

EtherCAT远程I/O



使用手册 ERM200系列(ERM242-40DTD) 多功能型EtherCAT远程I/O



目录

第一章	产品规格.....	- 1 -
1.1	规格列表.....	- 1 -
1.2	产品外观.....	- 6 -
1.3	产品尺寸.....	- 6 -
第二章	使用说明.....	- 8 -
2.1	组态准备.....	- 8 -
2.2	设备组态.....	- 10 -
2.3	数据映射.....	- 11 -
2.4	编程调试.....	- 13 -
2.5	参数配置.....	- 15 -
2.6	HSC 功能使用.....	- 18 -
2.7	CiA 402 轴与脉冲输出使用.....	- 22 -
2.8	ExtBus 拓展总线的使用.....	- 24 -
2.9	数据掉电保持的使用.....	- 30 -
第三章	EtherCAT 数据对象.....	- 37 -
3.1	自定义 EtherCAT 数据对象.....	- 37 -
3.2	标准 EtherCAT 数据对象.....	- 42 -
第四章	FAQ.....	- 47 -
敬告用户：	- 48 -

第一章 产品规格

ERM242-40DTD 是一款多功能型 EtherCAT 分布式 I/O，集成了数字量输入\输出、模拟量输入\输出、脉冲输入（差分信号）\输出以及锁存等功能。此外，内置 ExtBus 总线作为 EtherCAT 二级总线，可以连接 ExtBus 拓展模块。内置数据掉电保持功能，可以通过 PDO 循环写入数据，建议搭配本公司编写的 CoDeSys 功能块使用。

1.1 规格列表

表格 1 规格列表

编号	名称	数量	规格
1	数字量输入	24	默认滤波 3ms，可配置滤波范围(0~15ms)。
2	数字量输出	16	-
3	模拟量输入	2	0~10V,12bit，出厂校准，数值范围 0~10000。 (0~20mA,12bit，出厂校准，数值范围 0~20000) 有超量程报警指示灯。
4	模拟量输出	2	0~10V,12bit，出厂校准，数值范围 0~10000。 (0~20mA,12bit，出厂校准，数值范围 0~20000。)
5	脉冲输出	2	共用数字量输出的前 4 个点(bit0~bit3)。 最大频率 200kHz。 支持脉冲+方向、CW/CCW 和 AB 等 3 种形式。 可映射到 CiA 402 轴。
6	脉冲输入	2	内置于 DB15 接口内，最大频率 200kHz，差分信号。 支持脉冲+方向、CW/CCW、AB 和 AB4 等 4 种形式。 对应 HSC 信号指示灯。 脉冲输入端口 0，支持 ABZ 三相信号，Z 向可以用于清空编码器计数值或者作为锁存信号使用。
7	脉冲输入锁存	4	每个脉冲输入有 1 个锁存点，分别位于端子。 脉冲输入端口 1 的 Z 向信号，可以配置为锁存信号。 指示灯 LE 指示端子锁存点的状态。 触发条件可以配置上升沿或下降沿。 可配置滤波倍数。
8	ExtBus 拓展	1	ExtBus 作为 EtherCAT 的二级总线，最多可以拓展 8 个 ExtBus 模块

			ExtBus 上的模块的 I/O 信号, 可以被映射到 EtherCAT 网络中, 作为 EtherCAT 的 I/O 信号使用。
9	数据掉电保持	1	支持 192 个字节, 分为 3 个区, 每个区有 64 个字节。 可以通过 Data ID 标记数据区, 写入不同的数据区将会清空数据掉电保持区的数据。 *建议搭配本公司编写的 CoDeSys 功能块使用, 使用功能块的写入周期约为 8~9 个周期。 *无写入次数限制。
10	EtherCAT 通信	1	2 个 RJ45 网口, 通信速率 100Mbps。 SM 和 DC 同步模式。 CANopen 协议和 CSP 运行模式。
11	供电电源	1	24VDC, 指示灯 PWR 指示供电状态。

脉冲输出的方向规格, 见表格 2:

表格 2 脉冲输出的方向

脉冲+方向 (Pls+Dir)	正方向	
	负方向	
双脉冲 (CW/CCW)	正方向	

	负方向	
90 度相位差 (A/B)	正方向	
	负方向	

***注意：** 指示灯为低电平点亮，与表格中的逻辑相反。

脉冲输入方向规格（光耦导通逻辑），见表格 3：

表格 3 脉冲输入的方向

脉冲+方向 (Pls+Dir)	正方向	
	负方向	
双脉冲 (CW/CCW)	正方向	

	负方向	
90 度相位差 (A/B,A/B4)	正方向	
	负方向	

表格 4 Z 向信号逻辑

Z 向信号	
-------	--

RUN 指示灯表示 ESM (EtherCAT State Machine) 的状态, 亮灯颜色为绿色。

表格 5 RUN 指示灯的功能定义

LED 状态	内容
关闭(OFF)	ESM:INIT 状态
闪烁(Blinking)	ESM:Pre-Operational 状态
单闪(Single flash)	ESM:Safe-Operational 状态
点亮(ON)	ESM: Operational 状态

ERR 指示灯表示 AL status code 定义的报警状态, 亮灯颜色为红色

表格 6 ERR 指示灯的功能定义

LED 状态	内容
关闭(OFF)	未发生报警

闪烁(Blinking)	通信设定异常
单闪(Single flash)	同步事件异常
双闪(Double flash)	看门狗超时
轻闪(Flickering)	初始化异常
点亮(ON)	PDI 异常

脉冲输出功能共用数字量输出的前 4 个端子 (Q0.0~Q0.3)，关系下图:

表格 7 脉冲输出与数字量输出端子关系表

脉冲输出	通道 0	通道 1
A	Q0.0	Q0.2
B	Q0.1	Q0.3

脉冲输入功能位于 DB15 接口，关系见下，具体引脚功能见硬件手册。

脉冲输入	通道 0	通道 1
A+	5	1
B+	4	2
Z+	3	-
A-	10	6
B-	9	7
Z-	8	-
+5V	13	12
GND	14	11

1.2 产品外观

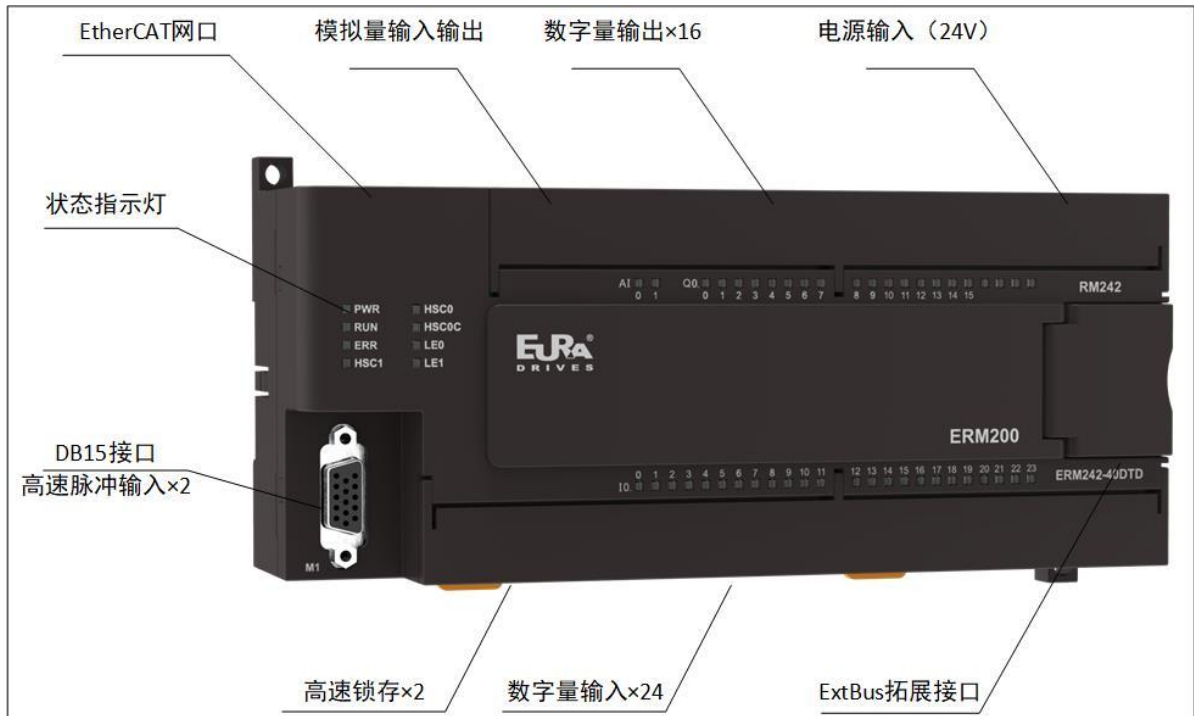


图 1 整体视图

1.3 产品尺寸

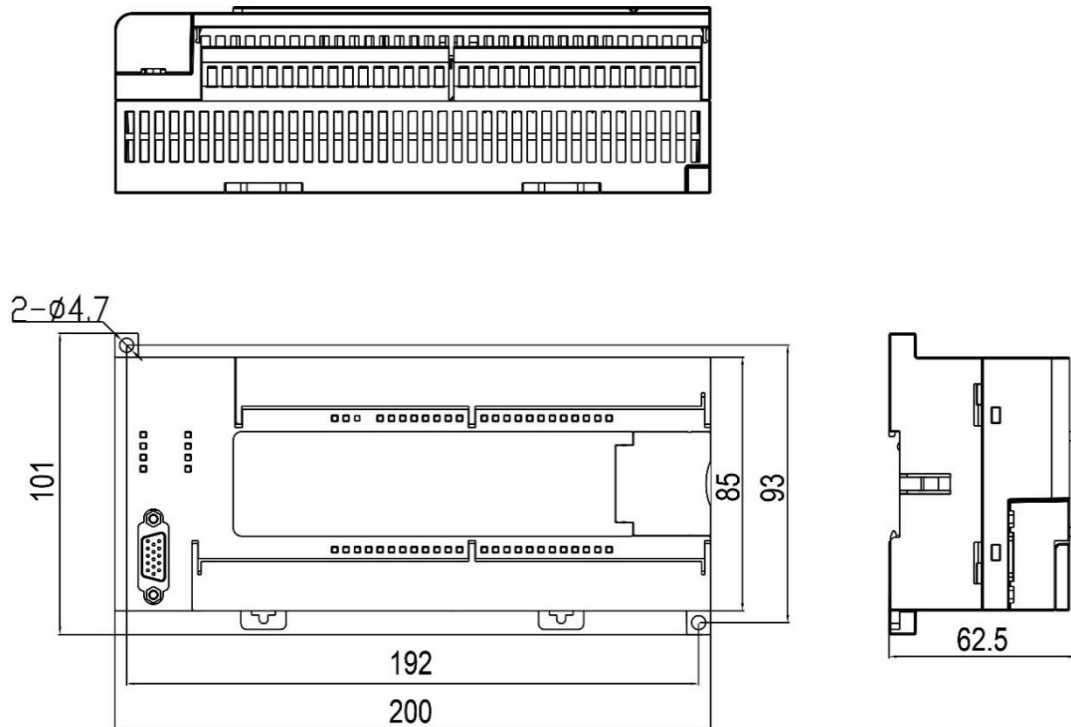


图 2 产品尺寸图 (单位: mm)



第二章 使用说明

本产品作为 EtherCAT 从站工作于 EtherCAT 网络中，接收 EtherCAT 主站下发的数据并反馈本模块采集的信息反馈至 EtherCAT 主站。这里以本公司的 EAC100/200 运动控制器为例，说明本产品具体的使用方法。EAC100/200 运动控制器内置高性能 EtherCAT 主站，支持 DC 同步模式，采用 CoDeSys 作为软件编程软件。

2.1 组态准备

1. 组态准备，导入 ERM242 的 ESI 文件（XML 文件格式）。
2. 打开 CoDeSys 软件。
3. 点击工具（Tools）-> 设备存储库（Device Respository）



图 3 打开设备安装界面

4. 点击 Install 按钮，弹出对话框。

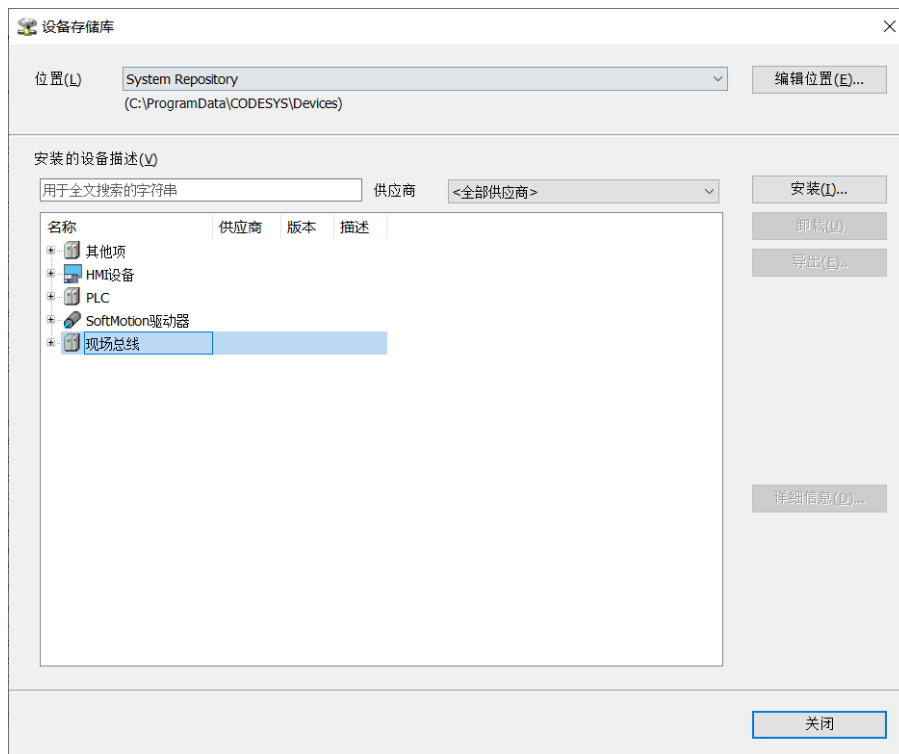


图 4 安装设备界面

5. 找到 ESI 文件（XML）所在文件夹，并选择该软件，点击打开按钮

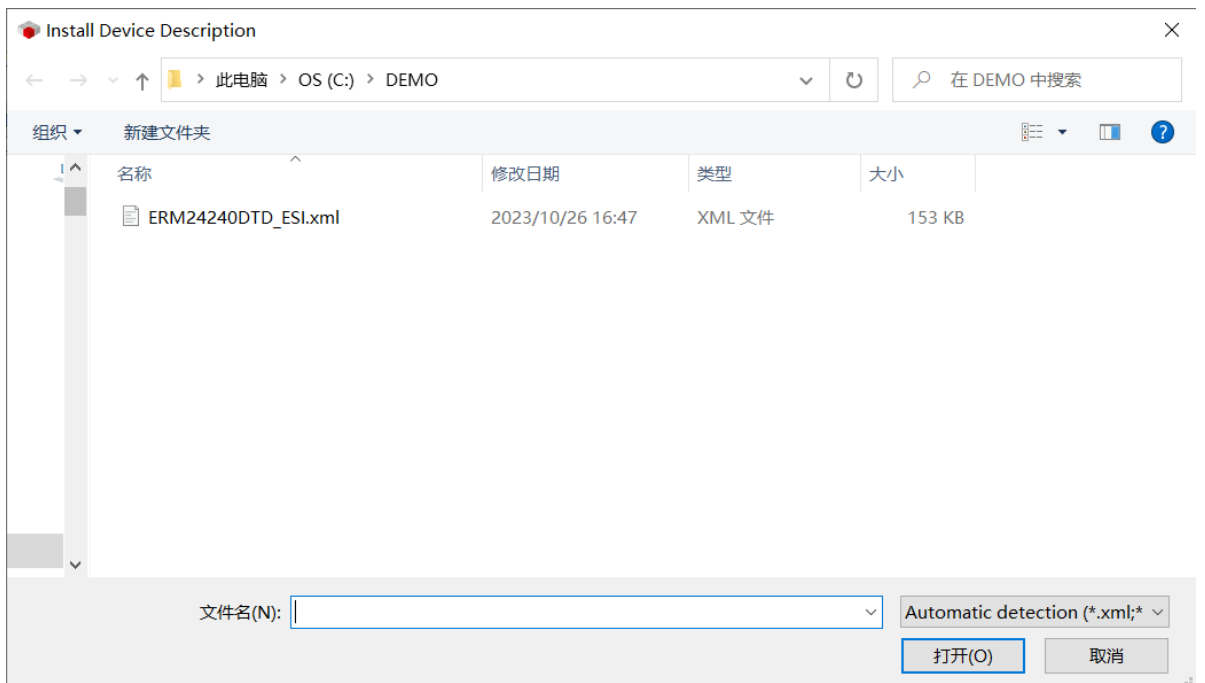


图 5 选择设备描述文件对话框

6. 成功安装 ESI 文件，见下图。

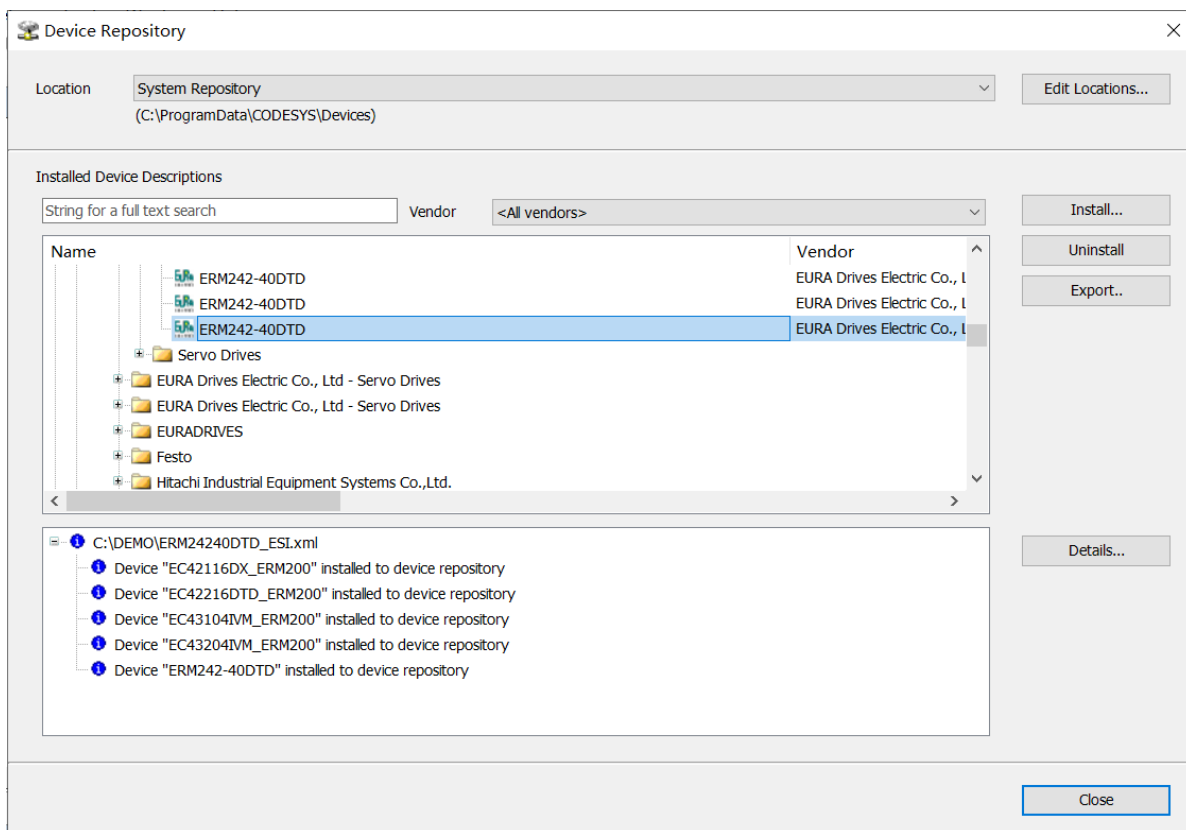


图 6 设备安装成功

2.2 设备组态

将 ERM242 作为 EtherCAT 从站添加到 CoDeSys 工程。

1. 新建 CoDeSys 工程，目标设备选择 EAC 系列运动控制器（参考 EAC100/200 运动控制器文档）。
2. 右击 Add Device。

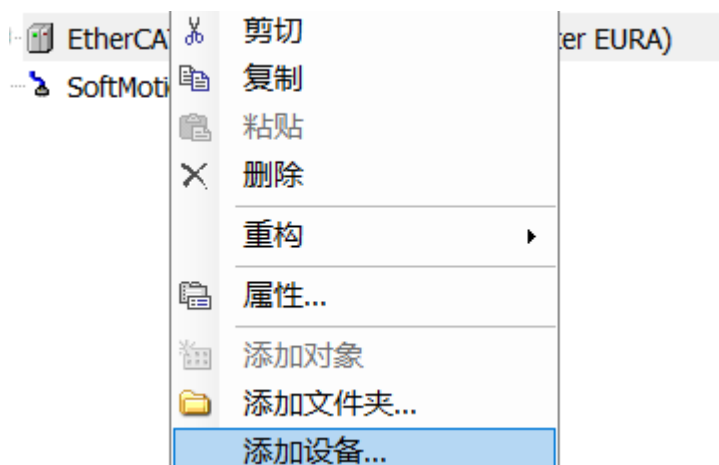


图 7 添加 EtherCAT 从站设备

3. 搜索 ERM 系列设备，选择 ERM242-40DTD，然后点击 Add Device。

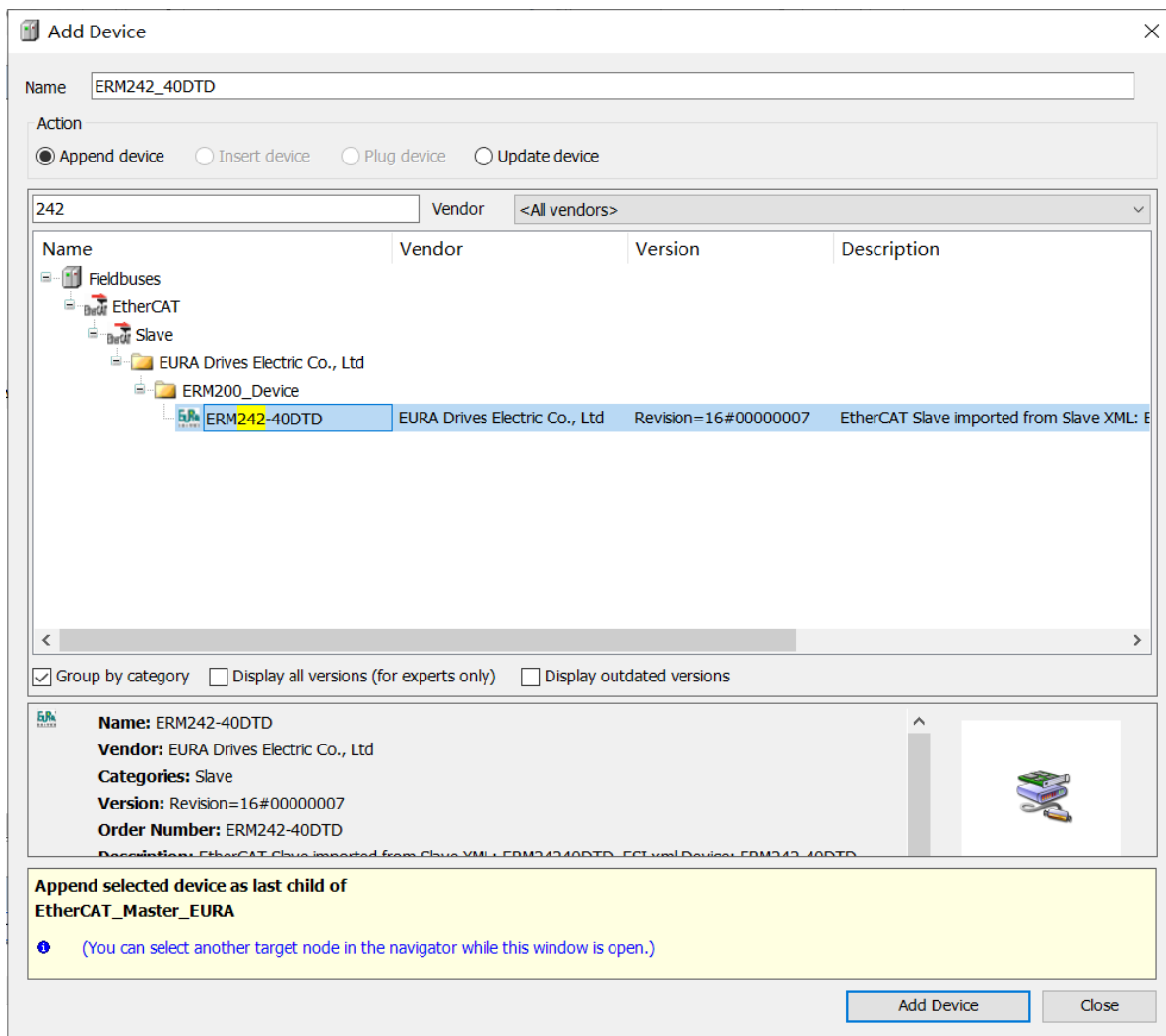


图 8 找到 EtherCAT 从站设备

4. 添加成功，如下图所示：

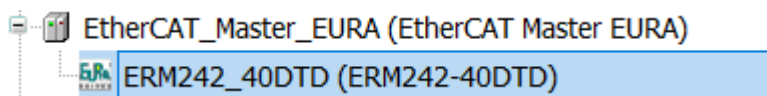


图 9 EtherCAT 从站设备添加到工程

2.3 数据映射

建立控制程序和 ERM242 设备之间的数据映射。

1. 创建变量。在程序中创建用于映射的变量，如图所示：

```

1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3      xArrOut:ARRAY[0..15] OF BOOL;
4      xArrIn:ARRAY[0..23] OF BOOL;
5      xValSetDO:bool;
6      wAO_0:WORD;
7      wAO_1:WORD;
8      wAI_0:WORD;
9      wAI_1:WORD;
10
11     wAO_2:WORD;
12     wAO_3:WORD;
13     wAI_2:WORD;
14     wAI_3:WORD;
15
16     wCfgAI:UDINT:=16#30303030;
17     wCfgAO:UDINT:=16#30303030;
18
19     diHsc0_Val:DINT:=0;
20     diHsc1_Val:DINT:=0;
21     diHsc0_Val_Set:DINT:=0;
22     diHsc1_Val_Set:DINT:=0;
23     dwCtrlHsc:DWORD;
24     dwCtrlLatch:DWORD:=16#00000000;
25
26     dwStatusLatch:DWORD;
27     diHsc0_Latch_0:DINT;
28     diHsc0_Latch_1:DINT;
29     diHsc1_Latch_0:DINT;
30     diHsc1_Latch_1:DINT;    // not support
31
32     xStartPrg:BOOL:=FALSE;
33     xStartPrg_old:BOOL:=FALSE;
34     xInited:bool:=false;

```

图 10 创建 PLC 映射变量

2. 建立映射。双击 ERM242-40DTD 打开设备标签页，如下图：

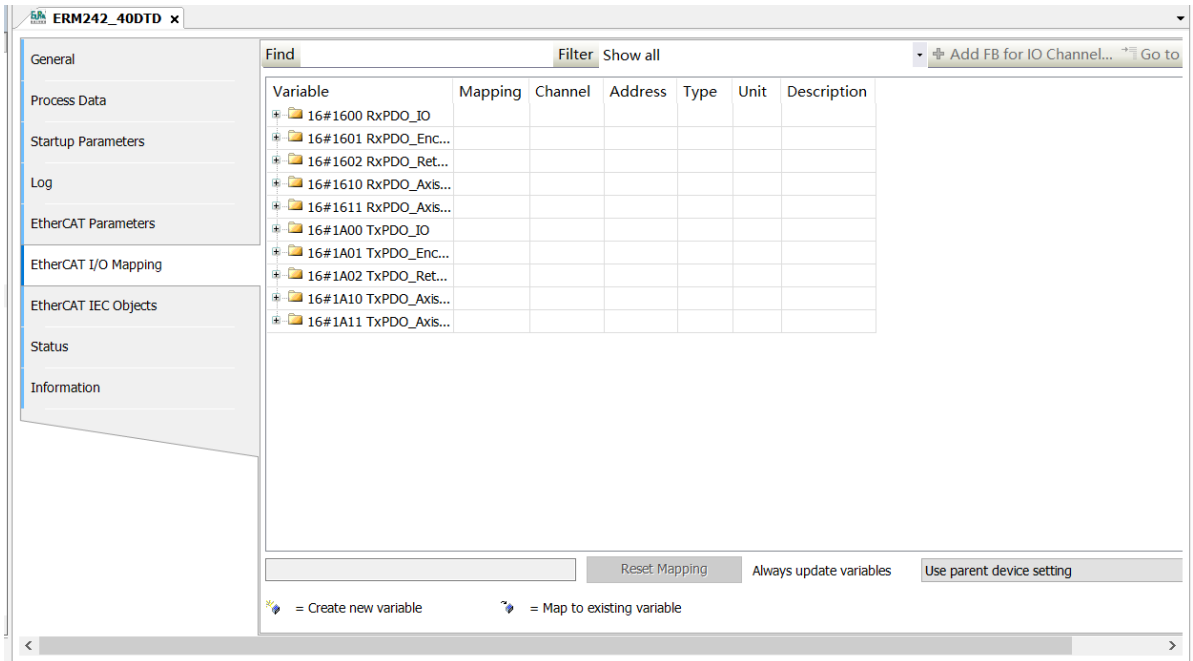


图 11 EtherCAT 从站 I/O 映射标签页

3. 成功建立映射，如下图：

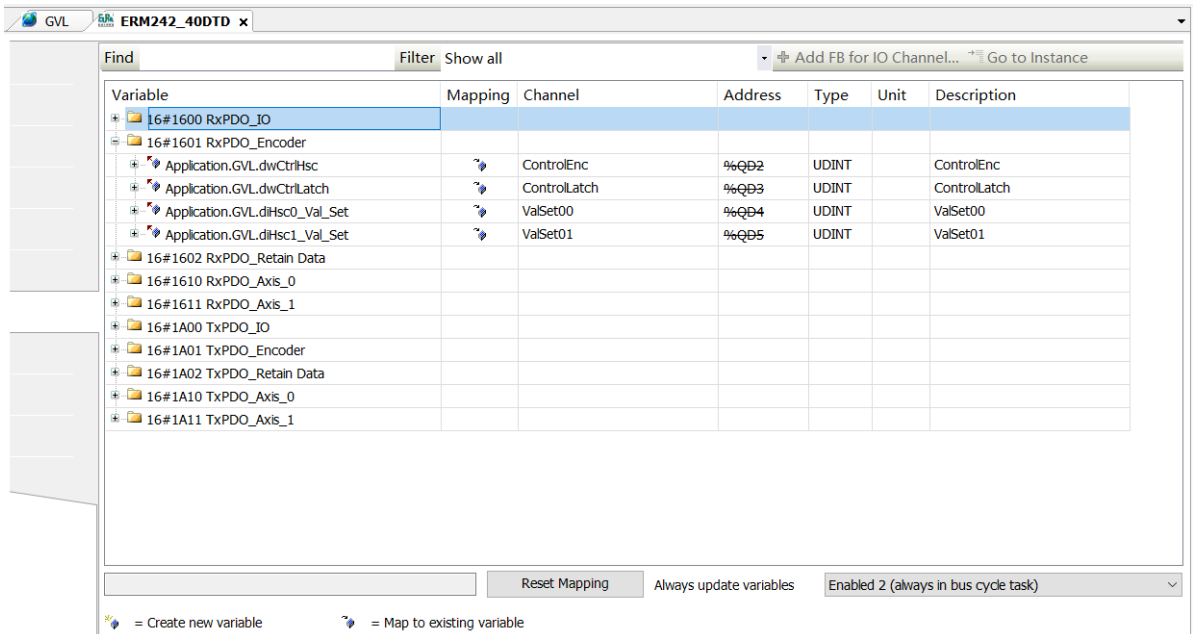


图 12 EtherCAT 从站 I/O 与 PLC 变量建立映射

2.4 编程调试

在建立数据映射之后，通过软件操作映射的变量即可操作 ERM242 的控制资源。

1. 编写代码。

```

6      uiAI_1:UINT;
7      uiAO_0:UINT;
8      uiAO_1:UINT;
9      END_VAR
10
11  VAR
12      udiCntTask:UDINT:=0;
13      udiBitOffset:UDINT:=0;
14      udiValSet0:UDINT:=0;
15      udiValGet0:UDINT:=0;
16      uiValSet_1:UINT:=0;
17      uiValSet_2:UINT:=0;
18      uiValGet_1:UINT:=0;

1  udiDO := udiValSet0;
2  udiValGet0 := udiDI;
3  udiCntTask := udiCntTask + 1;
4  IF udiCntTask >= 200 THEN
5      udiCntTask := 0;
6      udiValSet0 := SHL(1,udiBitOffset);
7      udiBitOffset := udiBitOffset + 1;
8      IF udiBitOffset >= 16 THEN
9          udiBitOffset := 0;
10     END_IF
11 END_IF

13 uiAO_0 := uiValSet_1;
14 uiAO_1 := uiValSet_2;
15
16 uiValGet_1 := uiAI_0;
17 uiValGet_2 := uiAI_1;

```

图 13 编写 PLC 代码

2. 编译代码，并到登录到目标设备（EAC100/200 运动控制器），下载工程并在线监控。该实例代码可以实现数字量输出信号被依次点亮。

Device.Application.PLC_PRG		
Expression	Type	Value
◆ uiAO_0	UINT	0
◆ uiAO_1	UINT	0
◆ udiCntTask	UDINT	25
◆ udiBitOffset	UDINT	13
◆ udiValSet0	UDINT	4096
◆ udiValGet0	UDINT	0
◆ uiValSet_1	UINT	0
◆ uiValSet_2	UINT	0
◆ uiValGet_1	UINT	11
◆ uiValGet_2	UINT	10


```

1  udiDO 4096 := udiValSet0 4096 ;
2  udiValGet0 0 := udiDI 0 ;
3  udiCntTask 25 := udiCntTask 25 + 1;
4  IF udiCntTask 25 >= 200 THEN
5      udiCntTask 25 := 0;
6      udiValSet0 4096 := SHL(1,udiBitOffset 13 );
7      udiBitOffset 13 := udiBitOffset 13 + 1;
8      IF udiBitOffset 13 >= 16 THEN
9          udiBitOffset 13 := 0;
10     END_IF
11 END_IF

13 uiAO_0 0 := uiValSet_1 0 ;
14 uiAO_1 0 := uiValSet_2 0 ;
15
16 uiValGet_1 11 := uiAI_0 11 ;
17 uiValGet_2 10 := uiAI_1 10 ;
18 RETURN

```

图 14 在线调试监控

2.5 参数配置

在设备的 Startup Parameters 标签页下，点击 Add 可以添加初始化参数。

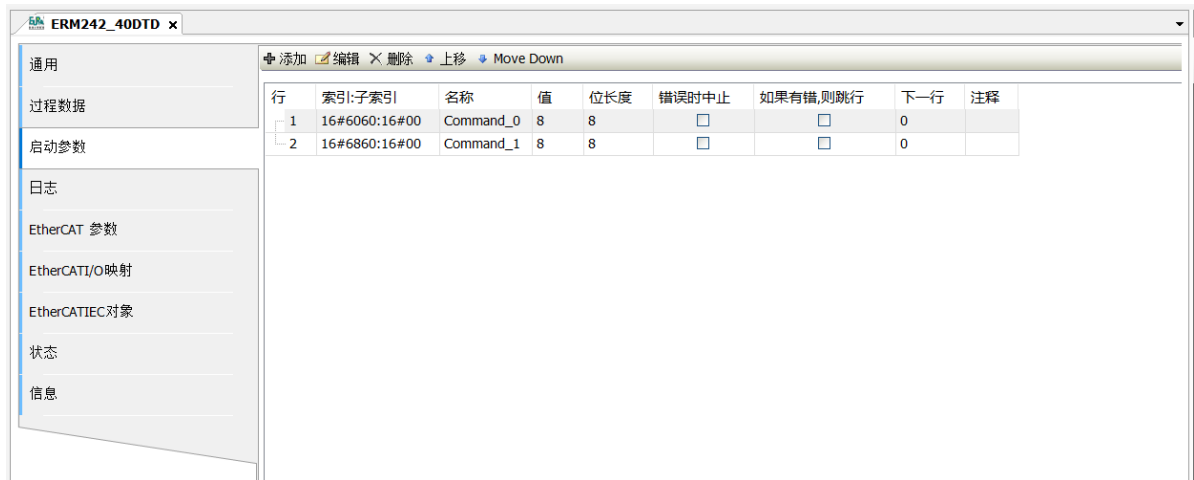


图 15 EtherCAT 从站参数初始化标签页

1. 配置数字量输入滤波:

一组 16 位的 DI 滤波值，包括 4 个 DI 信号，每个 DI 滤波数据占用 4 位数据，默认滤波值为 3ms，即一组 DI 滤波的数值为 $0x3333 = 13107$ 。

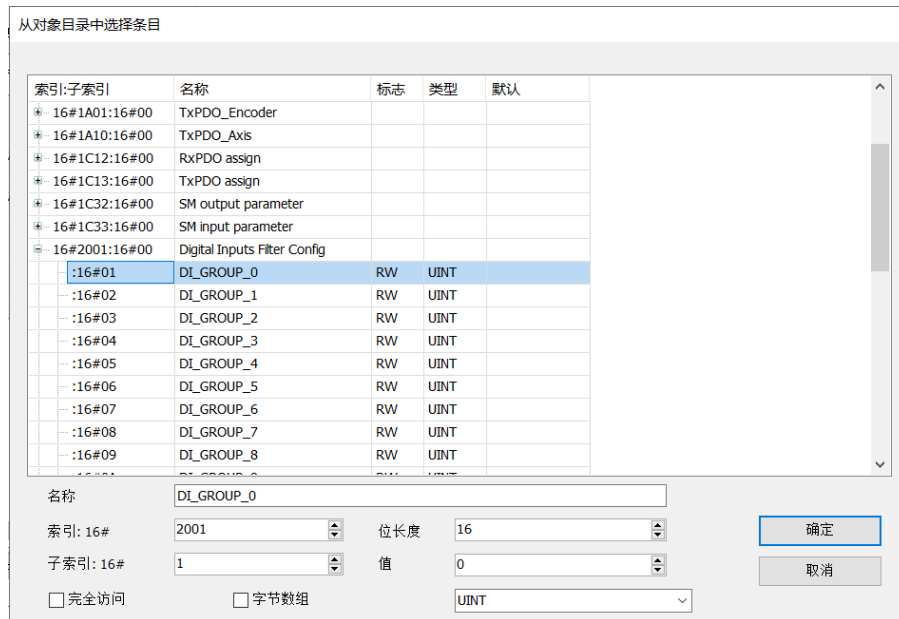


图 16 DI 滤波参数配置 (1)

设置 DI0 的滤波值为 5ms，那么需要写入数据为 0x3335 (16#3335)，使用 Windows 计算器将 0x3335 转换为 10 进制数据，写入到 Value 中，见下图。

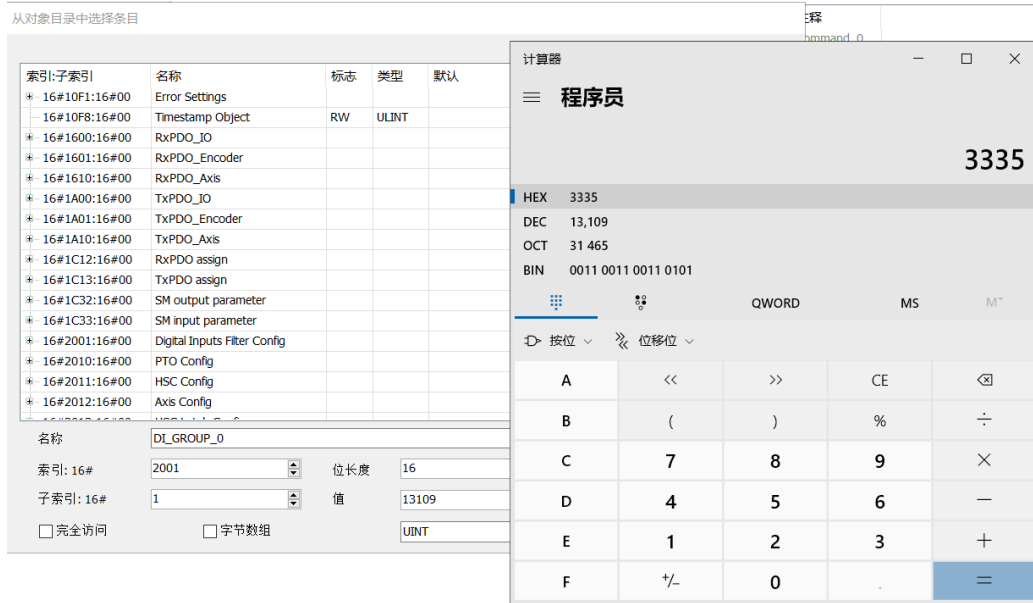


图 17 DI 滤波参数配置 (2)

2. 配置 PTO 输出形式:

以配置 PTO 的第 1 通道为脉冲+方向的模式为例，找到 0x2010 子索引 01，将 Value 值改为 1，然后点击 OK 按钮。

注意 1: 参数值 1 表示脉冲+方向模式，其他数值定义参考 EtherCAT 数据对象列表。

注意 2: 开启 PTO 之后，与 PTO 公用的 DO 端子将不再作为 DO 使用。

注意 3: 使用 PTO 功能时，ERM242 模块必须运行于 DC 同步模式。

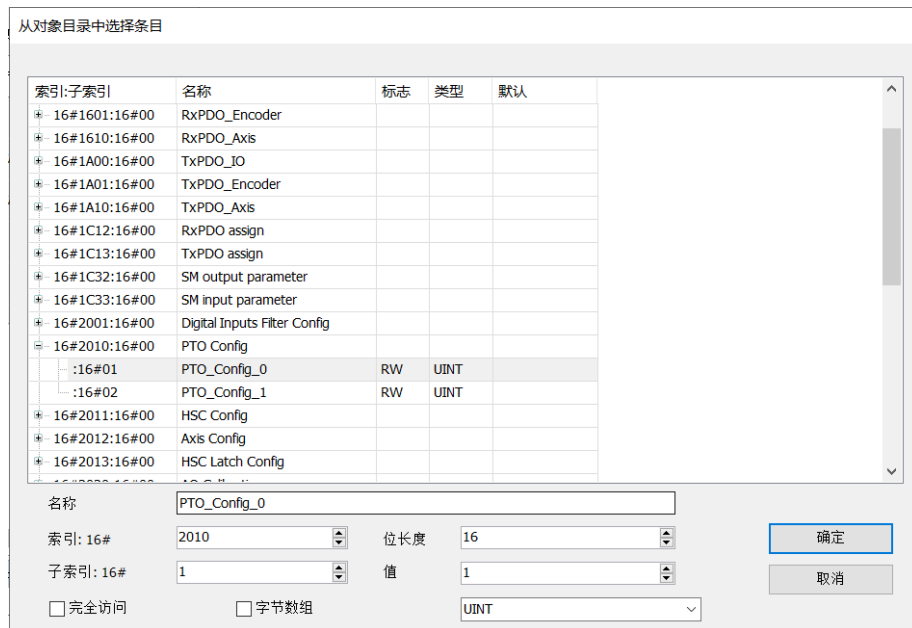


图 18 PTO 参数配置

3. 配置 HSC 输入形式:

以配置 HSC 的第 1 通道为脉冲+方向的模式为例，找到 0x2011 子索引 02 (HSC_Config_0)，将 Value 值改为 1，然后点击 OK 按钮。

注意 1：参数值 1 表示脉冲+方向模式，其他数值定义参考 EtherCAT 数据对象列表。

注意 2：如果不配置该参数，那么 HSC 默认也会采用脉冲+方向模式。

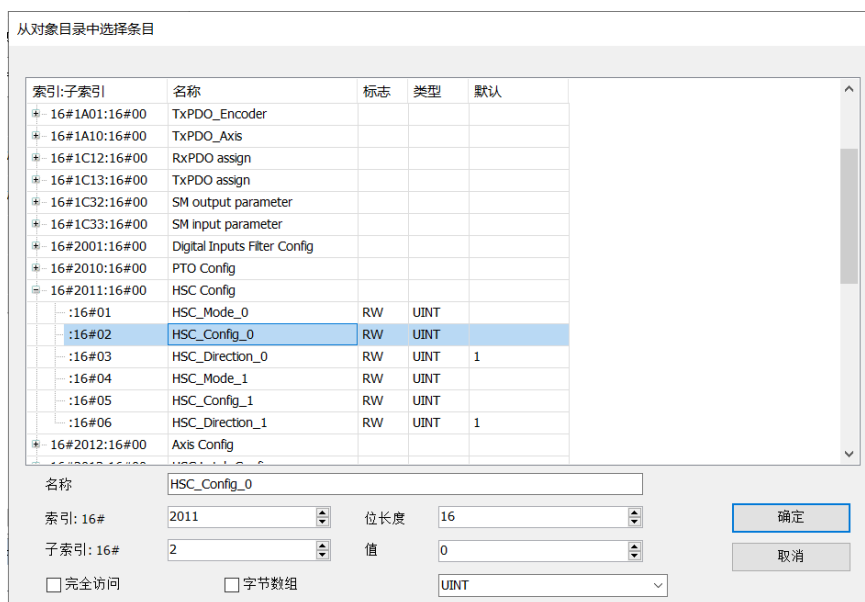


图 19 HSC 参数配置

4. 配置 HSC 锁存滤波参数：

根据需求，配置 HSC 锁存引脚的滤波参数，具体参数定义参照自定义 EtherCAT 数据对象 0x2013。

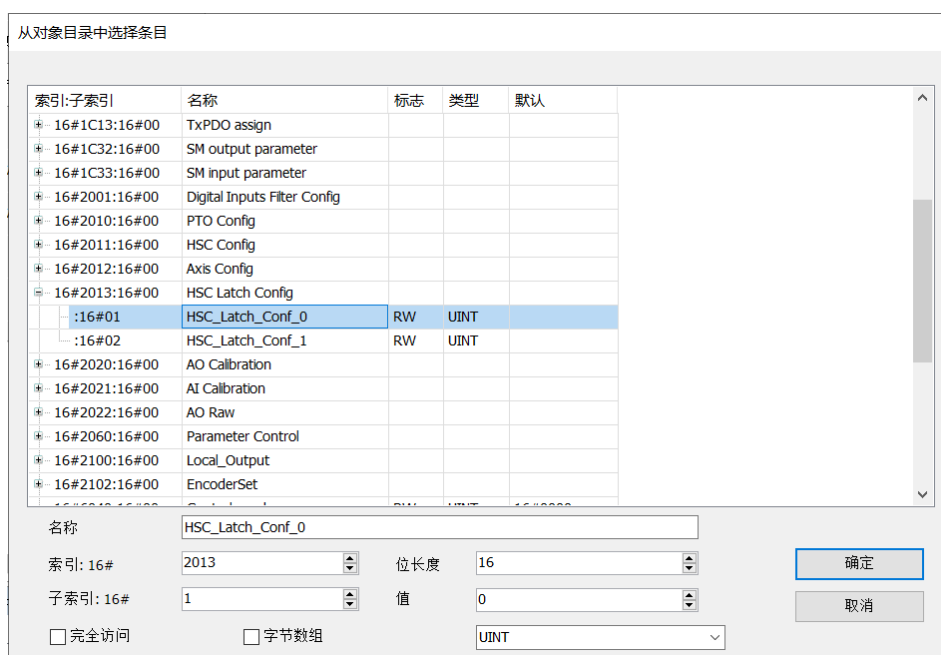


图 20 HSC 锁存参数配置

2.6 HSC 功能使用

1. HSC 配置参数:

本产品的 HSC 接口支持多种编码器输入信号（差分信号），在使用时首先需要配置编码器信号方式，参照 2.5 节

提示：处于脉冲+方向模式时，需将 HSC 配置为负逻辑（HSC_Direction = 0），使其与 PTO 的逻辑一致。

提示：ERM242 的脉冲输入信号为差分信号，而 PTO 输出信号为单端信号，不能将两者直接连接。

2. HSC 计数值读取

建立数据变量，用于存放采集的编码器数值，如图 21。

```

3   xArrOut:ARRAY[0..15] OF BOOL;
4   xArrIn:ARRAY[0..23] OF BOOL;
5   xValSetDO:bool;
6   wAO_0:WORD;
7   wAO_1:WORD;
8   wAI_0:WORD;
9   wAI_1:WORD;
10
11  wAO_2:WORD;
12  wAO_3:WORD;
13  wAI_2:WORD;
14  wAI_3:WORD;
15
16  wCfgAI:UDINT:=16#30303030;
17  wCfgAO:UDINT:=16#30303030;
18
19  diHsc0_Val:DINT:=0;
20  diHsc1_Val:DINT:=0;
21  diHsc0_Val_Set:DINT:=0;
22  diHsc1_Val_Set:DINT:=0;
23  dwCtrlHsc:DWORD;
24  dwCtrlLatch:DWORD:=16#00000000;
25
26  dwStatusLatch:DWORD;
27  diHsc0_Latch_0:DINT;
28  diHsc0_Latch_1:DINT;
29  diHsc1_Latch_0:DINT;
30  diHsc1_Latch_1:DINT;    // not support
31
32  xStartPrg:BOOL:=FALSE;
33  xStartPrg_old:BOOL:=FALSE;
34  xInited:bool:=false;
35  END_VAR

```

图 21 编码器计数变量区

将数据变量映射到设备中，用于存放采集编码器的数值，如图 22。

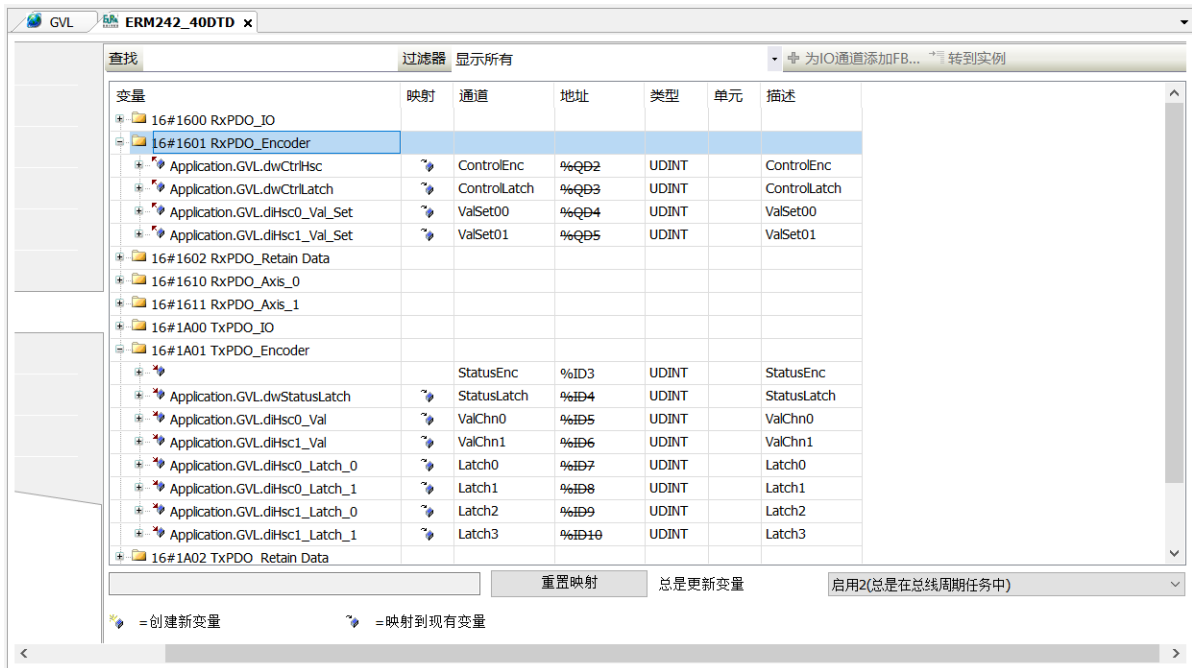


图 22 编码器计数的变量映射

在程序中编写代码，访问采集到的编码器数值，如图 23。

```

1  PROGRAM PLC_PRG
2  VAR
3      iCntDelay:INT:=0;
4      iCntDO:INT:=0;
5      iTemp:INT:=0;
6      xValSet:BOOL;
7      xArrIn_Read:ARRAY[0..23] OF BOOL;
8      wValSetAO_0:WORD:=1000;
9      wValSetAO_1:WORD:=2000;
10
44  wValGetAI_3 := gvl.wAI_3;
45
46  // hsc test
47  GVL.dwCtrlHsc := dwCtrlSet_Hsc; //reset hsc1:16#0100
48
49  gvl.diHsc0_Val_Set := diValSet_Hsc0;
50  gvl.diHsc1_Val_Set := diValSet_Hsc1;
51
52  diValHsc0_Read := gvl.diHsc0_Val;
53  diValHsc1_Read := gvl.diHsc1_Val;
54
55  // latch test
56
57  gvl.dwCtrlLatch := dwCtrlSet_Latch;
58  dwStatusGet_Latch := GVL.dwStatusLatch;
59
60  diValHsc0_Latch0 := gvl.diHsc0_Latch_0;
61  diValHsc0_Latch1 := gvl.diHsc0_Latch_1;
62  diValHsc1_Latch0 := gvl.diHsc1_Latch_0;
63
64  gvl_dio.xDO_Latch_0 := xDO_LatchSet_0;
65  gvl_dio.xDO_Latch_1 := xDO_LatchSet_1;
66  gvl_dio.xDO_Latch_2 := xDO_LatchSet_2;
67

```

图 23 在程序中访问 HSC 计数值

调试运行程序，通过跟踪(Trace)功能监控编码器数值的变化，如图 24。

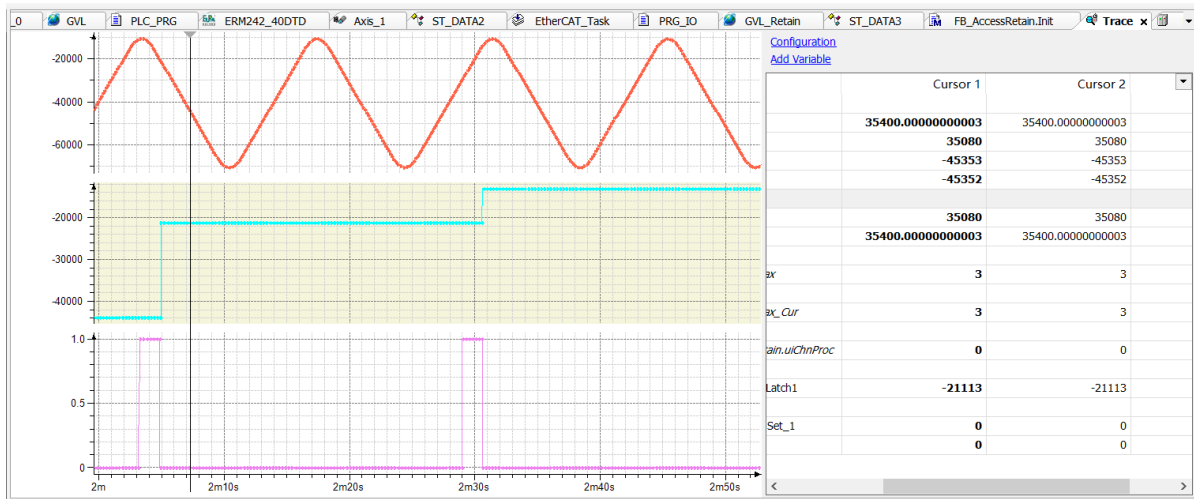


图 24 跟踪编码器计数值变化过程

3. HSC 计数值设置

通过设置 0x2102.1 和监控 0x2103.1，可以将 HSC 采集编码器数值复位或者设置为指定的数值。

以 HSC0 连接的编码器为例，将 0x2102.1 中数据由 0 切换为 1，则采集的编码器数值被清零；将 0x2102.1 的数据切换为 0x03，则将 0x2102.3 的数据写入到 HSC0 采集的编码器计数值中。

4. HSC 锁存使用

脉冲输入锁存功能，每个脉冲输入接口在输入端子上各有 1 个锁存输入，其中脉冲数据接口 0 的 Z 向也可以作为输入锁存信号使用。

在正确配置锁存功能之后，锁存触发信号才能将脉冲数值锁存，关联数据对象如 0x2102.1 和 0x2103.1, 0x2103.4~7。

锁存方式包括单次触发和连续触发等两种模式，区别如下：

单次触发：配置为单次触发模式之后，仅采集配置之后的第一次锁存触发信号的 HSC 计数值，如果需要再次锁存 HSC 计数值，需要重新配置触发模式。

连续触发：配置为连续触发模式之后，在每次触发信号到达时，都会锁存 HSC 计数值并更新到数据对象中。

以锁存信号 0 为例，配置其单次触发模式，采用上升沿触发。

向锁存控制字（0x2102.2）写入 0x00000011(bit0=1/bit1=0/bit5=1)，然后读取锁存状态字（0x2103.2），检查其 bit0 是否为 1（开启状态）。在检测到锁存信号时，锁存状态字（0x2103.2）的 bit1 变为 1，同时将锁存的 HSC 数值存放到 Latch0（0x2103.5）中。

2.7 CiA 402 轴与脉冲输出使用

本节介绍使用 CiA 402 通过脉冲输出接口（PTO），控制伺服驱动器。

1. 配置参数：

开启 PTO 输出形式，如 PTO 配置为 1（脉冲+方向模式）。

2. 添加 CiA 402 轴，见下图。

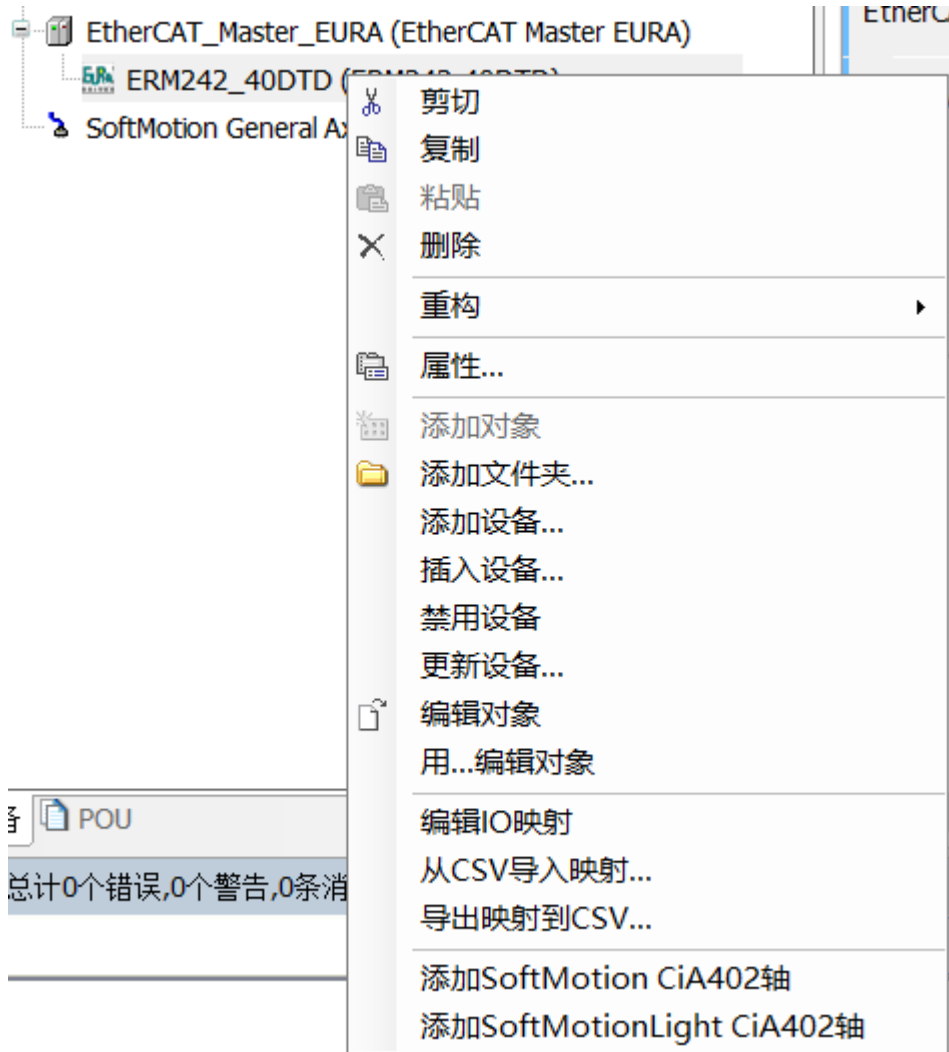


图 25 添加 CiA 402 轴

建议：为新建的 CiA402 轴取 1 个简单易懂的名称，如 Axis_A，修改缩放比例为 1:1，见下图。



图 26 配置 CiA 402 轴的缩放比例

3. 配置 DC 运行模式，见下图。

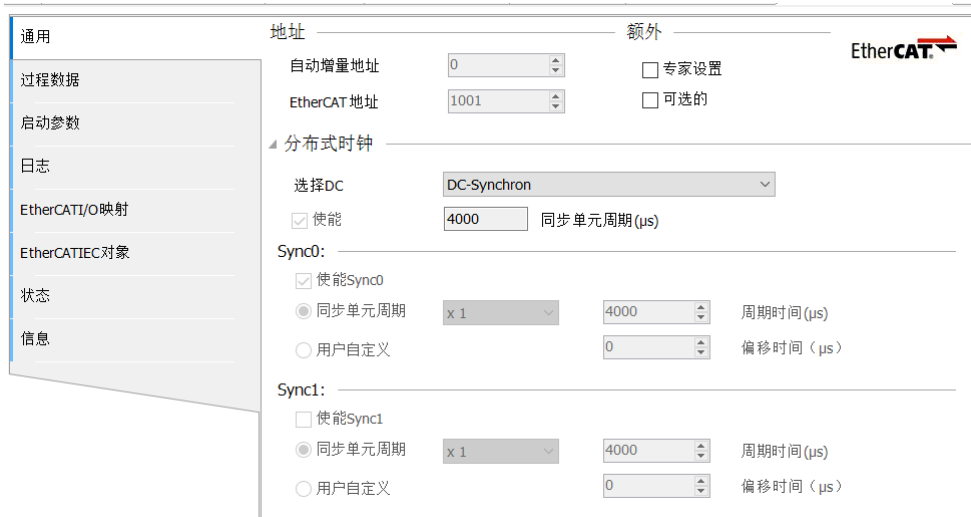


图 27 配置 DC 模式

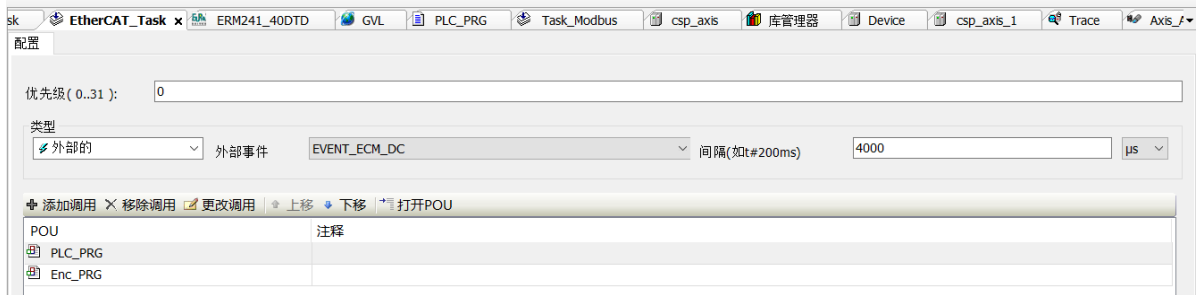


图 28 配置 EtherCAT 通信周期 (DC 模式)

4. 编写代码，使用以下功能块：
 - (1) MC_Power，控制 CiA 402 轴使能。
 - (2) MC_MoveRelative，使 CiA 402 轴执行相对运动。
5. 观察实际现象。

2.8 ExtBus 拓展总线的使用

1. 添加模块

右击 ERM242-40DTD 从站设备，点击“添加设备”，如图 29 所示。

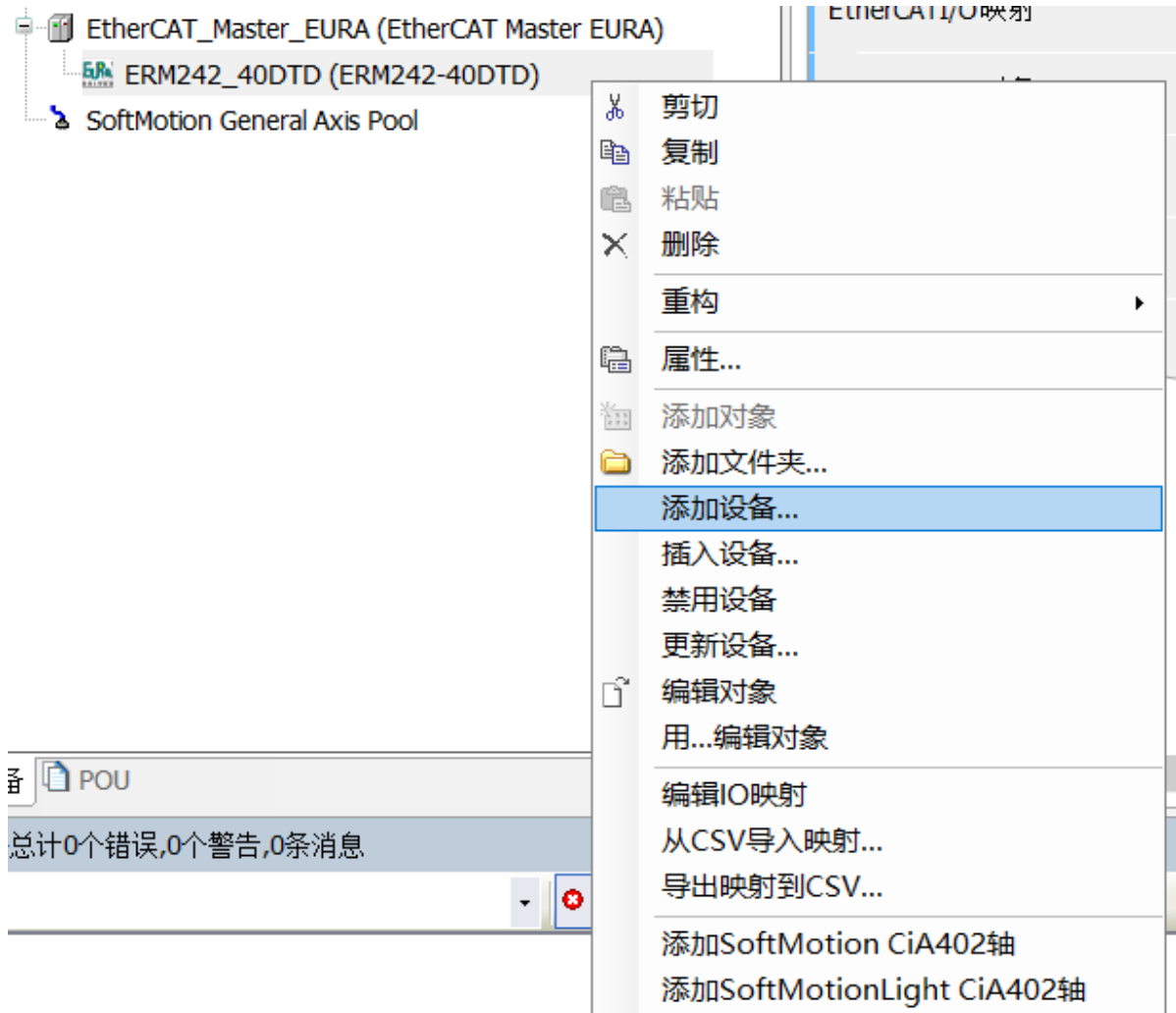


图 29 ERM242-40DTD 添加 ExtBus 模块

从弹出的设备列表对话框中，根据 ExtBus 模块类型和序号，选择相关的 ExtBus 模块，如图 30 所示。

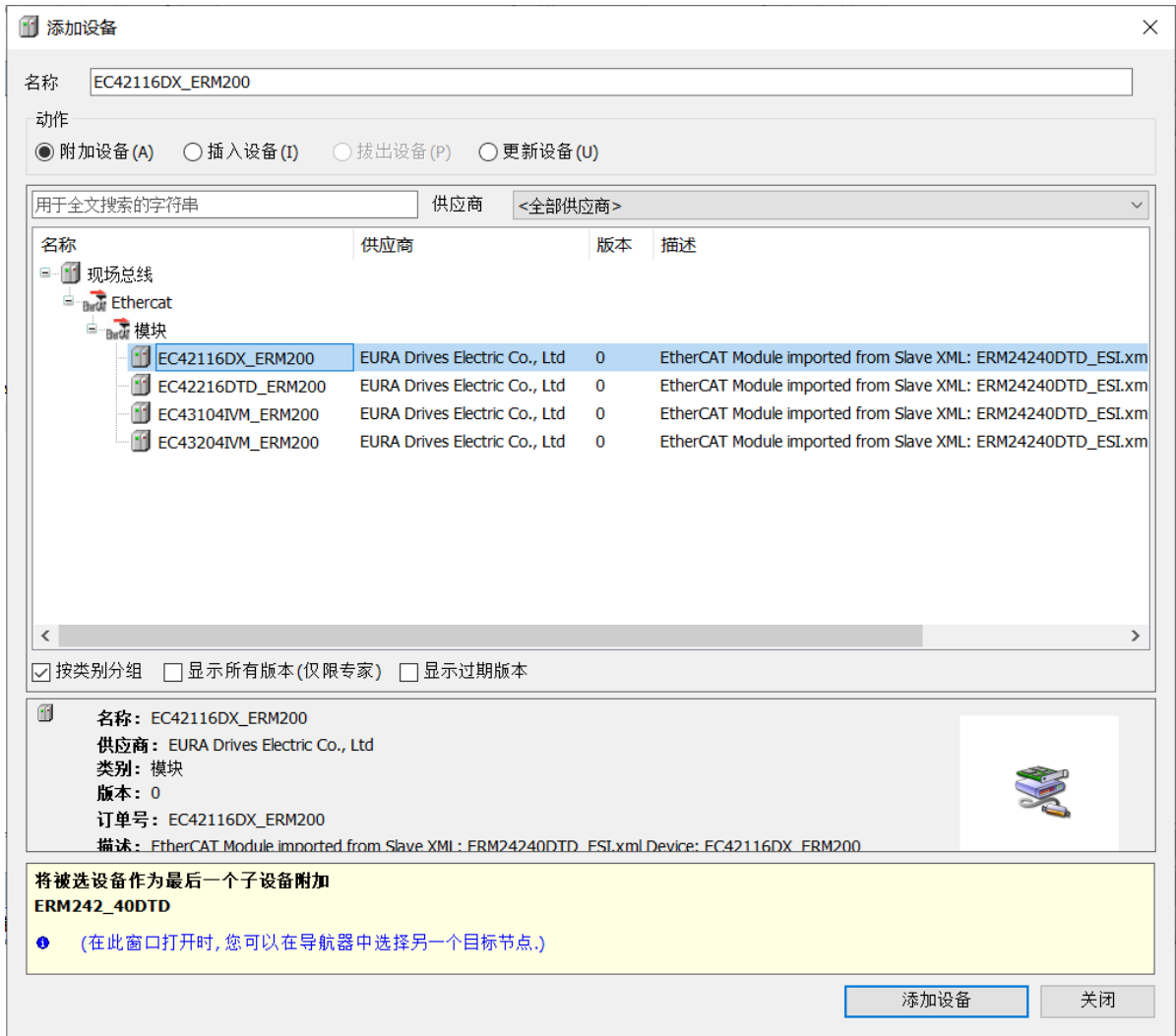


图 30 ExtBus 模块的设备列表

添加到工程之后的 ExtBus 模块, 如图 31 所示。

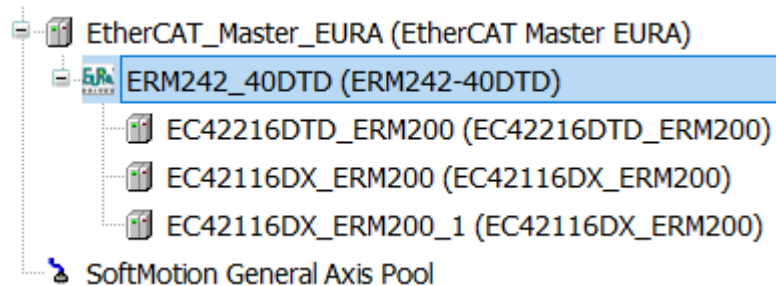


图 31 添加到工程中的 ExtBus 模块

2. 配置映射

打开新添加 ExtBus 模块, 如图 32 所示, 需要建立映射。

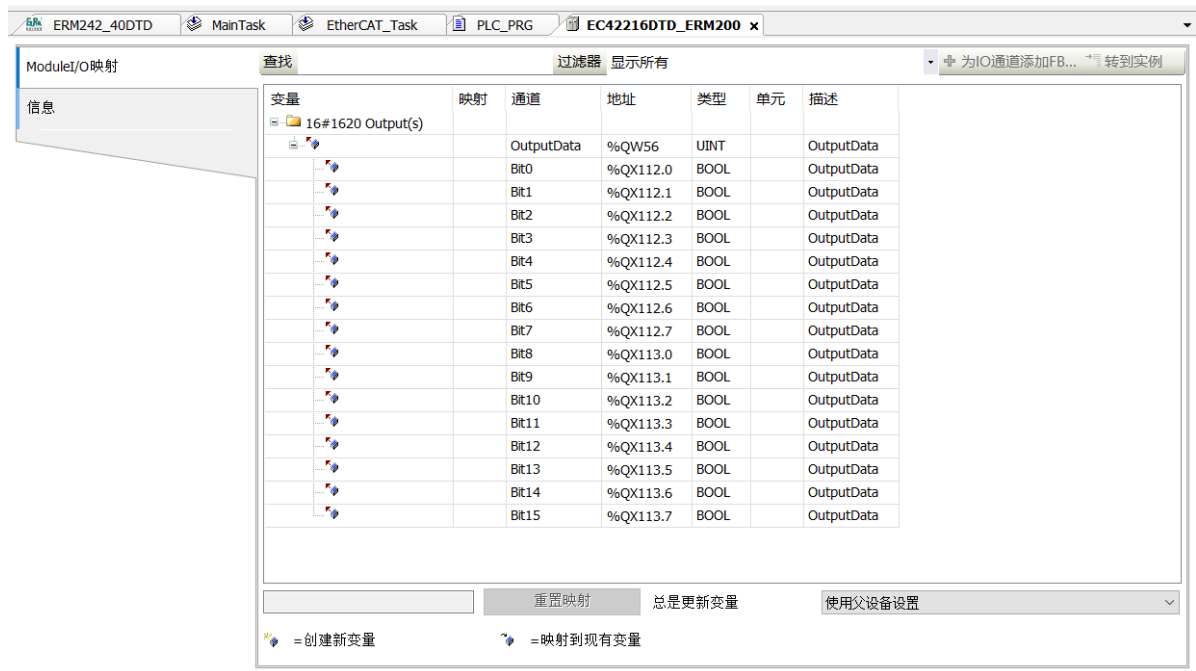


图 32 ExtBus 模块数据映射（空）

编写代码，建立数据映射，如图 33 所示。

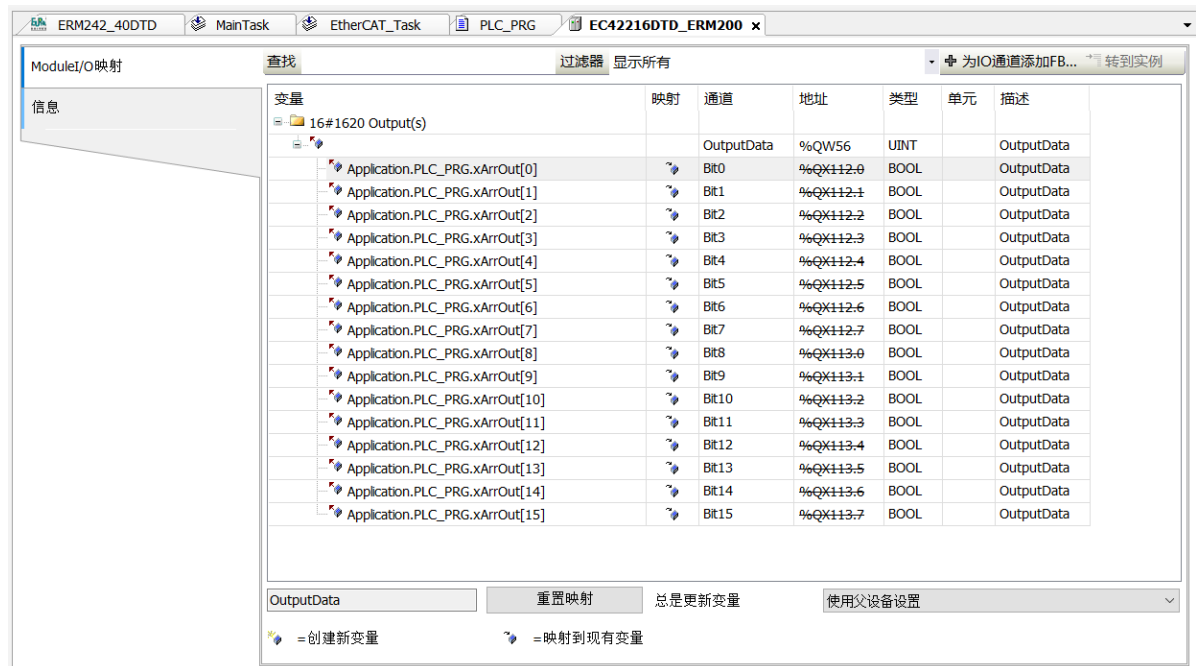


图 33 ExtBus 模块数据映射（已填充）

3. 编程访问

编写工程代码，如图 34 所示，然后编译之后在线调试，如图 35 和图 36 所示，至此完成 ExtBus 模块的调试使用，可以观察 EC422 模块的 DO 输出。

```

ERM242_40DTD  MainTask  EtherCAT_Task  PLC_PRG x  EC42216DTD_ERM200
1  PROGRAM PLC_PRG
2  VAR
3      xArrOut:ARRAY[0..15] OF BOOL;
4      xArrIn:ARRAY[0..23] OF BOOL;
5      iCntDelay:INT:=0;
6      iCntDO:INT:=0;
7      iTemp:INT:=0;
8      xValSet:BOOL;
9  END_VAR

10
11 iCntDelay := iCntDelay + 1;
12 IF iCntDelay > 200 THEN
13     iCntDelay := 0;
14
15
16 FOR iTemp:=0 TO 15 DO
17     xArrOut[iTemp] := FALSE;
18 END_FOR
19
20 xArrOut[iCntDO] := TRUE;
21
22 iCntDO:=iCntDO+1;
23 IF iCntDO > 15 THEN
24     iCntDO := 0;
25 END_IF
26
27 IF xValSet THEN
28     xValSet := FALSE;
29 ELSE
30     xValSet := TRUE;
31 END_IF
32
33 END_IF

```

图 34 代码实例

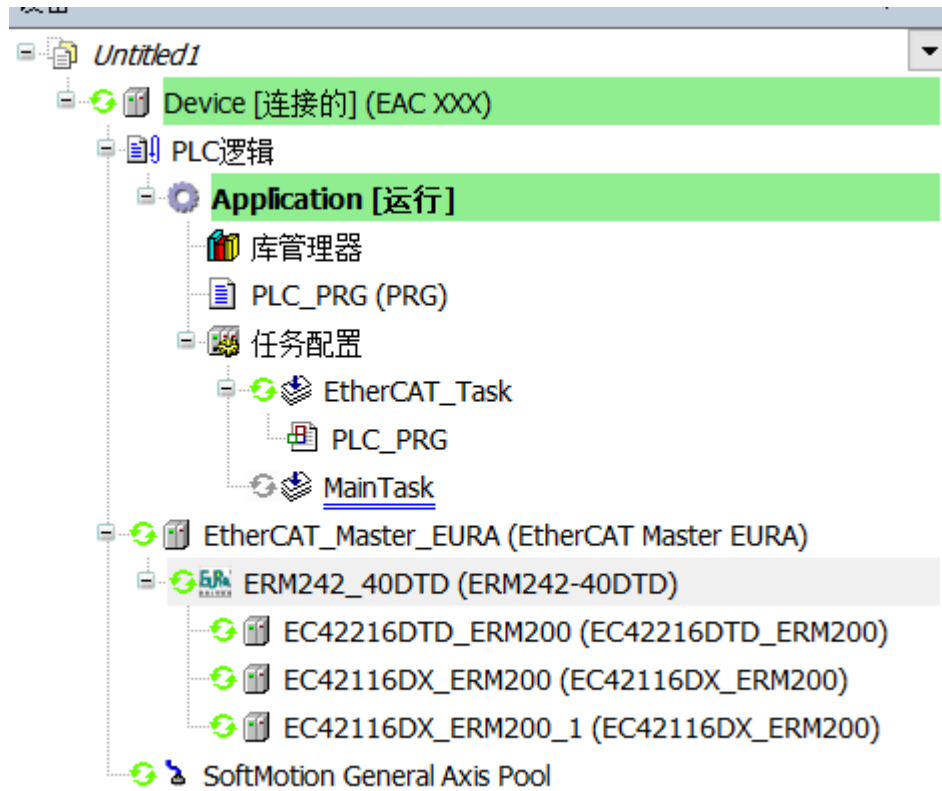


图 35 设备树（在线调试）

ERM242_40DTD MainTask EtherCAT_Task PLC_PRG x EC42216DTD_ERM200 EtherCAT_Master_EURA

Device.Application.PLC_PRG

表达式	类型	值
* xArrOut	ARRAY [0..15] ...	
* xArrIn	ARRAY [0..23] ...	
iCntDelay	INT	81
iCntDO	INT	11
iTemp	INT	16
xValSet	BOOL	TRUE

```

1 iCntDelay[81] := iCntDelay[81] + 1;
2 IF iCntDelay[81] > 200 THEN
3   iCntDelay[81] := 0;
4
5   FOR iTemp[16] := 0 TO 15 DO
6     xArrOut[iTemp[16], ???] := FALSE;
7   END_FOR
8
9   xArrOut[iCntDO[11], FALSE] := TRUE;
10
11   iCntDO[11] := iCntDO[11] + 1;
12 IF iCntDO[11] > 15 THEN
13   iCntDO[11] := 0;
14 END_IF
15
16 IF xValSet[TRUE] THEN
17   xValSet[TRUE] := FALSE;
18 ELSE
19   xValSet[TRUE] := TRUE;
20 END_IF
21
22 END_IF[RETURN]

```

图 36 代码实例（在线调试）

2.9 数据掉电保持的使用

ERM242-40DTD 的数据掉电保持功能可以保存 192 个字节 (64×3), 分为 3 个数据区, 每个区为 64 个字节。PDO 一次可以写入 1 个数据区。

数据对象 0x2104~0x2107 用于操作数据保持区:

0x2104.1 : Control word, 用于控制数据写入。

0: 空, 1: 执行写入, 2: 控制状态复位。

每个数据区占用 2 位数据。

数据区 0: bit0~1。

数据区 1: Bit2~3。

数据区 2: Bit4~5。

数据 ID: Bit14~15 (最高 2 位)。

0x2105.1: Status word, 用于检测写入状态。

0: 空, 1: 正在写入, 2: 写入完成。

数据区 0: bit0~1。

数据区 1: Bit2~3。

数据区 2: Bit4~5。

数据 ID: Bit14~15 (最高 2 位)。

0x2106: 写入缓冲区。

0x2107: 从 ERM242-40DTD 读取到的数据, 通过 SDO 方式读取。

实例 1: 读取设备数据

使用 SDO 方式, 读取 0x2107 的全部成员。

实例 2: 向数据区 0 写入数据:

1. 向 0x2106 写入数据。
2. 控制 0x2104.1 Control word, 由 0x0000 切换为 0x0001, 此时将 0x2106 中的数据写入至数据区 0
3. 检测 0x2105.1 Status word, 直到数据切换为 0x0002, 表示数据写入完成
4. 将 0x2104.1 Control word, 切换为 0x0002, 使 0x2105.1 Status word 复位至 0x0000
5. 等待 0x2105.1 Status word 切换为 0 时, 将 0x2104.1 Control word 复位至 0

实例 3: 写入新的 ID, 以清空数据

1. 向 0x2104.2 Data ID 写入新的 ID 数值, 与通过 0x2105.2 读取的数据不同。

2. 控制 0x2104.1 Control word, 由 0x0000 切换为 0x4000, 此时将 0x2104.2 的 Data ID 中的数据写入至 ERM242-40DTD
3. 检测 0x2105.1 Status word, 直到数据切换为 0x8000, 表示 Data ID 写入完成
4. 将 0x2104.1 Control word, 切换为 0x0002, 使 0x2105.1 Status word 复位至 0x0000
5. 等待 0x2105.1 Status word 切换为 0 时, 将 0x2104.1 Control word 复位至 0

实例 4: 使用 CoDeSys 功能块操作 ERM242-40DTD 数据掉电保持

1. 向工程中添加库文件 MemoryUtils。

打开库管理器, 点击“添加库”



图 37 库管理器

在弹出的对话框中, 点击高级, 如图 38 所示。

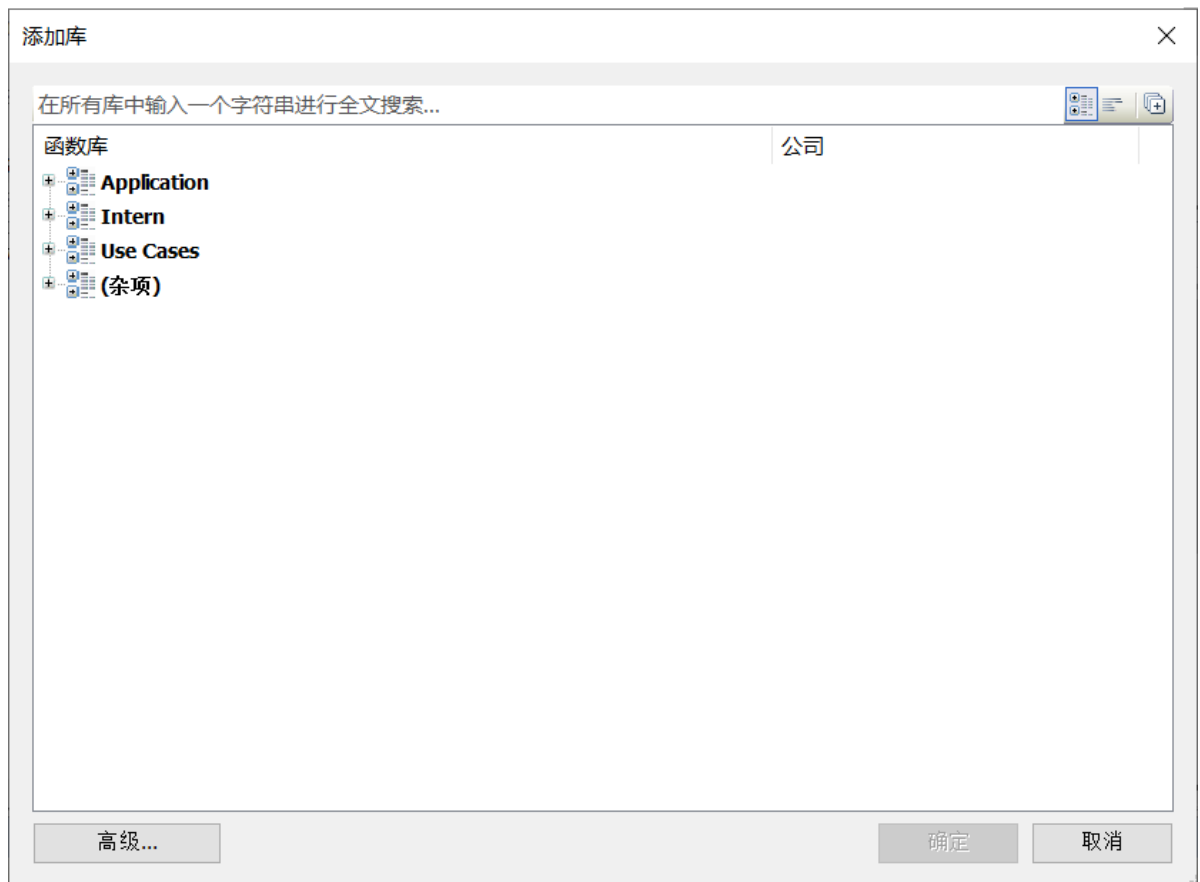


图 38 添加库界面

在图 39 中, 输入 MemoryUtils 找到对应的库, 并点击确定添加到工程中, 如图 40 所

示。

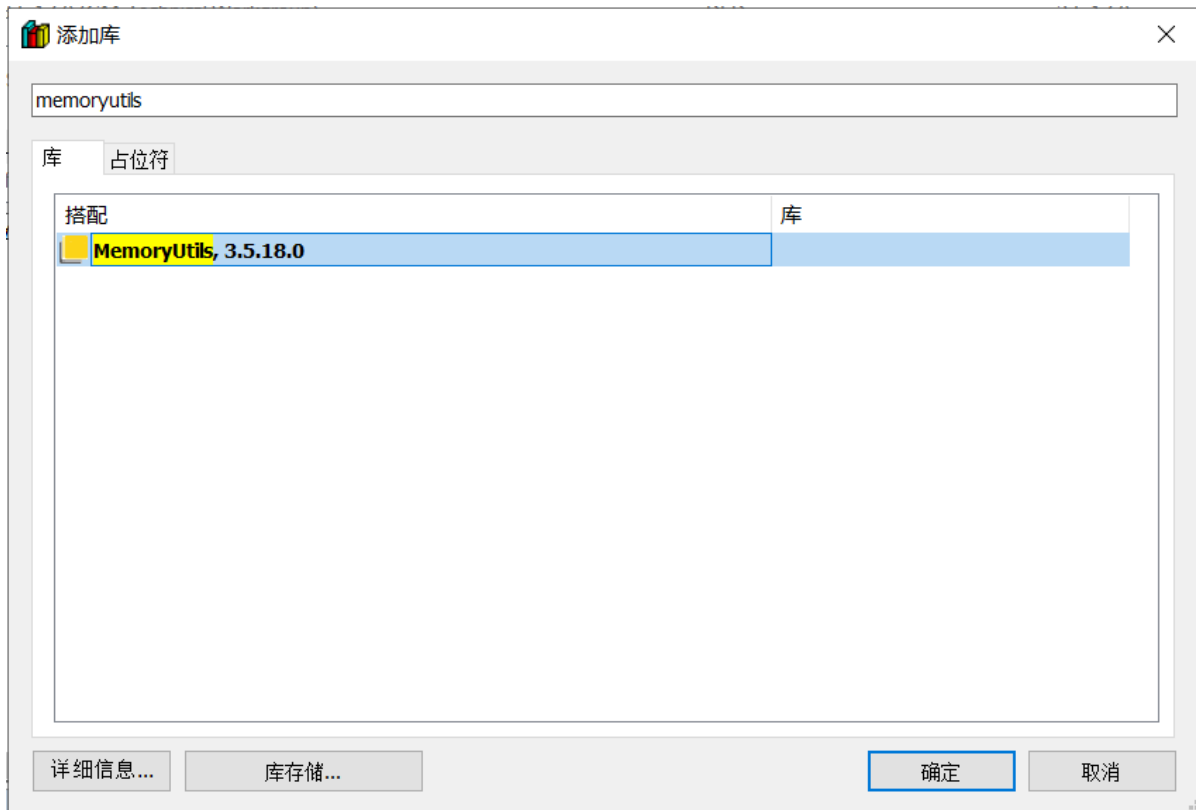


图 39 MemoryUtils 库文件

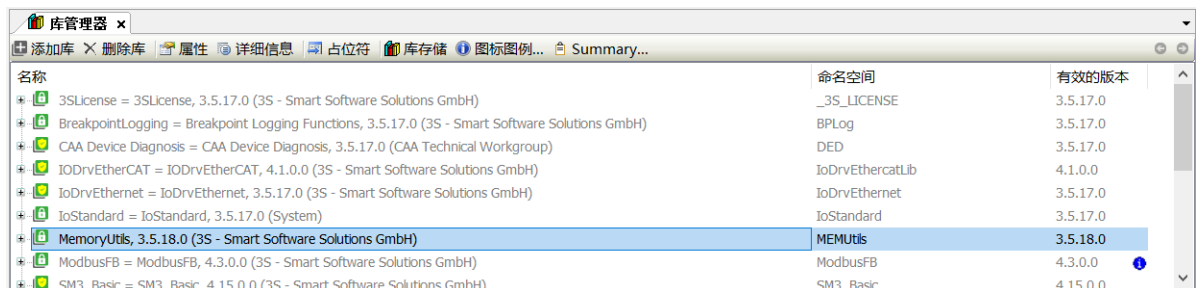


图 40 完成添加 MemoryUtils 库

2. 添加功能块

将 FB_AccessRetain 功能块添加到工程中，如图 41 所示。

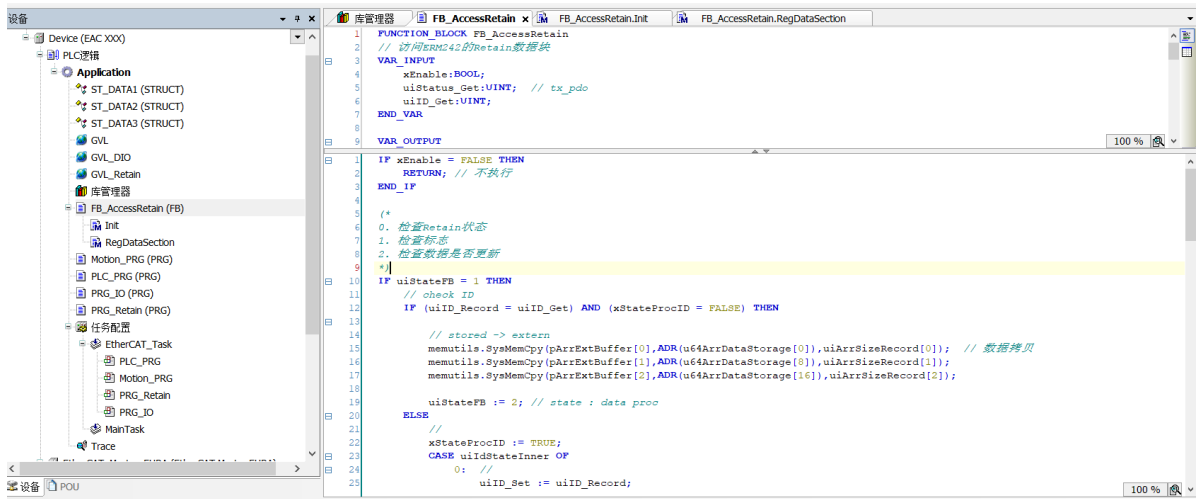


图 41 FB_AccessRetain 功能块

3. 自定义数据结构和数据实例化

右击工程，添加 DUT（结构体），并编辑结构体组成，依次如图 42，图 43 和图 44 所示。

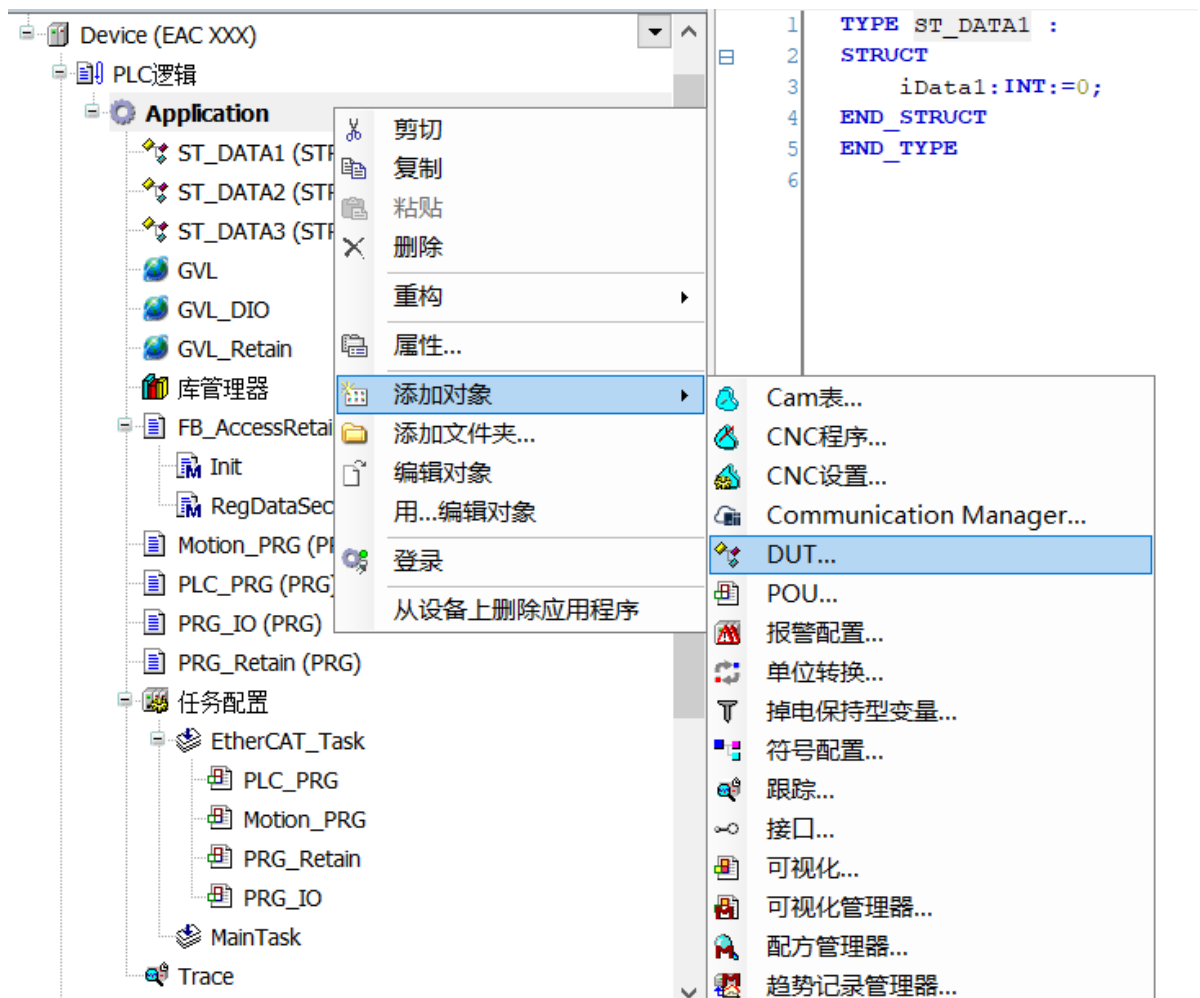


图 42 添加 DUT

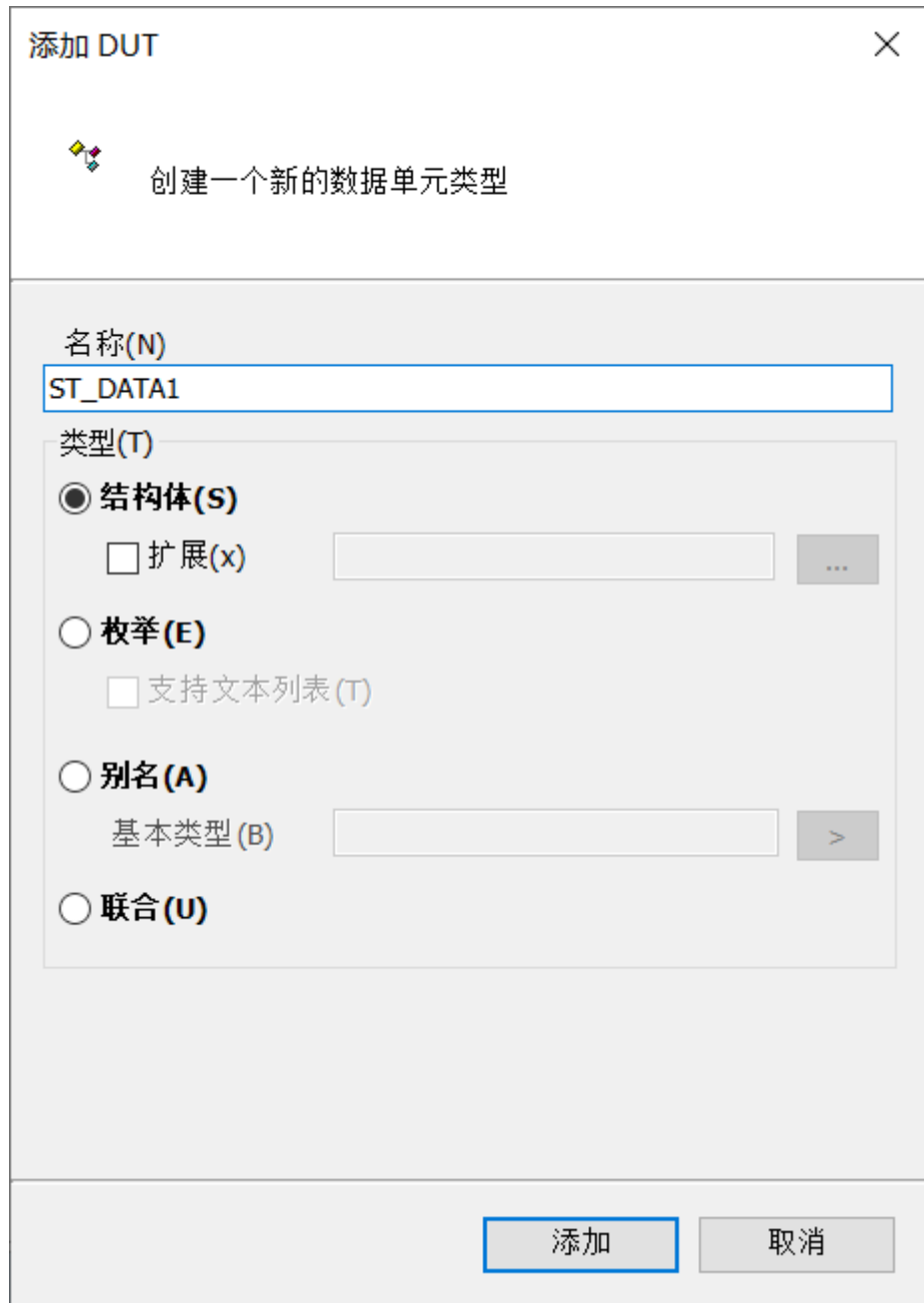


图 43 添加结构体

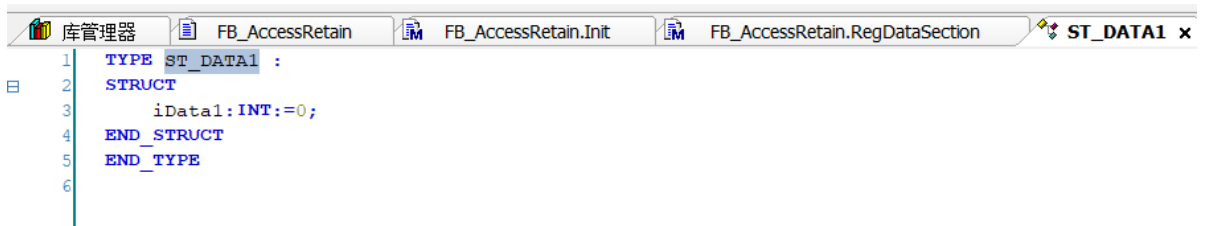


图 44 编辑结构体组成

定义一个全局变量区间 GVL_Retain，将新建的 3 个结构体实例化，图 45 所示。

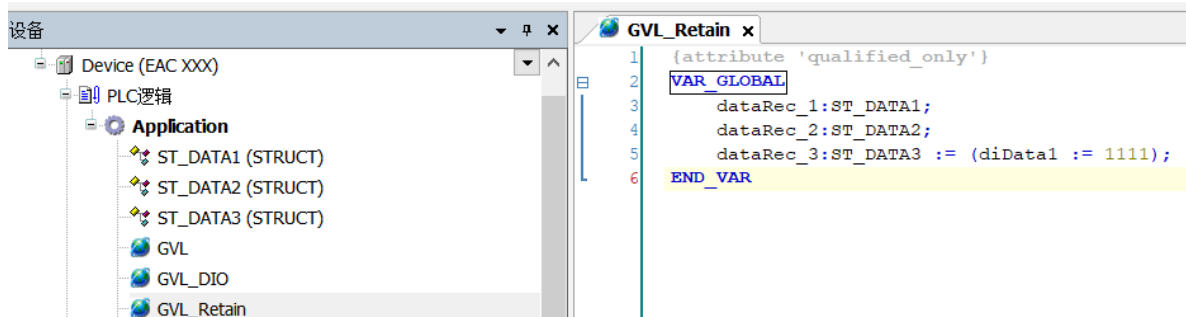


图 45 结构体实例化

4. 使用功能块

新建一个 PRG (PLC_Retain)，将其挂载至 EtherCAT_Task，然后在 PLC_Retain 中实例化 FB_AccessRetain 为 fbOpRetain，如图 46 所示。

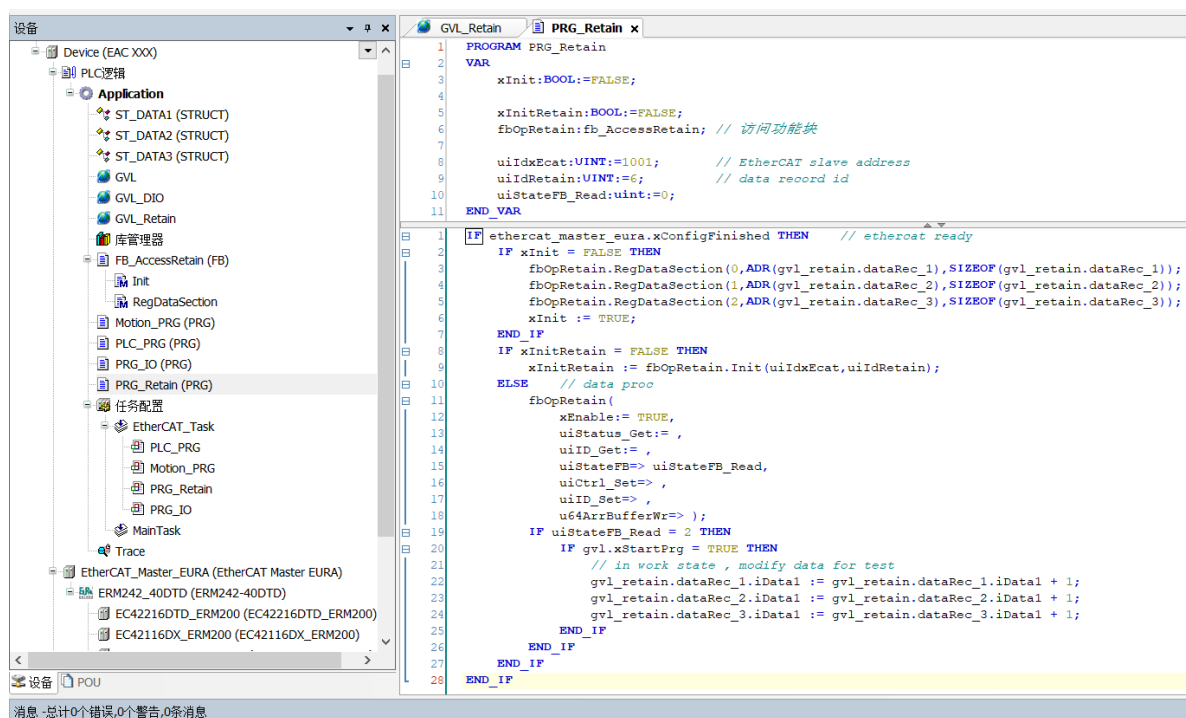


图 46 使用 FB_AccessRetain 功能块

FB_AccessRetain 功能块步骤：

1. RegDataSection: 用于注册数据区间。
 - (1) 参数 0: 数据区间编号
 - (2) 参数 1: 数据区对应的变量地址
 - (3) 参数 2: 数据区对应的变量长度，注意单个区间长度不能超过 64
2. Init: 功能块初始化。
 - (1) 参数 0: ERM242-40DTD 的 EtherCAT 地址。
 - (2) 参数 1: 数据 ID，如果读取与指定的不一致，将清除 ERM242-40DTD 保存的数

据。

3. FB_AccessRetain 主体：循环检查外部变量区间数据是否有改变。如果有改变，则将新数据下发至 ERM242-40DTD。

将功能块的数据与 ERM242-40DTD 的数据建立映射，如图 47 所示。

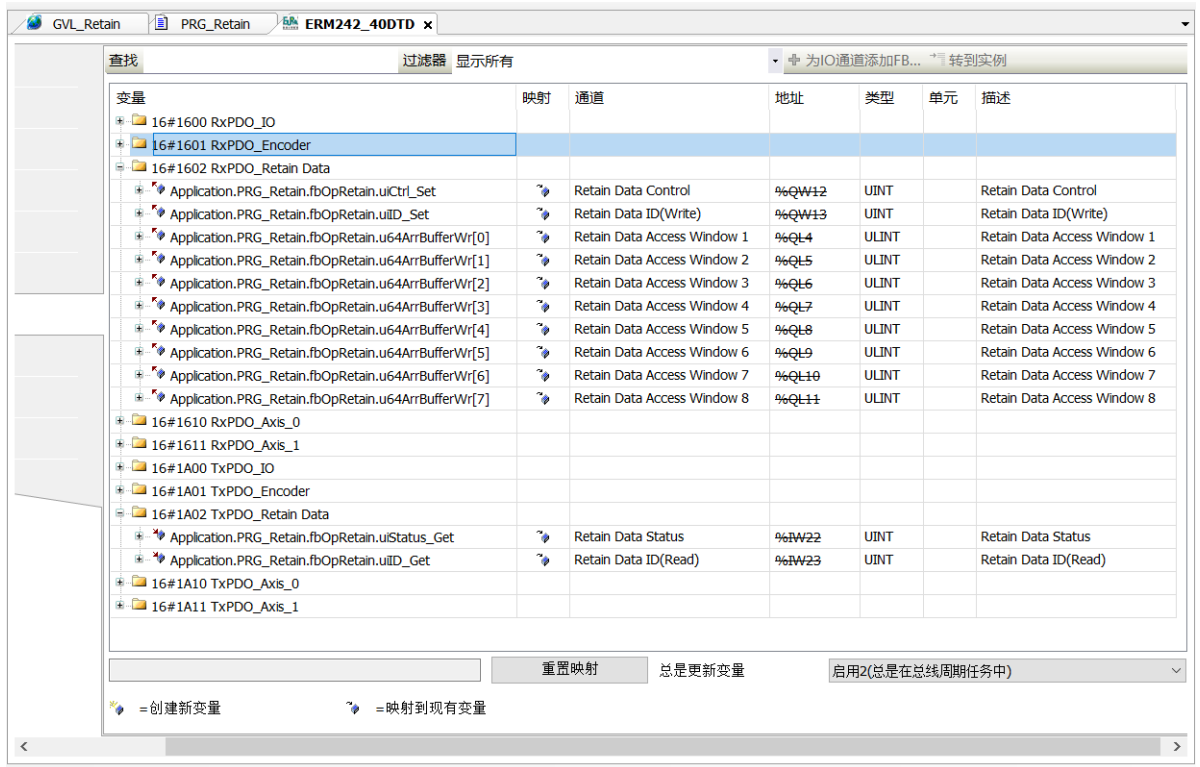


图 47 ERM242-40DTD 的 PDO 数据保持映射

连线调试，通过 Trace 观察写入区间的变化过程，如图 48 所示。

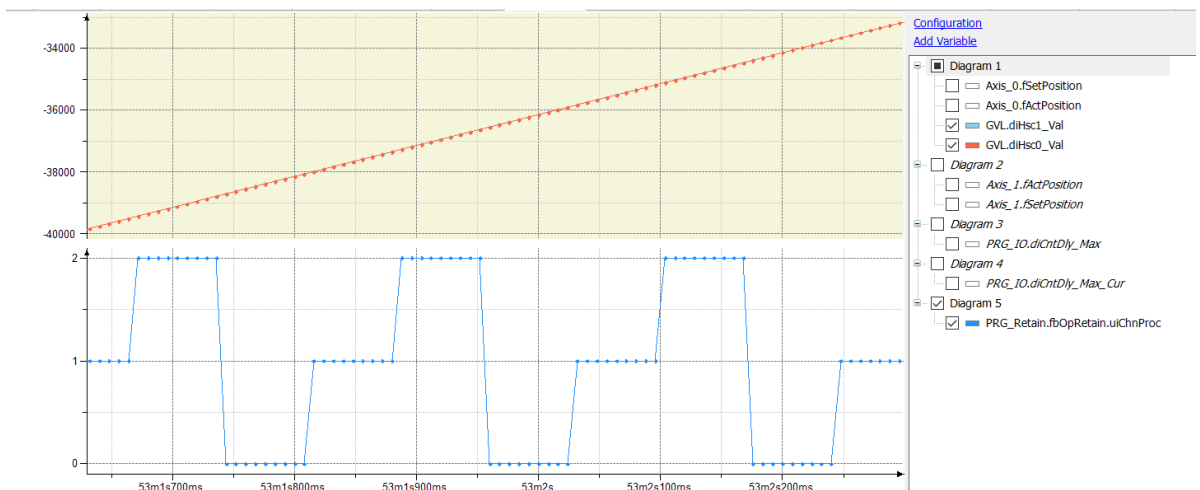


图 48 观察 ERM242-40DTD 的数据写入过程变化

第三章 EtherCAT 数据对象

3.1 自定义 EtherCAT 数据对象

表格 8 自定义 EtherCAT 数据对象列表

索引值	名称	类型	数据定义
0x2001	DI 滤波	Record	包含 10 个数据成员，每个 DI 占用 4 位数据。 最大滤波值 15ms(0xf)。 默认 3ms 滤波
0x2010	PTO 配置	Record	包含 2 个 16bit 数据成员，数据定义如下： PTO 的发送脉冲形式 0: 无 PTO (DO 模式) 1: 脉冲+方向 2: CW/CCW 3: AB
0x2011	HSC 配置	Record	包含 6 个成员,见 HSC 详细配置
0x2012	Axis 配置	Record	功能预留
0x2013	HSC 锁存配置	Record	包含 2 个子成员,见 Latch 详细配置，用于配置锁存信号的滤波参数。
0x2030	ExtBus 控制	Record	包含 3 个子成员,用于控制 ExtBus。 成员 1:16 位，控制字。 成员 2:16 位，启动延时，默认 100ms。 成员 3:16 位，错误重试次数，默认值 5。
0x2031	ExtBus 状态	Record	包含 3 个子成员，用于指示 ExtBus 状态。 成员 1:16 位，状态字。 成员 2:16 位，实际通信时间，单位 us。 成员 3:16 位，功能预留。
0x2060	参数写入控制	Record	包含 1 个成员：

			<p>成员 1: :16 位, 控制参数写入。</p> <p>0: 空闲, 1: 执行写入, 2: 复位状态</p>
0x2061	参数写入状态	Record	<p>包含 1 个成员:</p> <p>成员 1:16 位, 指示参数写入状态。</p> <p>0: 空闲, 1: 正在写入, 2: 写入完成</p>
0x2100	DO/AO 数据对象	Record	<p>包含 3 个成员:</p> <p>成员 1:32bit, 数字量输出</p> <p>成员 2:16bit,模拟量 1 通道输出</p> <p>成员 3:16bit,模拟量 2 通道输出</p>
0x2101	DI/AI 数据对象	Record	<p>包含 4 个成员:</p> <p>成员 1:32bit, 数字量输入</p> <p>成员 2:16bit,模拟量 1 通道输入</p> <p>成员 3:16bit,模拟量 2 通道输入</p> <p>成员 4:16bit,模拟量报警指示, bit1:模拟量 1 通道超量程报警; bit9:模拟量 2 超量程报警。</p> <p>如 0x0002: 模拟量 1 通道超量程报警。</p> <p>如 0x0200: 模拟量 2 通道超量程报警。</p>
0x2102	HSC 控制数据对象	Record	<p>包含 4 个成员:</p> <p>成员 1: ControlEnc, HSC 控制字</p> <p>成员 2: ControlLatch, 锁存控制字</p> <p>成员 3: ValSet00, HSC0 的设置值</p> <p>成员 4: ValSet01, HSC1 的设置值</p>
0x2103	HSC 状态数据对象	Record	<p>包含 8 个成员:</p> <p>成员 1: StatusEnc, HSC 状态字</p> <p>成员 2: StatusLatch, 锁存状态字</p> <p>成员 3: ValChn0, HSC0 的计数值</p> <p>成员 4: ValChn1, HSC1 的计数值</p>

			<p>成员 5: Latch0, 锁存 0 捕捉的计数值</p> <p>成员 6: Latch1, 锁存 1 捕捉的计数值</p> <p>成员 7: Latch2, 锁存 2 捕捉的计数值</p> <p>成员 8: 预留</p>
0x2104	掉电数据保持 控制对象	Record	<p>包含 2 个成员:</p> <p>成员 1: Control word, 控制字</p> <p>成员 2: Data ID, 数据 ID(写入)</p> <p>具体使用方法, 见 2.9 数据掉电保持的使用</p>
0x2105	掉电数据保持 状态对象	Record	<p>包含 2 个成员:</p> <p>成员 1: Status word, 状态字</p> <p>成员 2: Data ID, 数据 ID(读取)</p> <p>具体使用方法, 见 2.9 数据掉电保持的使用</p>
0x2106	掉电保持 写入缓冲	Record	<p>包含 8 个成员:</p> <p>每个成员数据长度为 64 位, 共 64 个字节。</p>
0x2107	掉电保持 数据读取区	Record	<p>包含 24 个成员:</p> <p>每个成员数据长度为 64 为, 共 192 个字节。</p>
0x2200~0x2270 0x2300~0x2370	ExtBus 数据映射区	Record	<p>根据 ExtBus 模块列表动态映射</p>

表格 9 HSC 配置的数据对象定义列表 (0x2011)

子索引	名称	位宽	说明
1	模式配置 1	16bit	对应 HSC0, 0: 计数模式, 1: Z 向清零功能

			注意：开启 Z 向清零功能时，在接收到 Z 向信号时，脉冲计数值被清零，此时 Z 向的 Latch 功能无效。 其他值无效，作为功能预留
2	功能配置 1	16bit	对应 HSC0, 1: 脉冲方向, 2: CW/CCW, 3: AB, 4: AB4 倍频 注意：默认值 0，仍然开启脉冲+方向采集
3	方向配置 1	16bit	对应 HSC0, 0: 负逻辑, 1: 正逻辑（默认值）
4	模式配置 2	16bit	对应 HSC1, 0: 计数模式 其他值无效，作为功能预留
5	功能配置 2	16bit	对应 HSC1, 1: 脉冲方向, 2: CW/CCW, 3: AB, 4: AB4 倍频 注意：默认值 0，仍然开启脉冲+方向采集
6	方向配置 2	16bit	对应 HSC1, 0: 负逻辑, 1: 正逻辑（默认值）

表格 10 HSC 锁存参数定义 (0x2013)

子索引	名称	位宽	说明
1	HSC0 锁存滤波值	16bit	该数值将同时作用于 1 个 HSC 的 2 个 Latch 引脚，见数据说明表格（表格 11）
2	HSC1 锁存滤波值	16bit	该数值将作用于 1 个 HSC 的 1 个 Latch 引脚，见数据说明表格（表格 11）

表格 11 HSC 的锁存参数定义列表

位数	7~4 bits	3~0 bits
内容	滤波倍数	开启滤波
数据定义	0:2x	0x01:开启第 1 通道滤波
	1:16x	0x02:开启第 2 通道滤波
	2:64x	0x03:开启 1 和 2 通道滤波

HSC 控制字 (0x2102.1, ControlEnc)

位	功能	说明
0	HSC0 的触发位	由 0 变为 1 时，并且 bit1 的值为 0，则 HSC0 的数值被设置为 0
1	HSC0 的设置位	由 0 变为 1 时，并且 bit0 的值为 1，则

		HSC0 的数值被设置为 0x2102.3 的数值
2~7	-	-
8	HSC1 的触发位	由 0 变为 1 时，并且 bit9 的值为 0，则 HSC1 的数值被设置为 0
9	HSC1 的设置位	由 0 变为 1 时，并且 bit8 的值为 1，则 HSC1 的数值被设置为 0x2102.4 的数值
10~31	-	-

HSC 状态字 (0x2103.1, StatusEnc)

位	功能	说明
0~31	HSC 状态字	指示 HSC 的状态，回传接收到的 HSC 控制字

表格 12 锁存控制字 (0x2102.2)

位	功能	说明
0	锁存 0 开关	0: 关闭; 1: 开启
1	锁存 0 触发方式	0: 单次; 1: 连续
2~3	-	-
4	锁存 0 上升沿触发开关	0: 关闭; 1: 开启
5	锁存 0 下降沿触发开关	0: 关闭; 1: 开启
6~7	-	-
8	锁存 1 开关	0: 关闭; 1: 开启
9	锁存 1 触发方式	0: 单次; 1: 连续
10~11	-	-
12	锁存 1 上升沿触发开关	0: 关闭; 1: 开启
13	锁存 1 下降沿触发开关	0: 关闭; 1: 开启
14~15	-	-
16	锁存 2 开关	0: 关闭; 1: 开启
17	锁存 2 触发方式	0: 单次; 1: 连续
18~19	-	-
20	锁存 2 上升沿触发开关	0: 关闭; 1: 开启
21	锁存 2 下降沿触发开关	0: 关闭; 1: 开启

22~31	-	-
-------	---	---

注意：在同时开启上升沿和下降沿触发时，将仅上升沿有效。

表格 13 锁存状态字（2103.2）：

位	功能	说明
0	锁存 0 开关指示	0：关闭；1：开启
1	锁存 0 上升沿触发状态	0：未检测到；1：检测到
2	锁存 0 下降沿触发状态	0：未检测到；1：检测到
3~7	-	-
8	锁存 1 开关指示	0：关闭；1：开启
9	锁存 1 上升沿触发状态	0：未检测到；1：检测到
10	锁存 1 下降沿触发状态	0：未检测到；1：检测到
11~15	-	-
16	锁存 2 开关指示	0：关闭；1：开启
17	锁存 2 上升沿触发状态	0：未检测到；1：检测到
18	锁存 2 下降沿触发状态	0：未检测到；1：检测到
19~31	-	-

表格 14 HSC 锁存与数据映射关系

HSC	锁存信号	数据映射
HSC0	锁存 0，位于端子 LE0	0x2103.5
	锁存 1，位于 Z 向信号	0x2103.6
HSC1	锁存 2，位于端子 LE1	0x2103.7
	-	-

3.2 标准 EtherCAT 数据对象

基础 EtherCAT 数据对象，见表格 15，用于表示设备的基本信息。

表格 15 EtherCAT 基础数据对象

索引值	名称	数据宽度	说明
0x1000	设备类型 (Device type)	32 位	固定值: 402 (只读)
0x1001	错误寄存器 (Error register)	8 位	功能预留
0x1008	设备名称 (Device name)	字符串	固定值: ERM242-40DTD (只读)
0x1009	硬件版本 (Hardware version)	字符串	只读
0x100A	软件版本 (Software version)	字符串	只读
0x1018	设备标识 (Identity)	Record	#1:厂商标识 (Vendor ID) #2:产品编码 (Product code) #3:版本号 (Revision) #4:序列号 (Serial number)
0x10F1	错误设置 (Error Settings)	Record	兼容性功能预留
0x10F8	时间戳对象 (Timestamp Object)	64 位	网络时间戳数据

标准 EtherCAT 数据对象遵循 CiA 402 规范，相邻 2 个轴之间的地址偏移为 0x800。控制字 (0x6040/0x6840) 和状态字 (0x6041/0x6841) 的操作规则符合 PDS 状态机，但是不包含功率控制相关部分。

表格 16 CSP 模式下标准 EtherCAT 数据对象列表

索引值	名称	数据宽度	说明
0x603F/0x683F	错误编码 (Error code)	16 位	0: 无错误 0x7500: 通信错误
0x6040/0x6840	控制字 (Control word)	16 位	见具体数据对象说明
0x6041/0x6841	状态字 (Status word)	16 位	见具体数据对象说明
0x605A/0x685A	快速停止选项编码 (Quickstop Option Code)	16 位	兼容性功能预留*
0x605B/0x685B	停机选项编码 (Shutdown option code)	16 位	兼容性功能预留*
0x605C/0x685C	屏蔽操作选项编码 (Disable operation option code)	16 位	兼容性功能预留*
0x605D/0x685D	停机选项编码 (Halt option code)	16 位	兼容性功能预留*
0x605E/0x685E	错误响应编码 (Fault Reaction Code)	16 位	兼容性功能预留*
0x6060/0x6860	操作模式 (Modes of Operation)	8 位	仅支持 CSP 模式(0x08)
0x6061/0x6861	操作模式显示值 (Modes of operation display)	8 位	仅支持 CSP 模式(0x08)
0x6062/0x6862	位置指令值	32 位	兼容性功能预留*

	(Position demand value)		
0x6063/0x6863	位置实际内部值 (Position actual internal value)	32 位	兼容性功能预留*
0x6064/0x6864	实际位置 (Position actual value)	32 位	目标位置的仿真数据
0x6076/0x6876	电机比例转矩 (Motor Rated Torque)	32 位	兼容性功能预留*
0x607A/0x687A	目标位置 (Target Position)	32 位	目标位置仿真， PTO 输出脉冲数据
0x607B/0x687B	位置范围限制 (Position range Limit)	Record	兼容性功能预留*
0x607D/0x687D	软件位置限制 (Software position limit)	Record	兼容性功能预留*
0x6085/0x6885	快速停止减速度 (Quick stop deceleration)	32 位	兼容性功能预留*
0x60C2/0x68C2	插补周期 (Interpolation Time Period)	Record	兼容性功能预留*
0x6502/0x6D02	支持的操作模式 (Supported drive modes)	32 位	支持的驱动模式，仅支持 CSP 模式 (0x00000080)

注意：兼容性功能预留*，表示为了兼容不同的 EtherCAT 主站功能而保留的数据对象，作为预留数据对象。

表格 17 控制字 (0x6040/0x6840)

Bit	8~15	7	4~6	3	2	1	0
-----	------	---	-----	---	---	---	---

名称	-	fr	-	eo	qs	ev	so
<p>Fr = fault reset, 复位</p> <p>eo= enable operation, 使能操作</p> <p>qs=quick stop, 快速停止</p> <p>ev=enable voltage, 使能电压</p> <p>so=switch on, 准备操作</p>							

表格 18 状态字 (0x6041/0x6841)

Bit	10~15	9	7~8	6	5	4	3	2	1	0
名称	-	rm	-	sod	qs	ve	f	oe	so	rtso
<p>rm = remote, 响应控制字处理</p> <p>sod = switch on disabled, 操作屏蔽</p> <p>qs = quick stop, 快速停止</p> <p>ve = voltage enabled, 电压使能</p> <p>f = fault, 错误</p> <p>oe = operation enabled, 使能操作</p> <p>so = switched on, 操作完成</p> <p>rtso = ready to switch on, 准备操作就绪</p>										

第四章 FAQ

1. CoDeSys 日志报警信息：AL Status 16#70(0x70)

原因：说明实际连接的 ExtBus 模块的数量或者类型与工程中的不相符

解决方法：请排查工程代码中使用的模块，或者在“关闭电源”的条件下排查实际连接的模块，修改配置使两者一致。

提示：对于新建工程，可以通过在线扫描的方式自动获取 ExtBus 模块列表，然后添加到工程中，使两者保持一致。

2. 在 CoDeSys 中 ExtBus 模块与 CiA402 轴的添加顺序？

先添加 ExtBus 模块，然后添加 CiA402 轴。

3. ERM242-40DTD 的数据掉电保持与 EAC 运动控制器数据掉电保持的差异？

ERM242-40DTD 内置一个 NVRAM，通过 PDO 周期性写入数据以打包保持数据的目的，数据从写入、发送和检测写入状态完成大约需要 8~9 个通信周期。

EAC 运动控制器有内置 UPS 电路，在掉电时触发掉电检测机制，将数据执行保存。

特点	EAC 运动控制器	ERM242-40DTD
保存容量	100 KB	192 B
写入时间	掉电时写入	周期性写入，无写入次数限制 每个区段的完整写入周期约为 8~9 个周期
使用方法	Retain/Persist 变量定义	搭配 CoDeSys 功能块
适用条件	-	可以搭配缺少数据掉电保持功能的控制器，如 ECP100 等高性价比控制器 如： 将设备参数用配方形式数据保存 将加工过程数据（如已完成工件数量）用 ERM242-40DTD 的掉电数据保持功能保存

敬告用户：

感谢您选用我司产品，为保证您正确使用本产品及得到我司最佳售后服务，请认真阅读下述条款，并做好相关事宜。

只有具备一定的电气知识的操作人员才能够对本产品进行接线、上电操作；手册中示例程序仅供参考，不保证其实用性。

本公司致力于产品的不断改善和升级，手册提供资料如有变更，恕不另行通知，请自行访问本公司网站获取。

产品保修范围：按使用要求正常使用情况下，所产生的故障。

产品保修期限：本公司产品的保修期为自出厂之日起，十二个月以内。保修期实行长期技术服务。

非保修范围：任何违反使用要求的认为意外、自然灾害等原因导致的损坏，以及未经许可而擅自对产品拆卸、改装及修理的行为，视为自动放弃保修服务。

从中间商处购入产品：凡从经销代理商处购买产品的用户，在产品发生故障时，请与经销商、代理商联系。

免责条款：因下列原因造成的产品故障不在厂家 12 个月免费保修服务范围之内：

(1)、厂家不依照《产品手册》中所列程序进行正确的操作；

(2)、用户未经与厂家沟通自行修理产品或擅自改造产品；

(3)、因用户环境不良导致产品器件异常老化或引发故障；

(4)、因用户超过产品的标准范围使用产品；

(5)、由于地震、火灾、风水灾害、雷击、异常电压或其他自然灾害等不可抗力的原因造成的产品损坏；

(6)、因购买后由于人为摔落及运输导致硬件损坏。

责任：无论从合同、保修期、疏忽、民事侵权行为、严格的责任、或其他任何角度讲，EURA 和他的供货商及分销商都不承担以下由于设备所造成的特殊的、间接的、继发的损失责任。其中包括但不仅仅局限于利润和收入的损失，使用供货设备和相关设备的损失，资金的花费，代用设备的花费，工具费和服务费，停机时间的花费，延误，及购买者的客户或任何第三方的损失。另外，除非用户能够提供有力的证据，否则公司及它的供货商将不对某些指控如：因使用不合格原材料、错误设计、或不规范生产所引发的问题责任。

解释权归欧瑞传动电气股份有限公司。

如果您对 EURA 的产品还有疑问，请与 EURA 公司或其办事处联系。技术数据、信息、规范均为出版时的最新资料，EURA 公司保留部事先通知而更改的权利，并对由此造成的损失不承担任何责任。解释权归 EURA 公司。

扫描关注
欧瑞官方网站



EURa® 欧瑞传动电气股份有限公司
DRIVES EURA DRIVES ELECTRIC CO.,LTD
24小时服务热线：4006-866-333
公司网址：www.euradrives.com