

# 英威腾中大型PLC

## 软件手册





# 前言

## 概述

感谢您使用英威腾中大型 PLC 产品。

本手册记载了使用英威腾中大型 PLC 所必需的信息。使用前请详细阅读本手册，充分理解其功能和性能，完成系统构建，发挥其优越性能。

## 读者对象

本手册适用于具有电工专业知识的人员（合格的电气工程师或具有同等知识的人员）。

## 适用产品

TM、TP 系列 PLC 控制器。

## 关于手册获取

本手册不随产品发货，如需获取电子版 PDF 文件，可以通过以下方式获取：

登录我司官网（[www.invt.com.cn](http://www.invt.com.cn)）→服务与支持→资料下载→搜索关键字并下载。

使用手机扫描产品机身二维码→搜索关键字并下载。

## 修改记录

由于产品版本升级或其他原因，本文档会不定期更新，恕不另行通知。

编号	修改内容摘要	版本	日期
1	创建	V1.0	2024.08



# 目录

<b>1 PLC 控制器简介 .....</b>	<b>1</b>
1.1 概述 .....	1
1.1.1 产品简介 .....	1
1.1.2 产品规格 .....	1
1.1.3 产品配置及模块说明 .....	2
1.1.4 系统应用流程 .....	4
1.2 Invtmatic Studio 概述 .....	4
1.2.1 软件安装与卸载 .....	4
1.2.2 与硬件连接 .....	9
1.3 PC 端通信配置 .....	10
1.3.1 PC 与 TM700 通信配置 .....	10
1.3.2 PC 与 TP6000 通信配置 .....	16
<b>2 入门指引 .....</b>	<b>18</b>
2.1 创建工程 .....	18
2.1.1 启动编程环境 .....	18
2.1.2 新建工程 .....	20
2.2 工程编写典型步骤 .....	21
2.3 程序编写与调试实例 .....	22
2.3.1 添加设备 .....	22
2.3.2 编写功能处理 POU .....	26
2.3.3 电机参数的设置 .....	27
2.3.4 电机正反转编写 .....	29
2.3.5 用户程序编译 .....	30
2.3.6 监控程序运行 .....	31
<b>3 基本功能 .....</b>	<b>33</b>
3.1 界面布局 .....	33
3.2 编译菜单 .....	33
3.3 变量使用情况表 .....	34
3.3.1 概述 .....	34
3.3.2 功能介绍 .....	34
3.3.3 菜单选项 .....	35
3.3.4 PLC 直接地址存储区域 .....	36
3.4 故障诊断 .....	36
3.5 自动扫描 .....	38
3.6 交叉引用 .....	38
3.7 监控表 .....	39
3.8 采样跟踪 .....	40
3.9 保持变量 .....	43
3.9.1 特性 .....	43
3.9.2 掉电保持变量表 .....	44
3.10 配方管理器 .....	45
3.11 符号配置 .....	49
3.12 任务配置 .....	55
3.12.1 添加任务 .....	56
3.12.2 任务设置 .....	57



3.12.3 TP6000 多核任务配置 .....	58
<b>4 硬件配置 .....</b>	<b>59</b>
4.1 本体高速 I/O 配置 .....	59
4.1.1 计数器界面组态 .....	59
4.1.2 计数功能 .....	61
4.1.3 输出端口功能说明 .....	63
4.1.4 普通输出功能 .....	63
4.1.5 高速脉冲输出功能 .....	63
4.1.6 比较输出功能 .....	64
4.1.7 外部中断说明 .....	65
4.2 I/O 本地扩展模块配置 .....	66
4.2.1 扩展模块组态 .....	66
4.2.2 数字量输入模块 .....	71
4.2.3 数字量输出模块 .....	72
4.2.4 模拟量输入模块 .....	73
4.2.5 模拟量输出模块 .....	75
4.2.6 温度模块 .....	77
4.3 各模块优先级设置（推荐值） .....	80
<b>5 网络配置 .....</b>	<b>81</b>
5.1 Modbus TCP .....	81
5.1.1 Modbus TCP 主站配置 .....	81
5.1.2 Modbus TCP 主站通信配置 .....	82
5.1.3 Modbus TCP 从站配置 .....	85
5.1.4 Modbus TCP 设备诊断 .....	86
5.2 Modbus RTU .....	86
5.2.1 Modbus RTU 主站配置 .....	87
5.2.2 Modbus RTU 主站通信配置 .....	88
5.2.3 Modbus RTU 从站配置 .....	91
5.2.4 Modbus RTU 设备诊断 .....	92
5.2.5 Modbus RTU 常见故障 .....	92
5.3 EtherCAT 主站 .....	92
5.3.1 EtherCAT 主站配置 .....	93
5.3.2 EtherCAT 从站配置 .....	96
5.3.3 EtherCAT 冗余环网功能 .....	105
5.3.4 EtherCAT 双主站 .....	106
5.4 CANopen .....	107
5.4.1 CANopen 主站配置 .....	108
5.4.2 CANopen 主站相关参数配置 .....	109
<b>6 诊断 .....</b>	<b>111</b>
6.1 诊断简介 .....	111
6.1.1 故障诊断 .....	111
6.2 设备自身诊断信息列表 .....	112
6.2.1 I/O 诊断 .....	112
6.2.2 Modbus RTU 诊断 .....	112
6.2.3 Modbus TCP 诊断 .....	112
6.3 在线日志 .....	112
6.4 诊断编程接口 .....	113
6.5 应用程序诊断 .....	114
6.6 PLC 程序运行异常处理 .....	116
6.6.1 常见的异常现象及处理措施 .....	116



6.6.2 因程序问题导致 PLC 失控的处理措施..... 119

6.7 设备故障诊断代码（背板总线仅适用 TM 系列 PLC） ..... 121

**7 升级与 PLC 设置 .....126**

7.1 软件升级..... 126

7.2 固件升级..... 127

7.3 时间设置..... 128

7.4 网络设置..... 128

7.5 恢复出厂设置..... 129

7.6 文件传输..... 129

**附录 A 工程实例.....130**

A.1 控制器与 Goodrive20 系列变频器 485 通讯配置实例..... 130

A.2 控制器与 DA200 系列伺服驱动器配置实例 ..... 134

A.3 控制器与 DA200 系列伺服 CANopen 配置实例 ..... 141

**附录 B SMC\_ERROR 说明.....145**



# 1 PLC 控制器简介

## 1.1 概述

### 1.1.1 产品简介

英威腾中大型 PLC 控制器目前包括 TM700 和 TP6000 两大系列，采用 IEC61131-3 编程语言体系，支持 IL、LD、FBD、ST、SFC、CFC 六种标准编程语言，通过 EtherCAT 总线可实现电子凸轮、电子齿轮、同步控制、定位等高阶运动控制功能，具备丰富通信接口，搭配功能丰富的 I/O 模块，为用户提供灵活、智能的自动化解决方案，满足用户多样化应用需求。

TM700 系列是采用模块化结构设计的高性能 PLC 控制器，主要应用于有较高运动控制要求和较复杂控制网络场景，在控制性能、通信能力、编程效率等方面有大幅提升和改善，让您更灵活的搭建控制网络，通过 OPC UA 更便捷的实现与信息层的数据交互，进一步提升设备节拍、缩短开发周期，带来更卓越的体验。

TP6000 系列基于 X86 架构和 EtherCAT 总线技术，以极高的性能、丰富的接口和功能满足现场高标准控制系统需求，双 EtherCAT 接口可实现多任务分配、环网控制，带来更可靠的解决方案，广泛应用于各类高速度、高精度、高响应设备及中大型生产线。。

### 1.1.2 产品规格

型号	TM750	TM751	TM752	TM753	TP6210-1201	TP6211-2201	TP6214-2201
额定工作电压	DC24V(-15%~+20%)						
存储器							
程序容量	20MB				256MB		
数据容量	64MB				256MB		
掉电保持数据容量	1MB				5MB		
扩展 SD 卡最大容量	32G				-		
I/O							
本地 IO 模块数量	16				-		
高速输入	4 组高速计数，支持单相、A/B 相、CW/CCW、脉冲+方向，其中 A/B 相支持 1 倍频、2 倍频和 4 倍频				2 组 AB 相或 1 组 ABZ 差分输入		
高速输出	8 通道 200kHz 漏型高速输出，支持 4 轴脉冲运动控制				-		
支持 IO 中断	8 通道高速中断				-		
PWM 输出	4 通道 PWM 输出				-		
通讯网络和接口							
Ethernet	×2, RJ45, 100Base-TX 支持 PLC 软件下载、Modbus TCP、TCP/IP、OPC UA 协议、EtherNet/IP						
EtherCAT	×1, RJ45, 100Base-TX, 两从站距离小于 100m				×2, RJ45, 100Base-TX, 两从站距离小于 100m		×2, RJ45, 支持双主站，

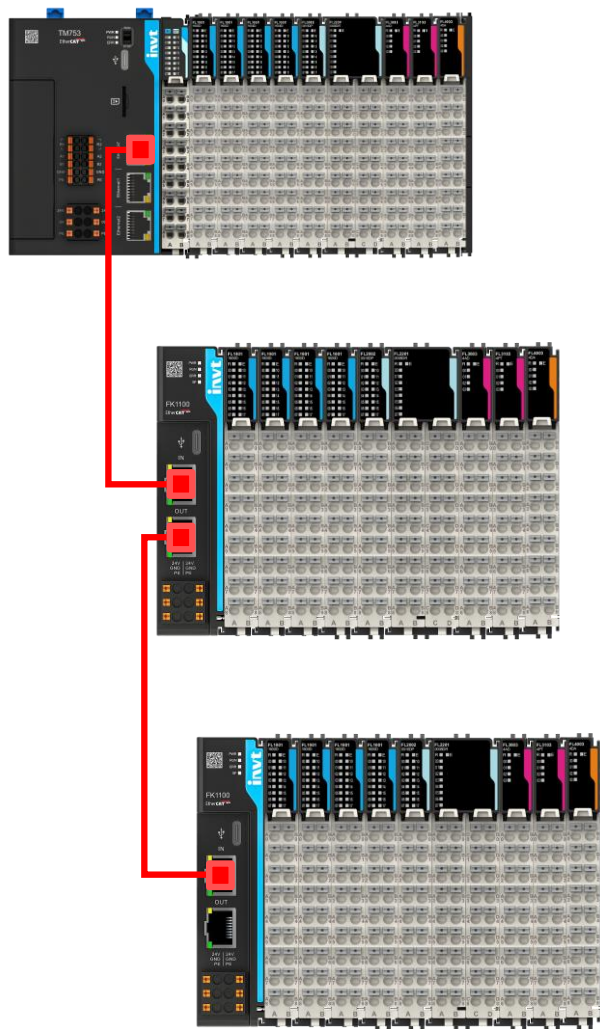


型号	TM750	TM751	TM752	TM753	TP6210-1201	TP6211-2201	TP6214-2201
							100Base-TX, 两从站距离小于 100m
串行通信 (RS485)	×2, Modbus RTU 主站/从站, 直插端子						
USB	×1, Type-C, PC 通讯, 程序下载及调试				×4, USB3.0, 文件拷贝		
存储卡	×1, Micro SD, 用于升级应用程序				x	x	x
通信扩展	CANopen, 4G, WI-FI				x	x	x
运动控制							
最大控制轴数	4	8	16	32	64	128	256
运控功能	点位运动、插补运动(直线、圆弧等)、电子齿轮、电子凸轮(飞剪、追剪等)、轴组、CNC 等						
组态编程							
编程语言	IL、ST、FBD、LD、CFC、SFC						

1.1.3 产品配置及模块说明

TM700 系列 CPU 本体可直接连接 Flex 系列 I/O 上, 也可以通过通信耦合器扩展到 Flex 系列 I/O 上, 如下图所示。

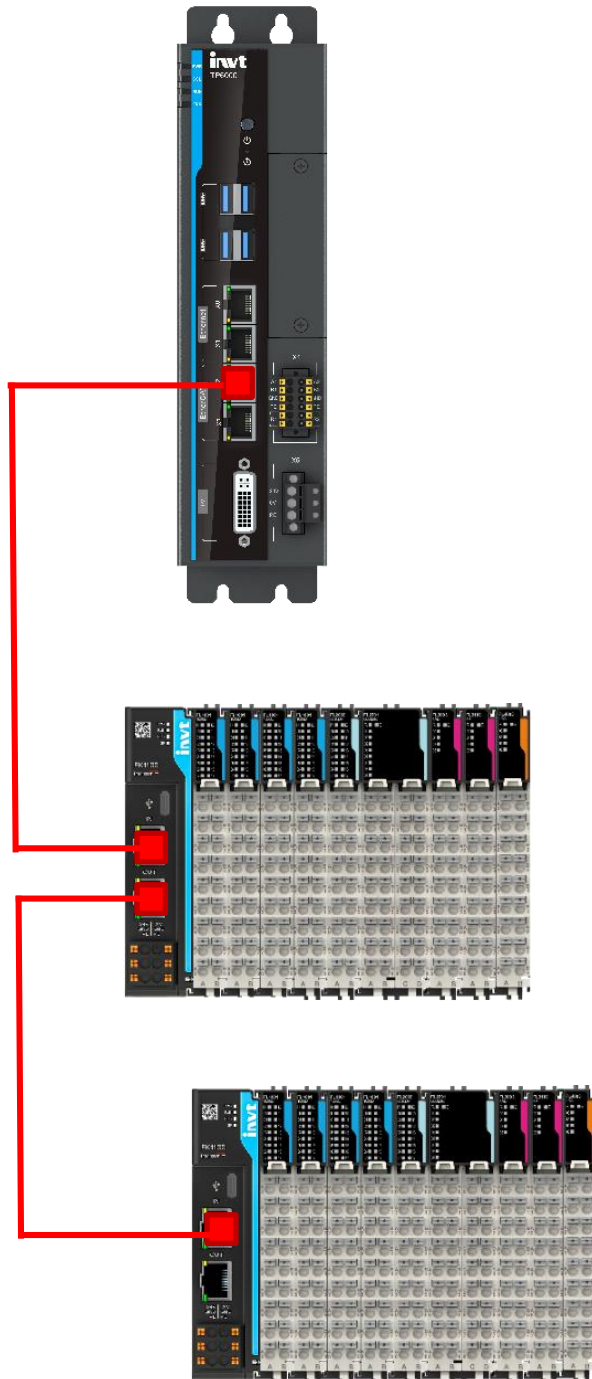
图 1-1 系统集成示意图 1





TP6000 系列 CPU 本体无法直接连接到 Flex 系列 I/O 产品上，只能通过通信耦合器扩展 Flex 系列 I/O 上，如下图所示。

图 1-2 系统集成示意图 2





### 1.1.4 系统应用流程



## 1.2 Invtmatic Studio 概述

Invtmatic Studio 是深圳市英威腾电气股份有限公司开发的编程平台, 采用 IEC61131-3 编程语言体系, 支持 IL、LD、FBD、SFC、ST、CFC 六种标准编程语言。

### 1.2.1 软件安装与卸载

#### 1.2.1.1 软件获取

英威腾 PLC 控制器用户编程软件采用 Invtmatic Studio 平台, 安装文件以及相关参考资料等, 用户可通过以下 2 种途径获取:

- 1、访问 [www.invt.com.cn](http://www.invt.com.cn), 选择服务与支持, 在资料下载页面免费下载软件安装包。
- 2、从英威腾各级经销商处获得软件安装光盘。

#### 1.2.1.2 软件安装要求

具备以下条件的台式 PC 或便携式 PC:

- 操作系统: Windows 7/Windows 8/Windows 10
- CPU 主频: 2GHz 以上
- 内存: 2GB 以上
- 空间: 可用硬盘空间 5G 以上
- 安装要求:
  - ✧ 使用管理员账户
  - ✧ 关闭杀毒软件
  - ✧ 关闭加密系统
  - ✧ 请勿使用中文路径



### 1.2.1.3 安装准备

- 若为首次安装 Invtmatic Studio，请先查看个人电脑的硬件条件是否具备上述的软件安装要求，确认满足安装条件后，直接安装即可。
- 若想安装最新版本 Invtmatic Studio，可先查看本机安装的 Invtmatic Studio 版本信息。打开软件，查看**帮助>关于**，若不是最新版本可采用在线升级模式升级软件。

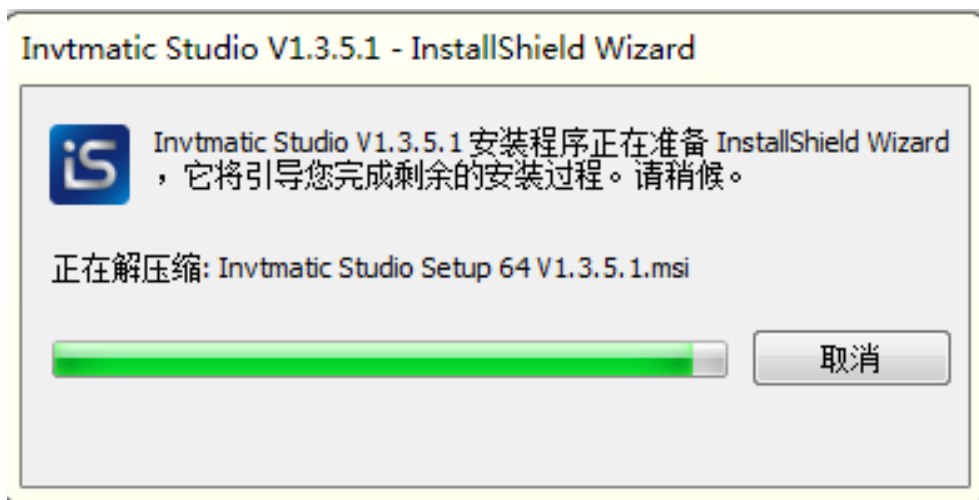
图 1-3 版本信息显示界面



### 1.2.1.4 安装步骤

- 步骤1 打开安装文件所在位置，双击打开 Invtmatic Studio Setup 64 Vx.x.x.exe 文件（以 V1.3.5 为例）。
- 步骤2 双击打开后，启动安装，可以看到如下界面，进入安装准备阶段。

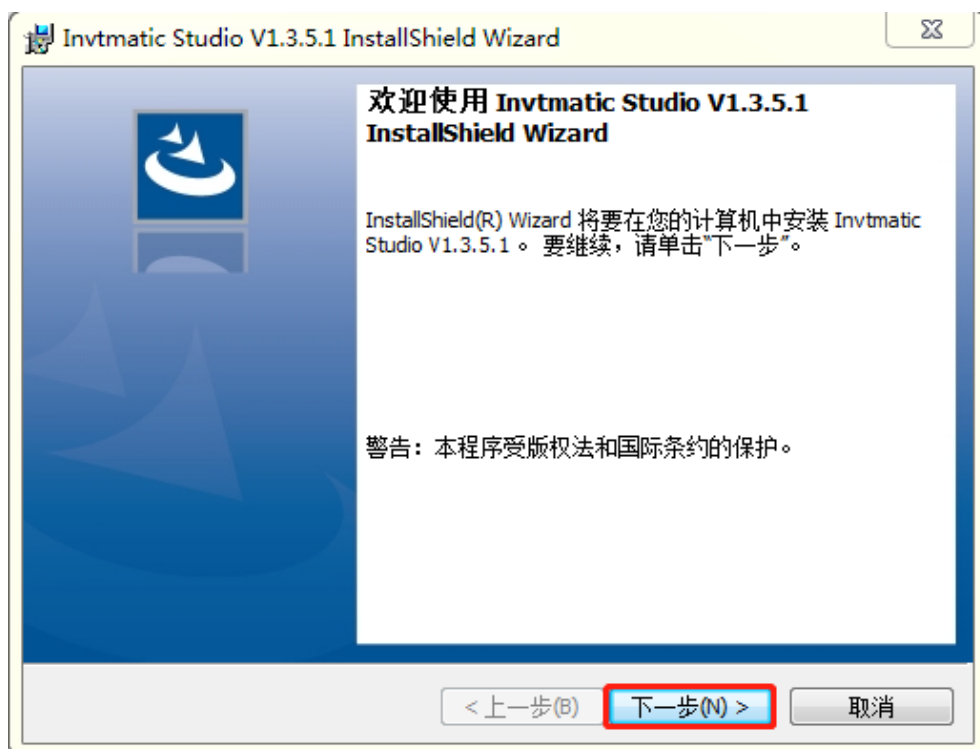
图 1-4 安装准备界面





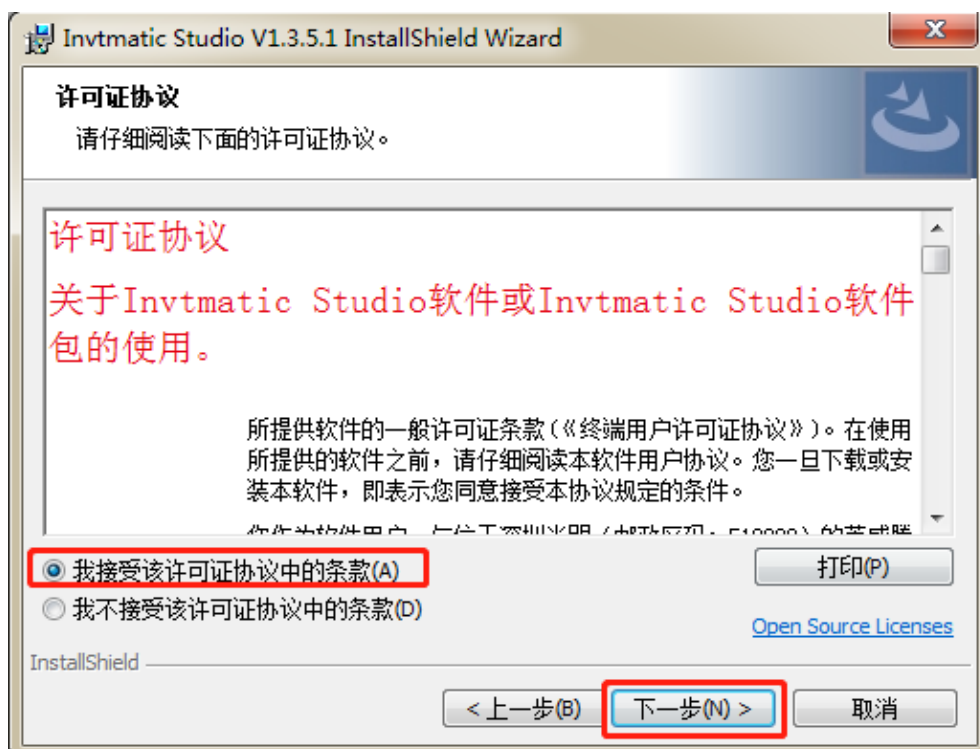
步骤3 出现如下提示界面，点击**下一步**开始安装。

图 1-5 安装向导界面



步骤4 进入许可证协议界面，勾选**我接受该许可证协议中的条款(A)**，然后点击**下一步**。

图 1-6 许可证协议界面

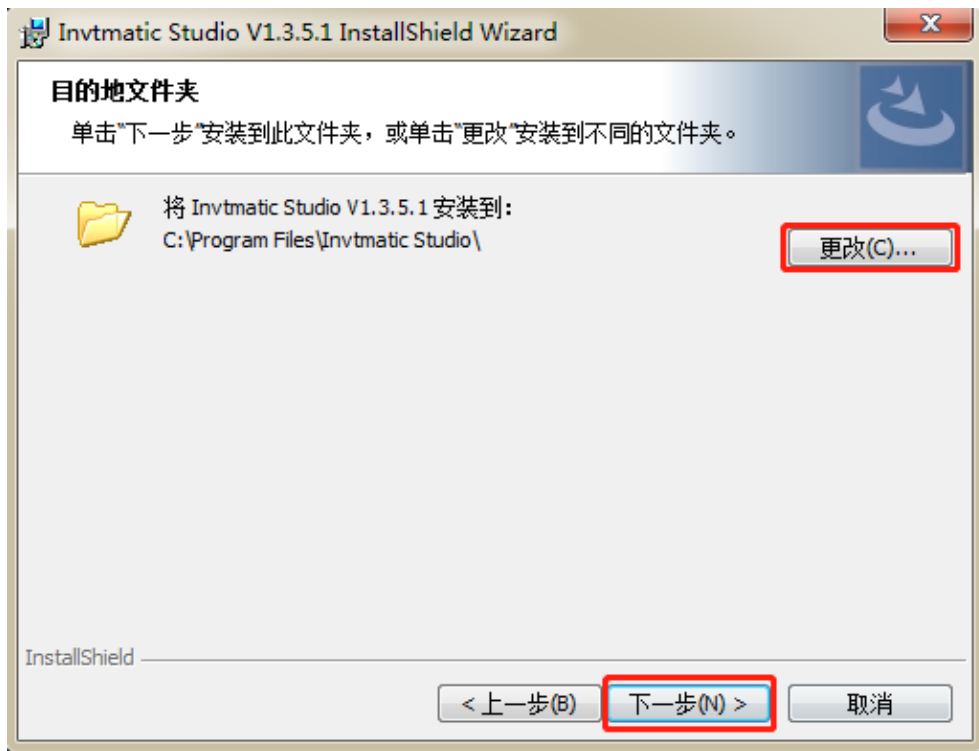


步骤5 设置好软件安装路径后，点击**下一步**。

**注意：**软件安装路径不能使用中文。

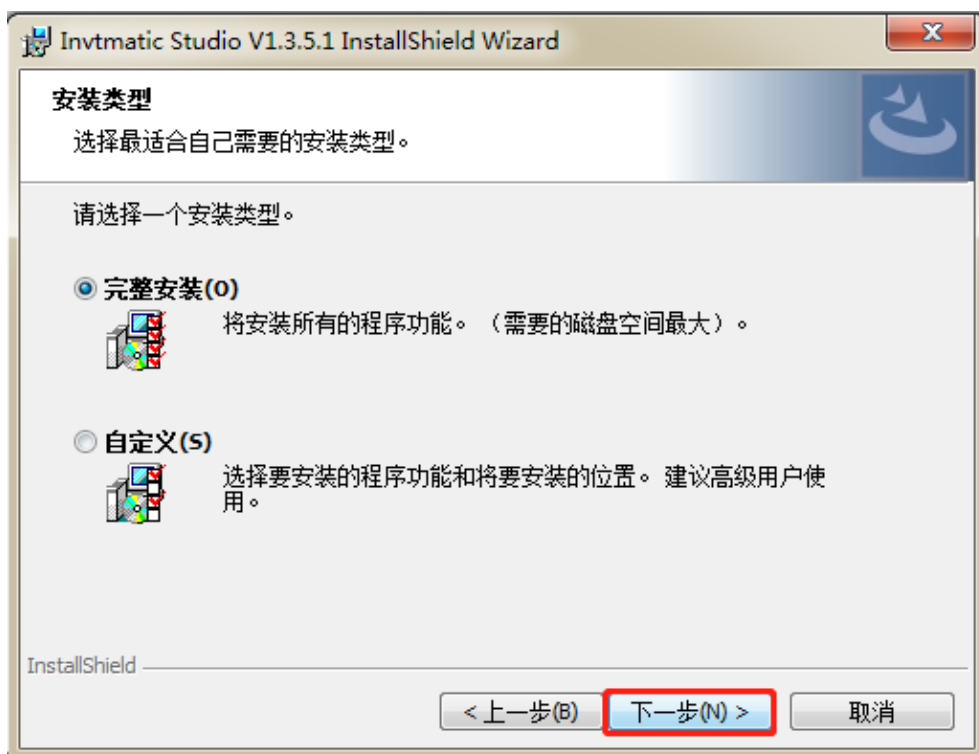


图 1-7 确认安装路径



步骤6 进入安装组件选择界面，可选择完整安装和自定义安装，若无特殊需求，按默认勾选即可，点击**下一步**。

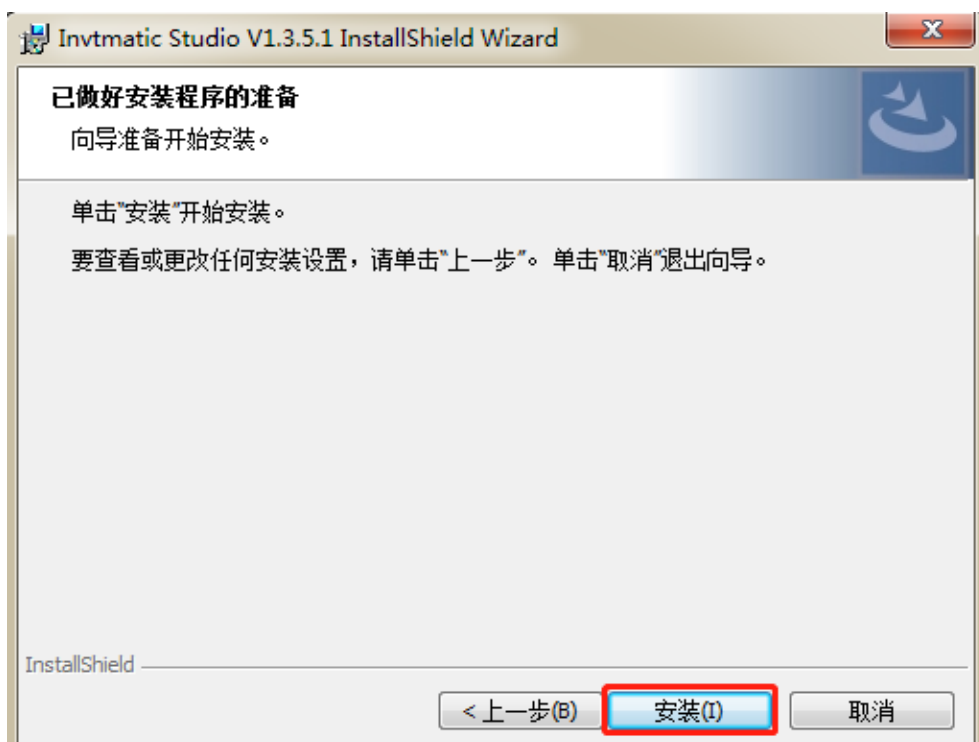
图 1-8 安装类型选择界面





步骤7 出现如下提示界面，点击**安装**。

图 1-9 准备安装界面



步骤8 出现如下界面，等待安装进度条，直到出现安装完成界面，点击**完成**，完成 Invntmatic Studio 的安装。

图 1-10 安装进度界面

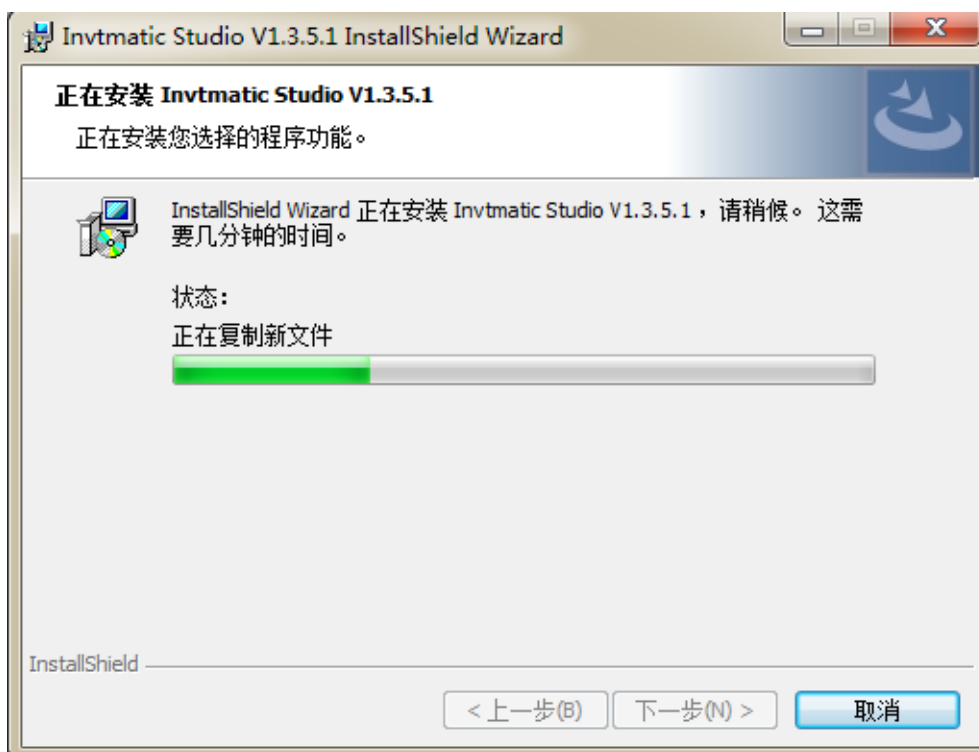




图 1-11 安装完成界面



### 1.2.1.5 卸载步骤

使用标准的 Windows 系统卸载软件方法卸载 Invtmatic Studio 即可，卸载步骤如下：

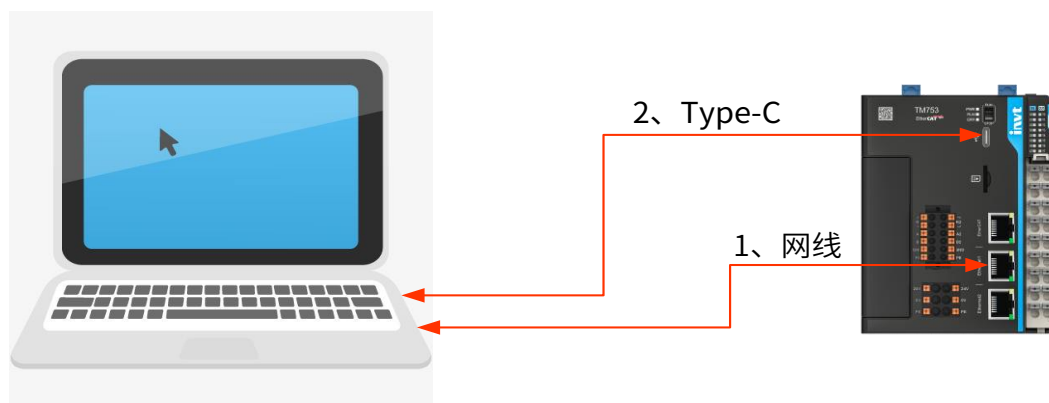
- 步骤1 关闭 Invtmatic Studio 运行程序，包括后台运行程序。
- 步骤2 进入控制面板，找到 Invtmatic Studio 程序，右键单击，选择**卸载**。
- 步骤3 确认并等待程序卸载完毕。

### 1.2.2 与硬件连接

上位机与 TM700 系列控制器的硬件连接方式有两种：

- 1、采用 LAN 网络电缆连接
- 2、采用 Type-C 线缆连接

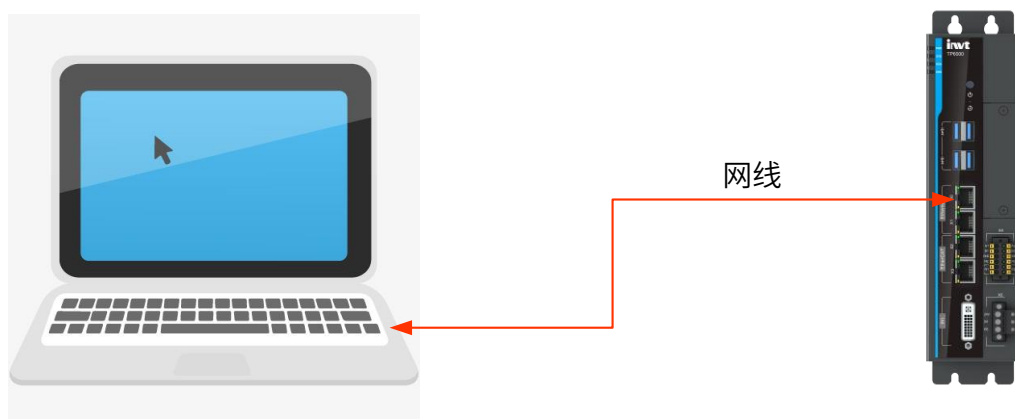
图 1-12 上位机与 TM700 系列控制器的硬件连接





上位机与 TP6000 系列控制器的硬件连接方式只能通过 LAN 网络线缆一种方式连接。

图 1-13 上位机与 TP6000 系列控制器的硬件连接



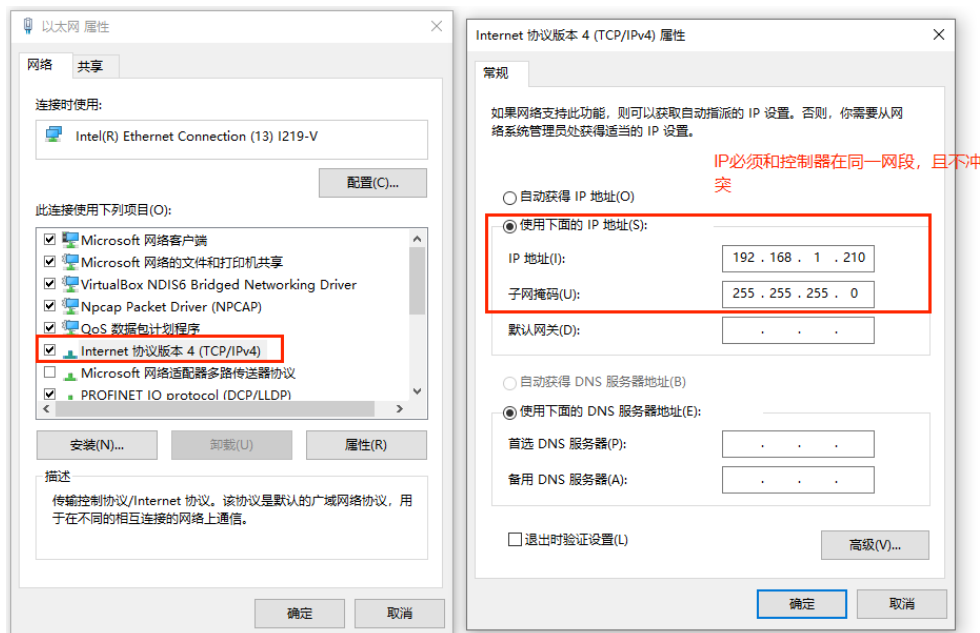
## 1.3 PC 端通信配置

### 1.3.1 PC 与 TM700 通信配置

#### ■ 采用 LAN 网络电缆连接

采用 LAN 网络电缆连接，需保证 PC 端 IP 地址和控制器的 IP 地址位于同一网段，TM700 系列可编程控制器的出厂默认 IP 地址为：Ethernet1: 192.168.1.10; Ethernet2: 192.168.2.10，都具备程序调试和下载功能，请确保网线连接到 Ethernet1/ Ethernet2，将 PC 端 IP 地址应设为 192.168.1.xxx/192.168.2.xxx。（xxx 表示除 10 外 1~254 范围内任一整数）

图 1-14 TM700 采用 LAN 网络电缆连接 PC 端通信配置



#### ■ 采用 Type-C 线缆连接

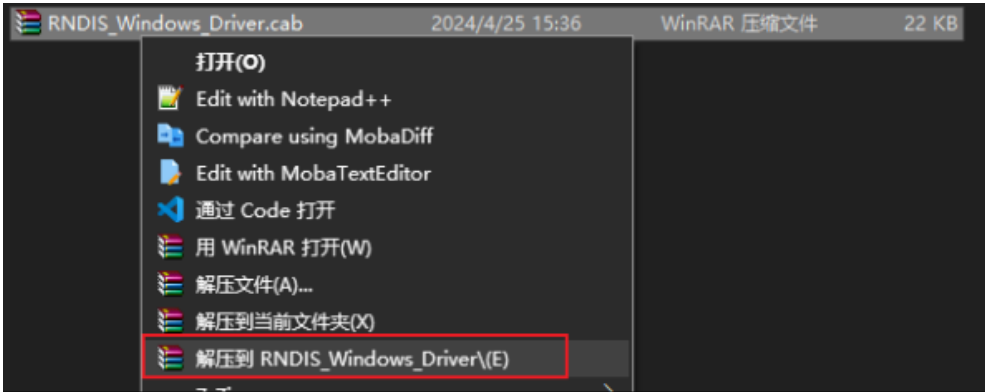
- usb 驱动安装

以 PC 端操作系统为 Windows10 时，安装 usb 驱动步骤如下：



此处以 kindle\_rndis.inf\_amd64 的 usb 驱动程序文件为例。

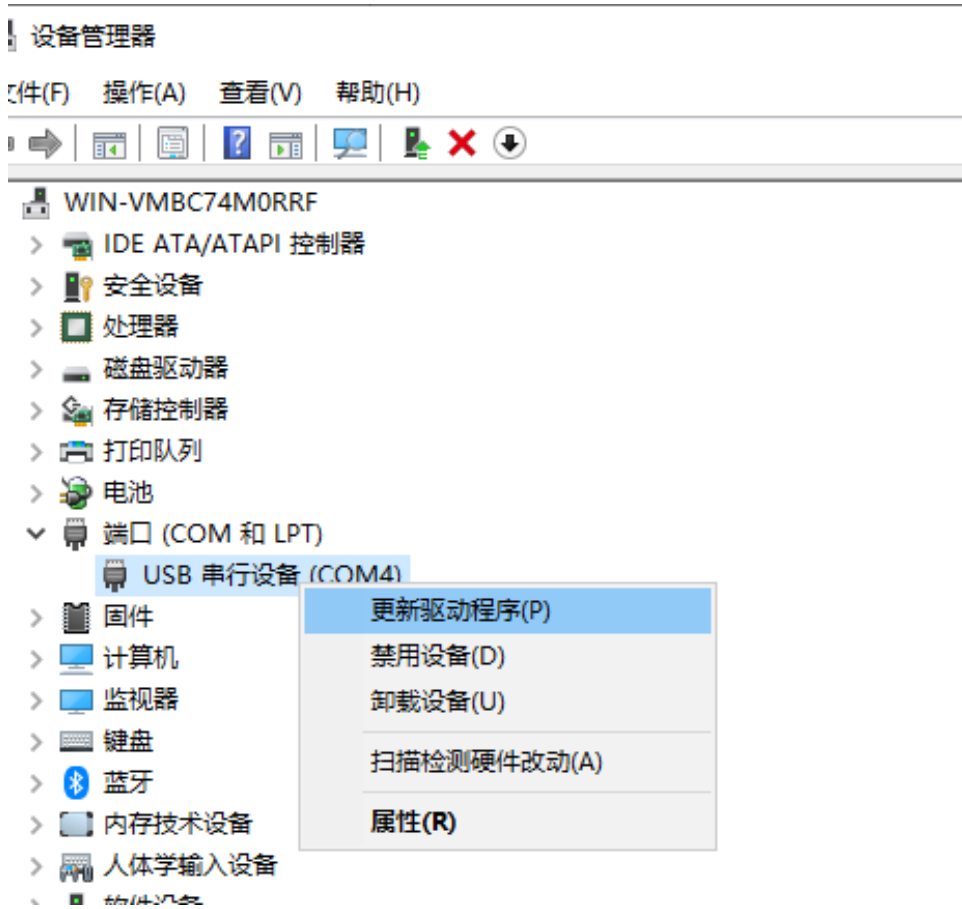
步骤1 右键点击文件"RNDIS\_Windows\_Driver.cab"，解压此文件。



步骤2 出现以下界面后，按任意键继续。

RNDIS_Windows_Driver\<div>The image shows a Windows Explorer window displaying the contents of the 'RNDIS_Windows_Driver' folder. The table below represents the data shown in the window's file list.</div>				
Name	Date modified	Type	Size	
rndis.cat	3/12/2010 10:59 PM	Security Catalog	7 KB	
RNDIS.inf	2/3/2010 1:42 AM	INF File	4 KB	

步骤3 连接电脑和 PLC 之间的 usb 数据线：打开设备管理器，选择端口>USB 串行设备，右键点击更新驱动程序。





步骤4 选择**浏览我的电脑以查找驱动程序**，选择驱动文件夹。

更新驱动程序 - USB 串行设备 (COM4)

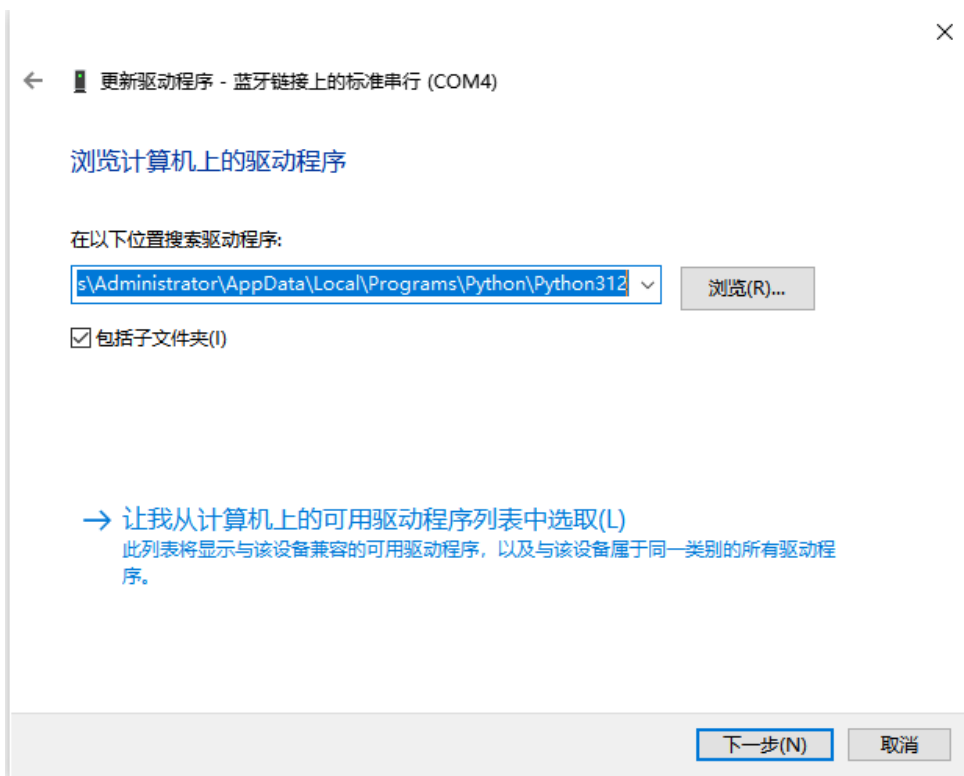
### 你要如何搜索驱动程序？

#### → 自动搜索驱动程序(S)

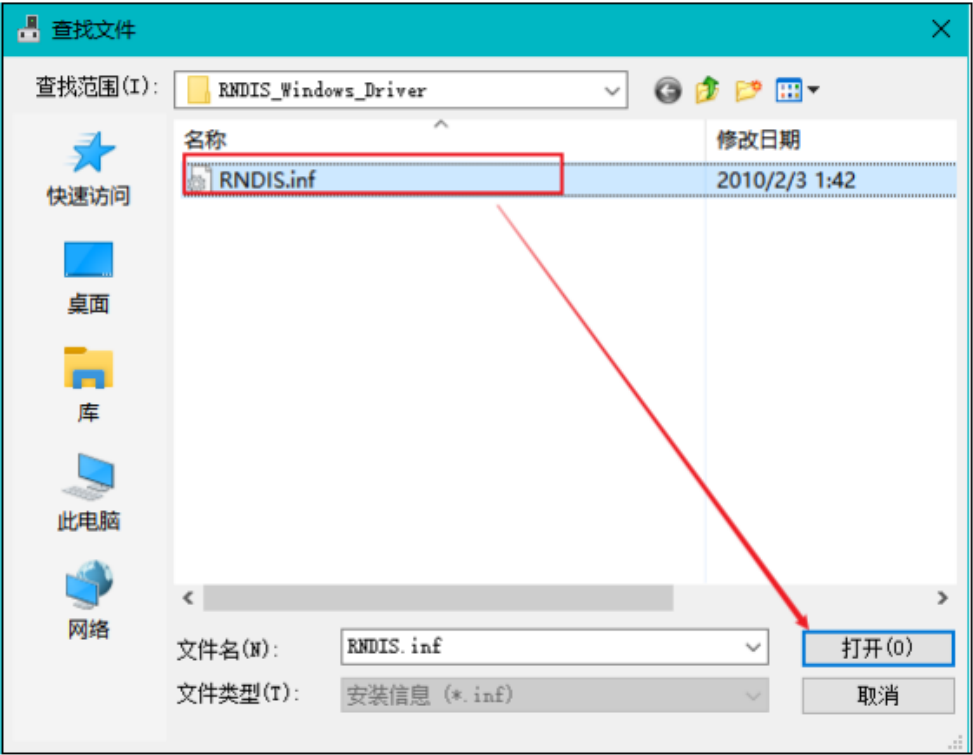
Windows 将在你的计算机中搜索最佳可用驱动程序，并将其安装在你的设备上。

#### → 浏览我的电脑以查找驱动程序(R)

手动查找并安装驱动程序。





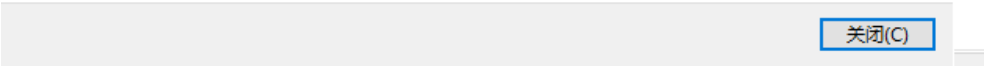
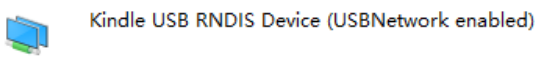


步骤5 安装成功后点击**关闭**。



Windows 已成功更新你的驱动程序

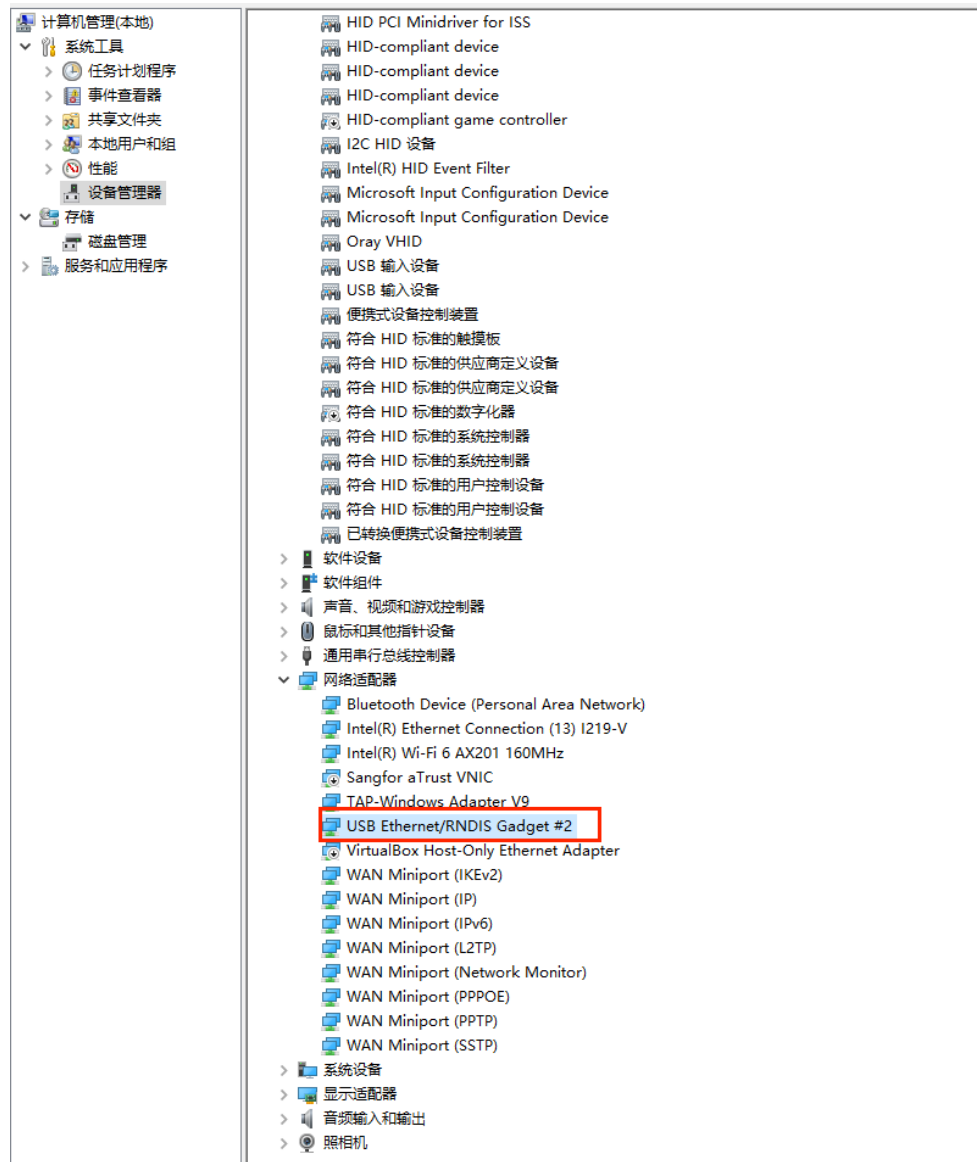
Windows 已安装完此设备的驱动程序:



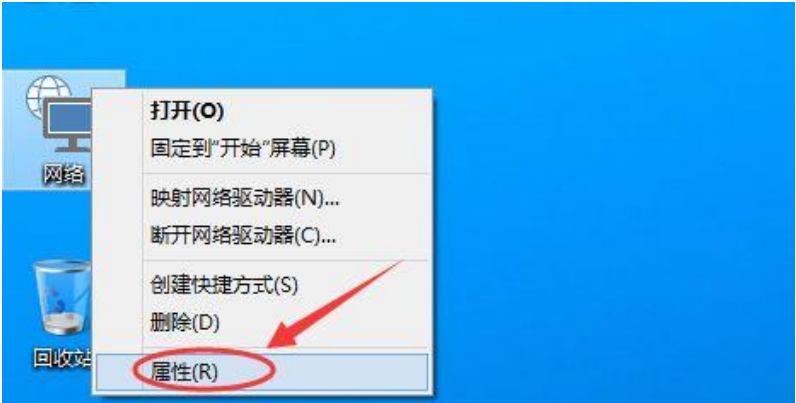


● USB 网口配置

设备管理器的网络适配器下增加网络适配器 USB RNDIS 项。

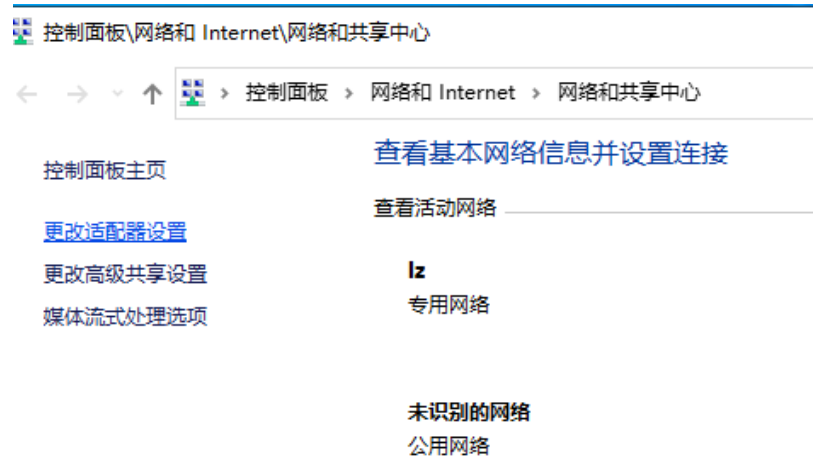


步骤1 右键点击网络>属性。





步骤2 单击更改适配器设置。

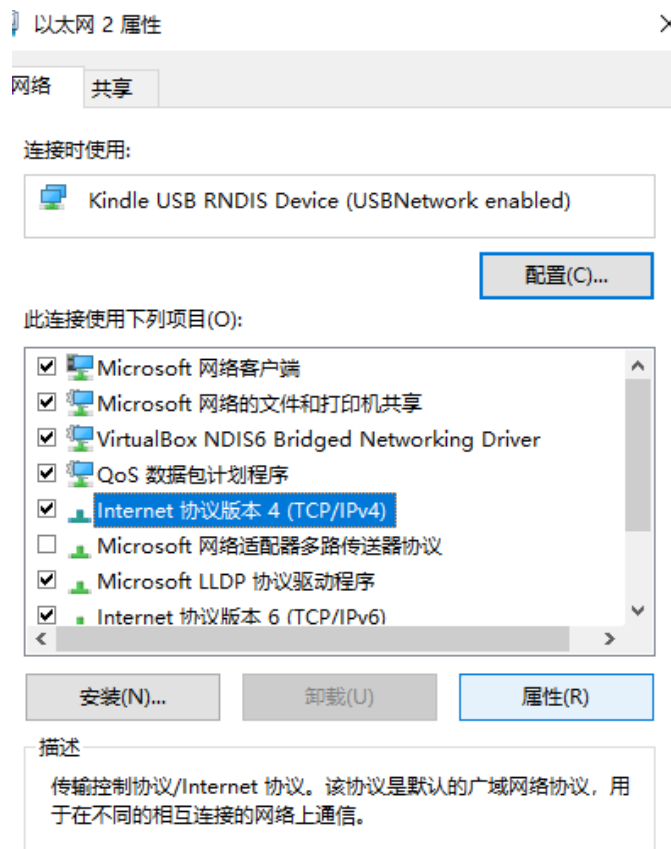


步骤3 右键单击未识别的网络（带 USB RNDIS 标识）>属性。



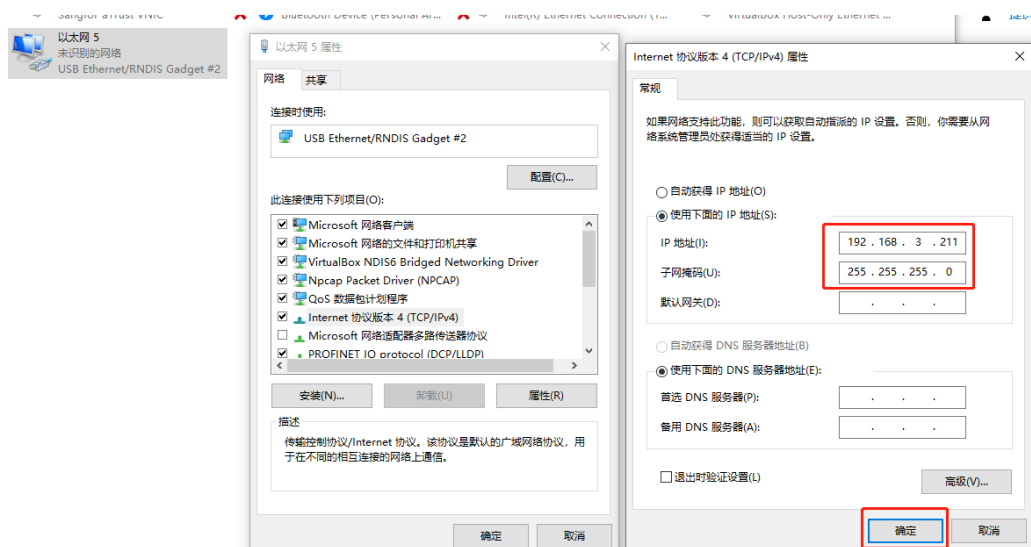


步骤4 勾选 **Internet 协议版本 4** 后，点击**属性**设置 IP 地址。



步骤5 设置手动 IP 地址。

**注意：**IP 地址设置必须在 192.168.3.xxx (xxx 表示除 10 外 1~254 范围内任一整数) 网段。

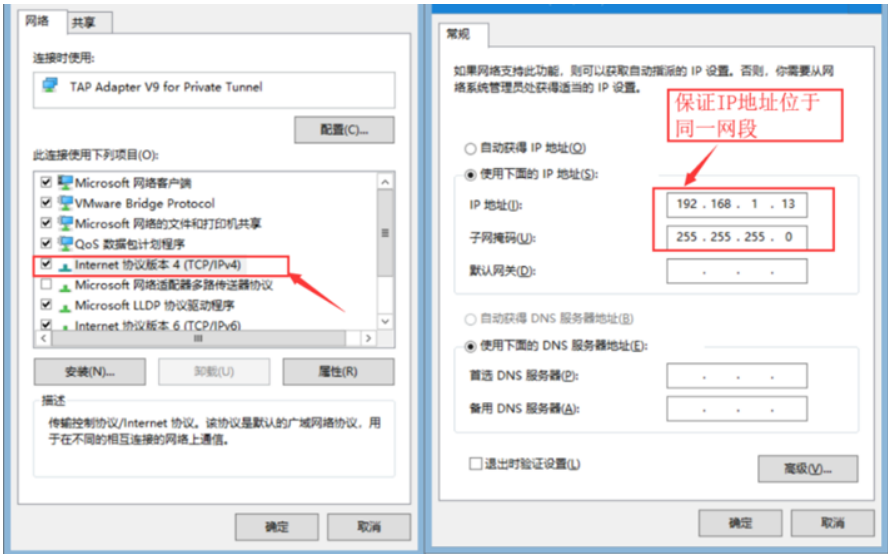


### 1.3.2 PC 与 TP6000 通信配置

采用 LAN 网络电缆连接，需保证 PC 端 IP 地址和控制器的 IP 地址位于同一网段，TM700 系列可编程控制器的出厂默认 IP 地址为：Ethernet1：192.168.1.10;Ethernet2：192.168.2.10，都具备程序调试和下载功能，请确保网线连接到 Ethernet1/Ethernet2，将 PC 端 IP 地址应设为 192.168.1.xxx/192.168.2.xxx (xxx 表示除 10 外 1~254 范围内任一整数)。



图 1-15 TP6000 采用 LAN 网络电缆连接 PC 端通信配置






## 2 入门指引

### 2.1 创建工程

#### 2.1.1 启动编程环境

以 Invtmatic Studio V1.3.5 为例，编程环境启动步骤如下：

步骤1 双击桌面 Invtmatic Studio 编程软件  图标，启动后的 Invtmatic Studio 编程环境如下图所示。



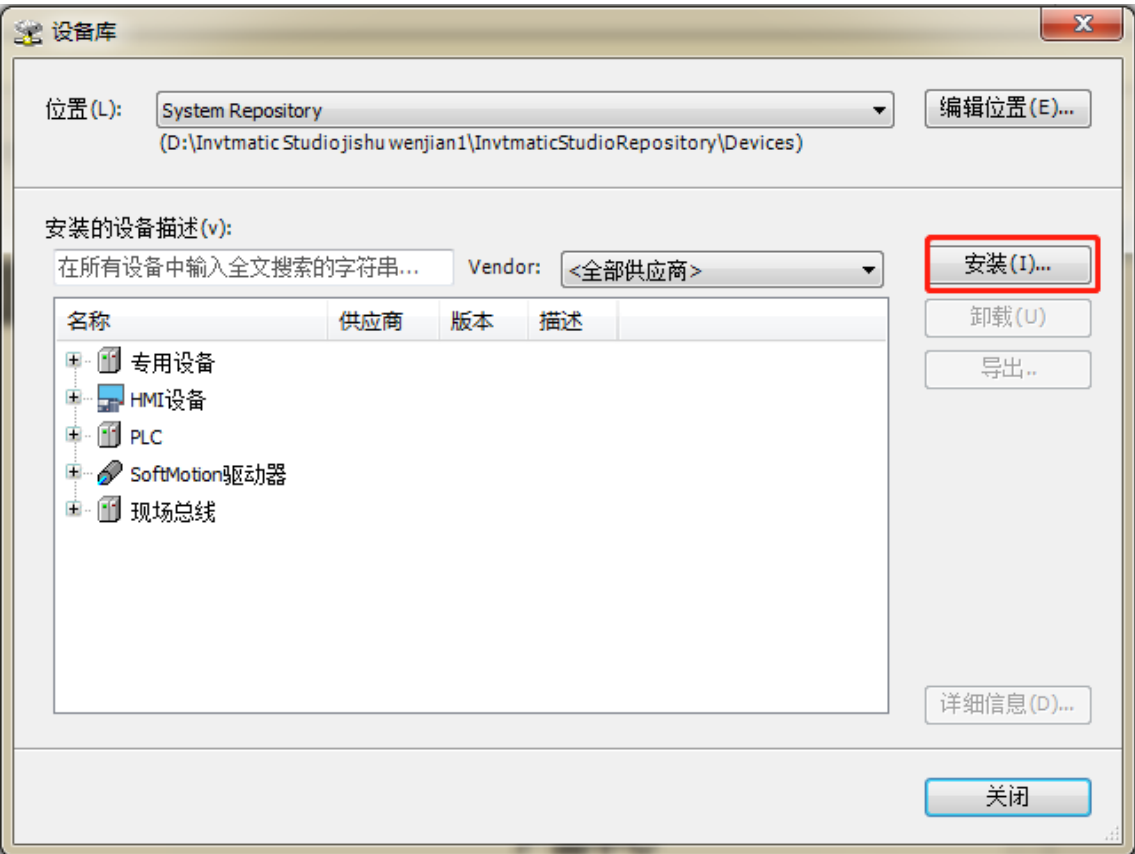
步骤2 添加设备描述文件，选择菜单栏中的 **工具>设备库**。

 **注意：**Invtmatic Studio V1.3.5 默认安装有 INVT 的设备描述文件，第三方设备需手动添加。

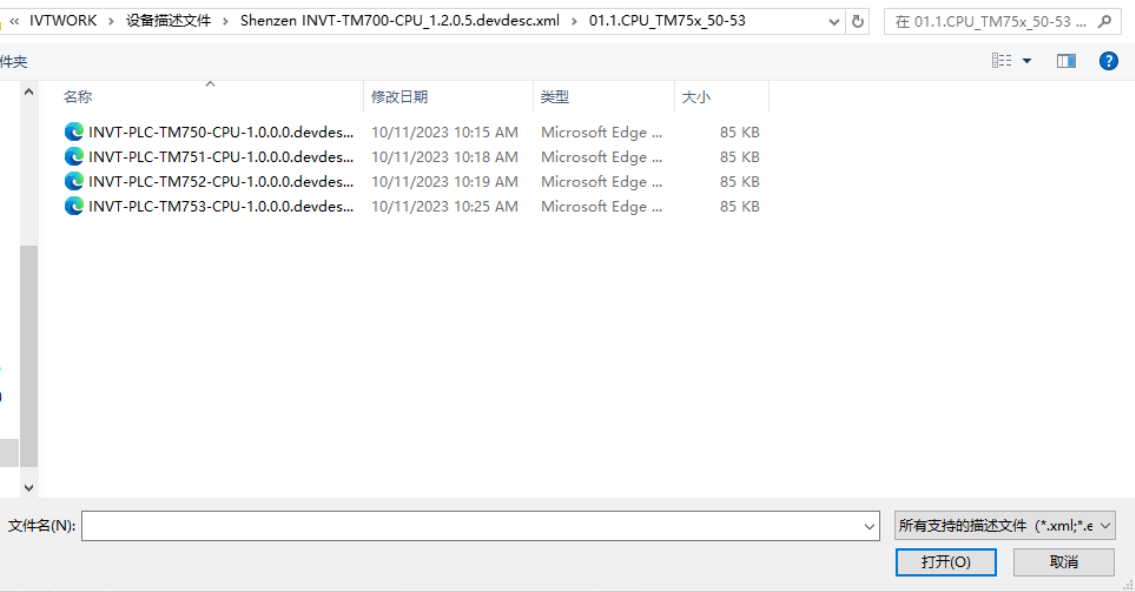




步骤3 在弹出的设备库窗口中点击**安装**。




步骤4 在弹出的窗口中找到本地文件夹中待安装的设备描述文件，将其选中并点击**打开**。

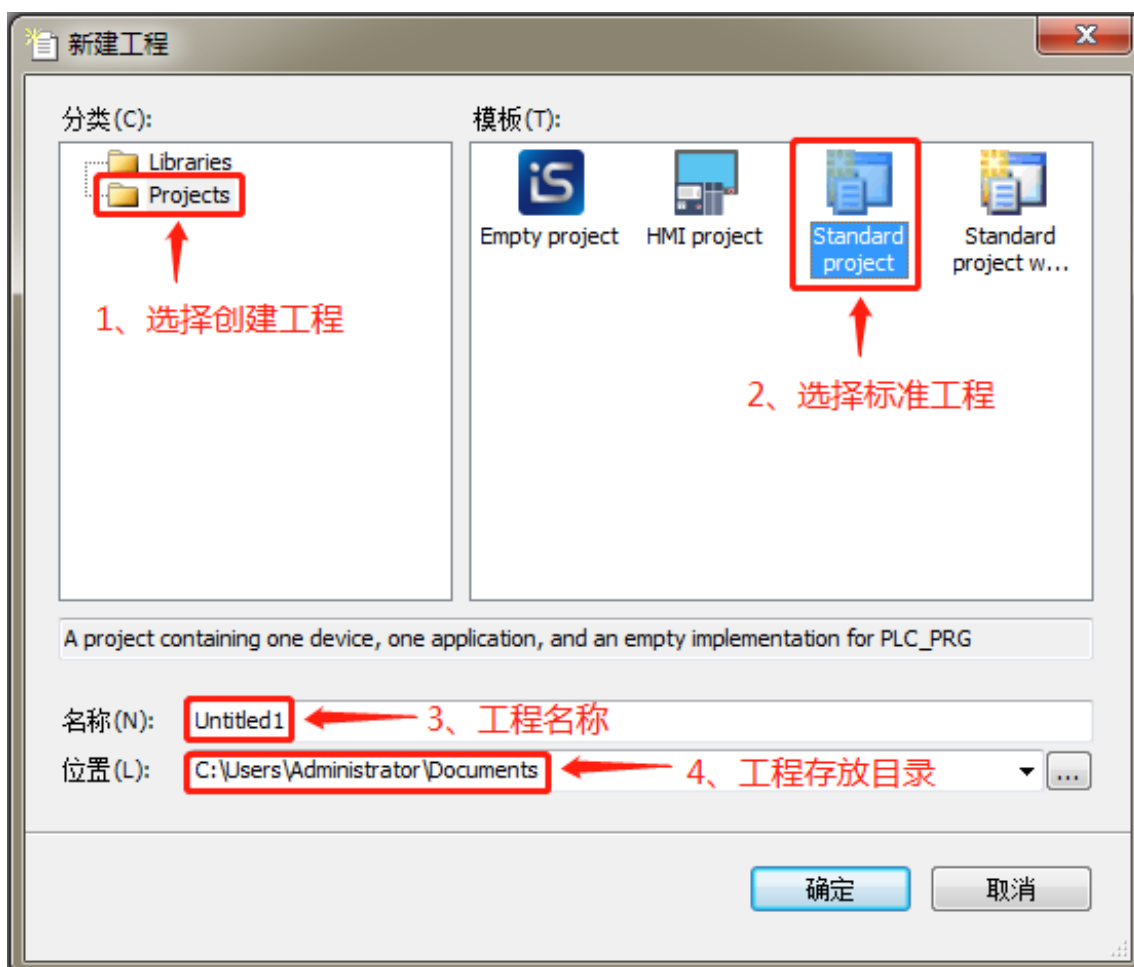


**注意：**此处可按上述步骤添加英威腾所提供的所有设备描述文件，或添加第三方的设备描述文件。

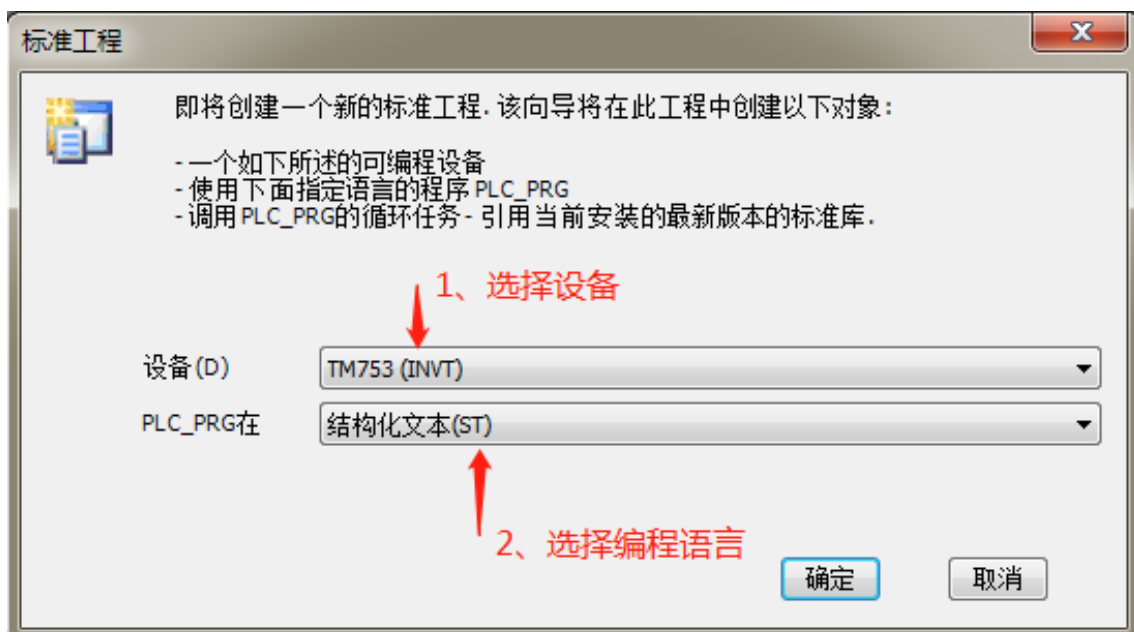


## 2.1.2 新建工程

点击左上角  新建工程图标或选择**文件>新建工程**，也可直接单击窗口中的**新建工程**快速建立工程。选择工程类型、工程模板、工程名称及工程存放目录，如下图所示。

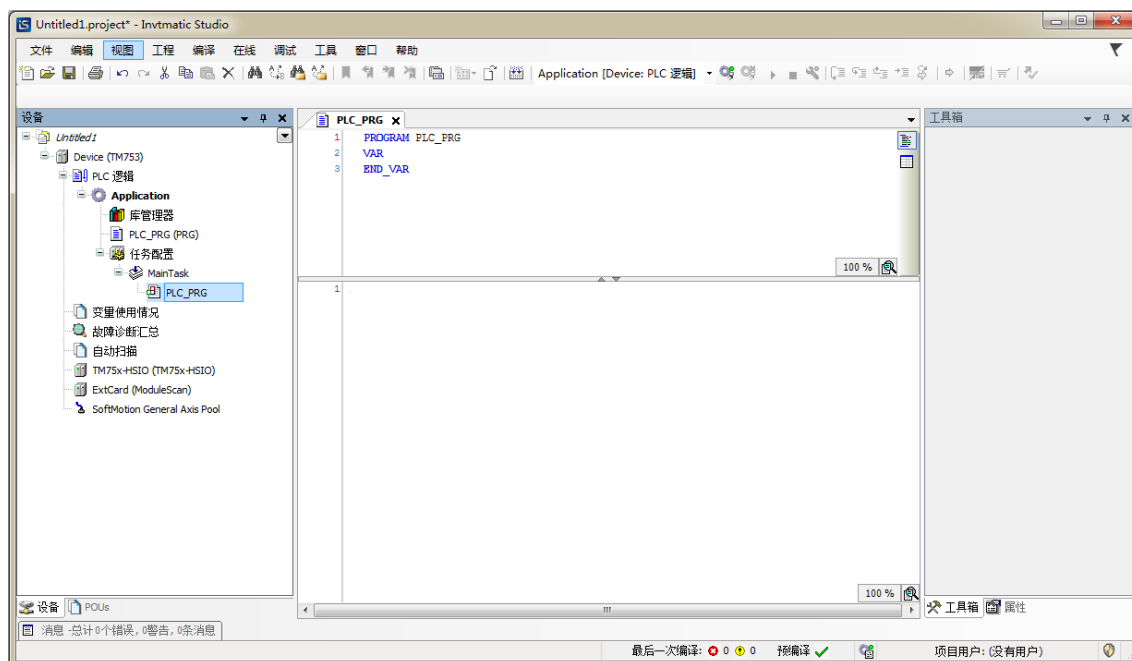


单击**确定**后，进入标准工程设置界面，在此界面用户可以选择设备类型和编程语言。





完成上述操作后，进入 Invtmatic Studio 组态配置与编程界面，双击设备中的 **PLC\_PRG(PRG)**，编写应用程序。



## 2.2 工程编写典型步骤

从 2.1.1 启动编程环境的示例来看，编写具有运动控制功能的用户程序，一般需要如下几个步骤：

### 步骤1 应用系统硬件配置

根据所使用的主控制器、扩展模块、网络类型、伺服从站等，进行网络配置。

### 步骤2 用户程序编写

根据所需实现的控制功能，用一个 POU 编写（如 POU1）运动控制，用另一个 POU 编写（如 POU2）普通逻辑控制。

### 步骤3 伺服驱动器参数配置

根据硬件配置中的伺服名称和运行模式，来配置 SDO、PDO 的对象，保证用户程序的 MC 功能块与伺服之间所需的通讯对象都填写在配置表中。

### 步骤4 伺服电机参数配置

必须准确填写伺服电机的编码器分辨率、机械结构的传动比、轴运动范围特点等，使得控制对象的位移指令与实际位移准确对应。

### 步骤5 任务安排

按控制的实时性要求，将运动控制功能 POU1 放在 EtherCAT 任务中执行，周期可设为 500μs、1ms、2ms 或 4ms 等，并与 EtherCAT 主站同步周期相同，优先级为 0；将普通逻辑控制 POU2 放在 MainTask 或其它普通任务下，周期可设为 1ms、4ms、20ms 等，优先级为 1。

### 步骤6 在线调试

将 TM 系列 PLC 控制器通过 LAN 网络线缆与 PC 连接，接线无误后上电。下载调试用户程序，排除用户程序 bug（若有条件，可将伺服驱动系统接入 TM 系列 PLC 控制器，再行调试。若手头没有伺服系统，可以将伺服驱动系统设为虚轴；若手头没有 TM 系列 PLC 控制器，还可以在 PC 上仿真调试用户程序，先排除用户程序中可能的错误，直到编译无错误）。



## 2.3 程序编写与调试实例

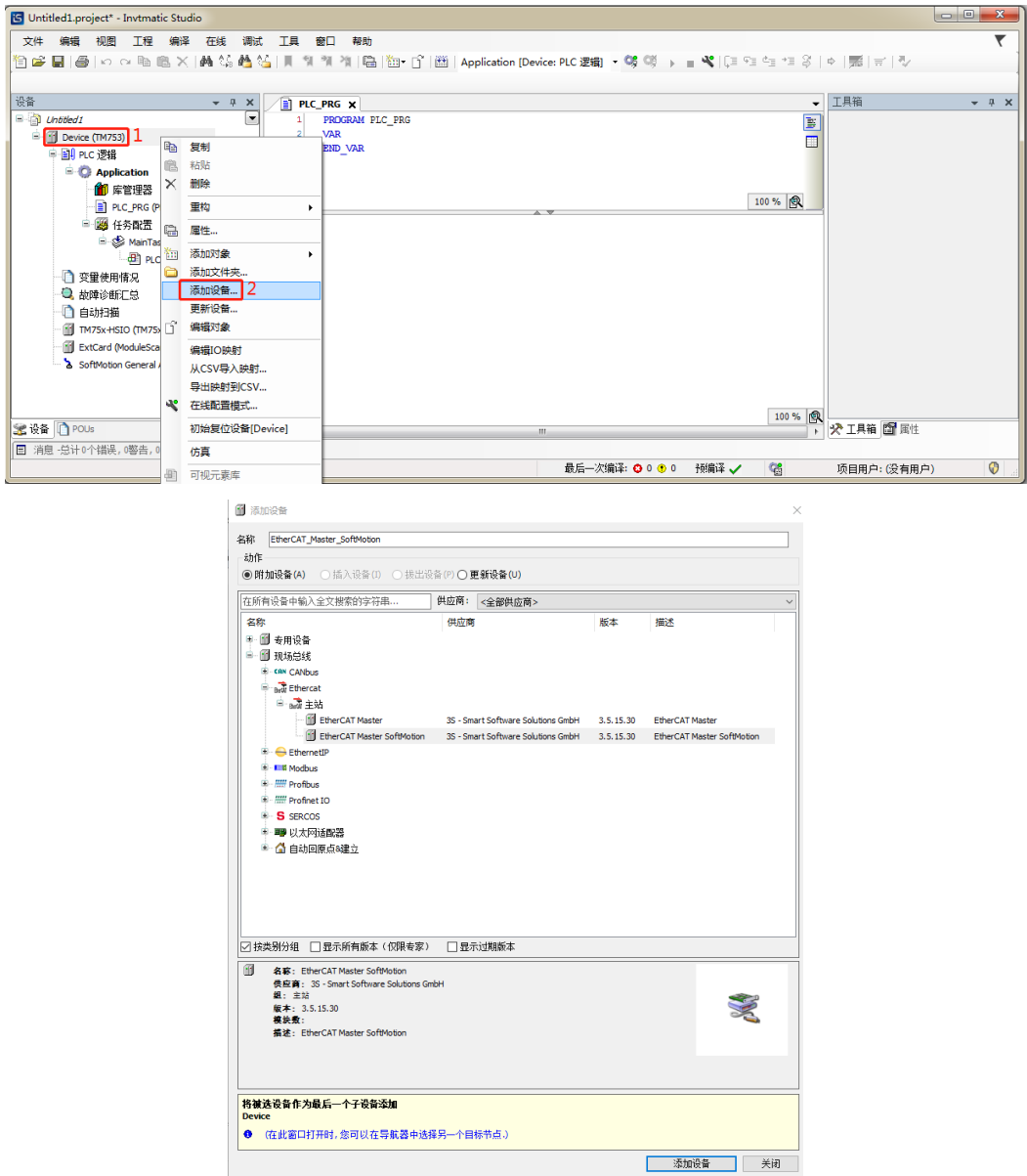
在说明编程系统原理与运控程序编写方法之前，先举例介绍一个基本的伺服控制程序，以便对编程过程有一个初步的了解。例如编写一个简单的程序，让 TM 系列 CPU 控制器实现让伺服电机正转 50 转，再反转 50 转，如此反复运行功能。

编程方法与步骤如下：

- 步骤1 添加相应的设备：EtherCAT 主站、伺服驱动器、电机轴。
- 步骤2 伺服的运动控制，需要放在高实时 EtherCAT 任务周期中处理。
- 步骤3 进行相关参数设置。
- 步骤4 编写程序。

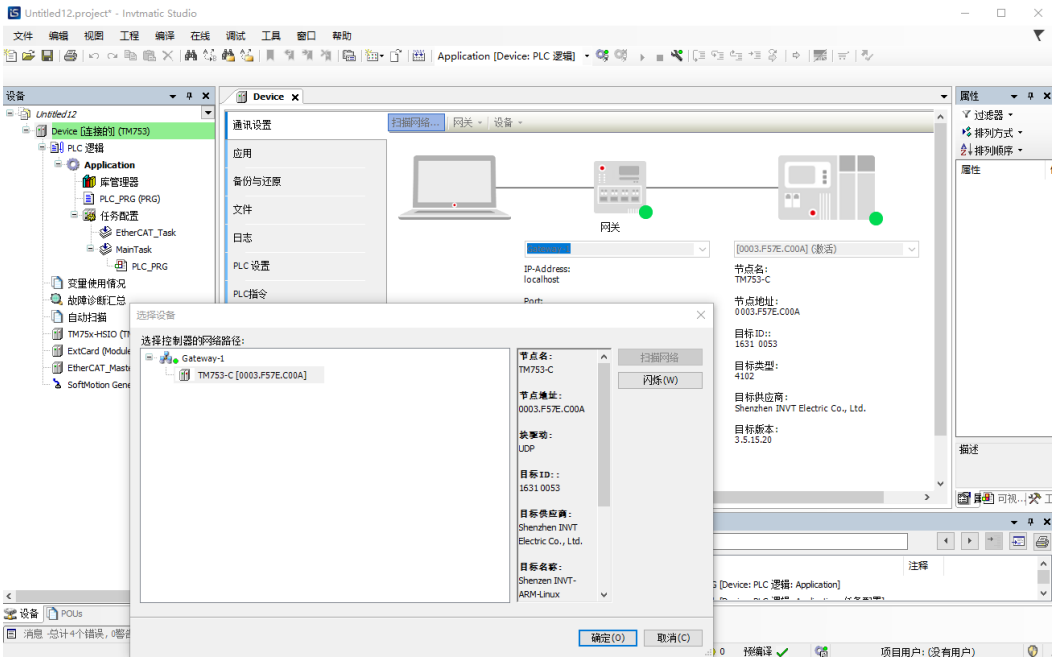
### 2.3.1 添加设备

- 步骤1 右键点击软件界面设备树中的 **Device**，选择**添加设备**后，再选择**现场总线>Ethercat>主站>EtherCAT Master SoftMotion**（对应的主站）后，点击右下角的**添加设备**。





步骤2 在软件界面左侧，双击 **Device** 后，在弹出的界面点击**扫描网络**，当右侧的 PLC 灯为绿色时，表示 PLC 设备连接成功，可下载 PLC 程序。



步骤3 添加从站。

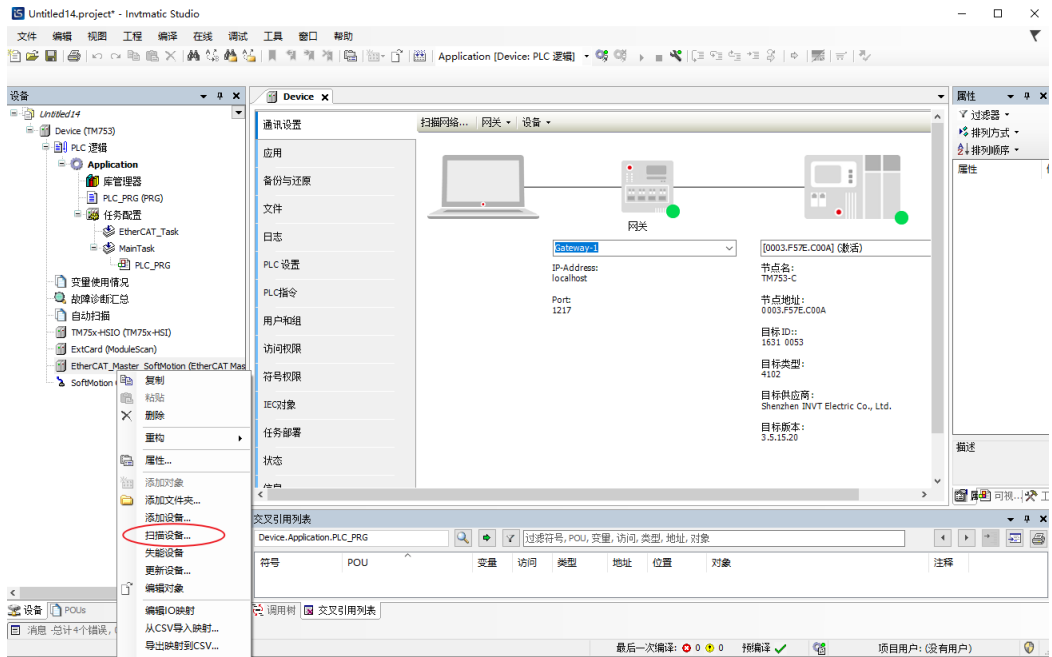
添加从站有两种方式：自动扫描设备组态和离线设备组态，下面分别介绍两种设备组态方法。

■ 自动扫描设备组态

通常在 PLC 和 EtherCAT 设备上电，且 EtherCAT 总线已将伺服电机、IO 模块等 EtherCAT 设备连接起来后，通过软件扫描总线设备，自动完成组态。操作步骤如下：

步骤1 在设备树中右键点击需要的主站设备，选择**扫描设备**，如图 2-1 所示，就能扫描出从站上的设备，并弹出扫描设备的结果界面，如图 2-2 所示。

图 2-1 EtherCAT 主站扫描界面

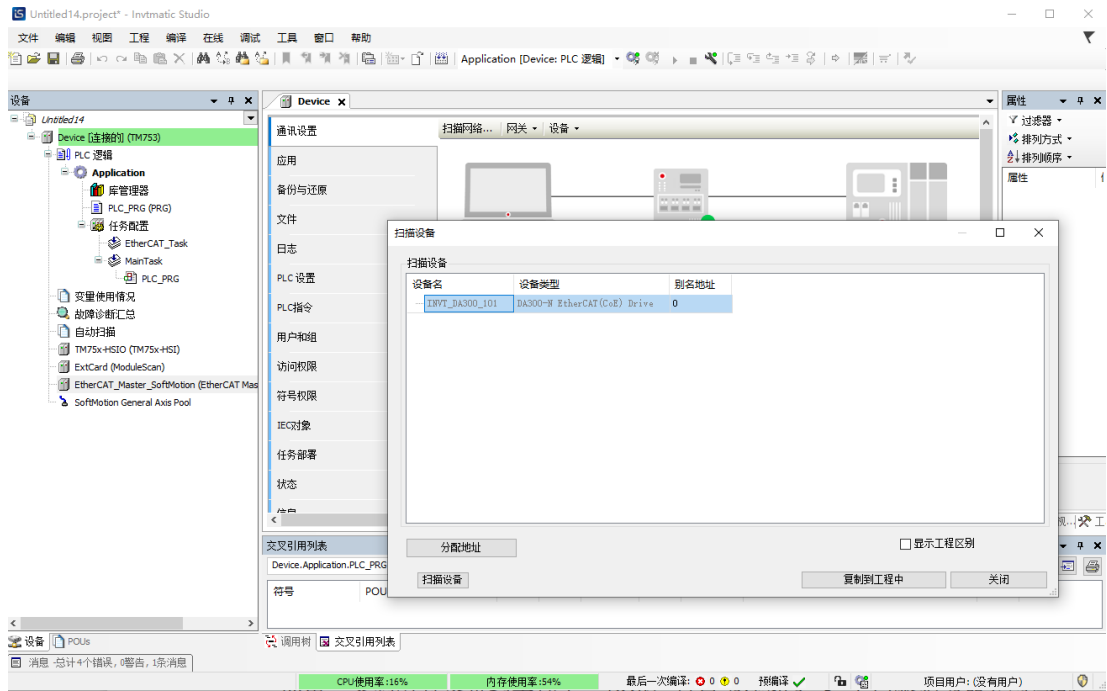




步骤2 点击扫描设备界面的**复制到工程中**，将自动添加伺服驱动器和对应的 402 轴。

**注意：**INVT 伺服电机不需要后期手动加轴，第三方伺服电机需要手动添加。

图 2-2 EtherCAT 主站添加从站界面



## ■ 离线设备组态

在没有 PLC、伺服电机等设备网络连接的环境时，可以直接在 Invtmatic Studio 软件上进行配置。

在设备树中右键点击 **EtherCAT\_Master\_SoftMotion** 主站，选择**添加设备**。

图 2-3 EtherCAT 离线添加从站

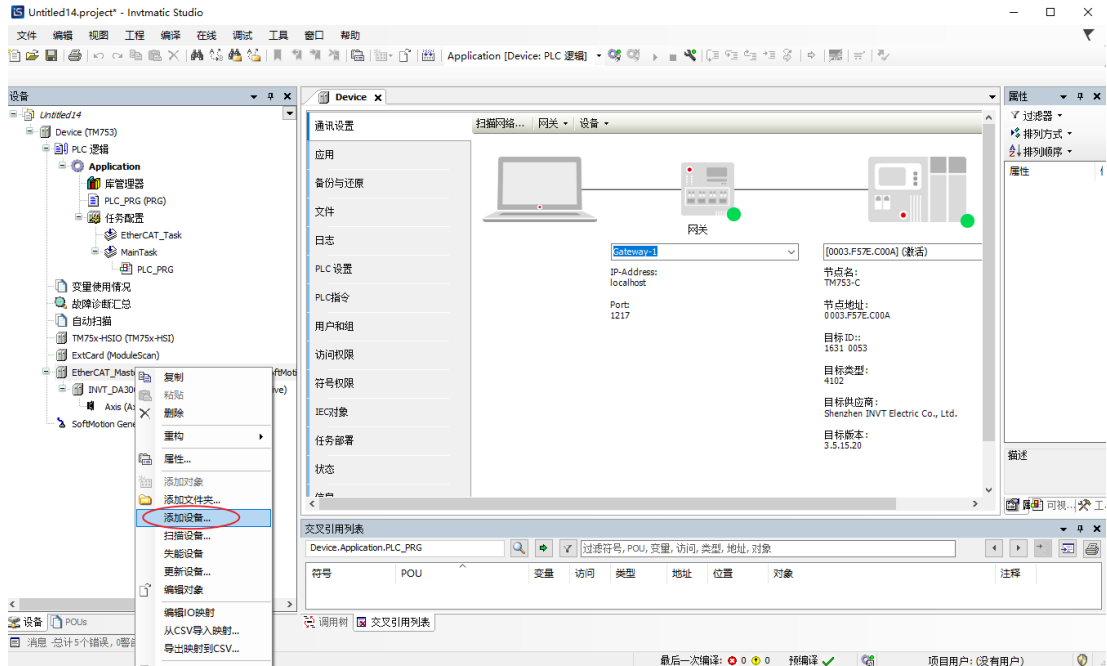
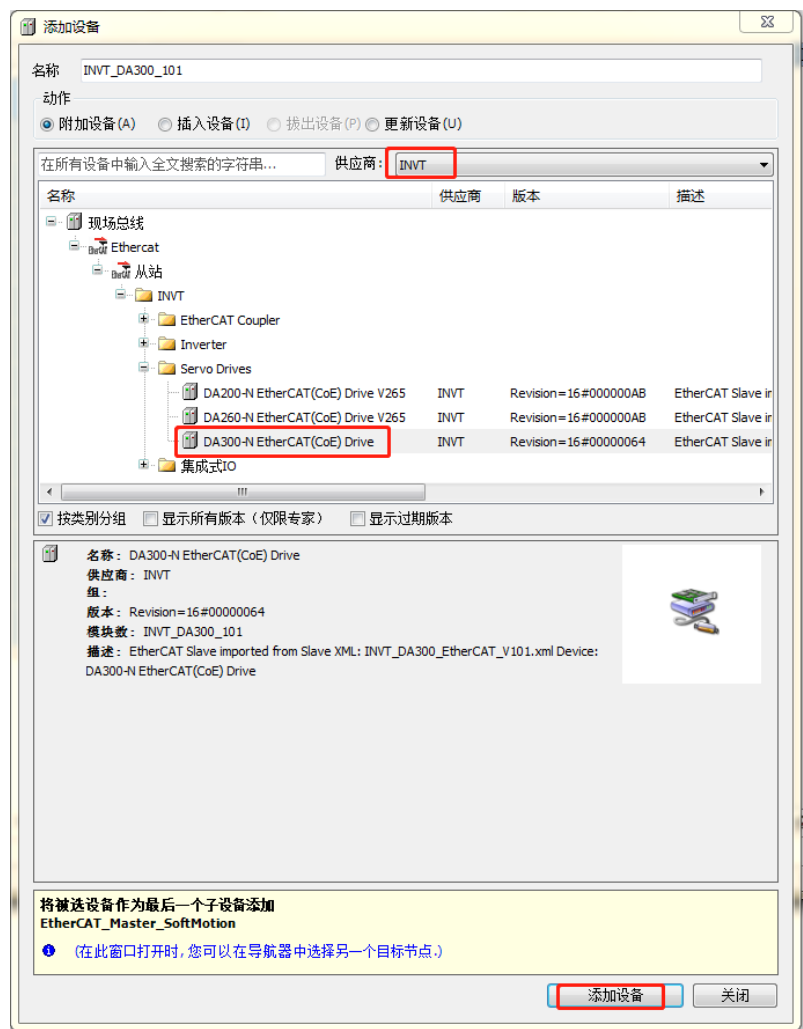
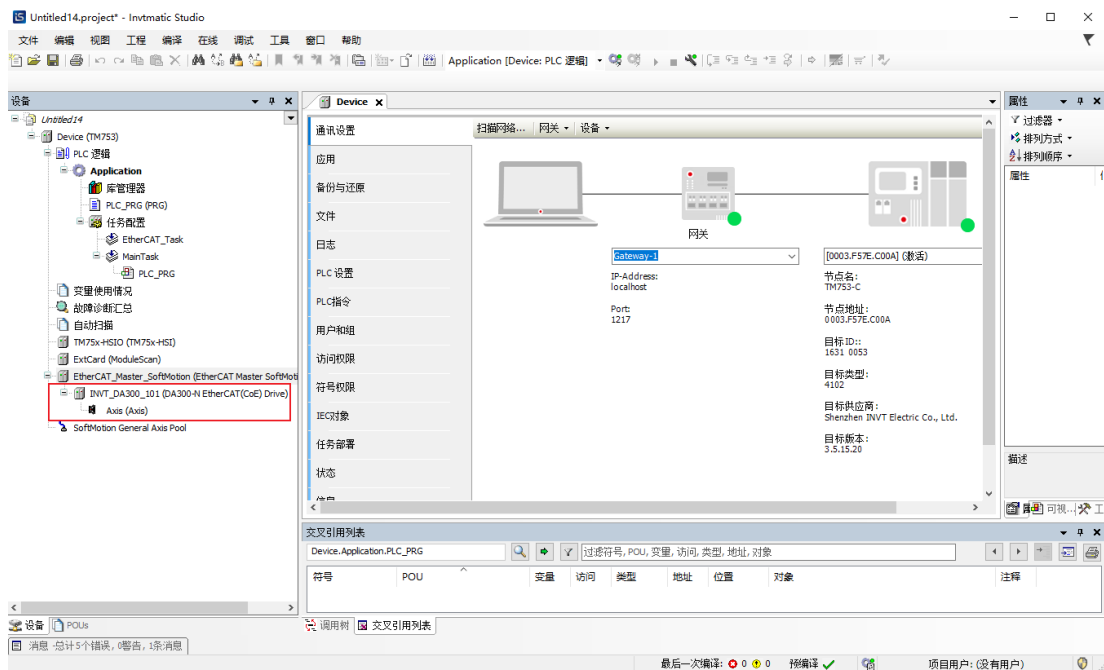




图 2-4 EtherCAT 选择从站设备



EtherCAT 添加伺服从站和 402 轴后如下图所示。

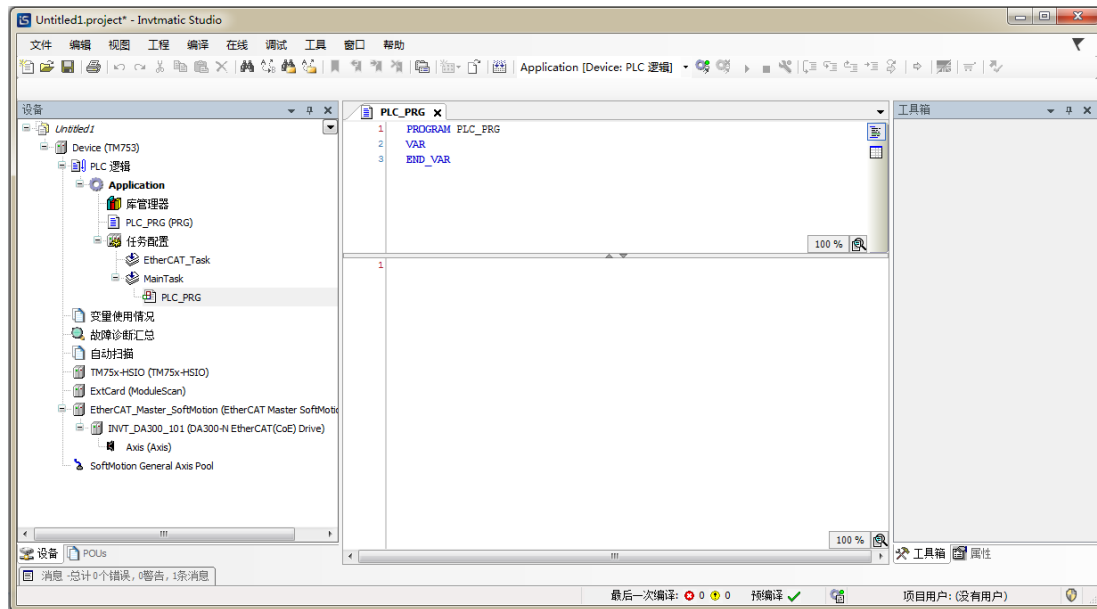




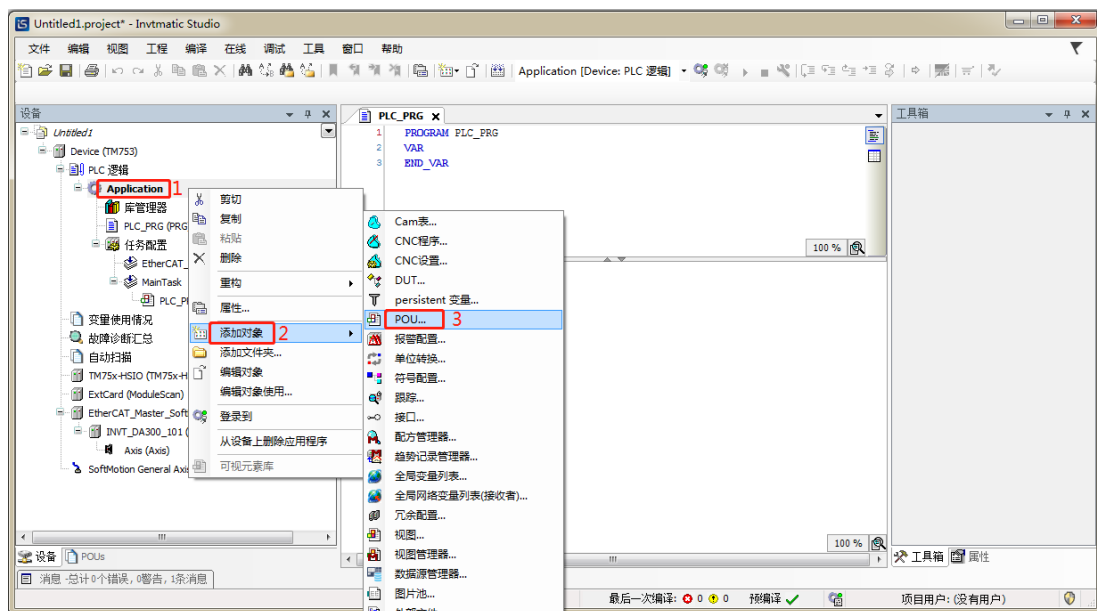
## 2.3.2 编写功能处理 POU

首先了解下 Invtmatic Studio 编程环境中的默认任务配置，默认有一个 EtherCAT\_Task 任务和 MainTask 任务，其中 MainTask 任务下有一个名称为 PLC\_PRG 的 POU，在新建工程时创建的，如图 2-5 所示。此时我们需要建立一个专用于伺服控制的 POU 放到 EtherCAT\_Task 任务下，建立步骤如下：

图 2-5 PLC\_PRG 编程界面



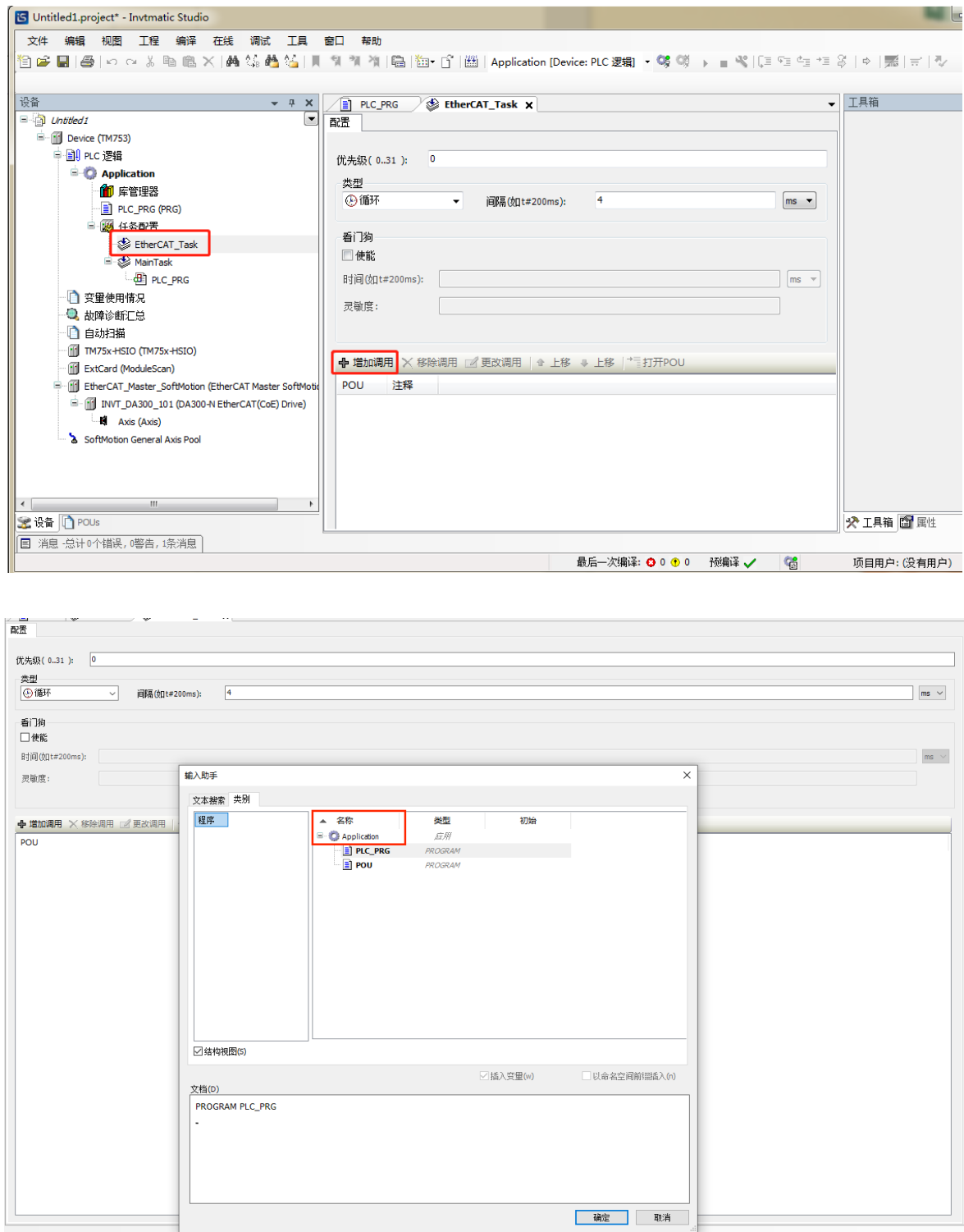
步骤1 在设备树中右键点击 **Application**，选择**添加对象>POU**，添加 EtherCAT 伺服控制专用 POU。





步骤2 双击设备树中的 **EtherCAT\_Task**，在配置界面点击**增加调用**，选择 **POU**。

图 2-6 EtherCAT 任务调用 POU

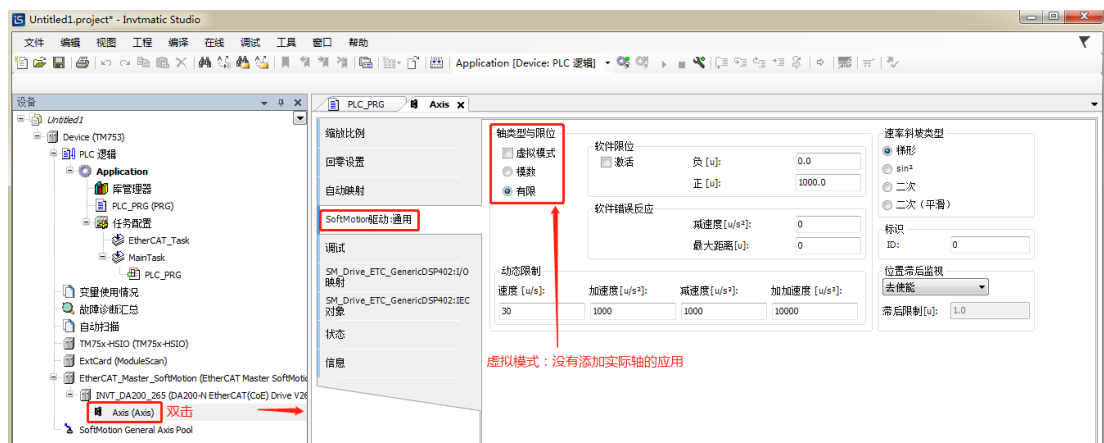


2.3.3 电机参数的设置

为了精确地控制运动位置，控制器必须准确计算伺服电机的位置，根据应用系统的运转特性、行程特点，如下图所示，选择**轴类型与限位**，以便控制器内部对读电机编码器反馈信息进行计算，得到准确位置，避免编码器脉冲数累积溢出造成错误。



图 2-7 电机参数设置

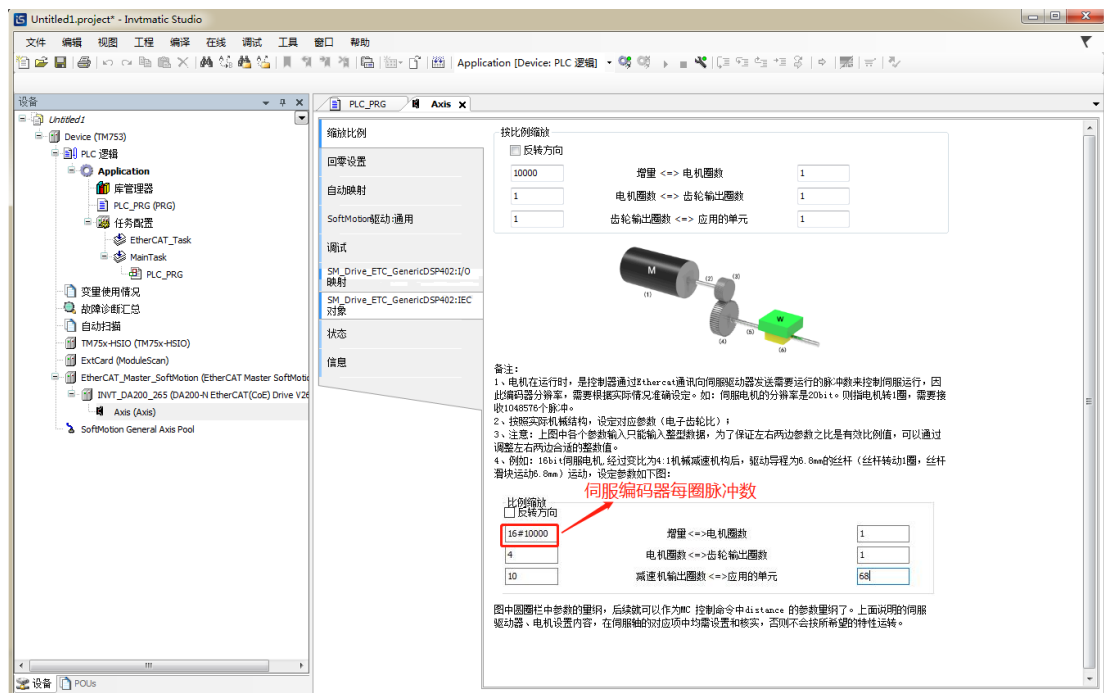


注意：

- 对于丝杆类型的往复运行机构，其行程是有限的，我们往往需要知道其在丝杆行程范围内的绝对位置，建议轴类型与限位选择有限。
- 若是单方向运转类型的转轴，采用线性模式容易出现位置计数溢出，导致位置计算错误，建议轴类型与限位选择模数。

电机的编码器参数（如分辨率），应用系统的机械减速比可能各不相同，在编程时也需要根据实际情况进行设定，如下图所示。

图 2-8 电机编码器参数设置



电机在运行时，是 PLC 通过 EtherCAT 通讯向伺服驱动器发送需要运行的脉冲数来控制伺服运行，因此编码器分辨率，需要根据实际情况准确设定。如：伺服电机的分辨率是 20bit，则指电机转 1 圈，需要接收 1048576 个脉冲。按照实际机械结构，设定对应参数（电子齿轮比）。

注意：上图中各个参数输入只能输入整型数据，为了保证左右两边参数之比是有效比例值，可以通过调整左右两边



合适的整数值。例如：20bit 伺服电机,经过变比为 3.75:1 机械减速机构后，驱动导程为 16mm 的丝杆（丝杆转动 1 圈，丝杆滑块运动 16mm）运动，设定参数如下图所示。

图 2-9 设定参数举例示意

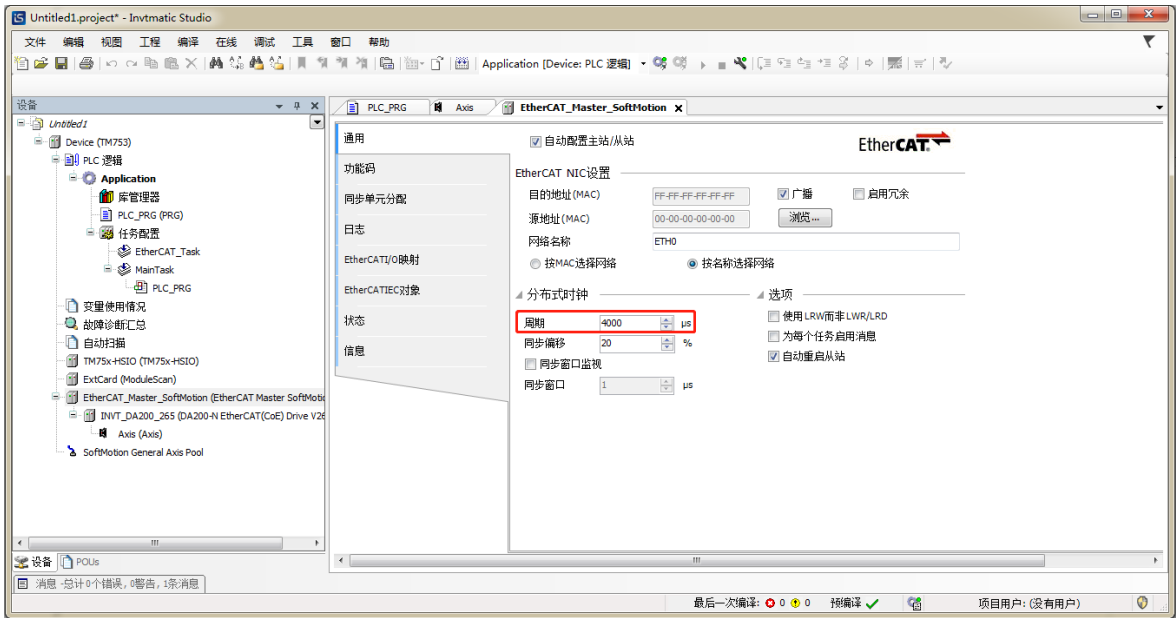
按比例缩放

<input type="checkbox"/> 反转方向		
1048576	增量 <=> 电机圈数	1
375	电机圈数 <=> 齿轮输出圈数	100
1	齿轮输出圈数 <=> 应用的单元	16

2.3.4 电机正反转编写

对于伺服轴的运动控制，默认同步周期为 4ms，用户可以根据实际需要进行选择，如下图所示。

图 2-10 伺服轴运动控制周期设置





如下图中的程序采用 ST 语言编写正反转逻辑，相关代码如下图所示。

图 2-11 ST 相关代码

```
2  VAR
3      MC_Power : MC_Power;
4      MC_MoveAbsolute: MC_MoveAbsolute;
5      iStatus: INT:=0;
6      i:UINT:=1000; //力矩限制
7  END_VAR

1 CASE iStatus OF
2  0:
3      MC_Power(Axis:= SM_Drive_GenericDSP402, Enable:= TRUE, bRegulatorOn:= TRUE, bDriveStart:=TRUE , );
4      IF MC_Power.Status
5      THEN
6          iStatus:=iStatus+1;
7      END_IF
8  1:
9      MC_MoveAbsolute(Axis:=SM_Drive_GenericDSP402 , Execute:= TRUE, Position:=200 , Velocity:=5 , Acceleration:= 5, Deceleration:= 5,);
10     IF MC_MoveAbsolute.Done
11     THEN
12         MC_MoveAbsolute(Axis:=SM_Drive_GenericDSP402 , Execute:= FALSE,);
13         iStatus:=iStatus+1;
14     END_IF
15  2:
16     MC_MoveAbsolute(Axis:=SM_Drive_GenericDSP402 , Execute:= TRUE, Position:=0 , Velocity:=4, Acceleration:= 5, Deceleration:= 5,);
17     IF MC_MoveAbsolute.Done
18     THEN
19         MC_MoveAbsolute(Axis:=SM_Drive_GenericDSP402 , Execute:= FALSE,);
20         iStatus:=1;
21     END_IF
22 END_CASE
```

2.3.5 用户程序编译

若有编写错误，图 2-11 中会列出错误类型与原因，双击其中的错误描述，光标会跳转到对应的程序编辑窗口，便于修订；逐一处理后，再进行编译，直到所有编译问题排除。操作步骤如下：

- 步骤1 在设备树中双击 **Device**，选择**通讯设置>扫描网络**。
- 步骤2 选择对应的设备后，点击**闪烁**，此时连接设备的 RUN 和 ERR 灯会闪烁三次。
- 步骤3 确认设备后，将用户程序下载到 CPU 模块中即可。

图 2-12 程序编译界面

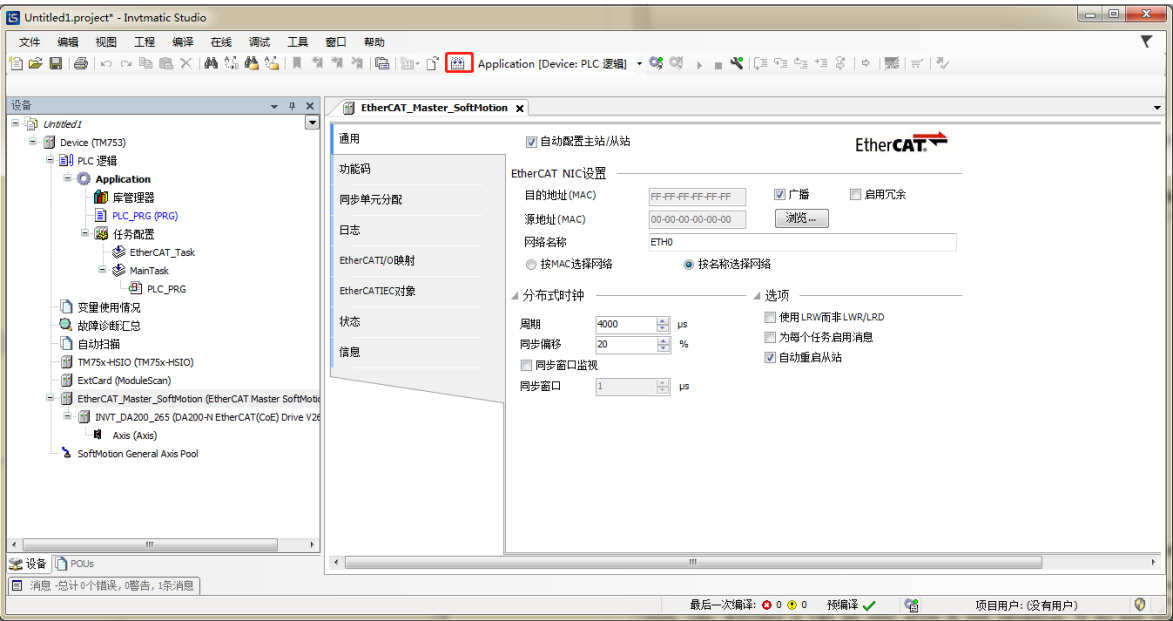




图 2-13 连接 PLC

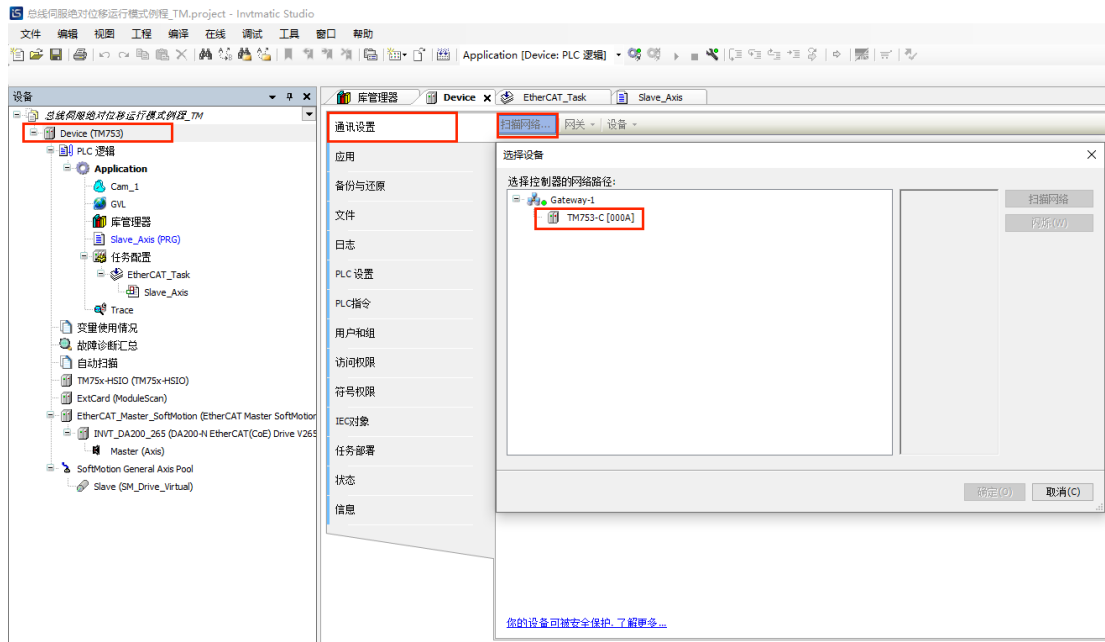
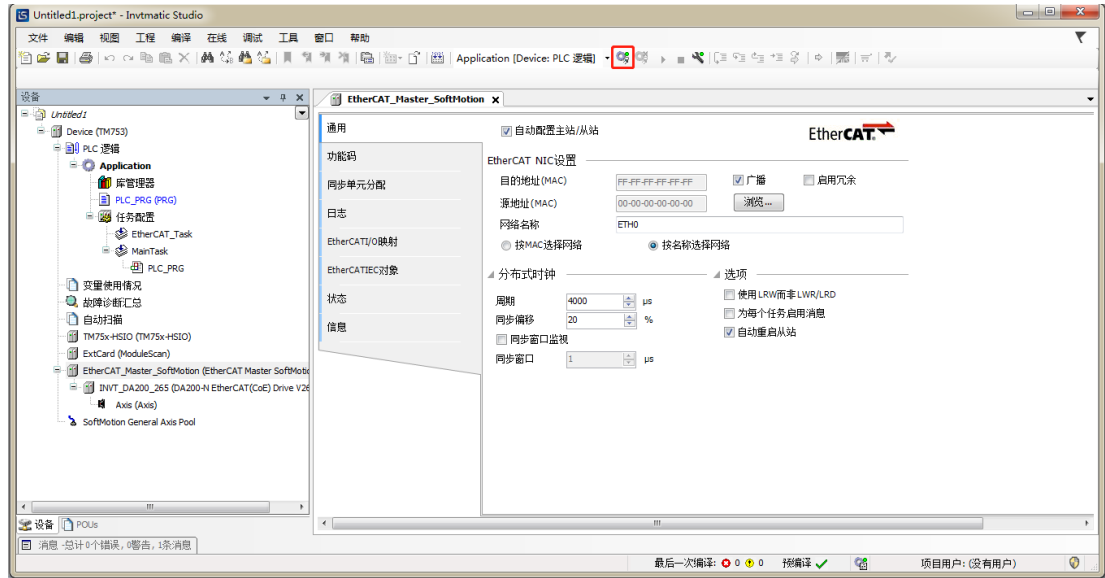


图 2-14 用户程序下载



2.3.6 监控程序运行

如图 2-14 登录到设备后，可以通过观测伺服的实际运行情况或查看上位机伺服轴的 position 值，就可以看到程序的运行，至此，编程所需的伺服点动、触发运行 2 圈的功能都已实现，一个简单的编程过程就完成了。并且可通过界面左下角查看 CPU 和内存占用率，如图 2-15 所示。在任务配置界面，可以查看各任务的执行周期时间（程序第一次上电运行后，需要右键点击，选择复位之后，才能查看准确的时间），如图 2-16 所示。



图 2-15 设备监控

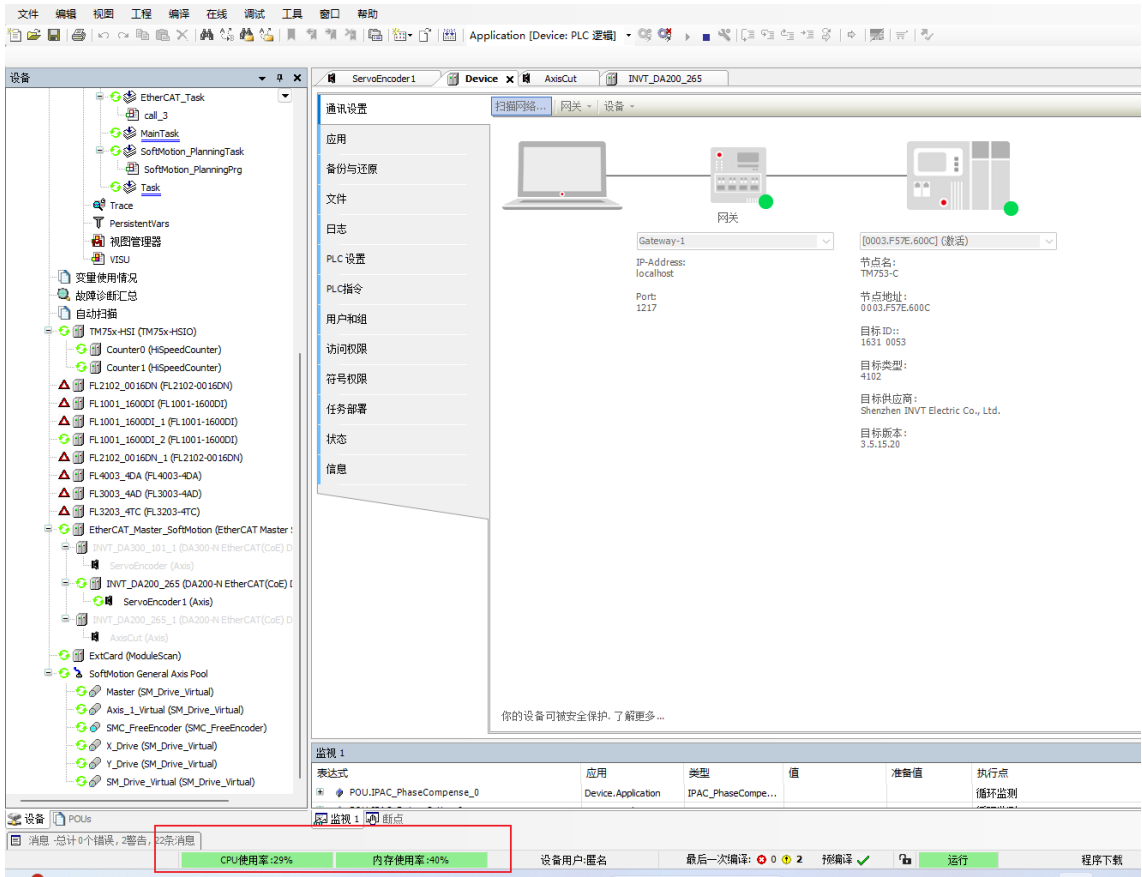
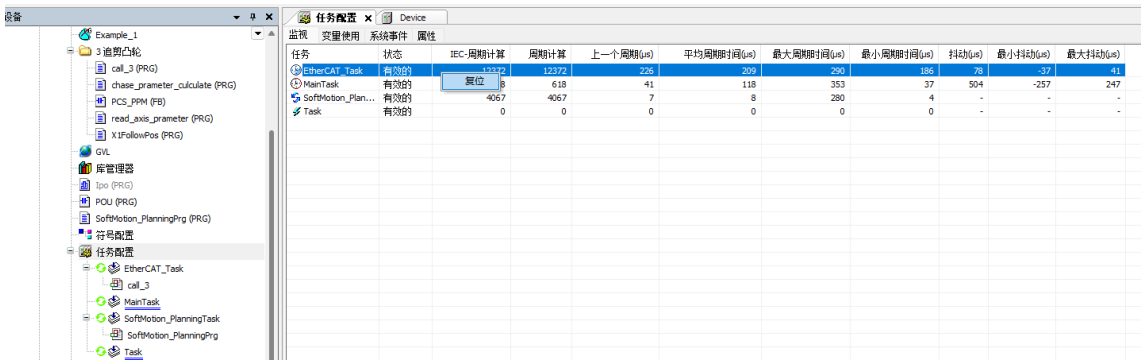


图 2-16 任务监视



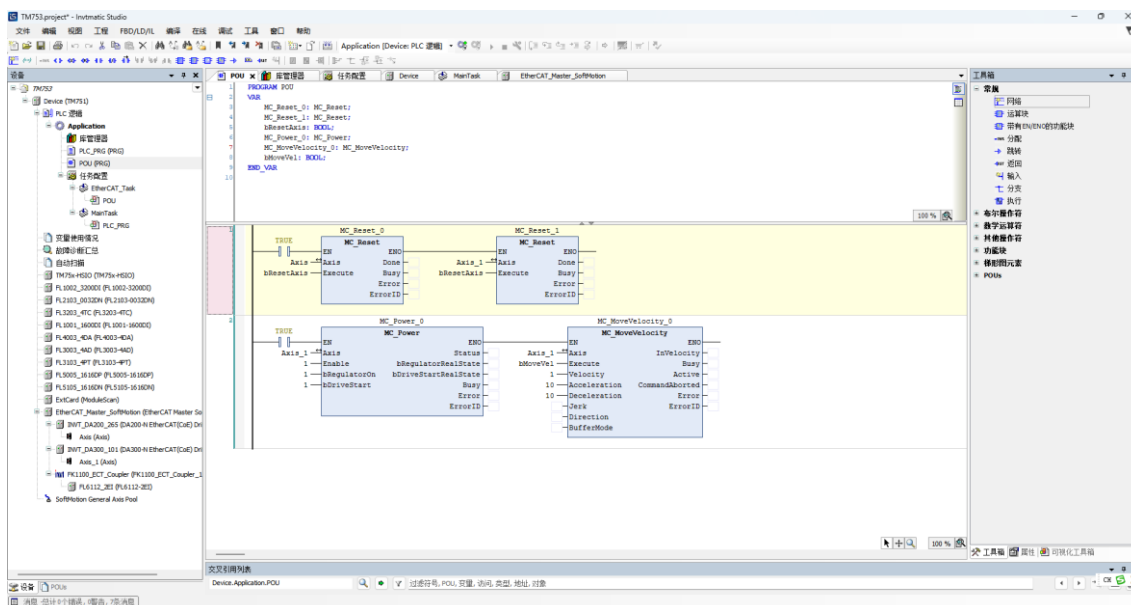


## 3 基本功能

### 3.1 界面布局

在 Windows 系统开始菜单中搜索或点击桌面的快速方式，打开 Invtmatic Studio 软件。

图 3-1 Invtmatic Studio 的 PLC 工程界面




界面主要包含以下内容：

- PLC 设备信息，点击鼠标右键 Device 可更换 PLC 型号。
- 用户程序管理单元 Application。
- EtherCAT 总线任务 EtherCAT\_Task。
- 主程序的任务 MainTask。
- 本地模块配置 TM75x-HSIO。
- EtherCAT 总线配置 EtherCAT\_Master\_SoftMotion。
- EtherCAT 总线从站 (INVT\_DA200\_265) 和 402 轴，或远程 IO 模块。
- 编译、登录及调试快捷键。

### 3.2 编译菜单

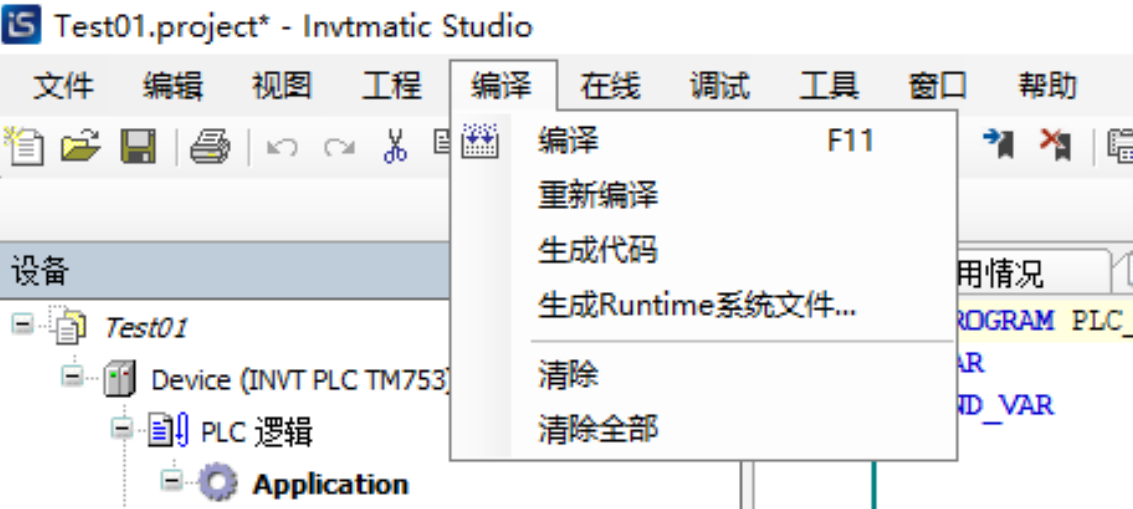
在软件编译菜单中集成了编译和清除等功能，主要说明如下：

- 编译：把所有代码编译链接为 PLC 可以执行的代码。
- 重新编译：清除上次的编译信息，重新执行编译命令。
- 生成代码：对于标签通讯中符号配置所需的生成文件功能，可以通过点击编译按钮来实现，或点击快捷键 。
- 生成 Runtime 文件：用于研发测试。



- 清除：清除上次的编译信息和下载信息。
- 清除全部：清除编译信息、下载信息、引用信息，所有的库、工程数据编译信息重新刷新。

图 3-2 编译菜单界面



### 3.3 变量使用情况表

#### 3.3.1 概述

变量使用情况表主要有以下功能：

- 显示 I/Q/M 区地址使用情况，包含已用地址、冲突地址、空闲地址的分类显示。
- 显示程序区、数据区、掉电保持区及 I/Q/M 区区域的使用大小、可用大小及使用率。

#### 3.3.2 功能介绍

在设备树中选择 **Device>变量使用情况**，双击**变量使用情况**，界面如下图所示。

**注意：**TM 系列程序区存储容量最大 20M，数据区最大 64M，掉电保持区最大 1M，M 区最大 5M，I 区最大 128K，Q 区最大 128K。

程序 (B)	总容量:	可用:	已用:	使用率:
程序 (B)	20971520	19013692	1957828	9.34%
数据 (B)	67108864	66843107	265757	0.4%
掉电保持区 (B)	1048576	1048458	118	0.01%
%M 区 (B)	5242880	5119980	122900	2.34%
%I 区 (B)	131072	130863	209	0.16%
%Q 区 (B)	131072	131006	66	0.05%

序号	地址	POU	变量名	变量类型	地址范围	状态
1	%MB0-%MB1	GVL	IRtu_1	WORD	%MB0 - %MB1	在用
2	%MB3	GVL	IRtu_2	WORD	%MB2 - %MB3	在用
3	%MB10	GVL	IMove	WORD	%MB10 - %MB11	冲突
4	%MB10	GVL	bRelative	BOOL	%MB10.1	冲突
5	%MB11	GVL	IMove	WORD	%MB10 - %MB11	在用
6	%MB200-%MB201	GVL	ITcp_1	WORD	%MB200 - %MB201	在用
7	%MB203	GVL	ITcp_2	WORD	%MB202 - %MB203	在用
8	%MB205	GVL	IMasterYout	INT	%MB204 - %MB205	在用
9	%MB207	GVL	IMasterPLC_In	INT	%MB206 - %MB207	在用
10	%MB210-%MB211	GVL	IDA_ErrorID	WORD	%MB210 - %MB211	在用



表 3-1 变量使用情况说明

序号	项目	说明
1	数据刷新区	点击界面上 <b>编译</b> 按钮，待成功后，界面会根据生成的数据，提取本界面需要的信息，刷新本界面。除此之外，点击图 3-2 中菜单 <b>编译&gt;生成代码</b> ，也会触发界面刷新。 <b>注意：</b> 此功能只有在编译成功后，才会刷新界面。
2	区域使用信息区	主要显示程序区、数据区、掉电保持区、I/Q/M 区的基本信息，包括区域总容量、已用区大小、可用区大小及区域使用率等信息。
3	I/M/Q 详细使用信息及功能块信息显示区	I/M/Q 区地址的使用情况，包括地址关联的变量信息、地址冲突状况、未用地址区信息等； 程序块大小信息主要包括程序中调用的功能块类型及大小信息，另外支持查找、地址范围设置、页码定位等功能。

其中信息显示区的界面如下图所示，该界面分为 4 个显示选项，分别是%M 区、%I 区、%Q 区和功能块大小，分别显示对应区域的使用信息。

变量使用情况

提示：当前工程已经是最新，不需要重新编译！

编译

程序 (B)	总容量:	20971520	可用:	19013692	已用:	1957828	使用率:	9.34%
数据 (B)	总容量:	67108864	可用:	66843107	已用:	265757	使用率:	0.4%
掉电保持区 (B)	总容量:	1048576	可用:	1048458	已用:	118	使用率:	0.01%
%M 区 (B)	总容量:	5242880	可用:	5119980	已用:	122900	使用率:	2.34%
%I 区 (B)	总容量:	131072	可用:	130863	已用:	209	使用率:	0.16%
%Q 区 (B)	总容量:	131072	可用:	131006	已用:	66	使用率:	0.05%

%M 区 %I 区 %Q 区 功能块

状态: 全部 地址范围: 0 - 5242879 变量名: 查询

序号	地址	POU	变量名	变量类型	地址范围	状态
1	%MB0-%MB1	GVL	IRtu_1	WORD	%MB0 - %MB1	在用
2	%MB3	GVL	IRtu_2	WORD	%MB2 - %MB3	在用
3	%MB10	GVL	IMove	WORD	%MB10 - %MB11	冲突
4	%MB10	GVL	bRelative	BOOL	%MX10.1	冲突
5	%MB11	GVL	IMove	WORD	%MB10 - %MB11	在用
6	%MB200-%MB201	GVL	ITcp_1	WORD	%MB200 - %MB201	在用
7	%MB203	GVL	ITcp_2	WORD	%MB202 - %MB203	在用
8	%MB205	GVL	IMasterYout	INT	%MB204 - %MB205	在用
9	%MB207	GVL	IMasterPLC_Xin	INT	%MB206 - %MB207	在用
10	%MB210-%MB211	GVL	IDA_ErrorID	WORD	%MB210 - %MB211	在用

第一页 前一页 1 / 2 下一页 尾页

每页显示: 10

3.3.3 菜单选项

菜单选项中包含：显示选项、查询、地址范围设置和翻页选项。

表 3-2 变量使用统计菜单说明

序号	项目	说明
1	显示选项	在%M 区、%I 区、%Q 区中，显示状态内容为全部、在用、系统使用、冲突、空闲；在功能块大小显示界面，显示状态内容为全部、功能块、联合体、结构体和别名。
2	查找选项	可对地址、POU、变量名、变量类型、地址范围四列选项中的名称进行字符匹配，任何一项匹配，即算满足条件。 <b>注意：</b> 查找选项中内容改变便会触发表格刷新。
3	地址范围设置	用户可以根据实际需要，查看自己设定的地址范围，范围不可以超出对应区域的极限范围。范围显示需要点击查询选项，才可生效。 <b>注意：</b> 功能块大小界面中无该项。
4	翻页选项	为保证表格刷新性能，表格默认最多添加 1000 个地址段单元，当数据超过 1000 个



序号	项目	说明
		地址段时，系统将采用分页显示，可通过点击前一页/下一页或输入指定页码项，系统便会跳转到对应页面。

表格选项：主要根据筛选条件显示符合条件的地址段使用信息。

表 3-3 变量使用统计表格说明

序号	项目	说明
1	地址	按照从小到大的顺序，显示地址区，最小单位为 Byte。当一个变量所占地址不冲突时，将以地址区域（如：%MB0~%MB3）的形式显示。一个地址下可能关联了一个变量，也可能关联多个变量，存在多个变量时，地址栏将显示多个重复地址。
2	POU	显示变量所有的 POU 名称。
3	变量名	显示该区域关联的变量名称。
4	变量类型	显示变量类型。
5	地址范围	显示变量的整个地址区间。
6	状态	<p>显示地址区的使用情况，已用时状态为在用，冲突时状态为冲突，空闲下为空。</p> <p><b>注意：</b></p> <ul style="list-style-type: none"> <li>地址冲突检测是按照 Byte 位进行检测的，当不同变量使用了同一地址不同 bit 位，系统检测时，将其标记为地址冲突，具体冲突情况以实际 bit 位使用情况为准。</li> <li>当用户定义变量与设备分配 IO 地址冲突时，只表示同一个地址存在多处使用，用户根据实际功能是否使用来判断是否存在问题。</li> </ul>

### 3.3.4 PLC 直接地址存储区域

不同 PLC 提供的直接存储区域不同。对于 PLC 数据，%I、%Q 区地址不能掉电保存，%M 区可以掉电保存。TM700、编程系统提供 128kB (Byte) 的输入区域（%I 区），128kB (Byte) 输出区域（%Q 区）和 512kB (Byte) 存储区域（%M 区）。其中存储区域中的前 480KB 用户可以直接使用，后 32KB 为系统使用区域（主要用作软元件），用户不要直接使用。编程时，用户可以直接访问地址，也可以定义变量后将变量映射到地址间接访问。存储区域定义及使用的地址范围如下表。

表 3-4 储存区域使用地址范围说明

区域	用途	大小	地址范围
I 区 (%I) 128kB	用户使用区域	64K Words	%IW0~%IW65535
Q 区 (%Q) 128kB	用户使用区域	64K Words	%QW0~%QW65535
M 区 (%M) 512kB	用户使用区域	240K Words	%MW0~%MW245759
	用 SD 元件	10000Words	%MW245760~%MW255759
	用 SM 元件	10000Bytes Words	%MB511520~%MB521519
	保留	2768Bytes	%MB521520~%MB524287

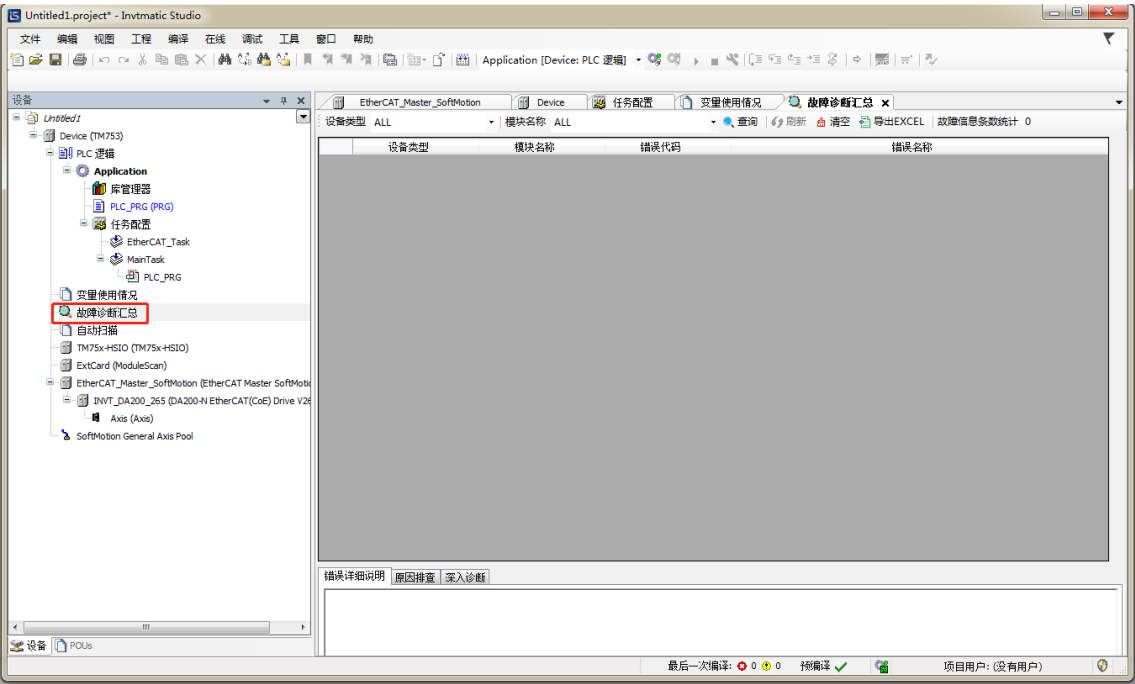
## 3.4 故障诊断

诊断是为了快速定位 PLC 运行过程出现的错误，通过错误信息和状态找出应对的解决方案，Invtmatic Studio 上的诊断界面只有登录 PLC 后才能获取和显示。Invtmatic Studio 编程系统支持各种通信设备的诊断，可以根据各通信设备实际运行状态生成故障、离线等信息。故障诊断涉及的模块类型主要包括 CPU 模块、Modbus、ModbusTCP 等。Invtmatic Studio 编程系统主要提供了四种获取诊断方式：组态诊断、诊断信息列表、设备自身诊断信息列表和诊断编程接口。所有诊断都是通过诊断码解析获取，且诊断码和诊断编程接口相对应。

故障诊断可以用于显示所有设备出现的故障信息，并提供相关故障信息的详细说明、原因排查方法，同时针对特殊情



况还可以提供更详细的诊断信息。设备创建连接后，双击设备树中的**设备诊断汇总**打开设备故障诊断界面。



设备类型	模块名称	错误代码	错误名称
Modules	FL3203_4TC	Er0033-0025	通道1信号源开路故障
Modules	FL3003_4AD	Er0032-0015	通道0信号源开路故障
Modules	FL4003_4DA	Er0032-2003	模块输出端口供电故障
Modules	FL4003_4DA	Er0032-2003	模块输出端口供电故障
Modules	FL4003_4DA	Er0032-2003	模块输出端口供电故障
Modules	FL4003_4DA	Er0032-2003	模块输出端口供电故障
Modules	FL2002_0016DP	Er0031-2003	模块输出端口供电故障
ModbusTCP	Modbus_TCP_Slave2	Er00A1-000f	通讯超时

故障诊断界面功能说明如下：

● 设备类型窗口

用于显示当前出现故障类型，同时提供故障显示过滤功能，可以针对特定设备类型进行故障信息显示。设备类型包括CPU 模块、Modbus 模块、Modbus TCP 模块、本地模块。选择不同的设备类型，诊断显示列表显示对应类型的诊断，默认为所有，显示所有设备诊断。

● 界面选项说明

- ✧ 设备类型：筛选某一类总线故障设备。
- ✧ 模块名称：筛选某一特定名称故障设备。
- ✧ 搜索：根据设备类型或模块名称选项信息，搜索匹配的故障信息。

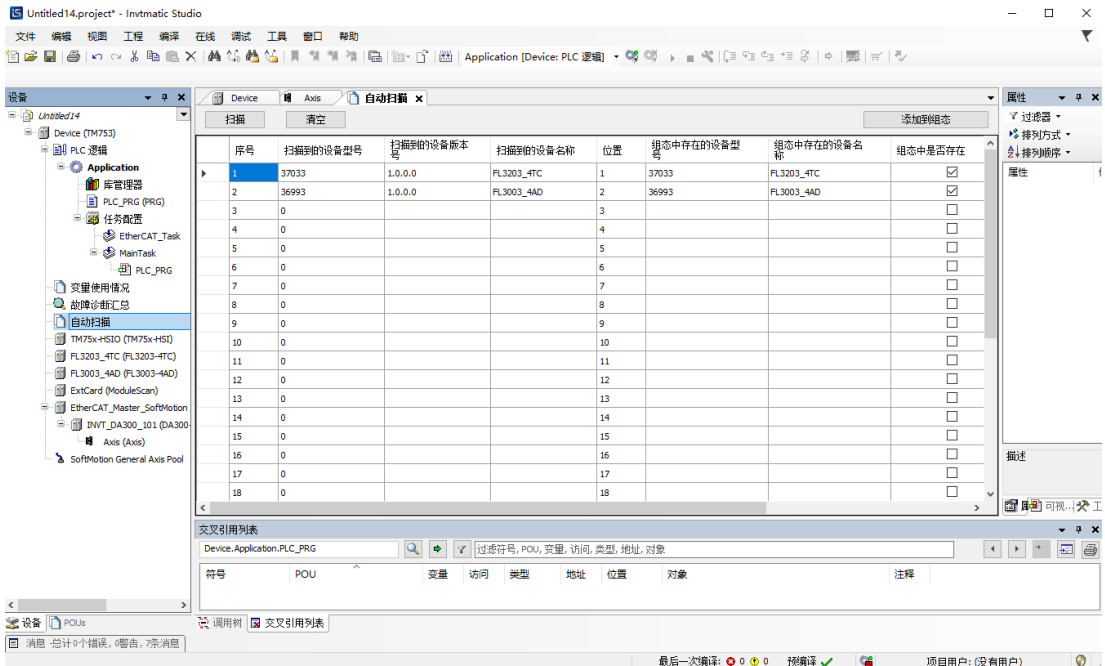


- ◇ 刷新：用于刷新设备故障信息。
- ◇ 清空：清空表格中的故障信息。
- ◇ 导出 excel：导出表格中故障信息。
- ◇ 故障信息条数统计：显示故障数量。
- ◇ 故障信息列表：主要用于显示具体模块故障信息，包括设备类型，模块名称，错误代码和错误名称。
- ◇ 信息详细说明窗口：当在故障信息列表中选中了某一条故障信息，在信息详细说明窗口会显示该故障的详细信息，该窗口包括错误详细说明、原因排查和深入诊断三个选项。错误详细说明窗口：显示故障可能原因说明。原因排查窗口：用于提供原因排查的具体操作方法方式。深入诊断窗口：针对某些复杂错误，需要获取更多细节信息定位的。

3.5 自动扫描

当用户需要增加 IO 口数量，可扩展 IO 模块，在软件中添加 PLC 本体扩展模块步骤如下：

- 步骤1 打开 Invtmatic Studio 编程软件，新建工程，选择编程语言。
- 步骤2 连接 PLC，扫描本地扩展模块需要登录 PLC（可不运行）。
- 步骤3 双击设备树中的**自动扫描**，在自动扫描界面，点击**扫描**，列表中会自动扫描出已安装的扩展模块。单击右上角**添加到组态**，即可完成扩展模块的扫描和组态添加。

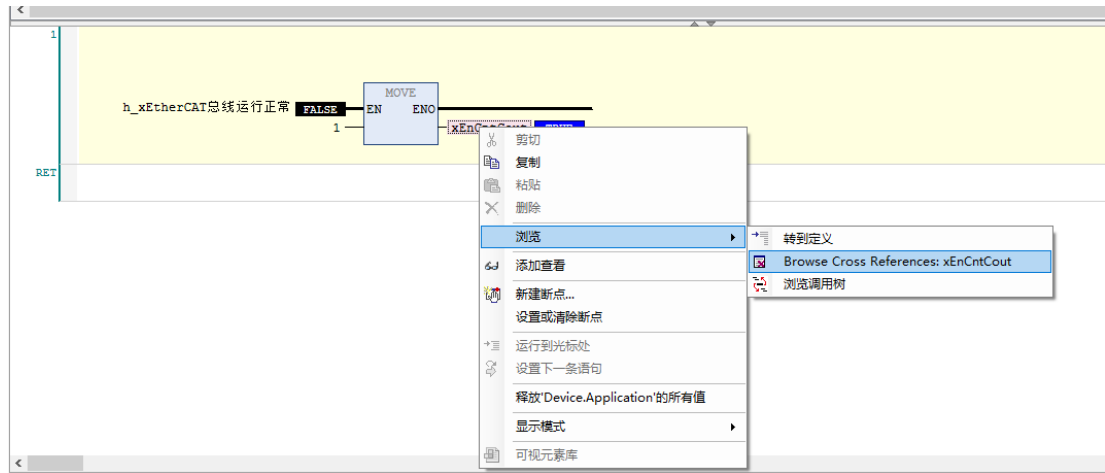


3.6 交叉引用

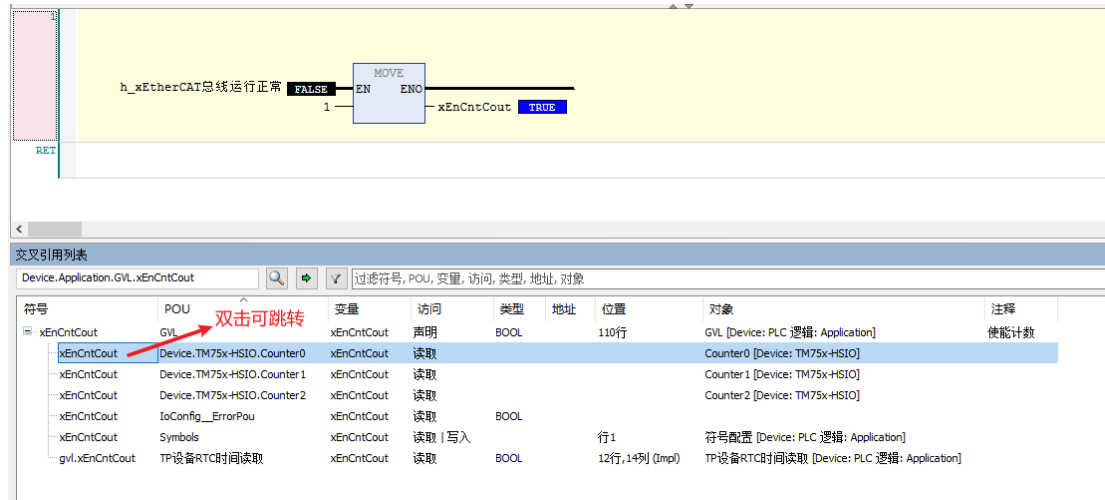
通过交叉引用功能可以快速查找目标对象在整个工程中的调用位置，操作步骤如下：

- 步骤1 找到需要交叉引用的对象，单击右键选择**浏览>Browse cross References:xEnCntCout**。





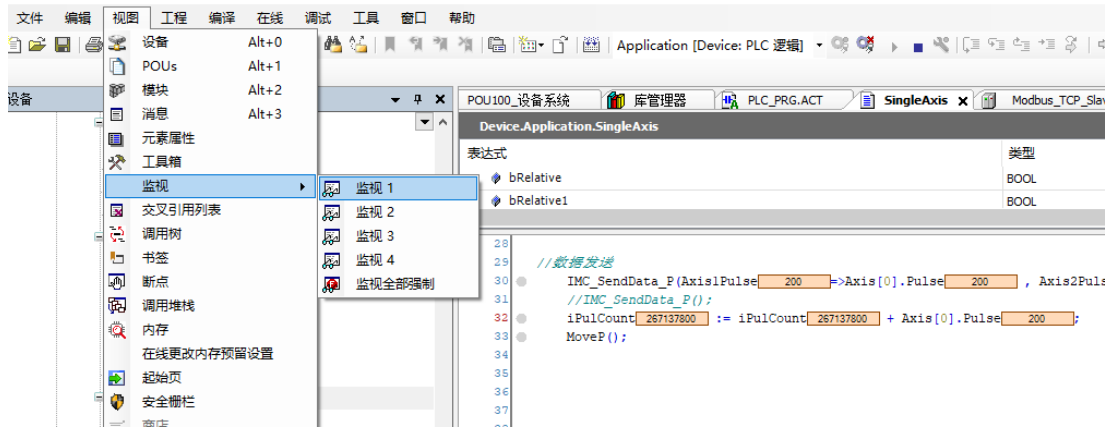
步骤2 在工程下方的交叉引用列表查看“目标对象”在整个工程中的使用情况，双击交叉引用列表中的信息可以跳转到工程中的具体使用位置。



### 3.7 监控表

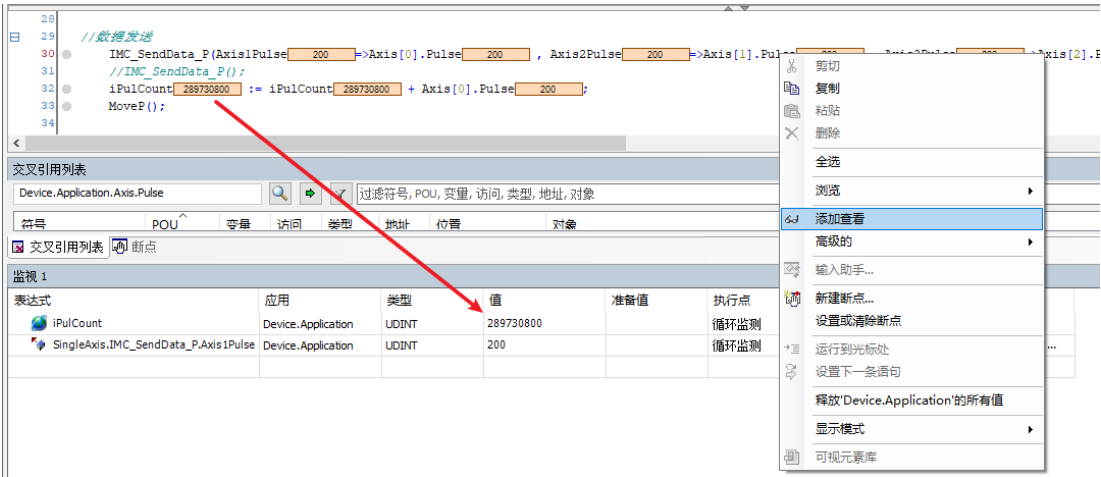
通过监控表功能可以对变量和地址进行监控，在程序运行中可以通过监控表查看监控变量的数据类型、当前值以及可以通过写入值为变量进行赋值，操作步骤如下：

步骤1 点击工具栏视图>监视，在监控选项中添加监视 1。



步骤2 在工程栏下方添加想要监控的变量，或右键点击变量，选择添加查看。

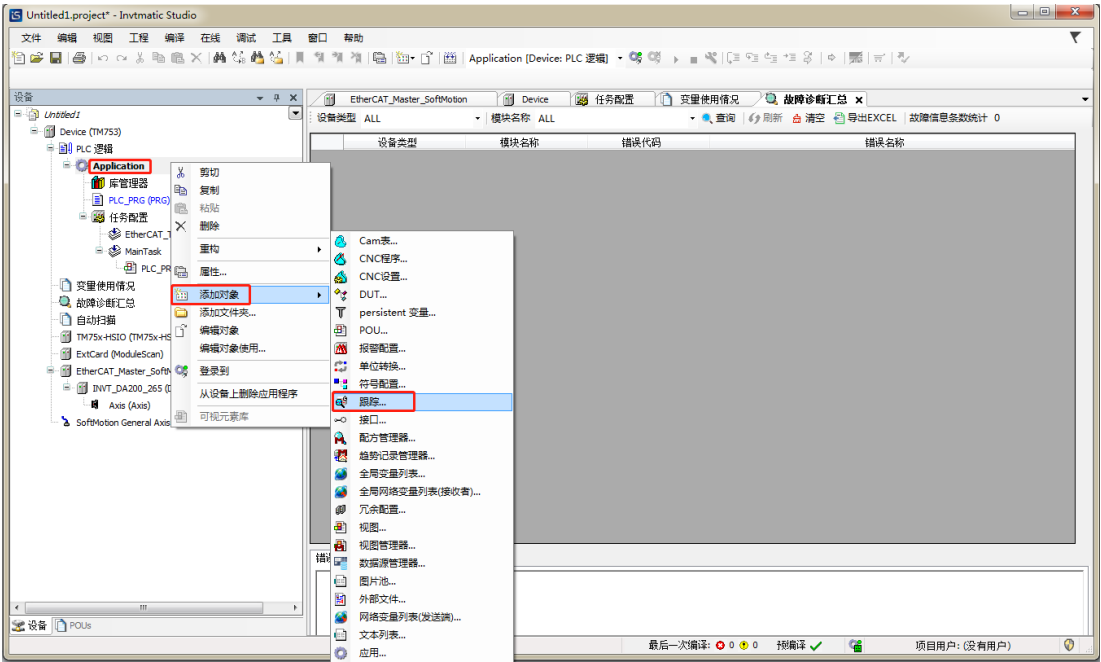




### 3.8 采样跟踪

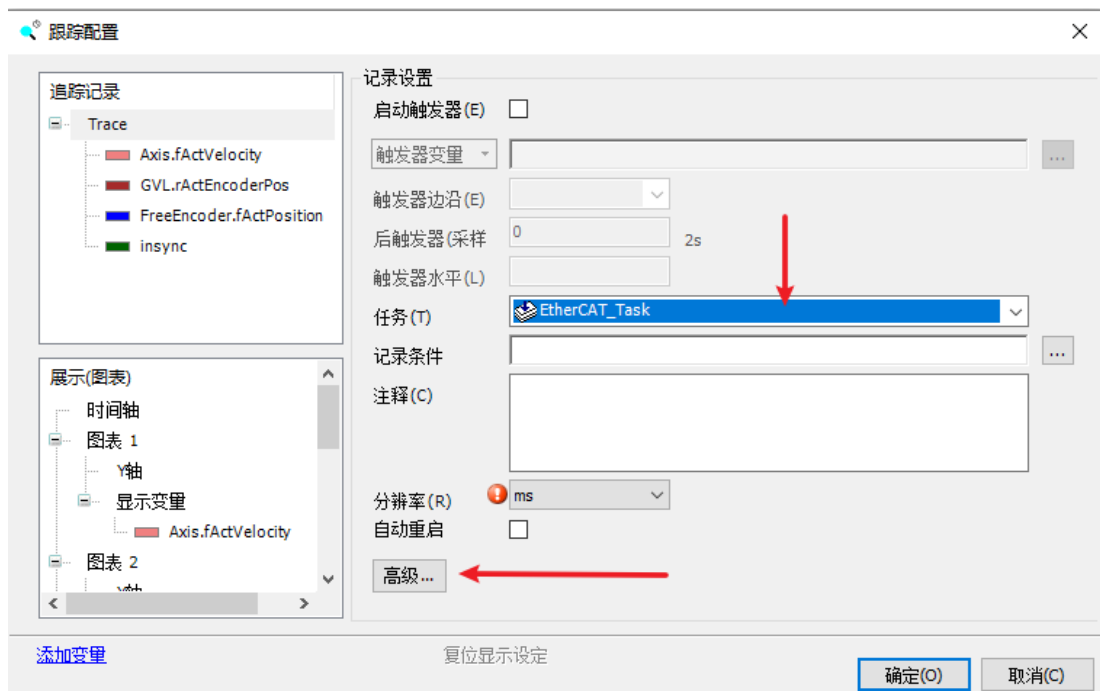
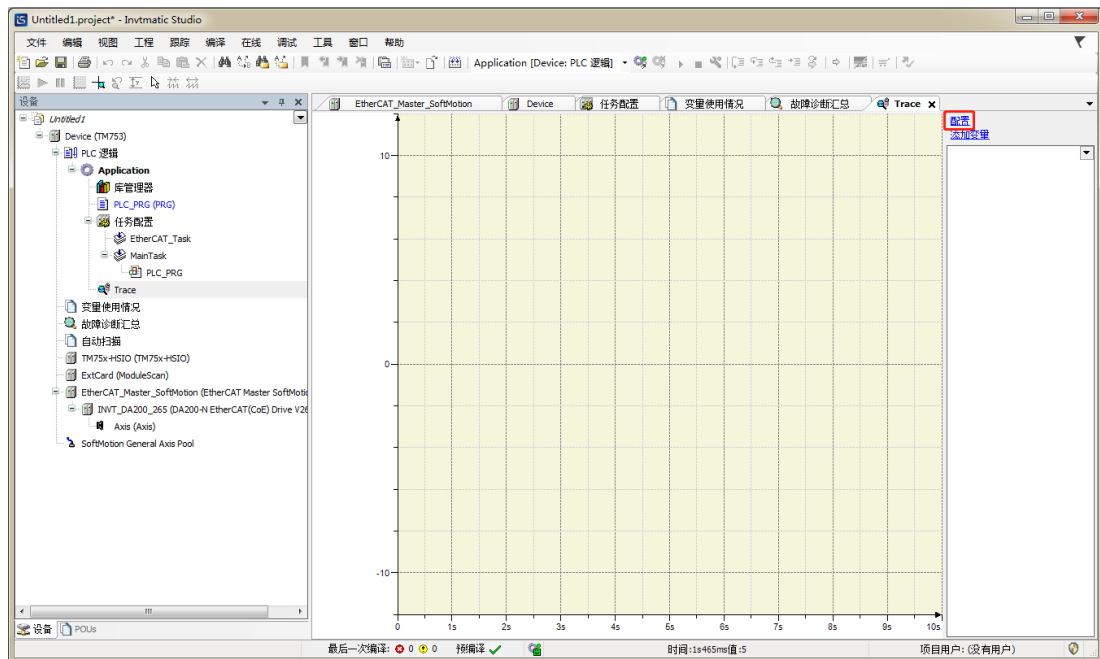
Invtmatic Studio 软件可显示在程序调试过程中变量变化的跟踪曲线，操作步骤如下：

步骤1 右键点击项目树中的 **Application**>添加对象>跟踪。



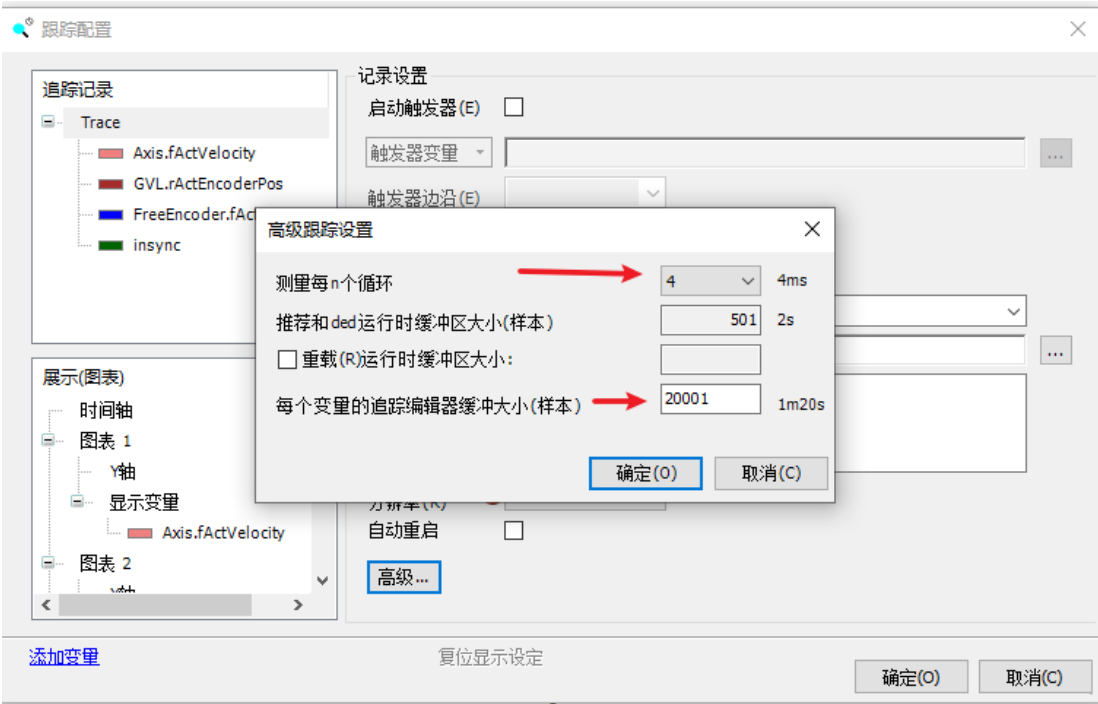


步骤2 在弹出的对话框中填写一个跟踪曲线的名称，例如下图中的 Trace。在项目树中双击跟踪曲线名，打开跟踪曲线的界面。在跟踪曲线界面的右上角，点击配置，弹出对话框如下图所示。在任务选项框中选择 EtherCAT\_Task。

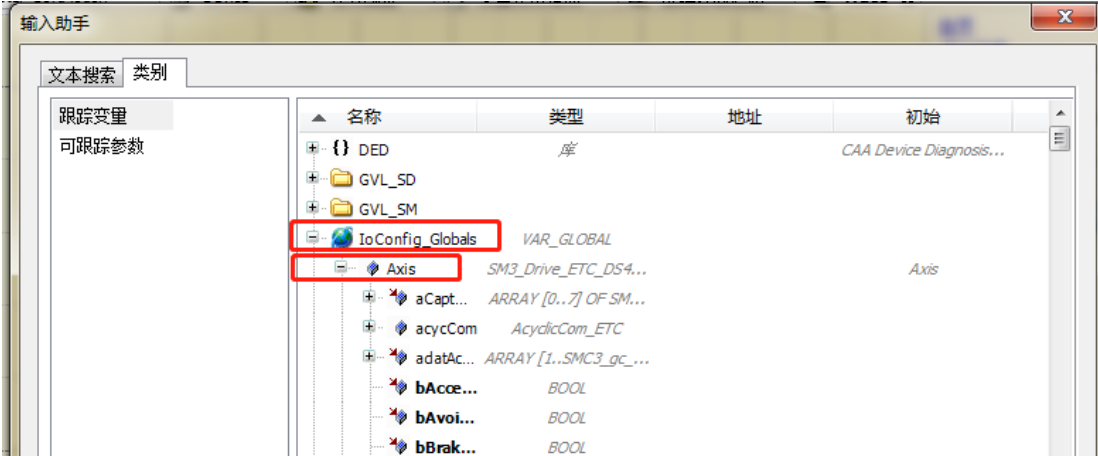




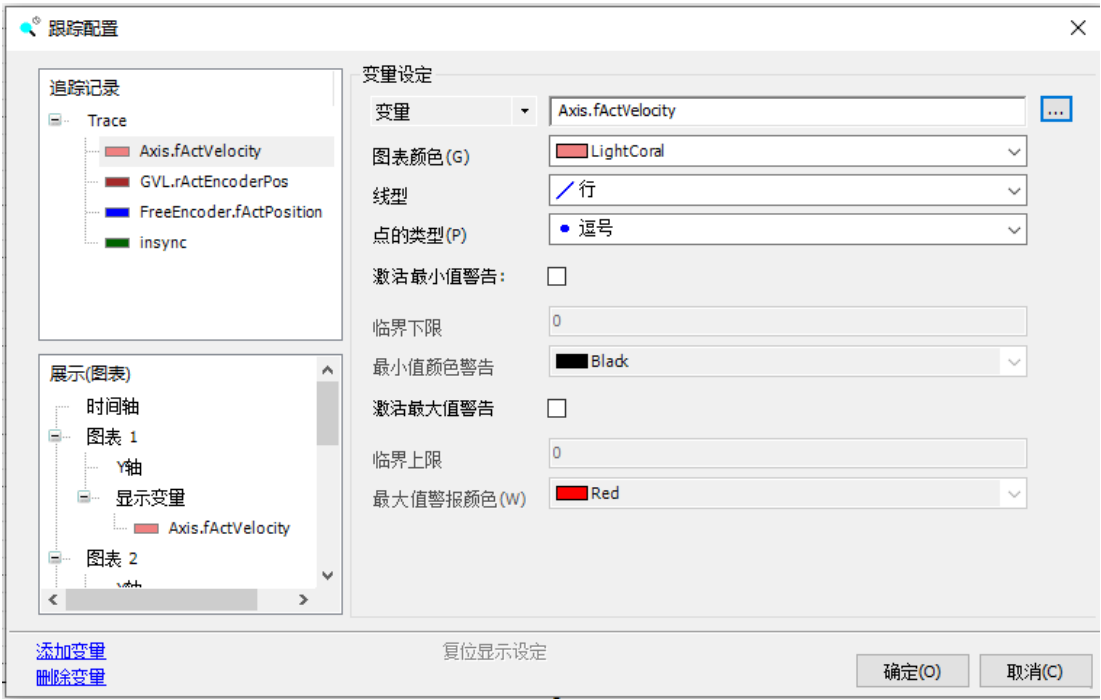
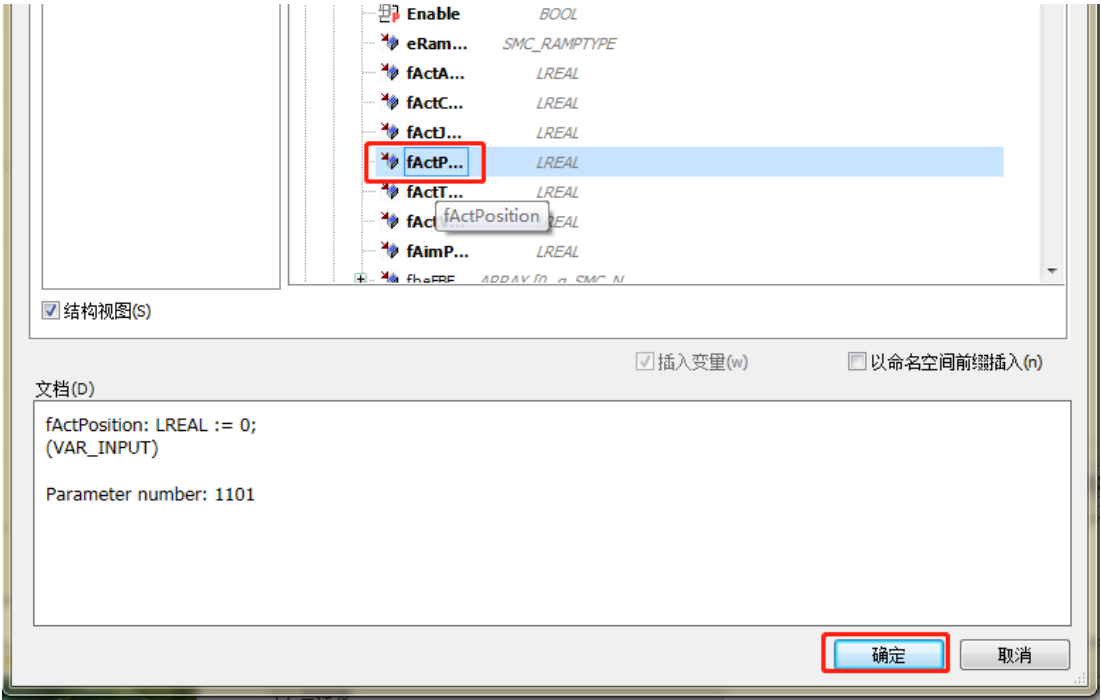
步骤3 点击跟踪配置界面中的高级，显示高级跟踪设置界面，在此界面可以设置测量每 **n** 个循环为采样周期，增加每个变量的追踪编辑器缓冲大小（样本），来加长跟踪时间。



步骤4 在跟踪配置界面变量中选择 **IOConfig\_Globals>Axis**，找到 Axis 轴的位置变量 **fActPosion**，点击确定，再添加轴的速度变量 **fActVelocity**。







### 3.9 保持变量

#### 3.9.1 特性

掉电保持变量是指在 PLC 掉电、程序下载后继续保留原来的值，常用来定义工程中重要的参数，防止 PLC 突发掉电或者程序下载而导致重要的参数丢失。掉电保持特性主要通过属性关键字 PERSISTENT RETAIN 来声明。

设置掉电保持变量代码示例：

```
VAR_GLOBAL PERSISTENT RETAIN
```



```
iVarPers1 AT %MW100: DINT;


bVarPers AT %MX1.1 : BOOL;

END_VAR
```

下表列出了执行复位、掉电等动作时，不同掉电保持变量的响应动作。

表 3-5 变量响应动作

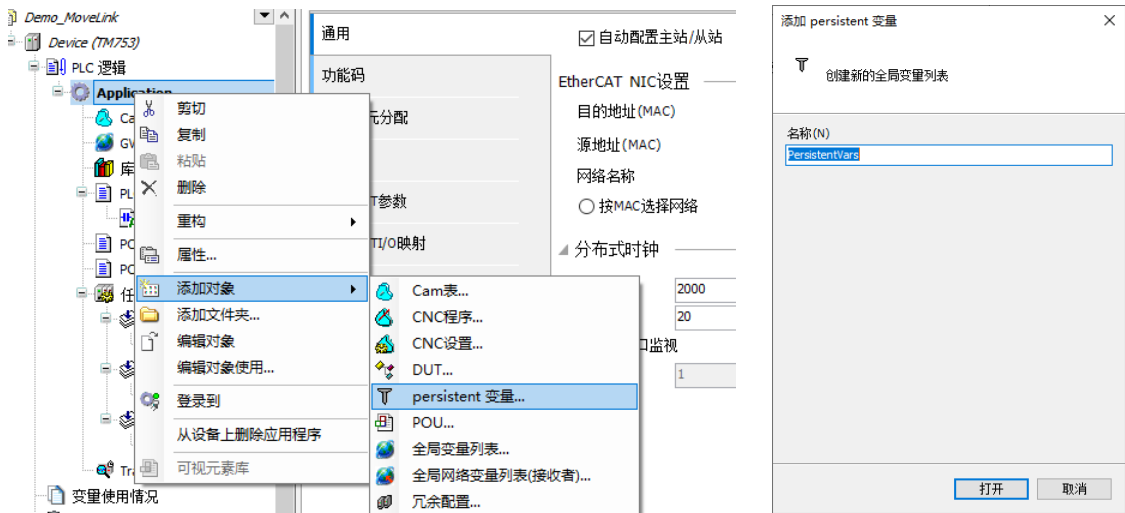
动作	VAR	VAR PERSISTENT RETAIN/ VAR RETAIN PERSISTENT	VAR RETAIN
掉电	初始化	保持原值	保持原值
热复位	初始化	保持原值	保持原值
冷复位	初始化	保持原值	初始化
初始值复位	初始化	初始化	初始化
程序下载	初始化	保持原值	初始化
在线修改	保持原值	保持原值	保持原值

-  **注意：**
- RETAIN 变量和 PERSISTENT RETAIN 变量都属于保持变量，但保持特性不同。
  - 映射到%M 地址的直接变量可以声明为保持变量，而映射到%I 和%Q 的直接变量不能声明为保持变量。


3.9.2 掉电保持变量表

如果工程中定义了掉电保持变量，则必须生成一个掉电保持变量表，否则定义的变量不具有掉电保持功能。用户可以使用掉电保持变量列表添加掉电保持变量，操作步骤如下：

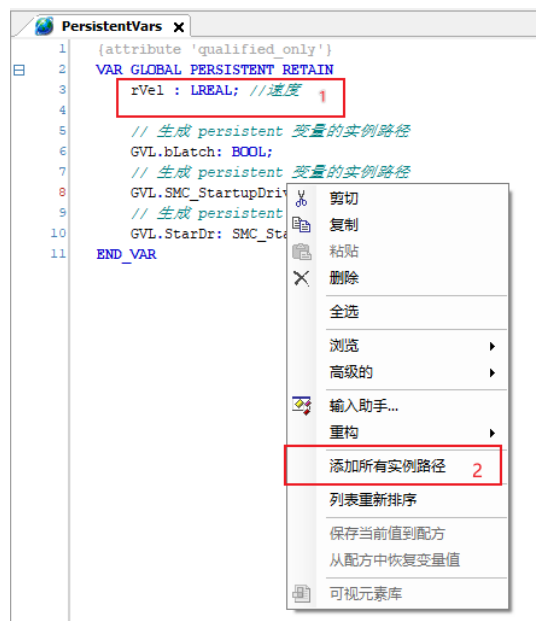
- 步骤1 在设备导航树中右键点击 **Application**，选择**添加对象>persistent 变量…**。
- 步骤2 在弹出的对话框中输入掉电保持变量列表名称后，单击**打开**即可。



用户也可以直接在掉电保持变量列表中声明掉电保持变量，如下图中的①所示。但该方式有一个缺点，不支持直接地址（如 MX0.0）定义的变量，建议用户在全局变量列表中声明 PERSISTENT RETAIN 或 RETAIN 关键字，然后通过右键点击空白界面选择**添加所有实例路径**，在掉电保持列表中生成实例路径，如下图的②所示。

-  **注意：**全局变量列表中声明的 PERSISTENT RETAIN 或 RETAIN 变量，至少有一个变量在程序中使用，才能添加到掉电保持变量列表里。



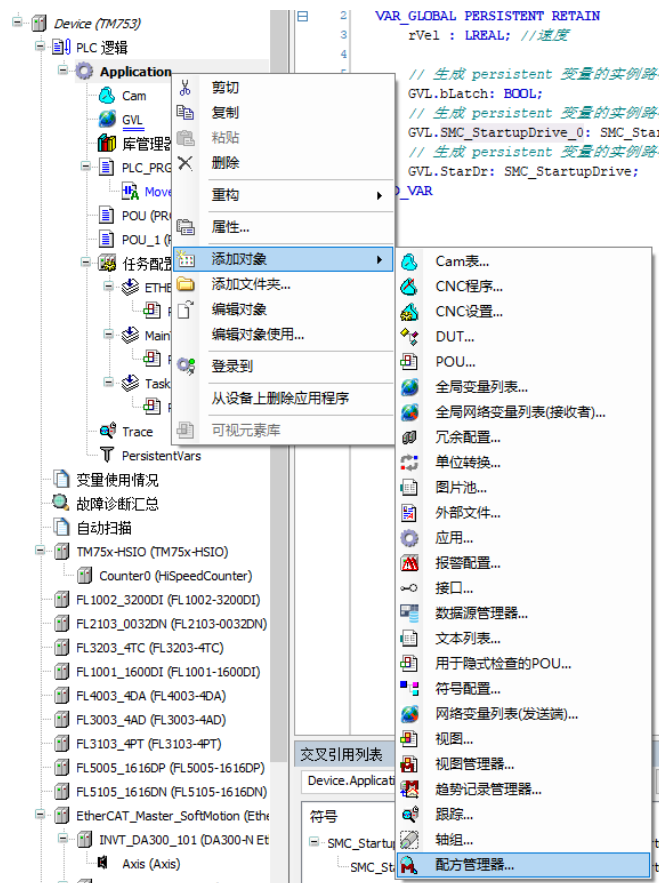


### 3.10 配方管理器

配方是一组参数值，用来提供生产产品和控制生产过程中所需的信息。例如：饼干的原料，包括白糖、鸡蛋、黄油、面粉等，烘焙时间等参数。配方也可以用于设置和监控 PLC 的控制参数。

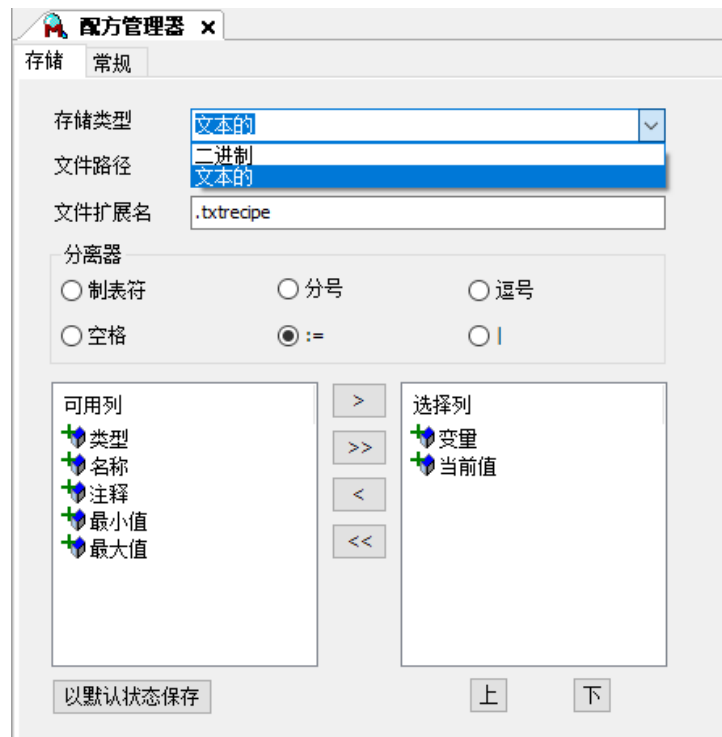
为了保存生产过程中所需的信息，用户可以从 PLC 读取配方，也可对其写入数据，还可以从文件中载入配方或者将配方存成文件，这些相互作用通过已经设置好的视图元素即可实现。

步骤1 在设备树中右键点击 **Application**，选择**添加对象>配方管理器**来添加配方管理器。





步骤2 添加配方管理器后，需要设置管理器名称，点击**打开**后，弹出下图所示的配方管理器的配置界面。

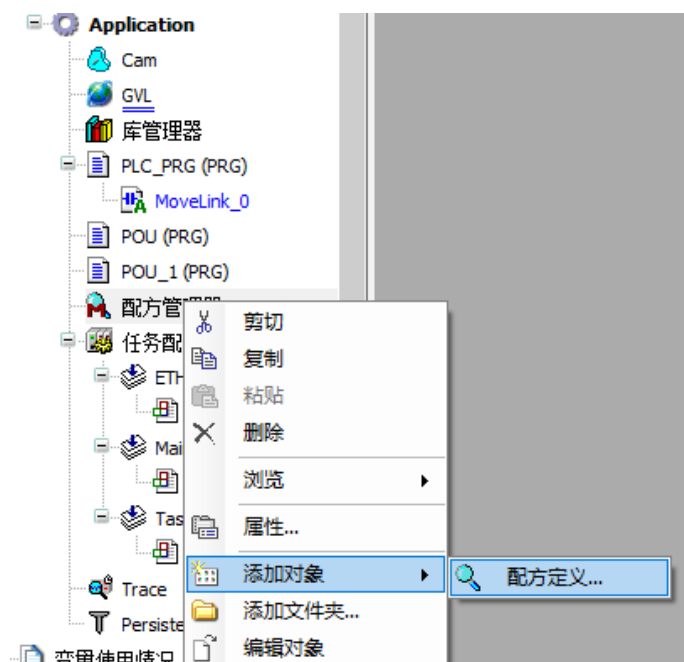


存储类型：可以选择文本或二进制类型。

文件路径：可以通过文件路径指定具体的存储位置，也可以定义存储配方的文件扩展名；文本的存储会根据所选的分离器将其分开，分离器只有在存储类型为文本时生效。

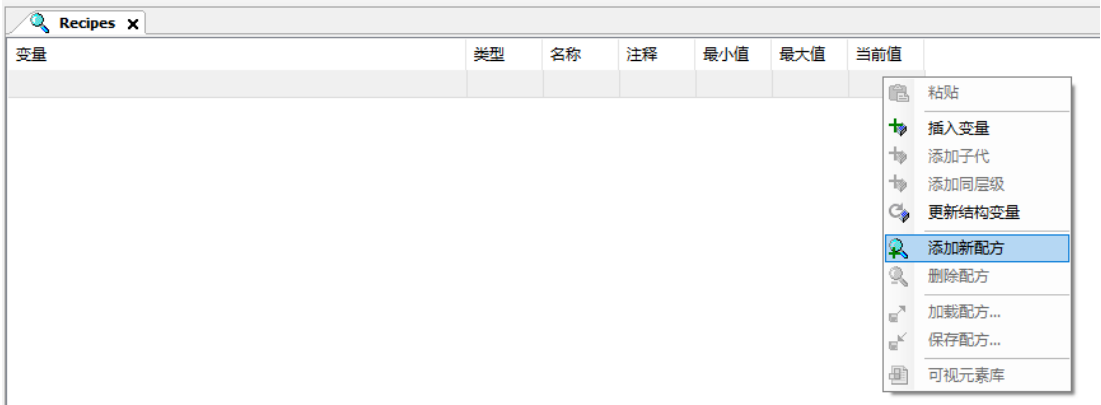
可用列：右侧为当前配方定义的列，即**选择列**，选择列中的内容会被存储。可通过单击 **>** 或 **<** 图标按钮在**可用列**和**选择列**之间移动，也可以通过 **>>** 或 **<<** 图标按钮将所有条目一次性移动。图标按钮 **上** 和 **下** 可用来调整配方定义的顺序，存储文件会按照选择列中设定的顺序保存。

步骤3 配置页面设置完成后，在设备导航树中通过鼠标右击**配方管理器**，选择**添加对象>配方定义...**，添加新的配方定义。

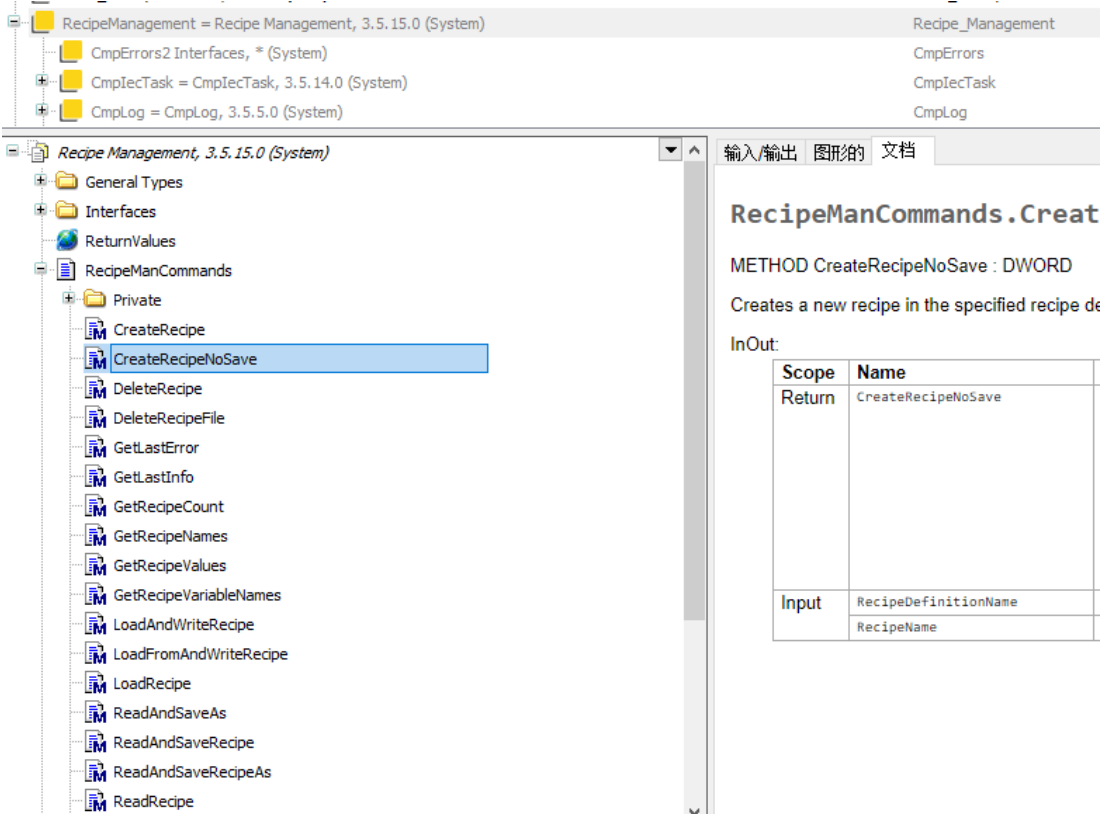





步骤4 设置新配方的名称，点击打开后，添加的配方页面如下图所示。用户可以通过点击右键点击空白位置，选择**添加新配方**来配置不同的配方值，也可以选中配方值（如配方 1）右键点击来删除配方、加载配方或保存配方。

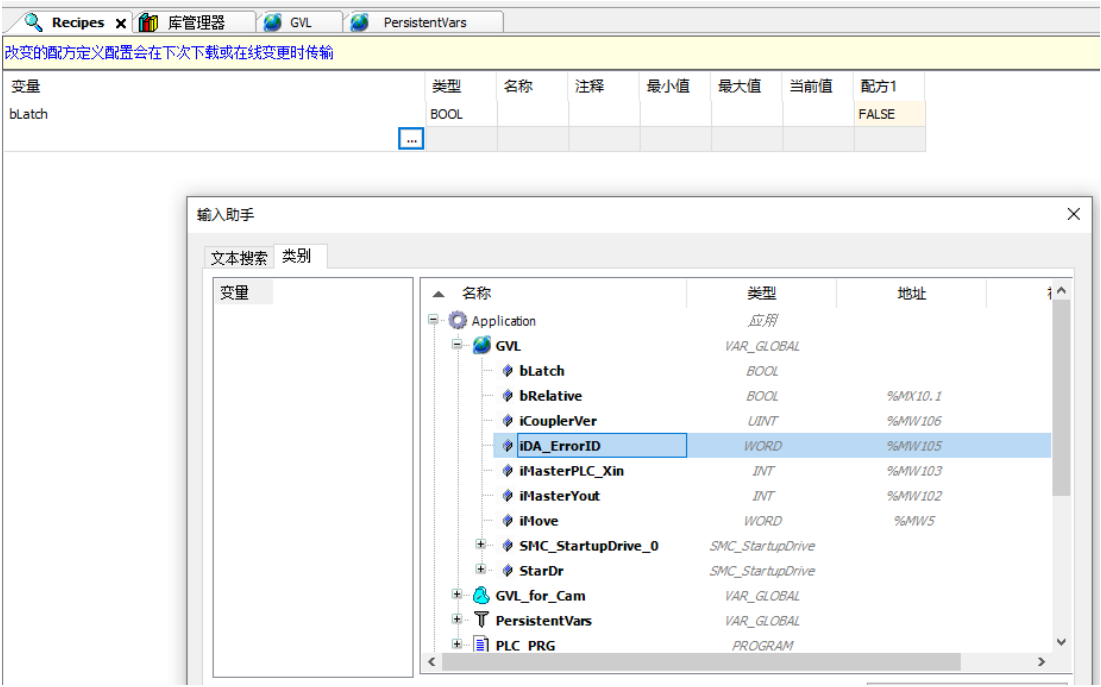


除了可以在界面中直接编辑配方文件，用户也可以通过添加“Recipe Management”库文件，使用代码对配方进行创建、删除、加载、保持等操作。

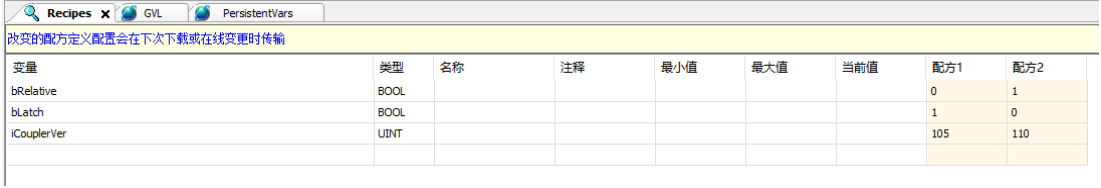




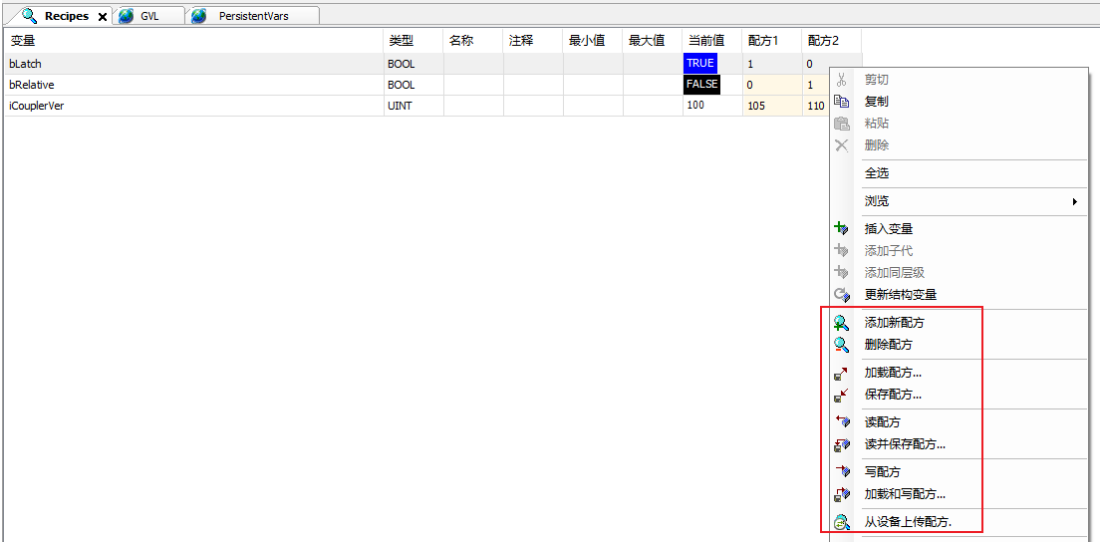
步骤5 添加完配方后，用户需在程序中定义配方需要使用的变量，然后在配方中插入变量，点击.



变量插入完成后，用户可以在配方 1、配方 2 中填写参数。



在线登入后，用户可以通过**写配方**将当前选中的配方值写入控制器中，也可以通过**读配方**从控制器中读取当前值到配方中。

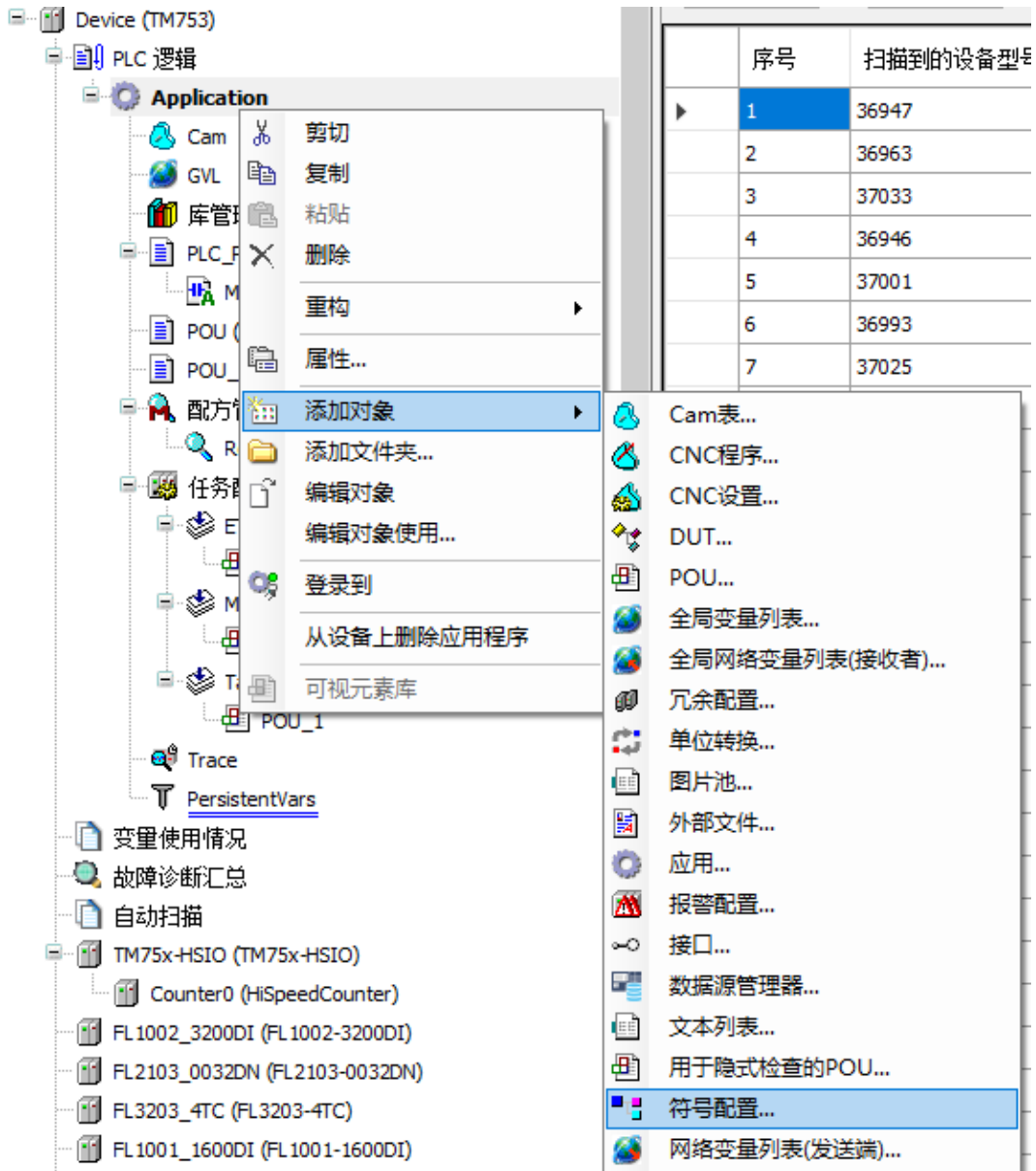




### 3.11 符号配置

使用符号配置功能，将项目变量配置为具有特定访问权限的符号。使用这些符号，可以从外部访问变量，比如 OPC 服务器。在菜单栏**编译**中选择**生成代码**时，还会生成一个符号配置文件（工程目录下后缀为\*.xml），命名方式为：设备名称+应用程序名称.xml，其中包含符号的说明。比如将该 xml 导入 HMI 中进行标签通信，访问变量。

右键点击设备树下的 **Application**，选择**添加对象>符号配置**。





选择符号配置后出现弹窗如下图所示。

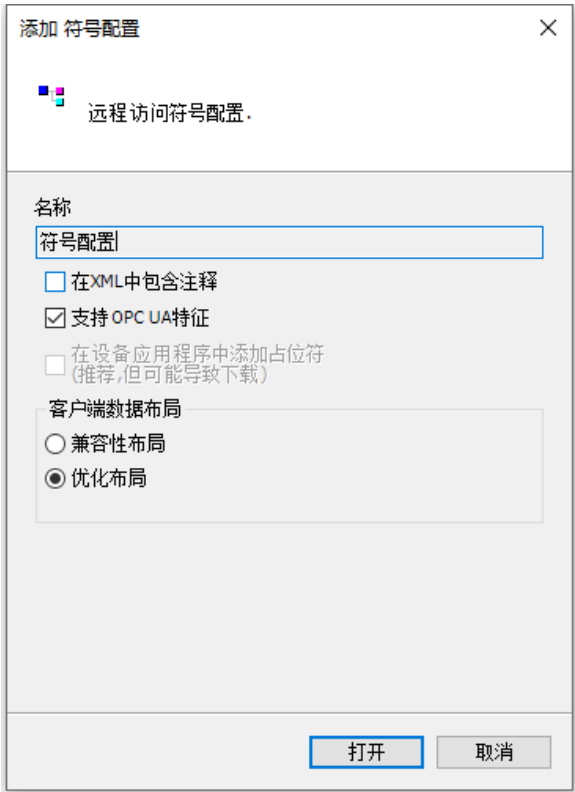
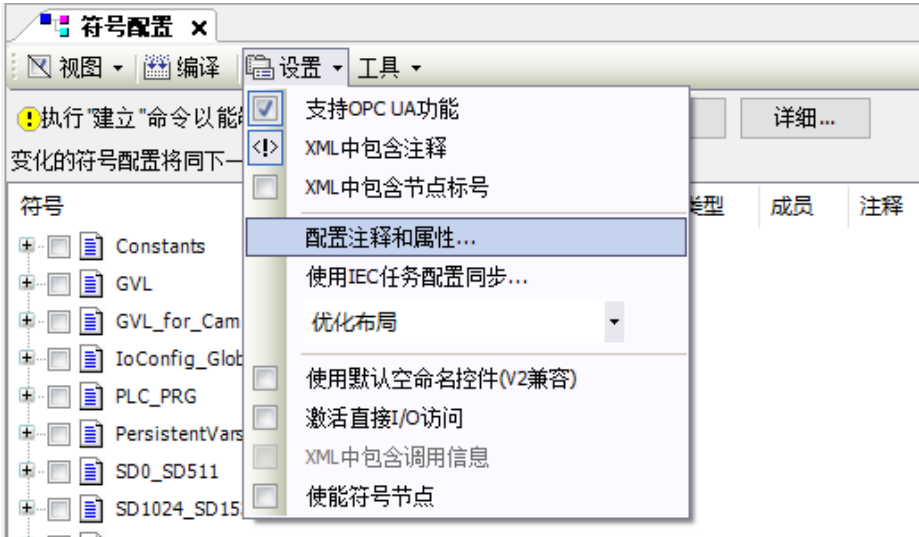


表 3-6 符号配置说明

选项	说明
在 XML 中包含注释	导出的 XML 包含变量注释信息
支持 OPC UA 特征	支持 OPC UA 访问符号变量
兼容性布局	和类型成员定义偏移大小一致，如果类型成员不完全支持符号访问，偏移大小以实际编译偏移，中间会有空白
优化布局	根据选择的类型成员计算偏移，未选择的不计算偏移

■ 符号配置-设置

生成后符号配置设置界面如下图所示。





配置注释和属性界面如下图所示。左上侧表示下载到 PLC 的数据，右上侧表示导出 XML 数据格式。

注释和属性

符号表内容

☒ 启动扩展 OPC UA 信息

☐ 包含注释

☐ 包含属性

☐ 同样包含类型节点的注释和属性

选择注释

☐ 包含文档注释: /// 它们以三斜线开始并通常 /// 在 ResS

☐ 包含正常注释: ((\* IEC /Pascal 式注释 \*) 带有双斜线的 C

☐ 总是包含两种注释

☒ 倾向于文档注释, 退回至正常注释

☐ 倾向于正常注释, 退却到文档注释

XML 标志文件内容

☐ 包含命名空间节点标志

☒ 包含注释

☐ 包含属性

☐ 同样包含类型节点的注释和属性

滤波器属性(不区分大小写)

☒ 包含所有属性("foo","bar","foo.bar")

☐ 匹配简单的标识符("foo","bar")

☐ 包含属性以以下内容开始:

☐ 带有正则表达式的滤波器属性:

确认

取消

表 3-7 符号配置注释和属性说明

选项	说明
符号	一般表示变量，符号属性表示变量特性（Attribute 信息）。
注释格式	表示注释下载或者显示格式。
属性匹配	导出 XML 或者下载到 PLC 时，哪些属性信息是否包含。

符号：一般表示变量，符号属性表示变量特性（Attribute 信息）。

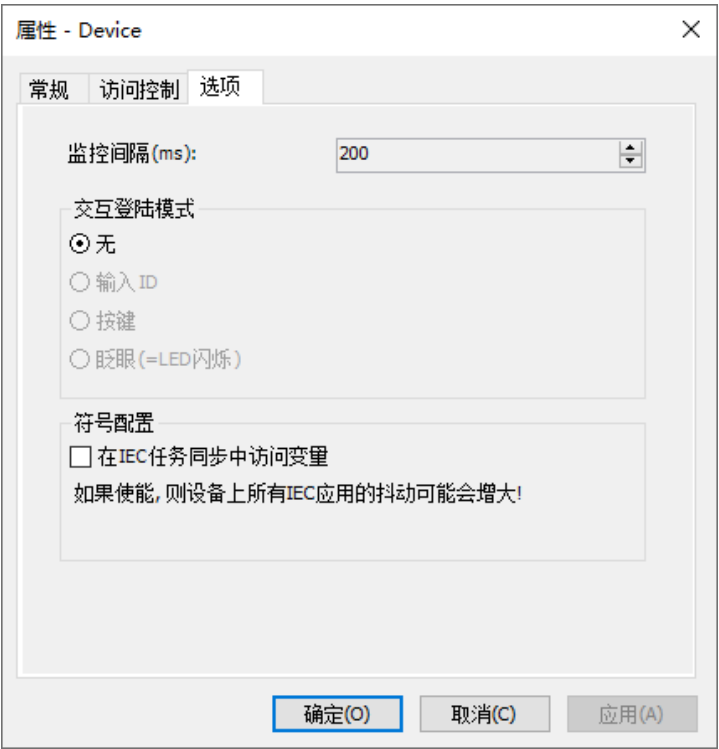
注释格式：表示注释下载或者显示格式。

属性匹配：导出 XML 或者下载到 PLC 时，哪些属性信息是否包含。

匹配规则四种为：包含所有属性、匹配简单的标识符，以特定字符串开始和带有正则表达式。

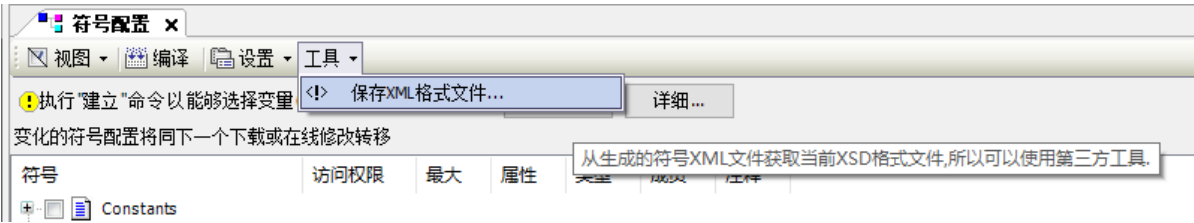
配置和 IEC 任务同步：在 Device 属性选项中，选择其他接口访问符号变量时，是否和 IEC 同步，不能 IEC 执行期间访问，防止变量不同步。





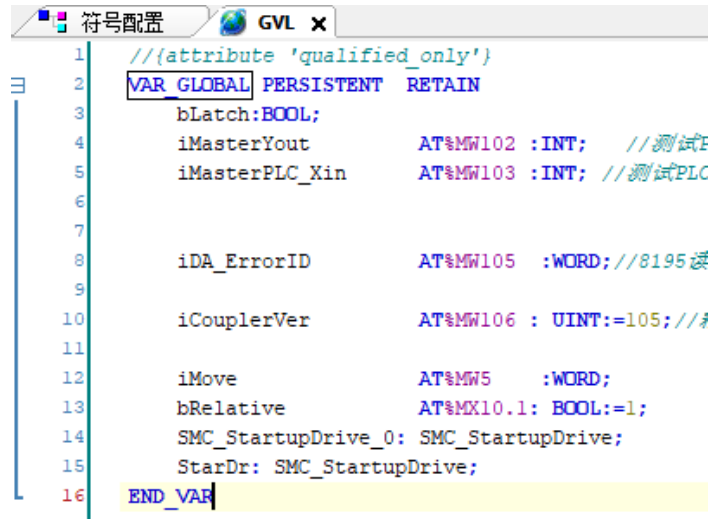
■ 符号配置-工具

保持 XML 格式文件可以导出 XML 数据模型，如果第三方解析离线符号，可以参考此数据模型。



■ 符号配置过程样例

新建全局变量 bLatch，iMasterYout 并且在用户程序中至少应用一个变量，如下图所示。

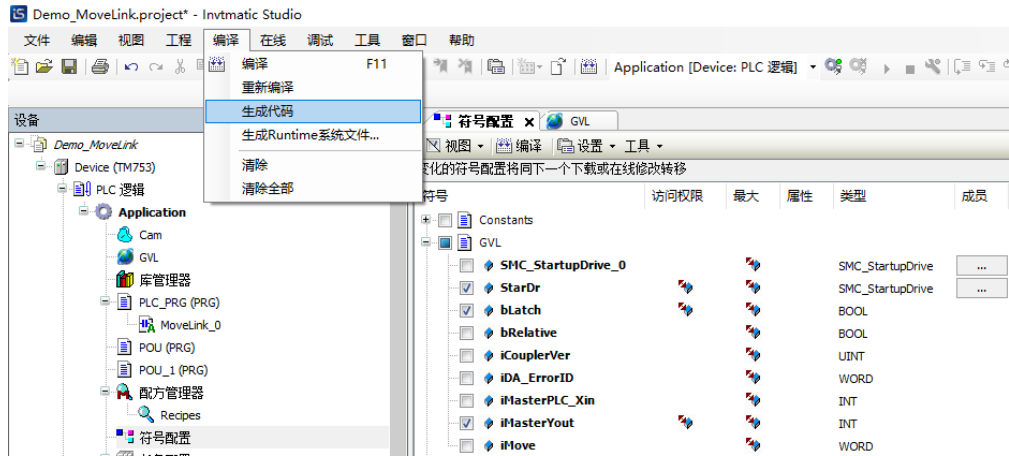


**注意：**如果单个全局变量表中任一变量都没有在用户程序中应用，则对应的变量表不会出现在符号配置中。



配置样例步骤如下：

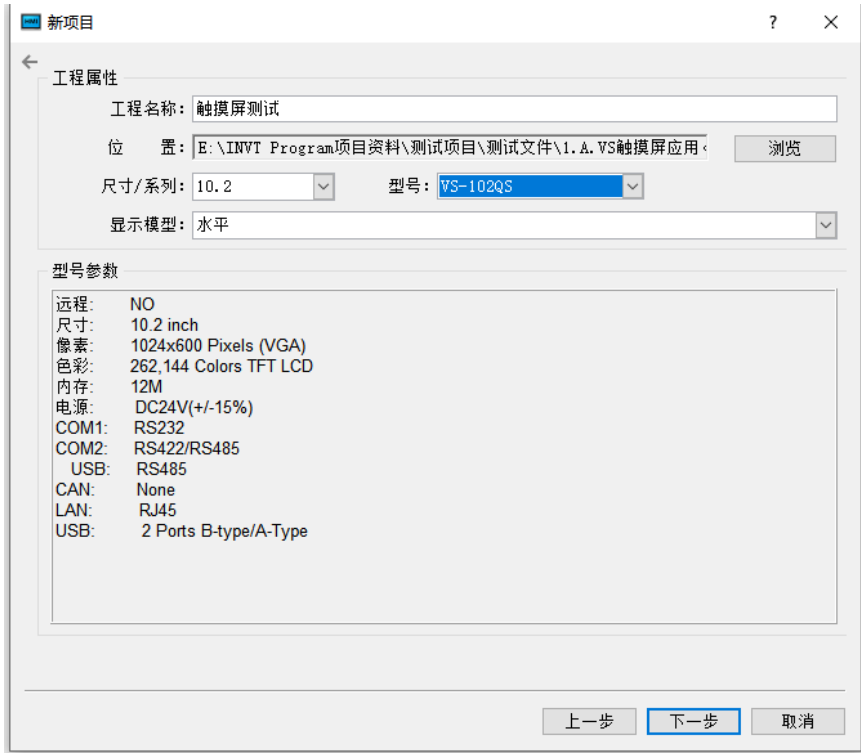
步骤1 在 **Application** 中添加符号配置后，勾选在 **XML 中包含注释**，点击上侧工具栏界面中**编译>编译**，对应变量表及变量出现在右侧符号配置中，对所需要配置的变量表进行勾选，并且配置好对应的访问权限（只读、只写、可读可写）后在上侧工具栏界面中选择**编译>生成代码**。



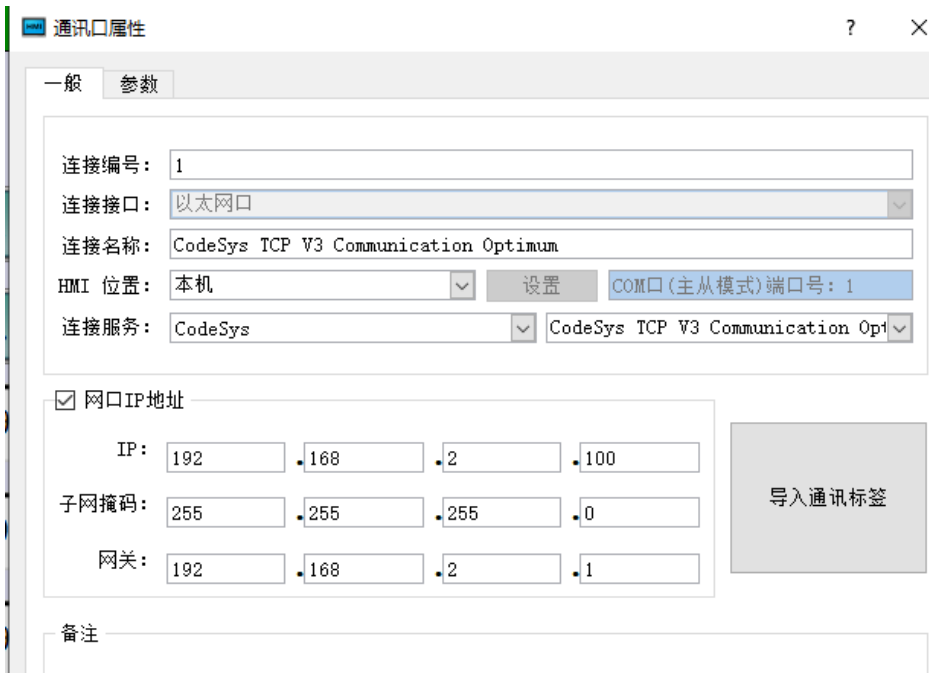
步骤2 可在工程所在目录文件夹下找到生成后缀为.xml 的文件，用来导入到触摸屏中进行标签通信，如下图所示。

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

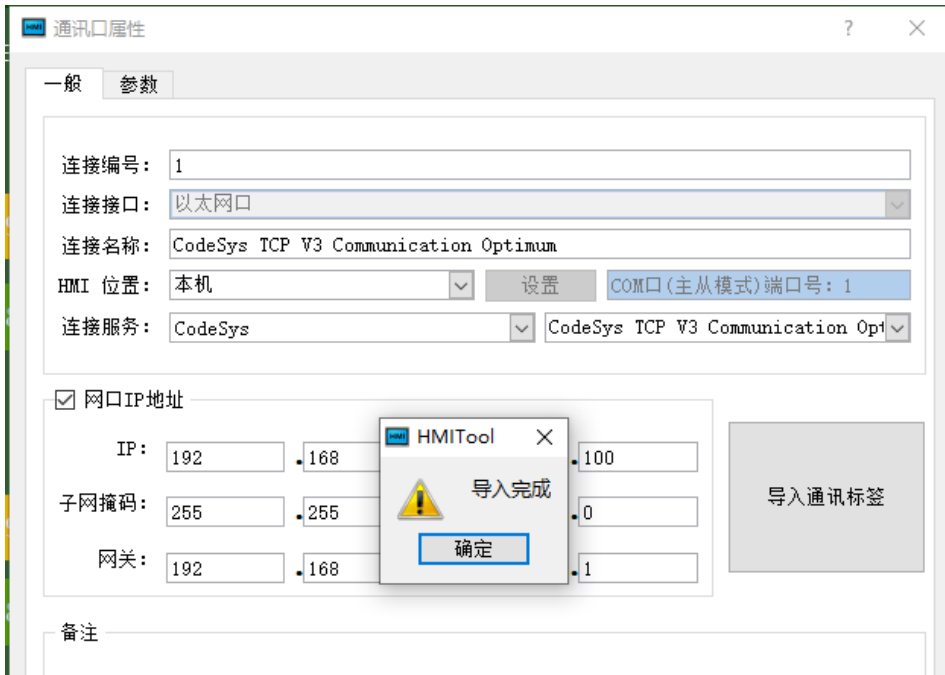
步骤3 做完 PLC 上述步骤后，以 VS-102QS 触摸屏为例，打开 HMITOOL 软件，点击**新建工程**，型号选择“VS-102QS”，连接端口选择“以太网口”，连接服务选择“CodeSys”和“CodeSys TCP V3 Communication Optimum”，如下图所示。





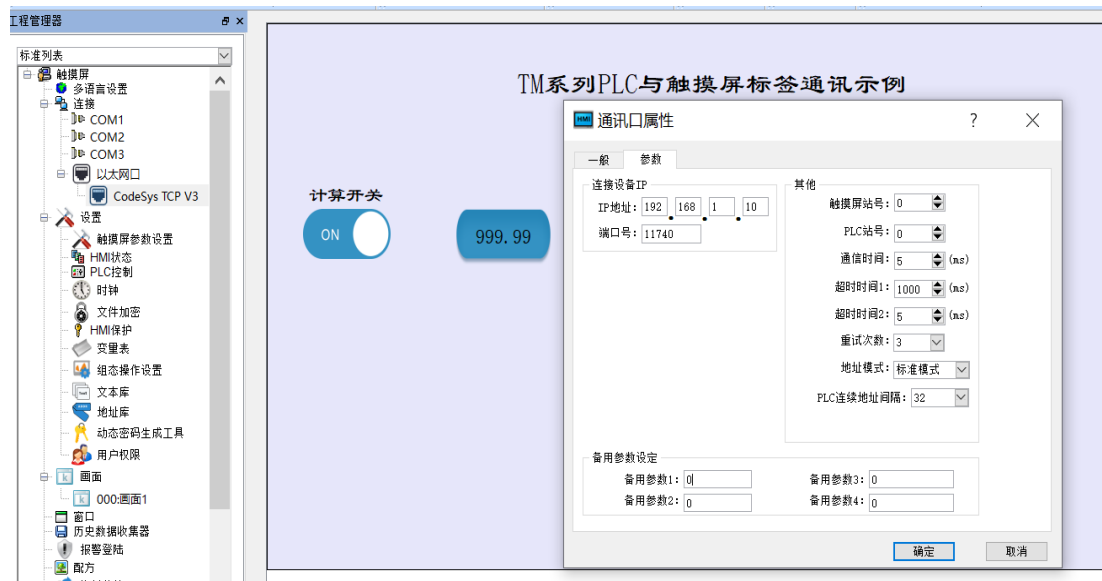


步骤4 点击**导入通讯标签**，选择刚刚生成的 XML 标签，点击**打开**，显示**导入完成**，点击**确定**，如下图所示。

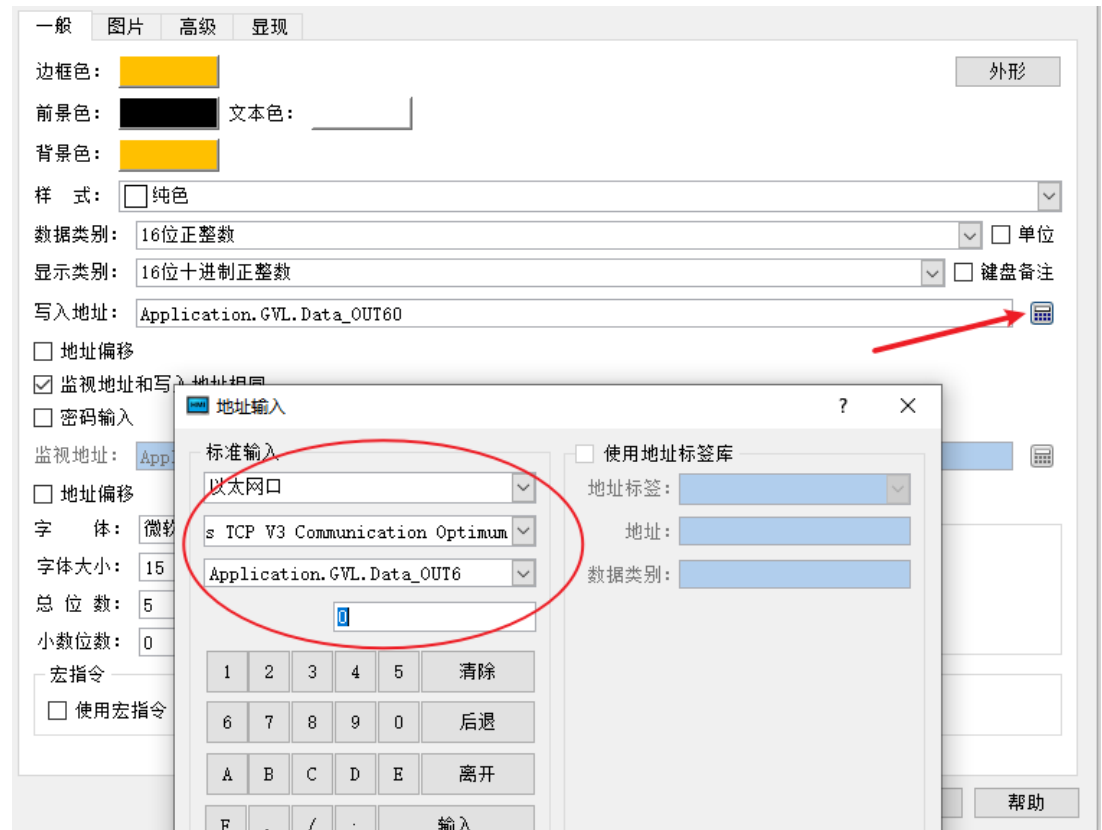




步骤5 将通讯参数的 IP 地址设置为“192.168.1.10”，端口号设置为“11740”，设置完成点击**确定**，如下图所示。



步骤6 选择上图画面中的**输入框**，并绑定对应 PLC 的地址，如下图所示，完成后即可实现标签通信。



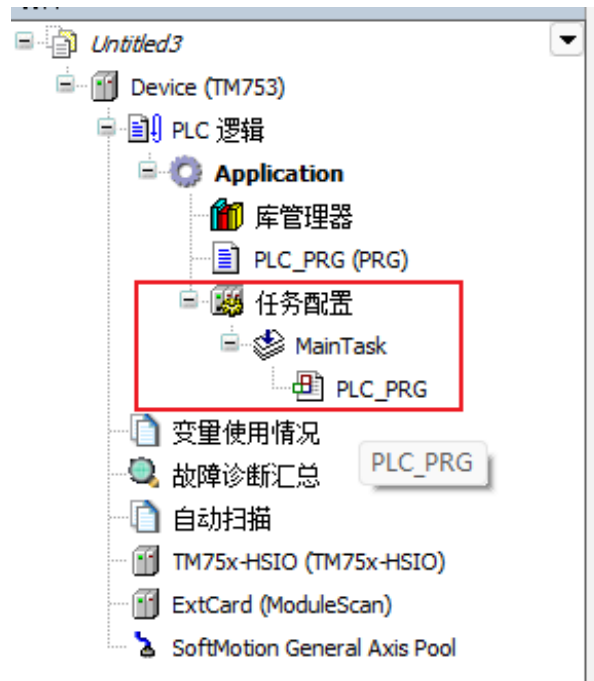
### 3.12 任务配置

每个 PLC 应用程序（工程）的工程目录树中都包含“任务配置”。在任务配置中，可定义一个或多个任务来控制和执行控制器中的应用程序。“任务”被配置后，可以使一系列程序或功能块周期性地执行或由一个特定的事件触发开始执行程序。一个任务可调用一个或多个程序块(POU)。通过设置任务的优先级和条件，可以定义处理任务的顺序。用户还可以为每个任务配置看门狗，当任务的执行周期过长时控制器提示异常。

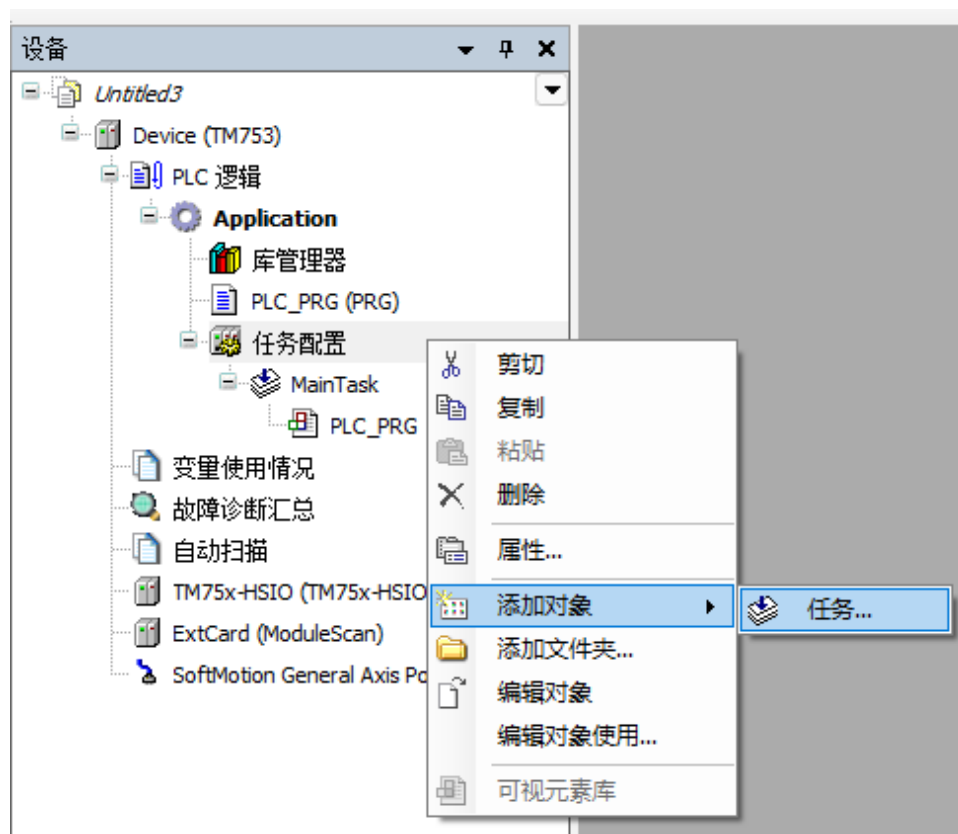


### 3.12.1 添加任务

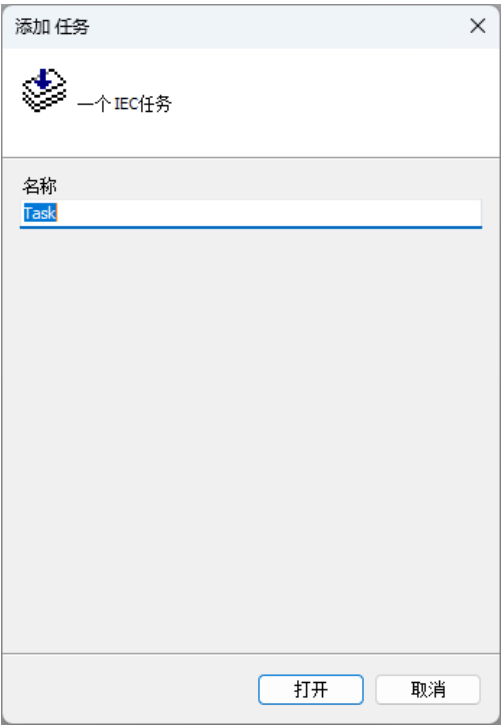
新建工程后，Invntmatic Studio 会自动添加**任务配置**，自动生成的任务如下图所示。



用户如需添加新的任务，可以鼠标右击**任务配置**，选择**添加对象>任务**。在弹出界面中输入自定义的任务名称后，点击**打开**即可完成任务添加，如下图所示。







3.12.2 任务设置

打开新建的任务后，会显示下图所示的设置界面。



- 优先级：可输入 0~31。数值越大优先级越低，0 级为最高优先级。
- 类型：任务类型及其执行逻辑，详见下表。

表 3-8 任务类型

类型	执行逻辑	设置项
循环	按照设定的间隔循环执行	时间间隔
事件	设置的全局变量触发上升沿后，任务开始执行	触发变量
惯性运行	在程序开始时和完整过程结束时以连续循环的方式自动开始处理任务	-



类型	执行逻辑	设置项
状态	设置的全局变量为 True 时，任务开始执行	触发变量

- 看门狗：如果任务超过了看门狗的当前设置时间，则该任务将以错误状态（异常）暂停。发生错误的任务所在的应用程序及其子应用程序也将暂停。这样，受影响的应用程序的所有任务也将停止。看门狗相关的功能详见下表。

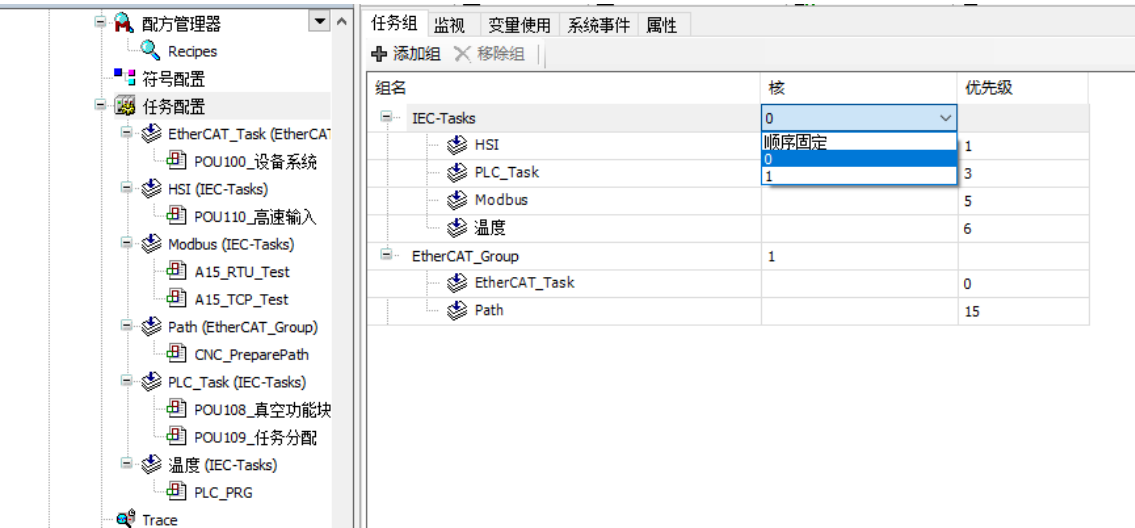
表 3-9 看门狗的功能

选项	功能
使能	是否使能看门狗功能
时间	设置看门狗超时时间
灵敏度	设置数值，即超过设定时间几倍后，触发看门狗

- POU 调用：可以对任务调用的 POU 进行设置，POU 调用的顺序会影响。

3.12.3 TP6000 多核任务配置

TP6000 系列 PLC 支持双核心配置，在任务配置界面，默认有 2 个任务组，用户可自行添加任务组，任务组可分配使用核 0 或核 1。每个任务可选择优先级，一般要求包含 EtherCAT\_Task 的 EtherCAT\_Group，配置为核 1，具有更好的实时性。





# 4 硬件配置

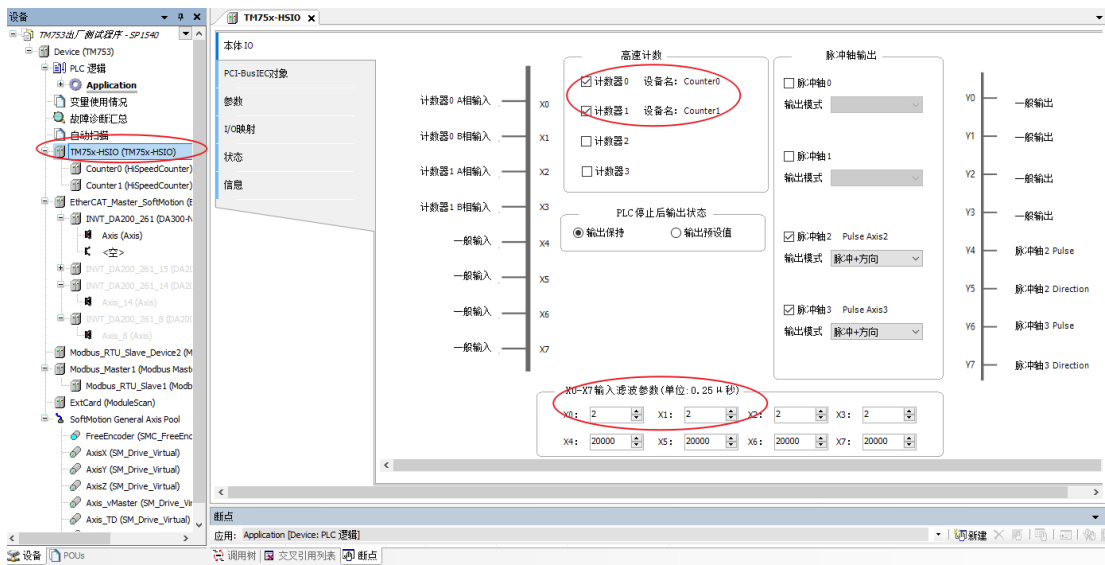
## 4.1 本体高速 I/O 配置

TM700 系列 PLC 双击设备窗口中的 **TM75x-HSIO** 选项，会弹出 HSIO 的参数配置界面。在该界面中可以对高速 I/O 功能及相关参数进行配置，具体包括高速计数功能、高速输出功能和高速输入沿中断的功能。

**注意：**

- 1、 若不勾选**计数器**，则可做普通输入端口使用。
- 2、 若要使用计数器功能，勾选**计数器 0(1、2、3)**，则相应的输入端口做计数器的输入信号源使用，默认 X0 和 X1 为计数器 0 的输入信号源，后续计数器与高速输入端口对应关系以此类推。启用计数后，生成 Counter 设备，可进行详细设置。

图 4-1 高速 I/O 端口的组态界面



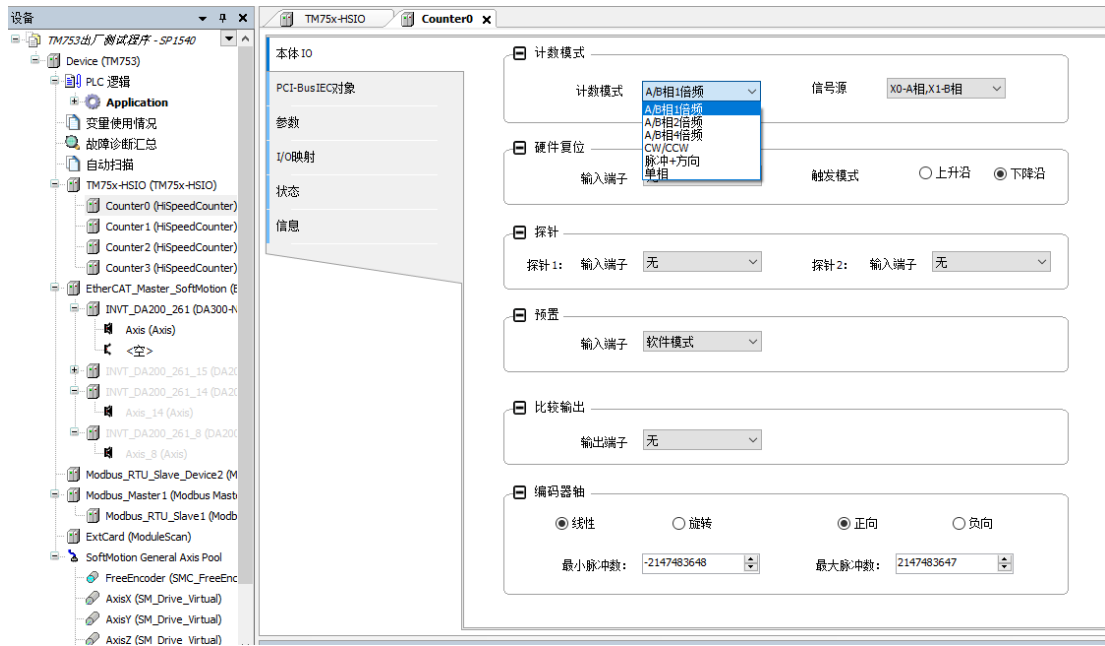
### 4.1.1 计数器界面组态

输入端口可以设置为 5 种功能，分别是：普通输入功能、计数功能、触发锁存、Z 信号功能和脉宽测量功能。计数功能模块可以对输入脉冲进行计数和计算，检测位置、速度、频率等，输入脉冲最大频率 200kHz。

双击 **TM75x-HSIO** 选项下的 **Counter** 设备，可以进行高速计数器详细设置。



图 4-2 高速计数器参数配置



- 计数模式：为计数器的计数模式，可选择 6 种计数模式，如：A/B 相 1 倍频、A/B 相 2 倍频、A/B 相 4 倍频、脉冲+方向、单相计数和 CW/CCW。若选择使用 A/B 相 1/2/4 倍频、脉冲+方向或 CW/CCW 计数模式，默认 X0 及 X1 为计数器 0 的输入信号源，后续计数器与高速输入端口对应关系以此类推，也可按需要设置信号源；选择单相计数，则计数器 0 的输入信号源可选择 X0~X7；
- 输入计数方向：可选择正常或负向。
- 计数模式：可选择旋转或线性。
- 最大脉冲数：默认为 2147483647，范围为-2147483648~2147483647。

**注意：**上限值不得小于下限值，若设置值小于下限值，系统自动将下限值变成与上限值一致。

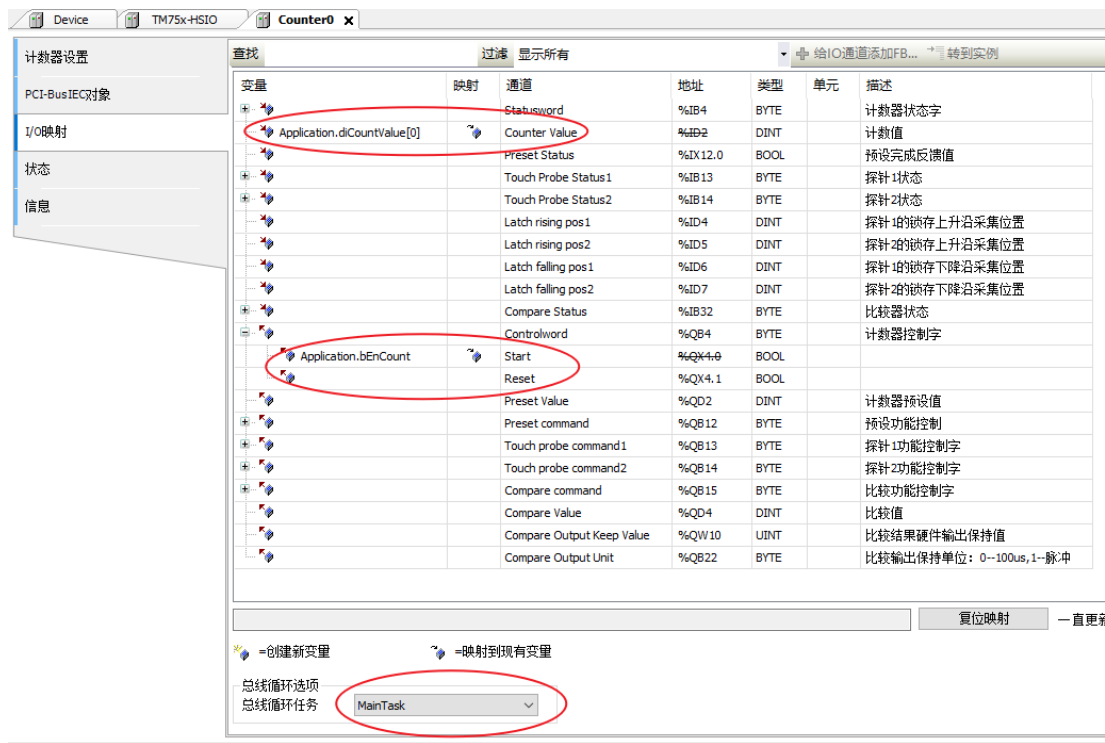
- 最小脉冲数：默认为-2147483648，范围为-2147483648~2147483647。
- 预置输入端子：当预置功能配置为外部 DI 预置模式时，需要配置该输入口，可在 IN0~IN7 中任意选择一个输入口，实现预置值功能，也可以设置为软件触发预置。
- 探针：每路计数器可配置 2 路探针输入口，实现锁存计数器值功能。探针 1 为 1 号探针输入口，可配置为 IN0~IN7 任一输入口。探针 2 为 2 号探针输入口，可配置为 IN0~IN7 任一输入口。
- 比较输出端子：使用高速比较（单点/线性/队列）输出功能时，配置该输出口，可配范围为 OUT0~OUT7。
- 硬件复位：当计数值复位功能配置为外部 DI 模式时，需要配置该输入口，可在 IN0~IN7 中任意选择一个输入口，实现清除计数值功能。



4.1.2 计数功能

计数功能控制，例如在 I/O 映射中，可以设置计数器清零及计数器使能控制位。

图 4-3 高速计数器参数映射




Controlword/Reset 为计数器清零控制位；Controlword/Start 为计数器使能控制位。其它功能可按下表说明配置。

表 4-1 计数器控制字和状态字说明

参数名称	参数类型	说明
Controlword	BYTE	计数器控制参数。 Bit0: 计数使能，高有效 Bit1: 计数清零，上升沿有效
Preset Value	DINT	计数器预设值。
Preset command	BYTE	预设功能控制。 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	BYTE	探针 1 功能控制字。 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	BYTE	探针 2 功能控制字。 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
Compare command	BYTE	比较功能控制字。 Bit0: Start Bit1: Enable Bit2: Reset



参数名称	参数类型	说明
Compare Value	DINT	比较值。
Compare Output Keep Value	UINT	比较结果硬件输出保持值。
Compare Output Unit	BYTE	比较输出保持单位。 0: 100μs, 1: 脉冲
Statusword	BYTE	计数器计数状态反馈。 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	DINT	计数器计数值。
Preset Status	BYTE	预设状态反馈。 Bit0: 预设完成状态反馈 (1: done)  注意：预设 done 信号需要清除，没清除之前不能修改预设值。
Touch Probe Status1	BYTE	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  注意：预设 done 信号需要清除，没清除之前不能修改预设值。
Touch Probe Status2	UDINT	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  注意：预设 done 信号需要清除，没清除之前不能修改预设值。
Latch rising pos1	DINT	探针 1 锁存上升沿采集位置。
Latch rising pos2	DINT	探针 2 锁存上升沿采集位置。
Latch falling pos1	DINT	探针 1 锁存下降沿采集位置。
Latch falling pos2	DINT	探针 2 锁存下降沿采集位置。
Compare Status	BYTE	比较器状态。 Bit0: 比较输出 busy 状态反馈 Bit1: 比较输出 done 状态反馈

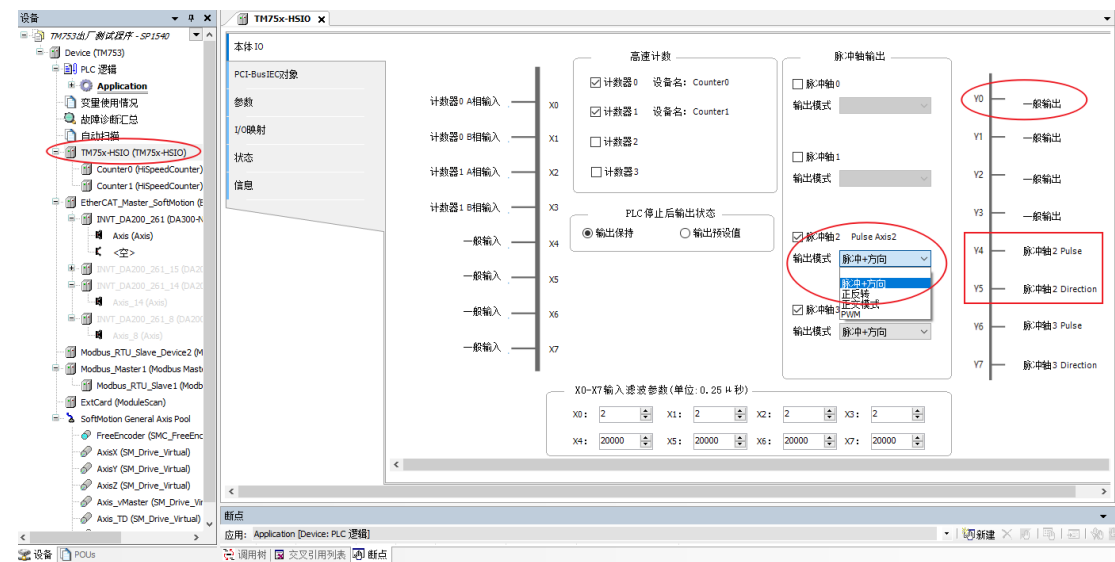
注意：计数预置控制，Preset command 可以使用软件操作预置计数器值，当使用 PresetActiveSoft 成功预设计数器值后，须使用 PresetDoneClean 清除 Preset Status 状态，才能再次重新预设计数器值。



4.1.3 输出端口功能说明

输出端口可以设置为 3 种功能，分别为普通输出功能、高速脉冲输出功能和比较输出功能。如下图所示，系统默认为普通输出功能 Y0~Y7。当需要脉冲输出功能时，需要勾选对应通道的脉冲轴。

图 4-4 高速输出端口配置



4.1.4 普通输出功能

系统默认为普通输出端口，范围为 Y0~Y7。  
TM 系列输出端口共有 8 路输出信号，仅支持单端输出，信号类型为漏型输出。Y0~Y7 共用公共端 COM。

表 4-2 TM700 系列普通输出端口说明

示意图	左侧信号	左侧端子	右侧端子	右侧信号
	X0 输入	A0	B0	Y0 输出
	X1 输入	A1	B1	Y1 输出
	X2 输入	A2	B2	Y2 输出
	X3 输入	A3	B3	Y3 输出
	X4 输入	A4	B4	Y4 输出
	X5 输入	A5	B5	Y5 输出
	X6 输入	A6	B6	Y6 输出
	X7 输入	A7	B7	Y7 输出
	SS 输入公共端	A8	B8	COM 输出公共端

4.1.5 高速脉冲输出功能

勾选脉冲轴后，则信号端口配置为高速脉冲输出功能，8 个输出端口均可配置为高速脉冲输出。  
高速脉冲输出支持脉冲+方向、正反向、正交模式、PWM 模式四种模式。



图 4-5 高速输出模式

脉冲轴

☒ 脉冲轴 0

Pulse Axis0

输出模式

脉冲 + 方向

☒ Z信号

X4

☒ 脉冲轴 1

Pulse Axis1

输出模式

正反向

☐ Z信号

☒ 脉冲轴 2

Pulse Axis2

输出模式

正交模式

☐ Z信号

☒ 脉冲轴 3

Pulse Axis3

输出模式

PWM

☐ Z信号

Y0

脉冲轴 0 Pulse

Y1

脉冲轴 0 Direction

Y2

脉冲轴 1 Forward

Y3

脉冲轴 1 Reverse

Y4

脉冲轴 2 A\_Phase

Y5

脉冲轴 2 B\_Phase

Y6

脉冲轴 3 PWM

Y7

脉冲轴 3 Direction

4.1.6 比较输出功能

比较输出是输出计数器单值比较的结果，每个计数通道都有比较输出功能。如果计数器的值与设定的比较值相等，则输出高电平，不相等则输出低电平。

- 比较输出端口在对应在计数器通道中配置

图 4-6 比较输出配置

计数器设置

PCI-BusIEC对象

I/O映射

状态

信息

计数模式

计数模式

A/B 相 倍频1

信号源

X0-A Phase,X1-B Phase

硬件复位

输入端子

触发模式

☐ 上升沿

☒ 下降沿

探针

探针 1

输入端子

X2

探针 2

输入端子

X3

输出预设值

输入端子

软件模式

比较输出

输出端子

Y6

编码器轴

☒ 线性

☐ 旋转

☒ 正向

☐ 负向

最小脉冲数

-2147483648

最大脉冲数

2147483647



■ 比较输出控制

图 4-7 比较输出控制映射

计数器设置

PCI-BusIEC对象

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
		Latch rising pos1	%ID4	DINT		探针1的锁存上升沿采集位置
		Latch rising pos2	%ID5	DINT		探针2的锁存上升沿采集位置
		Latch falling pos1	%ID6	DINT		探针1的锁存下降沿采集位置
		Latch falling pos2	%ID7	DINT		探针2的锁存下降沿采集位置
		Compare Status	%IB32	BYTE		比较器状态
		Busy	%IX32.0	BOOL		
		Done	%IX32.1	BOOL		
		Controlword	%QB4	BYTE		计数器控制字
		Start	%QX4.0	BOOL		
		Reset	%QX4.1	BOOL		
		Preset Value	%QD2	DINT		计数器预设置
		Preset command	%QB12	BYTE		预设功能控制
		PresetActiveSoft	%QX12.0	BOOL		
		PresetReset	%QX12.1	BOOL		
		PresetDoneClean	%QX12.2	BOOL		
		Touch probe command1	%QB13	BYTE		探针1功能控制字
		Touch probe command2	%QB14	BYTE		探针2功能控制字
		Compare command	%QB15	BYTE		比较功能控制字
		Start	%QX15.0	BOOL		
		Enable	%QX15.1	BOOL		
		Reset	%QX15.2	BOOL		
		Compare Value	%QD4	DINT		比较值
		Compare Output Keep Value	%QW10	UINT		比较结果硬件输出保持值
		Compare Output Unit	%QB22	BYTE		比较输出保持单位：0~100us,1~脉冲

4.1.7 外部中断说明

中断是指 CPU 正在执行一项任务 A，突然停止任务 A 去执行任务执行完任务 B，再回来继续执行任务 A 的过程。

实现外部中断功能步骤如下：

步骤1 设置输入端口为中断功能，选择上升沿、下降沿或上升下降沿触发，设置滤波参数，参数默认为 0.5μs（2\*0.25μs）。

图 4-8 外部中断输入配置

滤波参数（0.25微秒）

X0: 2 X1: 2 X2: 20000 X3: 20000

X4: 2 X5: 2 X6: 2 X7: 2

中断输入

☐ X0 ☐ X1 ☐ X2 ☐ X3

☐ X4 ☒ X5 ☒ X6 ☐ X7

步骤2 选择中断任务

在 Invtmatic Studio 任务中选择任务类型为 External，输入端口 X0~X7 对应中断 in(x)Interrupt，X 范围为 0~7，通过中断任务触发对应的 POU\_1 程序。

图 4-9 外部中断触发任务

Task x

配置

优先级 (0..31): 1

类型: 外部的

外部事件: in0Interrupt, in1Interrupt, in2Interrupt, in3Interrupt, in4Interrupt, in5Interrupt, in6Interrupt, in7Interrupt

看门狗: ☐ 使能

时间 (如 t=200ms):

灵敏度: 1

增加调用 移除调用 更改调用 上移 下移 打开POU

POU: POU\_1

注释



## 4.2 I/O 本地扩展模块配置

PLC 本体可直接扩展 Flex 系列模块，也可通过 EtherCAT 总线与 FK1100 耦合器连接来使用扩展模块，单个 TM700 系列 PLC 或单个 FK1100 耦合器最大可扩展 32 个 Flex 系列模块（加电源中继）。TP6000 系列 PLC 本体无法直接连接 Flex 系列模块，但是 PLC 可通过 EtherCAT 总线与 FK1100 耦合器连接，将扩展模块挂在耦合器上。Flex 系列模块型号及功能描述详见表 4-3。

表 4-3 扩展模块类型

模块名称	Flex 系列模块	说明
耦合器	FK1100	EtherCAT 总线适配器，可扩展 32 个总线节点
数字量输入模块	FL1001	16 路数字量输入模块，源型/漏型输入
	FL1002	32 路数字量输入模块，源型/漏型输入
数字量输出模块	FL2002	16 路数字量输出模块，源型输出
	FL2003	32 路数字量输出模块，源型输出
	FL2102	16 路数字量输出模块，漏型输出
	FL2103	32 路数字量输出模块，漏型输出
	FL2201	8 路继电器输出
模拟量输入模块	FL3003	4 路模拟量输入模块电流/电压输入
温度模块	FL3103	4 路温度模块，热电阻类型
	FL3203	4 路温度模块，热电偶类型
模拟量输出模块	FL4003	4 路模拟量输出模块，电流/电压输出
混合输入模块	FL5005	16 路数字量输入+16 路数字量输出(源型输出)模块
	FL5105	16 路数字量输入+16 路数字量输出(漏型输出)模块

### 4.2.1 扩展模块组态

TM700 系列、TP6000 系列 PLC 按照以下两种方式进行模块组态，手动添加和自动扫描，优先采用自动扫描方式。

#### 方式一：手动添加

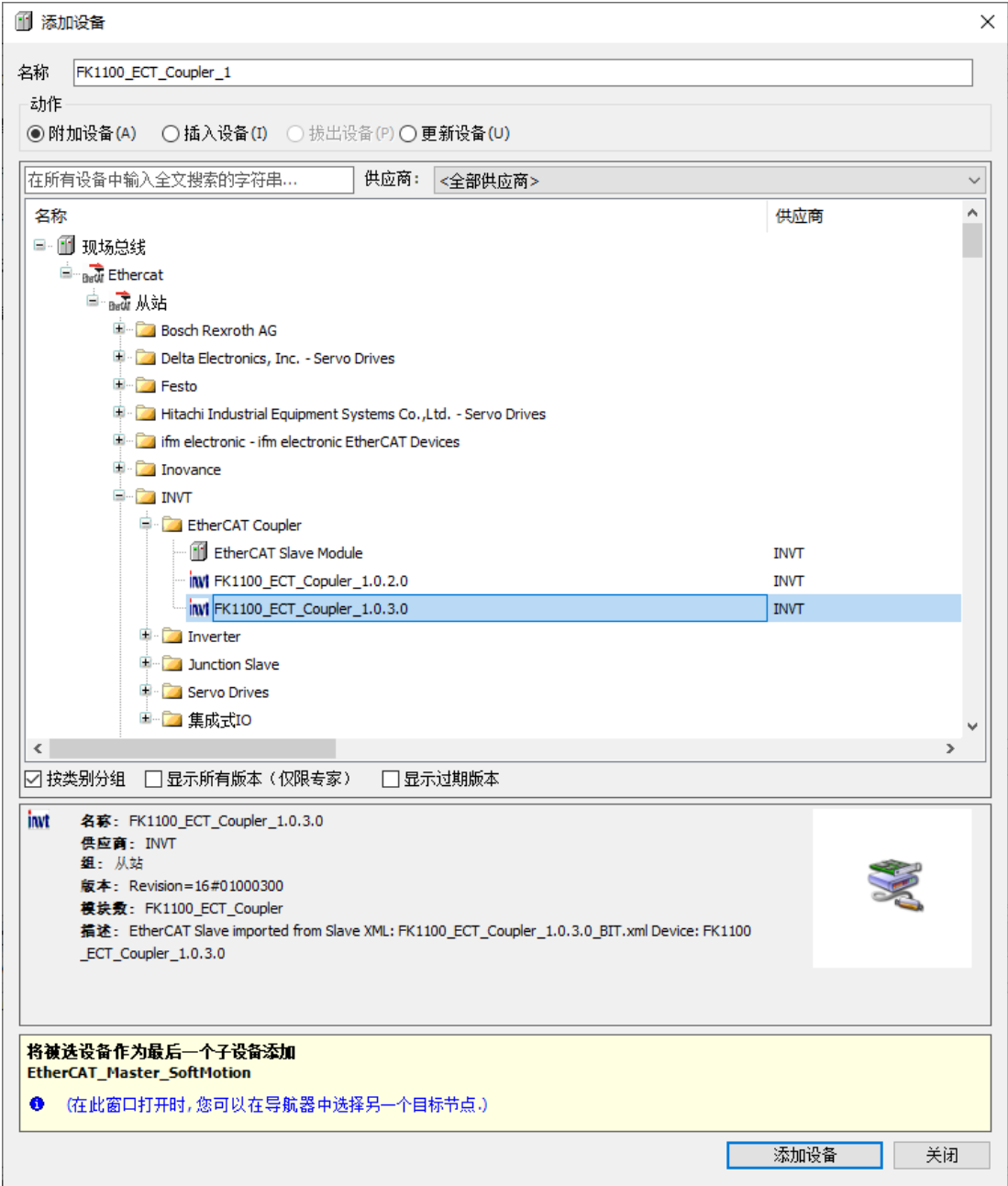
打开 Invtmatic Studio 编程软件，新建工程，选择编程语言（具体操作方法详见 2 入门指引）后，添加所需的扩展模块。

#### 1、添加 FK1100 耦合器使用模块

步骤1 在左侧设备树中右键点击 **Device**，点击添加 **EtherCAT\_Master\_SoftMotion** 主站设备后，再右键点击 **EtherCAT\_Master\_SoftMotion**，选择**添加设备**，添加耦合器到设备树中。



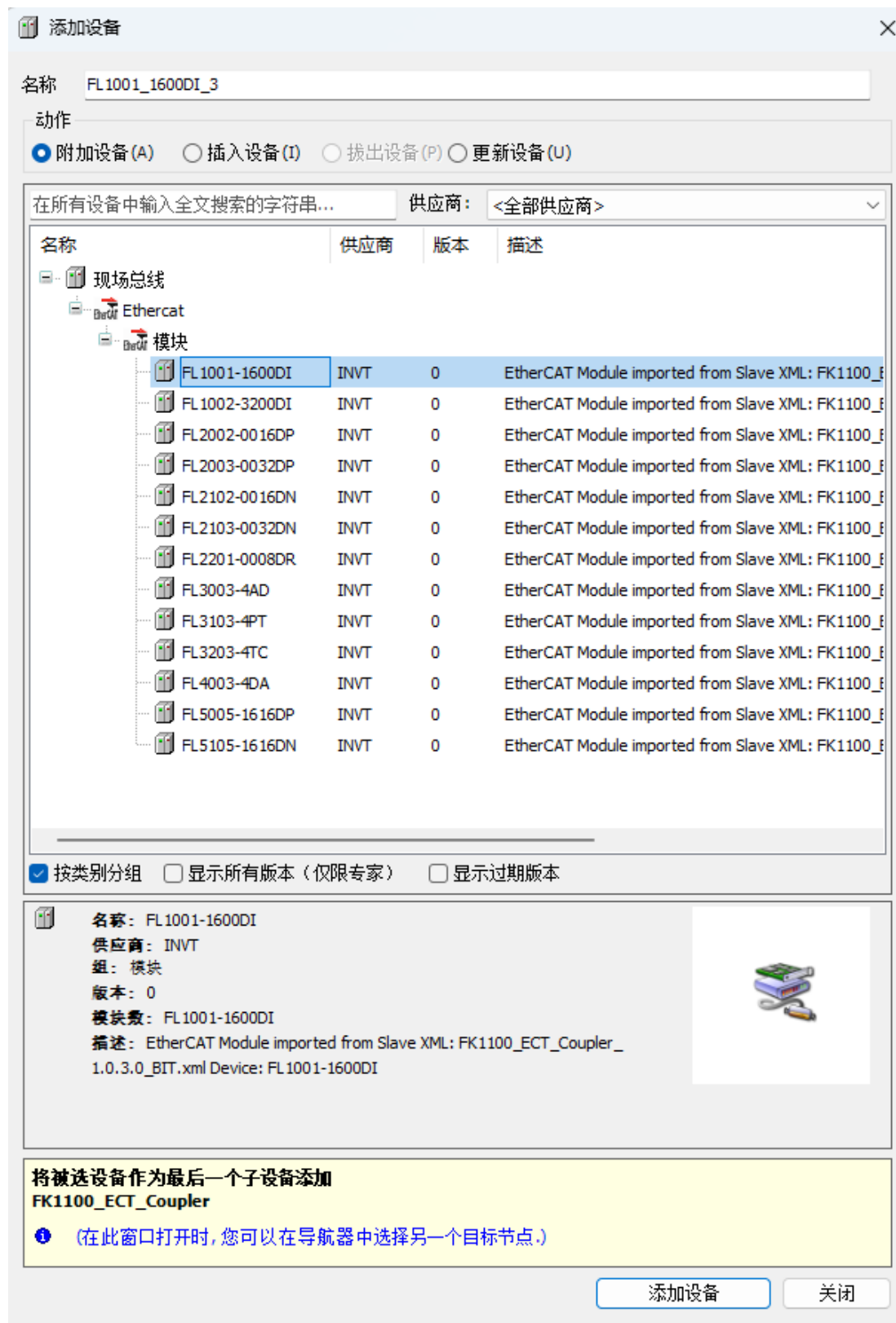
图 4-10 手动添加耦合器





步骤2 选择设备树中的 **FK100\_ECT\_Coupler**，点击**添加设备**后，在模块设备栏里，选择所需的模块，点击**添加设备**至设备树中。

图 4-11 手动添加扩展模块

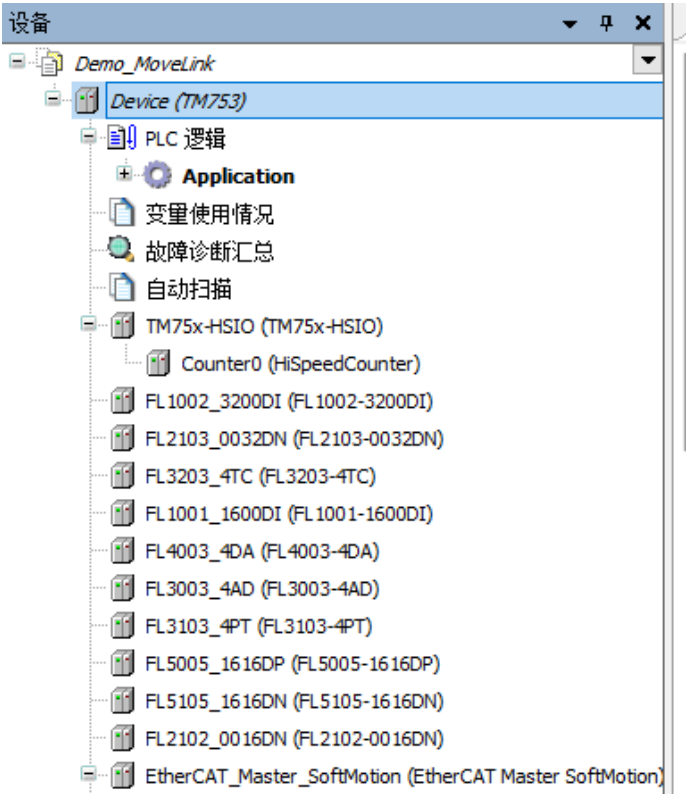
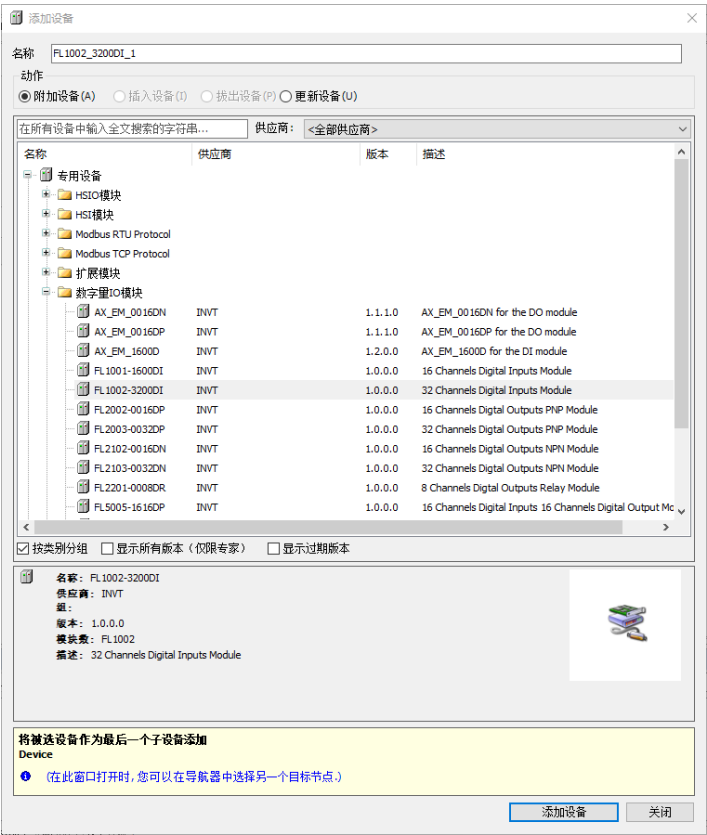




2、 在控制器本体添加模块（TM700 系列 PLC）

右键点击设备树中的 **Device**，选择**添加设备**，在专用设备栏界面，选择所需的模块，点击添加设备至设备树中。

图 4-12 PLC 本体添加扩展模块





方式二：扫描添加(推荐方式)

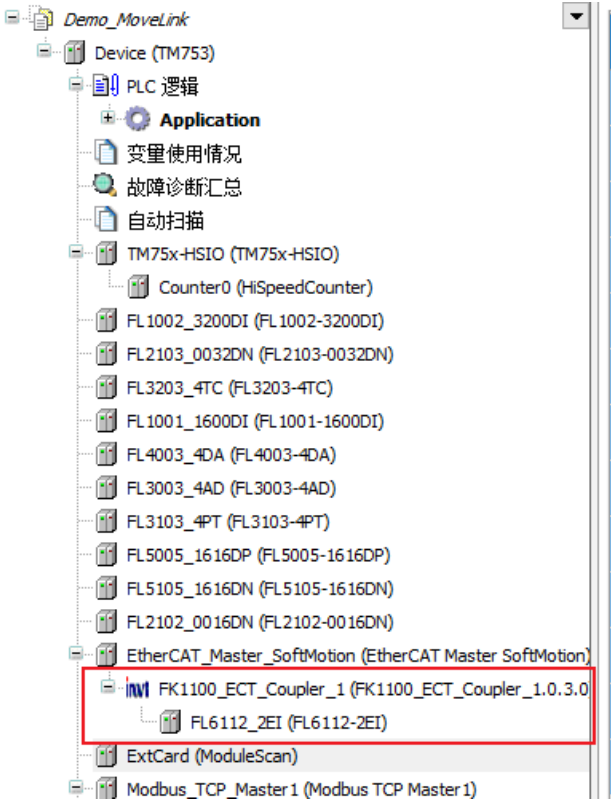
1、 扫描 EtherCAT 总线配置

- 步骤1 打开 Invtmatic Studio 编程软件，新建工程，选择编程语言。
- 步骤2 连接 PLC，添加 EtherCAT 主站，具体操作方法详见 2 入门指引。
- 步骤3 在 **EtherCAT\_Master\_SoftMotion** 上点击鼠标右键，选择扫描设备，连接的设备会自动显示，点击**复制所有设备到工程中**。

图 4-13 扫描耦合器和扩展模块



图 4-14 扫描后添加的耦合器和扩展模块





2、 扫描本地扩展模块

- 步骤1 打开 Invtmatic Studio 编程软件，新建工程，选择编程语言。
- 步骤2 连接 PLC，扫描本地扩展模块需要登录 PLC（可不运行），具体操作方法详见 2 入门指引。
- 步骤3 双击设备树中的**自动扫描**，点击**扫描**，列表中显示设备中连接的扩展模块，点击**添加到组态**，即可完成组态。

图 4-15 PLC 本体背板扫描

自动扫描 x									
扫描 清空 添加到组态									
序号	扫描到的设备型号	扫描到的设备版本号	扫描到的设备名称	位置	组态中存在的设备型号	组态中存在的设备名称	组态中是否存在	是否添加到组态	结果
1	36947	1.0.0.0	FL1002_32000I	1	36947	FL1002_32000I	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	36963	1.0.0.0	FL2103_0032DN	2	36963	FL2103_0032DN	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	37033	1.0.0.0	FL3203_4TC	3	37033	FL3203_4TC	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	36946	1.0.0.0	FL1001_16000I	4	36946	FL1001_16000I	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	37001	1.0.0.0	FL4003_4DA	5	37001	FL4003_4DA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	36993	1.0.0.0	FL3003_4AD	6	36993	FL3003_4AD	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	37025	1.0.0.0	FL3103_4PT	7	37025	FL3103_4PT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	36978	1.0.0.0	FL5005_1616DP	8	36962	FL2102_0016DN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	36982	1.0.0.0	FL5105_1616DN	9			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	0			10			<input type="checkbox"/>	<input type="checkbox"/>	

4.2.2 数字量输入模块

数字量输入模块主要配置滤波参数，有 I/O 映射、状态和信息页面，一般情况下只需要在 I/O 映射界面映射 I/O 变量以获取数字量输入值，以 32 点数字量输入模块为例。

- 滤波参数：默认为 10ms（1000\*10μs）。

图 4-16 输入模块配置

3200DI 配置

PCI-BusIEC对象

CPU DI32参数

CPU DI32I/O映射

状态

信息

滤波参数

滤波参数 1 1000 (10us)

滤波参数 2 1000 (10us)

滤波参数 3 1000 (10us)

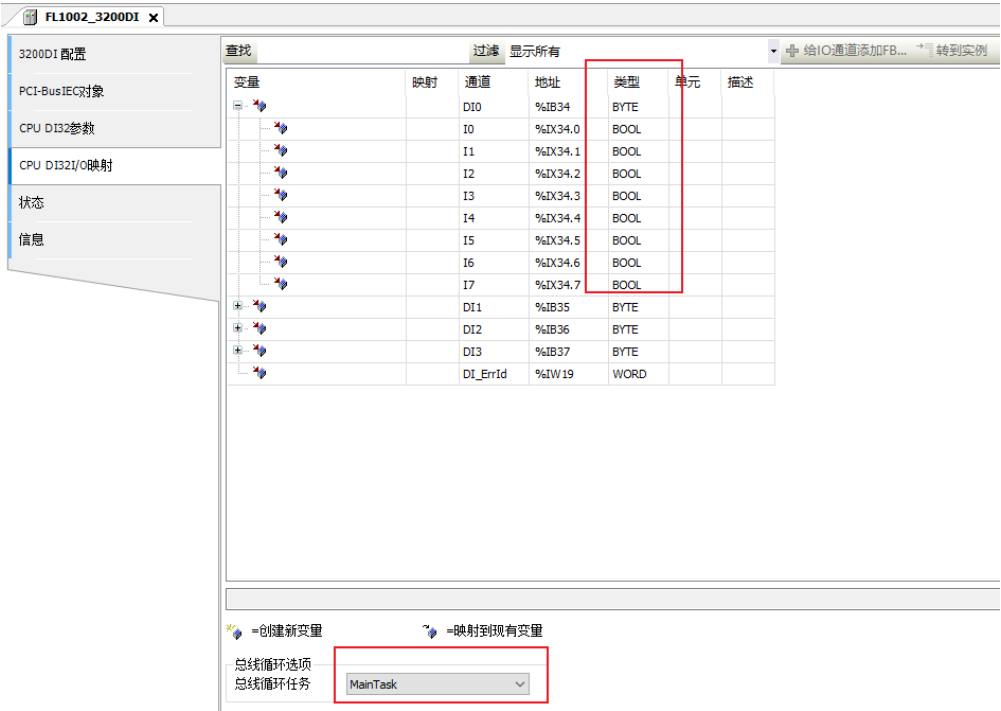
滤波参数 4 1000 (10us)

Version V-.-



- DI32 I/O 映射，可以通过 BYTE 或 BOOL 型获取输入状态。

图 4-17 输入模块变量映射



4.2.3 数字量输出模块

数字量出模块主要设置停止后输出状态选择，有 I/O 映射、状态和信息页面，一般情况下只需要在 I/O 映射界面映射 I/O 变量以获取数字量输出值，以 16 点数字量输出模块为例。

图 4-18 输出模块配置





DN16 I/O 映射，可以通过 BYTE 或 BOOL 型控制输出状态。

图 4-19 输出模块变量映射

0016DN参数设置

错误诊断

CPU D016参数

CPU D016I/O映射

状态

信息

变量	映射	通道	地址	类型	单元	描述
		DO0	%QB40	BYTE		
		Q0	%QX40.0	BOOL		
		Q1	%QX40.1	BOOL		
		Q2	%QX40.2	BOOL		
		Q3	%QX40.3	BOOL		
		Q4	%QX40.4	BOOL		
		Q5	%QX40.5	BOOL		
		Q6	%QX40.6	BOOL		
		Q7	%QX40.7	BOOL		
		DO1	%QB41	BYTE		
		DO_ErrId	%IW66	WORD		

4.2.4 模拟量输入模块

■ 一般配置

模拟量输入模块包括 4 个通道，每个通道都有参数配置和 I/O 映射寄存器（16 位）设置，下面仅对每个模块的一个通道进行说明。

图 4-20 模拟量输入模块配置

4AD参数设置

错误诊断

CPU AI4参数

CPU AI4I/O映射

状态

信息

通道 0

☒ 通道使能

☐ 断线检测

☐ 超限检测

☐ 超量程检测

☐ 增强型滤波

转换模式 0~5V (0~20000)

滤波参数 8 (1~255)

通道 1

☒ 通道使能

☐ 断线检测

☐ 超限检测

☐ 超量程检测

☐ 增强型滤波

转换模式 0~10V (0~20000)

滤波参数 8 (1~255)

通道 2

☒ 通道使能

☐ 断线检测

☐ 超限检测

☐ 超量程检测

☐ 增强型滤波

转换模式 4~20mA (0~20000)

滤波参数 8 (1~255)

通道 3

☒ 通道使能

☐ 断线检测

☐ 超限检测

☐ 超量程检测

☐ 增强型滤波

转换模式 -10~10V (-20000~20000)

滤波参数 8 (1~255)

版本 V-.-



- 通道使能：是否激活此通道，只有激活了，此通道才能使用。
- 滤波参数：模拟量输入通道滤波，范围 1~255。
- 转换模式：设置模拟量输入转换类型，此设置决定了通道输入转换类型和转换值的范围，转换类型详见表 4-4。
- 断线检测：设置模拟量输入通道是否检测断线，因不能区分模拟量输入 0 值和断线，所以在所有转换模式范围中，包含 0 值输入的，不能激活断线标志。
- 超限检测：设置模拟量输入通道是否检测超限。
- 超量程检测：设置模拟量输入通道是否超量程检测。
- 增强型滤波：设置模拟量输入通道是否使用增强型滤波。

■ 4AD I/O 映射

4AD 为 4 通道模拟量输入，每个通道模拟输入对应一个 16 位的整数值，模拟量值和数字量值关系见模拟量输入一般配置。可以在此界面对每个 16 位整数值映射一个变量，以获取一个输入通道模拟量对应的数字量值，详见表 4-4。

图 4-21 模拟量输入模块变量映射

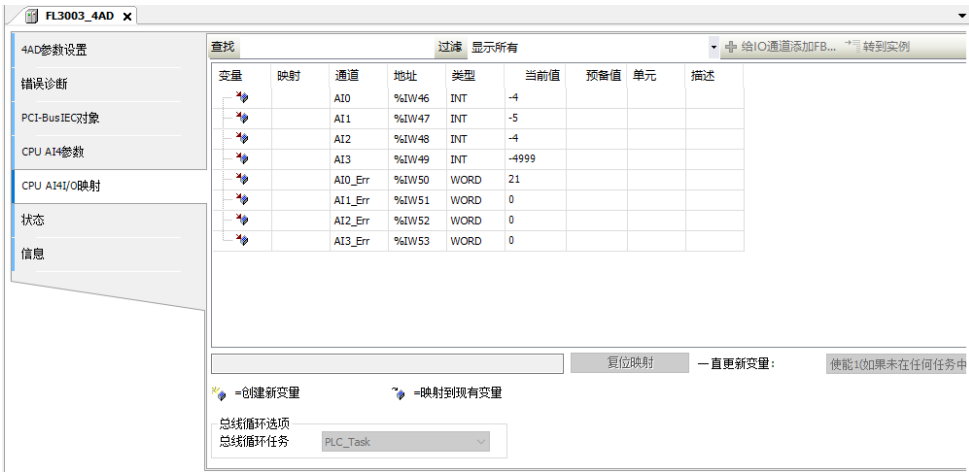


图 4-22 模块量输入模块 4AD 信息界面





表 4-4 映射与实际输入模拟量值对应关系

类型	输入额定范围	额定对应数字量
模拟电压输入	-10V~10V	-20000~+20000
	0V~10V	0~20000
	-5V~+5V	-20000~+20000
	0V~5V	0~20000
模拟电流输入	-20mA~20mA	-20000~20000
	0mA~20mA	0~20000
	4mA~20mA	0~20000

4.2.5 模拟量输出模块

■ 一般配置

模拟量输出模块包括 4 个通道，每个通道都包括参数配置和 I/O 映射寄存器设置（16 位），下面仅对每个模块的一个通道进行说明。

图 4-23 模拟量输出变量映射



- 转换模式：设置模拟量输出转换类型，此设置决定了通道输出转换类型和转换值的范围，转换类型详见表 4-5。
- 保持输出：模块运行停止后，输出一直保持上次输出值。
- 输出清零：模块运行停止后，输出一直为 0。



- 输出预设值：模块运行停止后，输出一直为预设值。预设值范围和当前转换模式有关，详见表 4-5。

表 4-5 映射与实际输入模拟量值对应关系

类型	输入额定范围	额定对应数字量
模拟电压输出	-10~10V	-20000~+20000
	0~10V	0~20000
	-5~+5V	-20000~+20000
	0~5V	0~20000
模拟电流输出	4~20mA	0~20000
	0~20mA	0~20000
	-20~20mA	-20000~+20000

■ 4DA I/O 映射

4DA 为 4 通道模拟量输出，每个通道模拟输出对应一个 16 位的整数值，模拟量值和数字量值关系详见表 4-5。可以在此界面对每个 16 位整数值映射一个变量，把此变量值输出到当前通道，然后经模拟量输出模块转换为模拟量值输出。

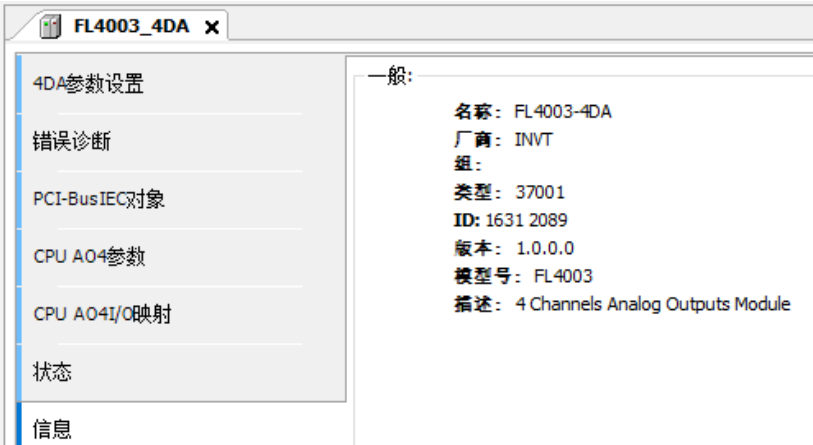
图 4-24 模拟量输出模块变量映射



■ 信息

显示 4DA 模块设备基本信息：名称、厂商、组、类型、ID、版本、型号、描述。另外，登陆后信息页面会读取 4DA 模块单板软件版本和逻辑软件版本并显示出来，单板软件版本为 4DA 嵌入式软件版本，逻辑软件版本为 4DA 模块内 FPGA 软件版本。

图 4-25 模拟量输出模块 4DA 信息





4.2.6 温度模块

温度模块包含 4TC（4 通道温度检测模块，支持热电偶）和 4PT（4 通道温度检测模块，支持热电阻）。

4.2.6.1 温度模块 4TC

温度模块 4TC（4 通道温度检测模块，支持热电偶），有对应的一般配置和通道配置。一般配置用于配置温度模块的单位类型和采样周期，通道配置分别配置每个通道传感器类型、滤波时间、超限、温度偏移等参数。

■ 通道配置

不同类型模块支持不同的通道，4TC 支持 4 通道。由于每个通道的配置参数基本相同，此处仅对一个通道进行说明。4TC 一个通道配置如下图。

图 4-26 温度模块 4TC 配置



- 温度单位：温度模块输入使用的单位，支持摄氏度或华氏度。
- 通道使能：是否激活此通道，只有激活了，此通道才能使用。
- 传感器类型：4TC 传感器类型、传感器规格详见表 4-6，默认值使用 K 传感器。
- 滤波参数：温度模块此通道使用的滤波，范围 1~255，默认 8。
- 超量程检测：使用此通道超量程检测功能，如果不在温度上下限之间，会报超限故障，范围详见表 4-6。
- 偏移值(°C)：设置温度模块偏移补偿值，范围-204.8~204.7。



## ■ I/O 映射

不同类型的温度模块包含不同个数的通道和 IO 映射。下图为 4TC 的 I/O 映射界面，每个通道参数值为温度值。

图 4-27 温度模块 4TC 变量映射

<b>查找</b>		<b>过滤 显示所有</b>					
变量	映射	通道	地址	类型	单元	描述	
		Temp0	%ID 17	REAL			
		Temp1	%ID 18	REAL			
		Temp2	%ID 19	REAL			
		Temp3	%ID 20	REAL			
		Temp0_ErrId	%IW 42	WORD			
		Temp1_ErrId	%IW 43	WORD			
		Temp2_ErrId	%IW 44	WORD			
		Temp3_ErrId	%IW 45	WORD			

#### 4.2.6.2 温度模块 4PT

温度模块 4PT（4 通道温度检测模块，支持热电阻），有对应的一般配置和通道配置。一般配置用于配置温度模块的单位类型和采样周期，通道配置分别配置每个通道传感器类型、滤波时间、超限、温度偏移等参数。

## ■ 通道配置

4PT 支持 4 通道，由于每个通道的配置参数基本相同，仅对一个通道进行说明，4PT 一个通道配置如下图。

图 4-28 温度模块 4PT 配置

The screenshot displays the 'FL3103\_4PT' software window. On the left is a sidebar with navigation options: '4PT参数设置' (4PT Parameter Settings), '错误诊断' (Error Diagnosis), 'PCI-BusIEC对象' (PCI-BusIEC Object), 'CPU PT4参数' (CPU PT4 Parameters), 'CPU PT4I/C映射' (CPU PT4I/C Mapping), '状态' (Status), and '信息' (Information). The main area is titled '温度单位设置' (Temperature Unit Settings) and contains two identical channel configuration sections, '通道 0' (Channel 0) and '通道 1' (Channel 1).

At the top of the main area, there are radio buttons for temperature units: '摄氏度 (°C)' (Celsius) and '华氏度 (°F)' (Fahrenheit). Below this, each channel section includes:

- A checkbox for '通道使能' (Channel Enable).
- A '滤波参数' (Filter Parameter) input field set to '8' with a range of '(1~255)'.
- A '偏移值 (°C)' (Offset Value) input field set to '0.0' with a range of '-204.8~204.7'.
- A '转换模式' (Conversion Mode) dropdown menu. For Channel 0, the menu is open, showing options: 'PT100', 'PT100' (highlighted), 'PT500', 'PT1000', and 'CU100'.
- A '热制模式' (Heat Mode) section with radio buttons for '2线制' (2-wire), '3线制' (3-wire), and '4线制' (4-wire).
- A checkbox for '超量程检测' (Over-range Detection).
- A '超量程检测温度上下限' (Over-range Detection Temperature Range) section with input fields for '温度上限 (°C)' (Temperature Upper Limit) set to '850.0' and '温度下限 (°C)' (Temperature Lower Limit) set to '-200.0', both with a range of '-200~850'.

- 温度单位：温度模块输入使用的单位，支持摄氏度或华氏度。
- 通道使能：是否激活此通道，只有激活了，此通道才能使用。
- 热电阻线制模式：2 线制、3 线制或 4 线制。




- 传感器类型：4PT 传感器类型、传感器规格详见表 4-6，默认值使用 PT100 传感器。
- 滤波参数：温度模块此通道使用的滤波，范围 1~255，默认 8。
- 超量程检测：使用此通道超量程检测功能，如果不在温度上下限之间，会报超限故障，范围详见表 4-6。
- 偏移值(℃)：设置温度模块偏移补偿值，范围-204.8~204.7。

■ I/O 映射

不同类型的温度模块包含不同个数的通道和 IO 映射。下图为 4PT 的 I/O 映射界面，每个通道参数值为温度值。

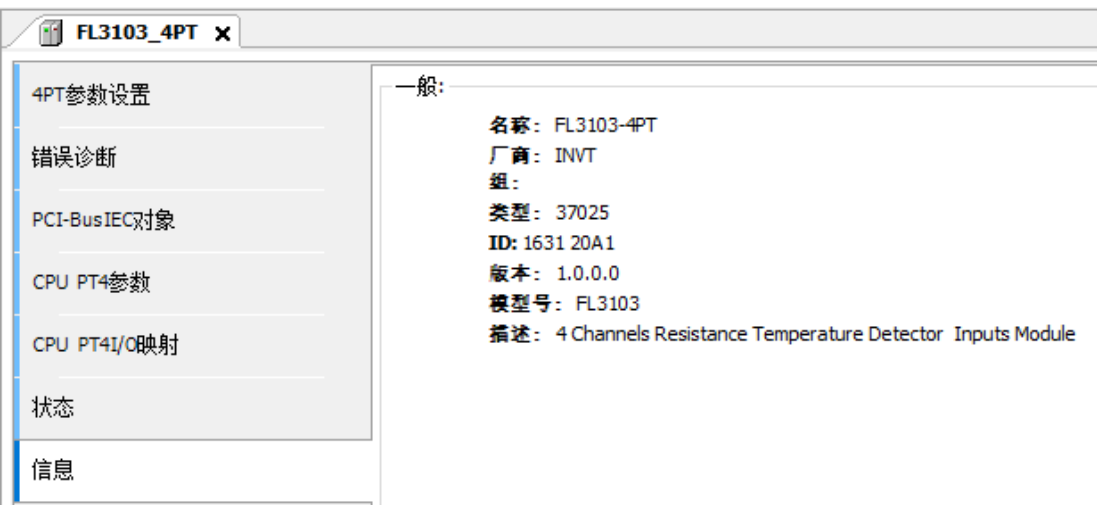
图 4-29 温度模块 4PT 变量映射



变量	映射	通道	地址	类型	单元	描述
		Temp0	%ID27	REAL		
		Temp1	%ID28	REAL		
		Temp2	%ID29	REAL		
		Temp3	%ID30	REAL		
		Temp0_ErrId	%IW62	WORD		
		Temp1_ErrId	%IW63	WORD		
		Temp2_ErrId	%IW64	WORD		
		Temp3_ErrId	%IW65	WORD		

■ 信息

图 4-30 温度模块 4PT 信息



一般:
名称: FL3103-4PT
厂商: INVT
组:
类型: 37025
ID: 1631 20A1
版本: 1.0.0.0
型号: FL3103
描述: 4 Channels Resistance Temperature Detector Inputs Module

表 4-6 支持的传感器类型和测量范围

项目	传感器类型	摄氏度温度范围	华氏度温度范围
热电阻类型	PT100	-200.0~850℃	-328.0~1562.0°F
	PT500	-200.0~850℃	-328.0~1562.0°F
	PT1000	-200.0~850℃	-328.0~1562.0°F
	CU100	-50.0~150℃	-58.0~302.0°F
热电偶类型	B	200.0~1800℃	392.0~3272.0°F
	E	-270.0~1000℃	-454.0~1832.0°F
	N	-200.0~1300℃	-328.0~2372.0°F



项目	传感器类型	摄氏度温度范围	华氏度温度范围
	J	-210.0~1200°C	-346.0~2192.0°F
	K	-270.0~1370°C	-454.0~2498.0°F
	R	-50.0~1765°C	-58.0~3209.0°F
	S	-50.0~1765°C	-58.0~3209.0°F
	T	-270.0~400°C	-454.0~752.0°F

4.3 各模块优先级设置（推荐值）

若创建的工程中包含多个功能模块，创建多个任务，任务优先级设置如下图所示，任务优先级推荐使用值详见表 4-7。

图 4-31 任务工程优先级设置参考示例

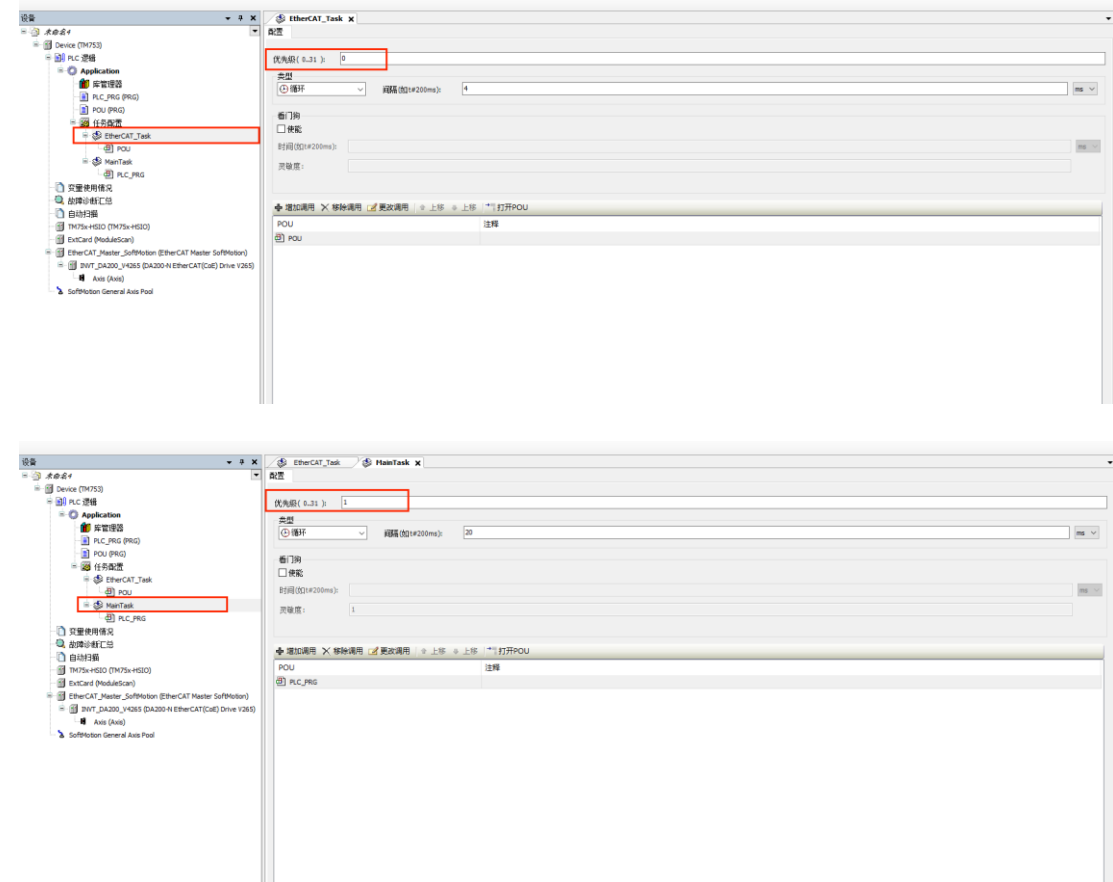


表 4-7 优先级设置

功能模块	推荐优先级
PlcCfg 模块	31
Modbus TCP	15~30
Modbus RTU	15~30
高速 I/O（默认配置到 MainTask）	1
模拟量输入输出	1~15
温度模块	1~15
EtherCAT	0



# 5 网络配置

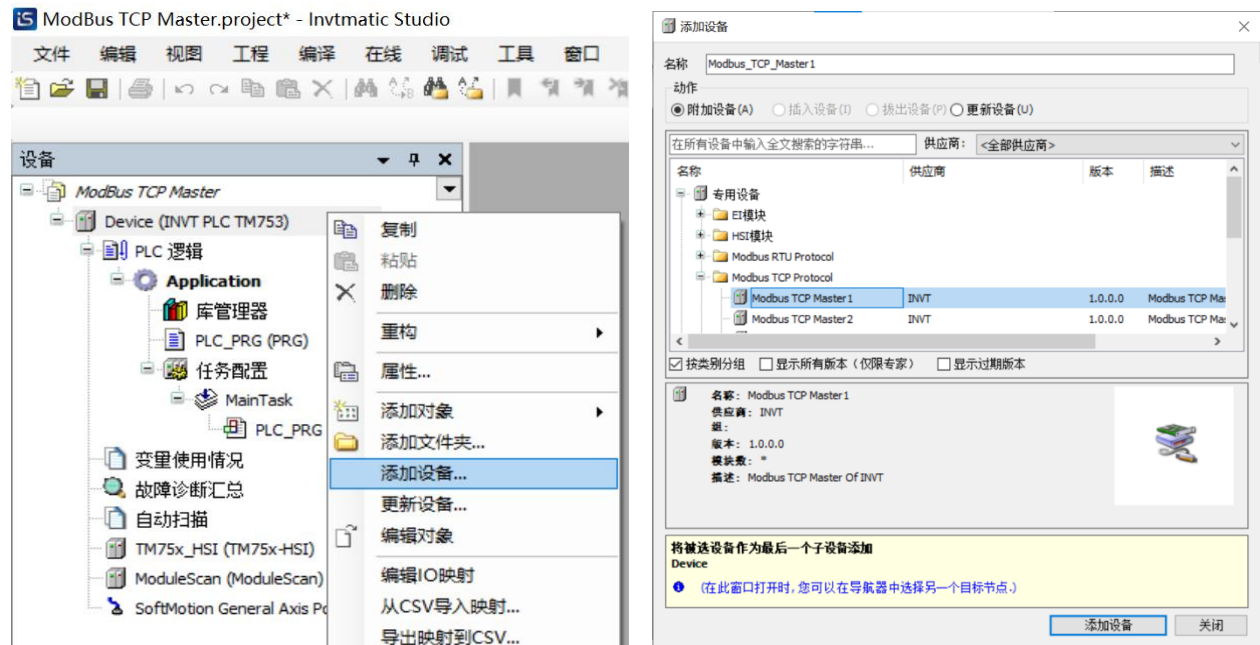
PLC 控制器网络配置主要包含以下网络：Modbus TCP、Modbus RTU、EtherCAT 和 CANOpen。

## 5.1 Modbus TCP

### 5.1.1 Modbus TCP 主站配置

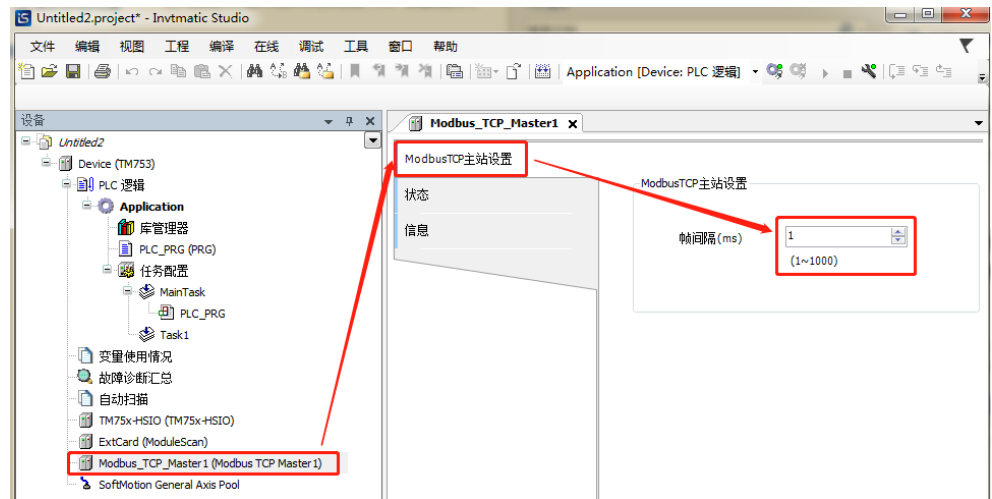
PLC 做 Modbus TCP 主站时，在左侧设备树中右键点击 **Device**，选择**添加设备**，在弹出的窗口中选择**专用设备>Modbus TCP Protocol>Modbus TCP Master 1**，再点击右下角的**添加设备**。

图 5-1 添加 Modbus TCP 主站



双击设备树中的主站设备 **Modbus\_TCP\_Master 1**，打开 Modbus 主站配置窗口，如下图所示。帧间隔(ms)指主站接收上一个响应数据帧到下一个请求数据帧之间等待的时间间隔，这个参数可用于调节数据交换率。

图 5-2 Modbus TCP 主站设置





ModbusTCP 可以访问的变量数量定义如下：

- 读线圈（0x01）：线圈数量 1~2000（0x7D0）
- 读离散线圈（0x02）：线圈数量 1~2000（0x7D0）
- 读保持寄存器（0x03）：寄存器数量 1~125（0x7D）
- 读输入寄存器（0x04）：寄存器数量 1~125（0x7D）
- 写单个线圈（0x05）
- 写单个寄存器（0x06）
- 写多个线圈（0x0F）线圈数量 1~1968（0x7B0）
- 写多个寄存器（0x10）寄存器数量 1~123（0x7B）

表 5-1 Modbus TCP 可以访问的变量数量定义

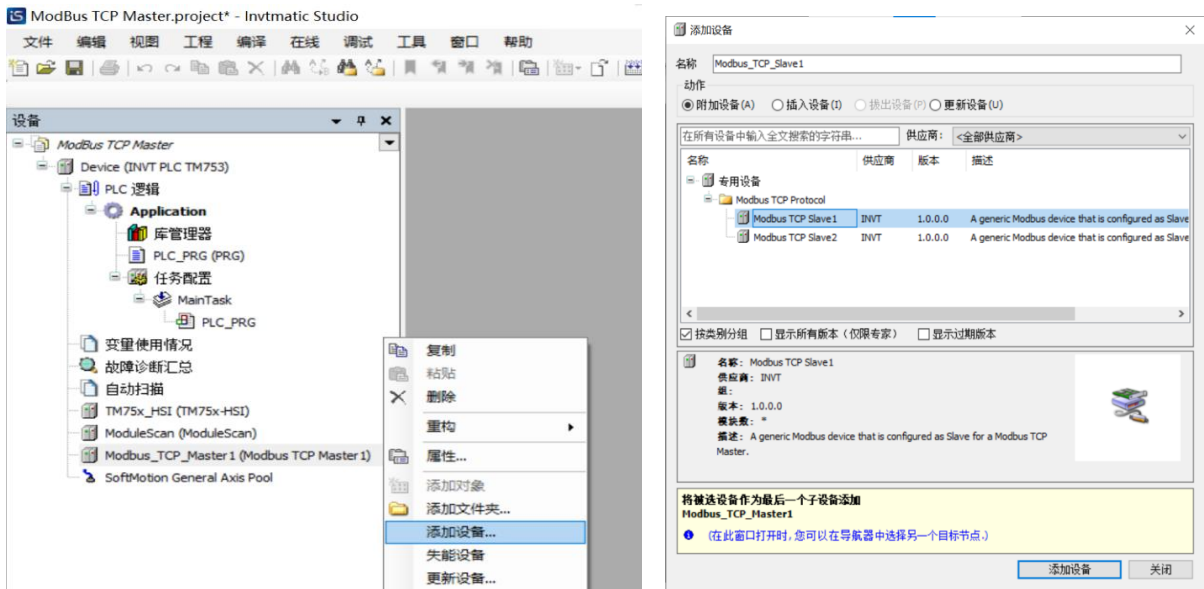
变量	数量
读线圈（0x01）	1~2000（0x7D0）
读离散线圈（0x02）	1~2000（0x7D0）
读保持寄存器（0x03）	1~125（0x7D）
读输入寄存器（0x04）	1~125（0x7D）
写单个线圈（0x05）	-
写单个寄存器（0x06）	-
写多个线圈（0x0F）	1~1968（0x7B0）
写多个寄存器（0x10）	1~123（0x7B）

5.1.2 Modbus TCP 主站通信配置

■ 添加 Modbus TCP 从站

在左侧设备树中右键点击 **Modbus\_TCP\_Master 1**，选择**添加设备**，在弹出窗口中选择 **Modbus TCP Slave 1**，点击右下角的**添加设备**。

图 5-3 添加 Modbus TCP 从站





■ 设置 Modbus TCP 从站

图 5-4 设置 Modbus TCP 从站

ModbusTCP从站设置

ModbusTCP通信设置

错误诊断

状态

信息

ModbusTCP从站设置

IP地址192 . 168 . 1 . 11

端口502

站点号1  
(0~255)

超时(ms)1000  
(2~65535)

从站使能变量 SM4001  
(0~7999)

表 5-2 Modbus TCP 从站设置参数说明

配置项	功能
IP 地址	主站连接 Modbus TCP 从站的 IP 地址
端口	主站连接 Modbus TCP 从站的 TCP 端口号
站点号	主站连接 Modbus TCP 从站的协议站点号
超时时间(ms)	主站发帧后，超过该时间从站未响应，主站报接收超时
从站使能变量	编程使能该变量后，主站开始向该从站发送通信帧

■ Modbus TCP 主站连接 Modbus TCP 从站通信设置

每一个通道代表一个独立的 Modbus TCP 请求，如下图所示，第三行定义了一个以 10ms 周期循环写单个寄存器（功能码 0x06）的操作，向偏移地址为(0x0014)20 的寄存器写长度为 2 字节的数据（1 个寄存器占 2 个字节）。

图 5-5 Modbus TCP 主站连接 Modbus TCP 从站通信设置

Modbus\_TCP\_Slave1 x

ModbusTCP从站设置

ModbusTCP通信设置

错误诊断

InternalI/O映射

状态

信息

	名称	功能码	使能类型	使能变量(SM)	循环时间(ms)	读取地址	读取长度	错误处理	写入地址	写入长度	重试次数	注释
0	Channel 00	(0x05)写单个线圈	循环执行		10				(0x000a) 10	1	1	
1	Channel 01	(0x01)读线圈	循环执行		10	(0x000a) 10	1	保持最后的值			1	
2	Channel 02	(0x06)写单个寄存器	循环执行		10				(0x0014) 20	1	1	
3	Channel 03	(0x03)读保持寄存器	循环执行		10	(0x0014) 20	1	保持最后的值			1	

添加删除编辑导出EXCEL导入

- 添加：点击添加后会出现一个为 Modbus TCP 从站添加新通道的对话框，再点击确定可新建一个通道。
- 编辑：在 Modbus TCP 从站通道列表表中选择一个通道，单击编辑将会出现一个对话框，通过修改其中的参数可改变通道的配置，点击确定可更新通道设置。
- 导出 EXCEL：点击导出 EXCEL，可以把通道参数批量导出到 EXCEL 表格中，再点击导入，可以把通道参数批量导入到设置中。



当添加或者编辑通道时，弹出以下对话框：

图 5-6 Modbus TCP 主站连接 Modbus TCP 从站通信设置对话框

Modbus通信设置

基础配置

名称

Channel 02

功能码

(0x06)写单个寄存器

使能类型

循环执行

循环时间(ms)

10

(1~65535)

使能变量(SM)

4101

(0~7999)

重试次数

1

注释

读取

地址(Hex)

0

(0~0xFFFF)

长度(WORD)

1

错误处理

保持最后的值

写入

地址(Hex)

20

长度(WORD)

1

确定

取消

表 5-3 Modbus TCP 通信设置参数说明

配置项		说明
名称		通道命名的字符串
功能码		读线圈状态（功能码 0x01） 读输入状态（功能码 0x02） 读保持寄存器（功能码 0x03） 读输入寄存器（功能码 0x04） 写单个线圈（功能码 0x05） 写单个寄存器（功能码 0x06） 写多个线圈（功能码 0x15） 写多个寄存器（功能码 0x16）
使能类型		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）：设置发 SM 元件（使能后完成该条通信后触发变量置位）
重试次数		本次发生通信故障未获得从站返回帧，则按重试次数进行重新发送
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
写寄存器	起始地址	写寄存器开始位置
	长度	写寄存器长度

■ Modbus TCP 从站 Internal I/O 映射

在 Modbus TCP 从站通信设置中添加主从站通信配置后，Internal I/O 映射中会自动分配每条配置的映射地址，如下图所示第三行的%IW98 表示将读取的一个线圈数值映射到%IW98 这个地址。另外，还可以通过输入助手或者直接输入示例变量路径，将程序中的自定义变量映射到 I/O 地址。



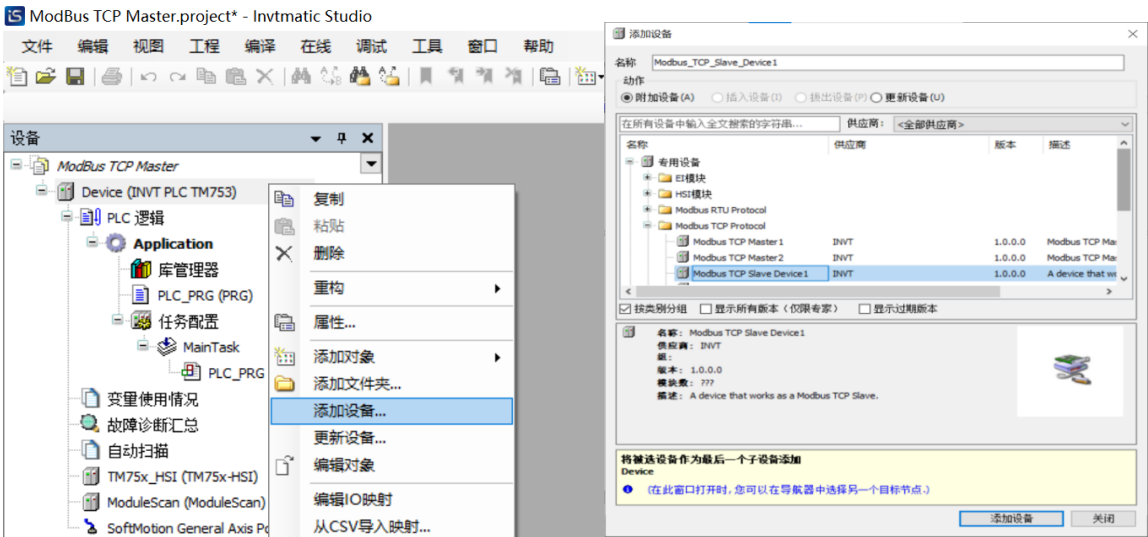
图 5-7 Modbus TCP 主站连接从站 Internal I/O 映射

Modbus_TCP_Slave1							
ModbusTCP从站设置		查找	过滤	显示所有			
ModbusTCP通信设置		变量	映射	通道	地址	类型	单元
错误诊断				Channel 00	%QW48	ARRAY [0..0] OF WORD	(0x05)写单个线圈
InternalI/O映射				Channel 02	%QW49	ARRAY [0..0] OF WORD	(0x06)写单个寄存器
状态				Channel 01	%IW98	ARRAY [0..0] OF WORD	(0x01)读线圈
信息				Channel 03	%IW99	ARRAY [0..0] OF WORD	(0x03)读保持寄存器

5.1.3 Modbus TCP 从站配置

PLC 做 Modbus TCP 从站时，在左侧设备树中右键点击 **Device**，选择**添加设备**，在弹出的窗口中选择**专用设备>Modbus TCP Slave Device1**，再点击右下角的**添加设备**。

图 5-8 Modbus TCP 添加从站



双击设备树中的从站设备，打开 Modbus TCP 从站配置窗口，如下图所示。

图 5-9 Modbus TCP 从站设置

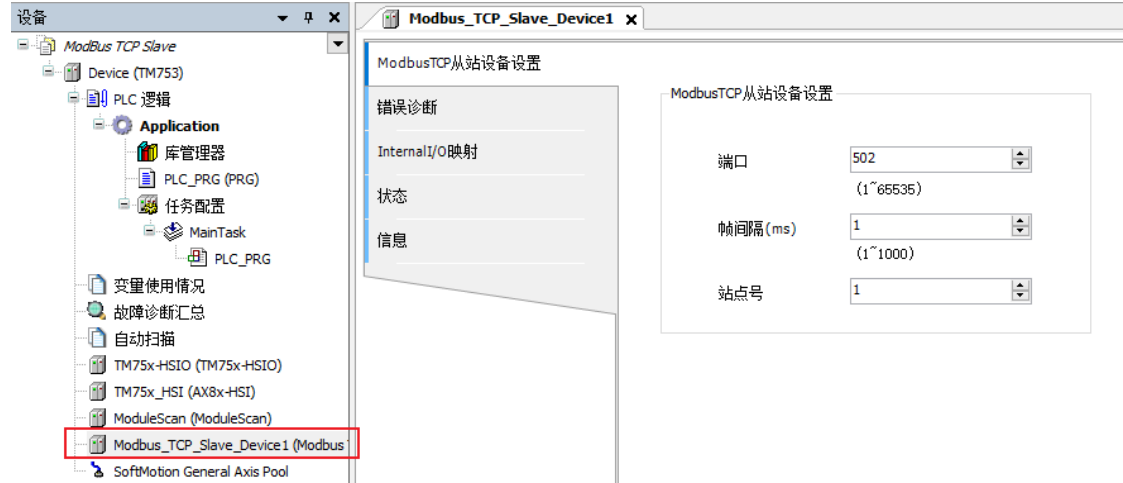




表 5-4 Modbus TCP 从站设置参数

配置项	功能
端口	Modbus TCP 从站的 TCP 端口号
帧间隔(ms)	Modbus TCP 从站接收通信帧后延迟时间发送回复帧
站点号	标识从站号

Modbus\_TCP\_Slave 从站定义了可供外部访问的存储区域，详细区域如下表所示。

表 5-5 Modbus\_TCP\_Slave 功能码

TCP 主站功能码	地址名称	范围	与标准 Modbus 地址偏移量
0x01	%QX	0-65535	无
0x05/0x15	%QX	0-65535	无
0x02	%IX	0-65535	无
0x04	%IW	0-65535	无
0x03	%MW	0-65535	无
0x06/0x16	%MW	0-65535	无

表 5-6 控制器位、字节、字、双字对应关系实例

%_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			

## 5.1.4 Modbus TCP 设备诊断

Modbus TCP 主站设备诊断可显示发生故障的从站和故障的通信配置项。

图 5-10 Modbus TCP 主站设备诊断界面

ModbusTCP从站设置				
ModbusTCP通信设置				
错误诊断				
InternalI/O映射				
状态				
信息				

序号	错误代码	错误名称	解决方案
1	Er00A0-0005	通讯超时，通讯时间超过了用户设定的通讯...	检查主从站通信连接是否正常

## 5.2 Modbus RTU

PLC 控制器支持两路 Modbus 串口通信，分别是 COM1\_RS485 和 COM2\_RS485，均支持标准的 Modbus RTU 协议，可独立配置为主站或从站，支持 2400、4800、9600、19200、38400、57600、115200 这 7 种波特率。

Modbus RTU 可以访问的变量数量定义如下：

- 读线圈（0x01）线圈数量 1~2000（0x7D0）
- 读离散线圈（0x02）线圈数量 1~2000（0x7D0）
- 读保持寄存器（0x03）寄存器数量 1~125（0x7D）



- 读输入寄存器 (0x04) 寄存器数量 1~125 (0x7D)
- 写单个线圈 (0x05)
- 写单个寄存器 (0x06)
- 写多个线圈 (0x0F) 线圈数量 1~1968 (0x7B0)
- 写多个寄存器 (0x10) 寄存器数量 1~123 (0x7B)

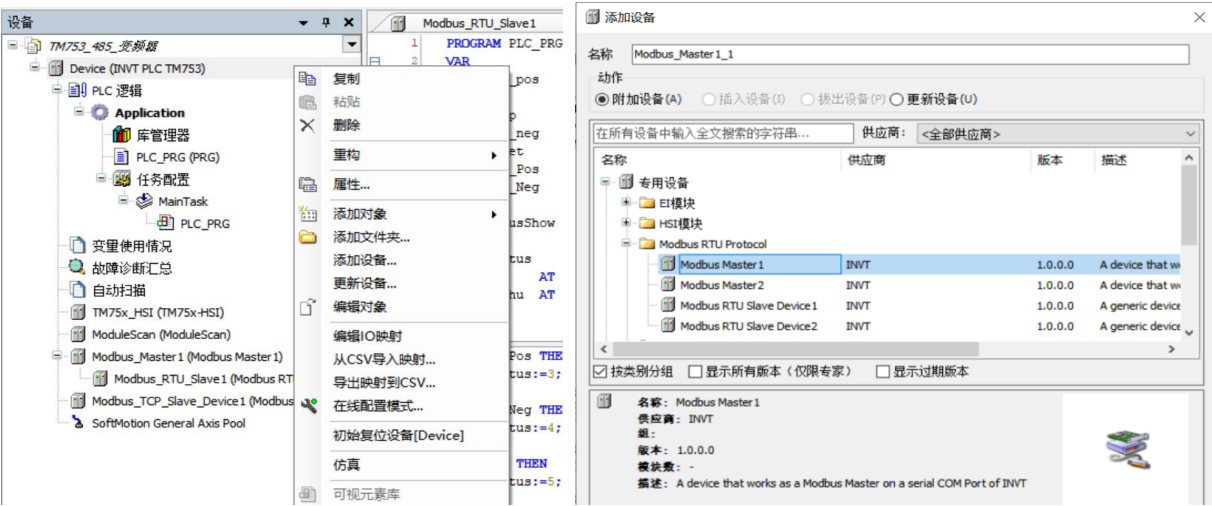
表 5-7 Modbus RTU 可以访问的变量数量定义

变量	数量
读线圈 (0x01)	1~2000 (0x7D0)
读离散线圈 (0x02)	1~2000 (0x7D0)
读保持寄存器 (0x03)	1~125 (0x7D)
读输入寄存器 (0x04)	1~125 (0x7D)
写单个线圈 (0x05)	-
写单个寄存器 (0x06)	-
写多个线圈 (0x0F)	1~1968 (0x7B0)
写多个寄存器 (0x10)	1~123 (0x7B)

5.2.1 Modbus RTU 主站配置

PLC 做 Modbus RTU 主站时，在左侧设备树中右键点击 **Device**，选择**添加设备**，在弹出的窗口中选择**专用设备>Modbus RTU Protocol>Modbus Master 1**，再点击右下角的**添加设备**。

图 5-11 添加 Modbus RTU 主站



打开 Modbus 主站配置窗口，如下图所示。


 **注意：**Modbus 主从站通信参数一致时，才能正常通信。



图 5-12 Modbus 主站设置



表 5-8 Modbus 主站设置参数说明

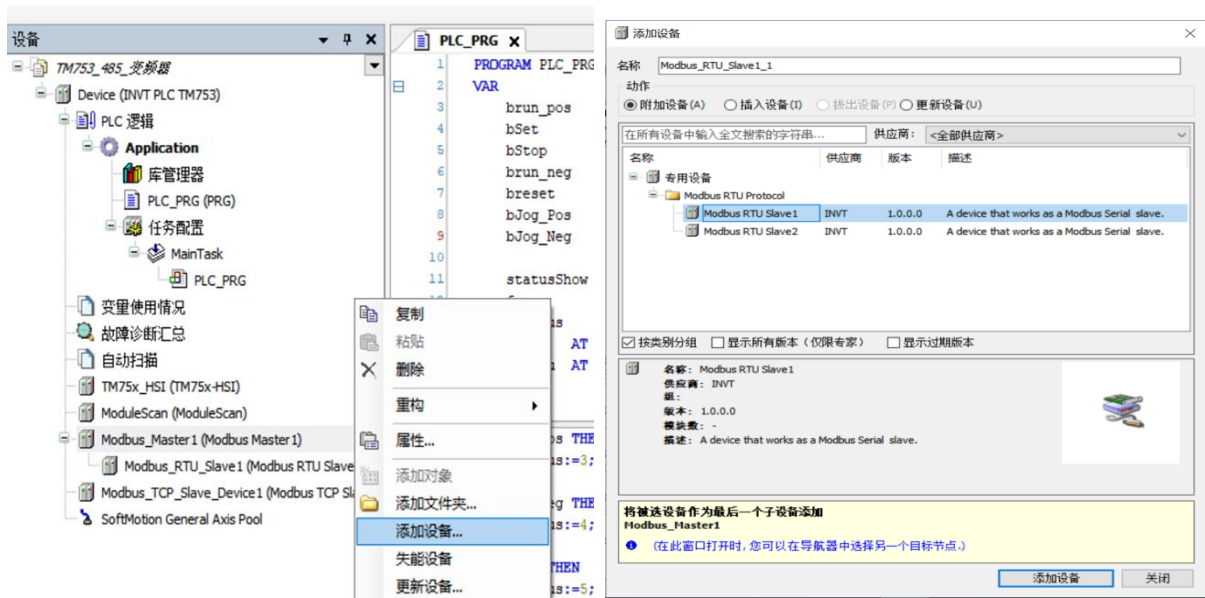
配置项	说明
波特率	通信时的速率:2400,4800,9600,19200,38400,57600,115200
校验位	通信帧的校验方式: None Parity,Odd Parity,Even Parity
数据位	通信帧包含的实际数据位
停止位	通信时标识单个包的最后位
帧间隔(ms)	主站接收上一个响应数据帧到下一个请求数据帧之间等待的时间间隔

5.2.2 Modbus RTU 主站通信配置

■ 添加 Modbus 从站

PLC 做 Modbus 主站时，在左侧设备树中右键点击 **Modbus Master 1**，选择**添加设备**，在弹出的窗口中选择**专用设备>Modbus RTU Protocol>Modbus RTU Slave1**，再点击右下角的**添加设备**。

图 5-13 端口作为主站时添加 Modbus RTU 从站





■ 设置 Modbus 从站

双击设备树中的从站设备打开 Modbus 从站设置窗口，如下图所示。

图 5-14 端口作为主站时的 Modbus RTU 从站设置

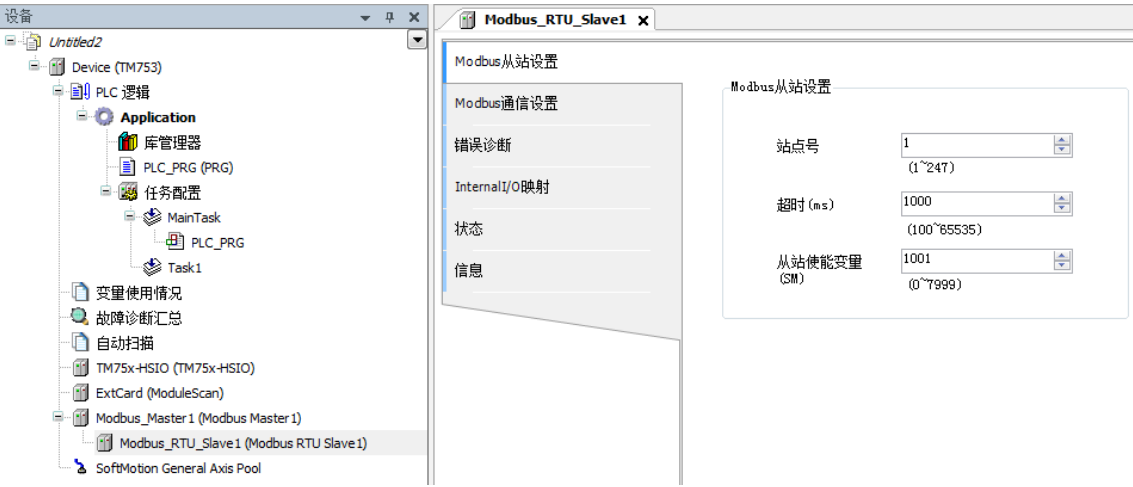


表 5-9 端口作为主站时的 Modbus RTU 从站设置参数说明

设置项	功能
站号	标识从站号，范围 1~247
超时时间(ms)	主站发帧后，超过该时间从站未响应，主站报接收超时
从站使能变量(SM)	编程使能该变量后，主站开始向该从站发送通信帧

■ Modbus 主站连接到 Modbus 从站通信设置

切换页面到 Modbus 从站通信设置窗口，添加 Modbus 从站通信配置，配置表最多支持 60 条配置。

每个通道代表一个独立的 Modbus 请求，其中第 1 行定义了一个循环执行写单个寄存器（功能码 0x06）的请求，向偏移量为 0x2000 的 1 个寄存器写一个字的数据。

图 5-15 端口做主站时的 Modbus 从站通信设置

名称	功能码	使能类型	使能变量(SM)	循环时间(ms)	读取地址	读取长度	错误处理	写入地址	写入长度	重试次数	注
0 Channel 00	(0x06)写单个寄存器	循环执行		1000				(0x2000) 8192	1	1	
1 Channel 01	(0x06)写单个寄存器	循环执行		1000				(0x2001) 8193	1	1	
2 Channel 02	(0x03)读保持寄存器	循环执行		1000	(0x2100) 8448	1	保持最后的值			1	
3 Channel 03	(0x03)读保持寄存器	循环执行		200	(0x3000) 12288	6	保持最后的值			1	

- 添加：点击**添加**后，会出现一个为 Modbus 从站添加新通道的对话框。点击**确定**按钮可新建一个通道。
- 编辑：在 Modbus 从站通道列表中选择 一个通道，点击**编辑**，将会出现一个对话框，通过修改其中的参数可改变通道的配置，点击**确定**可更新通道设置。
- 导出 EXCEL：点击**导出 EXCEL**，可以把通道参数批量导出到 EXCEL 表格中，再点击**导入**，可以把通道参数批量导入到设置中。



当添加或者编辑通道时，弹出以下对话框：

图 5-16 端口做主站时连接 Modbus 从站通信设置对话框

Modbus通信设置

基础配置

名称

Channel 00

功能码

(0x06)写单个寄存器

使能类型

循环执行

循环时间(ms)

1000

(1~65535)

使能变量(SM)

1101

(0~7999)

重试次数

1

注释

读取

地址(Hex)

0

(0~0xFFFF)

长度(WORD)

1

错误处理

保持最后的值

写入

地址(Hex)

8192

长度(WORD)

1

确定

取消

表 5-10 Modbus 通信设置参数说明

配置项		说明
名称		通道命名的字符串
功能码		读线圈状态（功能码 0x01） 读输入状态（功能码 0x02） 读保持寄存器（功能码 0x03） 读输入寄存器（功能码 0x04） 写单个线圈（功能码 0x05） 写单个寄存器（功能码 0x06） 写多个线圈（功能码 0x15） 写多个寄存器（功能码 0x16）
使能类型		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）：设置发 SM 元件，触发成功后，自动复位该元件
重试次数		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
写寄存器	起始地址	写寄存器开始位置
	长度	写寄存器长度



■ Modbus 从站 Internal I/O 映射

在 Modbus 从站通信设置中添加主从站通信配置后，Internal I/O 映射中会自动分配每条配置的映射地址，如下图第一行的%QW1 表示将读取的一个线圈数值映射到%QW1 这个地址。另外，还可以通过输入助手或者直接输入示例变量路径，将程序中的自定义变量映射到 I/O 地址。

图 5-17 端口做主站时的 Modbus 从站 Internal I/O 映射

Modbus_RTU_Slave1 X							
Modbus从站设置	查找	过滤	显示所有	给IO通道添加FB... 转到实例			
Modbus通信设置	变量	映射	通道	地址	类型	单元	描述
错误诊断	Application.PLC_PRG.betatus		Channel 00	%QW#1	ARRAY [0..0] OF WORD		(0x06)写单个寄存器
Internal I/O映射			Channel 00[0]	%QW#1	WORD		WRITE 16#2000(=8192)
状态	Application.PLC_PRG.fre		Channel 01	%QW#2	ARRAY [0..0] OF WORD		(0x06)写单个寄存器
信息			Channel 01[0]	%QW#2	WORD		WRITE 16#2001(=8193)
	Application.PLC_PRG.statusShow		Channel 02	%IWW#8	ARRAY [0..0] OF WORD		(0x03)读保持寄存器
			Channel 02[0]	%IWW#8	WORD		READ 16#2100(=8448)
	Application.PLC_PRG.canshu		Channel 03	%IWW#9	ARRAY [0..5] OF WORD		(0x03)读保持寄存器
			Channel 03[0]	%IWW#9	WORD		READ 16#3000(=12288)
			Channel 03[1]	%IWW#9	WORD		READ 16#3001(=12289)
			Channel 03[2]	%IWW#11	WORD		READ 16#3002(=12290)
			Channel 03[3]	%IWW#12	WORD		READ 16#3003(=12291)
			Channel 03[4]	%IWW#13	WORD		READ 16#3004(=12292)
			Channel 03[5]	%IWW#14	WORD		READ 16#3005(=12293)
			Diagnose Info	%IWW15	ARRAY [0..3] OF WORD		Channels 错误信息

5.2.3 Modbus RTU 从站配置

PLC 做 Modbus 从站时，在左侧设备树中右键点击 **Device**，选择**添加设备**，在弹出的窗口中选择**专用设备>Modbus RTU Protocol>Modbus RTU Slave Device1**，再点击右下角的**添加设备**。

双击设备树中的从站设备，打开 Modbus 从站设置窗口，如下图所示。

Modbus 从站设置参数中，串口配置部分与 Modbus 主站串口设置含义相同，Modbus 从站配置站点号是指本设备站点号；帧间隔为收到主站发送的通信帧后，延时回复主站的具体时间段。

图 5-18 Modbus RTU 作为从站时的配置

Modbus从站设置

错误诊断

状态

信息

串口设置

波特率19200

校验位EVEN

数据位8

停止位1


Modbus从站设置

站点号1

1~247

帧间隔(ms)3

3~200

 **注意：**Modbus 主从站通信参数配置一致时，才能正常通信。

Modbus\_RTU\_Slave 从站定义了可供外部访问的存储区域，其详细区域如下表所示。

表 5-11 Modbus\_RTU\_Slave 功能码

RTU 主站功能码	地址名称	范围	与标准 Modbus 地址偏移量
0x01	%QX	0-65535	无
0x05/0x15	%QX		
0x02	%IX		
0x04	%IW		

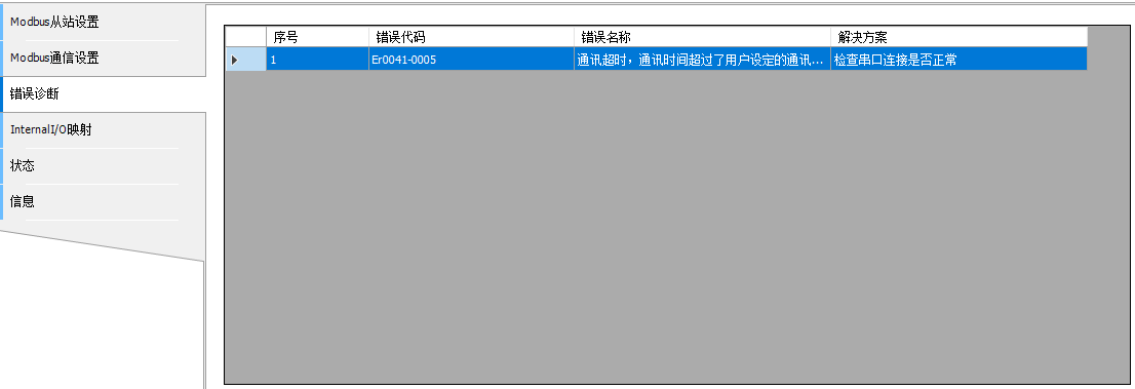


RTU 主站功能码	地址名称	范围	与标准 Modbus 地址偏移量
0x03	%MW		
0x06/0x16	%MW		

5.2.4 Modbus RTU 设备诊断

Modbus 主站设备诊断可显示发生故障的从站和故障的通信配置项。

图 5-19 Modbus 主站设备诊断



5.2.5 Modbus RTU 常见故障

Modbus 主站连接从站时发生的主要故障如下：

- Modbus 主站与 Modbus 从站配置不一致，导致主站与从站通信无法建立。
- Modbus 主站访问 Modbus 从站非法地址，返回错误应答。
- Modbus 主站操作 Modbus 从站写寄存器，但是 Modbus 从站该寄存器只支持读不支持写操作，Modbus 主站会收到 Modbus 从站返回的出错应答。

错误响应帧格式：从机地址+（命令码+0x80）+错误码+CRC 校验。

 **注意：**本错误帧适合所有的操作命令帧。

表 5-12 Modbus 错误响应帧说明

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	命令码+0x80	1 个字节	错误命令码
3	错误码	1 个字节	1~4

5.3 EtherCAT 主站

EtherCAT（Ethernet for Control Automation Technology）是一个架构开放、以太网为基础的现场总线系统。与其他现场总线相比，EtherCAT 具有性能好、设备同步精度高、拓扑结构灵活、应用容易、成本低等特点。当前，越来越多的设备使用 EtherCAT 总线进行通讯。



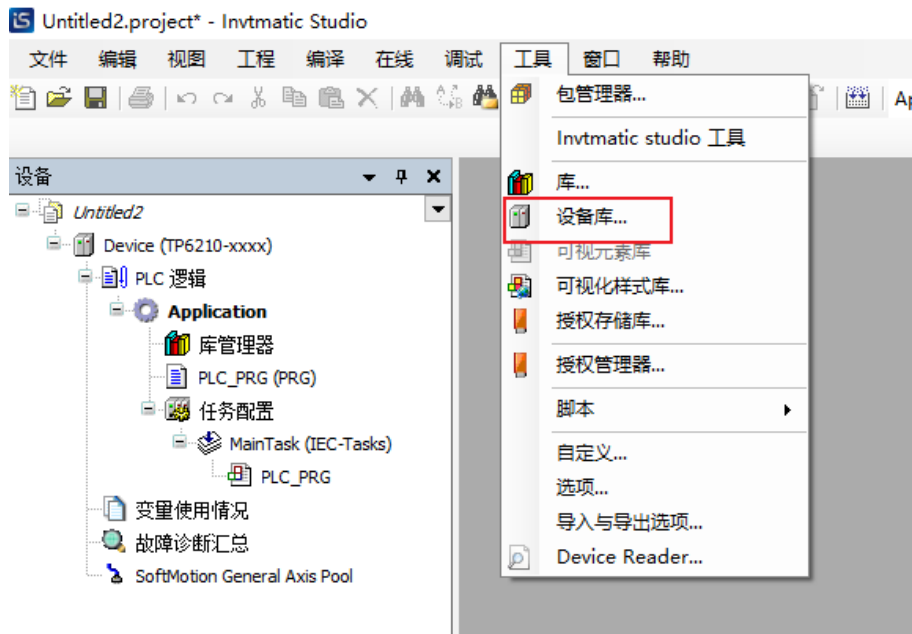
5.3.1 EtherCAT 主站配置

5.3.1.1 添加设备描述文件

Invtmatic Studio 中已经预添加了 INVT 的 EtherCAT 从站设备，用户如果使用 INVT 的产品，可直接跳过该步骤。用户在使用软件中尚未添加设备描述文件的 EtherCAT 从站时，需要自行将对应的描述文件添加到软件中，操作步骤如下：

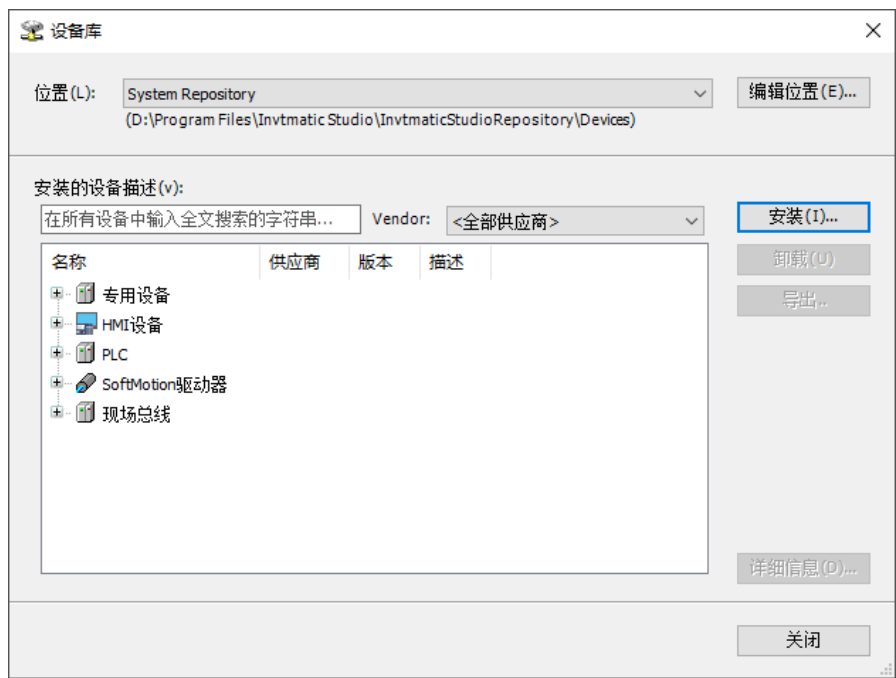
步骤1 打开 Invtmatic Studio，选择菜单栏中的**工具>设备库**。

图 5-20 选择设备库



步骤2 在弹出的窗口中点击**安装**。

图 5-21 安装设备库

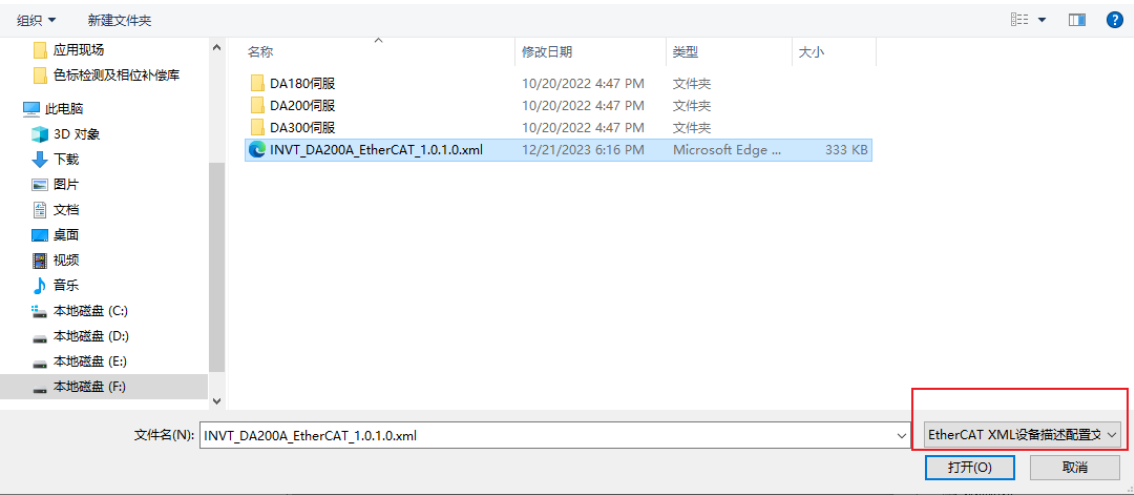




步骤3 在弹出窗口中找到存放设备描述文件的路径。

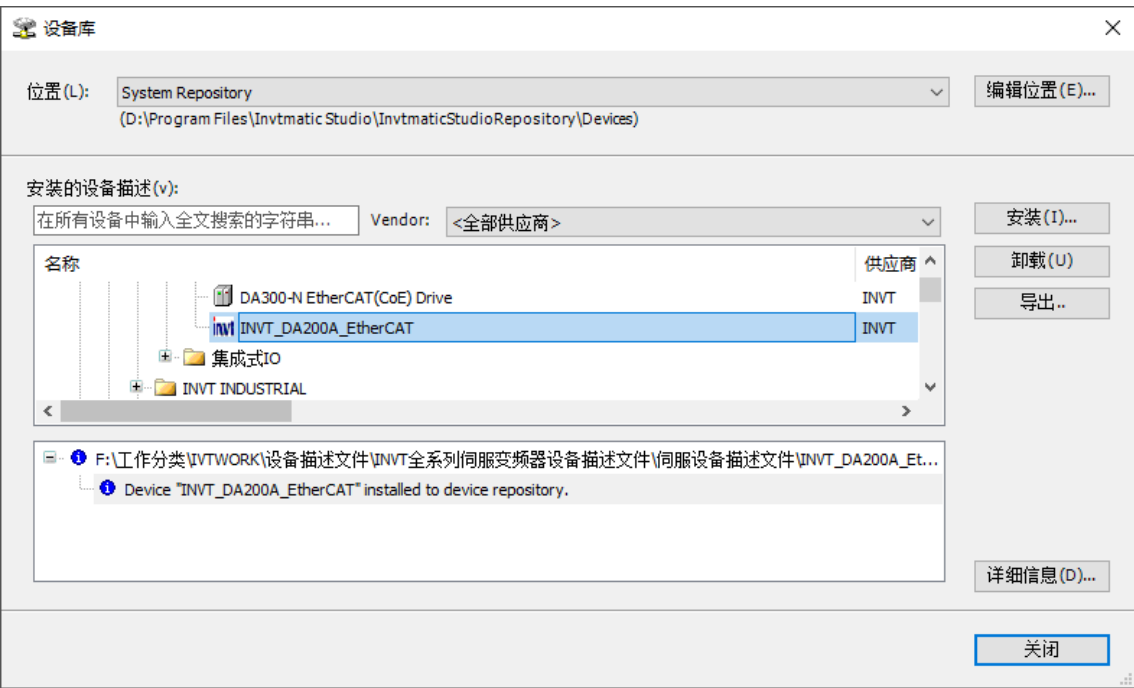
**注意：**此处将文件类型选择为 EtherCAT XML 设备描述配置文件，如果选择为其他文件类型，会无法正常选取 XML 文件，找到对应的 XML 后，点击**打开**。

图 5-22 选择设备描述文件的路径和类型



步骤4 打开文件之后，软件会自动导入设备，此时可观察信息输出框，确认安装完成后，点击**关闭**即可。

图 5-23 将设备描述文件导入设备



### 5.3.1.2 从站组态

在 Invtmatic Studio 软件中可以通过“扫描添加”和“手动添加”两种方式来实现 EtherCAT 从站的组态。操作步骤详见 2.3 程序编写与调试实例。通过扫描添加 EtherCAT 从站时，首先要保证工程中使用到从站设备硬件正确连接在 PLC 上，只有在能够正常通讯时，PLC 才能扫描到 EtherCAT 从站设备。

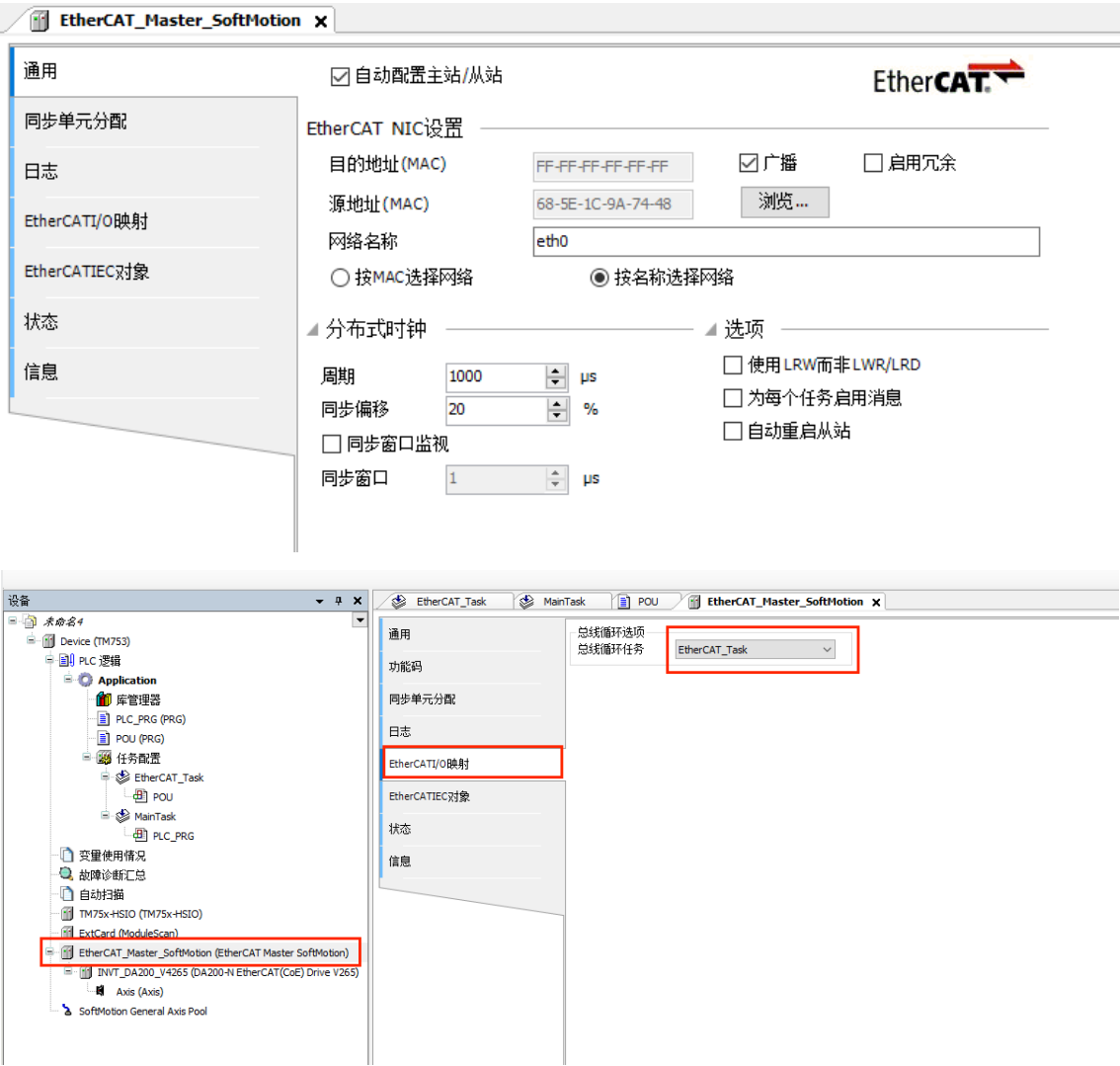


5.3.1.3 主站设置

EtherCAT 主站设置是设置 EtherCAT Master 的参数。如图 5-24 所示，要设置分布式时钟的周期和同步偏移。

- 周期：EtherCAT 数据帧发送间隔，该时间需要与 EtherCAT 任务的循环时间相同。
- 同步偏移：EtherCAT 任务相对于从站的 Sync0 中断的相对偏移的百分比，该值默认为 20%，无特殊情况请勿修改。

图 5-24 设置主站的周期和同步偏移



在 EtherCAT 通讯的 DC 模式中，从站在解析主站传输过来的数据后，还需要在 DC 中断中处理数据。由于主站数据传输到每个从站的时间长短不一，所以为了保证数据的同步，在数据帧和同步中断之间必须有足够的时间给从站接收和处理数据，这段时间即为偏移时间。

**注意：**在使用时，需要加上主站名，如主站名为 EtherCAT\_Master\_SoftMotion，则在重启主站时，需要触发的变量为 EtherCAT\_Master\_SoftMotion.xRestart。EtherCAT 主站常用参数如表 5-13 所示。

表 5-13 EtherCAT 主站常用参数

变量名	功能
xRestart	上升沿触发，触发后重启
EtherCAT	主站



变量名	功能
xConfigFinished	主站配置完成
xDistributedClockInSync	EtherCAT 分布式时钟同步
xError	EtherCAT 主站存在错误

5.3.2 EtherCAT 从站配置

5.3.2.1 常规设置

■ EtherCAT 地址

EtherCAT 从站的组态地址，是指从站设备在 Invtmatic Studio 设备树中的顺序地址，从 1001 开始，顺序递增，如图 5-25 所示。用户可以在从站的设置界面中，找到该从站的 EtherCAT 地址。在调用读写 sdo 功能块时，会使用到该地址，如图 5-26 所示。

图 5-25 EtherCAT 从站地址

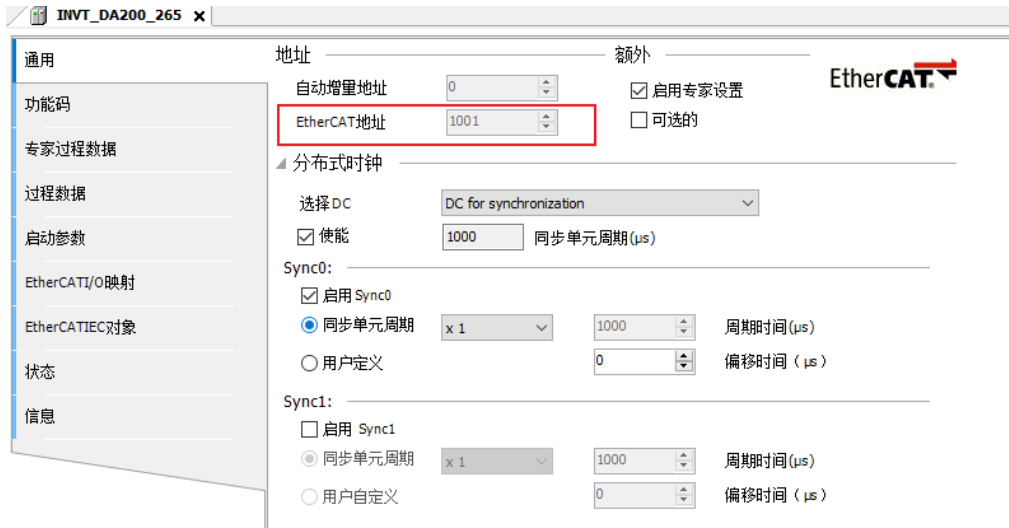
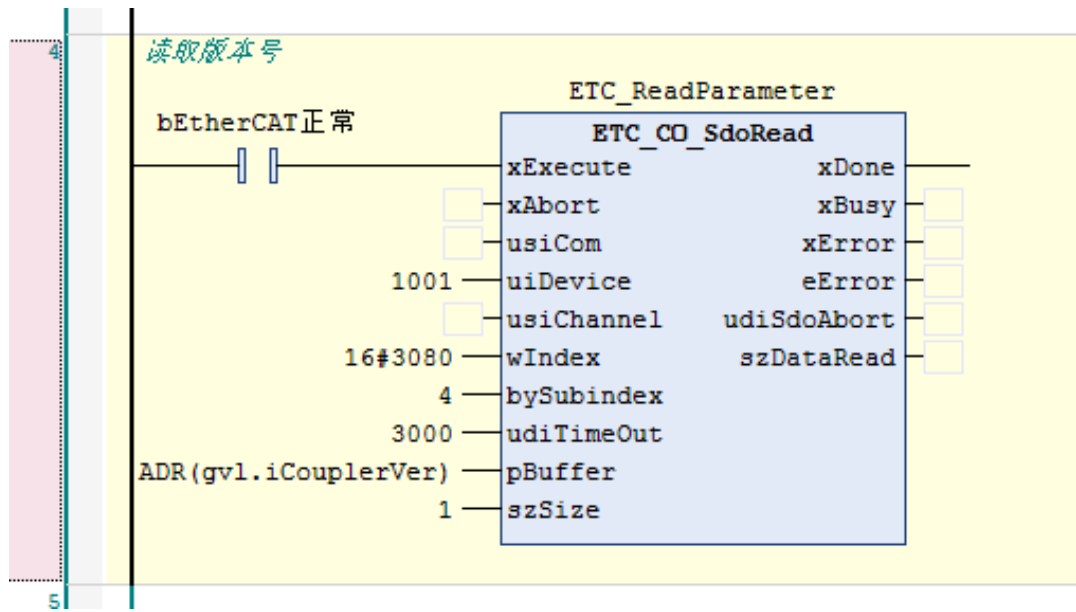


图 5-26 SdoRead 功能块



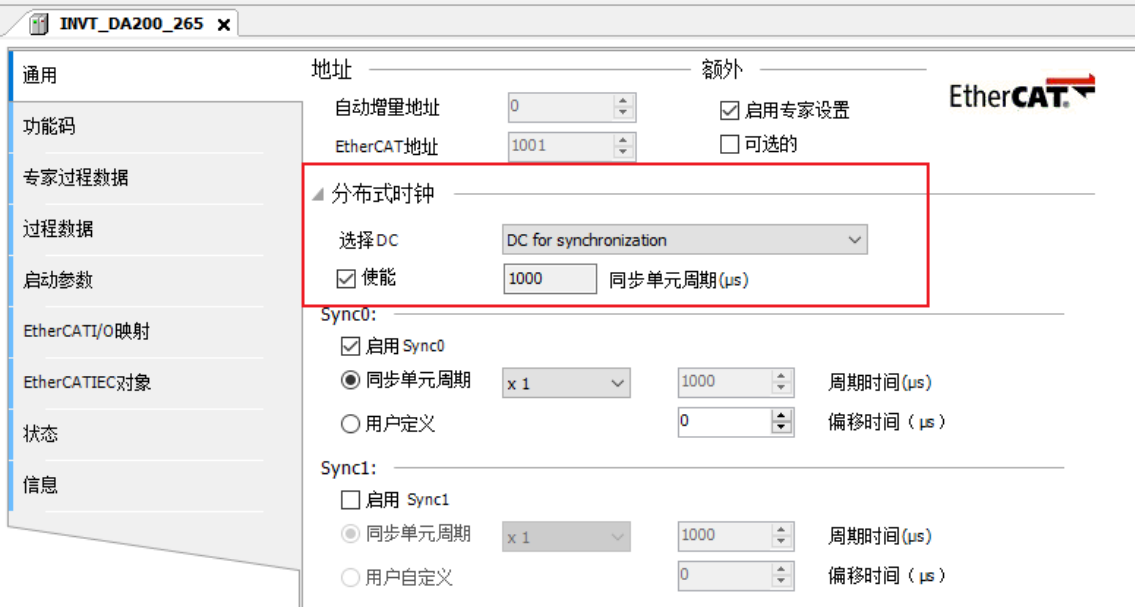


■ 分布式时钟

该选项用于设置从站的同步运行模式。同步模式有三种：自由运行模式（FreeRun）、同步于输入输出事件的运行模式（SM-Synchron）和同步于分布时钟的运行模式（DC-Synchron）。

选择的从站不同，同步模式支持的选项也会有不同。用户一般无需修改该选项，只要保证 DC-Synchron 运行模式下的同步单元周期与 EtherCAT 任务周期保持一致即可，如下图所示。

图 5-27 分布式时钟的参数

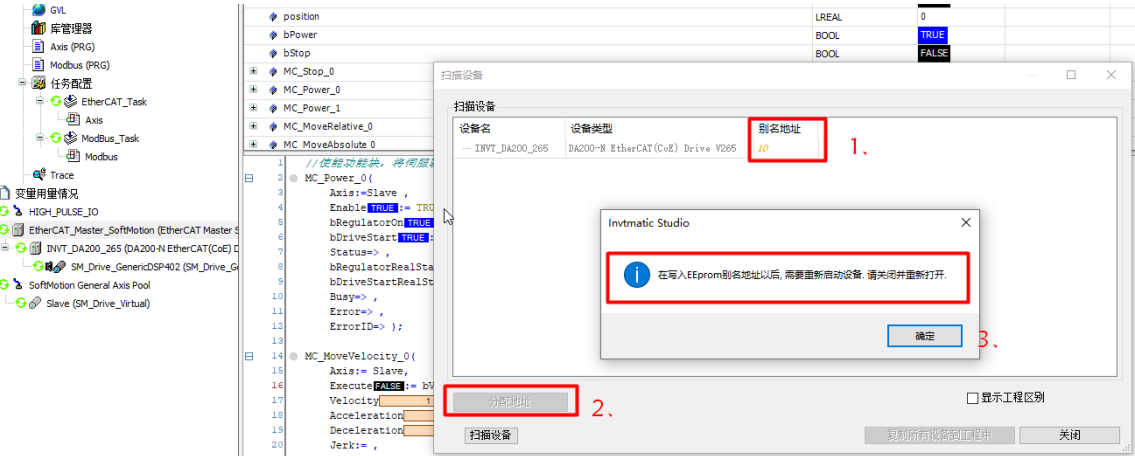


■ 站点别名

在使用 EtherCAT 地址时，当实际设备的连接顺序与组态顺序不一致，总线无法正常运行。如果用户希望实际从站连接顺序不受组态顺序的影响，可以使用别名功能，别名即是给该伺服重新命名，在连接过程中，不再通过自动分配的节点地址识别伺服，而是通过名称进行识别。要正确使用站点别名功能首先需要设置从站的站别名，设置步骤如下：

步骤1 在扫描到相关设备后，可以对别名地址进行设置。

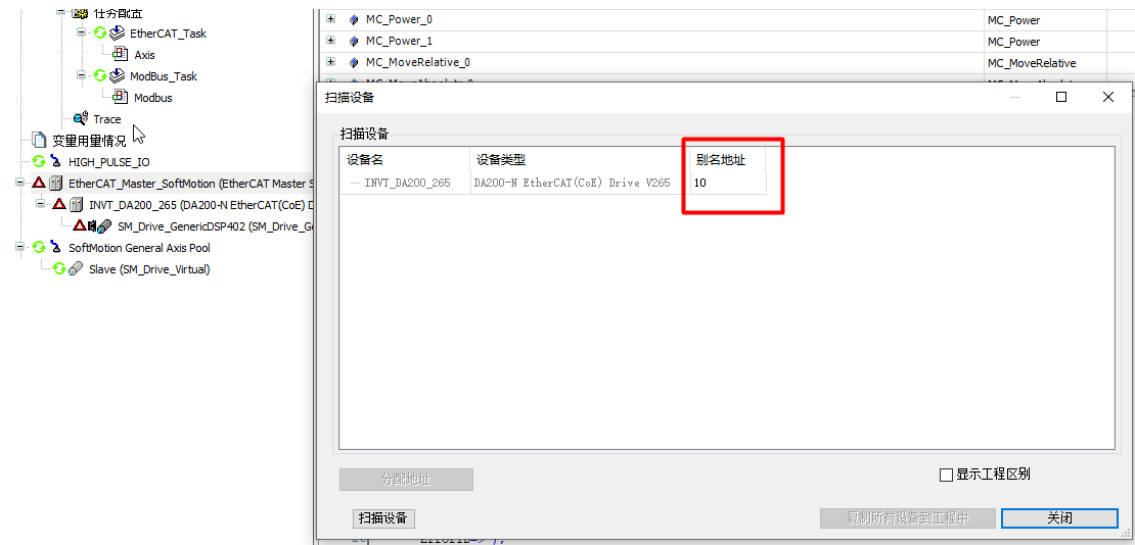
图 5-28 设置站别名





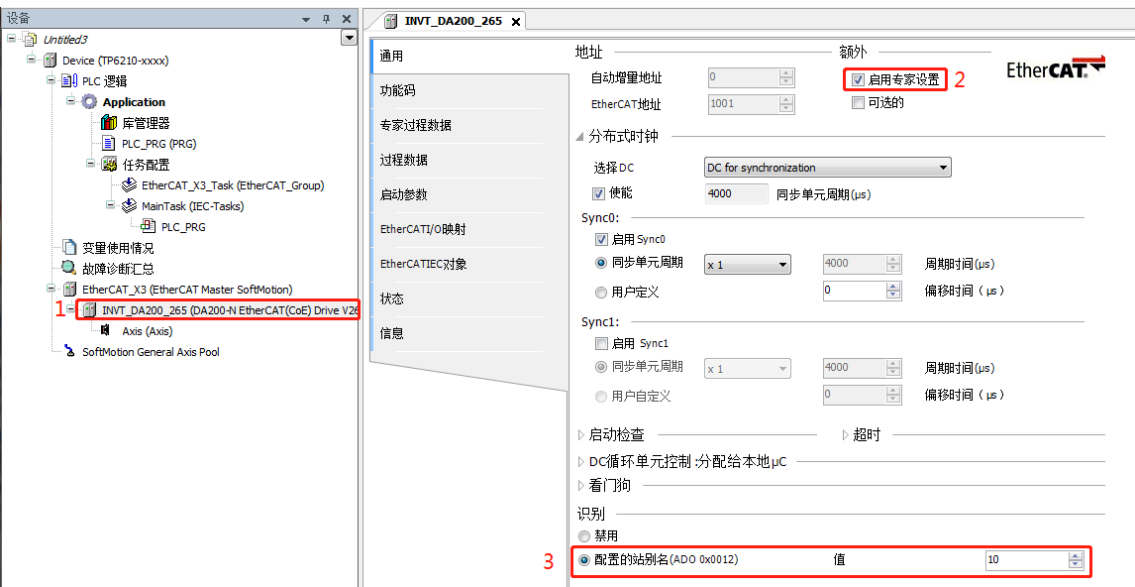
步骤2 待分配完成后，需要对伺服进行断电重启，才能生效。再次扫描，此时别名地址已被写入。

图 5-29 重启站别名生效



步骤3 双击设备树中的的 EtherCAT 从站设备 INVT\_DA200\_265，勾选启用专家设置，在识别位置，选择配置的站别名，并将正确的别名地址进行填写，如下图所示，重启控制器。

图 5-30 填写站别名



5.3.2.2 过程数据（PDO）

从站启用专家设置后，会增加专家过程数据界面，界面会显示从站描述文件自带的输入输出 PDO，用户可以根据自身的需要进行添加和删除，如图 5-31 输出 PDO（通常是控制字与给定参数），如图 5-32 输入 PDO（通常是状态字与反馈参数）。在 EtherCAT I/O 映射界面，可映射相应的变量在程序中调用，如图 5-33 所示。



图 5-31 输出 PDO

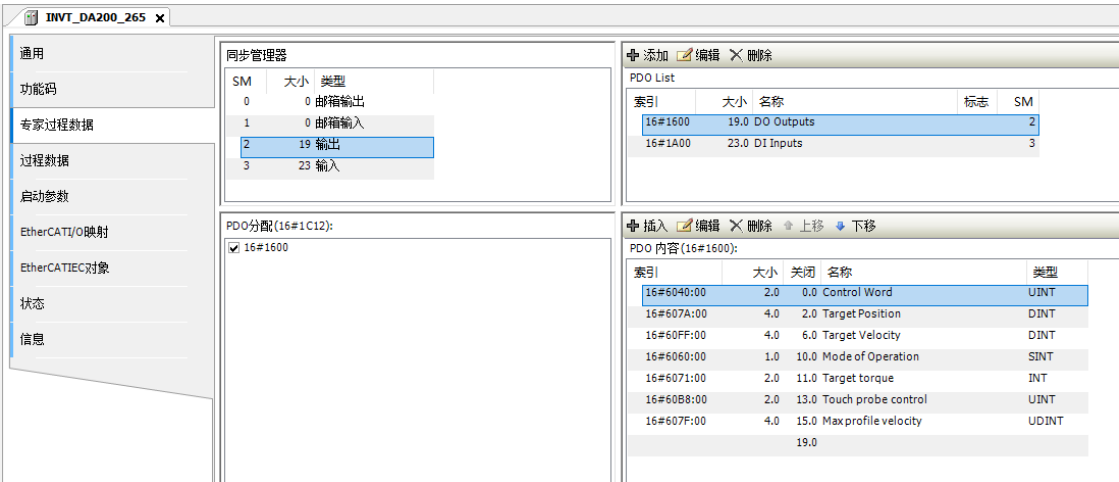


图 5-32 输入 PDO

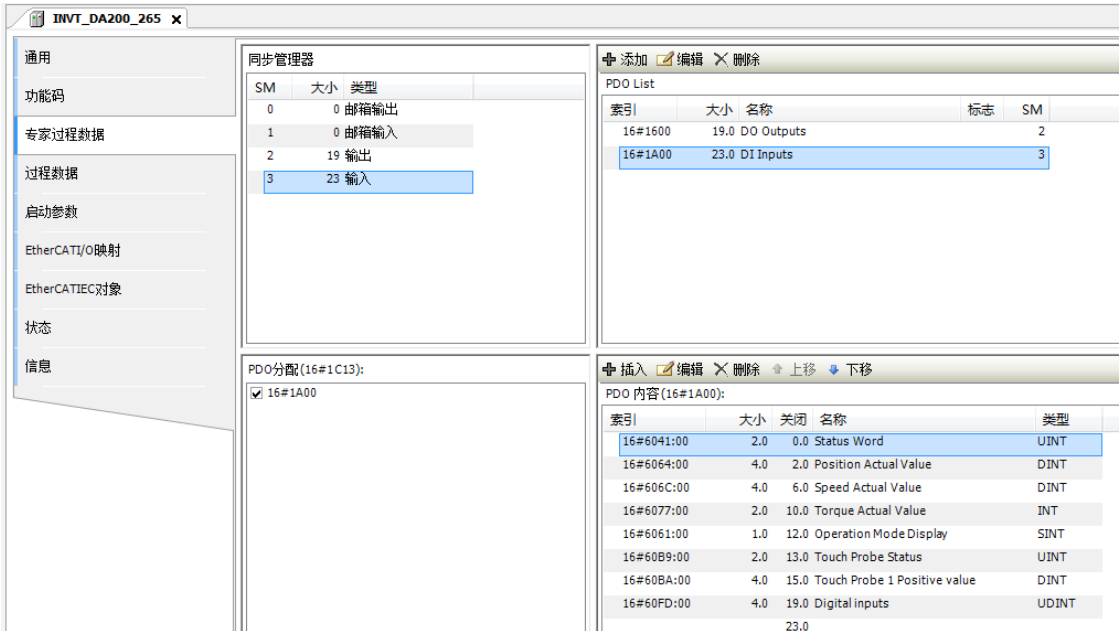
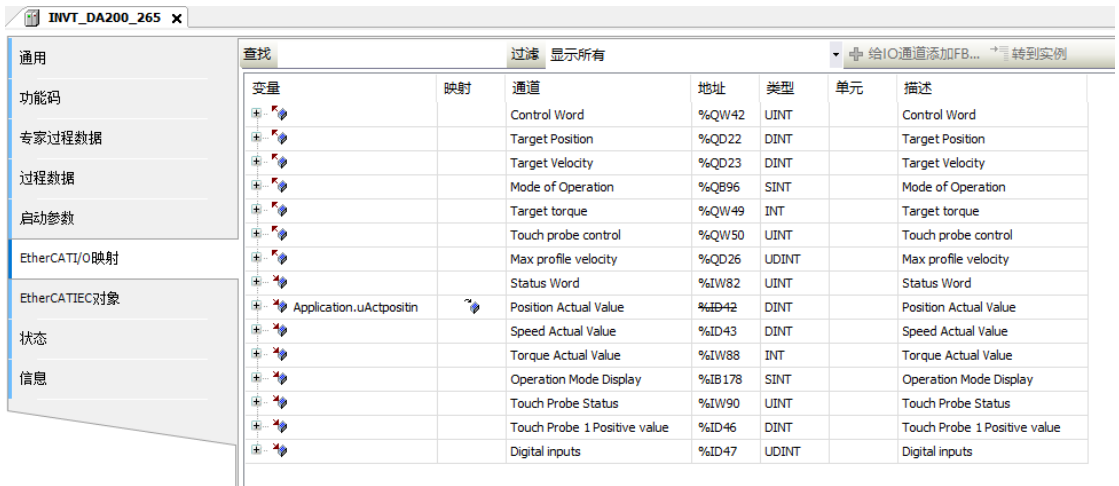


图 5-33 EtherCAT I/O 映射





5.3.2.3 启动参数

启动参数可以从站的一些参数在其处于 PreOP 状态时通过写 SDO 的形式写入从站，如下图所示。

图 5-34 启动参数

INVT\_DA200\_265 x

通用

功能码

过程数据

启动参数

EtherCAT I/O 映射

EtherCAT IEC 对象

状态

信息

添加 编辑 删除 上移 下移

行	索引: 子索引	名称	值	位长度	如果有错, 则退出	如果有错, 则至跳行	下一行	注释
1	16#6098:16#00	Homing method	22	8	<input type="checkbox"/>	<input type="checkbox"/>	0	
2	16#6099:16#02	Speed during search for zero	100	32	<input type="checkbox"/>	<input type="checkbox"/>	0	
3	16#609A:16#00	Speed during search for switch	1666667	32	<input type="checkbox"/>	<input type="checkbox"/>	0	
4	16#6099:16#01	Speed during search for switch	16667	32	<input type="checkbox"/>	<input type="checkbox"/>	0	

在实际使用过程中，用户可根据自己的需求添加启动参数。以 DA200 伺服驱动器为例，如需在启动时修改回原点模式 35，操作步骤如下：

- 步骤1 点击启动参数界面的添加，如图 5-34 所示。
- 步骤2 在弹出的窗口中找到对象字典 16#6098，即回原点模式，如图 5-35 所示，将值设置为 35，点击确定。

图 5-35 回原点模式选择

从对象目录中选择条目

索引: 子索引	名称	标志	类型
16#6083:16#00	Profile acceleration	RW	UDINT
16#6084:16#00	Profile deceleration	RW	UDINT
16#6091:16#00	Gear ratio		
16#6093:16#00	Position factor		
16#6098:16#00	Homing method	RW	SINT
16#6099:16#00	Homing speeds		
16#609A:16#00	Homing acceleration	RW	UDINT
16#60B2:16#00	Torque offset	RW	INT
16#60B8:16#00	Touch probe control	RW	UINT
16#60E0:16#00	Positive torque limit	RW	UINT
16#60E1:16#00	Negative torque limit	RW	UINT
16#60FE:16#00	Digital outputs	RW	UDINT
16#60FF:16#00	Target Velocity	RW	DINT

名称Homing method

索引: 16#6098 子索引: 16#0

位长度8 值35

☐ 字节数组

确定 取消

- 步骤3 添加完成后，在列表中会显示对应的项目，代表启动参数添加成功。如图 5-34 所示。



5.3.2.4 CiA 402 轴

在 Invtmatic Studio 中使用运动控制功能时,需要先在 EtherCAT 从站下添加 402 轴,该轴在程序中作为控制对象。有关 EtherCAT 主站相应参数配置详见 5.3.1 EtherCAT 主站配置,此处以 EtherCAT 主站连接 DA200 伺服驱动器从站使用为案例,仅供参考使用。

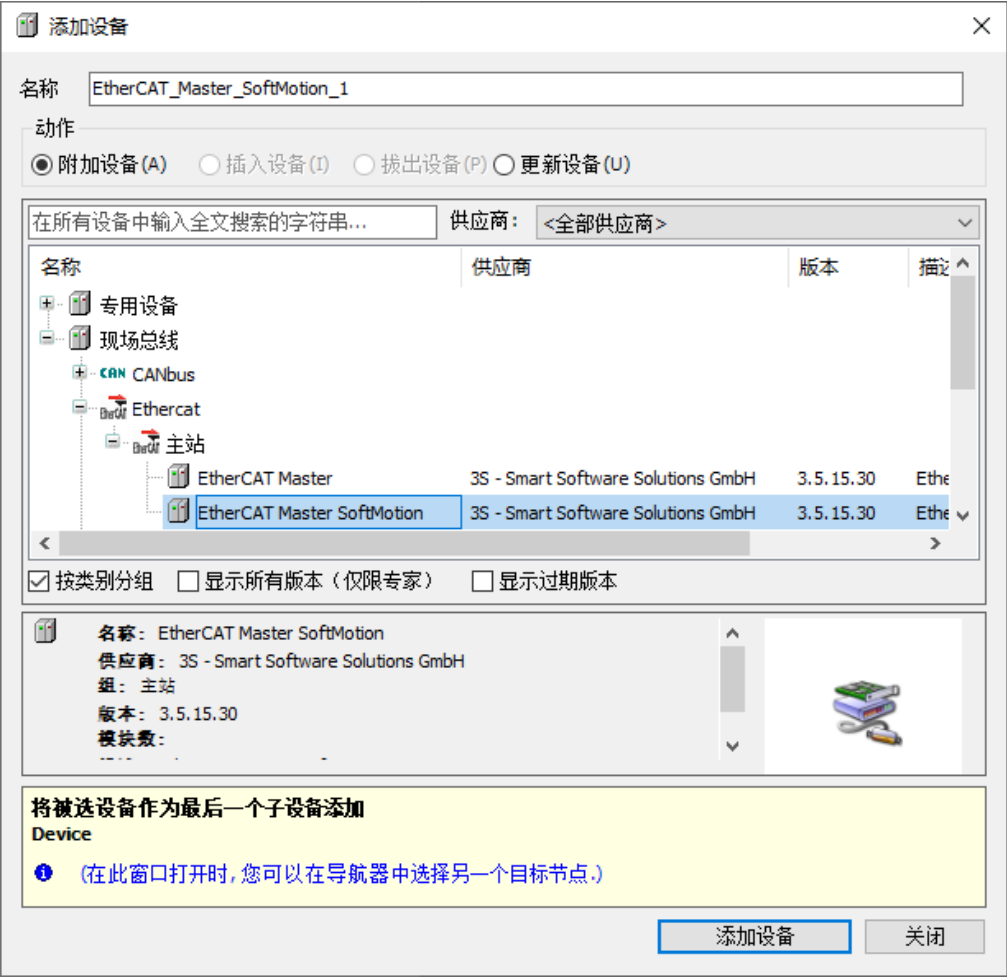
如果使用 INVT 公司的从站,Invtmatic Studio 已经进行了预处理,添加从站时会自动添加与其对应的轴。对于用户自己添加的第三方从站,在使用时需要手动添加 402 轴,添加本模块需要的库文件“INVT\_DA200\_xxx.devdesc.xml”,以 INVT\_DA200\_265 为例,操作步骤如下:

注意:

- 在创建 EtherCAT Master SoftMotion 工程时其任务优先级推荐使用最高优先级 0。
- 同步周期和任务周期此处推荐保持一致设置≥4ms。
- 创建 EtherCAT Master SoftMotion,推荐使用单独任务,例如 I/O、模拟量输入输出、modbus 通信等任务和 EtherCAT Master SoftMotion 任务需要分开。

步骤1 在设备树中右键点击 **Device**, 选择**添加设备**, 选中**现场总线>Ethercat>主站>EtherCAT Master SoftMotion**。

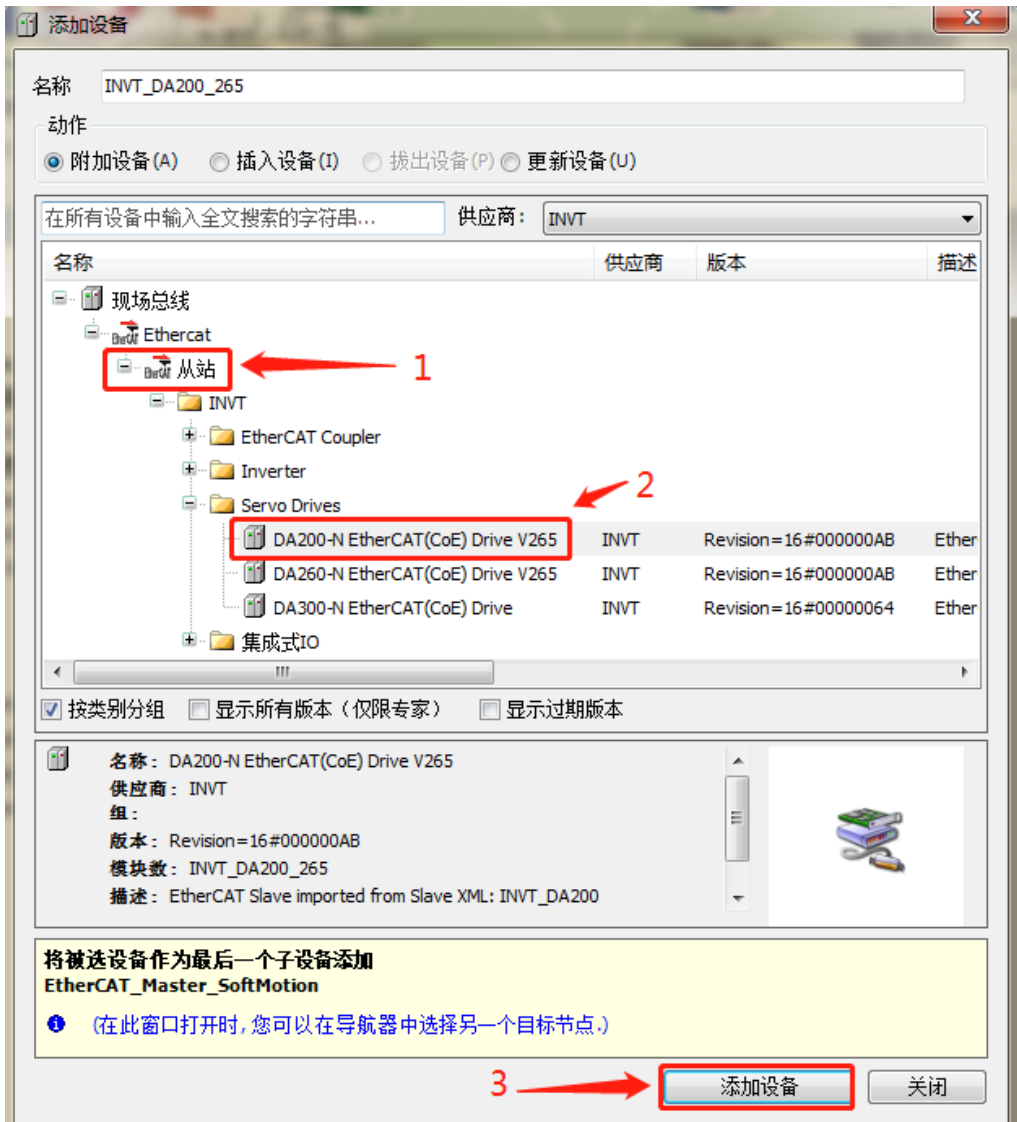
图 5-36 添加 EtherCAT 运动控制主站过程



步骤2 右键点击 **EtherCAT\_Master\_SoftMotion**, 选择**添加设备**, 添加 INVT DA200 伺服驱动器具体操作流程如下图所示。

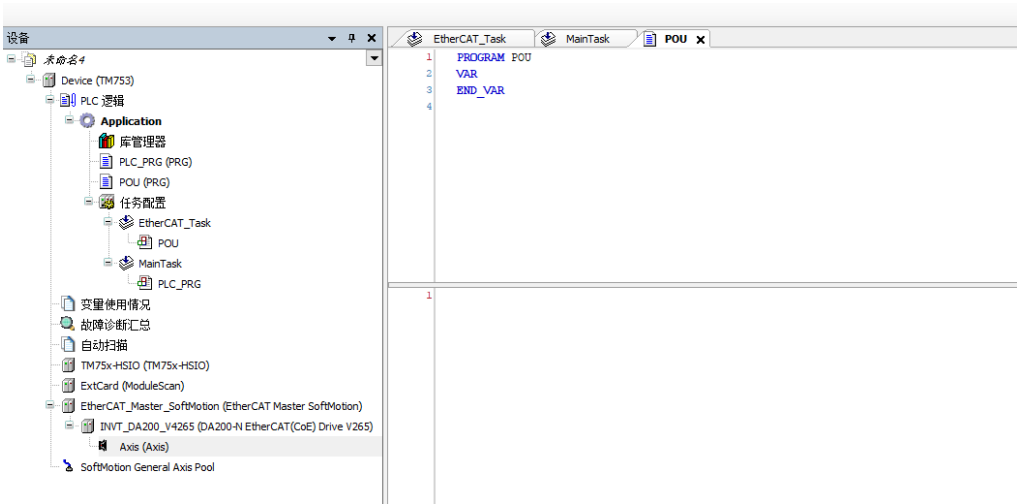


图 5-37 DA200 伺服驱动器添加步骤



步骤3 选中设备树中的 INVT\_DA200\_265，单击右键添加 **SoftMotion CiA 402 轴**，添加调用程序如下图所示。

图 5-38 DA200 伺服驱动应用范例





5.3.2.5 EtherCAT 从站常用参数

EtherCAT 从站常用参数如表 5-14 所示。在使用时，需要加上从站名，如从站名为 INVT\_DA200\_265，则读取从站状态时，对应的变量为 INVT\_DA200\_265.wState。

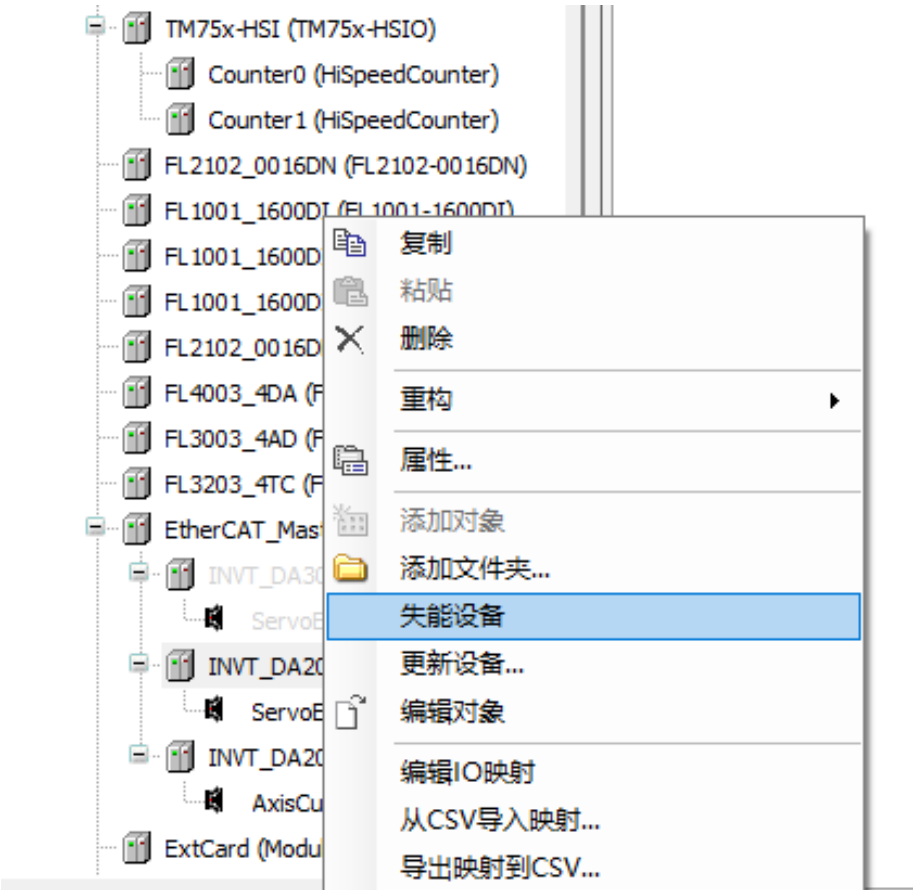
表 5-14 EtherCAT 从站常用参数

变量名	功能
wState	1: ETC_SLAVE_INIT 2: ETC_SLAVE_PREOPERATIONAL 3: ETC_SLAVE_BOOT 4: ETC_SLAVE_SAVEOPERATIONAL 8: ETC_SLAVE_OPERATIONAL
SlaveAddr	从站地址

5.3.2.6 使能/失能从站

在实际应用中，用户可能会遇到组态与实际硬件连接不一致的情况，导致总线无法正常运转。针对这种情况，Invtmatic Studio 提供了一种解决方案，用户可以失能没有连接的从站，以保证总线能正常运转。程序中使用到设备相关的变量时，采用失能设备的方式禁用设备，也不会导致编译报错。

图 5-39 设置失能设备

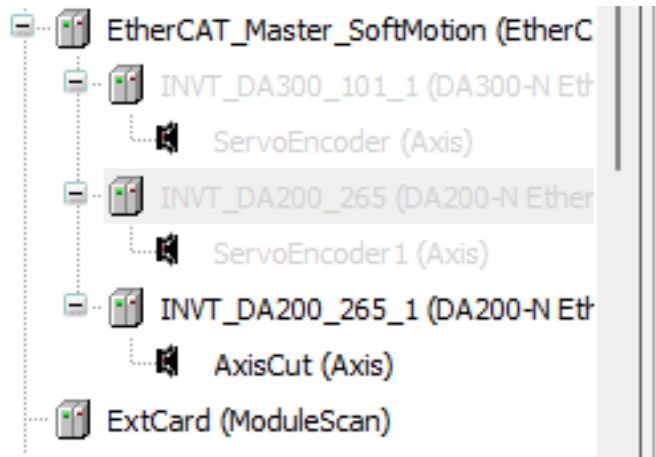


注意：

- 被失能的设备在树中的输入以浅灰色字体显示，如下图所示。

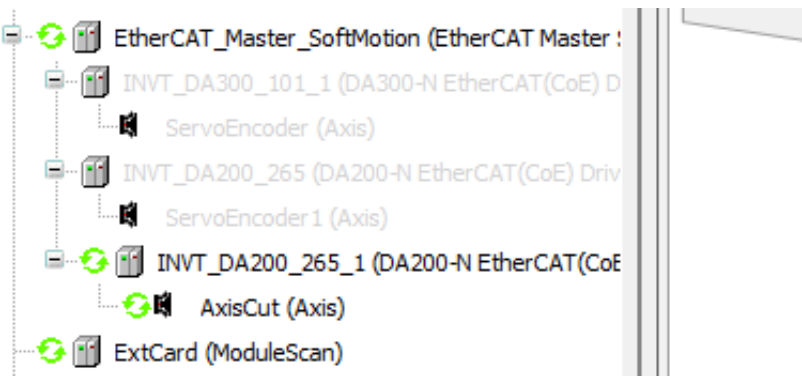


图 5-40 设备已失能



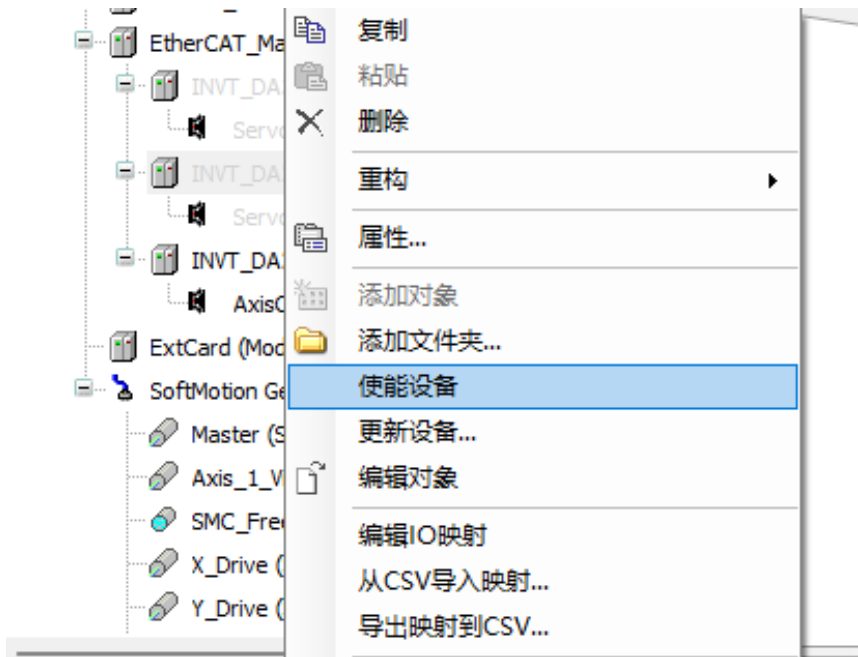
- 当登录时，被失能的设备将不会显示运行状态，如下图所示。

图 5-41 程序运行失能设备不运行



- 需要重新使能设备时，点击“使能设备”即可，如下图所示。

图 5-42 设置使能设备





### 5.3.3 EtherCAT 冗余环网功能

EtherCAT 可选用电缆冗余技术，满足快速增长的系统可靠性需求，它可以保证无需关闭网络即可进行设备更换。且增加冗余特性耗费不高，仅需在主站设备端增加一个标准的以太网端口（无需专用网卡或接口）和一根电缆，可以将线型拓扑结构转变为环型拓扑结构。当设备或电缆发生故障时，也仅需一个周期即可完成切换。因此，即使是针对运动控制要求的应用，电缆出现故障时也不会有任何问题。

EtherCAT 使用热备份功能支持主站冗余，一旦出现中断、设备故障等问题，EtherCAT 从站控制器可以立即自动返回以太网帧，所以不会导致整个网络关闭。例如标准 EtherCAT 拓扑结构如图 5-43 的 a) 所示，如果在该拓扑结构中 Slave 2 与 Slave N-2 之间出现了网络中断现象，如图中的红色部分，则 Slave N-2 后的所有从站通讯也会相应中断，这也是标准拓扑结构的缺点。

图 5-43 EtherCAT 冗余

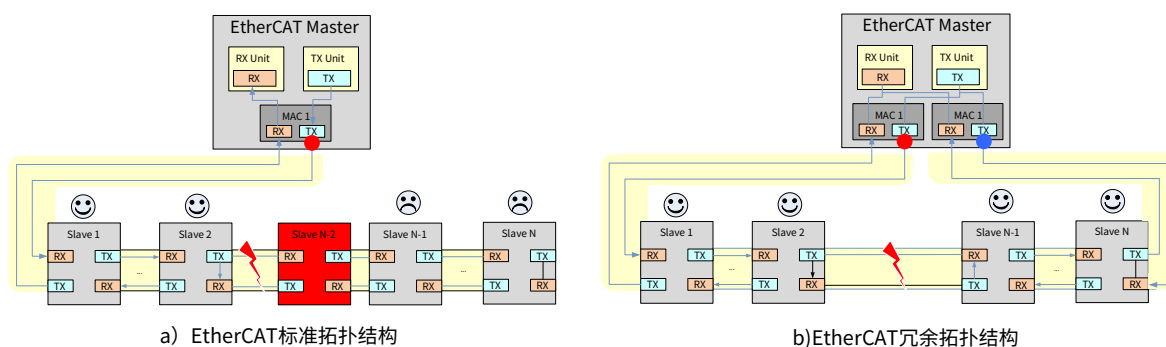


图 5-43 的 b) 为 EtherCAT 冗余模式的拓扑结构，主站只需要有两个标准网口即可实现该拓扑结构，使用这两个网口可以将所有从站构成一条环路，即使在使用过程中网络出现中断，如图 5-43 中红色的部分断开，主站马上会检测到错误，自动将通讯分为两路，所有的从站还能继续通讯，以保障系统的稳定运行。

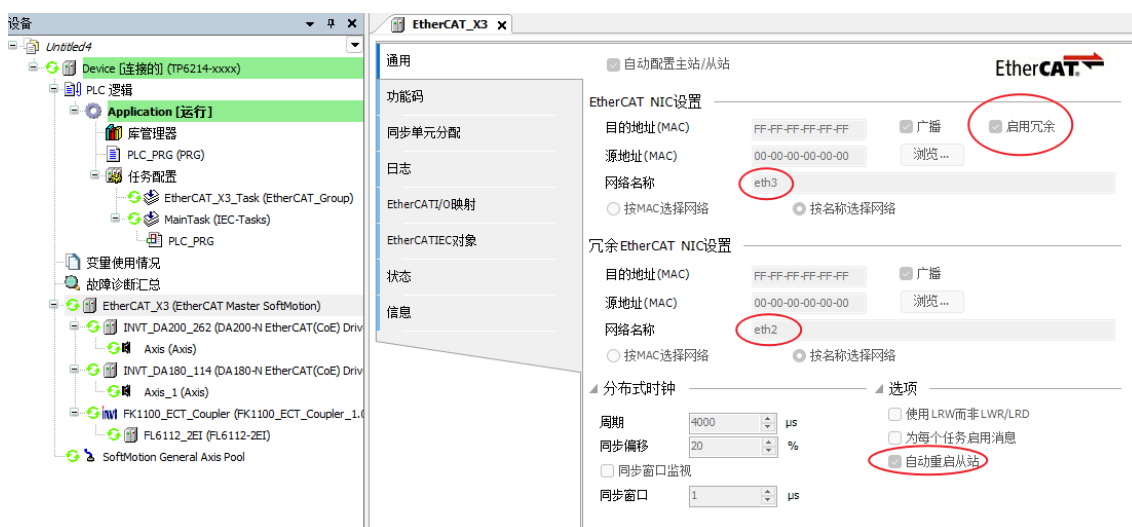
**注意：**按照 EtherCAT 协议标准，第一个从站是标准时钟，因此第一个从站故障，会影响整个总线的通信。

如图 5-44 所示，冗余环网的配置操作如下：

双击打开从站 EtherCAT\_X3，勾选的启用冗余后，会显示冗余 EtherCAT NIC 设置参数，在 EtherCAT NIC 设置和冗余 EtherCAT NIC 设置中，建议统一选用按 MAC 选择网络或按名称选择网络。Invtmatic Sstudio 默认采用按名称选择网络，点击浏览，分别配置好对应的 EtherCAT 网络接口，并勾选自动重启从站。

**注意：**当拨去第 1 个从站和第 2 个从站之间的网线时，总线上的 3 个从站都依然能正常运行。

图 5-44 EtherCAT 冗余环网配置

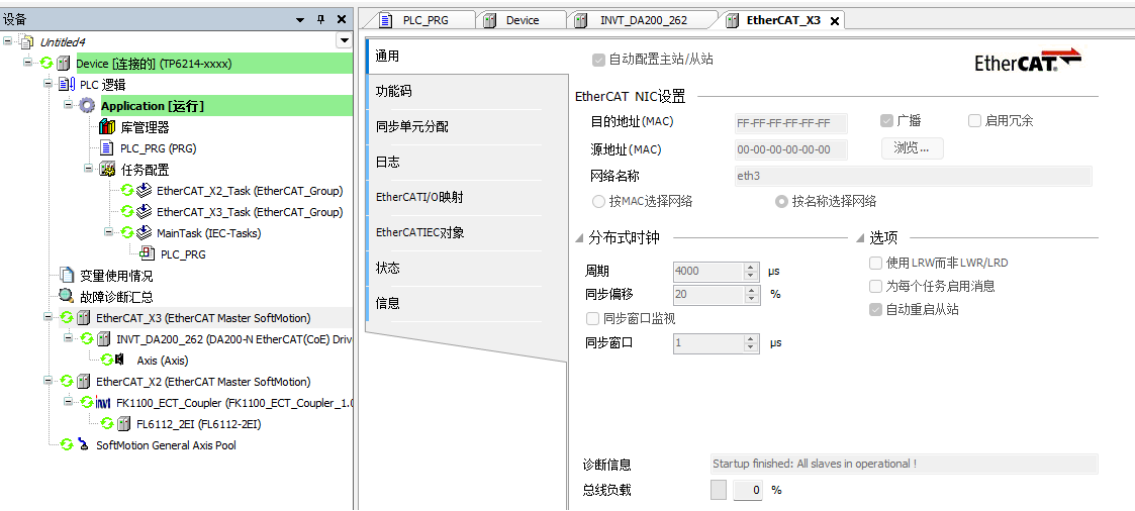




5.3.4 EtherCAT 双主站

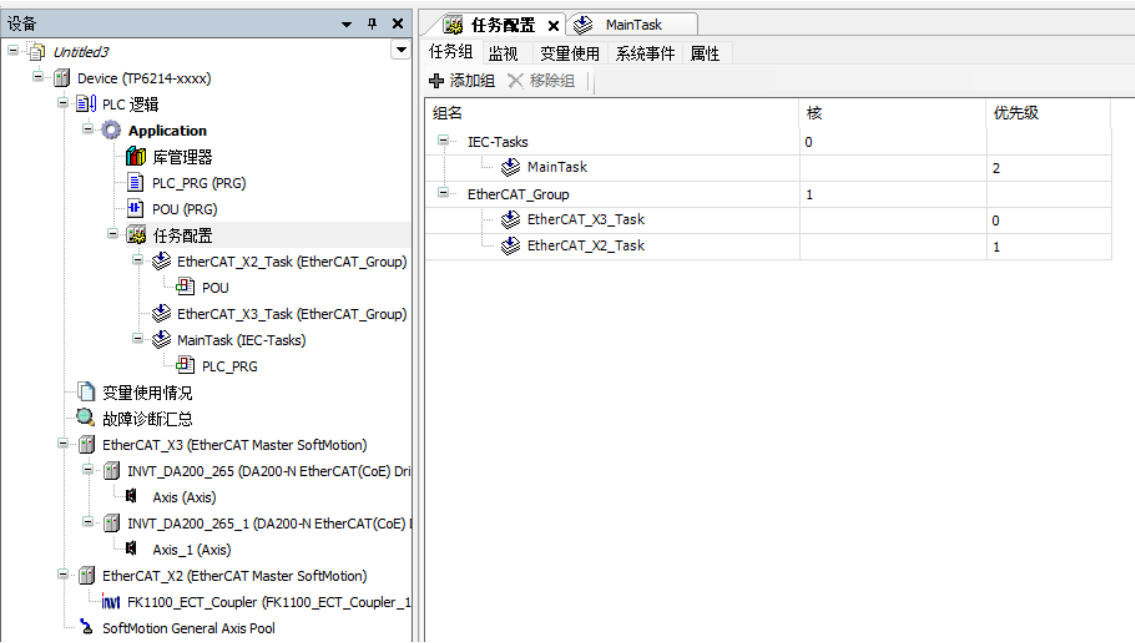
TP6214 支持 EtherCAT 双主站，在 Invtmatic Studio 软件中，当选择设备 TP6214 时，可以添加两个 EtherCAT 主站，并自动分配好 EtherCAT 网口 X3 和 X2，如图 5-45 所示。通常建议实时性较高的伺服和 402 轴连接在 EtherCAT\_X3，耦合器连接在 EtherCAT\_X2。

图 5-45 EtherCAT 双主站连接效果



**注意：**双主站任务配置要求 EtherCAT\_X3\_Task 任务优先级设置为最高 0，EtherCAT\_X2\_Task 任务优先级设置为 1，其它任务从优先级 2 开始设置，如下图所示。

图 5-46 双主站任务优先级配置





## 5.4 CANopen

CANopen 是一种架构在控制局域网路 (Controller Area Network, CAN) 上的基于 CAL 协议扩展的高层通讯协定，包括通讯子协议和设备子协议。

通讯模型定义了 4 种报文 (通讯对象)：

### 1、 管理报文

主要涉及层管理，网络管理和 ID 分配服务，例如初始化、配置和网络管理 (包括：节点保护) 等，这些服务遵循 CAL 中的 LMT、NMT 和 DBT 服务部分，采用主从通讯模式，即在 CAN 网络中，只能有一个 LMT，NMT 或 DBT 主节点以及一个或多个从节点。

### 2、 服务数据对象 SDO (Service Data)

通过使用索引和子索引 (在 CAN 报文的前几个字节)，SDO 允许客户机能够访问设备 (服务器) 对象字典中的项 (对象)。

SDO 通过 CAL 中多元域的 CMS 对象来实现，支持传送任何长度的数据 (当数据超过 4 个字节时分拆成几个报文)。

SDO 中有较多的协议规定，为每个消息生成一个应答 (一个 SDO 需要两个 ID)。

SDO 请求和应答报文总是包含 8 个字节 (没有意义的字节长度在第一个字节中表示，第一个字节携带协议信息)。

### 3、 过程数据对象 PDO (Process Data Object)

用来传输实时数据，数据从一个创建者传到一个或多个接收者。数据传送限制在 1 到 8 个字节 (例如，一个 PDO 可以传输最多 64 个数字 I/O 值或 4 个 16 位的 AD 值)。

PDO 通讯没有协议规定，PDO 数据内容只由它的 CAN ID 定义，假定创建者和接收者都知道这个 PDO 的数据内容。

每个 PDO 在对象字典中用 2 个对象描述：

- A、 PDO 通讯参数：包含哪个 COB-ID 将被 PDO 使用，传输类型、禁止时间和定时器周期。
- B、 PDO 映射参数：包含一个对象字典中对象的列表，这些对象映射到 PDO 里，包括它们的数据长度 (in bits)。创建者和接收者都必须知道这个映射，以解释 PDO 内容。

PDO 消息的内容是预定义的 (或者在网络启动时配置的)，映射应用对象到 PDO 中是在设备对象字典中描述的。如果设备 (创建者和接收者) 支持可变 PDO 映射，那么使用 SDO 报文可以配置 PDO 映射参数。

PDO 有 2 种传送方式：

#### A、 同步 (通过接收 SYNC 对象实现同步)

非周期：由远程帧预触发传送，或由设备子协议中规定的对象特定事件预触发传送。

周期：传送在每 1 到 240 个 SYNC 消息后触发。

#### B、 异步

由远程帧触发传送。

由设备子协议中规定的对象特定事件触发传送。

### 4、 预定义报文或者特殊功能对象

- A、 同步 (SYNC)
- B、 时间标记对象 (Time Stamp)
- C、 紧急事件 (Emergency)
- D、 节点保护 (Node guarding)



### 5.4.1 CANopen 主站配置

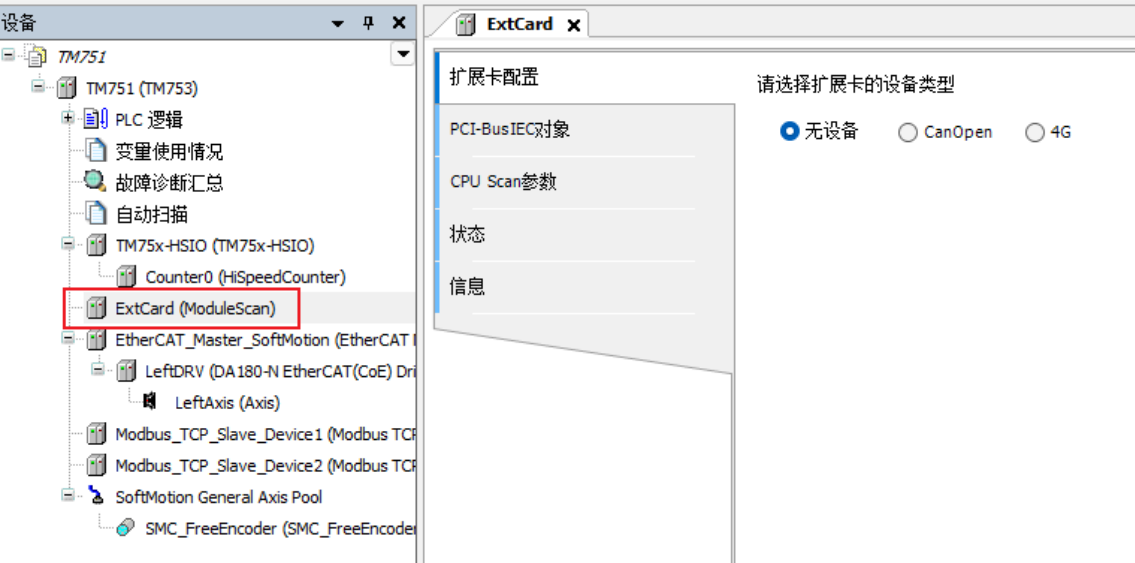
#### 5.4.1.1 主站的使用流程

相关的 CANopen 从站设备描述文件必须先被安装到系统中，这里提到的设备描述文件可以是\*.devdesc.xml 文件或制造商专用的 EDS（电子数据表）文件。

安装相应的 CANopen 从站设备，例如 TM 系列 PLC 使用 CANopen，需要先安装扩展模块，操作步骤如下：

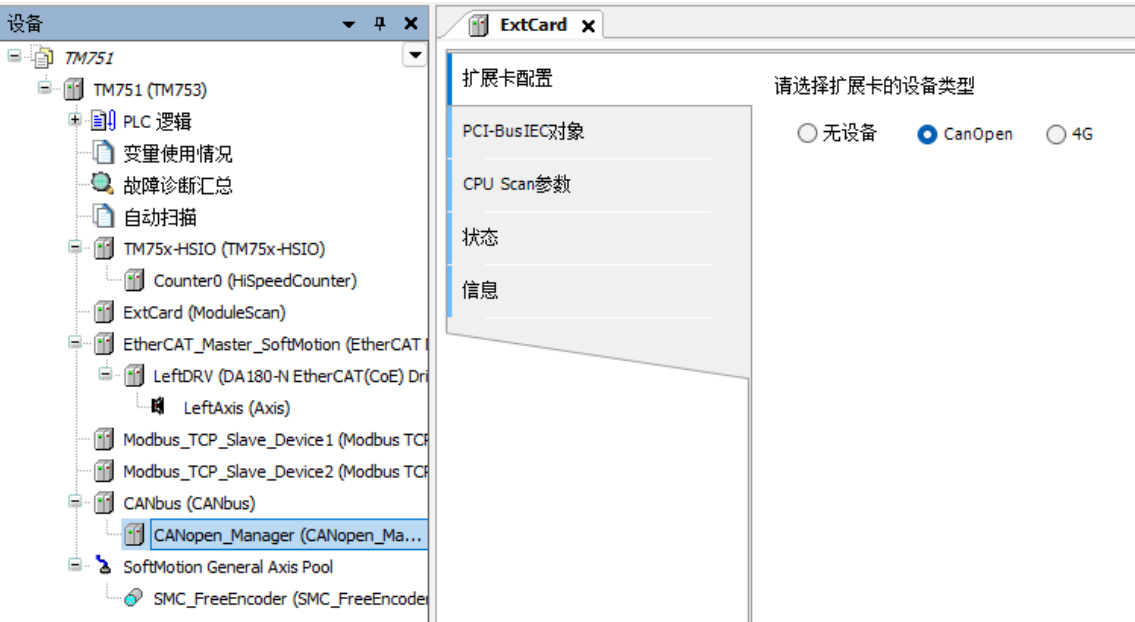
步骤1 在设备树中双击 **ExtCard**，进入设置界面，默认是无设备选项。

图 5-47 ExtCard 扩展卡界面



步骤2 选择 **CanOpen**，左侧设备树会自动添加 **CANbus** 总线和 **CANopen\_Manager**。

图 5-48 选择 CANopen 扩展

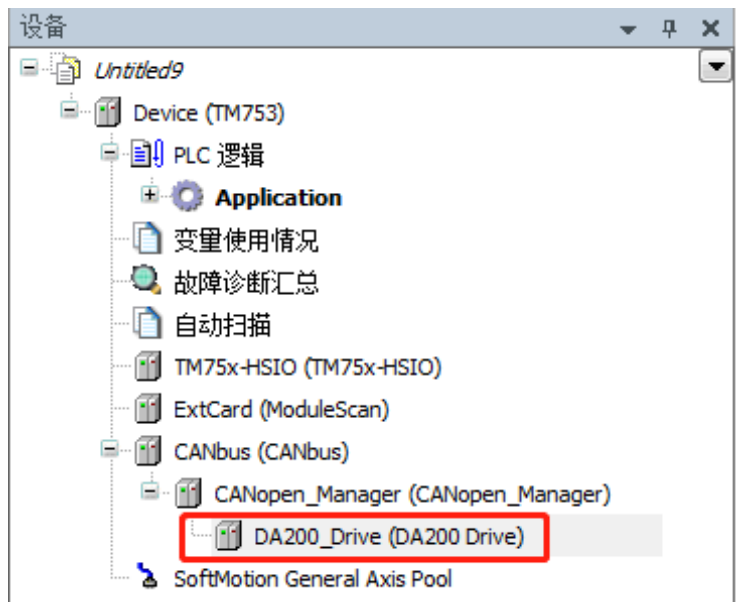




### 5.4.1.2 添加 CANopen 从站

此处以我司 DA200 CANopen 从站为例，在已完成本从站 EDS 文件添加的基础上，在 CANopen Manager 下添加 DA200 从站设备，添加效果图如下。

图 5-49 添加 CANopen 从站的设备树



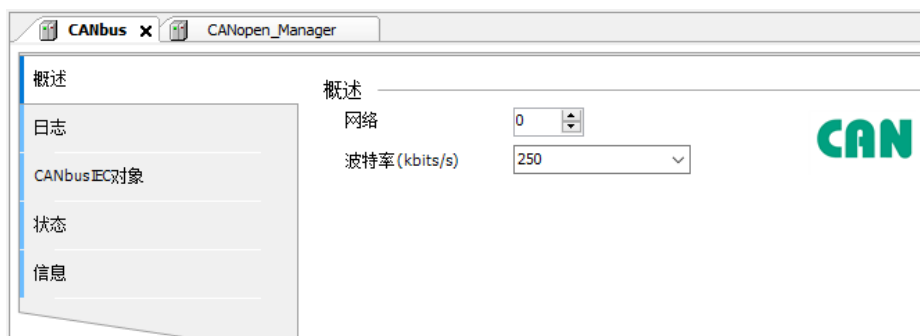
至此完成 CANopen 主站的软件组态工装。

### 5.4.2 CANopen 主站相关参数配置

#### ■ 网络和波特率配置参数

- 网络：通过 CAN 总线连接的 CAN 网络数量，范围 0~100。
- 波特率：总线上用于传输的波特率，可以设置不同的波特率(例如 10kbps、20kbps、50kbps、100kbps/s、125kbps、250kbps、500kbps、800kbps、1000kbps)。

图 5-50 CAN 总线参数配置



#### ■ 主站配置参数

CANopen 管理位于 CANbus 节点下的一个节点，用于通过内部函数支持 CANbus 配置，一般作为 CAN 总线的主站，其部分配置参数如下图所示。



图 5-51 CANopen 主站参数配置

概述

日志

CANopenI/O映射

CANopenEC对象

状态

信息

概述

节点 ID

☒ 自动启动 CANopenManager

☒ 可选从站轮询

☒ 启动从站

NMT错误行为

☐ NMT启动所有 (如果可能)

保护

☒ 使能心跳产生

节点 ID

Producer time (ms)

同步

☐ 启动同步生成

COB-ID (Hex) 16#

循环周期 (μs)

窗口长度 (μs)

☐ 启动同步消耗

时间

☐ 启动时间生成

COB-ID (Hex) 16#

Producer time (ms)

- 概述中的节点 ID: 节点 ID 提供 CANopen 管理器能一一对应的组数对模块, ID 值 1~127 (必须为十进制整数)。
- 保护: 心跳方式是一种传统的保护机制, 不同于节点保护功能, 此功能可以被主站以及从站模块进行处理, 通常情况下配置主站发送心跳到从站设备。
- 使能心跳产生: 如果这个选项被激活, 主站将会根据内部定义的“心跳时间”连续发送心跳信息。保护设置中的节点 ID: 总线上心跳产生(1~127)的唯一标识符。
- 产生时间(ms): 以毫秒定义内部心跳时间。



# 6 诊断

## 6.1 诊断简介

诊断是为了快速定位 PLC 运行过程出现的错误，通过错误信息和状态找出应对的解决方案。Invtmatic studio 上的诊断界面只有登录 PLC 后才能获取和显示。Invtmatic studio 编程系统支持各种通信设备的诊断，可以根据各通信设备实际运行状态生成故障、离线等信息。故障诊断涉及的模块类型主要包括：CPU、Modbus、ModbusTCP、EtherCAT、CANopen 等。Invtmatic Studio 编程系统主要提供了四种获取诊断方式：组态诊断、诊断信息列表、设备自身诊断信息列表和诊断编程接口。所有诊断都是通过诊断码解析获取，诊断码和诊断编程接口相对应。

组态中，每个通信模块诊断状态通过不同的图标来呈现不同状态：运行状态、停止状态、故障状态。

🟢：表示运行状态，设备无故障。

⏸️：表示停止状态，设备没有运行，处于停止状态。

🔴：表示故障状态，在 EtherCAT 和 CANopen 中为发生过故障但已恢复，在 Modbus 中显示为设备处于故障状态。

在组态中能直接看到设备的运行状态。

### 6.1.1 故障诊断

故障诊断用于显示所有设备出现的故障信息，并提供相关故障信息的详细说明、原因排查方法。同时，针对特殊情况还可以提供更详细的诊断信息。设备创建连接后，双击设备树中的故障诊断汇总打开设备故障诊断界面。

图 6-1 故障诊断汇总

设备类型	模块名称	错误代码	错误名称
ModbusTCP	Modbus_TCP_Slave2	8200A1-000F	通讯超时
ModbusTCP	Modbus_TCP_Slave1	E000A0-0005	通讯超时，通讯时间超过了用户设定的通讯最大时间发生了错误
ModbusRTU	Modbus_RTU_Slave2	E00041-0005	通讯超时，通讯时间超过了用户设定的通讯最大时间发生了错误

设备类型窗口可以用于显示当前出现的故障类型，同时提供故障显示过滤功能，可以针对特定的设备类型进行故障信息显示。设备类型包括 CPU 模块、Modbus 模块、Modbus TCP 模块和本地模块。选择不同的设备类型，诊断显示列表显示对应类型的诊断，默认为所有，显示所有设备诊断。

- 查询：根据设备类型或模块名称选项信息，搜索匹配的故障信息。
- 刷新：用于刷新设备故障信息。
- 清空：清空表格中的故障信息。
- 导出 EXCEL：导出表格中的故障信息。
- 故障信息列表：主要用于显示具体模块的故障信息，包括设备类型，模块名称，故障信息。设备类型可以筛选某一类总线故障设备；模块名称可以筛选某一特定名称故障设备。



- 信息详细说明窗口：当在故障信息列表中选中了某一条故障信息，在信息详细说明窗口会显示该故障的详细信息。该窗口包括错误详细说明、原因排查和深入诊断三个选项，错误详细说明窗口的第一栏为故障可能原因说明项，后边增加了四个附加信息，主要用于对故障提供更多的信息说明；原因排查窗口是用于提供原因排查的具体操作方法方式；深入诊断窗口是针对某些复杂错误，需要获取更多细节信息定位的。

## 6.2 设备自身诊断信息列表

### 6.2.1 I/O 诊断

I/O 可以添加在 CPU 本地背板总线下，其诊断信息显示如下，具体诊断码及诊断信息请参见 I/O 模块诊断码。

图 6-2 I/O 设备自身诊断信息

序号	错误代码	错误名称	解决方案
1	Er0032-0015	通道0 信号源开路故障	检查通道0 信号源物理连接

### 6.2.2 Modbus RTU 诊断

Modbus RTU 支持 Modbus 串口 0 和串口 1 两个串口总线。Modbus 串口 0 或串口 1 可以作为 Modbus 主站或 Modbus 从站。

Modbus 串口作为主站时，可以为主站添加从站（远程从站）。在从站的配置界面中，每个从站有一个“错误诊断”界面，详细说明了哪个配置项出现了具体的故障信息。

Modbus 串口作为从站时，也有一个“错误诊断”界面，显示此从站和主站通信出现的故障。

### 6.2.3 Modbus TCP 诊断

PLC 可以作为 Modbus TCP 主站，也可以作为 Modbus TCP 从站。Modbus TCP 作为主站时，可以为主站添加从站（远程从站）。在从站的配置界面中，有一个“错误诊断”界面，详细说明了哪个配置项出现了具体的故障信息。

ModbusTCP 作为从站时，也有一个“错误诊断”界面，显示此从站和主站通信出现的故障。

## 6.3 在线日志

在线日志指当用户连接 PLC 设备后，在线实时显示程序、设备及系统相关的日志信息。该功能用于帮助用户快速定位错误，及时解决问题，保证设备正常运行。

登录 PLC：点击登录，当软件与 PLC 实现登录连接后，会启动日志功能，日志界面如下图所示。

图 6-3 PLC 日志

严重	时间标记	描述	组件
严重	01.01.1970 08:08:30.157	SDO timeout Address: 1001 Index: 16#6099 SubIndex: 2 Data: 16#E20 Result: 16#00	IoDrvEtherCAT
严重	01.01.1970 08:05:11.858	SDO timeout Address: 1001 Index: 16#6099 SubIndex: 2 Data: 16#E20 Result: 16#00	IoDrvEtherCAT
严重	01.01.1970 08:00:10.077	SDO timeout Address: 1001 Index: 16#6099 SubIndex: 2 Data: 16#E20 Result: 16#00	IoDrvEtherCAT



界面中工具条各参数含义如下表所示。

参数名称	参数说明
警告/错误/例外/信息/调试信息	用于对诊断中不同等级信息进行显示和过滤操作。
清除	用于清除当前显示的诊断信息。清除后，再有新的日志信息时，系统会刷新最新信息。
导出	用于导出所有日志信息，导出文档格式为 CSV 格式。
导入	用于导入日志信息
刷新	用于刷新查看日志信息。
所有组件	查看全部组件日志信息，也可选择显示单个组件日志。
离线日志	用于在离线状态下导入保存的 CSV 格式的日志信息。

界面中表格各参数含义如下表所示。

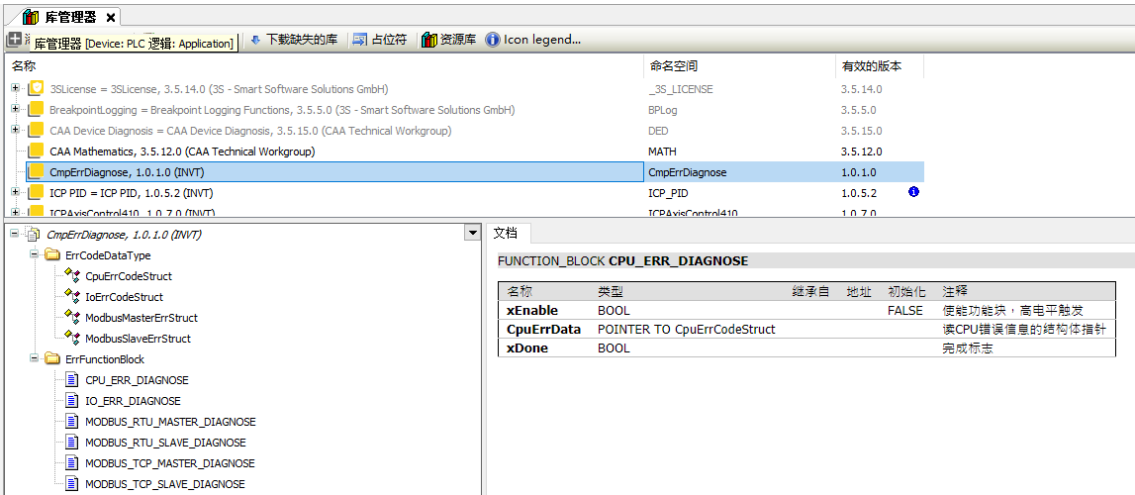
参数名称	参数说明
等级	显示信息的等级。
时间	显示信息发生的时间。
描述	描述故障出现的现象和原因。
组件	显示故障的发生组件。

## 6.4 诊断编程接口

诊断接口库 CmpErrDiagnose，提供了在用户程序中获取诊断的解决方案，可以在用户程序中判断各个设备模块的诊断信息，从而作出相关的处理。

诊断编程接口以库的形式存在，可以在库管理器中添加，添加后如下图所示。

图 6-4 添加诊断编程接口库



编程接口中提供了 CPU、Modbus、ModbusTCP、背板/O 对应的诊断编程接口，每种诊断对应一组功能块，用于获取对应的诊断码。在获取诊断数据时，自定义的诊断结果和诊断状态在 ErrCodDataType 定义（如图 6-4 所示）。比如图 6-5 演示的是 Modbus 主站诊断功能块，1 号串口设置了 3 个从站。其中 strRtuDia 声明为结构体数组，每个数组对应一个从站通道的故障信息，诊断结果为从站 1 有两个通道错误，从站 2 和从站 3 各有一个通道错误。



图 6-5 添加 Modbus 主站诊断功能块

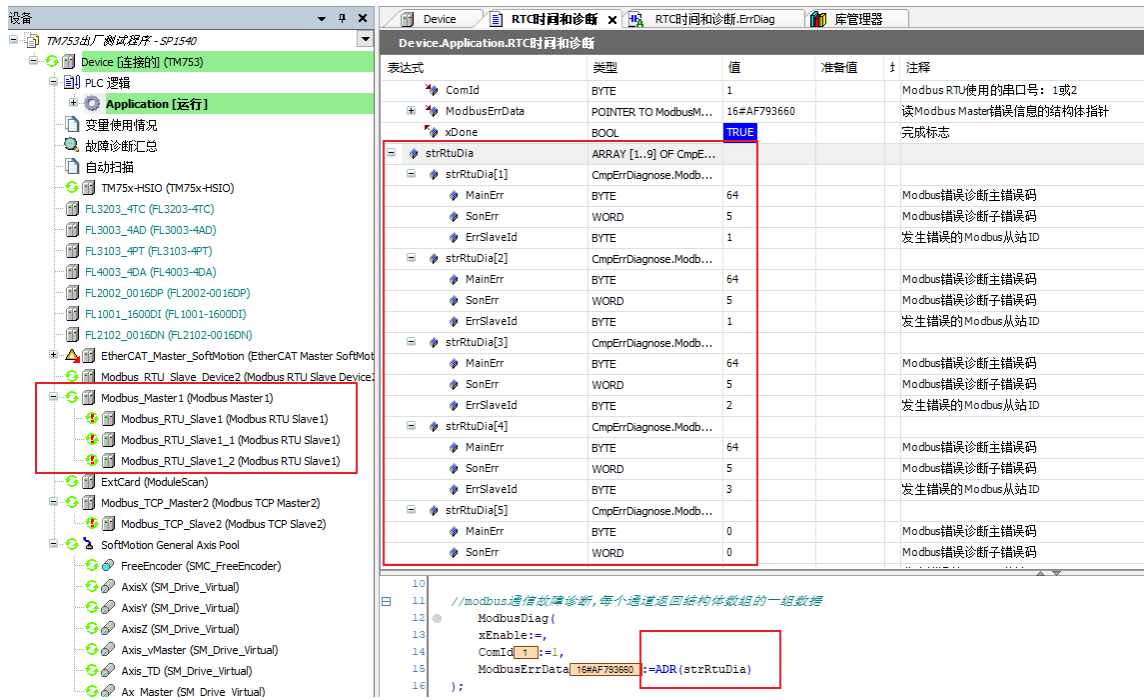


图 6-6 Modbus 主站诊断功能块参数说明

FUNCTION_BLOCK MODBUS_RTU_MASTER_DIAGNOSE						
名称	类型	继承自	地址	初始化	注释	
xEnable	BOOL			FALSE	使能功能块, 高电平触发	
ComId	BYTE				Modbus RTU使用的串口号: 1或2	
ModbusErrData	POINTER TO ModbusMasterErrStruct				读Modbus Master错误信息的结构体指针	
xDone	BOOL				完成标志	

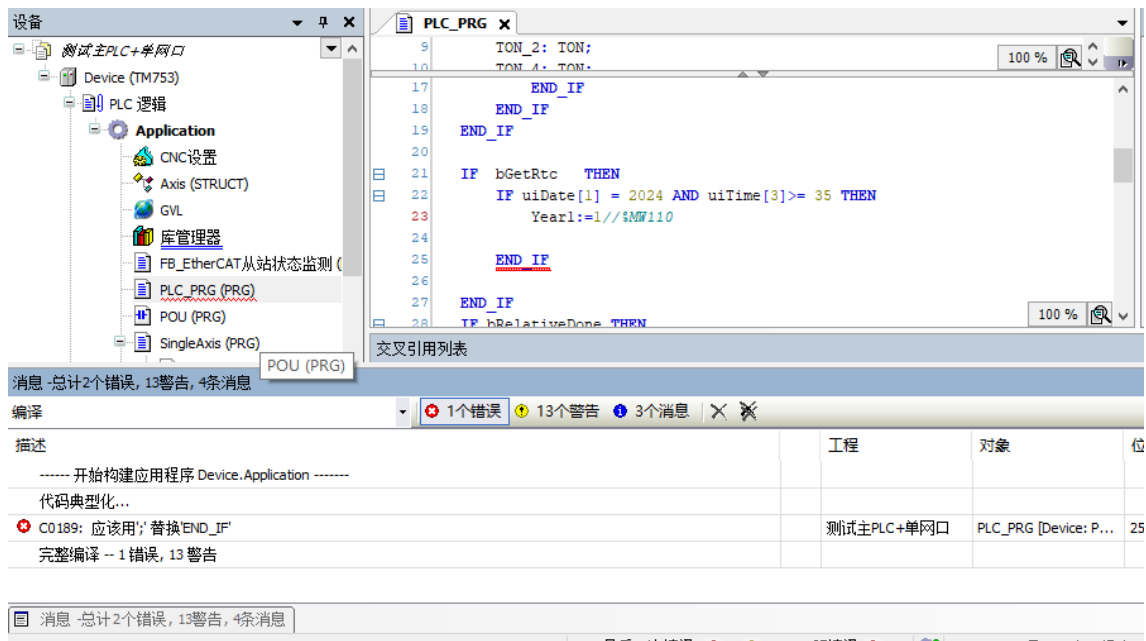
6.5 应用程序诊断

Invtmatic Studio 使用的编程语言具有执行效率高、编程灵活等特点，但是对用户的编程能力也有较高要求。用户在编写程序时需要避免指针非法访问、除数为 0、数组越界、类型隐式转换及死循环等异常操作，否则可能导致 PLC 系统出现执行异常甚至崩溃的情况。本章节主要针对 PLC 可能出现的异常情况提供排查措施，用户可根据实际情况参考使用。

编译工程后，Invtmatic studio 软件默认显示输出编译错误信息。在编译输出窗口，可以查看编译错误，双击错误显示行，可以定位到出错的代码，如下图所示。



图 6-7 定位到出错代码位置



算法库错误信息：用同样的方法可切换到**库管理器**，查看算法库错误提示信息，如图 6-8 所示。打开库管理器，如图 6-9 所示，如果不用到此库，可删除；用到了此库，则需要手动安装算法库。

图 6-8 库管理器显示的提示

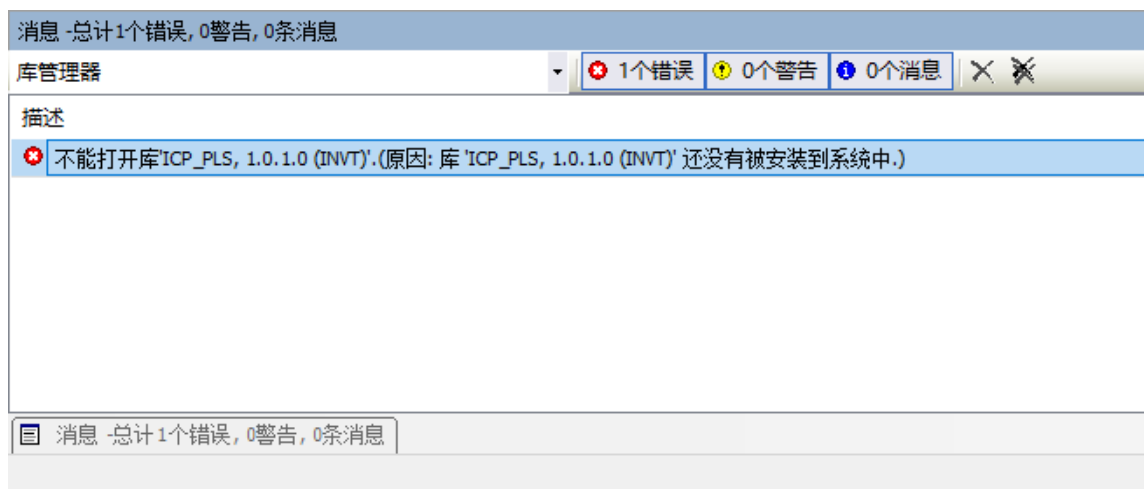
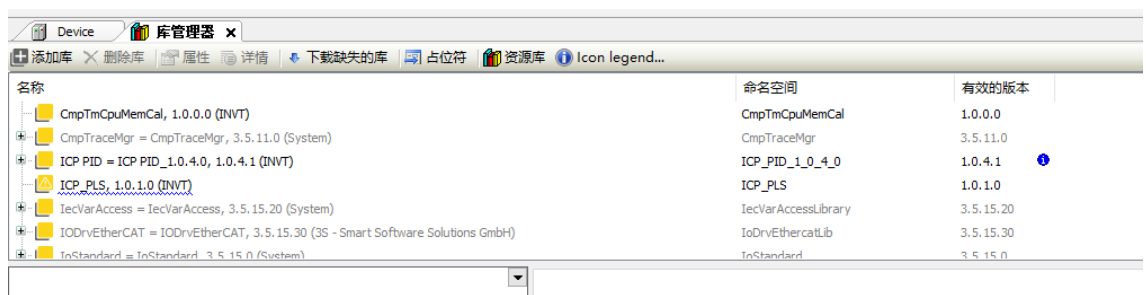


图 6-9 查看缺失的库、添加或删除某个库





## 6.6 PLC 程序运行异常处理

### 6.6.1 常见的异常现象及处理措施

在编写、调试软件的过程中可能遇见以下异常现象：

- 下载程序后编程软件无法扫描到相应的 PLC 设备，PLC 运行拨码置为 stop 之后再重新上电，可正常扫描。
- 下载程序时或者运行一段时间后，PLC 信息显示栏提示错误内容“程序下载异常”，如图 6-10 所示。

图 6-10 程序下载异常

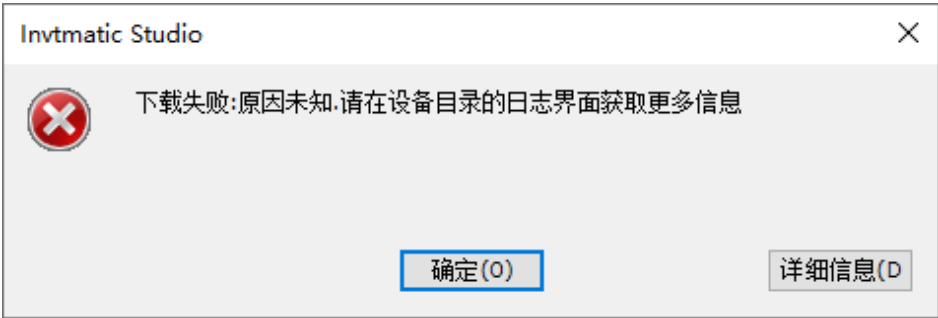


此时在日志界面可以看到有类型错误，程序无法正常运行，如图 6-11 所示。



下载程序时，弹出“下载失败”提示框，如下图所示。

图 6-11 程序下载失败提示框

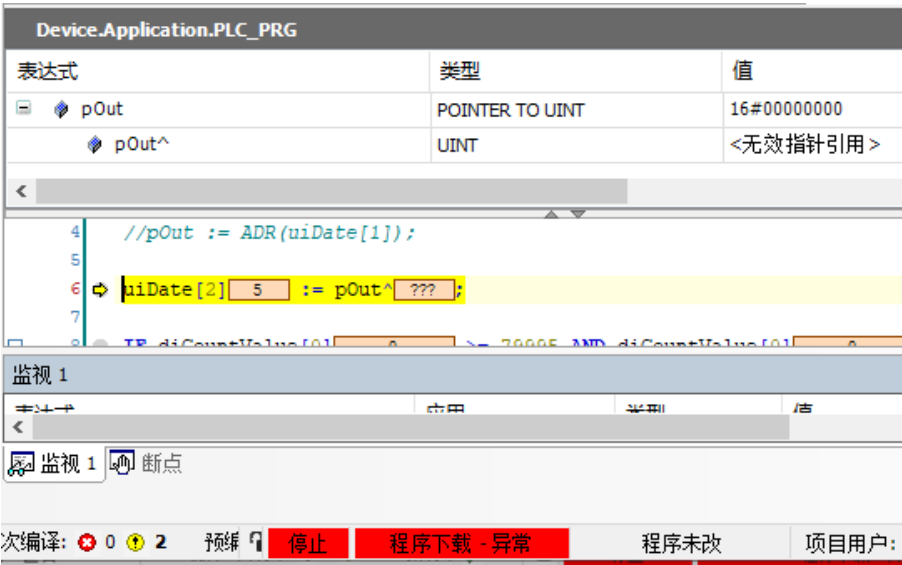


上述异常现象的原因有 3 种：



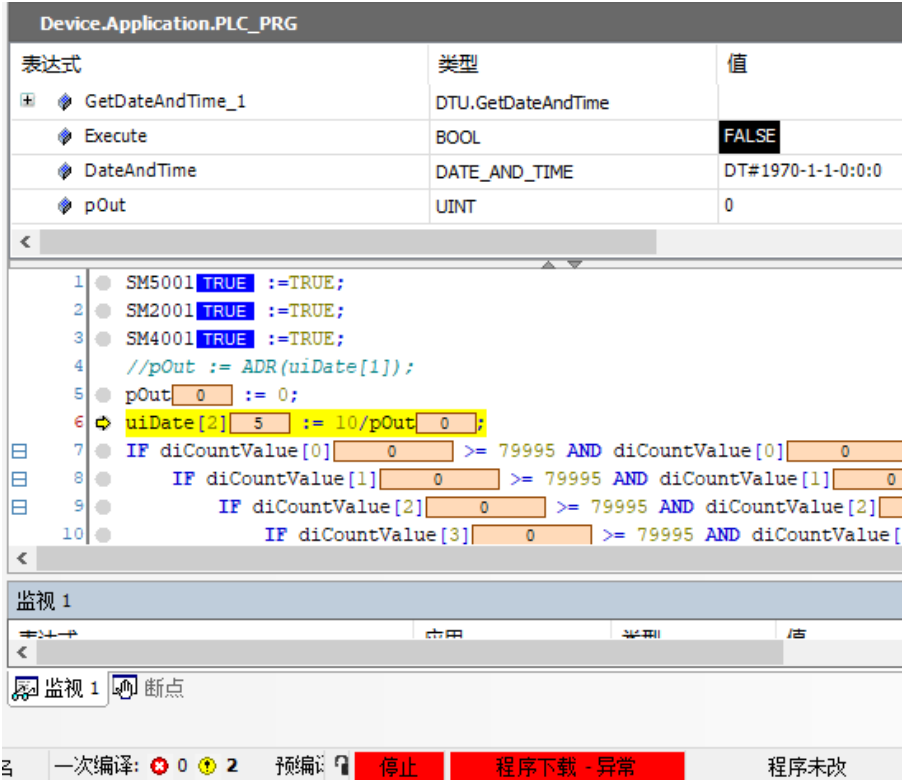
1、空指针（即指针为 0）的情况。

图 6-12 无效指针引用提示程序下载异常



2、除数为 0 的情况。

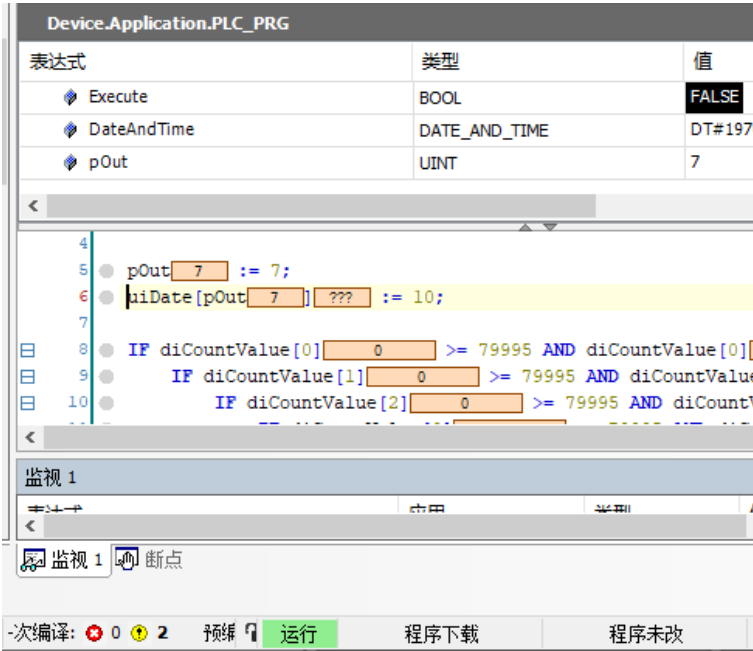
图 6-13 除数为 0 提示程序下载异常





3、 数组越界的情况。

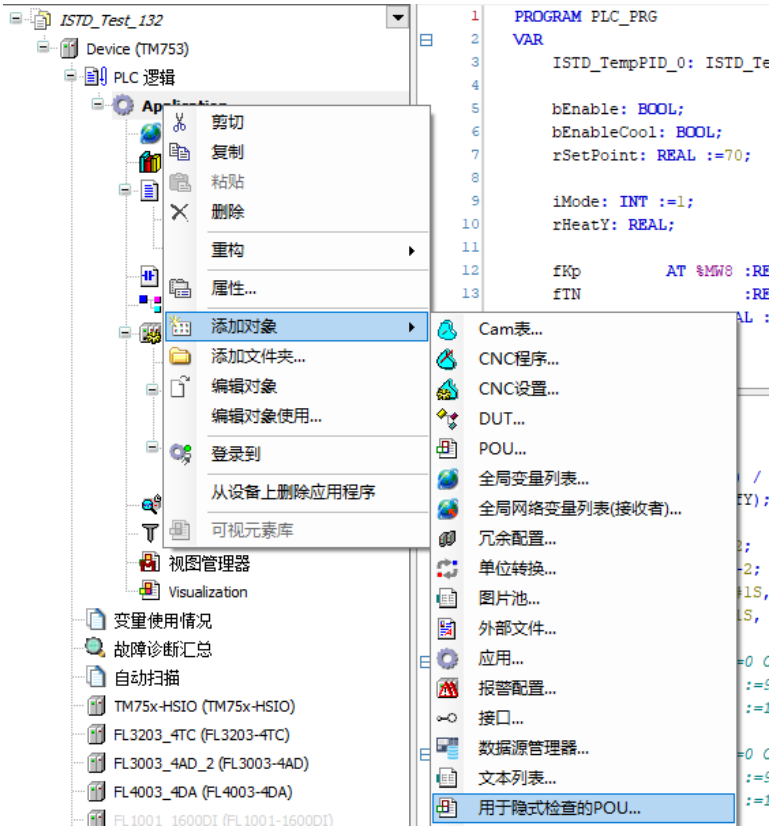
图 6-14 数组越界



空指针和除数为 0 的情况，监控程序运行时直接提示，但数组越界程序能正常运行，排查操作步骤如下：

步骤1 鼠标右键点击 **Application**，选择添加对象>用于隐式检查的 POU。

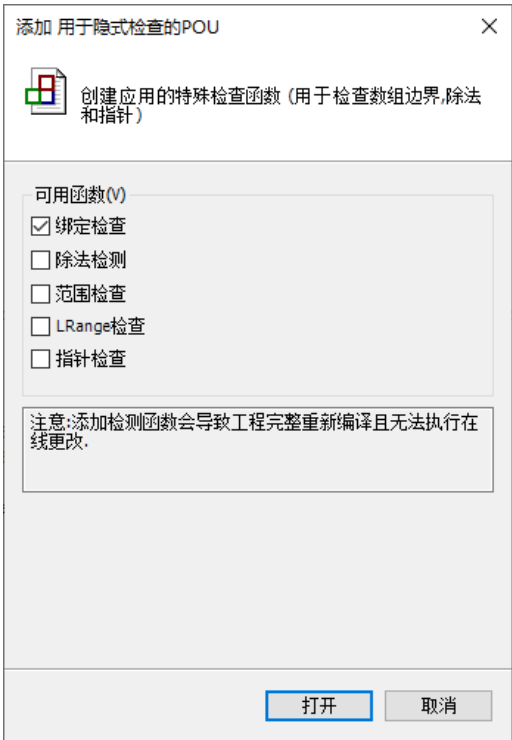
图 6-15 添加用于隐式检测的 POU



步骤2 在弹出界面勾选**绑定检查**，点击**打开**。



图 6-16 勾选绑定检查



- 步骤3 登录 PLC，停止程序运行，在新增的绑定函数“CheckBounds”里面的代码“CheckBounds := lower;”和“CheckBounds := upper;”处添加断点并激活（按快捷键 F9）。
- 步骤4 运行程序，当检测到数组越界后，进入断点，并检测到越界的数值（7）及定义数组上限值（3）、下限值（1），如图 6-17 所示。

图 6-17 数组越界、程序跳至断点处

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

- 步骤5 定位到问题程序段后，用户需要根据实际情况修改程序，避免异常出现。

6.6.2 因程序问题导致 PLC 失控的处理措施

常见的 PLC 应用程序问题如下：

- 应用程序中出现死循环或循环数过大。
- 应用程序访问空指针或指针越界。
- 应用程序调用底层功能块导致 runtime 崩溃。



因程序编写不当，下载该程序至 PLC，程序启动后，可能会出现 PLC 资源耗尽，Invtmatic Studio 软件对 PLC 无法控制的现象，导致扫描不到设备、ping 不通，或无法连接、在线或下载 PLC。恢复 PLC 至正常状态的方法如下：

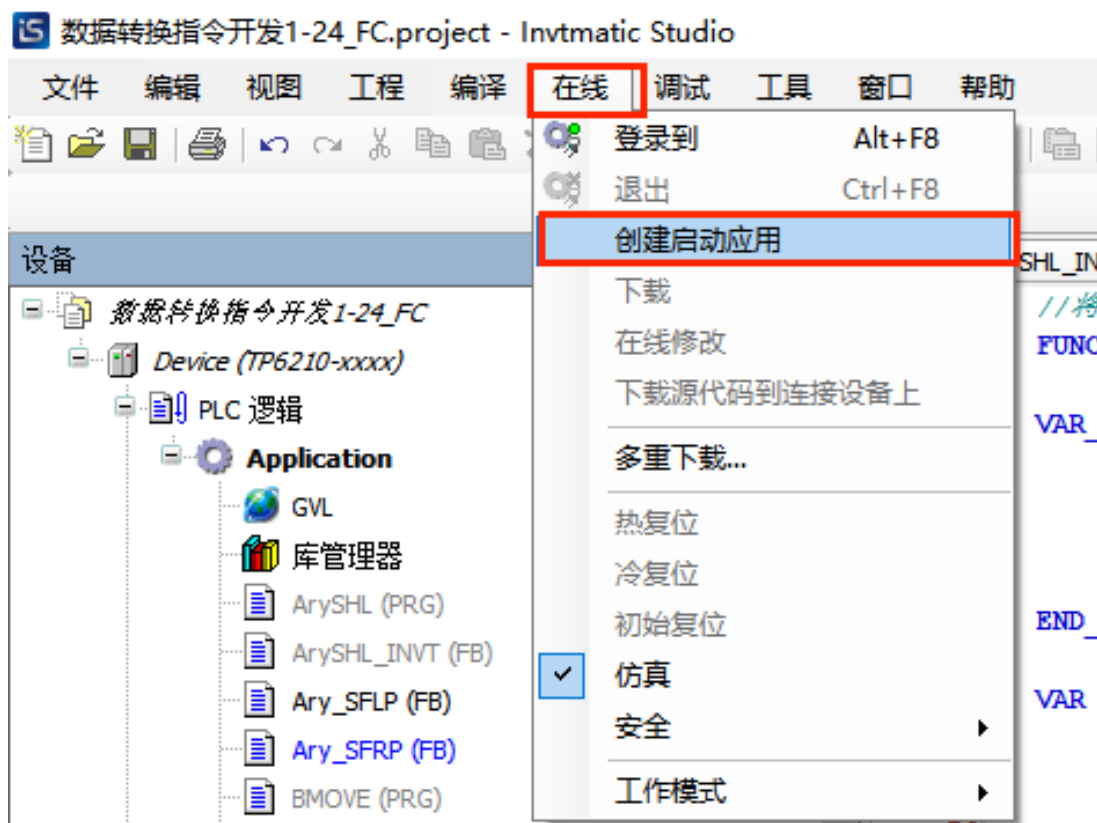
方法一：中型 PLC 将 RUN/STOP 拨到 STOP 状态，重新上电启动后，扫描到 PLC，下载一个正确的应用程序。

方法二：如果 PLC 网口还 ping 的通，可以在 Invtmatic Studio 软件上，菜单栏下**工具>Invtmatic Studio 工具**，在出厂设置界面，点击**一键复位**，清空 APP。

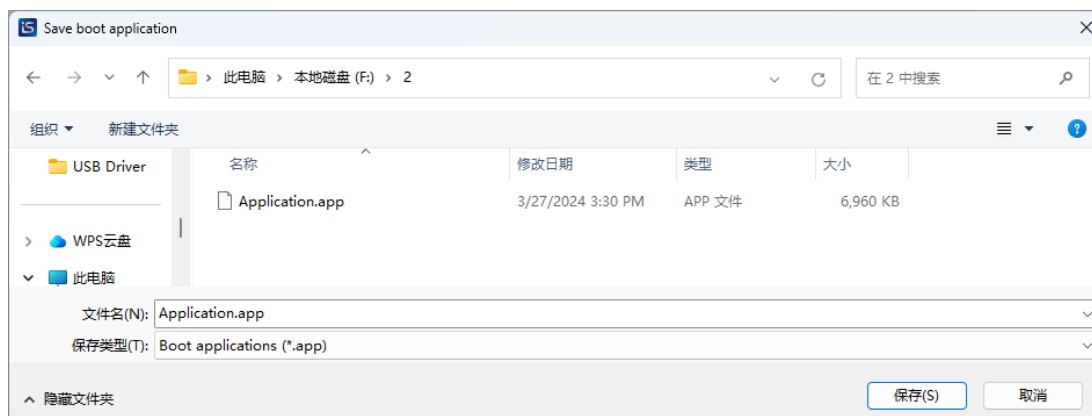
方法三：TP 系列大型 PLC 通过 U 盘下载一个正确 PLC 程序，可删除之前的应用程序。操作步骤如下：

**注意：**使用该功能时，U 盘仅可以有 1 个分区，且控制器仅可以插入 1 个 U 盘，U 盘根目录只能存在一份程序。

步骤1 在编程软件打开用户程序工程后，依次点击菜单栏中的**在线>创建启动应用**。



步骤2 选择启动程序 APP 存储路径，保持文件默认名称不变，单击**保存**。






步骤3 将用户启动程序文件“Application.app”和“Application.crc”拷贝至 U 盘根目录下。

 Application.app	7/25/2024 1:54 PM	APP 文件	1,914 KB
 Application.crc	7/25/2024 1:54 PM	CRC 文件	1 KB

步骤4 将 U 盘插入控制器任一 USB 接口中，给控制器断电重启，待控制器 **RUN** 指示灯常亮时，即表示用户程序加载成功。

 **注意：**采用上述方法恢复 PLC 后，需要排查应用程序错误后，再下载程序。

## 6.7 设备故障诊断代码

 **注意：**背板总线仅适用 TM 系列 PLC。

设备	错误类型	错误定位	主错误码	子错误码	错误说明
CPU	系统相关	硬件故障	0001	0001	纽扣电池未安装或电池电压过低
				0002	设备供电电压过低（小于 19V）
	系统组件相关	时钟系统组件故障	0008	0001	设置时间错误
				0002	写 RTC 时钟出错
				0003	读 RTC 时钟出错
		IP 系统组件故障	0009	0001	IP1 与 IP2 的 IP 网段重复报错
				XXX	预留
				0011	读:IP1 模块打开文件报错
				0012	读:IP1 模块无法获取 IP 信息
				0013	写:IP1 模块 IP 地址配置出错
				0014	写:IP1 模块掩码配置出错
				0015	写:IP1 模块网关配置出错
				0016	写:IP1 模块与 USB 网段重复报错
				0017	写:IP1 模块 IP 与网关不在同一网段报错
				XXX	预留
				0021	读:IP2 模块打开文件报错
				0022	读:IP2 模块无法获取 IP 信息
				0023	写:IP2 模块 IP 地址配置出错
				0024	写:IP2 模块掩码配置出错
				0025	写:IP2 模块网关配置出错
				0026	写:IP2 模块与 USB 网段重复报错
				0027	写:IP2 模块 IP 与网关不在同一网段报错
	背板总线相关	CPU IO 故障	0030	0001	模块组态故障
				0002	模块参数配置故障
		数字量故障	0031	0001	DI 模块组态故障
				0002	DI 模块参数配置故障
				XXX	预留
				2001	DO 模块组态故障
				2002	DO 模块参数配置故障
				2003	DO 模块输出端口供电故障
				2004	DO 模块输出故障
				XXX	预留
				XXX	预留



设备	错误类型	错误定位	主错误码	子错误码	错误说明
		模拟量故障	0032	0001	模块组态故障
				XXX	预留
				0012	AD 通道 0 参数配置故障
				0015	AD 通道 0 信号源开路故障
				0016	AD 通道 0 采样信号超极限故障
				0017	AD 通道 0 采样信号超量程上限故障
				0018	AD 通道 0 采样信号超量程下限故障
				XXX	预留
				0022	AD 通道 1 参数配置故障
				0025	AD 通道 1 信号源开路故障
				0026	AD 通道 1 采样信号超极限故障
				0027	AD 通道 1 采样信号超量程上限故障
				0028	AD 通道 1 采样信号超量程下限故障
				XXX	预留
				0032	AD 通道 2 参数配置故障
				0035	AD 通道 2 信号源开路故障
				0036	AD 通道 2 采样信号超极限故障
				0037	AD 通道 2 采样信号超量程上限故障
				0038	AD 通道 2 采样信号超量程下限故障
				XXX	预留
				0042	AD 通道 3 参数配置故障
				0045	AD 通道 3 信号源开路故障
				0046	AD 通道 3 采样信号超极限故障
				0047	AD 通道 3 采样信号超量程上限故障
				0048	AD 通道 3 采样信号超量程下限故障
				XXX	预留
				0003	模块输出端口供电故障
				XXX	预留
				2012	通道 0 参数配置故障
				2014	通道 0 输出故障
				XXX	预留
				2022	通道 1 参数配置故障
				2024	通道 1 输出故障
				XXX	预留
				2032	通道 2 参数配置故障
				2034	通道 2 输出故障
				XXX	预留
				2042	通道 3 参数配置故障
				2044	通道 3 输出故障
				XXX	预留
		测温模块故障	0033	0001	模块组态故障
				XXX	预留
				0012	通道 0 参数配置故障
				0015	通道 0 信号源开路故障
				0017	通道 0 采样信号超量程上限故障
				0018	通道 0 采样信号超量程下限故障
				XXX	预留



设备	错误类型	错误定位	主错误码	子错误码	错误说明
				0022	通道 1 参数配置故障
				0025	通道 1 信号源开路故障
				0027	通道 1 采样信号超量程上限故障
				0028	通道 1 采样信号超量程下限故障
				XXX	预留
				0032	通道 2 参数配置故障
				0035	通道 2 信号源开路故障
				0037	通道 2 采样信号超量程上限故障
				0038	通道 2 采样信号超量程下限故障
				XXX	预留
				0042	通道 3 参数配置故障
				0045	通道 3 信号源开路故障
				0047	通道 3 采样信号超量程上限故障
				0048	通道 3 采样信号超量程下限故障
现场总线	Modbus 相关	Modbus_RTU Master1	0040	0001	非法功能码
				0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时，通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定 或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败
		Modbus_RTU Master2	0041	0001	非法功能码
				0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时，通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定 或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败
		Modbus_RTU	0042	0001	非法功能码



设备	错误类型	错误定位	主错误码	子错误码	错误说明
		Slave1		0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时，通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败
		Modbus_RTU Slave2	0043	0001	非法功能码
				0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时：通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败
	Modbus TCP 相关	Modbus TCP Master1	00A0	0001	非法功能码
				0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时：通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定 或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败
		Modbus TCP	00A1	0001	非法功能码



设备	错误类型	错误定位	主错误码	子错误码	错误说明
		Master2		0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时，通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败
		Modbus TCP Slave1	00A2	0001	非法功能码
				0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时：通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败
		Modbus TCP Slave2	00A3	0001	非法功能码
				0002	非法地址
				0003	数据个数错误
				0004	从站设备故障
				0005	通讯超时：通讯时间超过了用户设定的通讯最大时间发生了错误
				XXX	预留
				0008	接收到的数据帧不符合 Modbus 协议
				0009	CRC/LRC 校验错误
				XXX	预留
				000B	接收到的数据长度不符合协议规定或者数量超过该功能码规定的最大限制
				000C	接收到的从站地址和请求的从站地址不匹配
				000D	接收到的功能码和请求的功能码不匹配
				000E	命令执行失败

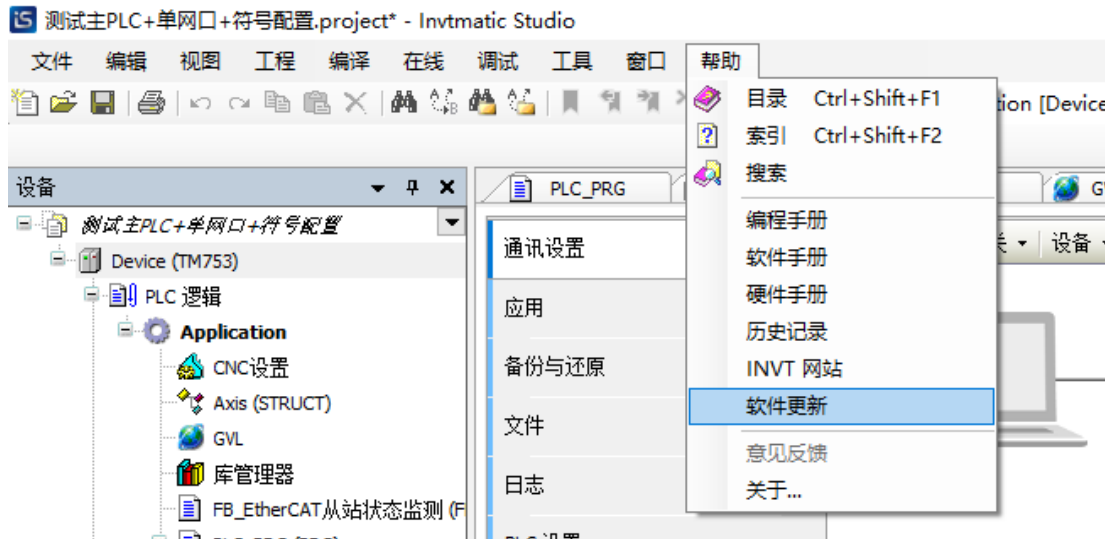


# 7 升级与 PLC 设置

## 7.1 软件升级

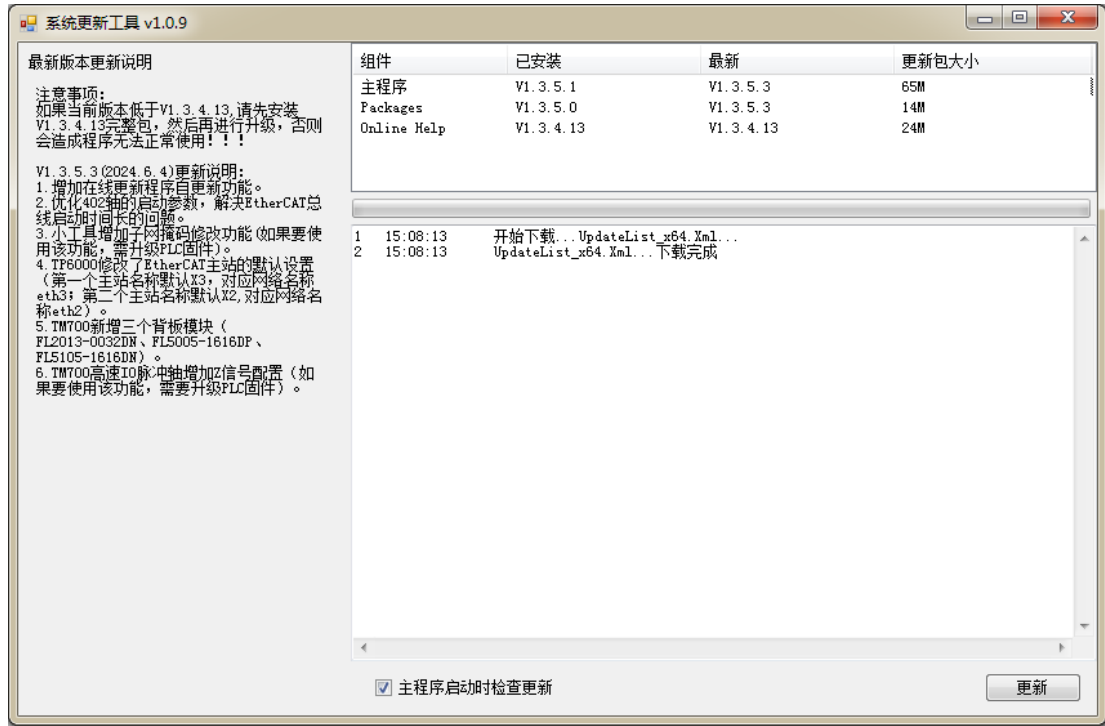
打开 Invtmatic Studio 软件后，点击菜单中的**帮助>软件更新**，可实现软件在线升级。

图 7-1 软件更新界面



如果有新版本，打开软件也会自动提示更新，下载完成后，点击**更新**即可。


图 7-2 下载和安装升级包





## 7.2 固件升级

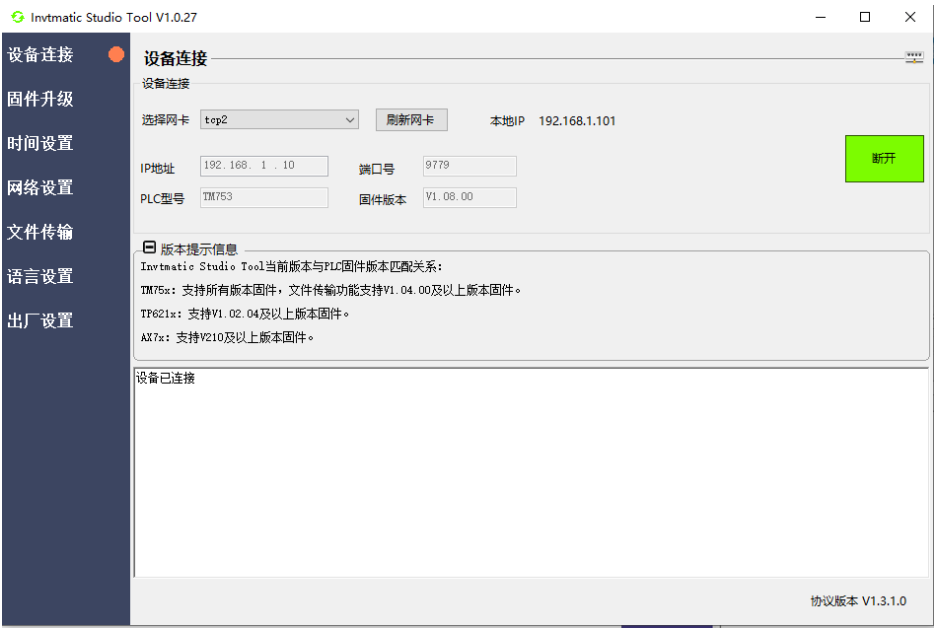
TM/TP 系列支持界面通过网口升级固件方式，在菜单界面点击**工具>Invtmatic studio 工具**。

 **注意：**Invtmatic studio 工具可以作为独立软件运行，不需要打开 Invtmatic studio 编程软件，方便使用。

### ■ 通讯连接

TM 系列有 Ethernet1、Ethernet2 和 TypeC 支持网络连接，其中 Ethernet1 支持广播通讯连接（一键扫描 IP 并自动连接）。TP 系列 X0 和 X1 均支持广播通讯网络连接。

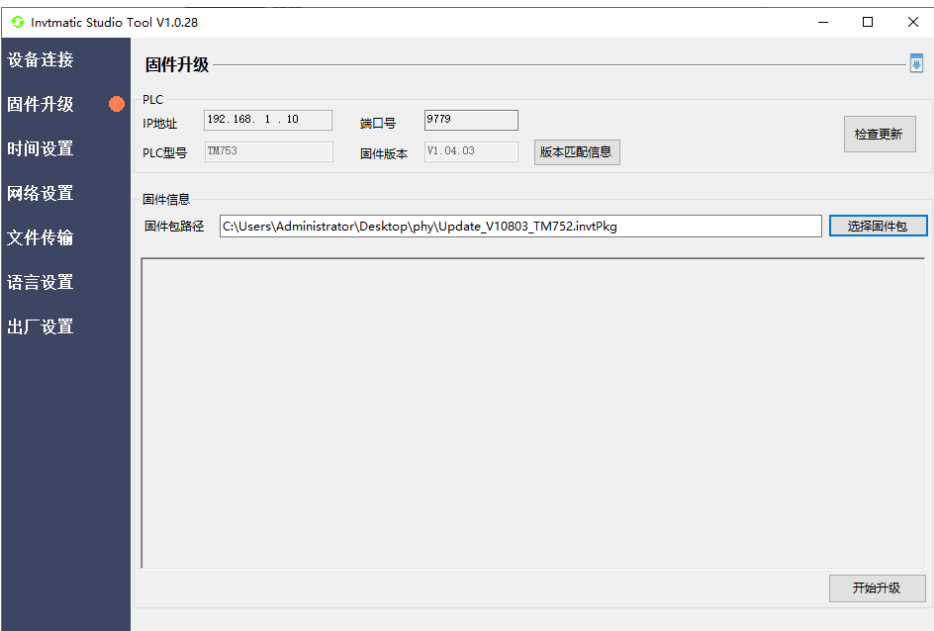
图 7-3 工具连接 PLC 界面



### ■ 固件升级

可查看版本号或选择升级包升级。

图 7-4 固件升级界面

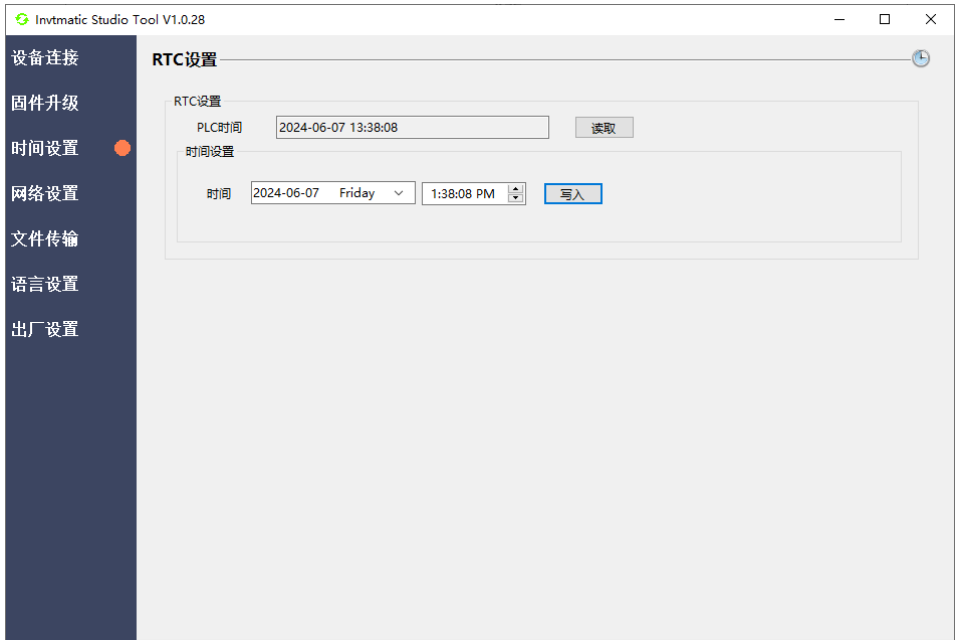




### 7.3 时间设置

可读取或修改系统时间。

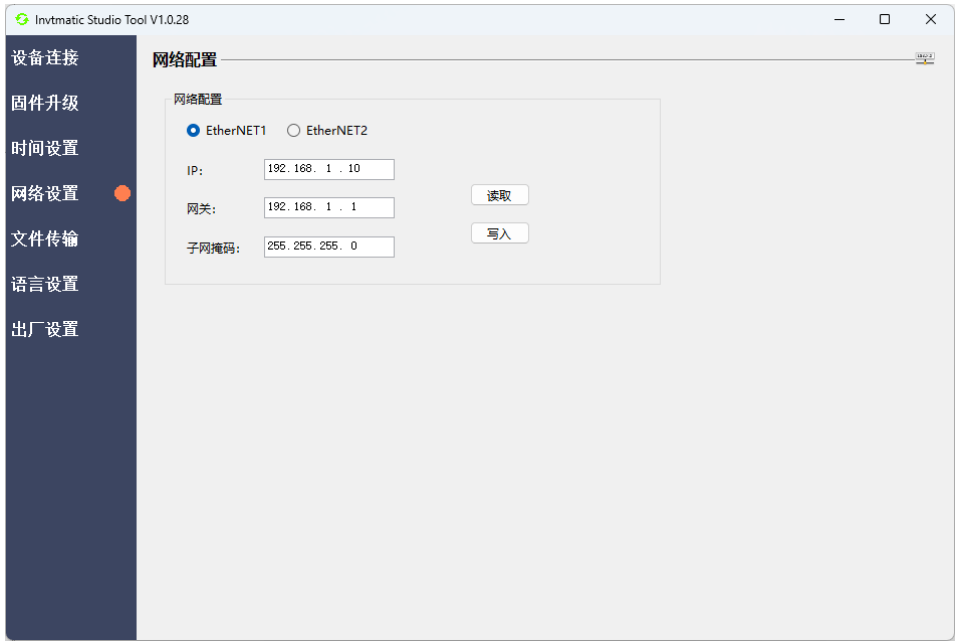
图 7-5 读取或写入系统时间界面



### 7.4 网络设置

网络设置可读取或修改网口 1、网口 2 的 IP 地址及子网掩码。

图 7-6 读取或修改 IP 地址





## 7.5 恢复出厂设置

电脑连接 PLC 的 Ethernet1，不需要知道 PLC 的 IP 地址，可一键复位 PLC 到出厂设置，并清空 APP。如果网络上连接了多台 PLC，需要先断开其它 PLC，仅连接需要恢复出厂的设备。

图 7-7 恢复 IP 地址或清空程序



## 7.6 文件传输

左侧为本地电脑文件目录，右边为 PLC 文件目录。通常用于用户拷贝工艺文件，比如用户要拷贝 CNC 文件到 PLC，目录选择 PlcLogic/\_cnc。

图 7-8 文件传输





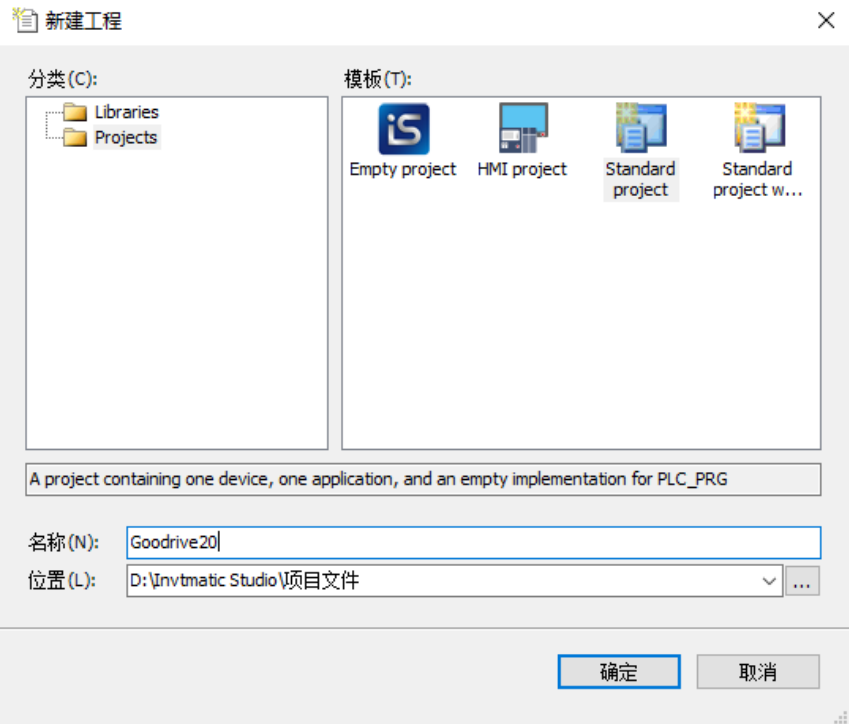
# 附录A 工程实例

## A.1 控制器与 Goodrive20 系列变频器 485 通讯配置实例

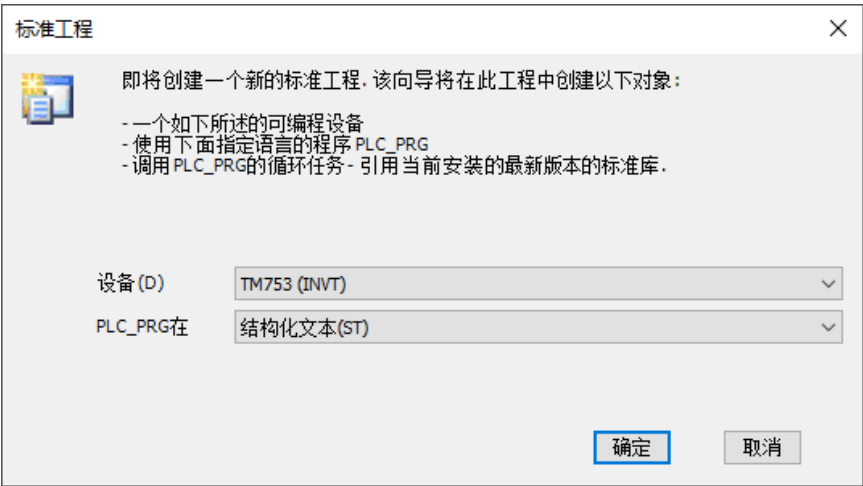
将 TM 系列控制器设为主机，将一台 Goodrive20 系列变频器设为从机，控制器使用 Modbus RTU 通讯协议，其物理层为两线制 RS485，通过 COM2 口与变频器通信。通过编写一个小程序，使用上位机对 Goodrive20 变频器的功能参数进行读写。PLC 硬件 A1 端口接变频器 485+，B1 端口接变频器 485-。操作步骤如下：



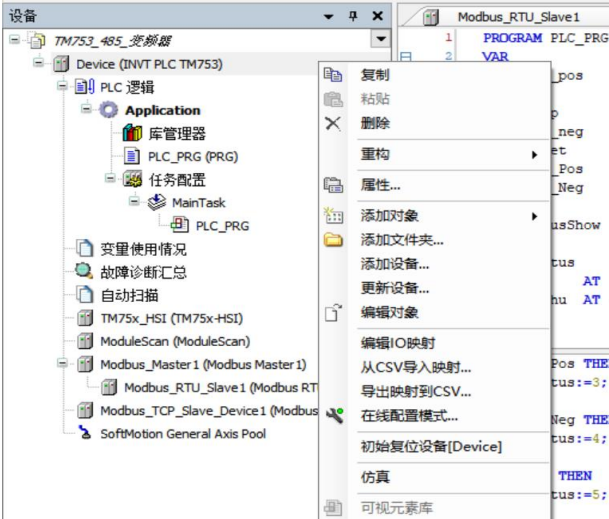
步骤1 新建一个工程，选择菜单 **Projects>Standard project**，此时已新建一个标准工程，设备为 TM753，编程语言为结构化文本（ST），根据实际需要，编辑工程信息。





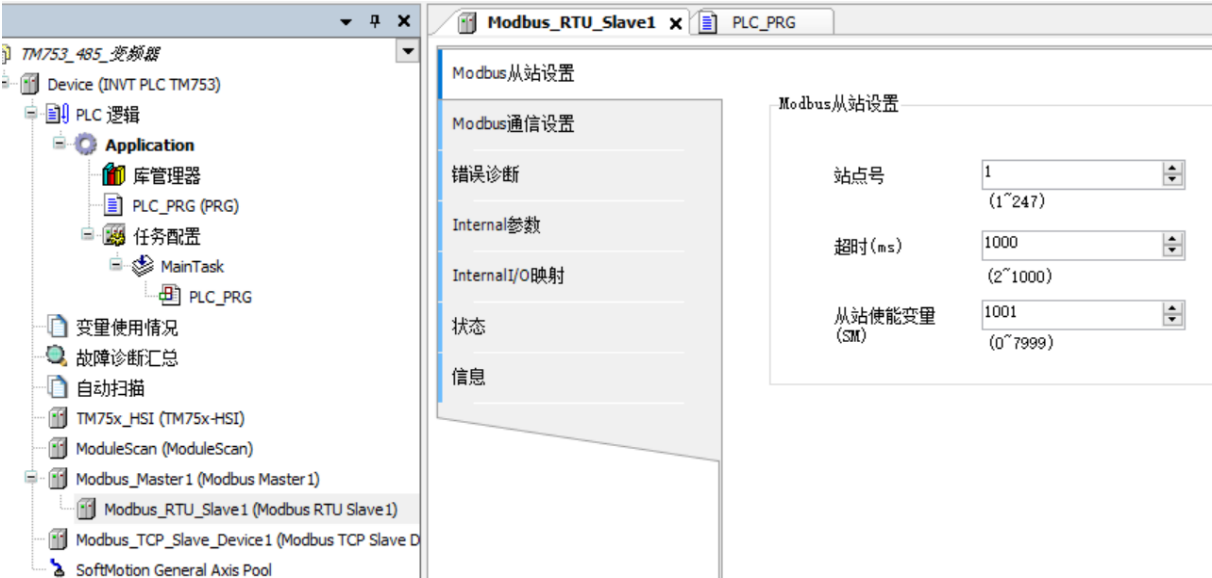


步骤2 参照 5.2.1 Modbus RTU 主站配置添加 **Modbus\_Master1** 和 **Modbus\_RTU\_Slave1**，并设置通信参数，将 PLC 设置成 Modbus\_TCP 从站，变量可显示到触摸屏上观察。





步骤3 将波特率设置为 19200，数据位设置为 8，停止位设置为 1，校验位设置为 EVEN（偶检验），超时时间设置为 1000ms；变频器站点号（从机地址）设置为 1，从站使能变量设置为 1001。



步骤4 在 Modbus 通信设置窗口，添加 Modbus 从站通信配置。



步骤5 双击 PLC\_PRG，在声明编辑器上输入以下代码，变量绑定地址，方便与触摸屏通讯。

```
PROGRAM PLC_PRG

VAR

    brun_pos      AT %MX0.0    :BOOL;//正转

    bSet           AT %MX0.1    :BOOL;//频率设定启动

    bStop         AT %MX0.2    :BOOL;//停止

    brun_neg      AT %MX0.3    :BOOL;//反转

    breset        AT %MX0.4    :BOOL;//故障复位

    bJog_Pos      AT %MX0.5    :BOOL;//正转点动

    bJog_Neg      AT %MX0.6    :BOOL;//反转点动


    statusShow    AT %MW100    :INT;//运行状态显示

    fre           :INT;//频率

    bstatus       :INT;//运行状态
```

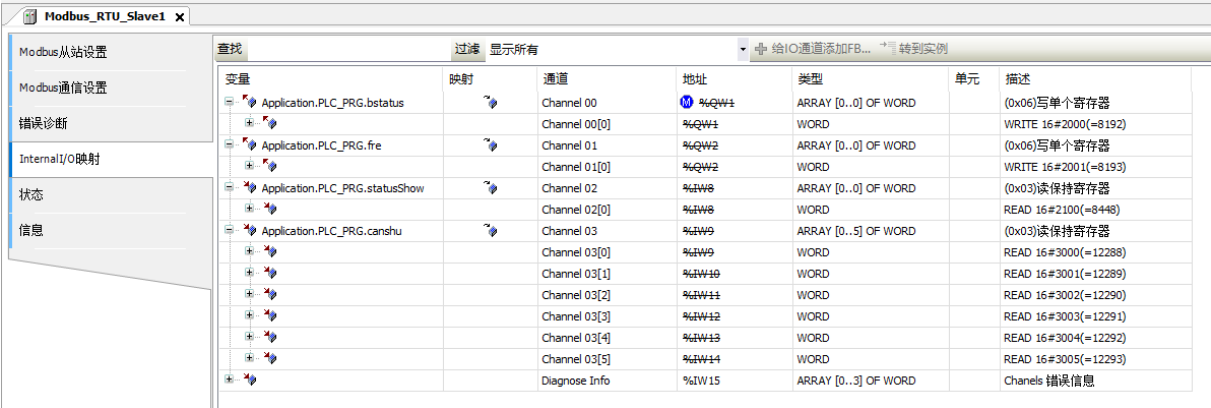


```
hfre      AT %MW10      :INT;//触摸屏频率值设定

canshu    AT %MW200     : ARRAY[1..6] OF INT;

END_VAR
```

在 Modbus 从站通信设置窗口，映射程序中的控制变量：



Channel03 中，从 16#3000 读取 6 个地址，并映射到 canshu 数组变量，在数组 canshu[2]即 3002H 地址中，如果读到值 3335，参考变频器产品手册，表示母线电压为 333.5V。

Channel02 中，从 16#2100 读取 1 个地址，并映射到 status Show 变量，如果读到值 3，参考变频器产品手册，表示变频器停机中。

在主体代码编辑器里输入以下代码：

```
SM1001:=TRUE;//从站使能变量

IF brun_pos THEN

    bstatus:=1;

END_IF

IF brun_neg THEN

    bstatus:=2;

END_IF

IF bJog_Pos THEN

    bstatus:=3;

END_IF

IF bJog_Neg THEN

    bstatus:=4;

END_IF

IF bStop THEN

    bstatus:=5;



END_IF

IF bSet THEN
```



```
fre:=hfre;  
  
END_IF
```

将变频器通过两线制 RS485 和控制器相连接后，启动变频器。通过变频器键盘设置功能码 P00.01 为 2，使其运行命令可由上位机通过通讯方式进行控制；设置 P00.06 为 8，即选择 Modbus 通讯方式；设置 P14 组的串行通讯参数，使其与上位机的初始化设定参数包括波特率、数据位、校验位、从机地址、超时时间等一致。

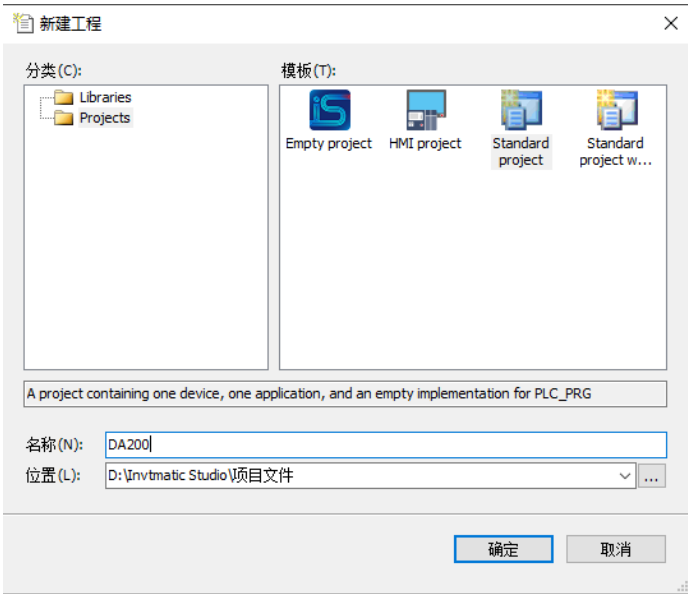
点击工具栏的  图标编译代码。编译没有错误后，点击工具栏的  按钮登录控制器，确保控制器数码管无报错，变频器 Goodrive20 产品与控制器顺利连接，通讯正常，触摸屏界面如下图所示。



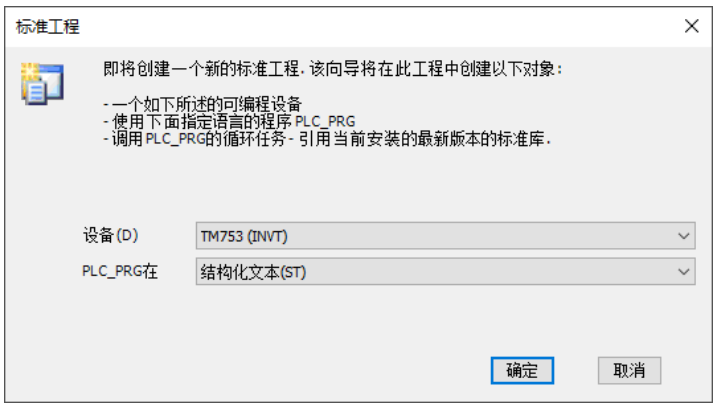
## A.2 控制器与 DA200 系列伺服驱动器配置实例

示例：通过编写一个小程序，控制 4 台 DA200 系列的伺服驱动器，驱动 4 台电机轴做匀速正反转运动。操作步骤如下：

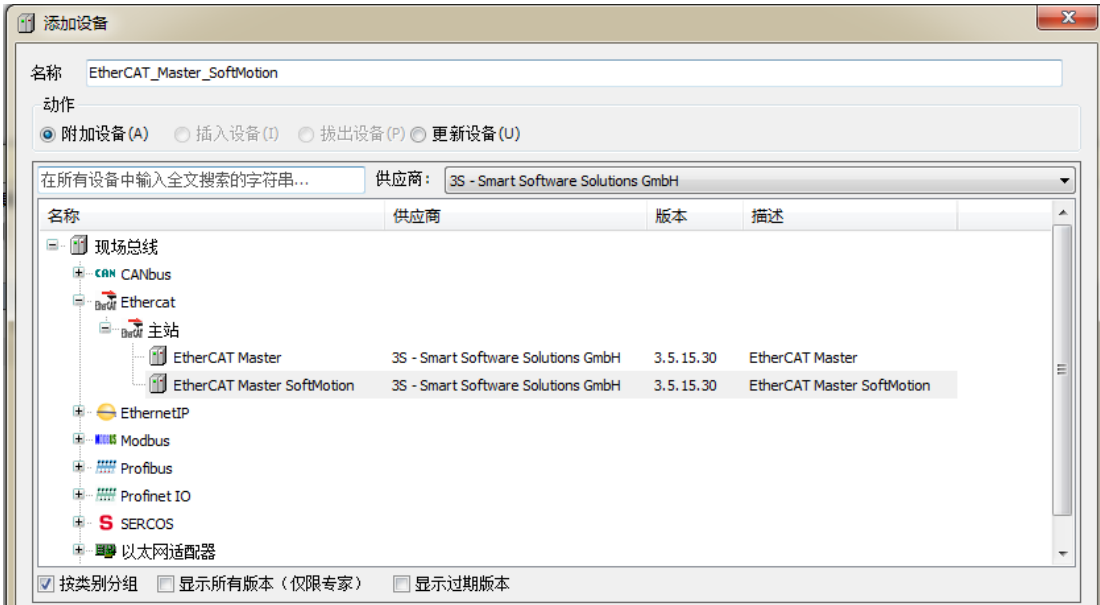
步骤1 新建一个工程，选择菜单 **Projects>Standard project**，此时已新建一个标准工程，设备为 TM753，编程语言为结构化文本（ST），根据实际需要，编辑工程信息。







步骤2 右键点击设备树中 **Device**, 选择**添加设备**, 添加 EtherCAT 主站设备, 此处选择 **EtherCAT Master SoftMotion, 版本 3.5.15.30**。

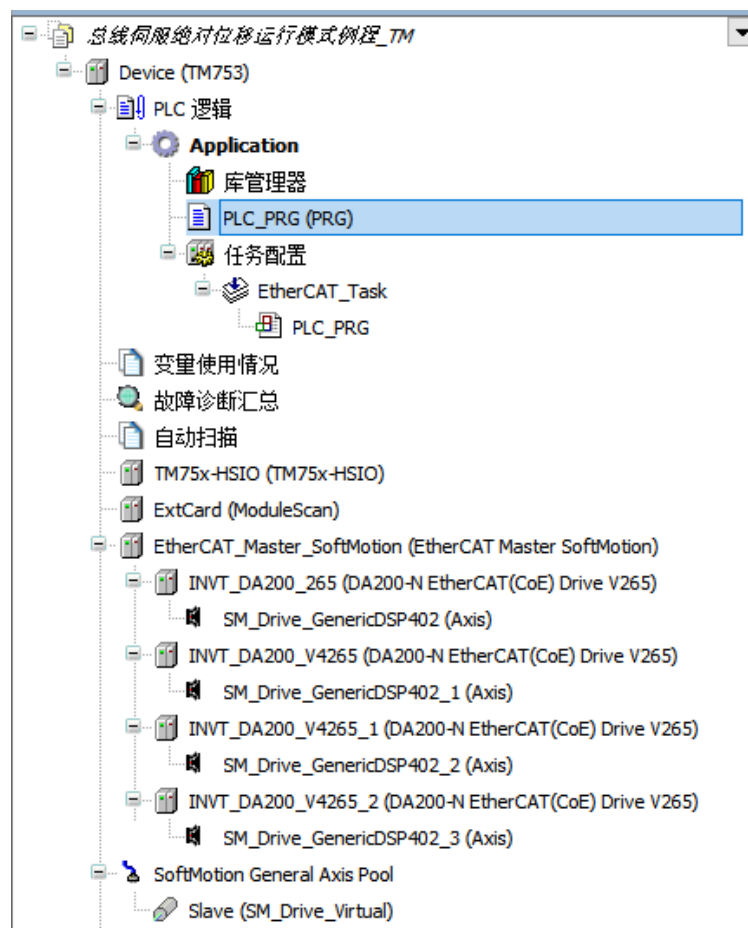




步骤3 右键点击设备树中的 **EtherCAT Master SoftMotion**，选择**添加设备**，添加 4 台伺服驱动器，此处选择 **DA200-N**。



步骤4 依次右键点击设备树中 **INVT\_DA200** 设备，选择添加 **SoftMotion** 的 **CiA402** 轴，完成 4 台伺服电机的添加。



步骤5 双击 **PLC\_PRG**，在声明编辑器上输入以下代码。

```
PROGRAM PLC_PRG

VAR

    iStatus: INT;

    MC_Power_0: MC_Power;
```



```

MC_Power_1: MC_Power;

MC_Power_2: MC_Power;

MC_Power_3: MC_Power;

MC_MoveAbsolute_0: MC_MoveAbsolute;

MC_MoveAbsolute_1: MC_MoveAbsolute;

MC_MoveAbsolute_2: MC_MoveAbsolute;

MC_MoveAbsolute_3: MC_MoveAbsolute;

END_VAR

```

**步骤6** 在主体代码编辑器里输入以下代码：

```

CASE iStatus OF

0:

MC_Power_0(Axis:= SM_Drive_GenericDSP402, Enable:= TRUE, bRegulatorOn:= TRUE,
bDriveStart:=TRUE , );

MC_Power_1(Axis:= SM_Drive_GenericDSP402_1, Enable:= TRUE, bRegulatorOn:= TRUE,
bDriveStart:=TRUE , );

MC_Power_2(Axis:= SM_Drive_GenericDSP402_2, Enable:= TRUE, bRegulatorOn:= TRUE,
bDriveStart:=TRUE , );

MC_Power_3(Axis:= SM_Drive_GenericDSP402_3, Enable:= TRUE, bRegulatorOn:= TRUE,
bDriveStart:=TRUE , );

IF MC_Power_0.Status AND MC_Power_1.Status AND MC_Power_2.Status AND
MC_Power_3.Status THEN

iStatus:=iStatus+1;

END_IF

1:

MC_MoveAbsolute_0(Axis:=SM_Drive_GenericDSP402 , Execute:= TRUE, Position:=50 ,
Velocity:=3 , Acceleration:= 2, Deceleration:= 100,);

MC_MoveAbsolute_1(Axis:=SM_Drive_GenericDSP402_1, Execute:= TRUE,
Position:=50 , Velocity:=3 , Acceleration:= 2, Deceleration:=100,);

MC_MoveAbsolute_2(Axis:=SM_Drive_GenericDSP402_2, Execute:= TRUE,
Position:=50 , Velocity:=3 , Acceleration:= 2, Deceleration:=100,);

MC_MoveAbsolute_3(Axis:=SM_Drive_GenericDSP402_3, Execute:= TRUE,
Position:=50 , Velocity:=3 , Acceleration:= 2, Deceleration:=100,);

IF MC_MoveAbsolute_0.Done AND MC_MoveAbsolute_1.Done AND
MC_MoveAbsolute_2.Done AND MC_MoveAbsolute_3.Done THEN

MC_MoveAbsolute_0(Axis:=SM_Drive_GenericDSP402 , Execute:= FALSE,);

MC_MoveAbsolute_1(Axis:=SM_Drive_GenericDSP402_1 , Execute:= FALSE,);

MC_MoveAbsolute_2(Axis:=SM_Drive_GenericDSP402_2 , Execute:=
FALSE,);

```



```

        MC_MoveAbsolute_3(Axis:=SM_Drive_GenericDSP402_3 ,      Execute:=
FALSE,);

        iStatus:=iStatus+1;

END_IF

2:

MC_MoveAbsolute_0(Axis:=SM_Drive_GenericDSP402 , Execute:= TRUE, Position:=0 ,
Velocity:=3, Acceleration:= 2, Deceleration:= 100,);

MC_MoveAbsolute_1(Axis:=SM_Drive_GenericDSP402_1, Execute:= TRUE, Position:=0 ,
Velocity:=3 , Acceleration:= 2, Deceleration:=100,);

MC_MoveAbsolute_2(Axis:=SM_Drive_GenericDSP402_2, Execute:= TRUE, Position:=0 ,
Velocity:=3, Acceleration:= 2, Deceleration:=100,);

MC_MoveAbsolute_3(Axis:=SM_Drive_GenericDSP402_3, Execute:= TRUE, Position:=0 ,
Velocity:=3 , Acceleration:= 2, Deceleration:=100,);

IF      MC_MoveAbsolute_0.Done      AND      MC_MoveAbsolute_1.Done      AND
MC_MoveAbsolute_2.Done AND MC_MoveAbsolute_3.Done THEN

        MC_MoveAbsolute_0(Axis:=SM_Drive_GenericDSP402 , Execute:= FALSE,);

        MC_MoveAbsolute_1(Axis:=SM_Drive_GenericDSP402_1 ,      Execute:=
FALSE,);

        MC_MoveAbsolute_2(Axis:=SM_Drive_GenericDSP402_2 ,      Execute:=
FALSE,);

        MC_MoveAbsolute_3(Axis:=SM_Drive_GenericDSP402_3 ,      Execute:=
FALSE,);

        iStatus:=1;

END_IF

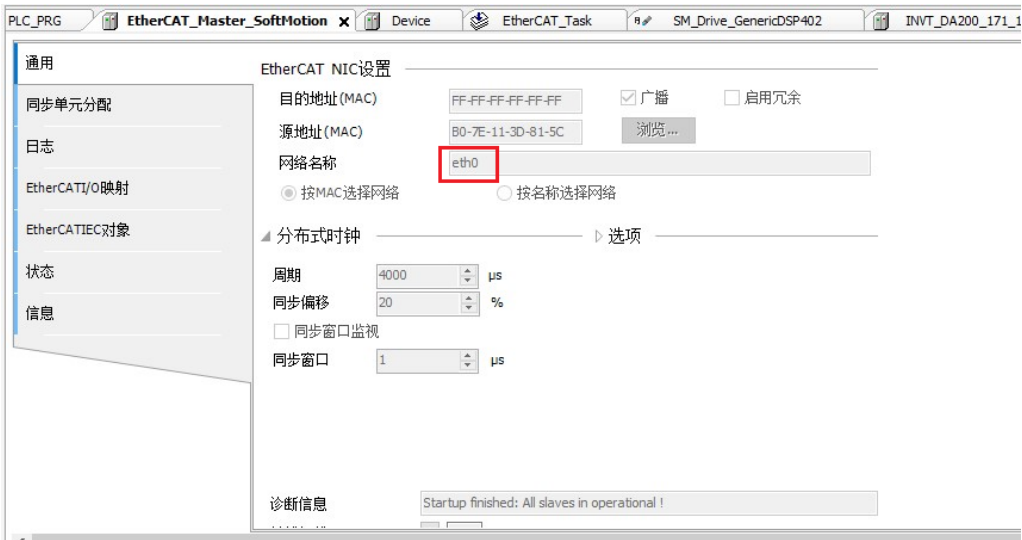
END_CASE



```

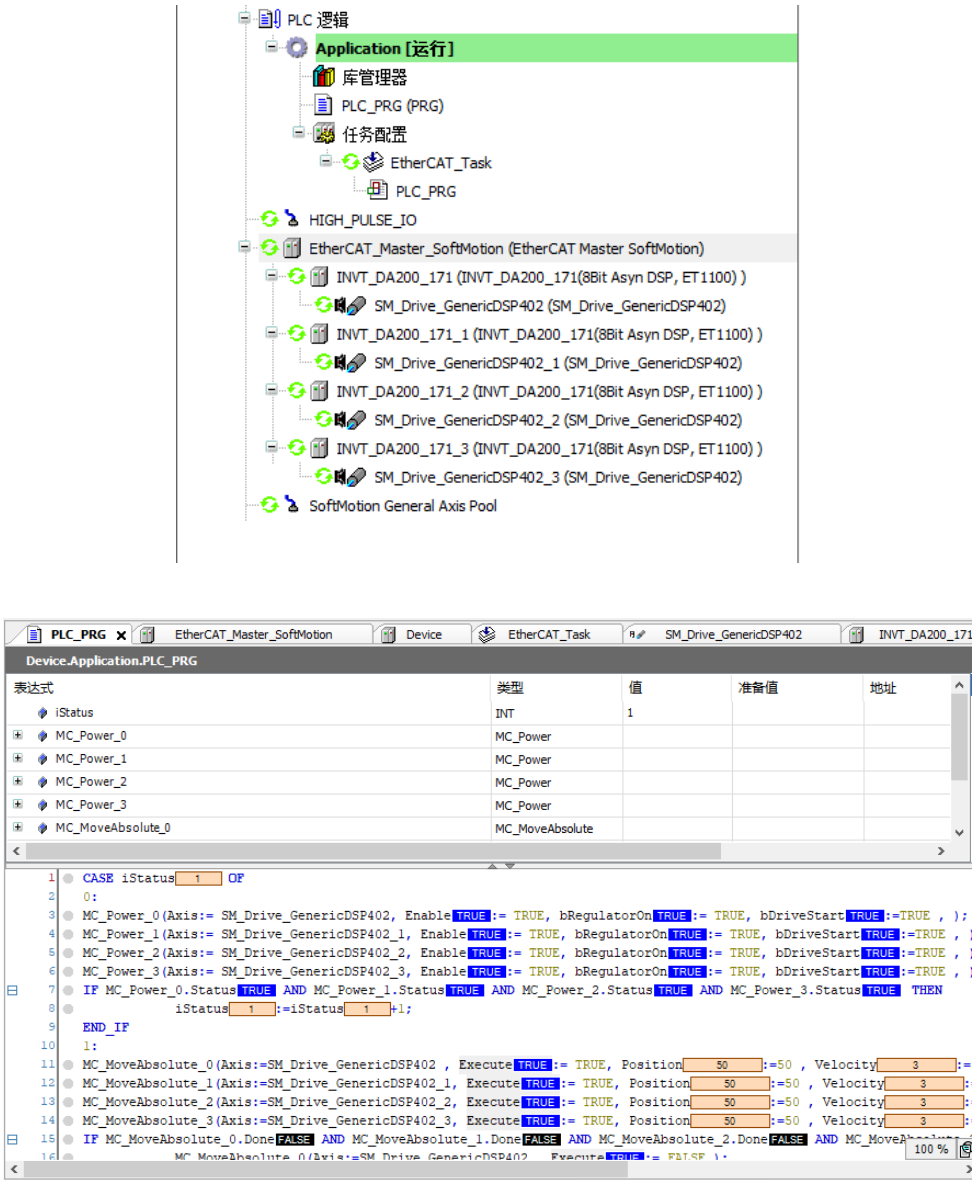
程序主体采用状态机的形式，通过判断 iStatus 的值来决定执行哪部分的代码。程序启动时，iStatus 值为 0，程序初始化 MC\_Power 功能块，使能相应的电机轴，如果使能成功，则 iStatus 值为 1，进入下一个状态。iStatus 值为 1 时，执行 MC\_MoveAbsolute 功能块，电机以指定的速度转动到指定的位置。若电机正常运动到指定的位置，则 iStatus 值加 1，进入下一个状态。iStatus 值为 2 时，继续执行另一个方向的 MC\_MoveAbsolute 功能块，电机继续以该功能块指定的速度转到指定的位置。若电机正常运动到指定的位置，则 iStatus 值置为 1。如此反复，实现电机的正反转运动。

步骤7 双击设备树中的 EtherCAT 主站设备 **EtherCAT\_Master\_SoftMotion**，点击**浏览**，选择相应的 EtherCAT 通信网口，**网络名称**处选择 **eth0**。根据需要选择分布式时钟，这里选择循环时间为 4000μs。



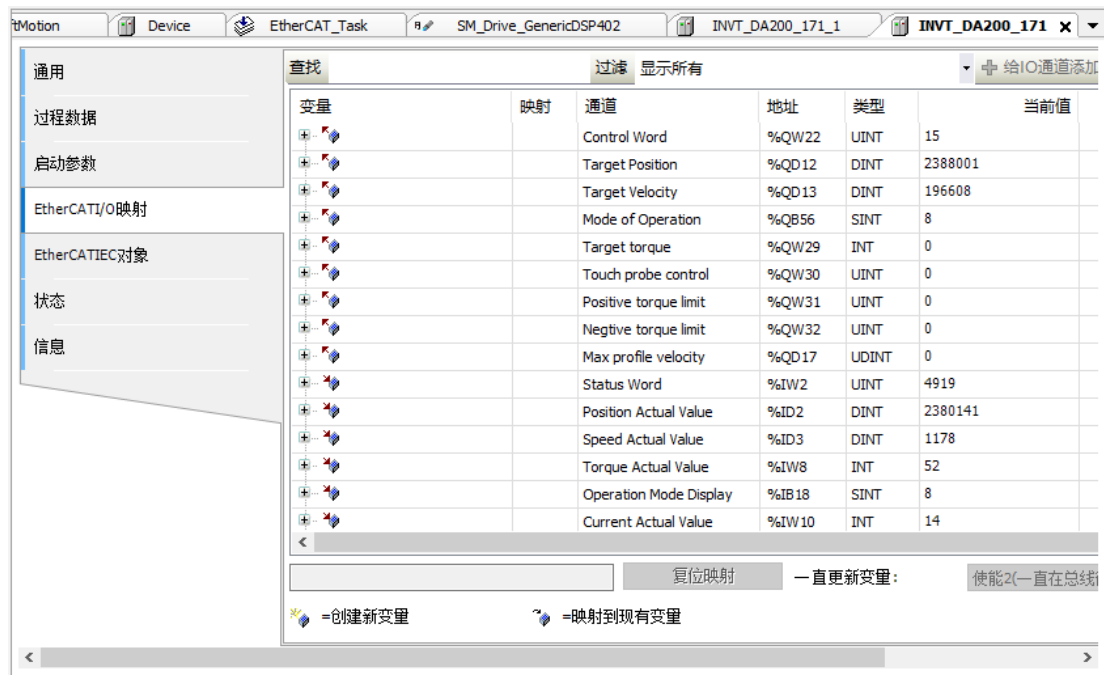



步骤8 点击工具栏的  编译代码。编译没有错误后，点击工具栏的  登录控制器。伺服正常启动，电机顺利运行，上位机界面如下图所示。

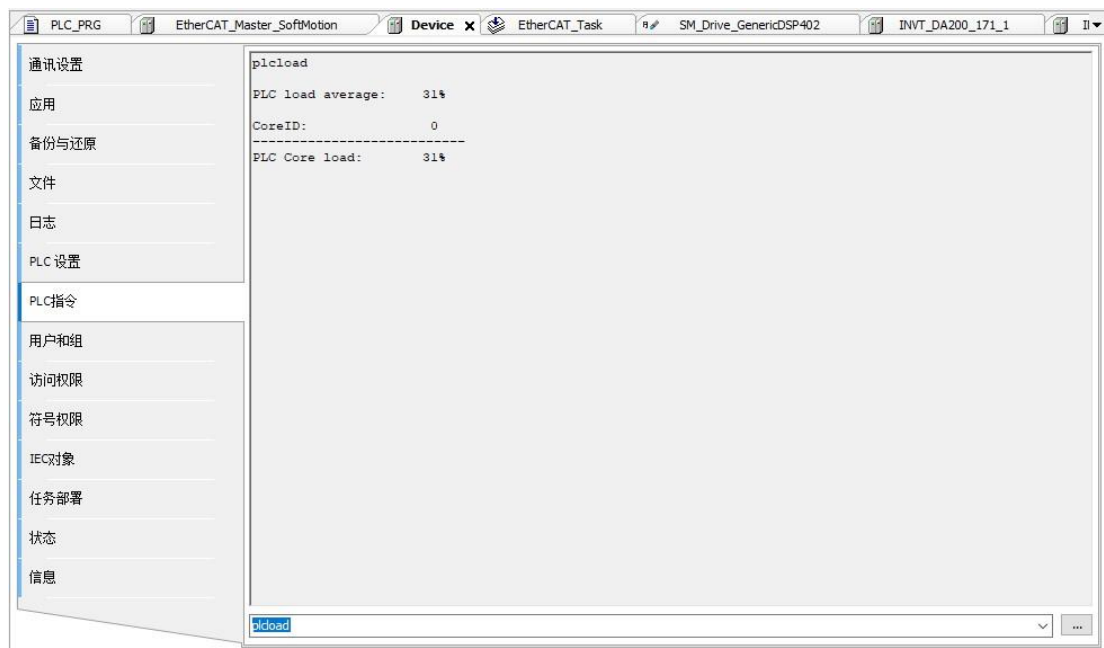




步骤9 双击设备树中的 **INVT\_DA200**，在 I/O 映射界面可查看或设置当前电机的运行参数。

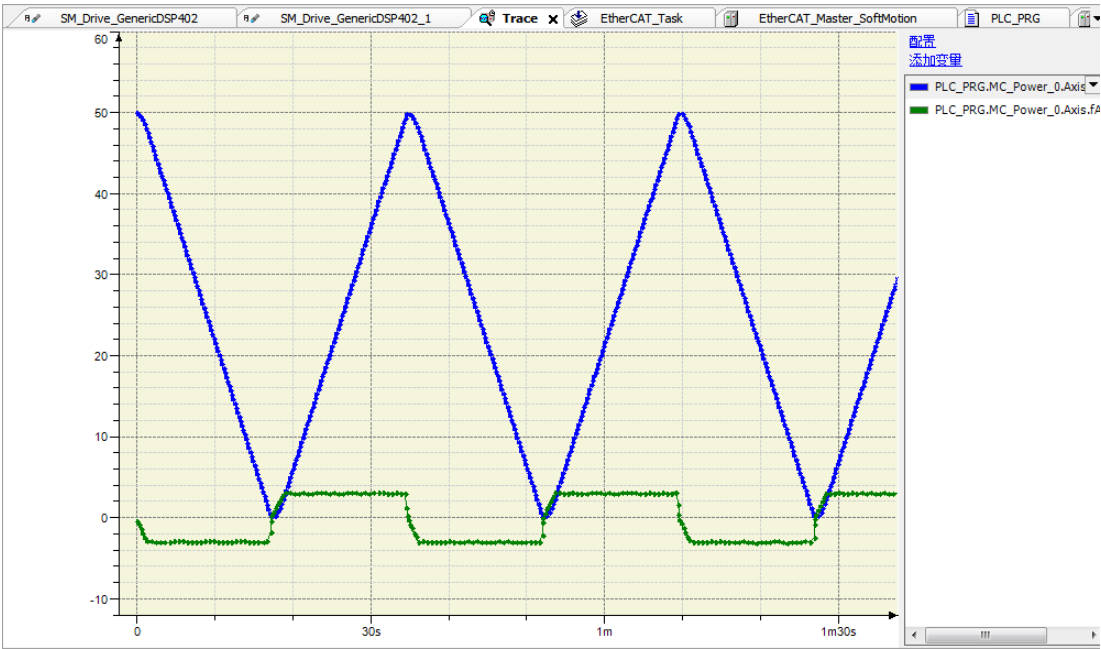


步骤10 在登录状态，选择 **Device>PLC 指令**，在右下角找到 ，栏中选择 **plcload**，执行后可以查看当前控制器的 CPU 负载率。



为了更直观地观察电机轴的运行情况，跟踪轴的实际位置，我们新建一个 trace。右键点击 **Application**，选择**添加对象>跟踪**，设置任务属性为 **EtherCAT\_Task**，在 Trace 里添加 **PLC\_PRG.MC\_Power\_0.Axis.fActPosition** 和 **PLC\_PRG.MC\_Power\_0.Axis.fActVelocity** 变量，适当调整坐标的显示属性，可以跟踪电机的实际位置和实际速度。

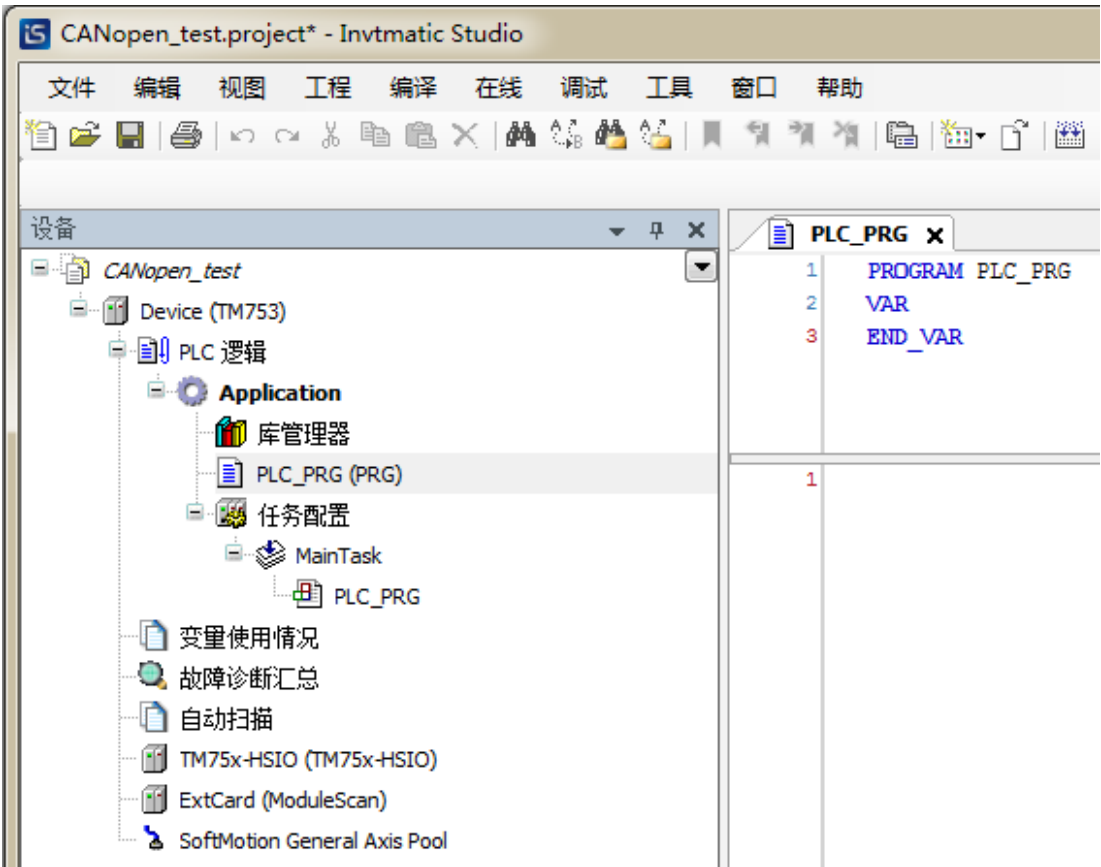




### A.3 控制器与 DA200 系列伺服 CANopen 配置实例

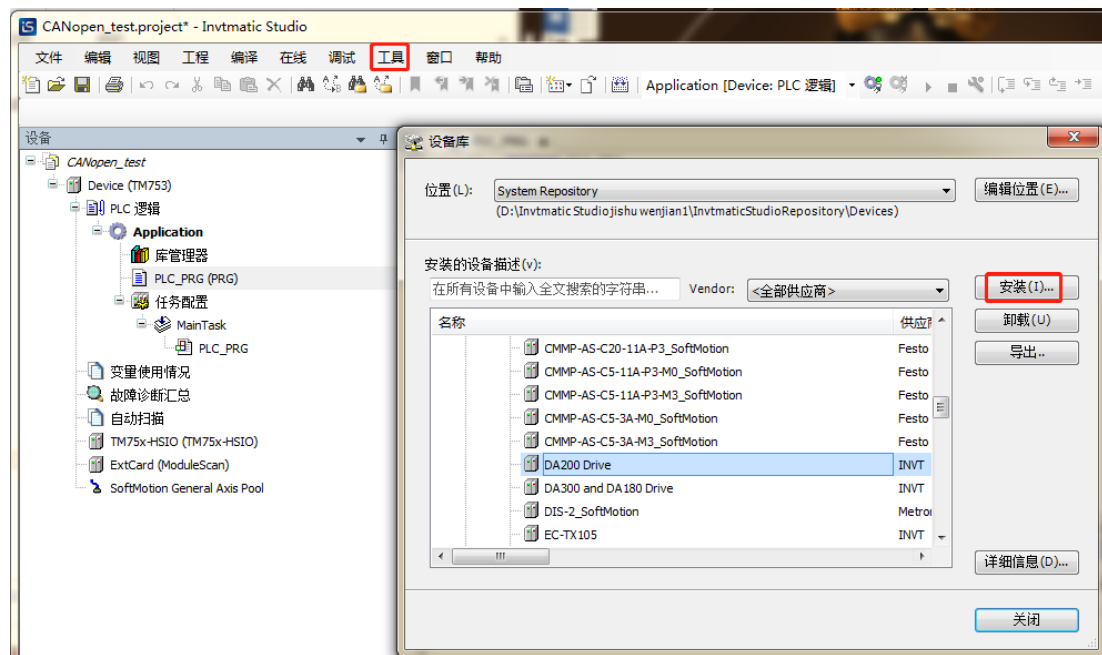
通过编写一个小程序，实现 CANopen 通讯连接 DA200 系列的伺服驱动器。操作步骤如下：

步骤1 参考 2.1.2 新建工程章节，创建如下工程。

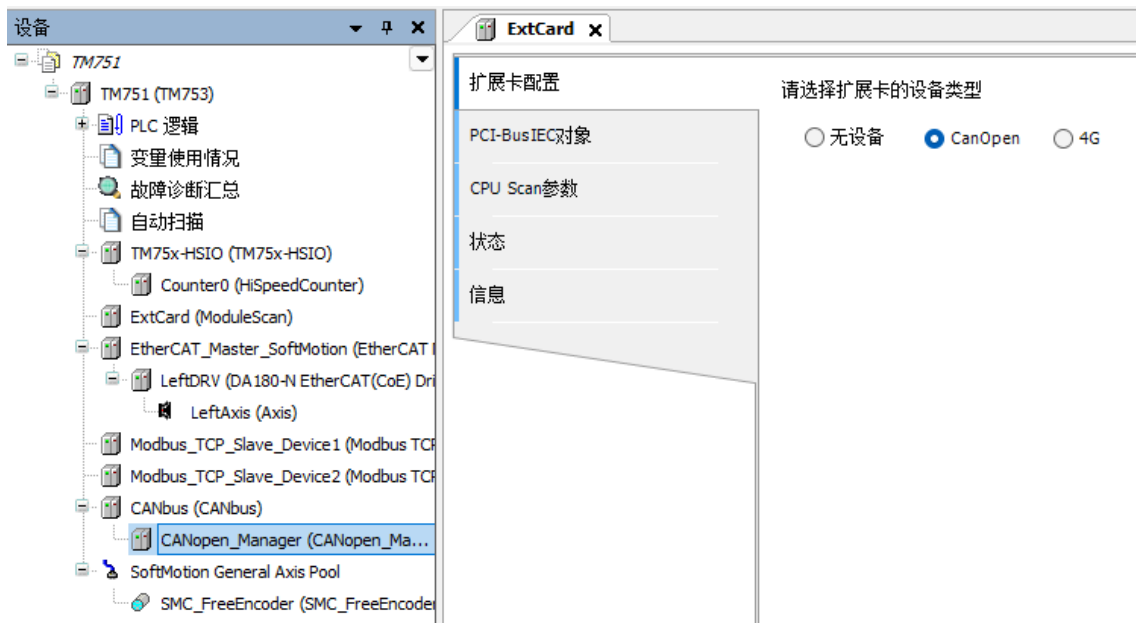




步骤2 在菜单栏中点击**工具>设备库**后，点击**安装**，找到 **INVT\_DA200\_CANopen.eds** 设备描述文件，点击打开，此时 DA200 CANopen 设备描述文件添加成功。

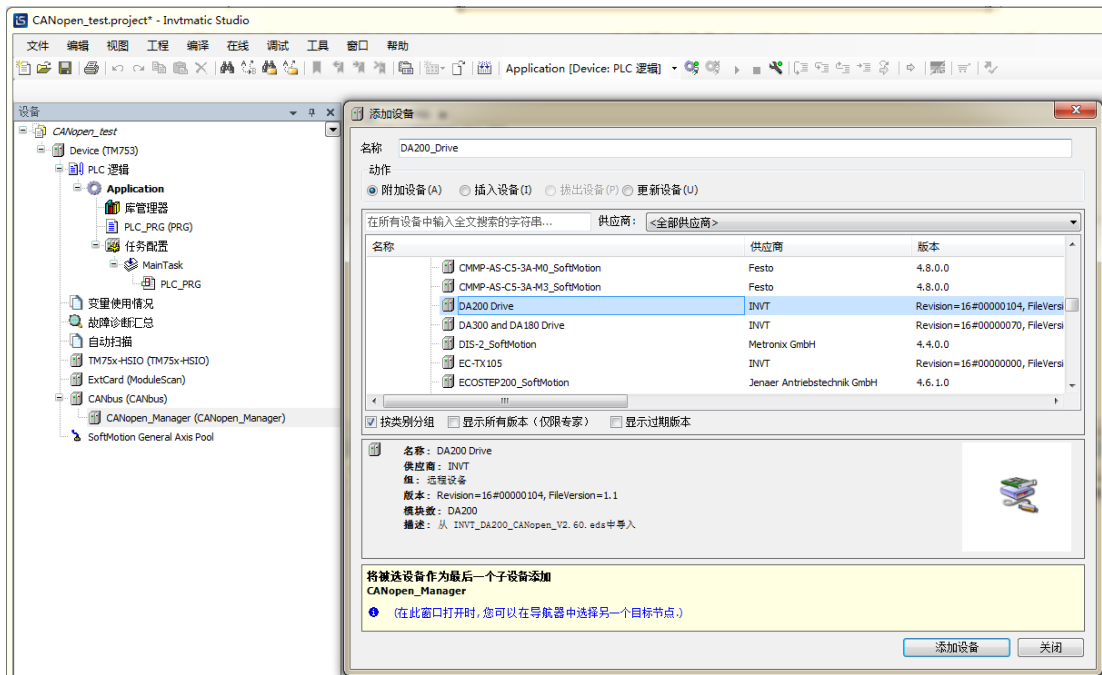


步骤3 在设备树中双击 **ExtCard**，进入设置界面，默认是无设备选项，选择 **CANopen**，左侧设备树会自动添加 **CANbus 总线** 和 **CANopen\_Manager**。



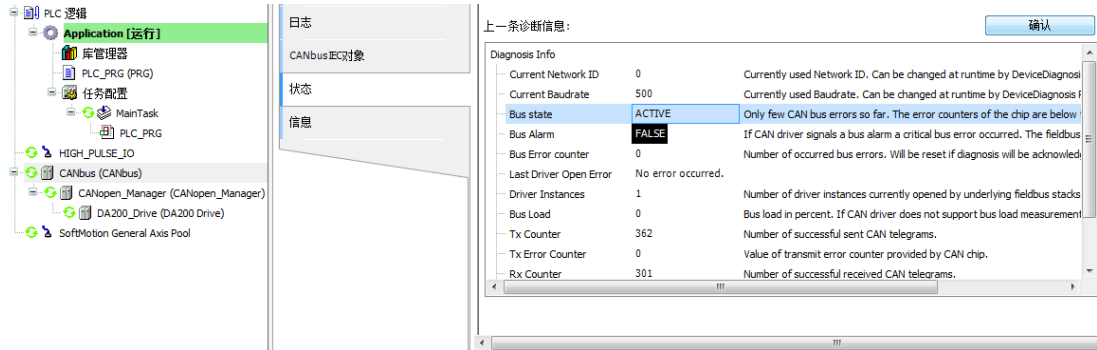


步骤4 在设备树中右键点击 **CANopen\_Manager**，选择**添加设备**后，再选择 CANopen 远程设备，选择 **DA200 Drive**，点击右下角**添加设备**，添加 DA200 CANopen 驱动器成功。



步骤5 在“CANbus”概述界面，波特率配置需要和 DA200 CANopen 伺服一致 (DA200 P4.02)；在“DA200\_Drive”概述界面节点 ID 配置需要和 DA200 CANopen 伺服一致 (DA200 P4.05)。

步骤6 完成设备物理连接后，下载程序，登录设备，可以查看到 CANopen 连接 DA200 通讯成功。



**注意：**

- 若对数据的实时性要求较高，为避免总线竞争导致数据收发的少量延时，CAN 总线负载应低于 30%。
- 有同步需求的 CAN 总线，总线的同步报文中的窗口长度设置值应略小于循环周期值。

☒ 启动同步生成

COB-ID(Hex) 16# 80

循环周期 (μs) 60000

窗口长度 (μs) 58000

☐ 启动同步消耗

- CANopen 所在任务的任务周期时间应略大于任务实际执行时间。



- 确保主站能正常监控从站，应勾选从站中的使能心跳产生选项。

#### 心跳

☐ 使能节点保护

保护时间(ms)

0

生命周期因子

0

☒ 使能心跳产生

发送端时间(ms)

200

✓ 心跳消费(1/1有效)



## 附录B SMC\_ERROR 说明

错误序号	模块	枚举变量	说明
0	All function blocks	SMC_NO_ERROR	没有错误
1	DriveInterface	SMC_DI_GENERAL_COMMUNICATION_ERROR	通讯错误（例如 Sercos 环断裂）
2	DriveInterface	SMC_DI_AXIS_ERROR	轴错误
10	DriveInterface	SMC_DI_SWLIMITS_EXCEEDED	允许范围内的位置输出(SWLimit)
11	DriveInterface	SMC_DI_HWLIMITS_EXCEEDED	硬件限位开关被激活
13	DriveInterface	SMC_DI_HALT_OR_QUICKSTOP_NOT_SUPPORTED	驱动器状态停止或者不支持快速停止
14	DriveInterface	SMC_DI_VOLTAGE_DISABLED	驱动器没有使能
15	DriveInterface	SMC_DI_IRREGULAR_ACTPOSITION	驱动器当前给予的位置格式不正确，检查通讯
16	DriveInterface	SMC_DI_POSITIONLAGERROR	位置滞后错误，在设置和当前位置超过限制值
20	All motion generating function blocks	SMC_REGULATOR_OR_START_NOT_SET	控制器没有使能或抱闸没有打开
21	Axis in wrong controller mode	SMC_WRONG_CONTROLLER_MODE	轴处于错误的控制模式
30	DriveInterface	SMC_FB_WASNT_CALLED_DURING_MOTION	运动控制创建的模块在运动结束之前没有被调用
31	All function blocks	SMC_AXIS_IS_NO_AXIS_REF	给出的 AXIS_REF 变量不是 AXIS_REF 类型
32	Axis in wrong controller mode	SMC_AXIS_REF_CHANGED_DURING_OPERATION	AXIS_REF 变量的返回值在模块激活前被处理
33	DriveInterface	SMC_FB_ACTIVE_AXIS_DISABLED	轴在移动时没有被激活 (MC_Power.bRegulatorOn)
34	All motion generating function blocks	SMC_AXIS_NOT_READY_FOR_MOTION	在当前状态下轴不能处理当前命令
40	VirtualDrive	SMC_VD_MAX_VELOCITY_EXCEEDED	达到最大速度(fMaxVelocity)
41	VirtualDrive	SMC_VD_MAX_ACCELERATION_EXCEEDED	达到最大加速度(fMaxAcceleration)
42	VirtualDrive	SMC_VD_MAX_DECELERATION_EXCEEDED	达到最大减速度 (fMaxDeceleration)
50	SMC_Homing	SMC_3SH_INVALID_VELACC_VALUES	无效的速度或加速度值
51	SMC_Homing	SMC_3SH_MODE_NEEDS_HWLIMIT	模块需要使用结束限位开关（安全用途）
70	SMC_SetControllerMode	SMC_SCM_NOT_SUPPORTED	模式不支持
71	SMC_SetControllerMode	SMC_SCM_AXIS_IN_WRONG_STATE	在当前模式下使用的控制模式不支持
75	SMC_SetTorque	SMC_ST_WRONG_CONTROLLER	轴不是一个正确的控制模式



错误序号	模块	枚举变量	说明
		MODE	
80	SMC_ResetAxisGroup	SMC_RAG_ERROR_DURING_STARTUP	在轴组启动时发生错误
90	SMC_ChangeGearin gRatio	SMC_CGR_ZERO_VALUES	不正确的变量
91	SMC_ChangeGearin gRatio	SMC_CGR_DRIVE_POWERED	驱动器控制模式下不能更改传动比
92	SMC_ChangeGearin gRatio	SMC_CGR_INVALID_POSPERIOD	不合适的位置周期( $\leq 0$ )
110	MC_Power	SMC_P_FTASKCYCLE_EMPTY	轴在扫描周期内不包含任何信息 (fTaskCycle=0)
120	MC_Reset	SMC_R_NO_ERROR_TO_RESET	轴没有错误复位
121	MC_Reset	SMC_R_DRIVE_DOESNT_ANSWER	轴没有执行错误复位
122	MC_Reset	SMC_R_ERROR_NOT_RESETTABLE	错误不能被复位
123	MC_Reset	SMC_R_DRIVE_DOESNT_ANSWER _IN_TIME	与轴之间的通讯没有回应
130	MC_ReadParameter, MC_ReadBoolParameter	SMC_RP_PARAM_UNKNOWN	参数序号未知
131	MC_ReadParameter, MC_ReadBoolParameter	SMC_RP_REQUESTING_ERROR	在将参数传送到驱动器过程中发生 错误, 参阅编程手册 ReadDriveParameter 的错误 (SM_DriveBasic.lib)
140	MC_WriteParameter, MC_WriteBoolParameter	SMC_WP_PARAM_INVALID	参数序号未知或不允许进行写操作
141	MC_WriteParameter, MC_WriteBoolParameter	SMC_WP_SENDING_ERROR	参阅编程手册 WriteDriveParameter 的错误 (Drive_Basic.lib)
170	MC_Home	SMC_H_AXIS_WASNT_STANDSTILL	轴不是标准状态
171	MC_Home	SMC_H_AXIS_DIDNT_START_HOMING	在执行回零时发生错误
172	MC_Home	SMC_H_AXIS_DIDNT_ANSWER	通讯错误
173	MC_Home	SMC_H_ERROR_WHEN_STOPPING	执行回零错误停止, 查阅是否设置 减速度
180	MC_Stop	SMC_MS_UNKNOWN_STOPPING_ERROR	停止时发生未知错误
181	MC_Stop	SMC_MS_INVALID_ACCDEC_VALUES	不合适的速度或加速度值
182	MC_Stop	SMC_MS_DIRECTION_NOT_APPLICABLE	Direction=shortest 不可用
183	MC_Stop	SMC_MS_AXIS_IN_ERRORSTOP	轴位于错误停止状态, 停止不能被 处理
184	MC_Stop	SMC_BLOCKING_MC_STOP_WASNT_CALLED	一个 MC_Stop 的实例, 存在多次调用
201	MC_MoveAbsolute	SMC_MA_INVALID_VELACC_VALUES	不合适的速度或加速度值



错误序号	模块	枚举变量	说明
		ES	
202	MC_MoveAbsolute	SMC_MA_INVALID_DIRECTION	方向错误
226	MC_MoveRelative	SMC_MR_INVALID_VELACC_VALUES	不合适的速度或加速度值
227	MC_MoveRelative	SMC_MR_INVALID_DIRECTION	方向错误
251	MC_MoveAdditive	SMC_MAD_INVALID_VELACC_VALUES	不合适的速度或加速度值
252	MC_MoveAdditive	SMC_MAD_INVALID_DIRECTION	方向错误
276	MC_MoveSuperImposed	SMC_MSI_INVALID_VELACC_VALUES	不合适的速度或加速度值
277	MC_MoveSuperImposed	SMC_MSI_INVALID_DIRECTION	方向错误
301	MC_MoveVelocity	SMC_MV_INVALID_ACCDEC_VALUES	不合适的速度或加速度值
302	MC_MoveVelocity	SMC_MV_DIRECTION_NOT_APPLICABLE	Direction=shortest/fastest 不支持
325	MC_PositionProfile	SMC_PP_ARRAYSIZE	错误排列尺寸
326	MC_PositionProfile	SMC_PP_STEP0MS	步时间=t#0ms
350	MC_VelocityProfile	SMC_VP_ARRAYSIZE	错误排列尺寸
351	MC_VelocityProfile	SMC_VP_STEP0MS	步时间=t#0ms
375	MC_AccelerationProfile	SMC_AP_ARRAYSIZE	错误排列尺寸
376	MC_AccelerationProfile	SMC_AP_STEP0MS	步时间=t#0ms
400	MC_TouchProbe	SMC_TP_TRIGGEROCCUPIED	触发条件已经被激活
401	MC_TouchProbe	SMC_TP_COULDNT_SET_WINDOW	驱动器接口不支持窗口功能
402	MC_TouchProbe	SMC_TP_COMM_ERROR	通讯错误
410	MC_AbortTrigger	SMC_AT_TRIGGERNOTOCCUPIED	触发条件已经被终止
426	SMC_MoveContinuousRelative	SMC_MCR_INVALID_VELACC_VALUES	不合适的速度或加速度值
427	SMC_MoveContinuousRelative	SMC_MCR_INVALID_DIRECTION	方向错误
451	SMC_MoveContinuousAbsolute	SMC_MCA_INVALID_VELACC_VALUES	不合适的速度或加速度值
452	SMC_MoveContinuousAbsolute	SMC_MCA_INVALID_DIRECTION	方向错误
453	SMC_MoveContinuousAbsolute	SMC_MCA_DIRECTION_NOT_APPLICABLE	Direction= fastest 不可用
600	SMC_CamRegister	SMC_CR_NO_TAPPETS_IN_CAM	CAM 中不包含任何挺杆
601	SMC_CamRegister	SMC_CR_TOO_MANY_TAPPETS	挺杆组 ID 达到 MAX_NUM_TAPPETS
602	SMC_CamRegister	SMC_CR_MORE_THAN_32_ACCESSES	在一个 CAM_REF 中超过 32 个接口
625	MC_CamIN	SMC_CI_NO_CAM_SELECTED	没有 CAM 被选中
626	MC_CamIN	SMC_CI_MASTER_OUT_OF_SCALE	主轴超出范围
627	MC_CamIN	SMC_CI_RAMPIN_NEEDS_VELACC	针对 ramp_in 功能块速度和加速度



错误序号	模块	枚举变量	说明
		_VALUES	必须被精确指定
628	MC_CamIN	SMC_CI_SCALING_INCORRECT	比例变量 fEditor/TableMasterMin/Max 不正确
640	SMC_CAMBounds, SMC_CamBounds_Pos	SMC_CB_NOT_IMPLEMENTED	给予的 CAM 格式的功能块不支持
675	MC_GearIn	SMC_GI_RATIO_DENOM	RatioDenominator=0
676	MC_GearIn	SMC_GI_INVALID_ACC	加速度不合适
677	MC_GearIn	SMC_GI_INVALID_DEC	减速度不合适
725	MC_Phase	SMC_PH_INVALID_VELACCDEC	速度、加速度、减速度不合适
726	MC_Phase	SMC_PH_ROTARYAXIS_PERIOD0	旋转轴 fPositionPeriod=0
750	All modules using MC_CAM_REF as input	SMC_NO_CAM_REF_TYPE	给定的 CAM 不是 MC_CAM_REF 类型
751	MC_CamTableSelect	SMC_CAM_TABLE_DOES_NOT_COVER_MASTER_SCALE	从 CamTable 中获取的数据不是通过数据转化得到的主轴范围(xStart and xEnd)
775	MC_GearInPos	SMC_GIP_MASTER_DIRECTION_CHANGE	在从轴耦合过程中主轴改变旋转方向
800	SMC_BacklashCompensation	SMC_BC_BL_TOO_BIG	齿轮返回比(fBacklash)太大 (>position periode/2)
1000	CNC function blocks which are supervising the licensing	SMC_NO_LICENSE	目标没有进行 CNC 的授权
1001	SMC_Interpolator	SMC_INT_VEL_ZERO	路径不能被处理因为速度=0
1002	SMC_Interpolator	SMC_INT_NO_STOP_AT_END	上一个路径对象 Vel_End>0
1003	SMC_Interpolator	SMC_INT_DATA_UNDERRUN	警告: GEOINFO 列表在 DataIn 进行处理, 但是列表最后没有被设置 原因: 忘记在 DataIn 中设置 EndOfList 或者 SMC_Interpolator 比路径编译模块处理速度快
1004	SMC_Interpolator	SMC_INT_VEL_NONZERO_AT_STOP	停止速度>0
1005	SMC_Interpolator	SMC_INT_TOO_MANY_RECURSIONS	使用太多 SMC_Interpolator 调用 SoftMotion 错误
1006	SMC_Interpolator	SMC_INT_NO_CHECKVELOCITIES	Input-OutQueue DataIn 没有作为 SMC_CheckVelocities 的最后处理模块
1007	SMC_Interpolator	SMC_INT_PATH_EXCEEDED	内部/数值错误
1008	SMC_Interpolator	SMC_INT_VEL_ACC_DEC_ZERO	速度、加速度或减速度为空或者太低
1009	SMC_Interpolator	SMC_INT_DWIPOTIME_ZERO	FB 调用 dwIpoTime=0
1050	SMC_Interpolator2Dir	SMC_INT2DIR_BUFFER_TOO_SMALL	数据缓冲区太小
1051	SMC_Interpolator2Dir	SMC_INT2DIR_PATH_FITS_NOT_IN_QUEUE	路径没有完全包含在队列中



错误序号	模块	枚举变量	说明
1100	SMC_CheckVelocities	SMC_CV_ACC_DEC_VEL_NONPOSITIVE	速度、减速度或加速度值不为正向
1120	SMC_Controlaxisbypass	SMC_CA_INVALID_ACCDEC_VALUES	变量 fGapVelocity/fGapAcceleration/fGapDeceleration 不是正值
1200	SMC_NCDecoder	SMC_DEC_ACC_TOO_LITTLE	加速度值不允许
1201	SMC_NCDecoder	SMC_DEC_RET_TOO_LITTLE	减速度值不允许
1202	SMC_NCDecoder	SMC_DEC_OUTQUEUE_RAN_EMPTY	低于 Queue 的数据被读取，且为空
1203	SMC_NCDecoder	SMC_DEC_JUMP_TO_UNKNOWN_LINE	因为行号未知所以跳转的行号不能执行
1204	SMC_NCDecoder	SMC_DEC_INVALID_SYNTAX	语法错误
1205	SMC_NCDecoder	SMC_DEC_3DMODE_OBJECT_NOT_SUPPORTED	对象不支持 3D 模式
1300	SMC_GCodeViewer	SMC_GCV_BUFFER_TOO_SMALL	缓冲区太小
1301	SMC_GCodeViewer	SMC_GCV_BUFFER_WRONG_TYPE	缓冲区元素类型错误
1302	SMC_GCodeViewer	SMC_GCV_UNKNOWN_IPO_LINE	当前插补行不能被找到
1500	All function blocks using SMC_CNC_REF	SMC_NO_CNC_REF_TYPE	给定的 CNC 程序不是 SMC_CNC_REF 类型
1501	All function blocks using SMC_OUTQUEUE	SMC_NO_OUTQUEUE_TYPE	给定的 OutQueue 不是 SMC_OUTQUEUE 类型
1600	CNC function blocks	SMC_3D_MODE_NOT_SUPPORTED	这个功能块只在 2D 路径中可用
2000	SMC_ReadNCFile	SMC_RNCF_FILE_DOESNT_EXIST	文件不存在
2001	SMC_ReadNCFile	SMC_RNCF_NO_BUFFER	没有缓冲分配
2002	SMC_ReadNCFile	SMC_RNCF_BUFFER_TOO_SMALL	缓冲区太小
2003	SMC_ReadNCFile	SMC_RNCF_DATA_UNDERRUN	缓冲区中低缓冲数据被读取，且为空
2004	SMC_ReadNCFile	SMC_RNCF_VAR_COULDNT_BE REPLACED	占位符变量不能被替换
2005	SMC_ReadNCFile	SMC_RNCF_NOT_VARLIST	输入的 pvl 不能指向 SMC_VARLIST 对象
2050	SMC_ReadNCQueue	SMC_RNCQ_FILE_DOESNT_EXIST	文件不能打开
2051	SMC_ReadNCQueue	SMC_RNCQ_NO_BUFFER	没有缓冲区定义
2052	SMC_ReadNCQueue	SMC_RNCQ_BUFFER_TOO_SMALL	缓冲区太小
2053	SMC_ReadNCQueue	SMC_RNCQ_UNEXPECTED_EOF	未知文件结尾
2100	SMC_AxisDiagnostic Log	SMC_ADL_FILE_CANNOT_BE OPENED	文件不能被打开
2101	SMC_AxisDiagnostic Log	SMC_ADL_BUFFER_OVERRUN	超过范围的缓冲，WriteToFile 必须调用更频繁
2200	SMC_ReadCAM	SMC_RCAM_FILE_DOESNT_EXIST	文件不能打开
2201	SMC_ReadCAM	SMC_RCAM_TOO_MUCH_DATA	保存到 CAM 数据太多
2202	SMC_ReadCAM	SMC_RCAM_WRONG_COMPILE_TYPE	错误编译模式
2203	SMC_ReadCAM	SMC_RCAM_WRONG_VERSION	文件版本错误



错误序号	模块	枚举变量	说明
2204	SMC_ReadCAM	SMC_RCAM_UNEXPECTED_EOF	未知的文件结尾
3001	SMC_WriteDriveParamsToFile	SMC_WDPF_CHANNEL_OCCUPIED	SMC_WDPF_TIMEOUT_PREPARING_LIST
3002	SMC_WriteDriveParamsToFile	SMC_WDPF_CANNOT_CREATE_FILE	文件不能被创建
3003	SMC_WriteDriveParamsToFile	SMC_WDPF_ERROR_WHEN_READING_PARAMS	读取文件参数错误
3004	SMC_WriteDriveParamsToFile	SMC_WDPF_TIMEOUT_PREPARING_LIST	准备参数列表时时间错误
5000	SMC_Encoder	SMC_ENC_DENOM_ZERO	译码器参考的转换因子(dwRatioTechUnitsDenom)为 0
5001	SMC_Encoder	SMC_ENC_AXISUSEDBYOTHERFB	其他模块正在处理译码轴
5002	DriveInterface	SMC_ENC_FILTER_DEPTH_INVALID	过滤器选择不合适



值得信赖的工控与能效解决方案提供者



深圳市英威腾电气股份有限公司

地址：深圳市光明区马田街道松白路英威腾光明科技大厦

苏州英威腾电力电子有限公司

地址：苏州高新区科技城昆仑山路1号

服务热线：400-700-9997 网址：[www.invt.com.cn](http://www.invt.com.cn)



英威腾微信公众号



英威腾电子手册



66001-01363

产品资料可能有所改动，恕不另行通知。版权所有，仿冒必究。

202408 (V1.0)