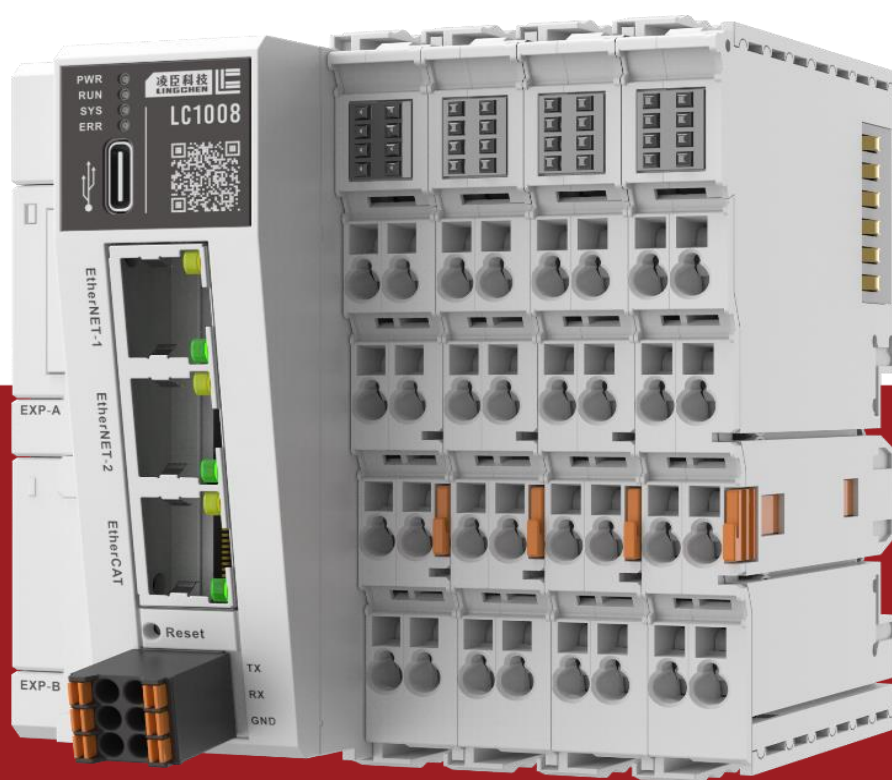


LC1000系列

PLC快速应用手册



公司简介



苏州市凌臣采集计算机有限公司成立于2006年，是一家本着与客户共赢、为客户创造方案价值的经营理念的企业。为客户提供各种测试测量、运动控制、机器视觉、机器人等自动化设备的核心零部件和系统解决方案。凌臣科技于2017年开始创立凌臣采集LCT品牌，研发了包括工控机、PLC控制器、EtherCAT步进驱动器、

PLC/PCIe运动控制卡、远程IO模块、气动阀岛等产品，为我司的客户提供了更具性价比的方案解决。

目录

硬件接口展示	2
电源接线端子排列	4
指示灯/RESET说明	5
1. 设备存储库的使用	6
2. 库文件的使用	8
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	23
5. 基于 EtherCAT 通信的电子凸轮程序例程	28
6. 添加拓展IO	32
7. Modbus	37
7.1 Modbus TCP Master (LC1000做主站、LC1200做从站)	37
7.2 Modbus TCP Slave (LC1000做从站、LC1200做主站)	44
7.2.1 LC本地从站 (LCTModbusTCP Slave)	44
7.2.2 Codesys自带从站 (ModbusTCP_Slave_Device)	50
7.3 Modbus RTU (LC1000串口通讯自测)	55
8. 断电保持功能的使用	61
9. OPC UA例程	63
10. PLC网口IP与MAC地址的设置	65
附录1	70

硬件接口展示

LC1000系列部件完全相同，以LC1008-CPU20TN机型为例进行说明：

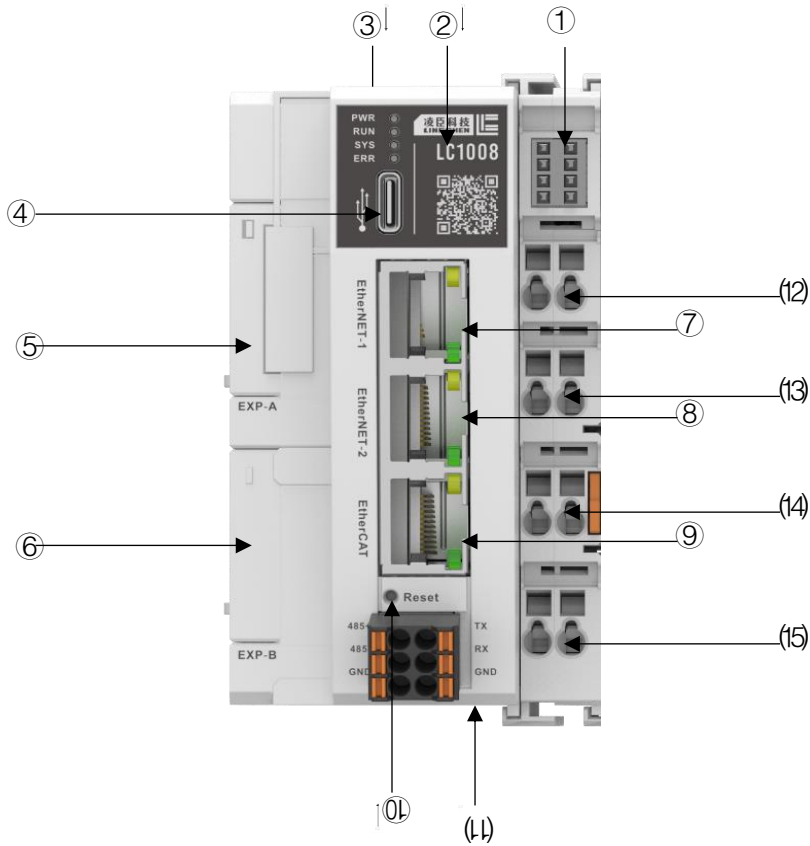


图 1 CPU 模块外部接口示意图

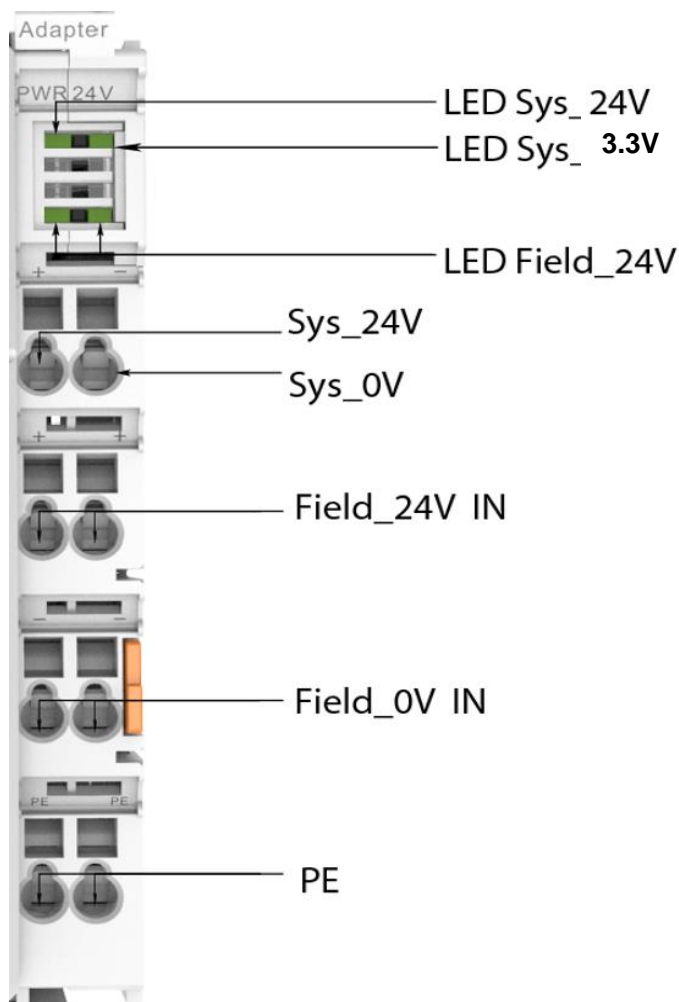
标号	接口名称	功能描述
1	电源模块供电指示灯	直流 24V 1A 电压输入， 颜色：翠绿 常亮：电源模块供电正常 常灭：电源模块供电异常
2	PWR	颜色：翠绿 常亮：表示供电正常 熄灭：表示未供电或供电异常
	RUN	颜色：翠绿 常亮：表示用户程序正在运行 熄灭：表示用户程序停止
	SYS	颜色：翠绿 常亮：启动中 熄灭：未启动/死机 闪烁：PLC 进程运行中（PLC正常工作状态指示）
	ERR	颜色：红 常亮：运行错误

		闪烁：总线错误 熄灭：无错误
3	拨码开关RUN/STOP	控制主机运行/停止
4	USB (Type-C)	程序导入/导出程序
5	扩展卡插槽A用于功能模块扩展	目前A扩展卡槽仅限于RS232/RS485扩展。（后续会开发数字量/模拟量扩展模块）
6	扩展卡插槽B用于功能模块扩展	目前B扩展卡槽仅限于RTC/TF卡扩展
7/8/9	网口LAN0 (eth0)	默认IP地址 eth0:192.168.0.99 1.MODBUS TCP 协议 2.标准以太网功能 3.系统程序调试 4.用户程序下载与调试（只支持 IPv4）
	网口LAN1 (eth1)	默认IP地址 eth1:192.168.2.99 1.MODBUS TCP 协议 2.Ethernet IP 协议 3.标准以太网功能
	网口LAN2 (eth2)	默认IP地址 eth2:192.168.8.99 EtherCAT 协议
10	RESET 按键	复位按键，详情见指示灯、RESET 按键说明部分
11	通讯串口	1路RS485 接口，1路RS232 接口，均支持MODBUS-RTU协议
12	主机电源接口	主机电源接口+、-
13	模块驱动电源接口	驱动电源+
14	模块驱动电源接口	驱动电源-
15	屏蔽保护接地	PE

【 注意】

[1]：若产品出现故障需要断电重启时，务必要在关闭电源后电源指示灯熄灭至少10秒后再进行上电启动操作。

电源接线端子排列



【⚠️注意】

- 系统供电电源Sys_24V/Sys_0V必须与扩展模块驱动供电Field_24V IN/Field_0V电源分开；
- 布线时，避免与动力线（高电压，大电流）等传输强干扰信号的电缆捆在一起，应该分开走线并且避免平行走线。
- LC1000系列主机不带电源端子

指示灯/RESET说明

RESET 按键可以用于清除PLC 内已下载的程序或将网口IP 地址复原。具体操作如下：

- 长按 RESET 按键 2~6s ， PLC 恢复出厂值，擦除已下载的用户工程；
- 长按 6s以上， PLC 的三个网口恢复默认 IP 地址。
- 长按 6s 以上， 断电重启 PLC 再次长按 6s 以上擦除断电保持区记录数据内存；

CPU 模块指示灯：

	常亮	常灭	闪烁	颜色
PWR	有供电	无供电		绿灯
RUN	run	stop		绿灯
SYS	启动中/死机	启动中/死机	PLC进程运行中	绿灯
ERR	运行错误	无错	总线错误	红灯

1. 设备存储库的使用

1.1 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-设备储存库”；

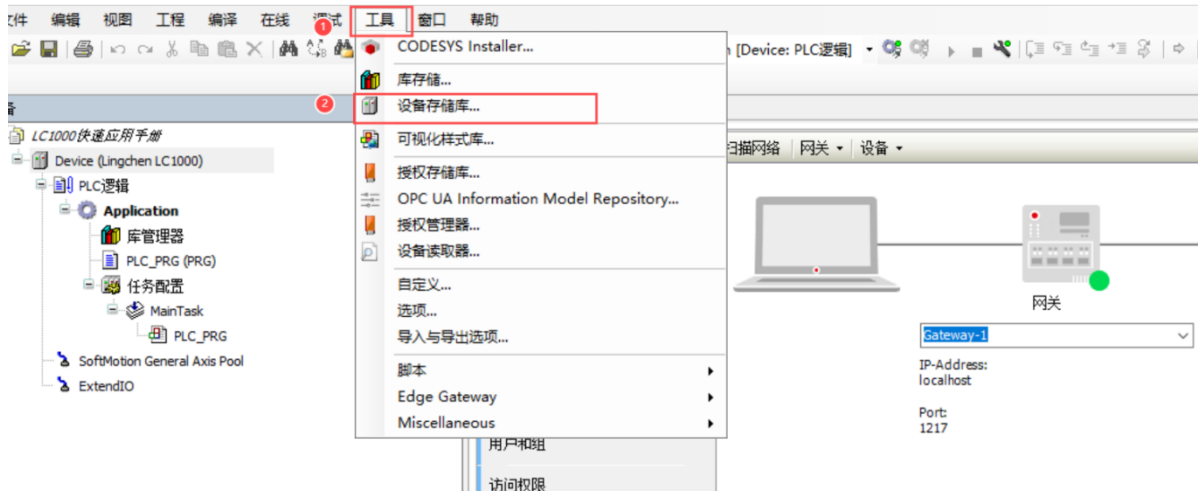


图1.1

1.2 点击“安装”；

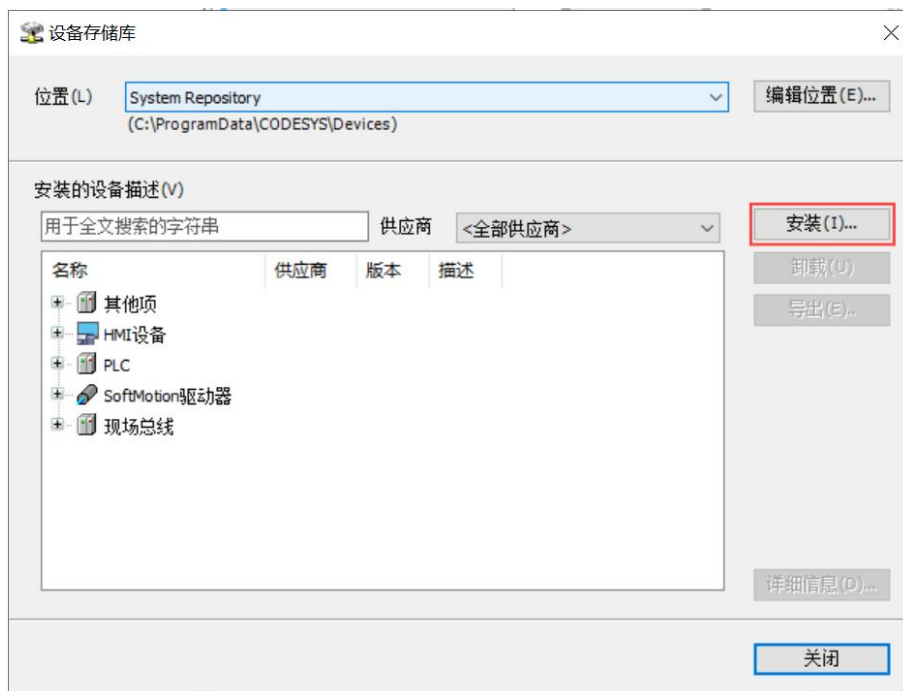


图1.2

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）；

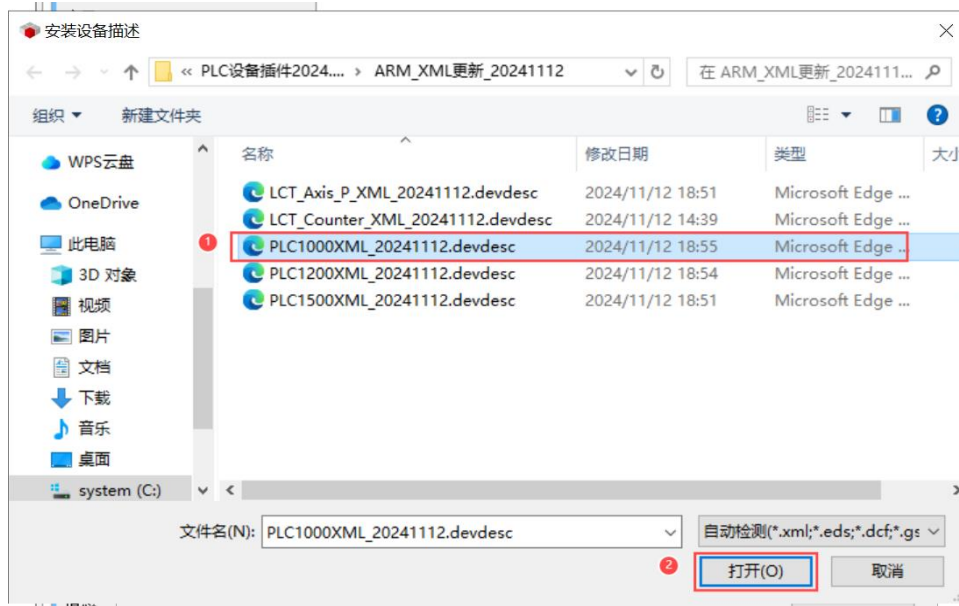


图1.3

1.4 安装成功，点击“关闭”；

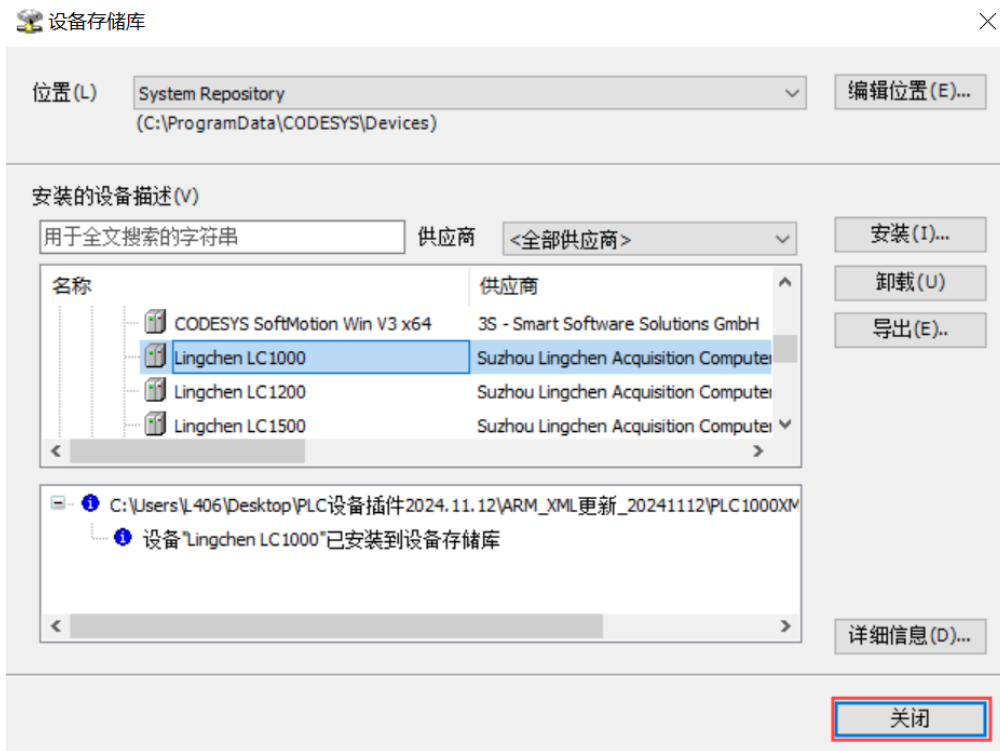


图1.4

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”

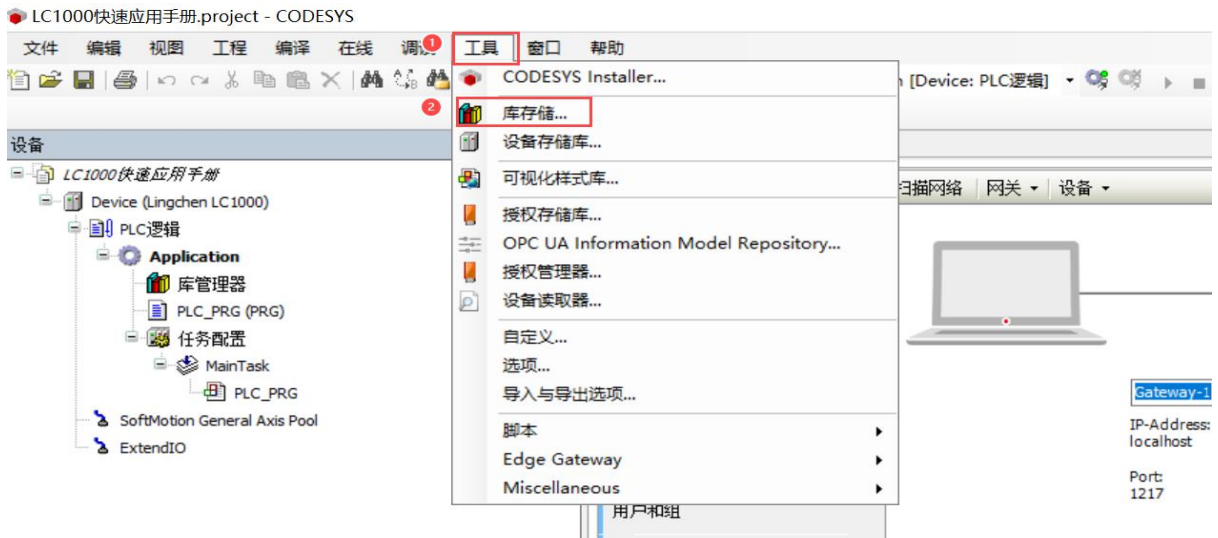


图2.1-1

2.2 点击“安装”；

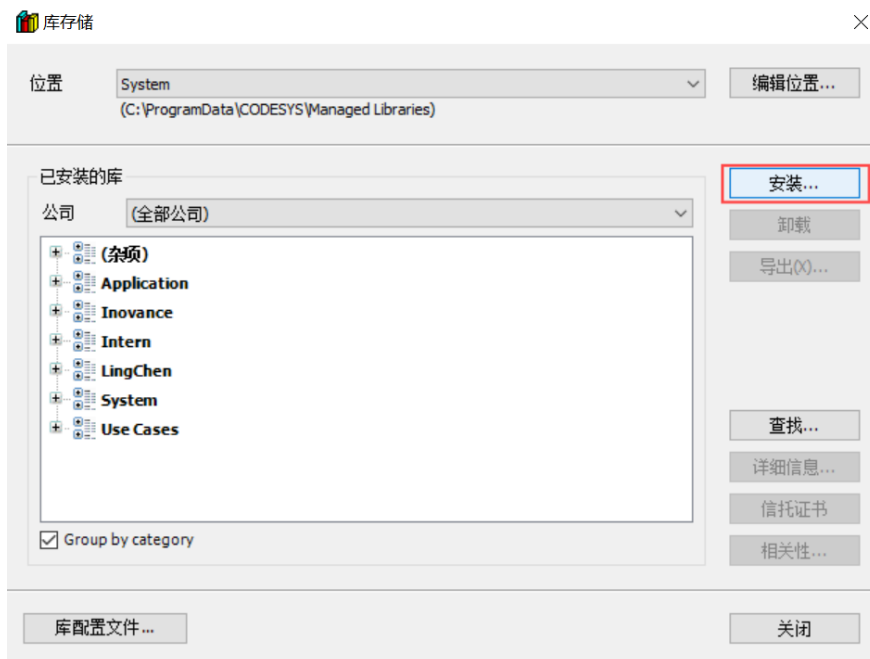


图2.2-1

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）；

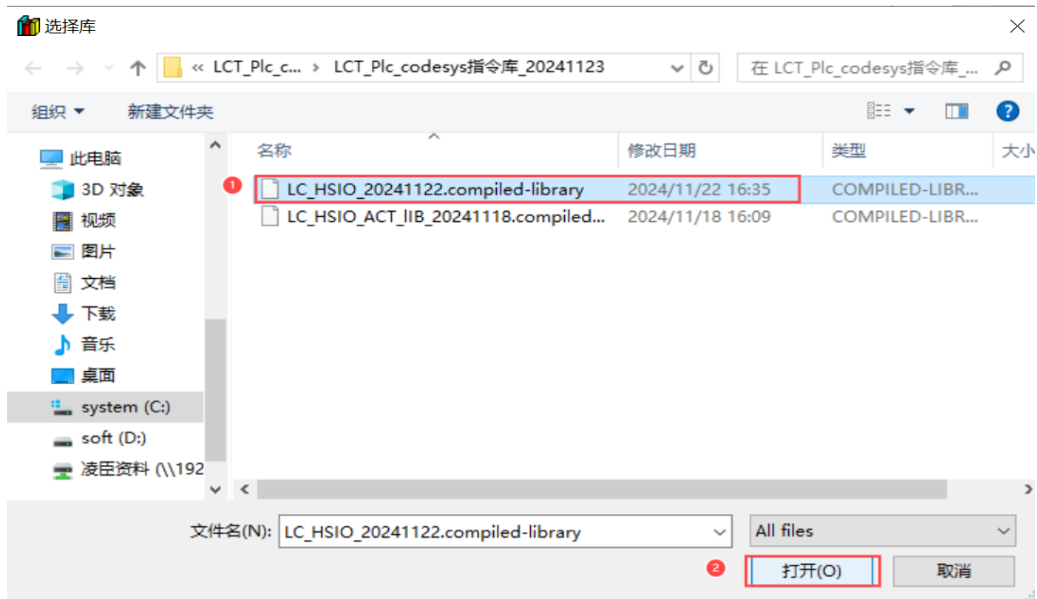


图2.3-1

2.4 添加成功，点击“关闭”。

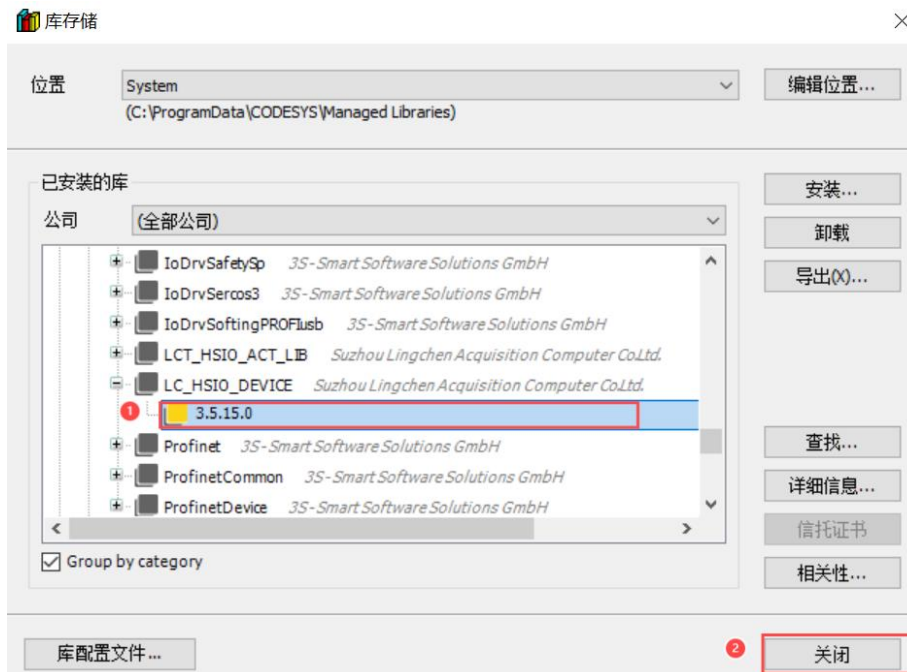


图2.4-1

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...”;

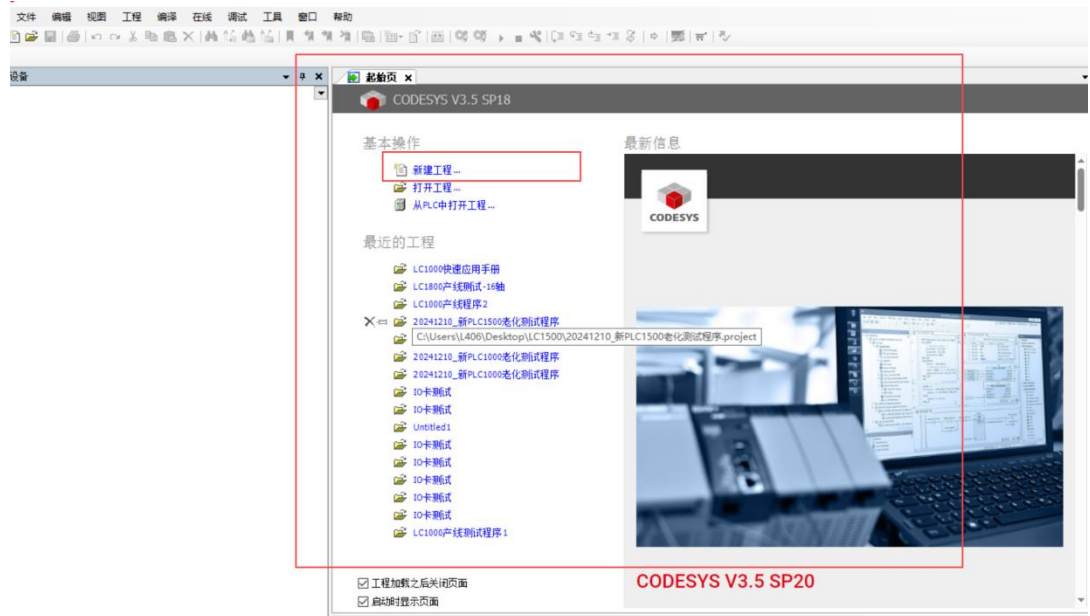


图3.1-1

方法2: 若没有“起始页”, 可以在菜单栏点击“文件—新建工程...”;

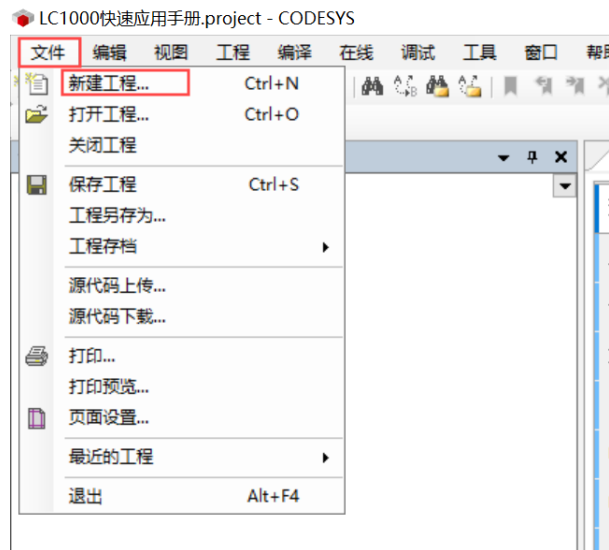


图3.1-2

【注: 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”;

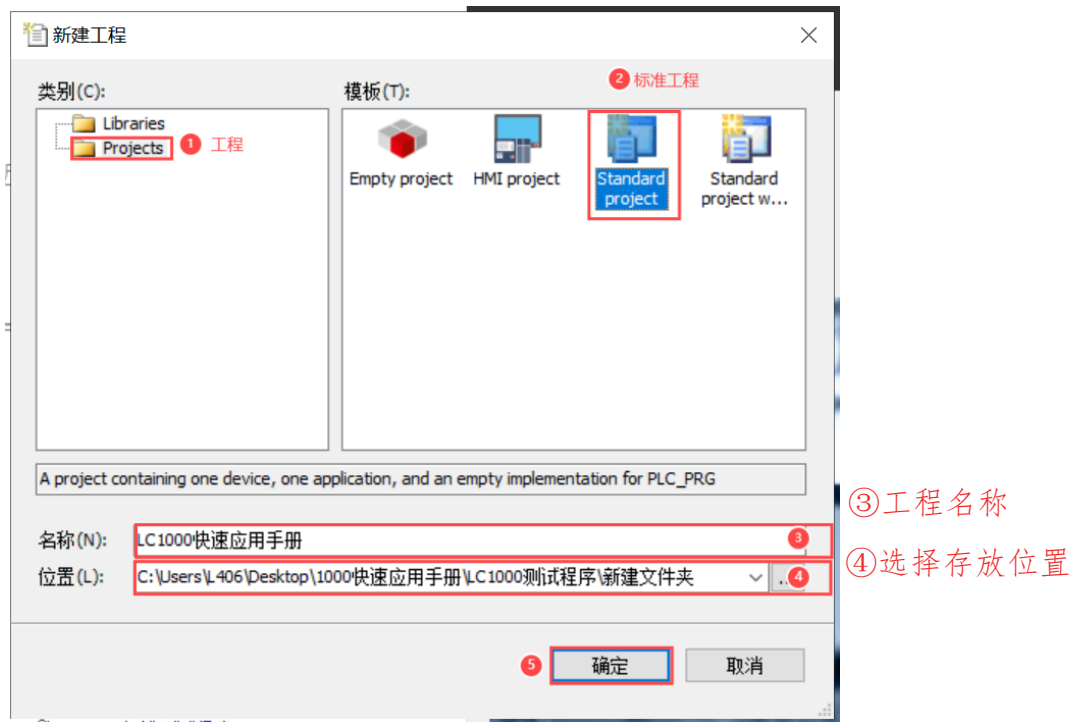


图3.2-1

3.3 选择设备（Lingchen LC1000）与编程语言（后续新建POU可以更改编程语言）后，点击“确定”；

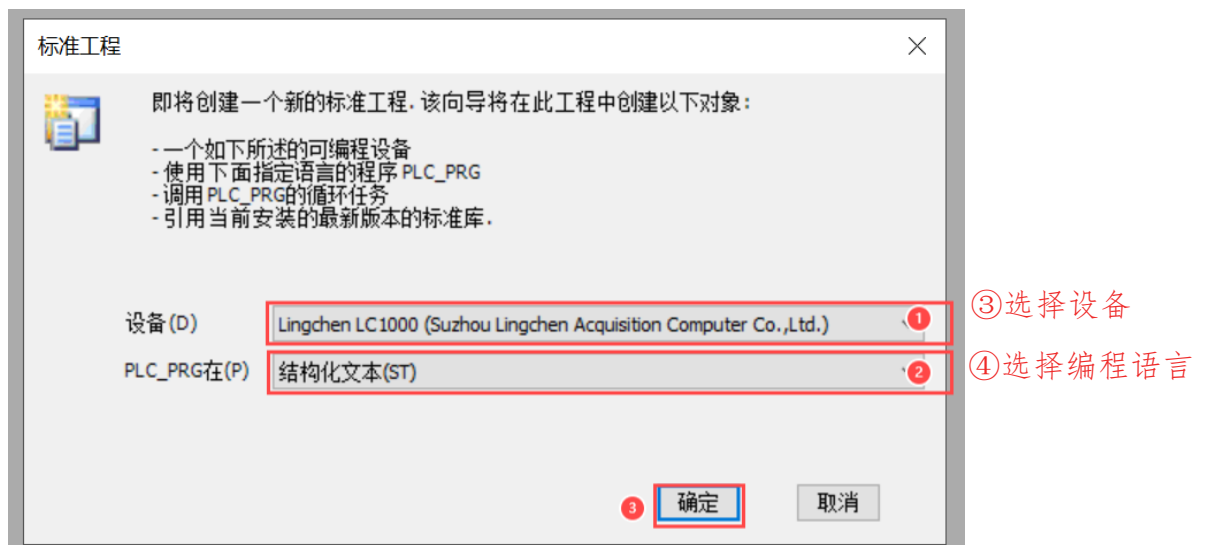


图3.3-1

【注：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；

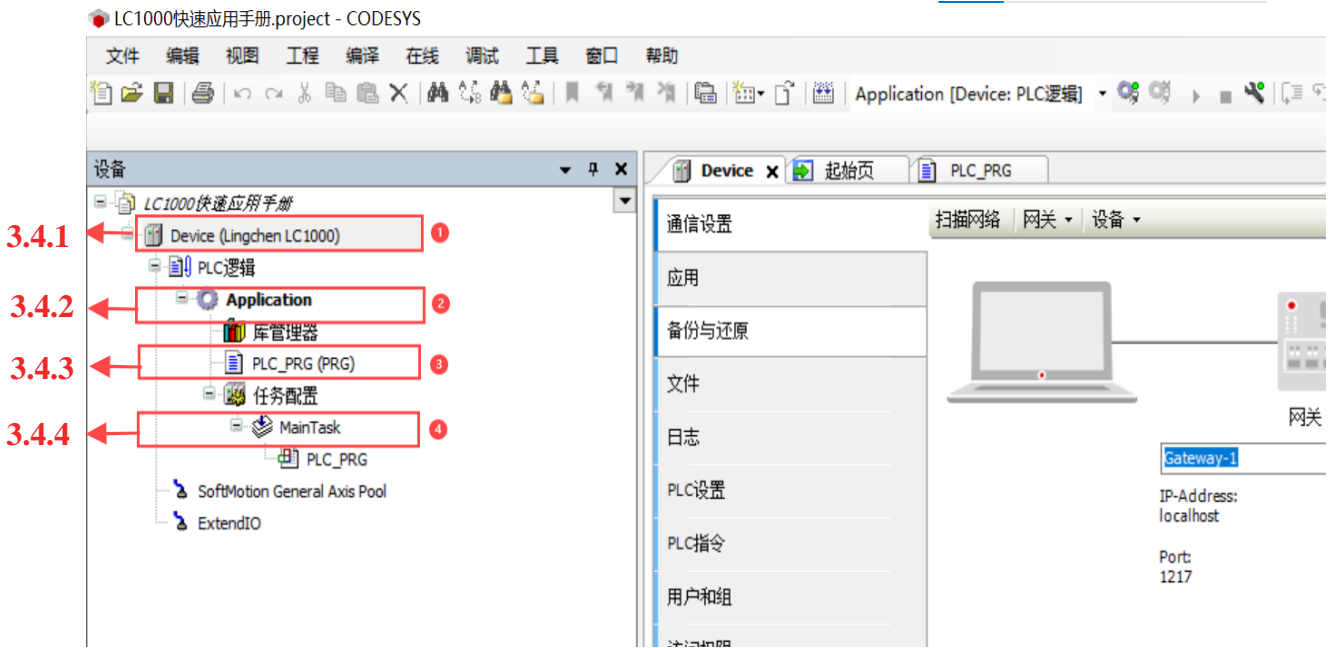


图3.4-1

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；

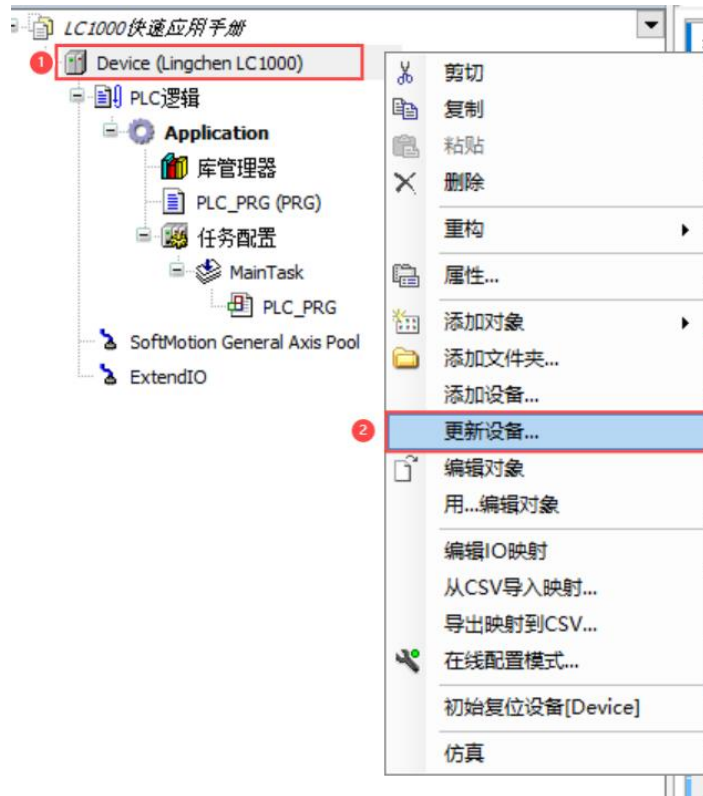


图3.4-2

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）。

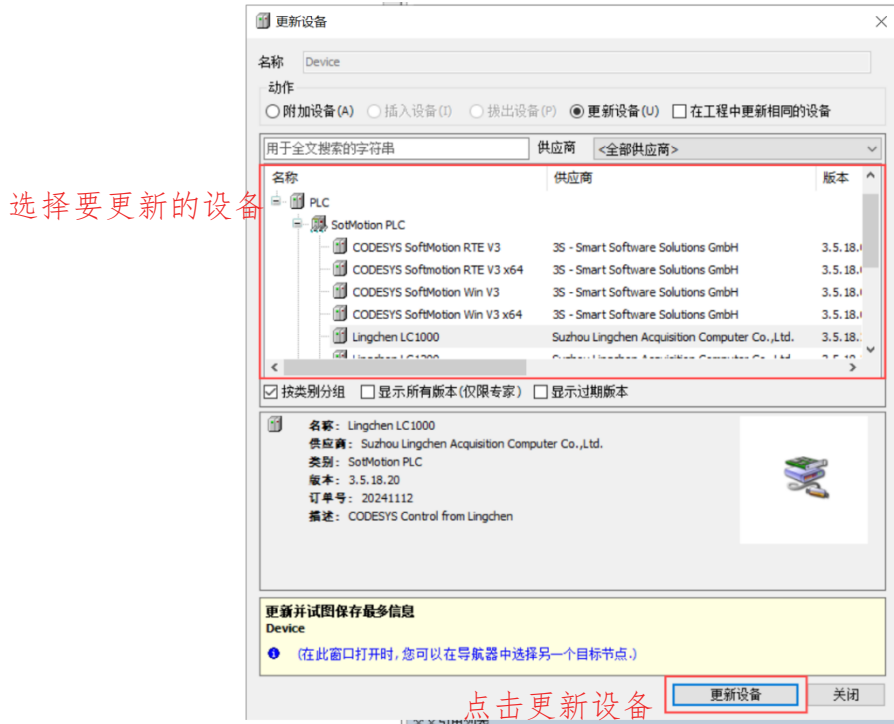


图3.4-3

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。

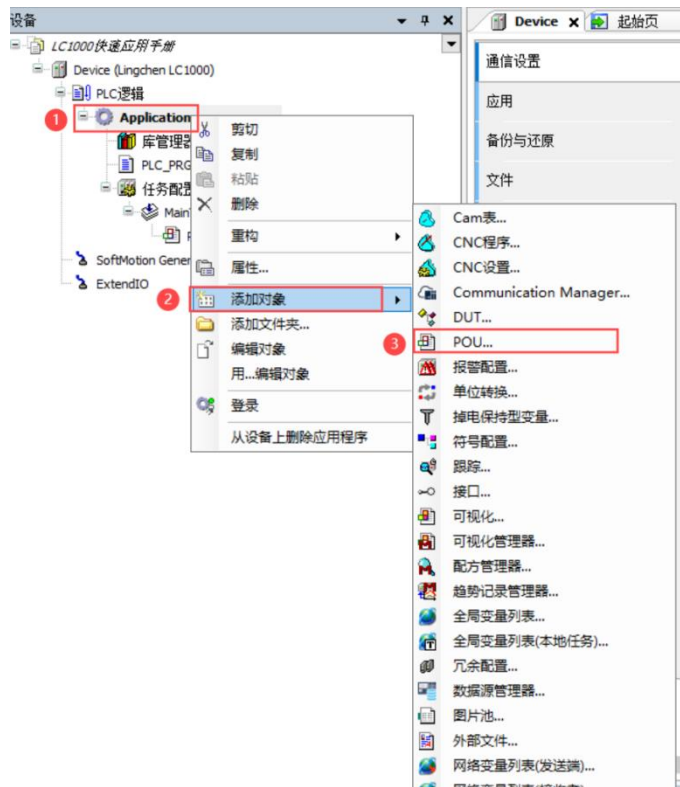


图3.4-4



图3.4-5

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。

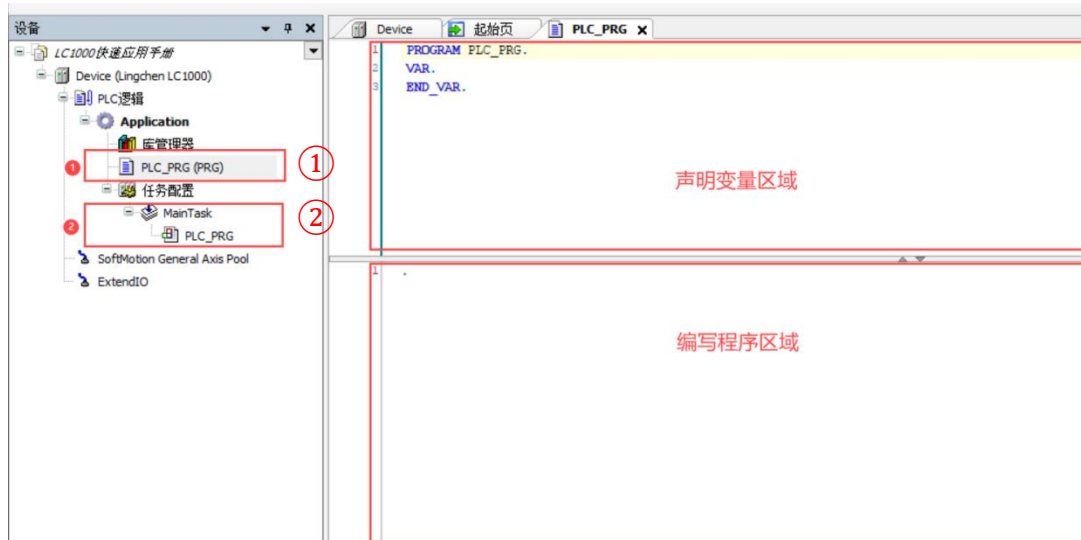


图3.4-6

【注：要将写好的程序①拖入任务配置②下，方可编译调用。】

3.4.4 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-LC1200”后，点击“确定”。

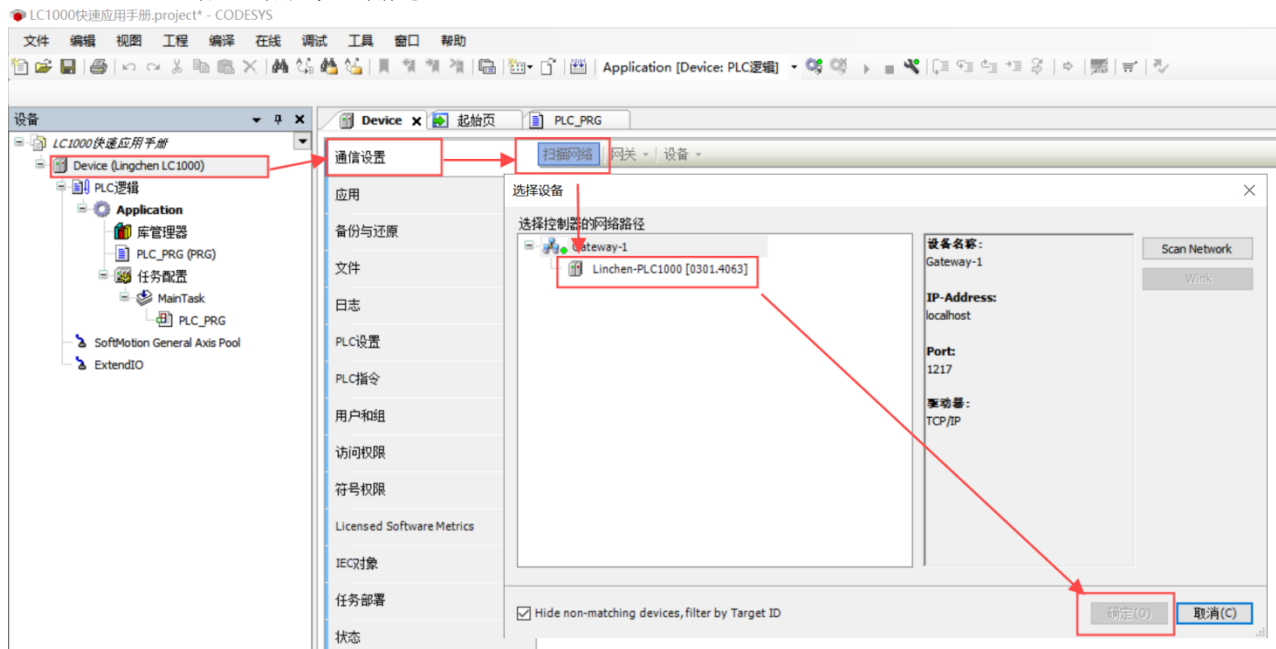


图3.6-1

【注意：所连电脑的TCP网段要与所选PLC的网口网段一致（Ether0:0、Ether1:2、Ether2:8）】

初次连接，会有以下弹窗，点击“是”即可。

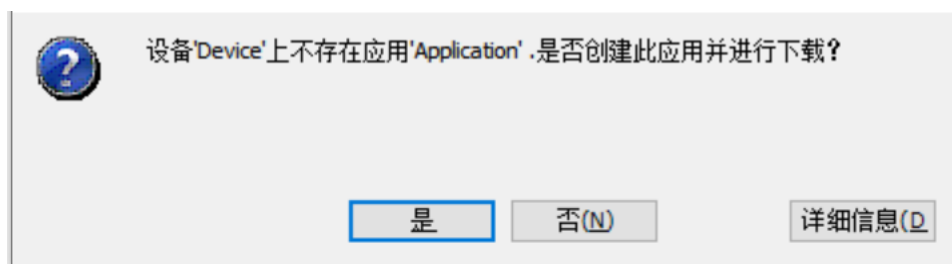


图3.6-2

连接成功，如图3.6-3。

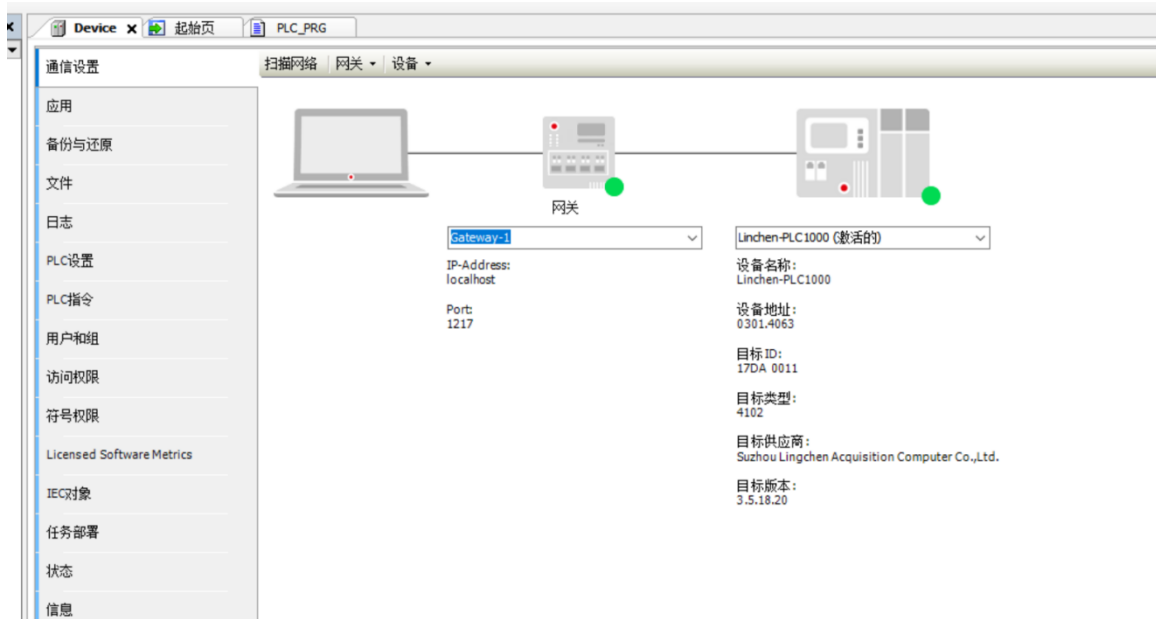


图3.6-3

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。

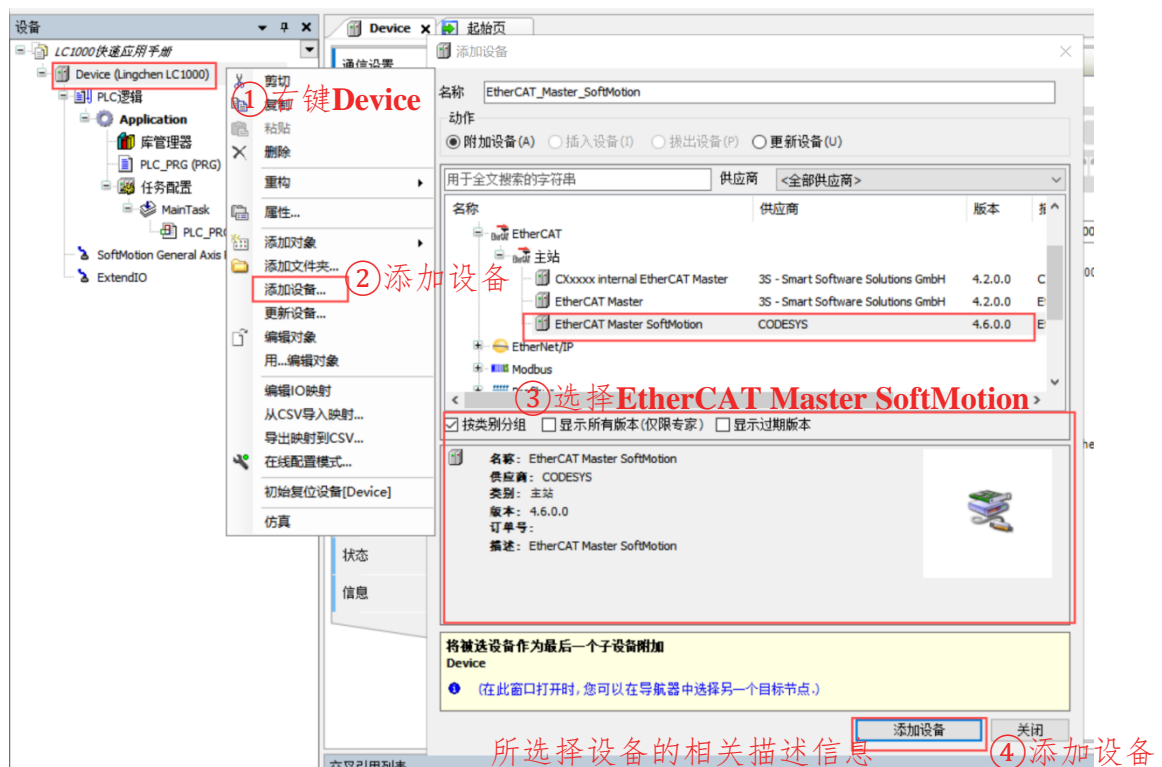


图3.6-4

添加后如图3.6-5，同时系统会分配一个任务EtherCAT_Task，可配置EtherCAT任务相关参数。



图3.6-5

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“AS760N（安驰伺服驱动_单轴）”。添加前必须先安装此设备的描述文件（.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”（图3.6-6），在“添加设备”弹窗中可以直接搜索设备关键词或者筛选对应供应商。设备型号为“AS760N_sAxis_V0.10”，选中该设备，点击“添加设备”（或直接双击要添加的设备）（图3.6-7）。

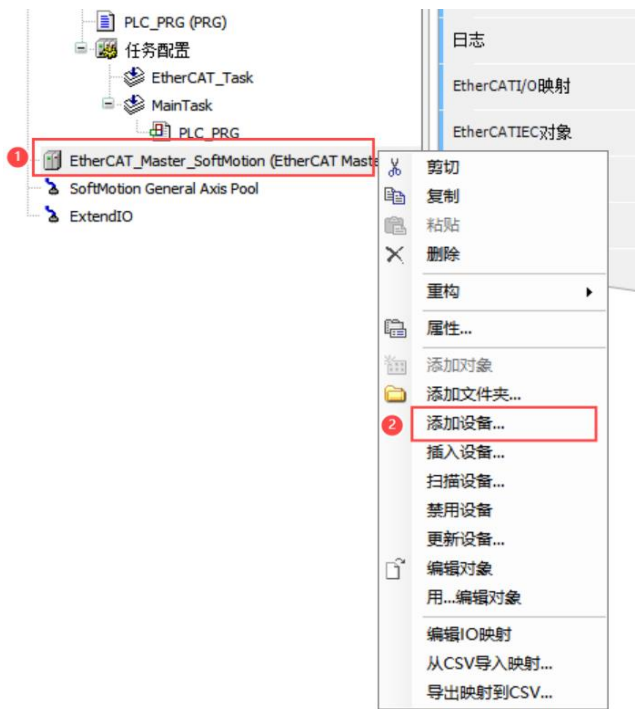


图3.6-6

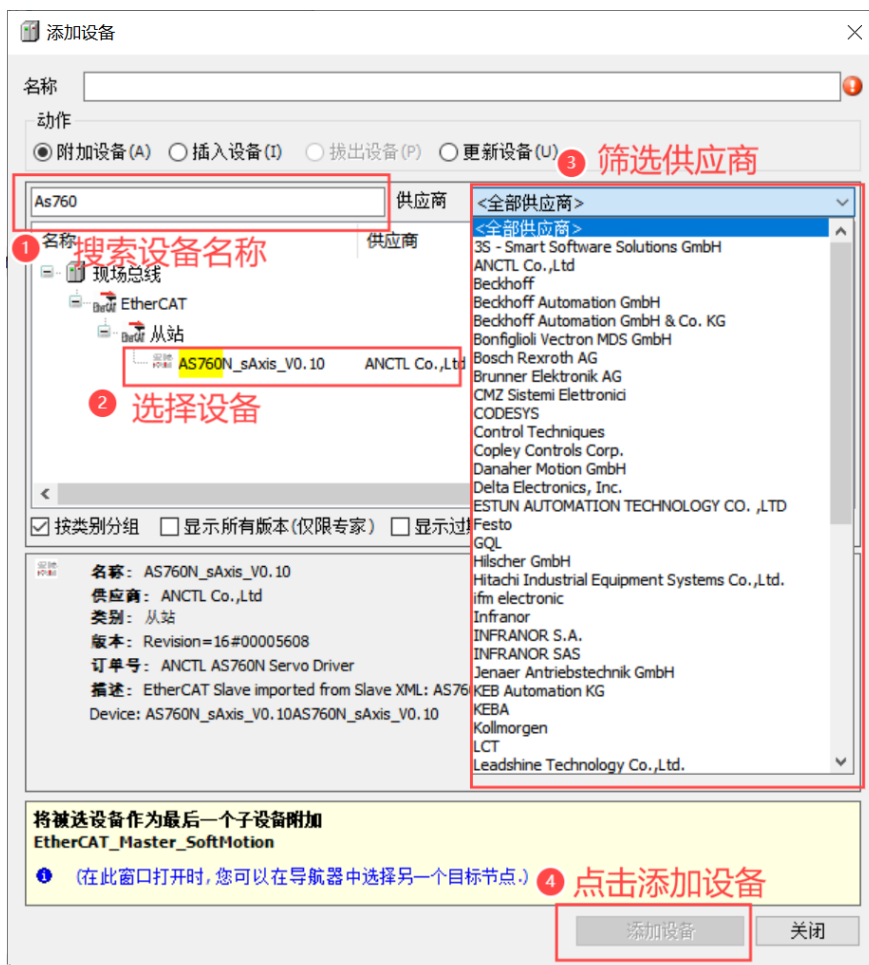


图3.6-7

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（图3.6-8、3.6-9）；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“eth1”（图3.6-10）。



图3.6-8



图3.6-9

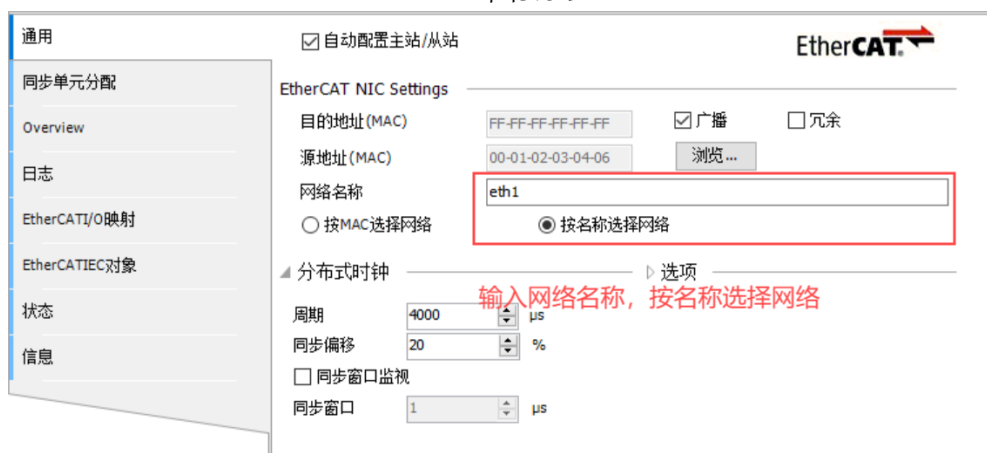


图3.6-10

(2) 选定EtherCAT网口后，点击“登录”（或者Alt+F8）或“在线配置模式”。



图3.6-11

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

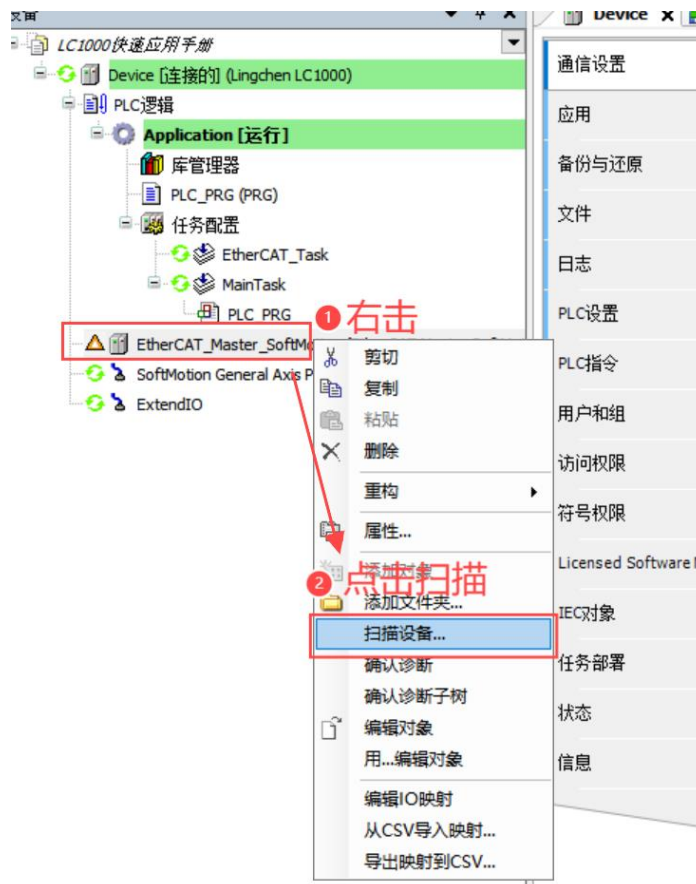


图3.6-12



图3.6-13

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”。

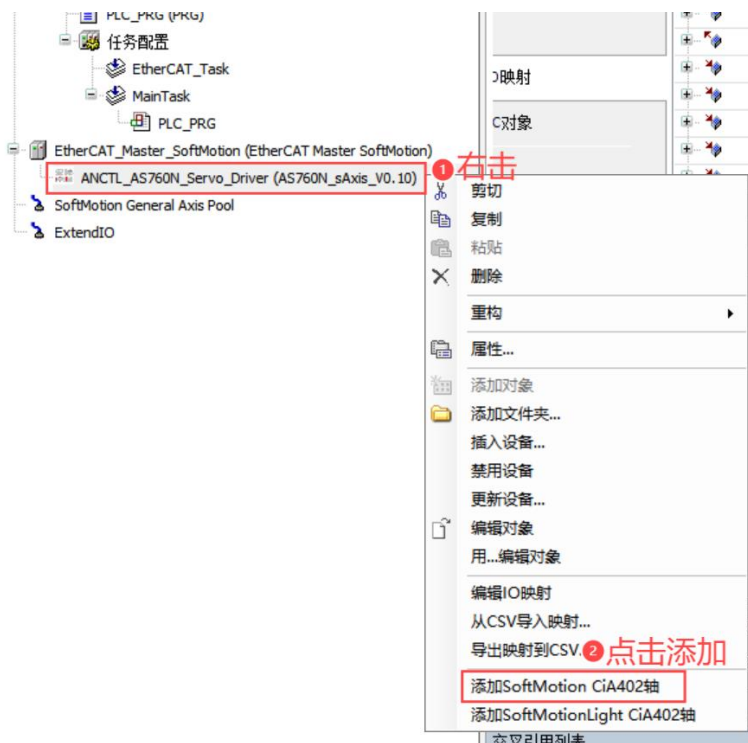


图3.6-14

(5) 添加完成，为方便编程，可以对新添加的轴重命名，“右键-重构-重命名”。

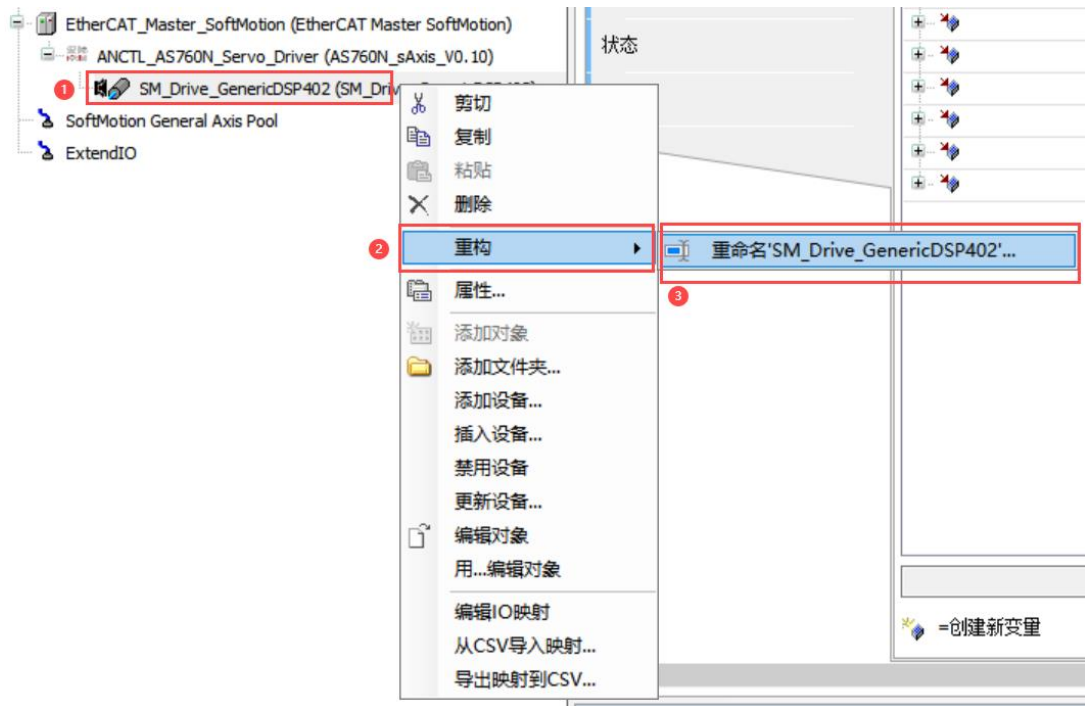


图3.6-15

(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。

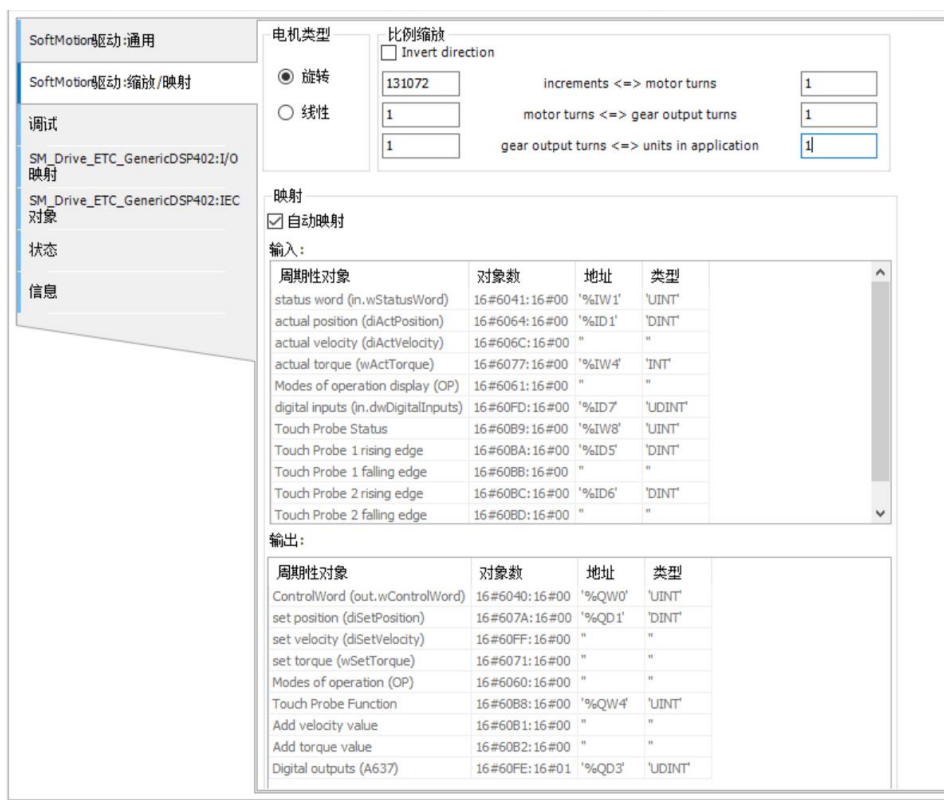


图3.6-16

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），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”，搜索到库“SM3_Basic”然后选择添加，也可以通过此方法添加更多的库。】

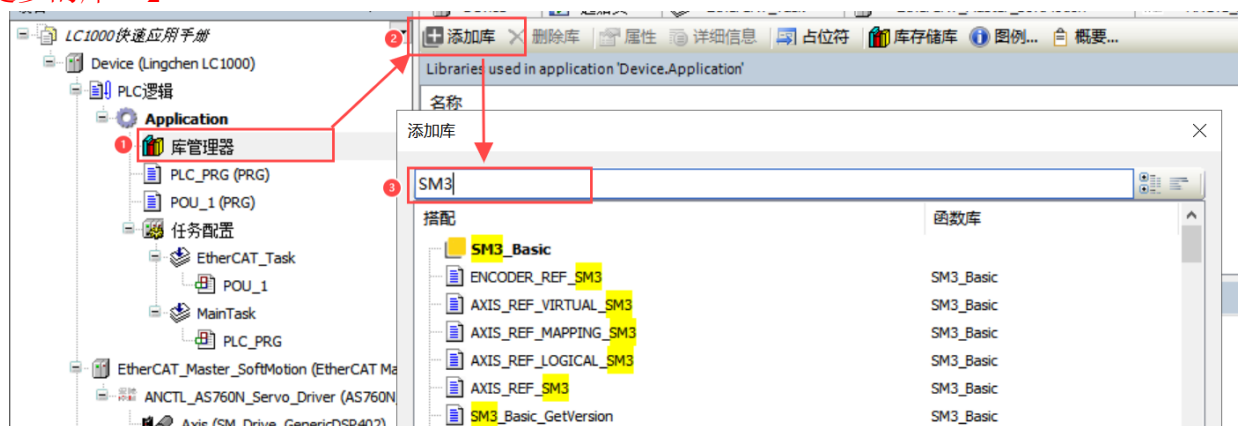


图4.1

(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O 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。



图4.2

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_1）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_1”，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。

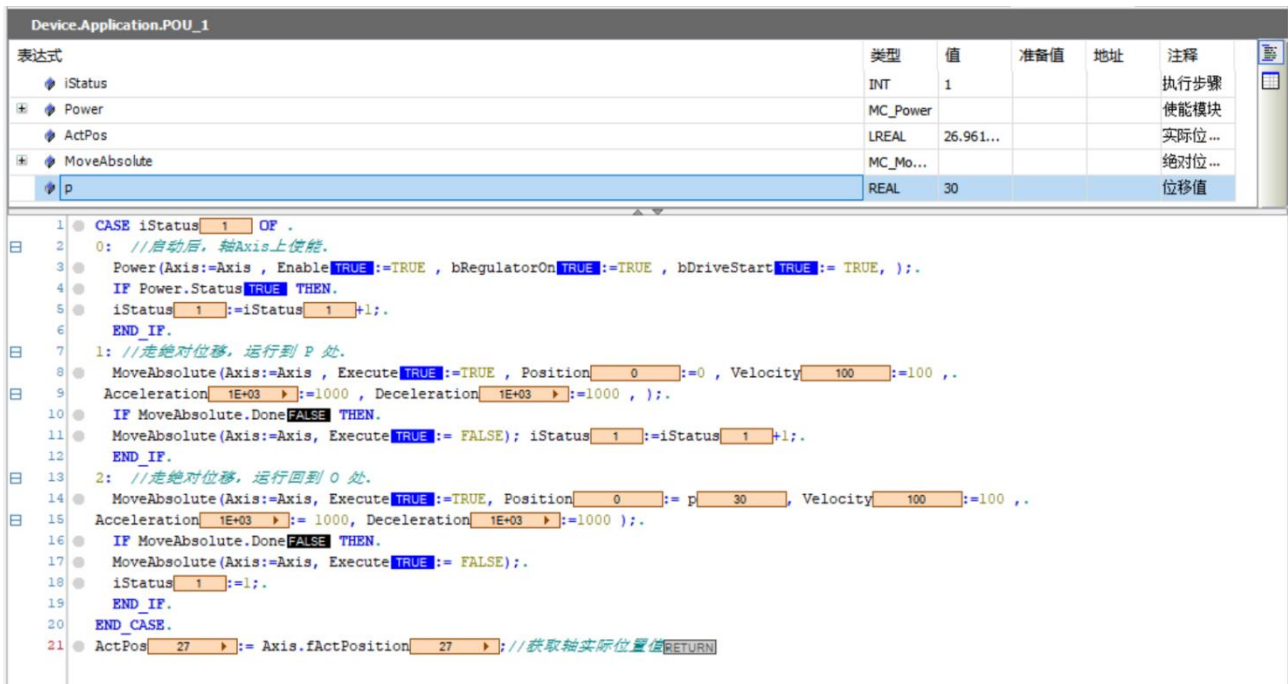


图4.3

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“60”，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

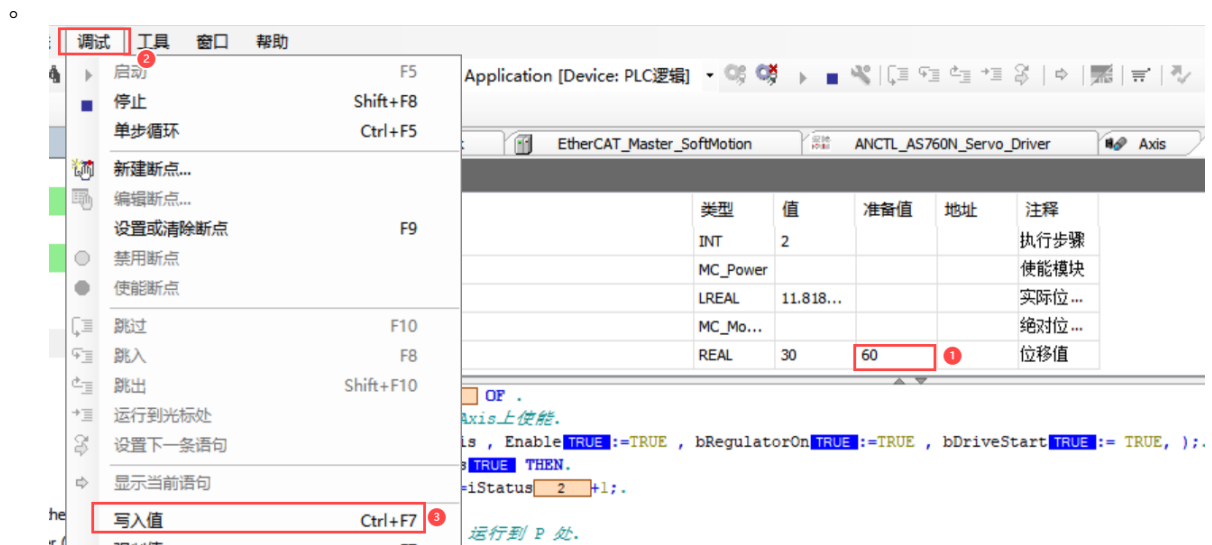


图4.4

再写一个简单的轴回原程序案例，来测试轴回原功能：

(1) 在设备栏中双击轴驱动设备—打开启动参数页面—点击“添加”—依次选择轴回原需要参数名称（图中选择的是“Homing method”回原方式）—点击“确定”。

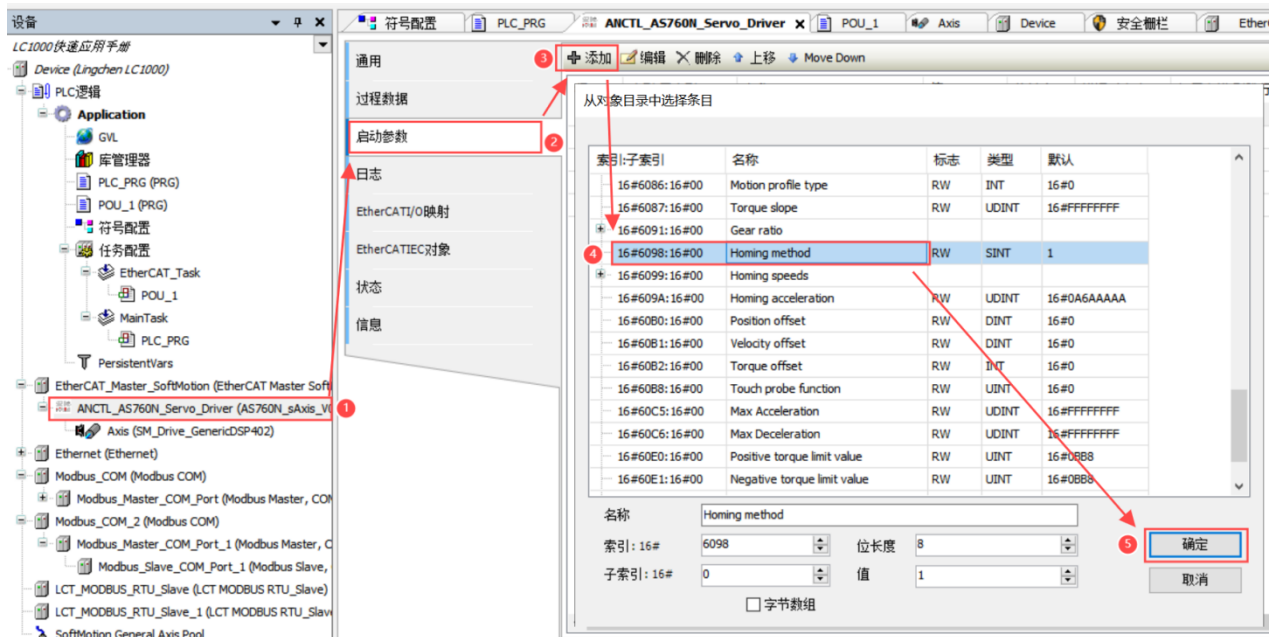


图4.5

(2) 如下图所示，依次添加好轴回原所需要的参数，并配置参数值根据图中行号排列说明：

- 1: 轴第二加速度，这里配置成1000，通常配置成第一速度的十分之一。
- 2: 轴回原加速度，这里配置成100000，通常配置成第一速度的十倍。
- 3: 轴回原第一速度，这里配置成10000和轴单圈脉冲数相同（可根据实际需求配置）。
- 4: 轴回原方式，这里选第34种回原（正方向找确定index点），目前回原方式有35种，可根据自身需求选择回原方式。

行	索引 子索引	名称	值	位长度	错误时中止	如果有错,则跳行	下一行	注释
1	16#6099:16#02	Speed during search for zero	1000	32	<input type="checkbox"/>	<input type="checkbox"/>	0	
2	16#609A:16#00	Homing acceleration	100000	32	<input type="checkbox"/>	<input type="checkbox"/>	0	
3	16#6099:16#01	Speed during search for switch	10000	32	<input type="checkbox"/>	<input type="checkbox"/>	0	
4	16#6098:16#00	Homing method	34	8	<input type="checkbox"/>	<input type="checkbox"/>	0	
5	16#6060:16#00	Modes of operation	8	8	<input type="checkbox"/>	<input type="checkbox"/>	0	Modes of opera

图4.6

(3) 建立一个POU并且调用到任务配置中

(4) 打开编程界面POU_1（详情参照3.4.3），在变量声明区域添加变量，如下所示

VAR

ActPos:LREAL; //实际位置值

Power:MC_Power; //使能模块

MoveAbsolute:MC_MoveAbsolute; //绝对位移模块

```
ExAbs:BOOL;//轴绝对定位触发
```

```
Home:MC_Home;//轴回原模块
```

```
Ex_home:BOOL;//轴回原触发
```

```
homedone:BOOL;//轴回原完成
```

```
END_VAR
```

(5) 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，先触发绝对定位到轴位置100，再触发轴回原，是轴回到原点，轴置位置为0。

```
ActPos:= Axis.fActPosition;//获取轴实际位置值
```

```
Power(Axis:=Axis , Enable:=TRUE , bRegulatorOn:=TRUE , bDriveStart:= TRUE, );
```

```
MoveAbsolute(Axis:=Axis , Execute:=ExAbs , Position:=100, Velocity:=100 ,Acceleration:=1000 ,  
Deceleration:=1000 , );
```

```
Home(Axis:= Axis, Execute:=Ex_home , Position:=0 , Done=>homedone , Busy=> , );
```

(6) “登录—启动”，赋值触发绝对定位运动，让轴走到位置100。

```
END_CASE*);  
ActPos 100 := Axis.fActPosition 100 ; //获取轴实际位置值。  
Power(Axis:=Axis , Enable TRUE :=TRUE , bRegulatorOn TRUE :=TRUE , bDriveStart TRUE := TRUE, );  
MoveAbsolute(Axis:=Axis , Execute TRUE :=ExAbs TRUE , Position 100 :=100, Velocity 100 :=100 ,.  
Acceleration 1E+03 :=1000 , Deceleration 1E+03 :=1000 , );  
Home(Axis:= Axis, Execute FALSE :=Ex_home FALSE , Position 0 :=0 , Done FALSE =>homedone FALSE , Busy=> , );  
(*Power/
```

(7) 赋值触发轴回原，轴回原完成后回到原点，此时轴位置为0。

```
END_CASE*);  
ActPos 0 := Axis.fActPosition 0 ; //获取轴实际位置值。  
Power(Axis:=Axis , Enable TRUE :=TRUE , bRegulatorOn TRUE :=TRUE , bDriveStart TRUE := TRUE, );  
MoveAbsolute(Axis:=Axis , Execute FALSE :=ExAbs FALSE , Position 100 :=100, Velocity 100 :=100 ,.  
Acceleration 1E+03 :=1000 , Deceleration 1E+03 :=1000 , );  
Home(Axis:= Axis, Execute TRUE :=Ex_home TRUE , Position 0 :=0 , Done TRUE =>homedone TRUE , Busy=> , );  
(*Power/
```

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 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5.2）所示（详情请参考3.6.4 单轴简单用户控制程序编写）。



图5.1

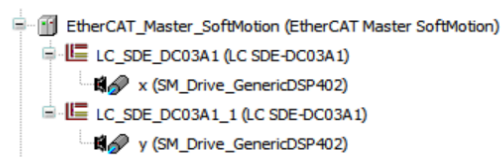


图5.2

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。

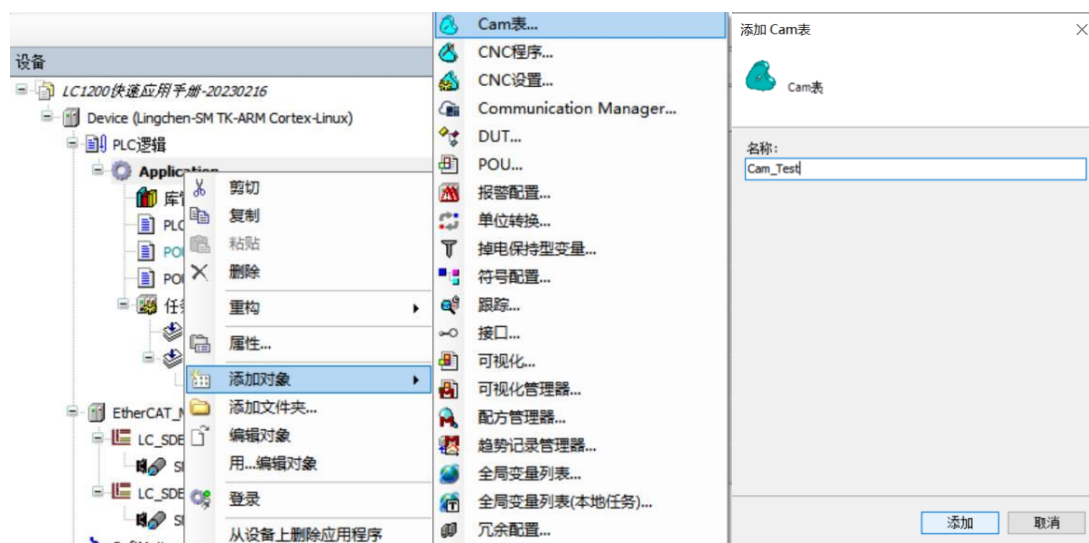


图5.3

点击“添加”。添加完成后，Cam表界面如图所示。

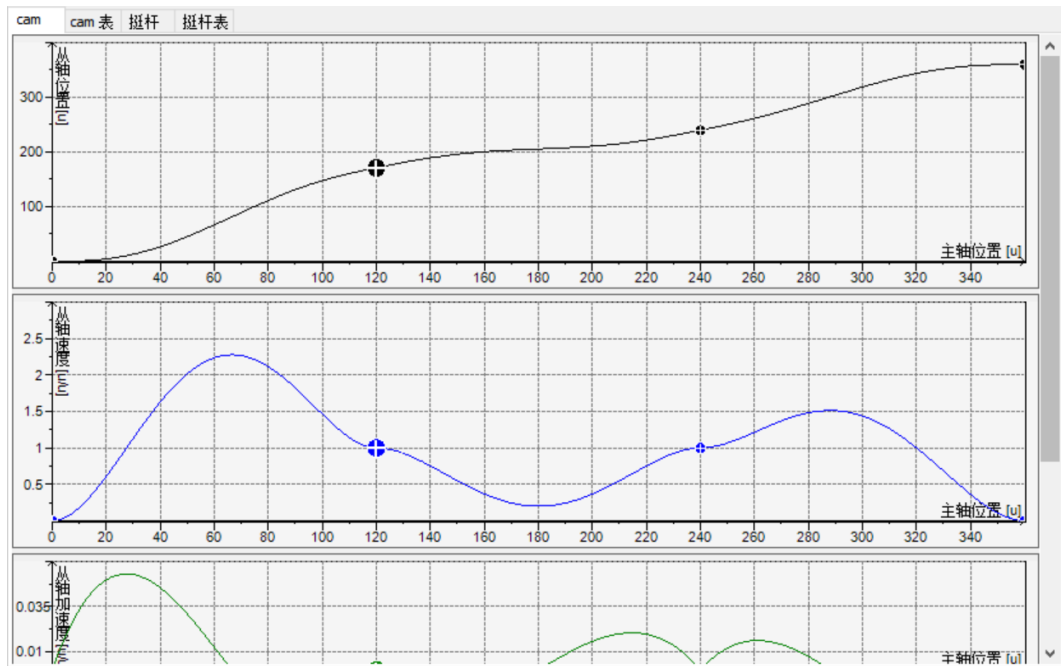


图5.4

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。



图5.5

6. 添加拓展IO

LC1000PLC可以通过两种方式添加拓展IO，第一种方式是在其自带的本地拓展IO通道上直接添加拓展IO设备模块（以1486和2486模块为例）；第二种是通过总线通讯连接远程IO耦合器，添加IO设备模块（以LC1100和1488、2488模块为例）。

第一种方式：

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. 在设备栏在，右键“ExtendIO”，选择“添加设备”。

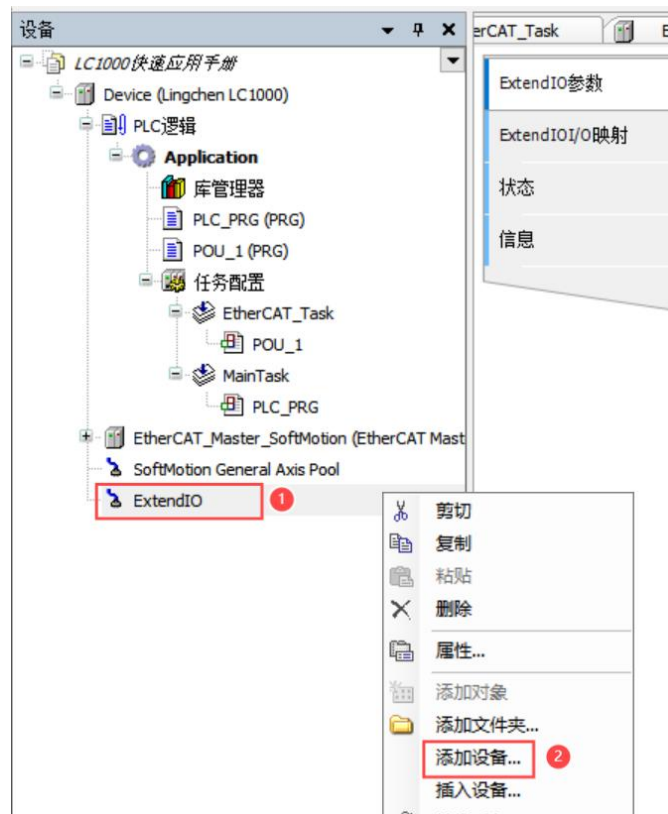


图6.1

3.进入添加设备弹窗，可以搜索设备关键词，也可以通过供应商筛选寻找设备，选择对应设备后点击“添加设备”即可。



图6.2

4. “登录-启动”，点击LD2486（输出模块）或LD1486（输入模块）打开界面，点击“ModuleI/O映射”子界面，可以在此为输出赋值或检测输入当前值。

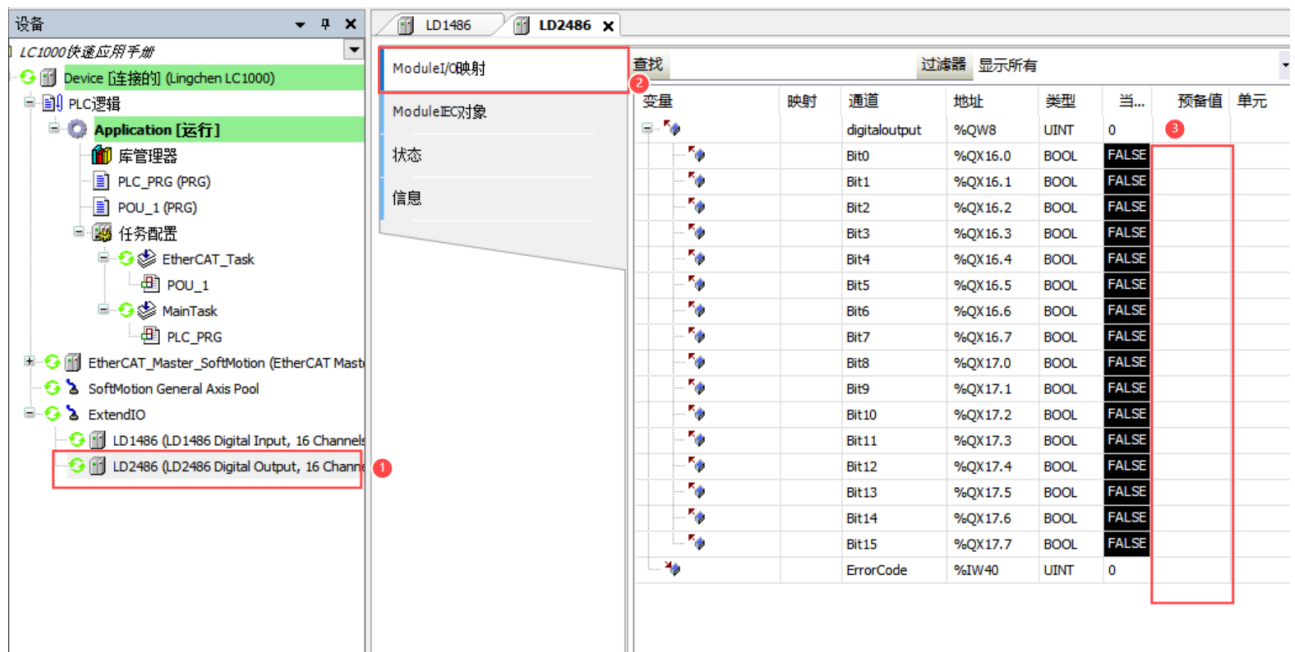


图6.3

第二种方式:

- 1.同上，建立工程和导入设备文件。
2. 在设备栏，右键“Device”，选择“添加设备”；

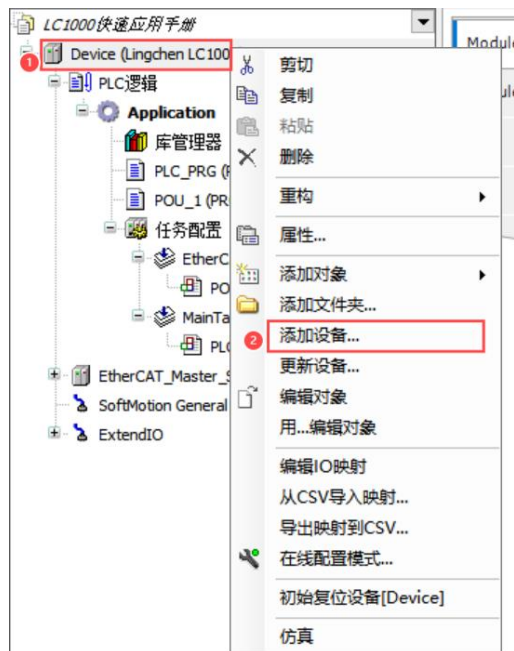


图6.4

3.添加主站;

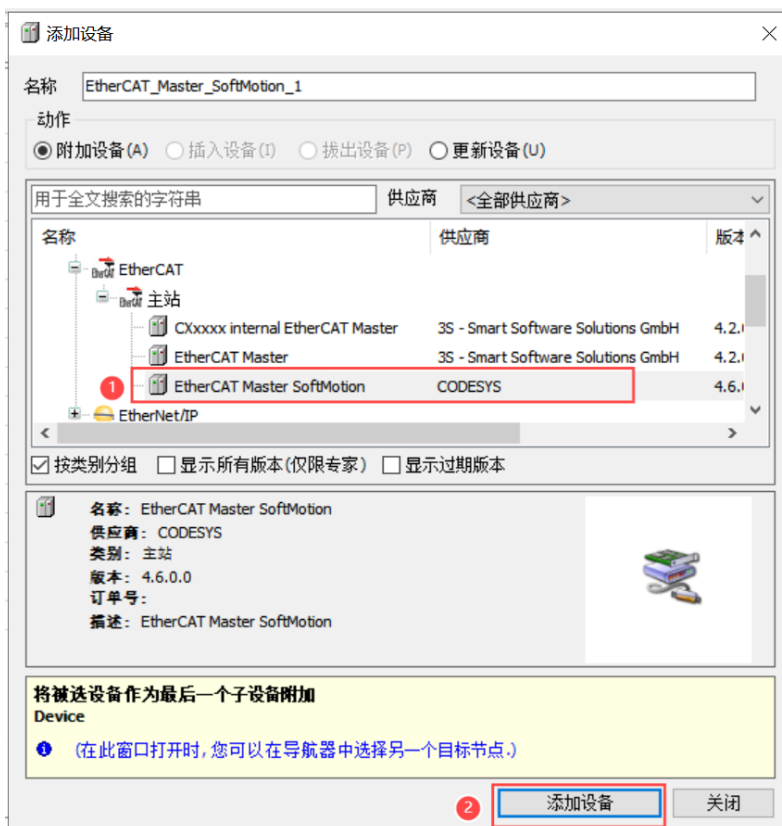


图6.5

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；

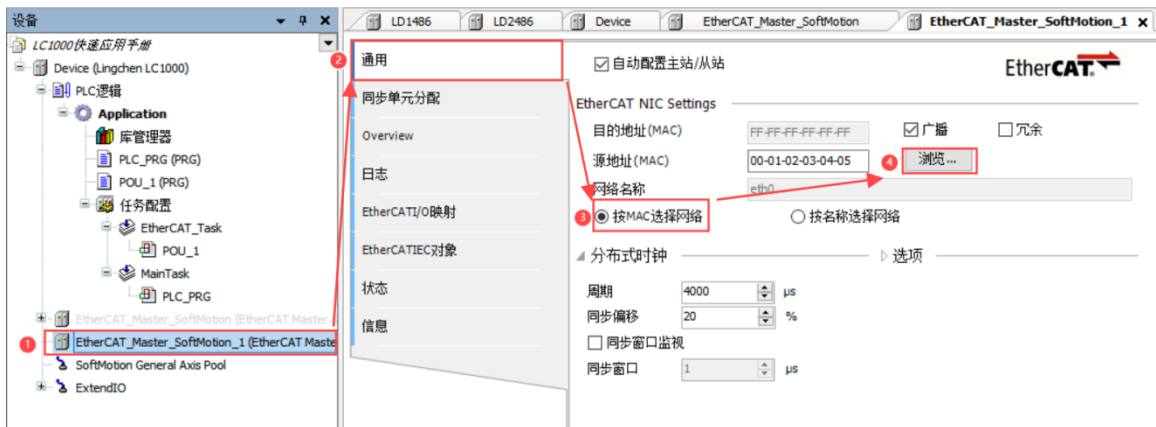


图6.6

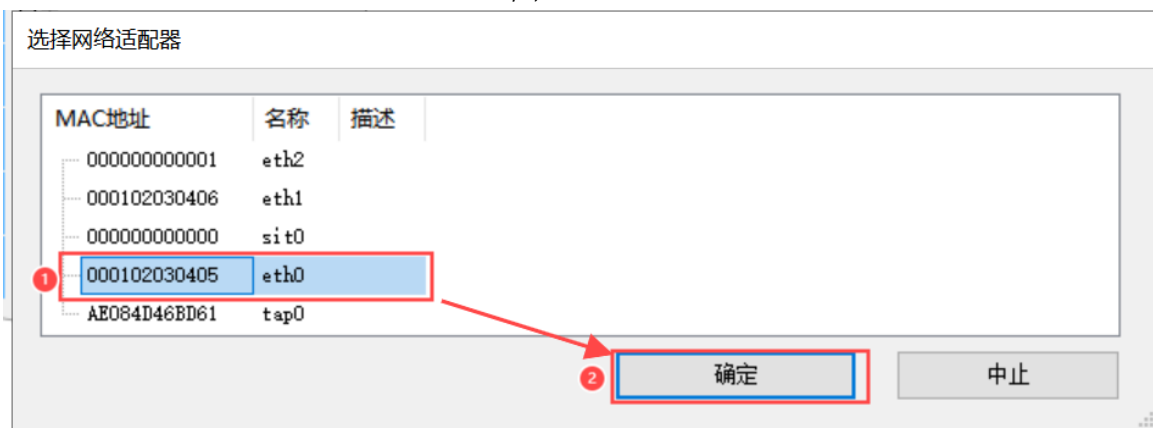


图6.7

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”；



图6.8

复制完成后，在设备栏展示如下：

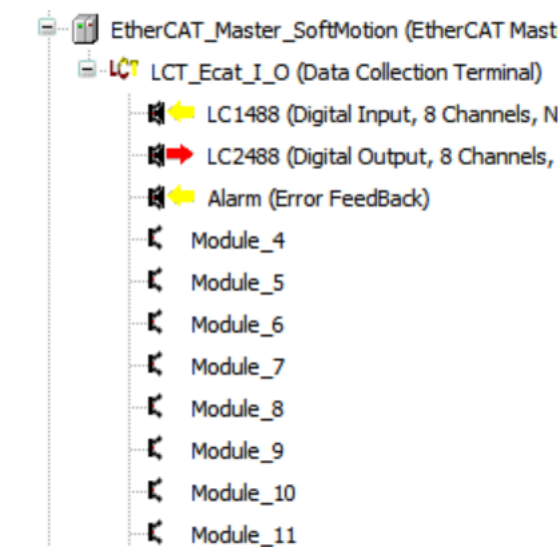


图6.9

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

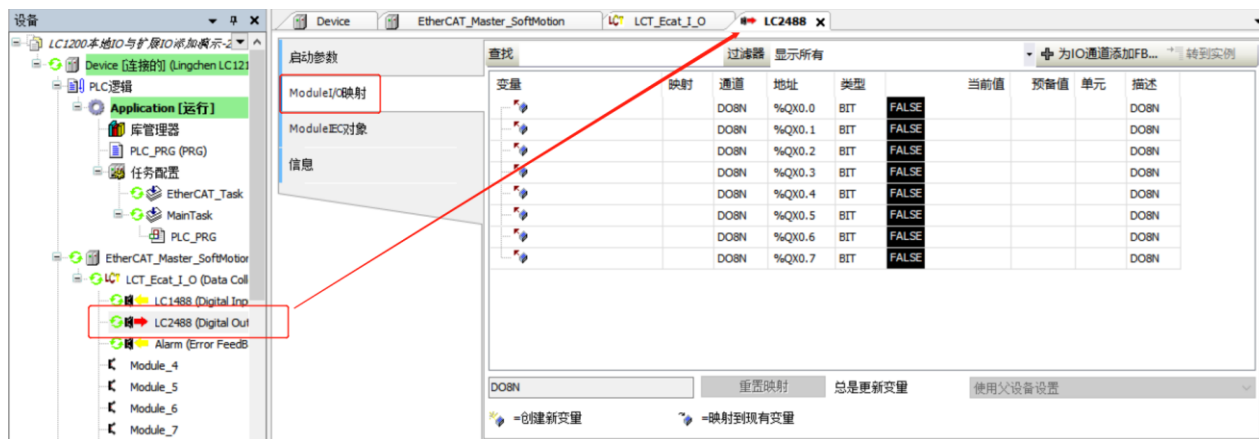


图6.10

7. Modbus

7.1 Modbus TCP Master (LC1000做主站、LC1200做从站)

配合使用备件：LC1200控制器具体操作步骤：

关闭电脑防火墙。

- (1) 新建LC1000的标准工程文件，用做主站；
- (2) 右键“Device” — 添加设备；

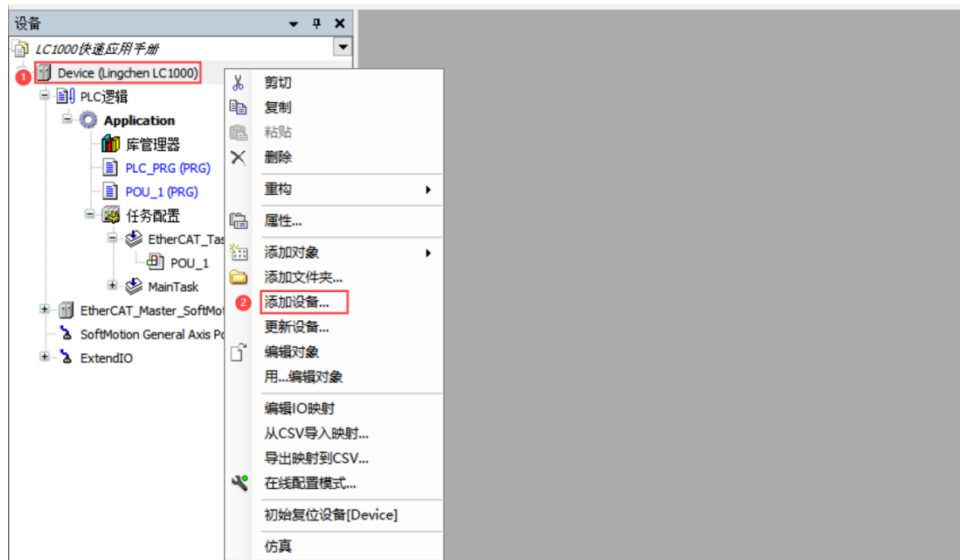


图7.1-1

- (3) 点开“以太网适配器”子选项，选中“Ethernet”，点击“添加设备”；

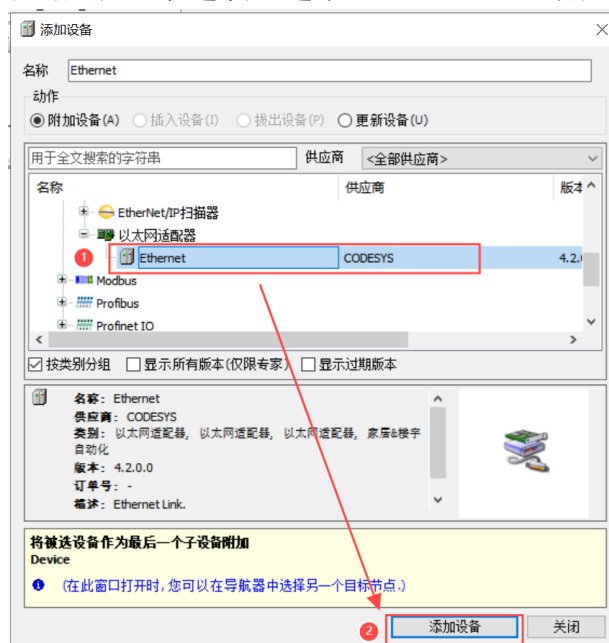


图7.1-2

(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），弹出网口选择界面，选中“eth0”（也可以自定义，当前eth0的IP: 192.168.0.99），点击“确认”。

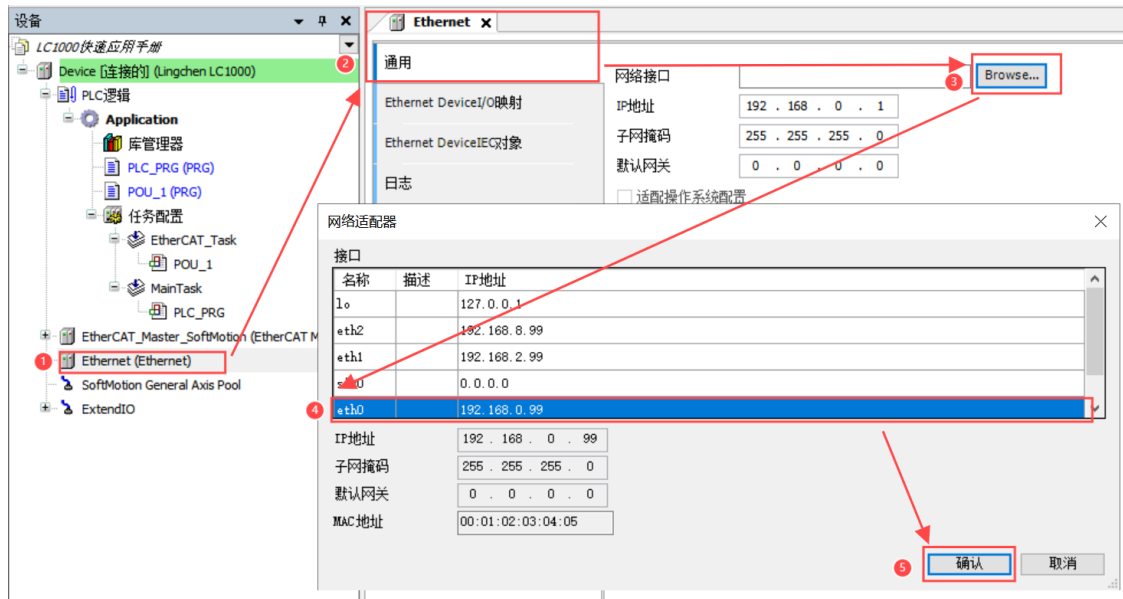


图7.1-3

(5) “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”；

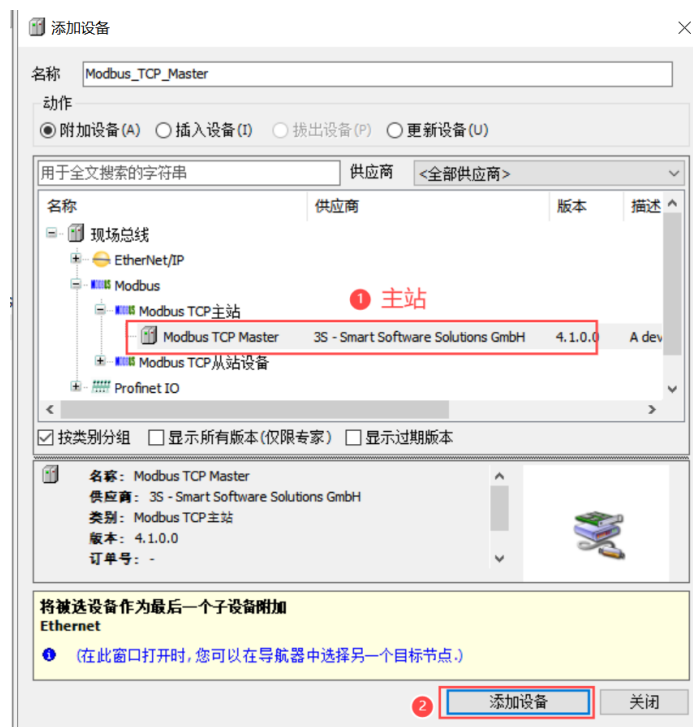


图7.1-4

(6) “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”；

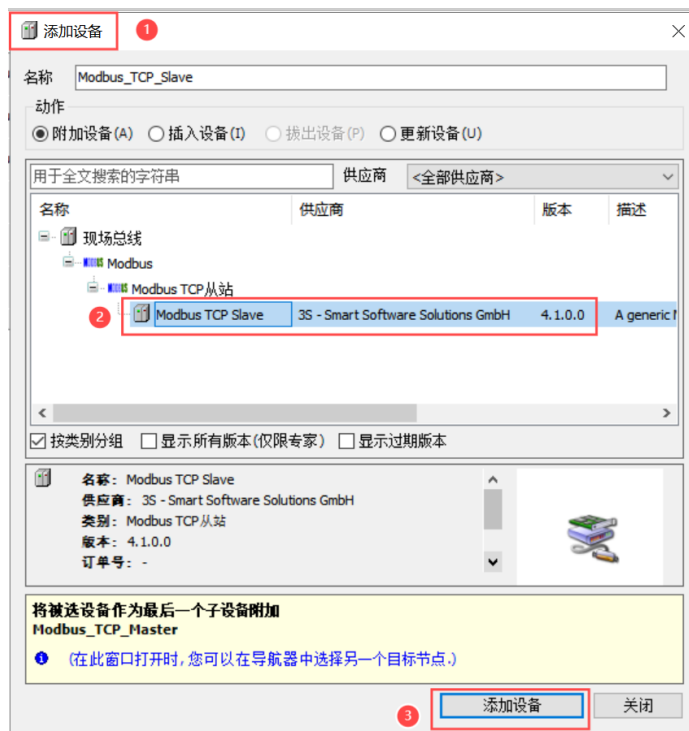


图7.1-5

(7) 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LC1200的eth0的IP，这里设定为192.168.0.89，因此如下图填入），端口号默认：502；



图7.1-6

【注：LC1000主站和LC1200从站用来通讯的网口IP和MAC地址不能一样，如果一样需要自行更改，参考下面10.章】

(8) 在Modbus从站通道中添加测试通道，选择读、写线圈或保持寄存器，设置偏移位置进行测试。（函数代码和读、写寄存器偏移地址与控制器对应地址表见[附录1](#)）

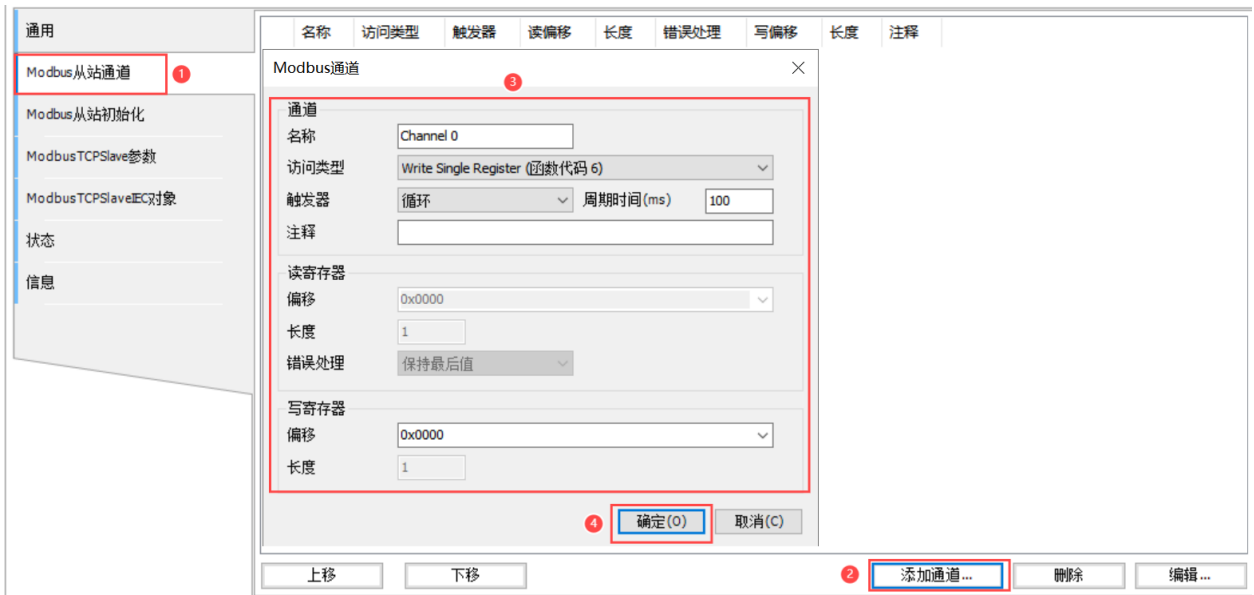


图7.1-7

【注：保持寄存器主站可以读写，输入寄存器只可读不能写】

名称	访问类型	触发器	读偏移	长度	错误处理	写偏移	长度	注释
0 Channel 1	Write Multiple Registers (函数代码 16)	循环, t=100ms				16#0000	2	
1 Channel 2	Write Multiple Coils (函数代码 15)	循环, t=100ms				16#0020	3	

图7.1-8

变量	映射	通道	地址	类型	单元	描述
		Channel 1	%QW9	ARRAY [0..1] OF WORD		Write Multiple Registers
		Channel 1[0]	%QW9	WORD		0x0000
		Channel 1[1]	%QW10	WORD		0x0001
		Channel 2	%QB22	ARRAY [0..0] OF BYTE		Write Multiple Coils

图7.1-9

- (9) 新建LC1200的标准工程文件，用做从站；
- (10) 在LC1200工程中，右键“Device”—添加设备；
- (11) 同主站一样，点开“以太网适配”子选项，选中“Ethernet”，点击“添加设备”；
- (12) 添加完成后，双击打开“Etherent”设备界面，打开“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），弹出网口选择界面，选中“eth0”（也可以自定义，此处网口IP为：192.168.0.89），点击“确认”。

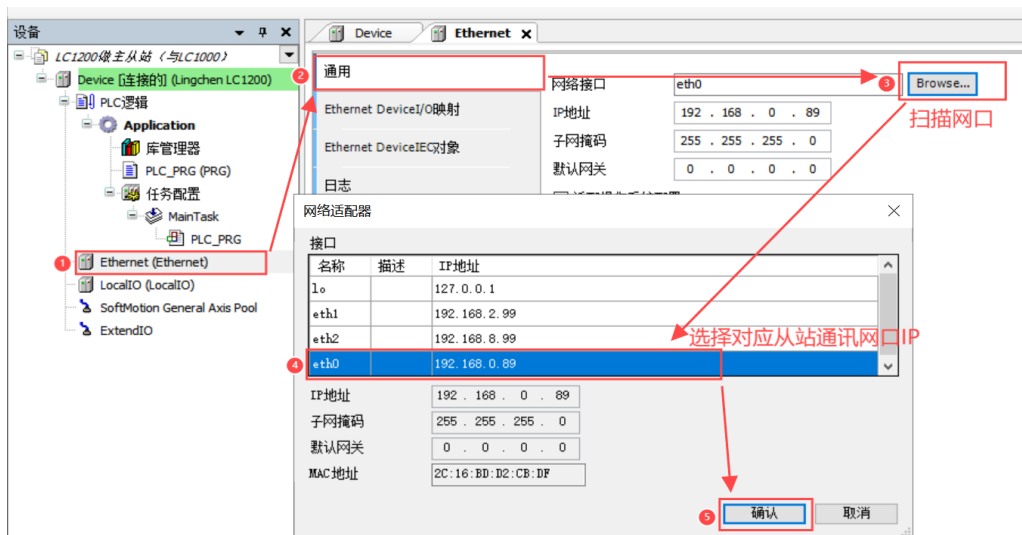


图7.1-10

(13) “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”。

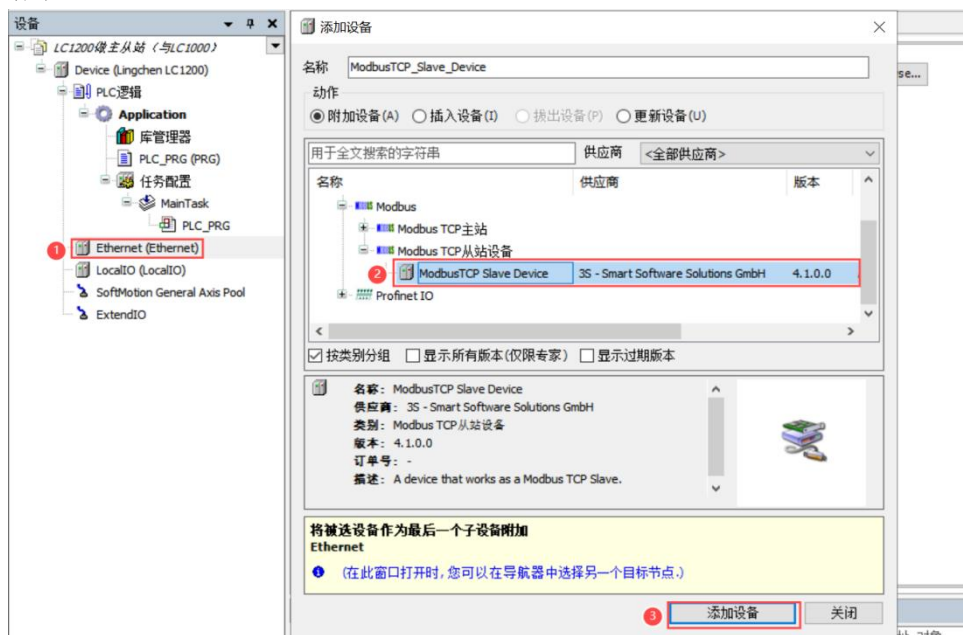


图7.1-11

(14) 双击刚添加的“TCP slave”从站，打开“通用”页面，配置从站的端口号、寄存器个数与起始地址等参数，注意端口号应与主站的下方从站配置相同。



图7.1-12

变量	映射	通道	地址	类型	单元	描述
		保持寄存器	%IW0	ARRAY [0..1] OF WORD		
		保持寄存器[0]	%IW0	WORD		
		保持寄存器[1]	%IW1	WORD		
		输入寄存器	%QW0	ARRAY [0..1] OF WORD		
		线圈	%IB4	ARRAY [0..0] OF BYTE		

图7.1-13

【注：从站的起始地址要和主站下面从站中变量通道的偏移地址相对应】

(15) 分别打开主从站程序，“登录—启动”，登录后在主站的从站通道中给保持寄存器赋值，从站相应地址变量值发生同样变化。



图7.1-14

变量	映射	通道	地址	类型	当前值
		保持寄存器	%IW0	ARRAY [0..1] OF WORD	Only subelements updated
		保持寄存器[0]	%IW0	WORD	5
		保持寄存器[1]	%IW1	WORD	10
		输入寄存器	%QW0	ARRAY [0..1] OF WORD	Only subelements updated
		线圈	%IB4	ARRAY [0..0] OF BYTE	Only subelements updated
		线圈[0]	%IB4	BYTE	5
		Bit0	%IX4.0	BOOL	TRUE
		Bit1	%IX4.1	BOOL	FALSE
		Bit2	%IX4.2	BOOL	TRUE

从站通道相应的输入变量随主站变化

图7.1-15

7.2 Modbus TCP Slave (LC1000做从站、LC1200做主站)

LC1000的Modbus TCP通讯带有两种从站，一种是Codesys自带的组件从站（“ModbusTCP_Slave_Device”），一种是凌臣自研的组件从站，本地从站（LCTModbusTCPSlave）。

7.2.1 凌臣自研组件从站 (LCTModbusTCPSlave)

Codesys自带从站与LC1200主站做Modbus TCP Slave通讯，具体操作步骤：

- (1) 新建LC1000的标准工程文件。
- (2) 右键“Device”—添加设备。
- (3) 在弹窗中选择本地从站设备“LCTModbusTCPSlave”—点击“添加设备”。

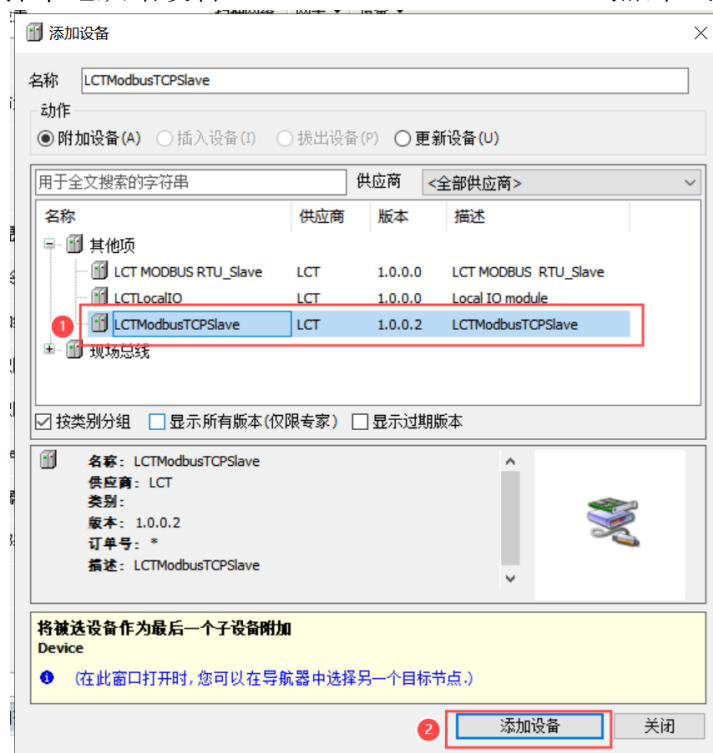


图7.2.1-1

(4) 双击刚添加的设备“LCTModbusTCPSlave”—打开“TCP_Slave参数”页面—配置参数，打开从站—选择要通讯的网口（这里用eth0网口：192.168.0.99）即可。



图7.2.1-2

【注：网口地址可以自行更改，参照下面10.章节】

(5) 打开一个已调用到任务配置的POU，声明两个WORD型变量（16位），并通过绝对地址用来做从站映射。【注：绝对地址要与1200从站偏移地址相对应】

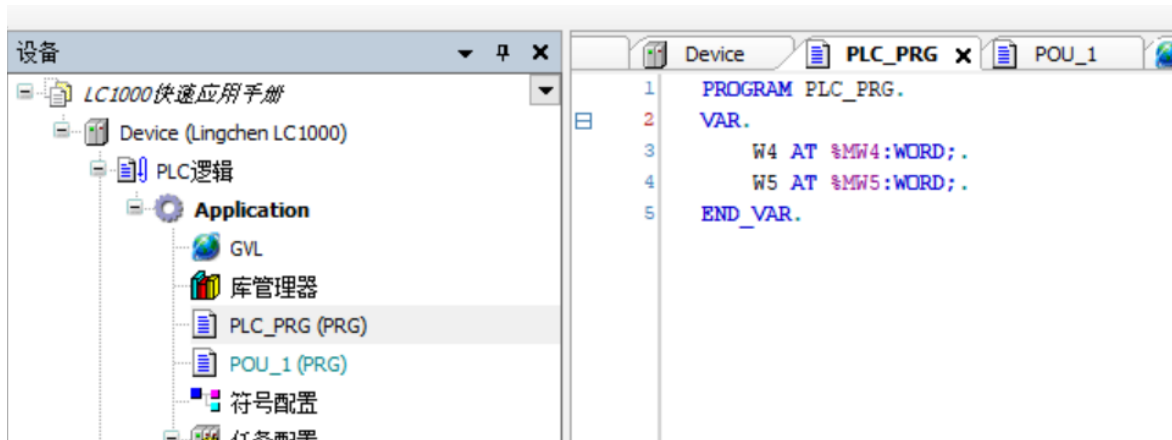


图7.2.1-3

- (6) 新建LC1200的标准工程文件，用做主站。
- (7) 打开LC1200主站程序，右键“Device”—添加设备。
- (8) 点开“以太网适配器”子选项，选中“Ethernet”，点击“添加设备”。
- (9) 双击刚添加的“Ethernet”，在“通用”子界面，点击“Browse...”选择网络接口，弹出网口选择页面，选中“eth0”（此处IP为：192.168.0.89），点击“确认”。
- (10) “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”。

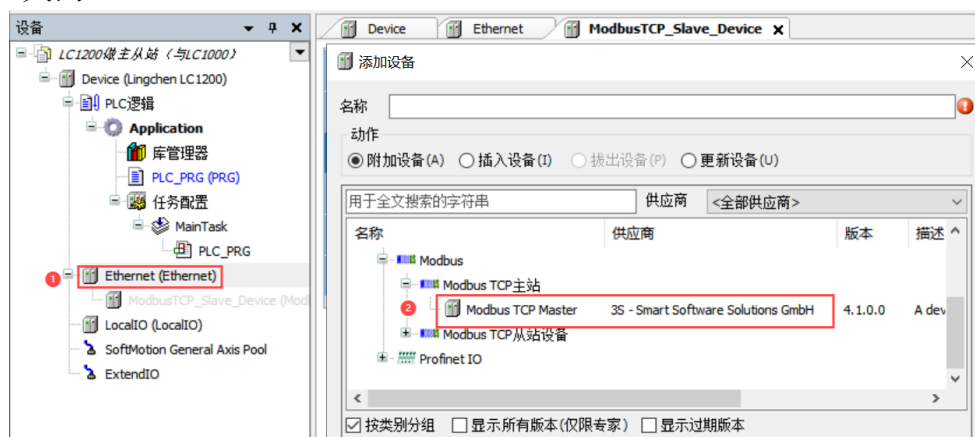


图7.2.1-4

- (11) “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”。

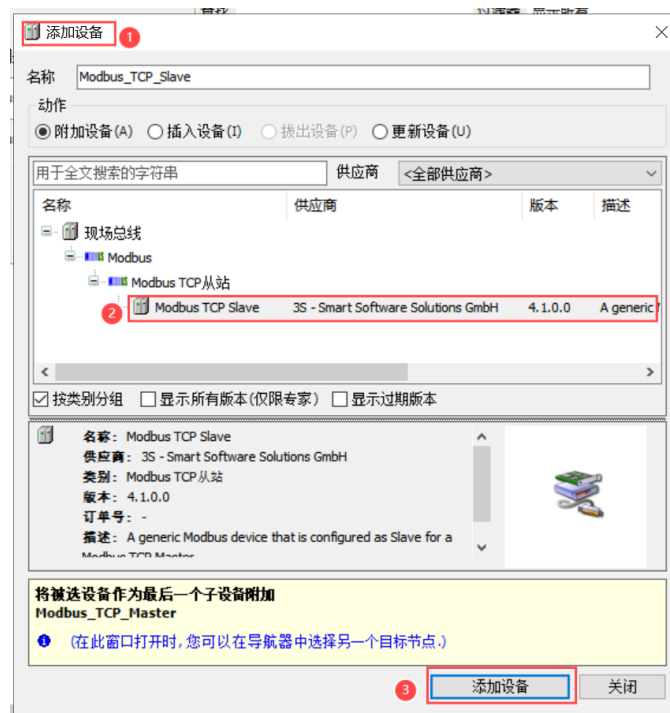


图7.2.1-5

(12) 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LC1000的eth0的IP，这里设定为192.168.0.99，因此如下图填入），端口号默认：502。

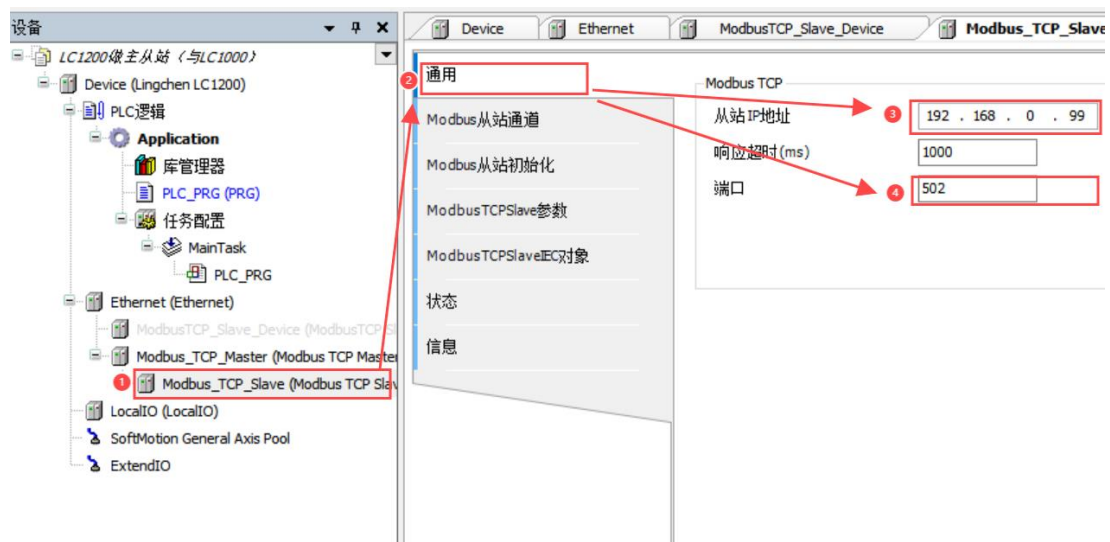


图7.2.1-6

【注：LC1000从站和LC1200主站用来通讯的网口IP和MAC地址不能一样，如果一样需要自行更改，参考下面10.章】

(13) 在Modbus从站通道中添加测试通道，选择读、写线圈或保持寄存器，设置偏移位置进行测试。（函数代码和读、写寄存器偏移地址与控制器对应地址表见附录1）

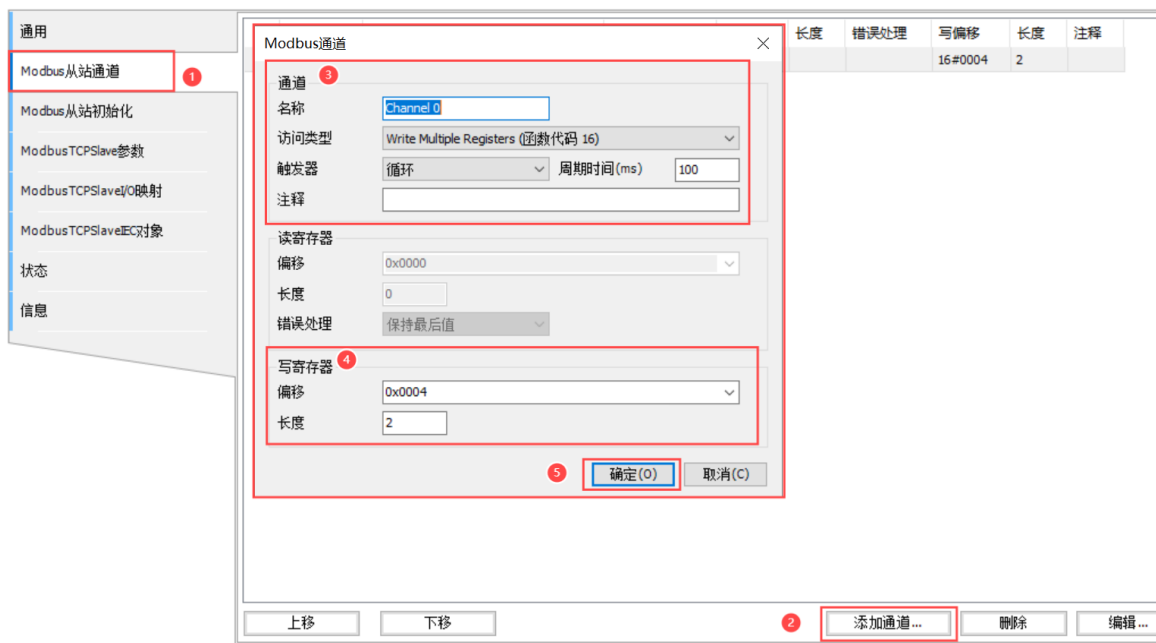


图7.2.1-7

【注：保持寄存器主站可以读写，输入寄存器只可读不能写且从站POU中变量的绝对地址要和主站下面从站中变量通道的偏移地址相对应】



图7.2.1-8



图7.2.1-9

(13) 分别打开主从站程序，“登录—启动”，登录后在主站的从站通道中给保持寄存器赋值，从站POU中相应地址变量值发生同样变化。

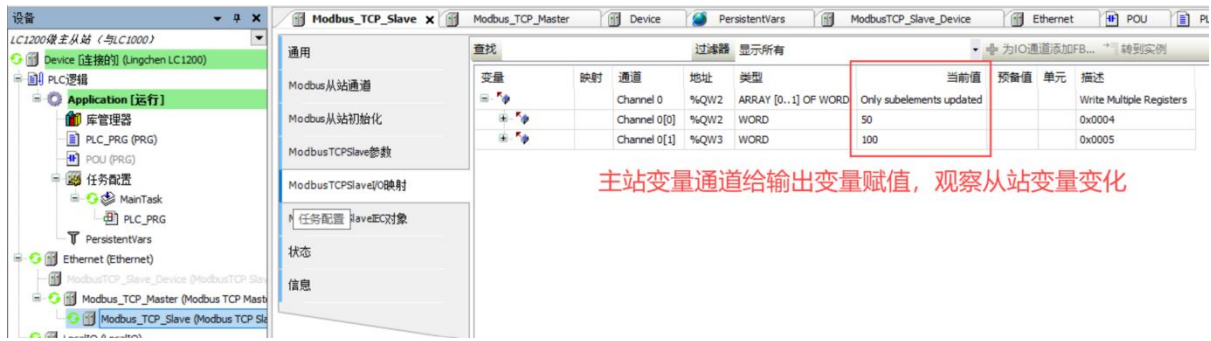


图7.2.1-10

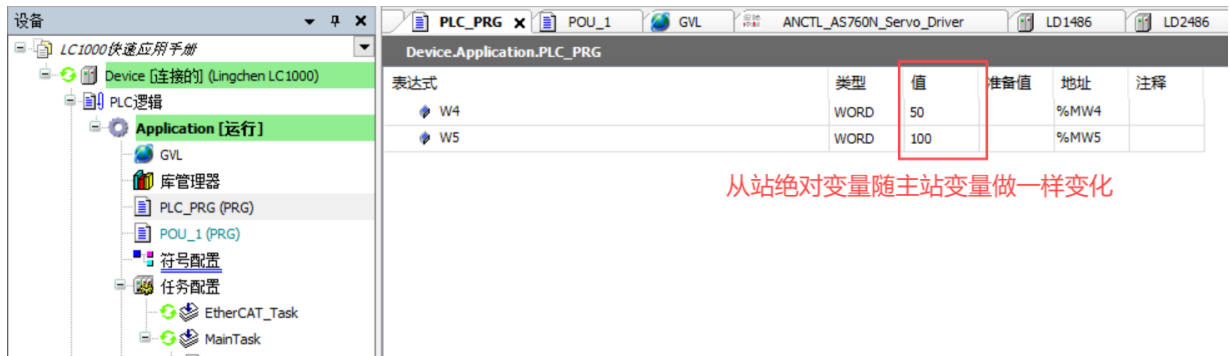


图7.2.1-1

7.2.2 Codesys自带组件从站（ModbusTCP_Slave_Device）

Codesys自带从站与LC1200主站做Modbus TCP Slave通讯，具体操作步骤：

- (1) 新建LC1000的标准工程文件。
- (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连接设备Device）。
- (5) 选中“eth0”（也可以自定义，此处IP为：192.168.0.99），点击确认。
- (6) “右键Etherent—添加设备—按照图中所示位置，找到Modbus TCP Slave选中—添加—关闭”。

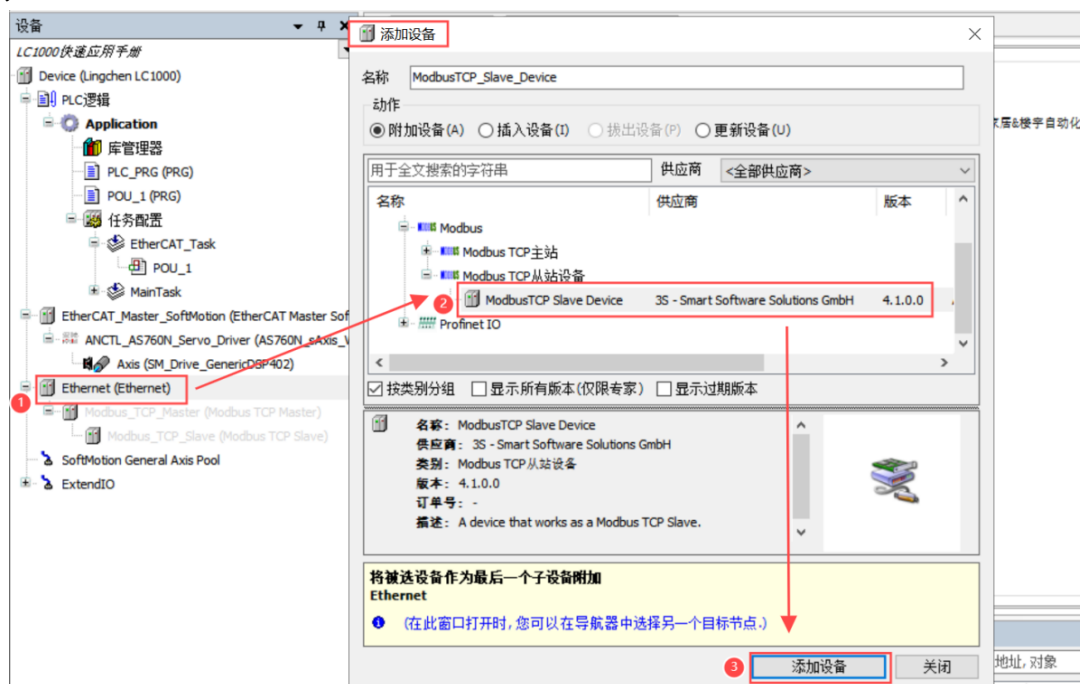


图7.2.2-1

- (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。

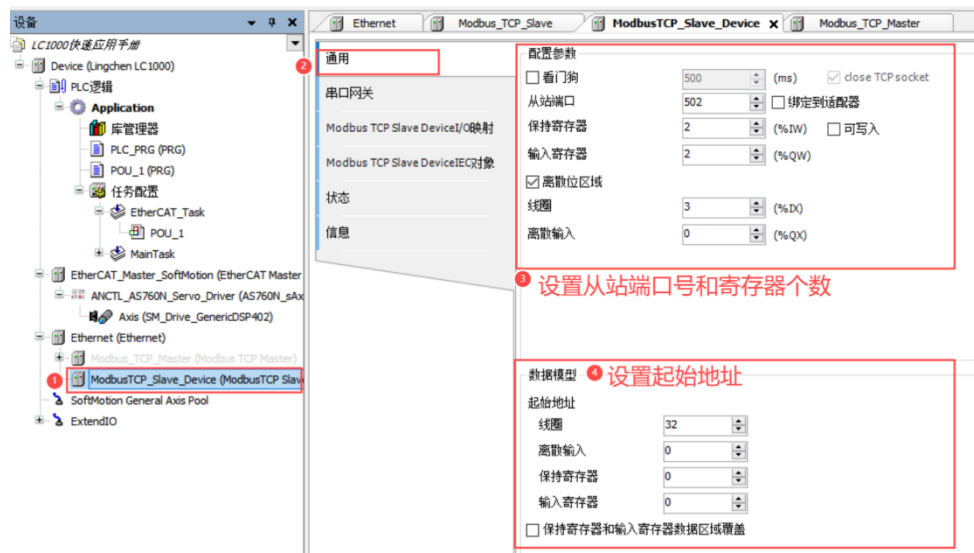


图7.2.2-2



图7.2.2-3

- (8) 新建LC1200的标准工程文件，用做主站。
- (9) 打开LC1200主站程序，右键“Device”—添加设备。
- (10) 点开“以太网适配器”子选项，选中“Ethernet”，点击“添加设备”。
- (11) 双击刚添加的“Ethernet”，在“通用”子界面，点击“Browse...”选择网络接口，弹出网口选择页面，选中“eth0”（此处IP为：192.168.0.89），点击“确认”。
- (12) “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”。

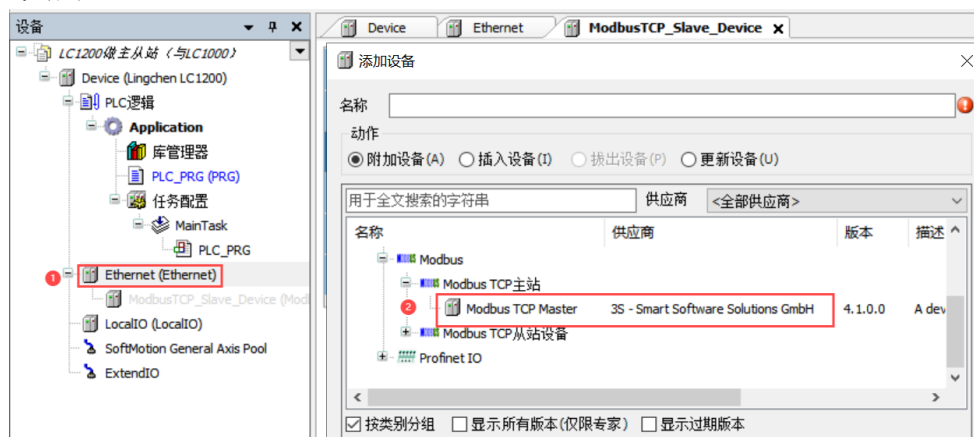


图7.2.2-4

(13) “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”。

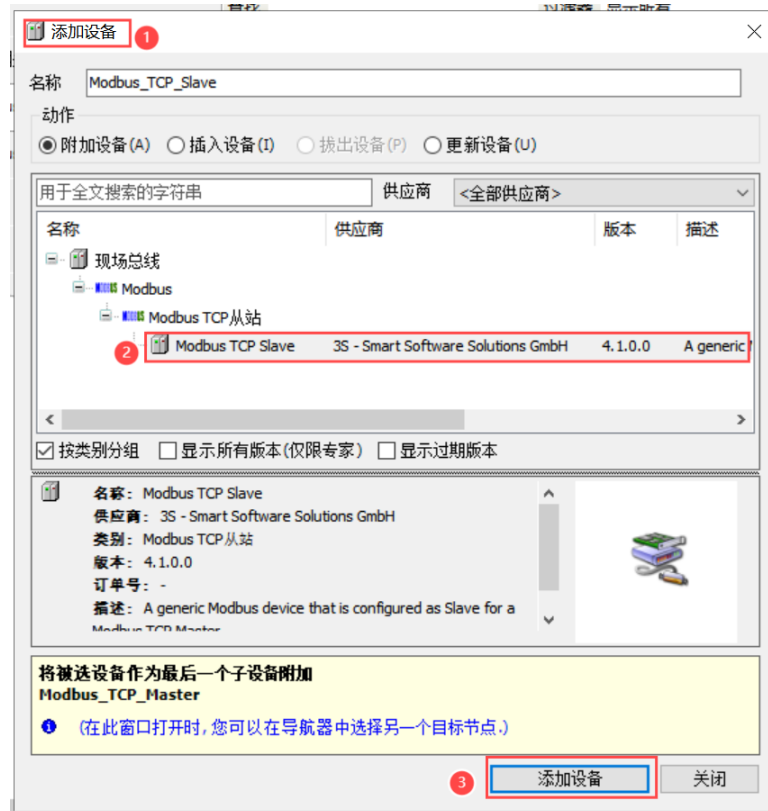


图7.2.2-5

(14) 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LC1000的eth0的IP，这里设定为192.168.0.99，因此如下图填入），端口号默认：502。

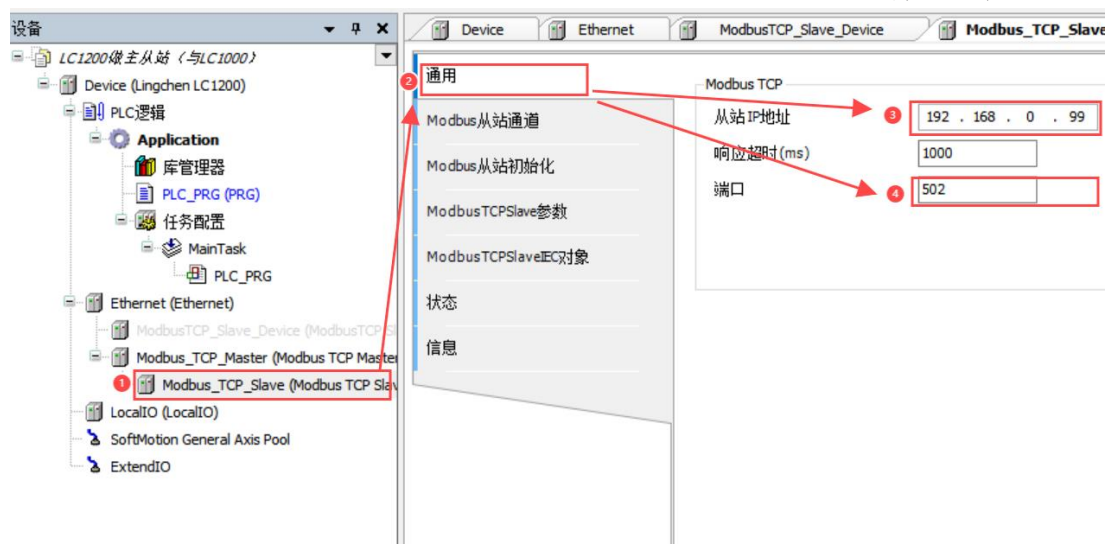


图7.2.2-6

【注：LC1000从站和LC1200主站用来通讯的网口IP和MAC地址不能一样，如果一样需要自行更改，参考下面10.章】

(15) 在Modbus从站通道中添加测试通道，选择读、写线圈或保持寄存器，设置偏移位置进行测试。（函数代码和读、写寄存器偏移地址与控制器对应地址表见附录1）。

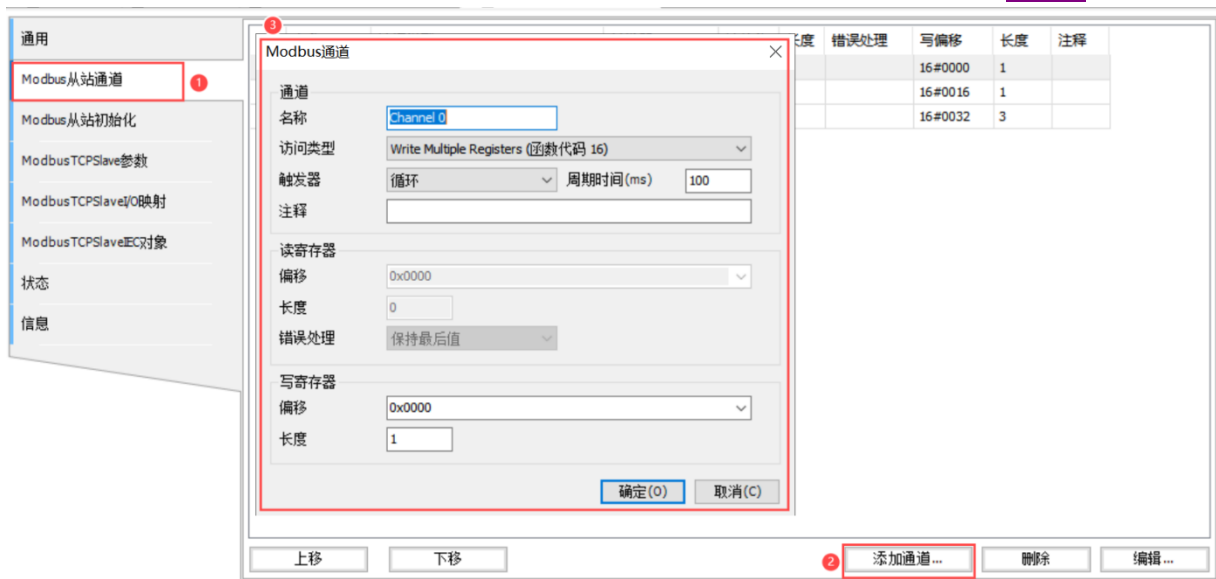


图7.2.2-7

【注：保持寄存器主站可以读写，输入寄存器只可读不能写且从站的起始地址要和主站下面从站中变量通道的偏移地址相对应】

图7.2.2-8

名称	访问类型	触发器	读偏移	长度	错误处理	写偏移	长度	注释
0 Channel 0	Write Multiple Registers (函数代码 16)	循环, t#100ms				16#0000	2	
1 Channel 2	Write Multiple Coils (函数代码 15)	循环, t#100ms				16#0020	3	

图7.2.2-9

变量	映射	通道	地址	类型	单元	描述
		Channel 0	%QW2	ARRAY [0..1] OF WORD		Write Multiple Registers
		Channel 0[0]	%QW2	WORD		0x0000
		Channel 0[1]	%QW3	WORD		0x0001
		Channel 2	%QB8	ARRAY [0..0] OF BYTE		Write Multiple Coils
		Channel 2[0]	%QB8	BYTE		Write Multiple Coils
		Bit0	%QX8.0	BOOL		0x0020
		Bit1	%QX8.1	BOOL		0x0021
		Bit2	%QX8.2	BOOL		0x0022

图7.2.2-10

(16) 分别打开主从站程序，“登录—启动”，登录后在主站的从站通道中给保持寄存器赋值，从站相应地址变量值发生同样变化。

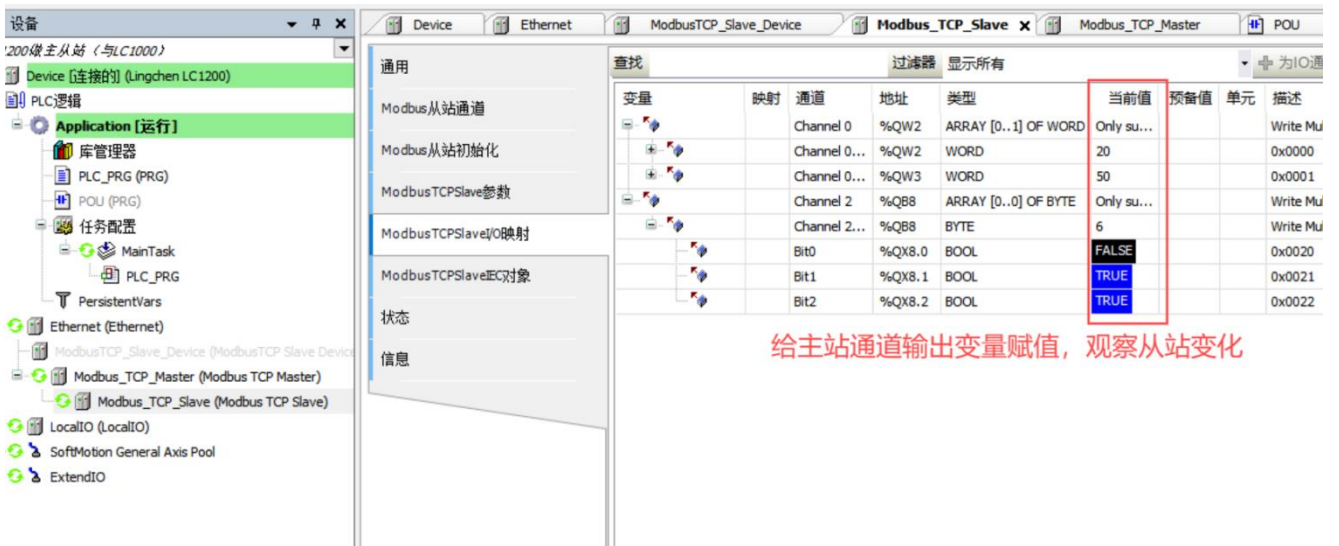


图7.2.2-11

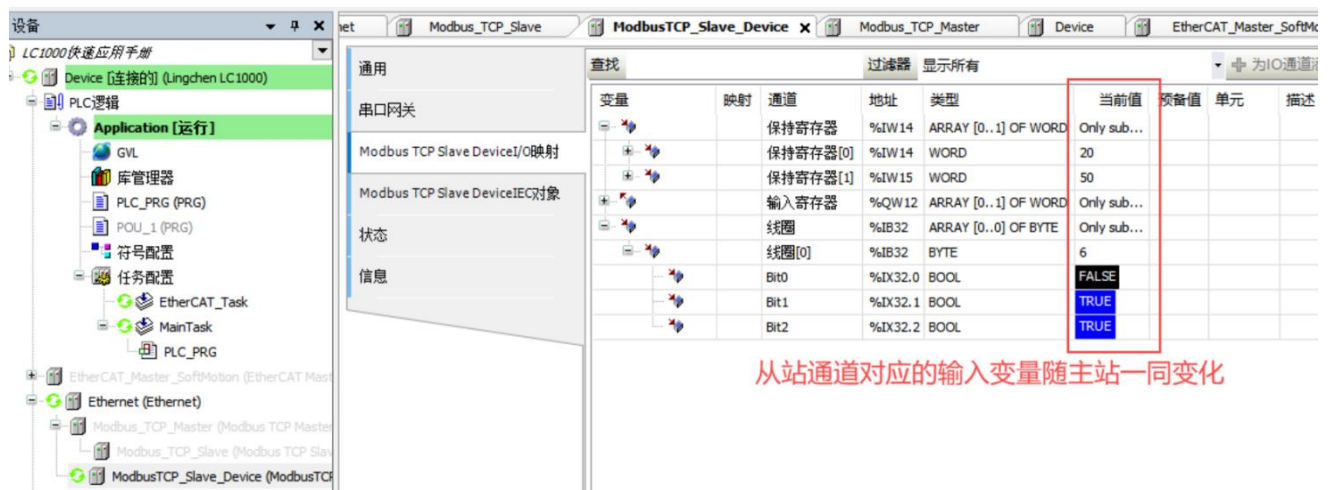


图7.2.2-12

7.3 Modbus RTU (LC1000串口通讯自测)

LC1000控制器带有一组本地串口（1个485和1个232通讯）和一组扩展卡串口（1个485和1个232通讯），扩展卡串口根据客户需求提供。

LC1000可以进行串口通讯自测，即扩展卡串口和本地串口互接，做主从站通讯，接线如下图所示：

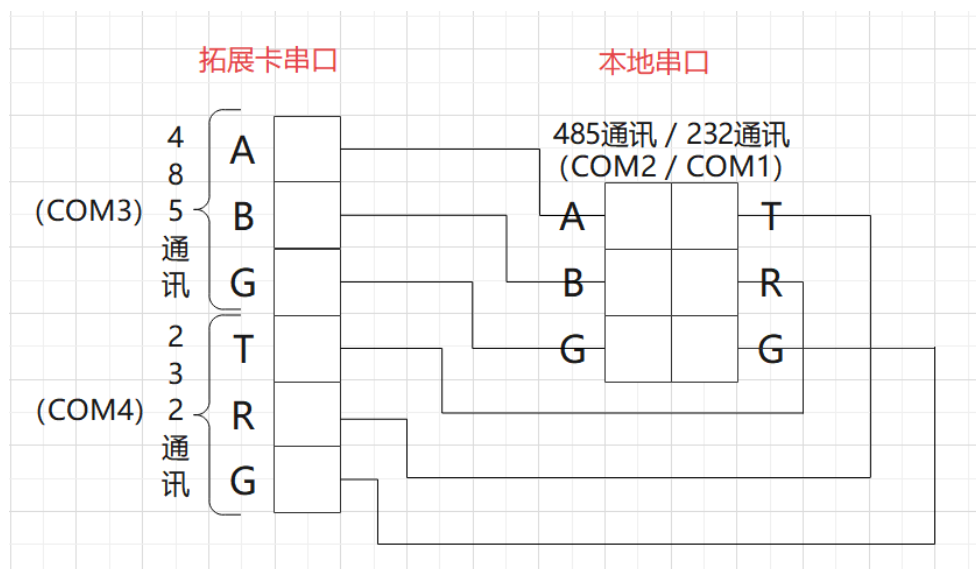


图7.3-1

【注：如上图所示，232通讯自测接线T端和R端要互接】

以485通讯自测为例，本地串口COM2设为主站，扩展卡串口COM3设为从站，操作步骤如下：

测试前关闭电脑防火墙。

(1) 右键“Device” — “添加设备”。

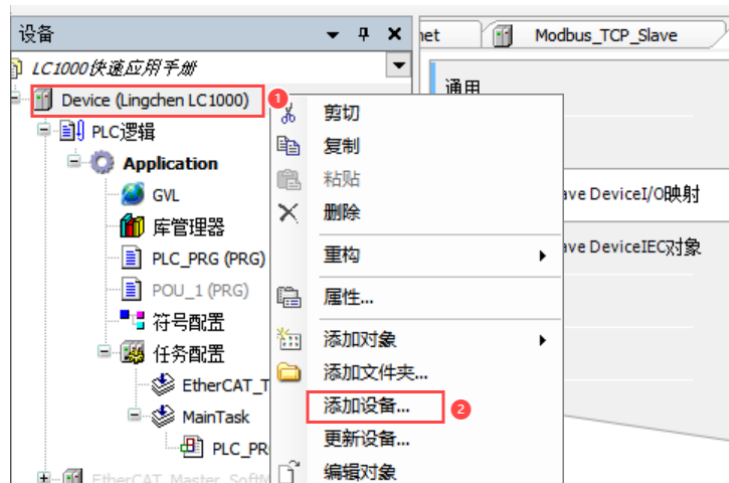


图7.3-2

(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”。

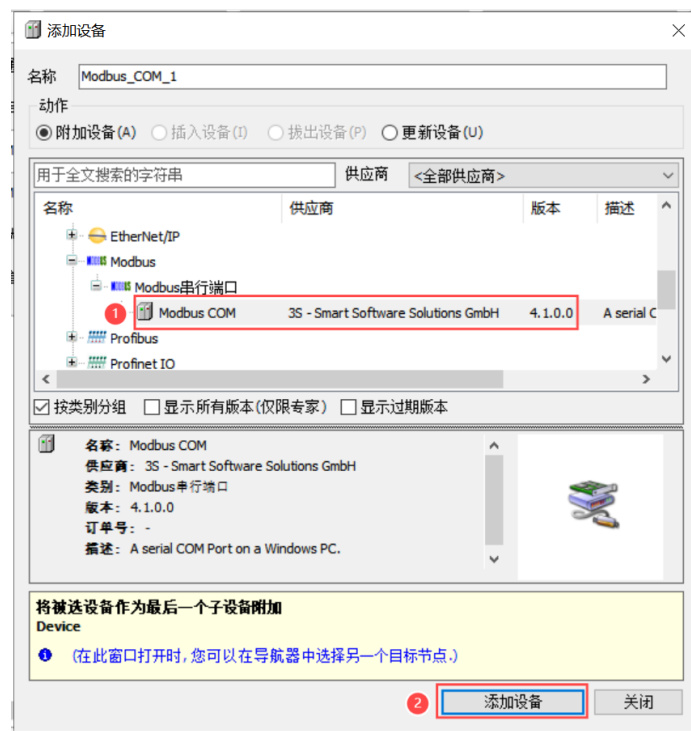


图7.3-3

(3) 接着不要关闭“添加设备”弹窗,直接点击设备栏刚添加的“Modbus_COM”,弹窗自动跳转,点击串行主站,选中添加。



图7.3-4

(4) 接着同样直接点击刚添加的“Modbus_COM”,弹窗自动跳转,选择“Modbus Slave”,选中添加。

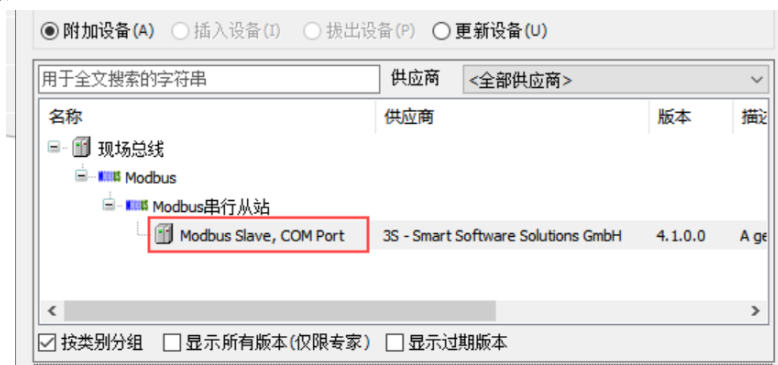


图7.3-5

(5) 添加完成后，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。



图7.3-6

【注：COM端口号配置要和实际硬件接线的端口对应】

(6) 双击“Modbus_Slav_Com_Port”打开页面选择“通用”界面，配置从站地址站号。



图7.3-7

(7) 打开“Modbus从站通道”页面，点击“添加通道”，选择读、写线圈或保持寄存器，设置偏移位置后点击“确定”即可。（函数代码和读、写寄存器偏移地址与控制器对应地址表见附录1）

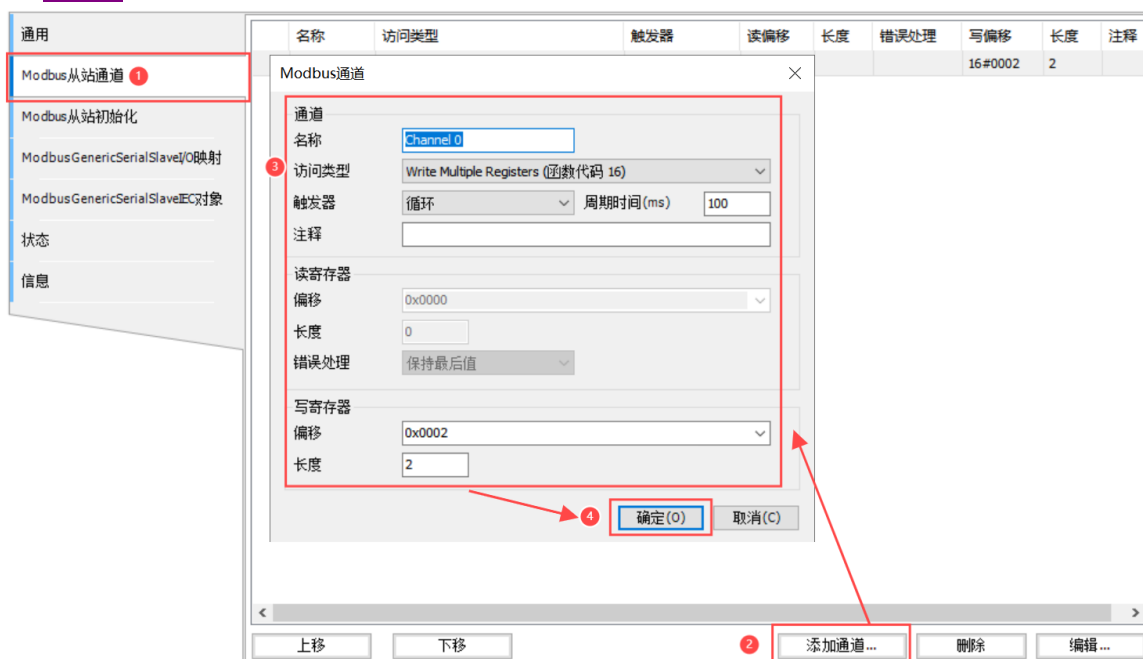


图7.3-8

(8) 添加完成后如图所示，可配合后续从站进行使用。

通用	名称	访问类型	触发器	读偏移	长度	错误处理	写偏移	长度	注释
Modbus从站通道	0 Channel 0	Write Multiple Registers (函数代码 16)	循环, t#100ms				16#0002	2	
Modbus从站初始化									

图7.3-9

通用	变量	映射	通道	地址	类型	单元	描述
Modbus从站通道			Channel 0	%QW16	ARRAY [0..1] OF WORD		Write Multiple Registers
Modbus从站初始化			Channel 0[0]	%QW16	WORD		0x0002
ModbusGenericSerialSlaveI/O映射			Channel 0[1]	%QW17	WORD		0x0003
ModbusGenericSerialSlaveEC对象							

图7.3-10

(9) 在设备栏中，右键“Device” — “添加设备” — 在弹窗打开“其他项” — 选择“LCT MODBUS_Slave”（本地串口从站） — 点击“添加设备”即可。

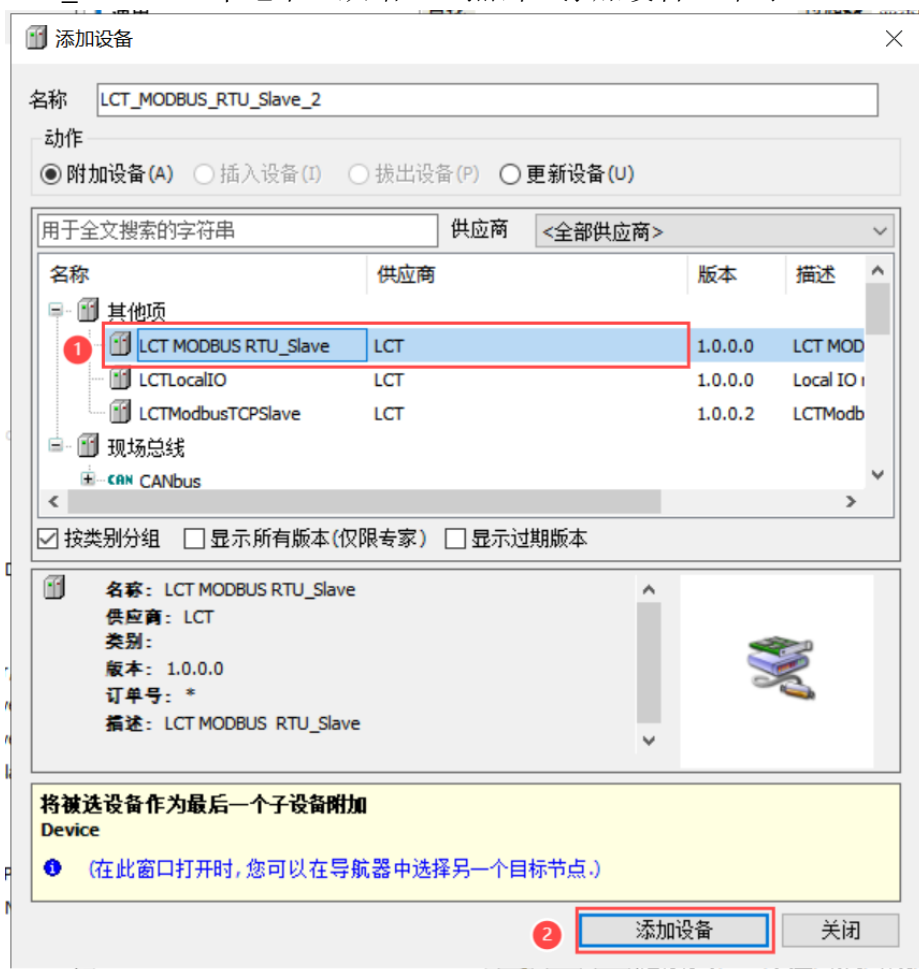


图7.3-11

(10) 双击刚添加的本地串口从站，打开“RTU_Slave参数”页面进行配置，选择“打开从站”；“波特率”和“校验位”要与主站“Modbus_COM”中一致；“站号”和主站下的“从站地址”一样；“Com口”选择实际所接线的串口，这里是COM3。

参数	类型	值	默认值	单元	描述
打开从站	Enumeration of BYTE	打开从站	关闭从站		打开从站
波特率	Enumeration of BYTE	115200	9600		波特率
校验位	Enumeration of BYTE	无校验	ODD		校验位
停止位	Enumeration of BYTE	1	1		停止位
数据位	BYTE	8	8		数据位
站号	BYTE(1..247)	2	1		站号
Com口	Enumeration of BYTE	COM3	COM1		Com口
传输协议	Enumeration of BYTE	RTU	RTU		传输协议
帧间隔	UINT(5..1000)	20	20		帧间隔

图7.3-12

(11) 打开一个已被任务配置调用的POU（这里参考3.4.4），声明两个WORD型变量（与从站通道中添加的寄存器位数一直），并通过绝对地址来映射从站通道。

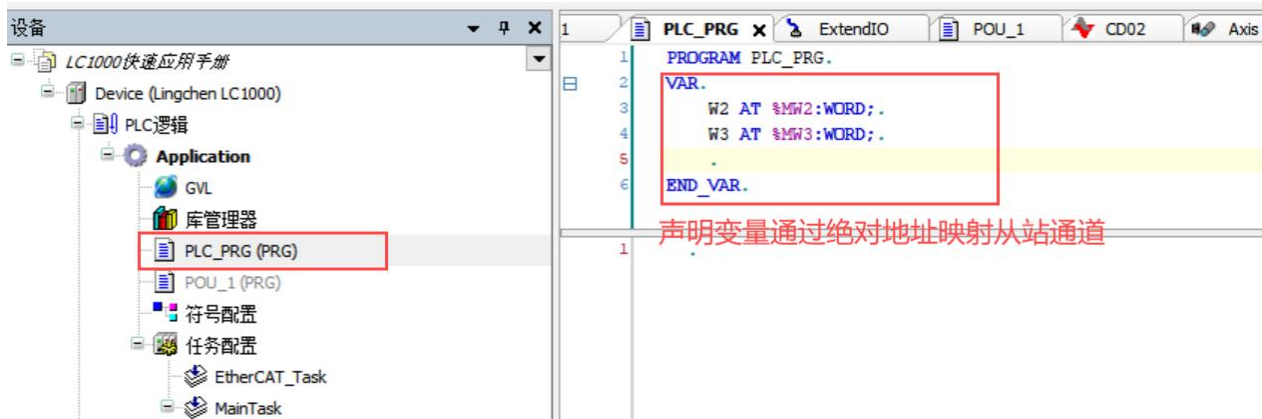


图7.3-13

【注：绝对地址位要与从站通道中寄存器偏移位置对应】

(12) “登录—启动”，给主站的从站通道中输出变量赋值，POU中绝对变量的值会发生同样变化。

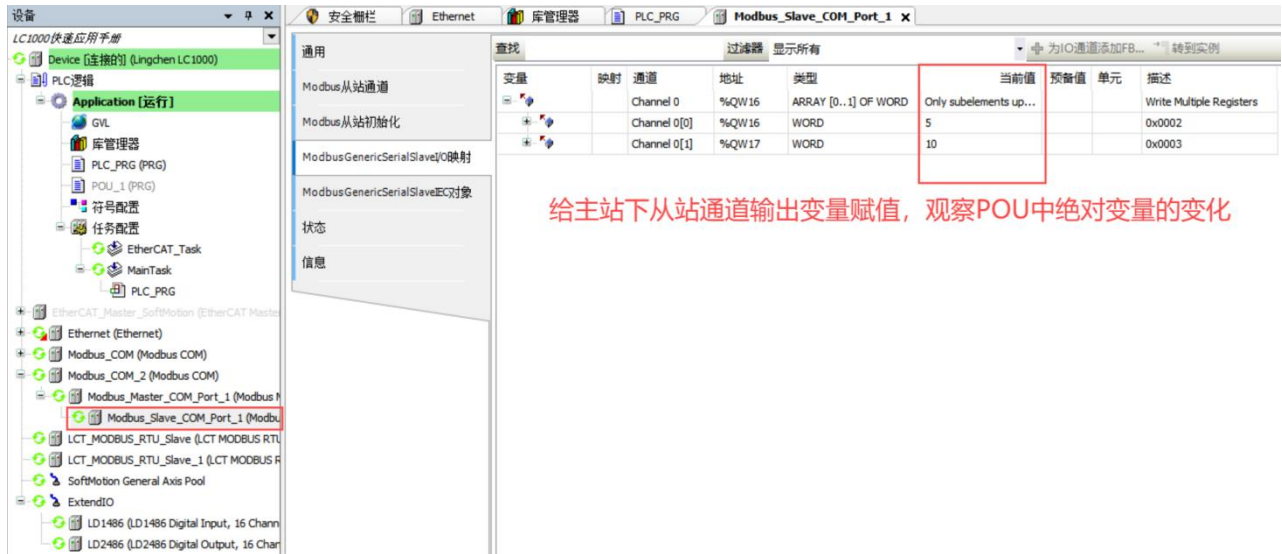


图7.3-14

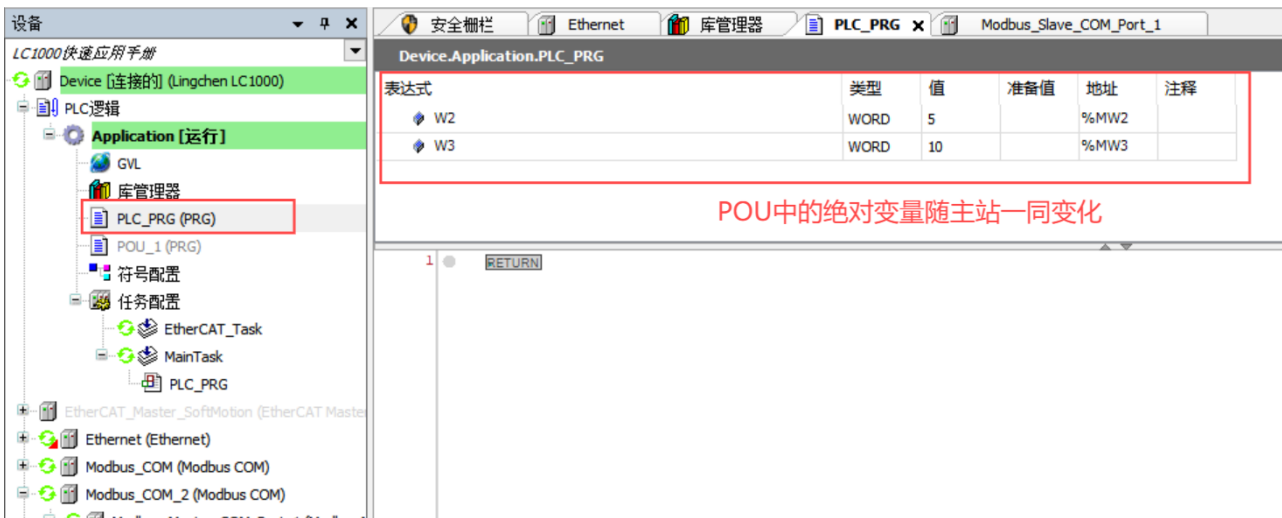


图7.3-15

【注：此时POU中的所对应的绝对变量只能读值，不能写值】

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” — 选择“掉电保持型变量” — 在弹窗中设置名称（这里使用默认名称） — 点击“添加”。

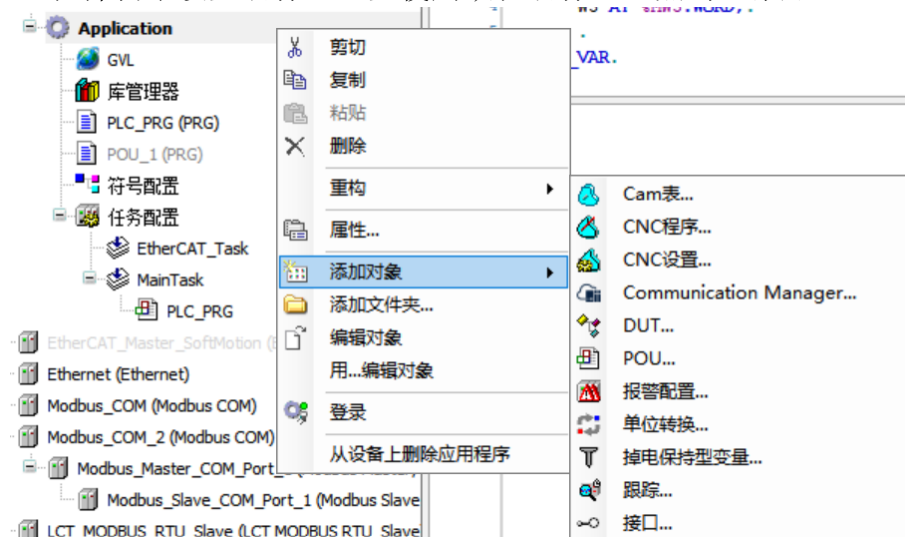


图8.1

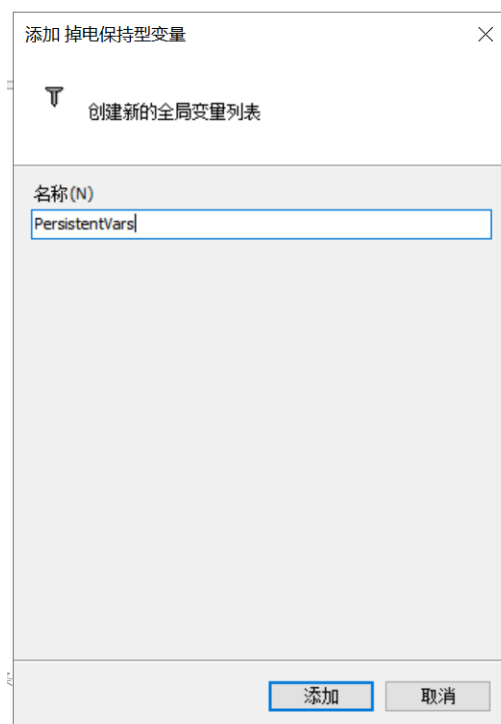


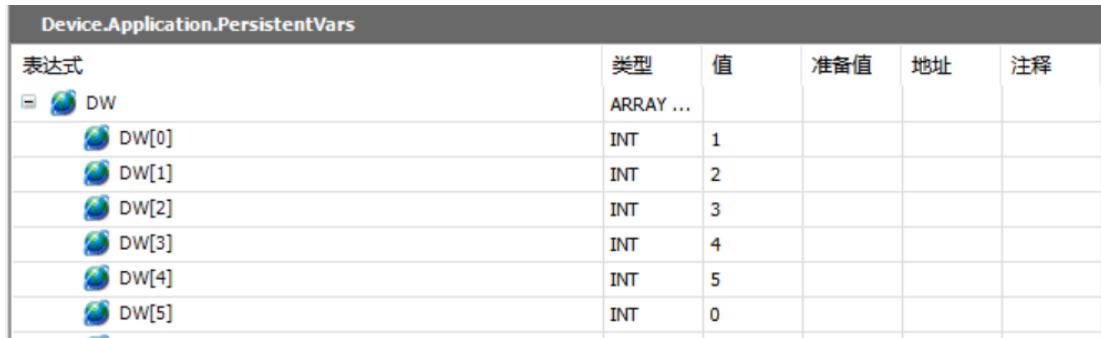
图8.2

(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  DW:ARRAY[0..9]OF INT;.  
4  END_VAR.
```

图8.3

(3) 登录一启动，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
DW	ARRAY ...				
DW[0]	INT	1			
DW[1]	INT	2			
DW[2]	INT	3			
DW[3]	INT	4			
DW[4]	INT	5			
DW[5]	INT	0			

图8.4

(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	地址	注释
DW	ARRAY ...				
DW[0]	INT	1			
DW[1]	INT	2			
DW[2]	INT	3			
DW[3]	INT	4			
DW[4]	INT	5			
DW[5]	INT	0			

图8.5

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...

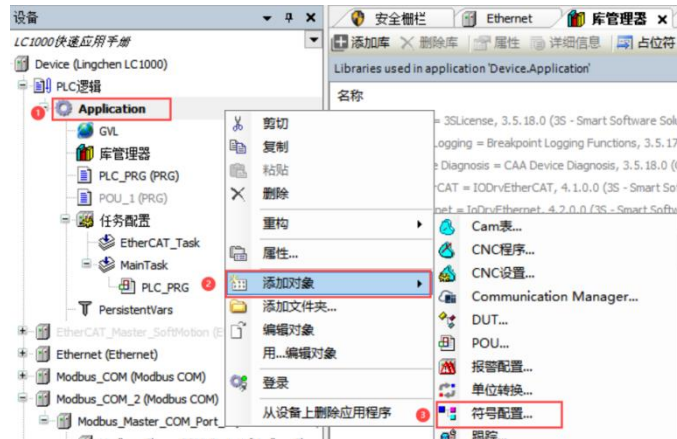


图9.1

(2) 在弹出的“添加符号配置”界面，注意是否勾选支持OPCUA特征，点击“添加”



图9.2

(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”。

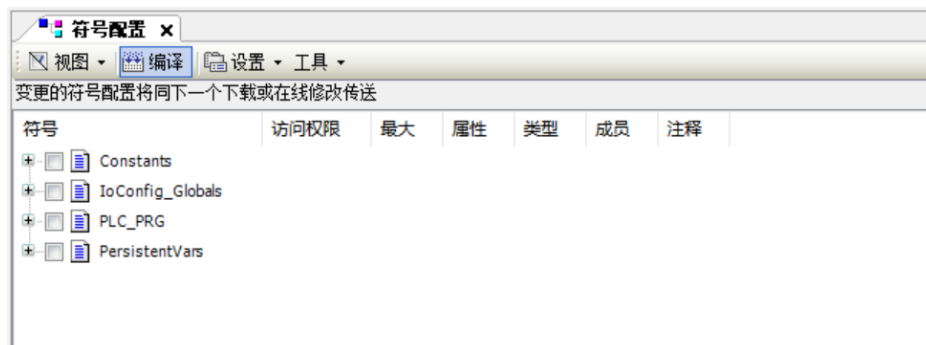


图9.3

(4) 请确认需要的变量所在POU是否已拖入任务配置，分配Task，勾选所需要的变量。

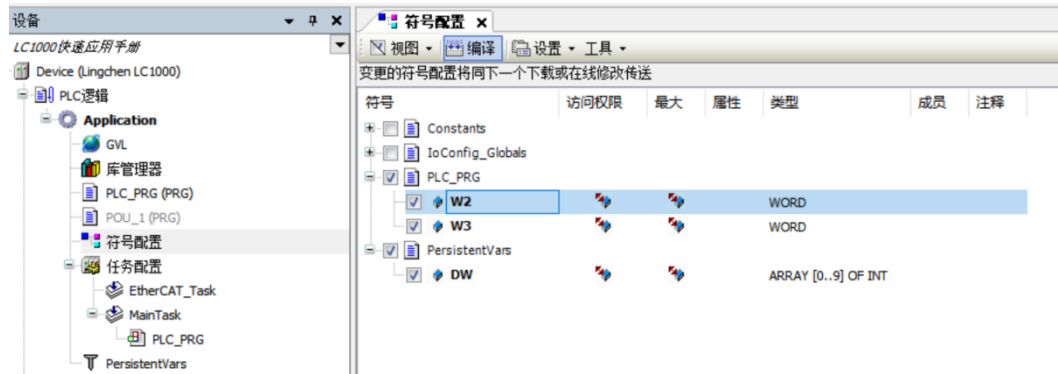


图9.4

(5) 勾选完后，在菜单栏点击“编译——生成代码”。

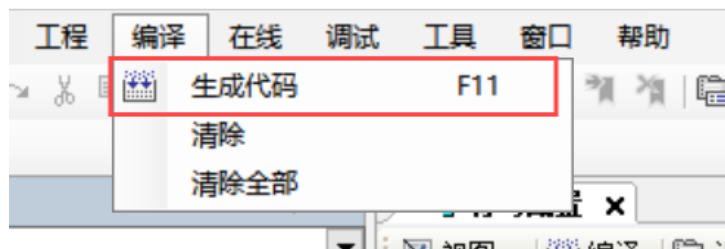


图9.5

(6) 生成代码结束，可以在消息栏看到相关信息。

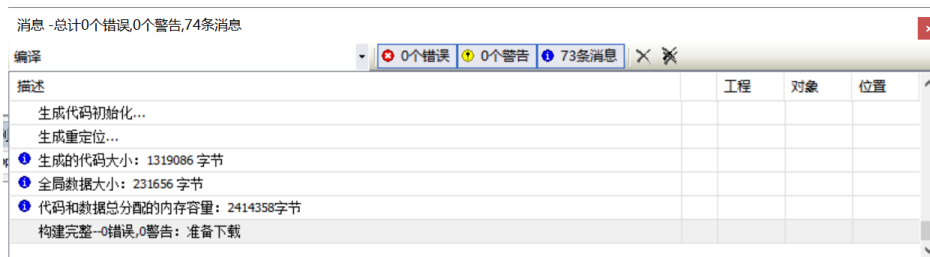


图9.6

(7) 此时，可以在该工程文件的存储目录下，找到生成的改工程应用文件。

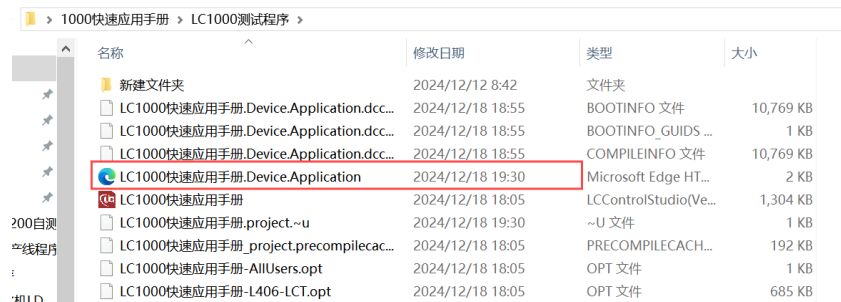


图9.7

(8) 之后使用人机界面的组态程序，扫描标签即可。

10. PLC网口IP与MAC地址的设置

凌臣系列PLC（包含LC1000系列）的各个网口IP与MAC地址是可以在软件程序中读取和修改的，操作方法如下：

一、网口IP的获取和修改

(1) 双击“Device”打开页面一点击“扫描网络”进行设备扫描—选择扫描出的设备—点击“确定”。

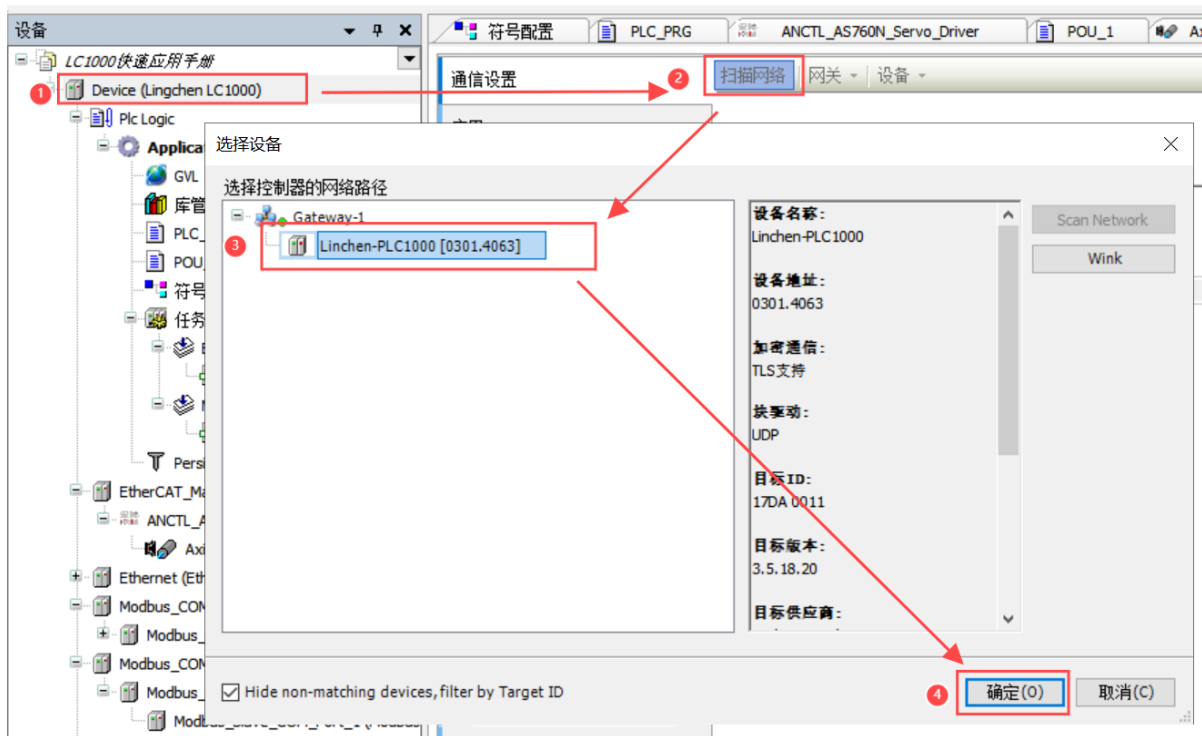


图10.1

(2) 点击打开“PLC指令”页面—单击下图中省略号方框—在“插入命令”弹窗中选择“GetIp”（获取IP指令）—点击“插入”即可。

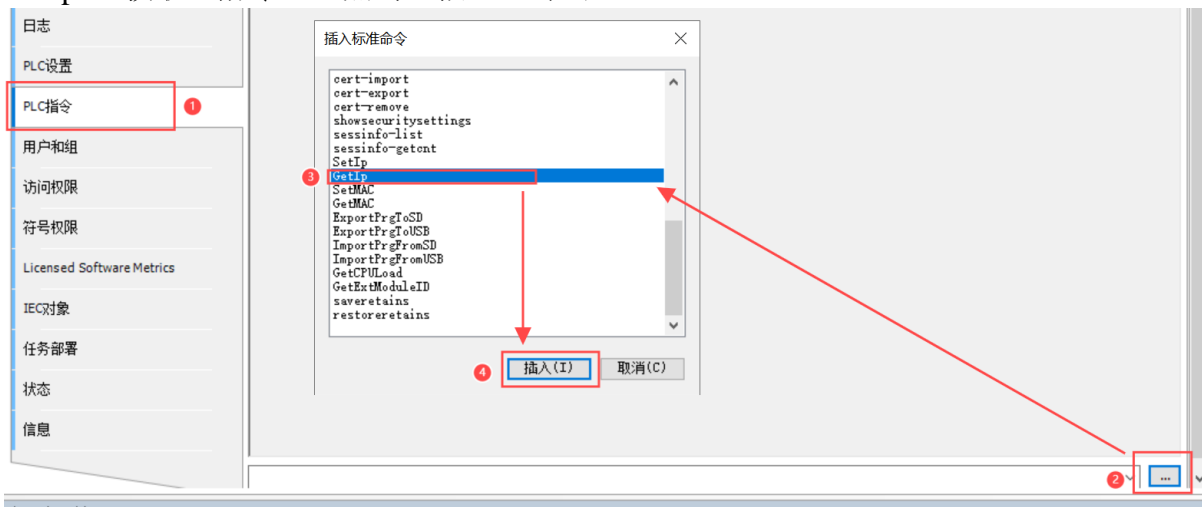


图10.2

(3) 插入的指令会显示在指令输入栏里，再按一下空格—输入需要读取的网口（这里输入eth0）—按键盘上回车“Enter”键即可读出此网口目前的IP地址（192.168.0.99）。

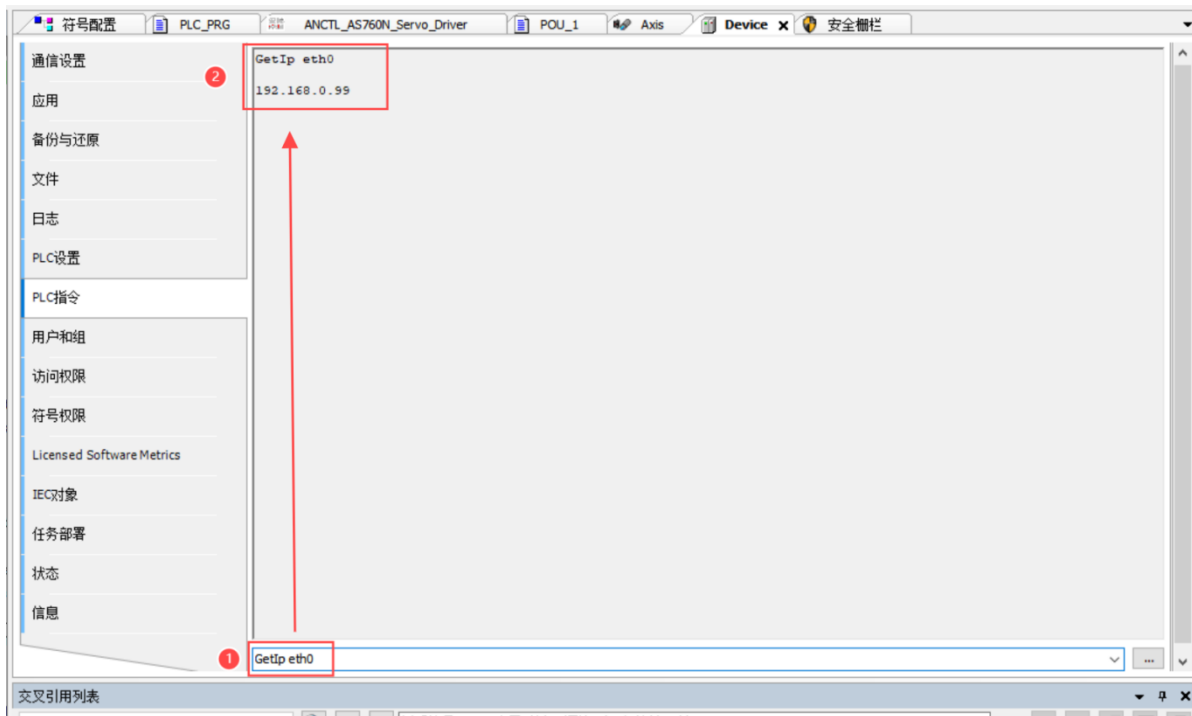


图10.3

(4) 继续点击省略号方框—在弹窗中选择“SetIp”（设置IP指令）—点击“插入”即可。

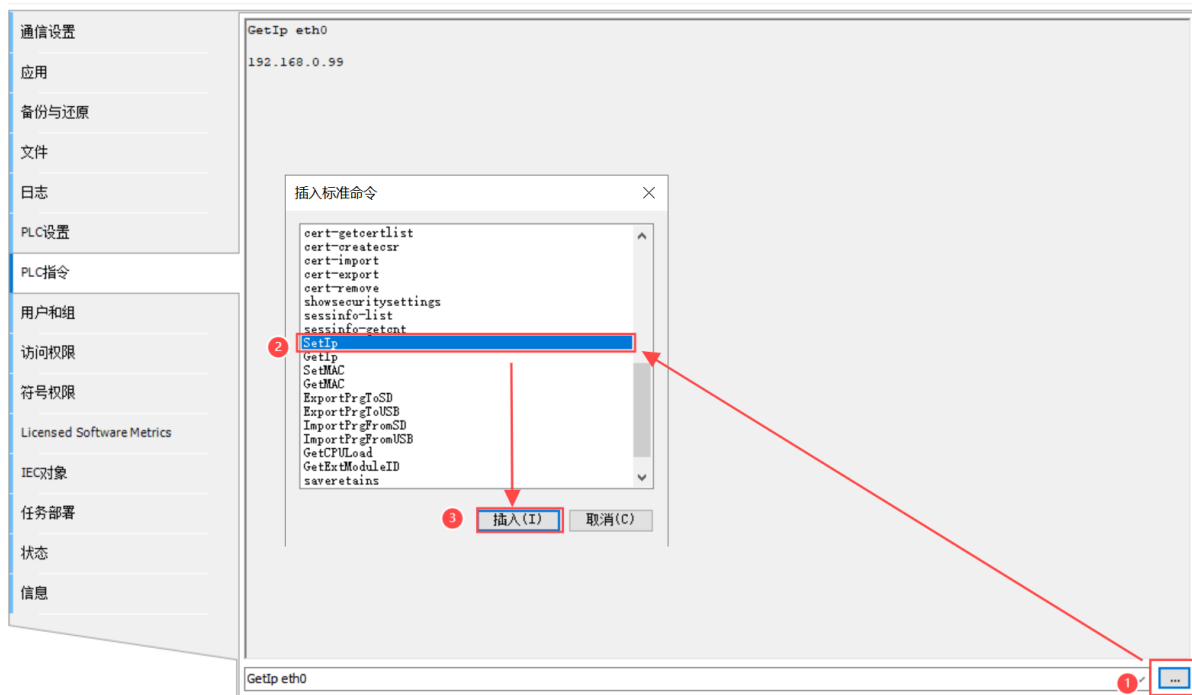


图10.4

(5) 之后“GetIp”指令会显示在输入栏里一按空格输入IP端口号（eth0）一再按空格输入需要更改成的IP地址（192.168.0.98）一按“Enter”键即可。

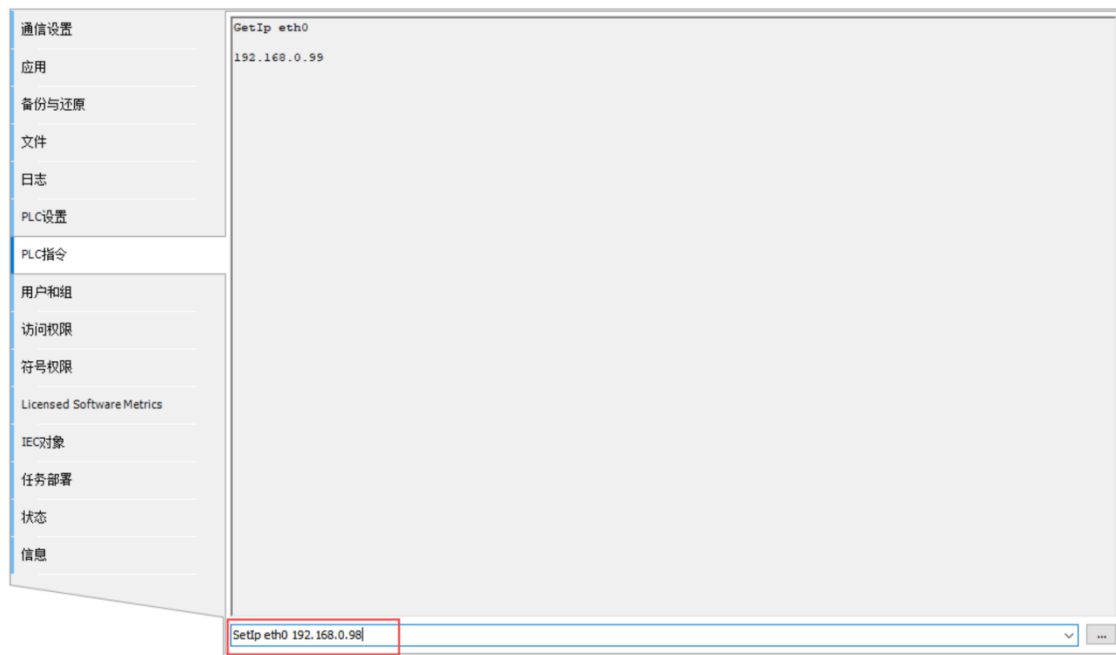


图10.5

(6) IP若更改成功，会跳出弹窗提示重新扫描连接网络设备。

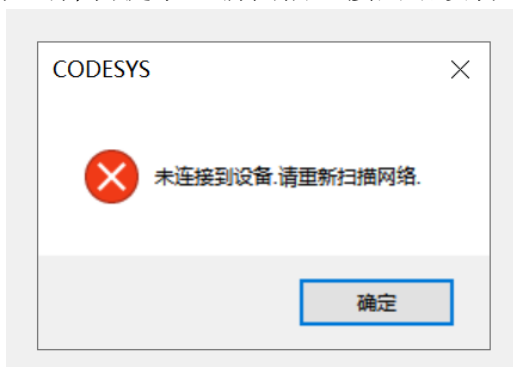


图10.6

(7) 可以重新读取一下IP，确认IP更改成功。



图10.6

二、网口MAC地址的获取和修改

(1) 与上原理，在扫描连接到网络设备后，在PLC指令页面单击下图中省略号方框一在“插入命令”弹窗中选择“GetMAC”（获取MAC地址指令）一点击“插入”即可。

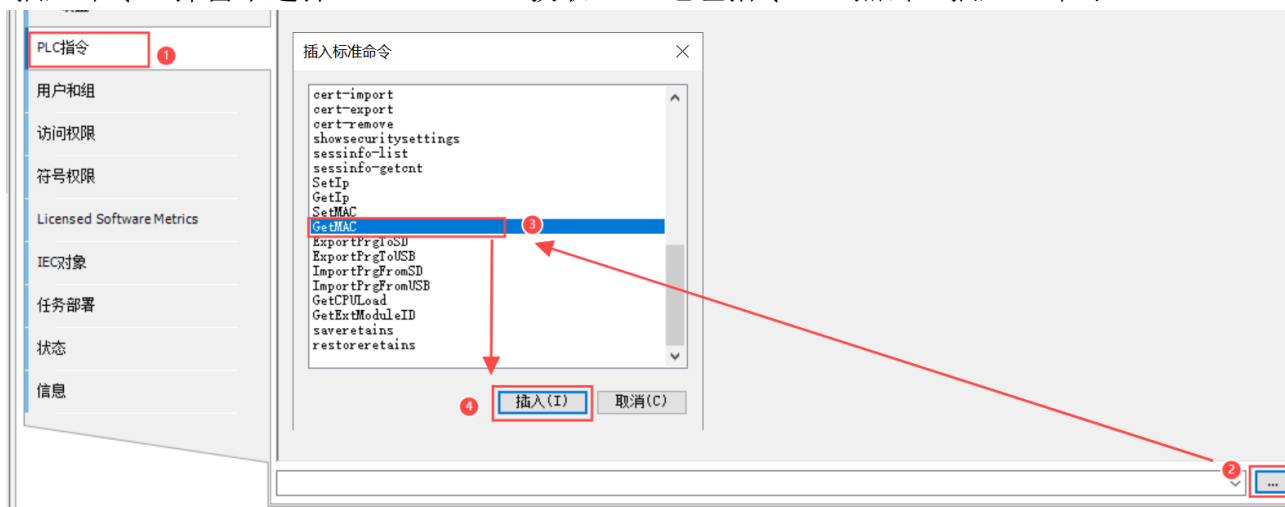


图10.7

(2) 插入后在显示栏上“GetMAC”指令后面，按空格输入网口号（这里输eth0）一按“Enter”键即可读出eth0网口当前的MAC地址。

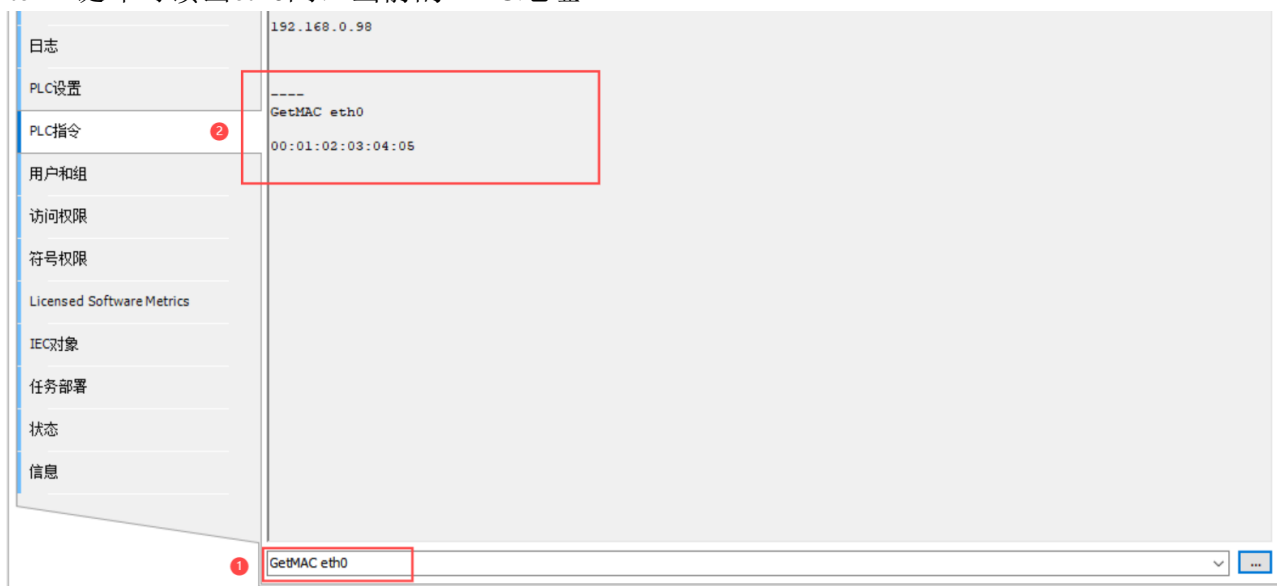


图10.8

(3) 继续点击省略号方框一在弹窗中选择“SetMAC”（设置MAC地址指令）一点击“插入”即可。

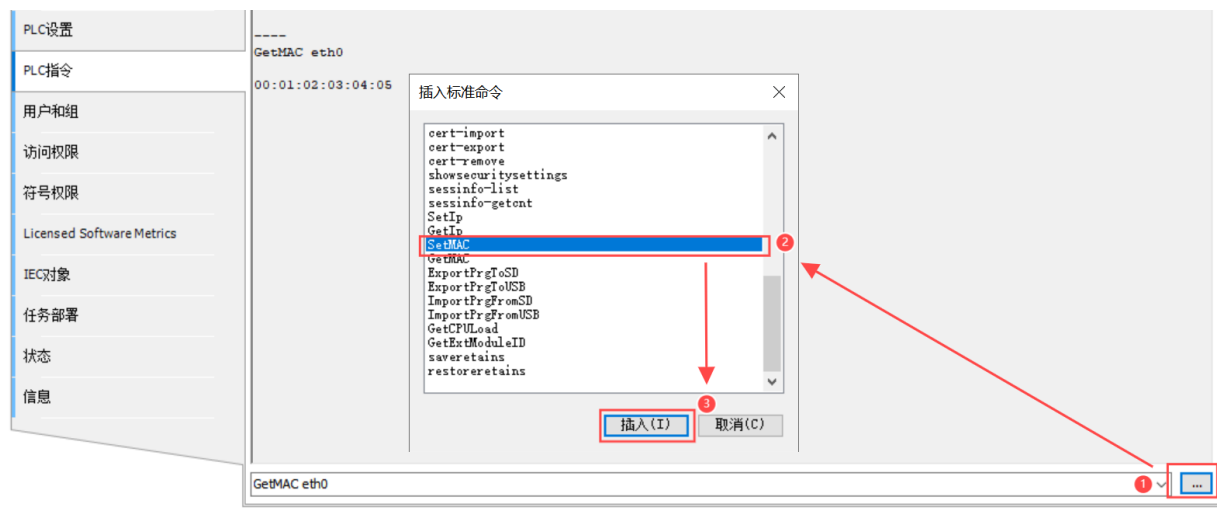


图10.9

(4) 在显示栏“SetMAC”后按空格—输入网口（这里输入eth0）—再按空格，输入要改成的MAC地址—按键盘上回车“Enter”即可。如图下所示更改成功和失败分别会提示的编码。



图10.10

(5) 可以再重新读取当前MAC地址，确认已更改成功。

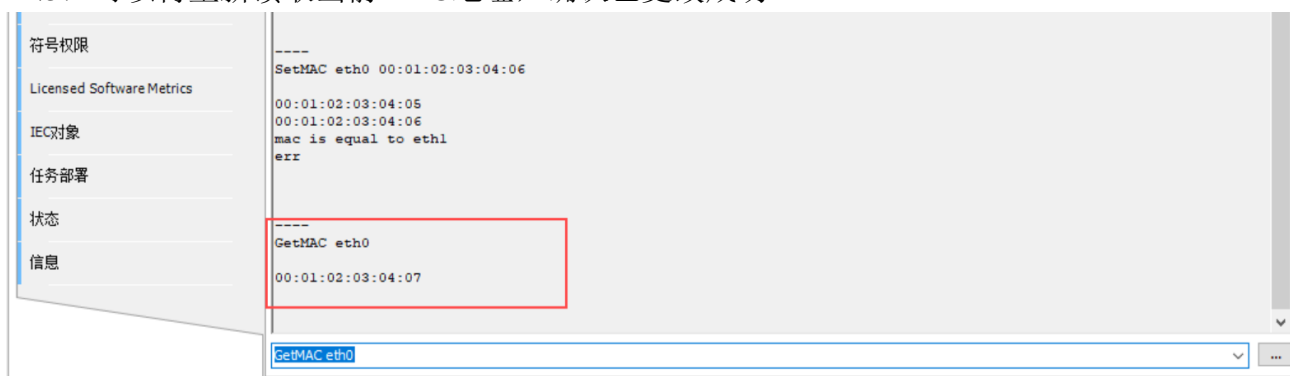


图10.10

附录1

1. Modbus通信设置。

Modbus通信设置配置

配置项	功能	
名称	通道命名的字符串	
存取类型	读线圈状态 (功能码01) 读输入状态 (功能码02) 读保持寄存器 (功能码03) 读输入寄存器 (功能码04) 写单个线圈 (功能码05) 写单个寄存器 (功能码06) 写多个线圈 (功能码15) 写多个寄存器 (功能码16)	
触发器	循环执行: 周期触发的请求	循环时间: 设置时间再次执行
	电平触发: 编程进行改变时触发	触发变量 (SM): 设置触发SM元件, 触发成功后, 自动复位该元件
重发次数	本次发生通信故障未获得从站返回帧, 则按重发次数进行重新发送。	
注释	可以对数据进行描述的简短文本区域	
读寄存器		
起始地址	读取的寄存器开始位置	
长度	读取的寄存器个数	
错误处理	保持最后的值: 使数据保持最后一次的有效值 设置为0: 使所有值归零	

2. 功能码和类型长度。

“长度”参数的有效范围取决于以下功能码:

功能码	类型访问	寄存器数
01	读线圈状态	1~2000
02	读输入状态	1~2000
03	读保持寄存器	1~125
04	读输入寄存器	1~125
05	写单个线圈	1
06	写单个寄存器	1
15	写多个线圈	1~1968
16	写多个寄存器	1~123

功能码01 05 15 0-11999 Q_bit(QX0.0-QX1499.7) 12000-65535 M_bit(MX0.0-QX6691.7)
 功能码02 0-65535 I_bit(IX0.0-IX8191.7)
 功能码03 06 16 0-49999 MW0-49999 50000-65535 QW0-15535 (QW0-QW15535)
 功能码04 0-65535 IW0-65535(IX0.0-IX8191.7)

3. CODESYS地址说明。

%_X	195.7 – 195.0	194.7 – 194.0	193.7 – 193.0	192.7 – 192.0
%_B	195 (高8位)	194 (低8位)	193 (高8位)	192 (低8位)
%_W	97 (高16位)		96 (低16位)	
%_D	48			

QB0= (QX0.0~QX0.7) ;

QW0= (QB0~QB1) = ((QX0.0~QX0.7) + (QX1.0~QX1.7)) ;

QD0= (QW0~QW1) = (QB0~QB3) = ((QX0.0~QX0.7) + (QX1.0~QX1.7) + (QX2.0~QX2.7) + (QX3.0~QX3.7)) ;

