

TM700系列PLC快速指南

www.invt.com.cn



CONTENTS

01 **TM700-产品介绍**

02 **通讯示例**

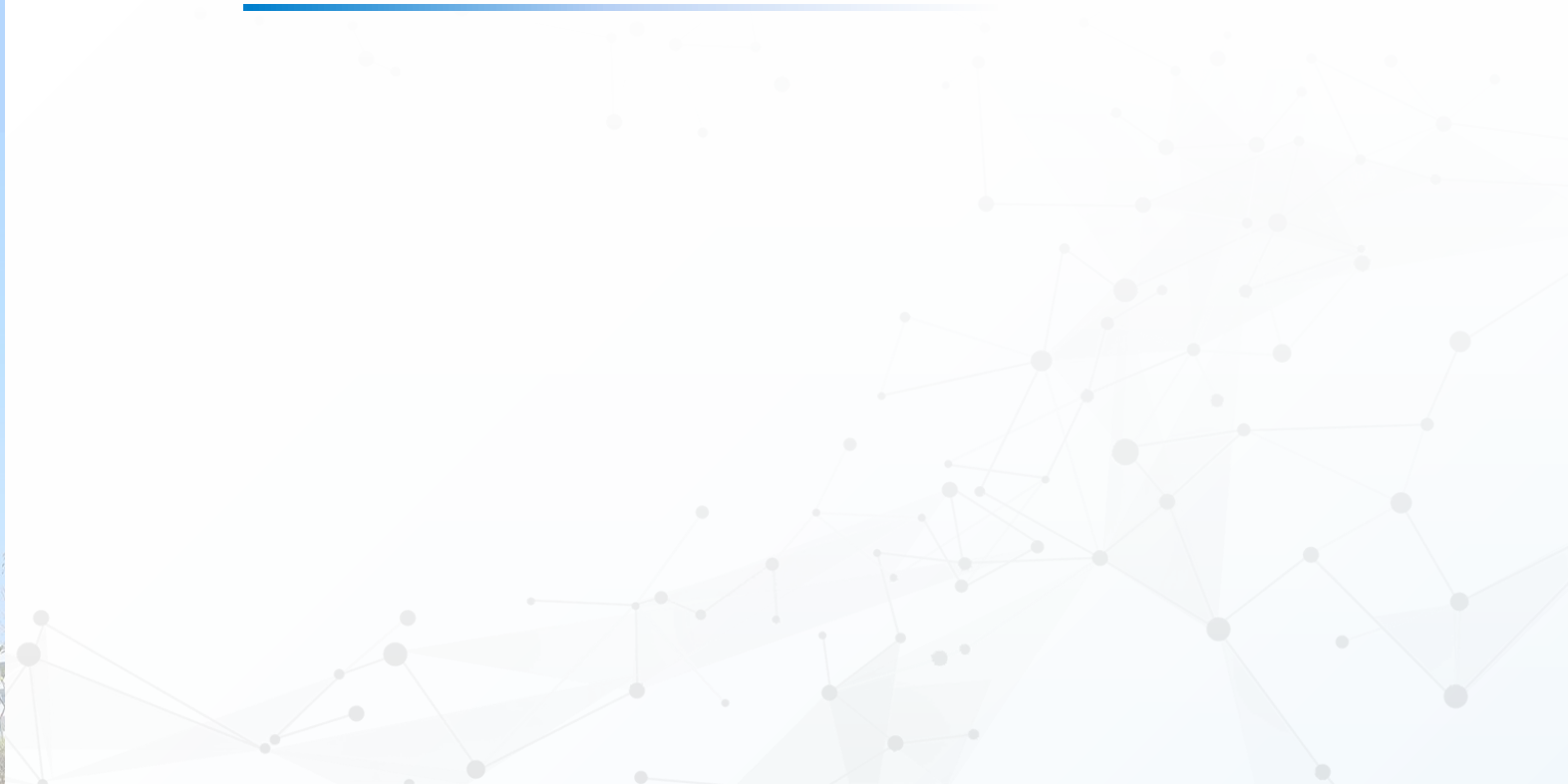
03 **高速计数器**

04 **本地脉冲轴**



PART 01

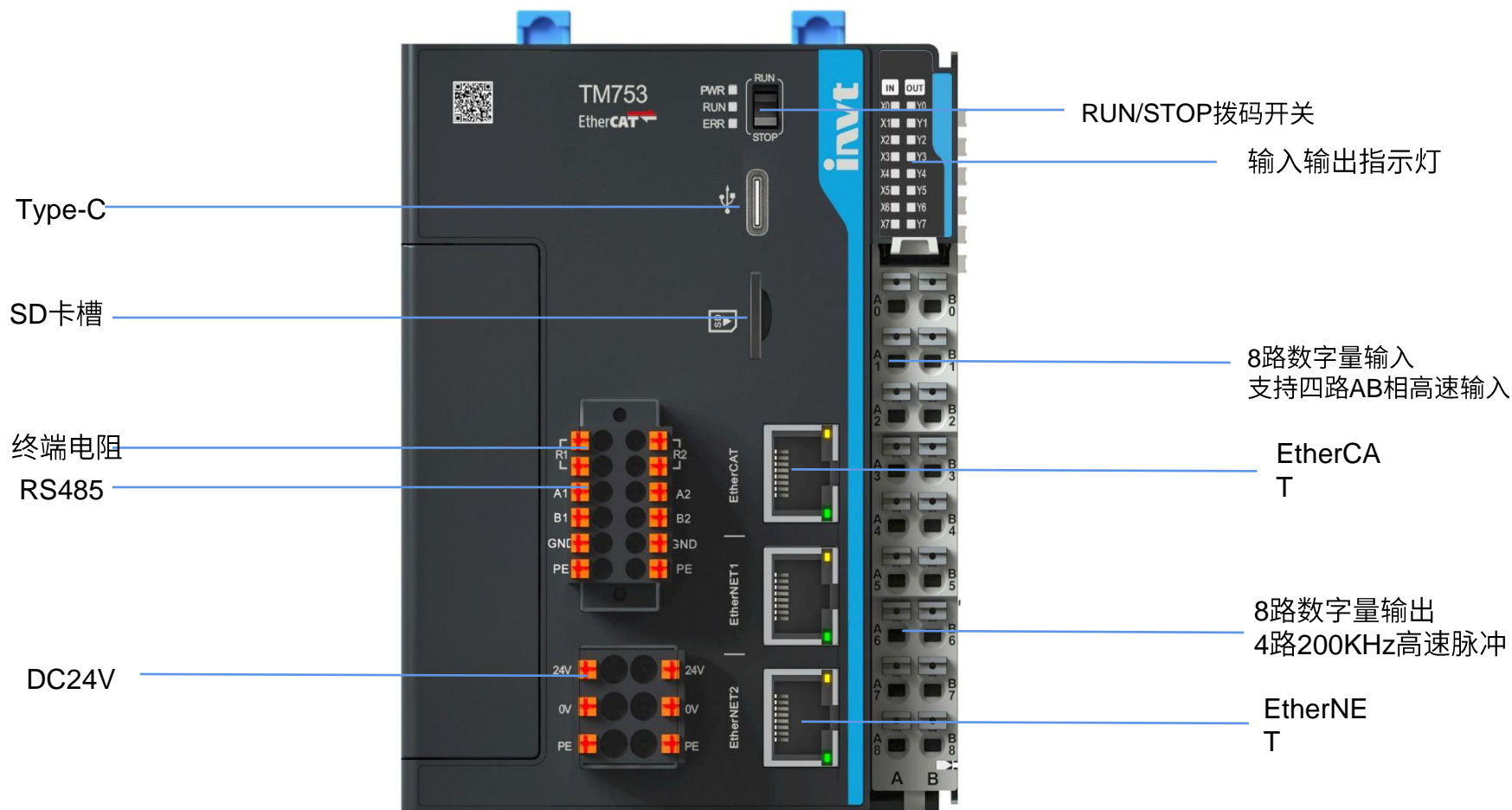
TM700-产品介绍





硬件接口介绍

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能





产品规格

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



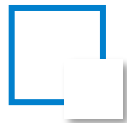
	TM750	TM751	TM752	TM753
数字量输入	8DI	8DI	8DI	8DI
数字量输出	8DO	8DO	8DO	8DO
高速输入	4路AB相	4路AB相	4路AB相	4路AB相
脉冲轴数	4路	4路	4路	4路
总线轴数	EtherCAT;4轴	EtherCAT;8轴	EtherCAT;16轴	EtherCAT;32轴
Ethernet	2路	2路	2路	2路
串口	RS485*2	RS485*2	RS485*2	RS485*2



PART 02

通讯示例

- **TM753与VS_102QS 通讯（串口）**
- **TM753与VS_102QS通讯（网口标签
通讯）**
- **MODBUS TCP通讯**
- **MODBUS RUT主站控制变频器**
- **TCP IP自由口协议**

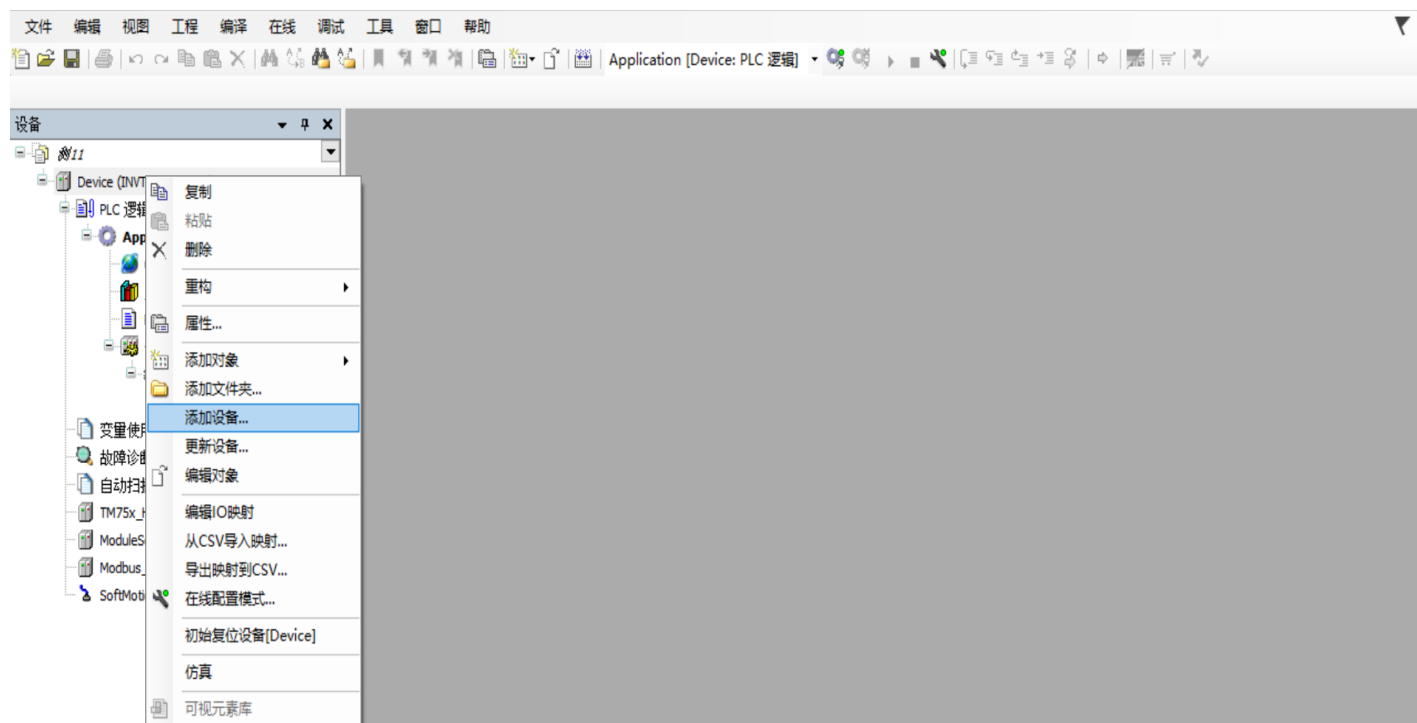


TM753与VS_102QS 通讯（串口）

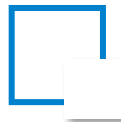
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753作为Modbus从站，以COM1为例：



1. Invtmatic Studio 新建工程
TM753 485 通讯，右击左侧
“Device”，点击“添加设备”。

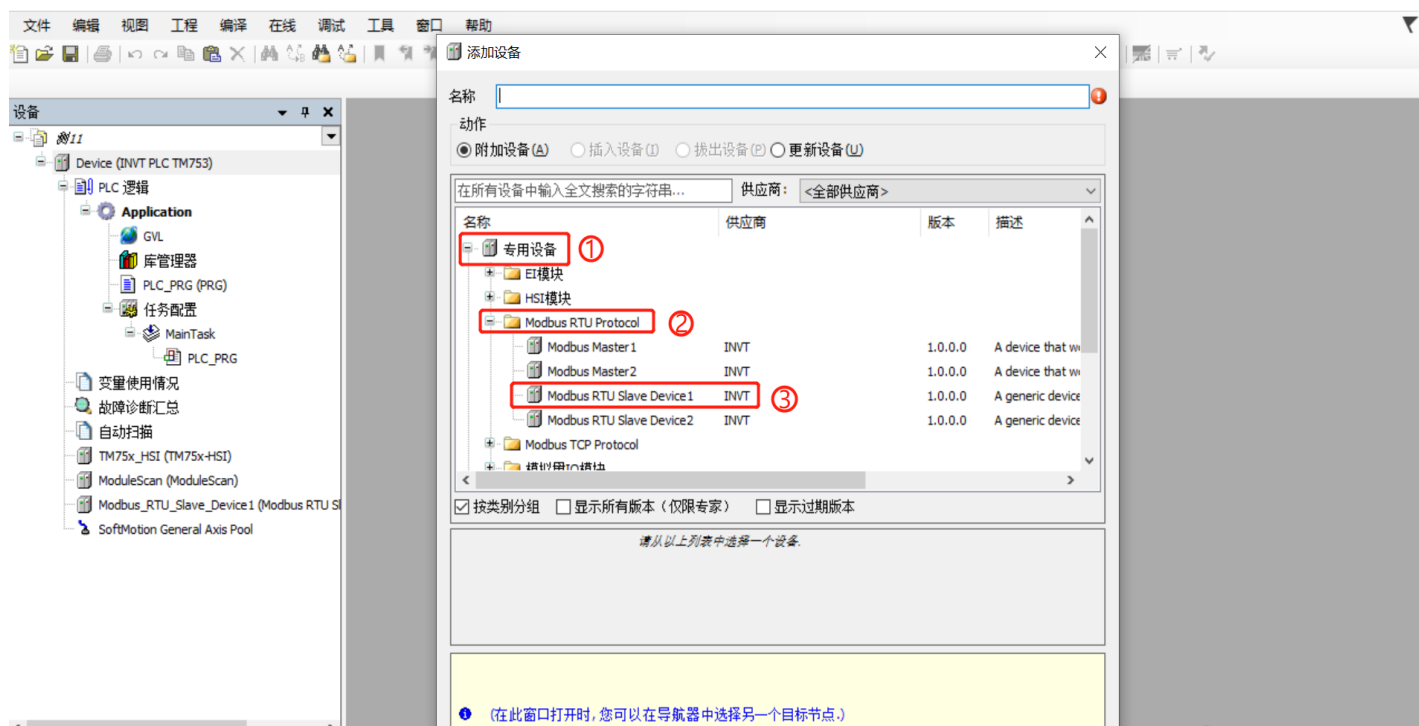


TM753与VS_102QS 通讯（串口）

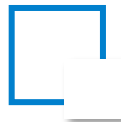
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



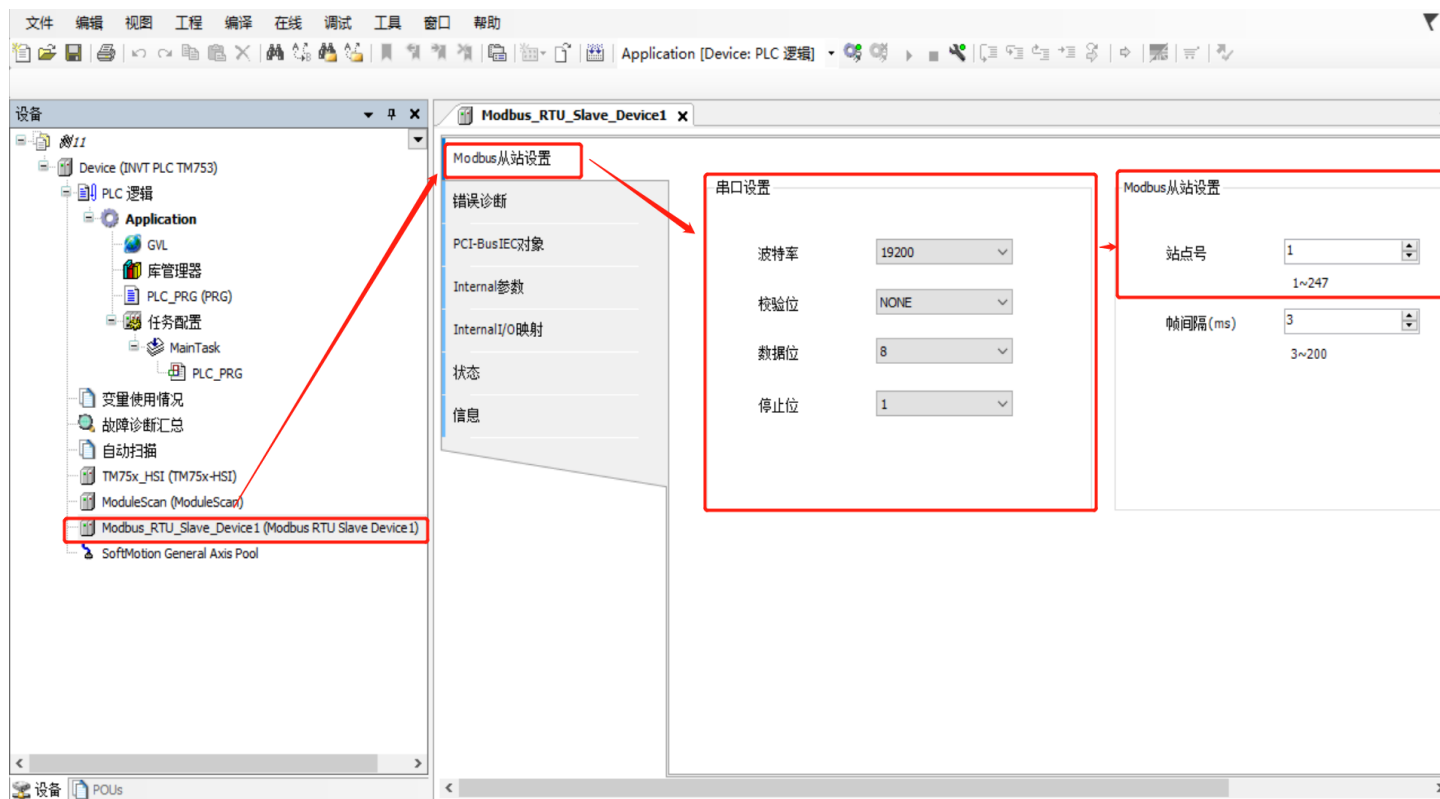
TM753作为Modbus从站，以COM1为例：



2. 点击“Modbus RTU protocol”，通过COM1使用485通讯则选择Modbus RTU Slave Device1。



TM753作为Modbus从站，以COM1为例：



3.双击左侧
Modbus_RTU_Slave_Device1进
行从站配置，串口配置参数与触摸
屏参数对应，Modbus从站配置中
的站号也要对应。

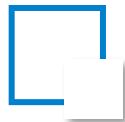


TM753作为Modbus从站，以COM1为例：

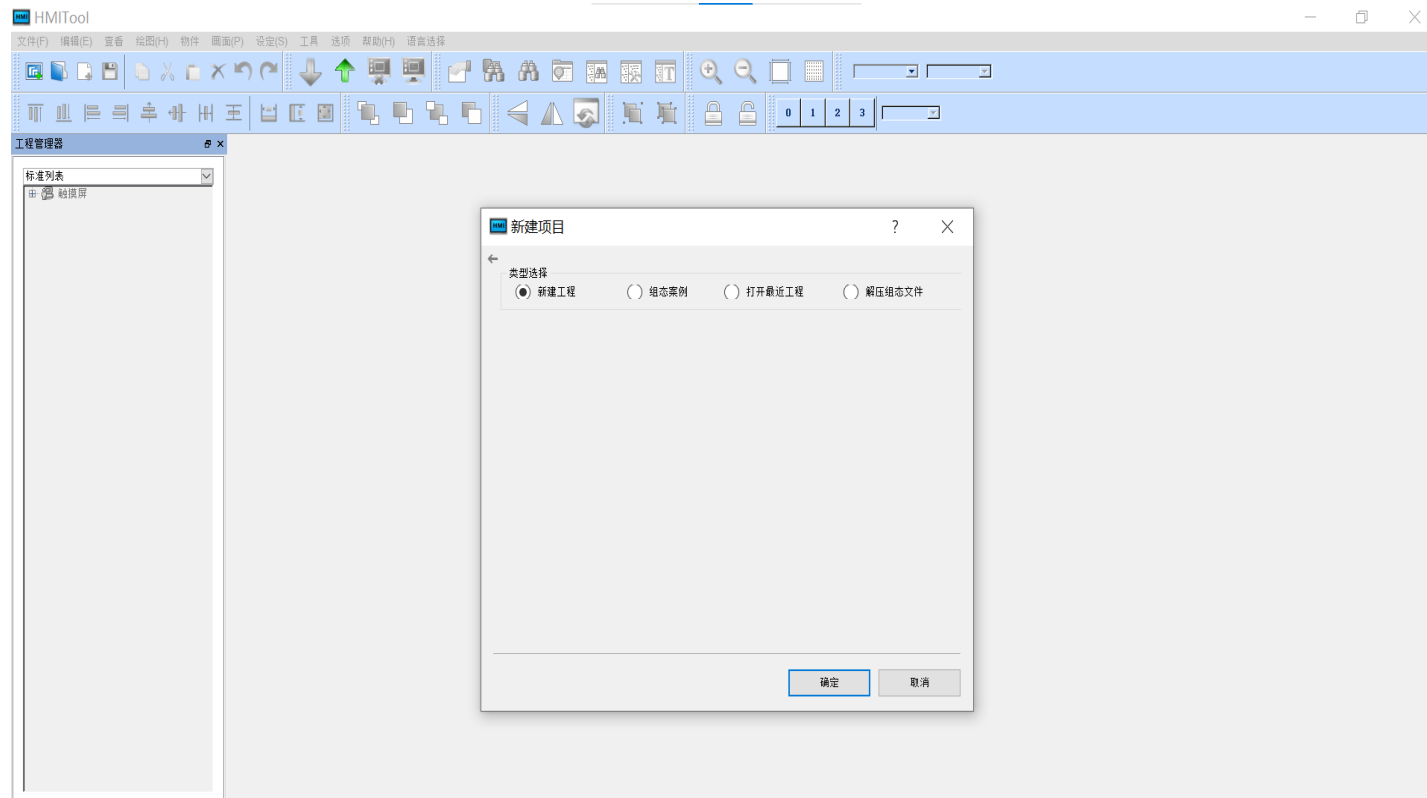
```
PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3      x0      AT  %MX0.0 : BOOL;
4      x1      AT  %MX0.1 : BOOL;
5      x2      AT  %MX0.2 : BOOL;
6      call    AT  %MD1  : REAL;
7      cal2    AT  %MD2  : REAL;
8      cal3    AT  %MD3  : INT;
9  END_VAR

1  x0:=x1;
2  x2:=TRUE;
3  IF x0 THEN
4      call:=cal2+cal3;
5  ELSE
6      call:=0;
7  END_IF
8
```

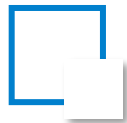
4.在 PLC_PRG 新建变量并绑定地址、编写的示例程序如图所示；
然后编译工程，下载；



VS_102QS作为Modbus主站，以COM1为例：



1.打开 HMITOOL 软件，点击右上方“文件”，点击“新建工程”，点击“确定”。

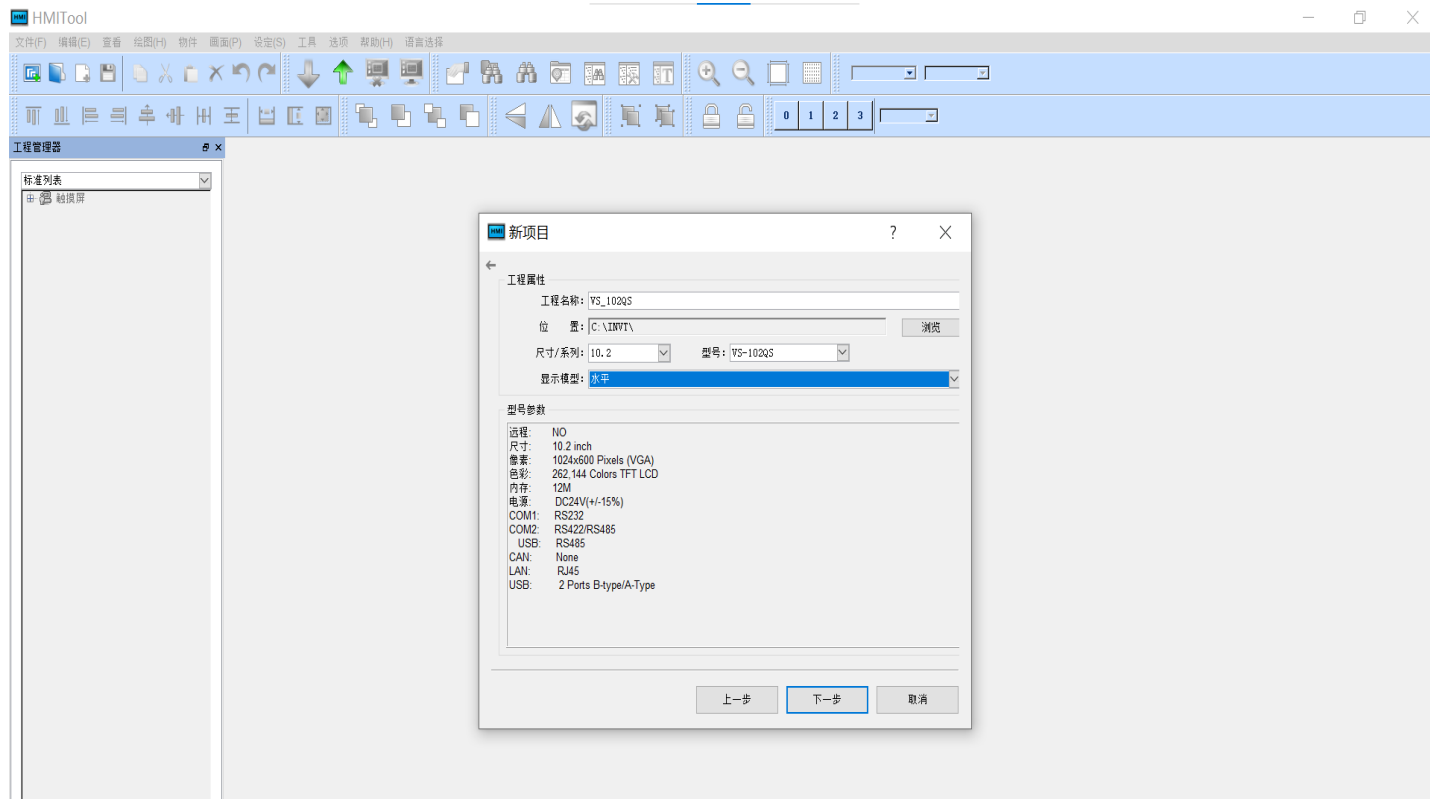


TM753与VS_102QS 通讯（串口）

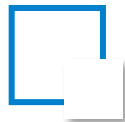
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



VS_102QS作为Modbus主站，以COM1为例：



2.工程名称填写“VS_102QS 通讯”，点击“浏览”选择文件存储位置，点击“尺寸/系列”选择10.2，点击“型号”选择“VS_102QS”，点击下一步。

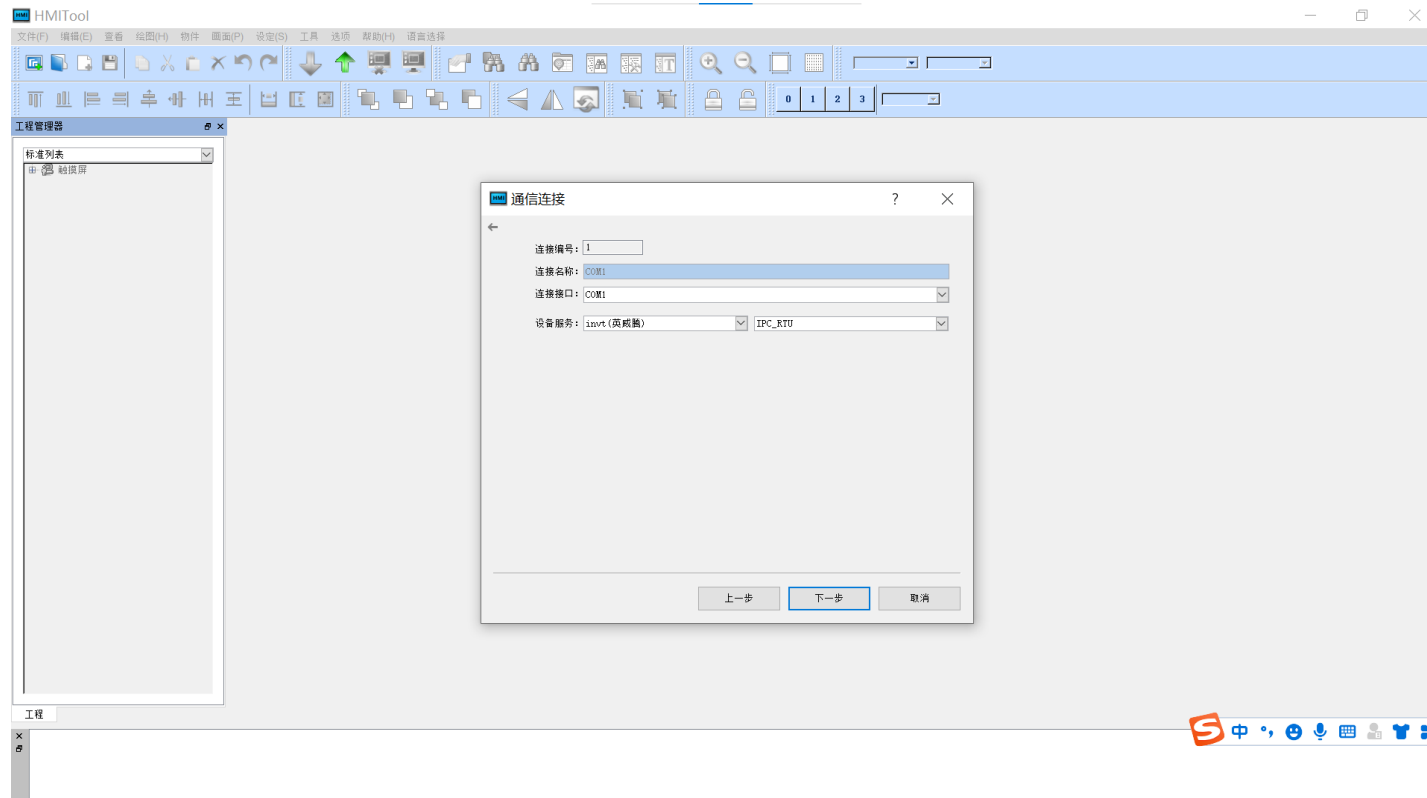


TM753与VS_102QS 通讯（串口）

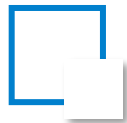
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



VS_102QS作为Modbus主站，以COM1为例：



3.连接端口选择“COM1”，设备服务选择“invt(英威腾)”，选择“IPC_RTU”，点击下一步，选择背景颜色，点击“完成”。

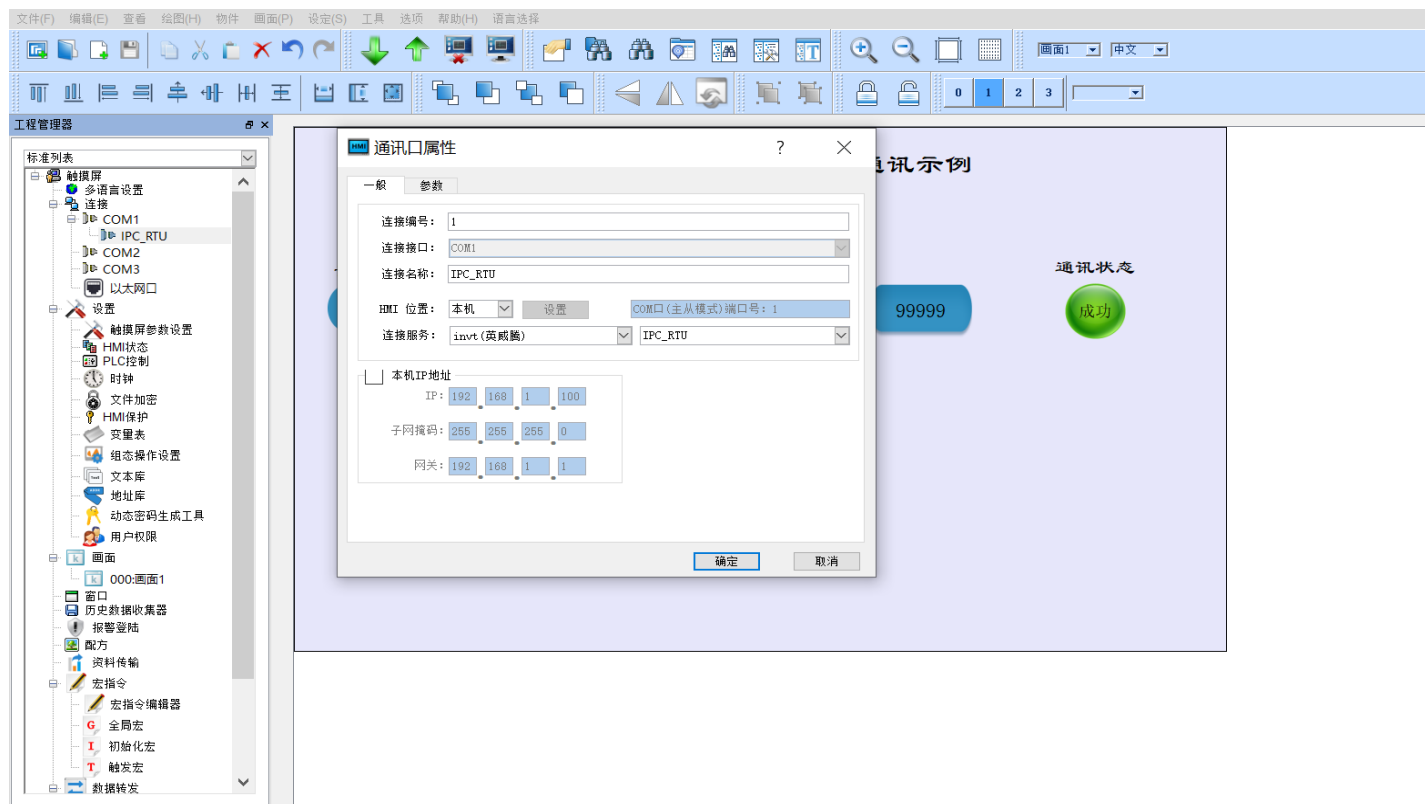


TM753与VS_102QS 通讯（串口）

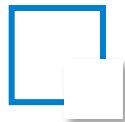
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



VS_102QS作为Modbus主站，以COM1为例：



4.在“连接”中选择“COM1”，双击“IPC_RTU”，HMI位置选择“本机”，连接服务选择“INVT（英威腾）”、“IPC_RTU”。

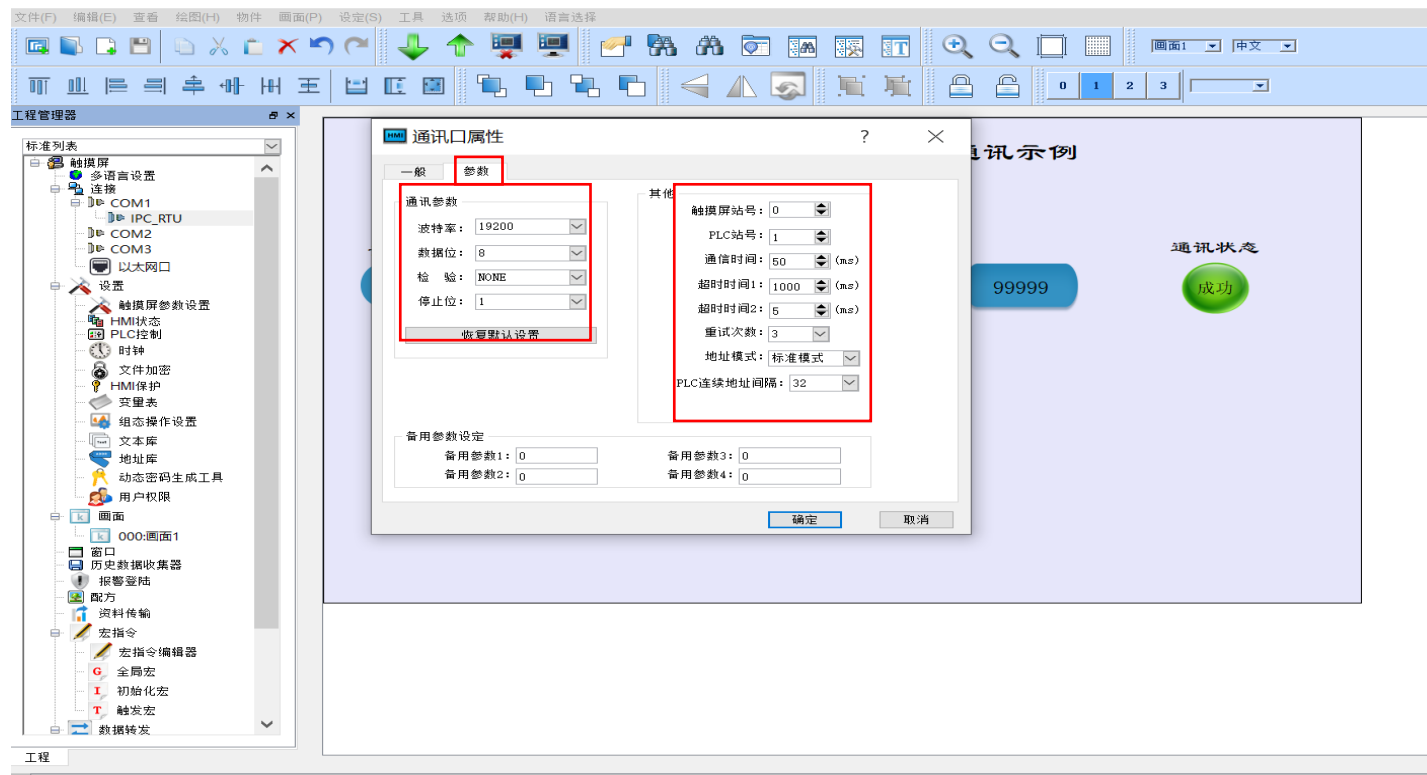


TM753与VS_102QS 通讯（串口）

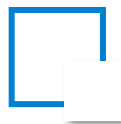
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



VS_102QS作为Modbus主站，以COM1为例：



5.点击“参数”设置通讯参数，通讯参数的设置要与前面 PLC 软件从站配置对应，设置完成点击“确定”



VS_102QS作为Modbus主站，以COM1为例：

TM系列PLC与触摸屏485通讯示例

计算开关



19.25

=

11.25

+

8

通讯状态



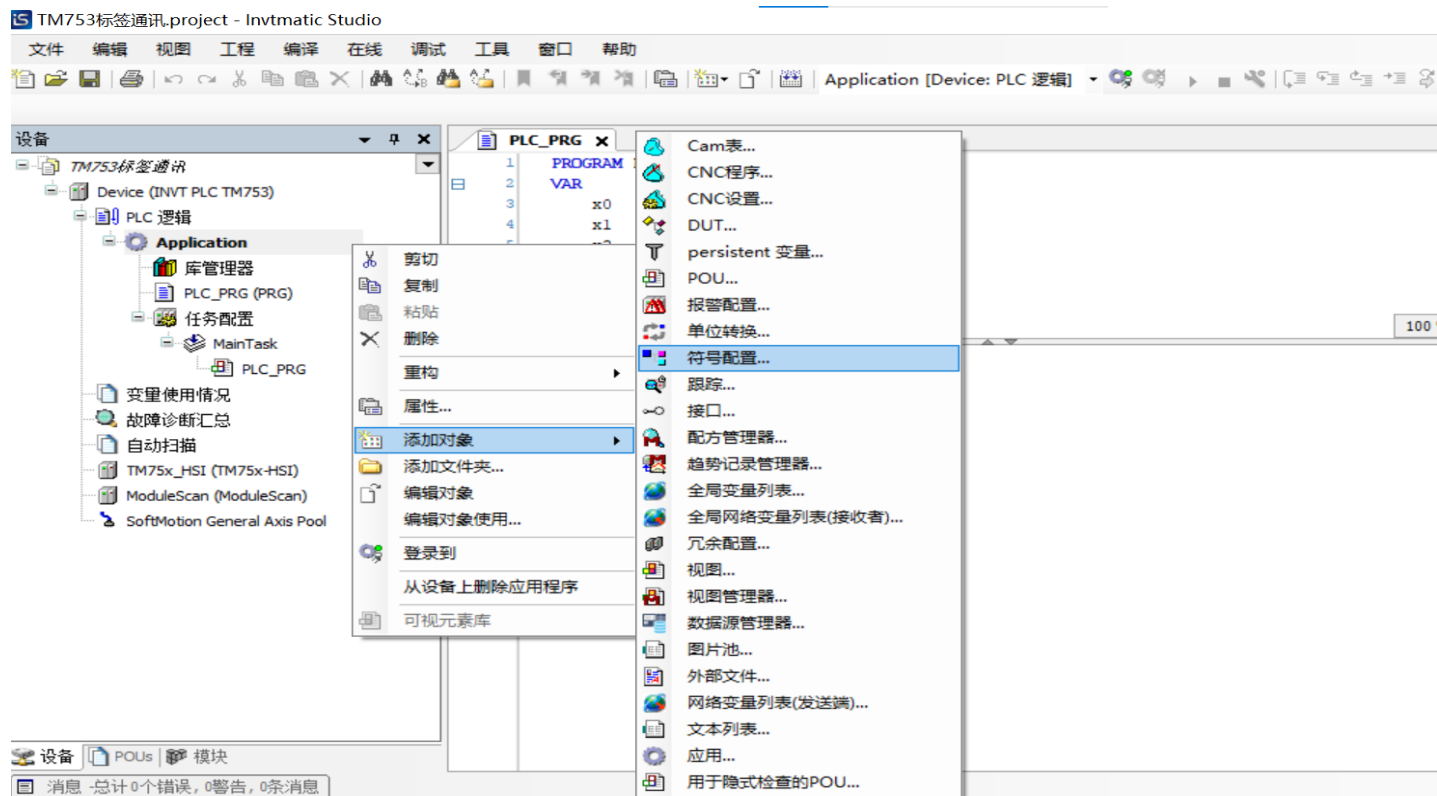
6.设计触摸屏界面，选择控件，并绑定对应PLC的地址，设计好触摸屏界面后，将触摸屏程序编译，选择合适的方式将触摸屏程序下载到触摸屏中，下载完成后，用 485 接线与触摸屏连接，单击触摸屏上开关，并进行相关操作，若能正常读到数据及开关状态，表明触摸屏与 PLC 通信正常。触摸界面如图所示：

PART 02

通讯示例

- TM753与VS_102QS 通讯（串口）
- **TM753与VS_102QS通讯（网口标签通讯）**
- MODBUS TCP通讯
- MODBUS RUT主站控制变频器
- TCP IP自由口协议

TM系列软件配置



1.新建工程 TM753标签通讯，在左侧设备树中鼠标右击“Application”，点击“添加对象”，点击“符号配置”。

TM系列软件配置

添加 符号配置 ×

 远程访问符号配置.

名称

符号配置

☒ 在XML中包含注释

☒ 支持 OPC UA特征

☐ 在设备应用程序中添加占位符
(推荐,但可能导致下载)

客户端数据布局

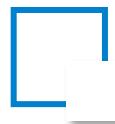
☐ 兼容性布局

☒ 优化布局

打开

取消

2.弹出下面窗口后，勾选“支持 OPC UA”特征和“在XML中包含注释”，点击“打开”。



TM753与VS_102QS通讯（网口标签通讯）

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



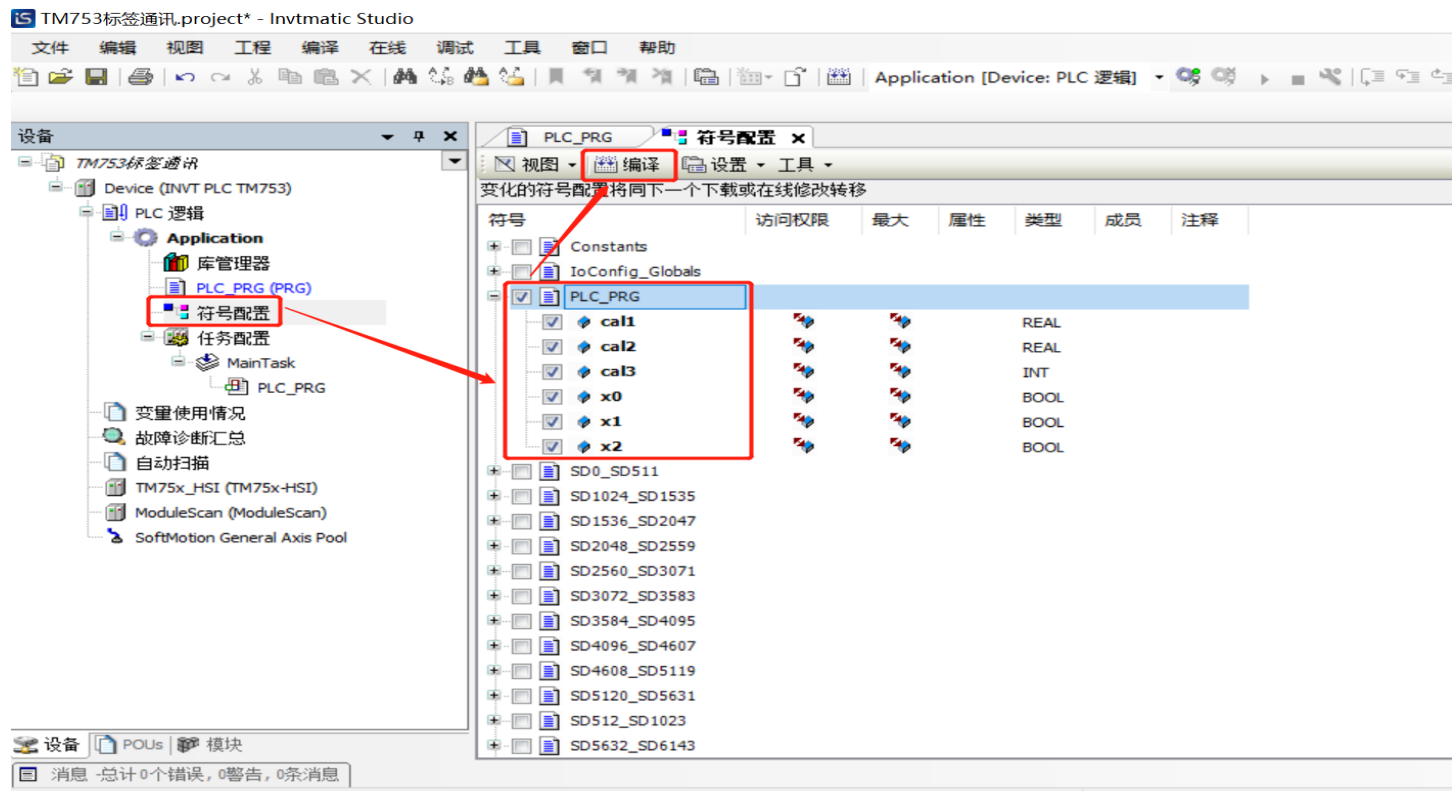
TM系列软件配置

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3     x0      AT  %MX0.0 : BOOL;
4     x1      AT  %MX0.1 : BOOL;
5     x2      AT  %MX0.2 : BOOL;
6     cal1    AT  %MD1   : REAL;
7     cal2    AT  %MD2   : REAL;
8     cal3    AT  %MD3   : INT;
9 END_VAR

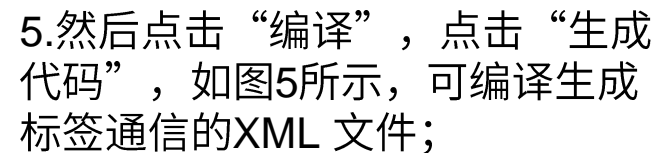
1 x0:=x1;
2 x2:=TRUE;
3 IF x0 THEN
4     cal1:=cal2+cal3;
5 ELSE
6     cal1:=0;
7 END_IF
8
```

3.在 PLC_PRG 新建变量、程序如图 所示；

TM系列软件配置



4.在左侧设备树中双击“符号配置”，打开符号配置页面，勾选需要进行标签通讯的变量，然后编译工程，如图所示.

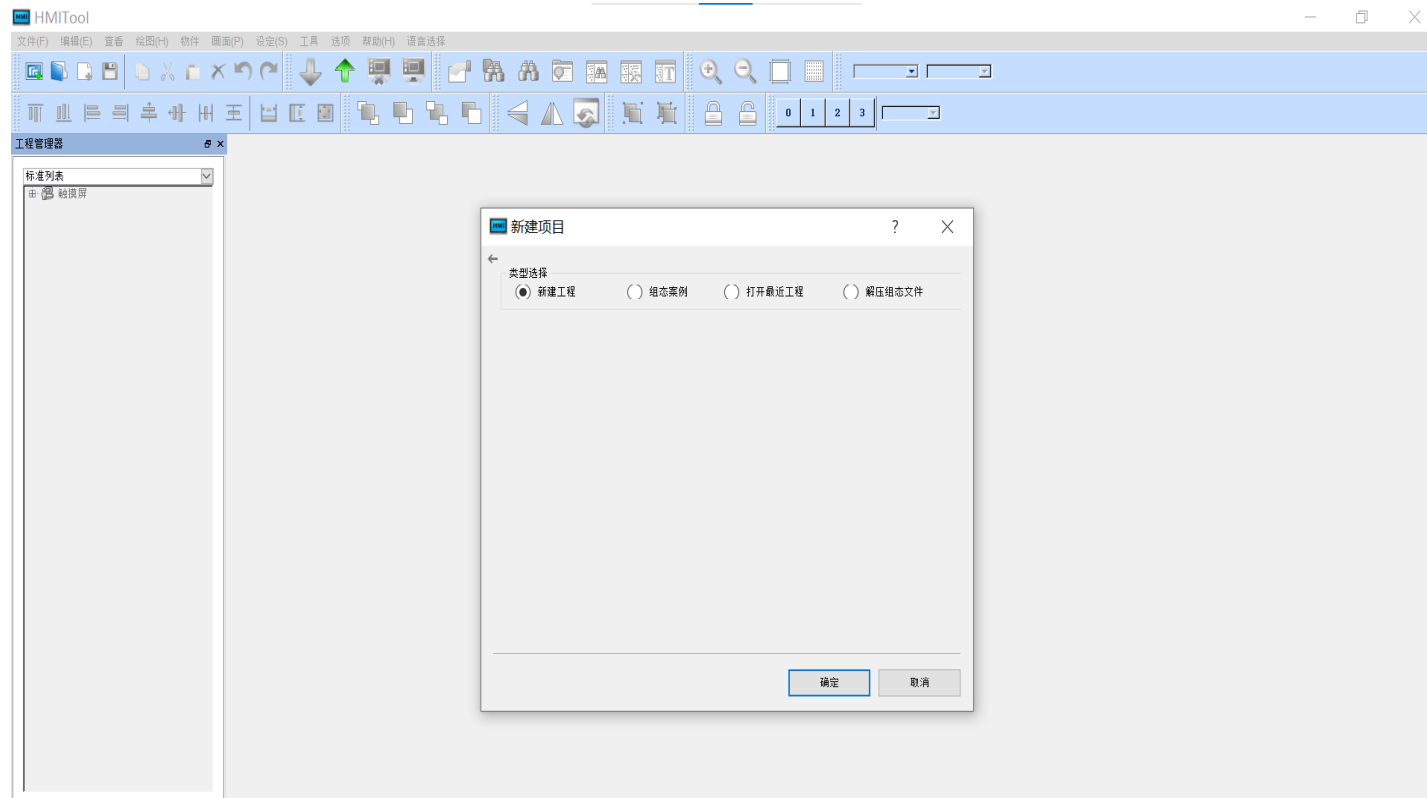


TM系列软件配置

名称	修改日期	类型
 TM753标签通讯.Device.Application.xml	2023/11/20 10:37	XML 文档
 TM753标签通讯.project	2023/11/20 10:12	Invtmatic Studi
 TM753标签通讯.project.~u	2023/11/20 10:40	~U 文件

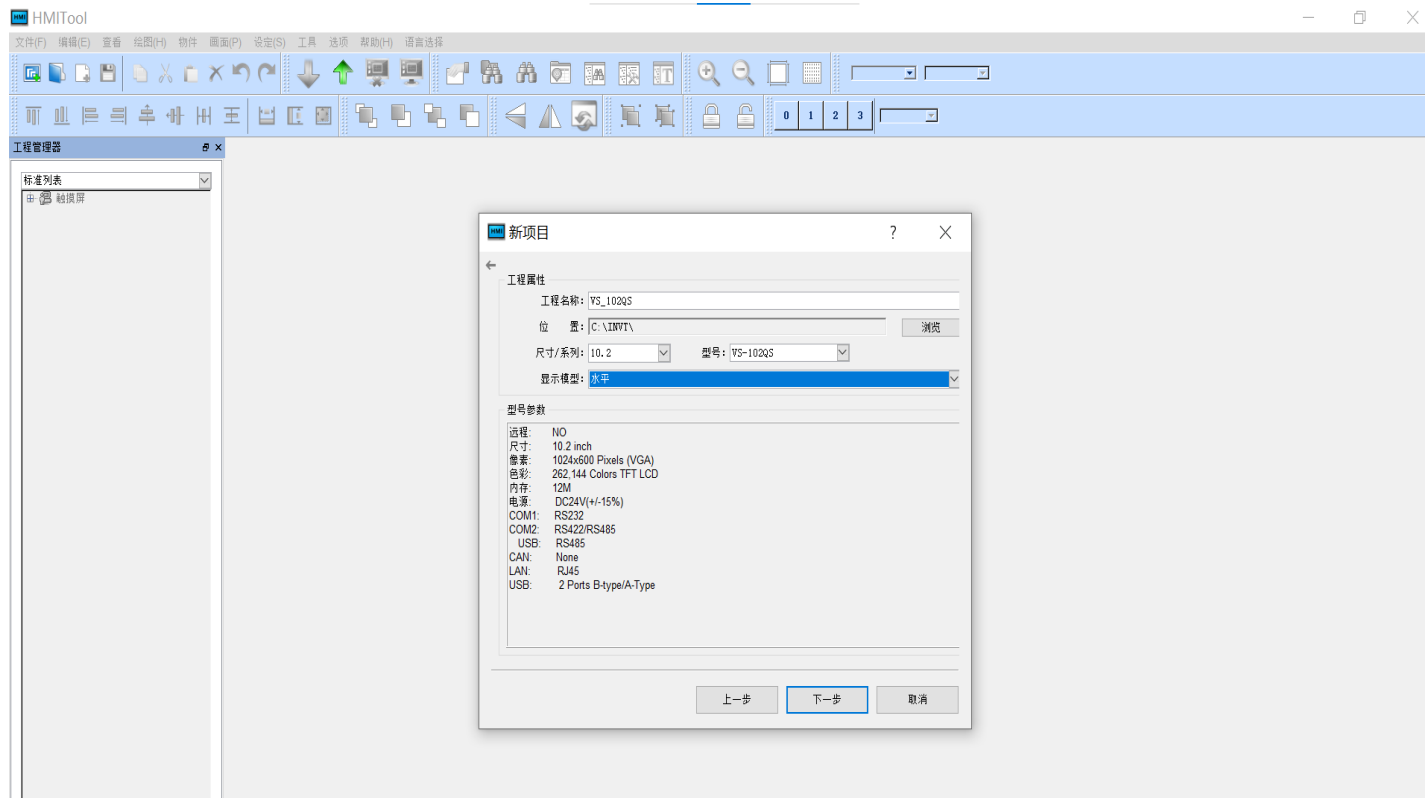
6.打开“TM753标签通讯”工程所在目录，可看见生成的 XML 文件，如下图所示，这个 XML 文件就是用于导入触摸屏的标签通信变量；至此，PLC 配置完成，将程序编译并下载到 PLC 中并运行。

VS_102QS软件配置



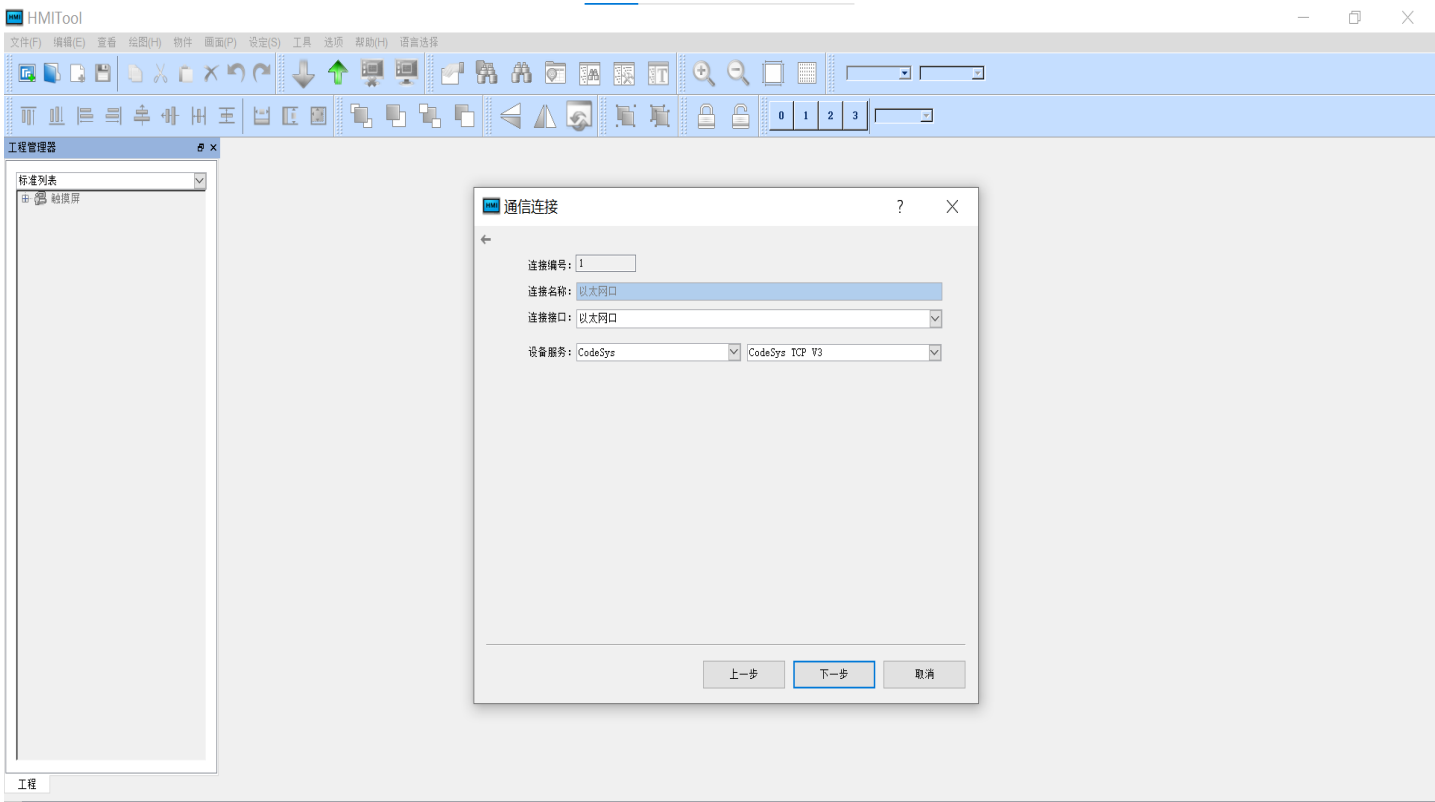
1.打开 HMITOOL 软件，点击右上方“文件”，点击“新建工程”，点击“确定”。

VS_102QS软件配置



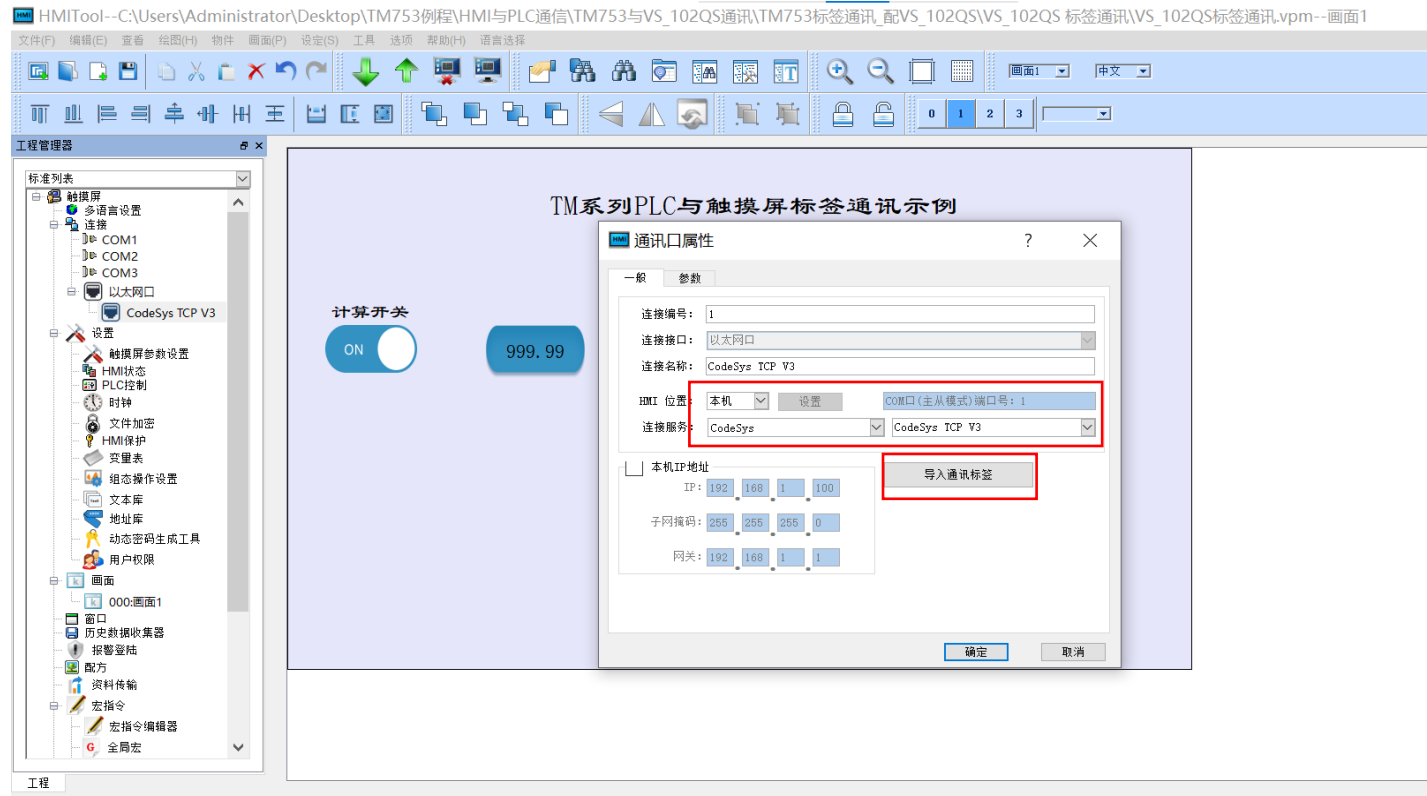
2.工程名称填写“VS_102QS 通讯”，点击“浏览”选择文件存储位置，点击“尺寸/系列”选择10.2，点击“型号”选择“VS_102QS”，点击下一步。

VS_102QS软件配置



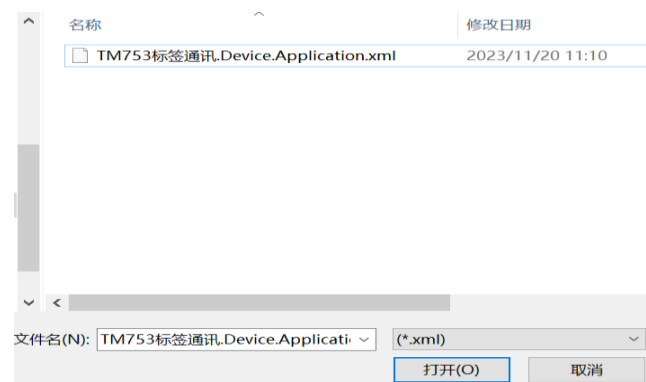
3.连接端口选择“以太网口”，设备服务选择“CodeSys”，选择“CodeSys TCP V3”，点击下一步，选择背景颜色，点击“完成”。

VS_102QS软件配置

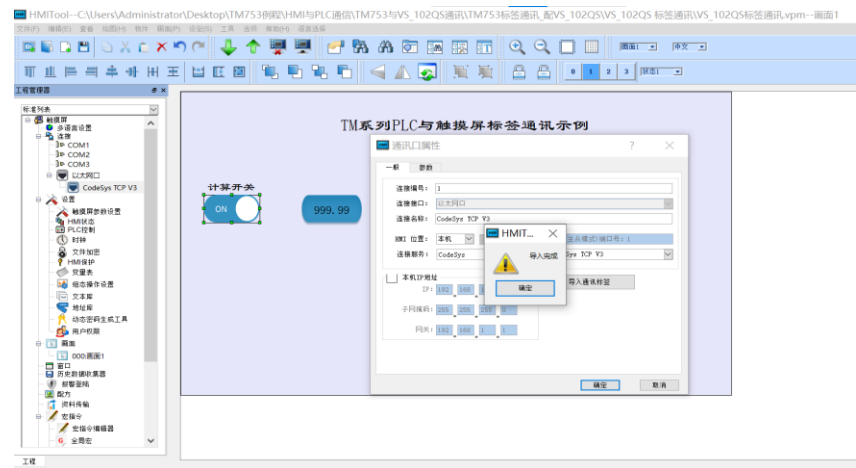


4.在“连接”中选择“以太网口”，双击“CodeSys TCP V3”，HMI位置选择“本机”，连接服务选择“CodeSys”、“CodeSys TCP V3”，点击“导入通讯标签”。

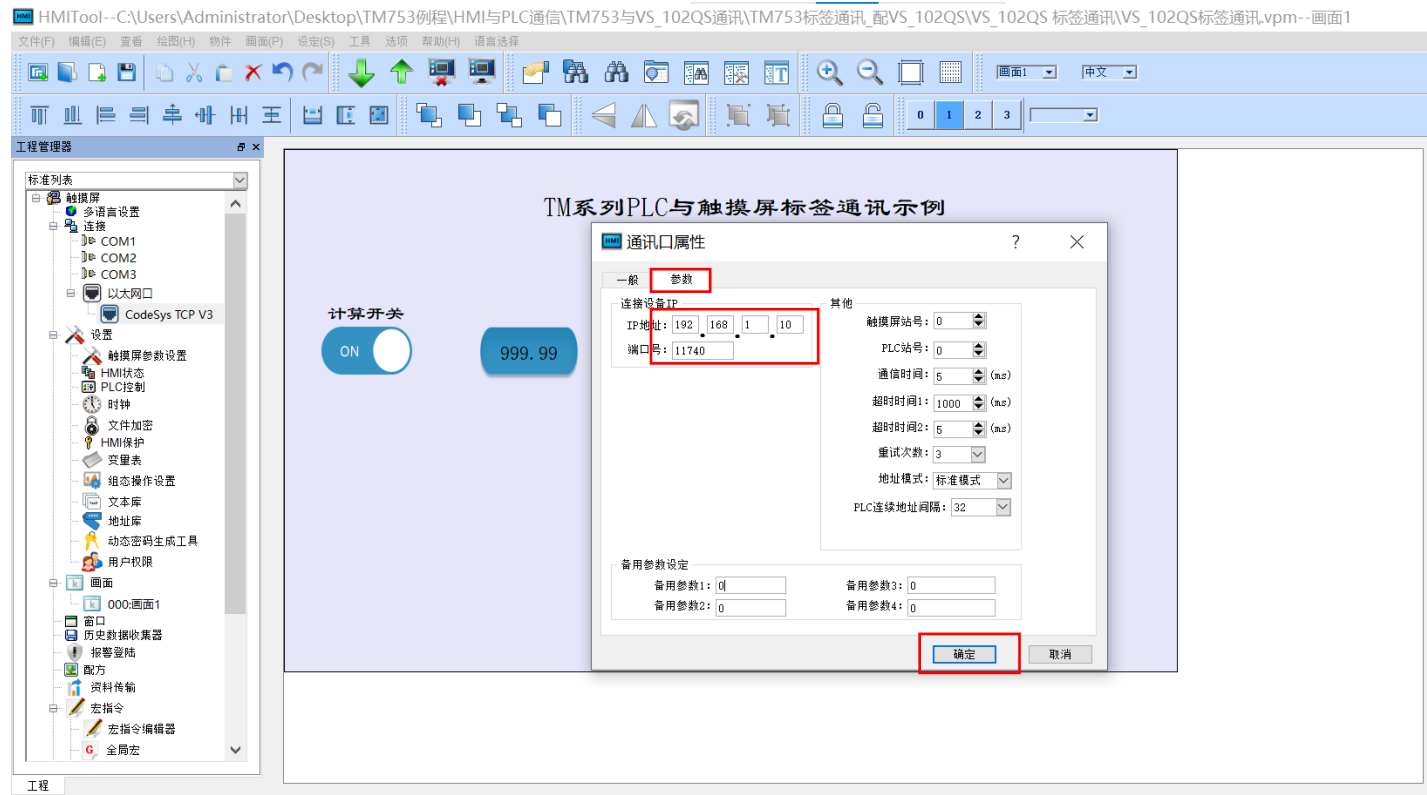
VS_102QS软件配置



5.选择刚刚生成的XML标签，点击打开，显示导入完成，点击“确定”；

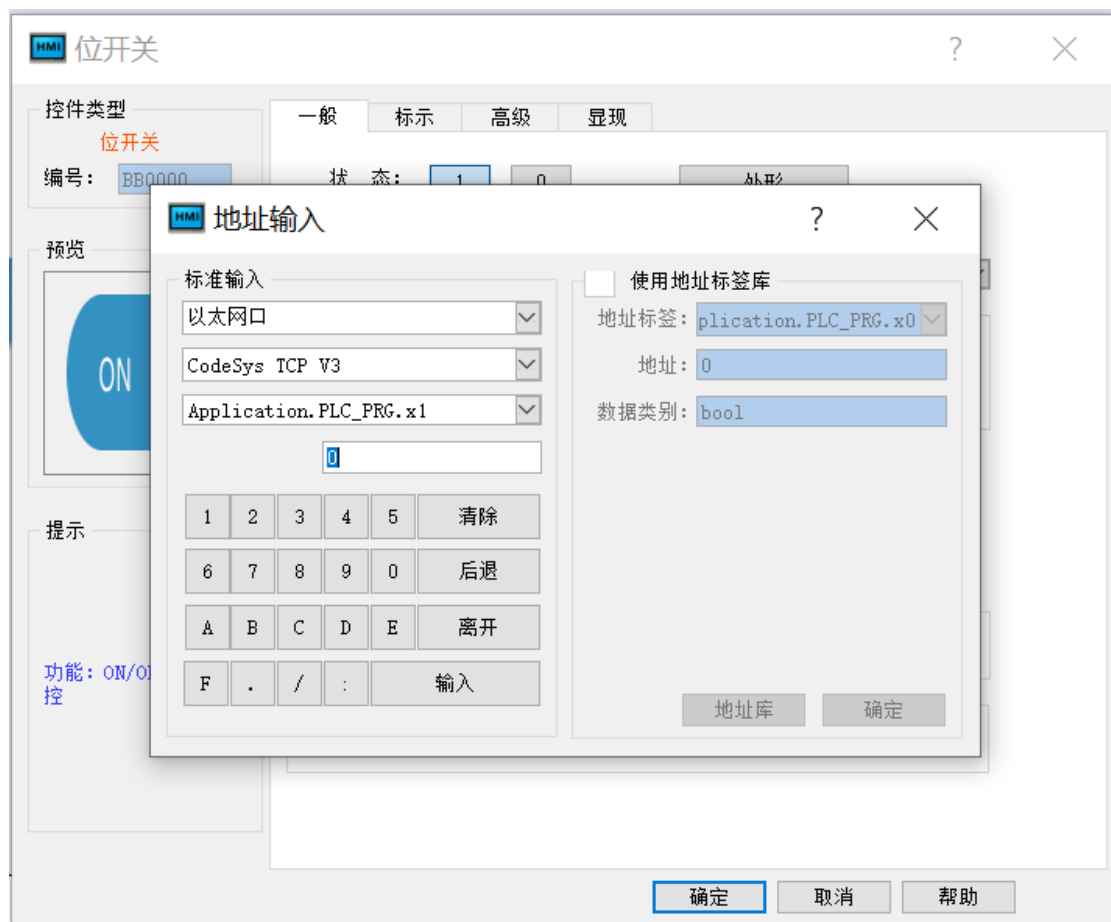


VS_102QS软件配置

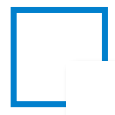


6. 点击“参数”设置通讯参数，通讯参数的IP地址设置为“192.168.1.10”，端口号设置为“11740”，设置完成点击“确定”，

VS_102QS软件配置



7.设计触摸屏界面，选择控件，并绑定对应PLC的地址；设计好触摸屏界面后，将触摸屏程序编译，选择合适的方式将触摸屏程序下载到触摸屏中。



TM753与VS_102QS通讯（网口标签通讯）

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



VS_102QS软件配置

TM系列PLC与触摸屏标签通讯示例

计算开关



8.00

=

3.00

+

5

通讯状态



8.下载完成后，用网线与触摸屏连接，单击触摸屏上开关，并进行相关操作，若能正常读到数据及开关状态，表明触摸屏与 PLC 标签通信正常。

PART 02

通讯示例

- TM753与VS_102QS 通讯（串口）
- TM753与VS_102QS通讯（网口标签通讯）
- **MODBUS TCP**通讯（主站）
- MODBUS RUT主站控制变频器
- TCP IP自由口协议



TM753做MODBUS TCP主站前的准备

Modbus 协议是主从交互式协议，即主机轮询、从机应答。在工控自动化中应用广泛，可基于 RS232/RS485/网口等硬件接口基础上应用。该协议的报文结构主要由地址、功能码、寄存器、数据、校验码等构成。

以下是基于网口的 Modbus TCP 主从站通信示例。

1) 功能运行前的准备(PLC 做主站): 1个控制器, 1台电脑, 电脑安装ModBus Slave 调试软件;

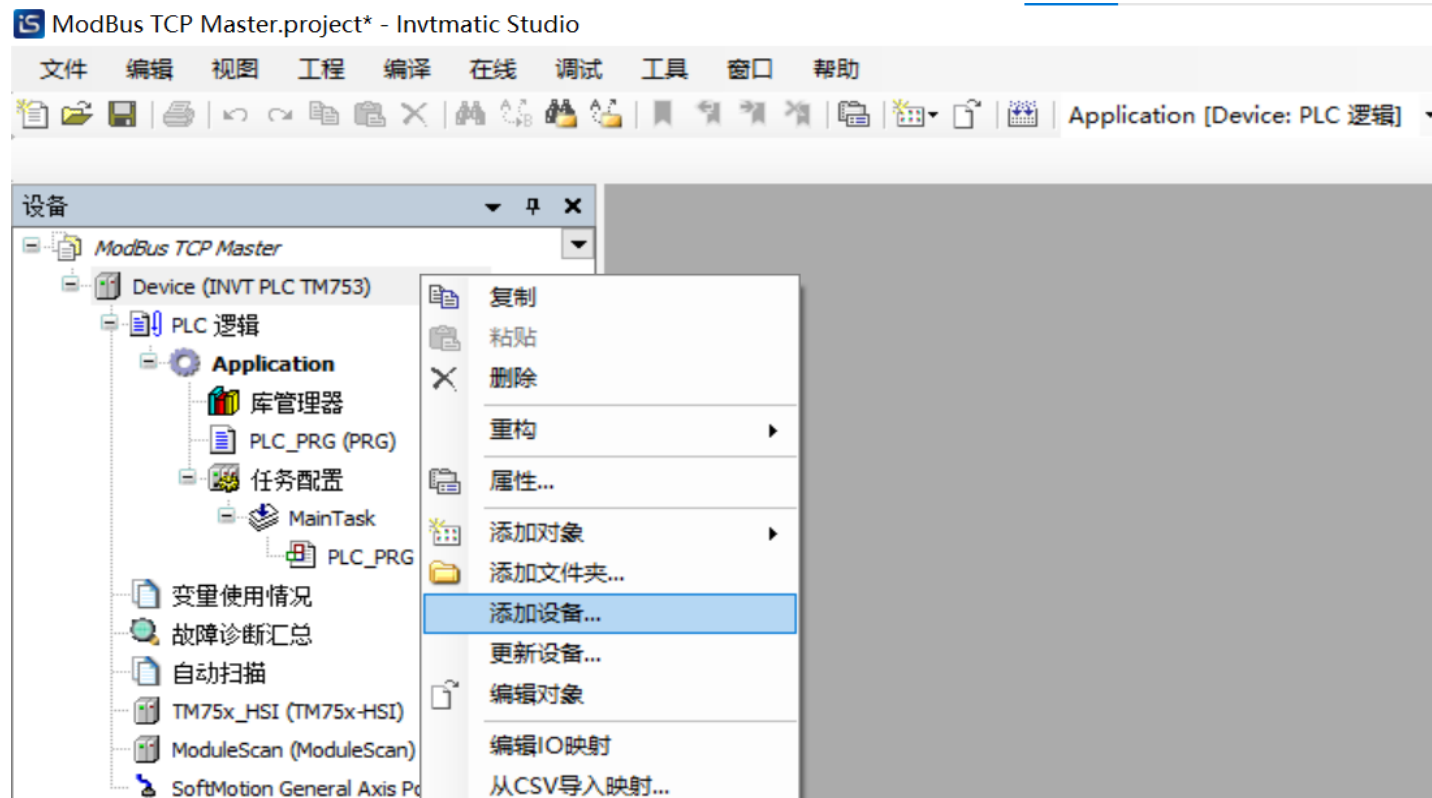
① TM753作主站 (Master) , 电脑作从站 (Slave) ;

② 主站 (Master) IP: 192.168.1.10;

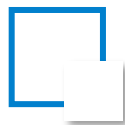
③ 从站 (Slave) IP: 192.168.1.123。



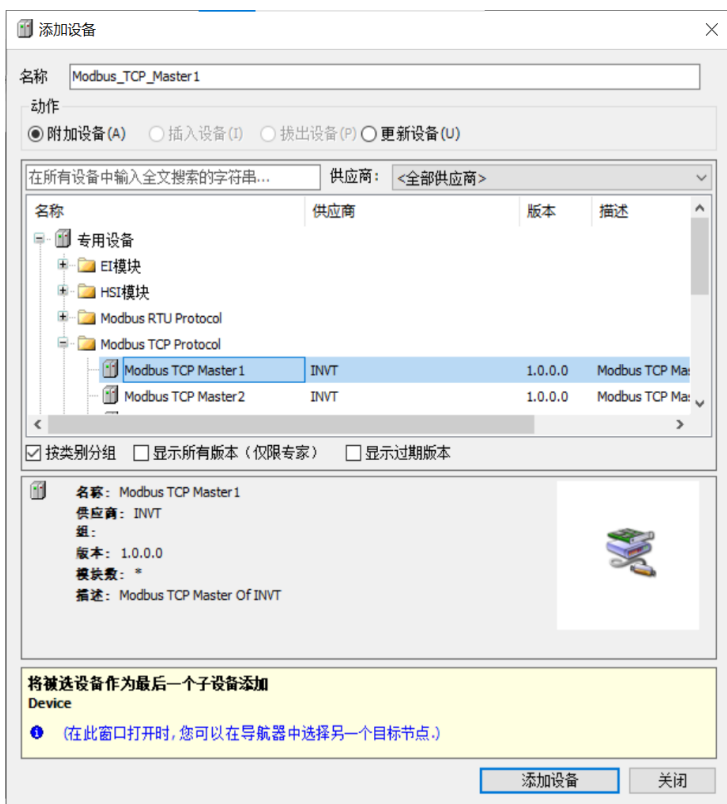
TM753做MODBUS主站，电脑做从站设置：



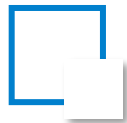
1.在设备树中右键点击“Device”，点击“添加设备”，在专用设备中选择“ModBus TCP Master 1”，点击“添加设备”；



TM753做MODBUS主站，电脑做从站设置：



2.在专用设备中选择“ModBus TCP Master 1”，点击“添加设备”。

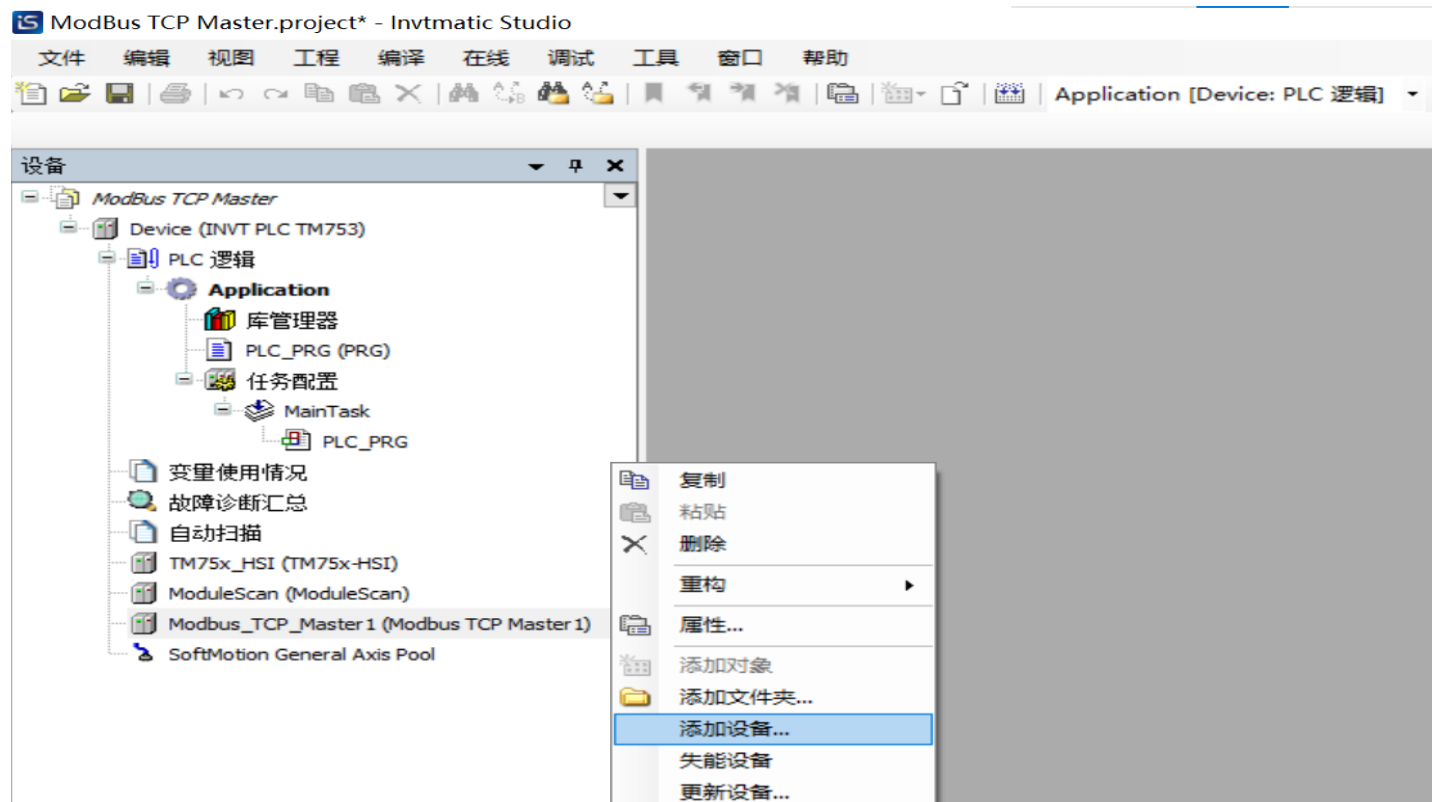


MODBUS TCP 主站

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS主站，电脑做从站设置：



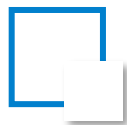
3.在设备树中右键点击
“ModBus_TCP_Master 1”，点
击“添加设备”。



TM753做MODBUS主站，电脑做从站设置：



4. 在弹出窗口中选择“ModBus TCP Slave 1”，点击“添加设备”。

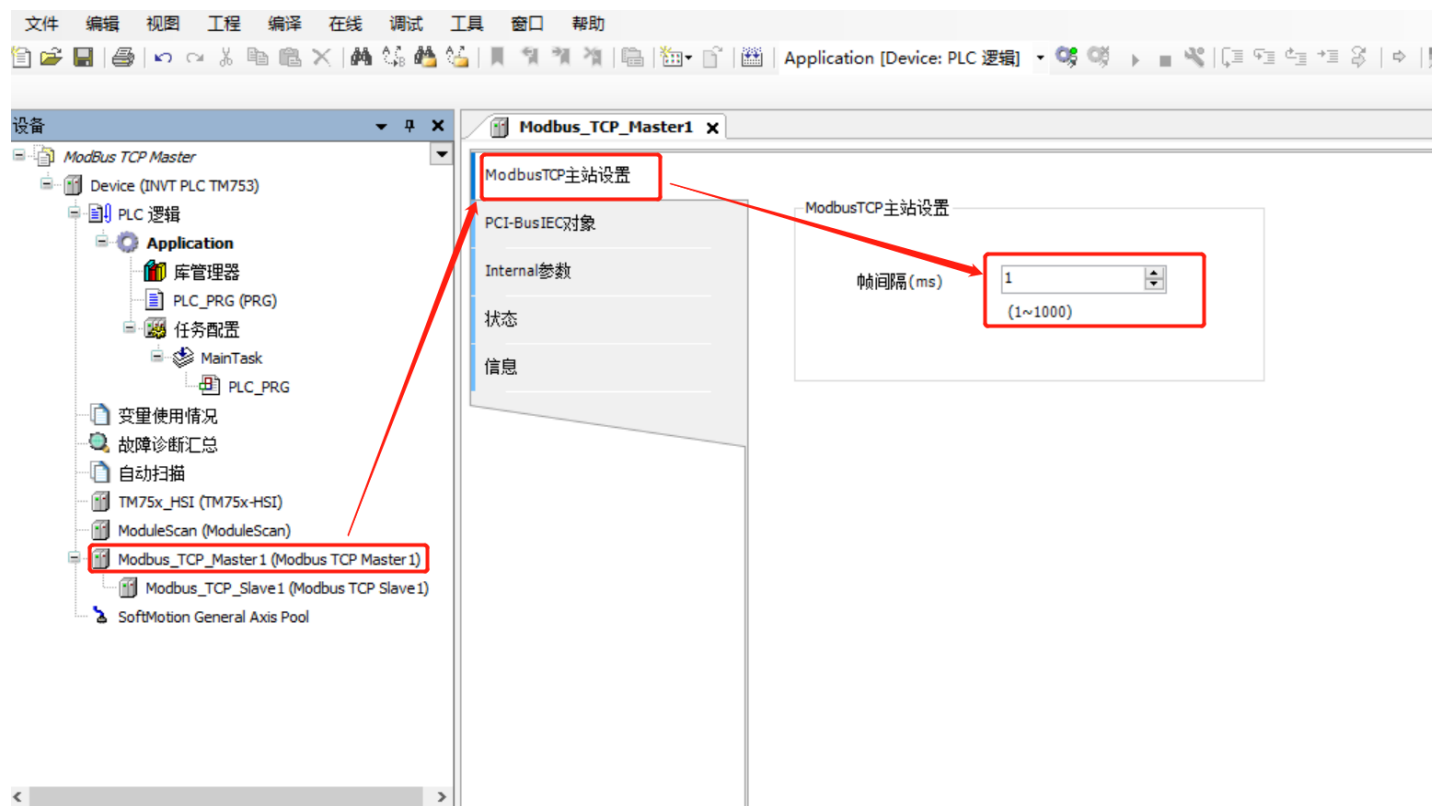


MODBUS TCP 主站

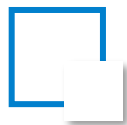
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



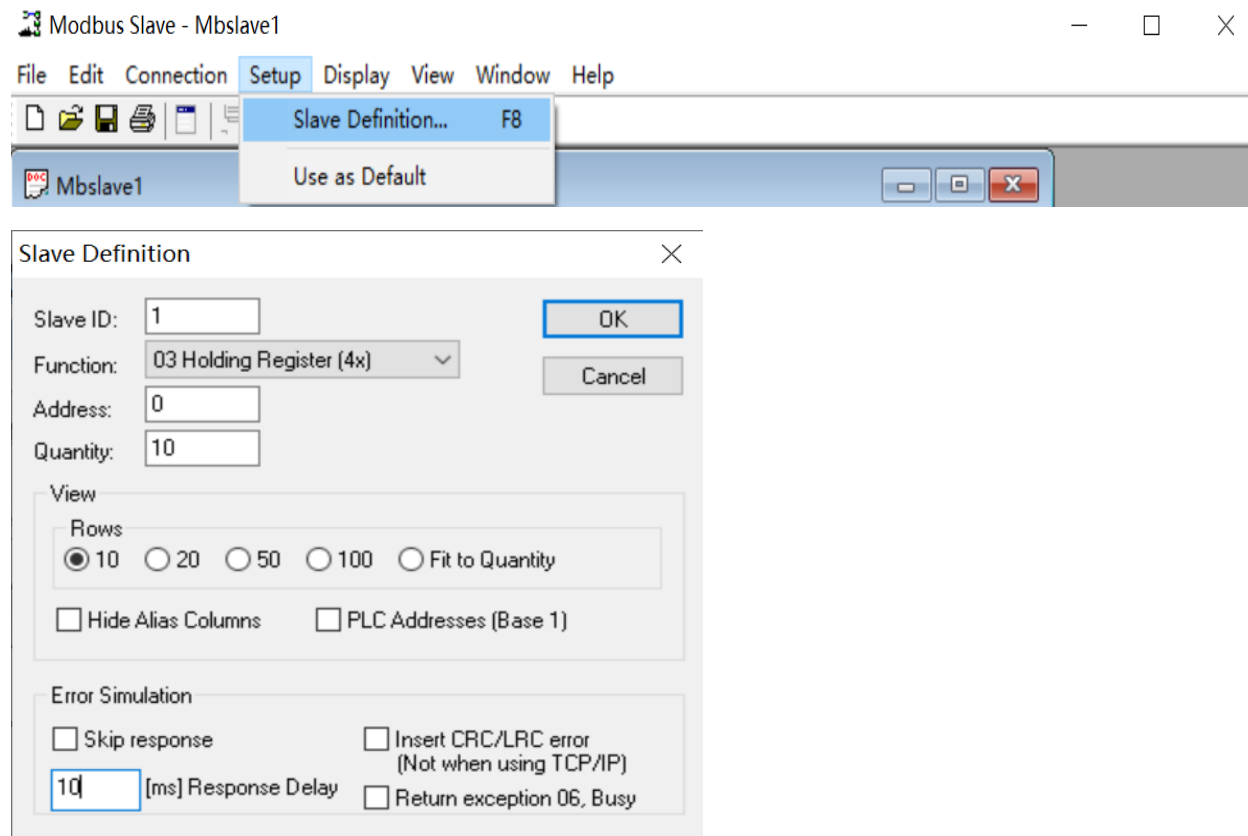
TM753做MODBUS主站，电脑做从站设置：



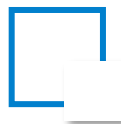
5.双击“ModBus_TCP_Master 1”，点击“ModBus TCP主站设置”，选用默认设置即可；



TM753做MODBUS主站，电脑做从站设置：



6.双击“Modbus Slave”打开调试助手软件；设置从站参数，左键单击“Setup”，“Slave Definition”进入设置界面；设置从站参数，“Slave ID”设置为1，“Function”设置为03 Holding Resgister(4x)，“Address”设置为0，“Quantity”设置为10，“Response Delay”设置为10；

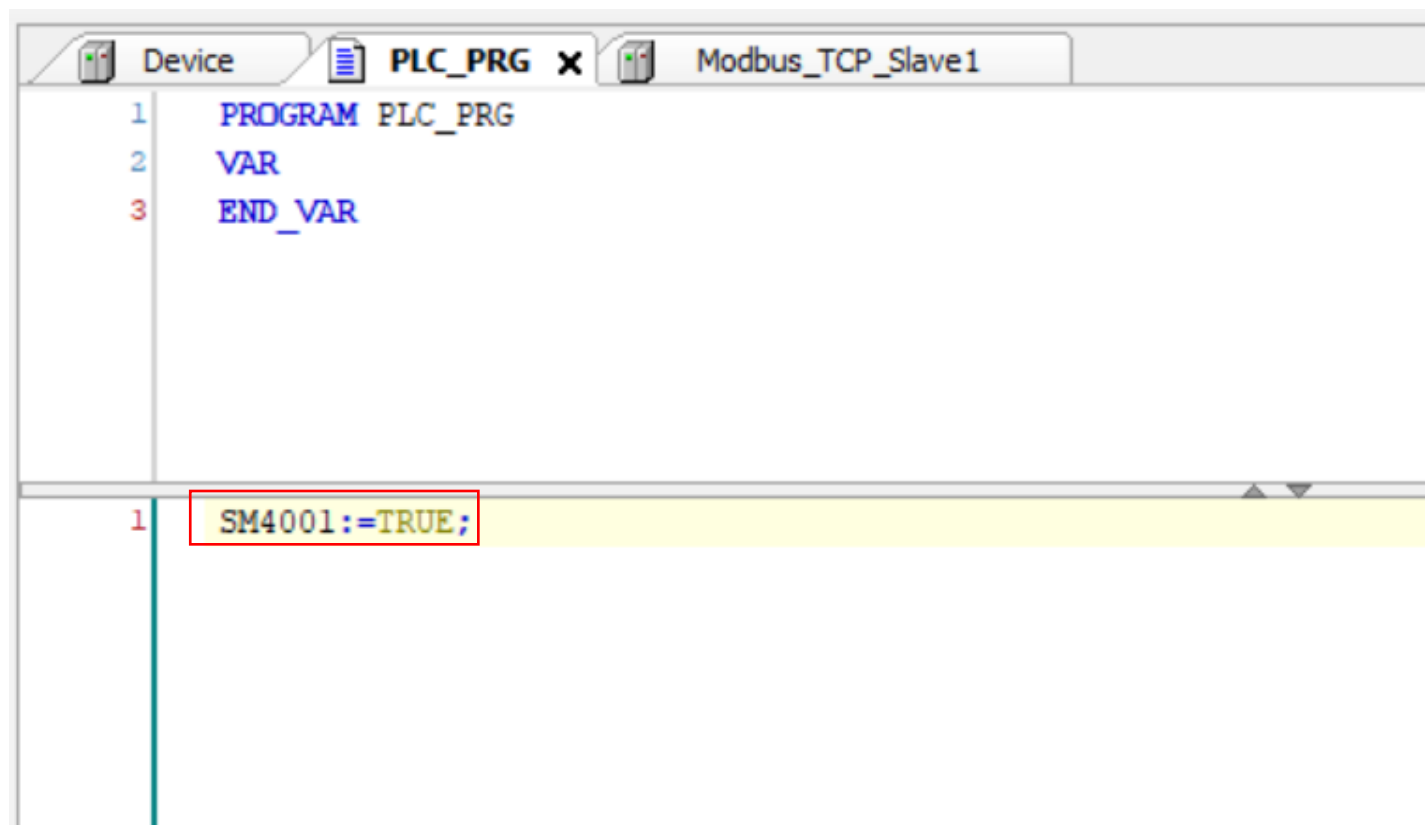


MODBUS TCP 主站

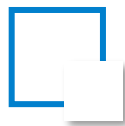
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS主站，电脑做从站设置：



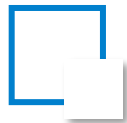
8.在程序中，设置从站使能变量为“TRUE”；



TM753做MODBUS主站，电脑做从站设置：



9.点击“ModBusTCP通信设置”，左键单击【添加】，进入通信设置界面。



TM753做MODBUS主站，电脑做从站设置：

Modbus通信设置

基础配置

名称: Channel 00

功能码: (0x03)读保持寄存器

使能类型: 循环执行

循环时间(ms): 100 (1~65535)

使能变量(SM): 4101 (0~7999)

重试次数: 3

注释:

读取

地址(Hex): 0 (0~0xFFFF)

长度(WORD): 10 (1~125)

错误处理: 保持最后的值

写入

地址(Hex): 0

长度(WORD): 1

确定 取消

10. “名称”采用默认 channel 00，“功能码”设置为 (0x03) 读保持寄存器，“使能类型”设置为循环，“循环时间”设置为100ms，“重试次数”设置为3，“地址”设置为0，“长度”设置为10，“错误处理”设置为保持最后的值，点击“确定”；

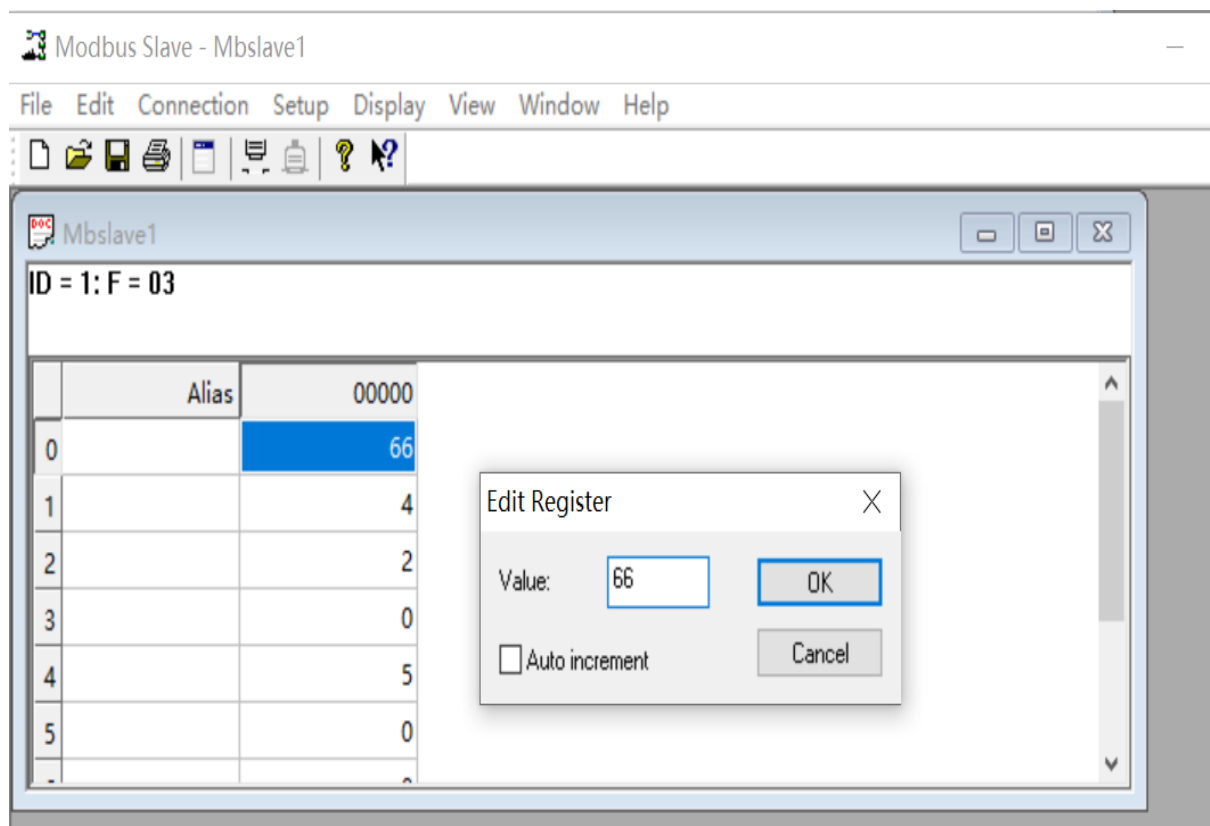


MODBUS TCP 主站

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



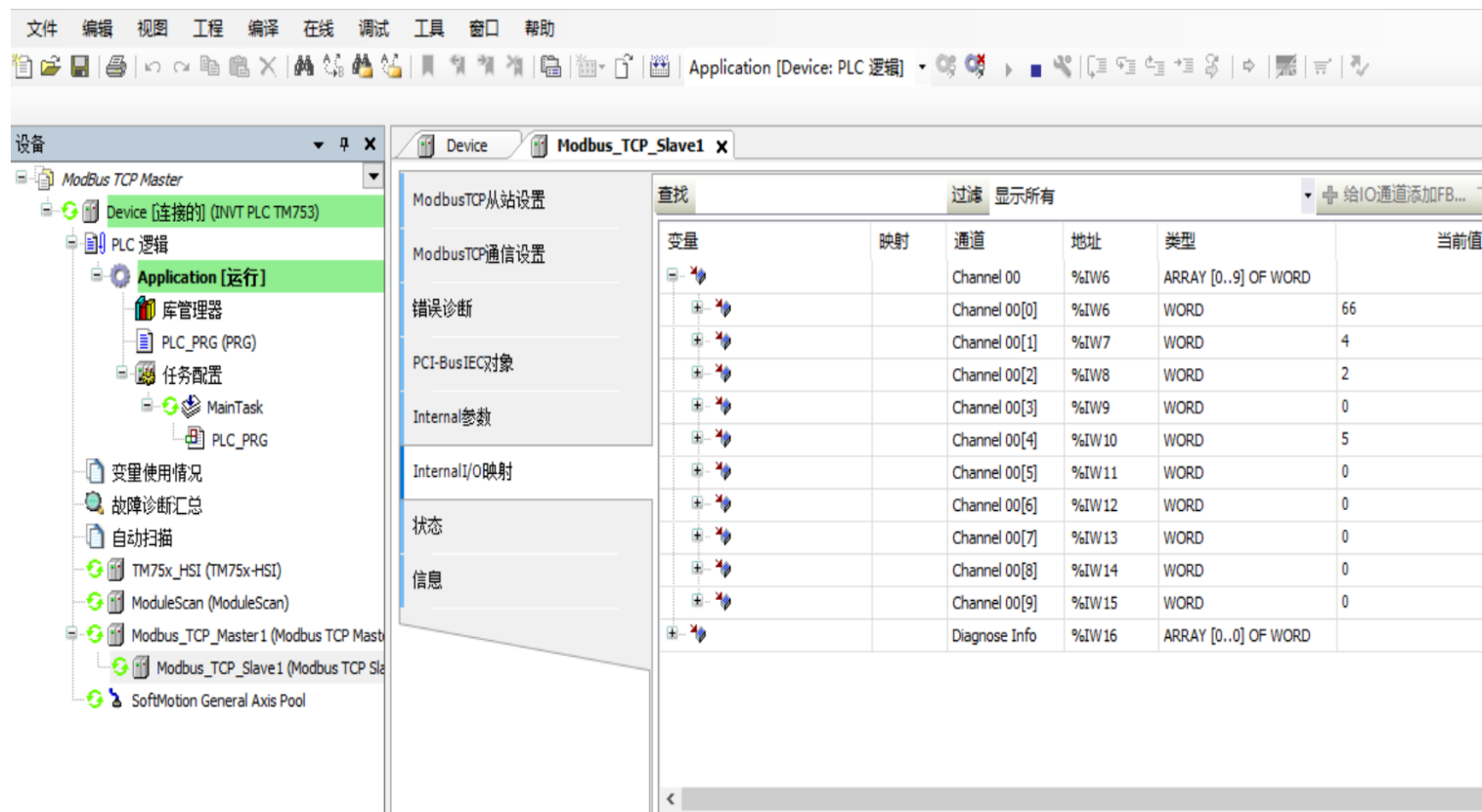
TM753做MODBUS主站，电脑做从站设置：



11.从站修改数值。



TM753做MODBUS主站，电脑做从站设置：



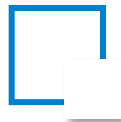
12.双击

“ModbusTCP_SubSlave”，点击“InternalI/O 映射”，可读取从站寄存器数值，则主站读取从站数值成功。（注意：在进行ModBus TCP通讯时，需要关闭掉电脑的防火墙，否则可能会报错）。

PART 02

通讯示例

- TM753与VS_102QS 通讯（串口）
- TM753与VS_102QS通讯（网口标签通讯）
- **MODBUS TCP通讯（从站）**
- MODBUS RUT主站控制变频器
- TCP IP自由口协议



TM753做MODBUS TCP从站，电脑做主站设置准备

Modbus 协议是主从交互式协议，即主机轮询、从机应答。在工控自动化中应用广泛，可基于 RS232/RS485/网口等硬件接口基础上应用。该协议的报文结构主要由地址、功能码、寄存器、数据、校验码等构成。

以下是基于网口的 Modbus TCP 主从站通信示例。

1)功能运行前的准备(PLC 做从站)：1个控制器，1 台电脑，电脑安装 ModBus Poll 调试 软件；

① TM753作从站（Slave），电脑作主站（Master）；

② 主站（Master）IP：192.168.1.123；

③ 从站（Slave）IP：192.168.1.10。

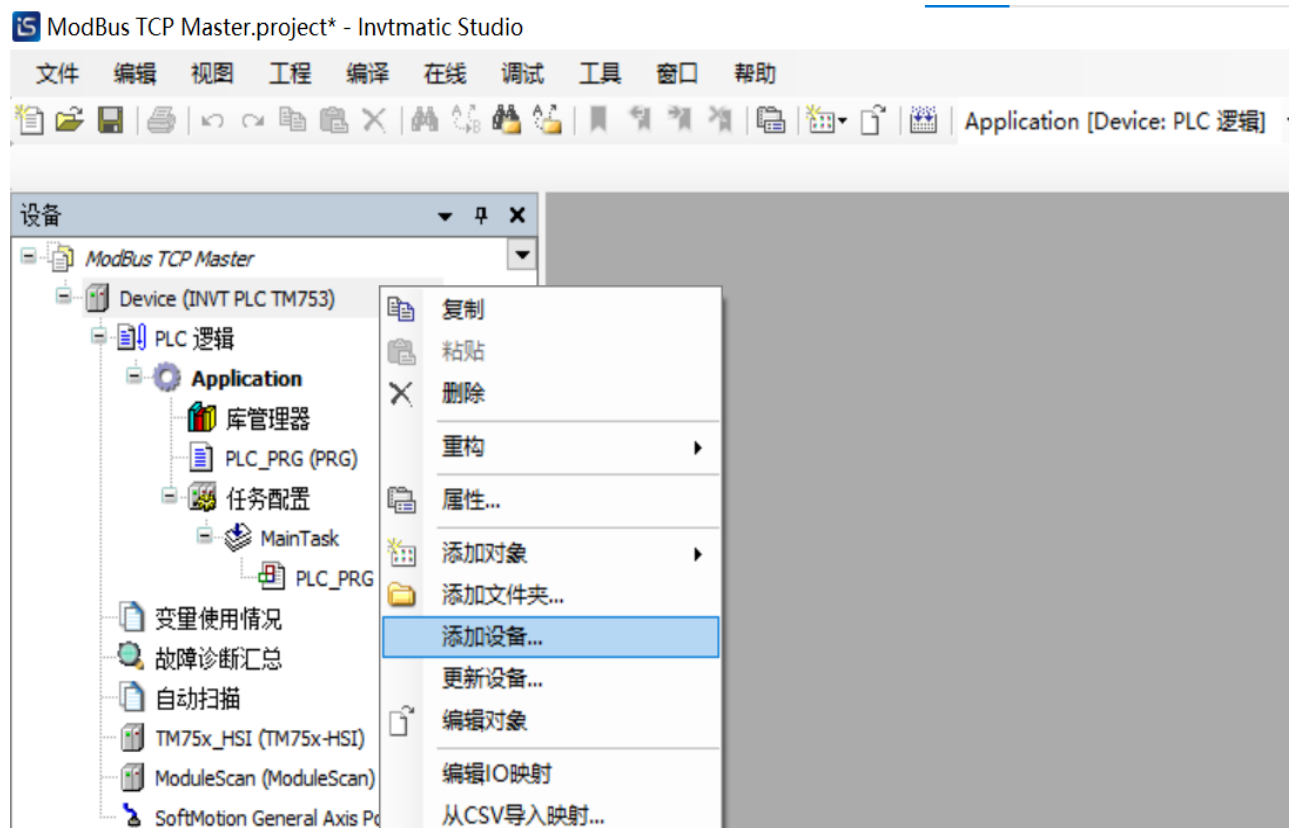


MODBUS TCP 从站

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS TCP从站，电脑做主站设置：



1.在设备树中右键点击“Device”，
点击“添加设备”。

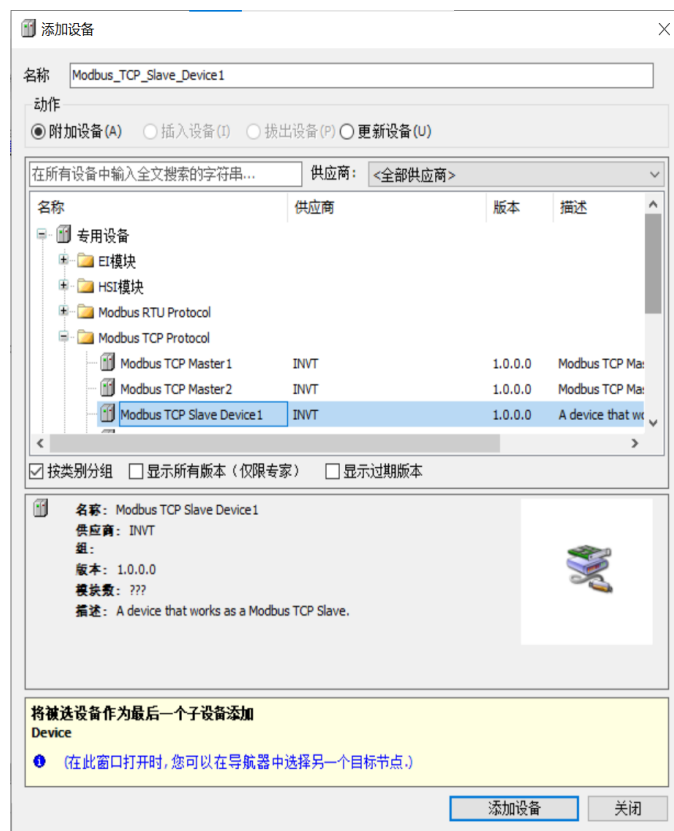


MODBUS TCP 从站

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



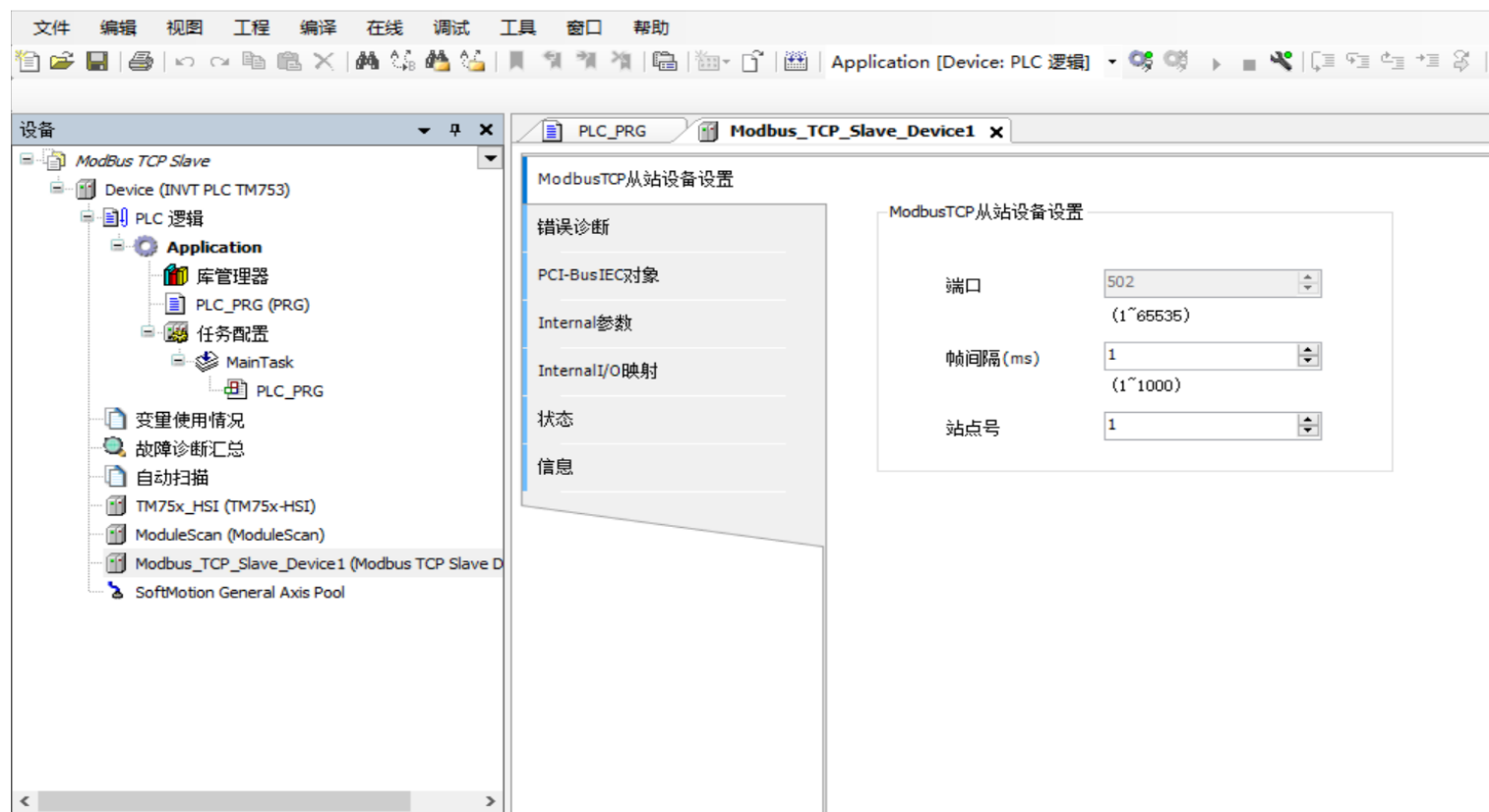
TM753做MODBUS TCP从站，电脑做主站设置：



2. 在专用设备中选择“ModBus TCP Master 1”，点击“添加设备”。

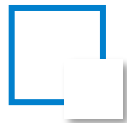


TM753做MODBUS TCP从站，电脑做主站设置：



3.双击

“Modbus_TCP_Slave_Device1”，点击“ModbusTCP从站设备设置”进入从站 参数设置界面；
“帧间隔”设置为100，“站点号”设置为1。

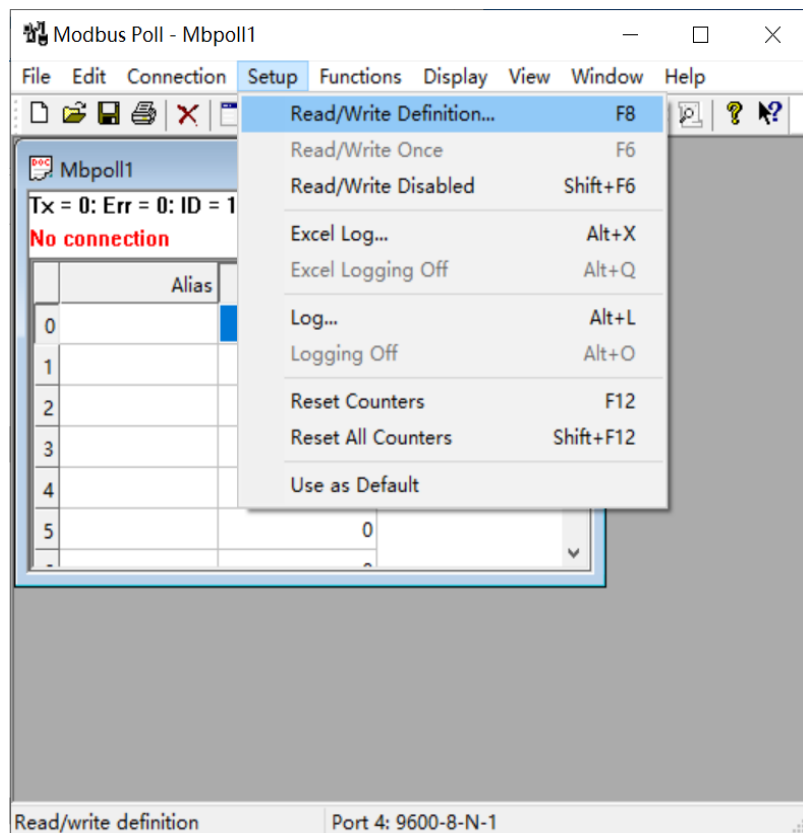


MODBUS TCP 从站

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能

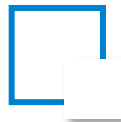


TM753做MODBUS TCP从站，电脑做主站设置：

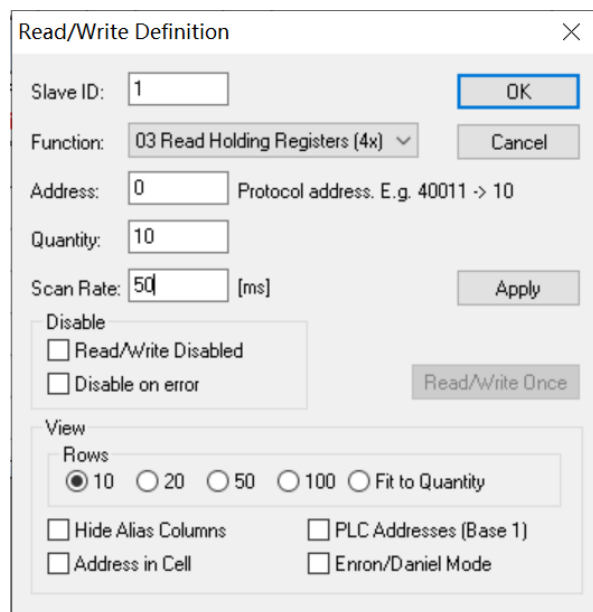


4.设置主站

- ① 双击“Modbus Poll”打开调试助手软件；
- ② 数据读写设置，左键单击“Setup”，点击“Read/Write Definition”进入设置界面；



TM753做MODBUS TCP从站，电脑做主站设置：



The image shows a 'Read/Write Definition' dialog box with the following settings:

- Slave ID: 1
- Function: 03 Read Holding Registers (4x)
- Address: 0 (Protocol address. E.g. 40011 -> 10)
- Quantity: 10
- Scan Rate: 50 [ms]
- Disable: ☐ Read/Write Disabled, ☐ Disable on error
- View: Rows (10 selected, 20, 50, 100, Fit to Quantity)
- Hide Alias Columns: ☐ PLC Addresses (Base 1): ☐ Address in Cell: ☐ Enron/Daniel Mode: ☐

Buttons: OK, Cancel, Apply, Read/Write Once

5. “Slave ID” 设置为1，
“Function” 设置为03 Read Holding Resgister(4x)，
“Address” 设置为0，“Quantity” 设置为 10，“Scan Rate” 设置为 20，点击 “OK”

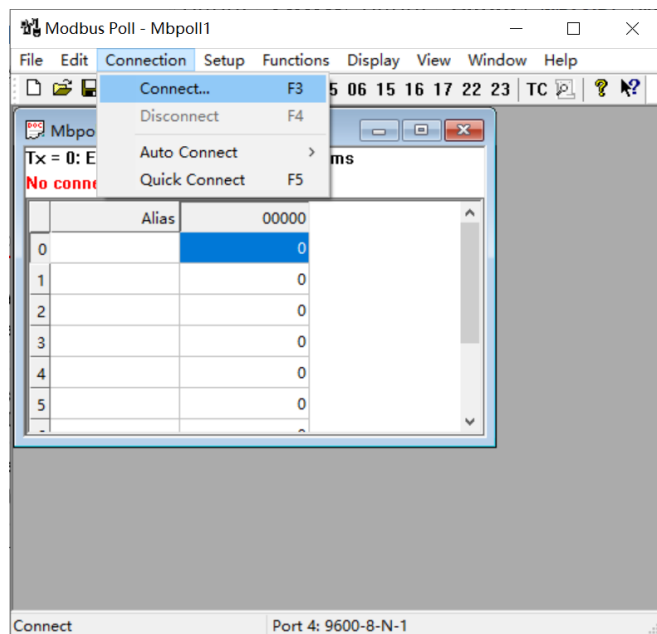


MODBUS TCP 从站

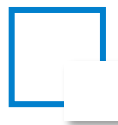
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS TCP从站，电脑做主站设置：



6. 点击“Connection”，点击“Connection”，进入连接设置界面；

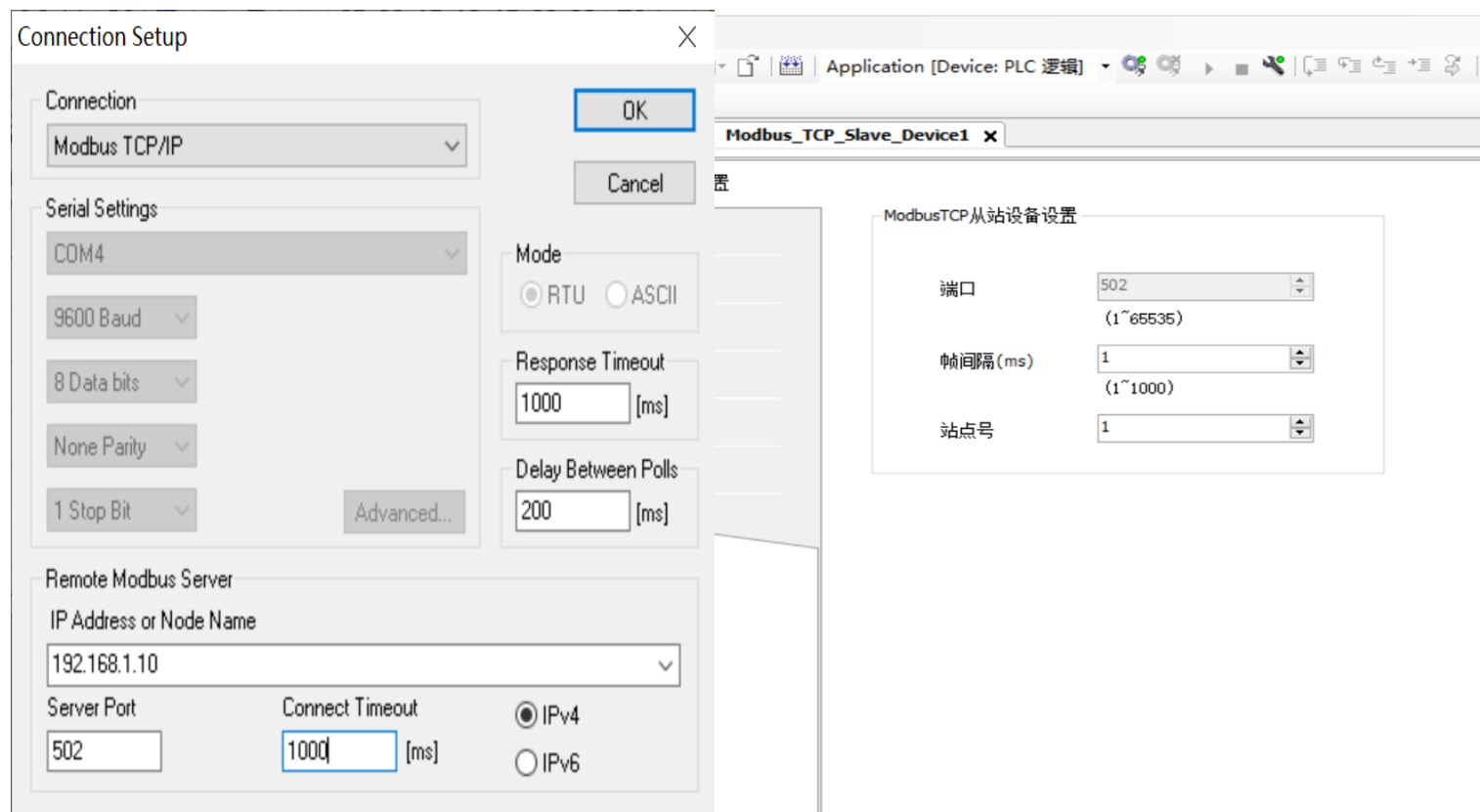


MODBUS TCP 从站

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS TCP从站，电脑做主站设置：



7. “Connection” 设置为ModBus TCP/IP, “IP Address or Node Name” 设置为192.168.1.10, “Server Port” 设置为502, “Connect Timeout” 设置为1000, 点击 “OK”



TM753做MODBUS TCP从站，电脑做主站设置：

```
1 PROGRAM PLC_PRG
2 VAR
3   ModBusTcp_Test AT %MW0 : ARRAY [0..10] OF WORD;
4 END_VAR
```

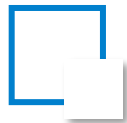
8.创建如图所示变量及绑定地址。



TM753做MODBUS TCP从站，电脑做主站设置：

表达式	类型	值	准备值	地址
ModBusTcp_Test	ARRAY [0..10] OF ...			%MW0
ModBusTcp_Test[0]	WORD	1		%MB0
ModBusTcp_Test[1]	WORD	2		%MB2
ModBusTcp_Test[2]	WORD	3		%MB4
ModBusTcp_Test[3]	WORD	4		%MB6
ModBusTcp_Test[4]	WORD	5		%MB8
ModBusTcp_Test[5]	WORD	6		%MB10
ModBusTcp_Test[6]	WORD	7		%MB12
ModBusTcp_Test[7]	WORD	8		%MB14
ModBusTcp_Test[8]	WORD	9		%MB16
ModBusTcp_Test[9]	WORD	10		%MB18
ModBusTcp_Test[10]	WORD	0		%MB20

9.从站对寄存器赋值。

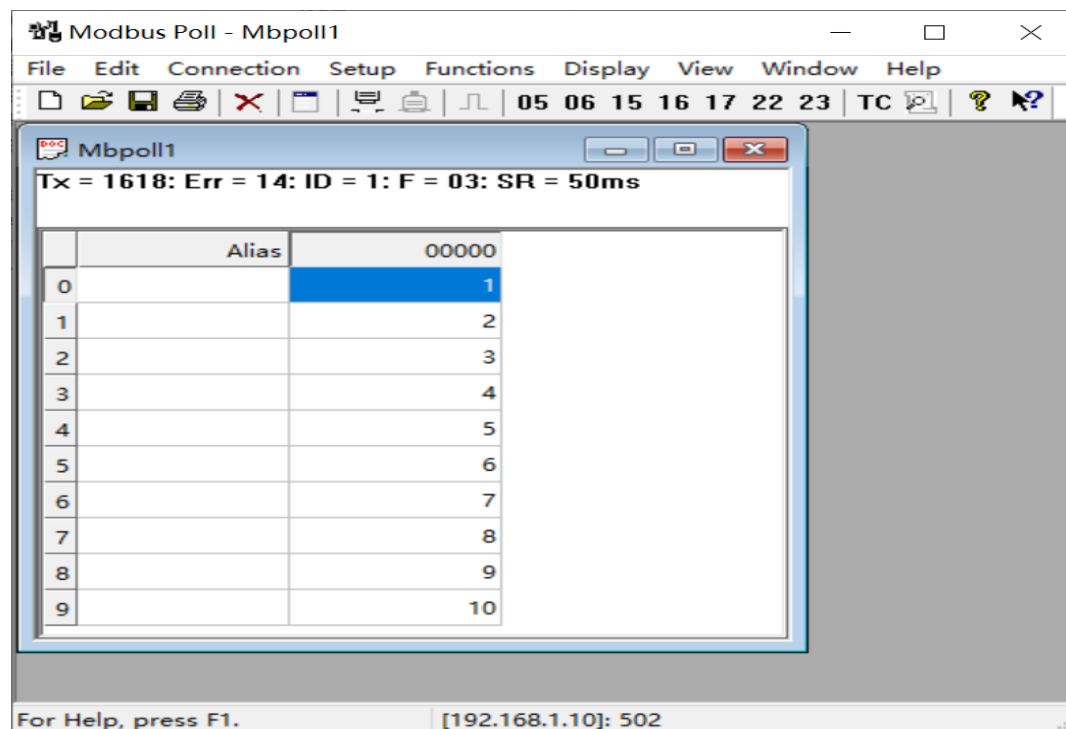


MODBUS TCP 从站

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS TCP从站，电脑做主站设置：



10.主站读取成功

PART 02

通讯示例

- TM753与VS_102QS 通讯（串口）
- TM753与VS_102QS通讯（网口标签通讯）
- MODBUS TCP通讯（从站）
- **MODBUS RTU主站控制变频器**
- TCP IP自由口协议

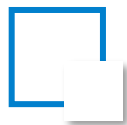


TM753做MODBUS 主站，GD20变频器做Modbus 从站

1.利用电脑上的HMITOOL软件做触摸屏，与TM753 PLC相连（触摸屏做主站，PLC做从站）；
利用TM753 PLC与GD20变频器相连（PLC做主站，变频器做从站）。



1.PLC硬件接线接A1,B1。



MODBUS RTU主站控制变频器

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站



2.变频器接线，485+,485-。

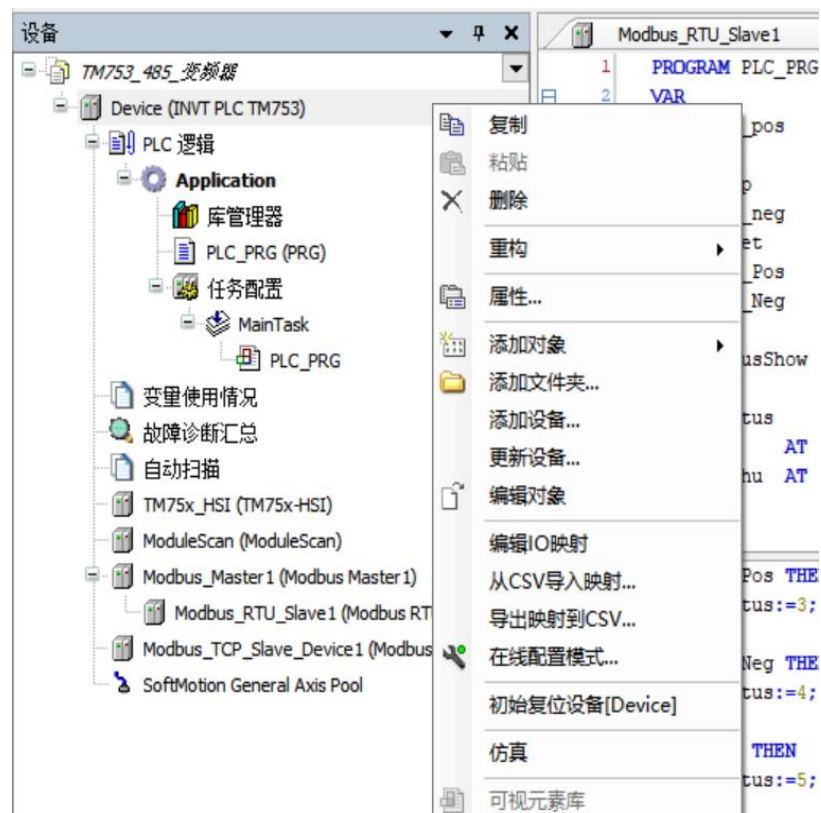


MODBUS RTU主站控制变频器

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站



3.触摸屏与PLC通过网口相连（TCP通讯），TM753 PLC做从站，需要在invtmatic Studio程序中进行配置,详细步骤如下：
在左侧设备树中右键点击“Device”，点击“添加设备”。



MODBUS RTU主站控制变频器

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站



4.在弹出的窗口中点击“专用设备”，点击“Modbus TCP Slave Device1”，点击“添加设备”。

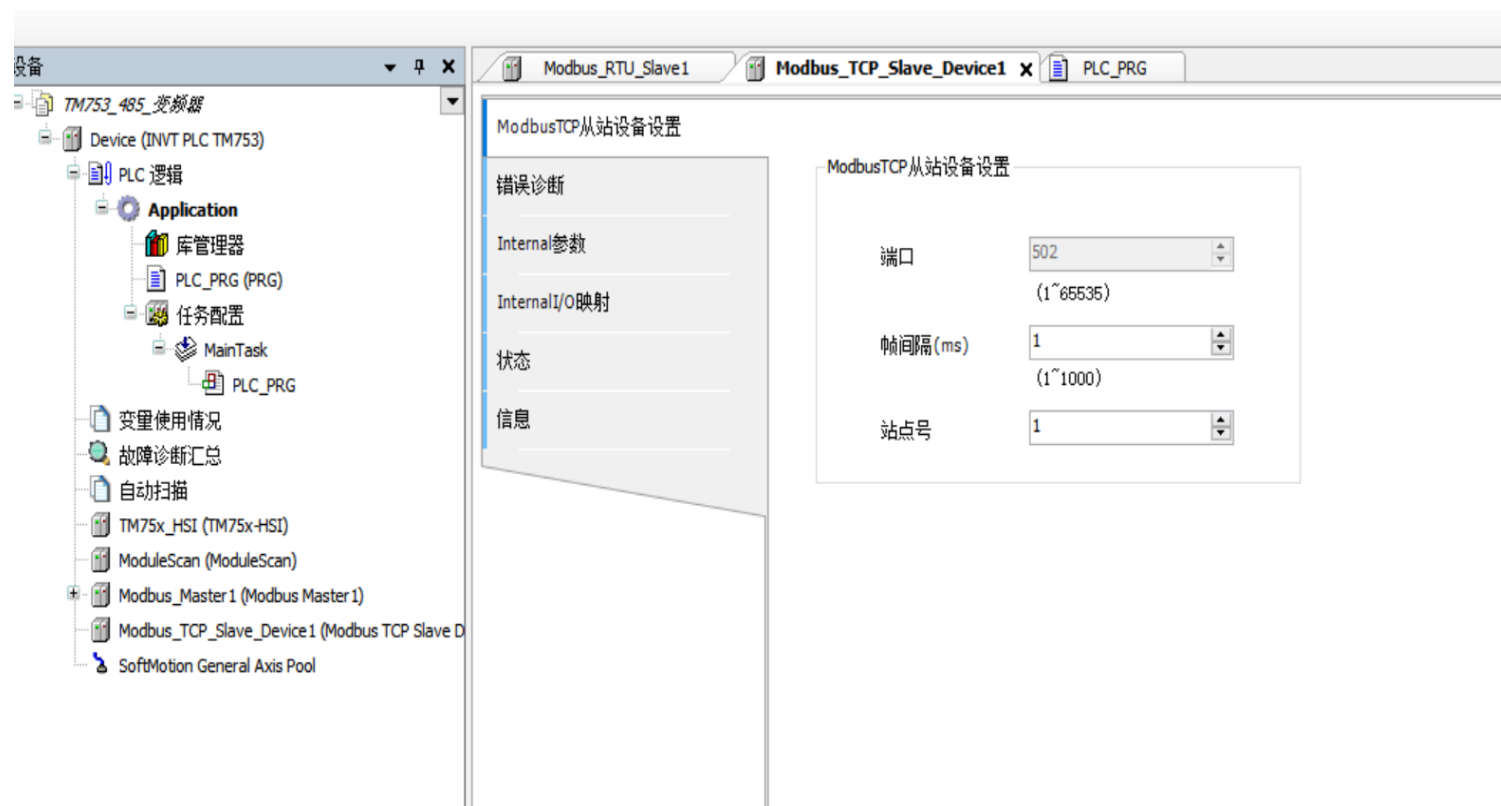


MODBUS RTU主站控制变频器

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能

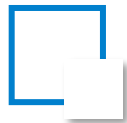


TM753做MODBUS 主站，GD20变频器做Modbus 从站



5.双击

“Modbus_TCP_Slave_Device1”，点击ModBusTCP从站设备设置，配置站点号。

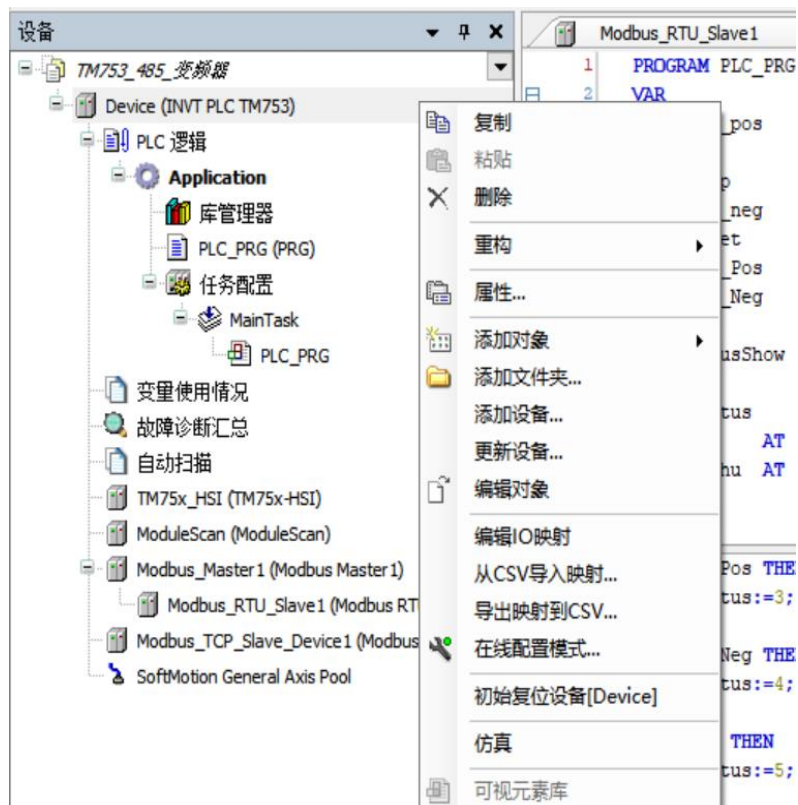


MODBUS RTU主站控制变频器

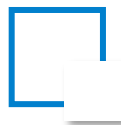
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站



6. PLC与变频器相连（485通讯），TM753做主站，需要在invtmatic Studio程序中进行配置，并设置485通讯的相关参数（波特率、奇偶校验、停止位等）详细步骤如下：
在左侧设备树中右键点击“Device”，点击“添加设备”。



MODBUS RTU主站控制变频器

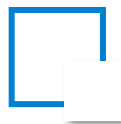
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站

7.在弹出的窗口中选择“专用设备”，选择“Modbus RTU Protocol”，选择“Modbus Master 1”，点击“添加设备”；。



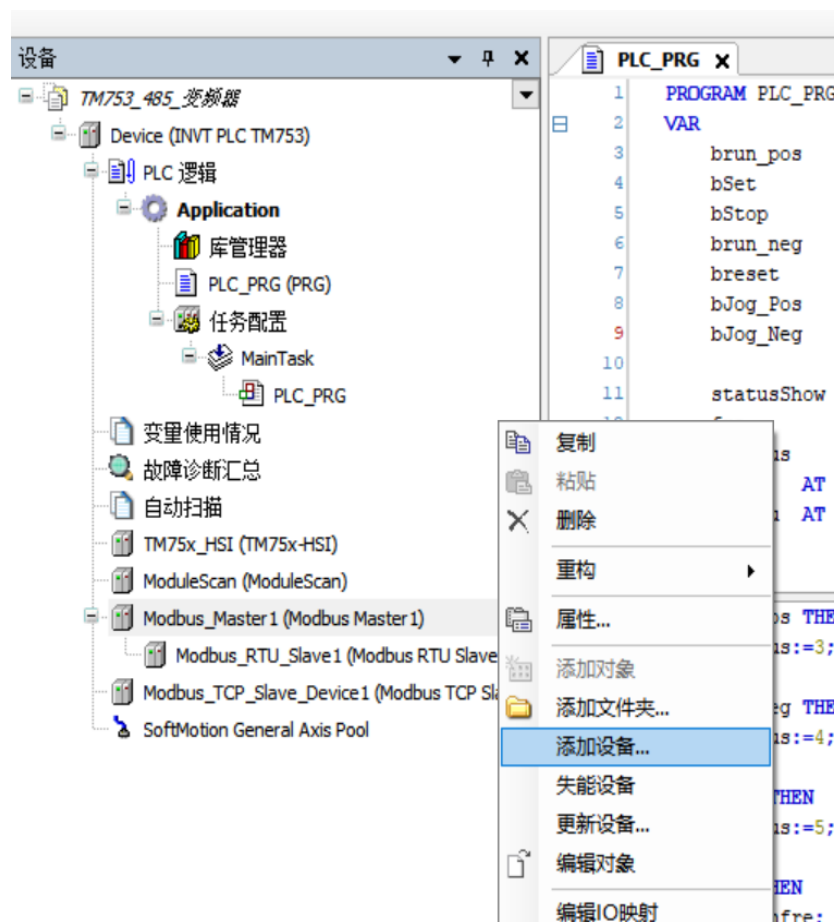


MODBUS RTU主站控制变频器

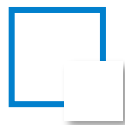
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站



8.右键点击“Modbus Master 1”，
点击“添加设备”。



MODBUS RTU主站控制变频器

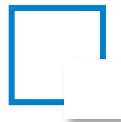
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站

9.在弹出的窗口中选择“Modbus RTU Slave1”;点击“添加设备”。



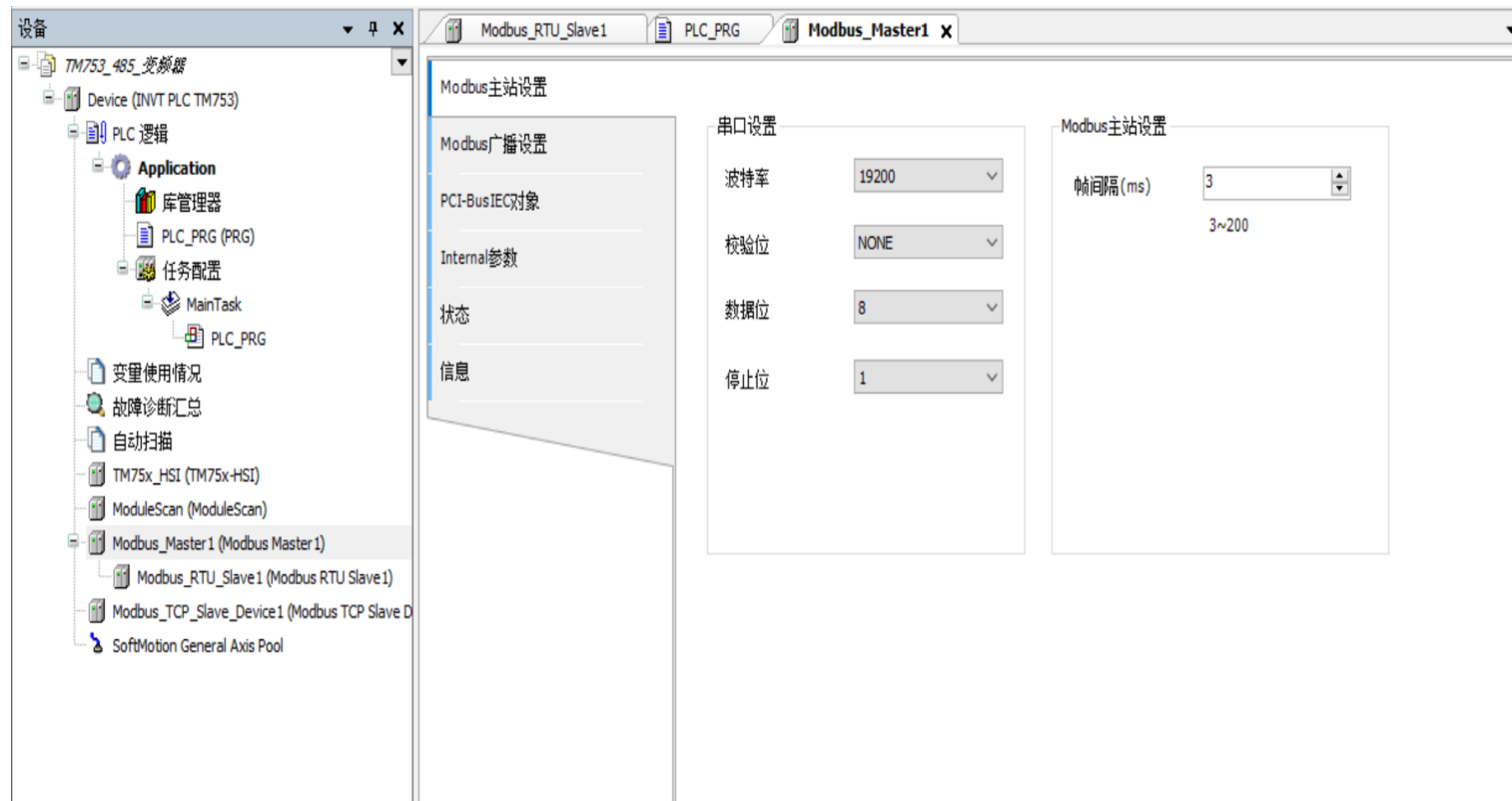


MODBUS RTU主站控制变频器

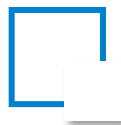
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



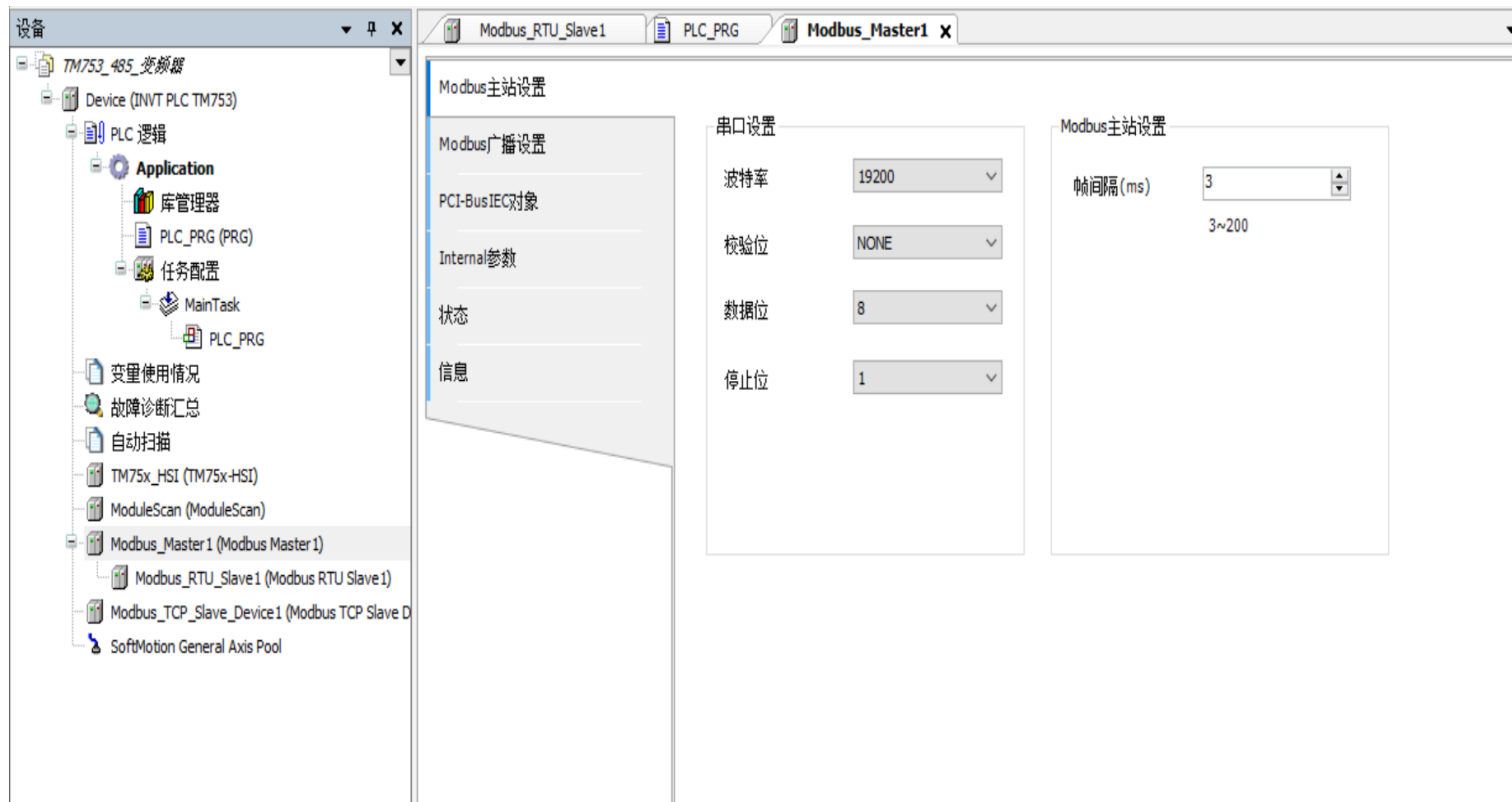
TM753做MODBUS 主站，GD20变频器做Modbus 从站



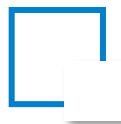
10.双击“Modbus_Master1”，点击“Modbus主站设置”，在“串口设置”中设置波特率、奇偶校验、停止位等参数；



TM753做MODBUS 主站，GD20变频器做Modbus 从站



10.双击“Modbus_Master1”，点击“Modbus主站设置”，在“串口设置”中设置波特率、奇偶校验、停止位等参数；
注意：PLC设置的波特率、奇偶校验、停止位等参数需要与变频器的参数设置相一致；

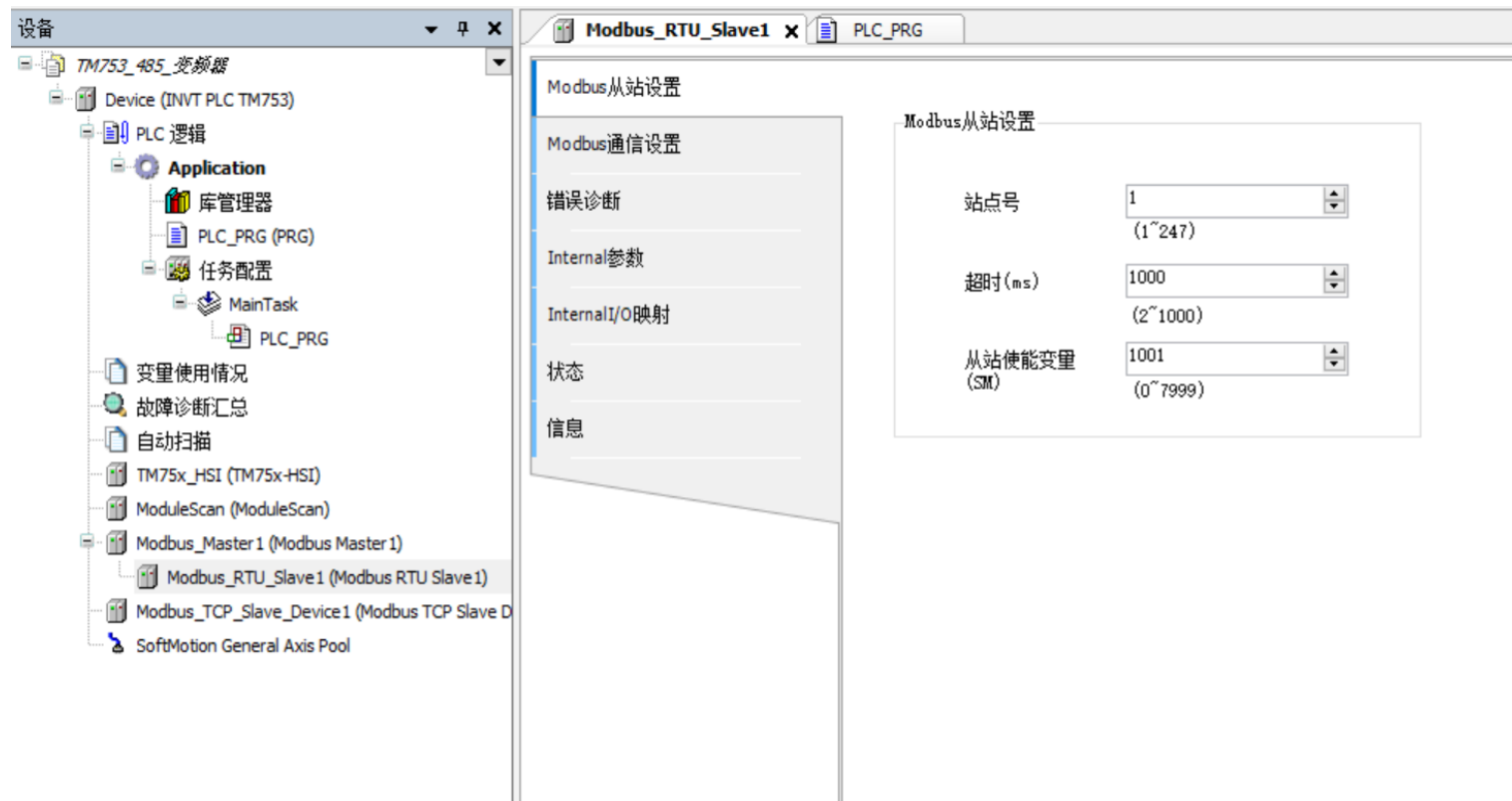


MODBUS RTU主站控制变频器

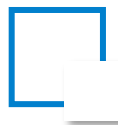
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站



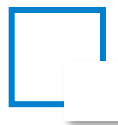
11.双击“Modbus_RTU_Slave1”，点击“Modbus从站设置”，设置Modbus从站的站点号、超时、从站使能变量。



TM753做MODBUS 主站，GD20变频器做Modbus 从站

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3     brun_pos AT %MX0.0 :BOOL; //正转
4     bSet AT %MX0.1 :BOOL; //频率设定启动
5     bStop AT %MX0.2 :BOOL; //停止
6     brun_neg AT %MX0.3 :BOOL; //反转
7     breset AT %MX0.4 :BOOL; //故障复位
8     bJog_Pos AT %MX0.5 :BOOL; //正转点动
9     bJog_Neg AT %MX0.6 :BOOL; //反转点动
10
11     statusShow AT %MW100 :INT; //运行状态显示
12     fre :INT; //频率
13     bstatus :INT; //运行状态
14     hfre AT %MW10 :INT; //触摸屏频率值设定
15     canshu AT %MW200 : ARRAY[1..6] OF INT;
16 END_VAR
```

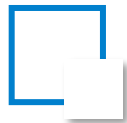
12. TM753 PLC编写程序，并将相关地址绑定到触摸屏变量上，实现变频器的正转、反转等动作；



TM753做MODBUS 主站，GD20变频器做Modbus 从站

```
1  SM1001:=TRUE;  
2  IF brun_pos THEN  
3      bstatus:=1;  
4  END_IF  
5  IF brun_neg THEN  
6      bstatus:=2;  
7  END_IF  
8  IF bJog_Pos THEN  
9      bstatus:=3;  
10 END_IF  
11 IF bJog_Neg THEN  
12     bstatus:=4;  
13 END_IF  
14 IF bStop THEN  
15     bstatus:=5;  
16 END_IF  
17 IF bSet THEN  
18     fre:=hfre;  
19 END_IF  
20
```

12. 需要设置从站使能变量
SM1001为TRUE，才能正常建立
通讯。

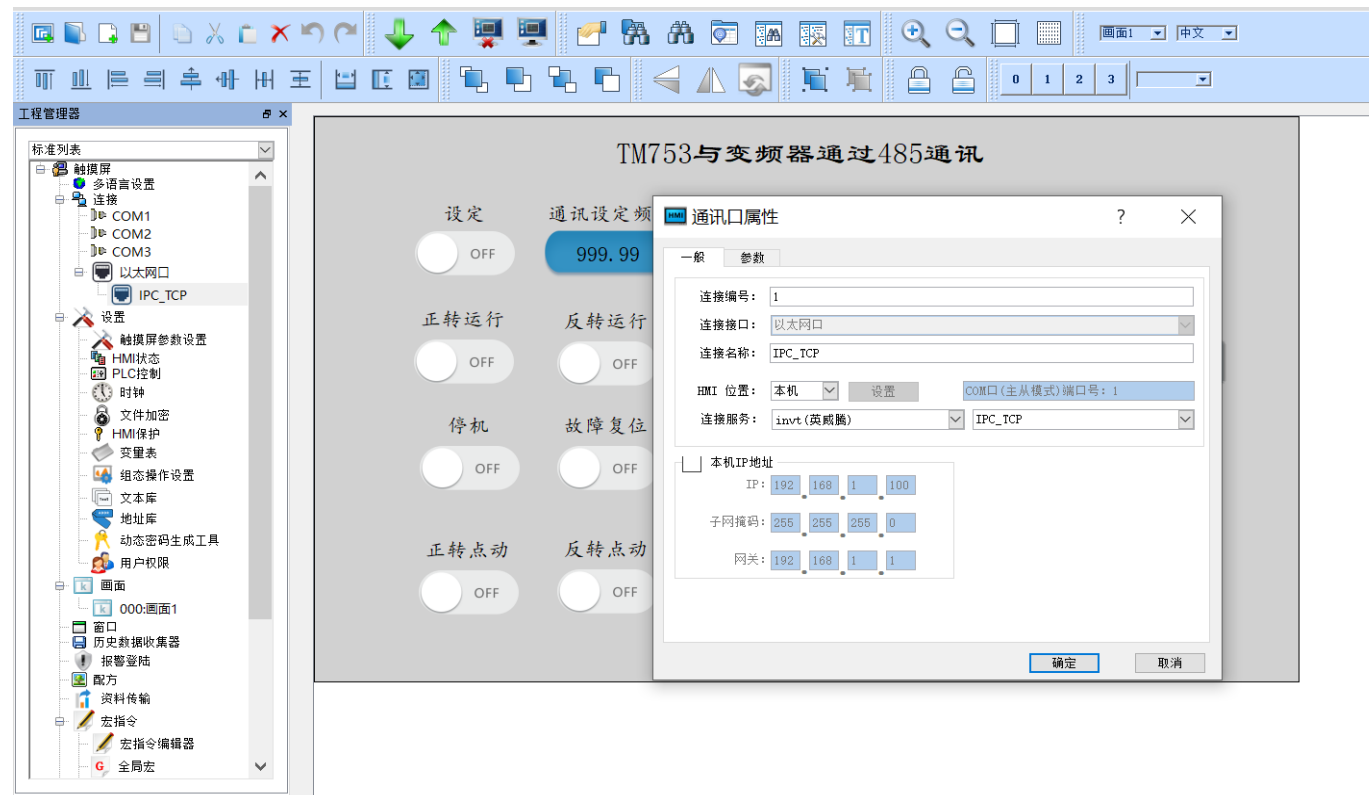


MODBUS RTU主站控制变频器

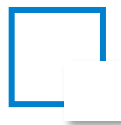
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站



13.使用HMITool，新建工程，在通信连接窗口中的连接接口中选择“以太网口”，设备服务选择“英威腾”，型号选择“IPC_ICP”。



MODBUS RTU主站控制变频器

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站

HMI 通讯口属性

一般 参数

连接设备IP

IP地址: 192 . 168 . 1 . 10

端口号: 502

其他

触摸屏站号: 0

PLC站号: 1

通信时间: 5 (ms)

超时时间1: 1000 (ms)

超时时间2: 5 (ms)

重试次数: 3

地址模式: 标准模式

PLC连续地址间隔: 32

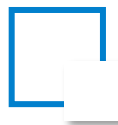
备用参数设定

备用参数1: 0 备用参数3: 0

备用参数2: 0 备用参数4: 0

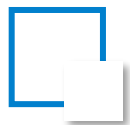
确定 取消

14.双击“IPC_ICP”,在连接服务处选择英威腾,“IPC_ICP”;在参数处将连接设备IP设置好,(TM753控制器默认IP为192.168.1.10),端口号为502;



TM753做MODBUS 主站，GD20变频器做Modbus 从站

15.设置触摸屏控件变量，绑定地址。



MODBUS RTU主站控制变频器

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753做MODBUS 主站，GD20变频器做Modbus 从站

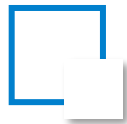


15.触摸屏监控页面。

PART 02

通讯示例

- TM753与VS_102QS 通讯（串口）
- TM753与VS_102QS通讯（网口标签通讯）
- MODBUS TCP通讯（从站）
- MODBUS RTU主站控制变频器
- TCP IP自由口协议（TM753为Client端）



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

运行前准备：

1 台 TM753，1 台电脑，NetAssist 调试软件

- ① TM753 做 Client 端，电脑做 Server 端；
- ② Client IP：192.168.1.10；
- ③ Server IP：192.168.1.123。

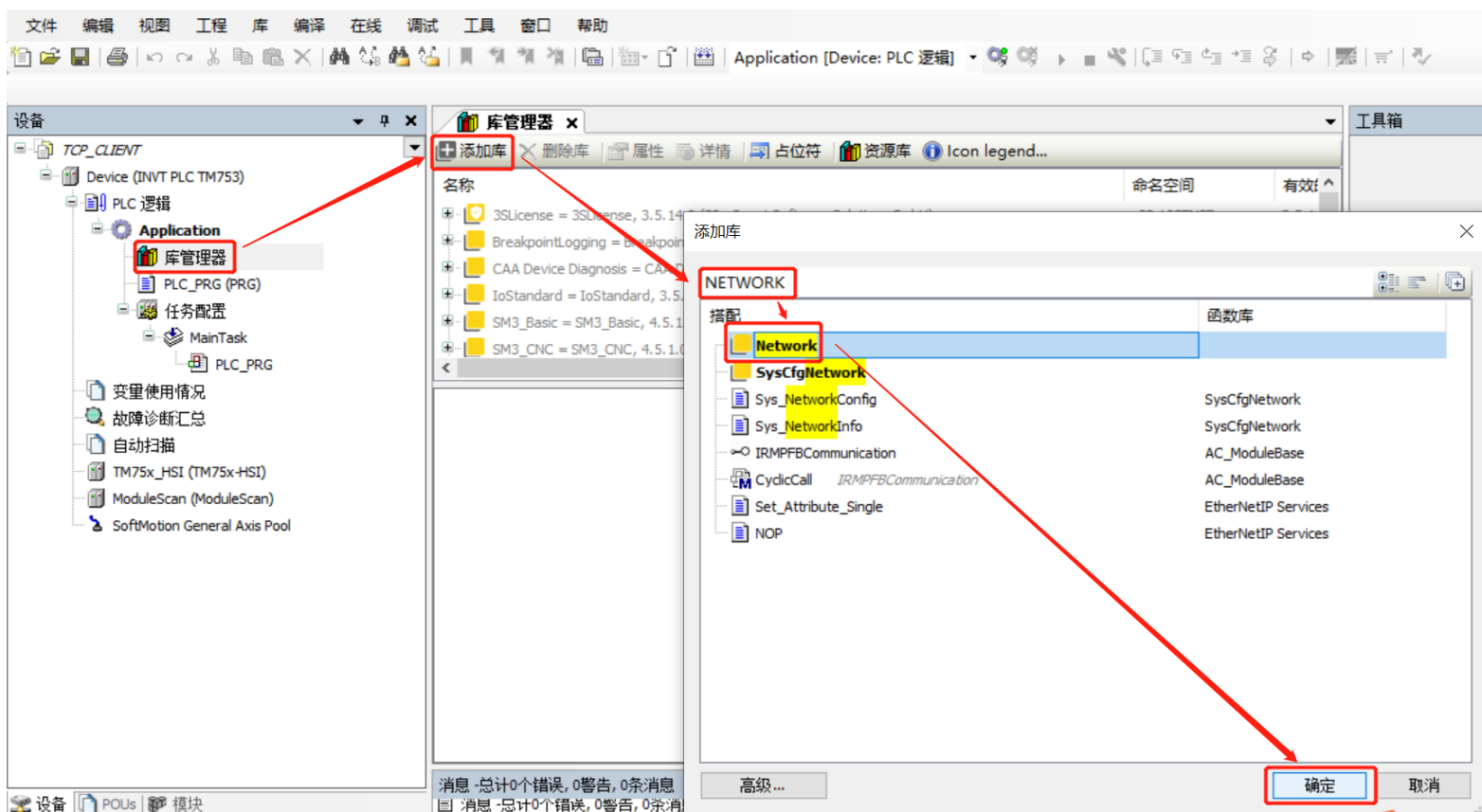


TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端



1.添加库

- ① 双击【库管理器】；
- ② 左键单击【添加库】；
- ③ 在添加库界面的文本框输入所需的库“Network”即可搜索完成；
- ④ 左键单击选择需要的库【Network】后左键单击右下角【确定】按钮即可添加完成；
- ⑤ 【Netwok,3.5.7.0】即为添加的库。



TCP IP自由口协议 (TM753为Client端)

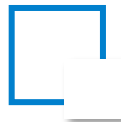
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

```
1  PROGRAM PLC_PRG
2  VAR
3      TCP_Client_0: NBS.TCP_Client;
4      TCP_Write_0: NBS.TCP_Write;
5      TCP_Read_0: NBS.TCP_Read;
6      xTCP_Client_Enable:BOOL;
7      IP:NBS.IP_ADDR;
8      hConn:CAA.HANDLE;
9      xTCP_Write_Exe:BOOL;
10     xTCP_Read_Exe:BOOL;
11     abyWriteData:ARRAY[0..14] OF BYTE;
12     abyReadData:ARRAY[0..14] OF BYTE;
13 END_VAR
```

2.创建变量，并编写程序，如图所示；



TCP IP自由口协议 (TM753为Client端)

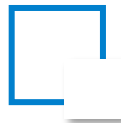
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

```
1  IP.sAddr:='192.168.1.123';//服务端IP地址
2  //PLC作客户端
3  TCP_Client_0(
4      xEnable:= xTCP_Client_Enable ,
5      xDone=> ,
6      xBusy=> ,
7      xError=> ,
8      udiTimeOut:=500000 ,
9      ipAddr:=IP ,
10     uiPort:= 8080,
11     eError=> ,
12     xActive=> ,
13     hConnection:=hConn );
14     //TCP自由协议Client端往Server端写参数
15 TCP_Write_0(
16     xExecute:= xTCP_Write_Exe AND TCP_Client_0.xBusy,
17     udiTimeOut:= 500000,
18     xDone=> ,
19     xBusy=> ,
20     xError=> ,
21     hConnection:= hConn,
22     szSize:= 15,
23     pData:= ADR(abyWriteData),
24     eError=> );
```

2.创建变量，并编写程序，如图所示；



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

```
25 //TCP自由协议Client端读取Server端参数
26 TCP_Read_0(
27     xEnable:=xTCP_Read_Exe AND TCP_Client_0.xBusy ,
28     xDone=> ,
29     xBusy=> ,
30     xError=> ,
31     hConnection:= hConn,
32     szSize:= 15,
33     pData:=ADR(abyReadData) ,
34     eError=> ,
35     xReady=> ,
36     szCount=> );
```

100 %

2.创建变量，并编写程序，如图所示；



TCP IP自由口协议 (TM753为Client端)

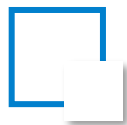
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端



3.设置 Client 端参数及获取通信句柄
① 打开 NetAssist 网络调试助手软件，“协议类型”为 TCP Server，“本地 IP 地址”为 192.168.1.123，“本地端口”为 8080，设置好参数后左键单击“打开”；



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能

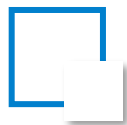


TM753 做 Client 端，电脑做 Server 端

```
Device.Application.PLC_PRG

1  IP.sAddr := '192.168.1.123'; //服务端IP地址
2  //PLC作客户端
3  TCP_Client_0(
4    xEnable := xTCP_Client_Enable TRUE ,
5    xDone=> ,
6    xBusy=> ,
7    xError=> ,
8    udiTimeOut := 500000 ,
9    ipAddr:=IP ,
10   uiPort := 8080 ,
11   eError=> ,
12   xActive=> ,
13   hConnection := hConn 0 );
14  //TCP自由协议Client端往Server端写参数
15  TCP_Write_0(
16    xExecute := xTCP_Write_Exe FALSE AND TCP_Client_0.xBusy FALSE ,
17    udiTimeOut := 500000 ,
18    xDone=> ,
19    xBusy=> ,
20    xError=> ,
21    hConnection := hConn 0 ,
22    szSize := 15 ,
23    pData := ADR(abyWriteData),
24    eError=> );
25  //TCP自由协议Client端读取Server端参数
26  TCP_Read_0(
27    xEnable := xTCP_Read_Exe FALSE AND TCP_Client_0.xBusy FALSE ,
28    xDone=> ,
29    xBusy=> ,
30    xError=> .
```

4. 调用 TCP_Client 指令，ipAddr为 NBS.IP_Client结构体变量，需要对结构体变量成员 sAddr进行赋值（Server端的IP地址，为字符串），即IP.sAddr:='192.168.1.123';（Server 端 IP 地址为 192.168.1.123）hConnection为输出句柄，CAA.HANDLE结构体变量；TCP_Client的xEnable 为TRUE后即可连接成功。



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

Device.Application.PLC_PRG					
表达式	类型	值	准备值	地址	注释
TCP_Client_0	NBS.TCP_Client				
TCP_Write_0	NBS.TCP_Write				
TCP_Read_0	NBS.TCP_Read				
xTCP_Client_Enable	BOOL	TRUE			
IP	NBS.IP_ADDR				
hConn	DWORD	0			
xTCP_Write_Exe	BOOL	FALSE			
xTCP_Read_Exe	BOOL	FALSE			
abyWriteData	ARRAY [0..14] OF B...				
abyWriteData[0]	BYTE	0	11		
abyWriteData[1]	BYTE	0	22		
abyWriteData[2]	BYTE	0	33		
abyWriteData[3]	BYTE	0			
abyWriteData[4]	BYTE	0			
abyWriteData[5]	BYTE	0			

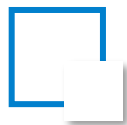
5. Client 端 (TM753端) 写数值，Server 端接收

① 确保 TCP_Client 端与 TCP_Server 端连接成功；

② 调用 TCP_Write 指令。

hConnection 为 TCP_Client 的输出句柄；pData 为 指针变 量，是写入数据变量所指向的地址；szSize 为写入数据变量的数据长度；

③ 将 abyWriteData[0]、abyWriteData[1]及 abyWriteData[2] 赋值十进制数值 11、22 及 33；



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

6.触发“xExecute”为 TRUE。

```
Device.Application.PLC_PRG

1  IP.sAddr := '192.168.1.123'; //服务端IP地址
2  //PLC作客户端
3  TCP_Client_0(
4      xEnable := xTCP_Client_Enable TRUE ,
5      xDone=> ,
6      xBusy=> ,
7      xError=> ,
8      udiTimeOut := 500000 ,
9      ipAddr:=IP ,
10     uiPort := 8080 ,
11     eError=> ,
12     xActive=> ,
13     hConnection := hConn 3018976556 );
14     //TCP自由协议Client端往Server端写参数
15 TCP_Write_0(
16     xExecute := xTCP_Write_Exe TRUE AND TCP_Client_0.xBusy TRUE ,
17     udiTimeOut := 500000 ,
18     xDone=> ,
19     xBusy=> ,
20     xError=> ,
21     hConnection := hConn 3018976556 ,
22     szSize := 15 ,
23     pData := ADR(abyWriteData) ,
24     eError=> );
25 //TCP自由协议Client端读取Server端参数
26 TCP_Read_0(
27     xEnable := xTCP_Read_Exe FALSE AND TCP_Client_0.xBusy TRUE ,
28     xDone=> ,
29     xBusy=> ,
30     xError=> ,
```



TCP IP自由口协议 (TM753为Client端)

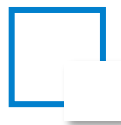
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端



7. Server 端可收到 16 进制数值 0B、16 及 21（对应的十进制数值为 11、22 及 33）。



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

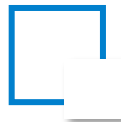
```
1 IP.sAddr := '192.168.1.123'; // 服务端IP地址
2 // PLC作客户端
3 TCP_Client_0(
4   xEnable := xTCP_Client_Enable,
5   xDone=>,
6   xBusy=>,
7   xError=>,
8   udiTimeOut := 500000,
9   ipAddr:=IP,
10  uiPort := 8080,
11  eError=>,
12  xActive=>,
13  hConnection := hConn;
14  // TCP自由协议Client端往Server端写参数
15 TCP_Write_0(
16  xExecute := xTCP_Write_Execute AND TCP_Client_0.xBusy,
17  udiTimeOut := 500000,
18  xDone=>,
19  xBusy=>,
20  xError=>,
21  hConnection := hConn,
22  szSize := 15,
23  pData := ADR(abyWriteData),
24  eError=> );
```

8.Client 端 (TM753) 读取 Server 端发送的数值

① 确保 TCP_Client 端的 xEnable 至 TRUE;

② 调用 TCP_Read 指令。

hConnection 为 TCP_Client 的输出句柄; pData 为 指针变量, 是读取数据变量所指向的地址; szSize 为读取数据变量的数据长度。



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

```
25 //TCP自由协议Client端读取Server端参数
26 TCP_Read_0(
27   xEnableFALSE:=xTCP_Read_ExeFALSE AND TCP_Client_0.xBusyTRUE ,
28   xDone=> ,
29   xBusy=> ,
30   xError=> ,
31   hConnection_3018976556:= hConn_3018976556 ,
32   szSize_15:= 15,
33   pData_3018977179:=ADR(abyReadData) ,
34   eError=> ,
35   xReady=> ,
36   szCount=> );RETURN
```

8.Client 端 (TM753) 读取 Server 端发送的数值

① 确保 TCP_Client 端的 xEnable 至 TRUE;

② 调用 TCP_Read 指令。

hConnection 为 TCP_Client 的输出句柄; pData 为 指针变量, 是读取数据变量所指向的地址; szSize 为读取数据变量的数据长度。

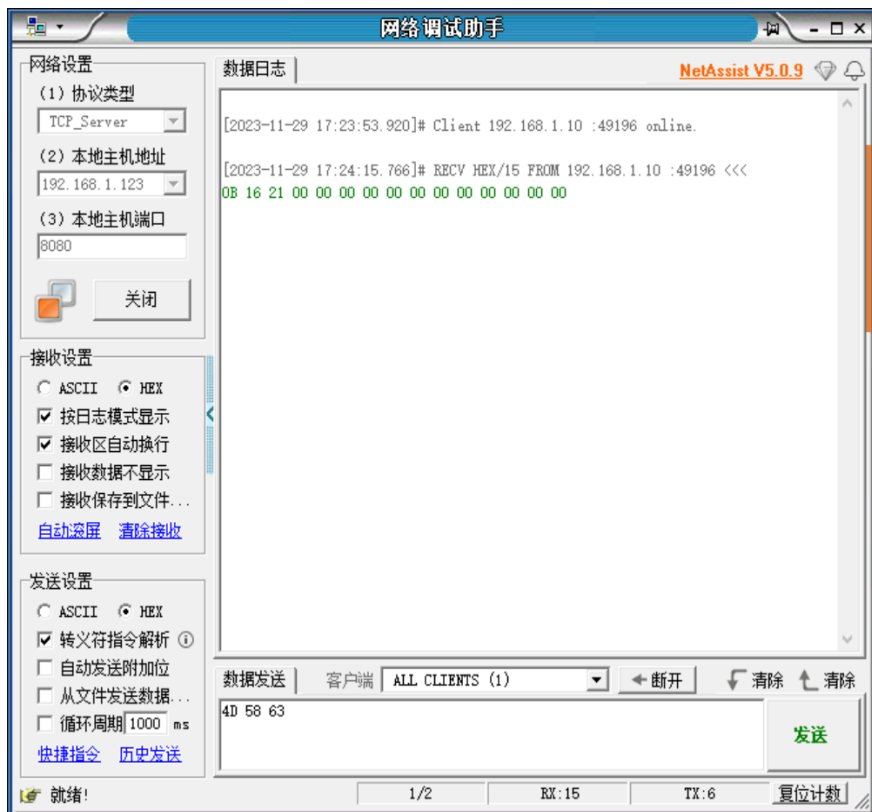


TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端



9.使用 NetAssist 网络调试助手作为 Server 端发送 16 进制数值 4D,58 ,63。



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能

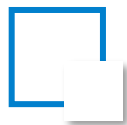


TM753 做 Client 端，电脑做 Server 端

```
Device.Application.PLC_PRG

8      udiTimeOut 500000 := 500000 ,
9      ipAddr:=IP ,
10     uiPort 8080 := 8080,
11     eError=> ,
12     xActive=> ,
13     hConnection 3018976556 =>hConn 3018976556 );
14     //TCP自由协议Client端往Server端写参数
15 ● TCP_Write_0(
16     xExecute FALSE := xTCP_Write_Exe FALSE AND TCP_Client_0.xBusy TRUE ,
17     udiTimeOut 500000 := 500000,
18     xDone=> ,
19     xBusy=> ,
20     xError=> ,
21     hConnection 3018976556 := hConn 3018976556 ,
22     szSize 15 := 15,
23     pData 3018977164 := ADR(abyWriteData),
24     eError=> );
25     //TCP自由协议Client端读取Server端参数
26 ● TCP_Read_0(
27     xEnable TRUE := xTCP_Read_Exe TRUE AND TCP_Client_0.xBusy TRUE ,
28     xDone=> ,
29     xBusy=> ,
30     xError=> ,
31     hConnection 3018976556 := hConn 3018976556 ,
32     szSize 15 := 15,
33     pData 3018977179 :=ADR(abyReadData) ,
34     eError=> ,
35     xReady=> ,
36 ●     szCount=> );RETURN
```

10.触发 TCP_Read 的 xExecute。



TCP IP自由口协议 (TM753为Client端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Client 端，电脑做 Server 端

Device.Application.PLC_PRG			
表达式	类型	值	准备值
✎ xTCP_Write_Exe	BOOL	FALSE	
✎ xTCP_Read_Exe	BOOL	TRUE	
✎ abyWriteData	ARRAY [0..14] OF B...		
✎ abyReadData	ARRAY [0..14] OF B...		
✎ abyReadData[0]	BYTE	77	
✎ abyReadData[1]	BYTE	88	
✎ abyReadData[2]	BYTE	99	
✎ abyReadData[3]	BYTE	0	
✎ abyReadData[4]	BYTE	0	
✎ abyReadData[5]	BYTE	0	
✎ abyReadData[6]	BYTE	0	
✎ abyReadData[7]	BYTE	0	

11. abyReadData[0]、
abyReadData[1]及 abyReadData[2]
的十进制数值分别为 77、88 及 99
(16 进制为 16#4D、16#58 及
16#63)，则读取成功。

PART 02

通讯示例

- TM753与VS_102QS 通讯（串口）
- TM753与VS_102QS通讯（网口标签通讯）
- MODBUS TCP通讯（从站）
- MODBUS RTU主站控制变频器
- **TCP IP自由口协议（TM753为Server端）**



TM753 做 Server 端，电脑做 Client 端

运行前准备：

1 台 TM753，1 台电脑，NetAssist 调试软件

- ① TM753 做 Server 端，电脑做Client 端；
- ② Client IP：192.168.1.123；
- ③ Server IP：192.168.1.10。

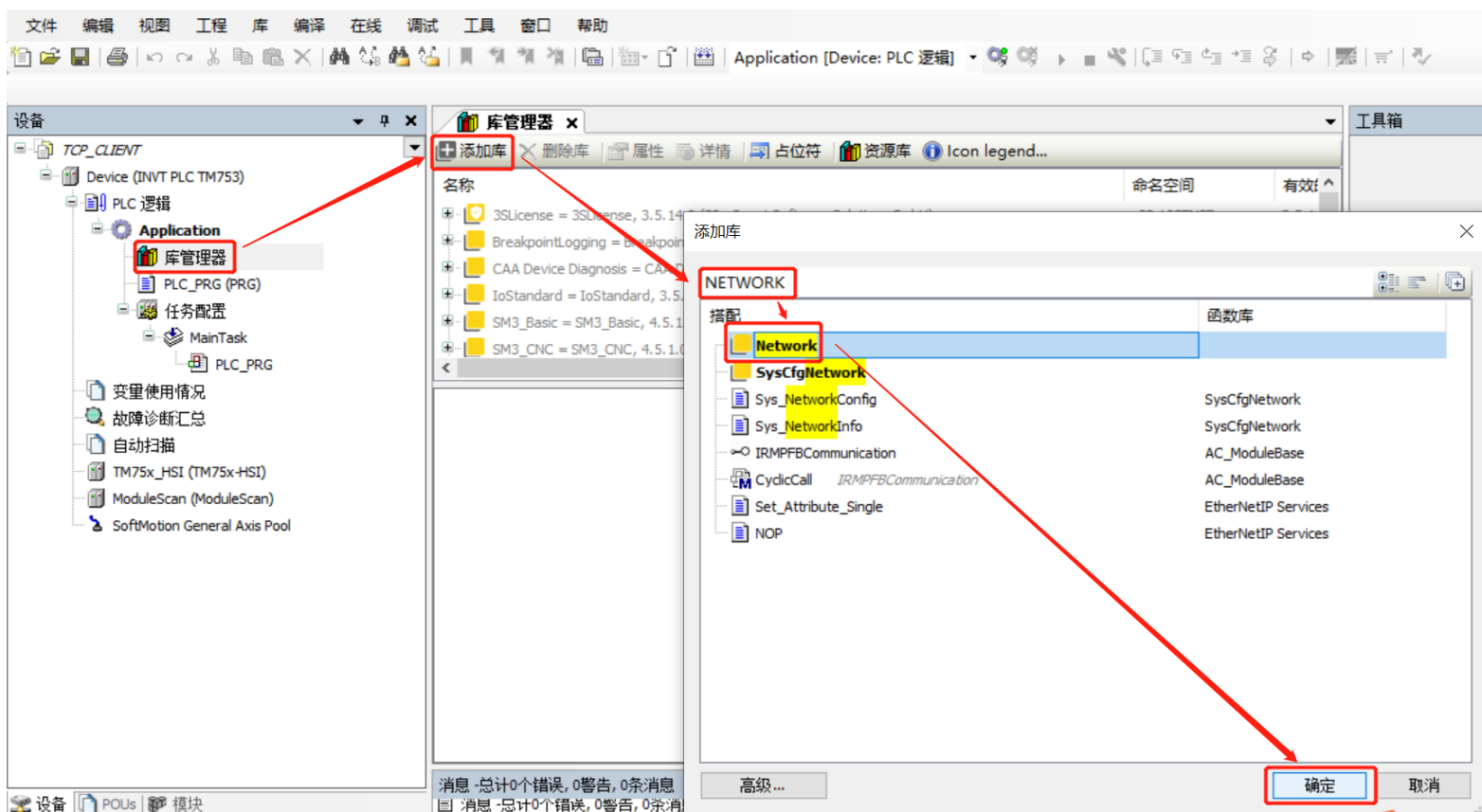


TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端



1.添加库

- ① 双击【库管理器】；
- ② 左键单击【添加库】；
- ③ 在添加库界面的文本框输入所需的库“Network”即可搜索完成；
- ④ 左键单击选择需要的库【Network】后左键单击右下角【确定】按钮即可添加完成；
- ⑤ 【Netwok,3.5.7.0】即为添加的库。



TCP IP自由口协议 (TM753为 Server端)

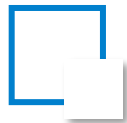
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

```
PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3      TCP_Server_0: NBS.TCP_Server;
4      TCP_Connection_0: NBS.TCP_Connection;
5      TCP_Write_0: NBS.TCP_Write;
6      TCP_Read_0: NBS.TCP_Read;
7      Server_IP:NBS.IP_ADDR;
8      hSer:CAA.HANDLE;
9      hConn:CAA.HANDLE;
10     xServer_Enable:BOOL;
11     xConnection_Enable:BOOL;
12     xTCP_Write_Exe:BOOL;
13     xTCP_Read_Exe:BOOL;
14     abyWriteData:ARRAY[0..14] OF BYTE;
15     abyReadData:ARRAY[0..14] OF BYTE;
16 END_VAR
17
```

2.创建变量，并编写程序，如图所示；



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

```
1  Server_IP.sAddr:='192.168.1.10';//Server端IP (PLC的IP地址)
2
3  //设置Server端的IP、端口号及获取Server端句柄
4  TCP_Server_0(xEnable:=xServer_Enable , xDone=> , xBusy=> , xError=> , ipAddr:=Server_IP , uiPort:= 8080,
5  eError=> , hServer=>hSer );
6
7  // 创建与Client端的连接, 获取Client端句柄
8  TCP_Connection_0(xEnable:= xConnection_Enable , xDone=> , xBusy=> , xError=> , hServer:= hSer, eError=> ,
9  xActive=> , hConnection:=hConn );
10
11 //Server端写入数据到Client端
12 TCP_Write_0(xExecute:= xTCP_Write_Exe AND TCP_Connection_0.xBusy, udiTimeOut:= 50000, xDone=> , xBusy=> , xError=> ,
13 hConnection:=hConn ,
14 szSize:= 15,
15 pData:= ADR(abyWriteData),
16 eError=> );
17
18 //Server端读取Client端发送的数据
19 TCP_Read_0(xEnable:=xTCP_Read_Exe AND TCP_Connection_0.xBusy, xDone=> , xBusy=> , xError=> ,
20 hConnection:=hConn ,
21 szSize:= 15,
22 pData:= ADR(abyReadData),
23 eError=> ,
24 xReady=> ,
25 szCount=> );
26
27
28
```

2.创建变量，并编写程序，如图所示；



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



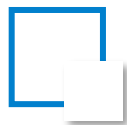
TM753 做 Server 端，电脑做 Client 端

```
Device.Application.PLC_PRG

1  ● Server_IP.sAddr := '192.168.1.10' ; //Server端IP (PLC的IP地址)
2
3  //设置Server端的IP、端口号及获取Server端句柄
4  ● TCP_Server_0(xEnable:=TRUE, xServer_Enable:=TRUE, xDone=>, xBusy=>, xError=>, ipAddr:=Server_IP, uiPort:=8080,
5  eError=>, hServer:=3019189636 =>hSer:=3019189636 );
6
7  // 创建与Client端的连接, 获取Client端句柄
8  ● TCP_Connection_0(xEnable:=FALSE, xConnection_Enable:=FALSE, xDone=>, xBusy=>, xError=>, hServer:=3019189636 := hSer:=3019189636,
9  xActive=>, hConnection:=0 =>hConn:=0 );
10
11 //Server端写入数据到Client端
12 ● TCP_Write_0(xExecute:=FALSE, xTCP_Write_Exec:=FALSE AND TCP_Connection_0.xBusy:=FALSE, udiTimeOut:=50000 := 50000, xDone=>, xBusy=
13 hConnection:=0 :=hConn:=0 ,
14 szSize:=15 := 15,
15 pData:=3019190884 := ADR(abyWriteData),
16 eError=> );
17
18 //Server端读取Client端发送的数据
19 ● TCP_Read_0(xEnable:=FALSE, xTCP_Read_Exec:=FALSE AND TCP_Connection_0.xBusy:=FALSE, xDone=>, xBusy=>, xError=>,
20 hConnection:=0 :=hConn:=0 ,
21 szSize:=15 := 15,
22 pData:=3019190899 := ADR(abyReadData),
23 eError=>,
24 xReady=>,
25 szCount=> );
26
27
28
29
```

3.设置 Server 端参数及获取通信句柄

- ① 调用 TCP_Server 指令;
- ② 设置相关参数, ipAddr 为 NBS.IP_Server 结构体变量, 需要对结构体变量 成员 sAddr 进行赋值 (Server 端的 IP 地址, 为字符串), 即 IP.sAddr:='192.168.1.10'; (Server 端 IP 地址为 192.168.1.10) ;
- ③ uiPort 为 Server 端的端口号, UINT 数据类型, 端口号为 8080 (自由定义, 但不能和其余通信端口号重复) ;
- ④ hServer 为通信句柄, CAA.HANDLE 结构体变量;
- ⑤ TCP_Server 的 xEnable 至 TRUE。



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



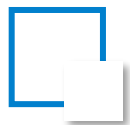
TM753 做 Server 端，电脑做 Client 端

```
Device.Application.PLC_PRG

1 Server_IP.sAddr := '192.168.1.10'; //Server端IP (PLC的IP地址)
2
3 //设置Server端的IP、端口号及获取Server端句柄
4 TCP_Server_0(xEnable:=xServer_Enable, xDone=>, xBusy=>, xError=>, ipAddr:=Server_IP, uiPort:=8080,
5 eError=>, hServer:=hSer );
6
7 // 创建与Client端的连接, 获取Client端句柄
8 TCP_Connection_0(xEnable:=xConnection_Enable, xDone=>, xBusy=>, xError=>, hServer:=hSer,
9 xActive=>, hConnection:=hConn );
10
11 //Server端写入数据到Client端
12 TCP_Write_0(xExecute:=xTCP_Write_Exe AND TCP_Connection_0.xBusy, udiTimeOut:=50000, xDone=>, xBusy=
13 hConnection:=hConn,
14 szSize:=15,
15 pData:=ADR(abyWriteData),
16 eError=> );
17
18 //Server端读取Client端发送的数据
19 TCP_Read_0(xEnable:=xTCP_Read_Exe AND TCP_Connection_0.xBusy, xDone=>, xBusy=>, xError=>,
20 hConnection:=hConn,
21 szSize:=15,
22 pData:=ADR(abyReadData),
23 eError=>,
24 xReady=>,
25 szCount=> );
26
27
28
29
```

4.建立与 Client 端连接

- ① 调用 TCP_Connection;
- ② hServer 为 Server 端通信句柄，使用 TCP_Server 获取的通信句柄即可；
- ③ hConnection 为 Client 端输出句柄，CAA.HANDLE 结构体变量；
- ④ TCP_Connection 的 xEnable 至 TRUE。



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

```
Device.Application.PLC_PRG

1  Server_IP.sAddr := '192.168.1.10'; //Server端IP (PLC的IP地址)
2
3  //设置Server端的IP、端口号及获取Server端句柄
4  TCP_Server_0(xEnable:=xServer_Enable, xDone=>, xBusy=>, xError=>, ipAddr:=Server_IP, uiPort:=8080,
5  eError=>, hServer:=hSer );
6
7  // 创建与Client端的连接, 获取Client端句柄
8  TCP_Connection_0(xEnable:=xConnection_Enable, xDone=>, xBusy=>, xError=>, hServer:=hSer,
9  xActive=>, hConnection:=hConn );
10
11 //Server端写入数据到Client端
12 TCP_Write_0(xExecute:=xTCP_Write_Exe AND TCP_Connection_0.xBusy, udiTimeOut:=50000, xDone=>, xBusy=
13 hConnection:=hConn,
14 szSize:=15,
15 pData:=ADR(abyWriteData),
16 eError=> );
17
18 //Server端读取Client端发送的数据
19 TCP_Read_0(xEnable:=xTCP_Read_Exe AND TCP_Connection_0.xBusy, xDone=>, xBusy=>, xError=>,
20 hConnection:=hConn,
21 szSize:=15,
22 pData:=ADR(abyReadData),
23 eError=>,
24 xReady=>,
25 szCount=> );
26
27
28
29
```

5. Server 端 写数字，Client 端接收

- ① TCP_Server及TCP_Connection的 xEnable至TRUE；
- ② 调用TCP_Write指令。

hConnection为TCP_Client的输出句柄；pData 为指针变量，是写入数据变量所指向的地址；szSize 为写入数据变量的数据长度。

TM753 做 Server 端，电脑做 Client 端



6.打开 NetAssist 网络调试助手软件，“协议类型”为 TCP Client，“本地 IP 地址”为 192.168.1.10，“本地端口”为 8080，设置好参数后左键单击“连接”，确保 TCP_Server 与 TCP_Client 端连接成功；



TCP IP自由口协议 (TM753为 Server端)

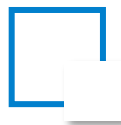
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

Device.Application.PLC_PRG					
表达式	类型	值	准备值	地址	
hConn	DWORD	3019190324			
xServer_Enable	BOOL	TRUE			
xConnection_Enable	BOOL	TRUE			
xTCP_Write_Exe	BOOL	FALSE			
xTCP_Read_Exe	BOOL	FALSE			
abyWriteData	ARRAY [0..14] OF B...				
abyWriteData[0]	BYTE	11			
abyWriteData[1]	BYTE	22			
abyWriteData[2]	BYTE	33			
abyWriteData[3]	BYTE	0			
abyWriteData[4]	BYTE	0			
abyWriteData[5]	BYTE	0			
abyWriteData[6]	BYTE	0			
abyWriteData[7]	BYTE	0			
abyWriteData[8]	BYTE	0			
abyWriteData[9]	BYTE	0			
abyWriteData[10]	BYTE	0			
abyWriteData[11]	BYTE	0			
abyWriteData[12]	BYTE	0			
abyWriteData[13]	BYTE	0			
abyWriteData[14]	BYTE	0			
abyReadData	ARRAY [0..14] OF B...				

7.将 abyWriteData[0]、
abyWriteData[1]及 abyWriteData[2]
赋值十进制数值 11、 22 及 33。



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

```
Device.Application.PLC_PRG

1  Server_IP.sAddr := '192.168.1.10'; //Server端IP (PLC的IP地址)
2
3  //设置Server端的IP、端口号及获取Server端句柄
4  TCP_Server_0(xEnable:=xServer_Enable TRUE, xDone=>, xBusy=>, xError=>, ipAddress:=Server_IP, uiPort:=8080,
5  eError=>, hServer:=hSer[3019189636]);
6
7  // 创建与Client端的连接, 获取Client端句柄
8  TCP_Connection_0(xEnable:=xConnection_Enable TRUE, xDone=>, xBusy=>, xError=>, hServer:=hSer[3019189636],
9  xActive=>, hConnection:=hConn[3019190324]);
10
11 //Server端写入数据到Client端
12 TCP_Write_0(xExecute:=xTCP_Write_Exe TRUE AND TCP_Connection_0.xBusy TRUE, udiTimeOut:=50000, xDone=>, xBusy=
13 hConnection:=hConn[3019190324],
14 szSize:=15,
15 pData:=ADR(abyWriteData),
16 eError=>);
17
18 //Server端读取Client端发送的数据
19 TCP_Read_0(xEnable:=xTCP_Read_Exe FALSE AND TCP_Connection_0.xBusy TRUE, xDone=>, xBusy=>, xError=>,
20 hConnection:=hConn[3019190324],
21 szSize:=15,
22 pData:=ADR(abyReadData),
23 eError=>,
24 xReady=>,
25 szCount=>);
26
27
28
29
```

8.触发 execute。

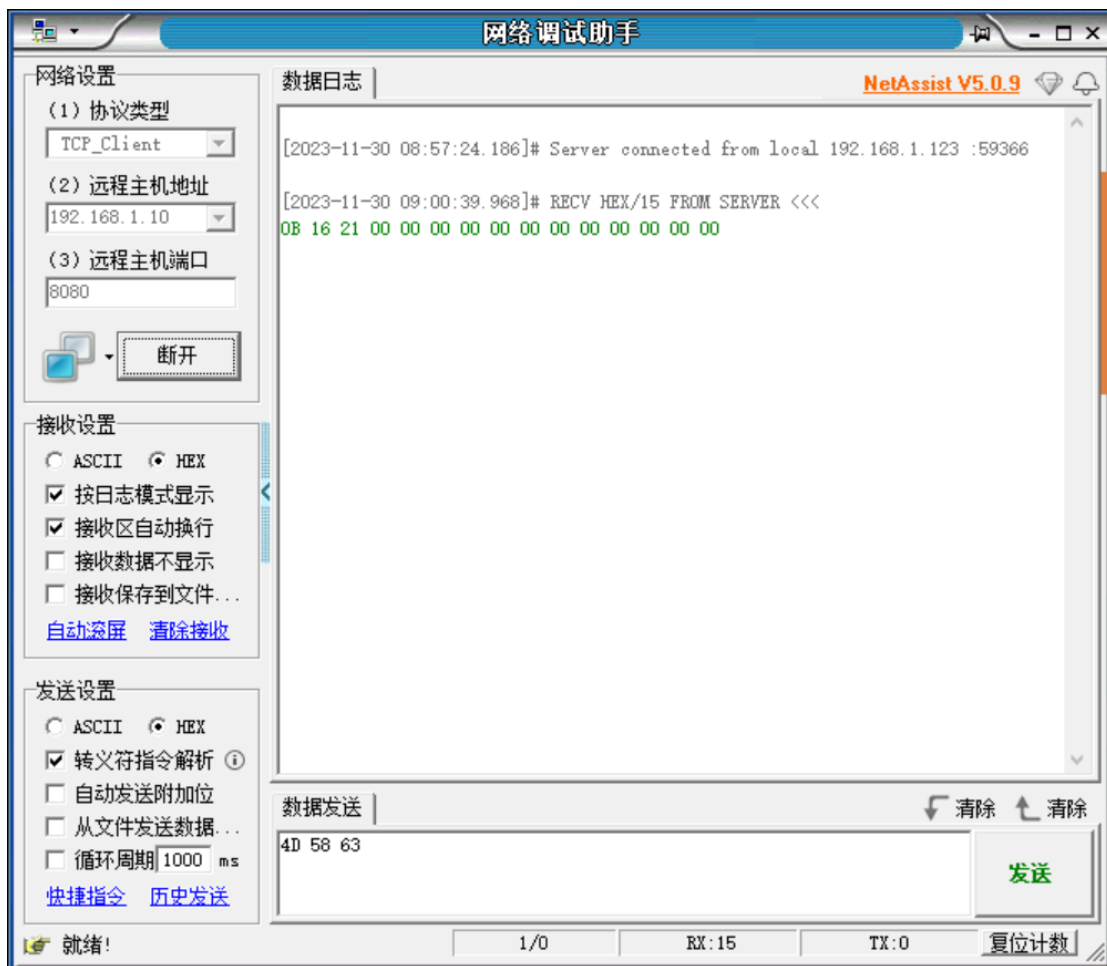


TCP IP自由口协议 (TM753为 Server端)

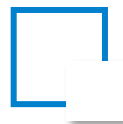
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端



9. Client 端可收到 16 进制数值 0B、16 及 21（为十进制数值 11、22 及 33）。



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

```
Device.Application.PLC_PRG

1  Server_IP.sAddr := '192.168.1.10'; //Server端IP (PLC的IP地址)
2
3  //设置Server端的IP、端口号及获取Server端句柄
4  TCP_Server_0(xEnable:=xServer_Enable, xDone=>, xBusy=>, xError=>, ipAddress:=Server_IP, uiPort:=8080,
5  eError=>, hServer:=hSer );
6
7  // 创建与Client端的连接, 获取Client端句柄
8  TCP_Connection_0(xEnable:=xConnection_Enable, xDone=>, xBusy=>, xError=>, hServer:=hSer,
9  xActive=>, hConnection:=hConn );
10
11 //Server端写入数据到Client端
12 TCP_Write_0(xExecute:=xTCP_Write_Exec, xDone=>, xBusy=>, xError=>, hConnection:=hConn,
13 szSize:=15,
14 pData:=ADR(abyWriteData),
15 eError=> );
16
17
18 //Server端读取Client端发送的数据
19 TCP_Read_0(xEnable:=xTCP_Read_Exec, xDone=>, xBusy=>, xError=>,
20 hConnection:=hConn,
21 szSize:=15,
22 pData:=ADR(abyReadData),
23 eError=>,
24 xReady=>,
25 szCount=> );
26
27
28
29
```

10. Server 端 读取 Client 端发送的数值

① TCP_Server 及 TCP_Connection 的 xEnable 设置为 TRUE;

② 调用 TCP_Read 指令。

hConnection 为 TCP_Client 的输出句柄; pData 为 指针变量, 是读取数据变量所指向的地址; szSize 为读取数据变量的数据长度。

○



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能

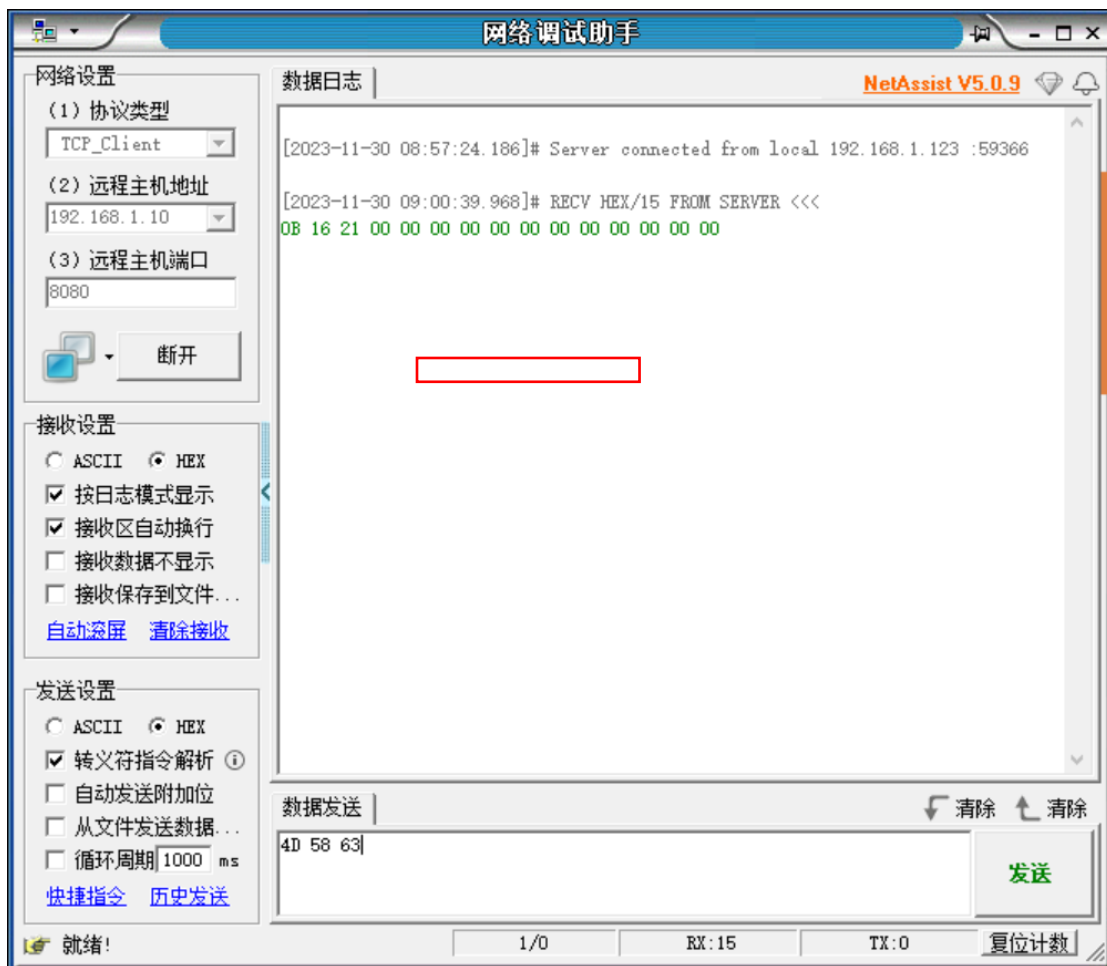


TM753 做 Server 端，电脑做 Client 端



11.打开 NetAssist 网络调试助手软件，“协议类型”为 TCP Client，“本地 IP 地址”为 192.168.1.10，“本地端口”为 8080，设置好参数后左键单击“连接”，确保 TCP_Server 与 TCP_Client 端连接成功。

TM753 做 Server 端，电脑做 Client 端



12.使用 NetAssist 网络调试助手作为 Client 端发送 16 进制数值 4D 58 63 (十进制为 77、88 及 99)。



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

```
Device.Application.PLC_PRG

1  ● Server_IP.sAddr := '192.168.1.10' ; //Server端IP (PLC的IP地址)
2
3  //设置Server端的IP、端口号及获取Server端句柄
4  ● TCP_Server_0(xEnable:=xServer_Enable TRUE , xDone=> , xBusy=> , xError=> , ipAddress:=Server_IP , uiPort:= 8080 ,
5  eError=> , hServer:=hSer );
6
7  // 创建与Client端的连接, 获取Client端句柄
8  ● TCP_Connection_0(xEnable:= xConnection_Enable TRUE , xDone=> , xBusy=> , xError=> , hServer:= hSer , e
9  xActive=> , hConnection:=hConn );
10
11 //Server端写入数据到Client端
12 ● TCP_Write_0(xExecute:= xTCP_Write_Exe FALSE AND TCP_Connection_0.xBusy TRUE , udiTimeOut:= 50000 , xDone=> , xBusy=
13 hConnection:=hConn ,
14 szSize:= 15 ,
15 pData:= ADR(abyWriteData),
16 eError=> );
17
18 //Server端读取Client端发送的数据
19 ● TCP_Read_0(xEnable:=xTCP_Read_Exe TRUE AND TCP_Connection_0.xBusy TRUE , xDone=> , xBusy=> , xError=> ,
20 hConnection:=hConn ,
21 szSize:= 15 ,
22 pData:= ADR(abyReadData),
23 eError=> ,
24 xReady=> ,
25 szCount=> );
26
27
28
29
```

13.触发 TCP_Read 的 xExecute 设置为 TRUE。

100 %



TCP IP自由口协议 (TM753为 Server端)

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



TM753 做 Server 端，电脑做 Client 端

Device.Application.PLC_PRG				
表达式	类型	值	准备值	地址
abyWriteData[9]	BYTE	0		
abyWriteData[10]	BYTE	0		
abyWriteData[11]	BYTE	0		
abyWriteData[12]	BYTE	0		
abyWriteData[13]	BYTE	0		
abyWriteData[14]	BYTE	0		
abyReadData	ARRAY [0..14] OF B...			
abyReadData[0]	BYTE	77		
abyReadData[1]	BYTE	88		
abyReadData[2]	BYTE	99		
abyReadData[3]	BYTE	0		
abyReadData[4]	BYTE	0		
abyReadData[5]	BYTE	0		
abyReadData[6]	BYTE	0		
abyReadData[7]	BYTE	0		
abyReadData[8]	BYTE	0		
abyReadData[9]	BYTE	0		
abyReadData[10]	BYTE	0		
abyReadData[11]	BYTE	0		
abyReadData[12]	BYTE	0		
abyReadData[13]	BYTE	0		
abyReadData[14]	BYTE	0		

14. abyReadData[0]、
abyReadData[1]及 abyReadData[2]
分别的数值为 77、88 及 99， 则读取
成功。



PART 03



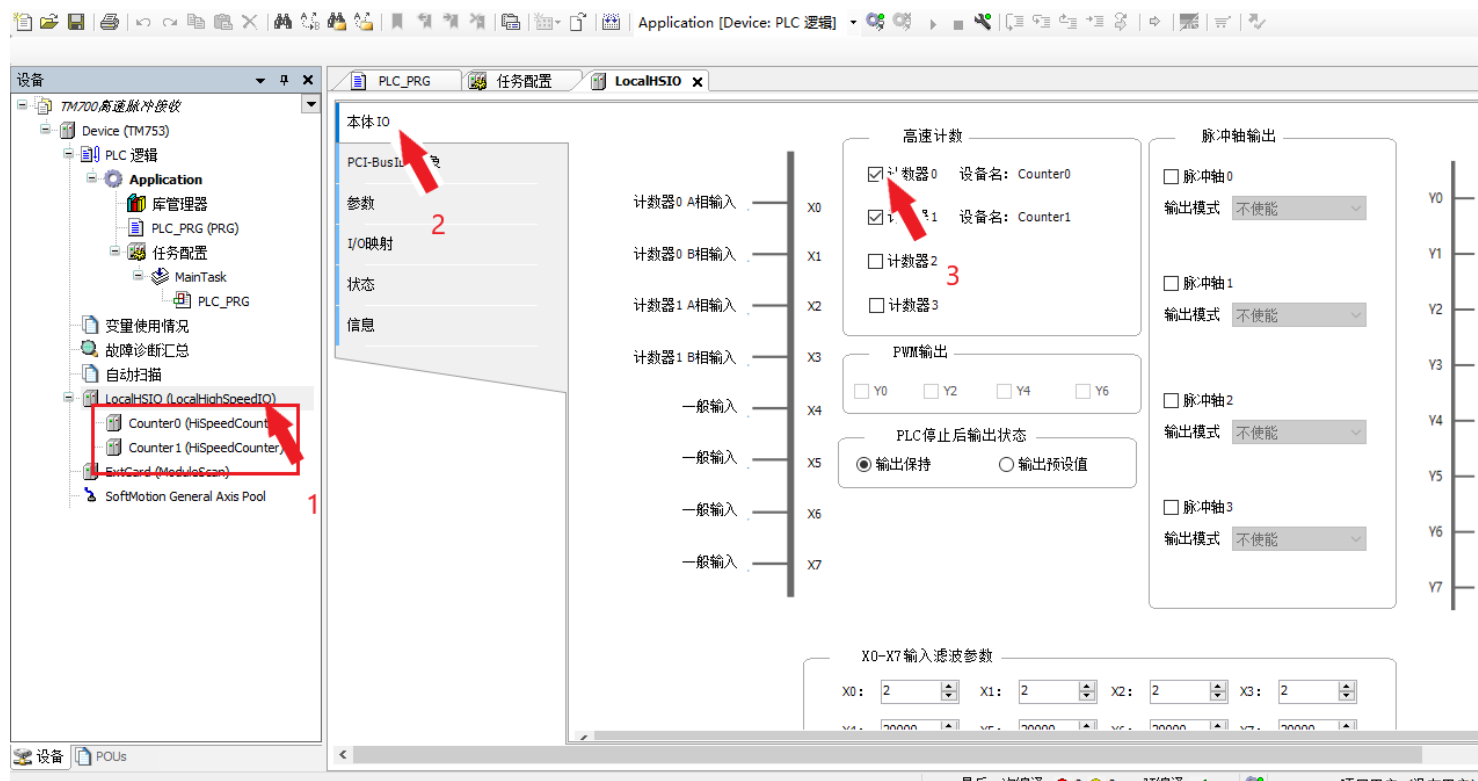
高速计数器

- 设置
- 应用案例

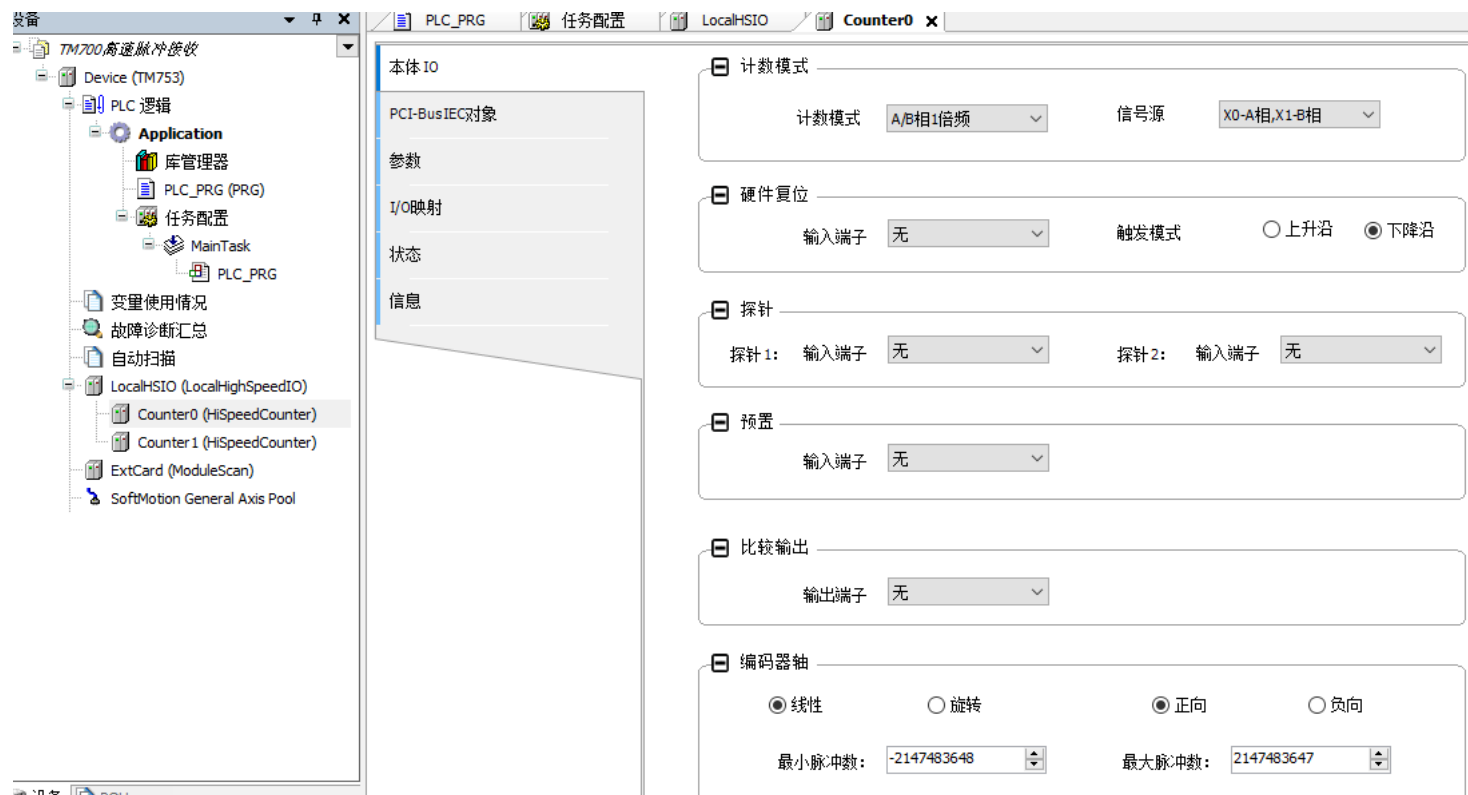


高速计数器设置

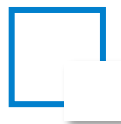
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



1. 新建TM753的程序，打开设备左侧 LocalHSIO (LocalHighSpeedIO) ,在本体IO 中，勾选需要的计数器，勾选完成计数器0 与计数器1后，下方会新建高速输入的配置块 (Counter0)



2. 双击Counter0，本体IO，根据实际应用，在勾选对应的功能。



本体IO PCI-BusIEC对象 参数 I/O映射 状态 信息	查找 过滤 显示所有 + 给IO通道添加FB... 转到实例						
	变量	映射	通道	地址	类型	单元	描述
			Statusword	%IW2	WORD		Counter statusword
			Counter Value	%ID2	DINT		Counter value of pulses
			Preset Status	%IW6	WORD		Status of preset function
			Compare Status	%IW7	WORD		Status of compare function
			Touch Probe Status1	%IW8	WORD		Status of touch probe1 function
			Touch Probe Status2	%IW9	WORD		Statusword of touch probe2 function
			Latch rising pos1	%ID5	DINT		Rising edge latch position of touch probe1
			Latch rising pos2	%ID6	DINT		Rising edge latch position of touch probe2
			Latch falling pos1	%ID7	DINT		Falling edge latch position of touch probe1
			Latch falling pos2	%ID8	DINT		Falling edge latch position of touch probe2
			Controlword	%QW2	WORD		Command Controlword
			Preset Value	%QD2	DINT		Value of preset function
			Preset command	%QW6	WORD		Command of preset function
			Compare Mode	%QW7	UINT		Work mode of compare function
			Compare Parameter	%QD4	UDINT		Parameter of compare function
			Compare Value	%QD5	DINT		Compare value of compare function
			Output Type	%QW12	UINT		Digital output type of compare function
			Output Parameter	%QW13	UINT		Digital output parameter of compare function
		复位映射	一直更新变量:		使能1(如果未在任何任务中使用则使用总线循环):		

3. 设置好匹配的功能，查看I/O映射，利用对应的控制字实现。



计数器控制字

Statusword	计数器状态字 bit0 计数器运行状态反馈 1:: 正在运行 bit1 计数器方向反馈 1: 反向 bit2 正向限制状态反馈 1: 达到cntMax bit3 负向限制状态反馈 1: 达到cntMin bit4 计数器错误状态 1: 有错误
Counter Value	实时计数值
Preset Status	预设状态反馈 bit0 预设完成状态反馈 1: done 注: 预设done信号需要清除, 没清除之前不能修改预设值
Compare Status	
Touch Probe Status1	计数器探针1状态反馈 bit0 探针busy状态反馈 bit1 探针done状态反馈 注: 探针done信号需要清除, 没清除之前不再更新锁存值
Touch Probe Status2	计数器探针2状态反馈 bit0 探针busy状态反馈 bit1 探针done状态反馈 注: 探针done信号需要清除, 没清除之前不再更新锁存值
Latch rising pos1	探针1 上升沿锁存位置
Latch rising pos2	探针2 上升沿锁存位置
Latch falling pos1	探针1 下降沿锁存位置
Latch falling pos2	探针2 下降沿锁存位置

Controlword	计数器控制字 bit0 计数器计数启动 bit1 计数器计数复位
Preset Value	预设值
Preset Command	预设控制字 bit0 软件预设使能, 上升沿有效 bit1 预设复位 bit2 预设done信号清零
Touch probe command1	探针1 控制字 bit0 探针启动信号 高有效 bit1 探针复位信号 bit2 探针done清零信号, 高有效
Touch probe command2	探针2 控制字 bit0 探针启动信号 高有效 bit1 探针复位信号 bit2 探针done清零信号, 高有效

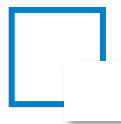


PART 03



高速计数器

- 设置
- 应用案例



运行前准备：

1.TS635 PLC (脉冲发送)

2.TM753 （高速计数）

软件版本

Auto Station Pro V1.2.7

PLC版本1.2.7

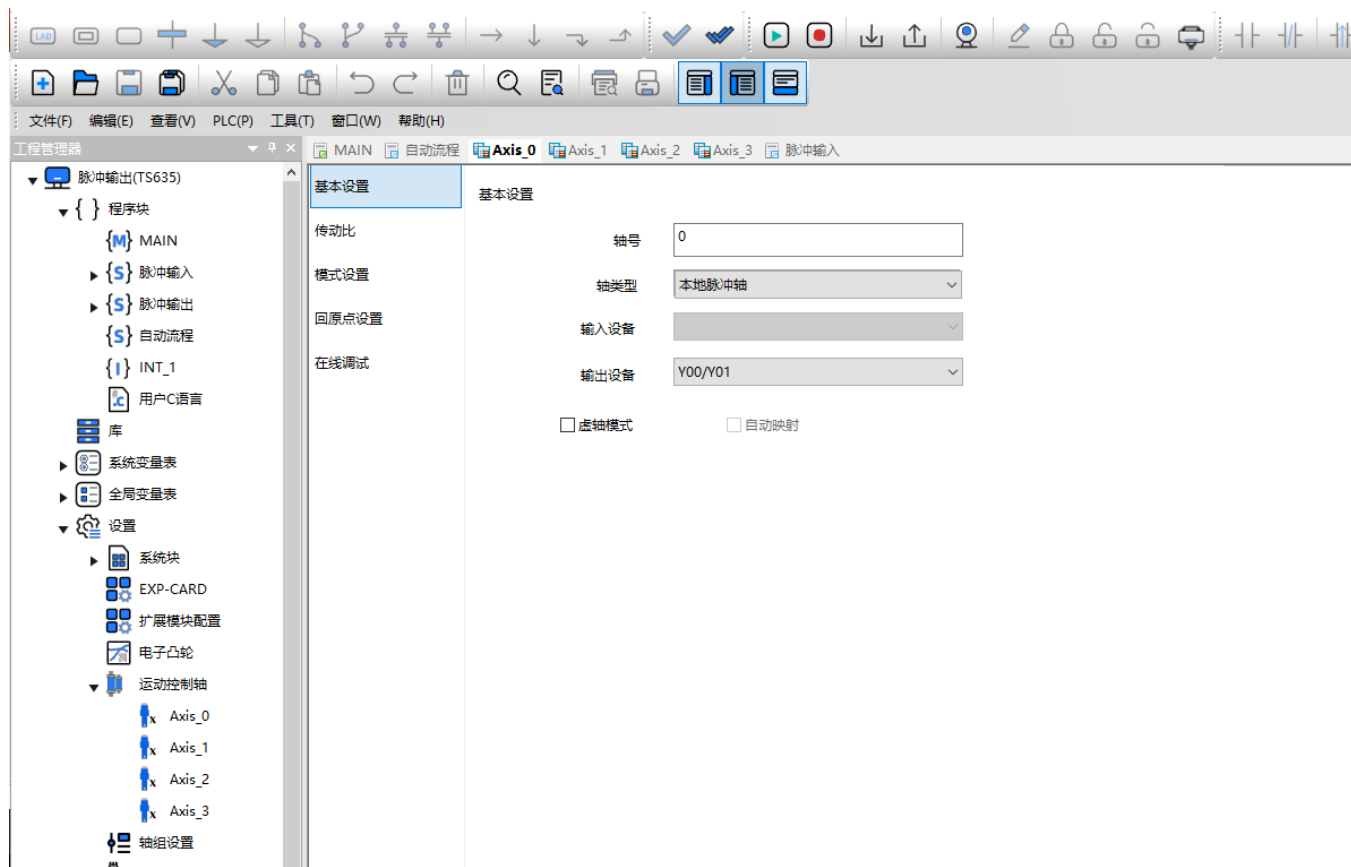
Invtronic Studio V1.3.4.15

PLC版本TM753_V1.04.00



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



1. Auto Station Pro新建工程 TS635完成后，在工程管理器-设置-运动控制轴-右键新建轴。
2. 基本设置选择本地脉冲轴，输出设备选择Y0/Y1。



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



基本设置

传动比

模式设置

回原点设置

在线调试

传动比:

电机/编码器旋转一圈的脉冲数:

10000

☒ 不使用变速装置

工作台旋转一圈的移动量:

10.000

☐ 使用变速装置

工作台旋转一圈的移动量:

10.000

齿轮比分子:

1

齿轮比分母:

1

齿轮比 = N/M

导程

3. 点击传动比，设置对应电机一圈的移动脉冲数，工作台一圈的移动量为10mm，脉冲当量为1个脉冲的移动量为0.001mm。



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



基本设置	
传动比	
模式设置	
回原点设置	
在线调试	

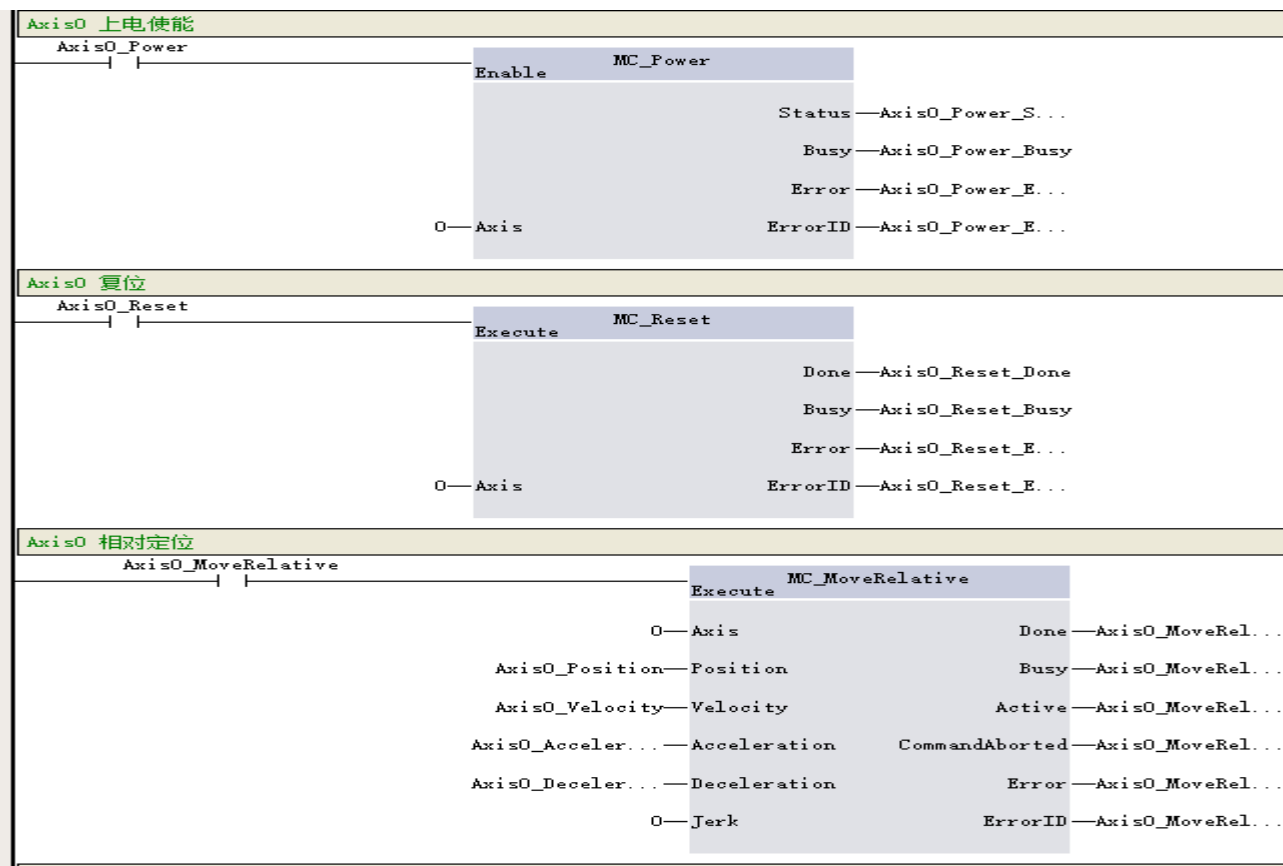
模式设置	<input checked="" type="radio"/> 线性模式 <input type="radio"/> 旋转模式
软件限位	<input type="checkbox"/> 使能 正向限位值: 1000.000 Unit 负向限位值: 0.000 Unit
软件出错响应	<input type="checkbox"/> 伺服报警使能 轴故障减速度: 10000.000 Unit 伺服报警: <input type="text"/>
轴速度设置	最大速度: 1000.000 Unit/s 最大加速度: 10000.000 Unit/s^2 最大加加速度: 100000.000 Unit/s^3 最大减速度: 10000.000 Unit/s^2 Jog最大速度: 1000.000 Unit/s
探针输出	<input type="checkbox"/> 探针0使能 <input type="checkbox"/> 探针1使能 探针0: <input type="text"/> 探针1: <input type="text"/>
输出设置	<input type="checkbox"/> 伺服使能 <input type="checkbox"/> 故障清除使能 伺服使能: <input type="text"/> 故障清除: <input type="text"/> 输出方式: 脉冲+方向
原点设置	<input type="checkbox"/> 原点使能信号 <input type="checkbox"/> Z信号使能 原点信号端子: <input type="text"/> Z信号端子: <input type="text"/>
硬限位	<input type="checkbox"/> 正限位使能 <input type="checkbox"/> 负限位使能 硬件正限位: <input type="text"/> 硬件负限位: <input type="text"/>

4.点击模式设置，Axis0输出方式选择脉冲+方向，Axis1选择正交脉冲。



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能

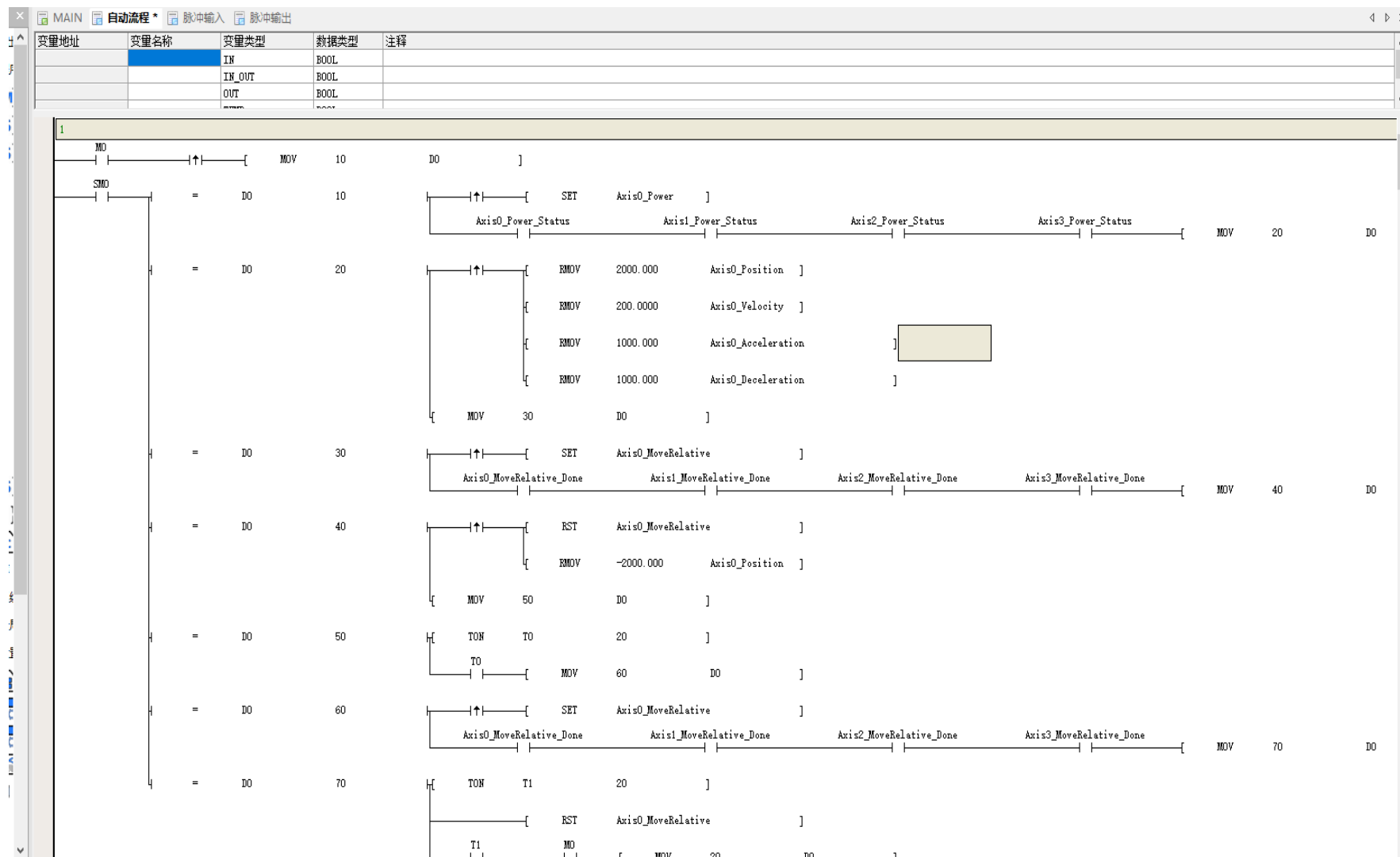


5.编写TS635PLC发送脉冲程序。



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



6.当M0启动后TS635循环反复正向以200KHZ的速度发送200K脉冲，发送完成后反向发送2000K脉冲。Y0/Y1（脉冲+方向）Y2/Y3(正交脉冲)



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



设备

T700接收脉冲 (高速计数器)

Device (TM753)

PLC 逻辑

Application

库管理器

PLC_PRG (PRG)

任务配置

MainTask

PLC_PRG

Trace

变量使用情况

故障诊断汇总

自动扫描

LocalHSIO (LocalHighSpeedIO)

Counter0 (HiSpeedCounter)

Counter1 (HiSpeedCounter)

ExtCard (ModuleScan)

SoftMotion General Axis Pool

PLC_PRG

Counter0

Device

LocalHSIO x

Counter1

编辑IO映射

本体 IO

PCI-BusIEC对象

参数

I/O映射

状态

信息

计数器0 脉冲输入 X0

计数器0 方向输入 X1

计数器1 A相输入 X2

计数器1 B相输入 X3

一般输入 X4

一般输入 X5

一般输入 X6

一般输入 X7

高速计数

☒ 计数器0 设备名: Counter0

☒ 计数器1 设备名: Counter1

☐ 计数器2

☐ 计数器3

PWM输出

☐ Y0 ☐ Y2 ☐ Y4 ☐ Y6

PLC停止后输出状态

☒ 输出保持 ☐ 输出预设值

X0-X7输入滤波参数

7.在LocalHSIO中，勾选计数器0，计数器1。



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



PLC_PRG

Counter0 x

Counter1

本体IO

PCI-BusIEC对象

参数

I/O映射

状态

信息

计数模式

计数模式 脉冲+方向 信号源 X0-脉冲,X1-方向

硬件复位

输入端子 无 触发模式 ☐ 上升沿 ☒ 下降沿

探针

探针1: 输入端子 无 探针2: 输入端子 无

预置

本体IO

PCI-BusIEC对象

参数

I/O映射

状态

信息

计数模式

计数模式 CW/CCW 信号源 X2-A相,X3-B相

硬件复位

输入端子 无 触发模式 ☐ 上升沿 ☒ 下降沿

探针

探针1: 输入端子 无 探针2: 输入端子 无

预置

8.对应的计数器0，与计数器1设置。Counter0 本体IO计数模式选择脉冲+方向，信号源X0脉冲，X1方向，Counter1本体IO计数模式选择CW/CCW，信号源X2-A相，X3-B相。

注意：如果出现下拉框选择变成灰色，不可选取的情况，关闭对应的Counter,重新打开选择。



启动高速计数器

TM700高速脉冲接收

Device [连接的] (TM753)

PLC 逻辑

Application [运行]

库管理器

PLC_PRG (PRG)

任务配置

MainTask

PLC_PRG

变量使用情况

故障诊断汇总

自动扫描

LocalHSIO (LocalHighSpeedIO)

Counter0 (HiSpeedCounter)

Counter1 (HiSpeedCounter)

ExtCard (ModuleScan)

SoftMotion General Axis Pool

本体 I/O

PCI-BusIEC对象

参数

I/O映射

状态

信息

查找 过滤 显示所有 + 给IO通道添加FB... 转到实例

变量	映射	通道	地址	类型	当前值	预备值	单元	描述
Statusword			%IW2	WORD	1			Coun
Counter Value			%ID2	DINT	1712253			Coun
Preset Status			%IW6	WORD	0			Statu
Compare Status			%IW7	WORD	0			Statu
Touch Probe Status1			%IW8	WORD	0			Statu
Touch Probe Status2			%IW9	WORD	0			Statu
Latch rising pos1			%ID5	DINT	0			Rising
Latch rising pos2			%ID6	DINT	0			Rising
Latch falling pos1			%ID7	DINT	0			Falling
Latch falling pos2			%ID8	DINT	0			Falling
Controlword			%QW2	WORD	1			Comm
Preset Value			%QD2	DINT	0			Value
Preset command			%QW6	WORD	0			Comm
Compare Mode			%QW7	UINT	0			Work
Compare Parameter			%QD4	UDINT	0			Paran

Command Controlword 复位映射 一直更新变量: 使能1(如果未在任何任务中使用则使用总线循环)

9. 参照高速计数器控制字，Controlword %QW2的BIT0为计数器的启动位,%ID2为Counter Value实时的计数值。



高速计数器预设值修改

	Statusword	%IW2	WORD	2		Counter statusword
	Counter Value	%ID2	DINT	12345		Counter value of pulses
	Preset Status	%IW6	WORD	1		Status of preset function
	Compare Status	%IW7	WORD	0		Status of compare function
	Touch Probe Status1	%IW8	WORD	0		Status of touch probe1 function
	Touch Probe Status2	%IW9	WORD	0		Statusword of touch probe2 function
	Latch rising pos1	%ID5	DINT	0		Rising edge latch position of touch probe 1
	Latch rising pos2	%ID6	DINT	0		Rising edge latch position of touch probe2
	Latch falling pos1	%ID7	DINT	0		Falling edge latch position of touch probe1
	Latch falling pos2	%ID8	DINT	0		Falling edge latch position of touch probe2
	Controlword	%QW2	WORD	0		Command Controlword
	Preset Value	%QD2	DINT	12345		Value of preset function
	Preset command	%QW6	WORD	1		Command of preset function
	Bit0	%QX12.0	BOOL	TRUE		
	Bit1	%QX12.1	BOOL	FALSE		
	Bit2	%QX12.2	BOOL	FALSE		
	Bit3	%QX12.3	BOOL	FALSE		
	Bit4	%QX12.4	BOOL	FALSE		
	Bit5	%QX12.5	BOOL	FALSE		
	Bit6	%QX12.6	BOOL	FALSE		
	Bit7	%QX12.7	BOOL	FALSE		
	Bit8	%QX13.0	BOOL	FALSE		

10.修改预设值，参照高速计数器控制字，Preset Value%QD2为预设值,设置为12345，当Preset Command的软件预设使能QX12.0使能后，实时计数值变成预设值，预设状态反馈Preset Status的Bit0为变成1。

注意：预设done信号需要清除，没清除之前修改预设值，修改的预设值不生效，仍按上一次成功修改的预设值为准。需QX12.2置位清除Done信号。



高速计数器案例

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



高速计数器复位

变量	映射	通道	地址	类型	当前值	预备值	单元	描述
		Statusword	%IW2	WORD	3			Counter statusword
		Counter Value	%ID2	DINT	0			Counter value of pulses
		Preset Status	%IW6	WORD	1			Status of preset function
		Compare Status	%IW7	WORD	0			Status of compare function
		Touch Probe Status1	%IW8	WORD	0			Status of touch probe1 function
		Touch Probe Status2	%IW9	WORD	0			Statusword of touch probe2 function
		Latch rising pos1	%ID5	DINT	0			Rising edge latch position of touch probe1
		Latch rising pos2	%ID6	DINT	0			Rising edge latch position of touch probe2
		Latch falling pos1	%ID7	DINT	0			Falling edge latch position of touch probe1
		Latch falling pos2	%ID8	DINT	0			Falling edge latch position of touch probe2
		Controlword	%QW2	WORD	3			Command Controlword
		Bit0	%QX4.0	BOOL	TRUE			
		Bit1	%QX4.1	BOOL	TRUE			
		Bit2	%QX4.2	BOOL	FALSE			
		Bit3	%QX4.3	BOOL	FALSE			
		Bit4	%QX4.4	BOOL	FALSE			
		Bit5	%QX4.5	BOOL	FALSE			
		Bit6	%QX4.6	BOOL	FALSE			
		Bit7	%QX4.7	BOOL	FALSE			
		Bit8	%QX5.0	BOOL	FALSE			
		Bit9	%QX5.1	BOOL	FALSE			
		Bit10	%QX5.2	BOOL	FALSE			
		Bit11	%QX5.3	BOOL	FALSE			
		Bit12	%QX5.4	BOOL	FALSE			
		Bit13	%QX5.5	BOOL	FALSE			
		Bit14	%QX5.6	BOOL	FALSE			
		Bit15	%QX5.7	BOOL	FALSE			

11.参照高速计数器控制字，
Controlword %QW2的BIT1为计
数器的复位,高电平生效。



PART 04



本地脉冲轴

● 配置



本地脉冲轴配置

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



The screenshot displays the ICP_PLS software interface. On the left, a tree view shows the project structure under 'TM735本地脉冲输出', including 'Device (TM753)', 'PLC 逻辑', 'Application', 'Axis (STRUCT)', 'Axis_Group (STRUCT)', '库管理器', 'AxisGroup (PRG)', 'SingleAxis (PRG)', '任务配置', 'MainTask', 'SingleAxis', and 'Trace'. The '库管理器' (Library Manager) is selected, showing a list of libraries in the center pane. The 'ICP_PLS, 1.0.0.0 (INVT)' library is highlighted. Below this, a detailed view of the 'ICP_PLS, 1.0.0.0 (INVT)' library is shown, including sub-libraries like 'Auxiliary', 'AxisGroup', 'Basic Types', 'Master/Slave', 'SingleAxis', and 'PEGLV'. The bottom status bar indicates '消息 - 总计0个错误, 0警告, 7条消息'.

名称	命名空间	有效的版本
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.14.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.15.0 (CAA Technical Workgroup)	DED	3.5.15.0
ICP_PLS, 1.0.0.0 (INVT)	ICP_PLS	1.0.0.0
IoStandard = IoStandard, 3.5.15.0 (System)	IoStandard	3.5.15.0
SM3_Basic = SM3_Basic, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Basic	4.10.0.0
SM3_CNC = SM3_CNC, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_CNC	4.10.0.0
SM3_Robotics = SM3_Robotics, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Robotics	4.10.0.0
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

1.通过ICP_PLS本地高速IO库，
可以查阅轴参数配置，轴使能等。

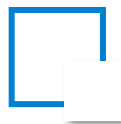


本地脉冲轴配置

工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



2.脉冲轴输出勾选脉冲轴0，脉冲轴1。



```
IMC_Axis_P:ARRAY [0..3] OF IMC_Axis_P; //轴定义
IMC_Power_P:ARRAY [0..3] OF IMC_Power_P; //轴使能
IMC_Jog_P:ARRAY [0..3] OF IMC_Jog_P;
bAxisError: BOOL;
bPositiveEnable,bNegativeEnable: BOOL;
IMC_SendData_P: IMC_SendData_P; //发送数据到底层
Axis:ARRAY[0..3]OF Axis;
IMC_MoveAbsolute_P:ARRAY [0..3] OF IMC_MoveAbsolute_P; //单轴绝对运动
bAbsolute,bAbsolutel:BOOL;
Axis1Pulse1,Pos,AbsPosition,AbsJerk: LREAL;
END_VAR
```

//轴定义

```
IMC_Axis_P[0](AxisID:=PEGVL.AxisID[0] , Execute:=1 , MaxVelocity:= , MaxAcceleration:= , MaxDeceleration:= ,LimitEnable:= , MaxPLimit:= , MaxNLimit:= , PulseData:=10000 , DistanceData:=10 ,
Mode:=1 ,HomeLimit := );
IMC_Axis_P[1](AxisID:=PEGVL.AxisID[1] , Execute:=1 , MaxVelocity:= , MaxAcceleration:= , MaxDeceleration:= ,LimitEnable:= , MaxPLimit:= , MaxNLimit:= , PulseData:=10000 , DistanceData:=10 ,
Mode:=1 ,HomeLimit := );
```

//轴使能

```
IMC_Power_P[0](AxisID:=PEGVL.AxisID[0] , Enable:=IMC_Axis_P[0].Done , AxisError:=bAxisError , AxisEnable:=1 ,);
IMC_Power_P[1](AxisID:=PEGVL.AxisID[1] , Enable:=IMC_Axis_P[1].Done , AxisError:=bAxisError , AxisEnable:=1 ,);
```

//JOG运动

```
IMC_Jog_P[0](AxisID:=PEGVL.AxisID[0] , PositiveEnable:=bPositiveEnable AND IMC_Power_P[0].Status, NegativeEnable:=bNegativeEnable , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , );
IMC_Jog_P[1](AxisID:=PEGVL.AxisID[1] , PositiveEnable:=bPositiveEnable AND IMC_Power_P[1].Status, NegativeEnable:=bNegativeEnable , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , );
```

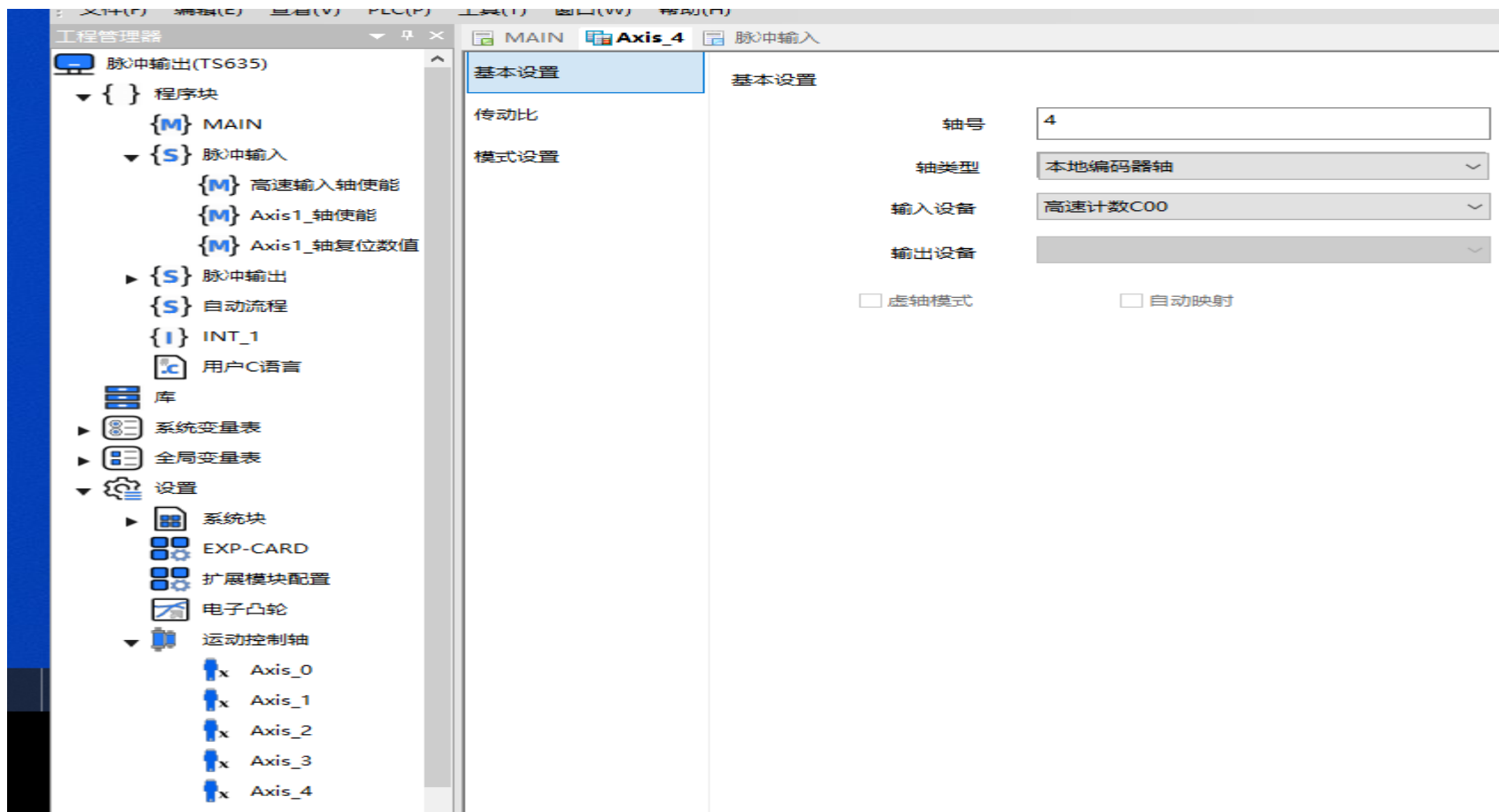
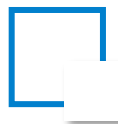
//绝对运动

```
IMC_MoveAbsolute_P[0](AxisID:=PEGVL.AxisID[0] , Execute:=bAbsolute , Position:=AbsPosition , Velocity:=800 , Acceleration:= , Jerk:= , );
IMC_MoveAbsolute_P[1](AxisID:=PEGVL.AxisID[0] , Execute:=bAbsolute , Position:=AbsPosition , Velocity:=800 , Acceleration:= , Jerk:= , );
```

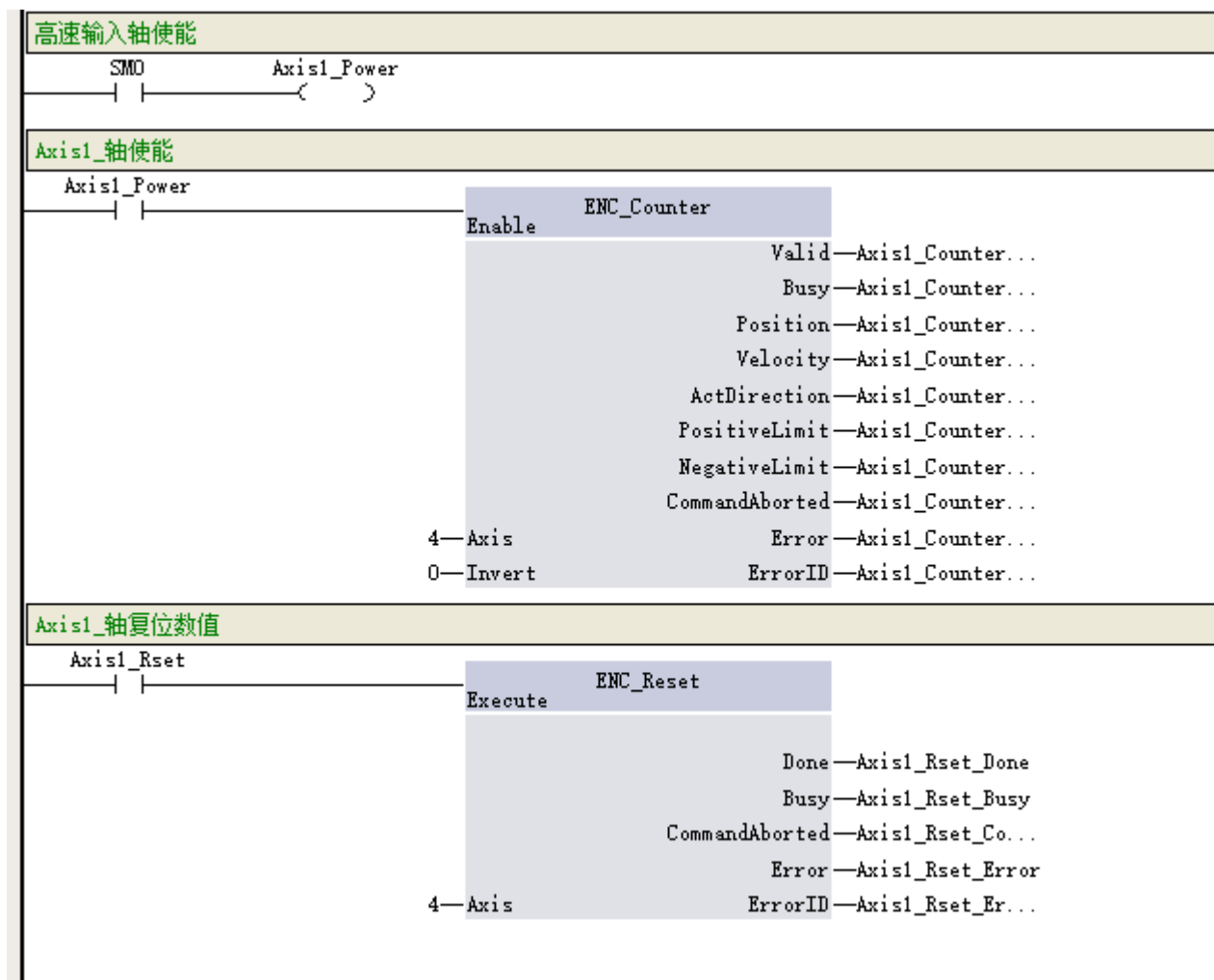
//数据发送

```
IMC_SendData_P(Axis1Pulse=>Axis[0].Pulse , Axis2Pulse=>Axis[1].Pulse , Axis3Pulse=>Axis[2].Pulse , Axis4Pulse=>Axis[3].Pulse , Axis1Direction=>Axis[0].Direction , Axis2Direction=>Axis[1].Direction , Axis3Direction=>Axis[2].Direction , Axis4
```

3.编写TM735 PLC程序。



4.TS635配置编码器轴，
工程管理器-设置-运动
控制轴-右键新增，轴类
型选择本地编码器轴。



5.TS635梯形图程序，
编码器轴使能高速计数。



```
1 //轴定义
2 IMC_Axis_P[0](AxisID:=_mcAxis_0:=PEGV.L.AxisID[0], Execute:=TRUE, MaxVelocity:=, MaxAcceleration:=, MaxDeceleration:=, LimitEnable:=, MaxPLimit:=, MaxNLimit:=, PulseData:=10000, DistanceData:=10,
3 Mode:=_mcCWCCW:=1, HomeLimit:=);
4 IMC_Axis_P[1](AxisID:=_mcAxis_1:=PEGV.L.AxisID[1], Execute:=TRUE, MaxVelocity:=, MaxAcceleration:=, MaxDeceleration:=, LimitEnable:=, MaxPLimit:=, MaxNLimit:=, PulseData:=10000, DistanceData:=10,
5 Mode:=_mcCWCCW:=1, HomeLimit:=);
6
7 //轴使能
8 IMC_Power_P[0](AxisID:=_mcAxis_0:=PEGV.L.AxisID[0], Enable:=IMC_Axis_P[0].Done, AxisError:=bAxisError, AxisEnable:=TRUE, );
9 IMC_Power_P[1](AxisID:=_mcAxis_1:=PEGV.L.AxisID[1], Enable:=IMC_Axis_P[1].Done, AxisError:=bAxisError, AxisEnable:=TRUE, );
10
11 //JOG运动
12 IMC_Jog_P[0](AxisID:=_mcAxis_0:=PEGV.L.AxisID[0], PositiveEnable:=bPositiveEnable AND IMC_Power_P[0].Status, NegativeEnable:=bNegativeEnable, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, );
13 IMC_Jog_P[1](AxisID:=_mcAxis_1:=PEGV.L.AxisID[1], PositiveEnable:=bPositiveEnable AND IMC_Power_P[1].Status, NegativeEnable:=bNegativeEnable, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, );
14
15 //相对运动
16 IMC_MoveRelative_P[0](AxisID:=_mcAxis_0:=PEGV.L.AxisID[0], Execute:=bRelative, Position:=Pos, Velocity:=, Acceleration:=, Jerk:=, );
17 IMC_MoveRelative_P[1](AxisID:=_mcAxis_1:=PEGV.L.AxisID[1], Execute:=bRelative, Position:=Pos, Velocity:=, Acceleration:=, Jerk:=, );
18
19 //数据发送
20 IMC_SendData_P(Axis1Pulse:=Axis[0].Pulse, Axis2Pulse:=Axis[1].Pulse, Axis3Pulse:=Axis[2].Pulse, Axis4Pulse:=Axis[3].Pulse, Axis1Direction:=_mcForward:=Axis[0].Direction
21
22
```

6.使能bRelative,
Position设置值为123。



本地脉冲轴配置

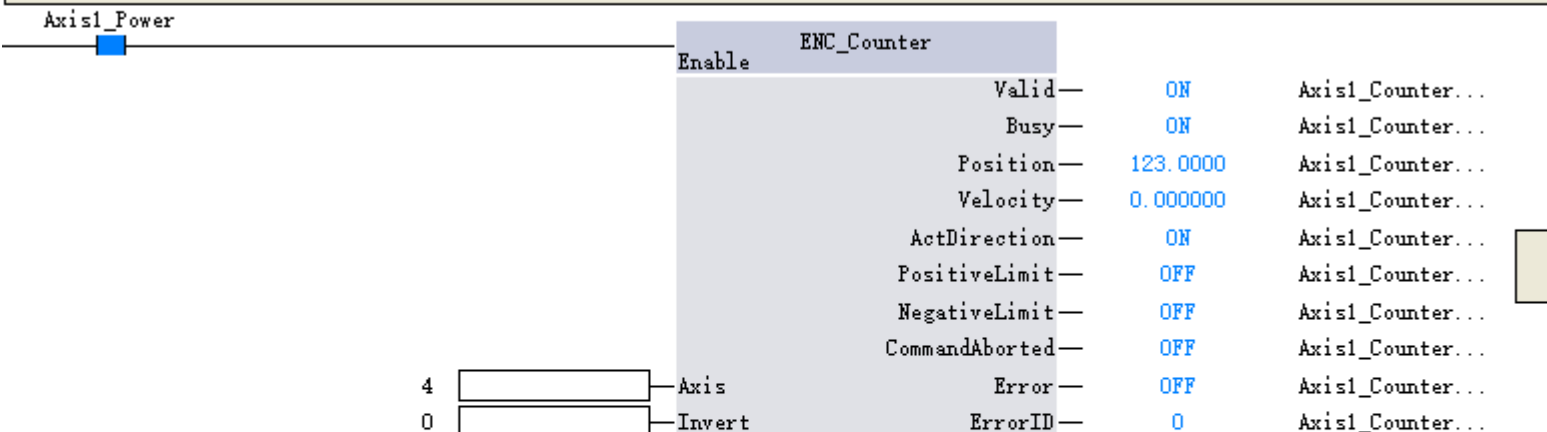
工业自动化 | 网络能源 | 新能源汽车 | 光伏储能



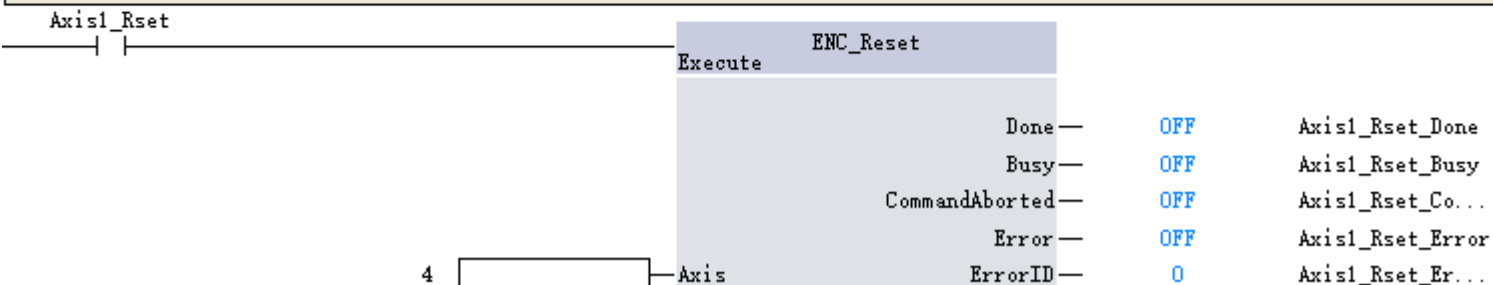
高速输入轴使能



Axis1_轴使能



Axis1_轴复位数值



7.TS635接收计数值为
123.0。

INVT Cloud



**值得信赖的
工控与能效解决方案提供者**