 to K50000]

(However, 6.0 Hz is for an angle of 0 deg or 90 deg only. Also, specify K6 when specifying 6.0 Hz.)

– When specifying K1 to K5, it is the same as 6.0 Hz (K6).

– If initial speed is set equal to maximum speed, pulses will be output with no acceleration/deceleration.

– Set the composite speed so that component speed of each axis is 6 Hz or greater.

– Composite speed (Initial speed): 30Hz or less

Note) Cautions regarding specification of composite speed (initial speed)

– The trafectory might not be linear if the initial composite speeds for CH0 and CH2 are not 6.0 Hz or higher in the formula below.

* When the formula below can't be worked out.

$$f \geqq \frac{6.0 \sqrt{(\Delta x^2 + \Delta y^2)}}{\Delta x}$$

Δx: Short CH of distance between target and current value

Δy: Long CH of distance between target and current value

***3: Acceleration time (ms), Deceleration time (ms) <K constant>**

K0 to K32767

If this is 0, pulses will be output for the initial speed (composite speed) as is, with no acceleration/deceleration.

Note: Specify the same value for the acceleration time and deceleration time.

***4: Target value (Movement distance)**

K−8388608 to K8388607

When operating only one axis,

a) In increment mode, set the target value for the axis which will not be operated to 0.

b) In absolute mode, set the target value for the axis which will not be operated the same as the current value.

Note: Infinite feed is not possible during linear interpolation.

***5: Component speed (Initial speed and and maximum speed of each axis)**

This is stored as 2 words in real numbers type.

$$\text{X−axis component speed} = \frac{(\text{Composite speed}) \times (\text{X−axis movement distance})}{\sqrt{((\text{X−axis movement distance})^2 + (\text{Y−axis movement distance})^2)}}$$

$$\text{Y−axis component speed} = \frac{(\text{Composite speed}) \times (\text{Y−axis movement distance})}{\sqrt{((\text{X−axis movement distance})^2 + (\text{Y−axis movement distance})^2)}}$$

Component speed and correction

Note the following characteristics according to the component speed of the initial speed calculated by the above formula *5.

(1) When the initial speed is 1 or higher, and lower than 46Hz, the control up to the maximum frequency to the degree of 10kHz can be performed.

(2) When the initial speed is 46 or higher, and lower than 184Hz, the control up to 50kHz can be performed.

(3) When the initial speed is 184 or higher, the control up to 50kHz can be performed.

The speed error around 50 kHz will be smallest.

Note that the vector of the composite speed may be deviated at the time the pulse output starts or stops when the value has been corrected.

Compare with the correction speed of initial speed in the special registers to check whether or not the specified initial speed is corrected.

Flag conditions

- Error flag (R9007):
- Error flag (R9008):
  - Turns on when the area is exceeded when an index modifier is used.
  - Turns on when the “n” is other than 0.
  - Turns on when the data “S, S+1 to S+!0, S+11” of data table are outside specification range.
  - Turns on when the composite speed designation satisfies: Maximum speed “S+4, S+5” > 50kHz
  - Turns on when increment mode is designated and the value of “current value + movement distance” is outside the range −8388608 to +8388607.
  - Turns on when absolute mode is designated and the target value is outside the range −8388608 to +8388607.
  - The acceleration time and deceleration time has not been set to the same value.

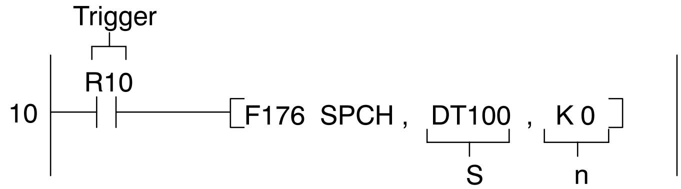
F176(SPCH) Pulse output (Circular interpolation)

Availability
FPΣ C32T2, C32T2H C28P2, C28P2H

Outline Pulses are output from channel ch0 and ch2, in accordance with the parameters in the designated data table, so that the path to the target position forms an circular.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F176		(SPCH)
	DT		100
	K		0

S	Starting address of area containing the data table.
n	0: Fixed

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

Pulses are output from the channel ch0 (for X−axis) and ch2 (for Y−axis) when the corresponding control flag turns off and the execution condition (trigger) turns on.

For FPΣ

Channel no.	Output	Output method	
ch0 (for X−axis)	Y0	CW	PLS
	Y1	CCW	SIGN
ch2 (for Y−axis)	Y3	CW	PLS
	Y4	CCW	SIGN

Designate the control code, composite speed, target position and pass position by creating the data table “S” to “S+11” on the next page with the user program.

Table of areas used

Channel no.	Control flag	Elapsed value area	Target value area
ch0	R903A	DT90044, DT90045	DT90046, DT90047
ch2	R903C	DT90200, DT90201	DT90202, DT90203

Flag for circular interpolation

R904E: Circular interpolation control flag

Turns ON when circular interpolation instruction F176 starts up and maintains that state until the target value is reached.

R904F: Set value change confirmation flag

When conducting control with the continuous mode for performing continuous circular interpolation actions, use this after circular interpolation instruction startup when overwriting the next target value.

Precautions during programming

Assume that the execution conditions for this instruction always hold. When the execution conditions are off, pulse output stops.

During the time that the circular interpolation control flag R904E is on, the pulse output instructions F166 to F176 cannot be executed.

When the target value has not been reached and the execution condition is off, circular interpolation control flag R904E turns on and other positioning instructions F171 to F176 cannot start up.

When restarting, use pulse output control instruction F0, below, to reset the pulse output instruction. This operation resets the Control flag for circular interpolation (R904E).

Designate settings for the target value and movement distance so they are within the following range.
−8,388,608 to +8,388,607

When using in combination with other positioning instructions like **F171**, designate so the target value is within the above range, even in those instructions.

The accuracy of circular interpolation may degrade if the scan time lengthens.

If both the regular program and the interrupt program contain code for the same channel, make sure both are not executed simultaneously.

If you make the current position equal the target value when specifying the center position setting method, a circle drawing operation will result.

When using in application requiring precision, check with the actual machine.

When using this instruction, set the channels corresponding to system registers 400 and 401 to “Not set as high−speed counter”.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area is exceeded when an index modifier is used.
  - The “n” is other than 0.
  - The data “S, S+1 to S+10, S+11” are outside specification range.
  - Incremental mode is designated and the value of “current value + movement distance” is outside the range −8388608 to +8388607.
  - Absolute mode is designated and the target value is outside the range −8388608 to +8388607.

  With pass position setting method,
  - Current position S ≒ Target position E
  - Current position S ≒ Pass position P
  - Pass position P ≒ Target position E
  - Current position S, Pass position P and Target position E approximate a straight line.

  With center position setting method,
  - Center position O=Target position E
  - Center position O=Current position S

Setting the data table

Pass position setting method

Address	Data	Note	Area
S, S+1	Control code	(*1)	**Setting area** Designated with user program
S+2, S+3	Composite speed (Frequency) Fv (Hz)	(*2)	
S+4, S+5	X−axis (CH0) Target position	(*3)	
S+6, S+7	Y−axis (CH2) Target position	(*3)	
S+8, S+9	X−axis (CH0) Pass position	(*3)	
S+10, S+11	Y−axis (CH2) Pass position	(*3)	
S+12, S+13	Radius		**Operation result storage area** Parameters for each axis component, calculated due to instruction execution, are stored here.
S+14, S+15	X−axis (CH0) Center position		
S+16, S+17	Y−axis (CH0) Center position		

Center position setting method

Address	Data	Note	Area
S, S+1	Control code	(*1)	**Setting area**
S+2, S+3	Composite speed (Frequency) Fv (Hz)	(*2)	
S+4, S+5	X−axis (CH0) Target position	(*3)	
S+6, S+7	Y−axis (CH2) Target position	(*3)	
S+8, S+9	X−axis (CH0) Center position		
S+10, S+11	Y−axis (CH2) Center position		
S+12, S+13	Radius		**Operation result storage area**

(*1): Specification of control code (specify with H constant)

S+1 S

H□□□□□□□□

0: Fixed

Operation connection mode(*4)
0: Stop
1: Continue

Rotation direction (*5)
0: from CH2−CW axis to CH0−CW axis
1: from CH0−CW axis to CH2−CW axis

Circular method (*6)
0: Pass position setting method
1: Center position setting method

Operation mode and output method

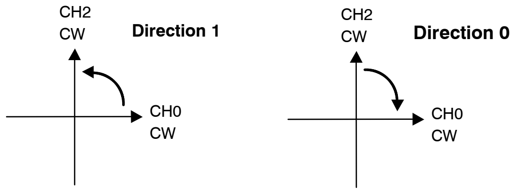
00: Incremental	CW/CCW	
02: Incremental	PLS + SIGN	(forward off / reverse on)
03: Incremental	PLS + SIGN	(forward on / reverse off)
10: Absolute	CW/CCW	
12: Absolute	PLS + SIGN	(forward off / reverse on)
13: Absolute	PLS + SIGN	(forward on / reverse off)

(*2): Composite speed (Frequency) “K constant”
100 Hz to 20 kHz [K100 to K20000]
As a guide, keep the composite speed within the range of the formula below.
Fv[Hz] <= radius[pulse] x 10/scantime[ms]

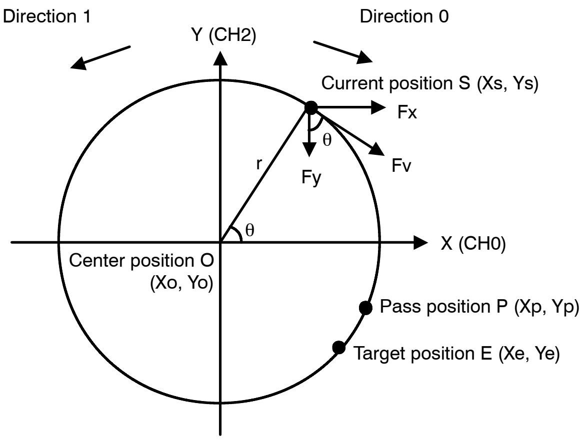
(*3): Target position and pass position
K−8388608 to K8388607

(*4): Operation connection mode
Stop:
When stop (0) is specified, it will stop when the target position is reached.
Continue:
When the following circular interpolation data table is overwritten when continue (1) is specified after circular interpolation action begins, the following circular interpolation begins when the first circular interpolation that was started up finishes (target position reached). To finish, specify stop (0) for this flag (operation connection mode) after the last circular interpolation action has started.

(*5): Rotation direction
Pulses are output according to the designated direction. Operation differs, as indicated below, depending on the pass position and rotation direction setting.



(*6): Circular method
Pass position setting method:
The center position and the radius of the circular are calculated by specifying the pass and target positions for the current position.
Center position setting method:
The radius of the circular is calculated by specifying the center and target positions for the current position.



Let CH0 be the X−axis, and CH2 be the Y−axis.

Fv:	Composite speed	O (Xo, Yo):	Center point (Center position)
Fx:	X−axis component speed	S (Xs, Ys):	Start point (Current position)
Fy:	Y−axis component speed	P (Xp, Yp):	Pass point (Pass position)
r:	Radius	E (Xe, Ye):	End point (Target position)

$Fx = Fv \sin\theta = Fv \frac{|Ye - Yo|}{r}$ $Fy = Fv \cos\theta = Fv \frac{|Xe - Xo|}{r}$

F177 (HOME) Pulse output (Home return)

Availability
FP0R

Outline Performs the home return operation on the specified pulse output channels.

Program example

Ladder Diagram

Address	Instruction		
10	ST	R	10
11	DF		
12	F177	(HOME)	
	DT		100
	K		2

S	Starting 16−bit area for registering data tables
n	Channels intended for pulse output

Operands

Operand	Relay			Timer/Counter		Register	Index register	Constant		Index modifier
	WX	WY	WR	SV	EV	DT	I	K	H	
S	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	A
n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	A	N/A

A: Available
N/A: Not Available

Description

When the corresponding pulse output instruction flag is off and the flag is on, pulses are output from the specified channels to perform the home return operation.

* For using the pulse output function, it is required to set how to use input/output by system registers.

* C10 and C14 is relay output type, therefore, pulse output cannot be performed.

Description of operation mode

Home return (Type 0) : The home input is available in all sections.

Home return (Type 1) : The home input is available only in the section of creep speed.

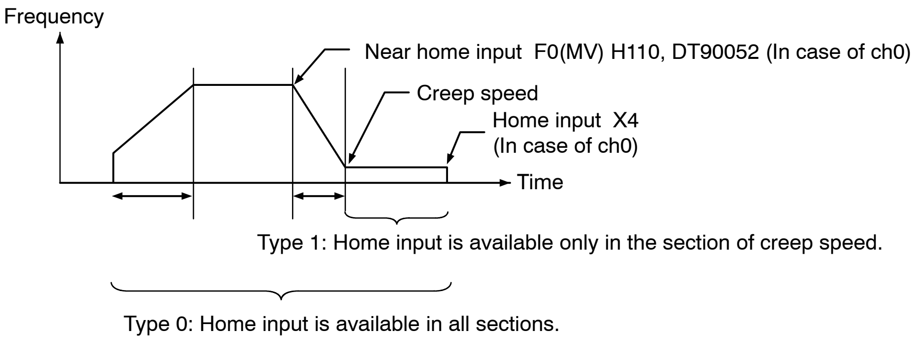


Table of areas used

FP0R

Pulse output channel No.	Output	Output type		Near home input	Home input	Deviation counter clear		Pulse output instruction flag	Elapsed value area	Target value area
						C16	C31,T32,F32			
ch0	Y0	CW	PLS	DT90052 <bit4>	X4	Y6	Y8	R9120	DT90400 DT90401	DT90402 DT90403
	Y1	CCW	SIGN							
ch1	Y2	CW	PLS		X5	Y7	Y9	R9121	DT90410 DT90411	DT90412 DT90413
	Y3	CCW	SIGN							
ch2	Y4	CW	PLS		X6	–	YA	R9122	DT90420 DT90421	DT90422 DT90423
	Y5	CCW	SIGN							
ch3	Y6	CW	PLS		X7	–	YB	R9123	DT90430 DT90431	DT90432 DT90433
	Y7	CCW	SIGN							

* In case of C16 type
Note1: As Y6 and Y7 of CH3 is also used for the deviation counter clear output of CH0/1, either one of those functions can be used.

* In case of C32, T32, F32
Note1: As X4, X5, X6 and X7 for the home return is also used for the high−speed counter, either one of those functions can be used.

S S+1	Control code	
S+2 S+3	Initial speed	Velocity range (Frequency) (Hz) 1Hz to 50kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time up to the target speed: Acceleration time range (ms) K1 to K32760 (Unit: ms)
S+8 S+9	Deceleration time (ms)	Deceleration time from the target speed: Deceleration time range (ms) K1 to K32760 (Unit: ms)
S+10 S+11	Creep speed (Hz)	Speed range (frequency) (Hz) 1Hz to 50kHz [K1 to k50000 (Unit: Hz)]
S+12 S+13	Deviation counter clear Signal output time	k0 to k200 k0 = Not output deviation counter clear signal. Kn = n*0.5ms

Note the following characteristics according to the specified initial speed.

(1) When the initial speed is 1 or higher, and lower than 46Hz, the control up to the maximum frequency to the degree of 10kHz can be performed.
If the frequency is higher than that, the speed error will be larger.

(2) When the initial speed is 46 or higher, and lower than 184Hz, the control up to 50kHz can be performed.

(3) When the initial speed is 184 or higher, the control up to 50kHz can be performed.
The speed error around 50kHz will be smallest.

Assignment of control code (Specify with H constant)

H □ □ □ □ □ □ □ □

10: Fixed

Control assignment
0: Home return

Control assignment type
0: Home return type 0
0: Home return type 1

00: Fixed

Operation mode assignment
0: Forward
1: Reverse

Output type assignment
0: CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

Output type

Mode selection / Operation mode	CW/CCW	PLS+SIGN Forward OFF Reverse ON	PLS+SIGN Forward ON Reverse OFF	Elapsed value
Forward	Output from CW	Pulse output when direction output is off	Pulse output when direction output is on	Addition
Reverse	Output from CCW	Pulse output when direction output is on	Pulse output when direction output is off	Subtraction

[Explanation of pulse output operation]
Pulses are output using a duty of 25% fixedly.
When using the PLS +SIGN method, pulses will be output approx. 300 us later after the output of direction signal. (The characteristics of a motor driver is considered.)

Precautions during programming

Even in the state that the home input turns on, once this instruction is executed, the pulse output starts.

If the near home input becomes effective during the acceleration, the deceleration operation will start.

When the same channel is described in a normal program and interrupt program both, do not execute them at the same time.

This instruction cannot be executed when the corresponding control flags to each channel are on.

If rewriting during RUN is performed during pulse output, pulses more than the setting may be output.

When performing the software reset, count prohibition, pulse output stop or near home operation, refer to the F0(MV) instruction, pulse output.

Flag conditions

- Error flag (R9007):
- Error flag (R9008):
  - Turns on when the area specified using the index modifier exceeds the limit.
  - Turns on when n is out of the specified range.
  - Turns on when each data of [S,S+1] to [S+4,S+5] is out of the specified range.
  - Turns on when initial speed [S+2,S+3] > target value [S+4,S+5].

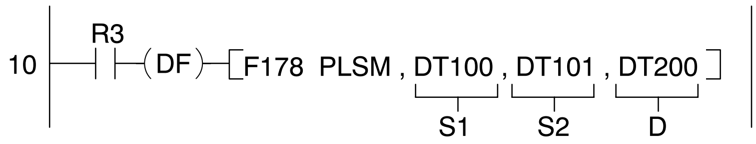
F178(PLSM) Input pulse measurement

Availability
FP0R

Outline Measures the number of pulses and the pulse period of the specified high−speed counter channel when using the high−speed counter function.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	3
11	F178	(PLSM)
	DT	100
	DT	101
	DT	200

S1	Specification of channel No. and No. of moving average.
S2	Counting period
D	Starting address of the destination area

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WR	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	A	N/A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	N/A	N/A	A	A	A
D	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) I0 to ID

A: Available
N/A: Not Available

Description

The number of pulses or the pulse period of the specified high−speed counter channel is measured based on the control data specified by [S1].

In the measurement of the number of pulses, the number of pulses of the specified high−speed counter counted during the period specified by S2 is counted.
The average of the number of moving average is calculated with the specified period and stored in D and D+1.
When the number of average is n, −1 is output during the (n*counting period) time after the execution of the instruction.

In the pulse period measurement in 1us unit, a period of 1 pulse right after the execution of this instruction is counted and stored in D+2 and D+3.

In the pulse period measurement in 1ms unit, the measured value is stored in D+4 and D+5 every time the measurement of a period of 1 pulse completes.

The same channel cannot be specified at the same time with other high−speed counter control instructions <F165(CAM0), F166(HC1S), F167(HC1R)>.
An exclusive control is implemented by the high−speed counter control flags (R9110 to R9115).

The number of channels that the instruction can be executed simultaneously is two.

The trigger should be always ON while the pulse is being measured with this instruction.
Turning OFF the trigger stops the measurement.

Specification of each item

Specifying the channel number and number of moving average [S1]
Specify the channel number of the high−speed counter and number of moving average.
If necessary, specify the measurement of pulse period.

Setting of measurement limit for measuring period in 1ms unit
0: No measurement limit process
1:100ms
2:200ms
3:300ms
4:500ms
5:1s
6:2s
7:5s
8:10s
9:60s
A: Undefined any more

S1	n			

Measurement of pulse period H0 to H1

No. of moving average H1 to H5

Channel No. H0 to H5

0: Pulse period is not measured.
1: Pulse period is measured in 1us unit.
2: Pulse period is measured in 1ms unit.
3: Pulse period is measured in 1us unit and 1 ms unit.

Measurement limit process for period measurement

The measurement limit process is a function which sets the measurement value to −1 when the period measurement has not completed in a given amount of time.

When measuring period in 1us unit

When measurement timer overflow has occurred
The measurement value is set to −1 when a short period could be measured although a time more than 174ms has elapsed after the previous measurement request.

When measurement has not completed
The measurement value is set to −1 when measurement has not completed although a time more than 350ms has elapsed after the previous request.
Even when measurement has completed after that, the result is disregarded and measurement is requested again.

When measuring period in 1ms unit

The elapsed value is set to −1 when the result of checking the period measurement counter _plsCycleTime0 has exceeded the measurement limit specified for the above n.
Even when measurement has completed after that, the data is disregarded and measurement is requested again.

Specify the counting period for the number of pulses. [S2]
Specify it in 1ms unit. K1 to K5000 (1ms to 5s)

Specifying the starting number of the destination area where the pulse is output. [D]
Specify the starting number of the destination area where the pulse is output.

D,	D+1	No. of pulses (Moving average value)	The latest value is stored with the measurement period specified by S2.
D+2,	D+3	Pulse period (1us unit)	The period of 1 pulse right after this instruction is executed is stored.
D+4	D+5	Pulse period (1ms unit)	The latest value is updated every time the period of 1 pulse is measured after the execution of this instruction.

A maximum of approx. 174.7 ms can be measured in 1us unit.
A maximum of approx. 49.7 days can be measured in 1ms unit.

Period measurement data

When measurement starts, −1 is set.
When measurement limit is exceeded, −1 is set.

Precautions during programming

The same channel cannot be specified at the same time with other high−speed counter control instructions <F165(CAM0), F166(HC1S), F167(HC1R)>.
An exclusive control is implemented by the high−speed counter control flags (R9110 to R9115).

Once the instruction is executed, the pulse measurement function will be effective until the control is cleared with F0(MV) S, DT90052 instruction.

The number of channels that the instruction can be executed simultaneously is two.

F178 instruction cannot be executed if the high−speed counter function is not used.

Do not execute this instruction in the normal program and the interrupt program at the same time.

FP0R

High−speed counter channel No.	Control flag	Elapsed value area	Target value area
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323

Example of input pulse measurement setting

[Condition]

(1) Set the channel number to 0 and the number of moving average to 5. Specify the pulse period measurement in 1us unit.

(2) Set the counting period to 10ms.

```
 R9013
─┤├─[ F0  MV , H   150,    DT100  ]     Specification of high−speed counter channel 0,
                                         No. of moving average: 5 times, specification of pulse period measurement (in 1us unit)
     [ F0  MV , K   10,     DT101  ]     Period of counting the number of pulses
 R3
─┤├─[ F178 PLSM, DT100, DT101, DT200]    Start of input pulse measurement
```

Execution of program

When the internal relay R3 is ON, the operation is performed as follows.
When pulses are input with a frequency at 10kHz.

DT200 to DT201	No. of pulses (Moving average value)	→	100 pulses Calculates the number of input pulses every 10ms, and calculates the average of the past 5 times with a period of counting.
DT202 to DT203	Pulse period in 1us unit	→	100 μs (The value is k100.)
DT204 to DT205	Pulse period in 1ms unit	→	Becomes 0 ms.

Note: The final numbers of actual measured values may vary due to measurement error.

Flag conditions

- Error flag (R9007):
- Error flag (R9008):

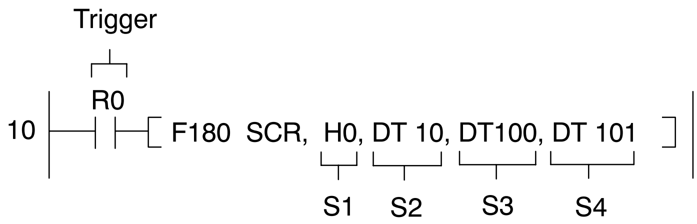
  – Turns on when the area specified using the index modifier exceeds the limit.
  – [S1] Turns on when the specified channel is out of the specified range.
  – [S1] Turns on when the number of moving average is out of the specified range.
  – [S2] Turns on when the counting period is out of the specified range.
  – [D] Turns on when the range data to be stored exceeds the area.
  – Turns on when the same channel has been already controlled with the same sorf of instruction.
  – Turns on when the number of execution channels is 3 or more.
  – Turns on when the high−speed counter has not been set for the specified channel by the system register.

Availability
FP-e

F180 (SCR) FP-e screen display registration

Outline Instruction to register the screen displayed in the N mode and S mode.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F180 (SCR) H 0 DT 10 DT 100 DT 101

S1	FP-e screen mode and number (Specify between 0 and 3.)
S2	Starting address of area specified for the FP-e display method
S3	Area for storing data to be displayed in the upper of the FP-e
S4	Area for storing data to be displayed in the lower of the FP-e

Note: A special data register cannot be specified for S4.

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
S1	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S3	A	A	A	A	A	A	A	A	N/A	N/A	A
S4	N/A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

Description

Register FP-e screens specified with S1 with the method whereby S2 to S2+2 is specified.
For S3, specify the address where data for display in the upper is stored. For S4, specify the address where data for display in the lower is stored.
When this instruction is executed, the registered screen is displayed in the FP-e panel.
To switch screens, use the mode switch on the FP-e, or instruction F180 or F181.

Specify the screens for setting with S1.
Specify the display method with S2, S2+1 and S2+2.
Specify the data to be displayed in the upper with S3.
Specify the data to be displayed in the lower with S4.

Note) For the numeric data display of S3 and S4, only 16-bit data is available.

Precautions during programming

This instruction cannot be used during the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The value of S1 or S2 exceeds the limit of specified range.

How to specify S1

Specify the type of FP−e mode.

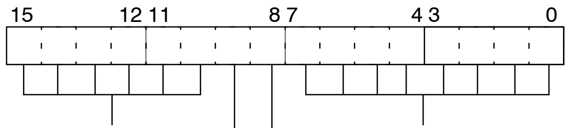
Value specified for S1	Type of mode
H0	N mode first screen
H1	N mode second screen
H2	S mode first screen
H3	S mode second screen

How to specify S2, S2+1, S2+2

With S2, S2+1 and S2+2 specify the display method of the screen specified with S1.
By writing the data below, the screen display method is specified in a 3−word range from the area specified with S2.
For example, when DT10 is specified for S2, DT10 to DT12 becomes the area below.

S2: First word

Specifies the method in which all units are displayed.



Undefined (bits 15 to 10)
[Specify 0]

Lower display setting (bit 9)
[0: display, 1: no display]

Upper display setting (bit 8)
[0: display, 1: no display]

Unit display setting (bits 7 to 0) [0: no display, 1: display]

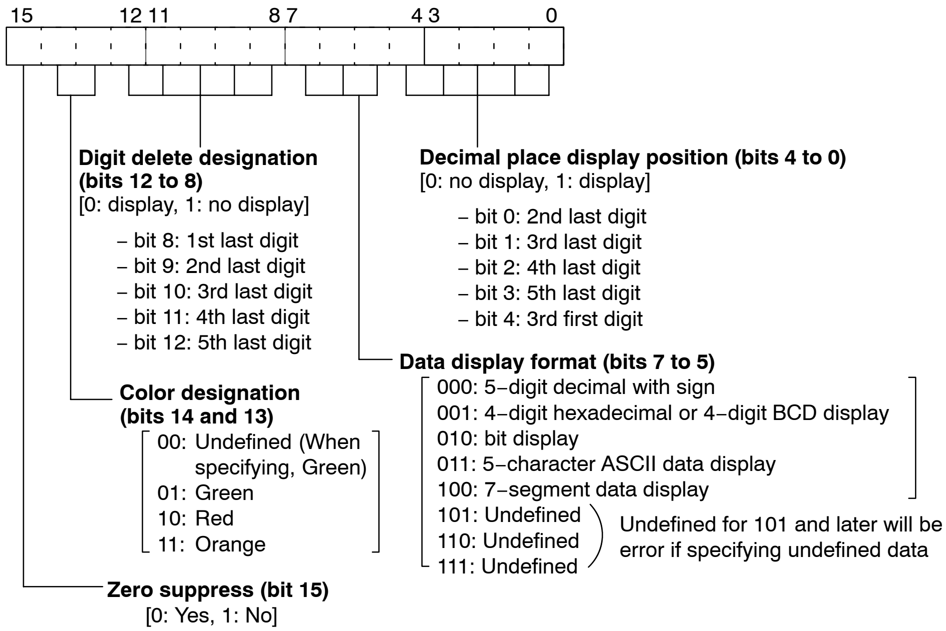
- bit 0: Unit “PV”
- bit 1: Unit “SV”
- bit 2: Unit “s”
- bit 3: Unit “m”
- bit 4: Unit “h”
- bit 5: Unit “℃”
- bit 6: Unit “F”
- bit 7: Unit “ ”

S2+1: Second word

Specifies the method for displaying data in the upper.
The bits shown in the figure below are allocated. Please specify with the H constant.

S2+2: Third word

Specifies the method for displaying data in the lower.
The bits shown in the figure below are allocated. Please specify with the H constant.



Remarks)

If displaying decimal point in the format of 5–digit decimal with sign, the value(s) before the decimal point should be displayed.

Example:

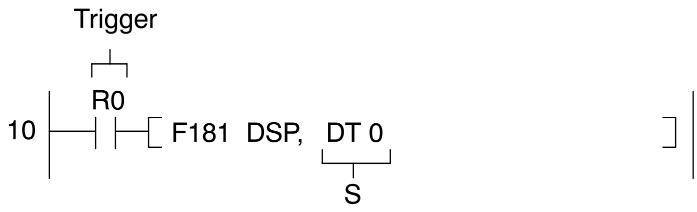
To change the color to red, put 10 for bits 14 and 13.
Specify in this way: 0100 0000 0000 0000 → H4000.

Availability
FP−e

F181 (DSP) FP−e screen display switching

Outline Specify the screen to be displayed on the FP−e.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10 11	ST R 0 F181 (DSP) DT 0

S	FP−e screen mode and number (Specify between 0 and 7.)

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
S	A	A	N/A	A	A	A	A	A	A	A	A

A: Available
N/A: Not Available

Description

Switches the FP−e screen to the screen of the mode specified with S.

How to specify S

Specify the type of FP−e mode.

Value specified for S	Type of mode
K0	N mode first screen
K1	N mode second screen
K2	S mode first screen
K3	S mode second screen
K4	R mode first screen
K5	R mode second screen
K6	I mode first screen
K7	I mode second screen

Precautions during programming

If specifying the value other than 0 to 7 for S, an operation error will occur.

This instruction cannot be used during the interrupt program.

Flag conditions

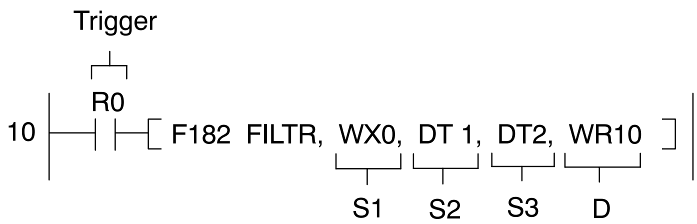
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The value of S is number except 0 to 7.

Availability
FP−X V2.0 or more FPΣ V3.10 or more FP0R

F182 (FILTR) Time constant processing

Outline The filter processing is executed for the specified bits and the bitwise results are output.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F182 (FILTR)
		WX 0
		DT 1
		DT 2
		WR 10

S1	16−bit area for storing object data for filter processing
S2	16−bit equivalent constant or 16−bit area for storing object bits for filter processing
S3	16−bit equivalent constant or 16−bit area for storing filter processing time
D	16−bit area for storing filter processing results

Operands

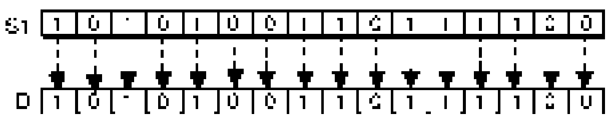
Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	N/A
N	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A

(*1) I0 to ID

A: Available
N/A: Not Available

Operation

In the 16−bit data stored in the area specified by S1, the bits of 0 specified by S2 are output directly, and the bits of 1 are output by performing the filter processing for the time specified by S3 (0 to 30000 in msec unit). The results are stored in the area specified by D in bit unit. (The bit positions are the same as S1.)



The bit position of S1 and D corresponds respectively.

Precautions during programming

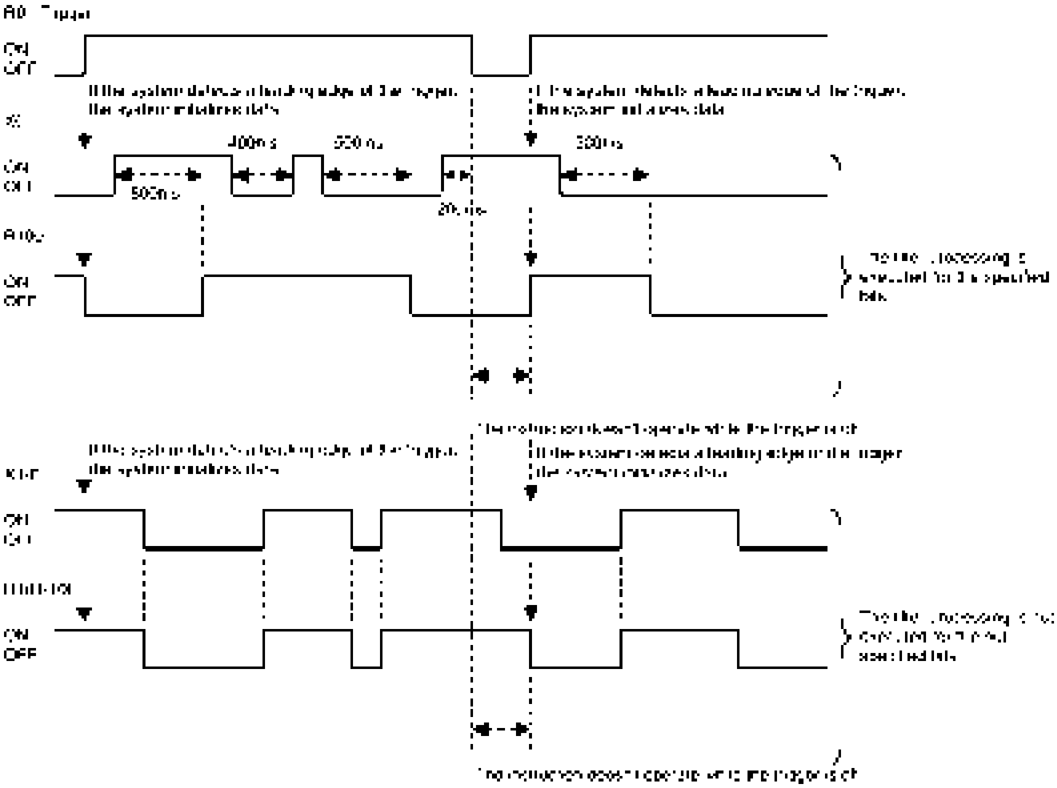
When the system detects a leading edge of the trigger, all the bits of the input specified by S1 is unconditionally output.

Max. 1 scan time error in the filter processing time occurs occasionally.

Explanation of example

The changes in values of R0 or X0 to XF, when the conditions prior to the execution of this instruction (R0=0) are as below, are explained with a time chart.

WX0 (Filter processing input data) = HA9BC
DT1 (Filter processing object bit) = H0001
DT2 8Filter processing time) = k500
WR10 (Filter processing result) = HFFFF



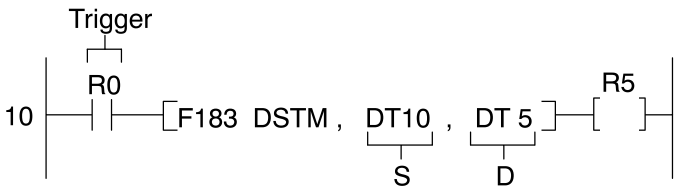
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The area specified using the index modifier exceeds the limit.
  - – The filter processing time specified by ”S3” is smaller than k0 or larger than k30000.

F183 (DSTM) Auxiliary timer (32-bit)

Outline Sets the 32−bit ON−delay timer for 0.01 s units (0.01 to 21474836.47 s)

Program example



Address	Instruction		
10	ST	R	0
11	F183		(DSTM)
	DT		10
	DT		5
16	OT	R	5

S	32-bit equivalent constant or lower 16-bit area of 32−bit data for timer set value
D	16-bit area for timer elapsed value

Operands

Operand	Relay			Timer/Counter		Register	Index register		Constant		Index modifier
	WX	WY	WR	SV	EV	DT	IX (*1)	IY	K	H	
S	A	A	A	A	A	A	A	N/A	A	A	N/A
D	N/A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A

A: Available
N/A: Not Available

(*1) With the FP0R, FPΣ, FP−X, FP2, FP2SH, and FP10SH, this is I0 to IC.

Explanation of example

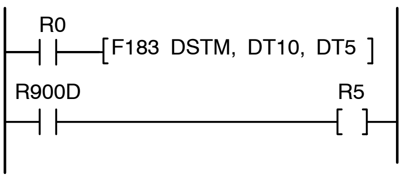
When the execution condition (trigger) has been satisfied, the auxiliary timer is activated, and the time equal to the values stored in data registers DT10 and DT11 x 0.01 seconds has elapsed, R5 goes on.

Description

This functions as a 32−bit addition−type On Delay timer set in 0.01−second units.
When the execution condition (trigger) is on, the elapsed time is added, and when the elapsed value (D + 1, D) (32 bits) exceeds the set value, the relays being used are turned on by the **OT** instruction which comes next in the program.

When the execution condition (trigger) is off, the elapsed value area is cleared to 0, and relays being used are turned off by the **OT** instruction.

When the time set for the special internal relay R900D has elapsed, the relay is turned on.
R900D can also be used as a timer contact.
(R900D is off when the execution condition (trigger) is off and while addition is being carried out.)



Operation is the same as that in the example shown above.

Timer set time

The timer setting is entered as a value of 0.01 x (timer set value).

The timer set value is specified as a K constant within the range of K1 to K2147483647.

The **F183 (DSTM)** is set between 0.01 and 21,474,836.47 seconds, in units of 0.01 seconds.

If the set value is K500, the set time will be 0.01 x 500 = 5 seconds.

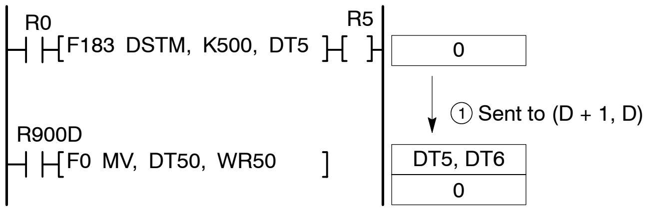
Precautions during programming

The area in which the set value is stored must be set so that the area specified for the elapsed value does not overlap any areas reserved for other timer or counter instructions, or memory areas used for high−level instruction operations.

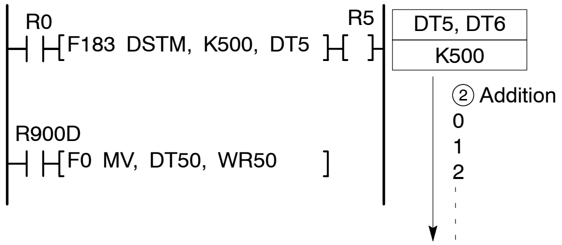
Because addition is carried out when operations are carried out, the program should be set up so that operations are carried out every scan.
(In cases such as programs where division is carried out, or for jump or loop instructions, where several operations are carried out during one scan, or where it was not possible to carry out any operation during the scan, correct results cannot be obtained.)

How the Auxiliary Timer Works

1 When the execution condition (trigger) changes from off to on, values of 0 are sent to the elapsed value area (D + 1, D).

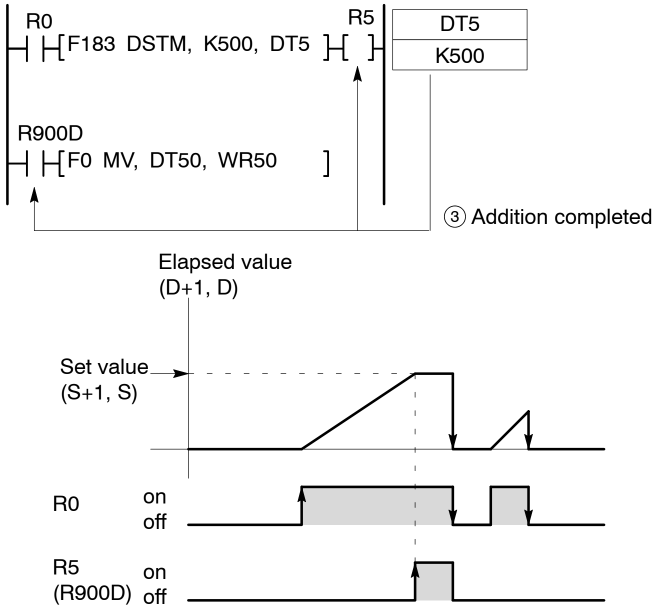


2 If the execution condition (trigger) stays on, the values in the elapsed value area (D + 1, D) are added.



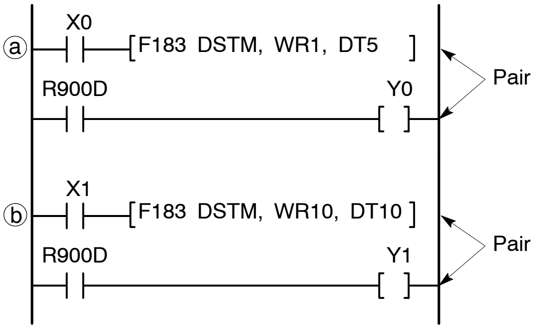
☛ next page

③ If the values in the elapsed value area (D +1, D) reach (S + 1, S), relays being used are turned on by the **OT** instruction which comes next in the program. The special internal relay R900D also goes on at this point.



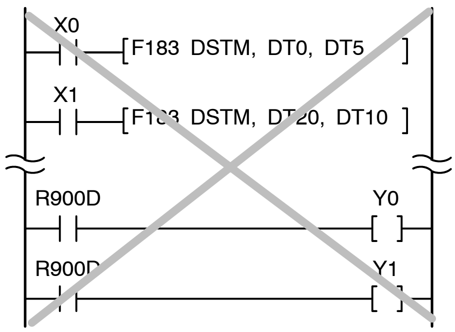
Precautions When Using R900D

If R900D is used and multiple auxiliary timers are being used, always use R900D in the line following the auxiliary timer instruction.



When timer ⓐ, which is activated by X0: on, expires, Y0 goes on. When timer ⓑ, which is activated by X1: on, expires, Y1 goes on.

If written as indicated below, R900D will not function correctly.

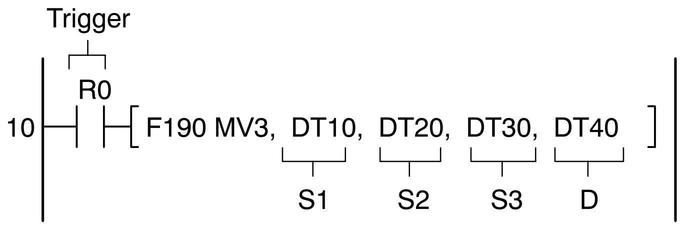


F190 (MV3) P190 (PMV3) Three 16-bit data move

Outline Copies three 16-bit data to the specified 48-bit area (3 words).
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P190 (PMV3)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F190		(MV3)
	DT		10
	DT		20
	DT		30
	DT		40

S1	16-bit equivalent constant or 16-bit area (source)
S2	16-bit equivalent constant or 16-bit area (source)
S3	16-bit equivalent constant or 16-bit area (source)
D	Starting 16-bit area of 48-bit area (3 words) (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

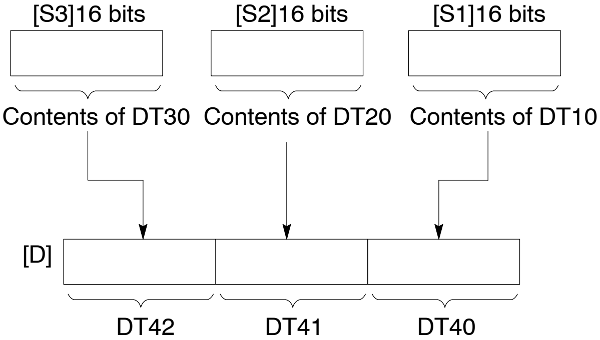
Explanation of example

When trigger R0 turns on,

– the contents of data register DT10 are copied to DT40.

– the contents of data register DT20 are copied to DT41.

– the contents of data register DT30 are copied to DT42.



Description

The 16-bit data or 16-bit equivalent constant specified by S1, S2 and S3 is copied to the area (3 words) specified by D when the trigger turns on.

Related instruction

To transfer two types of 16-bit data at once, use the **F7 (MV2)** instruction.

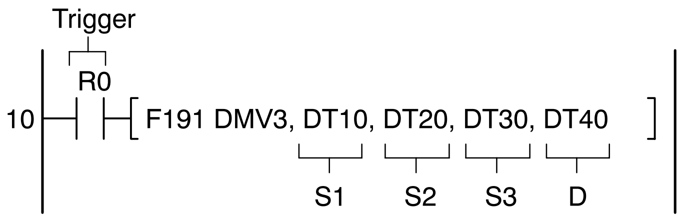
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F191 (DMV3) P191 (PDMV3) Three 32-bit data move

Outline Copies three 32-bit data to the specified 96-bit area (6 words).
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P191 (PDMV3)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F191 (DMV3)
		DT 10
		DT 20
		DT 30
		DT 40

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
S3	32-bit equivalent constant or lower 16-bit area of 32-bit data (source)
D	Starting 16-bit area of 6 words (96-bit area) (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

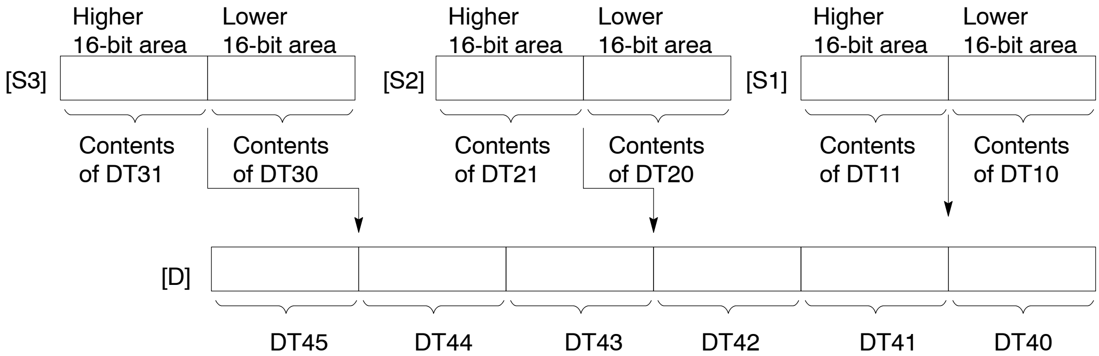
Explanation of example

When trigger R0 turns on,

– the contents of data register DT11 and DT10 are copied to data registers DT41 and DT40.

– the contents of data register DT21 and DT20 are copied to DT43 and DT42.

– the contents of data register DT31 and DT30 are copied to DT45 and DT44.



Description

The 32-bit data or 32-bit equivalent constant specified by S1, S2 and S3 is copied to the area (6 words) specified by D when the trigger turns on.

Related instruction

To transfer two types of 32-bit data at once, use the **F8 (DMV2)** instruction.

Flag conditions

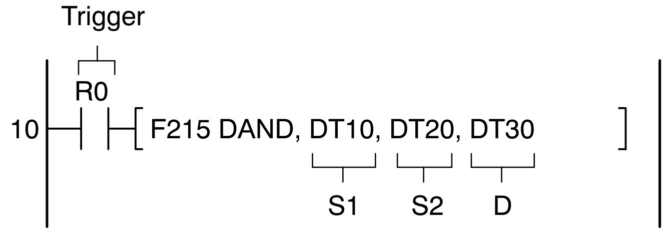
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F215 (DAND) P215 (PDAND) 32-bit data AND

Outline Performs bit-wise AND operation on two 32-bit data items.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P215 (PDAND)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST	R 0
11	F215	(DAND)
	DT	10
	DT	20
	DT	30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data
D	Lower 16-bit area of 32-bit data for storing AND operation result

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

	15															0
[DT10]	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
[DT11]	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

AND operation

	15															0
[DT20]	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
[DT21]	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

↓

	15															0
[DT30]	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
[DT31]	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1

Description

Performs AND operation on each bit in the 32-bit equivalent constant or 32-bit data specified by “S1+1 and S1” and “S2+1 and S2” when the trigger turns on. The AND operation result is stored in the 32-bit area specified by D.

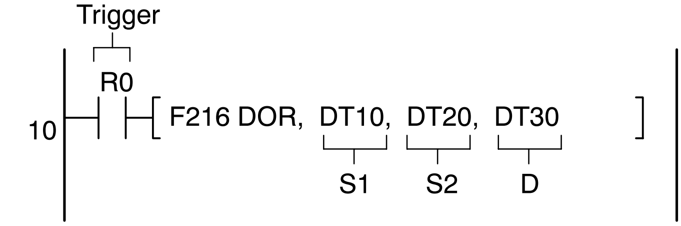
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F216 (DOR) P216 (PDOR) 32-bit data OR

Outline Performs bit-wise OR operation on two 32-bit data items.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P216 (PDOR)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F216 (DOR)
		DT 10
		DT 20
		DT 30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data
D	Lower 16-bit area of 32-bit data for storing OR operation result

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

	15															0
[DT10]	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
[DT11]	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

OR operation

	15															0
[DT20]	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
[DT21]	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

↓

	15															0
[DT30]	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1
[DT31]	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1

Description

Performs OR operation on each bit in the 32-bit equivalent constant or 32-bit data specified by “S1+1 and S1” and “S2+1 and S2” when the trigger turns on. The OR operation result is stored in the 32-bit area specified by D.

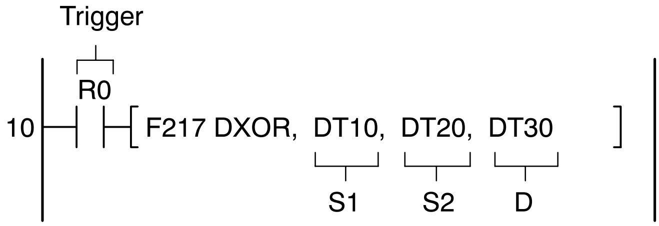
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F217 (DXOR) P217 (PDXOR) 32-bit data XOR

Outline Performs bit-wise exclusive OR operation on two 32-bit data items.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P217 (PDXOR)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F217 (DXOR) DT 10 DT 20 DT 30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data
D	Lower 16-bit area of 32-bit data for storing exclusive OR operation result

Operands

Operand	Relay WX	Relay WY	Relay WR	Relay WL	Timer/Counter SV	Timer/Counter EV	Register DT	Register LD	Register FL	Index register I	Constant K	Constant H	Constant f	Index modifier	Integer device
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

[S1]: HC6A99621

	15															0
[DT10]	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
[DT11]	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

[S2]: H00FFFF00 Exclusive OR

	15															0
[DT20]	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
[DT21]	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

[D]: H6921C656

	15															0
[DT30]	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	1
[DT31]	1	1	0	0	0	1	1	0	0	1	0	1	0	1	1	0

Description

Performs exclusive OR operation on each bit in the 32-bit equivalent constant or 32-bit data specified by “S1+1 and S1” and “S2+1 and S2” when the trigger turns on. The exclusive OR operation result is stored in the 32-bit area specified by D.

You can use this instruction to check how many bits in two 32-bit data items are the same.

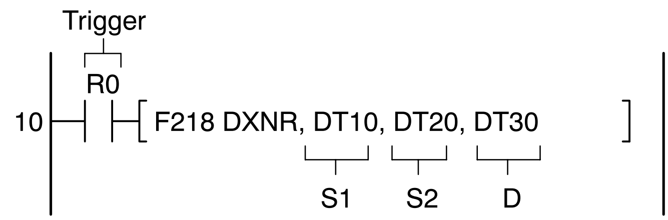
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

F218 (DXNR) P218 (PDXNR) 32-bit data XNR

Outline Performs bit-wise exclusive NOR operation on two 32-bit data items.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P218 (PDXNR)” is not available.

Program example

Ladder Diagram	Boolean: Address	Boolean: Instruction
	10	ST R 0
	11	F218 (DXNR)
		DT 10
		DT 20
		DT 30

S1	32-bit equivalent constant or lower 16-bit area of 32-bit data
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data
D	Lower 16-bit area of 32-bit data for storing exclusive NOR operation result

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

[S1]: HC6A99621

	15															0
[DT10]	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
[DT11]	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

[S2]: H00FFFF00 Exclusive NOR

	15															0
[DT20]	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
[DT21]	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

[D]: H39A996DE

	15															0
[DT30]	1	0	0	1	0	1	1	0	1	1	0	1	1	1	1	0
[DT31]	0	0	1	1	1	0	0	1	1	0	1	0	1	0	0	1

Description

Performs exclusive NOR operation on each bit in the 32-bit equivalent constant or 32-bit data specified by "S1+1 and S1" and "S2+1 and S2" when the trigger turns on. The exclusive NOR operation result is stored in the 32-bit area specified by D.

You can use this instruction to check how many bits in two 32-bit data items are the same.

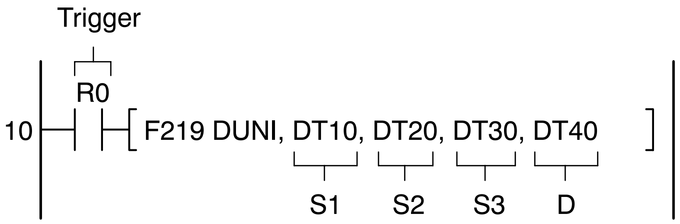
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as "0".

F219 (DUNI) P219 (PDUNI) 32-bit data unites

Outline Unites two 32-bit data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P219 (PDUNI)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F219 (DUNI)
		DT 10
		DT 20
		DT 30
		DT 40

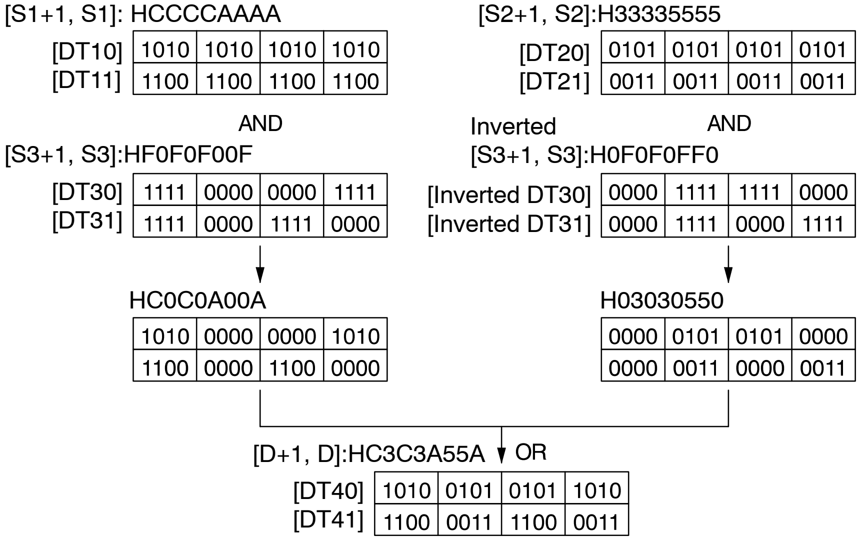
S1	32-bit equivalent constant or lower 16-bit area of 32-bit data
S2	32-bit equivalent constant or lower 16-bit area of 32-bit data
S3	Lower 16-bit area of 32-bit data which stores master data for combination or 32-bit equivalent constant
D	Lower 16-bit area of 32-bit data for storing calculated result

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example



Description

The two groups of double word data specified by “S1+1 and S1” and “S2+1 and S2” are combined by bit unit processing using the master data specified by “S3+1 and S3” and stored in the 32−bit area specified by D.

([S1+1, S1] AND [S3+1, S3]) OR ([S2+1, S2] AND [S3+1, S3]) ⟶ [D+1, D]

When [S3+1 and S3]) is H0, ([S2+1, S2] ⟶ [D+1, D]

When [S3+1, S3]) is HFFFFFFFF, ([S1+1, S1] ⟶ [D+1, D]

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- = flag (R900B): Turns on for an instant when the calculated result is recognized as “0”.

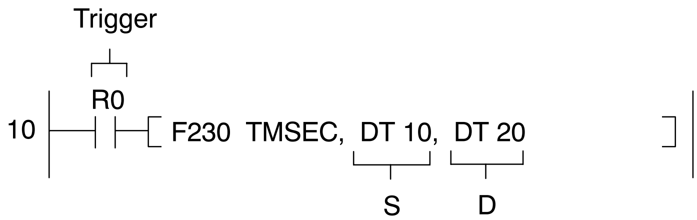
F230 (TMSEC) P230 (PTMSEC) Time data →second conversion

Availability
FP2/FP2SH/FP−X FPΣ 32k/FP0R

Outline The specified time data (a date and time) is changed into the number of seconds.

With FP2/FP2SH, this function is available from Ver. 1.50 or later.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F230 (TMSEC) DT 10 DT 20

S	Area in which the input time data stored
D	Area in which the converted second information stored (32 bits)

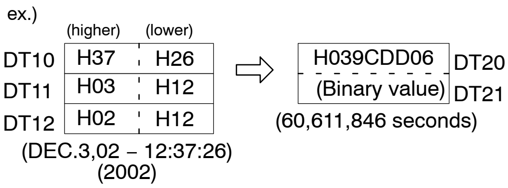
Operands

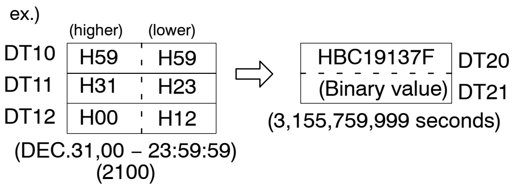
Operand	Relay				Timer/Counter		Register			Index register	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

A: Available
N/A: Not Available

Explanation of example

When the internal relay (R0) is on, conversion to the number of seconds from standard time is performed for the time data of the data registers DT10 to DT12, and the conversion result is stored in DT20 and DT21.





Description

1) Conversion to the number of seconds from standard time *1 is performed for the input time data [S ~ S+2], and a conversion result is stored in [D, D+1] by the 32−bit binary.

2) The conversion is in consideration of the leap year.

1 minute	−−−	60 seconds
1 hour	−−−	60 minutes
1 day	−−−	24 hours
1 year (leap year)	−−−	366 days
1 year (except a leap year)	−−−	365 days
A leap year (4 multiple years)	−−−	Feb.29

3) Time data (S) must be specified in the data sequence of BCD, and the value within the limits must be registered.

*1: Standard time is 00:00'00" on January 1, '01.
Moreover, a conversion result is output with a binary value.

Time data (S) (BCD)

	(higher)	(lower)
S:	Minutes (H00−59)	Seconds (H00−H59)
S+1:	Days (H01−H31)	Hours (H00−H23)
S+2:	Years (H00−H99)	Months (H01−H12)

⇩

Second data (D)

D:	Seconds data
D+1:	(H00000000 − HBC19137F)

The correspondence table of Time data and Second conversion

	Time data (S)	Seconds data (D)
2001	'01/01/01 00:00:00	H00000000
:	'01/01/01 00:00:01	H00000001
:	:	:
:	'01/01/01 00:01:00	H0000003C
:	:	:
:	'01/01/01 01:00:00	H00000E10
:	:	:
:	'01/01/02 00:00:00	H00015180
:	:	:
2099	'99/12/31 23:59:59	HBA368E7F
2100	'00/01/01 00:00:00	HBA368E80
:	:	:
2100	'00/12/31 23:59:59	HBC19137F

Flag conditions

ΣError flag (R9007) (R9008):

– It turns on, when the specified address using the index modifier exceeds a limit.
– It turns on, when values other than BCD are specified for [S].
– It turns on, when the value which exceeds the range in the time data of [S] is specified.
– It turns on, when the data of [S] exceeds the area.

F231 (SECTM) P231 (PSECTM) Second → time data conversion

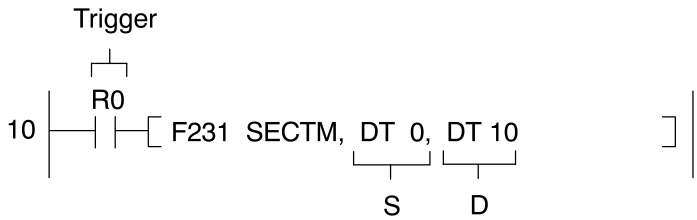
Availability
FP2/FP2SH/FP-X FPΣ 32k/FP0R

Outline The specified number of seconds is changed into time data (a date and time).

With FP2/FP2SH, this function is available from Ver. 1.50 or later.

Program example

Ladder Diagram



Boolean

Address	Instruction
10	ST R 10
11	F231 (SECTM)
	DT 0
	DT 10

S	Area in which the number of seconds stored (32 bits)
D	Head area in which time data stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

Explanation of example

When the internal relay (R0) is on, the number of seconds for the data registers DT0 and DT1 is converted to the time data based on the standard time, and stored in DT10 to 12.

ex.)

DT0, DT1: H039D0A6A (Binary value) (60,623,466秒) ⇨

	(higher)	(lower)
DT10	H51	H06
DT11	H03	H15
DT12	H02	H12

(Dec.3,02 – 15:51:06)
(2002)

ex.)

DT0, DT1: HBC19137F (Binary value) (3,155,759,999 seconds) ⇨

	(higher)	(lower)
DT10	H59	H59
DT11	H31	H23
DT12	H00	H12

(Dec.31,00 – 23:59:59)
(2100)

Description

1) The input number of seconds (S) is converted to the time data based on standard time *1, and stored in (D).

2) The conversion is in consideration of the leap year.

1 minute	−−−	60 seconds
1 hour	−−−	60 minutes
1 day	−−−	24 hours
1 year (leap year)	−−−	366 days
1 year (except a leap year)	−−−	365 days
A leap year (4 multiple years)	−−−	Feb.29

3) The range which can specify the number of seconds (S) is 100 years which can be expressed by time data.

H00000000 − HBC19137F	−−−	Normal conversion
HBC191380 − HFFFFFFFF	−−−	Conversion error

*1: Standard time is 00:00'00" on January 1, '01.

Second data (S)

S:	Seconds data	(lower)
S+1:	(H00000000 − HBC19137F)	(higher)

⇩

Time data (D) (BCD)

	(higher)	(lower)
D:	Minutes (H00−H59)	Seconds (H00−H59)
D+1:	Days (H01−H31)	Hours (H00−H23)
D+2:	Years (H00−H99)	Months (H01−H12)

Total Second Conversion

Second data (S)	Time data (D)	
H00000000	'01/01/01 00:00:00	2001
H00000001	'01/01/01 00:00:01	:
:	:	:
H0000003C	'01/01/01 00:01:00	:
:	:	:
H00000E10	'01/01/01 01:00:00	:
:	:	:
H00015180	'01/01/02 00:00:00	:
:	:	:
HBA368E7F	'99/12/31 23:59:59	2099
HBA368E80	'00/01/01 00:00:00	:
:	:	:
HBC19137F	'00/12/31 23:59:59	2100

Flag conditions

ΣError flag (R9007) (R9008):

– It turns on, when the specified address using the index modifier exceeds a limit.

– It turns on, when the number of seconds (S) is (S) >=HBC191380.

– It turns on, when the data memory of [D] exceeds the area.

F235 (GRY) P235 (PGRY)

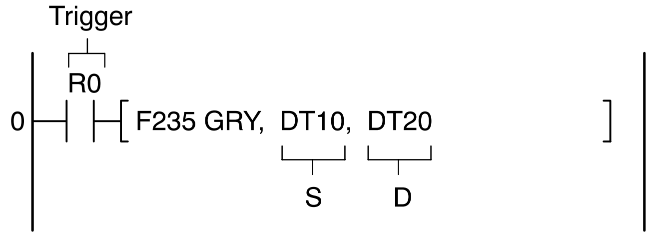
16-bit data → Gray code

Outline

Converts 16-bit data to gray code.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P235 (PGRY)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
0	ST	R	0
1	F235		(GRY)
	DT		10
	DT		20

S	16-bit equivalent constant or 16-bit area to be converted (source)
D	16-bit area for storing gray codes (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

Converts the 16-bit data specified by S to gray codes when the trigger turns on. The converted result is stored in the 16-bit area specified by D.

For detailed information about the gray code ☛ page 3 − 597

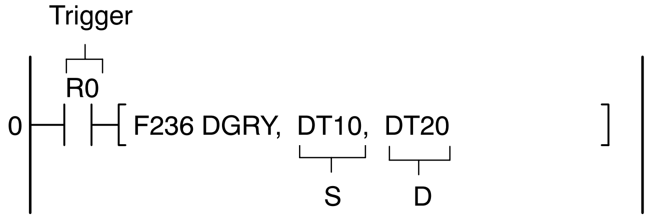
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F236 (DGRY) P236 (PDGRY) 32-bit data → Gray code

Outline Converts 32-bit binary data to gray code.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P236 (PDGRY)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST R 0
	1	F236 (DGRY)
		DT 10
		DT 20

S	32-bit equivalent constant or lower 16-bit area of 32-bit data to be converted (source)
D	Lower 16-bit area of 32-bit data for storing gray code (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

Converts the 32-bit data specified by S to gray code when the trigger turns on. The converted data is stored in D+1 and D.

For detailed information about the gray code ☛ page 3 − 597

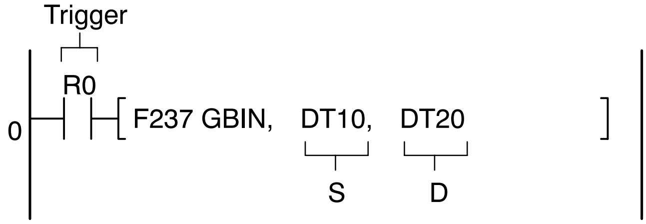
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F237 (GBIN) P237 (PGBIN) 16-bit Gray code → 16-bit binary data

Outline Converts 16-bit gray code to 16-bit binary data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P237 (PGBIN)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	0 1	ST R 0 F237 (GBIN) DT 10 DT 20

S	16-bit area for gray code (source)
D	16-bit area for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

Converts the gray codes in 16-bit are specified by S to 16-bit data when the trigger turns on. The converted result is stored in the area specified by D.

For detailed information about the gray code ☛ page 3 – 597

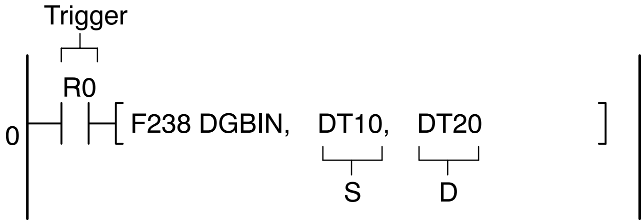
Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

F238 (DGBIN) P238 (PDGBIN) 32-bit Gray code → 32-bit binary data

Outline Converts gray code to 32-bit data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P238 (PDGBIN)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	0	ST R 0
	1	F238 (DGBIN)
		DT 10
		DT 20

S	Lower 16-bit area of 32-bit data for gray code (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

Converts gray code to 32-bit data when the trigger turns on. The converted result is stored in the 32-bit area specified by D+1 and D.

For detailed information about the gray code ☛ page 3 – 597

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

Binary/Hexadecimal/BCD/Gray Code Expressions

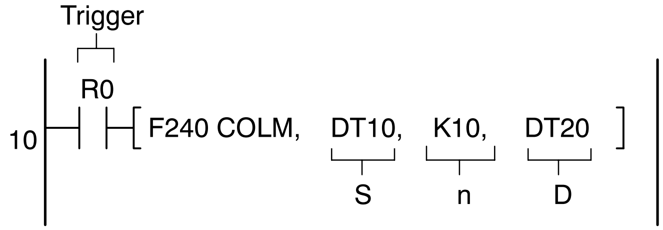
Decimal	Binary data	Gray code
0	0000 0000 0000 0000	0000 0000 0000 0000
1	0000 0000 0000 0001	0000 0000 0000 0001
2	0000 0000 0000 0010	0000 0000 0000 0011
3	0000 0000 0000 0011	0000 0000 0000 0010
4	0000 0000 0000 0100	0000 0000 0000 0110
5	0000 0000 0000 0101	0000 0000 0000 0111
6	0000 0000 0000 0110	0000 0000 0000 0101
7	0000 0000 0000 0111	0000 0000 0000 0100
8	0000 0000 0000 1000	0000 0000 0000 1100
9	0000 0000 0000 1001	0000 0000 0000 1101
10	0000 0000 0000 1010	0000 0000 0000 1111
11	0000 0000 0000 1011	0000 0000 0000 1110
12	0000 0000 0000 1100	0000 0000 0000 1010
13	0000 0000 0000 1101	0000 0000 0000 1011
14	0000 0000 0000 1110	0000 0000 0000 1001
15	0000 0000 0000 1111	0000 0000 0000 1000
16	0000 0000 0001 0000	0000 0000 0001 1000
17	0000 0000 0001 0001	0000 0000 0001 1001
18	0000 0000 0001 0010	0000 0000 0001 1011
19	0000 0000 0001 0011	0000 0000 0001 1010
20	0000 0000 0001 0100	0000 0000 0001 1110
21	0000 0000 0001 0101	0000 0000 0001 1111
22	0000 0000 0001 0110	0000 0000 0001 1101
23	0000 0000 0001 0111	0000 0000 0001 1100
24	0000 0000 0001 1000	0000 0000 0001 0100
25	0000 0000 0001 1001	0000 0000 0001 0101
26	0000 0000 0001 1010	0000 0000 0001 0111
27	0000 0000 0001 1011	0000 0000 0001 0110
28	0000 0000 0001 1100	0000 0000 0001 0010
29	0000 0000 0001 1101	0000 0000 0001 0011
30	0000 0000 0001 1110	0000 0000 0001 0001
31	0000 0000 0001 1111	0000 0000 0001 0000
·	▵	▵
·	▵	▵
·	▵	▵
63	0000 0000 0011 1111	0000 0000 0010 0000
·	▵	▵
·	▵	▵
·	▵	▵
255	0000 0000 1111 1111	0000 0000 1000 0000

F240 (COLM) P240 (PCOLM) Bit line to bit column conversion

Outline Converts a selected bit line to a bit column.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P240 (PCOLM)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F240	(COLM)	
	DT		10
	K		10
	DT		20

S	16-bit equivalent constant or 16-bit area (source)
n	16-bit equivalent constant or 16-bit area to specify bit position
D	Starting address of area which will be rewritten with bit column.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the specified bit position n = 10

	15															0
S	0	1	0	1	0	0	0	1	1	1	0	1	1	0	0	1

	15					10										0
D						1										
D+1						0										
D+2						0										
D+3						1										
D+4						1										
D+5						0										
D+6						1										
D+7						1										
D+8						1										
D+9						0										
D+10						0										
D+11						0										
D+12						1										
D+13						0										
D+14						1										
D+15						0										

Description

The bit data at the position specified by "n" of the 16-word data area with the head address D is rewritten using the 16-bit data of the area specified by S.

The contents of the bits of the 16-word data area with head address D that are not specified do not change.

"n" can be between 0 and 15.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If the specified bit position [n] is not in the range $0 \leqq n \leqq 15$.
  - If the result of the conversion overflows the storage area specified with D.

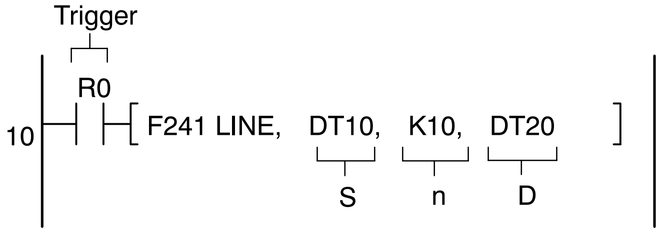
F241 (LINE) P241 (PLINE)

Bit column to bit line conversion

Outline Converts a specified bit column to a bit line.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P241 (PLINE)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F241		(LINE)
	DT		10
	K		10
	DT		20

S	Starting address of area where bit column will be read.
n	16-bit equivalent constant or 16-bit area to specify bit position
D	16-bit area for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the specified bit position n = 10

	15					10										0
S						1										
S+1						0										
S+2						0										
S+3						1										
S+4						1										
S+5						0										
S+6						1										
S+7						1										
S+8						1										
S+9						0										
S+10						0										
S+11						0										
S+12						1										
S+13						0										
S+14						1										
S+15						0										

↓

	15															0
D	0	1	0	1	0	0	0	1	1	1	0	1	1	0	0	1

Description

Reads the bit data at the position specified by “n” from the area specified by S and stores it in the area specified by D.

“n” can be set between 0 and 15.

Flag conditions

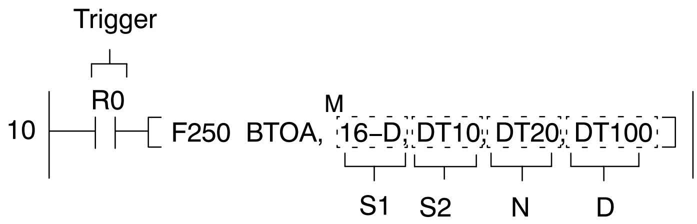
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If the specified bit position [n] is not in the range 0 ≦ n ≦ 15.
  - If the conversion range specified with S overflows the area.

Availability
FP−X/FPΣ 32k/FP0R

F250 (BTOA) Binary →ASCII conversion

Outline Converts 16−bit/32−bit binary data to ASCII code.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10	ST R 0
	11	F250(BTOA)
		M 16−D
		DT 10
		DT 20
		DT 100

S1	Control string
S2	Starting 16−bit area for storing binary data
N	Conversion method
D	Starting 16−bit area for storing ASCII codes of converted result

Operands

Operand	Relay				Timer/Counter		Register		Index register		Constant		M	Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	IX	IY	K	H		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
N	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

Operation

Converts the binary data stored in the area specified by S2 to ASCII codes using the conversion method of N according to 4 control characters specified by S1. The converted result is stored in the area specified by D.

Specifying the various items

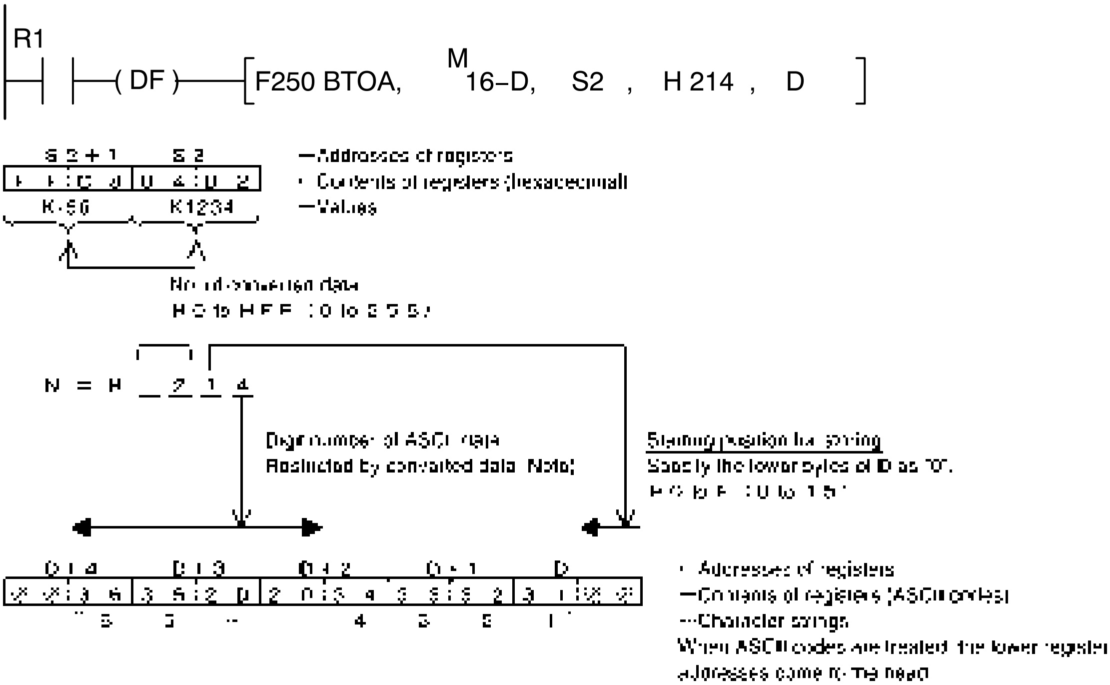
• Specifying control strings and the meanings [S1]

M 16−D Converts 16−bit data to decimal ASCII codes.
M 32−D Converts 32−bit data to decimal ASCII codes.
M 16+H Converts 16−bit data to hexadecimal ASCII codes. (Normal direction)
M 32+H Converts 32−bit data to hexadecimal ASCII codes. (Normal direction)
M 16−H Converts 16−bit data to hexadecimal ASCII codes. (Reverse direction)
M 32−H Converts 32−bit data to hexadecimal ASCII codes. (Reverse direction)

*The details of the normal and reverse directions are described later.

• Specifying the conversion method [N]

Example of converting 16−bit data (K1234 and K56) to decimal ASCII codes

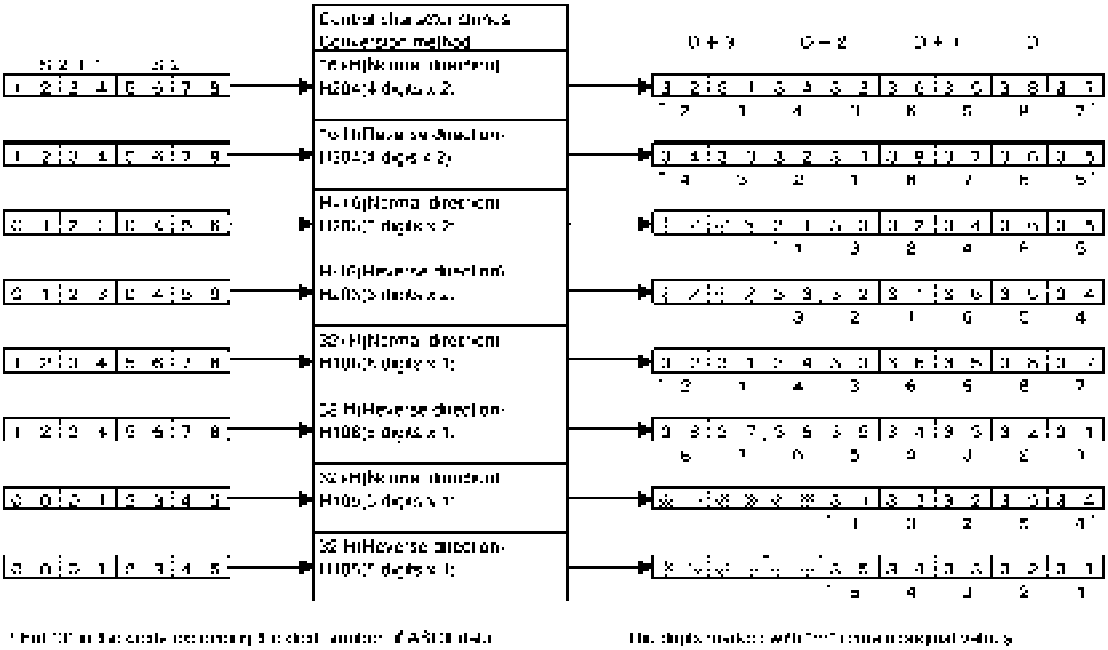


Notes

About the digit number of ASCII data

- **When converting 16−bit data to hexadecimal ASCII codes**
  Specified range: H1 to 4
  When less than H4, the specified number of digits is stored from the lower bytes.
  If the digit number of original data is larger with the specification less than H4, it is an error.
- **When converting 32−bit data to hexadecimal ASCII codes**
  Specified range: H1 to 8
  When less than H8, the specified number of digits is stored from the lower bytes.
  If the digit number of original data is larger with the specification less than H8, it is an error.
- **When converting to decimal ASCII codes**
  Specified range: H1 to F
  Source data is treated as signed binary data. When it is a negative number, the minus sign “−” is added.
  When the number of digit of ASCII codes is larger than the converted result, the space “_” is stored in the extra smaller addresses.

About normal direction and reverse direction (only when converting to hexadecimal ASCII data)



Conversion examples

- Converts 16−bit data (K1234 and K56) to decimal ASCII codes.

DT10 = K 1234 DT11 = K 56	→ ”1234__56”

When No. of converted data is “2”, Starting position for storing is “0”, Size of the area for storing is “4”.

```
R0
|—| |——( DF )——[F250 BTOA,  M 16−D,   DT10 , H 204 , DT 100 ]
```

DT11	DT10
0 0 ¦ 3 8	0 4 ¦ D 2
K56	K1234

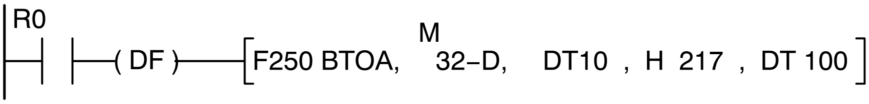
⇩

DT103	DT102	DT101	DT100
3 6 ¦ 3 5	2 0 ¦ 2 0	3 4 ¦ 3 3	3 2 ¦ 3 1
” 6 5	_ _	4 3	2 1”

• Converts 32−bit data (K1234 and K56789) to decimal ASCII codes.

DT10、11	=	K	1234	→ ”___1234__56789”
DT12、13	=	K	56789st	

When No. of converted data is “2”, Starting position for storing is “1”, Size of the area for storing is “7”.



DT13	DT12	DT11	DT10
0 0 ¦ 0 0	D D ¦ D 5	0 0 ¦ 0 0	0 4 ¦ D 2
K56789		K1234	

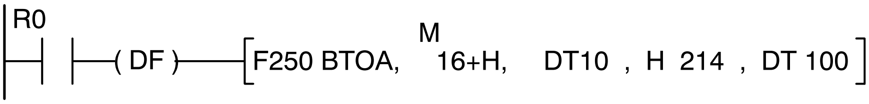
⇩

DT107	DT106	DT105	DT104	DT103	DT102	DT101	DT100
¦ 3 9	3 8 ¦ 3 7	3 6 ¦ 3 5	2 0 ¦ 2 0	3 4 ¦ 3 3	3 2 ¦ 2 1	2 0 ¦ 2 0	2 0 ¦
” 9	8 7	6 5	_ _	4 3	2 1	_ _	_ ”

• Converts 16−bit data (H0123 and H89AB) to hexadecimal ASCII codes.

DT10	=	H	123	→ ”2301AB89”
DT11	=	H	89AB	

When No. of converted data is “2”, Starting position for storing is “1”, Size of the area for storing is “4”. (Normal direction)



DT11	DT10
8 9 ¦ A B	0 1 ¦ 2 3

⇩

DT104	DT103	DT102	DT101	DT100
¦ 3 9	3 8 ¦ 4 2	4 1 ¦ 3 1	3 0 ¦ 3 3	3 2 ¦
’ 9	8 B	A 1	0 3	2 ’

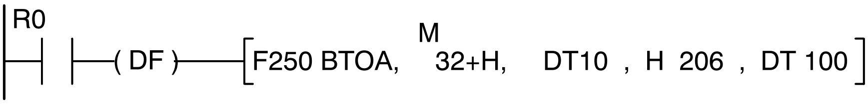
For the reverse direction (when ”16+H” is ”16−H)

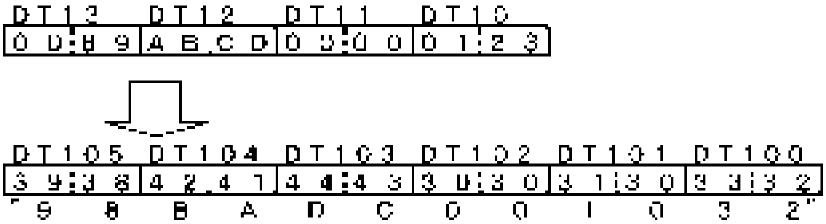
DT104	DT103	DT102	DT101	DT100
¦ 4 2	4 1 ¦ 3 9	3 8 ¦ 3 3	3 2 ¦ 3 1	3 0 ¦
’ B	A 9	8 3	2 1	0 ’

- Converts 32−bit data (H00000123 and H0089ABCD) to hexadecimal ASCII codes (Normal direction)

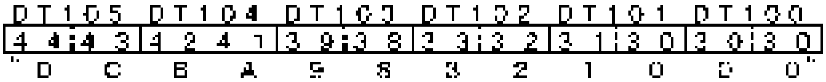
DT10、11	=	H	123	→ ”230100CDAB89”
DT12、13	=	H	89ABCD	

When No. of converted data is ”2”, Starting position for storing is ”0”, Size of the area for storing is ”6”.





For the reverse direction (when ”32+H” is ”32−H”)



Flag conditions

ΣError flag (R9007): Turns on and stays on when

Σ Error flag (R9008): Turns on for an instant when

− There is an error in the control string specified by S1.

− The direction of converted data is changed to the normal direction when the conversion format specified by S1 is in decimal.

− The size of the area for storing ASCII codes specified by N exceeds the rated value when the conversion format specified by S1 is in hexadecimal. (Rated value for 16−bit data: 4) (Rated value for 32−bit data: 8)

− The No. of the converted data specified by N is 0.

− The converted result exceeds the area for storing ASCII codes specified by N.

− The converted result exceeds the area.

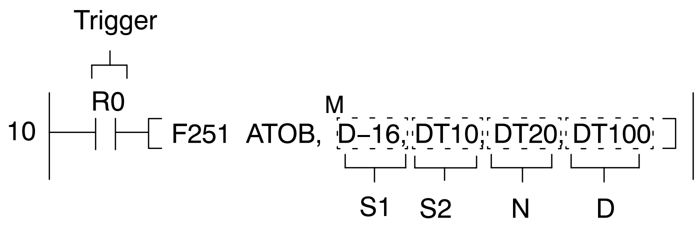
− The area specified using the index modifier exceeds the limit.

Availability
FP−X/FPΣ 32k/FP0R

F251 (ATOB) ASCII →Binary conversion

Outline Converts ASCII code to 16−bit/32−bit binary data.

Program example



Boolean	
Address	**Instruction**
10	ST R 0
11	F251 (ATOB)
	M D−16
	DT 10
	DT 20
	DT 100

S1	Control string
S2	Starting 16−bit area for storing ASCII codes
N	Conversion method
D	Starting 16−bit area for storing binary data of converted result

Operands

Operand	Relay				Timer/Counter		Register		Index register		Constant		M	Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	IX	IY	K	H		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
N	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

A: Available
N/A: Not Available

Operation

Converts the ASCII codes stored in the area specified by S2 to binary data using the conversion method of N according to 4 control characters specified by S1. The converted result is stored in the area specified by D.

Specifying the various items

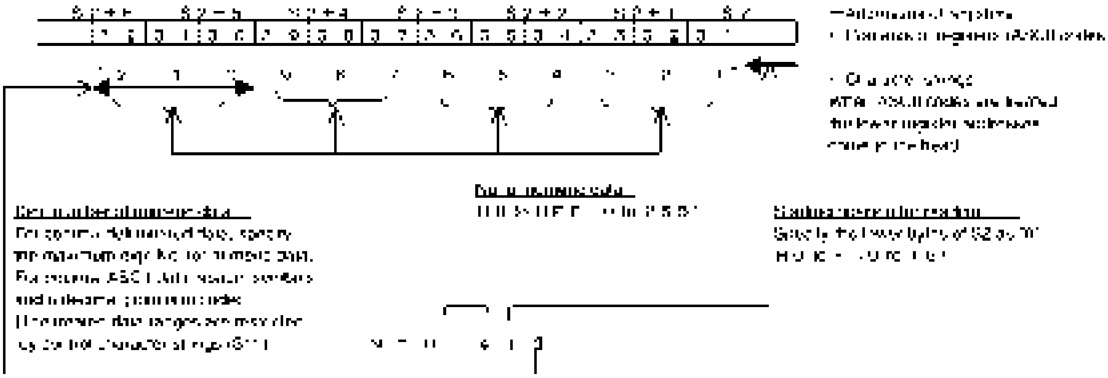
• Specifying control strings and the meanings [S1]

		Ranges of treated data
M D−16	Converts decimal ASCII codes to 16−bit data.	−32,768 to +32767
M D−32	Converts decimal ASCII codes to 32−bit data.	−2,147,483,648 to +2,147,483,647
M H+16	Converts hexadecimal ASCII codes to 16−bit data. (Normal direction)	0 to FFFF
M H+32	Converts hexadecimal ASCII codes to 32−bit data. (Normal direction)	0 to FFFFFFFF
M H−16	Converts hexadecimal ASCII codes to 16−bit data. (Reverse direction)	0 to FFFF
M H−32	Converts hexadecimal ASCII codes to 32−bit data. (Reverse direction)	0 to FFFFFFFF

*The details of the normal and reverse directions are described later.

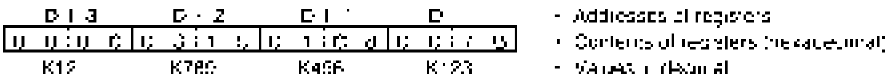
• Specifying the conversion method [N]

Example of converting the ASCII data string ”123456789012” to decimal 3 digits x 4 data



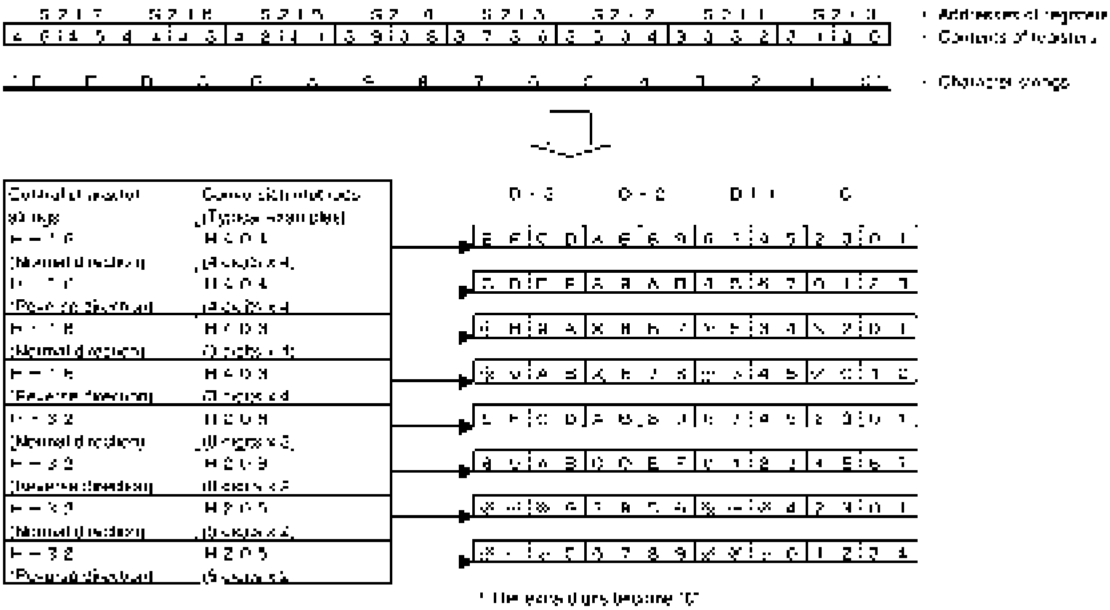
R0 (DF) [F251 ATOB, MD−16, S2 , H 413 , D]

When converting by the above program:



• About normal direction and reverse direction

The conversions in the normal direction and reverse direction are available for hexadecimal ASCII data.
Example of converting ”0123456789ABCDEF”.



Conversion examples

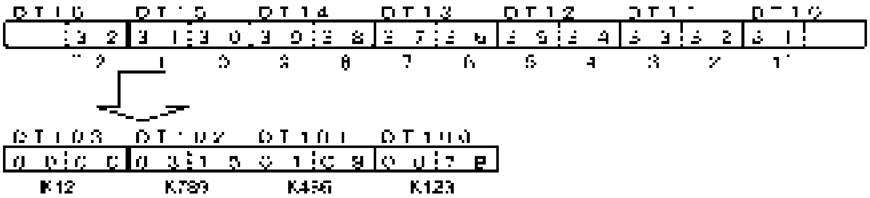
• Examples of converting to decimal 3 digits x 4 data (when no comma ”,” exists)

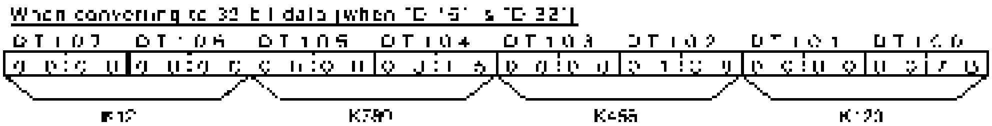
Converts to 16−bit data.

”123456789012”	→ DT100 = K 123 DT100 = K 456 DT102 = K 789 DT103 = K 12

When No. of numeric data is “4”, Starting position for reading is “1”, Digit No. of numeric data is “3”.

```
R0
|—| |——( DF )——[F251 ATOB,  M D−16,   DT10 ,  H 413 ,  DT 100 ]
```





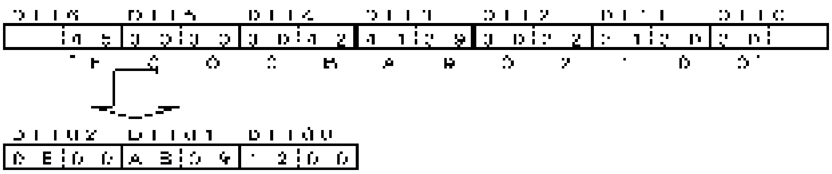
• Examples of converting to hexadecimal 4 digits x 3 data

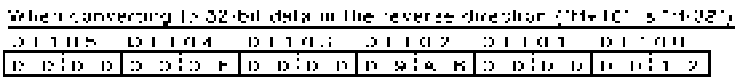
Converts to 16−bit data in normal direction.

”001209AB000E”	→ DT100 = H 1200 DT101 = H AB09 DT102 = H 0E00

When No. of numeric data is “3”, Starting position for reading is “1”, Digit No. of numeric data is “4”.

```
R0
|—| |——( DF )——[F251 ATOB,  M H+16,   DT10 ,  H 314 ,  DT 100 ]
```

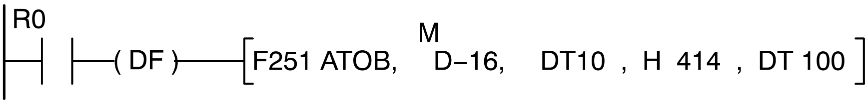




- Example of converting to decimal number x 4 data (in case of comma−deliminated “,” data)

”12,345,6789,0,” * The last of character strings is a comma.	→ DT100 = K 12 DT101 = K 345 DT102 = K 6789 DT103 = K 0

When No. of numeric data is “4”, Starting position for reading is “1”, Digit No. of numeric data is “4”.
(Converts to 16−bit data) * Specify the maximum digit number.



DT17		DT16		DT15		DT14		DT13		DT12		DT11		DT10	
	2 C	3 0	2 C	3 1	3 0	3 9	3 8	2 C	3 5	3 4	3 3	2 C	3 2	3 1	
	”,	0	,	9	8	7	6	,	5	4	3	,	2	1”	

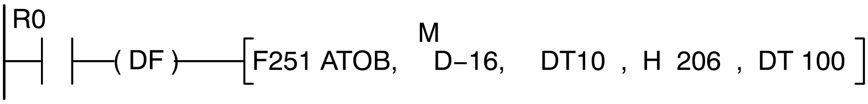
⇩

DT103		DT102		DT101		DT100	
0 0	0 0	1 A	8 5	0 1	5 9	0 0	0 C
K0		K6789		K345		K12	

- Example of converting to decimal 5 digits with a decimal point x 2 data (when no comma exists)

”1234.50006.7”	→ DT100 = K 12345 DT101 = K 67

When No. of numeric data is “2”, Starting position for reading is “0”, Digit No. of numeric data is “6”, and converting to 16−bit data. *A decimal point is also counted as a digit.



DT15		DT14		DT13		DT12		DT11		DT10	
3 7	2 E	3 6	3 0	3 0	3 0	3 5	2 E	3 4	3 3	3 2	3 1
” 7	.	6	0	0	0	5	.	4	3	2	1 ”

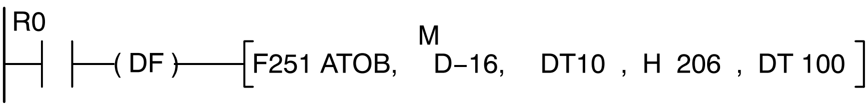
⇩

DT101		DT100	
0 0	4 3	3 0	3 9
K67		K12345	

- Example of converting to decimal number with a decimal point x 2 data (in case of comma−deliminated “,” data)

”1234.5,6.7,” * The last of character strings is a comma.	→ DT100 = K 12345 DT101 = K 67

When No. of numeric data is “2”, Starting position for reading is “0”, Digit No. of numeric data is “6”, and converting to 16−bit data. *A decimal point is also counted as a digit.



DT15		DT14		DT13		DT12		DT11		DT10	
	2 C	3 7	2 E	3 6	2 C	3 5	2 E	3 4	3 3	3 2	3 1
	,	7	.	6	,	5	.	4	3	2	1”

⇩

DT101		DT100	
0 0	4 3	3 0	3 9
K67		K12345	

Particular examples

- If there is numeric data larger than the specified digit number between commas.
  (Example: Decimal number x 4, the digit number of the numeric data is 4)

”1234,567890,12,345” → K 1234
K 5678
K 90 → The overflowed numbers become one numeric data.
K 12
~~K 345~~ → It is ignored.

- If there is no value between commas (Example: Decimal number x 4)

”123,456,,78” → Operation error

- If there is only a decimal point between commas (Example: Decimal number with a decimal point x 3)

”1234.5,.,6.7” → Operation error
*If any number exists like “2.” or “.2”, the data will be converted.

Flag conditions

ΣError flag (R9007): Turns on and stays on when

ΣError flag (R9008): Turns on for an instant when

- There is an error in the control string specified by S1.
- The direction of converted data is changed to the normal direction when the conversion format specified by S1 is in decimal.
- The size of the area for storing ASCII codes specified by N exceeds the rated value when the conversion format specified by S1 is in hexadecimal. (Rated value for 16−bit data: 4) (Rated value for 32−bit data: 8)
- Any code other than 0 to F, symbols, space, dot, comma exists in ASCII code specified by S2.
- The No. of the converted blocks specified by N is 0.
- The size of the area for storing ASCII codes specified by N is 0.
- The ASCII code to be converted exceeds the area.
- The converted result exceeds the area.
- The converted result exceeds the converted data scale specified by N.
- The area specified using the index modifier exceeds the limit.

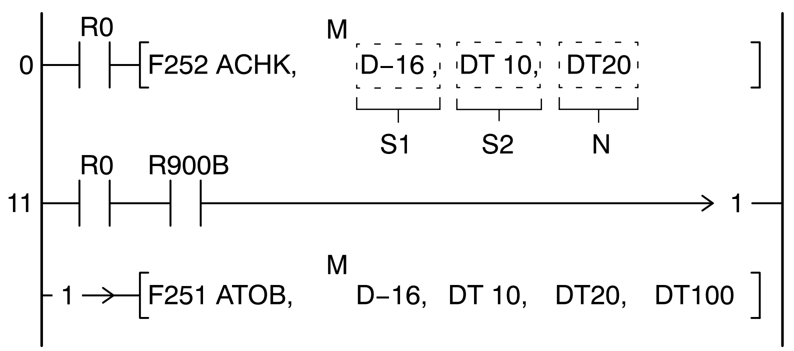
Availability
FP−X (V2.00 or more) FPΣ 32k/FP0R

F252 (ACHK) ASCII data check

Outline Checks whether the specified ASCII data is correct or not.

Program example

Ladder Diagram



Boolean		
Address	Instruction	
0	ST	R 0
1	F252(ACHK)	
	M	D−16
	DT	10
	DT	20
11	ST	R 0
12	AN	R 900B
14	F251(ATOB)	
	M	D−16
	DT	10
	DT	20
	DT	100

S1	Control string data or 16−bit are for storing control strings
S2	Starting 16−bit area for storing ASCII code
N	16−bit equivalent constant or 16−bit area for storing conversion method

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		M	Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H		
S1	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
N	A	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A

(*1) I0 to ID

A: Available
N/A: Not Available

Operation

- Checks whether the ASCII codes stored in the area specified by S2 can be converted correctly or not using the conversion method of N according to 4 control characters specified by S1.
  Checks whether the character strings can be converted by F251 (ATOB) instruction.
- Checks data before converting the data by F251 (ATOB) instruction.
  If an error is found in the data, controls not to execute F251 (ATOB) instruction.
  Specify the same values for S1, S2 and N as F251 (ATOB) instruction.
  If the results are correct, the special internal relay (R900B) turns on.
  If the results are incorrect, the special internal relay (R900B) turns off.

Specifying the various items

The way to specify S1, S2 and N is the same as F251 (ATOB) instruction. Refer to the explanation described in F251 (ATOB) ASCII Binary conversion.

Flag conditions

ΣError flag (R9007): Turns on and stays on when

Σ Error flag (R9008): Turns on for an instant when

- There is an error in the control string specified by S1.
- The direction of converted data is changed to the normal direction when the conversion format specified by S1 is in decimal.
- The size of the area for storing ASCII codes specified by N exceeds the rated value when the conversion format specified by S1 is in hexadecimal.
  (Rated value for 16−bit data: 4)
  (Rated value for 32−bit data: 8)
- The No. of the converted blocks specified by N is 0.
- The size of the area for storing ASCII codes specified by N is 0.
- The ASCII code to be converted exceeds the area.
- The area specified using the index modifier exceeds the limit.

Overview of Character String Instructions F257 (SCMP) to F265 (SREP)

Configuration of character string instruction data tables

Data tables for character strings show the character string size, the number of characters, and the character data.

	15	0	
S	Character string size		← Max. number of characters that can be stored
S+1	No. of characters		← Number of characters stored in the character string
S+2	Character 2 (Byte 1)	Character 1 (Byte 0)	Character data
S+3	Character 4 (Byte 3)	Character 3 (Byte 2)	
S+4	Character 6 (Byte 5)	Character 5 (Byte 4)	
.	.	.	
.	.	.	
	Higher 16 bits	Lower 16 bits	

Example:

The example shows a character string data table specifying the following: Character string size: 10. Number of characters: 5. Character data: “ABCDE”.

DT0	10	
DT1	5	
DT2	“B”	“A”
DT3	“D”	“C”
DT4		“E”
DT5		
DT6		
	Higher 16 bits	Lower 16 bits

How data tables are set

Specify the values for the character string size and number of characters. The **F0 (MV)** instruction is used to specify values.

Specify the characters. The **F95 (ASC)** instruction is used to specify characters.

Example:

The example shows (character string size “16 characters”, “no specification of characters”) for DT0.

```
   R0
|--| |--[F0(MV) ,  K16,   DT0          ]  |
                    ↑
                    └─ Character string size
```

Example:

The example shows a data table specifying the following for DT0: (character string size “20 characters”, number of characters “12 characters”, and character data “ABCDEFGHIJKL”).

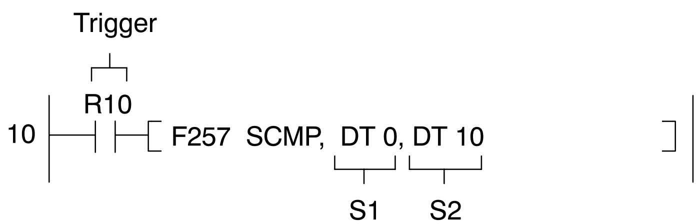
```
   R0
|--| |--[F0(MV),   K20,   DT0            ]  |
                    ↑
                    └─ Character string size
                       (20 characters)
        [F0(MV),   K12,   DT1            ]  |
                    ↑
                    └─ Number of characters
                       (12 characters)
        [F95 ASC,  M ABCDEFGHIJKL,  DT2  ]  |
                     Character data
```

F257 (SCMP) P257 (PSCMP) Comparing character strings

Outline These instructions compare two specified character strings and output the judgment results to a special internal relay.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P257 (PSCMP) cannot be specified.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 10 F257 (SCMP) DT 0 DT 10

S1	Character string 1 for comparison
S2	Character string 2 for comparison

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) Cannot be specified with FP0R, FPΣ and FP−X.
(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

When internal relay R10 is on, data register DT1 and DT11 are compared.
In this case, it is determined that “S1” < “S2”, and R900C goes on.

	Higher 16 bits	Lower 16 bits
DT0	10 (Character string size)	
DT1	4 (Number of characters)	
DT2	“B” (Byte 1)	“A” (Byte 0)
DT3	“D” (Byte 3)	“C” (Byte 2)
DT4	(Byte 5)	(Byte 4)
DT5	(Byte 7)	(Byte 6)
DT6	(Byte 9)	(Byte 8)

↔

	Higher 16 bits	Lower 16 bits
DT10	8 (Character string size)	
DT11	5 (Number of characters)	
DT12	“B” (Byte 1)	“A” (Byte 0)
DT13	“D” (Byte 3)	“C” (Byte 2)
DT14	(Byte 5)	“E” (Byte 4)
DT15	(Byte 7)	(Byte 6)

Description

The character string specified for "S1" is compared to that specified for S2, and the judgment result is output to special internal relays R9009 to R900C (judgment flags for comparison instructions).

R9009 to R900C are assigned based on whether "S1" or "S2" is larger, as shown in the table below.

Relationship of S1 and S2	**Flag**			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
S1<S2	OFF	OFF	ON	Fluctuates
S1=S2	OFF	ON	OFF	OFF
S1>S2	ON	OFF	OFF	Fluctuates

Precautions during programming

If the number of characters is different, the greater/lesser relationship is as shown below.

S1	**Greater/lesser**	**S2**
"ABCDE"	=	"ABCDE"
"ABCD"	<	"ABCDE"
"B"	>	"ABCDE"

Comparison of character strings is performed in sequence from byte 0, one character at a time.

If one character string has fewer characters than the other, it may still be handled as larger if a large character code is used when the comparison is made.

Example: "B" > "ABCDE"

To specify a character string, indicate the number of the area in which the character size and number of characters have been specified.

For detailed information about the table configuration of data area ☛ see page 3 – 615.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The specified range is exceeded when an index is modified.
  - – The number of characters is larger than the character string size.

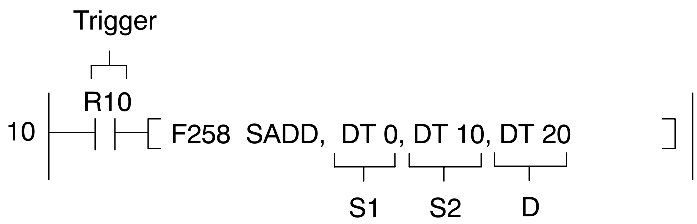
F258 (SADD) P258 (PSADD) Character string coupling

Outline

These instructions couple one character string with another.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P258 (PSADD) cannot be specified.

Program example



Address	Instruction		
10	ST	R	10
11	F258		(SADD)
	DT		0
	DT		10
	DT		20

S1	Character string to be coupled
S2	Character string to be coupled
D	Area in which the coupled character strings are stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

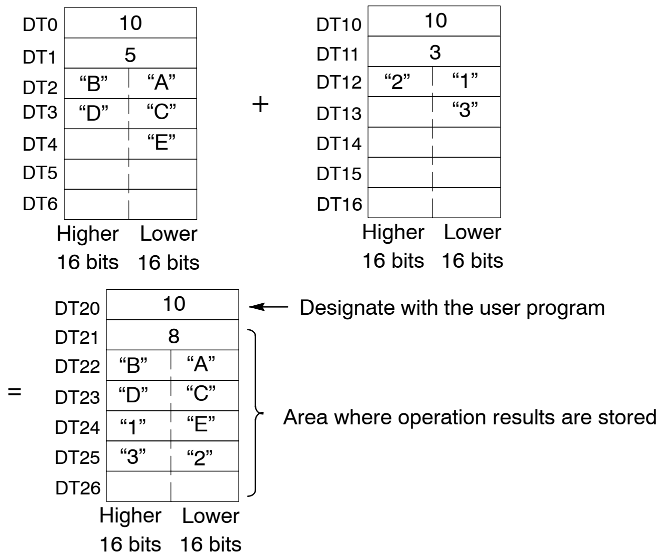
A: Available

N/A: Not Available

Description

The character string specified for “S1” is coupled to that specified for “S2”, and the result is stored in the character string specified by “D”.
At the starting address of the area for storing results “D”, designate the character string size using the user program.

Explanation of example



Precautions during programming

If the result of the coupling operation is larger than the character string size of “D”, only as many characters as will fit in “D” are stored.

Flag conditions

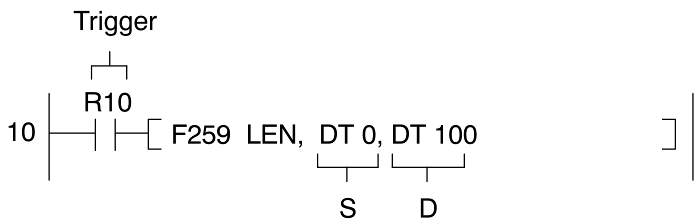
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The specified range is exceeded when an index is modified.
  - – The number of characters is larger than the character string size.
- Carry flag (R9009): Turns on for an instant when the operation result is larger than the character string size of “D”

F259 (LEN) P259 (PLEN) Number of characters in a character string

Outline These instructions determine the number of characters in a character string.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P259 (PLEN) cannot be specified.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 10 F259 (LEN) DT 0 DT 100

S	Character string
D	Area in which the coupled character strings are stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
D	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available

N/A: Not Available

Explanation of example

DT0	10	
DT1	8	
DT2	“B”	“A”
DT3	“D”	“C”
DT4	“1”	“E”
DT5	“3”	“2”
DT6		
	Higher 16 bits	Lower 16 bits

= DT100 | 8 |

Description

The number of characters in the character string specified by “S” is determined, and the result is stored in “D”.

Precautions during programming

If the number of characters is larger than the character size string, an operation error occurs.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The number of characters is larger than the character string size.

F260 (SSRC)
P260 (PSSRC)

Search for character string

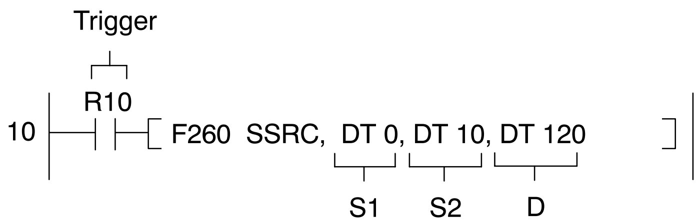
Outline

These instructions search for a specified character string.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P260 (PSSRC) cannot be specified.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 10
11	F260 (SSRC)
	DT 0
	DT 10
	DT 120

S1	Area in which the character data to be searched is stored (character string or character constant)
S2	Character string to be searched
D	Area in which the results of the search are stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

The DT0 character is searched from the character string of DT10, and the result is stored in DT120.

Characters for search

	Higher 16 bits	Lower 16 bits
DT0	10 (Character string size)	
DT1	3 (Number of characters)	
DT2	"F" (Byte 1)	"E" (Byte 0)
DT3	(Byte 3)	"G" (Byte 2)
DT4	(Byte 5)	(Byte 4)
DT5	(Byte 7)	(Byte 6)
DT6	(Byte 9)	(Byte 8)

Search →

Table of found characters

	Higher 16 bits	Lower 16 bits
DT10	10 (Character string size)	
DT11	8 (Number of characters)	
DT12	"B" (Byte 1)	"A" (Byte 0)
DT13	"D" (Byte 3)	"C" (Byte 2)
DT14	"F" (Byte 5)	"E" (Byte 4)
DT15	"H" (Byte 7)	"G" (Byte 6)
DT16	(Byte 9)	(Byte 8)

→

DT120	1	Number of matching character data
DT121	5	Position of first matching data

Description

The character data specified by "S1" is searched using the character string specified by "S2".

The number of characters that are the same, as resulting from the search, is stored in "D", and the first detected relative position (byte unit) is stored in "D + 1".

Precautions during programming

Specify a number of characters such that "S1" is less than or equal to "S2".

For the number of characters "S1 + 1" in the character string on the search side, designate the number of characters for performing search.

Example:

4 (Character string size)	
1 (Number of characters)	
"B"	"A"
"D"	"C"

← When the number of characters is 1, the system searches the character "A". When the number of characters is 2, the characters "AB" are searched as one unit.

Flag conditions

Error flag (R9007): Turns on and stays on when:

Error flag (R9008): Turns on for an instant when:

- The specified range is exceeded when an index is modified.
- The number of characters is larger than the character string size.

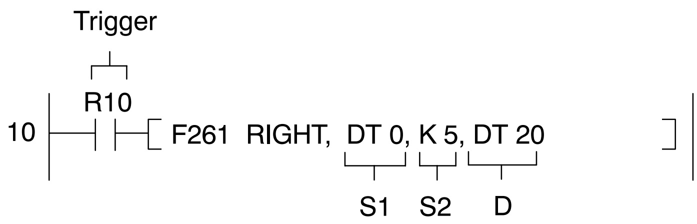
F261 (RIGHT) P261 (PRIGHT) Retrieving data from character strings (right side)

Outline These instructions retrieve a specified number of characters from the right side of the character string.

With the FP0R/FPΣ/FP–X, the differential execution type instruction P261 (PRIGHT) cannot be specified.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F261	(RIGHT)	
	DT		0
	K		5
	DT		20

S1	Character string
S2	Area in which the character string is stored, or constant data
D	Area in which the results of the search are stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

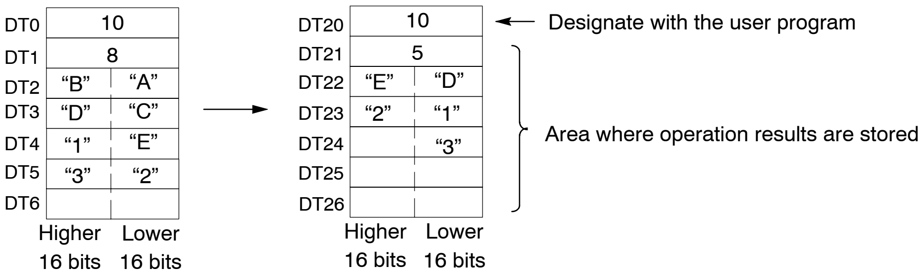
(*1) Cannot be specified with FP0R, FPΣ and FP–X.

(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

A character is retrieved from the end of the character string of DT0, and is sent to DT20.



Description

The number of characters specified by “S2” is searched starting from the right side (the end of the character data) of the character string specified by “S1”, and is sent to the character string specified by “D”.
At the starting address of the area for storing results “D”, designate the character string size using the user program.

Precautions during programming

The character data from “D” prior to the operation is cleared.

If the number of characters specified by “S2” is larger than the number of characters in the character string specified by “S1”, the number of characters of the character string specified by “S1” is sent.

If the number of characters specified by “S2” is larger than the size of the character string of “D”, data equal to the size of the character string specified by “D” is sent.

Flag conditions

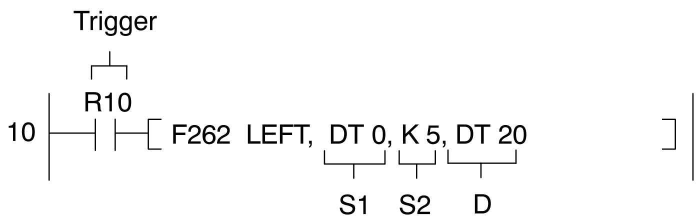
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The number of characters is larger than the character string size.
- Carry flag (R9009): Turns on for an instant when the result of the operation is larger than the size of the character string specified by “D”.

F262 (LEFT) P262 (PLEFT) Retrieving data from character strings (left side)

Outline These instructions retrieve a specified number of characters from the left side of the character string.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P262 (PLEFT) cannot be specified.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 10 F262 (LEFT) DT 0 K 5 DT 20

S1	Character string
S2	Area in which the character string is stored, or constant data
D	Area in which the character string is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

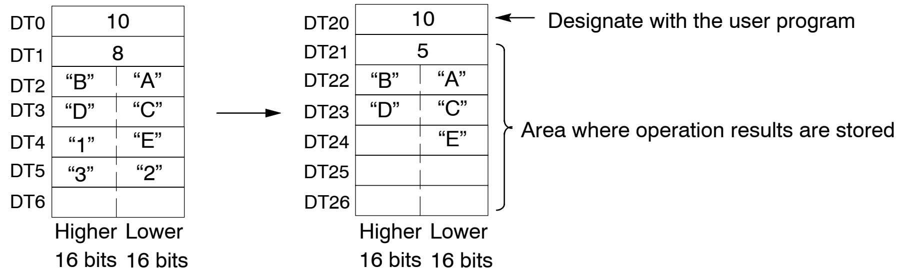
(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

A character is retrieved from the beginning of the character string of DT0, and is sent to DT20.



Description

The number of characters specified by “S2” is searched starting from the left side (the beginning of the character data) of the character string specified by “S1”, and is sent to the character string specified by “D”. At the starting address of the area for storing results “D”, designate the character string size using the user program.

Precautions during programming

The character data from “D” prior to the operation is cleared.

If the number of characters specified by “S2” is larger than the number of characters in the character string specified by “S1”, the number of characters of the character string specified by “S1” is sent.

If the number of characters specified by “S2” is larger than the size of the character string of “D”, data equal to the size of the character string specified by “D” is sent.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The number of characters is larger than the character string size.
- Carry flag (R9009): Turns on for an instant when the result of the operation is larger than the size of the character string specified by “D”.

F263 (MIDR) P263 (PMIDR)

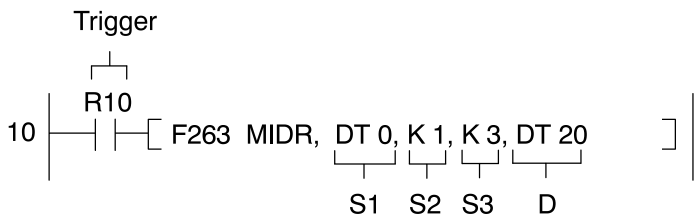
Retrieving a character string from a character string

Outline These instructions retrieve a character string consisting of a specified number of characters from the specified position in the character string.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P263 (PMIDR) cannot be specified.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	10
11	F263	(MIDR)
	DT	0
	K	1
	K	3
	DT	20

S1	Character string
S2	Area in which the character string position is stored, or constant data
S3	Area in which the number of characters is stored, or constant data
D	Area in which the character string is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

Three characters are retrieved from the position byte 1 (second character) of the character string of DT0, and are sent to DT20.

	Higher 16 bits	Lower 16 bits
DT0	10	
DT1	8	
DT2	"B"	"A"
DT3	"D"	"C"
DT4	"1"	"E"
DT5	"3"	"2"
DT6		

→

	Higher 16 bits	Lower 16 bits
DT20	10	
DT21	3	
DT22	"C"	"B"
DT23		"D"
DT24		
DT25		
DT26		

DT20 ← Designate with the user program

DT21–DT26: Area where operation results are stored

Description

The number of characters specified by "S3" is retrieved starting from the position specified by "S2" in the character string specified by "S1", and is sent to the character string specified by "D".
At the starting address of the area for storing results "D", designate the character string size using the user program.

Precautions during programming

The character data from "D" prior to the operation is cleared.

If the number of characters specified by "S3" is larger than the number of characters in the character string specified by "S1" from the position specified by "S2", the number of characters of the character string specified by "S1" is sent.

If the number of characters of the operation result is larger than the size of the character string of "D", data equal to the size of the character string specified by "D" is sent.

The position specified by "S2" sets the least significant byte as K0 (byte 0), and the positions are counted in the order of 0, 1, 2, etc., starting from the least significant byte.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The number of characters is larger than the character string size
  - The number of characters of "S1" is larger than the number of characters of "S2"
- Carry flag (R9009): Turns on for an instant when the result of the operation is larger than the size of the character string specified by "D"

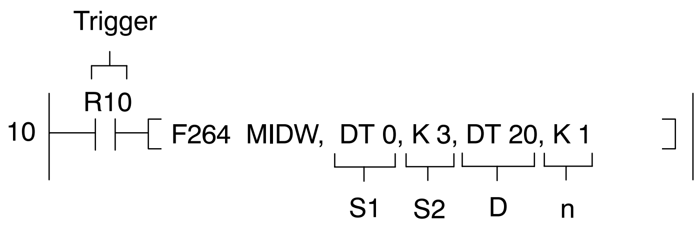
F264 (MIDW) P264 (PMIDW) Writing a character string to a character string

Outline These instructions write a specified number of characters from a character string to a specified position in the character string.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P264 (PMIDW) cannot be specified.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F264		(MIDW)
	DT		0
	K		3
	DT		20
	K		1

S1	Character string
S2	Area in which the character string position is stored, or constant data
D	Starting address of the area in which the character string is stored
n	Area in which the position of the character string is stored, or constant data

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

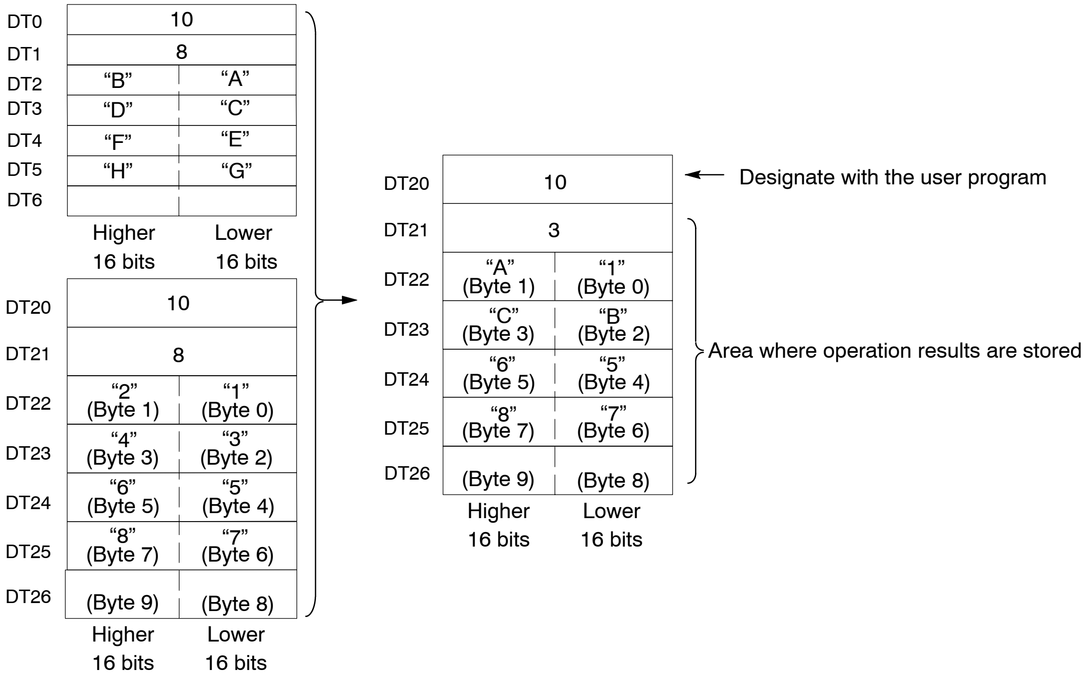
(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

Three characters are retrieved from the character string of DT0, and are sent to the position byte 1 (second character) of the character string block of DT20.



Description

The number of characters specified by "S2" is retrieved from the character string specified by "S1", and is sent to the "n" position of the character string specified by "D".

Precautions during programming

The character data from "D" prior to the operation is not cleared (it is overwritten).

If the number of characters specified by "S2" is larger than the number of characters in the character string specified by "S1", the number of characters of the character string specified by "S1" is sent.

If the position of "n" is larger than the number of characters of the character string of "D", an operation error occurs.
If the number of characters in the operation result is larger than the size of the character string in "D", then replacement is done only within a range the size of the character string in "D".

The position specified by "n" sets the least significant byte as K0 (byte 0), and the positions are counted in the order of 0, 1, 2, etc., starting from the least significant byte.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The number of characters is larger than the character string size
  - The number of characters of "D" is larger than the number of characters of "n"
- Carry flag (R9009): Turns on for an instant when the result of the operation is larger than the size of the character string specified by "D"

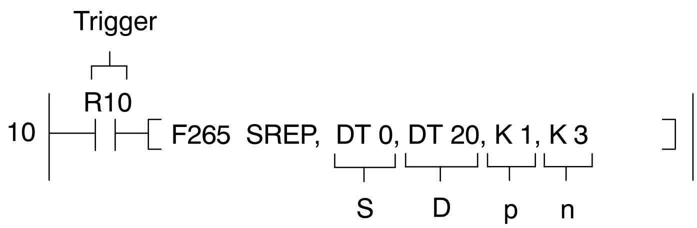
F265 (SREP) P265 (PSREP) Replacing character strings

Outline These instructions replace a specified number of characters in a character string with the same number of different characters, starting from a specified position.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P265 (PSREP) cannot be specified.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 10
11	F265 (SREP)
	DT 0
	DT 20
	K 1
	K 3

S	Replacement character string
D	Starting address of the area in which the character string is stored
p	Area storing the head byte position of the character to be replaced, or constant data
n	Area storing the number of characters to be replaced from the source data, or constant data

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant				Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	M	
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A
P	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	A

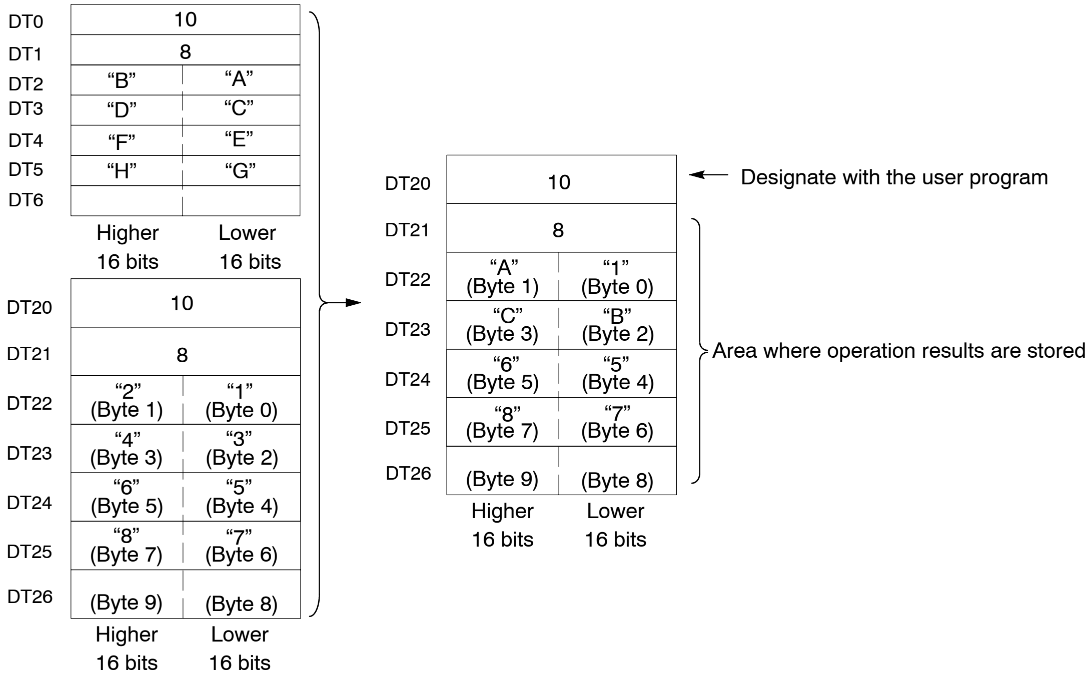
(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

The DT0 character string is replaced with the number of characters in DT1 (5 characters) from byte p=1 in DT20. In this case, n=3 characters of the data stored in the source are deleted in the replacement.



Description

The character string specified by "S" replaces the character string specified by "D", for the number of characters specified by "n", starting from the position specified by "P".

Precautions during programming

The character data from "D" prior to the operation is not cleared (it is overwritten).

If the number of characters in "n" is larger than the number of characters in the character string "S" subsequent to the point designated with "p", replacement is done for the number of characters in the character string "S" subsequent to the point designated with "p".

If the position specified by "p" is larger than the number of characters in the character string specified by "n", an operation error occurs.

The position specified by "p" sets the least significant byte as K0 (byte 0), and the positions are counted in the order of 0, 1, 2, etc., starting from the least significant byte.

Flag conditions

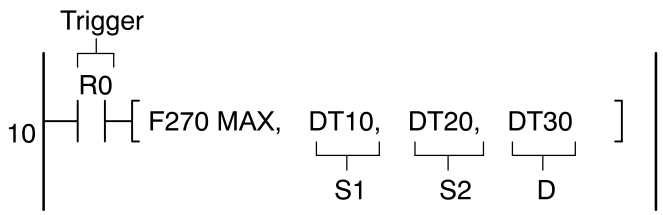
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The number of characters is larger than the character string size
  - The number of characters of "D" is larger than the number of characters of "n"
- Carry flag (R9009): Turns on for an instant when the result of the operation is larger than the size of the character string specified by "D"

F270 (MAX) P270 (PMAX)

Maximum value search in 16-bit data table

Outline Searches for a maximum value in a table of 16-bit areas.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P270 (PMAX)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F270 (MAX)
		DT 10
		DT 20
		DT 30

S1	Starting 16-bit area of data table
S2	Ending 16-bit area of data table
D	Lower 16-bit area of 32-bit data for storing maximum value and relative address

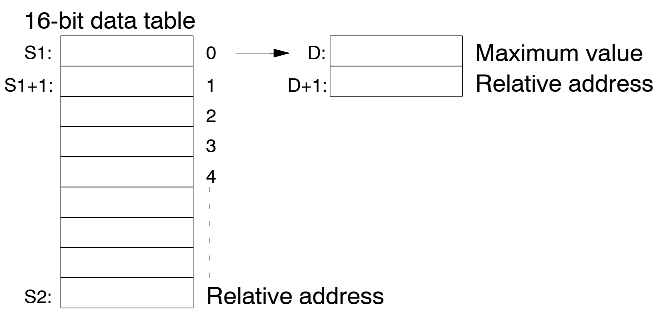
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

This instruction searches for the maximum value in the 16-bit data table between the area selected with S1 and the area selected with S2, and stores it in the area selected with D. The address relative to S1 is stored in D+1.



If there are several values which are a maximum value, the relative address of the first value found searching from S1 is stored in D+1.

Precaution during programming

Even if D+1 overflows the selected area, it will still be stored, and this may corrupt the data in the leading part of the other area. (An area overflow check is not performed.)

Flag conditions

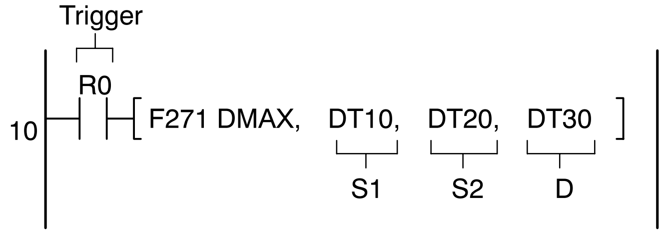
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.

F271 (DMAX) P271 (PDMAX) Maximum value search in 32-bit data table

Outline Searches for a maximum value in a table of 32-bit areas.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P271 (PDMAX)” is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F271 (DMAX)
	DT 10
	DT 20
	DT 30

S1	Starting 16-bit area of 32-bit data table
S2	Ending 16-bit area of double word (32-bit)
D	Starting 16-bit area for storing maximum value and relative address (3 words)

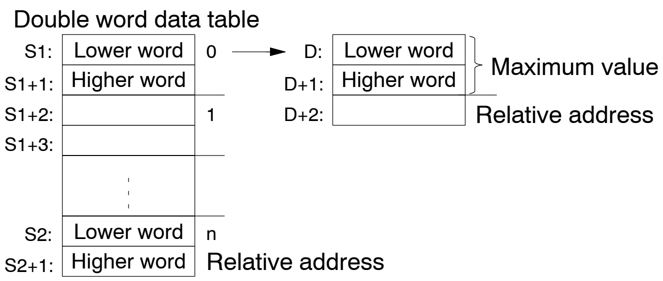
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

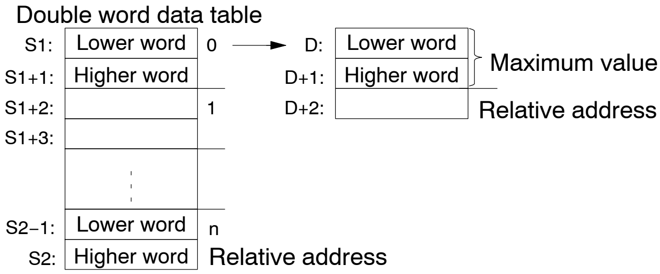
A: Available
N/A: Not Available

Description

This instruction searches for the maximum value in the double word data table between the area selected with S1 and the area selected with S2, and stores it in the area selected with D. The address relative to S1 is stored in D+2.



If S2 specifies a higher word of double word data, processing will take place over the same area as if the lower word had been specified.



If there are several values which are a maximum value, the relative address of the first value found searching from S1 is stored in D+2.

Precaution during programming

Even if D+2 overflows the selected area, it will still be stored, and this may corrupt the data in the leading part of the other area. (An area overflow check is not performed.)

Flag conditions

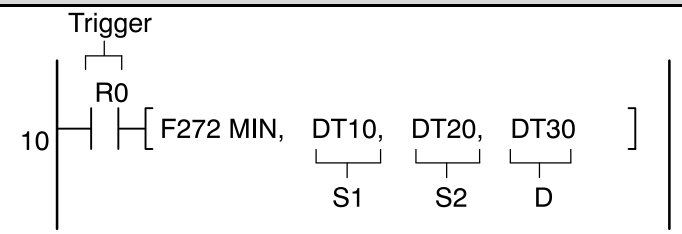
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.

F272 (MIN) P272 (PMIN) Minimum value search in 16-bit data table

Outline Searches for a minimum value in a table of 16-bit areas.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P272 (PMIN)” is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	0
11	F272		(MIN)
	DT		10
	DT		20
	DT		30

S1	Starting 16-bit area of data table
S2	Ending 16-bit area of data table
D	Starting 16-bit area for storing minimum value and relative address (2 words)

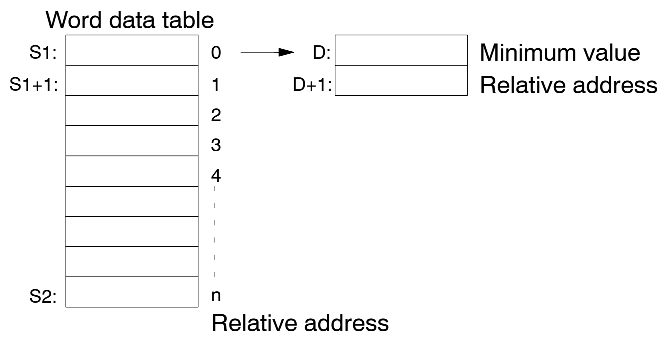
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

This instruction searches for the minimum value in the 16-bit data table between the area selected with S1 and the area selected with S2, and stores it in the area selected with D. The address relative to S1 is stored in D+1.



If there are several values which are a minimum value, the relative address of the first value found searching from S1 is stored in D+1.

Precaution during programming

Even if D+1 overflows the selected area, it will still be stored, and this may corrupt the data in the leading part of the other area. (An area overflow check is not performed.)

Flag conditions

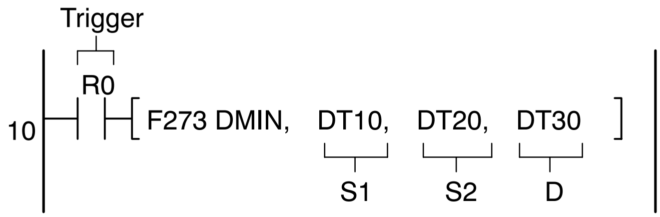
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.

F273 (DMIN) P273 (PDMIN) Minimum value search in 32-bit data table

Outline Searches for a minimum value in a table of 32-bit areas.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P273 (PDMIN)” is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F273 (DMIN)
	DT 10
	DT 20
	DT 30

S1	Starting 16-bit area of 32-bit data table
S2	Ending 16-bit area of double word (32-bit)
D	Starting 16-bit area for storing minimum value and relative address (3 words)

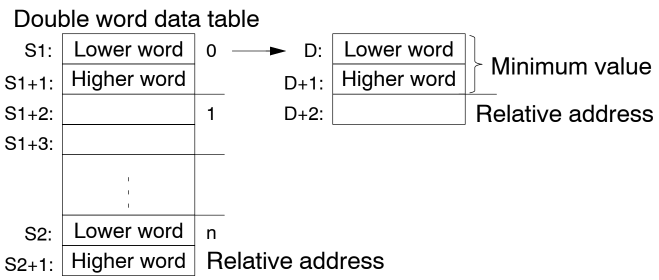
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

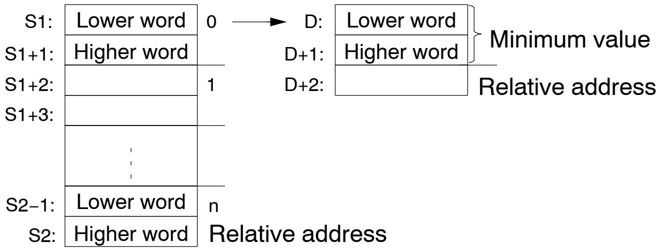
Description

This instruction searches for the minimum value in the double word data table between the area selected with S1 and the area selected with S2, and stores it in the area selected with D. The address relative to S1 is stored in D+2.



If S2 specifies a higher word of double word data, processing will take place over the same area as if the lower word had been specified.

Double word data table



If there are several values which are a minimum value, the relative address of the first value found searching from S1 is stored in D+2.

Precaution during programming

Even if D+2 overflows the selected area, it will still be stored, and this may corrupt the data in the leading part of the other area. (An area overflow check is not performed.)

Flag conditions

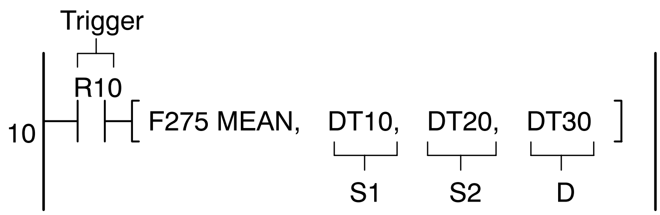
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.

F275 (MEAN)
P275 (PMEAN)

Total and mean numbers calculation in 16-bit data table

Outline Calculates the total and mean numbers in the specified word data table. For the FP0R/FPΣ/FP−X, the P type high−level instruction “P275 (PMEAN)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST X 10
	11	F275 (MEAN)
		DT 10
		DT 20
		DT 30

S1	Starting 16-bit area of data table
S2	Ending 16-bit area of data table
D	Starting 16-bit for storing total and mean numbers (3 words)

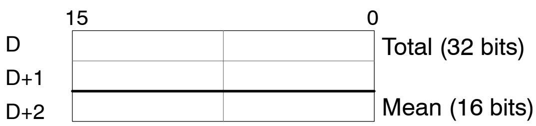
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

The total value and the average value of the word data (signed) from the area selected with S1 to the area selected with S2 are obtained and stored in the area selected with D.



Decimals of the average value are rounded off so that the average value is an integer.

Precaution during programming

Even if D+2 overflows the selected area, it will still be stored, and this may corrupt the data in the leading part of the other area. (An area overflow check is not performed.)

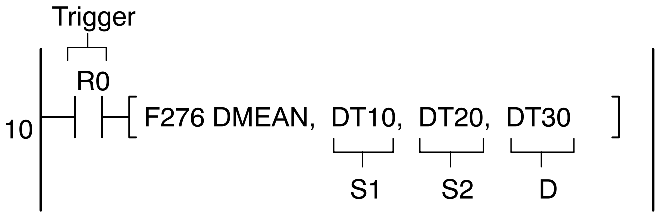
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.
- Carry flag (R9009): Turns on for an instant when overflows/underflows while calculating.

F276 (DMEAN) P276 (PDMEAN) Total and mean numbers calculation in 32-bit data table

Outline Calculates the total and mean numbers in the specified double word data table.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P276 (PDMEAN)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F276 (DMEAN) DT 10 DT 20 DT 30

S1	Starting 16-bit area of 32-bit data table
S2	Ending 16-bit area of double word (32-bit)
D	Starting 16-bit area for storing total and mean numbers (5 words)

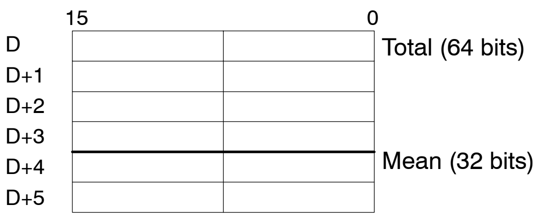
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

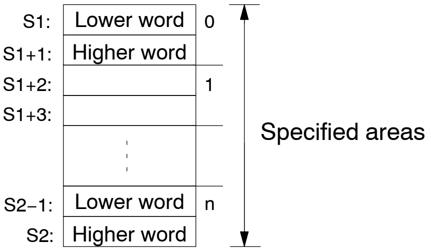
Description

The total value and the average value of the double word data (signed) from the area selected with S1 to the area selected with S2 are obtained and stored in the area selected with D.



If S2 specifies a higher word of double word data, processing will take place over the same area as if the lower word had been specified.

Double word data table



Decimals of the average value are rounded off so that the average value is an integer.

Precaution during programming

Even if D + 5 overflows the selected area, it will still be stored, and this may corrupt the data in the leading part of the other area. (An area overflow check is not performed.)

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The area specified using the index modifier exceeds the limit.
  - – S1 > S2.
  - – The areas of S1 and S2 are different.
- Carry flag (R9009): Turns on for an instant when overflows/underflows while calculating.

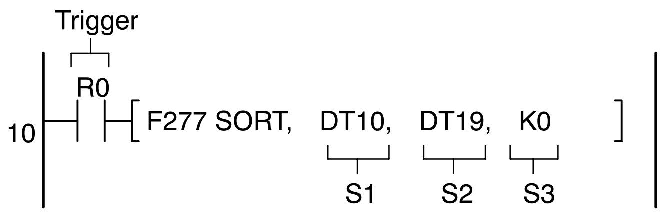
F277 (SORT) P277 (PSORT)

Sort data in 16-bit data table (in smaller or larger number order)

Outline Sorts a string of data words.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P277 (PSORT)” is not available.

Program example

Ladder Diagram



Address	Instruction		
10	ST	R	0
11	F277		(SORT)
	DT		10
	DT		19
	K		0

S1	Starting 16-bit area of sort data
S2	Ending 16-bit area of sort data
S3	Constant or area where sort condition is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the S3 is “K0 (ascending order)”

	Before		After
DT10	K300	DT10	K−30
11	K10	11	K−3
12	K3	12	K−1
13	K−1	13	K1
14	K1000	14	K3
15	K−30	15	K10
16	K100	16	K30
17	K30	17	K100
18	K1	18	K300
19	K−3	19	K1000

When the S3 is “K1 (descending order)”

	Before		After
DT10	K300	DT10	K1000
11	K10	11	K300
12	K3	12	K100
13	K−1	13	K30
14	K1000	14	K10
15	K−30	15	K3
16	K100	16	K1
17	K30	17	K−1
18	K1	18	K−3
19	K−3	19	K−30

Description

The data words (signed) from the area specified by S1 to the area specified by S2 are sorted in ascending order (the smallest word is first) or descending order (the largest word is first) depending on the condition set with S3.

If S1 = S2, sorting does not take place.

The sort condition is specified as follows in S3:

– K0: Ascending order

– K1: Descending order

Double sorting is used for the sorting method. Data is sorted from S1 to S2 in order following the sorting procedure. Note that the number of word comparisons increases in proportion to the square of the number of words, thus more time will be required for execution when there are a large number of words.

Flag conditions

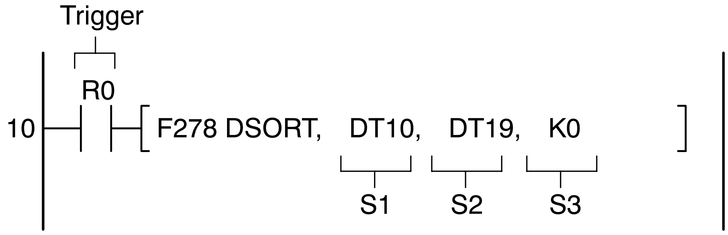
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The area specified using the index modifier exceeds the limit.
  - – S1 > S2.
  - – The areas of S1 and S2 are different.

F278 (DSORT) P278 (PDSORT) Sort data in 32-bit data table (in smaller or larger number order)

Outline Sorts a string of data double words.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P278 (PDSORT)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F278		(DSORT)
	DT		10
	DT		19
	K		0

S1	Starting 16-bit area of sort data (2 words)
S2	Ending 16-bit area of sort data (2 words)
S3	Constant or area where sort condition is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the S3 is “K0 (ascending order)”

DT10, 11	K25000
12, 13	K−4000
14, 15	K1500
16, 17	K−2600
18, 19	K100000

⇨

DT10, 11	K−4000
12, 13	K−2600
14, 15	K1500
16, 17	K25000
18, 19	K100000

When the S3 is “K1 (descending order)”

DT10, 11	K25000
12, 13	K−4000
14, 15	K1500
16, 17	K−2600
18, 19	K100000

⇨

DT10, 11	K100000
12, 13	K25000
14, 15	K1500
16, 17	K−2600
18, 19	K−4000

Description

The double data words (signed) from the area specified by S1 to the area specified by S2 are sorted in ascending order (the smallest word is first) or descending order (the largest word is first) depending on the condition set with S3.

If S1 = S2, sorting does not take place.

The sort condition is specified as follows in S3:

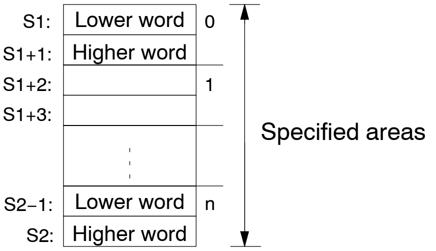
– K0: Ascending order

– K1: Descending order

Double sorting is used for the sorting method. Data is sorted from S1 to S2 in order following the sorting procedure. Note that the number of word comparisons increases in proportion to the square of the number of words, thus more time will be required for execution when there are a large number of words.

If S2 specifies a higher word of double word data, processing will take place over the same area as if the lower word had been specified.

Double word data table



Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The area specified using the index modifier exceeds the limit.
  - – S1 > S2.
  - – The areas of S1 and S2 are different.

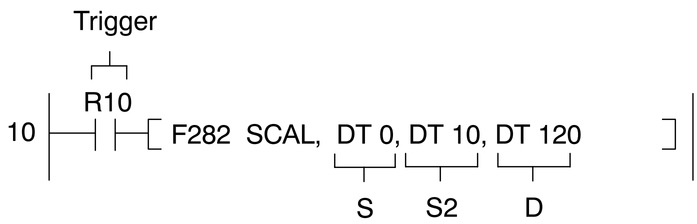
F282 (SCAL) P282 (PSCAL) Scaling of 16−bit data

Outline The output value Y is found for the input value X by performing scaling for the given data table.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P282 (PSCAL) cannot be specified.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	10
11	F282		(SCAL)
	DT		0
	DT		10
	DT		120

S1	16−bit data of the source corresponding to the input value X, or area storing data
S2	Starting address of data table used for scaling (linearization)
D	Area where output result Y is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available
N/A: Not Available

Explanation of example

The output value Y for the input value X stored in DT0 is found by accessing the data table starting from DT10, and the result is stored in DT120.

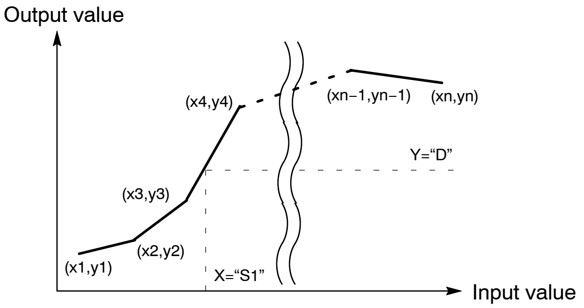
Description

The output value for the input value X is found by performing scaling according to the data table, where the 16−bit data designated in “S1” is designated in “S2”.
The number “n” of items in the data table is determined by the value “n” designated for the head “S2” of the data table.

Configuration of the data table used for scaling

When S2 = DT10 and n = K10

S2:	n	DT10
S2+1:	x1	DT11
S2+2:	x2	DT12
S2+3:	x3	DT13
S2+n−1:	xn−1	DT19
S2+n:	xn	DT20
S2+n+1:	y1	DT21
S2+n+2:	y2	DT22
S2+n+3:	y3	DT23
S2+2n−1:	yn−1	DT29
S2+2n:	yn	DT30



Precautions during programming

Make x_t greater than x_{t-1}.

xt and yt should be created as 16−bit data to indicate which line is specified.

If X(S1) is a value smaller than x1, the value of Y(D) will be the value of y1.

If X(S1) is larger than xn, Y(D) will be the value of yn.

The maximum value of n is 99.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The n specified by “S2” is smaller than 2, or if the n is larger than 99
  - The data table specified by “S2” exceeds the available area
  - Xn are not in ascending order

F283 (DSCAL) P283 (PDSCAL)

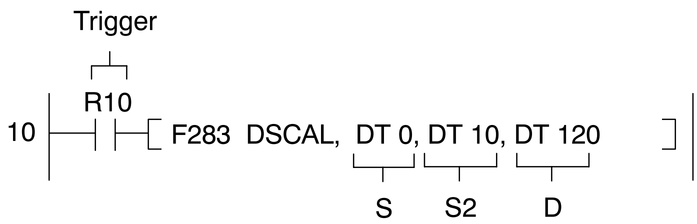
Scaling of 32−bit data

Outline The output value Y is found for the input value X by performing scaling for the given data table.

With the FP0R/FPΣ/FP−X, the differential execution type instruction P283 (PDSCAL) cannot be specified.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	10
11	F283	(DSCAL)	
	DT		0
	DT		10
	DT		120

S1	32−bit data of the source corresponding to the input value X, or area storing data
S2	Starting address of data table used for scaling (linearization)
D	Area where output result Y is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	FL (*1)	I (*2)	K	H	f	
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A

(*1) Cannot be specified with FP0R, FPΣ and FP−X.

(*2) I0 to ID

A: Available
N/A: Not Available

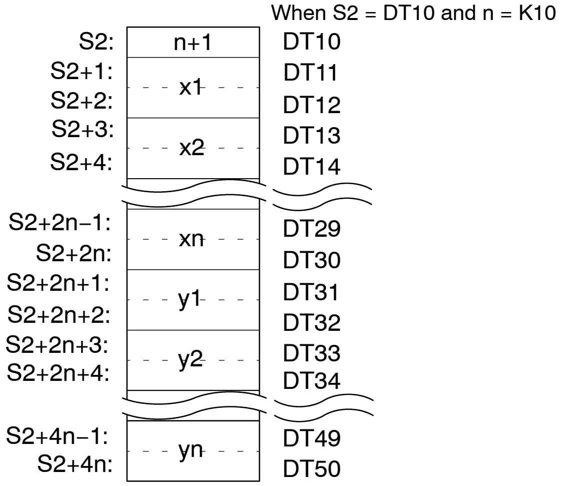
Explanation of example

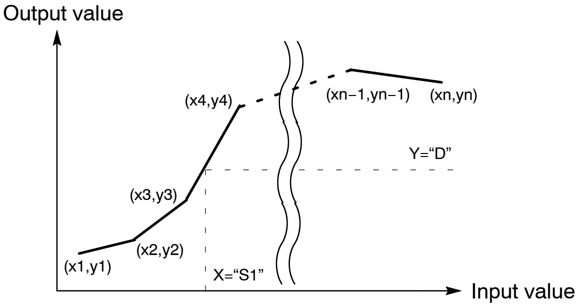
The output value Y for the input value X stored in DT0 is found by accessing the data table starting from DT10, and the result is stored in DT120 and DT121.

Description

The output value for the input value X is found by performing scaling according to the data table, where the 32−bit data designated in “S1” is designated in “S2”.
The number “n” of items in the data table is determined by the value “n” designated for the head “S2” of the data table.

Configuration of the data table used for scaling





Precautions during programming

Make x_t greater than x_{t-1}.

xt and yt should be created as 32−bit data to indicate which line is specified.

If X(S1) is a value smaller than x1, the value of Y(D) will be the value of y1.

If X(S1) is larger than xn, Y(D) will be the value of yn.

The maximum value of n is 99.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The specified range is exceeded when an index is modified.
  - The n specified by “S2” is smaller than 2, or if the n is larger than 99
  - The data table specified by “S2” exceeds the available area
  - Xn are not in ascending order

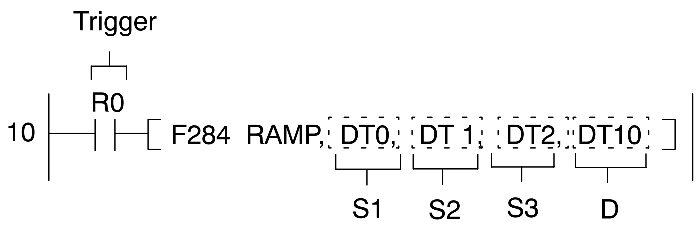
Availability
FP−X V2.0 or more FPΣ V3.10 or more FP0R

F284 (RAMP) Inclination output of 16−bit data

Outline Executes the linear output according to the elapsed time from the start by performing scaling with the output initial value, target value and time range.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F284 (RAMP)
	DT 0
	DT 1
	DT 2
	DT 10

S1	16−bit equivalent constant or 16−bit area for storing initial values
S2	16−bit equivalent constant or 16−bit area for storing target values
S3	16−bit equivalent constant or 16−bit area for storing time range
D	Area where output result is stored

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	A
N	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A

(*1) I0 to ID

A: Available
N/A: Not Available

Operation

Executes the linear output according to the elapsed time from the start by performing scaling with the 16−bit output initial value specified by S1, the 16−bit output target value specified by S2 and the 16−bit output time range (ms unit) specified by S3.

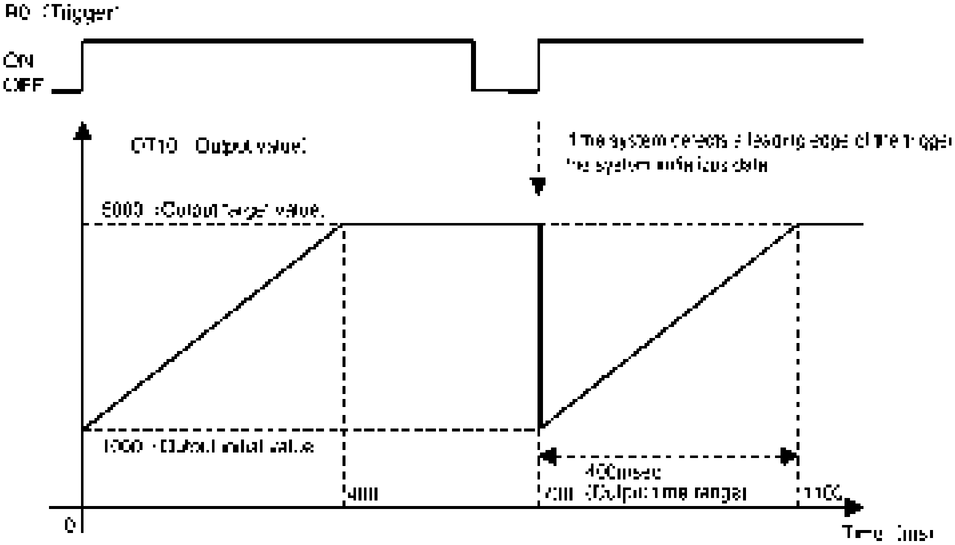
Precautions during programming

Max. 1 scan time error in the output time range occurs occasionally.

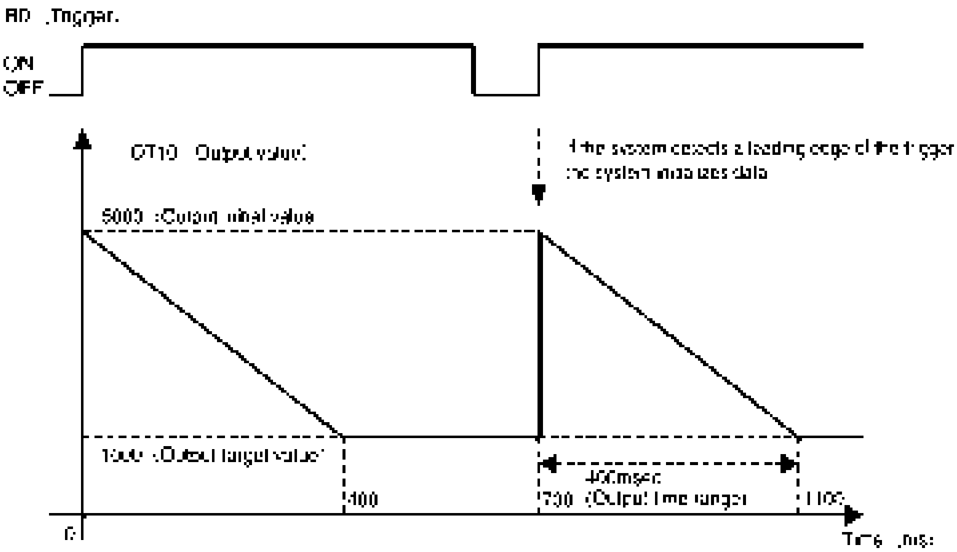
Explanation of example

When specifying each value as below by the program:

DT0 (Initial value) = k1000
DT1 (Target value) = k5000
DT2 (Time range) = k400



DT0 (Initial value) = k5000
DT1 (Target value) = k1000
DT2 (Time range) = k400



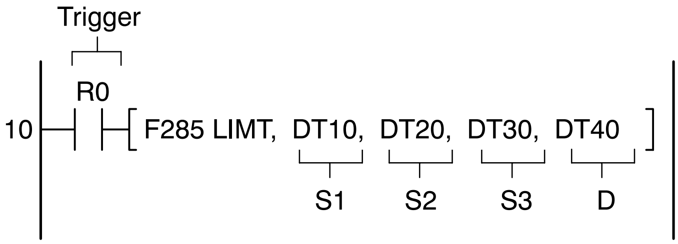
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - the area specified using the index modifier exceeds the limit.
  - the output time range specified by ”S3” is smaller than k1 or larger than k30000.

F285 (LIMT) P285 (PLIMT) 16-bit data upper and lower limit control

Outline This instruction carries out upper and lower limit control for 16-bit data. For the FP0R/FPΣ/FP−X, the P type high−level instruction “P285 (PLIMT)” is not available.

Program example



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F285		(LIMT)
	DT		10
	DT		20
	DT		30
	DT		40

S1	The area where the lower limit is stored or the lower limit data.
S2	The area where the upper limit is stored or the upper limit data.
S3	The area where the input value is stored or the input value data.
D	The area where the output value is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

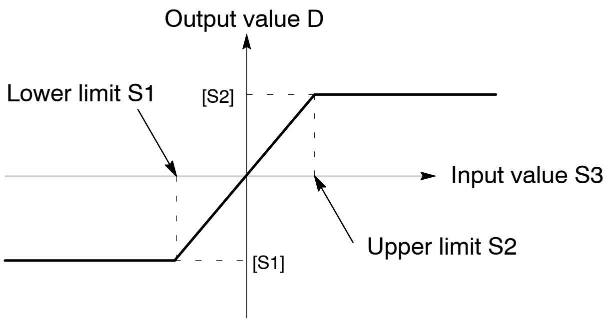
The 16-bit output value stored in the area specified by D is controlled based on whether or not the 16-bit input value specified by S3 falls within the range bounded by the upper and lower limits set in S2 and S1.

The output value is determined based on the following conditions:

– When the lower limit S1 is greater than the input value S3, the lower limit value S1 is stored in D as the output value.

– When the upper limit S2 is less than the input value S3, the upper limit value S2 is stored in D as the output value.

– When Lower limit S1 ≦ Input value S3 ≦ Upper limit S2, the input value S3 is stored in D as the output value.



To perform upper limit control only, set K−32768 (or H8000) for the lower limit S1.

To perform lower limit control only, set K32767 (or H7FFF) for the upper limit S2.

Flag conditions

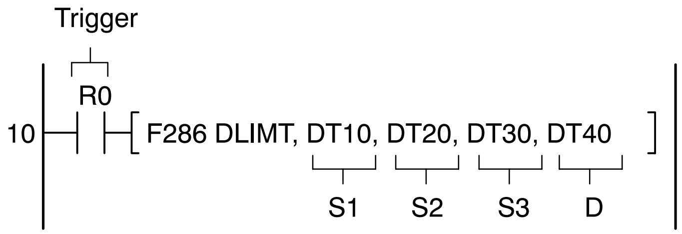
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – S1 > S2.
- = flag (R900B): Turns on when the result of processing is between the upper and lower limits.

F286 (DLIMT) P286 (PDLIMT) 32-bit data upper and lower limit control

Outline This instruction carries out upper and lower limit control for 32-bit data. For the FP0R/FPΣ/FP−X, the P type high−level instruction “P286 (PDLIMT)” is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F286 (DLIMT)
	DT 10
	DT 20
	DT 30
	DT 40

S1	The area where the lower limit is stored or the lower limit data. (2 words)
S2	The area where the upper limit is stored or the upper limit data. (2 words)
S3	The area where the input value is stored or the input value data. (2 words)
D	The area where the output value is stored. (2 words)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

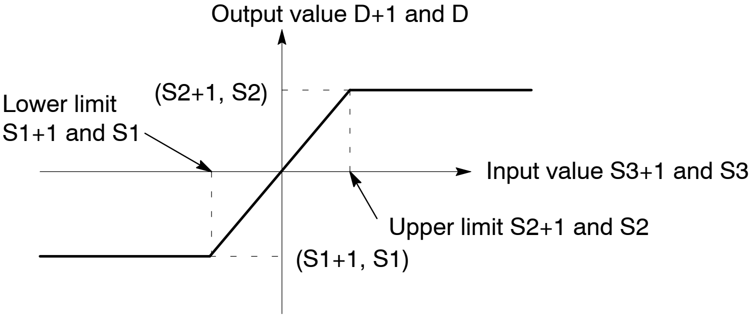
The output value (double words data) stored in the area specified by D is controlled based on whether or not the input value (double words data) specified by S3 falls within the range bounded by the upper and lower limits set in S2 and S1.

The output value is determined based on the following conditions:

– When the lower limits S1+1 and S1 are greater than the input value S3+1 and S3, the lower limit value S1+1 and S1 are stored in D+1 and D as the output value.

– When the upper limits S2+1 and S2 are less than the input value S3+1 and S3, the upper limit value S2+1 and S2 are stored in D+1 and D as the output value.

– When Lower limit S1+1 and S1 ≦ Input value S3+1 and S3 ≦ Upper limit S2+1 and S2, the input value S3+1 and S3 are stored in D+1 and D as the output value.



To perform upper limit control only, set K−2147483648 (or H80000000) for the lower limit S1+1 and S1.

To perform lower limit control only, set K2147483647 (or H7FFFFFFF) for the upper limit S2+1 and S2.

Flag conditions

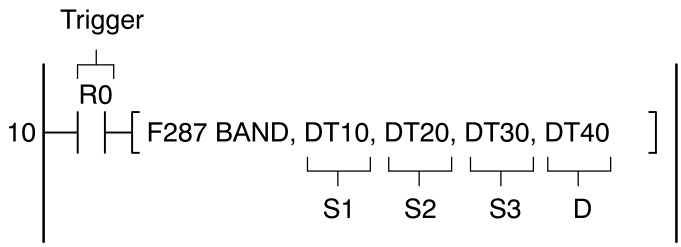
• Error flag (R9007):	Turns on and stays on when:
• Error flag (R9008):	Turns on for an instant when:
	– The area specified using the index modifier exceeds the limit.
	– S1 > S2.
• = flag (R900B):	Turns on when the result of processing is between the upper and lower limits.

F287 (BAND) P287 (PBAND) 16-bit data deadband control

Outline This instruction carries out dead-band control for 16-bit data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P287 (PBAND)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F287		(BAND)
	DT		10
	DT		20
	DT		30
	DT		40

S1	The area where the lower limit is stored or the lower limit data.
S2	The area where the upper limit is stored or the upper limit data.
S3	The area where the input value is stored or the input value data.
D	The area where the output value is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the DT10 is K−100 and DT20 is K100.

Value of DT30	Value of DT40
K−300 →	K−200
K−200 →	K−100
K−100 to K100 →	K0
K200 →	K100
K300 →	K200

Description

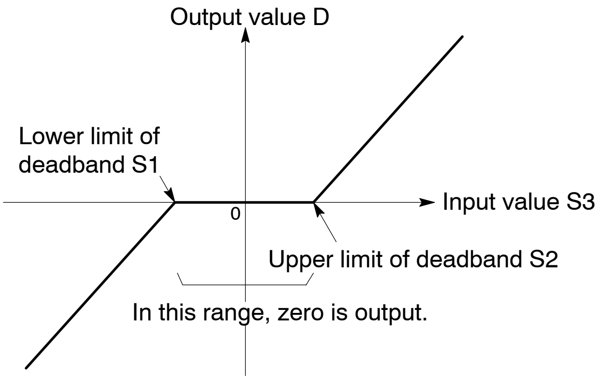
The output value (word data) stored in the area specified by D is controlled based on whether or not the input value (word data) specified by S3 falls within the dead-band bounded by the upper and lower limits set in S1 and S2.

The output value is determined based on the following conditions:

When the lower limit S1 is greater than the input value S3, the input value S3 minus the lower limit value S1 is stored in D as the output value.

When the upper limit S2 is less than the input value S3, the input value S3 minus the upper limit value S2 is stored in D as the output value.

When Lower limit S1 ≦ Input value S3 ≦ Upper limit S2, zero is stored in D as the output value.



Flag conditions

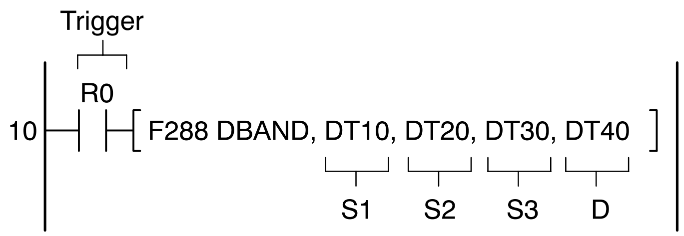
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
- Carry flag (R9009): Turns on for an instant when the calculated result is overflowed or underflowed.
- = flag (R900B): Turns on when the input value is recognized as “0.

F288 (DBAND) P288 (PDBAND) 32-bit data deadband control

Outline This instruction carries out dead-band control for 32-bit data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P288 (PDBAND)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F288		(DBAND)
	DT		10
	DT		20
	DT		30
	DT		40

S1	The area where the lower limit is stored or the lower limit data. (2 words)
S2	The area where the upper limit is stored or the upper limit data. (2 words)
S3	The area where the input value is stored or the input value data. (2 words)
D	The area where the output value is stored. (2 words)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When “DT10 and DT11” is “K−10000”and “DT20 and DT21” is “K10000”.

Value of DT30 and DT31	Value of DT40 and DT41
K−30000 →	K−20000
K−20000 →	K−10000
K−10000 to K10000 →	K0
K20000 →	K10000
K30000 →	K20000

Description

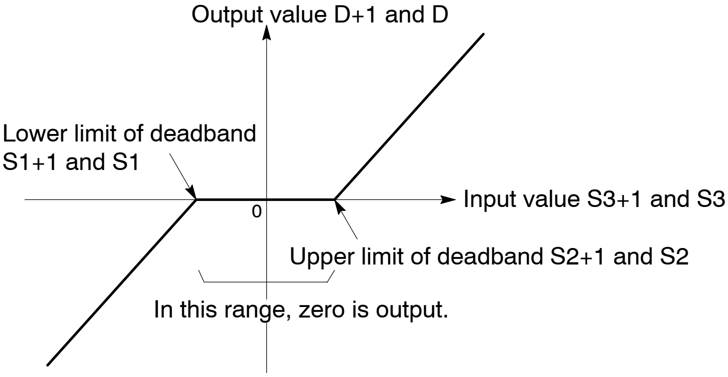
The output value (double word data) stored in the area specified by D is controlled based on whether or not the input value (double word data) specified by S3 falls within the dead-band bounded by the upper and lower limits set in S1 and S2.

The output value is determined based on the following conditions:

When the lower limit S1+1 and S1 are greater than the input value S3+1 and S3, the input value S3+1 and S3 minus the lower limit value S1+1 and S1 are stored in D+1 and D as the output value.

When the upper limit S2+1 and S2 are less than the input value S3+1 and S3, the input value S3+1 and S3 minus the upper limit value S2+1 and S2 are stored in D+1 and D as the output value.

When Lower limit S1+1 and S1 ≦ Input value S3+1 and S3 ≦ Upper limit S2+1 and S2, zero is stored in D+1 and D as the output value.



Flag conditions

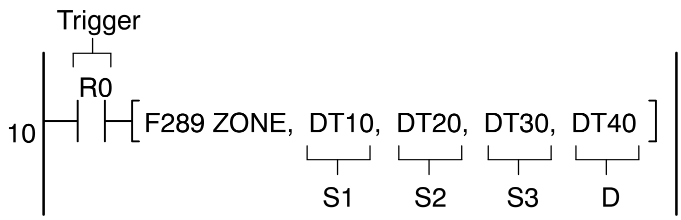
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
- Carry flag (R9009): Turns on for an instant when the calculated result is overflowed or underflowed.
- = flag (R900B): Turns on when the input value is recognized as “0.”

F289 (ZONE) P289 (PZONE) 16-bit data zone control

Outline This instruction carries out zone control for 16-bit data.
For the FP0R/FPΣ/FP−X, the P type high−level instruction “P289 (PZONE)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F289		(ZONE)
	DT		10
	DT		20
	DT		30
	DT		40

S1	Area where negative bias value is stored or negative bias value data
S2	Area where positive bias value is stored or positive bias value data
S3	Area where input value is stored or input value data
D	Area where output value is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the DT10 is “K−100” and DT20 is “K100”.

Value of DT30		Value of DT40
K−300	→	K−400
K−200	→	K−300
K−100	→	K−200
K0	→	K0
K100	→	K200
K200	→	K300
K300	→	K400

Description

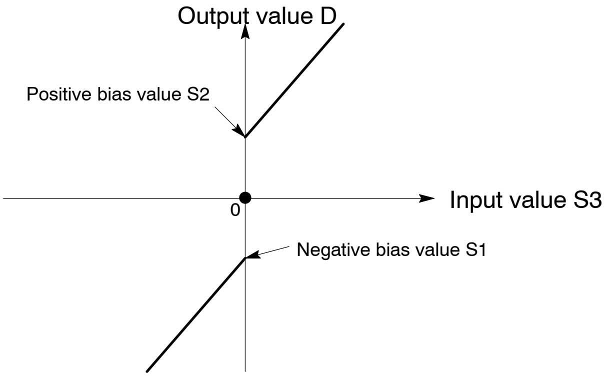
The bias value specified by S1 or S2 is added to the input value (word data) specified by S3, and the output value is stored in the area specified by D.

The output value is determined by the following conditions:

When the input value S3 is less than zero, the input value S3 plus the negative bias value S1 is stored in D as the output value.

When the input value S3 equals zero, zero is stored in D as the output value.

When the input value S3 is greater than zero, the input value S3 plus the positive bias value S2 is stored in D as the output value.



Flag conditions

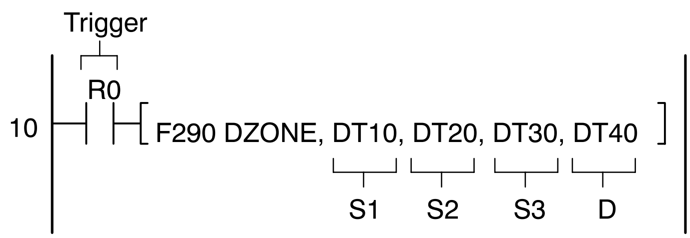
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the calculated result is overflowed or underflowed.
- = flag (R900B): Turns on for an instant when the input value S3 is recognized as “0”.

F290 (DZONE) P290 (PDZONE) 32-bit data zone control

Outline This instruction carries out zone control for 32-bit data. (double words) For the FP0R/FPΣ/FP−X, the P type high−level Instruction “P290 (PDZONE)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	0
11	F290	(DZONE)
	DT	10
	DT	20
	DT	30
	DT	40

S1	Area where negative bias value is stored or negative bias value data (double words)
S2	Area where positive bias value is stored or positive bias value data (double words)
S3	Area where input value is stored or input value data (double words)
D	Area (double words) where output value is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the “DT10 and DT11” is “K−10000” and the “DT20 and DT21” is “K10000”.

Value of DT30 and DT31	Value of DT40 and DT41
K−30000 →	K−40000
K−20000 →	K−30000
K−10000 →	K−20000
K0 →	K0
K10000 →	K20000
K20000 →	K30000
K30000 →	K40000

Description

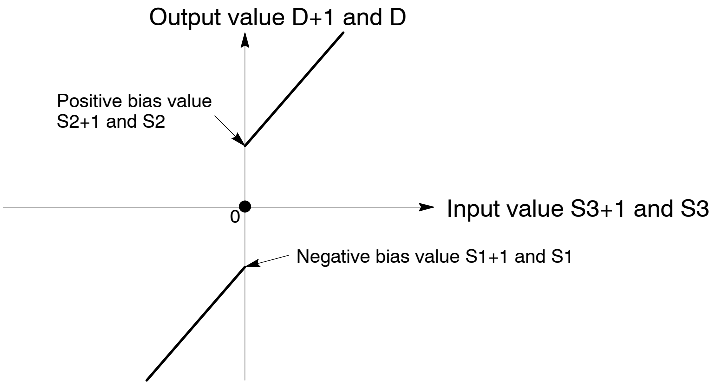
The bias value specified by S1 or S2 is added to the input value (double word data) specified by S3, and the output value is stored in the area specified by D.

The output value is determined by the following conditions:

When the input value S3+1 and S3 are less than zero, the input value S3+1 and S3 plus the negative bias value S1+1 and S1 are stored in D+1 and D as the output value.

When the input value S3+1 and S3 equals zero, zero is stored in D+1 and D as the output value.

When the input value S3+1 and S3 is greater than zero, the input value S3+1 and S3 plus the positive bias value S2+1 and S2 are stored in D+1 and D as the output value.



Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on for an instant when the calculated result is overflowed or underflowed.
- = flag (R900B): Turns on for an instant when the input value S3 is recognized as “0”.

F300 (BSIN) P300 (PBSIN)

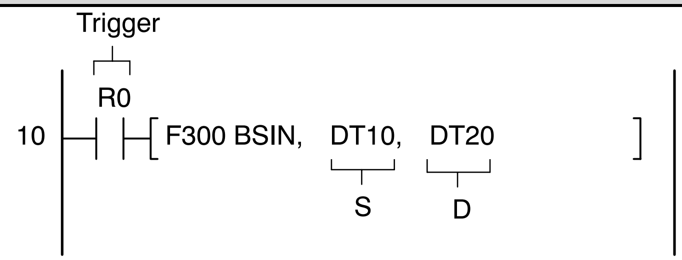
BCD type Sine operation

Outline

Triangle functions, calculates trigonometric functions and the sine [SIN( )] of BCD code angular data, and stores it as BCD.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F300		(BSIN)
	DT		10
	DT		20

S	Area where angle data is stored or angle data
D	Starting 16-bit area where calculated result is stored (3 words)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

Calculates the SIN ( ) of the angle 45 degrees.

DT10:	H45

↓

DT20:	H 0
DT21:	H 0
DT22:	H7071

Calculates the SIN ( ) of the angle 270 degrees.

DT10:	H270

↓

DT20:	H 1
DT21:	H 1
DT22:	H 0

Description

The SIN([S]) of an angle data (units are degrees) specified by S is calculated and the result stored in the 3-word area beginning at D.

SIN[S] → [D] [D+1]. [D+2]

D: Sign
D+1: Integer value
D+2: Decimal

Select a BCD value for S within the range 0° to 360° in units of 1 degree. Be sure to specify the value using BCD H data.

The sign stored in D is 0 when the result of processing is positive, and 1 when the result is negative.

The result of processing stored in D+1 and D+2 is a BCD value within the range −1.0000 to 1.0000.

The decimal stored in D+2 is rounded off to four digits.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data specified in S is not BCD value.
  - If data specified in S is not within 0° to 360°.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

F301 (BCOS) P301 (PBCOS) BCD type Cosine operation

Outline Triangle functions, calculates trigonometric functions and the cosine [COS ( )] of BCD code angular data, and stores it as BCD.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
Trigger R0; 10 ─┤├─[F301 BCOS, DT10 (S), DT20 (D)]	10	ST R 0
	11	F301 (BCOS)
		DT 10
		DT 20

S	Area where angle data is stored or angle data
D	Starting 16-bit area where calculated result is stored (3 words)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

Calculates the COS ( ) of the angle 30 degrees.

DT10: H30

↓

DT20: H 0
DT21: H 0
DT22: H8660

Calculates the COS ( ) of the angle 135 degrees.

DT10: H135

↓

DT20: H 1
DT21: H 1
DT22: H7071

Description

The COS([S]) of an angle data (units are degrees) specified by S is calculated and the result stored in the 3-word area beginning at D.

COS[S] → [D] [D+1]. [D+2]

D: Sign
D+1: Integer value
D+2: Decimal

Select a BCD value for S within the range 0° to 360° in units of 1 degree. Be sure to specify the value using BCD H data.

The sign stored in D is 0 when the result of processing is positive, and 1 when the result is negative.

The result of processing stored in D+1 and D+2 is a BCD value within the range −1.0000 to 1.0000.

The decimal stored in D+2 is rounded off to four digits.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data specified in S is not BCD value.
  - If data specified in S is not within 0° to 360°.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

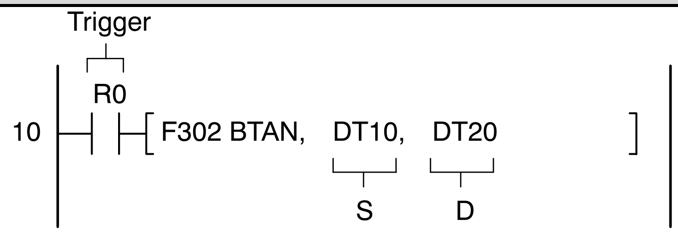
F302 (BTAN) P302 (PBTAN)

BCD type Tangent operation

Outline Triangle functions, calculates trigonometric functions and the tangent [TAN ( )] of BCD code angular data, and stores it as BCD.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F302 (BTAN)
	DT 10
	DT 20

S	Area where angle data is stored or angle data
D	Starting 16-bit area where calculated result is stored (3 words)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

Calculates the TAN ( ) of the angle 60 degrees.

DT10: H60

↓

DT20:	H 0
DT21:	H 1
DT22:	H7321

Calculates the TAN ( ) of the angle 135 degrees.

DT10: H135

↓

DT20:	H 1
DT21:	H 1
DT22:	H 0

Description

The TAN([S]) of an angle data (units are degrees) specified by S is calculated and the result stored in the 3-word area beginning at D.

TAN[S] → [D] [D+1]. [D+2]

D: Sign
D+1: Integer value
D+2: Decimal

Select a BCD value for S within the range 0° to 360° in units of 1 degree. Be sure to specify the value using BCD H data.

The sign stored in D is 0 when the result of processing is positive, and 1 when the result is negative.

The result of processing stored in D+1 and D+2 is a BCD value within the range −57.2900 to 57.2900.

The decimal stored in D+2 is rounded off to four digits.

Flag conditions

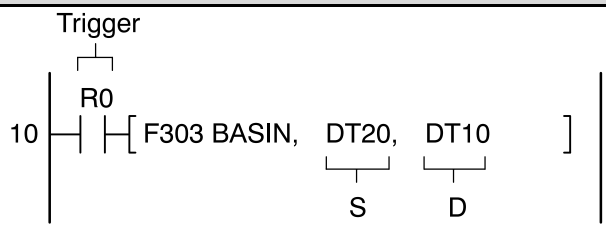
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data specified in S is not BCD value.
  - If data specified in S is not within 0° to 360°.
  - If data specified in S is 90° to 270°.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

F303 (BASIN) P303 (PBASIN) BCD type Arcsine operation

Outline Triangle functions, This instruction calculates arcsine [SIN^{-1} ( )].

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F303		(BASIN)
	DT		20
	DT		10

S	Starting 16-bit area where angle data is stored or angle data (3 words)
D	Area where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

Calculates the arc SIN of the value 0.7071.

DT20:	H 0
DT21:	H 0
DT22:	H7071

↓

DT10:	H45

Calculates the arc SIN of the value −0.5.

DT20:	H 1
DT21:	H 0
DT22:	H5000

↓

DT10:	H330

Description

SIN^{-1} (the arcsine) of the value specified in S, S+1, and S+2 is calculated, and the result (an angle) is stored in D.

SIN^{-1} ([S] [S+1]. [S+2]) → [D]

S: Sign
S+1: Integer value
S+2: Decimal

Set 0 for the sign in S when the data to be processed is positive, and set 1 for the sign when the data is negative.

Set the integer and decimal parts of the data each within a range of 0 to 1.0000 in S+1 and S+2.

The result of the calculation will be stored in D as a BCD value within the range 0° to 90° or 270° to 360° (in degrees).

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – If data specified in S+2, S+1 and S are not BCD value.
  – If data specified in S+2, S+1 and S are not within −1.0000 to 1.0000.
- = flag (R900B): Turns on when result of processing is recognized as "0."

F304 (BACOS) P304 (PBACOS)

BCD type Arccosine operation

Outline Triangle functions, This instruction calculates arccosine [COS^{-1} ( )].

Program example

Ladder Diagram

Boolean Address	Boolean Instruction		
10	ST	R	0
11	F304	(BACOS)	
	DT		20
	DT		10

S	Starting 16-bit area where angle data is stored or angle data (3 words)
D	Area where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

Calculates the arc COS of the value 0.8660.

DT20:	H 0
DT21:	H 0
DT22:	H8660

↓

DT10:	H30

Calculates the arc COS of the value −0.5.

DT20:	H 1
DT21:	H 0
DT22:	H5000

↓

DT10:	H120

Description

COS^{-1} (the arccosine) of the value specified in S, S+1, and S+2 is calculated, and the result (an angle) is stored in D.

COS−1 ([S][S+1]. [S+2]) → [D]

S: Sign
S+1: Integer value
S+2: Decimal

Set 0 for the sign in S when the data to be processed is positive, and set 1 for the sign when the data is negative.

Set the integer and decimal parts of the data each within a range of 0 to 1.0000 in S+1 and S+2.

The result of the calculation will be stored in D as a BCD value within the range 0° to 180° (in degrees).

Flag conditions

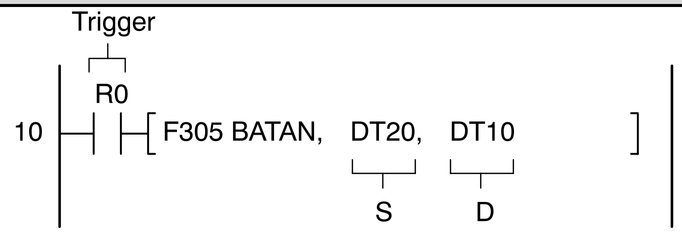
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data specified in S+2, S+1 and S are not BCD value.
  - If data specified in S+2, S+1 and S are not within −1.0000 to 1.0000.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

F305 (BATAN) P305 (PBATAN) BCD type Arctangent operation

Outline Triangle functions, This instruction calculates arctangent [TAN^{-1} ( )].

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F305 (BATAN)
	DT 20
	DT 10

S	Starting 16-bit area where angle data is stored or angle data (3 words)
D	Area where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

Calculates the arc TAN of the value 1.7321.

DT20:	H 0
DT21:	H 0
DT22:	H7321

↓

DT10:	H60

Calculates the arc TAN of the value −1.

DT20:	H 1
DT21:	H 1
DT22:	H 0

↓

DT10:	H315

Description

TAN^{-1} (the arctangent) of the value specified in S, S+1, and S+2 is calculated, and the result (an angle) is stored in D.

TAN^{-1} ([S][S+1]. [S+2]) → [D]

S: Sign
S+1: Integer value
S+2: Decimal

Set 0 for the sign in S when the data to be processed is positive, and set 1 for the sign when the data is negative.

Set the integer and decimal parts of the data each within a range of 0 to 9999.9999 in S+1 and S+2.

The result of the calculation will be stored in D as a BCD value within the range 0° to 90° or 270° to 360° (in degrees).

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data specified in S+2, S+1 and S are not BCD value.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

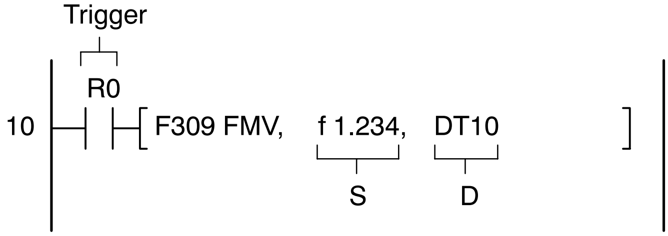
F309 (FMV) P309 (PFMV)

Floating point data move

Outline Copies floating point data (32 bits) to the specified 32-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P309 (PFMV)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	0
11	F309	(FMV)
	f	1.234
	DT	10

S	Floating point data (32 bits) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area for 32-bit area (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

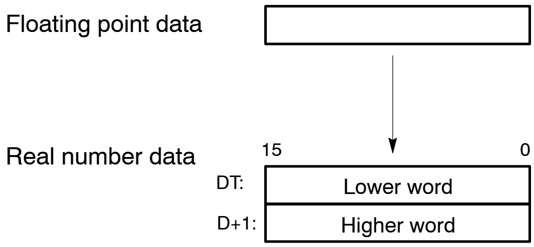
A: Available
N/A: Not Available

Explanation of example

The floating point data “f 1.234” are copied to data registers DT11 and DT10 when trigger R0 turns on.

DT10: DT11:	(f1.234)

Description
The floating point data (32 bits) specified by S is copied to the 32-bit area specified by D when the trigger turns on.



Range of real number data which can be set are as follows:

Positive: f0.0000001 to f9999999
Negative: f−9999999 to f−0.000001

Precaution during programming
For FP0, this instruction **F309 (FMV)** cannot be programmed in the interrupt program.

Flag conditions
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

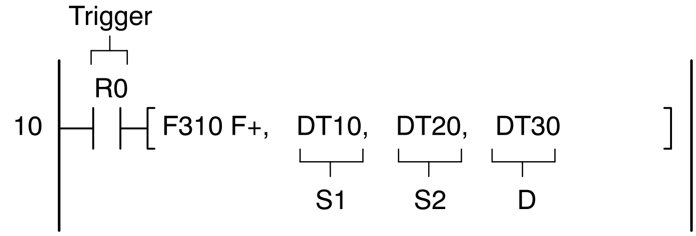
F310 (F+)
P310 (PF+)

Floating point data addition

Outline Adds two real number data items and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P310 (PF+)” is not available.

Program example

Ladder Diagram



Boolean		
Address	**Instruction**	
10	ST R	0
11	F310	(F+)
	DT	10
	DT	20
	DT	30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data (for augend)
S2	Real number data (2 words) or lower 16-bit area of 32-bit data (for addend)
D	Lower 16-bit area of 32-bit data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

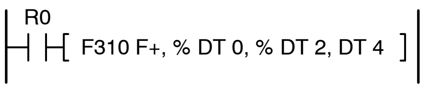
* Index modification of a real number is not possible.

Description

The real number data (2 words) specified by S1 and S2 are added together when the trigger turns on. The added result is stored in D+1 and D.

[S1+1, S1] + [S2+1, S2] → [D+1, D]

Specifying the integer device with [S1] and[S2], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
├┤ ├[ F310 F+,  DT 0,  DT 2,  % DT 4  ]
```

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Program example

The "f4.554" is stored to DT30 and DT31 when the R0 turns on.

```
 R0
├┤ ├[ F310 F+,  f1.414,  f3.14,  DT30  ]
```

The "f135.795" is stored to DT30 and DT31 when the R0 turns on.

```
 R0
├┤ ├[ F309 FMV,  f12.345  DT10        ]
    [ F309 FMV,  f12.345,  DT20       ]
    [ F310 F+,  DT10,  DT20,  DT30    ]
```

Precaution during programming

For FP0, this instruction **F310 (F+)** cannot be programmed in the interrupt program.

Flag conditions

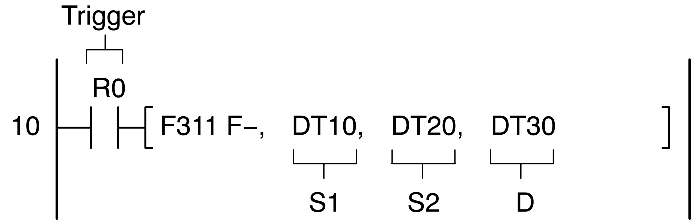
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in "S1+1 and S1" and "S2+1 and S2."
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F311 (F−) P311 (PF−) Floating point data subtraction

Outline Subtracts real number data from the minuend and stores the result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P311 (PF−)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F311		(F−)
	DT		10
	DT		20
	DT		30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data (for minuend)
S2	Real number data (2 words) or lower 16-bit area of 32-bit data (for subtrahend)
D	Lower 16-bit area of 32-bit data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

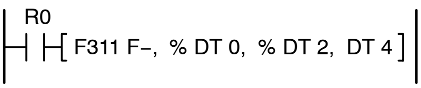
* Index modification of a real number is not possible.

Description

Subtracts the real number data (2 words) specified by S2 from the real number data (32-bit) specified by S1 when the trigger turns on. The subtracted result is stored in D+1 and D.

[S1+1, S1] − [S2+1, S2] → [D+1, D]

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
─┤ ├─[ F311 F−,  DT 0,  DT 2,  % DT 4  ]
```

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Program example

The “f0.445” is stored to DT30 and DT31 when the R0 turns on.

```
 R0
─┤ ├─[ F311 F−,  f1,  f0.555,  DT30    ]
```

The “f100.15” is stored to DT30 and DT31 when the R0 turns on.

```
 R0
─┤ ├─[ F309 FMV,  f100.1,  DT10        ]
     [ F309 FMV,  f0.05,  DT20         ]
     [ F311 F−,  DT10,  DT20,  DT30    ]
```

Precaution during programming

For FP0, this instruction **F311 (F–)** cannot be programmed in the interrupt program.

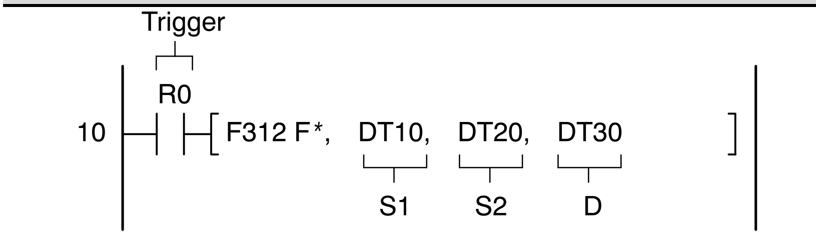
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in “S1+1 and S1” and “S2+1 and S2.”
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F312 (F*) P312 (PF*) Floating point data multiplication

Outline Multiplies two real number data items and stores the result in the specified 32-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instructions are not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F312 (F*) DT 10 DT 20 DT 30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data (for multiplicand)
S2	Real number data (2 words) or lower 16-bit area of 32-bit data (for multiplier)
D	Lower 16-bit area of 32-bit data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

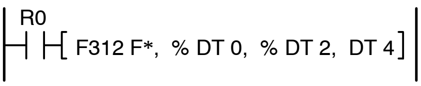
* Index modification of a real number is not possible.

Description

Multiplies the real number data (2 words) specified by S1 and the one specified by S2 when the trigger turns on.
The multiplied result is stored in D+1 and D (32-bit area).

[S1+1, S1] × [S2+1, S2] → [D+1, D]

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
─┤ ├─[ F312 F*,  DT 0,  DT 2,  % DT 4   ]
```

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Program example

The “f123.4000” is stored to DT30 and DT31 when the R0 turns on.

```
 R0
─┤ ├─[ F312 F*,  f1.234,  f100,  DT30   ]
```

Precaution during programming

For FP, this instruction **F312 (F*)** cannot be programmed in the interrupt program.

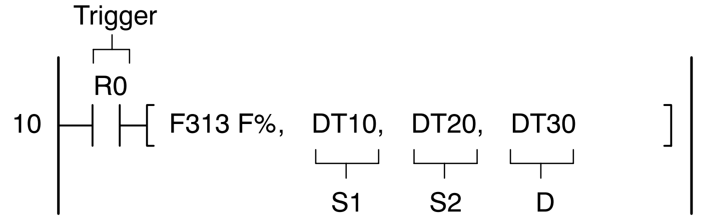
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in “S1+1 and S1” and “S2+1 and S2.”
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F313 (F%) P313 (PF%) Floating point data division

Outline Divides real number data by the divisor and stores the divided result in the specified 32-bit area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P313 (PF%)” is not available.

Program example



Address	Instruction		
10	ST	R	0
11	F313		(F%)
	DT		10
	DT		20
	DT		30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data (for dividend)
S2	Real number data (2 words) or lower 16-bit area of 32-bit data (for divisor)
D	Lower 16-bit area of 32-bit data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

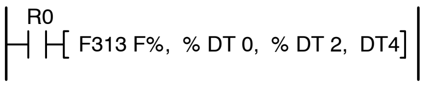
* Index modification of a real number is not possible.

Description

The real number data (2 words) specified by S1 is divided by the real number data (2 words) specified by S2 when the trigger turns on. The result is stored in D+1 and D.

[S1+1, S1] ÷ [S2+1, S2] → [D+1, D]

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
─┤ ├─[ F313 F%,  DT 0,  DT 2,  % DT 4  ]
```

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Program example

The “f5.432100” is stored to DT30 and DT31 when the R0 turns on.

```
 R0
─┤ ├─[ F312 F%,  f54.321,  f10,  DT30  ]
```

Precaution during programming

For FP, this instruction **F313 (F%)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in “S1+1 and S1” and “S2+1 and S2.”
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
  - The real number data (floating point data) for the divisor specified by S2 is “0.0”.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

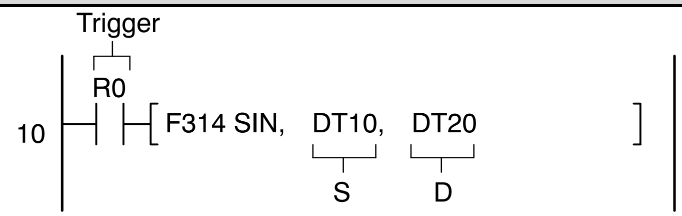
F314 (SIN) P314 (PSIN)

Floating point data Sine operation

Outline Triangle functions, This instruction calculates sine [SIN ( )].
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P314 (PSIN)" is not available.

Program example

Ladder Diagram



Address	Boolean Instruction		
10	ST	R	0
11	F314		(SIN)
	DT		10
	DT		20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

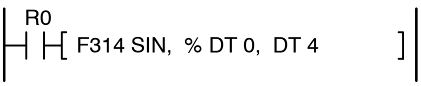
* Index modification of a real number is not possible.

Description

The SIN([S+1 and S]) of an angle data (units are radians) specified by S+1 and S is calculated and the result stored in D+1 and D.

SIN ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
  R0
├┤ ├[ F314 SIN,  DT 0,  % DT 4          ]┤
```

When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f0.4999999” is stored to DT20 and DT21 when the R0 turns on.

```
                    Radians of 30 deg.
  R0                      ↓
├┤ ├[ F314 SIN,  f0.5235987,  DT20      ]┤
```

Precautions during programming

The accuracy of the calculation decreases as the absolute value of the angle data specified in S+1 and S increases. We recommend that angle data be set within the following range:

-2π (radians) ≦ [S+1, S] ≦ 2π (radians)

For FP0, this instruction **F314 (SIN)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
  - Absolute value of S+1 and S is 52707176 or greater.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

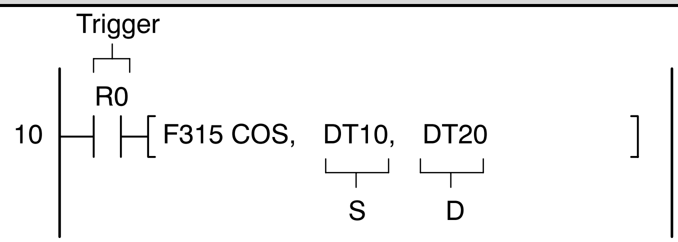
F315 (COS) P315 (PCOS)

Floating point data Cosine operation

Outline

Triangle functions, This instruction calculates cosine [COS ( )].
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P315 (PCOS)" is not available.

Program example



Ladder Diagram	Boolean Address	Boolean Instruction
(see ladder diagram above)	10	ST R 0
	11	F315 (COS)
		DT 10
		DT 20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

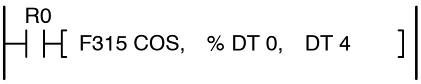
* Index modification of a real number is not possible.

Description

The COS ([S+1 and S]) of an angle data (units are radians) specified by S+1 and S is calculated and the result stored in D+1 and D.

COS ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
  R0
|-| |-[ F315 COS,   DT 0,   % DT 4      ]|
```

When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f0.7071068” is stored to DT20 and DT21 when the R0 turns on.

```
                       Radians of 45 °
                             |
  R0                         v
|-| |-[ F315 COS,  f0.7853981,  DT20   ]|
```

Precautions during programming

The accuracy of the calculation decreases as the absolute value of the angle data specified in S and S+1 increases. We recommend that angle data be set within the following range:

-2π (radians) $\leqq$ [S+1, S] $\leqq$ 2π (radians)

For FP0, this instruction **F315 (COS)** cannot be programmed in the interrupt program.

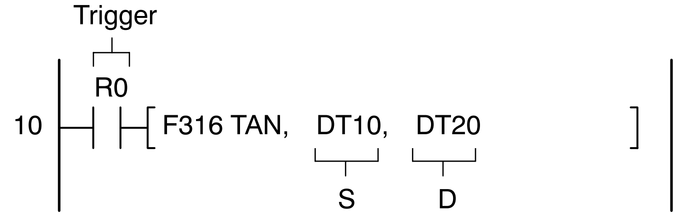
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
  - Absolute value of S+1 and S is 52707176 or greater.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

F316 (TAN) P316 (PTAN) Floating point data Tangent operation

Outline Triangle functions, This instruction calculates tangent [TAN ( )].
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P316 (PTAN)" is not available.

Program example

Ladder Diagram	Boolean: Address	Boolean: Instruction
	10 11	ST R 0 F316 (TAN) DT 10 DT 20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

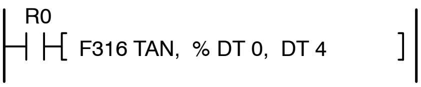
* Index modification of a real number is not possible.

Description

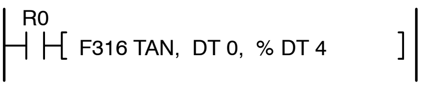
The TAN([S+1 and S]) of an angle data (units are radians) specified by S+1 and S is calculated and the result stored in D+1 and D.

TAN ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.



When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f1.732048” is stored to DT20 and DT22 when the R0 turns on.

```
                 Radians of 60 °
     R0                   ↓
|—| |—[ F316 TAN,  f1.047197,  DT20   ]|
```

Precautions during programming

The accuracy of the calculation decreases as the absolute value of the angle data specified in S+1 and S increases. We recommend that angle data be set within the following range:

-2π (radians) $\leqq$ [S+1, S] $\leqq$ 2π (radians)

For FP0, this instruction **F316 (TAN)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
  - Absolute value of S+1 and S is 52707176 or greater.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.
- = flag (R900B): Turns on when result of processing is recognized as “0.”

F317 (ASIN) P317 (PASIN) Floating point data Arcsine operation

Outline Triangle functions, This instruction calculates arcsine [SIN^{-1} ( )].
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P317 (PASIN)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F317 (ASIN) DT 10 DT 20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

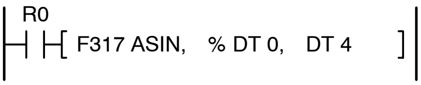
* Index modification of a real number is not possible.

Description

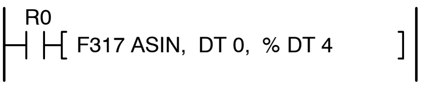
SIN of the value specified in S+1 and S is calculated, and the result [an angle (radians)] is stored in D+1 and D.

SIN^{-1} ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.



When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f0.5235986 (radians of 30 degrees)” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
-| |-[ F317 ASIN,  f0.4999999,  DT20  ]
```

Precautions during programming

D+1 and D is stored within the following range: $-\pi/2$ (radians) ≦ [D+1, D] ≦ $\pi/2$ (radians)

For FP0, this instruction **F317 (ASIN)** cannot be programmed in the interrupt program.

Flag conditions

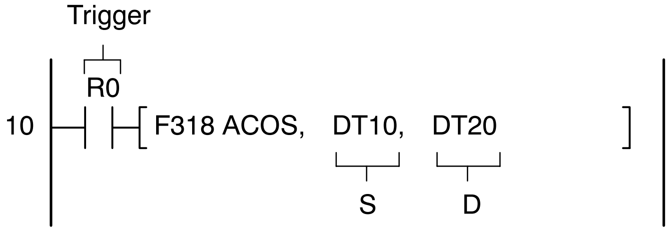
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - S+1 and S is not within the range −1.0 ≦ [S+1, S] ≦ 1.0
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F318 (ACOS) P318 (PACOS) Floating point data Arccosine operation

Outline Triangle functions, This instruction calculates arccosine [COS^{-1} ( )].
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P318 (PACOS)" is not available.

Program example

Ladder Diagram



Address	Instruction
10	ST R 0
11	F318 (ACOS)
	DT 10
	DT 20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

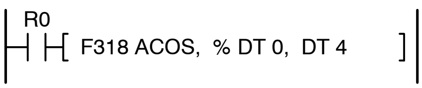
* Index modification of a real number is not possible.

Description

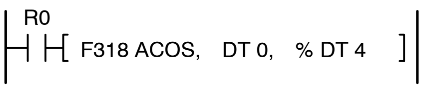
COS of the value specified in S+1 and S is calculated, and the result [an angle data (units and radians)] is stored in D+1 and D.

COS^{-1} ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.



When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f0.7853980 (radians of 45 degrees)” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
─┤ ├─[ F318 ACOS,  f0.7071069,  DT20 ]
```

Precautions during programming

D+1 and D is stored within the following range: 0.0 (radians) ≦ [D+1, D] ≦ π (radians)

For FP0, this instruction **F318 (ACOS)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - S+1 and S is not within the range −1.0 ≦ [S+1, S] ≦ 1.0
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F319 (ATAN) P319 (PATAN) Floating point data Arctangent operation

Outline Triangle functions, This instruction calculates arctangent [TAN^{-1} ( )].
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P319 (PATAN)" is not available.

Program example

Ladder Diagram

```
        Trigger
          R0
10 |---| |---[ F319 ATAN,  DT10,  DT20 ]---|
                            S      D
```

Boolean Address	Boolean Instruction	
10	ST R	0
11	F319	(ATAN)
	DT	10
	DT	20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Description

TAN^{-1} (the arctangent) of the value specified in S+1 and S is calculated, and the result [an angle data (units and radians)] is stored in D+1 and D.

TAN^{-1} ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.

```
  R0
|--| |--[ F319 ATAN,  % DT 0,  DT 4 ]--|
```

Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
  R0
|--| |--[ F319 ATAN,  DT 0,  % DT 4 ]--|
```

When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f1.047197 (radians of 60 degrees)” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
─┤ ├─[ F319 ATAN,  f1.73205,  DT20    ]
```

Precautions during programming

D+1 and D is stored within the following range: $-\pi/2$ (radians) < [D+1, D] < $\pi/2$ (radians)

For FP0, this instruction **F319 (ATAN)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

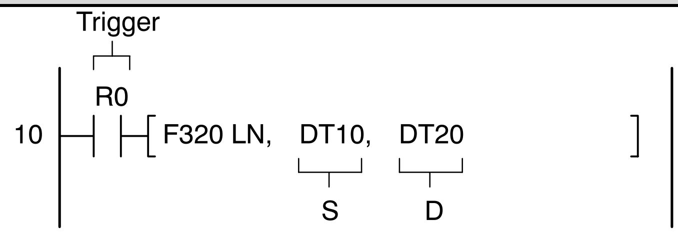
F320 (LN)
P320 (PLN)

Floating point data natural logarithm

Outline This instruction calculates a natural logarithm LN( ).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P320 (PLN)" is not available.

Program example

Ladder Diagram



Boolean

Address	Instruction		
10	ST	R	0
11	F320		(LN)
	DT		10
	DT		20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

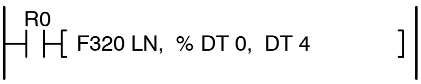
* Index modification of a real number is not possible.

Description

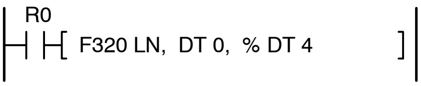
The natural logarithm LN(S+1 and S) is calculated of the data specified in S+1 and S, and the result is stored in D+1 and D.

LN ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.



Program example

The “f1.6094379” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
├┤ ├[ F320 LN,  K 5,  DT20              ]┤
```

The “f−0.3160815” is stored to DT30 and DT31 when the R0 turns on.

```
 R0
├┤ ├[ F320 LN,  f0.729,  DT30           ]┤
```

Precaution during programming

For FP0, this instruction **F320 (LN)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The S+1 and S is not greater than zero.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F321 (EXP)
P321 (PEXP)

Floating point data exponent

Outline This instruction calculates the exponent of a floating point real number EXP( ).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P321 (PEXP)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
Trigger R0 10 ⊣ ⊢⊣[F321 EXP, DT10, DT20] S D	10 11	ST R 0 F321 (EXP) DT 10 DT 20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Description

The exponent EXP(S+1 and S) is calculated from the real number data specified in S+1 and S, and the result is stored in D+1 and D.

EXP ([S+1, S]) → [D+1, D]

The calculation is performed with the exponent base (e) equal to 2.718282.

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.

```
R0
─┤ ├─[ F321 EXP,  % DT 0,  DT 4  ]
```

Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
R0
─┤ ├─[ F321 EXP,  DT 0,  % DT 4  ]
```

Program example

The “f7.389056” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
-| |-[ F321 EXP,  K 2,  DT20            ]
```

The “f221.406402” is stored to DT30 and DT31 when the R0 turns on.

```
 R0
-| |-[ F321 EXP,  f5.4,  DT30           ]
```

Precaution during programming

For FP0, this instruction **F321 (EXP)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F322 (LOG) P322 (PLOG) Floating point data logarithm

Outline This instruction calculates the logarithm of a floating point real number LOG( ).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P322 (PLOG)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
Trigger R0 10 ├─┤ ├─[F322 LOG, DT10, DT20] S D	10 11	ST R 0 F322 (LOG) DT 10 DT 20

S	Angle data (2 words) or lower 16-bit area of 32-bit data where angle data is stored
D	Lower 16-bit area of 32-bit data where calculated result is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Description

The logarithm LOG(S+1 and S) is calculated of the data specified in S+1 and S, and the result is stored in D+1 and D.

LOG ([S+1, S]) → [D+1, D]

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.

```
  R0
├┤ ├[ F322 LOG,  % DT 0,  DT 4        ]
```

Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
  R0
├┤ ├[ F322 LOG,  DT 0,  % DT 4        ]
```

When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f1.30103” is stored to DT20 and DT21 when the R0 turns on.

```
   R0
 ─┤ ├─[ F322 LOG,  K20,  DT20          ]
```

The “f0.0108932” is stored to DT30 and DT31 when the R0 turns on.

```
   R0
 ─┤ ├─[ F322 LOG,  f1.0254,  DT30      ]
```

Precaution during programming

For FP0, this instruction **F322 (LOG)** cannot be programmed in the interrupt program.

Flag conditions

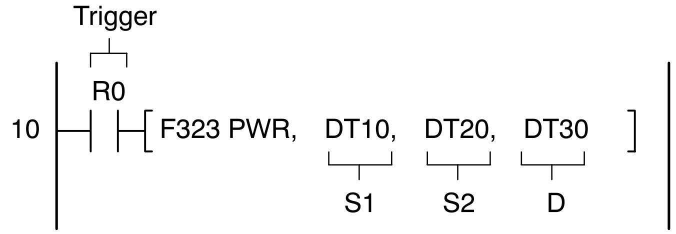
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The S+1 and S is not greater than zero.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F323 (PWR) P323 (PPWR) Floating point data power

Outline This instruction raises a floating point real number to the specified power.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P323 (PPWR)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F323		(PWR)
	DT		10
	DT		20
	DT		30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data (for multiplicand)
S2	Real number data (2 words) or lower 16-bit area of 32-bit data (for multiplier)
D	Lower 16-bit area of 32-bit data (for result)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

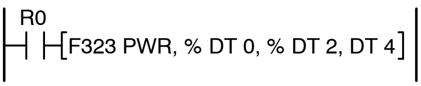
* Index modification of a real number is not possible.

Description

The real number data specified by S1+1 and S1 is raised to the power specified by the real number data of S2+1 and S2, and the result is stored in D+1 and D.

[S1+1, S] ^ [S2+1, S2] → [D+1, D]

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
├┤ ├[ F323 PWR,  DT 0,  DT 2,  % DT4 ]
```

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Program example

The “f625.0” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
├┤ ├[ F323 PWR,  K 5,  K 4,  DT20     ]
```

The “f30.51758” is stored to DT30 and DT31 when the R0 turns on.

```
 R0
├┤ ├[ F323 PWR,  f3.125,  K 3,  DT30 ]
```

Precaution during programming

For FP0, this instruction **F323 (PWR)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in “S1+1 and S1” and “S2+1 and S2.”
  - The power of negative number data is not an integer
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F324 (FSQR) P324 (PFSQR) Floating point data square root

Outline Takes the square root of the specified real number data and stores result in the specified area.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P324 (PFSQR)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
Trigger R0 10 ├┤ ├[F324 FSQR, DT10, DT20] S = DT10, D = DT20	10 11	ST R 0 F324 (FSQR) DT 10 DT 20

S	Real number data (2 words) or lower 16-bit area of 32-bit area for storing data to be calculated
D	Lower 16-bit area of 32-bit area for storing the calculated result

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Description

The square root of real number data specified by S is calculated and stored in the 32-bit area specified by D.

$\sqrt{[S1+1, S]} \rightarrow [D+1, D]$

Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.

```
 R0
├┤ ├[ F324 FSQR,  % DT 0,  DT 4        ]
```

Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
├┤ ├[ F324 FSQR,  DT 0,  % DT 4        ]
```

When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f1.41421” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
─┤ ├─[ F324 FSQR,  K 2,  DT20          ]
```

Precaution during programming

For FP0, this instruction **F324 (FSQR)** cannot be programmed in the interrupt program.

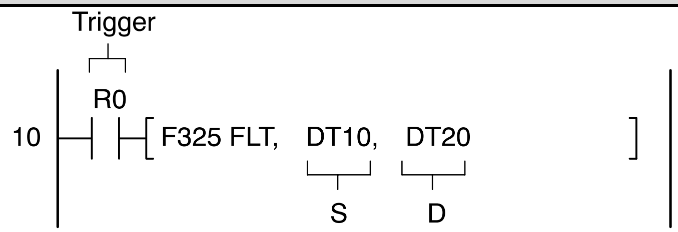
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The S+1 and S is not greater than zero.
  - If result of processing is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F325 (FLT) P325 (PFLT) 16-bit integer data → Floating point real number data

Outline Converts 16-bit integer data to floating point real number data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P325 (PFLT)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F325 (FLT) DT 10 DT 20

S	16-bit integer data or 16-bit area for storing integer data (source)
D	Lower 16-bit area of floating point real number data (destination)

Operands

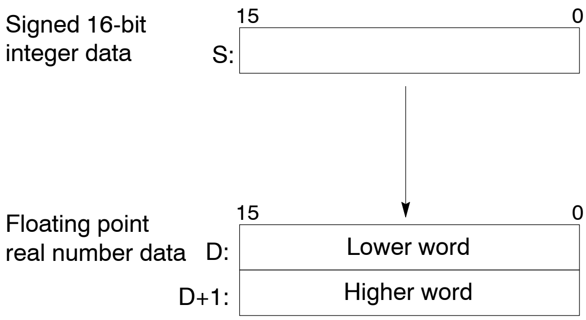
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

Description

Converts the 16-bit integer data with sign specified by S to real number data when the trigger turns on.

The converted data is stored in D.



Precaution during programming

For FP0, this instruction **F325 (FLT)** cannot be programmed in the interrupt program.

Flag conditions

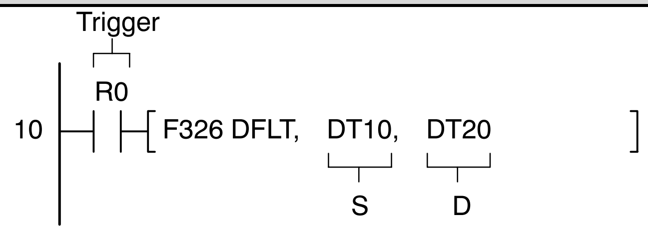
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- =lag (R900B): Turns on for an instant when the converted data is recognized as “0”.

F326 (DFLT) P326 (PDFLT) 32-bit integer data → Floating point real number data

Outline Converts 32-bit integer data to floating point real number data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P326 (PDFLT)" is not available.

Program example

Ladder Diagram



Boolean	
Address	Instruction
10	ST R 0
11	F326 (DFLT)
	DT 10
	DT 20

S	32-bit integer data or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area for 32-bit data floating point real number (destination)

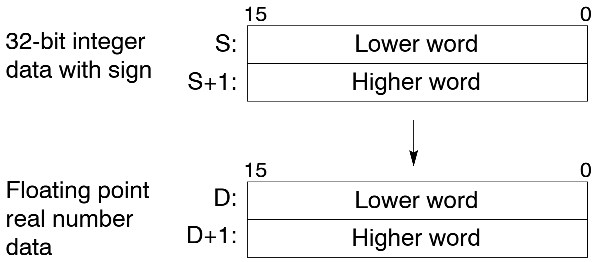
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Description

Converts the 32-bit integer data with sign specified by S to real number data when the trigger turns on. The converted data is stored in D+1 and D.



Precaution during programming

For FP0, this instruction **F326 (DFLT)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.

= flag (R900B): Turns on for an instant when the converted data is recognized as “0”.

- Carry flag (R9009): There are too many significant digits in mantissa of converted real number data.

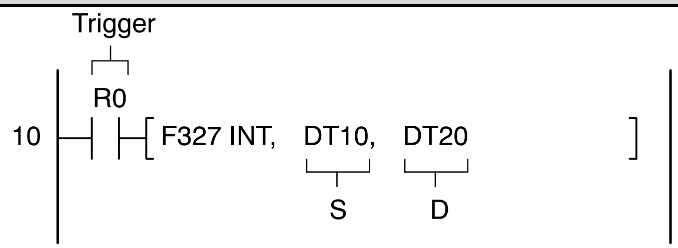
F327 (INT) P327 (PINT)

Floating point real number data → 16-bit integer data (largest integer not exceeding the floating point real number data)

Outline Converts real number data to 16-bit integer data (the largest integer not exceeding the floating point real number data).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P327 (PINT)” is not available.

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F327 (INT)
	DT 10
	DT 20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	16-bit area for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Explanation of example

When the real number data “1.234” is in DT10 and DT11, as shown below.

DT10: DT11: (f1.234)

↓

DT20: (K1)

When the real number data “−1.234” is in DT10 and DT11, as shown below.

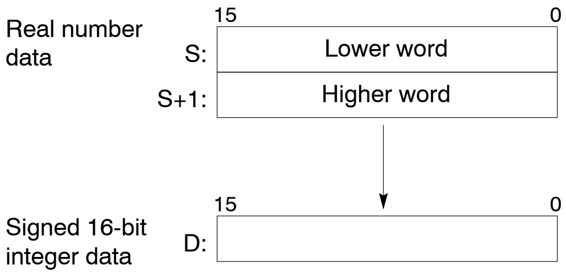
DT10: DT11: (f−1.234)

↓

DT20: (K−2)

Description

Converts real number data range: (+32767.99 to −32767.99) specified by S to signed 16-bit integer data (the largest integer not exceeding the floating point data) when the trigger turns on. The converted data is stored in D.



Precaution during programming

For FP0, this instruction **F327 (INT)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The converted data exceeds the range of 16-bit integer data.
- = flag (R900B): Turns on for an instant when the converted data is recognized as “0”.

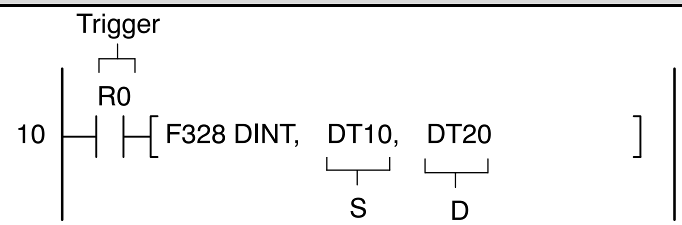
F328 (DINT) P328 (PDINT)

Floating point real number data → 32-bit integer data (largest integer not exceeding the floating point real number data)

Outline Converts real number data to 32-bit integer data (the largest integer not exceeding the floating point real number data).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P328 (PDINT)" is not available.

Program example

Ladder Diagram:



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F328		(DINT)
	DT		10
	DT		20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Explanation of example

When the real number data "12345.67" is in DT10 and DT11, as shown below.

DT10: DT11: (f12345.67)

↓

DT20: DT21: (K12345)

When the real number data "−12345.67" is in DT10 and DT11, as shown below.

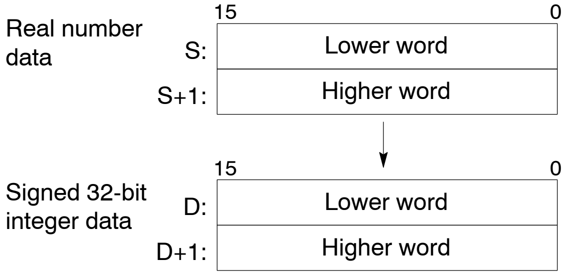
DT10: DT11: (f−12345.67)

↓

DT20: DT21: (K−12346)

Description

Converts real number data (range: +2147483000 to −2147483000) specified by S+1 and S to signed 32-bit integer data (the largest integer not exceeding the floating point data) when the trigger turns on. The converted data is stored in D+1 and D.



Precaution during programming

For FP0, this instruction **F328 (DINT)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The converted data exceeds the range of 32-bit integer data.
- = flag (R900B): Turns on for an instant when the converted data is recognized as “0”.

F329 (FIX) P329 (PFIX)

Floating point real number data → 16-bit integer data (rounding the first decimal point down to integer)

Outline Converts real number data to 16-bit integer data (rounding the first decimal point down to integer).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P329 (PFIX)” is not available.

Program example

Address	Instruction		
10	ST	R	0
11	F329		(FIX)
	DT		10
	DT		20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Explanation of example

When the real number data “1.234567” is in DT10 and DT11, as shown below.

DT10: DT11: (f1.234567)

↓

DT20: (K1)

When the real number data “−1.234567” is in DT10 and DT11, as shown below.

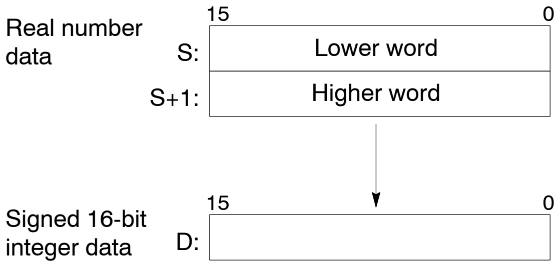
DT10: DT11: (f−1.234567)

↓

DT20: (K−1)

Description

Converts real number data (range: 32767.99 to −32768.99) specified by S to signed 16-bit integer data (rounding the first decimal point down to integer) when the trigger turns on. The converted data is stored in D.



Precaution during programming

For FP0, this instruction **F329 (FIX)** cannot be programmed in the interrupt program.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The converted data exceeds the range of 16-bit integer data.
- = flag (R900B): Turns on for an instant when the converted data is recognized as “0”.

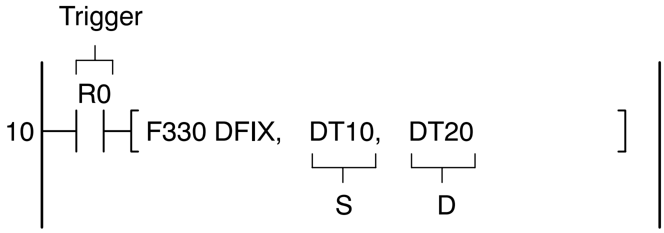
F330 (DFIX) P330 (PDFIX)

Floating point real number data → 32-bit integer data (rounding the first decimal point down to integer)

Outline Converts real number data to 32-bit integer data (rounding the first decimal point down to integer).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P330 (PDFIX)” is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	0
11	F330	(DFIX)
	DT	10
	DT	20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

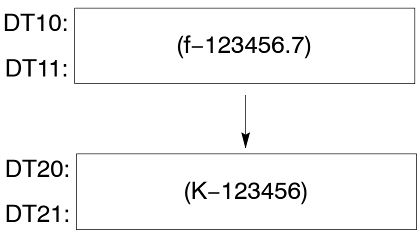
A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Explanation of example

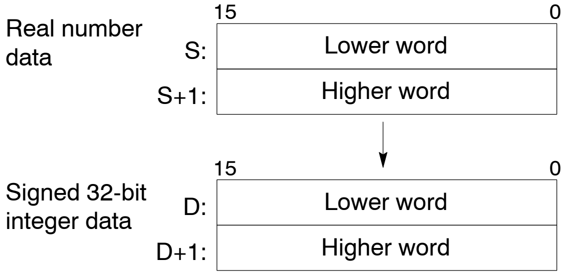
When the real number data “123456.7” is in DT10 and DT11, as shown below.

When the real number data “−123456.7” is in DT10 and DT11, as shown below.



Description

Converts real number data (range: −2,147,483,000 to 2,147,483,000) specified by S+1 and S to signed 32-bit integer data (rounding the first decimal point down to integer) when the trigger turns on. The converted data is stored in D+1 and D.



Precaution during programming

For FP0, this instruction **F330 (DFIX)** cannot be programmed in the interrupt program.

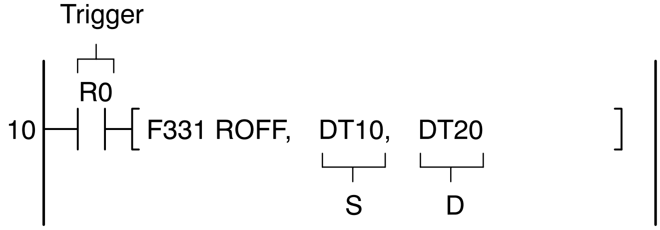
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The converted data exceeds the range of 32-bit integer data.
- = flag (R900B): Turns on for an instant when the converted data is recognized as “0”.

F331 (ROFF) P331 (PROFF) Floating point real number data → 16-bit integer data (rounding the first decimal point off to integer)

Outline Converts real number data to 16-bit integer data (rounding the first decimal point off to integer).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P331 (PROFF)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F331 (ROFF)
		DT 10
		DT 20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	16-bit area for storing converted data (destination)

Operands

Operand	Relay WX	Relay WY	Relay WR	Relay WL	Timer/Counter SV	Timer/Counter EV	Register DT	Register LD	Register FL	Index register I	Constant K	Constant H	Constant f	Index modifier	Integer device
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Explanation of example

When the real number data "1234.567" is in DT10 and DT11, as shown below.

DT10: DT11: (f1234.567)

↓

DT20: (K1235)

When the real number data "–1234.567" is in DT10 and DT11, as shown below.

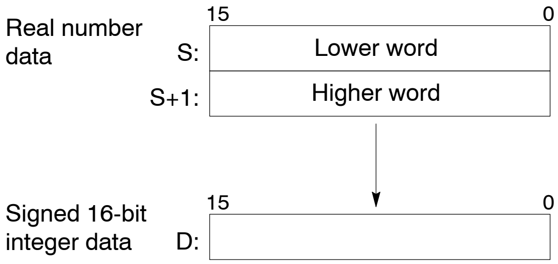
DT10: DT11: (f–1234.567)

↓

DT20: (K–1235)

Description

Converts real number data (range: +32767.49 to −32768.49) specified by S to signed 16-bit integer data (rounding the first decimal point off to integer) when the trigger turns on. The converted data is stored in D.



Precaution during programming

For FP0, this instruction **F331 (ROFF)** cannot be programmed in the interrupt program.

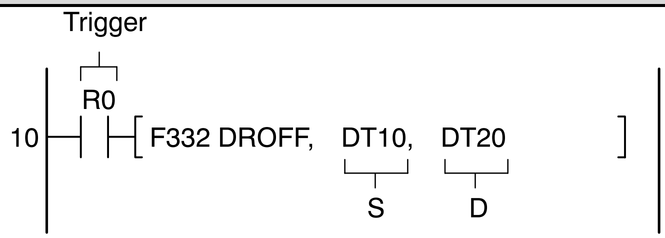
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The converted data exceeds the range of 16-bit integer data.
- = flag (R900B): Turns on for an instant when the converted data is recognized as “0”.

F332 (DROFF) P332 (PDROFF) Floating point real number data → 32-bit integer data (rounding the first decimal point off to integer)

Outline Converts real number data to 32-bit integer data (rounding the first decimal point off to integer).
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P332 (PDROFF)” is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F332 (DROFF) DT 10 DT 20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Explanation of example

When the real number data “45678.51” is in DT10 and DT11, as shown below.

DT10:
DT11: (f45678.51)

↓

DT20:
DT21: (K45679)

When the real number data “−45678.51” is in DT10 and DT11, as shown below.

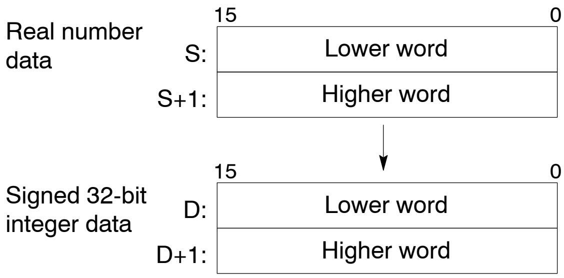
DT10:
DT11: (f−45678.51)

↓

DT20:
DT21: (K−45679)

Description

Converts real number data (range: −2,147,483,000 to 2,147,483,000) specified by S+1 and S to signed 32-bit integer data (rounding the first decimal point off to integer) when the trigger turns on. The converted data is stored in D+1 and D.



Precaution during programming

For FP0, this instruction **F332 (DROFF)** cannot be programmed in the interrupt program.

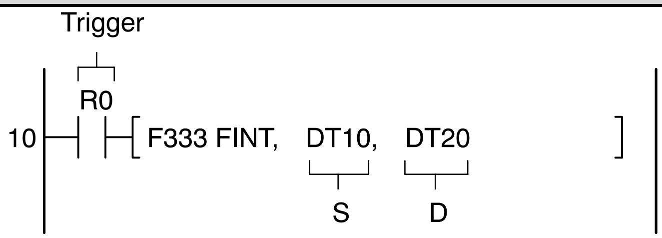
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - The converted data exceeds the range of 32-bit integer data.
- = flag (R900B): Turns on for an instant when the converted data is recognized as “0”.

F333 (FINT) P333 (PFINT) Floating point real number data rounding the first decimal point down

Outline This instruction rounds down the decimal part of real number data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P333 (PFINT)" is not available.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F333 (FINT) DT 10 DT 20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Explanation of example

When the real number data "1234.567" is in DT10 and DT11, as shown below.

DT10:
DT11: (f1234.567)

↓

DT20:
DT21: (f1234.000)

When the real number data "−1234.567" is in DT10 and DT11, as shown below.

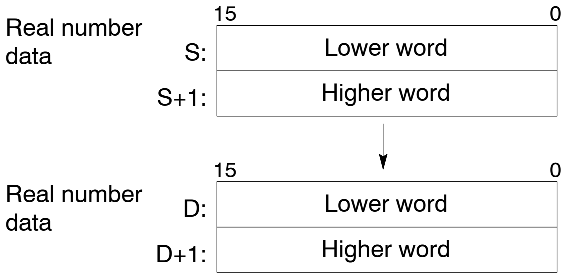
DT10:
DT11: (f−1234.567)

↓

DT20:
DT21: (f−1235.000)

Description

The decimal part of the real number data specified in S+1 and S is rounded down, and the result is stored in D+1 and D.



Precaution during programming

For FP0, this instruction **F333 (FINT)** cannot be programmed in the interrupt program.

Flag conditions

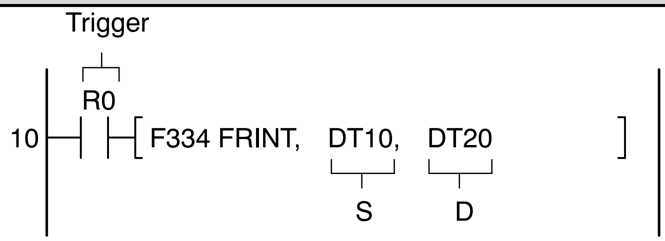
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F334 (FRINT) P334 (PFRINT) Floating point real number data rounding the first decimal point off

Outline This instruction rounds off the decimal part of real number data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P334 (PFRINT)" is not available.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction	
10	ST R	0
11	F334	(FRINT)
	DT	10
	DT	20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A*	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Explanation of example

When the real number data "1234.567" is in DT10 and DT11, as shown below.

DT10: DT11: (f1234.567)

↓

DT20: DT21: (f1235.000)

When the real number data "−1234.567" is in DT10 and DT11, as shown below.

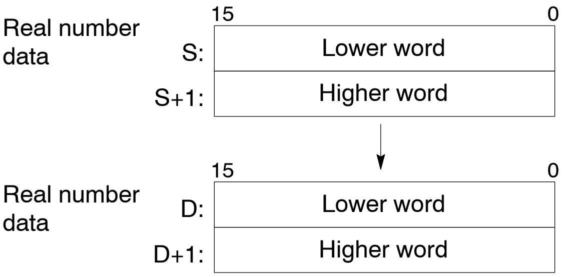
DT10: DT11: (f−1234.567)

↓

DT20: DT21: (f−1235.000)

Description

The decimal part of the real number data stored in S+1 and S is rounded off, and the result is stored in D+1 and D.



Precaution during programming

For FP0, this instruction **F334 (FRINT)** cannot be programmed in the interrupt program.

Flag conditions

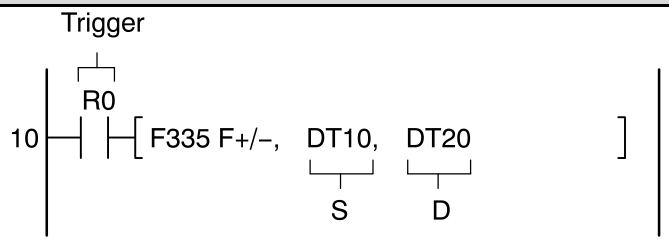
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  – The area specified using the index modifier exceeds the limit.
  – If data other than real number data is specified in S+1 and S.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F335 (F+/−) P335 (PF+/−)

Floating point real number data sign changes (negative/positive conversion)

Outline This instruction changes the sign of real number data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P335 (PF+/−)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10 11	ST R 0 F335 (F+/−) DT 10 DT 20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Explanation of example

When the real number data "–60000.00" is in DT10 and DT11, as shown below.

DT10:
DT11: (f−60000.00)

↓

DT20:
DT21: (f60000.00)

When the real number data "–30000.00" is in DT10 and DT11, as shown below.

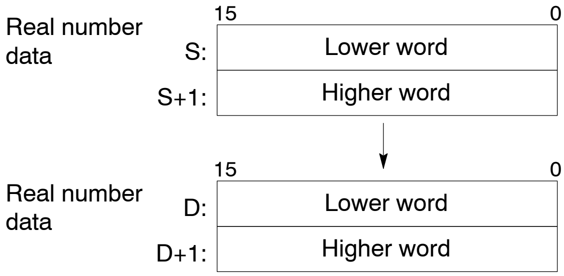
DT10:
DT11: (f−30000.00)

↓

DT20:
DT21: (f30000.00)

Description

The real number data stored in S+1 and S is changed sign bit, and the result is stored in D+1 and D.



Precaution during programming

For FP0, this instruction **F335 (F+/–)** cannot be programmed in the interrupt program.

Flag conditions

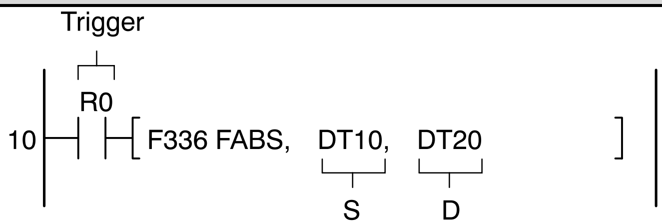
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The area specified using the index modifier exceeds the limit.
  - – If data other than real number data is specified in S+1 and S.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F336 (FABS) P336 (PFABS)

Floating point real number data absolute

Outline Takes absolute value of real number data.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P336 (PFABS)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F336 (FABS)
		DT 10
		DT 20

S	Real number data (2 words) or lower 16-bit area of 32-bit data (source)
D	Lower 16-bit area of 32-bit data for storing converted data (destination)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Explanation of example

When the real number data "1234.567" is in DT10 and DT11, as shown below.

DT10: DT11: (f1234.567)

↓

DT20: DT21: (f1234.567)

When the real number data "−1234.567" is in DT10 and DT11, as shown below.

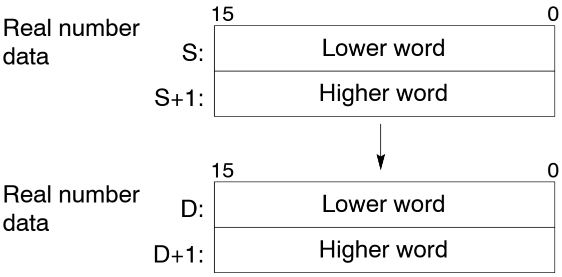
DT10: DT11: (f−1234.567)

↓

DT20: DT21: (f1234.567)

Description

Takes the absolute value of real number data specified by S when the trigger turns on. The result (absolute value) is stored in D+1 and D.



Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.

When the constant K is specified in [S], the operations are the same as when a integer device is specified.

Precaution during programming

For FP0, this instruction **F336 (FABS)** cannot be programmed in the interrupt program.

Flag conditions

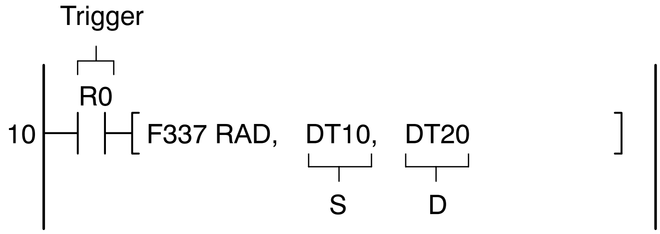
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F337 (RAD) P337 (PRAD)

Floating point real number data conversion of angle units (Degrees → Radians)

Outline This instruction converts the units of an angle from degrees to radians. For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction “P337 (PRAD)” is not available.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10 11	ST R 0 F337 (RAD) DT 10 DT 20

S	Angle data (degrees) (2 words) or lower 16-bit area of 32-bit data
D	Lower 16-bit area of 32-bit data for storing converted data

Operands

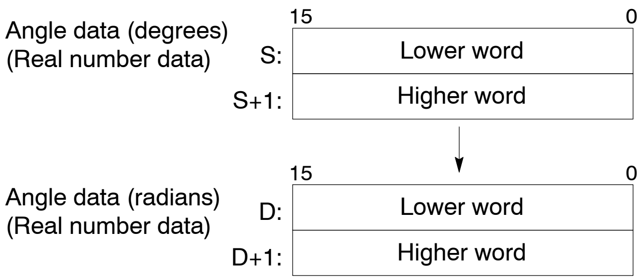
Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

* Index modification of a real number is not possible.

Description

The data in degrees of an angle specified in S+1 and S is converted to radians (real number data) and the result is stored in D+1 and D.



Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.

When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f0.7853981” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
├┤ ├[ F337  RAD,  f45,  DT20            ]
```

Precaution during programming

For FP0, this instruction **F337 (RAD)** cannot be programmed in the interrupt program.

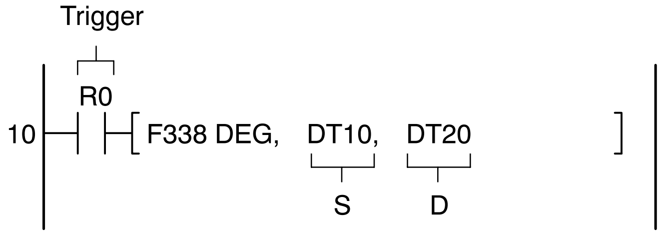
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F338 (DEG) P338 (PDEG) Floating point real number data conversion of angle units (Radians → Degrees)

Outline Converts the units of an angle from radians to degrees.
For the FP0R/FPΣ/FP−X/FP0/FP−e, the P type high−level instruction "P338 (PDEG)" is not available.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F338 (DEG)
		DT 10
		DT 20

S	Angle data (radians) (2 words) or lower 16-bit area of 32-bit data
D	Lower 16-bit area of 32-bit data for storing converted data

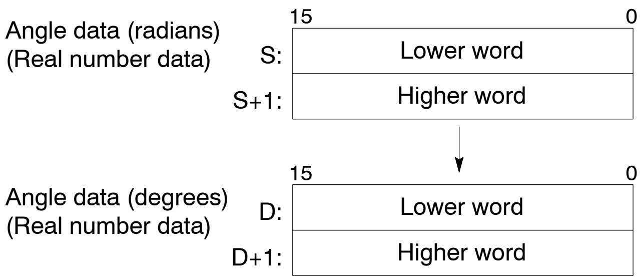
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Description

The angle data in radians (real number data) specified in S+1 and S is converted to angle data in degrees and the result is stored in D+1 and D.



Specifying the integer device with [S], the integer data is internally converted to real numbers before operations continue.

```
 R0
─┤ ├─[ F338 DEG,  % DT 0,  DT 4        ]─
```

Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
─┤ ├─[ F338 DEG,  DT 0,  % DT 4        ]─
```

When the constant K is specified in S, the operations are the same as when a integer device is specified.

Program example

The “f30.00000” is stored to DT20 and DT21 when the R0 turns on.

```
 R0
─┤ ├─[ F338  DEG,  f0.5235987,  DT20  ]─
```

Precautions during programming

When the constant or integer device is specified in S, the integer device cannot be set in D.

For FP0, this instruction **F338 (DEG)** cannot be programmed in the interrupt program.

Flag conditions

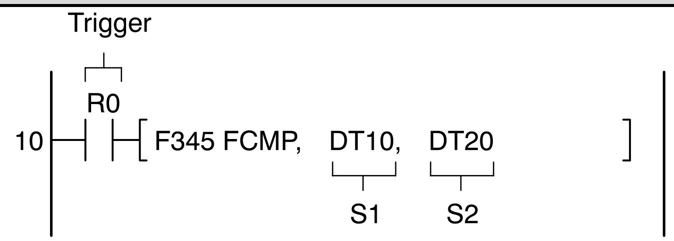
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in S+1 and S.
  - If result of processing is outside integer range when integer device specified in D+1 and D.
- = flag (R900B): Turns on when result of processing is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F345 (FCMP) P345 (PFCMP) Floating point real number data comparison

Outline Compares one real number data (floating point data) item with another.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F345 (FCMP)
	DT 10
	DT 20

S1	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared
S2	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A

A: Available

* Index modification of a real number is not possible.

Description

Compares the real number data (floating point data) specified by S1 with that specified by S2 when the trigger turns on. The comparison result is stored in special internal relays R9009, and R900A to R900C.

The following table lists the states of the carry flag (R9009), > flag (R900A), = flag (R900B), and < flag (R900C), depending on the relative sizes of (S1+1, S1) and (S2+1, S2).

Comparison between (S1+1, S1) and (S2+1, S2)	Flag R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (carry flag)
(S1+1, S1) < (S2+1, S2)	off	off	on	↕
(S1+1, S1) = (S2+1, S2)	off	on	off	off
(S1+1, S1) > (S2+1, S2)	on	off	off	↕

"↕": turns on or off according to the conditions

Specifying the integer device with [S1] and [S2], the integer data is internally converted to real numbers before operations continue.

When the constant K is specified in S1 and S2, the operations are the same as when a integer device is specified.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data are specified in (S1+1, S1) and (S2+1, S2).

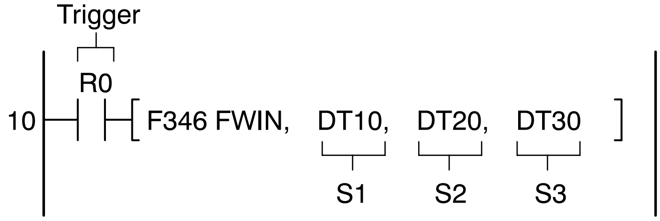
F346 (FWIN)
P346 (PFWIN)

Floating point real number data band comparison

Outline Compares one real number data item with the data band specified by two other real number data items.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F346		(FWIN)
	DT		10
	DT		20
	DT		30

S1	Real number data (2 words) or lower 16-bit area of 32-bit data to be compared
S2	Real number data (2 words) or lower 16-bit area of 32-bit data for lower limit
S3	Real number data (2 words) or lower 16-bit area of 32-bit data for upper limit

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A

A: Available

* Index modification of a real number is not possible.

Description

Compares the floating point real number data specified by S1 with the data band specified by S2 and S3, when the trigger turns on. This instruction checks whether S1 is in the data band between S2 (lower limit) and S3 (upper limit), larger than S3, or smaller than S2. The comparison result is stored in special internal relays R900A, R900B, and R900C.

The following table lists the states of the carry flag (R9009), > flag (R900A), = flag (R900B), and < flag (R900C).

Comparison between (S1+1, S1), (S2+1, S2) and (S3+1, S3)	Flag			
	R900A (> flag)	R900B (= flag)	R900C (< flag)	R9009 (Carry flag)
(S1+1, S1) < (S2+1, S2)	off	off	on	X
(S1+1, S1) ≦ (S3+1, S3) and (S2+1, S2) ≦ (S1+1, S1)	off	on	off	X
(S3+1, S3) < (S1+1, S1)	on	off	off	X

“X”: Not changed

Specifying the integer device with [S1], [S2] and [S3], the integer data is internally converted to real numbers before operations continue.

When the constant K is specified in S1, S2 and S3, the operations are the same as when a integer device is specified.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data are specified in “S1+1, S1”, “S2+1, S2”, and “S3+1, S3”.
  - (S2+1, S2) > (S3+1, S3).
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data are specified in “S1+1, S1”, “S2+1, S2”, and “S3+1, S3”.
  - (S2+1, S2) > (S3+1, S3).

F347 (FLIMT)
P347 (PFLIMT)

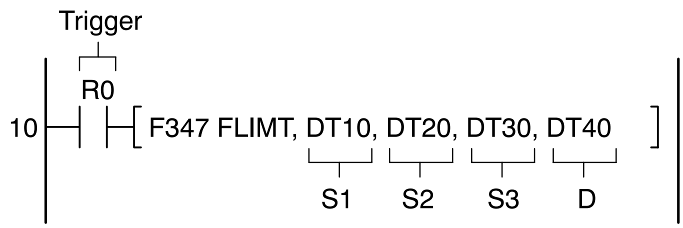
Floating point data upper and lower limit control

Outline

This instruction carries out upper and lower limit control for real number data.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F347		(FLIMT)
	DT		10
	DT		20
	DT		30
	DT		40

S1	The area where the lower limit is stored or the lower limit data. (2 words)
S2	The area where the upper limit is stored or the upper limit data. (2 words)
S3	The area where the input value is stored or the input value data. (2 words)
D	The area where the output value is stored. (2 words)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Description

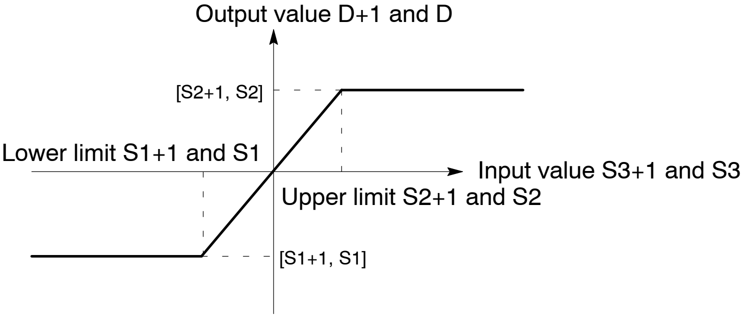
The output value (real number data) stored in the area specified by D is controlled based on whether or not the input value (real number data) specified by S3 falls within the range bounded by the upper and lower limits (real number data) set in S1 and S2.

The output value is determined based on the following conditions:

When the lower limit S1+1 and S1 are greater than the input value S3+1 and S3, the lower limit value S1+1 and S1 stored in D+1 and D as the output value.

When the upper limit S2+1 and S2 are less than the input value S3+1 and S3, the upper limit value S2+1 and S2 are stored in D+1 and D as the output value.

When Lower limit S1+1 and S1 ≦ Input value S3+1 and S3 ≦ Upper limit S2+1 and S2, the input value S3+1 and S3 stored in D+1 and D as the output value.



Specifying the integer device with [S1], [S2] and [S3], the integer data is internally converted to real numbers before operations continue.

```
 R0
─┤ ├─[ F347 FLIMT, % DT10, % DT20, DT30, DT40 ]
```

Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
─┤ ├─[ F347 FLIMT, DT10, DT20, DT30, % DT40 ]
```

When the constant K is specified in S1, S2 and S3, the operations are the same as when a integer device is specified.

Flag conditions

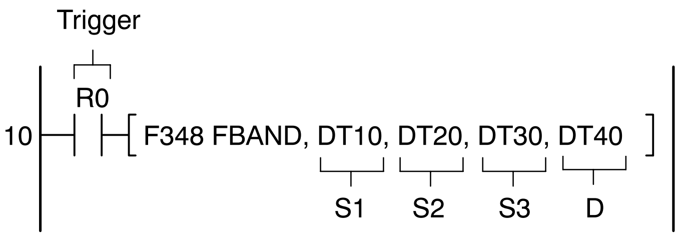
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in (S1+1 and S1), (S2+1 and S2) and (S3+1 and S3).
  - (S1+1, S1) > (S2+1, S2).
  - If result of operating is outside integer range when integer device is specified in D+1 and D.
- = flag (R900B): Turns on when result of operating is within the range of the upper and lower limits.

F348 (FBAND) P348 (PFBAND) Floating point real number data deadband control

Outline This instruction carries out dead-band control for real number data.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F348(FBAND)		
	DT		10
	DT		20
	DT		30
	DT		40

S1	The area where the lower limit is stored or the lower limit data. (2 words)
S2	The area where the upper limit is stored or the upper limit data. (2 words)
S3	The area where the input value is stored or the input value data. (2 words)
D	The area where the output value is stored. (2 words)

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Description

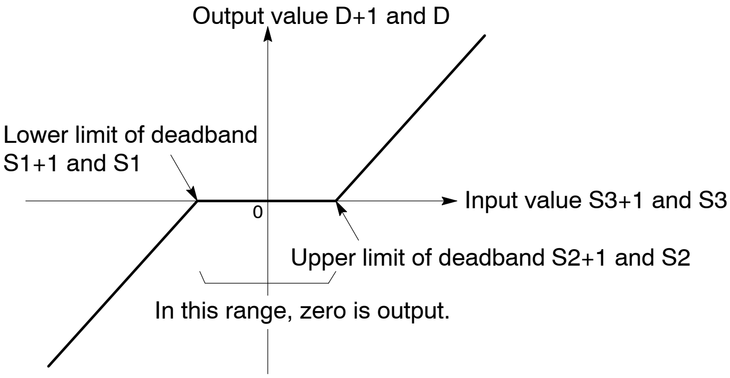
The output value (real number data) stored in the area specified by D is controlled based on whether or not the input value (real number data) specified by S3 falls within the dead-band bounded by the upper and lower limits (real number data) set in S1 and S2.

The output value is determined based on the following conditions:

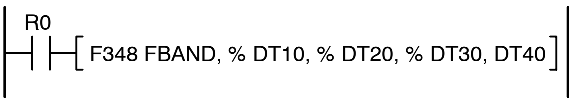
When the lower limit S1+1 and S1 are greater than the input value S3+1 and S3, the input value S3+1 and S3 minus the lower limit value S1+1 and S1 are stored in D+1 and D as the output value.

When the upper limit S2+1 and S2 are less than the input value S3+1 and S3, the input value S3+1 and S3 minus the upper limit value S2+1 and S2 are stored in D+1 and D as the output value.

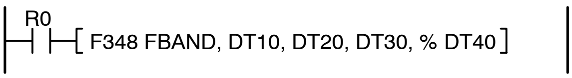
When Lower limit S1+1 and S1 ≦ Input value S3+1 and S3 ≦ Upper limit S2+1 and S2, 0.0 is stored in D+1 and D as the output value.



Specifying the integer device with [S1], [S2] and [S3], the integer data is internally converted to real numbers before operations continue.



Specifying the integer device with [D], the real numbers are automatically converted into integer data.



When the constant K is specified in S1, S2 and S3, the operations are the same as when a integer device is specified.

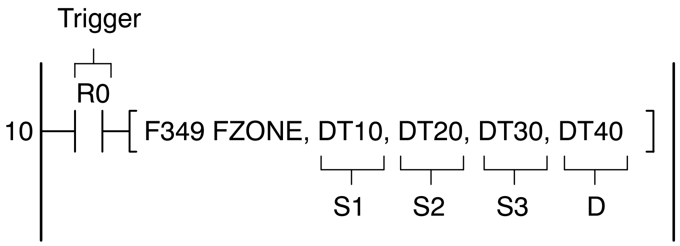
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in "S1+1, S1", "S2+1, S2" and "S3+1, S3".
  - "S1+1, S1" > "S2+1, S2".
  - If result of operating is outside integer range when integer device is specified in "D+1, D".
- = flag (R900B): Turns on when result of operating is within the range of the upper and lower limits.
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F349 (FZONE) P349 (PFZONE) Floating point real number data zone control

Outline This instruction carries out zone control for real number data.

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10	ST R 0
	11	F349 (FZONE)
		DT 10
		DT 20
		DT 30
		DT 40

S1	Area where negative bias value is stored or negative bias value data (double words)
S2	Area where positive bias value is stored or positive bias value data (double words)
S3	Area where input value is stored or input value data (double words)
D	Area (double words) where output value is stored

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S2	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
S3	A	A	A	A	A	A	A	A	A	A	A	A	A	A*	A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available
* Index modification of a real number is not possible.

Description

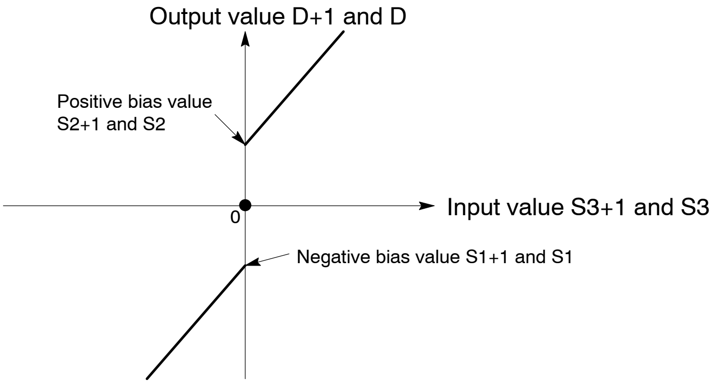
The bias value specified by S1 or S2 is added to the input value (real number data) specified by S3, and the output value is stored in the area specified by D.

The output value is determined by the following conditions:

When the input value S3+1 and S3 are less than 0.0, the input value S3+1 and S3 plus the negative bias value S1+1 and S1 are stored in D+1 and D as the output value.

When the input value S3+1 and S3 are equals 0.0, zero is stored in D+1 and D as the output value.

When the input value S3+1 and S3 are greater than 0.0, the input value S3+1 and S3 plus the positive bias value S2+1 and S2 are stored in D+1 and D as the output value.



Specifying the integer device with [S1], [S2] and [S3], the integer data is internally converted to real numbers before operations continue.

```
 R0
─┤ ├─[ F349 FZONE, % DT10, % DT20, % DT30, DT40 ]
```

Specifying the integer device with [D], the real numbers are automatically converted into integer data.

```
 R0
─┤ ├─[ F349 FZONE, DT10, DT20, DT30, % DT40 ]
```

When the constant K is specified in S1, S2 and S3, the operations are the same as when a integer device is specified.

Flag conditions

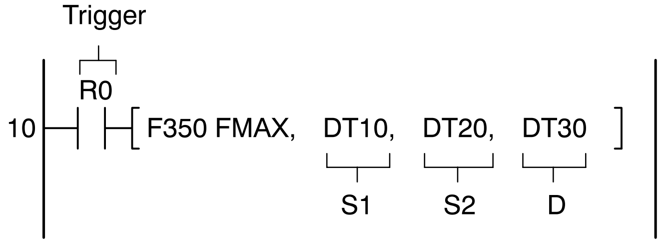
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - If data other than real number data is specified in “S1+1, S1”, “S2+1, S2” and “S3+1, S3.”
  - If result of operating is outside integer range when integer device is specified in “D+1, D”.
- = flag (R900B): Turns on when input value is recognized as “0.”
- Carry flag (R9009): Turns on for an instant when the result is overflowed.

F350 (FMAX) P350 (PFMAX) Maximum value search in floating point real number data table

Outline Searches for a maximum value in a table of real number data.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F350 (FMAX)
	DT 10
	DT 20
	DT 30

S1	Starting 16-bit area for storing the real number data
S2	Ending 16-bit area for storing the real number data
D	Starting 16-bit area for storing maximum value and relative address (3 words)

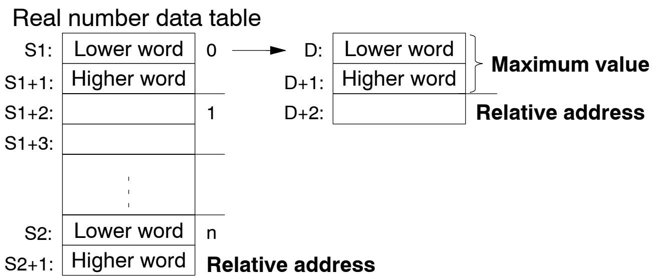
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

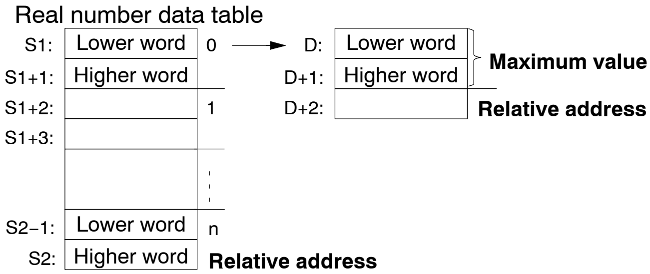
A: Available
N/A: Not Available

Description

This instruction searches for the maximum value in the real number data table between the area selected with S1 and the area selected with S2, and stores it in the area selected with D+1 and D. The address relative to S1 is stored in D+2.



If S2 specifies a higher word of real number data, processing will take place over the same area as if the lower word had been specified.



If there are several values which are a maximum value, the relative address of the first value found searching from S1 is stored in D+2.

Flag conditions

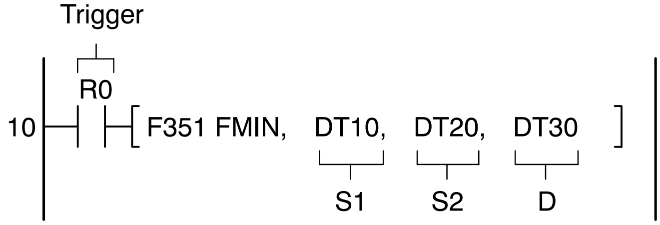
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.
  - The real number data is outside possible operating range.

F351 (FMIN) P351 (PFMIN) Minimum value search in floating point real number data table

Outline Searches for a minimum value in a table of real number.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F351 (FMIN)
	DT 10
	DT 20
	DT 30

S1	Starting 16-bit area for storing the real number data
S2	Ending 16-bit area for storing the real number data
D	Starting 16-bit area for storing minimum value and relative address (3 words)

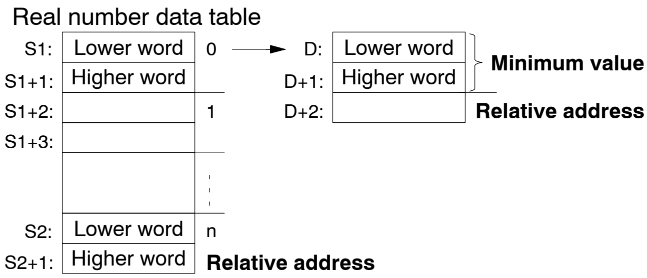
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

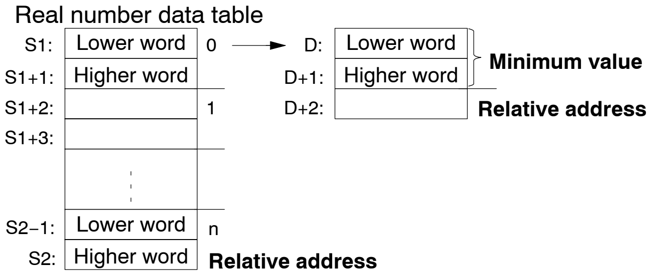
A: Available
N/A: Not Available

Description

This instruction searches for the minimum value in the real number data table between the area selected with S1 and the area selected with S2, and stores it in the area selected with D+1 and D. The address relative to S1 is stored in D+2.



If S2 specifies a higher word of real number data, processing will take place over the same area as if the lower word had been specified.



If there are several values which are a minimum value, the relative address of the first value found searching from S1 is stored in D+2.

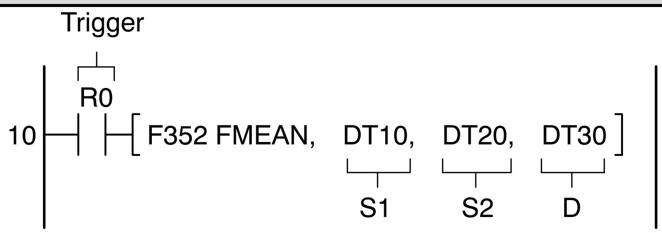
Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.
  - The real number data is outside possible operating range.

F352 (FMEAN) P352 (PFMEAN) Total and mean numbers calculation in floating point real number data table

Outline Calculates the total and mean numbers in the specified real number data table

Program example

Ladder Diagram	Boolean Address	Boolean Instruction
	10 11	ST R 0 F352 (FMEAN) DT 10 DT 20 DT 30

S1	Starting 16-bit area for storing the real number data
S2	Ending 16-bit area for storing the real number data
D	Starting 16-bit area for storing total and mean numbers (4 words)

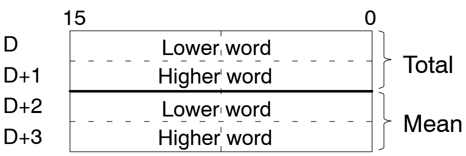
Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

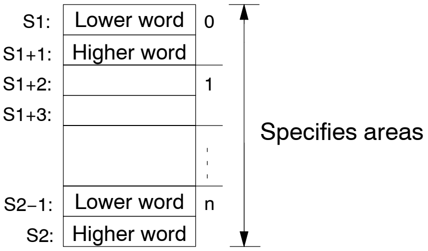
Description

The total value and the average value of the real number data from the area selected with S1 to the area selected with S2 are obtained and stored in the area selected with D.



If S2 specifies a higher word of real number data, processing will take place over the same area as if the lower word had been specified.

Real number data table



Precautions during programming

Even if D+2 overflows the selected area, it will still be stored, and this may corrupt the data in the leading part of the other area. (An area overflow check is not performed.)

Flag conditions

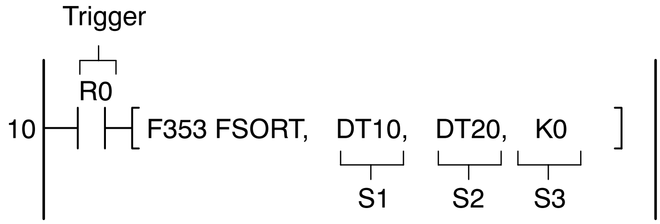
- Error flag (R9007): Turns on and stays on when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.
  - The real number data is outside possible operating range.
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.
  - The real number data is outside possible operating range.
- Carry flag (R9009): Turns on for an instant when overflows/underflows while calculating.

F353 (FSORT) P353 (PFSORT) Sort data in real number floating point data table

Outline Sorts a string of real number data (in smaller or larger number order).

Program example

Ladder Diagram



Boolean			
Address	Instruction		
10	ST	R	0
11	F353		(FSORT)
	DT		10
	DT		20
	K		0

S1	Starting 16-bit area of sort data (2 words)
S2	Ending 16-bit area of sort data (2 words)
S3	Constant or area where sort condition is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S1	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S2	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
S3	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

Description

The real number data from the area specified by S1 to the area specified by S2 are sorted in ascending order (the smallest word is first) or descending order (the largest word is first) depending on the condition set with S3.

If S1 = S2, sorting does not take place.

The sort condition is specified as follows in S3:

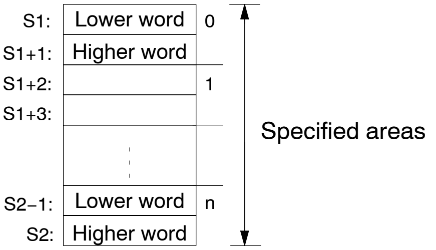
– K0: Ascending order

– K1: Descending order

Double sorting is used for the sorting method. Data is sorted from S1 to S2 in order following the sorting procedure. Note that the number of word comparisons increases in proportion to the square of the number of words, thus more time will be required for execution when there are a large number of words.

If S2 specifies a higher word of real number data, processing will take place over the same area as if the lower word had been specified.

Real number data table



Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - S1 > S2.
  - The areas of S1 and S2 are different.
  - The real number data is outside possible operating range.

F354 (FSCAL) P354 (PFSCAL) Scaling of real number data

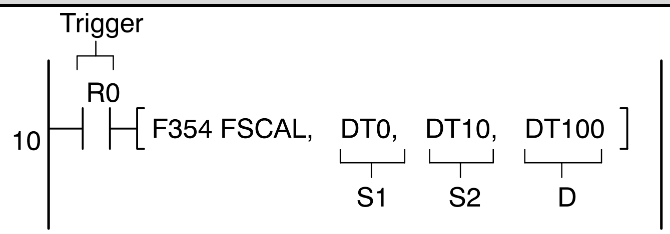
Availability
FP2/FP2SH FP−X (V1.13 or more) FPΣ 32k/FP0R

Outline Scaling(linearization) on a real number data table is performed, and the output (Y) to an input value (X) is calculated.

With FP2/FP2SH, this function is available from Ver. 1.50 or later.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction
10	ST R 0
11	F354 (FSCAL)
	DT 0
	DT 10
	DT 100

S1	Real numerical value or area which shows the input value (X)
S2	Head area of the data table used for scaling
D	Area in which the output value (Y) stored

Operands

Operand	Relay				Timer/ Counter		Register			Index register	Constant				Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	H	f		
S1	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	A	A
S2	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	A	A

A: Available
N/A: Not Available

Explanation of example

The output value Y is calculated for the input value stored in DT0 referring to the data table which starts with DT10, and the result is stored in DT100.

Description

1) Scaling (linearization) is performed according to the data table of the real number specified by [S2] in the inputted real numerical value [S1], and an output value is stored in [D].

2) An output value is calculated by searching the linear section of an input value [S1], and computing the linear interpolation between these two points from the linear table specified by [S2].

When the specified input value is out of the registration range of an linear table, the output value (Y0 or Yn) over a starting point (x0) or an ending point (xn) is stored, respectively.

[S1] <= x_0 −−− [D] = y_0

[S1] >= x_n −−− [D] = y_n

3) The linear table [S2] must be having the section of two or more points registered.
Moreover, the linear table must be registered in ascending order, from small to large number of the x sequences.
2 <= Registration mark (m) <= 99 (m=n+1)
xt−1 < xt (1 <= t <= n)

4) When the distance between two points of a scaling table is very large, an operation error occurs.
for example)
Point1: (x_0,y_0)=(HFF000000, HFF000000) =($-1.7*10^{34}$, $-1.7*10^{34}$)
Point2: (x_1,y_1)=(H7F000000, H7F000000) =($+1.7*10^{34}$, $+1.7*10^{34}$)

5) The error of an output result is proportional to the distance between two points of a scaling table.

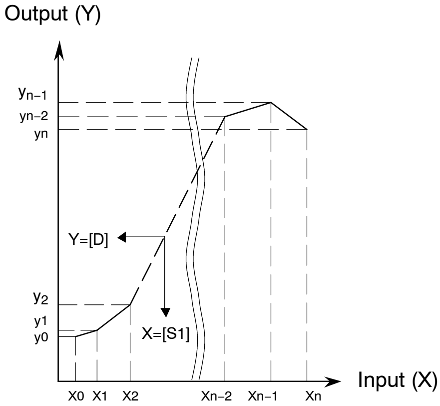
6) When the integer modifier is specified to be an input value [S1], scaling processing is performed after changing it into a real numerical value.

7) An output result is changed into an integer value and stored when the integer modifier is specified to be an output value [S2].

Data table(linear table) (S2)

S2 = DT10, n = K10

S2:	Registration mark (m)	DT10
S2+1:	x0	DT11
S2+2:	(real numerical value)	DT12
S2+3:	x1	DT13
S2+4:	(real numerical value)	DT14
S2+2n+1:	xn	DT29
S2+2n+2:	(real numerical value)	DT30
S2+2n+3:	y0	DT31
S2+2n+4:	(real numerical value)	DT32
S2+2n+5:	y1	DT33
S2+2n+6:	(real numerical value)	DT34
S2+4n+3:	yn	DT49
S2+4n+4:	(real numerical value)	DT50



Flag conditions

• Error flag (R9007)(R9008):

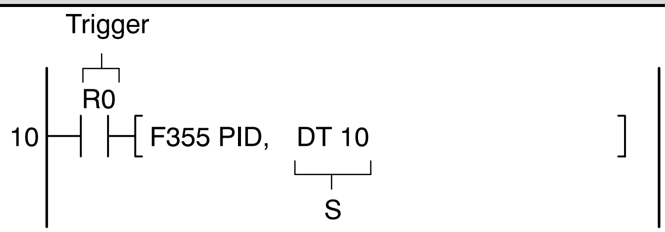
– It turns on, when the specified address using the index modifier exceeds a limit.
– It turns on, when a non−real number value is inputted into [S1].
– In the registration mark of [S2], it turns on at the time of m<2 or m>99.
– It turns on, when a non−real number value is specified to be the real numerical value (xt, yt) specified in [S2].
– It turns on, when the linear table of [S2] is not registered in ascending order of the x−sequence.
– It turns on, when the linear table of [S2] exceeds the area.
– It turns on, when the overflow (operaion is unable) occurs in the operation of scaling.
– It turns on, when integer modifier specification is carried out for [D] and an output result exceeds the integer range.

F355 (PID) PID processing

Outline This instruction carries out PID processing using data table.

Program example

Ladder Diagram



Address	Instruction		
10	ST	R	0
11	F355		(PID)
	DT		10

S	Starting number of PID parameter area (30 words)

Operands

Operand	Relay				Timer/Counter		Register	Index register	Constant			Index modifier
	WX	WY	WR	WL	SV	EV	DT	I	K	H	f	
S	N/A	N/A	N/A	N/A	N/A	N/A	A	N/A	N/A	N/A	N/A	N/A

A: Available
N/A: Not Available

Description

PID processing is performed to hold the measured value specified by S+2 at the set value S+1, and the result is output to S+3.

Derivative control or proportional−derivative control can be selected for the PID processing mode.

Set the PID processing coefficients (proportional gain, integral time and derivative time) and the processing mode and cycle in the parameter table. PID processing will be performed based on these settings.

Types of PID processing

Reverse operation and forward operation

When a process has been changed, the vertical direction of the output can be selected.

If the measured value drops, “Reverse operation” is specified to boost the output (heating, etc.).

If the measured value increases, “Forward operation” is specified to boost the output (cooling, etc.).

Derivative type (PI−D) / Proportional−derivative type (I−PD)

Generally, with “derivative PID control”, when a set value is changed, there is increased fluctuation in the output, but convergence is faster.

Generally, with “proportional−derivative PID control”, when a set value is changed, there is less fluctuation in the output, but convergence is slower.

☛ next page

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - – The value set for the parameter is out of range.
  - – The area specified using the index modifier exceeds the limit.

Parameter table settings

[S]			Control mode
[S+1]			Set value (SP)
[S+2]			Measured value (PV)
[S+3]			Output value (MV)
[S+4]			Output lower limit
[S+5]			Output upper limit
[S+6]			Proportional gain (Kp)
[S+7]			Integral time (Ti)
[S+8]			Derivative time (Td)
[S+9]			Control cycle (Ts)
[S+10]			Auto-tuning progress
[S+11]			PID processing work area (*)
⁓			
[S+29]			

* For the FP0 use the 20 words, [S+11] to [S+30], as the work area.

Explanation of parameters

① **Control mode** . . . . [S]

Select the type of PID processing and auto-tuning on/off with the H constants.

Control mode		Value of [S]	
		Auto−tuning when not executed	Auto−tuning when executed
Derivative type	Reverse operation	H0	H8000
	Forward operation	H1	H8001
Proportional−derivative type	Reverse operation	H2	H8002
	Forward operation	H3	H8003

Auto-tuning

The optimum values for the Kp, Ti, and Td of the PID parameters can be measured by measuring the process response. When auto tuning is executed, the estimated results are reflected in the parameter area after auto tuning has been completed. (There may be cases in which auto tuning cannot be executed, depending on the process. If this happens, processing returns to the original parameter operation.)
For precautions concerning execution of auto tuning, refer to the following page.

Reverse operation and forward operation

These parameters determine whether the output will increase or decrease when a change occurs in the process.

Reverse operation

If the measured process value decreases, the output will increase. (Example: Heating)

Forward operation

If the measured process value increases, the output will increase. (Example: Cooling)

Derivative type PID and proportional−derivative type PID

When the set value is changed, the output changes.

Derivative type

In general this produces a large change when a set value is changed, however, convergence is fast.

Proportional−derivative type

In general this produces a small change when a set value is changed, however, convergence is slow.

② **Set value (SP)** . . . . [S+1]

Set the target value which determines the amount of process control within the following range.

K0 to K10000

③ **Measured value (PV)** . . . . [S+2]

Input the current process control value with the A/D converter. Adjust so that it falls within the following range.

K0 to K10000

④ **Output value (MV)** . . . . [S+3]

The result of PID processing is stored. Use the D/A converter or other device to output it to the process.

K0 to K10000

⑤ **Output lower limit value** . . . . [S+4]

K0 to K9999 (< upper limit value)

⑥ **Output upper limit value** . . . . [S+5]

K1 to K10000 (> lower limit value)

Specify the output value (MV) range. Values specified for the range are output.

The limits should be as follows:
0 ≦ output lower limit value < output upper limit value ≦ 10000.

⑦ **Proportional gain (Kp)** . . . . [S+6]

Specify the coefficient used for PID processing.

The set value × 0.1 will be the actual proportional gain.

The setting range is K1 to K9999 (0.1 to 999.9, specify in increments of 0.1)

If auto-tuning is selected for the specified control mode, the set value will be automatically adjusted and rewritten.

⑧ **Integral time (Ti)** . . . . [S+7]

Specify the coefficient used for PID processing.

The set value × 0.1 will be the actual integral time.

The setting range is K1 to K30000 (0.1 to 3000 seconds, specify in increments of 0.1 seconds)

When the set value is 0, the integration is not executed.

If auto-tuning is selected for the specified control mode, the set value will be automatically adjusted and rewritten.

⑨ **Derivative time (Td)** . . . . [S+8]

Specify the coefficient used for PID processing.

The set value × 0.1 will be the actual derivative time.

The setting range is K1 to K10000 (0.1 to 1000 seconds, specify in increments of 0.1 seconds)

If auto-tuning is selected for the specified control mode, the set value will be automatically adjusted and rewritten.

⑩ **Control cycle (Ts)** . . . . [S+9]

Set the cycle for executing PID processing. The set value × 0.01 will be the actual control period.
The setting range is K1 to K6000 (0.01 to 60.0 seconds, specify in increments of 0.01 seconds).

⑪ **Auto-tuning progress** . . . . [S+10]

If auto−tuning is specified in the control mode, the degree to which auto−tuning has progressed is indicated. The values for K1 to K5 are stored based on the progress from the default value of 0, and the values return to the default values when auto−tuning has been completed.

⑫ **PID processing work area** . . . . [S+11] to [S+29]

The system uses this work area to perform PID processing.

For the FP0 use the 20 words, [S+11] to [S+30], as the work area.

Precautions when executing auto−tuning

If “Execute auto−tuning” is specified using the parameter table (control mode [S]), attention should be paid to the following points.

After auto−tuning has been completed, the control mode [S] area is automatically rewritten from H8000 to H8003 to H0 to H3. Make sure the mode is not rewritten again in the program.

After auto−tuning has been completed, the optimum values are stored for the proportional gain (Kp), the integration time (Ti) and derivative time (Td), but before executing auto−tuning, the appropriate values (for example, the lower limit value) within the setting range must be set.

After auto−tuning has been completed, the optimum values are stored for the proportional gain (Kp), the integration time (Ti) and derivative time (Td). Be careful that the stored values are not inadvertently rewritten.

Auto−tuning calculates the ideal Kp, Ti and Td values. This is done for setting (SP) by measuring the change of the measured value (PV) when the output value (MV) is set to the upper limit so that the measured value (PV) is caused to fluctuate, and measuring the change of the measured value (PV) when the output value (MV) is set to the lower limit.

The changes of the output value (MV) when auto−tuning will complete after a minimum of three changes: upper limit output−lower limit output−upper limit output. If the auto−tuning progress remains at 0 even after changes have occurred several times, please try again after shortening control synchronization Ts.

Precautions during programming

A 30 word area (31 words for the FP0), including the operation work area, is required for the parameter table. Be careful not to allow other instructions to overwrite values in this area.

Error detection will not occur even if the parameter table exceeds the area.
When specifying ”S” specify a number that is within at least 30 words (31 words for the FP0) from the last number.

Take care that the area is not exceeded due to index modification. An error will not be detected if the area is exceeded.

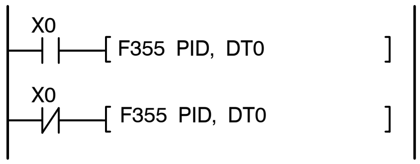
Use the A/D converter or other device to input the current measured value S+2.

Use the D/A converter or other device to output the result of PID processing S+3 to the process.

For the FP0, this instruction **F355 (PID)** cannot be programmed in the interrupt program.

If two or more PID instructions specifying the same table are described in the program, it may operate incorrectly.

(Example)



(Reason)

Because the PID instructions are internally operating using the specified table, even if the execution condition has not been effected.

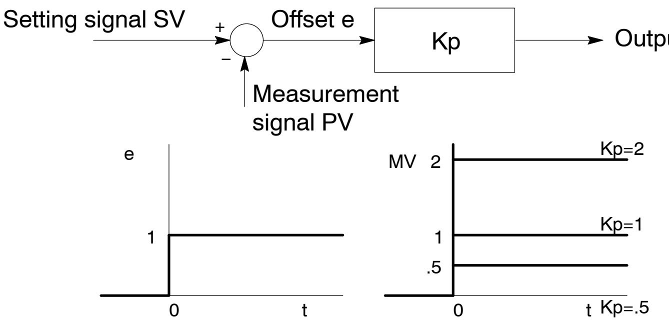
In such a case, specify the table in separate addresses.

Operation of PID control

PID is a control method widely used in the instrumentation field involving feedback control of process quantities such as temperature, pressure, flow, and fluid level.

1) Proportional operation

Proportional operation generates an output which is proportional to the input.



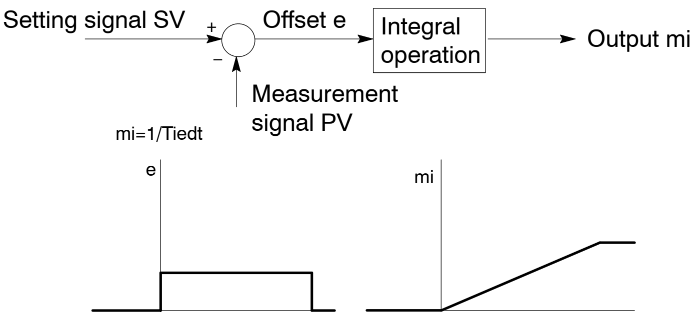
The amount of control is held constant.

An offset remains.

Proportional control grows stronger as Kp is increased.

2) Integral operation

Integral operation generates an output which is proportional to the integral time of the input.

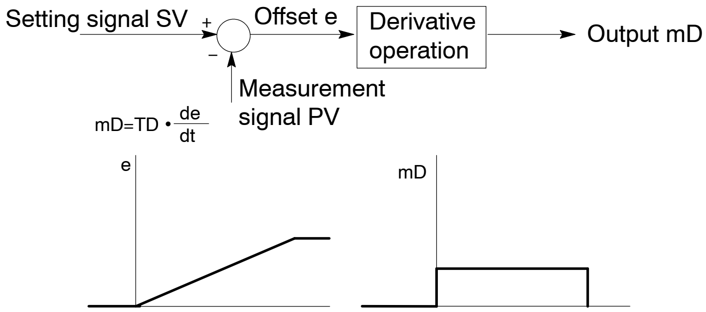


In combination with proportional operation or proportional−derivative operation, integral operation removes the offset produced by these methods.

Integral operation grows stronger as the integral time (Ti) is shortened.

3) Derivative operation

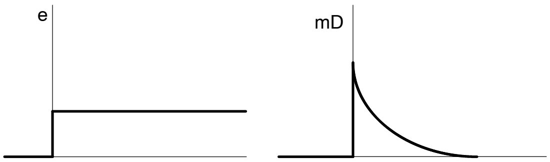
Derivative operation generates an output which is proportional to the derivative time of the input.



The advancing characteristic of derivative control alleviates the adverse effect which the delaying characteristic of the process exerts on control.

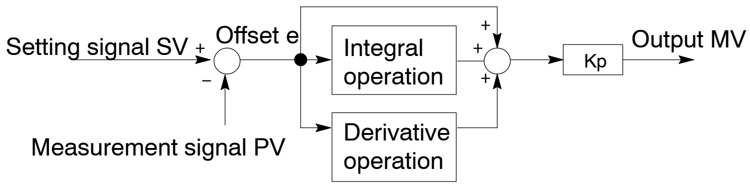
Derivative control grows stronger as the derivative time (Td) is increased.

In the case of pure derivative operation, control can temporarily become ineffective if noise is input, and this can have an adverse effect on the process being controlled. For this reason, incomplete differential operation is executed.



4) PID operation

PID operation is a combination of proportional, integral, and derivative operation.



If the parameters are set to the optimum values, PID control can quickly bring the amount of control to the target value and maintain it there.

F356 (EZPID) Easy PID

Availability
FP−X: Ver 1.20 or more FPΣ: 32k/FP0R

Outline Temperature control (PID) can be easily performed using the image of a temperature controller.

Program example

Ladder Diagram	Boolean	
	Address	**Instruction**
	10	ST R 1
	11	F356(EZPID)
		WR1
		WX2
		DT32710
		DT100
	21	OT Y 0

S1	Control data
S2	Measured value (PV)
S3	Starting No. of area storing PID control parameter
S4	Starting No. of PID processing work area

Operands

Operand	Relay				Timer/Counter		Register		Index register	SWR	SDT	Constant		Index modifier
	WX	WY	WR	WL	SV	EV	DT	LD	In (*1)			K	H	
S1	N/A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	–
S2	A	A	A	A	A	A	A	A	A	A	A	N/A	N/A	–
S3	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A	–
S4	N/A	A	A	A	A	A	A	A	N/A	N/A	N/A	N/A	N/A	–

(*1) I0 to ID.

A: Available
N/A: Not Available

Operation

PID processing is performed to hold the measured value (PV) at the set value (SP).
Writing OUT instruction immediately after this instruction enables the PWM output (on−off output) similar to a temperature controller.
Auto−tuning function is also available to calculate the control parameter of PID automatically.
It can be used with analog output as it outputs values as well as PWM output.

General explanation of used memories

S1: It is recommended to specify the non−hold type area (e.g. WR) where can be operated in bit unit.
When bit0 is 1, it is auto−tuning request. This bit is reset with this instruction when auto−tuning has completed.
Reset this bit to cancel auto−tuning.
When bit0 is 0, it is PID control.
When auto−tuning has completed successfully, 1 is set for bit 1.
Bit2 turns on to hold the output MV (S4) when the execution condition of this instruction changes from off to on.
When bit3 is 0, it is PWM control. And when bit3 is 1, it is ANALOG output control.
When bit4 is 0, the max. value of the internal output is the output upper limit value +20% of the output range (output upper limit value − output lower limit value), and the min. value is the output lower limit value −20% of the output range.
When bit4 is 1, the max. value of the internal output is the output upper limit value, and the min. value is the output lower limit value.
*The output lower limit value is specified by S4+1, and the output upper limit value is specified by S4+2.
Bits 5 to F are reservation bits. Normally use them as 0.

S2: Area storing measured value (PV) (1 word) — Effective range
The input WXn of a temperature input unit can be directly specified. — k−30000 to k+30000

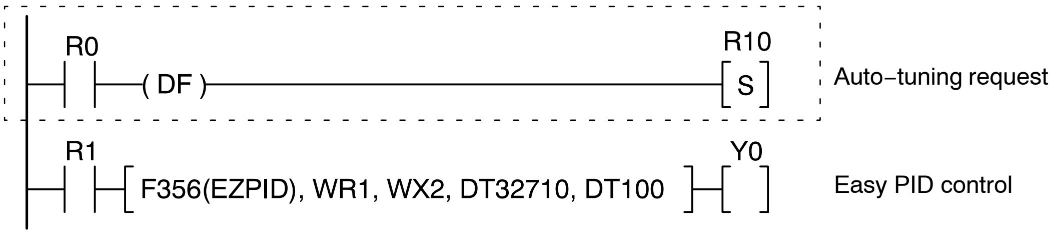
S3: Area to specify target value (SP) and control parameter (4 words)
It is recommended to use this area allocating to the hold−type operation memory. — Setting range

S3:	Store set value (SP).	It is necessary to specify by the instruction or an indicator.	k−30000 to k+30000
S3+1:	Store proportional gain (KP). Actual gain is Set value x 0.1.	After auto−tuning has been completed, it is automatically set.	k1 to K9999 (0.1 to 999.9)
S3+2:	Store integral time (TI). Actual integral time is Set value x 0.1.	After auto−tuning has been completed, it is automatically set.	k0 to k30000 (0 to 3000s)
S3+3:	S3+3: Store derivative time (TD). Actual derivative time is Set value x 0.1.	After auto−tuning has been completed, it is automatically set.	k0 to k10000 (0 to 1000s)

S4: It is divided into output (MV), specified area of control mode, auto−tuning related area and PID processing work area.
The area in the range of S4 to S4+29 is necessary for the instruction. (The detail is described hereinbelow.)
It is recommended to allocate it in the non−hold area. Also, do not use the data in this area for other purposes.

Easy usage

When PWM output in reverse operation (heating)



Explanation of operation

Specify set value (SP) with the instruction or an indicator before the operation.

If auto−tuning is requested with a device such as indicator, the above auto−tuning request program is not necessary.

When R1 turns on, work area DT100 to DT129 will be initialized. (However, only DT100 (MV) can be held.) The control conditions are that operation cycle is 1 sec, derivative type reverse operation (heating), and PWM resolution is 1000.
PID control starts from the next scan, and PWM output is executed for Y0.

Note) If execution condition R1 has turned off during PID control, PWM output Y0 also turns off. However, output value MV is held.

Program as described above to start auto−tuning with the instruction, and turn on R1 after turning on R0. When auto−tuning has completed successfully, R11 turns on and KP, TI and TD is set.
If R1 is on continuously, it will change to PID control automatically, and PWM output will be executed for Y0.

When changing control conditions

The area of S4+1 to S4+9 must be changed to change control conditions. Change it before the second execution of the F356 instruction.

<Details of S4>

S4: It is divided into output (MV), specified area of control mode, auto−tuning related area and PID processing work area. It is recommended to allocate it in the non−hold area. Also, do not use the data in this area for other purposes.

Output (MV) and control mode area
(Normally, the default values are used.)

		Default value	Range
S4:	The output value (MV) of PID processing is stored.	k0	k−10000 to 10000
S4+1:	Set the lower limit value of output value (MV).	k0	Min. k−10000
S4+2:	Set the upper limit value of output value (MV).	k10000	Max. k+10000
S4+3:	Set 100% output band (range where PID control is not performed).	k0	k0 to 80(%)
S4+4:	Set control cycle (TS). Setting unit=10ms, default value=1sec.	k100	k1 to 3000(0.01 to 30s)
S4+5:	Set control mode. (Refer to the table below.)	k0	k0 to 3

Control mode		Value	e.g.
Derivative type	Reverse	k0	Heating
	Forward	k1	Cooling
Proportional−derivative type	Reverse	k2	Heating
	Forward	k3	Cooling

Reverse operation and forward operation

Reverse operation: If the measured value drops, “Reverse operation” is specified to boost the output (heating, etc.).

Forward operation: If the measured value increases, “Forward operation” is specified to boost the output (cooling, etc.).

Derivative type (PI−D)/Proportional−derivative type (I−PD)

Derivative type: The speed is faster to get to set value, but it is easily overshooted.

Proportional−derivative type: The speed is slow to get to set value, but it is not easily overshooted.

Auto−tuning related area (The default value is normally used.)

S4+6:	Set bias value for performing auto−tuning.	k0	from k0
S4+7:	Set correction data (a1) of auto−tuning result (KP).	k125	k50 to k500%
S4+8:	Set correction data (a2) of auto−tuning result (TI).	k200	k50 to k500%
S4+9:	Set correction data (a3) of auto−tuning result (TD).	k100	k50 to k500%
S4+10:	The status while auto−tuning is being performed is stored.	k0	k0 to k5

PID processing work area

S4+11:	The area up to S4+29 is	0
to	the work area for PID processing and auto−tuning processing.	to
S4+29:		0

Note) The default value is written when the execution condition turns on.
Output (MV) is output only in the ranges of upper limit value and lower limit value.
“Also, set to be as −1000 ≦ lower limit value < upper limit value ≦ 10000.”

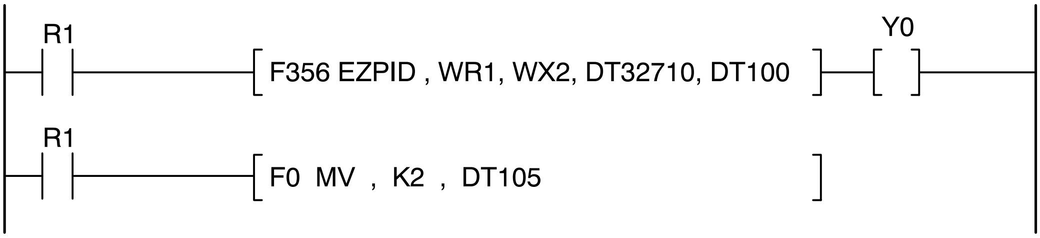
How to output PWM.
The cycle of a PWM output is decided by the setting value of S4+4. The default value is periodic 1 second. Duty of PWM is decided by the rate of the output MV (S4) that accounts for in the range of k0 to k10000. When either on of the minimum value and maximum value of Output MV specified by S4+1 and S4+2 is a negative value, the PWM output is always OFF.
A PWM output is always ON, when the output MV is k0, and it is always OFF when the output MV is k10000.

Explanation of specific usage

1: When changing control mode only with PWM output

Change the content of the control mode (S4+5) to k1 to k3 using an instruction such as F0(MV) instruction.

(Example) Change the control mode to the proportional derivative type from the derivative type that is the default.



2: When using an analog output unit for output

2−1: Set the bit3 of S1 to 1 in order to start ANALOG output control.

2−2: Set output lower limit value (S4+1) and output upper limit value (S4+2) according to the output range of an analog output unit.

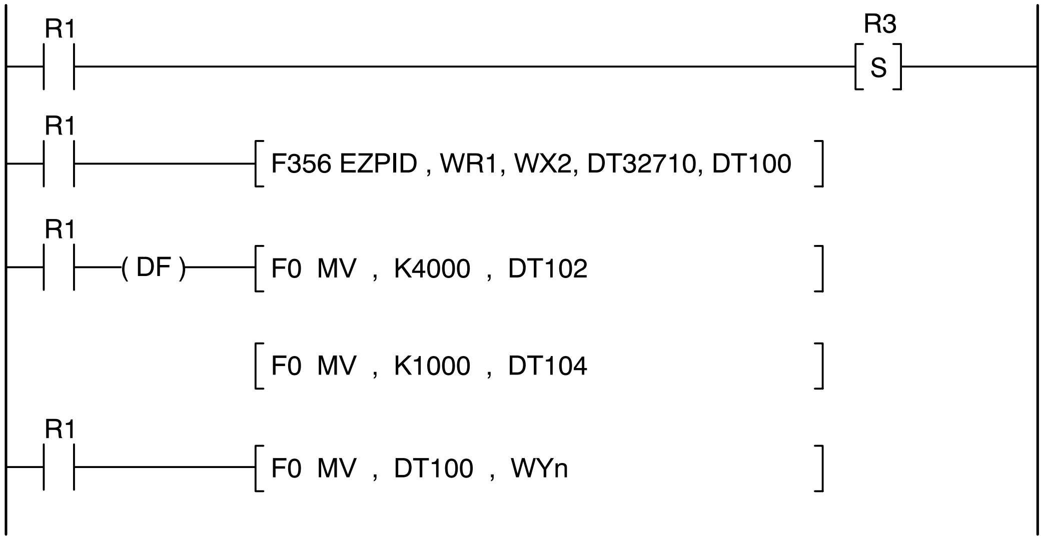
(Example) <Lower limit value=k0, upper limit value=k2000>, <Lower limit value=k0, upper limit value=k4000>

2−3: Change the value of control cycle (TS): (S4+4) according to the cycle of updating input of a temperature input unit (that is normally 0.15 or more).
(Example) TS=k10 (100ms)

2−4: Change the control mode if necessary.

2−5: Transmit output value (MV) to WY of an analog output unit.

Note) When analog output is used for output, it is not necessary to write OUT instruction immediately after this instruction.
Also, when analog output is used, PWM output is fixed to off.
(Example) When controlling with the settings that the output upper value (S4+2) is K4000 and the control cycle (S4+4) is 10 seconds



More detail on setting method

1: Setting for 100% output band (S4+3)

100% output band is to specify the timing of starting PID control when measured value (PV) becomes more than what percentage of set value.
100% output is performed in the area up to a specified measured value.
When measured value (PV) is smaller than set value (SP) *this setting, it has affect on reducing the arrival time to set value (SP) by performing 100% output.
Therefore, when it is set to k80, 100% output is performed up to 80% of set value (SP), and PID control starts from then.
When k0 has been set to the default value for this setting, PID control is performed from the beginning.

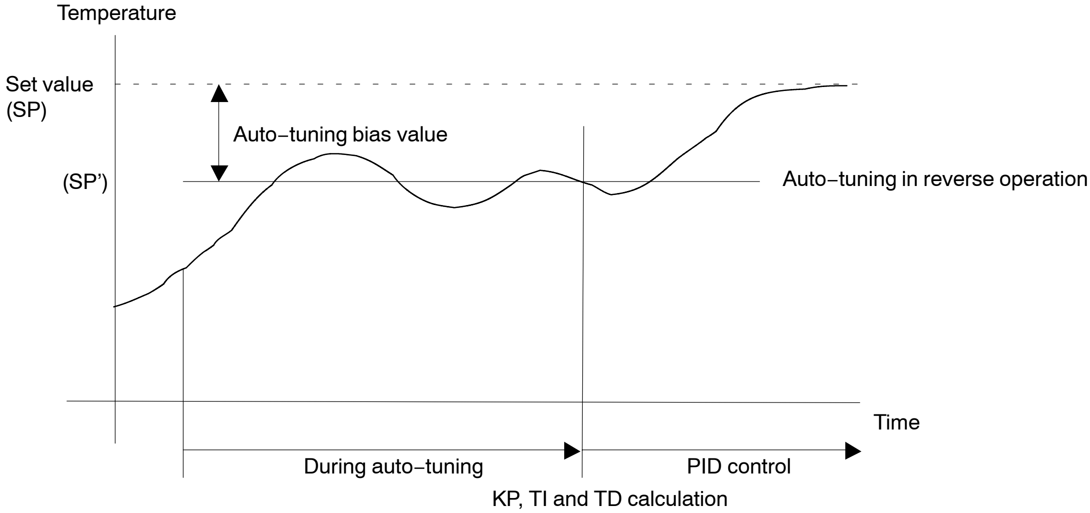
2: Fine adjustment of auto−tuning

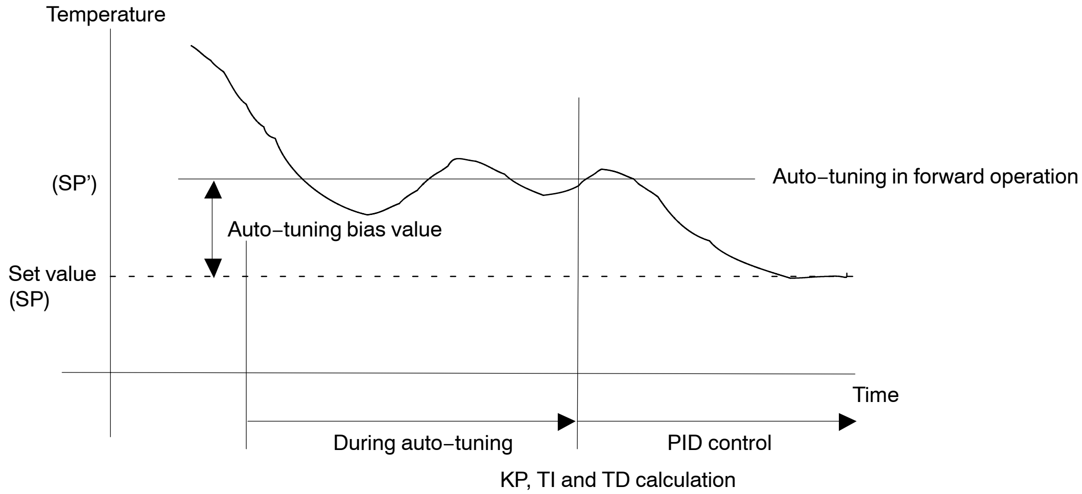
2−1: Correction of the result of auto−tuning (S4+7, S4+8 and S4+9)
When auto−tuning has completed, the parameters for KP, TI and TD are stored in (S3+1, S3+2 and S3+3).
That result can be corrected with this parameter.

(Example)

Set S4+7 to k200 (means to 200%) and perform auto−tuning to correct KP to double value.
Set S4+8 to K128 (means to 125%) and perform auto−tuning to correct TI to 1.25 times value.
Set S4+9 to k75 (means to 75%) and perform auto−tuning to correct TD to 0.75 times value.

2−1: Auto−tuning bias value (S4+6)
Auto−tuning is executed with (set value (SP) − auto−tuning bias value) as a set value (SP’).
It is used to control excessive temperature rise while auto−tuning is performed.
For the forward operation, auto−tuning is executed with (set value (SP) + this set value) as a set value.





Note) Even if starting auto−tuning in the condition that measured value (PV) is close to set value (SP), auto−tuning is performed with the above SP’.

Precautions on programming

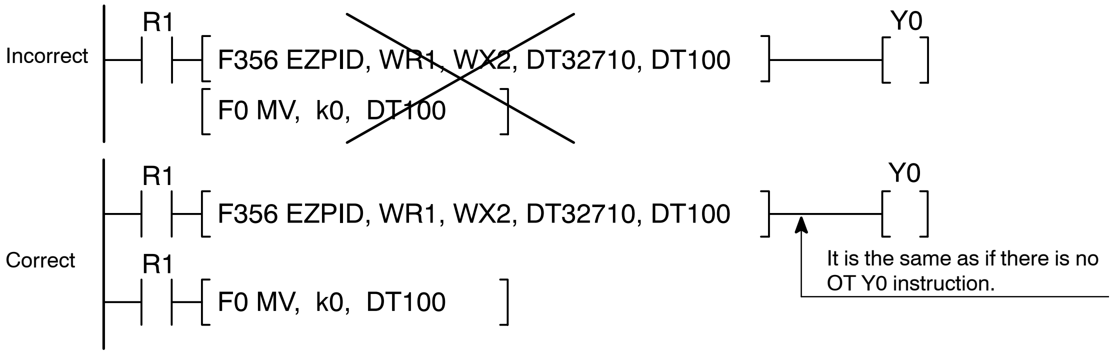
1: When the execution condition has turned on, the area of S4 to S4+29 is initialized.
If it is set to values other than the default values, write with F0(MV) instruction using always on relay R9010 as execution condition.

2: As operation cycle or timing of PWM output is always operated internally with PID processing instruction, always operate only once in 1 scan.
Therefore, do not execute it during the subroutine or interrupt program.
This instruction that the same operand has been specified cannot be written more than once.

3: Do not turn off the execution condition during PID processing. Otherwise, PID processing will be disabled.

4: If you do not want to synchronize PWM output cycle for controlling multiple objects, delay the timing of start−up by adjusting times such as the rise time for start−up condition.

5: As execution condition is changed after executing this instruction, after instructions cannot operate correctly with the program below.



Conditions when operation errors occur

1: S2: Measured value (PV), S3: Set value (SP), S3+1: KP, S3+2: TI, S3+3: TD
When each parameter of S4+1 to S4+9 is out of the setting range.

2: When the area specified with S3 or S4 exceeds the upper limit of the specified operation device.

Internal operation specifications

When the execution condition has turned on, the operation work is initialized.
If each parameter of KP, TI and TD is all 0 when PID operation has started, they are initialized at 1,1 and 0, respectively. And the operation is continued.

AT normal done flag or AT done code is cleared on the leading edge of AT signal.
The set value for AT operates regarding <set value (SP) − bias value> as target value. Default value for bias value is 0.
When AT has completed successfully, the results which is calculated by raising KP, TI and TD of calculated results to the power of correction data a1, a2 and a3 are stored. Default value is 100%.
When AT has completed successfully, AT normal done flag is set, and AT done code is stored in AT step.
When AT has abended, the parameters of KP, TI and TD are not changed.
PWM output is output at the duty when the output range of MV is 0 to 10000.

For analog output (when bit3 of S1 is 1), the internal calculated value output in the range of 0 to 10000 and it is converted into a specified range.

Conversion formula: (Upper limit value − Lower limit value) x internal calculated value / 10000 + Lower limit value

Example) When upper limit value = 40000, lower limit value = 0 and internal calculated value = 5000, output value is 2000.

Precautions when using MV holding function · · · · The usage varies according to models and versions.

1. For FPΣ, FP−X, FP0R (V1.05 or older)
Use the default upper limit and lower limit values for using the MV holding function.

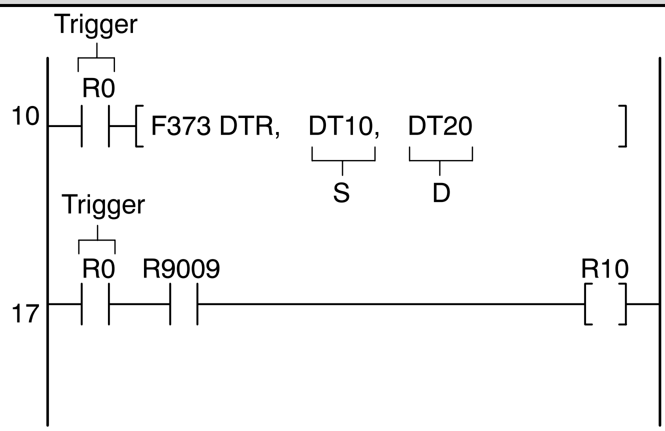
2. For FP0R (V1.06 or later)
Upper limit and lower limit values are held as well as MV value, set MV value, upper limit and lower limit values before executing this instruction.

F373 (DTR) P373 (PDTR) 16-bit data revision detection

Outline This instruction detects changes in 16-bit data values.

Program example

Ladder Diagram



Boolean			
Address	Instruction		
10	ST	R	0
11	F373		(DTR)
	DT		10
	DT		20
17	ST	R	0
18	AN		R9009
20	OT	R	10

S	16-bit area for detecting data changes.
D	Area where data of previous execution is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the trigger R0 turns on, R9009 will turn on if there has been a change in data register DT10 since the previous execution. Following this, the internal relay R10 will also turn on.

Description

If the data in the 16-bit area specified by S has changed since the previous execution, internal relay R9009 (carry flag) will turn on.
D is used to store the data of the previous execution, and when the current execution has been completed, the current data is stored in D.

Precautions during programming

The internal relay R9009 (carry flag) used for detection of data changes is updated at each execution of the instruction.

For this reason,

Programs using R9009 should insert it immediately after an **F373 (DTR)**/**P373 (PDTR)** instruction.

Output to an output relay or internal relay to hold the result. (Refer to the explanation of **F64 (BCMP)**/**P64 (PBCMP).**)

As in the program example on preceding page, be sure to add the trigger (X10) for the **F373 (DTR)**/**P373 (PDTR)** instruction before the internal relay R9009 (carry flag).
If the always on relay (R9010) is the execution condition, this trigger (X10) is not necessary.

Flag conditions

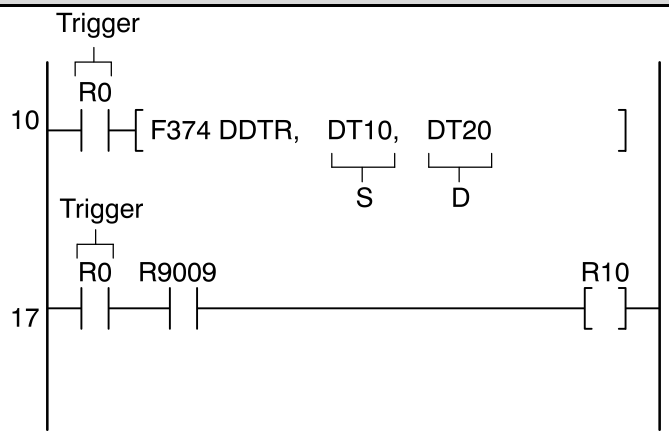
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on if a change has occurred in the specified data area.

F374 (DDTR) P374 (PDDTR) 32-bit data revision detection

Outline This instruction detects changes in 32-bit data values.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	R	0
11	F374		(DDTR)
	DT		10
	DT		20
17	ST	R	0
18	AN		R9009
20	OT	R	10

S	Lower 16-bit area of 32-bit data for detecting data changes.
D	Lower 16-bit area of 32-bit data where data of previous execution is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
S	A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A
D	N/A	A	A	A	A	A	A	A	A	A	N/A	N/A	N/A	A	N/A

A: Available
N/A: Not Available

Explanation of example

When the trigger R0 turns on, R9009 will turn on if there has been a change in data register DT10 since the previous execution. Following this, the internal relay R10 will also turn on.

Description

If the data in the 32-bit area specified by S has changed since the previous execution, internal relay R9009 (carry flag) will turn on.
D+1 and D is used to store the data of the previous execution, and when the current execution has been completed, the current data is stored in D+1 and D.

Precautions during programming

The internal relay R9009 (carry flag) used for detection of data changes is updated at each execution of the instruction.

For this reason,

Programs using R9009 should insert it immediately after an **F374 (DDTR)**/**P374 (PDDTR)** instruction.

Output to an output relay or internal relay to hold the result. (Refer to the explanation of **F64 (BCMP)**/**P64 (PBCMP)**.)

As in the program example on preceding page, be sure to add the trigger (X10) for the **F374 (DDTR)**/**P374 (PDDTR)** instruction before the internal relay R9009 (carry flag).
If the always on relay (R9010) is the execution condition, this trigger (X10) is not necessary.

Flag conditions

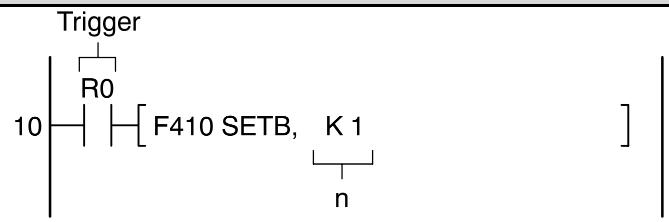
- Error flag (R9007): Turns on and stays on when the area specified using the index modifier exceeds the limit.
- Error flag (R9008): Turns on for an instant when the area specified using the index modifier exceeds the limit.
- Carry flag (R9009): Turns on if a change has occurred in the specified data area.

F410 (SETB) P410 (PSETB) Setting the index register bank number

Outline Setting the index register bank number

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F410 (SETB)
	K 1

n	Constant data or area where register bank number is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

Description

This instruction selects the current index register bank number.

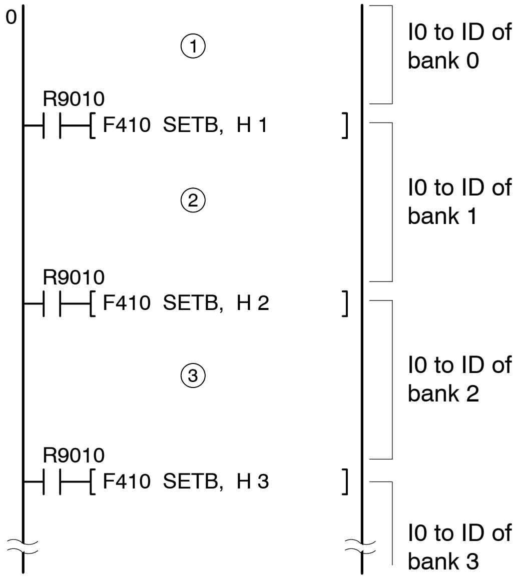
Precautions during programming

The bank number is automatically set to bank 0 before execution of the starting address of the program.

If the program memory is 120K steps, when the program is switched to the No. 1 program or the No. 2 program, the index register bank number is automatically set to 0.

Program example

Changing the index register banks



Flag conditions

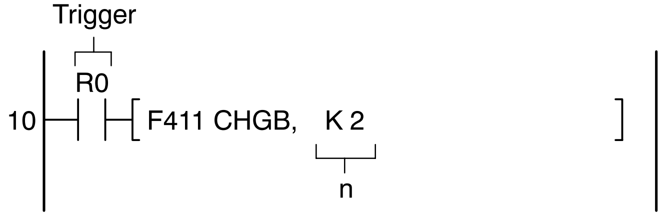
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The bank number is not from K0 to K15.

F411 (CHGB) P411 (PCHGB) Changing the index register bank number

Outline Index register bank number change over with remembering preceding bank number.

Program example

Ladder Diagram



Boolean

Address	Instruction
10	ST R 0
11	F411 (CHGB)
	K 2

n	Constant data or area where register bank number is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

Description

This instruction selects the current index register bank number.
At this time, the current index bank number is stored in the push area (the push area has only one effective level, thus previous data is overwritten).

The bank numbers of index registers used in interrupt programs, subroutines, and other sub programs should be specified in such a way that the **F411 (CHGB)** instruction is executed at the beginning of the sub program, and the **F412 (POPB)** instruction is executed at the end of the sub program.

Precautions during programming

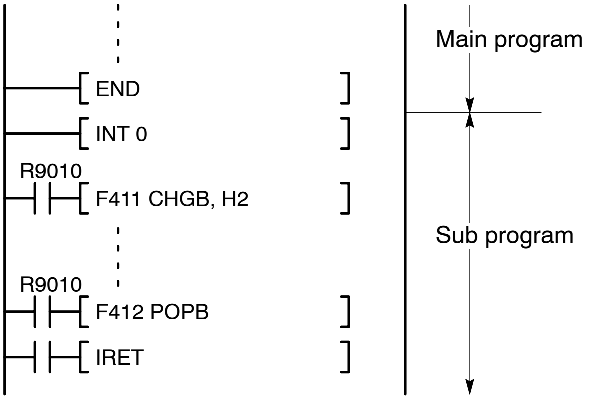
The bank number is automatically set to bank 0 before execution of the starting address of the program.

If the program memory is 120K steps, when the program is switched to the No. 1 program or the No. 2 program, the index register bank number is automatically set to 0.

The push area has only one effective level, thus the previous data is overwritten.

Program example

This is a program in which the index register bank is switched to “2” at the beginning of the interrupt program, and is then switched back again to the original index register bank just before the end of the interrupt program (before the **IRET** instruction).



Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The bank number is not from K0 to K15.

F412 (POPB) P412 (PPOPB) Restoring the index register bank number

Outline Changes index register bank number back to the bank before **F411 (CHGB)**/**P411 (PCHGB)** instructions.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F412 (POPB)

Description

The current index register bank number is changed to the number stored in the push area.

The contents of the push area are not changed at this time.

The bank numbers of index registers used in interrupt programs, subroutines, and other sub programs should be specified in such a way that the **F411 (CHGB)** instruction is executed at the beginning of the sub program, and the **F412 (POPB)** instruction is executed at the end of the sub program.

Precautions during programming

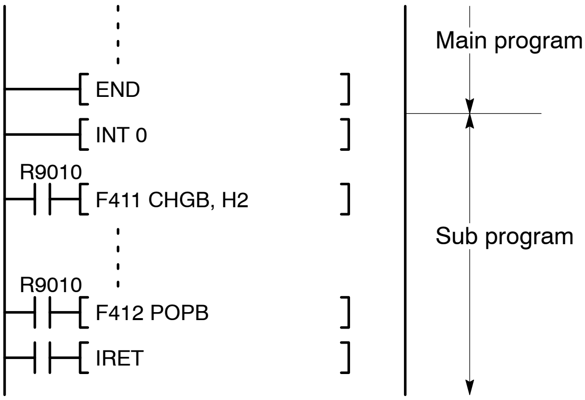
The bank number is automatically set to bank 0 before execution of the starting address of the program.

If the program memory is 120K steps, when the program is switched to the No. 1 program or the No. 2 program, the index register bank number is automatically set to 0.

The push area has only one effective level.

Program example

This is a program in which the index register bank is switched to “2” at the beginning of the interrupt program, and is then switched back again to the original index register bank just before the end of the interrupt program (before the **IRET** instruction).

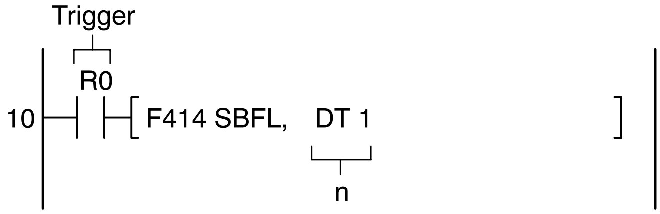


F414 (SBFL) P414 (PSBFL) Setting the file register bank number

Outline
Setting the file register bank number

Program example

Ladder Diagram



Boolean	
Address	**Instruction**
10	ST R 0
11	F414 (SBFL)
	DT 1

n	Constant data or area where register bank number is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

Description
This instruction selects the current file register bank number.
File register bank number: 0 to 2.

Precautions during programming
The file register bank is set to bank 0 at the first step of program.
The file register bank is also set to bank 0 at the first step of No. 2 program.
Special data register for file register bank.

DT90263	File register bank (current value)	The current value of file register bank is stored.
DT90264	File register bank (shelter number)	The shelter number of file register bank is stored.

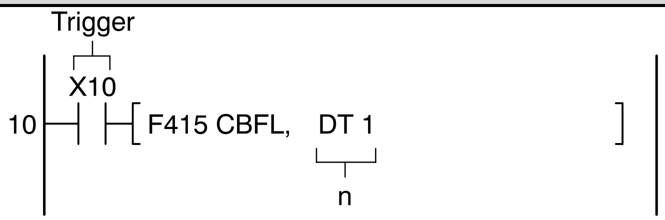
Flag conditions
- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The bank number is not from K0 to K2.

F415 (CBFL) P415 (PCBFL) Changing the file register bank number

Outline Changing the file register bank number.

Program example

Ladder Diagram



Boolean Address	Boolean Instruction		
10	ST	X	10
11	F415		(CBFL)
	DT		1

n	Constant data or area where register bank number is stored.

Operands

Operand	Relay				Timer/Counter		Register			Index register	Constant			Index modifier	Integer device
	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	K	H	f		
n	A	A	A	A	A	A	A	A	A	A	A	A	N/A	A	N/A

A: Available
N/A: Not Available

Description

This instruction selects the current file register bank number.
At this time, the current file bank number is stored in the push area (the push area has only one effective level, thus previous data is overwritten).
File register bank number: 0 to 2.

Precautions during programming

The file register bank is set to bank 0 at the first step of program.
The file register bank is also set to bank 0 at the first step of No. 2 program.
Special data register for file register bank.

DT90263	File register bank (current value)	The current value of file register bank is stored.
DT90264	File register bank (shelter number)	The shelter number of file register bank is stored.

Flag conditions

- Error flag (R9007): Turns on and stays on when:
- Error flag (R9008): Turns on for an instant when:
  - The area specified using the index modifier exceeds the limit.
  - The bank number is not from K0 to K2.

F416 (PBFL) P416 (PPBFL) Restoring the file register bank number

Outline Changes file register bank number back to the bank before **F415 (CBFL)**/**P415 (PCBFL)** instructions.

Program example

Ladder Diagram	Boolean	
	Address	Instruction
	10	ST R 0
	11	F416 (PBFL)

Description

The current file register bank number is changed to the number stored in the push area. The contents of the push area are not changed at this time.

The user must manage the push area data so that the desired data is restored. This instruction only checks the data range, it does not check changes made with the **F415 (CBFL)** instruction.

The push area has only one effective level.

Precautions during programming

The file register bank is set to bank 0 at the first step of program.

The file register bank is also set to bank 0 at the first step of No. 2 program.

Special data register for file register bank.

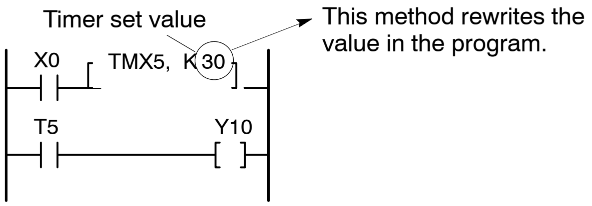
DT90263	File register bank (current value)	The current value of file register bank is stored.
DT90264	File register bank (shelter number)	The shelter number of file register bank is stored.

Chapter 4

Precautions Concerning Programs

4.1 Changing the Set Value of Timer/Counter During RUN

4.1.1 Method of Rewriting Constant in the Program



Changing the set values (constants) in the program

Constants in the program can be rewritten as long as the following conditions are observed.

Operation method:	RAM operation only
Rewriting method:	Method using the programming tool software Method using the FP Programmer II

Rewrite method using the programming tool software

Example of changing the set value of timer 5 from K30 to K50

1. **Place the cursor on the value of K30 set for the timer 0.**
2. **Press the "Delete" key of computer to clear the value.**
3. **Enter a new constant of K50, and press the "Enter" key.**

Rewrite method using FP programmer II

Example of changing the set value of timer 5 from K30 to K50

1. **Read the address containing the timer instruction.**

   SHIFT SC · TM T-SV · ST X-WX · 5 · SRC ▲ · READ ▼ · READ ▼

2. **Clear the constant (K30).**

   (HELP) CLR

3. **Enter the new constant (K50).**

   (BIN) K/H · 5 · 0 · WRT

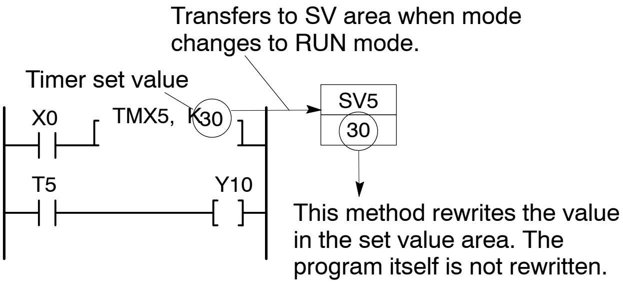
Operation and cautions after the change

After the change using the programming tool software or FP programmer II, the timer or counter in operation will continue to run. Operation based on the changed set value will be start the next time the execution condition changes from off to on.

If changing values using the boolean (ladder/non−ladder) mode input method available in the programming tool software, subtraction is interrupted when the values are rewritten, and starts again with the new value, starting from the next scan.

When method of constant rewriting in the program is used, the program itself will change. Thus, when the mode is changed and then set back to RUN or when the power is turned on, the changed set value will be preset.

4.1.2 Method of Rewriting a Value in the Set Value Area



Changing values in the set value area SV

Values in the set value area SV can be changed with the following conditions.

Operation method: RAM operation, ROM operation

Rewriting method: Method using the programming tool software
Method using the FP Programmer II
Method using the program (high−level instruction)

Operation and cautions after the change

After the change, the timer or counter in operation will continue to run. Operation based on the changed set value will be start the next time the execution condition changes from off to on.

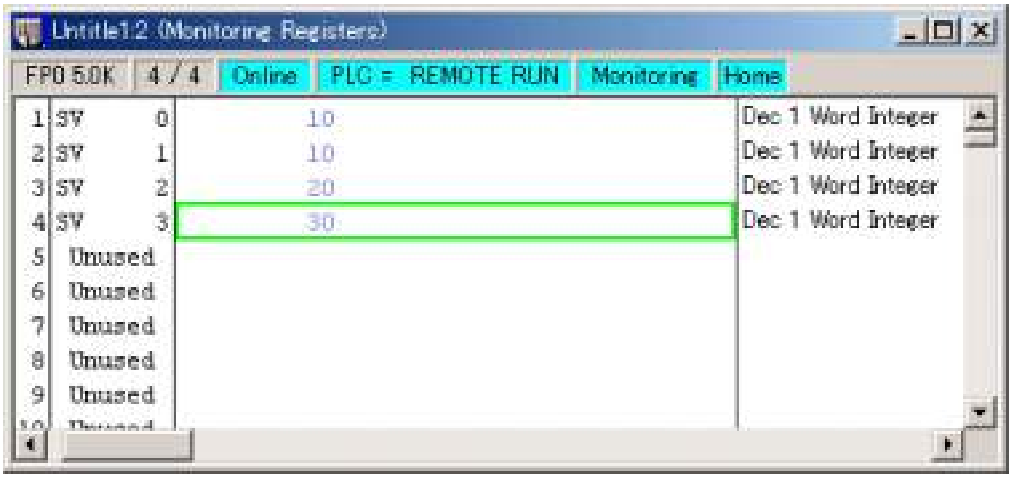
With these methods, the value in the set value area SV will change, however, the program itself will not change. Therefore, when the mode is changed and then set back to RUN or when the power is turned on, operation will take place as follows:

When a set value in the program is specified by a constant K
The constant K is preset in the set value area SV. After the change, it will no longer be effective.

When a set value in the program is specified by a set value area number
In the case of a non−hold type timer or counter, 0 is preset in the set value area SV. In the case of a hold type timer or counter, the value changed by the method on the following page is preset in the set value area SV.

Method 1: Method using the programming tool software

Select “MONITOR & TEST RUN” from the online menu, read the set value area SV of the timer or counter using the data monitor, and change the value.



Method 2: Method using the FP Programmer II

Use the word data monitor function to read the set value area SV of the timer or counter to be changed, and rewrite the value.

Example of changing the value of SV0 from K30 to K50.

1. **Execute word data monitor (OP8).**

   [(–) OP] [8] [ENT]

2. **Read SV0.**

   [TM T-SV] [0] [READ ▼]

3. **Clear SV0.**

   [(HELP) CLR]

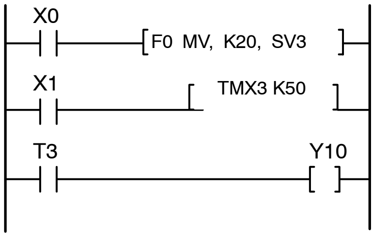
4. **Write the new changing value.**

   [(BIN) K/H] [5] [0] [WRT]

Method 3: Method using the program (high−level instruction)

To change a set value of timer/counter based on an input condition, use a high-level instruction as shown below to rewrite the value in the set value area SV of the desired timer or counter.

Example: Changing the set value to K20 when input X0 turns on



When X0 turns on, the set value of timer changes from 5 seconds to 2 seconds.

With the FP2SH and FP10SH, it is possible to specify the data register DT, as well the relay WR for handling word data, and other similar areas, as the set value area. The set value can be changed by changing the value to be transmitted, using the **F0 (MV)** instruction or a similar instruction.

4.2 Use of Duplicated Output

4.2.1 Duplicated Output

Duplicated output refers to repeatedly specifying the same output in a program.

If the same output is specified for the **"OT"** and **"KP"** instructions, it is considered to be duplicated output.
Even if the same output is used for multiple application instructions, such as the **SET** or **RST** instruction, or high–level instruction for data transfer, it is not regarded as duplicated output.

If you enter RUN mode while the duplicated output condition exists, under normal conditions, it will be flagged as an error. The ERROR (ERROR/ALARM) LED will light and the self-diagnostic error flag R9000 will go on.

How to check for duplicated use

You can check for duplicated outputs in the program using the programming tool, by the following method:

Using FP Programmer II:
Operate the TOTAL CHECK function.
(Key operation: [(–) OP] [9] [ENT] [READ ▼])
If there are any duplicated outputs, an error message (DUP USE) and the address will be displayed.

Using programming tool software (NPST-GR):
Excute the "TOTALLY CHECK A PROGRAM" on "CHECK A PROGRAM."
If there are any duplicated outputs, an error message (DUPLICATED OUTPUT ERROR) and the address will be displayed. If you execute "SEARCH AN ERROR," the error message will be displayed, and the first address number will be displayed.

Enabling Duplicated Output

If you need to use output repeatedly due to the content of the program, duplicated output can be enabled.

In this case, change the setting of system register 20 to "enable" (when using FP programmer II, set K1).

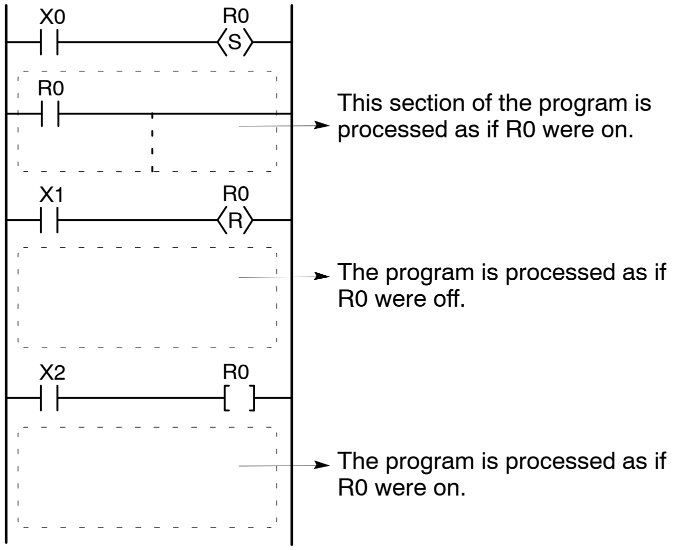
When this is done, an error will not occur when the program is executed.

4.2.2 When Output is Repeated with an OT, KP, SET, or RST Instruction

Condition of internal and output relays during operation

When instructions are repeatedly used which output to internal and output relays such as transfer instructions and **OT**, **KP**, **SET** and **RST** instructions, the contents are rewritten at each step during operation.

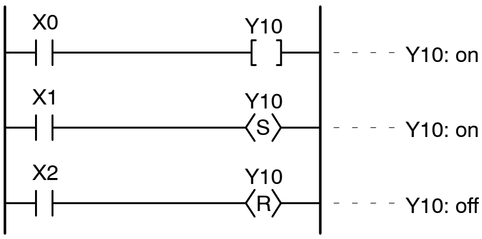
Example: **Processing when SET, RST and OT instructions are used (X0 to X2 are all on).**



The output is determined by the final operation results.

If the same output is used by several instructions such as the **OT**, **KP**, **SET**, **RST**, or data transfer instructions, the output obtained at the I/O update is determined by the results of the operation at the greatest program address.

Example: **Output to the same output relay Y10 with OT, SET and RST instructions.**



When X0 to X2 are all on, Y10 is output as off at I/O update according to the result of trigger X2.

If you need to output a result while processing is still in progress, use a partial I/O update instruction **F143 (IORF)**.

4.3 Leading Edge Detection Method

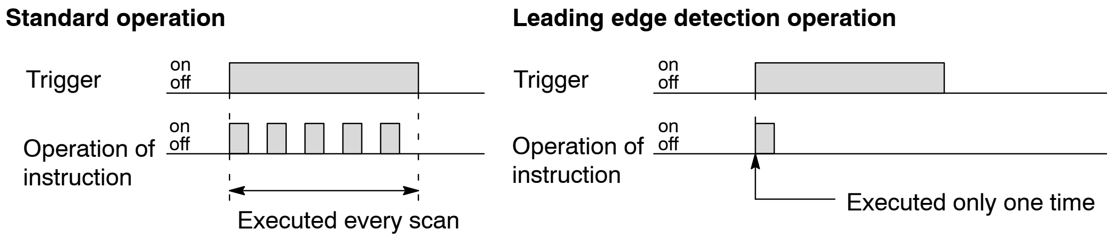
4.3.1 Instructions of Leading Edge Detection Method

Instructions using the leading edge detection operation:

- **DF** (leading edge differential) instructions
- Count input for **CT** instructions
- Count input for **F118 (UDC)** instructions
- Shift input for **SR** instructions
- Shift input for **F119 (LRSR)** instructions
- **NSTP** instructions
- P type high−level instructions (with the prefix “P”) for FP−C/FP2/FP2SH/FP10SH only

Leading edge detection method

An instruction with a leading edge detection method operates only in the scan where its trigger is detected switching from off to on.



The condition of the previous execution and the condition of the current execution are compared, and the instruction is executed only if the previous condition was off and the current condition is on. In any other case, the instruction is not executed.

Precautions when using an instruction which performs leading edge detection

When RUN begins, for example when the system is powered on, the off → on change of the trigger is not detected. The instruction is not executed. Execution of the instruction will take place as explained on the following page.

When used with one of the instructions indicated in instructions below which change the order of execution of instructions, the operation of the instruction may change depending on input timing. Take care regarding this point.

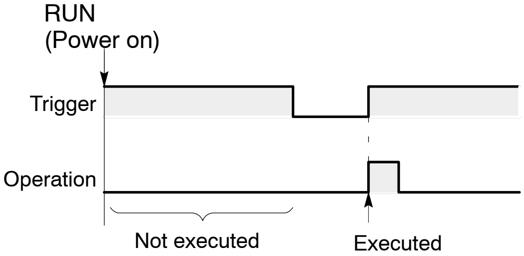
Be careful when using leading edge detection type instructions with control instructions, such as:

- **MC** and **MCE** instructions
- **JP** and **LBL** instructions
- **F19 (SJP)** and **LBL** instructions for FP−C/FP2/FP2SH/FP3/FP10SH only
- **LOOP** and **LBL** instructions
- **CNDE** instruction
- Step ladder instructions
- Subroutine instructions

4.3.2 Operation and Precautions at Run Start Time

Operation of first scan after RUN begins

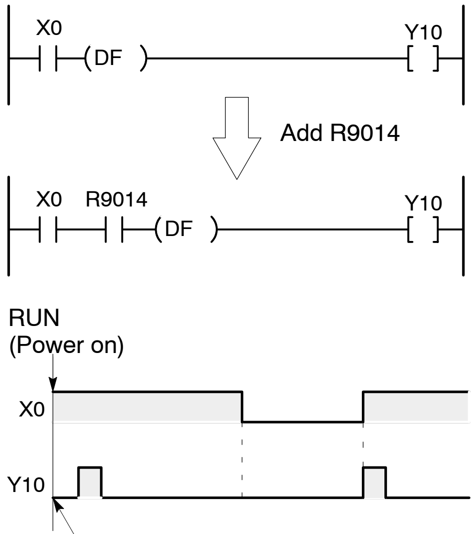
The leading edge detection instruction is not executed when the mode has been switched to the RUN mode, or when the power supply is booted in the RUN mode, if the execution condition is already on.



If you need to execute an instruction when the trigger (execution condition) is on prior to switching to RUN mode, use the special internal relay R9014 in your program as follows. (R9014 is a special internal relay which is off during the first scan and turns on at the second scan.)

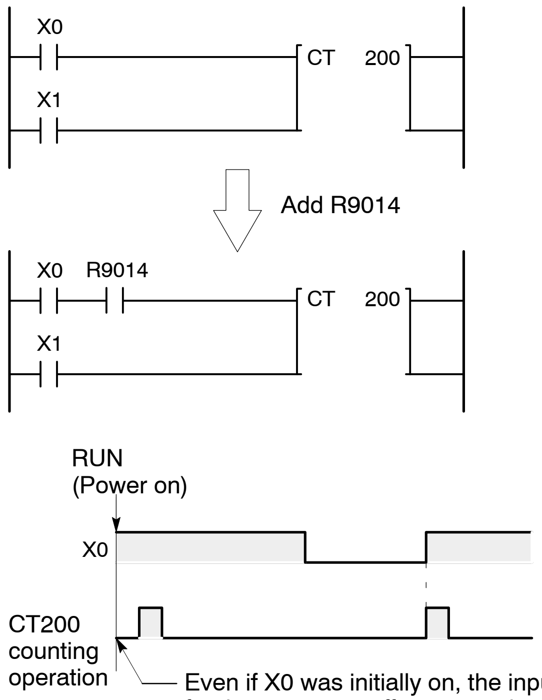
☛ next page

Example 1: DF (leading edge differential) instruction



Even if X0 was initially on, the input condition for the **DF** instruction is off−to−on at the second scan, therefore differential output is obtained.

Example 2: CT (counter) instruction



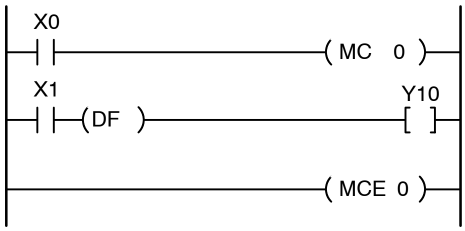
Even if X0 was initially on, the input condition for the counter is off−to−on at the second scan, therefore the count is incremented.

4.3.3 Precautions when Using a Control Instruction

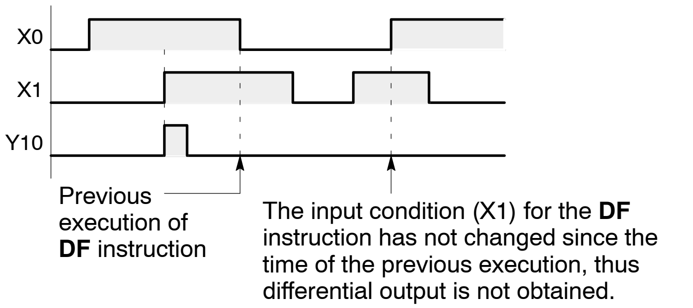
Instructions which leading edge detection compare the condition of the previous execution and the condition of the current execution, and execute the instruction only if the previous condition was off and the current condition is on. In any other case, the instruction is not executed.

When a leading edge detection instruction is used with an instruction which changes the order of instruction execution such as **MC**, **MCE**, **JP** or **LBL**, the operation of the instruction may change as follows depending on input timing. Take care regarding this point.

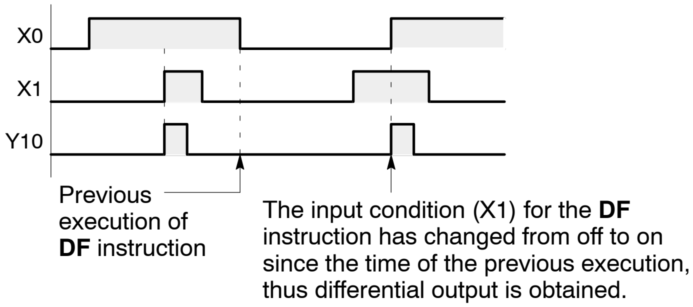
Example 1: Using the DF instruction between MC and MCE instructions



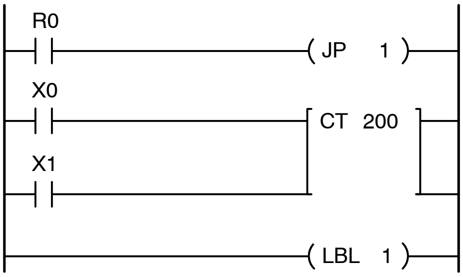
Time chart 1



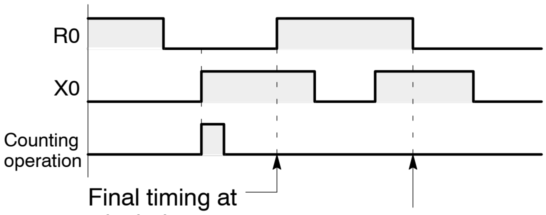
Time chart 2



Example 2: Using the CT instruction between JP and LBL instructions



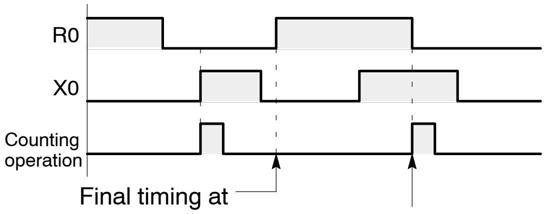
Time chart 1



Final timing at which the previous **JP** instruction was not executed

The count is not incremented, because the final timing at which the previous **JP** instruction was not executed has not been changed, and the execution condition X0 for the counter input has not changed.

Time chart 2



Final timing at which the previous **JP** instruction was not executed

The count is incremented, because the count input changed from off to on after the final timing at which the previous **JP** instruction was not executed.

4.4 Operation Errors

4.4.1 Operation Errors

An operation error is a condition in which operation is impossible when a high-level instruction is executed.

When an operation error occurs, the ERROR LED will light (for FP0, ERROR/ALARM LED will blink), and the operation error flags (R9007 and R9008) will turn on.

The operation error code K45 (H2D) is set at special data register DT9000/DT90000.

The error address is stored in special data registers DT9017 and DT9018/DT90017 and DT90018.

- With the FP0 C10, C14, C16, C32/FP−e, the self−diagnosis error codes are stored in DT9000, and addresses at which errors occurred are stored in DT9017 and DT9018.
- With the FP0R/FP0 T32C/FPΣ/FP−X/FP2/FP2SH/FP10SH, the self−diagnosis error codes are stored in DT90000, and addresses at which errors occurred are stored in DT90017 and DT90018.

Types of operation error

Address error

The memory address (number) specified by index modification is outside the area which can be used

BCD data error

Operation is attempted on non−BCD data when an instruction handling BCD data is executed, or BCD conversion is attempted on data which is not within the possible conversion range.

Parameter error

In an instruction requiring the specification of control data, the specified data is outside the possible range.

Over area error

The data manipulated by a block instruction exceeds the memory range.

4.4.2 Operation Mode when an Operation Error Occurs

Normally, the operation stops when an operation error occurs.

However, when you set system register 26 to “continuation” (K1), the CPU operates even if an operation error occurs.

System registers are specified as described below.

Using programming tool software

1. **Set the mode of the CPU to PROG.**
2. **Select the “Option” in “PLC Configuration” option from the menu.**
3. **On the “PLC Configuration” menu, select “Action on error”. This displays system registers 20 to 28.**
4. **The check of system register 26 is removed.**
5. **Press the “OK” to write the setting to the PLC.**

Using FP programmer II Ver. 2

1. **Set the mode of the CPU to PROG.**
2. **Press the keys on the FP programmer II, as shown below.**

   ACLR (–) OP 5 0 ENT

3. **Specify the register number (26) for the parameter to be set and read the parameter.**
   The value set in the selected system register 26 will be displayed.

   2 6 READ ▼

4. **To change a set value, press the (HELP) “CLR” key and write the K1 parameter.**

4.4.3 Dealing with Operation Errors

Procedure:

1. Check the location of the error.

Check the address where the error occurred, which is stored in DT9017 and DT9018 or in DT90017 and DT90018, and make sure the application instruction for that address is correct and appropriate.

2. Clear the error.

Use a programming tool to clear the error. (If the mode selector is set to RUN, RUN will resume as soon as the error is cleared.)

In the “STATUS DISPLAY” menu of the programming tool software (NPST−GR Ver. 3.1 or later), press the “F3” key.

In FP Programmer II, press the following keys.

ACLR | (–) OP | 1 | 1 | 2 | ENT | SHIFT SC | (DELT) INST

An error can be cleared by turning the power off and on in PROG. mode, however, the contents of the operation memory except the hold type data will be cleared.

An error can also be cleared by executing the self-diagnostic error set instruction **F148 (ERR)**.

4.4.4 Points to Check in Program

This is an example of a program in which an operation error is likely to occur.

Check if an extraordinarily large value or negative value was stored in the index register.

When a data register is modified using an index register

```
  X0
──┤ ├──[F0 MV, DT0, IXDT0          ]
```

In this case, index register (IX) modifies the address of data register DT0. If data in IX is larger than the last address of the data register, an operation error will occur. The same is true when the contents of IX are negative value.

Is there any data which cannot be converted using BCD e BIN data conversion?

When BCD−to−BIN conversion is attempted

```
  X0
──┤ ├──[F81 BIN, DT0, DT100         ]
```

In this case, if DT0 contains a hexadecimal number with one of the digits A through F such as 12A4, the data conversion will be impossible and an operation error will result.

When BIN−to−BCD conversion is attempted

```
  X0
──┤ ├──[F80 BCD, DT1, DT101         ]
```

In this case, if DT1 contains a negative value or a value greater than K9999, an operation error will occur.

Check if the divisor of a division instruction is K0.

```
  X0
──┤ ├──[F32 %, DT0, DT100, DT200 ]
```

In this case, if the content of DT100 is K0, an operation error will occur.

4.5 Handling Index Registers

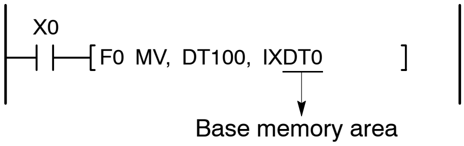
4.5.1 Index Registers

Index registers are used for indirect specification of values to number (addresses) and operands in relays and memory areas. (This is also called "index modification".)

Add the index register to the relay, memory area, or constant you want to modify, and then write the modifying value (16-bit data) to the index register. The FP0 and FP−e have two points, IX and IY. The FP0R, FPΣ, FP−X, FP2, FP2SH and FP10SH have 14 points, I0 to ID.

To modify a 32-bit constant, write the 32-bit data to two words of the index register.

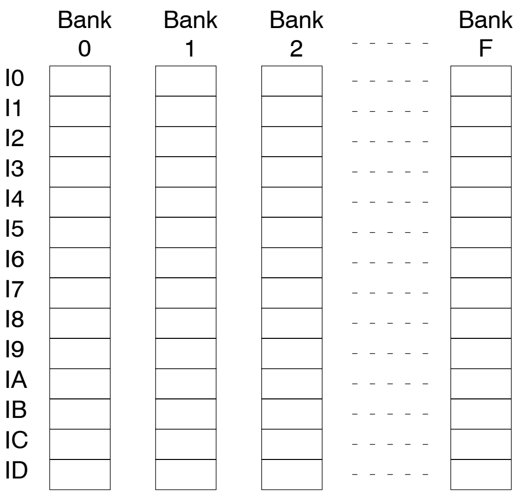
Example: **Transferring the contents of data register DT100 to the number specified by the contents of an index register.**



In this example, the number of the destination data register varies depending on the contents of IX with DT0 acting as a base. For example, when IX is K10, the destination will be DT10, and when IX is K20, the destination will be DT20.

In this way, index registers allow the specification of multiple memory areas with a single instruction, and thus index registers are very convenient when handling large amounts of data.

Changing banks in an index register of the FP2SH and FP10SH makes it possible to increase the number of points used in a program from 14 to a maximum of 224 (14 points, 16 banks).



4.5.2 Memory Areas Which can be Modified with Index Registers

Index registers can be used to modify other types of memory areas in addition to data registers DT.

IXWX0, IXWY1, IXWR0, IXSV0, IXEV2, I0WX10, I2WY1, I3WR0, IASV0, IBEV2

Constants can also be modified.

IXK10, IXH1001

In the FP2SH/FP10SH, the relay numbers can be modified.

I0X0, IAR10

In the FP2/FP2SH/FP10SH, an index register can be modified using another index register.

In the FP0/FP−e, an index register cannot modify another index register.

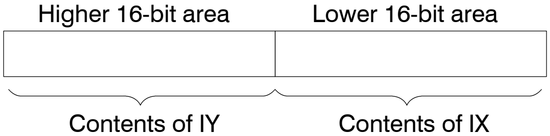
Possibility: I0ID

Impossibility: IXIY, IXIX (except FP2/FP2SH/FP10SH)
I0I0, IAIA (for FP2/FP2SH/FP10SH)

When a 32-bit constant is modified, the specified number and the following number are used in combination to handle the data as a 32-bit data.

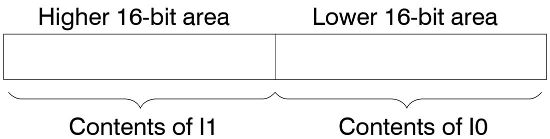
The result of the modification is a 32-bit data.

In the FP0/FP−e



When using index modification with an instruction which handles 32-bit data, specify with IX.

In the FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH (example of specify with I0)



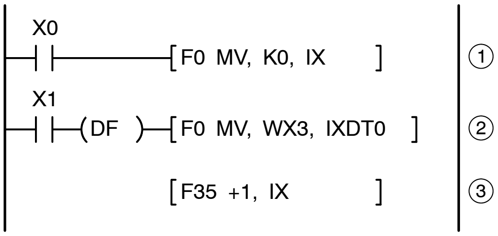
When modifying a 32-bit number, do not specify ID. Be aware that a syntax error will not occur even if this is not specified.

4.5.3 Example of Using an Index Register

Repeatedly reading in external data

With the FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH, any value between I0 and ID should be specified in place of IX.

Example: **Writing the contents of word external input relay WX3 to a sequence of data registers beginning from DT0.**

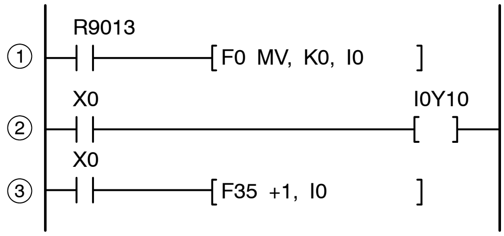


1. **When X0 turns on, K0 is written to index register IX.**
2. **When the X1 turns on, the contents of WX3 is transferred to the data register specified by IXDT0.**
3. **Add 1 to IX.**
   In this case, the contents of IX will change successively, and the destination data register will be as follows.

Input times of X1	Contents of IX	Destination data register
1st	0 → 1	DT0
2nd	1 → 2	DT1
3rd	2 → 3	DT2
:	:	:

Repeatedly changing the output destination (for FP2/FP2SH/FP10SH only)

Example: **Changing the output destination successively each time X0 turns on**



① **K0 is initially written to index register I0.**

② **When the X0 turns on, the first time Y10 will turn on.**

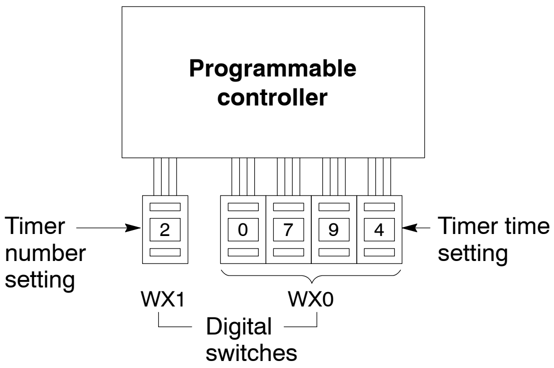
③ **Add 1 to the value of I0. From this point on, the output destinations successively change as follows each time X0 turns on.**

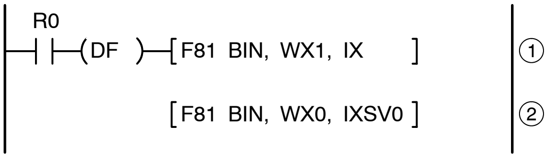
Input times of X0	Content of I0	Output destination
1st	0	Y10
2nd	1	Y11
3rd	2	Y12
:	:	:

Inputting and outputting data based on a number specified by an input

With the FP0R/FPΣ/FP−X/FP2/FP2SH/FP10SH, any value between I0 and ID should be specified in place of IX.

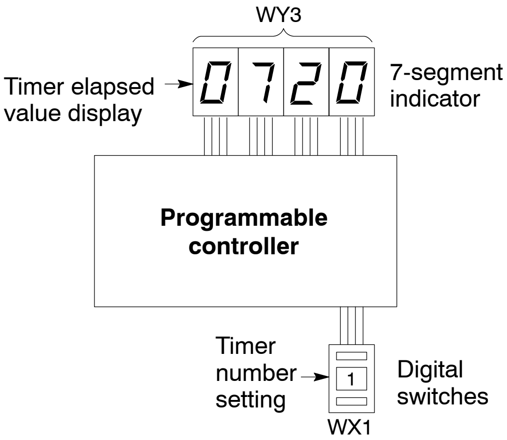
Example 1: Setting a timer number specified by a digital switch

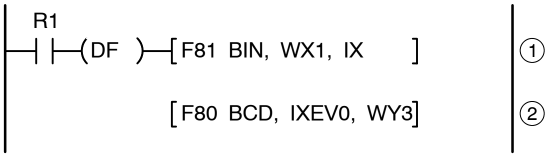




1. **Convert the BCD timer number data in WX1 to binary and set it in index register IX.**
2. **Convert the BCD timer set value in WX0 to binary and stored in the timer set value area SV specified by contents of IX.**

Example 2: External output of the elapsed value in a timer number specified by a digital switch





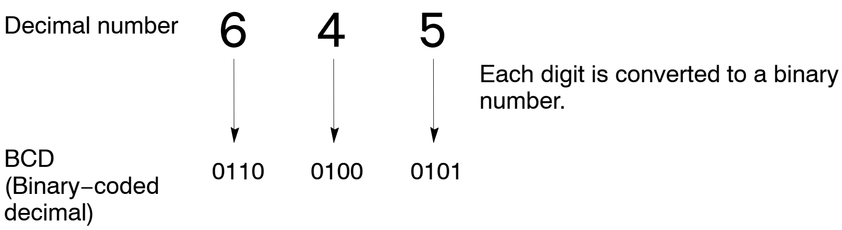
1. **Convert the BCD timer number data in WX1 to binary, and set it in index register IX.**
2. **Convert the elapsed value data EV in the timer specified by IX to BCD, and output it to word external output relay WY3.**

4.6 Handling BCD Data

4.6.1 BCD Data

BCD is an acronym for binary−coded decimal, and means that each digit of a decimal number is expressed as a binary number.

Example: Expressing a decimal number in BCD



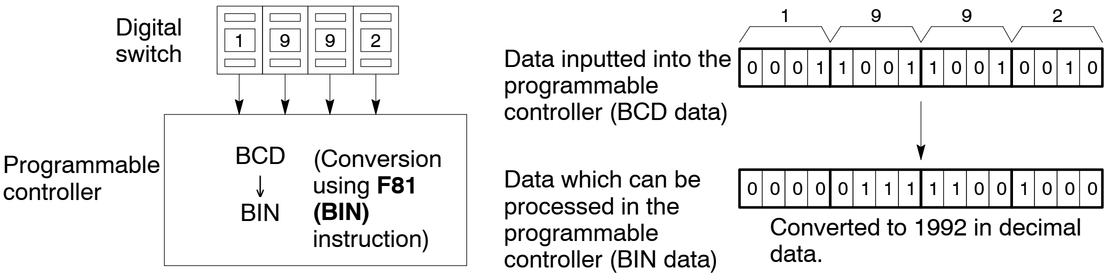
4.6.2 Handling BCD Data in the Programmable Controller

When inputting data from a digital switch to the programmable controller or outputting data to a 7−segment display (with decoder), the data must be in BCD form. In this case, use a data conversion instruction as shown in the examples at below.

BCD arithmetic instructions (**F40** through **F58**), also exist which allow direct operation on BCD data, however, it is normally most convenient to use BIN operation instructions (**F20** through **F38**) as operation in the programmable controller takes place in binary.

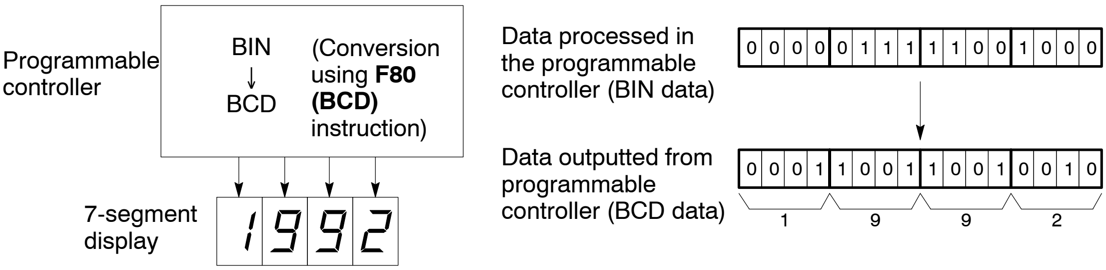
Input from a digital switch

Use the BCD−to−BIN conversion instruction **F81 (BIN)**.



Output to a 7–segment display (with decoder)

Use the BIN–to–BCD conversion instruction **F80 (BCD)**.



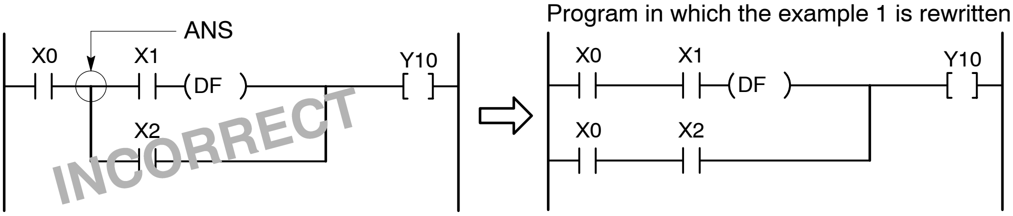
4.7 Precautions for Programming

Programs which do not execute correctly

Do not write the following programs as they will not execute correctly.

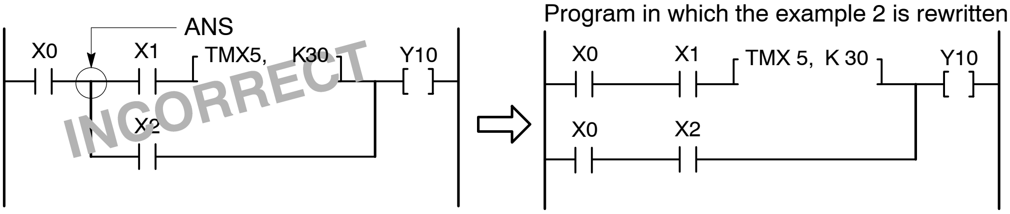
Program example 1:

Program in which the example 1 is rewritten



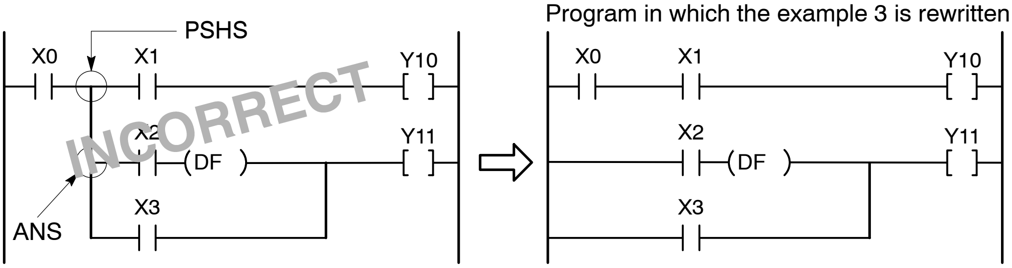
Program example 2:

Program in which the example 2 is rewritten



Program example 3:

Program in which the example 3 is rewritten



When a combination of contacts are set as the trigger (execution condition) of a differential instruction (**DF**) or timer instruction, do not use an AND stack (ANS), push stack (**PSHS**), read stack (**RDS**), or pop stack (**POPS**) instruction.

4.8 Rewrite Function During RUN

4.8.1 Operation of Rewrite During RUN

How operation of rewrite during RUN

Rewriting programs can be executed even in RUN mode. When a rewrite is attempted during RUN, the tool service time is temporarily extended, program rewriting is performed, and operation is resumed without the need to change the mode. For this reason, the time of the scan during the RUN rewrite extends from several ms to several hundreds of ms.

Operation during rewrite

External output (Y) is held.

External input (X) is ignored.

The timer (T) stops the clock.

Rise and fall changes in the inputs of differential instructions (DF), counter instructions (CT), and left/right shift registers are ignored.

Interrupt functions are stopped.

Internal clock relays (special internal relays) are also stopped.

Pulse output is stopped during the rewrite.

Set values for timer/counter instructions

All set values specified with decimal constants (K) in timer and counter instructions are preset in the corresponding set value areas (SV). Values in the elapsed value area (EV) do not change.

Operation of rewrite during RUN completed flag

The rewrite during RUN completed flag (R9034) is a special internal relay that goes on for only the first scan following the completion of rewriting in the RUN mode. It can be used instead of the initial pulse relay following a change in the program.

4.8.2 Cases Where Rewriting During Run is not Possible

When the timeout error message is indicated:

Even if the timeout error message is indicated, it is highly possible that the program in PLC has been already rewritten. Carry out the following operations.

1. When ladder symbol mode

As a ladder editing is left, set it to the offline edit mode. Complete the program conversion in the tool software, and then change to the online edit mode to check.

2. When boolean mode

A ladder editing is cleared.

Set it to the offline edit mode and carry out the editing operation again. After the operation, change to the online edit mode to check.

When the timeout error occurs using the through mode in GT series programmable display.

Extend the timeout time of the programmable display using the GTWIN.
(The default setting is 5 seconds.)

Select “Transfer” from “File” in the menu bar. The “transfer data” screen will open. Select “Condition” to open “Communication Setting” screen. Change the value for “Timeout”.Click “OK” button to complete the change of setting.

For FP0/FP‑e/FPΣ/FP‑X/FP0R

Cases where rewriting is not possible during RUN

1. When the result of rewriting is a syntax error.

<Example>
When executing the rewriting which does not form the following pair of instructions.
1. Step ladder instructions (SSTP/STPE)
2. Subroutine instructions (SUB/RET)
3. Interrupt instructions (INT/IRET)
4. JP/LBL

5. LOOP/LBL
6. MC/MCE

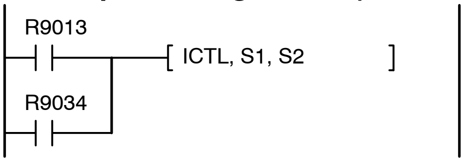
Also, rewritng is not possible during RUN in case of other syntax errors.

2. During the forced input/output operation

Interrupt restrictions

When using interrupt, high−speed counter, pulse output or PWM output functions, do not perform a rewrite during RUN.
If a rewrite during RUN is executed, the operation as below will be performed. Exercise caution.

1. **Interrupt programs will be disabled. Enable by executing an ICTL instruction once again.**
   <Example> Using R9034 (rewrite during RUN completed flag)



2. **The high−speed counter will continue to count.**
   Target value match on/off instructions (F166/F167) will continue.
   Coincidence interrupt programs will be disabled when the F166/F167 instruction is running.
3. **Pulse output and PWM output will be stopped.**

State	Instruction number	Name
Continue	F171 (SPDH)	Pulse output (with channel specification) (Home position return)
Stop	F172 (PLSH)	Pulse output (with channel specification) (JOG operation)
Stop	F173 (PWMH)	PWM output (with channel specification)
Continue	F174 (SP0H)	Pulse output (with channel specification) (Selectable data table control operation)
Continue	F175 (SPSH)	Pulse output (Linear interpolation)
Stop	F176 (SPCH)	Pulse output (Circular interpolation)

4. **The fixed time sampling trace will not be stopped.**

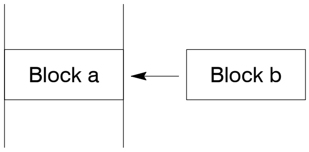
For FP2/FP2SH

Instructions that cannot be added or deleted by rewriting during RUN
1. Step ladder instructions (SSTP/STPE)
2. Subroutine instructions (SUB/RET)
3. Interrupt instructions (INT/IRET)
4. Control instructions (ED/LBL)
* The LBL instruction can be inserted/written, but cannot be deleted/erased.

Instructions that cannot be added or deleted during subprograms
1. JP/LBL
2. LOOP/LBL
3. MC/MCE

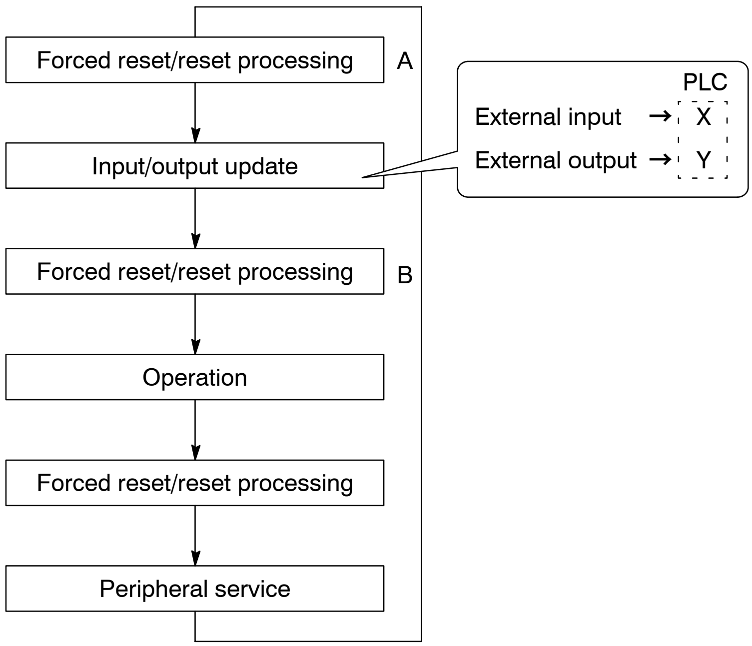
Cases where rewriting is not possible during RUN
1. When a syntax error occurred.
2. When the forced input/output operation is running.

4.8.3 Procedures and Operation of Rewrite During RUN

Item		**FPWIN GR Ladder symbol mode**	**FPWIN GR Boolean mode**
Rewrite procedure		Maximum jof 128 steps.Changes are performed by block.When PG conversion is executed online, the program will be rewritten. 	Rewriting performed by step.Caution is required as rewriting takes place simultaneously with the change.
Operation of each instruciton	**OT/KP**	If an instruction written in block a is deleted in block b, the condition before the rewrite will be held.s	If an instruction written in block a is deleted in block b, the condition before the rewrite will be held. Y contact relays which are on bill be held in the on sattus. To turn them off in the RUN mode, use forced output.
	TM/CT	– If an instruction written in block a is deleted in block b, the condition before the rewrite will be held. – Set values specified by K constants in TM/CT instructions are preset in all of the corresponding SV's in the program. (Elapsed values EV do not change.)	– If an instruction written in block a is deleted in block b, the condition before the rewrite will be held. – Set values specified by K constants in TM/CT instructions are preset in all of the corresponding SV's in the program. (Elapsed values EV do not change.)
	Fun High–level instructions	If an instruction written in block a is deleted in block b, the condition before the rewrite will be held.	– If deleted, the output memory area will be held.
	MC/MCE	When writing MC/MCE instructions, be sure to write the instructions as a pair.	Writing or deleting a single instruction during RUN is not possible. Write or delete the instruction in FPWIN GR ladder symbol mode.
	CALL/SUB/ RET	A subroutine is a program appearing between SUBn and RET instructions. Be sure to write it to an address which follows the ED instruction.	Write in the order: RET, SUB, CALL Delete in the order: CALL, SUB, RET
	INT/IRET	An interrupt program is an program appearing between INTn and IRET instructions. Be sure to write it to an address which follows the ED instruction.	Write in the orde: IRET, INTDelete in the order: INT, IRET
	SSTP/STPE	A distance with the same number cannot be defined twice. An SSTP instruction cannot be written in a subprogram.	Writign and deletion of a single instruction is not possible for a program with no step ladder area.Write or delete both instructions simultaneously in FPWIN GR ladder symbol mode. In the case of an SSTP instruction only, writing and deletion of a single instruction is possible for a program with a step ladder area.
	JP/LOOP/ LBL	Be sure to write the instruction for setting the loop number before LBL–LOOP instructions.	Write in the order: JP–LBL or LOOP–LBLDelete in the order: LBL–JP or LBL–LOOP

4.9 Processing During Forced Input and Output

4.9.1 Processing when forced input/output is initiated during RUN



1. Processing of external input (X)

Regardless of the state of the input from the input device, forced on/off operation will take precedence at a contact specified for forced input/output in the above procedure B. At this time, the input LED will not blink, however, the area of input X in the operation memory will be rewritten.

Contacts not specified will read in the on/off state according to the condition of the inptu from the input device.

2. Processing of external output (Y)

Regardless of the result of operation, forced on/off will take precedence at a contact specified for forced input/ouput in the above procedure A. At this time, the area of output Y in the operation memory will be forcedly rewritten. External output will take place according to the input/output update timing in the above diagram.

The on/off state of contacts not specified will be determined by the operation result.

3. Processing of Timer (T) and Counter (C)

Regardless of the timer/counter input condition, forced on/off operation will take precedence at a contact specified for forced input/output. At this time, the contact of the timer (T) or counter (C) in the operation memory will be rewritten. Timing and counting will not take place during control.

The on/off state of contacts not specified will be determined by the operation result.

Operation during operation

For small–sized PLCs FP0R, FP0, FPΣ and FP–X

Forced relay R and output Y are rewritten according to the results of operation.

For medium–sized PLCs FP2 and FP2SH

For the relay and output Y specified by OT or KP instruction, the value of the forced processing has a priority. When rewritten by a high–level instruction, the result of the instruction has a priority.

4.10 Second Program Area (FP2SH, FP10SH)

Explanation of operation method for FP2SH and FP10SH

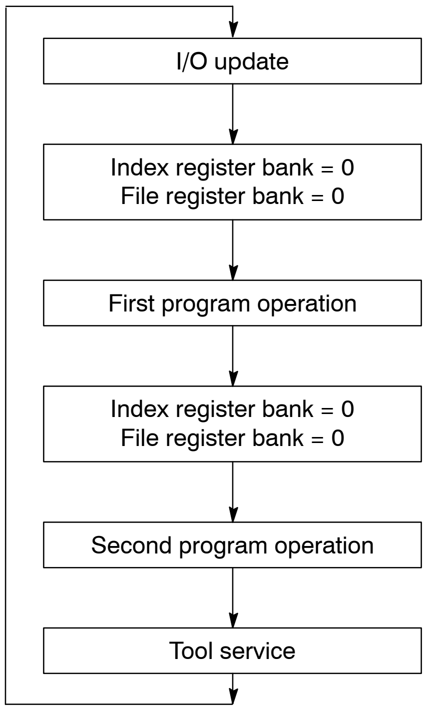
For the type of FP2SH of which program capacity exceeds 60k steps and for the type of FP10SH of which program capacity exceeds 60k steps if the memory is added, the program area is divided into the first program area and the second program area. The divided programs are separate program units, however, uploading and downloading with TOOL is performed simultaneously. There are following restrictions for the operation devices.

Device and function	First program	Second program
Bits X, Y, R, LWords WX, WY, WR, WL, DT, Ld, In, Fl	Common device	
SALL Subroutine call	The subroutine of the second program cannot be called up.	The subroutine of the first program cannot be called up.
SUB Subroutine entry	100	100
JP Jump	255 (However, jumping to the second program is not possible.)	255 (However, jumping to the first program is not possible.)
LBL Label	255	255
INT Interrupt program	Can be written in the first program only.	Cannot be used.
SSTP Step ladder	Can be written in the first program only.	Cannot be used.
MC, MCE Master control	255 (A pair must be formed in the first program.)	255 (A pair must be formed in the second program.)

Syntax check

- For SUB, JP, LBL, MC and MCE, the checks are performed as the above table.
- The duplicated output of OT and KP instructions and the duplicated use of timer and counter instructions are checked throughout the first and second programs.

Operation flow diagram of FP2SH and FP10SH



As shown in the left diagram, the second program is executed after the first program has completed.

At the points when the first program or the second program starts, the following settings will be automatically selected.
(Index register bank = 0
File register bank = 0)

Program switching method with FPWIN GR

Select "Edit" → "Switch Programming Area" in the menu bar to change the program area.

Note

The monitor data in the tool software is the data monitor when both first and second programs complete.

Chapter 5

Appendix

Appendix 5-1

5.1 System Registers / Special Internal Relays / Special Data Registers

Precation for System Registers

What is the system register area

- System registers are used to set values (parameters) which determine operation ranges and functions used. Set values based on the use and specifications of your program.
- There is no need to set system registers for functions which will not be used.

Type of system registers
The registers to be used depend on each PLC.

(1) Allocation of user memory (System registers 0, 1 and 2)
These registers set the size of the program area and file register area, allowing the user memory area to be configured for the environment used. The size of the memory area will vary depending on the type.
(2) Allocation of timers and counters (System register 5)
The number of timers and counters is set by specifying the starting counter number.
(3) Hold/non-hold type setting (System registers 6 to 18)
When these registers are set to “hold type”, the values in the relays and data memory will be retained even if the system is switched to PROG. mode or the power is turned off. If set to “non-hold type”, the values will be cleared to “0”.
(4) Operation mode setting on error (System registers 4, 20 to 28)
Set the operation mode when errors such as battery error, duplicated use of output, I/O verification error and operation error occur.
(5) Time settings (System registers 30 to 34)
Set time-out error detection time and the constant scan time.
(6) Remote I/O operation settings (System registers 35 and 36)
These registers are used to select whether or not to wait for a slave station connection when the remote I/O is started, and the remote I/O update timing.
(7) MEWNET-W0/MEWNET-W/P PLC link settings (System registers 40 to 47, 50 to 55, and 57)
These settings are for using link relays and link registers for MEWNET-W0/MEWNET-W/P PC(PLC) link communication.
Note) The default value setting is “no PC(PLC) link communication”.
(8) MEWNET-H PC(PLC) link settings (System register 49)
Set the data size to be processed during one scan in the MEWNET-H PC(PLC) link communication.
(9) Input settings (System registers 400 to 406)
When using the high-speed counter function, pulse catch function or interrupt function, set the operation mode and the input number to be used for the function.
(10) Input time constant settings (FP1/FP-M System registers 404 to 407)
Changing the input signal width to be loaded enables to prevent the malfunctions caused by chattering or noises.
(11) Number of temperature input averaging process settings (System register 409)
The number of averaging times can be set in order to even out the variation in the input thermocouple values. For normal use it, set the number of times to t least twenty. For default value “0”, the number of average processing times is 20.
(12) Tool and COM. ports communication settings (System registers 410 to 421)
Set these registers when the Tool port,and COM1 and COM2 ports are to be used for computer link, general-purpose serial communication, PC(PLC) link, and modem communication.Note that the default setting is computer link mode.

Checking and changing the set value of system register

If you are going to use a value which is already set(the value which appears when read), there is no need write it again.

Using programming tool software

Produce:

1. Set the control unit in the PROG mode.
2. Option ->PLC Configuration
3. When the function for which setting are to be entered is selected in the PLC Configuration dialog box,the value and setting status for the selected system register are displayed.
   To change the value and setting status,write in the new value and /or select the setting status.
4. To register these settings,choose OK

Precautions for system register setting

-System register settings are effective from the time they are set.
 However, input settings,tool port,COM port,and modem connection settings become effective when the mode is changed from PROG. to RUN. With regard to the modem connection setting, when the power is turned off and on or when the mode is changed from PROG. to RUN, the controller sends a command to the modem which enables it for reception.

-When the initialized operation is performed, all set system register values (parameters) will be initialized

5.1.1 Table of System Registers for FP0

Content of system register settings

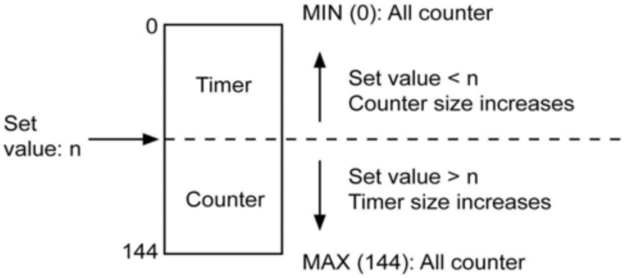
1. Setting the timers and counters (System register 5)

By indicating the counter start number, the timer and counter are split into two areas. The timer and counter together total 144 points, and the default value for th split is 100. Thus the point allotment is as shown in the table below.

Timer	100 points (No. 0 to No. 99)
Counter	44 points (No. 100 to No. 143)

Setting example

To increase the number of timers to 120, change the value of system register 5 to K120.



For T32, set the system registers 5 and 6 to the same value. This sets the timer to a non-hold type and counter to a hold type.

By setting system register 5 to “0”, the whole area becomes the counter. Also, by setting it to the value “144”, the whole area becomes the timer.

2. Hold types and non-hold type settings (System registers 6 to 8 and 14)

With the C10/C14/C16/C32/SL1, the areas held in the event of a power supply interruption are fixed at the areas shown in the table below, and the settings for system registers 6 to 8 and 14, will be invalid.

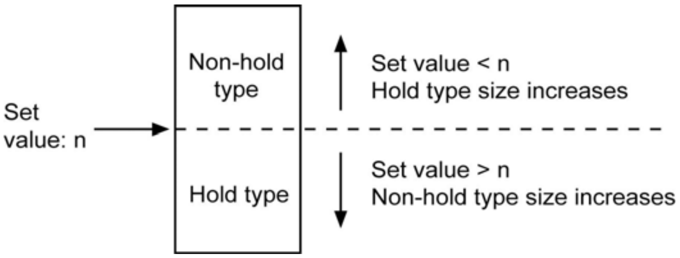
C10/C14/C16

Timer	Non-hold type: All points	
Counter	Non-hold type: From the set value to C139	
	Hold type: 4 points (elapsed values)C140 to C143	
Internal relay	Non-hold type:	976 points (R0 to R60F) 61 words (WR0 to WR60)
	Hold type:	32 points (R610 to R62F) 2 words (WR61 to WR62)
Data register	Non-hold type: 1652 words (DT0 to DT1651)	
	Hold type: 8 words (DT1652 to DT1659)	

C32/SL1

Timer	Non-hold type: All points	
Counter	Non-hold type: From the set value to C127	
	Hold type: 16 points (elapsed values)C128 to C143	
Internal relay	Non-hold type:	880 points (R0 to R54F) 55 words (WR0 to WR54
	Hold type:	128 points (R550 to R62F) 8 words (WR55 to WR62)
Data register	Non-hold type: 6112 words (DT0 to DT6111)	
	Hold type: 32 words (DT6112 to DT6143)	

With the T32, set each relay and register to a hold type or non-hold type.



For normal situations, set the system registers 5 and 6 to the same value. This sets the timer to a non-hold type and counter to a hold type.

By setting this value to “0”, the whole area becomes hold type. Also, by setting it to the valeu 1 higher than the last number, the whold area becomes non-hold type.

C32/SL1

Type / Area		T32
Timer		All non-hold type
Counter		All hold type
Internal relay	**Non-hold type**	Non-hold type: 10 words (WR0 to WR9)
	Hold type	Hold type: 53 words (WR10 to WR62)
Data register		All hold type

Table of system registers

C10, C14, C16, C32, T32 and SL1 in the table respectively indicate 10-point, 14-point, 16-point, 32-point type and S-LINK type FP0 control units.

<table>
<tr><th>Item</th><th>Add-ress</th><th>Name</th><th>Default value</th><th colspan="2">Descriptions</th></tr>
<tr><td>Alloca-tion of user memory</td><td>0</td><td>Sequence program area capacity</td><td>-</td><td colspan="2">The set values are fixed and cannot be changed.
The stored values vary depending on the type.
K3: 3K words (C10, C14, C16)
K5: 5K words (C32, SL1)
K10: 10K words (T32)</td></tr>
<tr><td rowspan="5">Hold/ Non-hold</td><td>5</td><td>Timer and counter division (setting of starting counter number)</td><td>100
(K100)</td><td>0 to 144
(K0 to K144)</td><td rowspan="2">Set the system registers 5 and 6 to the same value.</td></tr>
<tr><td>6</td><td>Hold type area starting number setting for timer and counter
(Available type: T32)</td><td>100
(K100)</td><td>0 to 144
(K0 to K144)</td></tr>
<tr><td>7</td><td>Hold type area starting number setting for internal relays (in word units)
(Available type: T32)</td><td>10
(K10)</td><td colspan="2">0 to 63 (K0 to K63)</td></tr>
<tr><td>8</td><td>Hold type area starting number setting for data registers
(Available type: T32)</td><td>0
(K0)</td><td colspan="2">0 to 16384 (K0 to K16384)</td></tr>
<tr><td>14</td><td>Hold or non-hold setting for step ladder process
(Available type: T32)</td><td>Non-hold
(K1)</td><td colspan="2">Hold (K10)
Non-hold (K1)</td></tr>
<tr><td rowspan="4">Action on error</td><td>20</td><td>Disable or enable setting for duplicated output</td><td>Disable
(K0)</td><td colspan="2">Disable (will be syntax error) (K0)
Enable (will not be syntax error) (K1)</td></tr>
<tr><td>23</td><td>Operation setting when an I/O verification error occurs</td><td>Stop
(K0)</td><td colspan="2">Stop (K0)
Operate (K1)</td></tr>
<tr><td>26</td><td>Operation setting when an operation error occurs</td><td>Stop
(K0)</td><td colspan="2">Stop (K0)
Operate (K1)</td></tr>
<tr><td>27</td><td>Operation settings when communication error occurs in the remote I/O (S-LINK) system</td><td>Operate
(K1)</td><td colspan="2">Stop (K0)
Operate (K1)</td></tr>
</table>

Note) The setting values of the system registers No. 6, 7, 8 and 14 becomes invalid with the types other than T32.

FP0

Item	Address	Name		Default value		Descriptions
Time setting	31	**Wait time setting for multi-frame communication**		6500 ms (K2600)		10 ms to 81900 ms (K4 to K32760) Used of default setting (K2600/6500 ms) is recommended. set value × 2.5 ms = Wait time setting for multi-frame communication (ms) In programming tool software, enter the time (a number divisible by 2.5). In FP Programmer Ⅱ, enter the set value (equal to the time divided by 2.5).
	34	**Constant value settings for scan time**		0 ms (K0)		2.5 ms to 160 ms (K1 to K64): Scans once each specified time interval. 0 (K0):Normal scan set value × 2.5 ms = Constant value setting for multi-frame communication (ms) In programming tool software, enter the time (a number divisible by 2.5). In FP Programmer Ⅱ, enter the set value (equal to the time divided by 2.5).
Input setting	400	**High-speed counter mode settings (X0 to X2)**	**Setting by progra-mming tool software**	Do not set X0 as high-speed counter.	CH0	Do not set X0 as high-speed counter. 2-phase input (X0, X1) 2-phase input (X0, X1), Reset input (X2) Incremental input (X0) Incremental input (X0), Reset input (X2) Decremental input (X0) Decremental input (X0), Reset input (X2) Individual input (X0, X1) Individual input (X0, X1), Reset input (X2) Direction decision (X0, X1) Direction decision (X0, X1), Reset input (X2)
				Do not set X1 as high-speed counter.	CH1	Do not set X1 as high-speed counter. Incremental input (X1) Incremental input (X1), Reset input (X2) Decremental input (X1) Decremental input (X1), Reset input (X2)

Note1) If the operation mode is set to 2-phase, individual, or direction differentiation, the setting for CH1 is invalid.

Note2) If reset input settings overlap, the setting of CH1 takes precedence.

Note3) If system register 400 to 403 have been set simultaneously for the same input relay, the following precedence order is effective: [High-speed counter] '[Pulse catch]' [Interrupt input].

FP0

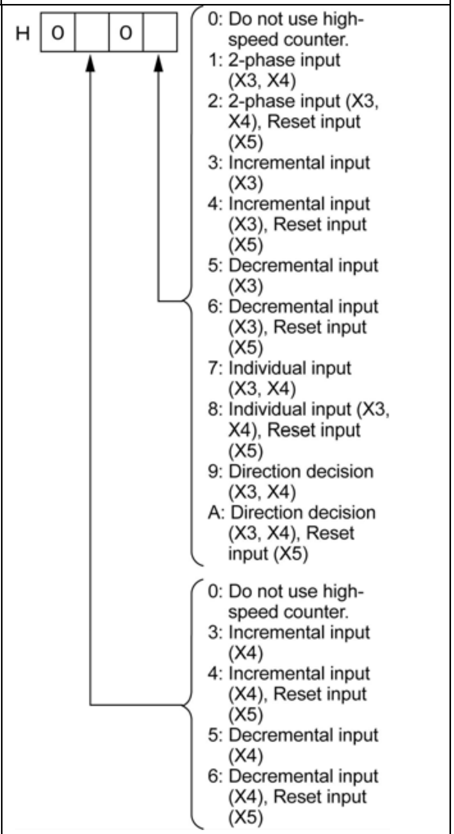
Item	Add-ress	Name		Default value	Descriptions	
Input setting	**400**	**High-speed counter mode settings (X0 to X2)**	**Setting by FP pro-grammer II**	H0	CH0/ CH1	H 0 □ 0 □ 0: Do not use high-speed counter. 1: 2-phase input (X0, X1) 2: 2-phase input (X0, X1), Reset input (X2) 3: Incremental input (X0) 4: Incremental input (X0), Reset input (X2) 5: Decremental input (X0) 6: Decremental input (X0), Reset input (X2) 7: Individual input (X0, X1) 8: Individual input (X0, X1), Reset input (X2) 9: Direction decision (X0, X1) A: Direction decision (X0, X1), Reset input (X2) 0: Do not use high-speed counter. 3: Incremental input (X1) 4: Incremental input (X1), Reset input (X2) 5: Decremental input (X1) 6: Decremental input (X1), Reset input (X2)

Note1) If the operation mode is set to 2-phase, individual, or direction differentiation, the setting for CH1 is invalid.

Note2) If reset input settings overlap, the setting of CH1 takes precedence.

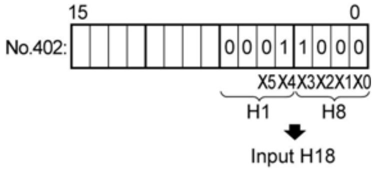
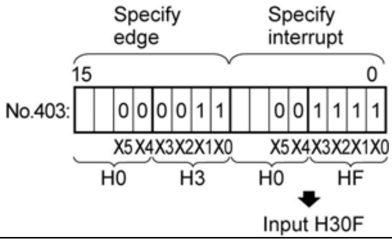
Note3) If system register 400 to 403 have been set simultaneously for the same input relay, the following precedence order is effective: [High-speed counter] ‘[Pulse catch]’ [Interrupt input].

FP0

Item	Add-ress	Name		Default value	Descriptions	
Input setting	401	High-speed counter mode settings (X3 to X5)	Setting by progra-mming tool soft-ware	Do not set X3 as high-speed counter.	CH2	Do not set X3 as high-speed counter. 2-phase input (X3, X4) 2-phase input (X3, X4), Reset input (X5) Incremental input (X3) Incremental input (X3), Reset input (X5) Decremental input (X3) Decremental input (X3), Reset input (X5) Individual input (X3, X4) Individual input (X3, X4), Reset input (X5) Direction decision (X3, X4) Direction decision (X3, X4), Reset input (X5)
				Do not set X4 as high-speed counter.	CH3	Do not set X4 as high-speed counter. Incremental input (X4) Incremental input (X4), Reset input (X5) Decremental input (X4) Decremental input (X4), Reset input (X5)
				H0	CH2/ CH3	

Note1) If the operation mode is set to 2-phase, individual, or direction differentiation, the setting for CH3 is invalid.
Note2) If reset input settings overlap, the setting of CH3 takes precedence.
Note3) If system register 400 to 403 have been set simultaneously for the same input relay, the following precedence order is effective: [High-speed counter] ‘[Pulse catch]’ [Interrupt input].

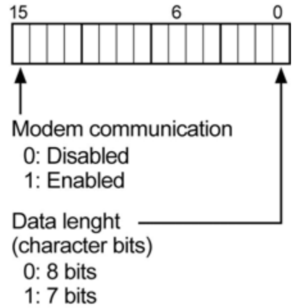
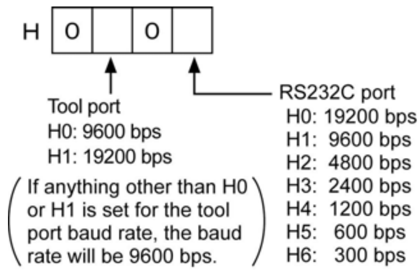
FP0

Item	Add-ress	Name	Default value	Descriptions
Input setting	**402**	**Pulse catch input function settings**	Not set (H0)	X0 X1 X2 X3 X4 X5 The checked contacts are set as pulse catch input. In FP Programmer II, enter the above settings in hexadecimal. Example: When X3 and X4 are set as pulse catch input  Settings X6 and X7 are invalid.
	403	**Interrupt input settings**	Not set (H0)	Using FPWIN GR X0 X1 X2 X3 X4 X5 The checked contacts are set as interrupt input. X0 X1 X2 X3 X4 X5 Specify the effective interrupt edge. (When set: ON→OFF is valid) Using FP Programmer II Example: When setting inputs X0, X1, X2 and X3 as interrupts, and X0 and X1 are set as interrupt inputs when going from on to off. 

Note1) With the TOOL software, “0” or “1” is set for each bit on the screen in the setting for system register 403.

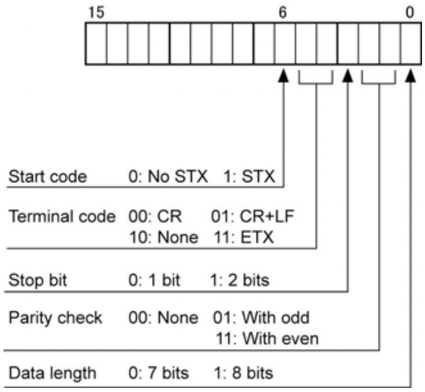
Note2) If system register 400 to 403 are set simultaneously for the same inptu relay, the following precedence order is effective:
[High-speed counter] ‘[Pulse catch]’ [Interrupt input].
When the high-speed counter is being used in the incremental input mode, even if input X0 is specified as an interrupt input and as pulse catch input, those settings are invalid, and input X0 functions as counter input for the high-speed counter.
No. 400: H1 a This setting will be valid. No. 402: H1 No. 403: H1

FP0

Item	Address	Name		Default value	Descriptions
Tool port setting	**410**	**Unit No. setting for tool port (when connecting C-NET)**		1 (K1)	1 to 32 (K1 to K32)
	411	**Communication format setting for tool port**		Modem: Disabled Data length: 8 bits (H0)	Using FPWIN GR Modem: Disable/Enable Data length: 7 bits/8 bits Using FP programmer II Specify the setting contents using H constants.  When connecting a modem, set the unit number to 1 with system register 410.
	414	**Baud rate setting**	**Setting by programming tool software**	9600 (H0)	0: 9600 bps 1: 19200 bps
Tool port/ RS232C port setting	**414**	**Baud rate setting for tool port and RS232C port**	**Setting by FP programmer II**	H1	 If 19200 bps is set for both the tool port and RS232C port, H100 should be written.

FP0

<table>
<tr><th>Item</th><th>Add-
ress</th><th colspan="2">Name</th><th>Default
value</th><th>Descriptions</th></tr>
<tr><td rowspan="5">RS232C
port
setting</td><td>412</td><td colspan="2">Selection of operation</td><td>Not used
(K0)</td><td>Using FPWIN GR
Not used
Computer link
General-purpose communication
Using FP programmer II
K0: RS232C port is not used.
K1: Computer link mode (when connecting C-NET)
K2: Serial data communication mode (general port)</td></tr>
<tr><td>413</td><td colspan="2">Communication format</td><td>Start code: None
Terminal code: CR
Stop bit: 1 bit
Paritycheck: With odd
Data length: 8 bits
(H3)</td><td>Using FPWIN GR
- Data length: 7 bits/8bits
- Parity check: None/Odd/Even
- Stop bit: 1/2
* The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”.
- Terminator CR/CR+LF/None/ETX
- Start code: STX not exist/STX exist

Using FP programmer II
Specify the setting contents using H constants.

</td></tr>
<tr><td>414</td><td>Baud rate setting</td><td>Setting by programming tool software</td><td>9600
(H1)</td><td>19200 bps
9600 bps
4800 bps
2400 bps
1200 bps
600 bps
300 bps</td></tr>
<tr><td>415</td><td colspan="2">Unit no. (when connecting C-NET)</td><td>1
(K1)</td><td>1 to 32 (K1 to K32)</td></tr>
<tr><td>416</td><td colspan="2">Modem connection</td><td>Disable
(H0)</td><td>Using FPWIN GR
Diable/Enable
Using FP programmer II
H0: Modem disabled
H8000: Modem enabled</td></tr>
</table>

Item	Add-ress	Name		Default value	Descriptions
	417	**Starting address setting for received buffer**		0 (K0)	C10C/C14C/C16C: 0 to 1659 (K0 to K1659) C32C/SL1: 0 to 6143 (K0 to K6143) T32C: 0 to 16383 (K0 to K16383)
	418	**Capacity setting for reception buffer**	**C10C/C14C/C16C**	1660 (K1660)	0 to 1660 (K0 to K1660)
			C32C/SL1	6144 (K6144)	0 to 6144 (K0 to K6144)
			T32C	16384 (K16384)	0 to 16384 (K0 to K16384)

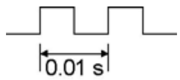
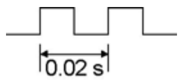
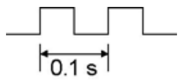
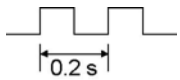
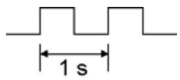
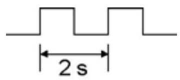
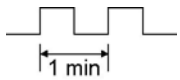
5.1.2 Table of Special Internal Relays for FP0

The special internal relays turn on and off under special conditions. The on and off states are not output externally. Writing is not possible with a programming tool or an instruction.

FP0

Address	Name	Description
R9000	**Self-diagnostic error flag**	Turns on when a self-diagnostic error occurs. The self-diagnostic error code is stored in DT9000.
R9001 to R9003	**Not used**	**-**
R9004	**I/O verification error flag**	Turns on when an I/O verification error occurs. The position number of the I/O where the verification error was occurred is stored in DT9010.
R9005, R9006	**Not used**	-
R9007	**Operation error flag (hold)**	Turns on and keeps the on state shen an operation error occurs. The address where the error occurred is stored in DT9017. (Indicates the first operation error which occurred).
R9008	**Operation error flag (non-hold)**	Turns on for an instant when an operation error occurs. The address where the operation error occurred is stored in DT9018. The contents change each time a new error occurs.
R9009	**Carry flag**	Turns on for an instant, - when an overflow or underflow occurs. – when “1” is set by one of the shift instructions.
R900A	**> Flag**	Turns on for an instant when the compared results become larger in the “F60 (CMP) to F63 (DWIN) comparison instructions.”
R900B	**= Flag**	Turns on for an instant, - when the compared results are equal in the comparison instructions (F60 to F63). - when the calculated results become 0 in the arithmetic instructions.
R900C	**< Flag**	Turns on for an instant when the compared results become smaller in the “F60 (CMP) to F63 (DWIN) comparison instructions.
R900D	**Auxiliary timer contact**	Turns on when the set time elapses (set value reaches 0) in the timing operation of the F137(STMR)/F183(DSTM) auxiliary timer instruction. It turns off when the trigger for auxiliary timer instruction turns off.
R900E	**Tool port error flag**	This turns on when an error occurs during communication with a programming tool.
R900F	**Constant scan error flag**	Turns on when the scan time exceeds the time specified in system register 34 during constant scan execution.
R9010	**Always on relay**	Always on.
R9011	**Always off relay**	Always off.
R9012	**Scan pulse relay**	Turns on and off alternately at each scan.

FP0

Address	Name	Description
R9013	**Initial on pulse relay**	Turns on only at the first scan in the operation. Turns off from the second scan and maintains the off state.
R9014	**Initial off pulse relay**	Turns off only at the first scan in the operation. Turns on from the second scan andmaintains the on state.
R9015	**Step ladder initial on pulse relay**	Turns on for an instant only in the first scan of the process the moment step ladder process is opened.
R9016, R9017	**Not used**	**-**
R9018	**0.01 s clock pulse relay**	Repeats on/off operations in 0.01 s cycles. 
R9019	**0.02 s clock pulse relay**	Repeats on/off operations in 0.02 s cycles. 
R901A	**0.1 s clock pulse relay**	Repeats on/off operations in 0.1 s cycles. 
R901B	**0.2 s clock pulse relay**	Repeats on/off operations in 0.2 s cycles. 
R901C	**1 s clock pulse relay**	Repeats on/off operations in 1 s cycles. 
R901D	**2 s clock pulse relay**	Repeats on/off operations in 2 s cycles. 
R901E	**1 min clock pulse relay**	Repeats on/off operations in 1 min cycles. 
R901F	**Not used**	-
R9020	**RUN mode flag**	Turns off while the mode selector is set to PROG. Turns on while the mode selector is set to RUN.
R9021 to R9025	**Not used**	-
R9026 (*Note)	**Message flag**	Turns on while the F149 (MSG) instruction is executed.
R9027 (*Note)	**Remote mode flag**	Turns on while the mode selector is set to REMOTE.
R9028	**Not used**	-

Note) Used by the system.

FP0

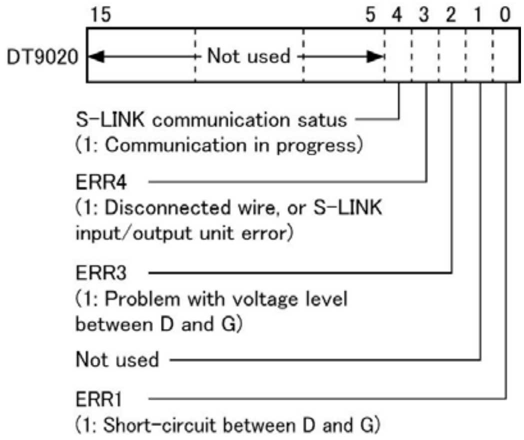
Address	Name		Description
R9029 (*Note)	Forcing flag		Turns on during forced on/off operation for input/output relay timer/counter contacts.
R902A (*Note)	External interrupt enable flag		Turns on while the external interrupt trigger is enabled by the ICTL instruction.
R902B (*Note)	Interrupt error flag		Turns on when an interrupt error occurs.
R902C to R902F	Not used		-
R9030, R9031	Not used		-
R9032	RS232C port mode flag		When “General-use port” is selected, “K2” goes on.
R9033	Printout instruction flag		Turns on while a F147 (PR) instruction is executed. Turns off when a F147 (PR) instruction is not executed.
R9034	Rewrite during RUN flag		This is a special internal relay that goes on for only the first scan following the completion of rewriting in the RUN mode. (CPU Ver. 2.1 or later available)
R9035	S-LINK I/O communi-cation error flag		Turns on when the S-LINK error (ERR 1, 3 or 4) occurs using S-LINK system.
R9036	S-LINK communica-tion status flag		Turns on when communication is taking place with an S-LINK input/Ooutput unit.
R9037	RS232C communica-tion error flag		Turns on when the serial data communication error occurs.
R9038	RS232C reception completed flag		Turns on when a terminator is received during the serial data communicating.
R9039	RS232C transmission completed flag		Turns on while data is not send during the serial data communicating. Turns off while data is being sent during the serial data communicating.
R903A	High-speed counter control flag	ch0	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903B	High-speed counter control flag	ch1	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903C	High-speed counter control flag	ch2	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903D	High-speed counter control flag	ch3	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903E, R903F	Not used		-

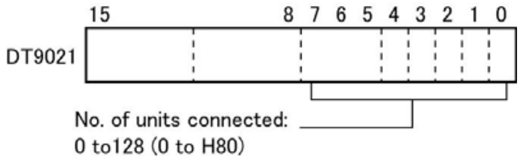
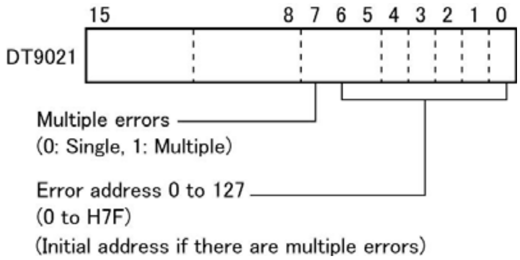
Note) Used by the system.

5.1.3 Table of Special Data Registers for FP0

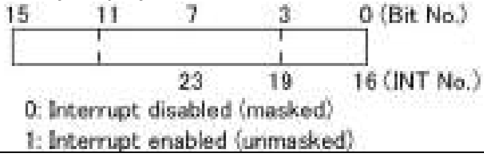
The special data registers are one word (16-bit) memory areas which store specific information. With the exception of registers for which "Writing is possible" is indicated in the "Description" column, these registers cannot be written to.

Address		Name	Descriptions
FP0 T32	FP0 C10, C14, C16, C32, SL1		
DT90000	**DT9000**	**Self-diagnostic error code**	The self-diagnostic error code is stored here when a self-diagnostic error occurs. Monitor the error code using decimal display.
DT90010	**DT9010**	**I/O verify error unit**	The position of the I/O for which an error occurred is stored in bits 0 to 3.
DT90014	**DT9014**	**Auxiliary register for operation**	One shift-out hexadecimal digit is stored in bit positions 0 to 3 when F105 (BSR) or F106 (BSL) instruction is executed.
DT90015	**DT9015**	**Auxiliary register for operation**	The divided remainder (16-bit) is stored in DT9015/DT90015 when F32(%) or F52(B%) instruction is executed. The divided remainder (32-bit) is stored in DT9015 and DT9016/DT90015 and DT90016 when F33(D%) or F53(DB%) instruction is executed.
DT90016	**DT9016**		
DT90017	**DT9017**	**Operation error address (hold)**	After commencing operation, the address where the first operation error occurred is stored. Monitor the address using decimal display.
DT90018	**DT9018**	**Operation error address (non-hold)**	The address where an operation error occurred is stored. Each time an error occurs, the new address overwrites the previous address. At the beginning of scan, the address is 0. Monitor the address using decimal display.
DT90019	**DT9019**	**2.5 ms ring counter**	The data stored here is increased by one every 2.5 ms. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 2.5 ms = Elapsed time between the two points.

Address		Name	Descriptions
FP0 T32	**FP0 C10, C14, C16, C32, SL1**		
-	**DT9020 (Available type: SL1)**	**S-LINK status flag/error flag**	 Notes - ERR1 and ERR3 occur even if the power supply on the S-LINK side is interrupted, but are canceled when the power supply is turned on again. - ERR4 is held. To cancel it, repair the disconnected wire in the S-LINK syste, or whatever iscausing the problem, and then either turn the power to the FP0 on again, press the SET switch to reset it, or turn the power supply on again on the S-LINK unit side.

Address		Name	Descriptions
FP0 T32	**FP0 C10, C14, C16, C32, SL1**		
-	**DT9021 (Available type: SL1)**	**No. of units connected to S-LINK/error address**	(When normal)  Note - When the SET switch is pressed, the number of input/output units connected to the S-LINK system is set. (If the same address has been specified for multiple units, the units are counted as a single unit. This is invalid, however, if an ERR4 error is in progress.) (If ERR4 occurs) 
DT90022	**DT9022**	**Scan time (current value) (*Note)**	The current scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 K50 indicates 5 ms.

Scan time display is only possible in RUN mode, and shows the operation cycle time. The maximum and minimum values are cleared when each the mode is switched between RUN mode and PROG. mode.

Address		Name	Descriptions
FP0 T32	FP0 C10, C14, C16, C32, SL1		
DT90023	**DT9023**	**Scan time (minimum value) (*Note1)**	The minimum scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.
DT90024	**DT9024**	**Scan time (maximum value) (*Note 1)**	The maximum scan time is stored here. The scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K125 indicates 12.5 ms.
DT90025 (*Note2)	**DT9025 (*Note2)**	**Mask condition monitoring register for interrupts (INT 0 to 5)**	The mask conditions of interrupts using ICTL instruction can be monitored here. Monitor using binary display. 
DT90026	**DT9026**	**Not used**	-
DT90027 (*Note2)	**DT9027 (*Note2)**	**Periodical interrupt interval (INT24)**	The value set by the ICTL instruction is stored. K0: periodical interrupt is not used. K1 to K3000: 10ms to 30s
DT90028	**DT9028**	**Not used**	-
DT90029	**DT9029**	**Not used**	-
DT90030 (*Note2)	**DT9030 (*Note2)**	**Character storage by F149 MSG instruction**	The contents of the specified message are stored in these special data registers when F149 (MSG) instruction is executed.
DT90031 (*Note2)	**DT9031 (*Note2)**		
DT90032 (*Note2)	**DT9032 (*Note2)**		
DT90033 (*Note2)	**DT9033 (*Note2)**		
DT90034 (*Note2)	**DT9034 (*Note2)**		
DT90035 (*Note2)	**DT9035 (*Note2)**		
DT90036	**DT9036**	**Not used**	-
DT90037	**DT9037**	**Work 1 for F96 (SRC) instruction**	The number of data that match the searched data is stored here when F96 (SRC) instruction is executed.

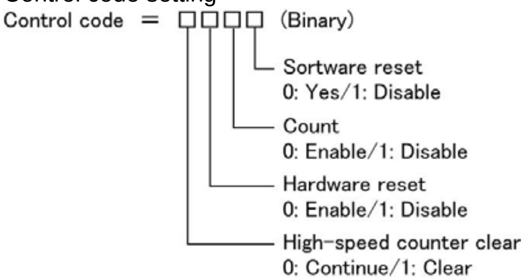
Note1) Scan time display is only possible in RUN mode and shows the operation cycle time. The maximum and minimum values are cleared when each mode is switched between RUN mode and PROG. mode.

Note2) Used by the system.

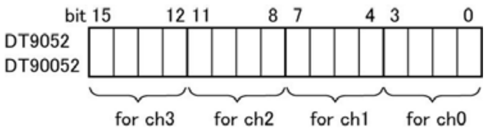
Address		Name	Descriptions
FP0 T32	**FP0 C10, C14, C16, C32, SL1**		
DT90038	**DT9038**	**Work 2 for F96 (SRC) instruction**	The position of the first matching data, counting from the starting 16-bit area, is stored here when an F96 (SRC) instruction is executed.
DT90039 to DT90043	**DT9039 to DT9043**	**Not used**	-
DT90044	**DT9044**	**High-speed counter elapsed value for ch0 (*Note1)**	The elapsed value (24-bit data) for the high-speed counter is stored here. Each time the ED instruction is executed, the elapsed value for the high-speed counter is automatically transferred to the special registers DT9044 and DT9045/DT90044 and DT90045. The value can be written by executing F1 (DMV) instruction.
DT90045	**DT9045**		
DT90046	**DT9046**	**High-speed counter target value for ch0 (*Note1)**	The target value (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction F166 to F170 is executed. These preset values can only be read, and cannot be written.
DT90047	**DT9047**		
DT90048	**DT9048**	**High-speed counter elapsed value area for ch1 (*Note1)**	The elapsed value (24-bit data) for the high-speed counter is stored here. Each time the ED instruction is executed, the elapsed value for the high-speed counter is automatically transferred to the special registers DT9048 and DT9049/DT90048 and DT90049. The value can be written by executing F1 (DMV) instruction.
DT90049	**DT9049**		
DT90050	**DT9050**	**High-speed counter target value area for ch1 (*Note1)**	The target value (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction F166 to F170 is executed. These preset values can only be read, and cannot be written.
DT90051	**DT9051**		

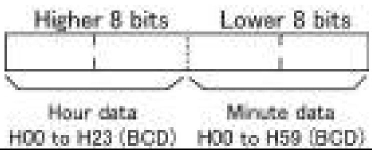
Note1) In the FP0 compatibility mode of FP0R, it is 32-bit data.

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th></tr>
<tr><th>FP0 T32</th><th>FP0 C10, C14, C16, C32, SL1</th></tr>
<tr><td>DT90052</td><td>DT9052</td><td>High-speed counter control flag</td><td>A value can be written with F0 (MV) instruction to reset the high-speed counter, disable counting, stop high-speed counter instruction (F168), and clear the high-speed counter.

Control code setting


Software is not reset: H0 (0000)
Perform software reset: H1 (0001)
Disable count: H2 (0010)
Disable hardware reset: H4 (0100)
Stop pulse output (clear instruction): H8 (1000)
Perform software reset and stop pulse output: H9 (1001)

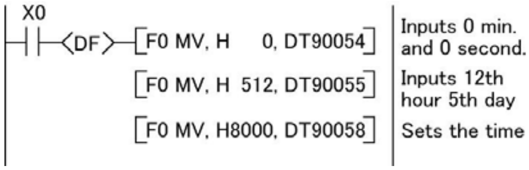
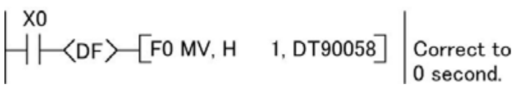
The 16 bits of DT9052/DT90052 are allocated in groups of four to high-speed channels 0 to 3 as shown below.


A hardware reset disable is only effective when using the reset input (X2 and X5). In all other cases it is ignored.
When using pulse output, a hardware reset input is equivalent to an home point proximate input.</td></tr>
<tr><td>DT90053</td><td>-</td><td>Real-Time Clock (Clock/Calendar) monitor (hour/minute)</td><td>Hour and minute data of the Real-Time Clock (Clock/Calendar) are stored here.
This data is read-only data; it cannot be overwritten.

</td></tr>
</table>

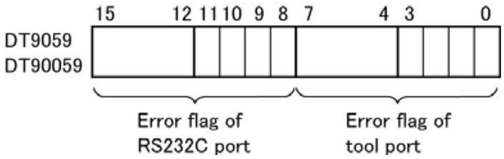
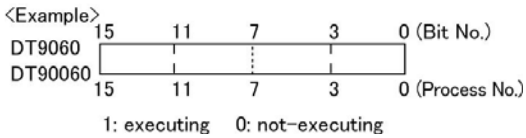
<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th></tr>
<tr><th>FP0 T32</th><th>FP0 C10, C14, C16, C32, SL1</th></tr>
<tr><td>DT90054</td><td>-</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (minute/second)</td><td rowspan="4">The year, month, day, hour, minute, second, and day-of-the-week data for the Real-Time Clock (Clock/Calendar) is stored. The built-in Real-Time Clock(Clock/Calendar) will operate correctly through the year 2099 and supports leap years. The Real-Time Clock (Clock/Calendar) can be set (the time set) by writing a value using a programming tool software or a program that uses the F0 (MV) instruction.

Higher 8 bits | Lower 8 bits

<table>
<tr><td>DT90054</td><td>Minute data
H00 to H59 (BCD)</td><td>Second data
H00 to H59 (BCD)</td></tr>
<tr><td>DT90055</td><td>Day data
H01 to H31 (BCD)</td><td>Hour data
H00 to H23 (BCD)</td></tr>
<tr><td>DT90056</td><td>Year data
H00 to H99 (BCD)</td><td>Month data
H01 to H12 (BCD)</td></tr>
<tr><td>DT90057</td><td>—</td><td>Day-of-the-week data
H00 to H06 (BCD)</td></tr>
</table>As a day of the week is not automatially set on FPWIN GR, fix what day is set to 00, and set each value for 00 to 06.</td></tr>
<tr><td>DT90055</td><td>-</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (day/hour)</td></tr>
<tr><td>DT90056</td><td>-</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (year/month)</td></tr>
<tr><td>DT90057</td><td>-</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (day-of-the-week)</td></tr>
</table>

Address		Name	Descriptions
FP0 T32	FP0 C10, C14, C16, C32, SL1		
DT90058	**-**	**Real-Time Clock (Clock/Calendar) time setting and 30 seconds correction**	The Real-Time Clock(Clock/Calendar) is adjusted as follows. When setting the Real-Time Clock (Clock/Calendar) by program By setting the highest bit of DT90058 to 1, the time becomes that written to DT90054 to DT90057 by F0 (MV) instruction. After the time is set, DT90058 is cleared to 0. (Cannot be performed with any instruction other than F0 (MV) instruction.) **Example:** Set the time to 12:00:00 on the 5th day when the X0 turns on.  If you changed the values of DT90054 to DT90057 with the data monitor functions of programming tool software, the time will be set when the new values are written. Therefore, it is unnecessary to write to DT90058. **When the correcting times less than 30 seconds** By setting the lowest bit of DT90058 to 1, the value will be moved up or down and become exactly 0 seconds. After the correction is completed, DT90058 is cleared to 0. **Example:** Correct to 0 seconds with X0 turns on  At the time of correction, if between 0 and 29 seconds, it will be moved down, and if the between 30 and 59 seconds, it will be moved up. In the example above, if the time was 5 minutes 29 seconds, it will become 5 minutes 0 second; and, if the time was 5 minutes 35 seconds, it will become 6 minutes 0 second.

Note) After discharging the battery (including when the power is turned on for the first time), the values of DT90053 to DT90058 change at random. Once the time and date have been set, these values will function normally.

Address		Name		Descriptions
FP0 T32	FP0 C10, C14, C16, C32, SL1			
DT90059	**DT9059**	**Serial communication error code**		 - Tool port bit 0=1: Over run error bit 1=1: Framing error bit 2=1: Parity error - RS232C port bit 8=1: Over run error bit 9=1: Framing error bit 10=1: Parity error
DT90060	**DT9060**	**Step ladder process**	**Process number: 0 to 15**	Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on"1". Monitor using binary display.  A programming tool software can be used to write data.
DT90061	**DT9061**		**Process number: 16 to 31**	
DT90062	**DT9062**		**Process number: 32 to 47**	
DT90063	**DT9063**		**Process number: 48 to 63**	
DT90064	**DT9064**		**Process number: 64 to 79**	
DT90065	**DT9065**		**Process number: 80 to 95**	
DT90066	**DT9066**		**Process number: 96 to 111**	
DT90067	**DT9067**		**Process number: 112 to 127**	

Address		Name	Descriptions
FP0 T32	FP0 C10, C14, C16, C32, SL1		
DT90104	**DT9104**	**High-speed counter elapsed value area for ch2 (*Note1)**	The elapsed value (24-bit data) for the high-speed counter is stored here. Each time the ED instruction is executed, the elapsed value for the high-speed counter is automatically transferred to the special registers DT9104 and DT9015/DT90104 and DT90105. The value can be written by executing a DMV (F1) instruciton.
DT90105	**DT9105**		
DT90106	**DT9106**	**High-speed counter target value area for ch2 (*Note1)**	The target value (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction F166 to F170 is executed. These preset values can only be read, and cannot be written.
DT90107	**DT9107**		
DT90108	**DT9108**	**High-speed counter elapsed value area for ch3 (*Note1)**	The elapsed value (24-bit data) of the high-speed counter is stored here. Each time the ED instruction is executed, the elapsed value for the high-speed counter is automatically transferred to the special registers DT9108 and DT9109/DT90108 and DT90109. The value can be written by executing a DMV (F1) instruction.
DT90109	**DT9109**		
DT90110	**DT9110**	**High-speed counter target value area for ch3 (*Note1)**	The target value (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction F166 to F170 is executed. These preset values can only be read, and cannot be written.
DT90111	**DT9111**		

Note1) In the FP0 compatibility mode of FP0R, it is 32-bit data.

5.1.4 Table of System Registers for FP-e

FP-e

	No.	Name	Default value	Descriptions	
Hold/ Non-hold	**5**	Starting number setting for counter	100	0 to 144	
	6	Hold type area starting number setting for timer and counter	140	0 to 144	(See note.)
	7	Hold type area starting number setting for internal relays	61	0 to 63	
	8	Hold type area starting number setting for data registers	1652	0 to 1660	
	14	Hold or non-hold setting for step ladder process	Non-hold	Hold/Non-hold	
Action on error	**20**	Disable or enable setting for duplicated output	Yes FPWIN GR: Disabled	Fixed FPWIN GR: Disabled/Enabled	
	26	Operation setting when an operation error occurs	Stop	Stop/Continuation of operation	
	4	Alarm battery error (Operating setting when battery error occurs)	Disabled	Dis- abled:	When a battery error occurs, a self-diagnostic error is not issued and the ERROR LED does not light.
				Ena- bled:	When a battery error occurs, a self-diagnostic error is issued and the ERROR LED lights.
Time set- ting	**31**	Wait time setting for multi-frame communication	6500.0 ms	10 to 81900 ms	
	34	Constant value settings for scan time	0.0 ms	0: Normal scan 0 to 160 ms: Scans once each specified time interval	

Note) Use models without a Real-Time Clock(Clock/Calendar) function with the default value left as is.
If you change the setting the hold/non-hold operation will be unstalbe.
Settings are valid for models with a Real-Time Clock(Clock/Calendar) time function.

FP-e

	No.	Name	Default value	Descriptions	
High-speed coun-ter	**400**	**High-speed counter operation mode settings (X0 to X2)**	CH0: Do not set input X0 as high-speed counter	CH0	Do not set input X0 as high-speed counter. Two-phase input (X0, X1) Two-phase input (X0, X1), Reset input (X2) Incremental input (X0) Incremental input (X0), Reset input (X2) Decremental input (X0) Decremental input (X0), Reset input (X2) incremental/decremental input (X0, X1) incremental/decremental input (X0, X1), Reset input (X2) Incremental/decremental control input (X0, X1) Incremental/decremental control input (X0, X1), Reset input (X2)
			CH1: Do not set input X1 as high-speed counter	CH1	Do not set input X1 as high-speed counter. Incremental input (X1) Incremental inptu (X1), Reset input (X2) Decremental input (X1) Decremental input (X1), Reset input (X2)
	401	**High-speed counter operation mode settings (X3 to X5)**	CH2: Do not set input X3 as high-speed counter	CH2	Do not set input X3 as high-speed counter. Two-phase input (X3, X4) Two-phase input (X3, X4), Reset input (X5) Incremental input (X3) Incremental input (X3), Reset input (X5) Decremental input (X3) Decremental input (X3), Reset input (X5) Incremental/decremental input (X3, X4) Incremental/decremental input (X3, X4), Reset input (X5) Incremental/decremental control input (X3, X4) Incremental/decremental control input (X3, X4), Reset input (X5)
			HC3: Do not set input X4 as high-speed counter	CH3	Do not set input X4 as high-speed counter. Incremental input (X4) Incremental input (X4), Reset input (X5) Decremental input (X4) Decremental input (X4), Reset input (X5)

FP-e

	No.	Name	Default value	Descriptions
Inter-rupt-input	**402**	**Pulse catch input settings**	Not set	X0 X1 X2 X3 X4 X5 Specify the input contacts used as pulse catch input.
	403	**Interrupt input settings**	Not set	X0 X1 X2 X3 X4 X5 Specify the input contacts used as intrrupt input. X0 X1 X2 X3 X4 X5 Specify the effective interrupt edge. (When set: ON→OFF is valid)

Note1) If the operation mode is set to two-phase, incremental/decremental, or incremental/decremental control, the setting for CH1 is invalid in part 2 of system register 400 and the setting for CH3 is invalid in part 2 of system register 401.

Note2) If reset input settings overlap, the CH1 setting takes precedence in system register 400 and the CH3 setting takes precedence in system register 401.

Note3) The settings for pulse catch and interrupt input can only be specified in system registers 402 and 403.

Note4) If system register 400 to 403 have been set simultaneously for the same input relay, the following precedence order is effective:

1. High-speed counter
2. Pulse catch
3. Interrupt input.

This means, the counter keeps counting even after an interrupt.

FP-e

	No.	Name	Default value	Descriptions
Tem-pera-ture inout	**409**	**Number of temperature input average processing times (Available PLC: model with thermocouple input)**	0	0 to 50 For default valeu “0”, the number of average processing times is 20.
Tool port set-ting	**410**	**Unit No. setting**	1	1 to 99
	411	**Communication format setting**	Disabled	Modem connection: enabled/Disabled
			Data length: 8 bits	Data length: 7 bits/8 bits When connecting a modem, the format will be as follows depending on the data length setting. 8 bits data length: no parity, 1 stop bit 7 bits data length: odd parity, 1 stop bit
	414	**Communication speed (Baud rate) setting**	9600 bps	9600 bps 19200 bps
COM. port set-ting	**412**	**Communication mode setting**	Computer link	Computer link General-purpose serial communication MODBUS RTU (Ver.1.2 and higher)
	413	**Communication format setting**	Data lenght bit: 8 bits Parity check: Odd Stop bit: 1 bit	Enter the settings for the various items. - Data lenght: 7 bits/8 bits - Parity check: none/with odd/with even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator CR/CR+LF/None - Header: STX not exist/STX exist
	414	**Communication speed (Baud rate) setting**	9600 bps	300 bps / 600 bps / 1200 bps / 2400 bps / 4800 bps / 9600 bps / 19200 bps
	415	**Unit no. setting**	1	1 to 99 (In Ver.1.2 and higher, settings can be changed in R mode even with the front operation switch.)
	416	**Selection of modem connection**	Disabled	Enabled/Disabled
	417	**Starting address for received buffer of general (serial data) communication mode**	0	0 to 1659
	418	**Buffer capacity setting for data received of general (serial data) communication mode**	1660	0 to 1660

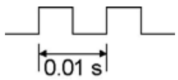
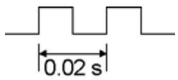
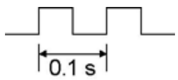
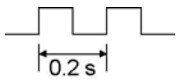
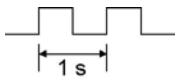
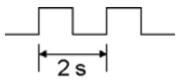
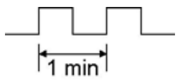
5.1.5 Table of Special Internal Relays for FP-e

The special internal relays turn on and off under special conditions. The on and off states are not output externally. Writing is not possible with a programming tool or an instruction.

FP-e

Relay No.	Name	Description
R9000	**Self-diagnostic error flag**	Turns on when a self-diagnostic error occurs. ⇒ The content of self-diagnostic error is stored in DT90000.
R9001	**Not used**	-
R9002	**Not used**	-
R9003	**Not used**	-
R9004	**Not used**	-
R9005	**Backup battery error flag (non-hold)**	Turns on for an instant when a backup battery error occurs.
R9006	**Backup battery error flag (hold)**	Turns on and keeps the on state when a backup battery error occurs. Once a battery error has been detected, this is held even after recovery has been made. It goes off if the power supply is turned off, or if the system is initialized.
R9007	**Operation error flag (hold)**	Turns on and keeps the on state shen an operation error occurs. ⇒The address where the error occurred is stored in DT9017. (Indicates the first operation error which occurred).
R9008	**Operation error flag (non-hold)**	Turns on for an instant when an operation error occurs. ⇒The address where the operation error occurred is stored in DT9018. The contents change each time a new error occurs.
R9009	**Carry flag**	This is set if an overflow or underflow occurs in the calculation results, and as a result of a shift system instruction being executed.
R900A	**> Flag**	Turns on for an instant when the compared results become larger in the comparison instructions.
R900B	**= Flag**	Turns on for an instant, - when the compared results are equal in the comparison instructions. - when the calculated results become 0 in the arithmetic instructions.
R900C	**< Flag**	Turns on for an instant when the compared results become smaller in the comparison instructions.
R900D	**Auxiliary timer instruction flag**	Turns on when the set time elapses (set value reaches 0) in the timing operation of the F137(STMR)/F183(DSTM) auxiliary timer instruction. The flag turns off when the trigger for auxiliary timer instruction turns off.
R900E	**Tool port communication error**	Turns on when a communication error at Tool port has occurred.
R900F	**Constant scan error flag**	Turns on when the scan time exceeds the time specified in system register 34 during constant scan execution. This goes on if 0 has been set using system register 34.

FP-e

Relay No.	Name	Description
R9010	**Always on relay**	Always on.
R9011	**Always off relay**	Always off.
R9012	**Scan pulse relay**	Turns on and off alternately at each scan.
R9013	**Initial (on type) pulse relay**	Goes on for only the first scan after operation (RUN) has been started, and goes off for the second and subsequent scans.
R9014	**Initial (off type) pulse relay**	Goes off for only the first scan after operation (RUN) has been started, and goes on for the second and subsequent scans.
R9015	**Step ladder initial pulse relay (on type)**	Turns on for only the first scan of a process after the boot at the step ladder control.
R9016	**Not used**	**-**
R9017	**Not used**	-
R9018	**0.01 s clock pulse relay**	Repeats on/off operations in 0.01 s cycles. 
R9019	**0.02 s clock pulse relay**	Repeats on/off operations in 0.02 s cycles. 
R901A	**0.1 s clock pulse relay**	Repeats on/off operations in 0.1 s cycles. 
R901B	**0.2 s clock pulse relay**	Repeats on/off operations in 0.2 s cycles. 
R901C	**1 s clock pulse relay**	Repeats on/off operations in 1 s cycles. 
R901D	**2 s clock pulse relay**	Repeats on/off operations in 2 s cycles. 
R901E	**1 min clock pulse relay**	Repeats on/off operations in 1 min cycles. 
R901F	**Not used**	-

FP-e

Relay No.	Name	Description
R9020	**RUN mode flag**	Turns off while the mode selector is set to PROG. Turns on while the mode selector is set to RUN.
R9021	**Not used**	-
R9022	**Not used**	-
R9023	**Not used**	-
R9024	**Not used**	-
R9025	**Not used**	-
R9026	**Message flag**	Turns on while the F149 (MSG) instruction is executed.
R9027	**Not used**	-
R9028	**Not used**	-
R9029	**Forcing flag**	Turns on during forced on/off operation for input/output relay timer/counter contacts.
R902A	**Interrupt enable flag**	Turns on while the external interrupt trigger is enabled by the ICTL instruction.
R902B	**Interrupt error flag**	Turns on when an interrupt error occurs.
R902C	**Not used**	-
R902D	**Not used**	-
R902E	**Not used**	-
R902F	**Not used**	-

FP-e

Relay No.	Name		Description
R9030	**Not used**		-
R9031	**Not used**		-
R9032	**Not used**		-
R9033	**Print instruction execution flag**		Off: Printing is not executed. On: Execution is in progress.
R9034	**RUN overwrite complete flag**		Goes on for ony the first scan following completion of a rewrite during RUN operation.
R9035	**Not used**		-
R9036	**Not used**		-
R9037	**COM port communication error flag**		- Goes on is a transmission error occurs during data communication.
R9038	**COM port reception done flag during general-purpose serial communication**		- Turns on when the terminator is received during general - purpose serial communication.
R9039	**COM port transmission done flag during general-purpose serial communication**		- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose serial communication.
R903A	**High-speed counter control flag**	**ch0**	Turns on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F168(SPD1) to F170(PWM) are executed.
R903B	**High-speed counter control flag**	**ch1**	Turns on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F168(SPD1) to F170(PWM) are executed.
R903C	**High-speed counter control flag**	**ch2**	Turns on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F168(SPD1) to F170(PWM) are executed.
R903D	**High-speed counter control flag**	**ch3**	Turns on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F168(SPD1) to F170(PWM) are executed.
R903E			-
R903F			-

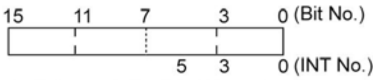
5.1.6 Table of Special Data Registers for FP-e

The special data registers are one word (16-bit) memory areas which store specific information.

FP-e (A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read -ing	Writ- ing
DT9000	**Self-diagnostic error code**	The self-diagnostic error code is stored here when a self-diagnostic error occurs.	A	N/A
DT9001	**FP-e screen display switching**	Switches the FP-escreen to the screen of the mode specified. K0: N mode first screen K1: N mode second screen K2: S mode first screen K3: S mode second screen K4: R mode first screen K5: R mode second screen K6: I mode first screen K7: I mode second screen	A	N/A
DT9002 **DT9003**	**Analog input data**	Ch.0 analog input data (2-word real data)	A	N/A
DT9004 **DT9005**	**Analog input data**	Ch.1 analog input data (2-word real data)	A	N/A
DT9014	**Operation auxiliary register for data shift instruction**	One shift-out hexadecimal digit is stored in bit positions 0 to 3 when the data shift instruction, F105 (BSR) or F106 (BSL) is executed. The value can be read and written by executing the F0 (MV) instruction.	A	A
DT9015 **DT9016**	**Operation auxiliary register for division instruction**	The divided remainder (16-bit) is stored in DT9015 when the division instruction F32(%) or F52(B%) instruction is executed. The divided remainder (32-bit) is stored in DT9015 and DT9016 when the division instruction F33(D%) or F53(DB%) is executed. The value can be read and written by executing the F0(MV) instruction.	A	A
DT9017	**Operation error address (hold type)**	After commencing operation, the address where the first operation error occurred is stored. Monitor the address using decimal display.	A	N/A
DT9018	**Operation error address (non-hold type)**	The address where an operation error occurred is stored. Each time an error occurs, the new address overwrites the previous address. At the beginning of a scan, the address is 0. Monitor the address using decimal display.	A	N/A
DT9019	**2.5 ms ring counter** Note1)	The data stored here is increased by one every 2.5 ms. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 2.5 ms = Elapsed time between the two points.	A	N/A

FP-e (A: Available, N/A: Not available)

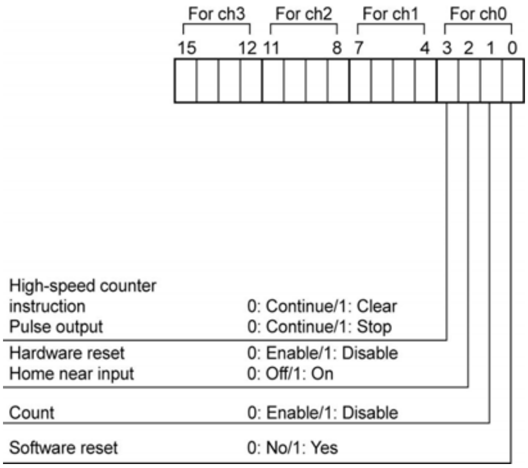
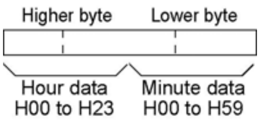
Register No.	Name	Descriptions	Read -ing	Writ -ing
DT9020	**Not used**	-	N/A	N/A
DT9021	**Not used**	-		
DT9022	**Scan time (current value) Note)**	The current scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.	A	N/A
DT9023	**Scan time (minimum value) Note)**	The minimum scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.		
DT9024	**Scan time (maximum value) Note)**	The maximum scan time is stored here. The scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K125 indicates 12.5 ms.		
DT9025	**Mask condition monitoring register for interrupts**	The mask conditions of interrupts using the instruction can be stored here. Monitor using binary display.  0: Interrupt disabled (masked) 1: Interrupt enabled (unmasked)		
DT9026	**Not used**	-	N/A	N/A
DT9027	**Periodical interrupt interval (INT24)**	The value set by the ICTL instruction is stored. K0: periodical interrupt is not used. K1 to K3000: 0.5ms to 1.5s or 10ms to 30s	A	N/A
DT9028	**Not used**	-	N/A	N/A
DT9029	**Not used**	-		
DT9030	**Message 0**	The contents of the specified message (Data lenght) are stored in these special data registers when F149 (MSG) instruction is executed.	A	N/A
DT9031	**Message 1**			
DT9032	**Message 2**			
DT9033	**Message 3**			
DT9034	**Message 4**			
DT9035	**Message 5**			

Note) Scan time display is only possible in RUN mode and shows the operation cycle time. (in PROG mode, the scan time for the operation is not displayed.) The maximum and minimum values are cleared each time the mode is switched from RUN to PROG.

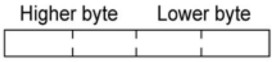
FP-e (A: Available, N/A: Not available)

Register No.	Name		Descriptions	Read -ing	Writ -ing
DT9036	**Not used**		-	N/A	N/A
DT9037	**Operation auxiliary register for search instruction F96(SRC)**		The number of data that match the searched data is stored here when F96 (SRC) insturction is executed.	A	A
DT9038	**Operation auxiliary register for search instruction F96(SRC)**		The position of the first matching data is stored here when an F96 (SRC) instruction is executed.		
DT9039	**Not used**		-	N/A	N/A
DT9040	**Temperature input ch.0**		The value of the temperature input before average processing is stored.	A	N/A
DT9041	**Temperature input ch.1**				
DT9042	**Not used**		-	N/A	N/A
DT9043	**Used by the system**		Used by the system (Battery).	A	N/A
DT9044	**High-speed counter elapsed value**	**For CH0**	The elapsed value (24-bit data) of the high-speed counter is stored here. The value can be read and written by executing F1 (DMV) instruction.	A	A
DT9045					
DT9046	**High-speed counter target value**	**For CH0**	The targe value (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions to be used when the high-speed counter related instruction is executed. The value can be read by executing F1 (DMV) instruction.	A	N/A
DT9047					
DT9048	**High-speed counter elapsed value area**	**For CH1**	The elapsed value (24-bit data) of the high-speed counter is stored here. The value can be read and written by executing F1 (DMV) instruction.	A	A
DT9049					

FP-e (A: Available, N/A: Not available)

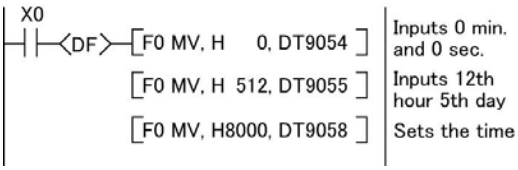
Register No.	Name		Descriptions	Read -ing	Writ -ing
DT9050 **DT9051**	**High-speed counter target value area**	**For CH1**	The target value (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions to be used when the high-speed counter related instruction is executed. The value can be read by executing F1 (DMV) instruction.	A	N/A
DT9052	**High-speed counter and pulse output control flag**		A value can be written with F0 (MV) instruction to reset the high-speed counter, disable counting, continue or clear high-speed counter instruction. Control code setting 	N/A	A
DT9053	**Real-Time Clock (Clock/Calendar) monitor (hour/minute)**		Hour and minute data of the Real-Time Clock (Clock/Calendar) are stored here. This data is read-only data. It cannot be overwritten. 	A	N/A

FP-e (A: Available, N/A: Not available)

<table>
<tr><th>Register No.</th><th>Name</th><th>Descriptions</th><th>Read -ing</th><th>Writ -ing</th></tr>
<tr><td>DT9054</td><td>Real-Time Clock (Clock/Calendar) setting (minute/second)</td><td rowspan="4">The year, month, day, hour, minute, second and day-of-the-week data for the Real-Time Clock (Clock/Calendar) is stored. The built-in Real-Time Clock(Clock/Calendar) will operate correctly through the year 2099 and supports leap years. The Real-Time Clock (Clock/Calendar) can be set by writing a value using a programming tool software or a program that uses the F0 (MV) instruction.(see example for DT90058)


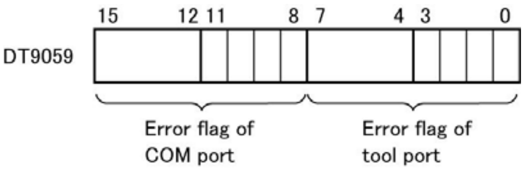
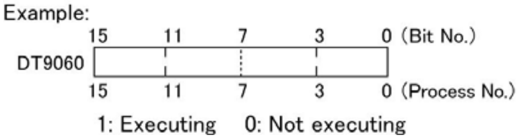
<table>
<tr><td>DT9054</td><td>Minute data (H00 to H59)</td><td>Second data (H00 to H59)</td></tr>
<tr><td>DT9055</td><td>Day data (H01 to H31)</td><td>Hour data (H00 to H23)</td></tr>
<tr><td>DT9056</td><td>Year data (H00 to H99)</td><td>Month data (H01 to H12)</td></tr>
<tr><td>DT9057</td><td>—</td><td>Day-of-the-week (H00 to H06)</td></tr>
</table></td><td rowspan="4">A</td><td rowspan="4">A</td></tr>
<tr><td>DT9055</td><td>Real-Time Clock (Clock/Calendar) setting (day/hour)</td></tr>
<tr><td>DT9056</td><td>Real-Time Clock (Clock/Calendar) setting (year/month)</td></tr>
<tr><td>DT9057</td><td>Real-Time Clock (Clock/Calendar) setting (day-of-the-week)</td></tr>
<tr><td>DT9058</td><td>Real-Time Clock (Clock/Calendar) time setting</td><td>By setting the highest bit of DT9058 to 1, the time becomes that written to DT9054 to DT9057 by the F0 (MV) instruction. After the time is set, DT9058 is cleared to 0. (Cannot be performed with any instruction other than the F0 (MV) instruction.)

<Example>
Set the time to 12:00:00 on the 5th day when X0 turns on.

FPWIN GR:


If you changed the values of DT9054 to DT9057 with the programming tool software, the time will be set when the new values are written. Therefore, it is unnecessary to write to DT9058.</td><td>A</td><td>A</td></tr>
</table>

FP-e (A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read -ing	Writ -ing
DT9059	**Serial communication error code**	Error code is sotred here when a communication error occurs.  ・Tool port bit 0 = 1: Over run error bit 1 = 1: Framing error bit 2 = 1: Parity error ・COM port bit 8 = 1: Over run error bit 9 = 1: Framing error bit 10 = 1: Parity error	A	N/A
DT9060	**Step ladder process (0 to 15)**	Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on. Monitor using binary display. Example:  A programming tool software can be used to write data.	A	A
DT9061	**Step ladder process (16 to 31)**			
DT9062	**Step ladder process (32 to 47)**			
DT9063	**Step ladder process (48 to 63)**			
DT9064	**Step ladder process (64 to 79)**			
DT9065	**Step ladder process (80 to 95)**			
DT9066	**Step ladder process (96 to 111)**			
DT9067	**Step ladder process (112 to 127)**			

FP-e (A: Available, N/A: Not available)

Register No.	Name		Descriptions	Read -ing	Writ -ing
DT9104	**High-speed counter elapsed value**	**For ch2**	The elapsed value (24-bit data) for the high-speed conter is stored here. The value can be read and written by executing the F1 (DMV) instruciton.	A	A
DT9105					
DT9106	**High-speed counter target value**	**For ch2**	The target valeu (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction is executed. The value can be read by executing the F1 (DMV) instruction.	A	N/A
DT9107					
DT9108	**High-speed counter elapsed value**	**For ch3**	The elapsed value (24-bit data) for the high-speed counter is stored here. The value can be read and written by executing the F1 (DMV) instruction.	A	A
DT9109					
DT9110	**High-speed counter target value**	**For ch3**	The target value (24-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction is executed. The value can be read by executing the F1 (DMV) instruction.	A	N/A
DT9111					

5.1.7 Table of System Registers for FP0R

	No.	Name	Default value	Descriptions
Hold/ Non-hold 1	5	Starting number setting for counter	1008	0 to 1024
	6	Hold type area starting number setting for timer and counter (T32/F32)	1008	0 to 1024
	7	Hold type area starting number setting for internal relays (T32/F32)	248	0 to 256
	8	Hold type area starting number setting for data registers (T32/F32)	0	0 to 32765
	14	Hold or non-hold setting for step ladder process (T32/F32)	Non-hold	Hold/Non-hold
	4	Previous value is held for a leading edge detection instruction (DF instrucion) with MC Note)	Hold	Hold/ Non-hold
Hold/ Non-hold 2	10	Hold type area starting word number for PC(PLC) link relays (for PC(PLC) link 0) (T32/F32)	0	0 to 64
	11	Hold type area starting word number for PC(PLC) link relays (for PC(PLC) link 1) (T32/F32)	64	64 to 128
	12	Hold type area starting number for PC(PLC) link registers (for PC(PLC) link 0) (T32/F32)	0	0 to 128
	13	Hold type area starting number for PC(PLC) link registers (for PC(PLC) link 1) (T32/F32)	128	128 to 256
Action on error	20	Disable or enable setting for duplicated output	Disabled	Disabled/Enabled
	23	Operation setting when an I/O verification error occurs	Stop	Stop/Continuation of operation
	26	Operation setting when an operation error occurs	Stop	Stop/Continuation of operation
Time set-ting	31	Wait time setting for multi-frame communication	6500.0 ms	10 to 81900 ms
	32	Communication timeout setting for SEND/RECV, RMRD/RMWT commands	10000.0 ms	10 to 81900 ms
	34	Constant value settings for scan time	Normal scan	0: Normal scan 0 to 600 ms: Scans once each specified time interval

	No.	Name	Default value	Descriptions
PC (PLC) link 0 set-ting	**40**	Range of link relays used for PC(PLC) link	0	0 to 64 words
	41	Range of link data registers used for PC(PLC) link	0	0 to 128 words
	42	Starting word number for link relay transmission	0	0 to 63
	43	Link relay transmission size	0	0 to 64 words
	44	Starting number for link data register tranmission	0	0 to 127
	45	Link data register transmission size	0	0 to 127 words
	46	PC(PLC) link switch flag	Normal	Normal/reverse
	47	Maximum unit number setting for MEWNET-W0 PC(PLC) link	16	1 to 16
PC (PLC) link 1 set-ting)	**50**	Range of link relays used for PC(PLC) link	0	0 to 64 words
	51	Range of link data registers used for PC(PLC) link	0	0 to 128 words
	52	Starting word number for link relay transmission	64	64 to 127
	53	Link relay transmission size	0	0 to 64 words
	54	Starting number for link data register tranmission	128	128 to 255
	55	Link data register transmission size	0	0 to 127 words
	57	Maximum unit number setting for MEWNET-W0 PC(PLC) link	16	1 to 16

		No.	Name	Default value	Descriptions	
Controller input settings 1	High-speed counter	400	High-speed counter operation mode settings (X0 to X2)	CH0: Do not set input X0 as high-speed counter	CH0	Do not set input X0 as high-speed counter. Two-phase input (X0, X1) Two-phase input (X0, X1), Reset input (X2) Incremental input (X0) Incremental input (X0), Reset input (X2) Decremental input (X0) Decremental input (X0), Reset input (X2) Individual input (X0, X1) Individual input (X0, X1), Reset input (X2) Incremental/decremental control input (X0, X1) Incremental/decremental control input (X0, X1), Reset input (X2)
				CH1: Do not set input X1 as high-speed counter	CH1	Do not set input X1 as high-speed counter. Incremental input (X1) Incremental inptu (X1), Reset input (X2) Decremental input (X1) Decremental input (X1), Reset input (X2)
		400	High-speed counter operation mode settings (X3 to X5)	CH2: Do not set input X3 as high-speed counter	CH2	Do not set input X3 as high-speed counter. Two-phase input (X3, X4) Two-phase input (X3, X4), Reset input (X5) Incremental input (X3) Incremental input (X3), Reset input (X5) Decremental input (X5) Decremental input (X5), Reset input (X5) Individual input (X3, X4) Individual input (X3, X4), Reset input (X5) Incremental/decremental control (X3, X4) Incremental/decremental control (X3, X4), Reset input (X5)
				CH3: Does not set input X4 as high-speed counter	CH3	Does not set input X4 as high-speed counter. Incremental input (X4) Incremental input (X4), Reset input (X5) Decremental input (X4) Decremental input (X4), Reset input (X5)
Controller input settings 2	High-speed counter	401	High-speed counter/ pulse output settings (X6 to X7)	CH4: Do not set input X6 as high-speed counter	CH4	Do not set input X6 as high-speed counter. Incremental input (X6) Decremental input (X6) Two-phase input (X6, X7) Individual input (X6, X7) Incremental/decremental control input (X6, X7)
				CH5: Do not set input X7 as high-speed counter	CH5	Do not set input X7 as high-speed counter. Incremental input (X7) Decremental input (X7)

Note1) If the operation mode is set to Two-phase, incremental/decremental, or incremental/decremental control, the setting for CH1 or CH3 is invalid in system register 400 and the setting for CH5 is invalid in system register 401.

Note2) If reset input settings overlap, the CH1 setting takes precedence in system register 400 and the CH3 setting takes precedence in system register 401.

Note3) If system register 400 to 403 have been set simultaneously for the same input relay, the follwing precedence order is effective: [High-speed counter]→[Pulse catch]→[Interrupt input].

<Example>

When the high-speed counter is being used in the addition input mode, even if input X0 is specified as an interrupt input or as pulse catch input, those settings are invalid, and X0 functions as counter input for the high-speed counter.

		No.	Name	Default value	Descriptions
Transistor type C16 or over	Controller output settings 2 (PLS/PWM)	402	**Pulse/ PWM output settings (Y0 to Y7)**	CH0: Normal output	Normal output (Y0, Y1) Pulse output (Y0, Y1) Pulse output (Y0, Y1)/Home input X4 Pulse output (Y0, Y1)/Home input X4/Position control starting input X0 PWM output (Y0), Normal output (Y1)
				CH1: Normal output	Normal output (Y2, Y3) Pulse output (Y2, Y3) Pulse output (Y2, Y3)/Home input X5 Pulse output (Y3, Y4)/Home input X5/Position control starting input X1 PWM output (Y2), Normal output (Y3)
				CH2: Normal output	Normal output (Y4, Y5) Pulse output (Y4, Y5) Pulse output (Y4, Y5)/Home input X6 Pulse output (Y4, Y5)/Home input X6/Position control starting input X2 PWM output (Y4), Normal output (Y5)
				CH3: Normal output	Normal output (Y6, Y7) Pulse output (Y6, Y7) Pulse output (Y6, Y7)/Home input X7 Pulse output (Y6, Y7)/Home input X7/Position control starting input X3 PWM output (Y6), Normal output (Y7)
Inter-rupt/ Pulse catch settings		403	**Pulse catch input settings**	Not set	X0 X1 X2 X3 X4 X5 X6 X7 Controller input The pressed contact is set for the pulse catch.
		404	**Interrupt input settings**	Not set	X0 X1 X2 X3 X4 X5 X6 X7 Controller input The pressed contact is set for the interrupt input.
Inter-rupt edge settings		405	**Interrupt edge setting for controller input**	Leading edge	X0 X1 X2 X3 X4 X5 X6 X7 Leading edge X0 X1 X2 X3 X4 X5 X6 X7 Trailing edge The pressed contact is up and set to trailing edge.

Note1) When using the pulse output/PWM output, the controller output settings must be specified.
The output that has been set to the pulse output/PWM output cannot be used as the normal output.

Note2) X4 to X7 can be used as the home input of the pulse output CH0 to CH3.
When using the home return function of the pulse output, always set the home input. In that case, X4 to X7 cannot be set as the high-speed counter.

Note3) C16 type:
- For performing the home return for the pulse output CH0 with deviation counter clear, the above Y6 should be set to the normal output to use Y6 for the deviation counter clear signal.
- For performing the home return for the pulse output CH1 with deviation counter clear, the above Y7 should be set to the normal output to use Y7 for the deviation counter clear signal.
- The home return cannot be performed for the pulse output CH2 with deviation counter clear.

Note4) C32/T32/F32 type:
When performing theo home return with deviation counter clear, the deviation counter clear signals corresponding to each CH are used fixedly as follows; CH0=Y8, CH1=Y9, CH2=YA, CH3=YB
For performing the home return for each type,
it is necessary to specify the home input corresponding to each channel to be used for the home return in the system register 401.
Home input corresponding to each channel: CH0=4, CH1=X5, CH2=X6, CH3=X7
For performing the JOG positioning for each type,
it is necessary to specify the position control starting input signal corresponding to each channel to be used for the JOG positioning.

Note3) The settings for pulse catch and interrupt input can only be specified in system registers 403 to 405.

	No.	Name	Default value	Descriptions
Tool port set-ting	**410**	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose communications Note2)
		Selection of modem connection	Disabled	Enabled/Disabled
	413	**Communication format setting**	Data lenght bit: 8 bits Parity check: “with odd” Stop bit: 1 bit	Enter the settings for the various items. - Data lenght bit: 7 bits/8 bits - Parity check: none/with odd/with even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator CR/CR+LF/None - Header: STX not exist/STX exist
	415	**Communication speed (Baud rate) setting**	9600 bps	2400 bps / 4800 bps / 9600 bps / 19200 bps / 38400 bps / 57600 bps / 115200 bps
	420	**Starting address for received buffer of general (serial data) communication mode**	4096	0 to 32764
	421	**Buffer capacity setting for data received of general (serial data) communication mode**	2048	0 to 2048
COM port set-ting	**410**	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose serial communication PC(PLC) link MODBUS RTU
		Selection of modem connection	Disabled	Enabled/Disabled
	413	**Communication format setting**	Data lenght bit: 8 bits Parity check: Odd Stop bit: 1 bit	Enter the settings for the various items. - Data lenght bit: 7 bits/8 bits - Parity check: none/with odd/with even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator CR/CR+LF/None/ETX - Header: STX not exist/STX exist
	415	**Communication speed (Baud rate) setting**	9600 bps	2400 bps / 4800 bps / 9600 bps / 19200 bps / 38400 bps / 57600 bps / 115200 bps
	416	**Starting address for received buffer of general (serial data) communication mode**	0	0 to 32764
	417	**Buffer capacity setting for data received of general (serial data) communication mode**	2048	0 to 2048

Note1) The communication format in a PLC link is fixed at the following settings:
Data length is 8 bits, odd parity, stop bit is 1.
The communication speed (baud rate) is fixed at 115200 bps.

Note2) The general-purpose communication with the tool port is available only in RUN mode. In PROG mode, the computer link mode must be used regardless of settings.

Item	Address	Name	Default value	Description
Controller input time constant settings	**430**	**Controller input time constant setting 1 X0 to X3**	1 ms	None 0.1 ms 0.5 ms 1 ms 2 ms 4 ms 8 ms 16 ms 32 ms 64 ms
	431	**Controller input time constant setting 1 X4 to X7**		
	432	**Controller input time constant setting 2 X8 to XB (C32/T32/F32)**		
	433	**Controller input time constant setting 2 XC to XF (C32/T32/F32)**		

Note) X6 and X7 is invalid for C10.

5.1.8 Table of Special Internal Relays for FP0R

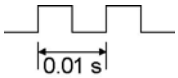
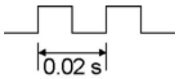
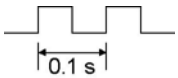
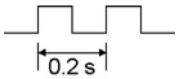
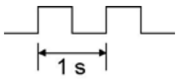
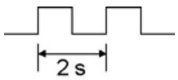
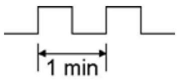
The special internal relays turn on and off under special conditions. The on and off states are not output externally. Writing is not possible with a programming tool or an instruction.

WR900 FP0R

Relay No.	Name	Description
R9000	**Self-diagnostic error flag**	Turns on when a self-diagnostic error occurs. ⇒ The content of self-diagnostic error is stored in DT90000.
R9001	**Not used**	
R9002	**Not used**	
R9003	**Not used**	
R9004	**I/O verification error flag**	Turns on when an I/O verification error occurs.
R9005	**Not used**	
R9006	**Not used**	
R9007	**Operation error flag (hold)**	Turns on and keeps the on state shen an operation error occurs. ⇒The address where the error occurred is stored in DT90017. (indicates the first operation error which occurred).
R9008	**Operation error flag (non-hold)**	Turns on for an instant when an operation error occurs. ⇒The address where the operation error occurred is stored in DT90018. The contents change each time a new error occurs.
R9009	**Carry flag**	This is set if an overflow or underflow occurs in the calculation results, and as a result of a shift system instruction being executed.
R900A	**> Flag**	Turns on for an instant when the compared results become larger in the comparison instructions (F60 to F63).
R900B	**= Flag**	Turns on for an instant, - when the compared results are equal in the comparison instructions (F60 to F63). - when the calculated results become 0 in the arithmetic instructions.
R900C	**< Flag**	Turns on for an instant when the compared results become smaller in the comparison instructions (F60 to F63).
R900D	**Auxiliary timer instruction flag**	Turns on when the set time elapses (set value reaches 0) in the timing operation of the F137(STMR)/F183(DSTM) auxiliary timer instruction. The flag turns off when the trigger for auxiliary timer instruction turns off.
R900E	**Tool port communication error**	Turns on when communication error at tool port is occurred.
R900F	**Constant scan error flag**	Turns on when scan time exceeds the time specified in system register 34 during constant scan execution. This goes on if 0 has been set using system register 34.

A: Available, N/A: Not available

WR901 FP0R

Relay No.	Name	Description
R9010	**Always on relay**	Always on.
R9011	**Always off relay**	Always off.
R9012	**Scan pulse relay**	Turns on and off alternately at each scan.
R9013	**Initial (on type) pulse relay**	Goes on for only the first scan after operation (RUN) has been started, and goes off for the second and subsequent scans.
R9014	**Initial (off type) pulse relay**	Goes off for only the first scan after operation (RUN) has been started, and goes on for the second and subsequent scans.
R9015	**Step ladder initial pulse relay (on type)**	Turns on for only the first scan of a process after the boot at the step ladder control.
R9016	**Not used**	
R9017	**Not used**	
R9018	**0.01 s clock pulse relay**	Repeats on/off operations in 0.01 sec. cycles. 
R9019	**0.02 s clock pulse relay**	Repeats on/off operations in 0.02 s. cycles. 
R901A	**0.1 s clock pulse relay**	Repeats on/off operations in 0.1 s. cycles. 
R901B	**0.2 s clock pulse relay**	Repeats on/off operations in 0.2 s. cycles. 
R901C	**1 s clock pulse relay**	Repeats on/off operations in 1 s. cycles. 
R901D	**2 s clock pulse relay**	Repeats on/off operations in 2 s. cycles. 
R901E	**1 min clock pulse relay**	Repeats on/off operations in 1 min. cycles. 
R901F	**Not used**	

A: Available, N/A: Not available

WR902 FP0R

Relay No.	Name	Description
R9020	**RUN mode flag**	Turns off while the mode selector is set to PROG. Turns on while the mode selector is set to RUN.
R9021	**Not used**	
R9022	**Not used**	
R9023	**Not used**	
R9024	**Not used**	
R9025	**Not used**	
R9026	**Message flag**	Turns on while the F149 (MSG) instruction is executed.
R9027	**Not used**	
R9028	**Not used**	
R9029	**Forcing flag**	Turns on during forced on/off operation for input/output relay timer/counter contacts.
R902A	**Interrupt enable flag**	Turns on while the external interrupt trigger is enabled by the ICTL instruction.
R902B	**Interrupt error flag**	Turns on when an interrupt error occurs.
R902C	**Sample point flag**	Sampling by the instruction=0 Sampling at constant time intervals=1
R902D	**Sample trace end flag**	When the sampling operation stops=1, When the sampling operation starts=0
R902E	**Sampling stop trigger flag**	When the sampling stop trigger activates=1 When the sampling stop trigger stops=0
R902F	**Sampling enable flag**	When sampling starts=1 When sampling stops=0

A: Available, N/A: Not available

WR903 FP0R

Relay No.	Name	Description
R9030	**Not used**	
R9031	**Not used**	
R9032	**COM port communication mode flag**	- Turns on when the general-purpose communication function is being used - Goes off when the MEWTOCOL-COM or the PLC link function is being used.
R9033	**Print instruction execution flag**	Off: Printing is not executed. On: Execution is in progress.
R9034	**RUN overwrite complete flag**	Goes on for ony the first scan following completion of a rewrite during the RUN operation.
R9035	**Not used**	
R9036	**Not used**	
R9037	**COM port communication error flag**	- Goes on is a transmission error occurs during data communication. - Goes off when a request is made to send data, using the F159 (MTRN) instruction.
R9038	**COM port reception done flag during general purpose communication**	- Turns on when the terminator is received during general -purpose serial communication.
R9039	**COM port transmission done flag during general-purpose serial communication**	- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose serial communication.
R903A	**Not used**	
R903B	**Not used**	
R903C	**Not used**	
R903D	**Not used**	
R903E	**TOOL port reception done flag during general purpose communication**	- Turns on the terminator is received during general -purpose serial communication.
R903F	**TOOL port transmission done flag during general-purpose serial communication**	- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose serial communication.

A: Available, N/A: Not available

Note) R9030 to R9030F can be changed during 1 scan.

WR904 FP0R

Relay No.	Name	Description
R9040	**TOOL port operation mode flag**	- Turns on when the general-purpose communication function is being used - Goes off when the computer link function is being used.
R9041	**COM port PLC link flag**	Turn on while the PLC link function is used.
R9042	**Not used**	
R9043	**Not used**	
R9044	**COM port SEND/RECV instruction execution flag**	Monitors whether the F145 (SEND) or F146 (RECV) instructions can be executed or not. Off: None of the above mentioned instructions can be executed. (During executing the instruction) On: One of the above mentioned instructions can be executed.
R9045	**COM port SEND/RECV instruction execution end flag**	Monitors if an abnormality has been detected during the execution of the F145 (SEND) or F146 (RECV) instructions as follows: Off: No abonormality detected. On: An abnormality detected. (communication error) End code: DT90124
R9046	**Not used**	
R9047	**Not used**	
R9048	**Not used**	
R9049	**Not used**	
R904A	**Not used**	
R904B	**Not used**	
R904C to R904F	**Not used**	

A: Available, N/A: Not available
Note) R9040 to R904F can be changed during 1 scan.

WR905 FP0R

Relay No.	Name	Description
R9050	**MEWNET-W0 PLC link transmission error flag**	When using MEWNET-W0 - Turns on when a transmission error occurs at PLC link. - Turns on when there is an error in the PLC link area settings.
R9051 to R905F	**Not used**	

WR906 FP0R

Relay No.	Name		Description
R9060	**MEWNET-W0 PC(PLC) link 0 trans-mission assurance relay**	**Unit No.1**	Turns on when Unit No. 1 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9061		**Unit No.2**	Turns on when Unit No. 2 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9062		**Unit No.3**	Turns on when Unit No. 3 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9063		**Unit No.4**	Turns on when Unit No. 4 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9064		**Unit No.5**	Turns on when Unit No. 5 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9065		**Unit No.6**	Turns on when Unit No. 6 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9066		**Unit No.7**	Turns on when Unit No. 7 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9067		**Unit No.8**	Turns on when Unit No. 8 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9068		**Unit No.9**	Turns on when Unit No. 9 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9069		**Unit No.10**	Turns on when Unit No. 10 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906A		**Unit No.11**	Turns on when Unit No. 11 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906B		**Unit No.12**	Turns on when Unit No. 12 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906C		**Unit No.13**	Turns on when Unit No. 13 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906D		**Unit No.14**	Turns on when Unit No. 14 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906E		**Unit No.15**	Turns on when Unit No. 15 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906F		**Unit No.16**	Turns on when Unit No. 16 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.

A: Available, N/A: Not available

WR907 FP0R

Relay No.	Name		Description
R9070	**MEWNET-W0 PC(PLC) link 0 operation mode relay**	**Unit No.1**	Turns on when Unit No. 1 is in the RUN mode. Turns off when Unit No. 1 is in the PROG. mode.
R9071		**Unit No.2**	Turns on when Unit No. 2 is in the RUN mode. Turns off when Unit No. 2 is in the PROG. mode.
R9072		**Unit No.3**	Turns on when Unit No. 3 is in the RUN mode. Turns off when Unit No. 3 is in the PROG. mode.
R9073		**Unit No.4**	Turns on when Unit No. 4 is in the RUN mode. Turns off when Unit No. 4 is in the PROG. mode.
R9074		**Unit No.5**	Turns on when Unit No. 5 is in the RUN mode. Turns off when Unit No. 5 is in the PROG. mode.
R9075		**Unit No.6**	Turns on when Unit No. 6 is in the RUN mode. Turns off when Unit No. 6 is in the PROG. mode.
R9076		**Unit No.7**	Turns on when Unit No. 7 is in the RUN mode. Turns off when Unit No. 7 is in the PROG. mode.
R9077		**Unit No.8**	Turns on when Unit No. 8 is in the RUN mode. Turns off when Unit No. 8 is in the PROG. mode.
R9078		**Unit No.9**	Turns on when Unit No. 9 is in the RUN mode. Turns off when Unit No. 9 is in the PROG. mode.
R9079		**Unit No.10**	Turns on when Unit No. 10 is in the RUN mode. Turns off when Unit No. 10 is in the PROG. mode.
R907A		**Unit No.11**	Turns on when Unit No. 11 is in the RUN mode. Turns off when Unit No. 11 is in the PROG. mode.
R907B		**Unit No.12**	Turns on when Unit No. 12 is in the RUN mode. Turns off when Unit No. 12 is in the PROG. mode.
R907C		**Unit No.13**	Turns on when Unit No. 13 is in the RUN mode. Turns off when Unit No. 13 is in the PROG. mode.
R907D		**Unit No.14**	Turns on when Unit No. 14 is in the RUN mode. Turns off when Unit No. 14 is in the PROG. mode.
R907E		**Unit No.15**	Turns on when Unit No. 15 is in the RUN mode. Turns off when Unit No. 15 is in the PROG. mode.
R907F		**Unit No.16**	Turns on when Unit No. 16 is in the RUN mode. Turns off when Unit No. 16 is in the PROG. mode.

A: Available, N/A: Not available

WR908 FP0R

Relay No.	Name		Description
R9080	**MEWNET-W0 PC(PLC) link 1 trans-mission assurance relay**	**Unit No.1**	Turns on when Unit No. 1 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9081		**Unit No.2**	Turns on when Unit No. 2 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9082		**Unit No.3**	Turns on when Unit No. 3 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9083		**Unit No.4**	Turns on when Unit No. 4 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9084		**Unit No.5**	Turns on when Unit No. 5 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9085		**Unit No.6**	Turns on when Unit No. 6 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9086		**Unit No.7**	Turns on when Unit No. 7 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9087		**Unit No.8**	Turns on when Unit No. 8 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9088		**Unit No.9**	Turns on when Unit No. 9 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9089		**Unit No.10**	Turns on when Unit No. 10 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908A		**Unit No.11**	Turns on when Unit No. 11 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908B		**Unit No.12**	Turns on when Unit No. 12 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908C		**Unit No.13**	Turns on when Unit No. 13 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908D		**Unit No.14**	Turns on when Unit No. 14 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908E		**Unit No.15**	Turns on when Unit No. 15 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908F		**Unit No.16**	Turns on when Unit No. 16 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.

A: Available, N/A: Not available

WR909 FP0R

Relay No.	Name		Description
R9090	**MEWNET-W0 PC(PLC) link 1 operation mode relay**	**Unit No.1**	Turns on when Unit No. 1 is in the RUN mode. Turns off when Unit No. 1 is in the PROG. mode.
R9091		**Unit No.2**	Turns on when Unit No. 2 is in the RUN mode. Turns off when Unit No. 2 is in the PROG. mode.
R9092		**Unit No.3**	Turns on when Unit No. 3 is in the RUN mode. Turns off when Unit No. 3 is in the PROG. mode.
R9093		**Unit No.4**	Turns on when Unit No. 4 is in the RUN mode. Turns off when Unit No. 4 is in the PROG. mode.
R9094		**Unit No.5**	Turns on when Unit No. 5 is in the RUN mode. Turns off when Unit No. 5 is in the PROG. mode.
R9095		**Unit No.6**	Turns on when Unit No. 6 is in the RUN mode. Turns off when Unit No. 6 is in the PROG. mode.
R9096		**Unit No.7**	Turns on when Unit No. 7 is in the RUN mode. Turns off when Unit No. 7 is in the PROG. mode.
R9097		**Unit No.8**	Turns on when Unit No. 8 is in the RUN mode. Turns off when Unit No. 8 is in the PROG. mode.
R9098		**Unit No.9**	Turns on when Unit No. 9 is in the RUN mode. Turns off when Unit No. 9 is in the PROG. mode.
R9099		**Unit No.10**	Turns on when Unit No. 10 is in the RUN mode. Turns off when Unit No. 10 is in the PROG. mode.
R909A		**Unit No.11**	Turns on when Unit No. 11 is in the RUN mode. Turns off when Unit No. 11 is in the PROG. mode.
R909B		**Unit No.12**	Turns on when Unit No. 12 is in the RUN mode. Turns off when Unit No. 12 is in the PROG. mode.
R909C		**Unit No.13**	Turns on when Unit No. 13 is in the RUN mode. Turns off when Unit No. 13 is in the PROG. mode.
R909D		**Unit No.14**	Turns on when Unit No. 14 is in the RUN mode. Turns off when Unit No. 14 is in the PROG. mode.
R909E		**Unit No.15**	Turns on when Unit No. 15 is in the RUN mode. Turns off when Unit No. 15 is in the PROG. mode.
R909F		**Unit No.16**	Turns on when Unit No. 16 is in the RUN mode. Turns off when Unit No. 16 is in the PROG. mode.

A: Available, N/A: Not available

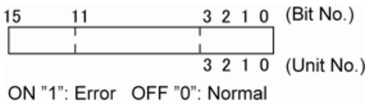
WR910 FP0R

Relay No.	Name		Description
R9110	High-speed counter control flag	HSC-CH0	- Turns on the channel of high-speed counter during the control using F165(CAM0), F166(HC1S), F167(HC1R), F178(PLSM) instructions. - Turns off when the control is cleared or this instruction is completed.
R9111		HSC-CH1	
R9112		HSC-CH2	
R9113		HSC-CH3	
R9114		HSC-CH4	
R9115		HSC-CH5	
R9116 to R911F	Not used		
R9120	Pulse output instruction flag	PLS-CH0	- Turns on while the pulses are being output using F171(SPDH), F172 (PLSH), F173(PWMH), F174 (SP0H), F175(SPSH), F177(HOME) instructions.
R9121		PLS-CH1	
R9122		PLS-CH2	
R9123		PLS-CH3	
R9124 to R912F	Not used		
R9130	Pulse output control flag	PLS-CH0	- Turns on the channel of pulse output during the control using F166(HC1S), F167(HC1R) instructions. - Turns off when the control is cleared or this instruction is completed.
R9131		PLS-CH1	
R9132		PLS-CH2	
R9133		PLS-CH3	
R9134 to R913F	Not used		

A: Available, N/A: Not available

5.1.9 Table of Special Data Registers for FP0R

FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90000	**Self-diagnostic error code**	The self-diagnostic error code is stored here when a self-diagnostic error occurs.	A	N/A
DT90001	**Not used**	-	N/A	N/A
DT90002	**Not used**		N/A	N/A
DT90003	**Not used**	-	N/A	N/A
DT90004	**Not used**	-	N/A	N/A
DT90005	**Not used**	-	N/A	N/A
DT90006	**Not used**	-	N/A	N/A
DT90007	**Not used**	-	N/A	N/A
DT90008	**Not used**	-	N/A	N/A
DT90009	**Not used**	-	N/A	N/A
DT90010	**Extension (right side) I/O verify error unit [0 to 3]**	When the state of installation of FP0 expansion I/O unit has changed since the power was turned on, the bit corresponding to the unit No. will turn on. Monitor using binary display. 	A	N/A
DT90011	**Not used**	-	N/A	N/A
DT90012	**Not used**	-	N/A	N/A
DT90013	**Not used**	-	N/A	N/A

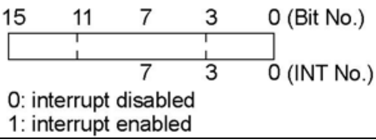
FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90014	**Operation auxiliary register for data shift instruction**	One shift-out hexadecimal digit is stored in bit positions 0 to 3 when the data shift instruction, F105 (BSR) or F106 (BSL) is executed. The value can be read and written by executing F0 (MV) instruction.	A	A
DT90015	**Operation auxiliary register for division instruction**	The divided remainder (16-bit) is stored in DT90015 when the division instruction F32(%) or F52(B%) instruction is executed. The divided remainder (32-bit) is stored in DT90015 and DT90016 when the division instruction F33(D%) or F53(DB%) is executed. The value can be read and written by executing F0(MV) instruction.	A	A
DT90016			A	A
DT90017	**Operation error address (hold type)**	After commencing operation, the address where the first operation error occurred is stored. Monitor the address using decimal display.	A	N/A
DT90018	**Operation error address (latest type)**	The address where an operation error occurred is stored. Each time an error occurs, the new address overwrites the previous address.	A	N/A
DT90019	**2.5 ms ring counter** **Note1)**	The data stored here is increased by one every 2.5 ms. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 2.5 ms = Elapsed time between the two points.	A	N/A
DT90020	**10 μs ring counter** **Note1) Note2)**	The data stored here is increased by one every 10.67 μs. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 10.67 μs = Elapsed time between the two points. Note) The exact value is 10.67 μs.	A	N/A
DT90021	**Not used**	-	N/A	N/A

Note1) It is renewed once at the beginning of each one scan.

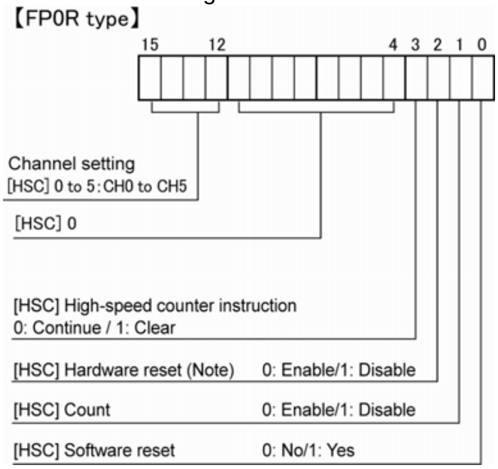
Note2) As DT90020 is renewed even if F0(MV), DT90020 and D instruction is being executed, it can be used to measure the block time.

FP0R (A: Available, N/A: Not available)

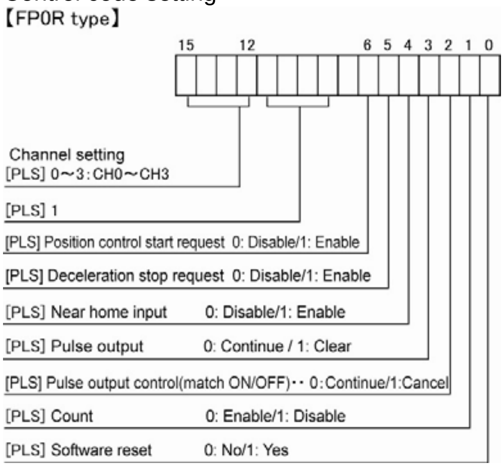
Address	Name	Description	Read-ing	Writ-ing
DT90022	**Scan time (current value)** Note)	The current scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.	A	N/A
DT90023	**Scan time (minimum value)** Note)	The minimum scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.	A	N/A
DT90024	**Scan time (maximum value)** Note)	The maximum scan time is stored here. The scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K125 indicates 12.5 ms.	A	N/A
DT90025	**Mask condition monitoring register for interrupts (INT0 to 11)**	The mask conditions of interrupts using the instruction can be stored here. Monitor using binary display.  	A	N/A
DT90026	**Not used**	-	N/A	N/A
DT90027	**Periodical interrupt interval (INT24)**	The value set by ICTL instruction is stored. K0: periodical interrupt is not used. K1 to K3000: 0.5ms to 1.5s or 10ms to 30s	A	N/A
DT90028	**Sample trace interval**	K0: Sampling by the SMPL instruction K1 to K3000 (x 10 ms): 10 ms to 30 s	A	N/A
DT90029	**Not used**	-	N/A	N/A
DT90030 **DT90031** **DT90032** **DT90033** **DT90034** **DT90035**	**Character storage by F149 MSG instruction**	The contents of the specified message (Data length) are stored in these special data registers when F149 (MSG) instruction is executed.	A	N/A
DT90036	**Not used**	-	N/A	N/A

Note) Scan time display is only possible in RUN mode, and shows the operation cycle time. (In PROG. mode, the scan time for the operation is not displayed.) The maximum and minimum values are cleared each time the mode is switched from RUN to PROG.

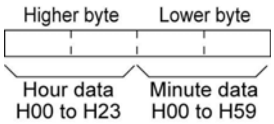
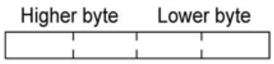
FP0R (A: Available, N/A: Not available)

Address	Name	Description	Reading	Writing
DT90037	**Work1 for SRC instructions**	The number of data that match the searched data is stored here when F96 (SRC) insturction is executed.	A	N/A
DT90038	**Work2 for SRC instructions**	The position of the first matching data is stored here when an F96 (SRC) instruction is executed.	A	N/A
DT90039	**Not used**	-	N/A	N/A
DT90040	**Not used**	-	N/A	N/A
DT90041	**Not used**	-	N/A	N/A
DT90042	**Not used**	-	N/A	N/A
DT90043	**Not used**	-	N/A	N/A
DT90044	**Not used**	-	N/A	N/A
DT90045	**Not used**	-	N/A	N/A
DT90046	**Not used**	-	N/A	N/A
DT90047	**Not used**	-	N/A	N/A
DT90048	**Not used**	-	N/A	N/A
DT90049	**Not used**	-	N/A	N/A
DT90050	**Not used**	-	N/A	N/A
DT90051	**Not used**	-	N/A	N/A
DT90052	**High-speed counter control flag**	The pulse output instruction can be continued or cleared by writing a value with MV instruction (F0). Control code setting 	A	A

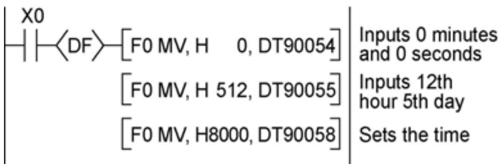
FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90052	**Pulse output control flag**	A value can be written with F0 (MV) instruction to reset the high-speed counter, disable counting, continue or clear high-speed counter instruction. Control code setting 	A	A

FP0R (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90053</td><td>Clock/calender monitor (hour/minute) (T32 only)</td><td>Hour and minute data of the clock/calender are stored here. This data is read-only data. It cannot be overwritten.
</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90054</td><td>Clock/calender setting (minute/second) (T32 only)</td><td rowspan="4">The year, month, day, hour, minute, second and day-of-the-week data for the clock/calender is stored. The built-in clock/calender will operate correctly through the year 2099 and supports leap years. The clock/calender can be set by writing a value using a programming tool software or a program that uses the F0 (MV) instruction.(see example for DT90058)
<table><tr><td>DT90054</td><td>Minute data (H00 to H59)</td><td>Second data (H00 to H59)</td></tr><tr><td>DT90055</td><td>Day data (H01 to H31)</td><td>Hour data (H00 to H23)</td></tr><tr><td>DT90056</td><td>Year data (H00 to H99)</td><td>Month data (H01 to H12)</td></tr><tr><td>DT90057</td><td>–</td><td>Day-of-the-week (H00 to H06)</td></tr></table>
As a day of the week is not automatially set on FPWIN GR, fix what day is set to 00, and set each value for 00 to 06.</td><td rowspan="4">A</td><td rowspan="4">A</td></tr>
<tr><td>DT90055</td><td>Clock/calender setting (day/hour) (T32 only)</td></tr>
<tr><td>DT90056</td><td>Clock/calender setting (year/month) (T32 only)</td></tr>
<tr><td>DT90057</td><td>Clock/calender setting (day-of-the-week) (T32 only)</td></tr>
</table>

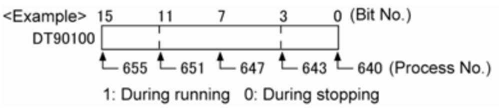
FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90058	**Clock/calender setting and 30 seconds correction register (T32 only)**	The clock/calender is adjusted as follows. **When setting the clock/calender by program** By setting the highest bit of DT90058 to 1, the time becomes that written to DT90054 to DT90057 by F0 (MV) instruction. After the time is set, DT90058 is cleared to 0. (Cannot be performed with any instruction other than F0 (MV) instruction.) <Example> Set the time to 12:00:00 on the 5th day when the X0 turns on.  Note) If the values of DT90054 to DT90057 are changed with the programming tool software, the time will be set when the new values are written. Therefore, it is unnecessary to write to DT90058. **When the correcting times less than 30 seconds** By setting the lowest bit of DT90058 to 1, the value will be moved up or down and become exactly 0 seconds. After the correction is completed, DT90058 is cleared to 0. Example: Correct to 0 seconds with X0: on  At the time of coorection, if between 0 and 29 seconds, it will be moved down, and if between 30 and 59 seconds, it will be moved up. In the example above, if the time was 5 minutes 29 seconds, it will become 5 minutes 0 seconds;and, if the time was 5 minutes 35 seconds, it will become 6 minutes 0 seconds.	A	A
DT90059	**Communication error code**	Error code is sotred here when a communication error occurs.	N/A	N/A

FP0R (A: Available, N/A: Not available)

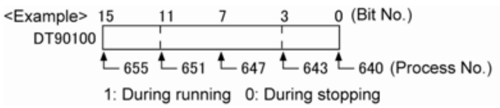
<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90060</td><td>Step ladder process (0 to 15)</td><td rowspan="22">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on.

Monitor using binary display.

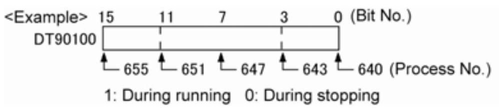


Note) A programming tool software can be used to write data.</td><td rowspan="22">A</td><td rowspan="22">A
Note)</td></tr>
<tr><td>DT90061</td><td>Step ladder process (16 to 31)</td></tr>
<tr><td>DT90062</td><td>Step ladder process (32 to 47)</td></tr>
<tr><td>DT90063</td><td>Step ladder process (48 to 63)</td></tr>
<tr><td>DT90064</td><td>Step ladder process (64 to 79)</td></tr>
<tr><td>DT90065</td><td>Step ladder process (80 to 95)</td></tr>
<tr><td>DT90066</td><td>Step ladder process (96 to 111)</td></tr>
<tr><td>DT90067</td><td>Step ladder process (112 to 127)</td></tr>
<tr><td>DT90068</td><td>Step ladder process (128 to 143)</td></tr>
<tr><td>DT90069</td><td>Step ladder process (144 to 159)</td></tr>
<tr><td>DT90070</td><td>Step ladder process (160 to 175)</td></tr>
<tr><td>DT90071</td><td>Step ladder process (176 to 191)</td></tr>
<tr><td>DT90072</td><td>Step ladder process (192 to 207)</td></tr>
<tr><td>DT90073</td><td>Step ladder process (208 to 223)</td></tr>
<tr><td>DT90074</td><td>Step ladder process (224 to 239)</td></tr>
<tr><td>DT90075</td><td>Step ladder process (240 to 255)</td></tr>
<tr><td>DT90076</td><td>Step ladder process (256 to 271)</td></tr>
<tr><td>DT90077</td><td>Step ladder process (272 to 287)</td></tr>
<tr><td>DT90078</td><td>Step ladder process (288 to 303)</td></tr>
<tr><td>DT90079</td><td>Step ladder process (304 to 319)</td></tr>
<tr><td>DT90080</td><td>Step ladder process (320 to 335)</td></tr>
<tr><td>DT90081</td><td>Step ladder process (336 to 351)</td></tr>
</table>

FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90082	**Step ladder process (352 to 367)**	Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on . Monitor using binary display.  Note) A programming tool software can be used to write data.	A	A Note)
DT90083	**Step ladder process (368 to 383)**			
DT90084	**Step ladder process (384 to 399)**			
DT90085	**Step ladder process (400 to 415)**			
DT90086	**Step ladder process (416 to 431)**			
DT90087	**Step ladder process (432 to 447)**			
DT90088	**Step ladder process (448 to 463)**			
DT90089	**Step ladder process (464 to 479)**			
DT90090	**Step ladder process (480 to 495)**			
DT90091	**Step ladder process (496 to 511)**			
DT90092	**Step ladder process (512 to 527)**			
DT90093	**Step ladder process (528 to 543)**			
DT90094	**Step ladder process (544 to 559)**			
DT90095	**Step ladder process (560 to 575)**			
DT90096	**Step ladder process (576 to 591)**			
DT90097	**Step ladder process (592 to 607)**			

FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90098	Step ladder process (608 to 623)	Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on. Monitor using binary display  A programming tool software can be used to write data.	A	A
DT90099	Step ladder process (624 to 639)			
DT90100	Step ladder process (640 to 655)			
DT90101	Step ladder process (656 to 671)			
DT90102	Step ladder process (672 to 687)			
DT90103	Step ladder process (688 to 703)			
DT90104	Step ladder process (704 to 719)			
DT90105	Step ladder process (720 to 735)			
DT90106	Step ladder process (736 to 751)			
DT90107	Step ladder process (752 to 767)			
DT90108	Step ladder process (768 to 783)			
DT90109	Step ladder process (784 to 799)			
DT90110	Step ladder process (800 to 815)			
DT90111	Step ladder process (816 to 831)			
DT90112	Step ladder process (832 to 847)			
DT90113	Step ladder process (848 to 863)			
DT90114	Step ladder process (864 to 879)			
DT90115	Step ladder process (880 to 895)			
DT90116	Step ladder process (896 to 911)			
DT90117	Step ladder process (912 to 927)			
DT90118	Step ladder process (928 to 943)			
DT90119	Step ladder process (944 to 959)			
DT90120	Step ladder process (960 to 975)			
DT90121	Step ladder process (976 to 991)			
DT90122	Step ladder process (992 to 999) (higher byte is not used.)			

FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90123	**Not used**	-	N/A	N/A
DT90124	**COM SEND/RECV instruction end code**	For details, refer to Programming Manual (F145 and F146).	N/A	N/A
DT90125	**Not used**	-	N/A	N/A
DT90126	**Forced ON/OFF operating station display**	Used by the system	N/A	N/A
DT90127 to DT90139	**Not used**	-	N/A	N/A
DT90140	**MEWNET-W0 PC(PLC) link 0 status**	The number of times the receiving operation is performed.	A	N/A
DT90141		The current interval between two receiving operations: value in the register x 2.5ms		
DT90142		The minimum inerval between two receiving operations: value in the register x 2.5ms		
DT90143		The maximum interval between two receiving operations: value in the register x 2.5ms		
DT90144		The number of times the sending operation is performed.		
DT90145		The current interval between two sending operations: value in the register x 2.5ms		
DT90146		The minimum interval between two sending operations: value in the register x 2.5ms		
DT90147		The maximum interval between two sending operations: value in the register x 2.5ms		
DT90148	**MEWNET-W0 PC(PLC) link 1 status**	The number of times the receiving operation is performed.	A	N/A
DT90149		The current interval between two receiving operations: value in the register x 2.5ms		
DT90150		The minimum inerval between two receiving operations: value in the register x 2.5ms		
DT90151		The maximum interval between two receiving operations: value in the register x 2.5ms		
DT90152		The number of times the sending operation is performed.		
DT90153		The current interval between two sending operations: value in the register x 2.5ms		
DT90154		The minimum interval between two sending operations: value in the register x 2.5ms		
DT90155		The maximum interval between two sending operations: value in the register x 2.5ms		
DT90156	**MEWNET-W0 PC(PLC) link 0 status**	Area used for measurement of receiving interval.	A	N/A
DT90157		Area used for measurement of sending interval.		

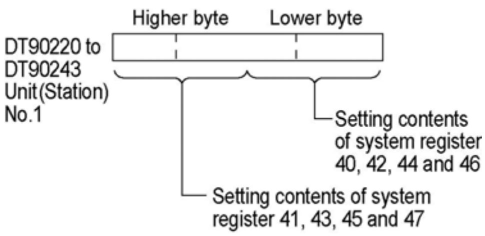
FP0R (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90158	**MEWNET-W0 PC(PLC) link 1 status**	Area used for measurement of receiving interval.	A	N/A
DT90159		Area used for measurement of sending interval.		
DT90160	**MEWNET-W0 PC(PLC) link 0 unit No.**	Stores the unit No. of PC(PLC) link 0.	A	N/A
DT90161	**MEWNET-W0 PC(PLC) link 0 error flag**	Stores the error contents of PC(PLC) link 0.	A	N/A
DT90162 to DT90169	**Not used**	-	N/A	N/A
DT90170	**MEWNET-W0 PC(PLC) link 0 status**	Duplicated destination for PC(PLC) inter-link address	A	N/A
DT90171		Counts how many times a token is lost.		
DT90172		Counts how many times two or more tokens are detected.		
DT90173		Counts how many times a signal is lost.		
DT90174		No. of times underfined commands have been received.		
DT90175		No. of times sum check errors have occurred during reception.		
DT90176		No. of times format errors have occurred in received data.		
DT90177		No. of times transmission errors have occurred.		
DT90178		No. of times procedural errors have occurred.		
DT90179		No. of times overlapping parent units have occurred.		
DT90180 to DT90189	**Not used**	-	N/A	N/A
DT90190	**Not used**	-	N/A	N/A
DT90191	**Not used**	-	N/A	N/A
DT90192	**Not used**	-	N/A	N/A
DT90193	**Not used**	-	N/A	N/A
DT90194 to DT90218	**Not used**	-	N/A	N/A

FP0R (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th colspan="2">Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90219</td><td colspan="2">Unit No. (Station No.) selection for DT90220 to DT90251</td><td>0: Unit No. (Station No.) 1 to 8,
1: Unit No. (Station No.) 9 to 16</td><td>A</td><td>A</td></tr>
<tr><td>DT90220</td><td rowspan="4">PC(PLC) link Unit (station) No. 1 or 9</td><td>System register 40 and 41</td><td rowspan="20">The contents of the system register settings partaining to the PLC inter-link function for the various unit numbers are stored as shown below.

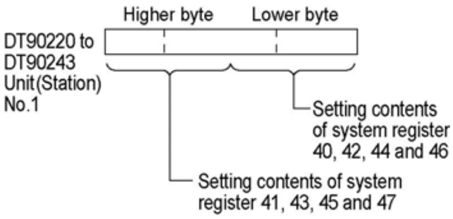
<Example>
When DT90219 is 0



• When the system register 46 in the home unit is in the standard setting, the values in the home unit are copied in the system registers 46 and 47.
When the system register 46 in the home unit is in the reverse setting, the registers 40 to 45 and 47 corresponding to the home unit mentioned in the left column will be changed to 50 to 55 and 57, and the system register 46 will be set as it is.
Also, the system registers 40 to 45 corresponding to other units will be changed to the values which the received values are corrected, and the registers 46 and 57 in the home unit are set for the registers 46 and 47.</td><td rowspan="20">A</td><td rowspan="20">N/A</td></tr>
<tr><td>DT90221</td><td>System register 42 and 43</td></tr>
<tr><td>DT90222</td><td>System register 44 and 45</td></tr>
<tr><td>DT90223</td><td>System register 46 and 47</td></tr>
<tr><td>DT90224</td><td rowspan="4">PC(PLC) link Unit (station) No. 2 or 10</td><td>System register 40 and 41</td></tr>
<tr><td>DT90225</td><td>System register 42 and 43</td></tr>
<tr><td>DT90226</td><td>System register 44 and 45</td></tr>
<tr><td>DT90227</td><td>System register 46 and 47</td></tr>
<tr><td>DT90228</td><td rowspan="4">PC(PLC) link Unit (station) No. 3 or 11</td><td>System register 40 and 41</td></tr>
<tr><td>DT90229</td><td>System register 42 and 43</td></tr>
<tr><td>DT90230</td><td>System register 44 and 45</td></tr>
<tr><td>DT90231</td><td>System register 46 and 47</td></tr>
<tr><td>DT90232</td><td rowspan="4">PC(PLC) link Unit (station) No. 4 or 12</td><td>System register 40 and 41</td></tr>
<tr><td>DT90233</td><td>System register 42 and 43</td></tr>
<tr><td>DT90234</td><td>System register 44 and 45</td></tr>
<tr><td>DT90235</td><td>System register 46 and 47</td></tr>
<tr><td>DT90236</td><td rowspan="4">PC(PLC) link Unit (station) No. 5 or 13</td><td>System register 40 and 41</td></tr>
<tr><td>DT90237</td><td>System register 42 and 43</td></tr>
<tr><td>DT90238</td><td>System register 44 and 45</td></tr>
<tr><td>DT90239</td><td>System register 46 and 47</td></tr>
</table>

FP0R (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th colspan="2">Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90240</td><td rowspan="4">PC(PLC) link Unit (station) No. 6 or 14</td><td>System regis-ter 40 and 41</td><td rowspan="12">The contents of the system register settings partaining to the PLC inter-link function for the various unit numbers are stored as shown below.

<Example> when DT90219 is 0.


• When the system register 46 in the home unit is in the standard setting, the values in the home unit are copied in the system registers 46 and 47. When the system register 46 in the home unit is in the reverse setting, the registers 40 to 45 and 47 corresponding to the home unit mentioned in the left column will be changed to 50 to 55 and 57, and the system register 46 will be set as it is. Also, the system registers 40 to 45 corresponding to other units will be changed to the values which the received values are corrected, and the registers 46 and 57 in the home unit are set for the registers 46 and 47.</td><td rowspan="12">A</td><td rowspan="12">N/A</td></tr>
<tr><td>DT90241</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90242</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90243</td><td>System regis-ter 46 and 47</td></tr>
<tr><td>DT90244</td><td rowspan="4">PC(PLC) link Unit (sta-tion) No. 7 or 15</td><td>System regis-ter 40 and 41</td></tr>
<tr><td>DT90245</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90246</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90247</td><td>System regis-ter 46 and 47</td></tr>
<tr><td>DT90248</td><td rowspan="4">PC(PLC) link Unit (sta-tion) No. 8 or 16</td><td>System regis-ter 40 and 41</td></tr>
<tr><td>DT90249</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90250</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90251</td><td>System regis-ter 46 and 47</td></tr>
<tr><td>DT90252</td><td colspan="2">Not used</td><td></td><td rowspan="4">N/A</td><td rowspan="4">N/A</td></tr>
<tr><td>DT90253</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90254</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90255</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90256</td><td colspan="2">Not used</td><td></td><td>N/A</td><td>N/A</td></tr>
</table>

FP0R (A: Available, N/A: Not available)

Address	Name			Description	Read-ing	Writ-ing
DT90300	**Elapsed value area**	**Lower words**	**HSC-CH0**	Counting area for input (X0) or (X0, X1) of the main unit.	A	A Note)
DT90301		**Higher words**			A	A Note)
DT90302	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90303		**Higher words**			A	A Note)
DT90304	**Elapsed value area**	**Lower words**	**HSC-CH1**	Counting area for input (X1) of the main unit.	A	A Note)
DT90305		**Higher words**			A	A Note)
DT90306	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90307		**Higher words**			A	A Note)
DT90308	**Elapsed value area**	**Lower words**	**HSC-CH2**	Counting area for input (X2) or (X2, X3) of the main unit.	A	A Note)
DT90309		**Higher words**			A	A Note)
DT90310	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90311		**Higher words**			A	A Note)
DT90312	**Elapsed value area**	**Lower words**	**HSC-CH3**	Counting area for input (X3) of the main unit.	A	A Note)
DT90313		**Higher words**			A	A Note)
DT90314	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90315		**Higher words**			A	A Note)
DT90316	**Elapsed value area**	**Lower words**	**HSC-CH4**	Counting area for input (X4) or (X4, X5) of the main unit.	A	A Note)
DT90317		**Higher words**			A	A Note)
DT90318	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90319		**Higher words**			A	A Note)

Note) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F166 (HC1S) and F167 (HC1R) instructions only.

FP0R (A: Available, N/A: Not available)

Address	Name			Description	Reading	Writing
DT90320	Elapsed value area	Lower words	HSC-CH5	Counting area for input (X5) of the main unit.	A	A Note1)
DT90321		Higher words			A	A Note1)
DT90322	Target value area	Lower words		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90323		Higher words			A	A Note1)
DT90324	Not used				N/A	N/A
DT90325	Not used				N/A	N/A
DT90326	Not used				N/A	N/A
DT90327	Not used				N/A	N/A
DT90328	Not used				N/A	N/A
DT90329	Not used				N/A	N/A
DT90330	Not used				N/A	N/A
DT90331	Not used				N/A	N/A
DT90332	Not used				N/A	N/A
DT90333	Not used				N/A	N/A
DT90334	Not used				N/A	N/A
DT90335	Not used				N/A	N/A
DT90336	Not used				N/A	N/A
DT90337	Not used				N/A	N/A
DT90338	Not used				N/A	N/A
DT90339	Not used				N/A	N/A
DT90340	Not used				N/A	N/A
DT90341	Not used				N/A	N/A
DT90342	Not used				N/A	N/A
DT90343	Not used				N/A	N/A
DT90344	Not used				N/A	N/A
DT90345	Not used				N/A	N/A
DT90346	Not used				N/A	N/A
DT90347	Not used				N/A	N/A
DT90348	Not used				N/A	N/A
DT90349	Not used				N/A	N/A
DT90350	Not used				N/A	N/A
DT90351	Not used				N/A	N/A
DT90352	Not used				N/A	N/A
DT90353	Not used				N/A	N/A
DT90354	Not used				N/A	N/A
DT90355	Not used				N/A	N/A
DT90356	Not used				N/A	N/A
DT90357	Not used				N/A	N/A
DT90358	Not used				N/A	N/A
DT90359	Not used				N/A	N/A
DT90360	Not used				N/A	N/A
DT90361	Not used				N/A	N/A
DT90362	Not used				N/A	N/A
DT90363	Not used				N/A	N/A

Note1) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F166 (HC1S) and F167 (HC1R) instructions only.

FP0R (A: Available, N/A: Not available)

Address	Name		Description	Read-ing	Writ-ing
DT90370	**Control flag monitor area**	**HSC-CH0**	When HSC control is executed by F0 (MV)S, DT90052 instruction, the setting value for the target CH is stored in each CH.	A	N/A
DT90371		**HSC-CH1**		A	N/A
DT90372		**HSC-CH2**		A	N/A
DT90373		**HSC-CH3**		A	N/A
DT90374		**HSC-CH4**		A	N/A
DT90375		**HSC-CH5**		A	N/A
DT90376	**Not used**		-	N/A	N/A
DT90377	**Not used**		-	N/A	N/A
DT90378	**Not used**		-	N/A	N/A
DT90379	**Not used**		-	N/A	N/A
DT90380	**Control flag monitor area (Transistor output type only)**	**PLS-CH0**	When pulse output control is executed by F0 (MV)S, DT90052 instruction, the setting value for the target CH is stored in each CH.	A	N/A
DT90381		**PLS-CH1**		A	N/A
DT90382		**PLS-CH2**		A	N/A
DT90383		**PLS-CH3**		A	N/A
DT90384	**Not used**		-	N/A	N/A
DT90385	**Not used**		-	N/A	N/A
DT90386	**Not used**		-	N/A	N/A
DT90387	**Not used**		-	N/A	N/A
DT90388	**Not used**		-	N/A	N/A
DT90389	**Not used**		-	N/A	N/A

FP0R (A: Available, N/A: Not available)

Address	Name			Description	Read-ing	Writ-ing
DT90400	Elapsed value area	Lower words	PLS-CH0	Available for the transistor output type only. Note) When controlling the pulse output CH by F166(HC1S), F167(HC1R) instructions, the target value is stored. The target value for match ON/OFF is stored.	A	A
DT90401		Higher words			A	A
DT90402	Target value area	Lower words			A	N/A
DT90403		Higher words			A	N/A
DT90404	Target value area for match ON/OFF	Lower words			A	N/A
DT90405		Higher words			A	N/A
DT90406	Corrected speed of initial speed	Lower words		The initial speed of the calculated result is stored.	A	N/A
DT90407	Deceleration minimum speed			The minimum speed for the change of speed.	A	N/A
DT90408	Acceleration forbidden area starting position	Lower words		If the elapsed value corsses over this position when changing the speed, acceleration cannot be performed.	A	N/A
DT90409		Higher words			A	N/A
DT90410	Elapsed value area	Lower words	PLS-CH1	Available for the transistor output type only. Note) When controlling the pulse output CH by F166(HC1S), F167(HC1R) instructions, the target value is stored. The target value for match ON/OFF is stored.	A	A
DT90411		Higher words			A	A
DT90412	Target value area	Lower words			A	N/A
DT90413		Higher words			A	N/A
DT90414	Target value area for match ON/OFF	Lower words			A	N/A
DT90415		Higher words			A	N/A
DT90416	Corrected speed of initial speed	Lower words		The initial speed of the calculated result is stored.	A	N/A
DT90417	Deceleration minimum speed			The minimum speed for the change of speed.	A	N/A
DT90418	Acceleration forbidden area starting position	Lower words		If the elapsed value corsses over this position when changing the speed, acceleration cannot be performed.	A	N/A
DT90419		Higher words			A	N/A

FP0R (A: Available, N/A: Not available)

Address	Name			Description	Read-ing	Writ-ing
DT90420	Elapsed value area	Lower words	PLS-CH2	Available for the transistor output type only. Note) When controlling the pulse output CH by F166(HC1S), F167(HC1R) instructions, the target value is stored. The target value for match ON/OFF is stored.	A	A
DT90421		Higher words			A	A
DT90422	Target value area	Lower words			A	N/A
DT90423		Higher words			A	N/A
DT90424	Target value area for match ON/OFF	Lower words			A	N/A
DT90425		Higher words			A	N/A
DT90426	Corrected speed of initial speed	Lower words		The initial speed of the calculated result is stored.	A	N/A
DT90427	Deceleration minimum speed			The minimum speed for the change of speed.	A	N/A
DT90428	Acceleration forbidden area starting position	Lower words		If the elapsed value corsses over this position when changing the speed, acceleration cannot be performed.	A	N/A
DT90429		Higher words			A	N/A
DT90430	Elapsed value area	Lower words	PLS-CH3	Available for the transistor output type only. Note) When controlling the pulse output CH by F166(HC1S), F167(HC1R) instructions, the target value is stored. The target value for match ON/OFF is stored.	A	A
DT90431		Higher words			A	A
DT90432	Target value area	Lower words			A	N/A
DT90433		Higher words			A	N/A
DT90434	Target value area for match ON/OFF	Lower words			A	N/A
DT90435		Higher words			A	N/A
DT90436	Corrected speed of initial speed	Lower words		The initial speed of the calculated result is stored.	A	N/A
DT90437	Deceleration minimum speed			The minimum speed for the change of speed.	A	N/A
DT90438	Acceleration forbidden area starting position	Lower words		If the elapsed value corsses over this position when changing the speed, acceleration cannot be performed.	A	N/A
DT90439		Higher words			A	N/A

5.1.10 Table of System Registers for FP Σ

	No.	Name	Default value	Descriptions	
Hold/ Non-hold 1	**5**	Starting number setting for counter	1008	0 to 1024	• These settings are effective if the optional backup battery is installed. • If no backup battery is used, do not change the default settings. Otherwise proper functioning of hold/non-hold values cannot be guaranteed.
	6	Hold type area starting number setting for timer and counter	1008	0 to 1024	
	7	Hold type area starting number setting for internal relays	12k: 90 32k: 0 to 256	12k: 0 to 98 32k: 0 to 256	
	8	Hold type area starting number setting for data registers	32710	0 to 32765	
	14	Hold or non-hold setting for step ladder process	Non-hold	Hold/Non-hold	
	4	Previous value is held for a leading edge detection instruction (DF instrucion) with MC Note)	Hold	Hold/ Non-hold	
Hold/ Non-hold 2	**10**	Hold type area starting word number for PC(PLC) link relays (for PC(PLC) link 0)	64	0 to 64	
	11	Hold type area starting word number for PC(PLC) link relays (for PC(PLC) link 1)	128 (32k only)	64 to 128	
	12	Hold type area starting number for PC(PLC) link registers (for PC(PLC) link 0)	128	0 to 128	
	13	Hold type area starting number for PC(PLC) link registers (for PC(PLC) link 1)	256 (32k only)	128 to 256	
Action on error	**20**	Disable or enable setting for duplicated output	Disabled	Disabled/Enabled	
	23	Operation setting when an I/O verification error occurs	Stop	Stop/Continuation of operation	
	26	Operation setting when an operation error occurs	Stop	Stop/Continuation of operation	
	4	Alarm battery error (Operating setting when battery error occurs)	Disabled	Dis-abled:	When a battery error occurs, a self-diagnostic error is not issued and the ERROR/ ALARM LED does not flash.
				Ena-bled:	When a battery error occurs, a self-diagnostic error is issued and the ERROR/ ALARM LED flashes.

Note) The 12k type is available with Ver. 1.4 to 1.9, 2.4 or later.

	No.	Name	Default value	Descriptions
Time set-ting	**31**	Wait time setting for multi-frame communication	6500.0 ms	10 to 81900 ms
	32	Communication timeout setting for SEND/RECV, RMRD/RMWT commands	10000.0 ms	10 to 81900 ms
	34	Constant value settings for scan time	Normal scan	0: Normal scan 0 to 350 ms: Scans once each specified time interval
PC (PLC) link 0 set-ting	**40**	Range of link relays used for PC(PLC) link	0	0 to 64 words
	41	Range of link data registers used for PC(PLC) link	0	0 to 128 words
	42	Starting word number for link relay transmission	0	0 to 63
	43	Link relay transmission size	0	0 to 64 words
	44	Starting number for link data register tranmission	0	0 to 127
	45	Link data register transmission size	0	0 to 127 words
	46	PC(PLC) link switch flag	Normal (32k only)	Normal/reverse
	47	Maximum unit number setting for MEWNET-W0 PC(PLC) link	16	1 to 16
PC (PLC) link 1 set-ting (32k only)	**50**	Range of link relays used for PC(PLC) link	0	0 to 64 words
	51	Range of link data registers used for PC(PLC) link	0	0 to 128 words
	52	Starting word number for link relay transmission	64	64 to 127
	53	Link relay transmission size	0	0 to 64 words
	54	Starting number for link data register tranmission	128	128 to 255
	55	Link data register transmission size	0	0 to 127 words
	57	Maximum unit number setting for MEWNET-W0 PC(PLC) link	16	1 to 16

	No.	Name	Default value	Descriptions	
High-speed counter	400	**High-speed counter operation mode settings (X0 to X2)**	CH0: Do not set input X0 as high-speed counter	CH0	Do not set input X0 as high-speed counter. Two-phase input (X0, X1) Two-phase input (X0, X1), Reset input (X2) Incremental input (X0) Incremental input (X0), Reset input (X2) Decremental input (X0) Decremental input (X0), Reset input (X2) incremental/decremental input (X0, X1) incremental/decremental input (X0, X1), Reset input (X2) Incremental/decremental control input (X0, X1) Incremental/decremental control input (X0, X1), Reset input (X2)
			CH1: Do not set input X1 as high-speed counter	CH1	Do not set input X1 as high-speed counter. Incremental input (X1) Incremental inptu (X1), Reset input (X2) Decremental input (X1) Decremental input (X1), Reset input (X2)
	401	**High-speed counter operation mode settings (X3 to X5)**	CH2: Do not set input X3 as high-speed counter	CH2	Do not set input X3 as high-speed counter. Two-phase input (X3, X4) Two-phase input (X3, X4), Reset input (X5) Incremental input (X3) Incremental input (X3), Reset input (X5) Decremental input (X5) Decremental input (X5), Reset input (X5) Incremental/decremental input (X3, X4) Incremental/decremental input (X3, X4), Reset input (X5) Incremental/decremental control (X3, X4) Incremental/decremental control (X3, X4), Reset input (X5)
			HC3: Does not set input X4 as high-speed counter	CH3	Does not set input X4 as high-speed counter. Incremental input (X4) Incremental input (X4), Reset input (X5) Decremental input (X4) Decremental input (X4), Reset input (X5)

<table>
<tr><th></th><th>No.</th><th>Name</th><th>Default value</th><th>Descriptions</th></tr>
<tr><td rowspan="2">Inter-rupt-input</td><td>402</td><td>Pulse catch input settings</td><td>Not set</td><td>X0 X1 X2 X3 X4 X5 X6 X7
Specify the input contacts used as pulse catch input.</td></tr>
<tr><td>403</td><td>Interrupt input settings</td><td>Not set</td><td>X0 X1 X2 X3 X4 X5 X6 X7
Specify the input contacts used as intrrupt input.
X0 X1 X2 X3 X4 X5 X6 X7
Specify the effective interrupt edge.
(When set: ON→OFF is valid)</td></tr>
</table>

Note1) If the operation mode is set to Two-phase, incremental/decremental, or incremental/decremental control, the setting for CH1 is invalid in part 2 of system register 400 and the setting for CH3 is invalid in part2 of system register 401.

Note2) If reset input settings overlap, the CH1 setting takes precedence in system register 400 and the CH3 setting takes precedence in system register 401.

Note3) The settings for pulse catch and interrupt input can only be specified in system registers 402 and 403.

Note4) If system register 400 to 403 have been set simultaneously for the same input relay,the follwing precedence order is effective: [High-speed counter]→[Pulse catch]→[Interrupt input].

<Example>

When the high-speed counter is being used in the addition input mode, even if input X0 is specified as an interrupt input or as pulse catch input, those settings are invalid, and X0 functions as counter input for the high-speed counter.

	No.	Name	Default value	Descriptions
Tool port setting	410	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose communications
		Selection of modem connection	Disabled	Enabled/Disabled
	413	**Communication format setting**	Data lenght bit: 8 bits Parity check: “with odd” Stop bit: 1 bit	Enter the settings for the various items. - Data lenght bit: 7 bits/8 bits - Parity check: none/with odd/with even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator CR/CR+LF/None - Header: STX not exist/STX exist
	415	**Communication speed (Baud rate) setting**	9600 bps	2400 bps / 4800 bps / 9600 bps / 19200 bps / 38400 bps / 57600 bps / 115200 bps
	420	**Starting address for received buffer of general (serial data) communication mode**	0	0 to 32764
	421	**Buffer capacity setting for data received of general (serial data) communication mode**	0	0 to 2048
COM 1 port setting	410	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose serial communication PC(PLC) link MODBUS RTU
		Selection of modem connection	Disabled	Enabled/Disabled
	413	**Communication format setting**	Data lenght bit: 8 bits Parity check: Odd Stop bit: 1 bit	Enter the settings for the various items. - Data lenght bit: 7 bits/8 bits - Parity check: none/with odd/with even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator CR/CR+LF/None - Header: STX not exist/STX exist
	415	**Communication speed (Baud rate) setting**	9600 bps	2400 bps / 4800 bps / 9600 bps / 19200 bps / 38400 bps / 57600 bps / 115200 bps
	416	**Starting address for received buffer of general (serial data) communication mode**	0	0 to 32764
	417	**Buffer capacity setting for data received of general (serial data) communication mode**	2048	0 to 2048

Note) The communication format in a PLC link is fixed at the following settings:
Data length is 8 bits, odd parity, stop bit is 1.
The communication speed (baud rate) is fixed at 115200 bps.
The transmission speed of the RS485 port (COM1) of AFPG806 must be identically set by the system register and the dip switch in the communication cassette.

	No.	Name	Default value	Descriptions
COM 2 port set-ting	**411**	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose serial communication MODBUS RTU
		Selection of modem connection	Disabled	Enabled/Disabled
	414	**Communication format setting**	Data lenght bit: 8 bits Parity check: “with odd” Stop bit: 1 bit	Enter the settings for the various items. - Data lenght bit: 7 bits/8 bits - Parity check: none/odd/even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator: CR/CR+LF/None - Header: STX not exist/STX exist
	415	**Communication speed (Baud rate) setting**	9600 bps	2400 bps 4800 bps 9600 bps 19200 bps 38400 bps 57600 bps 115200 bps
	416	**Starting address for received buffer of general (serial data) communication mode**	2048	0 to 32764
	417	**Buffer capacity setting for data received of general (serial data) communication mode**	2048	0 to 2048

Note) The communication format in a PLC link is fixed at the following settings:
the data length is 8 bits, odd parity, stop bit is 1.
The communication speed (baud rate) is fixed at 115200 bps.
The transmission speed of the RS485 port (COM1) of AFPG806 must be identically set by the system register and the dip switch in the communication cassette.

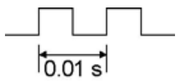
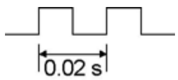
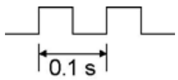
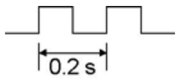
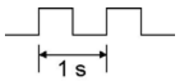
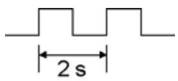
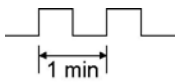
5.1.11 Table of Special Internal Relays for FP Σ

The special internal relays turn on and off under special conditions. The on and off states are not output externally. Writing is not possible with a programming tool or an instruction.

WR900

Relay No.	Name	Description
R9000	**Self-diagnostic error flag**	Turns on when a self-diagnostic error occurs. ⇒ The content of self-diagnostic error is stored in DT90000.
R9001	**Not used**	
R9002	**Not used**	
R9003	**Not used**	
R9004	**I/O verification error flag**	Turns on when an I/O verification error occurs.
R9005	**Backup battery error flag (non-hold)**	Turns on when an backup battery error occurs.
R9006	**Backup battery error flag (hold)**	Turns on when a backup battery error occurs. Once a battery error has been detected, this is held even after recovery has been made. It goes off if the power supply is turned off, or if the system is initialized.
R9007	**Operation error flag (hold)**	Turns on and keeps the on state shen an operation error occurs. ⇒The address where the error occurred is stored in DT90017. (indicates the first operation error which occurred).
R9008	**Operation error flag (non-hold)**	Turns on for an instant when an operation error occurs. ⇒The address where the operation error occurred is stored in DT90018. The contents change each time a new error occurs.
R9009	**Carry flag**	This is set if an overflow or underflow occurs in the calculation results, and as a result of a shift system instruction being executed.
R900A	**> Flag**	Turns on for an instant when the compared results become larger in the comparison instructions.
R900B	**= Flag**	Turns on for an instant, - when the compared results are equal in the comparison instructions. - when the calculated results become 0 in the arithmetic instructions.
R900C	**< Flag**	Turns on for an instant when the compared results become smaller in the comparison instructions.
R900D	**Auxiliary timer instruction flag**	Turns on when the set time elapses (set value reaches 0) in the timing operation of the F137(STMR)/F183(DSTM) auxiliary timer instruction. The flag turns off when the trigger for auxiliary timer instruction turns off.
R900E	**Tool port communication error**	Turns on when communication error at tool port is occurred.
R900F	**Constant scan error flag**	Turns on when scan time exceeds the time specified in system register 34 during constant scan execution. This goes on if 0 has been set using system register 34.

WR901

Relay No.	Name	Description
R9010	**Always on relay**	Always on.
R9011	**Always off relay**	Always off.
R9012	**Scan pulse relay**	Turns on and off alternately at each scan.
R9013	**Initial (on type) pulse relay**	Goes on for only the first scan after operation (RUN) has been started, and goes off for the second and subsequent scans.
R9014	**Initial (off type) pulse relay**	Goes off for only the first scan after operation (RUN) has been started, and goes on for the second and subsequent scans.
R9015	**Step ladder initial pulse relay (on type)**	Turns on for only the first scan of a process after the boot at the step ladder control.
R9016	**Not used**	**-**
R9017	**Not used**	-
R9018	**0.01 s clock pulse relay**	Repeats on/off operations in 0.01 sec. cycles. 
R9019	**0.02 s clock pulse relay**	Repeats on/off operations in 0.02 s. cycles. 
R901A	**0.1 s clock pulse relay**	Repeats on/off operations in 0.1 s. cycles. 
R901B	**0.2 s clock pulse relay**	Repeats on/off operations in 0.2 s. cycles. 
R901C	**1 s clock pulse relay**	Repeats on/off operations in 1 s. cycles. 
R901D	**2 s clock pulse relay**	Repeats on/off operations in 2 s. cycles. 
R901E	**1 min clock pulse relay**	Repeats on/off operations in 1 min. cycles. 
R901F	**Not used**	-

WR902

Relay No.	Name	Description
R9020	**RUN mode flag**	Turns off while the mode selector is set to PROG. Turns on while the mode selector is set to RUN.
R9021	**Not used**	
R9022	**Not used**	
R9023	**Not used**	
R9024	**Not used**	
R9025	**Not used**	
R9026	**Message flag**	Turns on while the F149 (MSG) instruction is executed.
R9027	**Not used**	
R9028	**Not used**	
R9029	**Forcing flag**	Turns on during forced on/off operation for input/output relay timer/counter contacts.
R902A	**Interrupt enable flag**	Turns on while the external interrupt trigger is enabled by the ICTL instruction.
R902B	**Interrupt error flag**	Turns on when an interrupt error occurs.
R902C	**Sample point flag** Note)	Sampling by the instruction=0 Sampling at constant time intervals=1
R902D	**Sample trace end flag** Note)	When the sampling operation stops=1, When the sampling operation starts=0
R902E	**Sampling stop trigger flag** Note)	When the sampling stop trigger activates=1 When the sampling stop trigger stops=0
R902F	**Sampling enable flag** Note)	When sampling starts=1 When sampling stops=0

Note) Available for the 32k type only.

WR903

Relay No.	Name		Description
R9030	**Not used**		-
R9031	**Not used**		-
R9032	**COM1 port communication mode flag**		- Turns on when the general-purpose communication function is being used - Goes off when the MEWTOCOL-COM or the PLC link function is being used.
R9033	**Print instruction execution flag**		Off: Printing is not executed. On: Execution is in progress.
R9034	**RUN overwrite complete flag**		Goes on for ony the first scan following completion of a rewrite during the RUN operation.
R9035	**Not used**		-
R9036	**Not used**		-
R9037	**COM1 port communication error flag**		- Goes on is a transmission error occurs during data communication. - Goes off when a request is made to send data, using the F159 (MTRN) instruction.
R9038	**COM1 port reception done flag during general purpose communication**		- Turns on when the terminator is received during general - purpose serial communication.
R9039	**COM1 port transmission done flag during general-purpose serial communication**		- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose serial communication.
R903A	**High-speed counter control flag**	**ch0**	Turn on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F171(SPDH) to F176(PWMH) are executed.
R903B	**High-speed counter control flag**	**ch1**	Turn on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F171(SPDH) to F176(PWMH) are executed.
R903C	**High-speed counter control flag**	**ch2**	Turn on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F171(SPDH) to F176(PWMH) are executed.
R903D	**High-speed counter control flag**	**ch3**	Turn on while the high-speed counter instructions F166(HC15), F167(HC1R) and the pulse output instructions F171(SPDH) to F176(PWMH) are executed.
R903E	**TOOL port reception done flag during general purpose communication**		- Turns on when the terminator is received during general - purpose serial communication.
R903F	**TOOL port transmission done flag during general-purpose serial communication**		- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose serial communication.

Note) R9030 to R9030F can be changed during 1 scan.

WR904

Relay No.	Name	Description
R9040	**TOOL port operation mode flag**	- Turns on when the general-purpose communication function is being used - Goes off when the computer link function is being used.
R9041	**COM1 port PLC link flag**	Turn on while the PLC link function is used.
R9042	**COM2 port communication mode flag**	- Goes on when the general-purpose serial communication is used. - Goes off when the MEWTOCOL is used.
R9043	**Not used**	-
R9044	**COM1 port SEND/RECV instruction execution flag**	Monitors whether the F145 (SEND) or F146 (RECV) instructions can be executed or not. Off: None of the above mentioned instructions can be executed. (During executing the instruction) On: One of the above mentioned instructions can be executed.
R9045	**COM1 port SEND/RECV instruction execution end flag**	Monitors if an abnormality has been detected during the execution of the F145 (SEND) or F146 (RECV) instructions as follows: Off: No abonormality detected. On: An abnormality detected. (communication error) The error code is stored in DT90039. End code: DT90124
R9046	**Not used**	-
R9047	**COM2 port communication error flag**	- Goes on if a transmission error occurs during data communication. - Goes off when a request is made to send data, using the F159 (MTRN) instruction.
R9048	**COM2 port port reception done flag during general-purpose communicating**	- Turn on when the terminator is received during general-purpose serial communication.
R9049	**COM2 port transmission done flag during general-purpose communication**	- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose communication.
R904A	**COM2 port SEND/RECV instruction execution flag**	Monitors whether the F145 (SEND) or F146 (RECV) instructions can be executed or not. Off: None of the above mentioned instructions can be executed. (During executing the instruction) On: One of the above mentioned instructions can be executed.
R904B	**COM2 port SEND/RECV instruction execution end flag**	Monitors if an abnormality has been detected during the execution of the F145 (SEND) or F146 (RECV) instructions as follows: Off: No abonormality detected. On: An abnormality detected. (communication error) The error code is stored in DT90039. End code: DT90125
R904C to R904D	**Not used**	-
R904E	**Circular interpolation control flag**	Goes on when the F176 (SPCH) circular interpolation instruction is executed.
R904F	**Circular interpolation data overwrite confirmation flag**	It is used to overwrite next data when the circular interpolation instruction is used in the continuation mode.

Note) R9040 to R904F can be changed during 1 scan.

WR905

Relay No.	Name	Description
R9050	**MEWNET-W0 PLC link transmission error flag**	When using MEWNET-W0 - Turns on when a transmission error occurs at PLC link. - Turns on when there is an error in the PLC link area settings.
R9051 to R905F	**Not used**	

WR906

Relay No.	Name		Description
R9060	**MEWNET-W0 PC(PLC) link 0 transmission assurance relay**	**Unit No.1**	Turns on when Unit No. 1 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9061		**Unit No.2**	Turns on when Unit No. 2 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9062		**Unit No.3**	Turns on when Unit No. 3 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9063		**Unit No.4**	Turns on when Unit No. 4 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9064		**Unit No.5**	Turns on when Unit No. 5 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9065		**Unit No.6**	Turns on when Unit No. 6 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9066		**Unit No.7**	Turns on when Unit No. 7 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9067		**Unit No.8**	Turns on when Unit No. 8 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9068		**Unit No.9**	Turns on when Unit No. 9 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R9069		**Unit No.10**	Turns on when Unit No. 10 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906A		**Unit No.11**	Turns on when Unit No. 11 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906B		**Unit No.12**	Turns on when Unit No. 12 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906C		**Unit No.13**	Turns on when Unit No. 13 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906D		**Unit No.14**	Turns on when Unit No. 14 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906E		**Unit No.15**	Turns on when Unit No. 15 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.
R906F		**Unit No.16**	Turns on when Unit No. 16 is communicating properly in PC(PLC) link 0 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 0 mode.

WR907

Relay No.	Name		Description
R9070	MEWNET-W0 PC(PLC) link 0 operation mode relay	Unit No.1	Turns on when Unit No. 1 is in the RUN mode. Turns off when Unit No. 1 is in the PROG. mode.
R9071		Unit No.2	Turns on when Unit No. 2 is in the RUN mode. Turns off when Unit No. 2 is in the PROG. mode.
R9072		Unit No.3	Turns on when Unit No. 3 is in the RUN mode. Turns off when Unit No. 3 is in the PROG. mode.
R9073		Unit No.4	Turns on when Unit No. 4 is in the RUN mode. Turns off when Unit No. 4 is in the PROG. mode.
R9074		Unit No.5	Turns on when Unit No. 5 is in the RUN mode. Turns off when Unit No. 5 is in the PROG. mode.
R9075		Unit No.6	Turns on when Unit No. 6 is in the RUN mode. Turns off when Unit No. 6 is in the PROG. mode.
R9076		Unit No.7	Turns on when Unit No. 7 is in the RUN mode. Turns off when Unit No. 7 is in the PROG. mode.
R9077		Unit No.8	Turns on when Unit No. 8 is in the RUN mode. Turns off when Unit No. 8 is in the PROG. mode.
R9078		Unit No.9	Turns on when Unit No. 9 is in the RUN mode. Turns off when Unit No. 9 is in the PROG. mode.
R9079		Unit No.10	Turns on when Unit No. 10 is in the RUN mode. Turns off when Unit No. 10 is in the PROG. mode.
R907A		Unit No.11	Turns on when Unit No. 11 is in the RUN mode. Turns off when Unit No. 11 is in the PROG. mode.
R907B		Unit No.12	Turns on when Unit No. 12 is in the RUN mode. Turns off when Unit No. 12 is in the PROG. mode.
R907C		Unit No.13	Turns on when Unit No. 13 is in the RUN mode. Turns off when Unit No. 13 is in the PROG. mode.
R907D		Unit No.14	Turns on when Unit No. 14 is in the RUN mode. Turns off when Unit No. 14 is in the PROG. mode.
R907E		Unit No.15	Turns on when Unit No. 15 is in the RUN mode. Turns off when Unit No. 15 is in the PROG. mode.
R907F		Unit No.16	Turns on when Unit No. 16 is in the RUN mode. Turns off when Unit No. 16 is in the PROG. mode.

WR908

Relay No.	Name		Description
R9080	**MEWNET-W0 PC(PLC) link 1 transmission assurance relay (32k only)**	**Unit No.1**	Turns on when Unit No. 1 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9081		**Unit No.2**	Turns on when Unit No. 2 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9082		**Unit No.3**	Turns on when Unit No. 3 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9083		**Unit No.4**	Turns on when Unit No. 4 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9084		**Unit No.5**	Turns on when Unit No. 5 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9085		**Unit No.6**	Turns on when Unit No. 6 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9086		**Unit No.7**	Turns on when Unit No. 7 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9087		**Unit No.8**	Turns on when Unit No. 8 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R9088		**Unit No.9**	Turns on when Unit No. 9 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9089		**Unit No.10**	Turns on when Unit No. 10 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908A		**Unit No.11**	Turns on when Unit No. 11 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908B		**Unit No.12**	Turns on when Unit No. 12 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908C		**Unit No.13**	Turns on when Unit No. 13 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908D		**Unit No.14**	Turns on when Unit No. 14 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908E		**Unit No.15**	Turns on when Unit No. 15 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.
R908F		**Unit No.16**	Turns on when Unit No. 16 is communicating properly in PC(PLC) link 1 mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link 1 mode.

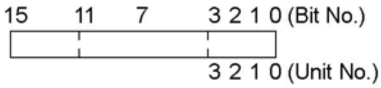
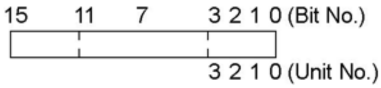
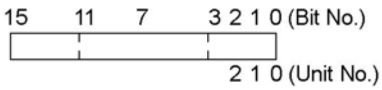
WR909

Relay No.	Name		Description
R9090	**MEWNET-W0 PC(PLC) link 1 operation mode relay (32k only)**	**Unit No.1**	Turns on when Unit No. 1 is in the RUN mode. Turns off when Unit No. 1 is in the PROG. mode.
R9091		**Unit No.2**	Turns on when Unit No. 2 is in the RUN mode. Turns off when Unit No. 2 is in the PROG. mode.
R9092		**Unit No.3**	Turns on when Unit No. 3 is in the RUN mode. Turns off when Unit No. 3 is in the PROG. mode.
R9093		**Unit No.4**	Turns on when Unit No. 4 is in the RUN mode. Turns off when Unit No. 4 is in the PROG. mode.
R9094		**Unit No.5**	Turns on when Unit No. 5 is in the RUN mode. Turns off when Unit No. 5 is in the PROG. mode.
R9095		**Unit No.6**	Turns on when Unit No. 6 is in the RUN mode. Turns off when Unit No. 6 is in the PROG. mode.
R9096		**Unit No.7**	Turns on when Unit No. 7 is in the RUN mode. Turns off when Unit No. 7 is in the PROG. mode.
R9097		**Unit No.8**	Turns on when Unit No. 8 is in the RUN mode. Turns off when Unit No. 8 is in the PROG. mode.
R9098		**Unit No.9**	Turns on when Unit No. 9 is in the RUN mode. Turns off when Unit No. 9 is in the PROG. mode.
R9099		**Unit No.10**	Turns on when Unit No. 10 is in the RUN mode. Turns off when Unit No. 10 is in the PROG. mode.
R909A		**Unit No.11**	Turns on when Unit No. 11 is in the RUN mode. Turns off when Unit No. 11 is in the PROG. mode.
R909B		**Unit No.12**	Turns on when Unit No. 12 is in the RUN mode. Turns off when Unit No. 12 is in the PROG. mode.
R909C		**Unit No.13**	Turns on when Unit No. 13 is in the RUN mode. Turns off when Unit No. 13 is in the PROG. mode.
R909D		**Unit No.14**	Turns on when Unit No. 14 is in the RUN mode. Turns off when Unit No. 14 is in the PROG. mode.
R909E		**Unit No.15**	Turns on when Unit No. 15 is in the RUN mode. Turns off when Unit No. 15 is in the PROG. mode.
R909F		**Unit No.16**	Turns on when Unit No. 16 is in the RUN mode. Turns off when Unit No. 16 is in the PROG. mode.

5.1.12 Table of Special Data Registers for FP Σ

The special data registers are one word (16-bit) memory areas which store specific information.

(A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read -ing	Writ- ing
DT90000	**Self-diagnostic error code**	The self-diagnostic error code is stored here when a self-diagnostic error occurs.	A	N/A
DT90001	**Not used**		N/A	N/A
DT90002	**Position of abnormal I/O unit for FPΣ left side expansion**	When an error occurs at FPΣ expansion I/O unit, the bit corresponding to the unit No. will be set on “1”. Monitor using binary display.  15 11 7 3 2 1 0 (Bit No.) 3 2 1 0 (Unit No.) on "1": error, off "0": normal	A	N/A
DT90003	**Not used**		N/A	N/A
DT90004	**Not used**		N/A	N/A
DT90005	**Not used**		N/A	N/A
DT90006	**Position of abnormal intelligent unit for FPΣ left side expansion**	When an error condition is detected in an intelligent unit, the bit corresponding to the unit No. will turn on . Monitor using binary display.  15 11 7 3 2 1 0 (Bit No.) 3 2 1 0 (Unit No.) on "1": error, off "0": normal	A	N/A
DT90007	**Not used**		N/A	N/A
DT90008	**Not used**		N/A	N/A
DT90009	**Communication error flag for COM2**	Stores the error contents when using COM2 port.	A	N/A
DT90010	**Position of I/O verify error unit for FP0 right side expansion**	When the state of installation of FP0 expansion I/O unit has changed since the power was turned on, the bit corresponding to the unit No. will turn on. Monitor using binary display.  15 11 7 3 2 1 0 (Bit No.) 2 1 0 (Unit No.) on "1": error, off "0": normal	A	N/A

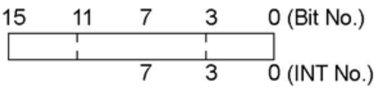
(A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read -ing	Writ-ing
DT90011	**Position of I/O verify error unit for FPΣ left side expansion**	When the state of installation of an FPΣ expansion I/O unit has changed since the power was turned on, the bit corresponding to the unit No. will turn on. Monitor using binary display. 15 11 7 3 2 1 0 (Bit No.) 3 2 1 0 (Unit No.) on "1": error, off "0": normal	A	N/A
DT90012	**Not used**		N/A	N/A
DT90013	**Not used**		N/A	N/A
DT90014	**Operation auxiliary register for data shift instruction**	One shift-out hexadecimal digit is stored in bit positions 0 to 3 when the data shift instruction, F105 (BSR) or F106 (BSL) is executed. The value can be read and written by executing F0 (MV) instruction.	A	A
DT90015	**Operation auxiliary register for division instruction**	The divided remainder (16-bit) is stored in DT90015 when the division instruction F32(%) or F52(B%) instruction is executed. The divided remainder (32-bit) is stored in DT90015 and DT90016 when the division instruction F33(D%) or F53(DB%) is executed. The value can be read and written by executing F0(MV) instruction.	A	A
DT90016			A	A
DT90017	**Operation error address (hold type)**	After commencing operation, the address where the first operation error occurred is stored. Monitor the address using decimal display.	A	N/A
DT90018	**Operation error address (non-hold type)**	The address where an operation error occurred is stored. Each time an error occurs, the new address overwrites the previous address. At the beginning of a scan, the address is 0. Monitor the address using decimal display.	A	N/A
DT90019	**2.5 ms ring counter Note1)**	The data stored here is increased by one every 2.5 ms. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 2.5 ms = Elapsed time between the two points.	A	N/A
DT90020	**10 μs ring counter Note1) Note2)**	The data stored here is increased by one every 10.24 μs. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 10.24 μs = Elapsed time between the two points. Note) The exact value is 10.24 μs.	A	N/A
DT90021	**Not used**		N/A	N/A

Note1) It is renewed once at the beginning of each one scan.

Note2) As DT90020 is renewed even if F0(MV), DT90020 and D instruction is being executed, it can be used to measure the block time.

(A: Available, N/A: Not available)

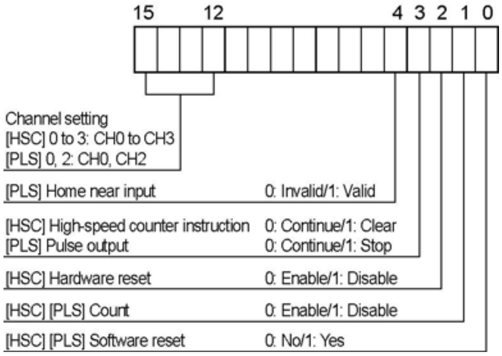
Register No.	Name	Descriptions	Read -ing	Writ- ing
DT90022	**Scan time (current value)** Note)	The current scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.	A	N/A
DT90023	**Scan time (minimum value)** Note)	The minimum scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.	A	N/A
DT90024	**Scan time (maximum value)** Note)	The maximum scan time is stored here. The scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K125 indicates 12.5 ms.	A	N/A
DT90025	**Mask condition monitoring register for interrupts (INT0 to 7)**	The mask conditions of interrupts using the instruction can be stored here. Monitor using binary display.  0: interrupt disabled 1: interrupt enabled	A	N/A
DT90026	**Not used**		N/A	N/A
DT90027	**Periodical interrupt interval (INT24)**	The value set by ICTL instruction is stored. K0: periodical interrupt is not used. K1 to K3000: 0.5ms to 1.5s or 10ms to 30s	A	N/A
DT90028	**Not used**		N/A	N/A
DT90029	**Not used**		N/A	N/A
DT90030	**Message 0**	The contents of the specified message (Data lenght) are stored in these special data registers when F149 (MSG) instruction is executed.	A	N/A
DT90031	**Message 1**			
DT90032	**Message 2**			
DT90033	**Message 3**			
DT90034	**Message 4**			
DT90035	**Message 5**			
DT90036	**Not used**		N/A	N/A

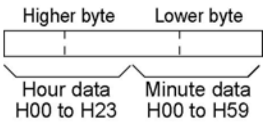
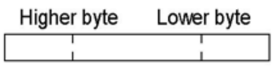
Note) Scan time display is only possible in RUN mode, and shows the operation cycle time. (In PROG. mode, the scan time for the operation is not displayed.) The maximum and minimum values are cleared earh time the mode is switched from RUN to PROG.

(A: Available, N/A: Not available)

Register No.	Name		Descriptions	Read-ing	Writ-ing
DT90037	**Operation auxiliary register for search instruction F96(SRC)**		The number of data that match the searched data is stored here when F96 (SRC) insturction is executed.	A	N/A
DT90038	**Operation auxiliary register for search instruction F96(SRC)**		The position of the first matching data is stored here when an F96 (SRC) instruction is executed.	A	N/A
DT90039	**Not used**			N/A	N/A
DT90040	**Potentiometer (volume) input V0**		The potentiometer value (K0 to K1000) is stored here. This value can be used in analog tiemrs and other applications by using the program to read this value to a data register. V0→DT90040 V1→DT90041	A	N/A
DT90041	**Potentiometer (volume) input V1**				
DT90042			Used by the system.	N/A	N/A
DT90043			Used by the system.	N/A	N/A
DT90044 **DT90045**	**High-speed counter elapsed value**	**For CH0**	The elapsed value (32-bit data) of the high-speed counter is stored here. The value can be read or written by executing F1 (DMV) instruction.	A	A
DT90046 **DT90047**	**High-speed counter target value**	**For CH0**	The targe value (32-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions to be used when the high-speed counter related instruction F166, F167, F171, F175 or F176 is executed. The value can be read by executing F1 (DMV) instruction.	A	N/A
DT90048 **DT90049**	**High-speed counter elapsed value area**	**For CH1**	The elapsed value (32-bit data) of the high-speed counter is stored here. The value can be read and written by executing F1 (DMV) instruction.	A	A
DT90050 **DT90051**	**High-speed counter target value area**	**For CH1**	The target value (32-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions to be used when the high-speed counter related instruction F166 or F167 is executed. The value can be read by executing F1 (DMV) instruction.	A	N/A

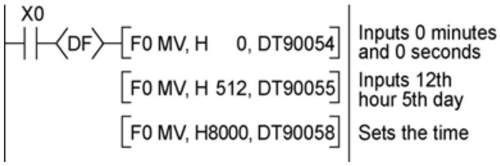
(A: Available, N/A: Not available)

<table>
<tr><th>Register No.</th><th>Name</th><th>Descriptions</th><th>Read -ing</th><th>Writ- ing</th></tr>
<tr><td>DT90052</td><td>High-speed counter and pulse output control flag</td><td>A value can be written with F0 (MV) instruction to reset the high-speed counter, disable counting, continue or clear high-speed counter instruction.
Control code setting


Note) Refer to the “Count for reset input” in “Count 6.3.2 “Input Mode and Count”</td><td>N/A</td><td>A</td></tr>
<tr><td>DT90053</td><td>Real-Time Clock (Clock/Calendar) monitor (hour/minute)</td><td>Hour and minute data of the Real-Time Clock (Clock/Calendar) are stored here. This data is read-only data. It cannot be overwritten.

</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90054</td><td>Real-Time Clock (Clock/Calendar) setting (minute/second)</td><td rowspan="4">The year, month, day, hour, minute, second and day-of-the-week data for the Real-Time Clock(Clock/Calendar) is stored.
The built-in Real-Time Clock(Clock/Calendar) will operate correctly through the year 2099 and supports leap years. The Real-Time Clock (Clock/Calendar) can be set by writing a value using a programming tool software or a program that uses the F0 (MV) instruction.(see example for DT90058)


<table><tr><td>DT90054</td><td>Minute data (H00 to H59)</td><td>Second data (H00 to H59)</td></tr><tr><td>DT90055</td><td>Day data (H01 to H31)</td><td>Hour data (H00 to H23)</td></tr><tr><td>DT90056</td><td>Year data (H00 to H99)</td><td>Month data (H01 to H12)</td></tr><tr><td>DT90057</td><td>–</td><td>Day-of-the-week (H00 to H06)</td></tr></table>As a day of the week is not automatially set on FPWIN GR, fix what day is set to 00, and set each value for 00 to 06.</td><td rowspan="4">A</td><td rowspan="4">A</td></tr>
<tr><td>DT90055</td><td>Real-Time Clock (Clock/Calendar) setting (day/hour)</td></tr>
<tr><td>DT90056</td><td>Real-Time Clock(Clock/Calendar) setting (year/month)</td></tr>
<tr><td>DT90057</td><td>Real-Time Clock (Clock/Calendar) setting (day-of-the-week)</td></tr>
</table>

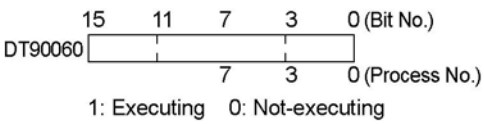
(A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read-ing	Writ-ing
DT90058	**Real-Time Clock (Clock/Calendar) time setting**	The Real-Time Clock(Clock/Calendar) is adjusted as follows. **When setting the Real-Time Clock(Clock/Calendar) by program** By setting the highest bit of DT90058 to 1, the time becomes that written to DT90054 to DT90057 by F0 (MV) instruction. After the time is set, DT90058 is cleared to 0. (Cannot be performed with any instruction other than F0 (MV) instruction.) <Example> Set the time to 12:00:00 on the 5th day when the X0 turns on.  Note) If the values of DT90054 to DT90057 are changed with the programming tool software, the time will be set when the new values are written. Therefore, it is unnecessary to write to DT90058.	A	A
DT90059	**Serial communication error code**	Error code is sotred here when a communication error occurs.	N/A	N/A

(A: Available, N/A: Not available)

<table>
<tr><th>Register No.</th><th>Name</th><th>Descriptions</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90060</td><td>Step ladder process (0 to 15)</td><td rowspan="22">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on.

Monitor using binary display.

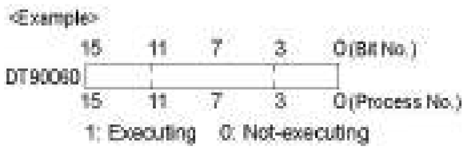
<Example>


A programming tool software can be used to write data.</td><td rowspan="22">A</td><td rowspan="22">A</td></tr>
<tr><td>DT90061</td><td>Step ladder process (16 to 31)</td></tr>
<tr><td>DT90062</td><td>Step ladder process (32 to 47)</td></tr>
<tr><td>DT90063</td><td>Step ladder process (48 to 63)</td></tr>
<tr><td>DT90064</td><td>Step ladder process (64 to 79)</td></tr>
<tr><td>DT90065</td><td>Step ladder process (80 to 95)</td></tr>
<tr><td>DT90066</td><td>Step ladder process (96 to 111)</td></tr>
<tr><td>DT90067</td><td>Step ladder process (112 to 127)</td></tr>
<tr><td>DT90068</td><td>Step ladder process (128 to 143)</td></tr>
<tr><td>DT90069</td><td>Step ladder process (144 to 159)</td></tr>
<tr><td>DT90070</td><td>Step ladder process (160 to 175)</td></tr>
<tr><td>DT90071</td><td>Step ladder process (176 to 191)</td></tr>
<tr><td>DT90072</td><td>Step ladder process (192 to 207)</td></tr>
<tr><td>DT90073</td><td>Step ladder process (208 to 223)</td></tr>
<tr><td>DT90074</td><td>Step ladder process (224 to 239)</td></tr>
<tr><td>DT90075</td><td>Step ladder process (240 to 255)</td></tr>
<tr><td>DT90076</td><td>Step ladder process (256 to 271)</td></tr>
<tr><td>DT90077</td><td>Step ladder process (272 to 287)</td></tr>
<tr><td>DT90078</td><td>Step ladder process (288 to 303)</td></tr>
<tr><td>DT90079</td><td>Step ladder process (304 to 319)</td></tr>
<tr><td>DT90080</td><td>Step ladder process (320 to 335)</td></tr>
<tr><td>DT90081</td><td>Step ladder process (336 to 351)</td></tr>
</table>

(A: Available, N/A: Not available)

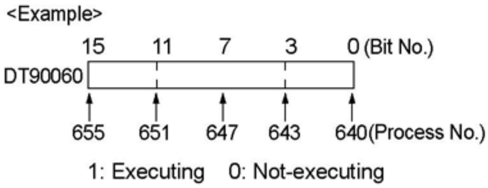
<table>
<tr><th>Register No.</th><th>Name</th><th>Descriptions</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90082</td><td>Step ladder process (352 to 367)</td><td rowspan="16">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on .

Monitor using binary display.



A programming tool software can be used to write data.</td><td rowspan="16">A</td><td rowspan="16">A</td></tr>
<tr><td>DT90083</td><td>Step ladder process (368 to 383)</td></tr>
<tr><td>DT90084</td><td>Step ladder process (384 to 399)</td></tr>
<tr><td>DT90085</td><td>Step ladder process (400 to 415)</td></tr>
<tr><td>DT90086</td><td>Step ladder process (416 to 431)</td></tr>
<tr><td>DT90087</td><td>Step ladder process (432 to 447)</td></tr>
<tr><td>DT90088</td><td>Step ladder process (448 to 463)</td></tr>
<tr><td>DT90089</td><td>Step ladder process (464 to 479)</td></tr>
<tr><td>DT90090</td><td>Step ladder process (480 to 495)</td></tr>
<tr><td>DT90091</td><td>Step ladder process (496 to 511)</td></tr>
<tr><td>DT90092</td><td>Step ladder process (512 to 527)</td></tr>
<tr><td>DT90093</td><td>Step ladder process (528 to 543)</td></tr>
<tr><td>DT90094</td><td>Step ladder process (544 to 559)</td></tr>
<tr><td>DT90095</td><td>Step ladder process (560 to 575)</td></tr>
<tr><td>DT90096</td><td>Step ladder process (576 to 591)</td></tr>
<tr><td>DT90097</td><td>Step ladder process (592 to 607)</td></tr>
</table>

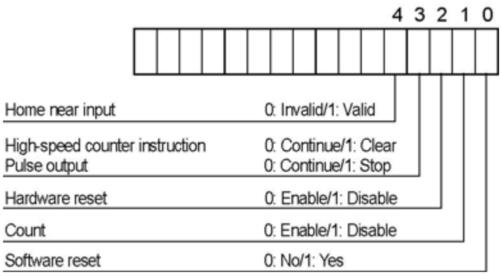
(A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read-ing	Writ-ing
DT90098	**Step ladder process (608 to 623)**	Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on “1”. Monitor using binary display  A programming tool software can be used to write data.	A	A
DT90099	**Step ladder process (624 to 639)**			
DT90100	**Step ladder process (640 to 655)**			
DT90101	**Step ladder process (656 to 671)**			
DT90102	**Step ladder process (672 to 687)**			
DT90103	**Step ladder process (688 to 703)**			
DT90104	**Step ladder process (704 to 719)**			
DT90105	**Step ladder process (720 to 735)**			
DT90106	**Step ladder process (736 to 751)**			
DT90107	**Step ladder process (752 to 767)**			
DT90108	**Step ladder process (768 to 783)**			
DT90109	**Step ladder process (784 to 799)**			
DT90110	**Step ladder process (800 to 815)**			
DT90111	**Step ladder process (816 to 831)**			
DT90112	**Step ladder process (832 to 847)**			
DT90113	**Step ladder process (848 to 863)**			
DT90114	**Step ladder process (864 to 879)**			
DT90115	**Step ladder process (880 to 895)**			
DT90116	**Step ladder process (896 to 911)**			
DT90117	**Step ladder process (912 to 927)**			
DT90118	**Step ladder process (928 to 943)**			
DT90119	**Step ladder process (944 to 959)**			
DT90120	**Step ladder process (960 to 975)**			
DT90121	**Step ladder process (976 to 991)**			
DT90122	**Step ladder process (992 to 999) (higher byte is not used.)**			

(A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read-ing	Writ-ing
DT90123	**Not used**	-	N/A	N/A
DT90124	**COM1 SEND/RECV instruction end code**	For details, refer to Programming Manual (F145 and F146).	N/A	N/A
DT90125	**COM2 SEND/RECV instruction end code**	For details, refer to Programming Manual (F145 and F146).	N/A	N/A
DT90126	**Forced Input/Outptu unit No.**	Used by the system	N/A	N/A
DT90127 to DT90139	**Not used**	-	N/A	N/A
DT90140	**MEWNET-W0 PC(PLC) link 0 status**	The number of times the receiving operation is performed.	A	N/A
DT90141		The current interval between two receiving operations: value in the register x 2.5ms		
DT90142		The minimum inerval between two receiving operations: value in the register x 2.5ms		
DT90143		The maximum interval between two receiving operations: value in the register x 2.5ms		
DT90144		The number of times the sending operation is performed.		
DT90145		The current interval between two sending operations: value in the register x 2.5ms		
DT90146		The minimum interval between two sending operations: value in the register x 2.5ms		
DT90147		The maximum interval between two sending operations: value in the register x 2.5ms		
DT90148	**MEWNET-W0 PC(PLC) link 1 status (32k type only)**	The number of times the receiving operation is performed.	A	N/A
DT90149		The current interval between two receiving operations: value in the register x 2.5ms		
DT90150		The minimum inerval between two receiving operations: value in the register x 2.5ms		
DT90151		The maximum interval between two receiving operations: value in the register x 2.5ms		
DT90152		The number of times the sending operation is performed.		
DT90153		The current interval between two sending operations: value in the register x 2.5ms		
DT90154		The minimum interval between two sending operations: value in the register x 2.5ms		
DT90155		The maximum interval between two sending operations: value in the register x 2.5ms		

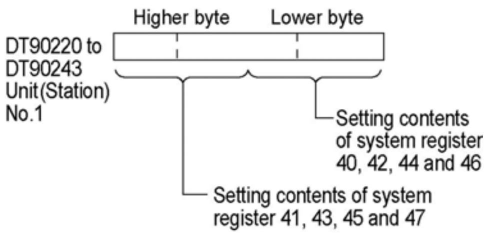
(A: Available, N/A: Not available)

Register No.	Name	Descriptions	Read-ing	Writ-ing
DT90156	**MEWNET-W0 PC(PLC) link 0 status**	Area used for measurement of receiving interval.	A	N/A
DT90157		Area used for measurement of sending interval.		
DT90158	**MEWNET-W0 PC(PLC) link 1 Status (32k type only)**	Area used for measurement of receiving interval.	A	N/A
DT90159		Area used for measurement of sending interval.		
DT90160	**MEWNET-W0 PLC link unit No.**	Stores the unit No. of PLC link	A	N/A
DT90161	**MEWNET-W0 PLC link error flag**	Stores the error contents of PLC link	A	N/A
DT90162 to DT90169	**Not used**	-	N/A	N/A
DT90170	**MEWNET-W0 PLC link status**	Duplicated destination for PLC inter-link address	A	N/A
DT90171		Counts how many times a token is lost.		
DT90172		Counts how many times two or more tokens are detected.		
DT90173		Counts how many times a signal is lost.		
DT90174		No. of times underfined commands have been received.		
DT90175		No. of times sum check errors have occurred during reception.		
DT90176		No. of times format errors have occurred in received data.		
DT90177		No. of times transmission errors have occurred.		
DT90178		No. of times procedural errors have occurred.		
DT90179		No. of times overlapping parent units have occurred.		
DT90180 to DT90189	**Not used**	-	N/A	N/A
DT90190	**High-speed counter control flag monitor for CH0**	This monitors the data specified in DT90052. 	A	N/A
DT90191	**High-speed counter control flag monitor for CH1**			
DT90192	**High-speed counter control flag monitor for CH2**			
DT90193	**High-speed counter control flag monitor for CH3**			

(A: Available, N/A: Not available)

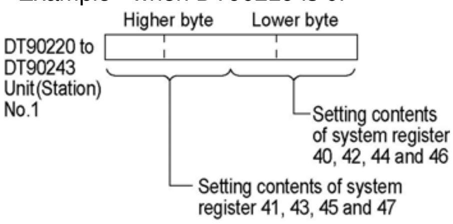
Register No.	Name		Descriptions	Read-ing	Writ-ing
DT90194 to DT90199	**Not used**		-	N/A	N/A
DT90200	**High-speed counter elapsed value**	**For CH2**	The elapsed value (32-bit data) for the high-speed counter is stored here. The value can be read and written by executing the F1 (DMV) instruction.	A	A
DT90201					
DT90202	**High-speed counter target value**	**For CH2**	The targe value (32-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction F166, F167, F171, F175 or F176 is executed. The value can be read by executing F1 (DMV) instruction.	A	N/A
DT90203					
DT90204	**High-speed counter elapsed value**	**For CH3**	The elapsed value (32-bit data) for the high-speed counter is stored here. The value can be read and written by executing F1 (DMV) instruction.	A	A
DT90205					
DT90206	**High-speed counter target value**	**For CH3**	The target value (32-bit data) of the high-speed counter specified by the high-speed counter instruction is stored here. Target values have been preset for the various instructions, to be used when the high-speed counter related instruction F166 or F167 is executed. The value can be read by executing the F1 (DMV) instruction.	A	N/A
DT90207					
DT90208 to DT90218	**Not used**			N/A	N/A

(A: Available, N/A: Not available)

Register No.	Name		Descriptions	Read-ing	Writ-ing
DT90219	**Unit No. (Station No.) selection for DT90220 to DT90251**		0: Unit No. (Station No.) 1 to 8, 1: Unit No. (Station No.) 9 to 16	A	N/A
DT90220	**PLC link Unit (station) No. 1 or 9**	**System register 40 and 41**	The contents of the system register settings partaining to the PLC inter-link function for the various unit numbers are stored as shown below. <Example> When DT90219 is 0 	A	N/A
DT90221		**System register 42 and 43**			
DT90222		**System register 44 and 45**			
DT90223		**System register 46 and 47**			
DT90224	**PLC link Unit (station) No. 2 or 10**	**System register 40 and 41**			
DT90225		**System register 42 and 43**			
DT90226		**System register 44 and 45**			
DT90227		**System register 46 and 47**			
DT90228	**PLC link Unit (station) No. 3 or 11**	**System register 40 and 41**			
DT90229		**System register 42 and 43**			
DT90230		**System register 44 and 45**			
DT90231		**System register 46 and 47**			
DT90232	**PLC link Unit (station) No. 4 or 12**	**System register 40 and 41**			
DT90233		**System register 42 and 43**			
DT90234		**System register 44 and 45**			
DT90235		**System register 46 and 47**			
DT90236	**PLC link Unit (station) No. 5 or 13**	**System register 40 and 41**			
DT90237		**System register 42 and 43**			
DT90238		**System register 44 and 45**			
DT90239		**System register 46 and 47**			

(A: Available, N/A: Not available)

<table>
<tr><th>Register No.</th><th colspan="2">Name</th><th>Descriptions</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90240</td><td rowspan="4">PLC link Unit (station) No. 6 or 14</td><td>System register 40 and 41</td><td rowspan="12">The contents of the system register settings partaining to the PLC inter-link function for the various unit numbers are stored as shown below.

<Example> when DT90219 is 0.

</td><td rowspan="12">A</td><td rowspan="12">N/A</td></tr>
<tr><td>DT90241</td><td>System register 42 and 43</td></tr>
<tr><td>DT90242</td><td>System register 44 and 45</td></tr>
<tr><td>DT90243</td><td>System register 46 and 47</td></tr>
<tr><td>DT90244</td><td rowspan="4">PLC link Unit (station) No. 7 or 15</td><td>System register 40 and 41</td></tr>
<tr><td>DT90245</td><td>System register 42 and 43</td></tr>
<tr><td>DT90246</td><td>System register 44 and 45</td></tr>
<tr><td>DT90247</td><td>System register 46 and 47</td></tr>
<tr><td>DT90248</td><td rowspan="4">PLC link Unit (station) No. 8 or 16</td><td>System register 40 and 41</td></tr>
<tr><td>DT90249</td><td>System register 42 and 43</td></tr>
<tr><td>DT90250</td><td>System register 44 and 45</td></tr>
<tr><td>DT90251</td><td>System register 46 and 47</td></tr>
<tr><td>DT90252</td><td colspan="2">Not used</td><td></td><td rowspan="4">N/A</td><td rowspan="4">N/A</td></tr>
<tr><td>DT90253</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90254</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>D590255</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90256</td><td colspan="2">Unit No. (Station No.) switch monitor for COM port</td><td>Used by the system</td><td>N/A</td><td>N/A</td></tr>
</table>

5.1.13 Table of System Registers for FP-X

Item	Address	Name	Default value	Description	
Hold/ Non-hold 1	**5**	Starting number setting for counter	1008	0 to 1024	• These settings are effective if the optional backup battery is installed. • If no backup battery is used, do not change the default settings. Otherwise proper functioning of hold/non-hold values cannot be guaranteed.
	6	Hold type area starting number setting for timer and counter	1008	0 to 1024	
	7	Hold type area starting number setting for internal relays	248	0 to 256	
	8	Hold type area starting number setting for data registers	C14: 12230 C30, C60: 32710	0 to 32765	
	14	Hold or non-hold setting for step ladder process	Non-hold	Hold/ Non-hold	
	4	Previous value is held for a leading edge detection instruction (DF instruction) with MC	Hold	Hold/ Non-hold	
Hold/ Non-hold 2	**10**	Hold type area starting number for PC(PLC) W0-0 link relays	64	0 to 64	
	11	Hold type area starting number for PC(PLC) W0-1 link relays	128	64 to 128	
	12	Hold type area starting number for PC(PLC) W0-0 link registers	128	0 to 128	
	13	Hold type area starting number for PC(PLC) W0-1 link registers	256	128 to 256	
Action on error	**20**	Disable or enable setting for duplicated output	Disabled	Disabled/Enabled	
	23	Operation setting when an I/O verification error occurs	Stop	Stop/Continuation of operation	
	26	Operation setting when an operation error occurs	Stop	Stop/Continuation of operation	
	4	Alarm battery error (Operating setting when battery error occurs)	Disabled	Dis-abled: When a battery error occurs, a self-diagnostic error is not issued and the ERROR/ALARM LED does not flash. Ena-bled: When a battery error occurs, a self-diagnostic error is issued and the ERROR/ALARM LED flashes.	
Time set-ting	**31**	Wait time setting for multi-frame communication	6500.0 ms	10 to 81900 ms	
	32	Timeout setting for SEND/RECV, RMRD/RMWT commands	10000.0 ms	10 to 81900 ms	
	34	Constant value settings for scan time	Normal scan	0: Normal scan 0 to 350 ms: Scans once each specified time interval	
	36	Expansion unit recognition time	0 (No wait time)	0 to 10 s (0.1 second bit)	

FP-X

Item	Add-ress	Name	Default value	Description
PC (PLC) link W0-0 set-ting	**40**	Range of link relays used for PC(PLC) link	0	0 to 64 words
	41	Range of link data registers used for PC(PLC) link	0	0 to 128 words
	42	Starting number for link relay transmission	0	0 to 63
	43	Link relay transmission size	0	0 to 64 words
	44	Starting number for link data register transmission	0	0 to 127
	45	Link data register transmission size	0	0 to 127 words
	46	PC(PLC) link switch flag	Normal	Normal/reverse
	47	Maximum unit number setting for MEWNET-W0 PC(PLC) link	16	1 to 16
PC (PLC) link W0-1 set-ting	**50**	Range of link relays used for PC(PLC) link	0	0 to 64 words
	51	Range of link data registers used for PC(PLC) link	0	0 to 128 words
	52	Starting number for link relay transmission	64	64 to 127
	53	Link relay transmission size	0	0 to 64 words
	54	Starting number for link data register transmission	128	128 to 255
	55	Link data register transmission size	0	0 to 127 words
	57	Maximum unit number setting for MEWNET-W0 PC(PLC) link	16	1 to 16

FP-X Tr type

Item	Add-ress	Name	Default value	Description
Cont-roller input settings 1 (HSC)	**400**	**High-speed counter settings (X0 to X3)**	CH0: Do not set input X0 as high-speed counter	Do not set input X0 as high-speed counter. Incremental input (X0) Decremental input (X0) Two-phase input (X0, X1) Individual input (X0, X1) Incremental/decremental control input (X0, X1)
			CH1: Do not set input X1 as high-speed counter	Do not set input X1 as high-speed counter. Incremental input (X1) Decremental input (X1)
			CH2: Do not set input X2 as high-speed counter	Do not set input X2 as high-speed counter. Incremental input (X2) Decremental input (X2) Two-phase input (X2, X3) Individual input (X2, X3) Incremental/decremental control input (X2, X3)
			CH3: Do not set input X3 as high-speed counter	Do not set input X3 as high-speed counter. Incremental input (X3) Decremental input (X3)
Cont-roller input settings 2 (HSC/ PLS)	**401**	**High-speed counter/ pulse output settings (X4 to X7)**	CH4: Do not set input X4 as high-speed counter	Do not set input X4 as high-speed counter. Incremental input (X4) Decremental input (X4) Two-phase input (X4, X5) Individual input (X4, X5) Incremental/decremental control input (X4, X5)
			X4: Normal input	Normal input Home input of pulse output CH0
			CH5: Do not set input X5 as high-speed counter	Do not set input X5 as high-speed counter. Incremental input (X5) Decremental input (X5)
			X5: Normal input	Normal input Home input of pulse output CH1
			CH6: Do not set input X6 as high-speed counter	Do not set input X6 as high-speed counter. Incremental input (X6) Decremental input (X6) Two-phase input (X6, X7) Individual input (X6, X7) Incremental/decremental control input (X6, X7)
			X6: Normal input	Normal input Home input of pulse output CH2 Reset input of high-speed counter CH0
			CH7: Do not set input X7 as high-speed counter	Do not set input X7 as high-speed counter. Incremental input (X7) Decremental input (X7)
			X7: Normal input	Normal input Home input of pulse output CH3 Reset input of high-speed counter CH2

FP-X Tr type

Item	Add-ress	Name	Default value	Description
Cont-roller output settings (PLS/ PWM)	**402**	**Pulse/ PWM output settings (Y0 to Y7)**	CH0: Normal output	Normal output (Y0, Y1) Pulse output (Y0, Y1) PWM output (Y0), Normal output (Y1)
			CH1: Normal output	Normal output (Y2, Y3) Pulse output (Y2, Y3) PWM output (Y2), Normal output (Y3)
			CH2: Normal output	Normal output (Y4, Y5) Pulse output (Y4, Y5) PWM output (Y4), Normal output (Y5)
			CH3: Normal output	Normal output (Y6, Y7) Pulse output (Y6, Y7) PWM output (Y6), Normal output (Y7)
Inter-rupt/ Pulse catch settings	**403**	**Pulse catch input settings**	Not set	X0 X1 X2 X3 X4 X5 X6 X7 Controller input The pressed contact is set for the pulse catch input.
	404	**Interrupt input settings**	Not set	X0 X1 X2 X3 X4 X5 X6 X7 Controller input The pressed contact is set for the interrupt input.
Inter-rupt edge settings	**405**	**Interrupt edge setting for controller input**	Leading edge	X0 X1 X2 X3 X4 X5 X6 X7 Leading edge X0 X1 X2 X3 X4 X5 X6 X7 Trailing edge The pressed contact is up and set to trailing edge.

Note1) If CH0, CH2, CH4 and CH6 of the high-speed counter is set to the two-phase input, individual input or incremental/decremental control input, the settings of CH1, CH3, CH and CH7 will be invalid.

Note2) Only CH0 and CH2 are available for the reset input of the high-speed counter.
X6 for CH0 and X7 for CH2 can be allocated.

Note3) X4 to X7 can be used as the home input of the pulse output CH0 to CH3.
When using the home return function of the pulse output, always set the home input. In that case, X4 to X7 cannot be set as the high-speed counter.

Note4) When using the pulse output/PWM output, the controller output settings must be specified.
The output that has been set to the pulse output/PWM output cannot be used as the normal output.

Note5) If the same input has been set to the high-speed, pulse catch and interrupt input simultaneously, the following precedence order is effective:
[High-speed counter] → [Pulse catch] → [Interrupt input]

FP-X Ry type

Item	Add-ress	Name	Default value	Description
Pulse I/O cassette settings (HSC/ PLS)	**400**	**High-speed counter settings (X100 to X102)**	CH8: Do not set input X100 as high-speed counter	Do not set input X100 as high-speed counter. Two-phase input (X100, X101) Two-phase input (X100, X101), Reset input (X102) Incremental input (X100) Incremental input (X100), Reset input (X102) Decremental input (X100) Decremental input (X100), Reset input (X102) Incremental/decremental input (X100, X101) Incremental/decremental input (X100, X101), Reset input (X102) Incremental/decremental control input (X100, X101) Incremental/decremental control input (X100, X101), Reset input (X102)
			CH9: Do not set input X101 as high-speed counter	Do not set input X101 as high-speed counter. Incremental input (X101) Incremental input (X101), Reset input (X102) Decremental input (X101) Decremental input (X101), Reset input (X102)
		Pulse output settings (Y100 to Y101)	CH0: Normal output	Normal output (Y100, Y101) Pulse output (Y100, Y101) PWM output (Y100), Normal output (Y101)
	401	**High-speed counter settings (X200 to X202)**	CHA: Do not set input X200 as high-speed counter	Do not set input X200 as high-speed counter. Two-phase input (X200, X201) Two-phase input (X200, X201), Reset input (X202) Incremental input (X200) Incremental input (X200), Reset input (X202) Decremental input (X202) Decremental input (X202), Reset input (X202) Incremental/decremental input (X200, X201) Incremental/decremental input (X200, X201), Reset input (X202) Incremental/decremental control (X200, X201) Incremental/decremental control (X200, X201), Reset input (X202)
			CHB: Do not set input X201 as high-speed counter	Does not set input X201 as high-speed counter. Incremental input (X201) Incremental input (X201), Reset input (X202) Decremental input (X201) Decremental input (X201), Reset input (X202)
		Pulse output settings (Y200 to Y201)	CH1: Normal output	Normal output (Y200, Y201) Pulse output (Y200, Y201) PWM output (Y200), Normal output (Y201)

Note1) If the operation mode is set to Two-phase, incremental/decremental, or incremental/decremental control, the setting for CH9 is invalid in system register 400 and the setting for CHB is invalid in system register 401.

Note2) If reset input settings overlap, the CH9 setting takes precedence in system register 400 and the CHB setting takes precedence in system register 401.

Note3) CHA, CHB and CH1 input signals in system register 401 are the signals when the pulse I/O cassette (AFPX-PLS) is installed in the cassette mounting part 2.

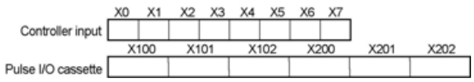
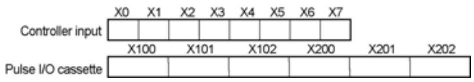
Note4) If the operation mode setting for the pulse output CH0 and CH1 is carried out, it cannot be used as normal output.

When the operation mode for the pulse output CH0 is set to 1, the reset input setting for the high-speed counter CH8 and CH9 is invalid.

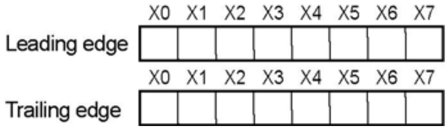
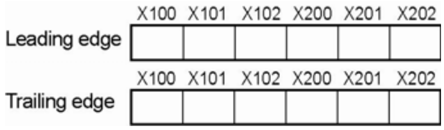
When the operation mode for the pulse output CH1 is set to 1, the reset input setting for the high-speed counter CHA and CHB is invalid.

Note5) Upgrade FPWIN GR to Ver2.6 or higher version if the No. of I/O allocation is indicated with 1-digit number such as X0 in the setting window No. 400 and 401 of FPWIN GR.

FP-X Ry type

Item	Add-ress	Name	Default value	Description
Cont-roller input settings (HSC)	**402**	**High-speed counter settings (X0 to X7)**	CH0: Do not set input X0 as high-speed counter	Do not set input X0 as high-speed counter. Incremental input (X0) Decremental input (X0) Two-phase input (X0, X1)
			CH1: Do not set input X1 as high-speed counter	Do not set input X1 as high-speed counter. Incremental input (X1) Decremental input (X1) Two-phase input (X0, X1)
			CH2: Do not set input X2 as high-speed counter	Do not set input X2 as high-speed counter. Incremental input (X2) Decremental input (X2) Two-phase input (X2, X3)
			CH3: Do not set input X3 as high-speed counter	Do not set input X3 as high-speed counter. Incremental input (X3) Decremental input (X3) Two-phase input (X2, X3)
			CH4: Do not set input X4 as high-speed counter	Do not set input X4 as high-speed counter. Incremental input (X4) Decremental input (X4) Two-phase input (X3 X4)
			CH5: Do not set input X5 as high-speed counter	Do not set input X5 as high-speed counter. Incremental input (X5) Decremental input (X5) Two-phase input (X4, X5)
			CH6: Do not set input X6 as high-speed counter	Do not set input X6 as high-speed counter. Incremental input (X6) Decremental input (X6) Two-phase input (X5, X6)
			CH7: Do not set input X7 as high-speed counter	Do not set input X7 as high-speed counter. Incremental input (X7) Decremental input (X7) Two-phase input (X6, X7)
Inter-rupt/ pulse catch settings	**403**	**Pulse catch input settings**	Not set	 The pressed contact is set for the pulse catch input.
	404	**Interrupt input settings**	Not set	 The pressed contact is set for the interrupt input.

FP-X Ry type

Item	Add-ress	Name	Default value	Description
Inter-rupt edge settings	**405**	**Interrupt edge setting for controller input**	Leading edge	 The pressed contact is up and set to trailing edge.
	406	**Interrupt edge setting for pulse I/O cassette**	Leading edge	 The pressed contact is up and set to trailing edge.

Note1) For counting two-phase input, only CH0, CH2, CH4 and CH6 can be used.
When two-phase input is specified for CH0, CH2, CH4 and CH6, the settings for CH1, CH3, CH5 and CH7 corresponding to each CH No. are ignored, however, specify the same setting for those channels.

Note2) The settings for pulse catch and interrupt input can only be specified in system registers 403 and 404.

Note3) If system register 400 to 404 have been set simultaneously for the same input relay, the following precedence order is effective: [High-speed counter]→[Pulse catch]→[Interrupt input].
<Example>
When the high-speed counter is being used in the addition input mode, even if input X0 is specified as an interrupt input or as pulse catch input, those settings are invalid, and X0 functions as counter input for the high-speed counter.

Note4) Upgrade FPWIN GR to Ver2.6 or higher version if the No. of I/O allocation is indicated with 1-digit number such as X0 in the setting window No. 403,404 and 406 of FPWIN GR.

FP-X

Item	Address	Name	Default value	Description
Tool port setting	**410**	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose communications
		Selection of modem connection	Disabled	Enabled/Disabled
	413	**Communication format setting**	Data length bit: 8 bits Parity check: “with odd” Stop bit: 1 bit	Enter the settings for the various items. - Data length bit: 7 bits/8 bits - Parity check: none/with odd/with even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator: CR/CR+LF/None - Header: STX not exist/STX exist
	415	**Baud rate setting**	9600 bps	2400 bps 4800 bps 9600 bps 19200 bps 38400 bps 57600 bps 115200 bps
	420	**Starting address for received buffer of general (serial data) communication mode**	0	0 to 32764
	421	**Buffer capacity setting for data received of general (serial data) communication mode**	2048	0 to 2048

FP-X

Item	Add-ress	Name	Default value	Description
COM. 1 port set-ting	410	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose serial communication PC(PLC) link MODBUS RTU
		Selection of modem connection	Disabled	Enabled/Disabled
	413	**Communication format setting**	Data length bit: 8 bits Parity check: Odd Stop bit: 1 bit	Enter the settings for the various items. - Data length bit: 7 bits/8 bits - Parity check: none/with odd/with even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to “General-purpose serial communication”. - Terminator: CR/CR+LF/None - Header: STX not exist/STX exist
	415	**Baud rate setting**	9600 bps	2400 bps 4800 bps 9600 bps 19200 bps 38400 bps 57600 bps 115200 bps
	416	**Starting address for received buffer of general (serial data) communication mode**	0	0 to 32764
	417	**Buffer capacity setting for data received of general (serial data) communication mode**	2048	0 to 2048

Note1) The communication format in a PC(PLC) link is fixed at the following settings:
Data length is 8 bits, odd parity, stop bit is 1.
The communication speed (baud rate) is fixed at 115200 bps.

Note2) **Reference:** For information on MODBUS RTU mode operation, <MODBUS RUT Specifications>.

FP-X

Item	Add-ress	Name	Default value	Description
COM. 2 port set-ting	**411**	**Unit No. setting**	1	1 to 99
	412	**Communication mode setting**	Computer link	Computer link General-purpose serial communication MODBUS RTU
		Selection of modem connection	Disabled	Enabled/Disabled
		Selection of port	Built-in USB	Built-in USB Communication cassette
	414	**Communication format setting**	Data length bit: 8 bits Parity check: "with odd" Stop bit: 1 bit	Enter the settings for the various items. - Data length bit: 7 bits/8 bits - Parity check: none/odd/even - Stop bit: 1 bit/2 bits - The following setting is valid only when the communication mode specified by system register 412 has been set to "General-purpose serial communication". - Terminator: CR/CR+LF/None - Header: STX not exist/STX exist
	415	**Baud rate setting**	9600 bps	2400 bps 4800 bps 9600 bps 19200 bps 38400 bps 57600 bps 115200 bps
	416	**Starting address for received buffer of general (serial data) communication mode**	2048	0 to 32764
	417	**Buffer capacity setting for data received of general (serial data) communication mode**	2048	0 to 2048

Note1) The communication format in a PC(PLC) link is fixed at the following settings:
the data length is 8 bits, odd parity, stop bit is 1.
The communication speed (baud rate) is fixed at 115200 bps.

Note2) The USB port for C30 and C60 can be selected by the system register setting.
The USB port has been selected for the COM2 port of C30 and C60 as default setting. The USB port is 115.2 kbps despite of the baud rate setting No. 415.
The setting for No. 412 must be changed to communication cassette for using the COM2 port of the communication cassette.
The COM2 port of the USB port and the communication cassette cannot be used at the same time.

FP-X

Item	Address	Name	Default value	Description
Controller input time constant settings (Note1)	**430**	**Controller input time constant setting 1 X0 to X3**	None	None 1 ms 2 ms 4 ms 8 ms 16 ms 32 ms 64 ms 128 ms 156 ms
	431	**Controller input time constant setting 1 X4 to X7**		
	432	**Controller input time constant setting 2 X8 to XB**		
	433	**Controller input time constant setting 2 XC to XF**		
	434	**Controller input time constant setting 3 X10 to X13**		
	435	**Controller input time constant setting 3 X14 to X17**		
	436	**Controller input time constant setting 4 X18 to X1B**		
	437	**Controller input time constant setting 4 X1C to X1F**		

Note1) These settings are available for the FP-X V2.0 or later.

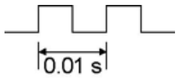
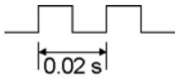
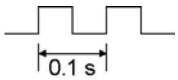
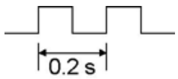
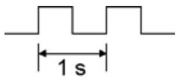
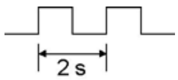
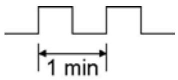
5.1.14 Table of Special Internal Relays for FP-X

The special internal relays turn on and off under special conditions. The on and off states are not output externally. Writing is not possible with a programming tool or an instruction.

WR900 FP-X

Address	Name	Description
R9000	**Self-diagnostic error flag**	Turns on when a self-diagnostic error occurs. ⇒ The content of self-diagnostic error is stored in DT90000.
R9001	**Not used**	-
R9002	**Application cassette I/O error flag**	Turns on when an error is detected in the I/O type application cassette.
R9003	**Application cassette abnormal error flag**	Turns on when an error is detected in the application cassette.
R9004	**I/O verification error flag**	Turns on when an I/O verification error occurs.
R9005	**Backup battery error flag (non-hold)**	Turns on when a backup battery error occurs. Turns on when the battery has run out even if the system register No. 4 has been set not to inform the battery error.
R9006	**Backup battery error flag (hold)**	Turns on when a backup battery error occurs. Turns on when the battery has run out even if the system register No. 4 has been set not to inform the battery error. Once a battery error has been detected, this is held even after recovery has been made. ⇒It goes off if the power supply is turned off, or if the system is initialized.
R9007	**Operation error flag (hold)**	Turns on and keeps the on state when an operation error occurs. ⇒The address where the error occurred is stored in DT90017. (indicates the first operation error which occurred).
R9008	**Operation error flag (non-hold)**	Turns on for an instant when an operation error occurs. ⇒The address where the operation error occurred is stored in DT90018. The contents change each time a new error occurs.
R9009	**Carry flag**	This is set if an overflow or underflow occurs in the calculation results, and as a result of a shift system instruction being executed.
R900A	**> Flag**	Turns on for an instant when the compared results become larger in the comparison instructions.
R900B	**= Flag**	Turns on for an instant, - when the compared results are equal in the comparison instructions. - when the calculated results become 0 in the arithmetic instructions.
R900C	**< Flag**	Turns on for an instant when the compared results become smaller in the comparison instructions.
R900D	**Auxiliary timer Contact**	Turns on when the set time elapses (set value reaches 0) in the timing operation of the F137(STMR)/F183(DSTM) auxiliary timer instruction. The flag turns off when the trigger for auxiliary timer instruction turns off.
R900E	**Tool port communication error**	Turns on when communication error at tool port is occurred.
R900F	**Constant scan error flag**	Turns on when scan time exceeds the time specified in system register 34 during constant scan execution. This goes on if 0 has been set using system register 34.

WR901 FP-X

Address	Name	Description
R9010	**Always on relay**	Always on.
R9011	**Always off relay**	Always off.
R9012	**Scan pulse relay**	Turns on and off alternately at each scan.
R9013	**Initial (on type) pulse relay**	Goes on for only the first scan after operation (RUN) has been started, and goes off for the second and subsequent scans.
R9014	**Initial (off type) pulse relay**	Goes off for only the first scan after operation (RUN) has been started, and goes on for the second and subsequent scans.
R9015	**Step ladder initial pulse relay (on type)**	Turns on for only the first scan of a process after the boot at the step ladder control.
R9016	**Not used**	**-**
R9017	**Not used**	-
R9018	**0.01 s clock pulse relay**	Repeats on/off operations in 0.01 sec. cycles. 
R9019	**0.02 s clock pulse relay**	Repeats on/off operations in 0.02 s. cycles. 
R901A	**0.1 s clock pulse relay**	Repeats on/off operations in 0.1 s. cycles. 
R901B	**0.2 s clock pulse relay**	Repeats on/off operations in 0.2 s. cycles. 
R901C	**1 s clock pulse relay**	Repeats on/off operations in 1 s. cycles. 
R901D	**2 s clock pulse relay**	Repeats on/off operations in 2 s. cycles. 
R901E	**1 min clock pulse relay**	Repeats on/off operations in 1 min. cycles. 
R901F	**Not used**	-

WR902 FP-X

Address	Name	Description
R9020	**RUN mode flag**	Turns off while the mode selector is set to PROG. Turns on while the mode selector is set to RUN.
R9021	**Not used**	-
R9022	**Not used**	-
R9023	**Not used**	-
R9024	**Not used**	-
R9025	**Not used**	-
R9026	**Message flag**	Turns on while the F149 (MSG) instruction is executed.
R9027	**Not used**	-
R9028	**Not used**	-
R9029	**Forcing flag**	Turns on during forced on/off operation for input/output relay timer/counter contacts.
R902A	**Interrupt enable flag**	Turns on while the external interrupt trigger is enabled by the ICTL instruction.
R902B	**Interrupt error flag**	Turns on when an interrupt error occurs.
R902C	**Sample point flag**	Sampling by the instruction=0 Sampling at constant time intervals=1
R902D	**Sample trace end flag**	When the sampling operation stops=1, When the sampling operation starts=0
R902E	**Sampling stop trigger flag**	When the sampling stop trigger activates=1 When the sampling stop trigger stops=0
R902F	**Sampling enable flag**	When sampling starts=1 When sampling stops=0

WR903 FP-X

Address	Name	Description
R9030	**Not used**	-
R9031	**Not used**	-
R9032	**COM1 port mode flag**	- Turns on when the general-purpose communication function is being used - Goes off when any function other than the general-purpose communication function is being used.
R9033	**PR instruction flag**	Off: Printing is not executed. On: Execution is in progress.
R9034	**Editing in RUN mode flag**	Goes on for ony the first scan following completion of a rewrite during the RUN operation.
R9035	**Not used**	-
R9036	**Not used**	-
R9037	**COM1 port communication error flag**	- Goes on if a transmission error occurs during data communication. - Goes off when a request is made to send data, using the F159 (MTRN) instruction.
R9038	**COM1 port reception done flag during general- purpose serial communication**	- Turns on when the terminator is received during general -purpose serial communication.
R9039	**COM1 port transmission done flag during general-purpose serial communication**	- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose serial communication.
R903A	**Not used**	-
R903B	**Not used**	-
R903C	**Not used**	-
R903D	**Not used**	-
R903E	**TOOL port reception done flag during general purpose communication**	- Turns on when the terminator is received during general -purpose serial communication.
R903F	**TOOL port transmission done flag during general-purpose serial communication**	- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose serial communication.

Note) R9030 to R903F can be changed during 1 scan.

WR904 FP-X

Address	Name	Description
R9040	**TOOL port mode flag**	- Goes on when the general-purpose serial communication is used. - Goes off when the MEWTOCOL is used.
R9041	**COM1 port PC(PLC) link flag**	Turn on while the PC(PLC) link function is used.
R9042	**COM2 port mode flag**	- Goes on when the general-purpose serial communication is used. - Goes off when the MEWTOCOL is used.
R9043	**Not used**	-
R9044	**COM1 port SEND/RECV instruction execution flag**	Monitors whether the F145 (SEND) or F146 (RECV) instructions can be executed or not for the COM1 port. Off: None of the above mentioned instructions can be executed. (During executing the instruction) On: One of the above mentioned instructions can be executed.
R9045	**COM1 port SEND/RECV instruction execution end flag**	Monitors if an abnormality has been detected during the execution of the F145 (SEND) or F146 (RECV) instructions for the COM1 port as follows: Off: No abonormality detected. On: An abnormality detected. (communication error) The error code is stored in DT90124.
R9046	**Not used**	-
R9047	**COM2 port communication error flag**	- Goes on if a transmission error occurs during data communication. - Goes off when a request is made to send data, using the F159 (MTRN) instruction.
R9048	**COM2 port reception done flag during general-purpose communicating**	- Turn on when the terminator is received during general-purpose serial communication.
R9049	**COM2 port transmission done flag during general-purpose communication**	- Goes on when transmission has been completed in general-purpose serial communication. - Goes off when transmission is requested in general-purpose communication.
R904A	**COM2 port SEND/RECV instruction execution flag**	Monitors whether the F145 (SEND) or F146 (RECV) instructions can be executed or not for the COM2 port. Off: None of the above mentioned instructions can be executed. (During executing the instruction) On: One of the above mentioned instructions can be executed.
R904B	**COM2 port SEND/RECV instruction execution end flag**	Monitors if an abnormality has been detected during the execution of the F145 (SEND) or F146 (RECV) instructions for the COM2 port as follows: Off: No abonormality detected. On: An abnormality detected. (communication error) The error code is stored in DT90125.
R904C to R904F	**Not used**	-

Note) R9040 to R904F can be changed during 1 scan.

WR905 FP-X

Address	Name	Description
R9050	**MEWNET-W0 PC(PLC) link transmission error flag**	When using MEWNET-W0 - Turns on when a transmission error occurs at PC(PLC) link. - Turns on when there is an error in the PC(PLC) link area settings.
R9051 to R905F	**Not used**	

WR906 FP-X

Address	Name		Description
R9060	**MEWNET-W0 PC(PLC) link 0 transmission assurance relay**	**Unit No.1**	Turns on when Unit No. 1 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9061		**Unit No.2**	Turns on when Unit No. 2 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9062		**Unit No.3**	Turns on when Unit No. 3 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9063		**Unit No.4**	Turns on when Unit No. 4 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9064		**Unit No.5**	Turns on when Unit No. 5 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9065		**Unit No.6**	Turns on when Unit No. 6 is communicating properly in PLC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PLC link mode.
R9066		**Unit No.7**	Turns on when Unit No. 7 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9067		**Unit No.8**	Turns on when Unit No. 8 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9068		**Unit No.9**	Turns on when Unit No. 9 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9069		**Unit No.10**	Turns on when Unit No. 10 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R906A		**Unit No.11**	Turns on when Unit No. 11 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R906B		**Unit No.12**	Turns on when Unit No. 12 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R906C		**Unit No.13**	Turns on when Unit No. 13 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R906D		**Unit No.14**	Turns on when Unit No. 14 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R906E		**Unit No.15**	Turns on when Unit No. 15 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R906F		**Unit No.16**	Turns on when Unit No. 16 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.

WR907 FP-X

Address	Name		Description
R9070	**MEWNET-W0 PC(PLC) link 0 operation mode relay**	**Unit No.1**	Turns on when Unit No. 1 is in the RUN mode. Turns off when Unit No. 1 is in the PROG. mode.
R9071		**Unit No.2**	Turns on when Unit No. 2 is in the RUN mode. Turns off when Unit No. 2 is in the PROG. mode.
R9072		**Unit No.3**	Turns on when Unit No. 3 is in the RUN mode. Turns off when Unit No. 3 is in the PROG. mode.
R9073		**Unit No.4**	Turns on when Unit No. 4 is in the RUN mode. Turns off when Unit No. 4 is in the PROG. mode.
R9074		**Unit No.5**	Turns on when Unit No. 5 is in the RUN mode. Turns off when Unit No. 5 is in the PROG. mode.
R9075		**Unit No.6**	Turns on when Unit No. 6 is in the RUN mode. Turns off when Unit No. 6 is in the PROG. mode.
R9076		**Unit No.7**	Turns on when Unit No. 7 is in the RUN mode. Turns off when Unit No. 7 is in the PROG. mode.
R9077		**Unit No.8**	Turns on when Unit No. 8 is in the RUN mode. Turns off when Unit No. 8 is in the PROG. mode.
R9078		**Unit No.9**	Turns on when Unit No. 9 is in the RUN mode. Turns off when Unit No. 9 is in the PROG. mode.
R9079		**Unit No.10**	Turns on when Unit No. 10 is in the RUN mode. Turns off when Unit No. 10 is in the PROG. mode.
R907A		**Unit No.11**	Turns on when Unit No. 11 is in the RUN mode. Turns off when Unit No. 11 is in the PROG. mode.
R907B		**Unit No.12**	Turns on when Unit No. 12 is in the RUN mode. Turns off when Unit No. 12 is in the PROG. mode.
R907C		**Unit No.13**	Turns on when Unit No. 13 is in the RUN mode. Turns off when Unit No. 13 is in the PROG. mode.
R907D		**Unit No.14**	Turns on when Unit No. 14 is in the RUN mode. Turns off when Unit No. 14 is in the PROG. mode.
R907E		**Unit No.15**	Turns on when Unit No. 15 is in the RUN mode. Turns off when Unit No. 15 is in the PROG. mode.
R907F		**Unit No.16**	Turns on when Unit No. 16 is in the RUN mode. Turns off when Unit No. 16 is in the PROG. mode.

WR908 FP-X

Address	Name		Description
R9080	**MEWNET-W0 PC(PLC) link 1 transmission assurance relay**	**Unit No.1**	Turns on when Unit No. 1 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9081		**Unit No.2**	Turns on when Unit No. 2 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9082		**Unit No.3**	Turns on when Unit No. 3 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9083		**Unit No.4**	Turns on when Unit No. 4 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9084		**Unit No.5**	Turns on when Unit No. 5 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9085		**Unit No.6**	Turns on when Unit No. 6 is communicating properly in PLC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PLC link mode.
R9086		**Unit No.7**	Turns on when Unit No. 7 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9087		**Unit No.8**	Turns on when Unit No. 8 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9088		**Unit No.9**	Turns on when Unit No. 9 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R9089		**Unit No.10**	Turns on when Unit No. 10 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R908A		**Unit No.11**	Turns on when Unit No. 11 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R908B		**Unit No.12**	Turns on when Unit No. 12 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R908C		**Unit No.13**	Turns on when Unit No. 13 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R908D		**Unit No.14**	Turns on when Unit No. 14 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R908E		**Unit No.15**	Turns on when Unit No. 15 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.
R908F		**Unit No.16**	Turns on when Unit No. 16 is communicating properly in PC(PLC) link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC(PLC) link mode.

WR909 FP-X

Address	Name		Description
R9090	**MEWNET-W0 PC(PLC) link 1 operation mode relay**	**Unit No.1**	Turns on when Unit No. 1 is in the RUN mode. Turns off when Unit No. 1 is in the PROG. mode.
R9091		**Unit No.2**	Turns on when Unit No. 2 is in the RUN mode. Turns off when Unit No. 2 is in the PROG. mode.
R9092		**Unit No.3**	Turns on when Unit No. 3 is in the RUN mode. Turns off when Unit No. 3 is in the PROG. mode.
R9093		**Unit No.4**	Turns on when Unit No. 4 is in the RUN mode. Turns off when Unit No. 4 is in the PROG. mode.
R9094		**Unit No.5**	Turns on when Unit No. 5 is in the RUN mode. Turns off when Unit No. 5 is in the PROG. mode.
R9095		**Unit No.6**	Turns on when Unit No. 6 is in the RUN mode. Turns off when Unit No. 6 is in the PROG. mode.
R9096		**Unit No.7**	Turns on when Unit No. 7 is in the RUN mode. Turns off when Unit No. 7 is in the PROG. mode.
R9097		**Unit No.8**	Turns on when Unit No. 8 is in the RUN mode. Turns off when Unit No. 8 is in the PROG. mode.
R9098		**Unit No.9**	Turns on when Unit No. 9 is in the RUN mode. Turns off when Unit No. 9 is in the PROG. mode.
R9099		**Unit No.10**	Turns on when Unit No. 10 is in the RUN mode. Turns off when Unit No. 10 is in the PROG. mode.
R909A		**Unit No.11**	Turns on when Unit No. 11 is in the RUN mode. Turns off when Unit No. 11 is in the PROG. mode.
R909B		**Unit No.12**	Turns on when Unit No. 12 is in the RUN mode. Turns off when Unit No. 12 is in the PROG. mode.
R909C		**Unit No.13**	Turns on when Unit No. 13 is in the RUN mode. Turns off when Unit No. 13 is in the PROG. mode.
R909D		**Unit No.14**	Turns on when Unit No. 14 is in the RUN mode. Turns off when Unit No. 14 is in the PROG. mode.
R909E		**Unit No.15**	Turns on when Unit No. 15 is in the RUN mode. Turns off when Unit No. 15 is in the PROG. mode.
R909F		**Unit No.16**	Turns on when Unit No. 16 is in the RUN mode. Turns off when Unit No. 16 is in the PROG. mode.

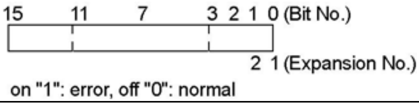
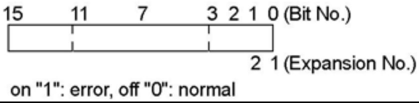
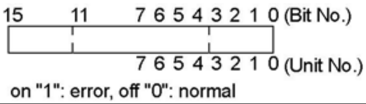
WR910　FP-X

Address	Name		Description
R9100 to R910F	**Not used**		-
R9110	**Control flag**	**HSC-CH0**	- Turns on while the F166 (HC1S) and F167 (HC1R) instructions are executed. - Turns off when the F166 (HC1S) and F167 (HC1R) instructions are completed.
R9111		**HSC-CH1**	
R9112		**HSC-CH2**	
R9113		**HSC-CH3**	
R9114		**HSC-CH4**	
R9115		**HSC-CH5**	
R9116		**HSC-CH6**	
R9117		**HSC-CH7**	
R9118		**HSC-CH8** Note1)	
R9119		**HSC-CH9** Note1)	
R911A		**HSC-CHA** Note1)	
R911B		**HSC-CHB** Note1)	
R911C		**PLS-CH0**	- Turns on while the pulses are being output by the F171 (SPDH), F172 (PLSH), F173 (PWMH) and F174 (SP0H) instructions.
R911D		**PLS-CH1**	
R911E		**PLS-CH2** Note2)	
R911F		**PLS-CH3** Note2)	

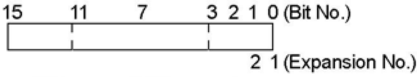
Note1) This relay is avaialble for the FP-X Ry type only.
Note2) This relay is available for the FP-X Tr type only.

5.1.15 Table of Special Data Registers for FP-X

FP-X (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90000	**Self-diagnostic error code**	The self-diagnostic error code is stored here when a self-diagnostic error occurs.	A	N/A
DT90001	**Not used**	-	N/A	N/A
DT90002	**Position of abnormal I/O board for application cassette**	When an error occurs at the I/O board for the application cassette, the bit corresponding to the board will be set on. 	A	N/A
DT90003	**Not used**	-	N/A	N/A
DT90004	**Not used**	-	N/A	N/A
DT90005	**Not used**	-	N/A	N/A
DT90006	**Position of abnormal application cassette**	When an error occurs at the intelligent board for the application cassette, the bit corresponding to the board will be set on. 	A	N/A
DT90007	**Not used**	-	N/A	N/A
DT90008	**Not used**	-	N/A	N/A
DT90009	**Communication error flag for COM2**	Stores the error contents when using COM2 port.	A	N/A
DT90010	**Extension I/O verify error unit**	When the state of installation of FP-X expansion I/O unit has changed since the power was turned on, the bit corresponding to the unit No. will turn on. Monitor using binary display. 	A	N/A

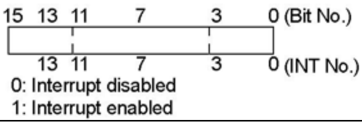
FP-X (A: Available, N/A: Not available)

Address	Name	Description	Reading	Writing
DT90011	**Add-on cassette verify error unit**	When the state of installation of an FP-X add-on cassette has changed since the power was turned on, the bit corresponding to the unit No. will turn on. Monitor using binary display.  on "1": error, off "0": normal	A	N/A
DT90012	**Not used**	-	N/A	N/A
DT90013	**Not used**	-	N/A	N/A
DT90014	**Operation auxiliary register for data shift instruction**	One shift-out hexadecimal digit is stored in bit positions 0 to 3 when the data shift instruction, F105 (BSR) or F106 (BSL) is executed. The value can be read and written by executing F0 (MV) instruction.	A	A
DT90015	**Operation auxiliary register for division instruction**	The divided remainder (16-bit) is stored in DT90015 when the division instruction F32(%) or F52(B%) instruction is executed. The divided remainder (32-bit) is stored in DT90015 and DT90016 when the division instruction F33(D%) or F53(DB%) is executed. The value can be read and written by executing F0(MV) instruction.	A	A
DT90016			A	A
DT90017	**Operation error address (hold type)**	After commencing operation, the address where the first operation error occurred is stored. Monitor the address using decimal display.	A	N/A
DT90018	**Operation error address (non-hold type)**	The address where an operation error occurred is stored. Each time an error occurs, the new address overwrites the previous address. At the beginning of a scan, the address is 0. Monitor the address using decimal display.	A	N/A
DT90019	**2.5 ms ring counter** Note1)	The data stored here is increased by one every 2.5 ms. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 2.5 ms = Elapsed time between the two points.	A	N/A
DT90020	**10 μs ring counter** Note1) Note2)	The data stored here is increased by one every 10.24 μs. (H0 to HFFFF) Difference between the values of the two points (absolute value) x 10.24 μs = Elapsed time between the two points. Note) The exact value is 10.24 μs.	A	N/A
DT90021	**Not used**	-	N/A	N/A

Note1) It is renewed once at the beginning of each one scan.

Note2) As DT90020 is renewed even if F0(MV), DT90020 and D instruction is being executed, it can be used to measure the block time.

FP-X (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90022	**Scan time (current value) Note)**	The current scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.	A	N/A
DT90023	**Scan time (minimum value) Note)**	The minimum scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K50 indicates 5 ms.	A	N/A
DT90024	**Scan time (maximum value) Note)**	The maximum scan time is stored here. The scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 ms Example: K125 indicates 12.5 ms.	A	N/A
DT90025	**Mask condition monitoring register for interrupts (INT0 to 13)**	The mask conditions of interrupts using the instruction can be stored here. Monitor using binary display.  	A	N/A
DT90026	**Not used**	-	N/A	N/A
DT90027	**Periodical interrupt interval (INT24)**	The value set by ICTL instruction is stored. K0: periodical interrupt is not used. K1 to K3000: 0.5ms to 1.5s or 10ms to 30s	A	N/A
DT90028	**Sample trace interval**	K0: Sampling by the SMPL instruction K1 to K3000 (x 10 ms): 10 ms to 30 s	A	N/A
DT90029	**Not used**	-	N/A	N/A
DT90030	**Message 0**	The contents of the specified message (Data length) are stored in these special data registers when F149 (MSG) instruction is executed.	A	N/A
DT90031	**Message 1**			
DT90032	**Message 2**			
DT90033	**Message 3**			
DT90034	**Message 4**			
DT90035	**Message 5**			
DT90036	**Not used**	-	N/A	N/A

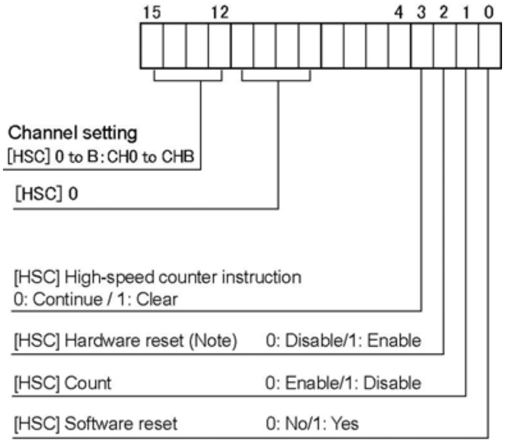
Note) Scan time display is only possible in RUN mode, and shows the operation cycle time. (In PROG. mode, the scan time for the operation is not displayed.) The maximum and minimum values are cleared each time the mode is switched from RUN to PROG.

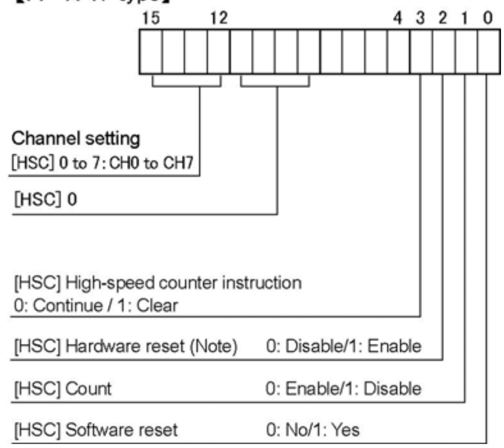
FP-X (A: Available, N/A: Not available)

Address	Name	Description	Reading	Writing
DT90037	**Work1 for SRC instructions**	The number of data that match the searched data is stored here when F96 (SRC) insturction is executed.	A	N/A
DT90038	**Work2 for SRC instructions**	The position of the first matching data is stored here when an F96 (SRC) instruction is executed.	A	N/A
DT90039	**Not used**	-	N/A	N/A
DT90040	**Volume input 0**	The potentiometer value (K0 to K1000) is stored here. This value can be used in analog timers and other applications by using the program to read this value to a data register. V0→DT90040 V1→DT90041	A	N/A
DT90041	**Volume input 1**			
DT90042	**Volume input 2**	For C60 only: The potentiometer value (K0 to K1000) is stored here. This value can be used in analog timers and other applications by using the program to read this value to a data register. V2→DT90042 V3→DT90043	A	N/A
DT90043	**Volume input 3**			
DT90044	**System work**	Used by the system.	A	A
DT90045	**Not used**	-	N/A	N/A
DT90046	**Not used**	-	N/A	N/A
DT90047	**Not used**	-	N/A	N/A
DT90048	**Not used**	-	N/A	N/A
DT90049	**Not used**	-	N/A	N/A
DT90050	**Not used**	-	N/A	N/A
DT90051	**Not used**	-	N/A	N/A

FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read
-ing</th><th>Writ-
ing</th></tr>
<tr><td>DT90052</td><td>High-speed counter control flag</td><td>A value can be written with F0 (MV) instruction to reset the high-speed counter, disable counting, continue or clear high-speed counter instruction.

Control code setting
【FP-X Ry type】


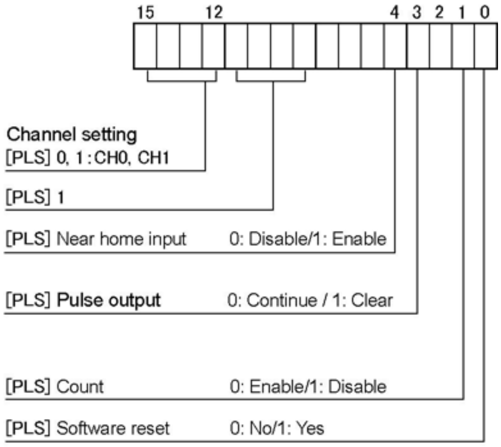
【FP-X Tr type】

</td><td>N/A</td><td>A</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

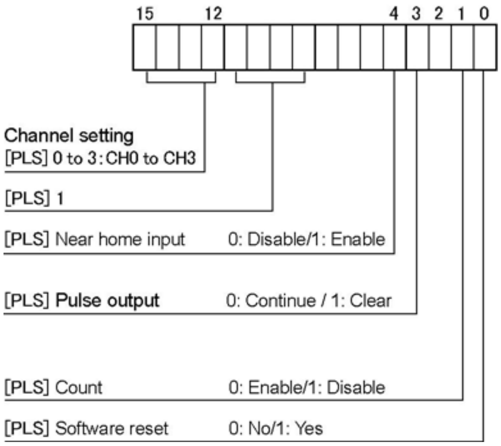
<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read
-ing</th><th>Writ-
ing</th></tr>
<tr><td>DT90052</td><td>Pulse output control flag</td><td>
A value can be written with F0 (MV) instruction to reset the high-speed counter, disable counting, continue or clear high-speed counter instruction.

Control code setting

【FP-X Ry type】

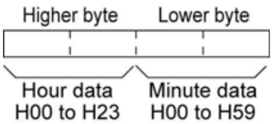
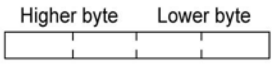


【FP-X Tr type】



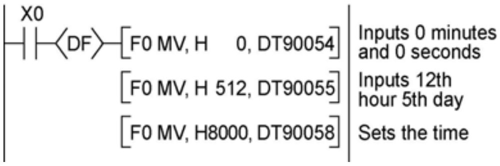
</td><td>N/A</td><td>A</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read
-ing</th><th>Writ-
ing</th></tr>
<tr><td>DT90053</td><td>Real-Time Clock monitor (hour/minute)</td><td>Hour and minute data of the Real-Time Clock are stored here. This data is read-only data. It cannot be overwritten.
</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90054</td><td>Real-Time Clock setting (minute/second)</td><td rowspan="4">The year, month, day, hour, minute, second and day-of-the-week data for the Real-Time Clock is stored. The built-in Real-Time Clock will operate correctly through the year 2099 and supports leap years. The Real-Time Clock can be set by writing a value using a programming tool software or a program that uses the F0 (MV) instruction.(see example for DT90058)

<table>
<tr><td>DT90054</td><td>Minute data (H00 to H59)</td><td>Second data (H00 to H59)</td></tr>
<tr><td>DT90055</td><td>Day data (H01 to H31)</td><td>Hour data (H00 to H23)</td></tr>
<tr><td>DT90056</td><td>Year data (H00 to H99)</td><td>Month data (H01 to H12)</td></tr>
<tr><td>DT90057</td><td>—</td><td>Day-of-the-week (H00 to H06)</td></tr>
</table>
As a day of the week is not automatially set on FPWIN GR, fix what day is set to 00, and set each value for 00 to 06.</td><td rowspan="4">A</td><td rowspan="4">A</td></tr>
<tr><td>DT90055</td><td>Real-Time Clock setting (day/hour)</td></tr>
<tr><td>DT90056</td><td>Real-Time Clock setting (year/month)</td></tr>
<tr><td>DT90057</td><td>Real-Time Clock setting (day-of-the-week)</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90058</td><td>Real-Time Clock time setting and 30 seconds correction register</td><td>
The Real-Time Clock is adjusted as follows.
When setting the Real-Time Clock by program
By setting the highest bit of DT90058 to 1, the time becomes that written to DT90054 to DT90057 by F0 (MV) instruction. After the time is set, DT90058 is cleared to 0. (Cannot be performed with any instruction other than F0 (MV) instruction.)

<Example>
Set the time to 12:00:00 on the 5th day when the X0 turns on.


Note) If the values of DT90054 to DT90057 are changed with the programming tool software, the time will be set when the new values are written. Therefore, it is unnecessary to write to DT90058.

When the correcting times less than 30 seconds
By setting the lowest bit of DT90058 to 1, the value will be moved up or down and become exactly 0 seconds. After the correction is completed, DT90058 is cleared to 0.

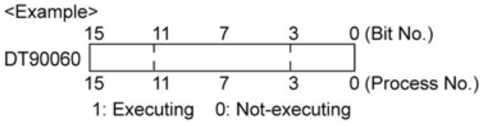
Example:
Correct to 0 seconds with X0: on


At the time of coorection, if between 0 and 29 seconds, it will be moved down, and if between 30 and 59 seconds, it will be moved up.
In the example above, if the time was 5 minutes 29 seconds, it will become 5 minutes 0 seconds;and, if the time was 5 minutes 35 seconds, it will become 6 minutes 0 seconds.
</td><td>A</td><td>A</td></tr>
<tr><td>DT90059</td><td>Communication error code</td><td>Error code is sotred here when a communication error occurs.</td><td>N/A</td><td>N/A</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>**DT90060**</td><td>**Step ladder process (0 to 15)**</td><td rowspan="22">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on.

Monitor using binary display.

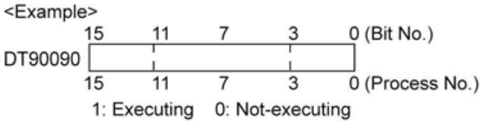
<Example>


A programming tool software can be used to write data.</td><td rowspan="22">A</td><td rowspan="22">A</td></tr>
<tr><td>**DT90061**</td><td>**Step ladder process (16 to 31)**</td></tr>
<tr><td>**DT90062**</td><td>**Step ladder process (32 to 47)**</td></tr>
<tr><td>**DT90063**</td><td>**Step ladder process (48 to 63)**</td></tr>
<tr><td>**DT90064**</td><td>**Step ladder process (64 to 79)**</td></tr>
<tr><td>**DT90065**</td><td>**Step ladder process (80 to 95)**</td></tr>
<tr><td>**DT90066**</td><td>**Step ladder process (96 to 111)**</td></tr>
<tr><td>**DT90067**</td><td>**Step ladder process (112 to 127)**</td></tr>
<tr><td>**DT90068**</td><td>**Step ladder process (128 to 143)**</td></tr>
<tr><td>**DT90069**</td><td>**Step ladder process (144 to 159)**</td></tr>
<tr><td>**DT90070**</td><td>**Step ladder process (160 to 175)**</td></tr>
<tr><td>**DT90071**</td><td>**Step ladder process (176 to 191)**</td></tr>
<tr><td>**DT90072**</td><td>**Step ladder process (192 to 207)**</td></tr>
<tr><td>**DT90073**</td><td>**Step ladder process (208 to 223)**</td></tr>
<tr><td>**DT90074**</td><td>**Step ladder process (224 to 239)**</td></tr>
<tr><td>**DT90075**</td><td>**Step ladder process (240 to 255)**</td></tr>
<tr><td>**DT90076**</td><td>**Step ladder process (256 to 271)**</td></tr>
<tr><td>**DT90077**</td><td>**Step ladder process (272 to 287)**</td></tr>
<tr><td>**DT90078**</td><td>**Step ladder process (288 to 303)**</td></tr>
<tr><td>**DT90079**</td><td>**Step ladder process (304 to 319)**</td></tr>
<tr><td>**DT90080**</td><td>**Step ladder process (320 to 335)**</td></tr>
<tr><td>**DT90081**</td><td>**Step ladder process (336 to 351)**</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read-
ing</th><th>Writ-
ing</th></tr>
<tr><td>DT90082</td><td>Step ladder process (352 to 367)</td><td rowspan="16">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on .

Monitor using binary display.

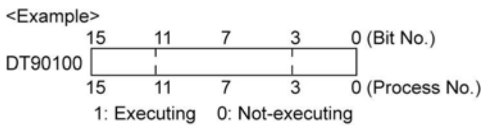
<Example>


A programming tool software can be used to write data.</td><td rowspan="16">A</td><td rowspan="16">A</td></tr>
<tr><td>DT90083</td><td>Step ladder process (368 to 383)</td></tr>
<tr><td>DT90084</td><td>Step ladder process (384 to 399)</td></tr>
<tr><td>DT90085</td><td>Step ladder process (400 to 415)</td></tr>
<tr><td>DT90086</td><td>Step ladder process (416 to 431)</td></tr>
<tr><td>DT90087</td><td>Step ladder process (432 to 447)</td></tr>
<tr><td>DT90088</td><td>Step ladder process (448 to 463)</td></tr>
<tr><td>DT90089</td><td>Step ladder process (464 to 479)</td></tr>
<tr><td>DT90090</td><td>Step ladder process (480 to 495)</td></tr>
<tr><td>DT90091</td><td>Step ladder process (496 to 511)</td></tr>
<tr><td>DT90092</td><td>Step ladder process (512 to 527)</td></tr>
<tr><td>DT90093</td><td>Step ladder process (528 to 543)</td></tr>
<tr><td>DT90094</td><td>Step ladder process (544 to 559)</td></tr>
<tr><td>DT90095</td><td>Step ladder process (560 to 575)</td></tr>
<tr><td>DT90096</td><td>Step ladder process (576 to 591)</td></tr>
<tr><td>DT90097</td><td>Step ladder process (592 to 607)</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th>Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90098</td><td>Step ladder process (608 to 623)</td><td rowspan="25">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on.

Monitor using binary display



A programming tool software can be used to write data.</td><td rowspan="25">A</td><td rowspan="25">A</td></tr>
<tr><td>DT90099</td><td>Step ladder process (624 to 639)</td></tr>
<tr><td>DT90100</td><td>Step ladder process (640 to 655)</td></tr>
<tr><td>DT90101</td><td>Step ladder process (656 to 671)</td></tr>
<tr><td>DT90102</td><td>Step ladder process (672 to 687)</td></tr>
<tr><td>DT90103</td><td>Step ladder process (688 to 703)</td></tr>
<tr><td>DT90104</td><td>Step ladder process (704 to 719)</td></tr>
<tr><td>DT90105</td><td>Step ladder process (720 to 735)</td></tr>
<tr><td>DT90106</td><td>Step ladder process (736 to 751)</td></tr>
<tr><td>DT90107</td><td>Step ladder process (752 to 767)</td></tr>
<tr><td>DT90108</td><td>Step ladder process (768 to 783)</td></tr>
<tr><td>DT90109</td><td>Step ladder process (784 to 799)</td></tr>
<tr><td>DT90110</td><td>Step ladder process (800 to 815)</td></tr>
<tr><td>DT90111</td><td>Step ladder process (816 to 831)</td></tr>
<tr><td>DT90112</td><td>Step ladder process (832 to 847)</td></tr>
<tr><td>DT90113</td><td>Step ladder process (848 to 863)</td></tr>
<tr><td>DT90114</td><td>Step ladder process (864 to 879)</td></tr>
<tr><td>DT90115</td><td>Step ladder process (880 to 895)</td></tr>
<tr><td>DT90116</td><td>Step ladder process (896 to 911)</td></tr>
<tr><td>DT90117</td><td>Step ladder process (912 to 927)</td></tr>
<tr><td>DT90118</td><td>Step ladder process (928 to 943)</td></tr>
<tr><td>DT90119</td><td>Step ladder process (944 to 959)</td></tr>
<tr><td>DT90120</td><td>Step ladder process (960 to 975)</td></tr>
<tr><td>DT90121</td><td>Step ladder process (976 to 991)</td></tr>
<tr><td>DT90122</td><td>Step ladder process (992 to 999) (higher byte is not used.)</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

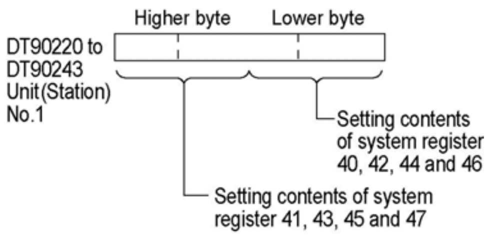
Address	Name	Description	Read -ing	Writ -ing
DT90123	**Not used**	-	N/A	N/A
DT90124	**COM1 SEND/RECV instruction end code**	For details, refer to Programming Manual (F145 and F146).	N/A	N/A
DT90125	**COM2 SEND/RECV instruction end code**	For details, refer to Programming Manual (F145 and F146).	N/A	N/A
DT90126	**Forced ON/OFF operating station display**	Used by the system	N/A	N/A
DT90127 to DT90139	**Not used**	-	N/A	N/A
DT90140	**MEWNET-W0 PC(PLC) link 0 status**	The number of times the receiving operation is performed.	A	N/A
DT90141		The current interval between two receiving operations: value in the register x 2.5ms		
DT90142		The minimum inerval between two receiving operations: value in the register x 2.5ms		
DT90143		The maximum interval between two receiving operations: value in the register x 2.5ms		
DT90144		The number of times the sending operation is performed.		
DT90145		The current interval between two sending operations: value in the register x 2.5ms		
DT90146		The minimum interval between two sending operations: value in the register x 2.5ms		
DT90147		The maximum interval between two sending operations: value in the register x 2.5ms		
DT90148	**MEWNET-W0 PC(PLC) link 1 status**	The number of times the receiving operation is performed.	A	N/A
DT90149		The current interval between two receiving operations: value in the register x 2.5ms		
DT90150		The minimum inerval between two receiving operations: value in the register x 2.5ms		
DT90151		The maximum interval between two receiving operations: value in the register x 2.5ms		
DT90152		The number of times the sending operation is performed.		
DT90153		The current interval between two sending operations: value in the register x 2.5ms		
DT90154		The minimum interval between two sending operations: value in the register x 2.5ms		
DT90155		The maximum interval between two sending operations: value in the register x 2.5ms		
DT90156	**MEWNET-W0 PC(PLC) link 0 status**	Area used for measurement of receiving interval.	A	N/A
DT90157		Area used for measurement of sending interval.		

FP-X (A: Available, N/A: Not available)

Address	Name	Description	Read-ing	Writ-ing
DT90158	MEWNET-W0 PC(PLC) link 1 status	Area used for measurement of receiving interval.	A	N/A
DT90159		Area used for measurement of sending interval.		
DT90160	MEWNET-W0 PC(PLC) link 0 unit No.	Stores the unit No. of PC(PLC) link 0.	A	N/A
DT90161	MEWNET-W0 PC(PLC) link 0 error flag	Stores the error contents of PC(PLC) link 0.	A	N/A
DT90162 to DT90169	Not used	-	N/A	N/A
DT90170	MEWNET-W0 PC(PLC) link 0 status	Duplicated destination for PC(PLC) inter-link address	A	N/A
DT90171		Counts how many times a token is lost.		
DT90172		Counts how many times two or more tokens are detected.		
DT90173		Counts how many times a signal is lost.		
DT90174		No. of times underfined commands have been received.		
DT90175		No. of times sum check errors have occurred during reception.		
DT90176		No. of times format errors have occurred in received data.		
DT90177		No. of times transmission errors have occurred.		
DT90178		No. of times procedural errors have occurred.		
DT90179		No. of times overlapping parent units have occurred.		
DT90180 to DT90189	Not used	-	N/A	N/A
DT90190	Not used	-	N/A	N/A
DT90191	Not used	-	N/A	N/A
DT90192	Not used	-	N/A	N/A
DT90193	Not used	-	N/A	N/A
DT90194 to DT90218	Not used	-	N/A	N/A

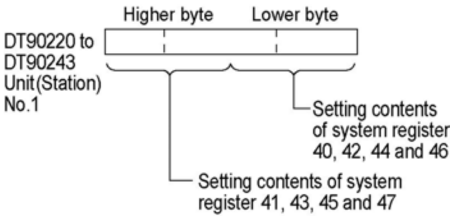
FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th colspan="2">Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90219</td><td colspan="2">Unit No. (Station No.) selection for DT90220 to DT90251</td><td>0: Unit No. (Station No.) 1 to 8,
1: Unit No. (Station No.) 9 to 16</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90220</td><td rowspan="4">PC(PLC) link Unit (station) No. 1 or 9</td><td>System regis-ter 40 and 41</td><td rowspan="20">The contents of the system register settings partaining to the PLC inter-link function for the various unit numbers are stored as shown below.

<Example>
When DT90219 is 0

• When the system register 46 in the home unit is in the standard setting, the values in the home unit are copied in the system registers 46 and 47.
When the system register 46 in the home unit is in the reverse setting, the registers 40 to 45 and 47 corresponding to the home unit mentioned in the left column will be changed to 50 to 55 and 57, and the system register 46 will be set as it is.
Also, the system registers 40 to 45 corresponding to other units will be changed to the values which the received values are corrected, and the registers 46 and 57 in the home unit are set for the registers 46 and 47.</td><td rowspan="20">A</td><td rowspan="20">N/A</td></tr>
<tr><td>DT90221</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90222</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90223</td><td>System regis-ter 46 and 47</td></tr>
<tr><td>DT90224</td><td rowspan="4">PC(PLC) link Unit (station) No. 2 or 10</td><td>System regis-ter 40 and 41</td></tr>
<tr><td>DT90225</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90226</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90227</td><td>System regis-ter 46 and 47</td></tr>
<tr><td>DT90228</td><td rowspan="4">PC(PLC) link Unit (station) No. 3 or 11</td><td>System regis-ter 40 and 41</td></tr>
<tr><td>DT90229</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90230</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90231</td><td>System regis-ter 46 and 47</td></tr>
<tr><td>DT90232</td><td rowspan="4">PC(PLC) link Unit (station) No. 4 or 12</td><td>System regis-ter 40 and 41</td></tr>
<tr><td>DT90233</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90234</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90235</td><td>System regis-ter 46 and 47</td></tr>
<tr><td>DT90236</td><td rowspan="4">PC(PLC) link Unit (station) No. 5 or 13</td><td>System regis-ter 40 and 41</td></tr>
<tr><td>DT90237</td><td>System regis-ter 42 and 43</td></tr>
<tr><td>DT90238</td><td>System regis-ter 44 and 45</td></tr>
<tr><td>DT90239</td><td>System regis-ter 46 and 47</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

<table>
<tr><th>Address</th><th colspan="2">Name</th><th>Description</th><th>Read-ing</th><th>Writ-ing</th></tr>
<tr><td>DT90240</td><td rowspan="4">PC(PLC) link Unit (station) No. 6 or 14</td><td>System register 40 and 41</td><td rowspan="12">The contents of the system register settings partaining to the PLC inter-link function for the various unit numbers are stored as shown below.

<Example> when DT90219 is 0.


• When the system register 46 in the home unit is in the standard setting, the values in the home unit are copied in the system registers 46 and 47.
When the system register 46 in the home unit is in the reverse setting, the registers 40 to 45 and 47 corresponding to the home unit mentioned in the left column will be changed to 50 to 55 and 57, and the system register 46 will be set as it is.
Also, the system registers 40 to 45 corresponding to other units will be changed to the values which the received values are corrected, and the registers 46 and 57 in the home unit are set for the registers 46 and 47.</td><td rowspan="12">A</td><td rowspan="12">N/A</td></tr>
<tr><td>DT90241</td><td>System register 42 and 43</td></tr>
<tr><td>DT90242</td><td>System register 44 and 45</td></tr>
<tr><td>DT90243</td><td>System register 46 and 47</td></tr>
<tr><td>DT90244</td><td rowspan="4">PC(PLC) link Unit (station) No. 7 or 15</td><td>System register 40 and 41</td></tr>
<tr><td>DT90245</td><td>System register 42 and 43</td></tr>
<tr><td>DT90246</td><td>System register 44 and 45</td></tr>
<tr><td>DT90247</td><td>System register 46 and 47</td></tr>
<tr><td>DT90248</td><td rowspan="4">PC(PLC) link Unit (station) No. 8 or 16</td><td>System register 40 and 41</td></tr>
<tr><td>DT90249</td><td>System register 42 and 43</td></tr>
<tr><td>DT90250</td><td>System register 44 and 45</td></tr>
<tr><td>DT90251</td><td>System register 46 and 47</td></tr>
<tr><td>DT90252</td><td colspan="2">Not used</td><td></td><td rowspan="4">N/A</td><td rowspan="4">N/A</td></tr>
<tr><td>DT90253</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90254</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90255</td><td colspan="2">Not used</td><td></td></tr>
<tr><td>DT90256</td><td colspan="2">Not used</td><td></td><td>N/A</td><td>N/A</td></tr>
</table>

FP-X (A: Available, N/A: Not available)

Address	Name			Description	Reading	Writing
DT90300	Elapsed value area	Lower words	HSC-CH0	Counting area for input (X0) or (X0, X1) of the main unit.	A	A Note)
DT90301		Higher words			A	A Note)
DT90302	Target value area	Lower words		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90303		Higher words			A	A Note)
DT90304	Elapsed value area	Lower words	HSC-CH1	Counting area for input (X1) of the main unit.	A	A Note)
DT90305		Higher words			A	A Note)
DT90306	Target value area	Lower words		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90307		Higher words			A	A Note)
DT90308	Elapsed value area	Lower words	HSC-CH2	Counting area for input (X2) or (X2, X3) of the main unit.	A	A Note)
DT90309		Higher words			A	A Note)
DT90310	Target value area	Lower words		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90311		Higher words			A	A Note)
DT90312	Elapsed value area	Lower words	HSC-CH3	Counting area for input (X3) of the main unit.	A	A Note)
DT90313		Higher words			A	A Note)
DT90314	Target value area	Lower words		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90315		Higher words			A	A Note)
DT90316	Elapsed value area	Lower words	HSC-CH4	Counting area for input (X4) or (X4, X5) of the main unit.	A	A Note)
DT90317		Higher words			A	A Note)
DT90318	Target value area	Lower words		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note)
DT90319		Higher words			A	A Note)

Note) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F166 (HC1S) and F167 (HC1R) instructions only.

FP-X (A: Available, N/A: Not available)

Address	Name			Description	Reading	Writing
DT90320	**Elapsed value area**	**Lower words**	**HSC-CH5**	Counting area for input (X5) of the main unit.	A	A Note1)
DT90321		**Higher words**			A	A Note1)
DT90322	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90323		**Higher words**			A	A Note1)
DT90324	**Elapsed value area**	**Lower words**	**HSC-CH6**	Counting area for input (X6) or (X6, X7) of the main unit.	A	A Note1)
DT90325		**Higher words**			A	A Note1)
DT90326	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90327		**Higher words**			A	A Note1)
DT90328	**Elapsed value area**	**Lower words**	**HSC-CH7**	Counting area for input (X7) of the main unit.	A	A Note1)
DT90329		**Higher words**			A	A Note1)
DT90330	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90331		**Higher words**			A	A Note1)
DT90332	**Elapsed value area**	**Lower words**	**HSC-CH8 Note2)**	Counting area for input (X0) or (X0, X1) of the main unit.	A	A Note1)
DT90333		**Higher words**			A	A Note1)
DT90334	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90335		**Higher words**			A	A Note1)

Note1) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F166 (HC1S) and F167 (HC1R) instructions only.
Note2) Available for the FP-X Ry type only.

FP-X (A: Available, N/A: Not available)

Address	Name			Description	Reading	Writing
DT90336	**Elapsed value area**	**Lower words**	**HSC-CH9** Note2)	Counting area for input (X1) of the pulse I/O cassette.	A	A Note1)
DT90337		**Higher words**			A	A Note1)
DT90338	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90339		**Higher words**			A	A Note1)
DT90340	**Elapsed value area**	**Lower words**	**HSC-CHA** Note2)	Counting area for input (X3) or (X3, X4) of the pulse I/O cassette.	A	A Note1)
DT90341		**Higher words**			A	A Note1)
DT90342	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90343		**Higher words**			A	A Note1)
DT90344	**Elapsed value area**	**Lower words**	**HSC-CHB** Note2)	Counting area for input (X4) of the pulse I/O cassette.	A	A Note1)
DT90345		**Higher words**			A	A Note1)
DT90346	**Target value area**	**Lower words**		The target value is set when instructions F166 (HC1S) and F167 (HC1R) are executed.	A	A Note1)
DT90347		**Higher words**			A	A Note1)

Note1) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F166 (HC1S) and F167 (HC1R) instructions only.
Note2) Available for the FP-X Tr type only.

FP-X Tr type FP-X (A: Available, N/A: Not available)

Address	Name			Description	Reading	Writing
DT90348	Elapsed value area	Lower words	PLS-CH0	Counting area for the pulse I/O CH0 (Y0, Y1).	A	A Note)
DT90349		Higher words			A	A Note)
DT90350	Target value area	Lower words		The target value is set when instructions F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) are executed.	A	A Note)
DT90351		Higher words			A	A Note)
DT90352	Elapsed value area	Lower words	PLS-CH1	Counting area for the pulse I/O CH1 (Y2, Y3).	A	A Note)
DT90353		Higher words			A	A Note)
DT90354	Target value area	Lower words		The target value is set when instructions F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) are executed.	A	A Note)
DT90355		Higher words			A	A Note)
DT90356	Elapsed value area	Lower words	PLS-CH2	Counting area for the pulse I/O CH2 (Y4, Y5).	A	A Note)
DT90357		Higher words			A	A Note)
DT90358	Target value area	Lower words		The target value is set when instructions F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) are executed.	A	A Note)
DT90359		Higher words			A	A Note)
DT90360	Elapsed value area	Lower words	PLS-CH3	Counting area for the pulse I/O CH3 (Y6, Y7).	A	A Note)
DT90361		Higher words			A	A Note)
DT90362	Target value area	Lower words		The target value is set when instructions F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) are executed.	A	A Note)
DT90363		Higher words			A	A Note)

Note) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) instructions only.

FP-X Tr type FP-X (A: Available, N/A: Not available)

Address	Name		Description	Read-ing	Writ-ing
DT90370	Control flag monitor area	HSC-CH0	When HSC control is executed by F0 (MV)S, DT90052 instruction, the setting value for the target CH is stored in each CH.	A	N/A
DT90371		HSC-CH1		A	N/A
DT90372		HSC-CH2		A	N/A
DT90373		HSC-CH3		A	N/A
DT90374		HSC-CH4		A	N/A
DT90375		HSC-CH5		A	N/A
DT90376		HSC-CH6		A	N/A
DT90377		HSC-CH7		A	N/A
DT90378					
DT90379					
DT90380		PLS-CH0		A	N/A
DT90381		PLS-CH1		A	N/A
DT90382		PLS-CH2		A	N/A
DT90383		PLS-CH3		A	N/A

Note) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) instructions only.

FP-X Ry type FP-X (A: Available, N/A: Not available)

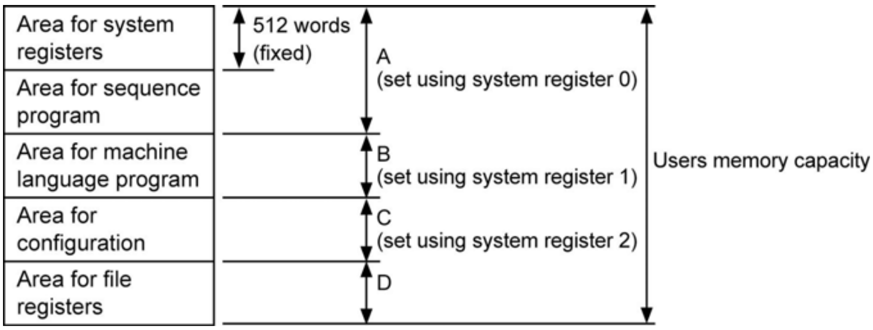
<table>
<tr><th>Address</th><th colspan="3">Name</th><th>Description</th><th>Reading</th><th>Writing</th></tr>
<tr><td>DT90348</td><td rowspan="2">Elapsed value area</td><td>Lower words</td><td rowspan="4">PLS-CH0</td><td rowspan="2">Counting area for output (Y100, Y101) of the pulse I/O cassette.</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90349</td><td>Higher words</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90350</td><td rowspan="2">Target value area</td><td>Lower words</td><td rowspan="2">The target value is set when instructions F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) are executed.</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90351</td><td>Higher words</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90352</td><td rowspan="2">Elapsed value area</td><td>Lower words</td><td rowspan="4">PLS-CH1</td><td rowspan="2">Counting area for output (Y200, Y201) of the pulse I/O cassette.</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90353</td><td>Higher words</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90354</td><td rowspan="2">Target value area</td><td>Lower words</td><td rowspan="2">The target value is set when instructions F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) are executed.</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90355</td><td>Higher words</td><td>A</td><td>A Note)</td></tr>
<tr><td>DT90356</td><td colspan="3">Not used</td><td>-</td><td>N/A</td><td>N/A</td></tr>
<tr><td>DT90357</td><td colspan="3">Not used</td><td>-</td><td>N/A</td><td>N/A</td></tr>
<tr><td>DT90358</td><td colspan="3">Not used</td><td>-</td><td>N/A</td><td>N/A</td></tr>
<tr><td>DT90359</td><td colspan="3">Not used</td><td>-</td><td>N/A</td><td>N/A</td></tr>
<tr><td>DT90360</td><td colspan="2" rowspan="14">Control flag monitor area</td><td>HSC-CH0</td><td rowspan="14">When HSC control is executed by F0 (MV)S, DT90052 instruction, the setting value for the target CH is stored in each CH.</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90361</td><td>HSC-CH1</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90362</td><td>HSC-CH2</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90363</td><td>HSC-CH3</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90364</td><td>HSC-CH4</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90365</td><td>HSC-CH5</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90366</td><td>HSC-CH6</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90367</td><td>HSC-CH7</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90368</td><td>HSC-CH8</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90369</td><td>HSC-CH9</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90370</td><td>HSC-CHA</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90371</td><td>HSC-CHB</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90372</td><td>PLS-CH0</td><td>A</td><td>N/A</td></tr>
<tr><td>DT90373</td><td>PLS-CH1</td><td>A</td><td>N/A</td></tr>
</table>

Note) Writing in the elapsed value area is available by F1 (DMV) instruction only.
Writing in the target value area is available by F171 (SPDH), F172 (PLSH), F174 (SP0H) and F175 (SPSH) instructions only.

5.1.16 Table of System Registers for FP2/FP2SH/FP10SH

Allocation of user memory (system registers 0, 1 and 2)
Available PLC: FP2

The configuration of user memory of FP2 is as follows:



Be sure to set the A (using system ergister 0), B (using system register 1), and C (using system register 2) as even numbers.
The area remaining in A after 512 words are subtracted is the sequence program area that can actually be used.
File register area D is the area that remains after A, B, and C have been subracted from the user memory capacity.
The configuration area is reserved for future expansion.

FP2 (16K)

Users memory capacity	: 16K words
Setting range of A	: 2K to 16K words (default value: 12k)
Setting range of B	: 0 to 14K words (default value: 0)
Setting range of C	:0 to 14K words (default value: 0)

Allocate so that A+B+C≧16

Setting example: The values of D when B = C = 0.

A	Area for sequence program (1024 x A-512)	Area for file registers (D)
2	1,535 steps	14,333 words
4	3,583 steps	12,285 words
6	5,631 steps	10,237 words
8	7,679 steps	8,189 words
10	9,727 steps	6,141 words
12	**11,775 steps (default value)**	**4,093 words (default value)**
14	13,823 steps	2,045 words
16	15,871 steps	0 word

FP2 (32K)

Users memory capacity : 32K words
Setting range of A : 2K to 32K words (default value: 12k)
Setting range of B : 0 to 30K words (default value: 0)
Setting range of C :0 to 30K words (default value: 0)

Allocate so that A + B + C ≦ 32.

Setting example: The values of D when B = C = 0.

A	Area for sequence program (1024 x A-512)	Area for file registers (D)
2	1,535 steps	30,717 words
4	3,583 steps	28,669 words
6	5,631 steps	26,621 words
8	7,679 steps	24,573 words
10	9,727 steps	22,525 words
12	**11,775 steps (default value)**	**20,477 words (default value)**
14	13,823 steps	18,429 words
16	15,871 steps	16,381 words
18	17,919 steps	14,333 words
20	19,967 steps	12,285 words
22	22,015 steps	10,237 words
24	24,063 steps	8,189 words
26	26,111 steps	6,141 words
28	28,159 steps	4,093 words
30	30,207 steps	2,045 words
32	32,255 steps	0 word

Setting example for each area
When not using the machine language program area
Refer to the tables for the different types given above.

When using the machine language program area

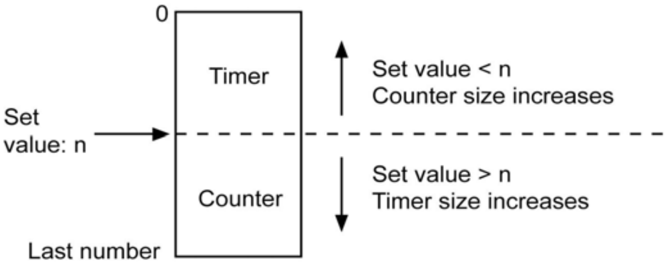
A	Area for machine language program
2	4,096 words
4	8,192 words
6	12,288 words
8	16,384 words
10	20,480 words
12	24,576 words
14	28,672 words
16	32,768 words

B	Area for machine language program
18	36,864 words
20	40,960 words
22	45,056 words
24	49,152 words
26	53,248 words
28	57,344 words
30	61,440 words

For example, for the FP2 (16K-step type), when the area for the sequence program (A) is set to 10K words and the area for configuration (C) is set to 0K words, the area for the machine language program can be set up to 6K words.

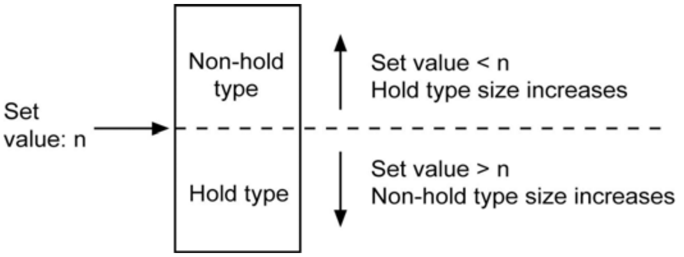
Setting the number of timers and counter (system register 5)
Timers and counters share the same area. If the method of dividing the area is changed, the number of timers and counters will also change.

Type	Total point numbers	Default value of system register 5	Timer	Counter
FP2	1,024 points	1000	1000 points (No. 0 to 999)	24 points (No. 1000 to 1023)
FP2SH/FP10SH	3,072 points	3000	3000 points (No. 0 to 2999)	72 points (No. 3000 to 3071)



For FP2/FP2SH, set the system registers 5 and 6 to the same value. This sets the timer to a non-hold type and counter to a hold type.
By setting system register 5 to “0”, the whole area becomes the counter. Also, by setting it to the value 1 higher than the last number, the whole area becomes the timer.

Hold type area starting address (system registers 6 to 13)
Set each relay and register to a hold type or non-hold type.



For normal situations, set the system registers 5 and 6 to the same value. This sets the timer to a non-hold type and counter to a hold type.
By setting this value to the first number, the whole area becomes hold type. Also, by setting it to the valeu 1 higher than the last number, the whold area becomes non-hold type.
The relays and registers for links not specified in the send area of system registers 40 to 55 are non-hold type regardless of what is set here.

For the FP2SH/FP10SH, the index registers can be set to hold type or non-hold type. The register numbers and settings are related as shown below.

Bank number	Set value for I0 to ID	Bank number	Set value for I0 to ID
Bank 0	0 to 13	**Bank 8**	112 to 125
Bank 1	14 to 27	**Bank 9**	126 to 139
Bank 2	28 to 41	**Bank A**	140 to 153
Bank 3	42 to 45	**Bank B**	154 to 167
Bank 4	56 to 69	**Bank C**	168 to 181
Bank 5	70 to 83	**Bank D**	182 to 195
Bank 6	84 to 97	**Bank E**	196 to 209
Bank 7	98 to 111	**Bank F**	210 to 223

Default value of hold type area setting

Type / Area	FP2	FP2SH
Timer	All non-hold type	
Counter	All hold type	
Internal relay	Non-hold type: 200 words (WR0 to WR199)	Non-hold type: 500 words (WR0 to WR499)
	Hold type: 53 words (WR200 to WR252)	Hold type: 387 words (WR500 to WR886)
Data register	All hold type	
File register	All hold type	
Link relay for MEWNET-W	All hold type	
Link register for MEWNET-W	All hold type	
Index register	-	All hold type

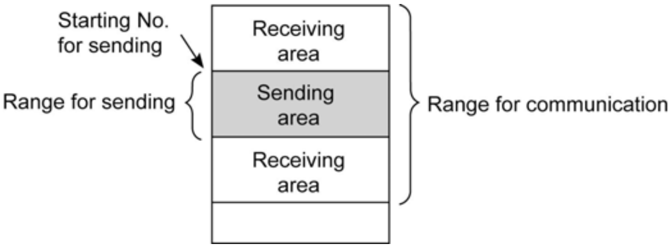
Type / Area	FP10SH
Timer	All non-hold type
Counter	All hold type
Internal relay	Non-hold type: 500 words (WR0 to WR499)
	Hold type: 387 words (WR500 to WR886)
Data register	All hold type
File register	All hold type
Link relay for MEWNET-W/P	All hold type
Link register for MEWNET-W/P	All hold type
Link relay for MEWNET-H	All hold type
Link register for MEWNET-H	All hold type
Index register for FP10SH	All hold type

MEWNET-W PC link setting

For PC link (W) 0: System register 40 to 45

For PC link (W) 1: System register 50 to 55

Regarding the link relays and link data registers, specify the range for communication and divide it up for sending and receiving.



The default settings have the range for communication (system register 40, 41, 50 and 51) set to 0 so that PC link communication is not possible.

If the range for sending (system register 43, 45, 53 and 55) is set to 0, the range for communication will all be for receiving.

The link relay and link data register ranges not used for communication, can each be used as internal relays and data registers.

Table of system registers for FP2/FP2SH/FP10SH

<table>
<tr><th>Item</th><th>Add-ress</th><th colspan="2">Name</th><th>Default value</th><th>Descriptions</th></tr>
<tr><td rowspan="3">Alloca-tion of user memory</td><td>0</td><td colspan="2">Sequence program area capacity setting
Available PLC: FP2</td><td>12K words</td><td>FP2 (16K): 2 to 16K words
FP2 (32K): 2 to 32K words</td></tr>
<tr><td>1</td><td colspan="2">Machine language program area capacity setting
Available PLC: FP2</td><td>0 word</td><td>FP2 (16K): 0 to 14K words
FP2 (32K): 0 to 30K words</td></tr>
<tr><td>2</td><td colspan="2">Configuration capacity setting
Available PLC: FP2</td><td>0 word</td><td>FP2 (16K): 0 to 14K words
FP2 (32K): 0 to 30K words</td></tr>
<tr><td rowspan="12">Action on error</td><td rowspan="12">4</td><td colspan="2">Battery error alarm</td><td>Enabled</td><td>Enabled: When a battery error occurs, a self-diagnostic error is issued and the ERROR LED lights. (BATT. LED lights.)
Disabled: When a battery error occurs, a self-diagnostic error is not issued and the ERROR LED does not light. (BATT. LED does not lights.)</td></tr>
<tr><td rowspan="8">Memory area contents setting at INITIALIZE position

Available PLC: FP2SH, FP10SH</td><td>Internal relay (R)</td><td>Cleared</td><td rowspan="8">Cleared: When the initialize/ test switch is set to INITIALIZE position while in the PROG. mode, you can specify the type of memory to be cleared.
Not cleared: When the initialize/test switch is set to INITIALIZE position while in the PROG. mode, you can specify the tyep of memory to be not cleared</td></tr>
<tr><td>Link relay (L)</td><td>Cleared</td></tr>
<tr><td>Timers/ Counters (T, C, SV, EV)</td><td>Cleared</td></tr>
<tr><td>Data register (DT)</td><td>Cleared</td></tr>
<tr><td>Link data register (LD)</td><td>Cleared</td></tr>
<tr><td>File register (FL)</td><td>Cleared</td></tr>
<tr><td>Index register (I)</td><td>Cleared</td></tr>
<tr><td>Error alarm relay (E)</td><td>Cleared</td></tr>
<tr><td colspan="2">Differential type instructions setting between MC and MCE instructions
Available PLC: FP2SH, FP10SH</td><td>Conven-tional</td><td>Conventional: Holds preceded result in the MC and MCE instruction set.
New: Disregards preceded result in the MC and MCE instruction set.</td></tr>
<tr><td colspan="2">TM instruction operation setting
Available PLC: FP2SH, FP10SH</td><td>Conven-tional</td><td>Conventional: Scan synchronous
New: Scan asynchronous</td></tr>
<tr><td colspan="2">Index modifier check setting</td><td>Enabled</td><td>Enabled: Checks for overflow of the index modifier area, and performs normal processing.
Disabled: Performs processing without checking for overflow of the index modifier area.</td></tr>
</table>

FP2/FP2SH/FP10SH

Item	Address	Name	Default value	Descriptions	
Hold/ Non-hold	5	**Counter starting address (setting the number of timers and counters)**	FP2SH/ FP10SH: 3000	FP2SH/ FP10SH: 0 to 3072	Set the system registers 5 and 6 to the same value.
			FP2: 1000	FP2: 0 to 1024	
	6	**Hold type area starting address setting for timer/counter**	FP2SH/ FP10SH: 3000	FP2SH/ FP10SH: 0 to 3072	
			FP2: 1000	FP2: 0 to 1024	
	7	**Hold type area starting address setting for internal relays (in word units)**	FP2SH/ FP10SH: 500	FP2SH/FP10SH: 0 to 887	
			FP2: 200	FP2: 0 to 253	
	8	**Hold type area starting address setting for data registers**	0	FP2SH/FP10SH: 0 to 10240	
				FP2: 0 to 6000	
	9	**Hold type area starting address setting for file registers (For FP2SH, bank 0)**	0	FP2SH/FP10SH: 0 to 32765	
				FP2 (16K): 0 to 14333 FP2 (32K): 0 to 30717	
	10	**Hold type area starting address setting for MEWNET-W/P link relays (for PC link 0) Note)**	0	0 to 64	
	11	**Hold type area starting address setting for MEWNET-W/P link relays (for PC link 1) Note)**	64	64 to 128	
	12	**Hold type area starting address setting for MEWNET-W/P link data registers (for PC link 0) Note)**	0	0 to 128	
	13	**Hold type area starting address setting for MEWNET-W/P link data registers (for PC link 1) Note)**	128	128 to 256	
	14	**Hold or non-hold setting for step ladder process**	Non-hold	Hold/non-hold	
	15	**Hold type area starting address setting for file register (for bank 1)**	0	0 to 32765	

Note) Available PLC MEWNET-W: FP10SH, FP2, FP2SH
MEWNET-P: FP10SH

FP2/FP2SH/FP10SH

Item	Add-ress	Name	Default value	Descriptions
Hold/ Non-hold	16	**Hold type area starting address setting for MEWNET-H link relays Available PLC: FP10SH**	128	128 to 640
	17	**Hold type area starting address setting for MEWNET-H link data registers Available PLC: FP10SH**	256	256 to 8448
	18	**Hold type area starting address setting for index register Available PLC: FP2SH/ FP10SH**	0	0 to 224
	19	**Hold type area starting address setting for file register (for bank 2)**	0	0 to 32765
Action on error	20	**Disable or enable setting for duplicated output**	Disable	Disable/enable
	21	**Operation settings when MEWNET-TR communication error occurs Available PLC: FP10SH**	Stop	Stop/continuation
		Operation setting when I/O error occurs Available PLC: FP2SH/ FP2	Stop	Stop/continuation
	22	**Operation settings when an intelligent unit error occurs**	Stop	Stop/continuation
	23	**Operation settings when an I/O verification error occurs**	Stop	Stop/continuation
	24	**Operation settings when a system watching dog timer error occurs Available PLC: FP2SH/ FP10SH**	Stop	Stop/continuation Set the time-out time for watching dog timer with system register 30.
	25	**Operation settings when connection time error occurs in the remote slave station Available PLC: FP2SH**	Stop	Stop/continuation
	26	**Operation settings when an operation error occurs**	Stop	Stop/continuation
	27	**Operation settings when communication error occurs in the MEWNET-F system**	Stop	Stop/continuation
	28	**Operation settings when error occurs in the slave station of the MEWNET-F system**	Stop	Stop/continuation

FP2/FP2SH/FP10SH

Item	Add-ress	Name	Default value	Descriptions
Time setting for FP2SH/ FP10SH	29	**Operation time setting for communication processing**	240 μs	0 to 52428 μs If the response of the connected programmable display is show, please make the value bigger.
	30	**Time-out time setting of system watching dog timer**	100 ms	0.4 to 640 ms
	31	**Multi-frame communication time settings in the computer link and communication time setting for data sending buffer**	6500 ms	10 to 81917.5 ms
	32	**Time-out time setting for the F145 (SEND)/P145 (PSEND), F146 (RECV)/P146 (PRECV), F152 (RMRD)/P152 (PRMRD) and F153 (RMWT)/P153 (PRMWT) instructions**	10000 ms	10 to 81917.5 ms
	33	**Effective time setting for monitoring**	163837.5 ms	2500 to 163837.5 ms
	34	**Constant scan time setting**	0 ms: Normal scan	0 to 640 ms: Scans once each specified time interval. Set “0”: Normal scan Setting time can be obtained using the formula “Set time” = “Set value” x 0.1 (ms)
Time setting for FP2	31	**Multi-frame communication time settings in the computer link**	6500 ms	10.0 to 8190.0 ms
	32	**Time-out time setting for the F145 (SEND)/P145 (PSEND), F146 (RECV)/P146 (PRECV), F152 (RMRD)/P152 (PRMRD) and F153 (RMWT)/P153 (PRMWT) instructions**	2000 ms	10.0 to 8190.0 ms
	33	**Program block-editing time in the RUN mode**	10000 μs	800.0 to 52428.0 μs
	34	**Constant scan time setting**	0 : Normal scan	0 to 640 ms: Scans once each specified time interval. Set “0”: Normal scan

FP2/FP2SH/FP10SH

Item	Add-ress	Name		Default value	Descriptions
Remote I/O control	**25**	**Operation settings when connection time error occurs in the remote slave station Available PLC: FP2SH**		Stop	Stop/continuation
	35	**Operation mode setting when the MEWNET-F system is used**		Enabled (wait for connec-tion)	Enabled: CPU starts operation after all the salve stations are recognized. Disabled: CPU starts operation without waiting for slave station connections. Only effective when registering remote I/O allocation.
	36	**I/O data updating mode settings for MEWNET-F system**		Scan syn-chronous	Scan asynchronous mode/ Scan synchronous mode
PC link 0 setting	**40**	**PC link 0 settings for MEWNET-W/-P link system Available PLC: MEWNET-W: FP10SH, FP2, FP2SH MEWNET-P: FP10SH**	**Size of link relays used for commu-nication**	0	0 to 64 words
	41		**Size of link data registers used for commu-nication**	0	0 to 128 words
	42		**Send area starting address of link relay**	0	0 to 63
	43		**Size of link relays used for send area**	0	0 to 64 words
	44		**Send area starting address of link data register**	0	0 to 127
	45		**Size of link data registers used for send area**	0	0 to 127 words
	46	**PC link 0 and 1 allocation setting for MEWNET-W/-P link system Available PLC: MEWNET-W: FP10SH, FP2, FP2SH MEWNET-P: FP10SH**		Normal allocation	Normal allocation: (PC link 0 for the link unit with a smaller slot number and PC link 1 for one with a larger slot number) Reverse allocation: (PC link 1 for the link unit with a smaller slot number and PC link 0 for one with a larger slot number)

FP2/FP2SH/FP10SH

Item	Add-ress	Name		Default value	Descriptions
MEWNET -H setting	**49**	**Processing capacity setting for PC link of MEWNET-H link system Available PLC: FP10SH**		4 (1024 bytes per scan)	0: All data in a scan 1 to 65535: Setting processing capacity per scan can be obtained using the formula “Capacity” = “Set value” x 256 bytes
PC link 1 setting	**50**	**PC link 1 settings for MEWNET-W/-P link system Available PLC: MEWNET-W: FP10SH, FP2, FP2SH MEWNET-P: FP10SH**	**Size of link relays used for commu-nication**	0	0 to 64 words
	51		**Size of link data registers used for commu-nication**	0	0 to 128 words
	52		**Send area starting address of link relay**	64	64 to 127
	53		**Size of link relays used for send area**	0	0 to 64 words
	54		**Send area starting address of link data register**	128	128 to 255
	55		**Size of link data registers used for send area**	0	0 to 127 words
Tool port setting	**410**	**Unit number setting for tool port Available PLC: FP2/ FP2SH**		1	1 to 99 (unit No. 1 to 99)
	411	**Communication format setting for tool port Available PLC: FP2/ FP2SH**		Commu-nication format (character bit): 8 bits, Modem communi-cation: Disabled	Character bits: 7 bits/8bits Modem communication: Enabled/Disabled When connecting a modem, set the unit number to 1 with system register 410.

FP2/FP2SH/FP10SH

<table>
<tr><th>Item</th><th>Add-ress</th><th>Name</th><th>Default value</th><th colspan="2">Descriptions</th></tr>
<tr><td>Tool port setting</td><td>414</td><td>Baud rate setting for the tool port</td><td>19200 bps</td><td colspan="2">In the FP10SH, when the dip switch SW1 on the CPU is off, the baud rate setting is effective.
In the FP2/FP2SH, when the dip switch SW1 on the rear of the CPU is off, the baud rate setting is effective.
19200 bps 19200 bps
1200 bps 38400 bps
2400 bps 57600 bps
4800 bps 115200 bps
9600 bps</td></tr>
<tr><td rowspan="6">COM port setting</td><td rowspan="2">412</td><td rowspan="2">Communication method setting for COM port</td><td>FP2: Not used</td><td colspan="2" rowspan="2">UNUSED: COM port is not used.
COMPUTER LINK: computer link mode (when connecting C-NET)
GENERAL: serial data communication mode</td></tr>
<tr><td>FP2SH/ FP10SH: Computer link</td></tr>
<tr><td>413</td><td>Communication format setting (Common setting for both computer link and serial data communication)
When used for computer link, the start and end code settings of format for MEWTOCOL-COM will not be effective.
Available PLC: FP2/ FP2SH</td><td>Character bit: 8 bits,
Parity chk: “With, odd”
Stop bit: 1 bit,
End code: CR,
Start code: NO STX</td><td colspan="2">Character bit: 7 bits/8 bits
Paritch chk: non/with odd/with even
Stop bit: 1 bit/2 bits
End code: CR/CR+LF/NON/ETX
Start code: NO STX/STX</td></tr>
<tr><td>414</td><td>Baud rate setting for the COM port
Available PLC: FP2/ FP2SH</td><td>19200 bps</td><td colspan="2">19200 bps 19200 bps
1200 bps 38400 bps
2400 bps 57600 bps
4800 bps 115200 bps
9600 bps</td></tr>
<tr><td>415</td><td>Unit number setting for COM port
Available PLC: FP2/ FP2SH</td><td>1</td><td colspan="2">1 to 99 (unit No. 1 to 99)</td></tr>
<tr><td>416</td><td>Modem compatibility setting for COM port
Available PLC: FP2/ FP2SH</td><td>Modem disabled</td><td colspan="2">Modem enabled/Modem disabled
When connecting a modem, set the unit number to 1 with system register 415.</td></tr>
<tr><td rowspan="2">General commu-nication setting</td><td>417</td><td>Starting address setting for received buffer of serial data communication mode (data register number)</td><td>0</td><td>FP2SH/FP10SH: 0 to 10240
FP2: 0 to 5999</td><td rowspan="2">For details about its usage, refer to the F144 (TRNS)/ P144 (PTRNS) instructions.</td></tr>
<tr><td>418</td><td>Capacity setting for received buffer of serial data communication mode (word number)</td><td>1024</td><td>0 to 1024</td></tr>
</table>

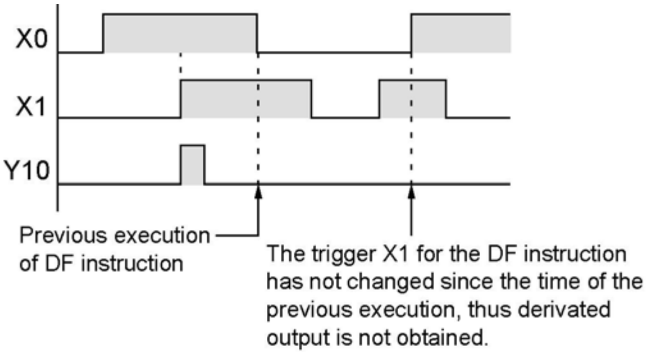
Operation of DF instruction between MC and MCE instructions

When a leading edge detection instruction (DF instruction) is used with the MC and MCE instructions, the derivative output may change as follows depending on the trigger of MC instruction and input timing of DF instruction. Take care regarding this point.

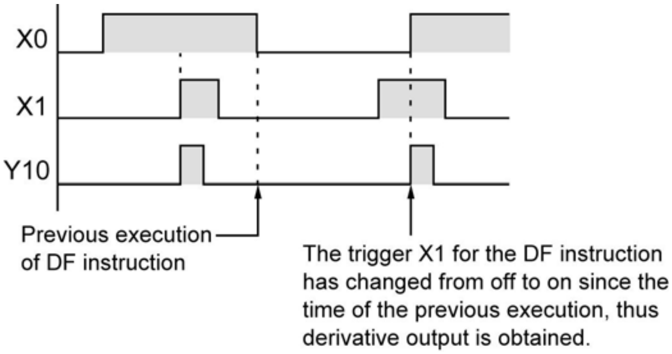
Example 1:

When system register 4 sets 0 (conventional)

Time chart 1

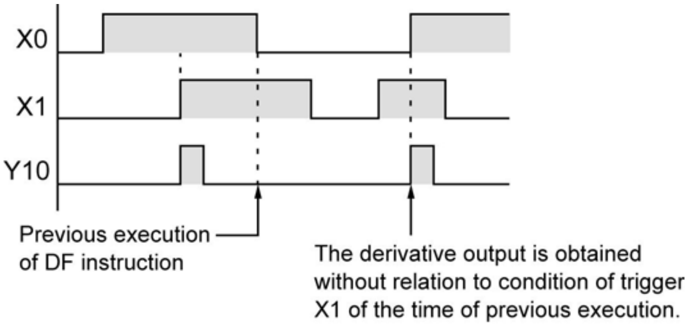


Time chart 2

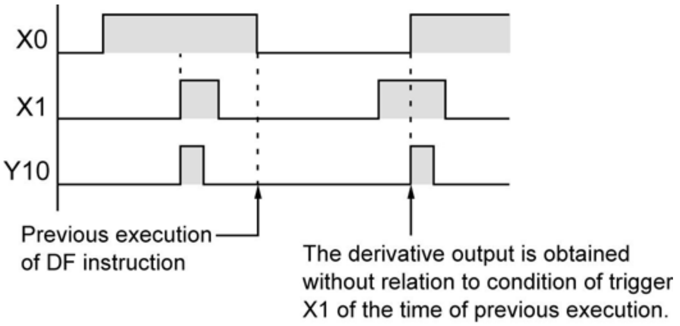


Example 2:
When system register 4 sets 1 (new)

Time chart 1



Time chart 2

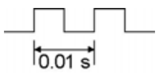
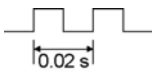
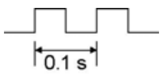
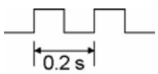
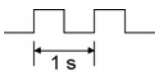


5.1.17 Table of Special Internal Relays for FP1/FP-M/FP2/FP2SH/FP10SH/FP3

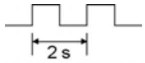
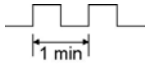
FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name	Description
R9000	**Self-diagnostic error flag (Available PLC: All types)**	Turns on when a self-diagnostic error occurs. The self-diagnostic error code is stored in: - FP1/FP-M/FP3: DT9000 - FP2/FP2SH/FP10SH: DT90000
R9001	**Not used**	-
R9002	**MEWNET-TR master error flag (Available PLC: FP3/ FP10SH)**	Turns on when a communication error occurs in the MEWNET-TR master unit or MEWNET-TR network. The slot, where the erroneous MEWNET-TR master unit is installed, can be checked using: - FP3: DT9002 and DT9003 - FP10SH: DT90002, DT90003
	I/O error flag (Available PLC: FP2/ FP2SH)	Turns on when the error occurs in the I/O unit. The slot number of the unit where the error was occurred is stored in DT90002, DT90003.
R9003	**Intelligent unit error flag**	Turns on when an error occurs in an intelligent unit. The slot number, where the erroneous intelligent unit is installed is stored in: - FP3: DT9006 and DT9007 - FP2/FP2SH/FP10SH: DT90006, DT90007
R9004	**I/O verification error flag**	Turns on when an I/O verification error occurs. The slot number of the I/O unit where the verification error was occurred is stored in: - FP3: DT9010 and DT9011 -FP2/FP2SH/FP10SH: DT90010, DT90011
R9005	**Backup battery error flag (non-hold) (Available PLC: FP-M C20, C32/FP1 C24, C40, C56, C72/FP2/ FP2SH/FP3/FP10SH)**	Turns on for an instant when a backup battery error occurs.
R9006	**Backup battery error flag (hold) (Available PLC: FP-M C20, C32/FP1 C24, C40, C56, C72/FP2/ FP2SH/FP3/FP10SH)**	Turns on and keeps the on state when a backup battery error occurs. To reset R9006, - turn the power to off and then turn it on, - initialize, after removing the cause of error.
R9007	**Operation error flag (hold)**	Turns on and keeps the on state when an operation error occurs. The address where the error occurred is stored in: - FP1/FP-M/FP3: DT9017 – FP2/FP2SH/FP10SH: DT90017 (Indicates the first operation error which occurred).
R9008	**Operation error flag (non-hold) (Available PLC: FP1/FP-M/FP2/FP2SH/ FP10SH)**	Turns on for an instant when an operation error occurs. The address where the operation error occurred is stored in: - FP1/FP-M/FP3: DT9018 -FP2/FP2SH/FP10SH: DT90018 The contents change each time a new error occurs.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name	Description
R9009	**Carry flag**	Turns on for an instant, - when an overflow or underflow occurs. - when “1” is set by one of the shift instructions.
R900A	**> Flag**	Turns on for an instant when the compared results become larger in the “F60 (CMP)/P60 (PCMP), F61(DCMP)P61(PDCMP),F62 (WIN)/P62 (PWIN) or F63 (DWIN)/P63 (PDWIN) comparison instructions.”
R900B	**= Flag**	Turns on for an instant, - when the compared results are equal in the comparison instructions. - when the calculated results become 0 in the arithmetic instructions.
R900C	**< Flag**	Turns on for an instant when the compared results become smaller in the “F60 (CMP)/P60 (PCMP), F61(DCMP)P61(PDCMP), F62 (WIN)/P62 ,(PWIN) or F63 (DWIN)/P63 (PDWIN) comparison instructions.”
R900D	**Auxiliary timer contact (Available PLC: FP-M C20, C32/FP1 C56, C72/FP2/FP2SH/FP3/ FP10SH)**	Turns on when the set time elapses (set value reaches 0) in the timing operation of the F137(STMR)/F183(DSTM) auxiliary timer instruction. Available PLC for F183 (DSTM) instruction: FP2/FP2SH/ FP10SH The R900D turns off when the trigger for auxiliary timer instruction turns off.
R900E	**Tool port error flag (Available PLC: FP1/ FP-M/FP2SH/FP10SH)**	Turns on when communication error at tool port is occurred.
R900F	**Constant scan error flag**	Turns on when the scan time exceeds the time specified in system register 34 during constant scan execution.
R9010	**Always on relay**	Always on.
R9011	**Always off relay**	Always off.
R9012	**Scan pulse relay**	Turns on and off alternately at each scan.
R9013	**Initial on pulse relay**	Turns on only at the first scan in the operation. Turns off from the second scan and maintains the off state.
R9014	**Initial off pulse relay**	Turns off only at the first scan in the operation. Turns on from the second scan andmaintains the on state.
R9015	**Step ladder initial on pulse relay**	Turns on for an instant only in the first scan of the process the moment step ladder process is opened.
R9016, R9017	**Not used**	-
R9018	**0.01 s clock pulse relay**	Repeats on/off operations in 0.01 s cycles. 
R9019	**0.02 s clock pulse relay**	Repeats on/off operations in 0.02 s cycles. 
R901A	**0.1 s clock pulse relay**	Repeats on/off operations in 0.1 s cycles. 
R901B	**0.2 s clock pulse relay**	Repeats on/off operations in 0.2 s cycles. 
R901C	**1 s clock pulse relay**	Repeats on/off operations in 1 s cycles. 

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name	Description
R901D	**2 s clock pulse relay**	Repeats on/off operations in 2 s cycles. 
R901E	**1 min clock pulse relay**	Repeats on/off operations in 1 min cycles. 
R901F	**Not used**	-
R9020	**RUN mode flag**	Turns off while the mode selector is set to PROG. Turns on while the mode selector is set to RUN.
R9021	**Test RUN mode flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on while the initialize/test switch of the CPU is set to TEST and mode selector is set to RUN. (test run operation start) Turns off during the normal RUN mode.
R9022	**Break flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on while the BRK instruction is executing or the step run is executing.
R9023	**Break enable flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on while the BRK instruction is enabled in the test RUN mode.
R9024	**Output update enable flag in the test RUN mode (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on while the output update is enabled in the test RUN mode.
R9025	**Single instruction flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on while the single instruction execution is selected in the test RUN mode.
R9026	**Message flag (Available PLC: FP-M C20, C32/FP1 C24, C40, C56, C72/FP2/ FP2SH/FP3/FP10SH)**	Turns on while the F149 (MSG)/P149 (PMSG) instruction is executed.
R9027	**Remote mode flag**	Turns on while the mode selector is set to REMOTE.
R9028	**Break clear flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on when the break operation is cleared.
R9029	**Forcing flag**	Turns on during forced on/off operation for I/O relay and timer/counter contacts.
R902A	**External interrupt enable flag (Available PLC: FP-M/FP1 C24, C40, C56, C72/FP2/FP2SH/FP3/ FP10SH)**	Turns on while the external interrupt trigger is enabled by the ICTL instruction.
	Interrupt flag (Available PLC: FP2)	Turns on while the periodical interrupt is executed by the ICTL instruction.
R902B	**Interrupt error flag (Available PLC: FP-M/FP1 C24, C40, C56, C72/FP2/FP2SH/ FP3/FP10SH)**	Turns on when an interrupt error occurs.
R902C	**Sampling point flag**	Turns off during instructed sampling. Turns on while sampling is triggered by the periodical interrupt.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name	Description
R902D	**Sampling trace end flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on when the sampling trace ends.
R902E	**Sampling trigger flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on when the sampling trace trigger of the F156 (STRG)/P156 (PSTGR) instruction is turned on.
R902F	**Sampling enable flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on when the starting point of sampling is specified.
R9030	**F145 (SEND)/P145 (PSEND) and F146 (RECV)/P146 (PRECV) instruction executing flag**	Monitors if CPU is in the F145 (SEND)/P145 (PSEND) and F146 (RECV)/P146 (PRECV) instructions executable condition as follows: - off: None of the above mentioned instructions can be executed. - on: One of the above mentioned instructions can be executed.
R9031	**F145 (SEND)/P145 (PSEND) and F146 (RECV)/P146 (PRECV) instruction end flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Monitors if an abnormality has been detected during the execution of the F145 (SEND)/P145 (PSEND) and F146 (RECV)/P146 (PRECV) instructions as follows: - off: No abnormality detected. - on: An abnormality detected. (communication error) The error code is stored in: - FP3: DT9039 - FP2/FP10SH: DT90039
R9032	**COM port mode flag (Available PLC: FP-M C20C, C32C/FP1 C24C, C40C, C56C, C72C/FP2/FP2SH/ FP10SH)**	Monitors the mode of the COM port as: - on: Serial data communication mode - off: Computer link mode
R9033	**F147 (PR) instruction flag (Available PLC: FP-M C20, C32/FP1 C24, C40, C56,C72/ FP2/FP2SH/FP3/ FP10SH)**	Turns on while a F147 (PR) instruction is executed. Turns off when a F147 (PR) instruction is not executed.
R9034	**Editing in RUN mode flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Turns on while editing a program in the RUN mode.
R9035	**F152 (RMRD)/P152 (PRMRD) and F153 (RMWT)/P153 (PRMWT) instruction execution flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Monitors if FP3/FP10SH is in the F152 (RMRD)/P152 (PRMRD) and F153 (RMWT)/P153 (PRMWT) instructions executable condition as follows: - off: None of the above mentioned instructions can be executed. - on: One of the above mentioned instructions can be executed.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name	Description
R9036	**F152 (RMRD)/P152 (PRMRD) and F153 (RMWT)/P153 (PRMWT) instruction end flag (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	Monitors if an abnormality has been detected during the execution of the F152 (RMRD)/P152 (PRMRD) and F153 (RMWT)/P153 (PRMWT) instructions as follows: - off: No abnormality detected. - on: An abnormality detected. (access error) The error code is stored in: - FP3: DT9036 - FP2/FP2SH/FP10SH: DT90036
	I/O link error flag (Available PLC: FP-M C20, C23/FP1)	Turns on when the erroro occurs using the I/O link function.
R9037	**COM port communica-tion error flag (Available PLC: FP-M C20C, C32C/FP1 C24C, C40C, C56C, C72C/FP2/FP2SH/ FP10SH)**	Turns on when the serial data communication error occurs using COM port. Turns off when data is being sent by the F144 (TRNS) instruction.
R9038	**COM port receive flag (Available PLC: FP-M C20C, C32C/FP1 C24C, C40C, C56C, C72C/FP2/FP2SH/ FP10SH)**	Turns on when the end code is received during the serial data communicating.
R9039	**COM port send flag (Available PLC: FP-M C20C, C32C/FP1 C24C, C40C, C56C, C72C/FP2/FP2SH/ FP10SH)**	Turns on while data is not send during the serial data communicating. Turns off while data is being sent during the serial data communicating.
R903A	**High-speed counter control flag (ch0) (Available PLC: FP-M C20, C32/FP1)**	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903B	**Cam control flag (Available PLC: FP-M/ FP1)**	Turns on while the cam control instruction F165 (CAMO) is executed.
	High-speed counter control flag (ch1)	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903C	**High-speed counter control flag (ch2)**	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903D	**High-speed counter control flag (ch3)**	Turns on while the high-speed counter instructions F166(HC1S) to F170(PWM) are executed.
R903E	**Not used**	-
R903F	**Not used**	-
R9040	**Error alarm (0 to 2047)**	Turns on while the error alarm relay (E0 to E2047) acts. Turns off when the all error alarm relay turns off.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name	Description
R9050	**MEWNET-W/-P link transmission error flag [W/P LINK 1] for FP3/ FP10SH [W LINK 1] for FP2/FP2SH**	When using MEWNET-W link unit or MEWNET-P link unit: - turns on when transmission error occurs at link 1. - turns on when there is an error in the link area settings.
R9051	**MEWNET-W/-P link transmission error flag [W/P LINK 2] for FP3/ FP10SH [W LINK 2] for FP2/FP2SH**	When using MEWNET-W link unit or MEWNET-P link unit: - turns on when transmission error occurs at link 2. - turns on when there is an error in the link area settings.
R9052	**MEWNET-W/-P link transmission error flag [W/P LINK 3] for FP3/ FP10SH [W LINK 3] for FP2/FP2SH**	When using MEWNET-W link unit or MEWNET-P link unit: - turns on when transmission error occurs at link 3. - turns on when there is an error in the link area settings.
R9053	**MEWNET-W/-P link transmission error flag [W/P LINK 4] for FP2/ FP10SH [W LINK 4] for FP2SH**	When using MEWNET-W link unit or MEWNET-P link unit: - turns on when transmission error occurs at link 4. - turns on when there is an error in the link area settings.
R9054	**MEWNET-W/-P link transmission error flag [W/P LINK 5] for FP2/ FP10SH [W LINK 5] for FP2SH**	When using MEWNET-W link unit or MEWNET-P link unit: - turns on when transmission error occurs at link 5. - turns on when there is an error in the link area settings.
R9055	**MEWNET-H link transmission error flag [H LINK 1]: for FP3/ FP10SH**	When using MEWNET -H link unit: - turns on when transmission error occurs at H link 1. - turns on when there is an error in the link area settings.
R9056	**MEWNET-H link transmission error flag [H LINK 2]: for FP3/ FP10SH**	When using MEWNET -H link unit: - turns on when transmission error occurs at H link 2. - turns on when there is an error in the link area settings.
R9057	**MEWNET-H link transmission error flag [H LINK 3]: for FP3/ FP10SH**	When using MEWNET-H link unit: - turns on when transmission error occurs at H link 3. - turns on when there is an error in the link area settings.
R9058	**Remote I/O transmis-sion error flag (master 1) (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	When using MEWNET -F (remote I/O) system: - turns on when transmission error occurs on master 1. - turns on when there is an error in the settings.
R9059	**Remote I/O transmis-sion error flag (master 2) (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	When using MEWNET -F (remote I/O) system: - turns on when transmission error occurs on master 2. - turns on when there is an error in the settings.
R905A	**Remote I/O transmission error flag (master 3) (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	When using MEWNET -F (remote I/O) system: - turns on when transmission error occurs on master 3. - turns on when there is an error in the settings.
R905B	**Remote I/O transmis-sion error flag (master 4) (Available PLC: FP2/ FP2SH/FP3/FP10SH)**	When using MEWNET -F (remote I/O) system: - turns on when transmission error occurs on master 4. - turns on when there is an error in the settings.
R905C to R905F	**Not used**	-

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name		Description
R9060	**MEWNET-W/ -P PC link transmission assurance relay [for PC link 0 (W/P)]**	**Unit No.1**	Turns on when Unit No. 1 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9061		**Unit No.2**	Turns on when Unit No. 2 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9062		**Unit No.3**	Turns on when Unit No. 3 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9063		**Unit No.4**	Turns on when Unit No. 4 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9064		**Unit No.5**	Turns on when Unit No. 5 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9065		**Unit No.6**	Turns on when Unit No. 6 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9066		**Unit No.7**	Turns on when Unit No. 7 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9067		**Unit No.8**	Turns on when Unit No. 8 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9068		**Unit No.9**	Turns on when Unit No. 9 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9069		**Unit No.10**	Turns on when Unit No. 10 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R906A		**Unit No.11**	Turns on when Unit No. 11 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R906B		**Unit No.12**	Turns on when Unit No. 12 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R906C		**Unit No.13**	Turns on when Unit No. 13 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R906D		**Unit No.14**	Turns on when Unit No. 14 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R906E		**Unit No.15**	Turns on when Unit No. 15 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R906F		**Unit No.16**	Turns on when Unit No. 16 is communicating properly I n the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.

Note) When the system registers 46=K0, PC link 0 for the link unit with a smaller slot No. and PC link 1 for one with a larger slot No.
When the system registers 46=K1, PC link 1 for the link unit with a smaller slot No. and PC link 0 for one with a larger slot No.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name		Description
R9070	**MEWNET-W/-P PC link operation mode relay [for PC link 0 (W/P)]**	**Unit No.1**	Turns on when unit No. 1 is in the RUN mode. Turns off when unit No. 1 is in the PROG. mode.
R9071		**Unit No.2**	Turns on when unit No. 2 is in the RUN mode. Turns off when unit No. 2 is in the PROG. mode.
R9072		**Unit No.3**	Turns on when unit No. 3 is in the RUN mode. Turns off when unit No. 3 is in the PROG. mode.
R9073		**Unit No.4**	Turns on when unit No. 4 is in the RUN mode. Turns off when unit No. 4 is in the PROG. mode.
R9074		**Unit No.5**	Turns on when unit No. 5 is in the RUN mode. Turns off when unit No. 5 is in the PROG. mode.
R9075		**Unit No.6**	Turns on when unit No. 6 is in the RUN mode. Turns off when unit No. 6 is in the PROG. mode.
R9076		**Unit No.7**	Turns on when unit No. 7 is in the RUN mode. Turns off when unit No. 7 is in the PROG. mode.
R9077		**Unit No.8**	Turns on when unit No. 8 is in the RUN mode. Turns off when unit No. 8 is in the PROG. mode.
R9078		**Unit No.9**	Turns on when unit No. 9 is in the RUN mode. Turns off when unit No. 9 is in the PROG. mode.
R9079		**Unit No.10**	Turns on when unit No. 10 is in the RUN mode. Turns off when unit No. 10 is in the PROG. mode.
R907A		**Unit No.11**	Turns on when unit No. 11 is in the RUN mode. Turns off when unit No. 11 is in the PROG. mode.
R907B		**Unit No.12**	Turns on when unit No. 12 is in the RUN mode. Turns off when unit No. 12 is in the PROG. mode.
R907C		**Unit No.13**	Turns on when unit No. 13 is in the RUN mode. Turns off when unit No. 13 is in the PROG. mode.
R907D		**Unit No.14**	Turns on when unit No. 14 is in the RUN mode. Turns off when unit No. 14 is in the PROG. mode.
R907E		**Unit No.15**	Turns on when unit No. 15 is in the RUN mode. Turns off when unit No. 15 is in the PROG. mode.
R907F		**Unit No.16**	Turns on when unit No. 16 is in the RUN mode. Turns off when unit No. 16 is in the PROG. mode.

Note) When the system registers 46=K0, PC link 0 for the link unit with a smaller slot No. and PC link 1 for one with a larger slot No.
When the system registers 46=K1, PC link 1 for the link unit with a smaller slot No. and PC link 0 for one with a larger slot No.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name		Description
R9080	**MEWNET-W/-P PC link transmission assurance relay [for PC link 1 (W/P)]**	**Unit No.1**	Turns on when unit No. 1 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9081		**Unit No.2**	Turns on when unit No. 2 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9082		**Unit No.3**	Turns on when unit No. 3 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9083		**Unit No.4**	Turns on when unit No. 4 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9084		**Unit No.5**	Turns on when unit No. 5 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9085		**Unit No.6**	Turns on when unit No. 6 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9086		**Unit No.7**	Turns on when unit No. 7 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9087		**Unit No.8**	Turns on when unit No. 8 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9088		**Unit No.9**	Turns on when unit No. 9 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R9089		**Unit No.10**	Turns on when unit No. 10 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R908A		**Unit No.11**	Turns on when unit No. 11 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R908B		**Unit No.12**	Turns on when unit No. 12 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R908C		**Unit No.13**	Turns on when unit No. 13 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R908D		**Unit No.14**	Turns on when unit No. 14 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R908E		**Unit No.15**	Turns on when unit No. 15 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.
R908F		**Unit No.16**	Turns on when unit No. 16 is communicating properly in the PC link mode. Turns off when operation is stopped, when an error occurs, or when not in the PC link mode.

Note) When the system registers 46=K0, PC link 0 for the link unit with a smaller slot No. and PC link 1 for one with a larger slot No.
When the system registers 46=K1, PC link 1 for the link unit with a smaller slot No. and PC link 0 for one with a larger slot No.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name		Description
R9090	**MEWNET-W/-P PC link operation mode relay [for PC link 1 (W/P)]**	**Unit No.1**	Turns on when unit No. 1 is in the RUN mode. Turns off when unit No. 1 is in the PROG. mode.
R9091		**Unit No.2**	Turns on when unit No. 2 is in the RUN mode. Turns off when unit No. 2 is in the PROG. mode.
R9092		**Unit No.3**	Turns on when unit No. 3 is in the RUN mode. Turns off when unit No. 3 is in the PROG. mode.
R9093		**Unit No.4**	Turns on when unit No. 4 is in the RUN mode. Turns off when unit No. 4 is in the PROG. mode.
R9094		**Unit No.5**	Turns on when unit No. 5 is in the RUN mode. Turns off when unit No. 5 is in the PROG. mode.
R9095		**Unit No.6**	Turns on when unit No. 6 is in the RUN mode. Turns off when unit No. 6 is in the PROG. mode.
R9096		**Unit No.7**	Turns on when unit No. 7 is in the RUN mode. Turns off when unit No. 7 is in the PROG. mode.
R9097		**Unit No.8**	Turns on when unit No. 8 is in the RUN mode. Turns off when unit No. 8 is in the PROG. mode.
R9098		**Unit No.9**	Turns on when unit No. 9 is in the RUN mode. Turns off when unit No. 9 is in the PROG. mode.
R9099		**Unit No.10**	Turns on when unit No. 10 is in the RUN mode. Turns off when unit No. 10 is in the PROG. mode.
R909A		**Unit No.11**	Turns on when unit No. 11 is in the RUN mode. Turns off when unit No. 11 is in the PROG. mode.
R909B		**Unit No.12**	Turns on when unit No. 12 is in the RUN mode. Turns off when unit No. 12 is in the PROG. mode.
R909C		**Unit No.13**	Turns on when unit No. 13 is in the RUN mode. Turns off when unit No. 13 is in the PROG. mode.
R909D		**Unit No.14**	Turns on when unit No. 14 is in the RUN mode. Turns off when unit No. 14 is in the PROG. mode.
R909E		**Unit No.15**	Turns on when unit No. 15 is in the RUN mode. Turns off when unit No. 15 is in the PROG. mode.
R909F		**Unit No.16**	Turns on when unit No. 16 is in the RUN mode. Turns off when unit No. 16 is in the PROG. mode.
R9100	**IC memory card installation flag (Available PLC: FP2SH/ FP10SH)**		Monitors whether the IC memory card is installed or not: - on: IC memory card is installed. - off: IC memory card is not installed.
R9101 (*Note)	**IC memory card backup battery flag 1 (Available PLC: FP2SH/ FP10SH)**		Monitors the voltage drop condition for the IC memory card as: - on: Data in the IC memory card cannot be guaranteed. - off: Data in the IC memory card can be maintained.
R9102 (*Note)	**IC memory card backup battery flag 2 (Available PLC: FP2SH/ FP10SH)**		Monitors the voltage drop condition for the IC memory card as: - on: Battery replacement is required. - off: Battery replacement is not required.

Note) When the system registers 46=K0, PC link 0 for the link unit with a smaller slot No. and PC link 1 for one with a larger slot No.
When the system registers 46=K1, PC link 1 for the link unit with a smaller slot No. and PC link 0 for one with a larger slot No.

FP1/FP-M/FP2/FP2SH/FP10SH/FP3

Address	Name	Description
R9103	**IC memory card protect switch flag (Available PLC: FP2SH/ FP10SH)**	Monitors the protective condition of the IC memory card as: - on: The protect switch is not in the write-protected (WP) position. - off: The protect switch is in the write-protected (WP) position.
R9104	**IC memory card access switch flag (Available PLC: FP2SH/ FP10SH)**	Monitors the condition of the IC memory card access enables switch as: - on (access enabled): The access enable switch is in the on position. – off (access disabled): The access enable switch is in the off position.
R9105 Through R910F	**Note used**	-

Note) The IC memory card backup battery condition can be judged using special internal relays R9101 and R9102 as follows:

R9101	R9102	IC memory card condition
OFF	**OFF**	Not battery replacement required.
ON	**OFF**	Replace backup battery. The data in the IC memory card is maintained.
ON	**ON**	The data in the IC memory card cannot be maintained. Replace backup battery.

5.1.18 Special Data Registers for FP2/FP2SH/FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read-ing</th><th rowspan="2">Writ-ing</th></tr>
<tr><th>FP3</th><th>FP2/ FP2SH FP10SH</th></tr>
<tr><td>DT9000</td><td>DT90000</td><td>Self-diagnostic error code</td><td>The self-diagnostic error code is stored here when a self-diagnostic error occurs. Monitor the error code using decimal display.</td><td>A</td><td>N/A</td></tr>
<tr><td>DT9001</td><td>DT90001</td><td>Not used</td><td>-</td><td>N/A</td><td>N/A</td></tr>
<tr><td rowspan="2">DT9002</td><td rowspan="2">DT90002</td><td>Communication error of MEWNET-TR master unit (slot No. 0 to 15) (Available PLC: FP3/FP10SH)</td><td rowspan="4">The slot number, where an erroneous unit is installed, can be monitored here. “1” (on) is set in the bit position corresponding to the slot number when one of the errors below id detected.
Communication error MEWNET-TR master unit
When a communication error occurs at the MEWNET-TR master unit, the bit corresponding to the slot no. of the unit will be set on “1”. Monitor using binary display.
(1: erroneous MEWNET-TR master unit, 0: normal)
Position of abnormal I/O slot
When an error occurs at an I/O unit, the bit corresponding to the slot of the unit will be set on “1”. Monitor sing binary display.
(1: error, 0: normal)
<table><tr><td>Bit position</td><td>15 . . 21</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>Slot No.</td><td>15 . . 21</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>DT9002/DT90002</td><td></td><td></td><td></td><td></td></tr></table>
<table><tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>Slot No.</td><td>31 . . 28</td><td>27 . . 24</td><td>23 . . 20</td><td>19 . . 16</td></tr><tr><td>DT9003/DT90003</td><td></td><td></td><td></td><td></td></tr></table></td><td rowspan="4">A</td><td rowspan="4">N/A</td></tr>
<tr><td>Position of abnormal I/O slot (slot No. 0 to 15) (Available PLC: FP2/FP2SH)</td></tr>
<tr><td rowspan="2">DT9003</td><td rowspan="2">DT90003</td><td>Communication error of MEWNET-TR master unit (slot No. 16 to 31) (Available PLC: FP3/FP10SH)</td></tr>
<tr><td>Position of abnormal I/O slot (slot No. 16 to 31) (Available PLC: FP2/FP2SH)</td></tr>
</table>

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read-
ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td>DT9006</td><td>DT90006</td><td>Abnormal intelligent unit (slot No. 0 to 15)</td><td rowspan="2">When an error condition is detected in an intelligent unit, the bit corresponding to the slot of the unit will be set to on. Monitor using binary display.
(1: abnormal intelligent unit, 0: normal intelligent unit)
<table><tr><td>Bit position</td><td>15 . . 21</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>Slot No.</td><td>15 . . 21</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>DT9006/DT90006</td><td></td><td></td><td></td><td></td></tr></table><table><tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>Slot No.</td><td>31 . . 28</td><td>27 . . 24</td><td>23 . . 20</td><td>19 . . 16</td></tr><tr><td>DT9007/DT90007</td><td></td><td></td><td></td><td></td></tr></table></td><td rowspan="9">A</td><td rowspan="9">N/A</td></tr>
<tr><td>DT9007</td><td>DT90007</td><td>Abnormal intelligent unit (slot No. 16 to 31)</td></tr>
<tr><td>DT9010</td><td>DT90010</td><td>I/O verify error unit (slot No. 0 to 15)</td><td rowspan="2">When the state of installation of an I/O unit has changed since the power was turned on, the bit corresponding to the slot of the unit will be set to on. Monitor using binary display.
(1: error, 0: normal)
<table><tr><td>Bit position</td><td>15 . . 21</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>Slot No.</td><td>15 . . 21</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>DT9010/DT90010</td><td></td><td></td><td></td><td></td></tr></table><table><tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>Slot No.</td><td>31 . . 28</td><td>27 . . 24</td><td>23 . . 20</td><td>19 . . 16</td></tr><tr><td>DT9011/DT90011</td><td></td><td></td><td></td><td></td></tr></table></td></tr>
<tr><td>DT9011</td><td>DT90011</td><td>I/O verify error unit (slot No. 16 to 31)</td></tr>
<tr><td>DT9014</td><td>DT90014</td><td>Auxiliary register for operation</td><td>One shift-out hexadecimal digit is stored in bit positions 0 to 3 when F105 (BSR)/P105 (PBSR) or f106 (BSL)/P106 (PBSL) instruction is executed.</td></tr>
<tr><td>DT9015</td><td>DT90015</td><td rowspan="2">Auxiliary register for operation</td><td rowspan="2">The divided remainder (16-bit) is stored in DT9015/DT90015 when F32 (%)/P32 (P%) or F52(B%)/P52 (PB%) instruction is executed.
The divided remainder (32-bit) is stored in DT9015 and DT9016/DT90015 and DT90016 when F33 (D%)/P33 (PD%) or F53(DB%)/P53 (PDB%) instruction is executed.</td></tr>
<tr><td>DT9016</td><td>DT90016</td></tr>
<tr><td>DT9017</td><td>DT90017</td><td>Operation error address (hold)</td><td>After commencing operation, the address whre the first operation error occurred is stored. Monitor the address using decimal display.
(Reference: DT90257)</td></tr>
</table>

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read-
ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td>DT9018</td><td>DT90018</td><td>Operation error address (non-hold)</td><td>The address whre an operation error occurred is stored. Each time an error occurs, the new address overwrites the previous address. At the beginning of scan, the addressis 0. Monitor the address using decimal display.
(Reference: DT90258)</td><td rowspan="2">A</td><td rowspan="2">N/A</td></tr>
<tr><td>DT9019</td><td>DT90019</td><td>2.5 ms ring counter</td><td>The data stored here is increased by one very 2.5 ms (H0 to HFFFF)
Difference between the values of the two points (absolute value) x 2.5 ms = Elapsed time between the two points.</td></tr>
<tr><td>DT9020</td><td>-</td><td>Maximu value of program (Available PLC: FP3)</td><td>The last address of sequence program area set in system register 0 is stored.</td><td rowspan="6">N/A</td><td rowspan="6">N/A</td></tr>
<tr><td rowspan="2">-</td><td rowspan="2">DT90020</td><td>Display of program capacity (Available PLC: FP10SH)</td><td>The program capacity is stored in decimal.
Example:
K30: approx. 30 K steps
K60: approx. 60 K steps (with memory expansion)</td></tr>
<tr><td>Display of program capacity (Available PLC: FP2)</td><td>The program capacity is stored in decimal.
Example:
K16: approx. 16 K steps (K15870)
K32: approx. 32 K steps (with memory expansion)</td></tr>
<tr><td>DT9021 (*Note)</td><td>-</td><td>Maximum value of file register (Available PLC: FP3)</td><td>The maximum (last) address of the file registers available are stored in here.</td></tr>
<tr><td>-</td><td>DT90021 (*Note)</td><td>Maximum value of file register (Available PLC: FP2/FP10SH)</td><td>The maximum (last) address of the file registers available are stored in here.</td></tr>
</table>

Note) Used by the system.

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address FP3	Address FP2/ FP2SH FP10SH	Name	Descriptions		Read-ing	Writ-ing
DT9022	**DT90022**	**Scan time (current value)**	The current scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 **Example:** K50 indicates 5 ms.	Scan time display is only possible in RUN mode, and shows the operation cycle time. The maximum and minimum values are cleared when each the mode is switched between RUN mode and PROG. mode.	A	N/A
DT9023	**DT90023**	**Scan time (minimum value)**	The minimum scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 **Example:** K50 indicates 5 ms.		A	N/A
DT9024	**DT90024**	**Scan time (maximum value)**	The maximum scan time is stored here. Scan time is calculated using the formula: Scan time (ms) = stored data (decimal) x 0.1 **Example:** K125 indicates 12.5 ms.		A	N/A
DT9025 (*Note)	**DT90025 (*Note)**	**Mask condition monitoring register for interrupt unit initiated interrupts (INT 0 to 15) (*FP2: Not used)**	The mask conditions of interrupt unit initiated interrupts using ICTL instruciton can be monitored here. Monitor using binary display. Bit position: 15 . . 12 | 11 . . 8 | 7 . . 4 | 3 . . 0 INT program: 15 . . 12 | 11 . . 8 | 7 . . 4 | 3 . . 0 DT9025/DT90025 0: interrupt disabled (masked) 1: interrupt enabled (unmasked)		A	N/A
DT9026 (*Note)	**DT90026**	**Mask condition monitoring register for interrupt unit initiated interrupts (INT 16 to 23) (*FP2: Not used)**	The mask conditions of interrupt unit initiated interrupts using ICTL instruciton can be monitored here. Monitor using binary display. Bit position: 15 . . 12 | 11 . . 8 | 7 . . 4 | 3 . . 0 INT program: —— | —— | 23 . . 20 | 19 . . 16 DT9026/DT90026 0: interrupt disabled (masked) 1: interrupt enabled (unmasked)		A	N/A
DT9027 (*Note)	**DT90027**	**Periodical interrupt interval (INT24)**	The value set by ICTL instruction is stored. K0: periodical interrupt is not used. K1 to K3000: 10ms to 30s or 0.5ms to 1.5s		A	N/A

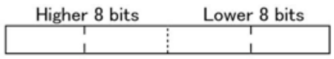
FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read-ing	Writ-ing
FP3	FP2/ FP2SH FP10SH				
DT9028 (*Note)	**DT90028 (*Note)**	**Sample trace interval**	The value registered using programming tool software is stored. - K0: sampling triggered by F155 (SMPL)/P155 (PSMPL) instruciton - K1 to K3000 (x 10ms): 10ms to 30s	A	N/A
DT9029 (*Note)	**DT90029 (*Note)**	**Break address**	The address (K constant) of a break in a test run is stored.		
DT9030 (*Note)	**DT90030 (*Note)**	**Message 0**	The contents of the specified message are stored in these special data registers when F149 (MSG)/P149 (PMSG) instruction is executed.		
DT9031 (*Note)	**DT90031 (*Note)**	**Message 1**			
DT9032 (*Note)	**DT90032 (*Note)**	**Message 2**			
DT9033 (*Note)	**DT90033 (*Note)**	**Message 3**			
DT9034 (*Note)	**DT90034 (*Note)**	**Message 4**			
DT9035 (*Note)	**DT90035 (*Note)**	**Message 5**			
DT9036	**DT90036**	**F152 (RMRD)/ P152 (PRMRD) and F153 (RMWT)/ P153 (PRMWT) instructions end code**	The error code is stored here if F152 (RMRD)/P152 (PRMRD) or F153 (RMWT)/P153 (PRMWT) instruction was executed abnormally. When the instruction was successfully executed, “0” is stored.	A	N/A
		Abnormal unit display	If an abnormal unit is installed to the backplane, the slot number of that unit will be stored. Monitor using decimal display.		
DT9037	**DT90037**	**Work 1 for F96 (SRC)/ P96 (PSRC) instructions**	The number of data that match the searched data is stored here when F96 (SRC)/P96 (PSRC) instruction is executed.	A	A
DT9038	**DT90038**	**Work 2 for F96 (SRC)/ P96 (PSRC) instructions**	The position of the first matching data, counting from the starting 16-bit area, is stored here when an F96 (SRC)/P96 (PSRC) instruction is executed.		
DT9039	**DT90039**	**F145 (SEND)/ P145 (PSEND) and F146 (RECV)/ P146 (PRECV) instructions end code**	The error code is stored here if F145 (SEND)/P145 (PSEND) or F146 (RECV)/ P146 (PRECV) instruction was executed abnormally. When the instruction was successfully executed, “0” is stored.	A	N/A

Note) Used by the system.

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read-ing</th><th rowspan="2">Writ-ing</th></tr>
<tr><th>FP3</th><th>FP2/ FP2SH FP10SH</th></tr>
<tr><td>DT9053</td><td>DT90053 (*Note)</td><td>Real-Time Clock (Clock/Calendar) monitor (hour/minute)</td><td>Hour and minute data of the Real-Time Clock(Clock/Calendar) are stored here. This data is read-only data. It cannot be overwritten.

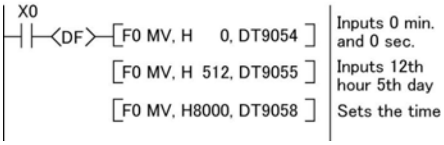
</td><td>A</td><td>N/A</td></tr>
<tr><td>DT9054</td><td>DT90054 (*Note)</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (minute/second)</td><td rowspan="4">The year, month, day, hour, minute, second and day-of-the-week data for the calender timer is stored. The built-in Real-Time Clock(Clock/Calendar) will operate correctly through the year 2099 and supports leap years.
The Real-Time Clock (Clock/Calendar) can be set (the time set) by writing a value using a programming tool software or a program that uses the F0 (MV) transfer instruction.


<table><tr><td>DT9054/ DT90054</td><td>Minute data H00 to H59 (BCD)</td><td>Second data H00 to H59 (BCD)</td></tr><tr><td>DT9055/ DT90055</td><td>Day data H01 to H31 (BCD)</td><td>Hour data H00 to H23 (BCD)</td></tr><tr><td>DT9056/ DT90056</td><td>Year data H00 to H99 (BCD)</td><td>Month data H01 to H12 (BCD)</td></tr><tr><td>DT9057/ DT90057</td><td>—</td><td>Day-of-the-week data H00 to H06 (BCD)</td></tr></table></td><td rowspan="4">A</td><td rowspan="4">A</td></tr>
<tr><td>DT9055</td><td>DT90055 (*Note)</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (day/hour)</td></tr>
<tr><td>DT9056</td><td>DT90056 (*Note)</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (year/month)</td></tr>
<tr><td>DT9057</td><td>DT90057 (*Note)</td><td>Real-Time Clock (Clock/Calendar) monitor and setting (day-of-the-week)</td></tr>
</table>

Note) In the FP2, an expansion memory unit is necessary.

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read-
ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td>DT9058</td><td>DT90058
(*Note)</td><td>Real-Time Clock (Clock /Calendar) setting and 30 seconds correction</td><td>
The Real-Time Clock(Clock/Calendar) is adjusted as follows.
When setting the Real-Time Clock (Clock/Calendar) by program
By setting the highest bit of DT9058/DT90058 to 1, the time becomes that written to DT9054 to DT9057/DT90054 to DT90057 by F0 (MV) instruction. After the time is set, DT9058/DT90058 is cleared to 0. (Cannot be performed with any instruciton other than F0 (MV) instruction.)

<Example>
Set the time to 12:00:00 on the 5th day when the X0 turns on.


If you changed the values of DT9054 to DT9057/DT90054 to DT90057 with programming tool software, the time will be set when the new values are written. Therefore, it is unnecessary to write to DT9058/DT90058.

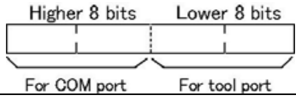
When the correcting times less than 30 seconds-
By setting the lowest bit of DT9058/DT90058 to 1, the value will be moved up or down and become exactly 0 seconds. After the correction is completed, DT9058/DT90058 is cleared to 0.

<Example>
Correct to 0 seconds with X0 turns on.


At the time of correction, if between 0 and 29 seconds, it will be moved down, and if the etween 30 and 59 seconds, it will be moved up. In the example above, if the time was 5 minutes 29 seconds, it will become 5 minutes 0 second; and, if the time was 5 minutes 35 seconds, it will become 6 minutes 0 second.
</td><td>A</td><td>A</td></tr>
</table>

Note) In the FP2, an expansion memory unit is necessary.

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read-ing	Writ-ing
FP3	FP2/ FP2SH FP10SH				
DT9059	**DT90059**	**Serial communication error code**	The system uses this as a communication status when communication error occurs.  	A	N/A
DT9060	**DT90060**	**Step ladder process (0 to 15)**	Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on “1”. Monitor using binary display. (0: not-executing, 1: executing) Example: Bit position: 15 . . 12 | 11 . . 8 | 7 . . 4 | 3 . . 0 Process number: 15 . . 12 | 11 . . 8 | 7 . . 4 | 3 . . 0 DT9060/DT90060: 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 1 Since bit position 0 of DT9060/DT90060 is “1”, step ladder process 0 is executing. A programming tool software can be used to write data.	A	A
DT9061	**DT90061**	**Step ladder process (16 to 31)**			
DT9062	**DT90062**	**Step ladder process (32 to 47)**			
DT9063	**DT90063**	**Step ladder process (48 to 63)**			
DT9064	**DT90064**	**Step ladder process (64 to 79)**			
DT9065	**DT90065**	**Step ladder process (80 to 95)**			
DT9066	**DT90066**	**Step ladder process (96 to 111)**			
DT9067	**DT90067**	**Step ladder process (112 to 127)**			
DT9068	**DT90068**	**Step ladder process (128 to 143)**			
DT9069	**DT90069**	**Step ladder process (144 to 159)**			
DT9070	**DT90070**	**Step ladder process (160 to 175)**			
DT9071	**DT90071**	**Step ladder process (176 to 191)**			
DT9072	**DT90072**	**Step ladder process (192 to 207)**			
DT9073	**DT90073**	**Step ladder process (208 to 223)**			
DT9074	**DT90074**	**Step ladder process (224 to 239)**			
DT9075	**DT90075**	**Step ladder process (240 to 255)**			
DT9076	**DT90076**	**Step ladder process (256 to 271)**			
DT9077	**DT90077**	**Step ladder process (272 to 287)**			

Note) Used by the system.

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read
-ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td>DT9078</td><td>DT90078</td><td>Step ladder process (288 to 303)</td><td rowspan="22">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on “1”.

Monitor using binary display.

(0: not-executing, 1: executing)

Example:
<table><tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr><tr><td>Process number</td><td>335 . 332</td><td>331 . 328</td><td>327. 324</td><td>323 . 320</td></tr><tr><td>DT9080/DT90080</td><td>0 0 0 0</td><td>0 0 0 0</td><td>0 0 0 0</td><td>0 0 0 1</td></tr></table>
Since bit position 0 of DT9080/DT90080 is “1”, step ladder process 320 is executing.

A programming tool software can be used to write data.</td><td rowspan="22">A</td><td rowspan="22">A</td></tr>
<tr><td>DT9079</td><td>DT90079</td><td>Step ladder process (304 to 319)</td></tr>
<tr><td>DT9080</td><td>DT90080</td><td>Step ladder process (320 to 335)</td></tr>
<tr><td>DT9081</td><td>DT90081</td><td>Step ladder process (336 to 351)</td></tr>
<tr><td>DT9082</td><td>DT90082</td><td>Step ladder process (352 to 367)</td></tr>
<tr><td>DT9083</td><td>DT90083</td><td>Step ladder process (368 to 383)</td></tr>
<tr><td>DT9084</td><td>DT90084</td><td>Step ladder process (384 to 399)</td></tr>
<tr><td>DT9085</td><td>DT90085</td><td>Step ladder process (400 to 415)</td></tr>
<tr><td>DT9086</td><td>DT90086</td><td>Step ladder process (416 to 431)</td></tr>
<tr><td>DT9087</td><td>DT90087</td><td>Step ladder process (432 to 447)</td></tr>
<tr><td>DT9088</td><td>DT90088</td><td>Step ladder process (448 to 463)</td></tr>
<tr><td>DT9089</td><td>DT90089</td><td>Step ladder process (464 to 479)</td></tr>
<tr><td>DT9090</td><td>DT90090</td><td>Step ladder process (480 to 495)</td></tr>
<tr><td>DT9091</td><td>DT90091</td><td>Step ladder process (496 to 511)</td></tr>
<tr><td>DT9092</td><td>DT90092</td><td>Step ladder process (512 to 527)</td></tr>
<tr><td>DT9093</td><td>DT90093</td><td>Step ladder process (528 to 543)</td></tr>
<tr><td>DT9094</td><td>DT90094</td><td>Step ladder process (544 to 559)</td></tr>
<tr><td>DT9095</td><td>DT90095</td><td>Step ladder process (560 to 575)</td></tr>
<tr><td>DT9096</td><td>DT90096</td><td>Step ladder process (576 to 591)</td></tr>
<tr><td>DT9097</td><td>DT90097</td><td>Step ladder process (592 to 607)</td></tr>
<tr><td>DT9098</td><td>DT90098</td><td>Step ladder process (608 to 623)</td></tr>
<tr><td>DT9099</td><td>DT90099</td><td>Step ladder process (624 to 639)</td></tr>
</table>

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read
-ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td>DT9100</td><td>DT90100</td><td>Step ladder process (640 to 655)</td><td rowspan="23">Indicates the startup condition of the step ladder process. When the process starts up, the bit corresponding to the process number turns on “1”.

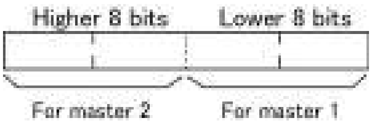
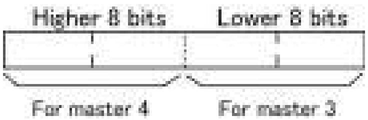
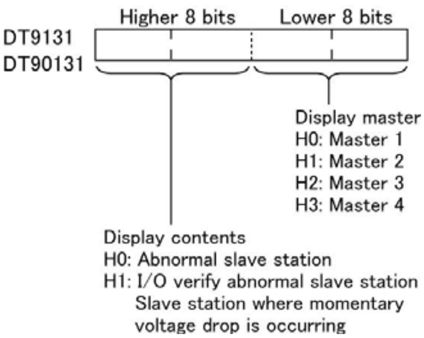
Monitor using binary display.

(0: not-executing, 1: executing)

Example:
<table><tr><td>Bit position</td><td colspan="4">15 . . 12</td><td colspan="4">11 . . 8</td><td colspan="4">7 . . 4</td><td colspan="4">3 . . 0</td></tr><tr><td>Process number</td><td colspan="4">655 . 652</td><td colspan="4">651 . 648</td><td colspan="4">647. 644</td><td colspan="4">643 . 640</td></tr><tr><td>DT9100/DT90100</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>
Since bit position 0 of DT9100/DT90100 is “1”, step ladder process 640 is executing.

A programming tool software can be used to write data.</td><td rowspan="23">A</td><td rowspan="23">A</td></tr>
<tr><td>DT9101</td><td>DT90101</td><td>Step ladder process (656 to 671)</td></tr>
<tr><td>DT9102</td><td>DT90102</td><td>Step ladder process (672 to 687)</td></tr>
<tr><td>DT9103</td><td>DT90103</td><td>Step ladder process (688 to 703)</td></tr>
<tr><td>DT9104</td><td>DT90104</td><td>Step ladder process (704 to 719)</td></tr>
<tr><td>DT9105</td><td>DT90105</td><td>Step ladder process (720 to 735)</td></tr>
<tr><td>DT9106</td><td>DT90106</td><td>Step ladder process (736 to 751)</td></tr>
<tr><td>DT9107</td><td>DT90107</td><td>Step ladder process (752 to 767)</td></tr>
<tr><td>DT9108</td><td>DT90108</td><td>Step ladder process (768 to 783)</td></tr>
<tr><td>DT9109</td><td>DT90109</td><td>Step ladder process (784 to 799)</td></tr>
<tr><td>DT9110</td><td>DT90110</td><td>Step ladder process (800 to 815)</td></tr>
<tr><td>DT9111</td><td>DT90111</td><td>Step ladder process (816 to 831)</td></tr>
<tr><td>DT9112</td><td>DT90112</td><td>Step ladder process (832 to 847)</td></tr>
<tr><td>DT9113</td><td>DT90113</td><td>Step ladder process (848 to 863)</td></tr>
<tr><td>DT9114</td><td>DT90114</td><td>Step ladder process (864 to 879)</td></tr>
<tr><td>DT9115</td><td>DT90115</td><td>Step ladder process (880 to 895)</td></tr>
<tr><td>DT9116</td><td>DT90116</td><td>Step ladder process (896 to 911)</td></tr>
<tr><td>DT9117</td><td>DT90117</td><td>Step ladder process (912 to 927)</td></tr>
<tr><td>DT9118</td><td>DT90118</td><td>Step ladder process (928 to 943)</td></tr>
<tr><td>DT9119</td><td>DT90119</td><td>Step ladder process (944 to 959)</td></tr>
<tr><td>DT9120</td><td>DT90120</td><td>Step ladder process (960 to 975)</td></tr>
<tr><td>DT9121</td><td>DT90121</td><td>Step ladder process (976 to 991)</td></tr>
<tr><td>DT9122</td><td>DT90122</td><td>Step ladder process (992 to 999) (higher byte is not used.)</td></tr>
</table>

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

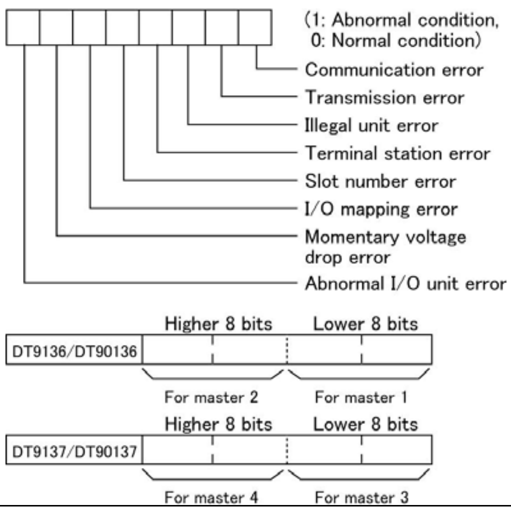
Address		Name	Descriptions	Read-ing	Writ-ing
FP3	**FP2/ FP2SH FP10SH**				
DT9123	**DT90123**	**Not used**	-	N/A	N/A
DT9124	**DT90124**	**Not used**	-		
DT9125	**DT90125**	**Not used**	-		
DT9126 (*Note)	**DT90126 (*Note)**	**Forced on/off operating station display**	This displays the unit number that has executed forced on/off operation.	A	N/A
DT9127 (*Note)	**DT90127 (*Note)**	**MEWNET-F system remote I/O service time**	The number of times, which MEWNET-F remote I/O service was performed by each master, is stored. 		
DT9128 (*Note)	**DT90128 (*Note)**		The number of times, which MEWNET-F remote I/O service was performed by each master, is stored. 		
DT9129	**DT90129**	**Not used**	-	N/A	N/A
DT9130	**DT90130**	**Not used**	-		
DT9131	**DT90131**	**MEWNET-F (remote I/O) slave stations abnormality checking (for selecting the display contents and master of DT9132 to DT9135/DT90132 to DT90135)**	The contents displayed by DT9132 to DT9135/DT90132 to DT90135 will change depending on the contents of stored in DT9131/DT90131. Use the programming tools software to write the settings for what you want to display (this can also be done with the F0 (MV) move instruction). Set the code (H0 or H1) specifying the display contents in the higher 8 bits and set the code (H0 to H3) specifying the display master in the lower 8 bits. 	A	N/A

Note) Used by the system.

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read
-ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td rowspan="2">DT9132
DT9133</td><td rowspan="2">DT90132
DT90133</td><td>MEWNET-F (remote I/O) error slave station number – current condition (when DT9131/ DT90131 is H0, H1, H2 or H3)</td><td>The bit corresponding to the station number of the MEWNET-F where an error is occurring is set to on. Monitor using binary display.
(1: Error slave station, 0: Normal slave station)
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>16 . . 13</td><td>12 . . 9</td><td>8 . . 5</td><td>4 . . 1</td></tr>
<tr><td>DT9132/DT90132</td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>32 . . 29</td><td>28 . . 25</td><td>24 . . 21</td><td>20 . . 17</td></tr>
<tr><td>DT9133/DT90133</td><td></td><td></td><td></td><td></td></tr>
</table></td><td rowspan="4">A</td><td rowspan="4">N/A</td></tr>
<tr><td>MEWNET-F (remote I/O) I/O verify error slave station number (when DT9131/ DT90131 is H100, H101, H102 or H103)</td><td>When the installed condition of a MEWNET-F slave station set unit has changed since the power was turned on, the bit corresponding to that slave station number will be set to on. Monitor using ibnary display.
(1: Error slave station, 0: Normal slave station)
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>16 . . 13</td><td>12 . . 9</td><td>8 . . 5</td><td>4 . . 1</td></tr>
<tr><td>DT9132/DT90132</td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>32 . . 29</td><td>28 . . 25</td><td>24 . . 21</td><td>20 . . 17</td></tr>
<tr><td>DT9133/DT90133</td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td rowspan="2">DT9134
DT9135</td><td rowspan="2">DT90134
DT90135</td><td>MEWNET-F (remote I/O) error slave station number – record (when DT9131/ DT90131 is H0, H1, H2 or H3)</td><td>The bit corresponding to the slave station number of the MEWNET-F where an error is occurring will be set to on. Monitor using binary display.
(1: Error slave station, 0: Normal slave station)
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>16 . . 13</td><td>12 . . 9</td><td>8 . . 5</td><td>4 . . 1</td></tr>
<tr><td>DT9134/DT90134</td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>32 . . 29</td><td>28 . . 25</td><td>24 . . 21</td><td>20 . . 17</td></tr>
<tr><td>DT9135/DT90135</td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>MEWNET-F (remote I/O) momentary voltage drop slave station number (when DT9131/ DT90131 is H100, H101, H102 or H103)</td><td>If a momentary voltage drop at MEWNET-F slave station set, the bit corresponding to that slave station number will be set to on. Monitor using binary display.
(1: Error slave station, 0: Normal slave station)
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>16 . . 13</td><td>12 . . 9</td><td>8 . . 5</td><td>4 . . 1</td></tr>
<tr><td>DT9134/DT90134</td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>Bit position</td><td>15 . . 12</td><td>11 . . 8</td><td>7 . . 4</td><td>3 . . 0</td></tr>
<tr><td>Slave station no.</td><td>32 . . 29</td><td>28 . . 25</td><td>24 . . 21</td><td>20 . . 17</td></tr>
<tr><td>DT9135/DT90135</td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read-ing	Writ-ing
FP3	**FP2/ FP2SH FP10SH**				
DT9136 **DT9137**	**DT90136** **DT90137**	**Error code of MEWNET-F (remote I/O) system**	Display the error conditions for 8 types of errors using 1 byte. 	A	N/A
DT9138	**DT90138**	**Not used**	-	N/A	N/A
DT9139	**DT90139**	**Not used**	-		
DT9140 (*Note1)	**DT90140 (*Note1)**	**MEWNET-W/-P PC link status [PC link 0 (W/P)] (*Note2, 3)**	The number of times the receiving operation is performed (counted using ring counter)	N/A	N/A
DT9141 (*Note1)	**DT90141 (*Note1)**		The current interval between two receiving operations: value in the register x 2.5 ms		
DT9142 (*Note1)	**DT90142 (*Note1)**		The minimum interval between two receiving operations: value in the register x 2.5 ms		
DT9143 (*Note1)	**DT90143 (*Note1)**		The maximum interval between two receiving operations: value in the register x 2.5 ms		
DT9144 (*Note1)	**DT90144 (*Note1)**		The number of times the sending operation is performed (counted using ring counter)		
DT9145 (*Note1)	**DT90145 (*Note1)**		The current interval between two sending operations: value in the register x 2.5 ms		
DT9146 (*Note1)	**DT90146 (*Note1)**		The minimum interval between two sending operations: value in the register x 2.5 ms		
DT9147 (*Note1)	**DT90147 (*Note1)**		The maximum interval between two sending operations: value in the register x 2.5 ms		

Note1) Used by the system.

Note2) When the system register 46 = K0, First: PC link 0, second: PC link 1
When the system register 46 = K1, First: PC link 1, second: PC link 0

Note3) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read-ing	Writ-ing
FP3	**FP2/ FP2SH FP10SH**				
DT9148 (*Note1)	**DT90148 (*Note1)**	**MEWNET-W/-P PC link status [PC link 1 (W/P)] (*Note2, 3)**	The number of times the receiving operation is performed (counted using ring counter)	N/A	N/A
DT9149 (*Note1)	**DT90149 (*Note1)**		The current interval between two receiving operations: value in the register x 2.5 ms		
DT9150 (*Note1)	**DT90150 (*Note1)**		The minimum interval between two receiving operations: value in the register x 2.5 ms		
DT9151 (*Note1)	**DT90151 (*Note1)**		The maximum interval between two receiving operations: value in the register x 2.5 ms		
DT9152 (*Note1)	**DT90152 (*Note1)**		The number of times the sending operation is performed (counted using ring counter)		
DT9153 (*Note1)	**DT90153 (*Note1)**		The current interval between two sending operations: value in the register x 2.5 ms		
DT9154 (*Note1)	**DT90154 (*Note1)**		The minimum interval between two sending operations: value in the register x 2.5 ms		
DT9155 (*Note1)	**DT90155 (*Note1)**		The maximum interval between two sending operations: value in the register x 2.5 ms		
DT9156 (*Note1)	**DT90156 (*Note1)**	**MEWNET-W/-P PC link status [PC link 0 (W/P)] (*Note2, 3)**	Area used for measurement of receiving interval.		
DT9157 (*Note1)	**DT90157 (*Note1)**		Area used for measurement of sending interval.		
DT9158 (*Note1)	**DT90158 (*Note1)**	**MEWNET-W/-P PC link status [PC link 1 (W/P)] (*Note2, 3)**	Area used for measurement of receiving interval.		
DT9159 (*Note1)	**DT90159 (*Note1)**		Area used for measurement of sending interval.		

Note1) Used by the system.
Note2) When the system register 46 = K0, First: PC link 0, second: PC link 1
When the system register 46 = K1, First: PC link 1, second: PC link 0
Note3) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Reading	Writing
FP3	FP2/ FP2SH FP10SH				
DT9160	**DT90160**	**Link unit no. [W/P link 1] (*Note)**	Stores the unit No. of link 1.	A	N/A
DT9161	**DT90161**	**Error flag [W/P link 1] (*Note)**	Stores the error flag of link 1.	A	N/A
DT9162	**DT90162**	**Link unit no. [W/P link 2] (*Note)**	Stores the unit No. of link 2.	A	N/A
DT9163	**DT90163**	**Error flag [W/P link 2] (*Note)**	Stores the error flag of link 2.	A	N/A
DT9164	**DT90164**	**Link unit no. [W/P link 3] (*Note)**	Stores the unit No. of link 3.	A	N/A
DT9165	**DT90165**	**Error flag [W/P link 3] (*Note)**	Stores the error flag of link 3.	A	N/A
DT9166	**DT90166**	**Link unit no. [W/P link 4] Available PLC: FP2SH, FP10SH**	Stores the unit No. of link 4.	A	N/A
DT9167	**DT90167**	**Error flag [W/P link 4] Available PLC: FP2SH, FP10SH**	Stores the error flag of link 4.	A	N/A
DT9168	**DT90168**	**Link unit no. [W/P link 5] Available PLC: FP2SH, FP10SH**	Stores the unit No. of link 5.	A	N/A
DT9169	**DT90169**	**Error flag [W/P link 5] Available PLC: FP2SH, FP10SH**	Stores the error flag of link 5.	A	N/A

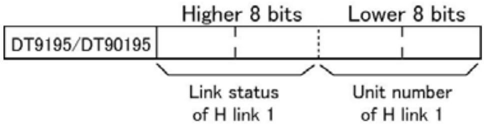
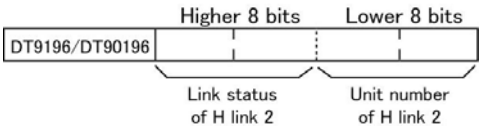
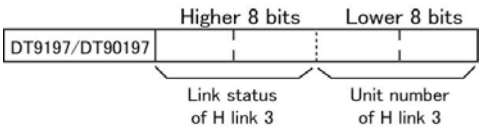
Note) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read-ing	Writ-ing
FP3	**FP2/ FP2SH FP10SH**				
DT9170	**DT90170**	**MEWNET-W/-P link status [W/P link 1] (*Note)**	Station number, whre the send area address for the PC link is overlapped with this station, is stored here.	A	N/A
DT9171	**DT90171**		Test result in the optical transmission path test mode for MEWNET-P link system is stored here.		
DT9172	**DT90172**		Counts how many times a token is lost.		
DT9173	**DT90173**		Counts how many times two or more tokens are detected.		
DT9174	**DT90174**		Counts how many times a signal is lost.		
DT9175	**DT90175**		Counts how many times a synchronous abnormality is detected.		
DT9176	**DT90176**		Send NACK		
DT9177	**DT90177**		Send NACK		
DT9178	**DT90178**		Send WACK		
DT9179	**DT90179**		Send WACK		
DT9180	**DT90180**		Send answer		
DT9181	**DT90181**		Send answer		
DT9182	**DT90182**		Unidentified command		
DT9183	**DT90183**		Counts how many times a parity error is detected.		
DT9184	**DT90184**		End code receiving error		
DT9185	**DT90185**		Format error		
DT9186	**DT90186**		Not support error		
DT9187	**DT90187**		Self-diagnostic result		
DT9188	**DT90188**		Counts how many times loop change is detected. Available PLC: FP3, FP10SH		
DT9189	**DT90189**		Counts home many times link error is detected.		
DT9190	**DT90190**		Counts how many times main loop break is detected. Available PLC: FP3, FP10SH		
DT9191	**DT90191**		Counts how many times sub loop break is detected. Available PLC: FP3, FP10SH		
DT9192	**DT90192**		Loop reconstruction condition Available PLC: FP3, FP10SH		
DT9193	**DT90193**		Loop operation mode Available PLC: FP3, FP10SH		
DT9194	**DT90194**		Loop input status Available PLC: FP3, FP10SH		

Note) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read
-ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td>DT9195</td><td>DT90195</td><td>MEWNET-H link status/ link unit number (H link 1) (For FP2/ FPSH, using W2 mode)</td><td>The link status for the MEWNET-H link is monitored as:

</td><td rowspan="3">A</td><td rowspan="3">N/A</td></tr>
<tr><td>DT9196</td><td>DT90196</td><td>MEWNET-H link status/ link unit number (H link 2) (For FP2/ FPSH, using W2 mode)</td><td>The link status for the MEWNET-H link is monitored as:

</td></tr>
<tr><td>DT9197</td><td>DT90197</td><td>MEWNET-H link status/ link unit number (H link 3) (For FP2/ FPSH, using W2 mode)</td><td>The link status for the MEWNET-H link is monitored as:

</td></tr>
<tr><td>DT9198</td><td>DT90198</td><td>Not used</td><td>-</td><td rowspan="2">N/A</td><td rowspan="2">N/A</td></tr>
<tr><td>DT9199</td><td>DT90199</td><td>Not used</td><td>-</td></tr>
<tr><td>DT9200</td><td>DT90200</td><td rowspan="13">MEWNET-W/-P link status [W/P link 2] (*Note)</td><td>Station number, whre the send area address for the PC link is overlapped with this station, is stored here.</td><td rowspan="13">A</td><td rowspan="13">N/A</td></tr>
<tr><td>DT9201</td><td>DT90201</td><td>Test result in the optical transmission path test mode for MEWNET-P link system is stored here.</td></tr>
<tr><td>DT9202</td><td>DT90202</td><td>Counts how many times a token is lost.</td></tr>
<tr><td>DT9203</td><td>DT90203</td><td>Counts how many times two or more tokens are detected.</td></tr>
<tr><td>DT9204</td><td>DT90204</td><td>Counts how many times a signal is lost.</td></tr>
<tr><td>DT9205</td><td>DT90205</td><td>Counts how many times a synchronous abnormality is detected.</td></tr>
<tr><td>DT9206</td><td>DT90206</td><td>Send NACK</td></tr>
<tr><td>DT9207</td><td>DT90207</td><td>Send NACK</td></tr>
<tr><td>DT9208</td><td>DT90208</td><td>Send WACK</td></tr>
<tr><td>DT9209</td><td>DT90209</td><td>Send WACK</td></tr>
<tr><td>DT9210</td><td>DT90210</td><td>Send answer</td></tr>
<tr><td>DT9211</td><td>DT90211</td><td>Send answer</td></tr>
<tr><td>DT9212</td><td>DT90212</td><td>Unidentified command</td></tr>
</table>

Note) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read -ing	Writ- ing
FP3	FP2/ FP2SH FP10SH				
DT9213	**DT90213**	**MEWNET-W/-P link status [W/P link 2] (*Note)**	Counts how many times a parity error is detected.	A	N/A
DT9214	**DT90214**		End code receiving error		
DT9215	**DT90215**		Format error		
DT9216	**DT90216**		Not support error		
DT9217	**DT90217**		Self-diagnostic result		
DT9218	**DT90218**		Counts how many times loop change is detected. Available PLC: FP3, FP10SH		
DT9219	**DT90219**		Counts home many times link error is detected.		
DT9220	**DT90220**		Counts how many times main loop break is detected. Available PLC: FP3, FP10SH		
DT9221	**DT90221**		Counts how many times sub loop break is detected. Available PLC: FP3, FP10SH		
DT9222	**DT90222**		Loop reconstruction condition Available PLC: FP3, FP10SH		
DT9223	**DT90223**		Loop operation mode Available PLC: FP3, FP10SH		
DT9224	**DT90224**		Loop input status Available PLC: FP3, FP10SH		
DT9225	**DT90225**	**Not used**	-		
DT9226	**DT90226**	**Not used**	-		
DT9227	**DT90227**	**Not used**	-		
DT9228	**DT90228**	**Not used**	-		
DT9229	**DT90229**	**Not used**	-		
DT9230	**DT90230**	**MEWNET-W/-P link status [W/P link 3] (*Note)**	Station number, whre the send area address for the PC link is overlapped with this station, is stored here.	A	N/A
DT9231	**DT90231**		Test result in the optical transmission path test mode for MEWNET-P link system is stored here.		
DT9232	**DT90232**		Counts how many times a token is lost.		
DT9233	**DT90233**		Counts how many times two or more tokens are detected.		
DT9234	**DT90234**		Counts how many times a signal is lost.		
DT9235	**DT90235**		Counts how many times a synchronous abnormality is detected.		
DT9236	**DT90236**		Send NACK		
DT9237	**DT90237**		Send NACK		

Note) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read-ing	Writ-ing
FP3	FP2/ FP2SH FP10SH				
DT9238	**DT90238**	**MEWNET-W/-P link status [W/P link 3] (*Note)**	Send WACK	A	N/A
DT9239	**DT90239**		Send WACK		
DT9240	**DT90240**		Send answer		
DT9241	**DT90241**		Send answer		
DT9242	**DT90242**		Unidentified command		
DT9243	**DT90243**		Counts how many times a parity error is detected.		
DT9244	**DT90244**		End code receiving error		
DT9245	**DT90245**		Format error		
DT9246	**DT90246**		Not support error		
DT9247	**DT90247**		Self-diagnostic result		
DT9248	**DT90248**		Counts how many times loop change is detected. Available PLC: FP3, FP10SH		
DT9249	**DT90249**		Counts home many times link error is detected.		
DT9250	**DT90250**		Counts how many times main loop break is detected. Available PLC: FP3, FP10SH		
DT9251	**DT90251**		Counts how many times sub loop break is detected. Available PLC: FP3, FP10SH		
DT9252	**DT90252**		Loop reconstruction condition Available PLC: FP3, FP10SH		
DT9253	**DT90253**		Loop operation mode Available PLC: FP3, FP10SH		
DT9254	**DT90254**		Loop input status Available PLC: FP3, FP10SH		
-	**DT90255**	**Monitoring tool port station No. (Available PLC: FP2SH/ FP10SH)**	Station number BCD (H1 to H32) set for tool port is stored here.	A	N/A
-	**DT90256**	**Monitoring COM port station No. (Available PLC: FP2SH/ FP10SH)**	Station number BCD (H1 to H32) set for tool port is stored here.		

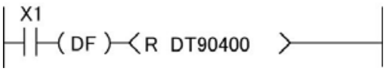
Note) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

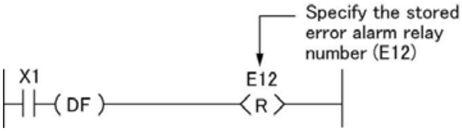
Address		Name	Descriptions	Read -ing	Writ- ing
FP3	FP2/ FP2SH FP10SH				
-	DT90257	**Operation error program No. (hold) (Available PLC: FP2SH/ FP10SH)**	An Operation error program block number is stored (higher byte) here when an operation error is detected. Program block number - H1: In the first program block - H2: In the 2nd program block	A	N/A
-	DT90258	**Operation error program No. (non-hold) (Available PLC: FP2SH/ FP10SH)**	The program block number for the latest operation error is stored here each time an operation error is detected. Program block number - H1: In the first program block - H2: In the 2nd program block	A	N/A
-	DT90259	**Break occurrence program number (Available PLC: FP2SH/ FP10SH)**	The program block number where the BRK instruction occurred is stored here. Program block number - H1: In the first program block - H2: In the 2nd program block	A	N/A
-	DT90260	**Type of IC memory card (Available PLC: FP2SH/ FP10SH)**	Type of IC memory card is monitored here as: - H5: Flash-EEPROM type IC memory card - H6: SRAM type IC memory card - H506: For FP10SH, flash-EEPROM/SRAM mixed type IC memory card - H6: No archival information is stored - H6: No data is written - Other than above: ERROneous condition (self-diagnostic error code E56)	A	N/A
-	DT90261	**Capacity of IC memory card 1 (Available PLC: FP2SH/ FP10SH)**	The capacity of IC memory card is stored in units of KB. If Flash-EEPROM/SRAM mixed type IC memory card is used, SRAM capacity is stored.	A	N/A
-	DT90262	**Capacity of IC memory card 2 (Available PLC: FP2SH/ FP10SH)**	If Flash-EEPROM/SRAM mised type IC memory card is used, flash-EEPROM capacity is stored in units of KB.	A	N/A
-	DT90263	**File register bank (current value)**	The current value of file register bank is stored here.	A	N/A

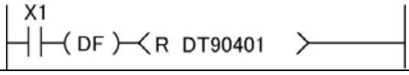
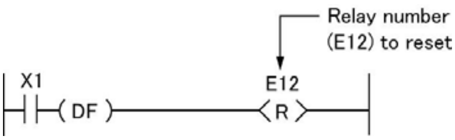
Note) For MEWNET-W system, available PLC type: FP2/FP2SH/FP3/FP10SH
For MEWNET-P system, available PLC type: FP3/FP10SH

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

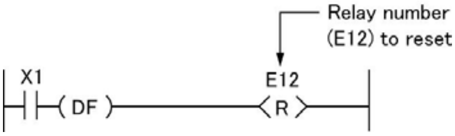
Address		Name	Descriptions	Read-ing	Writ-ing
FP3	FP2/ FP2SH FP10SH				
-	**DT90264**	**File register bank (shelter number) (Available PLC: FP2SH)**	The shelter number of the file register bank is stored here.		
-	**DT90265**	**Free compile memory capacity (Available PLC: FP2SH/ FP10SH)**	Free capacity of compile memory is stored here. If the program memory is 120K steps, the capacity of 1st program block is stored.	A	N/A
-	**DT90266**	**Free compile memory capacity for program block 2 (Available PLC: FP2SH/ FP10SH)**	If the program memory is 120K steps, free caqpacity of program block 2 compile memory is stored here.		
-	**DT90267**	**Not used**	-	N/A	N/A
-	**DT90268**	**Index register bank (current value) (Available PLC: FP2SH/ FP10SH)**	The current value of index register bank is stored here.	A	A
-	**DT90269**	**Index register bank (shelter number) (Available PLC: FP2SH/ FP10SH)**	The shelter number of index register bank is stored here.		
-	**DT90399**	**Not used**	-	N/A	N/A
-	**DT90400**	**Number of the error alarm relay which went on (Available PLC: FP2SH/ FP10SH)**	The total of the error alarm relay which went on is stored here. (Max. 500) To reset all data in the error alarm buffer, use an RST instruction and DT90400. 	A	N/A

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

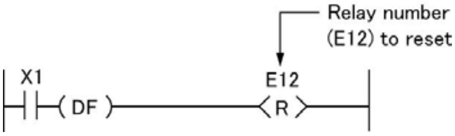
<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read
-ing</th><th rowspan="2">Writ-
ing</th></tr>
<tr><th>FP3</th><th>FP2/
FP2SH
FP10SH</th></tr>
<tr><td>-</td><td>DT90401</td><td>First error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td><td>The first error alarm relay number which went on is stored. The error has been reset by executing a RST instruction.
Example 1: Using RST instruction


Example 2: Using RST instruction and DT90401

</td><td rowspan="7">A</td><td rowspan="7">N/A</td></tr>
<tr><td>-</td><td>DT90402</td><td>Second error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td><td rowspan="6">The error alarm relay number which went on is stored. To reset the specified error alarm relay, use an RST instruction only.

</td></tr>
<tr><td>-</td><td>DT90403</td><td>Third error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90404</td><td>Forth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90405</td><td>Fifth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90406</td><td>Sixth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90407</td><td>Seventh error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
</table>

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

<table>
<tr><th colspan="2">Address</th><th rowspan="2">Name</th><th rowspan="2">Descriptions</th><th rowspan="2">Read -ing</th><th rowspan="2">Writ- ing</th></tr>
<tr><th>FP3</th><th>FP2/ FP2SH FP10SH</th></tr>
<tr><td>-</td><td>DT90408</td><td>Eighth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td><td rowspan="7">The error alarm relay number which went on is stored. To reset the specified error alarm relay, use an RST instruction only.

</td><td rowspan="7">A</td><td rowspan="7">N/A</td></tr>
<tr><td>-</td><td>DT90409</td><td>Ninth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90410</td><td>Tenth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90411</td><td>Eleventh error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90412</td><td>Twelth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90413</td><td>Thirteenth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
<tr><td>-</td><td>DT90414</td><td>Fourteenth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)</td></tr>
</table>

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read-ing	Writ-ing
FP3	FP2/ FP2SH FP10SH				
-	DT90415	**Fifteenth alarm relay which went on (Available PLC: FP2SH/ FP10SH)**	The error alarm relay number which went on is stored. To reset the specified error alarm relay, use an RST instruction only. 	A	N/A
-	DT90416	**Sixteenth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)**			
-	DT90417	**Seventeenth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)**			
-	DT90418	**Eighteenth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)**			
-	DT90419	**Nineteenth error alarm relay which went on (Available PLC: FP2SH/ FP10SH)**			

FP2/FP2SH/FP3/FP10SH (A: Available, N/A: Not available)

Address		Name	Descriptions	Read -ing	Writ- ing
FP3	FP2/ FP2SH FP10SH				
-	**DT90420**	**Time at which the first error alarm relay (DT90401) went on (for minute and second data) (Available PLC: FP2SH/ FP10SH)**	The time (minute and second) data at which the first error alarm relay in DT90401 went on is stored.	A	N/A
-	**DT90421**	**Time at which the first error alarm relay (DT90401) went on (for day and hour data) (Available PLC: FP2SH/ FP10SH)**	The time (day and hour) data at which the first error alarm relay in DT90401 went on is stored.		
	DT90422	**Time at which the first error alarm relay (DT90401) went on (for year and month data) (Available PLC: FP2SH/ FP10SH)**	The time (year and month) data at which the first error alarm relay in DT90401 went on is stored.		

5.2 Table of Basic Instructions

Name	Boolean	Symbol	Description	Steps *3	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
Sequence basic instructions											
Start	**ST**	X,Y,R,T,C,L,P,E	Begins a logic operation with a Form A (normally open) contact.	1 (2)	○	○	○	○	○	○	○
Start Not	**ST/**	X,Y,R,T,C,L,P,E	Begins a logic operation with a Form B (normally closed) contact.	1 (2)	○	○	○	○	○	○	○
Out	**OT**	Y,R,L,E	Outputs the operated result to the specified output.	1 (2)	○	○	○	○	○	○	○
Not	**/**	/	Inverts the operated result up to this instruction.	1	○	○	○	○	○	○	○
AND	**AN**	X,Y,R,T,C,L,P,E	Connects a Form A (normally open) contact serially.	1 (2)	○	○	○	○	○	○	○
AND Not	**AN/**	X,Y,R,T,C,L,P,E	Connects a Form B (normally closed) contact serially.	1 (2)	○	○	○	○	○	○	○
OR	**OR**	X,Y,R,T,C,L,P,E	Connects a Form A (normally open) contact in parallel.	1 (2)	○	○	○	○	○	○	○
OR Not	**OR/**	X,Y,R,T,C,L,P,E	Connects a Form B (normally closed) contact in parallel.	1 (2)	○	○	○	○	○	○	○
Leading edge start	**ST↑**	X,Y,R,T,C,L,P,E	Begins a logic operation only for one scan when the leading edge of the trigger is detected.	2	×	×	○	△ *2	△ *2	○	○
Trailing edge start	**ST↓**	X,Y,R,T,C,L,P,E	Begins a logic operation only for one scan when the trailing edge of the trigger is detected.	2	×	×	○	△ *2	△ *2	○	○
Leading edge AND	**AN↑**	X,Y,R,T,C,L,P,E	Connects a Form A (normally open) contact serially only for one scan when the leading edge of the trigger is detected.	2	×	×	○	△ *2	△ *2	○	○
Trailing edge AND	**AN↓**	X,Y,R,T,C,L,P,E	Connects a Form A (normally open) contact serially only for one scan when the trailing edge of the trigger is detected.	2	×	×	○	△ *2	△ *2	○	○
Leading edge OR	**OR↑**	X,Y,R,T,C,L,P,E	Connects a Form A (normally open) contact in parallel only for one scan when the leading edge of the trigger is detected.	2	×	×	○	△ *2	△ *2	○	○
Trailing edge OR	**OR↓**	X,Y,R,T,C,L,P,E	Connects a Form A (normally open) contact in parallel only for one scan when the trailing edge of the trigger is detected.	2	×	×	○	△ *2	△ *2	○	○
Leading edge out	**OT↑**	P	Outputs the operated result to the specified output only for one scan when leading edge of the trigger is detected. (for pulse relay)	2	×	×	×	×	×	○	○
Trailing edge out	**OT↓**	P	Outputs the operated result to the specified output only for one scan when trailing edge of the trigger is detected. (for pulse relay)	2	×	×	×	×	×	○	○
Alterna-tive out	**ALT**	Y,R,L,E A	Inverts the output condition (on/off) each time the leading edge of the trigger is detected.	3	×	×	○	○	○	○	○
AND stack	**ANS**		Connects the multiple instruction blocks serially.	1	○	○	○	○	○	○	○
OR stack	**ORS**		Connects the multiple instruction blocks in parallel.	1	○	○	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

*1) The type of the devices that can be specified depends on the models.

*2) This instruction is available for FP-X Ver. 2.0 or later, and FPΣ Ver. 3.10 or later.

*3) In the FP2/FP2SH/10SH, when using X1280, Y1280, R1120 (special internal relay included), L1280, T256, C256 or anything beyond for the ST, ST/, OT, AN, AN/, OR and OR/ instructions, the number of steps is shown in parentheses. Also, in the FP2/FP2SH/FP10SH, when a relay number has an index modifier, the number of steps is shown in parentheses. For the FPΣ and FP-X, the number of steps varies according to the relay number to be used.

Name	Boolean	Symbol	Description	Steps *5 *6	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
Push stack	**PSHS**		Stores the operated result up to this instruction. *2	1	○	○	○	○	○	○	○
Read stack	**RDS**		Reads the operated result stored by the PSHS instruction. *2	1	○	○	○	○	○	○	○
Pop stack	**POPS**		Reads and clears the operated result stored by the PSHS instruction	1	○	○	○	○	○	○	○
Leading edge differential	**DF**	(DF)	Turns on the contact for only one scan when the leading edge of the trigger is detected.	1	○	○	○	○	○	○	○
Trailing edge differential	**DF/**	(DF/)	Turns on the contact for only one scan when the trailing edge of the trigger is detected.	1	○	○	○	○	○	○	○
Leading edge differ-ential (initial execution type)	**DFI**	(DFI)	Turns on the contact for only one scan when the leading edge of the trigger is detected. The leading edge detection is possible on the first scan.	1	×	×	○	○	○	○	○
Set	**SET**	Y, R, L, E <S>	Output is set to and held at on.	3	○	○	○	○	○	○	○
Reset	**RST**	Y, R, L, E <R>	Output is set to and held at off.	3	○	○	○	○	○	○	○
Keep	**KP**	Set, Reset, KP	Outputs at set trigger and holds until reset trigger turns on.	1 (2)	○	○	○	○	○	○	○
No operation	**NOP**		No operation.	1	○	○	○	○	○	○	○
Basic function instructions											
On-delay timer	**TML**	TMa, n	After set value "n" x 0.001 seconds, timer contact "a" is set to on.	3 (4)	○	○	○	○	○	○	○ *3
	TMR		After set value "n" x 0.01 seconds, timer contact "a" is set to on.	3 (4)	○	○	○	○	○	○	○ *3
	TMX		After set value "n" x 0.1 seconds, timer contact "a" is set to on.	3 (4)	○	○	○	○	○	○	○ *3
	TMY		After set value "n" x 1 second, timer contact "a" is set to on.	4 (5)	○	○	○	○	○	○	○ *3
Auxiliary timer (16-bit)	**F137 (STMR)**	F137 STMR, S, D; Y,R,L,E	After set value "S" x 0.01 seconds, the specified output and R900D are set to on.	5	○	○	○	○	○	○	○
Auxiliary timer (32-bit)	**F183 (DSTM)**	F183 DSTM, S, D; Y,R,L,E	After set value "S" x 0.01 seconds, the specified output and R900D are set to on.	7	○	○	○	○	○	○	○
Time constant processing	**F182**	F182 FILTR S1, S2, S3, D	Executes the filter processing for the specified input.	9	×	×	○	△ *4	△ *4	×	×
Counter	**CT**	Count, Reset, CT, n	Decrements from the preset value "n"	3 (4)	○	○	○	○ *3	○ *3	○	○ *3

○: Available, ×: Not available, △: Not available partially

*1) The type of the devices that can be specified depends on the models.

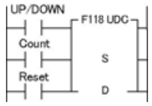
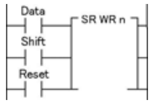
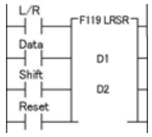
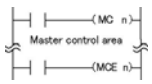
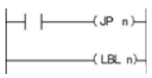
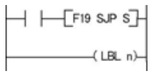
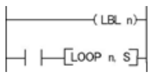
*2) The allowable number of using the PSHS and RDS instruction depends on the models.

*3) For FP2SH, FP10SH and FP-X Ver2.0 or later, any device can be set for the setting value of counter or timer instruction.

*4) This instruction is available for FP-X Ver. 2.0 or later.

*5) In the FP2/FP2SH/FP10SH, when using Y1280, R1120 (special internal relay included), L1280 or anything beyond for the KP instruction, the number of steps is shown in parentheses. Also, in the FP2/FP2SH/FP10SH, when a relay number has an index modifier, the number of steps is shown in parentheses.

*6) In the FP2/FP2SH/FP10SH, when timer 256 or higher, or counter 255 or lower, is used, the number of steps is the number in parentheses. Also, in the FP2/FP2SH/FP10SH, when a timer number or counter number has an index modifier, the number of steps is the number in parentheses. For the FPΣ and FP-X, the number of steps varies according to the specified timer number or counter number.

Name	Boolean	Symbol	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
UP/DOWN counter	**F118 (UDC)**		Increments or decrements from the preset value “S” based on up/donw input.	5	□	□	□	□	□	□	□
Shift register	**SR**		Shifts one bit of 16-bit [word internal relay (WR)] data to the left.	1 (2) *1	□	□	□	□	□	□	□
Left/right shift register	**F119 (LRSR)**		Shifts one bit of 16-bit data range specified by “D1” and “D2” to the left or to the right.	5	□	□	□	□	□	□	□
Control instructions											
Master control relay	**MC**		Starts the master control program.	2	□	□	□	□	□	□	□
Master control relay end	**MCE**		Ends the master control program.	2	□	□	□	□	□	□	□
Jump **Label**	**JP** **LBL**		The program jumps to the label instruction and continues from there.	2 (3) *2 1	□	□	□	□	□	□	□
Auxiliary jump **Label**	**F19 (SJP)** **LBL**		The program jumps to the label instruction specified by “S” and continues from there.	3 1	✕	✕	✕	✕	✕	□	□
Loop **Label**	**LOOP** **LBL**		The program jumps to the label instruction and continues from there (the number of jumps is set in “S”).	4 (5) *3 1	□	□	□	□	□	□	□
Break	**BRK**		Stops program execution when the predetermined trigger turns on in the TEST/RUN mode only.	1	✕	✕	✕	✕	✕	□	□

○: Available, ✕: Not available, △: Not available partially

*1) In the FP2/FP2SH/FP10SH, when internal relay WR240 or higher is used, the number of steps is the number in parentheses. Also, in the FP2/FP2SH/FP10SH, when the specified internal relay number (word address has an index modfier, the number of steps is the number in parentheses.

*2) In the FP2/FP2SH/FP10SH, when the number “n” in a jump instruction has an index modifier, the number of steps isthenumber in parentheses.

*3) In the FP2/FP2SH/FP10SH, when the number “n” in a loop instruction has an index modifier, the number of steps is the number in parentheses.

Name	Boolean	Symbol	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
End	**ED**	(ED)	The operation of program is ended. Indicates the end of a main program.	1	○	○	○	○	○	○	○
Conditional end	**CNDE**	(CNDE)	The operation of program is ended when the trigger turns on.	1	○	○	○	○	○	○	○
Eject	**EJECT**	(EJECT)	Adds page break fo ruse when printing.	1	×	×	○	○	○	○	○
Step ladder instructions											
Start step	**SSTP**	(SSTP n)	The start of program “n” for process control	3	○	○	○	○	○	○	○
Next step	**NSTL**	(NSTL n)	Start the specified process “n” and clear the process currently started. (Scan execution type)	3	○	○	○	○	○	○	○
	NSTP	(NSTP n)	Start the specified process “n” and clear the process currently started. (Pulse execution type)	3	○	○	○	○	○	○	○
Clear step	**CSTP**	(CSTP n)	Resets the specified process “n”.	3	○	○	○	○	○	○	○
Clear multi-ple steps	**SCLR**	[SCLR n1, n2]	Resets multiple processes specified by “n1” and “n2”.	5	○	×	○	○	○	○	○
Step end	**STPE**	(STPE)	End of step ladder area	1	○	○	○	○	○	○	○
Subroutine instructions											
Subroutine call	**CALL**	(CALL n)	When the trigger is on: Executes the subroutine. When the trigger is off: Not execute the subroutine. The output in the subroutine is maintained.	2 (3) *1	○	○	○	○	○	○	○
Output off type subroutine call	**FCAL**	(FCAL n)	When the trigger is on: Executes the subroutine. When the trigger is off: Not execute the subroutine. But, the output in the subroutine is cleared.	4 (5) *1	×	×	×	×	×	×	○
Subroutine entry	**SUB**	(SUB n)	Indicates the start of the subroutine program “n”.	1	○	○	○	○	○	○	○
Subroutine return	**RET**	(RET)	Ends the subroutine program.	1	○	○	○	○	○	○	○
Interrupt instructions											
Interrupt	**INT**	(INT n)	Indicates the start of the interrupt program “n”.	1	○	○	○	○	○	○	○
Interrupt return	**IRET**	(IRET)	Ends the interrupt program.	1	○	○	○	○	○	○	○
Interrupt control	**ICTL**	(DF) [ICTL S1, S2]	Select interrupt enable/disable or clear in “S1” and “S2” and execute.	5	○	○	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

*1) In the FP2/FP2SH/FP10SH, when the number “n” of a subroutine program has an index modifier, the number of steps is the number in parentheses.

Name	Boolean	Symbol	Description	Steps	FP-e	FP0	FP0 (FP0R mode)	FPΣ	FP-X	FP2	FP2SH/FP10SH
Special setting instructions											
Communica-tion condi-tions setting	**SYS1**	(DF) [SYS1, M]	Change the communication conditions for the COM port or tool port based on the contents specified by the character constant.	13	×	×	○	○ *1	○ *1	×	×
Password setting			Change the password specified by the PLC based on the contents specified by the character constant.		×	×	○	○ *2	○ *2	×	×
Interrupt setting			Set the interrupt input based on the contents specified by the character constant.		×	×	○	○	○	×	×
PLC link time setting			Set the system setting time when a PLC link is used, based on the contents specified by the character constant.		×	×	○	○	○	×	×
MEWTOCOL-COM response control			Change the communication conditions of the COM. port or tool port for MEWTOCOL-COM based on the contents specified by the character constant.		×	×	○	○	○	×	×
High-speed counter operation mode changing			Change the operation mode of the high-speed counter, based on the contents specified by the character constant.		×	×	○	○ *3	○ *3	×	×
System registers "No. 40 to No. 47" changing	**SYS2**	[SYS2, S, D1, D2]	Change the setting value of the system register for the PLC link function.	7	×	×	○	○	○	×	×

○: Available, ×: Not available, △: Not available partially

*1) With FP-X Ver2.0 or later, and FPΣ Ver 3.10 or later, the baud rate can be selected from 300, 600 or 1200 bps.

*2) With FPΣ 32k type, the 8-digit password can be selected.

*3) With FPΣ 32k type and FP-X Ver1.10 or later, it can be used.

Name	Boolean	Symbol	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
Data compare instructions											
16-bit data compare (Start)	**ST=**	= S1, S2	Begins a logic operation by comparing two 16-bit data in the comparative condition “S1=S2”.	5	□	□	□	□	□	□	□
	ST<>	< > S1, S2	Begins a logic operation by comparing two 16-bit data in the comparative condition “S1<S2” or “S1>S2”.	5	□	□	□	□	□	□	□
	ST>	> S1, S2	Begins a logic operation by comparing two 16-bit data in the comparative condition “S1>S2”.	5	□	□	□	□	□	□	□
	ST>=	> = S1, S2	Begins a logic operation by comparing two 16-bit data in the comparative condition “S1>S2” or “S1=S2”.	5	□	□	□	□	□	□	□
	ST<	< S1, S2	Begins a logic operation by comparing two 16-bit data in the comparative condition “S1<S2”.	5	□	□	□	□	□	□	□
	ST<=	< = S1, S2	Begins a logic operation by comparing two 16-bit data in the comparative condition “S1<S2” or “S1=S2”.	5	□	□	□	□	□	□	□
16-bit data compare (AND)	**AN=**	= S1, S2	Connects a Form A (normally open) contact serially by comparing two 16-bit data in the comparative condition “S1=S2”.	5	□	□	□	□	□	□	□
	AN<>	< > S1, S2	Connects a Form A (normally open) contact serially by comparing two 16-bit data in the comparative condition “S1<S2” or “S1>S2”.	5	□	□	□	□	□	□	□
	AN>	> S1, S2	Connects a Form A (normally open) contact serially by comparing two 16-bit data in the comparative condition “S1>S2”.	5	□	□	□	□	□	□	□
	AN>=	> = S1, S2	Connects a Form A (normally open) contact serially by comparing two 16-bit data in the comparative condition “S1>S2” or “S1=S2”.	5	□	□	□	□	□	□	□
	AN<	< S1, S2	Connects a Form A (normally open) contact serially by comparing two 16-bit data in the comparative condition “S1<S2”.	5	□	□	□	□	□	□	□
	AN<=	< = S1, S2	Connects a Form A (normally open) contact serially by comparing two 16-bit data in the comparative condition “S1<S2” or “S1=S2”.	5	□	□	□	□	□	□	□
16-bit data compare (OR)	**OR=**	= S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 16-bit data in the comparative condition “S1=S2”.	5	□	□	□	□	□	□	□
	OR<>	< > S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 16-bit data in the comparative condition “S1<S2” or “S1>S2”.	5	□	□	□	□	□	□	□
	OR>	> S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 16-bit data in the comparative condition “S1>S2”.	5	□	□	□	□	□	□	□
	OR>=	> = S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 16-bit data in the comparative condition “S1>S2” or “S1=S2”.	5	□	□	□	□	□	□	□
	OR<	< S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 16-bit data in the comparative condition “S1<S2”.	5	□	□	□	□	□	□	□
	OR<=	< = S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 16-bit data in the comparative condition “S1<S2” or “S1=S2”.	5	□	□	□	□	□	□	□

○: Available, ✕: Not available, △: Not available partially

Name	Boolean	Symbol	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
32-bit data compare (Start)	**STD=**	D= S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
	STD<>	D<> S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)>(S2+1, S2)”.	9	□	□	□	□	□	□	□
	STD>	D> S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)”.	9	□	□	□	□	□	□	□
	STD>=	D>= S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
	STD<	D< S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)”.	9	□	□	□	□	□	□	□
	STD<=	D<= S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
32-bit data compare (AND)	**AND=**	D= S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
	AND<>	D<> S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)>(S2+1, S2)”.	9	□	□	□	□	□	□	□
	AND>	D> S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)”.	9	□	□	□	□	□	□	□
	AND>=	D>= S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
	AND<	D< S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)”.	9	□	□	□	□	□	□	□
	AND<=	D<= S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
32-bit data compare (OR)	**ORD=**	D= S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
	ORD<>	D<> S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)>(S2+1, S2)”.	9	□	□	□	□	□	□	□
	ORD>	D> S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)”.	9	□	□	□	□	□	□	□
	ORD>=	D>= S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□
	ORD<	D< S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)”.	9	□	□	□	□	□	□	□
	ORD<=	D<= S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	□	□	□	□	□	□	□

○: Available, ✕: Not available, △: Not available partially

Name	Boolean	Symbol	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
Floating point type real number data compare (Start)	**STF=**	F= S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	STF<>	F<> S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)>(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	STF>	F> S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	STF>=	F>= S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	STF<	F< S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	STF<=	F<= S1, S2	Begins a logic operation by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
Floating point type real number data compare (AND)	**ANF=**	F= S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ANF<>	F<> S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)>(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ANF>	F> S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ANF>=	F>= S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ANF<	F< S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ANF<=	F<= S1, S2	Connects a Form A (normally open) contact serially by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
Floating point type real number data compare (OR)	**ORF=**	F= S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ORF<>	F<> S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)>(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ORF>	F> S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ORF>=	F>= S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)>(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ORF<	F< S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×
	ORF<=	F<= S1, S2	Connects a Form A (normally open) contact in parallel by comparing two 32-bit data in the comparative condition “(S1+1, S1)<(S2+1, S2)” or “(S1+1, S1)=(S2+1, S2)”.	9	×	×	○	△ *1	△ *1	×	×

○: Available, ×: Not available, △: Not available partially

*1) This instruction is available for FP-X V1.10 or later and FPΣ 32k type

5.3 Table of High-level Instructions

The high-level instructions are expressed by the prefixes “F” or “P” with numbers. For most of the high-level instructions, “F” and “P” types are available. The differences between the two types are explained as follows:
- Instructions with the prefix “F” are executed in every scan while its trigger is in the on.
- Instructions with the prefix “P” are executed only when the leading edge of its trigger is detected.

For the FP0/FP0R/FPΣ/FP-X, the P type high-level instructions are not available.

Num-ber	Name	Boo-lean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
Data transfer instructions												
F0 P0	**16-bit data move**	**MV PMV**	S, D	(S)→(D)	5	○	○	○	○	○	○	○
F1 P1	**32-bit data move**	**DMV PDMV**	S, D	(S+1, S)→(D+1, D)	7	○	○	○	○	○	○	○
F2 P2	**16-bit data invert and move**	**MV PMV/**	S, D	(S)→(D)	5	○	○	○	○	○	○	○
F3 P3	**32-bit data invert and move**	**DMV/ PDMV/**	S, D	$\overline{(S+1, S)}$→(D+1, D)	7	○	○	○	○	○	○	○
F4 P4	**Reading of head word No. of the specified slot**	**GETS PGETS**	S, D	The head word No. of the specified slot is read.	5	×	×	×	×	×	△ *1	△ *1
F5 P5	**Bit data move**	**BTM PBTM**	S, n, D	The specified one bit in “S” is transferred to the specified one bit in “D”. The bit is specified by “n”.	7	○	○	○	○	○	○	○
F6 P6	**Hexadecimal digit (4-bit) data move**	**DGT PDGT**	S, n, d	The specified one digit in “S” is transferred to the specified one digit in “D”. The digit is specified by “n”.	7	○	○	○	○	○	○	○
F7 P7	**Two 16-bit data move**	**MV2 PMV2**	S1, S2, D	(S1)→(D), (S2)→(D+1)	7	×	×	○	○	○	○	○
F8 P8	**Two 32-bit data move**	**DMV2 PDMV2**	S1, S2, D	(S1+1, S1)→(D+1, D), (S2+1, S2)→(D+3, D+2)	11	×	×	○	○	○	○	○
F10 P10	**Block move**	**BKMV PBKMV**	S1, S2, D	The data between “S1” and “S2” is transferred to the area starting at “D”.	7	○	○	○	○	○	○	○
F11 P11	**Block copy**	**COPY PCOPY**	S, D1, D2	The data of “S” is transferred to the all area between “D1” and “D2”.	7	○	○	○	○	○	○	○
F12	**Data read from EEP-ROM**	**ICRD**	S1, S2, D	The data stored in the expansion memory of the EEP-ROM specified by “S1” and “S2” are transferred to the area startign at “D”.	11	○	○ *2	×	×	×	×	×
P13	**Data write to EEP-ROM**	**PICWT**	S1, S2, D	The data specified by “S1” and “S2” are transferred to the EEP-ROM starting at “D”.	11	○	○ *2	×	×	×	×	×
F12	**Data read from F-ROM**	**ICRD**	S1, S2, D	The data stored in the expansion memory of the F-ROM specified by “S1” and “S2” are transferred to the area startign at “D”.	11	×	×	○	○	○	×	×
P13	**Data write to F-ROM**	**PICWT**	S1, S2, D	The data specified by “S1” and “S2” are transferred to the F-ROM starting at “D”.	11	×	×	○	○	○	×	×
F12 P12	**Data read from IC card**	**ICRD PICRD**	S1, S2, D	The data stored in the expansion memory of the IC card specified by “S1” and “S2” are transferred to the area startign at “D”.	11	×	×	×	×	×	×	○
F13 P13	**Data write to IC card**	**ICWT PICWT**	S1, S2, D	The data specified by “S1” and “S2” are transferred to the IC card expansion memory area starting at “D”.	11	×	×	×	×	×	×	○
F14 P14	**Program read from IC memory card**	**PGRD PPGRD**	S	The program specified using “S” is transferred into the CPU from IC memory card and executes it.	3	×	×	×	×	×	×	○

○ : Available, × : Not available, △ : Not available partially

*1) This instruction is available for FP2/FP2SH Ver. 1.5 or later.FP10SH cannot be used

*2) This instruction is available for FP0 Ver. 2.0 or later.

Num-ber	Name	Boo-lean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F15 **P15**	**16-bit data exchange**	**XCH** **PXCH**	D1, D2	(D1)→(D2), (D2)→(D1)	5	○	○	○	○	○	○	○
F16 **P16**	**32-bit data exchange**	**DXCH** **PDXCH**	D1, D2	(D1+1, D1)→(D2+1, D2) (D2+1, D2)→(D1+1, D1)	5	○	○	○	○	○	○	○
F17 **P17**	**Higher/lower byte in 16-bit data exchange**	**SWAP** **PSWAP**	D	The higher byte and lower byte of “D” are exchanged.	3	○	○	○	○	○	○	○
F18 **P18**	**16-bit data block exchange**	**BXCH** **PBXCH**	D1, D2, D3	Exchange the data between “D1” and “D2” with the data specified by “D3”.	7	×	×	○	○	○	○	○
Control instruction												
F19	**Auxiliary jump**	**SJP**	S	The program jumps to the label instruction specified by “S” and continues from there.	3	×	×	×	×	×	○	○
Binary arithmetic instructions												
F20 **P20**	**16-bit data addition**	**+** **P+**	S, D	(D)+(S)→(D)	5	○	○	○	○	○	○	○
F21 **P21**	**32-bit data addition**	**D+** **PD+**	S, D	(D+1, D)+(S+1, S)→(D+1, D)	7	○	○	○	○	○	○	○
F22 **P22**	**16-bit data addition**	**+** **P+**	S1, S2, D	(S1)+(S2)→(D)	7	○	○	○	○	○	○	○
F23 **P23**	**32-bit data addition**	**D+** **PD+**	S1, S2, D	(S1+1, S1)+(S2+1, S2)→(D+1, D)	11	○	○	○	○	○	○	○
F25 **P25**	**16-bit data subtraction**	**-** **P-**	S, D	(D)-(S)→(D)	5	○	○	○	○	○	○	○
F26 **P26**	**32-bit data subtraction**	**D-** **PD-**	S, D	(D+1, D)-(S+1, S)→(D+1, D)	7	○	○	○	○	○	○	○
F27 **P27**	**16-bit data subraction**	**-** **P-**	S1, S2, D	(S1)-(S2)→(D)	7	○	○	○	○	○	○	○
F28 **P28**	**32-bit data subtraction**	**D-** **PD-**	S1, S2, D	(S1+1, S1)-(S2+1, S2)→(D+1, D)	11	○	○	○	○	○	○	○
F30 **P30**	**16-bit data multiplication**	***** **P***	S1, S2, D	(S1)X(S2)→(D+1, D)	7	○	○	○	○	○	○	○
F31 **P31**	**32-bit data multiplication**	**D*** **PD***	S1, S2, D	(S1+1, S1)X(S2+1, S2)→(D+3, D+2, D+1, D)	11	○	○	○	○	○	○	○
F32 **P32**	**16-bit data division**	**%** **P%**	S1, S2, D	(S1)÷(S2)→quotient (D) remainder (DT9015)	7	○	○	○	○	○	○	○
F33 **P33**	**32-bit data division**	**D%** **PD%**	S1, S2, D	(S1+1, S1)÷(S2+1, S2)→quotient (D+1, D) remainder (DT9016, DT9015)	11	○	○	○	○	○	○	○
F34 **P34**	**16-bit data multiplication (result in 16 bits)**	***W** **P*W**	S1, S2, D	(S1)X(S2)→(D)	7	×	×	○	○	○	○	○
F35 **P35**	**16-bit data increment**	**+1** **P+1**	D	(D)+1→(D)	3	○	○	○	○	○	○	○
F36 **P36**	**32-bit data increment**	**D+1** **PD+1**	D	(D+1, D)+1→(D+1, D)	3	○	○	○	○	○	○	○
F37 **P37**	**16-bit data decrement**	**-1** **P-1**	D	(D)-1→(D)	3	○	○	○	○	○	○	○
F38 **P38**	**32-bit data decrement**	**D-1** **PD-1**	D	(D+1, D)-1→(D+1, D)	3	○	○	○	○	○	○	○
F39 **P39**	**32-bit data multiplication (result in 32 bits)**	**D*D** **PD*D**	S1, S2, D	(S1+1, S1)x(S2+1, S2)→(D+1, D)	11	×	×	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

Num-ber	Name	Boo-lean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
BCD arithmetic instructions												
F40 **P40**	**4-digit BCD data addition**	**B+** **PB+**	S, D	(D)+(S)→(D)	5	□	□	□	□	□	□	□
F41 **P41**	**8-digit BCD data addition**	**DB+** **PDB+**	S, D	(D+1, D)+(S+1, S)→(D+1, D)	7	□	□	□	□	□	□	□
F42 **P42**	**4-digit BCD data addition**	**B+** **PB+**	S1, S2, D	(S1)+(S2)→(D)	7	□	□	□	□	□	□	□
F43 **P43**	**8-digit BCD data addition**	**DB+** **PDB+**	S1, S2, D	(S1+1, S1)+(S2+1, S2)→(D+1, D)	11	□	□	□	□	□	□	□
F45 **P45**	**4-digit BCD data subtraction**	**B-** **PB-**	S, D	(D)-(S)→(D)	5	□	□	□	□	□	□	□
F46 **P46**	**8-digit BCD data subtraction**	**DB-** **PDB-**	S, D	(D+1, D)-(S+1, S)→(D+1, D)	7	□	□	□	□	□	□	□
F47 **P47**	**4-digit BCD data subtraction**	**B-** **PB-**	S1, S2, D	(S1)-(S2)→(D)	7	□	□	□	□	□	□	□
F48 **P48**	**8-digit BCD data subraction**	**DB-** **PDB-**	S1, S2, D	(S1+1, S1)-(S2+1, S2)→(D+1, D)	11	□	□	□	□	□	□	□
F50 **P50**	**4-digit BCD data multiplication**	**B*** **PB***	S1, S2, D	(S1)X(S2)→(D+1, D)	7	□	□	□	□	□	□	□
F51 **P51**	**8-digit BCD data multiplication**	**DB*** **PDB***	S1, S2, D	(S1+1, S1)X(S2+1, S2)→(D+3, D+2, D+1, D)	11	□	□	□	□	□	□	□
F52 **P52**	**4-digit BCD data division**	**B%** **PB%**	S1, S2, D	(S1)÷(S2)→quotient (D) remainder (DT9015)	7	□	□	□	□	□	□	□
F53 **P53**	**8-digit BCD data division**	**DB%** **PDB%**	S1, S2, D	(S1+1, S1)÷(S2+1, S2)→quotient (D+1, D) remainder (DT9016, DT9015)	11	□	□	□	□	□	□	□
F55 **P55**	**4-digit BCD data increment**	**B+1** **PB+1**	D	(D)+1→(D)	3	□	□	□	□	□	□	□
F56 **P56**	**8-digit BCD data increment**	**DB+1** **PDB+1**	D	(D+1, D)+1→(D+1, D)	3	□	□	□	□	□	□	□
F57 **P57**	**4-digit BCD data decrement**	**B-1** **PB-1**	D	(D)-1→(D)	3	□	□	□	□	□	□	□
F58 **P58**	**8-digit BCD data decrement**	**DB-1** **PDB-1**	D	(D+1, D)-1→(D+1, D)	3	□	□	□	□	□	□	□
Data compare instructions												
F60 **P60**	**16-bit data compare**	**CMP** **PCMP**	S1, S2	(S1)>(S2)→R900A: on (S1)=(S2)→R900B: on (S1)<(S2)→R900C: on	5	□	□	□	□	□	□	□
F61 **P61**	**32-bit data compare**	**DCMP** **PDCMP**	S1, S2	(S1+1, S1)>(S2+1, S2)→R900A: on (S1+1, S1)=(S2+1, S2)→R900B: on (S1+1, S1)<(S2+1, S2)→R900C: on	9	□	□	□	□	□	□	□
F62 **P62**	**16-bit data band compare**	**WIN** **PWIN**	S1, S2, S3	(S1)>(S3)→R900A: on (S2)< or=(S1)< or=(S3)→R900B: on (S1)<(S2)→R900C: on	7	□	□	□	□	□	□	□

○: Available, ✕: Not available, △: Not available partially

Num-ber	Name	Boo-lean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F63 **P63**	**32-bit data band compare**	**DWIN** **PDWIN**	S1, S2, S3	(S1+1, S1)>(S3+1, S3)→R900A: on (S2+1, S2)< or=(S1+1, S1)< or=(S3+1, S3)→R900B: on (S1+1, S1)<(S2+1, S2)→R900C: on	13	○	○	○	○	○	○	○
F64 **P64**	**Block data compare**	**BCMP** **PBCMP**	S1, S2, S3	Compares the two blocks beginning with “S2” and “S3” to see if they are equal.	7	○	○	○	○	○	○	○
Logic operation instructions												
F65 **P65**	**16-bit data AND**	**WAN** **PWAN**	S1, S2, D	(S1) AND (S2)→(D)	7	○	○	○	○	○	○	○
F66 **P66**	**16-bit data OR**	**WOR** **PWOR**	S1, S2, D	(S1) OR (S2)→(D)	7	○	○	○	○	○	○	○
F67 **P67**	**16-bit data exclusive OR**	**XOR** **PXOR**	S1, S2, D	{(S1) AND $\overline{(S2)}$} OR {$\overline{(S1)}$ AND (S2)}→(D)	7	○	○	○	○	○	○	○
F68 **P68**	**16-bit data exclusive NOR**	**XNR** **PXNR**	S1, S2, D	{(S1) AND (S2)} OR {$\overline{(S1)}$ AND $\overline{(S2)}$}→(D)	7	○	○	○	○	○	○	○
F69 **P69**	**16-bit data unite**	**WUNI** **PWUNI**	S1, S2, S3, D	([S1] AND [S3]) OR ([S2] AND $\overline{[S3]}$)→(D) When (S3) is H0, (S2)→(D) When (S3) is HFFFF, (S1) →(D)	9	✕	✕	○	○	○	○	○
Data conversion instructions												
F70 **P70**	**Block check code calculation**	**BCC** **PBCC**	S1, S2, S3, D	Creates the code for checking the data specified by “S2” and “S3” and stores it in “D”. The calculation method is specified by “S1”.	9	○	○	○	○	○	○	○
F71 **P71**	**Hexadecimal data → ASCII code**	**HEXA** **PHEXA**	S1, S2, D	Converts the hexadecimal data specified by “S1” and “S2” to ASCII code and stores it in “D”. Example: HABCD→ H 42 41 44 43 B A D C	7	○	○	○	○	○	○	○
F72 **P72**	**ASCII code → Hexadeci-mal data**	**AHEX** **PAHEX**	S1, S2, D	Converts the ASCII code specified by “S1” and “S2” to hexadecimal data and stores it in “D”. Example: H 44 43 42 41 → HCDAB D C B A	7	○	○	○	○	○	○	○
F73 **P73**	**4-digit BCD data → ASCII code**	**BCDA** **PBCDA**	S1, S2, D	Converts the four digits of BCD data specified by “S1” and “S2” to ASCII code and stores it in “D”. Example: H1234→ H 32 31 34 33 2 1 4 3	7	○	○	○	○	○	○	○
F74 **P74**	**ASCII code → 4-digit BCD data**	**ABCD** **PABCD**	S1, S2, D	Converts the ASCII code specified by “S1” and “S2” to four digits of BCD data and stores it in “D”. Example: H 34 33 32 31 → H3412 4 3 2 1	9	○	○	○	○	○	○	○
F75 **P75**	**16-bit binary data → ASCII code**	**BINA** **PBINA**	S1, S2, D	Converts the 16 bits of binary data specified by “S1” to ASCII code and stores it in “D” (area of “S2” bytes). Example: K-100→ H 30 30 31 2D 20 20 0 0 1 -	7	○	○	○	○	○	○	○

○: Available, ✕: Not available, △: Not available partially

Num-ber	Name	Boo-lean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F76 **P76**	**ASCII code → 16-bit binary data**	**ABIN** **PABIN**	S1, S2, D	Converts the ASCII code specified by “S1” and “S2” to 16 bits of binary data and stores it in “D”. Example: H 30 30 31 2D 20 20 → K-100 0 0 1 -	7	□	□	□	□	□	□	□
F77 **P77**	**32-bit binary data → ASCII code**	**DBIA** **PDBIA**	S1, S2, D	Converts the 32 bits of binary data (S1+1, S1) to ASCII code and stores it in D (area of “S2” bytes).	11	□	□	□	□	□	□	□
F78 **P78**	**ASCII code → 32-bit binary data**	**DABI** **PDABI**	S1, S2, D	Converts the ASCII code specified by “S1” and “S2” to 32 bits of binary data and stores it in (D+1, D).	11	□	□	□	□	□	□	□
F80 **P80**	**16-bit binary data → 4-digit BCD data**	**BCD** **PBCD**	S, D	Converts the 16 bits of binary data specified by “S” to four digits of BCD data and stores it in “D”. Example: K100 → H100	5	□	□	□	□	□	□	□
F81 **P81**	**4-digit BCD data → 16-bit binary data**	**BIN** **PBIN**	S, D	Converts the four digits of BCD data specified by “S” to 16 bits of binary data and stores it in “D”. Example: H100 → K100	5	□	□	□	□	□	□	□
F82 **P82**	**32-bit binary data → 8-digit BCD data**	**DBCD** **PDBCD**	S, D	Converts the 32 bits of binary data specified by (S+1, S) to eight digits of BCD data and stores it in (D+1, D).	7	□	□	□	□	□	□	□
F83 **P83**	**8-digit BCD data → 32-bit binary data**	**DBIN** **PDBIN**	S, D	Converts the eight digits of BCD data specified by (S+1, S) to 32 bits of binary data and stores it in (D+1, D).	7	□	□	□	□	□	□	□
F84 **P84**	**16-bit data invert (com-plement of 1)**	**INV** **PINV**	D	Inverts each bit of data of “D”.	3	□	□	□	□	□	□	□
F85 **P85**	**16-bit data complement of 2**	**NEG** **PNEG**	D	Inverts each bit of data of “D” and adds 1 (inverts the sign).	3	□	□	□	□	□	□	□
F86 **P86**	**32-bit data complement of 2**	**DNEG** **PDNEG**	D	Inverts each bit of data of (D+1, D) and adds 1 (inverts the sign).	3	□	□	□	□	□	□	□
F87 **P87**	**16-bit data absolute**	**ABS** **PABS**	D	Gives the absolute value of the data of “D”.	3	□	□	□	□	□	□	□
F88 **P88**	**32-bit data absolute**	**DABS** **PDABS**	D	Gives the absolute value of the data of (D+1, D).	3	□	□	□	□	□	□	□
F89 **P89**	**16-bit data sign extension**	**EXT** **PEXT**	D	Extends the 16 bits of data in “D” to 32 bits in (D+1, D).	3	□	□	□	□	□	□	□
F90 **P90**	**Decode**	**DECO** **PDECO**	S, n, D	Decodes part of the data of “S” and stores it in “D”. The part is specified by “n”.	7	□	□	□	□	□	□	□
F91 **P91**	**7-segment decode**	**SEGT** **PSEGT**	S, D	Converts the data of “S” for use in a 7-segment display and stores it in (D+1, D).	5	□	□	□	□	□	□	□
F92 **P92**	**Encode**	**ENCO** **PENCO**	S, n, D	Encodes part of the data of “S” and stores it in “D”. The part is specified by “n”.	7	□	□	□	□	□	□	□
F93 **P93**	**16-bit data combine**	**UNIT** **PUNIT**	S, n, D	The least significant digit of each of the “n” words of data beginning at “S” are stored (united) in order in “D”.	7	□	□	□	□	□	□	□

○: Available, ✕: Not available, △: Not available partially

Num-ber	Name	Boo-lean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F94 P94	**16-bit data distribute**	**DIST PDIST**	S, n, D	Each of the digits of the data of “S” are stored in (distriuted to) the least significant digits of the areas beginning at “D”.	7	○	○	○	○	○	○	○
F95 P95	**Character→ ASCII code**	**ASC PASC**	S, D	Twelve characters of the characer constants of “S” are converted to ASCII code and stored in “D” to “D+5”.	15	○	○	○	○	○	○	○
F96 P96	**16-bit table data search**	**SRC PSRC**	S1, S2, S3	The data of “S1” is searched for in the areas in the range “S2” to “S3” and the result is stored in DT9037 and DT9038	7	○	○	○	○	○	○	○
F97 P97	**32-bit table data search**	**DSRC PDSRC**	S1, S2, S3	The data of (S1+1, S1) is searched for in the 32-bit data designated by “S3”, beginning from “S2”, and the result if stored in DT90037 and DT90038.	11	×	×	○	○	○	○	○
Data shift instructions												
F98 P98	**Data table shift-out and compress**	**CMPR PCMPR**	D1, D2, D3	Transfer “D2” to “D3”. Any parts of the data between “D1” and “D2” that are 0 are compressed, and shifted in order toward “D2”.	7	×	×	○	○	○	○	○
F99 P99	**Data table shift-in and compress**	**CMPW PCMP W**	S, D1, D2	Transfer “S” to “D1”. Any parts of the data between “D1” and “D2” that are 0 are compressed, and shifted in order toward “D2”.	7	×	×	○	○	○	○	○
F100 P100	**Right shift of multiple bits (n bits) in a 16-bit data**	**SHR PSHR**	D, n	Shifts the “n” bits of “D” to the right.	5	○	○	○	○	○	○	○
F101 P101	**Left shift of multiple bits (n bits) in a 16-bit data**	**SHL PSHL**	D, n	Shifts the “n” bits of “D” to the left.	5	○	○	○	○	○	○	○
F102 P102	**Right shift of n bits in a 32-bit data**	**DSHR PDSHR**	D, n	Shifts the “n” bits of the 32-bit data area specified by (D+1, D) to the right.	5	×	×	○	○	○	○	○
F103 P103	**Left shift of n bits in a 32-bit data**	**DSHL PDSHL**	D, n	Shifts the “n” bits of the 32-bit data area specified by (D+1, D) to the left.	5	×	×	○	○	○	○	○
F105 P105	**Right shift of one hexadecimal digit (4-bit)**	**BSR PBSR**	D	Shifts the one digit of data of “D” to the right.	3	○	○	○	○	○	○	○
F106 P106	**Left shift of one hexade-cimal digit (4-bit)**	**BSL PBSL**	D	Shifts the one digit of data of “D” to the left.	3	○	○	○	○	○	○	○
F108 P108	**Right shift of multiple bits (n bits)**	**BITR PBITR**	D1, D2, n	Shifts the “n” bits of data range by “D1” and “D2” to the right.	7	×	×	○	○	○	○	○
F109 P109	**Left shift of multiple bits (n bits)**	**BITL PBITL**	D1, D2, n	Shifts the “n” bits of data range by “D1” and “D2” to the left.	7	×	×	○	○	○	○	○
F110 P110	**Right shift of one word (16-bit)**	**WSHR PWSHR**	D1, D2	Shifts the one word of the areas by “D1” and “D2” to the right.	5	○	○	○	○	○	○	○
F111 P111	**Left shift of one word (16-bit)**	**WSHL PWSHL**	D1, D2	Shifts the one word of the areas by “D1” and “D2” to the left.	5	○	○	○	○	○	○	○
F112 P112	**Right shift of one hexade-cimal digit (4-bit)**	**WBSR PWBSR**	D1, D2	Shifts the one digit of the areas by “D1” and “D2” to the right.	5	○	○	○	○	○	○	○
F113 P113	**Left shift of one hexade-cimal digit (4-bit)**	**WBSL PWBSL**	D1, D2	Shifts the one digit of the areas by “D1” and “D2” to the left.	5	○	○	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

Number	Name	Boolean	Operand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
FIFO instructions												
F115 P115	**FIFO buffer define**	**FIFT PFIFT**	n, D	The "n" words beginning from "D" are defined in the buffer.	5	×	×	○	○	○	○	○
F116 P116	**Data read from FIFO buffer**	**FIFR PFIFR**	S, D	The oldest data beginning from "S" that was written to the buffer is read and stored in "D".	5	×	×	○	○	○	○	○
F117 P117	**Data write into FIFO buffer**	**FIFW PFIFW**	S, D	The data of "S" is written to the buffer starting from "D".	5	×	×	○	○	○	○	○
Basic function instructions												
F118	**UP/DOWN counter**	**UDC**	S, D	Counts up or down from the value preset in "S" and stores the elapsed value in "D".	5	○	○	○	○	○	○	○
F119	**Left/right shift register**	**LRSR**	D1, D2	Shifts one bit to the left or right with the area between "D1" and "D2" as the register.	5	○	○	○	○	○	○	○
Data rotate instructions												
F120 P120	**16-bit data right rotate**	**ROR PROR**	D, n	Rotate the "n" bits in data of "D" to the right.	5	○	○	○	○	○	○	○
F121 P121	**16-bit data left rotate**	**ROL PROL**	D, n	Rotate the "n" bits in data of "D" to the left.	5	○	○	○	○	○	○	○
F122 P122	**16-bit data right rotate with carry flag (R9009) data**	**RCR PRCR**	D, n	Rotate the "n" bits in 17-bit area consisting of "D" plus the carry flag (R9009) data to the right.	5	○	○	○	○	○	○	○
F123 P123	**16-bit data left rotate with carry flag (R9009) data**	**RCL PRCL**	D, n	Rotate the "n" bits in 17-bit area consisting of "D" plus the carry flag (R9009) data to the left.	5	○	○	○	○	○	○	○
F125 P125	**32-bit data right rotate**	**DROR PDROR**	D, n	Rotate the number of bits specified by "n" of the double words data (32 bits) specified by (D+1, D) to the right.	5	×	×	○	○	○	○	○
F126 P126	**32-bit data left rotate**	**DROL PDROL**	D, n	Rotate the number of bits specified by "n" of the double words data (32 bits) specified by (D+1, D) to the left.	5	×	×	○	○	○	○	○
F127 P127	**32-bit data right rotate with carry flag (R9009) data**	**DRCR PDRCR**	D, n	Rotate the number of bits specified by "n" of the double words data (32 bits) specified by (D+1, D) to the right together with carry flag (R9009) data.	5	×	×	○	○	○	○	○
F128 P128	**32-bit data left rotate with carry flag (R9009) data**	**DRCL PDRCL**	D, n	Rotate the number of bits specified by "n" of the double words data (32 bits) specified by (D+1, D) to the left together with carry flag (R9009) data.	5	×	×	○	○	○	○	○
Bit manipulation instructions												
F130 P130	**16-bit data bit set**	**BTS PBTS**	D, n	Set the value of bit position "n" of the data of "D" to 1.	5	○	○	○	○	○	○	○
F131 P131	**16-bit data bit reset**	**BTR PBTR**	D, n	Set the value of bit position "n" of the data of "D" to 0.	5	○	○	○	○	○	○	○
F132 P132	**16-bit data invert**	**BTI PBTI**	D, n	Invert the value of bit position "n" of the data of "D".	5	○	○	○	○	○	○	○
F133 P133	**16-bit data bit test**	**BTT PBTT**	D, n	Test the value of bit position "n" of the data of "D" and output the result to R900B.	5	○	○	○	○	○	○	○
F135 P135	**Number of on (1) bits in 16-bit data**	**BCU PBCU**	S, D	Store the number of on bits in the data of "S" in "D".	5	○	○	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

Num -ber	Name	Boo- lean	Ope- rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F136 **P136**	**Number of on (1) bits in 32-bit data**	**DBCU** **PDBCU**	S, D	Store the number of on bits in the data of (S+1, S) in “D”.	7	○	○	○	○	○	○	○
Basic function instruction												
F137	**Auxiliary timer (16-bit)**	**STMR**	S, D	Turn on the specified output and R900D after 0.01 s × set value.	5	○	○	○	○	○	○	○
Special instructions												
F138 **P138**	**Hours, min- utes and sec- onds to seconds data**	**HMSS** **PHMSS**	S, D	Converts the hour, minute and second data of (S+1, S) to seconds data, and the converted data is stored in (D+1, D).	5	○	△ *1	○	○	○	○	○
F139 **P139**	**Seconds to hours, minutes and seconds data**	**SHMS** **PSHMS**	S, D	Converts the seconds data of (S+1, S) to hour, minute and second data, and the converted data is stored in (D+1, D).	5	○	△ *1	○	○	○	○	○
F140 **P140**	**Carry flag (R9009) set**	**STC** **PSTC**	-	Turns on the carry flag (R9009).	1	○	○	○	○	○	○	○
F141 **P141**	**Carry flag (R9009) reset**	**CLC** **PCLC**	-	Turns off the carry flag (R9009).	1	○	○	○	○	○	○	○
F142 **P142**	**Watching dog timer update**	**WDT** **PWDT**	S	The time (allowable scan time for the system) of watching dog timer is changed to “S” × 0.1 (ms) for that scan.	3	×	×	×	×	×	×	○
F143 **P143**	**Partial I/O update**	**IORF** **PIORF**	D1, D2	Updates the I/O from the number specified by “D1” to the number specified by “D2”.	5	○	○	○	○	○	○	○
F144	**Serial data communica- tion control**	**TRNS**	S, n	The COM port received flag (R9038) is set to off to enable reception. Beginning at “S”, “n” bytes of the data registers are sent from the COM port.	5	○	○ *4	×	×	×	○	○
F145 **P145**	**Data send**	**SEND** **PSEND**	S1, S2, D, N	Sends the data to another station in the network (MEWNET). (via link unit)	9	×	×	×	×	×	○	○
F146 **P146**	**Data receive**	**RECV** **PRECV**	S1, S2, N, D	Receives the data to another station in the network (MEWNET). (via link unit)	9	×	×	×	×	×	○	○
F145 **P145**	**Data send**	**SEND**	S1, S2, D, N	Sends the data to the slave station as the MOD bus master. (via COM port)	9	×	×	○	△ *2	○	×	×
F146 **P146**	**Data receive**	**RECV**	S1, S2, N, D	Receives the data from the slave station as the MOD bus master. (via COM port)	9	×	×	○	△ *2	○	×	×
F145 **P145**	**Data send**	**SEND**	S1, S2, D, N	Sends the data to the slave station of the MOD bus master, type II.	9	×	×	○	△ *3	△ *3	×	×
F146 **P146**	**Data receive**	**RECV**	S1, S2, N, D	Receives the data from the slave station of the MOD bus master, type II.	9	×	×	○	△ *3	△ *3	×	×
F145 **P145**	**Data send**	**SEND**	S1, S2, D, N	Sends the data to the slave station as the MEWTOCOL master. (via COM port)	9	×	×	○	△ *2	△ *2	×	×
F146 **P146**	**Data receive**	**RECV**	S1, S2, N, D	Receives the data from the slave station as the MEWTOCOL master. (via COM port)	9	×	×	○	△ *2	△ *2	×	×
F147	**Printout**	**PR**	S, D	Converts the ASCII code data in the area starting with “S” for printing, and outputs it to the word external output relay WY specified by “D”.	5	○	○	○	○	○	○	○
F148 **P148**	**Self- diagnostic error set**	**ERR** **PERR**	n (n: k100 to K299)	Stores the self-diagnostic error number “n” in (DT9000), turns R9000 on, and turns on the ERROR LED.	3	○	○	○	○	○	○	○
F149 **P149**	**Message display**	**MSG** **PMSG**	S	Displays the character constant of “S” in the connected programming tool.	13	○	○	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

*1) The instruction is available for FP0 T32 type (V2.3 or later).

*2) This instruction is available for FP-X V1.20 or later and FPΣ 32k type.

*3) This instruction is available for FP-X V2.50 or later and FPΣ V3.20 or later.

*4) This instruction is available for FP0 V1.20 or later.

Num-ber	Name	Boolean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F150 **P150**	**Data read from intelli-gent unit**	**READ** **PREAD**	S1, S2, n, D	Reads the data from the intelligent unit.	9	×	×	×	△ *3	×	○	○
F151 **P151**	**Data write into intelli-gent unit**	**WRT** **PWRT**	S1, S2, n, D	Writes the data into the intelligent unit.	9	×	×	×	△ *3	×	○	○
F152 **P152**	**Data read from MEWNET-F slave station**	**RMRD** **PRMRD**	S1, S2, n, D	Reads the data from the intelligent unit at the MEWNET-F (remote I/O) slave station.	9	×	×	×	×	×	○	○
F153 **P153**	**Data write into MEWNET-F slave station**	**RMWT** **PRMWT**	S1, S2, n, D	Writes the data into the intelligent unit at the MEWNET-F (remote I/O) slave station.	9	×	×	×	×	×	○	○
F155 **P155**	**Sampling**	**SMPL** **PSMPL**	-	Starts sampling data.	1	×	×	○	△ *5	△ *4	○	○
F156 **P156**	**Sampling trigger**	**STRG** **PSTRG**	-	When the trigger of this instruction turns on, the sampling trace stops.	1	×	×	○	△ *5	△ *4	○	○
F157 **P157**	**Time addition**	**CADD** **PCADD**	S1, S2, D	The time after (S2+1, S2) elapses from the time of (S1+2, S1+1, S1) is stored in (D+2, D+1, D).	9	○	△ *1	○	○	○	○	○
F158 **P158**	**Time substruction**	**CSUB** **PCSUB**	S1, S2, D	The time that results from subtracting (S2+1, S2) from the time (S1+2, S1+1, S1) is stored in (D+2, D+1, D).	9	○	△ *1	○	○	○	○	○
F159 **P159**	**Serial port communication**	**MTRN** **PMTRN**	S, n, D	This is used to send data to an external device through the specified CPU COM port or MCU COM port.	7	×	×	○	○	○	△ *2	△ *2
F161 **P161**	**MCU serial port reception**	**MRCV** **PMRCV**	S, D1, D2	Data is received from external equipment via the COM port of the specified MCU.	7	×	×	×	×	×	△ *2	△ *2
BIN arithmetic instruction												
F160 **P160**	**Double word (32-bit) data square root**	**DSQR** **PDSQR**	S, D	$\sqrt{(S)} \rightarrow (D)$	7	×	×	○	○	○	○	○
High speed counter/Pulse output instruction for FP0, FP-e												
F0	**High-speed counter and Pulse output controls**	**MV**	S, DT9052	Performs high-speed counter and Pulse output controls according to the control code specified by “S”. The control code is stored in DT9052.	5	○	○					
1	**Change and read of the elapsed value of high-speed counter and Pulse output**	**DMV**	S, DT9044	Transfers (S+1, S) to high-speed counter and Pulse output elapsed value area.	7	○	○					
			DT9044, D	Transfers value in high-speed counter and Pulse output elapsed value area to (D+1, D).	7	○	○					
F166	**High-speed counter output set (with channel specification)**	**HC1S**	n, S, Yn	Turns output Yn on when the elapsed value of the built-in high-speed counter reaches the target value of (S+1, S).	11	○	○					

○: Available, ×: Not available, △: Not available partially

*1) The instruction is available for FP0 T32 type (V2.3 or later).

*2) The instruction is available for FP2/FP2SH Ver. 1.5 or later, and the pulse execution type can be specified. FP10SH cannot be used.

*3) This instruction is available for FPΣ Ver. 2.0 or later.

*4) This instruction is only available for FP-X Ver.2.0 or later.

*5) This instruction is available for FPΣ Ver. 3.10 or later.

Number	Name	Boolean	Operand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F167	**High-speed counter output reset (with channel specification)**	**HC1R**	n, S, Yn	Turns output Yn off when the elapsed value of the built-in high-speed counter reaches the target value of (S+1, S).	11	□	□					
F168	**Positioning control (with channel specification)**	**SPD1**	S, n	Outputs a positioning pulse from the specified output (Y0 or Y1) according to the contents of the data table beginning at “S”.	5	□	□					
F169	**Pulse output (with channel specification)**	**PLS**	S, n	Outputs a pulse from the specified output (Y0 or Y1) according to the contents of the data table beginning at “S”.	5	□	□					
F170	**PWM output (with channel specification)**	**PWM**	S, n	Performs PWM output from the specified outptu (Y0 or Y1) according to the contents of the data table beginning at “S”.	5	□	□					
High speed counter/Pulse output instruction for FP0R												
F0	**High-speed counter and Pulse output controls**	**MV**	S, DT90052	Performs high-speed counter and Pulse output controls according to the control code specified by “S”. The control code is stored in DT90052.	5			□				
F1	**Change and read of the elapsed value of high-speed counter and Pulse output**	**DMV**	S, DT90300	Transfers (S+1, S) to high-speed counter and Pulse output elapsed value area (DT90045, DT90044).	7			□				
			DT90300, D	Transfers value in high-speed counter and Pulse output elapsed value area (DT90045, DT90044) to (D+1, D).	7			□				
F165	**Cam control**	**CAM0**	S	Controls cam operation (on/off patterns of each cam output) according to the elapsed value of the high-speed counter.	3			□				
F166	**Target value much on (with channel specification) (High-speed counter control/Pulse output control)**	**HC1S**	n, S, D	Turns output Yn on when the elapsed value of the high-speed counter or pulse output reaches the target value of (S+1, S).	11			□				
F167	**Target value much off (with channel specification) (High-speed counter control/Pulse output control)**	**HC1R**	n, S, D	Turns output Yn off when the elapsed value of the high-speed counter or pulse output reaches the target value of (S+1, S).	11			□				
F171	**Pulse output (JOG positioning type 0/1) (Trapezoidal control)**	**SPDH**	S, n	Positioning pulses are output from the specified channel, in accordance with the contents of the data table that starts with S.	5			□				
F172	**Pulse output (JOG operation 0 and 1)**	**PLSH**	S, n	Pulse strings are output from the specified output, in accordance with the contents of the data table that starts with S.	5			□				
F173	**PWM output (with channel specification)**	**PWMH**	S, n	PWM output is output from the specified output, in accordance with the contents of the data table that starts with S.	5			□				

Num-ber	Name	Boo-lean	Operand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F174	**Pulse output (Selectable data table control operation)**	**SP0H**	S, n	Outputs the pulses from the specified channel according to the data table specified by S.	5			□				
F175	**Pulse output (Linear interpolation)**	**SPSH**	S, n	Pulses are output from channel, in accordance with the designated data table, so that the path to the target position forms a straight line.	5			□				
F176	**Pulse output (Circular interpolation)**	**SPCH**	S, n	Pulses are output from channel, in accordance with the designated data table, so that the path to the target position forms an arc.	5			×				
F177	**Pulse output (Home return)**	**HOME**	S, n	Performs the home return according to the specified data table.	7			□				
F178	**Input pulse measurement (No. of pulses, cycle for input pulses)**	**PLSM**	S1, S2, D	Measures the number of pulses and cycle of pulses to be input to the high-speed counter of the specified channel.	5			□				

Num-ber	Name	Boo-lean	Operand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
High speed counter/Pulse output instruction for FPΣ/FP-X												
F0	**High-speed counter and Pulse output controls**	**MV**	S, DT90052	Performs high-speed counter and Pulse output controls according to the control code specified by “S”. The control code is stored in DT90052.	5				□	□		
F1	**Change and read of the elapsed value of high-speed counter and Pulse output**	**DMV**	FPΣ: S, DT90044 FP-X: S, DT90300	Transfers (S+1, S) to high-speed counter and Pulse output elapsed value area (DT90045, DT90044).	7				□	□		
			FPΣ: DT90044, D FP-X: DT90300, D	Transfers value in high-speed counter and Pulse output elapsed value area (DT90045, DT90044) to (D+1, D).	7				□	□		
F166	**Target value much on (with channel specification)**	**HC1S**	n, S, D	Turns output Yn on when the elapsed value of the built-in high-speed counter reaches the target value of (S+1, S).	11				□	□		
F167	**Target value much off (with channel specification)**	**HC1R**	n, S, D	Turns output Yn off when the elapsed value of the built-in high-speed counter reaches the target value of (S+1, S).	11				□	□		
F171	**Pulse output (with channel specification) (Trapezoidal control and home return)**	**SPDH**	S, n	Positioning pulses are output from the specified channel, in accordance with the contents of the data table that starts with S.	5				□	□		
F172	**Pulse output (with channel specification) (JOG operation)**	**PLSH**	S, n	Pulse strings are output from the specified output, in accordance with the contents of the data table that starts with S.	5				□	□		
F173	**PWM output (with channel specification)**	**PWMH**	S, n	PWM output is output from the specified output, in accordance with the contents of the data table that starts with S.	5				□	□		
F174	**Pulse output (with channel specification) (Selectable data table control operation)**	**SP0H**	S, n	Outputs the pulses from the specified channel according to the data table specified by S.	5				□	□		

○ : Available, ✕ : Not available, △ : Not available partially

*1) The elapsed value area differs depending on used channels.

Number	Name	Boolean	Operand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F175	**Pulse output (Linear interpolation)**	**SPSH**	S, n	Pulses are output from channel, in accordance with the designated data table, so that the path to the target position forms a straight line.	5				△ *3			
F176	**Pulse output (Circular interpolation)**	**SPCH**	S, n	Pulses are output from channel, in accordance with the designated data table, so that the path to the target position forms an arc.	5				△ *3			
Screen display instructions												
F180	**FP-e screen display registration**	**SCR**	S1, S2, S3, S4	Register the screen displayed on the FP-e.	9	○	×	×	×	×	×	×
F181	**FP-e screen display switching**	**DSP**	S	Specify the screen to be displayed on the FP-e.	3	○	×	×	×	×	×	×
Basic function instruction												
F182	**Time constant processing**	**FILTR**	S1, S2, S3, D	Executes the filter processing for the specified input.	9	×	×	○	△ *5	△ *4	×	×
F183	**Auxiliary timer (32-bit)**	**DSTM**	S, D	Turn on the specified output and R900D after 0.01 s. × set value.	7	○	○	○	○	○	○	○ *7
Data transfer instructions												
F190 P190	**Three 16-bit data move**	**MV3 PMV3**	S1, S2, S3, D	(S1)→(D), (S2)→(D+1), (S3)→(D+2)	10	×	×	○	○	○	○	○
F191 P191	**Three 32-bit data move**	**DMV3 PDMV3**	S1, S2, S3, D	(S1+1, S1)→(D+1, D), (S2+1, S2)→(D+3, D+2), (S3+1, S3)→(D+5, D+4)	16	×	×	○	○	○	○	○
Logic operation instructions												
F215 P215	**32-bit data AND**	**DAND PDAND**	S1, S2, D	(S1+1, S1) AND (S2+1, S2)→(D+1, D)	7	×	×	○	○	○	○	○
F216 P216	**32-bit data OR**	**DOR PDOR**	S1, S2, D	(S1+1, S1) OR (S2+1, S2)→(D+1, D)	12	×	×	○	○	○	○	○
F217 P217	**32-bit data XOR**	**DXOR PDXOR**	S1, S2, D	{$\underline{(S1+1,\ S1}$) AND $\overline{(S2+1,\ S2)}$} OR {(S1+1, S1) AND (S2+1, S2)}→(D+1, D)	12	×	×	○	○	○	○	○
F218 P218	**32-bit data XNR**	**DXNR PDXNR**	S1, S2, D	{$\underline{(S1+1,\ S1}$) AND (S2+1, S2)} OR {(S1+1, S1) AND (S2+1, S2)}→(D+1, D)	12	×	×	○	○	○	○	○
F219 P219	**Double word (32-bit) data unites**	**DUNI PDUNI**	S1, S2, S3, D	{(S1+1, S1) AND $\underline{(S3+1,\ S3)}$} OR {(S2+1, S2) AND (S3+1, S3)}→(D+1, D)	16	×	×	○	○	○	○	○
Data conversion instructions												
F230 P230	**Time data → second conversion**	**TMSEC PTMSEC**	S, D	The specified time data (a date and time) is changed to the second data.	6	×	×	○	△ *2	△ *6	△ *1	△ *1
F231 P231	**Second data→ time conversion**	**SECTM PSECTM**	S, D	The specified second data is changed into time data (a date and time).	6	×	×	○	△ *2	△ *6	△ *1	△ *1

○: Available, ×: Not available, △: Not available partially
*1) This instruction is available for FP2/FP2SH Ver. 1.5 or later.FP10SH cannot be used.
*2) This instruction is available for FPΣ 32k type.
*3) This instruction is available for FPΣ C32T2, C28P2, C32T2H and C28P2H.
*4) This instruction is only available for FP-X Ver.2.0 or later. *5) This instruction is available for FPΣ Ver. 3.10 or later.
*6) This instruction is available for FP-X Ver. 1.13 or later.
*7) This instruction is available for FP10SH Ver. 3.10 or later.

Num-ber	Name	Boolean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F235 P235	**16-bit binary data → Gray code conversion**	**GRY PGRY**	S, D	Converts the 16-bit binary data of “S” to gray codes, and the converted result is stored in the “D”.	6	×	×	○	○	○	○	○
F236 P236	**32-bit binary data → Gray code conversion**	**DGRY PDGRY**	S, D	Converts the 32-bit binary data of (S+1, S) to gray code, and the converted result is stored in the (D+1, D).	8	×	×	○	○	○	○	○
F237 P237	**16-bit gray code → binary data conversion**	**GBIN PGBIN**	S, D	Converts the gray codes of “S” to binary data, and the converted result is stored in the “D”.	6	×	×	○	○	○	○	○
F238 P238	**32-bit gray code → binary data conversion**	**DGBIN PDGBIN**	S, D	Converts the gray codes of (S+1, S) to binary data, and the converted result is stored in the (D+1, D).	8	×	×	○	○	○	○	○
F240 P240	**Bit line to bit column conversion**	**COLM PCOLM**	S, n, D	The values of bits 0 to 15 of “S” are stored in bit “n” of (D to DC+15).	8	×	×	○	○	○	○	○
F241 P241	**Bit column to bit line conversion**	**LINE PLINE**	S, n, D	The values of bit “n” of (S) to (S+15) are stored in bits 0 to 15 of “D”.	8	×	×	○	○	○	○	○
F250	**Binary data → ASCII conversion**	**BTOA**	S1, S2, n, D	Converts multiple binary data to multiple ASCII data.	12	×	×	○	△ *1	○	×	×
F251	**ASCII → binary data conversion**	**ATOB**	S1, S2, n, D	Converts multiple ASCII data to multiple binary data.	12	×	×	○	△ *1	○	×	×
F252	**ASCII data check**	**ACHK**	S1, S2, n	Checks the ASCII data strings to be used in F251 (ATOB) instruction.	10	×	×	○	△ *3	△ *2	×	×
Character strings instructions												
F257 P257	**Comparing character strings**	**SCMP**	S1, S2	These instructions compare two specified character strings and output the judgment results to a special internal relay.	10	×	×	○	○	○	○	○
F258 P258	**Character string coupling**	**SADD**	S1, S2, D	These instructions couple one character string with another.	12	×	×	○	○	○	○	○
F259 P259	**Number of characters in a character string**	**LEN**	S, D	These instructions determine the number of characters in a character string.	6	×	×	○	○	○	○	○
F260 P260	**Search for character string**	**SSRC**	S1, S2, D	The specified character is searched in a character string.	10	×	×	○	○	○	○	○
F261 P261	**Retrieving data from character strings (right side)**	**RIGHT**	S1, S2, D	These instructions retrieve a specified number of characters from the right side of the character string.	8	×	×	○	○	○	○	○
F262 P262	**Retrieving data from character strings (left side)**	**LEFT**	S1, S2, D	These instructions retrieve a specified number of characters from the left side of the character string.	8	×	×	○	○	○	○	○
F263 P263	**Retrieving a character string from a character string**	**MIDR**	S1, S2, S3, D	These instructions retrieve a character string consisting of a specified number of characters from the specified position in the character string.	10	×	×	○	○	○	○	○
F264 P264	**Writing a character string to a character string**	**MIDW**	S1, S2, D, n	These instructions write a specified number of characters from a character string to a specified position in the character string.	12	×	×	○	○	○	○	○
F265 P265	**Replacing character strings**	**SREP**	S, D, p, n	A specified number of characters in a character string are rewritten, starting from a specified position in the character string.	12	×	×	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

*1) This instruction is available for FPΣ 32k type.

*2) This instruction is only available for FP-X Ver.2.0 or later.

*3) This instruction is available for FPΣ Ver. 3.10 or later.

Num-ber	Name	Boolean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
Integer type data processing instructions												
F270 P270	**Maximum value (word data (16-bit))**	**MAX PMAX**	S1, S2, D	Searches the maximum value in the word data table between the “S1” and “S2”, and stores it in the “D”. The address relative to “S1” is stored in “D+1”.	8	△ *1	×	○	○	○	○	○
F271 P271	**Maximum value (double word data (32-bit))**	**DMAX PDMAX**	S1, S2, D	Searches for the maximum value in the double word data table between the area selected with “S1” and “S2”, and stores it in the “D”. The address relative to “S1” is stored in “D+2”.	8	△ *1	×	○	○	○	○	○
F272 P272	**Minimum value (word data (16-bit))**	**MIN PMIN**	S1, S2, D	Searches for the minimum value in the word data table between the area selected with “S1” and “S2”, and stores it in the “D”. The address relative to “S1” is stored in “D+1”.	8	△ *1	×	○	○	○	○	○
F273 P273	**Minimum value (double word data (32-bit))**	**DMIN PDMIN**	S1, S2, D	Searches for the minimum value in the double word data table between the area selected with “S1” and “S2”, and stores it in the “D”. The address relative to “S1” is stored in “D+2”.	8	△ *1	×	○	○	○	○	○
F275 P275	**Total and mean values (word data (16-bit))**	**MEAN PMEAN**	S1, S2, D	The total value and the mean value of the word data with sign from the area selected with “S1” to “S2” are obtained and stored in the “D”.	8	△ *1	×	○	○	○	○	○
F276 P276	**Total and mean values (double word data (32-bit))**	**DMEAN PDMEAN**	S1, S2, D	The total value and the mean value of the double word data with sign from the area selected with “S1” to “S2” are obtained and stored in the “D”.	8	△ *1	×	○	○	○	○	○
F277 P277	**Sort (word data (16-bit))**	**SORT PSORT**	S1, S2, S3	The word data with sign from the area specified by “S1” to “S2” are sorted in ascending order (the smallest word is first) or descending order (the largest word is first).	8	△ *1	×	○	○	○	○	○
F278 P278	**Sort (double word data (32-bit))**	**DSORT PDSORT**	S1, S2, S3	The double word data with sign from the area specified b “S1” ato “S2” are sorted in ascending order (the smallest word is first) or descending order (the largest word is first).	8	△ *1	×	○	○	○	○	○
F282 P282	**Scaling of 16-bit data**	**SCAL PSCAL**	S1, S2, D	The toutptu value Y is found for the input value X by performing scaling for the given data table.	8	△ *1	×	○	○	○	○	○
F283 P283	**Scaling of 32-bit data**	**DSCAL PDSCAL**	S1, S2, D	The toutptu value Y is found for the input value X by performing scaling for the given data table.	10	×	×	○	○	○	○	○
F284 P284	**Inclination output of 16-bit data**	**RAMP**	S1, S2, S3, D	Executes the linear output for the specified time from the specified initial value to the target value.	10	×	×	○	△ *2	△ *2	×	×
Integer type non-linear function instructions												
F285 P285	**Upper and lower limit control (16-bit data)**	**LIMT PLIMT**	S1, S2, S3, D	When S1>S3, S1→D When S1<S3, S2→D When S1<or = S3<or = S2, S3→D	10	△ *1	×	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

*1) This instruction is available for FP-e Ver.1.2 or later.

*2) This instruction is only available for FP-X Ver.2.0 or later, and FPΣ Ver. 3.10 or later.

Num-ber	Name	Boolean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F286 P286	**Upper and lower limit control (32-bit data)**	**DLIMT PDLIMT**	S1, S2, S3, D	When (S1+1, S1)>(S3+1, S3), (S1+1, S1)→(D+1, D) When (S2+1, S2)<(S3+1, S3), (S2+1, S2)→(D+1, D) When (S1+1, S1)<or = (S3+1, S3)<or = (S2+1, S2), (S3+1, S3)→(D+1, D)	16	△ *1	×	○	○	○	○	○
F287 P287	**Deadband control (16-bit data)**	**BAND PBAND**	S1, S2, S3, D	When S1>S3, S3–S1→D When S2<S3, S3–S2→D When S1<or = S3<or = S2, 0→D	10	△ *1	×	○	○	○	○	○
F288 P288	**Deadband control (32-bit data)**	**DBAND PDBAND**	S1, S2, S3, D	When (S1+1, S1)>(S3+1, S3), (S3+1, S3)–(S1+1, S1)→(D+1, D) When (S2+1, S2)<(S3+1, S3), (S3+1, S3)–(S2+1, S2)→(D+1, D) When (S1+1, S1)<or = (S3+1, S3)<or = (S2+1, S2),0→(D+1, D)	16	△ *1	×	○	○	○	○	○
F289 P289	**Zone control (16-bit data)**	**ZONE PZONE**	S1, S2, S3, D	When S3<0, S3+S1→D When S3=0, 0→D When S3>0, S3+S2→D	10	△ *1	×	○	○	○	○	○
F290 P290	**Zone control (32-bit data)**	**DZONE PDZONE**	S1, S2, S3, D	When (S3+1, S3)<0, (S3+1, S3)+(S1+1, S1)→(D+1, D) When (S3+1, S3)=0, 0→(D+1, D) When (S3+1, S3)>0, (S3+1, S3)+(S2+1, S2)→(D+1, D)	16	△ *1	×	○	○	○	○	○
BCD type real number operation instructions												
F300 P300	**BCD type sine operation**	**BSIN PBSIN**	S, D	SIN(S1+1, S1)→(D+1, D)	6	×	×	×	×	×	○	○
F301 P301	**BCD type cosine operation**	**BCOS PBCOS**	S, D	COS(S1+1, S1)→(D+1, D)	6	×	×	×	×	×	○	○
F302 P302	**BCD type tangent operation**	**BTAN PBTAN**	S, D	TAN(S1+1, S1)→(D+1, D)	6	×	×	×	×	×	○	○
F303 P303	**BCD type arcsine operation**	**BASIN PBASIN**	S, D	SIN^{-1}(S1+1, S1)→(D+1, D)	6	×	×	×	×	×	○	○
F304 P304	**BCD type arccosine operation**	**BACOS PBACOS**	S, D	COS^{-1}(S1+1, S1)→(D+1, D)	6	×	×	×	×	×	○	○
F305 P305	**BCD type arctangent operation**	**BATAN PBATAN**	S, D	TAN^{-1}(S1+1, S1)→(D+1, D)	6	×	×	×	×	×	○	○
Floating-point type real number operation instructions												
F309 P309	**Floating-point type data move**	**FMV PFMV**	S, D	(S+1, S)→(D+1, D)	8	○ *2	○ *2	○	○	○	○	○
F310 P310	**Floating-point type data addition**	**F+ PF+**	S1, S2, D	(S1+1, S1)+(S2+1, S2)→(D+1, D)	14	○ *2	○ *2	○	○	○	○	○
F311 P311	**Floating-point type data subtraction**	**F- PF-**	S1, S2, D	(S1+1, S1)–(S2+1, S2)→(D+1, D)	14	○ *2	○ *2	○	○	○	○	○
F312 P312	**Floating-point type data multiplication**	**F* PF***	S1, S2, D	(S1+1, S1)×(S2+1, S2)→(D+1, D)	14	○ *2	○ *2	○	○	○	○	○
F313 P313	**Floating-point type data division**	**F% PF%**	S1, S2, D	(S1+1, S1)÷(S2+1, S2)→(D+1, D)	14	○ *2	○ *2	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

*1) This instruction is available for FP-e Ver.1.2 or later.

*2) This instruction is available for FP-e Ver.1.21 or later, FP0 V2.1 or later.

Num-ber	Name	Boo-lean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F314 **P314**	**Floating-point type data sine operation**	**SIN** **PSIN**	S, D	SIN(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F315 **P315**	**Floating-point type data cosine operation**	**COS** **PCOS**	S, D	COS(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F316 **P316**	**Floating-point type data tangent operation**	**TAN** **PTAN**	S, D	TAN(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F317 **P317**	**Floating-point type data arcsine operation**	**ASIN** **PASIN**	S, D	SIN^{-1}(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F318 **P318**	**Floating-point type data arccosine operation**	**ACOS** **PACOS**	S, D	COS^{-1}(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F319 **P319**	**Floating-point type data arctangent operation**	**ATAN** **PATAN**	S, D	TAN^{-1}(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F320 **P320**	**Floating-point type data natural logarithm**	**LN** **PLN**	S, D	LN(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F321 **P321**	**Floating-point type data exponent**	**EXP** **PEXP**	S, D	EXP(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F322 **P322**	**Floating-point type data logarithm**	**LOG** **PLOG**	S, D	LOG(S+1, S)→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F323 **P323**	**Floating-point type data power**	**PWR** **PPWR**	S1, S2, D	(S1+1, S1) ^ (S2+1, S2)→(D+1, D)	14	○ *1	○ *1	○	○	○	○	○
F324 **P324**	**Floating-point type data square root**	**FSQR** **PFSQR**	S, D	$\sqrt{(S+1, S)}$→(D+1, D)	10	○ *1	○ *1	○	○	○	○	○
F325 **P325**	**16-bit integer data to floating-point type data conversion**	**FLT** **PFLT**	S, D	Converts the 16-bit integer data with sign specified by “S” to real number data, and the converted data is stored in “D”.	6	○ *1	○ *1	○	○	○	○	○
F326 **P326**	**32-bit integer data to floating-point type data conversion**	**DFLT** **PDFLT**	S, D	Converts the 32-bit integer data with sign specified by (S+1, S) to real number data, and the converted data is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F327 **P327**	**Floating-point type data to 16-bit integer con-version (the largest inte-ger not ex-ceeding the floating-point type data)**	**INT** **PINT**	S, D	Converts real number data specified by (S+1, S) to the 16-bit integer data with sign (the largest integer not exceeding the floating-point data), and the converted data is stored in “D”.	8	○ *1	○ *1	○	○	○	○	○
F328 **P328**	**Floating-point type data to 32-bit integer con-version (the largest inte-ger not ex-ceeding the floating-point type data)**	**DINT** **PDINT**	S, D	Converts real number data specified by (S+1, S) to the 32-bit integer data with sign (the largest integer not exceeding the floating-point data), and the converted data is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○

○: Available, ×: Not available, △: Not available partially

*1) This instruction is available for FP-e Ver.1.21 or later, FP0 V2.1 or later.

Num-ber	Name	Boolean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F329 P329	**Floating-point type data to 16-bit integer con-version (rounding the first decimal point down to integer)**	**FIX PFIX**	S, D	Converts real number data specified by (S+1, S) to the 16-bit integer data with sign (rounding the first decimal point down), and the converted data is stored in "D".	8	○ *1	○ *1	○	○	○	○	○
F330 P330	**Floating-point type data to 32-bit integer con-version (rounding the first decimal point down to integer)**	**DFIX PDFIX**	S, D	Converts real number data specified by (S+1, S) to the 32-bit integer data with sign (rounding the first decimal point down), and the converted data is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F331 P331	**Floating-point type data to 16-bit integer con-version (rounding the first decimal point off to integer)**	**ROFF PROFF**	S, D	Converts real number data specified by (S+1, S) to the 16-bit integer data with sign (rounding the first decimal point off), and the converted data is stored in "D".	8	○ *1	○ *1	○	○	○	○	○
F332 P332	**Floating-point type data to 32-bit integer con-version (rounding the first decimal point off to integer)**	**DROFF PDROFF**	S, D	Converts real number data specified by (S+1, S) to the 32-bit integer data with sign (rounding the first decimal point off), and the converted data is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F333 P333	**Floating-point type data round-ding the first decimal point down**	**FINT PFINT**	S, D	The decimal part of the real number data specified in (S+1, S) is rounded down, and the result is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F334 P334	**Floating-point type data round-ding the first decimal point off**	**FRINT PFRINT**	S, D	The decimal part of the real number data stored in (S+1, S) is rounded off, and the result is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F335 P335	**Floating-point type data sign changes**	**F+/- PF+/-**	S, D	The real number data stored in (S+1, S) is changed the sign, and the result is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F336 P336	**Floating-point type data absolute**	**FABS PFABS**	S, D	Takes the absolute value of real number data specified by (S+1, S), and the result (absolute value) is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F337 P337	**Floating-point type data degree → radian**	**RAD PRAD**	S, D	The data in degrees of an angle specified in (S+1, S) is converted to radians (real number data), and the result is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
F338 P338	**Floating-point type data radian → degree**	**DEG PDEG**	S, D	The angle data in radians (real number data) specified in (S+1, S) is converted to angle data in degrees, and the result is stored in (D+1, D).	8	○ *1	○ *1	○	○	○	○	○
Floating-point type real number data processing instructions												
F345 P345	**Floating-point type data compare**	**FCMP PFCMP**	S1, S2	(S1+1, S1)>(S2+1, S2)→ R900A: on (S1+1, S1)=(S2+1, S2)→ R900B on (S1+1, S1)<(S2+1, S2)→ R900C: on	10	✕	✕	○	○	○	○	○
F346 P346	**Floating-point type data band compare**	**FWIN PFWIN**	S1, S2, S3	(S1+1, S1)>(S3+1, S3)→ R900A: on (S2+1, S2)<or =(S1+1, S1)<or =(S3+1, S3) → R900B on (S1+1, S1)<(S2+1, S2)→ R900C: on	14	✕	✕	○	○	○	○	○

○: Available, ✕: Not available, △: Not available partially

*1) This instruction is available for FP-e Ver.1.21 or later, FP0 V2.1 or later.

Num-ber	Name	Boolean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
F347 P347	**Floating-point type data upper and lower limit control**	**FLIMT PFLIMT**	S1, S2, S3, D	When (S1+1, S1)>(S3+1, S3), (S1+1, S1) →(D+1, D) When (S2+1, S2)<(S3+1, S3), (S2+1, S2) → (D+1, D) When (S1+1, S1)<or = (S3+1, S3)<or =(S2+1, S2), (S3+1, S3)→(D+1, D)	17	×	×	○	○	○	○	○
F348 P348	**Floating-point type data dead-band control**	**FBAND PFBAND**	S1, S2, S3, D	When (S1+1, S1)>(S3+1, S3), (S3+1, S3)–(S1+1, S1)→(D+1, D) When (S2+1, S2)<(S3+1, S3), (S3+1, S3)–(S2+1, S2)→ (D+1, D) When (S1+1, S1)<or = (S3+1, S3)<or =(S2+1, S2), 0.0→(D+1, D)	17	×	×	○	○	○	○	○
F349 P349	**Floating-point type data zone control**	**FZONE PFZONE**	S1, S2, S3, D	When (S3+1, S3)<0.0, (S3+1, S3)+(S1+1, S1)→(D+1, D) When (S3+1, S3)=0.0, 0.0→ (D+1, D) When (S3+1, S3)>0.0, (S3+1, S3)+(S2+1, S2) →(D+1, D)	17	×	×	○	○	○	○	○
F350 P350	**Floating-point type data maxi-mum value**	**FMAX PFMAX**	S1, S2, D	Searches the maximum value in the real number data table between the area selected with “S1” and “S2”, and stores it in the (D+1, D). The address relative to “S1” is stored in (D+2).	8	×	×	×	×	×	○	○
F351 P351	**Floating-point type data mini-mum value**	**FMIN PFMIN**	S1, S2, D	Searches the minimum value in the real number data table between the area selected with “S1” and “S2”, and stores it in the (D+1, D). The address relative to “S1” is stored in (D+2).	8	×	×	×	×	×	○	○
F352 P352	**Floating-point type data total and mean values**	**FMEAN PFMEAN**	S1, S2, D	The total value and the mean value of the real number data from the area selected with “S1” to “S2” are obtained. The total value is stored in the (D+1, D) and the mean value is stored in the (D+3, D+2).	8	×	×	×	×	×	○	○
F353 P353	**Floating-point type data sort**	**FSORT PFSORT**	S1, S2, S3	The real number data from the area speciified by “S1” to “S2” are stored in ascending order (the smallest word is first) or descending order (the largest word is first).	8	×	×	×	×	×	○	○
F354 P354	**Scaling of real number data**	**FSCAL PFSCAL**	S1, S2, D	Scaling (linearization) on a real number data table is performed, and the output (Y) to an input value (X) is calculated.	12	×	×	○	△ *2	△ *3	△ *1	△ *1

○: Available, ×: Not available, △: Not available partially

*1) This instruction is available for FP2/FP2SH Ver. 1.5 or later. FP10SH cannot be used.

*2) This instruction is available for FPΣ 32k type.

*3) This instruction is available for FP-X Ver. 1.13 or later.

Num-ber	Name	Boolean	Ope-rand	Description	Steps	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH/FP10SH
Time series processing instruction												
F355	**PID processing**	**PID**	S	PID processing is performed depending on the control value (mode and parameter) specified by (S to S+2) and (S+4 to S+10), and the result is stored in the (S+3).	4	○	○ *3	○	○	○	○	○
F356	**Eaay PID**	**EZPID**	S1, S2, S3, S4	Temperature control (PID) can be easily performed using the image of a temperautre controller.	10	×	×	○	△ *2	△ *2	×	×
Compare instructions												
F373 P373	**16-bit data revision detection**	**DTR PDTR**	S, D	If the data in the 16-bit area specified by “S” has changed since the previous execution, internal relay R9009 (carry flag) will turn on. “D” is used to store the data of the previous execution.	6	×	×	○	○	○	○	○
F374 P374	**32-bit data revision detection**	**DDTR PDDTR**	S, D	If the data in the 32-bit area specified by (S+1, S) has changed since the previous execution, internal relay R9009 (carry flag) will turn on. (D+1, D) is used to store the data of the previous execution.	6	×	×	○	○	○	○	○
Index register bank processing instructions												
F410 P410	**Setting the index regis-ter bank number**	**SETB PSETB**	n	Index register (I0 to ID) bank number change over.	4	×	×	×	×	×	×	○
F411 P411	**Changing the index regis-ter bank number**	**CHGB PCHGB**	n	Index register (I0 to ID) bank number change over with remembering preceding bank number.	4	×	×	×	×	×	×	○
F412 P412	**Restoring the index regis-ter bank number**	**POPB PPOPB**	-	Changes index register (I0 to ID) bank number back to the bank before F411 (CHGB)/P411 (PCHGB) instruction.	2	×	×	×	×	×	×	○
File register bank processing instructions												
F414 P414	**Setting the file register bank number**	**SBFL PSBFL**	n	File register bank number change over.	4	×	×	×	×	×	×	△ *1
F415 P415	**Changing the file register bank number**	**CBFL PCBFL**	n	File register bank number change over with remembering preceding bank number.	4	×	×	×	×	×	×	△ *1
F416 P416	**Restoring the file register bank number**	**PBFL PPBFL**	-	Changes file register bank number back to the bank before F415 (CBFL)/P415 (PCBFL) instruction.	2	×	×	×	×	×	×	△ *1

○: Available, ×: Not available, △: Not available partially

*1) This instruction is not available for FP10SH.

*2) This instruction is available for FP-X V.1.20 or later, and FPΣ 32k type.

*3) This instruction is available for FP0 V2.1 or later.

5.4 Table of Error codes

■ Difference in ERROR display

There are differences in the way errors are displayed depending on the model.

Model	Display		Display method
FP1,FP-M,FP2,FP3,FP10SH	LED	ERROR.	Continually lit
FP Σ,FP0, FP0R, FP-X	LED	ERROR/ALARM	Flashes/contunually lit
FP-e	Screen display	ERR.	Continually lit

■ Error Confirmation When ERROR Turns ON

When the “ERROR” on the control unit (CPU unit) turns on or flashes, a self-diagnostic error or syntax check error has occurred. Confirm the contents of the error and take the appopriate steps.

-Error Confirmation Method

Procedure:1.Use the programming tool software to call up the error code.
By executing the “STATUS DISPLAY”, the error code and content of error are displayed.
2.Check the error contents in the table of error codes using the error code ascertained above.

-Syntax check error

This is an error detected by the total check function when there is a syntax error or incorrect setting written in the program. When the mode selector is switched to the RUN mode, the total check function automatically activates and eliminates the possibility of incorrect operation from syntax errors in the program.

When a syntax check error is detected

-ERROR turns on or flashes.
-Operation will not begin even after swirching to the RUN mode.
-Remote operation cannot be used to change to RUN mode.

Clearing a syntax check error

By changing to the PROG.mode, the error will clear and the ERROR will turn off.

Steps to take for syntax error

Change to the PROG. mode, and then execute the total check function while online mode with the programming tool connected. This will call up the content of error and the address where the error occurred.
Correct the program while referring to the content of error.

-Self-diagnostic Error
This error occurs when the control unit (CPU unit) self-diagnostic function detects the occurrence of an abnormality in the system. The self-diagnostic function monitors the memory abnormal detection, I/O abnomal detection, and other devices.

When a self-diagnostic error occurs
- The ERROR turns on or flashes.
- The operation of the control unit (CPU unit) might stop depending on the contect of error and the system
register setting.
- The error codes will be stored in the special data register DT9000(DT90000).
- In the case of operation error, the error address will stored in the DT9017(DT90017) and DT9018(DT90018).

Clearing the self-diagnostic error
At the “STATUS DISPLAY”, execute the “error clear”. Error codes 43 and higher can be cleared.
-You can use the initialize/test switch to clear an error. However, this will also clear the contents of operation memory.
-Errors can also be cleared by turning off and on the power while in the PROG.mode.
However, the contents of operation memory, not stored with the hold type data, will also be cleared.
-The error can also be cleared depending on the self-diagnostic error set instruction F148(ERR).

Steps to take for self-diagnostic error
The steps to be taken will differ depending on the error contents. For more details, use the error code obtained above and consult the table of aself-diagnostic error codes.

■ MEWTOCOL-COM Transmission Errors
These are error codes from a PC or other computer device that occur during an abnormal response when communicating with a PLC using MEWTOCOL-COM.

■ Table of Syntax Check Error

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FP Σ	FP-X	FP2	FP2SH	FP10SH
E1	**Syntax error**	Stops	A program with a syntax error has been written. ⇒ Change to PROG. mode and correct the error.	A	A	A	A	A	A	A	A
E2 (Note)	**Duplicated output error**	Stops	Two or more OT(Out) instructions and KP(Keep) instructions are programmed using the same relay.Also occurs when using the same timer/counter number. ⇒ Change to PROG. mode and correct the program so that one relay is not used for two or more OT instructions,Or, set the duplicated output to “enable” in system register20. A timer/counter instructon double definition error will be detected even if double output permission has been selected.	A	A	A	A	A	A	A	A
E3	**Not paired error**	Stops	For instructions which must be used in a pair such as jump (JP and LBL), one instruction is either missing or in an incorrect position. ⇒ Change to PROG. mode and enter the two instructions which must be used in a pair in the correct positions.	A	A	A	A	A	A	A	A
E4	**Parameter mismatch error**	Stops	An instruction has been written which does not agree with system register settings. For example, the number setting in a program does not agree with the timer/counter range setting. ⇒ Change to PROG. mode, check the system register settings, and change so that the settings and the instruction agree.	A	A	A	A	A	A	A	A
E5 (Note)	**Program area error**	Stops	An instruction which must be written in a specific area (main program area or subprogram area) has been written to a different area (for example, a subroutine SUB to RET is placed before an ED instruction). ⇒ Change to PROG. mode and enter the instruction into the correct area.	A	A	A	A	A	A	A	A

A:Available

Note) This error is also detected if you attempt to execute a rewrite containing a syntax error during RUN. In this case, nothing will be written to the CPU and operation will continue.

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FP Σ	FP-X	FP2	FP2SH	FP10SH
E6	**Compile memory full error**	Stops	The program is too large to compile in the program memory. ⇒ Change to PROG. mode and reduce the total number of steps for the program. -FP10SH If memory expansion is possible,compilation will become possible when the memory is expanded.	A	A	A	A	A		A	A
E7	**High-level instruction type error**	Stops	In the program, high-level instructions, which execute in every scan and at the leading edge of the trigger, are programmed to be triggered by one contact. (e.g. F0 (MV) and P0 (PMV) are programmed using the same trigger continuously.) ⇒ Correct the program so that the high-level instructions executed in every scan and only at the leading edge are triggered separately.			A	A	A	A	A	A
E8	**High-level instruction operand combina-tion error**	Stops	There is an incorrect operand in an instruction which requires a specific combination operands (for example, the operands must all be of a certain type). ⇒ Enter the correct combination of operands.	A	A	A	A	A	A	A	A
E9	**No program error**	Stops	Program may be damaged. ⇒Try to send the program again.							A	A
E10	**Rewrite during RUN syntax error**	Conti-nues	When inputting with the programming tool software,a delection,addition or change of order of an instruction(ED,LBL,SUB,RET,INT,IRET,SSTP ,and STPE) that cannot perform a rewrite during RUN is being attempted. Nothing is written to the CPU.						A	A	A

A:Available

■ Table of Self-Diagnostic Error

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FP Σ	FP-X	FP2	FP2SH	FP10SH
E20	**CPU error**	Stops	Probably a hardware abnormality ⇒Please contact your dealer.						A	A	A
E21	**RAM error1**	Stops	Probably an abnormality in the internal RAM. ⇒Please contact your dealer.						A	A	A
E22	**RAM error2**										
E23	**RAM error3**										
E24	**RAM error4**										
E25	**RAM error5**										
E25	**Master memory model unmatch error**	Stops	The models of master memories are different. Use the master memories created with the same model.					A *1)			
E26	**User's ROM error**	Stops	FP-e,FP0,FP0R,FP Σ ,and FP1 C14,C16:Probably a hardware abnormality. ⇒ Please contact your dealer. FP-X: When the master memory cassette is mounted, the master memor cassette may be damaged. Remove the master memory, and check whether the ERROR turns off. When the ERROR turned off, rewrite the master memory as its contents are damaged, and use it again. When the ERROR does not turn off, please contact your dealer. FP1 C24,C40,C56,C72,and FP-M: Probably an abnormality in the memory unit ⇒Program the memory unit again and try to operate. If the same error is detected, try to operate with another memory unit. FP2,FP2SH,FP10SH,and FP3: There may be a problem with the installed ROM. -ROM is not installed. -ROM contens are damaged. -Program size stored on the ROM is larger than the capacity of the ROM ⇒Check the contents of the ROM	A	A	A	A	A	A	A	A
E27	**Unit installation error**	Stops	Units installed exceed the limitations.(i.e.,4 or more link units) ⇒ Turn off the power and re-configure units referring to the hardware manual.			A	A	A	A	A	A
E28	**System register error**	Stops	Probably an abnormality in the system register. ⇒ Check the system register setting or initialize the system registers.						A		

*1) This error occurs on FP-X Ver2.0 or later.

A:Available

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH	FP10SH
E29	**Configu-ration parameter error**	Stops	A parameter error was detected in the MEWNET-W2 configuration area. Set a correct parameter.						A	A	
E30	**Interrupt error 0**	Stops	Probably a hardware abnormality. ⇒ Please contact your dealer.								
E31	**Interrupt error 1**	Stops	An interrupt occurred without an interrupt request . A hardware problem or error due to noise is possible. ⇒ Turn off the power and check the noise conditions.	A	A	A	A	A	A	A	A
E32	**Interrupt error 2**	Stops	There is no interrupt program for an interrupt which occurred. ⇒ Check the number of the interrupt program and change it to agree with the interrrupt request..	A	A	A	A	A	A	A	A
E33	**Multi-CPU data unmatch error**	CPU2 Stops	This error occurs when a FP3/FP10SH is used as CPU2 for a multi-CPU system. ⇒Refer to "Multi-CPU system Manual".							A	A
E34	**I/O status error**	Stops	An abnormal unit is installed. -FPΣ, FP0R(FP0R mode),FP-X, FP2,FP2SH and FP10SH: Check the contents of special data register DT90036 and locate the abnormal unit.Then turn off the power and replace the unit with a new one. -FP3: Check the contents of special data register DT9036 and locate the abnormal unit. Then turn off the power and replace the unit with a new one.			A	A	A		A	A
E35	**MEWNET-F slave illegal unit error**	Stops	A unit, which cannot be installed on the slave station of the MEWNET-F link system,is installed on the slave station. ⇒Remove the illegal unit from the slave station.						A	A	A
E36	**MEWNET-F (remore I/O) limitation error**	Stops	The number of slots or I/O points used for MEWNET-F(remote I/O) system exceeds the limitation. ⇒Re-configure the system so that the number of slots and I/O points is within the specified range.						A	A	A
E37	**MEWNET-F I/O mapping error**	Stops	I/O overlap or I/O setting that is over the range is detected in the allocated I/O and MEWNET-F I/O map. ⇒Re-configure the I/O map correctly						A	A	A

A:Available

Error code	Name	Operation status	Description and steps to take	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH	FP10SH
E38	**MEWNET-F slave I/O terminal mapping error**	Stops	I/O mapping for remote I/O terminal boards,remote I/O terminal units and I/O link is not correct. ⇒Re-configure the I/O map for slave stations according to the I/O points of the slave stations.						A	A	A
E39	**IC card read error**	Stops	When reading in the program from the IC memory card(due to automatic reading because of the dip switch setting or program switching due to F14(PGRD) instruction): - IC memory card is not installed. - There is no program file or it is damaged. - Writing is disabled. - There is an abnormality in the AUTOEXEC.SPG file. - Program size stored on the card is larger than the capacity of the CPU. ⇒Install an IC memory card that has the program proterly recorded and execute the read once again.							A	A
E40	**I/O error**	Sele-ctable	Abnormal I/O unit. FPΣ, FP-X: Check the contents of special data register DT90002 and abnormal FPΣ expansion unit (application cassette for FP-X). Then check the unit. FP2 and FP2SH: Check the contents of special data registers DT90002,DT90003 and abnormal I/O unit.Then check the unit. Selection of operation status using system register21: -to continue operation,set 1 -to stop operation,set 0 Verification is possible in FPWIN GR/Pro at"I/O error" in the status display function. MEWNET-TR communication error FP3 and FP10SH: Check the contents of special data registers(FP3:DT9002,DT9003,FP10SH:DT90002,DT90003) and the erroneous master unit and abnormal I/O unit. Then check the unit. Selection of operation status using system register21: -to continue operation,set 1 -to stop operation,set 0 Verification is possible in FPWIN GR/Pro at"I/O error" in the status display function.				A	A	A	A	A

A:Available

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FP Σ	FP-X	FP2	FP2SH	FP10SH
E41	**Intelligent unit error**	Selec-table	An abnormality in an intelligent unit. FPΣ, FP-X: Check the contetns of special data register "DT90006" and locate the abnormal FP intelligent unit (application cassette for FP-X). FP2,FP2SH,and FP10SH: Check the contents of special data registers DT90006,DT90007 and locate the abnormal intelligent unit.Then check the unit referring to its manual.. Selection of operation status using system register22: -to continue operation,set 1 -to stop operation,set 0 FP3: Check the contents of special data registers DT9006,DT9007 and locate the abnormal intelligent unit.Then check the unit referring to its manual.. Selection of operation status using system register22: -to continue operation,set 1 -to stop operation,set 0 Verification is possible in FPWIN GR/Pro at"I/O error" in the status display function.				A	A	A	A	A
E42	**I/O unit verify error**	Selec-table	I/O unit(Expansion unit) wiring condition has changed compared to that at time fo power-up. ⇒ Check the contents of special data register (FP0: DT9010, FPΣ, FP-X: DT90010,DT90011) and locate the erroneous expansion unit. It checks whether an expansion connector is in agreement. ⇒ Check the contents of special data register (FP2,FP2SH,and FP10SH:DT90010,DT90011,FP3 DT9010,DT9011) Selection of operation status using system register23: -to continue operation,set 1 -to stop operation,set 0 Verification is possible in FPWIN GR/Pro at"I/O error" in the status display function.		A	A	A	A	A	A	A

A:Available

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FP Σ	FP-X	FP2	FP2SH	FP10SH
E43	**System watching dog timer error**	Selec-table	Scan time required for program execution exceeds the setting of the system watching dog timer. ⇒ Check the program and modify it so that the program can execute a scan within the specified time. Selection of operation status using system register24: -to continue operation,set 1 -to stop operation,set 0							A	A
E44	**Slave staiton connecting time error for MEWNET-F system**	Selec-table	The time required for slave station connection exceeds the setting of the system register 35. Selection of operation status using system register25: -to continue operation,set 1 -to stop operation,set 0						A	A	A
E45	**Operation error**	Selec-table	Operation became impossible when a high-level instruction was executed. Selection of operation status using system register26: -to continue operation,set K1 -to stop operation,set K0 The address of operation error can be confirmed in either special data registers DT9017 and DT9018, or DT90017 and DT90018. (It varies according to the model to be used.) DT9017, DT9018: FP-e, FP0, FP0R(FP0 mode) DT90017, DT90018: FPΣ, FP-X, FP0R(FP0R mode), FP2, FP2SH, FP10SH Verification is possible in FPWIN GR/Pro at"I/O error" in the status display function.	A	A	A	A	A	A	A	A

A:Available

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FP Σ	FP-X	FP2	FP2SH	FP10SH
E46	**Remote I/O commu-nication error**	Selec-table	S-LINK error Occurs only in FP0-SL1 When one of the S-LINK errors (ERR1, 3 or 4) has been deteced,error code E46 (remote I/O (S-LINK) communication error) is stored. Selection of operation status using system register27: -to continue operation,set K1 -to stop operation,set K0		A						
		Selec-table	MEWNET-F communication error A communication abnormally was caused by a transmission cable or during the power-down of a slave station. FP2, FP2SH, and FP10SH: Check the contents of special data registers DT90131 to DT90137 and locate the abnormal slave station and recover the communication condition. FP3: Check the contents of special data registers DT9131 to DT9137 and locate the abnormal slave station and recover the communication condition. Selection of operation status using system register27: -to continue operation,set K1 -to stop operation,set K0						A	A	A
E47	**MEW-NET-F attribute error**	Selec-table	In the unit on the slave station, an abnormallty such as: -missing unit -abnormal intelligent unit was detected. FP2, FP2SH, and FP10SH: Check the contents of special data registers DT90131 to DT90137 and locate the abnormal slave station and recover the slave condition. FP3: Check the contents of special data registers DT9131 to DT9137 and locate the abnormal slave station and recover the slave condition. Selection of operation status using system register28: -to continue operation,set 1 -to stop operation,set 0						A	A	A
E49	**Expansion unit power supply sequence error**	Stops	The power supply for the expansion unit was turned on after the control unit. Turn on the power supply for the expansion unit at the same time or before the control unit is turend on.					A			
E50	**Backup battery errror**	Conti-nues	The voltage of the backup battery lowered or the backup battery of conrol unit is not installed. ⇒ Check the installation of the backup battery and then replace battery if necessary. By setting the system register 4, you can disregard this self-diagnostic error.				A	A	A	A	A

Error code	Name	Opera-tion status	Description and steps to take	FP-e	FP0	FP0R	FPΣ	FP-X	FP2	FP2SH	FP10SH
E51	**MEWNET-F terminal station error**	Conti-nues	Terminal station setting was not properly performed. Check stations at both ends of the communication path,and set them in the terminal station using the dip switches.						A	A	A
E52	**MEWNET-F I/O update synchro-nous error**	Conti-nues	Set the INITIALIZE/TEST selecto1inmjvbgycfrde892 r to the INITIALIZE position while keeping the mode selector in the RUN position.If the same error occurs after this,please contact your dealer.						A	A	A
E53	**Multi-CPU I/O regis-tration error (CPU2 only)**	Conti-nues	Abnormality was detected when the multi-CPU system ws used. Please contact your dealer.								A
E54	**IC memory card back-up battery error**	Conti-nues	The voltage of the backup battery for the IC memory card lowered. The BATT.LED does not turn on. Charge or replace the backup battry of IC memory card.(The contents of the IC memory card cannot be guaranteed.)							A	A
E55	**IC memory card back-up battery error**	Cont-inues	The voltage of the backup battery for IC memory card lowers.The BATT.LED does not turn on. Charge or replace the backup battery of IC memory card. (The contents of the IC memory card cannot be guaranteed.)							A	A
E56	**Incompat-ible IC memory card error**	Cont-inues	The IC memory card installed is not compatible. Replace the IC memory card compatible with FP2SH/FP10SH.							A	A
E57	**No unit for the configu-ration**	Conti-nues	MEWNET-W2/MCU The MEWNET-W2 link unit or MCU(Multi communication unit) is not installed in the slot specified using the configuration data. Either install a unit in the specified slot or change the parameter.						A	A	
E100 to E199	**Self-diagnostic error set by F148 (ERR)/P148 (PERR) instruction**	Stop	The error specified by the F148 (ERR)/P148(PERR) instruction is occurred. ⇒ Take steps to clear the error condition according to the specification you chose.	A	A	A	A	A	A		
E200 to E299		Conti-nues		A	A	A	A	A	A		

A:Available

■ Table of MEWTOCOL-COM Communication Error

Error code	Name	Description
!21	**NACK error**	Link system error
!22	**WACK error**	Link system error
!23	**Unit No. overlap**	Link system error
!24	**Transmission format error**	Link system error
!25	**Link unit hardware error**	Link system error
!26	**Unit No. setting error**	Link system error
!27	**No support error**	Link system error
!28	**No response error**	Link system error
!29	**Buffer closed error**	Link system error
!30	**Time-out error**	Link system error
!32	**Transmission impossible error**	Link system error
!33	**Communication stop**	Link system error
!36	**No destination error**	Link system error
!38	**Other communication error**	Link system error
!40	**BCC error**	A transfer error occurred in the received data.
!41	**Format error**	A command was received that does not fit the format.
!42	**No support error**	A command was received that is not supported.
!43	**Multiple frames procedure error**	A different command was received when processing multiple frames.
!50	**Link setting error**	A route number that does not exist was spacified. Verify the route number by designating the transmission station.
!51	**Transmission time-out error**	Transmission to anather device not possible because transmissition buffer is congested.
!52	**Transmit disable error**	Transmission processing to another device is not possible.(Link unit runaway,etc.)
!53	**Busy error**	Command process cannot be received because of multiple frame processing.Or,cannot be received because command being processed is congested.
!60	**Parameter error**	Content of spacified parameter does not exist or cannot be used.
!61	**Data error**	There was a mistake in the contact,data area,data number desigination,size designation,range,or format designation.
!62	**Registration over error**	Operation was does when number of registrations was exceeded or when there was no registration.
!63	**PC mode error**	PC command that cannot be processed was executed during RUN mode.

Error code	Name	Description
!64	**External memory error**	An abnormality occurred when loading RAM to ROM/IC memory card.There may be a problem with the ROM or IC memory card. -When loading,the specified contents exceeded the capacity. -Write error occurs. -ROM or IC memory card is not installed. -ROM or IC memory card does not conform to specifications -ROM or IC memory card board is not installed.
!65	**Protect error**	A program or system register write operation was executed when theb protect mode (password setting or DIP switch,etc.)or ROM operation mode was being used.
!66	**Address error**	There was an error in the code format of the address data. Alsi.when exceeded or insufficient of address data,there was a mistake in the range designation.
!67	**No program error and No data error**	Cannot be read because there is no program in the program area or the memory contains an error.Or,reading was attempted of data that was not registered.
!68	**Rewrite during RUN error**	When inputting with programming tool software,editing of an instruction (ED,SUB,RET,INT,IRET,SSTP,and STPE) that cannot perform a rewrite during RUN is being attempted. Nothing is written to the CPU.
!70	**SIM over error**	Program area was exceeded during a program write process.
!71	**Exclusive access control error**	A command that cannot be processed was executed at the same time as a command being processed.

5.5 MEWTOCOL-COM Communication Commands

Table of MEWTOCOL-COM commands

Command name	Code	Description
Read contact area	RC (RCS) (RCP) (RCC)	Reads the on and off status of contact. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Write contact area	WC (WCS) (WCP) (WCC)	Turns contacts on and off. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Read data area	RD	Reads the contents of a data area.
Write data area	WD	Writes data to a data area.
Read timer/counter set value area	RS	Reads the value set for a timer/counter.
Write timer/counter set value area	WS	Writes a timer/counter setting value.
Read timer/counter ellapsed value area	RK	Reads the timer/counter elapsed value.
Write timer/counter elapsed value area	WK	Writes the timer/counter elapsed value.
Register or Reset contacts monitored	MC	Registers the contact to be monitored.
Register or Reset data monitored	MD	Registers the data to be monitored.
Monitoring start	MG	Monitors a registered contact or data using the code “MC or MD”.
Preset contact area (fill command)	SC	Embeds the areaof a specified range in a 16-point on and off pattern.
Preset data area (fill command)	SD	Writes the same contents to the data area of a specified range.
Read system register	RR	Reads the contents of a system register.
Write system register	WR	Specifies the contents of a system register.
Read the status of PLC	RT	Reads the specifications of the programmable controller and error codes if an error occurs.
Remote control	RM	Switches the operation mode of the programmable controller.
Abort	AB	Aborts communication.

5.6 Hexadecimal/Binary/BCD

Decimal	Hexadecimal	Binary data	BCD data (Binary Coded Decimal)
0	0000	00000000 00000000	0000 0000 0000 0000
1	0001	00000000 00000001	0000 0000 0000 0001
2	0002	00000000 00000010	0000 0000 0000 0010
3	0003	00000000 00000011	0000 0000 0000 0011
4	0004	00000000 00000100	0000 0000 0000 0100
5	0005	00000000 00000101	0000 0000 0000 0101
6	0006	00000000 00000110	0000 0000 0000 0110
7	0007	00000000 00000111	0000 0000 0000 0111
8	0008	00000000 00001000	0000 0000 0000 1000
9	0009	00000000 00001001	0000 0000 0000 1001
10	000A	00000000 00001010	0000 0000 0001 0000
11	000B	00000000 00001011	0000 0000 0001 0001
12	000C	00000000 00001100	0000 0000 0001 0010
13	000D	00000000 00001101	0000 0000 0001 0011
14	000E	00000000 00001110	0000 0000 0001 0100
15	000F	00000000 00001111	0000 0000 0001 0101
16	0010	00000000 00010000	0000 0000 0001 0110
17	0011	00000000 00010001	0000 0000 0001 0111
18	0012	00000000 00010010	0000 0000 0001 1000
19	0013	00000000 00010011	0000 0000 0001 1001
20	0014	00000000 00010100	0000 0000 0010 0000
21	0015	00000000 00010101	0000 0000 0010 0001
22	0016	00000000 00010110	0000 0000 0010 0010
23	0017	00000000 00010111	0000 0000 0010 0011
24	0018	00000000 00011000	0000 0000 0010 0100
25	0019	00000000 00011001	0000 0000 0010 0101
26	001A	00000000 00011010	0000 0000 0010 0110
27	001B	00000000 00011011	0000 0000 0010 0111
28	001C	00000000 00011100	0000 0000 0010 1000
29	001D	00000000 00011101	0000 0000 0010 1001
30	001E	00000000 00011110	0000 0000 0011 0000
31	001F	00000000 00011111	0000 0000 0011 0001
.	.	.	.
.	.	.	.
.	.	.	.
63	003F	00000000 00111111	0000 0000 0110 0011
.	.	.	.
.	.	.	.
.	.	.	.
255	00FF	00000000 11111111	0000 0010 0101 0101
.	.	.	.
.	.	.	.
.	.	.	.
9999	270F	00100111 00001111	1001 1001 1001 1001

5.7 ASCII Codes

								b7								
								b6	0	0	0	0	1	1	1	1
								b5	0	0	1	1	0	0	1	1
								b4	0	1	0	1	0	1	0	1
b7	b6	b5	b4	b3	b2	b1	b0	R \ C	0	1	2	3	4	5	6	7
				0	0	0	0	0	NUL	DEL	SPACE	0	@	P	`	p
				0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
				0	0	1	0	2	STX	DC2	"	2	B	R	b	r
				0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
				0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
				0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
				0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
				0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
				1	0	0	0	8	BS	CAN	(	8	H	X	h	x
				1	0	0	1	9	HT	EM	)	9	I	Y	i	y
				1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
				1	0	1	1	B	VT	ESC	+	;	K	[	k	{
				1	1	0	0	C	FF	FS	,	<	L	¥	l	|
				1	1	0	1	D	CR	GS	-	=	M	]	m	}
				1	1	1	0	E	SO	RS	.	>	N	^	n	~
				1	1	1	1	F	SI	US	/	?	O	_	o	DEL

Record of changes

Manual No.	Date	Description of changes
ARCT1F313E/ ACG-M313E	MAR.2000	First edition
ARCT1F313E-1/ ACG-M313E-1	MAY.2000	2nd edition
ARCT1F313E-2/ ACG-M313E-2	SEP.2000	3rd edition
ARCT1F313E-3/ ACG-M313E-3	JUN.2003	4th edition Additions: FPSIGMA, FP-e
ARCT1F313E-4/ ACG-M313E-4	JUL.2003	5th edition
ARCT1F313E-5/ ACG-M313E-5	JUL.2004	6th edition Addition & New programming: ICTL, F4, F159, F161, F230, F231, F354
ARCT1F313E-6/ ACG-M313E-6	AUG.2004	7th edition PDF only
ARCT1F313E-7/ ACG-M313E-7	OCT.2004	8th edition PDF only
ARCT1F313E-8/ ACG-M313E-8	JUL.2005	9th edition PDF only
ARCT1F313E-9/ ACG-M313E-9	OCT.2005	10th edition PDF only Addition & New programming: STF, ANF, ORF, F145, F146, F356 Addition: FPSIGMA 32K type Chapter 1 & 8 SYS1 Chapter 4.8, 4.9, 4.10
ARCT1F313E-10/ ACG-M313E-10	DEC.2005	11th edition Addition: FP-X transistor type New programming: F182, F252, F284
ARCT1F313E-11/ ACG-M313E-11	JUL.2006	12th edition

Manual No.	Date	Description of changes
ARCT1F313E-12/ ACG-M313E-12	JUL.2006	13th edition
ARCT1F313E-13/ ACG-M313E-13	SEP.2006	14th edition
ARCT1F313E-14/ ACG-M313E-14	MAR.2007	15th edition
ARCT1F313E-15/ ACG-M313E-15	JAN.2008	16th edition
ARCT1F313E-16/ ACG-M313E-16	NOV.2008	17th edition Change in Corporate name
ARCT1F313E-17/ ACG-M313E-17	JUL.2009	18th edition
ARCT1F313E-18	AUG.2011	19th edition Change in Corporate name

Please contact ....

Panasonic Electric Works SUNX Co., Ltd.

- Overseas Sales Division (Head Office): 2431-1 Ushiyama-cho, Kasugai-shi, Aichi, 486-0901, Japan
- Telephone: +81-568-33-7861 ■ Facsimile: +81-568-33-8591

panasonic-electric-works.net/sunx

Europe Headquarter: Panasonic Electric Works Europe AG

- Head Office: Rudolf-Diesel-Ring 2, D-83607 Holzkirchen, Germany
- Telephone: +49-8024-648-0

US Headquarter: Panasonic Electric Works Corporation of America

- Head Office: 629 Central Avenue New Providence, New Jersey 07974 USA
- Telephone: +1-908-464-3550

ARCT1F313E-18