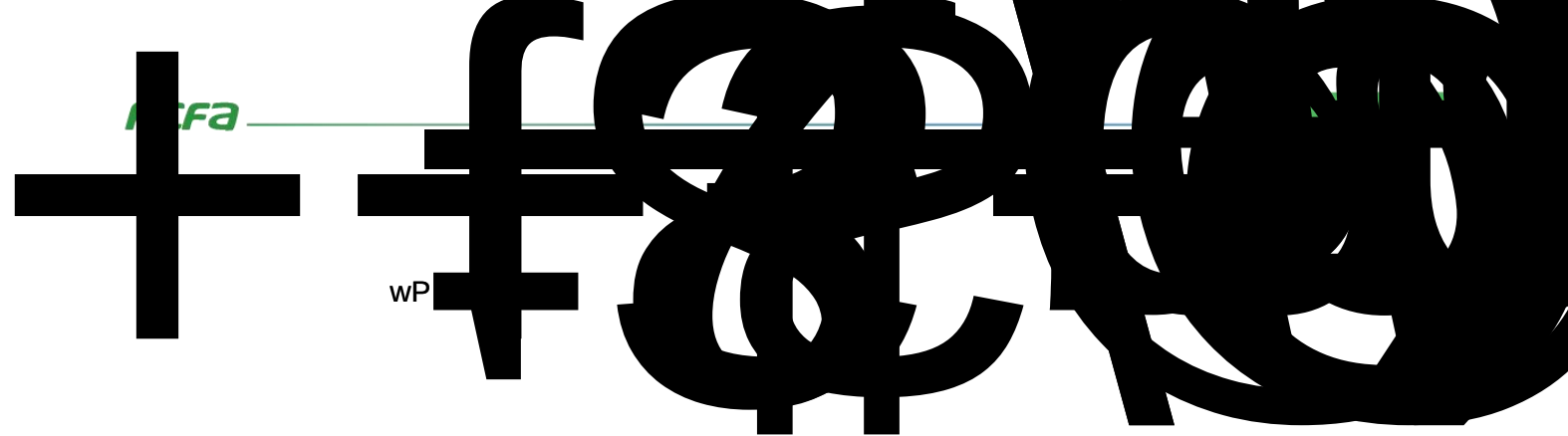




. © ú




```

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             rtteststring:=read;
24             END_IF
25         ELSE
26             rtteststring:= '';
27             END_IF
28         IF EtherNetIPScanner_0.bReadTag(ADR('rtteststring'),eCipTypes.STRING,ADR(rtteststring),SIZEOF(rtteststring),1,ADR(read),FALSE) AND NOT EtherNetIPScanner_0.bBusy THEN
29             rtteststring:=read;
30             END_IF
31         END_CASE

```

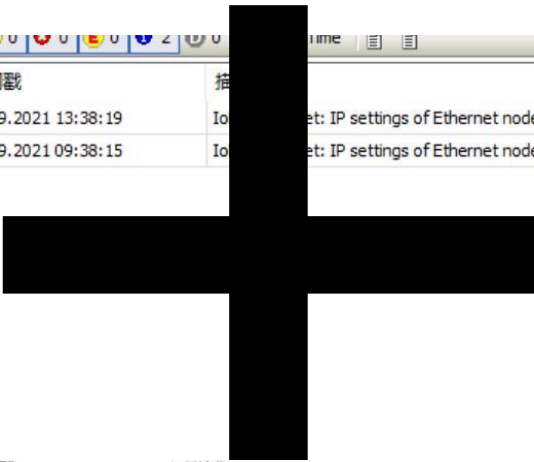
PRG05_Unit20_L.ACT11_AutoChild_装卸 GVL Ethernet x

Ethernet Device : 有一条诊断消息待定。

上一条诊断信息 确认

EthDiag		Ethernet diagnostic information
Current IP	'192.168.22.226'	
Current subnet mask	'255.255.252.0'	
Current gateway address	'192.168.20.254'	
IP changes	1	Amount of IP configuration changes since startup (IP or gateway address or subnet mask)

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



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	IoDrvEtherf
	28.05.2024 19:08:54...	IoDrvEthernetIP: Connection failure. (16#1)	IoDrvEthNe
	28.05.2024 19:08:54...	IoDrvEthernetIP: Close Connection with serial number 1	IoDr

t0

严重	时间戳	描述	组件
	17.06.2024 16:07:18	Modbus/TCP Slave[0] (192.168.188.2): Connected.	Modbus TCP Master
	17.06.2024 16:07:11	Modbus/TCP Slave[0] (192.168.188.2): Connection aborted: socket keep-alive expired.	Modbus TCP Master
	17.06.2024 16:07:05	Modbus/TCP Slave[0] (192.168.188.2): Connected.	Modbus TCP Master
	17.06.2024 16:06:57	Modbus/TCP Slave[0] (192.168.188.2): Connection aborted: socket keep-alive expired.	Modbus TCP Master
	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 dock settings	IoDrvEtherCAT
	17.06.2024 16:06:50	All slaves pre-operational	IoDrvEtherCAT
	17.06.2024 16:06:50		

```

P03_ChangeTCPSlave x
PROGRAM P03_ChangeTCPSlave
VAR
  iStep      :INT;
  bChange    :BOOL;
  ipAddr     :ARRAY[0..3] OF BYTE := [192,168,188,2];
  uiPort     :UINT:=502;
  uiBaudRate :UINT:=1000000;
  bSafeMode  :BOOL;
END_VAR

// Initialization
iStep := 0;
bChange := FALSE;
bSafeMode := FALSE;

// Main loop
WHILE TRUE LOOP
  // Step 1: Set operational mode
  IF bChange THEN
    iStep := 1;
    bChange := FALSE;
  END_IF

  // Step 2: Set safe operational
  IF iStep = 1 THEN
    bSafeMode := TRUE;
    iStep := 2;
  END_IF

  // Step 3: Synchronize slaves
  IF iStep = 2 THEN
    iStep := 3;
  END_IF

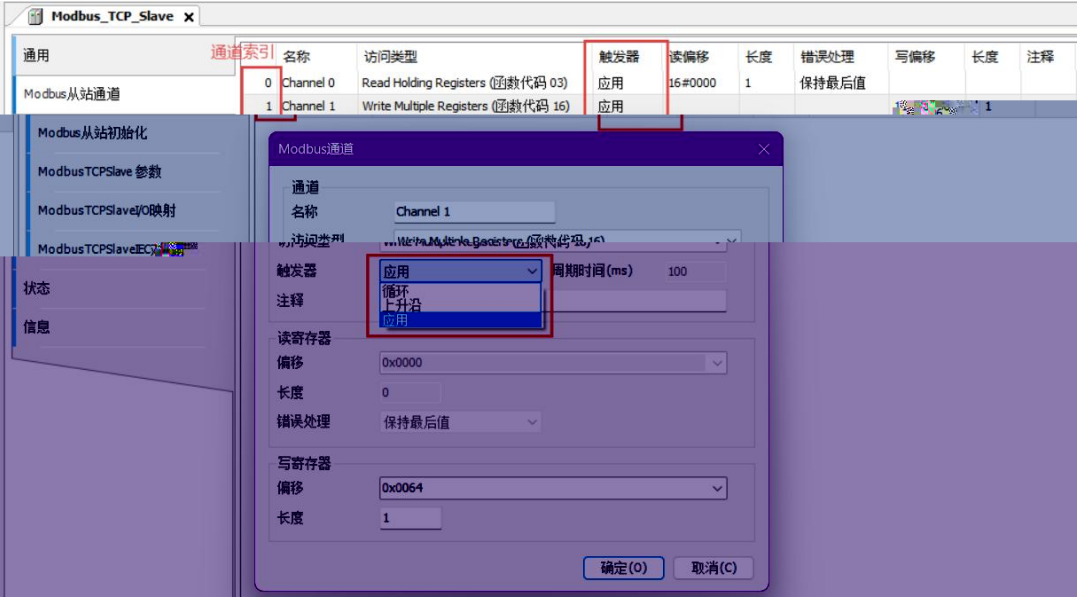
  // Step 4: Configure distributed dock settings
  IF iStep = 3 THEN
    iStep := 4;
  END_IF

  // Step 5: All slaves pre-operational
  IF iStep = 4 THEN
    iStep := 5;
  END_IF

  // Step 6: All slaves operational
  IF iStep = 5 THEN
    iStep := 6;
  END_IF

  // Step 7: Set operational mode
  IF iStep = 6 THEN
    bChange := TRUE;
    iStep := 0;
  END_IF
END LOOP

```



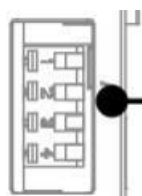
```
PROGRAM P04_TriggerModbusTCPChannel
VAR
    ModbusChannel_0 :ModbusChannel;
    ModbusChannel_1 :ModbusChannel;
    bRead            :BOOL;
    bWrite           :BOOL;
END_VAR

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=> );
```

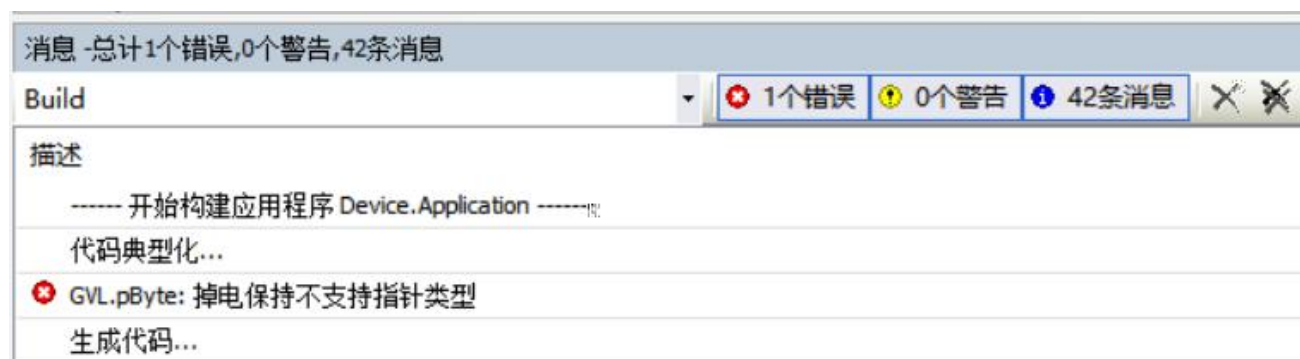
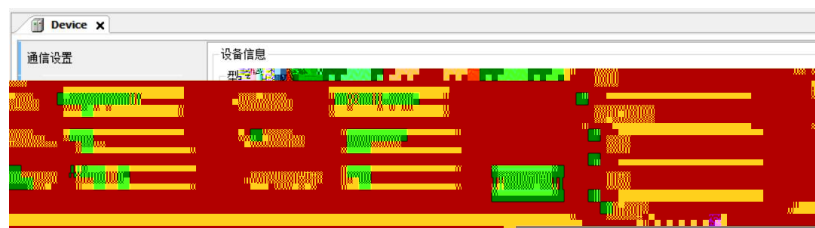
Q	Modbus			
%QX0.0-%QX8191.7	0-65535	01H		
		05H 0FH		
%IX0.0-%IX8191.7	0-65535	02H		
%MWO-%MW65535	0-65535	04H		
%MWO-%MW65535	0-65535	03H 06H 10H 17H		



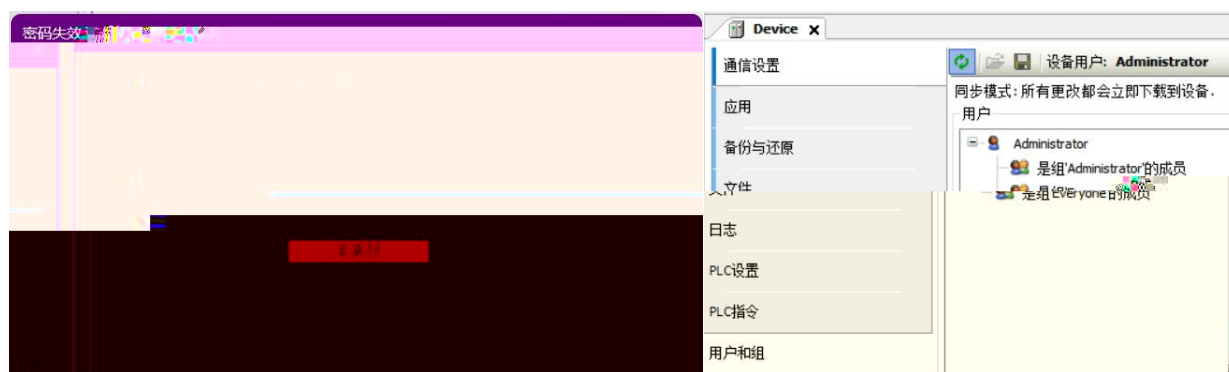
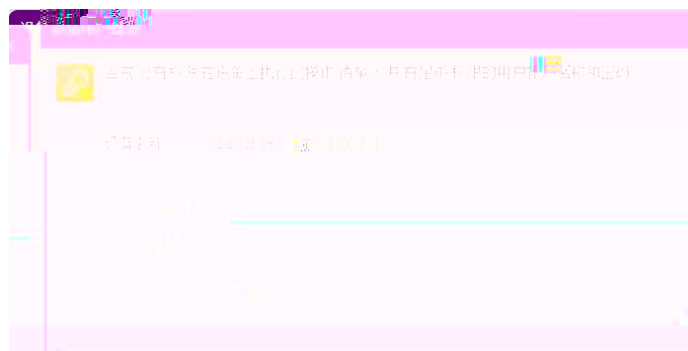
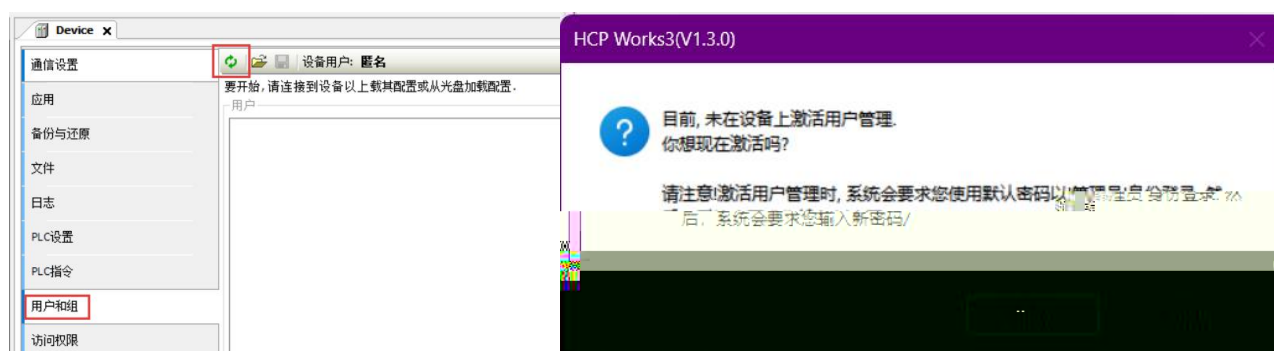


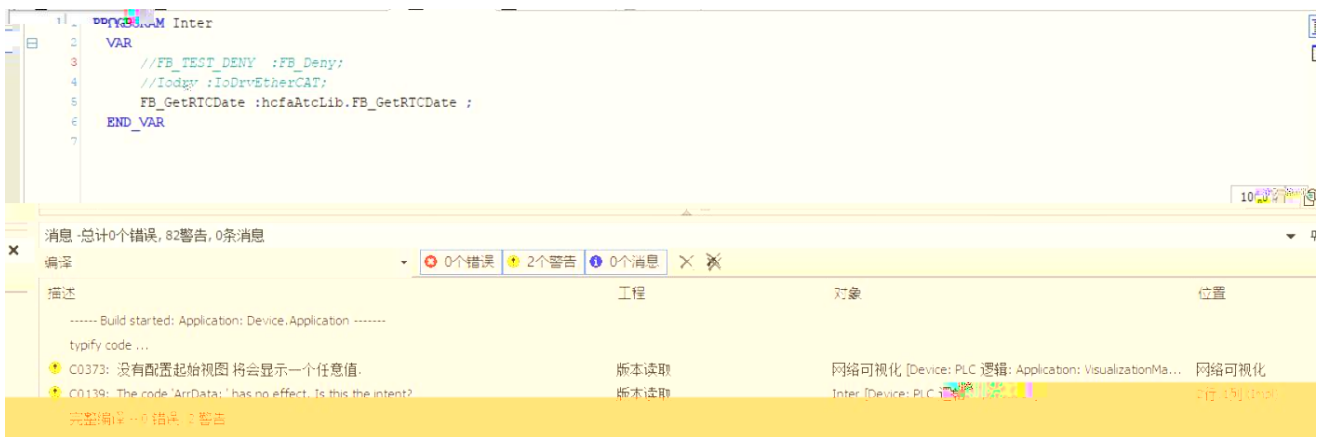
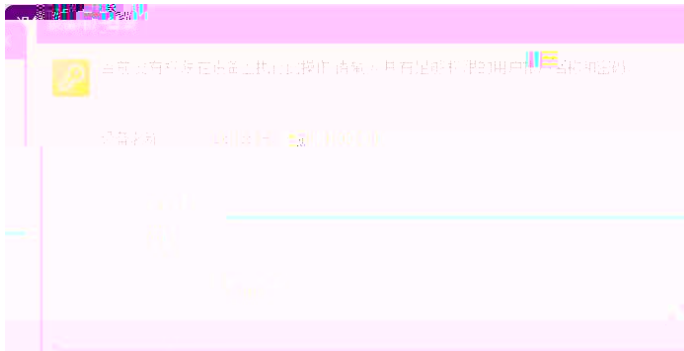
⑤ 拨码开关

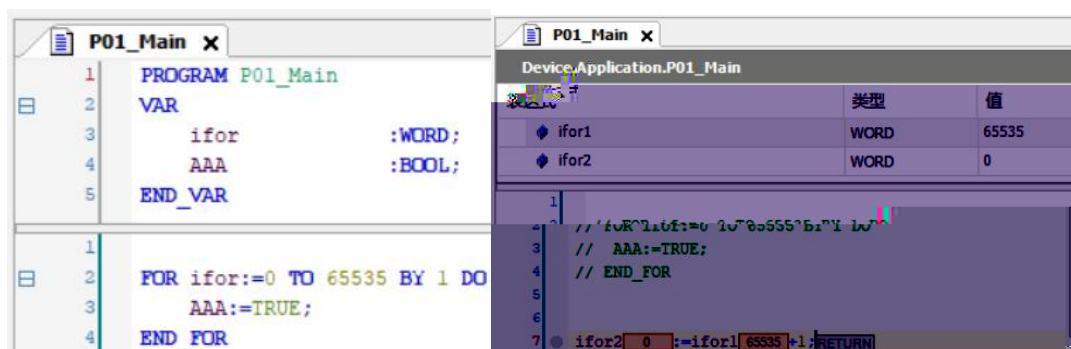
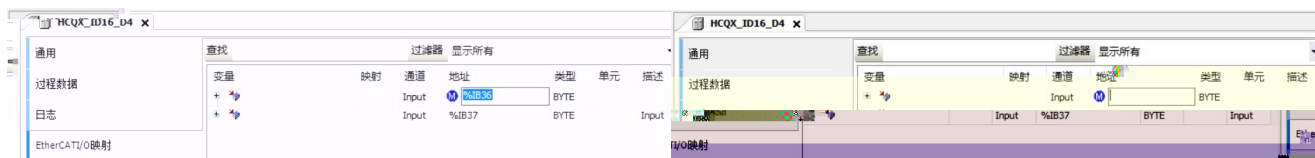
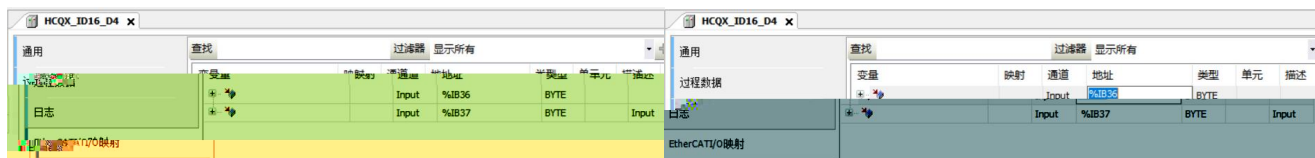
12	FN	FN/RUN/STOP switch	为复位EtherNet通讯接口IP地址，拨至RUN为运行控制器程序，拨到STOP为停止
10	STOP/RUN/FN 开关	STOP/RUN/FN dial switch	功能拨码，中间为运行状态，向右拨至FN，为port1/port2/port4 IP默认值



定义组合	解释
	8位编码字符的十六进制表示。
	美元符号
	单直引号
	换行
	新行
	换页
	回车
	标签







小心!

最终值<结束值>的值可能与循环变量的数据类型的上限不同。

如果循环变量的结束值等于循环变量的数据类型的上限，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。

库管理器

名称	命名空间	有效的版本
SM3_Robotics_Visu = SM3_Robotics_Visu, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Robotics_Visu	4.10.0.0
SM3_Transformation = SM3_Transformation, 4.10.0.0 (3S - Smart Software Solutions GmbH)	TRAFO	4.10.0.0
Standard = Standard, 3.5.17.0 (System)	Standard	3.5.17.0
StringUtils, 3.5.17.0 (System)	Stu	3.5.17.0

StringUtils, 3.5.17.0 (System)

- Public Parts
 - CharBufferPtr
 - Ansi
 - CharToUpper
 - StrCaseCmpA
 - StrCaseCmpEndA
 - StrCaseCmpStartA
 - StrCaseFindA
 - StrCmpA
 - StrCmpEndA
 - StrCmpStartA
 - StrConcatA
 - StrCpyA
 - StrDeleteA
 - StrFindA
 - StrIsNullOrEmptyA
 - StrLenA**
 - StrMidA
 - StrPadLeftA
 - StrPadRightA
 - StrReplaceA
 - StrToLowerA
 - StrToUpperA
 - StrTrimA

StrLenA (FUN)

FUNCTION StrLenA: DINT

This function determines the length of a string by searching for a terminating zero character. The function will do its work either with native IEC-Strings or with strings coming from external functions. It will return -1 if the given pointer to String is equal NULL

InOut

Scope	Name	Type	Comment
Return	StrLenA	DINT	
Input	StrData	CharBufferPtr	simply a pointer to byte

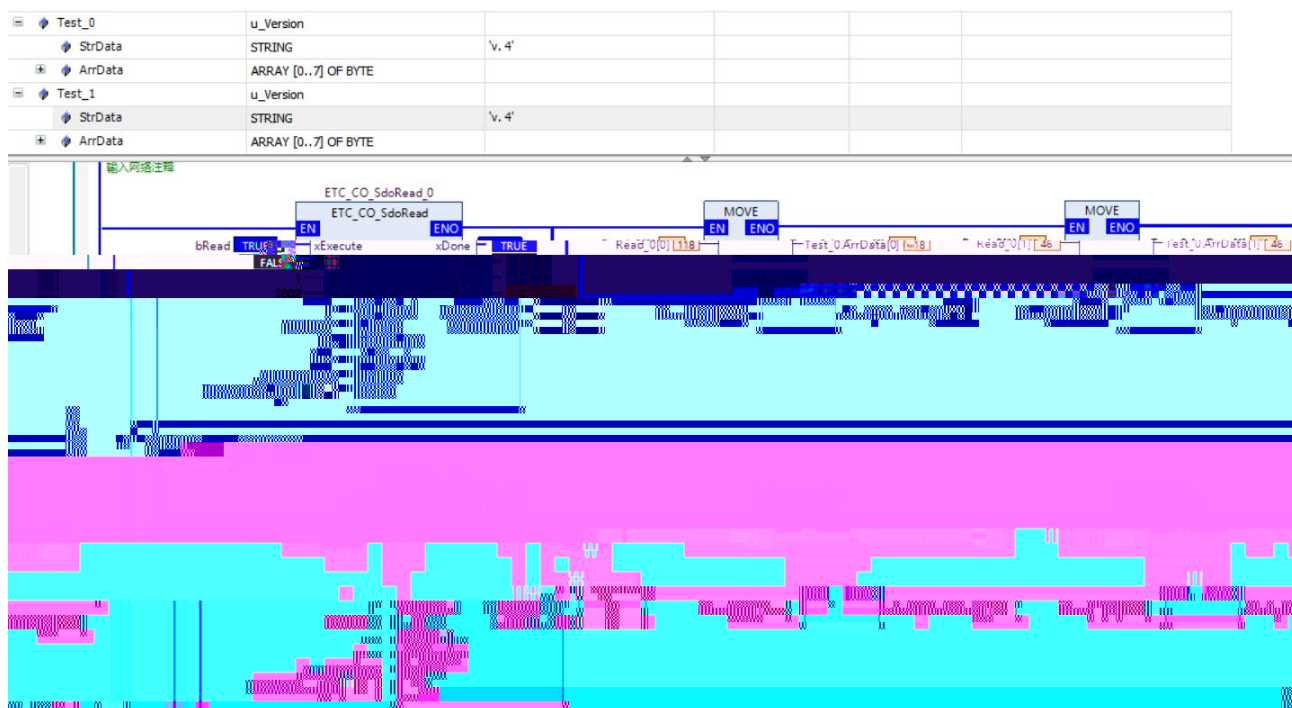
库管理器

名称	命名空间	有效的版本
CmpEventMgr = CmpEventMgr, 3.5.14.0 (System)	CmpEventMgr	3.5.14.0
CmpLog = CmpLog, 3.5.5.0 (System)	CmpLog	3.5.5.0
Collections, 3.5.17.0 (System)	Collections	3.5.17.0

Collections, 3.5.17.0 (System)

- Pair_DintDint

FUNCTION PSLE





//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

```
Axis_0.resetDrive()
    bReset:=bReset ;
    Execute:=bReset AND Axis_0.wCommunicationState<>100 ;
    bVirtualize:= ,
    bHome=> ,
    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>1 ,
    Position:= ,
    Done=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
```

//智能—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

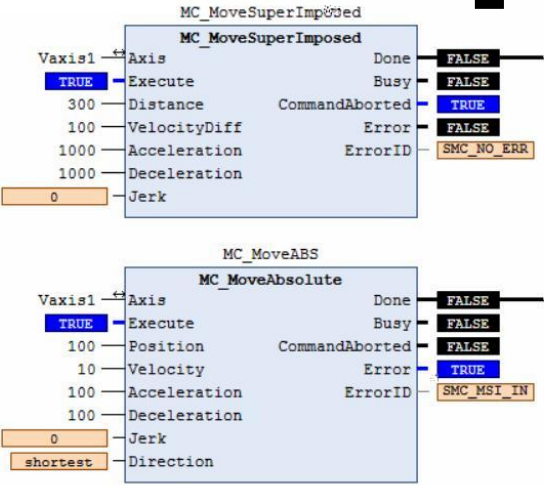
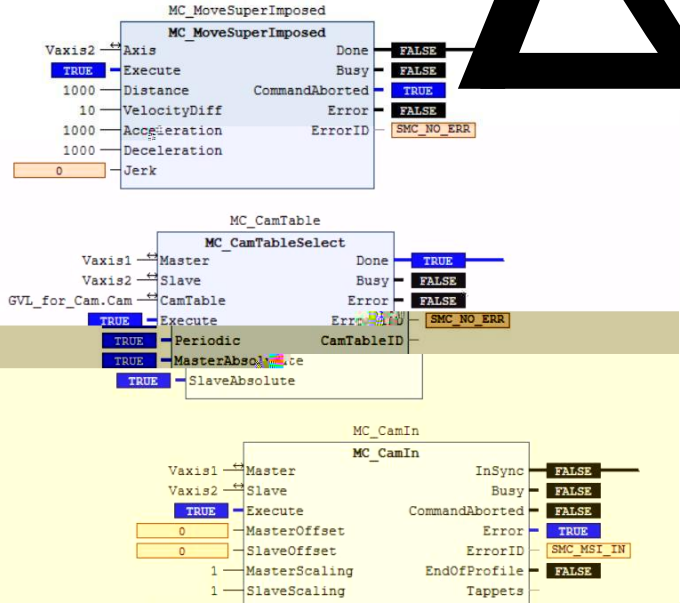
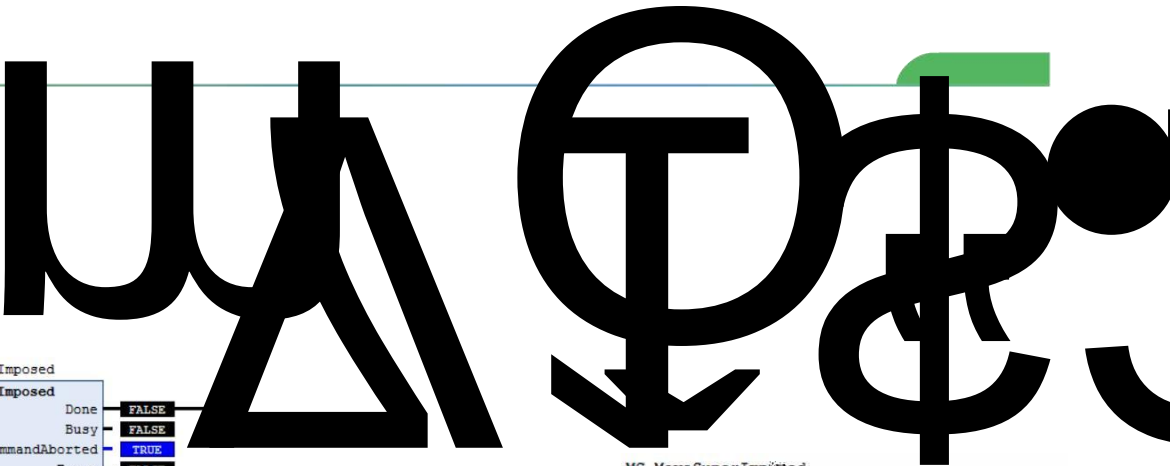
```
MC_Home_0(
    Axis:=Axis_0 ,
    Execute:=bHome AND Axis_0.nAxisState>1 ,
    Position:= ,
    Done=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
```

// Jog—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_Jog_0(
    Axis:=Axis_0 ,
    JogForward:=bJogF AND Axis_0.nAxisState>1 AND Axis_0.nAxisState<> ,
    JogBackward:=bJogB AND Axis_0.nAxisState>1 AND Axis_0.nAxisState<> ,
```

```
    Done=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
```

&© Ö“.



T; 0 \$?4090-\$?@

ct d) m; RljZ &:d)2C 'B%int4Et

Object 6080n:最大电机转速			
对象描述		对象入口描述	
属性	值	属性	值
索引	6080n		
名称	最大电机转速		

轴类型与限制

☐ 虚轴模式
☐ 模态
☒ 线性

软件限位

☒ 激活
 负向 [u]:
 正向 [u]:

软件错误处理

减速 [u/s²]:
 最大距离 [u]:

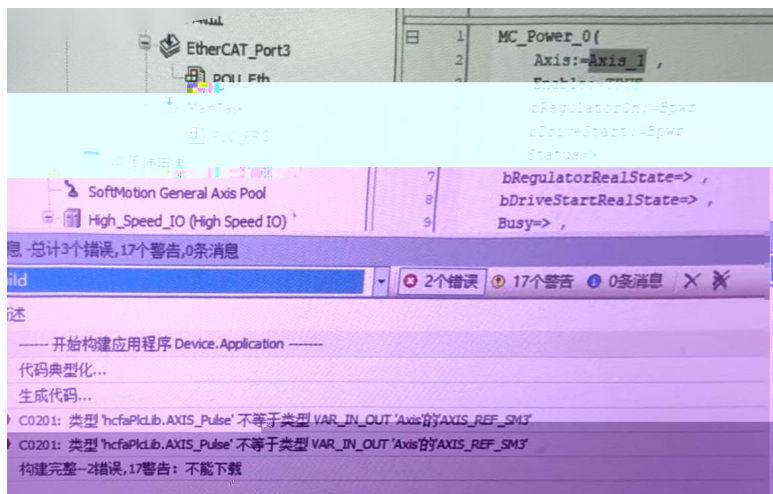
动态限制

速度 [u/s]:

加速度 [u/s²]:

减速度 [u/s²]:

加加速度 [u/s³]:



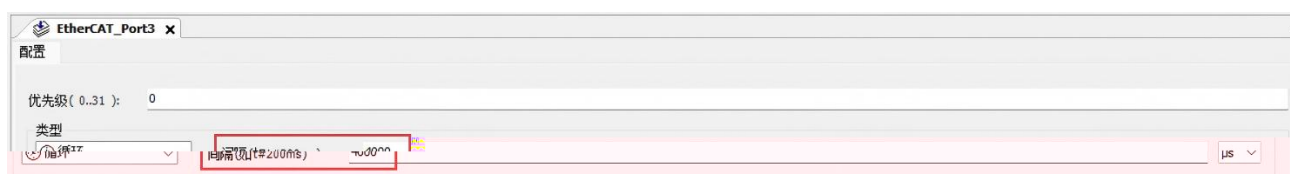
名称	类型	索引
☒ 16#1600 Outputs		
ControlWord	UINT	16#6040:00
Modes of Operation	SINT	16#6060:00
Target position	DINT	16#607A:00
Target velocity	DINT	16#60FF:00
Target torque	INT	16#6071:00
Max motor speed	UDINT	16#6080:00
☒ 16#1610 Outputs		
ControlWord	UINT	16#6040:00
Modes of Operation	SINT	16#6060:00
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
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
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
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



```

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;

```





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

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

调试

SM_Drive_ETC_GenericDSP402:I/O 映射

SM_Drive_ETC_GenericDSP402:IEC 对象

状态

信息

最大距离[u]: 0

ID: 0

动态限制

速度 [u/s]:	加速度 [u/s²]:	减速度 [u/s²]:	加加速度 [u/s³]:
30	1000	1000	10000

位置滞后监视
禁用驱动

滞后限制[u]: 1.0

在线

变量	设置值	实际值
位置 [u]	-2205.60	-2205.07
速度 [u/s]	0.00	-2.06
加速度 [u/s²]	0.00	-343.32
扭矩	0.00	0.00

状态: SMC_AXIS_STATE.errorstop

通讯: 操作 (100)

错误:

轴错误:

0 [16#00000000]

错误:

SMC_ERROR_SMC_DI_POSITIONLAGERROR

uiDriveInterfaceError:

0

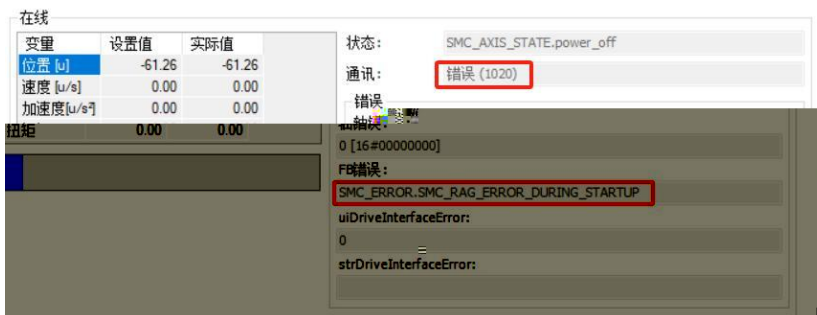
strDriveInterfaceError:



这个值等于轴的结构体中的fSetActTimeLagCycles 周期 (见“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方式实现)。
- **“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。
- **“保留当前状态”**：驱动器仍然位于当前状态，但是停止在运动中。

“滞后限制”：定义滞后限制。



状态和状态转化	操作	时间	操作
初始化(Init)	应用层没有通讯，从站只能读写ESM芯片寄存器。	26.03.2024 20:37:05	SDO write ok: Address: 1002 Index: 16
初始化向预运行转化 Init To Pre-OP(IP)	主站配置过程数据使用的地址寄存器。 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。	26.03.2024 20:37:05	SDO write ok: Address: 1002 Index: 16
预运行(Pre-OP)	应用层邮箱通讯。	26.03.2024 20:37:05	SDO write ok: Address: 1002 Index: 16
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PMU。 主站写入状态控制寄存器，请求Safe-OP状态。	26.03.2024 20:37:05	Set safe operational
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。	26.03.2024 20:37:05	Synchronize Slaves
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。	26.03.2024 20:37:05	Configure distributed clock settings
运行状态(Op)	输入输出全部有效。	26.03.2024 20:37:05	All slaves pre-operational

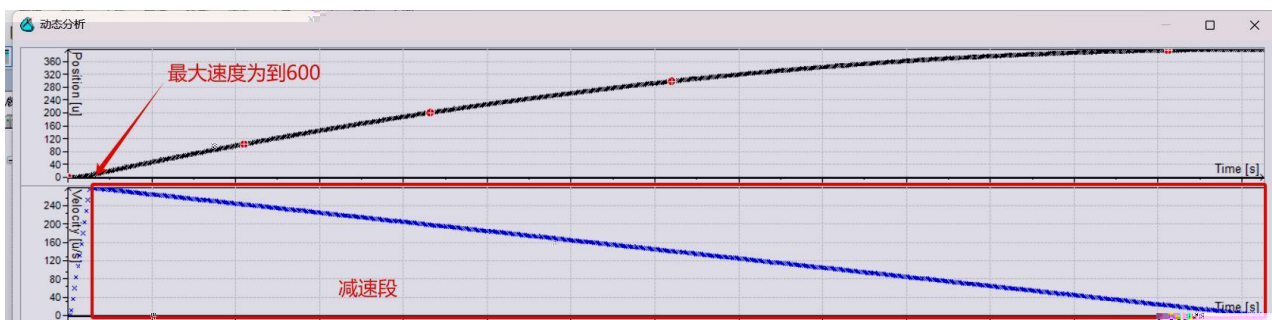
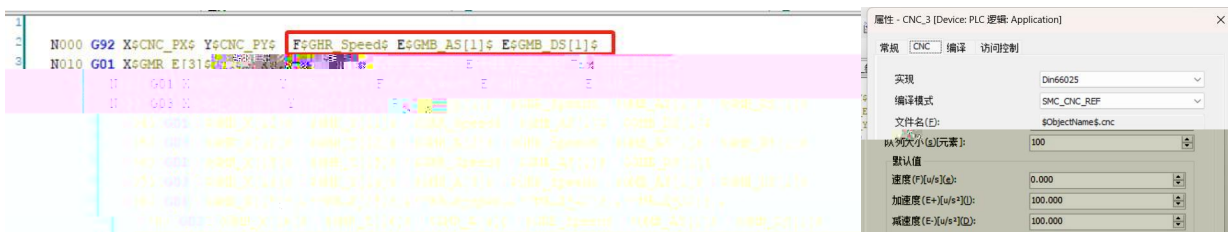
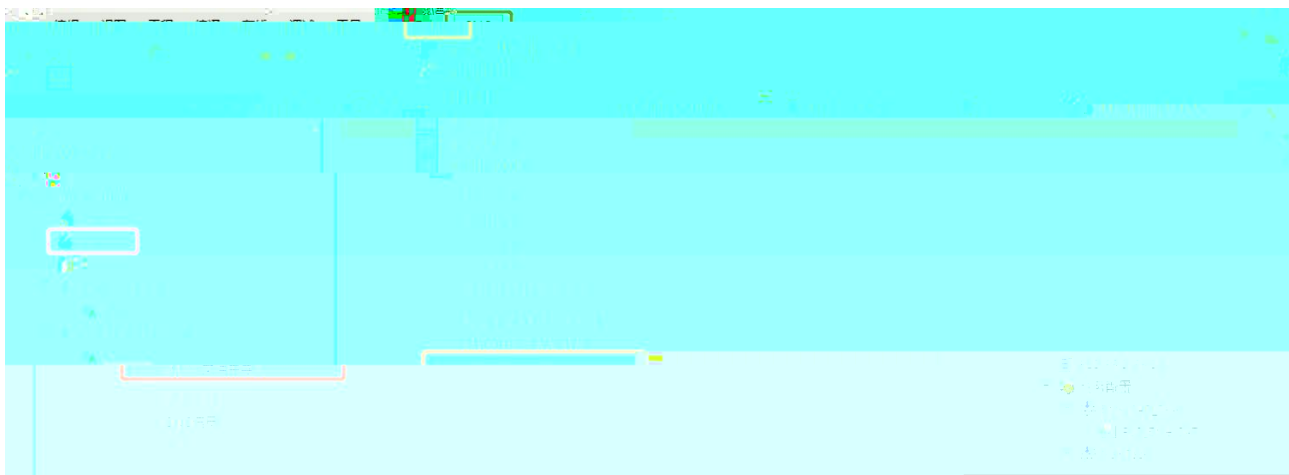
轴类型为旋转模式

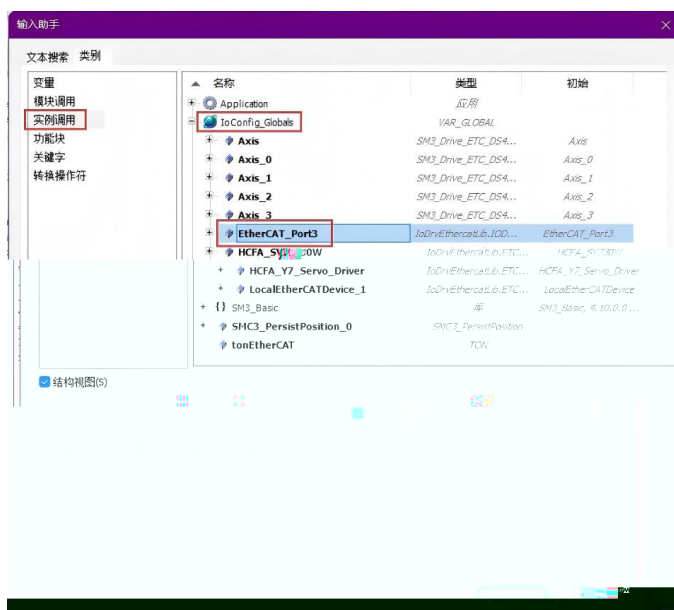
单位换算公式：

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

M:电机 W:工作台



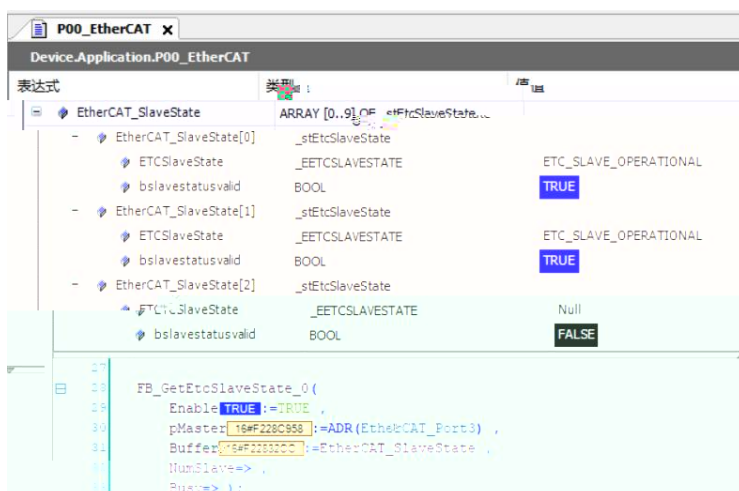




```

23
24 EtherCAT_Port3(
25     xRestart:= ,
26     xStopBus:= ,
27     xConfigFinished=> ,
28     xDistributedClockInSync=> ,
29     xError=> ,
30     xSyncInWindow=> );
31
32
33
34
35
36
37
38
39

```



```

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

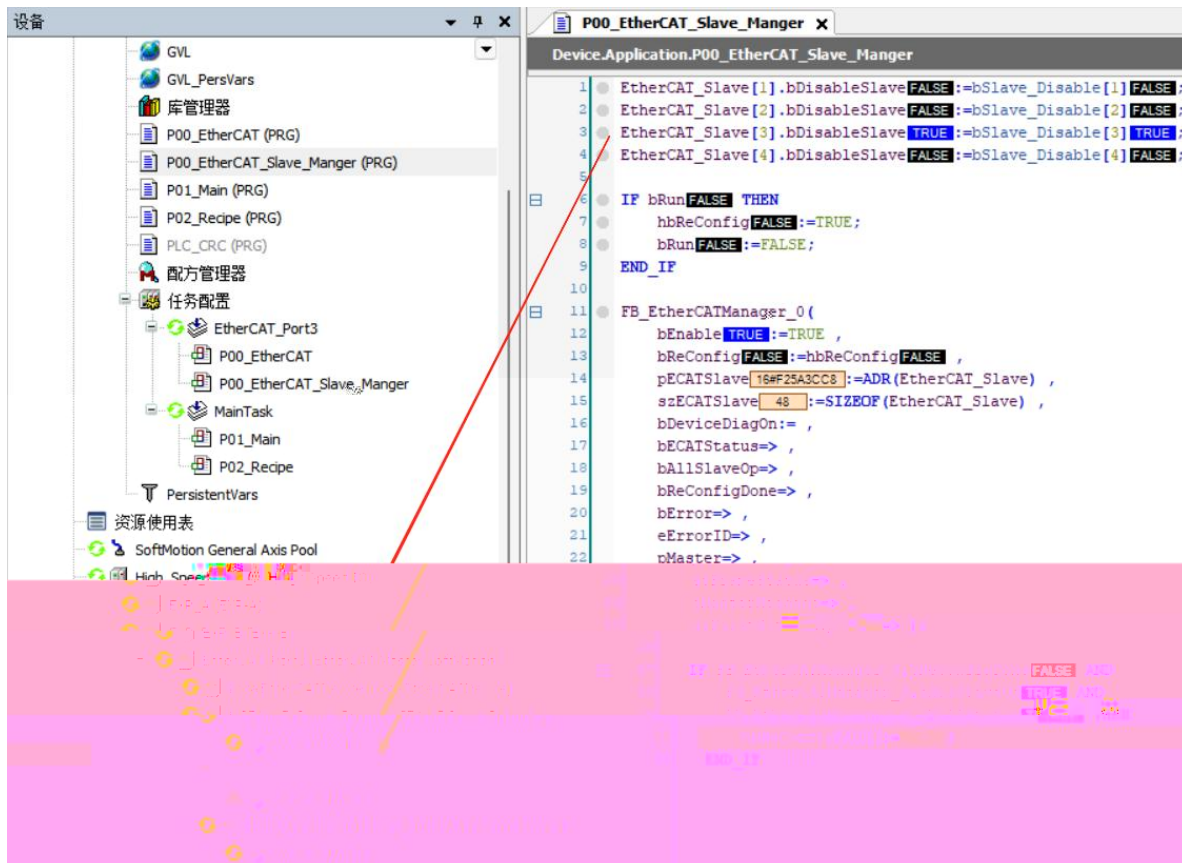
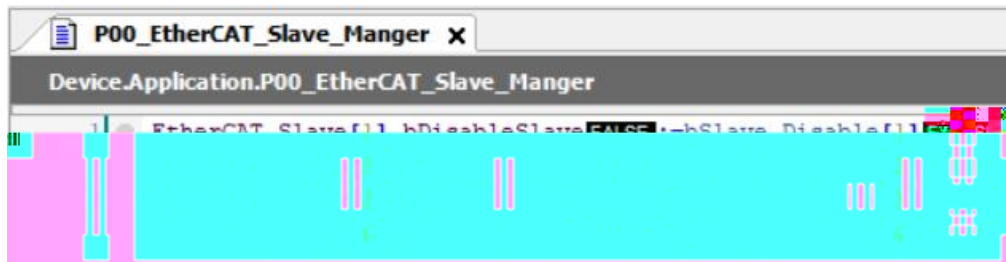
1  EtherCAT_Slave[1].bDisableSlave:=bSlave_Disable[1];
2  EtherCAT_Slave[2].bDisableSlave:=bSlave_Disable[2];
3  EtherCAT_Slave[3].bDisableSlave:=bSlave_Disable[3];
4  EtherCAT_Slave[4].bDisableSlave:=bSlave_Disable[4];

6  IF bRun THEN
7      hbReConfig:=TRUE;
8      bRun:=FALSE;
9  END_IF

11 FB_EtherCATManager_0 :=
    pEnable-> ;
    pReConfig:=hbReConfig ;
    pECATSlave:=ADR(EtherCAT_Slave) ;
    ecSlave:=SIZEOF(EtherCAT_Slave) ;
    pDeviceDiagIn:= ;
    pECATStatus:= ;
    pAllSlavesIn-> ;
    pSlaveConfigIn-> ;
    pRun-> ;
    pECATID-> ;
    pMaster-> ;
    pTransStatus-> ;
    pMasterMessage-> ;
    pMasterModifySlave-> ;

11 FB_EtherCATManager_0.pEnable AND
    FB_EtherCATManager_0.pAllSlavesIn AND
    FB_EtherCATManager_0.pECATStatus=TRUE
30     hbReConfig:=FALSE;
31 END_IF

```



```

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

```

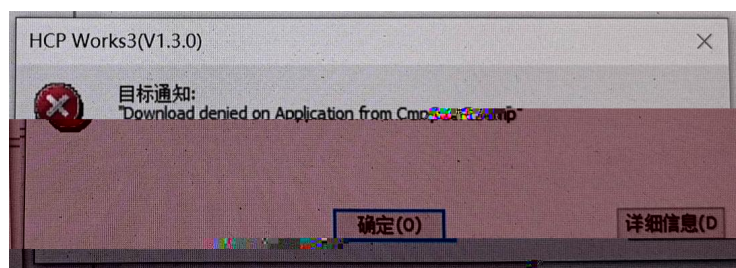
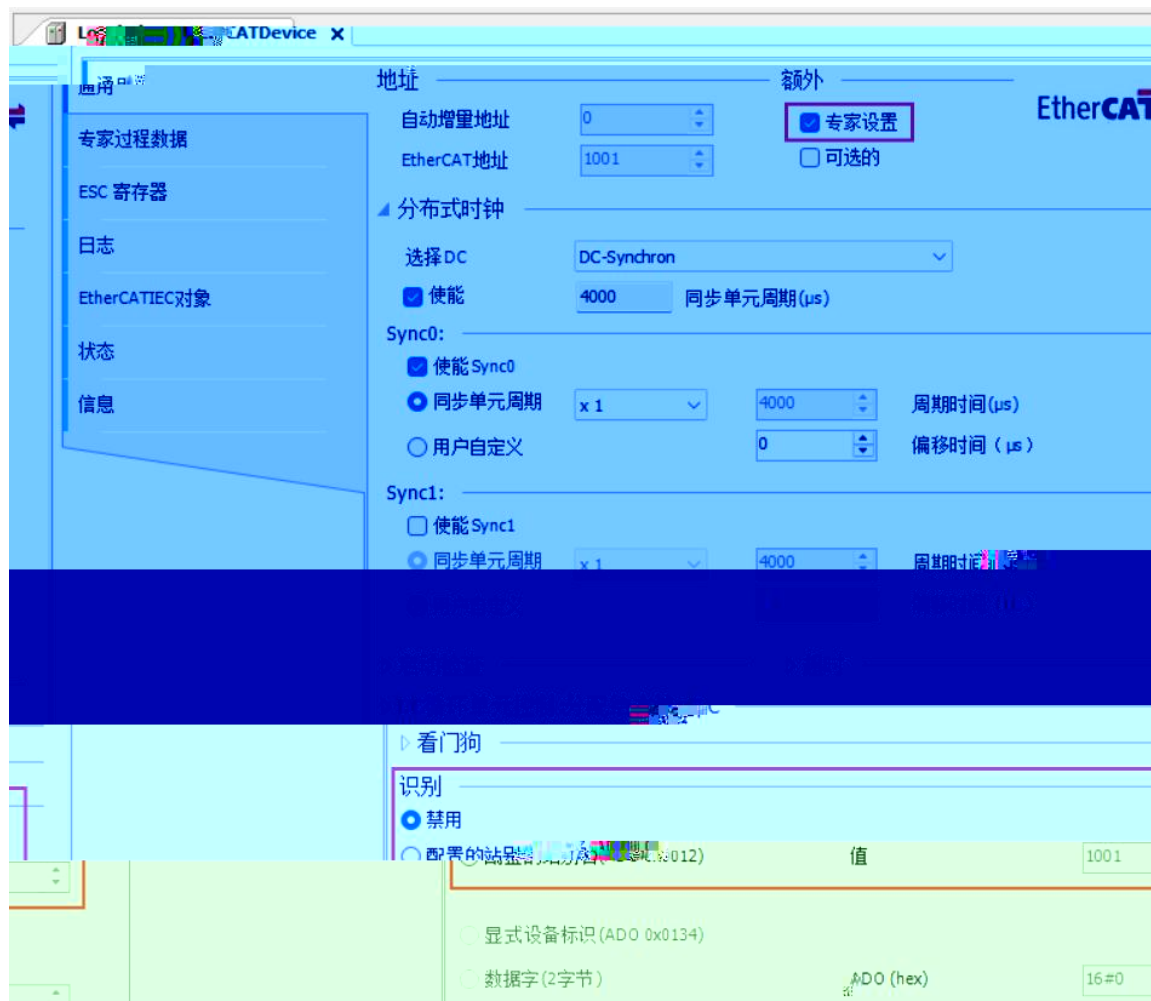
```

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);

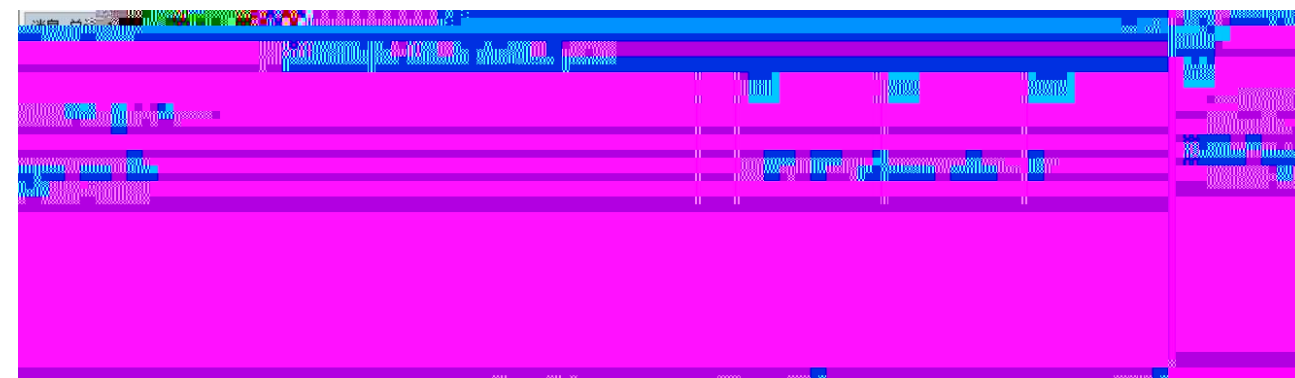
```


ⓘ	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_Task1]	CmpIecTask

fData[NaN]:=fData2[0]/fData2[0];RETURN



厂商	时间戳	描述	组件
	24.11.2023 22:51:38	ESQUBERBOSU/CM*Application/Array3, offset=11856	CmpApp
	24.11.2023 22:51:38	*EXCEPTION* [GlobalExit] code: App=[Application], Exception=[AccessViolation]	CmpApp



```

1 {attribute 'no_check'}
2 PROGRAM POU_EtherCAT
3 VAR
4     tonEtherCAT          :TON;
5     SMC3_PersistPosition_0 :SMC3_PersistPosition;
6     FB_GetEtcSlaveState_0 :FB_GetEtcSlaveState;
7     EtherCAT_SlaveState   :ARRAY[0..9] OF _stEtcSlaveState;
8     bRestart              :BOOL;
9
10 VAR
11     bEnable :BOOL;
12
13 END PROGRAM

```

设备

- QOP_测试
 - Device (HCQOP-1200-D4)
 - 设备组态
 - 本地模块配置
 - PLC逻辑
 - Application
 - 库管理器
 - P00_EtherCAT (PRG)
 - P01_AxisCtrl (PRG)
 - PLC_Func (PRG)
 - 任务配置

添加 用于隐式检查的POU

创建应用的特殊检查函数 (用于检查数组边界,除法和指针)

可用函数(U)

- ☒ 绑定检查
- ☐ 除法检测
- ☐ 范围检查
- ☐ LRange检查
- ☐ 指针检查

注意:添加检测函数会导致工程完全重新编译且无法执行在更改。

添加

```

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

```

```

1 // 隐式生成代码:只是实现代码的建议
2 IF index[0] < lower[0] THEN
3   CheckBounds[0] := lower[0];
4   dErrCount[0] := dErrCount[0] + 1;
5 ELSIF index[0] > upper[0] THEN
6   CheckBounds[0] := upper[0];
7   dErrCount[0] := dErrCount[0] + 1;
8 ELSE
9   CheckBounds[0] := index[0];
10 END_IF

```

```

1 // 隐式生成代码:只是实现代码的建议
2 IF index[0] < lower[1] THEN
3   CheckBounds[1] := lower[1];
4   dErrCount[0] := dErrCount[0] + 1;
5 ELSIF index[0] > upper[9] THEN
6   CheckBounds[1] := upper[9];
7   dErrCount[0] := dErrCount[0] + 1;
8 ELSE
9   CheckBounds[1] := index[0];
10 END_IF

```

调用堆栈

应用: Device Application

```

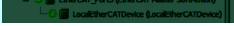
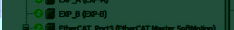
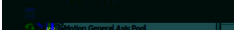
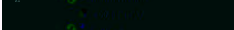
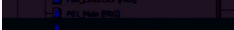
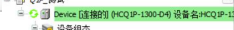
1 IF bStart THEN
2   FOR ifor[0] := 0 TO 9 DO
3     arTest[ifor[0]] := arTest[ifor[0]] + 1;
4   END_FOR
5 END_IF

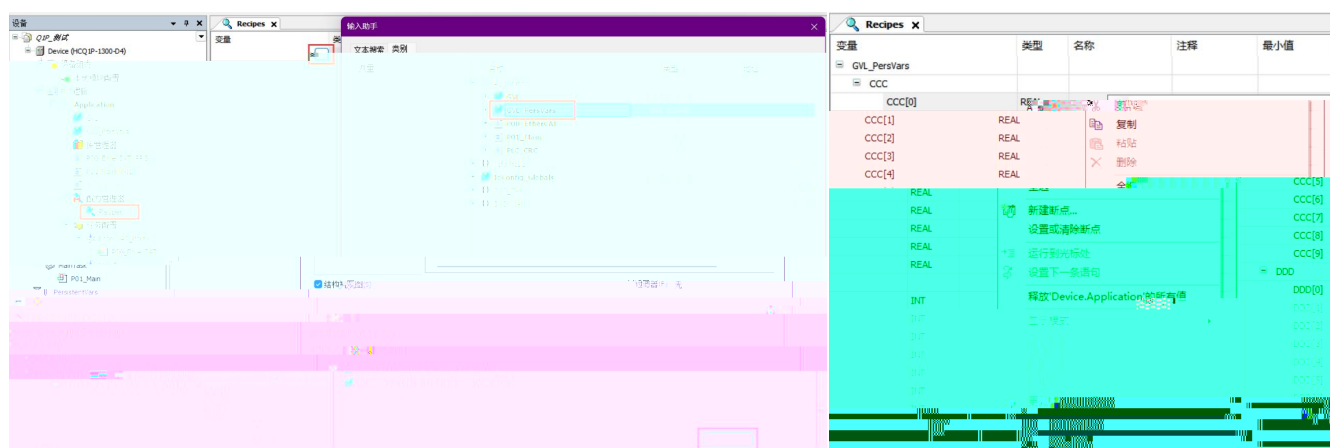
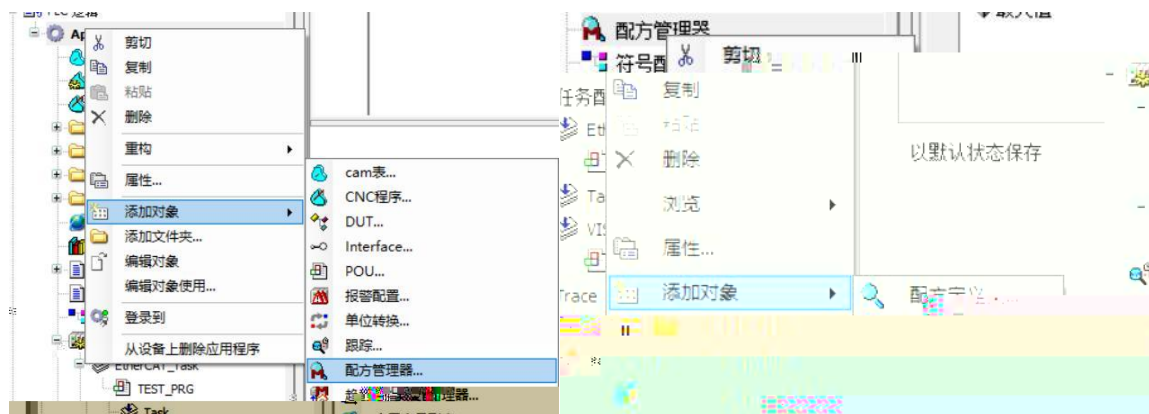
```

RUN:=bRun ,



The screenshot shows the 'PersistentVariables' table in the 'PLC' project. The table has columns for 'Name', 'Data Type', 'Value', and 'PersVars'. The 'PersVars' column is highlighted in red. The table lists variables like CCC[0], CCC[1], CCC[2], etc., with their data types (REAL, INT) and values. The 'PersVars' column is highlighted in red.





```

//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

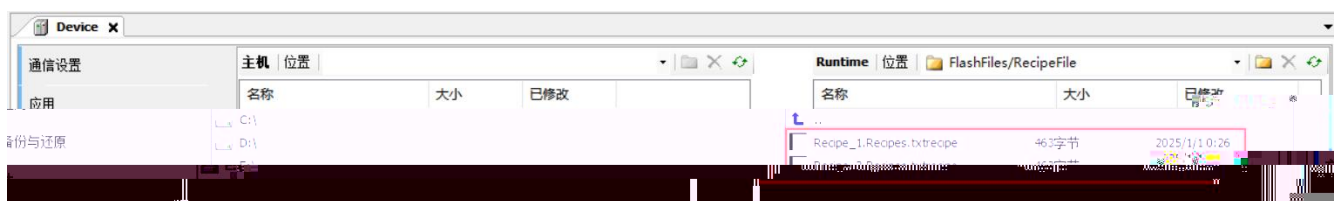
//生成配方名称
strRecipeNum:=TO_STRING(wRecipeNum);
strRecipeName:=CONCAT('Recipe_',strRecipeNum);

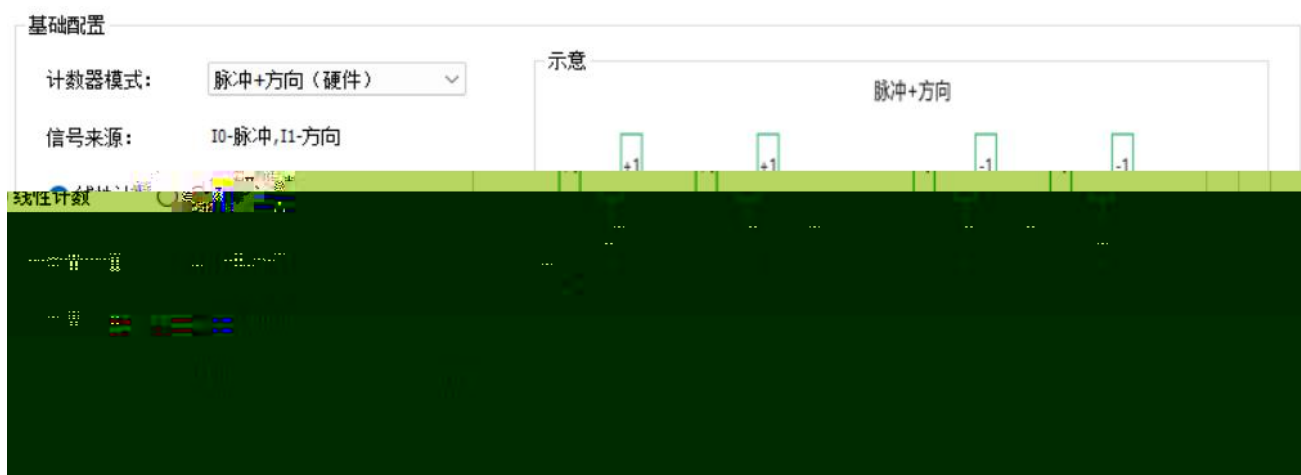
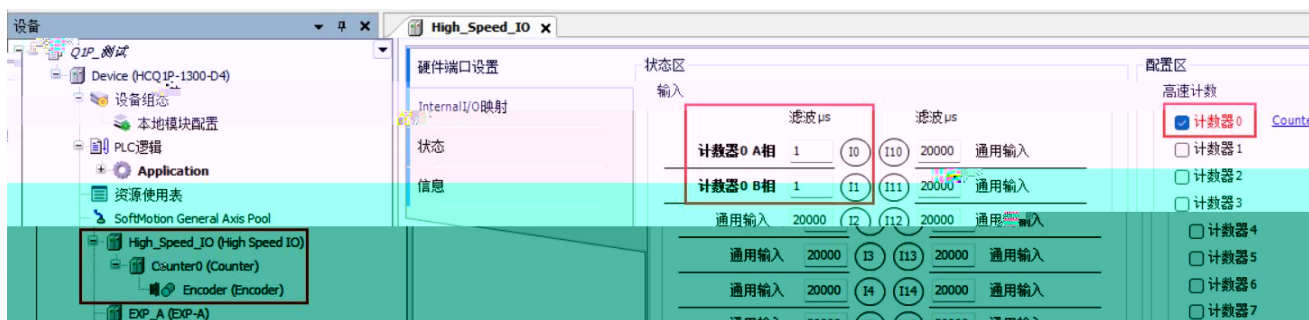
//创建配方文件
IF bCreateRecipe AND bSetStoragePath_OK THEN
    IF RecipeManCommands_0.CreateRecipe(RecipeDefinitionName:='Recipes',RecipeName:=strRecipeName)=16#0 THEN
        bCreateRecipe_OK:=TRUE;
        bCreateRecipe:=FALSE;
    END_IF
END_IF

//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，并写入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

```





库管理器

- 3SLicense = 3SLicense, 3.5.17.0 (3S - Smart Software Solutions GmbH)
- BreakpointLogging = Breakpoint Logging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)
- hcfaPklLib = hcfaPklLib, 1.3.3.3 (HCFA)
- IODrvEtherCAT = IODrvEtherCAT, 3.5.17.0 (3S - Smart Software Solutions GmbH)
- IoStandard = IoStandard, 3.5.13.0 (System)
- SM3_Basic = SM3_Basic_4_10.0.0 (3S - Smart Software Solutions GmbH)
- hcfaPklLib, 1.3.3.3 (HCFA)
 - Algorithm
 - Communication
 - DeviceSet
 - HighSpeed_IO
 - HighSpeed_V1
 - HighSpeed_V2
 - Data types

P01_Encoder_Test

```

1 PROGRAM P01_Encoder_Test
2 VAR
3     FB_Encoder_0 :FB_Encoder;
4     FB_PresetValue_0:FB_PresetValue;
5     bSoftwarePreset :BOOL;
6     编码器实时位置 :REAL;
7     编码器实时速度 :REAL;
8 END_VAR
9
10 //高速输入编码器
11 //编码器使能
12 FB_Encoder_0(
13     Encoder:=Encoder_0 ,
14     bEnable:=TRUE ,
15     bSoftDirection:= ,
16     bResetOverUpperRange:= ,
17     bResetOverLowerRange:= ,
18     xActive=> ,
19     xOverUpperRange=> ,
20     xOverLowerRange=> ,
21     eDirections=> ,
22     lrActPositon=>编码器实时位置 ,
23     lrActVelocity=>编码器实时速度 ,
24     xBusy=> ,
25     xError=> ,
26     eErrorID=> );
27
28 //位置预设
29 FB_PresetValue_0(
30     Encoder:=Encoder_0 ,
31     bEnable:=TRUE ,
32     bSoftwarePreset:=bSoftwarePreset ,
33     bResetHardwarePresetValueFlag:= ,
34     bResetSoftwarePresetValueFlag:= ,
35     xHardwarePresetValueFlag=> ,
36     xSoftwarePresetValueFlag=> ,
37     xBusy=> ,
38     xError=> ,
39     eErrorID=> );

```

脉冲轴

- ☒ 脉冲轴 0 [Pulser0](#)
- ☐ 脉冲轴 1
- ☐ 脉冲轴 2
- ☐ 脉冲轴 3
- ☐ 脉冲轴 4
- ☐ 脉冲轴 5
- ☐ 脉冲轴 6
- ☐ 脉冲轴 7

输出

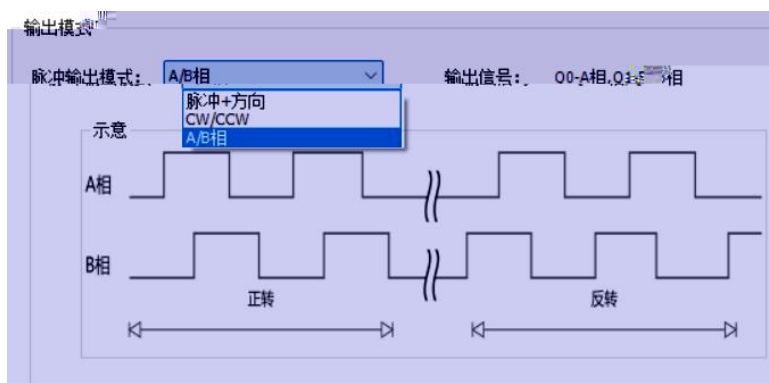
脉冲轴 0 A相 Q0 Q10 通用输出

脉冲轴 0 B相 Q1 Q11 通用输出

通用输出 Q2 Q12

High_Speed_IO (High Speed IO)

- Counter0 (Counter)
- Encoder (Encoder)
- Pulser0 (Pulse)
- Axis_3 (Axis)

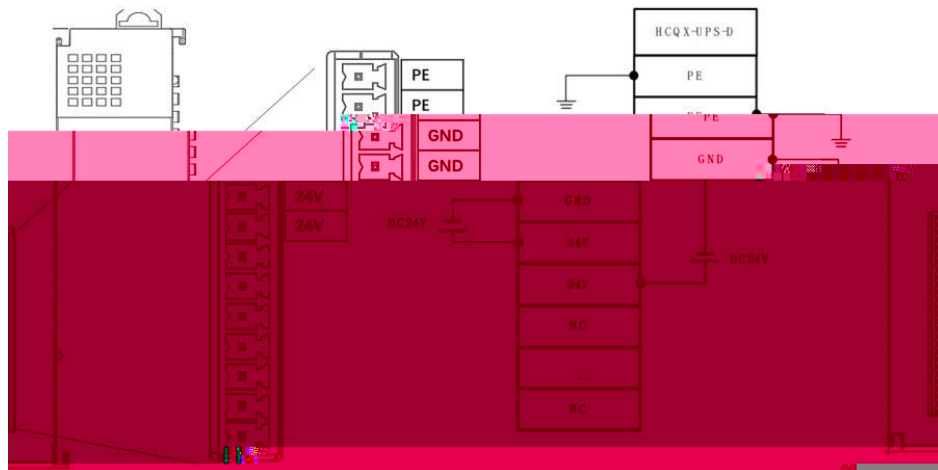


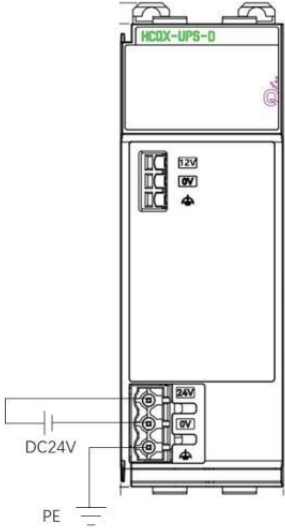
```

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

1 //-----
2 //高速输出轴
3 //脉冲轴使能
4 MC_Power_0(
5     Axis:=Axis_0 ,
6     Enable:=TRUE ,
7     bRegulatorOn:=bPower ,
8     bDriveStart:=bPower ,
9     Status=> ,
10    bRegulatorRealState=> ,
11    bDriveStartRealState=> ,
12    Busy=> ,
13    Error=> ,
14    ErrorID=> );
15
16 //脉冲轴点动
17 MC_Jog_0(
18     Axis:=Axis_0 ,
19     JogForward:=bJogF ,
20     JogBackward:=bJogB ,
21     Velocity:=rJogVel ,
22     Acceleration:=1000 ,
23     Deceleration:=1000 ,
24     Jerk:=10000 ,
25     Busy=> ,
26     CommandAborted=> ,
27     Error=> ,
28     ErrorID=> );

```





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 |
| | 7 | 17 |
| PE | 8 | 18 PE |

MAX
模块
16

15.2±0.1

13±0.1

注：PE（8，18）之间内部导通，任意接一个即可