

客服热线 400 - 820 - 9595

绵密网络 专业服务

中达电通已建立了 71 个分支机构及服务网点，并塑建训练有素的专业团队，提供客户最满意的服务，公司技术人员能在 2 小时内回应您的问题，并在 48 小时内提供所需服务。

上海 电话:(021)6301-2827	南昌 电话:(0791)8625-5010	合肥 电话:(0551)6281-6777	南京 电话:(025)8334-6585	杭州 电话:(0571)8882-0610
武汉 电话:(027)8544-8475	长沙 电话:(0731)8549-9156	南宁 电话:(0771)5879-599	厦门 电话:(0592)5313-601	广州 电话:(020)3879-2175
济南 电话:(0531)8690-7277	郑州 电话:(0371)6384-2772	北京 电话:(010)8225-3225	天津 电话:(022)2301-5082	太原 电话:(0351)4039-475
乌鲁木齐 电话:(0991)4678-141	西安 电话:(029)8836-0780	成都 电话:(028)8434-2075	重庆 电话:(023)8806-0306	哈尔滨 电话:(0451)5366-0643
沈阳 电话:(024)2334-16123	长春 电话:(0431)8892-5060			

AX 系列标准指令手册



AX 系列标准指令手册



中达电通股份有限公司

地址：上海市浦东新区民夏路238号
 邮编：201209
 电话：(021) 5863-5678
 传真：(021) 5863-0003
 网址：http://www.deltagreentech.com.cn



扫一扫，关注官方微信

AX-0329710-01
2020/10/20

中达电通公司版权所有
如有改动，恕不另行通知

www.deltaww.com



AX 系列标准指令手册

版本修订一览表

版本	变更内容	发行日期
第一版	第一版发行	2020/10/20

AX 系列标准指令手册

目录

前言

P.1 简介	II
P.1.1 适用产品	II
P.1.2 相关手册	II

第 1 章 数据搬移指令

1.1 DFC_NIBMOV	1-2
1.2 DFC_XCH	1-5
1.3 错误码与故障排除	1-7

第 2 章 比较操作指令

2.1 DFC_CMP	2-2
2.2 DFC_UCMP	2-4
2.3 DFC_LRCMP	2-6
2.4 DFC_ZCP	2-8
2.5 DFC_UZCP	2-10
2.6 DFC_LRZCP	2-12

第 3 章 计时计数指令

3.1 DFB_Capture	3-2
3.2 DFB_Compare	3-9
3.3 DFB_HCnt	3-13
3.4 DFB_HTmr	3-17
3.5 DFB_PresetValue	3-21
3.6 DFB_Sample	3-26
3.7 错误码与故障排除	3-31

第 4 章 EtherCAT 通讯指令

4.1 DFB_EcGetAllSlaveAddr	4-2
4.2 DFB_EcGetSlaveCount	4-6
4.3 DFB_EtherCATLink_Diag	4-10
4.4 DFB_GetAllIECATSlaveInfo	4-15
4.5 DFB_GetECATMasterError	4-20
4.6 DFB_GetECATMasterState	4-23
4.7 DFB_ResetECATMaster	4-27
4.8 DFB_ResetECATSlave	4-31
4.9 错误码与故障排除	4-36

第 5 章 检查码计算指令

5.1 DFC_LRC8	5-2
5.2 DFC_LRC16	5-4
5.3 DFC_LRC32	5-6
5.4 错误码与故障排除	5-8

第 6 章 模块读写指令

6.1 DFB_From	6-2
6.2 DFB_To	6-5
6.3 错误码与故障排除	6-8

第 7 章 串行通讯指令

7.1 DFB_COMRS	7-2
7.2 DFB_ModbusComChannel	7-7
7.3 DFB_ModbusRequest	7-10
7.4 DFB_ModbusRequest2	7-14
7.5 错误码与故障排除	7-18

第 8 章 网络通讯指令

8.1 DFB_TCP_Client	8-2
8.2 DFB_TCP_Server	8-8
8.3 DFB_UDP_Socket	8-14
8.4 DFB_ModbusTCPChannel	8-20
8.5 DFB_ModbusTCPRequest	8-23
8.6 错误码与故障排除	8-27

第 9 章 储存卡读写指令

9.1 DFB_MemoryRead	9-2
9.2 DFB_MemoryWrite	9-6
9.3 错误码与故障排除	9-10

第 10 章 其它指令

10.1 DFC_LogGetSize	10-2
10.2 DFB_LogDump	10-4
10.3 错误码与故障排除	10-7

MEMO



前言

目录

P.1	简介.....	II
P.1.1	适用产品.....	II
P.1.2	相关手册.....	II

P.1 简介

感谢您购买AX系列运动控制器，并采用我们为您提供的高端运动控制系统。AX系列采用CODESYS系统，涵盖可程序逻辑控制器 (PLC)、运动控制 (Motion Control)。

本手册介绍了台达自行开发相关应用指令集，方便客户进行PLC的应用程序开发。

P.1.1 适用产品

本手册适用或与以下产品相关：

- AX-3 系列

P.1.2 相关手册

AX系列运动控制器的相关手册组成如下：

1. DIADesigner-AX 使用手册

内容包含软件操作方式、程序编辑语言 (梯形图、顺序功能图、ST (Structured Text)、和功能块)、程序组织单元 (POU) 以及任务 (Task) 的概念、以及运动控制程序的编辑方式。

2. AX-3 系列操作手册

介绍运动控制系统概念、软硬件设定、软件操作简介、装置说明、运动控制程序架构、故障排除、模拟输入 / 输出模块、温度量测模块等等信息。

第1章 数据搬移指令

目录

1.1	DFC_NIBMOV	1-2
1.2	DFC_XCH	1-5
1.3	错误码与故障排除	1-7

1.1 DFC_NIBMOV

DFC_NIBMOV：数据搬移。

FB/FC	指令	指令图	ST 语法
FC	DFC_NIBMOV		<pre>DFC_NIBMOV(pSrc:= , wSrcStartPos:= , pDst:= , wDstStartPos:= , wNum:= , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
pSrc	来源变量内存地址	POINTER TO BYTE	内存地址 (0)
wSrcStartPos	来源变量搬移起始地址 (单位 : Nibble)	WORD*	正整数 (0)
pDst	目标变量内存地址	POINTER TO BYTE	内存地址 (0)
wDstStartPos	目标变量储存起始地址 (单位 : Nibble)	WORD*	正整数 (0)
wNum	所欲交换的数据长度 (单位 : Nibble)	WORD*	正整数 (0)

*注：输入参数可使用 BYTE 与 WORD 变量类型。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
DFC_NIBMOV	指令执行结果 (此参数为Return形式)	BOOL	True/False (False)
ErrorID	错误码	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

- 功能说明

当执行此 FC 指令后，即将变量一（pSrc）内容值复制至变量二（pDst），其数据复制的长度由输入参数 wNum（单位为 Nibble）决定。

- 范例程序

- 范例程序 1：

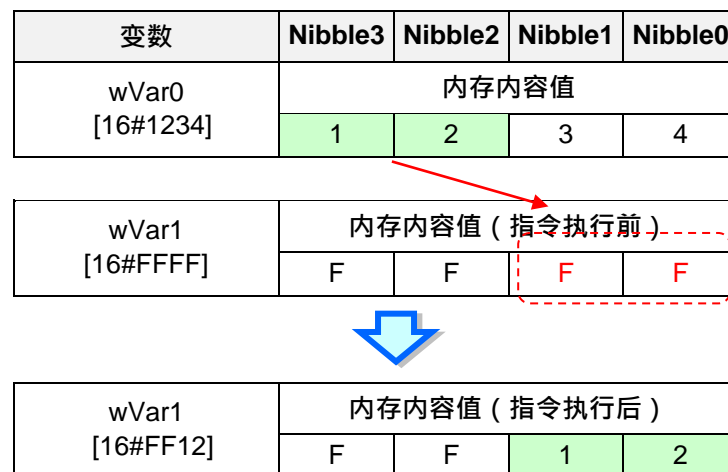
此范例利用 FC 指令（DFC_NIBMOV）将变量 wVar0（pSrc）内容值搬移至变量 wVar1（pDst）。

```

PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   bVar0: BOOL;
4   Length: WORD:=2;
5   wVar0: WORD := 16#1234;
6   wVar1: WORD := 16#FFFF;
7 END_VAR
8
9 IF bVar0 THEN
10  DFC_NIBMOV(
11    pSrc:=ADR(wVar0),
12    wSrcStartPos:=2,
13    pDst:=ADR(wVar1),
14    wDstStartPos:=0,
15    wNum:=Length,
16    ErrorID=>);
17  bVar0:=FALSE;
18 END_IF;

```

因 wSrcStartPos 为 2、wNum 为 2 且 wDstStartPos 为 0，则将变量 wVar0（pSrc）由 Nibble2 开始连续两个 Nibble（Length=2）搬移至变量 wVar1（pDst）内存内的 Nibble0 地址。



■ 范例程序 2 :

此范例利用 FC 指令(DFC_NIBMOV)将变量 ar_wVar0(pSrc)内容值搬移至变量 ar_wVar1(pDst)。

1

```

POU x
1 PROGRAM POU
2 VAR
3   bVar0: BOOL;
4   Length: WORD:=2;
5   ar_wVar0: ARRAY [0..1] OF WORD := [16#0123,16#4567];
6   ar_wVar1: ARRAY [0..1] OF WORD := [16#FFFF,16#FFFF];
7 END_VAR
8
9 IF bVar0 THEN
10   DFC_NIBMOV(
11     pSrc:= ADR(ar_wVar0),
12     wSrcStartPos:= 3,
13     pDst:= ADR(ar_wVar1),
14     wDstStartPos:= 0,
15     wNum:= Length,
16     ErrorID=> );
17   bVar0:=FALSE;
18 END_IF;
19

```

因 wSrcStartPos 为 3、wNum 为 2 且 wDstStartPos 为 0，则将变量 ar_wVar0 (pSrc) 由 Nibble3 开始连续两个 Nibble (Length=2) 搬移至变量 ar_wVar1 (pDst) 内存内的 Nibble0 地址。

变数	Nibble7	Nibble6	Nibble5	Nibble4	Nibble3	Nibble2	Nibble1	Nibble0
ar_wVar0 [16#0123,16#4567]	内存内容值							
	4	5	6	7	0	1	2	3

ar_wVar1 [16#FFFF,16#FFFF]	内存内容值 (指令执行前)							
	F	F	F	F	F	F	F	F



ar_wVar1 [16#FF70,16#FFFF]	内存内容值 (指令执行后)							
	F	F	F	F	F	F	7	0

- 支持机种
 - AX 系列

- 函式库
 - DL_Mov.library

1.2 DFC_XCH

DFC_XCH：两变数数据交换。

FB/FC	指令	指令图	ST 语法
FC	DFC_XCH		<pre>DFC_XCH(pSrc1:= , pSrc2:= , dwNum:= , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
pSrc1	变量一内存地址	POINTER TO BYTE	内存地址 (0)
pSrc2	变量二内存地址	POINTER TO BYTE	内存地址 (0)
dwNum	所欲交换的数据长度 (单位: Byte)	DWORD*	1 ~ 65535 正整数 (0)

*注: dNum 输入参数可使用 BYTE、WORD、DWORD 变量类型。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
DFC_XCH	指令执行结果 (此参数为Return形式)	BOOL	True/False (False)
ErrorID	错误码	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

● 功能说明

当执行此 FC 指令后，即将变量一 (pSrc1) 与变量二 (pSrc2) 内容值进行交换，其数据交换的长度由输入参数 dwNum 决定。

● 范例程序

此范例利用 FC 指令 (DFC_XCH) 将两变量内容值互相交换。

```

PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   bVar0,bVar1:BOOL;
4   Length:DWORD:=1;
5   wVar0:WORD:=16#1234;
6   wVar1:WORD:=16#5678;
7 END_VAR
1 IF bVar0 THEN
2   bVar1:=DFC_XCH(pSrc1:= ADR(wVar0), pSrc2:= ADR(wVar1) , dwNum:=Length , ErrorID=> )
3   bVar0:=FALSE;
4 END_IF;
5

```

因数据交换长度设定为一个 Byte (Length=1)，因此当执行 FC 指令 (DFC_XCH) 后会将变量一与变量二的 Low Byte 内容值进行交换 (如下表)。

指令执行前			指令执行后		
变数	wVar0	wVar1	变数	wVar0	wVar1
内容值	16#1234	16#5678	内容值	16#1278	16#5634

● 支持机种

- AX 系列

● 函式库

- DL_Mov.library

1.3 错误码与故障排除

说明	错误原因	处置方式
DFC_NIBMOV_ERR_PARAMETER	wNum 数值不正确	确认 wNum 值是否大于 0
DFC_XCH_ERR_PARAMETER	dwNum 数值不正确	确认 dwNum 值是否大于 0
DFC_XCH_ERR_NOMEMORY	控制器内存空间不足	检查下载的程序是否超出控制器所允许的大小并将控制器重启

MEMO

1

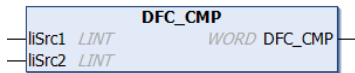
第2章 比较操作指令

目录

2.1	DFC_CMP	2-2
2.2	DFC_UCMP	2-4
2.3	DFC_LRCMP	2-6
2.4	DFC_ZCP	2-8
2.5	DFC_UZCP	2-10
2.6	DFC_LRZCP	2-12

2.1 DFC_CMP

DFC_CMP : LINT 变量比较。

FB/FC	指令	指令图	ST 语法
FC	DFC_CMP		<pre>DFC_CMP(liSrc1:= , liSrc2:=)</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
liSrc1	变量一	LINT*	LINT : $-2^{63} \sim 2^{63}-1$ (0)
liSrc2	变量二	LINT*	LINT : $-2^{63} \sim 2^{63}-1$ (0)

*注：输入参数可使用 SINT、INT、DINT 与 LINT 变量类型。

● 输出参数

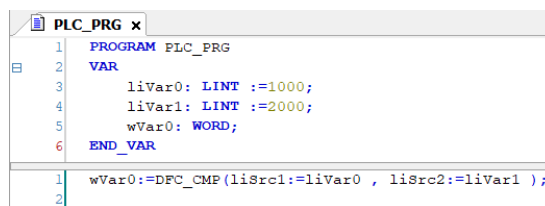
名称	功能	数据类型	输出值范围 (默认值)
DFC_CMP	指令执行结果 (此参数为Return形式)	WORD	1 : liSrc1 = liSrc2 2 : liSrc1 < liSrc2 3 : liSrc1 > liSrc2 (0)

● 功能说明

当执行此 FC 指令后，即对变量一 (liSrc1) 与变量二 (liSrc2) 内容值进行比较。

- 范例程序

此范例利用 FC 指令 (DFC_CMP) 将两变量内容值进行比较。



```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   liVar0: LINT :=1000;
4   liVar1: LINT :=2000;
5   wVar0: WORD;
6 END_VAR
1 wVar0:=DFC_CMP(liSrc1:=liVar0 , liSrc2:=liVar1 );
2
```

因变量一 (liVar0) 小于变量二 (liVar1)，则所计算的结果 (wVar0) 等于 2。

- 支持機種

- AX 系列

- 函式庫

- DL_Comparison.library

2.2 DFC_UCMP

DFC_UCMP : ULINT 变量比较。

FB/FC	指令	指令图	ST 语法
FC	DFC_UCMP		<pre>DFC_UCMP(lwSrc1:= , lwSrc2:=)</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
lwSrc1	变量一	ULINT/LWORD*	ULINT/LWORD : 0 ~ 2 ⁶⁴ -1 (0)
lwSrc2	变量二	ULINT/LWORD*	ULINT/LWORD : 0 ~ 2 ⁶⁴ -1 (0)

*注：输入参数可使用 USINT、UINT、UDINT、ULINT、BYTE、WORD、DWORD 与 LWORD 变量类型。

● 输出参数

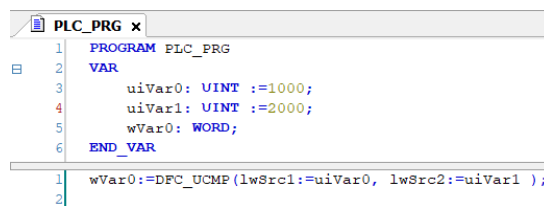
名称	功能	数据类型	输出值范围 (默认值)
DFC_UCMP	指令执行结果 (此参数为Return形式)	WORD	1 : lwSrc1 = lwSrc2 2 : lwSrc1 < lwSrc2 3 : lwSrc1 > lwSrc2 (0)

● 功能说明

当执行此 FC 指令后，即对变量一 (lwSrc1) 与变量二 (lwSrc2) 内容值进行比较。

- 范例程序

此范例利用 FC 指令 (DFC_UCMP) 将两变量内容值进行比较。



```
1 PROGRAM PLC_PRG
2 VAR
3   uiVar0: UINT :=1000;
4   uiVar1: UINT :=2000;
5   wVar0: WORD;
6 END_VAR
1 wVar0:=DFC_UCMP(lwSrc1:=uiVar0, lwSrc2:=uiVar1 );
2
```

因变量一 (uiVar0) 小于变量二 (uiVar1)，则所计算的结果 (wVar0) 等于 2。

- 支持机种

- AX 系列

- 函式库

- DL_Comparison.library

2.3 DFC_LRCMP

DFC_LRCMP : LREAL 变量比较。

FB/FC	指令	指令图	ST 语法
FC	DFC_LRCMP		<pre>DFC_LRCMP(lSrc1:= , lSrc2:=)</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
lSrc1	变量一	LREAL*	LREAL : -1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)
lSrc2	变量二	LREAL*	LREAL : -1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)

*注：输入参数可使用 REAL 与 LREAL 变量类型。

● 输出参数

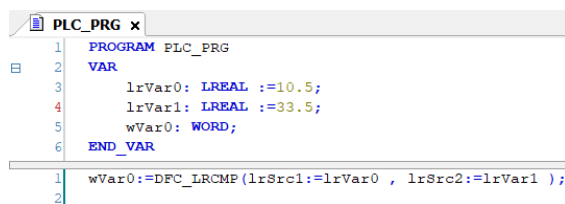
名称	功能	数据类型	输出值范围 (默认值)
DFC_LRCMP	指令执行结果 (此参数为Return形式)	WORD	1 : lSrc1 = lSrc2 2 : lSrc1 < lSrc2 3 : lSrc1 > lSrc2 (0)

● 功能说明

当执行此 FC 指令后，即对变量一 (lSrc1) 与变量二 (lSrc2) 内容值进行比较。

- 范例程序

此范例利用 FC 指令 (DFC_LRCMP) 将两变量内容值进行比较。



```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   lrVar0: LREAL :=10.5;
4   lrVar1: LREAL :=33.5;
5   wVar0: WORD;
6 END_VAR
1 wVar0:=DFC_LRCMP(lrSrc1:=lrVar0 , lrSrc2:=lrVar1 );
2
```

因变量一 (lrVar0) 小于变量二 (lrVar1)，则所计算的结果 (wVar0) 等于 2。

- 支持机种


- AX 系列

- 函式库

- DL_Comparison.library

2.4 DFC_ZCP

DFC_ZCP : LINT 变量区间比较。

FB/FC	指令	指令图	ST 语法
FC	DFC_ZCP		<pre>DFC_ZCP(liLowbound:= , liHighbound:= , liSrc:=);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
liLowbound	下限值	LINT*	LINT : $-2^{63} \sim 2^{63}-1$ (0)
liHighbound	上限值	LINT*	LINT : $-2^{63} \sim 2^{63}-1$ (0)
liSrc	变量	LINT*	LINT : $-2^{63} \sim 2^{63}-1$ (0)

*注：输入参数可使用 SINT、INT、DINT 与 LINT 变量类型。

● 输出参数

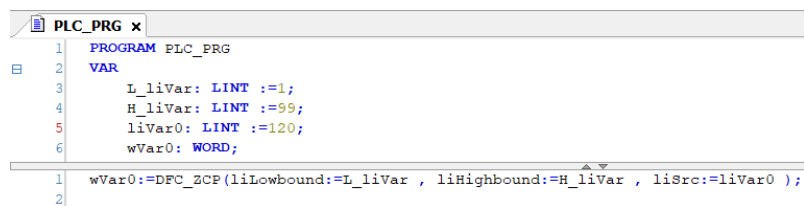
名称	功能	数据类型	输出值范围 (默认值)
DFC_ZCP	指令执行结果 (此参数为Return形式)	WORD	1 : liSrc < 下限值 2 : 下限值 < liSrc < 上限值 3 : liSrc > 上限值 (0)

● 功能说明

当执行此 FC 指令后，即对变量 (liSrc) 内容值与上限值和下限值进行比较。

- 范例程序

此范例利用 FC 指令 (DFC_ZCP) 将变量内容值与上限值和下限值进行比较。



```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   L_liVar: LINT :=1;
4   H_liVar: LINT :=99;
5   liVar0: LINT :=120;
6   wVar0: WORD;
1 wVar0:=DFC_ZCP(liLowbound:=L_liVar , liHighbound:=H_liVar , liSrc:=liVar0 );
2
```

因变量 (liVar0) 大于上限值 (H_liVar)，则所计算的结果 (wVar0) 等于 3。

- 支持机种

- AX 系列

- 函式库

- DL_Comparison.library

2.5 DFC_UZCP

DFC_UZCP : ULINT 变量区间比较。

FB/FC	指令	指令图	ST 语法
FC	DFC_UZCP		<pre>DFC_UZCP(lwLowbound:= , lwHighbound:= , lwSrc:=);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
lwLowbound	下限值	ULINT/LWORD*	ULINT/LWORD : 0 ~ 2 ⁶⁴ -1 (0)
lwHighbound	上限值	ULINT/LWORD*	ULINT/LWORD : 0 ~ 2 ⁶⁴ -1 (0)
lwSrc	变量	ULINT/LWORD*	ULINT/LWORD : 0 ~ 2 ⁶⁴ -1 (0)

*注：输入参数可使用 SINT、INT、DINT、LINT、BYTE、WORD、DWORD 与 LWORD 变量类型。

● 输出参数

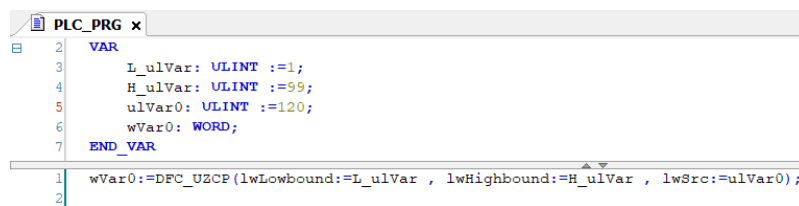
名称	功能	数据类型	输出值范围 (默认值)
DFC_UZCP	指令执行结果 (此参数为Return形式)	WORD	1 : lwSrc < 下限值 2 : 下限值 < lwSrc < 上限值 3 : lwSrc > 上限值 (0)

● 功能说明

当执行此 FC 指令后，即对变量 (lwSrc) 内容值与上限值和下限值进行比较。

- 范例程序

此范例利用 FC 指令 (DFC_UZCP) 将变量内容值与上限值和下限值进行比较。



```
PLC_PRG x
2  VAR
3    L_ulVar: ULINT :=1;
4    H_ulVar: ULINT :=99;
5    ulVar0: ULINT :=120;
6    wVar0: WORD;
7  END_VAR
1  wVar0:=DFC_UZCP(lwLowbound:=L_ulVar , lwHighbound:=H_ulVar , lwSrc:=ulVar0);
2
```

因变量 (ulVar0) 大于上限值 (H_ulVar)，则所计算的结果 (wVar0) 等于 3。

- 支持机种


- AX 系列

- 函式库

- DL_Comparison.library

2.6 DFC_LRZCP

DFC_LRZCP : LREAL 变量区间比较。

FB/FC	指令	指令图	ST 语法
FC	DFC_LRZCP		<pre>DFC_LRZCP(lrLowbound:= , lrHighbound:= , lrSrc:=);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
lrLowbound	下限值	LREAL*	LREAL : -1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)
lrHighbound	上限值	LREAL*	LREAL : -1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)
lrSrc	变量	LREAL*	LREAL : -1.7976931348623157E+308 ~ 1.7976931348623157E+308 (0)

*注：输入参数可使用 REAL 与 LREAL 变量类型。

● 输出参数

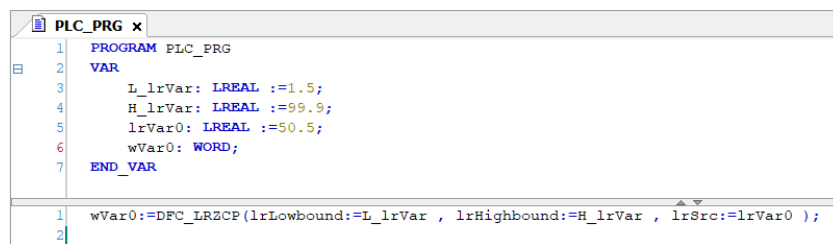
名称	功能	数据类型	输出值范围 (默认值)
DFC_LRZCP	指令执行结果 (此参数为Return形式)	WORD	1 : lrSrc < 下限值 2 : 下限值 < lrSrc < 上限值 3 : lrSrc > 上限值 (0)

● 功能说明

当执行此 FC 指令后，即对变量 (lrSrc) 内容值与上限值和下限值进行比较。

- 范例程序

此范例利用 FC 指令 (DFC_LRZCP) 将变量内容值与上限值和下限值进行比较。



```
1 PROGRAM PLC_PRG
2 VAR
3   L_lrVar: LREAL :=1.5;
4   H_lrVar: LREAL :=99.9;
5   lrVar0: LREAL :=50.5;
6   wVar0: WORD;
7 END_VAR

1 wVar0:=DFC_LRZCP(lrLowbound:=L_lrVar , lrHighbound:=H_lrVar , lrSrc:=lrVar0 );
2
```

因变量 (lrVar0) 小于上限值 (H_lrVar) 而且大于下限值 (L_lrVar), 则所计算的结果 (wVar0) 等于 2。

- 支持机种

- AX 系列

- 函式库

- DL_Comparison.library

MEMO

第3章 计时计数指令

目录

3.1	DFB_Capture	3-2
3.2	DFB_Compare	3-9
3.3	DFB_HCnt	3-13
3.4	DFB_HTmr	3-17
3.5	DFB_PresetValue.....	3-21
3.6	DFB_Sample.....	3-26
3.7	错误码与故障排除	3-31

3.1 DFB_Capture

DFB_Capture 根据指定的外部信号源触发并且捕获指定的高速计数器的命令值。

FB/FC	指令	指令图	ST 语法
FB	DFB_Capture	<pre> DFB_Capture Capture DFB_CAPTURE_REF Counter DFB_COUNTER_REF bEnable BOOL uiMaskValue UINT diDeltaMin DINT diDeltaMax DINT </pre>	<pre> DFB_Capture_instance(Capture :=, Counter :=, bEnable :=, uiMaskValue :=, diDeltaMin :=, diDeltaMax :=, bValid =>, bBusy =>, bError =>, ErrorID =>, bCapFlag =>, diCapValue =>, diCapValuePrevious =>, diDelta =>, bCapLenBeyondFlag =>, dwCapLenBeyondCount =>); </pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Counter	指定的高速计数器来源	DFB_COUNTER_REF ¹	DFB_COUNTER_REF (不能空值)	-
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-
uiMaskValue	指定捕捉器屏蔽范围	UINT	正数或 0 (0)	当 bEnable 上升沿时，会更新 uiMaskValue 的设定参数。
diDeltaMin	定义一个最小捕捉间距 ²	DINT	正数、负数或 0 (0)	当 bEnable 上升沿且 Busy 状态为 False
diDeltaMax	定义一个最大捕捉间距 ²	DINT	正数、负数或 0 (0)	当 bEnable 上升沿且 Busy 状态为 False

*注：

- DFB_Counter_REF (FB)：该功能块作为高速计数器的驱动接口，其中包含计数器的参数调用与驱动程序。
- diDeltaMin 与 diDeltaMax 都被设成 0，系统将不会确认捕捉间距是否在适当范围内。

● 输出参数

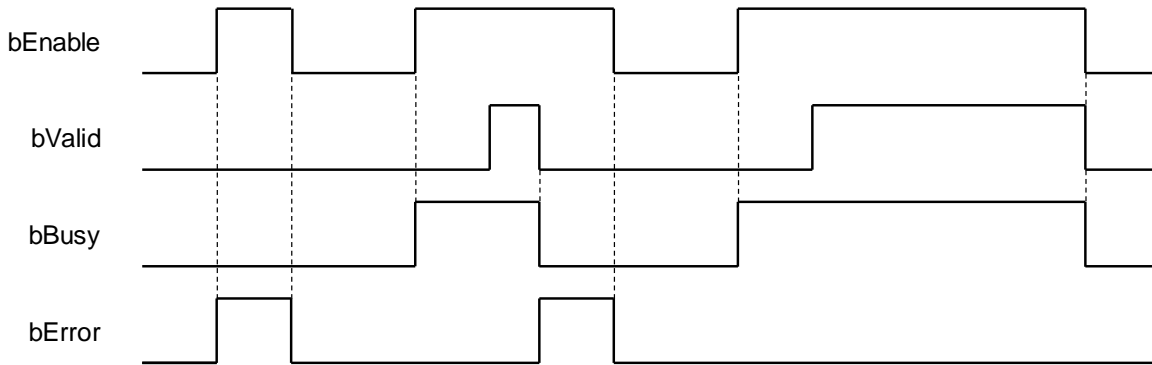
名称	功能	数据类型	输出值范围 (默认值)
bValid	当输出值有效时为 True	BOOL	True/False (False)
bBusy	当指令使能时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
bCapFlag	显示当前捕捉值为有效 (当此标志变为 True 以后, 只保持一个扫描周期之后便清除)	BOOL	True/False (False)
diCapValue	当前捕捉值	DINT	正数、负数或 0 (0)
diCapValuePrevious	前一次捕捉值	DINT	正数、负数或 0 (0)
diDelta	当前捕捉值与前一次捕捉值的差	DINT	正数、负数或 0 (0)
bCapLenBeyondFlag	捕捉间距异常检测 (当此标志变为 True 以后, 只保持一个扫描周期之后便清除)	BOOL	True/False (False)
dwCapLenBeyondCount	捕捉间距异常次数	DWORD	正数或 0 (0)

*注：DFB_HSIO_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bValid	<ul style="list-style-type: none"> 当 bEnable 上升沿之后一个程序扫描周期后，输出值有效时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bBusy	<ul style="list-style-type: none"> 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件或输入值发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时（清除 ErrorID 记录之错误码）
ErrorID		
bCapFlag	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新
diCapValue	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新
diCapValuePrevious	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新
diDelta	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新
bCapLenBeyondFlag	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新
dwCapLenBeyondCount	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新

● 参数时序图



● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Capture	指定的高速捕捉器来源	DFB_CAPTURE_REF (FB)*	DFB_CAPTURE_REF (不能空值)	当 bEnable 上升沿且 bBusy 状态为 False

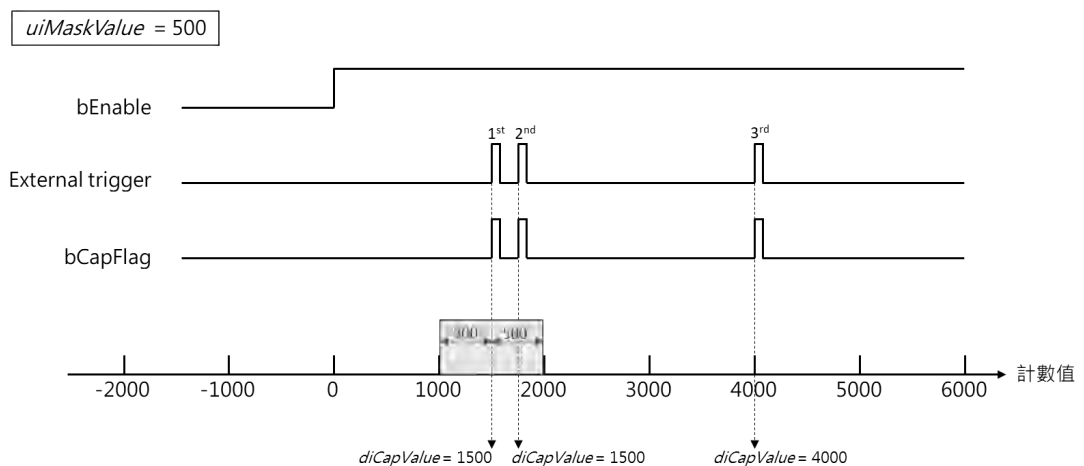
*注：DFB_CAPTURE_REF (FB)：该功能块作为高速捕捉器的驱动接口，其中包含捕捉器的参数调用与驱动程序。

- 功能说明

- **uiMaskValue**

输入参数 `uiMaskValue` 的功能说明请参考下图。

1. 指定 `uiMaskValue` 为 500，当 `bEnable` 被设为 `True`，捕捉功能启动，同时功能块参数 `bValid` 为 `true`，第一次捕捉值会被当成屏蔽的中心，当下一次捕捉值位在屏蔽范围内，则捕捉无效。
2. 在下图中，第一次的捕捉「1st」发生在 (-500~500) 时，捕捉值将会由 0 变成 1500。
3. 当捕捉值变为 1500，1500 将会变成新的屏蔽范围中心，因此捕捉值发生在 1000~2000 ($1000 < diCapValue < 2000$) 之内将视为无效捕捉值，所以当第二次的捕捉「2nd」被触发时（在屏蔽范围内），捕捉值将保持在 1500。
4. 当第三次的捕捉「3rd」在 1000 和 2000 之外被触发，则捕捉值将被更新为 4000。



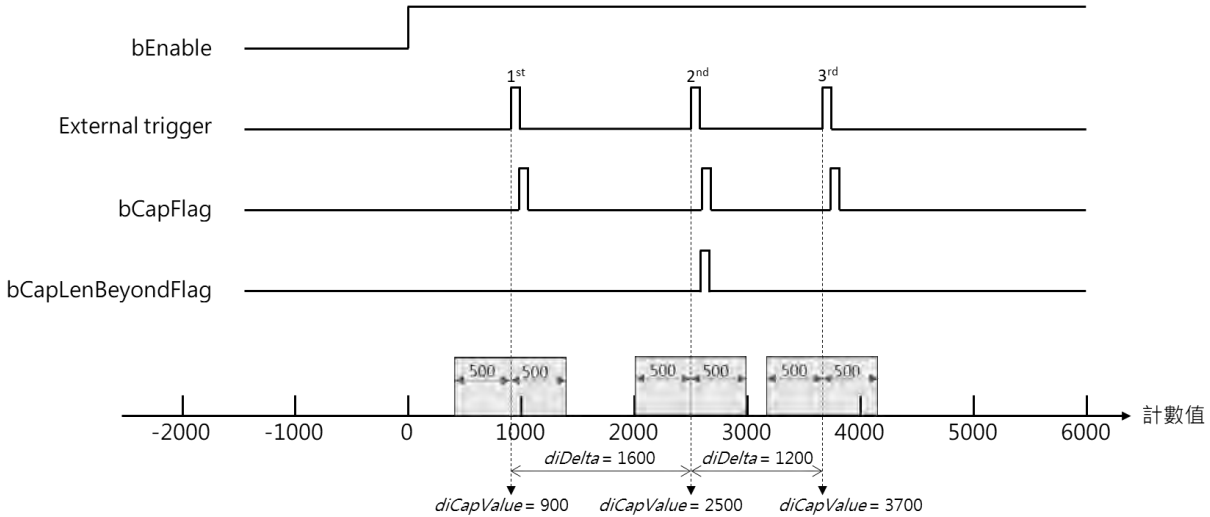
- **diDeltaMin · diDeltaMax · bCapLenBeyondFlag · dwCapLenBeyondCount**

`DeltaMin/DeltaMax` 定义最小 / 最大捕捉间距值；`CapLenBeyondFlag` 与 `CapLenBeyondCount` 为捕捉间距值异常标志与次数。

1. 在应用面上 `diDeltaMin` 与 `diDeltaMax` 常被视为捕捉信号的光标与光标允许的最小/最大范围，主要是用来判断是否出现漏光标未触发的设定值。例如：当 `diDeltaMin = 1000`；`diDeltaMax = 1300` 时，当光标与光标之间距（`diDelta`）触发大于 1300 或小于 1000 时，将视为漏目标情况进行处理。
2. 当出现漏目标情况时，`bCapLenBeyondFlag` 标志会执行 1 个扫描周期后自动复位，同时 `dwCapLenBeyondCount` 会自动累加数值 1 次。
3. 输入输出的说明请参考下列图示：
 - 设定屏蔽范围为 500 并且 1st 捕捉值为 900。
 - 2nd 捕捉值在 2500，由于设定 `diDeltaMin = 1000`，`diDeltaMax = 1300`，2nd (2500) 与 1st (900) 相差 1600，已超出捕捉区间范围 (1000~1300)，因此判定漏标并且 `bCapLenBeyondFlag` 会保持 `True` 为 1 个扫描周期。

- 3rd 捕捉值在 3700，与上一次捕捉值 (2nd = 2500) 相差 1200，捕捉值间距值符合设定范围 (1000 ~ 1300)，并且 3700 在屏蔽范围外，因此捕捉值更新为 3700 并且判断为非漏标捕捉，bCapLenBeyondFlag 保持 False 状态。

uiMaskValue = 500
diDeltaMin = 1100
diDeltaMax = 1300



● 故障排除

若指令执行中发生错误，且功能块输出参数 bError 将转为 True，捕捉功能将停止。可参考 ErrorID (错误码) 之内容，确认当前错误状态。

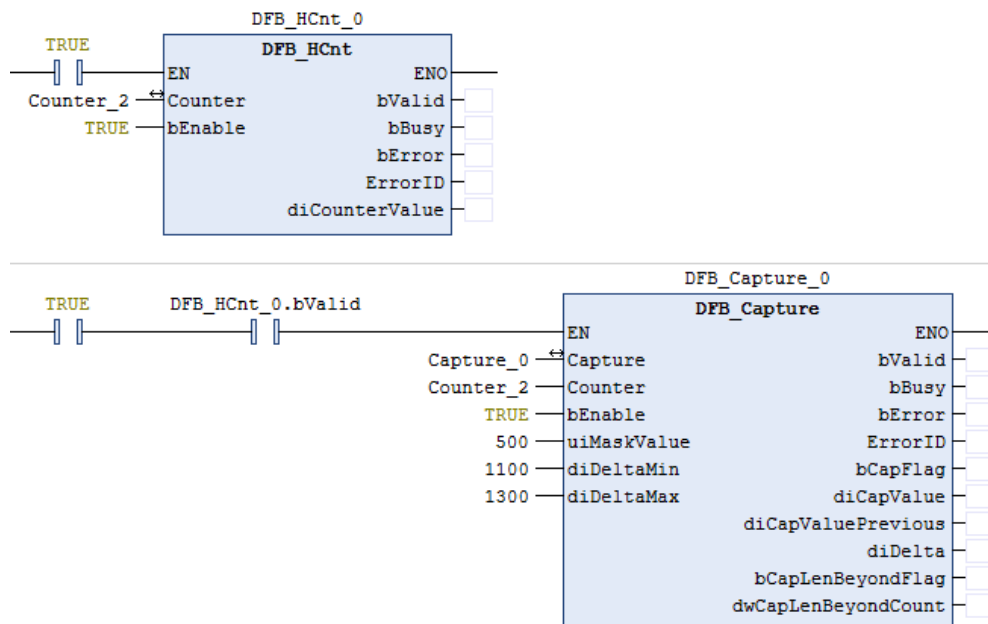
● 范例程序

此范例说明执行 DFB_HCnt 后再启动 DFB_Capture 来进行捕捉高速计数值之功能。

1. 如下图，在 BuiltIn_IO 中的 Hardware IO Configuration 勾选一组高计数器（Counter）与一组捕捉器（Capture），并将捕捉器的信号触发源选择硬件上的信号输入端点（e.g. IN15）。



2. 在 POU 中使用功能块 DFB_HCcnt 启动高速计数器（bEnable = True）后，再将 DFB_Capture 功能块启动（bEnable = True），如下图，此时等待外部信号（IN15）触发后，捕捉器将当下计数值记录在 DFB_Capture 功能块输出参数 diCapValue 上。




3. 有关 Hardware IO Configuration 设定与操作请参考 AX-3 系列操作手册。

- 支持机种
 - AX-308E

- 函式库
 - DL_BuiltInIO.library

3.2 DFB_Compare

DFB_Compare 比较指定的计数值和设定值，比较结果为 True 后可以输出或重置所需的装置。

FB/FC	指令	指令图	ST 语法
FB	DFB_Compare		<pre>DFB_Compare_instance(Compare :=, Counter :=, bEnable :=, Mode :=, wRefreshCycle :=, diCmpValue :=, bValid =>, bBusy =>, bError =>, ErrorID =>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Counter	指定的高速计数器来源	DFB_COUNTER_REF*1	DFB_COUNTER_REF (不能空值)	-
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-
Mode	指定比较条件	DFB_COMPARE_MODE*2	0 : Equal (=) 1 : Bigger_Equal (≥) 2 : Smaller_Equal (≤) (Equal)	当 bEnable 上升沿且 bBusy 状态为 False
wRefreshCycle	定义输出装置状态刷新时间	WORD	正数或 0 (0)	当 bEnable 上升沿且 bBusy 状态为 False
diCmpValue	指定与计数器比较值	DINT	正数、负数或 0 (0)	当 bEnable 上升沿且 bBusy 状态为 False

*注：

1. DFB_COUNTER_REF (FB)：该功能块作为高速计数器的驱动接口，其中包含计数器的参数调用与驱动程序。
2. DFB_COMPARE_MODE：枚举 (Enum)

● 输出参数

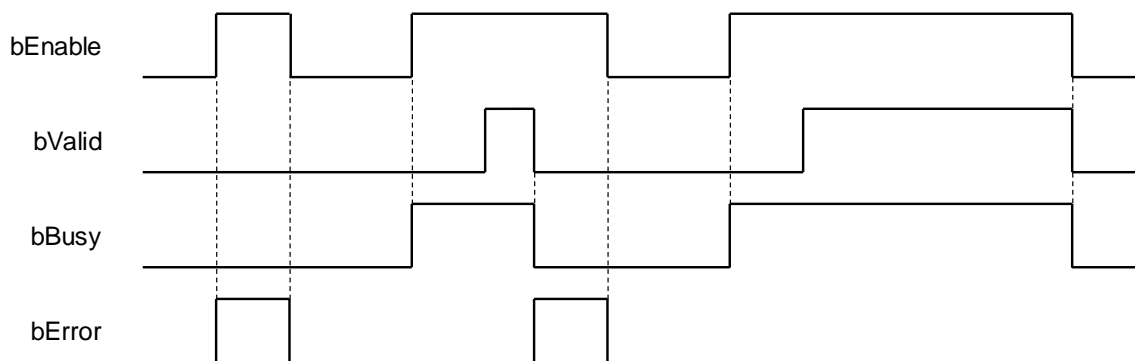
名称	功能	数据类型	输出值范围 (默认值)
bValid	当输出值有效时为 True	BOOL	True/False (False)
bBusy	当指令使能时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)

*注：DFB_HSIO_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bValid	<ul style="list-style-type: none"> 当 bEnable 上升沿之后，一个程序扫描周期后，输出值有效时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bBusy	<ul style="list-style-type: none"> 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件或输入值发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 (清除 ErrorID 记录之错误码)
ErrorID		

● 参数时序图



● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Compare	指定的高速比较器来源	DFB_COMPARE_REF (FB)*	DFB_COMPARE_REF (不能空值)	当 bEnable 上升沿且 bBusy 状态为 False

*注：DFB_COMPARE_REF (FB)：该功能块作为高速比较器的驱动接口，其中包含比较器的参数调用与驱动程序。

● 功能说明

1. 当比较条件成立时 (Counter Value = diCmpValue)，比较器会依 BuiltIn IO 中的 HW IO configuration 的 Compare 输出点做信号输出。
2. 当 DFB_Compare 输出参数 bValid = True 时，比较器将持续比较高速计数值，当比较条件成立并且依据设定输出装置的状态后，若比较条件再次成立但此时输出装置位于高准位时，输出装置保持高准位且不会重新输出一一次 (True → False → True)。若需要重置输出装置状态并将其准位由高变为低可以通过以下两种方法：

- ◆ 通过在 DIO 的 I/O mapping 中对应 Compare 的输出点定义变量，接着在 POU 程序区将此输出变量下降沿即可达到输出装置重置的效果。
- ◆ 通过功能块参数 wRefreshCycle 设定实现，此功能由 PLC 控制保持高准位一段时间后自动降到低准位。

以上两种方式都可以让输出装置由准位降至低准位，但若要再次让输出装置输出至高准位的话，需使计数器比较条件再次成立即可。

3. 通过功能块参数 wRefreshCycle 可以达到输出装置状态刷新重置。举例：设定 wRefreshCycle 值为 10000 (单位：0.1ms)，那比较器所指定的输出装置将会在比较条件成立后输出高准位 1 秒后由控制器将其拉低准位；如果将 wRefreshCycle 值设定为 0 时，则输出装置在比较条件成立后保持拉高准位，并不会将其重置。

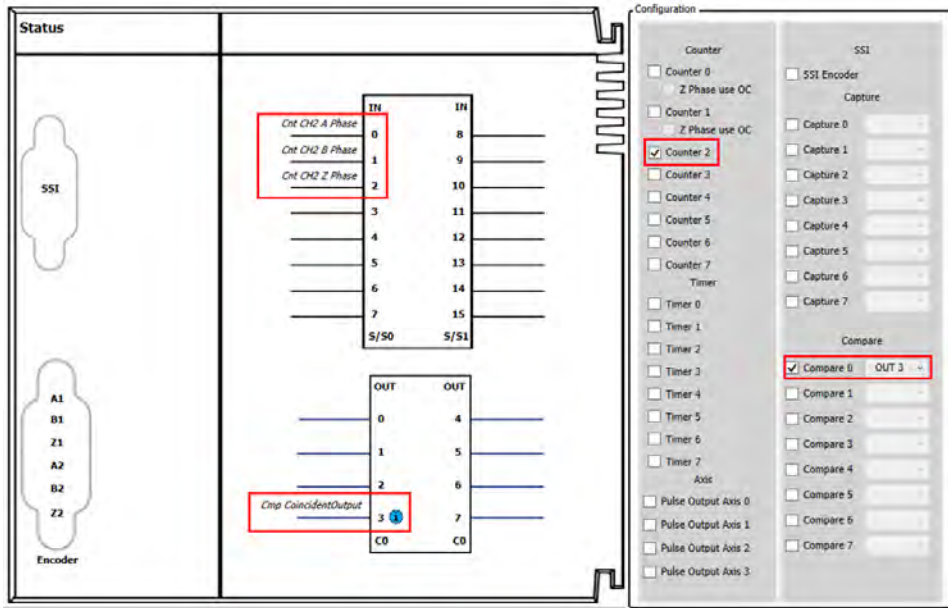
● 故障排除

若指令执行中发生错误，且功能块输出参数 bError 转为 True，捕捉功能将停止。可参考 ErrorID (错误码) 之内容，确认当前错误状态。

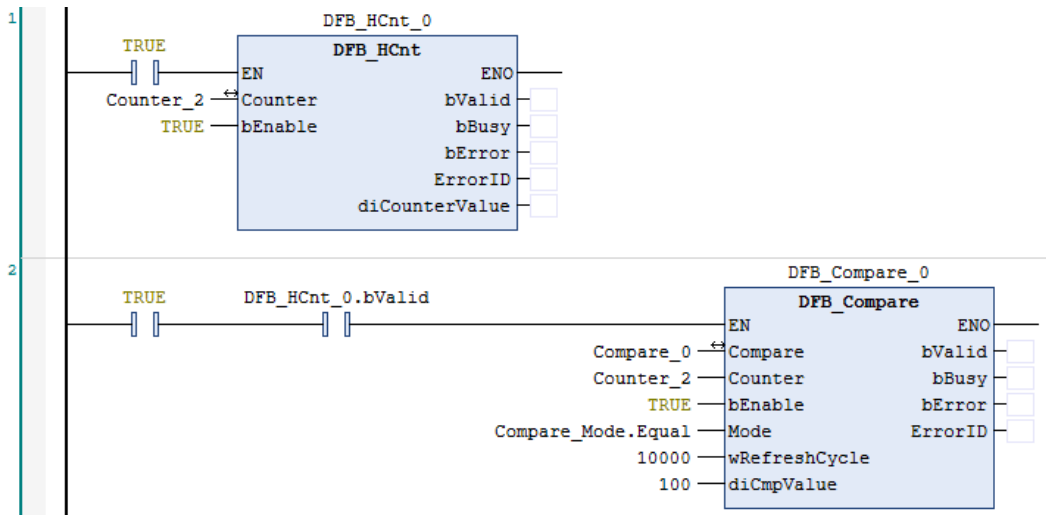
● 范例程序

此范例说明执行 DFB_HCnt 后再启动 DFB_Compare 来进行捕捉高速计数值之功能。

1. 如下图，在 BuiltIn_IO 中的 Hardware IO Configuration 勾选一组高计数器（Counter）与一组比较器（Compare），并将捕捉器的比较成立输出装置选择硬件上的信号输出端点（e.g. OUT3）。



2. 在 POU 中使用功能块 DFB_HCnt 启动高速计数器（bEnable = True）后，再将 DFB_Compare 功能块启动（bEnable = True），如下图，此时等待比较成立（DFB_HCnt_0.diCounterValue = DFB_Compare_0.diCmpValue）后，输出装置（OUT3）将输出信号。




3. 有关 Hardware IO Configuration 设定与操作请参考 AX-3 系列操作手册。

- 支持机种
 - AX-308E
- 函式库
 - DL_BuiltInIO.library

3.3 DFB_HCnt

DFB_HCnt 高速计数器计数功能与监控计数值功能。

FB/FC	指令	指令图	ST 语法
FB	DFB_HCnt		<pre>DFB_HCnt_instance(Counter :=, bEnable :=, bValid =>, bBusy =>, bError =>, ErrorID =>, diCounterValue =>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

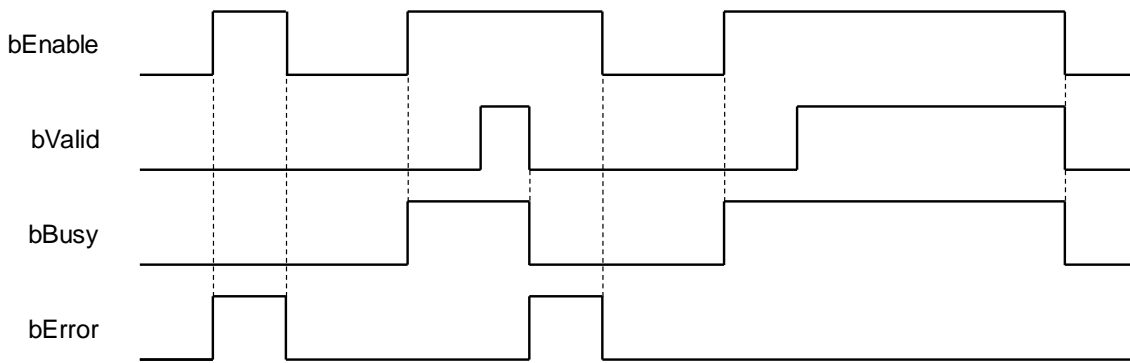
名称	功能	数据类型	输出值范围 (默认值)
bValid	当输出值有效时为 True	BOOL	True/False (False)
bBusy	当指令使能时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
diCounterValue	当前脉冲计数值	DINT	正数、负数或 0 (0)

*注：DFB_HSIO_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bValid	<ul style="list-style-type: none"> 当 bEnable 上升沿之后，一个程序扫描周期后，输出值有效时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bBusy	<ul style="list-style-type: none"> 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件或输入值发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时(清除 ErrorID 记录之错误码)
ErrorID		
diCounterValue	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新

● 参数时序图



● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Counter	指定的高速计数器来源	DFB_COUNTER_REF (FB)*	DFB_COUNTER_REF (不能空值)	当 bEnable 上升沿且 Busy 状态为 False

*注：DFB_COUNTER_REF (FB)：该功能块作为高速计数器的驱动接口，其中包含计数器的参数调用与驱动程序。

● 功能说明

1. 当功能块参数 bEnable 为 True 时，计数器会依 BuiltIn IO 中的 HW IO configuration 的 Counter 配置设定，开始向对应的输入硬件点位做脉冲计数，PLC 计数方式是依 BuiltIn IO 中的 Counter 装置设定页面勾选。
2. 可在计数过程中藉由 diCounterValue 得知计数值。

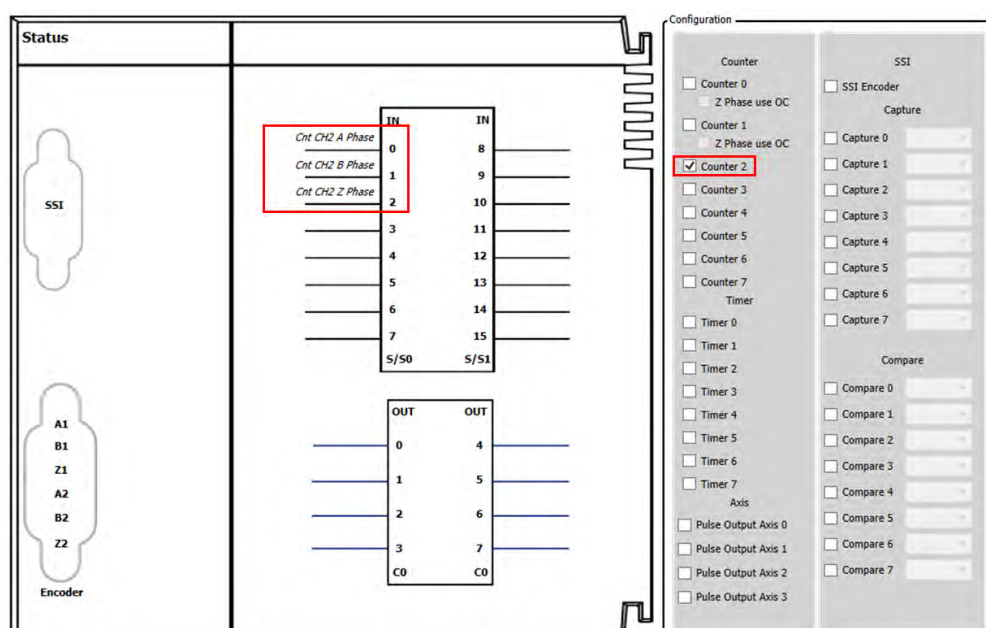
● 故障排除

若指令执行中发生错误，且功能块输出参数 `bError` 转为 `True`，捕捉功能将停止。可参考 `ErrorID` (错误码) 之内容，确认当前错误状态。

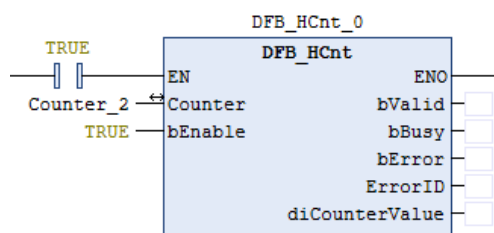
● 范例程序

此范例说明执行 `DFB_HCnt` 高速计数值之功能。

1. 如下图，在 `BuiltIn_IO` 中的 `Hardware IO Configuration` 勾选一组高计数器 (`Counter 2`)，并可以得知实体输入端口需要配置编码器的 A、B、Z 相的对应点位 (e.g. `IN 0`、`IN 1`、`IN 2`)，用户需将欲使用的脉冲信号输入源配线至此才能正常运用高速计数的功能。



2. 在 `POU` 中使用功能块 `DFB_HCnt` 启动高速计数器 (`bEnable = True`) 后，此时开始接收并计数外部信号 (`IN 0`、`IN 1`) 的脉冲，其计数模式依 `Counter Configuration` 里的模式设定，去计数脉冲值并将显示在 `diCounterValue` 上，用户需注意外部的信号来源所发送脉冲的模式要与计数模式匹配才能计数到正常的脉冲。




3. 有关 `Hardware IO Configuration` 与 `Counter Configuration` 里的设定请参考 `AX-3` 系列操作手册。

- 支持机种
 - AX-308E

- 函式库
 - DL_BuiltInIO.library

3.4 DFB_HTmr

DFB_HTmr 高速定时器频率波带宽功能与监控计数值功能。

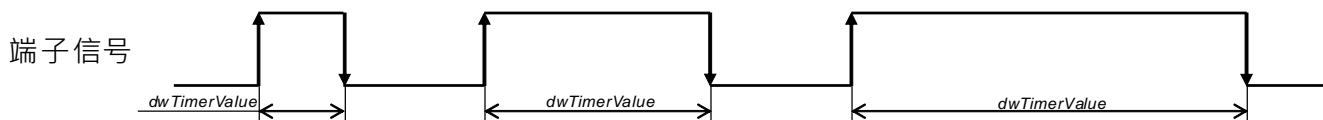
FB/FC	指令	指令图	ST 语法
FB	DFB_HTmr		<pre>DFB_HTmr_instance(Timer :=, bEnable :=, TriggerMode :=, bValid =>, bBusy =>, bError =>, ErrorID =>, dwTimerValue =>);</pre>

● 输入参数

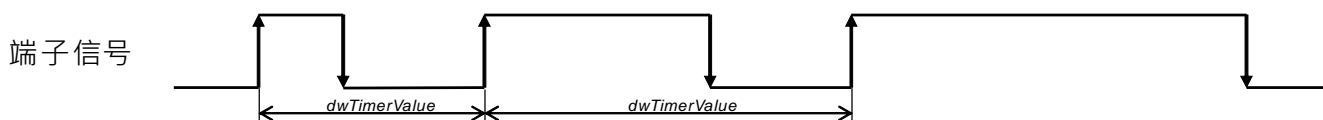
名称	功能	数据类型	设定值范围 (默认值)	生效时机
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-
TriggerMode	指定计时模式	DFB_TIMER_MODE *	0 : UP_DOWN 1 : UP_UP (UP_DOWN)	当 bEnable 上升沿且 bBusy 状态为 False

*注：DFB_TIMER_MODE：枚举（Enum）

上升-下降沿（Up - Down）模式：



上-上沿（Up - Up）模式：



● 输出参数

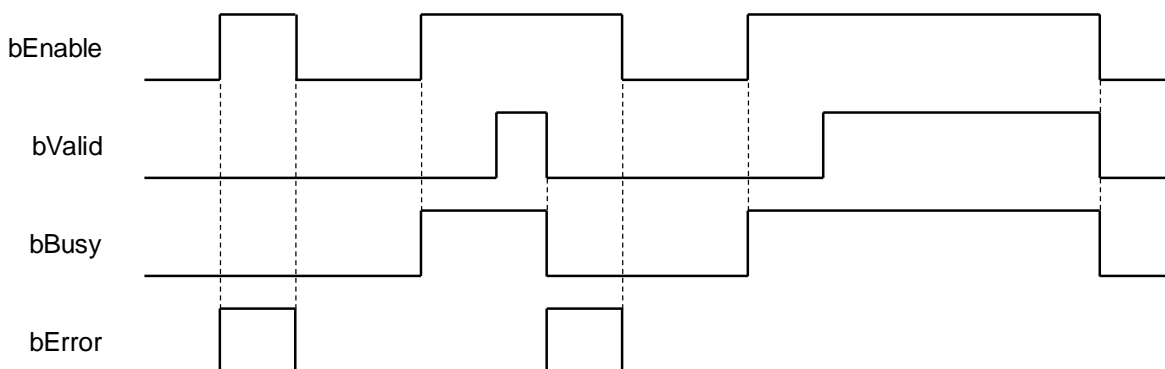
名称	功能	数据类型	输出值范围 (默认值)
bValid	当输出值有效时为 True	BOOL	True/False (False)
bBusy	当指令使能时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
dwTimerValue	量测当前脉冲计时值 (单位 : 0.1 μ s)	DWORD	正数或 0 (0)

*注 : DFB_HSIO_ERROR : 枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bValid	<ul style="list-style-type: none"> 当 bEnable 上升沿之后，一个程序扫描周期后，输出值有效时。 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bBusy	<ul style="list-style-type: none"> 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件或输入值发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时(清除 ErrorID 记录之错误码)
ErrorID		
dwTimerValue	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新

● 参数时序图



● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Timer	指定的高速 定时器来源	DFB_TIMER_REF (FB)*	DFB_TIMER_REF (不能空值)	当 bEnable 上升沿且 Busy 状态为 False

*注 : DFB_TIMER_REF (FB) : 该功能块作为高速定时器的驱动接口 , 其中包含定时器的参数调用与驱动程序。

● 功能说明

1. 当功能块参数 bEnable 为 True 时 , 定时器会依 BuiltIn IO 中的 HW IO configuration 的 Timer 配置设定开始向对应的输入硬件点位脉冲计数。
2. 可在计数过程中由 dwTimerValue 得知计时值。

● 故障排除

若指令执行中发生错误 , 且功能块输出参数 bError 将转为 True , 捕捉功能将停止。可参考 ErrorID (错误码) 之内容 , 确认当前错误状态。

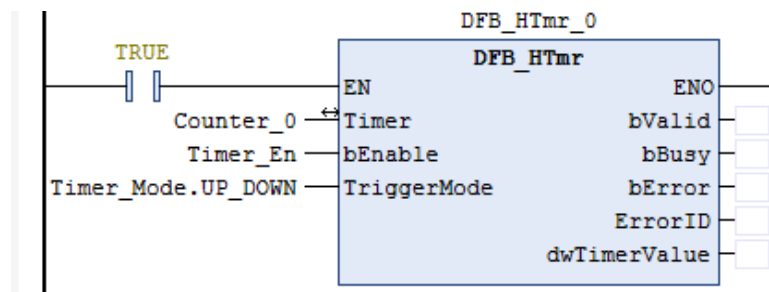
● 范例程序

此范例说明执行 DFB_HTmr 高速计数值之功能。

1. 如下图 , 在 BuiltIn_IO 中的 Hardware IO Configuration 勾选一组高速定时器 (Timer 2) , 并可以得知实体输入端口需要配置对应点位 (IN 0) , 用户需将欲使用的脉冲信号输入源配线至此才能正常运用高速计时的功能。



2. 在 POU 中使用功能块 DFB_HTmr 启动高速计数器 (bEnable = True) 后，此时开始接收并计数外部信号 (IN 0) 的脉冲，其计时模式依功能块设定去计数一个脉冲带宽的时间并将显示在 dwTimerValue 上。



3. 有关 Hardware IO Configuration 请参考 AX-3 系列操作手册。

- 支持机种
 - AX-308E
- 函式库
 - DL_BuiltInIO.library

3.5 DFB_PresetValue

DFB_PresetValue 是高速计数器的应用功能块，其作用是将默认值设定为当前的计数数值。

FB/FC	指令	指令图	ST 语法
FB	DFB_PresetValue		<pre>DFB_PresetValue_instance(Counter :=, bExecute :=, TriggerType :=, diPresetValue :=, bDone =>, bBusy =>, bCommandAborted =>, bError =>, ErrorID =>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bExecute	当 bExecute 上升沿时执行指令	BOOL	True/False (False)	-
TriggerType	指定默认计数值时机	DFB_PRESET_TRIGGER_TYPE *	0 : EXECUTE_TRIGGER 1 : EXTERNAL_TRIGGER (EXECUTE_TRIGGER)	当 bExecute 上升沿且 bBusy 状态为 False
diPresetValue	高速计数器的默认计数值	DINT	正数、负数或 0 (0)	当 bExecute 上升沿且 bBusy 状态为 False

*注：DFB_PRESET_TRIGGER_TYPE：枚举 (Enum)

- EXECUTE_TRIGGER：功能块参数 bExecute 触发后立即设定默认值
- EXTERNAL_TRIGGER：高数计数器外部信号触发后立即设定默认值

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	计数器默认值已变更完成	BOOL	True/False (False)
bBusy	当指令使能时为 True	BOOL	True/False (False)
bCommandAborted	当指令被中断且尚未完成时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)

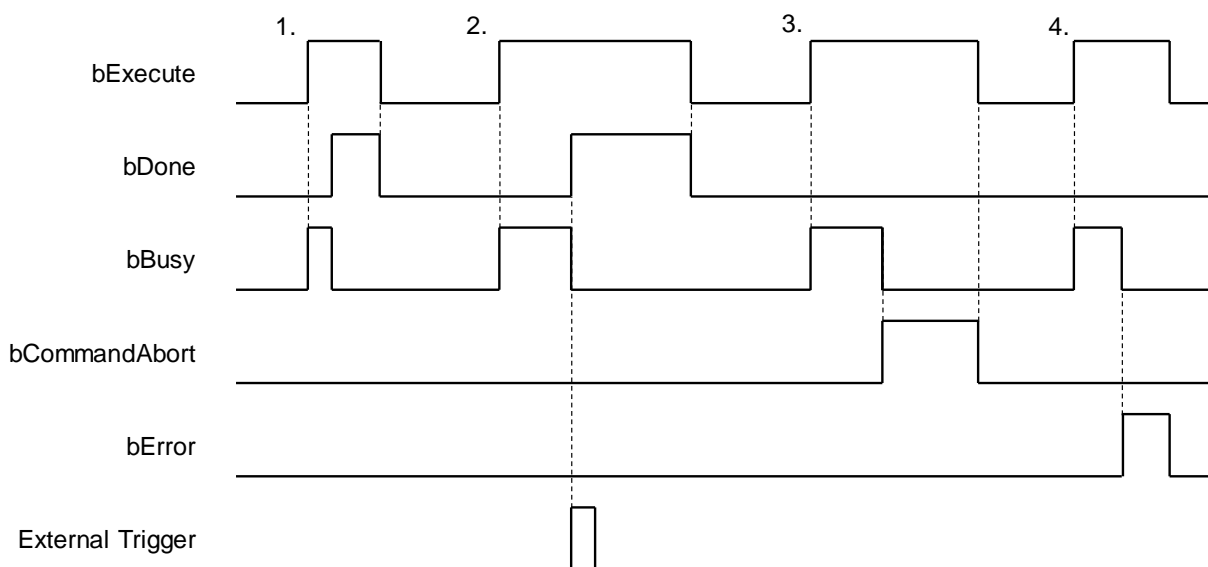
名称	功能	数据类型	输出值范围 (默认值)
ErrorID	错误码	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)

*注：DFB_HSIO_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	<ul style="list-style-type: none"> 当计数值已设为默认值时为 True 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 当 bError 上升沿时
bBusy	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 当 bError 上升沿时
bCommandAborted	<ul style="list-style-type: none"> 当功能块指令被中断时为 True 	<ul style="list-style-type: none"> 当 bExecute 下降沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件或输入值发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 (清除 ErrorID 记录之错误码)
ErrorID		

● 参数时序图



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

● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Counter	指定的高速计数器来源	DFB_COUNTER_REF (FB)*	DFB_COUNTER_REF (不能空值)	当 bExecute 上升沿且 bBusy 状态为 False

*注：DFB_COUNTER_REF (FB)：该功能块作为高速计数器的驱动接口，其中包含计数器的参数调用与驱动程序。

● 功能说明

1. 当功能块参数 TriggerType = EXECUTE_TRIGGER 时，功能块启动后立即将当下高速计数值变为默认值。
2. 当功能设定参数 TriggerType = EXTERNAL_TRIGGER 时，功能块启动后不会立即将当下高速计数值变更为默认值，需等到该计数器的 Z 相信号触发时，才会将计数值变更为默认值。

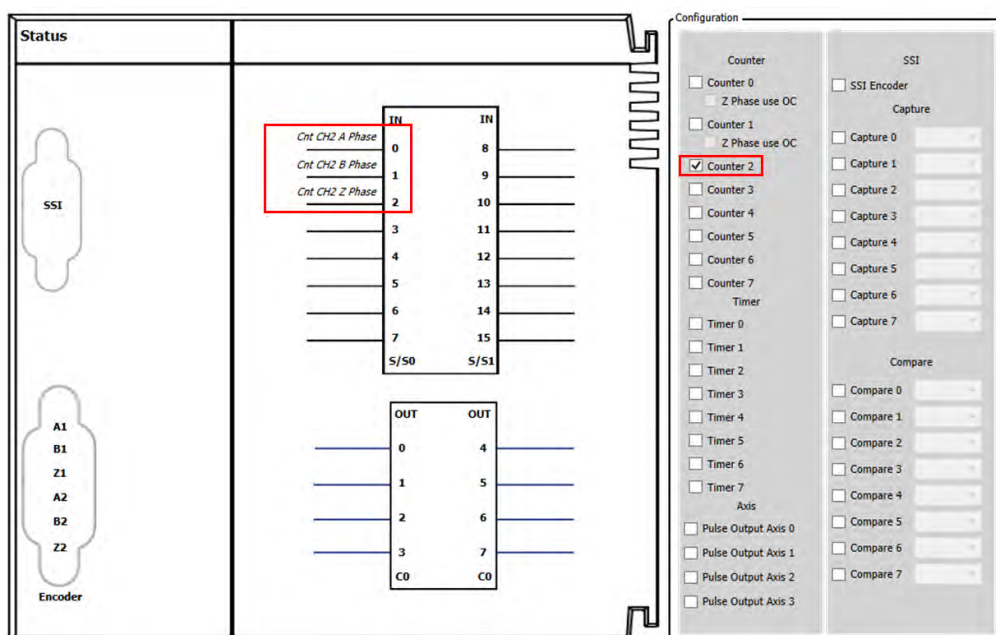
● 故障排除

若指令执行中发生错误，且功能块输出参数 bError 转为 True，捕捉功能将停止。可参考 ErrorID (错误码) 之内容，确认当前错误状态。

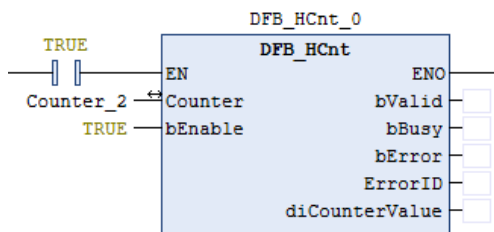
● 范例程序

此范例说明执行 DFB_HCnt 后再启动 DFB_PresetValue 来进行修改高速计数值之功能。

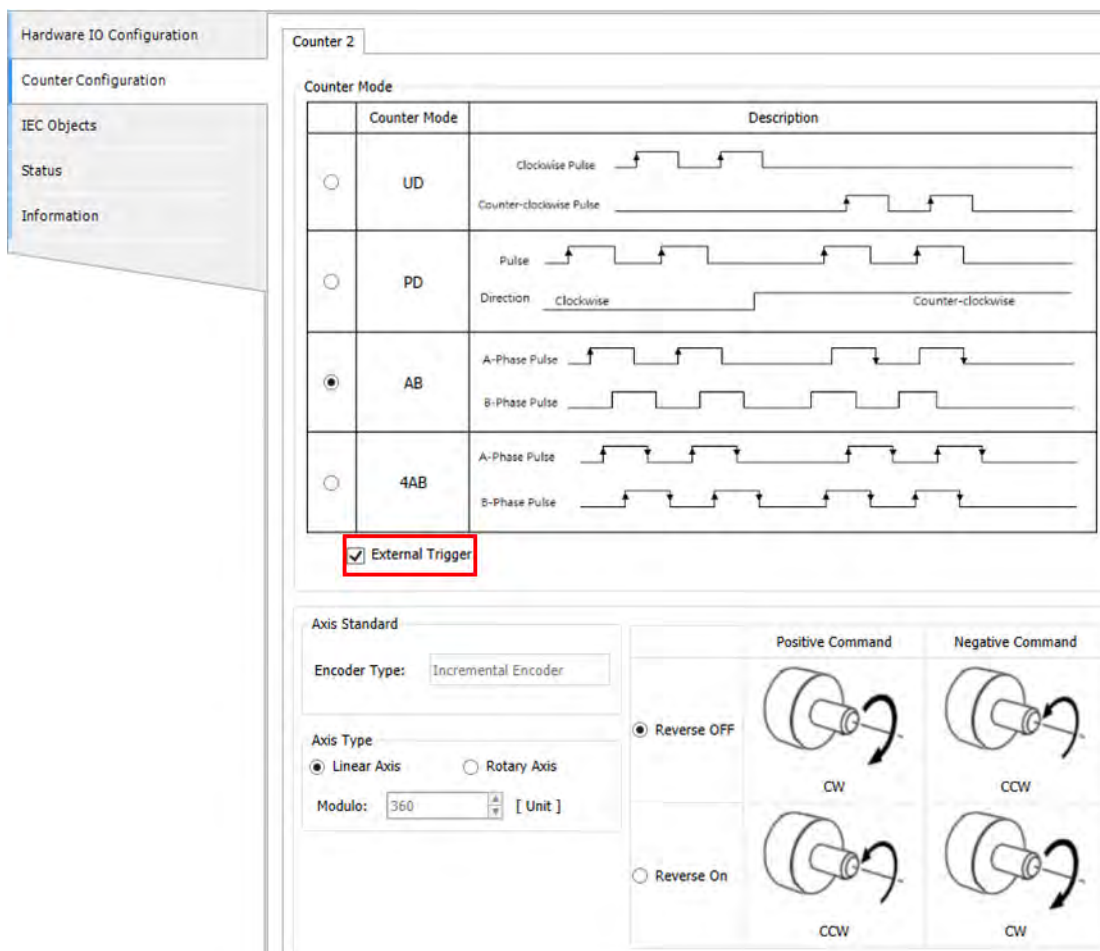
1. 如下图，在 BuiltIn_IO 中的 Hardware IO Configuration 勾选一组高计数器 (Counter 2)，并可以得知实体输入端口需要配置编码器的 A、B、Z 相的对应点位 (e.g. IN 0、IN 1、IN 2)，用户需将欲使用的脉冲信号输入源配线至此才能正常运用高速计数的功能。



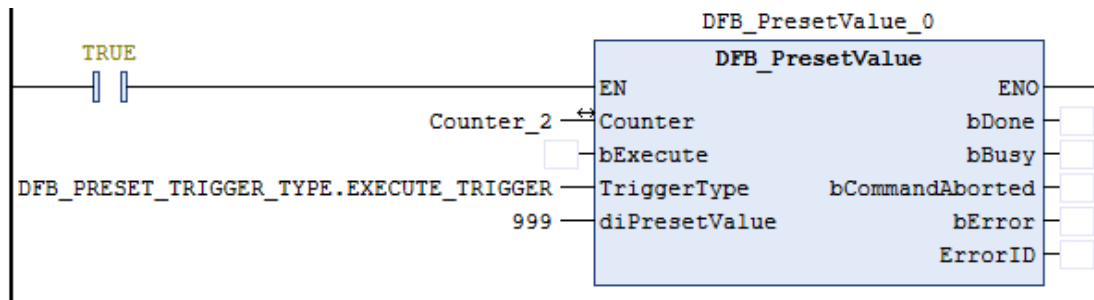
- 在 POU 中使用功能块 DFB_HCnt 启动高速计数器 (bEnable = True) 后，此时开始接收并计数外部信号 (IN 0、IN 1) 的脉冲，其计数模式依 Counter Configuration 里的模式设定，去计数脉冲值并将显示在 diCounterValue 上。用户需注意外部的信号来源所发送脉冲的模式要与计数模式匹配才能计数到正常的脉冲。



- 以外部信号触发作为默认值填入计数器为范例，需在 BuiltIn IO 中的 Counter Configuration 里，设定 External Trigger 为勾选表示启用，如下图红框处。



4. 承上述 3.，在 POU 中启动功能块 DFB_PresetValue 后 (bExecute = True)，此时功能块 DFB_PresetValue 将会等待该高速计数器的 Z 相信号触发默认值功能，当默认值设定完成且计数值等于默认值时 (DFB_HCnt.diCounterValue = DFB_PresetValue.diPresetValue)，DFB_PresetValue 功能块的参数 bDone 由 False 转为 True。




5. 有关 Counter Configuration 请参考 AX-3 系列操作手册。

- 支持機種
 - AX-308E
- 函式庫
 - DL_BuiltInIO.library

3.6 DFB_Sample

DFB_Sample 是高速计数器的应用功能块，其作用在设定的取样周期内，读取计数数值的增加或减少量。

FB/FC	指令	指令图	ST 语法
FB	DFB_Sample		<pre>DFB_Sample_instance(Counter :=, bEnable :=, wSampleTime :=, bValid =>, bBusy =>, bError =>, ErrorID => diSampleValue =>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-
wSampleTime	取样周期 (单位：1ms)	WORD	10 ~ 65535 (0)	当 bEnable 上升沿且 bBusy 状态为 False

● 输出参数

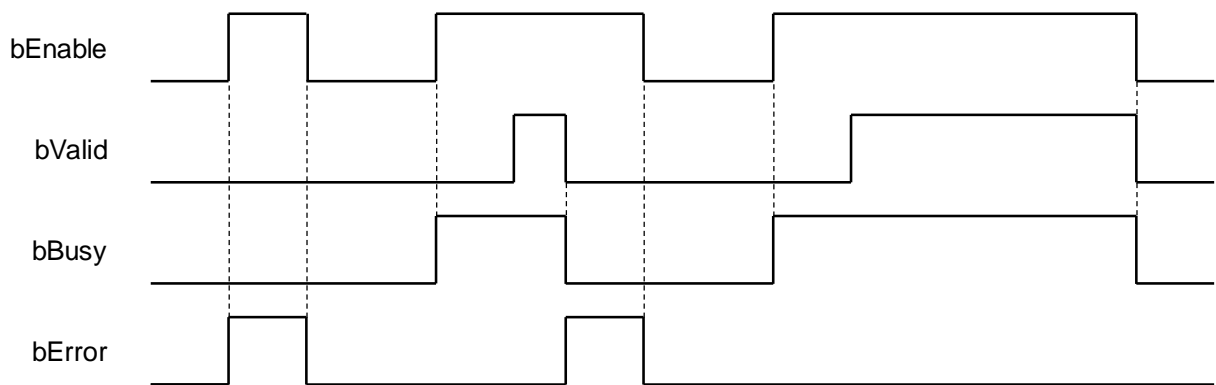
名称	功能	数据类型	输出值范围 (默认值)
bValid	当输出值有效时为 True	BOOL	True/False (False)
bBusy	当指令使能时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误代码	DFB_HSIO_ERROR*	DFB_HSIO_ERROR (DFB_HSIO_NO_ERR)
diSampleValue	每个取样周期的计数值增加量	DINT	正数、负数或 0 (0)

*注：DFB_HSIO_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bValid	<ul style="list-style-type: none"> 当 bEnable 上升沿之后一个程序扫描周期后，输出值有效时。 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bBusy	<ul style="list-style-type: none"> 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件或输入值发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 (清除 ErrorID 记录之错误码)
ErrorID		
diSampleValue	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新 	<ul style="list-style-type: none"> 当 bValid 为 True 时持续更新

● 参数时序图



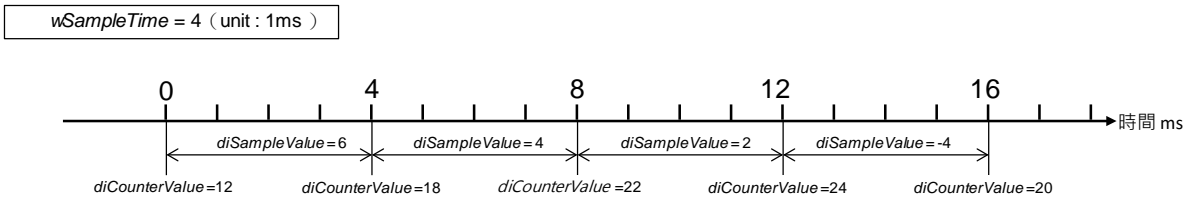
● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
Counter	指定的高速计数器来源	DFB_COUNTER_REF (FB)*	DFB_COUNTER_REF (不能空值)	当 bEnable 上升沿且 bBusy 状态为 False

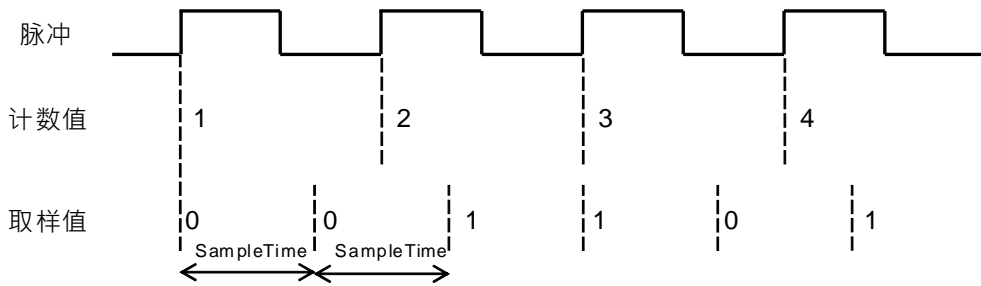
*注：DFB_COUNTER_REF (FB)：该功能块作为高速计数器的驱动接口，其中包含计数器的参数调用与驱动程序。

● 功能说明

1. 在指定的取样周期内 ($wSampleTime$)，功能块连续计数几个输入脉冲。



2. 当取样周期 $wSampleTime$ 小于脉冲的周期时，每个取样周期的计数值增加量 $diSampleValue$ 会 0 与 1 互相显示。



● 故障排除

若指令执行中发生错误，且功能块输出参数 $bError$ 转为 $True$ ，捕捉功能将停止。可参考 $ErrorID$ (错误码) 之内容，确认当前错误状态。

● 范例程序

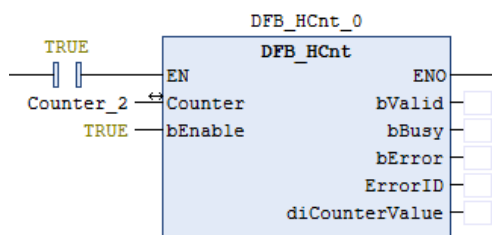
此范例说明执行 DFB_HCnt 后再启动 DFB_Sample 采样周期内的脉冲计数数量。

3

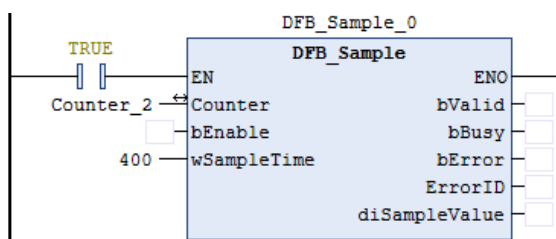
1. 如下图，在 BuiltIn_IO 中的 Hardware IO Configuration 勾选一组高计数器 (Counter 2)，并可以得知实体输入端口需要配置编码器的 A、B、Z 相的对应点位 (e.g. IN 0、IN 1、IN 2)，用户需将欲使用的脉冲信号输入源配线至此才能正常运用高速计数的功能。



2. 在 POU 中使用功能块 DFB_HCnt 启动高速计数器 (bEnable = True) 后，此时开始接收并计数外部信号 (IN 0、IN 1) 的脉冲，其计数模式依 Counter Configuration 里的模式设定，去计数脉冲值并将显示在 diCounterValue 上，用户需注意外部的信号来源所发送脉冲的模式要与计数模式匹配才能计数到正常的脉冲。



3. 在 POU 中启动功能块 DFB_Sample 后 (bEnable = True)，此时功能块 DFB_Sample 将会开始计算高速计数器在每一个取样周期内的脉冲计数增加量。



4. 有关 Counter Configuration 请参考 AX-3 系列操作手册。

- 支持機種
 - AX-308E
- 函式庫
 - DL_BuiltInIO.library

3.7 错误码与故障排除

下表列出运动功能块相对应错误码与处置方式。

说明	错误原因	处置方式
DFB_HSIO_NO_ERR	无错误信息	
DFB_CAP_INVALID_CAPTURE_REF	设定功能块输入参数类型的变量类型不是 Capture_REF	确认勾选 IO Configuration 上的 Capture 后，将 IEC Object 的变量输入至功能块 DFB_Capture 输入参数 "Capture"。
DFB_CAP_INVALID_COUNTER_REF	设定功能块输入参数类型的变量类型不是 Counter_REF	确认勾选 IO Configuration 上的 Counter 后，将 IEC Object 的变量输入至功能块 DFB_Capture 输入参数 "Counter"。
DFB_CAP_INVALID_VALUE_SETTING	DFB_Capture 的屏蔽设定值 (uiMaskValue) 超过旋转轴范围	将功能块参数 "uiMaskValue" 设定在编码器轴的旋转范围内 [0 ~ EncoderAxis.Modulo Value]
DFB_CAP_INVALID_DELTARANGE	当高速计数器的编码器轴是旋转轴时，输入参数 "diDeltaMax" 或 "diDeltaMin" 超过旋转轴范围。	将功能块参数 "diDeltaMax" 或 "diDeltaMin" 设定在编码器轴的旋转范围内 [0 ~ EncoderAxis.Modulo Value]
DFB_CAP_CAPTURE_ALREADY_ENABLE	高速捕捉装置已经被启用	确认此 Capture 装置是否被其它的 DFB_Capture 启用
DFB_CAP_DRIVE_ERROR	捕捉装置或计数装置驱动有发生错误	确认 BuiltIn_IO 上的错误信息，参考 AX-3 系列操作手册故障排除。
DFB_CMP_INVALID_COMPARE_REF	设定功能块输入参数类型的变量类型不是 Compare_REF	确认勾选 IO Configuration 上的 Compare 后，将 IEC Object 的变量输入至功能块 DFB_Compare 输入参数 "Compare"
DFB_CMP_INVALID_COUNTER_REF	设定功能块输入参数类型的变量类型不是 Counter_REF	确认勾选 IO Configuration 上的 Counter 后，将 IEC Object 的变量输入至功能块 DFB_Compare 输入参数 "Counter"
DFB_CMP_INVALID_CMPVALUE	当高速计数器的编码器轴是旋转轴时，输入参数 "diCompareValue" 超过旋转轴范围。	将功能块参数 "diCompareValue" 设定在编码器轴的旋转范围内 [0 ~ EncoderAxis.Modulo Value]
DFB_CMP_INVALID_REFRESHCYCLE	输入参数 "wRefreshCycle" 超过范围 [0-30000]，单位 0.1us	将功能块参数 "wRefreshCycle" 设定在范围 [0 ~ 30000]内
DFB_CMP_COMPARE_ALREADY_ENABLE	高速比较装置已经被启用	确认此 Compare 装置是否被其它的 DFB_Compare 启用
DFB_CMP_DRIVE_ERROR	比较装置或计数装置驱动有发生错误	确认 BuiltIn_IO 上的错误信息，参考 AX-3 系列操作手册故障排除。
DFB_HC_INVALID_COUNTER_REF	设定功能块输入参数类型的变量类型不是 Counter_REF	确认勾选 IO Configuration 上的 Counter 后，将 IEC Object 的变量输入至功能块 DFB_HCcnt 输入参数 "Counter"

说明	错误原因	处置方式
DFB_HC_COUNTER_ALREADY_ENABLE	高速计数装置已经被启用	确认此 Counter 装置是否被其它的 DFB_HCnt 启用
DFB_HC_COUNTER_REF_CHANGED_DURING_OPERATION	输入参数 "Counter" 在功能块运行时被修改替换	确认是否在 DFB_HCnt 功能块启动后有改变输入参数 Counter 的变量
DFB_HC_COUNTER_DRIVE_ERROR	计数装置驱动有发生错误	确认 BuiltIn_IO 上的错误信息，参考 AX-3 系列操作手册故障排除。
DFB_HT_INVALID_TIMER_REF	设定功能块输入参数类型的变量类型不是 Timer_REF	确认勾选 IO Configuration 上的 Timer 后，将 IEC Object 的变量输入至功能块 DFB_HTmr 输入参数 "Timer"
DFB_HT_TIMER_ALREADY_ENABLE	高速计时装置已经被启用	确认此 Timer 装置是否被其它的 DFB_HTmr 启用
DFB_HT_TIMER_REF_CHANGED_DURING_OPERATION	输入参数 "Timer" 在功能块运行时被修改替换	确认是否在 DFB_HTmr 功能块启动后有改变输入参数 Timer 的变量
DFB_HT_TIMER_DRIVE_ERROR	计时装置驱动有发生错误	确认 BuiltIn_IO 上的错误信息，参考 AX-3 系列操作手册故障排除。
DFB_PV_INVALID_COUNTER_REF	设定功能块输入参数类型的变量类型不是 Counter_REF	确认勾选 IO Configuration 上的 Counter 后，将 IEC Object 的变量输入至功能块 DFB_PresetValue 输入参数 "Counter"
DFB_PV_NOT_ENABLE_EXTERNAL_TRIGGER	计数器装置设定未勾选使用外部触发信号，但 DFB_PresetValue 模式设定 "EXTERNAL_TRIGGER"	请在 Counter 装置设定页面勾选 External Trigger
DFB_PV_PREVIOUS_PRESET_NOT_DONE	计数器的默认计数功能已被其它 DMC_PresetValue 功能块启用且尚未完成	请等 Counter 装置正在使用的 DFB_PresetValue 功能块作用完成后，再执行此功能块
DFB_PV_CANNOT_PRESET_WHEN_SAMPLING	计数器正在执行 DFB_Sample 功能	关闭此 Counter 装置的 Sample 功能，请将用此 Counter 的 DFB_Sample 关闭
DFB_PV_SETRING_NOT_DONE	计数器正在执行 DFB_SetRing 功能且尚未完成	请等 Counter 装置正在使用的 DFB_SetRing 功能块作用完成后，再执行此功能块
DFB_PV_INVALID_PRESET_VALUE	当高速计数器的编码器轴是旋转轴时，输入参数 "diPresetValue" 超过旋转轴范围。	将功能块参数 "diPresetValue" 设定在旋转轴范围 [0 ~ EncoderAxis.Modulo Value]
DFB_PV_COUNTER_REF_CHANGED_DURING_OPERATION	输入参数 "Counter" 在功能块运行时被修改替换	确认是否在 DFB_PresetValue 功能块启动后有改变输入参数 Counter 的变量
DFB_PV_COUNTER_DRIVE_ERROR	计数装置驱动有发生错误	确认 BuiltIn_IO 上的错误信息，参考 AX-3 系列操作手册故障排除。

说明	错误原因	处置方式
DFB_SP_INVALID_COUNTER_REF	设定功能块输入参数类型的变量类型不是 Counter_REF	确认勾选 IO Configuration 上的 Counter 后，将 IEC Object 的变量输入至功能块 DFB_Sample 输入参数 "Counter"
DFB_SP_COUNTER_NOT_ENABLE	DFB_Counter 尚未完成高速计数器使能	确认 Counter 装置已经被 DFB_HCnt 启动后，再使用此 DFB_Sample 功能块
DFB_SP_ALREADY_SAMPLING	计数器正在执行 DFB_Sample 功能	确认此 Counter 装置是否被其它的 DFB_Sample 启用
DFB_SP_PRESET_NOT_DONE	计数器正在执行 DFB_PresetValue 功能且尚未完成	请等 Counter 装置正在使用的 DFB_PresetValue 功能块作用完成后，再执行此功能块
DFB_SP_INVALID_SAMPLE_TIME	DFB_Sample 的输入参数 "wSampleTime" 超过范围 [10-65535]	将功能块参数 "wSampleTime" 设定在范围 [10 ~ 65535] 内
DFB_SP_COUNTER_REF_CHANGED_DURING_OPERATION	输入参数 "Counter" 在功能块运行时被修改替换	确认是否在 DFB_Sample 功能块启动后有改变输入参数 Counter 的变量
DFB_SP_COUNTER_DRIVE_ERROR	计数装置驱动有发生错误	确认 BuiltIn_IO 上的错误信息，参考 AX-3 系列操作手册故障排除。

MEMO


第4章 EtherCAT 通讯指令

目录

4.1	DFB_EcGetAllSlaveAddr	4-2
4.2	DFB_EcGetSlaveCount	4-6
4.3	DFB_EtherCATLink_Diag	4-10
4.4	DFB_GetAllIECATSlaveInfo	4-15
4.5	DFB_GetECATMasterError	4-20
4.6	DFB_GetECATMasterState	4-23
4.7	DFB_ResetECATMaster	4-27
4.8	DFB_ResetECATSlave	4-31
4.9	错误码与故障排除	4-36

4.1 DFB_EcGetAllSlaveAddr

DFB_EcGetAllSlaveAddr 取得所有从站地址。

FB/FC	指令	指令图
FB	DFB_EcGetAllSlaveAddr	
ST 语法		
<pre>DFB_EcGetAllSlaveAddr (bExecute :=, bDone =>, bBusy =>, bError =>, ErrorID =>, AddrArray =>, uSlaves =>,);</pre>		

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bExecute	当 bExecute 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

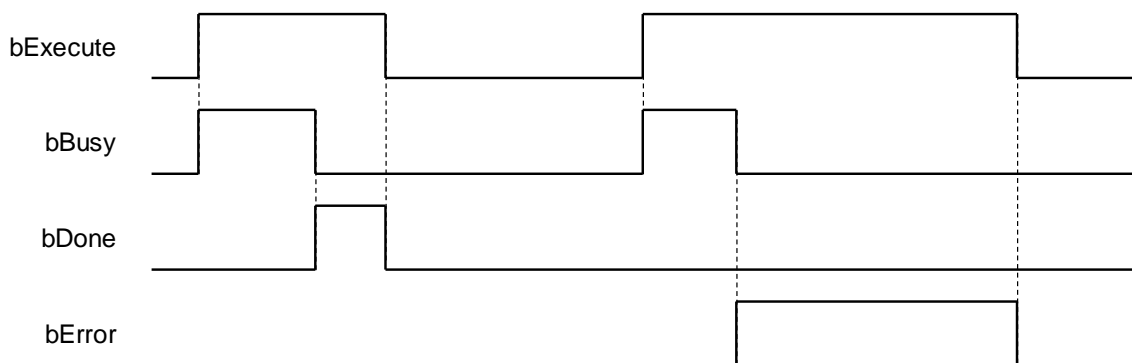
名称	功能	数据类型	输出值范围 (默认值)
bDone	功能块执行完毕	BOOL	True/False (False)
bBusy	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
AddrArray	从站地址数组	UINT[1..128]	(0)
uSlaves	从站数目	UINT	0~128 (0)

*注：DFB_ECAT_Diag_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	<ul style="list-style-type: none"> 当功能块执行完毕时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 若 bExecute 为 False 而 bDone 转为 True，此时 bDone 维持一个扫描周期的 True 状态后，立即转成 False。
bBusy	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bDone 上升沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 (清除 ErrorID 记录之错误码)
ErrorID		
AddrArray	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时
uSlaves	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时

● 参数时序图



● 功能说明

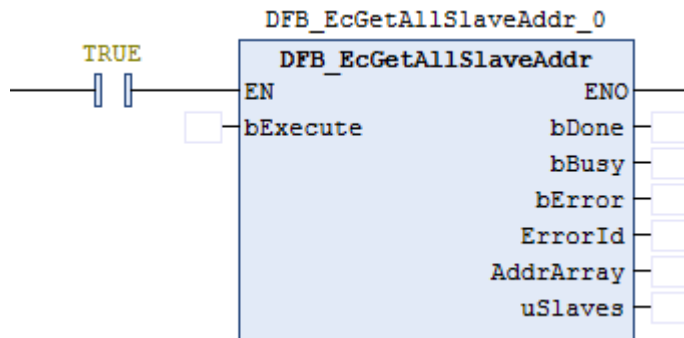
当 bExecute 上升沿时，AddrArray 参数输出装置树中所有 EtherCAT 从站地址，最大支持 128 站从站。uSlaves 参数输出 AddrArray 从站地址笔数，最大 128 笔。

● 故障排除

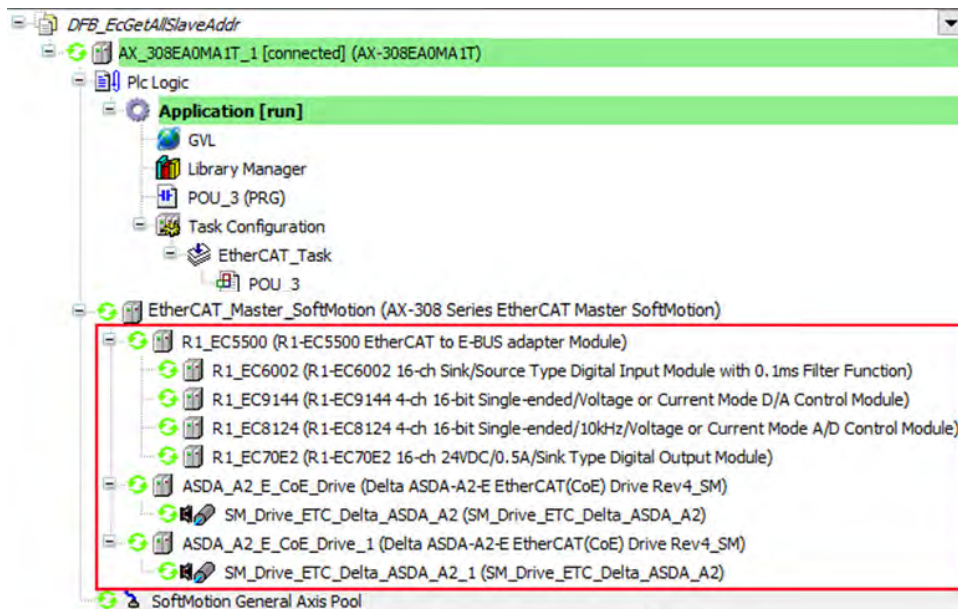
若指令执行中发生错误，此时 bError 将转为 True，可参考 ErrorID (错误码) 之内容，确认当前错误状态。

● 范例程序

此范例说明 DFB_EcGetAllSlaveAddr 的功能块行为。



1. EtherCAT_Master_SoftMotion 总共有七个 EtherCAT 从站



4

2. DFB_EcGetAllSlaveAddr bExecute 参数上升沿 · AddrArray 参数输出内容如下 · uSlaves 参数输出值为 7。

Expression	Type	Value
DFB_EcGetAllSlaveAddr_0	DFB_EcGetAllSlaveAddr	
bExecute	BOOL	TRUE
bDone	BOOL	TRUE
bBusy	BOOL	FALSE
bError	BOOL	FALSE
ErrorId	DFB_EC_CAT_DIAG_ERROR	DFB_EC_CAT_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

DFB_EcGetAllSlaveAddr_0

DFB_EcGetAllSlaveAddr

EN: TRUE

ENO: TRUE

bExecute: TRUE

bDone: TRUE

bBusy: FALSE

bError: FALSE

ErrorId: DFB_EC_CAT_D


AddrArray: [1001, 1002, 1003, 1004, 1005, 1006, 1007, 0]

uSlaves: 7

- 支持机种
 - AX-308E
- 函式库
 - DL_EtherCAT_Diag.library

4.2 DFB_EcGetSlaveCount

DFB_EcGetSlaveCount 取得装置树中从站数目。

FB/FC	指令	指令图
FB	DFB_EcGetSlaveCount	
ST 语法		
<pre>DFB_EcGetSlaveCount (bExecute :=, bDone =>, bBusy =>, bError =>, ErrorID =>, uSlaves =>,);</pre>		

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bExecute	当 bExecute 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	功能块执行完毕	BOOL	True/False (False)
bBusy	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
uSlaves	从站数目	UINT	0~128 (0)

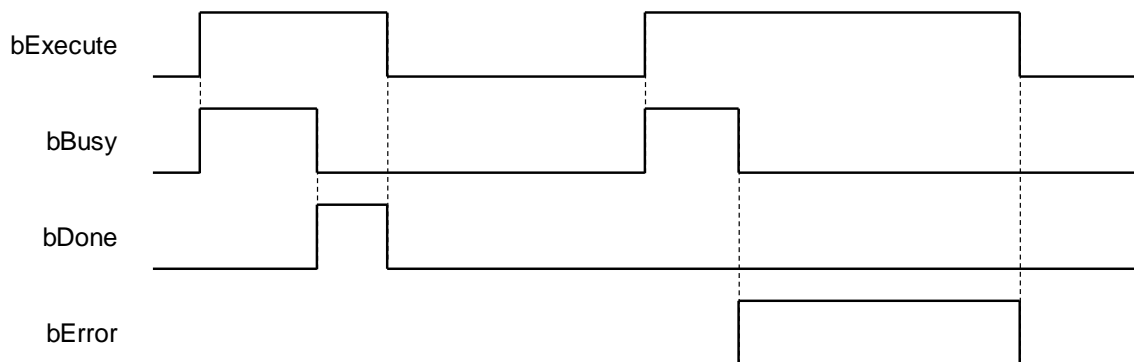
*注：DFB_ECAT_Diag_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	<ul style="list-style-type: none"> 当功能块执行完毕时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 若 bExecute 为 False 而 bDone 转为 True，此时 bDone 维持一个扫描周期的 True 状态后，立即转成 False。
bBusy	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bDone 上升沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 (清除 ErrorID 记录之错误码)
ErrorID		
uSlaves	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时

4

● 参数时序图



● 功能说明

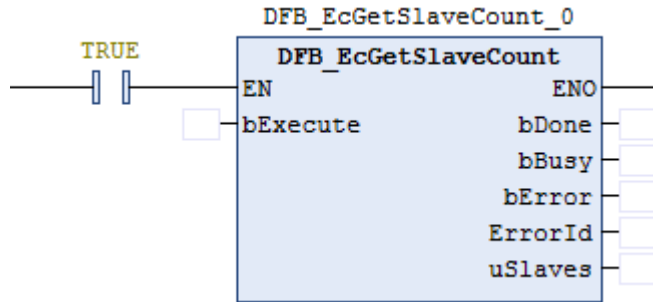
当 bExecute 上升沿时，uSlaves 参数输出装置树中 EtherCAT 从站数目。

● 故障排除

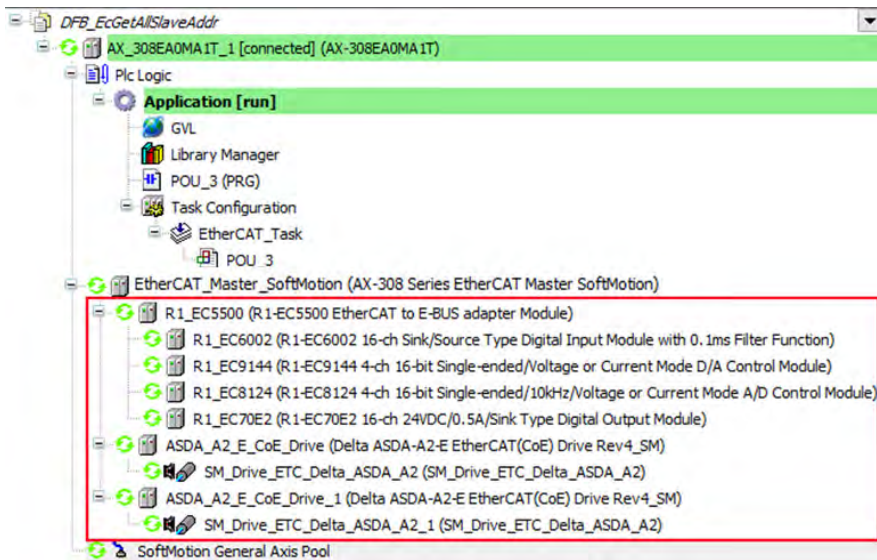
若指令执行中发生错误，此时 bError 将转为 True，可参考 ErrorID (错误码) 之内容，确认当前错误状态。

● 范例程序

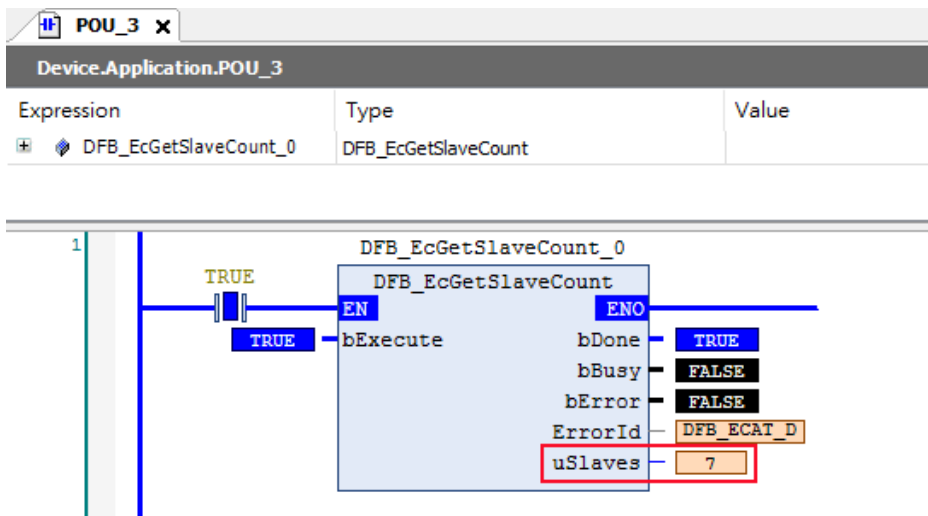
1. 此范例说明 DFB_EcGetSlaveCount 的功能块行为。



2. EtherCAT_Master_SoftMotion 总共有七个 EtherCAT 从站。



3. DFB_EcGetSlaveCount 的 bExecute 参数上升沿，uSlaves 参数输出值为 7。



- 支持机种
 - AX-308E

- 函式库
 - DL_EtherCAT_Diag.library

4.3 DFB_EtherCATLink_Diag

DFB_EtherCATLink_Diag 用于显示所有 EtherCAT 从站的联机状态。

FB/FC	指令	指令图
FB	DFB_EtherCATLink_Diag	
ST 语法		
<pre> DFB_EtherCATLink_Diag (bEnable :=, bValid =>, bError =>, ErrorID =>, StationNode =>); </pre>		

4

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bValid	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR*1	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
StationNode	从站地址和状态结构数组	StationStatus [1..128]*2*3	StationStatus

*注：

1. DFB_ECANT_Diag_ERROR：枚举 (Enum)
2. StationStatus 结构 (STRUCT)

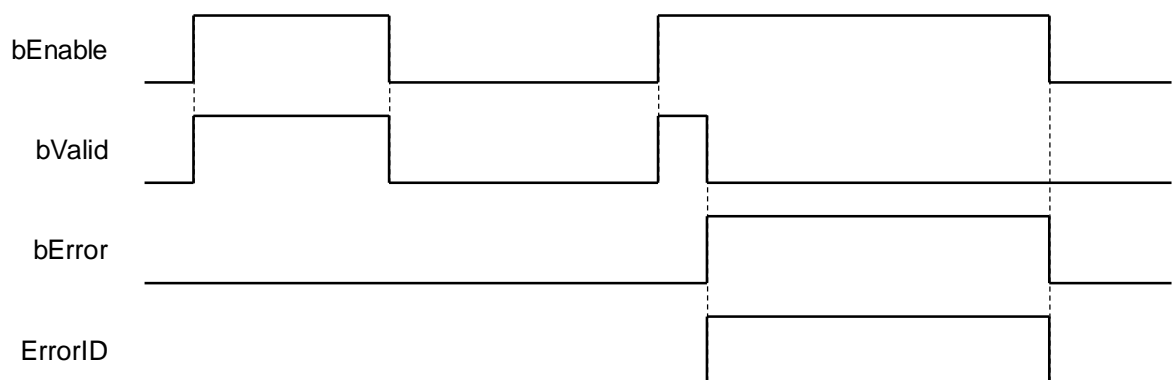
名称	功能	数据类型	设定范围 (默认值)
StationAddress	从站地址	UINT	(0)
Node	从站联机状态	BOOL	True：联机状态正常 False：联机状态异常 (False)

3. 数组中包含，第一从站开始的所有从站地址和联机状态，最大支持 128 站，bEnable 上升沿触发后，数组结构中 StationAddress 值为 0，表示该从站不存在。

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bValid	<ul style="list-style-type: none"> • 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> • 当 bEnable 下降沿时 • 当 bError 上升沿时
bError	<ul style="list-style-type: none"> • 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> • 当 bEnable 下降沿时(清除 ErrorID 记录之错误码)
ErrorID		
StationNode	<ul style="list-style-type: none"> • 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> • 当 bEnable 下降沿时

● 参数时序图



● 功能说明

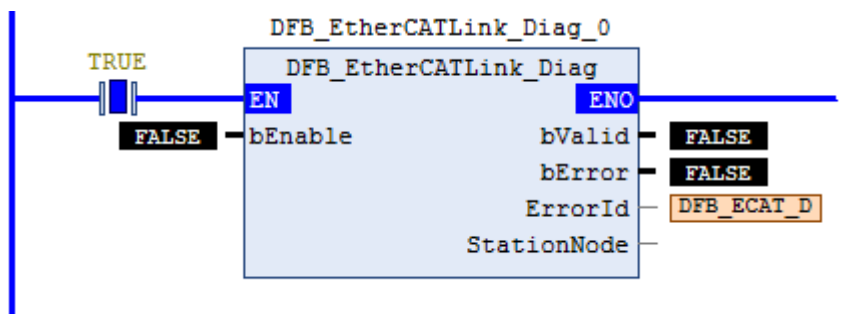
当 bEnable 上升沿时，此功能块输出参数 StationAddress 与 Node 为数组类型，输出所有从站地址与从站状态，功能块最大支持 128 站从站；数组结构中 StationAddress 值为 0，表示该从站不存在。当 bEnable 上升沿时，若搜索不到 EtherCAT master，功能块将会报错。

● 故障排除

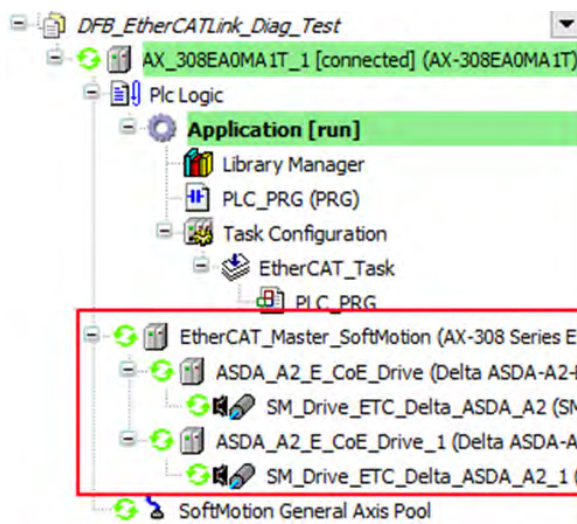
若指令执行中发生错误，此时 bError 将转为 True，可参考 ErrorID (错误码) 之内容，确认当前错误状态。

● 范例程序

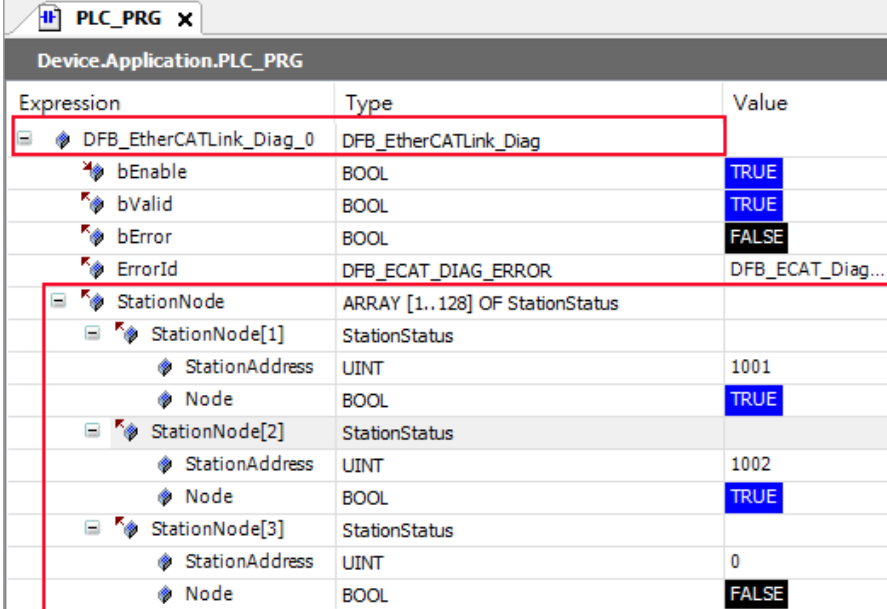
此范例说明 DFB_EtherCATLink_Diag 的功能块行为。



1. Device tree 总共有两站 EtherCAT 从站，联机状态 PASS。

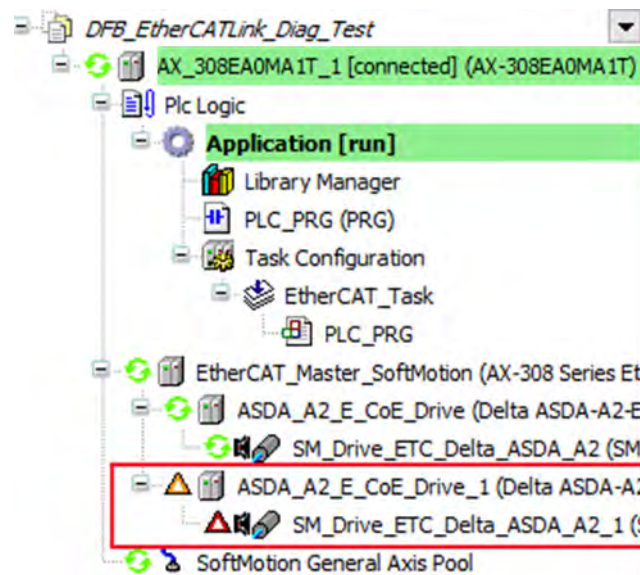


2. DFB_EtherCATLink_Diag 的 bEnable 上升沿之后，StationNode index 1 到 2 的数组显示两站从站地址和联机状态，StationNode 从 index 3 开始 StationAddress 值显示为 0。



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. 断开从站一与从站二的实体网络联机，从 Device tree 观察 EtherCAT 从站二处于 Fail 状态。



4. DFB_EtherCATLink_Diag StationNode index 2 Node 值显示为 False。

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

4

- 支持机种
 - AX-308E
- 函式库
 - DL_EtherCAT_Diag.library

4.4 DFB_GetAllECATSlaveInfo

DFB_GetAllECATSlaveInfo 取得所有从站信息。

FB/FC	指令	指令图
FB	DFB_GetAllECATSlaveInfo	
ST 语法		
<pre>DFB_GetAllECATSlaveInfo (bExecute :=, bDone =>, bBusy =>, bError =>, ErrorID =>, slaveInfoArray =>, uSlaves =>,);</pre>		

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bExecute	当 bExecute 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	功能块执行完毕	BOOL	True/False (False)
bBusy	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR*1	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
slaveInfoArray	从站信息数组	ECATSlaveInfo [1..128]*2	ECATSlaveInfo
uSlaves	从站数目	UINT	0~128 (0)

*注：

1. DFB_ECAT_Diag_ERROR：枚举（Enum）
2. slaveInfoArray 结构（STRUCT）。

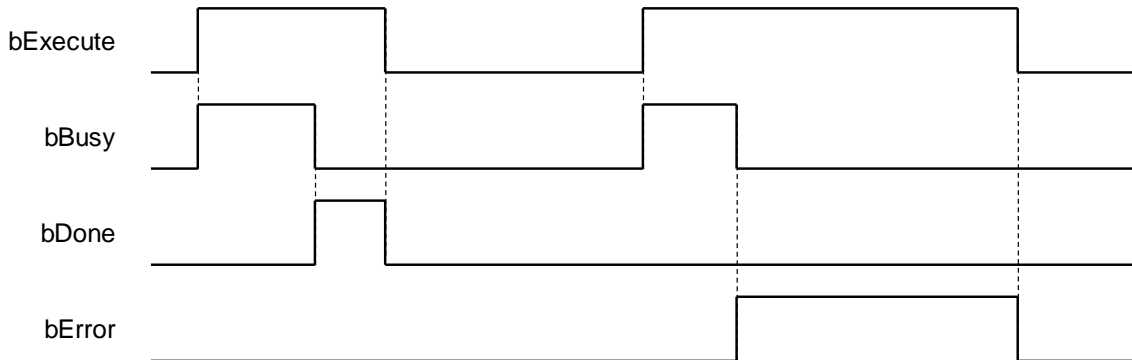
名称	功能	数据类型	设定范围 (默认值)
vendorId	从站 vendor id	UDINT	(0)
productCode	从站 product code	UDINT	(0)
revisionNo	从站 revision number	UDINT	(0)
serialNo	从站 serial number	UDINT	(0)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	<ul style="list-style-type: none"> • 当功能块执行完毕时 	<ul style="list-style-type: none"> • 当 bExecute 下降沿时 • 若 bExecute 为 False 而 bDone 转为 True，此时 bDone 维持一个扫描周期的 True 状态后，立即转成 False。
bBusy	<ul style="list-style-type: none"> • 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> • 当 bDone 上升沿时 • 当 bError 上升沿时
bError	<ul style="list-style-type: none"> • 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> • 当 bExecute 下降沿时（清除 ErrorID 记录之错误码）
ErrorID		
slaveInfoArray	<ul style="list-style-type: none"> • 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> • 当 bExecute 下降沿时
uSlaves	<ul style="list-style-type: none"> • 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> • 当 bExecute 下降沿时

4

- 参数时序图



- 功能说明

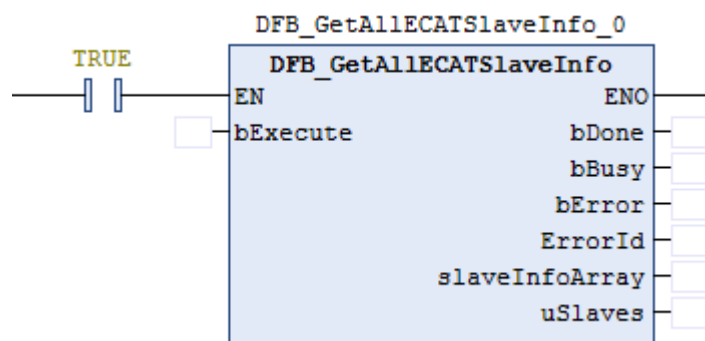
当 bExecute 上升沿时，slaveInfoArray 参数输出装置树中所有 EtherCAT 从站信息，信息包含从站 vendor id、product code、revision number 和 serial number。最大支持 128 站从站。uSlaves 参数输出 slaveInfoArray 从站数据笔数，最大 128 笔。

- 故障排除

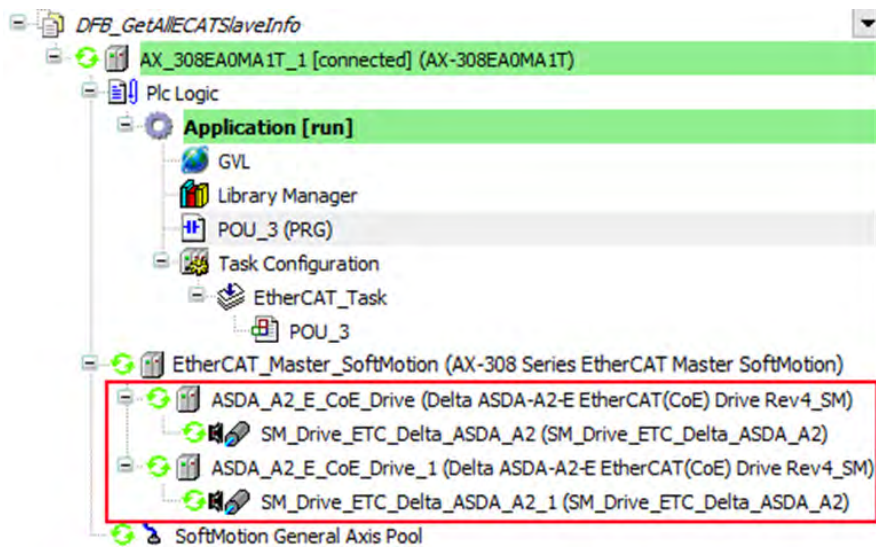
若指令执行中发生错误，此时 bError 将转为 True，可参考 ErrorID（错误码）之内容，确认当前错误状态。

- 范例程序

此范例说明 DFB_GetAllECATSlaveInfo 的功能块行为。



1. Device tree 总共有两站 EtherCAT 从站，皆为 ASDA_A2。



2. 单击装置树 ASDA_A2 观察从站信息如下。

ASDA_A2_E_CoE_Drive x

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			
DeviceIdentificationADO	UINT			0			
DeviceIdentificationMode	USINT			0			

3. DFB_GetAllECATSlaveInfo bExecute 参数上升沿，slaveInfoArray 参数输出内容如下。uSlaves 参数输出值为 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

The diagram shows a function block call for DFB_GetAllECATSlaveInfo_0. The EN input is connected to a TRUE pulse. The bExecute input is also connected to a TRUE pulse. The ENO output is TRUE, bDone is TRUE, bBusy is FALSE, bError is FALSE, ErrorId is DFB_ECAT_D, and uSlaves is 2.

- 支持机种
 - AX-308E
- 函式库
 - DL_EtherCAT_Diag.library

4.5 DFB_GetECATMasterError

DFB_GetECATMasterError 取得 EtherCAT 网络联机错误的错误码。

FB/FC	指令	指令图
FB	DFB_GetECATMasterError	
ST 语法		
<pre>DFB_GetECATMasterError (bEnable :=, bBusy =>, bError =>, ErrorID =>, EtherCATErrorld =>);</pre>		

4

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR* ¹	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
EtherCATErrorld	EtherCAT 状态错误码	ETC_LASTERROR* ²	ETC_LASTERROR (NO_ERROR)

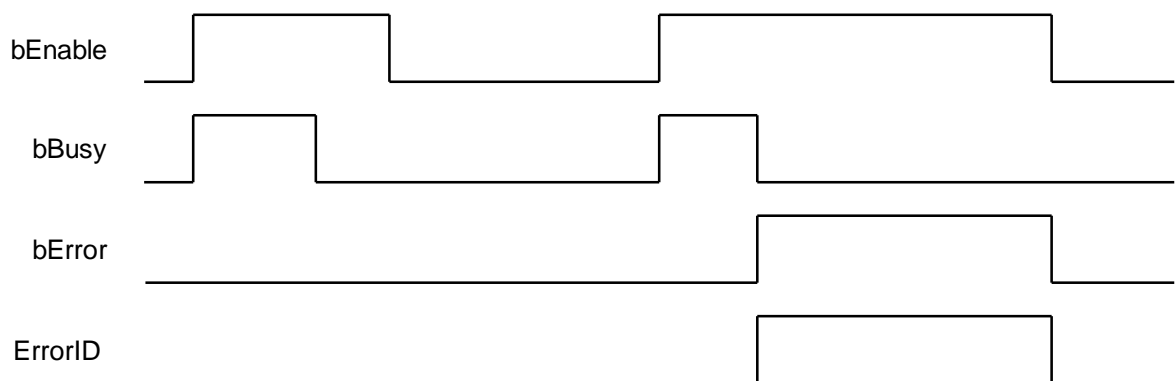
*注：

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

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bBusy	<ul style="list-style-type: none"> 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时(清除 ErrorID 记录之错误码)
ErrorID		
EtherCATErrorId	<ul style="list-style-type: none"> 当 EtherCAT 发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时

● 参数时序图



● 功能说明

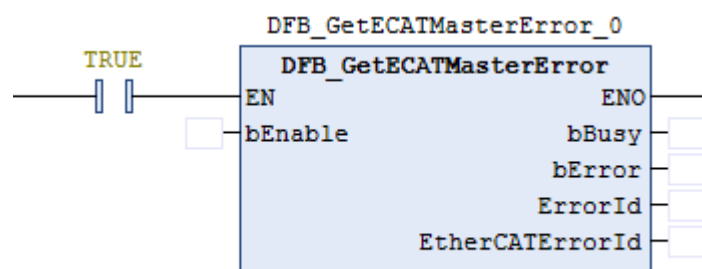
当 bEnable 上升沿时，每个周期 EtherCATErrorId 参数输出 EtherCAT 网络联机错误的错误码，如没错误则输出 NO_ERROR。EtherCAT 网络联机的错误码，可参考 Library 中 ETC_LASTERROR_STATE 之内容。

● 故障排除

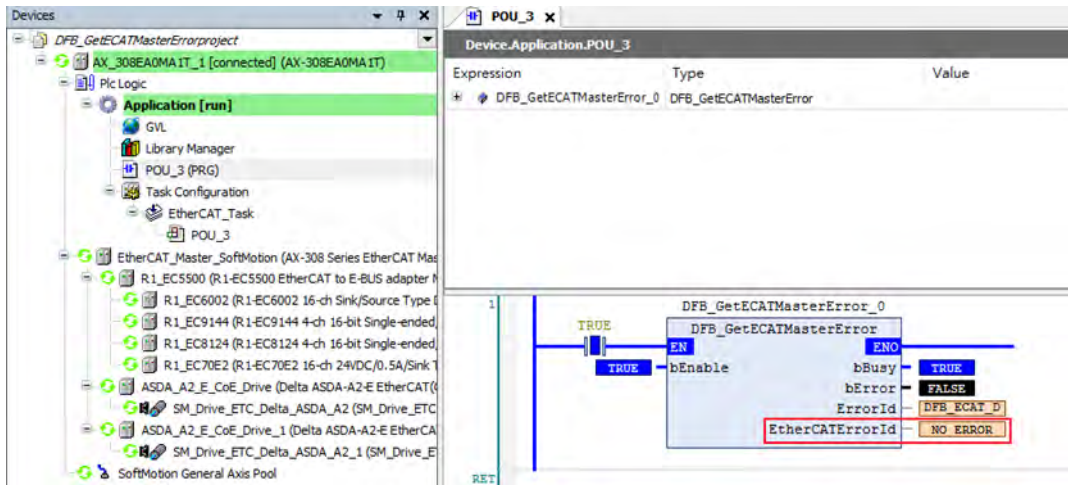
若指令执行中发生错误，此时 bError 将转为 True，可参考 ErrorID (错误码) 之内容，确认当前错误状态。

● 范例程序

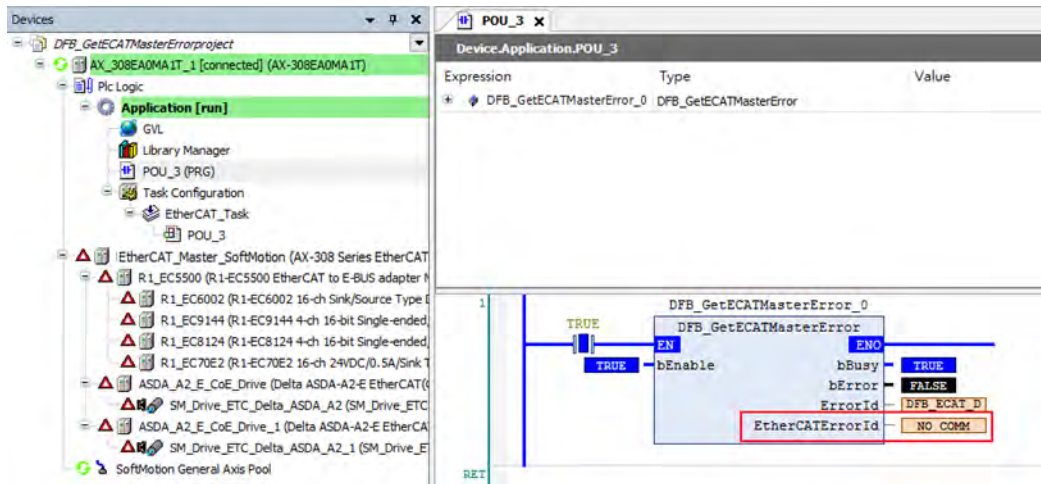
此范例说明 DFB_GetECATMasterError 的功能块行为。



1. EtherCAT 网络联机正常无错误发生。DFB_GetECATMasterError bEnable 参数上升沿，EtherCATErrorId 参数输出内容为 NO_ERROR。



2. 移除 EtherCAT 主站和从站网络联机。功能块 EtherCATErrorId 参数输出内容为 NO_COMM。



- 支持機種
 - AX-308E
- 函式庫
 - DL_EtherCAT_Diag.library

4.6 DFB_GetECATMasterState

DFB_GetECATMasterState 取得 EtherCAT Master 联机状态。

FB/FC	指令	指令图
FB	DFB_GetECATMasterState	
ST 语法		
<pre>DFB_GetECATMasterState (bEnable :=, bBusy =>, bError =>, ErrorID =>, bStatus =>,);</pre>		

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bEnable	当 bEnable 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

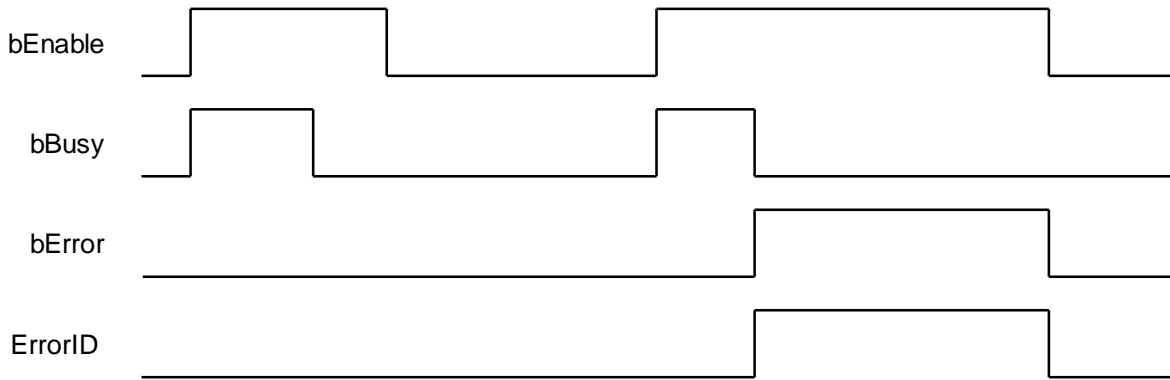
名称	功能	数据类型	输出值范围 (默认值)
bBusy	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)
bStatus	EtherCAT master 通讯状态	BOOL	True/False (False)

*注：DFB_ECAT_Diag_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bBusy	<ul style="list-style-type: none"> 当 bEnable 上升沿触发时 	<ul style="list-style-type: none"> 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> 当 bEnable 下降沿时(清除 ErrorID 记录之错误码)
ErrorID		
bStatus	<ul style="list-style-type: none"> 当 EtherCAT master 联机状态正常 	<ul style="list-style-type: none"> 当 bEnable 下降沿时 当 bError 上升沿时 当 EtherCAT master 联机状态异常

● 参数时序图



● 功能说明

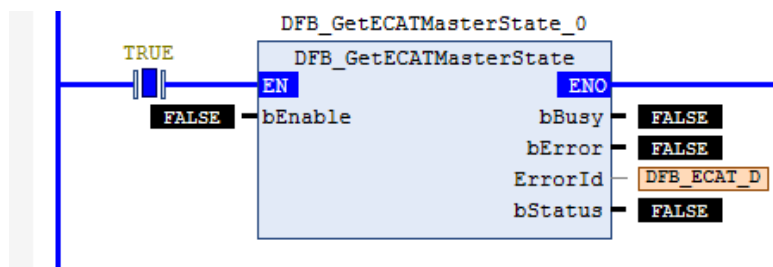
当 bEnable 上升沿时，功能块每个周期更新 EtherCAT master 通讯状态。

● 故障排除

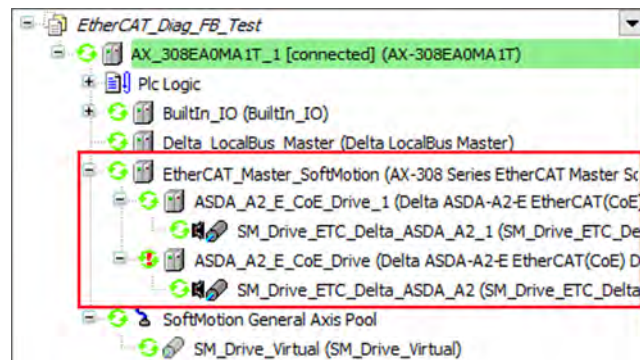
若指令执行中发生错误，此时 bError 将转为 True，可参考 ErrorID (错误码) 之内容，确认当前错误状态。

● 范例程序

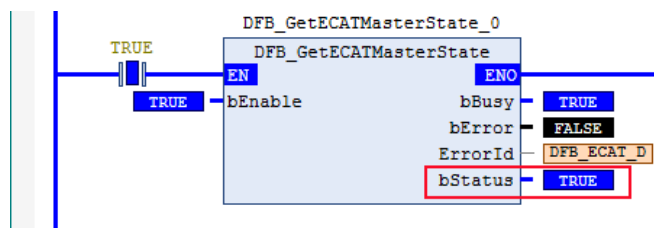
此范例说明 DFB_GetECATMasterState 的功能块行为。



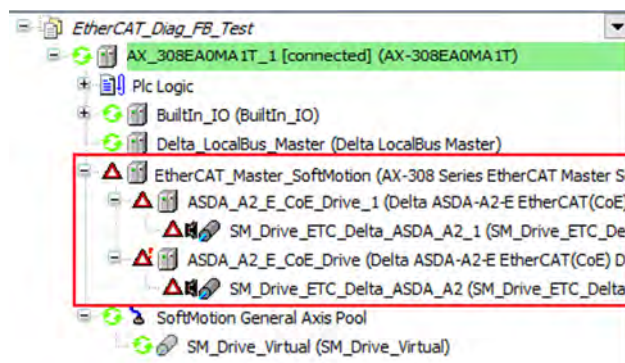
1. 从 Device tree 观察目前 EtherCAT master 联机状态 PASS



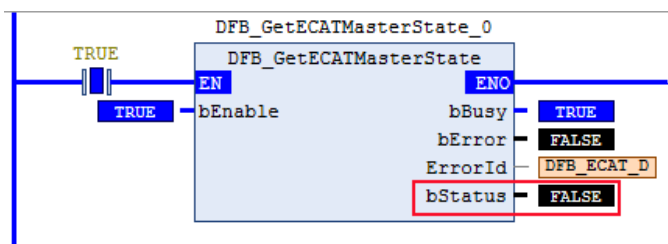
2. DFB_GetECATMasterState 功能块 bEnable 上升沿 · bStatus 显示为 True。



3. 断开 EtherCAT 主站与从站的实体网络联机，从 Device tree 观察目前 EtherCAT master 联机状态 Fail。



4. DFB_GetECATMasterState bStatus 显示为 False。



- 支持机种
 - AX-308E

- 函式库
 - DL_EtherCAT_Diag.library

4.7 DFB_ResetECATMaster

DFB_ResetECATMaster 重置联机状态异常的 EtherCAT 主站。

FB/FC	指令	指令图
FB	DFB_ResetECATMaster	
ST 语法		
<pre>DFB_ResetECATMaster (bExecute :=, bDone =>, bBusy =>, bError =>, ErrorID =>,);</pre>		

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bExecute	当 bExecute 上升沿时执行指令	BOOL	True/False (False)	-

● 输出参数

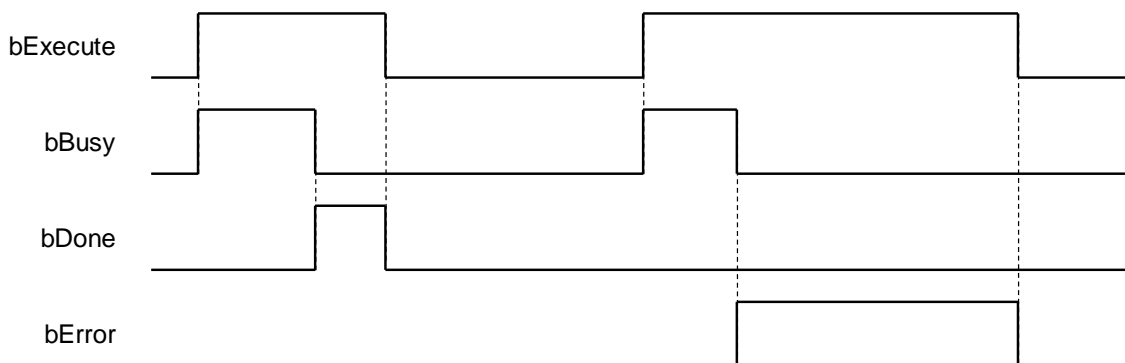
名称	功能	数据类型	输出值范围 (默认值)
bDone	功能块执行完毕	BOOL	True/False (False)
bBusy	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)

*注：DFB_ECAT_Diag_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	<ul style="list-style-type: none"> 当功能块执行完毕时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 若 bExecute 为 False 而 bDone 转为 True，此时 bDone 维持一个扫描周期的 True 状态后，立即转成 False。
bBusy	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bDone 上升沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 (清除 ErrorID 记录之错误码)
ErrorID		

● 参数时序图



● 功能说明

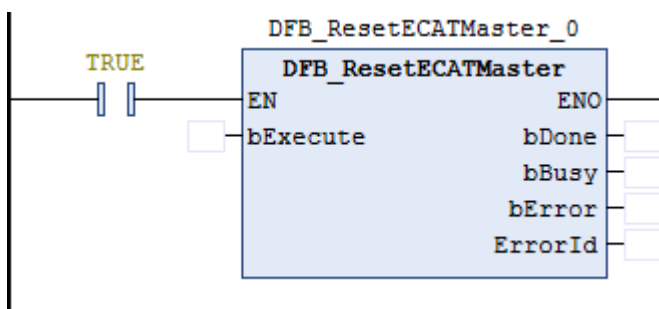
当 bExecute 上升沿时，若 EtherCAT 主站联机状态 Fail，则进行重置。

● 故障排除

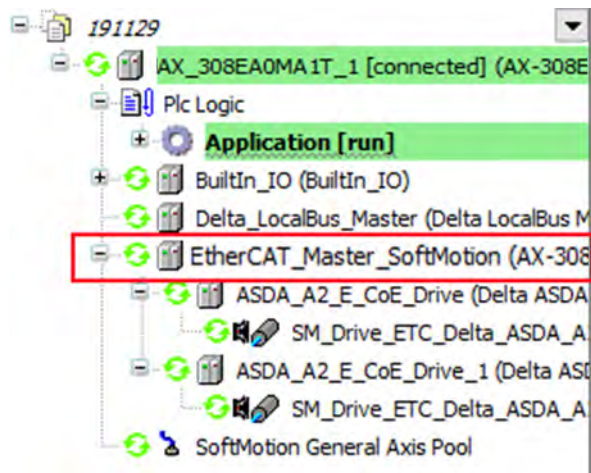
若指令执行中发生错误，此时 bError 将转为 True，可参考 ErrorID (错误码) 之内容，确认当前错误状态。

● 范例程序

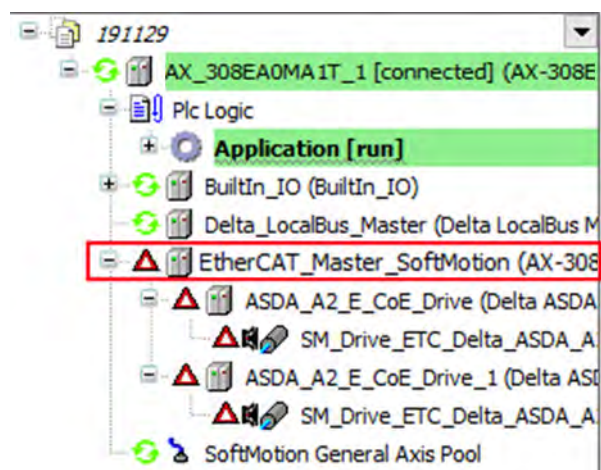
此范例说明 DFB_ResetECATMaster 的功能块行为。



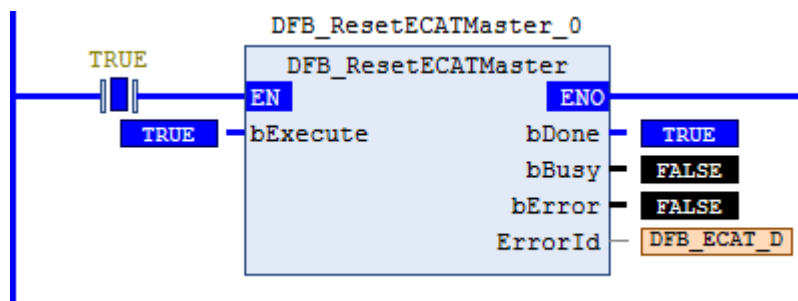
1. Device tree 显示 EtherCAT 主站联机状态 PASS



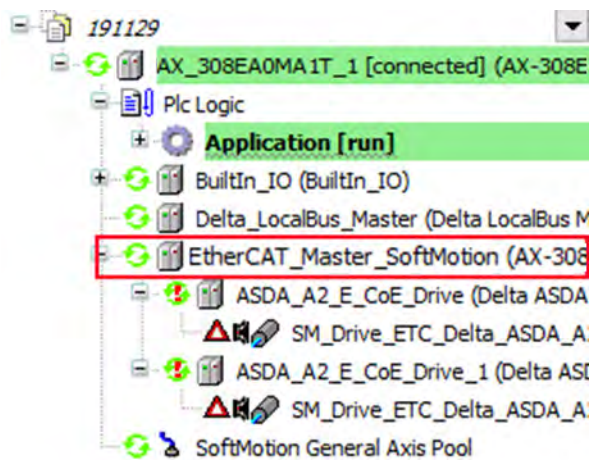
2. 断开主站与从站的实体网络联机，从 Device tree 观察 EtherCAT 主站处于 Fail 状态。



3. 恢复主站与从站的实体网络联机，功能块 bExecute 上升沿。



4. 功能块 bDone 为 True 之后，主站已恢复联机状态。



4

- 支持机种
 - AX-308E

- 函式库
 - DL_EtherCAT_Diag.library

4.8 DFB_ResetECATSlave

DFB_ResetECATSlave 重置联机状态异常的从站。

FB/FC	指令	指令图
FB	DFB_ResetECATSlave	
ST 语法		
<pre>DFB_ResetECATSlave(bExecute :=, uiSlaveAddr :=, tTimeout :=, bDone =>, bBusy =>, bError =>, ErrorID =>,);</pre>		

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)	生效时机
bExecute	当 bExecute 上升沿时执行指令	BOOL	True/False (False)	-
uiSlaveAddr	欲重置从站地址	UINT	正数 (0)	当 bExecute 为上升沿触发且 Busy 为 False
tTimeout	从站重置超时时间	TIME	正数 (0)	当 bExecute 为上升沿触发且 Busy 为 False

● 输出参数

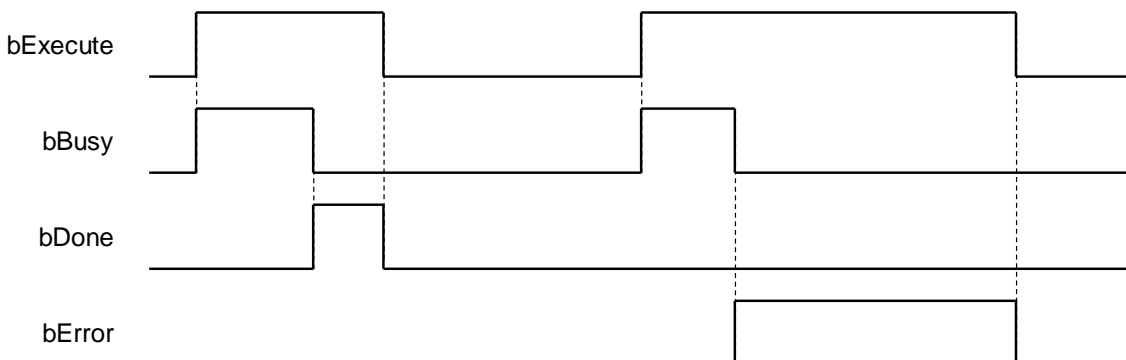
名称	功能	数据类型	输出值范围 (默认值)
bDone	功能块执行完毕	BOOL	True/False (False)
bBusy	指令正在执行时为 True	BOOL	True/False (False)
bError	错误发生时为 True	BOOL	True/False (False)
ErrorID	错误码	DFB_ECAT_Diag_ERROR*	DFB_ECAT_Diag_ERROR (DFB_ECAT_Diag_NO_ERROR)

*注：DFB_ECAT_Diag_ERROR：枚举 (Enum)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	<ul style="list-style-type: none"> 当功能块执行完毕时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 若 bExecute 为 False 而 bDone 转为 True，此时 bDone 维持一个扫描周期的 True 状态后，立即转成 False。
bBusy	<ul style="list-style-type: none"> 当 bExecute 上升沿触发时 	<ul style="list-style-type: none"> 当 bDone 上升沿时 当 bError 上升沿时
bError	<ul style="list-style-type: none"> 当指令的执行条件发生错误时 	<ul style="list-style-type: none"> 当 bExecute 下降沿时 (清除 ErrorID 记录之错误码)。
ErrorID		

● 参数时序图



● 功能说明

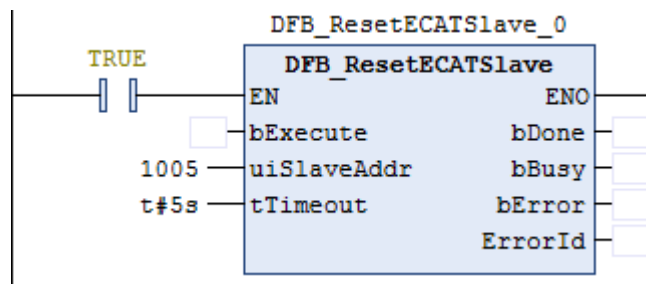
当 bExecute 上升沿时，功能块依据 uiSlaveAddr 值寻找目标从站，若目标从站联机状态 Fail，则进行重置。若 uiSlaveAddr 为 0，则对所有联机状态错误从站进行重置。

- 故障排除

若指令执行中发生错误，此时 `bError` 将转为 `True`，可参考 `ErrorID`（错误码）之内容，确认当前错误状态。

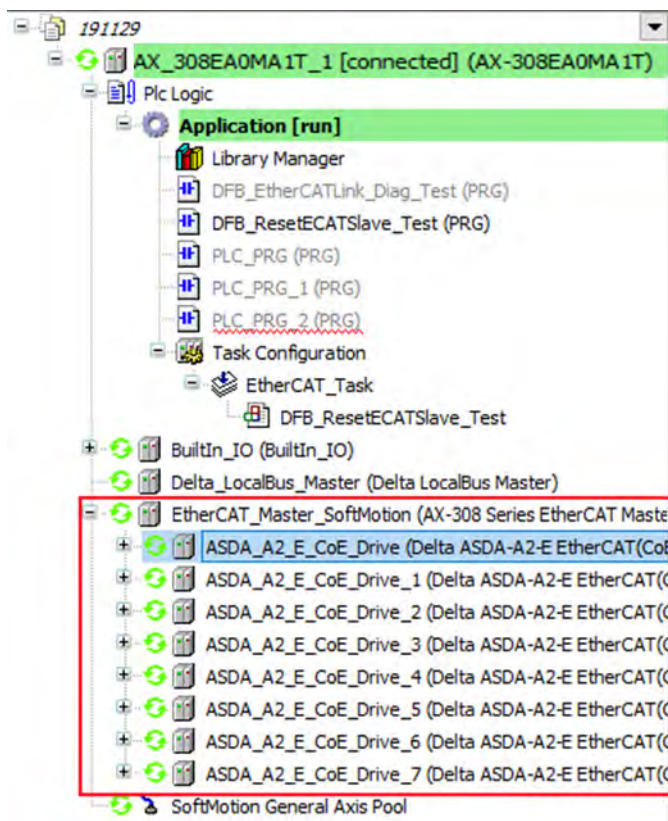
- 范例程序

此范例说明 `DFB_ResetECATSlave` 的功能块行为。

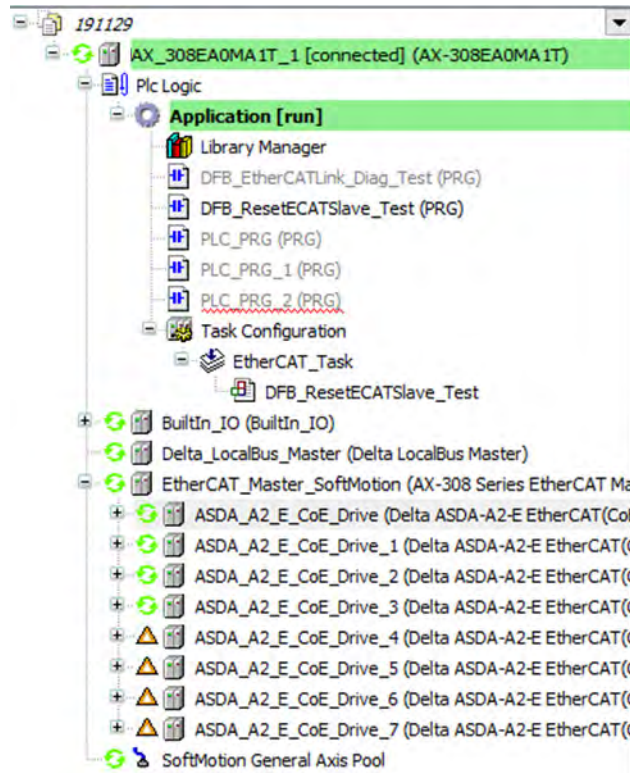


1. EtherCAT_Master_SoftMotion 总共有八站 EtherCAT 从站，联机状态 PASS。

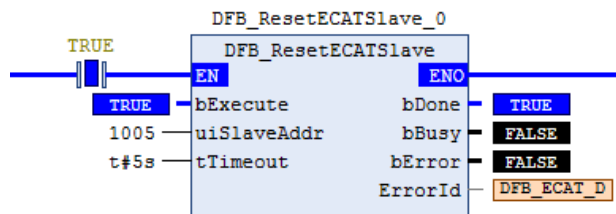
4



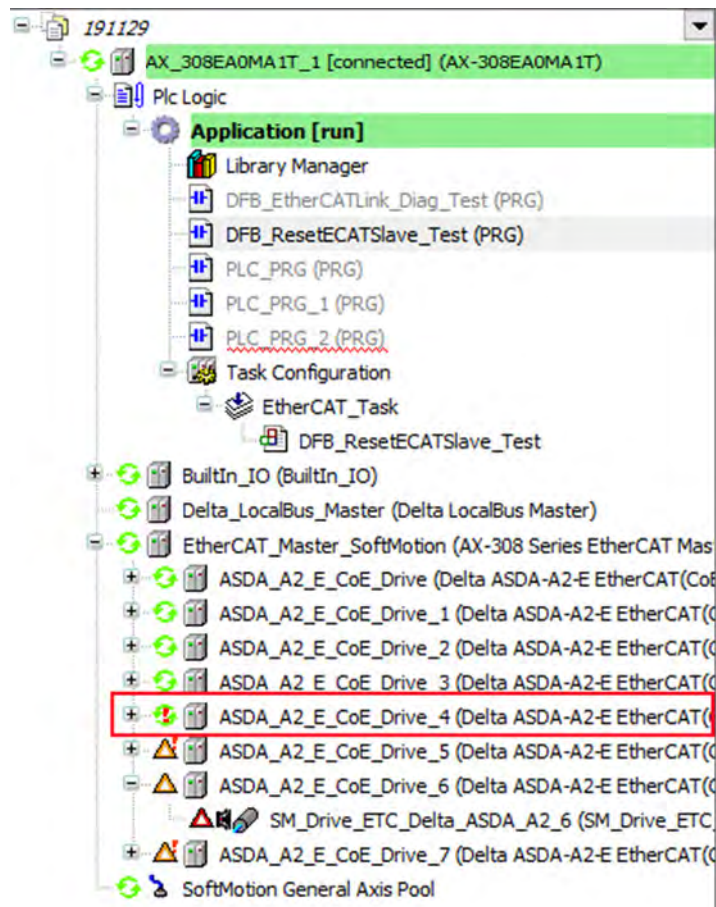
2. 断开从站四与从站五的实体网络联机，从 Device tree 观察 EtherCAT 从站第五站开始处于 Fail 状态。



3. 恢复从站四与从站五的实体网络联机，uiSlaveAddr 参数值输入 1005，bExecute 上升沿。



4. 功能块 bDone 为 True 之后，从站五已恢复联机状态。



5. 若 uiSlaveAddr 参数值输入 0 则会重置所有从站。

- 支持机种
 - AX-308E
- 函式库
 - DL_EtherCAT_Diag.library

4.9 错误码与故障排除

下表列出功能块相对应错误码与处置方式：

说明	错误原因	处置方式
DFB_ECANT_Diag_MASTER_CANT_BE_FOUND	搜索不到 EtherCAT 主站	此 EtherCAT 主站不在网络拓扑图中，请确认主站是否配置正常。
DFB_ECANT_Diag_MASTER_ERROR	EtherCAT 主站状态错误	请先排除 EtherCAT 主站错误，再执行此功能块。
DFB_ECANT_Diag_SLAVE_CANT_BE_FOUND	搜索不到 EtherCAT 从站	此 EtherCAT 从站不在网络拓扑图中，请输入正确的从站地址。
DFB_ECANT_Diag_MASTER_RESTART_TIMEOUT	EtherCAT 主站复位超时	请确认超时时间是否太短，或者主站网络已遗失。
DFB_ECANT_Diag_SLAVE_RESTART_TIMEOUT	EtherCAT 从站复位超时	请确认超时时间是否太短，或者从站网络已遗失。
DFB_ECANT_Diag_MASTER_DISABLE	EtherCAT 主站失能	请确认是否始能 EtherCAT 主站

第5章 检查码计算指令

目录

5.1	DFC_LRC8.....	5-2
5.2	DFC_LRC16.....	5-4
5.3	DFC_LRC32.....	5-6
5.4	错误码与故障排除.....	5-8

5.1 DFC_LRC8

DFC_LRC8 : LRC (8 位) 检查码计算。

FB/FC	指令	指令图	ST 语法
FC	DFC_LRC8		<pre>DFC_LRC8(pSrc:= , dwLen:= , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
pSrc	LRC 检查码计算起始地址	POINTER TO BYTE	内存地址 (0)
dwLen	LRC 检查码计算的数据长度	DWORD*	(0)

*注：dwLen 输入参数可使用 BYTE、WORD、DWORD 变量类型。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
DFC_LRC8	LRC检查码 (此参数为Return形式)	BYTE	(0)
ErrorID	错误码	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

● 功能说明

当执行此 FC 指令后，即从输入参数 (pSrc) 起始的内存地址，开始计算 LRC (8 位) 检查码，其检查码计算的范围由输入参数 dwLen 决定。

- 范例程序

此范例利用 FC 指令 (DFC_LRC8) 进行 LRC (8 位) 检查码的计算。

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

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

检查码计算的范围为 6 (dwLen = 6)，因此 FC 指令 (DFC_LRC8) 会从输入参数 pSrc (ar_byVar0[0]) 内存的地址连续六笔 BYTE 数据进行检查码的计算，最后得到的检查码为 16#CF。

- 支持机种

- AX 系列

- 函式库

- DL_LRC.library

5.2 DFC_LRC16

DFC_LRC16 : LRC (16 位) 检查码计算。

FB/FC	指令	指令图	ST 语法
FC	DFC_LRC16		<pre>DFC_LRC16(pSrc:= , dwLen:= , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
pSrc	LRC 检查码计算起始地址	POINTER TO BYTE	内存地址 (0)
dwLen	LRC 检查码计算的数据长度	DWORD*	(0)

*注：dwLen 输入参数可使用 BYTE、WORD、DWORD 变量类型。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
DFC_LRC16	LRC检查码 (此参数为Return形式)	WORD	(0)
ErrorID	错误码	DL_MOV_ERROR	DL_MOV_ERROR(DFC_NO_ERROR)

● 功能说明

当执行此 FC 指令后，即从输入参数 (pSrc) 起始的内存地址，开始计算 LRC (16 位) 检查码，其检查码计算的范围由输入参数 dwLen 决定。

- 范例程序

此范例利用 FC 指令 (DFC_LRC16) 进行 LRC (16 位) 检查码的计算。

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   bVar0: BOOL;
4   wVar0: WORD;
5   ar_wVar0: ARRAY [0..5] OF WORD := [16#3031,16#3132,16#3233,16#3334,16#3435,16#3536];
6 END_VAR
7
8 IF bVar0 THEN
9   wVar0:=DFC_LRC16(pSrc:=ADR(ar_wVar0[0]) , dwLen:=6 , ErrorID=> );
10  bVar0:=FALSE;
11 END_IF;
```

检查码计算的范围为 6 (dwLen = 6) · 因此 FC 指令 (DFC_LRC16) 会从输入参数 pSrc (ar_wVar0[0]) 内存的地址连续六笔 WORD 数据进行检查码的计算 · 最后得到的检查码为 16#CFCB。

- 支持机种

- AX 系列

- 函式库

- DL_LRC.library

5.3 DFC_LRC32

DFC_LRC32 : LRC (32 位) 检查码计算。

FB/FC	指令	指令图	ST 语法
FC	DFC_LRC32		<pre>DFC_LRC32(pSrc:= , dwLen:= , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
pSrc	LRC 检查码计算起始地址	POINTER TO BYTE	内存地址 (0)
dwLen	LRC 检查码计算的数据长度	DWORD*	(0)

*注 : dwLen 输入参数可使用 BYTE 、 WORD 、 DWORD 变量类型。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
DFC_LRC32	LRC检查码 (此参数为Return形式)	DWORD	(0)
ErrorID	错误码	DL_MOV_ERROR	DL_MOV_ERROR (DFC_NO_ERROR)

● 功能说明

当执行此 FC 指令后，即从输入参数 (pSrc) 起始的内存地址，开始计算 LRC (32 位) 检查码，其检查码计算的范围由输入参数 dwLen 决定。

- 范例程序

此范例利用 FC 指令 (DFC_LRC32) 进行 LRC (32 位) 检查码的计算。

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   bVar0: BOOL;
4   dwVar0: DWORD;
5   ar_dwVar0: ARRAY [0..3] OF DWORD := [16#30313233,16#31323334,16#32333435,16#33343536];
6 END_VAR
1 IF bVar0 THEN
2   dwVar0:=DFC_LRC32(pSrc:=ADR(ar_dwVar0[0]) , dwLen:=4 , ErrorID=> );
3   bVar0:=FALSE;
4 END_IF;
```

检查码计算的范围为 4 (dwLen = 4) · 因此 FC 指令 (DFC_LRC32) 会从输入参数 pSrc (ar_dwVar0[0]) 内存的地址连续四笔 DWORD 数据进行检查码的计算 · 最后得到的检查码为 16#3935312E。

- 支持机种

- AX 系列

- 函式库

- DL_LRC.library

5.4 错误码与故障排除

说明	错误原因	处置方式
DFC_LRC_ERR_PARAMETER	dwLen 数值不正确	确认 dwLen 值是否大于 0

第6章 程序编辑

目录

6.1	DFB_From.....	6-2
6.2	DFB_To.....	6-5
6.3	错误码与故障排除	6-8

6.1 DFB_From

DFB_From：模块 CR 数据读取。

FB/FC	指令	指令图	ST 语法
FB	DFB_From		<pre>DFB_From(bExecute:= , byRemoteID:= , byLocalID:= , wCRAAddr:= , iLength:= , pVal:= , bDone=> , bBusy=> , bError=> , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
byRemoteID*	主机或远程模块编号	BYTE	0 : 主机 1~16 : 远程模块 (0)
byLocalID	扩展模块编号	BYTE	0 ~ 31
wCRAAddr	模块 CR 数据位置	WORD	(0)
iLength	模块 CR 数据长度	INT	1~125 (0)
pVal	所读取的 CR 数据	POINTER TO WORD	

*注：目前仅支持模式 0。

● 输出参数

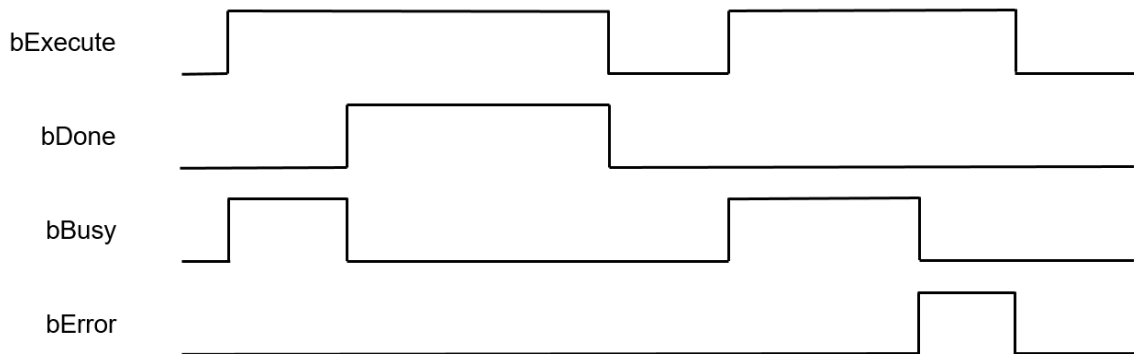
名称	功能	数据类型	输出值范围 (默认值)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)

名称	功能	数据类型	输出值范围 (默认值)
ErrorID	错误码	DFB_AS_MODULE_API_ERROR	DFB_AS_MODULE_API_ERROR (DFB_NO_ERROR)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ErrorID		

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_From) 读取模块的 CR 数据。

● 范例程序

此范例利用 FB 指令 (DFB_From) 读取主机右侧第二个模块内的 CR 编号 1 之值，并将其值储存至控制器的变量 (wVar)。

```

1  PROGRAM PLC_PRG
2  VAR
3      bVar0: BOOL :=TRUE;
4      bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
5      byRemoteID_Var,byLocalID_Var: BYTE;
6      wCRAddr_Var: WORD;
7      iLength_Var: INT;
8      ErrorID_Var: DFB_AS_MODULE_API_ERROR
9      wVar0: WORD;
10     FB0: DFB_From;
11 END_VAR
12 IF bVar0 THEN
13     bExecute_Var:=TRUE;
14     byRemoteID_Var:=0;
15     byLocalID_Var:=2;
16     wCRAddr_Var:=1;
17     iLength_Var:=1;
18     bVar0:=FALSE;
19 END_IF
20 IF bDone_Var THEN
21     bExecute_Var:=FALSE;
22 END_IF
23 FB0(
24     bExecute:=bExecute_Var ,
25     byRemoteID:=byRemoteID_Var ,
26     byLocalID:=byLocalID_Var ,
27     wCRAddr:=wCRAddr_Var ,
28     iLength:=iLength_Var ,
29     pVal:=ADR(wVar0) ,
30     bDone=>bDone_Var ,
31     bBusy=>bBusy_Var ,
32     bError=>bError_Var ,
33     ErrorID=>ErrorID_Var );

```

6

● 支持机种

- AX-308E

● 函式库

- DL_ASModuleAPI_AX3

6.2 DFB_To

DFB_To：模块 CR 数据写入。

FB/FC	指令	指令图	ST 语法
FB	DFB_To		<pre>DFB_To(bExecute:= , byRemoteID:= , byLocalID:= , wCRAddr:= , iLength:= , pVal:= , bDone=> , bBusy=> , bError=> , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
byRemoteID*	主机或远程模块编号	BYTE	0 : 主机 1~16 : 远程模块 (0)
byLocalID	扩展模块编号	BYTE	0 ~ 31
wCRAddr	模块 CR 地址	WORD	(0)
iLength	模块 CR 数据长度	INT	1~125 (0)
pVal	所读取的 CR 数据	POINTER TO WORD	

*注：目前仅支持模式 0。

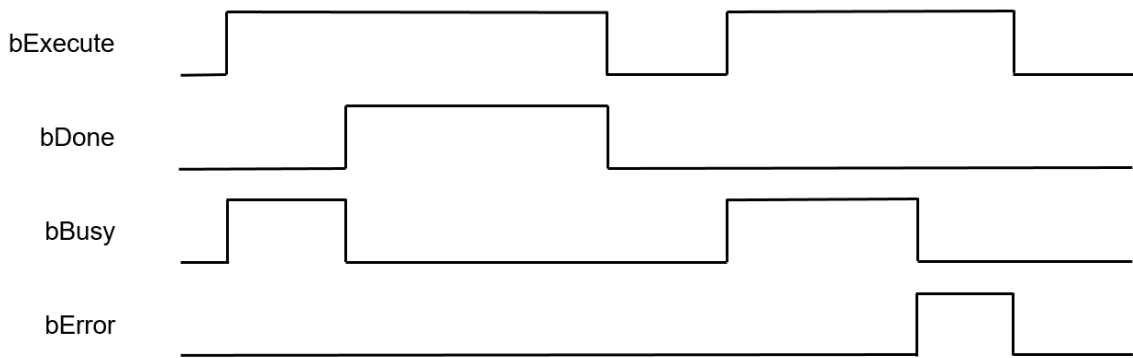
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DFB_AS_MODULE_API_ERROR	DFB_AS_MODULE_API_ERROR (DFB_NO_ERROR)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ErrorID		

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_To) 写入模块的 CR 数据。

- 范例程序

此范例利用 FB 指令 (DFB_To) 将控制器的变量 (wVar) 内容值，写入主机右侧第二个模块内的 CR 编号 1。

```

PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3      bVar0: BOOL :=TRUE;
4      bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
5      byRemoteID_Var,byLocalID_Var: BYTE;
6      wCRAddr_Var: WORD;
7      iLength_Var: INT;
8      ErrorID_Var: DFB_AS_MODULE_API_ERROR
9      wVar0: WORD :=2;
10     FB0: DFB_To;
11 END_VAR

1  IF bVar0 THEN
2      bExecute_Var:=TRUE;
3      byRemoteID_Var:=0;
4      byLocalID_Var:=2;
5      wCRAddr_Var:=1;
6      iLength_Var:=1;
7      bVar0:=FALSE;
8  END_IF
9  IF bDone_Var THEN
10     bExecute_Var:=FALSE;
11 END_IF
12 FB0(
13     bExecute:=bExecute_Var ,
14     byRemoteID:=byRemoteID_Var ,
15     byLocalID:=byLocalID_Var ,
16     wCRAddr:=wCRAddr_Var ,
17     iLength:=iLength_Var ,
18     pVal:=ADR(wVar0) ,
19     bDone=>bDone_Var ,
20     bBusy=>bBusy_Var ,
21     bError=>bError_Var ,
22     ErrorID=>ErrorID_Var );
23

```

- 支持机种

- AX-308E

- 函式库

- DL_ASModuleAPI_AX3

6.3 错误码与故障排除

说明	错误原因	处置方式
DFB_FROM_ERR_PARAMETER	输入参数错误	请确认输入参数是否正确
DFB_FROM_ERR_COMMUNICATION	CAN bus 通讯错误	请确认错误纪录
DFB_FROM_ERR_CRADDR	CR 地址错误	请检查 CR 地址是否正确
DFB_TO_ERR_PARAMETER	输入参数错误	请确认输入参数是否正确
DFB_TO_ERR_COMMUNICATION	CAN bus 通讯错误	请确认错误纪录
DFB_TO_ERR_CRADDR	CR 地址错误	请检查 CR 地址是否正确

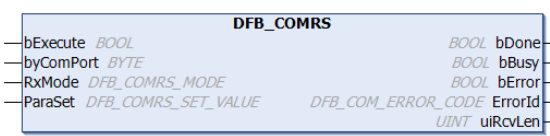
第7章 串行通讯指令

目录

7.1	DFB_COMRS.....	7-2
7.2	DFB_ModbusComChannel.....	7-7
7.3	DFB_ModbusRequest	7-10
7.4	DFB_ModbusRequest2	7-14
7.5	错误码与故障排除	7-18

7.1 DFB_COMRS

DFB_COMRS：串行通讯读写指令。

FB/FC	指令	指令图	ST 语法
FB	DFB_COMRS		<pre>DFB_COMRS (bExecute:= , byComPort:= , RxMode:= , ParaSet:= , bDone=> , bBusy=> , bError=> , ErrorId=> , uiRcvLen=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
byComPort*	串行通讯端口号	BYTE	(0xFF)
RxMode	通讯接收模式	DFB_COMRS_MODE	(NO_RECEIVING)
ParaSet	串行通讯参数	DFB_COMRS_SET_VALUE	(COMRS_SET_VALUE)

*注：各机种所定义的串行通讯端口不尽相同，需依照其定义的方式来设定。

■ DFB_COMRS_MODE

名称	说明
NO_RECEIVING	不接收数据模式： 当报文发送完毕后，即完成本次接收且完成标志即为 True。
DISCONTINUOUS_TIME	报文间隔时间模式： 当接收报文的时间间隔大于所设定的时间，即完成本次接收，且完成标志为 True。 报文间隔时间可通过 ParaSet.uiDiscontinuousTime 进行设定。 (*1)

名称	说明
SPECIFIC_END_CHAR	终止符模式： 当接收到特定长度字符时，即完成本次接收，且完成标志为 True。 终止符与其长度可由 ParaSet.pSpecificEndChar 与 ParaSet.byEndCharAmt 进行设定。 (*1, *2)
SPECIFIC_START_CHAR_AND_DISCONTINUOUS_TIME	起始字符与间隔时间模式： 当接受到特定长度字符开始接收报文，并判断当报文间隔时间大于所设定的时间，即完成本次接收，且完成标志为 True。 起始字符与其长度可由 ParaSet.pSpecificStartChar 与 ParaSet.byStartCharAmt 进行设定，而报文间隔时间可通过 ParaSet.uiDiscontinuousTime 进行设定。 (*1, *2)
SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR	起始字符与终止符模式： 当接受到特定长度字符时，即开始接收报文，直到接受到终止符，即完成本次接收。 起始字符与其长度可由 ParaSet.pSpecificStartChar 与 ParaSet.byStartCharAmt 进行设定而终止符与其长度可由 ParaSet.pSpecificEndChar 与 ParaSet.byEndCharAmt 进行设定。 (*1, *2)
SPECIFIC_LENGTH	报文长度模式： 当接受到特定长度报文后即完成本次接收。 报文长度可以通过 ParaSet.uiSetVarue 进行设定。

*注：

*1：当所接受到的报文长度达到 uiReadBufSize 时，即完成本次接收。

*2：报文长度计算包含了起始与终止符。

■ COMRS_SET_VALUE

名称	功能	数据类型	设定值范围 (默认值)
uiWriteLen	传送报文的长度 (单位：Byte)	UINT	0 ~ 1000 (0)
pWriteBuf	欲传送报文的内存地址	POINTER TO BYTE	--

名称	功能	数据类型	设定值范围 (默认值)
pReadBuf	接收报文所储存的内存地址	POINTER TO BYTE	--
uiReadBufSize	接收报文的储存内存大小 (单位 : Byte)	UINT	1 ~ 1,000 (100)
uiDiscontinuousTime	报文间隔时间 (单位 : ms)	UINT	2 ~ 3,000 (2)
byStartCharAmt	起始字符大小	BYTE (单位 : Byte)	1 ~ 255 (1)
pSpecificStartChar	起始字符内存地址	POINTER TO BYTE	内存地址 (0)
byEndCharAmt	终止符大小	BYTE (单位 : Byte)	1 ~ 255 (1)
pSpecificEndChar	终止符内存地址	POINTER TO BYTE	内存地址 (0)
uiSpecificRxLen	接收长度	UINT (单位 : Byte)	1 ~ 1000 (1)
tTimeout	通讯逾时时间	TIME	T#0ms ~ T#49d17h2m47s295ms (T#100ms) T#0ms : 无逾时判断

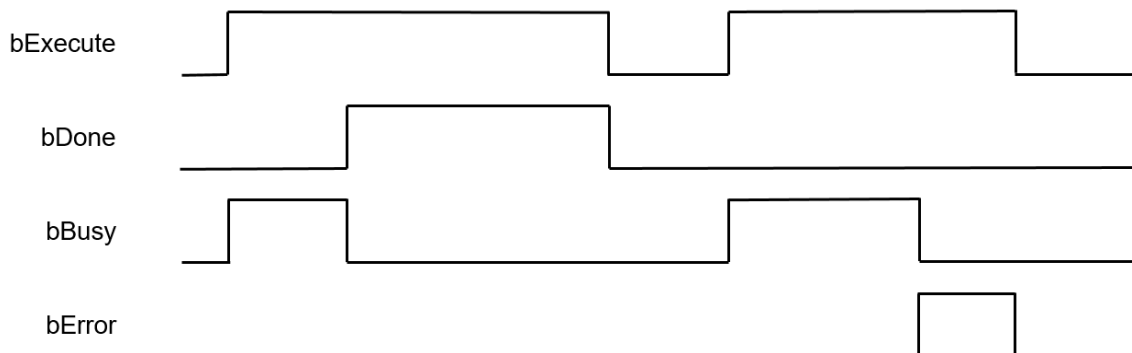
7 ● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DFB_COM_ERROR_CODE	(DFB_UNDEFINED)
uiRcvLen	接收数据长度	UINT (单位 : Byte)	(0)

■ 输出更新时间点

名称	输出参数上缘时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿且 FB 指令执行完毕时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ErrorID		

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_COMRS) 发送串行通讯报文, 在使用本指令之前, 必须要将主机 COM port 设定完毕并新增 Delta_Modbus_Master_COM_Port device (详细步骤请参考 AX-3 系列操作手册第 9.2 节「串行 (Serial Port) 通讯」)

● 范例程序

此范例利用 FB 指令 (DFB_COMRS) 发送串行通讯报文。

```

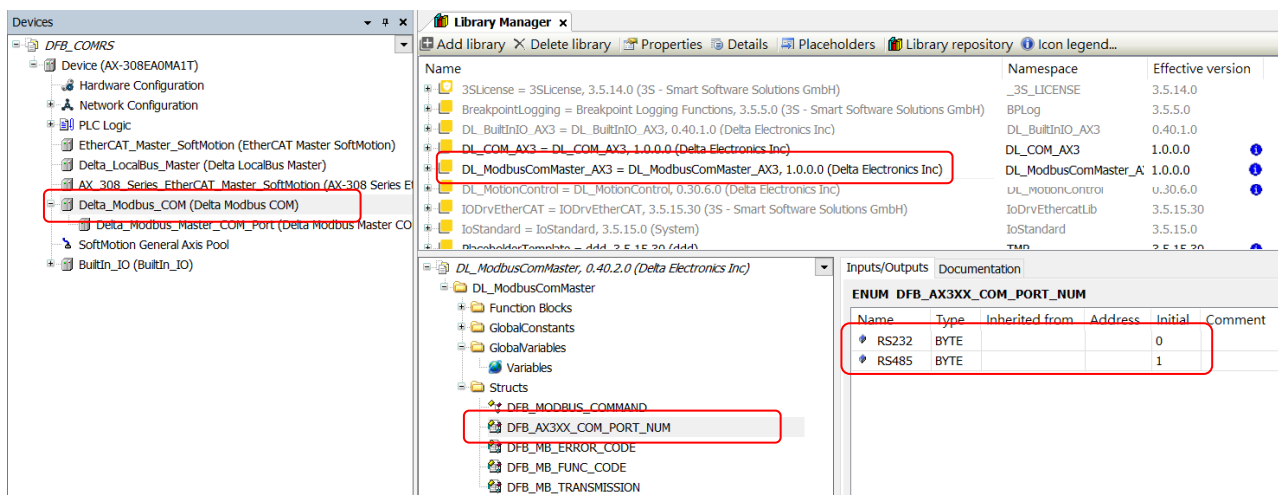
PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3      bVar0: BOOL:=TRUE;
4      bExecute_Var, bDone_Var, bBusy_Var, bError_Var: BOOL;
5      FB0: DFB_COMRS;
6      RxMode_Var: DL_COM_AX3.DFB_COMRS_MODE;
7      ParaSet_Var: DL_COM_AX3.DFB_COMRS_SET_VALUE;
8      ErrorID_Var: DL_COM_AX3.DFB_COM_ERROR_CODE;
9      uiRcvLen_Var: UINT;
10     pSpecificStartChar_Var: BYTE :=16#3A;
11     byStartCharAmt_Var: BYTE :=1;
12     pSpecificEndChar_Var: ARRAY[0..1] OF BYTE:=[16#0D,16#0A];
13     byEndCharAmt_Var: BYTE :=2;
14     ar_byVar0: ARRAY [0..13] OF BYTE;
15     ar_byVar1: ARRAY [0..16] OF BYTE:=[16#3A,16#30,16#32,16#30,16#31,16#30,
16     16#35,16#30,16#30,16#30,16#30,16#30,
17     16#31,16#46,16#37,16#0D,16#0A];
18     tTimeout_Var: TIME :=T#500ms;
19 END_VAR

```

```

1  IF bVar0 THEN
2      bVar0:=FALSE;
3      bExecute_Var:=TRUE;
4      RxMode_Var:=DL_COM_AX3.DFB_COMRS_MODE.SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR;
5      ParaSet_Var.pSpecificStartChar:=ADR(pSpecificStartChar_Var);
6      ParaSet_Var.byStartCharAmt:=byStartCharAmt_Var;
7      ParaSet_Var.pSpecificEndChar:=ADR(pSpecificEndChar_Var);
8      ParaSet_Var.byEndCharAmt:=byEndCharAmt_Var;
9      ParaSet_Var.pReadBuf:=ADR(ar_byVar0);
10     ParaSet_Var.pWriteBuf:=ADR(ar_byVar1);
11     ParaSet_Var.uiWriteLen:=SIZEOF(ar_byVar1);
12     ParaSet_Var.tTimeout:=tTimeout_Var;
13 END_IF
14 IF bDone_Var THEN
15     bExecute_Var:=FALSE;
16 END_IF
17 FB0(
18     bExecute:=bExecute_Var ,
19     byComPort:=DL_ModbusComMaster.DFB_AX3XX_COM_PORT_NUM.RS485 ,
20     RxMode:=RxMode_Var ,
21     ParaSet:=ParaSet_Var ,
22     bDone=>bDone_Var ,
23     bBusy=>bBusy_Var ,
24     bError=>bError_Var ,
25     ErrorId=>ErrorID_Var ,
26     uiRcvLen=>uiRcvLen_Var );
    
```

于 AX-3 系列控制器中，COM Port 代号定义可以至 Library Manager 查询，如下图所示。



- 支持机种
 - AX 系列 (暂不支持 AX-8)
- 函式库
 - DL_COM_AX3.library

7.2 DFB_ModbusComChannel

DFB_ModbusComChannel：串行通讯从站 Channel 控制指令。

FB/FC	指令	指令图	ST 语法
FB	DFB_ModbusComChannel		<pre>DFB_ModbusComChannel(Slave:= , bExecute:= , bAbort:= , iChannelIndex:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusError=>);</pre>

● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)
Slave	Delta Modbus slave device	DFB_ModbusComSlave	--

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
bAbort	无作用	BOOL	--
iChannelIndex	Channel 编号	INT	0 ~ 9 (0)

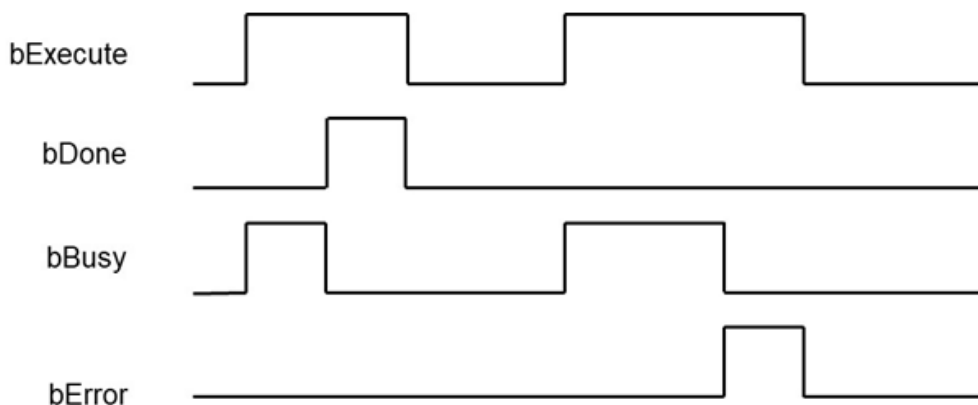
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
bAborted	无作用	BOOL	--
ModbusError	错误码	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (UNDEFINED)

■ 输出更新时间点

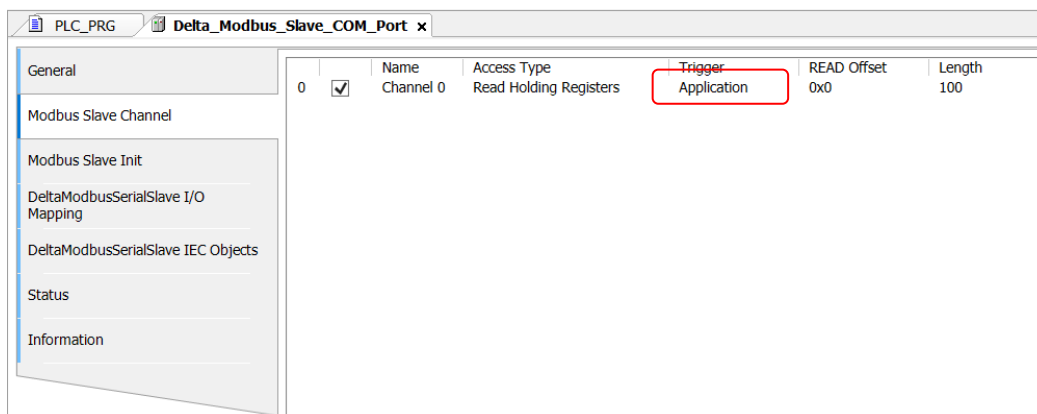
名称	输出参数上缘时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ModbusError		

● 参数时序图



● 功能说明

当串行通讯从站通道设定为 Application 的触发模式时，则用户可通过 FB 指令 (DFB_ModbusComChannel) 来触发此通道的 MODBUS request 动作。

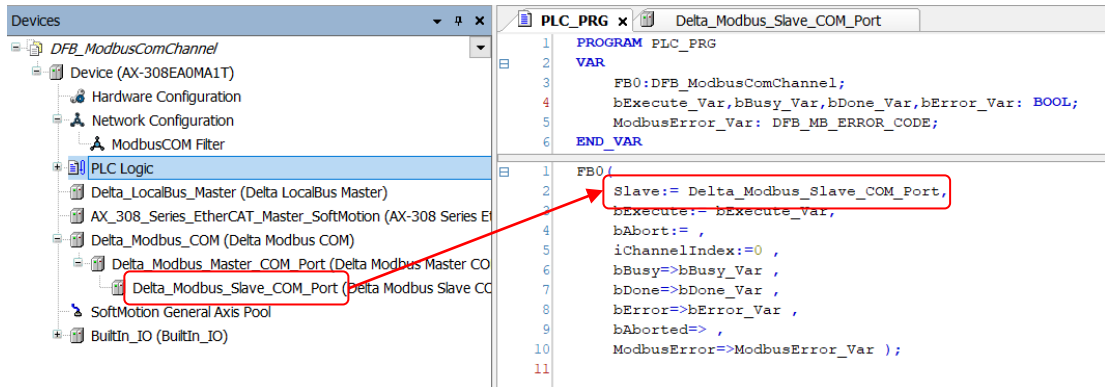


注 1：串行通讯从站相关设定，可参考 AX-3 系列操作手册第 9.2 节「串行 (Serial) 通讯」。

注 2：在使用时，此 Channel 必须设定成 Enable

- 范例程序

此范例利用 FB 指令 (DFB_ModbusComChannel) 来触发串行通讯数据交换。



注：Slave 输入脚位为 Modbus slave device 名称。

- 支持机种

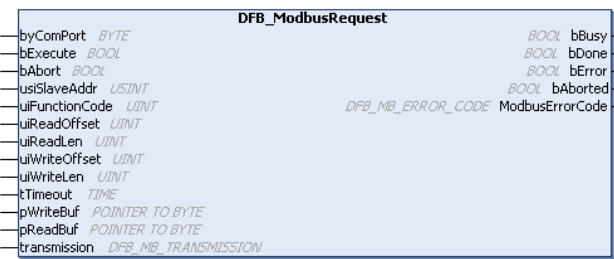
- AX-308E

- 函式库

- DL_ModbusComMaster_AX3.library

7.3 DFB_ModbusRequest

DFB_ModbusRequest：串行通讯命令。

FB/FC	指令	指令图	ST 语法
FB	DFB_ModbusRequest		<pre> DFB_ModbusRequest(byComPort:= , bExecute:= , bAbort:= , usiSlaveAddr:= , uiFunctionCode:= , uiReadOffset:= , uiWriteOffset:= , uiWriteLen:= , tTimeout:= , pWriteBuf:= , pReadBuf:= , transmission:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusErrorCode=>); </pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
byComPort*1	串行通讯端口号	BYTE	(0xFF)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
bAbort	无作用	BOOL	--
usiSlaveAddr	从站站号	USINT	1~247
uiFunctionCode	Modbus 功能码	DFB_MB_FUNC_CODE	支持的功能码： 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

名称	功能	数据类型	设定值范围 (默认值)
			0x10 : Write Multiple Registers 0x17 : Read/Write Multiple Registers (0x03)
uiReadOffset	欲读取的内存起始地址	UINT	0 ~ 65535 (0)
uiReadLen	读取的内存长度	UINT	Coil : 1 ~ 1920 Register : 1 ~ 120 (1)
uiWriteOffset	欲写入的内存起始地址	UINT	0 ~ 65535 (0)
uiWriteLen	写入的内存长度	UINT	Coil : 1 ~ 1920 Register : 1 ~ 120 (1)
tTimeout*2	通讯超时时间	TIME	T#0ms ~ T#49d17h2m47s295ms 0 : 无超时时间 (T#100ms)
pWriteBuf	欲传送报文的内存地址	POINTER TO BYTE	--
pReadBuf	接收报文所储存的内存地址	POINTER TO BYTE	--
Transmission*3	通讯格式	DFB_MB_TRANSMISSION	0 : ASCII 1 : RTU (ASCII)

*注 :

1. 各机种所定义的串行通讯端口号不尽相同，需依照其定义的方式来设定。
2. 通讯超时时间设定应大于 `mdbus Task` 所设定的 `Cyclic` 时间。
3. 如果选择 RTU 模式时，串行端口通讯格式的 `Data bit` 必须设定为 8。

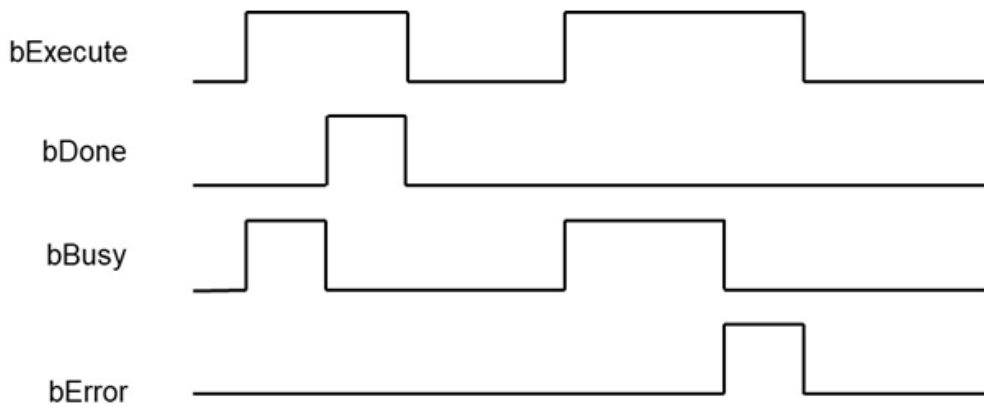
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
bAborted	无作用	BOOL	--
ModbusError	错误码	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (DFB_UNDEFINED)

● 输出更新时间点

名称	输出参数上缘时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ModbusError		

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_ModbusRequest) 发送标准 Modbus 通讯报文，在使用本指令之前，必须要把主机 COM port 设定完毕并新增 Delta_Modbus_Master_COM_Port device。(详细步骤请参考 AX-3 系列操作手册第 9.2 节「串行 (Serial Port) 通讯」)

- 范例程序

此范例利用 FB 指令 (DFB_ModbusRequest) 发送标准 Modbus 命令读取从站 (Slave address = 2) 内存起始地址 0x0000 · 长度 100 words 的数据 (Holding Registers)。

```

PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   bExecute_Var,bBusy_Var,bDone_Var,bError_Var: BOOL;
4   usiSlaveAddr_Var: USINT :=2 ;
5   ar_wVar0: ARRAY[0..200]OF WORD;
6   FB0: DFB_ModbusRequest;
7   ModbusErrorCode_Var: DFB_MB_ERROR_CODE;
8 END_VAR

1 FB0(
2   byComPort:=DL_ModbusComMaster_AX3.DFB_AX3_COM_PORT_NUM.RS485 ,
3   bExecute:=bExecute_Var ,
4   bAbort:= ,
5   usiSlaveAddr:=usiSlaveAddr_Var ,
6   uiFunctionCode:=DFB_MB_FUNC_CODE.READ_HOLDING_REGISTERS ,
7   uiReadOffset:=0 ,
8   uiReadLen:=100 ,
9   uiWriteOffset:= ,
10  uiWriteLen:= ,
11  tTimeout:=T#500MS ,
12  pWriteBuf:= ,
13  pReadBuf:=ADR(ar_wVar0) ,
14  transmission:= ,
15  bBusy=>bBusy_Var ,
16  bDone=>bDone_Var ,
17  bError=>bError_Var ,
18  bAborted=> ,
19  ModbusErrorCode=>ModbusErrorCode_Var );

```

- 支持机种

- AX-308E

- 函式库

- DL_ModbusComMaster_AX3.library

7.4 DFB_ModbusRequest2

DFB_ModbusRequest2：串行通讯命令。

FB/FC	指令	指令图	ST 语法
FB	DFB_ModbusRequest2	<p>The diagram shows the DFB_ModbusRequest2 instruction with the following parameters:</p> <ul style="list-style-type: none"> Inputs: bExecute (BOOL), bAbort (BOOL), byComPort (BYTE), usiSlaveAddr (USINT), ModbusCommand (DFB_MODBUS_COMMAND), tResponseTimeout (TIME), uiSendTimeout (UINT), pSendData (POINTER TO BYTE), pRecvData (POINTER TO BYTE), transmission (DFB_MB_TRANSMISSION). Outputs: bDone (BOOL), bBusy (BOOL), bError (BOOL), bAborted (BOOL), uiDataLength (BYTE), ModbusErrorCode (DFB_MB_ERROR_CODE). 	<pre>DFB_ModbusRequest2(bExecute:= , bAbort:= , byComPort:= , usiSlaveAddr:= , ModbusCommand:= , tResponseTimeout:= , uiSendTimeout:= , pSendData:= , pRecvData:= , transmission:= , bDone=> , bBusy=> , bError=> , bAborted=> , uiDataLength=> , ModbusErrorCode=>);;</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
bAbort	无作用	BOOL	--
byComPort*1	串行通讯端口号	BYTE	(0xFF)
usiSlaveAddr	从站站号	USINT	1~247
ModbusCommand	Modbus 参数设定	DFB_MODBUS_ COMMAND	--
tResponseTimeout*2	通讯逾时时间	TIME	T#0ms ~ T#49d17h2m47s295ms 0 : 无逾时时间 (T#100ms)
uiSendTimeout	无作用	UINT	(0)
pSendData	欲传送报文的内存地址	POINTER TO BYTE	--
pRecvData	接收报文所储存的内存地址	POINTER TO BYTE	--

名称	功能	数据类型	设定值范围 (默认值)
Transmission*3	通讯格式	DFB_MB_TRANSMISSION	0 : ASCII 1 : RTU (ASCII)

*注 :

1. 各机种所定义的串行通讯端口号不尽相同，需依照其定义的方式来设定。
2. 通讯逾时时间设定应大于 mdbus Task 所设定的 Cyclic 时间。
3. 如果选择 RTU 模式时，串行端口通讯格式的 Data bit 必须设定为 8。

● DFB_MODBUS_COMMAND

名称	功能	数据类型	输出值范围 (默认值)
uiFunctionCode	Modbus 功能码	DFB_MB_FUNC_CODE	支持的功能码： 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	欲读取的内存起始地址	UINT	0 ~ 65535 (0)
uiReadLen	读取的内存长度	UINT	Coil : 1 ~ 1920 Register : 1 ~ 120 (1)
uiWriteOffset	欲写入的内存起始地址	UINT	0 ~ 65535 (0)
uiWriteLen	写入的内存长度	UINT	Coil : 1 ~ 1920 Register : 1 ~ 120 (1)

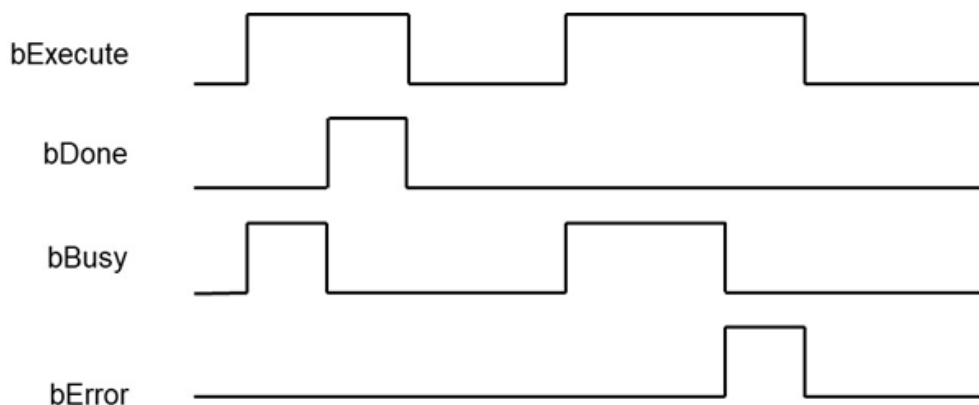
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
bAborted	无作用	BOOL	--
uiDataLength	接收的报文长度	BYTE (单位 : BYTE)	(0)
ModbusError	错误码	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (DFB_UNDEFINED)

● 输出更新时间点

名称	输出参数上缘时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ModbusError		

● 参数时序图



- 功能说明

用户可通过 FB 指令 (DFB_ModbusRequest2) 发送标准 Modbus 通讯报文。在使用本指令之前，必须要把主机 COM port 设定完毕并新增 Delta_Modbus_Master_COM_Port device。(详细步骤请参考 AX-3 系列操作手册第 9.2 节「串行 (Serial Port) 通讯」)

- 范例程序

此范例利用 FB 指令 (DFB_ModbusRequest2) 发送标准 Modbus 命令，读取从站 (Slave address = 2) 内存起始地址 0x0000，长度 100 words 的数据 (Holding Registers)。

```

PLC_PRG x
2  VAR
3  bVar0: BOOL:=TRUE;
4  bExecute_Var,bBusy_Var,bDone_Var,bError_Var: BOOL;
5  usiSlaveAddr_Var: USINT :=2 ;
6  ar_wVar0: ARRAY[0..200]OF WORD;
7  FB0: DFB_ModbusRequest2;
8  ModbusErrorCode_Var: DFB_MB_ERROR_CODE;
9  ModbusCommand_Var: DFB_MODBUS_COMMAND;
10 uiDataLength_Var: UINT;
11 END_VAR

1  IF bVar0 THEN
2  bVar0:=FALSE;
3  ModbusCommand_Var.uiFunctionCode:=3;
4  ModbusCommand_Var.uiReadLen:=100;
5  ModbusCommand_Var.uiReadOffset:=0;
6  END_IF
7  FB0(
8  bExecute:=bExecute_Var ,
9  bAbort:= ,
10 byComPort:=DL_ModbusComMaster.DFB_AX3XX_COM_PORT_NUM.RS485 ,
11 usiSlaveAddr:=usiSlaveAddr_Var ,
12 ModbusCommand:=ModbusCommand_Var ,
13 tResponseTimeout:=T#500MS ,
14 uiSendTimeout:= ,
15 pSendData:= ,
16 pRecvData:=ADR(ar_wVar0) ,
17 transmission:=DFB_MB_TRANSMISSION.ASCII ,
18 bDone=>bDone_Var ,
19 bBusy=>bBusy_Var ,
20 bError=>bError_Var ,
21 bAborted=> ,
22 uiDataLength=>uiDataLength_Var ,
23 ModbusErrorCode:=ModbusErrorCode_Var );
24

```

- 支持机种

- AX-308E

- 函式库

- DL_ModbusComMaster_AX3.library

7.5 错误码与故障排除

● DFB_COM_ERROR_CODE

说明	错误原因	处置方式
DFB_NO_ERROR	无错误	--
DFB_RESPONSE_TIMEOUT	从站回应超时	<ul style="list-style-type: none"> ● 请确认通讯超时时间设定，是否恰当 ● 请检查通讯线路是否正确
DFB_REQUEST_FAILED_TO_SEND	COM Port 异常	请联络原厂
DFB_INVALID_COMPORT	COM port 设定错误	请检查 COM port 设定是否正确
DFB_INVALID_BUFFER	接收与传送报文的内存地址设定错误	请检查下列参数设定是否正确 <ul style="list-style-type: none"> ● ParaSet.pWriteBuf ● ParaSet.pReadBuf
DFB_INVALID_LENGTH	数据长度设定错误	请确认下列参数设定是否正确 <ul style="list-style-type: none"> ● ParaSet.uiReadLen ● ParaSet.uiReadBufSize ● ParaSet.uiWriteLen
DFB_NO_MASTER_CONFIG	Delta_Modbus_Master_COM_Port device 不存在	请确认是否已经新增 Delta_Modbus_Master_COM_Port device 于 device 树形图内
DFB_MEMORY_NOT_ENOUGH	系统内存不足	请确认程序大小是否超出可使用上限
DFB_INVALID_MODE	DFB_COMRS 接收模式设定错误	请检查 RxMode 设定是否正确
DFB_INVALID_SETTING	参数设定错误	请确认下列参数设定是否正确 <ul style="list-style-type: none"> ● ParaSet.tTimeout ● ParaSet.uiDiscontinuousTime ● ParaSet.byEndCharAmt ● ParaSet.byStartCharAmt ● ParaSet.uiSpecificRxLen
DFB_INVALID_CHAR_BUFFER	字符内存地址设定错误	请确认下列参数是否正确 <ul style="list-style-type: none"> ● ParaSet.pSpecificStartChar ● ParaSet.pSpecificEndChar
DFB_UNDEFINED	未定义或尚未执行	待 FB 指令执行完成

● DFB_MB_ERROR_CODE

说明	错误原因	处置方式
DFB_NO_ERR	无错误	-
DFB_ILLEGAL_FUNCTION	未支持的功能码	请检查所使用的功能码是否正确
DFB_ILLEGAL_DATA_ADDRESS	欲写入/读取的内存地址不合法	请确认所欲写入/读取的内存地址是否正确

说明	错误原因	处置方式
DFB_ILLEGAL_DATA_VALUE	从站回复的报文异常	请确认线路是否正常或是布线方式是恰当
DFB_RESPONSE_TIMEOUT	从站通讯超时	<ul style="list-style-type: none"> ● 请确认发送报文的周期是否小于从站响应的能力 ● 请检查通讯线路是否正确
DFB_RESPONSE_CRC_ERROR	从站回复的报文异常(检查码错误)	请确认从站响应报文格式是否正确
DFB_RESPONSE_WRONG_SLAVE	从站回复的报文异常(站号错误)	请确认从站响应报文格式是否正确
DFB_RESPONSE_WRONG_FUNCTIONCODE	从站回复的报文异常(功能码错误)	请确认从站响应报文格式是否正确
DFB_REQUEST_FAILED_TO_SEND	报文发送失败	请联络原厂
DFB_RESPONSE_INVALID_PROTOCOL	从站回复的报文异常(非标准 Modbus 格式)	请确认从站响应报文格式是否正确
DFB_RESPONSE_INVALID_HEAD	从站回复的报文异常(报文长度错误)	请确认从站响应报文格式是否正确
DFB_INVALID_CHANNEL_INDEX	非法的从站通道编号	请检查从站通讯编号是否正确
DFB_CHANNEL_SETTING_NOT_SUPPORT	从站通道触发条件非 "Application" 模式	请检查从站通道触发条件是否为 "Application" 模式
DFB_INVALID_COMPORT	控制器串行通讯端口号设定错误	请检查控制器串行通讯端口号是否正确
DFB_INVALID_BUFFER	接收与传送报文的内存地址设定错误	<p>请检查下列参数设定是否正确</p> <p>ModbusRequest :</p> <ul style="list-style-type: none"> ● pWriteBuf ● pReadBuf <p>ModbusRequest2 :</p> <ul style="list-style-type: none"> ● ModbusCommand.pWriteBuf ● ModbusCommand.pReadBuf
DFB_INVALID_LENGTH	数据长度设定错误	<p>请检查下列参数设定是否正确</p> <p>ModbusRequest :</p> <ul style="list-style-type: none"> ● uiWriteLen ● uiReadLen <p>ModbusRequest2 :</p> <ul style="list-style-type: none"> ● ModbusCommand.uiWriteLen ● ModbusCommand.uiReadLen
DFB_INVALID_SLAVE_ADDRESS	从站站号设定错误	请确认从站站号为 1~247

说明	错误原因	处置方式
DFB_INVALID_FUNCTION_CODE	uiFunctionCode 设定不合法	请检查 uiFunctionCode 设定值是否正确
DFB_NO_MDBSCOM_CONFIG	Delta_Modbus_COM device 不存在	请确认是否已经新增 Delta_Modbus_COM device 于 device 树形图内
DFB_NO_MASTER_CONFIG	Delta_Modbus_Master_COM_Port device 不存在	请确认是否已经新增 Delta_Modbus_Master_COM_Port device 于 device 树形图内
DFB_MB_ERROR_CODE_MEMORY_NOT_ENOUGH	系统内存不足	请确认程序大小是否超出可使用上限
DFB_UNDEFINED	未定义或尚未执行	待 FB 指令执行完成


第8章 网络通讯指令

目录

8.1	DFB_TCP_Client	8-2
8.2	DFB_TCP_Server	8-8
8.3	DFB_UDP_Socket	8-14
8.4	DFB_ModbusTCPChannel	8-20
8.5	DFB_ModbusTCPRequest	8-23
8.6	错误码与故障排除	8-27

8.1 DFB_TCP_Client

DFB_TCP_Client : Socket Client 通讯指令。

FB/FC	指令	指令图	ST 语法
FB	DFB_TCP_Client		<pre>DFB_TCP_Client(bEnable:= , SocketInfo:= , bSend:= , bRecvRestart:= , bBusy=> , bConnected=> , bSent=> , bRcvd=> , bError=> , ErrorID=> , Status=> , uiRcvdLen=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bEnable	执行功能块 (*1 , *2)	BOOL	True/False (False)
SocketInfo	Server 端联机信息	tcpClientSocketInfo	--
bSend	发送报文 (上升沿触发)	BOOL	True/False (False)
bRecvRestart	重新开始接收报文*2 (上升沿触发)	BOOL	True/False (False)

*注 :

1. 当执行此功能块时，即开始建立 TCP 联机，当联机建立成功，则输出参数 bConnected 为 ON。
2. 当执行此功能块时，即开始接收网络报文，待接收完成后，bRcvd 即为 ON，且停止接收报文，如将 bRecvRestart 为 ON，则重新开始接收网络报文。

■ tcpClientsocketInfo

名称	功能	数据类型	设定值范围 (默认值)
byIPAddr	Server 端 IP 地址	ARRAY [0..3] OF BYTE	--
uiLPort	本机通讯端口	UINT	0 : 使用随机通讯端口号 0 ~ 65535 (0)

名称	功能	数据类型	设定值范围 (默认值)
uiRPort	远程通讯端口	UINT	0 : 非法值 1 ~ 65535 (0)
uiTimeout*	通讯响应超时时间 (单位 : ms)	UINT	0 : 无超时 1 ~ 65535 (0)
uiKeepAlive Timeout	Socket 联机保持时间 (单位 : sec)	UINT	0 : 无超时 1 ~ 65535 (0)
bReconnect	自动重新联机功能	BOOL	True/False (False) True : 当联机超时或联机中断发生后 , 即自动尝试重新建立联机。 False : 当联机超时或中断 , 则输出参数 bError 为 ON 。
uiSetValue	recvCondition 设定值	UINT	(0)
pSendBuf	欲传送报文的内存地址	POINT TO BYTE	--
uiSendLen	传送报文的长度 (单位 : Byte)	UINT	0 ~ 8192 (0)
pRecvBuf	接收报文所储存的内存 地址	POINT TO BYTE	--
uiRecvBufSize	接收报文的储存内存大 小 (单位 : Byte)	UINT	0 ~ 8192 (0)
recvCondition	接收完成条件	DFB_SOCK_RECV _MODE	(DFB_SOCKET_NO_ RECEIVING)

*注 : 通讯响应超时时间是从接收到第一次报文后 , 即开始进行超时时间的计算 , 如未收到任何报文 , 则不会进行超时时间的计算。

■ DFB SOCK_RECV_MODE

名称	说明
DFB SOCK_MODE_NO_RECEIVING	不接收数据模式。
DFB SOCK_MODE_SPECIFIC_LENGTH	报文长度模式： 接收特定长度报文，报文长度可利用 uiSetValue 进行设定。 (长度单位：Byte)
DFB SOCK_MODE_SPECIFIC_SINGLE_CHAR	终止符模式： 持续接收报文直到接收到终止符 (1 Byte) 为止，终止符可利用 uiSetValue 进行设定。 e.g. : 如 uiSetValue 设定为 16#0000D0A，则终止符为 16#0A。 (*1, *2)
DFB SOCK_RECV_MODE_DFB SOCK_MODE_SPECIFIC_TWO_CHARS	双终止符模式： 持续接收报文直到接收到终止符 (2 Bytes) 为止，终止符可利用 uiSetValue 进行设定 e.g. : 如 uiSetValue 设定为 16#0000D0A，则终止符为 16#0D0A。 (*1, *2)
DFB SOCK_RECV_MODE_DFB SOCK_MODE_SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR	起始字符与终止符模式： 当接收到起始字符后开始接收报文，直接接收到终止符后停止接收报文，起始字符与终止符可以利用 uiSetValue 进行设定。 e.g. : 如 uiSetValue 设定为 16#00003A0A，则起始字符为 16#3A，终止符为 16#0A。 (*1, *2)
DFB SOCK_RECV_MODE_DFB SOCK_MODE_ANY_LENGTH	任意长度模式： 此接收模式在接收到一笔完整个网络报文后即停止接收。 (*1)

*注：

1. 当所接受到的报文长度达到 uiRecvBufSize 所设定的大小时，即完成本次接收。
2. 报文长度计算包含起始与终止符。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bConnected	TCP 联机标志	BOOL	True/False (False)

名称	功能	数据类型	输出值范围 (默认值)
bSent	发送完成标志	BOOL	True/False (False)
bRcvd	接收完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DFB_SOCKET_ERROR	(DFB SOCK_NO_ERROR)
Status	Socket 指令执行状态	DFB_SOCKET_STATUS	(SOCKET_CLOSED)
uiRcvLen	目前所接收到报文的长度	UINT (单位 : Byte)	(0)

■ DFB_SOCKET_STATUS

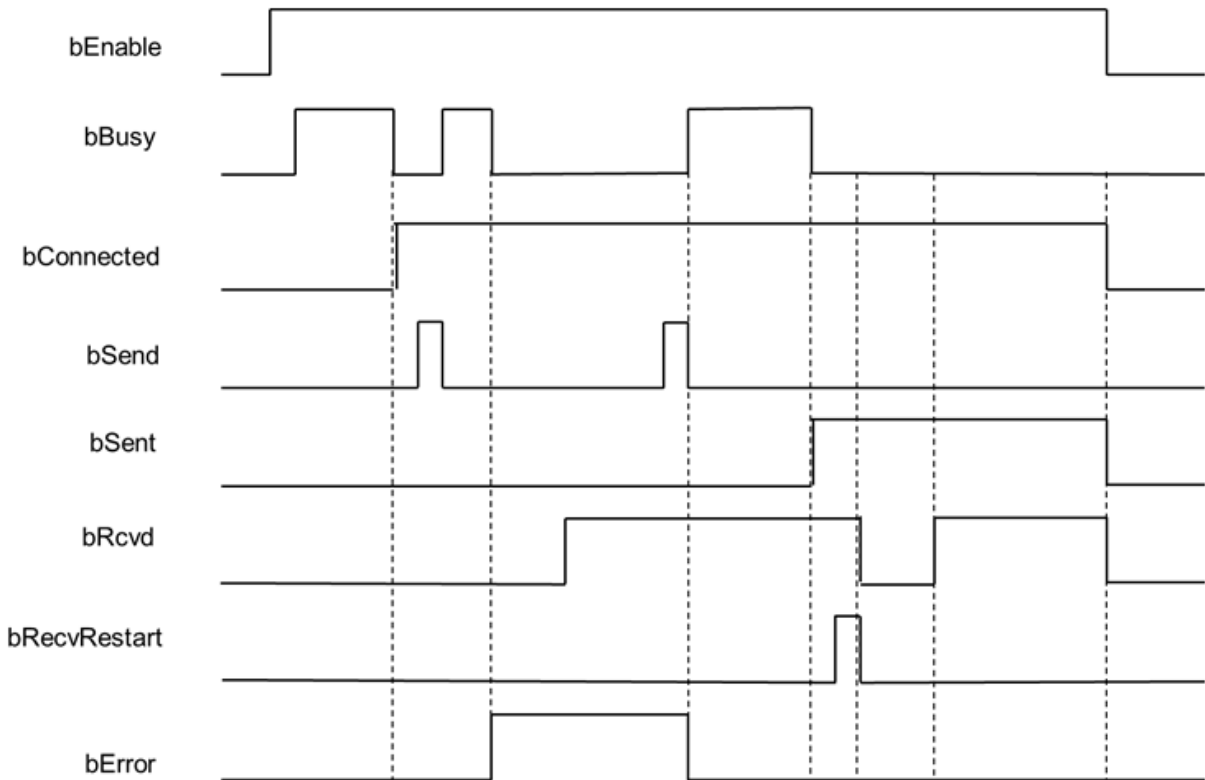
名称	说明	适用指令
SOCKET_CLOSED	SOCKET 联机已关闭	TCP / UDP
SOCKET_CONNECTING	SOCKET 联机建立中	TCP
SOCKET_CONNECTED	SOCKET 联机建立成功	TCP
SOCKET_SENDING	SOCKET 报文发送中	TCP / UDP
SOCKET_SENT	SOCKET 报文发送完成	TCP / UDP
SOCKET_RECEIVED	SOCKET 报文接收完成	TCP / UDP
SOCKET_ERROR	SOCKET 发生错误	TCP / UDP
SOCKET_ABORTED	SOCKET 联机被关闭	TCP
SOCKET_READY	SOCKET 建立成功	UDP

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bBusy	<ul style="list-style-type: none"> ● bEnable 上升沿时 ● bSend 上升沿时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● 动作执行完毕时
bConnected	<ul style="list-style-type: none"> ● TCP 联机建立完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● Server 端停止 TCP 联机时
bSent	<ul style="list-style-type: none"> ● TCP 报文发送完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● bSend 上升沿时

名称	输出参数上升沿时机	输出参数下降沿时机
bRcvd	<ul style="list-style-type: none"> ● TCP 报文接收完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● bRecvRestart 上升沿时
bError	<ul style="list-style-type: none"> ● 当 FB 指令执行错误或是输入值错误时 	<ul style="list-style-type: none"> ● bEnable 下降沿时

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_TCP_Client) 建立 TCP 联机并发送/接收 TCP 报文。

● 范例程序

此范例利用 FB 指令 (DFB_TCP_Client) 与 Server 端 (IP 地址 : 192.168.1.111 · Port : 502) 建立联机 · 并发送 1000 Bytes 报文 · 并预期 Server 端会回传 1000 Bytes 报文。

```

1  PROGRAM TCP_Client
2  VAR
3    bVar0 : BOOL := TRUE;
4    FB0: DFB_TCP_Client;
5    Server_IP_Address : ARRAY [0..3] OF BYTE := [192,168,1,111];
6    bEnable_Var,bSend_Var,bRestart_Var,bBusy_Var,bConnected_Var,bSent_Var,bRcvd_Var,bError_Var : BOOL;
7    RemoteInfo_Var : tcpClientSocketInfo;
8    ErrorID_Var : DFB_SOCKET_ERROR;
9    Status_Var : DFB_SOCKET_STATUS;
10   uiRcvdLen_Var : UINT;
11   ar_byVar0,ar_byVar1: ARRAY[0..2000] OF BYTE;
12 END_VAR
13

```

```
1 IF bVar0 THEN
2   RemoteInfo_Var.byIPAddr[0]:=192;
3   RemoteInfo_Var.byIPAddr[1]:=168;
4   RemoteInfo_Var.byIPAddr[2]:=1;
5   RemoteInfo_Var.byIPAddr[3]:=111;
6   RemoteInfo_Var.uiRPort:=502;
7   RemoteInfo_Var.uiLPort:=2000;
8   RemoteInfo_Var.uiTimeout:=3000; //ms
9   RemoteInfo_Var.uiKeepAliveTimeout:=300; //sec
10  RemoteInfo_Var.bReconnect:=TRUE;
11  RemoteInfo_Var.pSendBuf:=ADR(ar_byVar0);
12  RemoteInfo_Var.pSendBuf:=1000;
13  RemoteInfo_Var.pRecvBuf:=ADR(ar_byVar1);
14  RemoteInfo_Var.uiRecvBufSize:=2000;
15  RemoteInfo_Var.uiSetValue:=1000;
16  RemoteInfo_Var.recvCondition:=DFB SOCK_RECV_MODE.DFB SOCK_MODE_SPECIFIC_LENGTH;
17  bEnable_Var:=TRUE;
18  bSend_Var:=TRUE;
19 END_IF
20 FBO(
21   bEnable:=bEnable_Var ,
22   SocketInfo:=RemoteInfo_Var ,
23   bSend:= bSend_Var,
24   bRecvRestart:= bRestart_Var,
25   bBusy=> bBusy_Var,
26   bConnected=> bConnected_Var,
27   bSent=> bSent_Var,
28   bRcvd=> bRcvd_Var,
29   bError=> bError_Var,
30   ErrorID=> ErrorID_Var,
31   Status=> Status_Var,
32   uiRcvdLen=> uiRcvdLen_Var);
```

- 支持机种
 - AX-308E

- 函式库
 - DL_EthernetLib_AX3.library

8.2 DFB_TCP_Server

DFB_TCP_Server : Socket Server 通讯指令。

FB/FC	指令	指令图	ST 语法
FB	DFB_TCP_Server	<p>The diagram shows the DFB_TCP_Server instruction with the following connections:</p> <ul style="list-style-type: none"> Inputs: bEnable (BOOL), SocketInfo (tcpServerSocketInfo), bSend (BOOL), bRecvRestart (BOOL). Outputs: bBusy (BOOL), bConnected (BOOL), bSent (BOOL), bRcvd (BOOL), bError (BOOL), ErrorID (DFB_SOCKET_ERROR), Status (DFB_SOCKET_STATUS), uiRcvdLen (UINT). 	<pre>DFB_TCP_Server(bEnable:= , SocketInfo:= , bSend:= , bRecvRestart:= , bBusy=> , bConnected=> , bSent=> , bRcvd=> , bError=> , ErrorID=> , Status=> , uiRcvdLen=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bEnable	执行功能块 (*1 · *2)	BOOL	True/False (False)
SocketInfo	Server 端通信设置	tcpServerSocketInfo	--
bSend	发送报文 (上升沿触发)	BOOL	True/False (False)
bRecvRestart	重新开始接收报文*2 (上升沿触发)	BOOL	True/False (False)

*注：

1. 当执行此功能块时，即开启 TCP 联机，待 Client 端建立联机后，则输出参数 bConnected 为 ON。
2. 当执行此功能块时，即开始接收网络报文，待接收完成后，bRcvd 即为 ON，且停止接收报文，如将 bRcvRestart 为 ON，则重新开始接收网络报文。

■ tcpServersocketInfo

名称	功能	数据类型	设定值范围 (默认值)
byIPAddr	允许建立联机的 Client 端 IP 地址	ARRAY [0..3] OF BYTE	[0.0.0.0] : 无限制

名称	功能	数据类型	设定值范围 (默认值)
uiLPort	本机通讯端口	UINT	0 : 非法值 1 ~ 65535 (0)
uiTimeout*	通讯响应超时时间 (单位 : ms)	UINT	0 : 无超时 1 ~ 65535 (0)
uiKeepAlive Timeout	联机保持时间 (单位 : sec)	UINT	0 : 无超时 1 ~ 65535 (0)
uiSetValue	recvCondition 设定值	UINT	(0)
pSendBuf	欲传送报文的内存地址	POINT TO BYTE	--
uiSendLen	传送报文的长度 (单位 : Byte)	UINT	0 ~ 8192 (0)
pRecvBuf	接收报文所储存的内存地址	POINT TO BYTE	--
uiRecvBufSize	接收报文的储存内存大小 (单位 : Byte)	UINT	0 ~ 8192 (0)
recvCondition	接收完成条件	DFB SOCK RECV_MODE	(DFB_SOCKET_NO_ RECEIVING)

*注：通讯响应超时时间是从接收到第一次报文后，即开始进行超时时间的计算，如未收到任何报文，则不会进行超时时间的计算。

■ DFB SOCK_RECV_MODE

名称	说明
DFB SOCK_MODE_NO_ RECEIVING	不接收数据模式。
DFB SOCK_MODE_ SPECIFIC_LENGTH	报文长度模式： 接收特定长度报文，报文长度可利用 uiSetValue 进行设定。 (长度单位 : Byte)

名称	说明
DFB_SOCK_MODE_SPECIFIC_SINGLE_CHAR	<p>终止符模式：</p> <p>持续接收报文直到接收到终止符（ 1 Byte ）为止，终止符可利用 uiSetValue 进行设定。</p> <p>e.g.：如 uiSetValue 设定为 16#0000D0A，则终止符为 16#0A。</p> <p>(*1，*2)</p>
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_TWO_CHARS	<p>双终止符模式：</p> <p>持续接收报文直到接收到终止符（ 2 Bytes ）为止，终止符可利用 uiSetValue 进行设定。</p> <p>e.g.：如 uiSetValue 设定为 16#0000D0A，则终止符为 16#0D0A。</p> <p>(*1，*2)</p>
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR	<p>起始字符与终止符模式：</p> <p>当接收到起始字符后开始接收报文，直接接收到终止符后停止接收报文，起始字符与终止符可以利用 uiSetValue 进行设定。</p> <p>e.g.：如 uiSetValue 设定为 16#00003A0A，则起始字符为 16#3A，终止符为 16#0A。</p> <p>(*1，*2)</p>
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_ANY_LENGTH	<p>任意长度模式：</p> <p>此接收模式在接收到一笔完整网络报文后即停止接收。</p> <p>(*1)</p>

*注：

1. 当所接受到的报文长度达到 uiRecvBufSize 所设定的大小时，即完成本次接收。
2. 报文长度计算包含了起始与终止符。

● 输出参数

名称	功能	数据类型	输出值范围（默认值）
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bConnected	TCP 联机标志	BOOL	True/False (False)
bSent	发送完成标志	BOOL	True/False (False)
bRcvd	接收完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DFB_SOCKET_ERROR	(DFB_SOCK_NO_ERROR)

名称	功能	数据类型	输出值范围 (默认值)
Status	Socket 指令执行状态	DFB_SOCKET_STATUS	(SOCKET_CLOSED)
uiRcvLen	目前所接收到报文的长度	UINT (单位 : Byte)	(0)

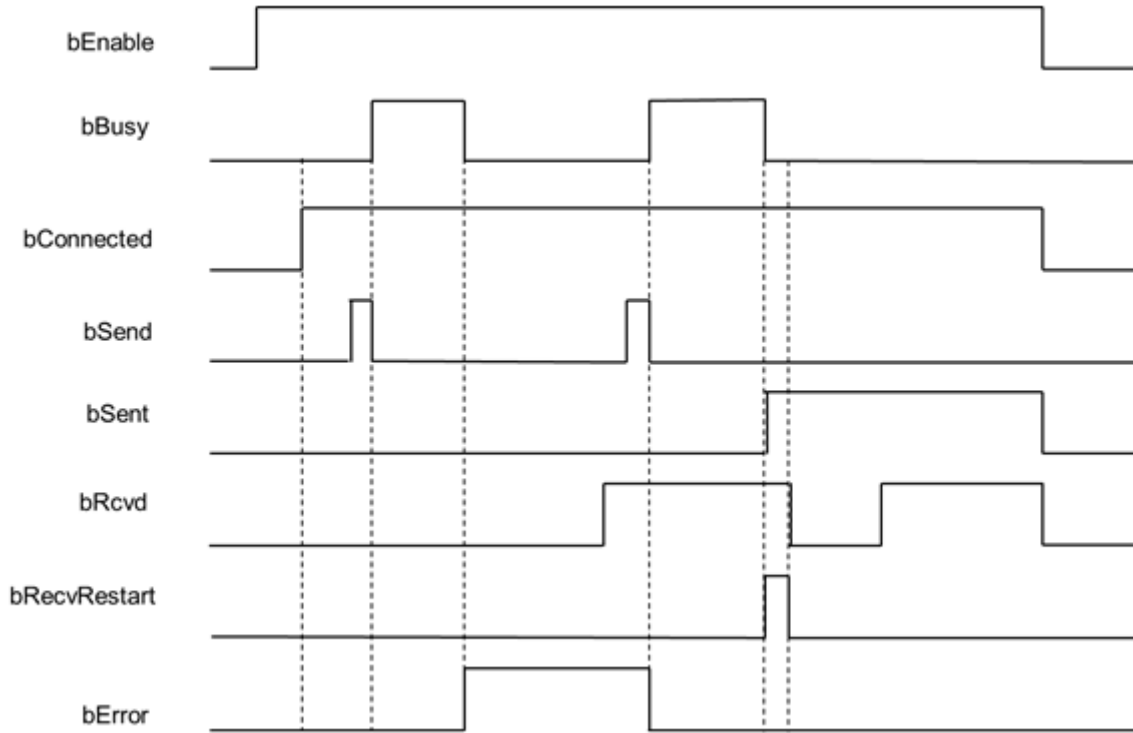
■ DFB_SOCKET_STATUS

名称	说明	适用指令
SOCKET_CLOSED	SOCKET 联机已关闭	TCP / UDP
SOCKET_CONNECTING	SOCKET 联机建立中	TCP
SOCKET_CONNECTED	SOCKET 联机建立成功	TCP
SOCKET_SENDING	SOCKET 报文发送中	TCP / UDP
SOCKET_SENT	SOCKET 报文发送完成	TCP / UDP
SOCKET_RECEIVED	SOCKET 报文接收完成	TCP / UDP
SOCKET_ERROR	SOCKET 发生错误	TCP / UDP
SOCKET_ABORTED	SOCKET 联机被关闭	TCP
SOCKET_READY	SOCKET 建立成功	UDP

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bBusy	<ul style="list-style-type: none"> ● bEnable 上升沿时 ● bSend 上升沿时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● 动作执行完毕时
bConnected	<ul style="list-style-type: none"> ● TCP 联机建立完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● Server 端停止 TCP 联机时
bSent	<ul style="list-style-type: none"> ● TCP 报文发送完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● bSend 上升沿时
bRcvd	<ul style="list-style-type: none"> ● TCP 报文接收完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● bRecvRestart 上升沿时
bError	<ul style="list-style-type: none"> ● 当 FB 指令执行错误或是输入值错误时 	<ul style="list-style-type: none"> ● bEnable 下降沿时

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_TCP_Server) 开启 TCP 联机并发送/接收 TCP 报文。

● 范例程序

范例利用 FB 指令 (DFB_TCP_Server) 开启 TCP (Port :502) 联机，限制 Client 端 IP 地址为 192.168.1.111，并预期接收报文长度为 1000 Bytes。

```

TCP_Server x
1 PROGRAM TCP_Server
2 VAR
3   bVar0 : BOOL := TRUE;
4   FB0 : DFB_TCP_Server;
5   bEnable_Var, bSend_Var, bRestart_Var, bBusy_Var, bConnected_Var, bSent_Var, bRcvd_Var, bError_Var : BOOL;
6   RemoteInfo_Var: tcpServerSocketInfo;
7   ErrorID_Var: DFB_SOCKET_ERROR;
8   Status_Var: DFB_SOCKET_STATUS;
9   uiRcvLen_Var: UINT;
10  by_arVar0, by_arVar1: ARRAY[0..2000] OF BYTE;
11  END_VAR
12
13 IF bVar0 THEN
14   bEnable_Var:=TRUE;
15   RemoteInfo_Var.byIPAddr[0]:=192;
16   RemoteInfo_Var.byIPAddr[1]:=168;
17   RemoteInfo_Var.byIPAddr[2]:=1;
18   RemoteInfo_Var.byIPAddr[3]:=111;
19   RemoteInfo_Var.uiLPort:=502;
20   RemoteInfo_Var.uiTimeout:=3000;
21   RemoteInfo_Var.uiKeepAliveTimeout:=300;
22   RemoteInfo_Var.pSendBuf:=ADR(by_arVar0);
23   RemoteInfo_Var.uiSendLen:=1000;
24   RemoteInfo_Var.pRecvBuf:=ADR(by_arVar1);
25   RemoteInfo_Var.uiRecvBufSize:=2000;
26   RemoteInfo_Var.uiSetValue:=1000;
27   RemoteInfo_Var.recvCondition:=DFB SOCK_RECV_MODE.DFB SOCK_MODE_SPECIFIC_LENGTH;
28 END_IF
    
```


```
18  FB0(  
19      bEnable:=bEnable_Var ,  
20      SocketInfo:=RemoteInfo_Var ,  
21      bSend:=bSend_Var ,  
22      bRecvRestart:=bRestart_Var ,  
23      bBusy:=bBusy_Var ,  
24      bConnected:=bConnected_Var ,  
25      bSent:=bSent_Var ,  
26      bRcvd:=bRcvd_Var ,  
27      bError:=bError_Var ,  
28      ErrorID:=ErrorID_Var ,  
29      Status:=Status_Var ,  
30      uiRcvdLen:=uiRcvLen_Var );
```

- 支持机种
 - AX-308E

- 函式库
 - DL_EthernetLib_AX3.library

8.3 DFB_UDP_Socket

DFB_UDP_Socket : Socket UDP 通讯指令。

FB/FC	指令	指令图	ST 语法
FB	DFB_UDP_Socket		<pre>DFB_UDP_Socket(bEnable:= , RemotelInfo:= , bSend:= , bRecvRestart:= , bBusy=> , bSent=> , bRcvd=> , bError=> , ErrorID=> , Status=> , uiRcvdLen=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bEnable	执行功能块*	BOOL	True/False (False)
SocketInfo	Socket 联机信息	udpSocketInfo	--
bSend	发送报文 (上升沿触发)	BOOL	True/False (False)
bRecvRestart	重新开始接收报文 (上升沿触发)	BOOL	True/False (False)

*注 : 当执行此功能块时 , 即开始接收网络报文 , 待接收完成后 , bRcvd 即为 ON , 且停止接收报文 , 如将 bRestart 为 ON , 则重新开始接收网络报文。

■ udpSocketInfo

名称	功能	数据类型	设定值范围 (默认值)
byIPAddr	建立/允许建立联机的从站 IP 地址	ARRAY [0..3] OF BYTE	[0.0.0.0] : 无限制
uiLPort (*1)	本机通讯端口	UINT	0 : 使用随机 Port 号发送报文 0 ~ 65535 (0)

名称	功能	数据类型	设定值范围 (默认值)
uiRPort (*1, *2)	远程通讯端口	UINT	0 : 接收任意 Port 号的 报文 1 ~ 65535 (0)
uiSetValue	recvCondition 设定值	UINT	(0)
pSendBuf	欲传送报文的内存地址	POINT TO BYTE	--
uiSendLen	传送报文的长度 (单位 : Byte)	UINT	0 ~ 8192
pRecvBuf	接收报文所储存的内存地址	POINT TO BYTE	--
uiRecvBufSize	接收报文的储存内存大小 (单位 : Byte)	UINT	0 ~ 8192 (0)
recvCondition	接收完成条件	DFB SOCK_RECV_ MODE	(DFB_SOCKET_NO_ RECEIVING)

*注 :

1. uiLPort 与 uiRPort , 不可以同时设定为 0 值。
2. 当 uiRPort 设定为 0 值 , 不允许发送 UDP 报文。

■ DFB SOCK_RECV_MODE

名称	说明
DFB SOCK_MODE_NO_ RECEIVING	不接收数据模式。
DFB SOCK_MODE_SPECIFIC_ LENGTH	报文长度模式 : 接收特定长度报文 , 报文长度可利用 uiSetValue 进行设定。 (长度单位 : Byte)
DFB SOCK_MODE_SPECIFIC_ SINGLE_CHAR	终止符模式 : 持续接收报文直到接收到终止符 (1 Byte) 为止 , 终止符可利用 uiSetValue 进行设定。 e.g. :如 uiSetValue 设定为 16#0000D0A , 则终止符为 16#0A。 (*1, *2)

名称	说明
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_TWO_CHARS	<p>双终止符模式：</p> <p>持续接收报文直到接收到终止符 (2 Bytes) 为止，终止符可利用 uiSetValue 进行设定</p> <p>e.g. : 如 uiSetValue 设定为 16#0000D0A，则终止符为 16#0D0A。</p> <p>(*1 · *2)</p>
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_SPECIFIC_START_CHAR_AND_SPECIFIC_END_CHAR	<p>起始字符与终止符模式：</p> <p>当接收到起始字符后开始接收报文，直接接收到终止符后停止接收报文，起始字符与终止符可以利用 uiSetValue 进行设定。</p> <p>e.g. : 如 uiSetValue 设定为 16#00003A0A，则起始字符为 16#3A，终止符为 16#0A。</p> <p>(*1 · *2)</p>
DFB_SOCK_RECV_MODE_DFB_SOCK_MODE_ANY_LENGTH	<p>任意长度模式：</p> <p>此接收模式在接收到一笔完整网络报文后即停止接收。</p> <p>(*1)</p>

*注：

1. 当所接受到的报文长度达到 uiRecvBufSize 所设定的大小时，即完成本次接收。
2. 报文长度计算包含起始与终止符。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bSent	发送完成标志	BOOL	True/False (False)
bRcvd	接收完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DFB_SOCKET_ERROR	(DFB_SOCK_NO_ERROR)
Status	Socket 指令执行状态	DFB_SOCKET_STATUS	(SOCKET_CLOSED)
uiRcvLen	目前所接收到报文的长度	UINT (单位 : Byte)	(0)

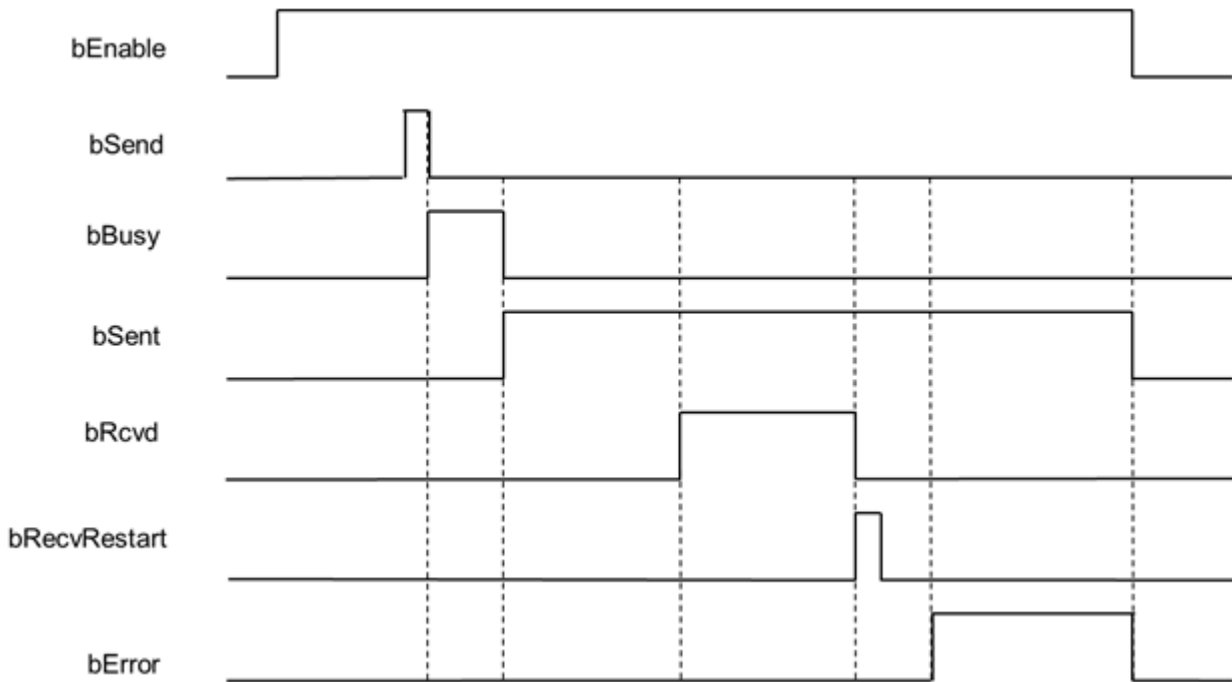
■ DFB_SOCKET_STATUS

名称	说明	适用指令
SOCKET_CLOSED	SOCKET 联机已关闭	TCP / UDP
SOCKET_CONNECTING	SOCKET 联机建立中	TCP
SOCKET_CONNECTED	SOCKET 联机建立成功	TCP
SOCKET_SENDING	SOCKET 报文发送中	TCP / UDP
SOCKET_SENT	SOCKET 报文发送完成	TCP / UDP
SOCKET_RECEIVED	SOCKET 报文接收完成	TCP / UDP
SOCKET_ERROR	SOCKET 发生错误	TCP / UDP
SOCKET_ABORTED	SOCKET 联机被关闭	TCP
SOCKET_READY	SOCKET 建立成功	UDP

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bBusy	<ul style="list-style-type: none"> ● bSend 上升沿时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● 动作执行完毕时
bSent	<ul style="list-style-type: none"> ● UDP 报文发送完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● bSend 上升沿时
bRcvd	<ul style="list-style-type: none"> ● UDP 报文接收完成时 	<ul style="list-style-type: none"> ● bEnable 下降沿时 ● bRecvRestart 上升沿时
bError	<ul style="list-style-type: none"> ● 当 FB 指令执行错误或是输入值错误时 	<ul style="list-style-type: none"> ● bEnable 下降沿时

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_UDP_Socket) 发送/接收 UDP 报文。

● 范例程序

范例利用 FB 指令 (DFB_UDP_Socket) 对 IP 地址 192.168.1.111 (Port : 3000) · 发送长度 1000 Bytes 的 UDP 报文 · 并预期接收长度 1000 Bytes 报文。

```

1  PROGRAM UDP
2  VAR
3    bVar0 : BOOL;
4    FB0: DL_EthernetLib_AX3.DFB_UDP_Socket;
5    bEnable_Var,bSend_Var,bRestart_Var,bBusy_Var,bSent_Var,bRcvd_Var,bError_Var : BOOL;
6    RemoteInfo_Var: DL_EthernetLib_AX3.udpSocketInfo;
7    ErrorID_Var: DL_EthernetLib_AX3.DFB_SOCKET_ERROR;
8    Status_Var: DL_EthernetLib_AX3.DFB_SOCKET_STATUS;
9    uiRcvdLen_Var: UINT;
10   by_arVar0,by_arVar1 : ARRAY[0..2000] OF BYTE;
11 END_VAR
12
13 IF bVar0 THEN
14   RemoteInfo_Var.byIPAddr[0]:=192;
15   RemoteInfo_Var.byIPAddr[1]:=168;
16   RemoteInfo_Var.byIPAddr[2]:=1;
17   RemoteInfo_Var.byIPAddr[3]:=111;
18   RemoteInfo_Var.uiLPort:=2000;
19   RemoteInfo_Var.uiRPort:=3000;
20   RemoteInfo_Var.pSendBuf:=ADR(by_arVar0);
21   RemoteInfo_Var.uiSendLen:=1000;
22   RemoteInfo_Var.pRecvBuf:=ADR(by_arVar1);
23   RemoteInfo_Var.uiRecvBufSize:=2000;
24   RemoteInfo_Var.uiSetValue:=1000;
25   RemoteInfo_Var.recvCondition:=DFB SOCK_RECV_MODE.DFB SOCK_MODE_SPECIFIC_LENGTH;
26 END_IF

```

```
15 FB0(  
16     bEnable:=bEnable_Var ,  
17     SocketInfo:=RemoteInfo_Var ,  
18     bSend:=bSend_Var ,  
19     bRecvRestart:=bRestart_Var ,  
20     bBusy=>bBusy_Var ,  
21     bSent=>bSent_Var ,  
22     bRcvd=>bRcvd_Var ,  
23     bError=>bError_Var ,  
24     ErrorID=>ErrorID_Var ,  
25     Status=>Status_Var ,  
26     uiRcvdLen=>uiRcvdLen_Var );
```

- 支持机种
 - AX-308E

- 函式库
 - DL_EthernetLib_AX3.library

8.4 DFB_ModbusTCPChannel

DFB_ModbusTCPChannel : Modus TCP 通讯从站 Channel 控制指令。

FB/FC	指令	指令图	ST 语法
FB	DFB_ModbusTCPChannel		<pre>DFB_ModbusTCPChannel(slave:= , bExecute:= , bAbort:= , iChannelIndex:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusError=>);</pre>

● 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)
Slave	Delta Modbus TCP slave device	DFB_ModbusTCPSlave	--

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
bAbort	无作用	BOOL	--
iChannelIndex	Channel 编号	INT	0 ~ 99 (0)

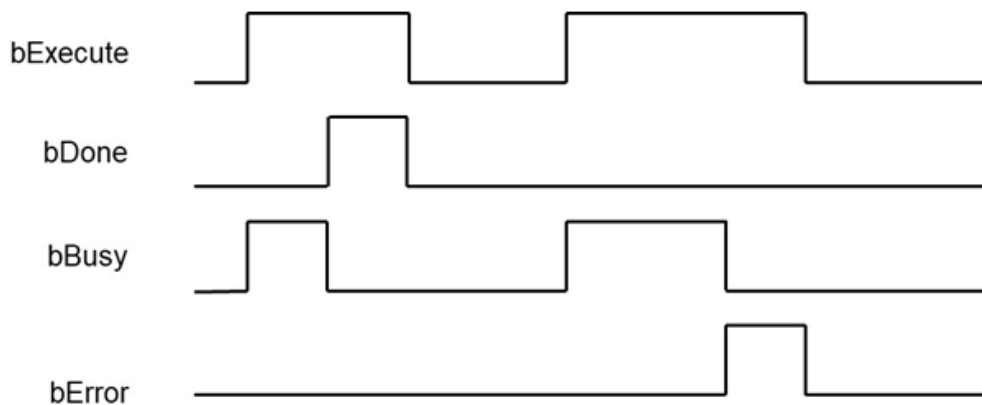
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
bAborted	无作用	BOOL	--
ModbusError	错误码	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (UNDEFINED)

■ 输出更新时间点

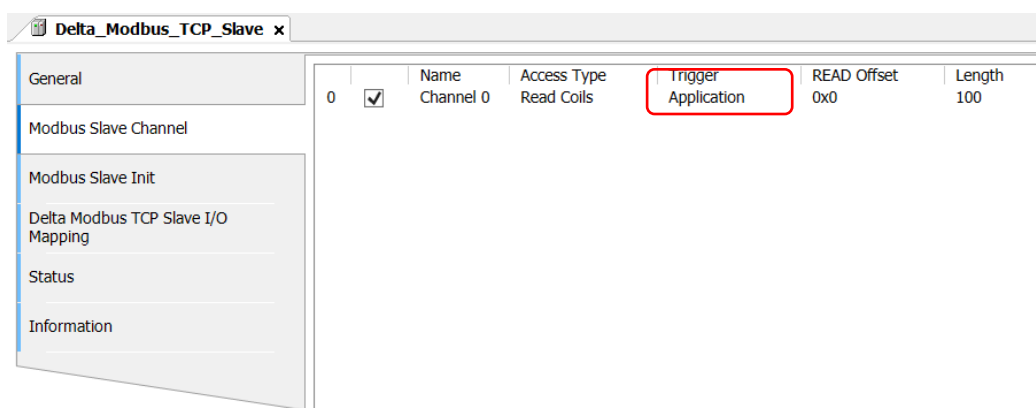
名称	输出参数上升沿时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ModbusError		

● 参数时序图



● 功能说明

当 Modbus TCP 通讯从站通道设定为 Application 的触发模式时，则用户可通过 FB 指令 (DFB_ModbusTCPChannel) 来触发此通道的 MODBUS TCP request 动作。

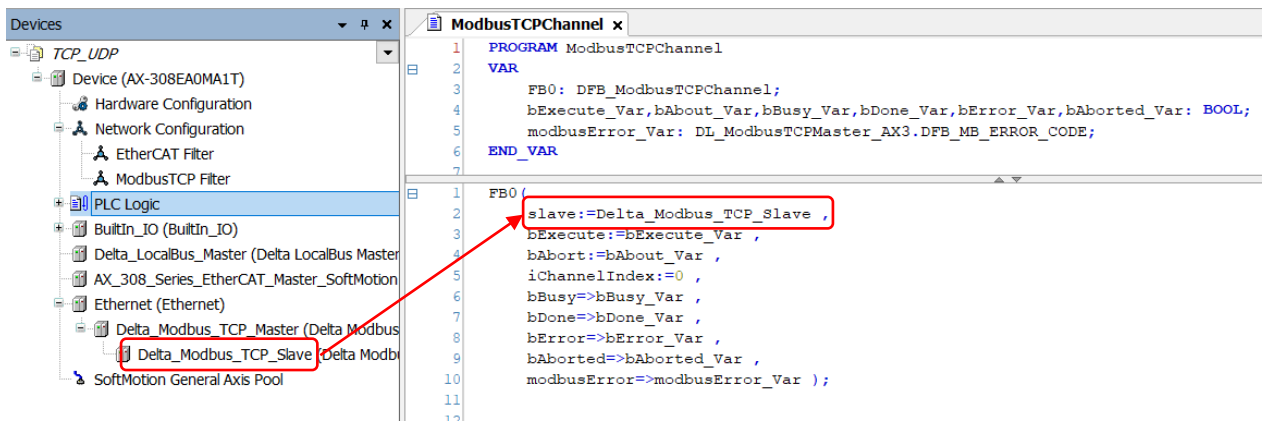


*注：

1. Modbus TCP 从站相关设定，可参考 AX-3 系列操作手册第 9.3 节「以太网 (Ethernet) 通讯」。
2. 在使用时，此 Channel 必须设定成 Enable。

● 范例程序

此范例利用 FB 指令 (DFB_ModbusTCPChannel) 来触发 Modbus TCP Slave 的 Channel 0 通道。



注：Slave 输入参数为 Delta_Modbus_TCP_Slave device 名称。

● 支持机种

- AX-308E

● 函式库

- DL_ModbusTCPMaster_AX3.library

8.5 DFB_ModbusTCPRequest

DFB_ModbusTCPRequest : Modbus TCP 通讯命令。

FB/FC	指令	指令图	ST 语法
FB	DFB_ModbusTCPRequest		<pre>DFB_ModbusTCPRequest(slave:= bExecute:= , bAbort:= , usiUnitID:= , ModbusCommand:= , pSendData:= , pRecvData:= , bBusy=> , bDone=> , bError=> , bAborted=> , ModbusError=>);;</pre>

- 输入/输出参数

名称	功能	数据类型	设定值范围 (默认值)
Slave	Delta Modbus TCP slave device	DFB_ModbusTCPSlave	--

- 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
bAbort	无作用	BOOL	--
usiSlaveAddr	从站站号	USINT	1~247
ModbusCommand	Modbus 参数设定	ModbusCommand	--
pSendData	欲传送报文的内存地址	POINTER TO BYTE	--
pRecvData	接收报文所储存的内存地址	POINTER TO BYTE	--

■ ModbusCommand

名称	功能	数据类型	输出值范围 (默认值)
FunctionCode	Modbus 功能码	DFB_MB_FUNC_CODE	支持的功能码： 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	欲读取的内存起始地址	UINT	0 ~ 65535 (0)
uiReadLen	读取的内存长度,	UINT	Coil : 1 ~ 1992 Register : 1 ~ 124 (1)
uiWriteOffset	欲写入的内存起始地址	UINT	0 ~ 65535 (0)
uiWriteLen	写入的内存长度	UINT	Coil : 1 ~ 1960 Register : 1 ~ 122 (1)

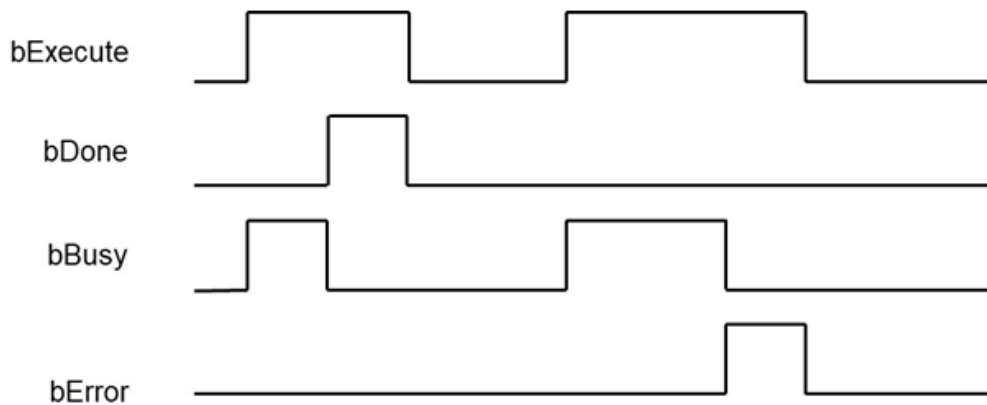
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
bAborted	无作用	BOOL	--
ModbusError	错误码	DFB_MB_ERROR_CODE	DL_MB_ERROR_CODE (DFB_UNDEFINED)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ModbusError		

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_ModbusTCPRequest) 发送标准 Modbus 通讯报文，在使用本指令之前，必须将 Delta_Modbus_TCP_Master 与 Delta_Modbus_TCP_Slave 设定完毕。(详细步骤请参考 AX-3 系列操作手册第 9.3 节「以太网 (Ethernet) 通讯」)

● 范例程序

此范例利用 FB 指令 (DFB_ModbusTCPRequest) 发送标准 Modbus 命令 (0x17) 读取从站 (Delta_Modbus_TCP_Slave) 内存起始地址 0x0000，长度 100 words 的数据，并写入内存起始地址 0x0100，长度 100 words 的数据。

```
ModbusTCPRequest x
2  VAR
3    FB0: DFB_ModbusTCPRequest;
4    bVar0: BOOL:=TRUE;
5    bEcecute_Var,bAbort_Var,bBusy_Var,bDone_Var,bError_Var,bAborted_Var: BOOL;
6    usiUnitID_Var: USINT;
7    ModbusCommand_Var: DL_ModbusTCPMaster_AX3.ModbusCommand;
8    ar_byVar0,ar_byVar1: ARRAY[0..2000] OF BYTE;
9    ModbusError_Var: DL_ModbusTCPMaster_AX3.DFB_MB_ERROR_CODE;
10  END_VAR
11
12  IF bVar0 THEN
13    bVar0:=FALSE;
14    ModbusCommand_Var.FunctionCode:=DL_ModbusTCPMaster_AX3.DFB_MODBUS_FUNC.ReadWrite_Multiple_Registers;
15    ModbusCommand_Var.uiReadLen:=100;
16    ModbusCommand_Var.uiReadOffset:=16#0000;
17    ModbusCommand_Var.uiWriteLen:=100;
18    ModbusCommand_Var.uiWriteOffset:=16#0100;
19    usiUnitID_Var:=12;
20    bEcecute_Var:=TRUE;
21  END_IF
22  FB0(
23    slave:=Delta_Modbus_TCP_Slave ,
24    bExecute:=bEcecute_Var ,
25    bAbort:=bAbort_Var ,
26    usiUnitID:=usiUnitID_Var ,
27    ModbusCommand:= ModbusCommand_Var,
28    pSendData:=ADR(ar_byVar0) ,
29    pReadData:=ADR(ar_byVar1) ,
30    bBusy=>bBusy_Var ,
31    bDone=>bDone_Var ,
32    bError=>bError_Var ,
33    bAborted=>bAborted_Var ,
34    ModbusError=>ModbusError_Var );
```

- 支持機種
 - AX-308E
- 函式庫
 - DL_ModbusTCPMaster_AX3.library

8.6 错误码与故障排除

● DFB_SOCKET_ERROR

说明	错误原因	处置方式
DFB SOCK_ERR_NO_ERROR	无错误	--
DFB SOCK_ERR_INITIALIZE_FAILED	Socket 联机建立失败	<ul style="list-style-type: none"> ● 请检查伺服器端是否存在 ● 请检查伺服器端设定是否正确
DFB SOCK_ERR_CONNREFUSED	建立 Sokcet 联机被拒绝	请检查伺服器端设定是否正确
DFB SOCK_ERR_TIMEDOUT	Server 端发生回应逾时	<ul style="list-style-type: none"> ● 请检查网络线是否正常连接 ● 请检查伺服器端设定是否正确
DFB SOCK_BUILD_CONN_TIMEDOUT	联机建立逾时	<ul style="list-style-type: none"> ● 请检查网络线是否正常连接 ● 请检查伺服器端设定是否正确
DFB SOCK_ERR_NOTCONNECTED	Socket 联机尚未建立	<ul style="list-style-type: none"> ● 请等待 Socket 联机建立完成 ● 请确认伺服器端设定是否正确
DFB SOCK_ERR_INVALID_SETTING	FB 指令设定值不合法	请检查设定值 tcpClientSocketInfo、tcpServerSocketInfo、udpSocketInfo 设定参数是否正确
DFB SOCK_INVALID_BUFFER	输入内存地址不合法	请确认 pSenbuf 与 pRecvbuf 所给定的内存地址是否正确
DFB SOCK_INVALID_LENGTH	报文长度设定值不合法	请确认输入参数 uiRecvLen 设定值是否正确

● DFB_MB_ERROR_CODE

说明	错误原因	处置方式
DFB_NO_ERROR	无错误	--
DFB_ILLEGAL_FUNCTION	未支持的功能码	请检查所使用的功能码是否正确
DFB_ILLEGAL_DATA_ADDRESS	欲写入/读取的内存地址不合法	请确认所欲写入/读取的内存地址是否正确
DFB_ILLEGAL_DATA_VALUE	从站回复的报文异常	请确认线路是否正常或是布线方式是恰当
DFB_RESPONSE_TIMEOUT	从站回应逾时	<ul style="list-style-type: none"> ● 请确认发送报文的周期是否小于从站回应的能力 ● 请检查通讯线路是否正确
DFB_RESPONSE_CRC_ERROR	从站回复的报文异常 (检查码错误)	请确认从站响应报文格式是否正确

说明	错误原因	处置方式
DFB_RESPONSE_WRONG_SLAVE	从站回复的报文异常 (站号错误)	请确认从站响应报文格式是否正确
DFB_RESPONSE_WRONG_FUNCTIONCODE	从站回复的报文异常 (功能码错误)	请确认从站响应报文格式是否正确
DFB_REQUEST_FAILED_TO_SEND	报文发送失败	请联络原厂
DFB_RESPONSE_INVALID_PROTOCOL	从站回复的报文异常 (非标准 Modbus 格式)	请确认从站响应报文格式是否正确
DFB_RESPONSE_INVALID_HEAD	从站回复的报文异常 (报文长度错误)	请确认从站响应报文格式是否正确
DFB_INVALID_CHANNEL_INDEX	非法的从站通道编号	请检查 Channel 编号是否正确
DFB_CHANNEL_SETTING_NOT_SUPPORT	从站通道触发条件非"Application"模式	请检查从站通道触发条件是否为 "Application" 模式
DFB_INVALID_SLAVE	Slave 设定错误	请检查 DFB_ModbusTCPRequest 的 Slave 设定是否正确
DFB_INVALID_BUFFER	接收与传送报文的内存地址设定错误	请检查 DFB_ModbusTCPRequest 的 pWriteBuf 与 pReadBuf 设定是否正确
DFB_INVALID_LENGTH	数据长度设定错误	请检查 DFB_ModbusTCPRequest 的 uiReadLen 与 uiWriteLen 设定是否正确
DFB_INVALID_SLAVE_ADDRESS	从站站号设定错误	请检查 DFB_ModbusTCPRequest 的 usiSlaveAddr 设定是否正确
DFB_NO_MDBSETH_CONFIG	Ethernet 组件不存在	请确认是否已经新增 Ethernet 组件于 device 树形图内
DFB_NO_MASTER_CONFIG	Delta_Modbus_TCP_Master device 不存在	请确认是否已经新增 Delta_Modbus_TCP_Master device 于 device 树形图内
DFB_MB_ERROR_CODE_MEMORY_NOT_ENOUGH	系统内存不足	请确认程序大小是否超出可使用上限

说明	错误原因	处置方式
DFB_CONNECTION_TIMEOUT	TCP 联机建立逾时	<ul style="list-style-type: none">● 请检查 Modbus TCP Slave 端设定是否正确● 请检查通讯线路是否正确
DFB_CONNECTION_FAILED	TCP 联机被拒绝	<ul style="list-style-type: none">● 请检查 Modbus TCP Slave 端设定是否正确
DFB_UNDEFINED	未定义或尚未执行	待 FB 指令执行完成

MEMO

第9章 储存卡读写指令

目录

9.1	DFB_MemoryRead	9-2
9.2	DFB_MemoryWrite	9-6
9.3	错误码与故障排除	9-10

9.1 DFB_MemoryRead

DFB_MemoryRead：读取储存卡。

FB/FC	指令	指令图	ST 语法
FB	DFB_MemoryRead		<pre>DFB_MemoryRead(bExecute:= , FileInfo:= , bDone=> , bBusy=> , bError=> , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
FileInfo	读取文件的参数定	DFB_READ_FILE_INFO	

■ DFB_READ_FILE_INFO

名称	功能	数据类型	设定值范围 (默认值)
sFilePath	所欲读取的文件名	STRING	(")
wDataMode	ASCII CODE / BINARY 模式	DFB_DATA_ MODE	DFB_DATA_MODE.ASCII_MODE DFB_DATA_MODE.BINARY_MODE (DFB_DATA_MODE.ASCII_MODE)
wAsciiShow Mode	读取数据的表示方式 (10 进制/16 进制)	DFB_ASCII SHOW_MODE	DFB_ASCII_SHOW_MODE.DECIMAL DFB_ASCII_SHOW_MODE.HEX (DFB_ASCII_SHOW_MODE.DECIMAL)
wAsciiDec DataType	读取的变量数据类型	DFB_DEC_ DATATYPE	DFB_DEC_DATATYPE.BYTE_SIZE DFB_DEC_DATATYPE.WORD_SIZE DFB_DEC_DATATYPE.DWORD_SIZE DFB_DEC_DATATYPE.LWORD_SIZE DFB_DEC_DATATYPE.SINT_SIZE DFB_DEC_DATATYPE.USINT_SIZE DFB_DEC_DATATYPE.INT_SIZE DFB_DEC_DATATYPE.UINT_SIZE DFB_DEC_DATATYPE.DINT_SIZE DFB_DEC_DATATYPE.UDINT_SIZE DFB_DEC_DATATYPE.LINT_SIZE DFB_DEC_DATATYPE.ULINT_SIZE

名称	功能	数据类型	设定值范围 (默认值)
			DFB_DEC_DATATYPE.REAL_SIZE DFB_DEC_DATATYPE.LREAL_SIZE (DFB_DEC_DATATYPE.BYTE_SIZE)
dwRead StartPos	读取控制器储存卡数 据起始地址*	DWORD	(0)
dwElement Length	读取控制器储存卡数 据长度*	DWORD	1 ~ 25,000 (0)
pDestination	控制器储存数据存储 器地址	POINTER TO BYTE	NULL

*注：以 DFB_READ_FILE_INFO.wAsciiDecDataType 所设定的数据类型为单位。

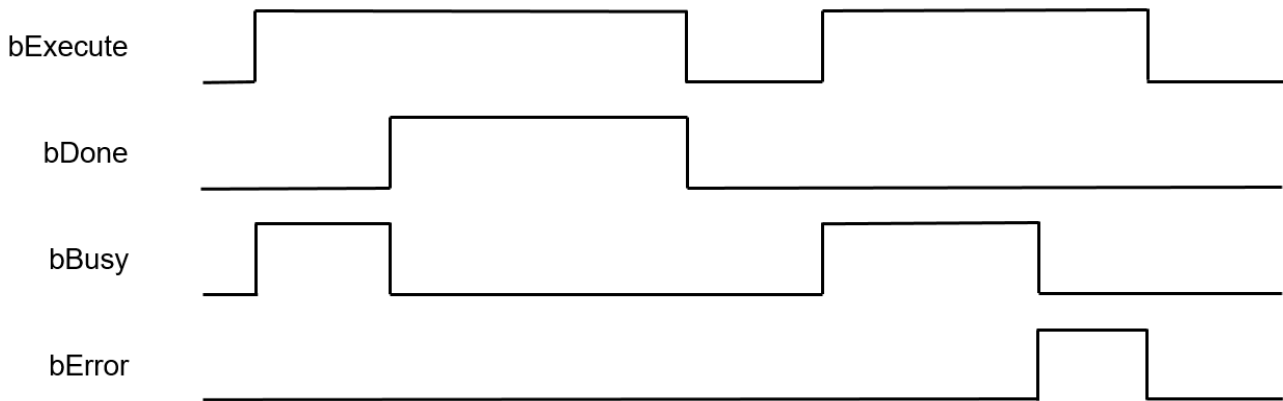
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DL_MEMRW_ERROR	DL_MEMRW_ERROR (DFB_NO_ERR)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ErrorID		

● 参数时序图



● 功能说明

用户可通过 FB 指令 (DFB_MemoryRead) 将储存卡数据读取至控制器内存。

● 范例程序

此范例利用 FB 指令 (DFB_MemoryRead) 储存卡内 Test.csv 文件的内容值读取至控制器 WORD 数组变量 (ar_wVar0)。

```

1  PROGRAM PLC_PRG
2  VAR
3      bVar0: BOOL := TRUE;
4      bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
5      FB0: DFB_MemoryRead;
6      FILE_INFO_Var: DFB_READ_FILE_INFO;
7      ar_wVar0: ARRAY[0..3] OF WORD;
8      ErrorID_Var: DL_MEMRW_ERROR;
9  END_VAR
10
11 IF bVar0 THEN
12     FILE_INFO_Var.sFilePath:='Test.csv';
13     FILE_INFO_Var.wDataMode:=DFB_DATA_MODE.ASCII_MODE;
14     FILE_INFO_Var.wAsciiShowMode:=DFB_ASCISHOW_MODE.HEX;
15     FILE_INFO_Var.wAsciiDecDataType:=DFB_DEC_DATATYPE.WORD_SIZE;
16     FILE_INFO_Var.dwReadStartPos:=0;
17     FILE_INFO_Var.dwElementLength:=4;
18     FILE_INFO_Var.pDestination:=ADR(ar_wVar0);
19     bExecute_Var:=TRUE;
20     bVar0:=FALSE;
21 END_IF;
22 IF bDone_Var THEN
23     bExecute_Var:=FALSE;
24 END_IF
25 FB0(
26     bExecute:=bExecute_Var ,
27     FileInfo:=FILE_INFO_Var ,
28     bDone=>bDone_Var ,
29     bBusy=>bBusy_Var ,
30     bError=>bError_Var ,
31     ErrorID=>ErrorID_Var );

```

储存卡内 Test.csv 文件内如下表。

Test.csv 文件显示的数值					
0	1	2	3	4	5

用户通过 FB 指令 (DFB_MemoryRead) 读取 Test.csv 从第 0 笔数据开始连续四笔数据，并将数据储存到数组变量 (ar_wVar0)，其结果为 ar_wVar0 := [0,1,2,3]。

- 支持機種
 - AX-308E

- 函式庫
 - DL_MemRW_AX3.library

9.2 DFB_MemoryWrite

DFB_MemoryWrite：写入储存卡。

FB/FC	指令	指令图	ST 语法
FB	DFB_MemoryWrite		<pre>DFB_MemoryWrite (bExecute:= , FileInfo:= , bDone=> , bBusy=> , bError=> , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
FileInfo	写入文件的参数设定	DFB_READ_FILE_INFO	

■ DFB_WRITE_FILE_INFO

名称	功能	数据类型	设定值范围 (默认值)
sFilePath	所欲建立的文件名	STRING	(")
wDataMode	ASCII CODE / BINARY 模式	DFB_DATA_MODE	DFB_DATA_MODE.ASCII_MODE DFB_DATA_MODE.BINARY_MODE (DFB_DATA_MODE.ASCII_MODE)
wAsciiShowMode	写入数据的表示方式 (10 进制/16 进制)	DFB_ASCII_SHOW_MODE	DFB_ASCII_SHOW_MODE.DECIMAL DFB_ASCII_SHOW_MODE.HEX (DFB_ASCII_SHOW_MODE.DECIMAL)
wAsciiDecData Type	写入的变量数据类型	DFB_DEC_DATATYPE	DFB_DEC_DATATYPE.BYTE_SIZE DFB_DEC_DATATYPE.WORD_SIZE DFB_DEC_DATATYPE.DWORD_SIZE DFB_DEC_DATATYPE.LWORD_SIZE DFB_DEC_DATATYPE.SINT_SIZE DFB_DEC_DATATYPE.USINT_SIZE DFB_DEC_DATATYPE.INT_SIZE DFB_DEC_DATATYPE.UINT_SIZE DFB_DEC_DATATYPE.DINT_SIZE DFB_DEC_DATATYPE.UDINT_SIZE DFB_DEC_DATATYPE.LINT_SIZE

名称	功能	数据类型	设定值范围 (默认值)
			DFB_DEC_DATATYPE.ULINT_SIZE DFB_DEC_DATATYPE.REAL_SIZE DFB_DEC_DATATYPE.LREAL_SIZE (DFB_DEC_DATATYPE.BYTE_SIZE)
wAccessMode	储存卡文件建立的方式	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 符号*	WORD	(0)
dwWriteStartPos	写入控制器储存卡文件的数据起始地址*	DWORD	(0)
dwElementLength	写入控制器储存卡文件的数据长度*	DWORD	1 ~ 25,000 (0)
pSource	控制器储存数据存储器地址	POINTER TO BYTE	NULL

*注：以 DFB_WRITE_FILE_INFO.wAsciiDecDataType 所设定的数据类型为单位。

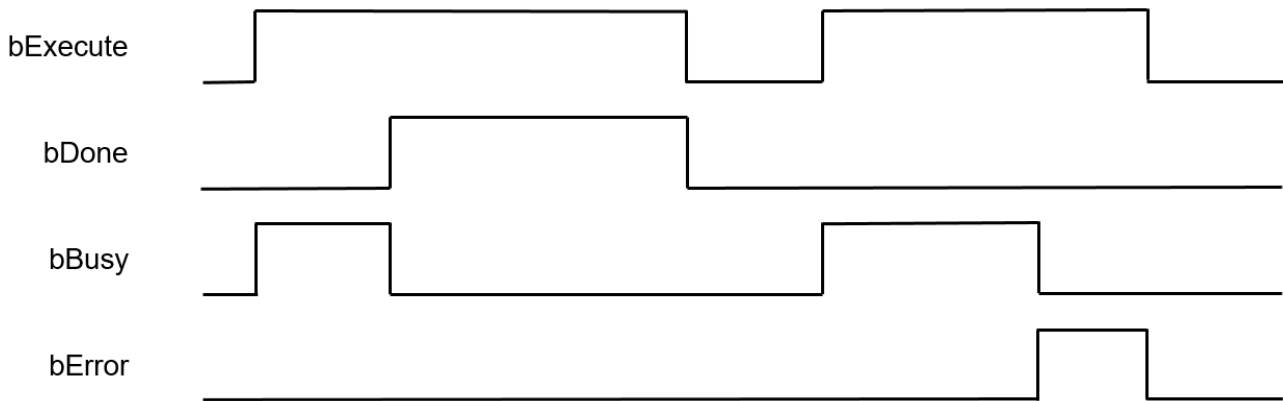
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DL_MEMRW_ERROR	DL_MEMRW_ERROR (DFB_NO_ERR)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 下降沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ErrorID		

● 参数时序图



● 功能说明

用户可通过 FB 指令（DFB_MemoryWrite）将控制器内部数据写入储存卡。

● 范例程序

此范例利用 FB 指令（DFB_MemoryWrite）将 WORD 数组变量写入储存卡。

```

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

10 IF bVar0 THEN
11     FILE_INFO_Var.sFilePath:='Test.csv';
12     FILE_INFO_Var.wDataMode:=DFB_DATA_MODE.ASCII_MODE;
13     FILE_INFO_Var.wAsciiShowMode:=DFB_ASCII_SHOW_MODE.DECIMAL;
14     FILE_INFO_Var.wAsciiDecDataType:=DFB_DEC_DATATYPE.WORD_SIZE;
15     FILE_INFO_Var.wAccessMode:=DFB_ACCESS_MODE.NEW;
16     FILE_INFO_Var.wCarriageReturn:=0;
17     FILE_INFO_Var.dwWriteStartPos:=0;
18     FILE_INFO_Var.dwElementLength:=4;
19     FILE_INFO_Var.pSource:=ADR(ar_wVar0);
20     bExecute_Var:=TRUE;
21     bVar0:=FALSE;
22 END_IF;
23 IF bDone_Var THEN
24     bExecute_Var:=FALSE;
25 END_IF
26 FB0(
27     bExecute:=bExecute_Var ,
28     FileInfo:=FILE_INFO_Var ,
29     bDone=>bDone_Var ,
30     bBusy=>bBusy_Var ,
31     bError=>bError_Var ,
32     ErrorID:=ErrorID_Var0 );
33

```

写入的储存卡的数据为：ar_wVar0: ARRAY [0..3] OF WORD := [0,1,2,10]，而使用计算机开启储存卡.csv 文件后，其显示的内容如下表。

Test.csv 文件显示的数值			
0	1	2	10

*注：

1. `wDataMode :=DFB_DATA_MODE.ASCII_MODE` 则控制器会先将 `ar_wVar0` 数组的内容值以 ASCII CODE 的形式写入储存卡。
 2. `wAsciiDecDataType:=DFB_DEC_DATATYPE.WORD_SIZE`，则 CSV 文件内的数据长度为 WORD。
- 支持机种
 - AX-308E
 - 函式库
 - `DL_MemRW_AX3.library`

9.3 错误码与故障排除

说明	错误原因	处置方式
DFB_NO_ERR	无错误	-
DFB_MEMREAD_ERR_FAILED	内部错误	请联络原厂
DFB_MEMREAD_ERR_PARAMETER	输入参数错误	请确认输入参数是否正确
DFB_MEMREAD_ERR_NOTINITIALIZED	指令无法执行，因为组件初始化尚未完成	请重启控制器
DFB_MEMREAD_ERR_VERSION	版本错误	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_TIMEOUT	运行时间超时	请将控制器恢复出厂设定 (Reset Origin)，如问题持续发生请联络原厂。
DFB_MEMREAD_ERR_NOBUFFER	内存不足	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFB_MEMREAD_ERR_PENDING	程序待执行	请重启控制器
DFB_MEMREAD_ERR_NUMPENDING	待执行程序太多	请将控制器恢复出厂设定 (Reset Origin)，如问题持续发生请联络原厂。
DFB_MEMREAD_ERR_NOTIMPLEMENTED	无此函式	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_INVALIDID	ID 不对	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_OVERFLOW	数值超出范围	请检查输入脚位的数据类型，并将控制器恢复出厂设定 (Reset Origin)，如问题持续发生请联络原厂。
DFB_MEMREAD_ERR_BUFFER_SIZE	内存大小太小	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFB_MEMREAD_ERR_NO_OBJECT	无此物件	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_NO_MEMORY	内存不足	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFB_MEMREAD_ERR_DUPLICATE	对象名称重复	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_MEMORY_OVERWRITE	内存超出可写入范围	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFB_MEMREAD_ERR_INVALID_HANDLE	无效的物件	请确认控制器与函式库版本是否支持

说明	错误原因	处置方式
DFB_MEMREAD_ERR_END_OF_OBJECT	已到达对象最大范围	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_NO_CHANGE	无改变发生	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_INVALID_INTERFACE	无效或未知的 interface	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_NOT_SUPPORTED	不支持此功能	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_NO_ACCESS_RIGHTS	无权限执行此操作指令	请确认控制器与函式库版本是否支持
DFB_MEMREAD_ERR_OUT_OF_LIMITS	已超过有限的资源	请将控制器恢复出厂设定 (Reset Origin) · 并将项目程序缩减后再重新下载 · 如问题持续发生请连络原厂。
DFB_MEMREAD_ERR_ENTRIES_REMAINING	因为有限的 buffer 导致无法传送	请将控制器恢复出厂设定 (Reset Origin) · 并将项目程序缩减后再重新下载 · 如问题持续发生请连络原厂。
DFB_MEMREAD_ERR_INVALID_SESSION_ID	无效的在线操作 ID	请重新登入或是重启控制器
DFB_MEMREAD_ERR_EXCEPTION	发生例外错误	请确认错误纪录

MEMO

第10章 其它指令

目录

10.1	DFC_LogGetSize.....	10-2
10.2	DFB_LogDump.....	10-4
10.3	错误码与故障排除	10-7

10.1 DFC_LogGetSize

DFC_LogGetSize：读取控制器 LOG 记录文件大小。

FB/FC	指令	指令图	ST 语法
FC	DFC_LogGetSize		<pre>DFC_LogGetSize(dwLogNum:= , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
dwLogNum*	欲计算 LOG 记录文件的数量	DWORD	0：计算当前控制器所有 LOG 记录文件数据大小 (0)

*注：dwLogNum 输入参数 可使用 BYTE、WORD、DWORD 变量类型，目前仅支持模式 0。

● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
DFC_LOG_GETSIZE	错误记录文件的数据大小 (此参数为Return形式)	DWORD (单位：BYTE)	0 ~ 65536 (0)
ErrorID	错误码	DL_LOGDMP_ERROR	DL_LOGDMP_ERR (DFC_NO_ERROR)

● 功能说明

当执行此 FC (DFC_LogGetSize) 指令后，即计算当前控制器 LOG 记录文件的数据大小。

- 范例程序

此范例利用 FC 指令 (DFC_LogGetSize) 读取控制器当前 LOG 记录文件数据大小。

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3   dwVar0: DWORD;
4   ErrorID_Var: DL_LOGDMP_ERROR;
5 END_VAR

1 dwVar0:=DFC_LogGetSize(dwLogNum:=0 , ErrorID=>ErrorID_Var );
2
3
```

- 支持机种


- AX-308E

- 函式库

- DL_LogDmp_AX3.library

10.2 DFB_LogDump

DFB_LogDump：读取控制器 LOG 记录文件。

FB/FC	指令	指令图	ST 语法
FB	DFB_LogDump		<pre>DFB_LogDump(bExecute:= , pDmpPos:= , dwLogNum:= , dwDmpLength:= , bDone=> , bBusy=> , bError=> , ErrorID=>);</pre>

● 输入参数

名称	功能	数据类型	设定值范围 (默认值)
bExecute	执行功能块 (上升沿触发)	BOOL	True/False (False)
pDmpPos	控制器储存的内存地址	POINTER TO BYTE	(0)
dwLogNum*	欲读取 LOG 记录文件的数量	DWORD	0：读取当前控制器所有 LOG 记录文件 (0)
dwDmpLength	欲读取的 LOG 记录文件的数据量大小	DWORD	(0)

*注：dwLogNum 输入参数 可使用 BYTE、WORD、DWORD 变量类型，目前仅支持模式 0。

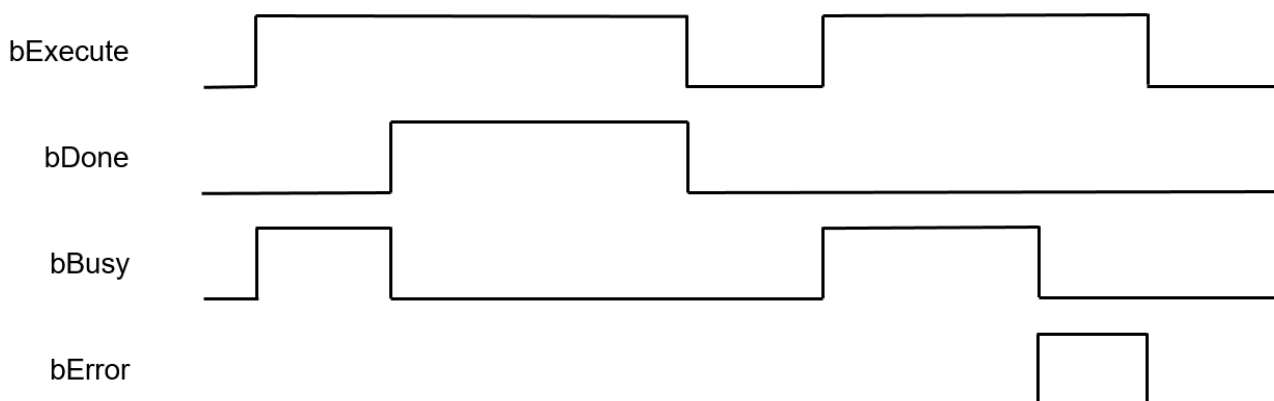
● 输出参数

名称	功能	数据类型	输出值范围 (默认值)
bDone	FB指令执行完成标志	BOOL	True/False (False)
bBusy	FB 指令执行中标志	BOOL	True/False (False)
bError	FB 指令错误标志	BOOL	True/False (False)
ErrorID	错误码	DL_LOGDMP_ERROR	DL_LOGDMP_ERROR (DFB_NO_ERR)

■ 输出更新时间点

名称	输出参数上升沿时机	输出参数下降沿时机
bDone	● FB 指令执行完毕时	● bExecute 上升沿时
bBusy	● FB 指令开始执行	● FB 指令执行完毕时 ● bExecute 下降沿时
bError	● 当 FB 指令执行错误或是输入值错误时	● bExecute 下降沿时
ErrorID		

● 参数 时序图



● 功能说明

用户可通过 FB 指令 (DFB_LogDump) 读取控制器 LOG 记录文件。

● 范例程序

此范例利用 FB 指令 (DFB_LogDump) 读取控制器 LOG 记录文件，并将其记录文件以 ASCII CODE 的方式储存至 Byte 数组变量 (ar_byVar) 内。

```

PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3      bExecute_Var,bDone_Var,bBusy_Var,bError_Var: BOOL;
4      ar_byVar: ARRAY[0..9999] OF BYTE;
5      dwVar0: DWORD;
6      FB0: DFB_LogDump;
7      ErrorID_Var: DL_LOGDMP_ERROR;
8  END_VAR
9
10 IF bDone_Var THEN
11     bExecute_Var:=FALSE;
12 END_IF
13 FB0(
14     bExecute:=bExecute_Var ,
15     pDmpPos:=ADR(ar_byVar) ,
16     dwLogNum:=0 ,
17     dwDmpLength:=65536 ,
18     bDone=>bDone_Var ,
19     bBusy=>bBusy_Var ,
20     bError=>bError_Var ,
21     ErrorID=>ErrorID_Var );

```


- 支持机种
 - AX-308E

- 函式库
 - DL_LogDmp_AX3.library

10.3 错误码与故障排除

说明	错误原因	处置方式
DFB_NO_ERR	无错误	-
DFC_DMP_ERR_FAILED	内部错误	请联络原厂
DFC_DMP_ERR_PARAMETER	输入参数错误	请确认输入参数是否正确
DFC_DMP_ERR_NOTINITIALIZED	指令无法执行，因为组件初始化尚未完成	请重启控制器
DFC_DMP_ERR_VERSION	版本错误	请确认控制器与函式库版本是否支持
DFC_DMP_ERR_TIMEOUT	运行时间逾时	请将控制器恢复出厂设定 (Reset Origin)，如问题持续发生请联络原厂。
DFC_DMP_ERR_NOBUFFER	内存不足	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFC_DMP_ERR_PENDING	程序待执行	请重启控制器
DFC_DMP_ERR_NUMPENDING	待执行程序太多	请将控制器恢复出厂设定 (Reset Origin)，如问题持续发生请联络原厂。
DFC_DMP_ERR_NOTIMPLEMENTED	无此函式	请确认控制器与函式库版本是否支持
DFC_DMP_ERR_INVALIDID	ID 不对	请确认控制器与函式库版本是否支持
DFC_DMP_ERR_OVERFLOW	数值超出范围	请检查输入脚位的数据类型，并将控制器恢复出厂设定 (Reset Origin)，如问题持续发生请联络原厂。
DFC_DMP_ERR_BUFFERSIZE	内存大小太小	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFC_DMP_ERR_NO_OBJECT	无此物件	请确认控制器与函式库版本是否支持
DFC_DMP_ERR_NOMEMORY	内存不足	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFC_DMP_ERR_DUPLICATE	对象名称重复	请确认控制器与函式库版本是否支持
DFC_DMP_ERR_MEMORY_OVERWRITE	内存超出可写入范围	请将控制器恢复出厂设定 (Reset Origin)，并将项目程序缩减后再重新下载，如问题持续发生请联络原厂。
DFC_DMP_ERR_INVALID_HANDLE	无效的物件	请确认控制器与函式库版本是否支持

说明	错误原因	处置方式
DFC_DMP_ERR_END_OF_OBJECT	已到达对象最大范围	请确认控制器与函数库版本是否支持
DFC_DMP_ERR_NO_CHANGE	无改变发生	请确认控制器与函数库版本是否支持
DFC_DMP_ERR_INVALID_INTERFACE	无效或未知的 interface	请确认控制器与函数库版本是否支持
DFC_DMP_ERR_NOT_SUPPORTED	不支持此功能	请确认控制器与函数库版本是否支持
DFC_DMP_ERR_NO_ACCESS_RIGHTS	无权限执行此操作指令	请确认控制器与函数库版本是否支持
DFC_DMP_ERR_OUT_OF_LIMITS	已超过有限的资源	请将控制器恢复出厂设定 (Reset Origin) · 并将项目程序缩减后再重新下载 · 如问题持续发生请联络原厂。
DFC_DMP_ERR_ENTRIES_REMAINING	因为有限的 buffer 导致无法传送	请将控制器恢复出厂设定 (Reset Origin) · 并将项目程序缩减后再重新下载 · 如问题持续发生请联络原厂。
DFC_DMP_ERR_INVALID_SESSION_ID	无效的在线操作 ID	请重新登入或是重启控制器
DFC_DMP_ERR_EXCEPTION	发生例外错误	请确认错误记录