

苏州市凌臣采集计算机有限公司 Suzhou Lingchen Acquisition Computer Co.,Ltd.

Lingchen-LC1800 系列 PLC 快速应用手册 V_1.0



2023 年 8 月

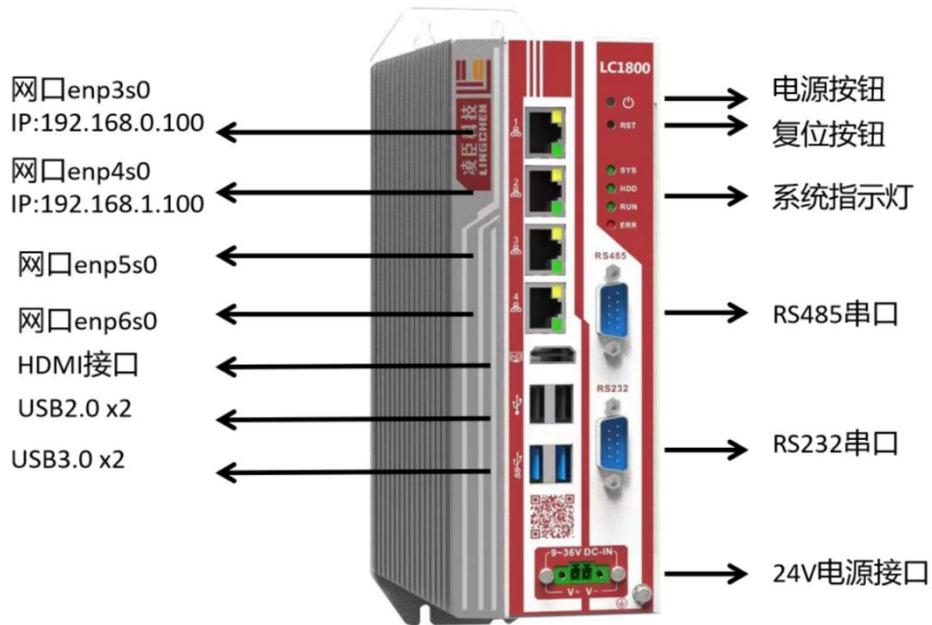
修改记录

序号	版本	修改时间	修改人	备注
1	V_1.0	2023.08	PLC 研发	LC1800 快速应用手册初版

目录

硬件接口展示	4
参数简介	4
LED 指示灯说明	5
1. 设备存储库的使用	6
2. 库文件的使用	7
3. 建立一个工程并下载调试	10
4. 单轴简单用户控制程序编写	24
5. 基于 EtherCAT 通信的电子凸轮程序例程	27
6. 添加拓展 IO (以 LC1100 和 1488、2488 模块为例)	31
7. Modbus	35
8. 断电保持功能的使用	52
9. OPC UA 例程	54

硬件接口展示



参数简介

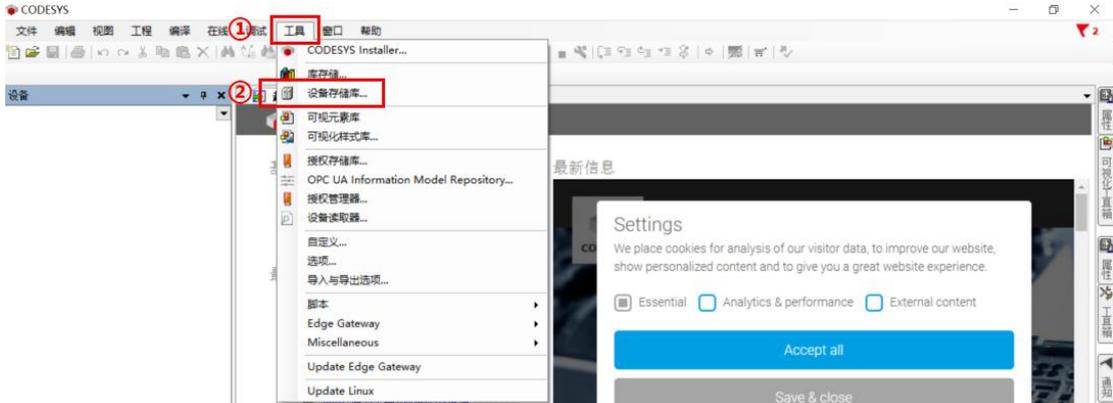
CPU	Celeron
内存	8G
SSD	128G
USB	2 路 USB 2.0 2 路 USB 3.0
网口	4 路
HDMI	1 个
复位	RST 按键
串口	2 个 RS232 (COM1) *1; RS485 (COM2) *1
电源	9~36V DC (配适配器)

LED 指示灯说明

序号	LED	名称	功能描述	状态颜色	定义
1	SYS	系统指示灯	Runtime 运行状态	绿色	常灭/常亮: runtime 运行异常 闪烁: runtime 正常运行
2	HDD	硬盘指示灯	硬盘状态	绿色	常灭: 未检测到硬盘 闪烁: 硬盘工作正常
3	RUN	运行指示灯	程序运行状态	绿色	常灭: 用户程序未运行 常亮: 用户程序运行中
4	ERR	错误指示灯	运行错误	红色	常灭: 正常 常亮: 系统故障报警

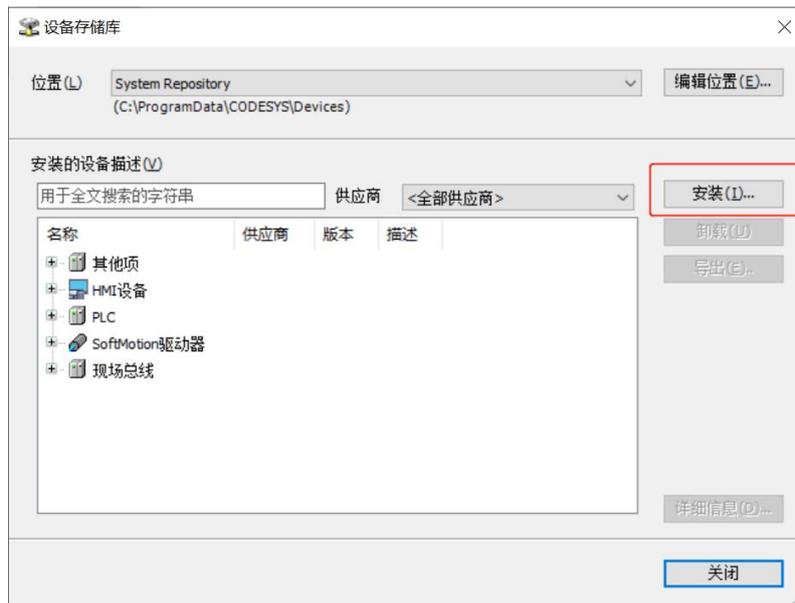
1. 设备存储库的使用

1.1 打开软件

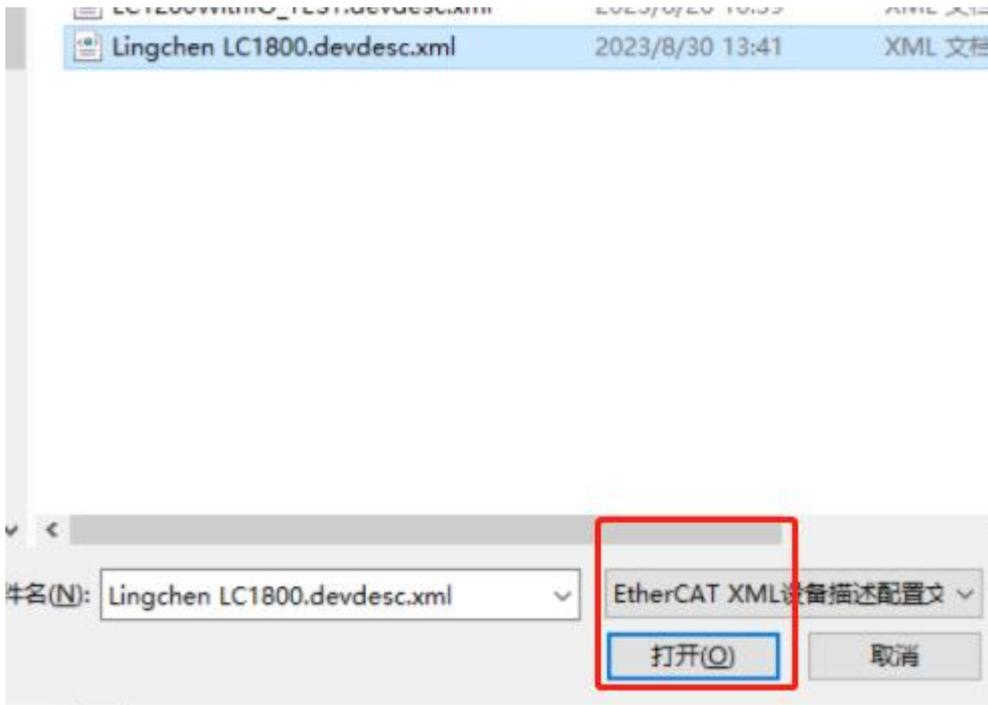


打开 CODESYS V3.5 SP18 软件，在界面最上方的菜单栏找到“工具”，点击“工具 - 设备存储库”；

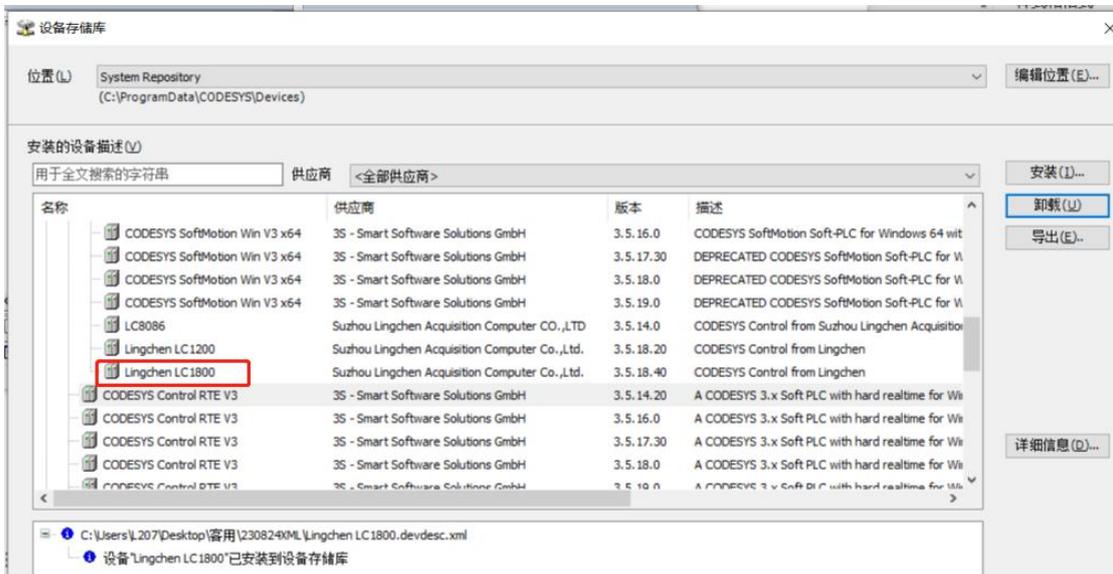
1.2 点击“安装”；



1.3 找到设备描述文件 (*.xml) 所存放的位置，选择相应设备描述文件，点击“确定”（或者直接双击相应设备描述文件）；

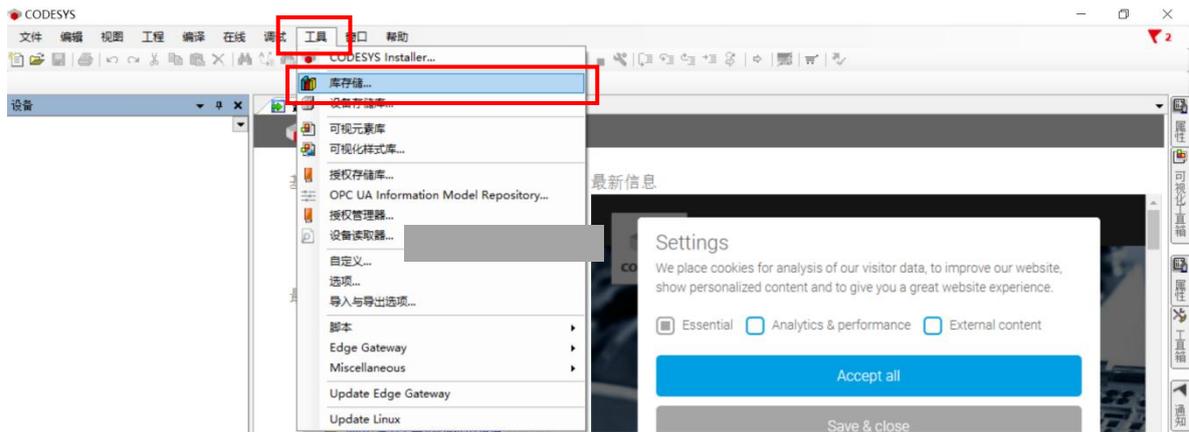


1.4 安装成功，点击“关闭”；

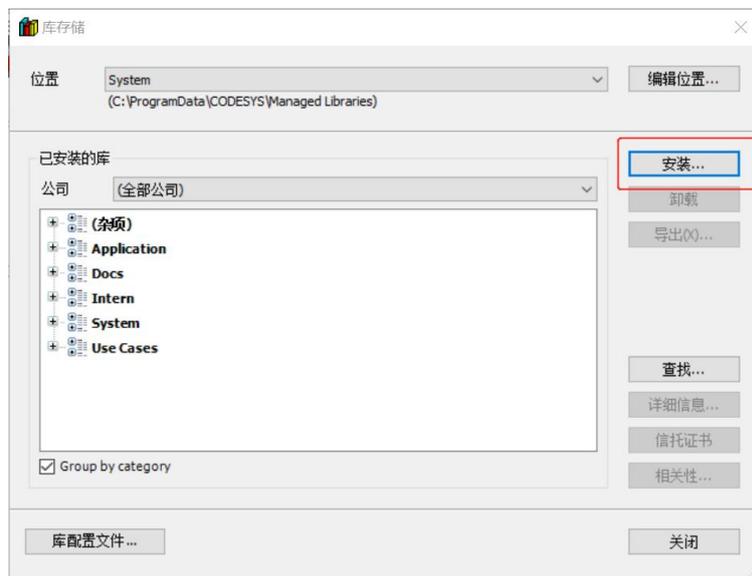


2. 库文件的使用

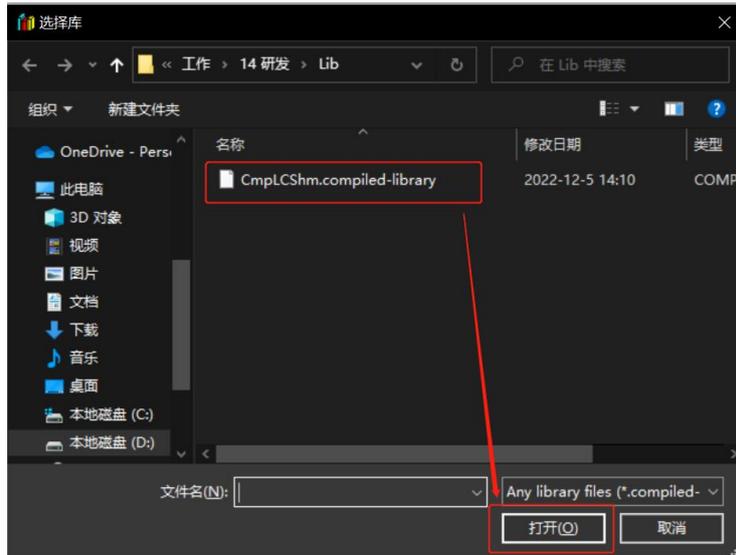
2.1 同样在菜单栏中找到“工具”，点击“工具 - 库存储”；



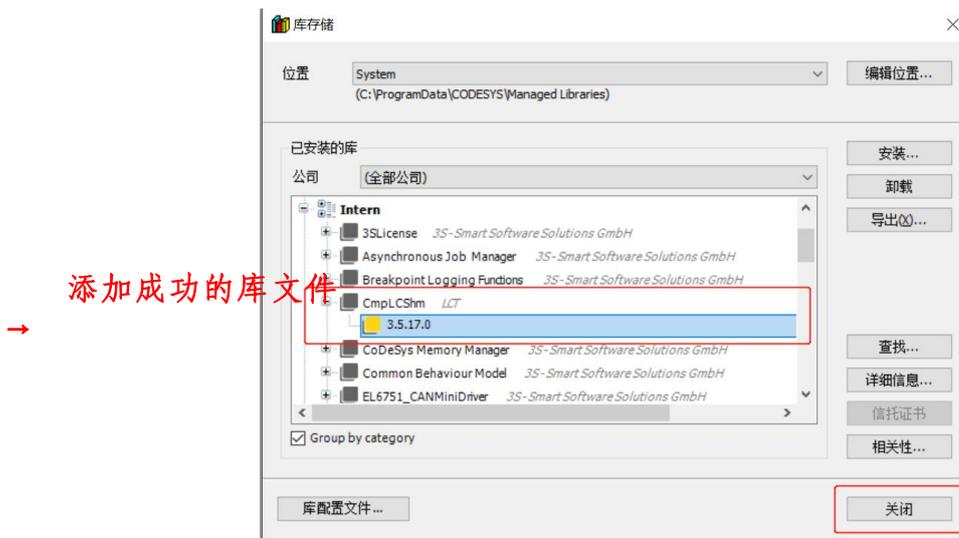
2.2 点击“安装”;



2.3 找到存放库文件的路径, 选择要添加到存储的库文件, 点击“打开”(或者直接双击要添加到存储的库文件);



2.4 添加成功，点击“关闭”。

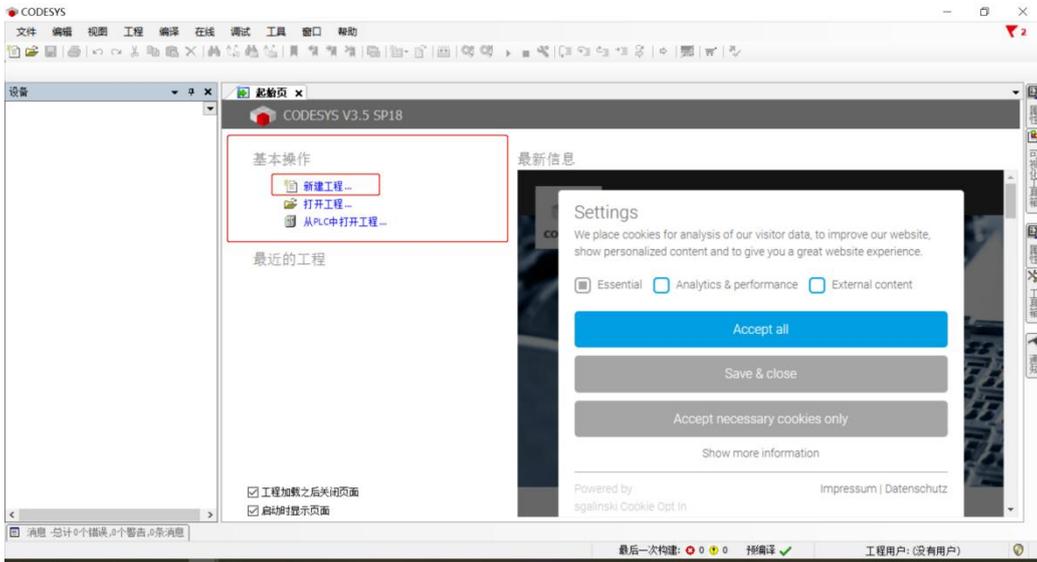


2.5 添加完成后，还需要安装到工程文件中，才能调用（详情见“3. 建立一个工程并下载调试”）。

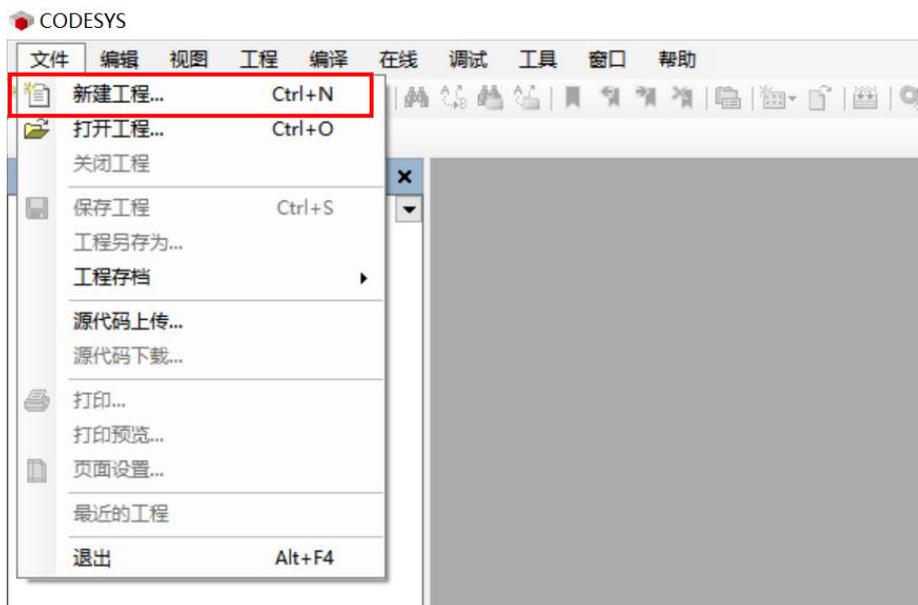
3. 建立一个工程并下载调试

3.1 创建一个新的 PLC 程序

方法 1: 在“起始页”的“基本操作”区域, 点击“新建工程...”;



方法 2: 若没有“起始页”, 可以在菜单栏点击“文件 - 新建工程...”;

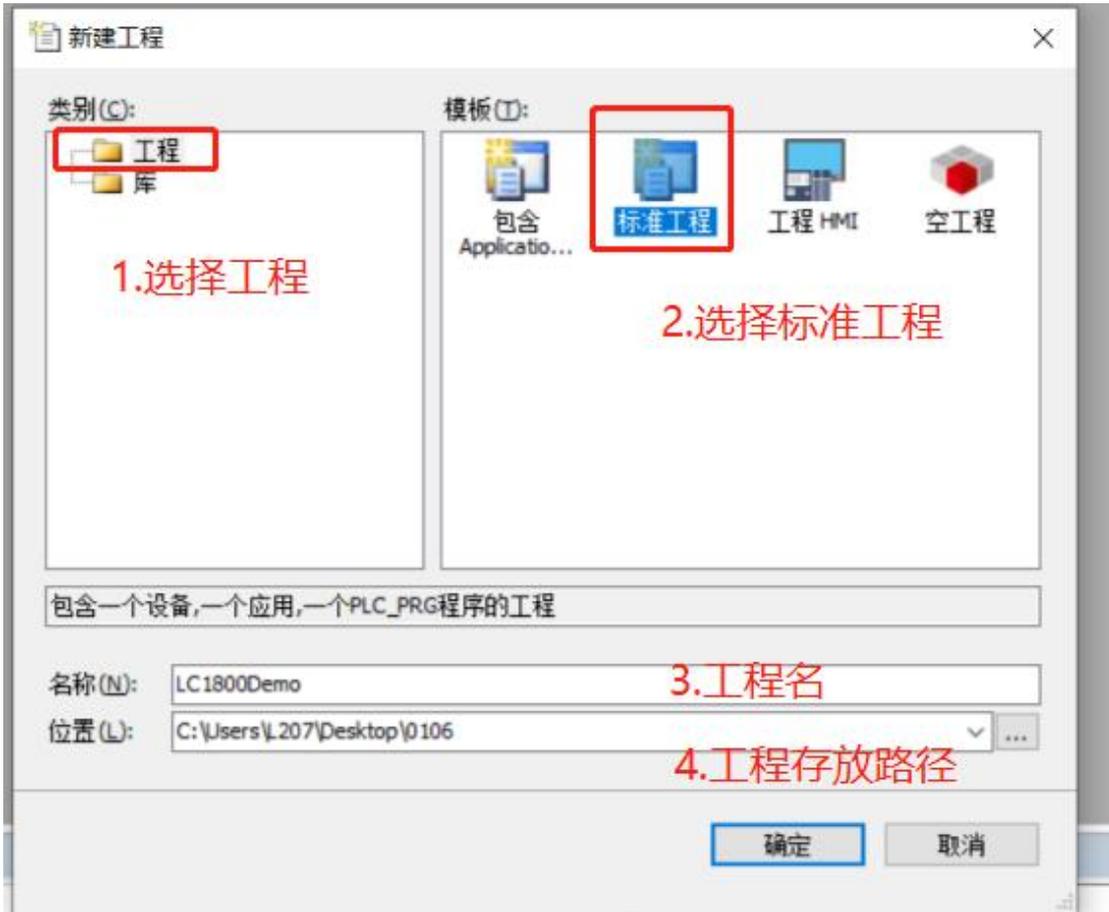


【注 1: 若想显示“起始页”, 可以点击菜单栏“视图 - 起始页”。】

方法 3: 使用快捷键 Ctrl + N;

3.2 类别选择“工程”, 模板选择“标准工程”, 输入用户工程的名称, 选择

存放位置, 点击“确定”;



3.3 选择设备 (Lingchen LC1800) 与编程语言 (后续新建 POU 可以更改编程语言) 后, 点击“确定”;



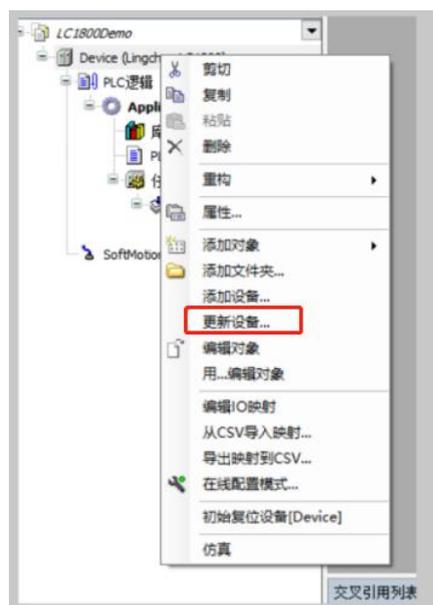
【注 2: 所有第一次使用的设备, 都必须要在“设备存储库”中安装该设备 (具体步骤参见本文档 1. 设备存储库 的使用)。】

3.4 新建工程完成;



3.4.1 Device 设备。

此处是新建工程时所选择的设备, 可以通过“右键 (Device) - 更新设备”来更换用户工程中的设备;

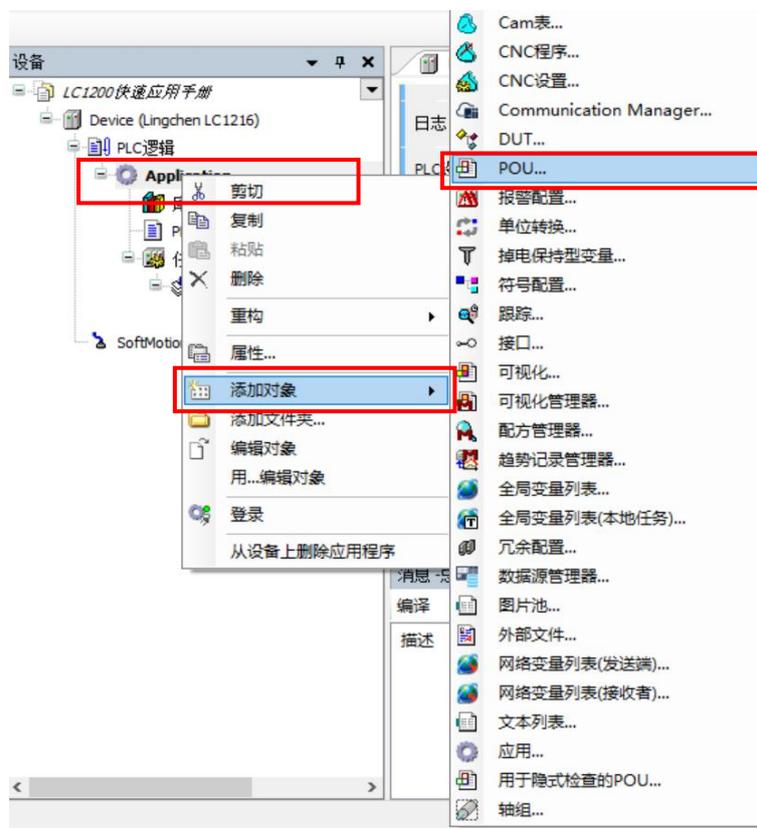


选择要更换的设备后，点击“更新设备”（或者直接双击要更换的设备）。



3.4.2 Application 应用。

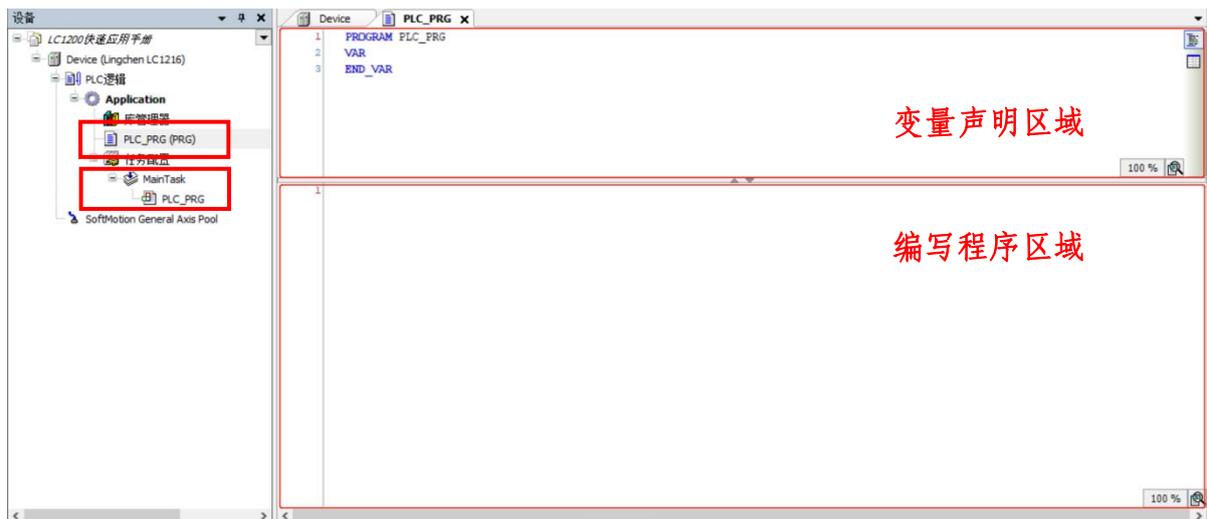
在此处添加用户程序单元 (POU) 等。具体操作方法为：“右键 Application - 添加对象 - POU”，输入 POU 名称，选择“程序”，选择该用户程序单元所要



使用的语言。



3.4.3 PLC_PRG(PRG) 用户程序。



双击进入该界面，在此处声明变量与编写程序。

【注 3: 要将写好的程序①拖入任务配置②下，方可编译调用。】

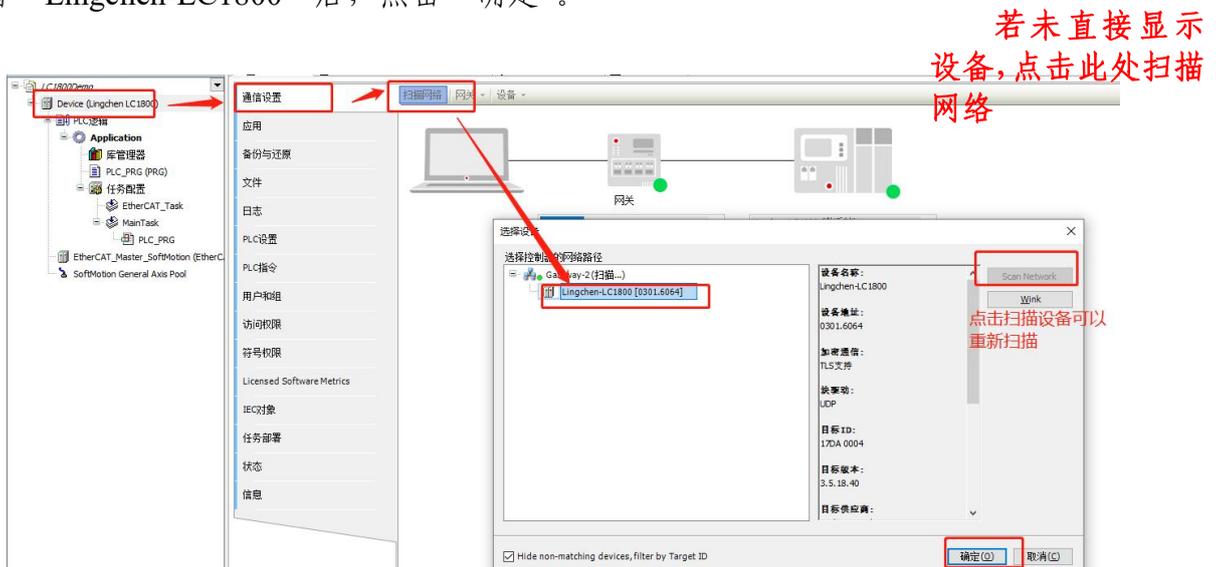
3.4.5 任务配置及程序调用选择。

3.5 编写程序;

3.6 系统配置及参数设定;

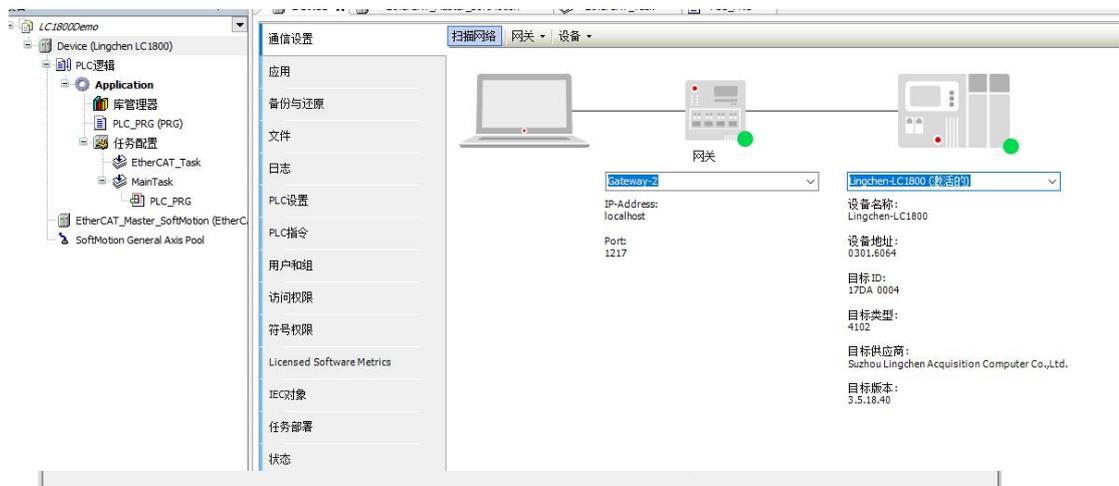
3.6.1 连接设备

双击设备栏“Device”，在“通信设置”子页面内，点击“扫描网络”，选择设备“Lingchen-LC1800”后，点击“确定”。



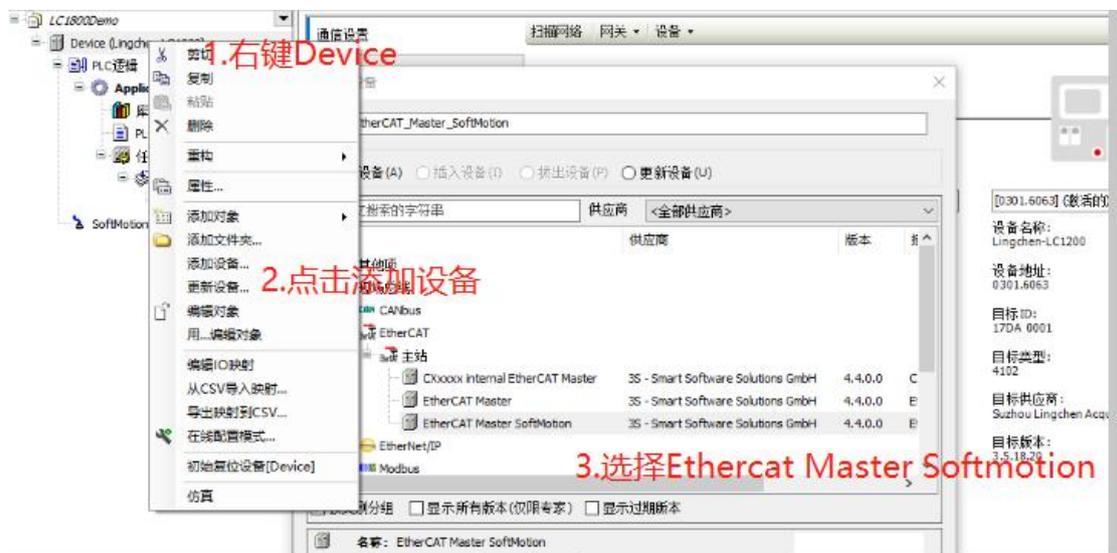
初次连接，会有以下弹窗，点击“是”即可。

连接成功，如 3.12。



3.6.2 添加 EtherCAT_Master_SoftMotion

EtherCAT Master SoftMotion 是带实时运动控制的 EtherCAT 主站模块。具体添加方法为：“右键 Device - 添加设备 - 现场总线 - EtherCAT - 主站 - EtherCAT Master SoftMotion - 添加设备”，添加 EtherCAT 主站。



添加后如 3.14，同时系统会分配一个任务 EtherCAT_Task，可配置 EtherCAT 任务相关参数。



在 EtherCAT_Task 配置界面，可以设定挂在该任务配置下的程序优先级、类型与扫描周期时间间隔等。

3.6.3 添加总线伺服

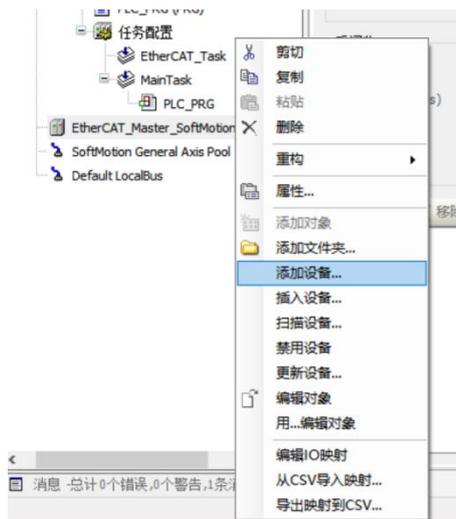
添加主站设备后，在该主站下方添加从站设备，此处添加设备为“LC_SDE_DC03A1（凌臣总线步进_单轴）”。添加前必须先安装设备描述文件

(.XML), 安装流程参考本文档 1.设备存储库的使用。

具体添加方式有两种:

第一种: 手动添加

未连接主站的离线状态下, 通过“右键 EtherCAT Master Softmotion - 添加设备”, 在“添加设备”弹窗中筛选对应供应商及设备型号“LC SDE-DC03A1”, 选中该设备, 点击“添加设备”(或直接双击要添加的设备)。





第二种：扫描添加

(1) “双击 EtherCAT Master SoftMotion” 打开主站设置界面，在“通用”子页面内，设置 EtherCAT 网口源地址，默认选项为“按 MAC 选择网络”时，可以通过点击“浏览”，选择 EtherCAT 网口 MAC 地址 (3.17, 3.18)；也可以更换为“按名称选择网络”选项，在“网络名称”输入框内，输入网口名称“enp3s0” (3.19)。



选择网络适配器

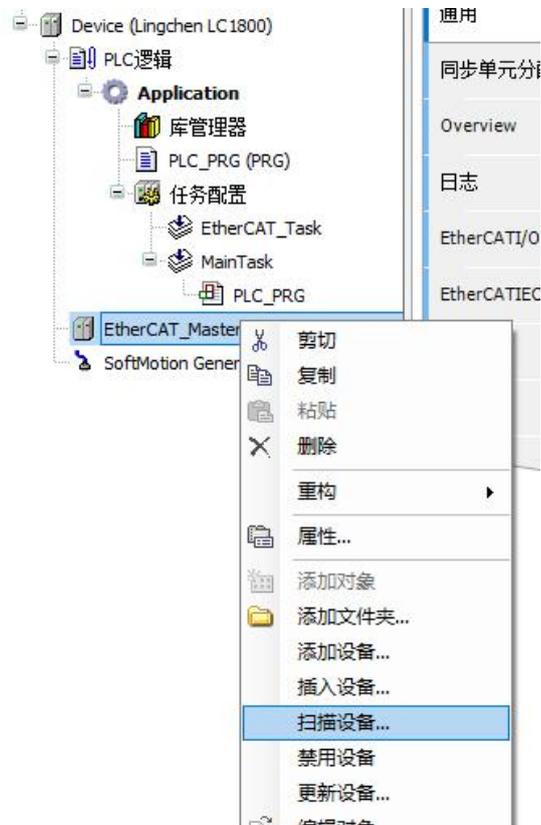




(2) 选定 EtherCAT 网口后，点击“登录”或“在线配置模式”。



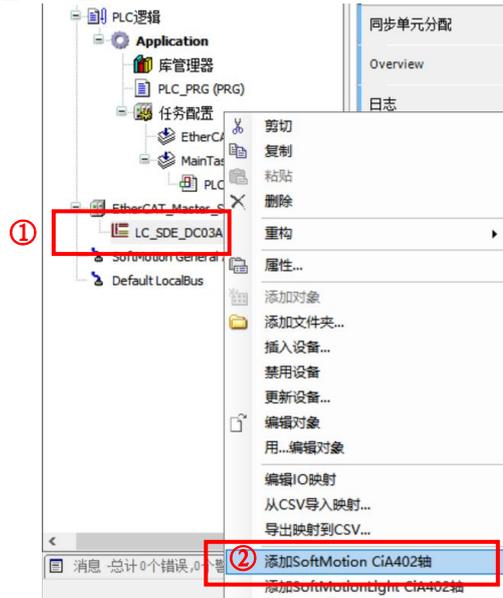
(3) 登录成功后，“右键 EtherCAT_Master_SoftMotion - 扫描设备”。



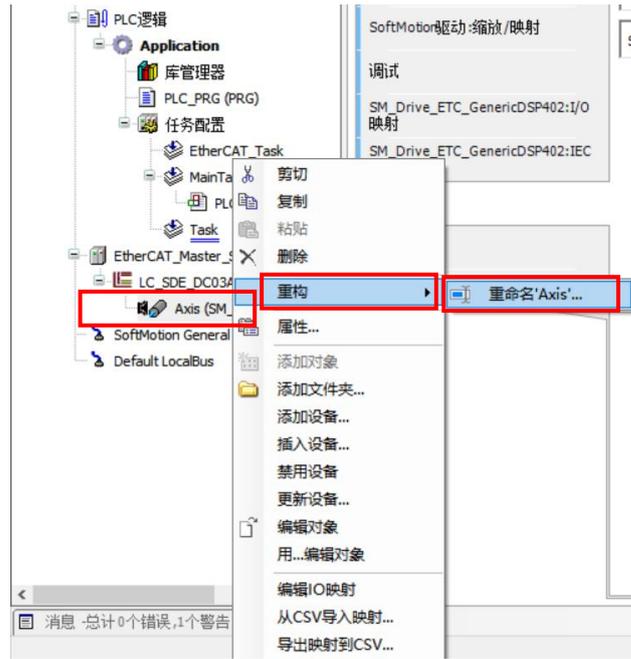
自动扫描到连接的设备后, 点击“复制所有设备到工程”。

【注: 设备的节点地址, 默认是自动分配, 即按照距主机距离从近到远来分配节点地址, 本例不做修改, 按默认设置。】

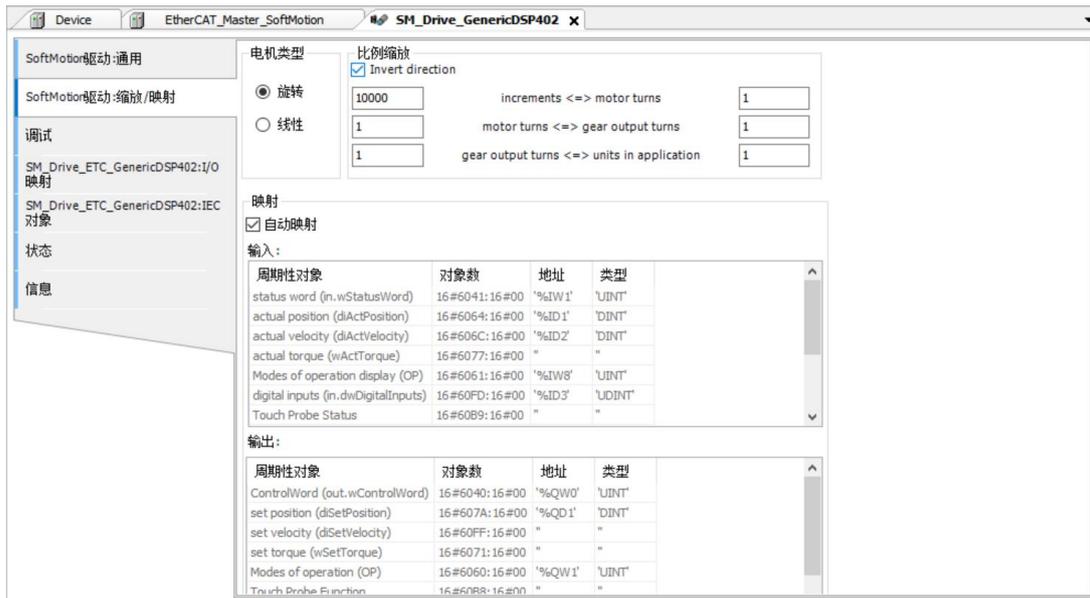
(4) 完成之后, 退出登录, 在新添加的设备处, 添加运动控制轴, 具体操作步骤为“右键 LC_SDE_DC03A1 - 添加 SoftMotion CiA402 轴”。



(5) 添加完成, 为方便编程, 可以对新添加的轴重命名。



(6) 双击新添加的 SoftMotion CiA402 轴，设置控制相关参数，在“SoftMotion 驱动：缩放/映射”子页面内调整电机类型、比例缩放与映射等。



4. 单轴简单用户控制程序编写

此处编写一个单轴的简单程序，使控制器控制伺服电机执行绝对位置指令，做往返运动。

(1) 首先建立对象 (POU) (详情参考 3.4.2)。

(2) 打开编程界面 POU_Abs (详情参照 3.4.3, 3.10)，在变量声明区域添加变量，变量声明代码如下：

```
PROGRAM POU_Abs

VAR

    iStatus:INT; // 执行步骤

    Power:MC_Power; // 使能模块

    ActPos:LREAL; // 实际位置值

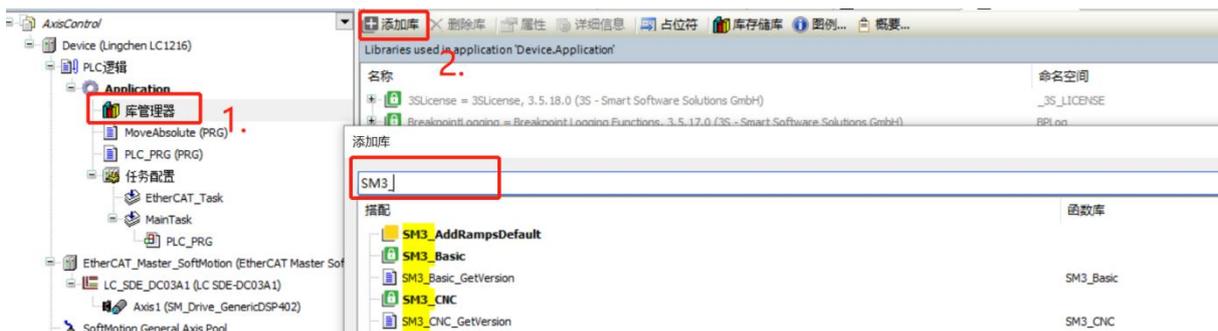
    MoveAbsolute:MC_MoveAbsolute; // 绝对位移模块

    p:REAL:=30; // 位移值

END_VAR
```

【注：声明变量时一般都是默认添

加库文件“SM3_Basic”，如若未添加则需要手动通过双击“库管理器 - 添加库”，找到库“SM3_Basic”然后选择添加，也可以通过此方法添加更多的库。】



(3) 在编程区域添加程序如下。(程序功能：程序执行时，立即使能伺服，等伺服使能成功后，控制电机在位置 P 与起点 O 之间做往返运动。)

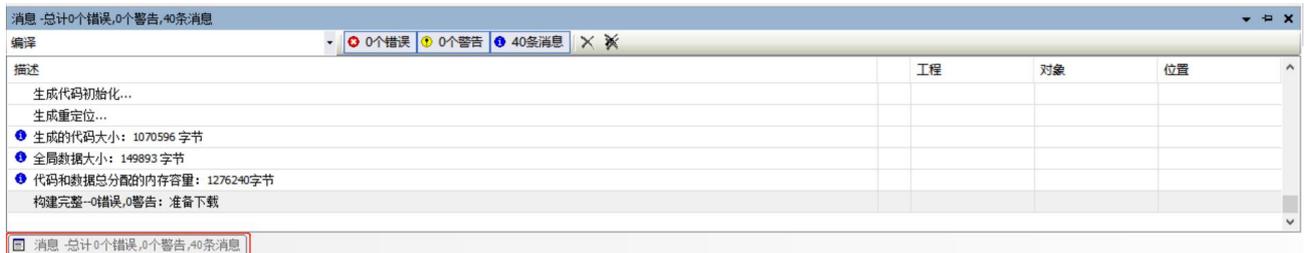
```
CASE iStatus OF

0: // 启动后，轴 Axis 上使能

    Power(Axis:=Axis, Enable:=TRUE, bRegulatorOn:=TRUE,
```

```
bDriveStart:=TRUE );  
  
    IF Power.Status THEN  
  
        iStatus:=iStatus+1;  
  
    END_IF  
  
    1: //走绝对位移, 运行到 P 处  
  
        MoveAbsolute(Axis:=Axis,      Execute:=TRUE,      Position:=      0,  
Velocity:=100 , Acceleration:= 100, Deceleration:=100 );  
  
        IF MoveAbsolute.Done THEN  
  
            MoveAbsolute(Axis:=Axis, Execute:= FALSE); iStatus:=iStatus+1;  
  
        END_IF  
  
    2: //走绝对位移, 运行回到 0 处  
  
        MoveAbsolute(Axis:=Axis,      Execute:=TRUE,      Position:=      p,  
Velocity:=100 , Acceleration:= 100, Deceleration:=100 );  
  
        IF MoveAbsolute.Done THEN  
  
            MoveAbsolute(Axis:=Axis, Execute:= FALSE);  
  
            iStatus:=1;  
  
        END_IF  
  
    END_CASE  
  
ActPos:= Axis.fActPosition;//获取轴实际位置值
```

(4) 程序编写完成后, 点击“编译”, 确认编写没有错误。

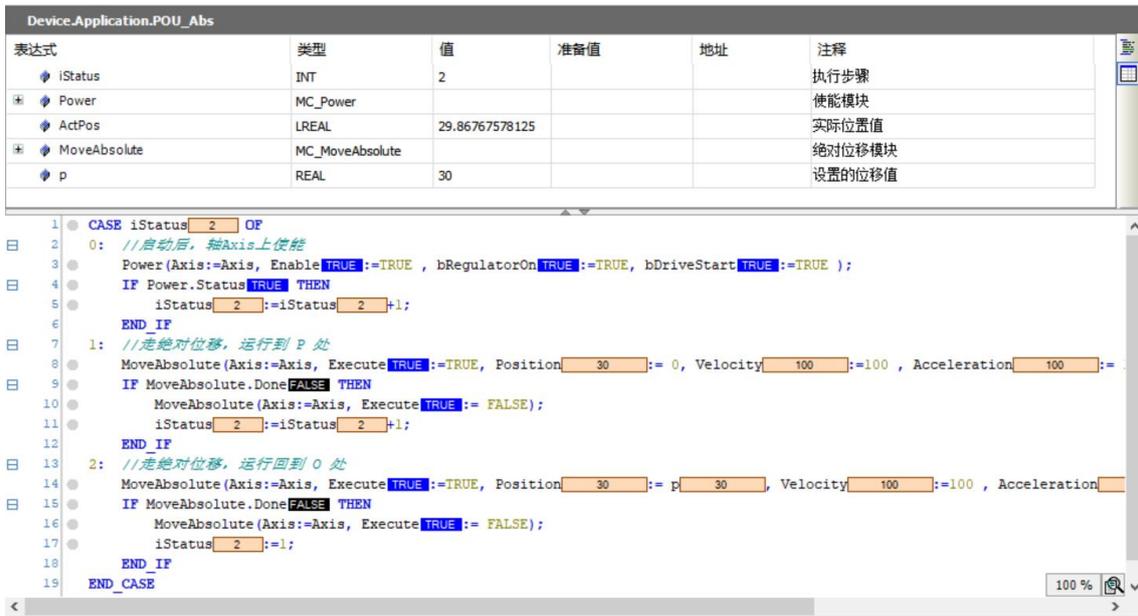


在左下角的“消息”提示框内会显示编译过程描述, 如果有错误, 也会显示在该区域。

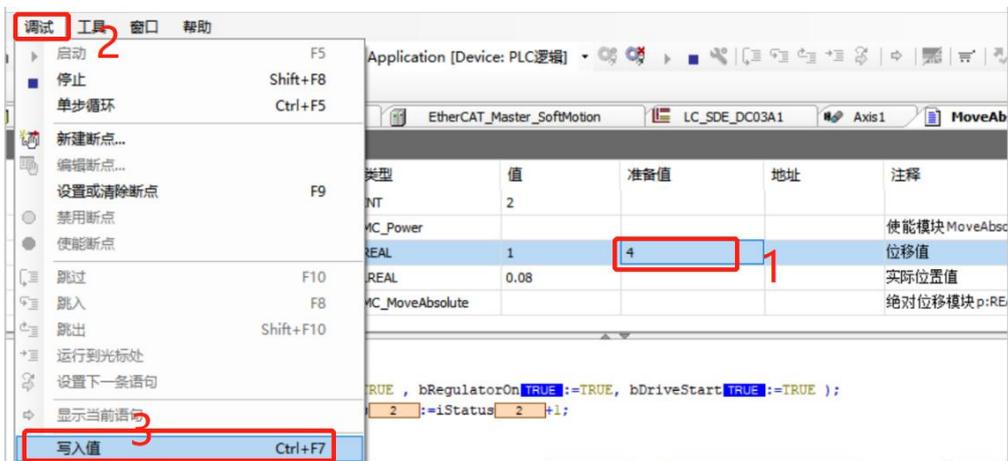
(5) 将编写完成的对象 (POU_Abs) 拖进任务节点下。

(6) 连接设备(参照 3.6.1), 无误后, 点击“登录 - 启动”, 打开“POU_Abs”, 程序运行如下图所示, 程序执行伺服使能后, 可以看到电机在位置 p 与起点位

置 O 之间做往返运动。

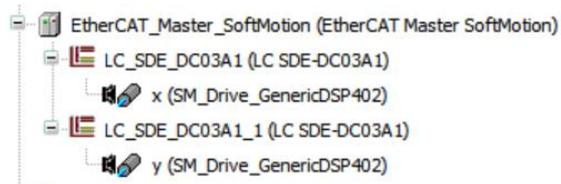


在线修改位置 P 的值: 单击变量“P”的预设值“准备值”, 使其为然后输入数值“10”, 然后选择“调试 写入值”或者快捷键 “Ctrl+F7” 将值写入到“值”中, 即可在线修改变量“P”的值。

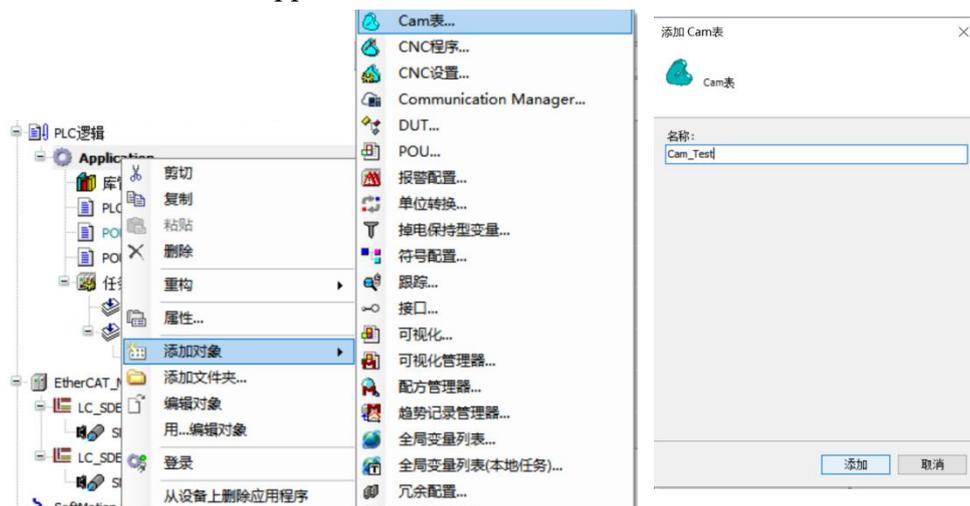


5. 基于 EtherCAT 通信的电子凸轮程序例程

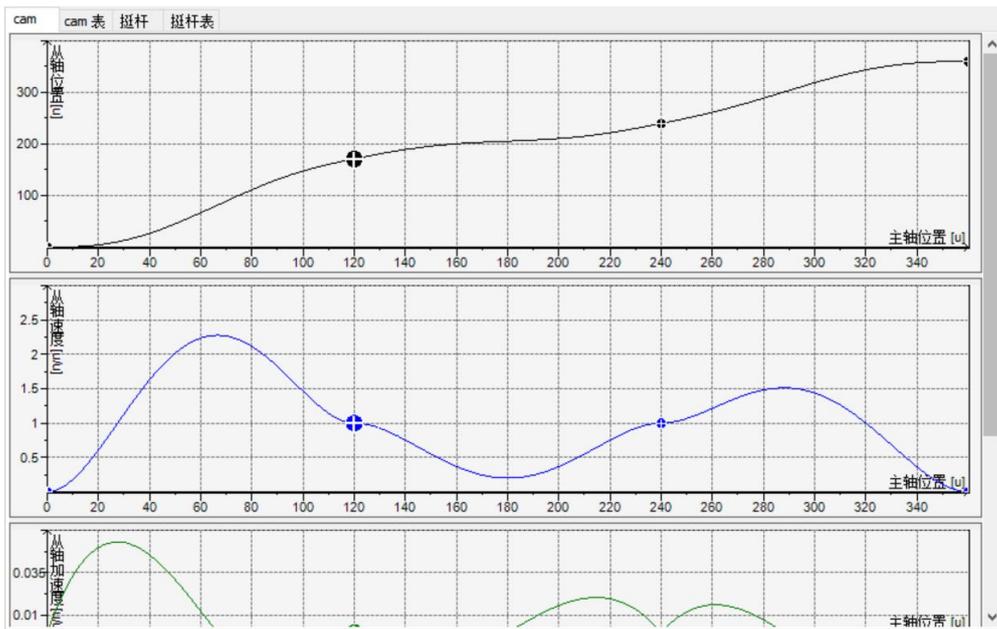
参照上面的操作步骤，新建工程，添加 EtherCAT Master SoftMotion 主站，更改 EtherCAT 网口后，“登录—右键 EtherCAT Master SoftMotion—扫描设备”，扫描到设备如下（本例程以两套“LC_SDE_DC03A1（凌臣总线步进_单轴）”为例），“复制所有设备到工程”，添加完毕并为每个步进添加与一个 402 轴，为方便编程示范分别重命名为 x, y，如下图（3.25）所示（详情请参考 3.6.4 单轴简单用户控制程序编写）。



在 Application 下添加一个对象（POU_Cam），再添加一个对象（Cam 表），具体操作步骤为“右键 Application - 添加对象 - Cam 表...”，自定义 Cam 表的名



称, 点击“添加”。添加完成后, Cam 表界面如图所示。



可以对相关项进行自定义设置 (本文档仅做示范作用, 因此使用默认设置)。

打开编程界面 POU_Cam, 在变量声明区域添加变量, 变量声明代码如下:

```
PROGRAM POU_Cam
```

```
VAR
```

```

xstart      :BOOL;//启动开关
power_x     :MC_Power;//轴 x 使能模块
power_y     :MC_Power;//轴 y 使能模块
mov         :MC_MoveRelative;//相对运动模块
SEL1        :MC_CamTableSelect;//Cam 表选择
camin       :MC_CamIn;
camout      :MC_CamOut;
t           :TON;
step        :INT;//执行步骤

```

```
END_VAR
```

在编程声明区域添加程序, 程序如下: (程序功能: 程序执行时, 通过 xstart 开启运动, 等伺服使能成功后, 控制轴 x 与轴 y 进行电子凸轮运动。)

```

power_x( Axis:= x , Enable:= xstart , bRegulatorOn:= 1 , bDriveStart:= 1 );
power_y( Axis:= y , Enable:= xstart , bRegulatorOn:= 1 , bDriveStart:= 1 );

```

```
mov( Axis:= x , Execute:= , Distance:=10 , Velocity:= 10, Acceleration:= 100,
Deceleration:=100 , Jerk:= 1000 );
```

```
SEL1( Master:= x , Slave:= y , CamTable:=Cam_Test , Execute:= , Periodic:= 1,
MasterAbsolute:=0 , SlaveAbsolute:= 0 );
```

```
camin( Master:= x, Slave:= y, Execute:= , MasterScaling:=1 , SlaveScaling:= 1,
StartMode:= absolute,
```

```
CamTableID:=SEL1.CamTableID , VelocityDiff:= , Acceleration:=100 ,
Deceleration:=100 , Jerk:=100 );
```

```
camout( Slave:= y );
```

```
t(IN:= , PT:= T#1S, Q=> , ET=> );// 设定延时时间为 1s
```

```
CASE step OF
```

```
0:// 等待轴使能
```

```
IF power_x.Status THEN
```

```
step:=100;
```

```
END_IF
```

```
100:// 选择 Cam 表, 开启 Cam 表: Cam_test
```

```
SEL1.Execute:=1;
```

```
step:=200;
```

```
200:
```

```
camin.Execute:=1;// 进入
```

```
camout.Execute:=0;
```

```
step:=500;
```

```
500:
```

```
mov.Execute:=1;// 开启相对运动
```

```
camin.Execute:=0;
```

```
step:=550;
```

```
550:// 运动到位时跳出
```

```
IF mov.Done THEN
```

```
camout.Execute:=1;
```

```
step:=600;
```

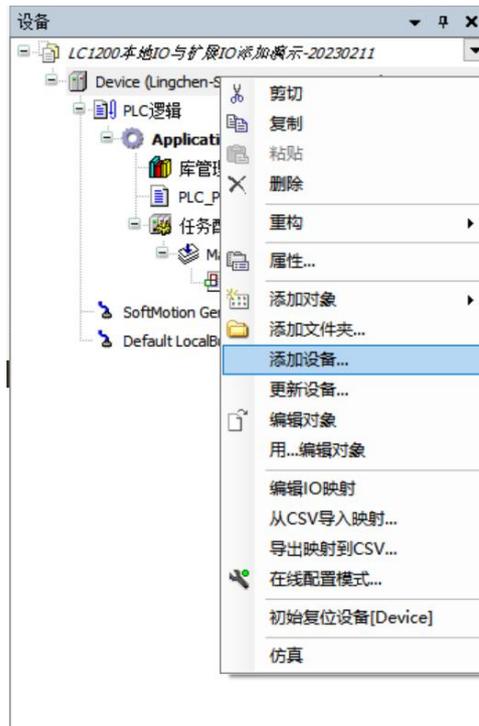
```
END_IF  
  
600:  
  
    mov.Execute:=0;  
  
    t.IN:=1;  
  
    IF t.Q THEN  
  
        t.IN:=0;  
  
        step:=200;  
  
    END_IF  
  
END_CASE
```

程序编写结束后,点击“编译”,确认无误,将POU_Cam放到“EtherCAT_Task”下。点击“登录”,登录成功后,点击“启动”,等驱动和轴前的图标变为绿色后,将“xstart”的值改为“true”,程序运行如下图所示。

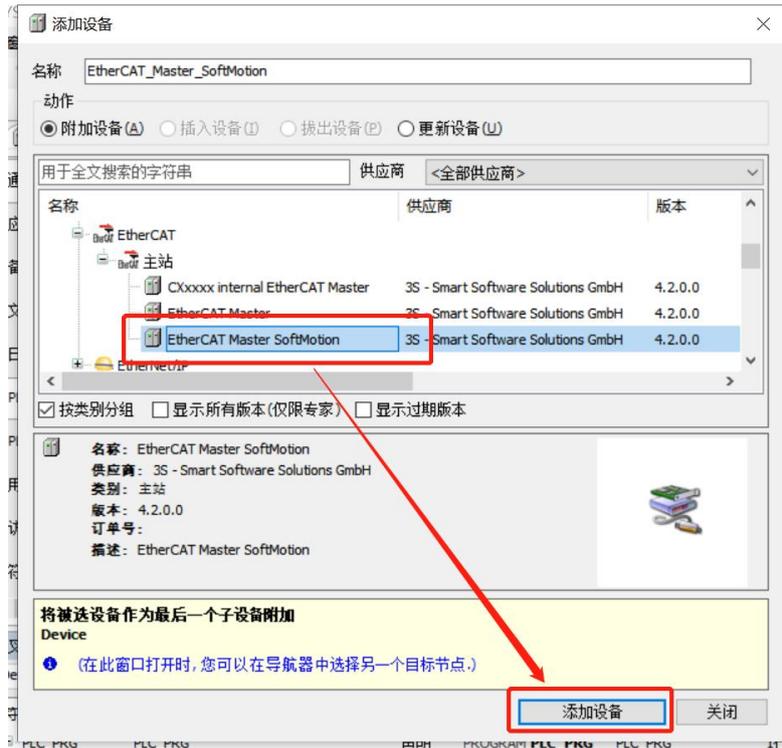


6. 添加拓展 IO (以 LC1100 和 1488、2488 模块为例)

1. 同样创建标准工程后, 在工具-设备存储库中导入设备文件(一次导入后, 下次不用重复操作);
2. 在设备栏, 右键“Device”, 选择“添加设备”;



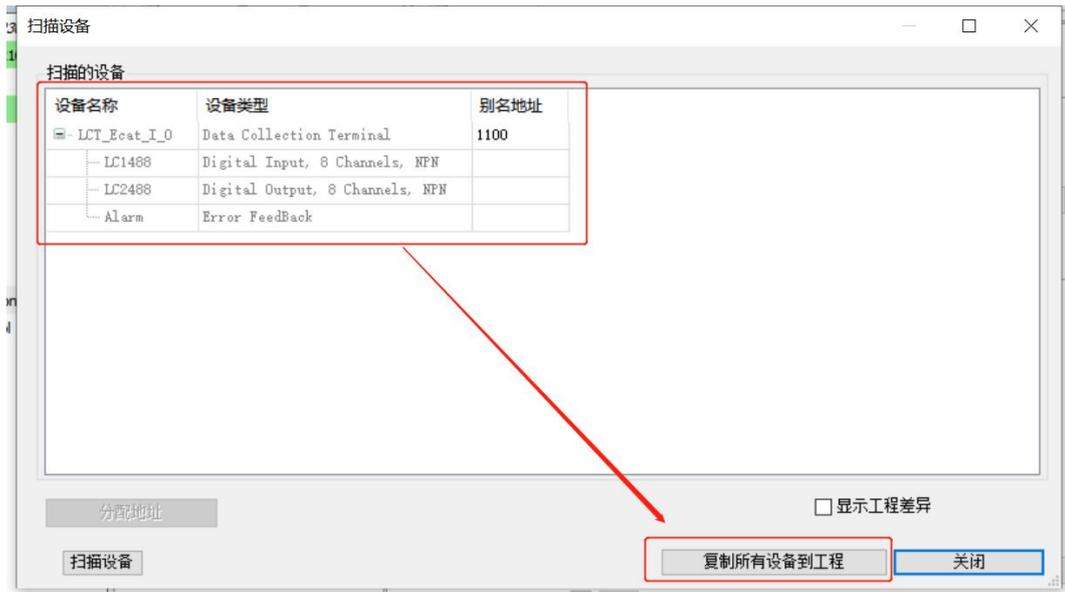
3. 添加主站;



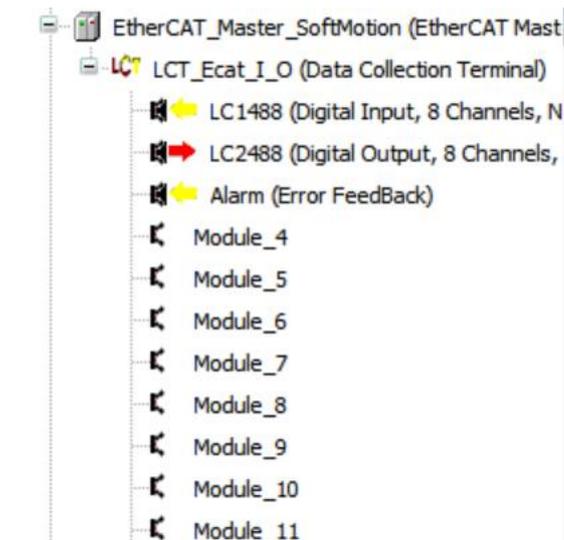
4. 选中刚刚添加的设备，在“通用”界面选择“源地址”，选择准备连接的网口源地址；



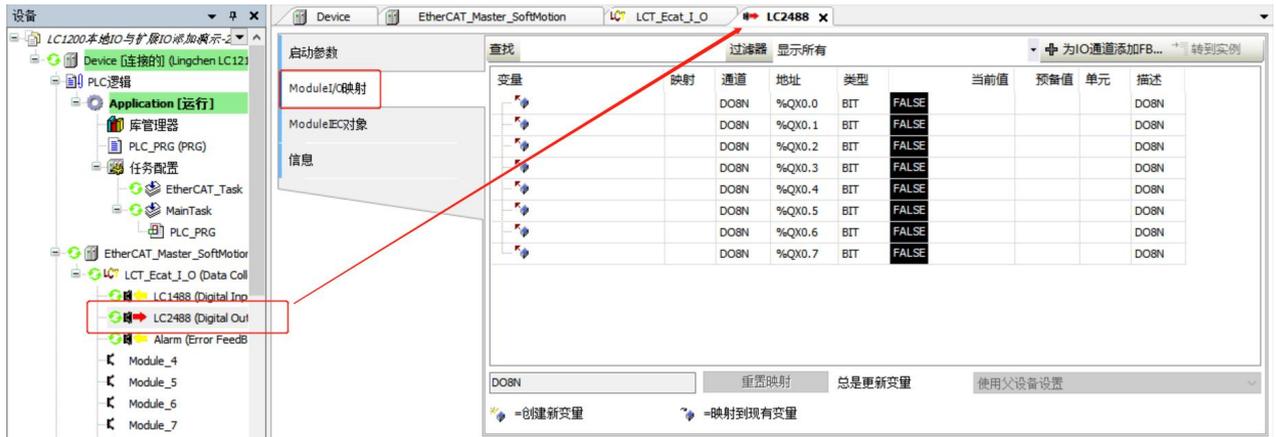
5. 选好后，“登录”或“在线”，右键主站（EtherCAT_Master_SoftMotion），“扫描设备”，在弹出的对话框内确认设备是否全部正常后，“复制所有设备到工程”；



复制完成后，在设备栏展示如下：



6. “登录 - 启动”，在“设备栏”双击 LC2488（输出模块）打开界面，点击“ModuleI/O 映射”子界面，可以在此为输出赋值，同理，在 LC1488（输入模块）界面的“ModuleI/O 映射”子界面，监测输入当前值。



The screenshot shows the configuration interface for the LC2488 module. On the left, a tree view shows the project structure, with 'LC2488 (Digital Out)' selected. The main window displays the 'Module I/O Mapping' table. A red box highlights 'Module I/O映射' in the left sidebar, and another red box highlights 'LC2488 (Digital Out)' in the tree view. A red arrow points from the 'LC2488 (Digital Out)' box to the 'Module I/O Mapping' table.

变量	映射	通道	地址	类型	当前值	预备值	单元	描述
		DO8N	%Q0.0	BIT	FALSE			DO8N
		DO8N	%Q0.1	BIT	FALSE			DO8N
		DO8N	%Q0.2	BIT	FALSE			DO8N
		DO8N	%Q0.3	BIT	FALSE			DO8N
		DO8N	%Q0.4	BIT	FALSE			DO8N
		DO8N	%Q0.5	BIT	FALSE			DO8N
		DO8N	%Q0.6	BIT	FALSE			DO8N
		DO8N	%Q0.7	BIT	FALSE			DO8N

At the bottom of the table, there are controls for 'DO8N', '重置映射' (Reset Mapping), '总是更新变量' (Always Update Variable), and '使用父设备设置' (Use Parent Device Settings). Below these are buttons for '=创建新变量' (Create New Variable) and '=映射到现有变量' (Map to Existing Variable).

7.Modbus

7.1 Modbus TCP Master

配合使用软件: Modbus Slave

具体操作步骤:

关闭电脑防火墙。

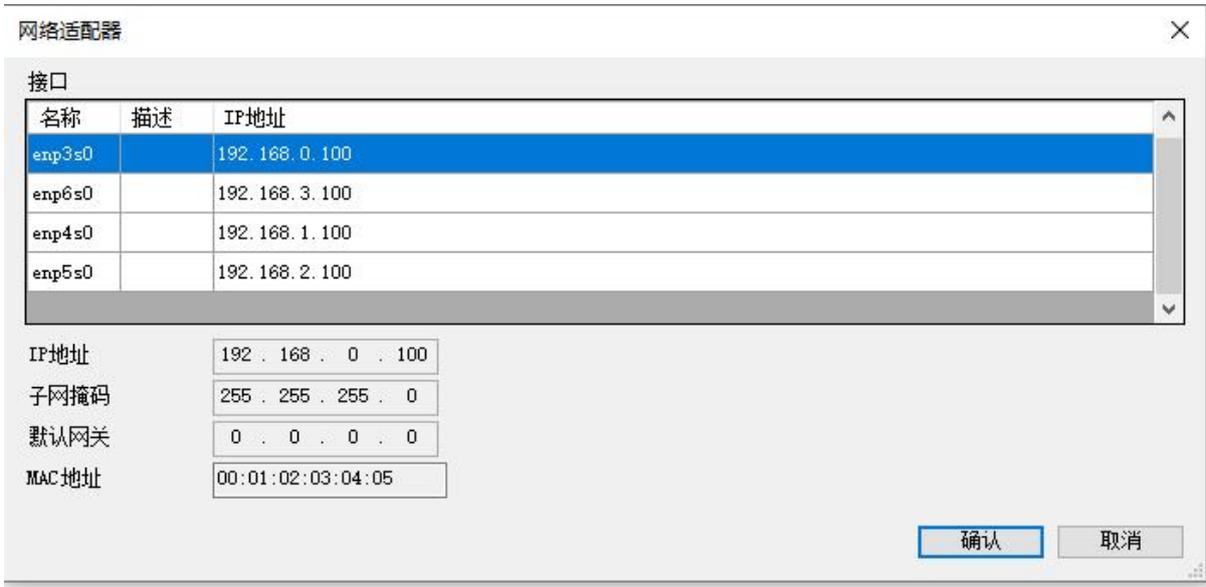
- (1) 新建标准工程文件;
- (2) 右键 “Device” — 添加设备;
- (3) 点开 “以太网适配器” 子选项, 选中 “Ethernet”, 点击 “添加设备”;



- (4) 添加完成后, 双击打开 Ethernet 设备界面, 在 “通用” 子界面, 点击 “Browse...” 选择网络接口 (选择网络接口前需要先连接设备 Device);



(5) 选中 “enp3s0” (也可以自定义), 点击确认;



(6) “右键 Ethernet—添加设备—按照图中所示位置, 找到 Modbus TCP Master 选中—添加设备—关闭”;



(7) “右键 Modbus TCP Master—添加设备—按照图中所示位置, 找到 Modbus TCP slave 选中—添加设备—关闭”;

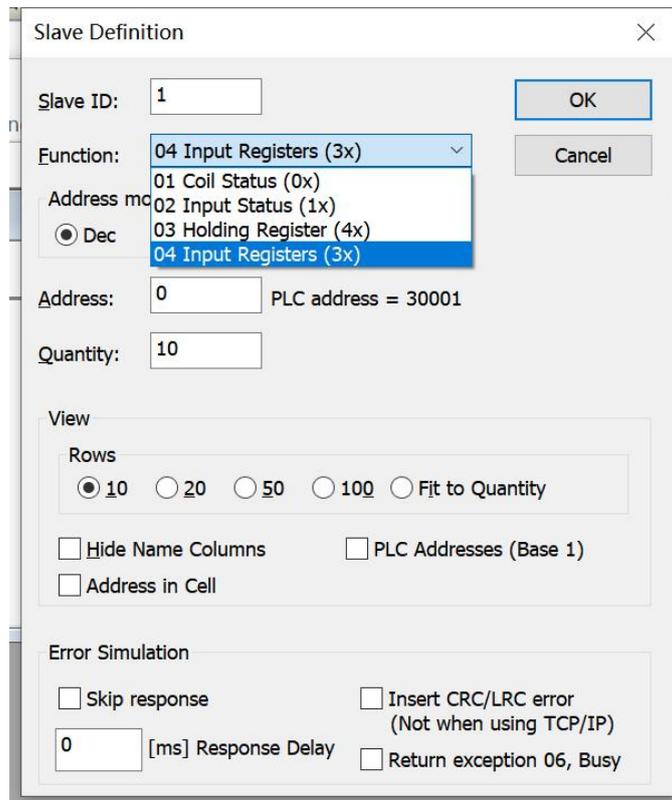
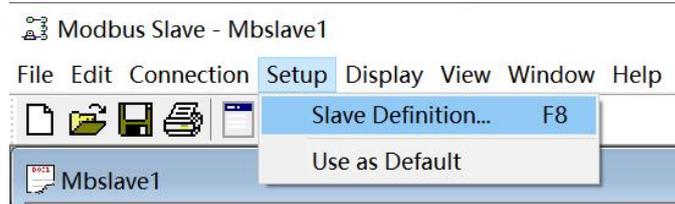


(8) Master 设备界面不需要做修改, 双击打开 slave 设备界面, 设置要连接的从站 ip (也即测试笔记本使用的 ip, 这里设定的是 192.168.0.2, 因此如下图填入);

Modbus TCP

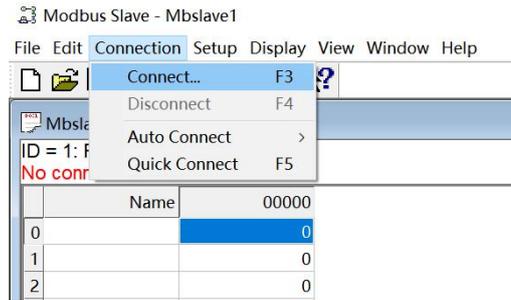
从站IP地址	192 . 168 . 0 . 2
响应超时(ms)	1000
端口	502

(9) 设置完成后, 打开 Modbus Slave 程序, 点击“Setup - Slave Definition...”;



功能码选择 04, 可以改变长度 (Quantity), 点击 “OK”。

(10) 点击 “Connection - Connect...”

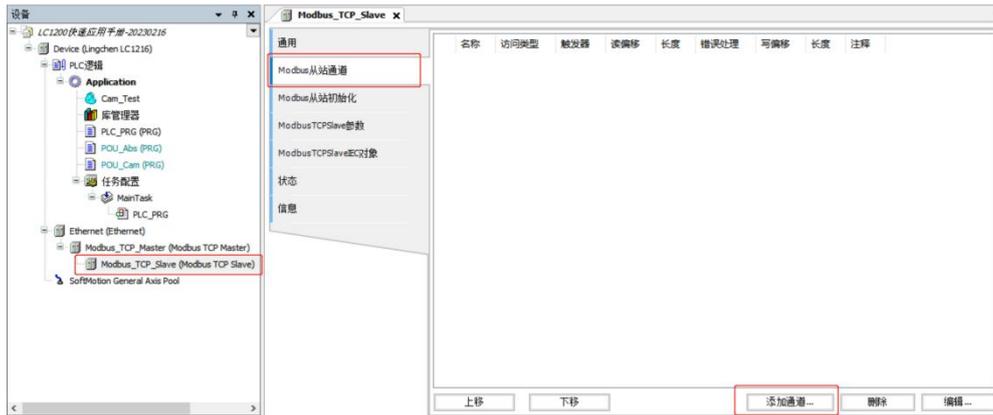


在弹出的界面内, 先选择“connection”为“Modbus TCP/IP”, 然后去掉“Any Address”前的勾选, 将测试电脑的 ip 地址 (192.168.0.2) 输入至“IP Address”

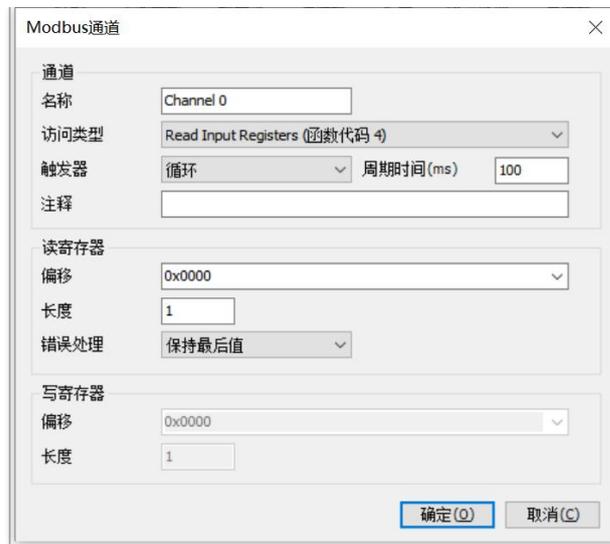


框内, 点击“OK”。

(11) 此时已经连接上, 返回 codesys 界面, 在 Modbus_TCP_Slave 界面的“Modbus 从站通道”子界面内, 点击“添加通道”。

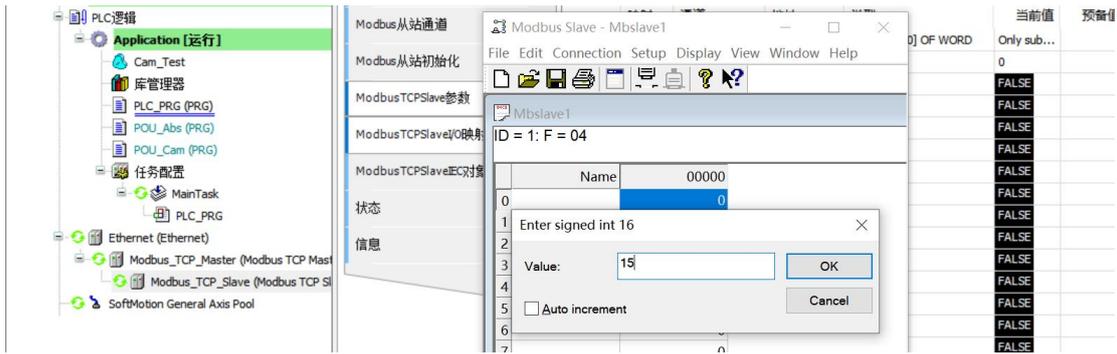


添加 Read Input Register 通道（访问类型选“Read Input Register 通道（函数代码 4）”），可对从站寄存器进行读操作；

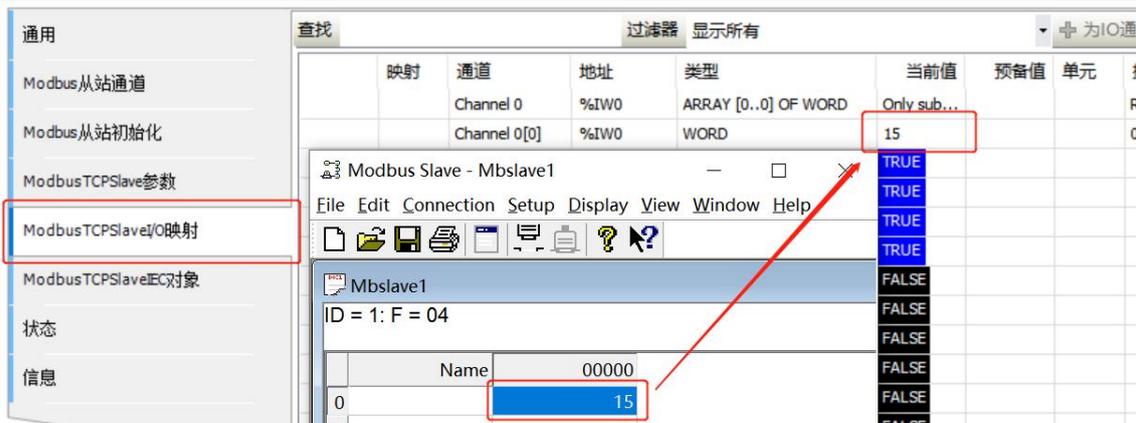


3	Channel 3	Read Input Registers (函数代码 04)	循环, t#100ms	16#0000	1	保持最后值
---	-----------	--------------------------------	-------------	---------	---	-------

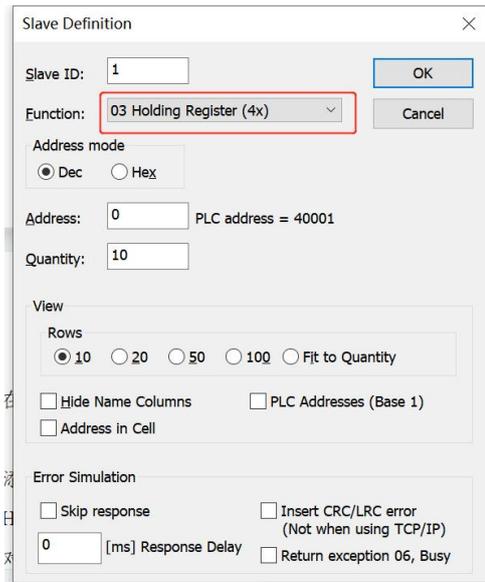
按照上图所示，读寄存器的地址偏移为 0x0000，因此在 Modbus Slave 中双击地址为 0 的地方，在弹出的窗口内，输入数值（如 15），点击“OK”写入。



在 Modbus_Slave_TCP 的“ModbusTCPSlaveI/O 映射”子界面内可以看到数值已经同步过去。



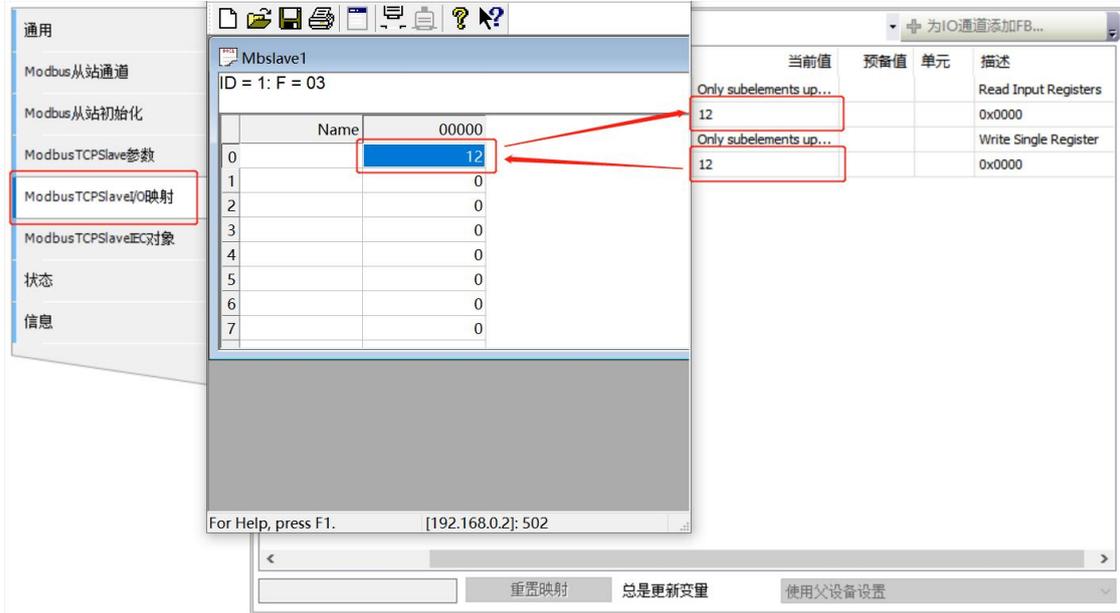
(12) 在 Modbus Slave 程序中, 选功能码 03 (03 Holding Register (4x))



在添加 Read Holding Register 和 Write Single Register 通道 (访问类型分别选

“Read Holding Register (函数代码 3)”和“Write Single Register (函数代码 6)”, 可对从站寄存器进行读写操作。

1	Channel 1	Write Single Register (函数代码 06)	上升沿				16#0000	1
2	Channel 2	Read Holding Registers (函数代码 03)	上升沿	16#0000	1	保持最后值		



如上图所示, 读写地址都为 0x0000, 因此在“ModbusTCPSlaveI/O 映射”子页面内, 写入“12”时, ModbusSlave 程序内和 I/O 映射界面的读输入寄存器位置都可以看到数值“12”。

7.2 Modbus TCP Slave

使用软件: Modbus Poll

具体操作步骤:

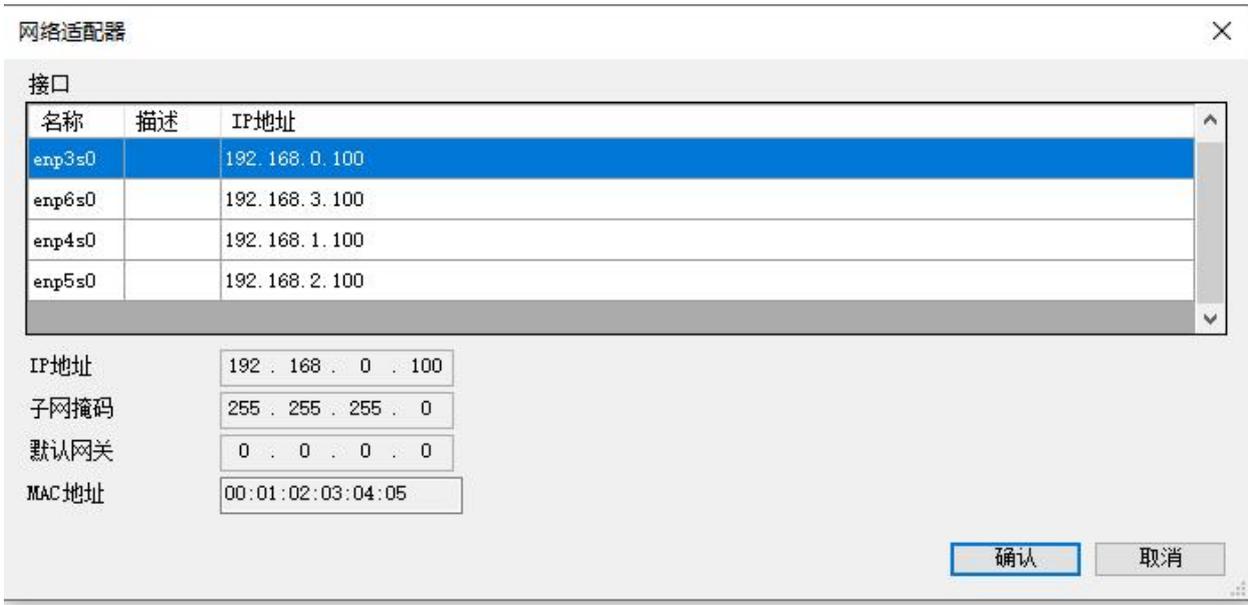
- (1) 新建标准工程文件;
- (2) 右键“Device”—添加设备;
- (3) 点开“以太网适配器”子选项, 选中“Ethernet”, 点击“添加设备”;



- (4) 添加完成后, 双击打开 Ethernet 设备界面, 在“通用”子界面, 点击“Browse...”选择网络接口 (选择网络接口前需要先连接设备 Device);



- (5) 选中“eth0” (也可以自定义), 点击确认;



(6) “右键 Ethernet—添加设备—按照图中所示位置，找到 Modbus TCP Slave 选中—添加设备—关闭”;



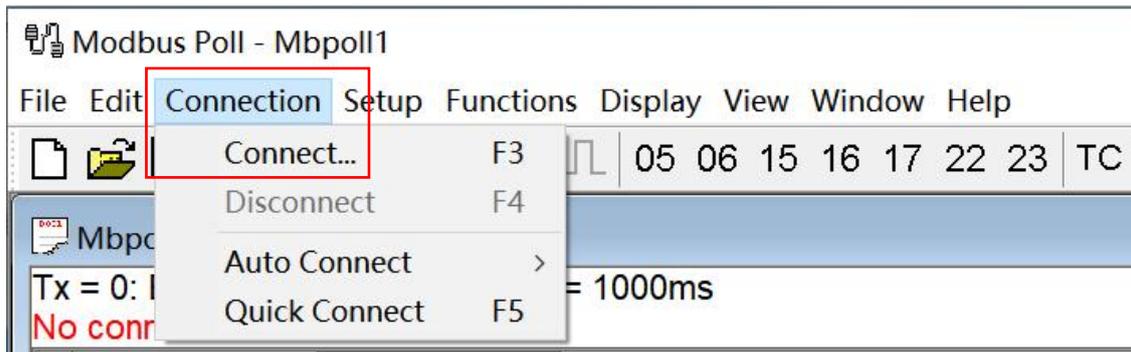
(7) 双击打开刚添加的“Slave”设备 - “通用”界面，可以设定保持寄存器和输入寄存器的长度及起始地址，若需要用到线圈和离散输入，请勾选“离散”

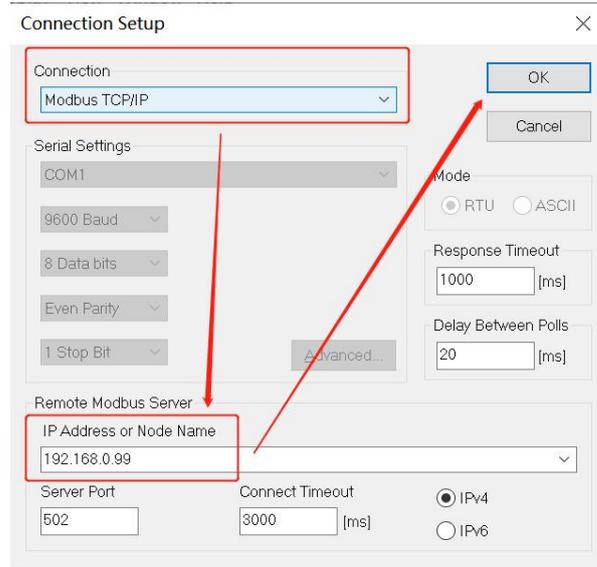
位区域”，并对其长度进行设置；



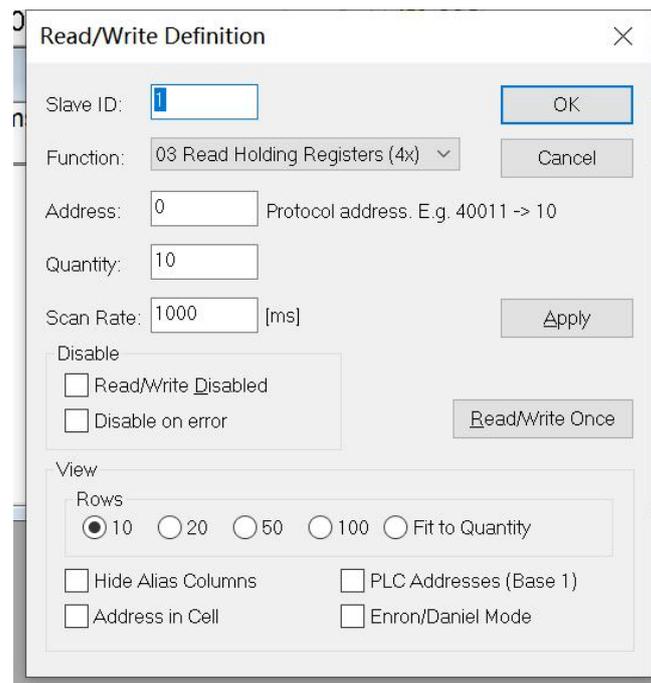
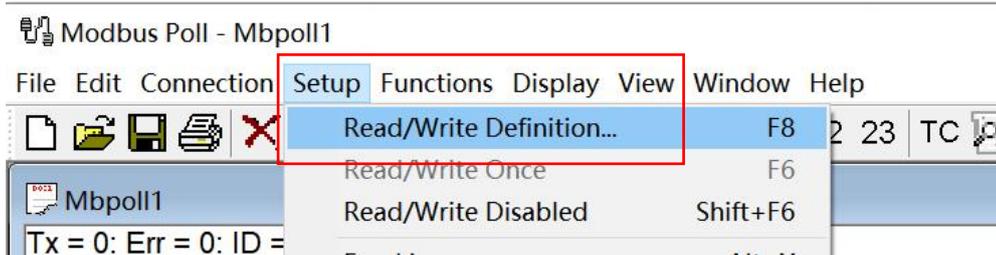
【注: 功能码 03, 06, 05 对应的是保持寄存器, 写线圈偏移 0 对应的是 bit8。】

(8) 打开 Modbus Poll 程序, 点击 “Connection - Connect...”, 输入 PLC 的 ip 地址;

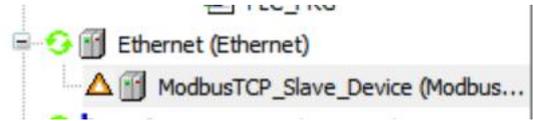




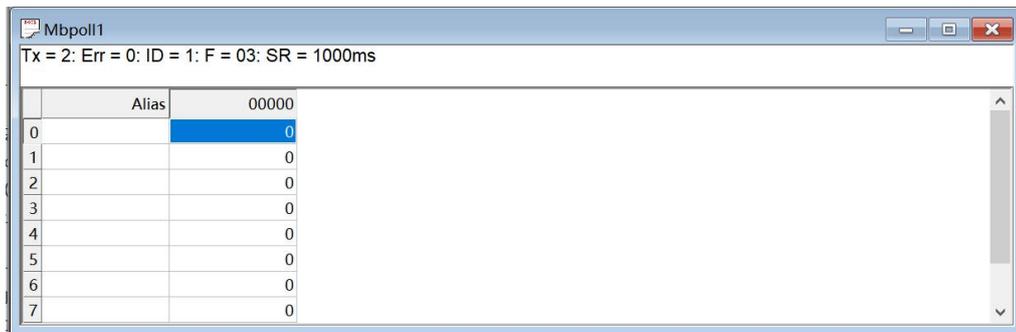
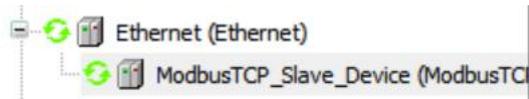
(9) 在“Setup - Read/Write Definition...”中，可以设置读写功能码（本例针对保持寄存器或输入寄存器）、读写地址以及长度等；



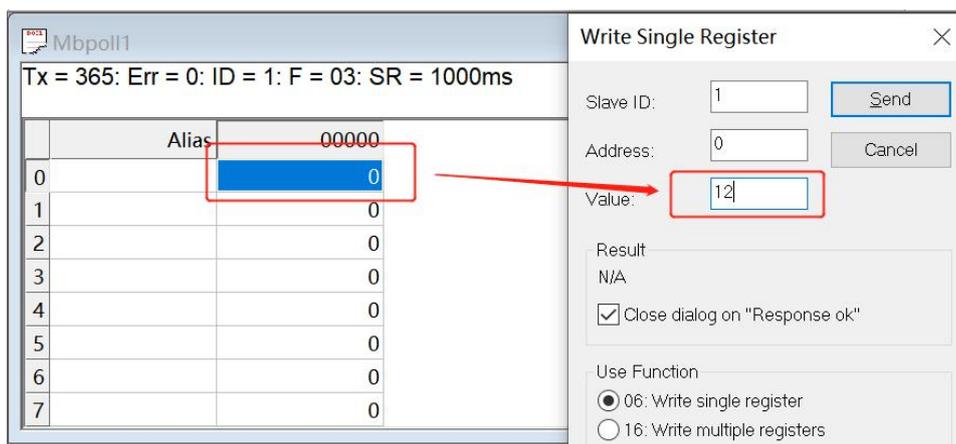
(10) 对保持寄存器操作: 在 Codesys 软件中, 登录设备, 启动, 会发现如下状态, 这是因为 Modbus Poll 软件未启动导致, 在上图中的 Function (功能) 处选择功能码 03, 确定之后, 点击 “Connection - Connect...”;



连接成功, 如下图所示,



此时可以在 Modbus Poll 中可写对应寄存器值,



输入值 12, 点击 “Send” 发送成功后, 可以在相应位置看到结果, 点开 “Modbus TCP Slave Device I/O” 界面, 发现保持寄存器 [0] 已经有数据收到。

变量	映射	通道	地址	类型	当前值
		保持寄存器	%IW0	ARRAY [0..9] OF WORD	Only sub...
		保持寄存器[0]	%IW0	WORD	12
		保持寄存器[1]	%IW1	WORD	0
		保持寄存器[2]	%IW2	WORD	0
		保持寄存器[3]	%IW3	WORD	0
		保持寄存器[4]	%IW4	WORD	0
		保持寄存器[5]	%IW5	WORD	0
		保持寄存器[6]	%IW6	WORD	0
		保持寄存器[7]	%IW7	WORD	0
		保持寄存器[8]	%IW8	WORD	0
		保持寄存器[9]	%IW9	WORD	0
		输入寄存器	%QW0	ARRAY [0..9] OF WORD	Only sub...

(11) 对输入寄存器操作: 在 Codesys 软件中, 登录设备, 启动, 在下图中的 Function (功能) 处选择功能码 04, 地址改为 10, 确定之后, 点击 “Connection - Connect...”;

同样连接成功后, 在 PLC 的 “Modbus TCP Slave DeviceI/O” 界面的输入寄存器写入值, 在 Modbus Poll 中可读对应寄存器值。

变量	映射	通道	地址	类型	当前值	预备值	单
		保持寄存器[9]	%IW9	WORD	0		
		输入寄存器	%QW0	ARRAY [0..9] OF WORD	Only sub...		
		输入寄存器[0]	%QW0	WORD	0		
		输入寄存器[1]	%QW1	WORD	0		
		输入寄存器[2]	%QW2	WORD	5		
		输入寄存器[3]	%QW3	WORD	0		
		输入寄存器[4]	%QW4	WORD	4		
		输入寄存器[5]	%QW5	WORD	0		

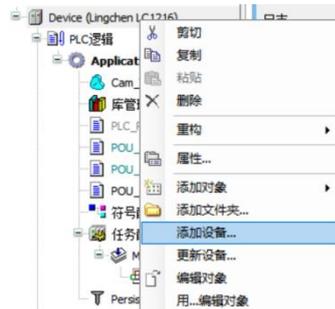
Alias	00010
	0
	0
	5
	0
	4
	0

7.3 Modbus RTU

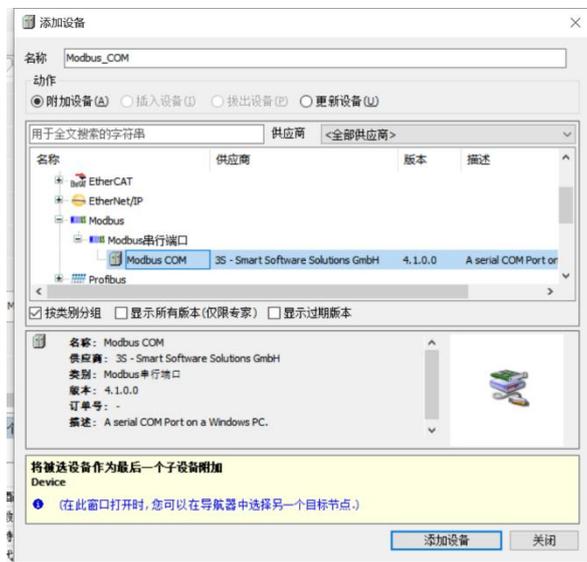
注意: 请先关闭电脑防火墙。

LC1200 只具有 Modbus Serial Master 功能, 因此将其设为主站, 操作步骤如下:

1. 右键 “Device” - “添加设备”。



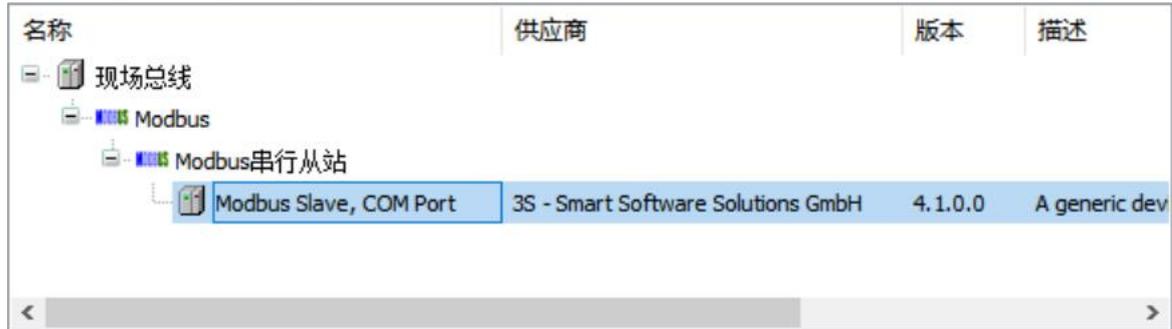
2. 依次展开 “现场总线 - Modbus - Modbus 串行端口 - Modbus COM”, 选中后, 点击 “添加设备”。



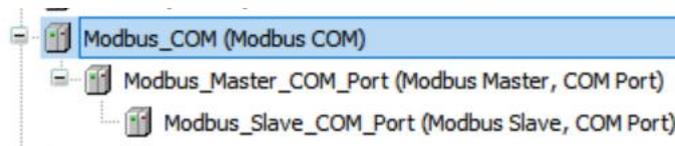
3. 添加完成后, 可以不用关闭 “添加设备” 窗口, 点击设备栏刚刚添加的 “Modbus_COM”, “添加设备” 的窗口会自行跳转, 点击串行主站, 选中添加。



添加完串行主站，同样不用关闭“添加设备”窗口，点击设备栏刚刚添加的 Modbus_Master_Com_Port，“添加设备”窗口跳转，点击串行设备，选中添加。



4. 添加完成后，设备栏如下图所示，双击“Modbus_COM”打开界面，选



择“通用”子界面，修改 COM 端口及其他串口配置参数。

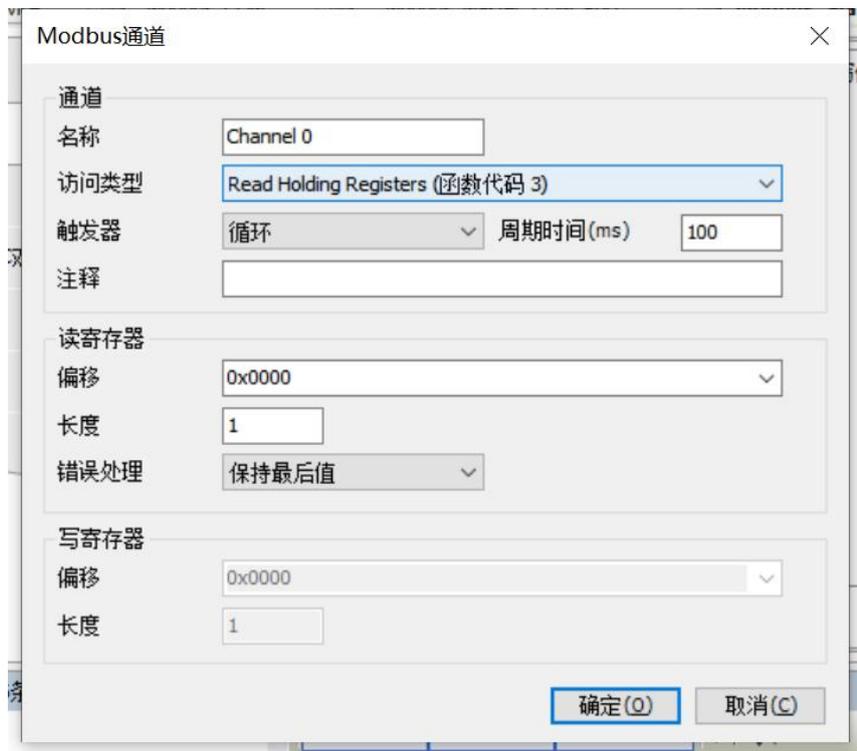


注意，端口号要和硬件连接的端口对应，如选 232，则端口号为 1，选 485，端口号为 2。

5. 双击“Modbus_Slave_Com_Port”打开界面，在“通用”界面可以选择从站地址（1~247），在“Modbus 从站通道”子界面点击“添加通道”。



6. 在弹出的界面选择访问类型及地址偏移。

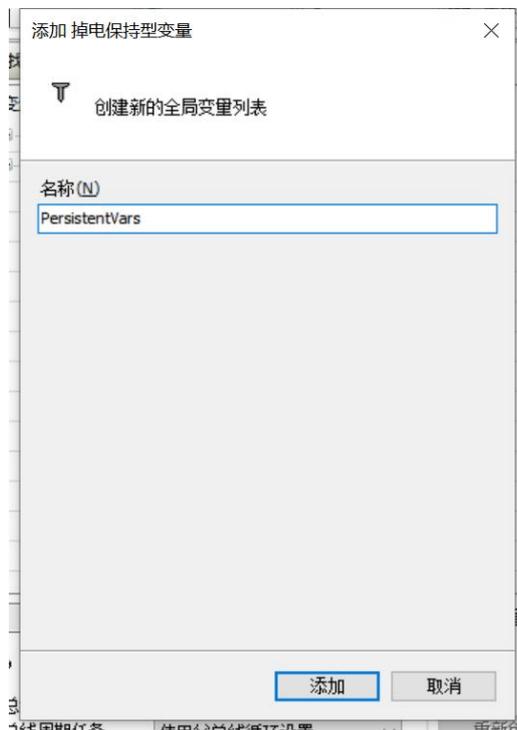
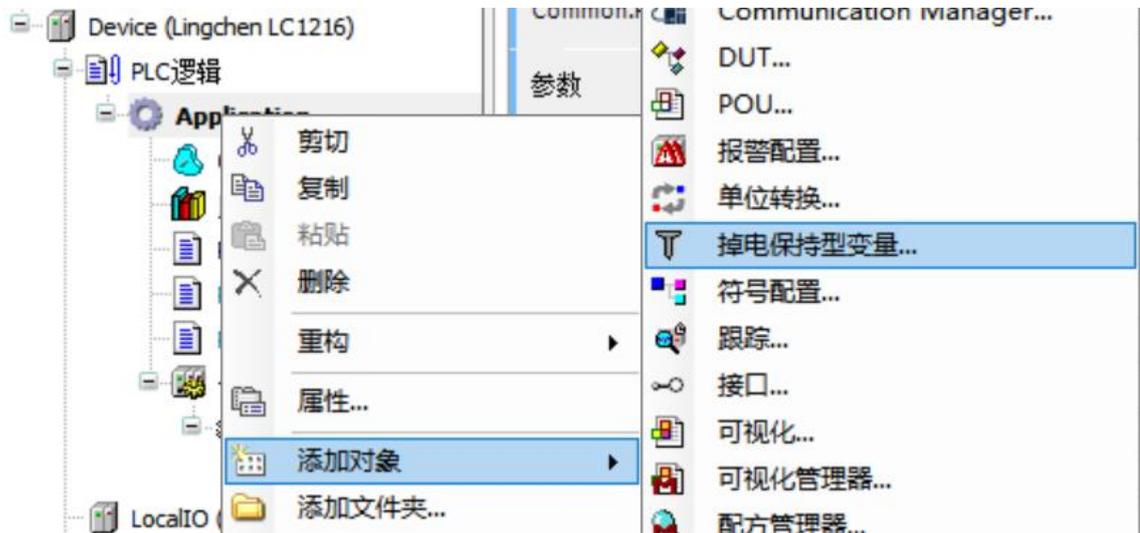


7. 添加完成后，如下图所示。可以配合其他串行从站进行使用。

名称	访问类型	触发器	读偏移	长度	错误处理	写偏移	长度	注
0 Channel 0	Read Holding Registers (函数代码 03)	循环, t#100ms	16#0000	1	保持最后值			
1 Channel 1	Write Single Register (函数代码 06)	循环, t#100ms				16#0000	1	

8. 断电保持功能的使用

1. 添加“掉电保持型变量”对象。右键“Application” - 添加对象 - 掉电保持型变量。



设定列表名称（这里使用默认名称），点击“添加”。

2. 添加完成后，会自动跳转该页面，注释掉首行，声明一个数组变量。

```
1  //{attribute 'qualified_only'}
2  VAR_GLOBAL PERSISTENT RETAIN
3  test:ARRAY [1..20] OF USINT;
4  END_VAR
```

3. 登录 - 启动, 打开掉电保持界面, 展开变量, 随意输入准备数字, “Ctrl+F7” 写入。



表达式	类型	值	准备值	地址	注释
test	ARRAY ...				
test[1]	USINT	1			
test[2]	USINT	0			
test[3]	USINT	2			
test[4]	USINT	0			
test[5]	USINT	3			
test[6]	USINT	0			
test[7]	USINT	0			

4. 断掉 LC1200 电源, 重新上电, 待 PLC 重新运行起来后, 登录查看。

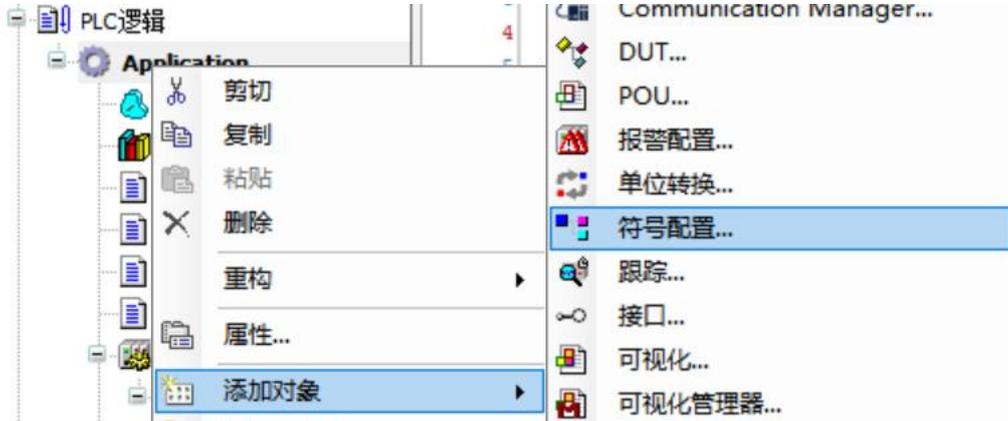


表达式	类型	值	准备值
test	ARRAY ...		
test[1]	USINT	1	
test[2]	USINT	0	
test[3]	USINT	2	
test[4]	USINT	0	
test[5]	USINT	3	
test[6]	USINT	0	

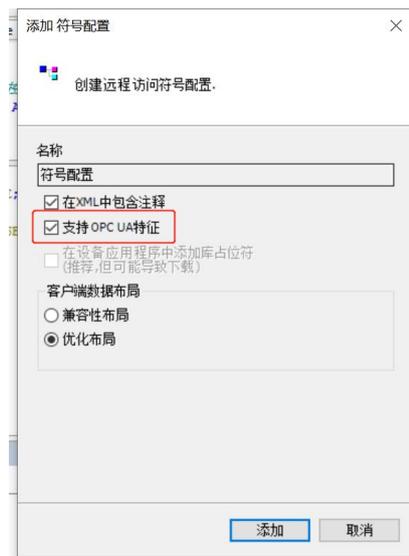
因此, 可将相关变量声明为掉电保持型变量, 掉电后再次上电, 依旧可以保存下来。

9. OPC UA 例程

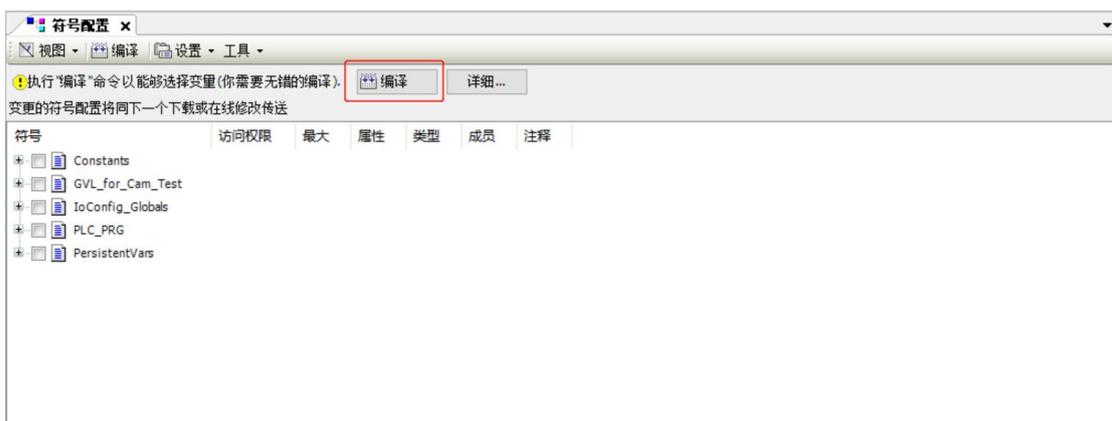
1. 编写好程序之后，右键“Application” - 添加对象 - 符号配置...



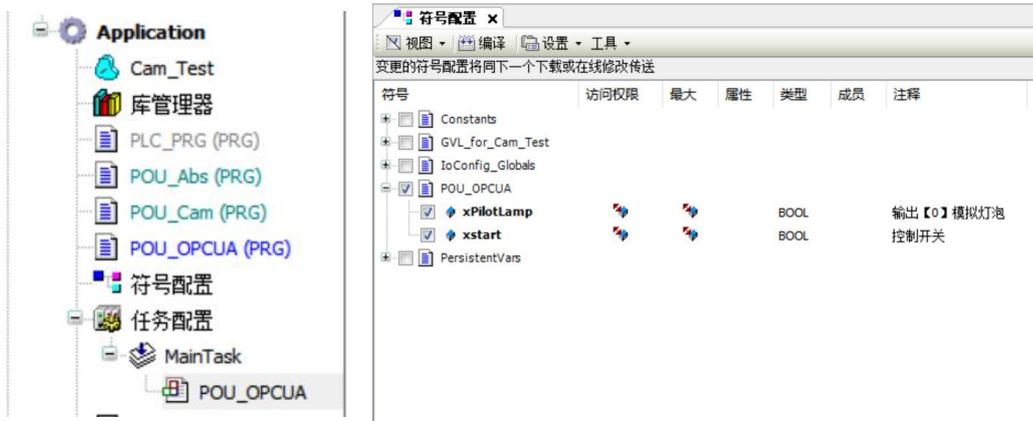
2. 在弹出的“添加 符号配置”界面，注意是否勾选支持 OPC UA 特征，点击“添加”。



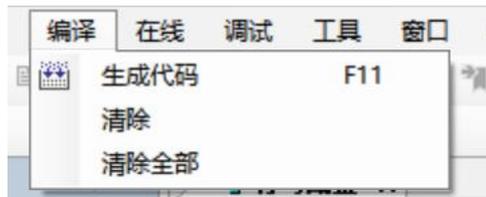
3. 添加完成后，可以看到“符号配置界面”，注意，需要先执行“编译”命



令, 才能够选择需要的变量, 点击“编译”。在此之前, 请确认需要的变量所在 POU 已拖入任务配置, 分配 Task。勾选变量。



4. 勾选完后, 在菜单栏点击“编译 - - 生成代码”。



5. 生成代码结束, 可以在消息栏看到相关信息。

描述	工程	对象	位置
代码和数据总分配的内存容量: 1116348字节			
内存区域0包含 数据, 输入, 输出, 内存 和 Nonsafe data: 大小: 1071344字节, 最高使用的地址: 824108, 最大连续存储器间距: 24723...			
内存区域2包含 持久性数据: 大小: 4096字节, 最高使用的地址: 64, 最大连续存储器间距: 4032字节 (98%)			
内存区域3包含 代码: 大小: 379912字节, 最高使用的地址: 292240, 最大连续存储器间距: 87672字节 (23%)			
构建完整-0错误,0警告: 准备下载			

6. 此时, 可以在该工程文件的存储目录下, 找到生成的 xx.Device.Application.xml 文件。

LC1200快速应用手册-20230216.Device.Application.xml	2023-3-17 19:09	XML Document	2 KB
LC1200快速应用手册-20230216.project	2023-3-17 15:00	CODESYS project	295 KB
LC1200快速应用手册-20230216.project.~u	2023-3-17 19:10	~U 文件	1 KB
LC1200快速应用手册-20230216_project.precompilecache	2023-3-17 15:00	PRECOMPILECACH...	172 KB
LC1200快速应用手册-20230216-AllUsers.opt	2023-3-17 15:01	OPT 文件	1 KB
LC1200快速应用手册-20230216-L209-LCT.opt	2023-3-17 15:01	OPT 文件	351 KB

7. 之后使用人机界面的组态程序, 扫描标签即可。