

安装手册

AZ系列 多轴驱动器 DC电源 EtherCAT对应

EtherCAT主站：倍福（BECKHOFF）

EPC(Embedded PC)CX2030、EK1110、CX2100

TwinCAT3控制篇

Profile位置模式（PP）、原点返回模式（HM）不进行运动控制而通过驱动器的内部profile运行的模式。

本手册介绍进行PP/HM运行时的设定以及简单操作的流程。

各个项目的Index，如果没有明确标识即表示Axis1。

本资料由2部构成。

1. BECKHOFF TwinCAT 3 的通信设定

2. Profile 位置模式 (PP) /原点返回模式 (HM) 的运行方法

初学者请从 1 开始,对EtherCAT产品有使用经验者请从2开始阅览。

2 以样本程序为基础进行说明。根据您购买的产品, 在一些设定上需要做相应的更改。请参考1的内容进行PLC的设定。

1. BECKHOFF TwinCAT 3 的通信设定	11
1.1 PC与EPC的通信设定	
Target 设定PC、EPC之间的通信,实施连接测试以确认能否正确通信	13
Target 在TWINCAT 3 上确认PC与EPC之间的通信	17
1.2 网络构成设定	
Target 自动检测连接到EtherCAT上的设备并确认通信状态	23
2. PP/HM模式的运行方法	27
2.1 运行前的各种设定	
Target 进行PP/HM运行前的各种设定	28
2.2 PP/HM模式的样本程序	40
2.3 电动机励磁	
Target 切换驱动器状态机,励磁电动机	44

2. PP/HM模式的运行方法

2.4 通过PP模式进行定位运行

Target PP模式运行电动机 ······ 46

2.5 通过HM模式进行原点返回运行

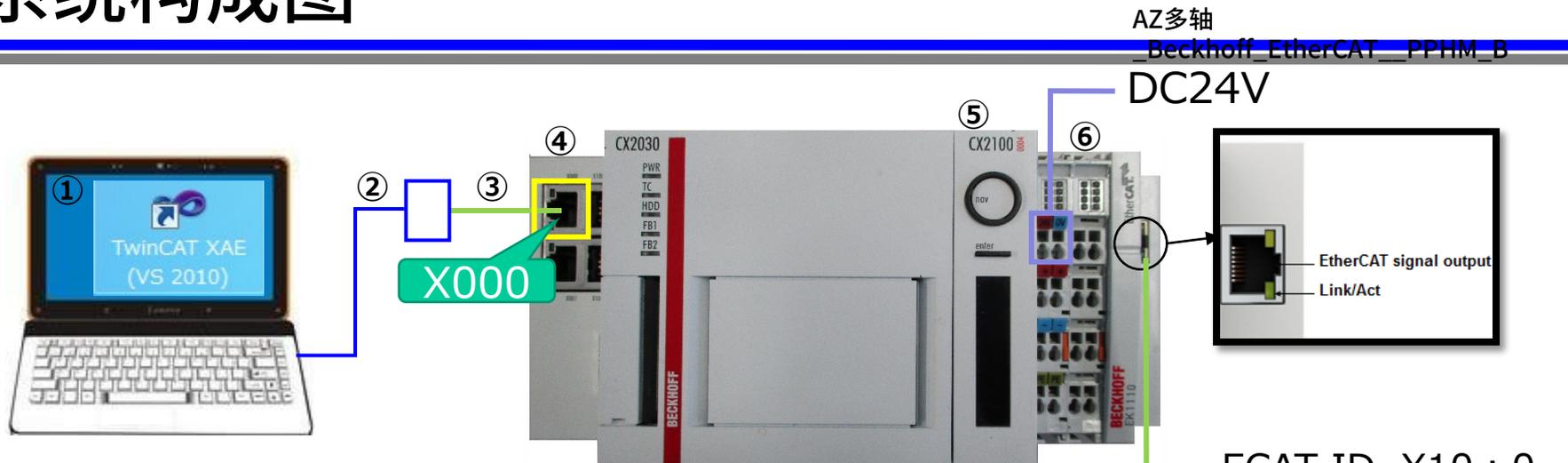
Target HM模式原点返回运行 ······ 51

2.6 通过SDO通信进行参数的读写

Target 通过SDO通信进行参数的读写 ······ 59

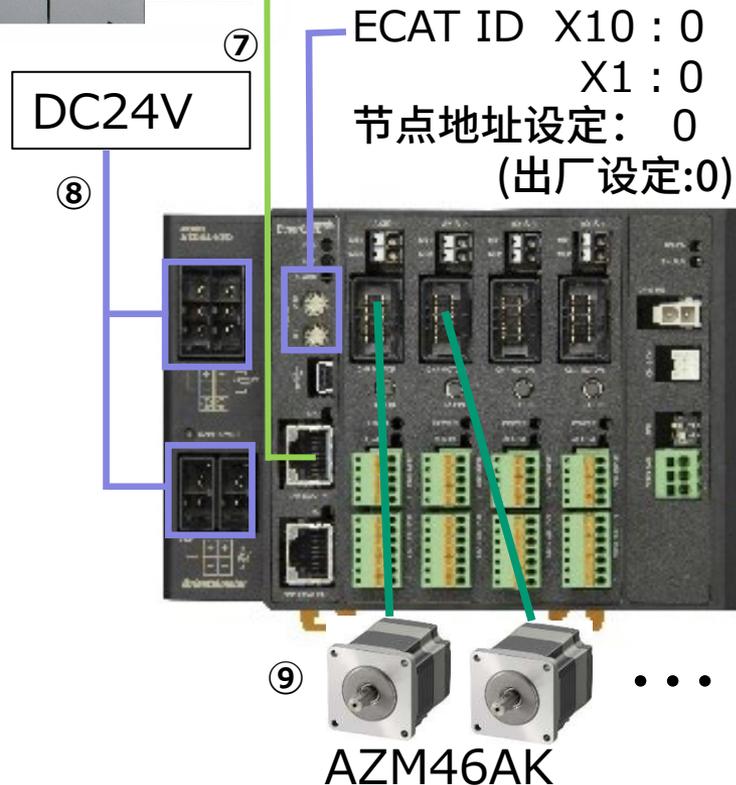
- (1) 在建立实际系统时,请确认好构成系统的各个机器・装置的规格后,采取在额定・性能上留有余地的用法,采取即使万一发生故障也可将危险降至最低的安全电路等安全对策。
- (2) 为安全使用系统,请获取构成系统的各个机器・装置的手册或使用说明书等,确认好「安全上的注意」「安全上的要点」等有关安全的注意事项、内容后使用。
- (3) 请客户自行确认系统应符合的规格・法规或限制。
- (4) 本资料的一部分或全部内容未经东方马达株式会社的许可,禁止复写、复制、再分发。
- (5) 本资料所记载的内容会因改良而有所变更,恕不另行通知。

系统构成图



构成品一览

Item	Name	Type/Model
①	PC Software	TwinCAT3 Ver.3.1.40.18
②	Ethernet Adapter(USB→Ethernet)	-
③	Ethernet Cable	-
④	EPC(Embedded PC)	-
⑤	Power supply units	-
⑥	EtherCAT extension	-
⑦	Ethernet Cable	-
⑧	Multi-axis driver	AZD ■ A-KED
⑨	Motor (AZ)	AZM46AK



参考：Profile位置模式/原点返回模式

AZ多轴

Beckhoff EtherCAT_PPHM_B

Modes of operation的profile位置模式（PP: Profile Position）与原点返回模式（HM: Homing），通过驱动器的内部profile运行。

因此,与EPC连接时,不设定运动轴。

※Profile速度（PV:Profile Velocity）也同样通过内部profile运行。

※由于未设定运动轴,MC_POWER等的功能块(FB)不能使用。

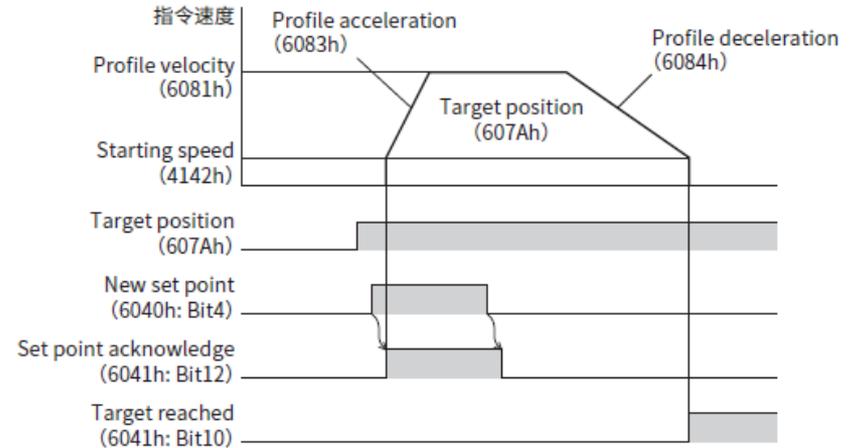
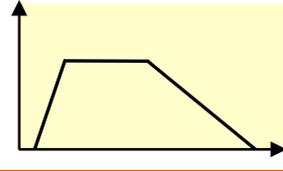
【PP时】

主站



起动指令、Target position、Profile速度、
(EtherCAT通信)

生成profile

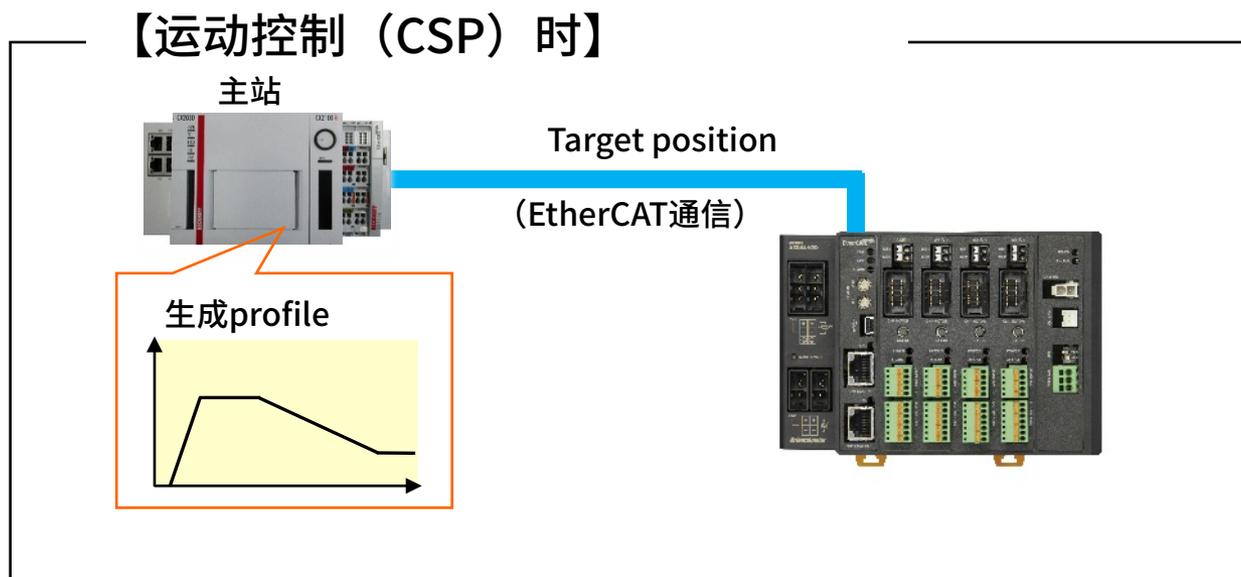


PP时，预先设定Target position（607Ah）、profile速度（6081h），使运行起动指令（controlwordBit4: New set point）由0→1，开始运行。

当通过EtherCAT通信发出运行起动指令时，将在驱动器内部进行位置指令及速度的加减速计算。

若在TwinCAT 3 上进行轴设定，则在主站侧形成位置指令和速度加减速等的profile。

- AZ驱动器的Modes of operation(operation mode)包括Cyclic同步位置模式（CSP）和Cyclic同步速度模式（CSV）。
- 在EPC上设定运动轴时，可使用MC_POWER等功能块（FB）。



CSP通过EtherCAT的Cyclic 通信（PDO通信）在每个通信周期都将 Target position（绝对值）发送到驱动器。

参考：AZ多轴驱动器分辨率

AZ多轴
Beckhoff EtherCAT_PPHM_B

• DC电源 电动机单体



	AZ单轴	AZ多轴
分辨率	0.36°/脉冲(=1,000P/R)	0.036°/脉冲(=10,000P/R)

POINT AZ多轴是以通信进行Motion运行为前提的商品。因此，分辨率的初始设定为10,000P/R。减速机输出轴分辨率的计算方法是用以上分辨率除以减速比。

• 电动滑台 EAS、EZX系列 电动缸 EAC系列



	AZ单轴	AZ多轴
最小移动量	0.01mm (数值与导程无关)	0.3μm、0.6μm、1.2μm (分别为导程3mm、6mm、12mm的数值)

POINT AZ单轴，根据实际组装的机构产品，事先设定好了最小移动量为0.01mm。AZ多轴产品，由于电动机分辨率是统一的10,000P/R，因此需要从导程进行计算。

参考：AZ多轴驱动器分辨率

• 小型电动缸 DRS2



	AZ单轴 	AZ多轴 
最小分辨率	0.001mm (数值与导程无关)	0.2 μ m、0.4 μ m、0.8 μ m (分别为导程2mm、4mm、8mm的数值)

POINT

AZ单轴，根据实际组装的机构产品，事先设定好了最小移动量为0.01mm。
AZ多轴产品，由于电动机分辨率是统一的10,000P/R，因此最小移动量不同。

• 中空旋转式传动装置 DG II



	AZ单轴	AZ多轴
输出工作台 最小移动量	0.01°/STEP	0.002°/STEP

POINT

本产品的机构内部为减速比18的减速器构造。
AZ单轴，事先将最小移动量设定为0.01°/STEP。
AZ多轴产品，由于电动机分辨率是统一的10,000P/R，因此最小移动量不同。

1. BECKHOFF TwinCAT 3 的通信

2. Profil位置模式 (PP) / 原点返回模式 (HM) 的运行方法

PC与EPC的通信设定

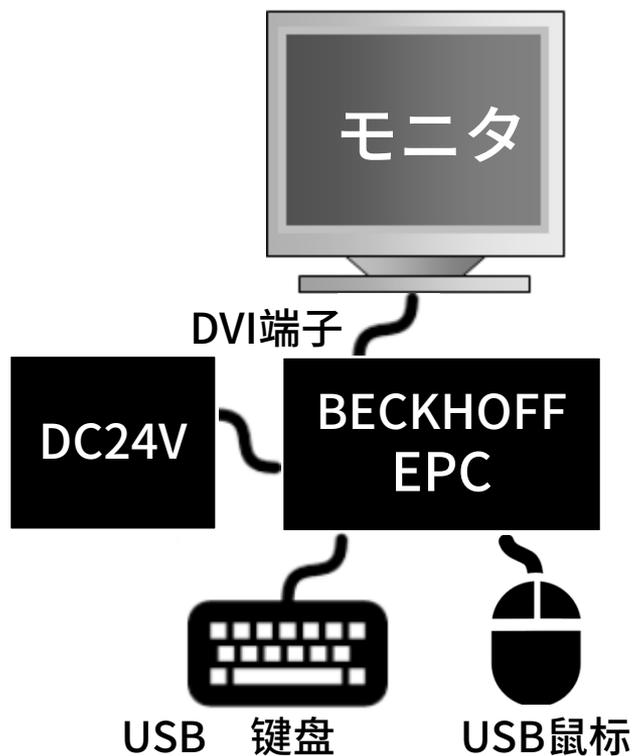
EPC与周边设备的配线示例

AZ多轴

Beckhoff EtherCAT_PPHM_B

- 进行PC和EPC的通信，需要设定各自的IP。
设定EPC的IP,首先**不要连接PC和EPC**,而要先将显示器(DVI)、键盘(USB)、鼠标(USB)等周边设备连接到EPC上。
- EPC与周边设备的连接示意图如下。

- 示意图



- 实际的连接环境 (EPC接通电源时)



- 如下所示,将设备连接到EPC上
- 打开EPC的主电源,启动Windows,然后登录EPC



至鼠标键盘

至显示器(屏幕)
*EPC的显示器输出是数字端子(DVI端子),因此请注意。



Target

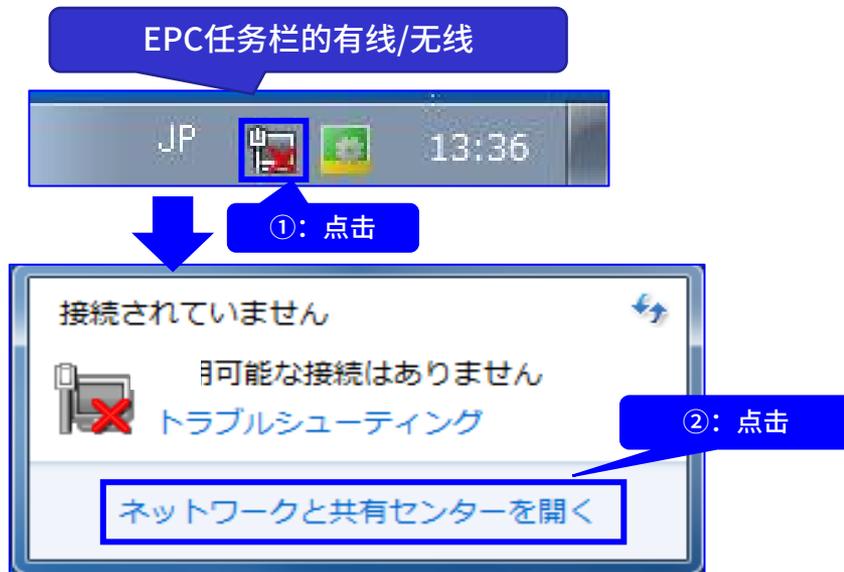
设定PC、EPC之间的通信,实施连接测试以确认能否正确通信。

EPC的Ethernet连接器:设定X000的IP地址和子网掩码。

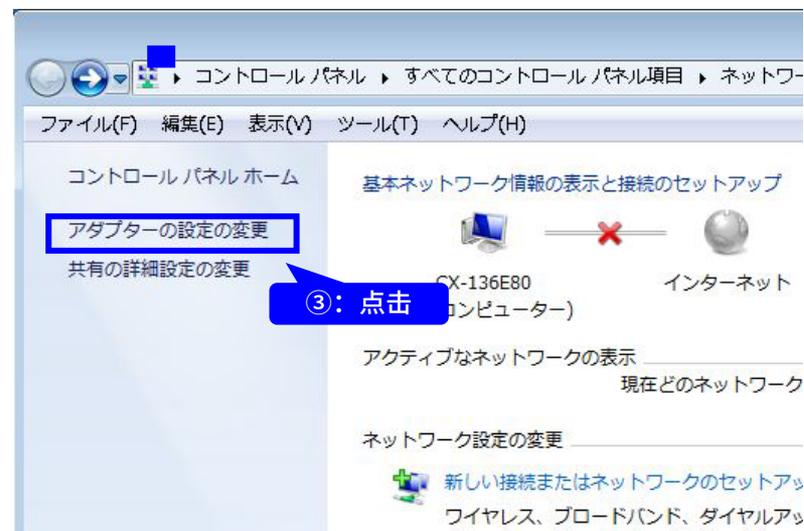
【注意】 设定之前,请与网络管理员确认IP地址。请不要在同一网络上设定重复的IP地址。

■ 设定步骤

1. 网络与共享中心



2. 变更适配器的设定



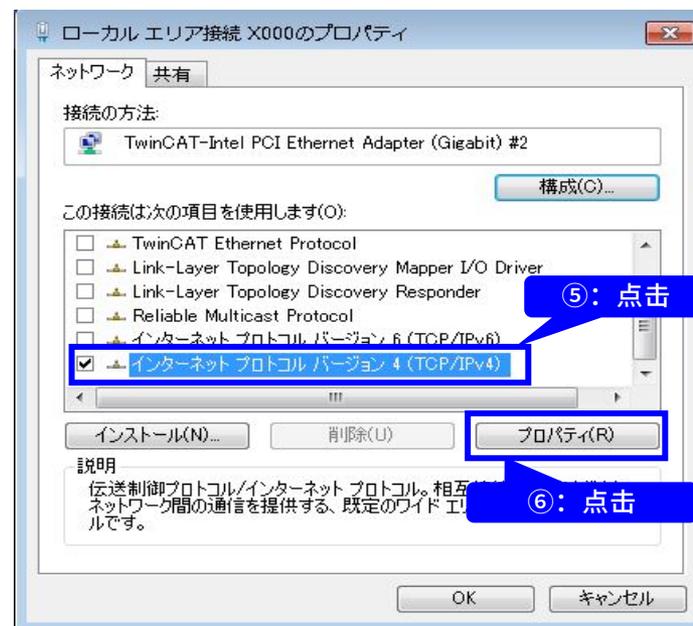
PC与EPC的通信设定

EPC的IP地址设定

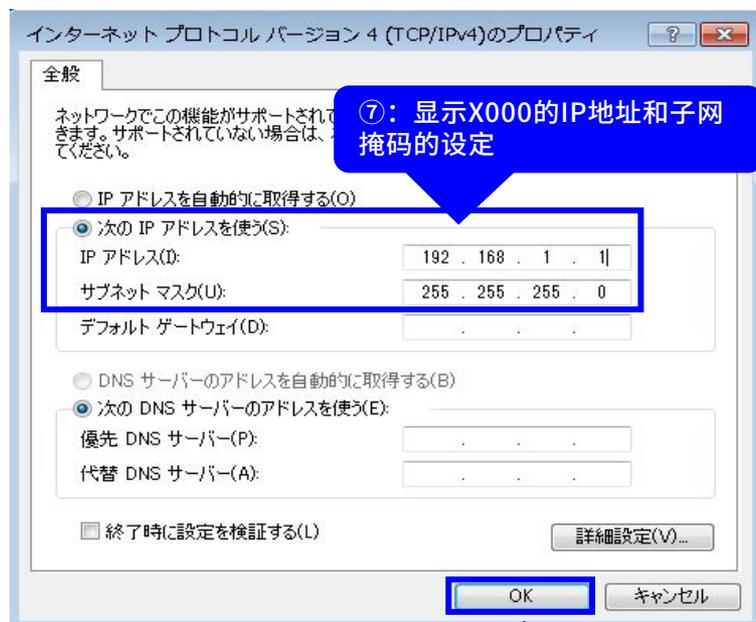
3. 选择本地连接X000,设定IP



4. 选择Internet协议版本4



5. 设定EPC的IP地址和子网掩码



- ⑨: 不仅仅是PC,还要考虑连接到HMI上,下面显示的是X001的IP设定示例。
- IP地址是192.168.1.2
 - 子网掩码: 255.255.255.0

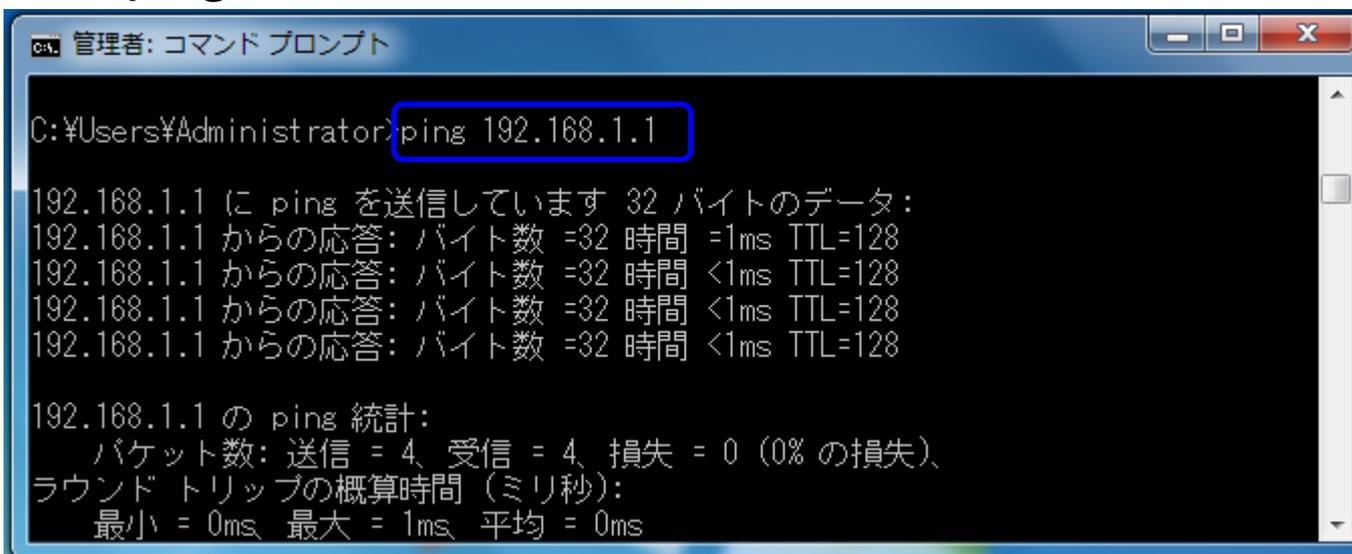
PC与EPC的通信设定

通过PING指令进行通讯测试

从PC向EPC传送数据，测试是否有应答。

IP地址的状态 EPC: 192.168.1.1 (X000ポート)
 PC : 192.168.1.3

从内置Windows的PC上打开命令提示符，
使用ping指令，确认从EPC上有以下的应答。



```
管理: コマンドプロンプト
C:\Users\Administrator>ping 192.168.1.1

192.168.1.1 に ping を送信しています 32 バイトのデータ:
192.168.1.1 からの応答: バイト数 =32 時間 =1ms TTL=128
192.168.1.1 からの応答: バイト数 =32 時間 <1ms TTL=128
192.168.1.1 からの応答: バイト数 =32 時間 <1ms TTL=128
192.168.1.1 からの応答: バイト数 =32 時間 <1ms TTL=128

192.168.1.1 の ping 統計:
    パケット数: 送信 = 4, 受信 = 4, 損失 = 0 (0% の損失),
ラウンド トリップの概算時間 (ミリ秒):
    最小 = 0ms、最大 = 1ms、平均 = 0ms
```

Point 没有应答时，请确认通信电缆线和通信时间的设定是否与问题。

PC与EPC的通信设定

下载ESI文件

Target

在TWINCAT 3 上确认PC与EPC之间的通信。

在PLC上安装ESI文件。

请从以下页面下载多轴驱动器的ESI文件。

https://www.orientalmotor.co.jp/products/stepping/az_driver/features/

※请确认驱动器上的生产日期。2017年2月以后生产的多轴驱动器的版本编号更新为□□□□0200(hex)。(□的数字根据轴数不同儿不同)
如果需要2017年2月以前生产的多轴驱动器的ESI文件，
请向最近的办事处，营业所或客服中心询问。

● 回生抵抗(オプション)

2DCAD	回生抵抗	E1209.dxf
3DCAD	回生抵抗(STEP形式)	E1209.zip

その他

● [EtherCAT ESIファイル](#)

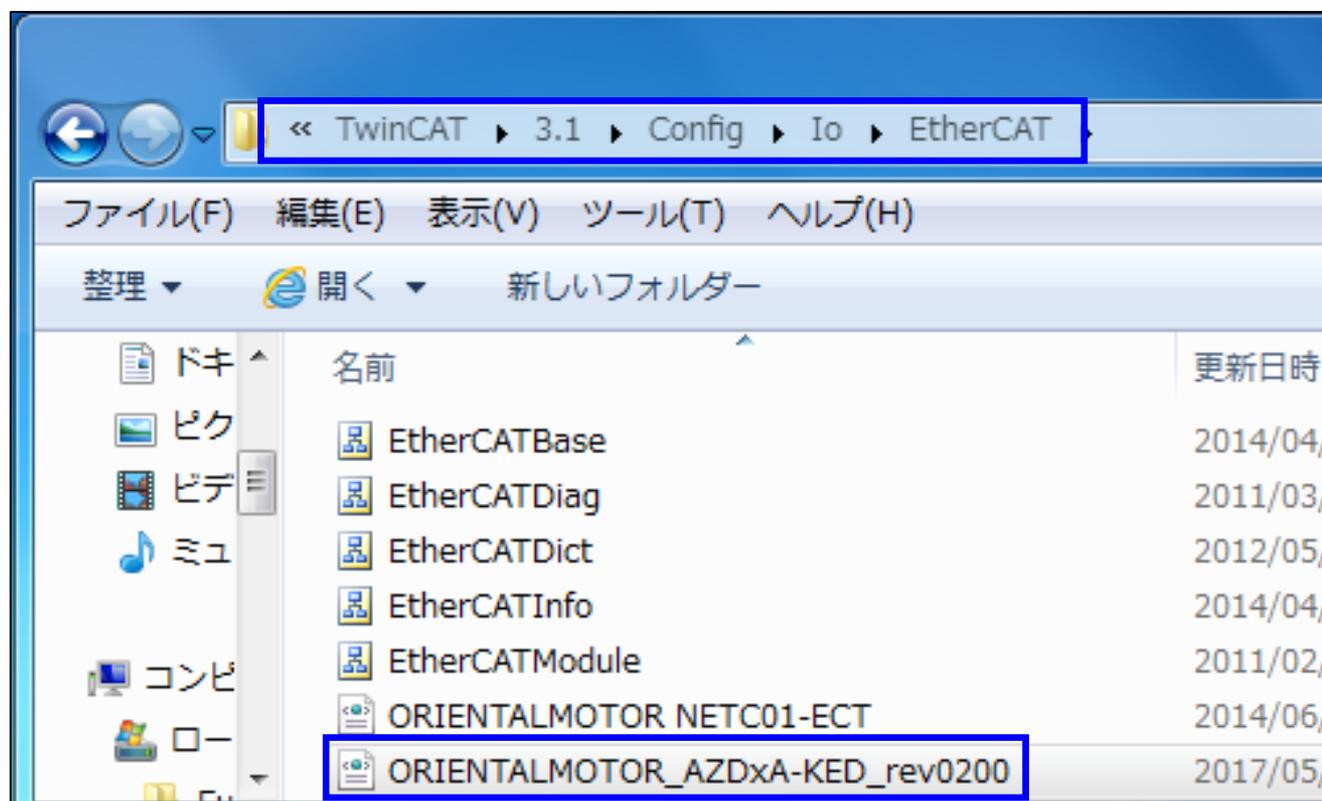
点击

EtherCATスレーブ製品を当社ドライバと接続してお使いいただく場合に必要なESIファイルです。

PC与EPC的通信设定

安装ESI文件

请将下载的ESI文件复制到安装了TwinCAT 3 的文件夹中。
(\\TwinCAT\3.1\Config\Io\EtherCAT)



粘貼

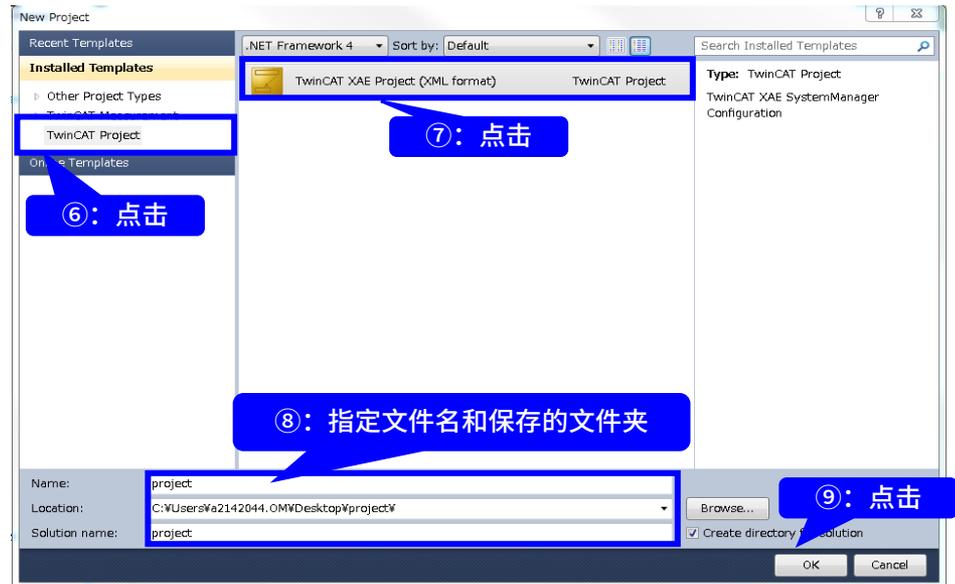
PC与EPC的通信设定

TwinCAT3新建项目

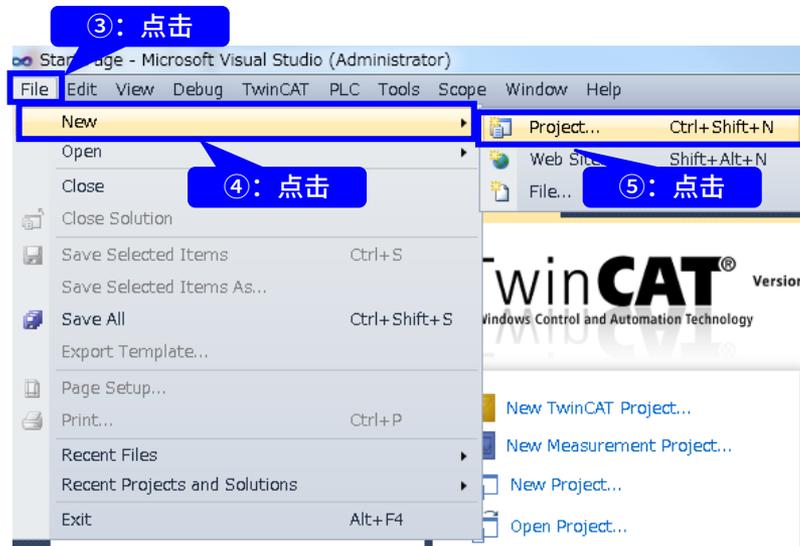
AZ多轴

Beckhoff_EtherCAT_PPHM_B

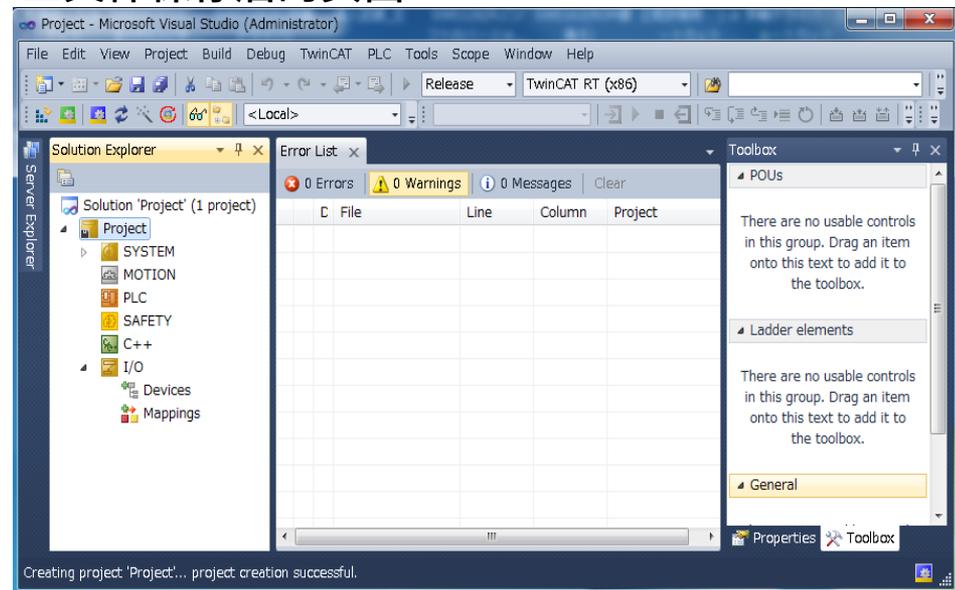
1. 启动TwinCAT XAE(VS2010)



2. 新建项目



文件保存后的页面

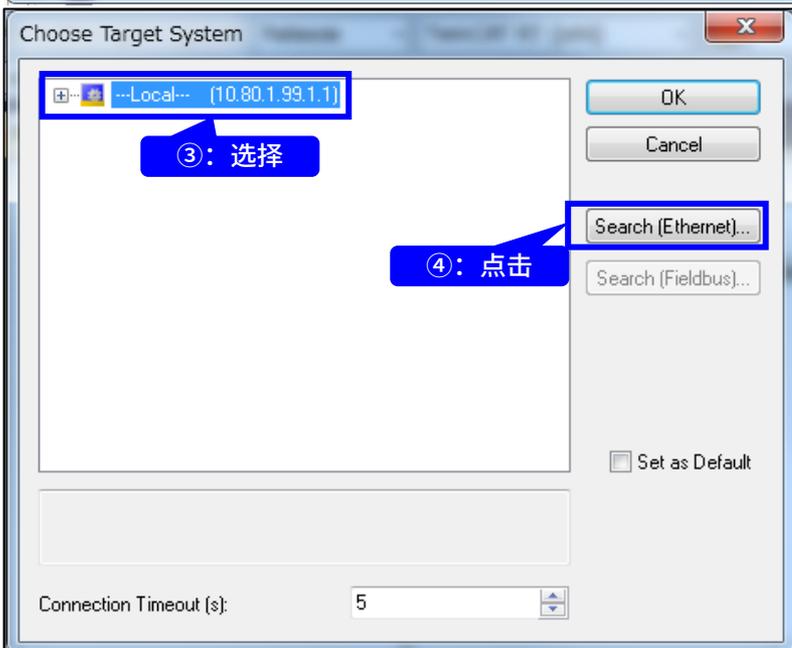
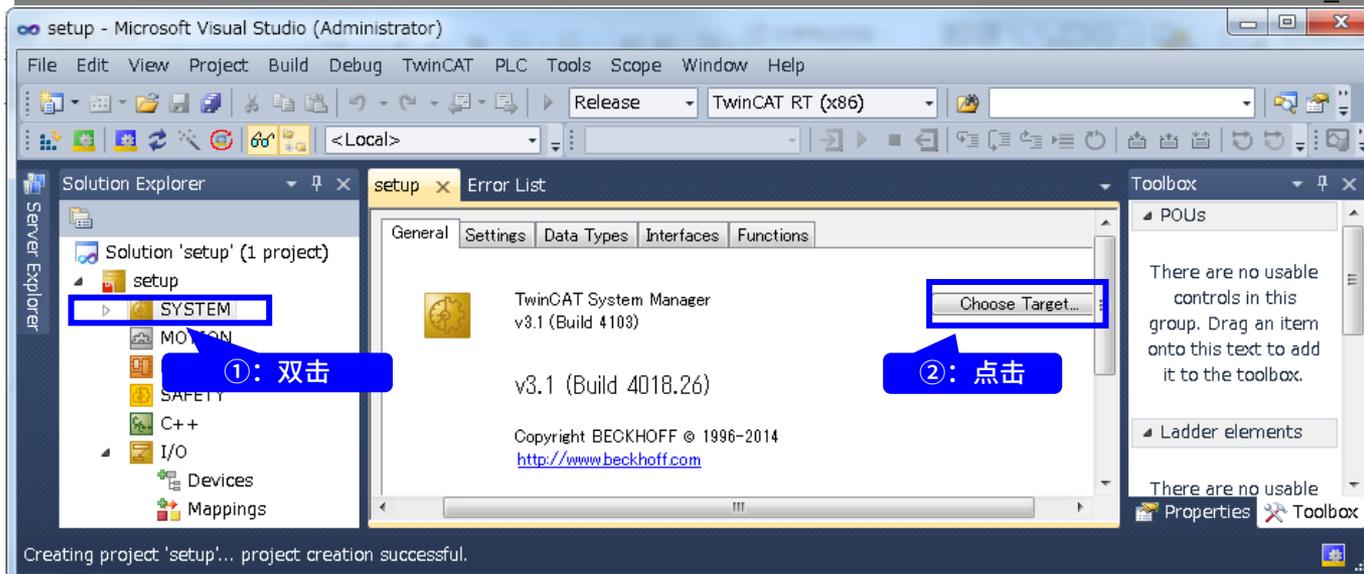


PC与EPC的通信设定

从TwinCAT3检测EPC

AZ多轴

Beckhoff_EtherCAT_PPHM_B



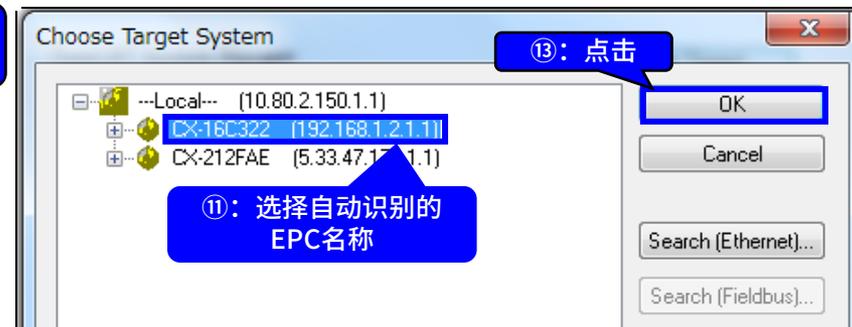
PC与EPC的通信设定

将检测出的EPC登录到Route

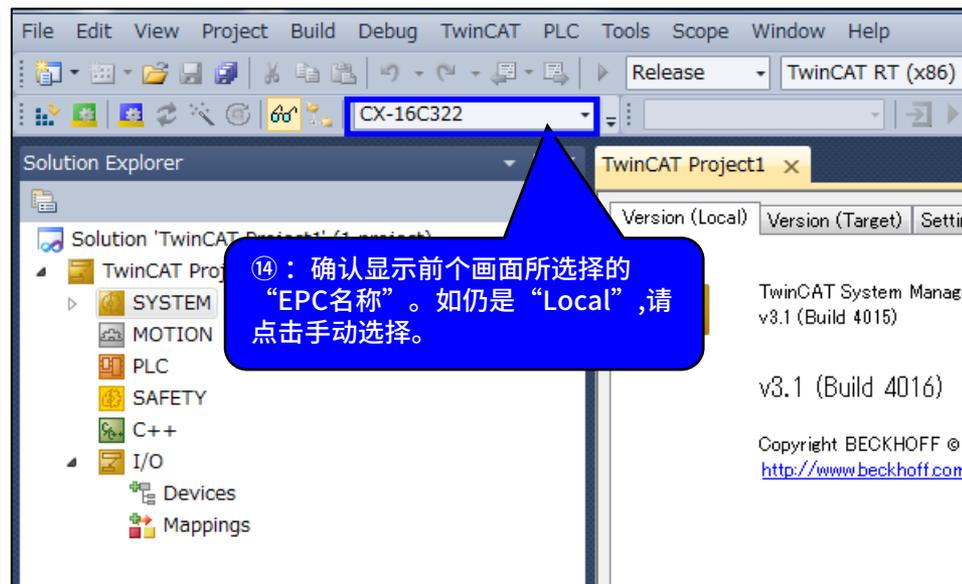
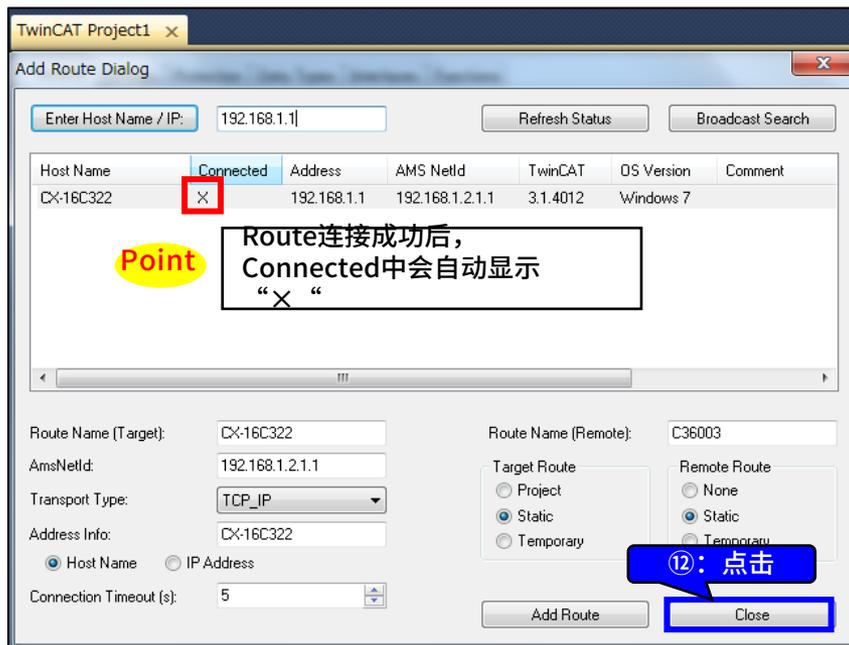
■按Add Route后的画面



■选择Target System



■Target System的確認



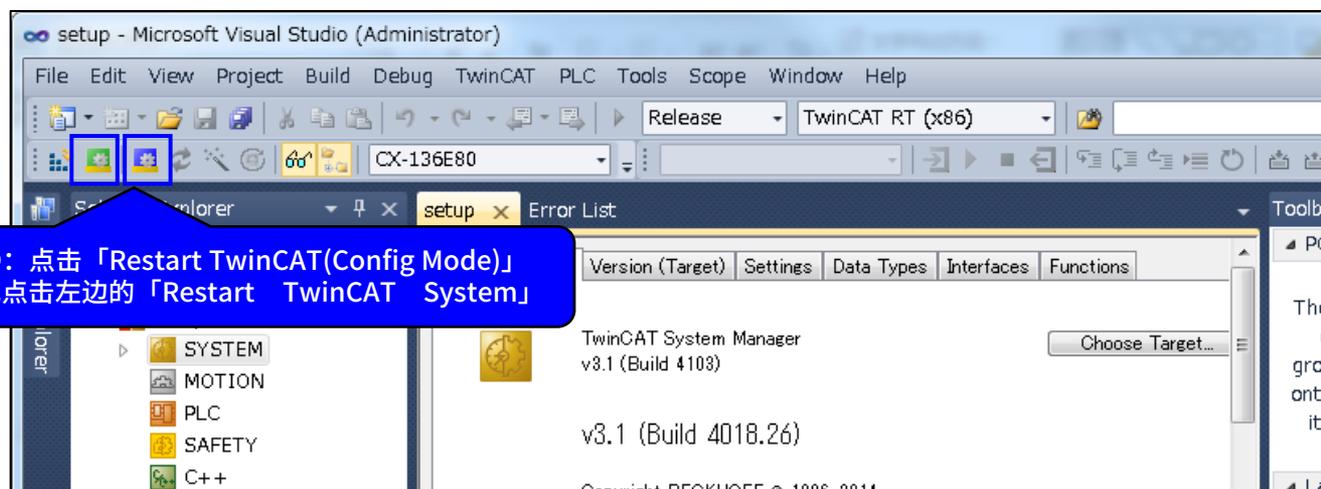
PC与EPC的通信设定

通过EPC的LED确认通信状态

PC与EPC之间的通信可通过EPC的TC的LED来确认

- 确认TC的LED的状态(绿：Run模式/蓝：Config模式)
- 点击「Restart TwinCAT (Config Mode)」 / 「Restart TwinCAT System」，切换Config模式 /Run模式
- 确认切换后的TC的LED的状态

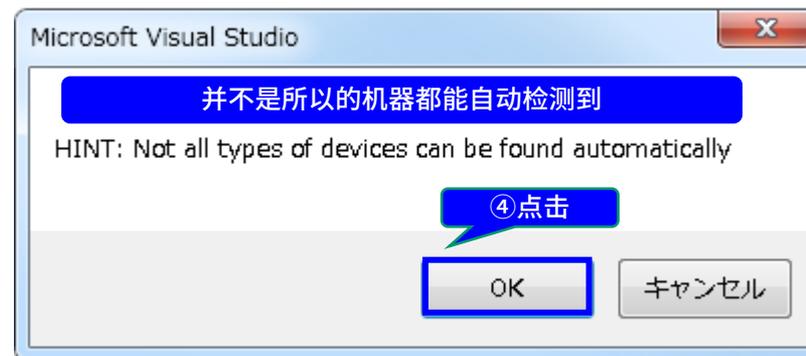
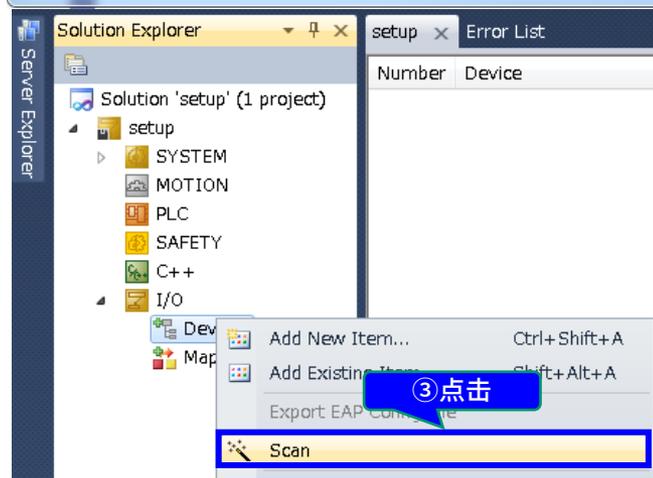
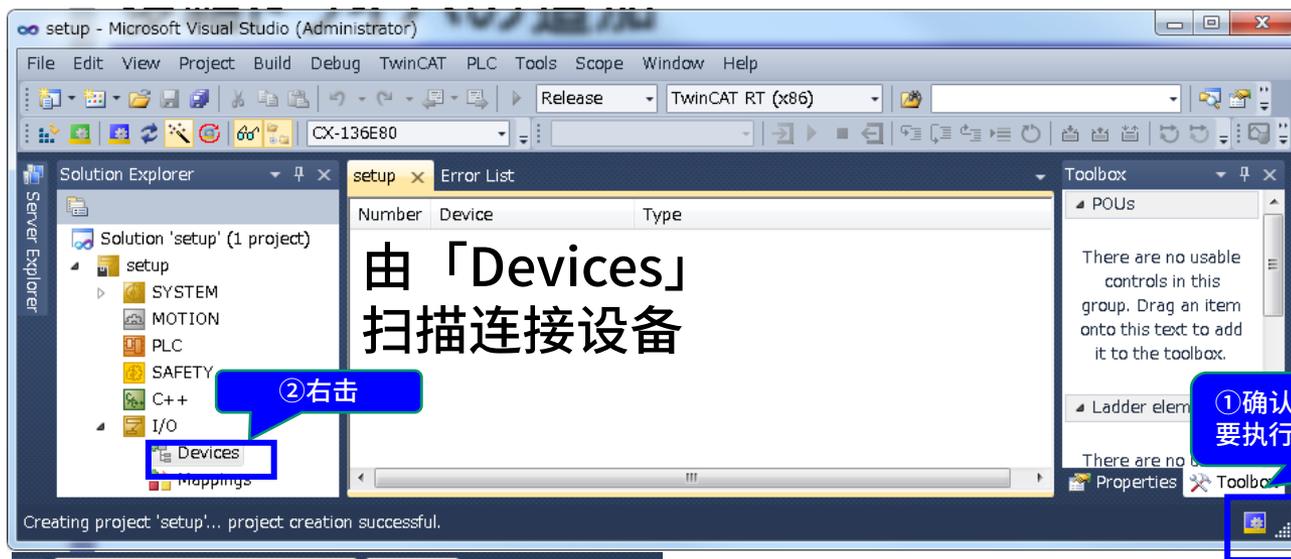
【例：Run模式→Config模式】



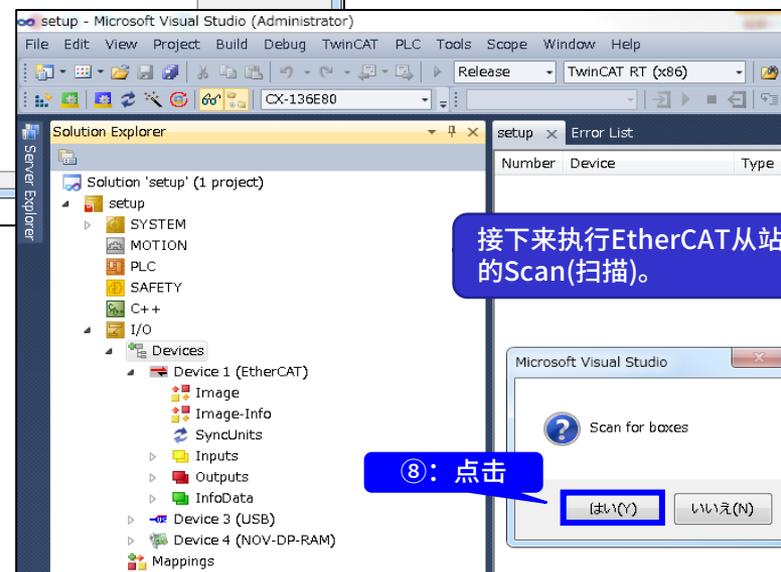
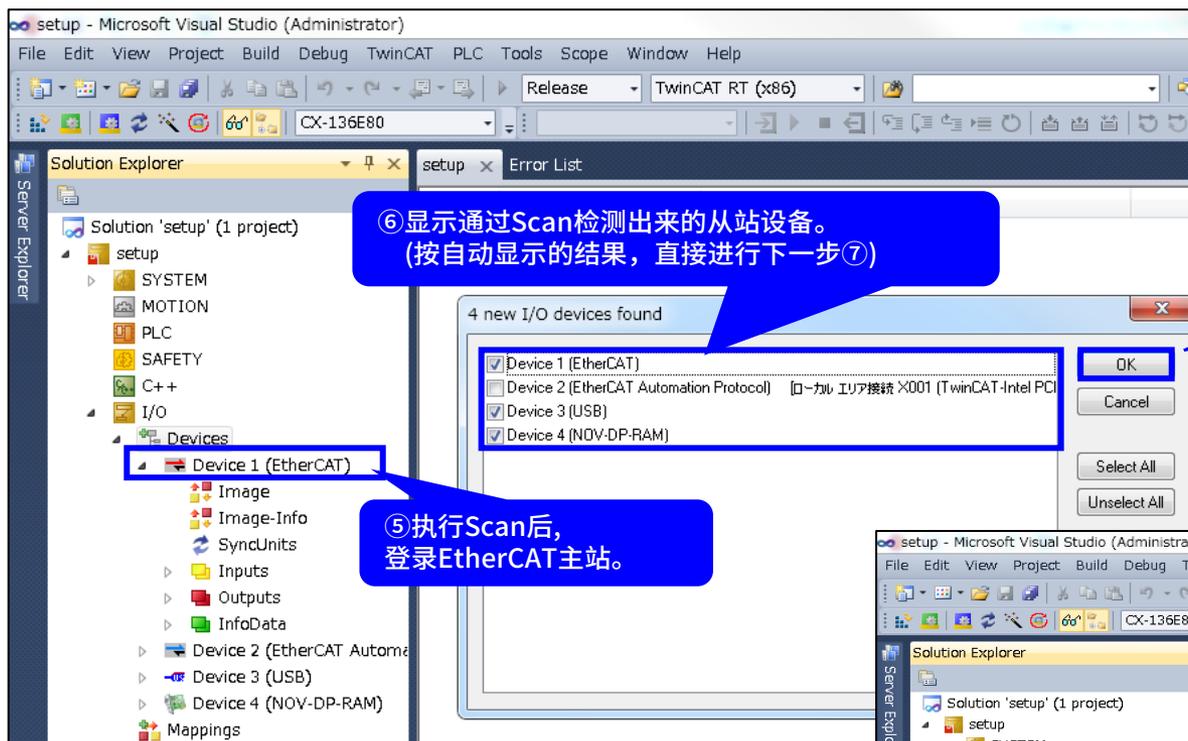
Target

自动检测连接到EtherCAT的设备并确认通信状态。

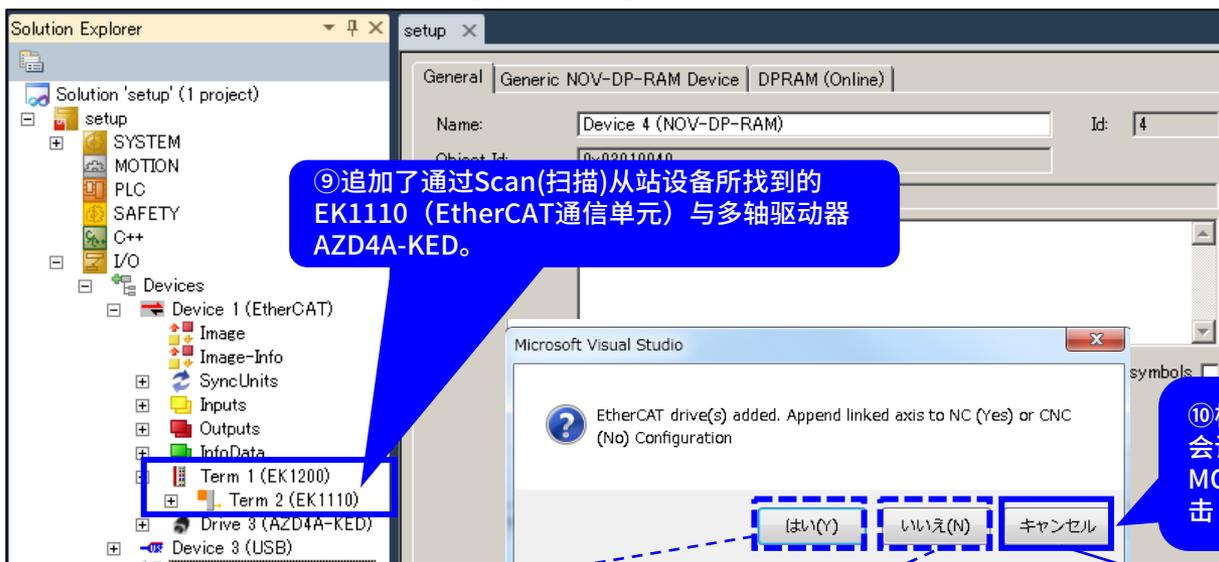
设定EtherCAT的网络构成。在此介绍如何在线检测设备、构成网络的步骤。



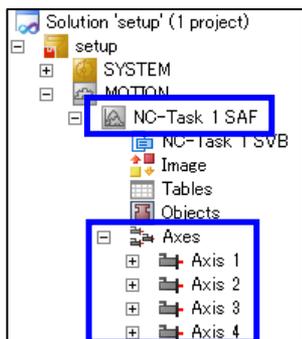
登录EtherCAT主站，并自动扫描从站设备。



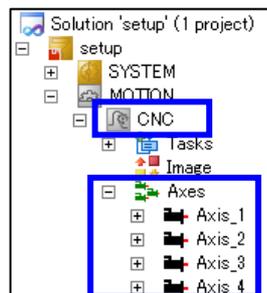
进行EtherCAT从站设备的登录。



选择「Yes」，将轴登录到MOTION的NC。

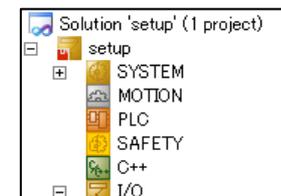


选择「No」，将轴登录到MOTION的CNC。



Motion控制，必须进行轴登录。
⇒请参阅Motion控制篇资料。

选择「Cancel」，不进行MOTION的轴登录。



本资料不进行Motion控制，因此不需要轴登录。

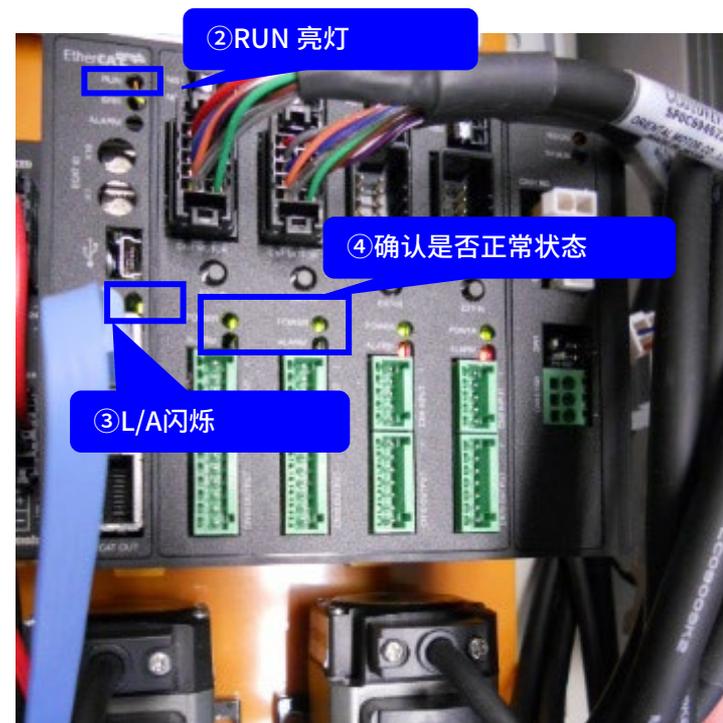
追加设备后,开始Active Free Run。
EPC与AZ驱动器的通信状态可通过EK1110和
驱动器的LED确认。



- EPC



- 多轴驱动器



④POWER LED: 亮灯(正常)
ALARM LED: 消灯(正常)

1. BECKHOFF TwinCAT 3 的通信设定
2. Profil位置模式（PP） /原点返回模式（HM） 的运行方法

运行前各种设定一览

Target

进行PP/HM运行前的各种设定。

制作程序前，执行程序内使用的变量等以下所示各种设定。

- ① 接收PDO映射的对象(object)登录
- ② PLC的新项目登录
- ③ 通信周期Cycle ticks的设定
- ④ 全局变量的定义
- ⑤ PLC输出变量与驱动器输入的连接设定

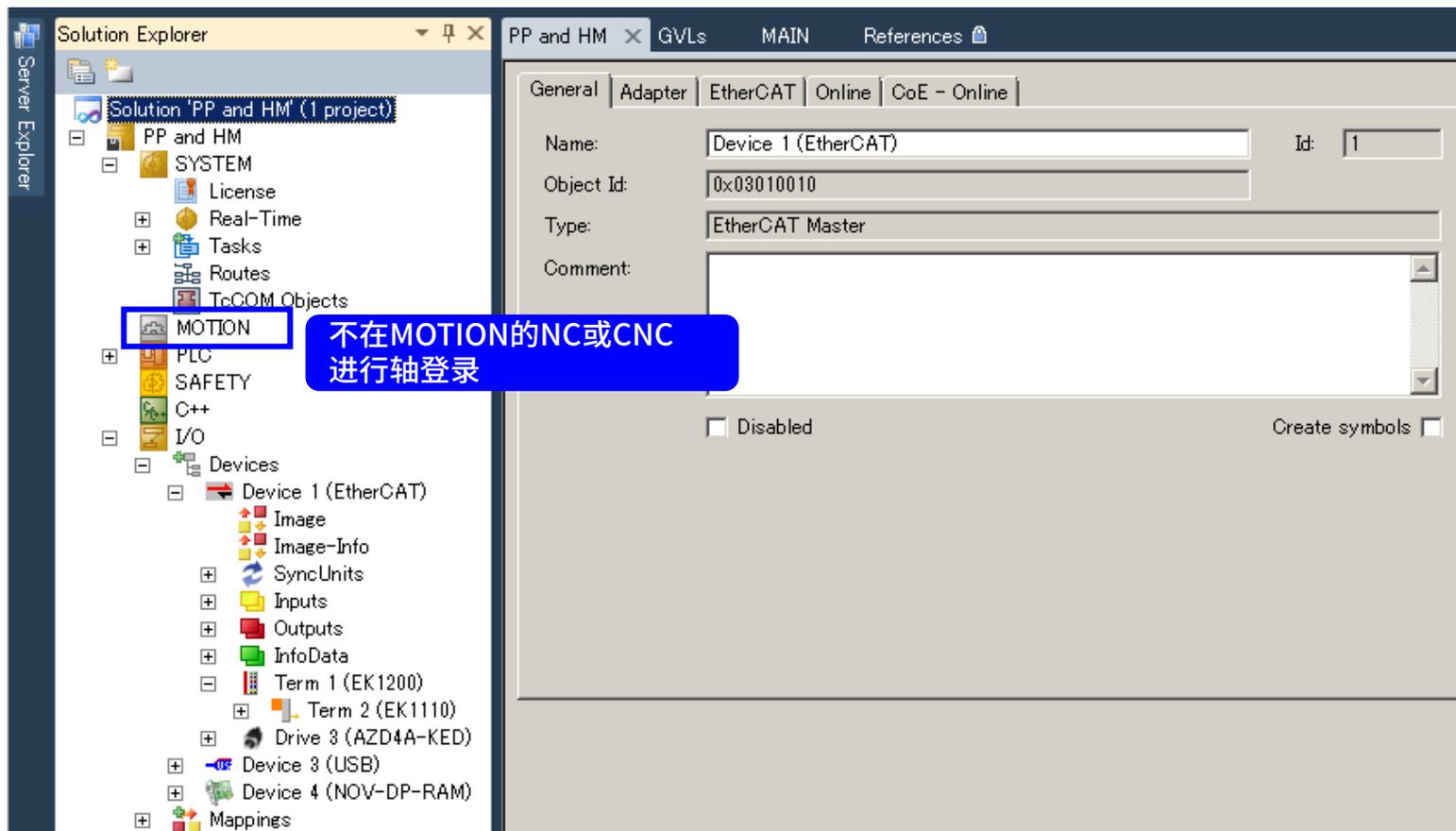
①接收PDO映射的对象(object)登录

AZ多轴

Beckhoff EtherCAT_PPHM_B

按照步骤，进行网络构成设定。

使用PP/HM模式，不需要进行MOTION的轴登录。



①接收PDO映射的对象(object)登录

AZ多轴

Beckhoff EtherCAT_PPHM_B

PP、HM使用以下对象(object)。

通过PDO映射设定的编辑进行输出对象(object)的追加・删除。

对象	Axis1	Axis2	Axis3	Axis4
Controlword	0x6040	0x6840	0x7040	0x7840
Target position	0x607A	0x687A	0x707A	0x787A
Modes of operation	0x6060	0x6860	0x7060	0x7860
Profile 速度	0x6081	0x6881	0x7081	0x7881

注意：请根据您使用的驱动器的轴数，进行相应的PDO映射对象设定。

各轴的对象 (object) 选择如下。Profile area的Object，每一轴偏置800h。本资料如果没有特别标注，则表示Axis1轴的情况。

轴	Profile area 的 Object	Manufacturer specific area 的 Object
驱动器轴 1	6000h ~ 67FFh	4000h ~ 4FFFh 的 Sub-index 1
驱动器轴 2	6800h ~ 6FFFh	4000h ~ 4FFFh 的 Sub-index 2
驱动器轴 3	7000h ~ 77FFh	4000h ~ 4FFFh 的 Sub-index 3
驱动器轴 4	7800h ~ 7FFFh	4000h ~ 4FFFh 的 Sub-index 4

运行前的各种设定

①接收PDO映射的对象(object)登录

AZ多轴

Beckhoff EtherCAT_PPHM_B

PDO追加Profile 速度 (6081h)。

⑧ 点击

⑦ 保留默认值

⑥ 选择

④ 右击

⑤ 选择

⑨ 添加后的画面

Index	Size	Offs	Name	Type
0x6040:00	2.0	0.0	Controlword	UINT
0x607A:00	4.0	2.0	Target position	DINT
0x6060:00	1.0	6.0	Modes of operation	SINT
0x6081:00	4.0	7.0	Profile velocity	UDINT

Creating project 'PP-Mode'... project creation successful.

运行前的各种设定

①接收PDO映射的对象(object)登录

AZ多轴

Beckhoff EtherCAT_PPHM_B

作为PDO映射的设定例，介绍如何将 Profile 速度(0x6081)登录到Axis1的“接受PDF映射(0x1600)”中。

CoE通信区域的 object	Axis1	Axis2	Axis3	Axis4
PDO映射	0x1600	0x1610	0x1620	0x1630

Profile区域的 object	Axis1	Axis2	Axis3	Axis4
Controlword	0x6040	0x6840	0x7040	0x7840
Target position	0x607A	0x687A	0x707A	0x787A
Modes of operation	0x6060	0x6860	0x7060	0x7860
Profile 速度	0x6081	0x6881	0x7081	0x7881

请用同样的步骤，将需要的对象（object）登录到各个轴的PDO映射中。

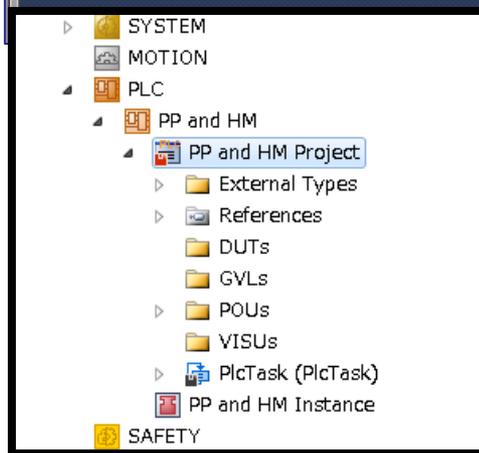
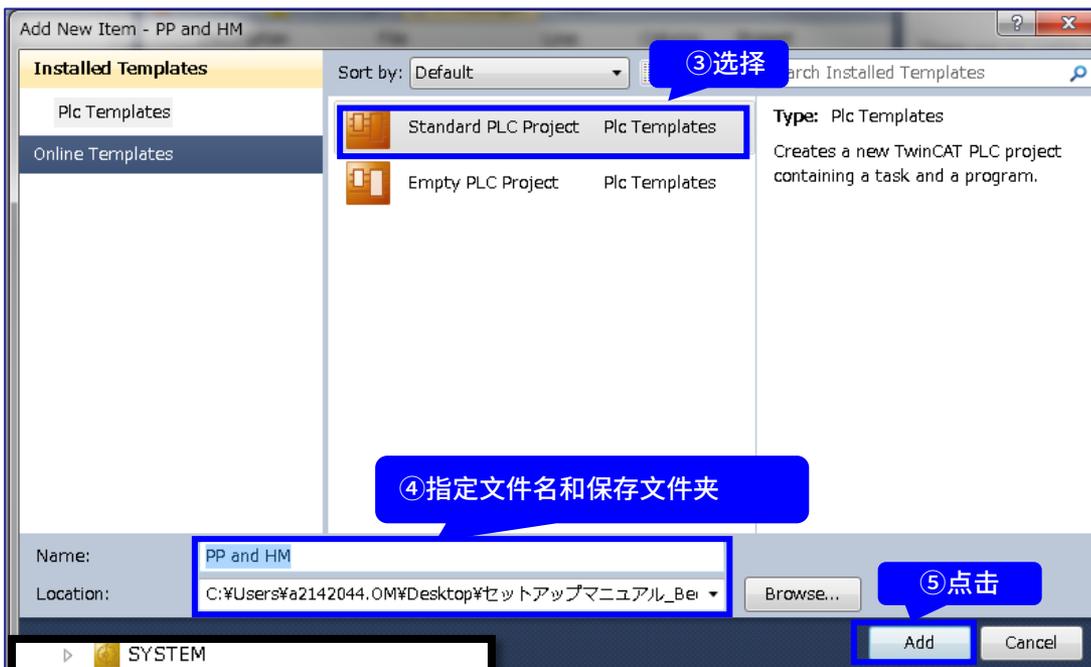
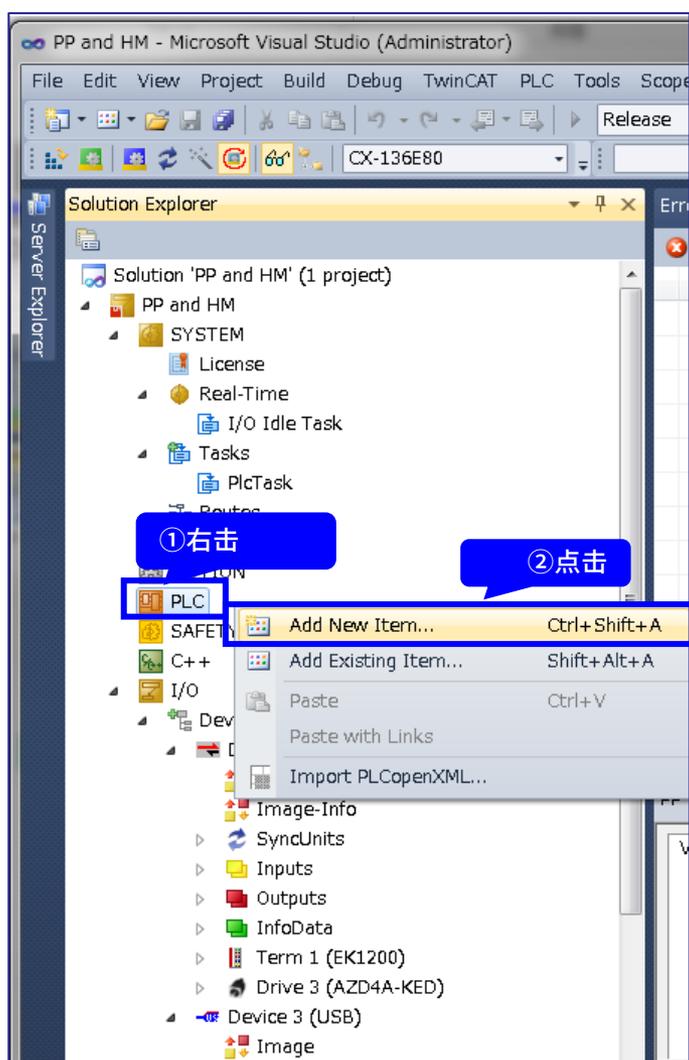
运行前的各种设定

② PLC的新项目登录

AZ多轴

Beckhoff EtherCAT_PPHM_B

追加PLC项目。并在此PLC项目中登录程序文件和程序中使用的变量。



运行前的各种设定

③通信周期Cycle ticks的设定

AZ多轴

Beckhoff EtherCAT_PPHM_B

EtherCAT通信周期（DC模式）为0.5~8.0[ms]。
为此,PP/HM要求Cycle ticks的设定小于8.0[ms]。
在此Cycle ticks设定为1.0[ms]。

The screenshot shows the TwinCAT Project setup manual interface. The Solution Explorer on the left shows the project structure with 'Tasks' expanded and 'PlcTask' selected. Two callouts point to 'Tasks' (① 点击) and 'PlcTask' (② 点击). The main window shows the 'Task' configuration for 'PlcTask'. The 'Cycle ticks' field is set to 10, with a callout (③ 默认值) pointing to the default value of 10.00 ms. A blue arrow points from this field to a second configuration window where the 'Cycle ticks' field is set to 1, with a callout (④ 设定1ms (例)) pointing to the value 1.000 ms.

运行前的各种设定

④全局变量的定义

AZ多轴

Beckhoff_EtherCAT_PPHM_B

定义程序输入，输出所使用的参数名称（变量名称）。

The screenshot illustrates the process of defining a global variable list in Visual Studio. The steps are as follows:

- ① 右击**: Right-click on the 'GV' folder in the Solution Explorer.
- ② 选择**: Select 'Global Variable List...' from the context menu.
- ③ 输入名称**: Enter the name 'GVLS' in the 'Name' field of the 'Add Parameter list' dialog.
- ④ 点击**: Click the 'Open' button to create the global variable list.

运行前的各种设定

④全局变量的定义

AZ多轴

Beckhoff EtherCAT_PPHM_B

② 点击“Build Solution”

① 由于四轴都使用同一个参数，因此使用了“ARRAY(配列)”定义输出输入参数。(使用ARRAY以外的方法也没有问题)

注意：此程序为4轴用。但2轴或3轴也适用。

③ 点击

扩大

```
VAR_GLOBAL
1 bCON AT%I*:ARRAY[0..3] OF BOOL; //True:Current ON
2 bOPMD AT%I*:ARRAY[0..3] OF BOOL; //TRUE:PP ,False:HM
3 bStart AT%I*:ARRAY[0..3] OF BOOL; //Positioning Mode TRUE:Start Abs
4 OD AT%I*:ARRAY[0..3] OF BOOL; //Setting the operation data of position
5 bHome AT%I*:ARRAY[0..3] OF BOOL; //Homing Mode
6 bStop AT%I*:ARRAY[0..3] OF BOOL; //TRUE: motor Stop
7
8 CTWD AT%Q*:ARRAY[0..3] OF UINT:=[6,6,6,6]; //Controlword
9 TPOS AT%Q*:ARRAY[0..3] OF DINT:=[10000,10000,10000,10000]; //Target Position[step]
10 PVEL AT%Q*:ARRAY[0..3] OF DINT:=[10000,10000,10000,10000]; //Profile Velocity[Hz]
11 OPMD AT%Q*:ARRAY[0..3] OF SINT:=[1,1,1,1]; //Operation Mode 1:PP /6:HM
12
13
14 END_VAR
```

④ Instance反映右侧定义的参数。

注意：如果程序有误,将不会在⑤中反映出来。

Point

向I/O组合等的外部设备输入或输出的参数(变量)请带上「I*」或「Q*」。
此外,外部参照变量build后追加到PLC的Instance。
AT%I*: Input用的变量 / AT%Q*: Output用的变量

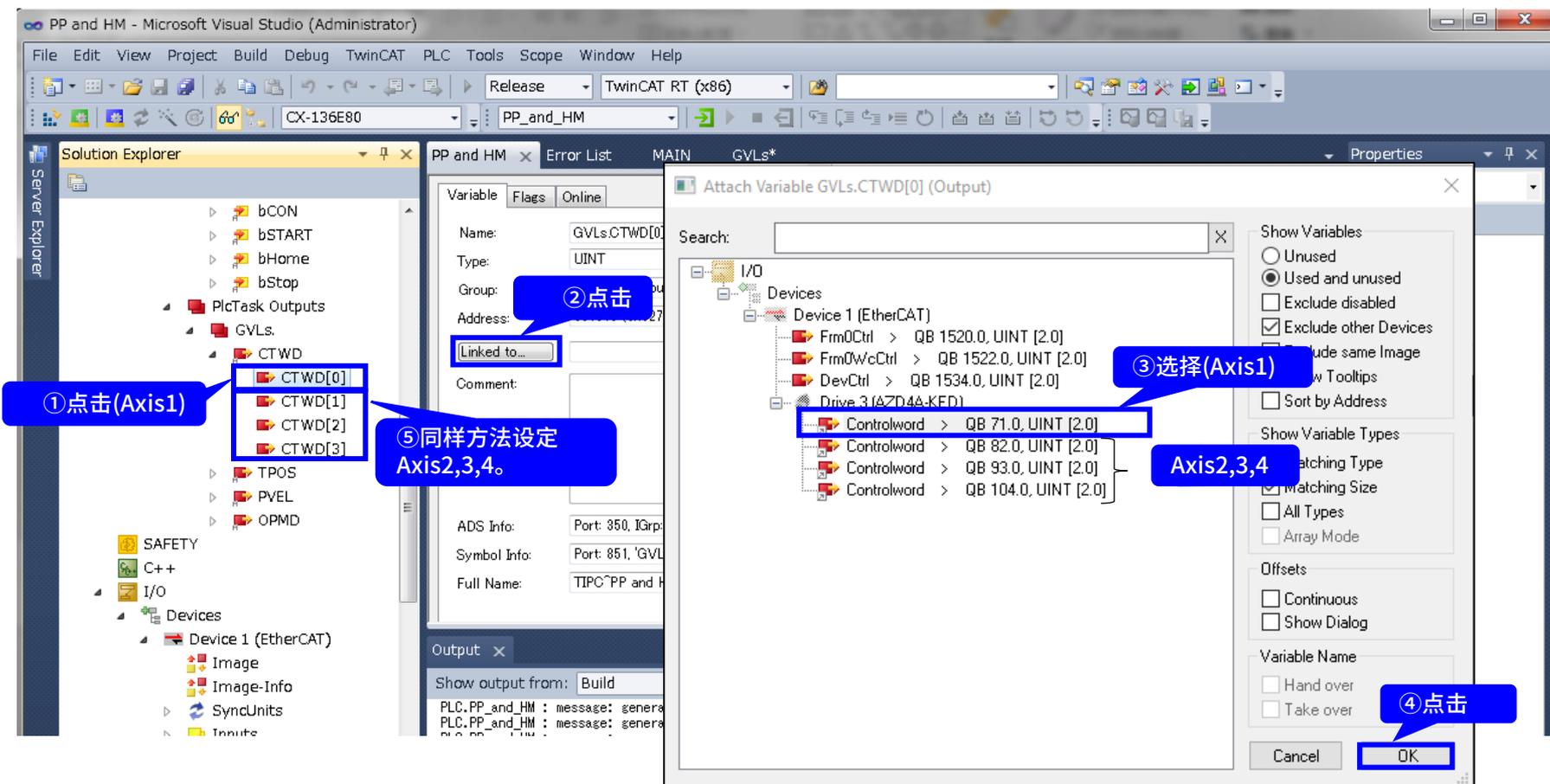
运行前的各种设定

⑤ PLC输出变量与驱动器输入的连接设定

AZ多轴

Beckhoff_EtherCAT_PPHM_B

链接设定定义了输出参数（变量）与controlword。
现在可将PLC输出变量与EtherCAT的对象（object）相关联起来。

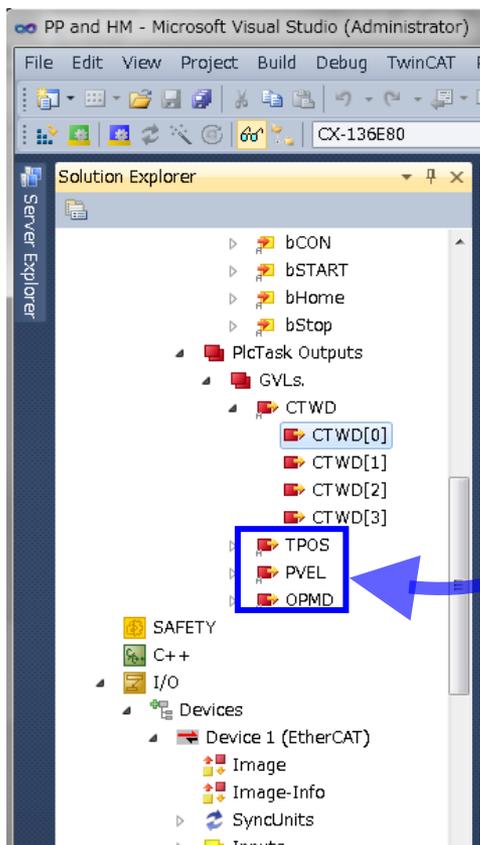


⑤ PLC输出变量与驱动器输入的连接设定

AZ多轴

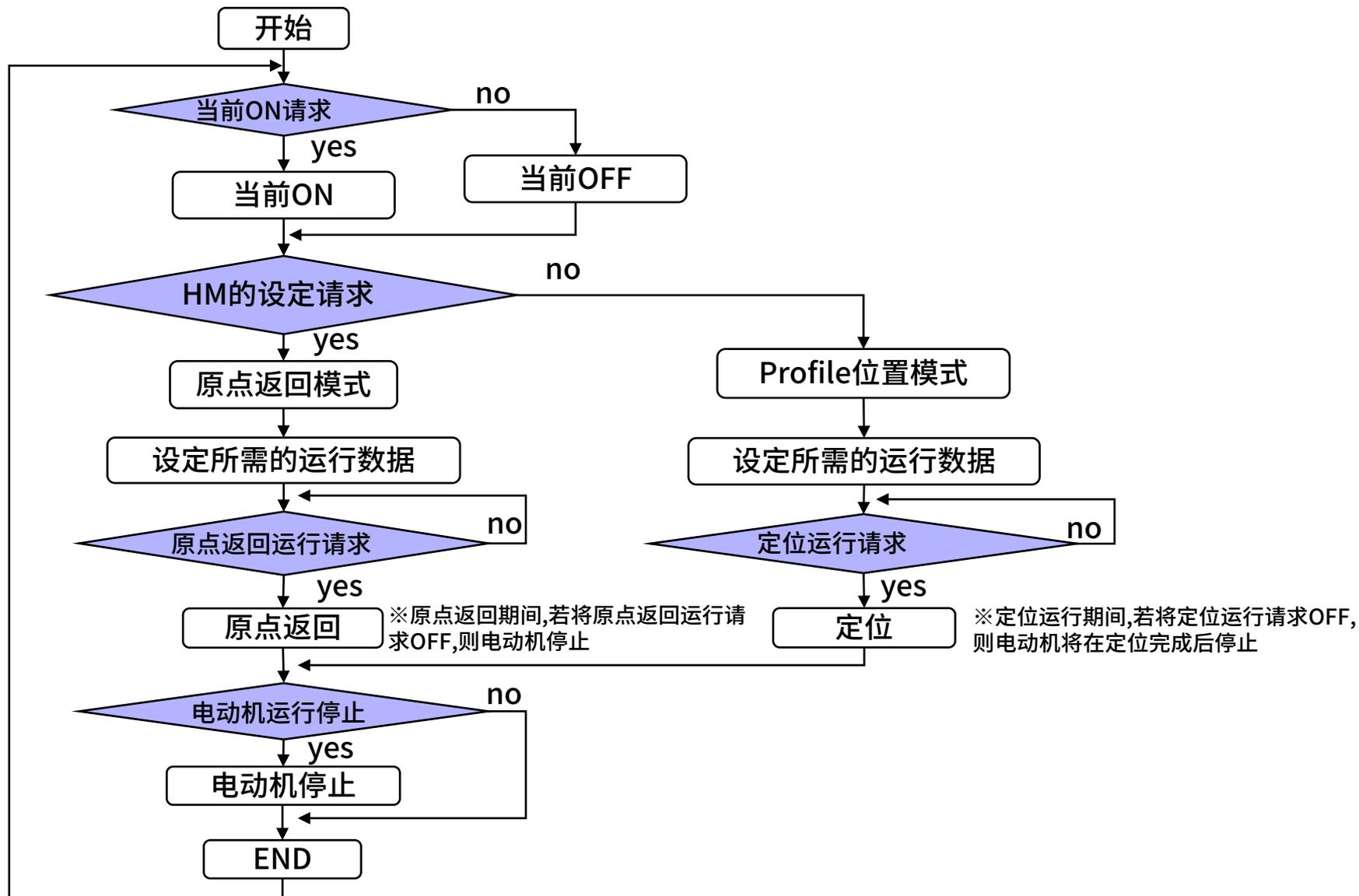
_Beckhoff_EtherCAT_PPHM_B

按同样的步骤,请将下表中的PLC变量链接设定到PDO映射所登录的对象(object)。



PLC变量	Link到的PDO映射
CTWD	Controlword
TPOS	Target Position
PVEL	Profile Velocity
OPMD	Modes of Operation

- 要制作的程序流程图如下所示。



PP/HM 样本程序

样本程序说明

ST编辑的样本程序如下所示。

```
1 PROGRAM MAIN
2 VAR
3 END_VAR
4
5 IF 1=bCON[0] THEN
6   CTWD[0]:= 15;
7 ELSE
8   CTWD[0]:= 6;
9 END_IF
10
11 IF bOPMD[0] THEN
12   OPMD[0]:= 6;
13 ELSE
14   OPMD[0]:= 1;
15 END_IF
16
17 IF OD[0] THEN
18   TPOS[0]:=1000000;
19   PVEL[0]:=10000;
20 END_IF
21
22 IF 1=bStart[0] AND (1=OPMD[0]) THEN
23   CTWD[0]:= CTWD[0] OR 16#0010;
24 END_IF
25
26 IF 1=bHome[0] AND (6=OPMD[0]) THEN
27   CTWD[0]:= CTWD[0] OR 16#0010;
28 END_IF
29
30 IF 1=bStop[0] THEN
31   CTWD[0]:= CTWD[0] OR 16#0100;
32 END_IF
```

①电动机励磁ON/OFF ※条件分支
“TRUE”：励磁ON 真：TRUE
“FALSE”：励磁OFF 假：FALSE

②PP/HM模式设定
“TRUE”：HM模式
“FALSE”：PP模式

③运行数据设定
“TRUE”：数据反映
“FALSE”：无反映

④相对定位运行
“TRUE”：定位运行开始
“FALSE”：无动作

⑤原点返回运行
“TRUE”：原点返回运行开始
“FALSE”：无动作

⑥电动机停止
“TRUE”：电动机停止
“FALSE”：无反映

①如右图所示编辑MAIN程序。

PP/HM 样本程序

执行文件下载到EPC

执行build，如果程序没有问题，请将其下载到EPC。

② 点击

⑤ 登录

④ 将EPC设定为RUN模式

⑥ 点击

⑦ 点击

③ 执行build，确认没有错误

Build Solution Ctrl+Shift+B

Rebuild Solution

Clean Solution

Build PP and HM

Rebuild PP and HM

Clean Selection

Batch Build...

Configuration Manager...

TwinCAT PLC Control

Application 'Port_851' does not exist on device 'TwinCAT_Device'. Do you want to create it and proceed with download?

Yes No Details...

Download done and autostart boot project active

Activate new boot project (next reboot will start new boot project)

Copy login information only (used to login from other PCs)

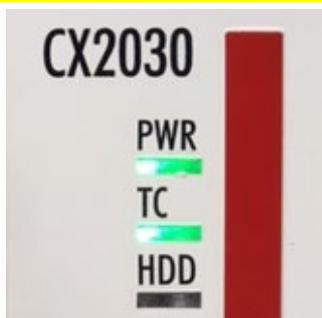
Hide this dialog until project will be reloaded (no action will be performed)

OK Cancel

Error List

0 Errors 0 Warnings 5 Messages Clear

注意：RUN模式下，EPC的TC LED为绿色亮灯。



PP/HM 样本程序

强行ON执行程序

登录后切换到以下画面。在线时还可确认全局变量的值。

The screenshot shows the TwinCAT PP and HM interface in Microsoft Visual Studio. The main window displays a ladder logic program for 'TwinCAT_Device.PP_and_HM.MAIN'. The program consists of several IF-THEN-ELSE blocks. The first block is highlighted with a blue box and a callout bubble: '① 点击开始' (Click Start). The code in this block is:

```
1 IF 1=bCON[0] FALSE THEN
2   CTWD[0] 6 := 15;
3   ELSE
4   CTWD[0] 6 := 6;
5 END_IF
```

A blue arrow points from this block to a callout bubble: '② 点击 FALSE→TRUE' (Click FALSE→TRUE). This indicates the modification of the condition in the IF statement. The resulting code is shown in a separate window:

```
1 IF 1=bCON[0] FALSE <TRUE> THEN
2   CTWD[0] 6 := 15;
3   ELSE
4   CTWD[0] 6 := 6;
5 END_IF
```

Another blue arrow points from this window to a final callout bubble: '④ CTWD的值可从6变更为15了。用以上的方法，改变条件，确认电动机是否按设定的程序运行。' (CTWD's value can be changed from 6 to 15. Use the above method to change the condition and confirm if the motor runs according to the set program.). The final code window shows the condition changed to TRUE:

```
1 IF 1=bCON[0] TRUE THEN
2   CTWD[0] 15 := 15;
3   ELSE
4   CTWD[0] 15 := 6;
5 END_IF
```

At the bottom of the screenshot, the Error List shows 0 Errors, 0 Warnings, and 0 Messages.

■ 下一页对程序做详细说明

电动机励磁ON/OFF

使用controlword执行状态迁移

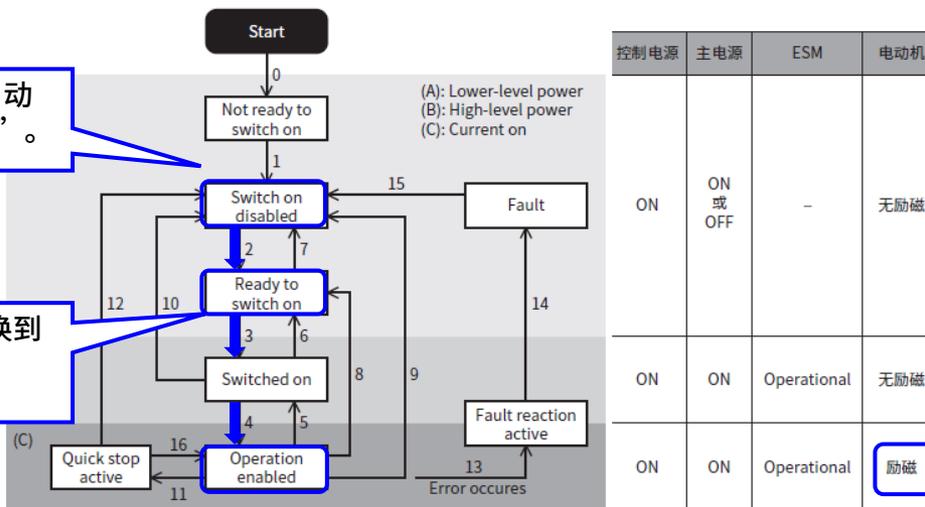
Target

切换驱动器状态机,励磁电动机

【驱动器状态机】

接通电源后,完成初始化后,自动切换到“Switch on disabled”。

controlword设定为6时,切换到“Ready to switch on”。处于当前OFF状态



controlword由6→15时,切换到“Operation enabled”当前ON。

Profile 位置模式的 controlword

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Manufacturer specific (ms)						oms	Halt
-	Wrap	Push	Base position of Rel	-	Reserved	Change on set point	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Operation mode specific (oms)				Enable operation	Quick stop	Enable voltage	Switch on
Fault reset	Abs/Rel	Change set immediately	New set point				

原点返回模式的 controlword

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Manufacturer specific (ms)						oms	Halt
-	-	-	-	-	Reserved	-	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Operation mode specific (oms)				Enable operation	Quick stop	Enable voltage	Switch on
Fault reset	-	-	Homing operation start				

利用 Controlword 的状态转换指令

状态控制指令	Bit7	Bit3	Bit2	Bit1	Bit0	图中的转换编号
Shutdown	-	-	1	1	0	2、6、8
Switch on	-	0	1	1	1	3*
Switch on + enable operation	-	1	1	1	1	3+4*
Disable voltage	-	-	-	0	-	7、9、10、12
Quick stop	-	-	0	1	-	7、10、11
Disable operation	-	0	1	1	1	5
Enable operation	-	1	1	1	1	4、16
Fault reset	0 → 1	-	-	-	-	15

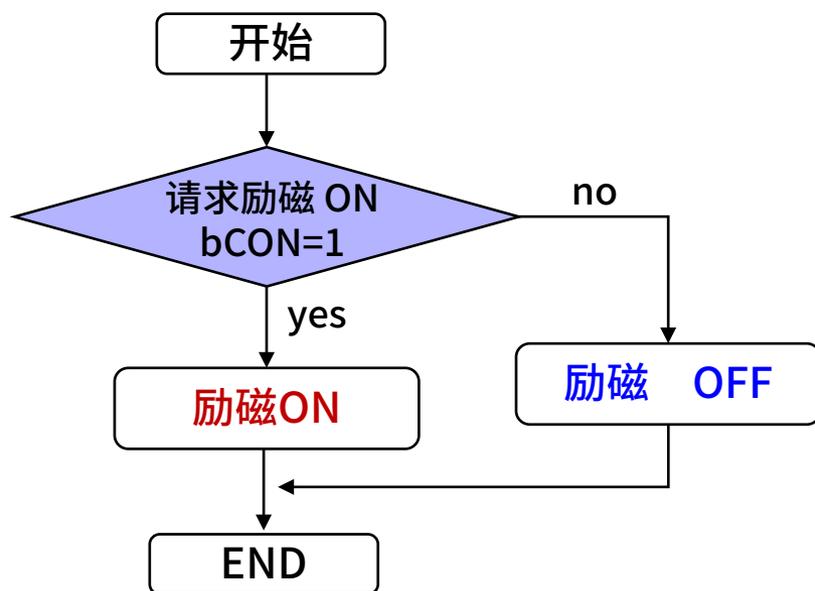
切换编号2: controlword 0→6

切换编号3+4: controlword 6→15

通过将状态切换指令写入controlword (6040 h) ,可切换驱动器状态机。
切换到“Operation Enabled”时,电动机自动当前ON(励磁)。

■程序示例 (Axis1)

- 流程图



- ST(Structured Text)

```
IF 1=bCON[0] THEN  
    CTWD[0]:= 15;  
ELSE  
    CTWD[0]:=6;  
END_IF
```

注意: []内数字表示轴数
[0]Axis1 [1]Axis2
[2]Axis3 [3]Axis4

Goal

用手旋转电动机输出轴, 确认是否励磁成功。

Profile位置模式 (PP) 的定位运行

样本程序的步骤说明

AZ多轴

Beckhoff_EtherCAT_PPHM_B

Target

在Profile位置模式 (PP) 下运行电动机

■ 按PP进行定位运行的步骤 (①~③顺序颠倒也没关系。)

①使电动机励磁ON
使其处于运行可能状态



②在Modes of operation(6060h)下
选择 1:PP



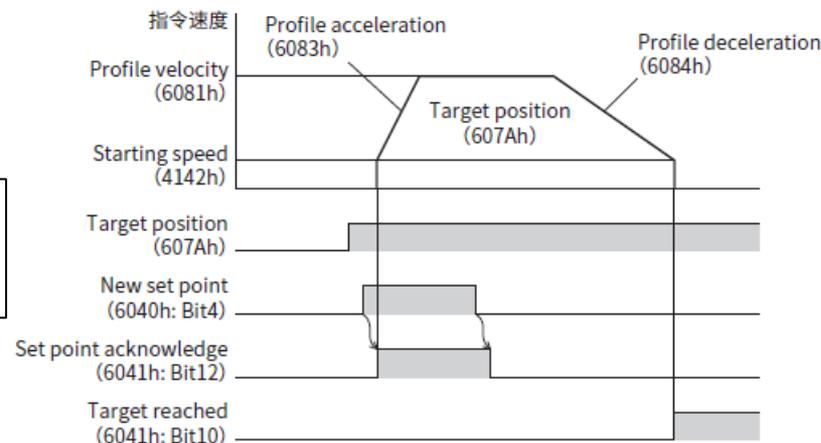
③运行数据设定



④controlword的Bit4:
New set point由0→1, 运行起动

● 定位运行

设定 Target position (607Ah), 并将 New set point (6040h: Bit4) 设为 1后, 将起动定位运行。



PP模式的定位运行

样本程序的说明 (Operation mode)

② Modes of operation (6060h) 的设定一览

Operation mode 的转换

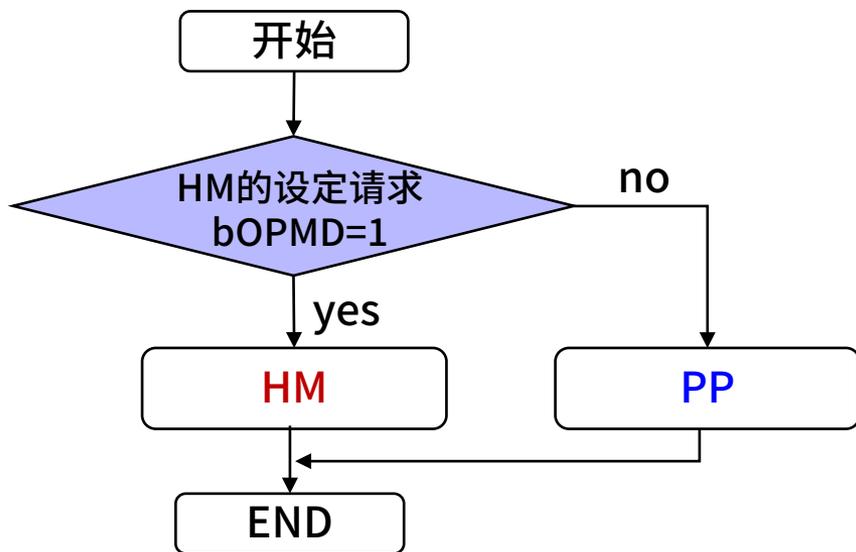
Operation mode 可通过 Modes of operation (6060h) 切换。

Operation mode 的设定值	Operation mode
0 (初始值)	运行功能无效
1	Profile 位置模式 (PP)
3	Profile 速度模式 (PV)
6	原点返回模式 (HM)
8	Cyclic 同步位置模式 (CSP)
9	Cyclic 同步速度模式 (CSV)

Operation mode, 请在电动机停止时转换。在运行过程中转换时, 新的 operation mode 将在停止运行后变为有效。可通过 Modes of operation display (6061h) 确认处于有效状态的 operation mode。

程序示例

流程图



ST(Structured Text)

```
IF bOPMD[0] THEN  
    OPMD[0]:=6;  
ELSE  
    OPMD[0]:=1;  
END_IF
```

PP模式的定位运行

样本程序的说明（运行数据设定）

③运行数据设定

PP的运行数据使用下表所示的各个Index。

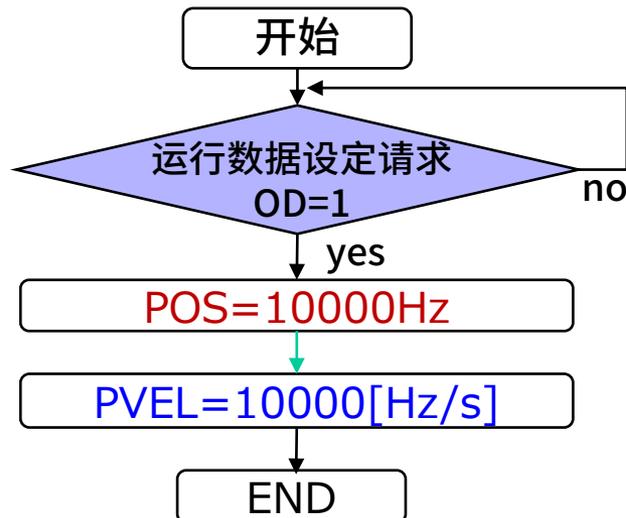
Index	Sub	名称	型	Access	PDO	保存	范围	反映
607Ah	00h	Target position [step]	INT32	RW	RxPDO	-	-2,147,483,648 ~ 2,147,483,647 (初始值: 0)	A
607Dh	01h	Min. position limit [step]	INT32	RW	No	○	-2,147,483,648 ~ 2,147,483,647 (初始值: -2,147,483,648)	A
	02h	Max. position limit [step]	INT32	RW	No	○	-2,147,483,648 ~ 2,147,483,647 (初始值: 2,147,483,647)	A
6081h	00h	Profile velocity [Hz]	U32	RW	RxPDO	○	0 ~ 4,000,000 (初始值: 10,000)	B
6083h	00h	Profile acceleration [step/s ²]	U32	RW	RxPDO	○	1 ~ 1,000,000,000 (初始值: 300,000)	B
6084h	00h	Profile deceleration [step/s ²]	U32	RW	RxPDO	○	1 ~ 1,000,000,000 (初始值: 300,000)	B

在此,变更Target position(607Ah)与Profile速度(6081h)。要使用其他参数时,请将其追加到PDO。

- Point** PDO以外的数据设定方法
- ①通过SDO进行变更
 - ②通过支援软件MEXE02进行变更

■程序示例

• 流程图



• ST(Structured Text)

```

IF OD[0] THEN
    TPOS[0]:=100000;
    PVEL[0]:=10000;
END_IF
  
```

PP模式的定位运行

样本程序的说明（起动定位运行）

④ controlword的Bit4: New set point由0→1, 运行起动

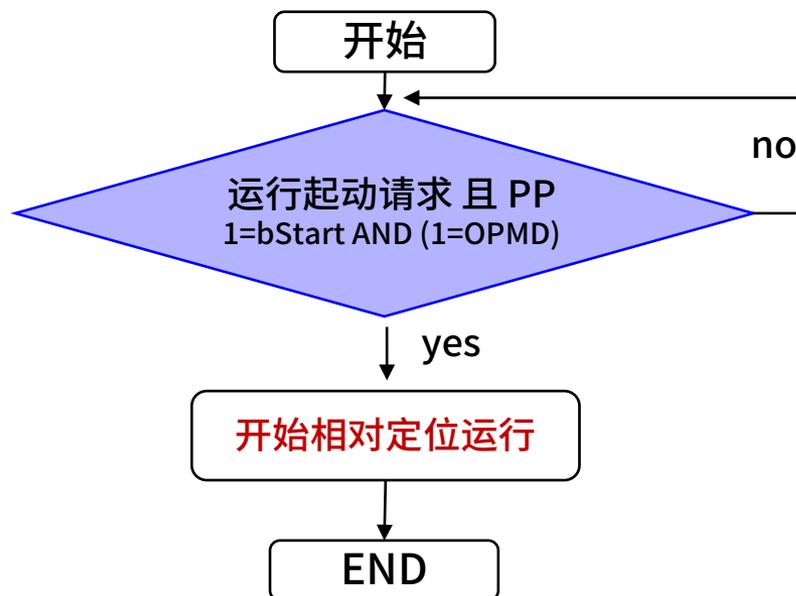
Profile 位置模式的 controlword

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Manufacturer specific (ms)						oms	Halt
-	Wrap	Push	Base position of Rel	-	Reserved	Change on set point	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Operation mode specific (oms)				Enable operation	Quick stop	Enable voltage	Switch on
Fault reset	Abs/Rel	Change set immediately	New set point				

Bit	名称	值	内容
6	Abs/Rel	0	绝对定位运行 Target position (607Ah) 为绝对定位运行的目标位置。
		1	相对定位运行 Target position (607Ah) 为相对定位运行的目标位置。
4	New set point	0 → 1	定位运行的起动 请在开始运行前选择 Wrap (6040h: Bit14)、Push (6040h: Bit13)、Base position of Rel (6040h: Bit12) 及 Abs/Rel (6040h: Bit6)。将 Halt (6040h: Bit8) 设为 1, 并从停止运行的状态起动定位运行时, 请先将 Halt (6040h: Bit8) 从 1 变更为 0, 空出通信周期 2 倍以上的间隔后, 再将 New set point (6040h: Bit4) 从 0 变更为 1。如果不空出通信周期 2 倍以上的间隔, 可能无法开始运行。 以下状态下, 将无法受理指令, 不开始运行。 <ul style="list-style-type: none">• Halt (6040h: Bit8) 变为 1。• STOP 输入变为 ON。• Drive State Machine 为 Operation enabled 以外• 电动机无励磁状态

程序示例

流程图



ST(Structured Text)

```
IF 1=GVLs.bStart AND (1=GVLs.OPMD) THEN  
  GVLs.CTWD:= GVLs.CTWD OR 16#0050;  
END_IF
```

PP模式的定位运行

补充：压推运行

可在PP模式下进行压推运行。请将controlword的Bit13:Push设为“1”，再使Bit4:New set point由0→1，起动运行。

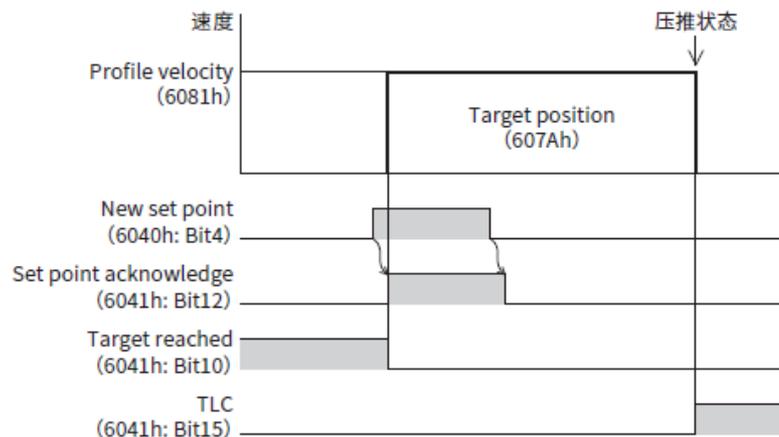
Point

controlword的Bit13和Bit4可同时设定

Profile 位置模式的 controlword

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Manufacturer specific (ms)					oms		
-	Wrap	Push	Base position of Rel	-	Reserved	Change on set point	Halt
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Operation mode specific (oms)				Enable operation	Quick stop	Enable voltage	Switch on
Fault reset	Abs/Rel	Change set immediately	New set point				

压推到负载时



Bit	名称	值	内容
13	Push	1	压推定位运行 将 Push 设为 1，再将 New set point (6040h: Bit4) 设为 1 并开始运行后，将开始压推定位运行。电动机电流变为 Push current (4121h)。将 Halt (6040h: Bit8) 为 1、或 STOP 输入设定为 ON，停止运行。停止状态下的电动机电流为 Stop current (4128h)。
6	Abs/Rel	0	绝对定位运行 Target position (607Ah) 为绝对定位运行的目标位置。
		1	相对定位运行 Target position (607Ah) 为相对定位运行的目标位置。
4	New set point	0 → 1	定位运行的起动 请在开始运行前选择 Wrap (6040h: Bit14)、Push (6040h: Bit13)、Base position of Rel (6040h: Bit12) 及 Abs/Rel (6040h: Bit6)。将 Halt (6040h: Bit8) 设为 1，并从停止运行的状态起动定位运行时，请先将 Halt (6040h: Bit8) 从 1 变更为 0，空出通信周期 2 倍以上的间隔后，再将 New set point (6040h: Bit4) 从 0 变更为 1。如果不空出通信周期 2 倍以上的间隔，可能无法开始运行。 以下状态下，将无法受理指令，不开始运行。 <ul style="list-style-type: none">• Halt (6040h: Bit8) 变为 1。• STOP 输入变为 ON。• Drive State Machine 为 Operation enabled 以外• 电动机无励磁状态

HM模式进行原点返回运行

样本程序的步骤说明

Target

在 原点返回模式 (HM) 下进行原点返回运行

- 按HM进行原点返回运行的步骤 (①~③顺序颠倒也没关系。)

①使电动机励磁ON
使其处于运行可能状态



②在Modes of operation (6060h) 下
选择『6: HM』



③设定所需的运行数据

※原点返回方法、原点返回速度等
请通过AZ驱动器的参数设定。



④controlword的Bit4:
Homing operation start由0→1,运行起动

HM模式进行原点返回运行

选择原点返回方式

③ 设定所需的运行数据

与原点返回运行相关的参数如下所示。请根据需要进行设定。

Index	Sub	名称	型	Access	PDO	保存	范围	反映
607Ch	00h	Home offset [step]	INT32	RW	No	<input type="radio"/>	-2,147,483,648 ~ 2,147,483,647 (初始值: 0)	A
6098h	00h	Homing method	INT8	RW	No	<input type="radio"/>	17、18、24 (初始值)、 28、35、37、-1 (⇒ “原 点返回方法选择”)	B
6099h	01h	Speed during search for switch [Hz]	U32	RW	No	<input type="radio"/>	1 ~ 4,000,000 (初始值: 10,000)	B
	02h	Speed during search for zero [Hz]	U32	RW	No	<input type="radio"/>	1 ~ 10,000 (初始值: 5,000)	B
609Ah	00h	Homing acceleration [step/sec ²]	U32	RW	No	<input type="radio"/>	1 ~ 1,000,000,000 (初始值: 300,000)	B

■ 原点返回方法选择

原点返回方法在 Homing method (6098h) 中选择。驱动器支持以下原点返回方法。

原点返回方法	内容
17	利用限位传感器 (FW-LS/RV-LS) 执行原点返回, 向负方向起动
18	利用限位传感器 (FW-LS/RV-LS) 执行原点返回, 向正方向起动
24	利用原点传感器 (HOMES) 执行原点返回, 向正方向起动
28	利用原点传感器 (HOMES) 执行原点返回, 向负方向起动
35、37*	原点预置
-1	本公司规格的原点返回

Point

原点返回方法17~37是基于 CiA402 drive profile 的原点返回模式。压推原点返回方法等是本公司独有的规格, 请选择-1。

HM模式进行原点返回运行

MEXE02的原点返回方法的设定



原点返回方法的设定可由TwinCAT或MEXE02来执行。
在此介绍MEXE02的设定方法。

①从“通信端口设定”
中打开通信端口。

②点击

④点击。写入驱动器。

【补充】变更原点返回运行的各种参数时，请将“电动机机构（坐标/JOG/原点返回设定”的“JOG/HOME/ZHOME operation setting”设定为“手动设定”。
0：优先ABZO设定（初始设定）
1：手动设定

The screenshot shows the MEXE02 software interface with several callouts and a detailed parameter table. Callout ② points to the '通信' (Communication) menu. Callout ④ points to the '写入驱动器' (Write to drive) button. Callout ⑤ points to the '组合信息监视' (Combined information monitoring) section. A callout box on the right explains the 'JOG/HOME/ZHOME operation setting' options. A red-bordered table shows the '驱动器参数' (Drive parameters) and 'ABZO(固定值)' (ABZO (fixed value)) for various homing methods.

驱动器参数	ABZO(固定值)
优先ABZO设定	无设定
0	0
0	0
0	0
+侧=CCW	+侧=CCW
优先ABZO设定	无设定
0 [ms]	0 [ms]
0.0 [%]	0.0 [%]
0 [step]	0 [step]
0 [Hz]	0 [r/min]
0.000 [kHz/s]	0.000 [s]
0 [Hz]	0 [r/min]
0.000 [kHz/s]	0.000 [s]
0 [Hz]	0 [r/min]
2传感器	2传感器
-方向	-方向

参数名	值	注释
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	10000	10000
23	5000	5000
24	300000	300000

HM模式进行原点返回运行

样本程序的说明 (起动原点返回运行)

④ controlword的Bit4: 使Homing operation start由0→1, 运行起动

原点返回模式的 controlword

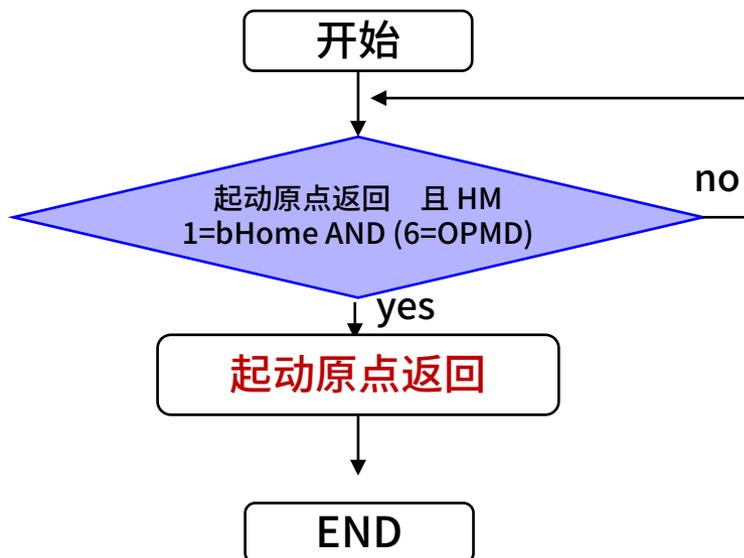
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Manufacturer specific (ms)						oms	Halt
-	-	-	-	-	Reserved	-	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Operation mode specific (oms)				Enable operation	Quick stop	Enable voltage	Switch on
Fault reset	-	-	Homing operation start				

Controlword 的详细

Bit	名称	值	内容
8	Halt	0	运行允许
		1	停止运行。停止方法遵照 Halt option code (605Dh) 的设置。
4	Homing operation start	0 → 1	原点返回运行的起动 如果在原点返回运行过程中将 Homing operation start 设为 0, 将减速停止。 以下任一状态下, 将无法受理指令, 不开始运行。 <ul style="list-style-type: none"> • 运行中 • Halt (6040h: Bit8) 变为 1。 • STOP 输入变为 ON。 • Drive State Machine 为 Operation enabled 以外 • 电动机无励磁状态

程序示例

流程图



ST(Structured Text)

```

IF 1=bHome[0] AND (6=OPMD[0]) THEN
    CTWD[0]:= CTWD[0] OR 16#0010;
END_IF
    
```

补充：PP/HM模式下运行 样本程序（通过Halt停止电动机）的说明

通过Halt option code(605Dh)，可以决定当Controlword的Bit8:Halt设定为1时电动机的停止方法。

设定值	内容
1	按照 Profile deceleration (6084h) 减速停止。停止后，保持 Operation enabled。
2	按照 Quick stop deceleration (6085h) 减速停止。停止后，保持 Operation enabled。
3	立即停止。停止后，保持 Operation enabled。

Bit	名称	值	内容
8	Halt	0	允许运行
		1	停止运行。停止方法遵照 Halt option code (605Dh) 的设定。

PP模式或HM模式，都可以选择Halt。

原点返回模式的 controlword

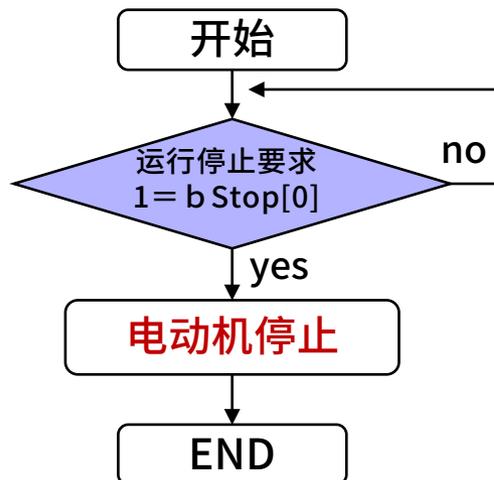
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Manufacturer specific (ms)						Reserved	oms
-	-	-	-	-	-	-	Halt
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Operation mode specific (oms)				Enable operation	Quick stop	Enable voltage	Switch on
Fault reset	-	-	Homing operation start				

Profile 位置模式的 controlword

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Manufacturer specific (ms)						oms	
-	Wrap	Push	Base position of Rel	-	Reserved	Change on set point	Halt
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Operation mode specific (oms)				Enable operation	Quick stop	Enable voltage	Switch on
Fault reset	Abs/Rel	Change set immediately	New set point				

程序示例(Axis1)

流程图



ST(Structured Text)

```

IF 1=bStop[0] THEN
    CTWD[0]:= CTWD[0] OR 16#0100;
END_IF
  
```

通过SDO通信读写参数

■驱动器的参数设定有以下3种方法。

- ① SDO通信的FB(功能块)
- ② BECKHOFF的TwinCAT
- ③ 支援软件MEXE02（可从敝公司的网站免费下载）

■关于参数的保存

写入的参数保存在驱动器的RAM或NV存储器中。
切断驱动器的控制电源后,写入RAM的参数将被删除。
即使切断控制电源也会保存写入NV存储器的参数。

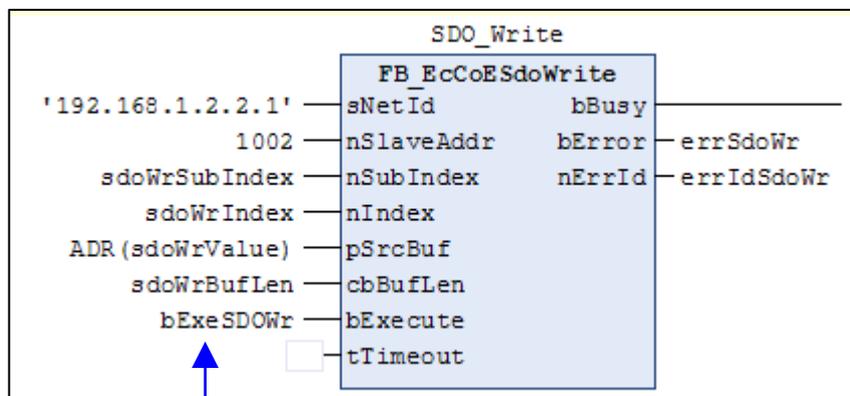
①②保存在RAM中。要将写入RAM的参数保存到NV存储器,请进行“NV存储器批量写入(40C9h)”。

NV存储器的可写入次数约为10万次。

③保存在NV存储器中。

●下面介绍如何使用SDO通信的FB写入参数。

• 编程例



例：将运行电流设定为75.0%时,输入运行电流的Index、Sub-Index、类型(数据长度(Data length)),然后执行写入。

Index	Sub	名称	型
4120h	00h	Operating current [1=0.1%]	INT16

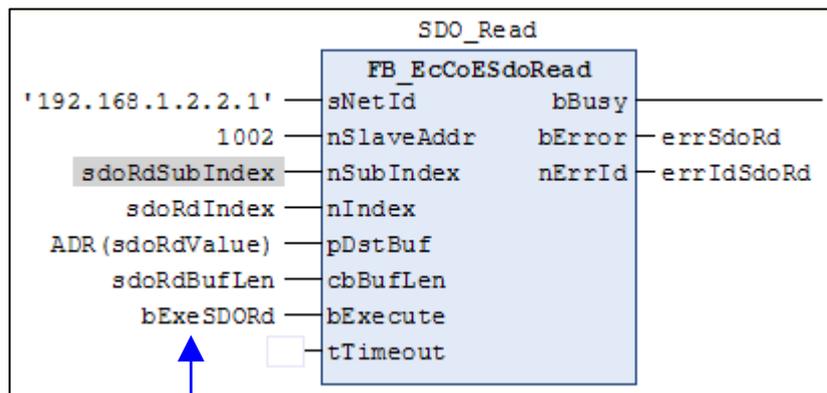
• FB的输入参数

```
SdoWrIndex := 16#4120;  
SdoWrSubIndex := 1; //Axis1  
SdoWrBufLen := 2; //2byte  
SdoWrValue := 750; //0.1%单位
```

参数写入，
当使bExecute由FALSE(0)→为TRUE(1)时,执行写入。

●下面介绍如何使用SDO通信的FB读取参数。

• 编程例



例：读取轴1的运行电流时，输入运行电流的Index、Sub-Index、类型(数据长度(Data length)),然后执行读取。

Index	Sub	名称	型
4120h	00h	Operating current [1=0.1%]	INT16

• FB的输入参数

```
SdoRdIndex := 16#4120;
SdoRdSubIndex := 1; //Axis1
SdoRdBufLen := 2; //2byte
```

输入要读取对象(object)的Index、Sub-Index、字节长度，当使bExecute由FALSE(0)→为TRUE(1)时,执行读取。读取的结果存储在SdoRdValue中。

通过SDO通信读写参数 添加Library

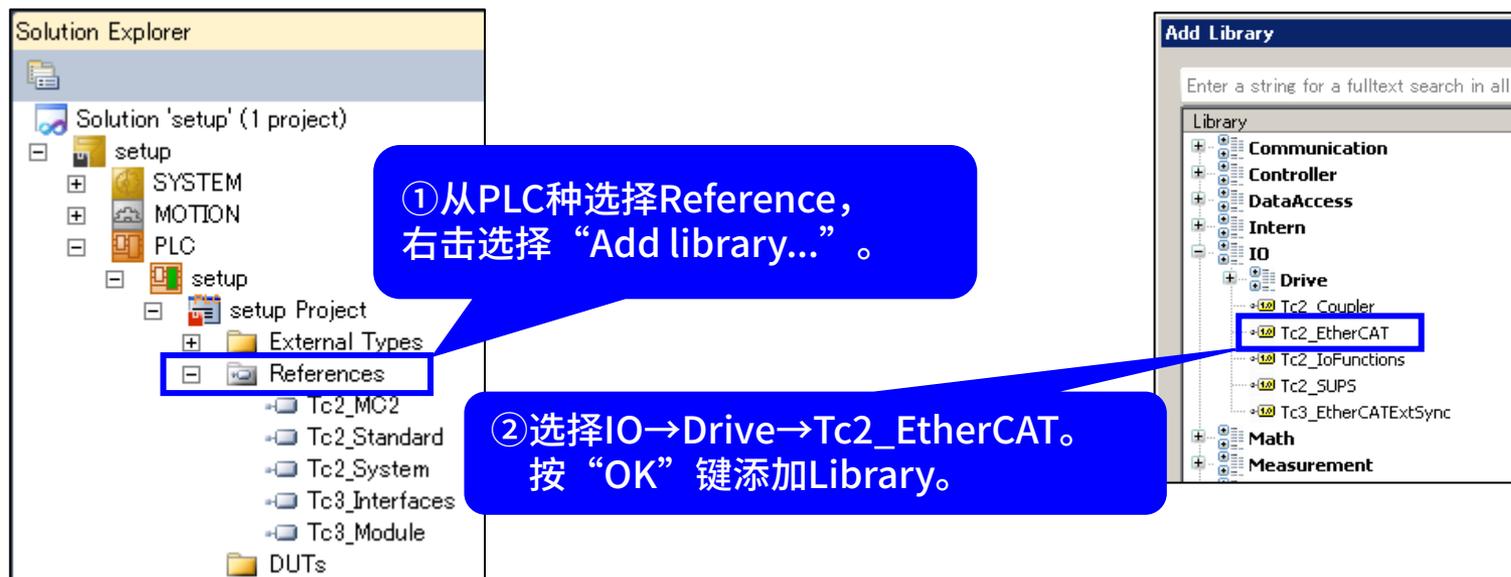
Target

用SDO通信读写参数。

使用SDO读写参数,需要追加以下的FB。

- 写入参数: FB_EcCoESdoWrite
- 读取参数: FB_EcCoESdoRead

使用FB_EcCoESdoWrite, FB_EcCoESdoRead, 请按以下步骤添加Tc2_EtherCAT的Library。



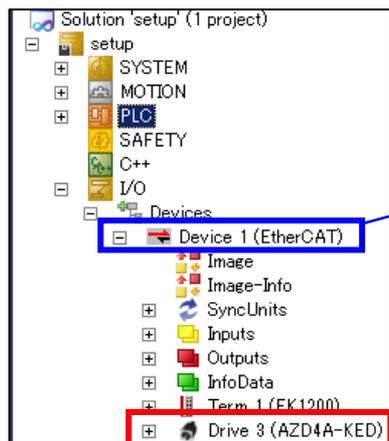
①从PLC种选择Reference, 右击选择“Add library...”。

②选择IO→Drive→Tc2_EtherCAT。按“OK”键添加Library。

通过SDO通信读写参数

FB输入参数的设定内容

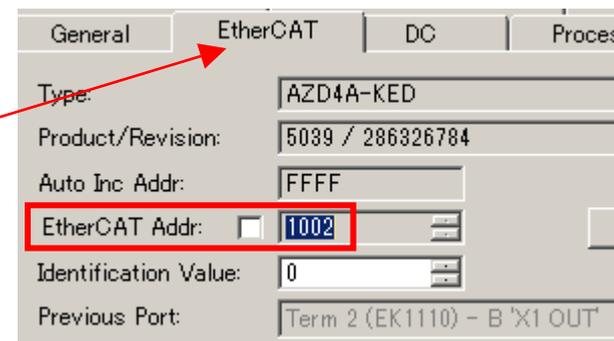
• 确认写入到FB的输入参数值



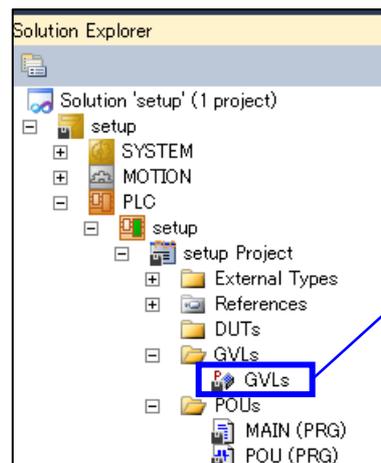
“sNetId” 的确认



“sSlaveAddr” 的确认



• 分配到各种输入参数的Global变量一览



```

VAR_GLOBAL
  bExeSDOWr      : BOOL;
  SdoWrValue     : DINT;
  SdoWrIndex     : WORD;
  SdoWrSubIndex  : BYTE;
  SdoWrSrcBuf    : PVOID;
  SdoWrBufLen    : UDINT;
  ErrSdoWr       : BOOL;
  ErrIdSdoWr     : UDINT;
END_VAR
    
```

例)写入用Global变量

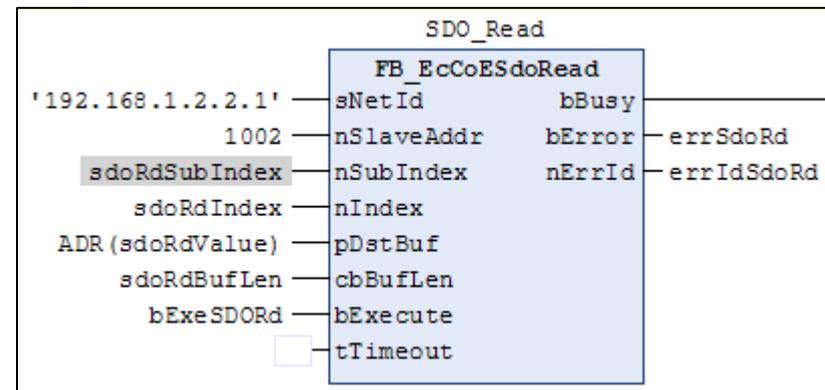
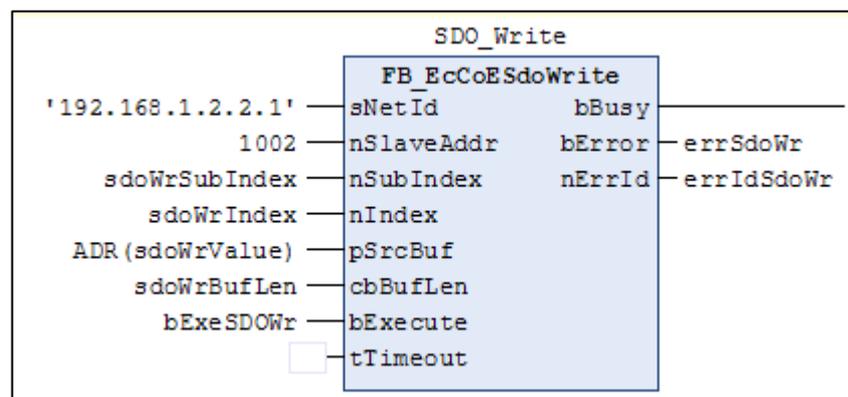
```

VAR_GLOBAL
  bExeSDORd      : BOOL;
  SdoRdValue     : DINT;
  SdoRdIndex     : WORD;
  SdoRdSubIndex  : BYTE;
  SdoRdSrcBuf    : PVOID;
  SdoRdBufLen    : UDINT;
  ErrSdoRd       : BOOL;
  ErrIdSdoRd     : UDINT;
END_VAR
    
```

例)读取用Global变量

FB_EcCoESdoWrite , FB_EcCoESdoRead的输入参数

- 程序例（使用View as function block diagram显示）



输入参数	输入的内容
sNetId	EtherCAT主站的AMS networkID
nSlaveAddr	EtherCAT从站地址
nSubIndex	读写对象的Sub-Index
nIndex	输入要读写对象的Index
pSrcBuf	指定读写用的数据缓冲器的地址。 制作数据缓冲器用的变量，用ADR()进行地址转换
cbBufLen	读写对象的最大字节长度

在GVLs变量种添加SDO用参数

将他们作为FB的输入参数使用

```
GVLs x
10 TPOS AT%Q*:ARRAY[0..3] OF DINT:= [10000,10000,10000,10000]; //Target
11 PVEL AT%Q*:ARRAY[0..3] OF DINT:= [10000,10000,10000,10000]; //Profile
12 OPMD AT%Q*:ARRAY[0..3] OF SINT:= [1,1,1,1]; //Operati
13
14 //SDO Write
15 bExeSDOWr AT%I*: BOOL; //SDO Write execution
16 sdoWrValue AT%I*: DINT; //Buffer Data for writing
17 sdoWrIndex AT%I*: WORD; //Index
18 sdoWrSubIndex AT%I*: BYTE; //Sub-Index
19 sdoWrSrcBuf AT%I*: PVOID; //Reference address of buffer data
20 sdoWrBufLen AT%I*: UDINT; //Byte length
21 errSdoWr AT%Q*: BOOL; //Error bit
22 errIdSdoWr AT%Q*: UDINT; //Error ID
23
24 //SDO Read
25 bExeSDORd AT%I*: BOOL; //SDO Read execution
26 sdoRdValue AT%Q*: DINT; //Buffer Data for reading
27 sdoRdIndex AT%I*: WORD; //Index
28 sdoRdSubIndex AT%I*: BYTE; //Sub-Index
29 sdoRdSrcBuf AT%I*: PVOID; //Reference address of buffer data
30 sdoRdBufLen AT%I*: UDINT; //Byte length
31 errSdoRd AT%Q*: BOOL; //Error bit
32 errIdSdoRd AT%Q*: UDINT; //Error ID
33 END_VAR
```

通过SDO通信读写参数

编辑SDO通信用程序

在POU中添加SDO的程序。

① 右击

② 点击

③ 输入程序名称
例如 POU_SDO

④ 选择FBD

⑤ 点击

POU...

POU for implicit checks...

DUT...

Global Variable List...

Referenced Task...

Visualization...

Visualization Manager...

Recipe Manager...

Image Pool...

Interface...

Parameter List...

Text List...

Class Diagram...

Existing Item... Shift+Alt+A

New Folder

Existing Folder Content...

Add

- Import PLCOpenXML...
- Export PLCOpenXML...
- Cut Ctrl+X
- Copy Ctrl+C
- Delete Del
- Rename
- Properties Alt+Enter

Create a new POU (Program Organization Unit)

Name: POU_SDO

Type:

- Program
- Function Block
 - Extends: [] ...
 - Implements: [] ...
- Function

Access specifier: []

Method implementation language: Function Block Diagram (FBD)

Return type: []

Implementation language: Function Block Diagram (FBD)

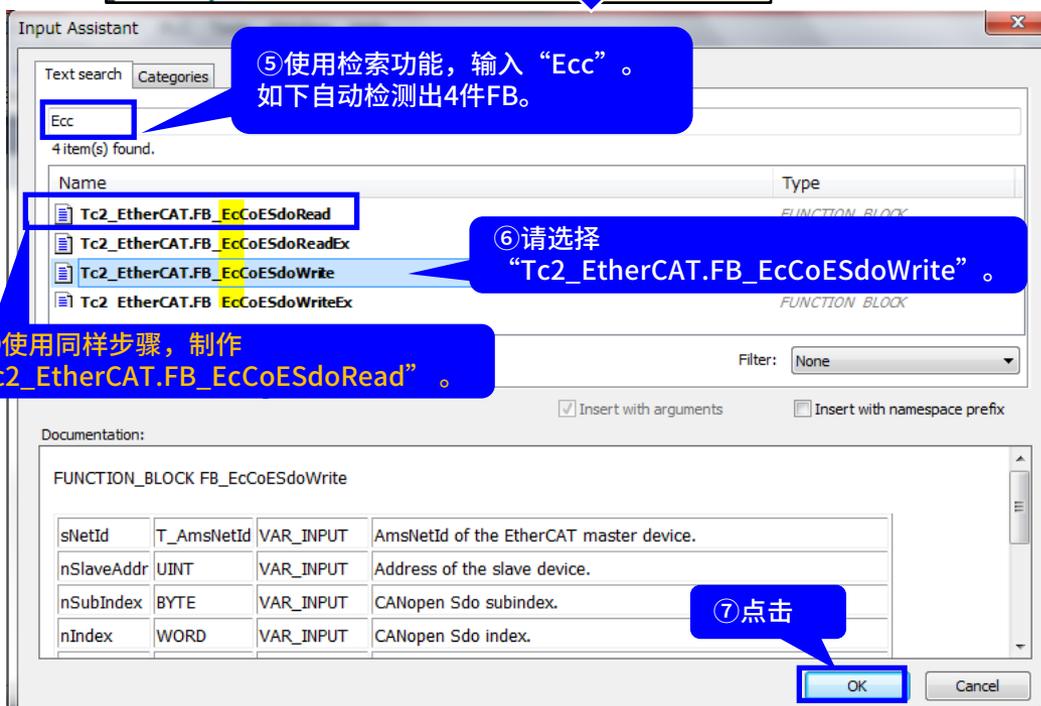
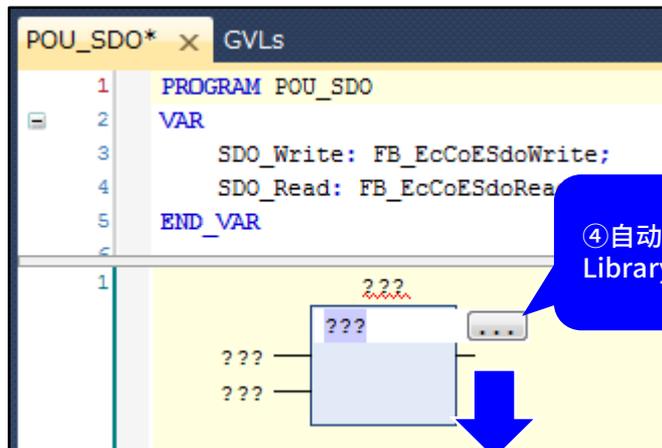
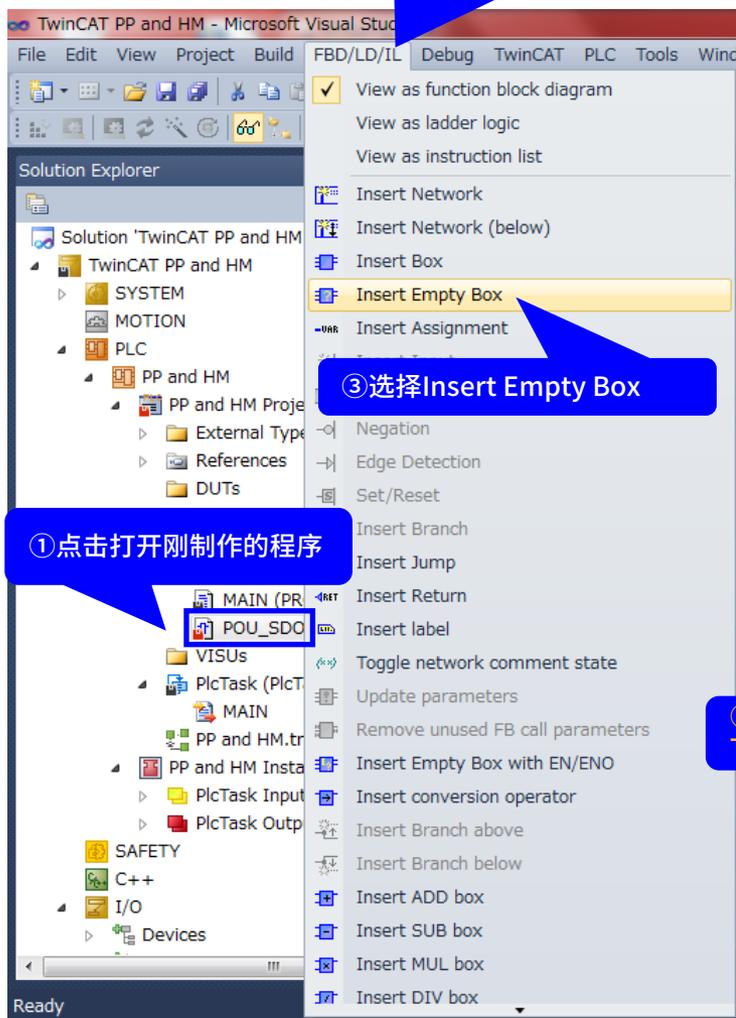
Open Cancel

Description	File	Line	Column	Project
0 Errors				
0 Warnings				
5 Messages				

通过SDO通信读写参数 使用功能块

制作SDO读写用的程序。

②选择需要插入FB的行,点击菜单



通过SDO通信读写参数

样本程序的说明

设定FB的输出输入变量

POU_SDO* x GVLs

```
1 PROGRAM POU_SDO
2 VAR
3   SDO_Write: FB_EcCoESdoWrite;
4   SDO_Read: FB_EcCoESdoRead;
5 END_VAR
```

①请分别设定以下红色部分。

SDO_Write

'192.168.1.2.2.1'	sNetId	bBusy	
1002	nSlaveAddr	bError	errSdoWr
sdoWrSubIndex	nSubIndex	nErrId	errIdSdoWr
sdoWrIndex	nIndex		
ADR (sdoWrValue)	pSrcBuf		
sdoWrBufLen	cbBufLen		
bExeSDOWr	bExecute		
<input type="checkbox"/>	tTimeout		

SDO_Read

'192.168.1.2.2.1'	sNetId	bBusy	
1002	nSlaveAddr	bError	errSdoRd
sdoRdSubIndex	nSubIndex	nErrId	errIdSdoRd
sdoRdIndex	nIndex		
ADR (sdoRdValue)	pDstBuf		
sdoRdBufLen	cbBufLen		
bExeSDORd	bExecute		
<input type="checkbox"/>	tTimeout		

在MAIN程序中添加POU_SDO。

MAIN x POU_SDO* GVLs

```
1 PROGRAM MAIN
2 VAR
3 END_VAR
4
5 //SDO READ/WRITE
6 POU_SDO();
7
8 //Controlword
9 IF 1=bCON[0] THEN
10   CTWD[0]:= 15;
```

②Main程序添加一行。
POU_SDO();

通过SDO通信读写参数

将执行文件下载到EPC

执行Build，确认没有错误后，下载到EPC。

① 点击

② Build后，确认内容是否正确。

③ 将EPC设定为RUN模式

④ 登录

⑤ 点击

⑥ 点击

注意：在RUN模式中，EPC的TC LED亮绿灯。

Ready

```
1=bCON[0] THEN
CTWD[0]:= 15;
ELSE
CTWD[0]:= 6;
END_IF

IF bOPMD[0] THEN
OPMD[0]:= 6;
ELSE
OPMD[0]:= 1;
END_IF

IF OD[0] THEN
IPOS[0]:=1000;
PVEL[0]:=1000;
END_IF

IF 1=bStart[0] AND
CTWD[0]:= CTWD[0];
END_IF

IF 1=bHome[0] AND
CTWD[0]:= CTWD[0] OR 16#0010;
END_IF

IF 1=bStop[0] THEN
CTWD[0]:= CTWD[0] OR 16
```

Error List
0 Errors 0 Warnings 5 Messages Clear

通过SDO通信读写参数

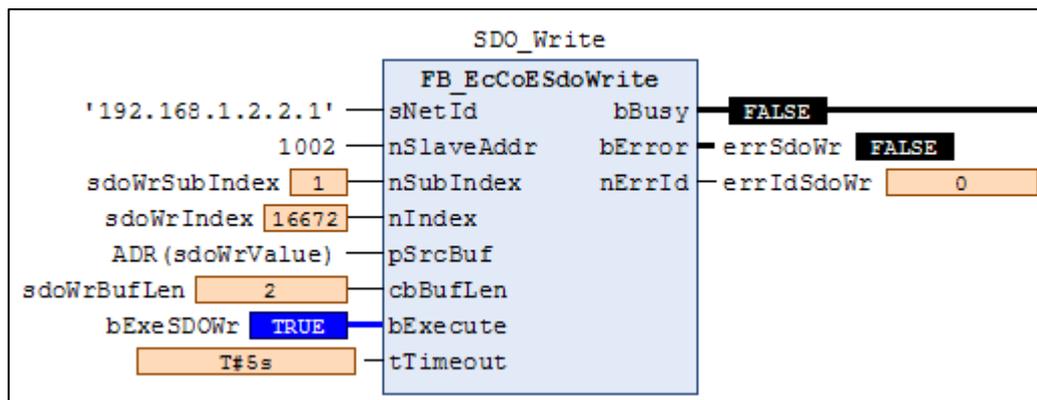
强行更改输入参数

登录后，从GVLs[Online]可以写入SDO写入用的FB输入变量。

Expression	Type	Value	Prepared value	Address
OPMD	ARRAY [0..3...]			%Q*
bExeSDOWr	BOOL	FALSE	TRUE	%I*
sdoWrValue	DINT	0	750	%I*
sdoWrIndex	WORD	0	16672	%I*
sdoWrSubIndex	BYTE	0	1	%I*
sdoWrSrcBuf	UDINT	0		%I*
sdoWrBufLen	UDINT	0	2	%I*
errSdoWr	BOOL	FALSE		%Q*
errIdSdoWr	UDINT	0		%Q*
bExeSDORd	BOOL	FALSE		%I*
sdoRdValue	DINT	0		%Q*

Expression	Type	Value	Prepared value
OPMD	ARRAY [0..3...]		
bExeSDOWr	BOOL	TRUE	
sdoWrValue	DINT	750	
sdoWrIndex	WORD	16672	
sdoWrSubIndex	BYTE	1	
sdoWrSrcBuf	UDINT	0	
sdoWrBufLen	UDINT	2	
errSdoWr	BOOL	FALSE	
errIdSdoWr	UDINT	0	
bExeSDORd	BOOL	FALSE	
sdoRdValue	DINT	0	

●通过FB的输出参数也可以确认是否发生错误。



通过SDO通信读写参数

通过MEXE02确认写入内容

确认SDO写入的“运行电流[%]”值。



使用支援软件“MEXE02”，确认参数。软件可从本公司官网免费下载。
<https://www.orientalmotor.co.jp/download/software/mexe02/>

① 点击

	Profile area 的对象	Axis 1	Axis 2
1	Motor user memo		
2	Driver user memo		
3			
4	Driver simulation mode	实际使用电动机	
5	Main power mode	自动识别	
6			
7	Base current [%]	100.0	100.0
8	Operating current [%]	75.0	100.0
9	Stop current [%]	50.0	50.0
10	Push current [%]	20.0	20.0
11	Command filter setting	LPF(速度平滑调整)	LPF(速度平滑调整)
12	Command filter time constant [ms]	1	1

② 运行电流[%]的初始值为100.0，通过SDO通信成功将其修改为75.0。

Point

PC与AZ多轴驱动器连接时使用的USB电缆线(连接器A型-miniB型)需要另行购买。使用MEXE02设定软件，可以确认EtherCAT通信时AZ多轴驱动器是否按设计接受到数据。对于设备前期准备时的测评，以及发生警报时的原因调查都能起到很大的帮助。

通过SDO通信读写参数

确认读取的内容

从GVLs[Online]可以写入SDO写入用的FB输入变量。
读取已写入的运行电流值，确认是否相同。

Expression	Type	Value	Prepared value
sdoWrSrcBuf	UDINT	0	
sdoWrBufLen	UDINT	2	
errSdoWr	BOOL	FALSE	FALSE
errIdSdoWr	UDINT	0	
bExeSDORd	BOOL	FALSE	TRUE
sdoRdValue	DINT	0	
sdoRdIndex	WORD	0	16672
sdoRdSubIndex	BYTE	0	1
sdoRdSrcBuf	UDINT	0	
sdoRdBufLen	UDINT		

②点击。写入 Prepared value的值。

①输入各变量的Prepared Value。

Expression	Type	Value	Prepared value
sdoWrSrcBuf	UDINT	0	
sdoWrBufLen	UDINT	2	
errSdoWr	BOOL	FALSE	
errIdSdoWr	UDINT	0	
bExeSDORd	BOOL	TRUE	
sdoRdValue	DINT	750	
sdoRdIndex	WORD	16672	
sdoRdSubIndex	BYTE	1	
sdoRdSrcBuf	UDINT	0	
sdoRdBufLen	UDINT	2	

③在SDO通信中，成功的读取了数值。

●可以确认FB的输出参数是否发生错误。

