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

*We reserve the right to change the information in this catalogue without prior notice.

AX Series - Standard Instructions Manual

AX Series - Standard Instructions Manual

www.deltaww.com

Smarter. Greener. Together.

Revision History

Version	Revision	Date
1st	The first version was published.	2020/10/30
2nd	1. Updated CH3: • Added supported products AX-364E & AX-324N for DFB_Capture, DFB_Compare, DFB_HCnt, DFB_HTmr, DFB_PresetValue, DFB_Sample • Updated library DL_BuiltInIO_AX3.library for DFB_Capture, DFB_Compare, DFB_HCnt, DFB_HTmrr, DFB_PresetValue, DFB_Sample • Updated “dwTimerValue” unit 2. Updated CH4: • Added supported product AX-364E for DFB_EcGetAllSlaveAddr, DFB_EcGetSlaveCount, DFB_EtherCATLink_Diag, DFB_GetAllECATSlaveInfo, DFB_GetECATMasterError, DFB_GetECATMasterState, DFB_ResetECATMaster, DFB_ResetECATSlave 3. Updated CH6: • Added supported products AX-364E, AX-300, and AX-324N for DFB_From, DFB_To 4. Updated CH7: • Added supported products AX-364E,AX-300, and AX-324N for DFB_ModbusComChannel, DFB_ModbusRequest, DFB_ModbusRequest2 “DFB_MR_ERROR“ update • Added DFB_SLAVE_DEVICE_FAILURE, DFB_ACKNOWLEDGE, DFB_SLAVE_DEVICE_BUSY, DFB_GATEWAY_PATH_UNAVAILABLE, DFB_GATEWAY_DEVICE_FAILED_TO_RESPOND • Removed DFB_NO_MDBSCOM_CONFIG • Updated DFB_ILLEGAL_DATA_VALUE 5. Updated CH8: • Added supported products AX-364E, AX-300, and AX-324N for DFB_TCP_Client, DFB_TCP_Server, DFB_UDP_Socket, DFB_ModbusTCPChannel, DFB_ModbusTCPRequest “DFB_MR_ERROR“ update • Added DFB_SLAVE_DEVICE_FAILURE, DFB_ACKNOWLEDGE, DFB_SLAVE_DEVICE_BUSY, DFB_GATEWAY_PATH_UNAVAILABLE, DFB_GATEWAY_DEVICE_FAILED_TO_RESPOND, DFB_INVALID_FUNCTION_CODE, DFB_NO_ETHERNET_CONFIG	2021/02/28

Version	Revision	Date
	• Removed DFB_NO_MDBSETH_CONFIG • Updated DFB_ILLEGAL_DATA_VALUE, DFB_MEMORY_NOT_ENOUGH 6. Updated CH9: • Added supported products AX-364E, AX-300, and AX-324N for DFB_MemoryRead, DFB_MemoryWrite 7. Added CH10 High Speed Output Instructions 8. Updated CH11: • Added supported products AX-364E, AX-300, and AX-324N for DFC_LogGetSize, DFB_LogDump	
3rd	1. Updated Section 1.1: Added Setting Value range of wNum 2. Updated Section 2.5: Updated input pin note 3. Updated Section 4.3: Updated DFB_EtherCATLink_Diag library: DL_EtherCAT_Diag.library 4. Added Section 6.3 ~ Section 6.19: DFB_DLCCAL, DFB_DLCWEI, DFB_DPUCONF, DFB_PUSTAT, DFB_DPUPLS, DFB_DPUPLS, DFB_DPUDRI, DFB_DPUDRA, DFB_DPUZRN, DFB_DPUJOG, DFB_DPUCNT, DFB_DMPID library: DL_ASModuleAPI_AX3.library, DFB_DHCCNT, DFB_DHCCAP, DFB_HCDO, DFB_DHCCMP, DFB_DHCCMPT, DFB_DHCMEAS 5. Updated Section 7.3: Added Notes 4 & 5 6. Updated Section 7.4: Added Notes 1 & 2 7. Updated Sections 8.1, 8.2, and 8.3: Updated example program 8. Updated Sections 8.1 ~ 8.5: Added Function notes	2021/07/31
4th	1. Updated Section 5.1~5.3: ● Corrected output Error_ID data type ● Adjusted dwLen Input to wLen ● Updated Input pSrc range and added notes ● Updated error code description 2. Added Section 6.20 ~ 6.21: DFB_DADLOG, DFB_DADPEAK 3. Updated Section 6.22: Added DFB_DADLOG, DFB_DADPEAK error codes 4. Updated Section 9.2: Updated Input FileInfo data type 5. Updated Section 9.3: Updated error code content 6. Updated Section 11.3: Updated error code content, added DFB_FTPClient	2022/06/01
5th	1. Update Section 3.1: Added bEdgeSelect and bCycle. 2. Update Section 3.2: Added OutputAction, diTablePosition, diTableSize. 3. Update Section 3.3: Added xUD_Select, bTriggerMode.	2023/03/31

TOC

P1 Preface

P1.1 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.

P1.1.1 Applicable Products

This manual applies to the following products:

- AX-3 Series

P1.1.2 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

Chapter 1 Move Instructions

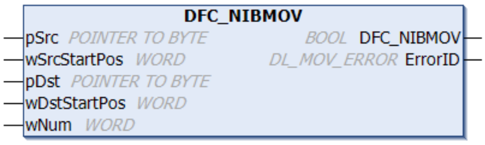
1.1 DFC_NIBMOV

- **Supported Products**

  - AX series

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*	1 ~ 256 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)

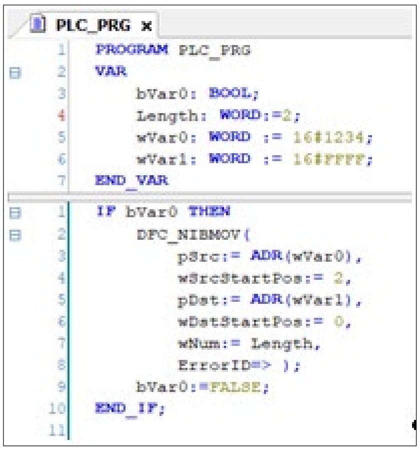
- **Function**

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).



```
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).

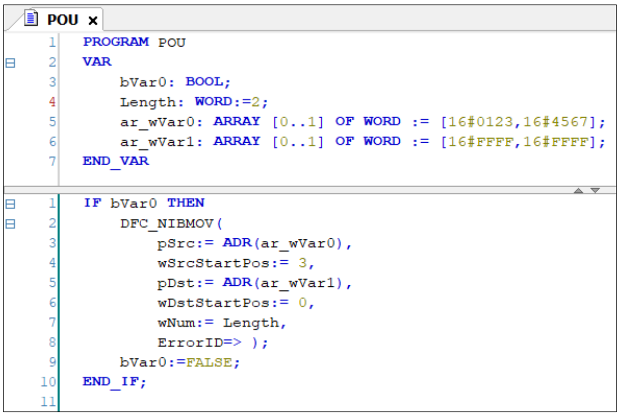
Variable	Nibble3	Nibble2	Nibble1	Nibble0
wVar0 [16#1234]	**Memory content**			
	1	2	3	4

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).



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

```
IF bVar0 THEN
    DFC_NIBMOV(
        pSrc:= ADR(ar_wVar0),
        wSrcStartPos:= 3,
        pDst:= ADR(ar_wVar1),
        wDstStartPos:= 0,
        wNum:= Length,
        ErrorID=> );
    bVar0:=FALSE;
END_IF;
```

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

- **Library**

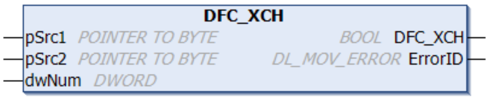
  - DL_Mov.library

1.2 DFC_XCH

- **Supported Products**
  - AX series

DFC_XCH: Data exchange between two variables.

FB/FC	Instruction	Graphic Expression	ST Language

FC	DFC_XCH	 DFC_XCH
		pSrc1 POINTER TO BYTE
		pSrc2 POINTER TO BYTE
		dwNum DWORD
		BOOL DFC_XCH
		DL_MOV_ERROR ErrorID
			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**

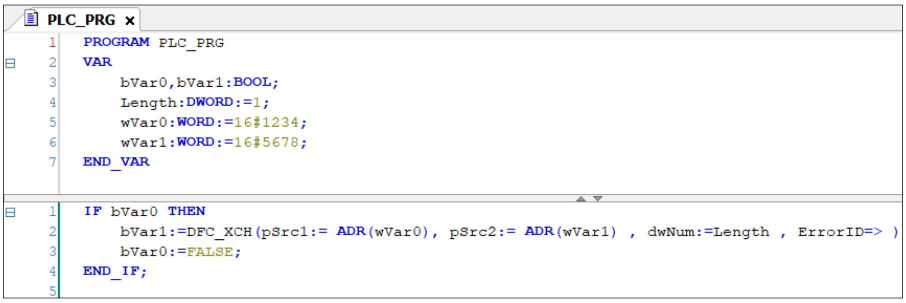
Name	Function	Data Type	Output Range (Default value)
DFC_XCH	Execution result (Return type)	BOOL	True/False(False)
ErrorID	Error code	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

- **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

- **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.

Chapter 2 Comparison Instructions

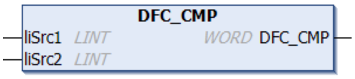
2.1 DFC_CMP

- **Supported Products**

  - AX series

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**

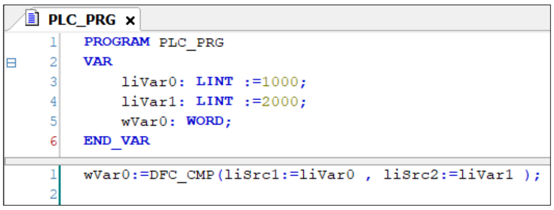
Name	Function	Data Type	Output Range (Default value)
DFC_CMP	Execution result (Return type)	WORD	1:liSrc1 = liSrc2 2:liSrc1 < liSrc2 3:liSrc1 > liSrc2 (0)

- **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.



```
PROGRAM PLC_PRG
VAR
    liVar0: LINT :=1000;
    liVar1: LINT :=2000;
    wVar0: WORD;
END_VAR
```

```
wVar0:=DFC_CMP(liSrc1:=liVar0 , liSrc2:=liVar1 );
```

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

- **Library**

  - DL_Comparison.library

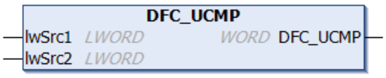
2.2 FC_UCMP

- **Supported Products**

  - AX Series

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**

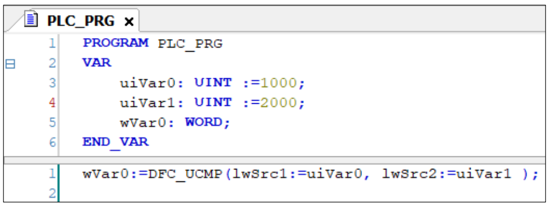
Name	Function	Data Type	Output Range (Default Value)
DFC_UCMP	Execution result (Return type)	WORD	1:lwSrc1 = lwSrc2 2:lwSrc1 < lwSrc2 3:lwSrc1 > lwSrc2 (0)

- **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.



```
PROGRAM PLC_PRG
VAR
    uiVar0: UINT :=1000;
    uiVar1: UINT :=2000;
    wVar0: WORD;
END_VAR
```

```
wVar0:=DFC_UCMP(lwSrc1:=uiVar0, lwSrc2:=uiVar1 );
```

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

- **Library**
  - DL_Comparison.library

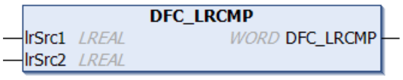
2.3 DFC_LRCMP

- **Supported Products**

  - AX Series

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**

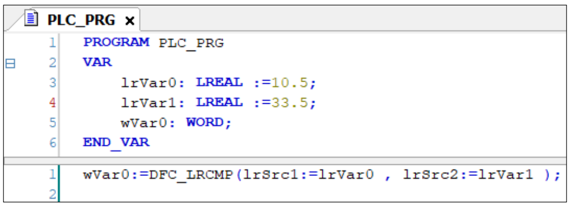
Name	Function	Data Type	Output Range (Default Value)
DFC_LRCMP	Execution result (Return type)	WORD	1:lrSrc1 = lrSrc2 2:lrSrc1 < lrSrc2 3:lrSrc1 > lrSrc2 (0)

- **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.



```
PROGRAM PLC_PRG
VAR
    lrVar0: LREAL :=10.5;
    lrVar1: LREAL :=33.5;
    wVar0: WORD;
END_VAR
```

```
wVar0:=DFC_LRCMP(lrSrc1:=lrVar0 , lrSrc2:=lrVar1 );
```

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

- **Library**

    - DL_Comparison.library

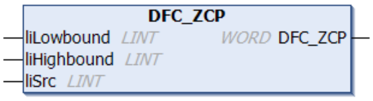
2.4 DFC_ZCP

- **Supported Products**

  - AX Series

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

FB/FC	Instruction	Graphic Expression	ST Language

FC	DFC_ZCP	 DFC_ZCP
		liLowbound LINT
		WORD DFC_ZCP
		liHighbound LINT
		liSrc LINT
			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**

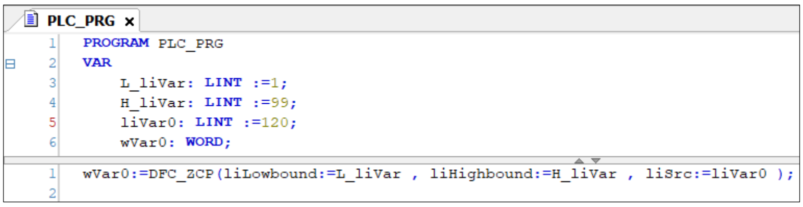
Name	Function	Data Type	Output Range (Default Value)
DFC_ZCP	Execution result (Return type)	WORD	1:liSrc < Lower value 2: Lower value < liSrc < Upper value 3:liSrc > Upper value (0)

- **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.



```
PROGRAM PLC_PRG
VAR
    L_liVar: LINT :=1;
    H_liVar: LINT :=99;
    liVar0: LINT :=120;
    wVar0: WORD;
```

```
wVar0:=DFC_ZCP(liLowbound:=L_liVar , liHighbound:=H_liVar , liSrc:=liVar0 );
```

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

- **Library**

  - DL_Comparison.library

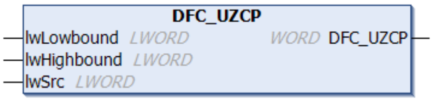
2.5 DFC_UZCP

- **Supported Products**

  - AX Series

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

FB/FC	Instruction	Graphic Expression	ST Language

FC	DFC_UZCP	 DFC_UZCP
		lwLowbound LWORD
		WORD DFC_UZCP
		lwHighbound LWORD
		lwSrc LWORD
			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 USINT, UINT, UDINT, ULINT, BYTE, WORD, DWORD, and LWORD can be used for inputs.

- **Output**

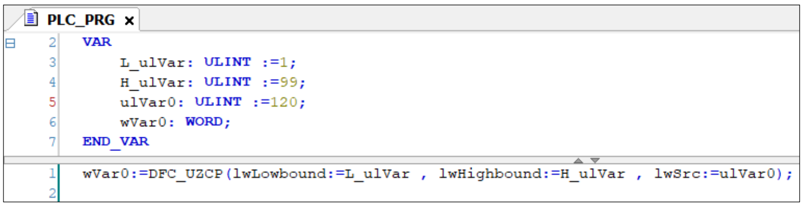
Name	Function	Data Type	Output Range (Default Value)
DFC_UZCP	Execution result (Return type)	WORD	1:lwSrc < Lower value 2: Lower value < lwSrc < Upper value 3:lwSrc > Upper value (0)

- **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.



```
VAR
    L_ulVar: ULINT :=1;
    H_ulVar: ULINT :=99;
    ulVar0: ULINT :=120;
    wVar0: WORD;
END_VAR
```

```
wVar0:=DFC_UZCP(lwLowbound:=L_ulVar , lwHighbound:=H_ulVar , lwSrc:=ulVar0);
```

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

- **Library**

  - DL_Comparison.library

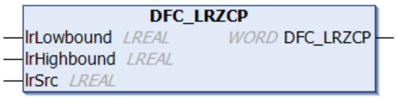
2.6 DFC_LRZCP

- **Supported Products**

  - AX Series

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

FB/FC	Instruction	Graphic Expression	ST Language

FC	DFC_LRZCP	 DFC_LRZCP
		lrLowbound LREAL
		lrHighbound LREAL
		lrSrc LREAL
		WORD 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**

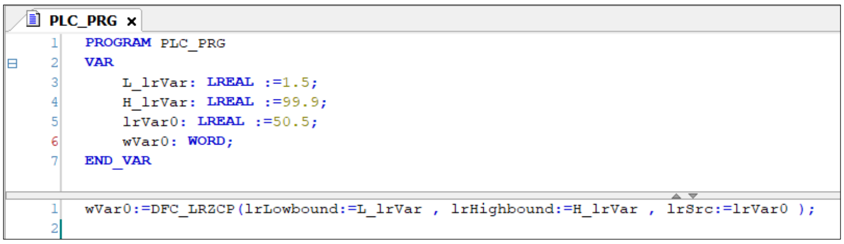
Name	Function	Data Type	Output Range (Default Value)
DFC_LRZCP	Execution result (Return type)	WORD	1:lrSrc < Lower value 2: Lower value < lrSrc < Upper value 3:lrSrc > Upper value (0)

- **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.



```
PROGRAM PLC_PRG
VAR
    L_lrVar: LREAL :=1.5;
    H_lrVar: LREAL :=99.9;
    lrVar0: LREAL :=50.5;
    wVar0: WORD;
END_VAR
```

```
wVar0:=DFC_LRZCP(lrLowbound:=L_lrVar , lrHighbound:=H_lrVar , lrSrc:=lrVar0 );
```

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.

- **Library**

  - DL_Comparison.library

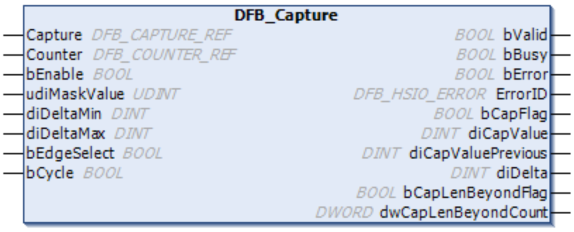
Chapter 3 Timers and Counters Instructions

3.1 DFB_Capture

- **Supported Products**
  - AX-308E, AX-364E, AX-324N, AX-332E

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
		Capture DFB_CAPTURE_REF
		Counter DFB_COUNTER_REF
		bEnable BOOL
		udiMaskValue UDINT
		diDeltaMin DINT
		diDeltaMax DINT
		bEdgeSelect BOOL
		bCycle BOOL
		BOOL bValid
		BOOL bBusy
		BOOL bError
		DFB_HSIO_ERROR ErrorID
		BOOL bCapFlag
		DINT diCapValue
		DINT diCapValuePrevious
		DINT diDelta
		BOOL bCapLenBeyondFlag
		DWORD dwCapLenBeyondCount
			DFB_Capture_instance(
			Capture :=,
			Counter :=,
			bEnable :=,
			uiMaskValue :=,
			diDeltaMin :=,
			diDeltaMax :=,
			bEdgeSelect :=,
			bCycle :=,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
bEdgeSelect[*3]	Define the rising edge or falling edge capture	BOOL	True/False (False)	When b Eable is rising edge, update the parameter of bEdgeSelect

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bCycle[*3]	Define whether capture values continuously	BOOL	True/False (False)	When b Eable is rising edge, update the parameter of bCycle

***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.
3. Only supports AX-332E.

- **Output**

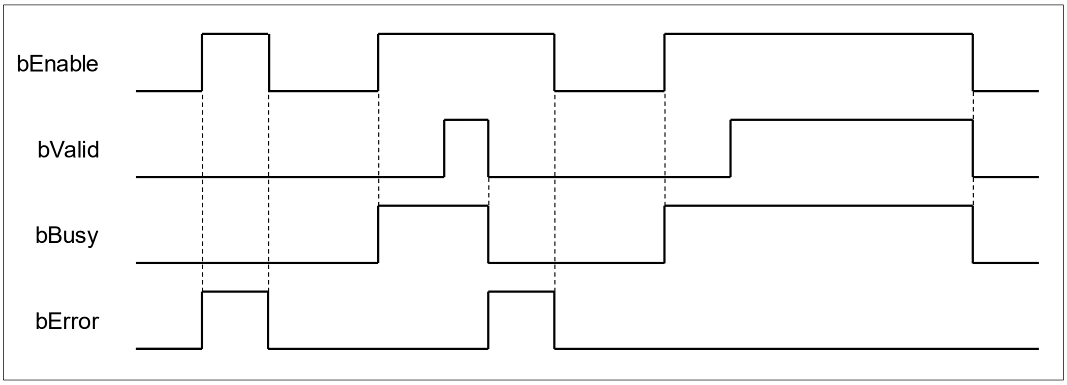
Name	Function	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*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

- **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**

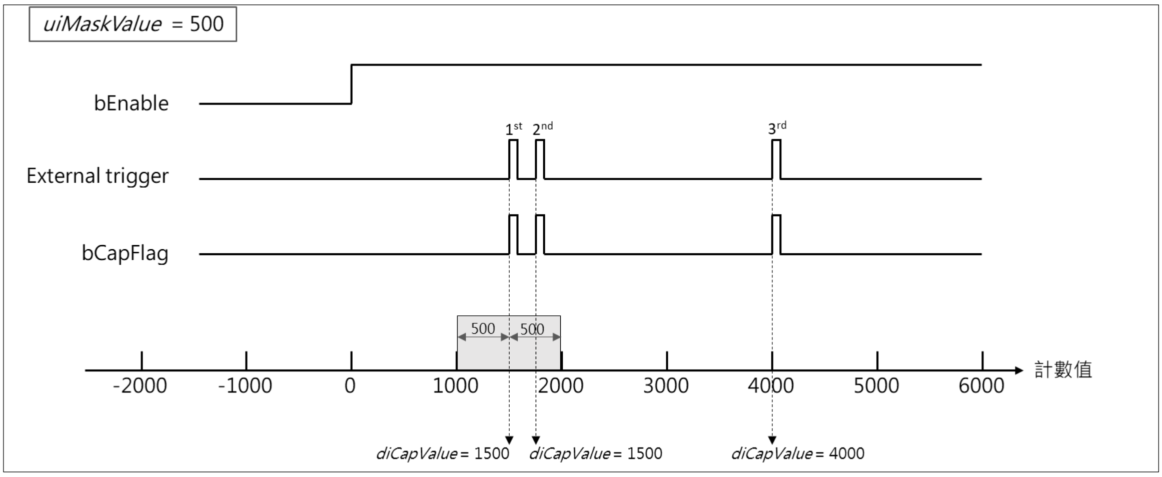
  - **bEdgeSelect**

    This funxtion is only supported in AX-3 firmware version v1.0.5.0 and later and not supported in AX-332.

  - **uiMaskValue**

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

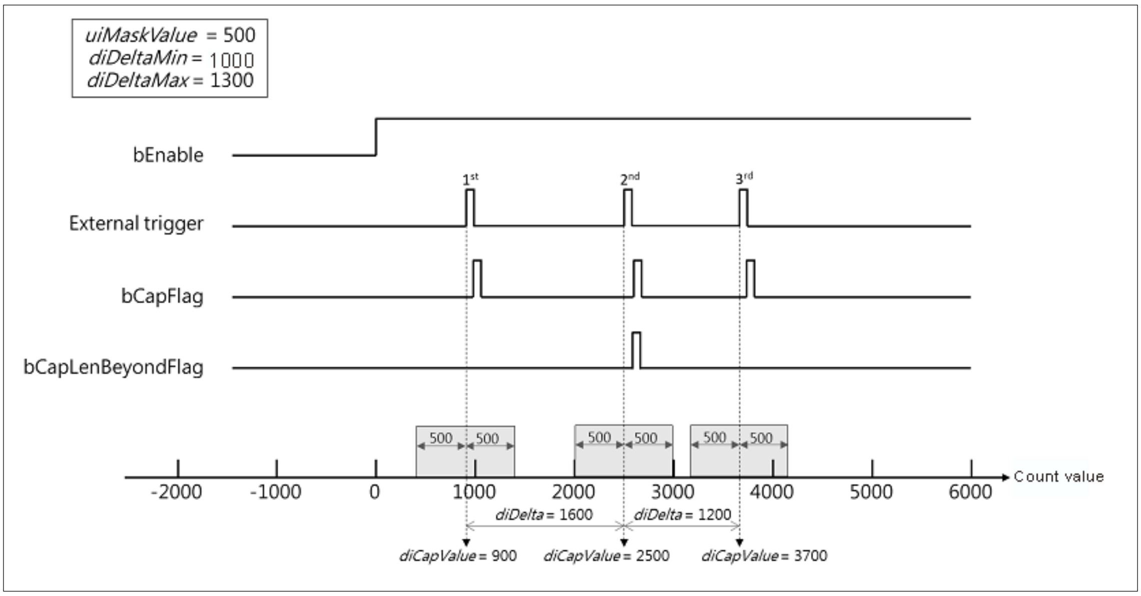
    1. Set uiMaskValue to 500 and bEnable to True, then Capture function is enabled. At the same time, the output bValid is True and the first captured value would be the center value of the mask range. In addition, the next capture action will be invalid if the next captured value is within the mask range.
    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 from 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.

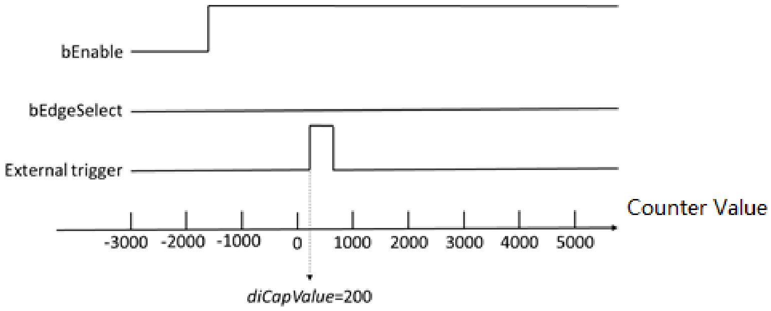


- **bEdgeSelect**

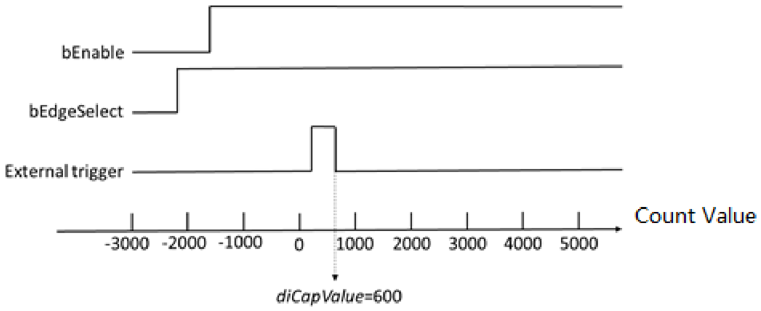
EdgeSelect defines whether the DI signal is the rising edge triggered or the falling-edge triggered to capture the signal. This function is only applicable to AX-332E models.

1. When EdgeSelect=False, the capture is triggered when DI generates the rising edge signal, and when EdgeSelect=True, the capture is triggered when DI generates the fallingr edge signal.
2. Refer to the below diagram for the explanation of these inputs and outputs:

When EdgeSelect=False:



When EdgeSelect+True

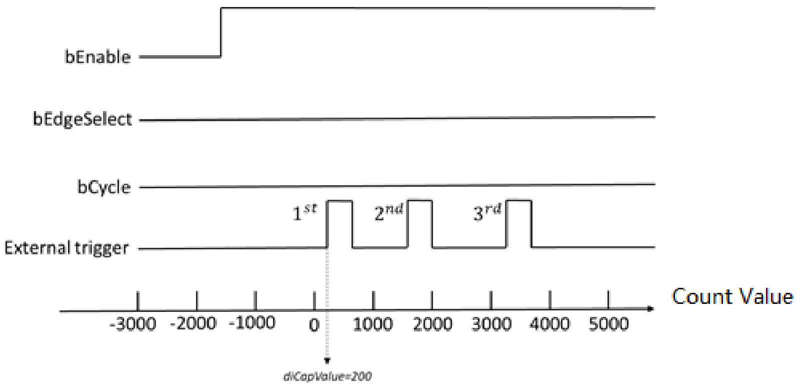


- **bCycle**

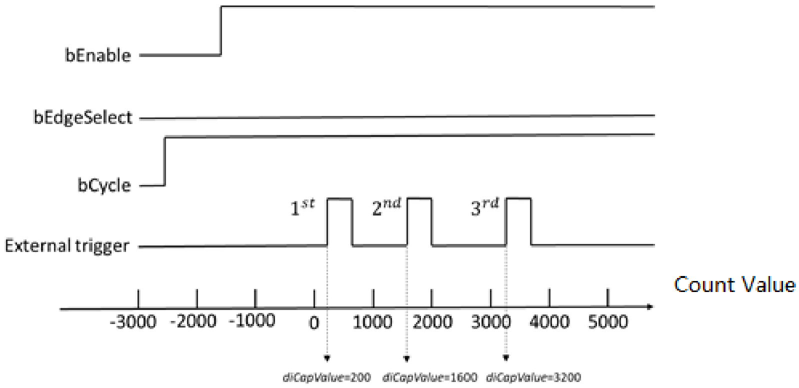
  The Cycle setting determines whether the Capture function block will capture values continuously. This function is only applicable to the AX-332EI.

  1. When Cycle=False, the Capture function block can only capture the value once triggered by DI. When Cycle=True, the Capture function block can capture multiple values for multiple times.
  2. Refer to the below diagram for the explanation of these inputs and outputs:

     When Cycle=False



     When Cycle=True



  3. Only support AX-332E v1.0.4.2 or later.

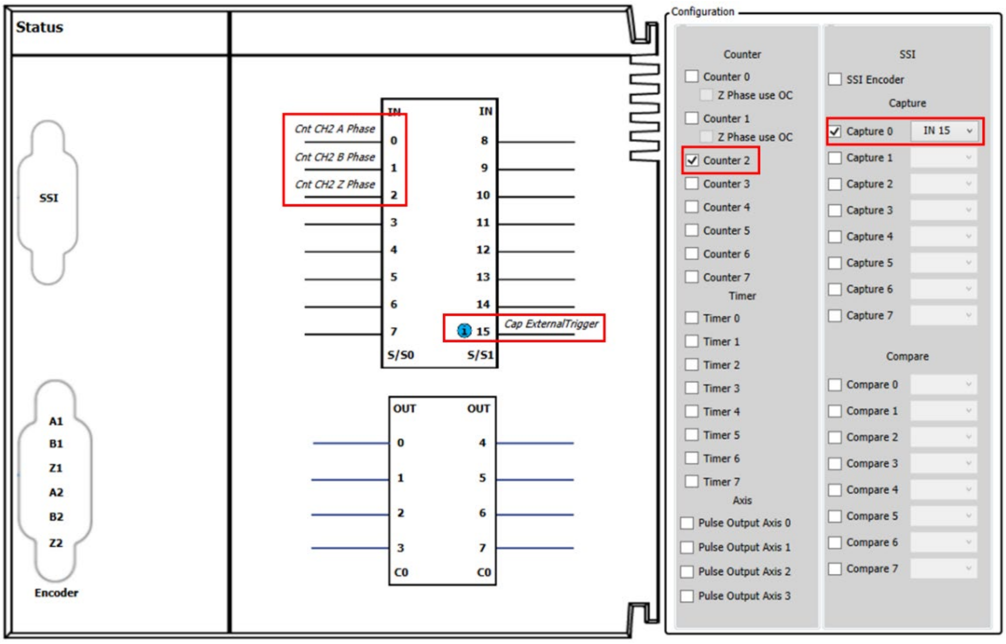
- **Troubleshooting**

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 in AX-308E 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).



Configuration

Counter

□ Counter 0
□ Z Phase use OC
□ Counter 1
□ Z Phase use OC
☑ Counter 2
□ Counter 3
□ Counter 4
□ Counter 5
□ Counter 6
□ Counter 7

Timer

□ Timer 0
□ Timer 1
□ Timer 2
□ Timer 3
□ Timer 4
□ Timer 5
□ Timer 6
□ Timer 7

Axis

□ Pulse Output Axis 0
□ Pulse Output Axis 1
□ Pulse Output Axis 2
□ Pulse Output Axis 3

SSI

□ SSI Encoder

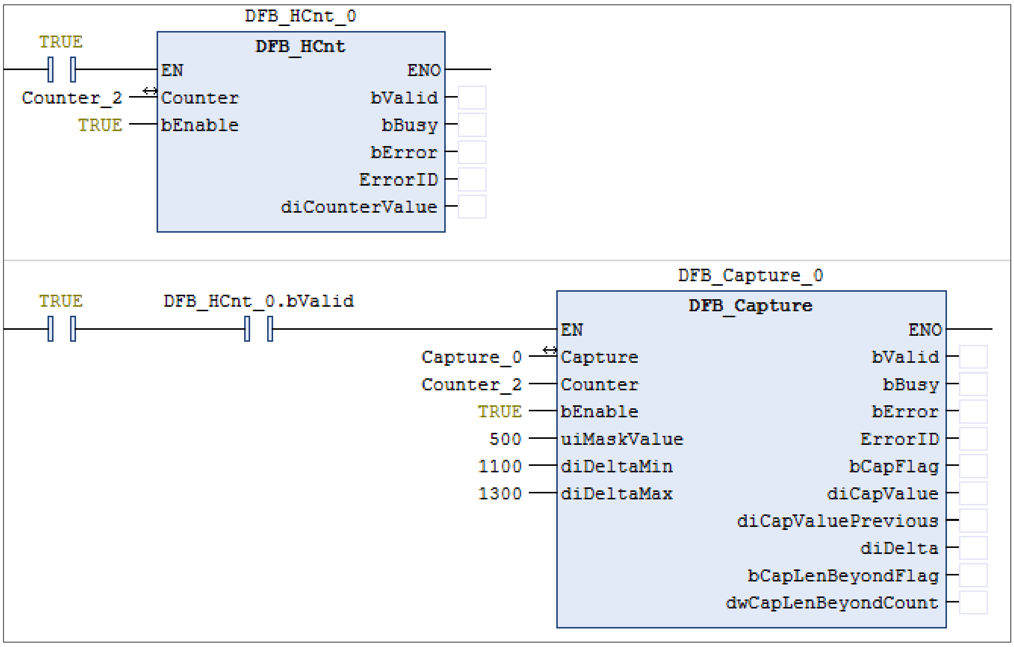
Capture

☑ Capture 0 IN 15
□ Capture 1
□ Capture 2
□ Capture 3
□ Capture 4
□ Capture 5
□ Capture 6
□ Capture 7

Compare

□ Compare 0
□ Compare 1
□ Compare 2
□ Compare 3
□ Compare 4
□ Compare 5
□ Compare 6
□ Compare 7

2. Enable 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.



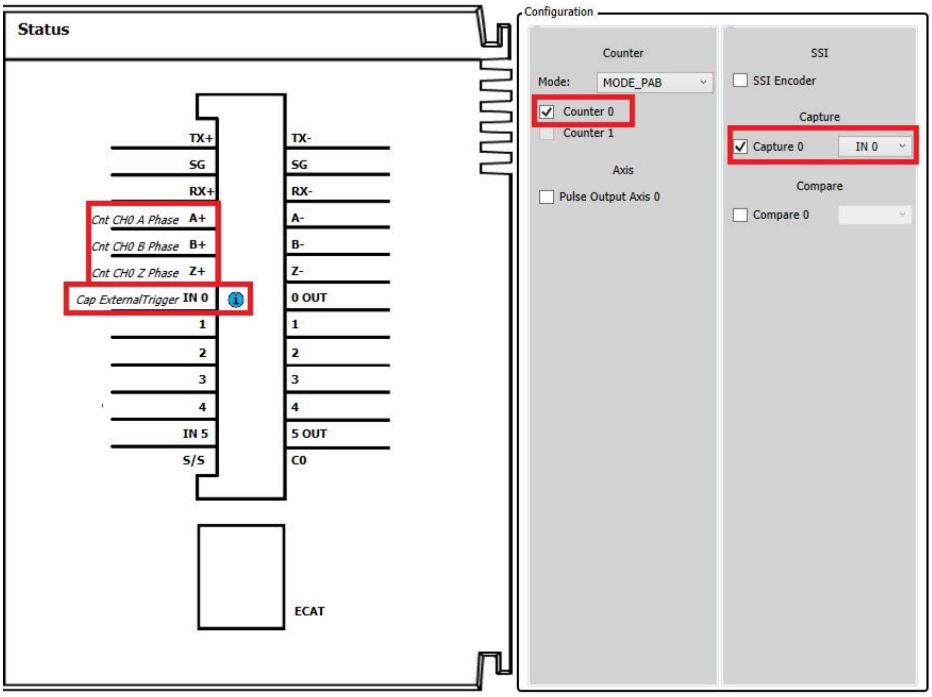
Counter_2 — Counter
TRUE — bEnable

Capture_0 — Capture
Counter_2 — Counter
TRUE — bEnable
500 — uiMaskValue
1100 — diDeltaMin
1300 — diDeltaMax

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

- This example uses DFB_HCnt and DFB_Capture in AX-308E to capture the high-speed counter.

   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. IN0).



Configuration

Counter

Mode: MODE_PAB

☑ Counter 0

□ Counter 1

Axis

□ Pulse Output Axis 0

SSI

□ SSI Encoder

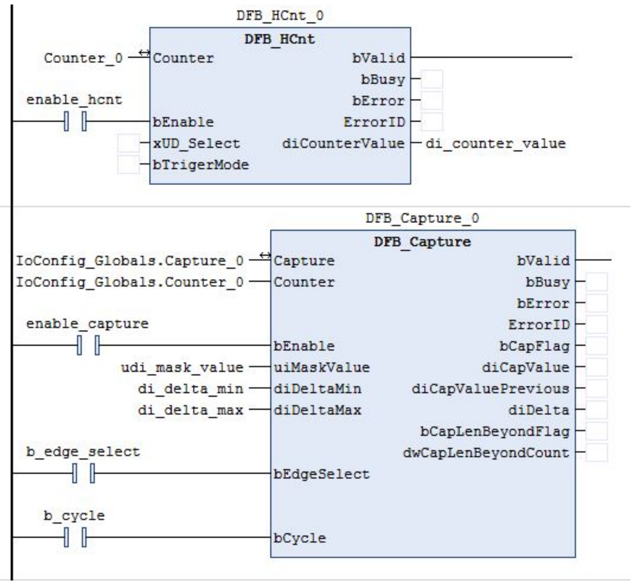
Capture

☑ Capture 0 IN 0

Compare

□ Compare 0

2. Enable 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 (IN0) being triggered.



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

- **Library**

  ▪ DL_BuiltInIO_library*

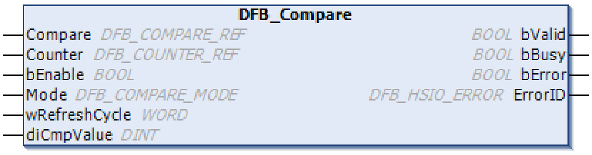
Note: From version 1.0.5.0 and later, lilbrary DL_BuiltInIO_AX3 is changed to DL_BuiltInIO_library.

3.2 DFB_Compare

- **Supported Products**
  - AX-308E, AX-364E, AX-324N, AX-332E

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
		Compare DFB_COMPARE_REF
		Counter DFB_COUNTER_REF
		bEnable BOOL
		Mode DFB_COMPARE_MODE
		wRefreshCycle WORD
		diCmpValue DINT
		BOOL bValid
		BOOL bBusy
		BOOL bError
		DFB_HSIO_ERROR ErrorID
			DFB_Compare_instance(
			Compare :=,
			Counter :=,
			bEnable :=,
			Mode :=,
			OutputAction :=,
			wRefreshCycle :=,
			diCmpValue :=,
			diTablePosition :=,
			uiTableSize :=,
			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 to True and bBusy is False.

OutputAction*4	Define the output mode.	DFB_ COMPARE_ OUTPUT_ACTION	1: SET 2: RESET (SET)	When bEnable is rising edge 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 to True and bBusy is False.
diCmpValue	Specifies the compare value	DINT	Positive number, negative number or 0(0)	When bEnable shifts to True and bBusy is False.

Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
diTablePosition[*3]	Specify the compare value array	ARRAY OF DINT		When bEnable is rising edge and bBusy is False.
diTableSize[*3]	Specify the compare value array scope	DINT	Positive number 0 to 256 (0)	When bEnable is rising edge 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).
3. Except for AX-332E, other AX-3 series with DL_BuiltInIO_AX3 version V1.0.4.2 or later also supports this function.
4. This function is applicable to AX-332E. This function block cannot be executed in the Ethercat Task, and it will affect the synchronization cycle of other Ethercat devices.
5. In AX-332E, diCmpValue will not refer to the value of this variable for comparison.

- **Output**

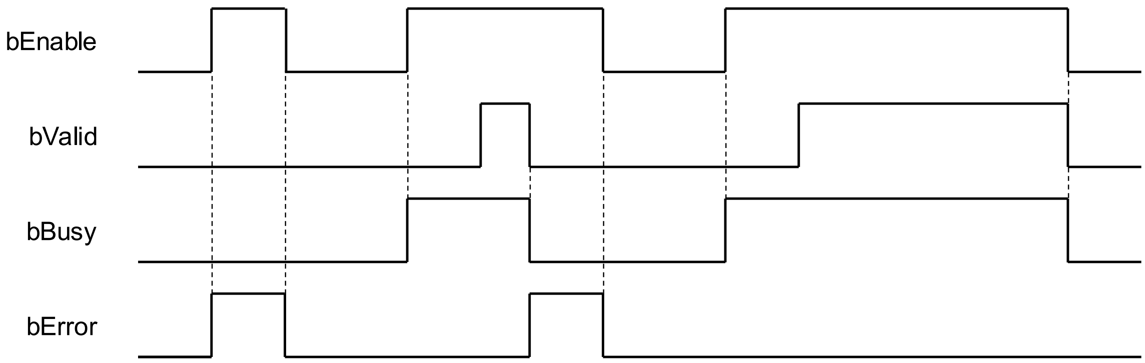
Name	Function	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*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

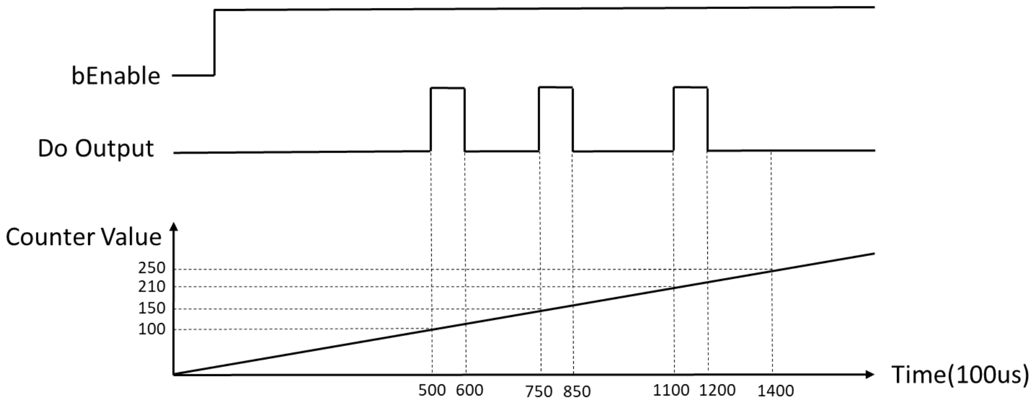
Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Compare	Reference to the source of high-speed comparator.	DFB_COMPARE_REF (FB)*	DFB_COMPARE_REF (Cannot be null.)	When bEnable shifts to True and bBusy is False.

***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. Cannot run this function block in Ethercat Task in AX-332, or it will affect update cycle of other device in Ethercat.
2. 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.
3. When bValid output of DFB_Compare is True, the comparator would continue to compare on the high-speed counter 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.
4. 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.
5. diTablePosition is the numerical array to be compared, and diTableSize is the number of values to be compared. For example, the size of the diTablePosition array is 256, if diTableSize=100 is set, the 0-99 elements of the array will be set as the values to be compared. The timing diagram is as follows:



diTablePosition[0] = 100 wRefreshCycle = 100 (100us)

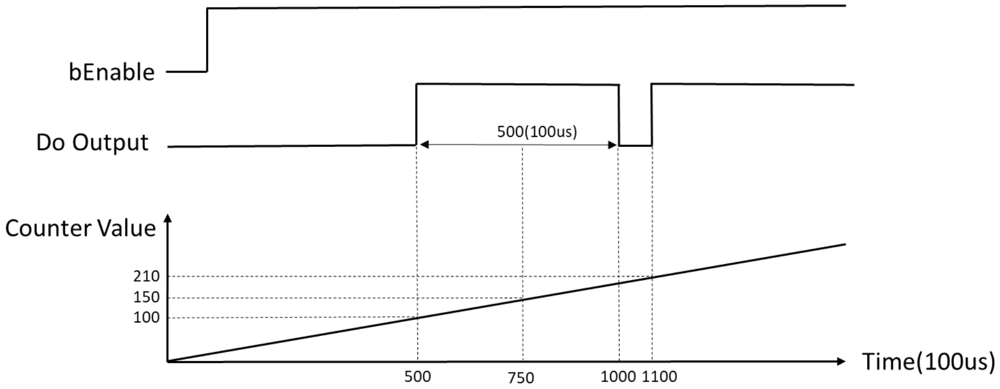
diTablePosition[1] = 150 diTableSize=3

diTablePosition[2] = 210 Mode=0(Equal)

diTablePosition[3] = 250

If comparing multiple value at the same time, the following two situations will affect the ouput of Do.

Case1: In the process of Do output, other value of diTablezpodtion is compared, the value will not generate Do output. The output will be the Do output of the current value. Refer the following diagram for the timing and parameter:

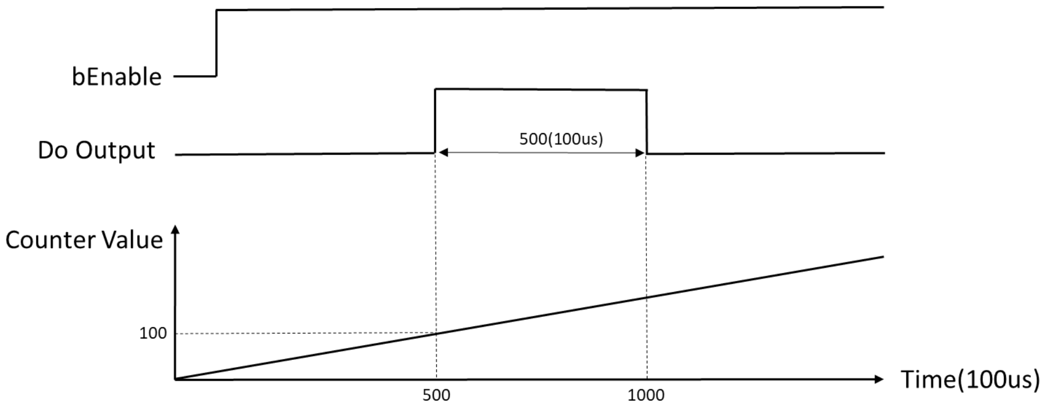


diTablePosition[0] = 100 wRefreshCycle = 500 (100us)

diTablePosition[1] = 150 diTableSize=3

diTablePosition[2] = 210 Mode=0(Equal)

Case2: When comparing multiple values of diTablePosion at the same time, the defined Do output time is the time of of one cycle of wRefreshCycle. Refer the following diagram for the timing and parameter:



diTablePosition[0] = 100 wRefreshCycle = 500 (100us)

diTablePosition[1] = 100 diTableSize=3

diTablePosition[2] = 100 Mode=0(Equal)

OutputAction is an output variable that define the action of the output device once the comparison condition is met. When it is set to SET, the ouput device will be set to high level. When it is set to RESET, the output device will be set to low level.

Only support AX-332E version 1.0.4.2 or later.

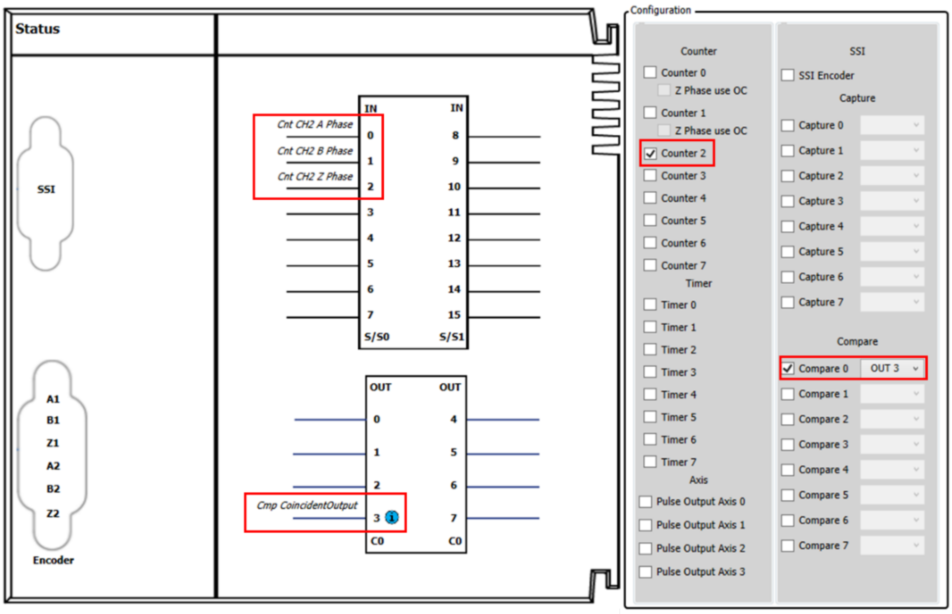
- **Troubleshooting**

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**

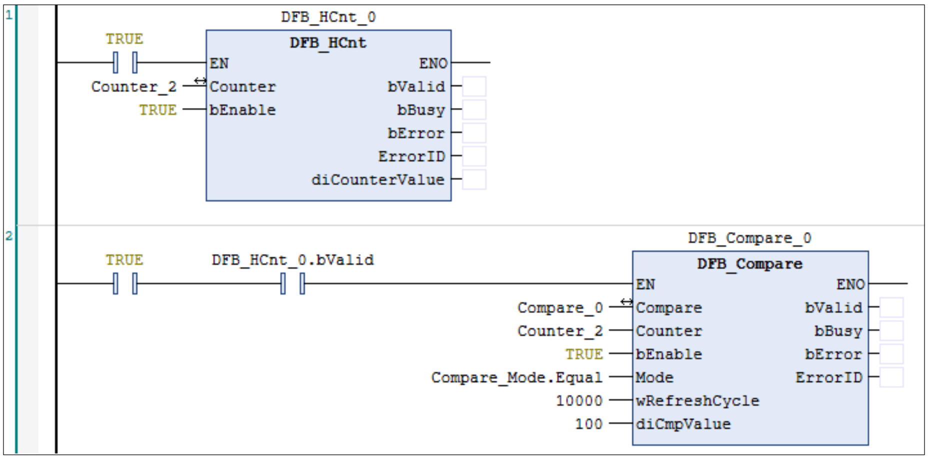
- This example uses DFB_HCnt and DFB_Compare in AX-308E to perform the Compare function.

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

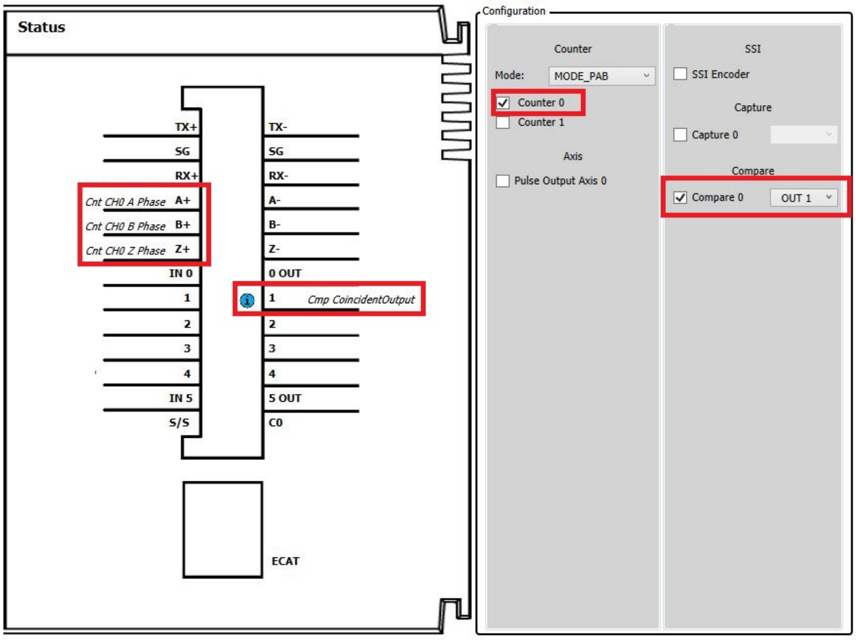
shown in follows. At the same time, the output device (OUT3) will output the signal once the comparison condition is fulfilled (DFB_HCnt_0.diCounterValue = DFB_Compare_0.diCmpValue).



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

- This example uses DFB_HCnt and DFB_Compare in AX-332E 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. OUT1).



Mode: MODE_PAB

☑ Counter 0

□ Counter 1

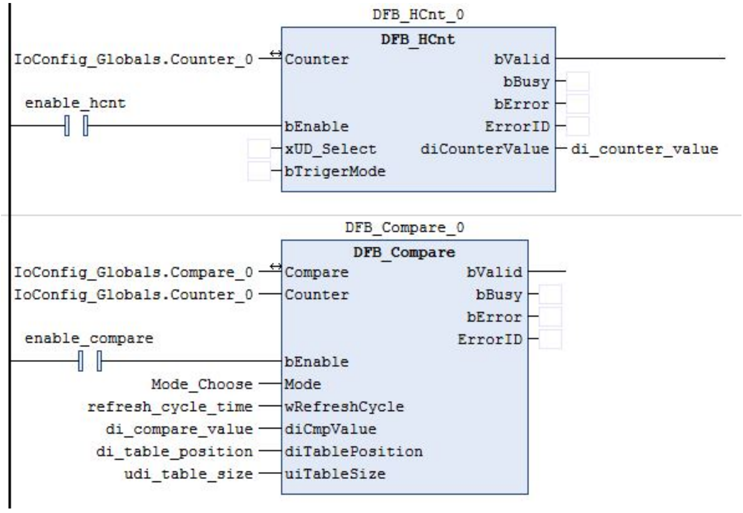
□ Pulse Output Axis 0

□ SSI Encoder

□ Capture 0

☑ Compare 0 OUT 1

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 (OUT1) 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.

- **Library**

  - DL_BuiltInIO_AX3.library

Note: From version 1.0.5.0 and later, lilbrary DL_BuiltInIO_AX3 is changed to DL_BuiltInIO_.library.

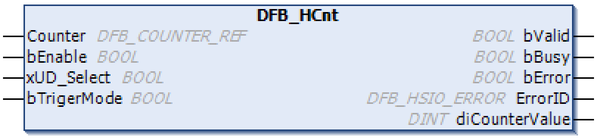
3.3 DFB_HCnt

- **Supported Products**

  - AX-308E, AX-364E, AX-324N, AX-332E

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

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_HCnt	 DFB_HCnt
		Counter DFB_COUNTER_REF
		bEnable BOOL
		xUD_Select BOOL
		bTrigerMode BOOL
		BOOL bValid
		BOOL bBusy
		BOOL bError
		DFB_HSIO_ERROR ErrorID
		DINT diCounterValue
			DFB_HCnt_instance(
			Counter :=,
			bEnable :=,
			bTrigerMode :=,
			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)	-
xUD_Select[*2]	Define the high-speed counter to count up or down	BOOL	True/False (False)	When bEnable is rising edge, update parameter of xUD_Select.
bTriggerMode[*2]	Define the capture is triggered by rising edge or falling edge	BOOL	True/False (False)	When bEnable is rising edge, update parameter of tiggerMode.

***Note:**

1. DFB_HSIO_ERROR: Enumeration (Enum)

2. Only applicable to AX-332.

- **Output**

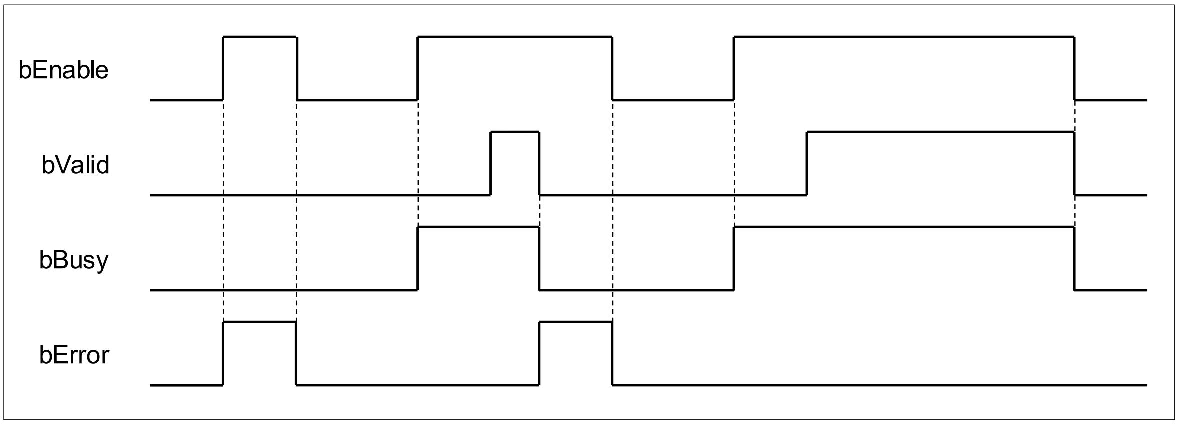
Name	Function	Data Type	Output Range (Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*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

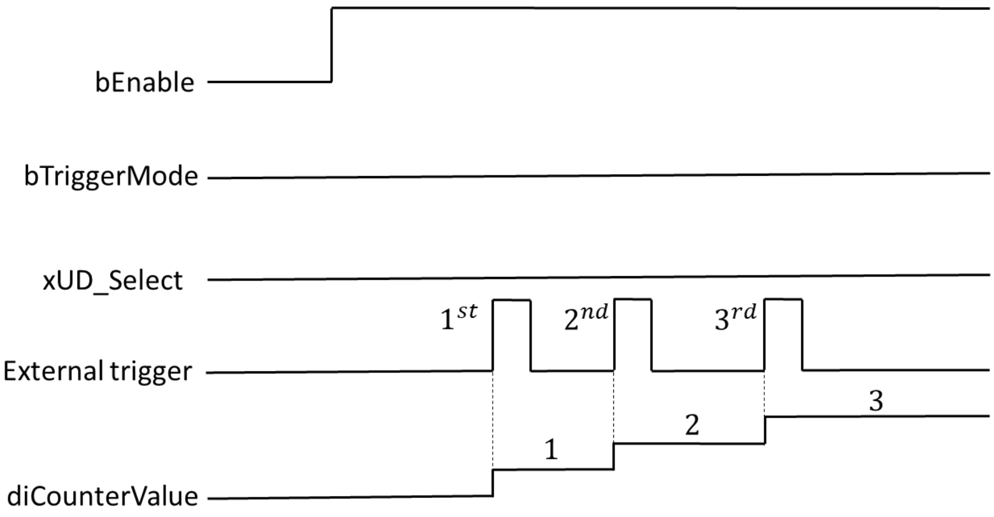
Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
Counter	Reference to the source of specified high-speed counter.	DFB_COUNTER_REF (FB)*	DFB_COUNTER_REF (Cannot be null.)	When bEnable shifts to True and Busy is False.

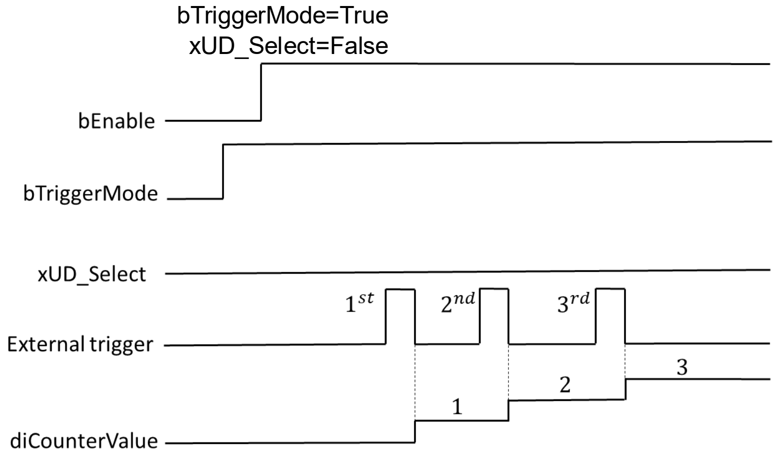
***Note:** DFB_Counter_REF (FB): As the I/O interface of the high-speed counter to perform actions include parameter adjustment and the driver.

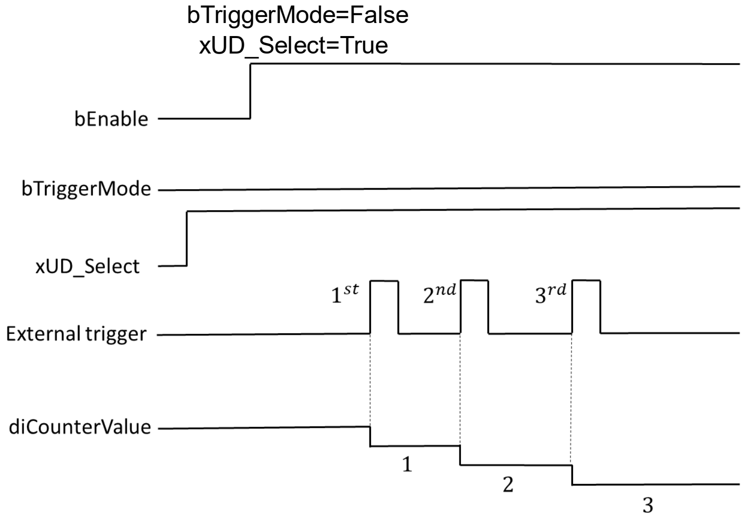
- **Function**

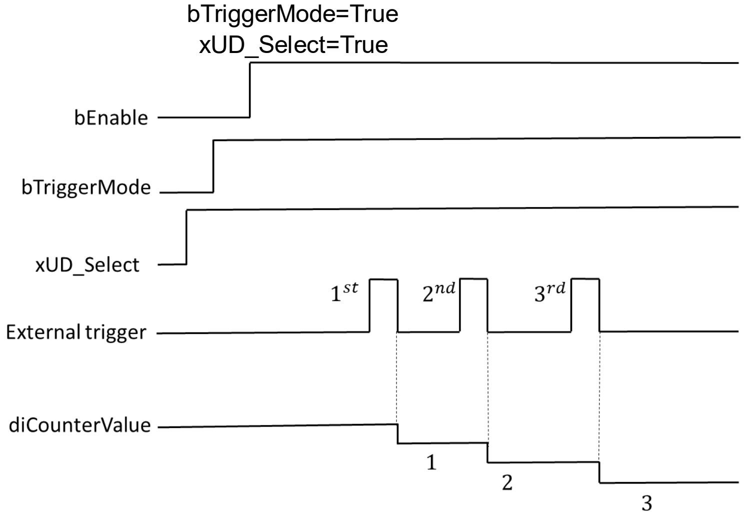
1. When the input bEnable is True, the counter would start calculating pulses to the corresponding input points based on the Counter configuration of HW IO configuration in BuiltIn IO.
2. The counter value is given through the output diCounterValue during the counting process.
3. bTriggerMode defines whether the DI signal is triggered by the rising edge or the falling edge for high-speed counting. It is only applicable to AX-332E. When EdgeSelect=False, high-speed counting is triggered when DI generates the rising edge signal; when EdgeSelect=True, high-speed counting is triggered when DI generates the falling edge signal. This function is only available when the corresponding input Counter setting mode is MODE_UD/ MODE_UDR/ MODE_UDRE/MODE_UDRE2.
4. xUD_Select defines whether to count up or count down when the external DI signal triggers counting.It is only applicable to AX-332E. When xUD_Select=False, the DI signal is triggered to count up, and when xUD_Select=True, the DI signal is triggered to count down. This function is only available when the corresponding input Counter setting mode is MODE_UD/ MODE_UDR/ MODE_UDRE.

   Refer the following timing diagram for bTriggerMode and xUD_Select:bTriggerMode=False xUD_Select=False









5. Only supports AX-332Eversion 1.0.4.2 or later.

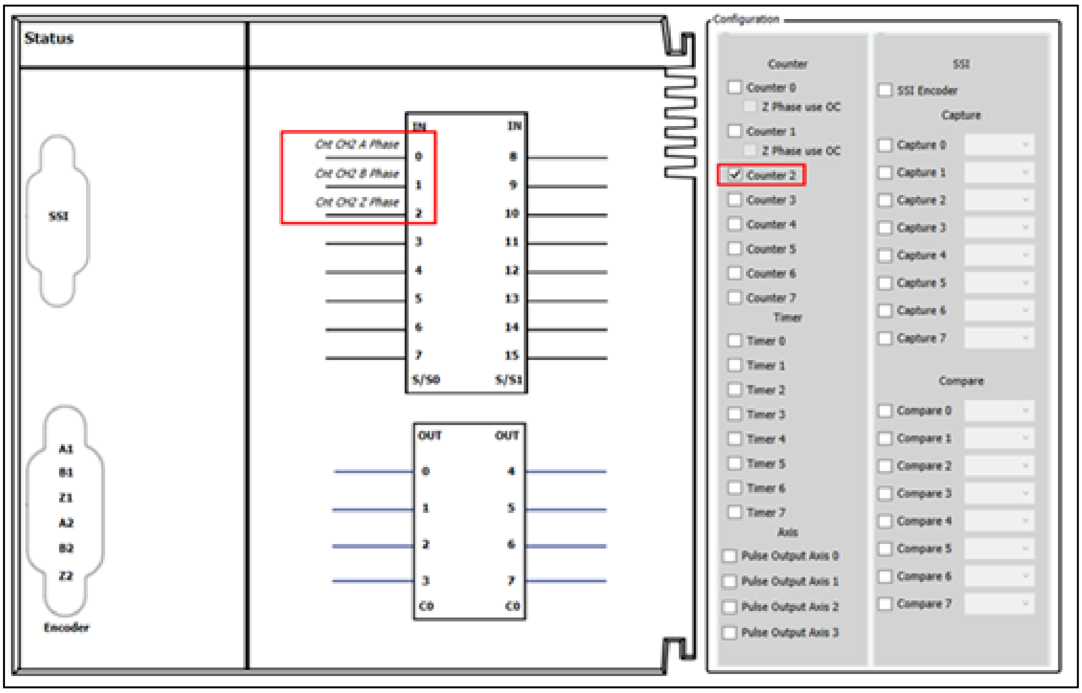
- **Troubleshooting**

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 in AX-308 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.



Configuration

Counter
□ Counter 0
□ Z Phase use OC
□ Counter 1
□ Z Phase use OC
☑ Counter 2
□ Counter 3
□ Counter 4
□ Counter 5
□ Counter 6
□ Counter 7

Timer
□ Timer 0
□ Timer 1
□ Timer 2
□ Timer 3
□ Timer 4
□ Timer 5
□ Timer 6
□ Timer 7

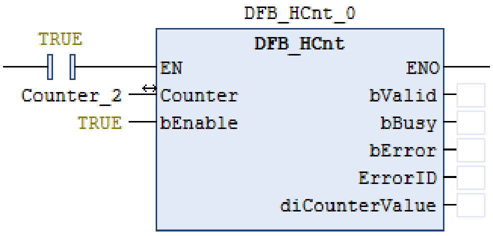
Axis
□ Pulse Output Axis 0
□ Pulse Output Axis 1
□ Pulse Output Axis 2
□ Pulse Output Axis 3

SSI
□ SSI Encoder

Capture
□ Capture 0
□ Capture 1
□ Capture 2
□ Capture 3
□ Capture 4
□ Capture 5
□ Capture 6
□ Capture 7

Compare
□ Compare 0
□ Compare 1
□ Compare 2
□ Compare 3
□ Compare 4
□ Compare 5
□ Compare 6
□ Compare 7

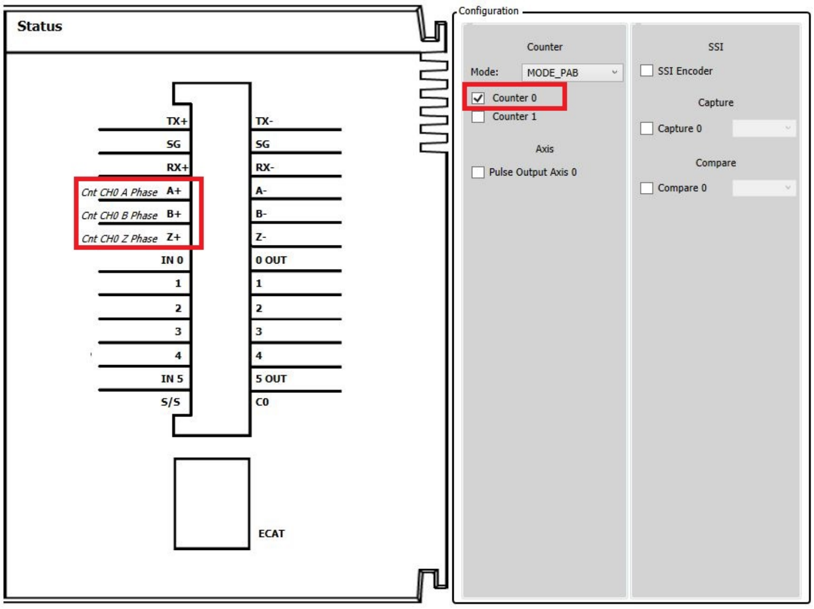
2. After using the FB DFB_HCnt in the POU to activate the high-speed counter(bEnable = True), it starts receiving and counting the pulses from the external signals(IN 0, IN 1) based on the counting mode set in Counter Configuration, then the counter value would be displayed in the output diCounterValue. 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 counter 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.

This example uses DFB_HCnt in AX-332E to perform the Count function

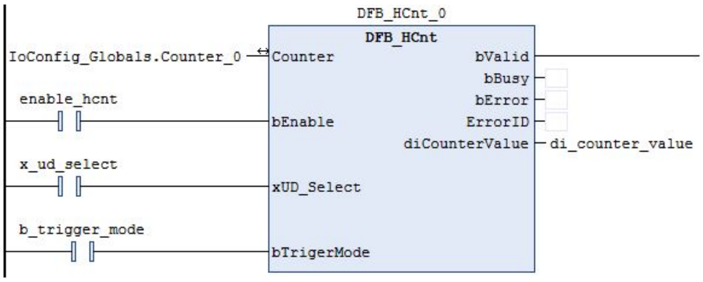
1. As the following figure shows, select a Counter (Counter 0) in Hardware IO Configuration and you will see the input points (e.g. A+A-/B+B-/C+C-)) matched to the corresponding encoder A, B, Z phase outputs, which the wiring should follows the configuration so as to perform the normal function of high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



Mode: MODE_PAB

☑ Counter 0
□ Counter 1
□ Pulse Output Axis 0
□ SSI Encoder
□ Capture 0
□ Compare 0

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(A+A-/B+B-) based on the counting mode set in Counter Configuration, then the counter 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 counter 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.

- **Library**
  - DL_BuiltInIO_AX3.library

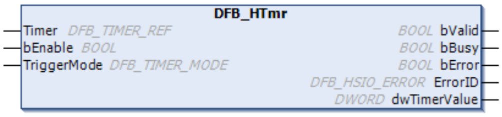
3.4 DFB_HTmr

- **Supported Products**

  - AX-308E, AX-364E, AX-324N

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

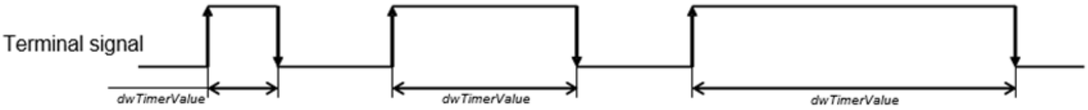
FB	DFB_HTmr	 DFB_HTmr
		Timer DFB_TIMER_REF
		bEnable BOOL
		TriggerMode DFB_TIMER_MODE
		BOOL bValid
		BOOL bBusy
		BOOL bError
		DFB_HSIO_ERROR ErrorID
		DWORD dwTimerValue
			DFB_HTmr_instance(
			Timer :=,
			bEnable :=,
			TriggerMode :=,
			bValid =>,
			bBusy =>,
			bError =>,
			ErrorID =>,
			dwTimerValue =>);

- **Input**

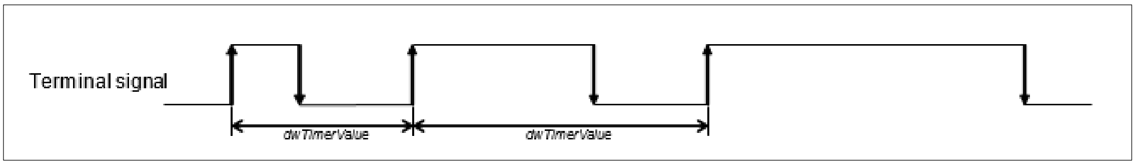
Name	Function	Data Type	Setting Value (Default value)	Timing for Updating
bEnable	Execute the instruction when bEnable changes to True.	BOOL	True/False (False)	-
TriggerMode	Timing mode settings.	DFB_TIMER_MODE *	0:UP_DOWN 1:UP_UP (UP_DOWN)	When bEnable shifts to True and bBusy is False.

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

Up-Down mode:



Up-Up mode:



- **Output**

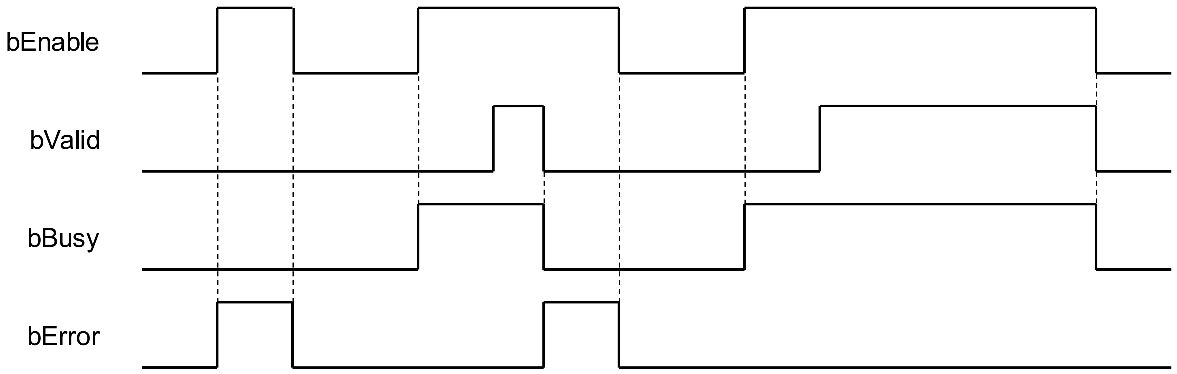
Name	Function	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
dwTimerValue	Timed value (Unit: 0.01us)	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 (Cannot 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.

- **Function**

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 counter value is given through the output dwTimerValue during the counting process.

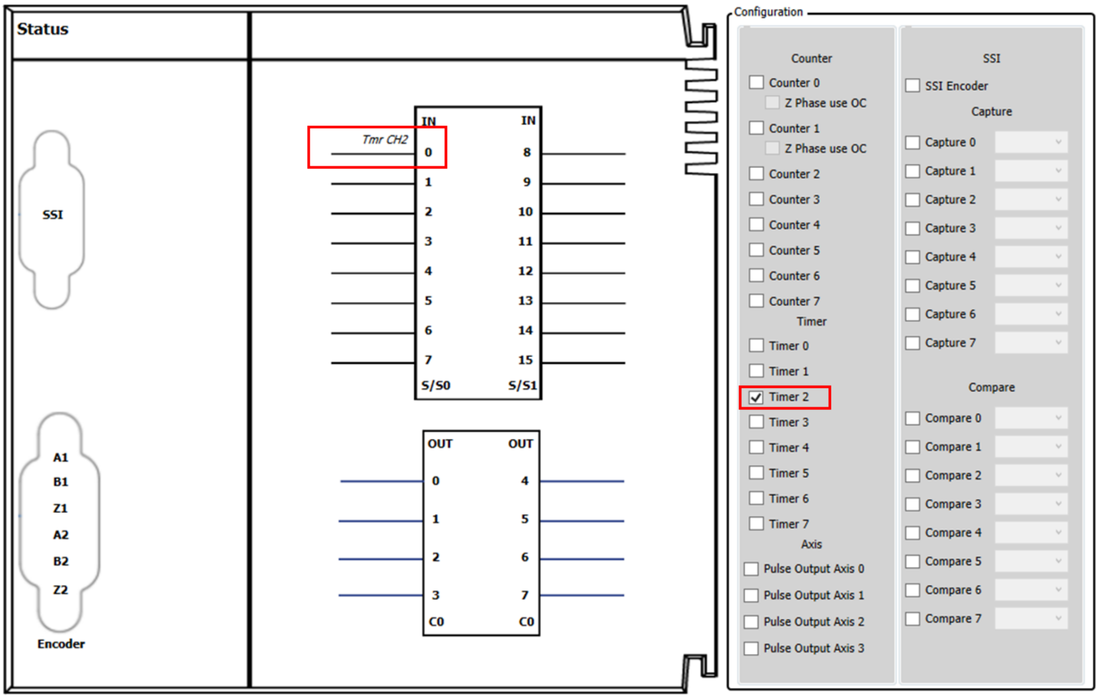
- **Troubleshooting**

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.



Configuration

Counter
□ Counter 0
□ Z Phase use OC
□ Counter 1
□ Z Phase use OC
□ Counter 2
□ Counter 3
□ Counter 4
□ Counter 5
□ Counter 6
□ Counter 7

Timer
□ Timer 0
□ Timer 1
☑ Timer 2
□ Timer 3
□ Timer 4
□ Timer 5
□ Timer 6
□ Timer 7

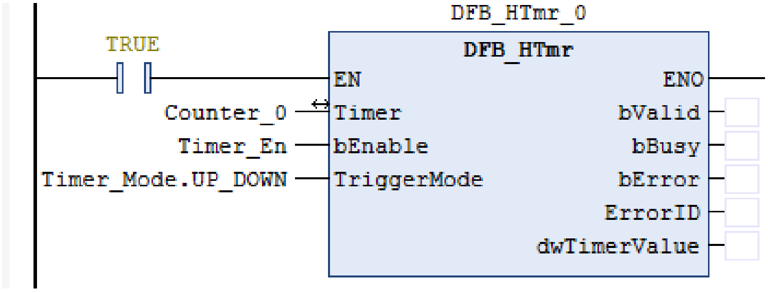
Axis
□ Pulse Output Axis 0
□ Pulse Output Axis 1
□ Pulse Output Axis 2
□ Pulse Output Axis 3

SSI
□ SSI Encoder

Capture
□ Capture 0
□ Capture 1
□ Capture 2
□ Capture 3
□ Capture 4
□ Capture 5
□ Capture 6
□ Capture 7

Compare
□ Compare 0
□ Compare 1
□ Compare 2
□ Compare 3
□ Compare 4
□ Compare 5
□ Compare 6
□ Compare 7

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.

- **Library**

  - DL_BuiltInIO_AX3.library

Note: From version 1.0.5.0 and later, lilbrary DL_BuiltInIO_AX3 is changed to DL_BuiltInIO_library.

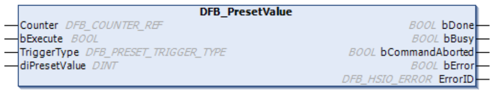
3.5 DFB_PresetValue

- **Supported Products**

  - AX-308E, AX-364E, AX-324N

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

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_PresetValue	 DFB_PresetValue
		Counter DFB_COUNTER_REF
		bExecute BOOL
		TriggerType DFB_PRESET_TRIGGER_TYPE
		diPresetValue DINT
		BOOL bDone
		BOOL bBusy
		BOOL bCommandAborted
		BOOL bError
		DFB_HSIO_ERROR ErrorID
			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 is False.
diPresetValue	The preset counter value for high speed counters.	DINT	Positive number, negative number or 0(0)	When bExecute shifts to True and bBusy is False

***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	BOOL	True/False(False)

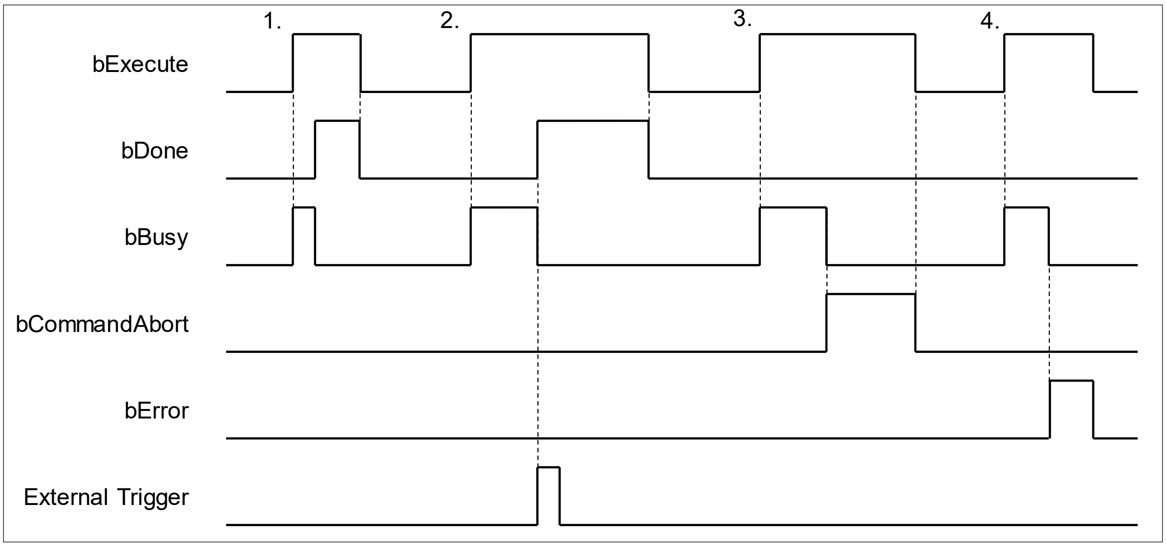
Name	Function	Data Type	Output Range(Default value)
	enabled.		
bCommandAborted	True when the instruction is aborted before it’s completed.	BOOL	True/False(False)
bError	True when an error occurs.	BOOL	True/False(False)
ErrorID	Error codes.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)

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

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bDone	● True when the counter 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 counter value would be set back to the default value right after activating the function block.

2. When TriggerType = EXTERNAL_TRIGGER, the counter value would not be set back to the default until the Z phase signal of the counter rises.

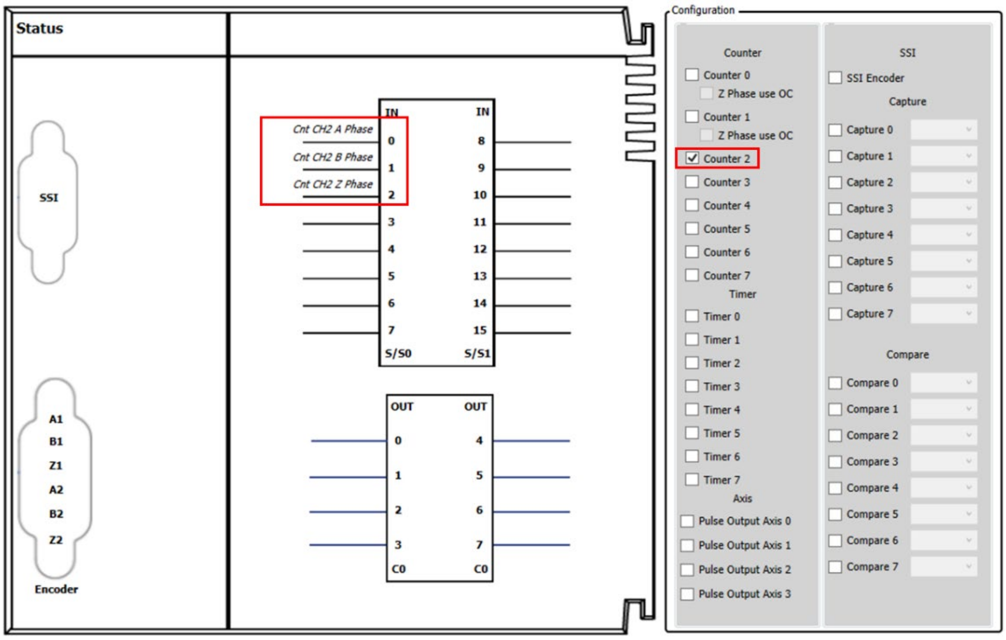
- **Troubleshooting**

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.



Configuration

Counter
□ Counter 0
□ Z Phase use OC
□ Counter 1
□ Z Phase use OC
☑ Counter 2
□ Counter 3
□ Counter 4
□ Counter 5
□ Counter 6
□ Counter 7

Timer
□ Timer 0
□ Timer 1
□ Timer 2
□ Timer 3
□ Timer 4
□ Timer 5
□ Timer 6
□ Timer 7

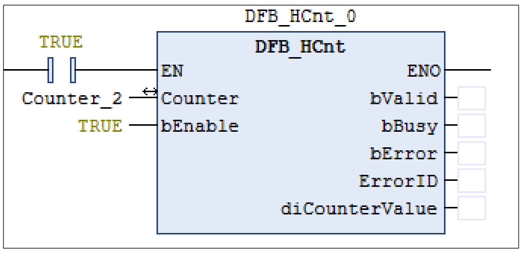
Axis
□ Pulse Output Axis 0
□ Pulse Output Axis 1
□ Pulse Output Axis 2
□ Pulse Output Axis 3

SSI
□ SSI Encoder

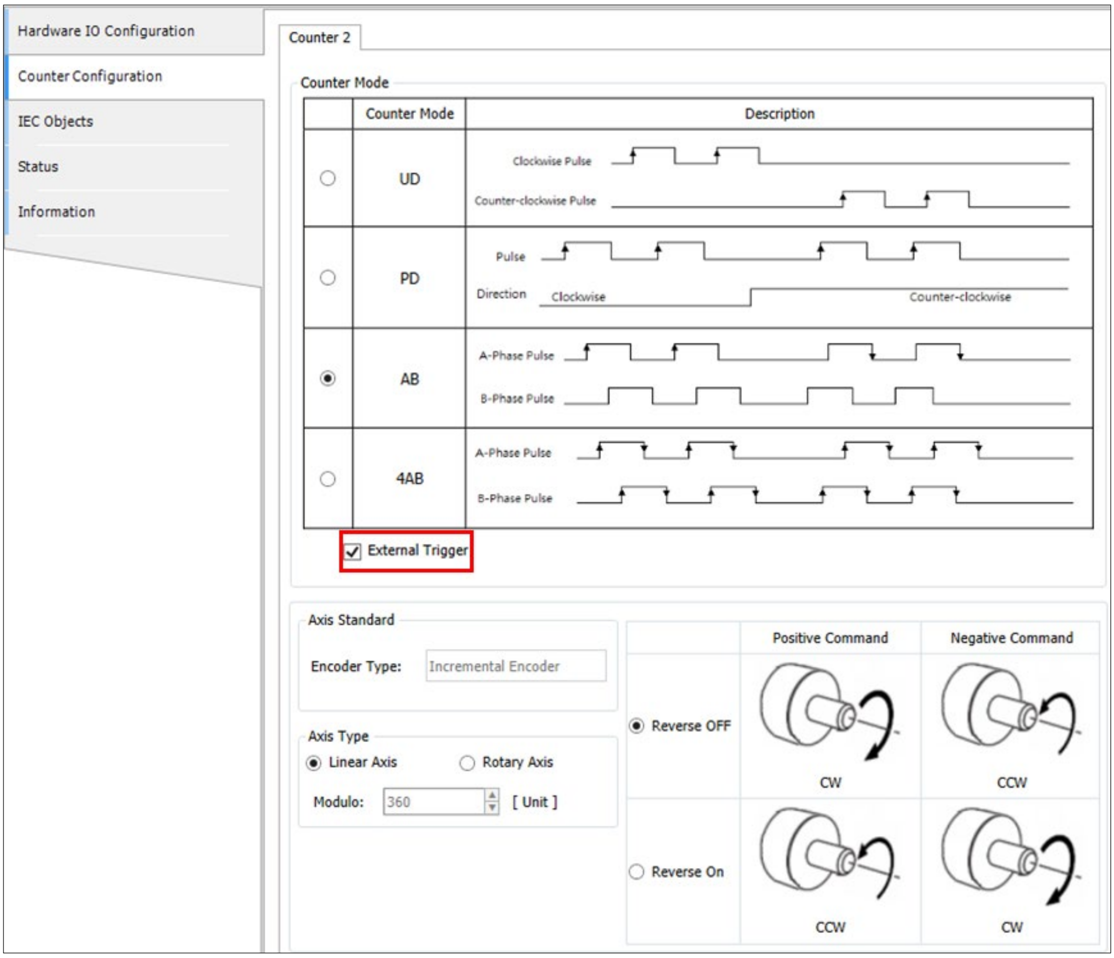
Capture
□ Capture 0
□ Capture 1
□ Capture 2
□ Capture 3
□ Capture 4
□ Capture 5
□ Capture 6
□ Capture 7

Compare
□ Compare 0
□ Compare 1
□ Compare 2
□ Compare 3
□ Compare 4
□ Compare 5
□ Compare 6
□ Compare 7

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 counter 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 counter values.



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.



Counter Mode

	Counter Mode	Description
○	UD	Clockwise Pulse Counter-clockwise Pulse
○	PD	Pulse Direction Clockwise Counter-clockwise
●	AB	A-Phase Pulse B-Phase Pulse
○	4AB	A-Phase Pulse B-Phase Pulse

☑ External Trigger

Axis Standard

Encoder Type: Incremental Encoder

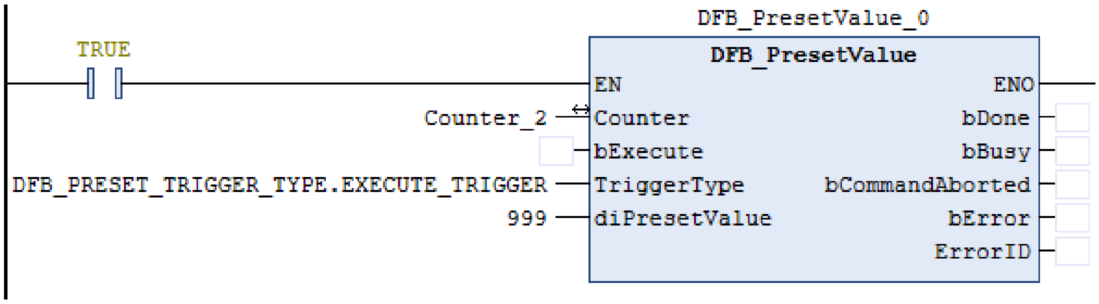
Axis Type

● Linear Axis ○ Rotary Axis

Modulo: 360 [Unit]

	Positive Command	Negative Command
● Reverse OFF	CW	CCW
○ Reverse On	CCW	CW

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 counter value being set to the default (DFB_HCnt.diCounterValue = fb), 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.

- **Library**

  - DL_BuiltInIO_AX3.library

Note: From version 1.0.5.0 and later, lilbrary DL_BuiltInIO_AX3 is changed to DL_BuiltInIO_library.

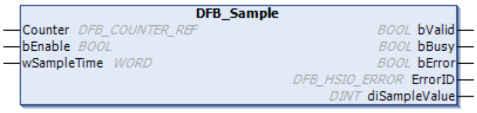
3.6 DFB_Sample

- **Supported Products**

  - AX-308E, AX-364E, AX-324N

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

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_Sample	 DFB_Sample
		Counter DFB_COUNTER_REF
		bEnable BOOL
		wSampleTime WORD
		BOOL bValid
		BOOL bBusy
		BOOL bError
		DFB_HSIO_ERROR ErrorID
		DINT diSampleValue
			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**

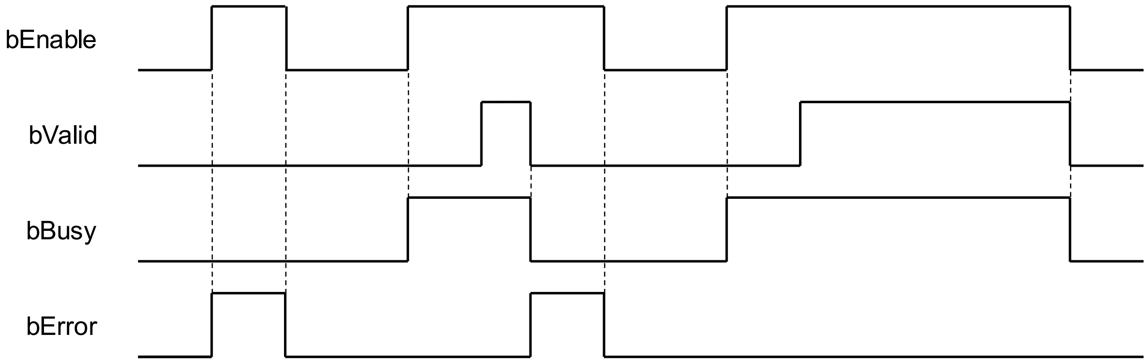
Name	Function	Data Type	Output Range(Default value)
bValid	True when the output value is valid.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
diSampleValue	Increasing number of the counter 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**



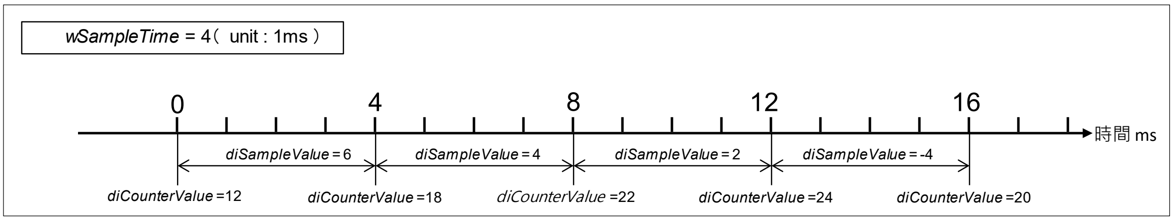
- **In/ Outs**

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.

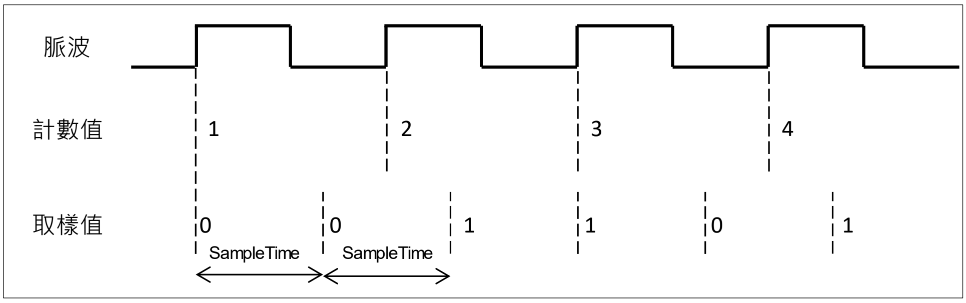
***Note:** DFB_COUNTER_REF (FB): As the I/O interface of the high-speed counter to perform actions include parameter adjustment and the driver.

- **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.



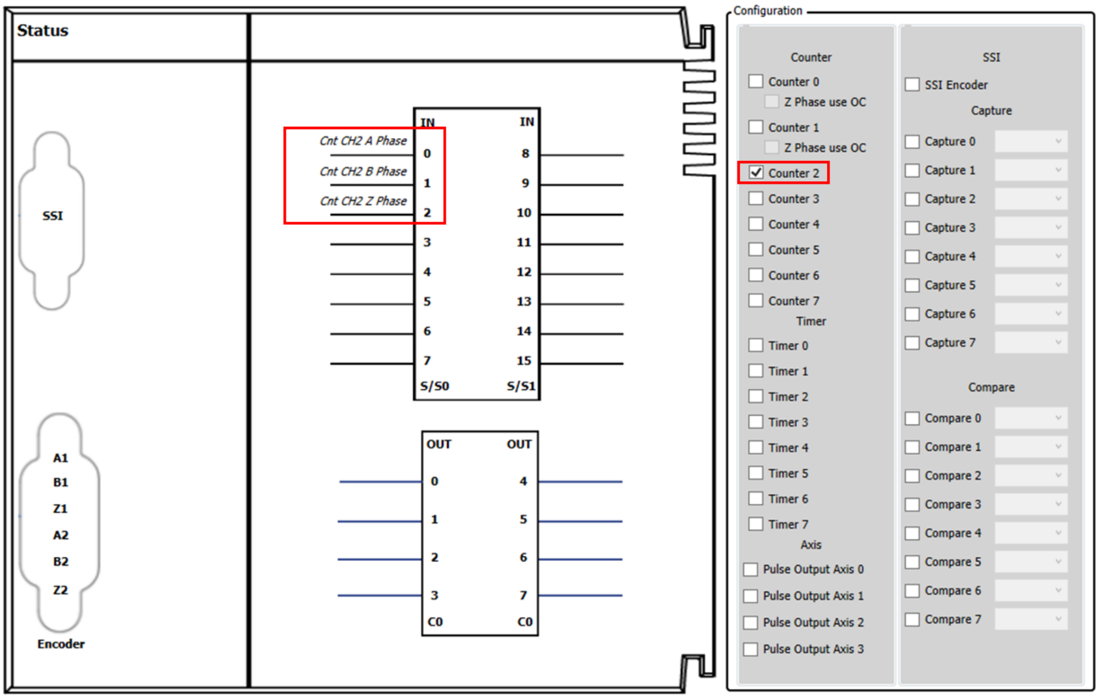
- **Troubleshooting**

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.



Configuration

Counter
- □ Counter 0
- □ Z Phase use OC
- □ Counter 1
- □ Z Phase use OC
- ☑ Counter 2
- □ Counter 3
- □ Counter 4
- □ Counter 5
- □ Counter 6
- □ Counter 7

Timer
- □ Timer 0
- □ Timer 1
- □ Timer 2
- □ Timer 3
- □ Timer 4
- □ Timer 5
- □ Timer 6
- □ Timer 7

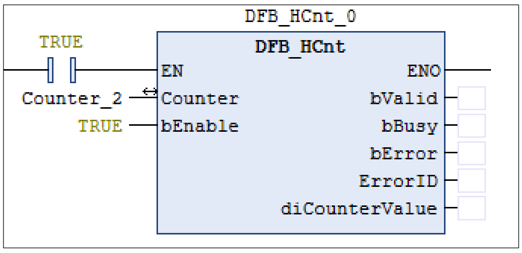
Axis
- □ Pulse Output Axis 0
- □ Pulse Output Axis 1
- □ Pulse Output Axis 2
- □ Pulse Output Axis 3

SSI
- □ SSI Encoder

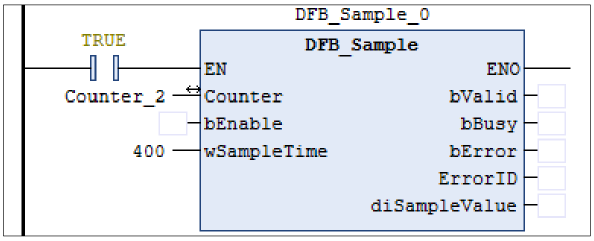
Capture
- □ Capture 0
- □ Capture 1
- □ Capture 2
- □ Capture 3
- □ Capture 4
- □ Capture 5
- □ Capture 6
- □ Capture 7

Compare
- □ Compare 0
- □ Compare 1
- □ Compare 2
- □ Compare 3
- □ Compare 4
- □ Compare 5
- □ Compare 6
- □ Compare 7

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 counter 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 counter values.



3. After enabling DFB_Sample in the POU (bEnable = True), the FB starts counting the increasing number of the pulse counter 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.

- **Library**

  ▪ DL_BuiltInIO_AX3.library

Note: From version 1.0.5.0 and later, lilbrary DL_BuiltInIO_AX3 is changed to 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_ALREADY_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_ALREADY_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.

Description	Cause of Error	Corrective Action
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.
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_CHANGED_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_TRIGGER	“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

Description	Cause of Error	Corrective Action
		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.
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.

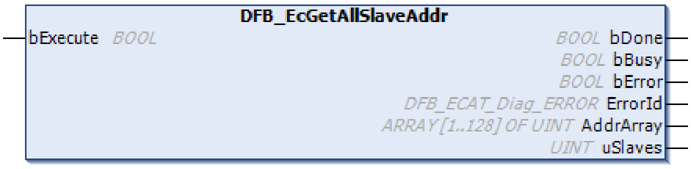
Chapter 4 EtherCAT Network Instructions

4.1 DFB_EcGetAllSlaveAddr

- **Supported Products**

  - AX-308E, AX-364E

DFB_EcGetAllSlaveAddr gets all the slave addresses.

FB/FC	Instruction	Graphic Expression
FB	DFB_EcGetAllSlaveAddr	 DFB_EcGetAllSlaveAddr bExecute BOOL BOOL bDone BOOL bBusy BOOL bError DFB_ECAT_Diag_ERROR ErrorId ARRAY [1..128] OF UINT AddrArray UINT uSlaves

ST Language

```
DFB_EcGetAllSlaveAddr (
bExecute :=,
bDone =>,
bBusy =>,
bError =>,
ErrorID =>,
AddrArray =>,
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)
AddrArray	Slave address array.	UINT[1..128]	(0)

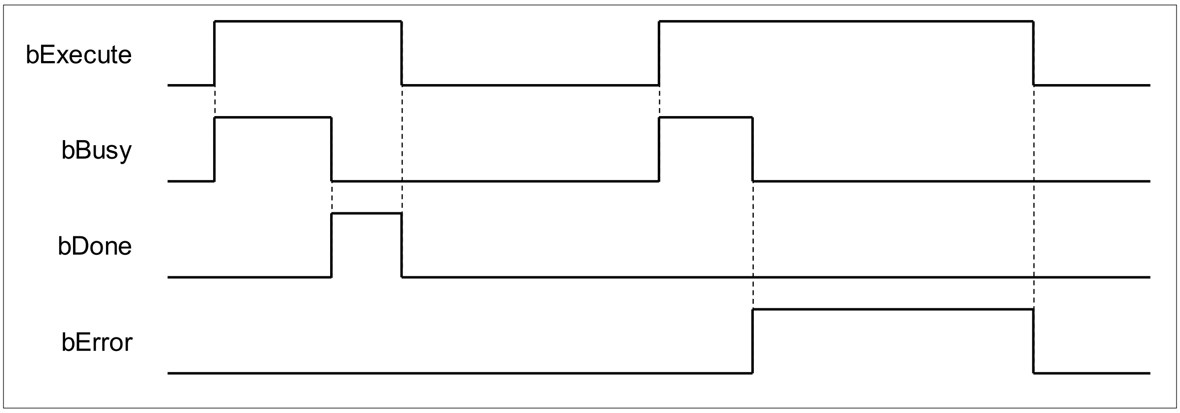
Name	Function	Data Type	Output Range (Default value)
uSlaves	The number of slaves.	UINT	0~128(0)

***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**

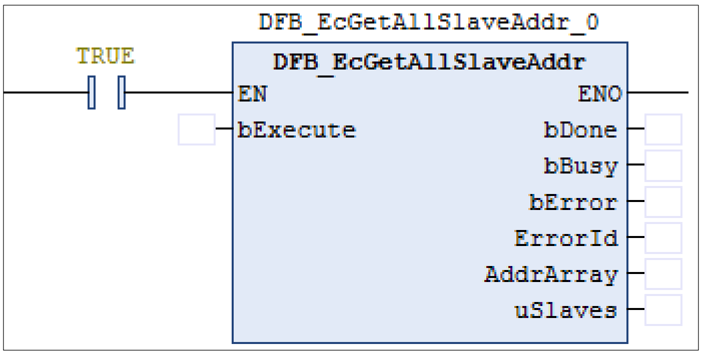
When bExecute shifts to True, the output AddrArray gives the addresses of all the EtherCAT slaves in the project tree, which supports up to 128 stations. Therefore, the maximum number of slave addresses output by AddrArray would be 128 given by the output uSlaves.

- **Troubleshooting**

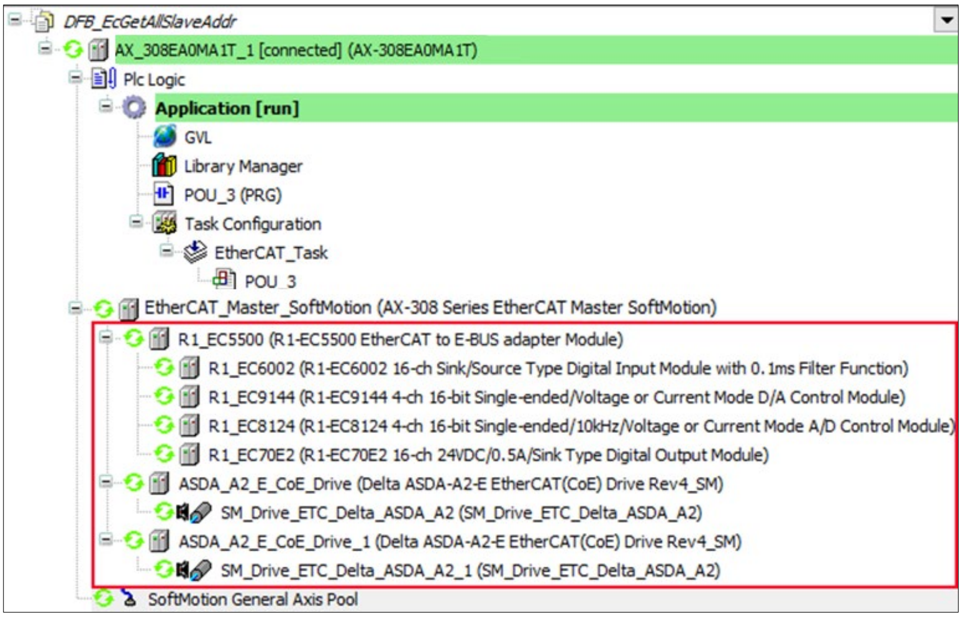
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

The following example demonstrates the behavior of DFB_EcGetAllSlaveAddr.

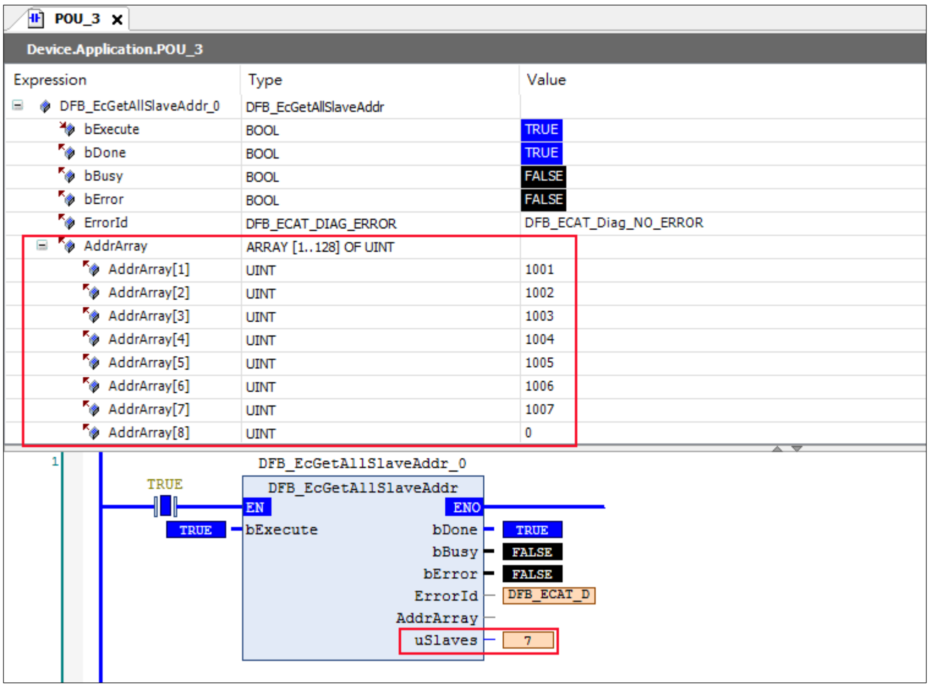


1. There're a total of 7 EtherCAT slaves in the category EtherCAT_Master_SoftMotion.



```
DFB_EcGetAllSlaveAddr
  AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
    Plc Logic
      Application [run]
        GVL
        Library Manager
        POU_3 (PRG)
        Task Configuration
          EtherCAT_Task
            POU_3
    EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Master SoftMotion)
      R1_EC5500 (R1-EC5500 EtherCAT to E-BUS adapter Module)
        R1_EC6002 (R1-EC6002 16-ch Sink/Source Type Digital Input Module with 0.1ms Filter Function)
        R1_EC9144 (R1-EC9144 4-ch 16-bit Single-ended/Voltage or Current Mode D/A Control Module)
        R1_EC8124 (R1-EC8124 4-ch 16-bit Single-ended/10kHz/Voltage or Current Mode A/D Control Module)
        R1_EC70E2 (R1-EC70E2 16-ch 24VDC/0.5A/Sink Type Digital Output Module)
      ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE) Drive Rev4_SM)
        SM_Drive_ETC_Delta_ASDA_A2 (SM_Drive_ETC_Delta_ASDA_A2)
      ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(CoE) Drive Rev4_SM)
        SM_Drive_ETC_Delta_ASDA_A2_1 (SM_Drive_ETC_Delta_ASDA_A2)
    SoftMotion General Axis Pool
```

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.



POU_3

Device.Application.POU_3

Expression	Type	Value
DFB_EcGetAllSlaveAddr_0	DFB_EcGetAllSlaveAddr	
bExecute	BOOL	TRUE
bDone	BOOL	TRUE
bBusy	BOOL	FALSE
bError	BOOL	FALSE
ErrorId	DFB_ECAT_DIAG_ERROR	DFB_ECAT_Diag_NO_ERROR
AddrArray	ARRAY [1..128] OF UINT	
AddrArray[1]	UINT	1001
AddrArray[2]	UINT	1002
AddrArray[3]	UINT	1003
AddrArray[4]	UINT	1004
AddrArray[5]	UINT	1005
AddrArray[6]	UINT	1006
AddrArray[7]	UINT	1007
AddrArray[8]	UINT	0

- **Library**

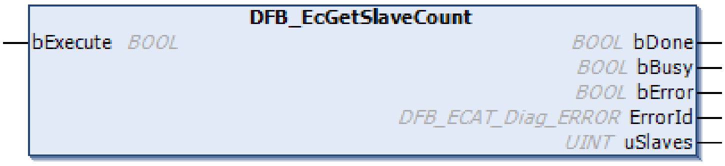
  ▪ DL_EtherCAT_Diag.library

4.2 DFB_EcGetSlaveCount

- **Supported Products**

  - AX-308E, AX-364E

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

FB/FC	Instruction	Graphic Expression
FB	DFB_EcGetSlaveCount	 DFB_EcGetSlaveCount bExecute BOOL BOOL bDone BOOL bBusy BOOL bError DFB_ECAT_Diag_ERROR ErrorId UINT uSlaves

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**

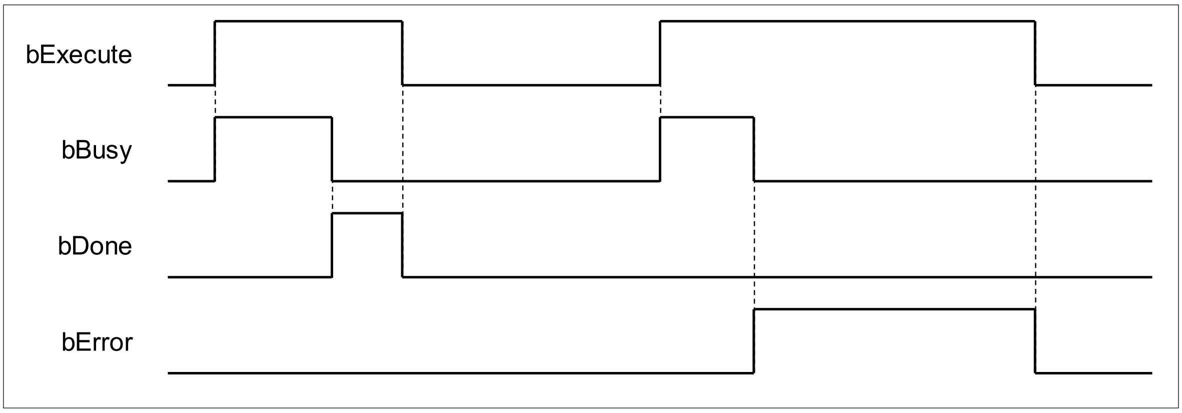
Name	Function	Data Type	Output Range (Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*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

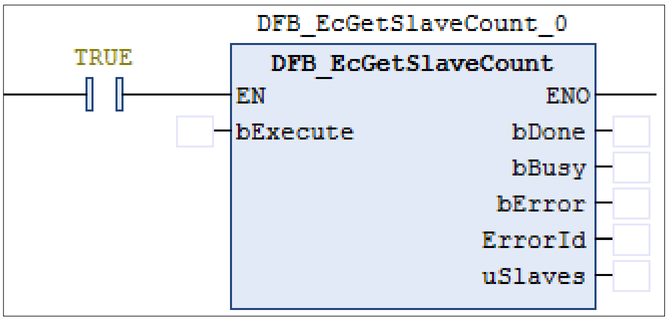
When bExecute shifts to True, the output uSlaves gives the number of EtherCAT slaves in the project tree.

- **Troubleshooting**

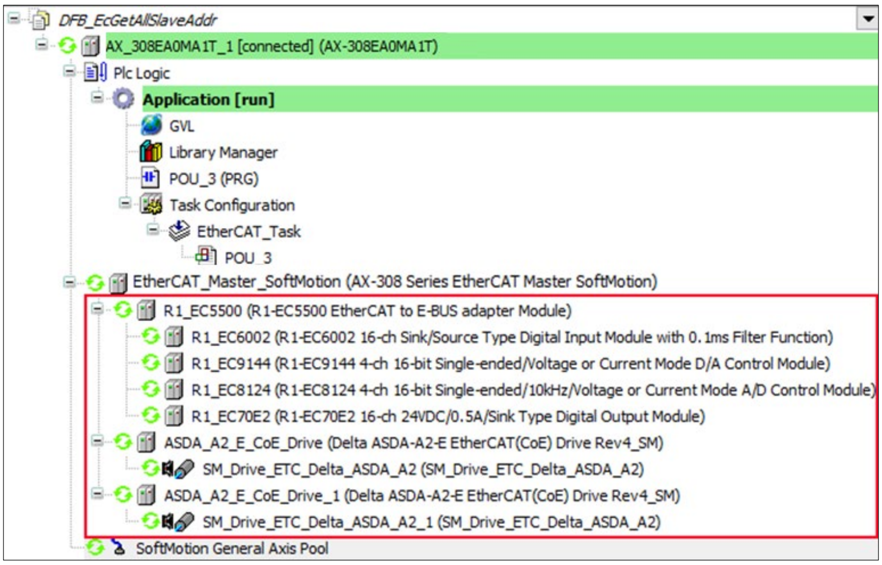
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

1. The following example demonstrates the behavior of DFB_EcGetSlaveCount.

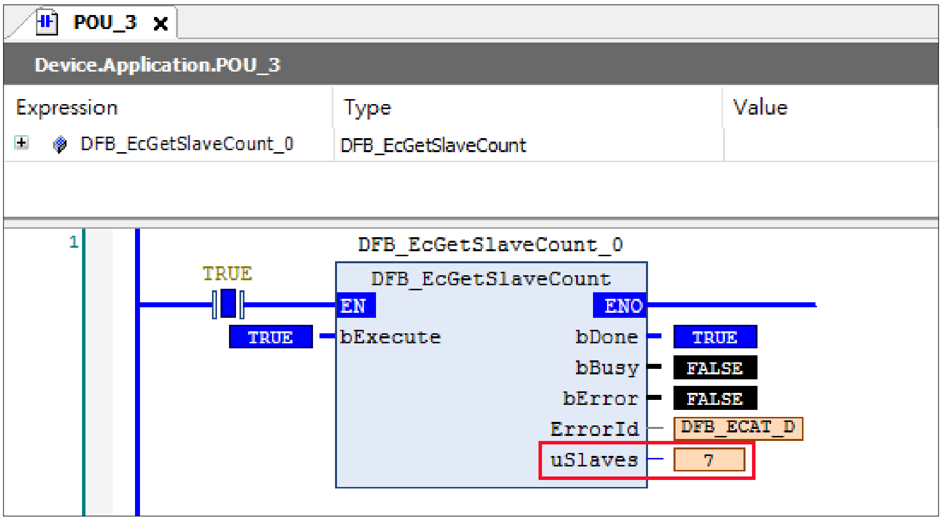


2. There're a total of 7 EtherCAT slaves in the category EtherCAT_Master_SoftMotion.



DFB_EcGetAllSlaveAddr
- AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
  - Plc Logic
    - Application [run]
      - GVL
      - Library Manager
      - POU_3 (PRG)
      - Task Configuration
        - EtherCAT_Task
          - POU_3
  - EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Master SoftMotion)
    - R1_EC5500 (R1-EC5500 EtherCAT to E-BUS adapter Module)
      - R1_EC6002 (R1-EC6002 16-ch Sink/Source Type Digital Input Module with 0.1ms Filter Function)
      - R1_EC9144 (R1-EC9144 4-ch 16-bit Single-ended/Voltage or Current Mode D/A Control Module)
      - R1_EC8124 (R1-EC8124 4-ch 16-bit Single-ended/10kHz/Voltage or Current Mode A/D Control Module)
      - R1_EC70E2 (R1-EC70E2 16-ch 24VDC/0.5A/Sink Type Digital Output Module)
    - ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE) Drive Rev4_SM)
      - SM_Drive_ETC_Delta_ASDA_A2 (SM_Drive_ETC_Delta_ASDA_A2)
    - ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(CoE) Drive Rev4_SM)
      - SM_Drive_ETC_Delta_ASDA_A2_1 (SM_Drive_ETC_Delta_ASDA_A2)
  - SoftMotion General Axis Pool

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



POU_3

Device.Application.POU_3

Expression	Type	Value
DFB_EcGetSlaveCount_0	DFB_EcGetSlaveCount	

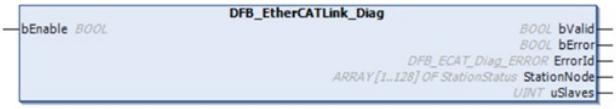
- **Library**
  - DL_EtherCAT_Diag.library

4.3 DFB_EtherCATLink_Diag

- **Supported Products**

  - AX-308E, AX-364E

DFB_EtherCATLink_Diag is used to display all the EtherCAT slave diagnostics.

<table>
<tr><th>FB/FC</th><th>Instruction</th><th>Graphic Expression</th></tr>
<tr><td>FB</td><td>DFB_EtherCATLink_Diag</td><td>

DFB_EtherCATLink_Diag
bEnable BOOL
BOOL bValid
BOOL bError
DFB_ECAT_Diag_ERROR ErrorId
ARRAY [1..128] OF StationStatus StationNode
UINT uSlaves

</td></tr>
<tr><th colspan="3">ST Language</th></tr>
<tr><td colspan="3">DFB_EtherCATLink_Diag (
bEnable :=,
bValid =>,
bError =>,
ErrorID =>,
StationNode =>
uSlaves =>
);</td></tr>
</table>

- **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

uSlaves	Number of Slaves	UINT	0 ~ 65535

***Note:**

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

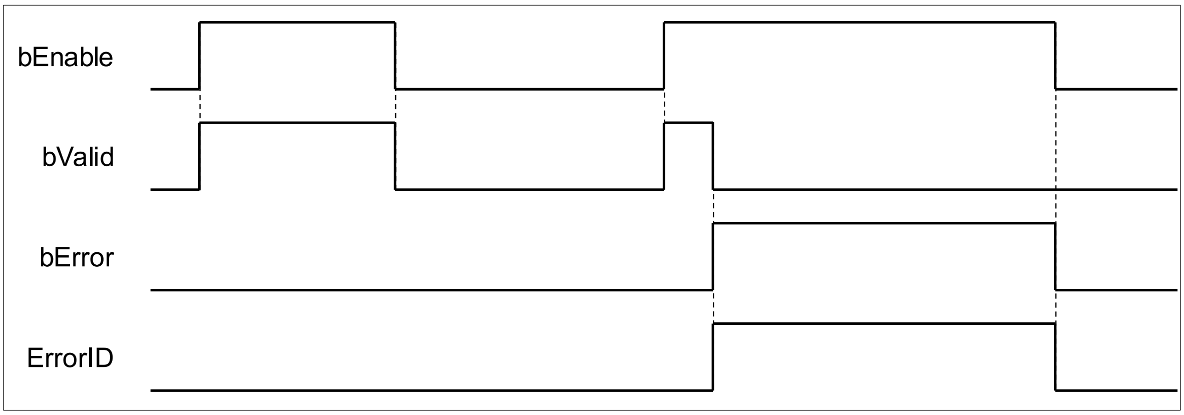
Name	Function	Data Type	Setting Value (Default value)
StationAddress	Slave station address	UINT	(0)
Node	Connection status of slave stations	BOOL	True: Connected and functioning properly. False: Abnormal connection status. (False)
LinkStatus	Slave physical connection status	BOOL	True: Physical connection status normal False: Physical connection status abnormal (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.
uSlaves	● When bEnable is rising edge triggered.	● When bEnable shifts to False.

- **Timing Diagram**



- **Function**

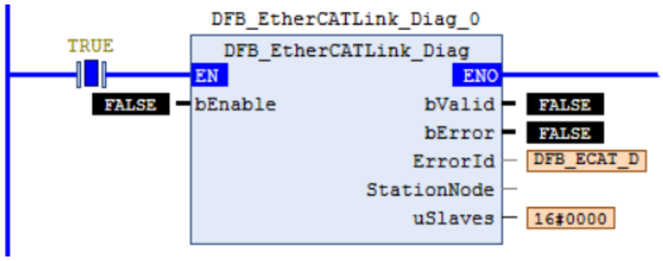
When bEnable shifts to True, StationAddress and Node output from StationAddress are in array type to show all the slave addresses and status with the support up to 128 slave stations. If the value of StationAddress is shown as 0 in the struct array after bEnable is rising edge triggered, it indicates 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**

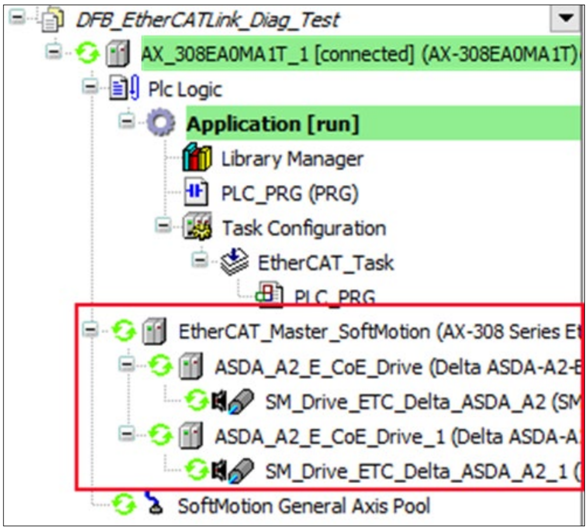
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

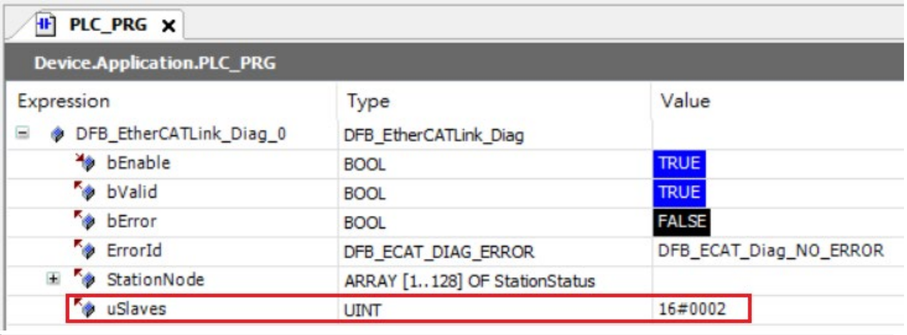
The following example demonstrates the behavior of DFB_EtherCATLink_Diag.



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


DFB_EtherCATLink_Diag_Test
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
Application [run]
Library Manager
PLC_PRG (PRG)
Task Configuration
EtherCAT_Task
PLC_PRG
EtherCAT_Master_SoftMotion (AX-308 Series Et
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E
SM_Drive_ETC_Delta_ASDA_A2 (SM
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A
SM_Drive_ETC_Delta_ASDA_A2_1 (
SoftMotion General Axis Pool

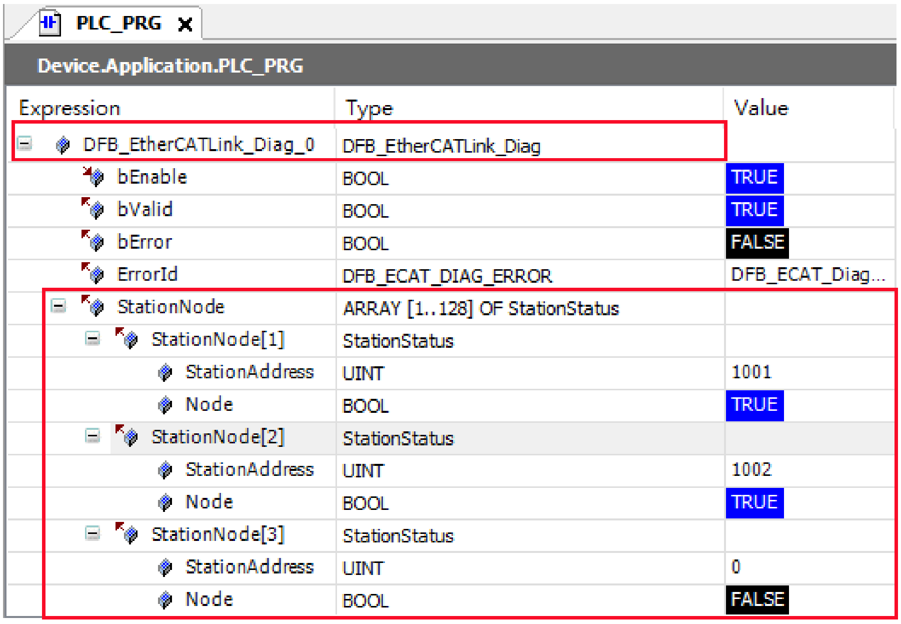
2. After bEnable of DFB_EtherCATLink_Diag shifts to True, the output uSlave shows 2, indicating that there are two EtherCAT slaves in total. The array of StationNode index 1 to 2 shows the two slave addresses, statuses, and physical connection statuses, and the StationAddress of StationNode shows 0 from index 3.



PLC_PRG

Device.Application.PLC_PRG

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_NO_ERROR
StationNode	ARRAY [1..128] OF StationStatus	
uSlaves	UINT	16#0002

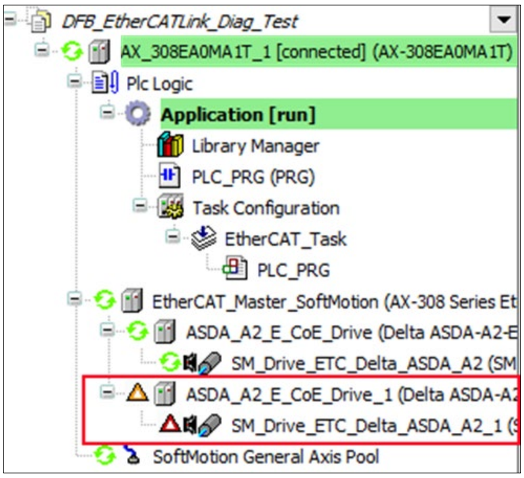


PLC_PRG

Device.Application.PLC_PRG

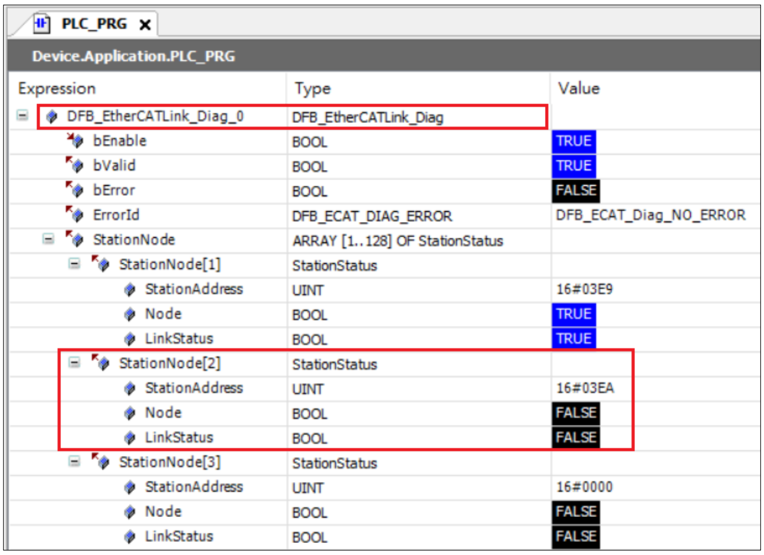
Expression	Type	Value
DFB_EtherCATLink_Diag_0	DFB_EtherCATLink_Diag	
bEnable	BOOL	TRUE
bValid	BOOL	TRUE
bError	BOOL	FALSE
ErrorId	DFB_ECAT_DIAG_ERROR	DFB_ECAT_Diag...
StationNode	ARRAY [1..128] OF StationStatus	
StationNode[1]	StationStatus	
StationAddress	UINT	1001
Node	BOOL	TRUE
StationNode[2]	StationStatus	
StationAddress	UINT	1002
Node	BOOL	TRUE
StationNode[3]	StationStatus	
StationAddress	UINT	0
Node	BOOL	FALSE

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.



DFB_EtherCATLink_Diag_Test
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
Application [run]
Library Manager
PLC_PRG (PRG)
Task Configuration
EtherCAT_Task
PLC_PRG
EtherCAT_Master_SoftMotion (AX-308 Series Et
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E
SM_Drive_ETC_Delta_ASDA_A2 (SM
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2
SM_Drive_ETC_Delta_ASDA_A2_1 (S
SoftMotion General Axis Pool

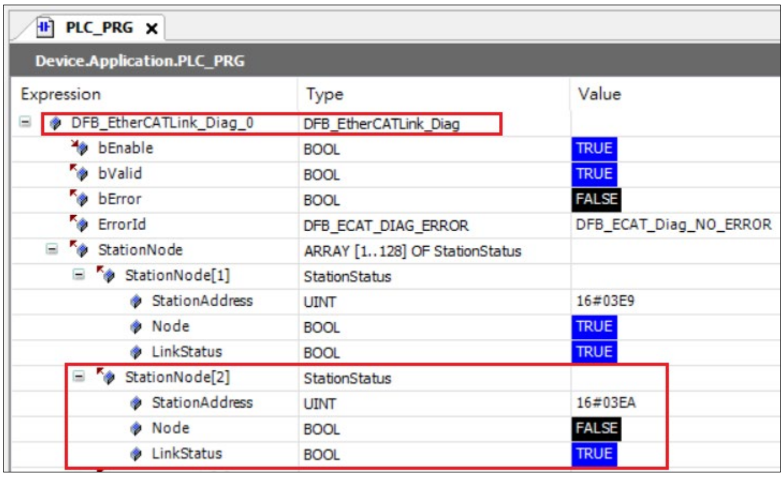
4. Now, StationNode index 2 Node and LinkStatus of DFB_EtherCATLink_Diag are False respectively.



Device.Application.PLC_PRG

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_NO_ERROR
StationNode	ARRAY [1..128] OF StationStatus	
StationNode[1]	StationStatus	
StationAddress	UINT	16#03E9
Node	BOOL	TRUE
LinkStatus	BOOL	TRUE
StationNode[2]	StationStatus	
StationAddress	UINT	16#03EA
Node	BOOL	FALSE
LinkStatus	BOOL	FALSE
StationNode[3]	StationStatus	
StationAddress	UINT	16#0000
Node	BOOL	FALSE
LinkStatus	BOOL	FALSE

5. If connecting the physical network of Slave 1 and Slave 2, StationNode index 2 Node and LinkStatus of DFB_EtherCATLink_Diag will be False and True respectively.



Device.Application.PLC_PRG

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_NO_ERROR
StationNode	ARRAY [1..128] OF StationStatus	
StationNode[1]	StationStatus	
StationAddress	UINT	16#03E9
Node	BOOL	TRUE
LinkStatus	BOOL	TRUE
StationNode[2]	StationStatus	
StationAddress	UINT	16#03EA
Node	BOOL	FALSE
LinkStatus	BOOL	TRUE

- **Library**

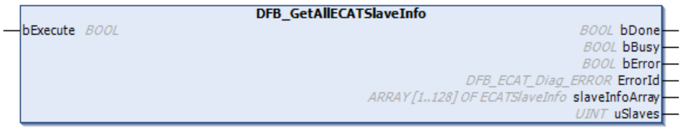
  ▪ DL_EtherCAT_Diag.library

4.4 DFB_GetAllECATSlaveInfo

- **Supported Products**

  - AX-308E, AX-364E

DFB_GetAllECATSlaveInfo gets all the slaves' information.

<table>
<tr><th>FB/FC</th><th>Instruction</th><th>Graphic Expression</th></tr>
<tr><td>FB</td><td>DFB_GetAllECATSlaveInfo</td><td>
</td></tr>
<tr><th colspan="3">ST Language</th></tr>
<tr><td colspan="3">DFB_GetAllECATSlaveInfo (
bExecute :=,
bDone =>,
bBusy =>,
bError =>,
ErrorID =>,
slaveInfoArray =>,
uSlaves =>,);</td></tr>
</table>

- **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	DFB_ECAT_Diag_ERROR*[1]	DFB_ECAT_Diag_ERROR

Name	Function	Data Type	Output Range (Default value)
	if an error occurs.		(DFB_ECAT_Diag_NO_ERROR)
slaveInfoArray	Slave information array.	ECATSlaveInfo [1..128] *2	ECATSlaveInfo
uSlaves	The number of slaves.	UINT	0~128(0)

***Note:**

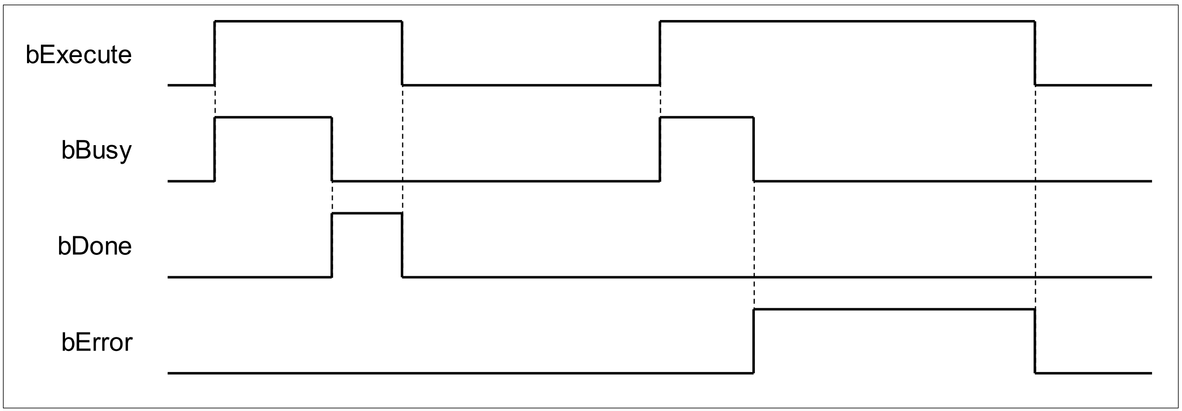
1. DFB_ECAT_Diag_ERROR: Enumeration(Enum)
2. slaveInfoArray: Structure(STRUCT)。

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**

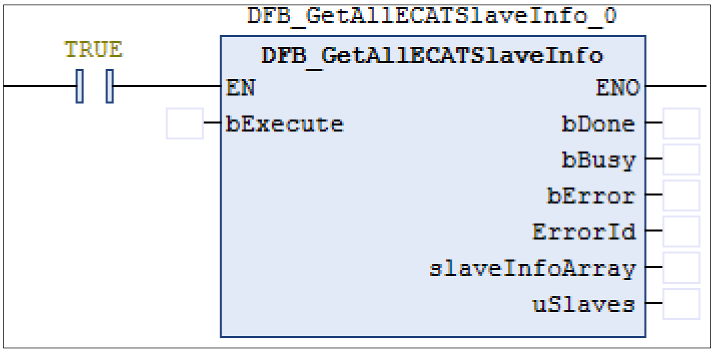
When bExecute shifts to True, slaveInfoArray gives the information of all the EtherCAT slaves in the device tree, which includes vendor id, product code, revision number and serial number. Support up to 128 stations as well as the maximum number of slaves and the corresponding information output from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

- **Troubleshooting**

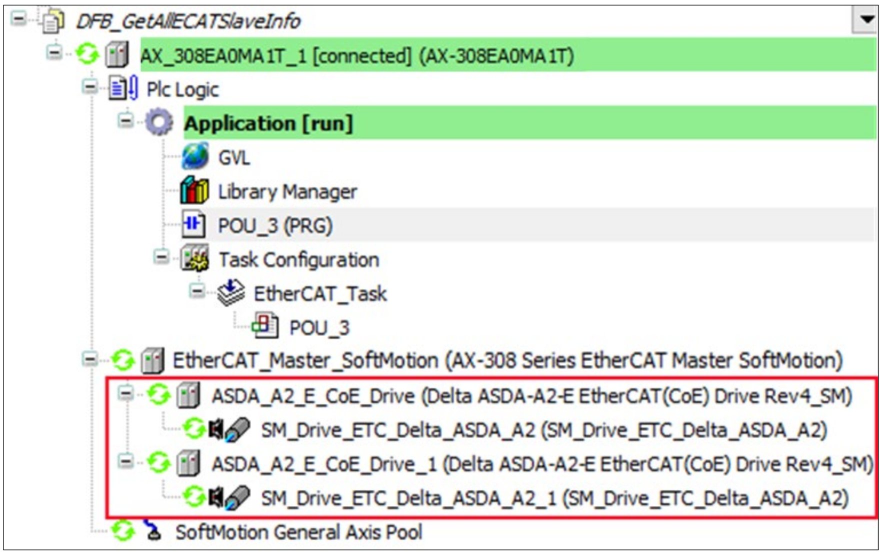
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

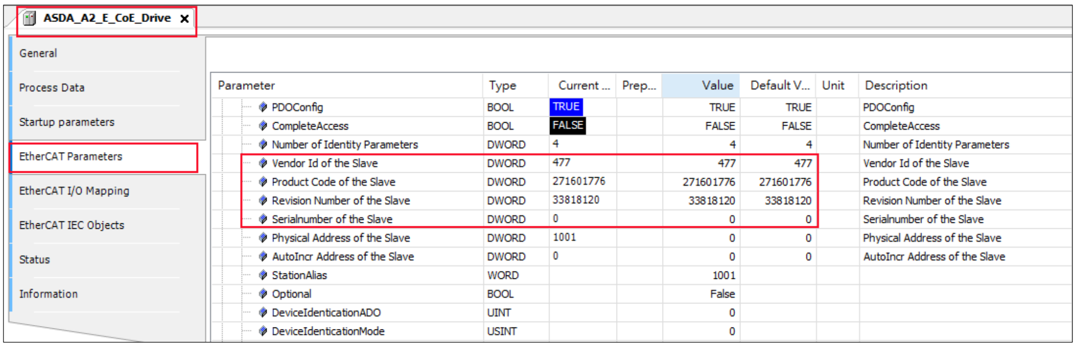
The following example demonstrates the behavior of DFB_GetAllECATSlaveInfo.



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

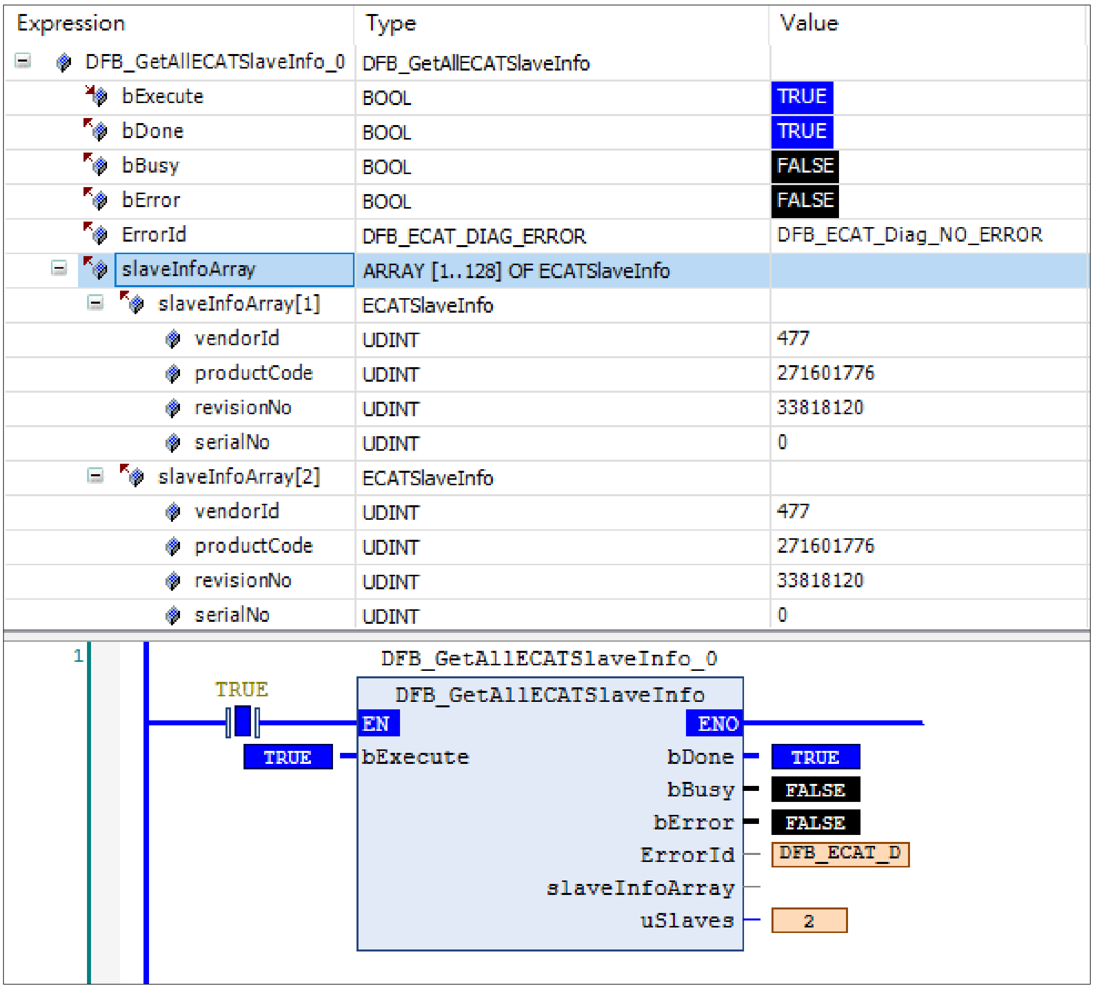

DFB_GetAllECATSlaveInfo
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
Application [run]
GVL
Library Manager
POU_3 (PRG)
Task Configuration
EtherCAT_Task
POU_3
EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Master SoftMotion)
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE) Drive Rev4_SM)
SM_Drive_ETC_Delta_ASDA_A2 (SM_Drive_ETC_Delta_ASDA_A2)
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(CoE) Drive Rev4_SM)
SM_Drive_ETC_Delta_ASDA_A2_1 (SM_Drive_ETC_Delta_ASDA_A2)
SoftMotion General Axis Pool

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



Parameter	Type	Current ...	Prep...	Value	Default V...	Unit	Description
PDOConfig	BOOL	TRUE		TRUE	TRUE		PDOConfig
CompleteAccess	BOOL	FALSE		FALSE	FALSE		CompleteAccess
Number of Identity Parameters	DWORD	4		4	4		Number of Identity Parameters
Vendor Id of the Slave	DWORD	477		477	477		Vendor Id of the Slave
Product Code of the Slave	DWORD	271601776		271601776	271601776		Product Code of the Slave
Revision Number of the Slave	DWORD	33818120		33818120	33818120		Revision Number of the Slave
Serialnumber of the Slave	DWORD	0		0	0		Serialnumber of the Slave
Physical Address of the Slave	DWORD	1001		0	0		Physical Address of the Slave
AutoIncr Address of the Slave	DWORD	0		0	0		AutoIncr Address of the Slave
StationAlias	WORD			1001			
Optional	BOOL			False			
DeviceIdenticationADO	UINT			0			
DeviceIdenticationMode	USINT			0			

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

- **Library**

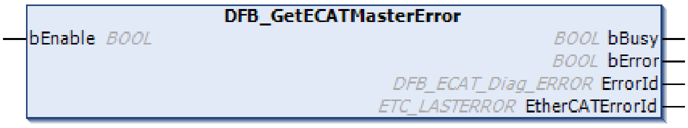
  ▪ DL_EtherCAT_Diag.library

4.5 DFB_GetECATMasterError

- **Supported Products**

  - AX-308E, AX-364E

DFB_GetECATMasterError gets the error code of failed EtherCAT network connection.

FB/FC	Instruction	Graphic Expression
FB	DFB_GetECATMasterError	 DFB_GetECATMasterError bEnable BOOL BOOL bBusy BOOL bError DFB_ECAT_Diag_ERROR ErrorId ETC_LASTERROR EtherCATErrorId
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)

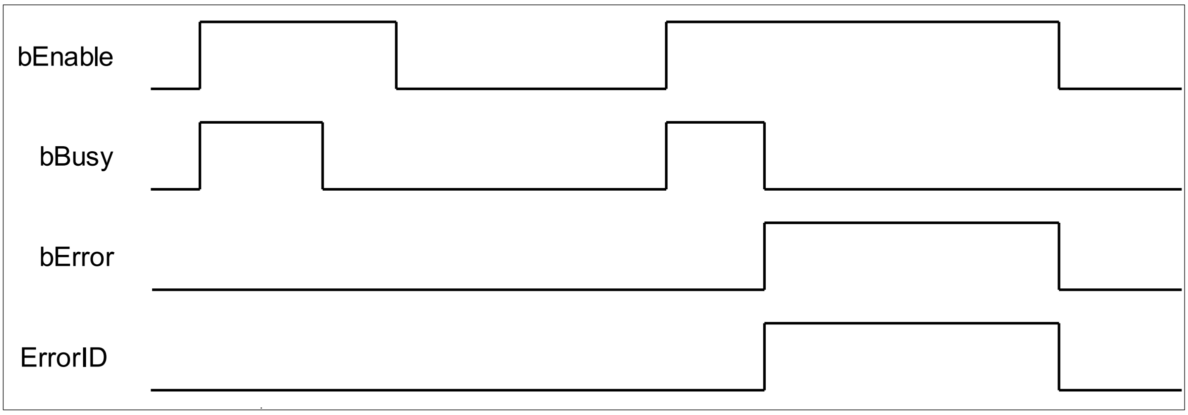
***Note:**

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**

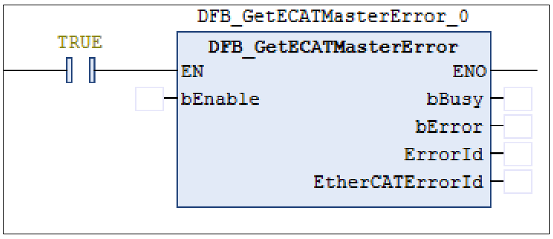
When bEnable shifts to True, the output EtherCATErrorId gives the error codes of failed EtherCAT network connection during each cycle. If there's no error, the output would be displayed as NO_ERROR. For more details of error codes, please refer to the content of ETC_LASTERROR_STATE in the Library.

- **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.

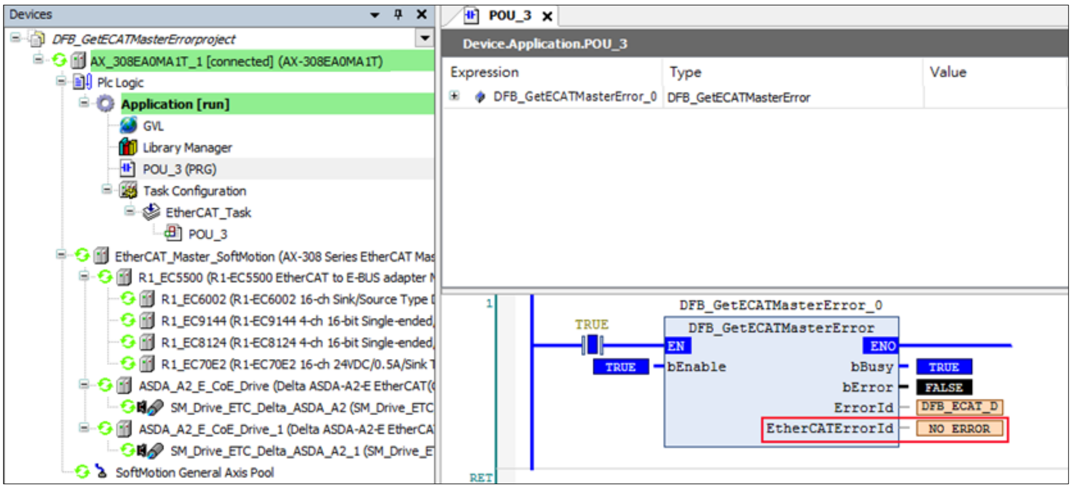
- **Programming Example**

The following example demonstrates the behavior of DFB_GetECATMasterError.

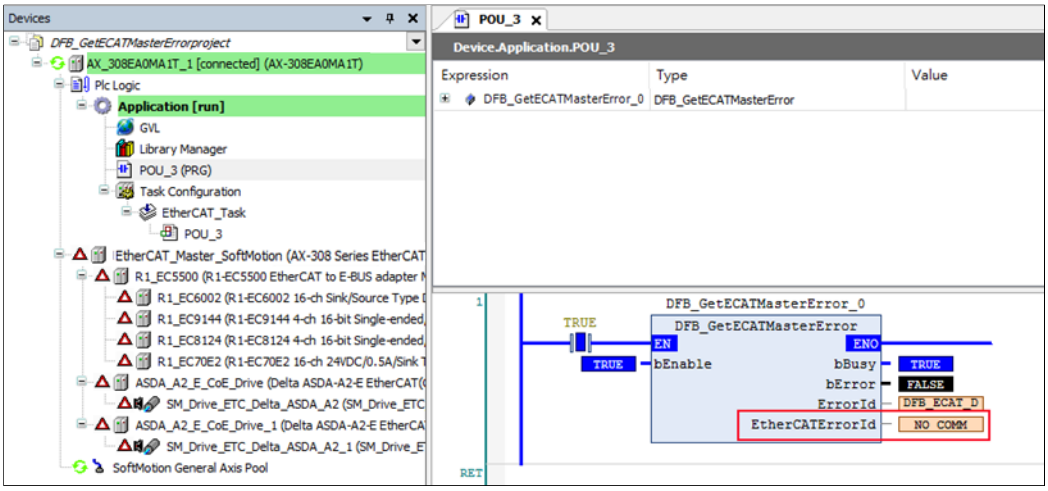


1. If the EtherCAT connection is normal without any existing errors, the output content of EtherCATErrorId would be

shown as NO_ERROR after the input bEnable of DFB_GetECATMasterError shifts to True.


Devices
DFB_GetECATMasterErrorproject
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
Application [run]
GVL
Library Manager
POU_3 (PRG)
Task Configuration
EtherCAT_Task
POU_3
EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Mas
R1_EC5500 (R1-EC5500 EtherCAT to E-BUS adapter
R1_EC6002 (R1-EC6002 16-ch Sink/Source Type
R1_EC9144 (R1-EC9144 4-ch 16-bit Single-ended
R1_EC8124 (R1-EC8124 4-ch 16-bit Single-ended
R1_EC70E2 (R1-EC70E2 16-ch 24VDC/0.5A/Sink T
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(
SM_Drive_ETC_Delta_ASDA_A2 (SM_Drive_ETC
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCA
SM_Drive_ETC_Delta_ASDA_A2_1 (SM_Drive_E
SoftMotion General Axis Pool
POU_3
Device.Application.POU_3
Expression
Type
Value
DFB_GetECATMasterError_0
DFB_GetECATMasterError
1
RET
DFB_GetECATMasterError_0
TRUE
DFB_GetECATMasterError
EN
ENO
TRUE
bEnable
bBusy
TRUE
bError
FALSE
ErrorId
DFB_ECAT_D
EtherCATErrorId
NO_ERROR

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


Devices
DFB_GetECATMasterErrorproject
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
Application [run]
GVL
Library Manager
POU_3 (PRG)
Task Configuration
EtherCAT_Task
POU_3
EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT
R1_EC5500 (R1-EC5500 EtherCAT to E-BUS adapter
R1_EC6002 (R1-EC6002 16-ch Sink/Source Type
R1_EC9144 (R1-EC9144 4-ch 16-bit Single-ended
R1_EC8124 (R1-EC8124 4-ch 16-bit Single-ended
R1_EC70E2 (R1-EC70E2 16-ch 24VDC/0.5A/Sink T
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(
SM_Drive_ETC_Delta_ASDA_A2 (SM_Drive_ETC
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCA
SM_Drive_ETC_Delta_ASDA_A2_1 (SM_Drive_E
SoftMotion General Axis Pool
POU_3
Device.Application.POU_3
Expression
Type
Value
DFB_GetECATMasterError_0
DFB_GetECATMasterError
1
RET
DFB_GetECATMasterError_0
TRUE
DFB_GetECATMasterError
EN
ENO
TRUE
bEnable
bBusy
TRUE
bError
FALSE
ErrorId
DFB_ECAT_D
EtherCATErrorId
NO_COMM

- **Library**

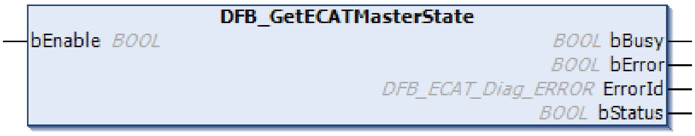
  - DL_EtherCAT_Diag.library

4.6 DFB_GetECATMasterState

- **Supported Products**

  - AX-308E, AX-364E

DFB_GetECATMasterState gets the connection status of EtherCAT Master.

FB/FC	Instruction	Graphic Expression
FB	DFB_GetECATasterState	 DFB_GetECATMasterState bEnable BOOL BOOL bBusy BOOL bError DFB_ECAT_Diag_ERROR ErrorId BOOL bStatus

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**

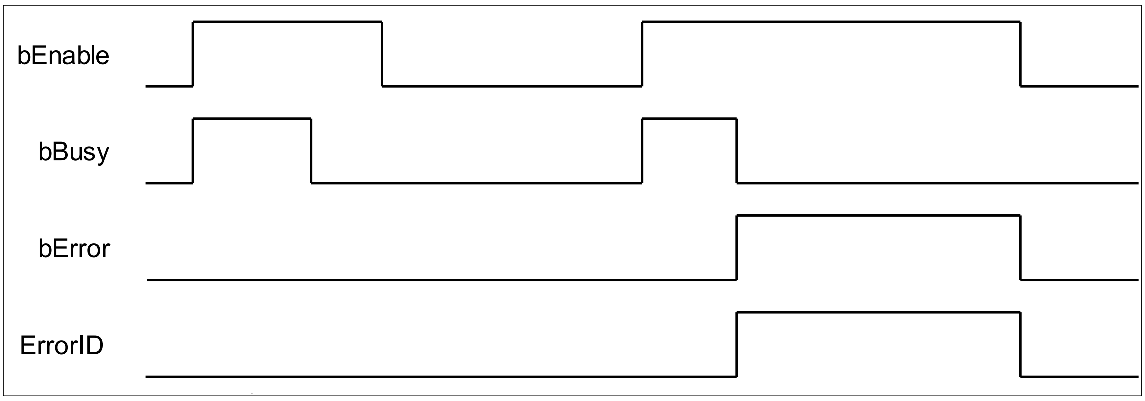
Name	Function	Data Type	Output Range (Default value)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
bStatus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

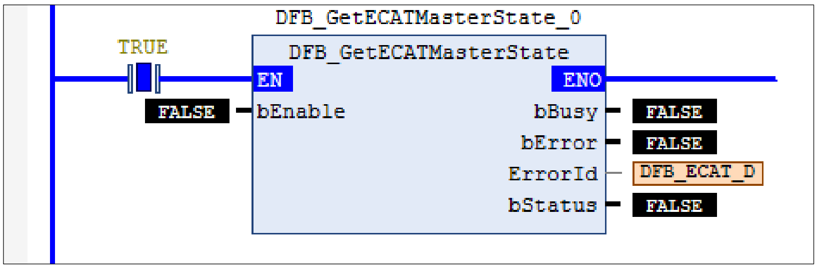
When bEnable shifts to True, the function block perform cyclical status updates of EtherCAT master communication.

- **Troubleshooting**

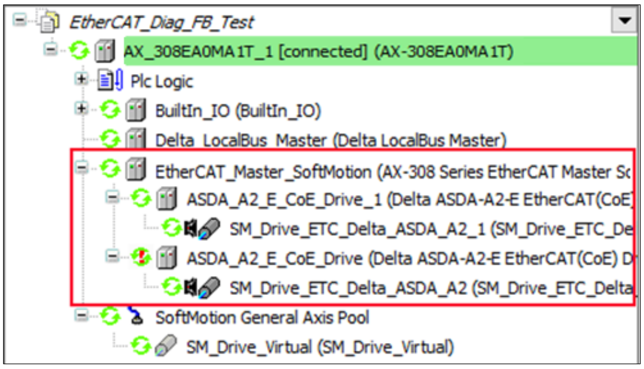
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

The following example demonstrates the behavior of DFB_GetECATMasterState.

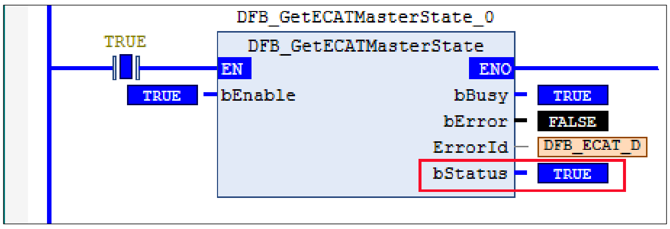


1. The connection status of EtherCAT master shows PASS in the device tree.

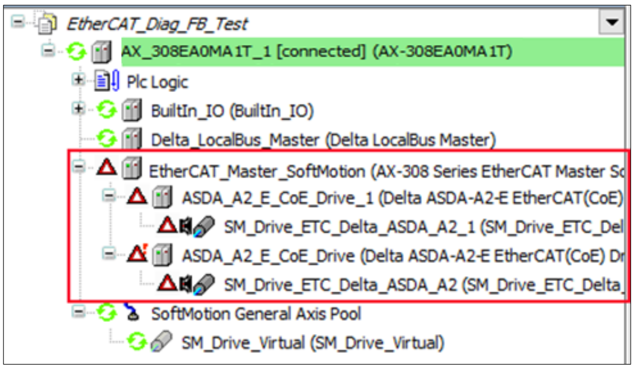


EtherCAT_Diag_FB_Test
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus Master)
EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Master Sc
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(CoE
SM_Drive_ETC_Delta_ASDA_A2_1 (SM_Drive_ETC_De
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE) D
SM_Drive_ETC_Delta_ASDA_A2 (SM_Drive_ETC_Delta
SoftMotion General Axis Pool
SM_Drive_Virtual (SM_Drive_Virtual)

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

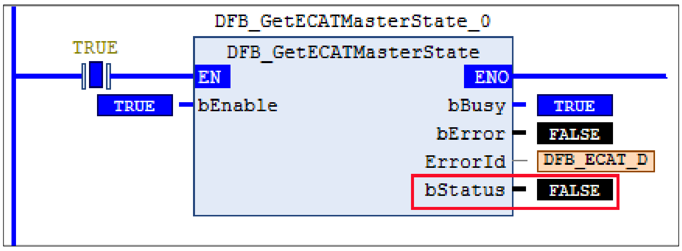


3. Remove the network connection between the master and the slave, and the current connection status of EtherCAT master would show Fail in the device tree.



EtherCAT_Diag_FB_Test
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus Master)
EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Master Sc
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(CoE)
SM_Drive_ETC_Delta_ASDA_A2_1 (SM_Drive_ETC_Del
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE) Dr
SM_Drive_ETC_Delta_ASDA_A2 (SM_Drive_ETC_Delta
SoftMotion General Axis Pool
SM_Drive_Virtual (SM_Drive_Virtual)

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



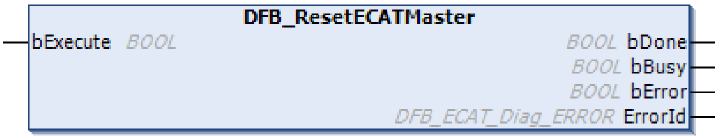
- **Library**

DL_EtherCAT_Diag.library

4.7 DFB_ResetECATMaster

- **Supported Products**
  - AX-308E, AX-364E

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

FB/FC	Instruction	Graphic Expression
FB	DFB_ResetECATMaster	 DFB_ResetECATMaster bExecute BOOL BOOL bDone BOOL bBusy BOOL bError DFB_ECAT_Diag_ERROR ErrorId
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)

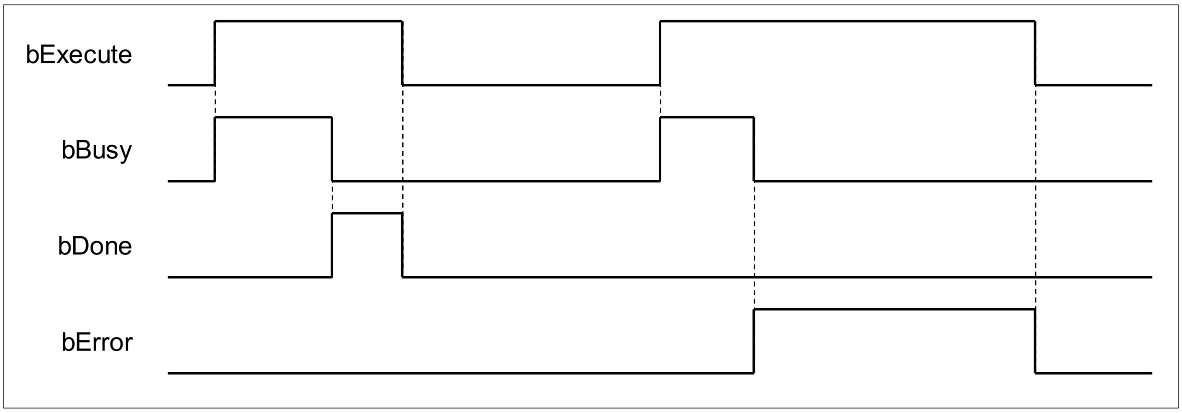
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)

***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**

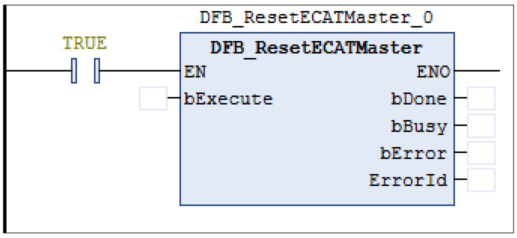
When bExecute shifts to True and the connection status of EtherCAT master shows Fail, the function block would perform reset action.

- **Troubleshooting**

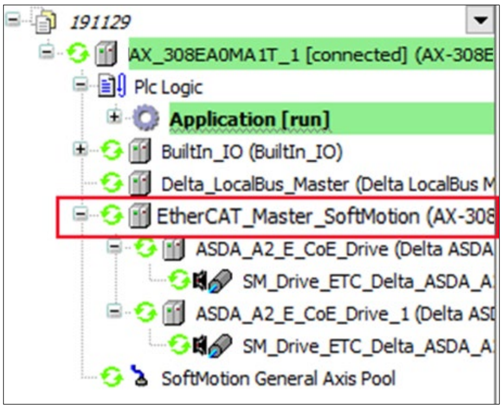
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

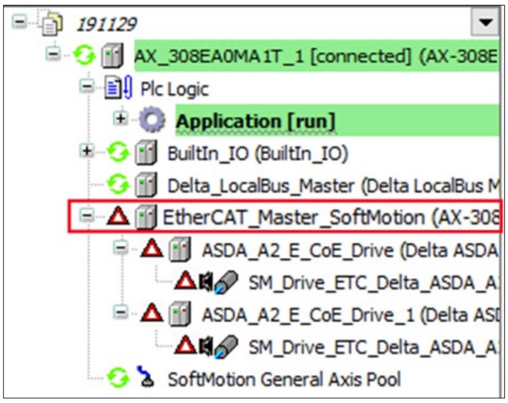
The following example demonstrates the behavior of DFB_ResetECATMaster.



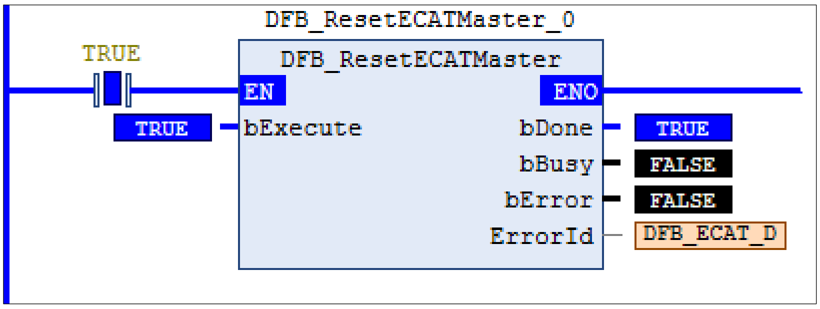
1. The connection status of EtherCAT master shows PASS in the device tree.


191129
AX_308EA0MA1T_1 [connected] (AX-308E
Plc Logic
Application [run]
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus M
EtherCAT_Master_SoftMotion (AX-308
ASDA_A2_E_CoE_Drive (Delta ASDA
SM_Drive_ETC_Delta_ASDA_A
ASDA_A2_E_CoE_Drive_1 (Delta ASI
SM_Drive_ETC_Delta_ASDA_A
SoftMotion General Axis Pool

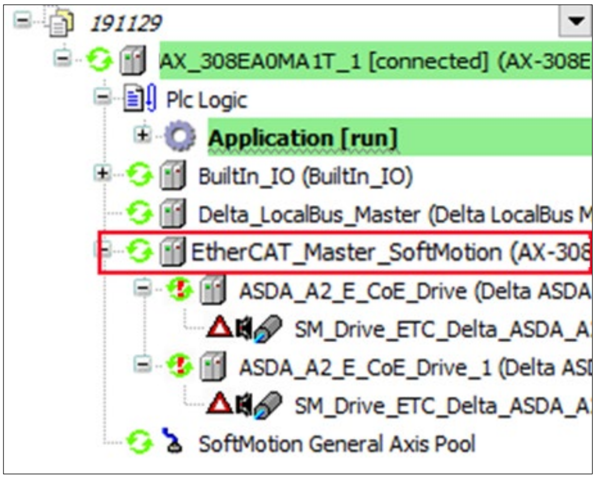
2. Remove the network connection between the master and the slave, and the current connection status of EtherCAT master would show Fail in the device tree.


191129
AX_308EA0MA1T_1 [connected] (AX-308E
Plc Logic
Application [run]
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus M
EtherCAT_Master_SoftMotion (AX-308
ASDA_A2_E_CoE_Drive (Delta ASDA
SM_Drive_ETC_Delta_ASDA_A
ASDA_A2_E_CoE_Drive_1 (Delta ASI
SM_Drive_ETC_Delta_ASDA_A
SoftMotion General Axis Pool

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.


191129
AX_308EA0MA1T_1 [connected] (AX-308E
Plc Logic
Application [run]
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus M
EtherCAT_Master_SoftMotion (AX-308
ASDA_A2_E_CoE_Drive (Delta ASDA
SM_Drive_ETC_Delta_ASDA_A
ASDA_A2_E_CoE_Drive_1 (Delta ASD
SM_Drive_ETC_Delta_ASDA_A
SoftMotion General Axis Pool

- **Library**

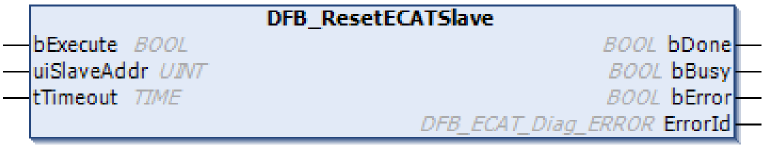
  - DL_EtherCAT_Diag.library

4.8 DFB_ResetECATSlave

- **Supported Products**

  - AX-308E, AX-364E

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**

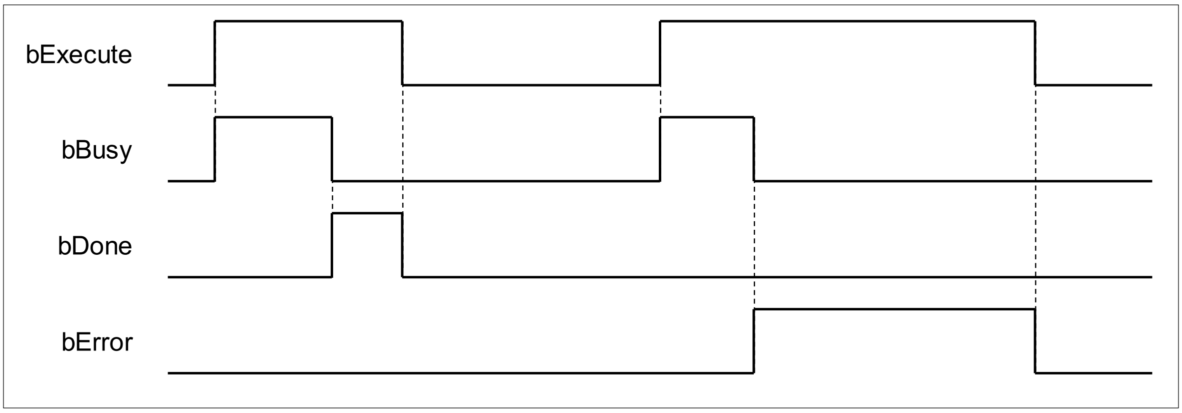
Name	Function	Data Type	Output Range (Default value)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bBusy	True when the instruction is enabled.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
ErrorID	Indicates the error code if an error occurs.	DFB_ECAT_Diag_ERROR*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

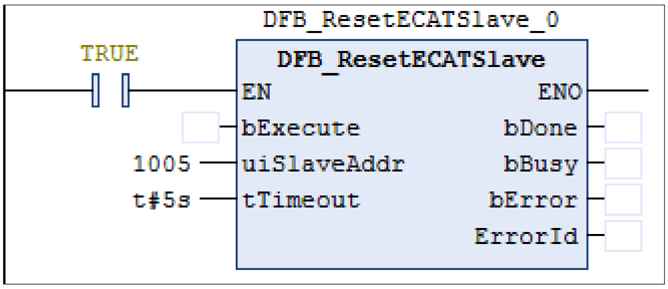
When bExecute shifts to True, the function block starts searching for the target slave station and resets the EtherCAT slave, if the status of target slave shows Fail. If the input value of uiSlaveAddr is 0, the function block would reset all the slave stations which have errors in connection.

- **Troubleshooting**

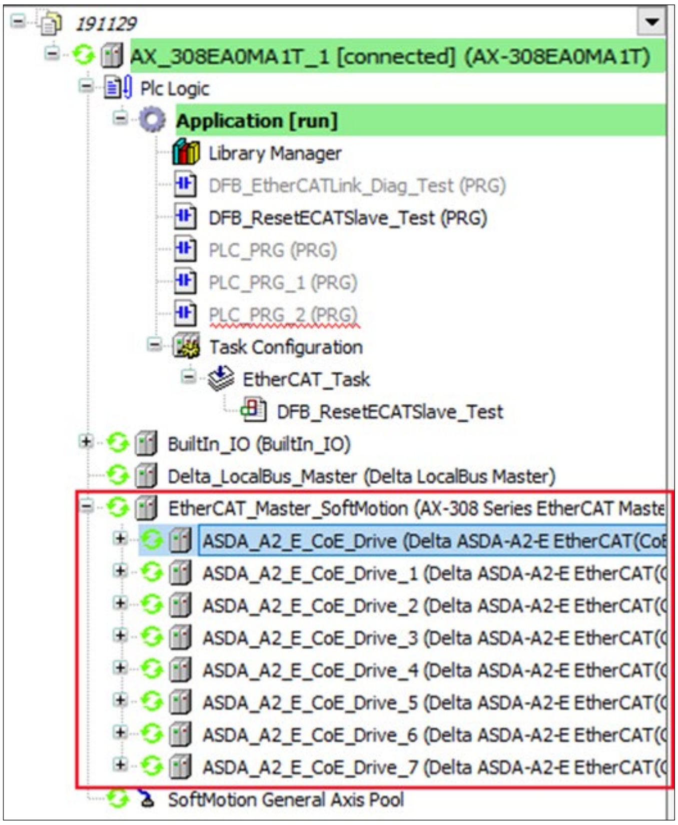
If an error occurs during the execution of the instruction, bError will change to True and the Capture will stop. You can refer to ErrorID (Error Code) to address the problem.

- **Programming Example**

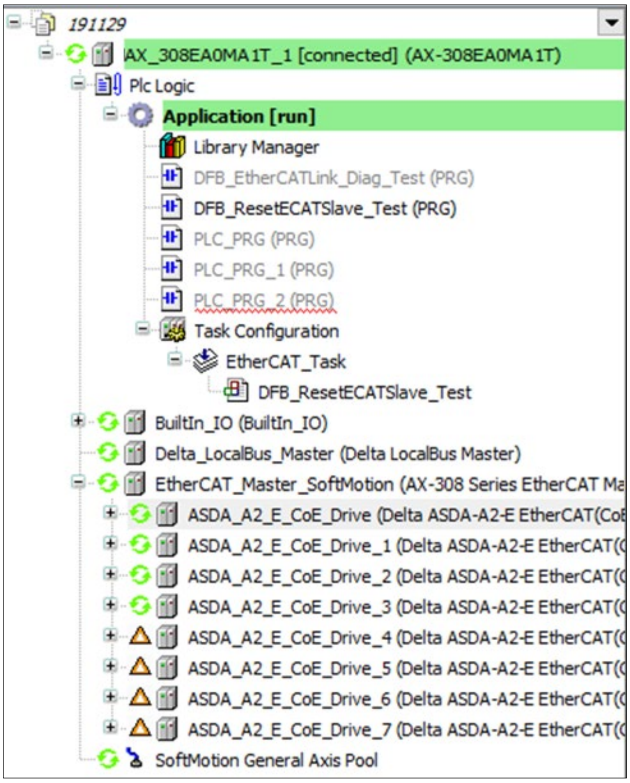
The following example demonstrates the behavior of DFB_ResetECATSlave.



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

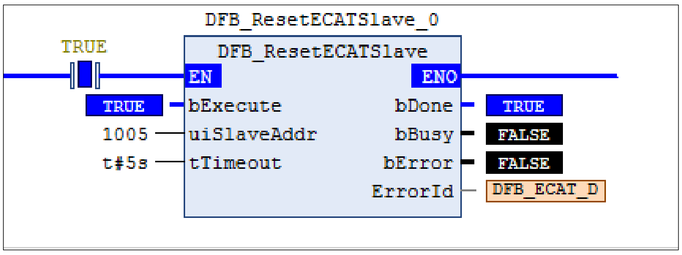

191129
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
Application [run]
Library Manager
DFB_EtherCATLink_Diag_Test (PRG)
DFB_ResetECATSlave_Test (PRG)
PLC_PRG (PRG)
PLC_PRG_1 (PRG)
PLC_PRG_2 (PRG)
Task Configuration
EtherCAT_Task
DFB_ResetECATSlave_Test
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus Master)
EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Maste
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(C
ASDA_A2_E_CoE_Drive_2 (Delta ASDA-A2-E EtherCAT(C
ASDA_A2_E_CoE_Drive_3 (Delta ASDA-A2-E EtherCAT(C
ASDA_A2_E_CoE_Drive_4 (Delta ASDA-A2-E EtherCAT(C
ASDA_A2_E_CoE_Drive_5 (Delta ASDA-A2-E EtherCAT(C
ASDA_A2_E_CoE_Drive_6 (Delta ASDA-A2-E EtherCAT(C
ASDA_A2_E_CoE_Drive_7 (Delta ASDA-A2-E EtherCAT(C
SoftMotion General Axis Pool

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.



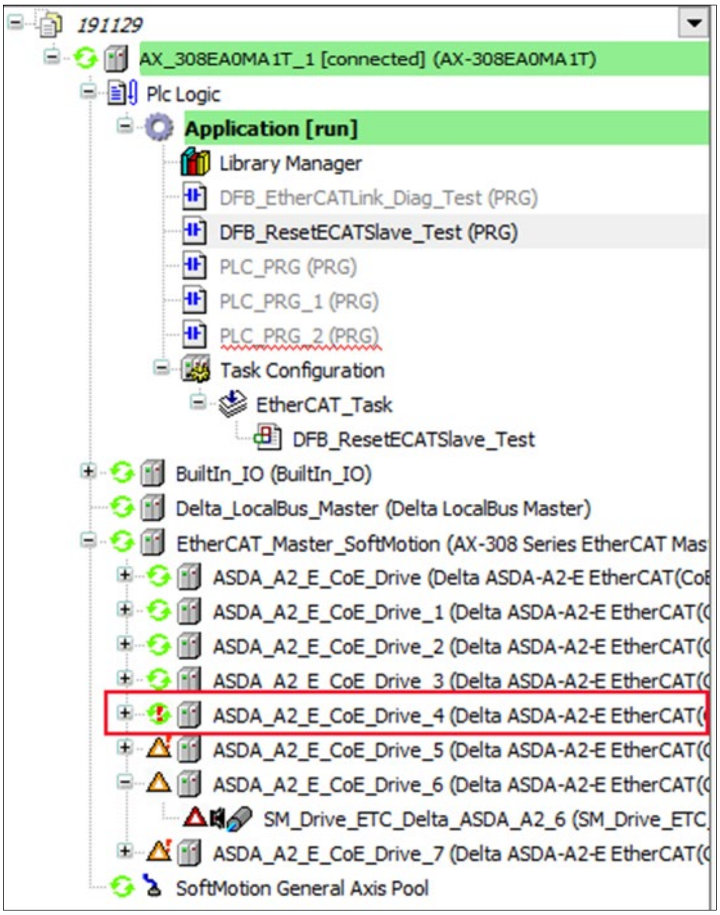
191129
AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
Plc Logic
Application [run]
Library Manager
DFB_EtherCATLink_Diag_Test (PRG)
DFB_ResetECATSlave_Test (PRG)
PLC_PRG (PRG)
PLC_PRG_1 (PRG)
PLC_PRG_2 (PRG)
Task Configuration
EtherCAT_Task
DFB_ResetECATSlave_Test
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus Master)
EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Ma
ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE
ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(
ASDA_A2_E_CoE_Drive_2 (Delta ASDA-A2-E EtherCAT(
ASDA_A2_E_CoE_Drive_3 (Delta ASDA-A2-E EtherCAT(
ASDA_A2_E_CoE_Drive_4 (Delta ASDA-A2-E EtherCAT(
ASDA_A2_E_CoE_Drive_5 (Delta ASDA-A2-E EtherCAT(
ASDA_A2_E_CoE_Drive_6 (Delta ASDA-A2-E EtherCAT(
ASDA_A2_E_CoE_Drive_7 (Delta ASDA-A2-E EtherCAT(
SoftMotion General Axis Pool

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



DFB_ResetECATSlave_0
DFB_ResetECATSlave
TRUE
EN ENO
TRUE — bExecute bDone — TRUE
1005 — uiSlaveAddr bBusy — FALSE
t#5s — tTimeout bError — FALSE
ErrorId — DFB_ECAT_D

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



```
191129
  AX_308EA0MA1T_1 [connected] (AX-308EA0MA1T)
    Plc Logic
      Application [run]
        Library Manager
        DFB_EtherCATLink_Diag_Test (PRG)
        DFB_ResetECATSlave_Test (PRG)
        PLC_PRG (PRG)
        PLC_PRG_1 (PRG)
        PLC_PRG_2 (PRG)
        Task Configuration
          EtherCAT_Task
            DFB_ResetECATSlave_Test
    BuiltIn_IO (BuiltIn_IO)
    Delta_LocalBus_Master (Delta LocalBus Master)
    EtherCAT_Master_SoftMotion (AX-308 Series EtherCAT Mas
      ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE
      ASDA_A2_E_CoE_Drive_1 (Delta ASDA-A2-E EtherCAT(C
      ASDA_A2_E_CoE_Drive_2 (Delta ASDA-A2-E EtherCAT(C
      ASDA_A2_E_CoE_Drive_3 (Delta ASDA-A2-E EtherCAT(C
      ASDA_A2_E_CoE_Drive_4 (Delta ASDA-A2-E EtherCAT(
      ASDA_A2_E_CoE_Drive_5 (Delta ASDA-A2-E EtherCAT(C
      ASDA_A2_E_CoE_Drive_6 (Delta ASDA-A2-E EtherCAT(C
        SM_Drive_ETC_Delta_ASDA_A2_6 (SM_Drive_ETC
      ASDA_A2_E_CoE_Drive_7 (Delta ASDA-A2-E EtherCAT(C
    SoftMotion General Axis Pool
```

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

- **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.

Chapter 5 Checksum Instructions

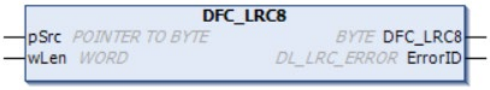
5.1 DFC_LRC8

- **Supported Products**

  - AX Series

DFC_LRC8: LRC (8-bit) checksum calculation.

FB/FC	Instruction	Graphic Expression	ST Language

FC	DFC_LRC8	 DFC_LRC8
		pSrc
		POINTER TO BYTE
		wLen
		WORD
		BYTE
		DFC_LRC8
		DL_LRC_ERROR
		ErrorID
			DFC_LRC8(
			pSrc:= ,
			wLen:= ,
			ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
pSrc	The start address for LRC calculation.	POINTER TO BYTE	Memory address 1~256 (0)
wLen	The data length for LRC calculation.	DWORD*	(0)

***Note 1:** The variable type BYTE and WORD can be used for dwLen input.
*__Note 2__: The memory address given by the pSrc exceeds the usable range, it may cause controller exceptions, such as %I, %Q and %M.

- **Output**

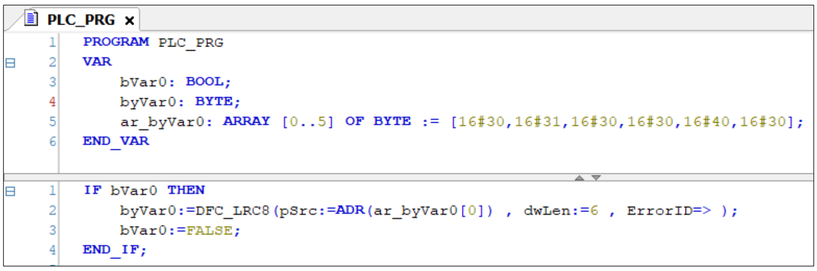
Name	Function	Data Type	Output Range(Default value)
DFC_LRC8	LRC checksum (Return type)	BYTE	(0)
ErrorID	Error codes	DL_LRC_ERROR	DL_LRC_ERROR(DFC_NO_ERROR)

- **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 wLen.

- **Programming Example**

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



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

```
IF bVar0 THEN
    byVar0:=DFC_LRC8(pSrc:=ADR(ar_byVar0[0]) , dwLen:=6 , ErrorID=> );
    bVar0:=FALSE;
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.

Note: In library version 1.0.0.1, change the input from dwLen to wLen.

- **Library**

  - DL_LRC.library

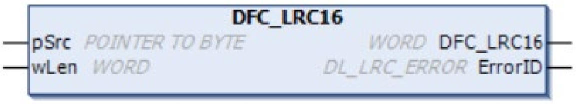
5.2 FC_LRC16

- **Supported Products**

  - AX Series

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 1~512 (0)
wLen	The data length for LRC calculation.	DWORD*	(0)

***Note 1:** The variable type BYTE and WORD can be used for dwLen input.
*__Note 2__: The memory address given by the pSrc exceeds the usable range,
it may cause controller exceptions, such as %I, %Q and %M.

- **Output**

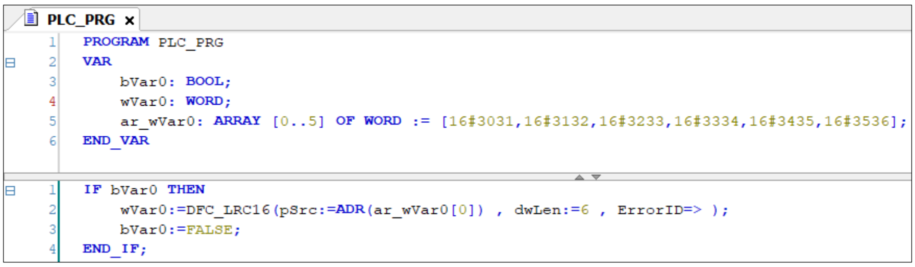
Name	Function	Data Type	Output Range(Default value)
DFC_LRC16	LRC checksum (Return type)	WORD	(0)
ErrorID	Error codes	DL_LRC_ERROR	DL_LRC_ERROR(DFC_NO_ERROR)

- **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 wLen.

- **Programming Example**

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



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

```
IF bVar0 THEN
    wVar0:=DFC_LRC16(pSrc:=ADR(ar_wVar0[0]) , dwLen:=6 , ErrorID=> );
    bVar0:=FALSE;
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.

Note: In library version 1.0.0.1, change the input from dwLen to wLen.

- **Library**

  - DL_LRC.library

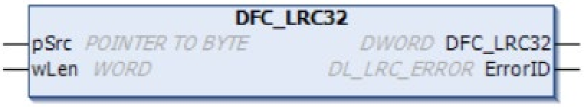
5.3 DFC_LRC32

- **Supported Products**

  - AX Series

DFC_LRC32: LRC (32-bit) checksum calculation.

FB/FC	Instruction	Graphic Expression	ST Language

FC	DFC_LRC32	 DFC_LRC32
		pSrc
		POINTER TO BYTE
		wLen
		WORD
		DWORD
		DFC_LRC32
		DL_LRC_ERROR
		ErrorID
			DFC_LRC32(
			pSrc:= ,
			wLen:= ,
			ErrorID=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
pSrc	The start address for LRC calculation.	POINTER TO BYTE	Memory address 1~512 (0)
dwLen	The data length for LRC calculation.	DWORD*	(0)

***Note 1:** The variable type BYTE and WORD can be used for dwLen input.
*__Note 2__: The memory address given by the pSrc exceeds the usable range, it may cause controller exceptions, such as %I, %Q and %M.

- **Output**

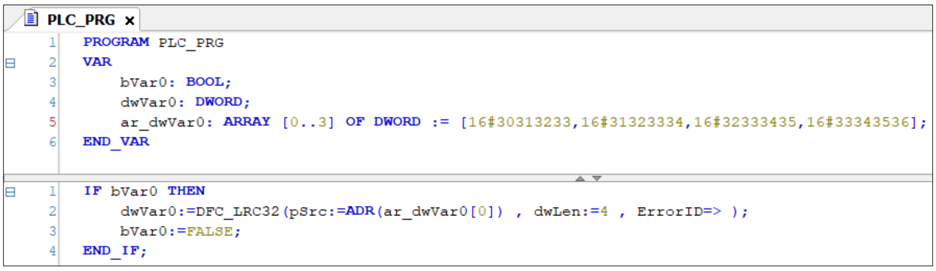
Name	Function	Data Type	Output Range(Default value)
DFC_LRC32	LRC checksum (Return type)	DWORD	(0)
ErrorID	Error codes	DL_LRC_ERROR	DL_LRC_ERROR(DFC_NO_ERROR)

- **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 wLen.

- **Programming Example**

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



PLC_PRG

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

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

Note: In library version 1.0.0.1, change the input from dwLen to wLen.

- **Library**

  - DL_LRC.library

5.4 Error Codes and Troubleshooting

Description	Cause of Error	Corrective Action
DFC_LRC_ERR_ PARAMETER	The value of wLen is incorrect.	Make sure the wLen value is greater than zero and does not exceed the upper limit.

Chapter 6 Module Read-write Instructions

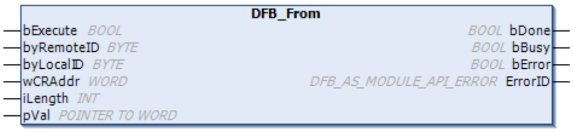
6.1 DFB_From

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_From: Read the CR data in the module.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_ From	 DFB_From
		bExecute BOOL
		byRemoteID BYTE
		byLocalID BYTE
		wCRAddr WORD
		iLength INT
		pVal POINTER TO WORD
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorID
			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 1~15: 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	1~8 (0)
pVal	The CR data to read.	POINTER TO WORD	

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

- **Output**

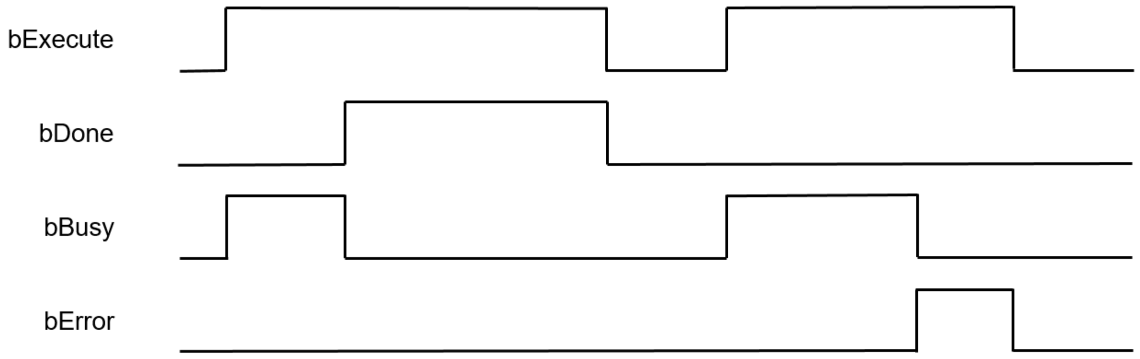
Name	Function	Data Type	Output Range(Default value)
bDone	True when the execution of the instruction is completed.	BOOL	True/False(False)

Name	Function	Data Type	Output Range(Default value)
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**

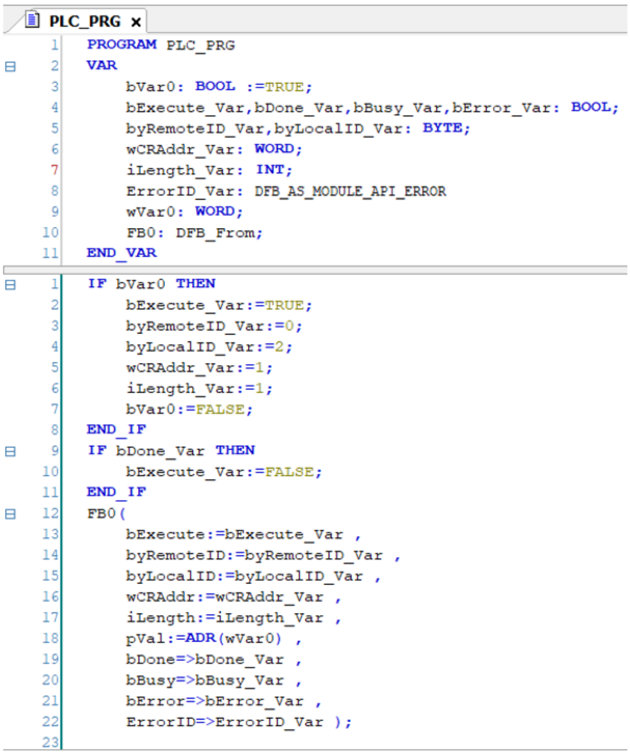


- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. 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.



```
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 );

```

- **Library**

  - DL_ASModuleAPI_AX3

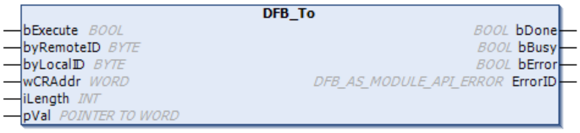
6.2 DFB_To

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

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 1~15: 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	1~8 (0)
pVal	The CR data to be written.	POINTER TO WORD	

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

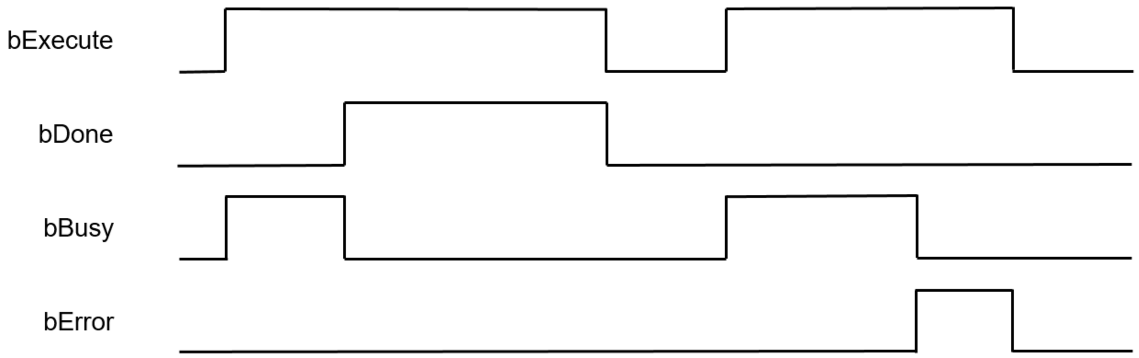
- **Output**

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**

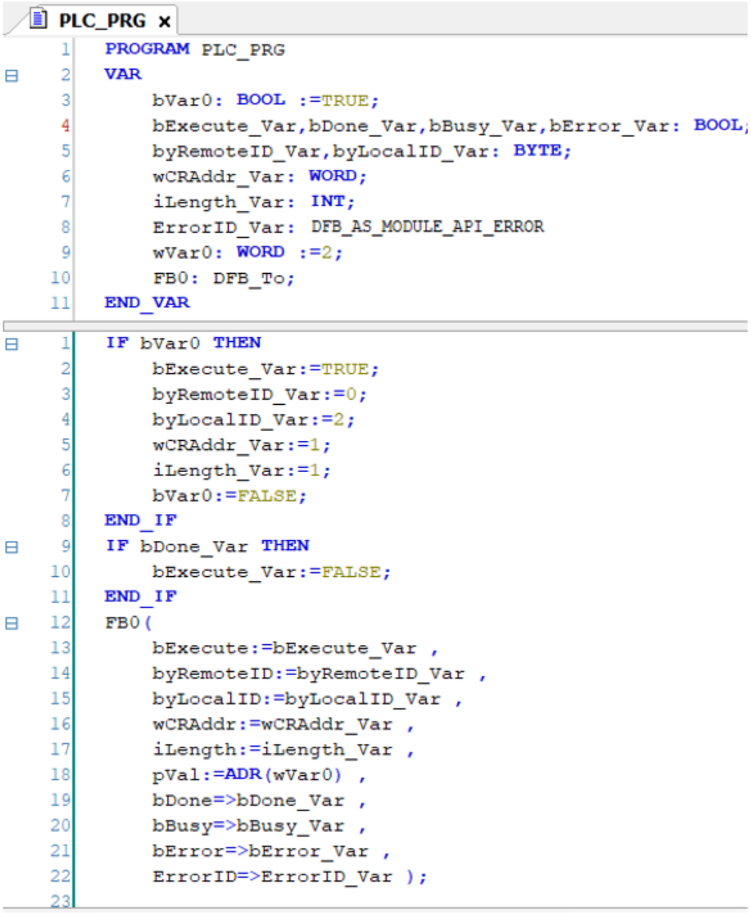


- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. 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.



```
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 );
```

- **Library**

  - DL_ASModuleAPI_AX3

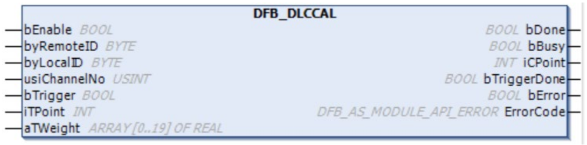
6.3 DFB_DLCCAL

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DLCCAL:AS02LC weighing module weight calibration instruction

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DLCCAL	 DFB_DLCCAL
		bEnable BOOL
		byRemoteID BYTE
		byLocalID BYTE
		usiChannelNo USINT
		bTrigger BOOL
		iTPoint INT
		aTWeight ARRAY[0..19] OF REAL
		BOOL bDone
		BOOL bBusy
		INT iCPoint
		BOOL bTriggerDone
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DLCCAL(
			bEnable:= ,
			byRemoteID:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			bTrigger:= ,
			iTPoint:= ,
			aTWeight:= ,
			bDone=> ,
			bBusy=> ,
			iCPoint=> ,
			bTriggerDone=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byRemoteID*	The CPU or remote module ID	BYTE	0: CPU 1~15: Remote module (0)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
usiChannelNo	Specify channel number	USINT	1 ~ 2(1)
bTrigger	Trigger single-point calibration	BOOL	True/False(False)
iTPoint	Total number of calibration points	INT	2~20(2)
aTWeight	Calibration weight value	ARRAY[0..19] OF REAL	1.0E-44~3.402823E+38(0)

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

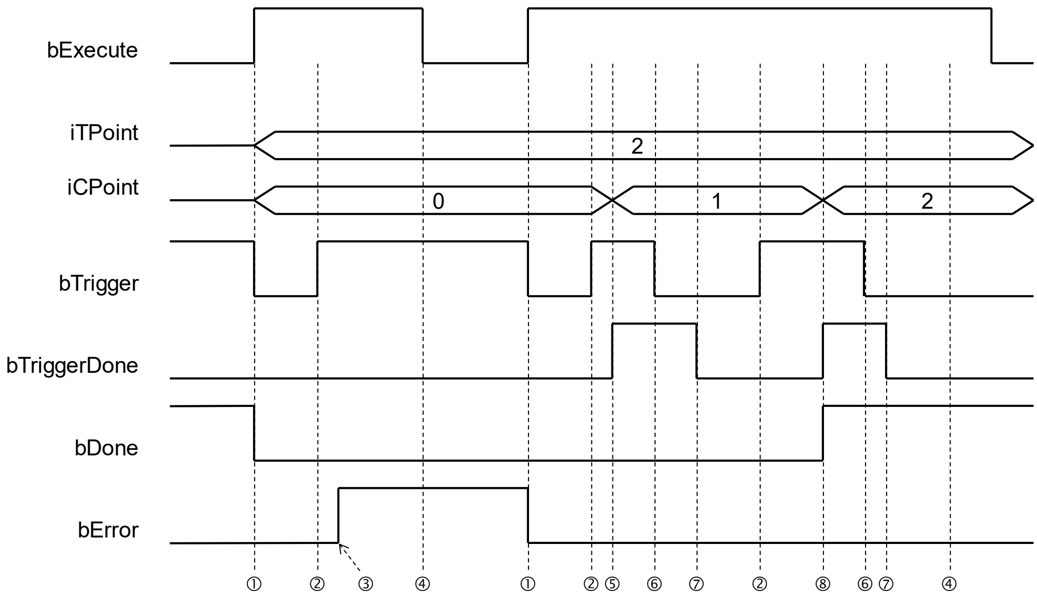
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	All calibration is done.	BOOL	True/False(False)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
iCPoint	Points number calibration is compeletd.	INT	0~20(0)
bTriggerDone	Single calibration is done.	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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 all calibration is done.	● When bExecute shifts to False.
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
iCPoint	● Add one when each calibration is done.	● When bEnable shifts to False, clear to zero.
bTriggerDone	● Every time when the calibration is done.	● When bEnable shifts to False.
bError	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**



The timing points in the timing diagram are described as below:
① → Instruction starts, automatically clear the iCPoint value and bTrigger, bTriggerDone, bDone & bError flags.
② → User-triggered calibration flag.
③ → Instruction judgement module number error.
④ → User closing instruction.
⑤ → After being triggered by users, LC module completes single-point calibration, iCPoint adds one, and bTriggerDone is set to ON.
⑥ → User clears the bTrigger trigger signal.
⑦ → The instruction follows to clear the bTriggerDone signal.
⑧ → After being triggered by users, LC modules are completely calibrated, iCPoint adds one, and bTriggerDone & bDone are set to ON.

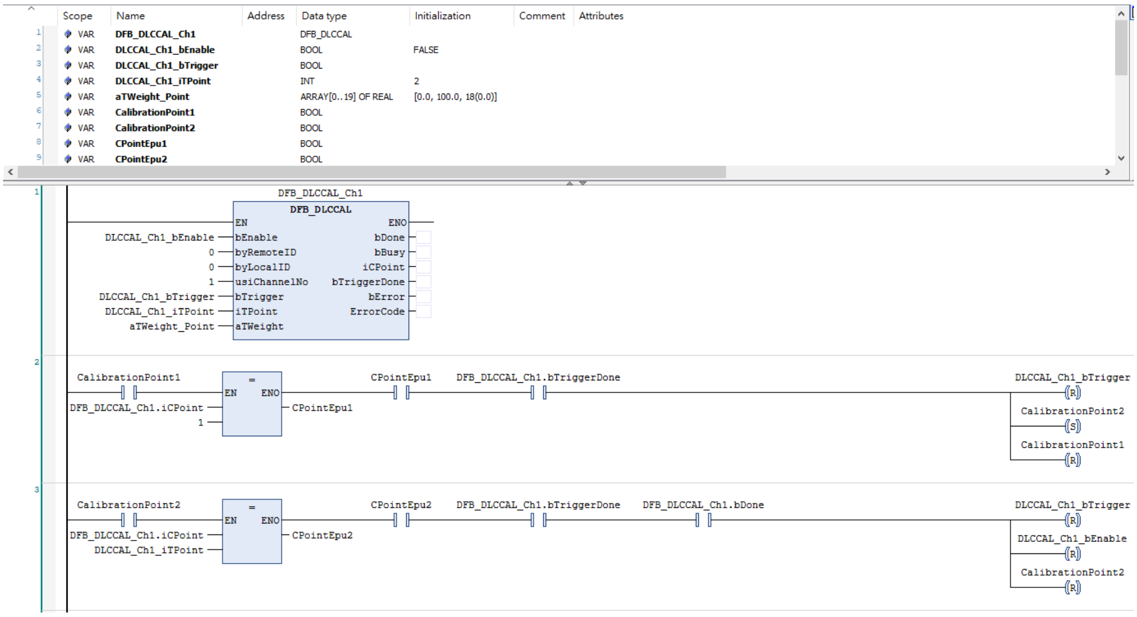
- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. This instruction is only for use with the AS02LC weighting module, and the supported version is AS02LC V1.04 and later.
4. This is weighting module (AS02LC-A) dedicated instruction, and its function is to activate/deactivate the weight calibration function of the module.
5. **byRemoteID** specifies that the weighting module should connect to the right of CPU or the right of the remote module group numbers. The CPU number is 0, the first remote module number is 1, and so on. The maximum group number is 15.
6. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
7. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.
8. **bTrigger** is the instruction that triggers single-point calibration. When the bTrigger status changes from OFF to ON, LC module will be notified to perform a single calibration, and bTriggerDone flag will be set to ON after completion. If the calibrations of all calibrated points are done, the bDone flag will also be ON. Before proceeding the next calibration point, users need to check the bTriggerDone flag is ON, and changes bTrigger to OFF. At this time, the instruction will also monitor the bTrigger flag when it turns from ON to OFF, and the bTriggerDone flag will be automatically cleared.

9. **iTPoint** is the total number of points for this calibration. After the instruction is activated by users, this value cannot be changed because this iTPoint has been sent to the LC module for calibration at the first activation.
10. **aTWeight** is the calibration weight value for each calibrated point, and the maximum number of calibration points for the LC module is 20. After the instruction is activated, users cannot change this value because this aTWeight has sent data to the LC module for calibration at the first activation. The first point calibration weight value must be 0, if it is not 0, the bError flag will be set to ON. For example, the total points of iTPoint calibration is 3, aTWeight gives [0.0, 100.0, 200.0, 17(0.0)] a total of 20 REAL-type ARRAY, of which 17(0.0) means there are 17 0.0.

- **Programming Example**

This example uses the FB instruction (DFB_DLCCAL) to calibrate the first channel in the first module on the right of CPU.



Scope	Name	Address	Data type	Initialization	Comment	Attributes
VAR	DFB_DLCCAL_Ch1		DFB_DLCCAL			
VAR	DLCCAL_Ch1_bEnable		BOOL	FALSE		
VAR	DLCCAL_Ch1_bTrigger		BOOL			
VAR	DLCCAL_Ch1_iTPoint		INT	2		
VAR	aTWeight_Point		ARRAY[0..19] OF REAL	[0.0, 100.0, 18(0.0)]		
VAR	CalibrationPoint1		BOOL			
VAR	CalibrationPoint2		BOOL			
VAR	CPointEpu1		BOOL			
VAR	CPointEpu2		BOOL			

1. Set CalibrationPoint1 to ON.
2. Make sure that the weight platform is unloaded first, set DLCCAL_CH1_bEnable to ON, and then set DLCCAL_CH1_bTrigger to ON. When the iCPoint value becomes 1, and bTriggerDone becomes TRUE, it means that this first point calibration is done.
3. Place 100.0g weights on the weight platform and after ensuring the platform is steady, set DLCCAL_CH1_bTrigger to ON. Now, iCPoint becomes 2, and bTriggerDone & bDone are TRUE, which means that all calibrations are done. After making the third network established, deactivate DFB_DLCCAL function block to complete this calibration.

- **Library**
  - DL_ASModuleAPI_AX3.library

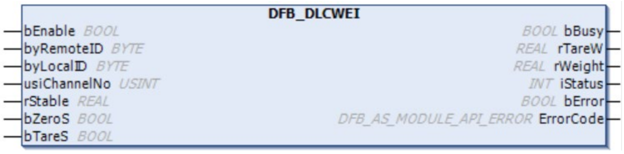
6.4 DFB_DLCWEI

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DLCWEI:AS02LC weighting module weight measurement instruction

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DLCWEI		DFB_DLCWEI(
			bEnable:= ,
			byRemoteID:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			rStable:= ,
			bZeroS:= ,
			bTareS:= ,
			bBusy=> ,
			rTareW=> ,
			rWeight=> ,
			iStatus=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byRemoteID*	The CPU or remote module ID	BYTE	0: CPU 1~15: Remote module (0)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
usiChannelNo	Specify channel number	USINT	1 ~ 2(1)
rStable	Set the weight stability range	REAL	0.0~100000.0 (0)
bZeroS	Set the weight to zero flag	BOOL	True/False(False)
bTareS	Set the tare weight flag	BOOL	True/False(False)

*Note: Currently only support mode 0.

- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
rTareW	Tare weight	REAL	1.0E-44~3.402823E+38(0)
rWeight	Current weight	REAL	1.0E-44~3.402823E+38(0)
iStatus*	LC module status code	INT	0~5(0)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	DFB_AS_MODULE_ API_ERROR	DFB_AS_MODULE_ API_ERROR(DFB_NO_ERROR)

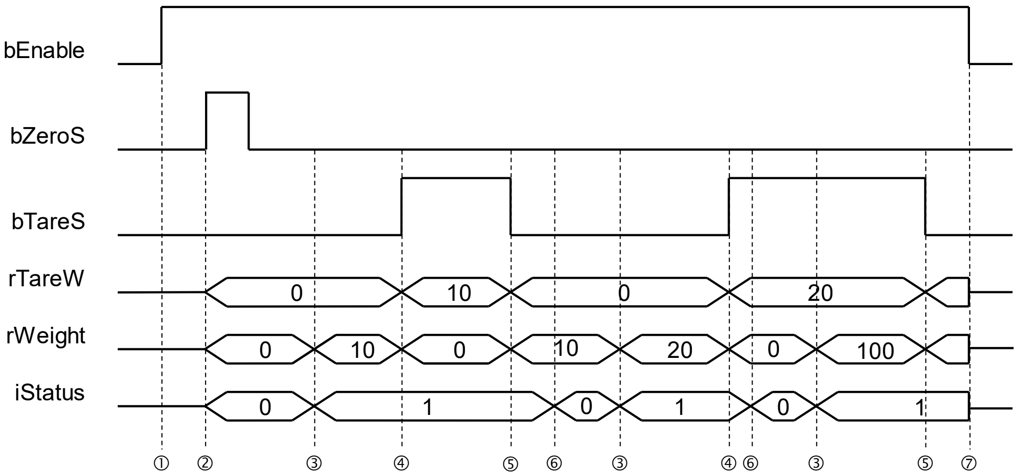
*__Note__: iStatus is the common status code of this instruction integrating the LC module. Its statuses are as follows:

Value	0	1	2	3	4	5
Description	Measuring or unloaded	Weight is steady.	Hardware/Calibration error	Calibrating	Weight is out of range.	Module number/Channel error

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
rTareW	● Continuously update the value when bEnable is TRUE.	● When bEnable shifts to False, clear to zero.
rWeight	● Continuously update the weight value when bEnable is TRUE	● When bEnable shifts to False, clear to zero.
iStatus	● Continuously update the status when bEnable is TRUE	● When bEnable shifts to False, clear to zero.
bError	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**



The timing points in the timing diagram are described as below:
① → Activate the instruction
② → When users order to clear to 0, the instruction will clear the rTareW & rWeight values and the iStatus status.
③ → Users place objects on the weight platform, and when the weight values is steady, iStatus becomes 1 and rWeight weight value is shown.
④ → Users order to set the tare weight to ON. Now, the rWeight weight is transferred to rTareW, and then the rWeight weight is cleared.
⑤ → Users clear the tare weight setting OFF. Now the rTareW weight is transferred back to rWeight, and then the rTareW weight is cleared.
⑥ → Users place another object on the weight platform again. Now, iStatus turns to measuring.
⑦ → Instruction is deactivated, and rTareW, rWeight, iStatus are cleared to 0.

- **Function**

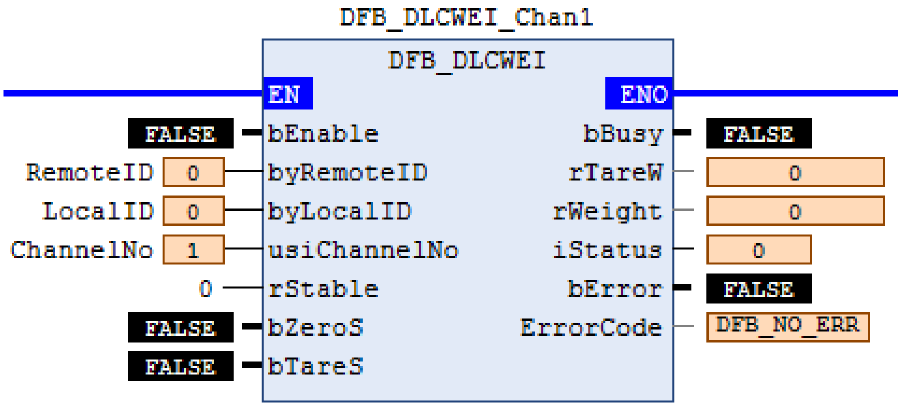
1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. This instruction is only for use with the AS02LC weighting module, and the supported version is AS02LC V1.04 and later.
4. This is weighting module (AS02LC-A) dedicated instruction, and its function is to activate/deactivate the weight measurement function of the module.
5. **byRemoteID** specifies that the weighting module should connect to the right of CPU or the right of the remote module group numbers. The CPU number is 0, the first remote module number is 1, and so on. The maximum group number is 15.
6. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
7. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.
8. **rStable** is the weight stability range value. Its data type is REAL, and the inputable floating-point number value range is 0.0~100000.0. If the setting is out of range, the instruction will automatically set the value to the minimum/maximum values. The timing of LC module parameter setting is when the instruction is started for the first time. To modify the value in the LC module later, users need to deactivate the instruction, set a new range value, and then open the instruction to reset.
9. After this instruction is activated, the specified channels will be automatically changed to “Net Weight” display mode.

If users need to know the “Gross weight (total weight)” value, add rTareW and rWeight.

10. **bZeroS** is the flag that sets the current weight to 0. When this flag is from OFF to ON, rTareW and rWeight will be cleared to 0.
11. **bTareS** is the flag that sets the tare weight. When the bTareS flag is from OFF to ON, the current rWeight weight will be transferred to rTareW, and the rWeight value will be cleared to 0. When the bTareS flag is from ON to OFF, rTareW will be back to the rWeight current weight value, and the rTareW value will be cleared to 0.
12. **rWeight** is the weight value after deducting the tare weight. Users can monitor if rTareW has a value to determine whether the tare function is activated. When the value is 0, it represents that the tare weight has not been set.

- **Programming Example**

After completing DLCCAL calibration, the DLCWEI instruction can used to perform weight measurement.



1. Weight measurement: Place 500g weights on the weight platform. When bEnable is On, rWeight displays the current weight 500.0.
2. Tare weight setting:
   - Place package material on the weight platform (e.g. 100 g). Now, rWeight shows that the current weight is 100.0.
   - When bTareS is ON, the rWeight weight will be transferred to rTareW, and then the rWeight weight will be cleared.
   - When rWeight = 0.0 (maybe a little unstable) and rTareW = 100.0, the tare weight setting is complete.
3. Clear tare weight setting:
   - When bTareS is OFF, the rTareW weight will be back to rWeight, and then the rTareW weight will be cleared.
   - When rWeight = 100.0 and rTareW = 0.0, tare weight setting is cleared.
4. Weight stability range setting (Stability checking function):
   - Before bEnable is ON, set rStable = 10.0.
   - After placing 500g weights and setting bEnable to ON, when the measurement range is between 490~510g, iStatus = 1 (Weight is stable).

- **Library**

- DL_ASModuleAPI_AX3.library

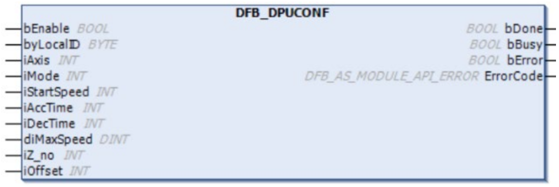
6.5 DFB_DPUCONF

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DPUCONF: parameter setting of PU module output control

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DPUCONF		DFB_DPUCONF(
			bEnable:= ,
			byLocalID:= ,
			iAxis:= ,
			iMode:= ,
			iStartSpeed:= ,
			iAccTime:= ,
			iDecTime:= ,
			diMaxSpeed:= ,
			iZ_no:= ,
			iOffset:= ,
			bDone=> ,
			bBusy=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iAxis	Output axis number	INT	1~4(1)
iMode	Output mode setting	INT	0~3(1)
iStartSpeed	Start/End speed	INT	0~10000 Hz(100)
iAccTime	Acceleration time	INT	0~10000 ms(100)
iDecTime	Deceleration time	INT	0~10000 ms(100)
diMaxSpeed	Maximum output frequency	DINT	AS02PU:100~200000 Hz(100K) AS04PU:100~100000 Hz(100K)
iZ_no	Homing function and find the number of z-phase signals	INT	-100~100 times(0)
iOffset	Homing function is done, and the z-phase is found, then output the offset position.	INT	-10000~10000 numbers(0)

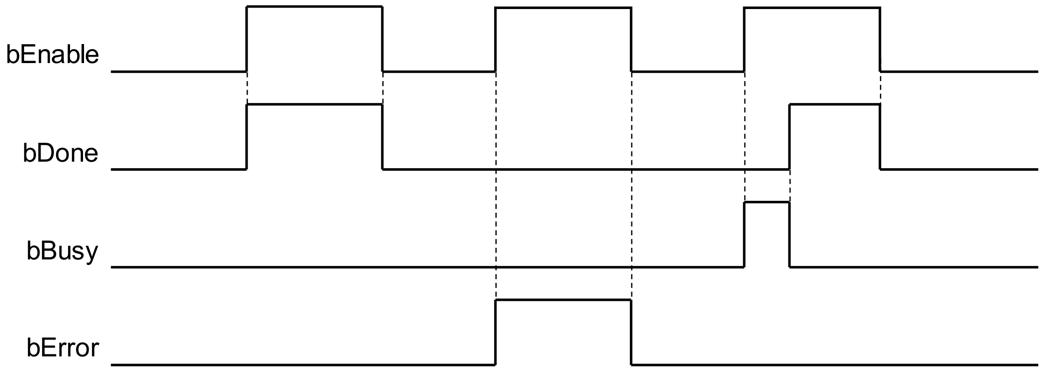
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	Parameter setting completion flag	BOOL	True/False(False)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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 parameter setting is done.	● When bEnable shifts to False.
bBusy	● When bEnable shifts to TRUE.	● When bEnable shifts to False.
bError	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32. This PU dedicated instruction is only for the PU module on the right of CPU and is **not** for the PU module on the right of the remote module. If the specified module is not the PU module, the bError flag will be set to ON.
4. **iAxis** is the axis number of the specified output PU module. The input values 1~4 respectively represent the specified PU module axis 1 ~ axis 4 output. If the PU module does not have this axis number, the bError flag will be set to ON. The combination of the axis number and the corresponding output points is as follows:

PU Module Name	Axis1 Combination	Axis2 Combination	Axis3 Combination	Axis4 Combination
AS02PU	Y0.0 / Y0.1	Y0.2 / Y0.3	NA	NA
AS04PU	Y0.0 / Y0.1	Y0.2 / Y0.3	Y0.4 / Y0.5	Y0.6 / Y0.7

5. **iMode** selects parameters for the output mode of setting the output axis, and the setting values are as shown in the following table:

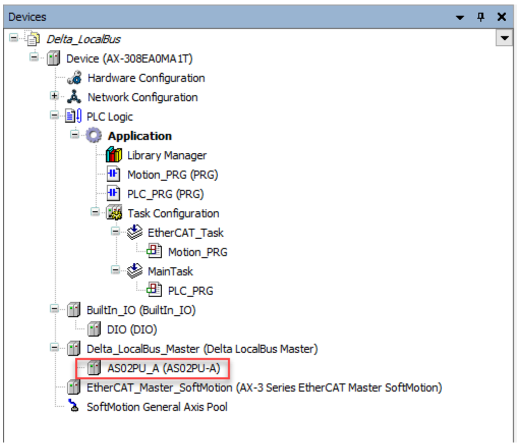
Output Mode Value	Mode	Note
N	Single-point pulse output (even-point output only)	For example: Y0.0 or Y0.2 output
1	Pulse (even points) + Direction (odd points)	For example: Y0.0 is pulse, Y0.1 is direction. When the direction is positive, Y0.1 is OFF; when the direction is negative, Y0.1 is ON.
2	CW (even points) + CCW (odd points)	For example: Y0.0 is CW (Positive direction), and Y0.1 is CCW (negative direction).
3	A-phase (even points) + B-phase (odd points)	For example: Y0.0 is A, and Y0.1 is B. When A is ahead of B, it represents positive direction output; when B is ahead of A, it represents negative direction output.
Others	Automatically change to mode 1 (Default)	-

6. **iStartSpeed~iOffset** are non-power outage persistence value. If the setting value is out of range, bError will be reported.
7. **bDone** is the output axis of the specified PU module, and is the parameter setting completion flag. When the flag is ON, it represents that parameter setting is successful. Users can perform the subsequent positioning output function according to the flag status. The bDone flag clearance needs to be executed by users, and this instruction will set this flag once when the setting is done.
8. **bError** is the output axis of the specified PU module, and is the parameter error flag. Because most of parameter ranges are automatically filtered by PLC, if this error flag occurs, it means that there is no specified PU module, the PU module number is incorrect, or the output axis number is incorrect.
9. This parameter setting instruction is pulse execution instruction. Even if the A contact method is used for the user conditional contact, this instruction will set the parameters of the PU module at the time when it’s started. Therefore, when the axis parameters are changed, please re-start the instruction and reset the parameters.
10. Because the parameter setting is ordered through the module communication, please check the bDone or bError flags of the setting results each time the parameters are modified, and then perform the related output action.

- **Programming Example**

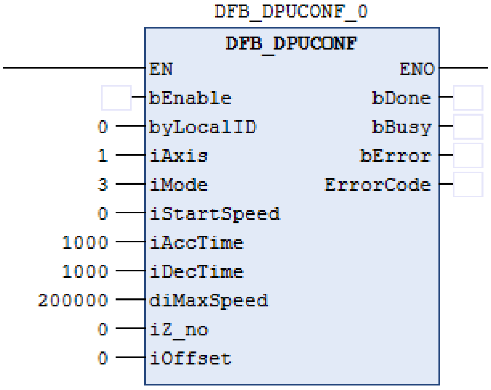
The following example shows hoe to execute the DFB_DPUCONF function block to set the parameter setting of the PU output module.

1. Delta_LocalBus_Master configures an AS02PU_A.



```
Devices
Delta_LocalBus
  Device (AX-308EA0MA1T)
    Hardware Configuration
    Network Configuration
    PLC Logic
      Application
        Library Manager
        Motion_PRG (PRG)
        PLC_PRG (PRG)
        Task Configuration
          EtherCAT_Task
            Motion_PRG
          MainTask
            PLC_PRG
    BuiltIn_IO (BuiltIn_IO)
      DIO (DIO)
    Delta_LocalBus_Master (Delta LocalBus Master)
      AS02PU_A (AS02PU-A)
    EtherCAT_Master_SoftMotion (AX-3 Series EtherCAT Master SoftMotion)
    SoftMotion General Axis Pool
```

2. Use the DFB_DPUCONF function block to set the first axis parameter of the 02PU module on the right.



```
                 DFB_DPUCONF_0
            DFB_DPUCONF
       EN                    ENO
       bEnable             bDone
     0 byLocalID           bBusy
     1 iAxis              bError
     3 iMode           ErrorCode
     0 iStartSpeed
  1000 iAccTime
  1000 iDecTime
200000 diMaxSpeed
     0 iZ_no
     0 iOffset
```

- **Library**

  ▪ DL_ASModuleAPI_AX3.library

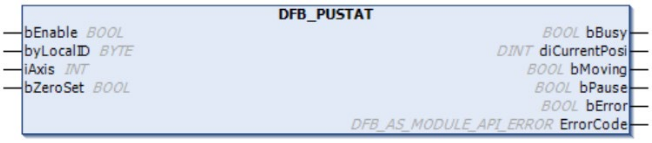
6.6 DFB_PUSTAT

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_PUSTAT: PU module output status read-back

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_ PUSTAT	 DFB_PUSTAT
		bEnable BOOL
		byLocalID BYTE
		iAxis INT
		bZeroSet BOOL
		BOOL bBusy
		DINT diCurrentPosi
		BOOL bMoving
		BOOL bPause
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_PUSTAT(
			bEnable:= ,
			byLocalID:= ,
			iAxis:= ,
			bZeroSet:= ,
			bBusy=> ,
			diCurrentPosi=> ,
			bMoving=> ,
			bPause=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iAxis	Output axis number	INT	1~4(1)
bZeroSet	The current output location is cleared to 0.	BOOL	True/False(False)

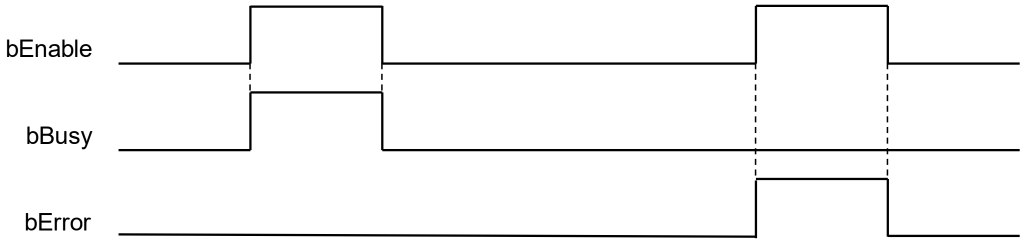
- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
diCurrentPosi	Current output position	DINT	0~ 2,147,483,648(0)
bMoving	Output executing flag	BOOL	True/False(False)
bPause	Output pause flag	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
diCurrentPosi	● Continuously updating when bEnable shifts to True.	● When bEnable shifts to False.
bMoving	● If the module is outputting pulse after bEnable shifts to True.	● When bEnable shifts to False.
bPause	● If the module is not outputting pulse after bEnable shifts to True.	● When bEnable shifts to False.
bError ErrorCode	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.

- **Timing Diagram**



- **Function**

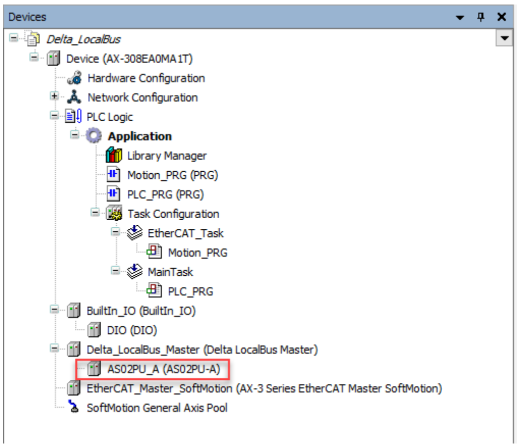
1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32. This PU dedicated instruction is only for the PU module on the right of CPU and is **not** for the PU module on the right of the remote module. If the specified module is not the PU module, the bError flag will be set to ON.
4. **iAxis** is the axis number of the specified output PU module. The input values 1~4 respectively represent the specified PU module axis 1 ~ axis 4 output. If the PU module does not have this axis number, the bError flag will be set to ON.
5. **diCurrentPosi** is the current position of the output axis of the specified PU module. This values is outage persistent, and is stored in the PU module. If users want to clear this value, clear the bZeroSet flag to 0 (OFF → ON) when the instruction is started.
6. **bMoving** is the output executing flag (Read-only) of the output axis of the PU module. When this flag is ON, it means that the output is in progress; when the flag is OFF, it means that the output axis is not being used and can accept the next output instruction.
7. **bPause** is the output pause flag (Read-only) of the output axis of the PU module. When this flag is ON, it means that the output is paused, the current speed is 0, and the current position has not yet reached the target position of the specified output. If users resume the output, this flag will be automatically cleared. Note: When the bPause flag is ON, the bMoving flag becomes OFF.

8. **BError** is the specified PU module and is the read error flag (Read-only). When the error occurs, please refer to ErrorCode description.

- **Programming Example**

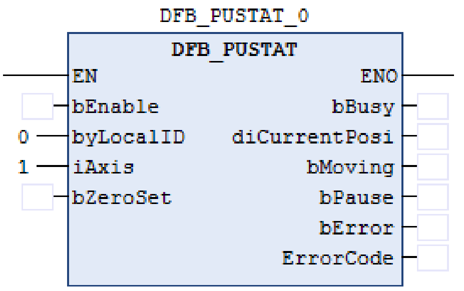
This example shows how to execute the DFB_PUSTAT function block to set the current output position of the first axis of the PU output module.

1. Delta_LocalBus_Master configures an AS02PU_A.



```
Delta_LocalBus
  Device (AX-308EA0MA1T)
    Hardware Configuration
    Network Configuration
    PLC Logic
      Application
        Library Manager
        Motion_PRG (PRG)
        PLC_PRG (PRG)
        Task Configuration
          EtherCAT_Task
            Motion_PRG
          MainTask
            PLC_PRG
    BuiltIn_IO (BuiltIn_IO)
      DIO (DIO)
    Delta_LocalBus_Master (Delta LocalBus Master)
      AS02PU_A (AS02PU-A)
    EtherCAT_Master_SoftMotion (AX-3 Series EtherCAT Master SoftMotion)
    SoftMotion General Axis Pool
```

2. Use the DFB_PUSTAT function block to set the current output position of the first axis of the 02PU module.



DFB_PUSTAT_0

DFB_PUSTAT

	EN	ENO	
	bEnable	bBusy	
0	byLocalID	diCurrentPosi	
1	iAxis	bMoving	
	bZeroSet	bPause	
		bError	
		ErrorCode	

- **Library**

  - DL_ASModuleAPI_AX3.library

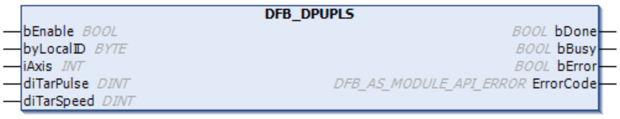
6.7 DFB_DPUPLS

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DPUPLS: PU module pulse output (No acceleration/deceleration)

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DPUPLS	 DFB_DPUPLS
		bEnable BOOL
		byLocalID BYTE
		iAxis INT
		diTarPulse DINT
		diTarSpeed DINT
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DPUPLS(
			bEnable:= ,
			byLocalID:= ,
			iAxis:= ,
			diTarPulse:= ,
			diTarSpeed:= ,
			bDone=> ,
			bBusy=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iAxis	Output axis number	INT	1~4(1)
diTarPulse	Target output number	DINT	-2,147,483,648~ 2,147,483,648(0)
diTarSpeed	Target output frequency (Unit: Hz)	DINT	AS02PU: -200K~200K(0) AS04PU: -100K~100K(0)

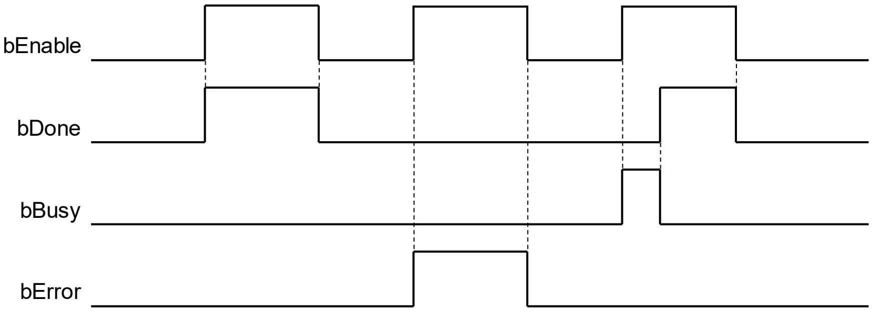
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	Pulse output completion flag	BOOL	True/False(False)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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 pulse output is done.	● When bEnable shifts to False.
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
bError	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32. This PU dedicated instruction is only for the PU module on the right of CPU and is **not** for the PU module on the right of the remote module. If the specified module is not the PU module, the bError flag will be set to ON.
4. **iAxis** is the axis number of the specified output PU module. The input values 1~4 respectively represent the specified PU module axis 1 ~ axis 4 output. If the PU module does not have this axis number, the bError flag will be set to ON.
5. **diTarPulse** is the pulse number of the specified output. The pulse number that can be input is the signed numbers 32-bit positive value. When the vale is 0, it means keeping outputting and the output numbers are not limited until the instruction is deactivated and the output stops; when the value is less than 0, PLC will automatically use the 2's complement method to convert to the number output of a positive integer number.
6. **diTarSpeed** is the target speed (Unit: Hz) of the specified output and the number that can be input is the signed numbers 32-bit value. Users can modify the target frequency any time after the instruction starts output, and the PU module will switch to the latest target frequency after outputting a complete pulse. Note: Before changing the target frequency, please consider if the change speed and the PLC scan time are appropriate. The corresponding diTarSpeed setting range of the module is as follows:

PU Module Name	diTarSpeed Setting Range
AS02PU	-200,000(-200K) ~ 200,000(200K)
AS04PU	-100,000(-100K) ~ 100,000(100K)

7. When the target speed of diTarSpeed is positive (>0), it means that the output point of "positive direction" is OFF;

when the target speed of diTarSpeed is negative (<0), it means that the output point of “negative direction” is ON; when the target speed of diTarSpeed is 0, it means that after outputting a complete executing pulse, it enters the pause output status.

8. This output instruction does not provide the acceleration/deceleration function. For acceleration/deceleration function requirement, please use the DPUDRI instruction.
9. This output instruction can be used for changing speed. When the instruction is executing output, users can change the target frequency value of diTarSpeed to achieve the purpose of changing the output speed.
10. When the output has reached the specified diTarPulse pulse number, the bDone completion flag will be set to ON. The bDone flag clearance needs to be executed by users, and this instruction will set this flag once when the setting is done
11. If any error situation occurs during output startup, the bError error flag will be set to ON. Users can refer to ErrorCode for troubleshooting.

- **Library**

  - DL_ASModuleAPI_AX3.library

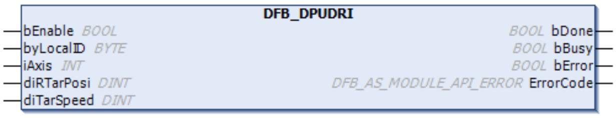
6.8 DFB_DPUDRI

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DPUDRI: PU module relative positioning output (With acceleration/deceleration)

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DPUDRI	 DFB_DPUDRI
		bEnable BOOL
		byLocalID BYTE
		iAxis INT
		diRTarPosi DINT
		diTarSpeed DINT
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DPUDRI(
			bEnable:= ,
			byLocalID:= ,
			iAxis:= ,
			diRTarPosi:= ,
			diTarSpeed:= ,
			bDone=> ,
			bBusy=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iAxis	Output axis number	INT	1~4(1)
diRTarPosi	Number of relative positioning outputs	DINT	-2,147,483,648~ 2,147,483,648(0)
diTarSpeed	Target output frequency (Unit: Hz)	DINT	AS02PU: -200K~200K(0) AS04PU: -100K~100K(0)

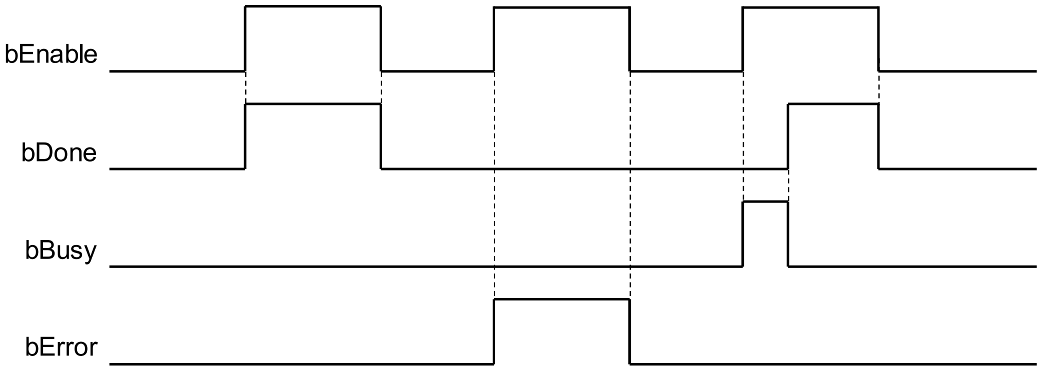
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	Pulse output completion flag	BOOL	True/False(False)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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 pulse output is done.	• When bEnable shifts to False.
bBusy	• When bEnable shifts to True.	• When bEnable shifts to False.
bError	• When FB instruction execution is incorrect or the input value is incorrect.	• When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**

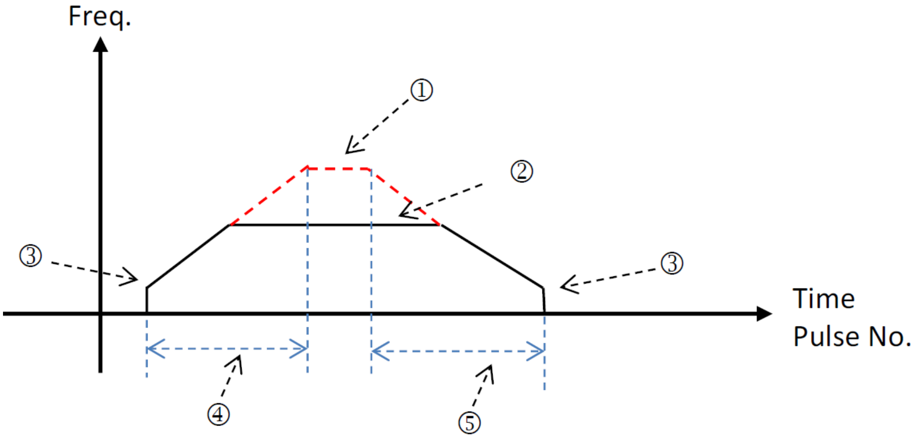


- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32. This PU dedicated instruction is only for the PU module on the right of CPU and is **not** for the PU module on the right of the remote module. If the specified module is not the PU module, the bError flag will be set to ON.
4. **iAxis** is the axis number of the specified output PU module. The input values 1~4 respectively represent the specified PU module axis 1 ~ axis 4 output. If the PU module does not have this axis number, the bError flag will be set to ON.
5. **diRTarPosi** is the position of the specified output relative positioning, and the pulse number that can be input is the signed numbers 32-bit value. When the value is greater than 0, the output is toward the positive direction (direction output point OFF); when the value is less than 0, the output is toward the negative direction (direction output point ON); when the value equals 0, this instruction will instantly set the bDone output completion flag to ON.
6. **diTarSpeed** is the target speed (Unit: Hz) of the specified output and the frequency value that can be input is the signed numbers 32-bit value. When the value is less than 0, the instruction will use the 2’s complement method to convert to a positive integer number; when the values equals to 0, the instruction will inform the module to enter the pause mode. The actual output will decelerate according to deceleration slope until the output speed reaches 0, and users set the pause flag (refer to the PUSTAT instruction). The corresponding diTarSpeed setting range of the module is as follows:

PU Module Name	diTarSpeed Setting Range
AS02PU	-200,000(-200K) ~ 200,000(200K)
AS04PU	-100,000(-100K) ~ 100,000(100K)

7. After starting the output, the target frequency can be changed anytime; however, when the frequency is actually changed, PLC will automatically change the frequency according to the acceleration/deceleration rate slope set by the DPUCONF instruction.
8. When the output has reached the specified diRTarPosi relative positioning position, the bDone completion flag will be set to ON. The bDone flag clearance needs to be executed by users, and this instruction will set this flag once when the setting is done.
9. During the output startup process, if any error situation occurs, the bError flag will be set to ON. Users can refer to ErrorCode to perform troubleshooting.
10. Acceleration/Deceleration curve of the PU module positioning output instruction is as follows:



① The setting value of the maximum output frequency. Please refer to the DPUCONF instruction setting for this parameter.

② The target frequency specified by the PU module output instruction. The target frequency cannot output the frequency that exceeds the maximum output frequency. The output will be limited to the maximum output frequency.

③ Start/End output frequency setting value. Refer to the DPUCONF instruction setting for this parameter.

④ Acceleration time setting value. Refer to the DPUCONF instruction setting for this parameter.

④ Deceleration time setting value. Refer to the DPUCONF instruction setting for this parameter.

The acceleration/deceleration of the PU module is a fixed slope, so the actual acceleration/deceleration time will change according to the target frequency of the specified output. The acceleration/deceleration slope conversion formulas are (1) (maximum output frequency - startup frequency) / acceleration time, and (2) (maximum output frequency - end frequency) / deceleration time.

- **Library**
  - DL_ASModuleAPI_AX3.library

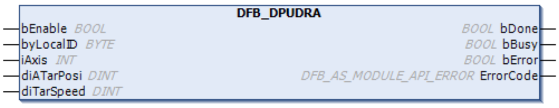
6.9 DFB_DPUDRA

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DPUDRA: PU module absolute positioning output (With acceleration/deceleration)

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DPUDRA	 DFB_DPUDRA
		bEnable BOOL
		byLocalID BYTE
		iAxis INT
		diATarPosi DINT
		diTarSpeed DINT
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DPUDRI(
			bEnable:= ,
			byLocalID:= ,
			iAxis:= ,
			diATarPosi:= ,
			diTarSpeed:= ,
			bDone=> ,
			bBusy=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iAxis	Output axis number	INT	1~4(1)
diATarPosi	Number of relative positioning outputs	DINT	-2,147,483,648~ 2,147,483,648(0)
diTarSpeed	Target output frequency (Unit: Hz)	DINT	AS02PU: -200K~200K(0) AS04PU: -100K~100K(0)

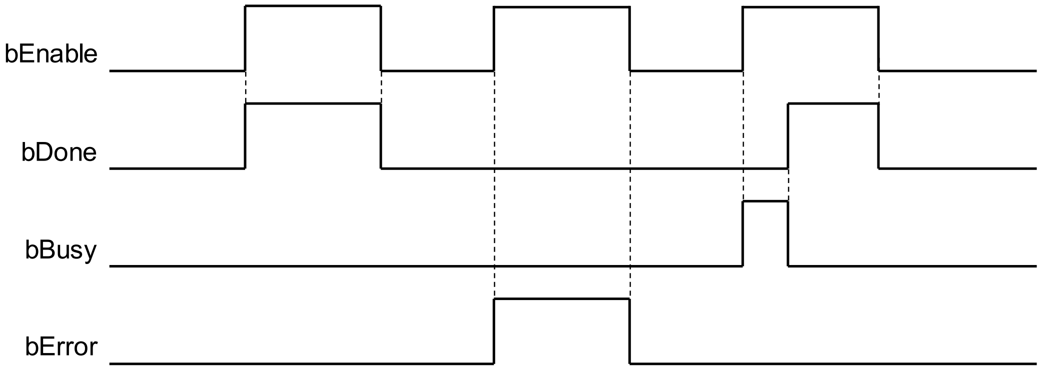
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	Pulse output completion flag	BOOL	True/False(False)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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 pulse output is done.	● When bEnable shifts to False.
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
bError	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. **diATarPosi** is the position of the absolute positioning of the specified output. Its pulse number that can be input is the signed numbers 32-bit value. The PU module will automatically compare the current position in the record. After comparison, if the value is greater than 0, it means that the output is toward the positive direction; if the value is less than 0, it means that the output is toward the negative direction; if the value is 0, this instruction will instantly set the bDone output completion to ON.
4. Refer to the DPUDRI instruction for the description of other parameters.

- **Library**

▪ DL_ASModuleAPI_AX3.library

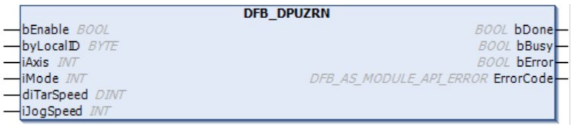
6.10 DFB_DPUZRN

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DPUZRN: PU module homing

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DPUZRN	 DFB_DPUZRN
		bEnable BOOL
		byLocalID BYTE
		iAxis INT
		iMode INT
		diTarSpeed DINT
		iJogSpeed INT
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DPUZRN(
			bEnable:= ,
			byLocalID:= ,
			iAxis:= ,
			iMode:= ,
			diTarSpeed:= ,
			iJogSpeed:= ,
			bDone=> ,
			bBusy=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iAxis	Output axis number	INT	1~4(1)
iMode	Homing mode selection	INT	0~8, 255(0)
diTarSpeed	Homing maximum output frequency	DINT	AS02PU:-200K~-100 和 100~200K(100) AS04PU: -100K~-100 100~100K(100)
iJogSpeed	Homing inching output frequency	INT	1~10000(1)

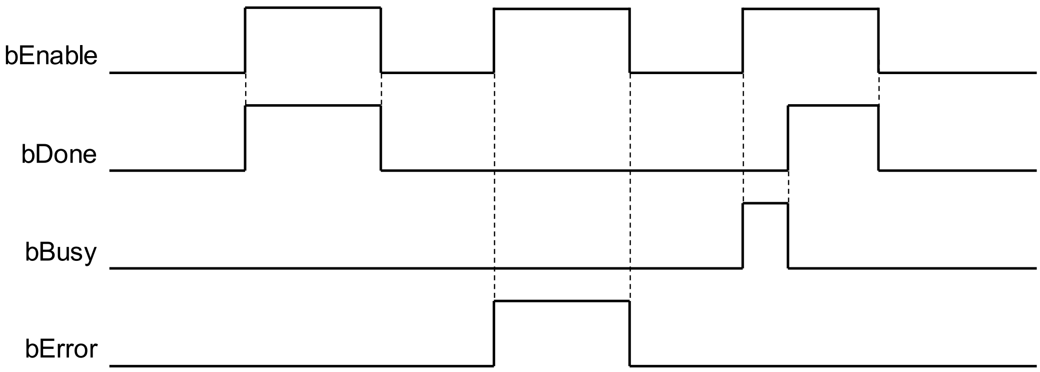
- **Output**

Name	Function	Data Type	Output Range(Default value)
bDone	Completion flag	BOOL	True/False(False)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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 pulse output is done.	● When bEnable shifts to False.
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
bError ErrorCode	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32. This PU dedicated instruction is only for the PU module on the right of CPU and is **not** for the PU module on the right of the remote module. If the specified module is not the PU module, the bError flag will be set to ON.
4. **iAxis** is the axis number of the specified output PU module. The input values 1~4 respectively represent the specified PU module axis 1 ~ axis 4 output. If the PU module does not have this axis number, the bError flag will be set to ON.
5. **iMode** is the mode selection for homing. The mode description is as shown below:

Mode Parameter	Function	Select Matching Input Point (Parameter Setting of PU Module)	Remark
0	Directly clear the current position to 0.	None	
1	Stops when the negative direction leaves the origin.	DOG	
2	Stops when the positive direction leaves the origin.	DOG	
3	Look for z-phase times after mode 1 is done.	DOG and Z-phase input	Set Z-phase times with DPUCONF
4	Look for z-phase times after mode 2 is done.	DOG and Z-phase input	
5	Output the offset position after mode 1 is done.	DOG	Set output offset position with DPUCONF

Mode Parameter	Function	Select Matching Input Point (Parameter Setting of PU Module)	Remark
6	Output the offset position after mode 2 is done.	DOG	
7	After mode 1 is done, look for z-phase first, then output the offset position	DOG and Z-phase input	Set z-phase times and output the offset position with DPUCONF
8	After mode 2 is done, look for z-phase first, then output the offset position	DOG and Z-phase input	
255	Modify the current output position of the axis	None	Used with the **TarSpeed** parameter
Others	Reserved		

Note: If the mode selects the required input points that are not matched with the Parameter setting of the PU module, the homing function may fail.

Note: For the above **iMode 1~4**, the PU module firmware version V1.02.00, when the action is complete, the current output position of the axis will not be cleared to 0, and it can be cleared with the PUSTAT instruction. For the PU module firmware version V1.02.10 and later, when the action is complete, the current output position will be cleared to 0.

Note: For the above **iMode** 5~8, the PU module firmware version V1.02.00, when the action is complete, the current output position is now the output result, and the current position must be modified to the specified position with the mode 255. For the PU module firmware version V1.02.10 (included) and later, when the action is complete, the axis current output position will be cleared to 0.

6. When **diTarSpeed** is selected to be mode is 1~8, it is the highest output speed for specifying to return to the origin, and this value is the signed numbers 32-bit value. Positive/Negative represents the default startup direction to fine the origin. For example, positive means that the instruction starts to look for the origin from the positive direction; the corresponding diTarSpeed setting range of the module is as follows: (If the mode parameter is specified as 255, the diTarSpeed will become the value of updating the PU module current position.)

PU Module Name	When Mode is 1~8, diTarSpeed Setting Range
AS02PU	-200,000 ~ -100(Hz) and 100 ~ 200,000(Hz)
AS04PU	-100,000 ~ -100(Hz) and 100 ~ 100,000(Hz)

7. **iJogSpeed** is the inching output speed when it touches the origin. This value is the signed number 16-bit value, and the setting value is 1~10,000 Hz.

8. When the output has reached the specified origin position, the bDone completion flag will be se5t to ON. This bDone flag clearance needs to be executed by users, and this instruction will set this flag once when the output is done.

9. During the output startup process, if any error situation occurs, the bError flag will be set to ON. Users can refer to ErrorCode to perform troubleshooting.

- **Library**

  - DL_ASModuleAPI_AX3.library

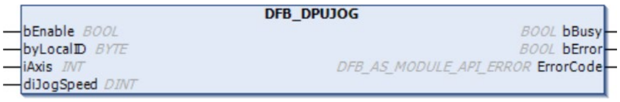
6.11 DFB_DPUJOG

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DPUJOG: PU module inching output

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DPUJOG	 DFB_DPUJOG
		bEnable BOOL
		byLocalID BYTE
		iAxis INT
		diJogSpeed DINT
		BOOL bBusy
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DPUJOG(
			bEnable:= ,
			byLocalID:= ,
			iAxis:= ,
			diJogSpeed:= ,
			bBusy=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iAxis	Output axis number	INT	1~4(1)
diJogSpeed	Inching output frequency	DINT	AS02PU: -200K~200K(0) AS04PU: -100K~100K(0)

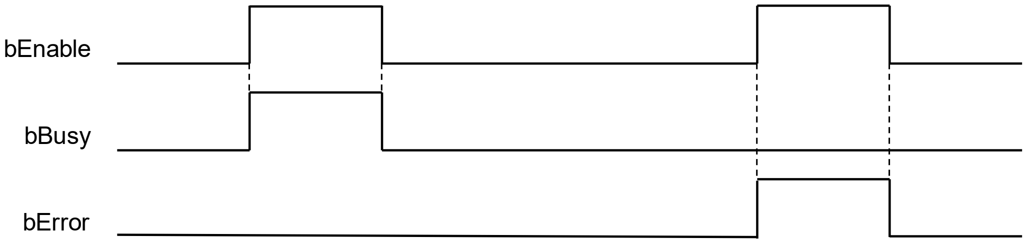
- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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
bBusy	• When bEnable shifts to True.	• When bEnable shifts to False.
bError ErrorCode	• When FB instruction execution is incorrect or the input value is incorrect.	• When bEnable shifts to False.

- **Timing Diagram**



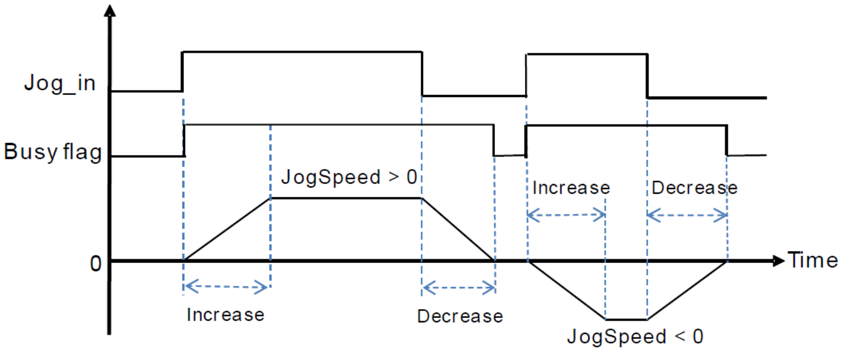
- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32. This PU dedicated instruction is only for the PU module on the right of CPU and is **not** for the PU module on the right of the remote module. If the specified module is not the PU module, the bError flag will be set to ON.
4. **iAxis** is the axis number of the specified output PU module. The input values 1~4 respectively represent the specified PU module axis 1 ~ axis 4 output. If the PU module does not have this axis number, the bError flag will be set to ON.
5. **diJogSpeed** is the output speed of the specified inching, and this value is the signed number 32-bit value. When the value is greater than 0, it means that the output is toward the positive direction (Direction output point is OFF); when the value is less than 0, it means that the output is toward the negative direction (Direction output point is ON); when the value is 0, it means that the output stops. The corresponding diJogSpeed setting range of the module is as follows:

PU Module Name	JogSpeed Setting Range
AS02PU	-200,000(-200K) ~ 200,000(200K)
AS04PU	-100,000(-100K) ~ 100,000(100K)

6. If any error situation occurs during output startup, the bError error flag will be set to ON. Users can refer to ErrorCode for troubleshooting.

7. The output timing diagram of the DPUJOG instruction is as follows: (In the timing diagram, Jog_in is the instruction startup switch, and Busy flag is the flag of bBusy being executing.)



8. When the DPUJOG instruction is deactivated, it can perform other output control after the bBusy flag is already OFF.

- **Library**

  - DL_ASModuleAPI_AX3.library

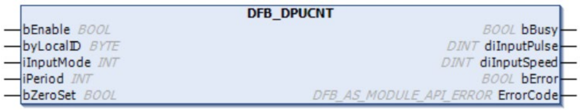
6.12 DFB_DPUCNT

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DPUCNT: PU module high-speed counter function

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DPUCNT	 DFB_DPUCNT
		bEnable BOOL
		byLocalID BYTE
		iInputMode INT
		iPeriod INT
		bZeroSet BOOL
		BOOL bBusy
		DINT diInputPulse
		DINT diInputSpeed
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DPUCNT(
			bEnable:= ,
			byLocalID:= ,
			iInputMode:= ,
			iPeriod:= ,
			bZeroSet:= ,
			bBusy=> ,
			diInputPulse=> ,
			diInputSpeed=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)
iInputMode	Encoder input mode and counting frequency multiplication setting	INT	1~7(4)
iPeriod	Speed fetch cycle time	INT	10~1000(1000)
bZeroSet	Counter is cleared to 0.	BOOL	True/False(False)

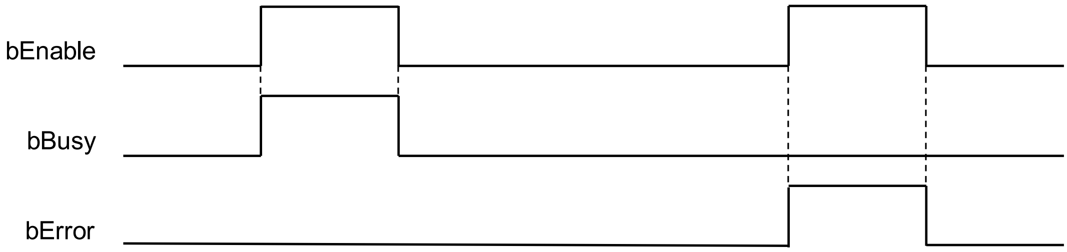
- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	Indicates that the function block is being executed.	BOOL	True/False(False)
diInputPulse	The entered number display	DINT	0~ 2,147,483,648(0)
diInputSpeed	Counting number of every cycle time	DINT	0~ 2,147,483,648(0)
bError	FB instruction error flag.	BOOL	True/False(False)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
diInputPulse	● Continuously updating after bEnable shifts to True.	● When bEnable shifts to False.
diInputSpeed	● Continuously updating after bEnable shifts to True.	● When bEnable shifts to False.
bError	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. This instruction is supported only by the AS02PU module.
4. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32. This PU dedicated instruction is only for the PU module on the right of CPU and is **not** for the PU module on the right of the remote module. If the specified module is not the PU module, the bError flag will be set to ON.
5. **iInputMode** is the encoder source input mode and counting frequency multiplication setting, and its setting parameters are as follows:
   Note: When A-phase is ahead of B-phase, it represents the positive counting. When B-phase is ahead of A-phase, it represents the negative counting.

iInputMode Setting Description	
Value	Input Mode (Set the following reserved setting, and the module will execute in default values.)
16#0000	Reserved
16#0001	1 time frequency multiplication A/B phase input
16#0002	2 times frequency multiplication A/B phase input
16#0003	Reserved
16#0004	4 times frequency multiplication A/B phase input (default value)
16#0005	Pulse + direction input (A+/A- is pulse input; B+/B- is direction input.) When B-phase is Off, it represents positive counting, and when it is On, it represents negative counting. A-phase uses rising edge-triggered counting.
16#0006	Pulse + direction input (A+/A- is pulse input; B+/B- is direction input.) When B-phase is On, it represents positive counting, and when it is Off, it represents negative counting. A-phase uses rising edge-triggered counting.
16#0007	Single-phase pulse input (A+/A- is pulse input.) A-phase uses rising edge-triggered counting.

iInputMode Setting Description	
Value	Input Mode (Set the following reserved setting, and the module will execute in default values.)
Other value	Automatically transferred to (default value).

6. **iPeriod** is the cycle time setting value of speed fetching, and the settable range is 10ms ~ 1000ms. If the value is out of range, it will be set to the minimum/maximum value.
7. **diInputPulse** displays the entered pulse counting numbers (signed number 32-bit values), which are outage persistence value. If there is a need to clear it to 0, during the instruction startup, use the bZeroS setting flag (OFF → ON) to perform the clearance.
8. **diInputSpeed** displays values that each iPeriod time counts (signed number 32-bit values). If users need to convert the values in Hz, please use calculation formulas.
9. When the instruction is activated/ deactivated, it means that CPU needs to inform the high-speed counter of the module to activate/deactivate. This instruction cannot be used with API1409 DPUMPG at the same time, otherwise the instructions may activate or deactivate the module counting with each other.

- **Library**

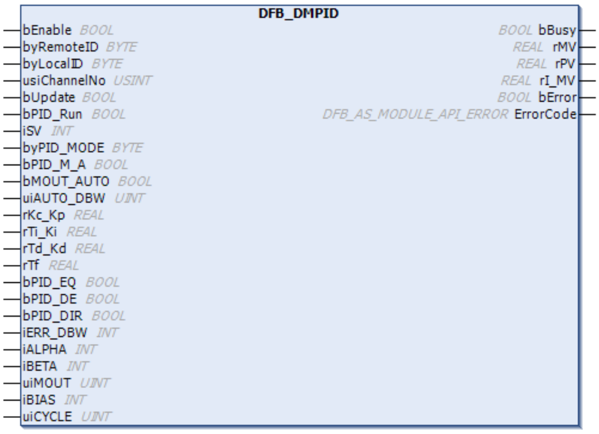
  - DL_ASModuleAPI_AX3.library

6.13 DFB_DMPID

- **Supported Products**
  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DMPID: RTD/TC module PID operation

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DMPID	 DFB_DMPID
		bEnable BOOL
		byRemoteID BYTE
		byLocalID BYTE
		usiChannelNo USINT
		bUpdate BOOL
		bPID_Run BOOL
		iSV INT
		byPID_MODE BYTE
		bPID_M_A BOOL
		bMOUT_AUTO BOOL
		uiAUTO_DBW UINT
		rKc_Kp REAL
		rTi_Ki REAL
		rTd_Kd REAL
		rTf REAL
		bPID_EQ BOOL
		bPID_DE BOOL
		bPID_DIR BOOL
		iERR_DBW INT
		iALPHA INT
		iBETA INT
		uiMOUT UINT
		iBIAS INT
		uiCYCLE UINT
		BOOL bBusy
		REAL rMV
		REAL rPV
		REAL rI_MV
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DMPID(
			bEnable:= ,
			byRemoteID:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			bUpdate:= ,
			bPID_Run:= ,
			iSV:= ,
			byPID_MODE:= ,
			bPID_M_A:= ,
			bMOUT_AUTO:= ,
			uiAUTO_DBW:= ,
			rKc_Kp:= ,
			rTi_Ki:= ,
			rTd_Kd:= ,
			rTf:= ,
			bPID_EQ:= ,
			bPID_DE:= ,
			bPID_DIR:= ,
			iERR_DBW:= ,
			iALPHA:= ,
			iBETA:= ,
			uiMOUT:= ,
			iBIAS:= ,
			uiCYCLE:= ,
			bBusy=> ,
			rMV=> ,
			rPV=> ,
			rI_MV=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byRemoteID[*1]	CPU or remote module number	BYTE	0: CPU 1~15: remote module(0)
byLocalID	Expansion module ID	BYTE	0 ~ 31(0)

Name	Function	Data Type	Setting Value (Default value)
usiChannelNo	Specified channel number	USINT	(0)
bPID_Run	Activate PID operation	BOOL	True/False(False)
iSV	Target value	INT	-32768~32767(0)
bPID_M_A	PID Auto/Manual mode	BOOL	TRUE: Manual, MV value will output according to the MOUT value. When PID_MODE is 1, this setting is invalid. FALSE: Auto, MV value will output according to the PID formula. (False)
bMOUT_AUTO	Manual value (MOUT) automatic update mode	BOOL	TRUE: Auto, MOUT value changes with the MV value. FALSE: Normal, MOUT value does not change with the MV value.
uiAUTO_DBW*2	When it applies to Auto tuning, the SV ± dead band range does not operate.	USINT	0~32000(0)
bPID_EQ	PID calculation formula selection	BOOL	True/False(False)
bPID_DE	PID differential error calculation selection	BOOL	True/False(False)
bPID_DIR	PID positive negative direction	BOOL	FALSE: heating operation (E=SV-PV) TRUE: cooling operation (E=PV-SV)
iERR_DBW	Deviation amount (E) ineffective range	INT	-32768~32767(0)
iBIAS	Feedforward control output value	INT	-32768~32767(0)

*Note:

1. Support only mode0 currently.
2. If the PID parameter write value exceeds the parameter upper limit, only the upper limit value will be written; if the PID parameter write value is lower than the parameter lower limit, only the lower limit value will be written.

- **Input Output Parameters**

Name	Function	Data Type	Setting/Output value range (Default Value)
bUpdate	Update PID parameter flag	BOOL	True/False(False) After Enable is activated, if users want to change parameters, please set parameters to new values, then set the bUpdate flag to TRUE. When the instruction change is done, the instruction will clear the bUpdate flag to FALSE.
byPIDMode	PID contol mode	BYTE	0: auto control function 1: auto adjust parameter function
rKc_Kp	scale coefficient (Kc or Kp, determin which coefficient to use accordingt o the bPID_EQ parameter)	REAL	Positive single-precision floating-point number range
rTi_Ki	Integral coefficient (Ti or Ki, determin which coefficient to use accordingt o the bPID_EQ	REAL	Positive single-precision floating-point number range

Name	Function	Data Type	Setting/Output value range (Default Value)
	parameter)		(Unit: Ti = sec; Ki = 1/sec)
rTd_Kd	Differential coefficient (Td or Kd, determin which coefficient to use accordingt o the bPID_EQ parameter)	REAL	Positive single-precision floating-point number range (Unit: sec)
rTf	Differential action time constant (Tf)	REAL	Positive single-precision floating-point number range (Unit: sec)
iALPHA	Initial integral compensation parameter (heating)	INT	0~100(0) (Unit: 1%)
iBETA	Initial integral compensation parameter (cooling)	INT	0~100(0) (Unit: 1%)
uiMOUT *	MV manual value	UINT	0~1000(0)

*Note: If the PID parameter write value exceeds the parameter upper limit, only the upper limit value will be written to the module; if the PID parameter write value is lower than the parameter lower limit, only the lower limit value will be written to the module.

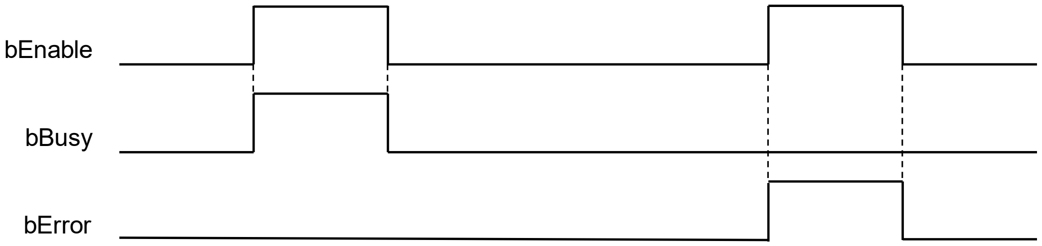
- **Output**

Name	Function	Data Type	Output Range(Default value)
bBusy	Instruction execution flag	BOOL	True/False(False)
lrMV	MV output value	LREAL	0.0~100.0(0) (Unit: 1%)
lrPV	Current value	LREAL	Positive/Negative number or 0(0)
lrI_MV	Accumulated integral value	LREAL	Positive/Negative number or 0(0)
bError	FB instruction error flag	BOOL	True/False(False)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to True.	● When bEnable shifts to False.
lrMV	● Continuously updating the record value when bBusy is True	● Clear to 0 when bEnable shifts to True.
lrPV	● Continuously updating the record value when bBusy is True	● Clear to 0 when bEnable shifts to True.
lrI_MV	● Continuously updating the record value when bBusy is True	● Continuously updating the record value when bValid is True
bError	● When FB instruction execution is incorrect or the input value is incorrect.	● When bEnable shifts to False.
ErrorCode		

- **Timing Diagram**

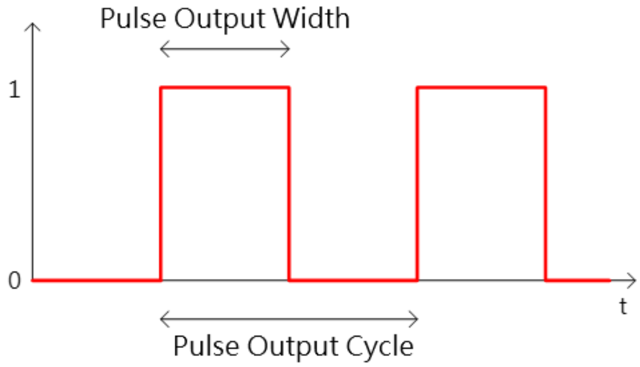


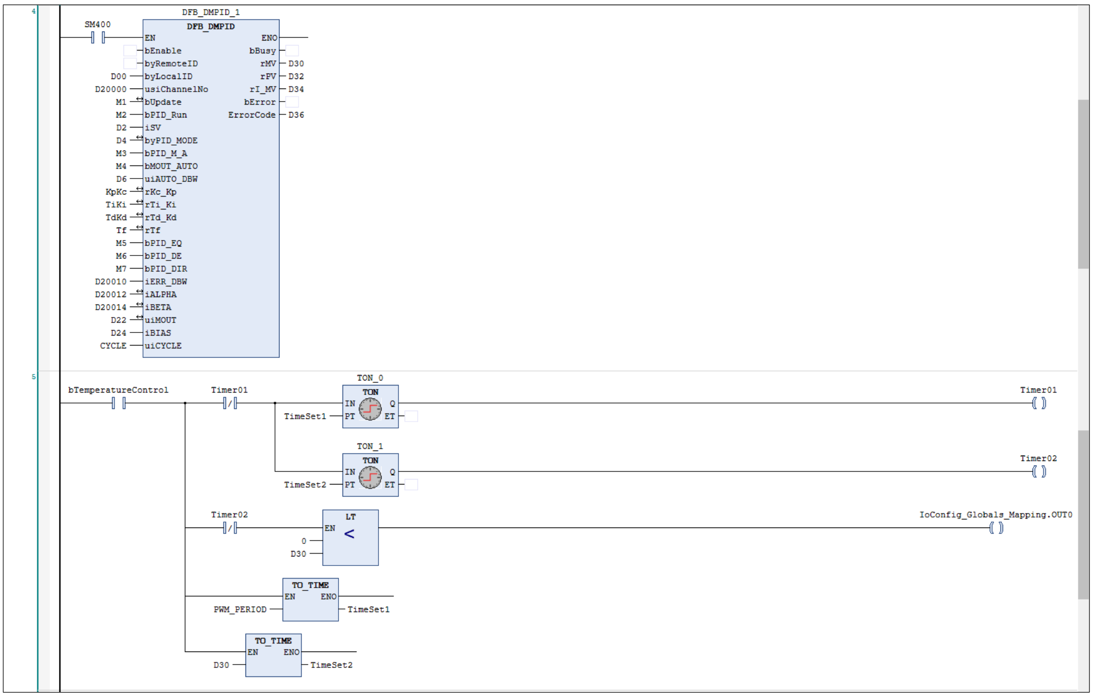
- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported only when AX-3 series firmware version is V1.0.1 and later.
3. The instruction is supported only by AS series temperature modules (supported version: AS04RTD-A V1.04 and later/AS06RTD-A V1.00 and later；AS04TC-A V1.04 and later /AS08TC-A V1.00 and later)
4. **byRemoteID** specifies that the analog input module should connect to the right of CPU or the right of the remote module group numbers. The CPU number is 0, the first remote module number is 1, and so on. The maximum group number is 15.
5. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
6. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.
7. When PID_RUN is TRUE → FALSE, the **rMV** output value will be cleared to 0. If users want the **rMV** output value to be kept as the last **rMV** output value when PID is not working, make bEnable TRUE → FALSE to deactivate the instruction.
8. Complete the parameter setting before executing the DMPID instruction. If this is the first time to write parameters, or users want to change parameters during execution (including **bPID_RUN** ~ **uiCYCLE**), set the **bUPDATE** flag. When the instruction completes the parameter change, the **bUPDATE** flag will be cleared.
9. If the auto adjust parameter function is used (**byPID_MODE** =1), it will automatically enter the autocontrol mode when the adjustment is complete (**byPID_MODE** changes to 0 automatically), and the debugged **rKc_Kp**, **rTi_Ki**, **rTd_Kd**, **rTf**, **iALPHA**, and **iBETA** parameters will be filled back to the corresponding parameter variables of the DMPID input. Users can also use the outage persistence variables to meet the requirement of saving PID control parameters.

- **Programming Example**

1. When **bEnable**=TRUE, the instruction is executed. When M2=TRUE, the DMPID instruction starts to operate. When M2=FALSE, MV value is 0 and is sent to D30. When **bEnable** become OFF, the instruction is not executed, and the parameter values in the original instruction does not change.
2. The **MV** output value range is 0.0~100.0. Users need to convert the required control quantity per requirement. This application example converts to PWM output. Convert the **MV** output value to the pulse output width TimeSet2 with a range of 0%~100%, and then multiply the PWM cycle to convert to the time TimeSet1.
3. Input the pulse output width and pulse output cycle to the TON function block, PWM control can be realized in the specified pulse output device.
   Duty cycle = pulse output width/pulse output cycle


Pulse Output Width
1
0
t
Pulse Output Cycle


4
DFB_DMPID_1
SM400
DFB_DMPID
EN
ENO
bEnable
bBusy
byRemoteID
rMV
D30
D00
byLocalID
rPV
D32
D20000
usiChannelNo
rI_MV
D34
M1
bUpdate
bError
M2
bPID_Run
ErrorCode
D36
D2
iSV
D4
byPID_MODE
M3
bPID_M_A
M4
bMOUT_AUTO
D6
uiAUTO_DBW
KpKc
rKc_Kp
TiKi
rTi_Ki
TdKd
rTd_Kd
Tf
rTf
M5
bPID_EQ
M6
bPID_DE
M7
bPID_DIR
D20010
iERR_DBW
D20012
iALPHA
D20014
iBETA
D22
uiMOUT
D24
iBIAS
CYCLE
uiCYCLE
5
bTemperatureControl
Timer01
TON_0
TON
IN
Q
TimeSet1
PT
ET
Timer01
TON_1
TON
IN
Q
TimeSet2
PT
ET
Timer02
Timer02
LT
EN
<
0
D30
IoConfig_Globals_Mapping.OUT0
TO_TIME
EN
ENO
PWM_PERIOD
TimeSet1
TO_TIME
EN
ENO
D30
TimeSet2

- **PID Calculation Formula:**

1. When the PID_MODE control mode is set to 0, it is the autocontrol mode. PID calculation formula is as follows:

- **Independent Formula & Derivative of E (PID_EQ**=False & **PID_DE**=False)

$$MV = K_P E + Ki\int_0^t Edt + K_d * \frac{dE}{dt} + BIAS$$ *E = SV – PV* or *E = PV – SV*

- **Independent Formula & Derivative of PV (PID_EQ**=False **& PID_DE**=True**)**

$$MV = K_P E + Ki\int_0^t Edt - K_d * \frac{dPV}{dt} + BIAS$$ *E = SV – PV*

Or

$$MV = K_P E + Ki\int_0^t Edt + K_d * \frac{dPV}{dt} + BIAS$$ *E =PV – SV*

- **Dependent Formula & Derivative of E (PID_EQ**=True & **PID_DE**=False)

$$MV = K_c\left[E + \frac{1}{T_i}\int_0^t Edt + T_d * \frac{dE}{dt}\right] + BIAS$$ *E = SV – PV* or *E = PV – SV*

- **Dependent Formula & Derivative of PV (PID_EQ**=True & **PID_DE**=True)

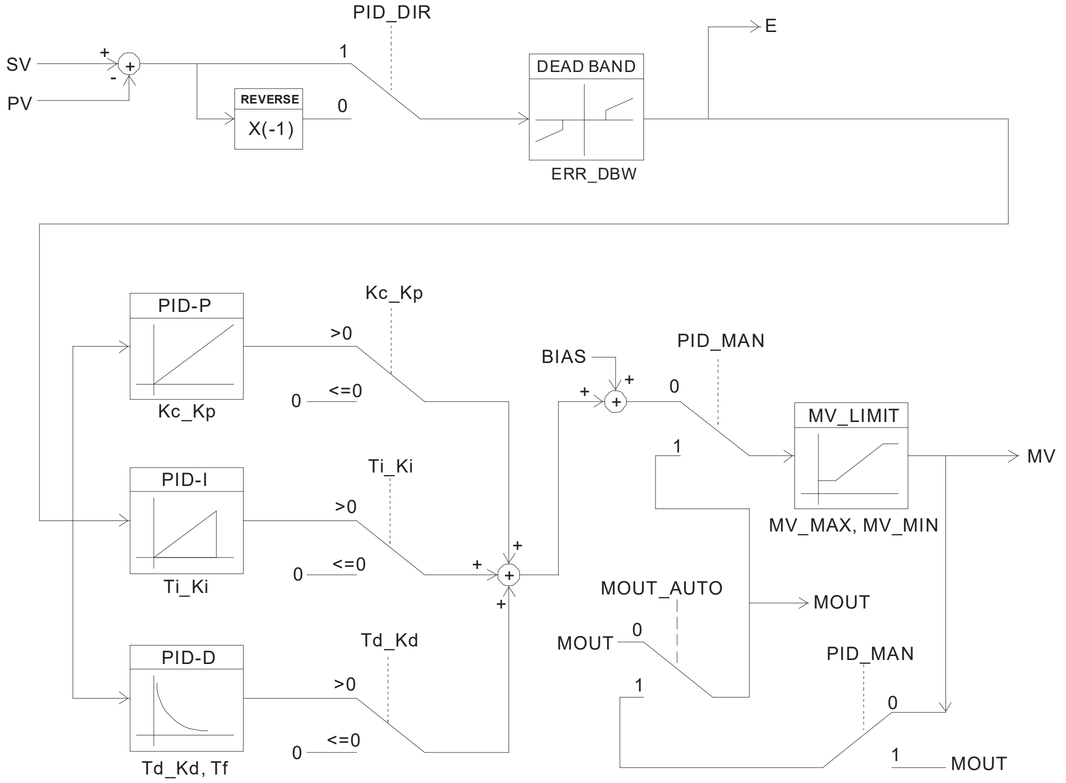
$$MV = K_c\left[E + \frac{1}{T_i}\int_0^t Edt - T_d * \frac{dE}{dt}\right] + BIAS$$ *E = SV – PV*

Or

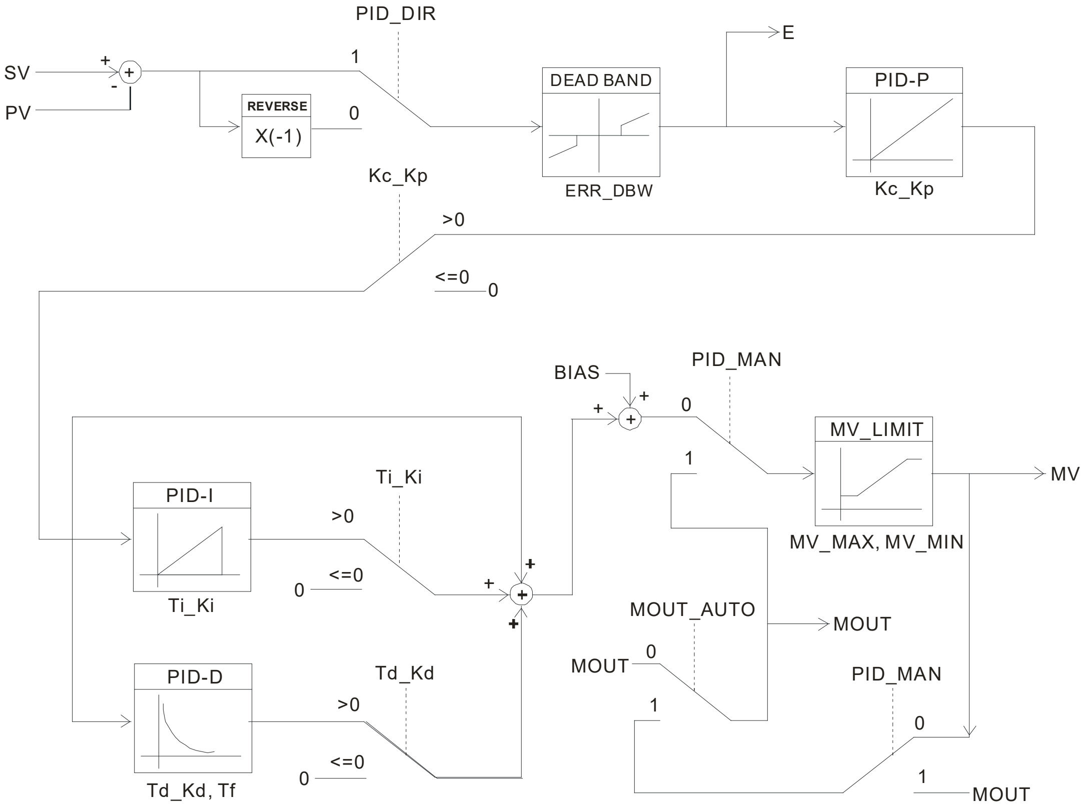
$$MV = K_c\left[E + \frac{1}{T_i}\int_0^t Edt + T_d * \frac{dE}{dt}\right] + BIAS$$ *E = PV – SV*

- **PID Block Diagram:**

PID Block Diagram (Independent)



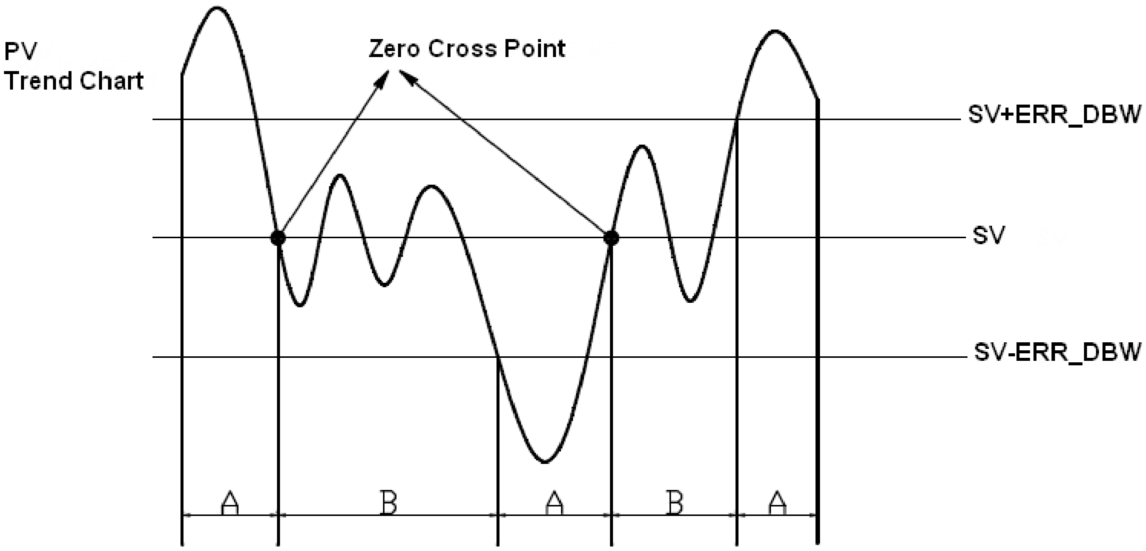
PID Block Diagram (Dependent)



- **Deviation Ineffective Range**

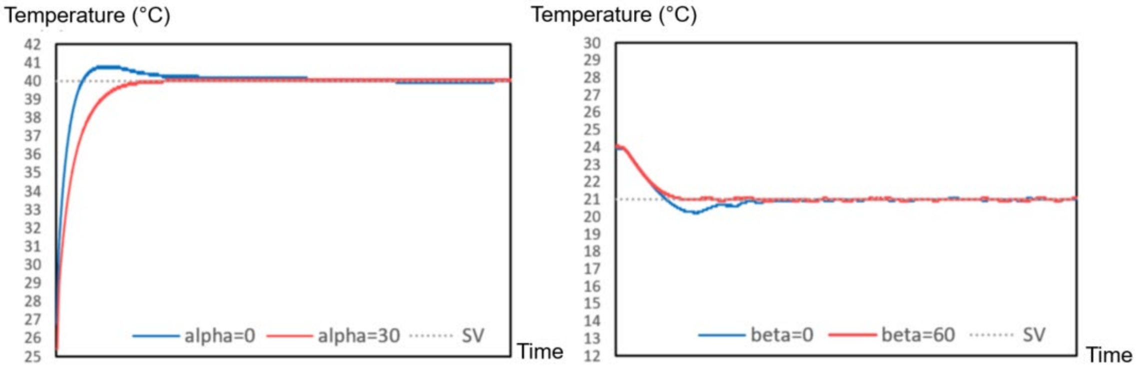
When the PV value enters the ERR_DBW range, at first the CPU still performs the PID calculation according to the E value until PV crosses the SV value (Zero Cross Point), which means that Cross Status is established, and E = 0 will be substituted into the PID calculation. Then, when the PV value exceeds the ERR_DBW range, the E value will be resumed for the PID calculation. If PID_DE=True, it means that using the PV value to perform the differential calculation. After the Cross Status conditions are established, the CPU will take Delta PV as 0 to perform the PID differential calculation. (Delta PV= Current PV-previous PV).

For example, in the following PV trend diagram, CPU in A area will perform the normal PID calculation; however, CPU in B area will take E or Delta PV as 0 to perform the PID calculation.



- **α, β VALUE:**

  ALPHA and BETA are used to compensate initial integral when PID starts and the SV target values changes, the aim of which is to reduce the overshoot phenomenon. As shown in the following figures, ALPHA parameter is used to slow down the rising overshoot; BETA parameter is used to slow down the falling overshoot.



- **Notes and recommendations:**

1. When users adjust three main parameters, Kc_Kp, Ti_Ki, and Td_Kd (**PID_MODE**=0), please adjust the Kc_Kp value at first (based on past experience), and then set Ti_Ki and Td_Kd values to 0. When the adjustment is generally controllable, then adjust the Ti_Ki value (small → large) and the Td_Kd value (large → small) in sequence. When Kc_Kp = 1, it represents 100%, that is, the gain of the deviation value is 1. A value less than 100% will attenuate the deviation value; a value more than 100% will increase the deviation value.
2. The automatically adjusted parameters are not guaranteed to be suitable for each control environment; therefore, users can certainly modify the adjusted parameters. It is recommended to modify only the Ti_Ki or Td_Kd value.
3. The CYCLE parameter is how much time for the PID function to calculate once and update the output value (MV).
4. Please note that when the number of channels which is open for measurement changes, the update time of the measurement value will change (for example, when opening only one channel for measurement, the measurement value will be updated every 200ms. When opening the other three channels for measurement, the measurement value will be updated every 800ms). The Kc_Kp, Ti_Ki, Td_Kd, etc. parameters may not be applicable.

- **Library**
  - DL_ASModuleAPI_AX3.library

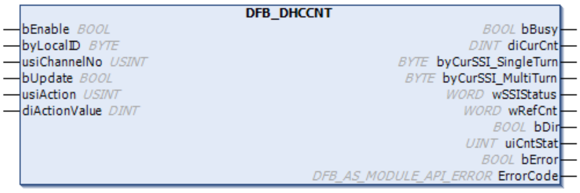
6.14 DFB_DHCCNT

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DHCCNT: This instruction is for AS02HC-A instruction only. Its functions include starting/shutting counter and setting/modifying counter value.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DHCCNT		DFB_DHCCNT(
			bEnable:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			bUpdate:= ,
			usiAction:= ,
			diActionValue:= ,
			bBusy=> ,
			diCurCnt=> ,
			byCurSSI_SingleTurn=> ,
			byCurSSI_MultiTurn=> ,
			wSSIStatus=> ,
			wRefCnt=> ,
			bDir=> ,
			uiCntStat=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expand the module number	BYTE	0 ~ 31(0)
usiChannelNo	Specify channel number	USINT	1 ~ 2(1)
usiAction*	Counter Action mode setting	USINT	0 ~ 7(0)
diActionValue	New counter value/offset	DINT	-32768~32767(0)

* Note: usiAction is the action mode setting of the counter, and diActionValue usage descriptions are as below:

usiAction	Function	Description
0	Keep the current mode and not to change.	-
1	Set/Change the current counter value diCurCnt to diActionValue	When setting the new counter value, please set diActionValue value. Note: When using SSI encoder and the counter form is set as Absolute Position,

usiAction	Function	Description
		this usiAction is invalid.
2	Set SSI encoder offset as diActionValue	When the counter form is set as Absolute Position, users can set the offset of the SSI encoder counter value. Set the offset through diActionValue, and its counter value diCurCnt = encoder original counter value + diActionValue. Note: 1. diActionValue range: $-2^{(MT+ST^{length})}$ < diActionValue < $2^{(MT+ST^{length})}$ Setting is not allowed when it is out of range. 2. diActionValuerange: $-2^{(MT+ST^{length})}$ < diActionValue < $2^{(MT+ST^{length})}$. Setting is not allowed when it is out of range. 3. When using SSI encoder and the counter form is set as Ring counter, this diActionValue is invalid. When the device area HC module configuration is re-downloaded, the offset will be cleared to 0.
3	Set/Change SSI encoder absolute position value to diActionValue	When the counter form is set as Absolute Position, the SSI encoder counter value will be automatically shifted to the diActionValue vale, and its counter value diCurCnt = diActionValue. Note: 1. diActionValue range: 0< diActionValue < $2^{(MT+ST^{length})}$. Setting is not allowed when it is out of range. 2. When using SSI encoder and the counter form is set as Ring counter, this usiAction is invalid. Please use usiAction=1 to change the counter value. 3. When the device configuration is re-downloaded, the offset will be cleared to 0.
4	Reset the current counter value diCurCnt	1. Reset diCurCnt as 0. 2. Reset iCurrentNo of table compare output instruction DFB_DHCCMPT. 3. Reset the bMatch1 and bMatch2 flags of compare output instruction DFB_DHCCMP. Note: When using SSI encoder, and the counter form is set as Absolute Position, the counter value cannot be reset. However, the iCurrentNo, bMatch1, and bMatch2 flags will be reset.
5	Reset diCurCnt current counter value diCurCnt, and reset table compare output instructions/specified Y output point of compare output instructions.	In addition to the above usiAction=4 reset content, compare output instructions/specified Y output point of compare output instructions (ON → OFF) are also reset.
6	Preset the current counter value diCurCnt	1. diCurCnt is specified to be modified to diActionValue. 2. Reset iCurrentNo of table compare output instruction DFB_DHCCMPT. Please note that after Preset, DFB_DHCCMPT will wait the comparison to arrive from the first compare value; therefore, if the counter value is bigger than the first compare value after Preset, the table comparison cannot be done correctly. Please execute Preset, then set the bUpdate flag of DFB_DHCCMPT to ON. 3. Reset the bMatch1/2 flag of compare output instruction DFB_DHCCMP.

usiAction	Function	Description
		Note: When using SSI encoder, and the counter form is set as Absolute Position, the counter value cannot be preset, but the iCurrentNo, bMatch1, and bMatch2 flags will be reset.
7	Preset the current counter value diCurCnt, and reset table compare output instruction/ specified Y output point of compare output instructions.	In addition to the above usiAction=6 reset content, table compare output instructions/specified Y output point of compare output instructions (ON → OFF) are also reset.

- **Input/Output Pin**

Name	Function	Data Type	Setting/Output value range (Default Value)
bUpdate	Update DFB_DHCCNT parameter flag	BOOL	True/False(False)

- **Output**

Name	Function	Data Type	Setting/Output value range (Default Value)
bBusy	Instruction execution flag	BOOL	True/False(False)
diCurCnt	Specify the current counter value of module count channels	DINT	Positive number, negative number or 0(0)
byCurSSI_SingleTurn	Absolute SSI encoder Single-Turn Data	BYTE	Positive number or 0(0)
byCurSSI_MultiTurn	Absolute SSI encoder Multi-Turn Data	BYTE	Positive number or 0(0)
wSSIstatus	Absolute SSI encoder status information	WORD	Positive number or 0(0)
wRefCnt	Absolute SSI encoder data refresh counter	WORD	Positive number or 0(0)
bDir	Counting direction display	BOOL	True: Positive direction False: Negative direction (False)
uiCntStat*	Counter status	UINT	Positive number or 0(0)
bError	Instruction error flag	BOOL	True/False(False)
ErrorCode	Error code	DFB_AS_MODULE_ API_ERROR	DFB_AS_MODULE_API_ ERROR(DFB_NO_ERROR)

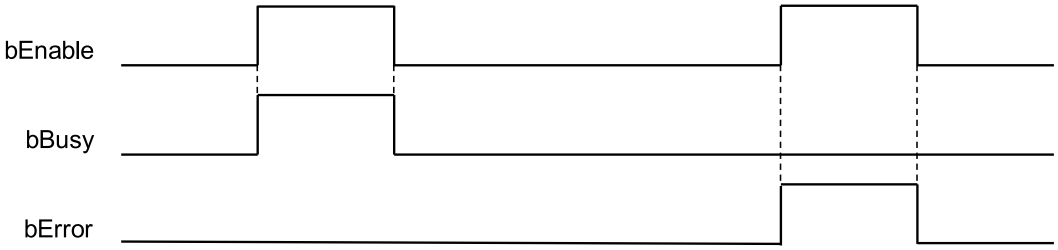
* Note: uiCntStat is counting module action status, and its usage descriptions are as below:

Bit Number	Status Name	Status Description	Note
15 ~ 11	Reserved	Reserved	
10	SSI absolute position across zero point	0: Normal 1: Abnormal	The device configuration area starts or shuts detection, and the default is no detection. When the counter type is Absolute Position: When users have set the offset, the over-zero detection will be performed with the position value after the offset is adjusted. When the counter type is Ring Counter: Because the offset will be reset to 0, the over-zero detection will be performed with the encoder original position value. Cause of error: over-zero occurs. Clear method: DFB_DHCCNT performs resetting or presetting instructions, or the external input point performs resetting.
9	SSI communication anomaly	0: Normal 1: Abnormal	Cause of error: communication anomaly 5 times in a row. Clear method: communication is restored to normal.
8	SSI parity check error	0: Normal 1: Abnormal	Cause of error: parity check error Clear method: next reading value parity check is correct.
7	SSI data change size exceeds protection setting	0: Normal 1: Abnormal	Cause of error: The change size of two position data exceeds protection setting. Clear method: Next data change size is within a reasonable range.
6	Reserved	Reserved	
5	Ring counting Overflow	0: Normal 1: Abnormal	The device configuration area starts or shuts the detection, and the default is not to detect the cause of error: when hardware counter Overflow (> 2^{*32}− 1) or Underflow (< -2^{*32}− 1) occurs, there will be no more counting. The counting will be restored after executing the following: The DFB_DHCCNT instruction performs resetting or presetting instructions, or the external input point performs resetting.
4	Ring counting Underflow	0: Normal 1: Abnormal	
3	Linear count is higher than the set limit	0: Normal 1: Abnormal	Cause of error: Linear count exceeds the user set range. Clear method: Linear counter value is back to the upper and lower limit. Note: When count exceeds the upper/lower limit, the counter value remains at the upper/lower limit. When counter value is back to the upper/lower limit range, the counting resumes.
2	Linear count is lower than the set limit	0: Normal 1: Abnormal	
1	Linear count Overflow	0: Normal 1: Abnormal	Cause of error: When linear count exceeds upper/lower limit, hardware counter continues counting. When hardware counter Overflow(> 2^{*32}− 1) or Underflow (< -2^{*32}− 1) occurs, there will be no more counting. The counting will be restored after executing the following: The DFB_DHCCNT instruction performs resetting or presetting instructions, or the external input point performs resetting.
0	Linear count Underflow	0: Normal 1: Abnormal	

- **Outputs Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bBusy	● When bEnable shifts to true and the function is enabled in the module.	● When bEnable shifts to false and the function is disabled in the module.
diCurCnt	● Continuously update the record when bBusy is True.	-
byCurSSI_SingleTurn	● Continuously update the record when bBusy is True, and the counter form is Absolute Position.	-
byCurSSI_MultiTurn	● Continuously update the record when bBusy is True, and the counter form is Absolute Position.	-
wSSIstatus	● Continuously update the record when bBusy is True, and the counter form is Absolute Position.	-
wRefCnt	● Continuously update the record when bBusy is True, and the counter form is Absolute Position.	-
bDir	● Continuously update the record when bBusy is True. ● True when the count direction is positive.	● False when the count direction is negative.
uiCntStat	● Continuously update the record when bBusy is True.	● Clear the error when using usiAction clearance function. ● When error count module status is cleared.
bError	● When FB instruction execution is error or input value is error.	● When bEnable shifts to false.
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported when AX-3 Series firmware is V1.0.2 and above.
3. This instruction is only supported by AS Series count module (The supported version is AS02HC-A V1.00 and above).
4. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
5. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.

6. Complete the **usiAction** and **diActionValue** settings before executing this instruction. When **bEnable** is started for the first time, **usiAction** and **usiAction** of HC module will be set once. When **usiAction** and **diActionValue** need to be re-changed during execution, set **usiAction** and **diActionValue** as new values, and then set the **bUpdate** flag to On. When this instruction completes changes, the instruction will clear **bUpdate** as Off itself.

7. **diCurCnt** is to display the current counter value of the specified module count channel.

8. **byCurSSI_SingleTurn** is absolute SSI encoder Single-Turn Data display (When the counter form is set as Ring Position, **byCurSSI_SingleTurn** shows the value after deviation; when setting the counter as Ring counter, **byCurSSI_SingleTurn** will show the original SSI encoder value), when the channel mode is set as incremental encoder, the **byCurSSI_SingleTurn** will be 0.

9. **byCurSSI_MultiTurn** is absolute SSI encoder Multi-Turn Data display (When the counter form is set as Absolute Position, **byCurSSI_MultiTurn** shows the value after deviation; when setting the counter as Ring counter, **byCurSSI_MultiTurn** will show the original SSI encoder value), when the channel mode is set as incremental encoder, the **byCurSSI_MultiTurn** will be 0.

10. **WSSIstatus** is absolute SSI encoder Status Data display. When the channel mode is set as incremental encoder, **wSSIstatus** will be 0.

11. **wRefCnt** is absolute SSI encoder data refresh counter, and its length is 16 bits. When new SSI data is captured, refresh counter will be incremented by one, and when this counter overflows, it will re-accumulate from 0. When the channel mode is set as incremental encoder, **wRefCnt** will be 0.

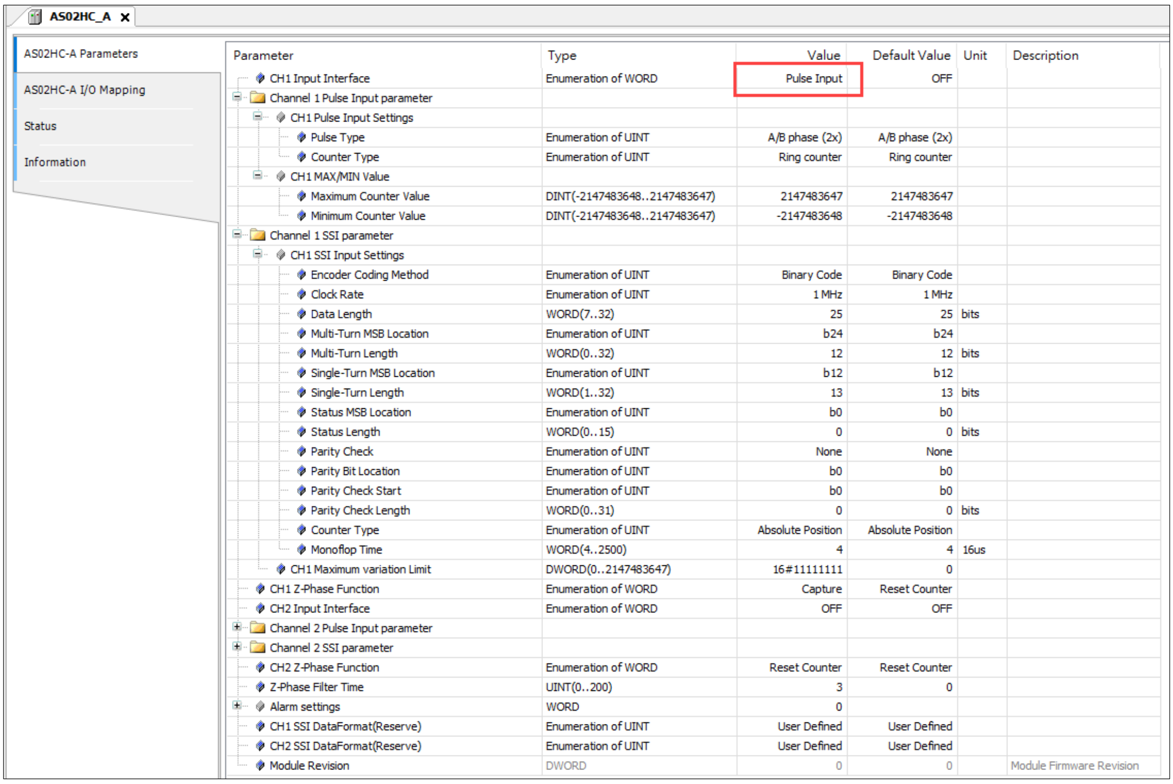
12. **bDir** is the count direction display. When it shows On, it represents going in positive direction; when it shows Off, it represents going in negative direction.

13. If the instruction is closed, the specify channel will stop updating the values of the right half of the instruction.

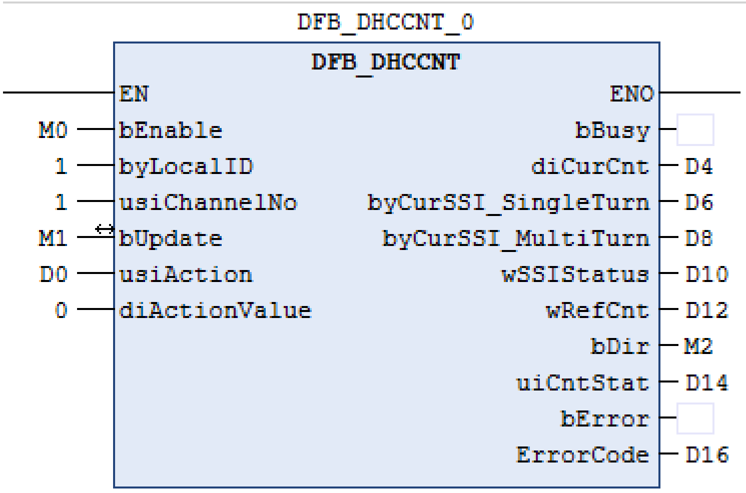
- **Programming Example**

▪ Programming Example :

This example uses the FB instruction (DFB_DHCCNT) to read the first channel in the right module (AS02HC) of the host, and after setting the count parameter to the module, start updating counter value to a new variable (D4) through the function block.



Parameter	Type	Value	Default Value	Unit	Description
CH1 Input Interface	Enumeration of WORD	Pulse Input	OFF		
Channel 1 Pulse Input parameter					
CH1 Pulse Input Settings					
Pulse Type	Enumeration of UINT	A/B phase (2x)	A/B phase (2x)		
Counter Type	Enumeration of UINT	Ring counter	Ring counter		
CH1 MAX/MIN Value					
Maximum Counter Value	DINT(-2147483648..2147483647)	2147483647	2147483647		
Minimum Counter Value	DINT(-2147483648..2147483647)	-2147483648	-2147483648		
Channel 1 SSI parameter					
CH1 SSI Input Settings					
Encoder Coding Method	Enumeration of UINT	Binary Code	Binary Code		
Clock Rate	Enumeration of UINT	1 MHz	1 MHz		
Data Length	WORD(7..32)	25	25	bits	
Multi-Turn MSB Location	Enumeration of UINT	b24	b24		
Multi-Turn Length	WORD(0..32)	12	12	bits	
Single-Turn MSB Location	Enumeration of UINT	b12	b12		
Single-Turn Length	WORD(1..32)	13	13	bits	
Status MSB Location	Enumeration of UINT	b0	b0		
Status Length	WORD(0..15)	0	0	bits	
Parity Check	Enumeration of UINT	None	None		
Parity Bit Location	Enumeration of UINT	b0	b0		
Parity Check Start	Enumeration of UINT	b0	b0		
Parity Check Length	WORD(0..31)	0	0	bits	
Counter Type	Enumeration of UINT	Absolute Position	Absolute Position		
Monoflop Time	WORD(4..2500)	4	4	16us	
CH1 Maximum variation Limit	DWORD(0..2147483647)	16#11111111	0		
CH1 Z-Phase Function	Enumeration of WORD	Capture	Reset Counter		
CH2 Input Interface	Enumeration of WORD	OFF	OFF		
Channel 2 Pulse Input parameter					
Channel 2 SSI parameter					
CH2 Z-Phase Function	Enumeration of WORD	Reset Counter	Reset Counter		
Z-Phase Filter Time	UINT(0..200)	3	0		
Alarm settings	WORD	0			
CH1 SSI DataFormat(Reserve)	Enumeration of UINT	User Defined	User Defined		
CH2 SSI DataFormat(Reserve)	Enumeration of UINT	User Defined	User Defined		
Module Revision	DWORD	0	0		Module Firmware Revision

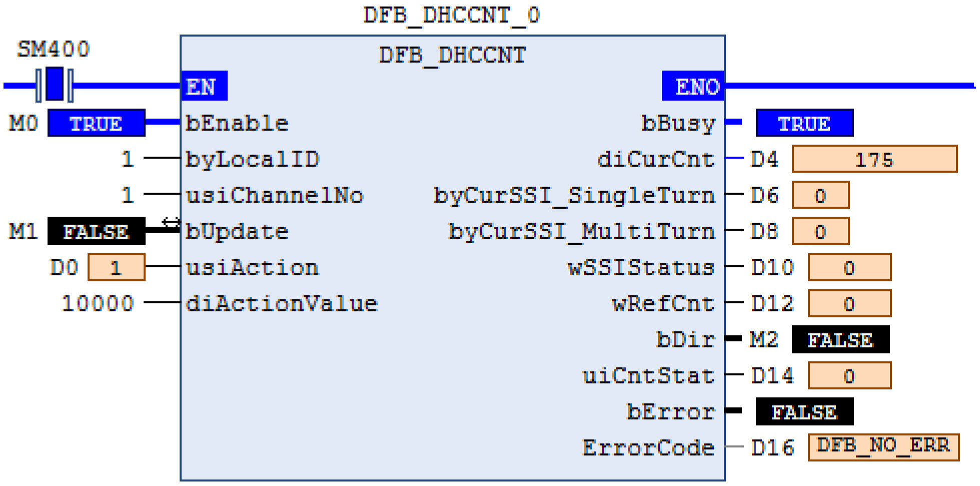


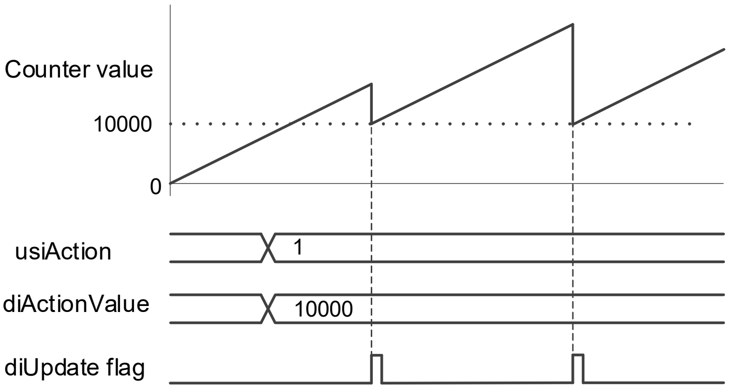
◆ Example 1: Set/Change the current counter value. (usiAction = 1)

1. When setting M0 = ON, Counter starts to count.

2. When setting usiAction parameter of the DHCCNT instruction as 1 and setting diActionValue parameter as 10000; when users set the bUpdate flag M1=ON, the current counter value CurNo will be changed to 10000.

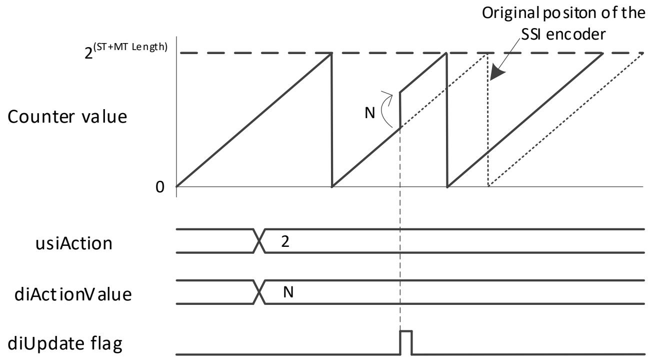
3. When setting bUpdate flag is complete, M1 will be cleared as OFF.





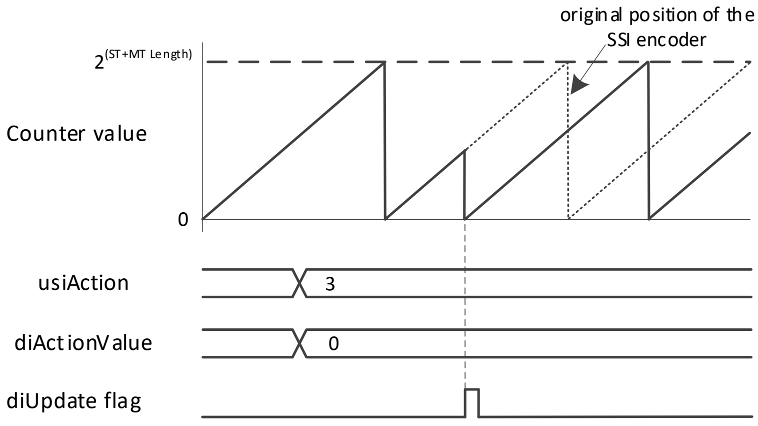
◆ Example 2 – set absolute SSI encoder offset (usiAction = 2)

1. 1Module device parameter sets the channel 1 as SSI Input, and select the counter form as Absolute Position.
2. When setting M0=ON, counter starts to count.
3. Set the usiAction parameter of the DHCCNT instruction as 2, and set diActionValue parameter as 500. Suppose the current counter value diCurCnt is 2500, and users set the bUpdate flag M1=ON, now the counter value will be changed to 3000.
4. After setting the bUpdate flag is complete, M1 will be clear to OFF automatically.



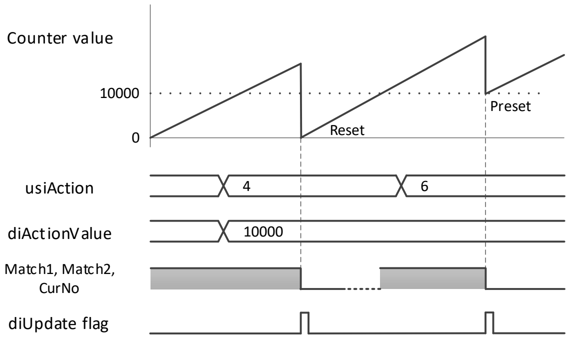
◆ Example 3: Set/Change SSI encoder absolute position value (Action = 3)

1. Module device parameter sets channel 1 as SSI Input, and select the counter form as Absolute Position.
2. When setting M0=ON, counter starts to count.
3. Set the usiAction parameter of the DHCCNT instruction to 3, and set the diActionValue parameter to 0. When users set the bUpdate flag M1=ON, the HC module will calculate the deviation amount automatically, and shift current counter value diCurCnt (encoder absolute position) to 0.
4. When setting the bUpdate flag is complete, M1 will be cleared to OFF.



- Example 4: Reset or preset the current counter value (usiAction = 4、6)

1. When setting M0=ON, counter starts to count.
2. Set the usiAction parameter of the DHCCNT instruction as 4. When users set the bUpdate flag M1=ON, the current counter value CurNo will be reset to 0. When setting the update flag is complete, M1 will be reset as OFF.
3. Set the usiAction parameter of the DHCCNT instruction to 6, and set the diActionValue parameter to 10000. When users set the bUpdate flag to M1=ON, the current counter value diCurCnt will be preset as 10000. When setting the bUpdate flag is complete, M1 will be reset as OFF.



- **Library**

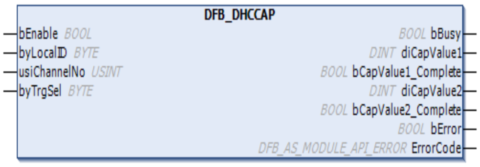
  - DL_ASModuleAPI_AX3.library

6.15 DFB_DHCCAP

- **Supported Products**
  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DHCCAP：This instruction is for HC counting only. Its functions include starting/shutting catcher.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DHCCAP	 DFB_DHCCAP
		bEnable BOOL
		byLocalID BYTE
		usiChannelNo USINT
		byTrgSel BYTE
		BOOL bBusy
		DINT diCapValue1
		BOOL bCapValue1_Complete
		DINT diCapValue2
		BOOL bCapValue2_Complete
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DHCCAP(
			bEnable:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			byTrgSel:= ,
			bBusy=> ,
			diCapValue1=> ,
			bCapValue1_Complete=> ,
			diCapValue2 => ,
			bCapValue2_Complete=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute the function block	BOOL	True/False(False)
byLocalID	Expand the module number	BYTE	0 ~ 31(0)
usiChannelNo	Specify the channel number	USINT	1 ~ 2(1)
byTrgSel*	TRIGGER method selection	BYTE	0 ~ 1(0)

＊Note: byTrgSel is the trigger method selection captured by counter value. Its usage is as follows:

byTrgSel	Function	Note
0	Triggered by the digital input of the specified channel.	Capture function is selected for the external input point function with the HC device configuration.
1	Triggered by bMatch1 and bMatch2 of another channel compare output instruction.	Example: When channel 2 starts DFB_DHCCAP and sets **byTrgSel**=1, and when instruction channel 1 is used for compare output instruction DFB_DHCCMP, when channel 1 compare arrives bMatch1 and bMatch2, the current counter values captured by channel 2 will be triggered and saved in **diCapValue1** or **diCapValue2.** Note: When **byTrgSel**=1, even if the external input point function of the HC device configuration is selected as Capture, the external input point capture function will be considered invalid.

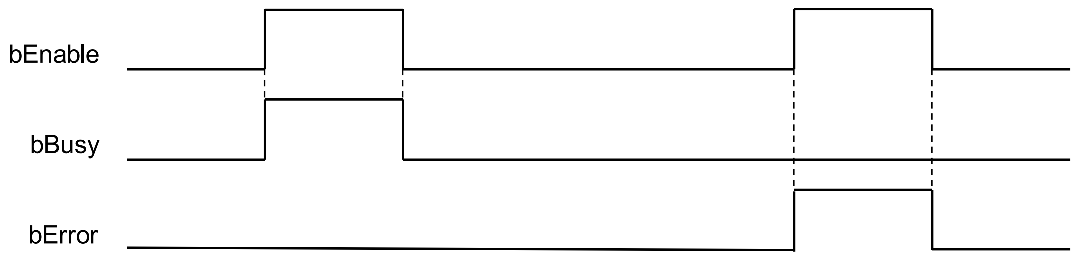
- **Output**

Name	Function	Data Type	Output Value Range (Default)
bBusy	Instruction execution flag	BOOL	True/False(False)
diCapValue1	Capture counter value1	DINT	Positive number, negative number or 0(0)
bCapValue1_Complete	diCapValue1 capture completion flag	BOOL	True/False(False)
diCapValue2	Capture counter value2	DINT	Positive number, negative number or 0(0)
bCapValue2_Complete	diCapValue2 capture completion flag	BOOL	True/False(False)
bError	Instruction error flag	BOOL	True/False(False)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to true and the function is enabled in the module.	● When bEnable shifts to false and the function is disabled in the module.
diCapValue1	● When bBusy is true and update the record when the capture signal is established.	-
bCapValue1_Complete	● When bBusy is true and after the capture value is updated to diCapValue1.	● When bEnable shifts to false. ● Cleared by users
diCapValue2	● When bBusy is True and updated the record when the capture signal is established.	-
bCapValue2_Complete	● When bBusy is true and after the capture value is updated to diCapValue2.	● When bEnable shifts to false. ● Cleared by users
bError	● FB instruction execution error or input value error	● When bEnable shifts to false.
ErrorCode		

- **Timing Diagram**

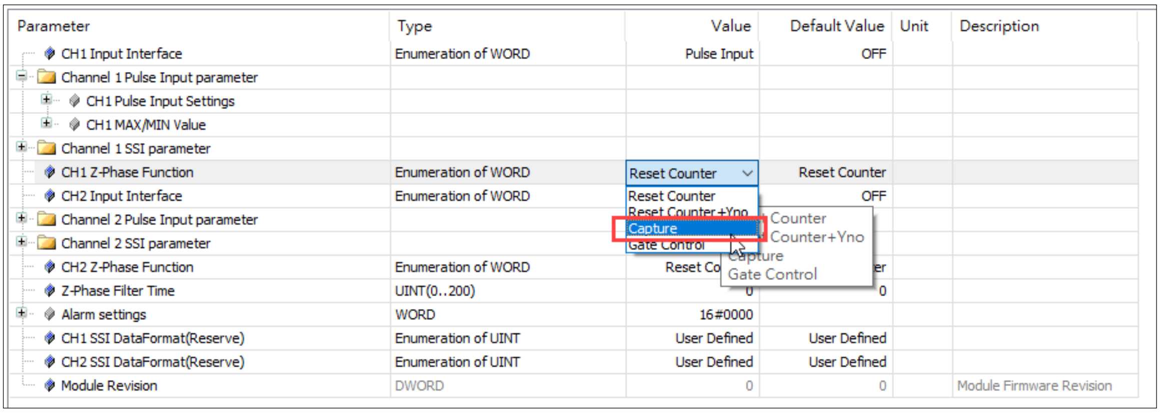


- **Function**

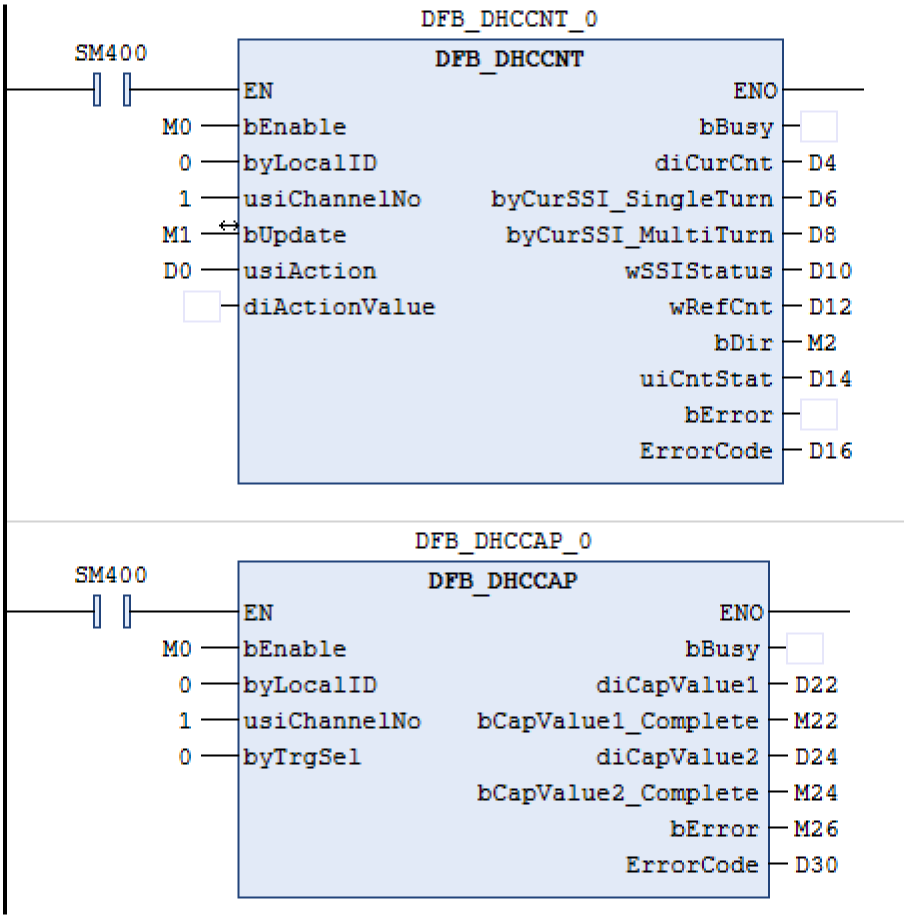
1. It is suggested that this instruction be placed under Main Task.
2. This function is supported when AX-3 Series firmware is V1.0.2 and above.
3. This instruction is only supported by AS Series count module (The supported version is AS02HC-A V1.00 and above).
4. DFB_DHCCAP needs to be used with the DFN_DHCCNT instruction. Only when DFB_DHCCNT is enabled, the counter value will count according to the input signals, and the counter value captured by DFB_DHCCAP is valid. When DFB_DHCCNT is off, the counter value will stop receiving input signals and stop updating counter vale. At this time, the captured counter value will not change.
5. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
6. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.
7. Complete **byTrgSel** setting before executing this instruction. When the **En** instruction starts for the first time, **byTrgSel** will be set once for the HC module. During execution, if **byTrgSel** needs to be re-changed, please disable the instruction and start again.
8. **diCapValue1** and **diCapValue2** are capture counter value 1 and capture counter value 2 respectively. When **byTrgSel**=0, **diCapValue1** is the counter value stored in external input point rising edge, and **diCapValue2** is the counter value stored in external input point falling edge. When **byTrgSel**=1, **diCapValue1** is the counter value that stored in another channel compare input instruction bMatch1 from Off→On, and **diCapValue2** is the counter value that stored in another channel compare input instruction bMatch2 from Off→On.
9. **bCapValue1_Complete** and **bCapValue2_Complete** are the flags of capture completion counter value1 and capture completion counter value2. When **bCapValue1_Complete/bCapValue2_Complete** is Off-**>**On, it means that **diCapValue1/ diCapValue2** are already the latest capture values, and users need to clear the **bCapValue1_Complete/ bCapValue2_Complete** flags after reading capture values. When next **bCapValue1_Complete** and **bCapValue2_Complete** are Off-**>**On, there are new capture values. If users do not clear the **bCapValue1_Complete** and **bCapValue2_Complete** flags, the module latest capture values will keep updating until **diCapValue1** and **diCapValue2.**
10. If **bEnable** is from On to Off, it means that disabling the instruction Capture function. At this time, **diCapValue2** and **diCapValue2** content values will remain the same and will not be updated. However, **bCapValue1_Complete** and **bCapValue2_Complete** flags will be cleared.

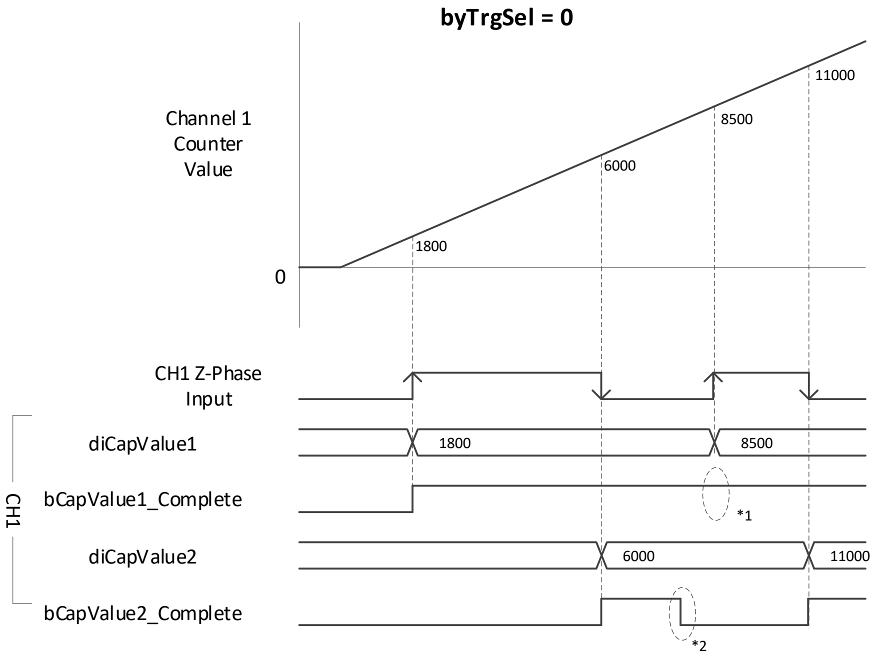
- Programming Example：

This example uses the FB instruction (DFB_DHCCNT) to start the first channel counting function in the right module (AS02HC) of the host, and use the FB instruction (DFB_DHCCAP) to capture the counter value.



Parameter	Type	Value	Default Value	Unit	Description
CH1 Input Interface	Enumeration of WORD	Pulse Input	OFF		
Channel 1 Pulse Input parameter					
CH1 Pulse Input Settings					
CH1 MAX/MIN Value					
Channel 1 SSI parameter					
CH1 Z-Phase Function	Enumeration of WORD	Reset Counter	Reset Counter		
CH2 Input Interface	Enumeration of WORD		OFF		
Channel 2 Pulse Input parameter					
Channel 2 SSI parameter					
CH2 Z-Phase Function	Enumeration of WORD	Reset Co	er		
Z-Phase Filter Time	UINT(0..200)	0	0		
Alarm settings	WORD	16#0000			
CH1 SSI DataFormat(Reserve)	Enumeration of UINT	User Defined	User Defined		
CH2 SSI DataFormat(Reserve)	Enumeration of UINT	User Defined	User Defined		
Module Revision	DWORD	0	0		Module Firmware Revision

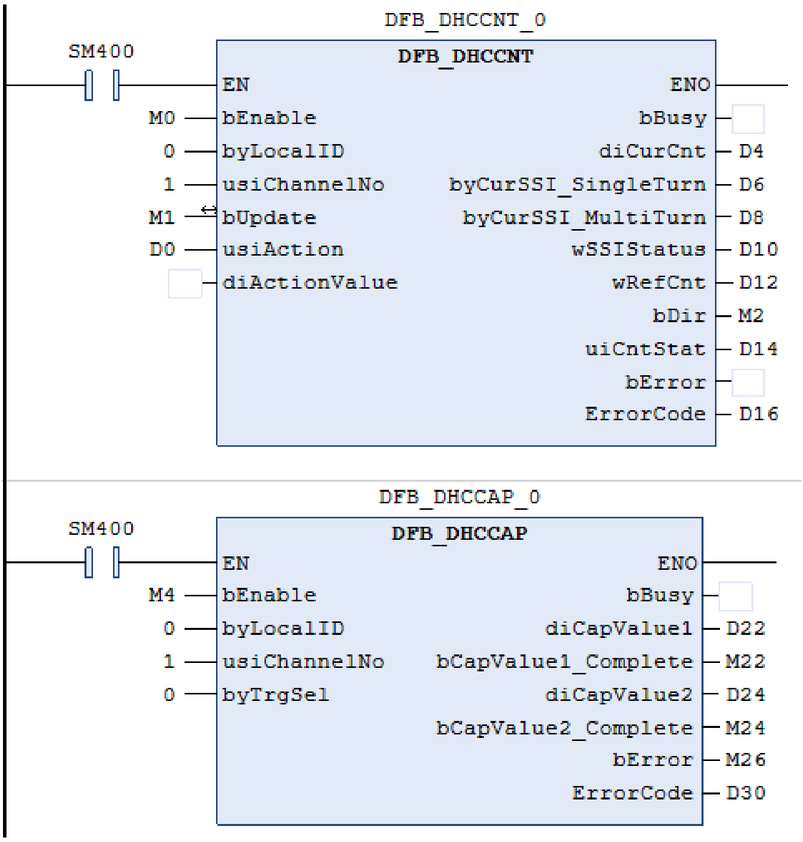


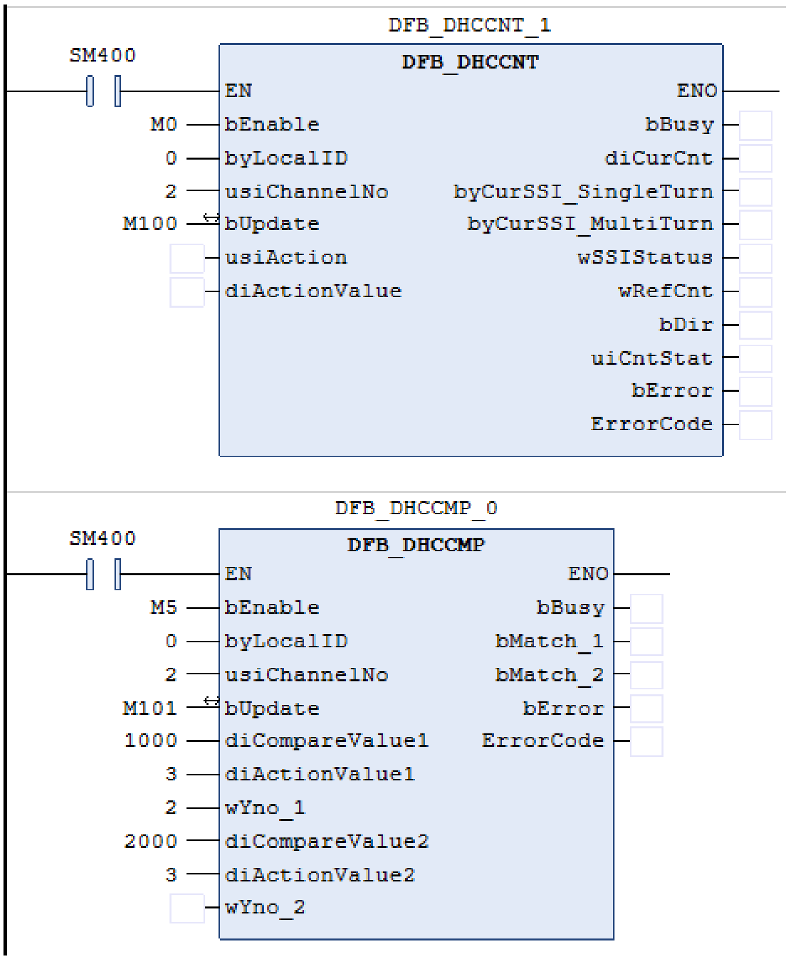


* 1: bCapValue1_Complete is not cleared to OFF, the captured value still updates in di CapValue 1.
*2: bCapValue2_Complete is cleared to OFF, the captured value still updates in CapValue 2.

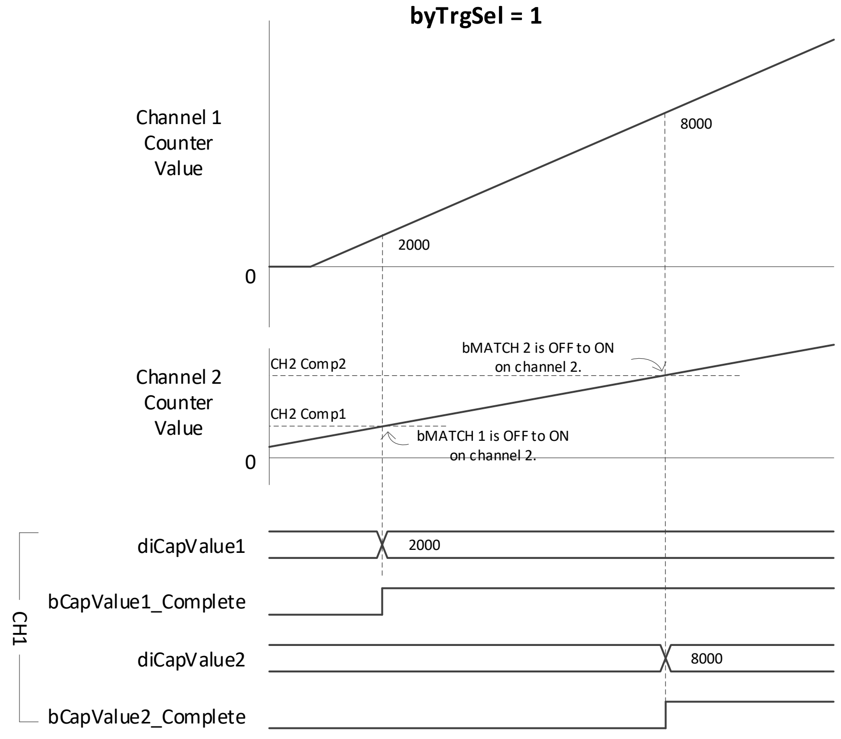
- Programming example: Another channel DHCCMP compares arrival and captures the counter value.

1. Set the **byTrgSel** parameter of the DHCCAP instruction in channel 1. When M0=ON, the DHCCAP instruction will starts to wait for compare arrival of the other channel (channel 2).
2. When channel 2 counter value arrives compare value diCompareValue1, and at this time channel 1 counter value 2000 will immediately output to **diCapValue1**, and the **bCapValue1_Complete** flag will be set to ON.
3. When channel 2 counter value arrives compare value diCompareValue2, and at this time counter value 8000 will be immediately output to **diCapValue2**, and the **bCapValue2_Complete** flag will be set to ON.
4. Same as Example1, even if the **bCapValue1_Complete/bCapValue2_Complete** flags are not cleared to OFF, the new captured values will still output to **diCapValue1/ diCapValue2** when channel 2 has a compare arrival event again.




DFB_DHCCNT_1
SM400
DFB_DHCCNT
EN
ENO
M0
bEnable
bBusy
0
byLocalID
diCurCnt
2
usiChannelNo
byCurSSI_SingleTurn
M100
bUpdate
byCurSSI_MultiTurn
usiAction
wSSIStatus
diActionValue
wRefCnt
bDir
uiCntStat
bError
ErrorCode
DFB_DHCCMP_0
SM400
DFB_DHCCMP
EN
ENO
M5
bEnable
bBusy
0
byLocalID
bMatch_1
2
usiChannelNo
bMatch_2
M101
bUpdate
bError
1000
diCompareValue1
ErrorCode
3
diActionValue1
2
wYno_1
2000
diCompareValue2
3
diActionValue2
wYno_2

Timing diagram:



- **Library**
  - DL_ASModuleAPI_AX3.library

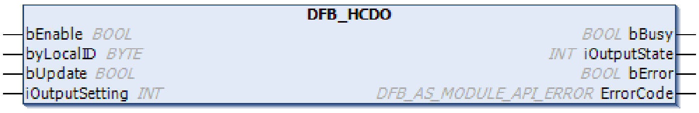
6.16 DFB_HCDO

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_HCDO：HC module output point control

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_HCDO	 DFB_HCDO
		bEnable BOOL
		byLocalID BYTE
		bUpdate BOOL
		iOutputSetting INT
		BOOL bBusy
		INT iOutputState
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_HCDO(
			bEnable:= ,
			byLocalID:= ,
			bUpdate:= ,
			iOutputSetting:= ,
			bBusy=> ,
			iOutputState=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute function block	BOOL	True/False(False)
byLocalID	Expand module number	BYTE	0 ~ 31(0)
bUpdate	Update parameter flag	BOOL	True/False(False)
iOutputSetting	Output point action setting	INT	0~15(0)

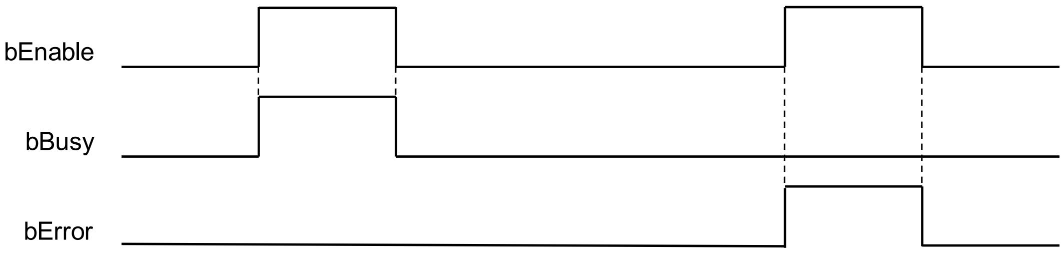
- **Output**

Name	Function	Data Type	Setting Value (Default value)
bBusy	Shows function block is been executed	BOOL	True/False(False)
iOutputState	Output point status display	INT	0~15(0)
bError	FB instruction error flag	BOOL	True/False(False)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to true	● When bEnable shifts to false
iOutputState	● Continuously update after bEnable	● When bEnable shifts to false
bError	● FB instruction execution error or input vale error	● When bEnable shifts to false
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported when AX-3 Series firmware is V1.0.2 and above.
3. The output point Y0.0~Y0.3 in this instruction is the output point on the 02HC module.
4. The HCDO instruction is AS02HC-A dedicated instruction, and it can control the output point Y0.0~Y0.3 and show the status of the output point Y0.0~Y0.3.
5. Complete the **iOutputSetting** setting before executing the instruction. When **bEnable** is started, **iOutputSetting** will be set once for the HC module, so the **iOutputSetting** set before starting will be taken as the initial state of the output point Y0.0~Y0.3. If users want to change the output point status during execution, set **iOutputSetting** as a new value, and then set the **bUpdate** flag as On. After the instruction completes parameter change, the **bUpdate** flag will be cleared.
6. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
7. **iOutputSetting** is the setting value of the output point action:

b15~b4	**b3**	**b2**	**b1**	**b0**
NA	Y0.3 action	Y0.2 action	Y0.1 action	Y0.0 action
NA	0：OFF 1：ON			

8 **iOutputState** is the status display of the output point

b15~b4	b3	b2	b1	b0
NA	Y0.3 status	Y0.2 status	Y0.1 status	Y0.0 status
NA	0：OFF 1：ON			

9 When the instruction is off, **iOutputState** will be 0.

10 When compare output instruction DHCCMP or table compare output instruction DHCCMPT starts, output point cannot be changed through **iOutputSetting.** However, the display of the **iOutputState** status can be continuously updated.

11 As stated before, if compare output instruction needs to set the initial value to output point, start the HCDO instruction first, and then start the DHCCMP or DHCCMPT instruction. If changing output point is needed, please shut the DHCCMP or DHCCMPT instruction, and then **IOutputSetting** of the HCDO instruction will update the latest output action.

12 During start-up, when any error condition occurs, this **bError** error flag will be set to ON. Please refer to error codes of **ErrorCode** for troubleshooting.

- **Library**

  ▪ DL_ASModuleAPI_AX3.library

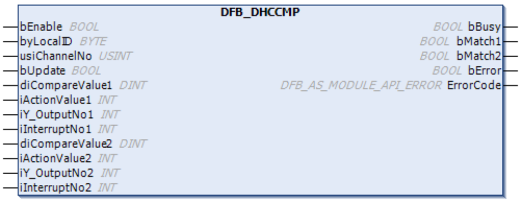
6.17 DFB_DHCCMP

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DHCCMP：HC module compare output.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DHCCMP		DFB_DHCCMP(
			bEnable:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			bUpdate:= ,
			diCompareValue1:= ,
			iActionValue1:= ,
			iY_OutputNo1:= ,
			iInterruptNo1:= ,
			diCompareValue2:= ,
			iActionValue2:= ,
			iY_OutputNo2:= ,
			iInterruptNo2:= ,
			bBusy=> ,
			bMatch1=> ,
			bMatch2=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute function block	BOOL	True/False(False)
byLocalID	Expand module number	BYTE	0 ~ 31(0)
usiChannelNo	Channel number	USINT	1 ~ 2(1)
bUpdate	Update parameter flag	BOOL	True/False(False)
diCompareValue1	Compare value 1	DINT	-2, 147, 483, 648 ~ 2, 147, 483, 648(0)
iActionValue1	Action of arrival compare value1	INT	0 ~ 8(0)
iY_OutputNo1	Y output point number of arrival compare value1	INT	0 ~ 3(0)
iInterruptNo1	Arrive compare value1, External Event interrupts number.	INT	0、400 ~ 431(0)
diCompareValue2	Compare value2	DINT	-2, 147, 483, 648 ~ 2, 147, 483, 648(0)
iActionValue2	Action of arrival compare value2	INT	0 ~ 8(0)

Name	Function	Data Type	Setting Value (Default value)
iY_OutputNo2	Y output point number of arrival compare value2	INT	0 ~ 3(0)
iInterruptNo2	Arrive compare value2, External Event interrupts number.	INT	0, 400 ~ 431(0)

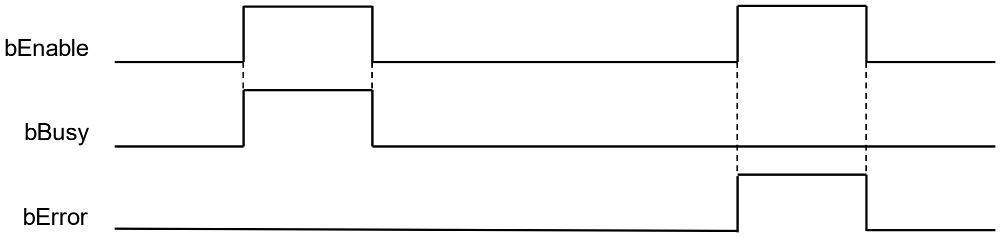
- **Output**

Name	Function	Data Type	Setting Value (Default value)
bBusy	Shows function block is been executed	BOOL	True/False(False)
bMatch1	Arrival compare value1 flag	BOOL	True/False(False)
bMatch2	Arrival compare value2 flag	BOOL	True/False(False)
bError	FB instruction error flag	BOOL	True/False(False)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to true	● When bEnable shifts to false
bMatch1	● Arrival compare value1	● When bEnable shifts to false
bMatch2	● Arrival compare value2	● When bEnable shifts to false
bError	● FB instruction execution error or input value error	● When bEnable shifts to false
ErrorCode		

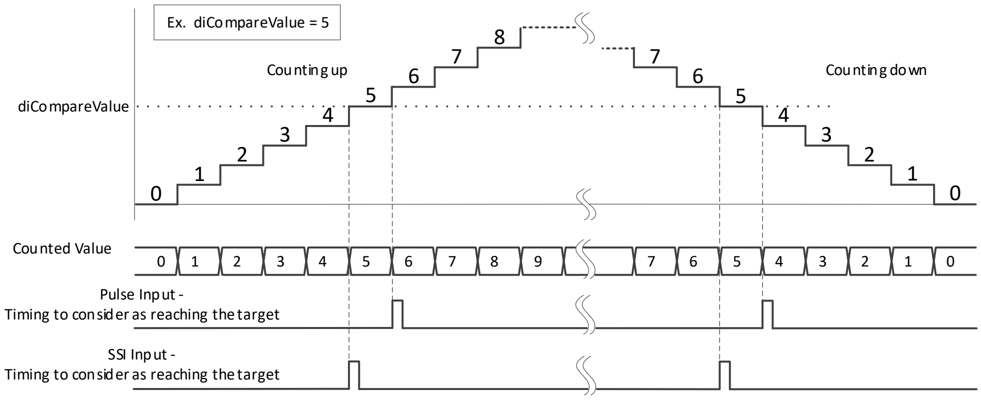
- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported when AX-3 Series firmware is V1.0.2 and above.
3. The output point Y0.0~Y0.3 in this instruction is the output point on the 02HC module.
4. The DHCCMP instruction is AS02HC-A dedicated instruction. When the counting up/counting down value reaches the preset compare value, please perform actions to output point and counter according to users' settings.
5. DHCCMP needs to be used with the DHCCNT instruction. Only when DHCCNT is started, counter value counts according to input signals and performs comparison action.
6. Complete **diCompareValue1, diCompareValue2, iActionValue1**, **iActionValue2**, **iY_OutputNo1**, **iY_OutputNo2, iInterruptNo1, iInterruptNo2** parameters setting before executing the instruction. When the instruction is enabled, the parameters of the AS02HC-A module will be set once.
7. **diCompareValue1, diCompareValue2**, **iActionValue1**, **iActionValue2**, **iY_OutputNo1**, **iY_OutputNo2, iInterruptNo1, iInterruptNo2** parameters can be changed during execution. The change method is to set new values first, and then set the **bUpdate** flag to On. When this instruction completes changing compare value, the instruction will clear the **bUpdate** flag to Off, and clear the **bMatch1** and **bMatch2** to Off.
8. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
9. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.
10. **diCompareValue1** and **diCompareValue2** are compare value number 1 and compare value number 2 respectively (diCompareValue1 and diCompareValue2 need to be different values). When the **usiChannelNo** channel counter value reaches the compare value of the number, the corresponding **bMatch1** and **bMatch2** flags will be set to On, and the specified **iY_OutputNo1** and **iY_OutputNo2** output points will perform actions according to **iActionValue1** and **iActionValue2.**

    When selecting the counter form as Linear Counter, **diCompareValue1** and **diCompareValue2** need to meet "Minimum counter value < **diCompareValue** < Maximum counter value". When using SSI encoder and set the counter value as Absolute Position, **diCompareValue1** and **diCompareValue2** need to meet "0 ≤ **diCompareValue** < $2^{(\mathrm{MT}+\mathrm{ST}^{\mathrm{length}})}$".
11. Please note that compare arrival timing will be different according to the input interfaces "pulse input" or "SSI input" as shown in the following figure.



    Take **diCompareValue** = 5 as an example, SSI input compare arrival timing occurs in the counter value 4→5 and 6→5 instantaneous. For pulse input, when counting up, compare arrival timing occurs in the counter value 5→6 instantaneous; when counting down, compare arrival timing occurs in the counter value 5→4 instantaneous.
12. **iActionValue1** and **iActionValue2** are the specified execution actions when compare arrival. The functions are as

below:

iActionValue1	Function	iActionValue2	Function	Note
0	No action	0	No action	
1	**iY_OutputNo1** specifies output point Off.	1	**iY_OutputNo2** specifies output point Off.	
2	**iY_OutputNo1** specifies output point On.	2	**iY_OutputNo2** specifies output point On.	
3	**iY_OutputNo1** specifies output point Toggle output (alternating ON/OFF)	3	**iY_OutputNo2** specifies output point Toggle output (alternating ON/OFF)	
4	**iY_OutputNo1** specifies output point Off + clears the channel counter value.	4	**iY_OutputNo2** specifies output point Off + clears the channel counter value.	Since the **bMatch1** and **bMatch2** flags will be cleared when clearing the counter value, it is not suggested to judge the two bMatch flags when selecting these five modes.
5	**iY_OutputNo1** specifies output point On + clears the channel counter value.	5	**iY_OutputNo2** specifies output point On + clears the channel counter value.	
6	**iY_OutputNo1** specifies output point Toggle (alternating ON/OFF) + clears the channel counter value.	6	**iY_OutputNo2** specifies output point Toggle (alternating ON/OFF) + clears the channel counter value.	
7	Clear the channel counter value	7	Clear the channel counter value	
8	Clear the channel counter value + **iY_OutputNo1/iY_OutputNo2** specifies output point Off.	8	Clear the channel counter value + **iY_OutputNo1/iY_OutputNo2** specifies output point Off.	

Note: There will be a little delay between compare arrival occurrence and **iActionValue** execution. The maximum delay time is 100us.

13. **iY_OutputNo1** and **iY_OutputNo2** specify Y output point numbers for compare arrival number 1 and number 2 respectively. After compare instruction starts, the HCDO instruction will not be able to control output point:

iY_OutputNo1/iY_OutputNo2	Specified Output Point
0	Y0.0
1	Y0.1
2	Y0.2
3	Y0.3

14. **iInterruptNo1** and **iInterruptNo2** are specified interrupt numbers. The functions are as below:

iInterruptNo	Action
0	Not to set interrupt
4□□	□□ = 00~31, Interrupt_400_ModuleIN0 to Interrupt_431_ModuleIN31 of the corresponding External Event The corresponding Task will be started when compare arrives.

15. **bMatch1** and **bMatch2** are the status flag display for the comparison values of compare arrival number1 and number2. When the counter value of the specified channel and number1 compare value & number compare value arrive, the corresponding **bMatch1** or **bMatch2** flag will be set to On; When the host STOP, clear counter, DHCCMP

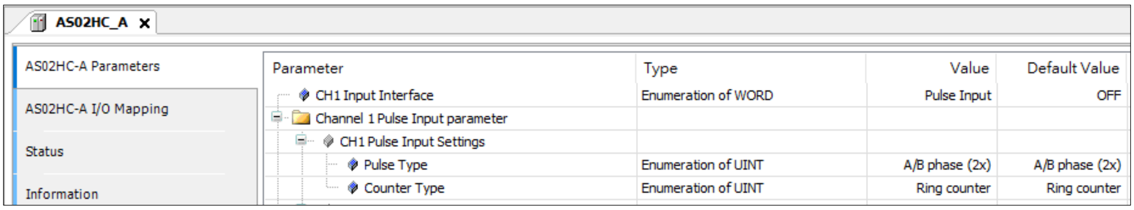
is OFF➔ON, or the **bUpdate** flag is set, the instruction will clear the **bMatch1** and **bMatch2** flags to Off.

16. When the instruction is off, the relevant compare values and the output functions will not be updated.

17. If any error situation occurs during startup, the **bError** error flag will be set to ON. Users can refer to **ErrorCode** for troubleshooting.

Programming Example 1: Control Y point output when compare arrives.

1. **Set** CH1 Input Interface in AS02HC-A Parameters as “Pulse Input”, set Pulse Type as “A/B phase(2x)”, and set Counter Type as “Ring counter”, as the following:



AS02HC_A

AS02HC-A Parameters
AS02HC-A I/O Mapping
Status
Information

Parameter	Type	Value	Default Value
CH1 Input Interface	Enumeration of WORD	Pulse Input	OFF
Channel 1 Pulse Input parameter			
CH1 Pulse Input Settings			
Pulse Type	Enumeration of UINT	A/B phase (2x)	A/B phase (2x)
Counter Type	Enumeration of UINT	Ring counter	Ring counter

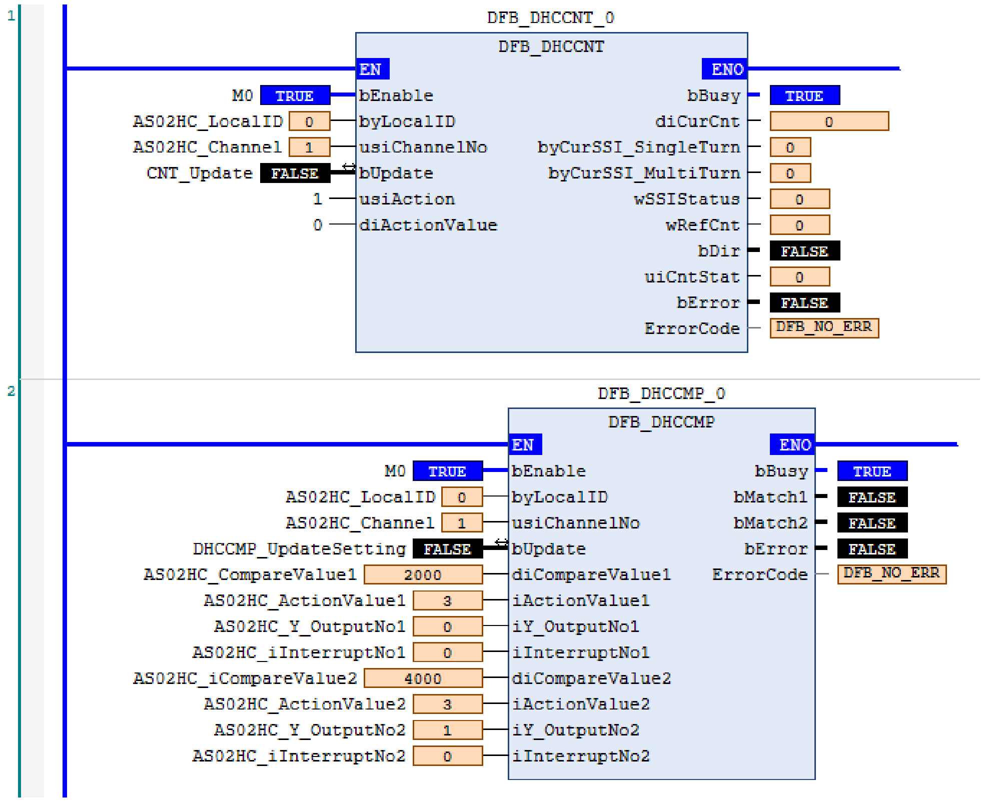
2. Set **diCompareValue1** to 2000, set **iActionValue1** to 3, set **iY_OutputNo1** to 0, and set **iInterruptNo1** to 0; set **diCompareValue2** to 4000, set **iActionValue2** to 3, set **iY_OutputNo2** to 1, and set **iInterruptNo2** to 0.

3. When setting M0 = ON, DHCCNT counter starts to count. At the same time, set the DHCCMP parameter for the module, and start to wait for counter compare arrival.

4. When counter values reach diCompareValue1, the bMatch1 flag is set to ON, and Y0.0 is OFF➔ON.

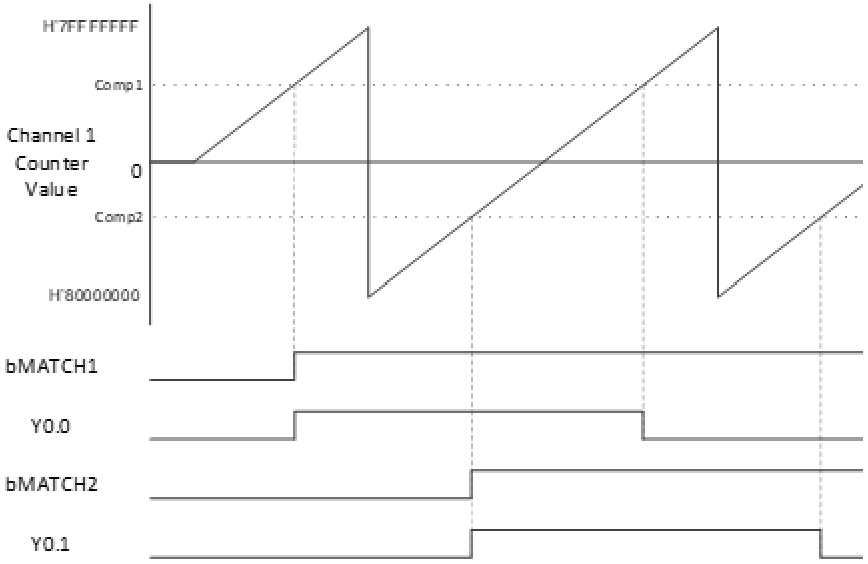
5. Continuous counting until **diCompareValue2** is reached, and the bMatch2 flag is set to ON, and Y0.1 is OFF➔ON.

6. Continuous counting until **diCompareValue1** is reached again, Y0.0 is ON➔OFF; Continuous counting until **diCompareValue2** is reached again, Y0.1 is ON➔OFF.



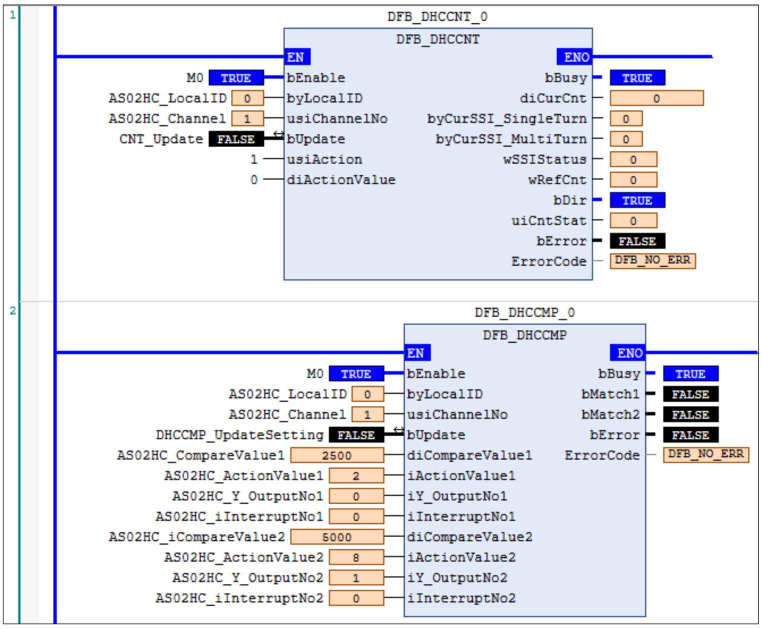
```
1                          DFB_DHCCNT_0
                            DFB_DHCCNT
                     EN                              ENO
             M0 TRUE     bEnable                 bBusy  TRUE
 AS02HC_LocalID 0        byLocalID            diCurCnt  0
 AS02HC_Channel 1        usiChannelNo byCurSSI_SingleTurn 0
 CNT_Update FALSE        bUpdate       byCurSSI_MultiTurn 0
              1          usiAction            wSSIStatus 0
              0          diActionValue           wRefCnt 0
                                                    bDir FALSE
                                                uiCntStat 0
                                                  bError FALSE
                                               ErrorCode DFB_NO_ERR

2                          DFB_DHCCMP_0
                            DFB_DHCCMP
                     EN                              ENO
                  M0 TRUE     bEnable            bBusy  TRUE
      AS02HC_LocalID 0        byLocalID        bMatch1  FALSE
      AS02HC_Channel 1        usiChannelNo     bMatch2  FALSE
 DHCCMP_UpdateSetting FALSE   bUpdate           bError  FALSE
 AS02HC_CompareValue1 2000    diCompareValue1 ErrorCode DFB_NO_ERR
  AS02HC_ActionValue1 3       iActionValue1
   AS02HC_Y_OutputNo1 0       iY_OutputNo1
 AS02HC_iInterruptNo1 0       iInterruptNo1
AS02HC_iCompareValue2 4000    diCompareValue2
  AS02HC_ActionValue2 3       iActionValue2
   AS02HC_Y_OutputNo2 1       iY_OutputNo2
 AS02HC_iInterruptNo2 0       iInterruptNo2
```

Timing Diagram:



Programming Example2: Clear the counter value when compare arrives.

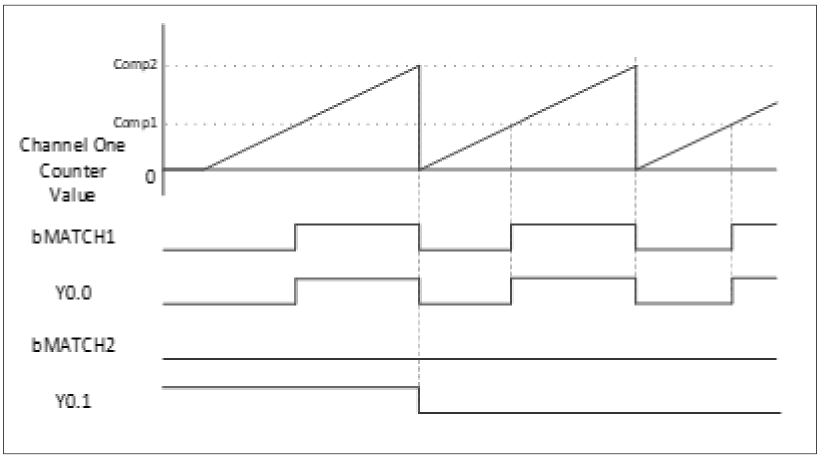
1. Set **diCompareValue1** to 2500, set **iActionValue1** to 2, set **iY_OutputNo1** to 0 and set **iInterruptNo1** to 0; set **diCompareValue2** to 5000, set **iActionValue2** to 8, set **iY_OutputNo2** to 1, and set **iInterruptNo2** to 0.
2. When setting M0 = ON, DHCCNT counter starts to count. At the same time, set the DHCCMP parameter for the module, and start to wait for counter compare arrival.
3. When counter values reach **diCompareValue1**, the **bMatch1** flag is set to ON, and Y0.0 is set to ON.
4. Continuous counting until **diCompareValue2** is reached. Since the action of **iActionValue2** = 8 is “clear the channel counter value + clear **iY_OutputNo1** and **iY_OutputNo2** specified output point”, counter value is cleared to 0, **bMatch1** and **bMatch2** are cleared to False, and Y0.0 & Y0.1 are cleared to OFF.
5. Continuous counting until **diCompareValue1** is reached again, the **bMatch1** flag will be set to on again, Y0.0 will be set to ON, and so on.



DFB_DHCCNT_0
DFB_DHCCNT
EN ENO
M0 TRUE bEnable bBusy TRUE
AS02HC_LocalID 0 byLocalID diCurCnt 0
AS02HC_Channel 1 usiChannelNo byCurSSI_SingleTurn 0
CNT_Update FALSE bUpdate byCurSSI_MultiTurn 0
1 usiAction wSSIStatus 0
0 diActionValue wRefCnt 0
bDir TRUE
uiCntStat 0
bError FALSE
ErrorCode DFB_NO_ERR

DFB_DHCCMP_0
DFB_DHCCMP
EN ENO
M0 TRUE bEnable bBusy TRUE
AS02HC_LocalID 0 byLocalID bMatch1 FALSE
AS02HC_Channel 1 usiChannelNo bMatch2 FALSE
DHCCMP_UpdateSetting FALSE bUpdate bError FALSE
AS02HC_CompareValue1 2500 diCompareValue1 ErrorCode DFB_NO_ERR
AS02HC_ActionValue1 2 iActionValue1
AS02HC_Y_OutputNo1 0 iY_OutputNo1
AS02HC_iInterruptNo1 0 iInterruptNo1
AS02HC_iCompareValue2 5000 diCompareValue2
AS02HC_ActionValue2 8 iActionValue2
AS02HC_Y_OutputNo2 1 iY_OutputNo2
AS02HC_iInterruptNo2 0 iInterruptNo2

Timing Diagram:

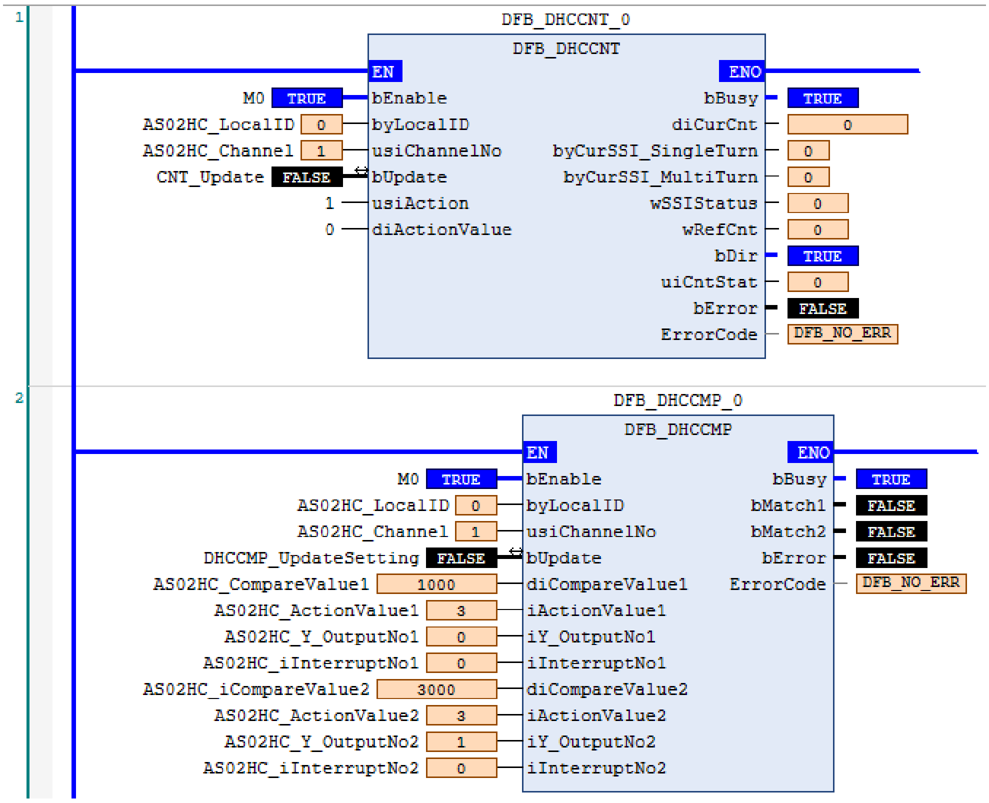


Programming Example3: Update the compare value

1. Set **diCompareValue1** to 1000, set **iActionValue1** to 3, set **iY_OutputNo1** to 0,and set **iInterruptNo1** to 0; set **diCompareValue2** to 3000, set **iActionValue2** to 3, set **iY_OutputNo2** to 1, and set **iInterruptNo2** to 0.
2. When setting M0 = ON, DHCCNT counter starts to count. At the same time, set the DHCCMP parameter for the module, and start to wait for counter compare arrival.
3. When counter value reaches **diCompareValue1**, the **bMatch1** flag will be set to ON, and Y0.0 will be OFF➔ON.
4. When counter value reaches **diCompareValue2**, the **bMatch2** flag will be set to ON, and Y0.1 will be ON➔OFF.
5. Then, set new compare value: set **diCompareValue1** to 5000, set **iActionValue1** to 3, set **OutputNo1** to 0, set **iInterruptNo1** to 0, set **diCompareValue2** to 7000, set **iActionValue2** to 6, set **iY_OutputNo2** to 1, and set **iInterruptNo2** to 0.
6. Set the **bUpdate** flag to ON. When the setting completes, the **bUpdate** flag will be automatically cleared to OFF. At the same time, the **bMatch1** and **bMatch2** flags will be cleared to OFF.
7. When counter value reaches new compare value **diCompareValue1**, the **bMatch1** flag will be set to ON, and Y0.0

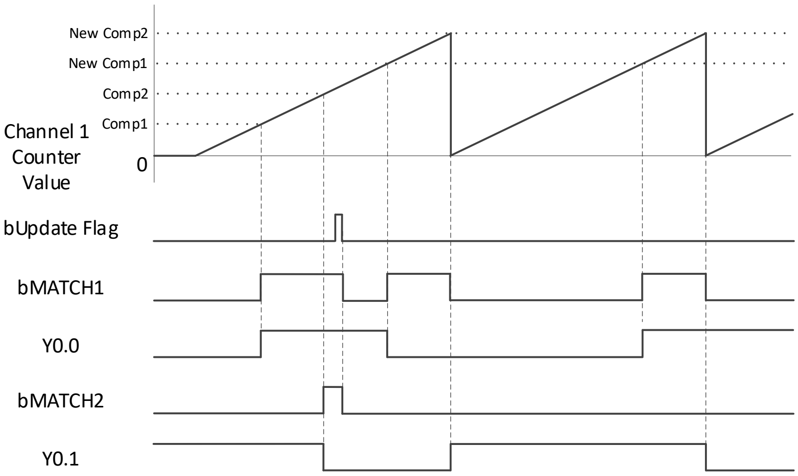
will be ON➔OFF.

8. When counter value reaches new compare value **diCompareValue2**, Y0.1 will be OFF➔ON. Because **iActionValue2** = 6 include Reset, counting value will be cleared to 0. At the same time, the **bMatch1** and **bMatch2** flags will be cleared to OFF.



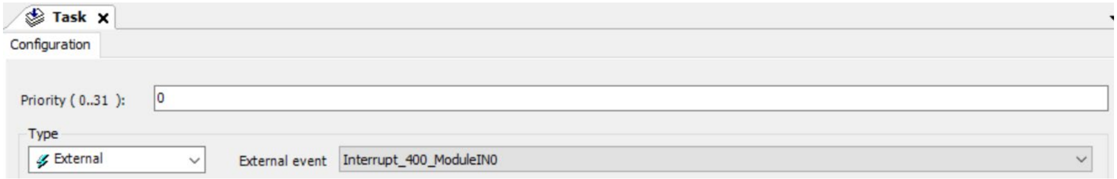
```
1
                         DFB_DHCCNT_0
                         DFB_DHCCNT
                  EN                                 ENO
M0            TRUE   bEnable                   bBusy      TRUE
AS02HC_LocalID   0   byLocalID               diCurCnt     0
AS02HC_Channel   1   usiChannelNo  byCurSSI_SingleTurn    0
CNT_Update   FALSE   bUpdate        byCurSSI_MultiTurn    0
                 1   usiAction               wSSIStatus   0
                 0   diActionValue               wRefCnt  0
                                                    bDir  TRUE
                                                uiCntStat 0
                                                  bError  FALSE
                                               ErrorCode  DFB_NO_ERR

2
                         DFB_DHCCMP_0
                         DFB_DHCCMP
                            EN                         ENO
M0                      TRUE   bEnable           bBusy    TRUE
AS02HC_LocalID             0   byLocalID         bMatch1  FALSE
AS02HC_Channel             1   usiChannelNo      bMatch2  FALSE
DHCCMP_UpdateSetting   FALSE   bUpdate           bError   FALSE
AS02HC_CompareValue1    1000   diCompareValue1   ErrorCode DFB_NO_ERR
AS02HC_ActionValue1        3   iActionValue1
AS02HC_Y_OutputNo1         0   iY_OutputNo1
AS02HC_iInterruptNo1       0   iInterruptNo1
AS02HC_iCompareValue2   3000   diCompareValue2
AS02HC_ActionValue2        3   iActionValue2
AS02HC_Y_OutputNo2         1   iY_OutputNo2
AS02HC_iInterruptNo2       0   iInterruptNo2
```

Timing Diagram:



Programming Example4: Compare arrival interruption

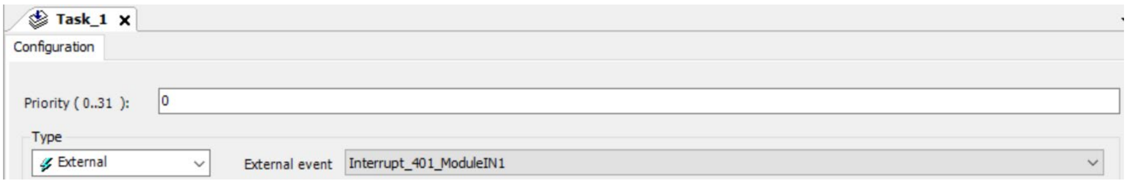
1. DHCCMP can also configure compare arrival interruption. Set **iInterruptNo1** and **iInterruptNo2** to 400 and 401 respectively, and configure two Tasks at the same time. In the tasks, select Interrupt_400_ModuleIN0 or Interrupt_401_ModuleIN1 for External event as shown in the following figures.



Priority (0..31): 0

Type: External

External event: Interrupt_400_ModuleIN0

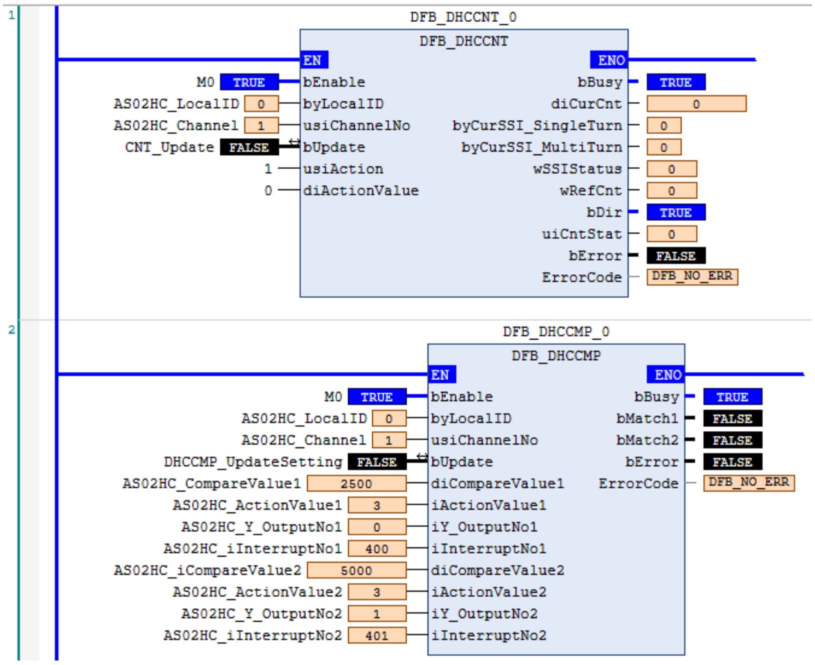


Priority (0..31): 0

Type: External

External event: Interrupt_401_ModuleIN1

2. Set **diCompareValue1** to 2500, set **iActionValue1** to 3, set **iY_OutputNo1** to 0 set **iInterruptNo1** to 400; set **diCompareValue2** to 5000, set **iActionValue2** to 3, set **iY_OutputNo2** to 1, and set **iInterruptNo2** to 401.
3. When setting M0 = ON, DHCCNT counter starts to count. At the same time, set the DHCCMP parameter for the module, and start to wait for counter compare arrival.
4. When counter value reaches **diCompareValue1,** External event corresponding to **iInterruptNo1** will be executed, and Y0.0 will be toggled (alternating ON/OFF).
5. When counter value reaches **diCompareValue2,** External event corresponding to **iInterruptNo2** will be executed, and Y0.1 will be toggled (alternating ON/OFF).



```
1                          DFB_DHCCNT_0
                           DFB_DHCCNT
                           EN                                ENO
                M0  TRUE   bEnable                         bBusy   TRUE
   AS02HC_LocalID  0       byLocalID                    diCurCnt   0
   AS02HC_Channel  1       usiChannelNo      byCurSSI_SingleTurn   0
   CNT_Update  FALSE       bUpdate            byCurSSI_MultiTurn   0
                   1       usiAction                   wSSIStatus  0
                   0       diActionValue                  wRefCnt  0
                                                             bDir  TRUE
                                                        uiCntStat  0
                                                           bError  FALSE
                                                        ErrorCode  DFB_NO_ERR

2                                     DFB_DHCCMP_0
                                      DFB_DHCCMP
                                      EN                    ENO
                         M0  TRUE     bEnable             bBusy   TRUE
            AS02HC_LocalID  0         byLocalID          bMatch1  FALSE
            AS02HC_Channel  1         usiChannelNo       bMatch2  FALSE
   DHCCMP_UpdateSetting  FALSE        bUpdate             bError  FALSE
   AS02HC_CompareValue1  2500         diCompareValue1  ErrorCode  DFB_NO_ERR
    AS02HC_ActionValue1  3            iActionValue1
     AS02HC_Y_OutputNo1  0            iY_OutputNo1
   AS02HC_iInterruptNo1  400          iInterruptNo1
  AS02HC_iCompareValue2  5000         diCompareValue2
    AS02HC_ActionValue2  3            iActionValue2
     AS02HC_Y_OutputNo2  1            iY_OutputNo2
   AS02HC_iInterruptNo2  401          iInterruptNo2
```

- **Library**
  - DL_ASModuleAPI_AX3.library

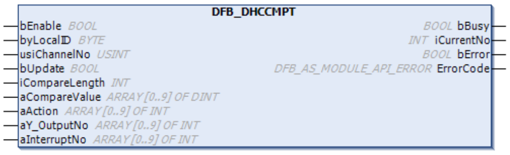
6.18 DFB_DHCCMPT

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DHCCMPT：HC module table compare output

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DHCCMPT		DFB_DHCCMPT(
			bEnable:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			bUpdate:= ,
			iCompareLength:= ,
			aCompareValue:= ,
			aAction:= ,
			aY_OutputNo:= ,
			aInterruptNo:= ,
			bBusy=> ,
			iCurrentNo=> ,
			bError=> ,
			ErrorCode=>);

DFB_DHCCMPT
bEnable BOOL
byLocalID BYTE
usiChannelNo USINT
bUpdate BOOL
iCompareLength INT
aCompareValue ARRAY [0..9] OF DINT
aAction ARRAY [0..9] OF INT
aY_OutputNo ARRAY [0..9] OF INT
aInterruptNo ARRAY [0..9] OF INT
BOOL bBusy
INT iCurrentNo
BOOL bError
DFB_AS_MODULE_API_ERROR ErrorCode

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute function block	BOOL	True/False (False)
byLocalID	Expand module number	BYTE	0 ~ 31 (0)
usiChannelNo	Channel number	USINT	1 ~ 2 (1)
bUpdate	Update parameter flag	BOOL	True/False (False)
iCompareLength	Compare table groups lengths	INT	2~10 (2)
aCompareValue	32-bit compare value storage sources of compare table	ARRAY[0..9] OF DINT	-2, 147, 483, 648 ~ 2, 147, 483, 648 (0)
aAction	Action storage sources of compare arrival of compare table	ARRAY[0..9] OF INT	0 ~ 8 (0)
aY_OutputNo	Y output point number storage sources of compare arrival of compare table	ARRAY[0..9] OF INT	0 ~ 3 (0)
aInterruptNo	External Event interruption number storage sources of compare arrival of compare table	ARRAY[0..9] OF INT	0, 400 ~ 431 (0)

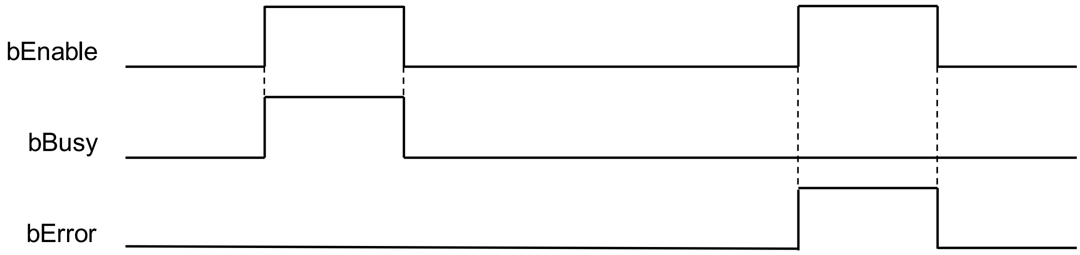
- **Output**

Name	Function	Data Type	Setting Value (Default value)
bBusy	Shows function block is been executed	BOOL	True/False (False)
iCurrentNo	The number of groups that the comparator has executed so far.	INT	0 ~ 10 (0)
bError	FB instruction error flag	BOOL	True/False (False)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to true	● When bEnable shifts to false
iCurrentNo	● Display according to the number of the group that currently has compare arrived.	● When bEnable shifts to false
bError	● FB instruction execution error or input error	● When bEnable shifts to false
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported when AX-3 Series firmware is V1.0.2 and above.
3. The output point Y0.0~Y0.3 in this instruction is the output point on the 02HC module.
4. The DHCCMPT instruction is AS02HC-A dedicated instruction. Its compare action is “cyclical” compare according to the set groups. When the number of comparison step by step reaches the last compare value, the next compare value will be reset as the first compare value. The DHCCMPT instruction is limited to be used when the counter counts in only one direction (Changing the direction will cause incorrect action), and can be set up to 10 compare values. Compare values need to be sorted by incremental or decremental numbers. For incremental sorting, counter needs to counting up, and only positive numbers for compare value. For decremental sorting, counter needs to counting down, and only negative numbers for compare value. When the counting reaches the preset compare value, actions will be performed to output pint and counter according to users setting, and the request to the host interrupts.
5. DHCCMPT needs to be used with the DHCCNT instruction. Only when DHCCNT starts, counter value counts according to the input signal, and perform compare action.
6. Complete **iCompareLength, aCompareValue, aAction, aY_OutputNo,** and **aInterruptNo** parameter setting

before executing the instruction. **bEnable** will write the parameter once when it is first started.

7. The parameters of **iCompareLength, aCompareValue, aAction, aY_OutputNo, and aInterruptNo** can be changed during execution. The changing methods is to set them as new values, and then set the **bUpdate** flag to On. When the changing is complete, the instruction will clear the **bUpdate** flag to Off.

8. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.

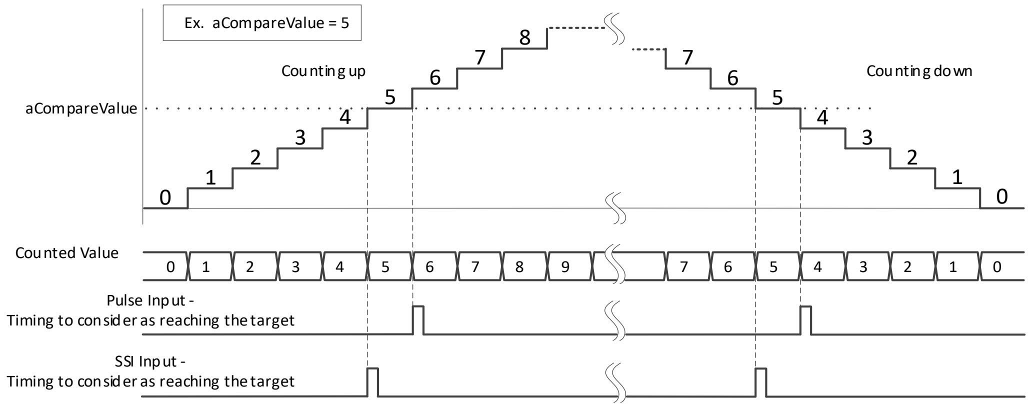
9. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.

10. **iCompareLength** is the number of groups of compare tables. Its compare action is cyclical according to the set group numbers. The group number length can be set to 2 ~ 10. When the set value is out of this range, the instruction will not be executed, and error code shows.

11. **aCompareValue** is the storage source of the compare vale, and its type is ARRAY[0..9] OF DINT. Compare values needs to be sorted by incremental or decremental numbers (each compare value needs to be different). For incremental sorting, counter needs to counting up, and only positive numbers for compare value. For decremental sorting, counter needs to counting down, and only negative numbers for compare value.
When selecting the counter form as Linear Counter, each vale in aCompareValue needs to meet “Minimum counter value < aCompareValue < Maximum counter value”.

    When using SSI encoder and the counter form is set as Absolute Position, each value in aCompareValue needs to meet “0 ≤ aCompareValue ≤$2^{(\mathrm{MT}+\mathrm{ST}^{\mathrm{length}})}$”.

12. Please note that compare arrival timing will be different according to the input interface “Pulse input” or “SSI input” as shown in the following figure.



Take **aCompareValue** = 5 as an example, SSI input compare arrival timing occurs in the counter value 4→5 and 6→5 instantaneous. For pulse input, when counting up, compare arrival timing occurs in the counter value 5→6 instantaneous; when counting down, compare arrival timing occurs in the counter value 5→4 instantaneous.

13. **aAction** is the specified action code when compare arrives. The functions are as below:

aAction	Function	Note
0	No action	
1	**aY_OutputNo** specifies the output point Off.	
2	**aY_OutputNo** specifies the output point On.	
3	**aY_OutputNo** specifies the output point Toggle output (alternating ON/OFF).	
4	**aY_OutputNo** specifies the output point Off + clear the channel counter value.	In addition to clearing the channel counter value, clear **iCurrentNo** to 0.
5	**aY_OutputNo** specifies the output point On + clear the channel counter value.	
6	**aY_OutputNo** specifies the output point Toggle (alternating ON/OFF) + clear the channel counter value.	
7	Clear the channel counter value.	
8	Clear the channel counter value＋ all **aY_OutputNo** specify the output point Off.	Same as **aAction** = 7 clearance, and clear all specified **aY_OutputNo** output point.

Note: There will be a little delay between compare arrival occurrence and Action execution. The maximum delay time is 100us.

14. **aY_OutputNo** is specified output point number. The function is as the following:

aY_OutputNo	Output Point
0	Y0.0
1	Y0.1
2	Y0.2
3	Y0.3

15. **aInterruptNo** is specified interruption number. The function is as the following:

aInterruptNo	動作
0	Not to set interruption
4□□	□□ = 00~31, Interrupt_400_ModuleIN0 to Interrupt_431_ModuleIN31 of the corresponding External Event The corresponding Task will be started when compare arrives.

16. **iCurrentNo** displays the number of the group that currently has compare arrived. For example, when the counter value is 200, which is lower than the group number 1 in the following compare table, the **iCurrentNo** value is 0; when the counter value reaches 1000, which equals to the group number 1 in the following compare table, the **iCurrentNo** value is 1.

17. Please refer to the following table for the operator description of **aCompareValue**, **aAction, aY_OutputNo,**

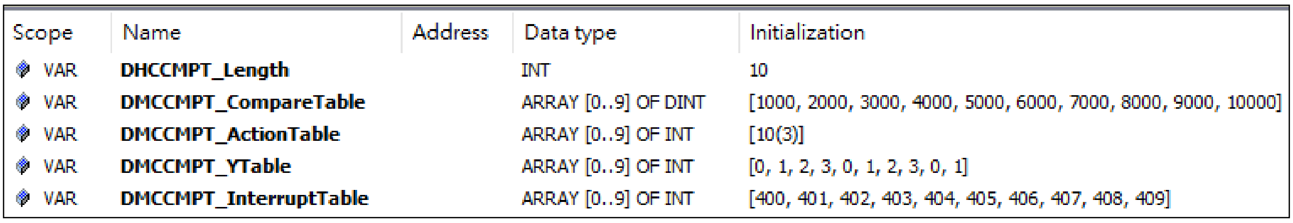
aInterruptNo, and **iCurrentNo.** In the table, the **iCompareLength** value is assumed as 6.

ICurrentNo Group Number	**aCompareValue** Source Compare Value	**aAction** Compare Arrival Specified Action	**aY_OutputNo** Output Point Number	**aInterruptNo** Interruption Number
1	1000	2(On)	0(Y0.0)	400
2	2000	2(On)	1(Y0.1)	401
3	3000	3(Toggle)	0(Y0.0)	402
4	4500	2(On)	2(Y0.2)	403
5	5500	1(Off)	3(Y0.3)	404
6	6500	1(Off)	1(Y0.1)	405

18. When the instruction is started (**bEnable** is from Off to On) and the **bUpdate** flag is set to On to change the parameter, the instruction will compare all the compare values in the compare table set by users according to the current counter vale. The group values that are smaller than the current counter value will execute the compare arrival action. For example, when the instruction starts with the counter value 3500, the actions numbered 1 ~3 (compare value < current counter value) in the compare table will be executed once with the compare arrival action (Y0.0=On, Y0.1=On, Y0.0 Toggle (alternating ON/OFF)), and **iCurrentNo** will be set to 3.
19. If **bEnable** is from On to Off, it means that shutting the table compare output function. **iCurrentNo** status display will return to 0. The output point state is not changed by the instruction being off. Counter remains counting, but no more comparison.
20. If any error situation occurs during startup, the **bError** error flag will be set to ON. Users can refer to **ErrorCode** for troubleshooting.

Programming Example:

1. Set the parameters of the relevant input pins as shown in the following figure (The settings in the following figure are equivalent to those shown in the following table).



Scope	Name	Address	Data type	Initialization
VAR	DHCCMPT_Length		INT	10
VAR	DMCCMPT_CompareTable		ARRAY [0..9] OF DINT	[1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000]
VAR	DMCCMPT_ActionTable		ARRAY [0..9] OF INT	[10(3)]
VAR	DMCCMPT_YTable		ARRAY [0..9] OF INT	[0, 1, 2, 3, 0, 1, 2, 3, 0, 1]
VAR	DMCCMPT_InterruptTable		ARRAY [0..9] OF INT	[400, 401, 402, 403, 404, 405, 406, 407, 408, 409]

iCurrentNo Group Number	**aCompareValue** Source Compare Value	**aAction** Compare Arrival Specified Action	**aY_OutputNo** Output Point Number	**aInterruptNo** Interruption Number
1	1000	3 (Toggle)	0 (Y0.0)	400
2	2000	3 (Toggle)	1 (Y0.1)	401
3	3000	3 (Toggle)	2 (Y0.2)	402
4	4000	3 (Toggle)	3 (Y0.3)	403
5	5000	3 (Toggle)	0 (Y0.0)	404
6	6000	3 (Toggle)	1 (Y0.1)	405
7	7000	3 (Toggle)	2 (Y0.2)	406
8	8000	3 (Toggle)	3 (Y0.3)	407
9	9000	3 (Toggle)	0 (Y0.0)	408
10	10000	3 (Toggle)	1 (Y0.1)	409

2. When **EN** is OFF→ON, the instruction will compare all the compare values in the compare table set by users. The group values that are smaller than the current counter value will execute the compare arrival action. Because the counter value is still smaller than the first compare value 1000 when startup. No compare arrival specified action will be executed, and **iCurrentNo** is 0 currently.

3. When the counter value reaches 1000, compare arrives the first compare value (If it is pulse input, compare arrival specified action will be executed when the counter value is 1000→1001).

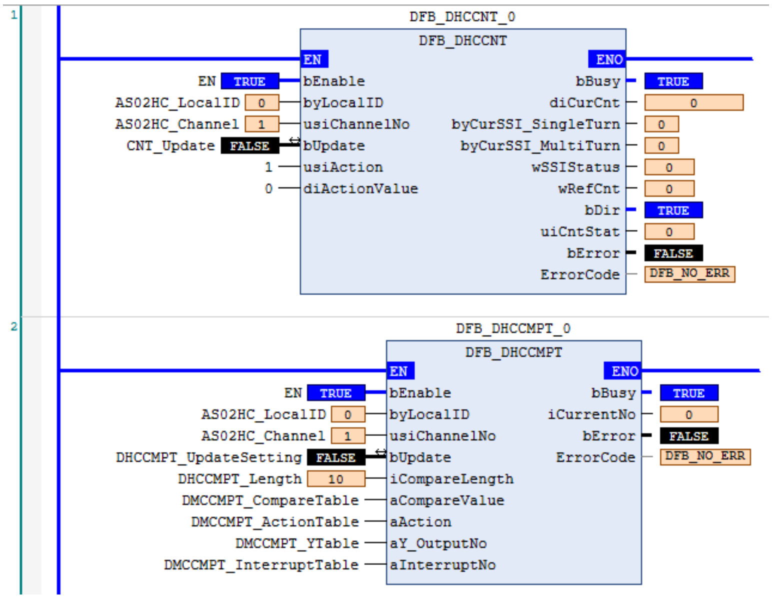
4. When the counter value reaches 2000, compare arrives the second compare value, Y0.1 is OFF➔ON, 401 interruption program is executed, and **iCurrentNo**=2.

5. When the counter value reaches 3000, compare arrives the third compare value, Y0.2 is OFF➔ON, 402 interruption program is executed, and **iCurrentNo**=3.

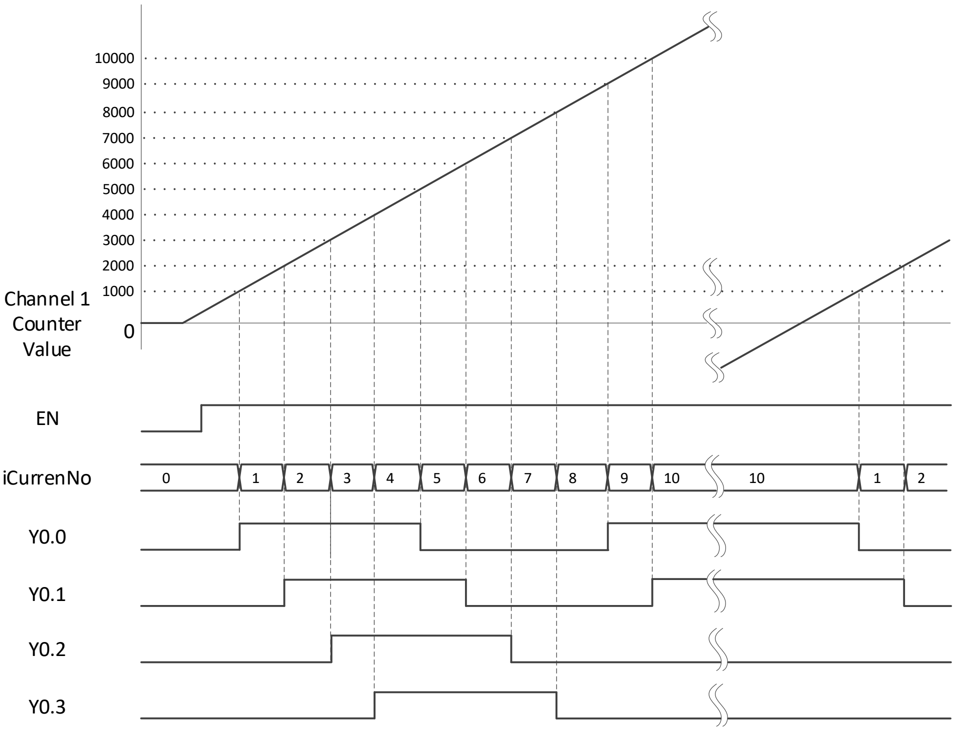
6. When the counter value reaches 4000, compare arrives the fourth compare value, Y0.3 is OFF➔ON, 403 interruption program is executed, and **iCurrentNo**=4.

7. Follow this rule to continue compare arrival from the fifth to tenth compare value. At this time **iCurrentNo**=10. Because the last comparison has done, next compare vale will be set as the first compare value 1000.

8. When the ring counter counts to 1000 again, compare arrives the first counter value, Y1.0 is ON➔OFF, 400 interruption program is executed, and **iCurrentNo**=1. Follow this rule to execute the subsequent comparison.



Timing Diagram:



- **Library**

  - DL_ASModuleAPI_AX3.library

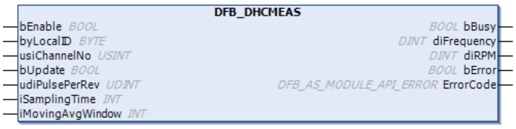
6.19 DFB_DHCMEAS

- **Supported products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DHCMEAS：HC module frequency and rotation speed measurement

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DHCMEAS	 DFB_DHCMEAS
		bEnable BOOL
		byLocalID BYTE
		usiChannelNo USINT
		bUpdate BOOL
		udiPulsePerRev UDINT
		iSamplingTime INT
		iMovingAvgWindow INT
		BOOL bBusy
		DINT diFrequency
		DINT diRPM
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DHCMEAS(
			bEnable:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			bUpdate:= ,
			udiPulsePerRev:= ,
			iSamplingTime:= ,
			iMovingAvgWindow:= ,
			bBusy=> ,
			diFrequency=> ,
			diRPM=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute function block	BOOL	True/False (False)
byLocalID	Expand module number	BYTE	0 ~ 31 (0)
usiChannelNo	Channel number	USINT	1 ~ 2 (1)
bUpdate	Update parameter flag	BOOL	True/False (False)
udiPulsePerRev	Counting quantity per lap	UDINT	1~4294967295 (1)
iSamplingTime	Sampling period (unit: ms)	INT	1 ~ 1000 (1)
iMovingAvgWindow	Average number of moves	INT	1 ~ 10 (1)

- **Output**

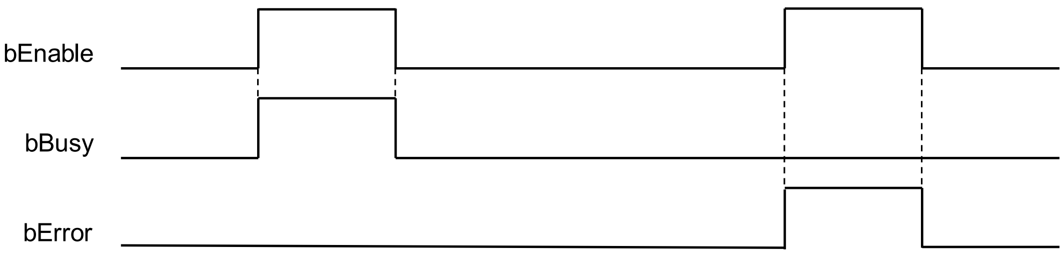
Name	Function	Data Type	Setting Value (Default value)
bBusy	Shows function block is been executed	BOOL	True/False (False)
diFrequency	Average measurement frequency results	DINT	-2, 147, 483, 648 ~ 2, 147, 483, 648 (0)
diRPM	Average rotation speed measurement results	DINT	-2, 147, 483, 648 ~ 2, 147, 483, 648 (0)
bError	FB instruction error flag	BOOL	True/False (False)

Name	Function	Data Type	Setting Value (Default value)
ErrorCode	Error code	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
bBusy	● When bEnable shifts to true.	● When bEnable shifts to false.
diFrequency	● Continuously update after bEnable	● When bEnable shifts to false.
diRPM	● Continuously update after bEnable	● When bEnable shifts to false.
bError	● FB instruction execution error or input value error	● When bEnable shifts to false.
ErrorCode		

- **Timing Diagram**



- **Function**

1. It is suggested that this instruction be placed under Main Task.
2. This function is supported when AX-3 Series firmware is V1.0.2 and above.
3. The output point Y0.0~Y0.3 in this instruction is the output point on the 02HC module.
4. The DHCMEAS instruction is AS02HC-A dedicated instruction. Its functions are frequency and rotation speed measurements.
5. DHCMEAS needs to be used with the DHCCNT instruction. Only when DHCCNT is started, the counter value counts according to input signals, and the measurement results are calculated by the change of the counter values.
6. Complete setting **udiPulsePerRev, iSamplingTime,** and **iMovingAvgWindow** parameters before executing the instruction. **bEnable** will write the parameter once when it is first started. If users want to change parameters during execution, the change method is to set new values first, and then set the **bUpdate** flag to On. When this instruction completes the change, the instruction will clear the **bUpdate** flag to Off.
7. **byLocalID** specifies module numbers. The number of the first module on the right of CPU is 0, the number of the second module on the right of CPU is 1, and so on. Regardless of any type of modules, all modules must be counted. The maximum number of modules is 32.
8. **usiChannelNo** specifies the channel numbers. The number of channel one is 1, and the number of channel two is 2.
9. **udiPulsePerRev** is the counter value of one rotation of the encoder, and its setting range is 1~4294967295 (H’00000001~H’FFFFFFFF).
10. **iSamplingTime** is sampling period, and its setting range is 1~1000 (unit: ms). According to the setting of **iSamplingTime, diFrequency** frequency measurement result output and **diRPM** rotation speed measurement result output will have different resolution.

$$\textbf{diFrequency}\ \text{resolution} = 1000 \div \textbf{iSamplingTime}\ (\text{unit: Hz})$$

$$\textbf{diRPM}\ \text{resolution} = 60000 \div (\textbf{iSamplingTime} \times \textbf{udiPulsePerRev})(\text{unit: rev/min})$$

Because the rotation speed calculation method is based on the CurCnt counter value of the beginning and end of the sampling period, the following situations need to be excluded when designing sampling periods.

Encoder Type	Counter Type	Factors to Effect Measurement Accuracy
Incremental encoder	Ring Counter	Displacement exceeds 2^{31} in the sampling period.
	Linear Counter	Displacement exceeds 2^{31} in the sampling period, or the counter value exceeds the upper/lower limit.
Absolute SSI encoder	Shows absolute position	Displacement exceeds $2^{(\mathrm{MT}+\mathrm{ST}^{\mathrm{length}})-1}$ in the sampling period.
	Ring counter	Displacement exceeds 2^{31} in the sampling period.

11. **iMovingAvgWindow** is the average number of times, which performs moving average to measurement results. Its setting range is 1 ~ 10.
12. **diFrequency** is the result of average measurement frequency (unit: Hz). The calculation method of the frequency is as below:

$$\textbf{diFrequency}(\text{Hz}) = \frac{\text{CurCnt}(\text{t} + \textbf{iSamplingTime}) - \text{CurCnt}(\text{t})}{\textbf{iSamplingTime}(\text{ms}) \times 10^{-3}}$$

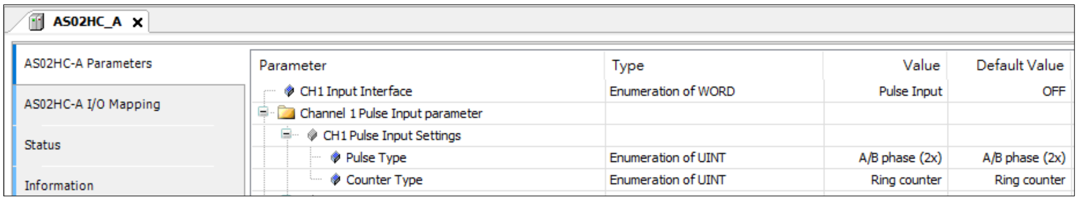
13. **diRPM** is the result of the average rotation speed measurement (unit: rev/min). The calculation method of the rotation speed is as below:

$$\text{diRPM}(\text{rev/min}) = \frac{(\ \text{CurCnt}(\text{t} + \textbf{iSamplingTime}) - \text{CurCnt}(\text{t})\ ) \times 60}{\textbf{udiPulsePerRev} \times \textbf{iSamplingTime}(\text{ms}) \times 10^{-3}}$$

14. When the instruction is shut off, it means that the measurement function is disabled, and **diFrequency** & **diRPM** will remain the same and no more update.
15. Please note that when the counter value is cleared or users change the counter value, the calculation result of **diFrequency** and **diRPM** of the sampling period will be affected.
16. If any error situation occurs during startup, the **bError** error flag will be set to ON. Users can refer to **ErrorCode** for troubleshooting.

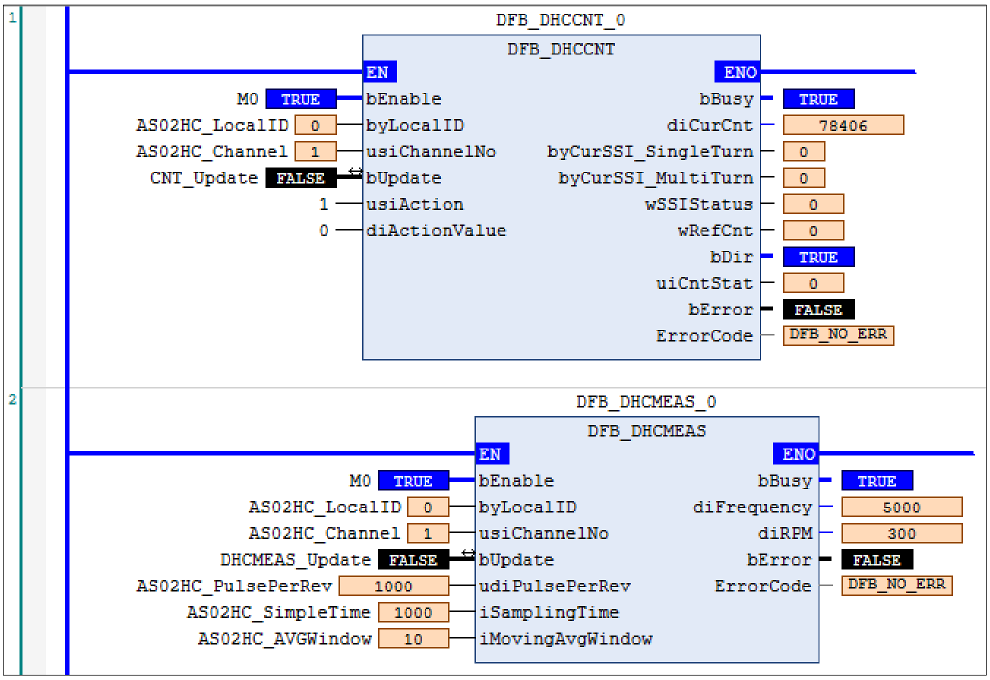
Programming Example: Incremental encoder

1. Set CH1 Input Interface in AS02HC-A Parameters to “Pulse Input”, set Pulse Type to “A/B phase (2x)”, and set Counter Type to “Ring counter” as shown in the following figure:



Parameter	Type	Value	Default Value
CH1 Input Interface	Enumeration of WORD	Pulse Input	OFF
Channel 1 Pulse Input parameter			
CH1 Pulse Input Settings			
Pulse Type	Enumeration of UINT	A/B phase (2x)	A/B phase (2x)
Counter Type	Enumeration of UINT	Ring counter	Ring counter

2. Set **udiPulsePerRev** to 1000, set **iSamplingTime** to 1000, and set **iMovingAvgWindow** to 10.
3. When setting M0=ON, the DHCCNT counter starts to count. At the same time, set the DHCMEAS parameters to the module, and start to measure frequency and rotation speed. The pulse number counted every 1000ms is displayed in **diFrequency** and **diRPM.**
4. When the motor operation frequency is 5kHz, **diFrequency** shows 5kHz, and **diRPM** shows 300rev/min.



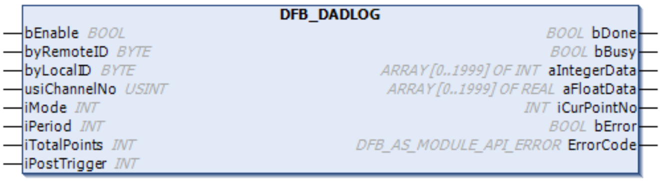
- **Library**
  - DL_ASModuleAPI_AX3.library

6.20 DFB_DADLOG

- **Supported products**
  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DADLOG：Analog input module data record.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_DADLOG		DFB_DADLOG(
			bEnable:= ,
			byRemoteID:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			iMode:= ,
			iPeriod:= ,
			iTotalPoints:= ,
			iPostTrigger:= ,
			bDone=> ,
			bBusy=> ,
			aIntegerData=> ,
			aFloatData=> ,
			iCurPointNo=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute function block	BOOL	True/False (False)
byRemoteID*	CPU or remote module	BYTE	0: CPU 1~15: remote module(0)
byLocalID	Expansion module ID	BYTE	0 ~ 31 (0)
usiChannelNo	Specified channel number	USINT	1 ~ 2 (1)
iMode	Output mode setting	INT	0 ~ 3 (0)
iPeriod	Speed fetch cycle time	INT	10~1000(1000)
iTotalPoints	total number of records	INT	Please refer to the description in the Function description (1)
iPostTrigger	Number of records after triggering	INT	Please refer to the description in the Function description (0)

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

- **Output**

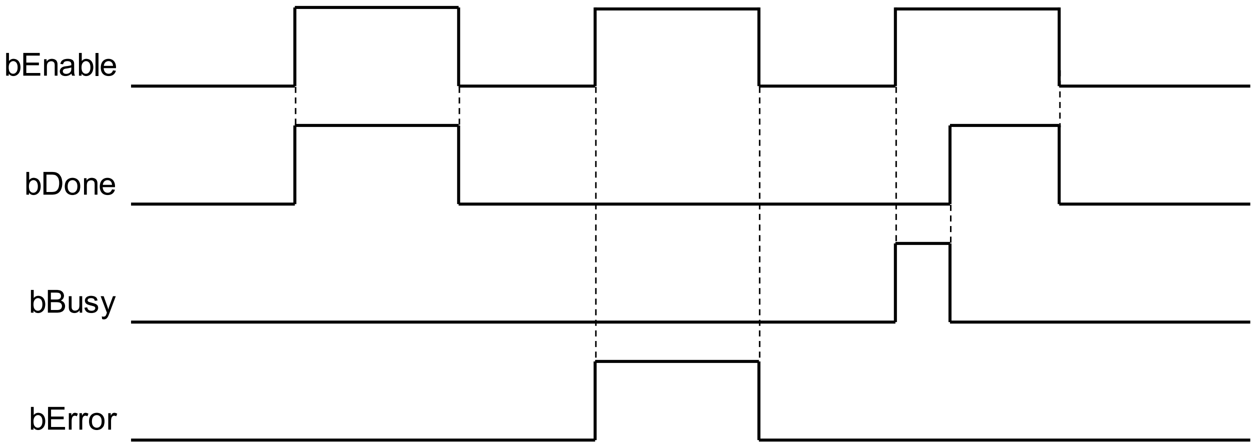
Name	Function	Data Type	Output Range (Default value)
bDone	When the pulse output is done.	BOOL	True/False (False)
bBusy	Shows function block is been executed	BOOL	True/False (False)

Name	Function	Data Type	Output Range (Default value)
aIntegerData	The device that stores the record value when Format is Integer	ARRAY[0~1999] OF INT	Please refer to chapter 15.2.1 of AS Module Manual (0)
aFloatData	The device that stores the recorded value when Format is Floating	ARRAY[0~1999] OF REAL	Please refer to chapter 15.2.1 of AS Module Manual (0)
iCurPointNo	Accumulated record points	INT	0 ~ 2000 (0)
bError	FB instruction error flag	BOOL	True/False (False)
ErrorCode	Error Code	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 bExecute shifts to False.
aIntegerData	• Keep updating until the record is complete.	• When bExecute shifts to False.
aFloatData	• Keep updating until the record is complete.	• When bExecute shifts to False.
iCurPointNo	• Keep updating until the record is complete.	• When bExecute shifts to False.
bError	• When the FB instruction is executed incorrectly or the input value is incorrect	• When bExecute shifts to False.
ErrorCode		

- **Timing Diagram**



- **Funtion**

1. This Function is only supported by the AX-3 series firmware version V1.0.3 or above.
2. This command is a dedicated command for analog input modules (AS04AD-A, AS06XA-A, AS02ADH-A). Function is to enable/disable the recording function, and send the recorded data from the module to the specified aIntegerData or aFloatData (Determined according to the Format setting in the module Parameters page).
3. The specifications of the Record Period and the number of points corresponding to each Model are described in the following table. For detailed usage, please refer to the iMode parameter description.

Model	Record Period	Number of Records
AS04AD-A	Fixed period mode:	Fixed to 500 records

AS06XA-A	The setting range is 1~100, and the unit time is fixed at 10ms	
AS02ADH-A	Fixed period (Fixed period) mode, trigger start types fixed period (Fixed period + Trigger Start) mode, trigger position designation (Fixed period + Trigger position Assign) recording mode: Setting range 1~32000, selectable time unit 20us/40us/80us (Note: The time unit is the sampling period set by HWCONFIG) Point Logging mode: One point is recorded every time the external input point is triggered, and there is no fixed period; the trigger timing of the external input point is set by HWCONFIG. Channel 1 triggers when X0.0 shift to True or False Channel 2 triggers when X0,1 shift to True or False	Can set from 1 to 2000 records

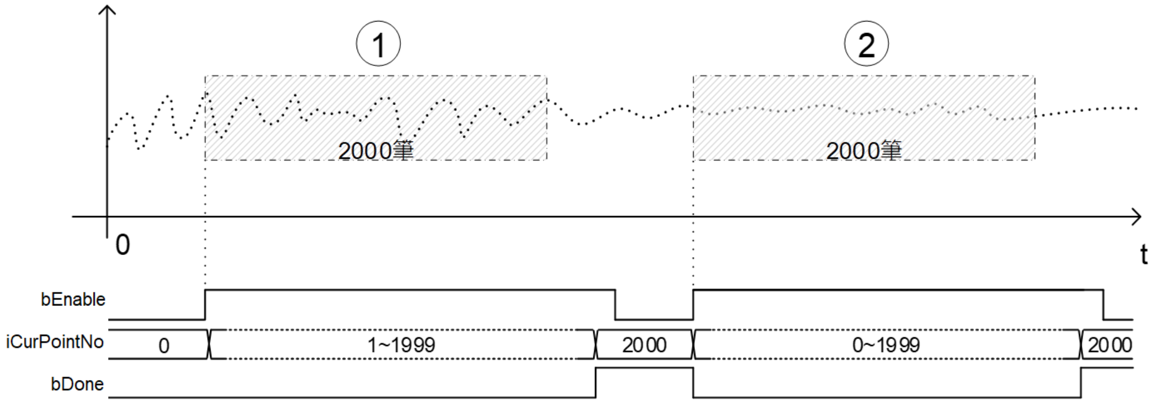
4. **byRemoteID** specifies the group number of the analog input module connected to the right side of the host or the right side of the remote module, the host number is 0, the number of the first remote module is 1, and so on, the maximum number of groups is 15.
5. **byLocalID** specifies the module number. The sequence number of the modules connected to the right side of the host starts from 0, second module is 1, and so on. Regardless of any type of modules, they must be counted, and the maximum number of modules is limited to 32 units.
6. **usiChannelNo** specifies the channel number. Channel 1 is numbered 1, channel 2 is numbered 2.
7. **iMode** is recording mode. Modes supported by each Model are shown in the table below.

Applicable Model	iMode Value	Name	Recording Mode Descroption
AS04AD-A AS06XA-A	0	Fixed Period mode	Execute fixed period records
AS02ADH-A	0	Fixed Period mode	Execute fixed period records
	1	Fixed period + Trigger Start mode	Waiting for the trigger signal of the external input point. When the trigger is received, the fixed period recording is executed immediately.
	2	Point Logging mode	No fixed period. One record is recorded every time the external input point is triggered.
	3	Fixed period + Trigger Start mode	The external trigger timing point can be specified to record the data recorded in the fixed period before / after.

(1) Fixed period (Fixed period) mode:
Set iMode=0, the command bEnable is turned on to execute the recording with the set Record Period. When the set total number of records is completed, the bDone flag will be automatically set to TRUE.

Example：

Set **iTotalPoints** = 2000



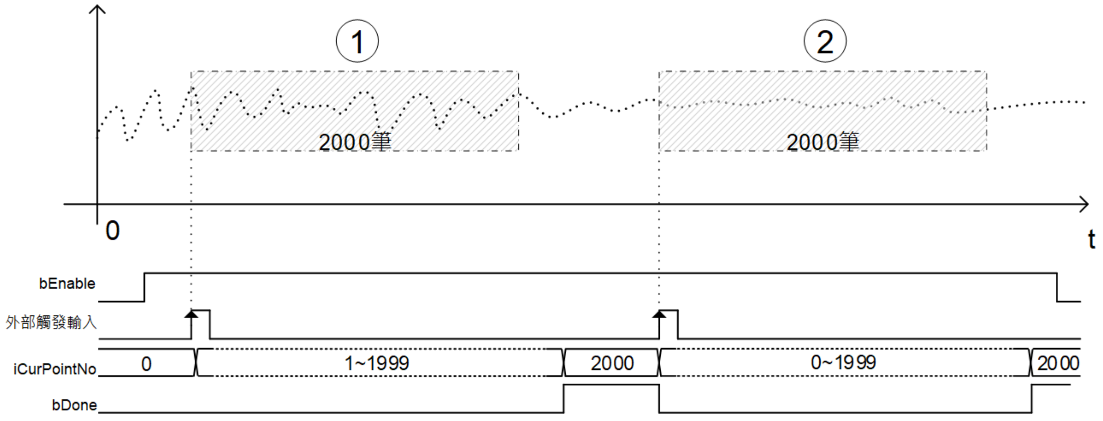
(2) Fixed period + Trigger Start mode：

Set iMode=1, turn on the command bEnable before starting the recording. When the external trigger input point is triggered, the recording will be executed with the set Record Period immediately, and the bDone flag will be automatically set to True when the recording is completed. Any operation on the external trigger input point will not affect the recording until the set total number of records is completed; however, when the record number has been completed and the bDone flag is True, the external input point can be re-triggered to start a new record. A round of records does not need to be closed and restarted by bEnable.

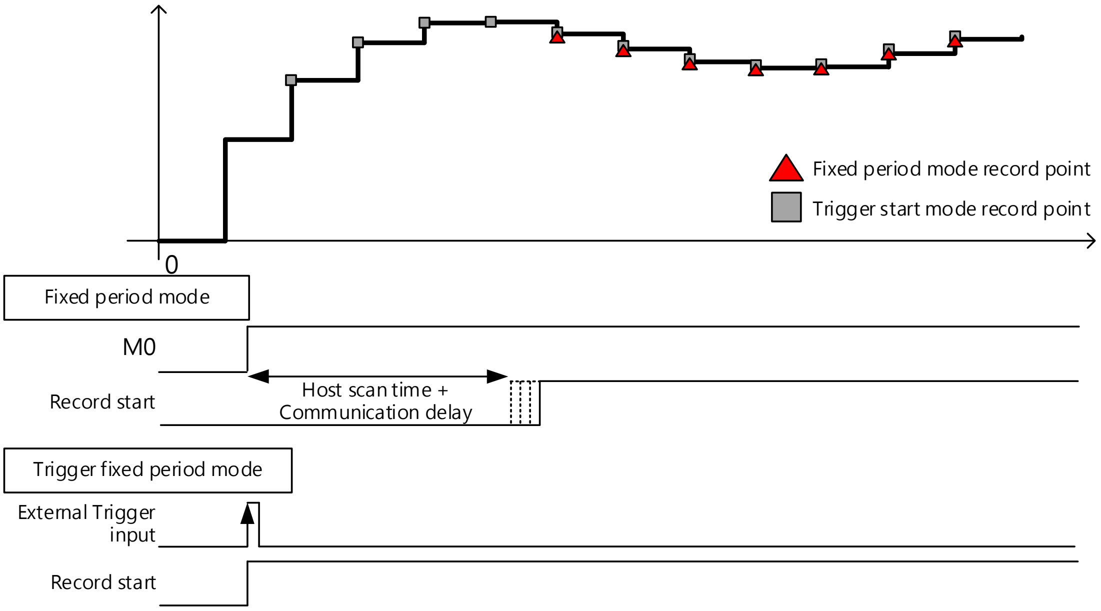
Record Channel	Corresponding to external trigger input signal source (Set the trigger timing of the external input point in the External Trigger Input of the module Parameters)
Channel 1	Triggers when X0.0 shift to True or False
Channel 2	Triggers when X0,1 shift to True or False

Example：

Set **iTotalPoints** = 2000,The trigger timing of the external input point is set as the rising edge trigger.



The fixed period + trigger start mode is similar to the fixed period mode, but the recording start timing of the fixed period mode will be affected by the host scan time and module communication time, causing delay, please refer to below picture. In the Fixed period mode, it is assumed that M0 is the device that controls the DADLOG command bEnable. When M0 is turned OFF->ON, the module does not start recording immediately, but with a small delay.



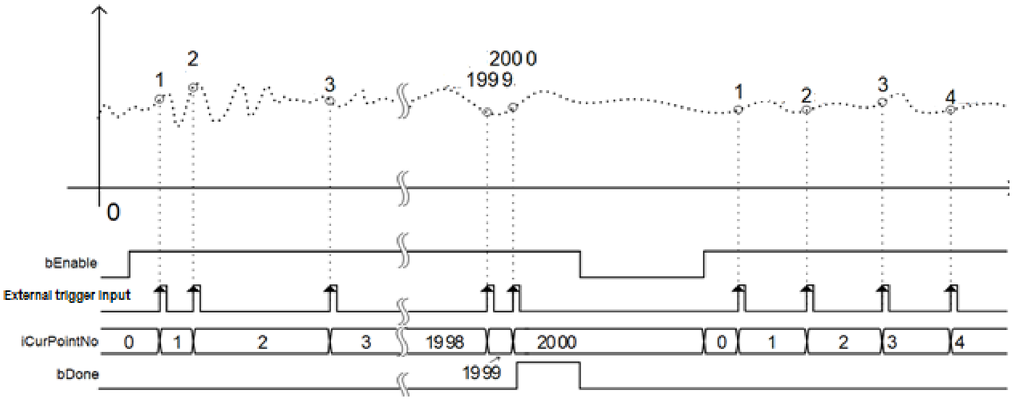
(3) Point Logging Mode :

Set **iMode**=2. Turn on the command bEnable before starting to record. Each time the external trigger input point is triggered, one record will be recorded. When the set total number of records is reached, the bDone flag will be automatically set to True; if wish to continue recording after the bDone flag is True, user must restart the command.

Record Channel	Corresponding to external trigger input signal source (Set the trigger timing of the external input point in the External Trigger Input of the module Parameters)
Channel 1	Triggers when X0.0 shift to True or False
Channel 2	Triggers when X0,1 shift to True or False

Example：

Set **iTotalPoints** = 2000, The trigger timing of the external input point is set as the rising edge.



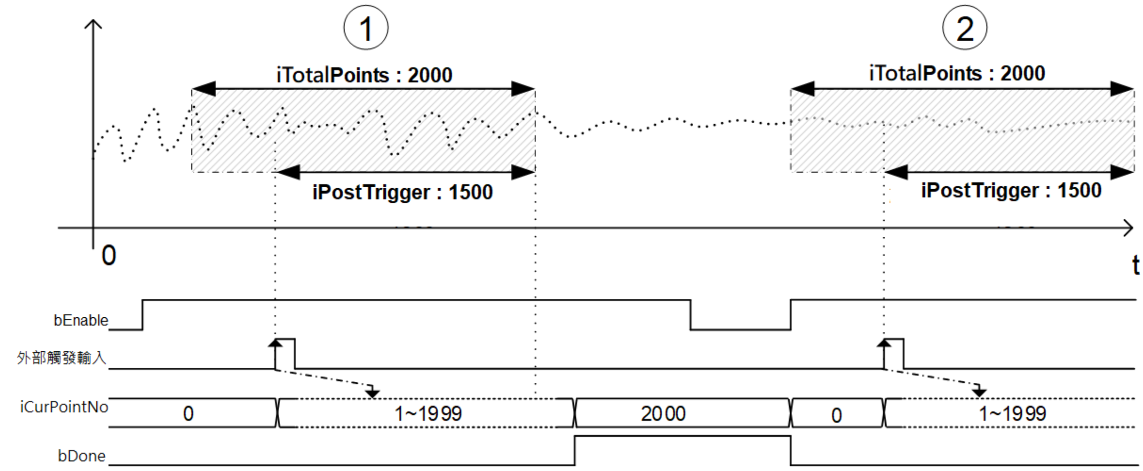
(4) Fixed period + Trigger position Assign Mode：

Set **iMode**=3, and set the iTotalPoints and iPostTrigger parameters. This mode is triggered by an external input point and records a specific number of strokes before/after the trigger according to the setting. After using the command bEnable to enable this recording mode, AS02ADH-A will start to wait for the external trigger input signal, and start sampling immediately after the signal is triggered. When the Number of Records reaches the set number of records, the bDone flag will be automatically set to True. The iCurPointNo value is 0 before the trigger. After the trigger, the module starts to send the record data before the trigger to the host, so the iCurPointNo value will gradually catch up with the accumulated record number.

Record Channel	Corresponding to external trigger input signal source (Set the trigger timing of the external input point in the External Trigger Input of the module Parameters)
Channel 1	Triggers when X0.0 shift to True or False
Channel 2	Triggers when X0,1 shift to True or False

Example：

Set **iMode**=3, **iTotalPoints** = 2000**, iPostTrigger** = 1500, means that the position of the 501st point (iTotalPoints - iPostTrigger) will be the first data recorded by the current external trigger.



1. **iPeriod** is set for Record Period, the description is as follows：

Model	iPeriod Range	iPeriod Unit
AS04AD-A AS06XA-A	1~100	Fixed as 10ms, cannot be changed.
AS02ADH-A	1~32000	The sampling cycle time must be set by AS02ADH-A Parameters, 20us, 40us and 80us can be selected. If "iMode=2: Point Logging Mode" is used, this iPeriod setting is invalid.

2. **iTotalPoints** is the total number of records. This parameter is only valid for AS02ADH-A, and can be set up to 2000 points; other models are fixed at 500 points regardless of the setting of this parameter.

Model	Total Recordm Setting Range
AS04AD-A AS06XA-A	The setting is invalid, fixed at 500.
AS02ADH-A	Can be set, 1~2000 records.

3. **iPostTrigger** is the points recorded after the trigger occurs. This parameter is used in combination with the total number of records iTotalPoints to record the data before and after the trigger. This parameter is only used in Fixed period + Trigger position Assign mode (iMode=3), and is invalid in other modes. Please be cautious that this value should not be greater than the total number of iTotalPoints. If it exceeds, it will automatically use the iTotalPoints value as Number of Records.

   Example : If set **iTotalPoints** = 100**, iPostTrigger** = 200. Since iPostTrigger is greater than the total number of records, iPostTrigger will be automatically regarded as 100, so only 100 records after the trigger will be recorded.

Model	**iPostTrigger** Setting Range
AS02ADH-A	0 ~2000 records (Must not be greater than the total number of iTotalPoints records)

Example：

Set **iTotalPoints**= 100, **iPostTrigger**= 700, the 1000 records include the first 300 records and the last 700 records triggered by the external trigger input point.

1. **aIntegerData** and **aFloatData** are specified arrays to store record values. (According to the Format setting in the parameters page of the module to determine which array to store in, the value in the other array will be cleared to 0).
2. **iCurPointNo** is to display the number of records that the module has sent back to the host. When the record is in progress, the **iCurPointNo** value may not be displayed in a consistent value due to the influence of the scanning cycle.
3. **bDone** is the flag of record completion. When **bDone** is OFF->ON, it means that all record values have been transferred to the specified **aIntegerData** or **aFloatData** array. The **bDone** flag will be automatically initialized to OFF when bEnable changes from OFF->ON.
4. If the command is closed, the specified channel will stop updating the values in the right half of the command.
5. When any error occurs during startup, the **bError** error flag will be set to ON. Please refer to the error code of **ErrorCode** for troubleshooting.

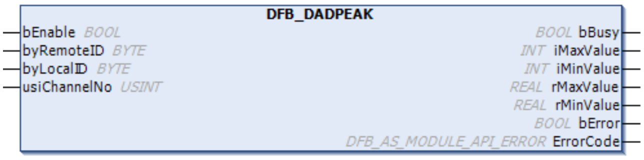
- **Library**
  - DL_ASModuleAPI_AX3.library

6.21 DFB_DADPEAK

- **Supported Products**
  - AX-308E, AX-364E, AX-300, AX-324N

DFB_DADPEAK：Analog input module peak record.

FB/FC	Instruction	Graphic Expression	ST LANGUAGE

FB	DFB_DADPEAK	 DFB_DADPEAK
		bEnable BOOL
		byRemoteID BYTE
		byLocalID BYTE
		usiChannelNo USINT
		BOOL bBusy
		INT iMaxValue
		INT iMinValue
		REAL rMaxValue
		REAL rMinValue
		BOOL bError
		DFB_AS_MODULE_API_ERROR ErrorCode
			DFB_DADPEAK(
			bEnable:= ,
			byRemoteID:= ,
			byLocalID:= ,
			usiChannelNo:= ,
			bBusy=> ,
			iMaxValue=> ,
			iMinValue=> ,
			rMaxValue=> ,
			rMinValue=> ,
			bError=> ,
			ErrorCode=>);

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bEnable	Execute function block	BOOL	True/False（False）
byRemoteID*	The CPU or remote module ID	BYTE	0: CPU 1~15: Remote module （0）
byLocalID	Expansion module ID	BYTE	0 ~ 31（0）
usiChannelNo	Specify channel number	USINT	1 ~ 2（1）

*Note: Currently only support mode 0.

- **Output**

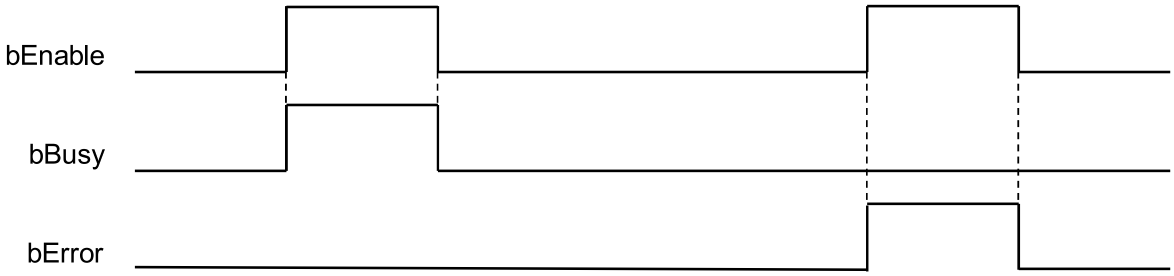
Name	Function	Data Type	Output Range (Default value)
bBusy	Indicates that the function block is being executed.	BOOL	True/False（False）
iMaxValue	Maximum value when Format is Integer.	INT	Please refer to AS Module Manual chapter 15.2.1 of（0）
iMinValue	Minimum value when Format is Integer.	INT	Please refer to AS Module Manual chapter 15.2.1 of（0）
rMaxValue	Maximum value when Format is Floating.	REAL	Please refer to AS Module Manual chapter 15.2.1 of（0）
rMinValue	Minimum value when Format is Floating.	REAL	Please refer to AS Module Manual chapter 15.2.1 of（0）
bError	FB instruction error flag.	BOOL	True/False（False）

Name	Function	Data Type	Output Range (Default value)
ErrorCode	Error Code	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
bBusy	● When the execution of FB starts.	● When bExecute shifts to False.
iMaxValue	● Continuous update after bEnable	● When bExecute shifts to False.
iMinValue	● Continuous update after bEnable	● When bExecute shifts to False.
rMaxValue	● Continuous update after bEnable	● When bExecute shifts to False.
rMinValue	● Continuous update after bEnable	● When bExecute shifts to False.
bError	● When the FB instruction is executed incorrectly or the input value is incorrect	● When bExecute shifts to False.
ErrorCode		

- **Timing Diagram**



- **Funtion**

1. This Function is only supported by the AX-3 series firmware version V1.0.3 or above.
2. The DADPEAK command is a dedicated command for analog input modules (AS04AD-A, AS08AD-B, AS08AD-C, AS06XA-A, AS02ADH-A), and its Function is to enable/disable the module to record the peak value.
3. **byRemoteID** specifies the group number of the analog input module connected to the right side of the host or the right side of the remote module, the host number is 0, the number of the first remote module is 1, and so on, the maximum number of groups is 15.
4. **byLocalID** specifies the module number, the sequence number of the modules connected to the right side of the host, the number of the first module is 0, the number of the second module is 1, and so on, regardless of any type of modules, they must be counted, and the maximum number of modules is limited. for 32 units.
5. **usiChannelNo** specifies the number of the counter channel to be controlled, the number of Channel 1 is 1, and the number of Channel 2 is 2.
6. According to the Format setting in the module Parameters page, determine whether the maximum and minimum values are placed in **iMaxValue** and **iMinValue** or in **rMaxValue** and **rMinValue**. The maximum and minimum values of another Data Type will be cleared to 0.
7. The following describes the maximum and minimum values of INT or REAL type represented by **MaxValue** and **MinValue**.
8. **MaxValue** and **MinValue** are the maximum and minimum values respectively. When **bEnable** is changed from OFF->ON, **MaxValue** and **MinValue** will be initialized to the latest measurement value, and then the peak value recording Function will start, and the channel will continue to detect the maximum and minimum values. value and update to **MaxValue** and **MinValue**.
9. If **bEnable** changes from ON to OFF, it means that the peak value recording function is turned off. At this time, the content values of **MaxValue** and **MinValue** will remain unchanged and will not be updated.

10. When any error occurs during the startup process, the **bError** flag will be set to ON. Please refer to the error code of **ErrorCode** for troubleshooting.

- **Library**

  - DL_ASModuleAPI_AX3.library

6.22 Error Codes and Troubleshooting

Description	Cause of Error	Corrective Action
DFB_FROM_ERR_PARAMETER	Enter parameter error	Confirm if the input parameters are correct.
DFB_FROM_ERR_COMMUNICATION	CAN bus communication error	Confirm the error record.
DFB_FROM_ERR_CRADDR	CR address error	Check if the CR address is correct.
DFB_TO_ERR_PARAMETER	Enter parameter error	Confirm if the input parameters are correct.
DFB_TO_ERR_COMMUNICATION	CAN bus communication error	Confirm the error record.
DFB_TO_ERR_CRADDR	CR address error	Check if the CR address is correct.
DFB_DLCCAL_ERR_NOT_SUPPORT _MODULE	byRemoteID and byLocalID correspond to the module that is not the AS02LC module.	Confirm byRemoteID, byLocalID, and the corresponding module.
DFB_DLCCAL_ERR_INVALID_GROUP	byRemoteID input error	Confirm if byRemoteID input value is correct.
DFB_DLCCAL_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DLCCAL_ERR_INVALID_CHNO	usiChannelNo input error	Confirm if usiChannelNo input value is correct.
DFB_DLCCAL_ERR_INVALID_TWEIGHT	aTWeight input error	Confirm if aTWeight input value is correct.
DFB_DLCCAL_ERR_INVALID_TPOINT	iTPoint input error	Confirm if iTPoint input value is correct.
DFB_DLCCAL_ERR_MODULE_ REPORTS_AN_ERROR	Module on the right reports error.	Check the error code reported in the Diagnosis Message in the Status of the module page, and check this error code in the AS Series module manual.
DFB_DLCWEI_ERR_NOT_SUPPORT_ MODULE	byRemoteID and byLocalID correspond to the module that is not the AS02LC module.	Confirm byRemoteID, byLocalID, and the corresponding module.
DFB_DLCWEI_ERR_INVALID_GROUP	byRemoteID input error	Confirm if byRemoteID input value is correct.
DFB_DLCWEI_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DLCWEI_ERR_INVALID_CHNO	usiChannelNo input error	Confirm if usiChannelNo input value is correct.
DFB_DLCWEI_ERR_INVALID_STABLE	rStable input error	Confirm if rStable input value is correct.
DFB_DLCWEI_ERR_MODULE_REPORTS _AN_ERROR	Module on the right reports error.	Check the error code reported in the Diagnosis Message in the Status of the module page, and check this error code in the AS Series module manual.
DFB_DPUCONF_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DPUCONF_ERR_NOT_SUPPORT _MODULE	byLocalID and iAxis correspond to the module that is not the PU module.	Confirm byLocalID, iAxis, and the corresponding module.

Description	Cause of Error	Corrective Action
DFB_DPUCONF_ERR_INVALID_AXIS	iAxis input error	Confirm if iAxis input value is correct.
DFB_DPUCONF_ERR_INVALID_MODE	iMode input error	Confirm if iMode input value is correct.
DFB_DPUCONF_ERR_INVALID_SSPEED	iStartSpeed input error	Confirm if iStartSpeed input value is correct.
DFB_DPUCONF_ERR_INVALID_ATIME	iAccTime input error	Confirm if iAccTime input value is correct.
DFB_DPUCONF_ERR_INVALID_DTIME	iDecTime input error	Confirm if iDecTime input value is correct.
DFB_DPUCONF_ERR_INVALID_MSPEED	diMaxSpeed input error	Confirm if diMaxSpeed input value is correct.
DFB_DPUCONF_ERR_INVALID_ZNO	iZ_no input error	Confirm if iZ_no input value is correct.
DFB_DPUCONF_ERR_INVALID_OFFSET	iOffset input error	Confirm if iOffset input value is correct.
DFB_PUSTAT_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_PUSTAT_ERR_NOT_SUPPORT _MODULE	byLocalID and iAxis correspond to the module that is not the PU module.	Confirm byLocalID, iAxis, and the corresponding module.
DFB_PUSTAT_ERR_INVALID_AXIS	iAxis input error	Confirm if iAxis input value is correct.
DFB_DPUPLS_DPUDRI_DPUDRA_ERR _INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DPUPLS_DPUDRI_DPUDRA_ERR _NOT_SUPPORT_MODULE	byLocalID and iAxis correspond to the module that is not the PU module.	Confirm byLocalID, iAxis, and the corresponding module.
DFB_DPUPLS_DPUDRI_DPUDRA_ERR_ INVALID_AXIS	iAxis input error	Confirm if iAxis input value is correct.
DFB_DPUPLS_DPUDRI_DPUDRA_ERR_ INVALID_TARSPEED	diTarSpeed input error	Confirm if diTarSpeed input value is correct.
DFB_DPUPLS_DPUDRI_DPUDRA_ERR_ POSITIVELIMIT_EXCEEDED	Exceed the positive limit position setting	Close the function block, set to run to the opposite direction and restart.
DFB_DPUPLS_DPUDRI_DPUDRA_ERR_ NEGATIVELIMIT_EXCEEDED	Exceed the negative limit position setting	Close the function block, set to run to the opposite direction and restart.
DFB_DPUZRN_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DPUZRN_ERR_NOT_SUPPORT_MODULE	byLocalID and iAxis correspond to the module that is not the PU module.	Confirm byLocalID, iAxis, and the corresponding module.
DFB_DPUZRN_ERR_INVALID_AXIS	iAxis input error	Confirm if iAxis input value is correct.
DFB_DPUZRN_ERR_INVALID_MODE	iMode input error	Confirm if iMode input value is correct.
DFB_DPUZRN_ERR_INVALID_TARSPEED	diTarSpeed input error	Confirm if diTarSpeed input value is correct.
DFB_DPUZRN_ERR_INVALID_JOGSPEED	iJogSpeed input error	Confirm if iJogSpeed input value is correct.
DFB_DPUJOG_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DPUJOG_ERR_NOT_SUPPORT _MODULE	byLocalID and iAxis correspond to the module that is not the PU module.	Confirm byLocalID, iAxis, and the corresponding module.
DFB_DPUJOG_ERR_INVALID_AXIS	iAxis input error	Confirm if iAxis input value is correct.

Description	Cause of Error	Corrective Action
DFB_DPUJOG_ERR_INVALID_JOGSPEED	diJogSpeed input error	Confirm if diJogSpeed input value is correct.
DFB_DPUJOG_ERR_POSITIVELIMIT _EXCEEDED	Exceed the positive limit position setting	Close the function block, set to run to the opposite direction and restart.
DFB_DPUJOG_ERR_NEGATIVELIMI T_EXCEEDED	Exceed the negative limit position setting	Close the function block, set to run to the opposite direction and restart.
DFB_DPUMPG_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DPUMPG_ERR_NOT_SUPPORT _MODULE	byLocalID and iAxis correspond to the module that is not the PU module.	Confirm byLocalID, iAxis, and the corresponding module.
DFB_DPUMPG_ERR_BEUSED_BY _DPUCNT	The pin is currently being used by the DPUCNT function block.	After the DPUCNT function block is closed, re-trigger DPUMPG.
DFB_DPUMPG_ERR_INVALID_AXIS	iAxis input error	Confirm if iAxis input value is correct.
DFB_DPUCNT_ERR_INVALID_MODULE	byLocalID input error	Confirm if byLocalID input value is correct.
DFB_DPUCNT_ERR_NOT_SUPPORT _MODULE	The module corresponding to byLocalID is not the PU module.	Confirm byLocalID and the corresponding module.
DFB_DPUCNT_ERR_BEUSED_BY_OTHER	The pin is currently being used by other function blocks.	After other function blocks are closed, re-trigger DPUCNT.
DFB_DMPID_ERR_NOT_SUPPORT _MODULE	Module does not support this instruction.	Confirm if the module is the TC module.
DFB_DMPID_ERR_INVALID_GROUP _OR_MODULE_ID	Group number or module number setting error	Confirm if the input pin is in the correct range.
DFB_DMPID_ERR_COMMUNICAION	No response from module, communication timeout	Check if the module in the device area is properly connected.
DFB_DMPID_ERR_NO_CHANNEL	Channel setting error	Confirm if the input parameter is correct.
DFB_DMPID_ERR_CHANNEL_IS_ EXECUTING_PID	The channel is executing the PID function, and repeating designation is not allowed.	Confirm if the PID function is open and is been executed.
DFB_DHCCNT_ERR_NOT_SUPPORT _MODULE	Module does not support this instruction.	Confirm if the module is the HC module.
DFB_DHCCNT_ERR_INVALID_INPUT_ VALUE_TO_HC_MODULE	When updating parameter to the module, module response error.	Confirm if the input pin is in the correct range.
DFB_DHCCNT_ERR_COMMUNICAION	No response from module, communication timeout	Check if the module in the device area is properly connected.
DFB_DHCCNT_ERR_HC_MODULE_ CONFIG_ERROR	Module configuration setting error	Confirm if the setting parameter of the input device is correct.
DFB_DHCCNT_ERR_NO_CHANNEL	HC module does not have this counter channel.	Confirm if the used channel number is the module support number.
DFB_DHCCNT_ERR_INTERFACE_OF_ CHANNEL_IS_DISABLE	Input interface is not selected, instruction operation is not allowed.	Confirm if the channel input interface enable the corresponding functions.
DFB_DHCCNT_ERR_INVALID_ACTION _VALUE	usiAction value is invalid.	Confirm if the input pin usiAction is correct.
DFB_DHCCNT_ERR_CHANNEL_IS _EXECUTING_CNT	The module channel is executing the counting function, and repeating designation is not allowed.	Confirm if the HC counting function is enabled and is being executed.

Description	Cause of Error	Corrective Action
DFB_DHCCAP_ERR_NOT_SUPPORT _MODULE	Module does not support this instruction.	Confirm if the module is the HC module.
DFB_DHCCAP_ERR_INVALID_INPUT _VALUE_TO_HC_MODULE	When updating parameters to the module, module response error.	Confirm if the input pin is in the correct range.
DFB_DHCCAP_ERR_COMMUNICAION	No response from module, communication timeout	Check if the module in the device area is properly connected.
DFB_DHCCAP_ERR_HC_MODULE _CONFIG_ERROR	Module configuration setting error	Confirm if the setting parameter of the input device is correct.
DFB_DHCCAP_ERR_NO_CHANNEL	HC module does not have this counter channel.	Confirm if the used channel number is the module support number.
DFB_DHCCAP_ERR_INTERFACE_OF _CHANNEL_IS_DISABLE	Input interface is not selected, instruction operation is not allowed.	Confirm if the channel input interface enable the corresponding functions.
DFB_DHCCAP_ERR_INVALID_TRGSEL_VALUE	byTrgSel value is invalid.	Confirm if input pin byTrgSel is correct.
DFB_DHCCAP_ERR_CHANNEL_IS_EXECUTING_CAP	The module channel is executing the capture function, and repeating designation is not allowed.	Confirm if the HC capture function is enabled and is being executed.
DFB_HCDO_ERR_INVALID_MODULE	byLocalID input error	Confirm the input value is correct.
DFB_HCDO_ERR_NOT_SUPPORT_MODULE	byLocalID corresponds to a module that is not the HC module.	Confirm the module to which the function block corresponds.
DFB_HCDO_ERR_BEUSED_BY_OTHER_DFB	HC module is current being used by other function blocks.	After other function blocks are closed, re-trigger HCDO.
DFB_DHCCMP_ERR_INVALID_MODULE	byLocalID input error	Confirm the input value is correct.
DFB_DHCCMP_ERR_NOT_SUPPORT_MODULE	byLocalID corresponds to a module that is not the HC module.	Confirm the module to which the function block corresponds.
DFB_DHCCMP_ERR_INVALID_CHNO	usiChannelNo input error	Confirm the input value is correct.
DFB_DHCCMP_ERR_INVALID_ACTION1	iActionValue1 input error	Confirm the input value is correct.
DFB_DHCCMP_ERR_INVALID_YNO1	iY_OutputNo1 input error	Confirm the input value is correct.
DFB_DHCCMP_ERR_INVALID_ACTION2	iActionValue2 input error	Confirm the input value is correct.
DFB_DHCCMP_ERR_INVALID_YNO2	iY_OutputNo2 input error	Confirm the input value is correct.
DFB_DHCCMP_ERR_INVALID_INTERRUPTNO1	iInterruptNo1 input error	Confirm the input value is correct.
DFB_DHCCMP_ERR_INVALID_INTERRUPTNO2	iInterruptNo2 input error	Confirm the input value is correct.
DFB_DHCCMPT_ERR_INVALID_MODULE	byLocalID input error	Confirm the input value is correct.
DFB_DHCCMPT_ERR_NOT_SUPPORT_MODULE	byLocalID corresponds to a module that is not the HC module.	Confirm the module to which the function block corresponds.
DFB_DHCCMPT_ERR_INVALID_CHNO	usiChannelNo input error	Confirm the input value is correct.
DFB_DHCCMPT_ERR_INVALID_COMPARELENGTH	iCompareLength input error	Confirm the input value is correct.
DFB_DHCCMPT_ERR_INVALID_COMPARETABLE	aCompareValue input error	Confirm the input value is correct.

Description	Cause of Error	Corrective Action
DFB_DHCCMPT_ERR_INVALID_ACTIONTABLE	aAction input error	Confirm the input value is correct.
DFB_DHCCMPT_ERR_INVALID_YNOTABLE	aY_OutputNo input error	Confirm the input value is correct.
DFB_DHCCMPT_ERR_INVALID_INTERRUPTNOTABLE	aInterruptNo input error	Confirm the input value is correct.
DFB_DHCMEAS_ERR_INVALID_MODULE	byLocalID input error	Confirm the input value is correct.
DFB_DHCMEAS_ERR_NOT_SUPPORT_MODULE	byLocalID corresponds to a module that is not the HC module.	Confirm the module to which the function block corresponds.
DFB_DHCMEAS_ERR_INVALID_CHNO	usiChannelNo input error	Confirm the input value is correct.
DFB_DHCMEAS_ERR_INVALID_PULSEPERREV	udiPulsePerRev input error	Confirm the input value is correct.
DFB_DHCMEAS_ERR_INVALID_SAMPLINGTIME	iSamplingTime input error	Confirm the input value is correct.
DFB_DHCMEAS_ERR_INVALID_AVERAGETIMES	iMovingAvgWindow input error	Confirm the input value is correct.
DFB_DADLOG_ERR_INVALID_GROUP	byRemoteID input error	Please confirm whether the input value of byRemoteID is correct
DFB_DADLOG_ERR_NOT_SUPPORT_MODULE	The module corresponding to byLocalID is not an AD module	Please confirm the module corresponding to the function block
DFB_DADLOG_ERR_INVALID_MODULE	byLocalID input error	Please confirm whether the input value is correct
DFB_DADLOG_ERR_INVALID_POINTS	iTotalPoints typo	Please confirm whether the input value is correct
DFB_DADLOG_ERR_INVALID_CHNO	usiChannelNo input error	Please confirm whether the input value is correct
DFB_DADLOG_ERR_INVALID_MODE	iMode input error	Please confirm whether the input value is correct
DFB_DADLOG_ERR_INVALID_PERIOD	iPeriod input error	Please confirm whether the input value is correct
DFB_DADPEAK_ERR_INVALID_GROUP	byRemoteID input error	Please confirm whether the input value of byRemoteID is correct
DFB_DADPEAK_ERR_NOT_SUPPORT_MODULE	The module corresponding to byLocalID is not an AD module	Please confirm the module corresponding to the function block
DFB_DADPEAK_ERR_INVALID_MODULE	byLocalID input error	Please confirm whether the input value is correct
DFB_DADPEAK_ERR_INVALID_CHNO	usiChannelNo input error	Please confirm whether the input value is correct

Chapter 7 Modbus Communication Instructions

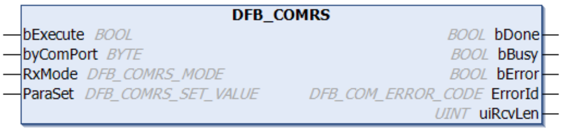
7.1 DFB_COMRS

- **Supported Products**

  - AX Series(Without supporting AX-8)

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 BOOL
		byComPort BYTE
		RxMode DFB_COMRS_MODE
		ParaSet DFB_COMRS_SET_VALUE
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DFB_COM_ERROR_CODE ErrorId
		UINT uiRcvLen
			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.

Name	Description
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)
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	UINT	1 ~ 1,000(100)

Name	Function	Data Type	Setting Value (Default value)
	stored. (Unit: Byte)		
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)
tTimeout	Communication timeout	TIME	T#0ms ~ T#49d17h2m47s295ms(T#100ms) T#0ms: No timeout

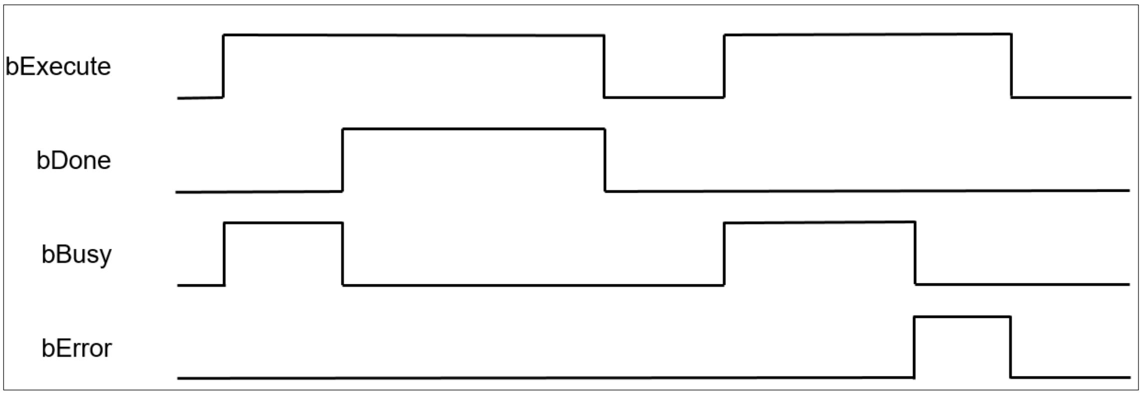
- **Output**

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**

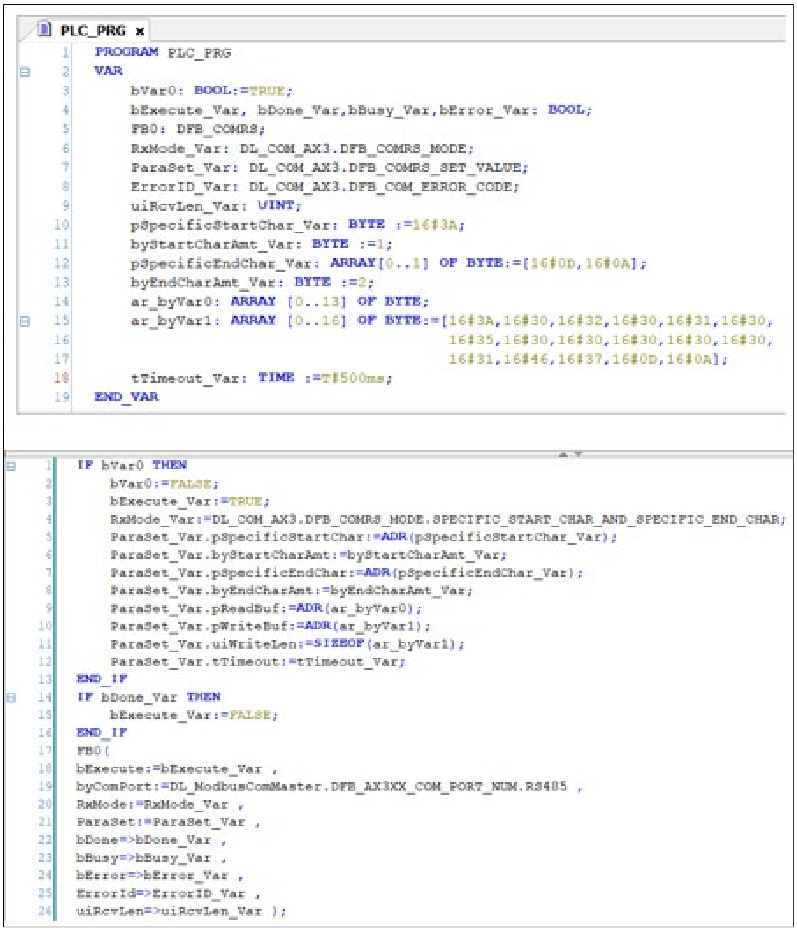


- **Function**

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**

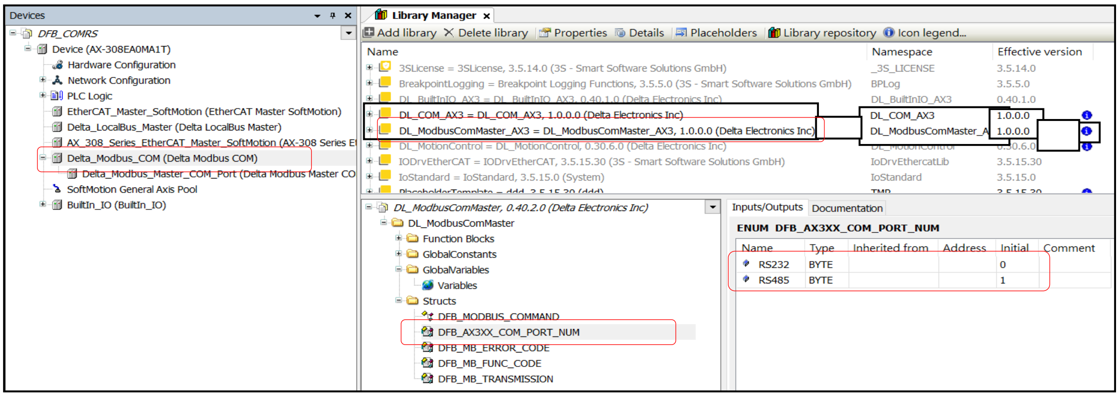
This example used DFB_COMRS to send COM communication data.



```
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.



Devices

- DFB_COMRS
  - Device (AX-308EA0MA1T)
    - Hardware Configuration
    - Network Configuration
    - PLC Logic
    - EtherCAT_Master_SoftMotion (EtherCAT Master SoftMotion)
    - Delta_LocalBus_Master (Delta LocalBus Master)
    - AX_308_Series_EtherCAT_Master_SoftMotion (AX-308 Series Et
    - Delta_Modbus_COM (Delta Modbus COM)
      - Delta_Modbus_Master_COM_Port (Delta Modbus Master CO
    - SoftMotion General Axis Pool
    - BuiltIn_IO (BuiltIn_IO)

Library Manager

Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.14.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
DL_BuiltInIO_AX3 = DL_BuiltInIO_AX3, 0.40.1.0 (Delta Electronics Inc)	DL_BuiltInIO_AX3	0.40.1.0
DL_COM_AX3 = DL_COM_AX3, 1.0.0.0 (Delta Electronics Inc)	DL_COM_AX3	1.0.0.0
DL_ModbusComMaster_AX3 = DL_ModbusComMaster_AX3, 1.0.0.0 (Delta Electronics Inc)	DL_ModbusComMaster_A	1.0.0.0
DL_MotionControl = DL_MotionControl, 0.30.6.0 (Delta Electronics Inc)	DL_MotionControl	0.30.6.0
IODrvEtherCAT = IODrvEtherCAT, 3.5.15.30 (3S - Smart Software Solutions GmbH)	IoDrvEthercatLib	3.5.15.30
IoStandard = IoStandard, 3.5.15.0 (System)	IoStandard	3.5.15.0
PlaceholderTemplate = ddd, 3.5.15.30 (ddd)	TMP	3.5.15.30

- DL_ModbusComMaster, 0.40.2.0 (Delta Electronics Inc)
  - DL_ModbusComMaster
    - Function Blocks
    - GlobalConstants
    - GlobalVariables
      - Variables
    - Structs
      - DFB_MODBUS_COMMAND
      - DFB_AX3XX_COM_PORT_NUM
      - DFB_MB_ERROR_CODE
      - DFB_MB_FUNC_CODE
      - DFB_MB_TRANSMISSION

Inputs/Outputs Documentation

ENUM DFB_AX3XX_COM_PORT_NUM

Name	Type	Inherited from	Address	Initial	Comment
RS232	BYTE			0	
RS485	BYTE			1	

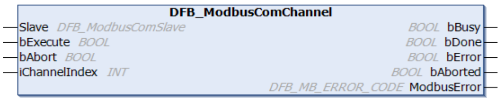
- **Library**
  - DL_COM_AX3.library

7.2 DFB_ModbusComChannel

- **Supported Products**
  - AX-308E, AX-364E, AX-300, AX-324N

DFB_ModbusComChannel: Modbus Slave COM Port Channel control instruction.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_
	ModbusComChannel
		 DFB_ModbusComChannel
		Slave DFB_ModbusComSlave
		bExecute BOOL
		bAbort BOOL
		iChannelIndex INT
		BOOL bBusy
		BOOL bDone
		BOOL bError
		BOOL bAborted
		DFB_MB_ERROR_CODE ModbusError
			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)

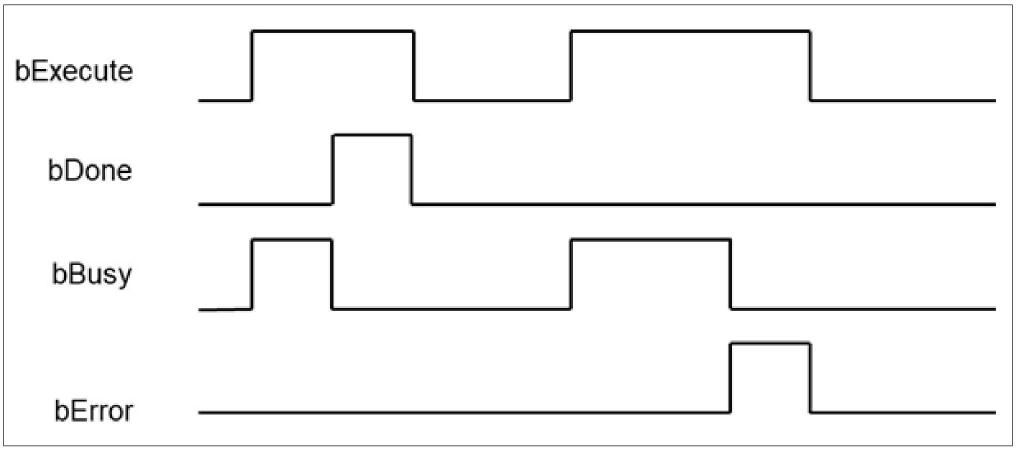
- **Output**

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**

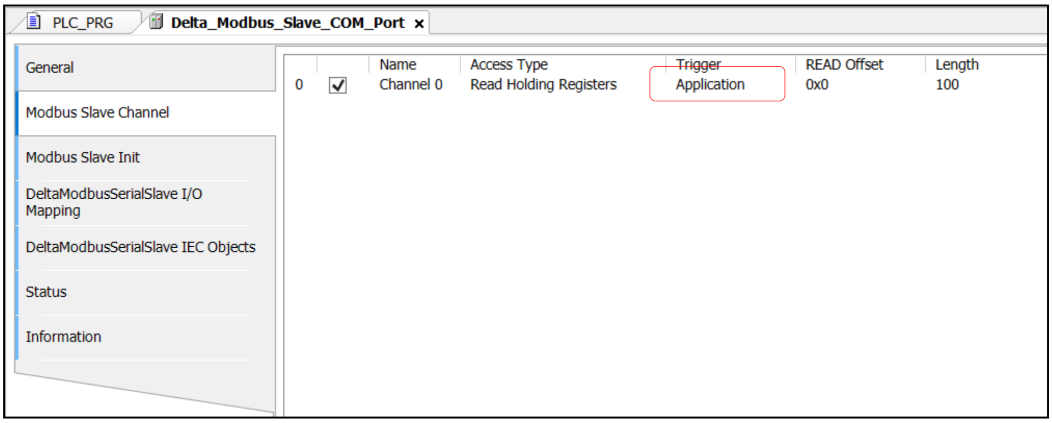
Name	Timing for shifting to True	Timing for shifting to False
bDone	● When the execution of FB is completed.	● When bExecute shifts to False.
bBusy	● When the execution of FB starts.	● When the execution of FB is completed. ● bExecute shifts to False and the execution of FB is completed.
bError	● When 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.


PLC_PRG
Delta_Modbus_Slave_COM_Port
General
Modbus Slave Channel
Modbus Slave Init
DeltaModbusSerialSlave I/O Mapping
DeltaModbusSerialSlave IEC Objects
Status
Information

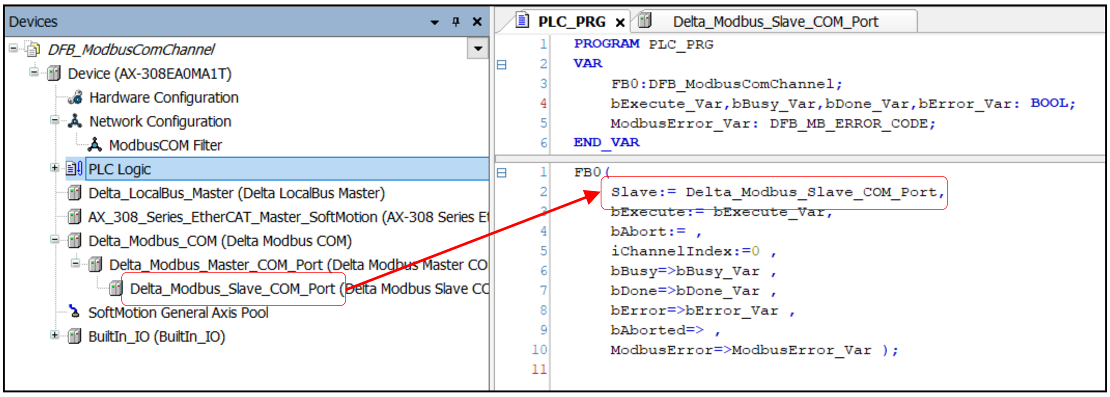
		Name	Access Type	Trigger	READ Offset	Length
0	☑	Channel 0	Read Holding Registers	Application	0x0	100

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.


Devices
DFB_ModbusComChannel
Device (AX-308EA0MA1T)
Hardware Configuration
Network Configuration
ModbusCOM Filter
PLC Logic
Delta_LocalBus_Master (Delta LocalBus Master)
AX_308_Series_EtherCAT_Master_SoftMotion (AX-308 Series E
Delta_Modbus_COM (Delta Modbus COM)
Delta_Modbus_Master_COM_Port (Delta Modbus Master CO
Delta_Modbus_Slave_COM_Port (Delta Modbus Slave CC
SoftMotion General Axis Pool
BuiltIn_IO (BuiltIn_IO)
PLC_PRG
Delta_Modbus_Slave_COM_Port

```
PROGRAM PLC_PRG
VAR
    FB0:DFB_ModbusComChannel;
    bExecute_Var,bBusy_Var,bDone_Var,bError_Var: BOOL;
    ModbusError_Var: DFB_MB_ERROR_CODE;
END_VAR
```

```
FB0(
    Slave:= Delta_Modbus_Slave_COM_Port,
    bExecute:= bExecute_Var,
    bAbort:= ,
    iChannelIndex:=0 ,
    bBusy=>bBusy_Var ,
    bDone=>bDone_Var ,
    bError=>bError_Var ,
    bAborted=> ,
    ModbusError=>ModbusError_Var );
```

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

- **Library**

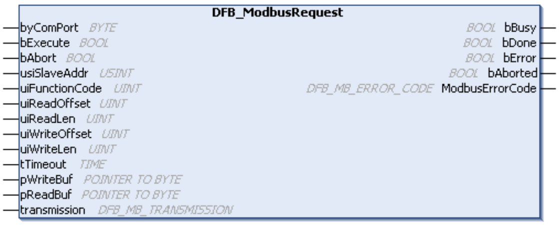
  - DL_ModbusComMaster_AX3.library

7.3 DFB_ModbusRequest

- **Supported Products**
  - AX-308E, AX-364E, AX-300, AX-324N

DFB_ModbusRequest: Modbus communication commands.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_ ModbusRequest	 DFB_ModbusRequest
		byComPort BYTE
		bExecute BOOL
		bAbort BOOL
		usiSlaveAddr USINT
		uiFunctionCode UINT
		uiReadOffset UINT
		uiReadLen UINT
		uiWriteOffset UINT
		uiWriteLen UINT
		tTimeout TIME
		pWriteBuf POINTER TO BYTE
		pReadBuf POINTER TO BYTE
		transmission DFB_MB_TRANSMISSION
		BOOL bBusy
		BOOL bDone
		BOOL bError
		BOOL bAborted
		DFB_MB_ERROR_CODE ModbusErrorCode
			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)

Name	Function	Data Type	Setting Value (Default value)
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*5	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.
4. If uiReadLen is set to 0, the Modbus read command will not be sent.
5. If uiWriteLen is set to 0, the Modbus write command will not be sent.

- **Output**

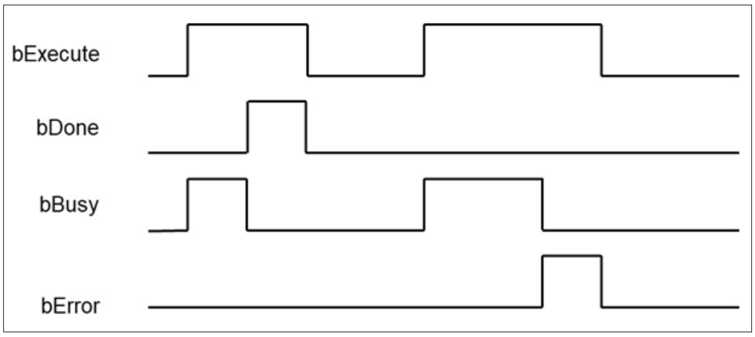
Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False(False)
bDone	The execution of FB is completed.	BOOL	True/False(False)
bError	True if an error occurs.	BOOL	True/False(False)
bAborted	No function	BOOL	--
ModbusError	Error code	DFB_MB_ERROR_CODE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

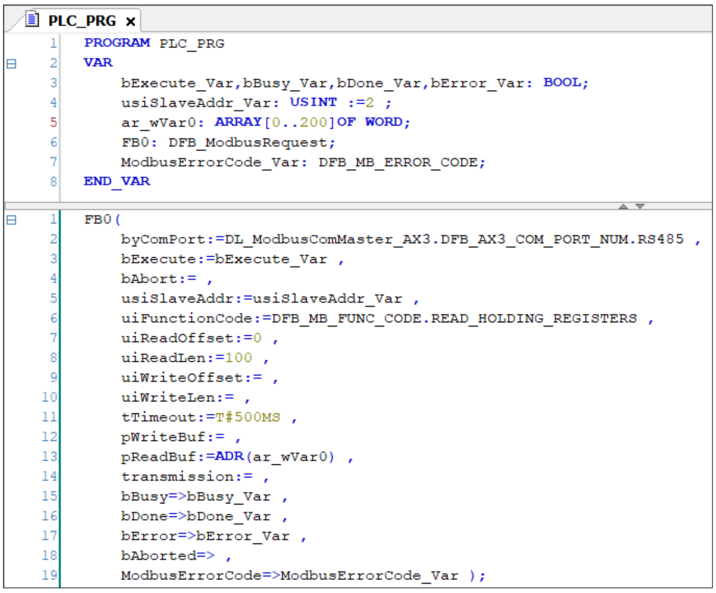


- **Function**

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.



```
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 );
```

- **Library**

    - DL_ModbusComMaster_AX3.library

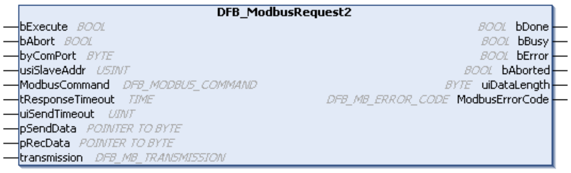
7.4 DFB_ModbusRequest2

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

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	--

Name	Function	Data Type	Setting Value (Default value)
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)

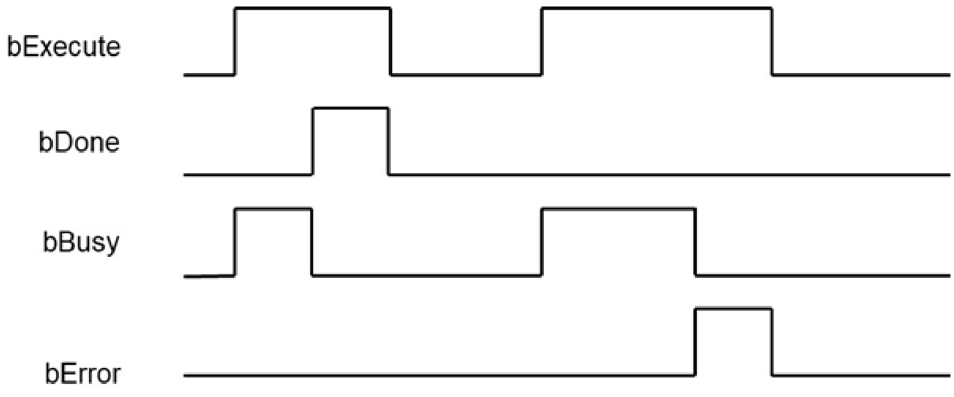
- **Output**

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**

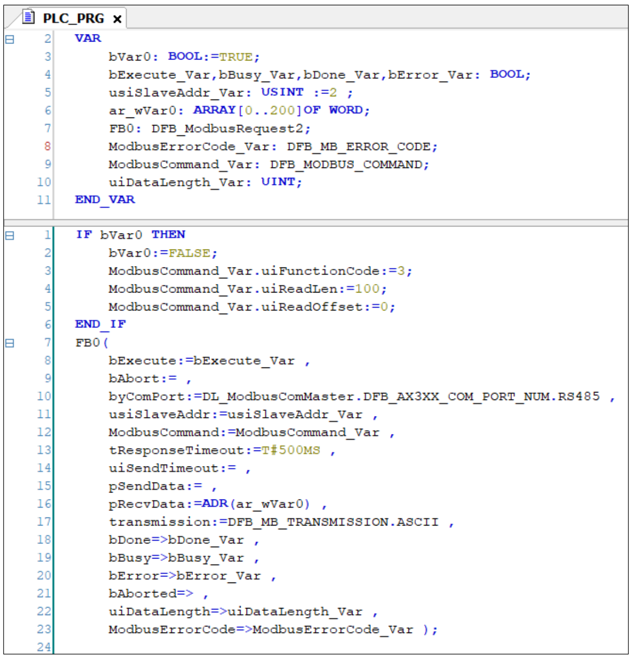


- **Function**

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.



```
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 );
```

- **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_	Illegal data values responded by	Please check if the slave wires function normally as well as the

Description	Cause of Error	Corrective Action
VALUE	slave.	proper wiring.
DFB_SLAVE_DEVICE_ FAILURE	Slave failure	Please check slave settings and statuses.
DFB_ACKNOWLEDGE	Slave has received request, but it takes longer to handle.	N/A
DFB_SLAVE_DEVICE_ BUSY	Slave is busy.	N/A
DFB_GATEWAY_PATH_ UNAVAILABLE	Wrong Gateway path	Please check Gateway configuration, or Gateway is busy.
DFB_GATEWAY_DEVICE_FAILED_TO_RESPOND	Slave device in Gateway fails to respond.	Please check if the slave 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_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_CODE	Invalid setting for uiFunctionCode.	Please check if the setting value of uiFunctionCode is correct,

Description	Cause of Error	Corrective Action
DFB_NO_MASTER_ CONFIG	Delta_Modbus_Master_COM_Port device does not exist.	Make sure that Delta_Modbus_Master_COM_Port 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 allowable limit.
DFB_UNDEFINED	Undefined or has not yet been executed.	Wait for the execution of FB instruction being completed.

Chapter 8 Network Communication Instructions

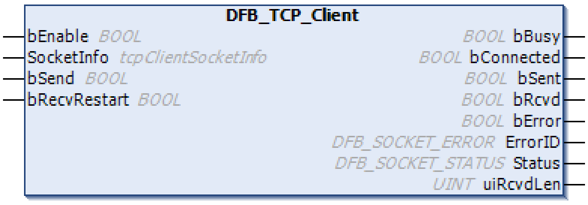
8.1 DFB_TCP_Client

- **Supported Products**

  - AX308E, AX-364E, AX-300, AX-324N

DFB_TCP_Client: TCP socket client instruction.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_ TCP_Client	 DFB_TCP_Client
		bEnable BOOL
		SocketInfo tcpClientSocketInfo
		bSend BOOL
		bRecvRestart BOOL
		BOOL bBusy
		BOOL bConnected
		BOOL bSent
		BOOL bRcvd
		BOOL bError
		DFB_SOCKET_ERROR ErrorID
		DFB_SOCKET_STATUS Status
		UINT uiRcvdLen
			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)
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)
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	(0)

Name	Function	Data Type	Output Range(Default value)
		(Unit: Byte)	

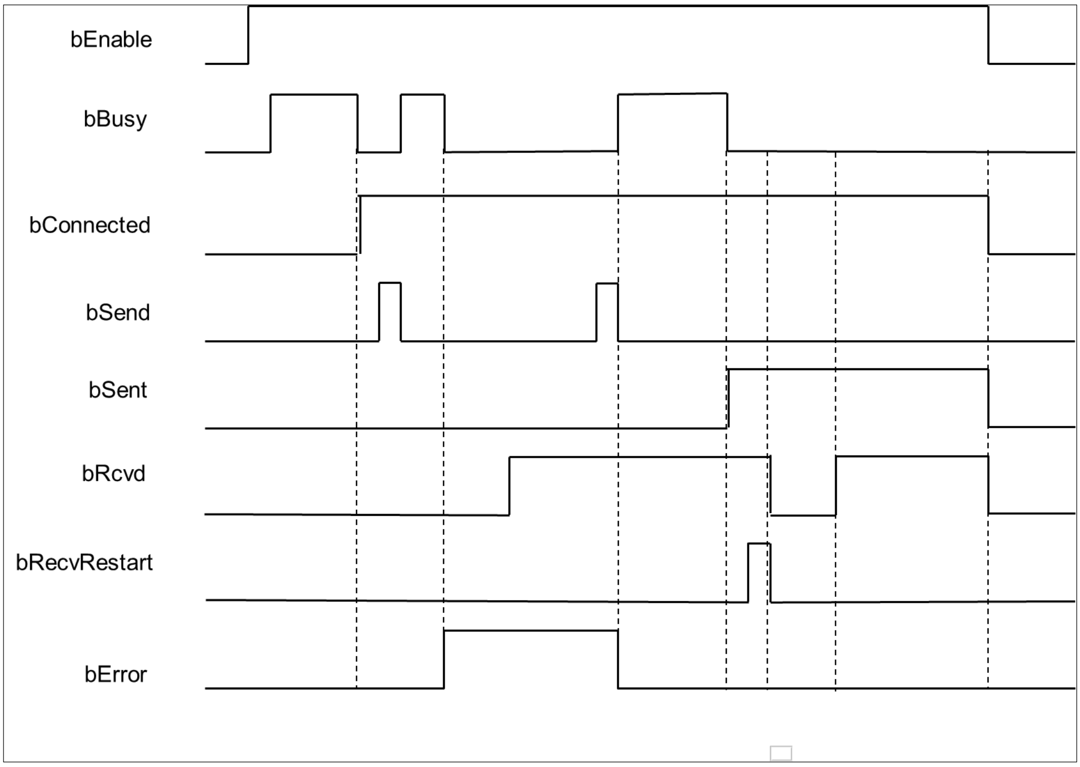
- **DFB_SOCKET_STATUS**

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**



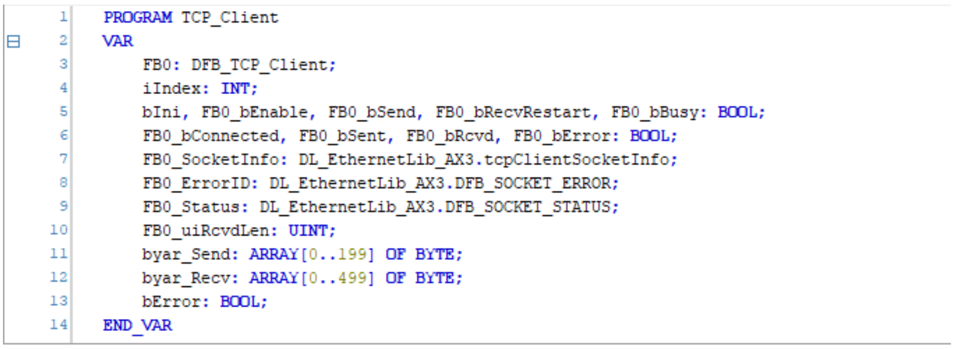
- **Function**

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

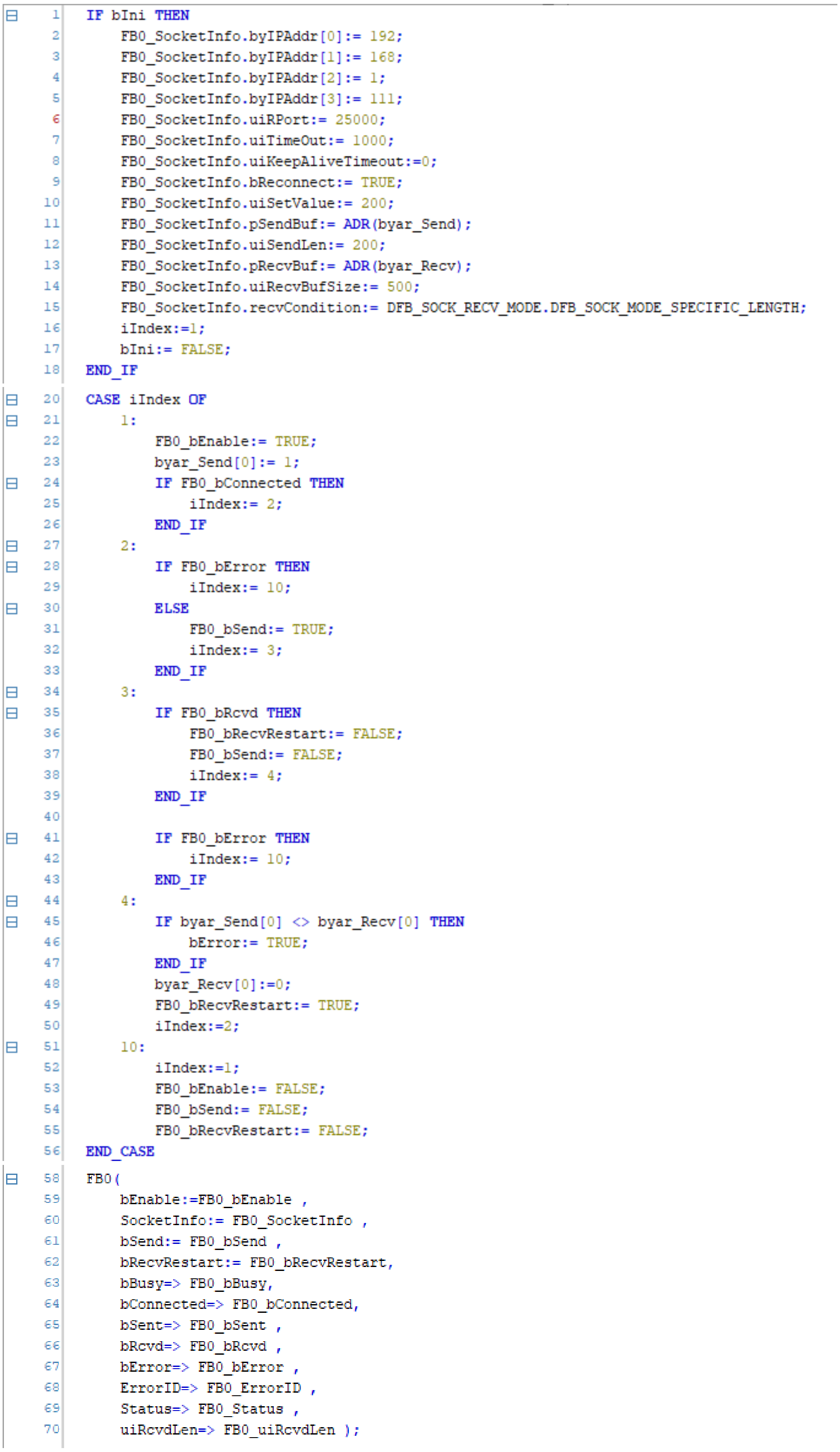
Note: DFB_TCP_Client, DFB_TCP_Server, and DFB_UDP_Socket instructions can be used in a maximum of 32 groups at the same time.

- **Programming Example**

This example uses the FB instruction (DFB_TCP_Client) and Server (IP address: 192.168.1.111, Port: 25000) to establish connection and send 200 Bytes packet. It is expected that Server will send back the same 200 Bytes package. After Server has finished sending back the package, 200 Bytes package will be resent.



```
PROGRAM TCP_Client
VAR
    FB0: DFB_TCP_Client;
    iIndex: INT;
    bIni, FB0_bEnable, FB0_bSend, FB0_bRecvRestart, FB0_bBusy: BOOL;
    FB0_bConnected, FB0_bSent, FB0_bRcvd, FB0_bError: BOOL;
    FB0_SocketInfo: DL_EthernetLib_AX3.tcpClientSocketInfo;
    FB0_ErrorID: DL_EthernetLib_AX3.DFB_SOCKET_ERROR;
    FB0_Status: DL_EthernetLib_AX3.DFB_SOCKET_STATUS;
    FB0_uiRcvdLen: UINT;
    byar_Send: ARRAY[0..199] OF BYTE;
    byar_Recv: ARRAY[0..499] OF BYTE;
    bError: BOOL;
END_VAR
```



```
IF bIni THEN
    FB0_SocketInfo.byIPAddr[0]:= 192;
    FB0_SocketInfo.byIPAddr[1]:= 168;
    FB0_SocketInfo.byIPAddr[2]:= 1;
    FB0_SocketInfo.byIPAddr[3]:= 111;
    FB0_SocketInfo.uiRPort:= 25000;
    FB0_SocketInfo.uiTimeOut:= 1000;
    FB0_SocketInfo.uiKeepAliveTimeout:=0;
    FB0_SocketInfo.bReconnect:= TRUE;
    FB0_SocketInfo.uiSetValue:= 200;
    FB0_SocketInfo.pSendBuf:= ADR(byar_Send);
    FB0_SocketInfo.uiSendLen:= 200;
    FB0_SocketInfo.pRecvBuf:= ADR(byar_Recv);
    FB0_SocketInfo.uiRecvBufSize:= 500;
    FB0_SocketInfo.recvCondition:= DFB_SOCK_RECV_MODE.DFB_SOCK_MODE_SPECIFIC_LENGTH;
    iIndex:=1;
    bIni:= FALSE;
END_IF

CASE iIndex OF
    1:
        FB0_bEnable:= TRUE;
        byar_Send[0]:= 1;
        IF FB0_bConnected THEN
            iIndex:= 2;
        END_IF
    2:
        IF FB0_bError THEN
            iIndex:= 10;
        ELSE
            FB0_bSend:= TRUE;
            iIndex:= 3;
        END_IF
    3:
        IF FB0_bRcvd THEN
            FB0_bRecvRestart:= FALSE;
            FB0_bSend:= FALSE;
            iIndex:= 4;
        END_IF

        IF FB0_bError THEN
            iIndex:= 10;
        END_IF
    4:
        IF byar_Send[0] <> byar_Recv[0] THEN
            bError:= TRUE;
        END_IF
        byar_Recv[0]:=0;
        FB0_bRecvRestart:= TRUE;
        iIndex:=2;
    10:
        iIndex:=1;
        FB0_bEnable:= FALSE;
        FB0_bSend:= FALSE;
        FB0_bRecvRestart:= FALSE;
END_CASE

FB0(
    bEnable:=FB0_bEnable ,
    SocketInfo:= FB0_SocketInfo ,
    bSend:= FB0_bSend ,
    bRecvRestart:= FB0_bRecvRestart,
    bBusy=> FB0_bBusy,
    bConnected=> FB0_bConnected,
    bSent=> FB0_bSent ,
    bRcvd=> FB0_bRcvd ,
    bError=> FB0_bError ,
    ErrorID=> FB0_ErrorID ,
    Status=> FB0_Status ,
    uiRcvdLen=> FB0_uiRcvdLen );
```

- **Library**
    - DL_EthernetLib.library

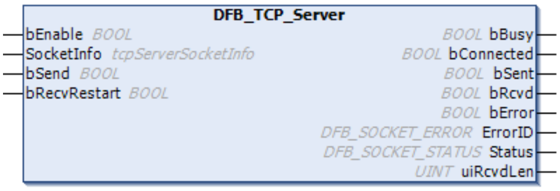
8.2 DFB_TCP_Server

- **Supported Products**

  - AX308E, AX-364E, AX-300, AX-324N

DFB_TCP_Server: TCP socket server instruction

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_ TCP_Server	 DFB_TCP_Server
		bEnable BOOL
		SocketInfo tcpServerSocketInfo
		bSend BOOL
		bRecvRestart BOOL
		BOOL bBusy
		BOOL bConnected
		BOOL bSent
		BOOL bRcvd
		BOOL bError
		DFB_SOCKET_ERROR ErrorID
		DFB_SOCKET_STATUS Status
		UINT uiRcvdLen
			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)
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.
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)

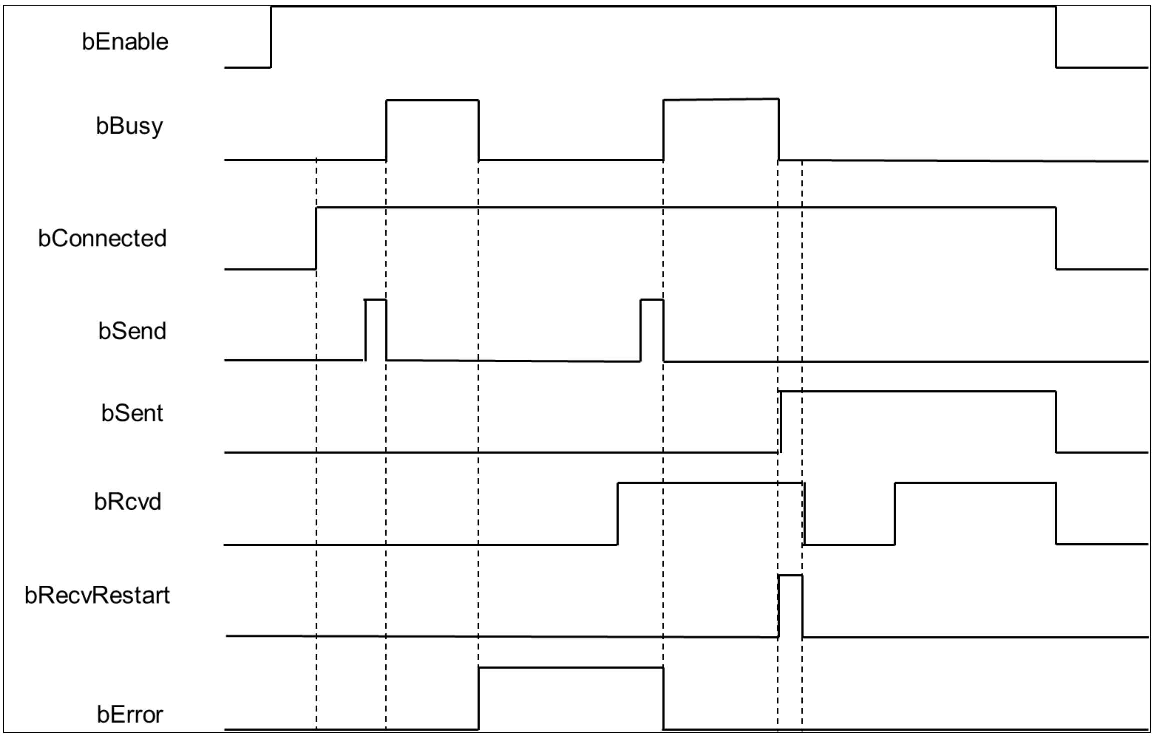
- **DFB_SOCKET_STATUS**

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
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**



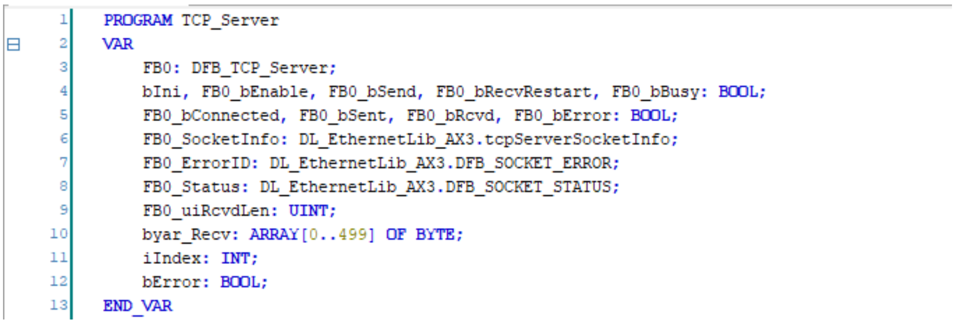
- **Function**

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

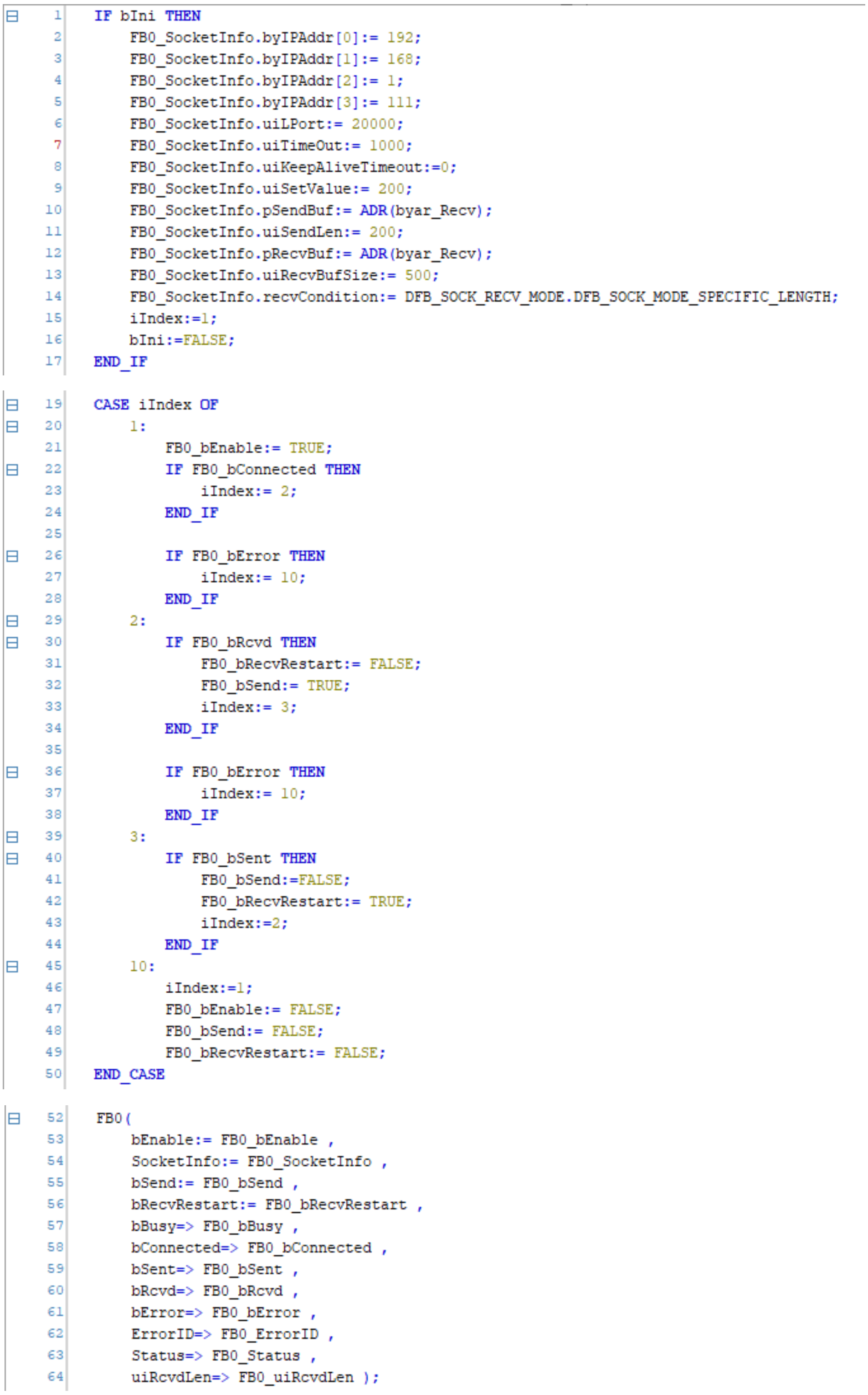
Note: DFB_TCP_Client, DFB_TCP_Server, and DFB_UDP_Socket instructions can be used in a maximum of 32 groups at the same time.

- **Programming Example**

  - This example uses the FB instruction (DFB_TCP_ Server) to open TCP (Port: 20000) and restrict the IP address of Client to be 192.168.1.111, while expecting to receive the package of 200 Bytes data length. After receiving is finished, the same 200 Bytes package will be sent back.



```
PROGRAM TCP_Server
VAR
    FB0: DFB_TCP_Server;
    bIni, FB0_bEnable, FB0_bSend, FB0_bRecvRestart, FB0_bBusy: BOOL;
    FB0_bConnected, FB0_bSent, FB0_bRcvd, FB0_bError: BOOL;
    FB0_SocketInfo: DL_EthernetLib_AX3.tcpServerSocketInfo;
    FB0_ErrorID: DL_EthernetLib_AX3.DFB_SOCKET_ERROR;
    FB0_Status: DL_EthernetLib_AX3.DFB_SOCKET_STATUS;
    FB0_uiRcvdLen: UINT;
    byar_Recv: ARRAY[0..499] OF BYTE;
    iIndex: INT;
    bError: BOOL;
END_VAR
```



```
IF bIni THEN
    FB0_SocketInfo.byIPAddr[0]:= 192;
    FB0_SocketInfo.byIPAddr[1]:= 168;
    FB0_SocketInfo.byIPAddr[2]:= 1;
    FB0_SocketInfo.byIPAddr[3]:= 111;
    FB0_SocketInfo.uiLPort:= 20000;
    FB0_SocketInfo.uiTimeOut:= 1000;
    FB0_SocketInfo.uiKeepAliveTimeout:=0;
    FB0_SocketInfo.uiSetValue:= 200;
    FB0_SocketInfo.pSendBuf:= ADR(byar_Recv);
    FB0_SocketInfo.uiSendLen:= 200;
    FB0_SocketInfo.pRecvBuf:= ADR(byar_Recv);
    FB0_SocketInfo.uiRecvBufSize:= 500;
    FB0_SocketInfo.recvCondition:= DFB_SOCK_RECV_MODE.DFB_SOCK_MODE_SPECIFIC_LENGTH;
    iIndex:=1;
    bIni:=FALSE;
END_IF

CASE iIndex OF
    1:
        FB0_bEnable:= TRUE;
        IF FB0_bConnected THEN
            iIndex:= 2;
        END_IF

        IF FB0_bError THEN
            iIndex:= 10;
        END_IF
    2:
        IF FB0_bRcvd THEN
            FB0_bRecvRestart:= FALSE;
            FB0_bSend:= TRUE;
            iIndex:= 3;
        END_IF

        IF FB0_bError THEN
            iIndex:= 10;
        END_IF
    3:
        IF FB0_bSent THEN
            FB0_bSend:=FALSE;
            FB0_bRecvRestart:= TRUE;
            iIndex:=2;
        END_IF
    10:
        iIndex:=1;
        FB0_bEnable:= FALSE;
        FB0_bSend:= FALSE;
        FB0_bRecvRestart:= FALSE;
END_CASE

FB0(
    bEnable:= FB0_bEnable ,
    SocketInfo:= FB0_SocketInfo ,
    bSend:= FB0_bSend ,
    bRecvRestart:= FB0_bRecvRestart ,
    bBusy=> FB0_bBusy ,
    bConnected=> FB0_bConnected ,
    bSent=> FB0_bSent ,
    bRcvd=> FB0_bRcvd ,
    bError=> FB0_bError ,
    ErrorID=> FB0_ErrorID ,
    Status=> FB0_Status ,
    uiRcvdLen=> FB0_uiRcvdLen );
```

- **Library**

  - DL_EthernetLib.library

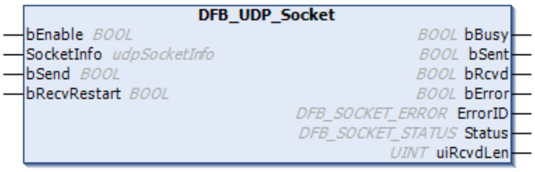
8.3 DFB_UDP_Socket

- **Supported Products**

  - AX308E, AX-364E, AX-300, AX-324N

DFB_UDP_Socket: UDP socket instruction

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_ UDP_Socket	 DFB_UDP_Socket
		bEnable BOOL
		SocketInfo udpSocketInfo
		bSend BOOL
		bRecvRestart BOOL
		BOOL bBusy
		BOOL bSent
		BOOL bRcvd
		BOOL bError
		DFB_SOCKET_ERROR ErrorID
		DFB_SOCKET_STATUS Status
		UINT uiRcvdLen
			DFB_UDP_Socket(
			bEnable:= ,
			RemoteInfo:= ,
			bSend:= ,
			bRecvRestart:= ,
			bBusy=> ,
			bSent=> ,
			bRcvd=> ,
			bError=> ,
			ErrorID=> ,
			Status=> ,
			uiRcvdLen=>);

- **Input**

Name	Function	Data Type	Setting Value (Default Value)
Enable	Execute the function block.	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 sent (Unit: Byte)	UINT	0 ~ 8192
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)

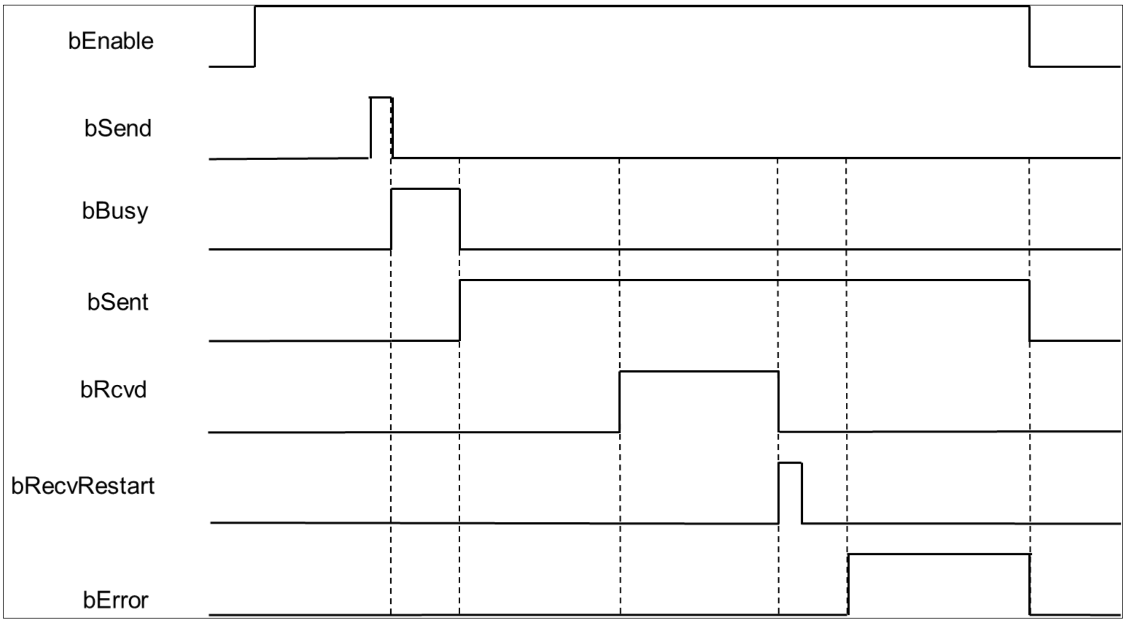
- **DFB_SOCKET_STATUS**

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**



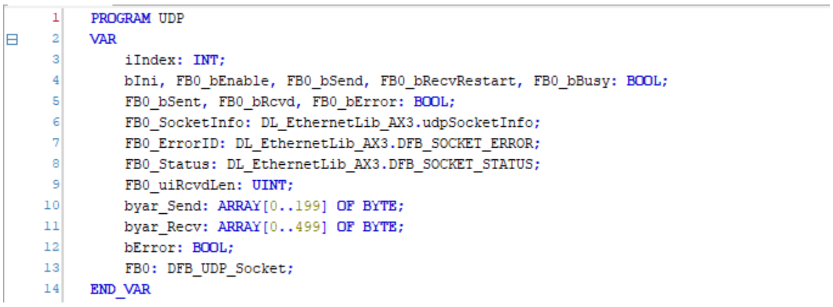
- **Function**

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

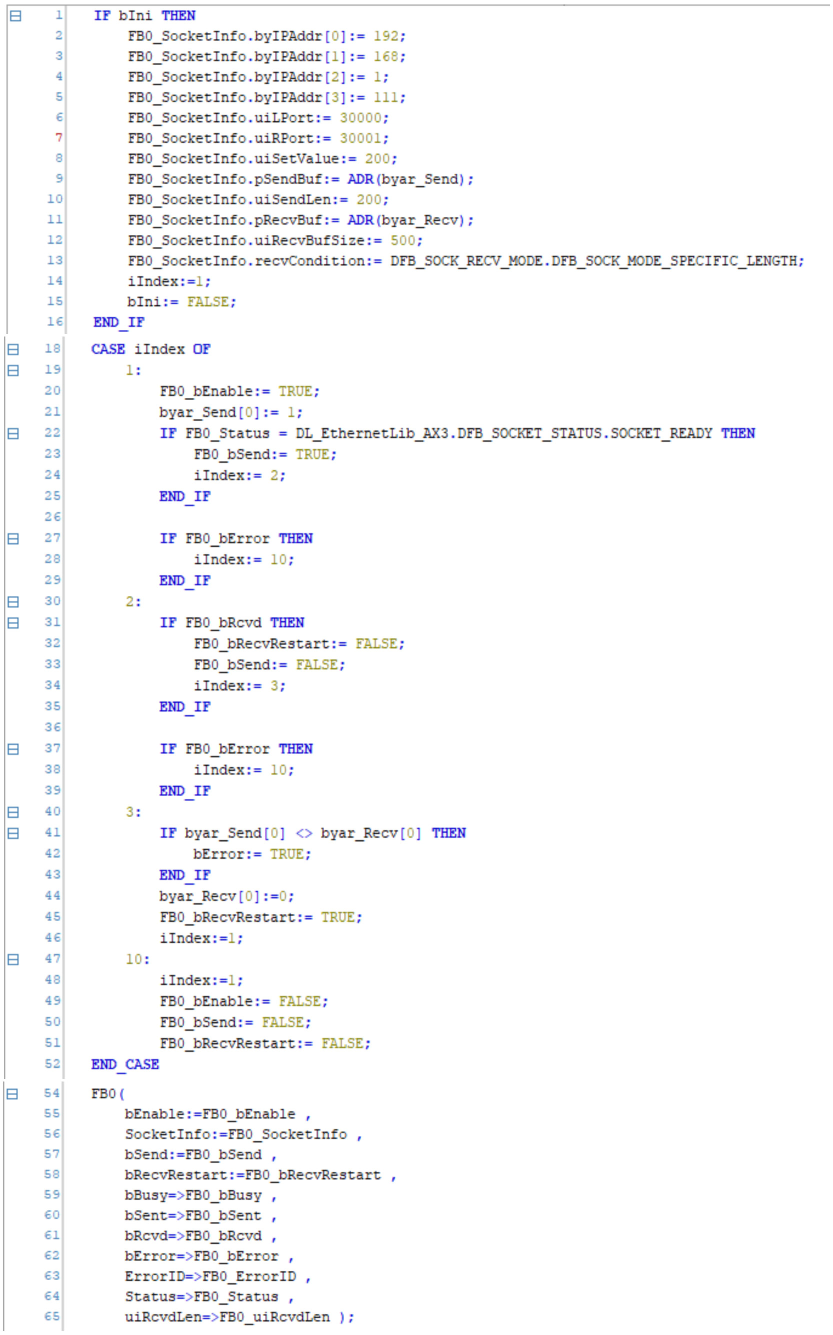
Note: DFB_TCP_Client, DFB_TCP_Server, and DFB_UDP_Socket instructions can be used in a maximum of 32 groups at the same time.

- **Programming Example**

This example uses the FB instruction (DFB_UDP_Socket) to send 200 the package of 200 Bytes data length to the device (IP address: 192.168.1.111, Port: 30001) from UDP (Port: 30000). It is expected that the same package will be replied. After the device has finished returning the same package, 200 Bytes package will be resent.



```
PROGRAM UDP
VAR
    iIndex: INT;
    bIni, FB0_bEnable, FB0_bSend, FB0_bRecvRestart, FB0_bBusy: BOOL;
    FB0_bSent, FB0_bRcvd, FB0_bError: BOOL;
    FB0_SocketInfo: DL_EthernetLib_AX3.udpSocketInfo;
    FB0_ErrorID: DL_EthernetLib_AX3.DFB_SOCKET_ERROR;
    FB0_Status: DL_EthernetLib_AX3.DFB_SOCKET_STATUS;
    FB0_uiRcvdLen: UINT;
    byar_Send: ARRAY[0..199] OF BYTE;
    byar_Recv: ARRAY[0..499] OF BYTE;
    bError: BOOL;
    FB0: DFB_UDP_Socket;
END_VAR
```



```
IF bIni THEN
    FB0_SocketInfo.byIPAddr[0]:= 192;
    FB0_SocketInfo.byIPAddr[1]:= 168;
    FB0_SocketInfo.byIPAddr[2]:= 1;
    FB0_SocketInfo.byIPAddr[3]:= 111;
    FB0_SocketInfo.uiLPort:= 30000;
    FB0_SocketInfo.uiRPort:= 30001;
    FB0_SocketInfo.uiSetValue:= 200;
    FB0_SocketInfo.pSendBuf:= ADR(byar_Send);
    FB0_SocketInfo.uiSendLen:= 200;
    FB0_SocketInfo.pRecvBuf:= ADR(byar_Recv);
    FB0_SocketInfo.uiRecvBufSize:= 500;
    FB0_SocketInfo.recvCondition:= DFB_SOCK_RECV_MODE.DFB_SOCK_MODE_SPECIFIC_LENGTH;
    iIndex:=1;
    bIni:= FALSE;
END_IF
CASE iIndex OF
    1:
        FB0_bEnable:= TRUE;
        byar_Send[0]:= 1;
        IF FB0_Status = DL_EthernetLib_AX3.DFB_SOCKET_STATUS.SOCKET_READY THEN
            FB0_bSend:= TRUE;
            iIndex:= 2;
        END_IF

        IF FB0_bError THEN
            iIndex:= 10;
        END_IF
    2:
        IF FB0_bRcvd THEN
            FB0_bRecvRestart:= FALSE;
            FB0_bSend:= FALSE;
            iIndex:= 3;
        END_IF

        IF FB0_bError THEN
            iIndex:= 10;
        END_IF
    3:
        IF byar_Send[0] <> byar_Recv[0] THEN
            bError:= TRUE;
        END_IF
        byar_Recv[0]:=0;
        FB0_bRecvRestart:= TRUE;
        iIndex:=1;
    10:
        iIndex:=1;
        FB0_bEnable:= FALSE;
        FB0_bSend:= FALSE;
        FB0_bRecvRestart:= FALSE;
END_CASE
FB0(
    bEnable:=FB0_bEnable ,
    SocketInfo:=FB0_SocketInfo ,
    bSend:=FB0_bSend ,
    bRecvRestart:=FB0_bRecvRestart ,
    bBusy=>FB0_bBusy ,
    bSent=>FB0_bSent ,
    bRcvd=>FB0_bRcvd ,
    bError=>FB0_bError ,
    ErrorID=>FB0_ErrorID ,
    Status=>FB0_Status ,
    uiRcvdLen=>FB0_uiRcvdLen );
```

- **Library**

  - DL_EthernetLib.library

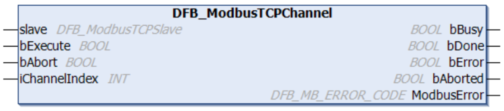
8.4 DFB_ModbusTCPChannel

- **Supported Products**

  - AX308E, AX-364E, AX-300, AX-324N

DFB_ModbusTCPChannel: Modus TCP slave Channel control instruction

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_
	ModbusTCPChannel
		 DFB_ModbusTCPChannel
		slave DFB_ModbusTCPSlave
		bExecute BOOL
		bAbort BOOL
		iChannelIndex INT
		BOOL bBusy
		BOOL bDone
		BOOL bError
		BOOL bAborted
		DFB_MB_ERROR_CODE ModbusError
			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**

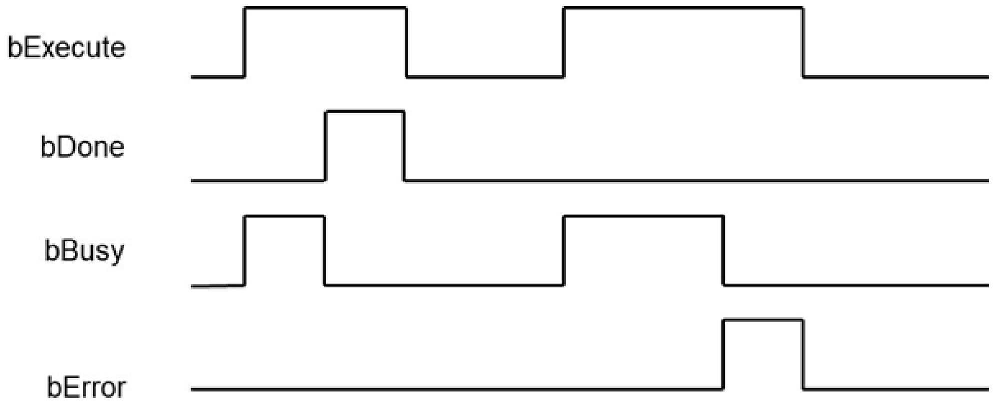
Name	Function	Data type	Output range (Default value)
bBusy	The FB instruction is being executed.	BOOL	True/False (False)
bDone	The FB instruction execution is completed.	BOOL	True/False (False)

Name	Function	Data type	Output range (Default value)
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**

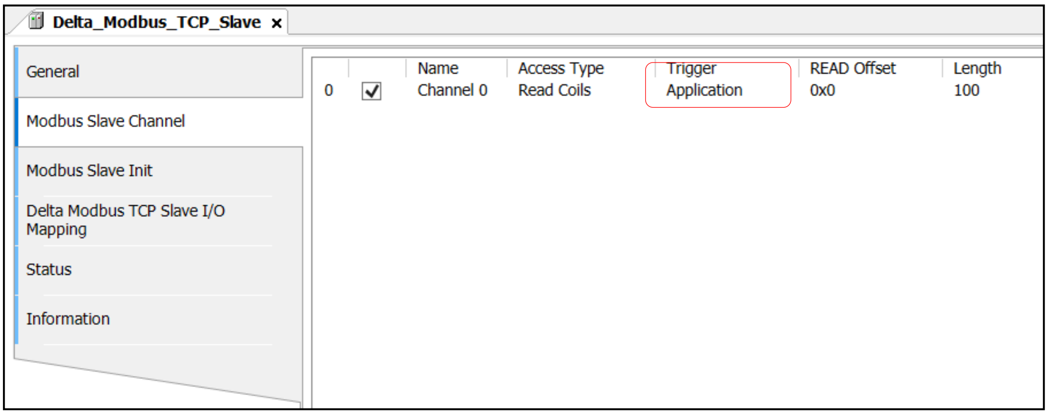
Name	Timing for shifting to True	Timing for shifting to False
bDone	● When FB instruction execution is completed.	● When bExecute shifts to False.
bBusy	● When FB instruction execution starts.	● When FB instruction execution is completed. ● When bExecute shifts to False.
bError	● When an error occurs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

- **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.



		Name	Access Type	Trigger	READ Offset	Length
0	☑	Channel 0	Read Coils	Application	0x0	100

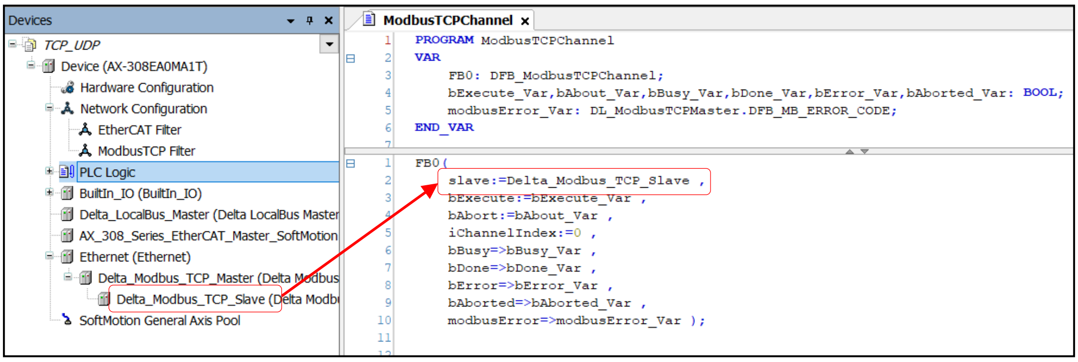
Note 1: For more details of Modbus TCP slave configuration, please refer to chapter 9.3 “Ethernet Communication” in AX-3 Series Operational Manual.

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

Note 3: DFB_ModbusTCPChannel and DFB_ModbusTCPRequest instructions can be used in a maximum of 32 groups at the same time.

- **Programming Example**

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


Devices
TCP_UDP
Device (AX-308EA0MA1T)
Hardware Configuration
Network Configuration
EtherCAT Filter
ModbusTCP Filter
PLC Logic
BuiltIn_IO (BuiltIn_IO)
Delta_LocalBus_Master (Delta LocalBus Master
AX_308_Series_EtherCAT_Master_SoftMotion
Ethernet (Ethernet)
Delta_Modbus_TCP_Master (Delta Modbus
Delta_Modbus_TCP_Slave (Delta Modbu
SoftMotion General Axis Pool
ModbusTCPChannel

```
PROGRAM ModbusTCPChannel
VAR
    FB0: DFB_ModbusTCPChannel;
    bExecute_Var,bAbout_Var,bBusy_Var,bDone_Var,bError_Var,bAborted_Var: BOOL;
    modbusError_Var: DL_ModbusTCPMaster.DFB_MB_ERROR_CODE;
END_VAR
```

```
FB0(
    slave:=Delta_Modbus_TCP_Slave ,
    bExecute:=bExecute_Var ,
    bAbort:=bAbout_Var ,
    iChannelIndex:=0 ,
    bBusy=>bBusy_Var ,
    bDone=>bDone_Var ,
    bError=>bError_Var ,
    bAborted=>bAborted_Var ,
    modbusError=>modbusError_Var );
```

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

- **Library**

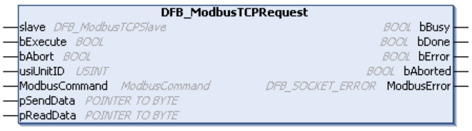
  - DL_ModbusTCPMaster.library

8.5 DFB_ModbusTCPRequest

- **Supported Products**

  - AX308E, AX-364E, AX-300, AX-324N

DFB_ModbusTCPRequest: Mobus TCP command.

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

- **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**

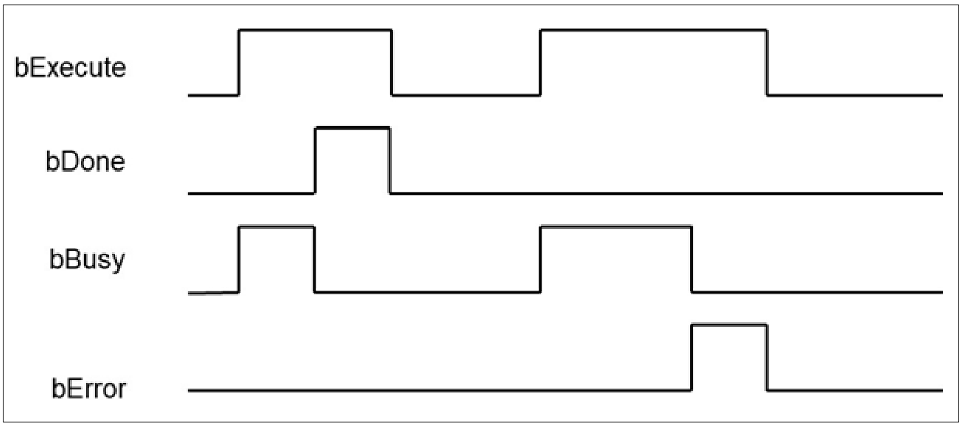
Name	Function	Data Type	Output Range(Default value)
bBusy	True when the instruction is being executed.	BOOL	True/False (False)
bDone	The execution of FB is completed.	BOOL	True/False (False)
bError	True if an error occurs.	BOOL	True/False (False)
bAborted	No function	BOOL	--
ModbusError	Error code	DFB_MB_ERROR_CODE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



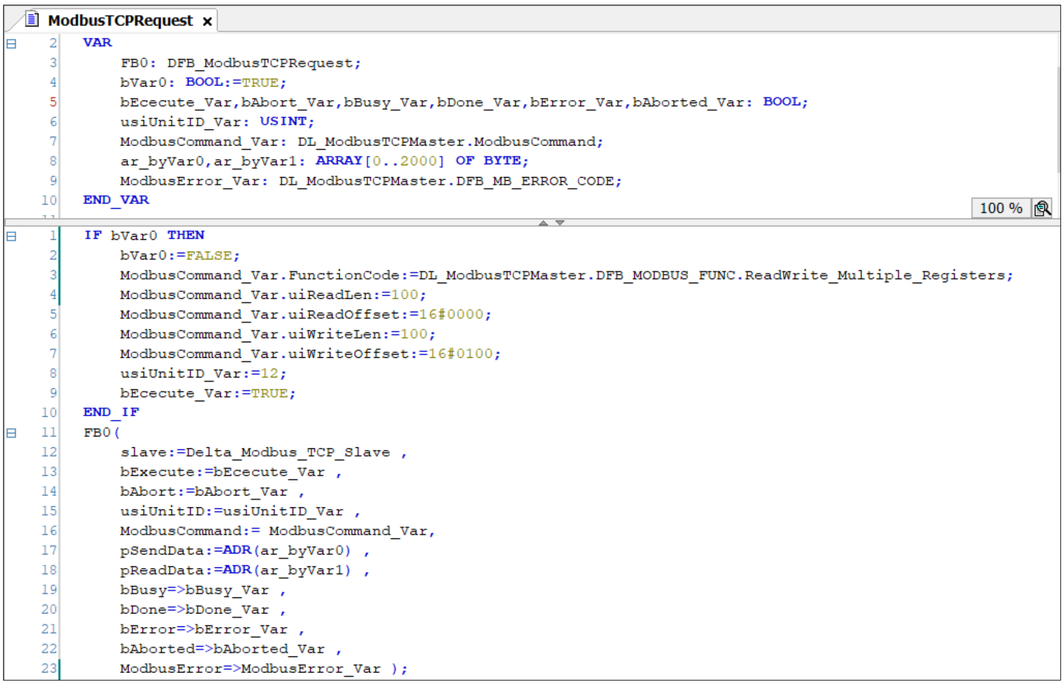
- **Function**

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.)

Note: DFB_ModbusTCPChannel and DFB_ModbusTCPRequest instructions can be used in a maximum of 32 groups at the same time.

- **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.



```
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
```

```
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 );
```

- **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 slave wires function normally as well as the proper wiring.
DFB_SALVE_DEVICE_ FAILURE	Slave failure	Please check slave settings and statuses.
DFB_ACKNOWLEDGE	Slave has received request, but it takes longer to handle.	N/A
DFB_SLAVE_DEVICE_ BUSY	Slave is busy.	N/A
DFB_GATEWAY_PATH_UNAVAILABLE	Wrong Gateway path	Please check Gateway configuration, or Gateway is

Description	Cause of Error	Corrective Action
		busy.
DFB_GATEWAY_DEVICE_FAILED_TO_RESPOND	Slave device in Gateway fails to respond.	Please check if the slave 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.

Description	Cause of Error	Corrective Action
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_INVALID_ FUNCTION_CODE	Function block does not support this function code.	Please check function block settings.
DFB_NO_ETHERNET_ CONFIG	Ethernet Adapter device does not exist.	Please check if Ethernet Adapter device 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_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.

Chapter 9 Instructions for Reading and Writing a Memory Card

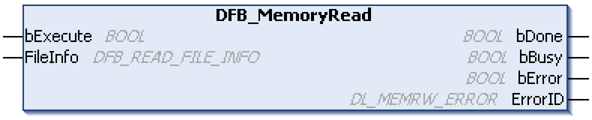
9.1 DFB_MemoryRead

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

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

Name	Function	Data Type	Setting Value (Default value)
			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)
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.

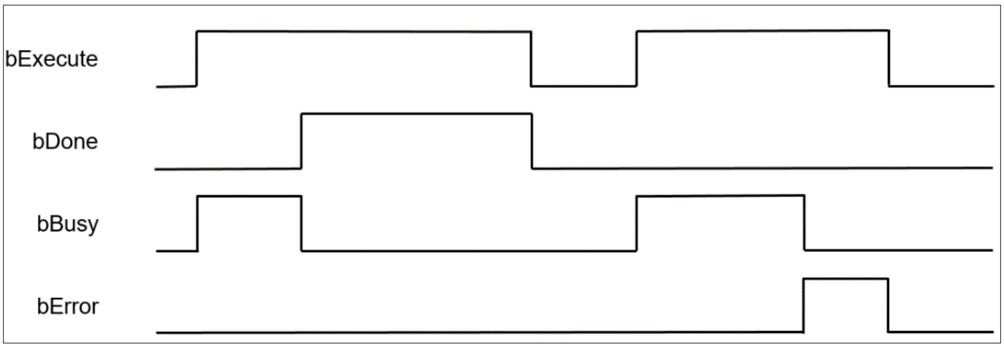
- **Output**

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**

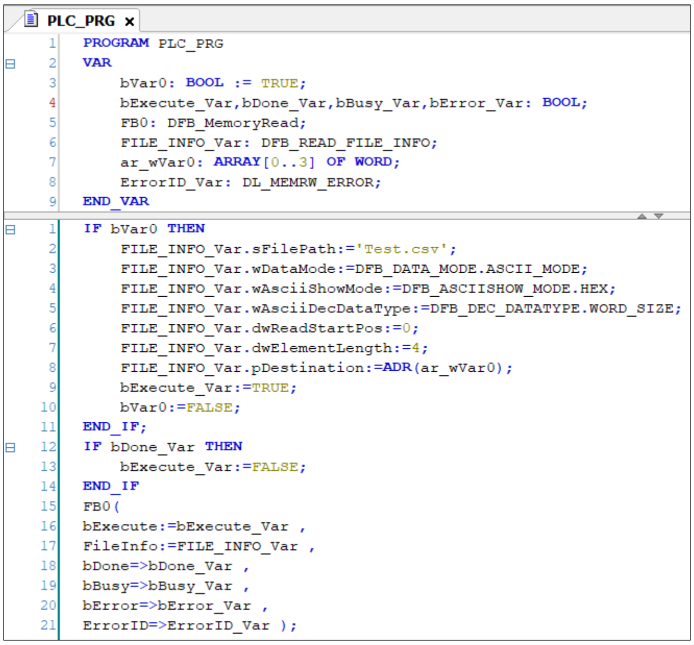


- **Function**

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).



```
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].

- **Library**

  - DL_MemRW_AX3.library

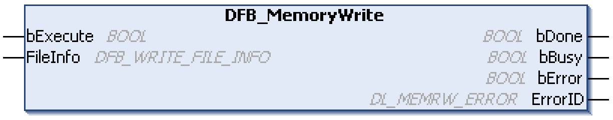
9.2 DFB_MemoryWrite

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_MemoryWrite: Write a memory card.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_MemoryWrite	 DFB_MemoryWrite
		bExecute BOOL
		FileInfo DFB_WRITE_FILE_INFO
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DL_MEMRW_ERROR ErrorID
			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_WRITE_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)
wAsciiDecDataType	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

Name	Function	Data Type	Setting Value (Default value)
			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 file to be created.	DFB_ ACCESS_MODE	DFB_ACCESS_MODE.NEW 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.

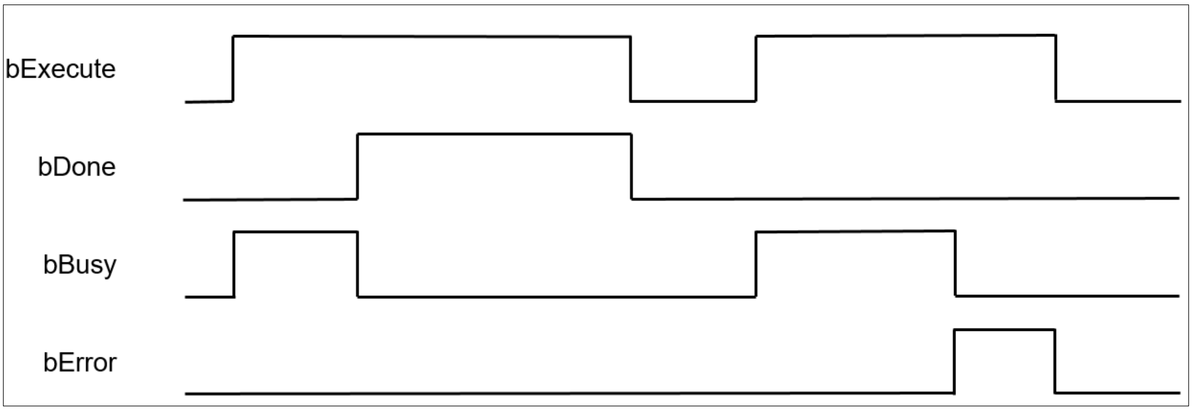
- **Output**

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**

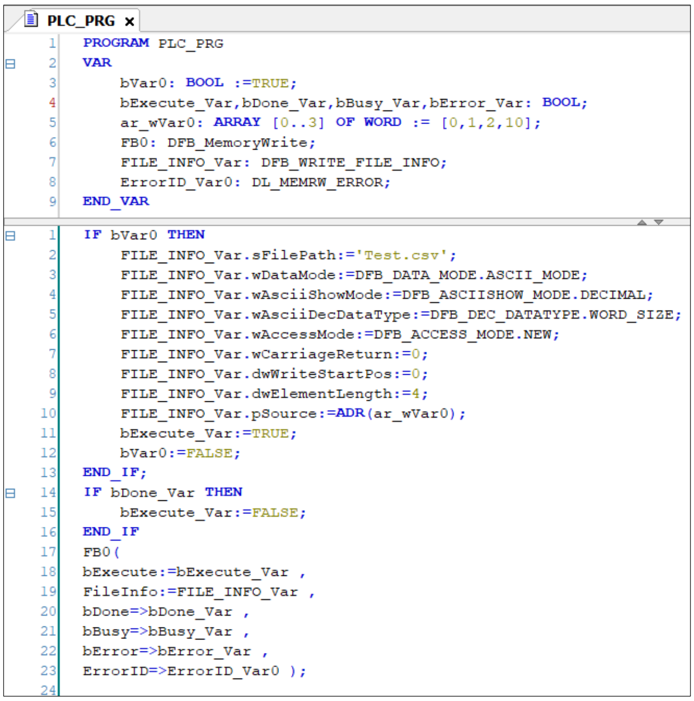


- **Function**

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.



```
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.

- **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 check external SD card Please contact us directly
DFB_MEMREAD_ERR_ PARAMETER	Invalid parameter inputs.	Please check if the input parameters 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 exist.	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_OBJECT	The object does not exist.	1. Please confirm if the controller and library versions support the object 2. Please confirm whether the file exists. 3. Please confirm whether the value meets the specification 4. Please confirm whether the starting position is greater than the length of the data
DFB_MEMREAD_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.。
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

Description	Cause of Error	Corrective Action
		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_CHANGE	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_ACCESS_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.
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 session ID.	Please log in again or reboot the controller.
DFB_MEMREAD_ERR_ EXCEPTION	Exception occurs.	Please check the error log.
DFB_MEMWRITE_ERR_ FAILED	Internal error	1. Please check the external SD card 2. Please contact the original manufacturer
DFB_MEMWRITE_ERR_ PARAMETER	Input parameter error	Please confirm whether the input parameters are correct.
DFB_MEMWRITE_ERR_ NOTINITIALIZED	The instruction cannot be executed because the element initialization has not been completed	Please restart the controller.
DFB_MEMWRITE_ERR_ VERSION	Wrong version	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ TIMEOUT	Execution time out	Please restore the controller to factory settings (Reset Origin). If the problem persists, please contact us directly.
DFB_MEMWRITE_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_MEMWRITE_ERR_ PENDING	Program to be executed	Please restart the controller.
DFB_MEMWRITE_ERR_ NUMPENDING	Too many programs in execute query	Please restore the controller to factory settings (Reset Origin). If the problem persists, please contact us directly.
DFB_MEMWRITE_ERR_ NOTIMPLEMENTED	No such function	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ INVALIDID	Incorrect ID	Please check if the firmware and the library version are supported.

Description	Cause of Error	Corrective Action
DFB_MEMWRITE_ERR_ OVERFLOW	Value out of range	Please restore the controller to factory settings (Reset Origin). If the problem persists, please contact us directly.
DFB_MEMWRITE_ERR_ BUFFERSIZE	Memory size 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_MEMWRITE_ERR_ NO_OBJECT	No such object	1. Please check the external SD card 2. Please contact the original manufacturer
DFB_MEMWRITE_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.
DFB_MEMWRITE_ERR_ DUPLICATE	Duplicate object name	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ MEMORY_OVERWRITE	The memory exceeds the writable range	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_MEMWRITE_ERR_ INVALID_HANDLE	Invalid object	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ END_OF_OBJECT	The maximum range of the object has been reached	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ NO_CHANGE	No change	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ INVALID_INTERFACE	Invalid or unknown interface	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ NOT_SUPPORTED	This feature is not supported	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_NO_ ACCESS_RIGHTS	Not authorized to execute this command	Please check if the firmware and the library version are supported.
DFB_MEMWRITE_ERR_ OUT_OF_LIMITS	Exceeded limited resources	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_MEMWRITE_ERR_ ENTRIES_REMAINING	Unable to transmit due to limited resources	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_MEMWRITE_ERR_ INVALID_SESSION_ID	Invalid online ID	Please restart the controller.
DFB_MEMWRITE_ERR_ EXCEPTION	An exception error occurred	Please confirm the error log.

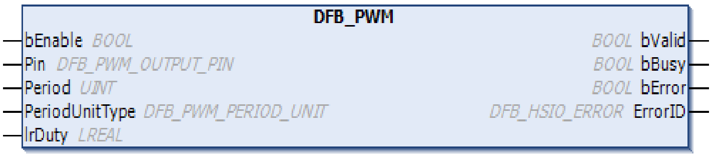
Chapter 10 High Speed Output Instructions

10.1 DFB_PWM

- **Supported Products**

  - AX-308E, AX-364E, AX-324N

DFB_PWM：DFB_PWM is used to generate Pulse-width modulation output signals which is frequency-adjustable.

FB/FC	Instruction	Graphic Expression
FB	DFB_PWM	 DFB_PWM bEnable BOOL Pin DFB_PWM_OUTPUT_PIN Period UINT PeriodUnitType DFB_PWM_PERIOD_UNIT lrDuty LREAL BOOL bValid BOOL bBusy BOOL bError DFB_HSIO_ERROR ErrorID

ST Language

```
DFB_PWM_instance (
bEnable:=,
Pin:=,
Period:=
PeriodUnitType:=,
leDuty:=,
bValid =>,
bBusy =>,
bError =>,
ErrorID =>
);
```

- **Input Pins**

Name	Function	Data Type	Setting Value (Default value)	Effective time
bEnable	Execute the instruction when bEnable changes from False to True.	BOOL	True/False (True)	-
Pin	PWM output pins number	DFB_PWM_OUTPUT_PIN	0~7 (OUT00)	When bEnable is TRUE.
PeriodUnitType	PWM period unit	DFB_PWM_PERIOD_UNIT	MicroSecond or MilliSecond (MicroSecond)	When bEnable is TRUE.
Period	PWM period	UINT	MicroSecond :1~65535 MilliSecond :1~42949 (1)	When bEnable is TRUE.
IrDuty	PWM duty cycle	LREAL	0~100 (0)	Continuously effective when bEnable is TRUE.

Note:

1. AX-332E only support scope: 0-3
2. AX-332E only support scope: MicroSecond: 1000-2000, MilliSecond: 1-20

- **Output Pins**

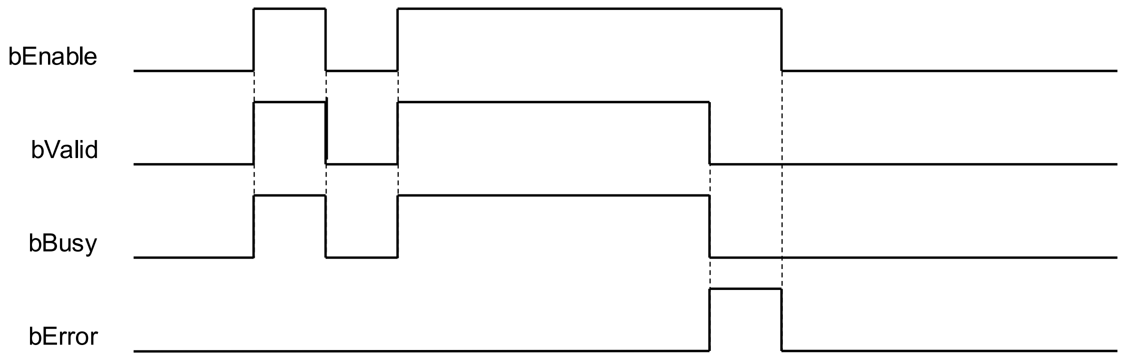
Name	Function	Data Type	Output Value Range (Default value)
bValid	Function block output is valid.	BOOL	True/False (False)
bBusy	Function block is being busy.	BOOL	True/False (False)
bError	Function block error	BOOL	True/False (False)
ErrorID	Error ID	DFB_HSIO_ERRO*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)

***Note:** DMC_ERROR：Enumeration (Enum)

- **Output Updating Timing**

Name	Timing for shifting to True	Timing for shifting to False
bValid	● When bExecute shifts to True.	● When bError shifts to True. ● When bEnable shifts to False.
bBusy	● When bExecute shifts to True.	● When bError shifts to True. ● When bEnable shifts to False.
bError	● When instruction input parameter is illegal, or an error occurs during instruction execution.	● When bExecute shifts to False.
ErrorID		

- **Timing Diagram**



- **Function Description**

- Will generate a square wave signals of specified frequency and duty cycle which can be used to control plant that needs to receive continuously changing physical quantities.
- This function only supports output contacts.
- Firmware needs to be V1.01.0 or later.

- **Troubleshooting**

If an error occurs during the execution of the instruction, Error will change to True, and the axis motion will stop. Refer to ErrorID (Error Code) to address the problem.

Programming Example

This programming example uses 8 sets of PWM output sine wave PWM, and the phase of each set is 45 degrees, which will make OUT LED appear water lamp behavior on the machine. The speed and direction are determined by the size and sign of "F_SIN_Hz" respectively.

- Variable Declaration

```
VAR CONSTANT
        M_PI : LREAL := 3.14159265358979323846;
END_VAR

VAR
        F_SIN_Hz : LREAL := 0.3;
        T_SIN_ms : LREAL;
        tms : UDINT := 0;


        i : UINT;
        PWM : ARRAY[0..7] OF DFB_PWM;
END_VAR
```

- Program

```
FOR i := 0 TO 7 BY 1 DO
        PWM[i].bEnable := TRUE;
        PWM[i].Pin := i;
        PWM[i].PeriodUnitType := DFB_PWM_PERIOD_UNIT.MicroSecond; //usec
        PWM[i].Period := 100;
        PWM[i].lrDuty := 100.0*(0.5*SIN(2*M_PI*tms/T_SIN_ms + i*2*M_PI/8)+0.5);

        PWM[i]();
END_FOR


IF 0.0 = F_SIN_Hz THEN
        F_SIN_Hz := 1.0;
END_IF

T_SIN_ms := TO_DINT(1000/F_SIN_Hz);
tms := tms + 1;
IF tms >= ABS(T_SIN_ms) THEN
        tms := 0;
END_IF
```

- **Library**
  - DL_LogDmp_AX3.library

Note: From version 1.0.5.0 and later, lilbrary DL_BuiltInIO_AX3 is changed to DL_BuiltInIO_library.

10.2 Error Codes and Troubleshooting

Description	Cause of Error	Corrective Action
DFB_PWM_UNSUPPORT_FW_VERSION	Firmware does not match.	Update firmware
DFB_PWM_PIN_ID_OVER_RANGE	Pin number is out of range.	Enter an appropriate pin ID
DFB_PWM_THIS_PIN_ID_IS_USED_ON_OTHER_PWM_FB	Pin is reused.	Do not enter the pin ID that has been used.
DFB_PWM_PERIOD_UNIT_IS_NO_DEFINITION	Wrong unit	Enter an appropriate period unit
DFB_PWM_PERIOD_OVER_RANGE	Period is out of range	Enter an appropriate period value

Chapter 11 Additional Instructions

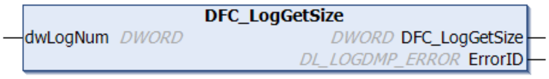
11.1 DFC_LogGetSize

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

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**

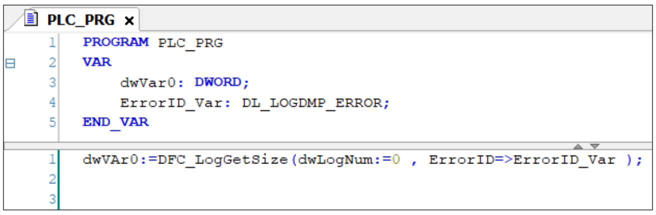
Name	Function	Data type	Output range (Default value)
DFC_LOG_ GETSIZE	Data size of the log files. (Return type)	DWORD (Unit: BYTE)	0 ~ 65536 (0)
ErrorID	Error codes	DL_LOGDMP_ERROR	DL_LOGDMP_ERR (DFC_NO_ERROR)

- **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.



```
PROGRAM PLC_PRG
VAR
    dwVar0: DWORD;
    ErrorID_Var: DL_LOGDMP_ERROR;
END_VAR
```

```
dwVAr0:=DFC_LogGetSize(dwLogNum:=0 , ErrorID=>ErrorID_Var );
```

- **Library**

  - DL_LogDmp_AX3.library

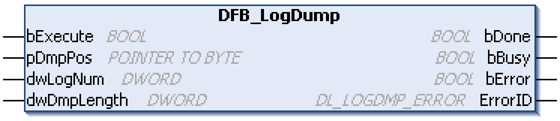
11.2 DFB_LogDump

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_LogDump: Read the log files of the controller.

FB/FC	Instruction	Graphic Expression	ST Language

FB	DFB_ LogDump	 DFB_LogDump
		bExecute BOOL
		pDmpPos POINTER TO BYTE
		dwLogNum DWORD
		dwDmpLength DWORD
		BOOL bDone
		BOOL bBusy
		BOOL bError
		DL_LOGDMP_ERROR ErrorID
			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.

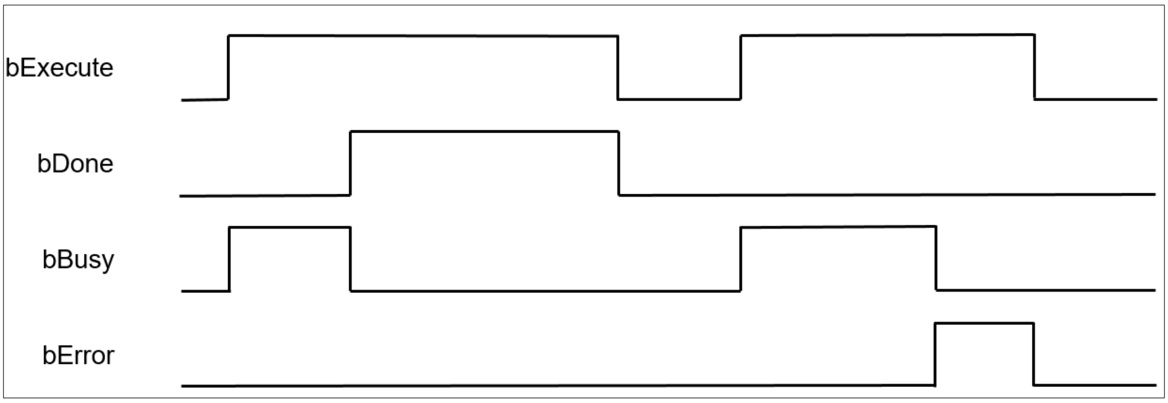
- **Output**

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**

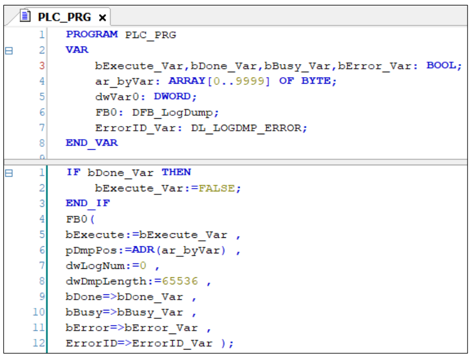


- **Function**

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.



```
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 );
```

- **Library**

  - DL_LogDmp_AX3.library

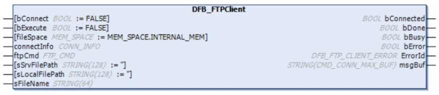
11.3 DFB_FTPClient

- **Supported Products**

  - AX-308E, AX-364E, AX-300, AX-324N

DFB_FTPClient：Access FTP server and upload and download data.

FB/FC	Instruction	Graphic Expression	ST LANGUAGE

FB	DFB_ FTPClient		DFB_FTPClient(
			bConnect:= ,
			bExecute:= ,
			fileSpace:= ,
			connectInfo:= ,
			ftpCmd:= ,
			sSrvFilePath:= ,
			sLocalFilePath:= ,
			sFileName:= ,
			bConnected=> ,
			bDone=> ,
			bBusy=> ,
			bError=> ,
			ErrorId=> ,
			msgBuf=>);

DFB_FTPClient
[bConnect BOOL := FALSE]
[bExecute BOOL := FALSE]
[fileSpace MEM_SPACE := MEM_SPACE.INTERNAL_MEM]
connectInfo CONN_INFO
ftpCmd FTP_CMD
[sSrvFilePath STRING(128) := '']
[sLocalFilePath STRING(128) := '']
sFileName STRING(64)
BOOL bConnected
BOOL bDone
BOOL bBusy
BOOL bError
DFB_FTP_CLIENT_ERROR ErrorId
STRING(CMD_CONN_MAX_BUF) msgBuf

- **Input**

Name	Function	Data Type	Setting Value (Default value)
bConnect	Establish FTP server connection	BOOL	True/False（False）
bExecute	Execute the ftpCmd command	BOOL	True/False（False）
fileSpace	File storage location	MEM_SPACE	INTERNAL_MEM /EXTENAL_SD (INTERNAL_MEM)
connectInfo	FTP server connection information	CONN_INFO	NULL
ftpCmd	FTP execute command	FTP_CMD	NULL
sSrvFilePath	FTP server file path	STRING（128）	（‘’）
sLocalFilePath	File path for upload or download	STRING（128）	（‘’）
sFileName	Uploaded or downloaded filename	STRING（64）	（‘’）

Note: If the file storage location is INTERNAL_MEM, all files will be placed in the FTP folder of the AX308E series host.

- MEM_SPACE

Name	Description
EXTERNAL_SD	External memory card
INTERNAL_MEM	Internal memory
PCI_MEM (*1)	PCI memory
USB_PORT0 (*1)	USB port0
USB_PORT1 (*1)	USB port1
USB_PORT2 (*1)	USB port2
USB_PORT3 (*1)	USB port3

*Note 1: This memory location cannot be selected in this Function block.

■ CONN_INFO

Name	Function	Data Type	Setting Value (Default value)
ftpServerIP	Fftp server ip address	STRING(16)	(‘’)
ftpUsrName	Login account	STRING(64)	(‘’)
ftpPassWord	Login password	STRING(64)	(‘’)
activeMode	Active mode	BOOL	True/False (False)

■ FTP_CMD

Name	Description
NOOP	No action
PUT_FILE	Upload file
GET_FILE	Download file
DELETE	Delete file
APPEND	Add data to the end of the original file

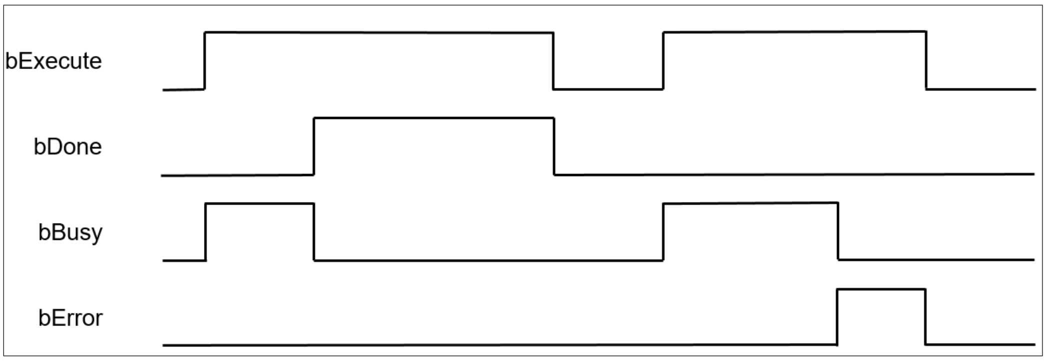
- **Output**

Name	Function	Data Type	Output Range (Default value)
bDone	FBInstruction execution complete flag	BOOL	True/False (False)
bBusy	FBInstruction executing flag	BOOL	True/False (False)
bError	FBInstruction error flag	BOOL	True/False (False)
ErrorID	Error Code	DFB_FTP_CLIENT_ERROR	DFB_FTP_CLIENT_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.	● F When FB instruction execution is completed. ● When bExecute shifts to False.
bError	● When the FB instruction is executed incorrectly or the input value is incorrect	● When bExecute shifts to False.
ErrorID		

- **Timing Diagram**



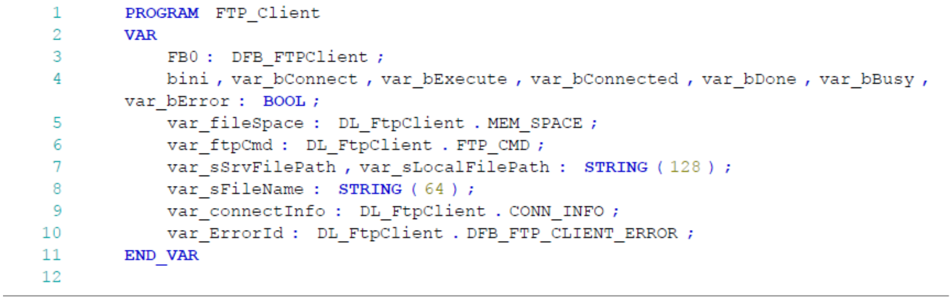
- **Funtion**

Users can upload files to FTP server or download files from FTP server through FBInstruction (DFB_FTPClient).

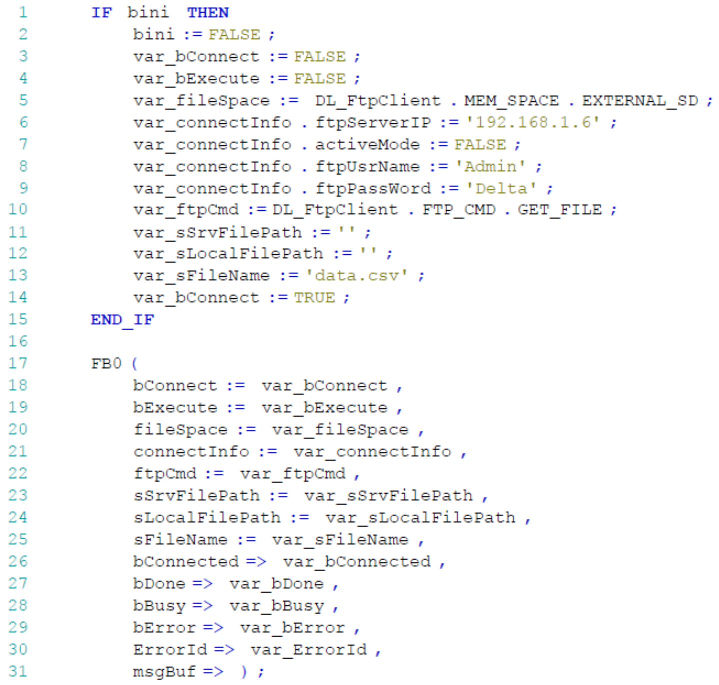
- **Programming Example**

In this example, two AX308E series hosts are used as FTP server and FTP client respectively, and FBInstruction (DFB_FTPClient) is used to read the FTP server file. When the user sets bini to True, the connection with the FTP server is established. After the connection is established successfully (when var_bconnected is True), set var_bEexecute to True, that is, download the FTP server file data.csv to the FTP folder of the external memory card of the AX308E series host.

Note: For the setting method of FTP server, please refer to Chapter 9.6 "FTPFunction" of "AX3 Series Operation Manual"



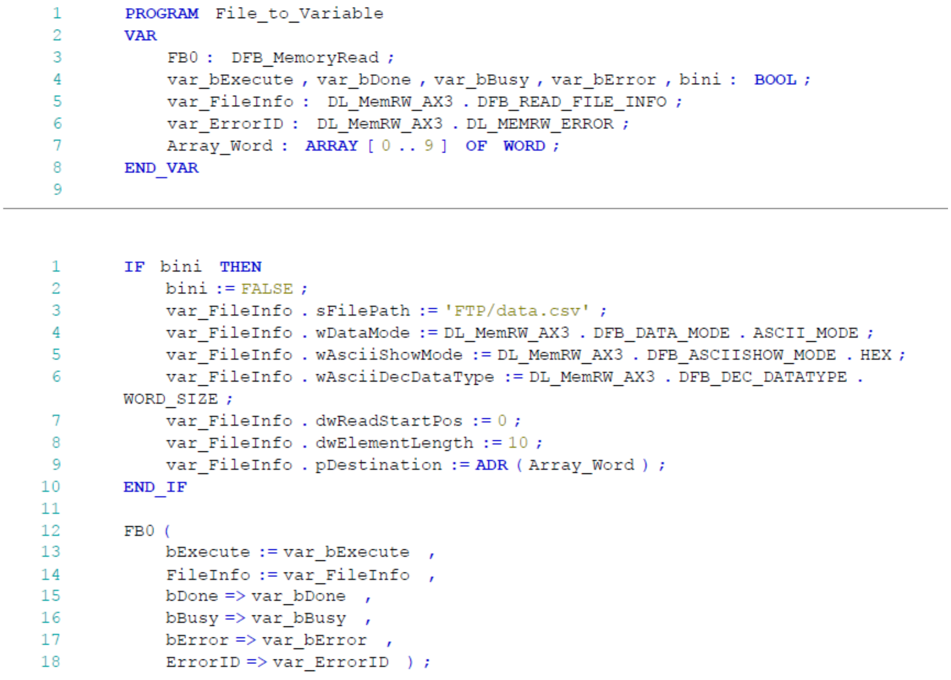
```
PROGRAM  FTP_Client
VAR
    FB0 :  DFB_FTPClient ;
    bini , var_bConnect , var_bExecute , var_bConnected , var_bDone , var_bBusy ,
var_bError :  BOOL ;
    var_fileSpace :  DL_FtpClient . MEM_SPACE ;
    var_ftpCmd :  DL_FtpClient . FTP_CMD ;
    var_sSrvFilePath , var_sLocalFilePath :  STRING ( 128 ) ;
    var_sFileName :  STRING ( 64 ) ;
    var_connectInfo :  DL_FtpClient . CONN_INFO ;
    var_ErrorId :  DL_FtpClient . DFB_FTP_CLIENT_ERROR ;
END_VAR
```



```
IF  bini  THEN
    bini := FALSE ;
    var_bConnect := FALSE ;
    var_bExecute := FALSE ;
    var_fileSpace :=  DL_FtpClient . MEM_SPACE . EXTERNAL_SD ;
    var_connectInfo . ftpServerIP := '192.168.1.6' ;
    var_connectInfo . activeMode := FALSE ;
    var_connectInfo . ftpUsrName := 'Admin' ;
    var_connectInfo . ftpPassWord := 'Delta' ;
    var_ftpCmd := DL_FtpClient . FTP_CMD . GET_FILE ;
    var_sSrvFilePath := '' ;
    var_sLocalFilePath := '' ;
    var_sFileName := 'data.csv' ;
    var_bConnect := TRUE ;
END_IF

FB0 (
    bConnect :=  var_bConnect ,
    bExecute :=  var_bExecute ,
    fileSpace :=  var_fileSpace ,
    connectInfo :=  var_connectInfo ,
    ftpCmd :=  var_ftpCmd ,
    sSrvFilePath :=  var_sSrvFilePath ,
    sLocalFilePath :=  var_sLocalFilePath ,
    sFileName :=  var_sFileName ,
    bConnected =>  var_bConnected ,
    bDone =>  var_bDone ,
    bBusy =>  var_bBusy ,
    bError =>  var_bError ,
    ErrorId =>  var_ErrorId ,
    msgBuf =>  ) ;
```

This example will export the file downloaded from the FTP server to the variable. When the user sets bini to True, the memory card reading Instruction setting is completed. After the setting is completed, Var_bExecute is set to True, that is, the data in the data.csv file Export the array "Array_Word".

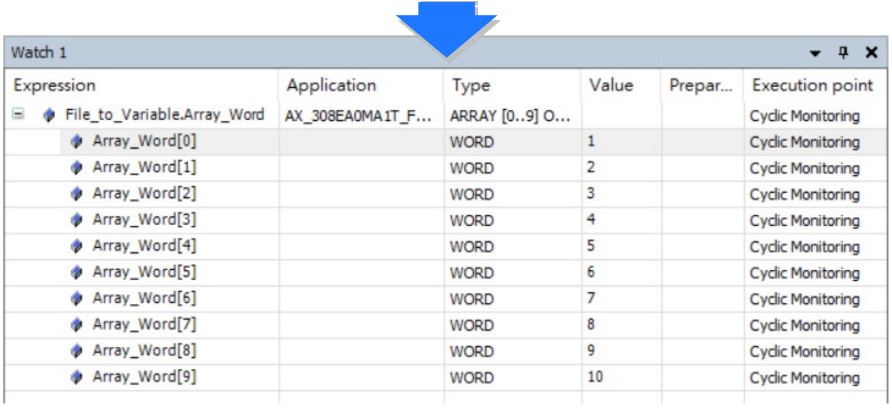
Note: Please refer to Chapter 9.1 "DFB_MemoryRead" for the instruction of reading the memory card.



```
PROGRAM  File_to_Variable
VAR
    FB0 :  DFB_MemoryRead ;
    var_bExecute , var_bDone , var_bBusy , var_bError , bini :  BOOL ;
    var_FileInfo :  DL_MemRW_AX3 . DFB_READ_FILE_INFO ;
    var_ErrorID :  DL_MemRW_AX3 . DL_MEMRW_ERROR ;
    Array_Word :  ARRAY [ 0 .. 9 ]  OF  WORD ;
END_VAR

```

```
IF  bini  THEN
    bini := FALSE ;
    var_FileInfo . sFilePath := 'FTP/data.csv' ;
    var_FileInfo . wDataMode := DL_MemRW_AX3 . DFB_DATA_MODE . ASCII_MODE ;
    var_FileInfo . wAsciiShowMode := DL_MemRW_AX3 . DFB_ASCIISHOW_MODE . HEX ;
    var_FileInfo . wAsciiDecDataType := DL_MemRW_AX3 . DFB_DEC_DATATYPE .
WORD_SIZE ;
    var_FileInfo . dwReadStartPos := 0 ;
    var_FileInfo . dwElementLength := 10 ;
    var_FileInfo . pDestination := ADR ( Array_Word ) ;
END_IF

FB0 (
    bExecute := var_bExecute  ,
    FileInfo := var_FileInfo  ,
    bDone => var_bDone  ,
    bBusy => var_bBusy  ,
    bError => var_bError  ,
    ErrorID => var_ErrorID  ) ;
```

data.csv									
1	2	3	4	5	6	7	8	9	10



Expression	Application	Type	Value	Prepar...	Execution point
⊟ File_to_Variable.Array_Word	AX_308EA0MA1T_F...	ARRAY [0..9] O...			Cyclic Monitoring
Array_Word[0]		WORD	1		Cyclic Monitoring
Array_Word[1]		WORD	2		Cyclic Monitoring
Array_Word[2]		WORD	3		Cyclic Monitoring
Array_Word[3]		WORD	4		Cyclic Monitoring
Array_Word[4]		WORD	5		Cyclic Monitoring
Array_Word[5]		WORD	6		Cyclic Monitoring
Array_Word[6]		WORD	7		Cyclic Monitoring
Array_Word[7]		WORD	8		Cyclic Monitoring
Array_Word[8]		WORD	9		Cyclic Monitoring
Array_Word[9]		WORD	10		Cyclic Monitoring

- **Library**
  - DL_FtpClient.library

11.4 Error Codes and Troubleshooting

- DL_LOGDMP_ERROR

Description	Cause of Error	Corrective Action
DFB_NO_ERR	No errors.	-
DFB_DMP_ERR_FAILED	Internal errors.	Please contact us directly
DFB_DMP_ERR_ PARAMETER	Invalid parameter inputs.	Please check if the input parameters are correct.
DFB_DMP_ERR_ NOTINITIALIZED	The instruction cannot be executed owning to the component has not been initialized.	Please reboot the controller.
DFB_DMP_ERR_ VERSION	Wrong version.	Please check if the firmware and the library version are supported.
DFB_DMP_ERR_ TIMEOUT	Operation timeout.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFB_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.
DFB_DMP_ERR_ PENDING	The program is pending for execution.	Please reboot the controller.
DFB_DMP_ERR_ NUMPENDING	Too many pending programs.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFB_DMP_ERR_ NOTIMPLEMENTED	The function does not exist.	Please check if the firmware and the library version are supported.
DFB_DMP_ERR_ INVALIDID	Incorrect ID.	Please check if the firmware and the library version are supported.
DFB_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.
DFB_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.。
DFB_DMP_ERR_NO_ OBJECT	The object does not exist.	Please check if the firmware and the library version are supported.
DFB_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.。
DFB_DMP_ERR_ DUPLICATE	Duplicate object name.	Please check if the firmware and the library version are supported.
DFB_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.。
DFB_DMP_ERR_INVALID_HANDLE	Invalid handle for the object.	Please check if the firmware and the library version are supported.
DFB_DMP_ERR_END_OF_OBJECT	The end of the object has been reached.	Please check if the firmware and the library version are supported.
DFB_DMP_ERR_NO_ CHANGE	No changes happened.	Please check if the firmware and the library version are supported.
DFB_DMP_ERR_INVALID_INTERFACE	Invalid or unknown interface	Please check if the firmware and the library version are supported.

Description	Cause of Error	Corrective Action
DFB_DMP_ERR_NOT_ SUPPORTED	The function is not supported.	Please check if the firmware and the library version are supported.
DFB_DMP_ERR_NO_ ACCESS_RIGHTS	No rights to access the operation.	Please check if the firmware and the library version are supported.
DFB_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.。
DFB_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.。
DFB_DMP_ERR_INVALID_SESSION_ID	Invalid online session ID.	Please log in again or reboot the controller.
DFB_DMP_ERR_ EXCEPTION	Exception occurs.	Please check the error log.
DFC_GETSIZE_ERR_FAILED	Internal errors.	Please contact us directly
DFC_ GETSIZE _ERR_ PARAMETER	Invalid parameter inputs.	Please check if the input parameters are correct.
DFC_ GETSIZE _ERR_ NOTINITIALIZED	The instruction cannot be executed owning to the component has not been initialized.	Please reboot the controller
DFC_ GETSIZE _ERR_ VERSION	Incorrect version.	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _ERR_ TIMEOUT	Operation timeout.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFC_ GETSIZE _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_ GETSIZE _ERR_ PENDING	The program is pending for execution.	Please reboot the controller
DFC_ GETSIZE _ERR_ NUMPENDING	Too many pending programs.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFC_ GETSIZE _ERR_ NOTIMPLEMENTED	The function does not exist.	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _ERR_ INVALIDID	Incorrect ID.	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _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_ GETSIZE _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_ GETSIZE _ERR_NO_ OBJECT	The object does not exist	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _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_ GETSIZE _ERR_ DUPLICATE	Duplicate object name	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _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.

Description	Cause of Error	Corrective Action
DFC_ GETSIZE _ERR_INVALID _HANDLE	Invalid handle for the object	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _ERR_END_OF _OBJECT	The end of the object has been reached.	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _ERR_NO_ CHANGE	No changes happened.	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _ERR_INVALID _INTERFACE	Invalid or unknown interface	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _ERR_NOT_ SUPPORTED	The function is not supported.	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _ERR_NO_ ACCESS_RIGHTS	No rights to access the operation.	Please check if the firmware and the library version are supported.
DFC_ GETSIZE _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_ GETSIZE _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_ GETSIZE _ERR_INVALID _SESSION_ID	Invalid online session ID.	Please log in again or reboot the controller.
DFC_ GETSIZE _ERR_ EXCEPTION	Exception occurs.	Please check the error log.

- DFB_FTP_CLIENT_ERROR

Description	Cause of Error	Corrective Action
DFB_NO_ERR	No errors.	-
DFB_SD_CARD_NOT_FOUND	No SD card	Please confirm whether the SD card is inserted into the SD card slot
DFB_CONNECT_TIMEOUT	Operation timeout.	Please reset the controller to default (Reset Origin). If the problem remains, please contact us directly.
DFB_INPUT_ERROR	Parameter error	Please check if the Function block parameters are correct.
DFB_NOT_CONNECT	Disconnected	Please confirm whether the idle time of the FTP client exceeds the timeout setting of the FTP server.
DFB_CMD_ERROR	FTP Client command execution error	Please confirm whether the FTP server supports the current command operation.
DFB_OPEN_FILE_ERROR	Open file error	Please confirm whether the file is normal.
DFB_WRITE_FILE_ERROR	Error writing to file	Please confirm whether the file is normal.
DFB_CREATE_FILE_ERROR	File creation error	Please confirm whether the file is normal.
DFB_READ_FILE_ERROR	Error reading file	Please confirm whether the file is normal.
DFB_NOT_SUPPORT_MEM	The selected memory space is not supported	Please select the correct memory space
DFB_LOGIN_FAIL	FTP server login failed	Please confirm that the login account password is correct.
DFB_UNDEFINE	Undefined or has not yet been executed.	Wait for the execution of FB instruction being completed.