

Q 系列控制器常见问题 100 问

基本信息说明

产 品 类 型	Q 系列	产 品 型 号	Q 系列控制器	保密等级	☒ 公开 ☐ 内部公开 ☐ 保密
				文档编号	-
修 订	-	作 者		发布日期	2025-11-11

本文档使用硬件设备和软件工具

- Q 系列上位编程软件 HCP Works3 V1.3.0
- 控制器：Q 系列控制器
- 扩展卡：HCQXB-CAN-BD、HCQXB-RTC-BD、HCQXB-2RS232-BD、HCQXB-2RS485-BD

本文档目标：方便市场技术人员快捷查询 Q 系列控制器应用问题

文档更新和发布状态：

发布日期	版本	更新内容	发布状态
2025-11-11	V1.0	市场问题	已发布

目 录

一、 通讯应用	1
1. EIP 通讯基础知识.....	1
2. EIP 通信 1 类传输 (Class1) 配置过程.....	1
3. EIP 通信做客户端 3 类传输 (Class3) 配置过程	1
4. EIP 常见错误—IP 配置更改	1
5. EIP 常见错误—Extended Status: 16#112	2
6. EIP 常见错误—Extended Status: Invalid O->T size (16#127)	2
7. EIP 常见错误—Extended Status: Target for connection not configured (16#110)	2
8. EIP 常见错误—Extended Status: null forward open not supported (16#132)	2
9. EIP 常见错误—Extended Status: Invalid O->T Fix/var value (16#11F)	2
10. Modbus 通讯基础知识	3
11. ModbusTCP 主站日志报错 Connection aborted:socket keep-alive expired.....	3
12. ModbusTCP 从站在程序中如何切换 IP 和端口?	3
13. ModbusTCP 读写通道在程序中如何触发?	4
14. Modbus 寄存器地址分配	4
15. 串口模块 HCQX-RS02-D4-M 添加多个从站, 通讯状态是否与实际物理连接顺序有关?	5
16. 上位机与控制器 ModbusTCP 通讯控制伺服等机构, 为什么会偶发动作错误?	5
17. Socket 自由协议通讯使用说明.....	5
18. Socket 自由协议通讯使用字符串作为缓存区, 接收端会受到很多 0.....	5
19. Socket 自由协议通讯 TCP_Cilent 报错 TIME_OUT.....	5
20. Q0 控制器本体两个 COM 口通讯报错.....	5
二、 编程应用	6

1. 基础知识	6
2. 从 PLC 中将文件存到 U 盘，但在电脑上看不到 U 盘中存入的文件	6
3. BOOL 型变量在线显示 “? ? ? ”	6
4. 新掉电保持模式对应控制器固件为多少?	6
5. 字符串数据如何分配地址?	6
6. 控制器恢复默认 IP 方法	6
7. 指针类型数据的大小	7
8. 十进制转化十六进制字符串	7
9. 十进制转化 ASCII	7
10. 设备系统信息	7
11. 掉电保持区不支持指针类型的变量	8
12. 文件操作怎么换行以及美元符号 ‘\$’ 怎么输入?	8
13. 控制器加密	9
14. 控制器解密	10
15. 声明功能块后编译报错 C0136: ambiguous use of name “功能块名字”	10
16. IO 地址强制与解除	10
17. 为什么 FOR 循环变量的目标值设置成数据类型上限后导致死循环?	11
18. 如何获取字符个数超过 255 的字符串长度?	11
19. 欧姆龙控制器配 EC02 报错检测到 AL 状态错误	13
20. 拔掉 U 盘后 USBDevices 文件夹依然存在	13
21. 如何读取超过 SMC3_ETC_ReadParameter CoE 指令 DWORD 类型数据长度的 SDO?	14
22. STRING 变量可以直接通过 MOVE 指令赋值中文吗?	14
三、运动控制	15

1. 轴控状态机及功能块触发示例.....	15
2. 功能块报错 'SMC_MSI_INVALID_EXECUTION_ORDER'	17
3. 转矩模式下轴在动作，轴状态显示 standstill	17
4. 使用 MC_Home 指令回零，动作异常	17
5. 执行 MC_Home 指令后，16#6064 反馈不会变为 0	17
6. 上升沿触发 MC_Home 指令报错 SMC_FB_WASNT_CALLED_DURING_MOTION	17
7. 控制器带轴能正常使能和复位，但是不能点动、定位、匀速等操作。	18
8. 如何在程序中修改轴的减速比？	18
9. 如何设置轴的软限位？	18
10. 开启软限位后，定位超过目标位置，报错触发软限位	18
11. 双主站控制器在使用 Codesys 官方软件时添加 Port4 口后，Port3 口丢同步	19
12. HCP Works3 V1.3.0 版本软件新建脉冲轴，报错轴类型不匹配	19
13. SV730W 驱动只有第一个轴的转矩可以写入，其它轴写不进去	19
14. 绝对值电机线性轴位置溢出后，断电重启会丢失位置	20
15. 伺服转矩单位不同	21
16. BufferMode 设置了缓存，但是运行还是减速到零才触发第二段运动	21
17. 映射原点信号，回原时报错触发硬限位	21
18. 改变 EtherCAT 任务周期后，轴运行速度不对	21
19. 切换控制模式后报错 SMC_MOVING_WITHOUT_ACTIVE_MOVEMENT	22
20. 轴报错 SMC_DI_POSITIONLAGERROR 位置滞后错误	22
21. 轴启动过程中卡状态在 1020	23
22. 轴报错 SMC_CGR_INVALID_POSPERIOD（模态轴上限）	24
23. CNC 中 DXF 图形导入	24

24. CNC 运行无法达到设定速度	25
25. 写时钟指令导致 EtherCAT 掉线	25
26. 程序中重启 EtherCAT 总线.....	26
27. 如何获取所有从站状态	26
28. 在程序中启用禁用 EtherCAT 从站.....	27
29. 如何将 DXF 文件导入控制器中，转换成 CNC 程序？	28
四、异常问题	30
1. 控制器异常——面板报错 ER0000	30
2. 控制器异常——面板报错 ER0051	30
3. 控制器异常——面板报错 ER0102、ER0152.....	30
4. 控制器异常——面板卡 HCFA	30
5. EtherCAT 总线异常——丢同步、掉线.....	30
6. EtherCAT 总线异常——扫描设备时提示“存储库中找不到此设备”	30
7. EtherCAT 总线异常——扫描不到设备.....	30
8. EtherCAT 总线异常——扫描正常，但总线起不来.....	31
9. 下载报错——Download Denied on Application from CmpCoreDump.....	31
10. 下载报错——原因未知	32
11. 编译报错——没有可用于此对象的源代码，因为它位于已编译的库'sm3basic,4.10.0.0.....	32
12. 编译报错——Internal error:System.NullReferenceException.....	32
13. 编译报错——The Library *** is only supported in 32 bit applications	32
14. 编译报错——Not enough persistent memory	33
五、典型应用	34
1. 隐式函数检查.....	34

2.	PID 温度控制	35
3.	配方功能使用	36
4.	高速输入输出功能应用	38
5.	UPS 模块应用说明	41

一、通讯应用

1. EIP 通讯

- 1) 同一网口可以同时做扫描器和适配器；
- 2) 扫描器可以连接多个适配器；
- 3) 适配器可以通过不同的连接被多个扫描器访问（HCP Works3 V1.3.0 软件）；
- 4) 标准数据帧长度 1514 字节，建议 IO 交互数据 ≤ 1400 字节；
- 5) 做适配器时，最多 61 个连接（扫描器），每个连接最大通讯字节数 505 输入/509 输出（单通道或多通道都是该大小）；
- 6) 做扫描器时，共 253 个设备树（255-本体地址-广播地址），单设备总字节数同适配器；

2. EIP 通信 1 类传输（Class1）配置过程

如果控制器做 EIP 适配器，首先是设备组态，添加 EtherNet 端口，端口下面添加适配器，适配器下面添加模块；其次是在模块中配置输入输出程序集；最后选择网络接口并导出 EDS 文件。

如果控制器做 EIP 扫描器，首先要安装适配器的 EDS 文件；其次添加 EtherNet 端口，端口下面添加扫描器，在扫描器上添加刚才安装的适配器；最后是配置适配器的 IP 地址，关闭兼容性检查。

详细操作步骤可以联系产品线技术人员索要 Q 系列控制器 EIP 应用文档。

3. EIP 通信做客户端 3 类传输（Class3）配置过程

安装库文件 EtherNetIP_Tag_V1.0.3，在程序中实例化 EtherNetIPScanner 并调用，使用 Case 语句轮询数据读取、写入。

```
1 //实例客户端功能块
2 EtherNetIPScanner_0(
3     bEnable:=bEnable , //使能连接
4     bAutoReconnect:=TRUE , //自动重连
5     sIpAddress:= '192.168.88.210' , //服务器ip
6     uiPort:= , //服务器端口，默认44818不用修改
7     bTcpActive=> , //TCP通讯正常
8     bEipCipActive=> , //EIP通讯正常
9     bBusy=> ,
10    bDataReady=> ,
11    bError=> ,
12    sStatus=> ,
13    sError=> );
14
15 //轮询数据读取
16 CASE step OF
17     0://判断连接成功
18         IF EtherNetIPScanner_0.bEipCipActive THEN
19             step:=10;
20         END_IF
21     10://读取rtteststring标签，数据类型string
22         IF EtherNetIPScanner_0.bReadTag (ADR('rtteststring'),eCipTypes._STRING,ADR(rtteststring),SIZEOF(rtteststring),1,ADR(read),FALSE) AND NOT EtherNetIPScanner_0.bBusy THEN
23             step:=step+10;
24         END_IF
25     20://写入wteststring标签w,类型STRING
26         IF EtherNetIPScanner_0.bWriteTag (ADR('wteststring'),eCipTypes._STRING,ADR(wteststring),SIZEOF(wteststring),1,ADR(write),FALSE) AND NOT EtherNetIPScanner_0.bBusy THEN
27             step:=0;
28         END_IF
29 END_CASE
```

可以使用第三方软件做服务器，添加对应的标签变量类型，监控读写效果。

4. EIP 常见错误—IP 配置更改

设备运行，存在诊断信息，一般是网卡配置信息和实际信息不一致，可以运行但是存在错误，需要重新浏览配置下网卡



严重	时间戳	描述	组件
	08.09.2021 13:38:19	IoDrvEthernet: IP settings of Ethernet node aren't as configured.	IoDrvEthernet
	08.09.2021 09:38:15	IoDrvEthernet: IP settings of Ethernet node aren't as configured.	IoDrvEthernet

5. EIP 常见错误—Extended Status: 16#112

RPI 值不可接受，及远程适配器设置的 RPI 值和从设备的 Adapter IOTask 时间 设置不匹配导致。

- 1) 单 Adapter: $RPI \geq n * IOTask$ (n 为大于 0 的整数) ;
- 2) 同时 Scanner 和 Adapter: $RPI \geq n * ScannerIOTask$ (n 为大于 0 的整数);

严重	时间戳	描述	组件
	28.05.2024 19:09:01....	IoDrvEthernetIP: Connection failure. (16#1)	IoDrvEtherNe
		Extended Status: 16#112	
	28.05.2024 19:09:01....	IoDrvEthernetIP: Open Connection with serial number 1	IoDrvEtherNe
	28.05.2024 19:08:59....	IoDrvEthernetIP: Connection failure. (16#1)	IoDrvEtherNe
	28.05.2024 19:08:59....	IoDrvEthernetIP: Open Connection with serial number 1	IoDrvEtherNe
	28.05.2024 19:08:54....	IoDrvEthernetIP: Connection failure. (16#1)	IoDrvEtherNe
	28.05.2024 19:08:54....	IoDrvEthernetIP: Close Connection with serial number 1	IoDrvEtherNe

6. EIP 常见错误—Extended Status: Invalid O->T size (16#127)

O->T 设置的字节大小和 Adapter 配置的字节大小不一致。

严重	时间戳	描述
	30.05.2024 10:39:34.848	IoDrvEthernetIP: Connection failure. (16#1)
		Extended Status: Invalid O->T size. (16#127)

7. EIP 常见错误—Extended Status: Target for connection not configured (16#110)

配置的实例路径和 Adapter 的不一致。

严重	时间戳	描述	组件
	30.05.2024 10:44:41.860	IoDrvEthernetIP: Connection failure. (16#1)	IoDrvEtherNetIP
		Extended Status: Target for connection not configured. (16#110)	

8. EIP 常见错误—Extended Status: null forward open not supported (16#132)

配置的连接类型为 空，和 adapter 的配置不一致。

严重	时间戳	描述	组件
	30.05.2024 11:04:29.812	IoDrvEthernetIP: Connection failure. (16#1)	IoDrvEtherNetIP
		Extended Status: Null forward open not supported. (16#132)	
	30.05.2024 11:04:29.662	IoDrvEthernetIP: Open Connection with serial number 1	IoDrvEtherNetIP

连接类型	空	连接类型	点对点
连接优先	预订	连接优先	预订

9. EIP 常见错误—Extended Status: Invalid O->T Fix/var value (16#11F)

固定/变量和 adapter 配置不一致。

严重	时间戳	描述
	30.05.2024 11:11:57.675	IoDrvEthernetIP: Connection failure. (16#1)
		Extended Status: Invalid O->T Fix/Var value. (16#11F)
	30.05.2024 11:11:57.524	IoDrvEthernetIP: Open Connection with serial number 1
	30.05.2024 11:11:55.516	IoDrvEthernetIP: Connection failure. (16#1)

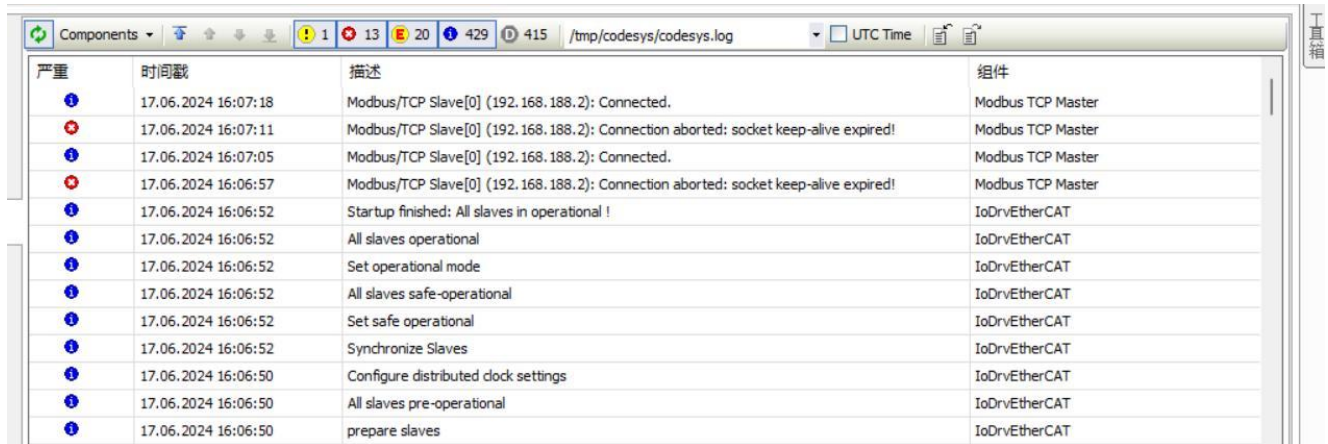
10. Modbus 通讯

- 1) 同一网口可以同时做 ModbusTCP 主站与从站;
- 2) 不同网口 (不同 IP) 可以同时启用 ModbusTCP 通讯;
- 3) ModbusTCP 从站 (服务器) 最多连接 16 个端口 (客户端), 每个端口最多 64 个连接;
- 4) ModbusTCP 主站对从站有限制, 最大 64 节点, 最多 99 个通道, 单通道最大读 125 个字节, 写 123 个字节;
- 5) ModbusRTU 支持的连续地址最大 100 个字;
- 6) ModbusRTU 从站最大个数为 31;

11. ModbusTCP 主站日志报错 Connection aborted:socket keep-alive expired

此报错为 ModbusTCP 从站响应超时, 默认超时时间为 1 秒, 当主站发送请求, 未在 1 秒内接收到响应则报错保活过期。请检查配置信息以及主站、从站、上位机 IP 地址是否正确。局域网中不要有相同的 IP 地址, 否则会造成通讯错误。

注意: socket keep-alive expired, 保活过期非 TCP 协议栈报错保活过期, 属于应用层方面报错, 而非报文中的 keep-alive。



严重	时间戳	描述	组件
1	17.06.2024 16:07:18	Modbus/TCP Slave[0] (192.168.188.2): Connected.	Modbus TCP Master
13	17.06.2024 16:07:11	Modbus/TCP Slave[0] (192.168.188.2): Connection aborted: socket keep-alive expired!	Modbus TCP Master
20	17.06.2024 16:07:05	Modbus/TCP Slave[0] (192.168.188.2): Connected.	Modbus TCP Master
429	17.06.2024 16:06:57	Modbus/TCP Slave[0] (192.168.188.2): Connection aborted: socket keep-alive expired!	Modbus TCP Master
415	17.06.2024 16:06:52	Startup finished: All slaves in operational !	IoDrvEtherCAT
	17.06.2024 16:06:52	All slaves operational	IoDrvEtherCAT
	17.06.2024 16:06:52	Set operational mode	IoDrvEtherCAT
	17.06.2024 16:06:52	All slaves safe-operational	IoDrvEtherCAT
	17.06.2024 16:06:52	Set safe operational	IoDrvEtherCAT
	17.06.2024 16:06:52	Synchronize Slaves	IoDrvEtherCAT
	17.06.2024 16:06:50	Configure distributed clock settings	IoDrvEtherCAT
	17.06.2024 16:06:50	All slaves pre-operational	IoDrvEtherCAT
	17.06.2024 16:06:50	prepare slaves	IoDrvEtherCAT

12. ModbusTCP 从站在程序中如何切换 IP 和端口?

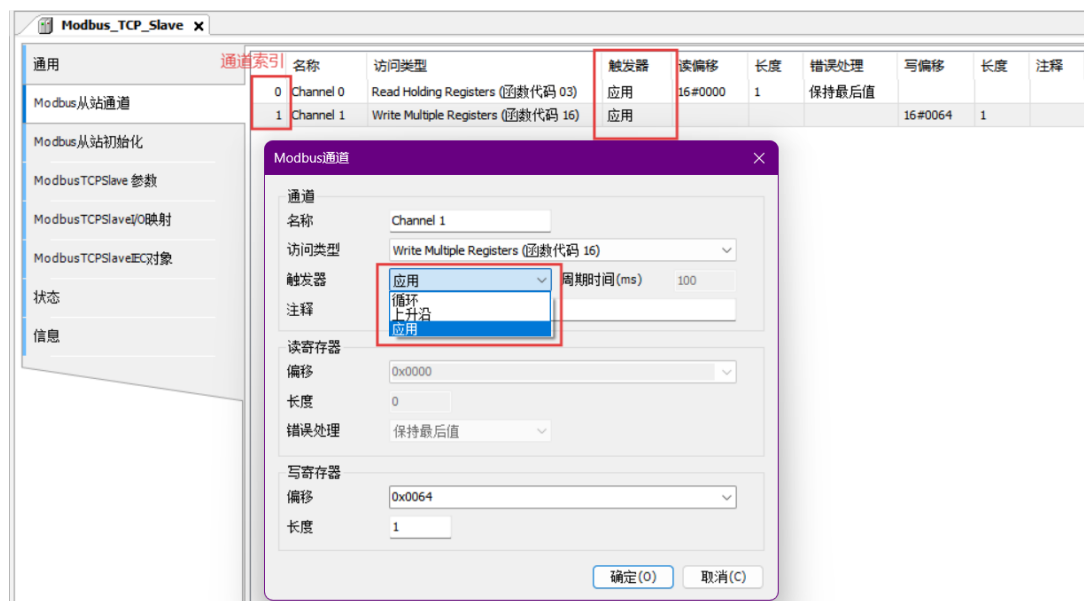
程序中调用从站的 UpdateCommunicationSettings 方法就可以修改对应从站的 IP 和端口, 具体程序示例如下:

```
PROGRAM P03_ChangeTCPSlave
VAR
    iStep      :INT;
    bChange    :BOOL;
    ipAddr     :ARRAY[0..3] OF BYTE:= [192,168,188,100];
    uiPort     :UINT:=502;
    udResult   :UDINT;
    bChangeOK  :BOOL;
END_VAR

//切换Modbus_TCP从站IP地址和端口
CASE iStep OF
    0://开始判断
        IF bChange THEN
            bChangeOK:=FALSE;
            bChange:=FALSE;
            iStep:=10;
        END_IF
    10://关闭从站使能
        Modbus_TCP_Slave.Enable:=FALSE;
        IF Modbus_TCP_Slave.ComState=0 THEN
            iStep:=20;
        END_IF
    20://改变IP和端口
        udResult:=Modbus_TCP_Slave.UpdateCommunicationSettings(ipAddress:=ipAddr,uiPort:=uiPort);
        IF udResult=0 THEN
            bChangeOK:=TRUE;
            iStep:=30;
        END_IF
    30://开启从站使能
        Modbus_TCP_Slave.Enable:=TRUE;
        iStep:=0;
END_CASE
```

13. ModbusTCP 读写通道在程序中如何触发？

读写通道的触发设置为应用程序



在程序中调用 ModbusChannel 功能块，具体示例如下：

```

PROGRAM P04_TriggerModbusTCPChannel
VAR
    ModbusChannel_0      :ModbusChannel;
    ModbusChannel_1      :ModbusChannel;
    bRead                :BOOL;
    bWrite                :BOOL;
END_VAR

ModbusChannel_0(
    slave:=Modbus_TCP_Slave , //从站名
    xExecute:=bRead ,         //上升沿触发
    xAbort:= ,
    iChannelIndex:=0 ,        //通道索引
    xBusy=> ,
    xDone=> ,
    xError=> ,
    xAborted=> ,
    ModbusError=> );

ModbusChannel_1(
    slave:=Modbus_TCP_Slave , //从站名
    xExecute:=bWrite ,        //上升沿触发
    xAbort:= ,
    iChannelIndex:=1 ,        //通道索引
    xBusy=> ,
    xDone=> ,
    xError=> ,
    xAborted=> ,
    ModbusError=> );
  
```

14. Modbus 寄存器地址分配

Q 地址	Modbus 地址	功能码	类型	读写状态
%QX0.0-%QX8191.7	0-65535	01H	线圈	可读
		05H、0FH		可写
%IX0.0-%IX8191.7	0-65535	02H	离散输入	可读
%MW0-%MW65535	0-65535	04H	输入寄存器	可读
%MW0-%MW65535	0-65535	03H、06H、10H、17H	保持寄存器	可读可写

15. 串口模块 HCQX-RS02-D4-M 添加多个从站，通讯状态是否与实际物理连接顺序有关？

RS485 通讯理论上是通过识别从站 ID 发送和接收数据包，和实际物理接线顺序无关。需要仔细排查接线与参数配置。线缆建议使用屏蔽双绞线，两端增加终端电阻，接地良好等。

16. 上位机与控制器 ModbusTCP 通讯控制伺服等机构，为什么会偶发动作错误？

这种情况一般都是上位机给的信号时间短，加上通讯延迟等因素造成控制不稳定。建议增加互锁信号，比如上位机控制伺服 STOP，要等到 PLC 把伺服轴的 Stopping 状态返回给上位机，上位机才能复位触发 STOP 信号。

17. Socket 自由协议通讯使用说明

- 1) TCP_Read.pData 对应缓存区大小必须大于或等于 TCP_Read.szSize 的值，否则可能出现内存覆盖，引起系统宕机；
- 2) TCP_Read.xEnable 串联 TCP_Client.xActive 引脚（建立连接）；
- 3) TCP_Client.xEnable 串联 NOT TCP_Read.XError 引脚；

18. Socket 自由协议通讯使用字符串作为缓存区，接收端会受到很多 0

TCP_Write 的 szSize 引脚不是定义的缓冲区的大小，是需要发送数据的字节数。使用字符串的时候，把 SIZEOF（字符串）给到 szSize 引脚，功能块会发送字符串定义的大小加一的数据。可以把 LEN（字符串）给到 szSize 引脚，即为需要发送数据的大小。

19. Socket 自由协议通讯 TCP_Client 报错 TIME_OUT

TCP_Write 的 udiTimeOut 引脚单位为微秒，如下图所示设置为 1000us，受扫描周期影响，将无法正常工作连接到服务器，报错 TIME_OUT，可以改大 udiTimeOut 的值避免此错误。

表达式	类型	值	准备值
TCP_Client_0	NBS.TCP_Client		
SUPER^	LTrig		
udiTimeOut	UDINT	1000	
ipAddr	IP_ADDR		
uiPort	UINT	20000	
eError	ERROR	TIME_OUT	
xActive	BOOL	FALSE	
hConnection	DWORD	0	
TCP_Read_0	NBS.TCP_Read		
TCP_Write_0	NBS.TCP_Write		

```
1 //建立连接
2 ipAddr.sAddr :=strIP '192.168.88';
3 TCP_Client_0(
4   xEnable:=bEnable TRUE AND NOT TCP_Read_0.xError FALSE AND NOT TCP_Write_0.xError FALSE,
5   udiTimeOut 1000 :=uiTimeOut 1*1000,
6   ipAddr:=ipAddr,
7   uiPort[20000]:=uiPort[20000],
8   xDone=>,
9   xBusy=>,
10  xError=>,
11  eError=>,
12  xActive FALSE =>bConnect_OK FALSE,
13  hConnection=> );
```

20. Q0 控制器本体两个 COM 口通讯报错

Q0 的 COM1 主从都支持，COM2 只有支持主站，客户设置的 COM2 为从站所以报错，主站没有添加从站通道也会报错。将 COM1 改为从站，将 COM2 改为主站，并且添加从站通道可解决。

二、编程应用

1. 基础知识

- 1) 热复位时，Retain/Persistent 变量保持，普通变量会恢复初始值；
- 2) 冷复位时，Persistent 变量保持，Retain 变量/普通变量恢复初始值；
- 3) 模块的 IO 映射页面当前值为灰色，是因为右下角总是更新变量选的是使用父设备设置，可以选择启用 2 总是在总线任务中更新；

2. 从 PLC 中将文件存到 U 盘，但在电脑上看不到 U 盘中存入的文件

检查文件存储路径，不能直接放在 U 盘的根目录下，需将文件放入 U 盘下的 PORT0 文件夹下。

3. BOOL 型变量在线显示 “? ? ? ”

此种情况有以下几种可能：

- 1) 这个变量前边有数组越界产生（就是数组变量在使用时超出了声明的元素个数）；
- 2) 指针重叠；
- 3) 变量关联的地址区有重叠；

4. 新掉电保持模式对应控制器固件为多少？

Q0：应用程序版本 V1.04.03.01 以上

Q1：应用程序版本 V3.40.20.00 以上

Q0P/Q1P：全部支持

Q3P：应用程序版本 V1.00.01.14 以上

Q5P/Q7P/Q9P：应用程序版本 V1.01.00.02 以上

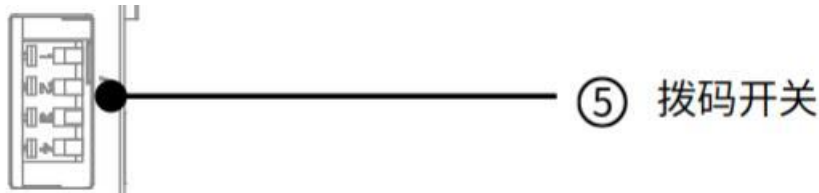
5. 字符串数据如何分配地址？

字符串数据类型分 STRING 和 WSTRING 类型：

- 1) STRING 类型默认是 80 个字符，所占地址长度为 81 个字节（字符+结束符），每个字符占 1 个字节，存储英文字符；
 - 2) WSTRING 类型默认也是 80 个字符，所占地址长度为 162 个字节（字符+结束符），每个字符占 2 个字节，存储中文字符；
- 在使用字符串类型变量时，可根据数据长度合理定义字符串数据长度，比如 10 个英文字符的字符串变量定义为 STRING(10)类型，所占地址长度为 11 个字节。

6. 控制器恢复默认 IP 方法

Q0：SW1 拨到左，SW2 拨到右，长按 FN 两秒；



Q1-322 版本：旋钮拨到 5，断电重启；

Q1-330+ 版本：旋钮拨到 5，按下绿色功能键，面板显示 0000，断电重启；

注意：恢复完成后，请将旋钮拨回 0；

Q5：按下绿色按钮 5 秒以上，断电重启；

Q0P/Q1P: 拨码开关拨向 FN 并保持 3s 以上触发恢复 IP, 完成后断电重启;

12	FN/RUN/STOP开关	FN/RUN/STOP switch	拨到FN为复位EtherNet通讯接口IP地址, 拨到RUN为运行控制器程序, 拨到STOP为停止控制器程序
----	---------------	--------------------	--

Q3P/Q5P/Q7P/Q9P: 拨码开关拨向 FN 并保持 3s 以上触发恢复 IP, 完成后断电重启;

10	STOP/RUN/FN 拨码开关	STOP/RUN/FN dial switch	功能拨码, 向左为控制器停止状态, 中间为运行状态, 向右拨至FN并停留3s为恢复Port1/port2/port4 IP默认值
----	------------------	-------------------------	--

7. 指针类型数据的大小

指针类型数据的大小在不通操作系统上运行结果不同 (同_XWORD/_XINT/_UXINT), 32 位系统上大小为 4 个字节, 64 位系统上大小为 8 个字节;

8. 十进制转化十六进制字符串

如下图所示, 把十进制转化为 16 进制字符串表示, 利用 Byte_To_HexString 函数实现, Byte_To_HexString 函数位于库 SM3_Shared 中。

表达式	类型	值	准备值	地址
v_by_data	BYTE	255		
v_str_data	STRING(3)	'FF'		


```

1  v_str_data := SM0.Byte_To_HexString(B:=v_by_data 255 );RETURN

```

9. 十进制转化 ASCII

可以利用联合体共用同一内存地址的特征, 声明十进制的数据类型和字符串类型, 向十进制数据类型中写入数据, 读取字符串类型即为对应的 ASCII 码。

表达式	类型	值
unToAscii	unToAscii	
bData	BYTE	65
AsciiData	STRING	'A'


```

1  unToAscii.bData := 65;RETURN

```

10. 设备系统信息

新版 HCP Works3 V1.3.0 软件双击 Device 选择系统信息栏, 可以读取/写入系统时间、时区、IP 地址等信息; 选择设备信息栏, 可以读取当前设备信息, 用来判断设备版本等。(当遇到弹窗提示需要升级固件时, 可以联系产品线技术人员升级设备固件)

The top screenshot shows the 'Device' configuration window with the 'System Settings' (系统设置) tab selected. It includes sections for 'Date/Time' (日期时间), 'Time Zone' (时区), and 'IP Settings' (IP设置). The bottom screenshot shows the 'Device Information' (设备信息) tab, displaying details like 'PLC Model' (PLC型号), 'System Version' (系统版本), and 'Application Version' (应用版本).

11. 掉电保持区不支持指针类型的变量

在掉电保持列表中不能使用 POINTER TO 数据类型的变量，地址会在程序重复下载的情况下会发生改变，指针指向的地址很可能不是之前的数据类型。

在保持区声明指针类型的变量时，编译后会报错。

The screenshot shows the 'Build' message window with a summary bar indicating '1 error, 0 warnings, 42 messages'. The message list shows an error: 'GVL.pByte: 掉电保持不支持指针类型' (GVL.pByte: Power-on memory retention does not support pointer type).

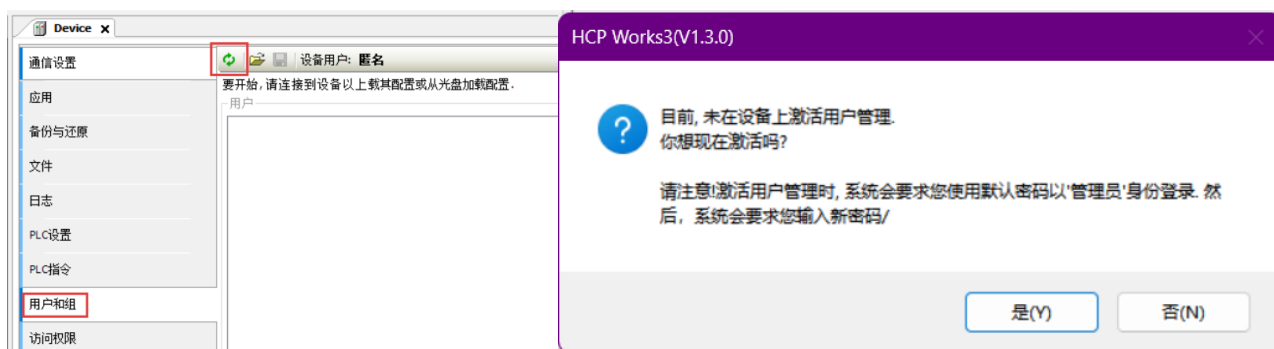
12. 文件操作怎么换行以及美元符号 '\$' 怎么输入？

写文件操作使用 `\r\n`，无法换行，CODESYS 中换行符为 `$L/$I` 或者 `$N/$n`。

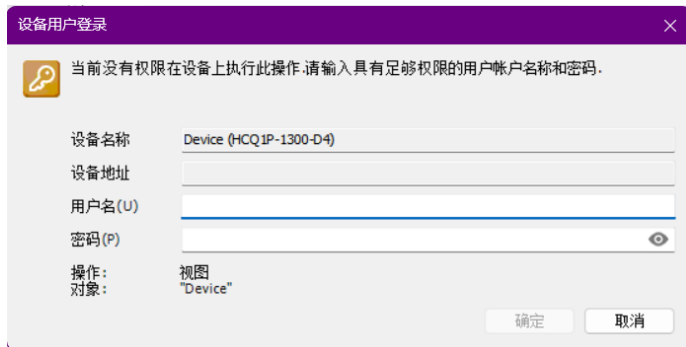
定义组合	解释
\$<两个十六进制数>	8位编码字符的十六进制表示。
\$\$	美元符号
\$'	单直引号
\$L或\$I	换行
\$N 或\$n	新行
\$P 或\$p	换页
\$R 或\$r	回车
\$T 或\$t	标签

13. 控制器加密

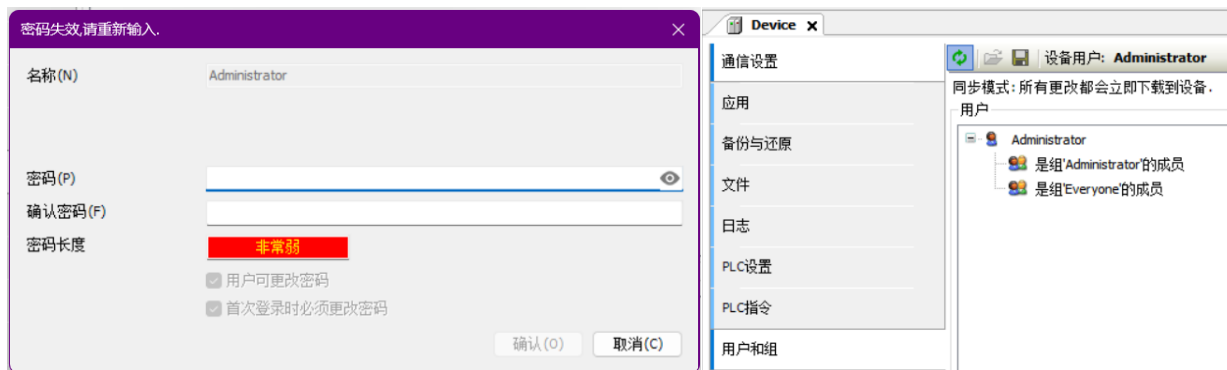
点击 Device 下的用户和组选项中左上角的绿色图标，弹出窗口确认是否激活用户管理。



点击“是”弹出设备用户登录窗口，默认用户名和密码都是 Administrator。

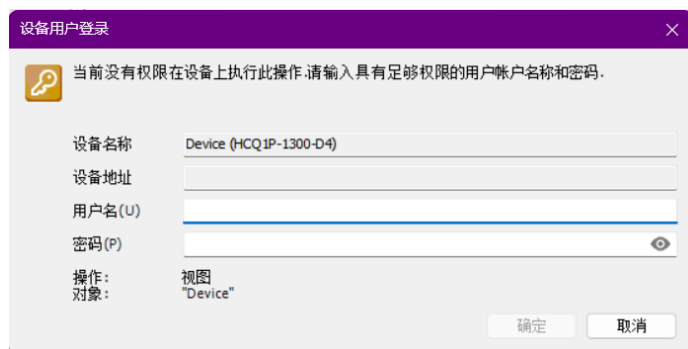


输入用户名密码后，弹出密码失效对话框，重新设置密码，**请牢记此密码!!!** 因为后续打开工程扫描设备登录都需要登录用户名密码。

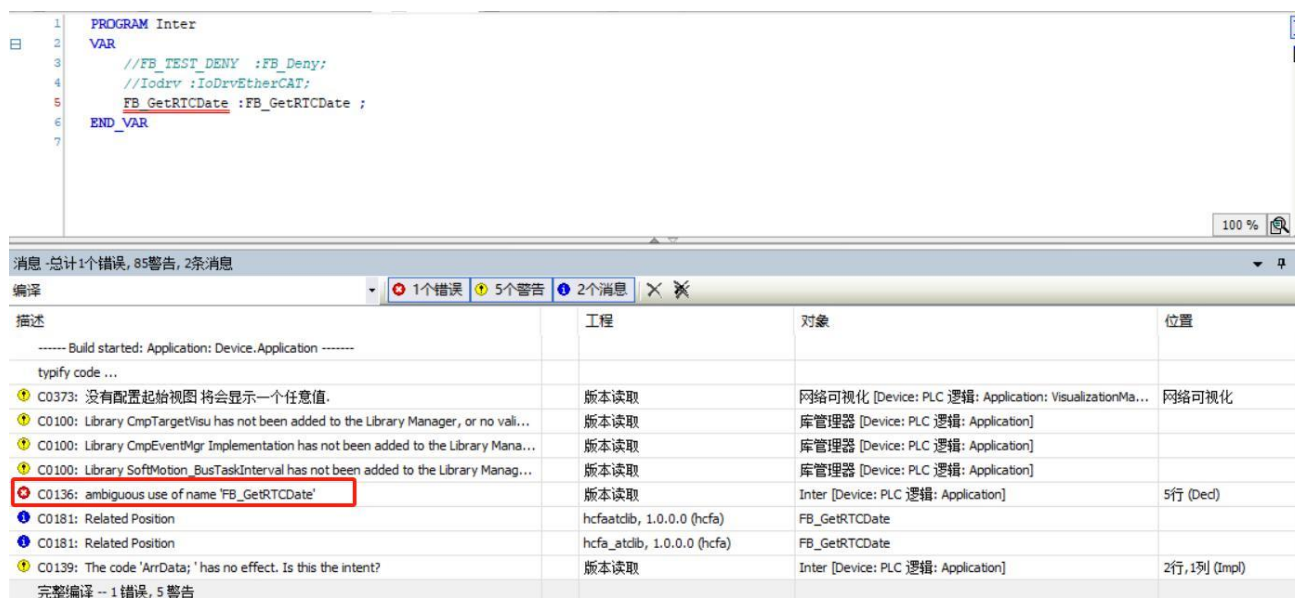


14. 控制器解密

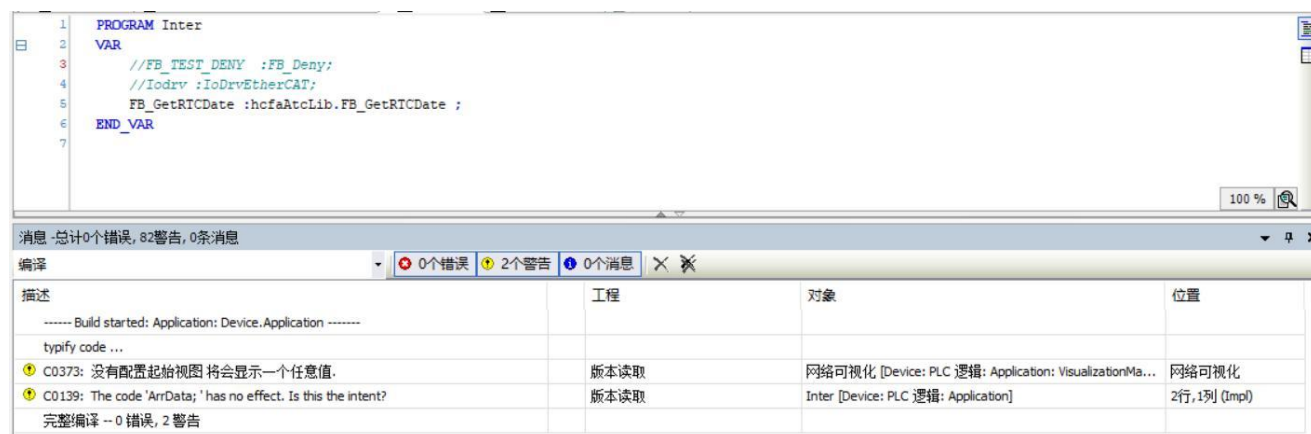
在 Device 上右键选择“复位原点设备[device]”，弹出设备用户登录窗口，登录成功后会取消控制器加密。



15. 声明功能块后编译报错 C0136: ambiguous use of name “功能块名字”

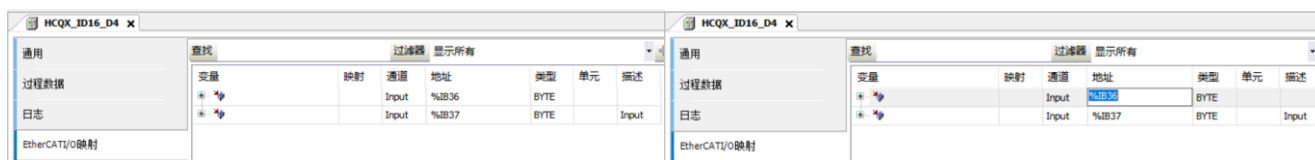


这是因为这个功能块名称在两个不同的库中存在，只需要删掉一个库或者在功能块前面添加库的 命名空间即可。



16. IO 地址强制与解除

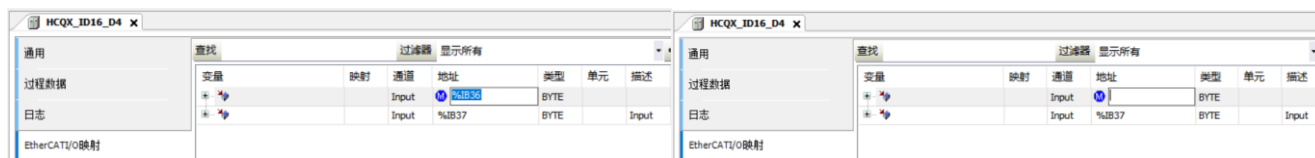
在模块 IO 映射页面双击地址。



然后鼠标点击空白处，就会强制这个通道的地址。



再次双击这个地址，然后删除地址。

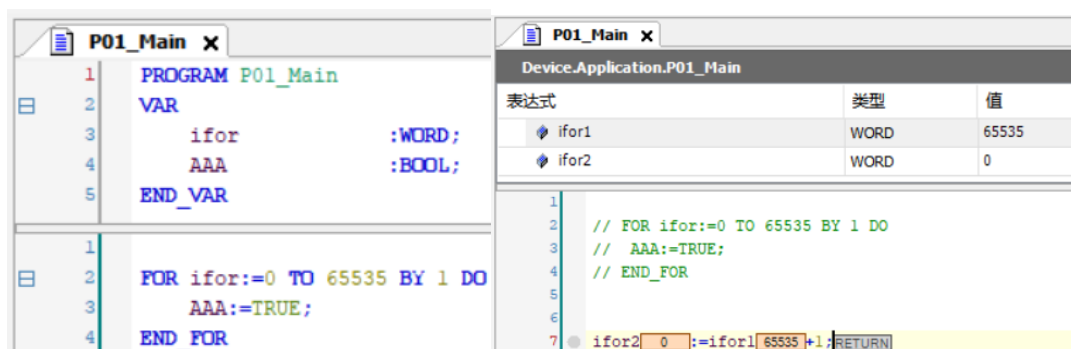


然后鼠标点击空白处，就会解除强制地址。

注意事项：请谨慎操作强制地址，有些时候可能影响总线轴，可以预留部分地址。

17. 为什么 FOR 循环变量的目标值设置成数据类型上限后导致死循环？

如下图所示，ifor 的目标值为 65535，数据类型为 word。FOR 循环的循环结束条件是目标值+1，所以 ifor 的结束条件就是大于等于目标值+1（65535+1），此时 ifor 就会变为 0（word 的数据范围是 0~65535）。所以这个循环就不会结束，会一致刷新，导致死循环。



小心！

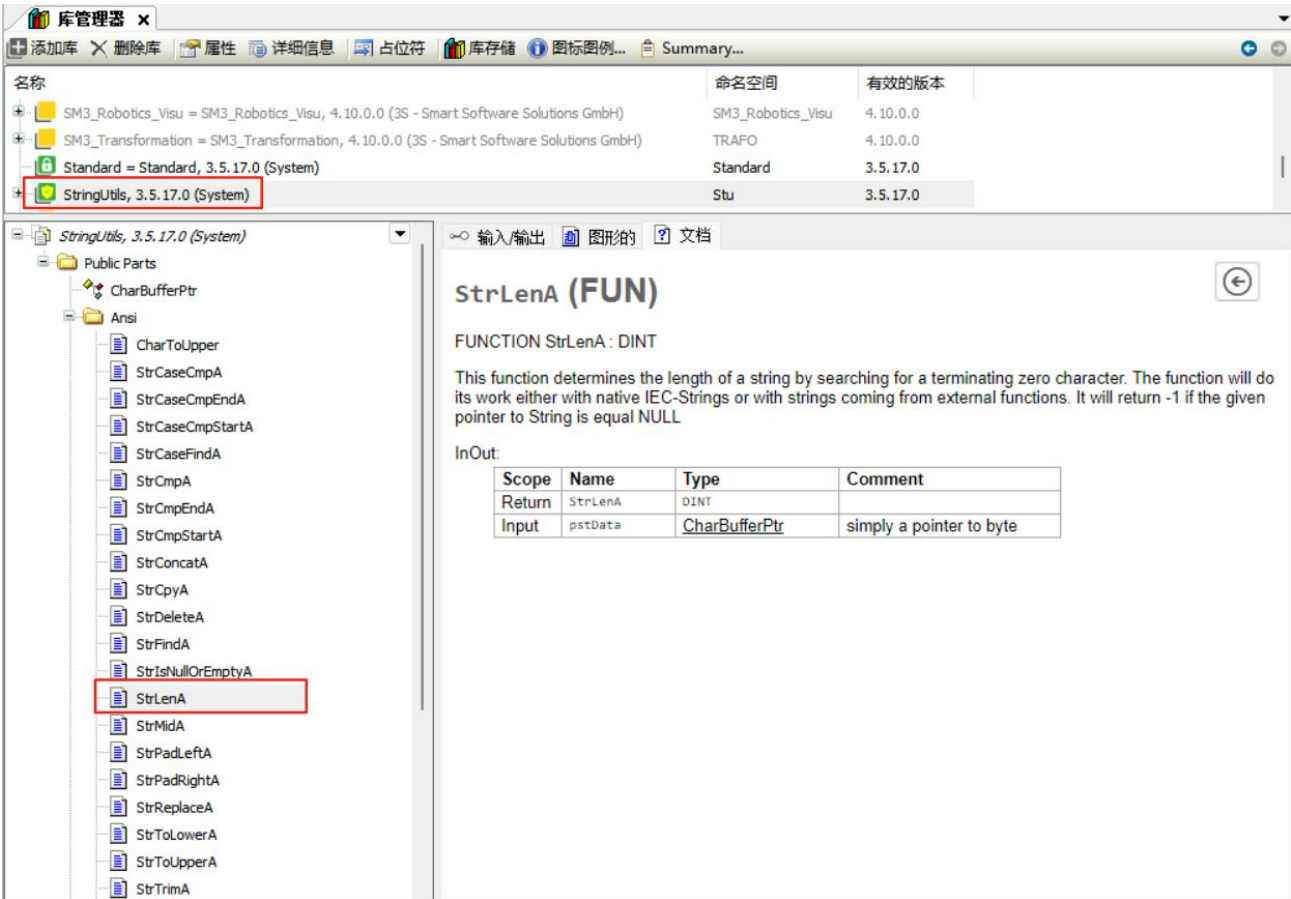
最终值<结束值>的值可能与循环变量的数据类型上限不同。

如果循环变量的结束值等于循环变量的数据类型上限，则会导致无限循环。例如，如果 iCounter 的数据类型为 SINT 并且 <end value> 等于 127，则无限循环将导致上述示例，因为数据类型 SINT 的上限值为 127。

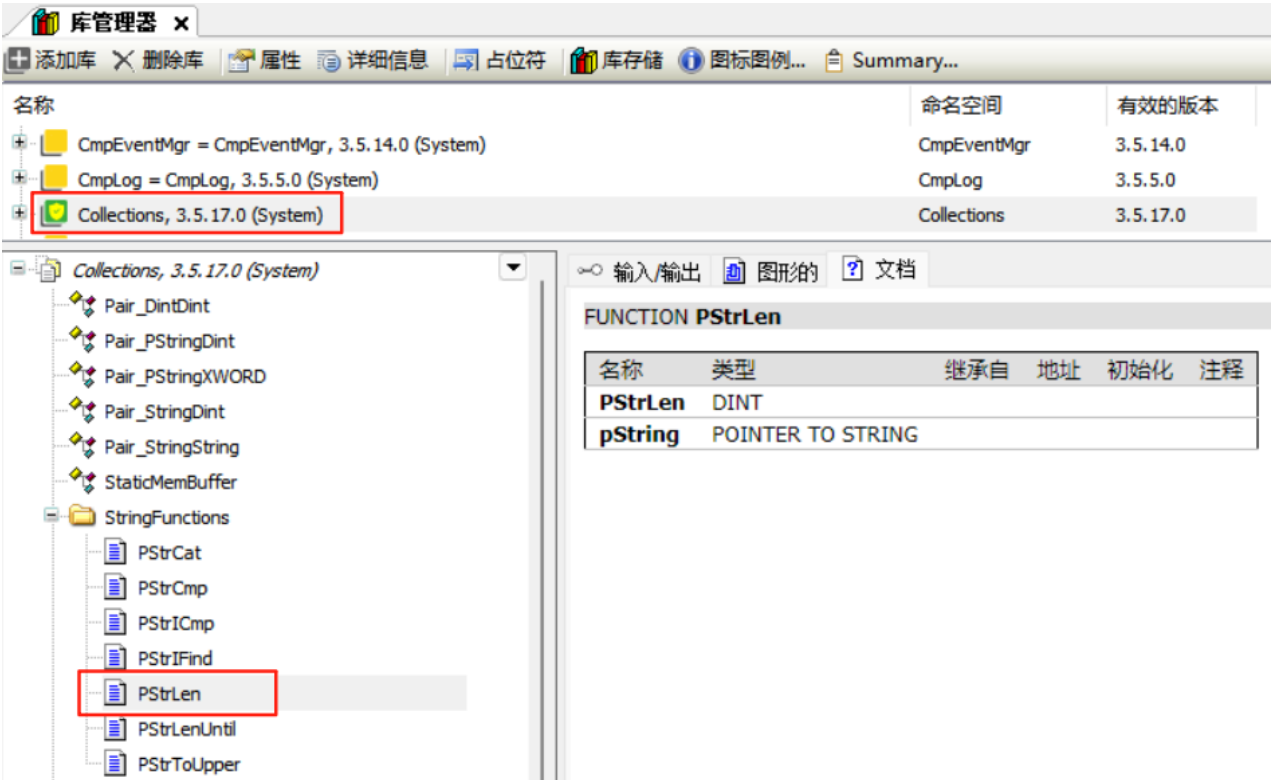
18. 如何获取字符个数超过 255 的字符串长度？

Standard 库中 LEN 函数只能处理 255 长度以内的字符串，要获取超过 255 个字符的字符串长度，可以尝试以下方法：

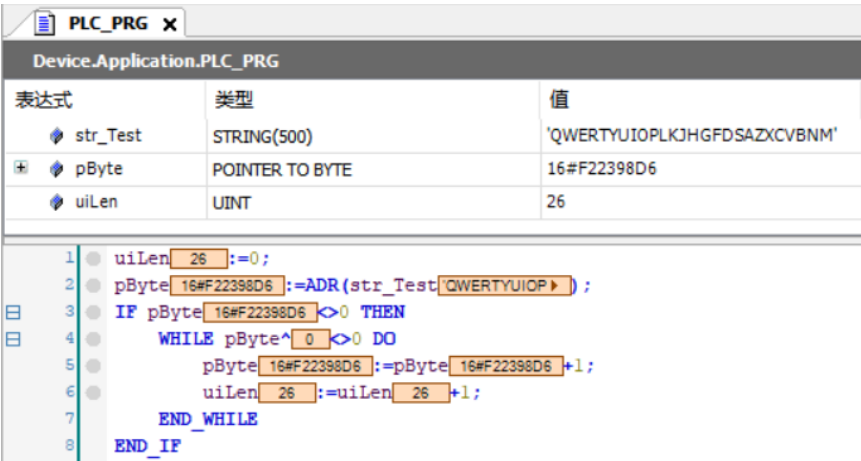
- 1) StringUtils 库中的 StrLenA 函数，CharBufferPtr 类型为 POINTER TO BYTE，使用时通过 ADR(字符串)获取字符串的指针给到 CharBufferPtr 即可，函数返回值即为字符串数据的长度。如果给空指针，则返回值为-1。



2) Collectiobs 库中的 PstrLen 函数，pSting 为字符串地址，通过 ADR 获取指针，函数返回值为字符串长度，空指针会抛异常。

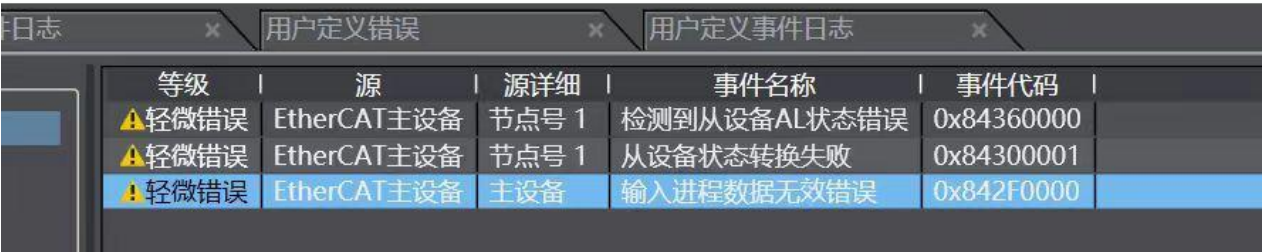
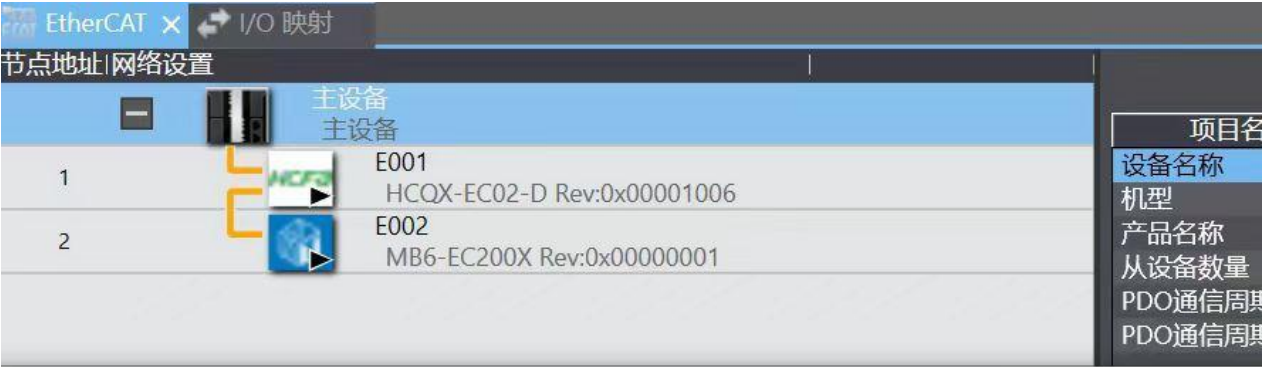


3) 通过循环寻找字符串的结束符来实现，字符串的结束符（空字符，0），实现方式如下：

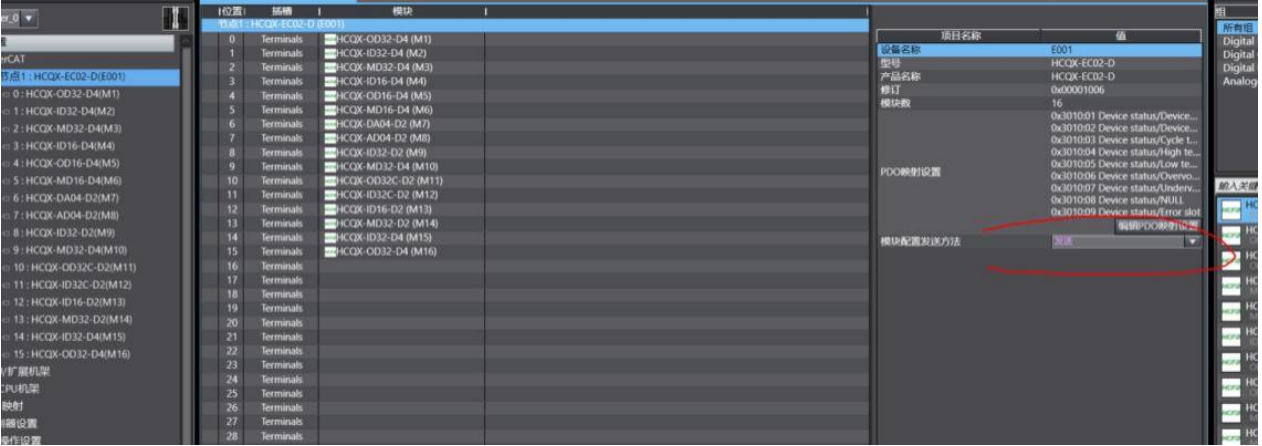


19. 欧姆龙控制器配 EC02 报错检测到 AL 状态错误

报错如下：

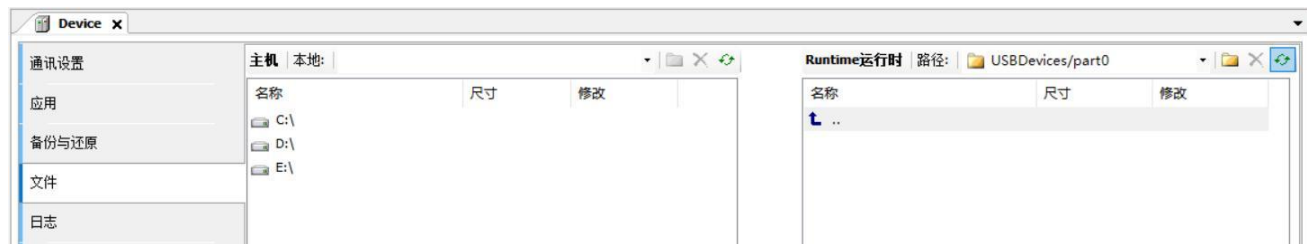


将模块配置发送方法配置为发送即可



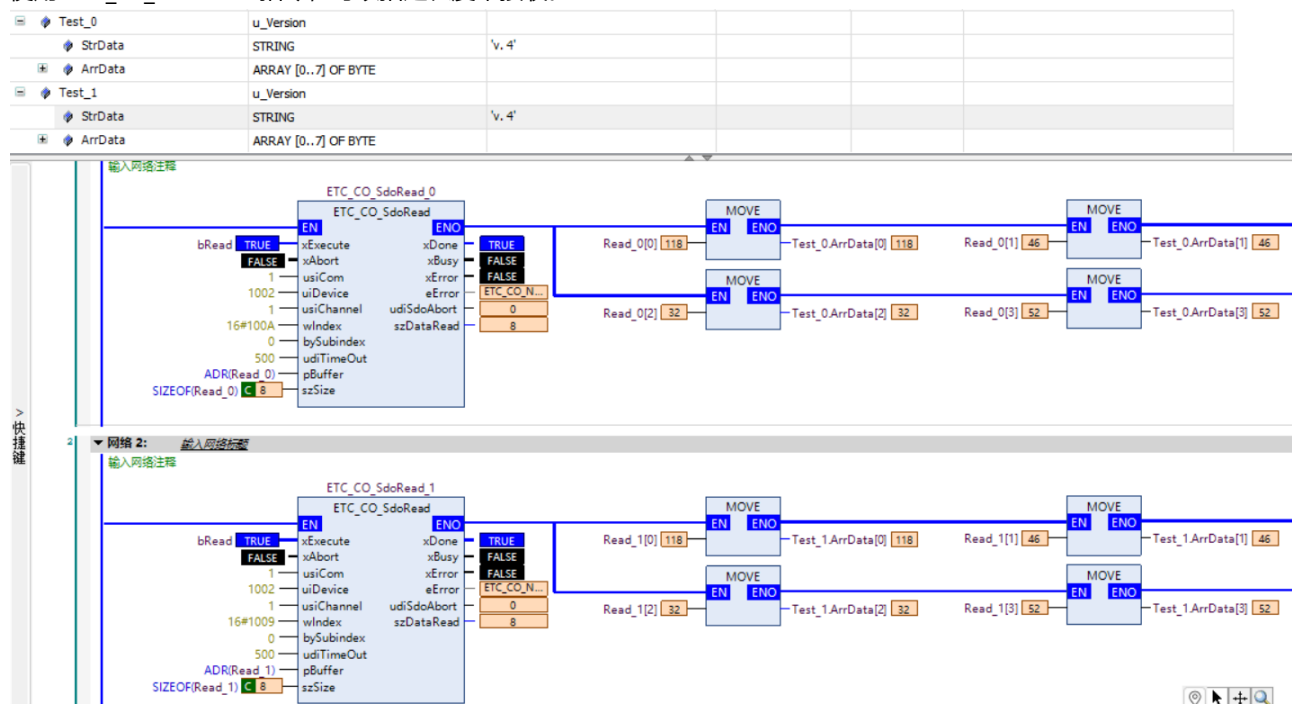
20. 拔掉 U 盘后 USBDevices 文件夹依然存在

拔掉 U 盘后，在文件中依然可以看到 USBDevices 文件夹，进入 part0 后不显示任何东西，再次插入 U 盘，PLC 无法正常识别。这是因为程序中调用了打开文件操作，未经过关闭操作就拔掉 U 盘，Codesys 会一直访问这个文件路径，导致无法正常识别 U 盘，只能断电重启。



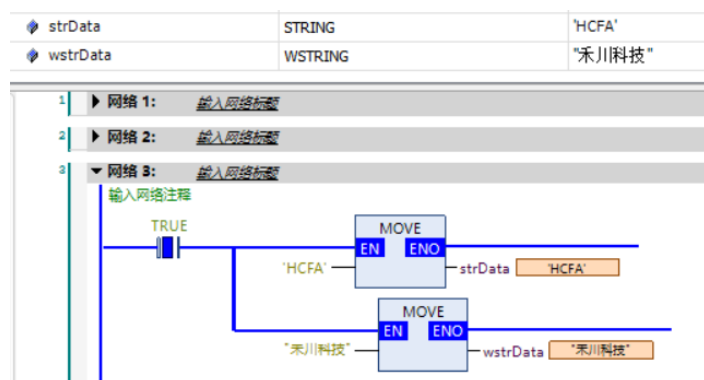
21. 如何读取超过 SMC3_ETC_ReadParameter CoE 指令 DWORD 类型数据长度的 SDO?

使用 ETC_co_sdoRead 指令，可以指定长度来接收。



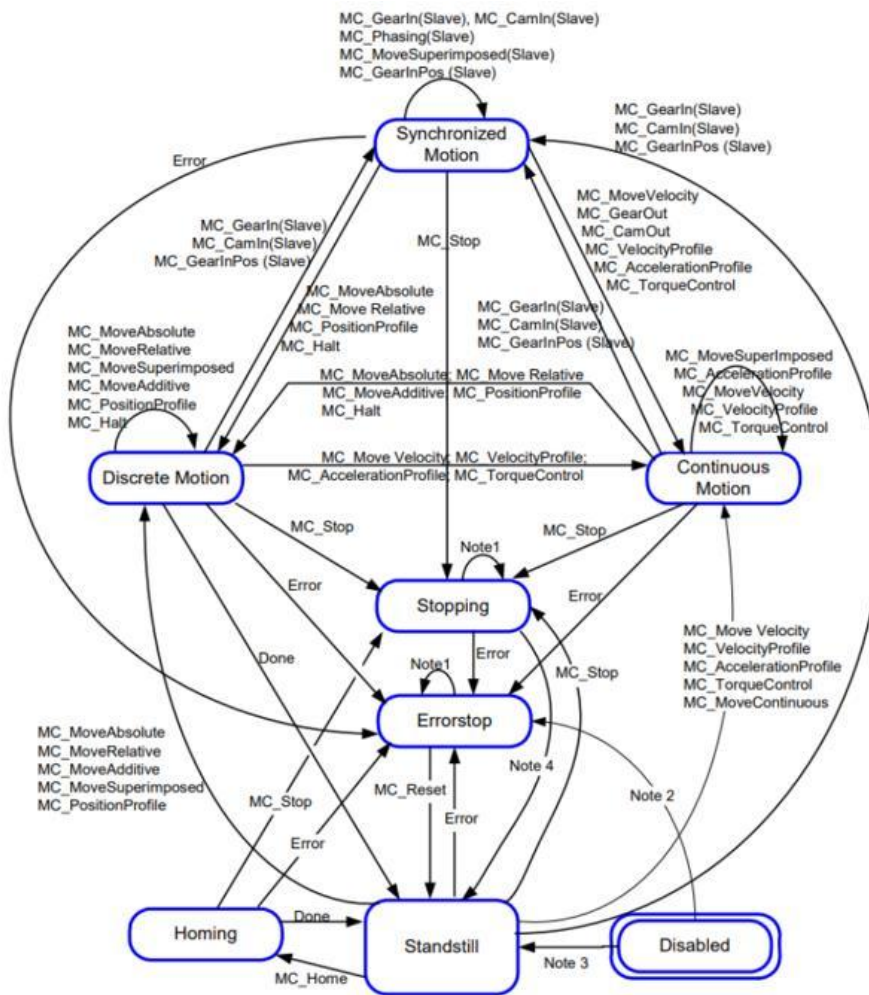
22. STRING 变量可以直接通过 MOVE 指令赋值中文吗?

STRING 变量不能被赋值中文，中文需要使用 Wstring 类型变量。中文赋值需要使用双引号括上。



三、运动控制

1. 轴控状态机及功能块触发示例



//使能

```
MC_Power_0(
    Axis:=Axis_0 ,
    Enable:=TRUE ,           //此引脚是给MC_Power指令使能，所以直接置TRUE就可以
    bRegulatorOn:=bPower ,
    bDriveStart:=bPower ,
    Status=>bPowerOK ,
    bRegulatorRealState=> ,
    bDriveStartRealState=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

//停止—需要防止轴处于未使能 (Power_off)、错误状态 (Errorstop) 下被触发

```
MC_Stop_0(
    Axis:=Axis_0 ,
    Execute:=bStop AND Axis_0.nAxisState>1 ,
    Deceleration:=10000 ,
    Jerk:= ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```


//复位—需要轴处于错误状态，并且通讯状态正常

```
MC_Reset_0(
    Axis:=Axis_0 ,
    Execute:=bReset AND Axis_0.nAxisState=1 AND Axis_0.wCommunicationState=100 ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

//驱动器复位—当总线异常时可以执行复位驱动器

```
SMC3_ReinitDrive_0(
    Axis:=Axis_0 ,
    bExecute:=bReset AND Axis_0.wCommunicationState<>100 ,
    bVirtual:= ,
    bDone=> ,
    bBusy=> ,
    bError=> ,
    nErrorID=> );
```

//回原—需要防止轴处于未使能（Power_off）、错误状态（Errorstop）、停止状态（Stopping）下被触发

```
MC_Home_0(
    Axis:=Axis_0 ,
    Execute:=bHome AND Axis_0.nAxisState>2 ,
    Position:=0 ,
    Done=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
```

//暂停—需要防止轴处于未使能（Power_off）、错误状态（Errorstop）、停止状态（Stopping）下被触发

```
MC_Halt_0(
    Axis:=Axis_0 ,
    Execute:=bHalt AND Axis_0.nAxisState>2 ,
    Deceleration:=10000 ,
    Jerk:= ,
    Done=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
```

//点动—需要防止轴处于未使能（Power_off）、错误状态（Errorstop）、停止状态（Stopping）、回原（Homing）下被触发

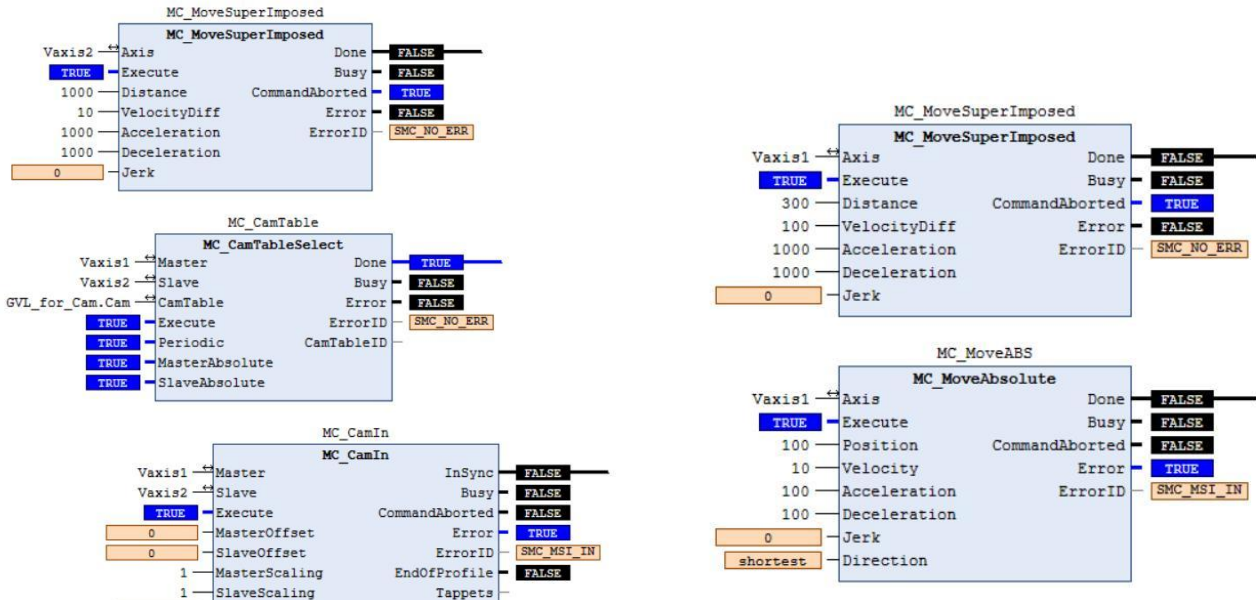
```
MC_Jog_0(
    Axis:=Axis_0 ,
    JogForward:=bJogF AND Axis_0.nAxisState>2 AND Axis_0.nAxisState<>7 ,
    JogBackward:=bJogB AND Axis_0.nAxisState>2 AND Axis_0.nAxisState<>7 ,
    Velocity:=rJogVel ,
    Acceleration:=1000 ,
    Deceleration:=1000 ,
    Jerk:= ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorId=> );
```

MC_MoveAbsolute、MC_MoveRelative、MC_MoveSuperImpsed、MC_MoveAdditive 指令与 MC_Jog 指令一样,都需要防止轴处于未使能（Power_off）、错误状态（Errorstop）、停止状态（Stopping）、回原（Homing）下被触发。除此之外，

MC_MoveSuperImpsed 指令在使用时应该放置在运动功能块（定位、齿轮、凸轮等）之后，否则可能会导致执行不正确。

2. 功能块报错 'SMC_MSI_INVALID_EXECUTION_ORDER'

注意 MC_MoveSuperImposed 指令的摆放位置，如果要补偿从轴，需要把 MC_MoveSuperImposed 指令摆放在耦合功能块后边，否则肯能会导致指令执行错误。



3. 转矩模式下轴在动作，轴状态显示 standstill

转矩模式下，伺服没有类似状态机的概念，设定转矩和速度限制不为 0，轴使能后即可动作，动作过程中一直为 standstill。

EtherCAT 总线伺服 PDO 一般有两个最大速度参数：16#607F 和 16#6080，这两个参数单位不一样，一个是指令单位，一个是电机转速。最终限制速度以最小值为准。

Object 6080h:最大电机转速		Object 607Fh:最大转速限制	
对象描述		对象入口描述	
属性	值	属性	值
索引	6080h	子索引	00h
名称	最大电机转速	访问属性	rw
数据结构	Variable	PDO 映射类型	TPDO
数据类型	Unsigned32	数据范围	0~4294967295
操作模式	ALL	默认值	6500

6080h 设置最大电机转速，用以保护电机，所有模式下均有效；单位：Rpm/min

- 速度模式下，最大转速限制以 607Fh 和 6080h 两者中较小值为准
- 转矩模式下，最大转速限制以 607Fh、6080h、内部速度限制(P03.27,P03.28)三者中的较小值为准
- 位置模式下，PP 模式最大转速限制以 607Fh 和 6080h 两者中较小值为准

CSP 模式最大转速限制以 6080h 为准，伺服内部功能码 P09.17 的右起第一位可选择设定 6080h 限制与否：

- CSP 模式，当 P09.17 右起第一位设 0，6080h 不做速度限制，超过最大转速会报 Er.78 指令异常故障
- CSP 模式，当 P09.17 右起第一位设 1，电机运行最大转速按照 6080h 设定值

4. 使用 MC_Home 指令回零，动作异常

首先要确认伺服驱动器支持所选的回零方式；其次检查正负限位与原点信号是否接混，以及常开常闭信号确认（一般限位开关用常闭信号，原点用常开信号）；最后检查伺服驱动器的参数配置是否正确，包括限位和原点信号的电平有效参数。

5. 执行 MC_Home 指令后，16#6064 反馈不会变为 0

执行 MC_SetPosition 后，轴结构体下的 fOffsetposition 会发生改变，在使用 MC_Home 回原的时候，16#607C(Home offset)会对应变换成 fOffsetposition 的偏移值，16#6064 会改为 16#607C 的值，此时 402 轴 fActpositon/fSetposition 显示为 0。

6. 上升沿触发 MC_Home 指令报错 SMC_FB_WASNT_CALLED_DURING_MOTION

此问题跟 softmotion 的版本有关：

- 1) 在 softmotion 版本 4.11.0.0 版本及以下，边沿触发 MC_HOME 功能块正常执；
- 2) 在 softmotion 版本 4.12.0.0 及以上边沿触发 MC_HOME 功能块时会报错，需要在回原完成前一直保持触发信号存在就可以

避免此问题。

7. 控制器带轴能正常使能和复位，但是不能点动、定位、匀速等操作。

首先检查运动控制指令是否放在总线任务下调用；其次如果是双主站的控制器，要根据轴所连接的端口下的主站任务中调用，不能两个主站交叉调用。

8. 如何在程序中修改轴的减速比？

通过功能块 SMC_ChangeGearingRatio 实现此功能，使用注意事项如下：

- 1) 必须是断使能状态下才能执行 SMC_ChangeGearingRatio 功能块；
- 2) 执行完 SMC_ChangeGearingRatio 后立即执行 SMC3_ReinitDrive；
- 3) 执行完 SMC3_ReinitDrive，需要等 EtherCAT 总线恢复运行后再上使能；

```
//轴断使能后执行SMC_ChangeGearingRatio
smcChangeGearingRatio(
    Axis:=Axis2 ,
    bExecute:=bChange AND NOT bPower AND NOT mcPower.Status , //必须是断使能状态下才能执行
    dwRatioTechUnitsDenom:=Axis2.dwRatioTechUnitsDenom , //齿轮比分母，根据触摸屏参数修改此值
    iRatioTechUnitsNum:=Axis2.iRatioTechUnitsNum , //齿轮比分子，根据触摸屏参数修改此值
    fPositionPeriod:=Axis2.fPositionPeriod ,
    iMovementType:=Axis2.iMovementType ,
    bDone=> ,
    bBusy=> ,
    bError=> ,
    nErrorID=> );
IF bChange AND smcChangeGearingRatio.bDone THEN
    bChange:=FALSE;
END_IF

//执行完SMC_ChangeGearingRatio后立即执行SMC3_ReinitDrive
smc3ReinitDrive(
    Axis:=Axis2 ,
    bExecute:=smcChangeGearingRatio.bDone ,
    bVirtual:= ,
    bDone=> ,
    bBusy=> ,
    bError=> ,
    nErrorID=> );
//执行完SMC3_ReinitDrive, EtherCAT总线恢复运行后再上使能
```

9. 如何设置轴的软限位？

可以直接给轴结构中软限位接口赋值触发轴的软限位功能：

Axis.bSWLimitEnable——激活软限位；

Axis.fSWLimitPositive——正向软限位；

Axis.fSWLimitNegative——正向软限位；

10. 开启软限位后，定位超过目标位置，报错触发软限位

开启软限位后，会以动态限制的参数计算路径能否在范围内停止，如果不能在范围内停止，则会报错触发软限位，且以动态限制的减速度进行减速停止，实际可能看到轴位置未到达软限位位置。把动态参数设置的比程序中使用的参数大即可。

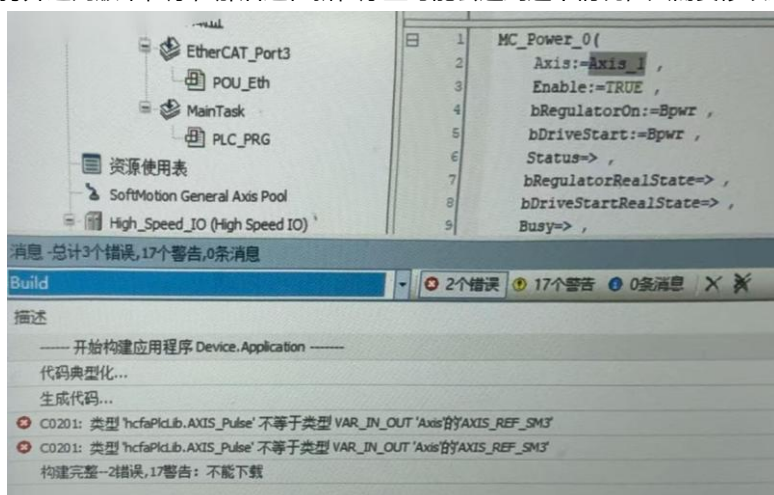
轴类型与限制			
☐ 虚轴模式			
☐ 模态			
☒ 线性			
软件限位			
☒ 激活	负向 [u]:	-10	
	正向 [u]:	1000.0	
软件错误处理			
	减速 [u/s ²]:	0	
	最大距离 [u]	0	
动态限制			
速度 [u/s]:	加速度 [u/s ²]:	减速度 [u/s ²]:	加加速度 [u/s ³]:
30	1000	1000	10000

11. 双主站控制器在使用 Codesys 官方软件时添加 Port4 口后，Port3 口丢同步

这是因为 Codesys 官方软件在添加第二个 EtherCAT 主站时不会自动创建 EtherCAT 任务，如果不手动添加 EtherCAT 任务就会挤掉先添加的 Port3 端口 EtherCAT 任务。

12. HCP Works3 V1.3.0 版本软件新建脉冲轴，报错轴类型不匹配

这个版本软件的 Softmotion 版本应该选择 4.10.0，如果电脑里安装有高版本的 Codesys 软件，并且使用 HCP Works3 V1.3.0 软件打开过高版本程序，那后边在新程序里可能会遇到这个情况，只需要修改对应 Softmotion 版本就好了。



13. SV730W 驱动只有第一个轴的转矩可以写入，其它轴写不进去

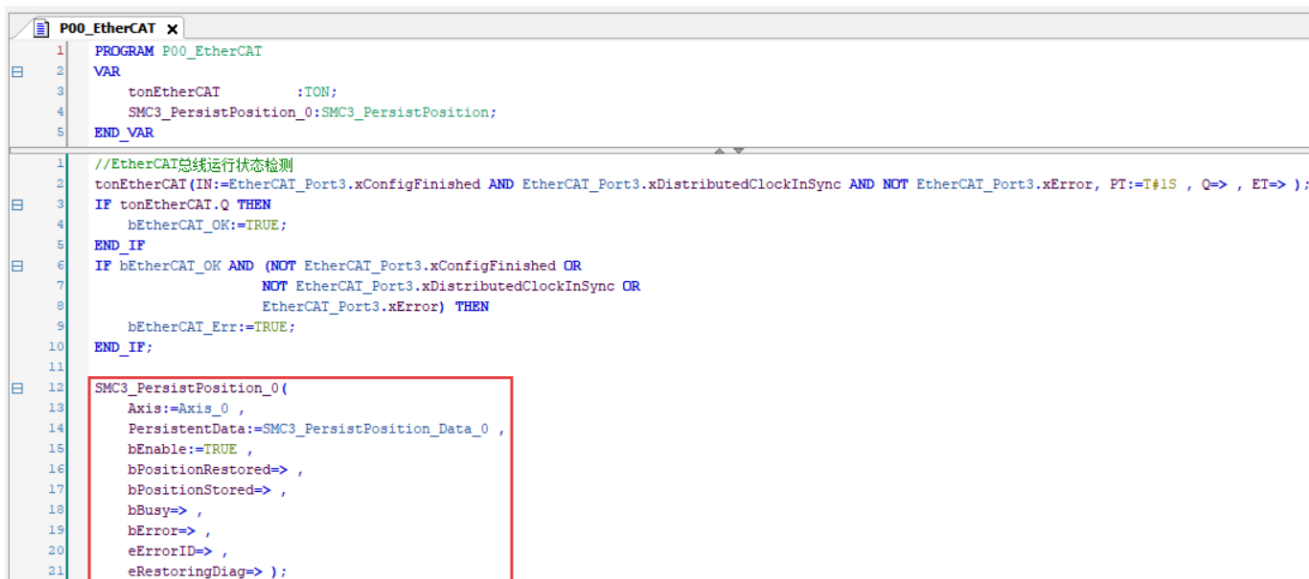
SV730W 的 XML 默认没有配置 16#6071、16#607F、16#6080 等参数，需要手动配置，但需要注意，四个轴的 PDO 地址是不同的，正确配置如下：

名称	类型	索引
☒ 16#1600 Outputs		
ControlWord	UINT	16#6040:00
Modes of Operation	SINT	16#6060:00
Target position	DINT	16#607A:00
Target velocity	DINT	16#60FF:00
Target torque	INT	16#6071:00
Max motor speed	UDINT	16#6080:00
☒ 16#1610 Outputs		
ControlWord	UINT	16#6840:00
Modes of Operation	SINT	16#6860:00
Target position	DINT	16#687A:00
Target velocity	DINT	16#68FF:00
Target torque	INT	16#6871:00
Max motor speed	UDINT	16#6880:00
☒ 16#1620 Outputs		
ControlWord	UINT	16#7040:00
Modes of Operation	SINT	16#7060:00
Target position	DINT	16#707A:00
Target velocity	DINT	16#70FF:00
Target torque	INT	16#7071:00
Max motor speed	UDINT	16#7080:00
☒ 16#1630 Outputs		
ControlWord	UINT	16#7840:00
Modes of Operation	SINT	16#7860:00
Target position	DINT	16#787A:00
Target velocity	DINT	16#78FF:00
Target torque	INT	16#7871:00
Max motor speed	UDINT	16#7880:00

14. 绝对值电机线性轴位置溢出后，断电重启会丢失位置

这种情况应该在总线起来后的而第一时间执行 SMC3_PersistPosition 功能块，并且要给 PersistentData 引脚赋一个掉电保持型的 SMC3_PersistPosition_Data 数据类型变量。

```
VAR_GLOBAL RETAIN PERSISTENT
    SMC3_PersistPosition_Data_0 AT $MB131072 : SMC3_PersistPosition_Data;
END_VAR
```



注意：

- 1) 功能块仅用于实轴，上电后必须优先执行 SMC3_PersistPosition 功能块，功能块的 Enable 引脚建议默认为 TRUE。

2) 功能块上电恢复位置数据时，轴的缩放比例等参数必须和断电前一致。

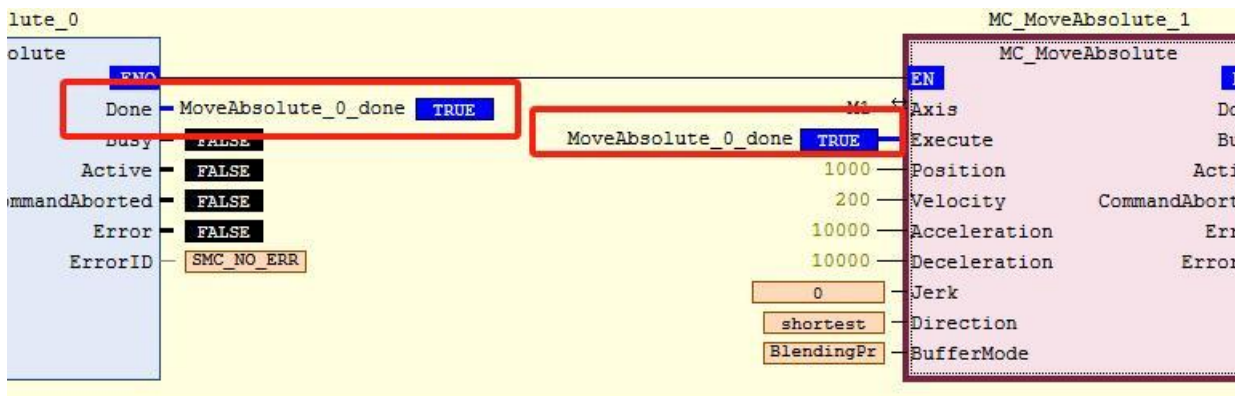
15. 伺服转矩单位不同

X 系列伺服转矩单位为 Nm，Y7 为千分比。这是因为 X 系列伺服会自动把额定转矩乘以一千写入到 6076（转矩比率）中，Y7 是不会写入这个值的。

如果 6076 有值，会将程序中给定的转矩值经过转化，给到 6071（目标转矩），如果 6076 为 0，程序中给定的转矩的单位为千分比。

16. BufferMode 设置了缓存，但是运行还是减速到零才触发第二段运动

设置了缓存模式，但是在上一段运动结束才触发第二段运动是不会进行缓存的，只有在第一段运动过程中触发别的功能块才会缓存运动指令。



17. 映射原点信号，回原时报错触发硬限位

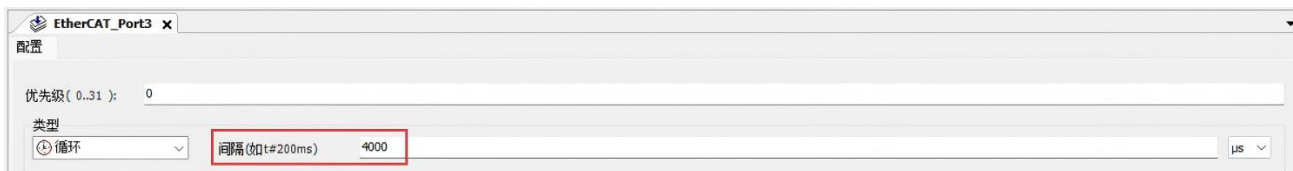
不要在驱动 IO 映射中去映射 16#60FD，想要读取限位、原点、探针等输入信号可以直接读取 Axis.in.dwDigitalInputs。

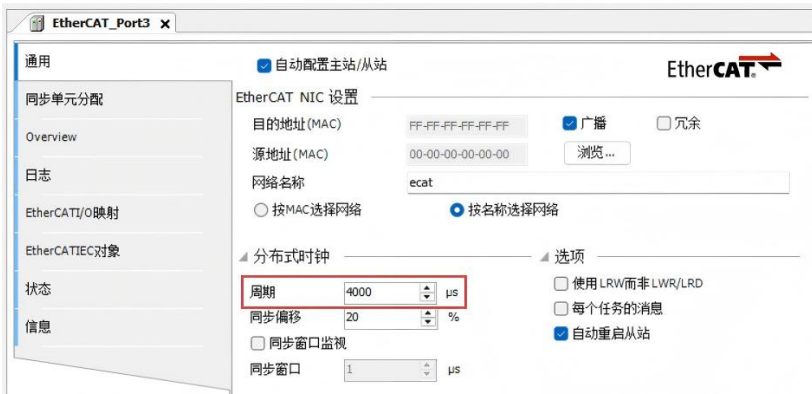
以 Y7S 为例：

```
b负限位 :=Axis_0.in.dwDigitalInputs.0;
b正限位 :=Axis_0.in.dwDigitalInputs.1;
b原点   :=Axis_0.in.dwDigitalInputs.2;
b探针1  :=Axis_0.in.dwDigitalInputs.11;
b探针2  :=Axis_0.in.dwDigitalInputs.12;
```

18. 改变 EtherCAT 任务周期后，轴运行速度不对

检查任务设置下 EtherCAT 任务周期时间是否与 EtherCAT 主站的周期时间一致，不一致就可能导致速度不对。





19. 切换控制模式后报错 SMC_MOVING_WITHOUT_ACTIVE_MOVEMENT

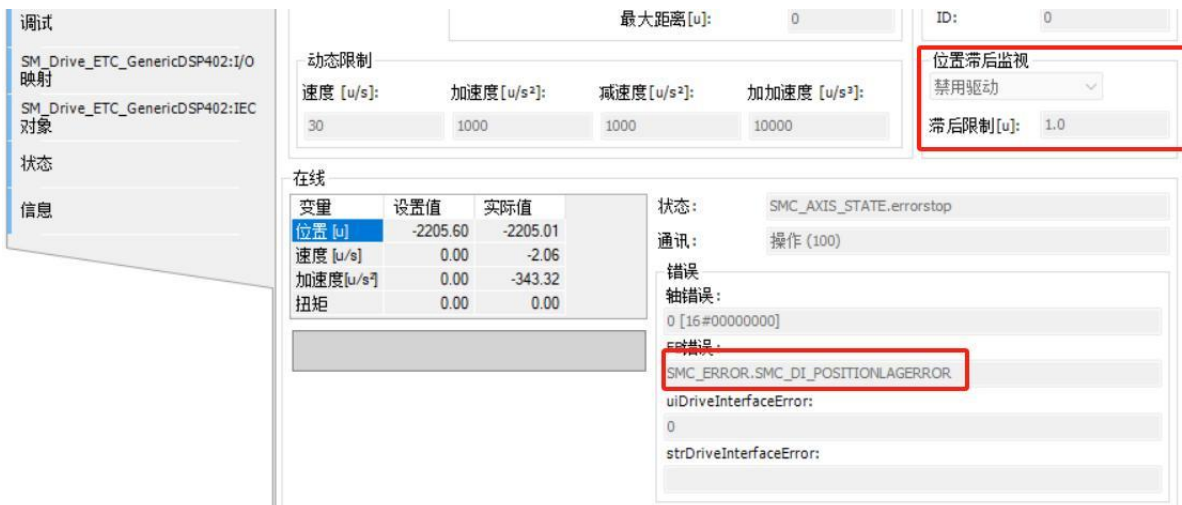
从转矩切换到位置模式时，SMC_SetControllerMode 的 Done 信号输出后，没有函数块继续提供设置位置的适当值，则轴立即停止并报错，需要在 Done 输出的同一周期使用别的功能块来接管轴的控制。如 Halt、Stop 等。（和 Softmotion 有关，4.4.0.2 的 softmotion 不会有此问题，4.10.0.0 的 softmotion 不接管会报错）。

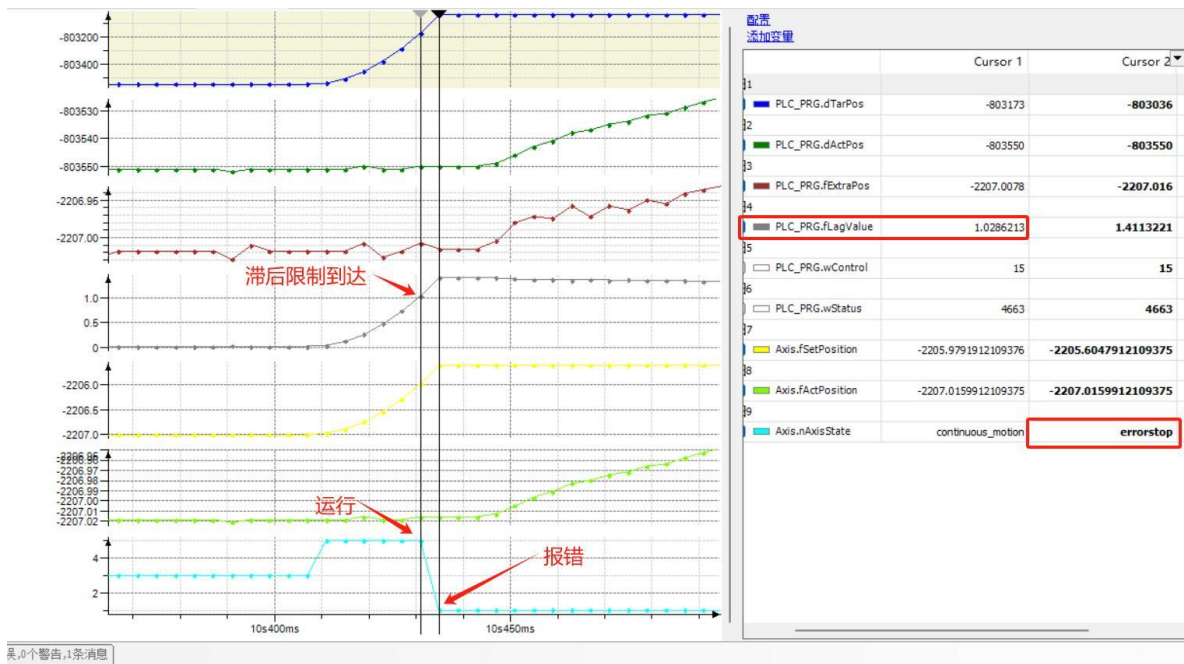
```
SMC_SetControllerMode_0(
    Axis:=Axis_0 ,
    bExecute:= ,
    nControllerMode:=3 ,
    bDone=> ,
    bBusy=> ,
    bError=> ,
    nErrorID=> );

MC_Stop_0(
    Axis:=Axis_0 ,
    Execute:=SMC_SetControllerMode_0.bDone ,
    Deceleration:=10000 ,
    Jerk:= ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

20. 轴报错 SMC_DI_POSITIONLAGERROR 位置滞后错误

这个是由于使能了位置滞后监视，然后额定位置和补偿的实际位置之间的差值超过了滞后限制 ($\text{abs}(\text{fSetPositon}-\text{fActPositon}) > \text{滞后限制的值}$)。





位置滞后有关说明如下：（翻译问题，导致各软件版本名称可能不同，如停用为取消激活、去使能等）

“位置滞后监督”

定义系统的检测滞后误差后的反应。当额定位置和补偿的实际位置之间的差值超过滞后限制就会产生滞后误差。实际位置的计算公式：

Extrapolated actual position := actual position + actual velocity * cycle time * Axis.fSetActTimeLagCycles

这个值等于轴的结构体中的fSetActTimeLagCycles 周期（“功能块 AXIS_REF_SM3”）补偿用于因为通讯而导致设置值与当前值之间的偏差。

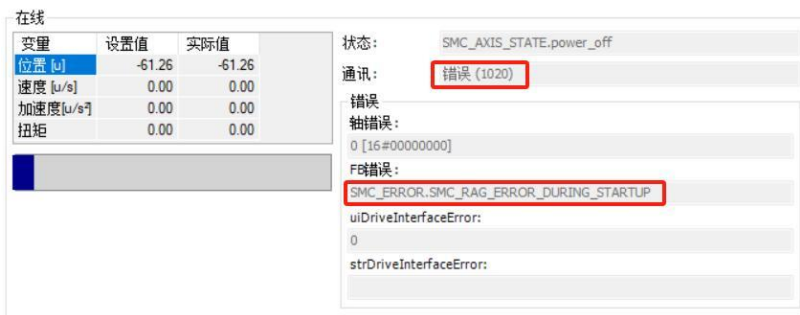
检测到滞后误差可能出现的反应是：

- “停用”：没有任何反应，因为滞后监督被取消。
- “禁用驱动器”：位bRegulatorOn被强制为 FALSE (与 MC_Power 输入比较)，这导致驱动器被禁用 (当然依据驱动器减速度实现方式)。
- “执行快速停止”：位bDriveStart被强制为 FALSE (与 MC_Power 输入比较)，这导致驱动器快速停止。
- “停留当前状态”：驱动器仍然位于当前状态，但是停止所有运动。

“滞后限制”：定义滞后限制。

21. 轴启动过程中卡状态在 1020

轴报错 SMC_RAG_ERROR_DURING_STARTUP，通讯状态卡在 20 一段时间（超时）后，变为 1020，在 SMC_ERROR 错误描述中此错误为“Error at startup of the axis group”（在轴组启动时报错），这个报错时由于轴在重启过程中有过多 SDO 操作导致的（如 MC_ReadAxisError、MC_ReadParameter、MC_WriteParameter 等）。报错界面如下：



按照状态机说明，预运行才能进行 SDO 操作，预运行到安全运行是主站通过 SDO 对从站进行操作，在主站启动过程中，建议不要运行 SDO 相关操作，以免影响总线启动（如诊断功能块、CRC、SDO 读写等导致轴启动异常等）。

状态和状态转化	操作
初始化(Init)	应用层没有通讯，从站只能读写 ESC 芯片寄存器。
初始化向预运行转化 Init To Pre-OP(IP)	主站配置从站地址寄存器。 如果支持邮箱通讯，配置邮箱相关寄存器。 如果支持分布式时钟，配置 DC 相关寄存器。 主站写入状态控制寄存器，以请求 Pre-OP 状态。
预运行(Pre-OP)	应用层邮箱通讯
预运行向安全运行状态转化 Pre-OP To Safe-OP(PS)	主站使用邮箱初始化过程数据映射。 主站配置过程数据使用的 SM 通道。 主站配置 FMMU。 主站写入状态控制寄存器，请求 Safe-OP 状态。
安全运行(Safe-OP)	应用层支持邮箱通讯。 有过程数据通讯，但只允许读入数据，不产生输出信号。
安全运行向运行状态转化 Safe-OP To Op(SO)	主站发送有效输出数据。 主站写入状态控制寄存器，请求 Op 状态。
运行状态(Op)	输入输出全部有效。

①	26.03.2024 20:37:05	SDO write ok: Address: 1002 Index: 16#:
①	26.03.2024 20:37:05	SDO write ok: Address: 1002 Index: 16#:
①	26.03.2024 20:37:05	SDO write ok: Address: 1002 Index: 16#:
①	26.03.2024 20:37:05	SDO write ok: Address: 1002 Index: 16#:
①	26.03.2024 20:37:05	Set safe operational
①	26.03.2024 20:37:05	Synchronize Slaves
①	26.03.2024 20:37:04	Configure distributed clock settings
①	26.03.2024 20:37:04	All slaves pre-operational
①	26.03.2024 20:37:04	prepare slaves
①	26.03.2024 20:37:04	All slaves init mode

注意：正常断电上电卡在 1020 时，可以看下日志是否有 SDO 超时（参考从站掉线后重启很慢），偶发的 SDO 超时可以尝试更改 softmotion 到 4.10.0.0 或者把 14 的主站切换成 17 的主站，或者使用 SMC3_ReinitDrive 功能块重启轴。

SMC3_ReinitDrive 功能块执行时，如果有 SDO 操作（MC_ReadAxisError，读取 603F 对象字典等），大概率会卡在 20 一段之后变成 1020。

22. 轴报错 SMC_CGR_INVALID_POSPERIOD (模态轴上限)

此报错为设置的模态周期超过了可设置的模态周期上限，模态周期上限计算如下：

$$\text{模态周期上限} = \frac{2^{32} \times \text{工作台旋转一圈的工作行程} \times \text{齿轮比分子}}{\text{电机旋转一圈的指令脉冲数} \times \text{齿轮比分母}} \div 2$$

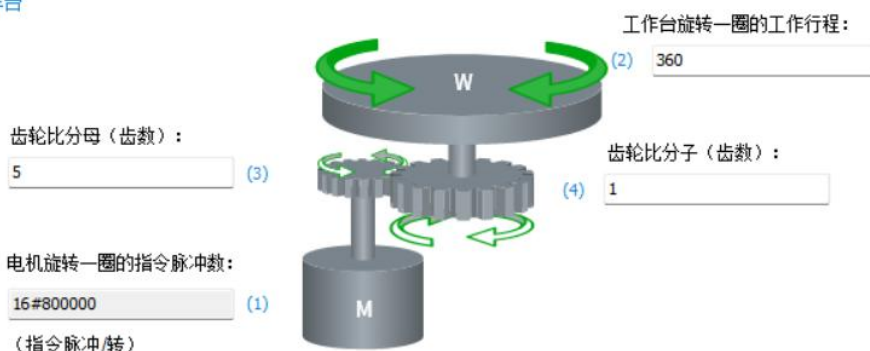
如下图所示，模态周期上限为 18432

轴类型为旋转模式

单位换算公式：

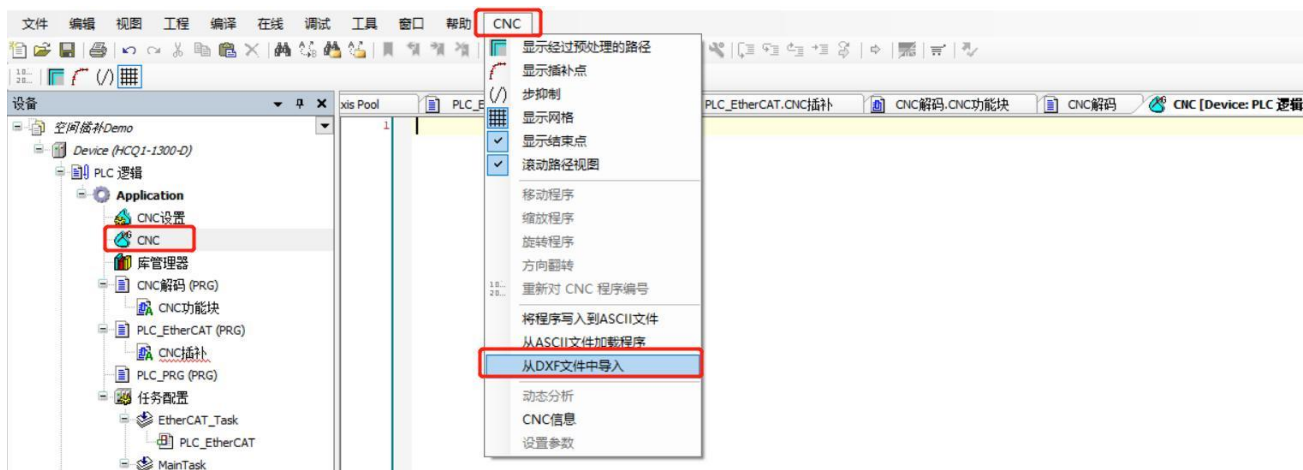
参考	脉冲数(pulse) =	(1) 电机旋转一圈的指令脉冲数[DINT]	(4) 齿轮比分子 [DINT]	* 移动距离[用户单位]
		(2) 工作台旋转一圈的工作行程[LREAL]	(3) 齿轮比分母 [DINT]	

M:电机 W:工作台



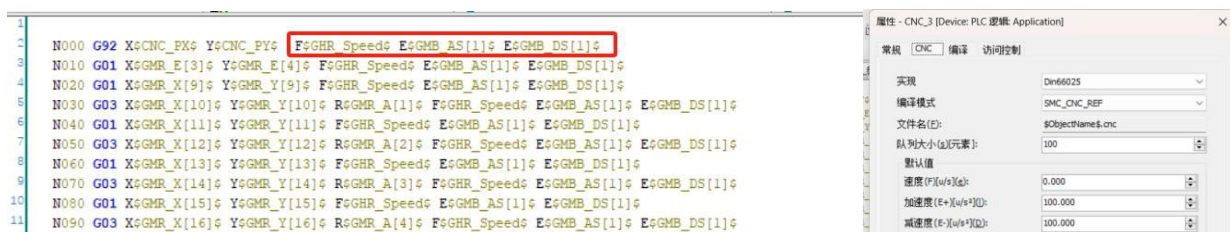
23. CNC 中 DXF 图形导入

Codesys 集成了图形导入功能，可将 CAD 等软件生成的 DXF 文件自动生成对应的 G 代码。具体操作如下：打开或新建一个 CNC-选择 CNC 选项卡-选择从 DXF 文件中导入-选中需要的 DXF 文件。

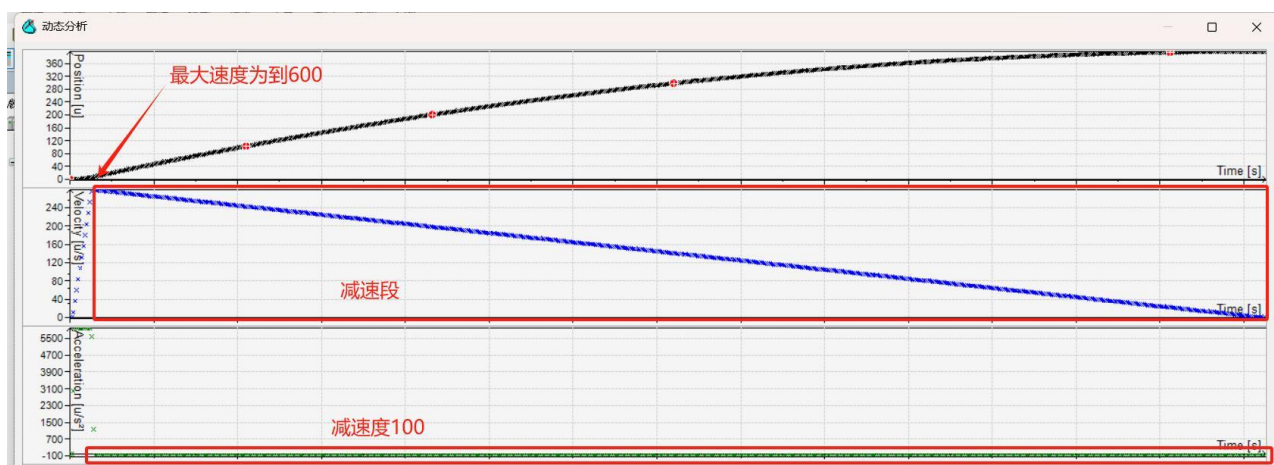


24. CNC 运行无法达到设定速度

速度、加速度、减速度都是从程序中获取，加速度需要为正数，减速度需要为负数，如果减速度设置错误，程序中赋值为正数，则会以默认 100 的减速度减速。



以速度 600 为例，减速时间为 6 秒，如果路程较短，则会出现达不到最高速度即开始减速，减速时间内速度会越来越慢，实际运行起来的结果为同样的速度，路径前半程运行较快，后半程运行很慢。



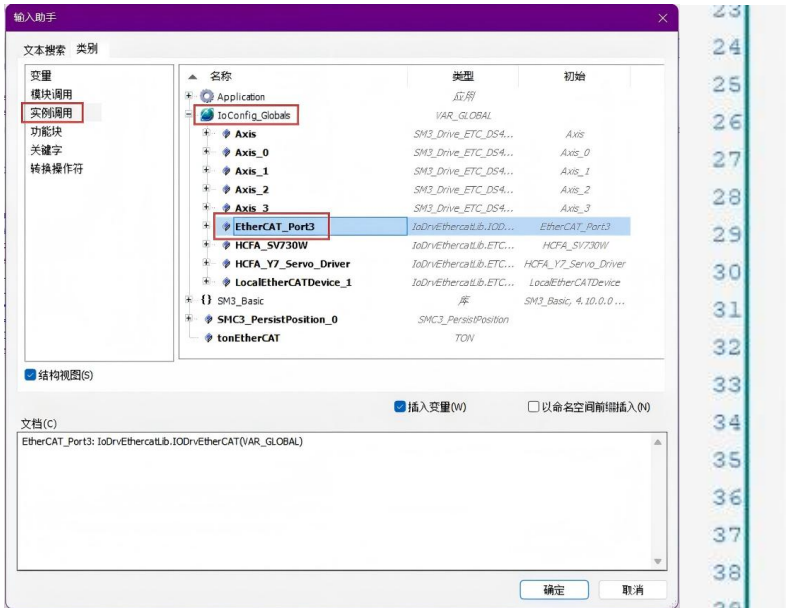
25. 写时钟指令导致 EtherCAT 掉线

FB_SetRTCDDate 功能块执行需要比较长的时间，如下图所示，如果功能块放在 EtherCAT 总线任务下，会导致 EtherCAT 总线任务超周期，出现轴报错 SMC_DI_FIELDBUS_LOST_SYNCRONICITY 的情况，建议写时钟功能块放到低优先级较大周期的任务中。



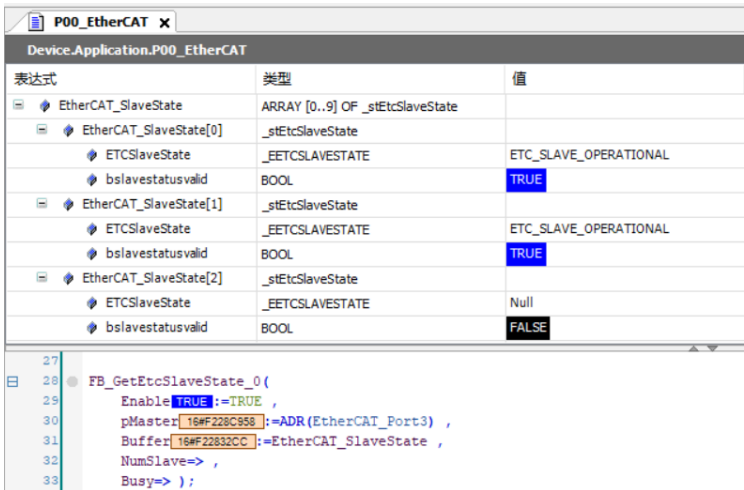
26. 程序中重启 EtherCAT 总线

添加 EtherCAT 主站后，会自动声明主站功能块，在程序中按 F2 调用功能块，触发 xRestart 可以重启 EtherCAT 总线。



27. 如何获取所有从站状态

可以通过 ATC 库中功能块 FB_GetEtcSlaveState 获取所有从站的状态，pMaster 为主站的指针，Buffer 为从站状态缓存区，声明_stEtcSlaveState 变量类型的数组赋值即可，NumSlave 显示当前连接的从站数量，_stEtcSlaveState 中 ETCSlaveState 表示从站的状态，Null 表示没有获取到从站状态。



28. 在程序中启用禁用 EtherCAT 从站

有两种方法实现此功能：

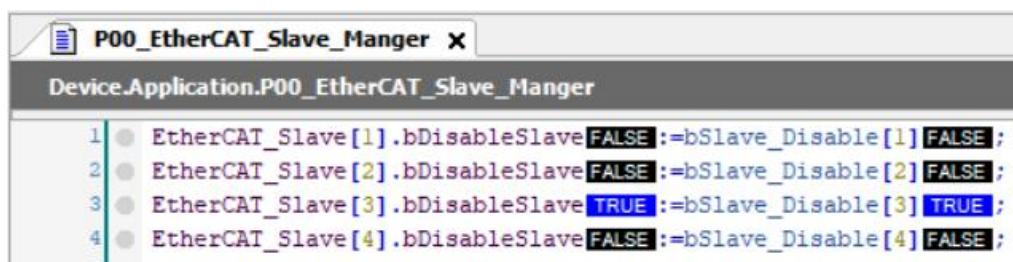
第一种方法是需要调用 hcfaAtcLib 库中的 FB_EtherCATManager 功能块，当前程序有几个从站就声明有几个 stECATSlave 元素的从站数组（一般 LocalEtherCATDevice 为第一个从站），变量声明和程序编辑如下：

```
GVL_PersVars X
1  //{attribute 'qualified_only'}
2  VAR_GLOBAL RETAIN PERSISTENT
3  bSlave_Disable AT $MB131072 : ARRAY [1..4] OF BOOL;
4  END_VAR

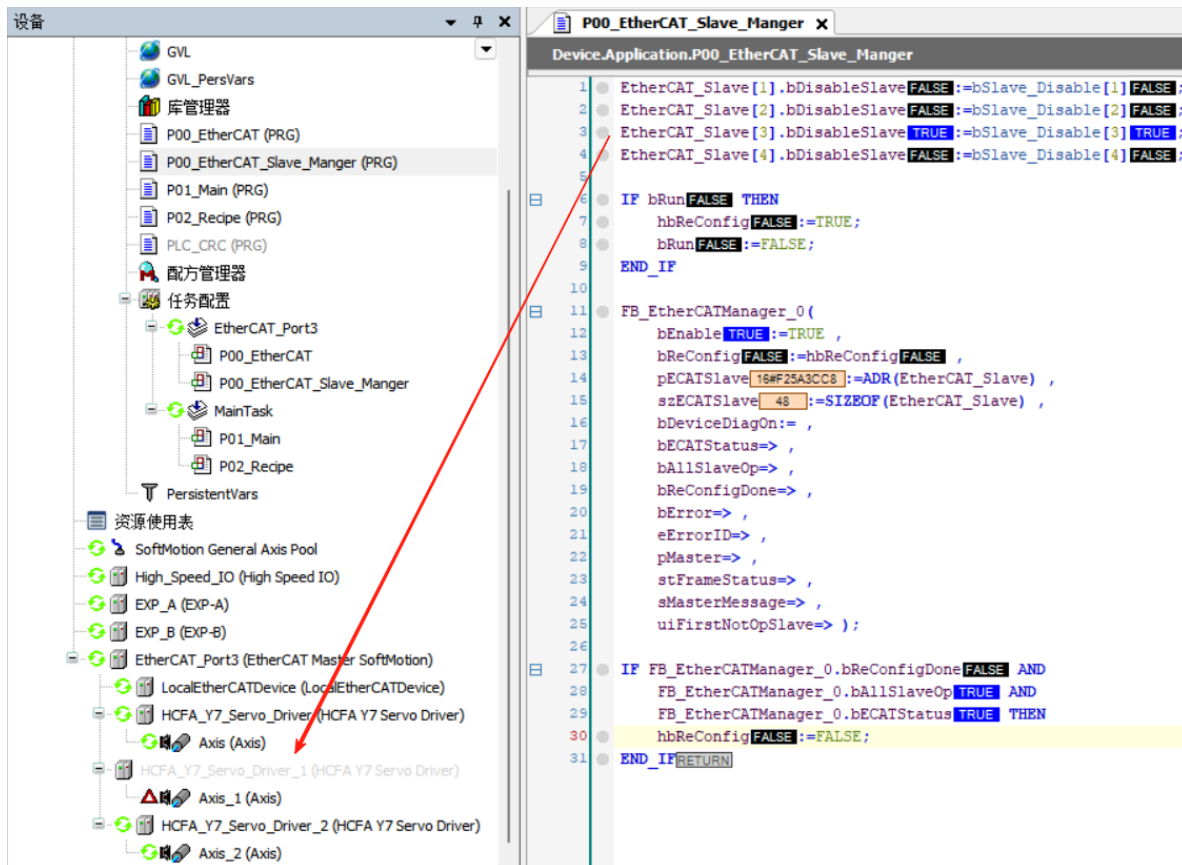
P00_EtherCAT_Slave_Manger X
1  PROGRAM P00_EtherCAT_Slave_Manger
2  VAR
3      FB_EtherCATManager_0      :FB_EtherCATManager;
4      EtherCAT_Slave           :ARRAY[1..4] OF stECATSlave;
5      bRun                     :BOOL:=TRUE;
6      hbReConfig               :BOOL;
7  END_VAR

1  EtherCAT_Slave[1].bDisableSlave:=bSlave_Disable[1];
2  EtherCAT_Slave[2].bDisableSlave:=bSlave_Disable[2];
3  EtherCAT_Slave[3].bDisableSlave:=bSlave_Disable[3];
4  EtherCAT_Slave[4].bDisableSlave:=bSlave_Disable[4];
5
6  IF bRun THEN
7      hbReConfig:=TRUE;
8      bRun:=FALSE;
9  END_IF
10
11 FB_EtherCATManager_0(
12     bEnable:=TRUE ,
13     bReConfig:=hbReConfig ,
14     pECATSlave:=ADR(EtherCAT_Slave) ,
15     szECATSlave:=SIZEOF(EtherCAT_Slave) ,
16     bDeviceDiagOn:= ,
17     bECATStatus=> ,
18     bAllSlaveOp=> ,
19     bReConfigDone=> ,
20     bError=> ,
21     eErrorID=> ,
22     pMaster=> ,
23     stFrameStatus=> ,
24     sMasterMessage=> ,
25     uiFirstNotOpSlave=> );
26
27 IF FB_EtherCATManager_0.bReConfigDone AND
28     FB_EtherCATManager_0.bAllSlaveOp AND
29     FB_EtherCATManager_0.bECATStatus THEN
30     hbReConfig:=FALSE;
31 END_IF
```

首先下载程序到控制器中并运行，置位需要禁用的从站，然后设备断电。



更改禁用从站后的网络拓扑后，上电等待总线启动，效果如下：



第二种方法是通过从站的 SetEnabled 函数搭配主站总线重启功能块或者断电重启等操作实现此功能。

```

EtherCAT_Port3(
  xRestart:=bRestart ,
  xStopBus:= ,
  xConfigFinished=> ,
  xDistributedClockInSync=> ,
  xError=> ,
  xSyncInWindow=> );

```

```

HCFA_Y7_Servo_Driver_1.SetEnabled(pConnector:=HCFA_Y7_Servo_Driver_1.m_pConnector,xEnabled:=NOT bEnable);
HCFA_Y7_Servo_Driver_1.SetEnabled(pConnector:=Axis_1.m_pConnector,xEnabled:=NOT bEnable);

```

29. 如何将 DXF 文件导入控制器中，转换成 CNC 程序？

可以通过禾川简易 CAM 软件将 DXF 文件转换成 CNC 程序，再将 CNC 程序放到 U 盘\SD 卡中，控制器读取 CNC 文件解析并运行就可以了。详细可咨询控制产品线技术人员。

四、异常问题

1. 控制器异常——面板报错 ER0000

这个报错说明控制器中没有应用程序，可以下载个空程序进去测试是否能正常启动。一般会在刷固件后出现此报错，没有显示面板的控制器 Q0P/Q1P 在刷固件后 ERR 也会闪烁红灯。

2. 控制器异常——面板报错 ER0051

这个报错是程序中出现了非法地址访问，一般是也因为空指针或数组越界导致的。它会侵占系统变量或者相邻变量，进而导致 PLC 宕机或程序数据异常等情况。

检查程序中是否存在空指针和数组越界。可以采用二分法，逐步调用程序，重点排查指针类型变量的使用 and FOR 循环等程序。有些功能块的输入引脚声明了指针变量，但在调用时却没给它赋值，就会导致空指针。数组越界一般发生在 FOR 循环之类的程序处，尤其是改造项目特别容易出现这个错误。

3. 控制器异常——面板报错 ER0102、ER0152

查看日志发现抛异常 FPU DivisionByZero

12.09.2023 13:21:41	*SOURCEPOSITION* App=[Application] area=3, offset=2931	CmpIecTask
12.09.2023 13:21:41	*EXCEPTION* [FPU DivisionByZero] occurred: App=[Application], Task=[EtherCAT_Task]	CmpIecTask

这是程序中存在浮点数除零导致的，可以添加隐式 POU 中的 REAL/LREAL 除零检查，或者程序中加代码判断，防止除零（隐式 POU 不会检查库中的除零代码）。

注意：浮点数除零在各版本运行结果可能不一致，某些版本不会抛异常，会显示非数，如下图所示。

```
fData := fData2 / fData2 ; RETURN
```

4. 控制器异常——面板卡 HCFA

面板卡 HCFA 说明控制器还没有进系统，首先检查控制器供电电压是否稳定，或者换个开关电源测试；其次尝试能否扫描到控制器，扫描到就下载个空程序进去观察是否能正常运行；如果换电源、断电上电等操作后依然卡 HCFA，并且也扫描不到控制器就联系产品线技术人员协助排查。

5. EtherCAT 总线异常——丢同步、掉线

在总线任务下添加 CRC 计数监控程序，观察 Port0 和 Port1 口的寄存器值是否持续变化，如果 Link Lost Counter 的值持续增加说明对应的节点网口松动或者网线质量不良，可以更换网线或设备对比测试；

6. EtherCAT 总线异常——扫描设备时提示“存储库中找不到此设备”

缺少对应设备的 XML 文件，或是 XML 文件的版本号不匹配。

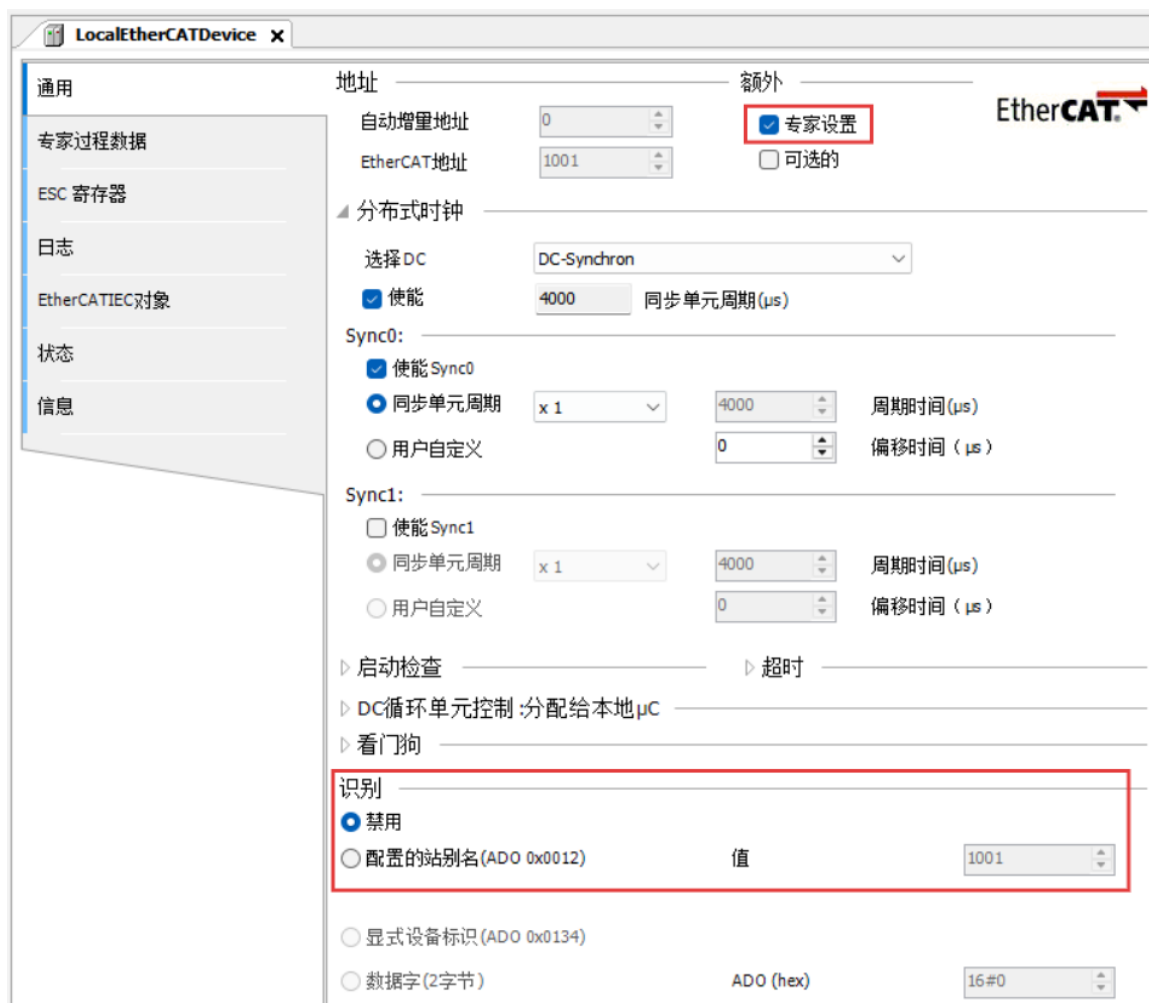
7. EtherCAT 总线异常——扫描不到设备

可以分几步排查：

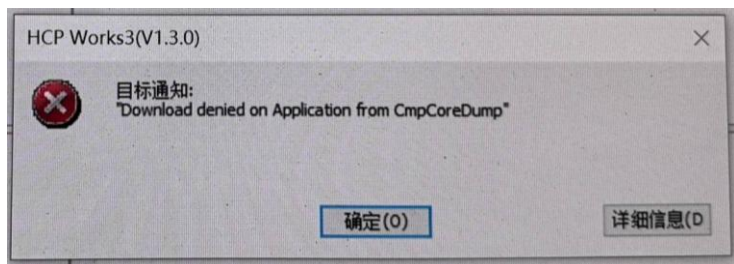
- 1) 首先检查从站设备是否正常供电、网线是否正常连接（In、Out）、模块是否插紧等；
- 2) 检查网口指示灯是否闪烁；
- 3) 控制器热复位后，不用运行控制器，直接在 STOP 状态下扫描从站设备；
- 4) 断开所有从站，逐个安装、上电、扫描直至全部能扫描到；
- 5) 如果能排查到具体哪个模块有问题，可以联系产品线提供技术支持；

8. EtherCAT 总线异常——扫描正常，但总线起不来

检查 LocalEtherCATDevice 的站地址，这个是控制器硬件决定的第一个 EtherCAT 从站，EtherCAT 地址必须是 1001，不能修改站别名（站地址）。



9. 下载报错——Download Denied on Application from CmpCoreDump



查看设备日志，如果发现有“EXCEPTION”字样，说明程序中可能存在非法指针操作。

时间	时间戳	消息	来源
24.11.2023 22:51:38		*SOURCEPOSITION* App=[Application] area=3, offset=1145650	CmpApp
24.11.2023 22:51:38		*EXCEPTION* [GlobalExit] code: App=[Application], Exception=[AccessViolation]	CmpApp

解决办法：

- 1) PLC 拨到 STOP 断电重启，尝试重新下载程序；
- 2) 初始复位 PLC，右键 Device→初始复位设备，断电重启；

3) 两种方式都试过不行的, 刷固件 (非保留配置版本) 清除内部程序, 即可重新下载;

10. 下载报错——原因未知

查看日志发现有非法地址访问, GlobalInit 异常

严重	时间标记	描述	组件
⚠	28.03.2024 02:29:58	*SOURCEPOSITION* App=[Application] area=3, offset=1707470	CmpApp
⚠	28.03.2024 02:29:58	*EXCEPTION* [GlobalInit] code: App=[Application], Exception=[AccessViolation]	CmpApp

结构体中默认以 4 字节对齐为准, 如果浮点数没有位于一个能被 4 整除的地址上, 在 ARM 平台上无法一次获取 32 位数据, 从而出现异常 (64 位平台无此问题)。

解决办法:

- 1) 声明地址时注意把浮点数映射在偶数地址上;
- 2) 更改结构体中对齐方式, 通过属性 Pack_mode, 更改对齐方式, 设置为 0, 1, 2 时均可正常, 如下图所示:

```
{attribute 'pack_mode' := '0'}  
TYPE AAA :  
STRUCT  
    Data :REAL;  
END_STRUCT  
END_TYPE
```

11. 编译报错——没有可用于此对象的源代码, 因为它位于已编译的库'sm3basic,4.10.0.0...

此报错是因为程序中存在空指针或数组溢出导致的“踩内存”。

12. 编译报错——Internal error:System.NullReferenceException

此错误一般为版本不兼容导致, 需要切换到对应的版本解决, 如库版本、softmotion、可视化、编译器。



13. 编译报错——The Library *** is only supported in 32 bit applications

此报错说明调用的库文件仅支持 32 位应用程序, 比如下面这个报错:

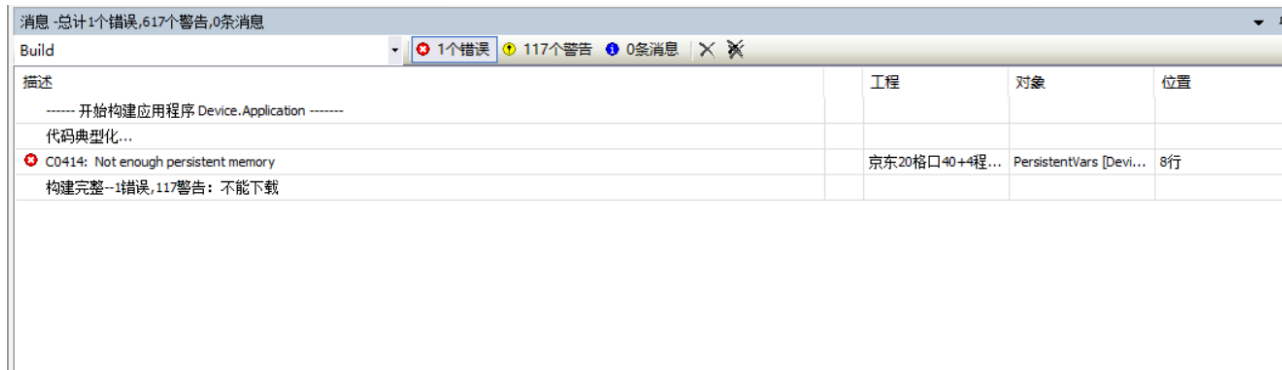
消息: 总计1个错误, 12个警告, 0条消息	编译	描述	工程	对象	位置
编译	1个错误	0个警告	0条消息		
描述	开始构建应用程序 Device.Application				
代码典型化...					
代码和数据总内存容量: 786432字节					
内存区域0包含 数据, 输入, 输出, 内存和 Non-safe data: 大小: 1048576字节, 最高使用的地址: 786432, 最大连续存储器间距: 262144字节 (25%)					
内存区域3包含 代码: 大小: 4096字节, 最高使用的地址: 0, 最大连续存储器间距: 4096字节 (100%)					
Internal error: System.NullReferenceException: 未将对象引用设置到对象的实例。 在 C:\..._IndexAccessExpression 在 C:\..._JCalExpression ...					
构建完整--1错误, 0警告: 不能下载					

选择不带 23 字样的系统库即可兼容 32 位系统和 64 位系统, 如 sysmem23 库只能在 32 位 系统上应用, 改用 sysmem 库即可

在 32 位/64 位系统上应用。

14. 编译报错——Not enough persistent memory

这是因为在掉电保持列表中删除较多的变量，然后创建新的掉电保持变量，产生了较多的占位符 变量（间隔项），添加新的掉电保持变量后，存储的区域不足够存储占位符变量。清除/清除全部可以 清除这些间隔项。



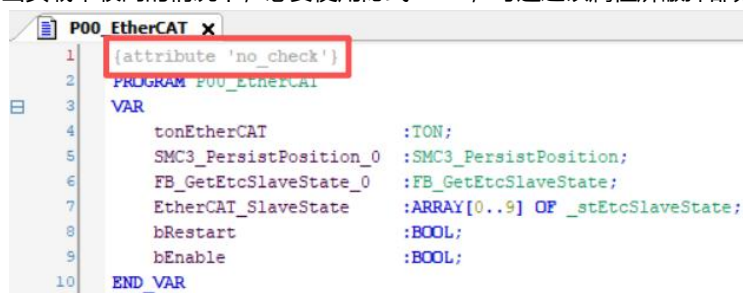
五、典型应用

1. 隐式函数检查

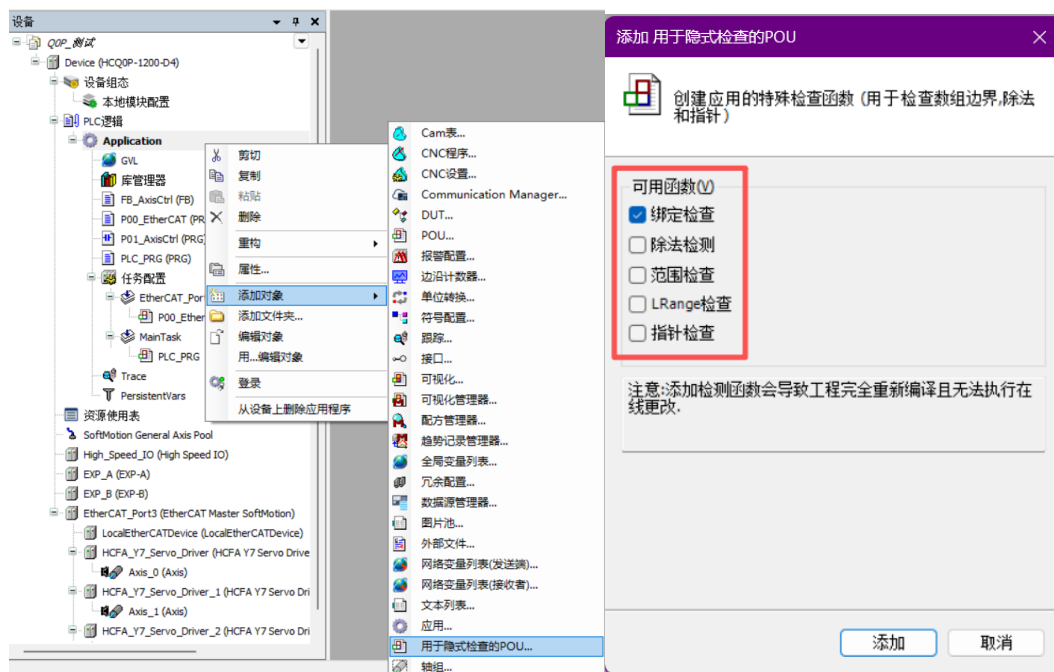
程序隐式函数检查功能，用户在编写程序的过程中，可能会发生数组越界/空指针/除零的情况，诊断这几种情况，Codesys 默认提供了用于隐式检查的函数，用户可以添加这些函数检查已编辑程序，隐式 POU 属性中提供了绑定检查/除法检查/范围检查/LRange 检查/指针检查。

用于隐式检查的函数添加后，会在程序在线运行的时，自动在后台运行并根据所提供的功能对程序进行检查，检查过程会占用 CPU 资源。实际运行过程中，如果负载率不高的情况下，建议添加这些检查函数，以保证程序正常运行。

注意：当不想对程序中特定 POU 进行隐式检查的情况下，可使用属性 “no_check”，在程序中引用该属性的 POU 则不会被检查。当负载率较高的情况下，想要使用隐式 POU，可通过该属性屏蔽掉部分 POU。使用方式如下：



添加隐式 POU：右键 Application→添加对象→用于隐式检查的 POU,勾选需要的功能→添加



以绑定检查为例：

通过绑定检查可以检查数组的边界是否超出，使用该检查函数可以有效保证数组不出现边界溢出的情况。

所有隐式检查函数的输入变量声明都不可编辑，如果用户进行变量的删除和添加等操作，程序编译将出现错误，但是可以添加局部变量，代码实现区是可编辑的，用户可以在不改变原有代码实现架构的情况下对代码实现进行自定义编辑。

数组越界排查步骤：

在全局变量中声明一个错误计数的变量（dErrCount），用于检查是否存在越界的情况，如果数组存在越界的情况 dErrCount 的

值会自加 1，最终查看 dErrCount 的数值即可判断数组是否存在越界。

```
1 // 隐式生成代码:不需要编辑
2 FUNCTION CheckBounds : DINT
3   VAR_INPUT
4     index, lower, upper: DINT;
5   END_VAR
6
7   // 隐式生成代码:只是实现代码的建议
8   IF index < lower THEN
9     CheckBounds := lower;
10    dErrCount:=dErrCount+1;
11  ELSIF index > upper THEN
12    CheckBounds := upper;
13    dErrCount:=dErrCount+1;
14  ELSE
15    CheckBounds := index;
16  END_IF
```

当看到 dErrCount 数值不为 0 时，程序中存在数组越界的情况，在绑定检查函数中打上断点，来确定越界代码所在位置。发生越界时，程序停在断点处。

```
1 // 隐式生成代码:只是实现代码的建议
2 IF index < lower THEN
3   CheckBounds := lower;
4   dErrCount:=dErrCount+1;
5 ELSIF index > upper THEN
6   CheckBounds := upper;
7   dErrCount:=dErrCount+1;
8 ELSE
9   CheckBounds := index;
10 END_IF
```

点击视图，选择调用堆栈。点击消息栏中具体代码位置。

调用堆栈

应用: Device.Application	任务: MAINTASK
POU	位置
CheckBounds [Device: PLC逻辑: Application]	行4,列2 (Impl)
PLC_PRG [Device: PLC逻辑: Application]	行3,列1 (Impl)

CheckBounds PLC_PRG

表达式	类型	值
ifor	INT	0
arTest	ARRAY [1..9] OF REAL	

```
1 IF bStart THEN
2   FOR ifor:=0 TO 9 DO
3     arTest[ifor] := arTest[ifor] + 1;
4   END_FOR
5 END_IF
```

代码实现区可以自己修改，例如：数组越界产生越界异常，在 PLC 界面上显示越界异常

2. PID 温度控制

需要安装库 PIDLIB 2.0.0.0.compiled-library，使用 PID、PWS 功能块组合即可实现自调整 PID 参数功能，具体使用范例如下：

```

rActValue:=TO_REAL($IW2)/10;

PID_0(
    RUN:=bRun ,                                //功能块开始运行
    ManualCtrl:=FALSE ,
    AutoTune:=bAutoTune ,                       //开始自动调整
    SV:=rSetValue ,
    PV:=rActValue ,
    CycleTime:=LTIME#100MS ,
    Proportional:=rP ,                          //在自调整时随便给个值，但不能为0，给个初始值就好
    IntegrationTime:=ltI ,                     //在自调整时随便给个值，但不能为0，给个初始值就好
    DerivativeTime:=ltD ,                      //在自调整时随便给个值，但不能为0，给个初始值就好
    RangLow:=0 ,                               //输入下限
    RangUp:=300 ,                              //输入上限
    MVLow:=0 ,                                 //输出下限
    MVUp:=100 ,                               //输出上限
    ManualMV:=rManualMV ,
    Direction:= ,
    Options:= ,
    ATDone=> ,
    ATBusy=> ,
    Error=> ,
    ErrorID=> ,
    MV=> );                                     //输出值的范围是0-100

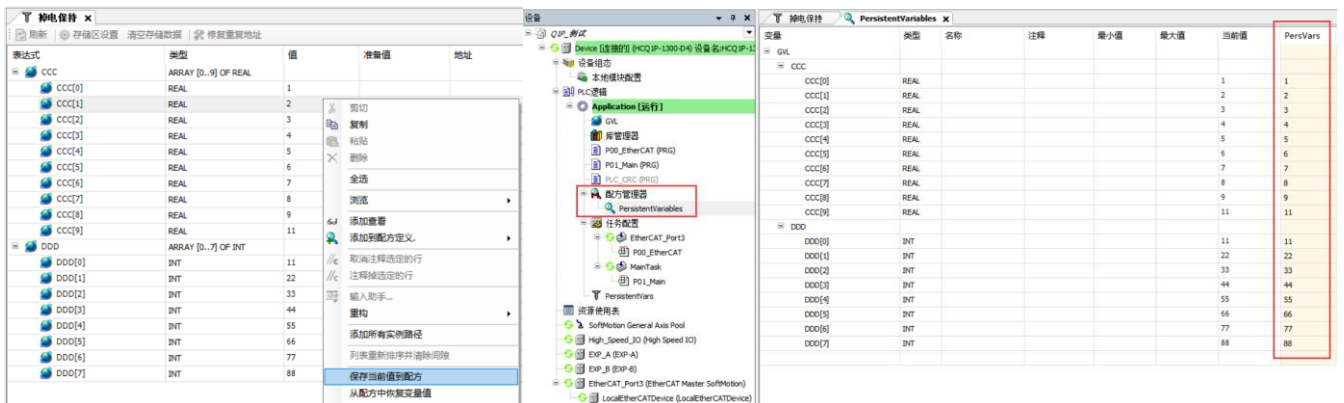
PWM_0(
    Enable:=bOut ,
    OnTime:=REAL_TO_INT(PID_0.MV*10) ,          //PID输出值*10就是相当于将ON时间调成在1s中所占百分比
    CycleTime:=1000 ,                          //周期为1000ms
    Out=>$QX0.0 );

//执行RUN,功能块开始运行，ATBUSY输出代表开始执行自增益PID
//当ATDone被置位代表运算完成，PID新的参数也会生成，结果会覆盖到比例系数，微分时间，积分时间引脚上
//如需下次使用该组参数无需自调节，将AutoTune复位，直接run即可使用该组PID参数。

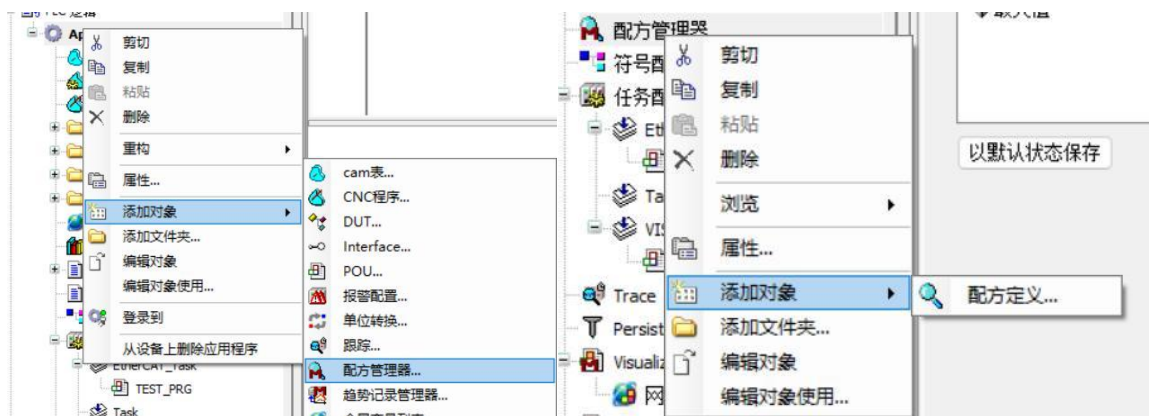
```

3. 配方功能使用

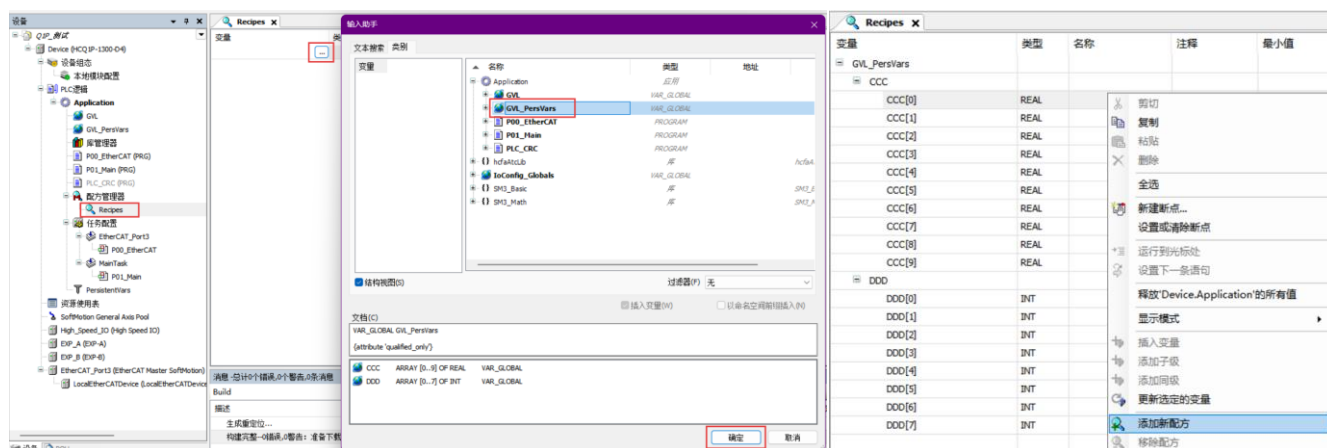
掉电保持变量 PersistentVars 旧模式下可以快速添加变量到配方中。在线状态下，在 PersistentVars 中右键保存当前值到配方。设备树中会自动创建配方管理器以及配方定义 PersistentVariables，并且会将当前变量创建为配方 PersVars。



掉电保持变量 PersistentVars 新模式下需要在离线时，在 Application 右键添加对象→添加配方管理器，然后在配方管理器上右键添加对象→添加配方定义。



在配方定义中，选择需要的配方变量，然后右键添加配方，可以添加多组配方，实现设备工艺参数快速切换。



在配方列下面右键读取配方就能将当前值存入当前配方中，还可以将当前配方保存到电脑上防止丢失。

还可以在配方管理器→常规选项卡中，勾选 PLC 中的配方管理，这样就可以在程序中使用 RecipeManagement.library 中的 RecipeManCommands，实现在控制器运行时保存与加载配方数据，配方文件可以放在控制器断电保持存储区（“FlashFiles/” 路径下）。

具体程序实现方法可以参考下图：

```
//实例调用
RecipeManCommands_0();

//设置存储路径
IF bSetStoragePath THEN
    IF RecipeManCommands_0.SetStoragePath(stPath:='FlashFiles/RecipeFile/') THEN
        bSetStoragePath_OK:=TRUE;
        bSetStoragePath:=FALSE;
    END_IF
END_IF

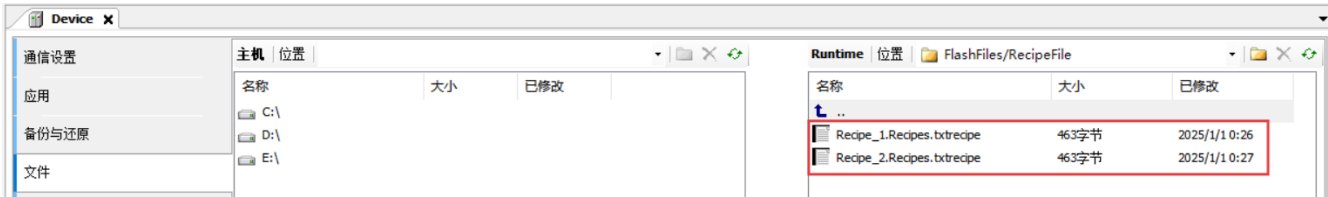
//生成配方名称
strRecipeNum:=TO_STRING(wRecipeNum);
strRecipeName:=CONCAT('Recipe_',strRecipeNum);

//创建配方文件
IF bCreateRecipe AND bSetStoragePath_OK THEN
    IF RecipeManCommands_0.CreateRecipe(RecipeDefinitionName:='Recipes',RecipeName:=strRecipeName)=16#0 THEN
        bCreateRecipe_OK:=TRUE;
        bCreateRecipe:=FALSE;
    END_IF
END_IF

//读取PLC变量到配方，并保存配方到配方文件
IF bReadSaveRecipe AND bSetStoragePath_OK THEN
    IF RecipeManCommands_0.ReadAndSaveRecipe(RecipeDefinitionName:='Recipes',RecipeName:=strRecipeName)=16#0 THEN
        bReadSaveRecipe_OK:=TRUE;
        bReadSaveRecipe:=FALSE;
        bCreateRecipe_OK:=FALSE;
    END_IF
END_IF

//从配方文件加载配方，并且写入配方值到PLC变量
IF bLoadWriteRecipe AND bSetStoragePath_OK THEN
    IF RecipeManCommands_0.LoadAndWriteRecipe(RecipeDefinitionName:='Recipes',RecipeName:=strRecipeName)=16#0 THEN
        bLoadWriteRecipe_OK:=TRUE;
        bLoadWriteRecipe:=FALSE;
    END_IF
END_IF
```

配方文件生成效果及存储路径如下图：



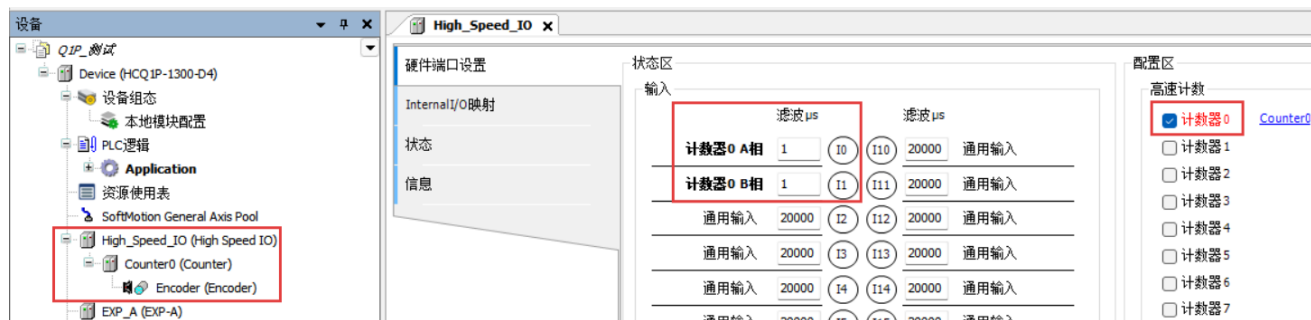
注意事项：

- 1) 不能在 EtherCAT 任务下执行该指令，配方变量多的时候占用 PLC 资源，建议放在 低优先级，任务扫描周期为 100ms 及以上；
- 2) 上图示例程序仅展示了基本操作，不包含错误检测与复位，标志位复位等，需要可自行添加；
- 3) 上图示例程序中存储路径为 “FlashFiles/RecipeFile/”，其中 FlashFiles 文件夹是本身就存在的，需要在这个文件夹内手动创建一个 RecipeFile 文件夹来存储配方文件；
- 4) 只有存在 FlashFiles 文件夹下的文件才不会断电丢失；

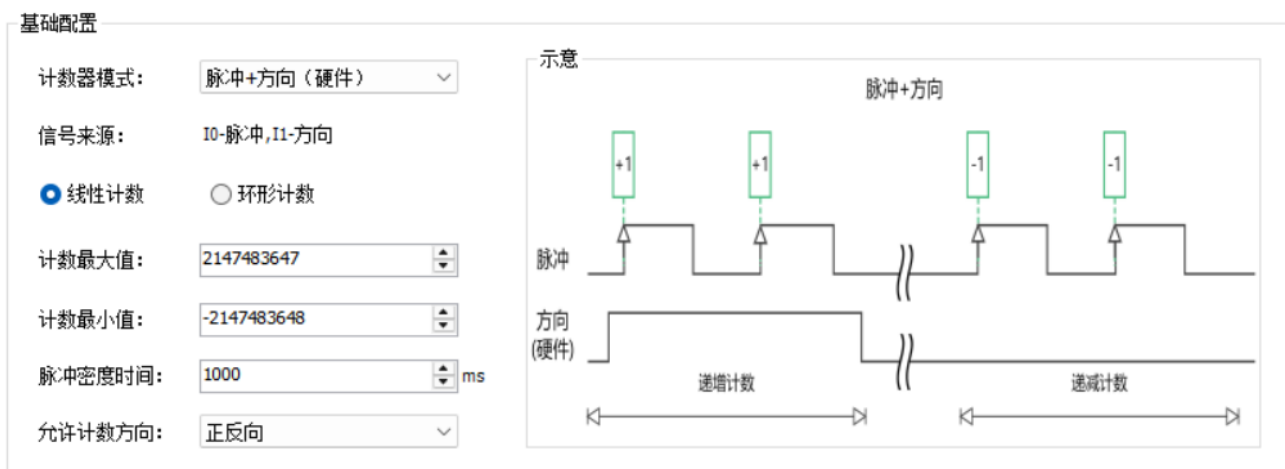
4. 高速输入输出功能应用

高速输入功能——双击设备树中的 High_Speed_IO，在硬件端口页面左侧勾选计数器 0，它会在输入状态区自动显示对应端子

信息。在设备树中也会自动生成 Counter0 和 Encoder（可以对计数器名重构，例如 Encoder_0）。



双击设备树中的 Counter0，在基础配置中选择计数器模式，有 A/B 相 1、2、4 倍频，CW/CCW 和脉冲+方向等方式。如果用线性计数方式，计数范围是-2147483648~2147483647，超过这个范围就会溢出，可以选用环形计数方式避免此问题。

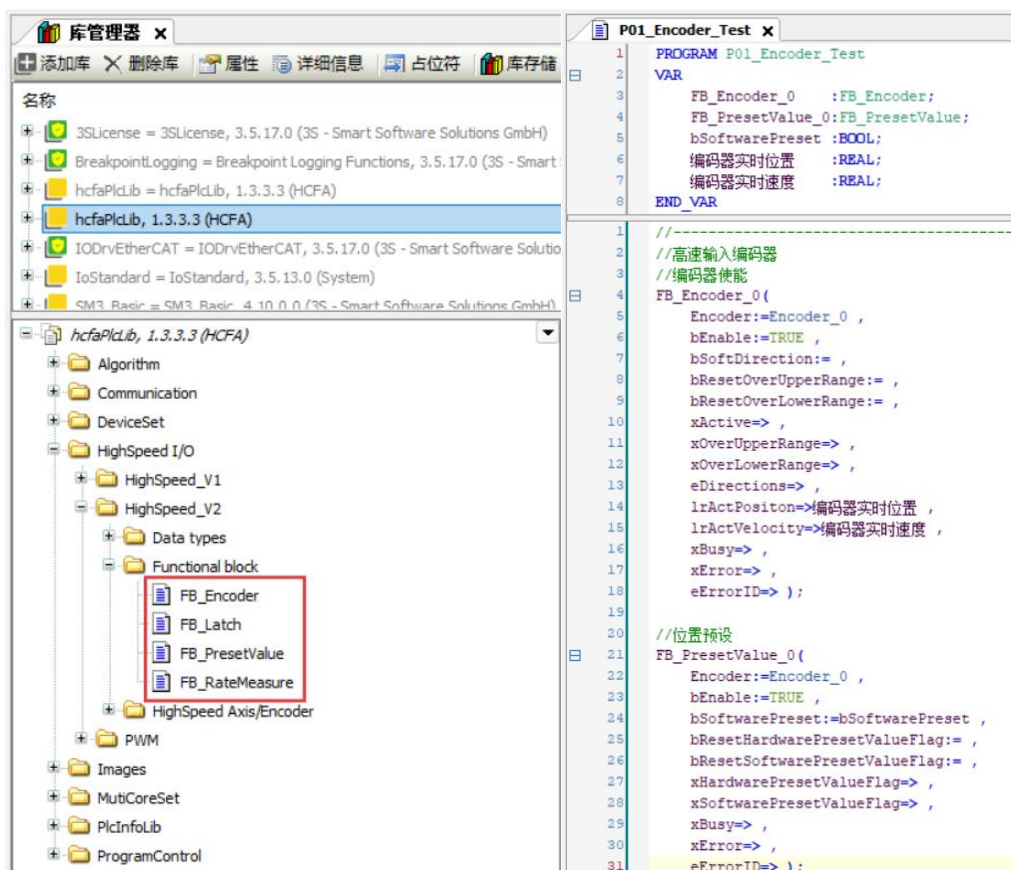


在下面的硬件复位处，可以先勾选使能，在后面的预置值处修改需要的预置值，选择外部硬件 IO 复位计数值（预置值为脉冲值）。

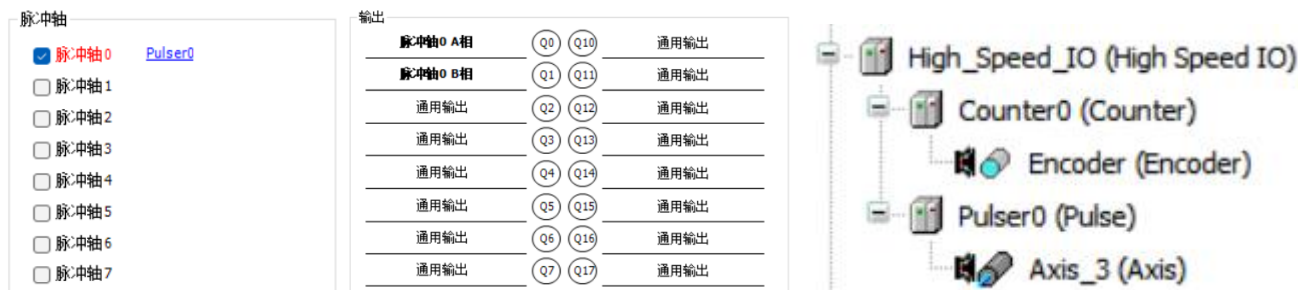


也可以不用外部 IO，通过 FB_PresetValue 指令预置计数器值，此时不勾选使能，只设置预置值即可。根据编码器参数可以设置自动生成的 Encoder 参数（编码器线数和用户单位）。每路编码器可以使用 2 路探针，程序调用 FB_Latch 功能块即可。

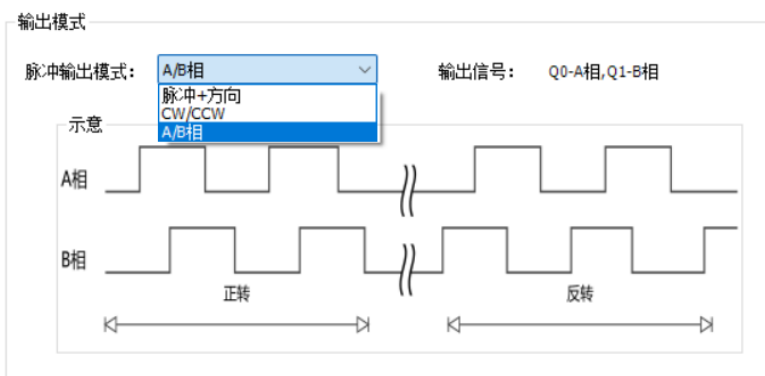




高速输出功能——双击设备树中的 High_Speed_IO，在硬件端口页面右侧勾选脉冲轴 0，它会在输出状态区自动显示对应端子信息。在设备树中也会自动生成 Pulser0 和 Axis。



双击设备树中的 Pulser0，在基础配置中选择脉冲输出模式，输出模式有脉冲+方向、CW/CCW、A/B 相。



在程序中调用**标准 Codesys 轴控 MC 指令** (MC_Home 除外)，程序也要放在 EtherCAT 任务下才能正常运行。


```

1  PROGRAM P00_Pulser_Test
2  VAR
3      MC_Power_0      :MC_Power;
4      MC_Jog_0        :MC_Jog;
5      bPower          :BOOL;
6      bJogF           :BOOL;
7      bJogB           :BOOL;
8      rJogVel         :REAL;
9  END_VAR

10 //-----
11 //高速输出轴
12 //脉冲轴使能
13 MC_Power_0(
14     Axis:=Axis_0 ,
15     Enable:=TRUE ,
16     bRegulatorOn:=bPower ,
17     bDriveStart:=bPower ,
18     Status=> ,
19     bRegulatorRealState=> ,
20     bDriveStartRealState=> ,
21     Busy=> ,
22     Error=> ,
23     ErrorID=> );
24
25 //脉冲轴点动
26 MC_Jog_0(
27     Axis:=Axis_0 ,
28     JogForward:=bJogF ,
29     JogBackward:=bJogB ,
30     Velocity:=rJogVel ,
31     Acceleration:=1000 ,
32     Deceleration:=1000 ,
33     Jerk:=10000 ,
34     Busy=> ,
35     CommandAborted=> ,
36     Error=> ,
37     ErrorID=> );

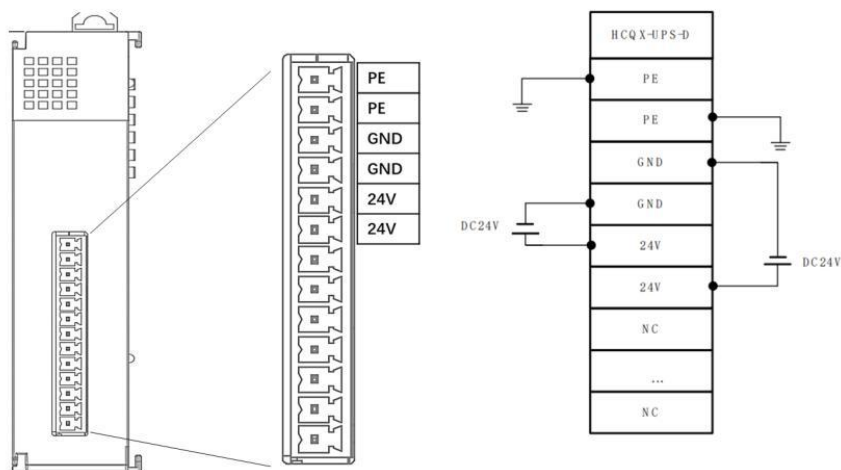
```

5. UPS 模块应用说明

UPS 模块右侧要求有良好的通风散热，不可遮挡，尤其不能接触发热源

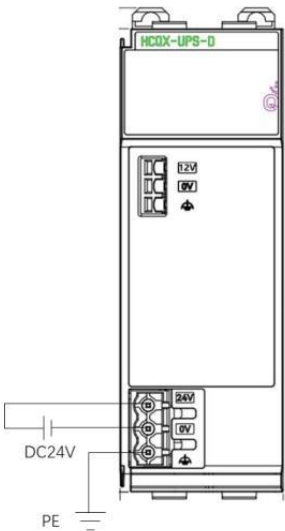
HCQX-UPS-D-T05：挂在主机最右侧，中间只能最多跨两个扩展模块，瞬时功率 96W(MAX)

端子及接线说明：



HCQX-UPS-D-T09：挂在主机最右侧，中间最多跨 16 个扩展模块，瞬时功率 60.48W(MAX)。

端子及接线说明：



针对空间狭窄，无法安装 HCQX-UPS-D-T09 的情况，可以通过 HCQX-UPS-D-TRF 中转，需要的空间为一个 16 点模块大小，HCQX-UPS-D-TRF 说明及接线如下：

表三 中转模块 HCQX-UPS-D-TRF 端子排列说明

说明	NO.		说明
GND	0	10	12V
	1	11	
	2	12	
	3	13	
	4	14	
	5	15	
	6	16	
PE	7	17	
	8	18	PE

注：PE（8，18）之间内部导通，任意接一个即可

