Smarter. Greener. Together.

Industrial Automation Headquarters

Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan District,
Taoyuan City 33068, Taiwan
TEL: 886-3-362-6301 / FAX: 886-3-371-6301

Asia

Delta Electronics (Shanghai) Co., Ltd.
No.182 Minyu Rd., Pudong Shanghai, P.R.C.
Post code : 201209
TEL: 86-21-6872-3988 / FAX: 86-21-6872-3996
Customer Service: 400-820-9595

Delta Electronics (Japan), Inc.
Tokyo Office
Industrial Automation Sales Department
2-1-14 Shibadaimon, Minato-ku
Tokyo, Japan 105-0012
TEL: 81-3-5733-1155 / FAX: 81-3-5733-1255

Delta Electronics (Korea), Inc.
Seoul Office
1511, 219, Gasan Digital 1-Ro., Geumcheon-gu,
Seoul, 08501 South Korea
TEL: 82-2-515-5305 / FAX: 82-2-515-5302

Delta Energy Systems (Singapore) Pte Ltd.
4 Kaki Bukit Avenue 1, #05-04, Singapore 417939
TEL: 65-6747-5155 / FAX: 65-6744-9228

Delta Electronics (India) Pvt. Ltd.
Plot No.43, Sector 35, HSIIDC Gurgaon,
PIN 122001, Haryana, India
TEL: 91-124-4874900 / FAX : 91-124-4874945

Delta Electronics (Thailand) PCL.
909 Soi 9, Moo 4, Bangpoo Industrial Estate (E.P.Z),
Pattana 1 Rd., T.Phraksa, A.Muang,
Samutprakarn 10280, Thailand
TEL: 66-2709-2800 / FAX : 662-709-2827

Delta Electronics (Australia) Pty Ltd.
Unit 20-21/45 Normanby Rd., Notting Hill Vic 3168, Australia
TEL: 61-3-9543-3720

Americas

Delta Electronics (Americas) Ltd.
Raleigh Office
P.O. Box 12173, 5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3813 / FAX: 1-919-767-3969

Delta Electronics Brazil
São Paulo Sales Office
Rua Itapeva, 26 - 3°, andar Edificio Itapeva,
One - Bela Vista 01332-000 - São Paulo - SP - Brazil
TEL: 55-12-3932-2300 / FAX: 55-12-3932-237

Delta Electronics International Mexico S.A. de C.V.
Mexico Office
Gustavo Baz No. 309 Edificio E PB 103
Colonia La Loma, CP 54060
Tlalnepantla, Estado de México
TEL: 52-55-3603-9200

EMEA

Headquarters: Delta Electronics (Netherlands) B.V.
Sales: Sales.IA.EMEA@deltaww.com
Marketing: Marketing.IA.EMEA@deltaww.com
Technical Support: iatechnicalsupport@deltaww.com
Customer Support: Customer-Support@deltaww.com
Service: Service.IA.emea@deltaww.com
TEL: +31(0)40 800 3900

BENELUX: Delta Electronics (Netherlands) B.V.
De Witbogt 20, 5652 AG Eindhoven, The Netherlands
Mail: Sales.IA.Benelux@deltaww.com
TEL: +31(0)40 800 3900

DACH: Delta Electronics (Netherlands) B.V.
Coesterweg 45, D-59494 Soest, Germany
Mail: Sales.IA.DACH@deltaww.com
TEL: +49(0)2921 987 0

France: Delta Electronics (France) S.A.
ZI du bois Challand 2, 15 rue des Pyrénées,
Lisses, 91090 Evry Cedex, France
Mail: Sales.IA.FR@deltaww.com
TEL: +33(0)1 69 77 82 60

Iberia: Delta Electronics Solutions (Spain) S.L.U
Ctra. De Villaverde a Vallecas, 265 1º Dcha Ed.
Hormigueras – P.I. de Vallecas 28031 Madrid
TEL: +34(0)91 223 74 20

Carrer Llacuna 166, 08018 Barcelona, Spain
Mail: Sales.IA.Iberia@deltaww.com

Italy: Delta Electronics (Italy) S.r.l.
Via Meda 2–22060 Novedrate(CO)
Piazza Grazioli 18 00186 Roma Italy
Mail: Sales.IA.Italy@deltaww.com
TEL: +39 039 8900365

Russia: Delta Energy System LLC
Vereyskaya Plaza II, office 112 Vereyskaya str.
17 121357 Moscow Russia
Mail: Sales.IA.RU@deltaww.com
TEL: +7 495 644 3240

Turkey: Delta Greentech Elektronik San. Ltd. Sti. (Turkey)
Şerifali Mah. Hendem Cad. Kule Sok. No:16-A
34775 Ümraniye – İstanbul
Mail: Sales.IA.Turkey@deltaww.com
TEL: + 90 216 499 9910

GCC: Delta Energy Systems AG (Dubai BR)
P.O. Box 185668, Gate 7, 3rd Floor, Hamarain Centre
Dubai, United Arab Emirates
Mail: Sales.IA.MEA@deltaww.com
TEL: +971(0)4 2690148

Egypt + North Africa: Delta Electronics
Unit 318, 3rd Floor, Trivium Business Complex, North 90 street,
New Cairo, Cairo, Egypt
Mail: Sales.IA.MEA@deltaww.com

AX-0329720-01
2020-10-30

AX Series - Standard Instructions Manual

AX Series - Standard Instructions Manual

www.deltaww.com

Smarter. Greener. Together.

AX Series Standard Instructions Manual

Revision History

Version	Revision	Date
1st	The first version was published.	2020/10/30

AX Series Standard Instructions Manual

Table of Contents

P

Preface

Table of Contents

P.1 Introduction

Thank you for purchasing our product. The AX series motion controller provides a high-level motion control system based on CODESYS to integrate the control function of PLCs and Motion Control.

This manual introduces Delta self-developed function blocks and functions for customers to perform PLC application development.

P.1.1 Applicable Products

This manual applies to the following products:

- AX-3 Series

P.1.2 Associated Manuals

1. **DIADesigner-AX User Manual**
   Includes the information of software operation, programming languages(Ladder Diagram, Sequential function charts, ST（Structured Text） and function blocks), concept of POU and Task, as well as motion control programming.

2. **AX-3 Series Operational Manual**
   Introduces the concept of motion control system, while gives the information of hardware and software configuration, motion control programming framework, troubleshooting, analog input-output module and temperature measurement module.

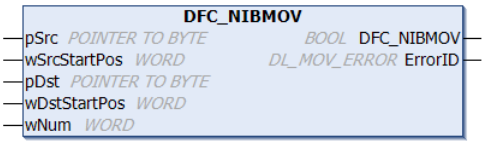
1

Chapter 1 Move Instructions

Table of Contents

1.1 DFC_NIBMOV

DFC_NIBMOV: Data shift.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_NIBMOV		DFC_NIBMOV(pSrc:= , wSrcStartPos:= , pDst:= , wDstStartPos:= , wNum:= , ErrorID=>);

- **Inputs**

Name	Function	Data Type	Setting Value (Default value)
pSrc	Memory address of source variables	POINTER TO BYTE	Memory address (0)
wSrcStartPos	Start address for source variable shift (Unit: Nibble)	WORD*	Positive integer (0)
pDst	Memory address of target variables	POINTER TO BYTE	Memory address (0)
wDstStartPos	Start address for storing target variable (Unit: Nibble)	WORD*	Positive integer (0)
wNum	The data length for data shift (Unit: Nibble)	WORD*	Positive integer (0)

***Note:** The variable types BYTE and WORD can be used for inputs.

- **Outputs**

Name	Function	Data Type	Output Range (Default value)
DFC_NIBMOV	Execution result (Return type)	BOOL	True/False(False)
ErrorID	Error code	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

- **Function**

After executing this Function, the value of variable1 (pSrc) will be copied to variable2 (pDst), while the length of copied data is determined by wNum input. (Unit: Nibble)

- **Example**

  - Program example 1:

  In this example, FC instruction (DFC_NIBMOV) is used for shifting the content of wVar0(pSrc) to the variable wVar1(pDst).

```
PLC_PRG
PROGRAM PLC_PRG
VAR
    bVar0: BOOL;
    Length: WORD:=2;
    wVar0: WORD := 16#1234;
    wVar1: WORD := 16#FFFF;
END_VAR
```

```
IF bVar0 THEN
    DFC_NIBMOV(
        pSrc:= ADR(wVar0),
        wSrcStartPos:= 2,
        pDst:= ADR(wVar1),
        wDstStartPos:= 0,
        wNum:= Length,
        ErrorID=> );
    bVar0:=FALSE;
END_IF;
```

Since wSrcStartPos=2, wNum=2 and wDstStartPos=0, two consecutive Nibbles(Length=2), which start from Nibble2 of variable wVar0(pSrc), are shifted to the address Nibble0 inside the memory of wVar1(pDst).

Variable	Nibble3	Nibble2	Nibble1	Nibble0
wVar0 [16#1234]	Memory content			
	1	2	3	4

wVar1 [16#FFFF]	Memory content			
	F	F	F	F

wVar1 [16#FF12]	Memory content (after executing FC)			
	F	F	1	2

■ Program example 2:

In this example, FC instruction (DFC_NIBMOV) is used for shifting the content of ar_wVar0(pSrc) to the variable ar_wVar1(pDst).

```
POU ×
1  PROGRAM POU
2  VAR
3      bVar0: BOOL;
4      Length: WORD:=2;
5      ar_wVar0: ARRAY [0..1] OF WORD := [16#0123,16#4567];
6      ar_wVar1: ARRAY [0..1] OF WORD := [16#FFFF,16#FFFF];
7  END_VAR

1  IF bVar0 THEN
2      DFC_NIBMOV(
3          pSrc:= ADR(ar_wVar0),
4          wSrcStartPos:= 3,
5          pDst:= ADR(ar_wVar1),
6          wDstStartPos:= 0,
7          wNum:= Length,
8          ErrorID=> );
9      bVar0:=FALSE;
10 END_IF;
11
```

Since wSrcStartPos=3, wNum=2 and wDstStartPos=0, two consecutive Nibbles (Length=2), which start from Nibble3 of variable ar_wVar0(pSrc), are shifted to the address Nibble0 inside the memory of ar_wVar1(pDst).

Variable	Nibble7	Nibble6	Nibble5	Nibble4	Nibble3	Nibble2	Nibble1	Nibble0
ar_wVar0 [16#0123,16#4567]	Memory content							
	4	5	6	7	0	1	2	3

ar_wVar1 [16#FFFF,16#FFFF]	Memory content							
	F	F	F	F	F	F	F	F

ar_wVar1 [16#FF70,16#FFFF]	Memory content (after executing FC)							
	F	F	F	F	F	F	7	0

- **Supported Products**
  - ■ AX series
- **Library**
  - ■ DL_Mov.library

1.2 DFC_XCH

DFC_XCH: Data exchange between two variables.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_XCH		DFC_XCH(pSrc1:= , pSrc2:= , dwNum:= , ErrorID=>);

- **Inputs**

Name	Function	Data Type	Setting Value (Default value)
pSrc1	Memory address of variable1	POINTER TO BYTE	Memory address (0)
pSrc2	Memory address of variable2	POINTER TO BYTE	Memory address (0)
dwNum	The length of data for exchange.(Unit: Byte)	DWORD*	1 ~ 65535 Positive integer(0)

***Note:** The variable types BYTE, WORD and DWORD can be used for dNum input.

- **Outputs**

Name	Function	Data Type	Output Range (Default value)
DFC_XCH	Execution result (Return type)	BOOL	True/False(False)
ErrorID	Error code	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

- **Function**

After executing this Function, the value of variable1 (pSrc1) will be copied to variable2 (pSrc2), while the length of copied data is determined by dwNum input.

- **Example**

In this example, Function (DFC_XCH) is used for exchanging contents of two variables.

```
PLC_PRG
PROGRAM PLC_PRG
VAR
    bVar0,bVar1:BOOL;
    Length:DWORD:=1;
    wVar0:WORD:=16#1234;
    wVar1:WORD:=16#5678;
END_VAR
```

```
IF bVar0 THEN
    bVar1:=DFC_XCH(pSrc1:= ADR(wVar0), pSrc2:= ADR(wVar1) , dwNum:=Length , ErrorID=> )
    bVar0:=FALSE;
END_IF;
```

Since the data length for data exchange is set to one Byte(Length=1), low-byte of variable1 and 2 will be switched after executing Function(DFC_XCH).

Before execution

Variable	wVar0	wVar1
Content	16#1234	16#5678

After executing Function

Variable	wVar0	wVar1
Content	16#1278	16#5634

- **Supported Products**
  - AX series

- **Library**
  - DL_Mov.library

1.3 Error Code and Troubleshooting

Description	Reasons for error	Troubleshooting
DFC_NIBMOV_ERR_PARAMETER	Incorrect value of wNum	Check if the value of wNum is bigger than 0.
DFC_XCH_ERR_PARAMETER	Incorrect value of dwNum	Check if the value of dwNum is bigger than 0.
DFC_XCH_ERR_NOMEMORY	Not enough memory space in controller.	Check if the size of downloaded program exceeds the limit, then reboot the controller.

MEMO

2

Chapter 2 Comparison Instructions

Table of Content

2.1 DFC_CMP

DFC_CMP: Comparison between LINT variables.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_CMP		DFC_CMP(liSrc1:= , liSrc2:=)

- **Inputs**

Name	Function	Data Type	Setting Value (Default value)
liSrc1	Variable1	LINT*	LINT:-2^{63} ~ 2^{63}-1 (0)
liSrc2	Variable2	LINT*	LINT:-2^{63} ~ 2^{63}-1 (0)

***Note:** The variable types SINT、INT、DINT and LINT can be used for inputs.

- **Outputs**

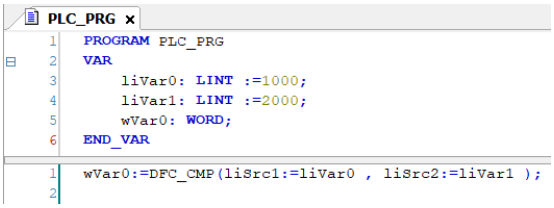
Name	Function	Data Type	Output Range (Default value)
DFC_CMP	Execution result (Return type)	WORD	1:liSrc1 = liSrc2 2:liSrc1 < liSrc2 3:liSrc1 > liSrc2 (0)

- **Function**

The FC instruction is used to compare the values in variable 1 with that in variable 2.

- **Programing Example**

This example use FC instruction (DFC_CMP) to do comparison between two variable values.



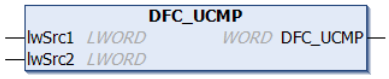
Since variable1 (liVar0) is smaller than variable2 (liVar1), the calculation result (wVar0) would be 2.

- **Supported Products**
  - AX series

- **Library**
  - DL_Comparison.library

2.2 DFC_UCMP

DFC_UCMP: Comparison between ULINT variables.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_UCMP		DFC_UCMP(lwSrc1:= , lwSrc2:=)

- **Input**

Name	Function	Data Type	Setting Value (Default Value)
lwSrc1	Variable 1	ULINT/LWORD*	ULINT/LWORD:0 ~ 2^{64}-1 (0)
lwSrc2	Variable 2	ULINT/LWORD*	ULINT/LWORD:0 ~ 2^{64}-1 (0)

***Note:** The variable types USINT、UINT、UDINT、ULINT、BYTE、WORD、DWORD and LWORD can be used for inputs.

- **Output**

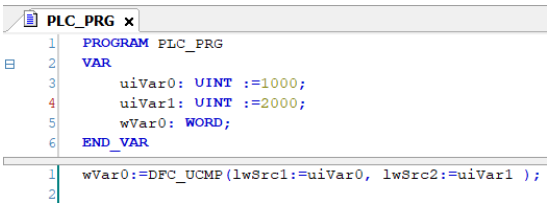
Name	Function	Data Type	Output Range (Default Value)
DFC_UCMP	Execution result (Return type)	WORD	1:lwSrc1 = lwSrc2 2:lwSrc1 < lwSrc2 3:lwSrc1 > lwSrc2 (0)

- **Function**

The FC instruction is used to compare the values in variable 1(lwSrc1) with that in variable 2(lwSrc2).

- **Programming Example**

This example use FC instruction (DFC_UCMP) to do comparison between two variable values.



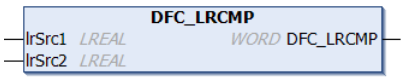
Since variable1 (uiVar0) is smaller than variable2 (uiVar1), the calculation result (wVar0) would be 2.

- **Supported Products**
  - AX Series

- **Library**
  - DL_Comparison.library

2.3 DFC_LRCMP

DFC_LRCMP: Comparison between LREAL variables.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_LRCMP		DFC_LRCMP(lrSrc1:= , lrSrc2:=)

- **Input**

Name	Function	Data Type	Setting Value (Default Value)
lrSrc1	Variable 1	LREAL*	LREAL:-1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)
lrSrc2	Variable 2	LREAL*	LREAL:-1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)

***Note:** The variable types REAL and LREAL can be used for inputs.

- **Output**

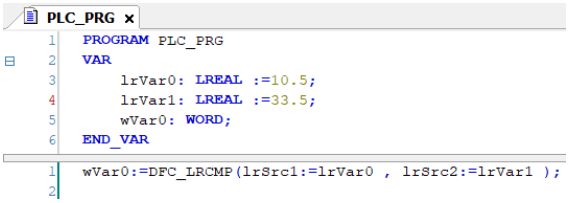
Name	Function	Data Type	Output Range (Default Value)
DFC_LRCMP	Execution result (Return type)	WORD	1:lrSrc1 = lrSrc2 2:lrSrc1 < lrSrc2 3:lrSrc1 > lrSrc2 (0)

- **Function**

The FC instruction is used to compare the values in variable 1(lrSrc1) with that in variable 2(lrSrc2).

- **Programming Example**

This example use FC instruction (DFC_LRCMP) to do comparison between two variable values.



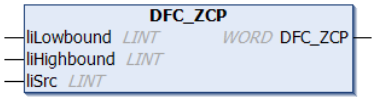
Since variable1 (lrVar0) is smaller than variable2 (lrVar1), the calculation result (wVar0) would be 2.

- **Supported Products**
  - AX Series

- **Library**
  - DL_Comparison.library

2.4 DFC_ZCP

DFC_ZCP: Compares a range with a value of LINT variable.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_ZCP		DFC_ZCP(liLowbound:= , liHighbound:= , liSrc:=);

- **Input**

Name	Function	Data Type	Setting Value (Default Value)
liLowbound	Lower value	LINT*	LINT:-2^{63} ~ 2^{63}-1 (0)
liHighbound	Upper value	LINT*	LINT:-2^{63} ~ 2^{63}-1 (0)
liSrc	Variable	LINT*	LINT:-2^{63} ~ 2^{63}-1 (0)

***Note:** The variable types SINT、INT、DINT and LINT can be used for inputs.

- **Output**

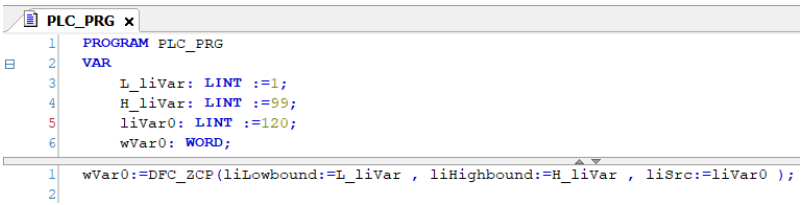
Name	Function	Data Type	Output Range (Default Value)
DFC_ZCP	Execution result (Return type)	WORD	1:liSrc < Lower value 2: Lower value < liSrc < Upper value 3:liSrc > Upper value (0)

- **Function**

The FC instruction is used to compare the values in variable (liSrc) with the upper and lower value of the range.

- **Programming Example**

This example use FC instruction (DFC_ ZCP) to compare variable values with the upper and lower value.



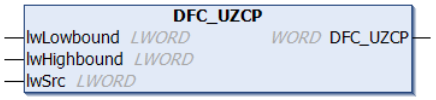
Since the value in variable (liVar0) is larger than the upper value (H_liVar), the calculation result (wVar0) is 3.

- **Supported Products**
  - AX Series

- **Library**
  - DL_Comparison.library

2.5 DFC_UZCP

DFC_UZCP: Compares a range with a value of ULINT variable.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_UZCP		DFC_UZCP(lwLowbound:= , lwHighbound:= , lwSrc:=);

- **Input**

Name	Function	Data Type	Setting Value (Default Value)
lwLowbound	Lower value	ULINT/LWORD*	ULINT/LWORD:0 ~ 2^{64}-1 (0)
lwHighbound	Upper value	ULINT/LWORD*	ULINT/LWORD:0 ~ 2^{64}-1 (0)
lwSrc	Variable	ULINT/LWORD*	ULINT/LWORD:0 ~ 2^{64}-1 (0)

***Note:** The variable types SINT、INT、DINT, BYTE, WORD, DWORD and LWORDcan be used for inputs.

- **Output**

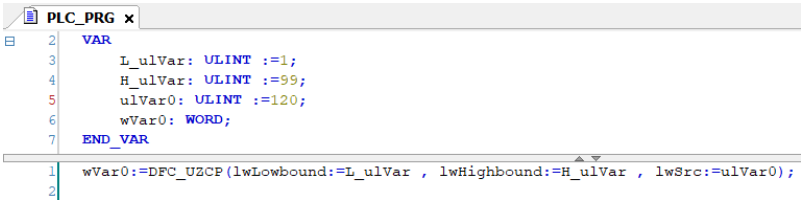
Name	Function	Data Type	Output Range (Default Value)
DFC_UZCP	Execution result (Return type)	WORD	1:lwSrc < Lower value 2: Lower value < lwSrc < Upper value 3:lwSrc > Upper value (0)

- **Function**

The FC instruction is used to compare the values in variable (lwSrc) with the upper and lower value of the range.

- **Programming Example**

This example use FC instruction (DFC_ UZCP) to compare variable values with the upper and lower value.



Since the value in variable (ulVar0) is larger than the upper value (H_ulVar), the calculation result (wVar0) is 3.

- **Supported Products**
  - AX Series

- **Library**
  - DL_Comparison.library

2.6 DFC_LRZCP

DFC_LRZCP: Compares a range with a value of LREAL variable.

FB/FC	Instruction	Graphic Expression	ST language
FC	DFC_LRZCP		DFC_LRZCP(lrLowbound:= , lrHighbound:= , lrSrc:=);

- **Input**

Name	Function	Data Type	Setting Value (Default Value)
lrLowbound	Lower value	LREAL*	LREAL:-1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)
lrHighbound	Upper value	LREAL*	LREAL:-1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)
lrSrc	Variable	LREAL*	LREAL:-1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)

***Note:** The variable types REAL and LREAL can be used for inputs.

- **Output**

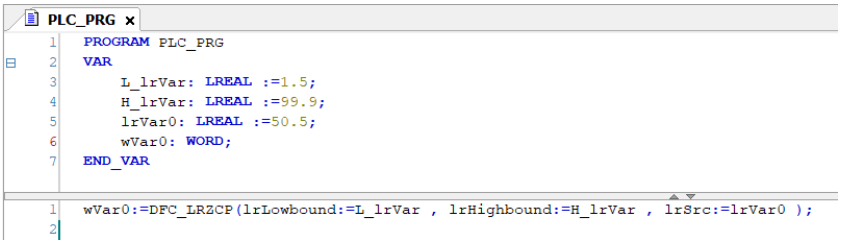
Name	Function	Data Type	Output Range (Default Value)
DFC_LRZCP	Execution result (Return type)	WORD	1:lrSrc < Lower value 2: Lower value < lrSrc < Upper value 3:lrSrc > Upper value (0)

- **Function**

The FC instruction is used to compare the values in variable (lrSrc) with the upper and lower value of the range.

- **Programming Example**

This example use FC instruction (DFC_LRZCP) to compare variable values with the upper and lower value.



Since the value in variable (lrVar0) is smaller than the upper value (H_lrVar) and larger than the lower value (L_lrVar), the calculation result (wVar0) is 2.

- **Supported Products**
  - AX Series

- **Library**
  - DL_Comparison.library

MEMO

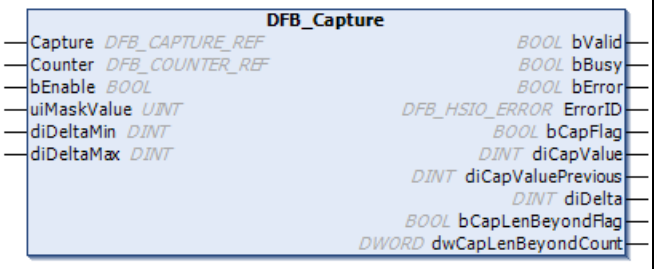
3

Chapter 3 Timers and Counters Instructions

Table of Content

3.1 DFB_Capture

DFB_Capture captures the commanded pulses of the specified high-speed counter according to the designated external trigger device.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_Capture		DFB_Capture_instance(Capture :=, Counter :=, bEnable :=, uiMaskValue :=, diDeltaMin :=, diDeltaMax :=, bValid =>, bBusy =>, bError =>, ErrorID =>, bCapFlag =>, diCapValue =>, diCapValuePrevious =>, diDelta =>, bCapLenBeyondFlag =>, dwCapLenBeyondCount =>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Counter	Designate the source of the specified high-speed counter.	DFB_COUNTER_REF[*1]	DFB_COUNTER_REF (Cannot be null.)	-
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-
uiMaskValue	Define the mask range of Capture.	UINT	Positive number or 0(0)	When bEnable shifts to True, the setting parameters of uiMaskValue will be updated.
diDeltaMin	Define the minimum difference between each Capture[*2].	DINT	Positive number, negative number or 0(0)	When bEnable shifts to True and Busy is False
diDeltaMax	Define the maximum difference between each Capture[*2].	DINT	Positive number, negative number or 0(0)	When bEnable shifts to True and Busy is False

***Note:**

1. DFB_Counter_REF(FB): As the I/O interface of the high-speed counter to perform actions include parameter adjustment and the driver.
2. Once diDeltaMin and diDeltaMax are set to 0, the system will not check whether the capture range is appropriate or not.

- **Output**

Name	Function	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
bCapFlag	Indicates that the current Capture is valid. (The flag shifts to True for one scan cycle and will be reset immediately)	BOOL	True/False(False)
diCapValue	The captured value.	DINT	Positive number, negative number or 0(0)
diCapValuePrevious	The previous captured value.	DINT	Positive number, negative number or 0(0)
diDelta	The difference between the previous and the current captured values.	DINT	Positive number, negative number or 0(0)
bCapLenBeyondFlag	Indicates that a capture is failed. (The flag shifts to True for one scan cycle and will be reset immediately)	BOOL	True/False(False)
dwCapLenBeyondCount	Counts the number of the failed Capture.	DWORD	Positive number or 0(0)

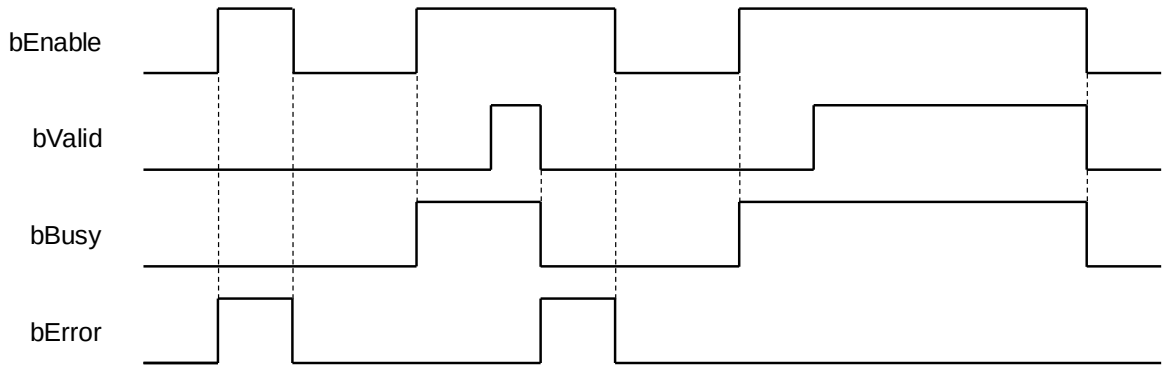
***Note:** DFB_HSIO_ERROR: Enumeration (Enum)

■ **Outputs Update Timing**

Name	Timing for shifting to True	Timing for shifting to False
bValid	• When the values at the outputs are valid after bEnable being True for one scan cycle.	• When bEnable shifts to False. • When bError shifts to False.
bBusy	• When bEnable is rising edge triggered.	• When bEnable shifts to False. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When bEnable shifts from True to False. (Error code is cleared)
ErrorID		
bCapFlag	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.
diCapValue	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.
diCapValuePrevious	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.
diDelta	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.
bCapLenBeyondFlag	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.
dwCapLenBeyondCount	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.

3

- **Timing Diagram**



- **In-Outs**

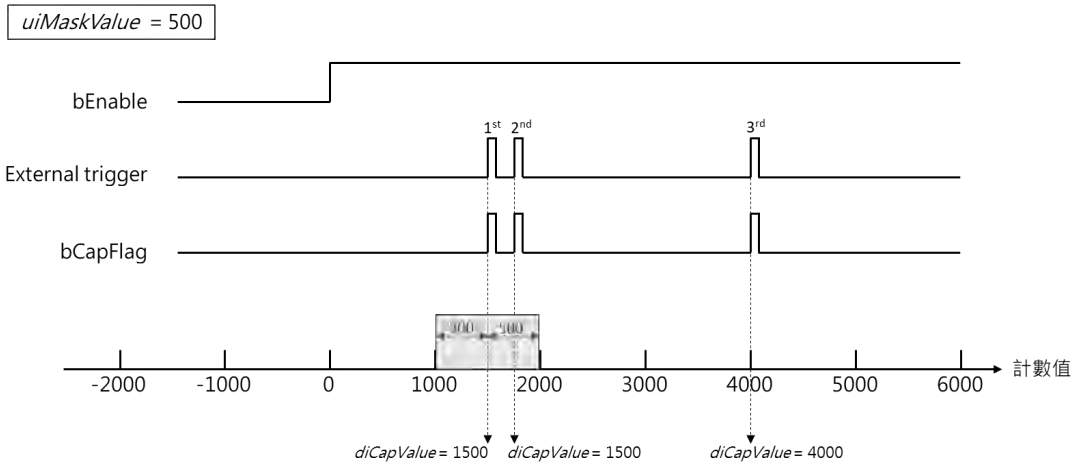
Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Capture	Refer to the source of the specified high-speed capture	DFB_CAPTURE_REF(FB)*	DFB_CAPTURE_REF(Cannot be null.)	When bEnable shifts to True and bBusy is False.

***Note:** DFB_CAPTURE_REF(FB): The I/O function block of the high-speed counter which contains parameter adjustment and the driver.

- **Function**
  - **uiMaskValue**

    Please refer to the following figure for the function description of uiMaskValue input.

    1. Set uiMaskValue to 500 and bEnable to True, then Capture function is enabled. At the same time, the output bValid is True and the first captured value would be the center value of the mask range. In addition, the next capture action will be invalid if the next captured value is within the mask range.
    2. In the figure below, the 1st capture happens in a -500~500 range and the captured value changes from 0 to 1500.
    3. The captured value 1500 becomes the new center of mask range. Therefore, the next captured value which locates between 1000 to 2000 (1000 < diCapValue < 2000) will be invalid. So when the 2nd capture is triggered (in the mask range), the captured value would remain as 1500.
    4. Since the 3rd capture is triggered outside of the mask range, the captured value would be updated to 4000.

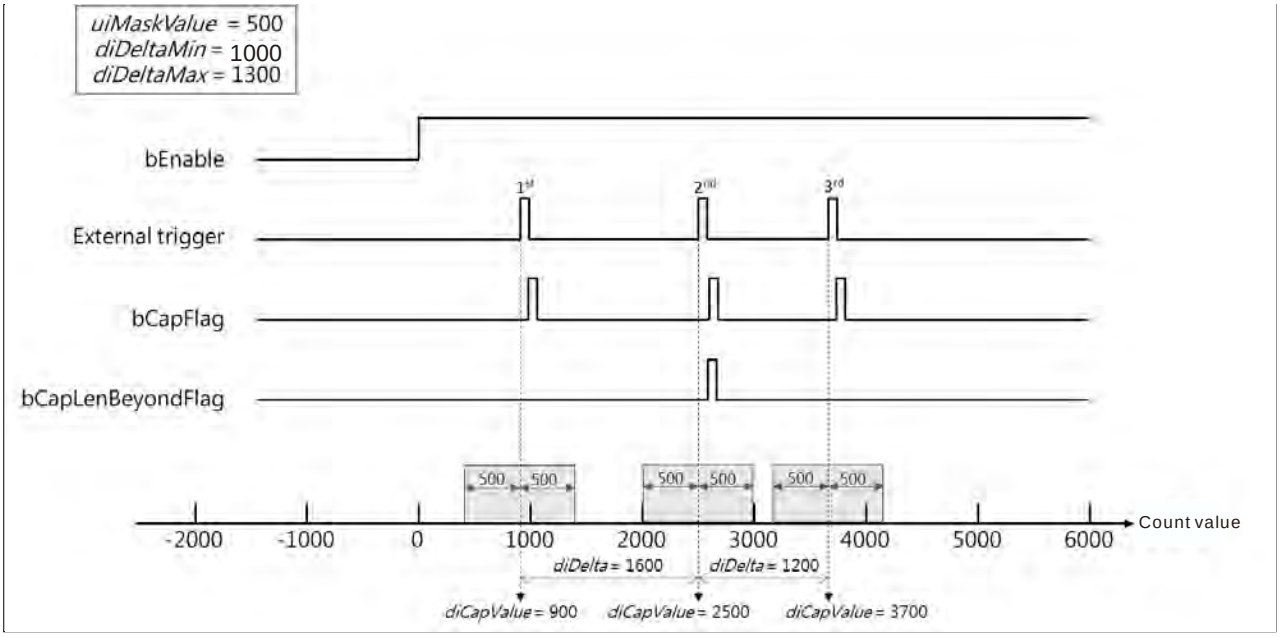


  - **diDeltaMin · diDeltaMax · bCapLenBeyondFlag · dwCapLenBeyondCount**

    DeltaMin/DeltaMax define the minimum and maximum distance between each Capture, while CapLenBeyondFlag and CapLenBeyondCount represent the error flag and the number of the failed Capture.

    1. The function of diDeltaMin/diDeltaMax is to judge if a trigger mark is missed and the Capture is not executed. For example, if the value of DeltaMin is 1000 and DeltaMax is 1300, when the detected distance between 2 Capture exceeds 1000~1300, the system will flag this situation as trigger mark missing.
    2. When a mark missing condition occurs, CapLenBeyondFlag shifts to Ture for one scan cycle and will be reset immediately. At the same time dwCapLenBeyondCount counts 1.
    3. Refer to the below diagram for the explanation of these inputs and outputs:
       - The mask range is between -500~500 and the 1st Capture occurs at 900.
       - The 2nd Capture occurs at 2500. Because DeltaMax is set to 1300 and DeltaMin is set to 1000 (1000-1300), the detected distance between two captures has exceeded the range of 1000~1300. Therefore, a trigger mark missing condition is flagged for a scan cycle, while bCapLenBeyondFlag remains as TRUE.

➢ The 3rd Capture occurs at 3700. Because the difference between 3700 and the previous captured value 2500 is 1200, which is within the range of 1000~1300 (DeltaMin/DeltaMax), also 3700 is out of the mask range 2000~3000, the captured value changes to 3700 in this case, and bCapLenBeyondFlag will not change to True.



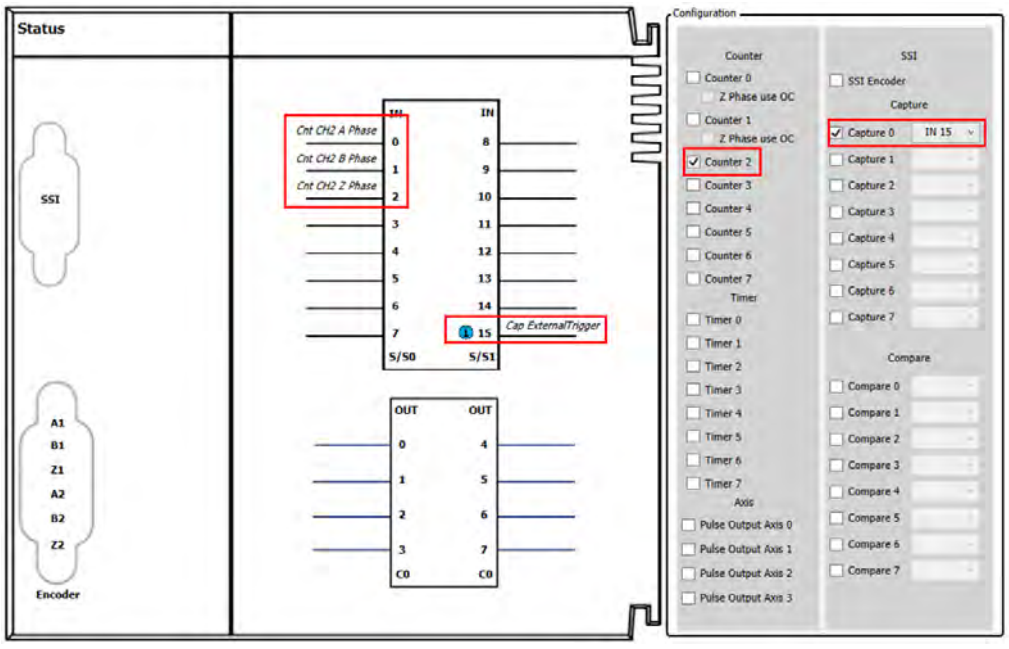
- **Troubleshooting**

If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

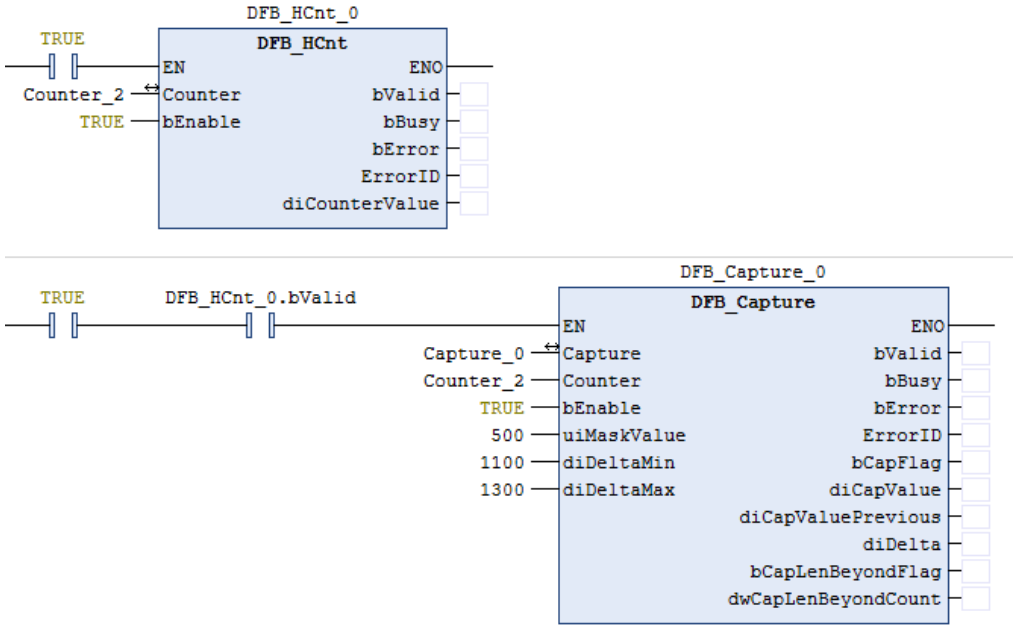
- **Programming Example**

This example uses DFB_HCnt and DFB_Capture to perform the Capture function.

1. As the following figure shows, select a Counter and a Capture for Hardware IO Configuration in BuiltIn_IO and set the trigger source of Capture to a signal input on the hardware (e.g. IN15).



2. Enabe the FB DFB_Capture (bEnable = True) after using the FB DFB_HCnt to activate the high-speed counter (bEnable = True) in the POU, then the present counter value would be captured and shown on the diCapValue output of DFB_Capture after the external signal (IN15) being triggered.



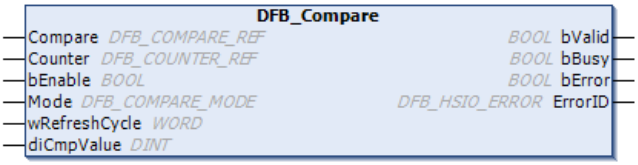
3. Please refer to AX-3 series operational manual for more details related to the settings and operation of Hardware IO Configuration.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_BuiltInIO.library

3.2 DFB_Compare

DFB_Compare compares the designated source value and the setting value and then to Set or Reset the desired device when the comparison result is True or False.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_Compare		DFB_Compare_instance(Compare :=, Counter :=, bEnable :=, Mode :=, wRefreshCycle :=, diCmpValue :=, bValid =>, bBusy =>, bError =>, ErrorID =>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Counter	Designates the source of high-speed counter.	DFB_ COUNTER_ REF*1	DFB_COUNTER_REF (Cannot be null.)	-
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-
Mode	Comparison condition	DFB_ COMPARE_ MODE*2	0:Equal(=) 1:Bigger_Equal(≧) 2:Smaller_Equal(≦) (Equal)	When bEnable shifts toTrue and bBusy is False.
wRefreshCycle	Define the cycle time to refresh the status of the output device.	WORD	Positive number or 0(0)	When bEnable shifts toTrue and bBusy is False.
diCmpValue	Specifies the comparison value	DINT	Positive number, negative number or 0(0)	When bEnable shifts toTrue and bBusy is False.

***Note:**

1. DFB_Counter_REF(FB): As the I/O interface of the high-speed counter to perform actions include parameter adjustment and the driver.
2. DFB_COMPARE_MODE: Enumeration (Enum)

- **Output**

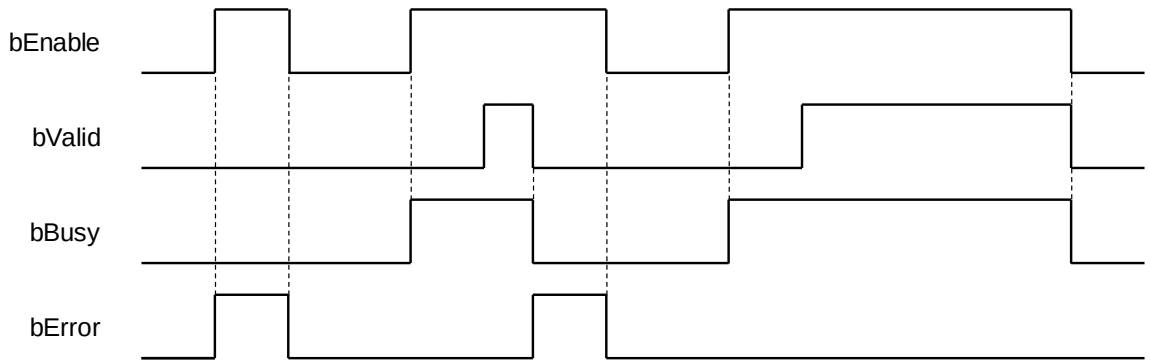
Name	Function	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*	DFB _HSIO_ERROR (DFB _HSIO_NO_ERR)

***Note:** DFB _HSIO_ERROR: Enumeration (Enum)

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bValid	• When the values at the outputs are valid after bEnable being True for one scan cycle.	• When bEnable shifts to False. • When bError shifts to True.
bBusy	• When bEnable is rising edge triggered.	• When bEnable shifts to False. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When bEnable shifts from True to False. (Error code is cleared)
ErrorID		

- **Timing Diagram**



- **In/ Outs**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Compare	Reference to the source of high-speed comparator.	DFB_COMPARE_REF(FB)*	DFB_COMPARE_REF (Cannot be null.)	When bEnable shifts to True and bBusy is False.

***Note:** DFB_COMPARE_REF(FB): As the I/O interface of the high-speed comparator to perform actions include parameter adjustment and the driver.

- **Function**

1. When the comparison result is True (Counter Value = diCmpValue), DFB_Compare will outputs the results according to the settings of HW IO configuration in BuiltIn IO.
2. When bValid output of DFB_Compare is True, the comparator would continue to compare on the high-speed count values. In case that the comparison condition is fulfilled and the output result is given according to the settings, the device would remain at a high-level signal and would not retrigger the output (True → False → True) after the condition is fulfilled once again. If you need to reset the output device and change the high-level signal to low, please find the following methods.
   - ◆ Define the variable at the output of Compare via I/O mapping in DIO, then set the output variable to falling-edge in the POU programming area so as to reset the output device.
   - ◆ Use the setting of wRefreshCycle to change the high-level signal to low automatically after the PLC keeps it at a high-level signal for a period of time.

   Either ways, the purpose of changing from high-level signals to low can be reached. However, the comparison conditions must be fulfilled again if you intend to make the output back to high-level.
3. The output device status can be refreshed by using the input wRefreshCycle. For example, set the value of wRefreshCycle to 10000(Unit: 0.1ms), then the designated output device will be pulled to a low level by the controller after the condition is fulfilled and remains a high-level output for one second. If wRefreshCycle is set to zero, the output device would keep at a high level without being reset.

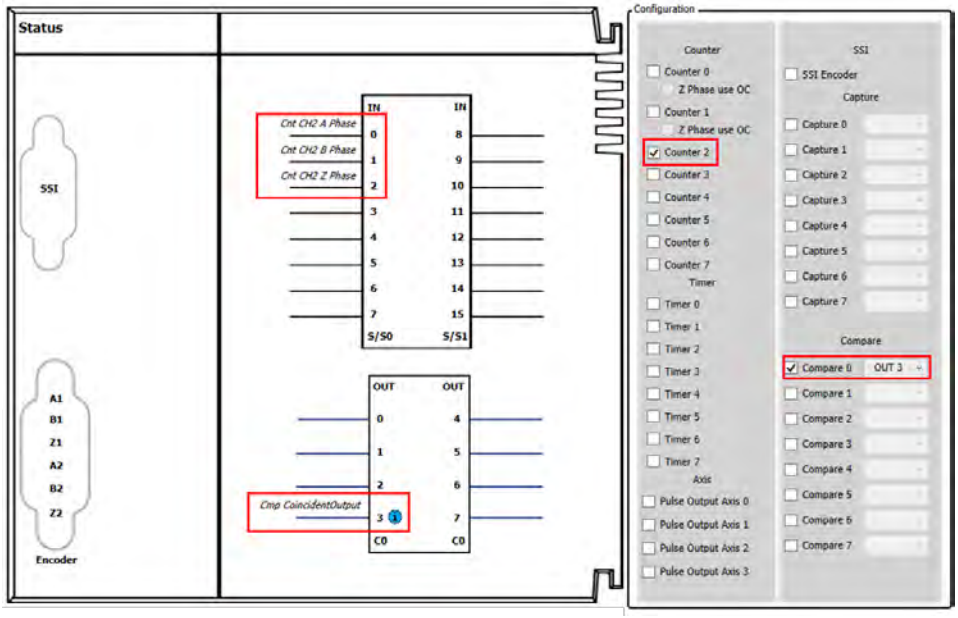
- **Troubleshooting**

If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

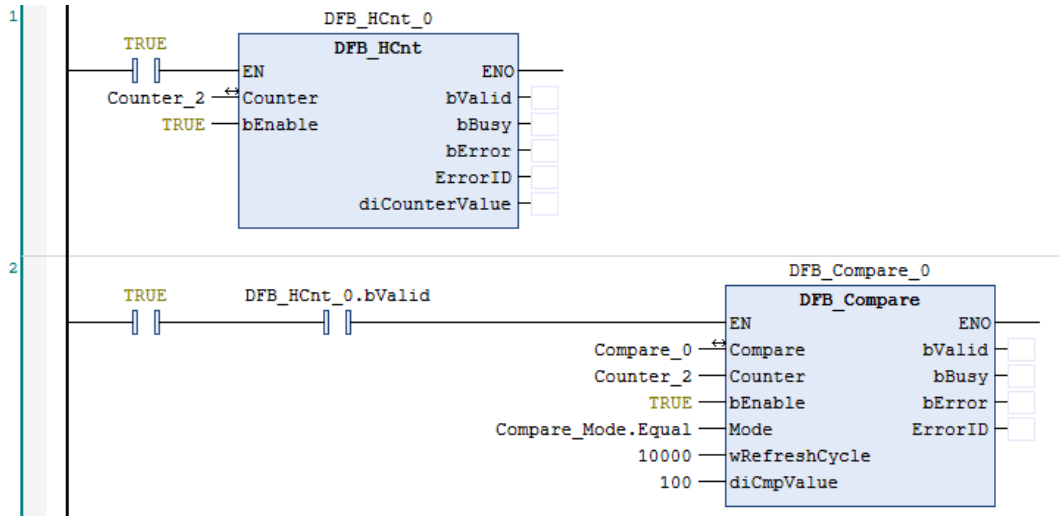
- **Programming Example**

This example uses DFB_HCnt and DFB_Compare to perform the Compare function.

1. As the following figure shows, select a Counter and a Compare for Hardware IO Configuration in BuiltIn_IO and set a signal output on the hardware as the output device of Compare (e.g. OUT3).



2. Execute the function block DFB_Compare after enable the high-speed counter by using DFB_HCnt in the POU as shown in follows. At the same time, the output device(OUT3) will output the signal once the comparison condition is fulfilled(DFB_HCnt_0.diCounterValue = DFB_Compare_0.diCmpValue).



3. Please refer to AX-3 series operational manual for more details related to the settings and operation of Hardware IO Configuration.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_BuiltInIO.library

3.3 DFB_HCnt

DFB_HCnt enables the specified high speed counter according to the specified parameters and monitors the count value.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_HCnt		DFB_HCnt_instance(Counter :=, bEnable :=, bValid =>, bBusy =>, bError =>, ErrorID =>, diCounterValue =>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-

- **Output**

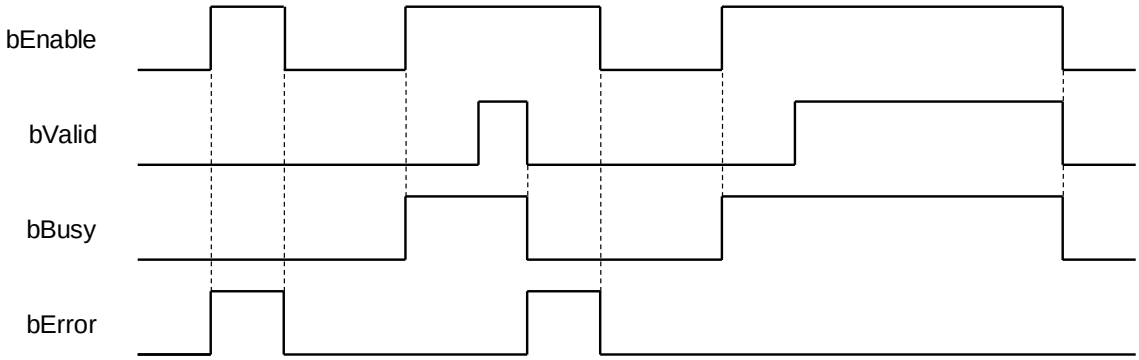
Name	Function	Data Type	Output Range (Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
diCounterValue	The present count value of the counter	DINT	Positive number, negative number or 0(0)

***Note:** DFB_HSIO_ERROR: Enumeration (Enum)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bValid	• When the values at the outputs are valid after bEnable being True for one scan cycle.	• When bEnable shifts to False. • When bError shifts to True.
bBusy	• When bEnable is rising edge triggered.	• When bEnable shifts to False. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When bEnable shifts from True to False. (Error code is cleared)
ErrorID		
diCounterValue	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.

- **Timing Diagram**



- **In/ Outs**

Name	Functoin	Data Type	Setting Value (Default value)	Timing for Updating
Counter	Reference to the source of specified high-speed counter.	DFB_COUNTER_REF(FB)*	DFB_COUNTER_REF (Cannot be null.)	When bEnable shifts to True and Busy is False.

***Note:** DFB_Counter_REF(FB): As the I/O interface of the high-speed counter to perform actions include parameter adjustment and the driver.

- **Function**

1. When the input bEnable is True, the counter would start calculating pulses to the corresponding input points based on the Counter configuration of HW IO configuration in BuiltIn IO.
2. The count value is given through the output diCounterValue during the counting process.

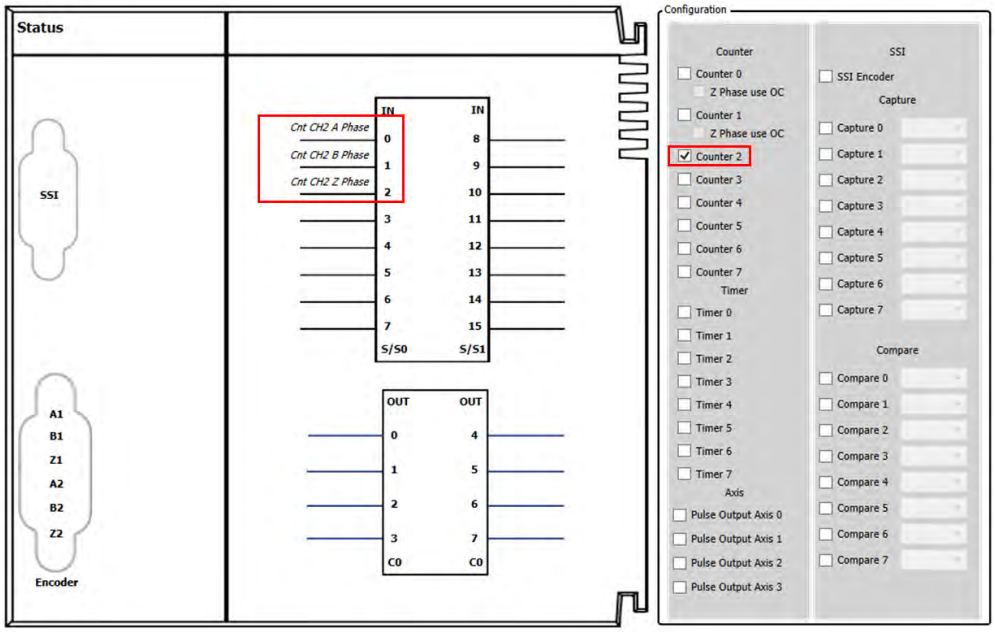
- **Troubleshooting**

If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

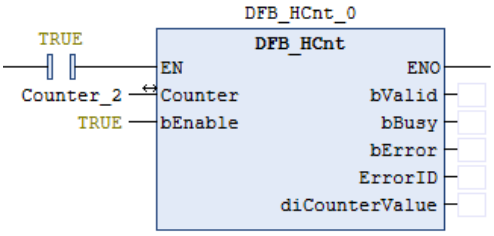
- **Programming Example**

This example uses DFB_HCnt to perform the Count function

1. As the following figure shows, select a Counter (Counter 2) in Hardware IO Configuration and you will see the input points (e.g. IN 0、IN 1、IN 2) matched to the corresponding encoder A, B, Z phase outputs, which the wiring should follows the configuration so as to perform the normal function of high speed counting.



2. After using the FB DFB_HCnt in the POU to activate the high-speed counter(bEnable = True), it starts receiving and counting the pulses from the external signals(IN 0, IN 1) based on the counting mode set in Counter Configuration, then the count value would be displayed in the output diCounterValue. In addition, you should make sure that the mode of sending pulses from the external signal source matches the counting mode so as to get the correct count values.



3. Please refer to AX-3 series operational manual for more details related to the settings and operation of Hardware IO Configuration and Counter Configuration.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_BuiltInIO.library

3.4 DFB_HTmr

DFB_HTmr enables the specified high speed timer channel according to the specified parameters and monitors and timed value.

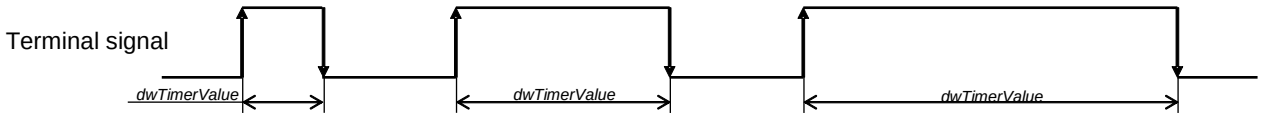
FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_HTmr		DFB_HTmr_instance(Timer :=, bEnable :=, TriggerMode :=, bValid =>, bBusy =>, bError =>, ErrorID =>, dwTimerValue =>);

- **Input**

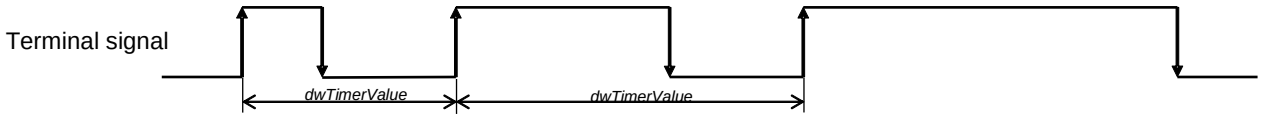
Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-
TriggerMode	Timing mode settings.	DFB_TIMER_MODE *	0:UP_DOWN 1:UP_UP (UP_DOWN)	When bEnable shifts to True and bBusy is False.

***Note:** DFB_TIMER_MODE: Enumeration (Enum)

Up-Down mode:



Up-Up mode:



- **Output**

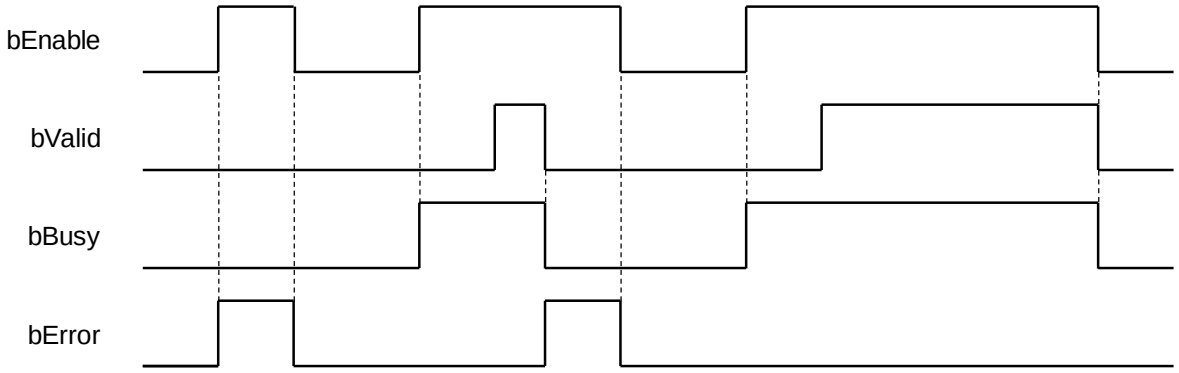
Name	Functon	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
dwTimerValue	Timed value (Unit: 0.1us)	DWORD	Positive number or 0(0)

***Note:** DFB_HSIO_ERROR: Enumeration (Enum)

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bValid	• When the values at the outputs are valid after bEnable being True for one scan cycle.	• When bEnable shifts to False. • When bError shifts to True.
bBusy	• When bEnable is rising edge triggered.	• When bEnable shifts to False. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When bEnable shifts from True to False. (Error code is cleared)
ErrorID		
dwTimerValue	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.

- **Timing Diagram**



- **In/ Outs**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Timer	Reference to the source of the specified high-speed timer.	DFB_TIMER_REF(FB)*	DFB_TIMER_REF(Can not be null.)	When bEnable shifts to True and Busy is False

***Note:** DFB_TIMER_REF(FB): As the I/O interface of the high-speed timer to perform actions include parameter adjustment and the driver.

- **Functon**

1. When the input bEnable is True, the timer would start calculating pulses to the corresponding input points based on the Timer configuration of HW IO configuration in BuiltIn IO.
2. The count value is given through the output dwTimerValue during the counting process.

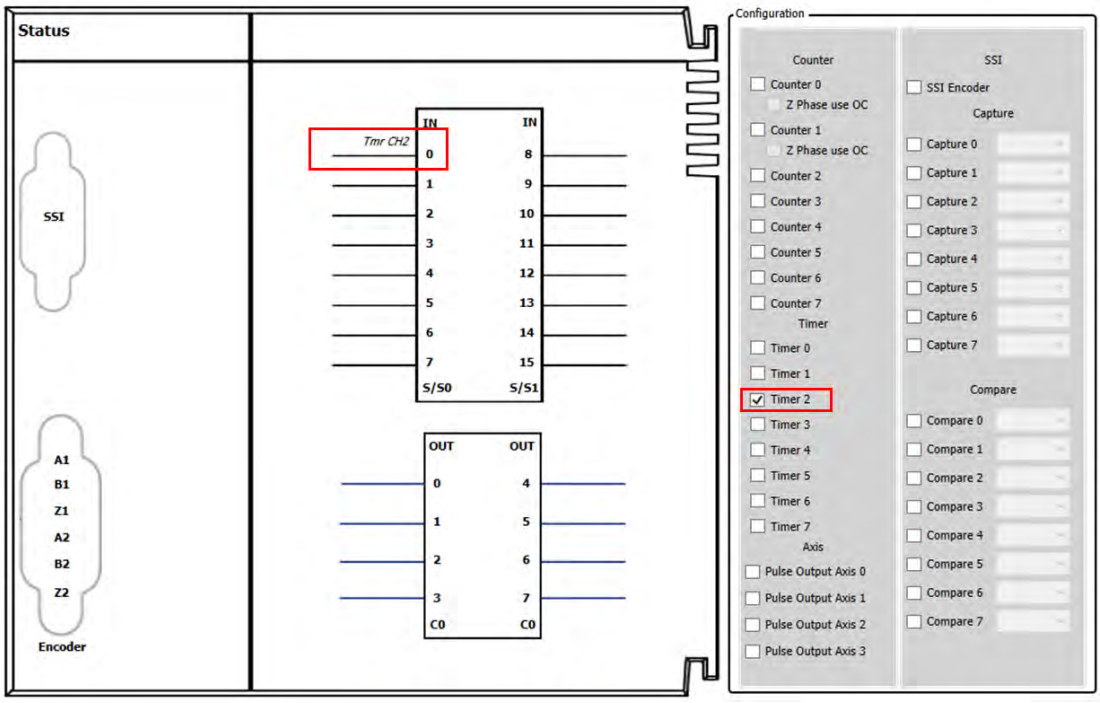
- **Troubleshooting**

If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

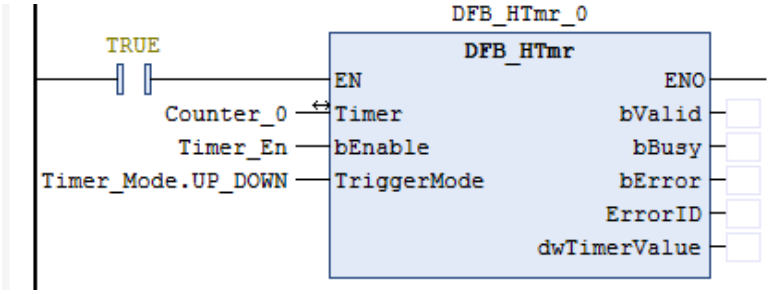
- **Programming Example**

This example demonstrates the function performed by DFB_HTmr.

1. As the following figure shows, select a Timer (Timer 2) in Hardware IO Configuration and you will see the input point (IN 0) matched to the corresponding timer input channel, which the wiring should follows the configuration so as to perform the normal function of high speed timing.



2. After using the FB DFB_HTmr in the POU to activate the high-speed timer(bEnable = True), it starts receiving and counting the pulses from the external signals(IN 0) based on the timing mode set in Timer Configuration, then the timed value would be displayed in the output dwTimerValue.



3. Please refer to AX-3 series operational manual for more details related to the settings and operation of Hardware IO Configuration.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_BuiltInIO.library

3.5 DFB_PresetValue

DFB_PresetValue is the application function block for high-speed counters, its role is to reset the current count value back to the default value.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_PresetValue		DFB_PresetValue_instance(Counter :=, bExecute :=, TriggerType :=, diPresetValue :=, bDone =>, bBusy =>, bCommandAborted =>, bError =>, ErrorID =>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bExecute	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-
TriggerType	Define when the default value would be preset.	DFB_PRESET_TRIGGER_TYPE *	0:EXECUTE_TRIGGER 1:EXTERNAL_TRIGGER (EXECUTE_TRIGGER)	When bExecute shifts to True and bBusy isFalse.
diPresetValue	The preset count value for high speed counters.	DINT	Positive number, negative number or 0(0)	When bExecute shifts to True and bBusy isFalse

***Note:** DFB_PRESET_TRIGGER_TYPE: Enumeration (Enum)

- EXECUTE_TRIGGER: Set the default value right after the input bExecute shifts to True.
- EXTERNAL_TRIGGER: Set the default value right after the external signal of high-speed counter being triggered.

- **Output**

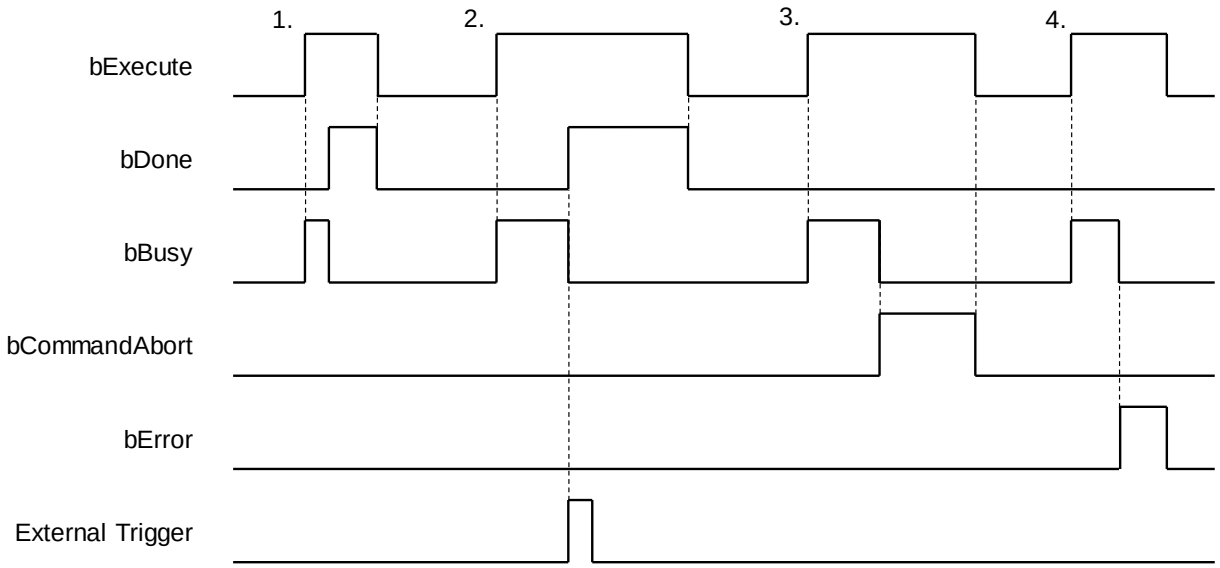
Name	Function	Data Type	Output Range(Default value)
bDone	The default value of the counter has been changed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bCommandAborted	True when the instruction is aborted before it's completed.	BOOL	True/False(False)
bError	True when an error occurs.	BOOL	True/False(False)
ErrorID	Error codes.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)

***Note:** DFB_HSIO_ERROR: Enumeration (Enum)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	• True when the count value has been set back to default.	• When bExecute shifts to False. • When bError shifts to True.
bBusy	• When bExecute is rising edge triggered.	• When bExecute shifts to False. • When bError shifts to True.
bCommandAborted	• True when the FB is aborted.	• When bExecute shifts to False.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When bEnable shifts from True to False. (Error code is cleared)
ErrorID		

- **Timing Diagram**



1. TriggerType = 0(EXECUTE_TRIGGER)
2. TriggerType = 1(EXTERNAL_TRIGGER)
3. bCommandAborted = TRUE
4. bError = TRUE

- **In/ Outs**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Counter	Reference to the source of high-speed counter.	DFB_COUNTER_REF(FB)*	DFB_COUNTER_REF(Cannot be null.)	When bExecute shifts to True and bBusy is False.

***Note:** DFB_COUNTER_REF(FB): As the I/O interface of the high-speed counter to perform actions include parameter adjustment and the driver

- **Function**

1. When TriggerType = EXECUTE_TRIGGER, the count value would be set back to the default value right after activating the function block.
2. Whern TriggerType = EXTERNAL_TRIGGER, the count value would not be set back to the default until the Z phase signal of the counter rises.

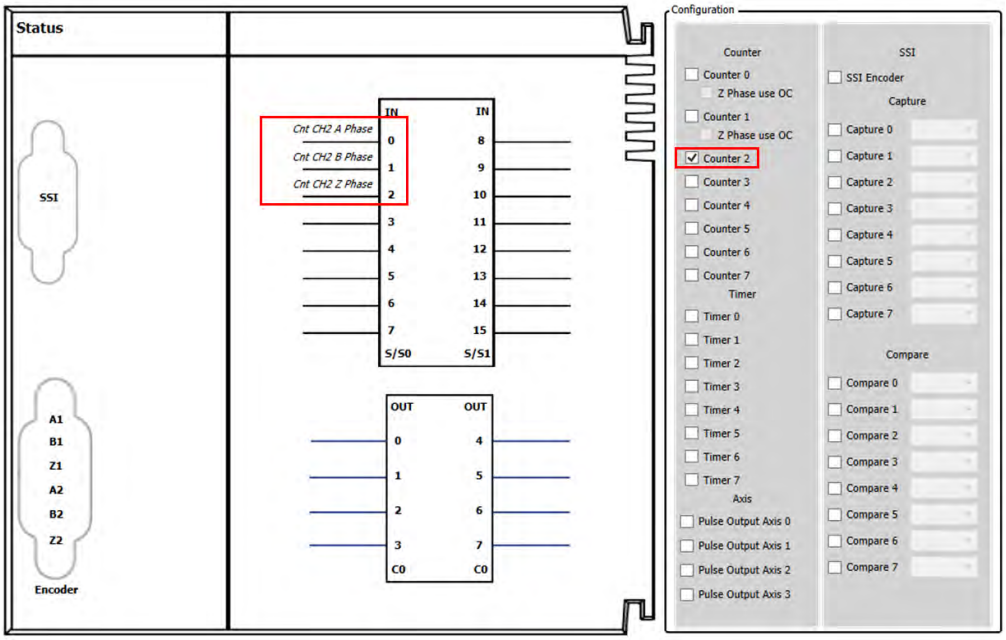
- **Troubleshooting**

If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

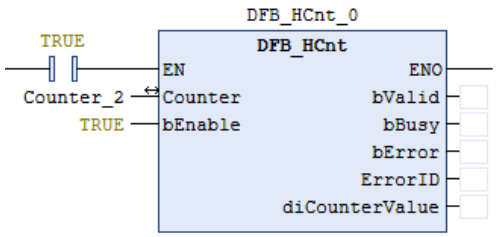
- **Programming Example**

This example demonstrates the function performed by DFB_HCnt and DFB_PresetValue.

1. As the following figure shows, select a Counter (Counter 2) in Hardware IO Configuration and you will see the input points (e.g. IN 0、IN 1、IN 2) matched to the corresponding encoder A, B, Z phase outputs, which the wiring should follows the configuration so as to perform the normal function of high-speed counting.



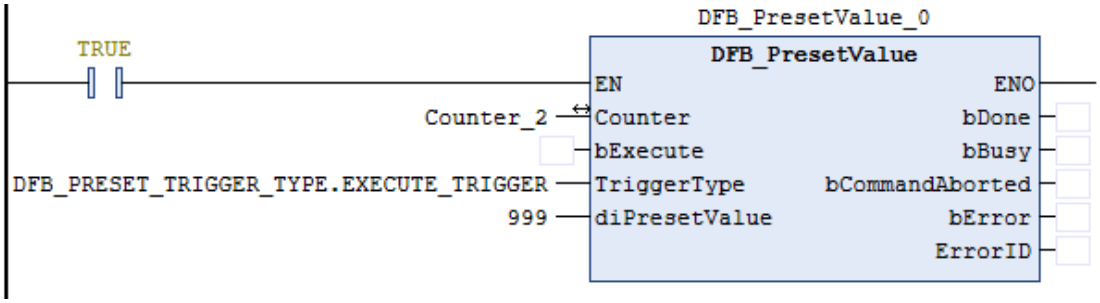
2. After using the FB DFB_HCnt in the POU to activate the high-speed counter(bEnable = True), it starts receiving and counting the pulses from the external signals(IN 0, IN 1) based on the counting mode set in Counter Configuration, then the count value would be displayed in the output diCounterValue. In addition, you should make sure that the mode of sending pulses from the external signal source matches the counting mode so as to get the correct count values.



3

3. If you want to use external signal as the trigger, check the box of External trigger in Counter Configuration as the following figure shows.

4. Then the input bExecute of DFB_PresetValue shifts to True and the FB DFB_PresetValue would wait for the Z phase of high-speed counter to trigger the Default value function. After the count value being set to the default (DFB_HCnt.diCounterValue = DFB_PresetValue.diPresetValue), the output bDone will shift from False to True.



5. Please refer to AX-3 series operational manual for more details related to the settings and operation of Counter Configuration.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_BuiltInIO.library

3.6 DFB_Sample

DFB_Sample is the application function block for high-speed counters, its role is to read the increasing and decreasing number of the count value during the sampling period.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_Sample		DFB_Sample_instance(Counter :=, bEnable :=, wSampleTime :=, bValid =>, bBusy =>, bError =>, ErrorID => diSampleValue =>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-
wSampleTime	Sampling period (Unit: 1ms)	WORD	10 ~ 65535 (0)	When bEnable shifts to True and bBusy is False

- **Output**

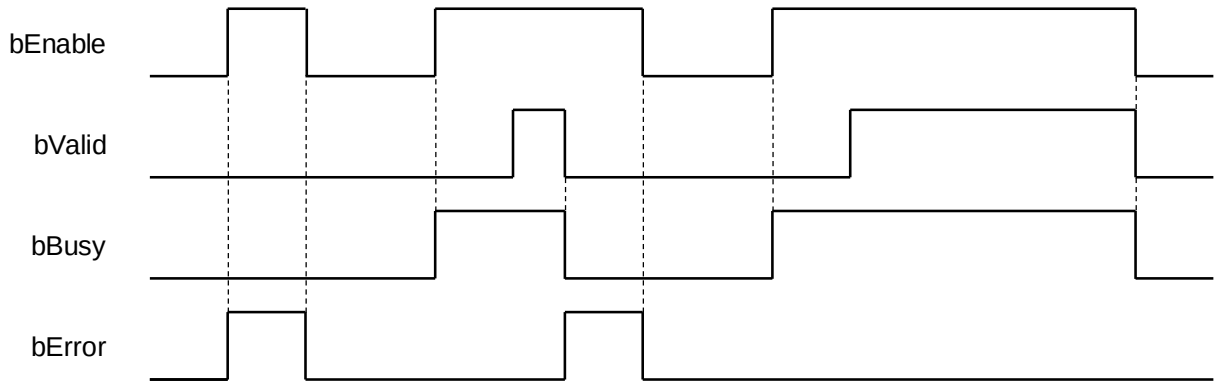
Name	Function	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
diSampleValue	Increasing number of the count value during each sampling period.	DINT	Positive number, negative number or 0(0)

***Note:** DFB_HSIO_ERROR: Enumeration(Enum)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bValid	• When the values at the outputs are valid after bEnable being True for one scan cycle.	• When bEnable shifts to False. • When bError shifts to True.
bBusy	• When bEnable is rising edge triggered.	• When bEnable shifts to False. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When bEnable shifts from True to False. (Error code is cleared)
ErrorID		
diSampleValue	• Updates value continuously when bValid is True.	• Updates value continuously when bValid is True.

- **Timing Diagram**



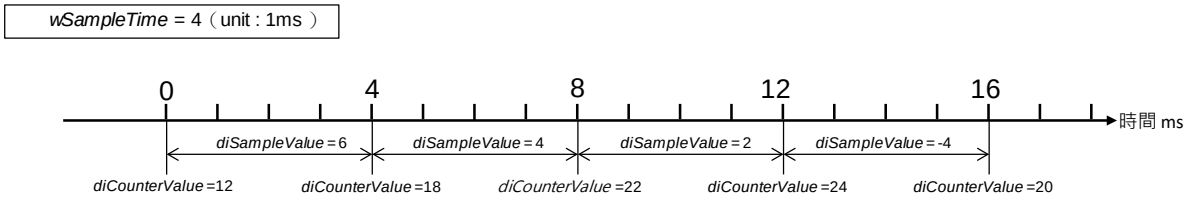
- **In/ Outs**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Counter	Reference to the source of high-speed counter.	DFB_COUNTER_REF(FB)*	DFB_COUNTER_REF(Cannot be null.)	When bEnable shifts to True and bBusy is False.

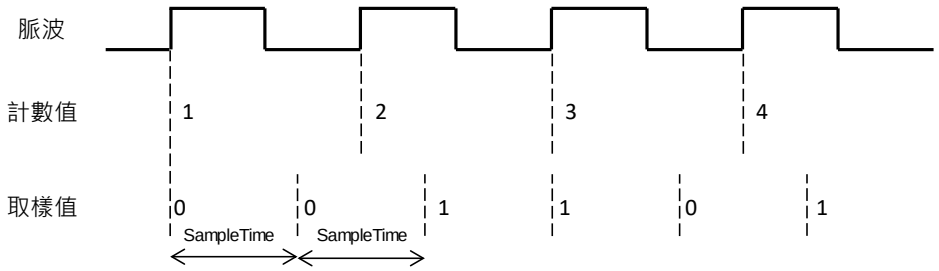
***Note:** DFB_COUNTER_REF(FB): As the I/O interface of the high-speed counter to perform actions include parameter adjustment and the driver.

- **Function**

1. DFB_Sample counts incoming pulses during a specified sampling period (wSampleTime).



2. When wSampleTime is shorter than the pulse period, the increasing number (diSampleValue) would be shown between 0 and 1 for each SampleTime.



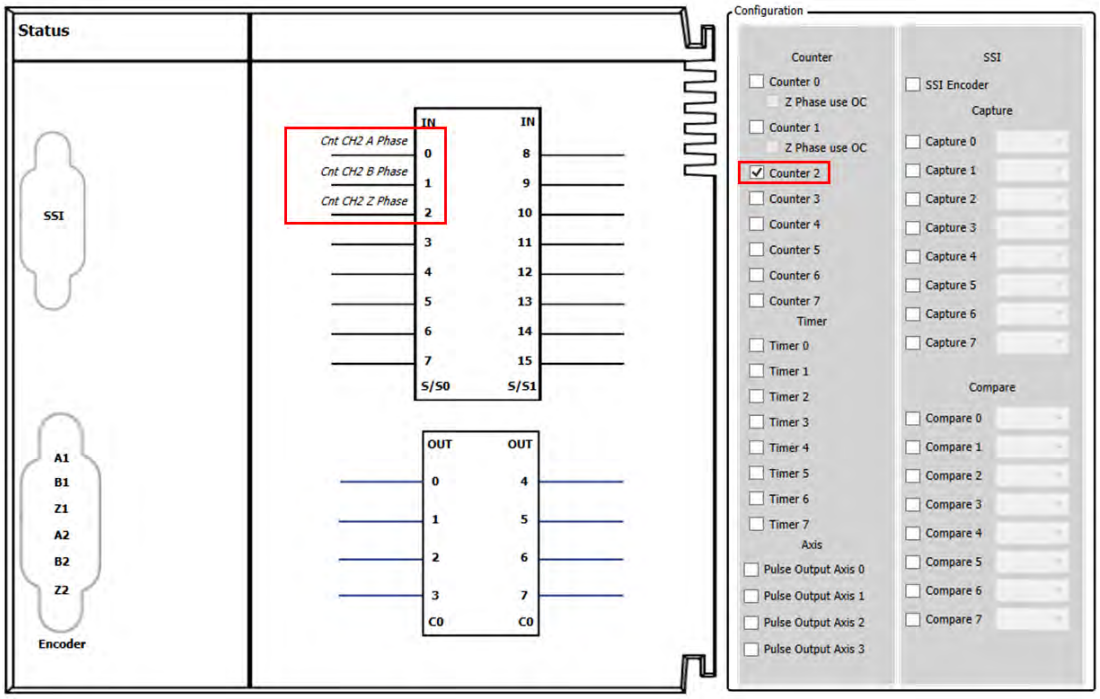
- **Troubleshooting**

If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

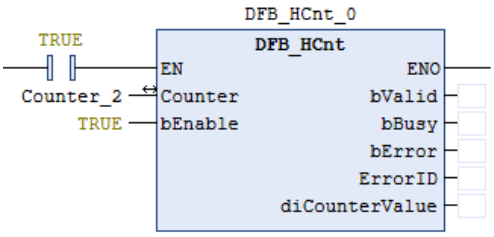
- **Programming Example**

This example uses DFB_HCnt and DFB_Sample to perform pulse counting during the sampling period.

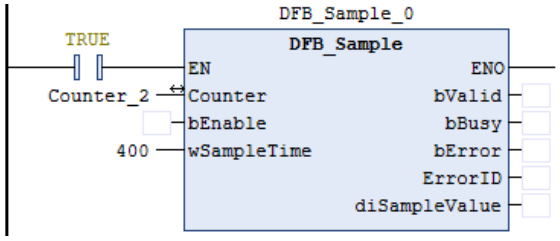
1. As the following figure shows, select a Counter (Counter 2) in Hardware IO Configuration and you will see the input points (e.g. IN 0、IN 1、IN 2) matched to the corresponding encoder A, B, Z phase outputs, which the wiring should follows the configuration so as to perform the normal function of high speed counting



2. After using the FB DFB_HCnt in the POU to activate the high-speed counter(bEnable = True), it starts receiving and counting the pulses from the external signals(IN 0, IN 1) based on the counting mode set in Counter Configuration, then the count value would be displayed in the output diCounterValue. In addition, you should make sure that the mode of sending pulses from the external signal source matches the counting mode so as to get the correct count values



3. After enabling DFB_Sample in the POU (bEnable = True), the FB starts counting the increasing number of the pulse count value during each sampling period.



4. Please refer to AX-3 series operational manual for more details related to the settings and operation of Counter Configuration.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_BuiltInIO.library

3.7 Error Codes and Troubleshooting

The following table lists the error codes corresponding to the FBs and the contents of the errors:

Description	Cause of Error	Corrective Action
DFB_HSIO_NO_ERR	No error messages.	
DFB_CAP_INVALID_ CAPTURE_REF	The variable type set for the FB input is not Capture_REF.	After make sure Capture in IO Configuration is selected, input the variable of IEC Object to the “Capture” input of DFB_Capture.
DFB_CAP_INVALID_ COUNTER_REF	The variable type set for the FB input is not Counter_REF	After make sure Counter in IO Configuration is selected, input the variable of IEC Object to the “Counter” input of DFB_Capture.
DFB_CAP_INVALID_ VALUE_SETTING	The mask range of DFB_Capture (uiMaskValue) exceeds the rotation range of the axis.	Reset the input value of uiMaskValue to be in the rotation range of encoder axis. [0 ~ EncoderAxis.Modulo Value]
DFB_CAP_INVALID_ DELTARANGE	When a rotary axis is used as the encoder axis, the min/max difference between each Capture exceeds the rotation range.	Reset the input value of “diDeltaMax” or “diDeltaMin” to be in the rotation range of encoder axis. [0 ~ EncoderAxis.Modulo Value]
DFB_CAP_CAPTURE_AL READY_ENABLE	The high-speed capture device has been activated.	Check if this Capture device is currently being used by another DFB_Capture.
DFB_CAP_DRIVE_ ERROR	Errors occur in the Capture device or Count device driver.	Check the error message on the BuiltIn_IO page and refer to the AX-3 operational manual to troubleshoot the errors.
DFB_CMP_INVALID_ COMPARE_REF	The variable type set for the FB input is not Compare_REF	After make sure Compare in IO Configuration is selected, input the variable of IEC Object to the “Compare” input of DFB_Compare.
DFB_CMP_INVALID_ COUNTER_REF	The variable type set for the FB input is not Counter_REF	After make sure Counter in IO Configuration is selected, input the variable of IEC Object to the “Counter” input of DFB_Capture.
DFB_CMP_INVALID_ CMPVALUE	When a rotary axis is used as the encoder axis, the input "diCompareValue" exceeds the rotation range.	Reset the input value of diCompareValue to be in the rotation range of encoder axis. [0 ~ EncoderAxis.Modulo Value]
DFB_CMP_INVALID_ REFRESHCYCLE	The input "wRefreshCycle" exceeds the range of ~30000 (Unit: 0.1us)	Set the value of "wRefreshCycle" to be within the range of 0 ~ 30000.
DFB_CMP_COMPARE_A LREADY_ENABLE	The high-speed comparator has been activated.	Check if this Compare device is currently being used by another DFB_Compare.
DFB_CMP_DRIVE_ ERROR	Errors occur in the Compare device or Count device driver.	Check the error message on the BuiltIn_IO page and refer to the AX-3 operational manual to troubleshoot the errors.
DFB_HC_INVALID_ COUNTER_REF	The variable type set for the FB input is not Counter_REF	After make sure Counter in IO Configuration is selected, input the variable of IEC Object to the “Counter” input of DFB_Hcnt.
DFB_HC_COUNTER_ ALREADY_ENABLE	The high-speed counter has been activated.	Check if this Counter device is currently being used by another DFB_HCnt.
DFB_HC_COUNTER_ REF_CHANGED_ DURING_OPERATION	The input value of “Counter” is changed while the FB is being executed.	Check if the value of the input Counter changes after the FB DFB_HCnt being executed.
DFB_HC_COUNTER_ DRIVE_ERROR	Errors occur in the Count device driver.	Check the error message on the BuiltIn_IO page and refer to the AX-3 operational manual to troubleshoot the errors.
DFB_HT_INVALID_ TIMER_REF	The variable type set for the FB input is not Timer_REF	After make sure Timer in IO Configuration is selected, input the variable of IEC Object to the “Timer” input of DFB_HTmr.

Description	Cause of Error	Corrective Action
DFB_HT_TIMER_ ALREADY_ENABLE	The high-speed timer has been activated.	Check if this Timer device is currently being used by another DFB_HTmr.
DFB_HT_TIMER_REF_C HANGED_DURING_ OPERATION	The input value of “Timer” is changed while the FB is being executed.	Check if the value of the input Timer changes after the FB DFB_HTmr being executed.
DFB_HT_TIMER_ DRIVE_ERROR	Errors occur in the Timer device driver.	Check the error message on the BuiltIn_IO page and refer to the AX-3 operational manual to troubleshoot the errors.
DFB_PV_INVALID_ COUNTER_REF	The variable type set for the FB input is not Counter_REF.	After make sure Counter in IO Configuration is selected, input the variable of IEC Object to the “Counter” input of DFB_PresetValue.
DFB_PV_NOT_ ENABLE_EXTERNAL_TRI GGER	“External Trigger” in Counter mode configuration is not selected while the input TriggerType of DFB_PresetValue is set to “EXTERNAL_TRIGGER”.	Please check the box of External Trigger on the Counter configuration page.
DFB_PV_PREVIOUS_ PRESET_NOT_DONE	The preset value function of the counter has been used by other DMC_PresetValue FBs.	Please wait for the previous preset value task of another DFB_PresetValue completed, then you’ll be able to execute the current task.
DFB_PV_CANNOT_ PRESET_WHEN_ SAMPLING	The counter is executing DFB_Sample.	Disable DFB_Sample of the counter to turn off the Sample function in this counter.
DFB_PV_SETRING_ NOT_DONE	The counter is executing DFB_SetRing and not completed.	Please wait for the counter to finish executing DFB_SetRing and then DFB_PresetValue can be executed.
DFB_PV_INVALID_ PRESET_VALUE	When a rotary axis is used as the encoder axis, the input " diPresetValue" exceeds the rotation range.	Reset the input value of diPresetValue to be in the rotation range of encoder axis. [0 ~ EncoderAxis.Modulo Value]
DFB_PV_COUNTER_ REF_CHANGED_ DURING_OPERATION	The input value of “Counter” is changed while the FB is being executed.	Check if the value of the input Counter changes after the FB DFB_PresetValue being executed.
DFB_PV_COUNTER_ DRIVE_ERROR	Errors occur in the Timer device driver.	Check the error message on the BuiltIn_IO page and refer to the AX-3 operational manual to troubleshoot the errors.
DFB_SP_INVALID_ COUNTER_REF	The variable type set for the FB input is not Counter_REF	After make sure Counter in IO Configuration is selected, input the variable of IEC Object to the “Counter” input of DFB_Sample.
DFB_SP_COUNTER_ NOT_ENABLE	DFB_Counter has not enabled the high-speed counter.	Make sure the counter device has been enabled by DFB_HCnt and then you can execute the FB DFB_Sample.
DFB_SP_ALREADY_ SAMPLING	The counter is executing DFB_Sample.	Check if this counter device is currently being used by another DFB_Sample.
DFB_SP_PRESET_ NOT_DONE	The counter is executing DFB_PresetValue and not completed.	Please wait for the counter to finish executing DFB_PresetValue and then DFB_Sample can be executed.
DFB_SP_INVALID_ SAMPLE_TIME	The input “wSampleTime” of DFB_Sample exceeds the range of 10~65535.	Reset the input value of “wSampleTime” to be in the range of 10 ~ 65535.
DFB_SP_COUNTER_ REF_CHANGED_ DURING_OPERATION	The input value of “Counter” is changed while the FB is being executed.	Check if the value of the input Counter changes after the FB DFB_Sample being executed.

Description	Cause of Error	Corrective Action
DFB_SP_COUNTER_ DRIVE_ERROR	Errors occur in the Counter device driver..	Check the error message on the BuiltIn_IO page and refer to the AX-3 operational manual to troubleshoot the errors.

MEMO

4

Chapter4 EtherCAT Network Instructions

Table of Contents

4.1 DFB_EcGetAllSlaveAddr

DFB_EcGetAllSlaveAddr gets all the slave addresses.

FB/FC	Instruction	Graphic Expression
FB	DFB_EcGetAllSlaveAddr	

ST language

```
DFB_EcGetAllSlaveAddr (
bExecute :=,
bDone =>,
bBusy =>,
bError =>,
ErrorID =>,
AddrArray =>,
uSlaves =>, );
```

4

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bExecute	Execute the instruction when bExecute changes to True.	BOOL	True/False (False)	-

- **Output**

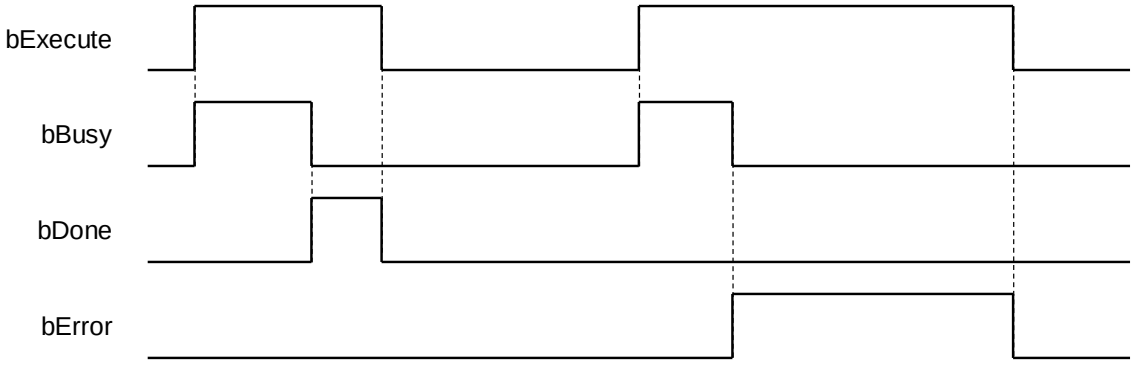
Name	Function	Data Type	Output Range (Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
AddrArray	Slave address array.	UINT[1..128]	(0)
uSlaves	The number of slaves.	UINT	0~128(0)

***Note:** DFB_ECAT_Diag_ERROR: Enumeration (Enum)

- **Output Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	• When the execution of FB is completed.	• When bExecute shifts to False. • If bExecute is False and bDone shifts to True, bDone will be True for only one period and immediately shift to False.
bBusy	• When bEnable is rising edge triggered.	• When bDone shifts to True. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When *Execute* shifts from True to False.(Error code is cleared)
ErrorID		
AddrArray	• When bExecute is rising edge triggered.	• When bExecute is falling edge triggered.
uSlaves	• When bExecute is rising edge triggered.	• When bExecute is falling edge triggered.

- **Timing Diagram**



- **Function**

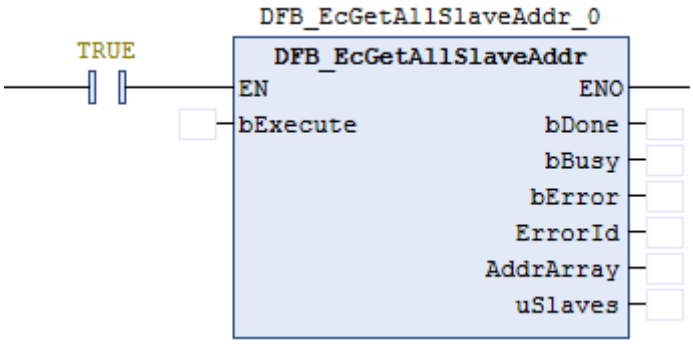
When bExecute shifts to True, the output AddrArray gives the addresses of all the EtherCAT slaves in the project tree, which supports up to 128 stations. Therefore, the maximum number of slave addresses output by AddrArray would be 128 given by the output uSlaves.

- **Troubleshooting**

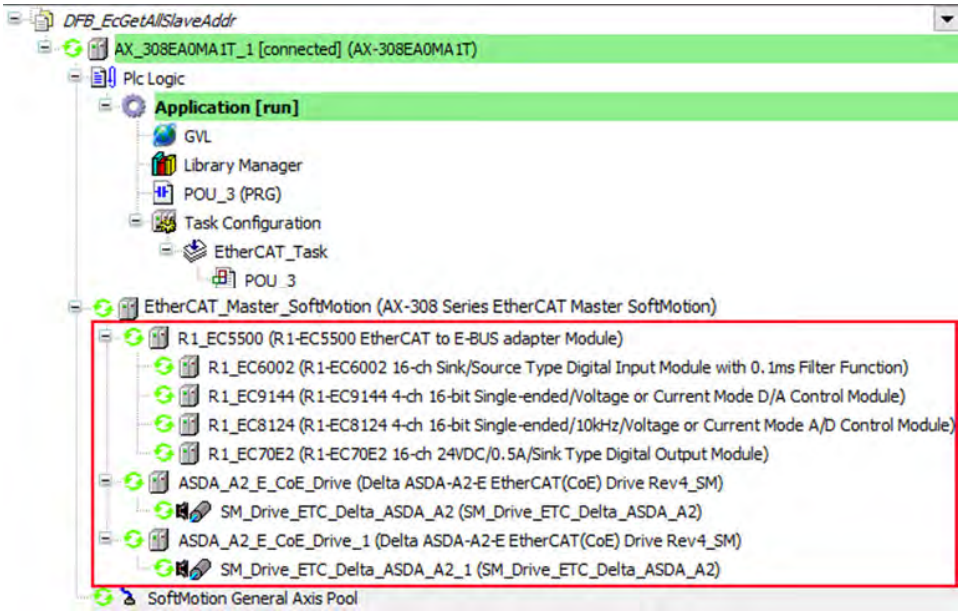
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

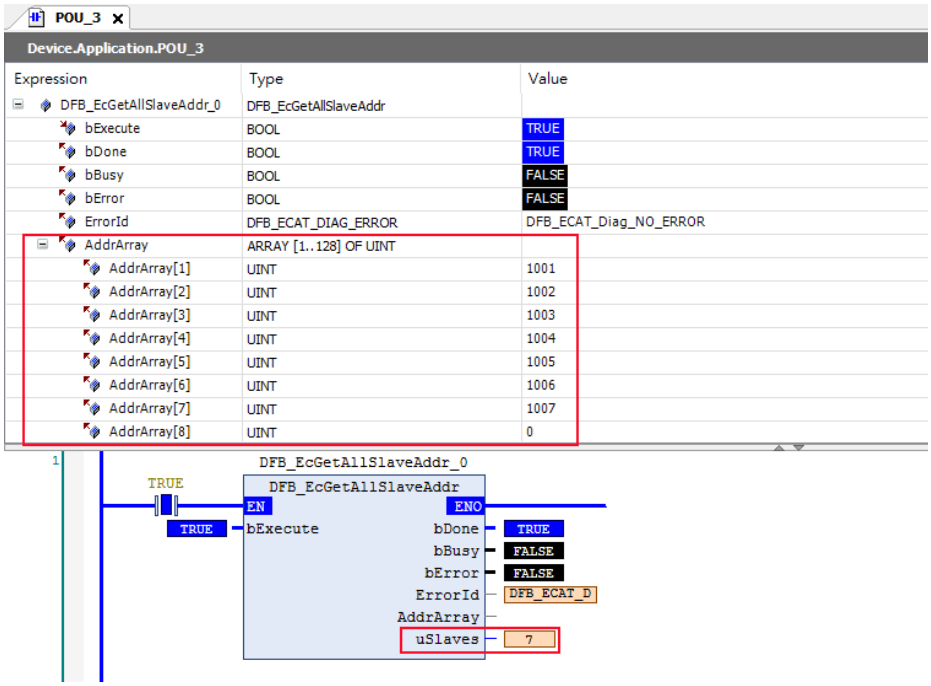
The following example demonstrates the behavior of DFB_EcGetAllSlaveAddr.



1. There're a total of 7 EtherCAT slaves in the category EtherCAT_Master_SoftMotion.



2. After the input bExecute of DFB_EcGetAllSlaveAddr bExecute shifts to True, the output of AddrArray is shown as below and the output value of uSlaves is 7.



- **Supported Products**
  - AX-308E

- **Library**
  - DL_EtherCAT_Diag.library

4.2 DFB_EcGetSlaveCount

DFB_EcGetSlaveCount gets the number of slaves that are connected to the master.

FB/FC	Instruction	Graphic Expression
FB	DFB_EcGetSlaveCount	

ST Language

```
DFB_EcGetSlaveCount (
bExecute :=,
bDone =>,
bBusy =>,
bError =>,
ErrorID =>,
uSlaves =>,
);
```


- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bExecute	Execute the instruction when bExecute changes to True.	BOOL	True/False (False)	-

- **Output**

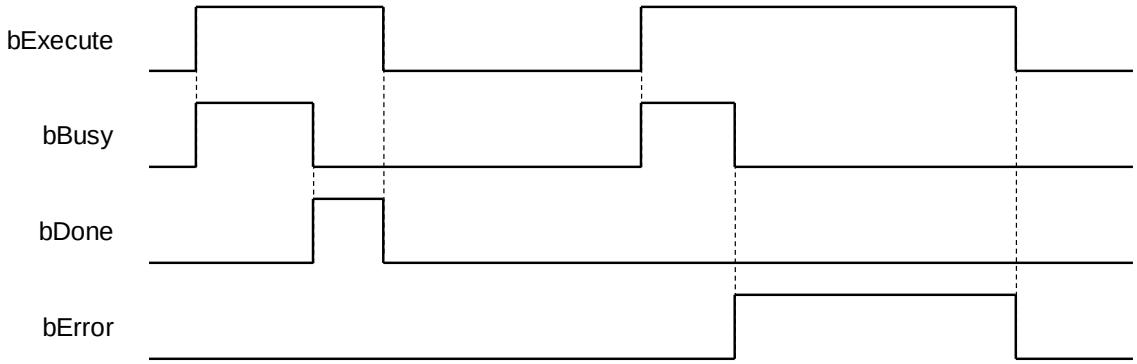
Name	Function	Data Type	Output Range (Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
uSlaves	The number of slaves.	UINT	0~128(0)

***Note:** DFB_ECAT_Diag_ERROR: Enumeration (Enum)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	• When the execution of FB is completed.	• When bExecute shifts to False. • If bExecute is False and bDone shifts to True, bDone will be True for only one period and immediately shift to False.
bBusy	• When bEnable is rising edge triggered.	• When bDone shifts to True. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When *Execute* shifts from True to False.(Error code is cleared)
ErrorID		
uSlaves	• When bExecute is rising edge triggered.	• When bExecute is falling edge triggered.

- **Timing Diagram**



- **Function**

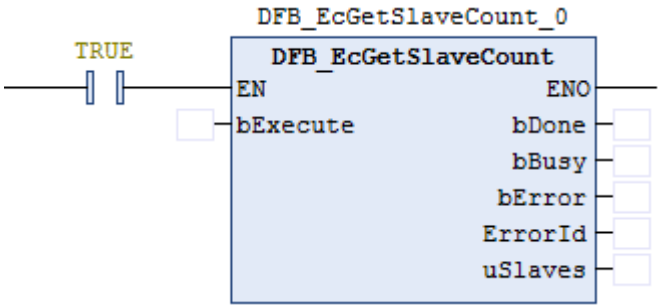
When bExecute shifts to True, the output uSlaves gives the number of EtherCAT slaves in the project tree.

- **Troubleshooting**

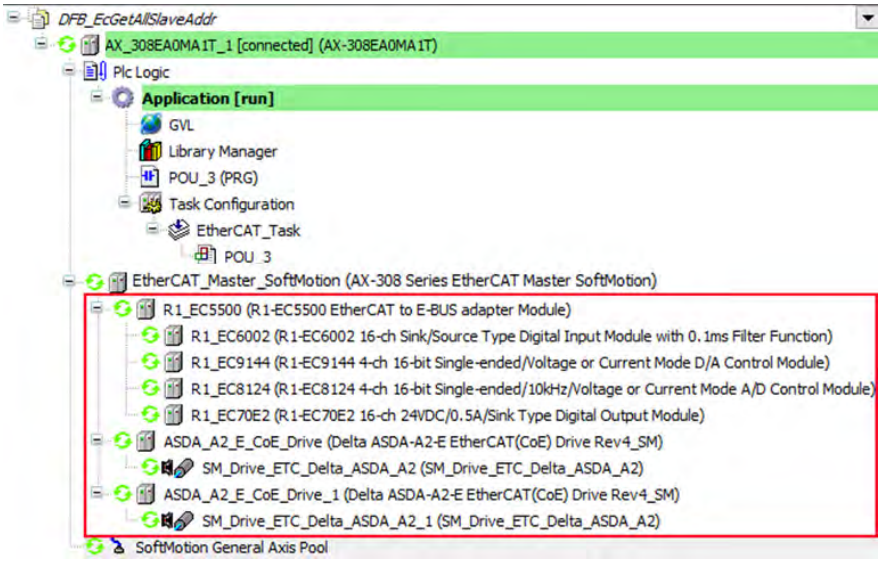
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

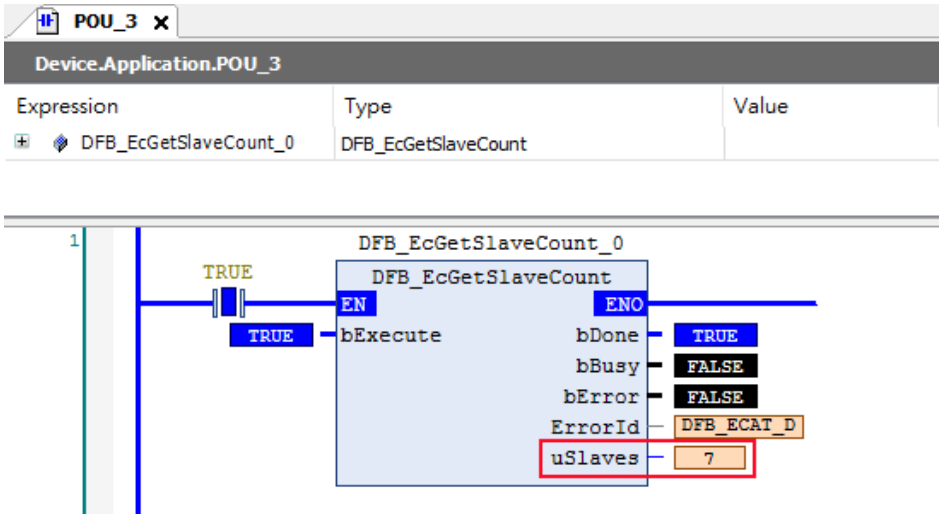
1. The following example demonstrates the behavior of DFB_EcGetSlaveCount.



2. There're a total of 7 EtherCAT slaves in the category EtherCAT_Master_SoftMotion.



3. When the input bExecute of DFB_EcGetSlaveCount shifts to True, the output value of uSlaves is 7.

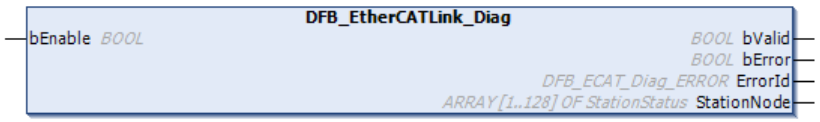


- **Supported Products**
  - AX-308E

- **Library**
  - DL_EtherCAT_Diag.library

4.3 DFB_EtherCATLink_Diag

DFB_EtherCATLink_Diag is used to display all the EtherCAT slave diagnostics.

FB/FC	Instruction	Graphic Expression
FB	DFB_EtherCATLink_Diag	

ST Language

```
DFB_EtherCATLink_Diag (
bEnable :=,
bValid =>,
bError =>,
ErrorID =>,
StationNode =>
);
```


- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-

- **Output**

Name	Function	Data Type	Output Range (Default value)
bValid	True when the instruction is being executed.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*1	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
StationNode	Slave addresses and structure array of slave status.	StationStatus [1..128] *2 *3	StationStatus

***Note:**

1. DFB_ECAT_Diag_ERROR: Enumeration (Enum)
2. StationStatus Structure (STRUCT)

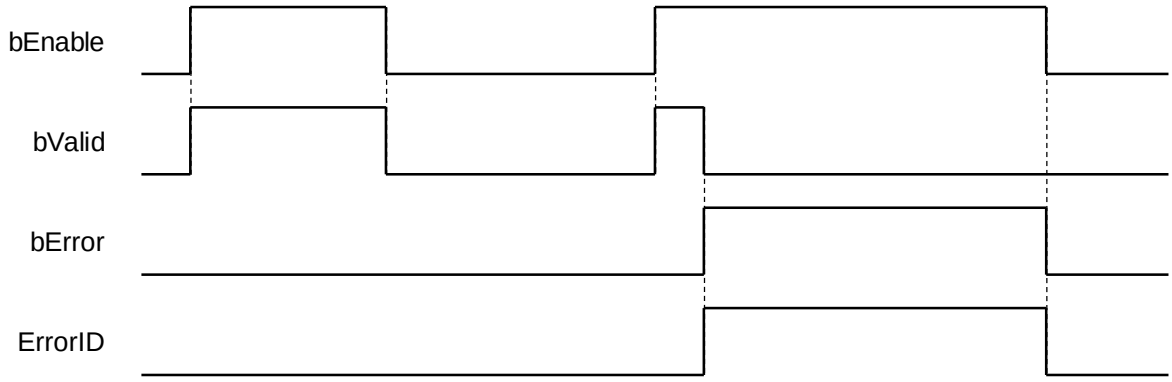
Name	Function	Data Type	Setting Value (Default value)
StationAddress	Slave station address	UINT	(0)
Node	Connection status of slave stations	BOOL	True: Connected and functioning properly. False: Abnormal connection status. (False)

3. The array includes all the slave addresses and connection status, which starts from the first slave station. (Supports up to 128 stations) In addition, If the value of StationAddress is shown as 0 in the struct array after bEnable is rising edge triggered, it indicates that the slave station does not exist.

■ **Outputs Updating Time**

Name	Timing for shifting to True	Timing for shifting to False
bValid	• When bEnable is rising edge triggered.	• When bEnable shifts to False. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When bEnable shifts from True to False. (Error code is cleared)
ErrorID		
StationNode	• When bEnable is rising edge triggered.	• When bEnable shifts to False.

- **Timing Diagram**



- **Function**

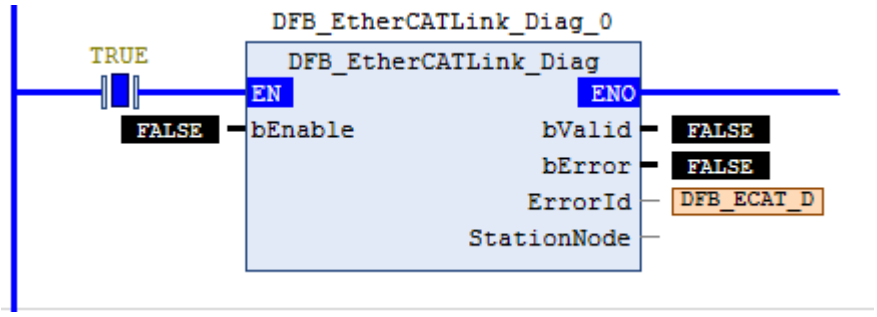
When bEnable shifts to True, StationAddress and Node output from StationAddress are in array type to show all the slave addresses and status with the support up to 128 slave stations. If the value of StationAddress is shown as 0 in the struct array after bEnable is rising edge triggered, it indicates that the slave station does not exist. An error will be reported by the function block if EtherCAT master is not found when bEnable shifts to True.

- **Troubleshooting**

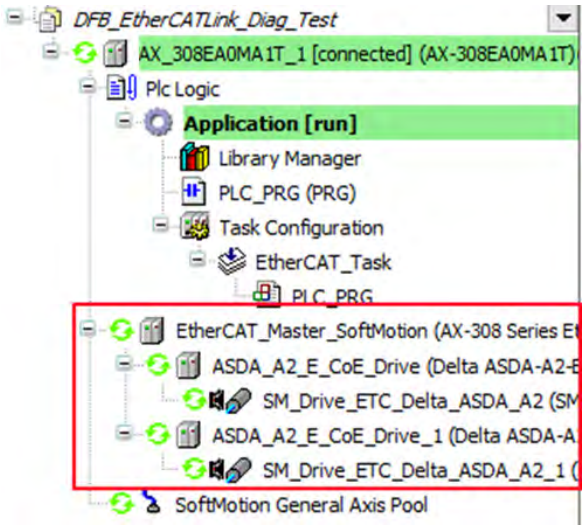
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

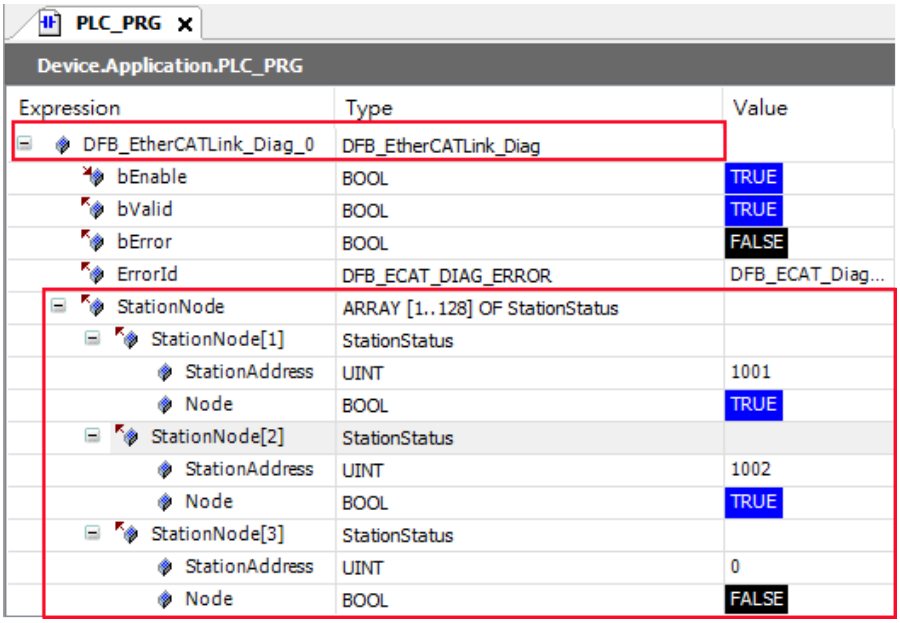
The following example demonstrates the behavior of DFB_EtherCATLink_Diag.



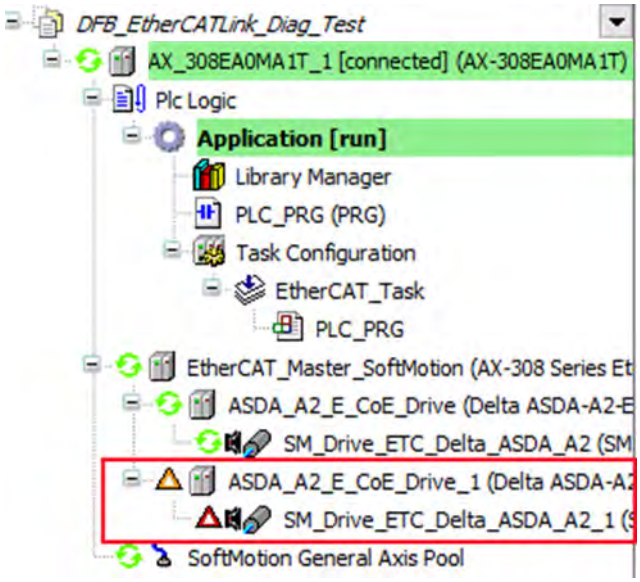
1. There's a total of two EtherCAT slave stations in the Device tree and the connection status shows PASS.



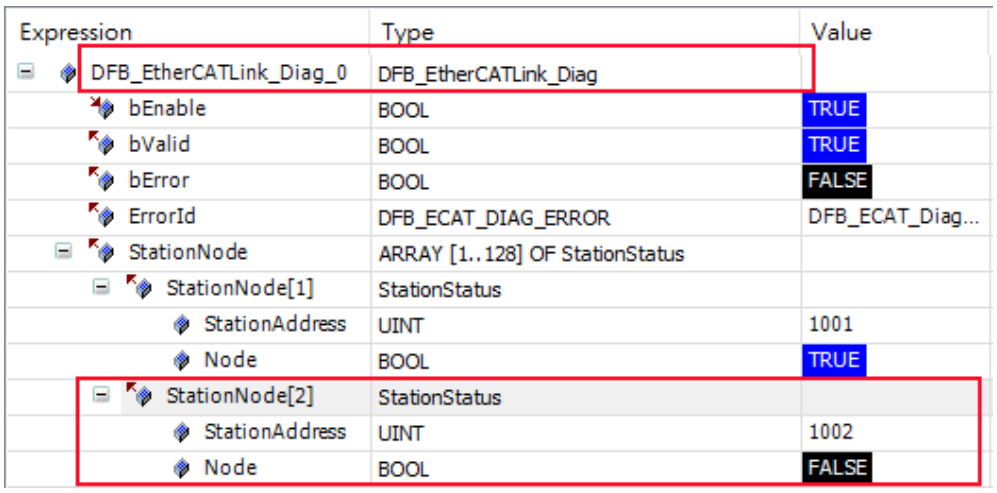
2. After bEnable of DFB_EtherCATLink_Diag shifts to True, the arrays of StationNode index 1 and 2 show the slave addresses and connection status, while the values of StationAddress, starting from index 3 of StationNode, would be shown as zero.



3. Disconnect the cable for internet connection between slave station 1 and 2 and you can see the status of slave 2 is shown to be Fail in the device tree.



4. The Node value of StationNode index 2 would also be displayed as False.



Expression	Type	Value
DFB_EtherCATLink_Diag_0	DFB_EtherCATLink_Diag	
bEnable	BOOL	TRUE
bValid	BOOL	TRUE
bError	BOOL	FALSE
ErrorId	DFB_ECAT_DIAG_ERROR	DFB_ECAT_Diag...
StationNode	ARRAY [1..128] OF StationStatus	
StationNode[1]	StationStatus	
StationAddress	UINT	1001
Node	BOOL	TRUE
StationNode[2]	StationStatus	
StationAddress	UINT	1002
Node	BOOL	FALSE

- **Supported Products**
  - AX-308E

- **Library**
  - DL_EtherCAT_Diag.library

4.4 DFB_GetAllECATSlaveInfo

DFB_GetAllECATSlaveInfo gets all the slaves' information.

FB/FC	Instruction	Graphic Expression
FB	DFB_GetAllECATSlaveInfo	

ST Language

```
DFB_GetAllECATSlaveInfo (
bExecute :=,
bDone =>,
bBusy =>,
bError =>,
ErrorID =>,
slaveInfoArray =>,
uSlaves =>,
);
```

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bExecute	Execute the instruction when bExecute changes to True.	BOOL	True/False (False)	-

- **Output**

Name	Function	Data Type	Output Range (Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*1	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
slaveInfoArray	Slave information array.	ECATSlaveInfo [1..128] *2	ECATSlaveInfo
uSlaves	The number of slaves.	UINT	0~128(0)

***Note:**

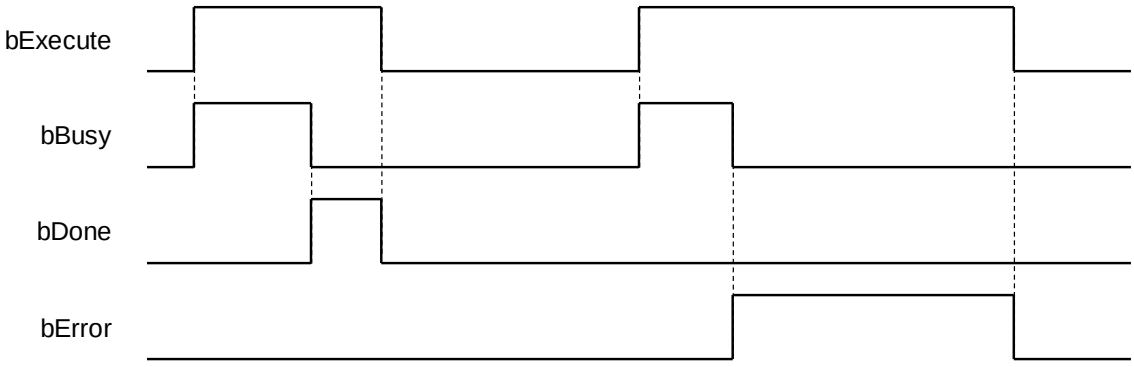
1. DFB_ECAT_Diag_ERROR: Enumeration(Enum)
2. slaveInfoArray: Structure(STRUCT)。

Name	Function	Data Type	Output Range (Default value)
vendorId	Slave vendor id	UDINT	(0)
productCode	Slave product code	UDINT	(0)
revisionNo	Slave revision number	UDINT	(0)
serialNo	Slave serial number	UDINT	(0)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	• When the execution of FB is completed.	• When bExecute shifts to False. • If bExecute is False and bDone shifts to True, bDone will be True for only one period and immediately shift to False.
bBusy	• When bExecute is rising edge triggered.	• When bDone shifts to True. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When b*Execute* shifts from True to False.(Error code is cleared)
ErrorID		
slaveInfoArray	• When bExecute is rising edge triggered.	• When *Execute* shifts from True to False.
uSlaves	• When bExecute is rising edge triggered.	• When *Execute* shifts from True to False.

- **Timing Diagram**



- **Function**

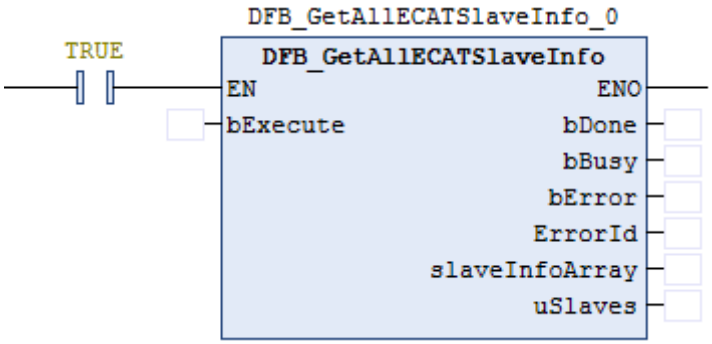
When bExecute shifts to True, slaveInfoArray gives the information of all the EtherCAT slaves in the device tree, which includes vendor id, product code, revision number and serial number. Support up to 128 stations as well as the maximum number of slaves and the corresponding information output from uSlaves and slaveInfoArray.

- **Troubleshooting**

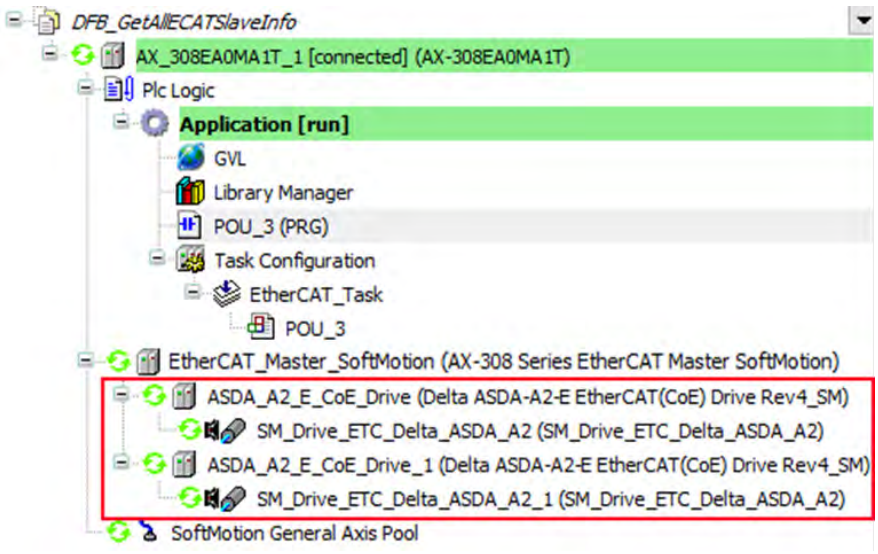
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

The following example demonstrates the behavior of DFB_GetAllECATSlaveInfo.

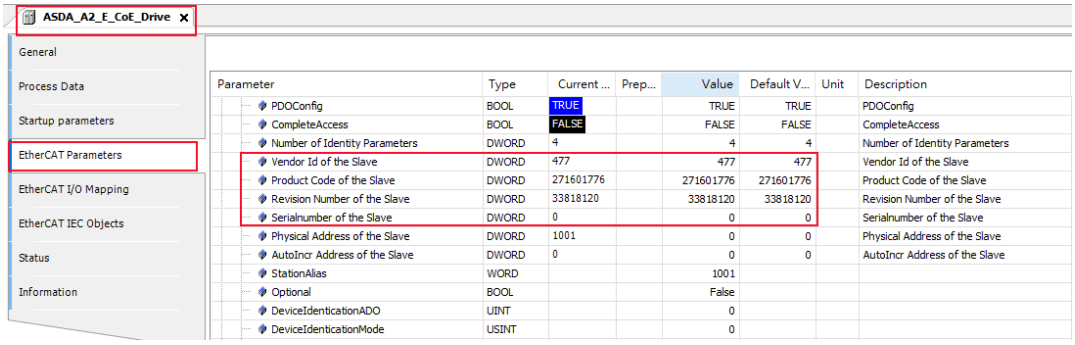


1. There're two EtherCAT slave stations in the device tree, both are ASDA_A2.



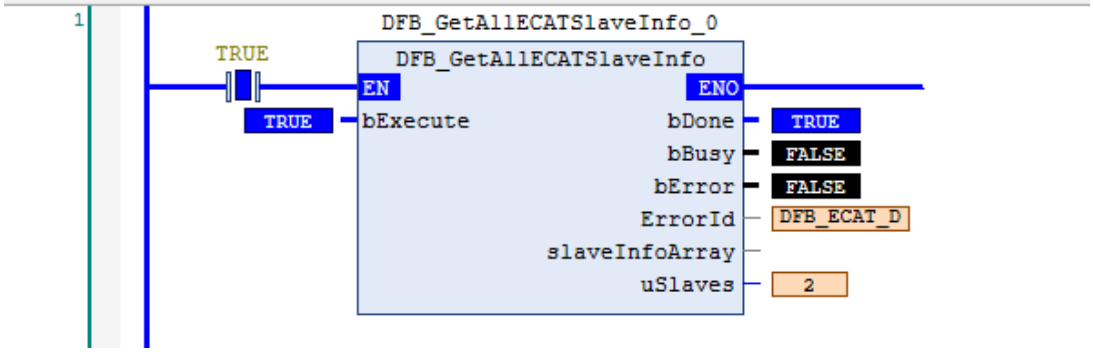
4

2. Double-click on the target ASDA_A2 in the device tree to view its slave information.



3. The input bExecute of DFB_GetAllECATSlaveInfo bExecute shifts to True, then the output of slaveInfoArray is shown as below and the output value of uSlaves is 2.

Expression	Type	Value
DFB_GetAllECATSlaveInfo_0	DFB_GetAllECATSlaveInfo	
bExecute	BOOL	TRUE
bDone	BOOL	TRUE
bBusy	BOOL	FALSE
bError	BOOL	FALSE
ErrorId	DFB_ECAT_DIAG_ERROR	DFB_ECAT_Diag_NO_ERROR
slaveInfoArray	ARRAY [1..128] OF ECATSlaveInfo	
slaveInfoArray[1]	ECATSlaveInfo	
vendorId	UDINT	477
productCode	UDINT	271601776
revisionNo	UDINT	33818120
serialNo	UDINT	0
slaveInfoArray[2]	ECATSlaveInfo	
vendorId	UDINT	477
productCode	UDINT	271601776
revisionNo	UDINT	33818120
serialNo	UDINT	0

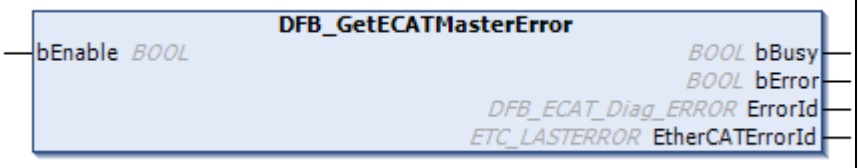


- **Supported Products**
  - AX-308E

- **Library**
  - DL_EtherCAT_Diag.library

4.5 DFB_GetECATMasterError

DFB_GetECATMasterError gets the error code of failed EtherCAT network connection.

FB/FC	Instruction	Graphic Expression
FB	DFB_GetECATMasterError	

ST Language

```
DFB_GetECATMasterError (
bEnable :=,
bBusy =>,
bError =>,
ErrorID =>,
EtherCATErrorId =>
);
```


- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-

- **Output**

Name	Function	Data Type	Output Range (Default value)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ ERROR*1	DFB_ECAT_Diag_ERROR(DFB_ECAT _Diag_NO_ERROR)
EtherCATErrorId	EtherCAT error codes	ETC_LASTERROR*2	ETC_LASTERROR(NO_ERROR)

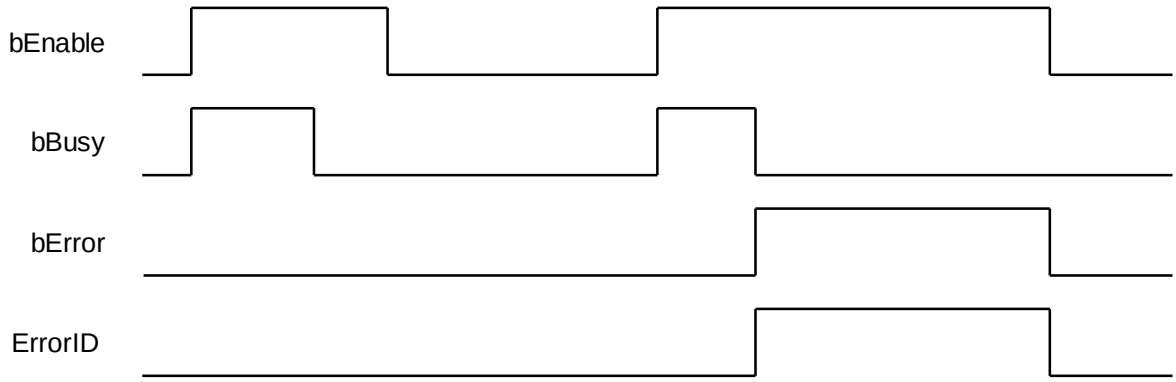
***Note:**

1. DFB_ECAT_Diag_ERROR: Enumeration (Enum)
2. ETC_LASTERROR: Enumeration (Enum)

■ **Outputs Updating Time**

Name	Timing for shifting to True	Timing for shifting to False
bBusy	• When bEnable is rising edge triggered.	• When bEnable shifts to False. • When bError shifts to True.
bError	• When an error occurs in the execution conditions for the instruction.	• When bEnable shifts from True to False.(Error code is cleared)
ErrorID		
EtherCATErrorId	• When an error occurs in the EtherCAT connection.	• When bEnable shifts to False.

- **Timing Diagram**



- **Function**

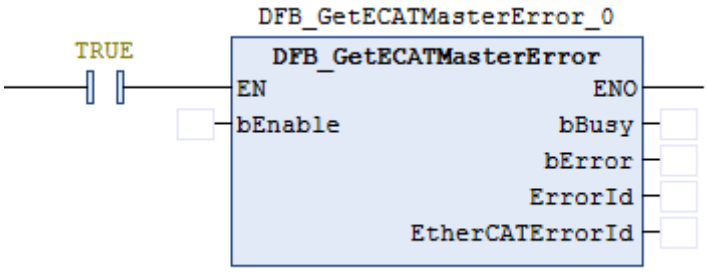
When bEnable shifts to True, the output EtherCATErrorId gives the error codes of failed EtherCAT network connection during each cycle. If there's no error, the output would be displayed as NO_ERROR. For more details of error codes, please refer to the content of ETC_LASTERROR_STATE in the Library.

- **Troubleshooting**

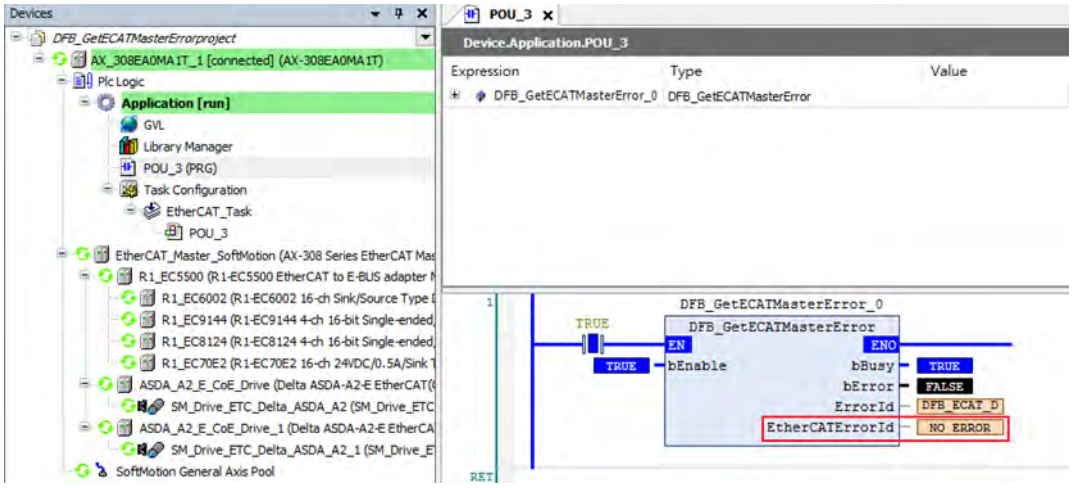
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

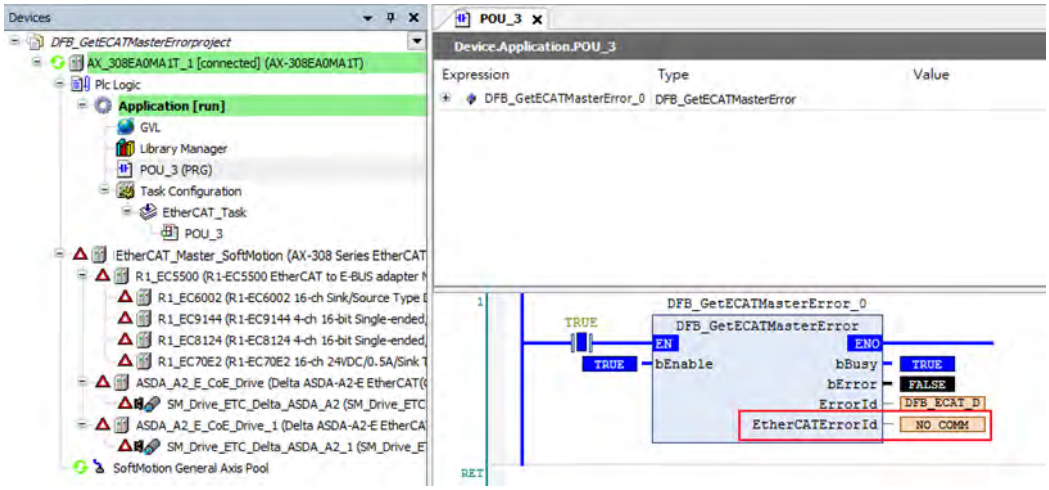
The following example demonstrates the behavior of DFB_GetECATMasterError.



1. If the EtherCAT connection is normal without any existing errors., the output content of EtherCATErrorId would be shown as NO_ERROR after the input bEnable of DFB_GetECATMasterError shifts to True.



2. Remove the network connection between the master and the slave, then the output content of EtherCATErrorId would be shown as NO_COMM.



- **Supported Products**
  - AX-308E

- **Library**
  - DL_EtherCAT_Diag.library

4.6 DFB_GetECATMasterState

DFB_GetECATMasterState gets the connection status of EtherCAT Master.

FB/FC	Instruction	Graphic Expression
FB	DFB_GetECATasterState	

ST Language

```
DFB_GetECATMasterState (
bEnable :=,
bBusy =>,
bError =>,
ErrorID =>,
bStatus =>,
);
```

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-

- **Output**

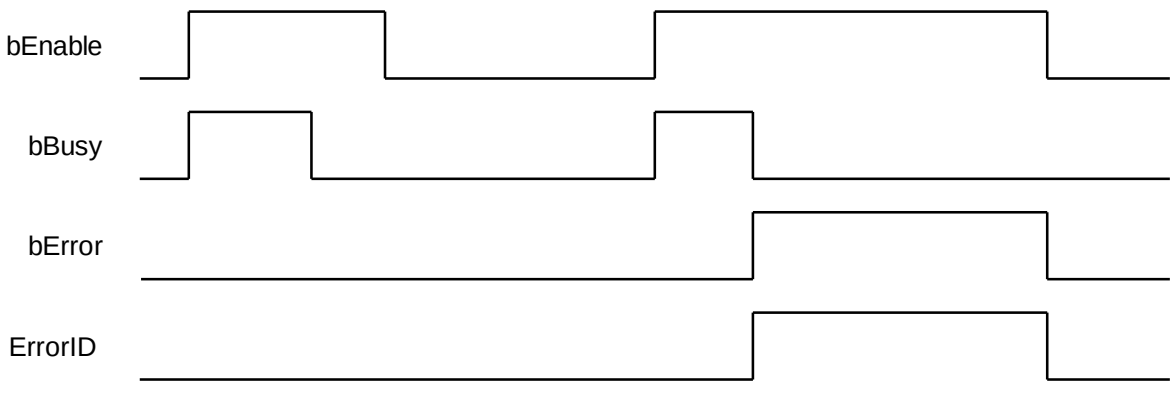
Name	Function	Data Type	Output Range (Default value)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
bStatus	EtherCAT master communication status.	BOOL	True/False(False)

***Note:** DFB_ECAT_Diag_ERROR: Enumeration (Enum)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bBusy	• When bEnable is rising edge triggered.	• When bError shifts to True.
bError ErrorID	• When an error occurs in the execution conditions for the instruction.	• When bEnable shifts from True to False.(Error code is cleared)
bStatus	• When the EtherCAT master connection is normal.	• When bEnable shifts to False. • When bError shifts to True • When the connection is abnormal.

- **Timing Diagram**



- **Function**

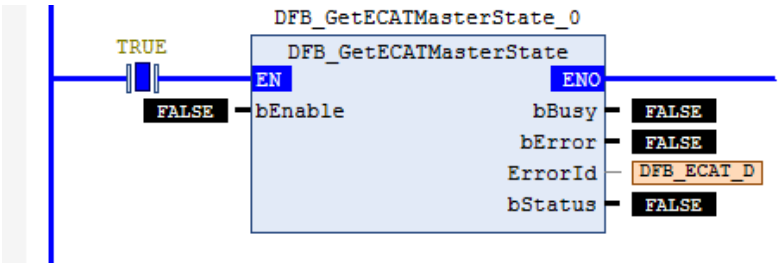
When bEnable shifts to True, the function block perform cyclical status updates of EtherCAT master communication.

- **Troubleshooting**

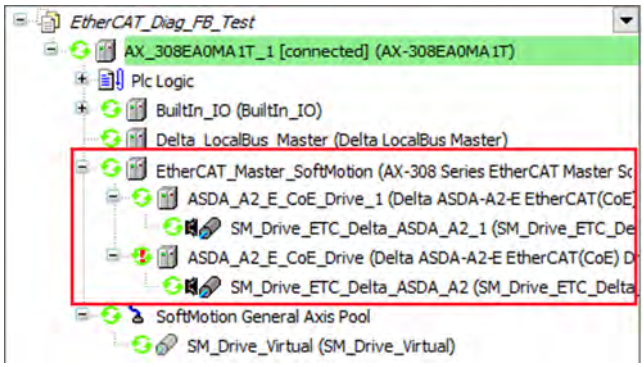
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

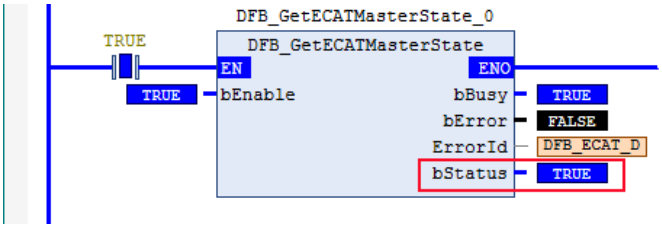
The following example demonstrates the behavior of DFB_GetECATMasterState.



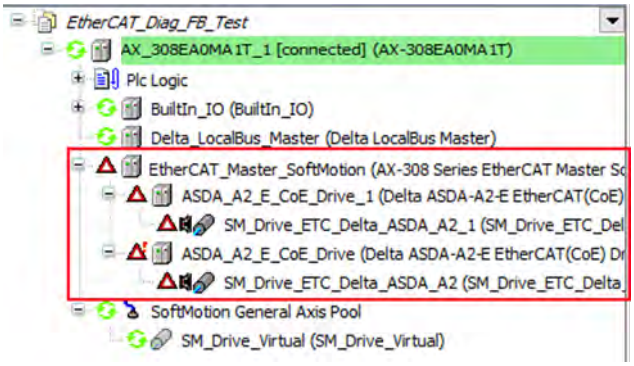
1. The connection status of EtherCAT master shows PASS in the device tree.



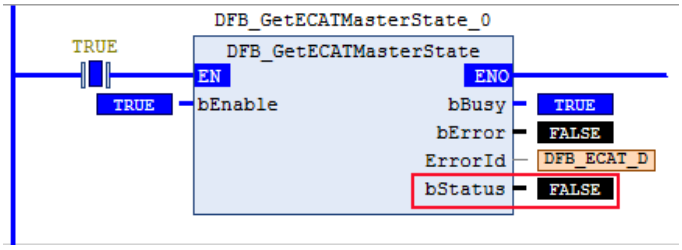
2. When the input bEnable of DFB_GetECATMasterState shifts to True, the output of bStatus is displayed as True.



3. Remove the network connection between the master and the slave, and the current connection status of EtherCAT master would show Fail in the device tree.



4. The output bStatus of DFB_GetECATMasterState is displayed as False.



- **Supported Products**
  - AX-308E

- **Library**

  DL_EtherCAT_Diag.library

4.7 DFB_ResetECATMaster

DFB_ResetECATMaster resets the EtherCAT master, which has errors in connection.

FB/FC	Instruction	Graphic Expression
FB	DFB_ResetECATMaster	

ST Language

```
DFB_ResetECATMaster (
bExecute :=,
bDone =>,
bBusy =>,
bError =>,
ErrorID =>,
);
```

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bExecute	Execute the instruction when bExecute changes to True.	BOOL	True/False (False)	-

- **Output**

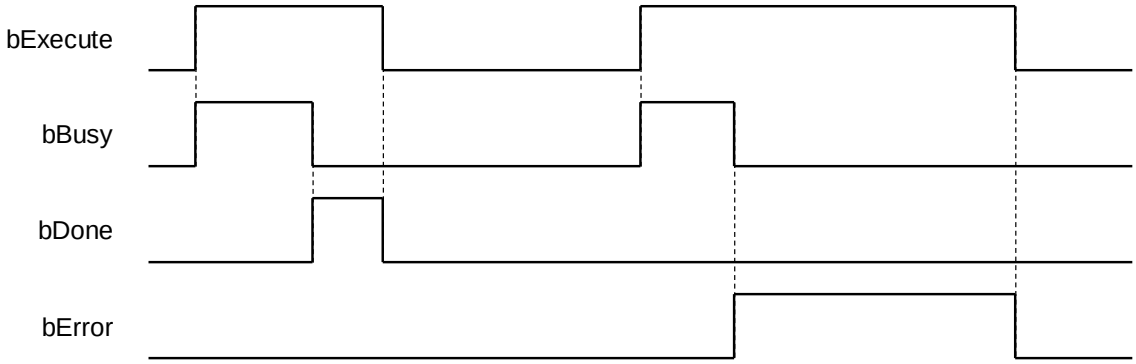
Name	Function	Data Type	Output Range (Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)

***Note:** DFB_ECAT_Diag_ERROR: Enumeration (Enum)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	• When the execution of FB is completed.	• When bExecute shifts to False. • If bExecute is False and bDone shifts to True, bDone will be True for only one period and immediately shift to False.
bBusy	• When bExecute is rising edge triggered.	• When bDone shifts to True. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When b*Execute* shifts from True to False.(Error code is cleared)
ErrorID		

- **Timing Diagram**



- **Function**

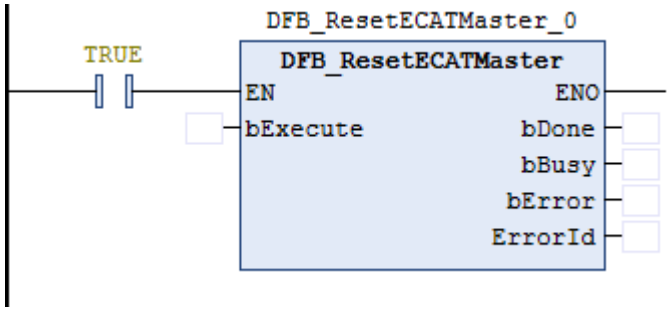
When bExecute shifts to True and the connection status of EtherCAT master shows Fail, the function block would perform reset action.

- **Troubleshooting**

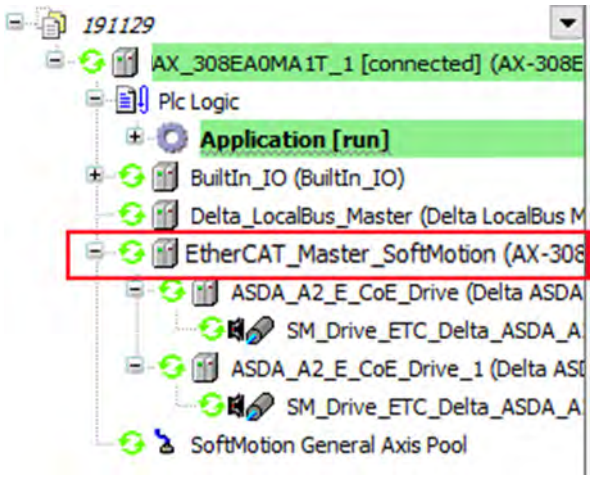
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

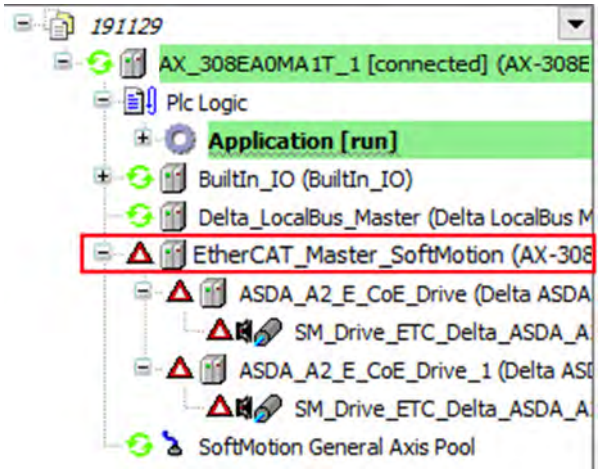
The following example demonstrates the behavior of DFB_ResetECATMaster.



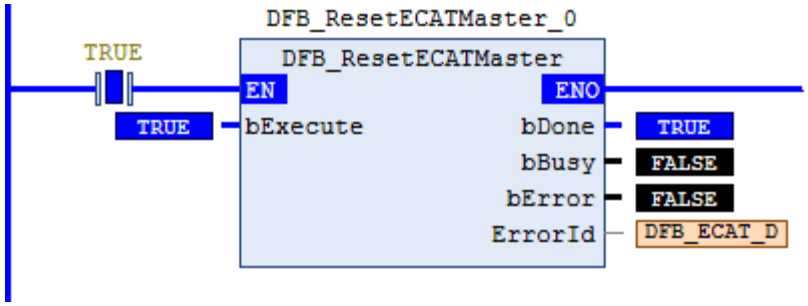
1. The connection status of EtherCAT master shows PASS in the device tree.



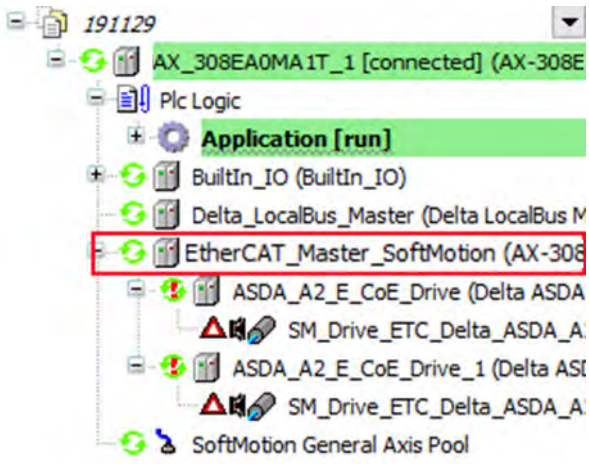
2. Remove the network connection between the master and the slave, and the current connection status of EtherCAT master would show Fail in the device tree.



3. To restore the network connection between the master and the slave, shift the input bExecute of DFB_ResetECATMaster to True.



4. The network connectivity has been recovered after the output bDone shifting to True.



- **Supported Products**
  - AX-308E

- **Library**
  - DL_EtherCAT_Diag.library

4.8 DFB_ResetECATSlave

DFB_ResetECATSlave resets the EtherCAT slave, which has errors in connection.

FB/FC	Instruction	Graphic Expression
FB	DFB_ResetECATSlave	

ST Language

```
DFB_ResetECATSlave(
bExecute :=,
uiSlaveAddr :=,
tTimeout :=,
bDone =>,
bBusy =>,
bError =>,
ErrorID =>,
);
```

- **Input**

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bExecute	Execute the instruction when bExecute changes to True.	BOOL	True/False (False)	-
uiSlaveAddr	Reset the slave address.	UINT	Positive number (0)	When bExecute is rising edge triggered and Busy is False
tTimeout	Slave resets the timeout.	TIME	Positive number (0)	When bExecute is rising edge triggered and Busy is False

- **Output**

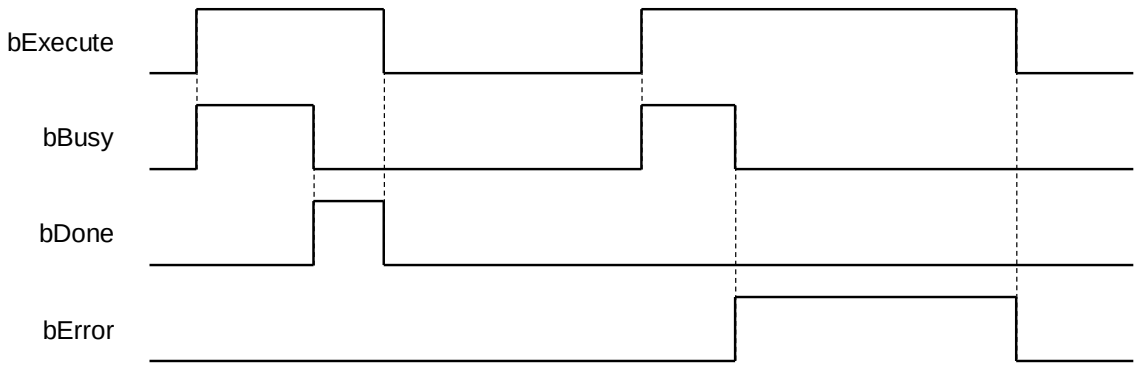
Name	Function	Data Type	Output Range (Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)

***Note:** DFB_ECAT_Diag_ERROR: Enumeration (Enum)

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	• When the execution of FB is completed.	• When bExecute shifts to False. • If bExecute is False and bDone shifts to True, bDone will be True for only one period and immediately shift to False.
bBusy	• When bExecute is rising edge triggered.	• When bDone shifts to True. • When bError shifts to True.
bError	• When an error occurs in the execution conditions or input values for the instruction.	• When b*Execute* shifts from True to False.(Error code is cleared)
ErrorID		

- **Timing Diagram**



- **Function**

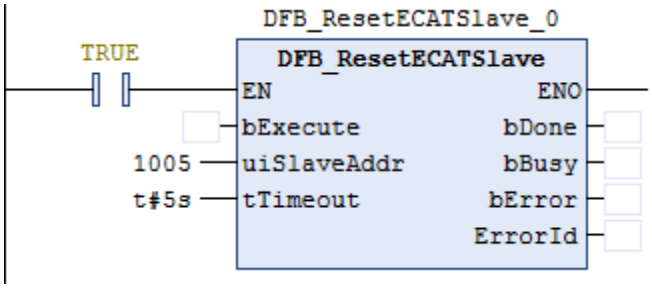
When bExecute shifts to True, the function block starts searching for the target slave station and resets the EtherCAT slave, if the status of target slave shows Fail. If the input value of uiSlaveAddr is 0, the function block would reset all the slave stations which have errors in connection.

- **Troubleshooting**

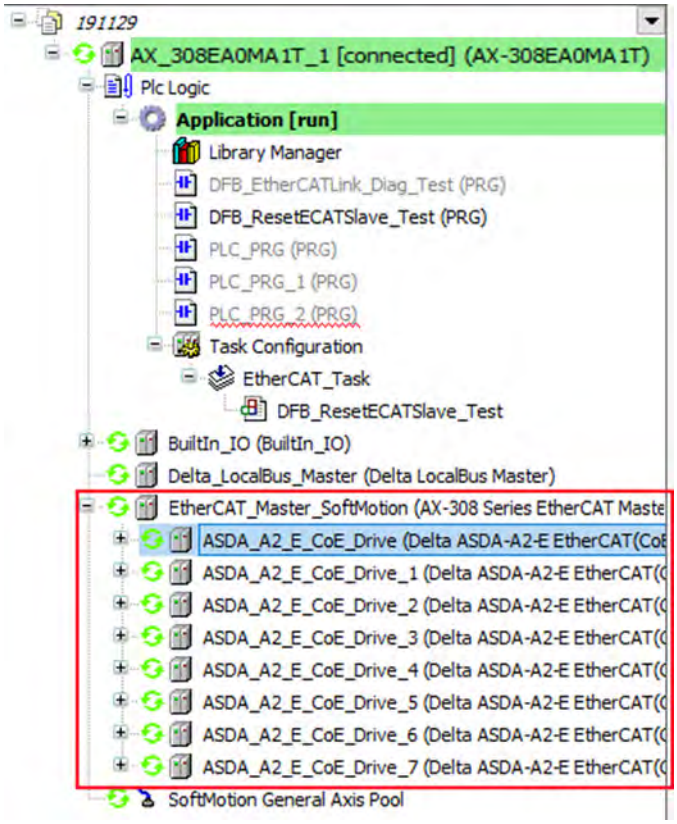
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

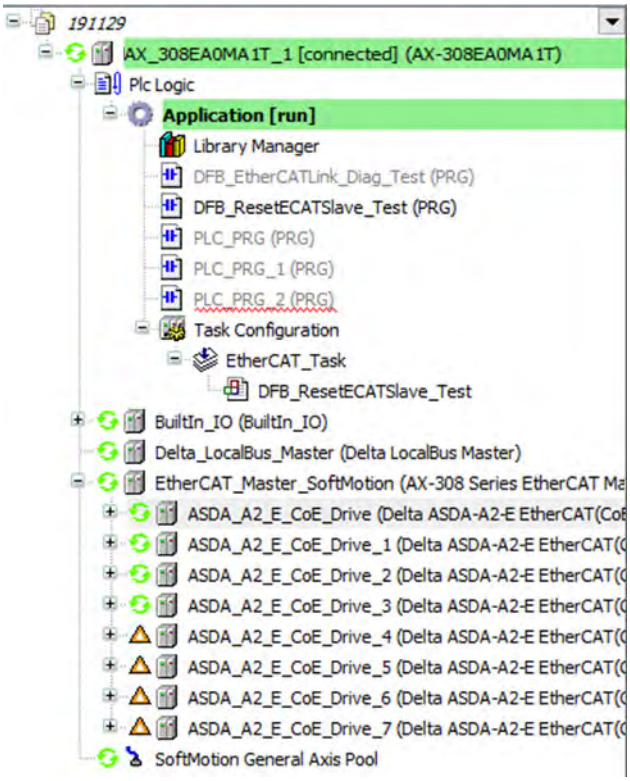
The following example demonstrates the behavior of DFB_ResetECATSlave.



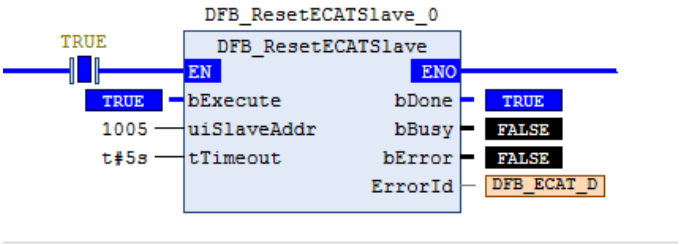
1. There's a total of 8 EtherCAT slave stations in EtherCAT_Master_SoftMotion and all their connection status shows PASS.



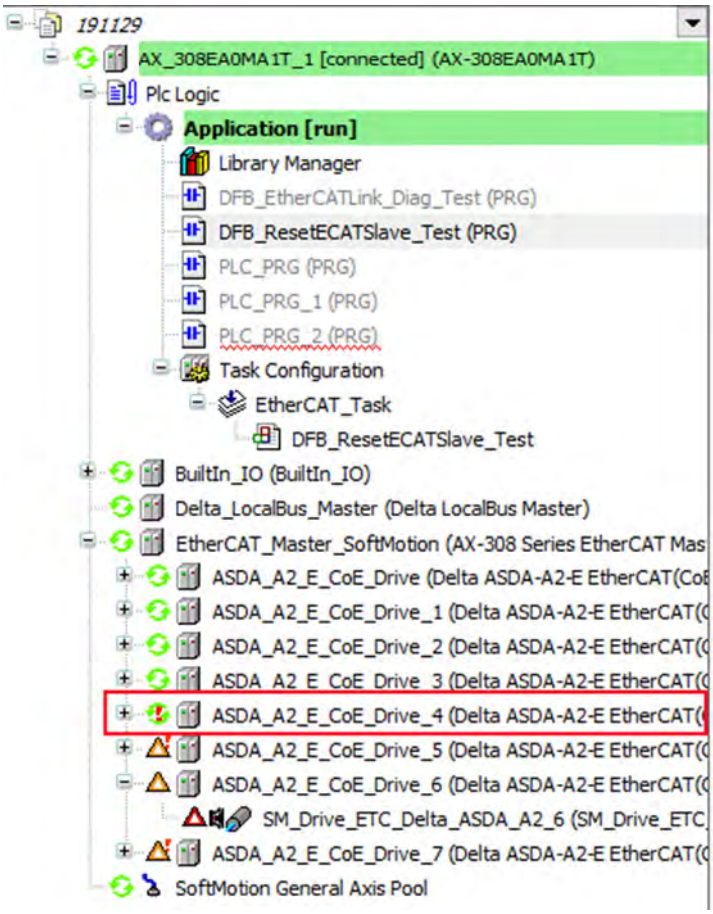
2. Remove the network connection of slave 4 and 5, and the current connection status of EtherCAT slave starting from slave 5 would show Fail in the device tree.



3. To restore the network connection of slave 4 and 5, enter 1005 to the input uiSlaveAddr and shift the input bExecute to True.



4. The network connectivity of slave 5 has been recovered after the output bDone of the FB shifting to True.



5. All the slave stations would be reset if you enter 0 to the input uiSlaveAddr.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_EtherCAT_Diag.library

4.9 Error Codes and Troubleshooting

The following table lists the error codes corresponding to the FBs and the contents of the errors:

Description	Cause of Error	Corrective Action
DFB_ECAT_Diag_MASTER_CANT_BE_FOUND	EtherCAT master cannot be found.	The EtherCAT master is not in the mapping, please make sure the configuration of EtherCAT master is correct.
DFB_ECAT_Diag_MASTER_ERROR	Diagnostics of EtherCAT master state is wrong.	Please troubleshoot the errors in EtherCAT master before execute the FB.
DFB_ECAT_Diag_SLAVE_CANT_BE_FOUND	EtherCAT slave cannot be found.	The EtherCAT master is not in the mapping, please make sure the EtherCAT master address is correct
DFB_ECAT_Diag_MASTER_RESTART_TIMEOUT	Time out occurs when restart EtherCAT master.	Please check if the timeout is too short or the internet has been lost.
DFB_ECAT_Diag_SLAVE_RESTART_TIMEOUT	Time out occurs when restart EtherCAT slave.	Please check if the timeout is too short or the internet has been lost.
DFB_ECAT_Diag_MASTER_DISABLE	EtherCAT master is disabled.	Please check whether to enable the EtherCAT master.

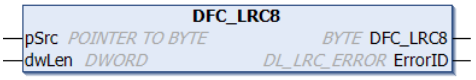
5

Chapter 5 Checksum Instructions

Table of Content

5.1 DFC_LRC8

DFC_LRC8: LRC (8-bit) checksum calculation.

FB/FC	Instruction	Graphic Expression	ST Language
FC	DFC_LRC8		DFC_LRC8(pSrc:= , dwLen:= , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
pSrc	The start address for LRC calculation.	POINTER TO BYTE	Memory address(0)
dwLen	The data length for LRC calculation.	DWORD*	(0)

***Note:** The variable type BYTE, WORD and DWORD can be used for dwLen input.

- **Output**

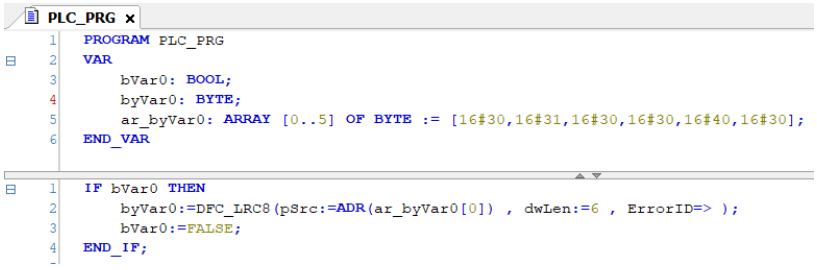
Name	Function	Data Type	Output Range(Default value)
DFC_LRC8	LRC checksum (Return type)	BYTE	(0)
ErrorID	Error codes	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

- **Function**

After executes the FC instruction, it begins to calculate LRC (8-bit) checksum, starting from the memory address input to pSrc, while the calculation scope is determined by the input dwLen.

- **Programming Example**

The example uses FC instruction (DFC_LRC8) to perform calculating the LRC (8-bit) checksum.



```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3      bVar0: BOOL;
4      byVar0: BYTE;
5      ar_byVar0: ARRAY [0..5] OF BYTE := [16#30,16#31,16#30,16#30,16#40,16#30];
6  END_VAR

1  IF bVar0 THEN
2      byVar0:=DFC_LRC8(pSrc:=ADR(ar_byVar0[0]) , dwLen:=6 , ErrorID=> );
3      bVar0:=FALSE;
4  END_IF;
```

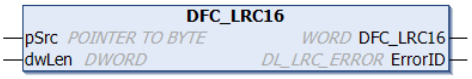
The checksum calculation scope is 6(dwLen = 6), therefore, the FC instruction(DFC_LRC8) will starts calculating checksums of six consecutive BYTE data from the memory address input to pSrc(ar_byVar0[0]) and will result in a checksum value of 16#CF.

- **Supported Products**
  - AX Series

- **Library**
  - DL_LRC.library

5.2 DFC_LRC16

DFC_LRC16: LRC (16-bit) checksum calculation.

FB/FC	Instruction	Graphic Expression	ST Language
FC	DFC_LRC16		DFC_LRC16(pSrc:= , dwLen:= , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
pSrc	The start address for LRC calculation.	POINTER TO BYTE	Memory address(0)
dwLen	The data length for LRC calculation.	DWORD*	(0)

***Note:** The variable type BYTE, WORD and DWORD can be used for dwLen input.

- **Output**

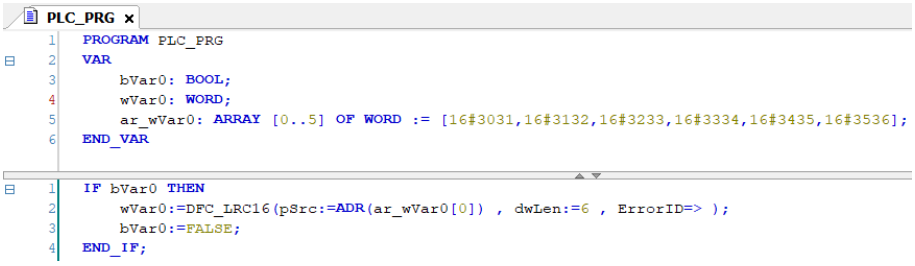
Name	Function	Data Type	Output Range(Default value)
DFC_LRC16	LRC checksum (Return type)	WORD	(0)
ErrorID	Error codes	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

- **Function**

After executes the FC instruction, it begins to calculate LRC (16-bit) checksum, starting from the memory address input to pSrc, while the calculation scope is determined by the input dwLen.

- **Programming Example**

The example uses FC instruction (DFC_LRC16) to perform calculating the LRC (16-bit) checksum.



```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3      bVar0: BOOL;
4      wVar0: WORD;
5      ar_wVar0: ARRAY [0..5] OF WORD := [16#3031,16#3132,16#3233,16#3334,16#3435,16#3536];
6  END_VAR

1  IF bVar0 THEN
2      wVar0:=DFC_LRC16(pSrc:=ADR(ar_wVar0[0]) , dwLen:=6 , ErrorID=> );
3      bVar0:=FALSE;
4  END_IF;
```

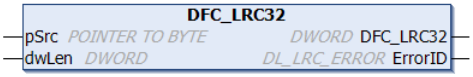
The checksum calculation scope is 6(dwLen = 6), therefore, the FC instruction(DFC_LRC16) will starts calculating checksums of six consecutive BYTE data from the memory address input to pSrc(ar_byVar0[0]) and will result in a checksum value of 16#CFCB.

- **Supported Products**
  - AX Series

- **Library**
  - DL_LRC.library

5.3 DFC_LRC32

DFC_LRC32: LRC (32-bit) checksum calculation.

FB/FC	Instruction	Graphic Expression	ST Language
FC	DFC_LRC32		DFC_LRC32(pSrc:= , dwLen:= , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
pSrc	The start address for LRC calculation.	POINTER TO BYTE	Memory address(0)
dwLen	The data length for LRC calculation.	DWORD*	(0)

***Note:** The variable type BYTE, WORD and DWORD can be used for dwLen input.

- **Output**

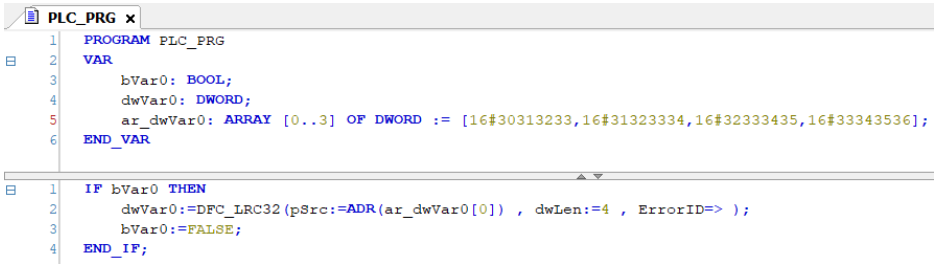
Name	Function	Data Type	Output Range(Default value)
DFC_LRC32	LRC checksum (Return type)	DWORD	(0)
ErrorID	Error codes	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

- **Function**

After executes the FC instruction, it begins to calculate LRC (32-bit) checksum, starting from the memory address input to pSrc, while the calculation scope is determined by the input dwLen.

- **Programming Example**

The example uses FC instruction (DFC_LRC32) to perform calculating the LRC (32-bit) checksum.



```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3      bVar0: BOOL;
4      dwVar0: DWORD;
5      ar_dwVar0: ARRAY [0..3] OF DWORD := [16#30313233,16#31323334,16#32333435,16#33343536];
6  END_VAR

1  IF bVar0 THEN
2      dwVar0:=DFC_LRC32(pSrc:=ADR(ar_dwVar0[0]) , dwLen:=4 , ErrorID=> );
3      bVar0:=FALSE;
4  END_IF;
```

The checksum calculation scope is 4(dwLen = 4), therefore, the FC instruction(DFC_LRC32) will starts calculating checksums of four consecutive BYTE data from the memory address input to pSrc(ar_byVar0[0]) and will result in a checksum value of 16#3935312E.

- **Supported Products**
  - AX Series

- **Library**
  - DL_LRC.library

5.4 Error Codes and Troubleshooting

Description	Cause of Error	Corrective Action
DFC_LRC_ERR_ PARAMETER	The value of dwLen is incorrect.	Make sure the dwLen value is greater than zero.

6

Chapter 6 Module Read-write Instructions

Table of Content

6.1 DFB_From

DFB_From: Read the CR data in the module.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ From		DFB_From(bExecute:= , byRemoteID:= , byLocalID:= , wCRAddr:= , iLength:= , pVal:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False (False)
byRemoteID*	The CPU or remote module ID	BYTE	0: CPU 1~16: Remote module (0)
byLocalID	Expansion module ID	BYTE	0 ~ 31
wCRAddr	The CR data positon in the module.	WORD	(0)
iLength	The CR data length	INT	1~125 (0)
pVal	The CR data to read.	POINTER TO WORD	

***Note:** Currently only support mode 0.

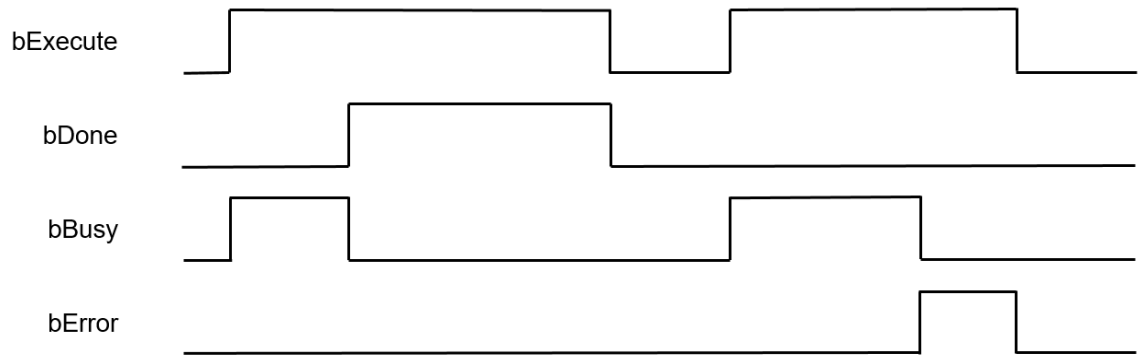
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	True when the execution of the instruction is completed.	BOOL	True/False (False)
bBusy	True when the instruction is being executed.	BOOL	True/False (False)
bError	True if an error occurs.	BOOL	True/False (False)
ErrorID	Indicates the error code if an error occurs.	DFB_AS_MODULE_API_ERROR	DFB_AS_MODULE_API_ERROR (DFB_NO_ERROR)

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When the execution of FB starts.	● When the execution of FB is completed. ● When bExecute shifts to False.
bError	● When an error occurs in the execution conditions or input values for the instruction.	● When bExecute shifts to False.
ErrorID		

- **Timing Diagram**



- **Function**

The Function block DFB_From reads the CR data in the module.

- **Programming Example**

This example uses DFB_From to read the value of CR1 in the second module on the right side of the CPU and store the value in the variable(wVar) of the controller.

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bVar0: BOOL :=TRUE;
    bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
    byRemoteID_Var,byLocalID_Var: BYTE;
    wCRAddr_Var: WORD;
    iLength_Var: INT;
    ErrorID_Var: DFB_AS_MODULE_API_ERROR
    wVar0: WORD;
    FB0: DFB_From;
END_VAR
```

```
IF bVar0 THEN
    bExecute_Var:=TRUE;
    byRemoteID_Var:=0;
    byLocalID_Var:=2;
    wCRAddr_Var:=1;
    iLength_Var:=1;
    bVar0:=FALSE;
END_IF
IF bDone_Var THEN
    bExecute_Var:=FALSE;
END_IF
FB0(
    bExecute:=bExecute_Var ,
    byRemoteID:=byRemoteID_Var ,
    byLocalID:=byLocalID_Var ,
    wCRAddr:=wCRAddr_Var ,
    iLength:=iLength_Var ,
    pVal:=ADR(wVar0) ,
    bDone=>bDone_Var ,
    bBusy=>bBusy_Var ,
    bError=>bError_Var ,
    ErrorID=>ErrorID_Var );
```


- **Supported Products**
  - AX-308E

- **Library**
  - DL_ASModuleAPI_AX3

6.2 DFB_To

DFB_To: Write a value to the CR data in the module.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ To		DFB_To(bExecute:= , byRemoteID:= , byLocalID:= , wCRAddr:= , iLength:= , pVal:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False (False)
byRemoteID*	The CPU or remote module ID	BYTE	0:CPU 1~16: Remote module (0)
byLocalID	Expansion module ID	BYTE	0 ~ 31
wCRAddr	The CR data positon in the module.	WORD	(0)
iLength	The CR data length	INT	1~125 (0)
pVal	The CR data to read.	POINTER TO WORD	

***Note:** Currently only support mode 0.

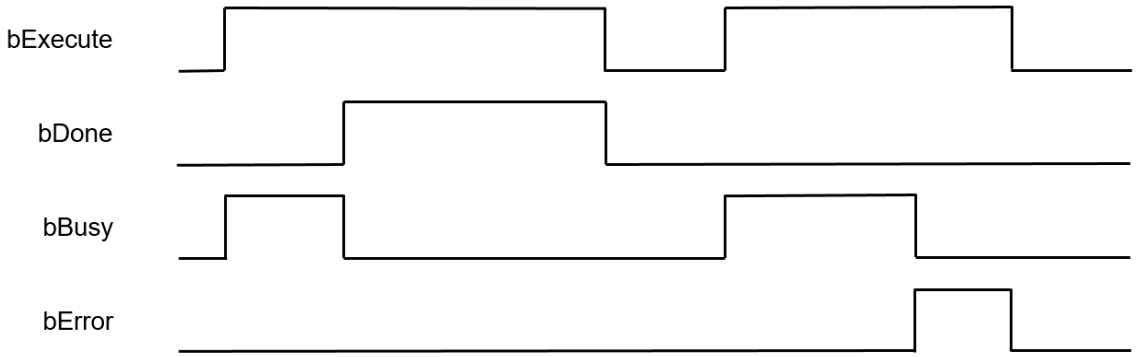
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	True when the execution of the instruction is completed.	BOOL	True/False (False)
bBusy	True when the instruction is being executed.	BOOL	True/False (False)
bError	True if an error occurs.	BOOL	True/False (False)
ErrorID	Indicates the error code if an error occurs.	DFB_AS_MODULE_API_ERROR	DFB_AS_MODULE_API_ERROR (DFB_NO_ERROR)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When the execution of FB starts.	● When the execution of FB is completed. ● When bExecute shifts to False.
bError	● When an error occurs in the execution conditions or input values for the instruction.	● When bExecute shifts to False.
ErrorID		

- **Timing Diagram**



- **Function**

The Function block DFB_To writes a value to the CR in the module.

- **Programming Example**

This example uses DFB_ To to write the value of variable(wVar) to CR1 in the second module on the right side of the CPU.

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bVar0: BOOL :=TRUE;
    bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL,
    byRemoteID_Var,byLocalID_Var: BYTE;
    wCRAddr_Var: WORD;
    iLength_Var: INT;
    ErrorID_Var: DFB_AS_MODULE_API_ERROR
    wVar0: WORD :=2;
    FB0: DFB_To;
END_VAR
```

```
IF bVar0 THEN
    bExecute_Var:=TRUE;
    byRemoteID_Var:=0;
    byLocalID_Var:=2;
    wCRAddr_Var:=1;
    iLength_Var:=1;
    bVar0:=FALSE;
END_IF
IF bDone_Var THEN
    bExecute_Var:=FALSE;
END_IF
FB0(
    bExecute:=bExecute_Var ,
    byRemoteID:=byRemoteID_Var ,
    byLocalID:=byLocalID_Var ,
    wCRAddr:=wCRAddr_Var ,
    iLength:=iLength_Var ,
    pVal:=ADR(wVar0) ,
    bDone=>bDone_Var ,
    bBusy=>bBusy_Var ,
    bError=>bError_Var ,
    ErrorID=>ErrorID_Var );
```

- **Supported Products**
  - AX-308E

- **Library**
  - DL_ASModuleAPI_AX3

6.3 Error Codes and Troubleshooting

Description	Cause of Error	Corrective Action
DFB_FROM_ERR_PARAMETER	Wrong input parameters.	Please check if the input parameter is correct.
DFB_FROM_ERR_COMMUNICATION	CAN bus communication error.	Please check on the error records.
DFB_FROM_ERR_CRADDR	Wrong CR address	Please check if the CR address is correct.
DFB_TO_ERR_PARAMETER	Wrong input parameters	Please check if the input parameter is correct.
DFB_TO_ERR_COMMUNICATION	CAN bus communication error.	Please check on the error records.
DFB_TO_ERR_CRADDR	Wrong CR address	Please check if the CR address is correct.

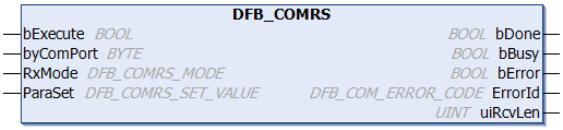
7

Chapter 7 Modbus Communication Instructions

Table of Content

7.1 DFB_COMRS

DFB_COMRS: Instruction to send and receive communication data via COM port.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_COMRS		DFB_COMRS (bExecute:= , byComPort:= , RxMode:= , ParaSet:= , bDone=> , bBusy=> , bError=> , ErrorId=> , uiRcvLen=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False(False)
byComPort*	COM port number	BYTE	(0xFF)
RxMode	Receiving mode	DFB_COMRS_MODE	(NO_RECEIVING)
ParaSet	COM port parameters	DFB_COMRS_SET_VALUE	(COMRS_SET_VALUE)

***Note:** You need to configure the settings based on the definitions of COM port numbers varied from model to model.

- **DFB_COMRS_MODE**

Name	Description
NO_RECEIVING	Not receiving data mode: After the data is sent, the receiving task is completed. Then a completion flag is set to True.
DISCONTINUOUS_TIME	Discontinuous time mode: When the time interval between each data receiving is greater than the specified duration, the receiving task is completed. Then a completion flag is set to True. The discontinuous time for receiving data can be configured via ParaSet.uiDiscontinuousTime.(*1)
SPECIFIC_END_CHAR	Specific end character mode: The data received ends with a specific character. Then a completion flag is set to True. The end character and the length can be configured via ParaSet.pSpecificEndChar and ParaSet.byEndCharAmt. (*1、*2)

Name	Description
SPECIFIC_START_CHAR_ AND_DISCONTINUOUS_ TIME	Specific start character and discontinuous time mode: The data received starts with a specific character and the time interval between each data receiving is greater than the specified duration, the receiving task is completed. The start character and the length can be configured via ParaSet.pSpecificStartChar and ParaSet.byStartCharAmt while the discontinuous time can be set via ParaSet.uiDiscontinuousTime. (*1、*2)
SPECIFIC_START_CHAR_ AND_SPECIFIC_END_ CHAR	Specific start character and end character mode: The data received starts with a specific character, and ends with a specific character. The start character and the length can be configured via ParaSet.pSpecificStartChar 與 ParaSet.byStartCharAmt while the end character and the length can be configured via ParaSet.pSpecificEndChar and ParaSet.byEndCharAmt. (*1、*2)
SPECIFIC_LENGTH	Specific data length mode: A specific quantity of data is received and the receiving task is completed. The data length can be specified via ParaSet.uiSetVarue.

***Note:**

*1: When the received data length reaches the size defined in uiReadBufSize, the receiving of data is completed.

*2: The data length includes both start and end characters.

■ **COMRS_SET_VALUE**

Name	Function	Data Type	Setting Value (Default value)
uiWriteLen	The length of data to be sent. (Unit: Byte)	UINT	0 ~ 1000(0)
pWriteBuf	The memory address of data to be sent.	POINTER TO BYTE	--
pReadBuf	The memory address of data to be stored.	POINTER TO BYTE	--
uiReadBufSize	The memory size of data to be stored. (Unit: Byte)	UINT	1 ~ 1,000(100)
uiDiscontinuousTime	Setting values for discontinuous time. (Unit: ms)	UINT	2 ~ 3,000(2)
byStartCharAmt	Size of the start character	BYTE (Unit: Byte)	1 ~ 255(1)
pSpecificStartChar	Memory address of the start character	POINTER TO BYTE	Memory address (0)
byEndCharAmt	Size of the end character	BYTE (Unit: Byte)	1 ~ 255(1)
pSpecificEndChar	Memory address of the end character	POINTER TO BYTE	Memory address (0)
uiSpecificRxLen	Specified receiving length	UINT (Unit: Byte)	1 ~ 1000(1)

Name	Function	Data Type	Setting Value (Default value)
tTimeout	Communication timeout	TIME	T#0ms ~ T#49d17h2m47s295 ms(T#100ms) T#0ms: No timeout

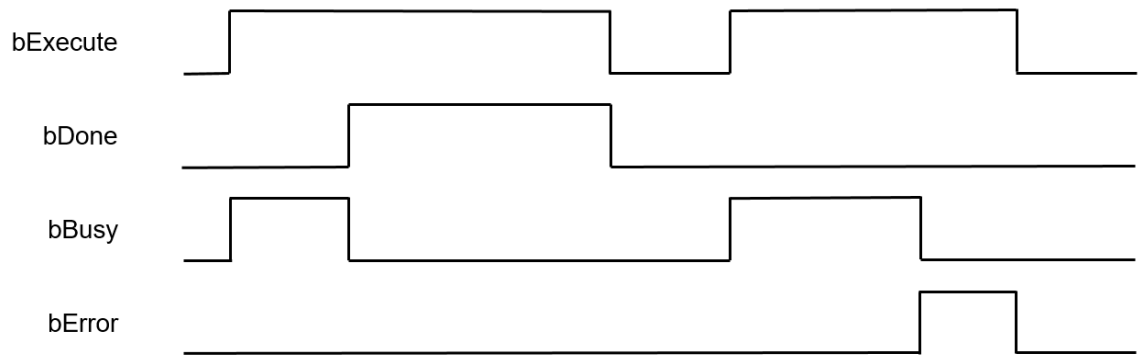
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is being executed.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_COM_ERROR_ CODE	(DFB_UNDEFINED)
uiRcvLen	The length of received data.	UINT (Unit: Byte)	(0)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When the execution of FB starts.	● When the execution of FB is completed. ● bExecute shifts to False and the execution of FB is completed.
bError	● When an error occurs in the execution conditions or input values for the instruction.	● When bExecute shifts to False.
ErrorID		

- **Timing Diagram**



- **Function**

The FB instruction (DFB_COMRS) is used for sending communication data. You must finish the configuration of COM port of CPU and add Delta_Modbus_Master_COM_Port device before using this instruction. (for more details, please refer to chapter 9.2 “Serial Port Communication” in AX-3 Series Operational Manual.)

- **Programming Example**

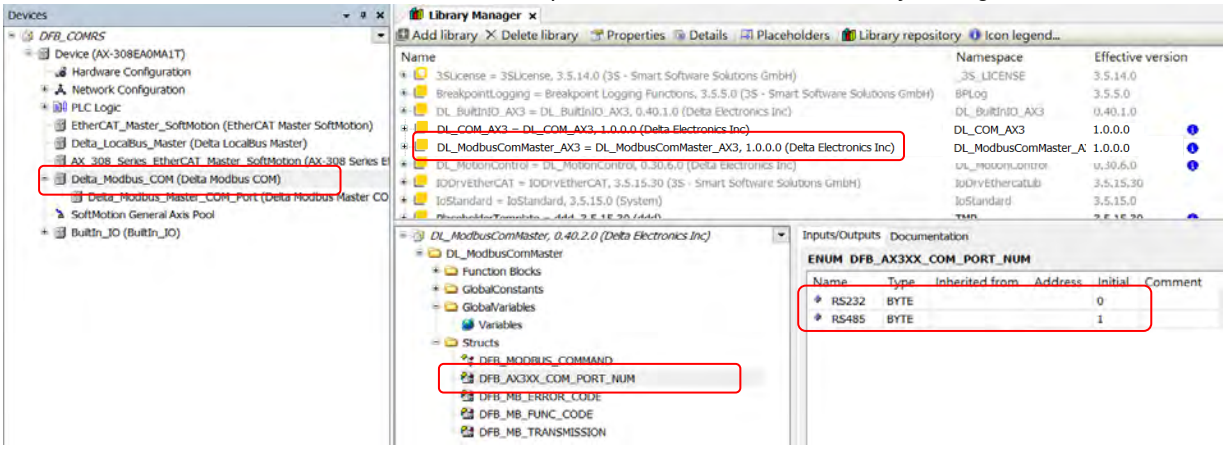
This example used DFB_COMRS to send COM communication data.

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bVar0: BOOL:=TRUE;
    bExecute_Var, bDone_Var,bBusy_Var,bError_Var: BOOL;
    FB0: DFB_COMRS;
    RxMode_Var: DL_COM_AX3.DFB_COMRS_MODE;
    ParaSet_Var: DL_COM_AX3.DFB_COMRS_SET_VALUE;
    ErrorID_Var: DL_COM_AX3.DFB_COM_ERROR_CODE;
    uiRcvLen_Var: UINT;
    pSpecificStartChar_Var: BYTE :=16#3A;
    byStartCharAmt_Var: BYTE :=1;
    pSpecificEndChar_Var: ARRAY[0..1] OF BYTE:=[16#0D,16#0A];
    byEndCharAmt_Var: BYTE :=2;
    ar_byVar0: ARRAY [0..13] OF BYTE;
    ar_byVar1: ARRAY [0..16] OF BYTE:=[16#3A,16#30,16#32,16#30,16#31,16#30,
                                       16#35,16#30,16#30,16#30,16#30,16#30,
                                       16#31,16#46,16#37,16#0D,16#0A];
    tTimeout_Var: TIME :=T#500ms;
END_VAR
```

```
IF bVar0 THEN
    bVar0:=FALSE;
    bExecute_Var:=TRUE;
    RxMode_Var:=DL_COM_AX3.DFB_COMRS_MODE.SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR;
    ParaSet_Var.pSpecificStartChar:=ADR(pSpecificStartChar_Var);
    ParaSet_Var.byStartCharAmt:=byStartCharAmt_Var;
    ParaSet_Var.pSpecificEndChar:=ADR(pSpecificEndChar_Var);
    ParaSet_Var.byEndCharAmt:=byEndCharAmt_Var;
    ParaSet_Var.pReadBuf:=ADR(ar_byVar0);
    ParaSet_Var.pWriteBuf:=ADR(ar_byVar1);
    ParaSet_Var.uiWriteLen:=SIZEOF(ar_byVar1);
    ParaSet_Var.tTimeout:=tTimeout_Var;
END_IF
IF bDone_Var THEN
    bExecute_Var:=FALSE;
END_IF
FB0(
bExecute:=bExecute_Var ,
byComPort:=DL_ModbusComMaster.DFB_AX3XX_COM_PORT_NUM.RS485 ,
RxMode:=RxMode_Var ,
ParaSet:=ParaSet_Var ,
bDone=>bDone_Var ,
bBusy=>bBusy_Var ,
bError=>bError_Var ,
ErrorId=>ErrorID_Var ,
uiRcvLen=>uiRcvLen_Var );
```

For AX-3 series controller, the definition of COM port name can be found in Library Manager as shown below.

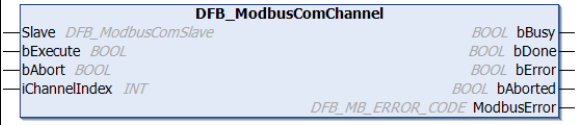


- **Supported Products**
  - AX Series(Without supporting AX-8)

- **Library**
  - DL_COM_AX3.library

7.2 DFB_ModbusComChannel

DFB_ModbusComChannel: Modbus Slave COM Port Channel control instruction.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ ModbusComChannel		DFB_ModbusComChannel(Slave:= , bExecute:= , bAbort:= , iChannelIndex:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusError=>);

- **In/ Outs**

Name	Function	Data Type	Setting Value (Default value)
Slave	Delta Modbus slave device	DFB_ModbusComSlave	--

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False(False)
bAbort	No function	BOOL	--
iChannelIndex	Channel index	INT	0 ~ 9 (0)

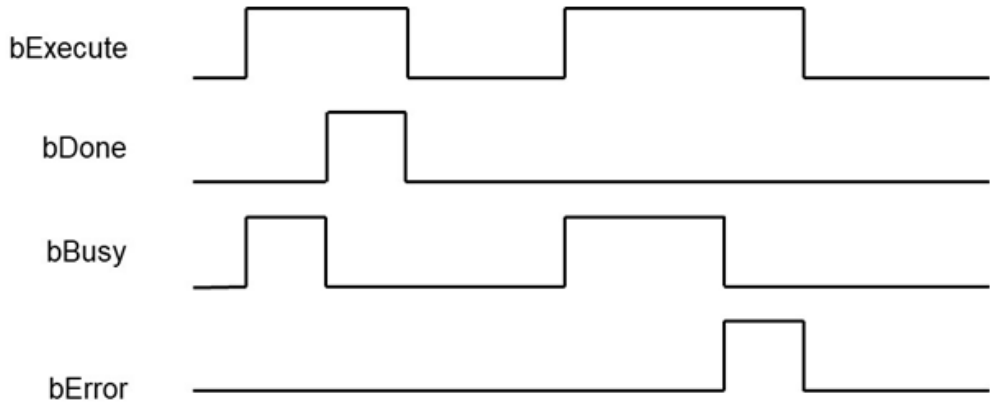
- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False(False)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
bAborted	No function	BOOL	--
ModbusError	Error code	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (UNDEFINED)

■ **Output Updating Timing**

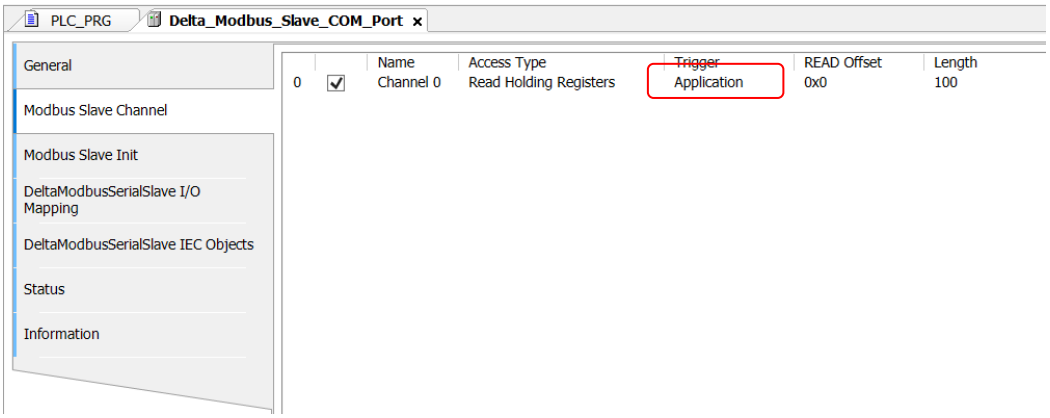
Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When the execution of FB starts.	● When the execution of FB is completed. ● bExecute shifts to False and the execution of FB is completed.
bError	● When an error occurs in the execution conditions or input values for the instruction.	● When bExecute shifts to False.
ModbusError		

● **Timing Diagram**



● **Function**

When the trigger mode of the Modbus slave channel is set to Application, the Modbus request action can be triggered by DFB_ModbusComChannel.

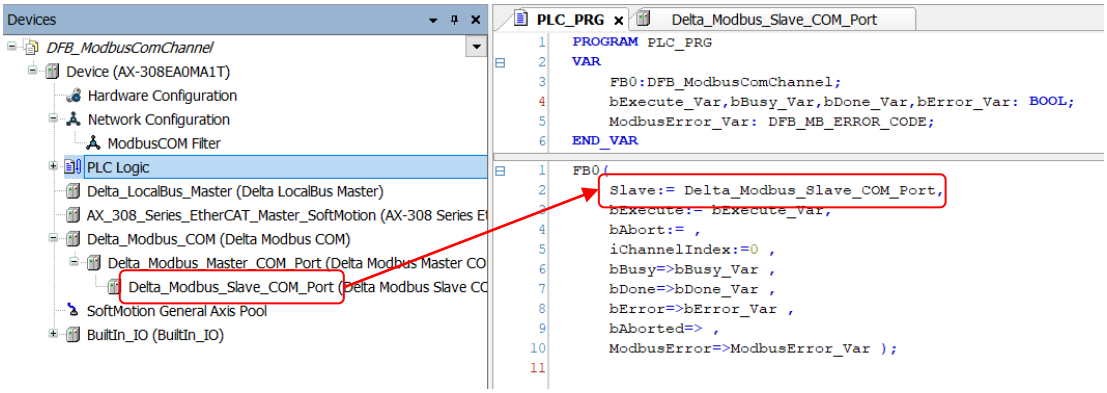


Note 1: For more details of Modbus slave COM port configuration, you can refer to chapter 9.2 “Serial Communication” in AX-3 Series Operational Manual.

Note 2: While using, the channel must be set to “Enable”.

- **Programming Example**

This example uses DFB_ModbusComChannel to trigger data exchange with Modbus COM port communication.



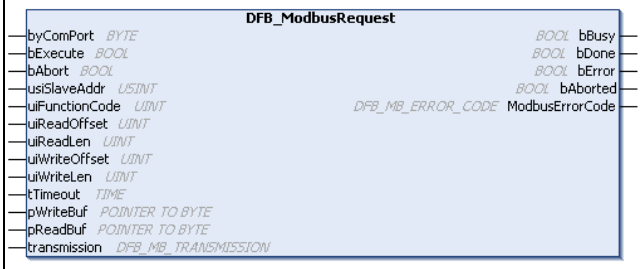
***Note:** The input of Slave would be the name of Modbus slave device.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_ModbusComMaster_AX3.library

7.3 DFB_ModbusRequest

DFB_ModbusRequest: Modbus communication commands.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ ModbusRequest		DFB_ModbusRequest(byComPort:= , bExecute:= , bAbort:= , usiSlaveAddr:= , uiFunctionCode:= , uiReadOffset:= , uiReadLen:= , uiWriteOffset:= , uiWriteLen:= , tTimeout:= , pWriteBuf:= , pReadBuf:= , transmission:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
byComPort*1	COM port number	BYTE	(0xFF)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False(False)
bAbort	No function	BOOL	--
usiSlaveAddr	Slave station number	USINT	1~247
uiFunctionCode	Modbus function code	DFB_MB_FUNC_CODE	Supported function codes: 0x01: Read Coils 0x02: Read Discrete Inputs 0x03: Read Holding Registers 0x04: Read Input Registers 0x05: Write Single Coil 0x06: Write Single Register 0x0F: Write Multiple Coils 0x10: Write Multiple Registers 0x17: Read/Write Multiple Registers (0x03)
uiReadOffset	The start address of memory to be read.	UINT	0 ~ 65535 (0)
uiReadLen	The data length of the memory to be read.	UINT	Coil: 1 ~ 1920 Register: 1 ~ 120 (1)
uiWriteOffset	The start address of memory to be written.	UINT	0 ~ 65535(0)

Name	Function	Data Type	Setting Value (Default value)
uiWriteLen	The data length of the memory to be written	UINT	Coil: 1 ~ 1920 Register: 1 ~ 120 (1)
tTimeout*2	Communication timeout	TIME	T#0ms ~ T#49d17h2m47s295ms 0: No timeout (T#100ms)
pWriteBuf	The memory address of data to be sent.	POINTER TO BYTE	--
pReadBuf	The memory address of data to be stored.	POINTER TO BYTE	--
Transmission*3	Transmission mode	DFB_MB_ TRANSMISSION	0: ASCII 1: RTU (ASCII)

***Note:**

1. You need to configure the settings based on the definitions of COM port numbers varied from model to model.
2. The timeout should be greater than the Cycle time set in mdbus Task.
3. When the transmission mode is set to RTU, the data bit of Modbus COM port must be set to 8.

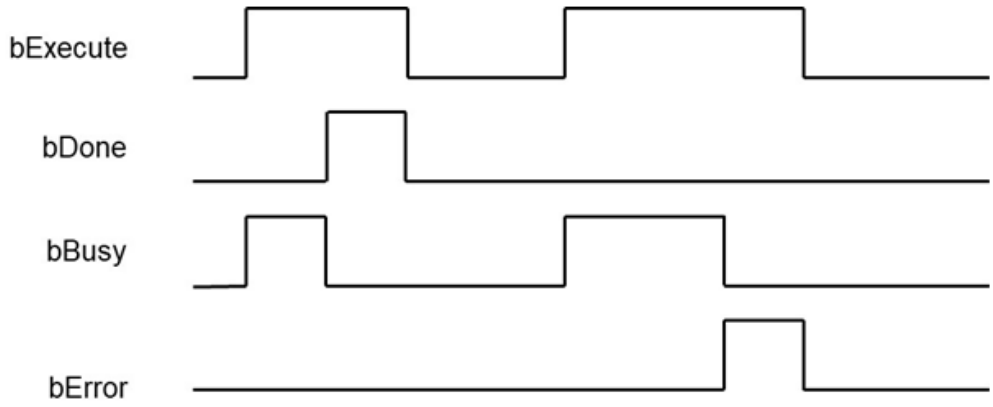
- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False(False)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
bAborted	No function	BOOL	--
ModbusError	Error code	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (DFB_UNDEFINED)

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When the execution of FB starts.	● When the execution of FB is completed. ● bExecute shifts to False and the execution of FB is completed.
bError	● When an error occurs in the execution conditions or input values for the instruction.	● When bExecute shifts to False.
ModbusError		

- **Timing Diagram**



- **Function**

The FB instruction (DFB_ModbusRequest) is used for sending Modbus communication data. You must finish the configuration of COM port of CPU and add Delta_Modbus_Master_COM_Port device before using this instruction. (For more details, please refer to chapter 9.2 “Serial Port Communication” in AX-3 Series Operational Manual.)

- **Programming Example**

This example uses DFB_ModbusRequest to send Modbus commands for reading a 10-word long data(Holding Registers) in the slave station (Slave address = 2), which the start address is 0x0000.

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bExecute_Var,bBusy_Var,bDone_Var,bError_Var: BOOL;
    usiSlaveAddr_Var: USINT :=2 ;
    ar_wVar0: ARRAY[0..200]OF WORD;
    FB0: DFB_ModbusRequest;
    ModbusErrorCode_Var: DFB_MB_ERROR_CODE;
END_VAR
```

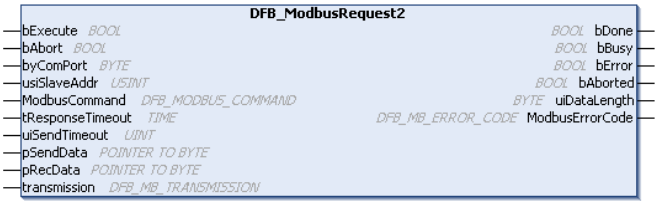
```
FB0(
    byComPort:=DL_ModbusComMaster_AX3.DFB_AX3_COM_PORT_NUM.RS485 ,
    bExecute:=bExecute_Var ,
    bAbort:= ,
    usiSlaveAddr:=usiSlaveAddr_Var ,
    uiFunctionCode:=DFB_MB_FUNC_CODE.READ_HOLDING_REGISTERS ,
    uiReadOffset:=0 ,
    uiReadLen:=100 ,
    uiWriteOffset:= ,
    uiWriteLen:= ,
    tTimeout:=T#500MS ,
    pWriteBuf:= ,
    pReadBuf:=ADR(ar_wVar0) ,
    transmission:= ,
    bBusy=>bBusy_Var ,
    bDone=>bDone_Var ,
    bError=>bError_Var ,
    bAborted=> ,
    ModbusErrorCode=>ModbusErrorCode_Var );
```

- **Supported Products**
  - AX-308E

- **Library**
  - DL_ModbusComMaster_AX3.library

7.4 DFB_ModbusRequest2

DFB_ModbusRequest2: Modbus communication commands.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ ModbusRequest2		DFB_ModbusRequest2(bExecute:= , bAbort:= , byComPort:= , usiSlaveAddr:= , ModbusCommand:= , tResponseTimeout:= , uiSendTimeout:= , pSendData:= , pRecvData:= , transmission:= , bDone=> , bBusy=> , bError=> , bAborted=> , uiDataLength=> , ModbusErrorCode=>);;

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False(False)
bAbort	No function	BOOL	--
byComPort*1	COM port number	BYTE	(0xFF)
usiSlaveAddr	Slave station number	USINT	1~247
ModbusCommand	Modbus parameter setting	DFB_MODBUS_ COMMAND	--
tResponseTimeout*2	Communication timeout	TIME	T#0ms ~ T#49d17h2m47s295ms 0: No timeout (T#100ms)
uiSendTimeout	No function	UINT	(0)
pSendData	The memory address of data to be sent.	POINTER TO BYTE	--
pRecvData	The memory address of received data to be stored.	POINTER TO BYTE	--
Transmission*3	Transmission mode	DFB_MB_ TRANSMISSION	0: ASCII 1: RTU (ASCII)

***Note:**

1. You need to configure the settings based on the definitions of COM port numbers varied from model to model.
2. The timeout should be greater than the Cycle time set in mdbus Task.
3. When the transmission mode is set to RTU, the data bit of Modbus COM port must be set to 8.

- DFB_MODBUS_COMMAND

Name	Function	Data Type	Output Range(Default value)
uiFunctionCode	Modbus function code	DFB_MB_FUNC_ CODE	Supported function code: 0x01: Read Coils 0x02: Read Discrete Inputs 0x03: Read Holding Registers 0x04: Read Input Registers 0x05: Write Single Coil 0x06: Write Single Register 0x0F: Write Multiple Coils 0x10: Write Multiple Registers 0x17: Read/Write Multiple Registers (0x03)
uiReadOffset	The start address of memory to be read.	UINT	0 ~ 65535 (0)
uiReadLen	The data length of the memory to be read.	UINT	Coil: 1 ~ 1920 Register: 1 ~ 120 (1)
uiWriteOffset	The start address of memory to be written.	UINT	0 ~ 65535 (0)
uiWriteLen	The data length of the memory to be written	UINT	Coil: 1 ~ 1920 Register: 1 ~ 120 (1)

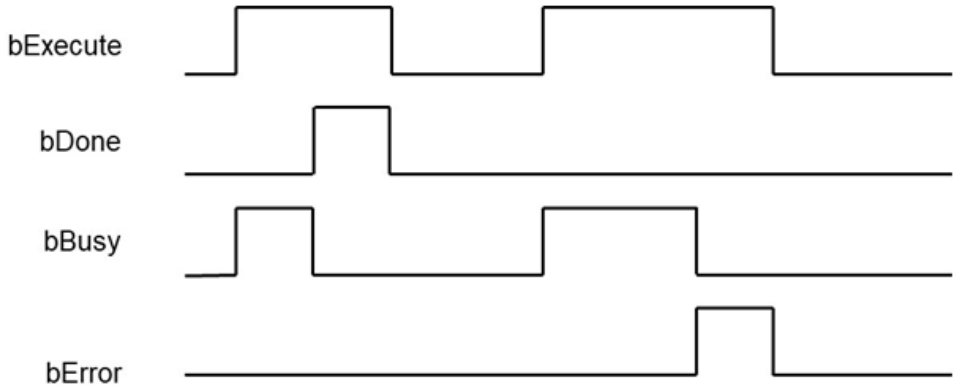
- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False(False)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
bAborted	No function	BOOL	--
uiDataLength	The received data length	BYTE (Unit: BYTE)	(0)
ModbusError	Error code	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (DFB_UNDEFINED)

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When the execution of FB starts.	● When the execution of FB is completed. ● bExecute shifts to False.
bError	● When an error occurs in the execution conditions or input values for the instruction.	● When bExecute shifts to False.
ModbusError		

- **Timing Diagram**



- **Function**

The FB instruction (DFB_ModbusRequest2) is used for sending Modbus communication data. You must finish the configuration of COM port of CPU and add Delta_Modbus_Master_COM_Port device before using this instruction. (For more details, please refer to chapter 9.2 “Serial Port Communication” in AX-3 Series Operational Manual.)

- **Programming Example**

This example uses DFB_ModbusRequest2 to send Modbus commands for reading a 100-word long data(Holding Registers) in the slave station (Slave address = 2), which the start address is 0x0000.

PLC_PRG

```
VAR
    bVar0: BOOL:=TRUE;
    bExecute_Var,bBusy_Var,bDone_Var,bError_Var: BOOL;
    usiSlaveAddr_Var: USINT :=2 ;
    ar_wVar0: ARRAY[0..200]OF WORD;
    FB0: DFB_ModbusRequest2;
    ModbusErrorCode_Var: DFB_MB_ERROR_CODE;
    ModbusCommand_Var: DFB_MODBUS_COMMAND;
    uiDataLength_Var: UINT;
END_VAR
```

```
IF bVar0 THEN
    bVar0:=FALSE;
    ModbusCommand_Var.uiFunctionCode:=3;
    ModbusCommand_Var.uiReadLen:=100;
    ModbusCommand_Var.uiReadOffset:=0;
END_IF
FB0(
    bExecute:=bExecute_Var ,
    bAbort:= ,
    byComPort:=DL_ModbusComMaster.DFB_AX3XX_COM_PORT_NUM.RS485 ,
    usiSlaveAddr:=usiSlaveAddr_Var ,
    ModbusCommand:=ModbusCommand_Var ,
    tResponseTimeout:=T#500MS ,
    uiSendTimeout:= ,
    pSendData:= ,
    pRecvData:=ADR(ar_wVar0) ,
    transmission:=DFB_MB_TRANSMISSION.ASCII ,
    bDone=>bDone_Var ,
    bBusy=>bBusy_Var ,
    bError=>bError_Var ,
    bAborted=> ,
    uiDataLength=>uiDataLength_Var ,
    ModbusErrorCode=>ModbusErrorCode_Var );
```

- **Supported Products**
  - AX-308E

- **Library**
  - DL_ModbusComMaster_AX3.library

7.5 Error codes and Troubleshooting

- DFB_COM_ERROR_CODE

Description	Cause of Error	Corrective Action
DFB_NO_ERROR	No errors.	--
DFB_RESPONSE_TIMEOUT	Slave response timeout	● Please check whether the setting for timeout is appropriate or not. ● Please check on the correctness of communication wiring.
DFB_REQUEST_FAILED_ TO_SEND	COM Port errors	Please contact us directly.
DFB_INVALID_COMPORT	COM port setting errors	Please check on the correctness of the COM port settings.
DFB_INVALID_BUFFER	Invalid memory address for sending and receiving data.	Please check if the below parameter settings are correct. ● ParaSet.pWriteBuf ● ParaSet.pReadBuf
DFB_INVALID_LENGTH	Invalid data length setting	Please check if the below parameter settings are correct. ● ParaSet.uiReadLen ● ParaSet.uiReadBufSize ● ParaSet.uiWriteLen
DFB_NO_MASTER_ CONFIG	Delta_Modbus_Master_COM_Port device does not exist.	Please check if Delta_Modbus_Master_COM_Port device have been added to the device tree.
DFB_MEMORY_NOT_ ENOUGH	Not enough system memory	Please check if the program size exceeds the allowable limit.
DFB_INVALID_MODE	Invalid receiving mode set in DFB_COMRS.	Please check if the RxMode setting is correct.
DFB_INVALID_SETTING	Invalid parameter setting	Please check if the below parameter settings are correct. ● ParaSet.tTimeout ● ParaSet.uiDiscontinuousTime ● ParaSet.byEndCharAmt ● ParaSet.byStartCharAmt ● ParaSet.uiSpecificRxLen
DFB_INVALID_CHAR_ BUFFER	Invalid memory address of characters.	Please check if the below parameter settings are correct. ● ParaSet.pSpecificStartChar ● ParaSet.pSpecificEndChar
DFB_UNDEFINED	Undefined or has not yet been executed.	Wait for the execution of FB instruction being completed.

- DFB_MB_ERROR_CODE

Description	Cause of Error	Corrective Action
DFB_NO_ERR	No errors	-
DFB_ILLEGAL_FUNCTION	Unsupported function code	Please check on the correctness of the function code you're using.
DFB_ILLEGAL_DATA_ ADDRESS	Illegal memory address to write and read.	Please check on the correctness of memory address you intend to write and read.
DFB_ILLEGAL_DATA_ VALUE	Illegal data values responded by slave.	Please check if the wires function normally as well as the proper wiring.
DFB_RESPONSE_ TIMEOUT	No response from slave in time.	● Please check if the duration set for the timeout is less than the responded time of slave. ● Please check on the correctness of the wiring.

Description	Cause of Error	Corrective Action
DFB_RESPONSE_CRC_ ERROR	Illegal data values responded by slave. (Invalid check code)	Please check on the correctness of data format responded by slave.
DFB_RESPONSE_ WRONG_SLAVE	Illegal data values responded by slave. (Invalid station number)	Please check on the correctness of data format responded by slave.
DFB_RESPONSE_ WRONG_ FUNCTIONCODE	Illegal data values responded by slave. (Invalid function code)	Please check on the correctness of data format responded by slave.
DFB_REQUEST_FAILED_TO _SEND	Failed to send data.	Please contact the vendor directly.
DFB_RESPONSE_ INVALID_PROTOCOL	Illegal data values responded by slave. (Non-standard Modbus format)	Please check on the correctness of data format responded by slave.
DFB_RESPONSE_ INVALID_HEAD	Illegal data values responded by slave. (Invalid data length)	Please check on the correctness of data format responded by slave.
DFB_INVALID_CHANNEL_IN DEX	Invalid index of the slave channel	Please check if the index of slave channel is correct.
DFB_CHANNEL_SETTING_N OT_SUPPORT	The trigger mode of slave channel is not set to “Application”.	Make sure the trigger mode has been set to “Application”.
DFB_INVALID_COMPORT	Invalid COM port number of the controller.	Please check if the COM port number is correct.
DFB_INVALID_BUFFER	Invalid memory address setting to send and receive data.	Please check if the below parameter settings are correct. ModbusRequest: ● pWriteBuf ● pReadBuf ModbusRequest2: ● ModbusCommand.pWriteBuf ● ModbusCommand.pReadBuf
DFB_INVALID_LENGTH	Invalid data length setting.	Please check if the below parameter settings are correct. ModbusRequest: ● uiWriteLen ● uiReadLen ModbusRequest2: ● ModbusCommand.uiWriteLen ● ModbusCommand.uiReadLen
DFB_INVALID_SLAVE_ ADDRESS	Invalid slave station number.	Make sure the station number is set to be within 1~247.
DFB_INVALID_FUNCTION_C ODE	Invalid setting for uiFunctionCode.	Please check if the setting value of uiFunctionCode is correct,
DFB_NO_MDBSCOM_ CONFIG	Delta_Modbus_COM device does not exist.	Make sure that Delta_Modbus_COM device has been added to the device tree.
DFB_NO_MASTER_ CONFIG	Delta_Modbus_Master_COM_Por t device does not exist.	Make sure that Delta_Modbus_Master_COM_Port device has been added to the device tree.
DFB_MB_ERROR_CODE_ME MORY_NOT_ENOUGH	Not enough system memory.	Please check if the program size exceeds the allowable limit.
DFB_UNDEFINED	Undefined or has not yet been executed.	Wait for the execution of FB instruction being completed.

8

Chapter 8 Network Communication Instructions

Table of Content

8.1 DFB_TCP_Client

DFB_TCP_Client: TCP socket client instruction.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ TCP_Client		DFB_TCP_Client(bEnable:= , SocketInfo:= , bSend:= , bRecvRestart:= , bBusy=> , bConnected=> , bSent=> , bRcvd=> , bError=> , ErrorID=> , Status=> , uiRcvdLen=>);

■ **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block. *1*2	BOOL	True/False (False)
SocketInfo	Connection information on Server	tcpClientSocketInfo	--
bSend	Send data packets. (Rising-edge triggered)	BOOL	True/False (False)
bRecvRestart	Restart to receive data packets.*2 (Rising-edge triggered)	BOOL	True/False (False)

*1 As soon as this function block is executed, TCP connection will start to be created. Once connected, the output bConnected would be ON.

*2 After the function block is executed, it starts receiving data packets. When the data receiving is completed and stopped, bRcvd would be ON. If you shift bRecvRestart to ON, the FB will restart to receive data.

■ **tcpClientsocketInfo**

Name	Function	Data Type	Setting Value (Default value)
byIPAddr	Server's IP address	ARRAY [0..3] OF BYTE	--
uiLPort	Communication ports on local device	UINT	0: Use a random port number 0 ~ 65535 (0)
uiRPort	Communication ports on remote device	UINT	0: Illegal 1 ~ 65535 (0)

Name	Function	Data Type	Setting Value (Default value)
uiTimeout	Response timeout (Unit: ms)	UINT	0: No timeout 1 ~ 65535 (0)
uiKeepAliveTimeout	The time that the socket keeps alive. (Unit: sec)	UINT	0: No timeout 1 ~ 65535 (0)
bReconnect	Auto-reconnect function	BOOL	True/False (False) True: When connection timeout or failed, it would try to rebuilt the connection automatically. False: When connection timeout or failed, the output bError would be ON.
pSendBuf	The memory address of data to be sent	POINT TO BYTE	--
uiSendLen	The length of data to be sent (Unit: Byte)	UINT	0 ~ 8192 (0)
pRecvBuf	The memory address where the received data to be stored.	POINT TO BYTE	--
uiRecvBufSize	The memory size of received data (Unit: Byte)	UINT	0 ~ 8192 (0)
uiSetValue	The setting value of recvCondition	UINT	(0)
recvCondition	Conditions for data receiving completion	DFB_SOCK_RECV_MODE	(DFB_SOCKET_NO_RECEVING)

■ **DFB_SOCK_RECV_MODE**

Name	Description
DFB_SOCK_MODE_NO_RECEVING	No receiving data mode.
DFB_SOCK_MODE_SPECIFIC_LENGTH	Specific data length mode: A specific quantity of data is received and the receiving task is completed. The data length can be specified via uiSetValue. (Unit: Byte)
DFB_SOCK_MODE_SPECIFIC_SINGLE_CHAR	Specific end character mode: The data received ends with a specific character (1 Byte). The end character can be configured via uiSetValue. e.g.: If uiSetValue is set to 16#00000D0A, the end character would be 16#0A.(*1*2)

Name	Description
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_TWO_CHARS	Specific two end characters mode: The data received ends with the two specific characters (2 Bytes) The end character can be configured via uiSetValue e.g.: If uiSetValue is set to 16#00000D0A, the end characters would be 16#0D0A. (*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR	Specific start character and end character mode: The data received starts with a specific character, and ends with a specific character. Both the start and the end character can be configured via uiSetValue. e.g.: If uiSetValue is set to 16#00003A0A, the start character would be 16#3A and the end character is 16#0A. (*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_ANY_LENGTH	Any length mode: The receiving ends with a complete data of any length. (*1)

*1: When the length of received data reaches the limit set in pRecvLen, the receiving task would be completed.
*2: The data length includes both start and end characters.

■ **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False (False)
bConnected	TCP is connected.	BOOL	True/False (False)
bSent	Sending completed	BOOL	True/False (False)
bRcvd	Receiving completed	BOOL	True/False (False)
bError	True if an error occurs.	BOOL	True/False (False)
ErrorID	Indicates the error code if an error occurs.	DFB_SOCKET_ERROR	(DFB_SOCK_NO_ERROR)
Status	The execution status of socket.	DFB_SOCKET_STATUS	(SOCKET_CLOSED)
uiRcvLen	The length of received data.	UINT (Unit: Byte)	(0)

■ **DFB_SOCKET_STATUS**

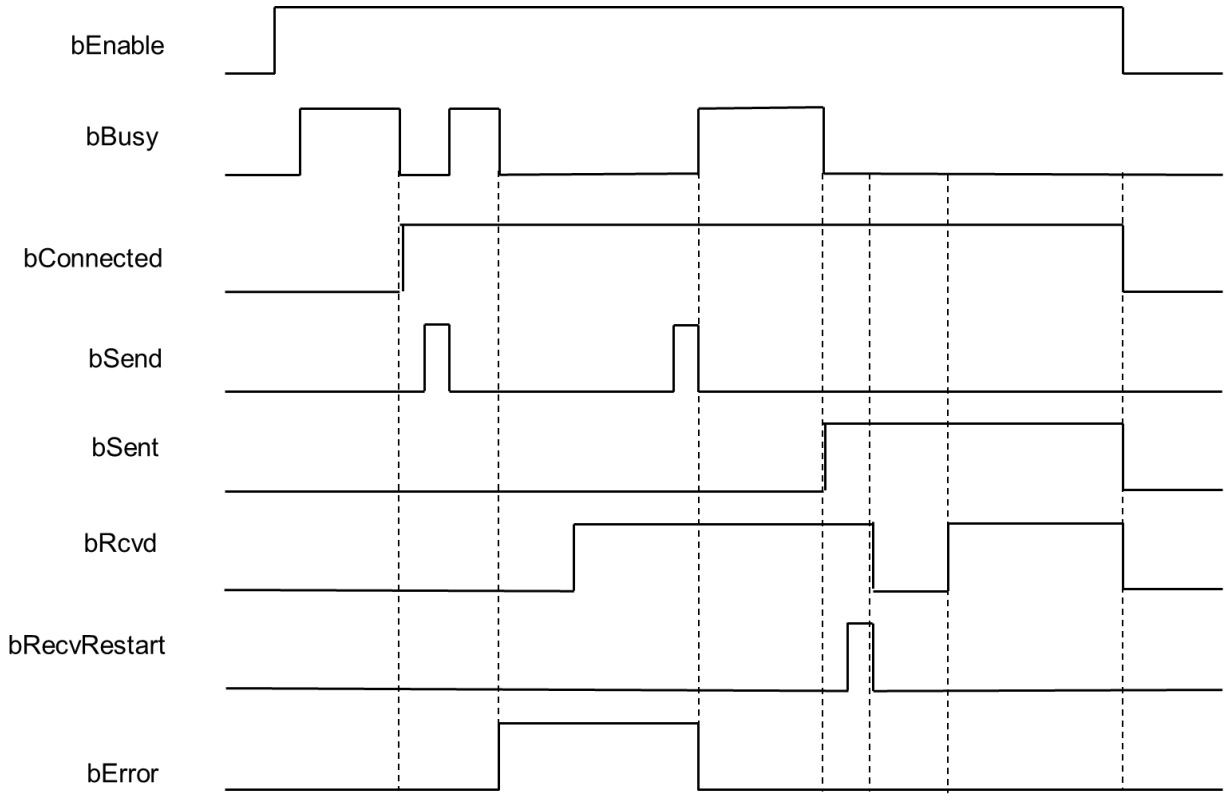
Name	Description	Applicable Protocol
SOCKET_CLOSED	SOCKET connection is closed.	TCP / UDP
SOCKET_CONNECTING	SOCKET is connecting.	TCP
SOCKET_CONNECTED	SOCKET is connected.	TCP
SOCKET_SENDING	SOCKET is sending the data packet.	TCP / UDP

Name	Description	Applicable Protocol
SOCKET_SENT	SOCKET has sent the data packet.	TCP / UDP
SOCKET_RECEIVED	SOCKET has received the data packet.	TCP / UDP
SOCKET_ERROR	SOCKET has errors.	TCP / UDP
SOCKET_ABORTED	SOCKET connection is aborted.	TCP
SOCKET_READY	SOCKET connection is ready.	UDP

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bBusy	● When bEnable shifts to True. ● When bSend shifts to True.	● When bEnable shifts to False. ● When the task is completed.
bConnected	● When the TCP connection is created.	● When bEnable shifts to False. ● When the TCP connection is aborted on the server side.
bSent	● When a data packet is sent over TCP.	● When bEnable shifts to False. ● When bSend shifts to True.
bRcvd	● When a data packet is received over TCP.	● When bEnable shifts to False. ● When bRecvRestart shifts to True.
bError	● When an error occurs during execution or the input value of the instruction is incorrect.	● When bEnable shifts to False.

■ **Timing Diagram**



■ Function

Use the FB instruction (DFB_TCP_Client) to create TCP connection so as to send or receive TCP data packets.

● Programming Example

This example use the FB instruction (DFB_TCP_Client) to create the connection with the server (IP address: 192.168.1.111 , Port: 502) and send a 1000-byte data packet while expecting a 1000-byte data packet will be sent back from the server side.

TCP_Client ×

```
PROGRAM TCP_Client
VAR
    bVar0 : BOOL := TRUE;
    FB0: DL_EthernetLib.DFB_TCP_Client;
    Server_IP_Address : ARRAY [0..3] OF BYTE := [192,168,1,111];
    bEnable_Var,bSend_Var,bRestart_Var,bBusy_Var,bConnected_Var,bSent_Var,bRcvd_Var,bError_Var : BOOL;
    RemoteInfo_Var : DL_EthernetLib.tcpClientSocketInfo;
    ErrorID_Var : DL_EthernetLib.DFB_SOCKET_ERROR;
    Status_Var : DL_EthernetLib.DFB_SOCKET_STATUS;
    uiRcvdLen_Var : UINT;
    ar_byVar0,ar_byVar1: ARRAY[0..2000] OF BYTE;
END_VAR
```

100 %

```
IF bVar0 THEN
    bVar0:=FALSE;
    RemoteInfo_Var.byIPAddr[0]:=192;
    RemoteInfo_Var.byIPAddr[1]:=168;
    RemoteInfo_Var.byIPAddr[2]:=1;
    RemoteInfo_Var.byIPAddr[3]:=111;
    RemoteInfo_Var.uiRPort:=502;
    RemoteInfo_Var.uiLPort:=2000;
    RemoteInfo_Var.uiTimeOut:=3000; //ms
    RemoteInfo_Var.uiKeepAliveTimeout:=300; //Sec
    RemoteInfo_Var.bReconnect:=TRUE;
    RemoteInfo_Var.pSendBuf:=ADR(ar_byVar0);
    RemoteInfo_Var.uiSendLen:=1000;
    RemoteInfo_Var.pRecvBuf:=ADR(ar_byVar1);
    RemoteInfo_Var.uiRecvBufSize:=2000;
    RemoteInfo_Var.uiSetValue:=1000;
    RemoteInfo_Var.recvCondition:=DL_EthernetLib.DFB_SOCK_RECV_MODE.DFB_SOCK_MODE_SPECIFIC_LENGTH;
    bEnable_Var:=TRUE;
    bSend_Var:=TRUE;
END_IF
FB0(
    bEnable:=bEnable_Var ,
    SocketInfo:=RemoteInfo_Var ,
    bSend:= bSend_Var,
    bRecvRestart:= bRestart_Var,
    bBusy=> bBusy_Var,
    bConnected=> bConnected_Var,
    bSent=> bSent_Var,
    bRcvd=> bRcvd_Var,
    bError=> bError_Var,
    ErrorID=> ErrorID_Var,
    Status=> Status_Var,
    uiRcvdLen=> uiRcvdLen_Var);
```

8

● Supported Products

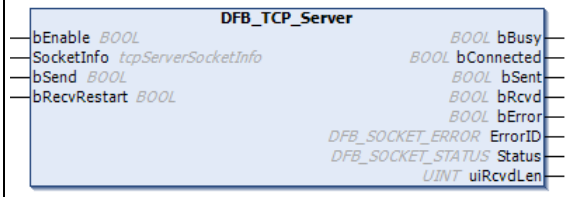
■ AX308E controller

● Library

■ DL_EthernetLib.library

8.2 DFB_TCP_Server

DFB_TCP_Server: TCP socket server instruction

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ TCP_Server		DFB_TCP_Server(bEnable:= , SocketInfo:= , bSend:= , bRecvRestart:= , bBusy=> , bConnected=> , bSent=> , bRcvd=> , bError=> , ErrorID=> , Status=> , uiRcvdLen=>);

■ **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block. *1*2	BOOL	True/False (False)
SocketInfo	Connection information on Server	tcpServerSocketInfo	--
bSend	Send data packets. (Rising-edge triggered)	BOOL	True/False (False)
bRecvRestart	Restart to receive data packets.*2 (Rising-edge triggered)	BOOL	True/False (False)

*1 As soon as this function block is executed, TCP connection will start to be created. Once connected, the output bConnected would be ON.

*2 After the function block is executed, it starts receiving data packets. When the data receiving is completed and stopped, bRcvd would be ON. If you shift bRecvRestart to ON, the FB will restart to receive data.

■ **tcpServersocketInfo**

Name	Function	Data Type	Setting Value (Default value)
byIPAddr	The IP address on Client side allowed to be connected.	ARRAY [0..3] OF BYTE	[0.0.0.0]: No limit.
uiLPort	Communication ports on local device	UINT	0: Illegal values. 1 ~ 65535 (0)
uiTimeout	Communication timeout (Unit: ms)	UINT	0: No timeout. 1 ~ 65535 (0)

Name	Function	Data Type	Setting Value (Default value)
uiKeepAliveTimeout	The time that the connection keeps alive. (Unit: sec)	UINT	0: No timeout. 1 ~ 65535 (0)
pSendBuf	The memory address of data to be sent	POINT TO BYTE	--
uiSendLen	The length of data to be sent (Unit: Byte)	UINT	0 ~ 8192 (0)
pRecvBuf	The memory address where the received data to be stored.	POINT TO BYTE	--
uiRecvBufSize	The memory size of received data (Unit: Byte)	UINT	0 ~ 8192 (0)
uiSetValue	The setting value of recvCondition	UINT	(0)
recvCondition	Conditions for data receiving completion	DFB_SOCK_RECV_MODE	(DFB_SOCKET_NO_RECEVING)

■ **DFB_SOCK_RECV_MODE**

Name	Description
DFB_SOCK_MODE_NO_RECEVING	No receiving data mode.
DFB_SOCK_MODE_SPECIFIC_LENGTH	Specific data length mode: A specific quantity of data is received and the receiving task is completed. The data length can be specified via uiSetValue. (Unit: Byte)
DFB_SOCK_MODE_SPECIFIC_SINGLE_CHAR	Specific end character mode: The data received ends with a specific character (1 Byte). The end character can be configured via uiSetValue. e.g.: If uiSetValue is set to 16#00000D0A, the end character would be 16#0A.(*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_TWO_CHARS	Specific two end characters mode: The data received ends with the two specific characters (2 Bytes) The end character can be configured via uiSetValue e.g.: If uiSetValue is set to 16#00000D0A, the end characters would be 16#0D0A. (*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR	Specific start character and end character mode: The data received starts with a specific character, and ends with a specific character. Both the start and the end character can be configured via uiSetValue. e.g.: If uiSetValue is set to 16#00003A0A, the start character would be 16#3A and

Name	Description
	the end character is 16#0A. (*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_ANY_LENGTH	Any length mode: The receiving ends with a complete data of any length. (*1)

*1: When the length of received data reaches the limit set in pRecvLen, the receiving task would be completed.
*2: The data length includes both start and end characters.

■ **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False (False)
bConnected	TCP is connected.	BOOL	True/False (False)
bSent	Sending completed	BOOL	True/False (False)
bRcvd	Receiving completed	BOOL	True/False (False)
bError	True if an error occurs.	BOOL	True/False (False)
ErrorID	Indicates the error code if an error occurs.	DFB_SOCKET_ERROR	(DFB_SOCK_NO_ERROR)
Status	The execution status of socket.	DFB_SOCKET_STATUS	(SOCKET_CLOSED)
uiRcvLen	The length of received data.	UINT (Unit: Byte)	(0)

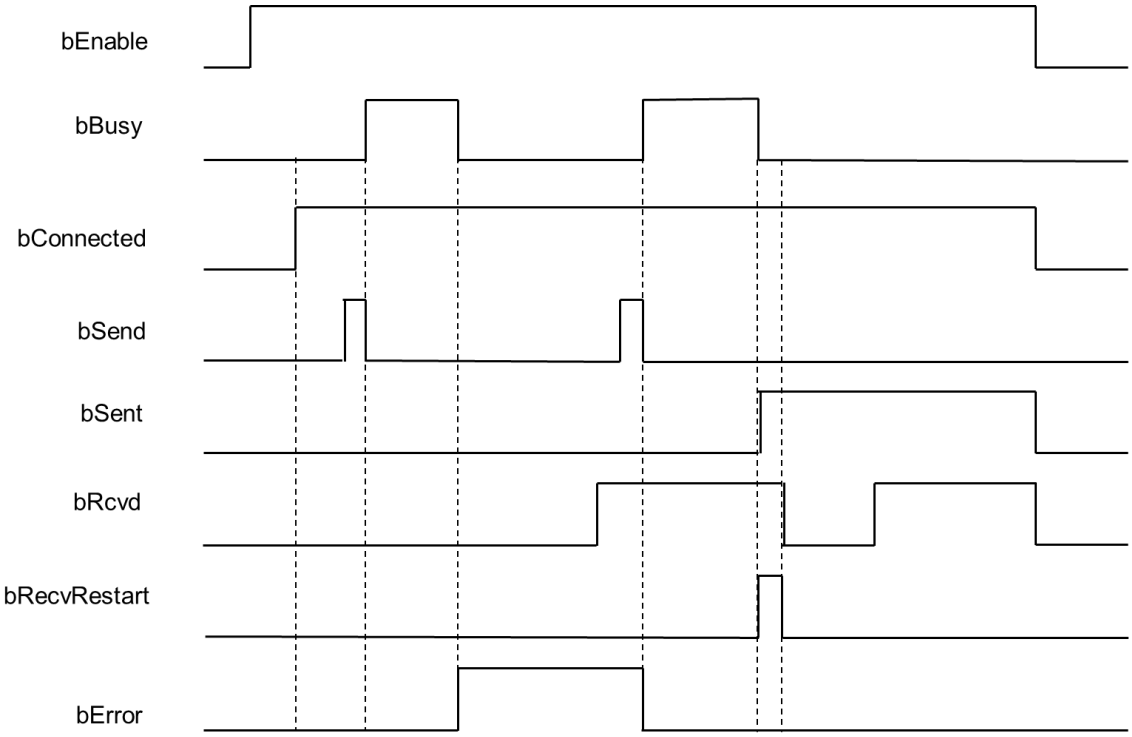
■ **DFB_SOCKET_STATUS**

Name	Description	Applicable Protocol
SOCKET_CLOSED	SOCKET connection is closed.	TCP / UDP
SOCKET_CONNECTING	SOCKET is connecting.	TCP
SOCKET_CONNECTED	SOCKET is connected.	TCP
SOCKET_SENDING	SOCKET is sending the data packet.	TCP / UDP
SOCKET_SENT	SOCKET has sent the data packet.	TCP / UDP
SOCKET_RECEIVED	SOCKET has received the data packet.	TCP / UDP
SOCKET_ERROR	SOCKET has errors.	TCP / UDP
SOCKET_ABORTED	SOCKET connection is aborted.	TCP
SOCKET_READY	SOCKET connection is ready.	UDP

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bBusy	● When bEnable shifts to True. ● When bSend shifts to True.	● When bEnable shifts to False. ● When the task is completed.
bConnected	● When the TCP connection is created.	● When bEnable shifts to False. ● When the TCP connection is aborted on the server side.
bSent	● When a data packet is sent over TCP.	● When bEnable shifts to False. ● When bSend shifts to True.
bRcvd	● When a data packet is received over TCP.	● When bEnable shifts to False. ● When bRecvRestart shifts to True.
bError	● When an error occurs during execution or the input value of the instruction is incorrect.	● When bEnable shifts to False.

■ **Timing Diagram**



■ **Functon**

Use the FB instruction (DFB_TCP_Server) to create TCP connection so as to send or receive TCP data packets.

● **Programming Example**

■ This example uses the FB instruction (DFB_TCP_ Server) to create TCP connection (Port: 502) and restrict the IP address of Client to be 192.168.1.111, while expecting the data length of packet received would be 1000 Bytes.

TCP_Server

```
PROGRAM TCP_Server
VAR
    bVar0 : BOOL := TRUE;
    FB0 : DL_EthernetLib.DFB_TCP_Server;
    bEnable_Var,bSend_Var,bRestart_Var,bBusy_Var,bConnected_Var,bSent_Var,bRcvd_Var,bError_Var : BOOL;
    RemoteInfo_Var: DL_EthernetLib.tcpServerSocketInfo;
    ErrorID_Var: DL_EthernetLib.DFB_SOCKET_ERROR;
    Status_Var: DL_EthernetLib.DFB_SOCKET_STATUS;
    uiRcvLen_Var: UINT;
    by_arVar0,by_arVar1:ARRAY[0..2000] OF BYTE;
END_VAR
```

100 %

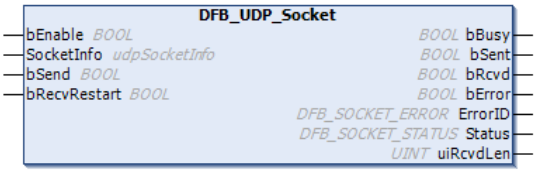
```
IF bVar0 THEN
    bVar0:=FALSE;
    RemoteInfo_Var.byIPAddr[0]:=192;
    RemoteInfo_Var.byIPAddr[1]:=168;
    RemoteInfo_Var.byIPAddr[2]:=1;
    RemoteInfo_Var.byIPAddr[3]:=111;
    RemoteInfo_Var.uiLPort:=502;
    RemoteInfo_Var.uiTimeOut:=3000; //ms
    RemoteInfo_Var.uiKeepAliveTimeout:=300; //Sec
    RemoteInfo_Var.pSendBuf:=ADR(by_arVar0);
    RemoteInfo_Var.uiSendLen:=1000;
    RemoteInfo_Var.pRecvBuf:=ADR(by_arVar1);
    RemoteInfo_Var.uiRecvBufSize:=2000;
    RemoteInfo_Var.uiSetValue:=1000;
    RemoteInfo_Var.recvCondition:=DL_EthernetLib.DFB_SOCK_RECV_MODE.DFB_SOCK_MODE_SPECIFIC_LENGTH;
    bEnable_Var:=TRUE;
END_IF

FB0(
    bEnable:=bEnable_Var ,
    SocketInfo:=RemoteInfo_Var ,
    bSend:=bSend_Var ,
    bRecvRestart:=bRestart_Var ,
    bBusy=>bBusy_Var ,
    bConnected=>bConnected_Var ,
    bSent=>bSent_Var ,
    bRcvd=>bRcvd_Var ,
    bError=>bError_Var ,
    ErrorID=>ErrorID_Var ,
    Status=>Status_Var ,
    uiRcvdLen=>uiRcvLen_Var );
```

- **Supported Products**
  - AX308E controller

- **Library**
  - DL_EthernetLib.library

8.3 DFB_UDP_Socket

DFB_UDP_Socket: UDP socket instruction

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ UDP_Socket		DFB_UDP_Socket(bEnable:= , RemoteInfo:= , bSend:= , bRecvRestart:= , bBusy=> , bSent=> , bRcvd=> , bError=> , ErrorID=> , Status=> , uiRcvdLen=>);

■ **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block. *1*2	BOOL	True/False (False)
SocketInfo	Connection information of socket	udpSocketInfo	--
bSend	Send data packets. (Rising-edge triggered)	BOOL	True/False (False)
bRecvRestart	Restart to receive data packets*2 (Rising-edge triggered)	BOOL	True/False (False)

*1 After the function block is executed, it starts receiving data packets. When the data receiving is completed and stopped, bRcvd would be ON. If you shift bRecvRestart to ON, the FB will restart to receive data.

■ **udpSocketInfo**

Name	Function	Data Type	Setting Value (Default value)
byIPAddr	The slave IP address allowed to be connected.	ARRAY [0..3] OF BYTE	[0.0.0.0]: No limit.
uiLPort(*1)	Communication ports on local device	UINT	0: Use a random port number to send data packets. 0 ~ 65535 (0)
uiRPort(*1*2)	Communication ports on remote device	UINT	0: Receive data packets from a random port. 1 ~ 65535 (0)
pSendBuf	The memory address of data to be sent	POINT TO BYTE	--
uiSendLen	The length of data to be	UINT	0 ~ 8192

Name	Function	Data Type	Setting Value (Default value)
	sent (Unit: Byte)		
pRecvBuf	The memory address where the received data to be stored.	POINT TO BYTE	--
uiRecvBufSize	The memory size of received data (Unit: Byte)	UINT	0 ~ 8192 (0)
uiSetValue	The setting value of recvCondition	UINT	(0)
recvCondition	Conditions for data receiving completion	DFB_SOCK_RECV_MODE	(DFB_SOCKET_NO_RECEVING)

*1: The values of uiLPort and uiRPort cannot be 0 at the same time.

*2: UDP data packets are not allowed to be sent when uiRPort is set to 0.

■ **DFB_SOCK_RECV_MODE**

Name	Description
DFB_SOCK_MODE_NO_RECEVING	No receiving data mode.
DFB_SOCK_MODE_SPECIFIC_LENGTH	Specific data length mode: A specific quantity of data is received and the receiving task is completed. The data length can be specified via uiSetValue. (Unit: Byte)
DFB_SOCK_MODE_SPECIFIC_SINGLE_CHAR	Specific end character mode: The data received ends with a specific character (1 Byte). The end character can be configured via uiSetValue. e.g.: If uiSetValue is set to 16#00000D0A, the end character would be 16#0A.(*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_TWO_CHARS	Specific two end characters mode: The data received ends with the two specific characters (2 Bytes) The end character can be configured via uiSetValue e.g.: If uiSetValue is set to 16#00000D0A, the end characters would be 16#0D0A. (*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR	Specific start character and end character mode: The data received starts with a specific character, and ends with a specific character. Both the start and the end character can be configured via uiSetValue. e.g.: If uiSetValue is set to 16#00003A0A, the start character would be 16#3A and the end character is 16#0A. (*1*2)
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_ANY_LENGTH	Any length mode: The receiving ends with a complete data of any length. (*1)

*1: When the length of received data reaches the limit set in pRecvLen, the receiving task would be completed.
*2: The data length includes both start and end characters.

■ **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False (False)
bSent	TCP is connected.	BOOL	True/False (False)
bRcvd	Sending completed	BOOL	True/False (False)
bError	Receiving completed	BOOL	True/False (False)
ErrorID	True if an error occurs.	DFB_SOCKET_ERROR	(DFB_SOCK_NO_ERROR)
Status	Indicates the error code if an error occurs.	DFB_SOCKET_STATUS	(SOCKET_CLOSED)
uiRcvLen	The execution status of socket.	UINT (Unit: Byte)	(0)

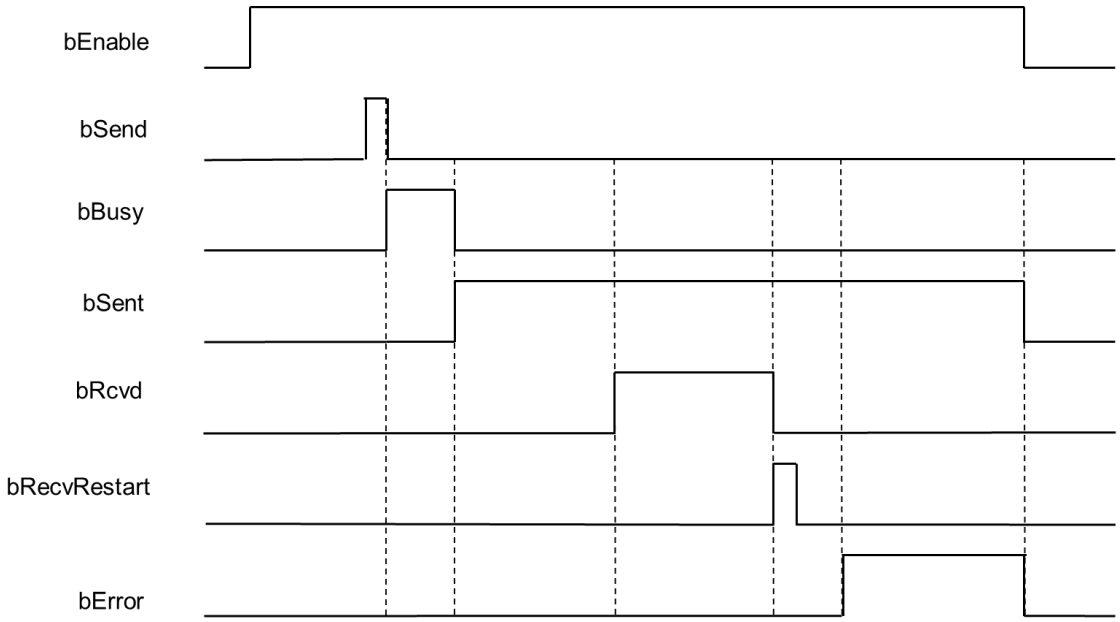
■ **DFB_SOCKET_STATUS**

Name	Description	Applicable Protocol
SOCKET_CLOSED	SOCKET connection is closed.	TCP / UDP
SOCKET_CONNECTING	SOCKET is connecting.	TCP
SOCKET_CONNECTED	SOCKET is connected.	TCP
SOCKET_SENDING	SOCKET is sending the data packet.	TCP / UDP
SOCKET_SENT	SOCKET has sent the data packet.	TCP / UDP
SOCKET_RECEIVED	SOCKET has received the data packet.	TCP / UDP
SOCKET_ERROR	SOCKET has errors.	TCP / UDP
SOCKET_ABORTED	SOCKET connection is aborted.	TCP
SOCKET_READY	SOCKET connection is ready.	UDP

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bBusy	● When bSend shifts to True.	● When bEnable shifts to False. ● When the task is completed.
bSent	● When the UDP data packet has been sent successfully.	● When bEnable shifts to False. ● When bSend shifts to True.
bRcvd	● When the UDP data packet has been received.	● When bEnable shifts to False. ● When bRecvRestart shifts to True.
bError	● When an error occurs during execution or the input value of the instruction is incorrect.	● When bEnable shifts to False.

■ **Timing Diagram**



■ **Function**

Use the FB instruction (DFB_UDP_Socket) to send or receive UDP data packets.

■ **Programming Example**

- In the following example, the FB instruction (DFB_UDP_Socket) sends a 1000-byte length UDP data packet to the IP address 192.168.1.111 (Port: 3000), while expecting to receive a 1000-byte length UDP data packet.

UDP

```
PROGRAM UDP
VAR
    bVar0 : BOOL :=TRUE;
    FB0: DL_EthernetLib.DFB_UDP_Socket;
    bEnable_Var,bSend_Var,bRestart_Var,bBusy_Var,bSent_Var,bRcvd_Var,bError_Var : BOOL;
    RemoteInfo_Var: DL_EthernetLib.udpSocketInfo;
    ErrorID_Var: DL_EthernetLib.DFB_SOCKET_ERROR;
    Status_Var: DL_EthernetLib.DFB_SOCKET_STATUS;
    uiRcvdLen_Var: UINT;
    by_arVar0,by_arVar1 : ARRAY[0..2000] OF BYTE;
END_VAR
```

100 %

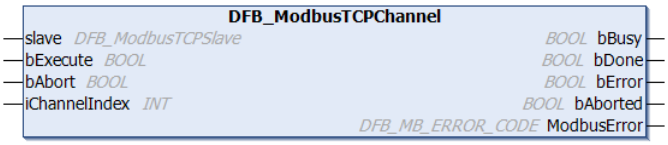
```
IF bVar0 THEN
    bVar0:=FALSE;
    RemoteInfo_Var.byIPAddr[0]:=192;
    RemoteInfo_Var.byIPAddr[1]:=168;
    RemoteInfo_Var.byIPAddr[2]:=1;
    RemoteInfo_Var.byIPAddr[3]:=111;
    RemoteInfo_Var.uiLPort:=2000;
    RemoteInfo_Var.uiRPort:=3000;
    RemoteInfo_Var.pSendBuf:=ADR(by_arVar0);
    RemoteInfo_Var.uiSendLen:=1000;
    RemoteInfo_Var.pRecvBuf:=ADR(by_arVar1);
    RemoteInfo_Var.uiRecvBufSize:=2000;
    RemoteInfo_Var.uiSetValue:=1000;
    RemoteInfo_Var.recvCondition:=DL_EthernetLib.DFB_SOCK_RECV_MODE.DFB_SOCK_MODE_SPECIFIC_LENGTH;
    bEnable_Var:=TRUE;
    bSend_Var:=TRUE;
END_IF
FB0(
    bEnable:=bEnable_Var ,
    SocketInfo:=RemoteInfo_Var ,
    bSend:=bSend_Var ,
    bRecvRestart:=bRestart_Var ,
    bBusy=>bBusy_Var ,
    bSent=>bSent_Var ,
    bRcvd=>bRcvd_Var ,
    bError=>bError_Var ,
    ErrorID=>ErrorID_Var ,
    Status=>Status_Var ,
    uiRcvdLen=>uiRcvdLen_Var );
```

- **Supported Products**
  - AX308E controller

- **Library**
  - DL_EthernetLib.library

8.4 DFB_ModbusTCPChannel

DFB_ModbusTCPChannel: Modus TCP slave Channel control instruction

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ ModbusTCPChannel		DFB_ModbusTCPChannel(slave:= , bExecute:= , bAbort:= , iChannelIndes:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusError=>);

■ **In/ Outs**

Name	Function	Data type	Setting value (Default value)
Slave	Delta Modbus TCP slave device	DFB_ModbusTCPSlave	--

■ **Input**

Name	Function	Data type	Setting value (Default value)
bExcute	Execute the function block. (Rising-edge triggered)	BOOL	True/False (False)
bAbort	No function	BOOL	--
iChannelIndex	Channel number	INT	0 ~ 99 (0)

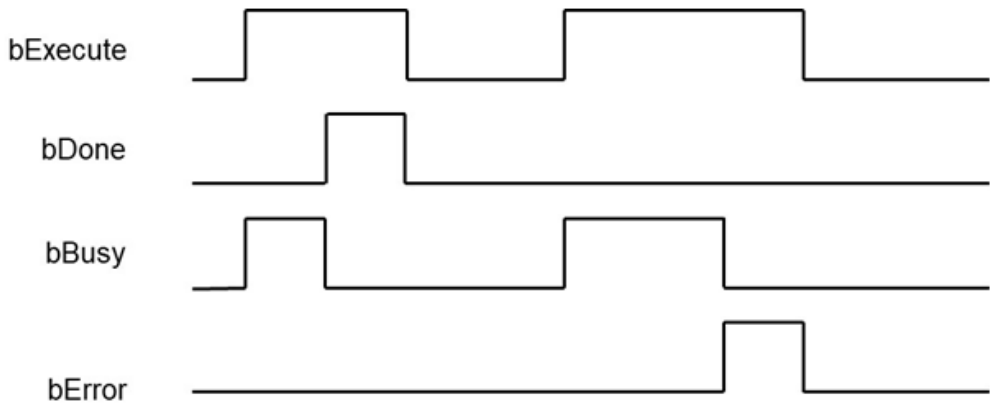
■ **Output**

Name	Function	Data type	Output range (Default value)
bBusy	The FB instruction is being executed.	BOOL	True/False (False)
bDone	The FB instruction execution is completed.	BOOL	True/False (False)
bError	FB instruction error flags.	BOOL	True/False (False)
bAborted	No function	BOOL	--
ModbusError	Error codes	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (UNDEFINED)

■ Outputs Updating Timing

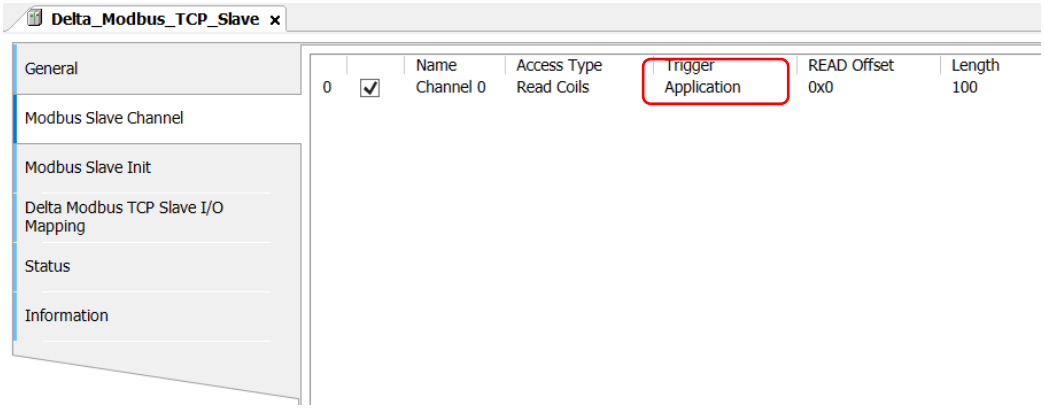
Name	Timing for shifting to True	Timing for shifting to False
bDone	● When FB instruction execution is completed.	● When bExecute shifts to False.
bBusy	● When FB instruction execution starts.	● When FB instruction execution is completed. ● When bExecute shifts to False.
bError ModbusError	● When an error occurs during execution or the input value of the instruction is incorrect.	● When bExecute shifts to False.

■ Timing Diagram



■ Function

When the trigger mode of the Modbus slave channel is set to Application, the Modbus TCP request action can be triggered by DFB_ModbusTCPChannel.

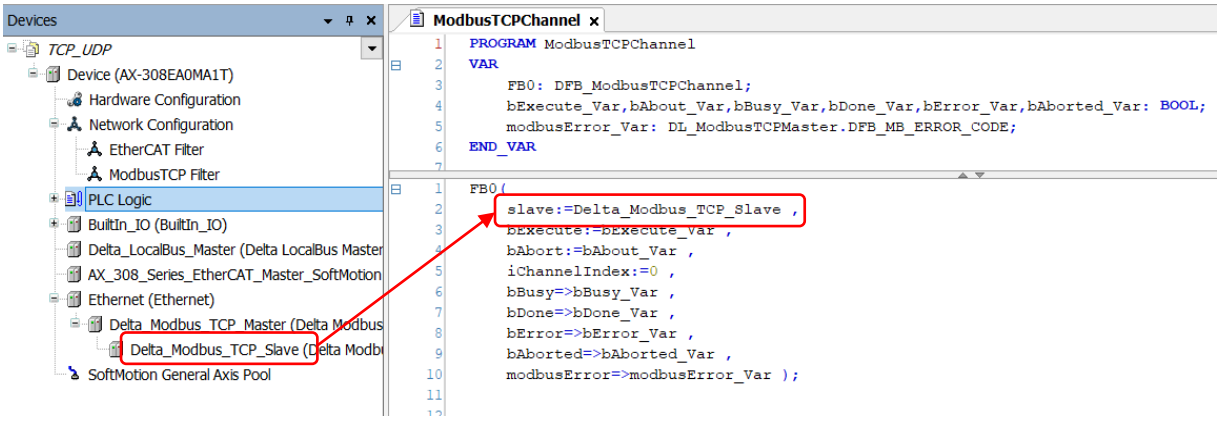


Note 1: For more details of Modbus TCP slave configuration, please refer to chapter 9.3 “Ethernet Communication” in AX-3 Series Operational Manual.

Note 2: While using, the channel must be set to “Enable”.

- **Programming Example**

  This example uses DFB_ModbusTCPChannel to trigger channel 0 of Modbus TCP Slave.



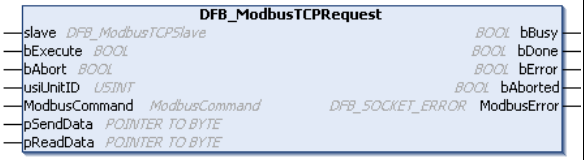
  ***Note:** The input of Slave would be the name of Delta_Modbus_TCP_Slave device.

- **Supported Products**
  - AX308E controller

- **Library**
  - DL_ModbusTCPMaster.library

8.5 DFB_ModbusTCPRequest

DFB_ModbusTCPRequest: Mobus TCP command.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ ModbusTCPRequest		DFB_ModbusTCPRequest(slave:= bExecute:= , bAbort:= , usiUnitID:= , ModbusCommand:= , pSendData:= , pRecvData:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusError=>);;

■ **In/ Outs**

Name	Function	Data type	Setting value (Default value)
Slave	Delta Modbus TCP slave device	DFB_ModbusTCPSlave	--

■ **Input**

Name	Function	Data type	Setting value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False (False)
bAbort	No function	BOOL	--
usiSlaveAddr	Slave station number	USINT	1~247
ModbusCommand	Modbus parameter setting	ModbusCommand	--
pSendData	The memory address of data to be sent.	POINTER TO BYTE	--
pRecvData	The memory address of received data to be stored.	POINTER TO BYTE	--

■ **ModbusCommand**

Name	Function	Data Type	Output Range(Default value)
FunctionCode	Modbus function codes	DFB_MB_FUNC_CODE	Supported function codes: 0x01: Read Coils 0x02: Read Discrete Inputs 0x03: Read Holding Registers 0x04: Read Input Registers 0x05: Write Single Coil

			0x06: Write Single Register 0x0F: Write Multiple Coils 0x10: Write Multiple Registers 0x17:Read/Write Multiple Registers (0x03)
uiReadOffset	The start address of memory to be read.	UINT	0 ~ 65535. (0)
uiReadLen	The data length of the memory to be read.	UINT	Coil: 1 ~ 1992 Register: 1 ~ 124. (1)
uiWriteOffset	The start address of memory to be written.	UINT	0 ~ 65535. (0)
uiWriteLen	The data length of the memory to be written	UINT	Coil: 1 ~ 1960 Register: 1 ~ 122. (1)

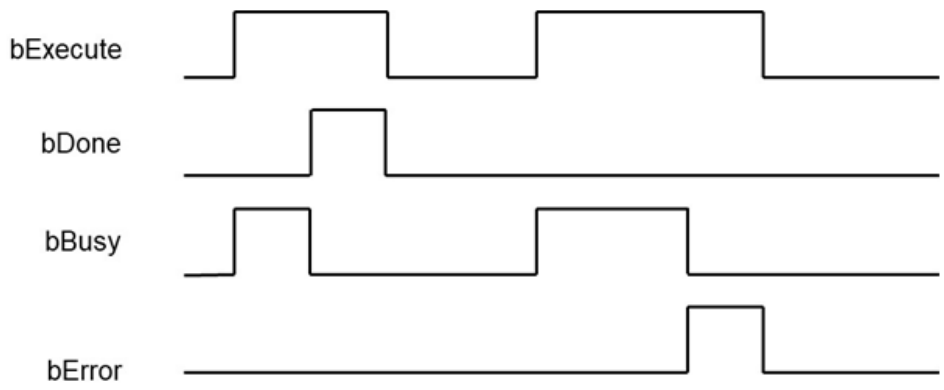
■ **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False (False)
bDone	The execution of FB is completed.	BOOL	True/False (False)
bError	True if an error occurs.	BOOL	True/False (False)
bAborted	No function	BOOL	--
ModbusError	Error code	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (DFB_UNDEFINED)

■ **Output Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When FB instruction execution starts.	● When the execution of FB is completed. ● When bExecute shifts to False.
bError	● When an error occurs in the execution conditions or input values for the instruction.	● When bExecute shifts to False.
ModbusError		

■ **Timing Diagram**



■ **Function**

The FB instruction (DFB_ModbusTCPRequest) is used for sending Modbus communication data. You must finish the configuration of Delta_Modbus_TCP_Master and Delta_Modus_TCP_Slave before using this instruction. (For more details, please refer to chapter 9.3 “Ethernet Communication” in AX-3 Series Operational Manual.)

● **Programming Example**

■ This example uses DFB_ModbusTCPRequest to send standard Modbus command (0x17) for reading a 100-word long data in the slave station (Delta_Modbus_TCP_Slave), which the start address is 0x0000, and writing a data of 100-word long to the start address 0x0100 in the memory.

ModbusTCPRequest

```
VAR
    FB0: DFB_ModbusTCPRequest;
    bVar0: BOOL:=TRUE;
    bEcecute_Var,bAbort_Var,bBusy_Var,bDone_Var,bError_Var,bAborted_Var: BOOL;
    usiUnitID_Var: USINT;
    ModbusCommand_Var: DL_ModbusTCPMaster.ModbusCommand;
    ar_byVar0,ar_byVar1: ARRAY[0..2000] OF BYTE;
    ModbusError_Var: DL_ModbusTCPMaster.DFB_MB_ERROR_CODE;
END_VAR
```

100 %

```
IF bVar0 THEN
    bVar0:=FALSE;
    ModbusCommand_Var.FunctionCode:=DL_ModbusTCPMaster.DFB_MODBUS_FUNC.ReadWrite_Multiple_Registers;
    ModbusCommand_Var.uiReadLen:=100;
    ModbusCommand_Var.uiReadOffset:=16#0000;
    ModbusCommand_Var.uiWriteLen:=100;
    ModbusCommand_Var.uiWriteOffset:=16#0100;
    usiUnitID_Var:=12;
    bEcecute_Var:=TRUE;
END_IF
FB0(
    slave:=Delta_Modbus_TCP_Slave ,
    bExecute:=bEcecute_Var ,
    bAbort:=bAbort_Var ,
    usiUnitID:=usiUnitID_Var ,
    ModbusCommand:= ModbusCommand_Var,
    pSendData:=ADR(ar_byVar0) ,
    pReadData:=ADR(ar_byVar1) ,
    bBusy=>bBusy_Var ,
    bDone=>bDone_Var ,
    bError=>bError_Var ,
    bAborted=>bAborted_Var ,
    ModbusError=>ModbusError_Var );
```

● **Supported Products**

■ AX308E controller

● **Library**

■ DL_ModbusTCPMaster.library

8.6 Error Codes and Troubleshooting

- DFB_SOCKET_ERROR

Description	Cause of Error	Corrective Action
DFB_SOCK_ERR_NO_ERROR	No errors.	--
DFB_SOCK_ERR_INITIALIZE_FAILED	Socket connection failed.	● Please check if the server exists. ● Please make sure the server configuration is correct.
DFB_SOCK_ERR_CONNREFUSED	Socket connection refused.	Please make sure the server configuration is correct.
DFB_SOCK_ERR TIMEDOUT	Server timeout error	● Please make sure the internet connection is normal. ● Please make sure the server configuration is correct.
DFB_SOCK_ERR NOTCONNECTED	Socket has not been connected.	● Please wait for Socket being connected. ● Please make sure the server configuration is correct.
DFB_SOCK_ERR CLOSED	FB instruction has not yet effective.	Please make sure the input bEnable is ON.
DFB_SOCK_ERR INVALID_SETTING	Invalid setting values for FB instruction.	Please check if the setting value of uiSetValue is correct.
DFB_INVALID_BUFFER	Invalid memory address	Please make sure the memory addresses given to pSenbuf and pRecvbuf are correct.
DFB_INVALID_LENGTH	Invalid setting value for data length	Please make sure the input values of uiSendLen and uiRecvLen are correct.

- DFB_MB_ERROR_CODE

Description	Cause of Error	Corrective Action
DFB_NO_ERROR	No errors	--
DFB_ILLEGAL_FUNCTION	Unsupported function code.	Please check on the correctness of the function code you're using.
DFB_ILLEGAL_DATA_ADDRESS	Illegal memory address to write and read.	Please check on the correctness of memory address you intend to write and read.
DFB_ILLEGAL_DATA_VALUE	Illegal data values responded by slave.	Please check if the wires function normally as well as the proper wiring.
DFB_RESPONSE_TIMEOUT	No response from slave in time.	● Please check if the duration set for the timeout is less than the responded time of slave. ● Please check on the correctness of the wiring.
DFB_RESPONSE_CRC_ERROR	Illegal data values responded by slave. (Invalid check code)	Please check on the correctness of data format responded by slave.
DFB_RESPONSE_WRONG_SLAVE	Illegal data values responded by slave. (Invalid station number)	Please check on the correctness of data format responded by slave.
DFB_RESPONSE_WRONG_FUNCTIONCODE	Illegal data values responded by slave. (Invalid function code)	Please check on the correctness of data format responded by slave.
DFB_REQUEST_FAILED_TO_SEND	Failed to send data.	Please contact the vendor directly.
DFB_RESPONSE_INVALID_PROTOCOL	Illegal data values responded by slave. (Non-standard Modbus format)	Please check on the correctness of data format responded by slave.
DFB_RESPONSE_INVALID_HEAD	Illegal data values responded by slave. (Invalid data length)	Please check on the correctness of data format responded by slave.

DFB_INVALID_CHANNEL_INDEX	Invalid index of the slave channel	Please check if the index of slave channel is correct.
DFB_CHANNEL_SETTING_NOT_SUPPORT	The trigger mode of slave channel is not set to “Application”.	Make sure the trigger mode has been set to “Application”.
DFB_INVALID_SLAVE	Slave configuration error.	Please check the correctness of DFB_ModbusTCPRequest Slave configuration.
DFB_INVALID_BUFFER	Invalid memory address setting to send and receive data.	Please check if the settings of pWriteBuf and pReadBuf of DFB_ModbusTCPRequest are correct.
DFB_INVALID_LENGTH	Invalid data length setting.	Please check if the settings of uiReadLen and uiWriteLen of DFB_ModbusTCPRequest are correct.
DFB_INVALID_SLAVE_ADDRESS	Invalid slave station number.	Please check if the settings of usiSlaveAddr of DFB_ModbusTCPRequest are correct.
DFB_NO_MDBSCOM_CONFIG	There is no Ethernet device in project.	Please check if the Ethernet component has been added to the device tree.
DFB_NO_MASTER_CONFIG	Delta_Modbus_TCP_Master device does not exist.	Please check if Delta_Modbus_TCP_Master device has been added to the device tree.
DFB_MB_ERROR_CODE_MEMORY_NOT_ENOUGH	Not enough system memory	Please check if the program size exceeds the limit
DFB_CONNECTION_ TIMEOUT	TCP connection timeout.	● Please check if the setting of Modbus TCP Slave is correct. ● Please check on the correctness of the wiring
DFB_CONNECTION_ FAILED	TCP connection failed.	Please check if the setting of Modbus TCP Slave is correct.
DFB_UNDEFINED	Undefined or has not yet been executed.	Please wait for the execution of the FB instruction completed.

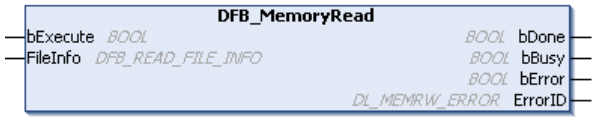
9

Chapter 9 Instructions for Reading and Writing a Memory Card

Table of Content

9.1 DFB_MemoryRead

DFB_MemoryRead: Read a memory card.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_MemoryRead		DFB_MemoryRead(bExecute:= , FileInfo:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False(False)
FileInfo	Parameter setting for reading a file.	DFB_READ_FILE_INFO	

■ **DFB_READ_FILE_INFO**

Name	Function	Data Type	Setting Value (Default value)
sFilePath	The name of the file to read.	STRING	('')
wDataMode	ASCII CODE / BINARY mode	DFB_DATA_ MODE	DFB_DATA_MODE. ASCII_MODE DFB_DATA_MODE.BINARY_MODE (DFB_DATA_MODE.ASCII_MODE)
wAsciiShowMode	The display mode of data to be read. (Deximal/ Hexadecimal)	DFB_ASCII SHOW_MODE	DFB_ASCIISHOW_MODE.DECIMAL DFB_ASCIISHOW_MODE.HEX (DFB_ASCIISHOW_MODE.DECIMAL)
wAsciiDec DataType	Data type of the variables to be read.	DFB_DEC_ DATATYPE	DFB_DEC_DATATYPE.BYTE_SIZE DFB_DEC_DATATYPE.WORD_SIZE DFB_DEC_DATATYPE.DWORD_SIZE DFB_DEC_DATATYPE.LWORD_SIZE DFB_DEC_DATATYPE.SINT_SIZE DFB_DEC_DATATYPE.USINT_SIZE DFB_DEC_DATATYPE.INT_SIZE DFB_DEC_DATATYPE.UINT_SIZE DFB_DEC_DATATYPE.DINT_SIZE DFB_DEC_DATATYPE.UDINT_SIZE DFB_DEC_DATATYPE.LINT_SIZE DFB_DEC_DATATYPE.ULINT_SIZE DFB_DEC_DATATYPE.REAL_SIZE DFB_DEC_DATATYPE.LREAL_SIZE (DFB_DEC_DATATYPE.BYTE_SIZE)
dwRead StartPos	The address of the start position to read the memory card's data.*	DWORD	(0)

Name	Function	Data Type	Setting Value (Default value)
dwElement Length	The length of the data in the controller's memory card.*	DWORD	1 ~ 25,000 (0)
pDestination	The address of the destination to store the controller's memory data.	POINTER TO BYTE	NULL

***Note:** The unit is defined in DFB_READ_FILE_INFO.wAsciiDecDataType.

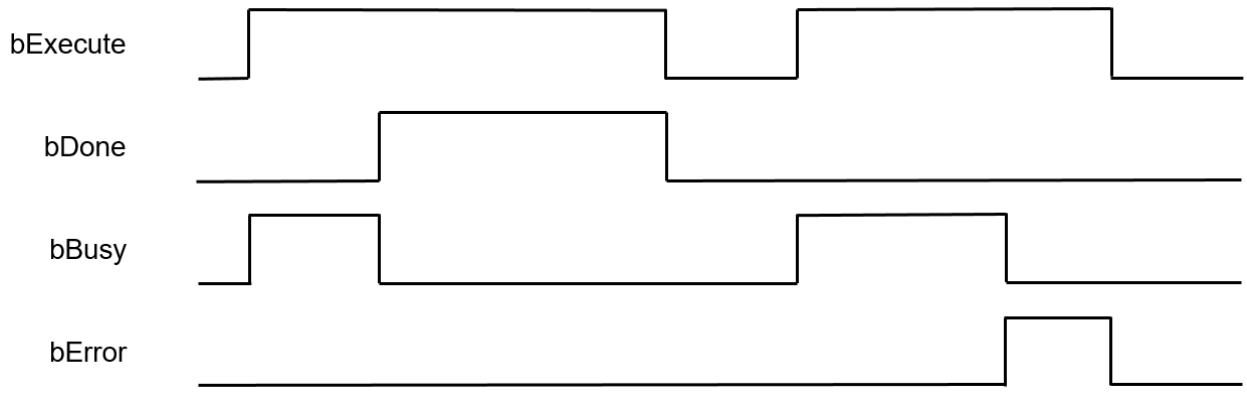
- **Output**

Name	Function	Data type	Output range (Default value)
bDone	The FB instruction execution is completed.	BOOL	True/False(False)
bBusy	The FB instruction is being executed.	BOOL	True/False(False)
bError	FB instruction error flags.	BOOL	True/False(False)
ErrorID	Error codes	DL_MEMRW_ERROR	DL_MEMRW_ERROR (DFB_NO_ERR)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When FB instruction execution is completed.	● When bExecute shifts to False.
bBusy	● When FB instruction execution starts.	● When FB instruction execution is completed. ● When bExecute shifts to False.
bError	● When an error occurs during execution or the input value of the instruction is incorrect.	● When bExecute shifts to False.
ErrorID		

- **Timing Diagram**



- **Function**

Use the FB instruction(DFB_MemoryRead) to store the retrieved memory card data in the controller's memory.

- **Programming Example**

This example uses the FB instruction(DFB_MemoryRead) to read the content of Test.csv file in the memory card and store the data in the controller's WORD-type array variable(ar_wVar0).

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bVar0: BOOL := TRUE;
    bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
    FB0: DFB_MemoryRead;
    FILE_INFO_Var: DFB_READ_FILE_INFO;
    ar_wVar0: ARRAY[0..3] OF WORD;
    ErrorID_Var: DL_MEMRW_ERROR;
END_VAR
```

```
IF bVar0 THEN
    FILE_INFO_Var.sFilePath:='Test.csv';
    FILE_INFO_Var.wDataMode:=DFB_DATA_MODE.ASCII_MODE;
    FILE_INFO_Var.wAsciiShowMode:=DFB_ASCIISHOW_MODE.HEX;
    FILE_INFO_Var.wAsciiDecDataType:=DFB_DEC_DATATYPE.WORD_SIZE;
    FILE_INFO_Var.dwReadStartPos:=0;
    FILE_INFO_Var.dwElementLength:=4;
    FILE_INFO_Var.pDestination:=ADR(ar_wVar0);
    bExecute_Var:=TRUE;
    bVar0:=FALSE;
END_IF;
IF bDone_Var THEN
    bExecute_Var:=FALSE;
END_IF
FB0(
bExecute:=bExecute_Var ,
FileInfo:=FILE_INFO_Var ,
bDone=>bDone_Var ,
bBusy=>bBusy_Var ,
bError=>bError_Var ,
ErrorID=>ErrorID_Var );
```

The content of Test.csv file in the memory card is shown as follows.

Values displayed in the Test.csv file					
0	1	2	3	4	5

Read the four consecutive data starting from data 0 in the Test.csv file via the FB instruction(DFB_MemoryRead), then store the retrieved data in the variable array(ar_wVar0), which the result would be ar_wVar0 := [0,1,2,3].

- **Supported Product**
  - AX-308E

- **Library**
  - DL_MemRW_AX3.library

9.2 DFB_MemoryWrite

DFB_MemoryWrite: Write a memory card.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_MemoryWrite		DFB_MemoryWrite (bExecute:= , FileInfo:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False(False)
FileInfo	Parameter setting for reading a file.	DFB_READ_FILE_INFO	

■ **DFB_WRITE_FILE_INFO**

Name	Function	Data Type	Setting Value (Default value)
sFilePath	The name of the file to create.	STRING	(‘’)
wDataMode	ASCII CODE / BINARY mode	DFB_DATA_MODE	DFB_DATA_MODE.ASCII_MODE DFB_DATA_MODE.BINARY_MODE (DFB_DATA_MODE.ASCII_MODE)
wAsciiShowMode	The display mode of data to be written. (Deximal/ Hexadecimal)	DFB_ASCIISHOW_MODE	DFB_ASCIISHOW_MODE.DECIMAL DFB_ASCIISHOW_MODE.HEX (DFB_ASCIISHOW_MODE.DECIMAL)
wAsciiDecData Type	Data type of the variables to be written.	DFB_DEC_DATATYPE	DFB_DEC_DATATYPE.BYTE_SIZE DFB_DEC_DATATYPE.WORD_SIZE DFB_DEC_DATATYPE.DWORD_SIZE DFB_DEC_DATATYPE.LWORD_SIZE DFB_DEC_DATATYPE.SINT_SIZE DFB_DEC_DATATYPE.USINT_SIZE DFB_DEC_DATATYPE.INT_SIZE DFB_DEC_DATATYPE.UINT_SIZE DFB_DEC_DATATYPE.DINT_SIZE DFB_DEC_DATATYPE.UDINT_SIZE DFB_DEC_DATATYPE.LINT_SIZE DFB_DEC_DATATYPE.ULINT_SIZE DFB_DEC_DATATYPE.REAL_SIZE DFB_DEC_DATATYPE.LREAL_SIZE (DFB_DEC_DATATYPE.BYTE_SIZE)
wAccessMode	The access mode of the	DFB_ACCESS_M	DFB_ACCESS_MODE.NEW

Name	Function	Data Type	Setting Value (Default value)
	file to be created.	ODE	DFB_ACCESS_MODE.APPEND DFB_ACCESS_MODE.OVERWRITE DFB_ACCESS_MODE.INSERT (DFB_ACCESS_MODE.NEW)
wCarriageReturn	CRLF character*	WORD	(0)
dwWriteStartPos	The address of the start position to write the memory card's data.*	DWORD	(0)
dwElementLength	The length of the data to write to the controller's memory card.*	DWORD	1 ~ 25,000 (0)
pSource	The memory address for the controller to store the data.	POINTER TO BYTE	NULL

***Note:** The unit is defined DFB_WRITE_FILE_INFO.wAsciiDecDataType.

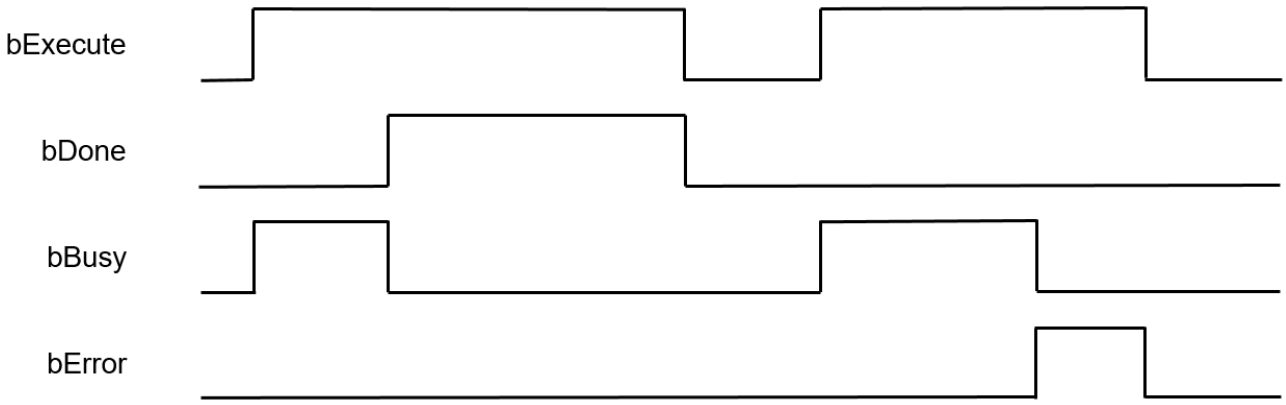
- **Output**

Name	Function	Data type	Output range (Default value)
bDone	The FB instruction execution is completed.	BOOL	True/False(False)
bBusy	The FB instruction is being executed.	BOOL	True/False(False)
bError	FB instruction error flags.	BOOL	True/False(False)
ErrorID	Error codes	DL_MEMRW_ERROR	DL_MEMRW_ERROR (DFB_NO_ERR)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When FB instruction execution is completed.	● When bExecute shifts to False.
bBusy	● When FB instruction execution starts.	● When FB instruction execution is completed. ● When bExecute shifts to False.
bError	● When an error occurs during execution or the input value of the instruction is incorrect.	● When bExecute shifts to False.
ErrorID		

- **Timing Diagram**



- **Function**

Write the internal data of the controller to the memory card via the FB instruction(DFB_MemoryWrite).

- **Programming Example**

This example uses the FB instruction(DFB_MemoryWrite) to write the WORD-type array variable to the memory card.

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bVar0: BOOL :=TRUE;
    bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
    ar_wVar0: ARRAY [0..3] OF WORD := [0,1,2,10];
    FB0: DFB_MemoryWrite;
    FILE_INFO_Var: DFB_WRITE_FILE_INFO;
    ErrorID_Var0: DL_MEMRW_ERROR;
END_VAR
```

```
IF bVar0 THEN
    FILE_INFO_Var.sFilePath:='Test.csv';
    FILE_INFO_Var.wDataMode:=DFB_DATA_MODE.ASCII_MODE;
    FILE_INFO_Var.wAsciiShowMode:=DFB_ASCIISHOW_MODE.DECIMAL;
    FILE_INFO_Var.wAsciiDecDataType:=DFB_DEC_DATATYPE.WORD_SIZE;
    FILE_INFO_Var.wAccessMode:=DFB_ACCESS_MODE.NEW;
    FILE_INFO_Var.wCarriageReturn:=0;
    FILE_INFO_Var.dwWriteStartPos:=0;
    FILE_INFO_Var.dwElementLength:=4;
    FILE_INFO_Var.pSource:=ADR(ar_wVar0);
    bExecute_Var:=TRUE;
    bVar0:=FALSE;
END_IF;
IF bDone_Var THEN
    bExecute_Var:=FALSE;
END_IF
FB0(
bExecute:=bExecute_Var ,
FileInfo:=FILE_INFO_Var ,
bDone=>bDone_Var ,
bBusy=>bBusy_Var ,
bError=>bError_Var ,
ErrorID=>ErrorID_Var0 );
```

Suppose that the written data is ar_wVar0: ARRAY [0..3] OF WORD := [0,1,2,10]. After open the .csv file in the memory card, the content would be displayed as follows.

Values displayed in the Test.csv file			
0	1	2	10

***Note:**

1. In case of wDataMode: =DFB_DATA_MODE.ASCII_MODE, the controller, the controller would write the content of array ar_wVar0 to the memory card in ASCII CODE format.
2. If wAsciiDecDataType:=DFB_DEC_DATATYPE.WORD_SIZE, the data length would be word size in the CSV file.

- **Supported Products**
  - AX-308E

- **Library**
  - DL_MemRW_AX3.library

9.3 Error Codes and Troubleshooting

Description	Cause of Error	Corrective Action
DFB_NO_ERR	No errors.	-
DFB_MEMREAD_ERR_ FAILED	Internal errors.	Please contact us directly
DFB_MEMREAD_ERR_ PARAMETER	Invalid parameter inputs.	Please check if the input parmeters are correct.
DFB_MEMREAD_ERR_ NOTINITIALIZED	The instruction cannot be executed owning to the component has not been initialized.	Please reboot the controller.
DFB_MEMREAD_ERR_ VERSION	Wrong version.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ TIMEOUT	Operation timeout.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFB_MEMREAD_ERR_ NOBUFFER	Insufficient memory.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly.
DFB_MEMREAD_ERR_ PENDING	The program is pending for execution.	Please reboot the controller.
DFB_MEMREAD_ERR_ NUMPENDING	Too many pending programs.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFB_MEMREAD_ERR_ NOTIMPLEMENTED	The function does not exeist.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ INVALIDID	Incorrect ID.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ OVERFLOW	Integer overflow.	Please check the data type of inputs and reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFB_MEMREAD_ERR_ BUFFERSIZE	The buffer size is too small.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFB_MEMREAD_ERR_NO_O BJECT	The object does not exist.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_NOM EMORY	Insufficient memory.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFB_MEMREAD_ERR_ DUPLICATE	Duplicate object name.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ MEMORY_OVERWRITE	Memory overwrite error.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFB_MEMREAD_ERR_ INVALID_HANDLE	Invalid handle for the object.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ END_OF_OBJECT	The end of the object has been reached.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_NO_C HANGE	No changes happened.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ INVALID_INTERFACE	Invalid or unknown interface	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ NOT_SUPPORTED	The function is not supported.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_NO_A CCESS_RIGHTS	No rights to access the operation.	Please check if the firmware and the library version are supported.
DFB_MEMREAD_ERR_ OUT_OF_LIMITS	Exceeds the limited sources.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦

Description	Cause of Error	Corrective Action
DFB_MEMREAD_ERR_ ENTRIES_REMAINING	Remaining entries that could not be transmitted because of the buffer limitation.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFB_MEMREAD_ERR_ INVALID_SESSION_ID	Invalid online sessionid.	Please log in again or reboot the controller.
DFB_MEMREAD_ERR_ EXCEPTION	Exception occurs.	Please check the error log.

MEMO

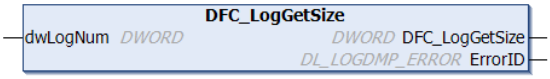
10

Chapter 10 Additional Instructions

Table of Contents

10.1 DFC_LogGetSize

DFC_LogGetSize: Read the size of controller's log files.

FB/FC	Instruction	Graphic Expression	ST Language
FC	DFC_LogGetSize		DFC_LogGetSize(dwLogNum:= , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
dwLogNum*	The number of the target log files.	DWORD	0: Calculate the current data size of all log files in the controller. (0)

***Note:** Data types such as BYTE, WORD and DWORD can be used for dwLogNum input. Currently only supports mode0.

- **Output**

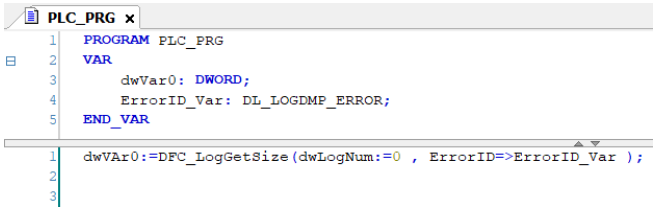
Name	Function	Data type	Output range (Default value)
DFC_LOG_ GETSIZE	Data size of the log files. (Return type)	DWORD (Unit: BYTE)	0 ~ 65536 (0)
ErrorID	Error codes	DL_LOGDMP_ERROR	DL_LOGDMP_ERR (DFC_NO_ERROR)

- **Function**

After executes the function(DFC_LogGetSize), the data size of the current log files will be calculated.

- **Programming Example**

This example uses the FC instruction(DFC_LogGetSize) to read the data size of the current log files of the controller.



- **Supported Products**
  - AX-308E

- **Library**
  - DL_LogDmp_AX3.library

10.2 DFB_LogDump

DFB_LogDump: Read the log files of the controller.

FB/FC	Instruction	Graphic Expression	ST Language
FB	DFB_ LogDump		DFB_LogDump(bExecute:= , pDmpPos:= , dwLogNum:= , dwDmpLength:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bExecute	Execute the function block. (Rising-edge triggered)	BOOL	True/False(False)
pDmpPos	The memory address for controller's storage	POINTER TO BYTE	(0)
dwLogNum*	The number of the target log files to read	DWORD	0: Read all the current log files of the controller (0)
dwDmpLength	The size of the target log files to read	DWORD	(0)

***Note:** Data types such as BYTE, WORD and DWORD can be used for dwLogNum input. Currently only supports mode0.

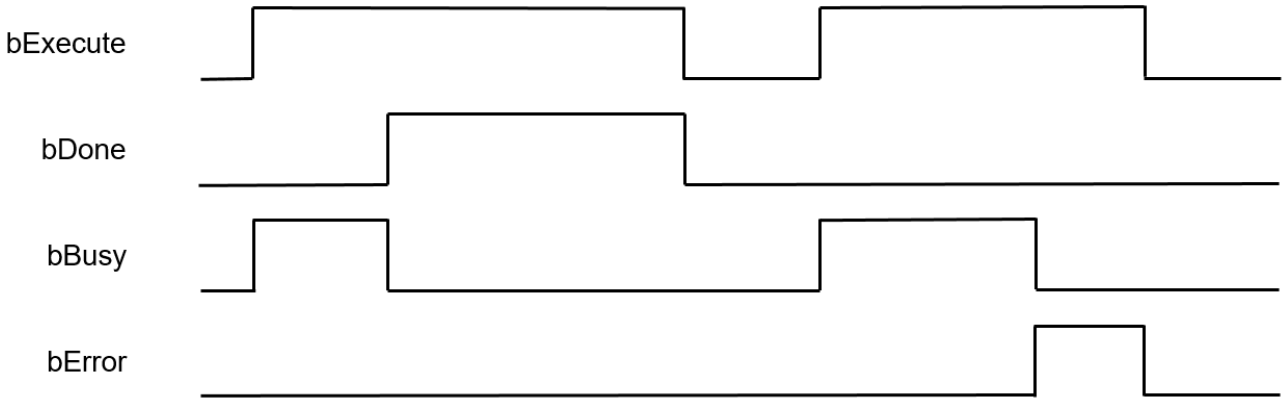
- **Output**

Name	Function	Data type	Output range (Default value)
bDone	The FB instruction execution is completed.	BOOL	True/False(False)
bBusy	The FB instruction is being executed.	BOOL	True/False(False)
bError	FB instruction error flags.	BOOL	True/False(False)
ErrorID	Error codes	DL_LOGDMP_ERROR	DL_LOGDMP_ERROR (DFB_NO_ERR)

■ **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● When FB instruction execution is completed.	● When bExecute shifts to False.
bBusy	● When FB instruction execution starts.	● When FB instruction execution is completed. ● When bExecute shifts to False.
bError	● When an error occurs during execution or the input value of the instruction is incorrect.	● When bExecute shifts to False.
ErrorID		

● **Timing Diagram**



● **Function**

Use the FB instruction(DFB_LogDump) to read the log files of the controller.

● **Programming Example**

This example uses the FB instruction(DFB_LogDump) to read the log files and store it in the Byte type array variable(ar_byVar) in ASCII CODE format.

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
    ar_byVar: ARRAY[0..9999] OF BYTE;
    dwVar0: DWORD;
    FB0: DFB_LogDump;
    ErrorID_Var: DL_LOGDMP_ERROR;
END_VAR
```

```
IF bDone_Var THEN
    bExecute_Var:=FALSE;
END_IF
FB0(
bExecute:=bExecute_Var ,
pDmpPos:=ADR(ar_byVar) ,
dwLogNum:=0 ,
dwDmpLength:=65536 ,
bDone=>bDone_Var ,
bBusy=>bBusy_Var ,
bError=>bError_Var ,
ErrorID=>ErrorID_Var );
```

● **Supported Products**

■ AX-308E

● **Library**

■ DL_LogDmp_AX3.library

10.3 Error Codes and Troubleshooting

Description	Cause of Error	Corrective Action
DFB_NO_ERR	No errors.	-
DFC_DMP_ERR_FAILED	Internal errors.	Please contact us directly
DFC_DMP_ERR_ PARAMETER	Invalid parameter inputs.	Please check if the input parmeters are correct.
DFC_DMP_ERR_ NOTINITIALIZED	The instruction cannot be executed owning to the component has not been initialized.	Please reboot the controller.
DFC_DMP_ERR_ VERSION	Wrong version.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_ TIMEOUT	Operation timeout.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFC_DMP_ERR_ NOBUFFER	Insufficient memory.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly.
DFC_DMP_ERR_ PENDING	The program is pending for execution.	Please reboot the controller.
DFC_DMP_ERR_ NUMPENDING	Too many pending programs.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFC_DMP_ERR_ NOTIMPLEMENTED	The function does not exeist.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_ INVALIDID	Incorrect ID.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_ OVERFLOW	Integer overflow.	Please check the data type of inputs and reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFC_DMP_ERR_ BUFFERSIZE	The buffer size is too small.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFC_DMP_ERR_NO_ OBJECT	The object does not exist.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_ NOMEMORY	Insufficient memory.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFC_DMP_ERR_ DUPLICATE	Duplicate object name.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_ MEMORY_OVERWRITE	Memory overwrite error.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFC_DMP_ERR_INVALID_HANDLE	Invalid handle for the object.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_END_OF_OBJECT	The end of the object has been reached.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_NO_ CHANGE	No changes happened.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_INVALID_INTERFACE	Invalid or unknown interface	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_NOT_ SUPPORTED	The function is not supported.	Please check if the firmware and the library version are supported.
DFC_DMP_ERR_NO_ ACCESS_RIGHTS	No rights to access the operation.	Please check if the firmware and the library version are supported.

Description	Cause of Error	Corrective Action
DFC_DMP_ERR_OUT_OF_LIMITS	Exceeds the limited sources.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFC_DMP_ERR_ENTRIES_REMAINING	Remaining entries that could not be transmitted because of the buffer limitation.	Please reset the controller to default (Reset Origin). Then download the project again after compressing the program. If the problem remains, please contact us directly. ◦
DFC_DMP_ERR_INVALID_SESSION_ID	Invalid online sessionid.	Please log in again or reboot the controller.
DFC_DMP_ERR_ EXCEPTION	Exception occurs.	Please check the error log.

MEMO