



XS 系列 PLCopen 标准控制器

用户手册【软件篇】(XS Studio)

无锡信捷电气股份有限公司

资料编号：PS06 20260819 1.7

XS 系列 PLCopen 标准控制器
用户手册[软件篇] (XS Studio)

目录

产品简介	1
快速入门	2
网络配置	3
EtherNet/IP 通讯	4
EtherCAT 配置	5
编程基础	6
编程语言	7
特殊功能	8
资源使用表	9
XS Studio 升级工具	10
设备诊断	11
可视化	12
任务分核	13
Codesys 加密	14
掉电保持	15
工程比较	16
常见问题及解决方法	17
手册更新日志	

基本说明

- ◆ 感谢您购买了信捷 XS 系列可编程控制器。
- ◆ 本手册主要介绍 XS 系列可编程控制器的软件的相关使用内容。
- ◆ 在使用产品之前，请仔细阅读本手册，并在充分理解手册内容的前提下，进行编程。
- ◆ 请将本手册交付给最终用户。

用户须知

- ◆ 只有具备一定的电气知识的操作人员才可以对产品进行接线等其他操作，如有使用不明的地方，请咨询本公司的技术部门。
- ◆ 手册等其他技术资料中所列举的示例仅供用户理解、参考用，不保证一定动作。
- ◆ 将该产品与其他产品组合使用的时候，请确认是否符合有关规格、原则等。
- ◆ 使用该产品时，请自行确认是否符合要求以及安全。
- ◆ 请自行设置后备及安全功能，以避免因本产品故障而可能引发的机器故障或损失。

责任申明

- ◆ 手册中的内容虽然已经过仔细的核对，但差错难免，我们不能保证完全一致。
- ◆ 我们会经常检查手册中的内容，并在后续版本中进行更正，欢迎提出宝贵意见。
- ◆ 手册中所叙述的内容如有变动，恕不另行通知。

关联手册

关于 XS 系列 PLC 的硬件相关、高级运动控制指令应用等内容，请查询以下手册。

手册下载网址：www.xinje.com。

- ◆ 《XS 系列可编程控制器用户手册【硬件篇】》
- ◆ 《XS 系列可编程控制器用户手册【基本指令篇】》
- ◆ 《XS 系列可编程控制器用户手册【运动控制篇】》

WUXI XINJE ELECTRIC CO., LTD. 版权所有

未经明确的书面许可，不得复制、传翻或使用本资料及其中的内容，违者要对造成的损失承担责任。保留包括实用模块或设计的专利许可及注册中提供的所有权力。

二〇二三年 一月

目录

1. 产品简介	1
1-1. 概述	1
1-1-1. 产品简介	1
1-1-2. 系统构成	4
1-2. XS STUDIO 概述	5
1-2-1. XS Studio 简介	5
1-2-2. XS Studio 与硬件的连接	5
1-2-3. 软件安装&卸载	6
2. 快速入门	14
2-1. 界面导航	14
2-2. XS STUDIO 编程举例	14
2-2-1. 基本编程操作	16
2-2-2. 任务配置	21
2-2-3. 设备联机	25
2-2-4. 程序下载/读取	33
2-2-5. 程序调试	38
2-2-6. 仿真	40
2-2-7. PLC 脚本功能	40
2-3. XS STUDIO 编写流水灯程序样例	41
2-4. 如何登录设备	44
2-4-1. 登录设备的必备条件和操作简介	44
2-4-2. 扫描不到设备或者连接不上设备的解决方法	44
3. 网络配置	47
3-1. 设备组态	47
3-1-1. 网络组态	47
3-1-2. 硬件组态	53
3-1-3. 设备树操作	56
3-1-4. 组态编辑错误定位	56
3-2. MODBUS 通讯	57
3-2-1. MODBUS 主站配置	57
3-2-2. MODBUS 从站 (Other) 配置	62
3-2-3. MODBUS 从站 (XINJE) 配置	63
3-2-4. MODBUS 通讯帧格式说明	66
3-3. 串口自由协议通讯	70
3-3-1. 概要	70
3-3-2. 串口配置	70
3-3-3. 通讯配置	70
3-3-4. 应用举例	71
3-4. MODBUS TCP 通讯	77
3-4-1. MODBUS TCP 主站配置	77

3-4-2.	MODBUS TCP 从站 (Other) 配置.....	79
3-4-3.	MODBUS TCP 从站 (XINJE) 配置.....	82
3-4-4.	MODBUS TCP 常见故障.....	89
3-4-5.	MODBUS TCP 通讯帧格式说明.....	89
3-5.	CAN 总线.....	91
3-5-1.	参数配置.....	91
3-5-2.	CANopen 网络.....	93
3-5-3.	CANopen 主站配置.....	95
3-5-4.	应用举例.....	104
3-6.	OPC UA 通讯.....	108
3-6-1.	通讯概述.....	108
3-6-2.	参数配置.....	108
3-6-3.	OPC UA 样例.....	111
3-7.	PROFINET 通信.....	119
3-7-1.	通讯概述.....	119
3-7-2.	通讯规格.....	119
3-7-3.	PROFINET 通讯说明.....	120
3-7-4.	PROFINET 从站参数配置.....	121
3-7-5.	PROFINET 样例.....	124
4.	ETHERNET/IP 通讯.....	133
4-1.	ETHERNET/IP 概要.....	133
4-2.	ETHERNET/IP 隐式通讯.....	134
4-2-1.	EtherNet/IP 作为从站的样例.....	134
4-2-2.	EtherNet/IP 作为主站的样例.....	139
4-3.	ETHERNET/IP 显式通讯.....	143
4-3-1.	功能介绍.....	143
4-3-2.	EIP 标签全局变量表定义.....	143
4-3-3.	EtherNet/IP 显式通讯样例.....	146
5.	ETHERCAT 配置.....	150
5-1.	ETHERCAT 概要.....	150
5-1-1.	概述.....	150
5-1-2.	系统构成.....	150
5-1-3.	通讯规格.....	150
5-1-4.	EtherCAT 通讯连接说明.....	151
5-2.	ETHERCAT 通讯规格.....	152
5-2-1.	EtherCAT 帧结构.....	152
5-2-2.	状态机 ESM.....	152
5-2-3.	从站控制器 ESC.....	153
5-2-4.	SII 区域.....	156
5-2-5.	SD0.....	156
5-2-6.	PD0.....	158
5-2-7.	通信同步模式.....	160
5-3.	ETHERCAT 参数配置.....	162
5-3-1.	EtherCAT 主站.....	162
5-3-2.	EtherCAT 从站.....	165
5-3-3.	轴配置.....	169
5-3-4.	EtherCAT 诊断.....	175
5-3-5.	EtherCAT 控制工程.....	185
5-4.	伺服参数调试.....	187

5-4-1. 伺服功能码	187
5-4-2. ESC 寄存器	189
6. 编程基础	191
6-1. 直接地址	191
6-1-1. 定义语法	191
6-1-2. PLC 直接地址存储区域	192
6-2. 变量	193
6-2-1. 概述	193
6-2-2. 变量定义	193
6-2-3. 变量类型	198
6-2-4. 变量导入与导出	200
6-3. 配方操作	202
6-3-1. 应用举例	202
7. 编程语言	207
7-1. XS STUDIO 支持的编程语言	207
7-2. 结构化文本语言 (ST)	207
7-2-1. 简介	207
7-2-2. ST 程序执行顺序	208
7-2-3. 语句	209
7-2-4. ST 编辑	217
7-3. 梯形图	220
7-3-1. 简介	220
7-3-2. LD 术语解释	220
7-3-3. 编辑器显示	221
7-3-4. 梯形图元素	225
7-3-5. 梯形图操作	229
7-3-6. 网络/程序段功能	233
7-3-7. 分支功能	240
7-3-8. 快捷键提示栏	245
7-3-9. 编辑器字体和颜色设置	245
7-3-10. 单键操作	248
7-3-11. 变量注释	249
8. 特殊功能	251
8-1. 外部中断	251
8-2. 高速计数	253
8-3. 高速 I/O 配置	255
8-4. 系统设置	262
8-4-1. 固件 1.1.0 及以上版本系统设置应用举例	262
8-4-2. NTP 同步时间	263
8-5. PLC 指令	269
8-5-1. 应用举例	269
8-6. 固定地址分配	275
8-6-1. 设置地址分配方式	275
8-6-2. 连续 I/O 地址	276
8-7. 时钟	278
8-7-1. 功能概述	278
8-7-2. 应用举例	278
8-8. 微分监视器	280

8-8-1. 功能概述	280
8-8-2. 参数说明	280
8-8-3. 启动/停止功能说明	281
8-9. 隐式 POU 检查	285
8-9-1. 适用设备	285
8-9-2. 功能概述	285
8-9-3. 故障诊断	286
9. 资源使用表	289
9-1. 资源使用表	289
9-2. 功能介绍	289
9-2-1. 更新资源使用表	289
9-2-2. 界面功能介绍	290
10. XS STUDIO 升级工具	293
10-1. 通信配置	294
10-1-1. 联机	294
10-1-2. 网络设置	297
10-2. 升级	299
10-2-1. PLC 固件升级	299
10-2-2. 模块固件升级	301
10-2-3. 用户程序管理	301
10-2-4. 备份/还原	304
10-2-5. 恢复出厂设置	308
10-3. 文件	309
10-3-1. 文件传输	309
10-4. 选项	312
10-5. 帮助	312
11. 设备诊断	313
11-1. 设备诊断	313
11-2. 具体功能说明	313
12. 可视化	318
12-1. 视图	318
12-1-1. 视图管理器	318
12-1-2. 可视化工具箱	319
12-1-3. 元素属性	320
12-2. 本地可视化	321
12-2-1. 配置步骤	321
12-2-2. 远程进入桌面	322
12-3. 网络可视化	325
13. 任务分核	327
13-1. 适用设备	327
13-2. 自动分核	328
13-2-1. 上位机分核策略	328
13-2-2. 下位机分核策略	329
13-2-3. 配置步骤	330
13-3. 手动分核	331
13-3-1. 参数说明	331
13-3-2. 配置步骤	331

14. CODESYS 加密.....	333
14-1. 工程加密.....	333
14-1-1. 功能说明	333
14-1-2. 配置步骤	333
14-2. POU 加密.....	334
14-2-1. 功能说明	334
14-2-2. 配置步骤	334
14-2-3. 注意点	338
14-3. PLC 加密.....	340
14-3-1. 功能说明	340
14-3-2. 配置方法一	340
14-3-3. 配置方法二	342
14-3-4. 取消设备加密	345
14-3-5. 注意点	347
15. 掉电保持	348
15-1. 传统保持.....	348
15-1-1. 配置步骤	348
15-1-2. 保持范围	349
15-2. 标准保持.....	350
15-2-1. 变量定义	350
15-2-3. 掉电保持模式	354
15-2-4. 掉电保持规则	354
15-2-5. 地址分配	354
15-2-6. 配方操作	355
15-2-7. 掉电保持说明	357
15-2-8. 变量支持情况说明	358
15-2-9. M 区范围说明.....	359
16. 工程比较	360
16-1. 离线工程比较.....	360
16-1-1. 工程比较	360
16-1-2. 对象比较	364
16-2. 在线工程比较.....	366
16-2-1. 适用设备	366
16-2-2. 操作说明	366
17. 常见问题及解决方法	369
17-1. PACKAGE	369
17-1-1. Package 命名规则.....	369
17-1-2. Package 的获取.....	369
17-1-3. Package 的安装.....	369
17-2. XS 系列 PLC 固件升级.....	370
17-2-1. 固件命名规则	370
17-2-2. 固件获取	370
17-2-3. 固件安装及其注意事项	370
17-3. XS 系列的本地扩展模块.....	373
17-4. XS 系列的远程扩展模块.....	375
17-5. 拨码	377
17-6. 用户新安装 XS STUDIO 软件, 打开后进行编译, 出现了很多报错.....	377
17-7. 网关显示红点.....	377

17-8. 添加多个 ETHERCAT 从站后，有 WARNING 提示 378

17-9. ETHERCAT 轴一运行通讯就断了 378

17-10. 用户怎么取消密码登录 378

17-11. 为什么无法连接到 PLC 设备 378

17-12. IP 地址修改不成功 379

17-13. 提示“没有可用于此对象的源码，是否浏览原始库以显示源代码” 379

17-14. SETPOSITION 清完位置之后断电上电绝对值伺服位置发生变化 380

17-15. PLC 死机 380

17-16. 在线下载程序丢失 380

17-17. 不同电脑在同一局域网有时会连接到其他设备 380

17-18. 添加隐含检查功能 380

17-19. 实现掉电保持的注意点 383

17-20. 打开工程报错，工程保存必须以存档的方式保存 384

17-21. 设置可以添加行注释和节点注释 384

手册更新日志 385

1. 产品简介

1-1. 概述

1-1-1. 产品简介

XS Studio 涵盖 XSDH、XSLH、XSA、XSF 等系列，为用户提供智能自动化解决方案。采用国际标准 IEC61131-3 架构，支持梯形图 LD、结构化文本 ST、功能块图 FBD、顺序功能流程图 SFC、控制流程框图 CFC 等多种编程语言。支持的总线包括：EtherCAT、Modbus/ModbusTCP、EtherNet/IP、OPC UA(Server)、CAN。

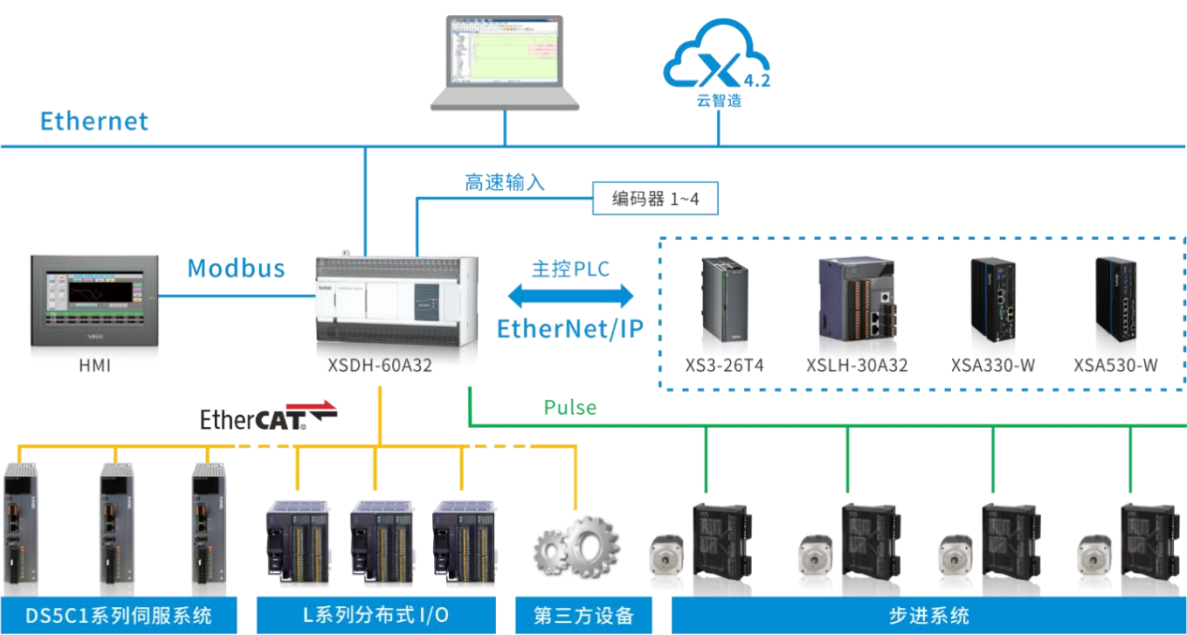
支持的扩展模块型号如下：

型号	功能
XSDH 系列	
XD-EnXmY	n 点输入，m 点输出，PNP/NPN 型输入，继电器/晶体管输出
XD-E4AD	14Bit，4 通道模拟量输入（电流电压可选），一阶系数可调，每个通道可单独使能；
XD-E2DA	12Bit，2 通道模拟量输出模块（电流电压可选）；
XD-E4DA	12Bit，4 通道模拟量输出模块（电流电压可选）；
XD-E4DA-H	12Bit，4 通道模拟量输出模块（电流电压可选）；通道间隔离处理，抗干扰性能更优；
XD-E8AD	14Bit，8 通道模拟量输入模块；前 4 路为电压（0~5V、0~10V、-5~5V、-10~10V）输入，后 4 路为电流（0~20mA、4~20mA、-20~20mA）输入，一阶系数可调，每个通道可单独使能；（注：硬件版本需 H2.2 及以上支持双极性）
XD-E8AD-A	14Bit，8 通道模拟量电流（0~20mA、4~20mA、-20~20mA）输入，一阶系数可调，每个通道可单独使能；（注：硬件版本需 H2.2 及以上支持双极性）
XD-E8AD-V	14Bit，8 通道模拟量电压（0~5V、0~10V、-5~5V、-10~10V）输入，一阶系数可调，每个通道可单独使能；（注：硬件版本需 H2.2 及以上支持双极性）
XD-E12AD-V	14Bit，12 通道模拟量电压（0~5V、0~10V、-5~5V、-10~10V）输入，一阶系数可调，每个通道可单独使能；
XD-E4AD2DA	14Bit，4 通道模拟量输入（电流电压可选），电流 0~20mA、4~20mA、-20~20mA 可选，电压 0~5V、0~10V、-5~5V、-10~10V 可选；12Bit 2 通道模拟量输出模块（电流电压可选），电压 0~5V、0~10V、-5~5V、-10~10V 可选，电流 0~20mA、4~20mA 可选，电流一阶系数可调，每个通道可单独使能；（注：V6 及以下版本的 XD-E4AD2DA 模块不支持-5~5V、-10~10V、-20~20mA 范围）
XD-E2AD2PT2DA	2 通道 PT100 温度采集（分辨率为 0.1℃）；16Bit，2 通道模拟量输入（电流、电压可选）；10Bit，2 通道模拟量输出（电压电流可选）；每个通道可单独

型号	功能
	使能；
XD-E3AD4PT2DA	4 通道 PT100 温度采集(分辨率为 0.1℃)；14Bit, 3 通道模拟量输入 (0~20mA, 4~20mA 可选)；10Bit, 2 通道模拟量输出 (0~5V, 0~10V 可选)；每个通道可单独使能；
XD-E2TC-P	2 通道热电偶, 支持多种类型热电偶温度传感器用模拟输入, 分辨率 0.1℃, 2 通道独立输出 PID 参数；
XD-E6TC-P	6 通道热电偶, 支持多种类型热电偶温度传感器用模拟输入, 分辨率 0.1℃, 6 通道独立输出 PID 参数；
XD-E6TC-P-H	6 通道热电偶, 支持多种类型热电偶温度传感器用模拟输入, 通道间隔离, 分辨率 0.1℃, 6 通道晶体管输出, 6 组独立的 PID 参数, 支持自整定功能, 内置冷端补偿；
XD-E6PT-P	-100~500℃, 6 通道 PT100 温度采集模块, 分辨率 0.1℃, PID 输出；
XD-E4PT3-P	-100~500℃, 4 通道 PT100 (三线制) 温度采集模块, 分辨率 0.1℃, 4 通道独立 PID 输出；
XD-E1WT-D	可采集一路压力传感器的模拟量电压信号 (-20~20mV), 22 位的高精度 A/D 转换, 采用 Δ - Σ ADC 的 A/D 转换方式, 更高更快的 CPU 处理速度, 更加优化的算法, 抗共振性能更好, 供电电源 DC24V；
XD-E2WT-D	可采集二路压力传感器的模拟量电压信号 (-20~20mV), 22 位的高精度 A/D 转换, 采用 Δ - Σ ADC 的 A/D 转换方式, 更高更快的 CPU 处理速度, 更加优化的算法, 抗共振性能更好, 供电电源 DC24V；
XD-E4WT-D	可采集四路传感器模拟电压信号 (-20~20mV), 22 位高精度 AD 转换, 采用 Δ - Σ ADC 的 A/D 转换方式, 更高更快的 CPU 处理速度, 更加优化的算法, 抗共振性能好, 供电电源 DC24V；
XD-E4SSI	XD 系列连接 SSI 信号编码器专用扩展模块, 一个模块同时最多可以连接 4 路, 通讯速度可达 400us/通道；
XD-NES-ED	XD 系列 PLC 扩展 ED 模块, 可以扩展 1 个 RS232 或者 RS485 通讯口； (注: 只能使用其一)
XD-NS-BD	XD 系列 PLC 扩展 BD, RS-232 通讯功能；
XD-NE-BD	XD 系列 PLC 扩展 BD, 总线通讯功能, X-NET 标准线接口, 此 BD 板还可用作 RS485 通讯扩展板；
XSLH 系列	
XL-EnXmY	n 点输入, m 点输出, PNP/NPN 型输入, 输入滤波时间可调, 继电器/晶体管输出 (注: -A 类型的扩展模块为牛角端子, 需配合端子排以及专用扩展电缆同时使用)
XL-E4AD	14Bit 4 通道模拟量输入 (电压可选 0~10V、0~5V、-5~5V、-10~10V；电流可选 0~20mA、4~20mA、-20~20mA), 一阶系数可调, 每个通道可单独使能, 供电电源 DC24V；
XL-E4AD2DA	14Bit 4 通道模拟量输入 (电压可选 0~10V、0~5V、-5~5V、-10~10V；电流可选 0~20mA、4~20mA、-20~20mA)；12Bit 2 通道模拟量输出模块 (电压电流可选 0~10V、0~5V、-5~5V、-10~10V、0~20mA、4~20mA), 一阶系数可调, 每个通道可单独使能, 供电电源 DC24V；
XL-E4DA	12Bit 4 通道模拟量输出模块 (电压可选 0~10V、0~5V、-5~5V、-10~10V；电流可选 0~20mA、4~20mA), 一阶系数可调, 每个通道可单独使能, 供电电源 DC24V；
XL-E8AD-A	14Bit, 8 通道模拟量输入 (电流可选 0~20mA、4~20mA、-20~20mA), 供电电源 DC24V；

型号	功能
XL-E8AD-A-S	16Bit, 8 通道模拟量输入 (电流可选 0~20mA、4~20mA、-20~20mA), 供电电源 DC24V;
XL-E8AD-V	14Bit, 8 通道模拟量输入 (电压可选 0~10V、0~5V、-10~10V、-5~5V), 供电电源 DC24V;
XL-E8AD-V-S	16Bit, 8 通道模拟量输入 (电压可选 0~5V、0~10V、-5~5V、-10~10V), 供电电源 DC24V;
XL-E4TC-P	4 通道热电偶, 支持多种类型热电偶温度传感器用模拟输入, 分辨率 0.1°C, 4 通道独立输出 PID 参数, 供电电源 DC24V;
XL-E4PT3-P	-100~500°C, 4 路 PT100 (三线制) 温度采集, 分辨率 0.1°C, 模块自带 PID 控制输出功能, 供电电源 DC24V;
XL-E1WT-D	可采集一路压力传感器的模拟量电压信号 (-20~20mV), 24 位的高精度 A/D 转换, 采用 Δ - Σ ADC 的 A/D 转换方式, 更高更快的 CPU 处理速度, 更加优化的算法, 抗共振性能更好, 供电电源 DC24V;
XL-E2WT-D	可采集二路压力传感器的模拟量电压信号 (-20~20mV), 24 位的高精度 A/D 转换, 采用 Δ - Σ ADC 的 A/D 转换方式, 更高更快的 CPU 处理速度, 更加优化的算法, 抗共振性能更好, 供电电源 DC24V;
XL-E4WT-D	可采集四路压力传感器的模拟电压信号 (-20~20mV), 24 位高精度 AD 转换, 采用 Δ - Σ ADC 的 A/D 转换方式, 更高更快的 CPU 处理速度, 更加优化的算法, 抗共振性能更好, 供电电源 DC24V;
XL-ETR	当 XL 系列扩展模块数量超过 5 个及以上时增加此终端电阻模块;
XL-P50-E	XL 系列电源模块, AC220V 输入, DC24V 输出, 输出功率 50W;
XSLH 系列	
XL-NES-ED	XL 系列 PLC 扩展 ED 模块, 可以扩展 1 个 RS232 或者 RS485 通讯口; (注: 只能使用其一)
XSF 系列	
XF-E16X	16 通道数字量输入, 支持 NPN、PNP 输入
XF-E16(P)YT	16 通道数字量输出, 支持 NPN、PNP 输出
XF-E8NX8YT	8 通道数字量输入, 支持 NPN 输入; 8 通道数字量输出, 支持 NPN 输出
XF-E32X	32 通道数字量输入, 支持 NPN、PNP 输入
XF-E32YT	32 通道数字量输出, 支持 NPN 输出
XF-E16X16YT	16 通道数字量输入, 支持 NPN、PNP 输入, 16 通道数字量输出, 支持 NPN 输出
XF-E4AD	4 通道模拟量输入, 支持电流、电压输入
XF-E4DA	4 通道模拟量输出, 支持电流、电压输出
XF-E2COM24	2 个串口通道, 支持 232、485 通讯
XF-E4RTD	4 通道热电阻温度采集, 供电电源 DC24V,
XF-E4TC	4 通道热电偶温度采集, 适用于纯采集的场景, 没有输出通道, 供电电源 DC24V
XF-E1HSC	高速计数扩展模块, 供电电源 DC24V
XF-E2HSP	8 个通道数字量输入, 6 个通道数字量输出。可支持 2 路脉冲输出, 最大输出频率 200KHz。也可以配置为普通输入/输出

1-1-2. 系统构成



1-2. XS Studio 概述

1-2-1. XS Studio 简介

XS Studio 是基于 CODESYS 开发,面向 XS 系列的编程组态软件。集成了 PLC 编程、可视化 HMI、安全 PLC、控制器实时核、现场总线及运动控制,能够提供一套完整的配置、编程、调试、监控环境,可以灵活自由地处理功能强大的 IEC 语言。

- ◆ 强大的软件仿真、在线调试及程序检查能力,不需要连接 PLC 硬件,即可完成程序调试仿真。
- ◆ 具备方便的产品配置功能,可以轻松快速的实现包括 CPU 配置、IO 模块配置和高速 IO 等。
- ◆ 提供智能的调试查错功能。当用户输入了错误的应用程序代码时,立刻会接收到编译器发出的语法错误警告及错误信息,让编程人员可以迅速做出相应纠正。
- ◆ 强大的运动控制模块。基于 PLCopen 的工具包 可以实现单轴、多轴运动,电子凸轮传动,电子齿轮传动,复杂多轴 CNC 控制等。

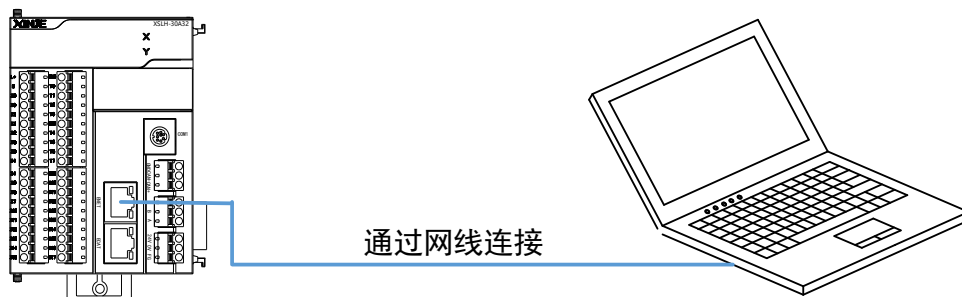
1-2-2. XS Studio 与硬件的连接



XS 系列 PLC 的串口不支持与 XS Studio 软件联机,仅用于串口相关协议通讯。

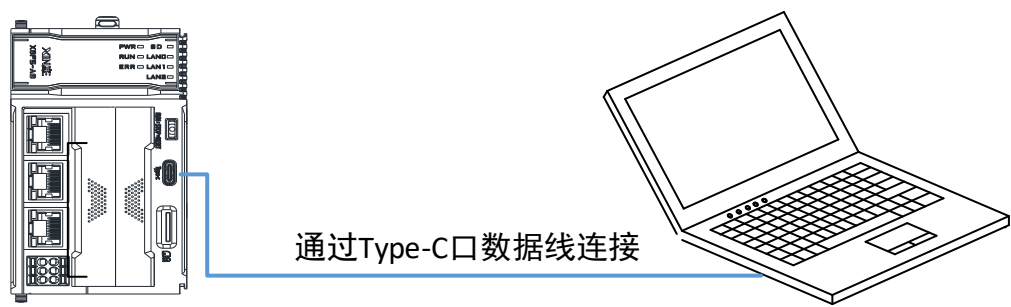
1-2-2-1. Ethernet 口连接

XS 系列 PLC 都支持用 Ethernet 网口连接电脑与 XS Studio 软件联机,用 XS Studio 软件将程序下载到 PLC 后进行程序监控并控制 PLC。



1-2-2-2. Type-C 口连接

XS F 系列 PLC 除了支持用 Ethernet 网口连接电脑与 XS Studio 软件联机外，还额外支持用 Type-C 口连接电脑与 XS Studio 软件联机。



1-2-3. 软件安装&卸载

1-2-3-1. 配置要求&软件获取

1、系统配置要求

项目	要求
操作系统	windows 7、windows 8、windows 10 或 windows 11 操作系统，推荐 64 位操作系统
内存	4GB 及以上
硬盘空间	12GB 以上



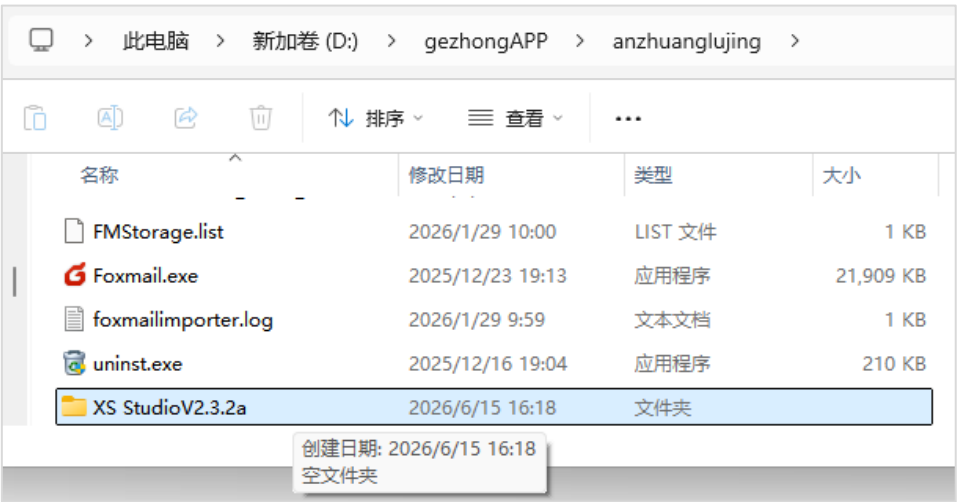
- XS Studio 软件安装在非系统盘路径下时，也需注意确保系统盘至少预留 10GB 以上的硬盘空间以确保软件可以正常安装。

2、软件获取

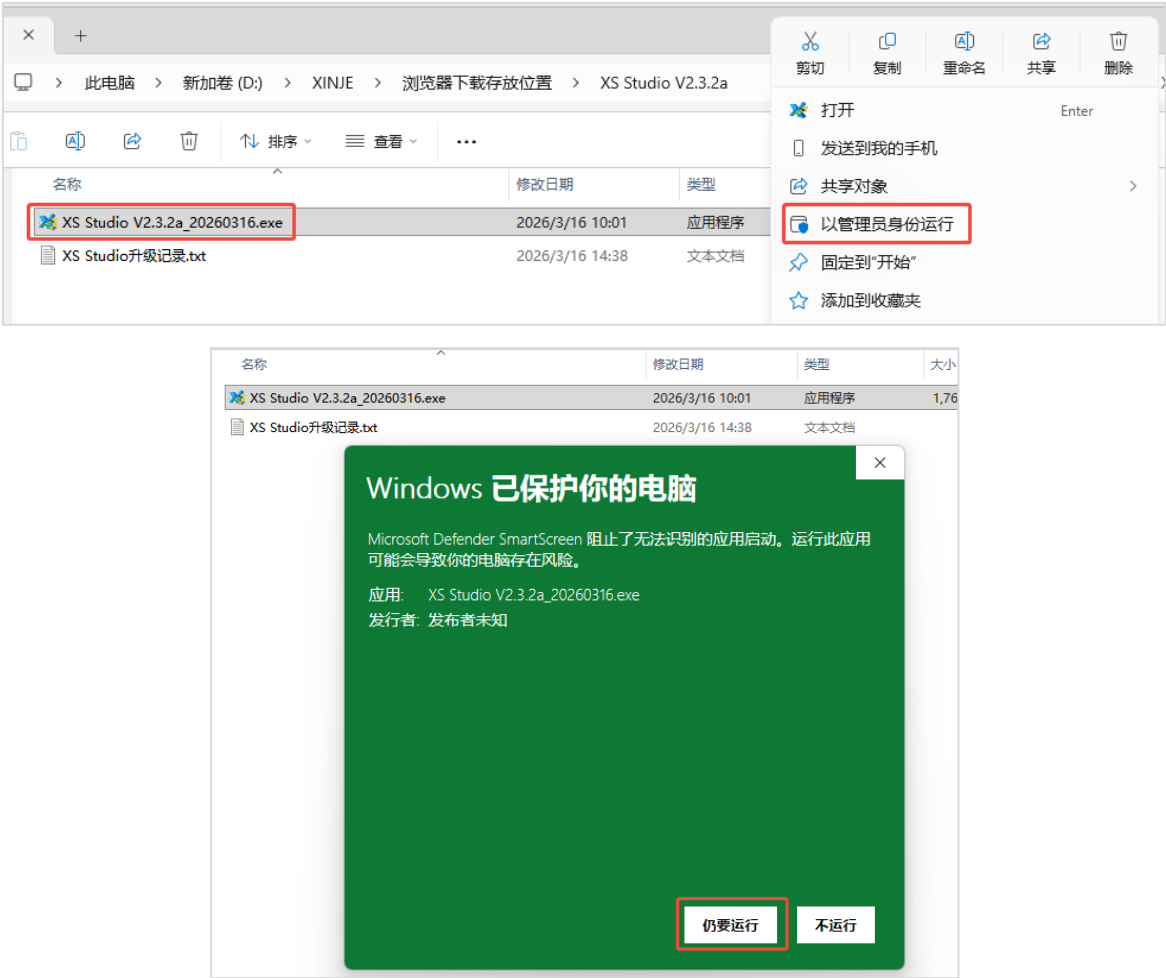
用户可以在“信捷官网”-“服务与支持”-“下载中心”搜索并下载软件安装包进行安装，信捷官网网址：<https://www.xinje.com/>。

1-2-3-2. 安装步骤

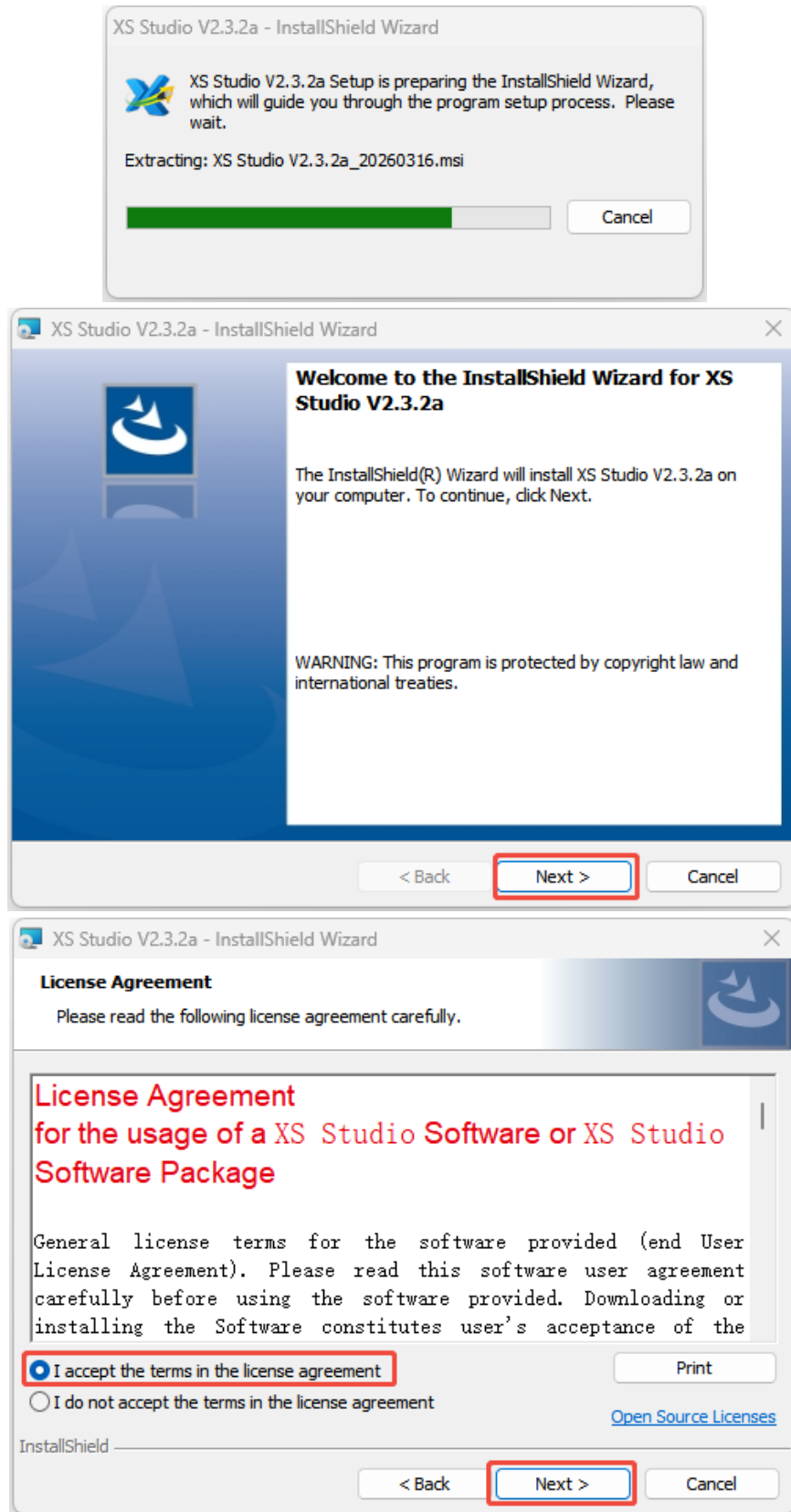
- 1、用户自主选择XS Studio 软件安装的电脑磁盘位置，新建一个空白文件夹作为软件的安装路径。

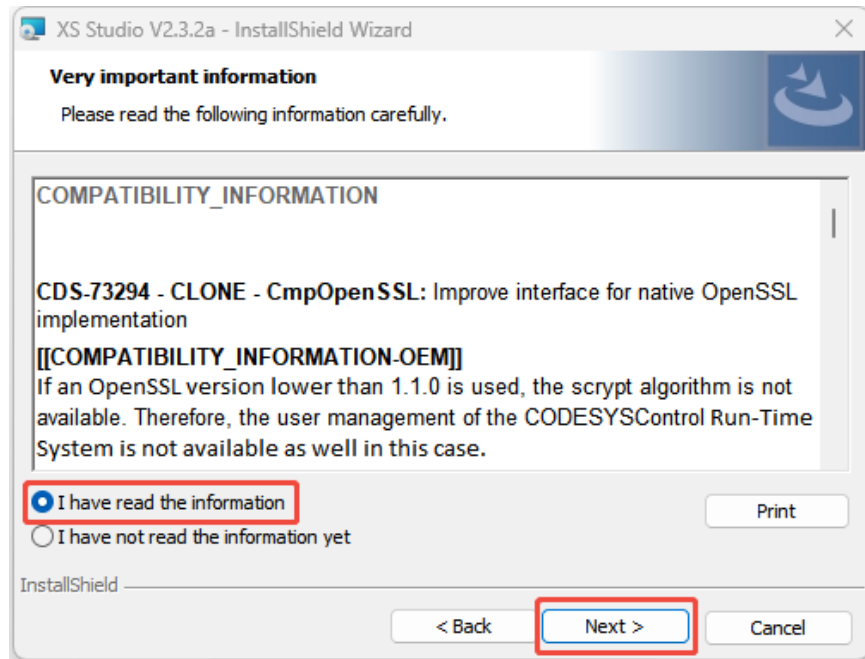


2、将从官网下载的 XS Studio 软件安装包先解压缩，然后右键以管理员身份运行 XS Studio.exe 应用程序。

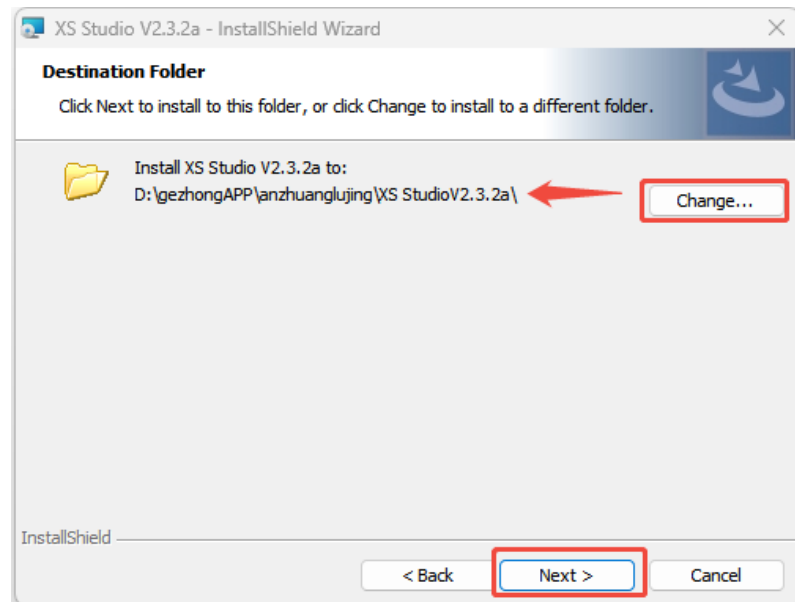


3、在弹出的安装对话框里点击“Next”，然后点选“同意协议条款”和“Next”、“已经阅读相关信息”和“Next”。

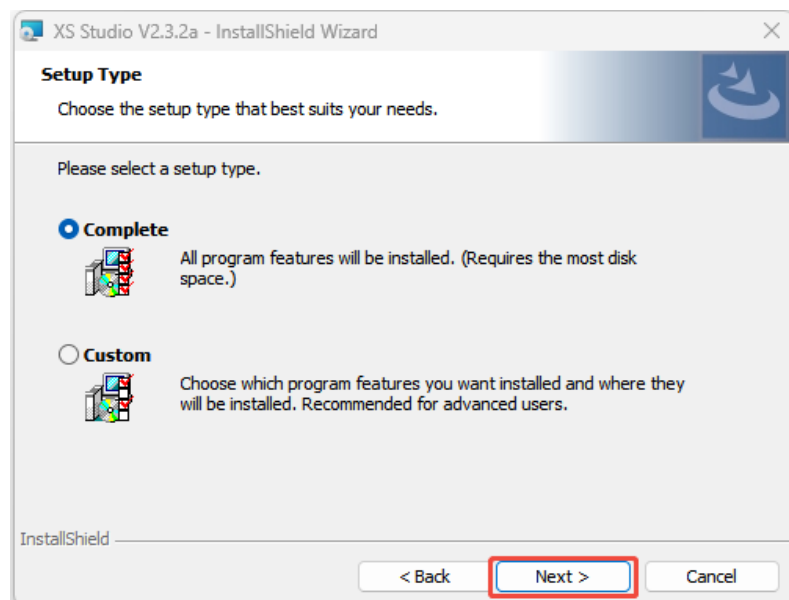


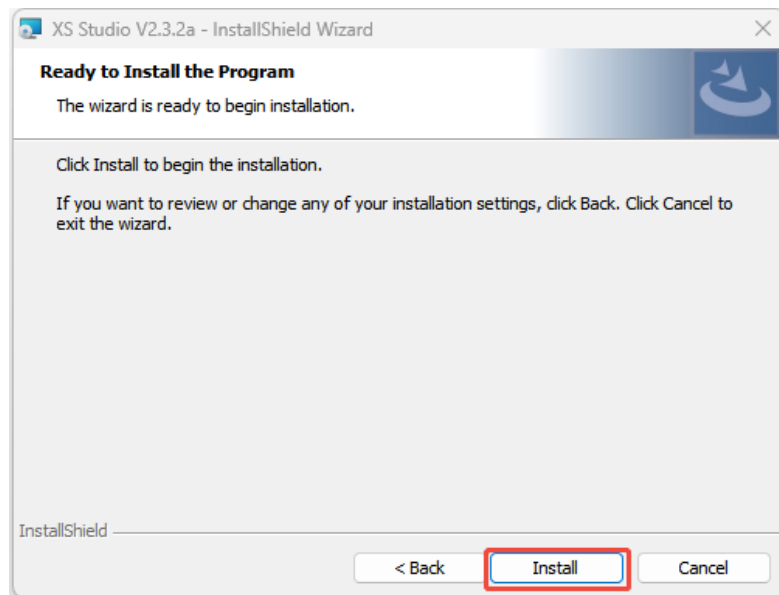


4、安装路径选择一开始新建好的空白文件夹，然后再次点击“Next”。

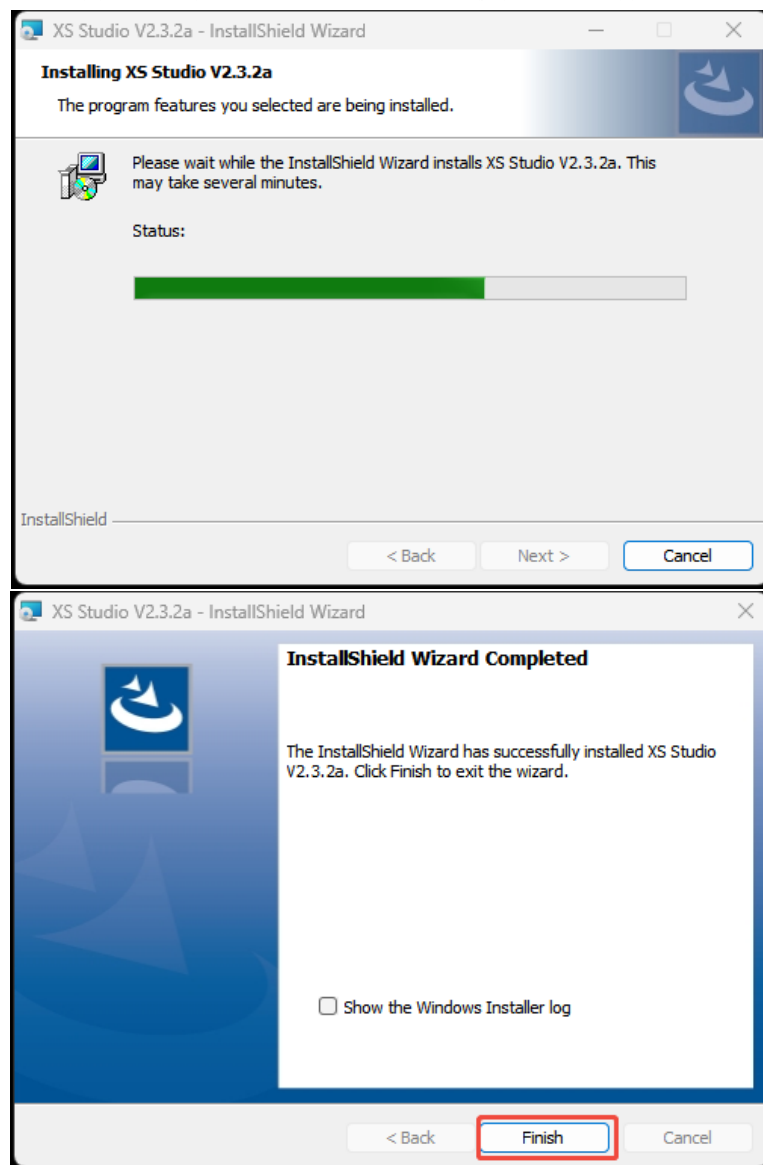


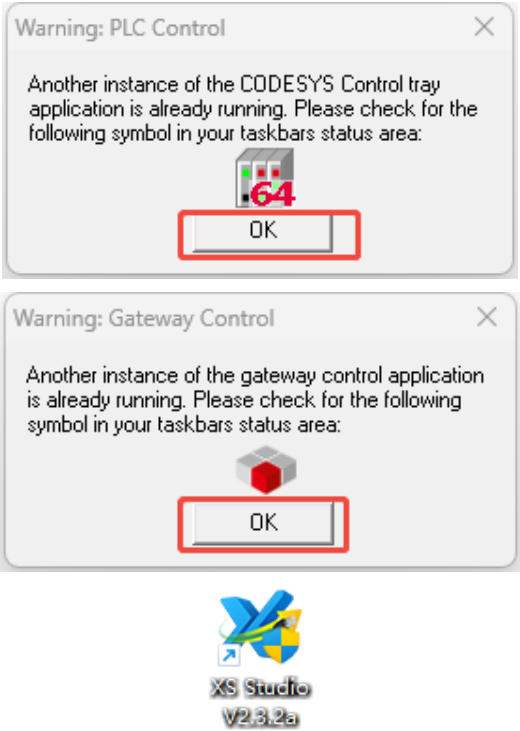
5、依次点击“Next”、“Install”。





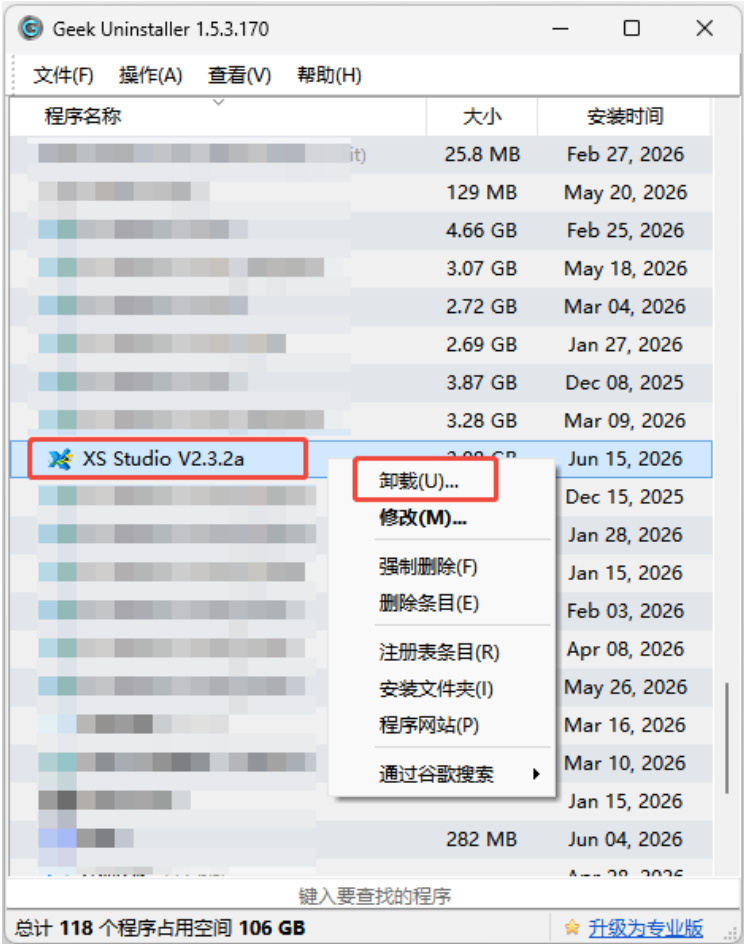
6、耐心等待安装进度条完整结束，之后点击“Finish”，便完成了软件的安装。

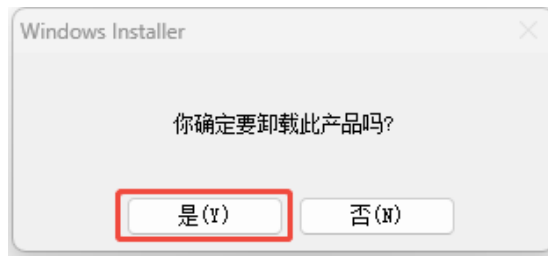




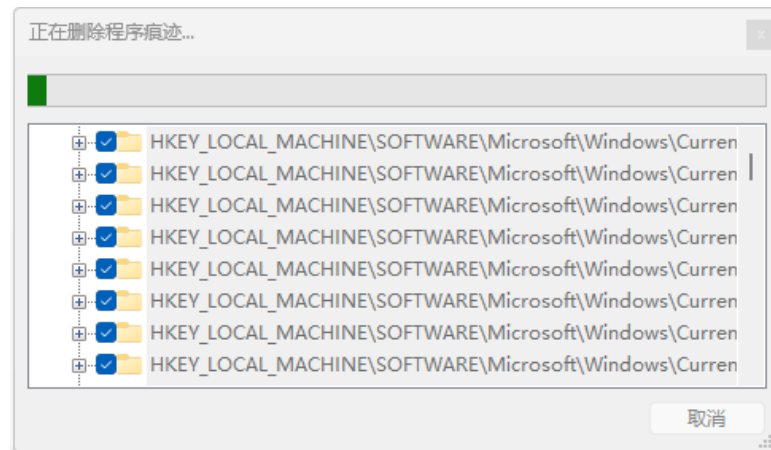
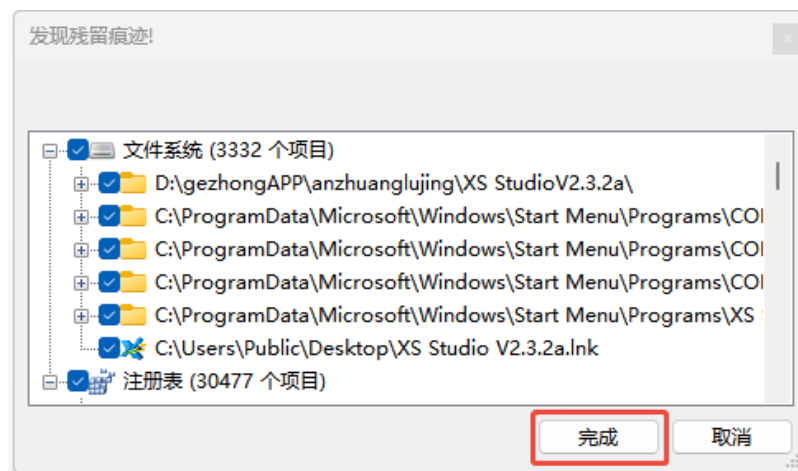
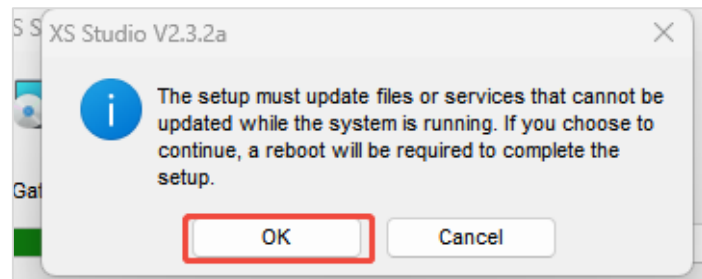
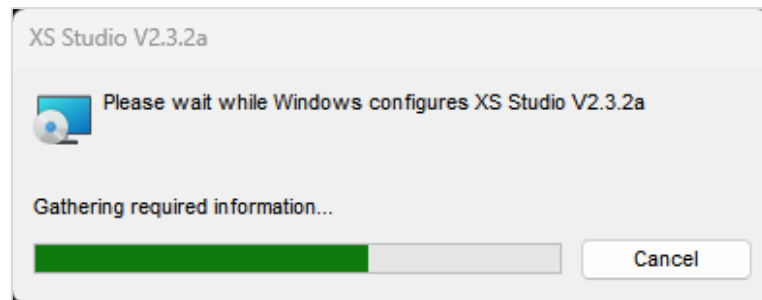
1-2-3-3. 卸载步骤

- 1、运行电脑上的 Geek 软件，选择所需卸载的 XS Studio 软件后点击鼠标右键，点击“卸载”，确认卸载。





2、卸载进度条结束后会弹出“发现残留痕迹”对话框，此处点击“完成”让其删除残留痕迹，删除完毕后点击“关闭”，此时用户原先在软件安装路径自主创建的“XS StudioV2.3.2a”文件夹等都被彻底删除。软件已完成卸载。



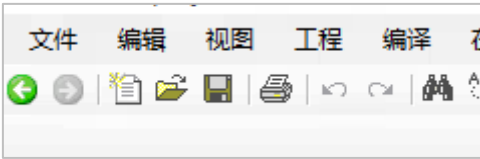


- 使用电脑系统自带的卸载功能卸载 XS Studio 软件时可能出现部分文件无法彻底删除的情况，此时用户可以手动操作将剩余文件删除。若遇到电脑权限问题无法删除时（权限问题与软件无关），则可以根据遇到的提示自行百度查询 windows 权限操作步骤来解决。
- 推荐使用 Geek 软件卸载 XS Studio 软件。可百度搜索其官网下载使用软件。

2. 快速入门

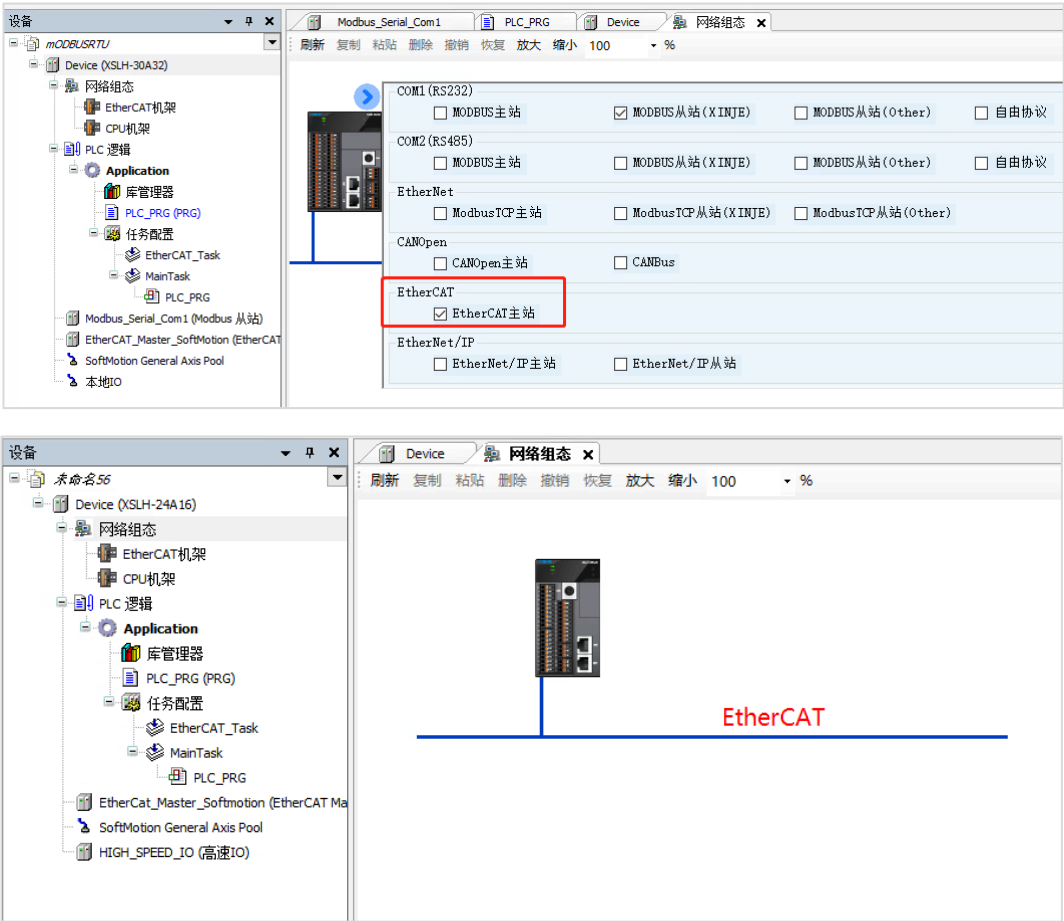
2-1. 界面导航

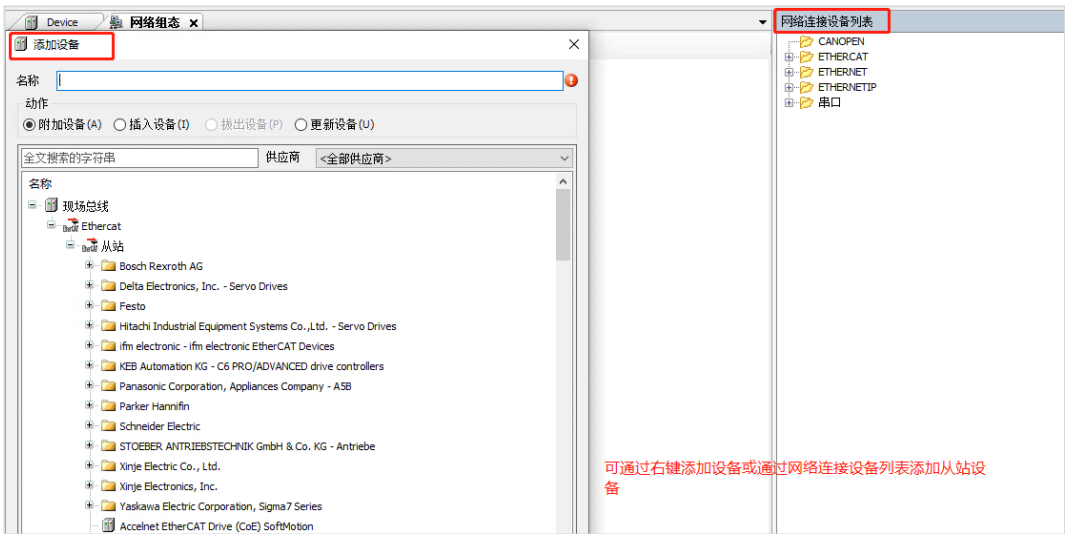
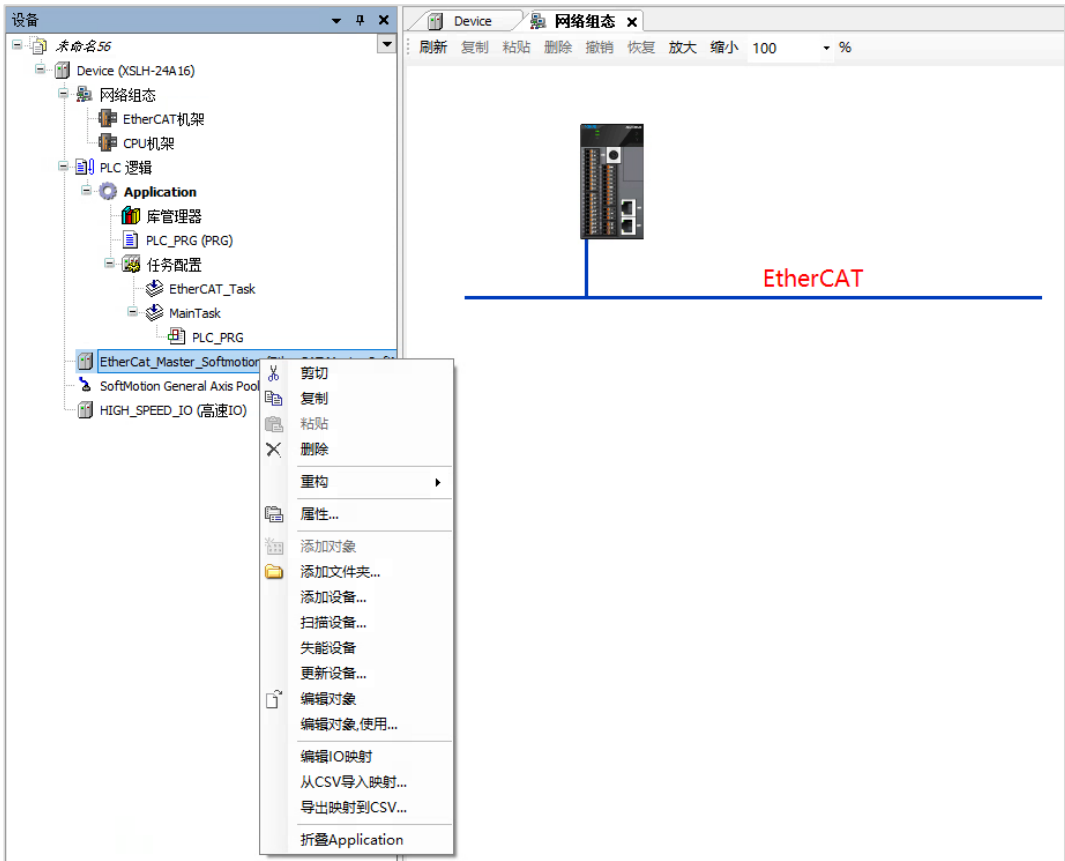
向左和向右的按钮，分别代表返回到上一次编辑位置和恢复到下一个编辑位置。鼠标点击后，可帮助用户更快定位修改用户程序位置。



2-2. XS Studio 编程举例

根据实际拓扑配置对应设备。



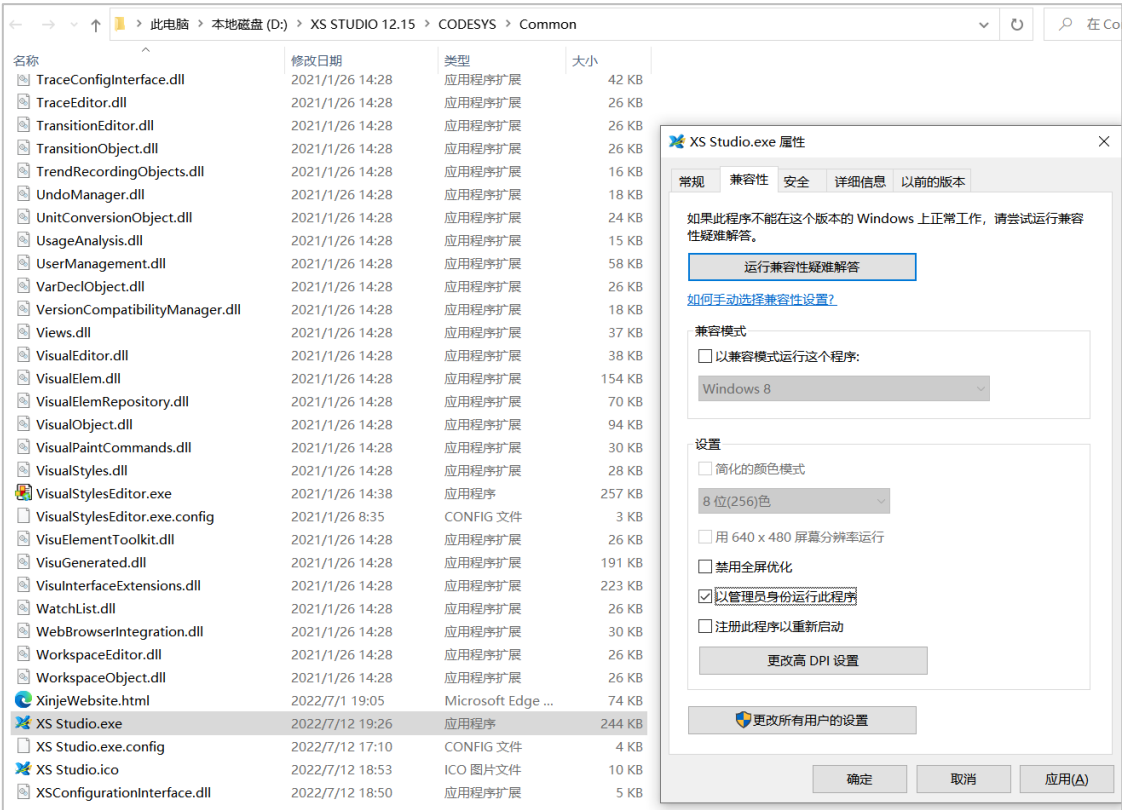


2-2-1. 基本编程操作

2-2-1-1. 启动 XS Studio

1、设置管理员权限

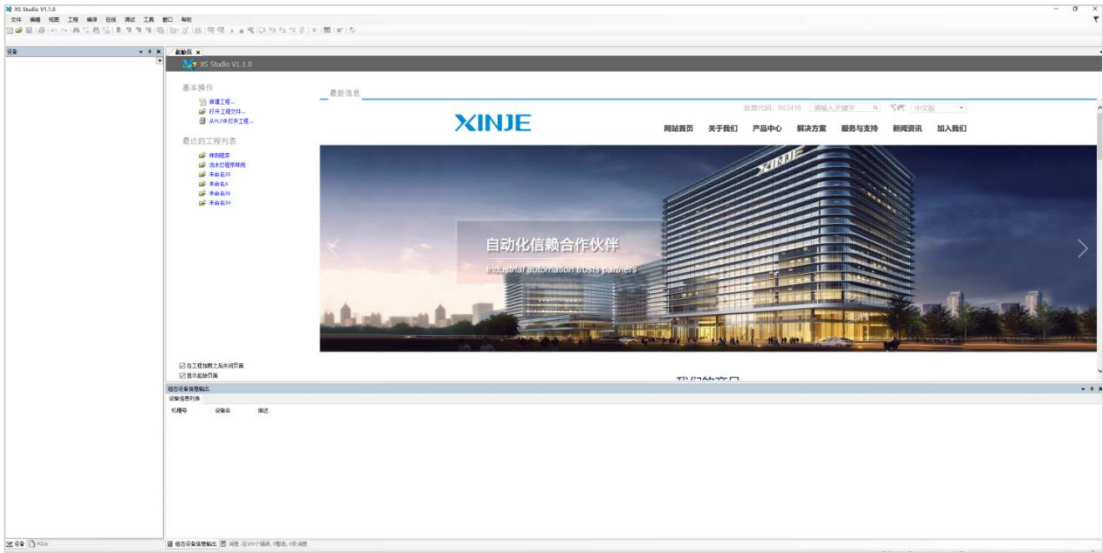
在 Win7 系统下需要以管理员权限打开软件，在 XS Studio 的默认安装路径下找到 XS Studio.EXE 文件，选中该文件后点击鼠标右键，选择属性，将“Run This Program as an administrator”或者是“以管理员身份运行此程序”的勾选上，点击“OK”确认，如图所示，确认后每次运行 XS Studio 系统则会默认以管理员权限自动进入 XS Studio。



2、启动 XS Studio

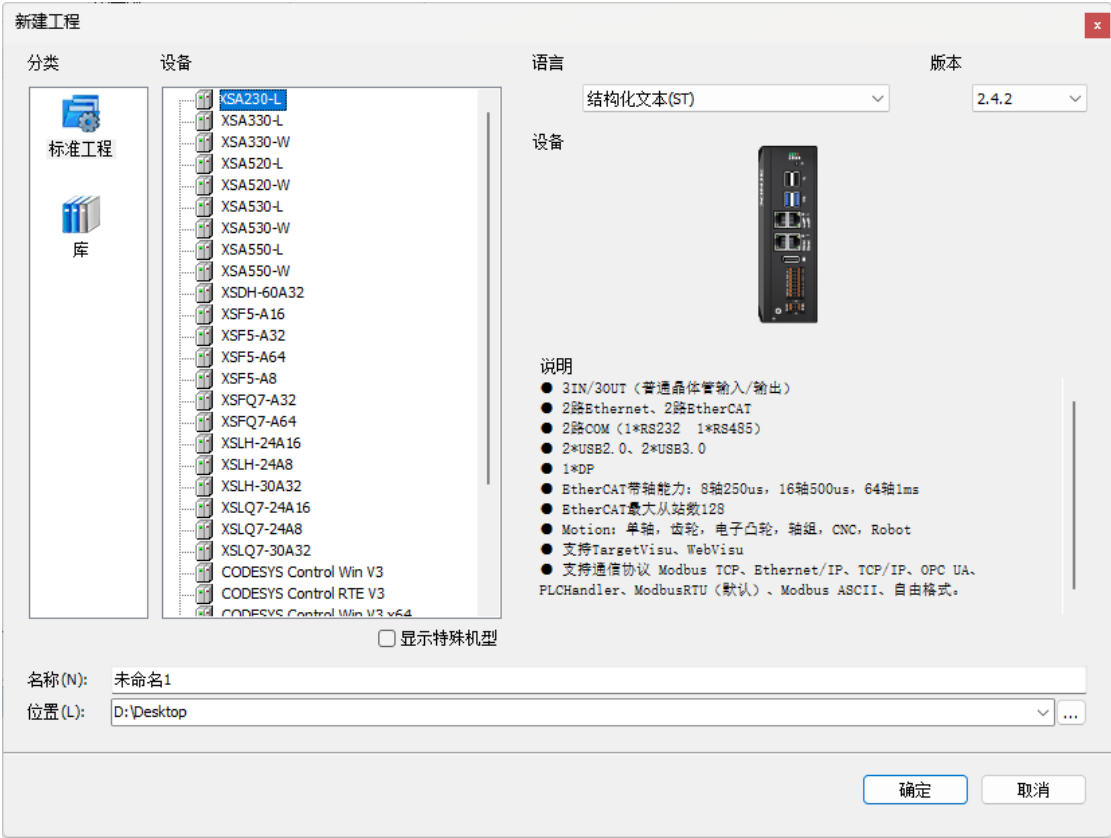


从开始菜单选择>XS Studio > XS Studio 或直接双击桌面上的图标启动 XS Studio。



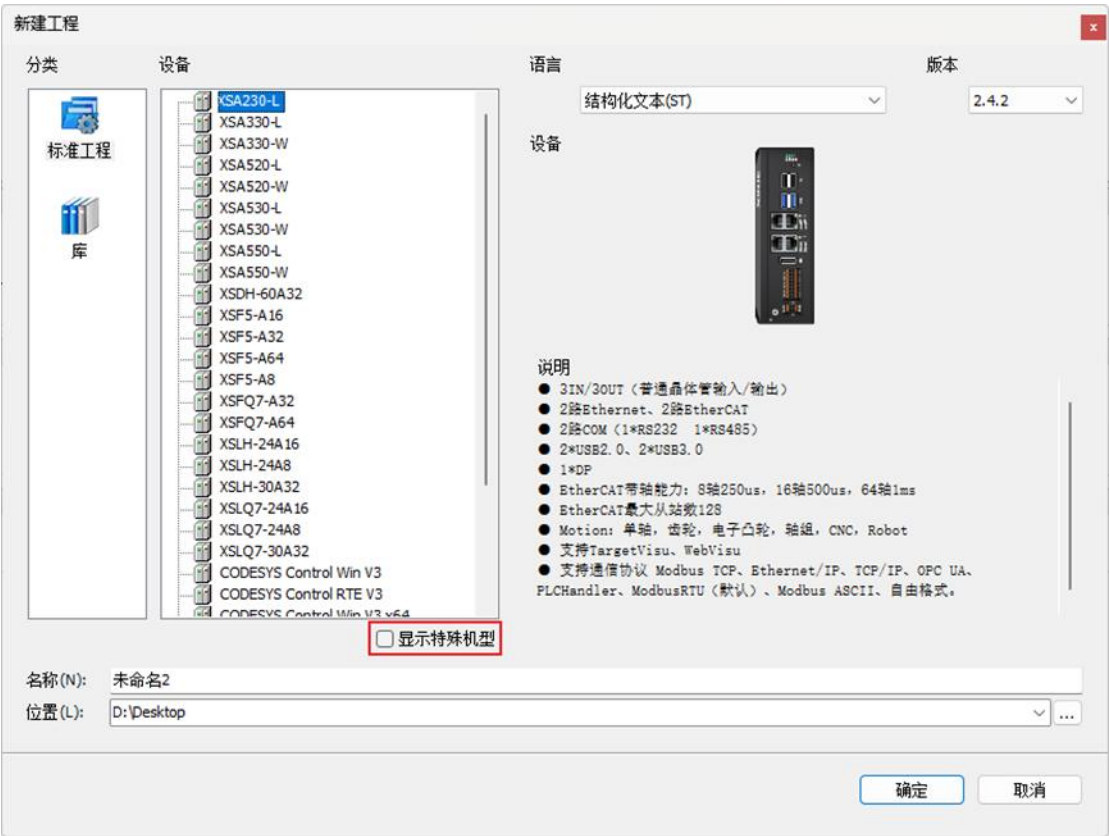
3、创建工程项目

点击  图标，新建工程。选择“标准工程”，选择对应机型，选择熟悉的编程语言，填写工程名称，选择文件保存位置。

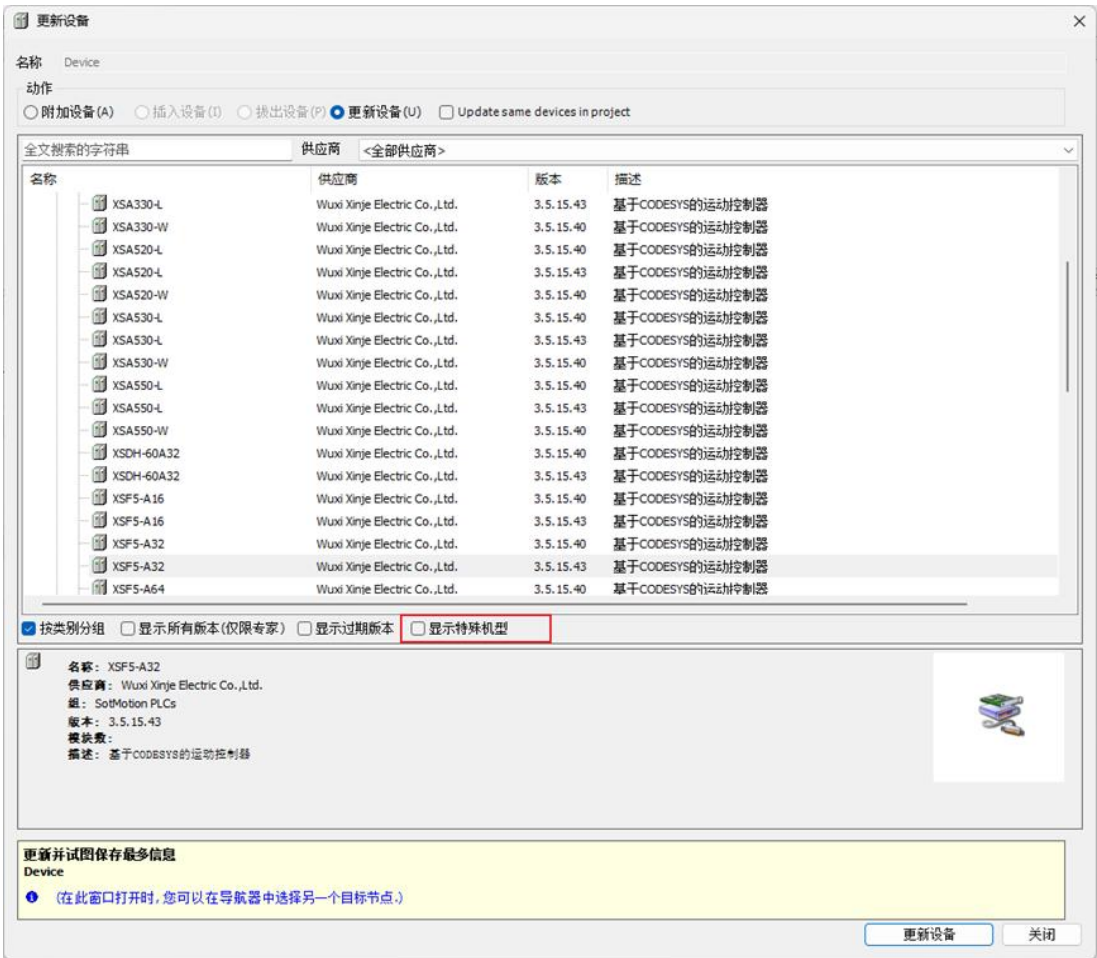


4、显示特殊机型

新建工程选择机型时，有【显示特殊机型】按钮，勾选后可显示定制型号及描述信息，不勾选则不显示定制型号



更新设备界面有勾选【显示特殊机型】按钮；勾选后会显示收敛的定制机型号，不勾选则不显示定制型号。



2-2-1-2. PLC 程序文件的建立

PLC 程序文件的建立，是运行结构的运行顺序的建立，也是编程模式的建立，甚至包括了数据区域的分割。在建立程序文件之前，应当对运行结构进行详细的划分，确定连续型、周期型和事件触发型任务，并对周期型和事件触发型任务安排优先级别。新建一个 XS Studio 项目后，自动生成一个默认连续型任务，任务下有一个默认的程序及 PLC_PRG。

1、创建任务

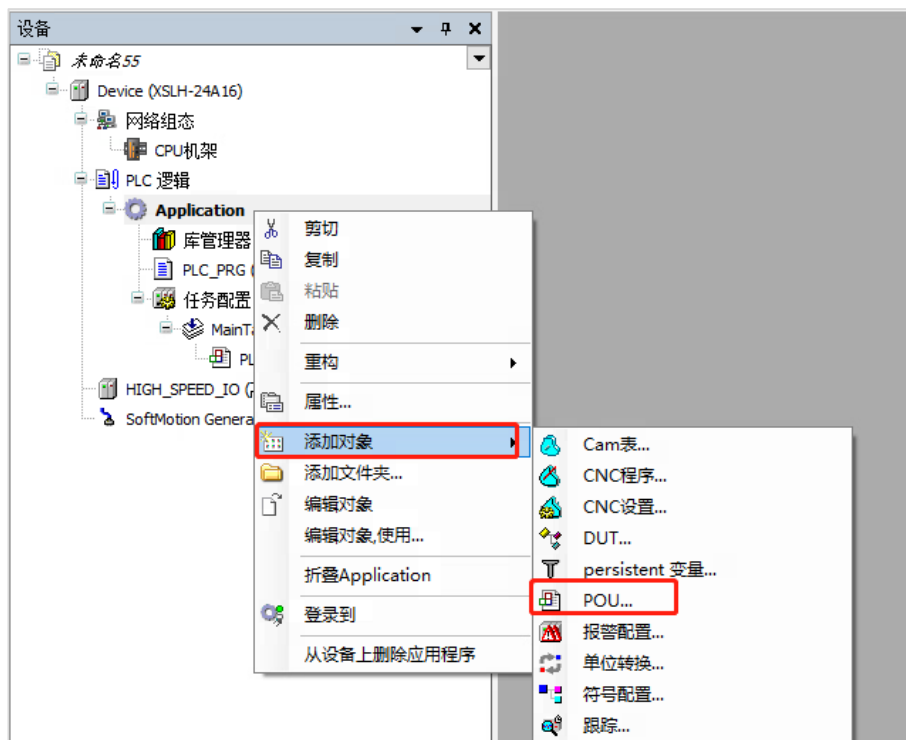
首先，在“任务配置”中管理任务，通常的项目应用中，可以分为主逻辑任务、通讯任务，通讯由于要更新数据源，会将其放在比较高的任务优先等级及较短的循环时间。此外，如项目中涉及到运动控制，也会将其独立出一个任务，并将其放在最高的任务优先等级，如图所示：

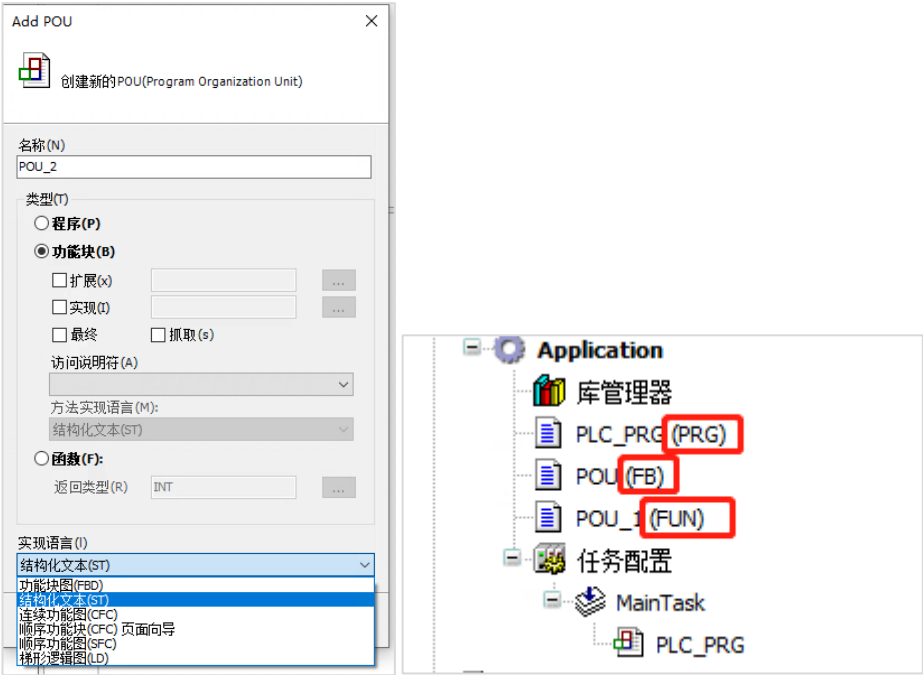


2、添加 POU

1) 自定义程序/功能块/函数

用户可以在项目中使用右键菜单的命令“添加对象”，选择“POU”程序组织单元，会弹出如下图所示的对话框，用户可以选择添加程序，功能块或函数，下拉菜单中可以选择对应的编程语言。添加后，可以在左边的项目设备树中查看程序组织单元括号内对应的属性，FB 为功能块，FUN 为函数，PRG 为程序。

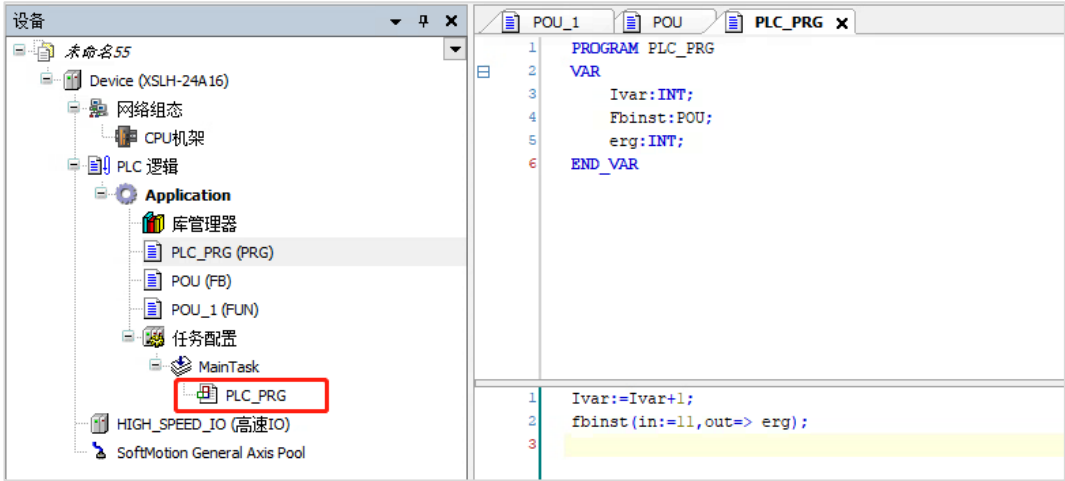




2) 声明变量

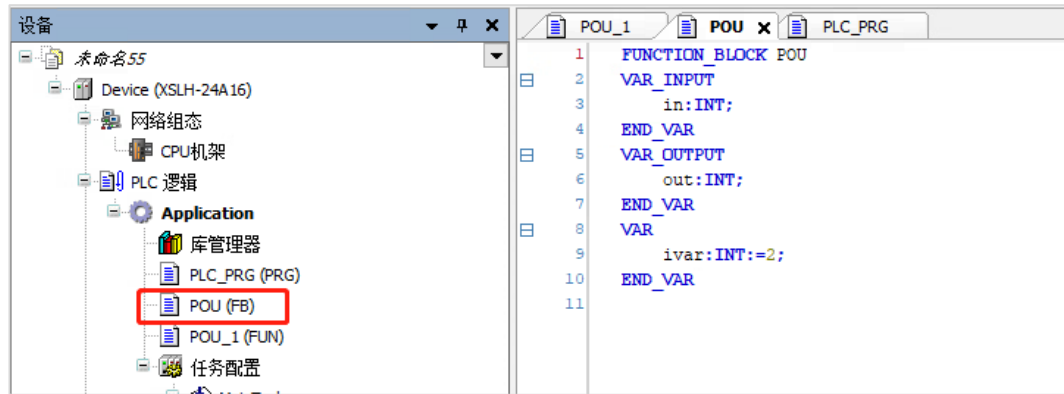
- 在“PLC_PRG”程序内声明变量

双击设备树中的“PLC_PRG”，自动在 XS Studio 用户界面的 ST 语言编辑器中打开。语言编辑器包括声明部分（上部），实现部分（下部），由一个可调的分割线分开。声明部分包括：显示在左侧边框中的行号、POU 类型和名称（如“PROGRAM PLC_PRG”），以及在关键字“VAR”和“END_VAR”之间的变量声明。如下图所示：



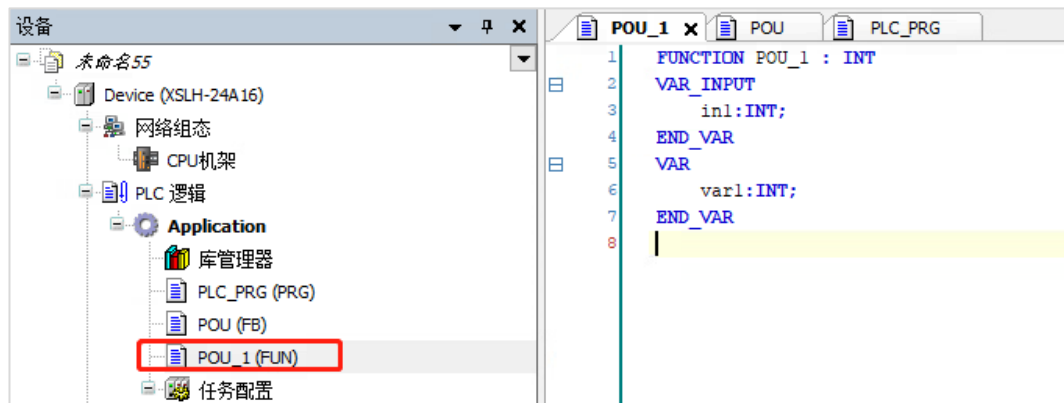
- 在功能块 FB 内声明变量

功能块语言编辑器界面与程序的编辑器界面类似，也包括声明部分与代码部分。用户声明的所有变量最终是给程序组织单元所用，变量声明中可声明接口变量、静态变量和本地变量，具体如下图所示：



● 在函数 FUN 内声明变量

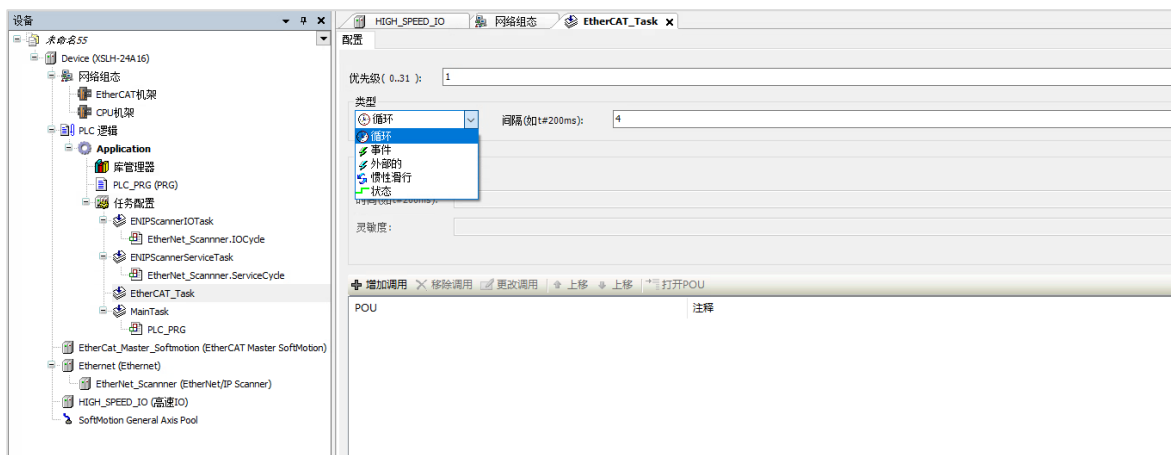
函数是有至少一个输入变量、无私有数据、仅有一个返回值的基本算法单元。函数是没有静态变量的程序组织单元。即用相同的输入参数调用某一函数时，该函数总能生成相同的结果作为函数值（返回值）。函数的一个重要特性是它们不能使用内部变量存储数值，这点与功能块完全不同。具体如下图所示：



2-2-2. 任务配置

一个程序可以用不同的编程语言来编写。典型的程序由许多互连的功能块组成，各功能块之间可互相交换数据。在一个程序中不同部分的执行通过“任务”来控制。“任务”被配置以后，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。在设备树中有“任务管理器”选项卡，使用它除了声明特定的 PLC_PRG 程序外，还可以控制工程内其他子程序的执行处理。任务是用于规定程序组织单元在运行时的属性，它是一个执行控制元素，具有调用的能力。在一个任务配置中可以建立多个任务，而一个任务中，可以调用多个程序组织单元，一旦任务被设置，它就可以控制程序周期执行或者通过特定的事件触发开始执行。

在任务配置中，用名称、优先级和任务的启动类型来定义它。这启动类型可以通过时间（周期的，随机的）或通过内部或外部的触发任务时间来定义，例如使用一个布尔型全局变量的上升沿或系统中的某一特定事件。对每个任务，可以设定一串由任务启动的程序。如果在当前周期内执行此任务，那么这些程序会在一个周期的长度内被处理。优先权和条件的结合将决定任务执行的时序，任务设置界面如下图所示：



在任务配置时有如下的属性，编程者需遵循如下规则：

- ◆ 循环任务的最大数为 100。
- ◆ 惯性滑行任务的最大数为 100。
- ◆ 事件触发任务的最大数为 100。
- ◆ 根据目标系统，PLC_PRG 可能会在任何情况下作为一个自由程序执行，而不用插入任务配置中。
- ◆ 处理和调用程序是根据任务编辑器内自上而下的顺序所执行的。

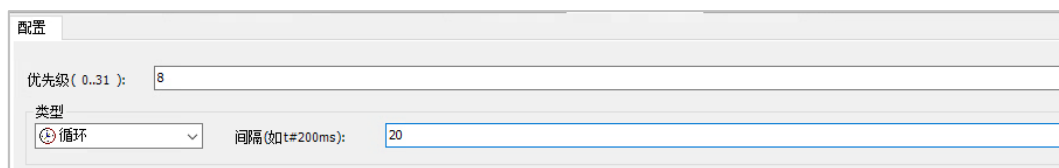
1、任务优先级 Priority

XS Studio 中的可以对任务的优先级进行设置，一共可以设 32 个级别（0~31 之间的一个数字，0 是最高优先级，31 是最低优先级）。当一个程序在执行时，优先级高的任务优先于优先级任务低的任务，高优先级任务 0 能中断同一资源中较低优先级的程序执行，使较低优先级程序执行被放缓。



在任务优先级等级分配时，请勿分配具有相同优先级的任务。如果还存在其他任务试图先于具有相同优先级的任务，则结果可能不确定且不可预知。

如果任务的类型为“循环”，则按照“间隔”中的时间循环执行，具体设置如下图所示：



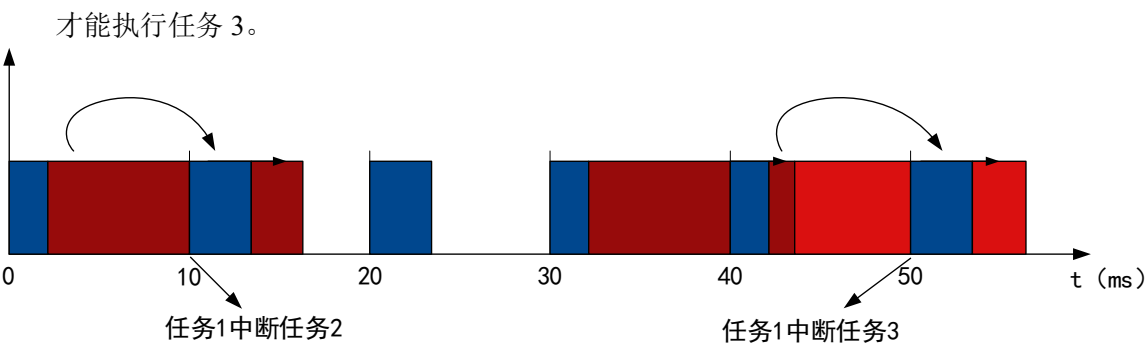
应用举例：

假设有 3 个不同的任务，分别对应三种不同的优先等级，具体分配如下：

- ◆ 任务 1 具有优先级 0 和循环时间 10ms；
- ◆ 任务 2 具有优先级 1 和循环时间 30ms；
- ◆ 任务 3 具有优先级 2 和循环时间 40ms。

在控制器内部，各任务的时序关系如图所示，具体说明如下：

- ◆ 0~10ms: 先执行任务 1（优先级最高），如在本周期内已将程序执行完，剩余时间执行任务 2 程序。但是如果此时任务 2 没有被完全执行完，但时间已经到了第 10ms 时，由于任务 1 是每 10ms 执行一次的且优先级更高，此时将会打断任务 2 的执行。
- ◆ 10~20ms: 先将任务 1 的程序执行完毕，如有剩余时间，再执行上个周期为完成的任务 2。
- ◆ 20~30ms: 由于任务 2 是每 30ms 执行一次的，在 10~20ms 之间任务 2 已经全部执行完毕，此时不需要再执行任务 2，只需将优先级最高的任务 1 执行一次即可。
- ◆ 30~40ms: 与之前类似。
- ◆ 40~50ms: 此时出现了任务 3，任务 3 的优先级更低，所以只有在确保任务 2 彻底执行完后，



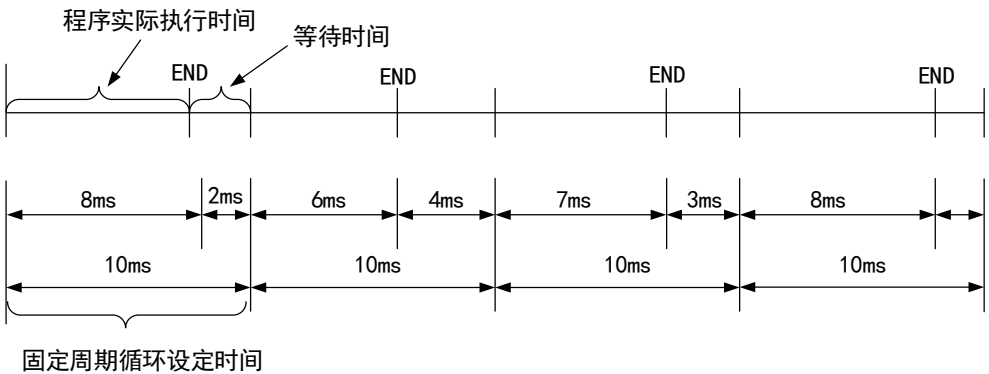
2、任务的执行类型

针对每个独立的任务可以对其进行执行的类型编辑及配置。包括固定周期循环、事件触发、外部触发、惯性滑行触发和状态触发 5 种类型。

1) 固定周期循环

根据程序中所使用的指令执行与否，程序的处理时间会有所不同，所以实际执行时间在每个扫描周期都发生不同的变化，执行时间有长有短。通过使用固定周期循环方式，能保持一定的循环时间反复执行程序。即使程序的执行时间发生变化，也可以保持一定的刷新闻隔时间。在这里，也推荐大家优先选择固定周期循环任务启动方式。

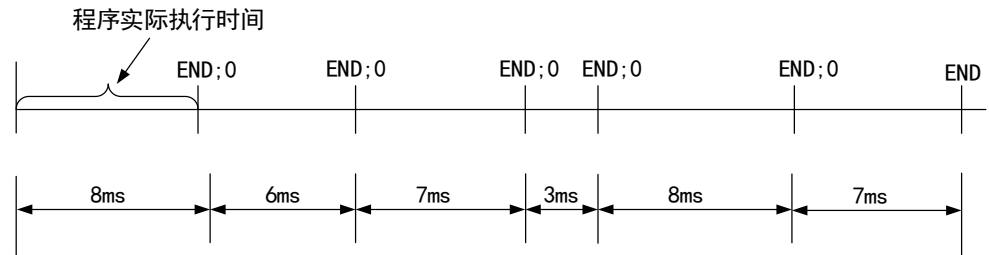
例如，假设将程序对应的任务设定为固定周期循环方式，间隔时间设定为 10ms 时，实际程序执行的时序图如下图所示。



如果程序实际执行时间在规定的固定周期循环设定时间内执行完，则空余时间用作等待。如应用中还有优先级较低的任务未被执行，则剩下的等待时间用来执行低优先级的任务。

2) 惯性滑行

程序一开始运行任务就被处理，一个运行周期结束后任务将在下一个循环中被自动重新启动。不受程序扫描周期（间隔时间）的影响。即确保每次执行完程序的最后一条指令后才进入下一个循环周期。否则不会结束该程序周期。



该执行方式因为没有固定的任务时间，所以每次执行的时间可能都不一样。故不能保证程序的实时性，在实际的应用中选用此方式的场合较少。

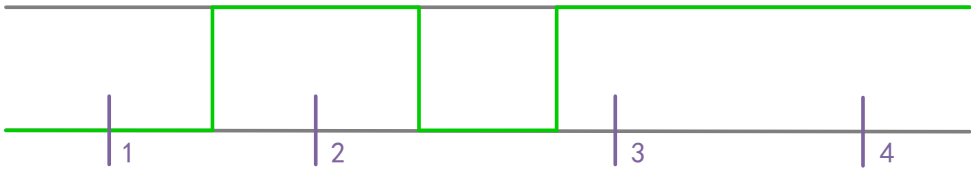
3) 事件触发

如果事件区域的变量得到一个上升沿，任务开始。

4) 状态触发

如果事件区域的变量为 TRUE，任务开始。

下图中针对事件触发和状态触发分别进行了比较，绿色实线为两种触发方式选择的布尔变量状态，下表为比较的结果。



任务输入触发信号

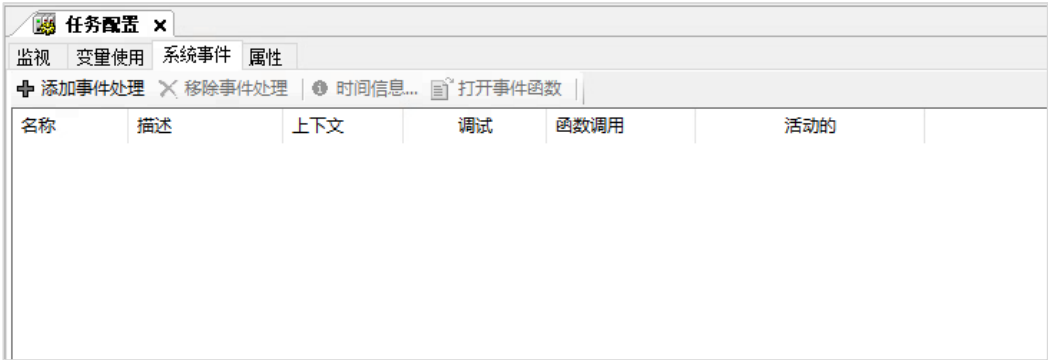
状态触发方式与事件触发功能类似，区别在于状态触发的触发变量只要为 TRUE 程序就执行，为 FALSE 则不执行。而事件触发只采集触发变量的上升沿有效信号。

在采样点 1-4（紫色）不同类型的任务展示了不同的反应。这个具体的事件为 TRUE 完成了状态驱动任务的条件，然而一个事件驱动任务需要事件从 FALSE 变为 TRUE。如果任务计划的采样频率过低，事件的上升沿可能检测不到。

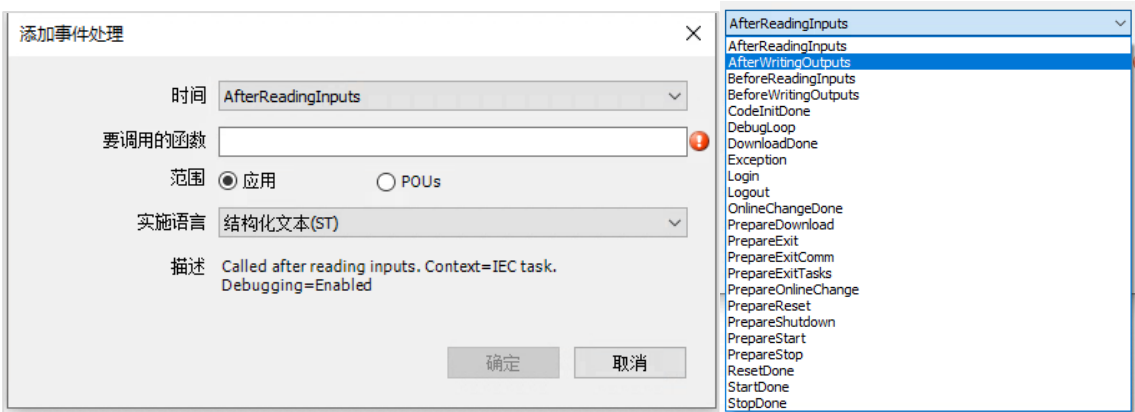
执行点	1	2	3	4
事件触发（Event）	不执行	执行	执行	不执行
状态触发（Status）	不执行	执行	执行	执行

5) 系统事件

用户可选择的系统事件是根据实际的硬件目标系统而定的，由目标系统对应的库文件提供相应的系统事件，所以不同的目标硬件设备对应的系统事件可能会不同。但通常来说，通用的系统事件有：停止，开始，登入，改变等。在任务配置中，可以对任务配置中的系统事件进行设置。



用户可以通过鼠标选择“任务配置”-->“系统事件”进入添加事件处理界面，单击“添加事件处理”按钮即可进行添加系统事件。支持用户可下拉选择时间，分别如下图所示：



6) 外部中断

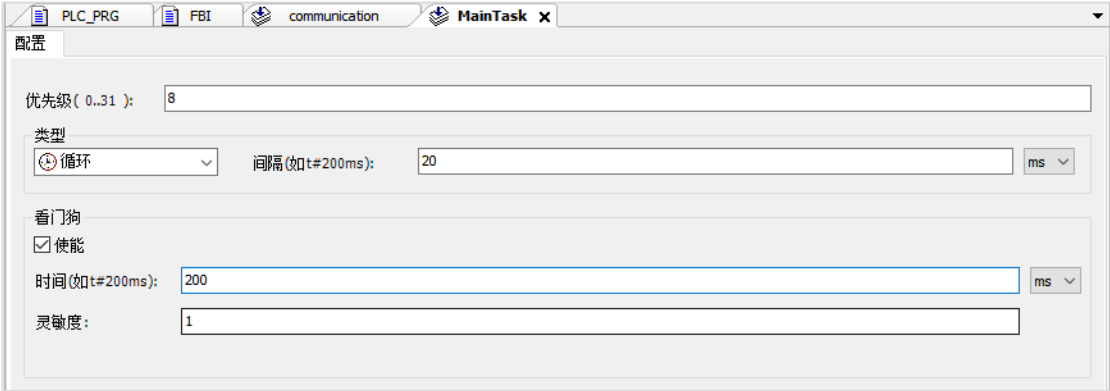
如果事件区域的变量得到一个外部中断信号 X 的上升沿或者下降沿，任务开始。

输入端子 X 可以作为外部中断的输入用，每一输入端对应于一个外部中断，也可指定上升沿或下降沿或上升下降沿作为触发条件。

7) 看门狗

看门狗是一种控制器硬件式的计时设备，XS Studio 内可以通过“任务配置”对其进行使能，默认不使用看门狗功能。

看门狗的主要功能是监控程序执行时出现的异常或内部时钟发生的故障。如当系统出现死机或当程序进入死循环时，看门狗计时器就会对系统发出重置信号或停止 PLC 当前运行的程序。我们可以形象的将它理解为一只小狗需要主人定时的去给它喂食，如果超过规定的时间没有喂，则他马上就会饿。要配置看门狗，必须定义两个参数，时间和灵敏度，看门狗的配置如图所示：



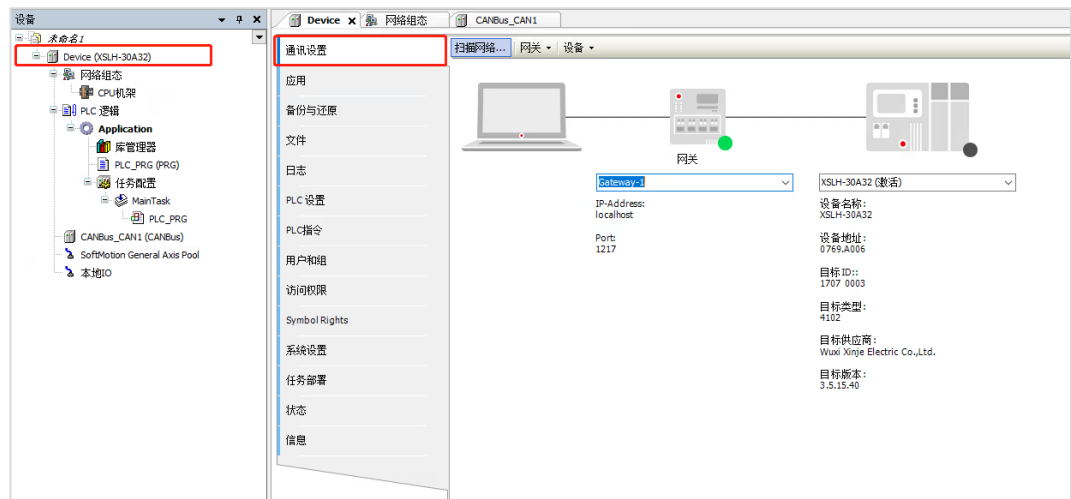
配置项	说明
时间	XS Studio 针对每个任务可以配置独立的看门狗。如果目标硬件支持长看门狗时间设置，则可以设置上限和下限。默认的看门狗时间单位为毫秒（ms）。如果程序执行周期超过看门狗触发时间，将激活看门狗功能，并将中止当前任务。
灵敏度	“灵敏度”用于定义必须在控制器检测到应用程序错误之前发生的任务看门狗例外数。默认为 1。

最终的看门狗触发时间=时间×灵敏度。如果程序实际执行时间超过看门狗触发时间，则激活看门狗。例如，时间为 10ms，灵明度设为 5，则看门狗触发时间为 50ms，一旦任务的执行时间超过 50ms，则立即激活看门狗并将任务中止。

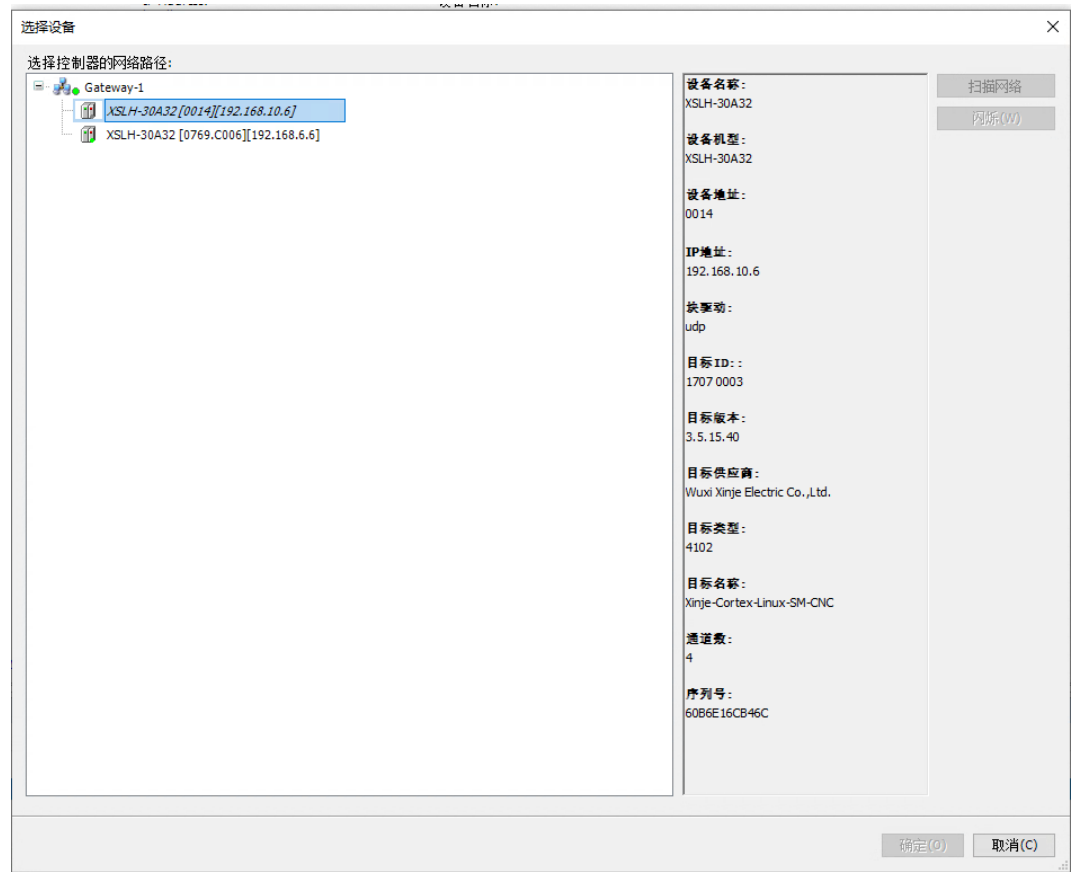
2-2-3. 设备联机

2-2-3-1. 界面介绍

双击左侧设备树内的 Device 节点，打开“通讯设置”界面：

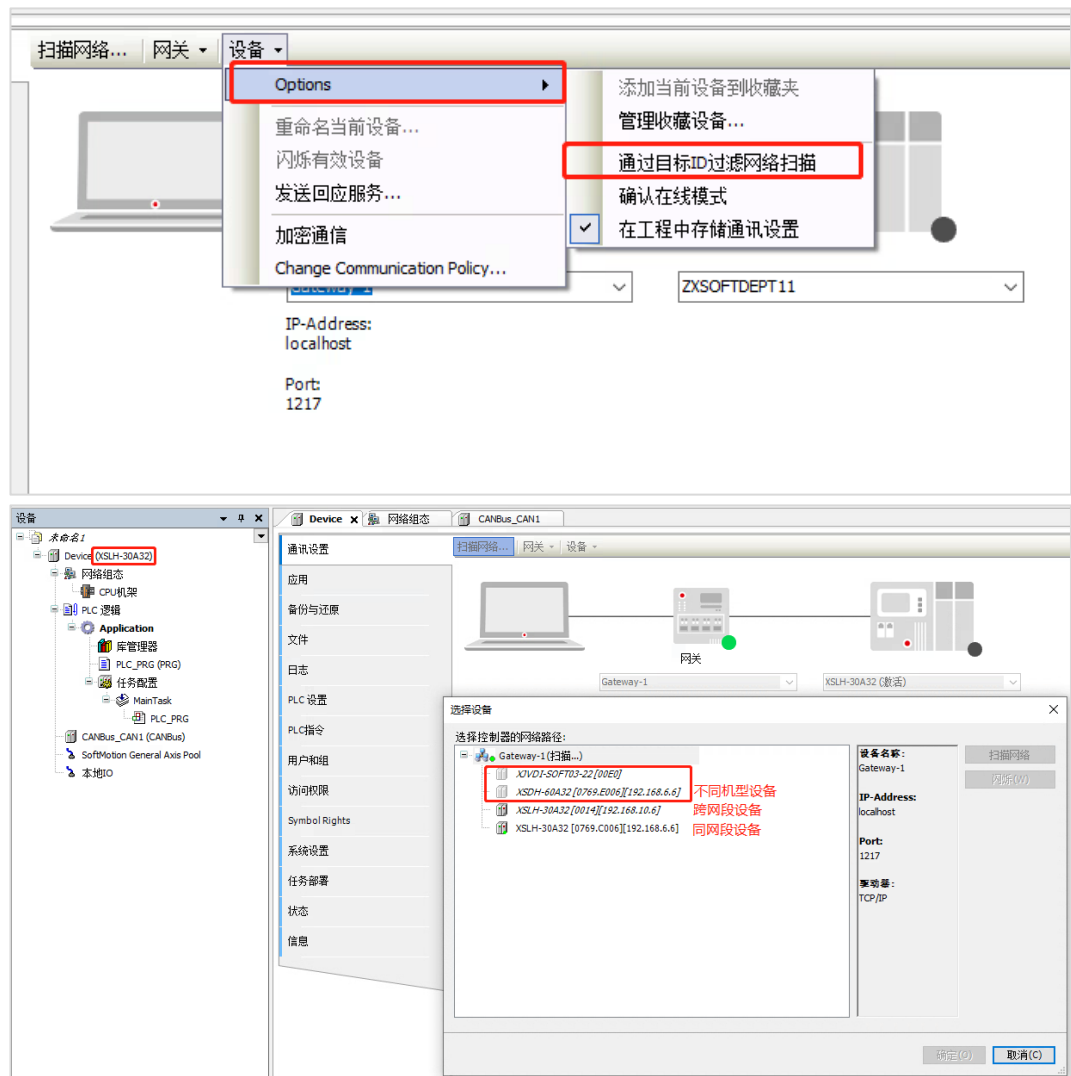


默认情况下，不修改设备连接的配置参数。单击“扫描网络”选项，打开“选择设备”界面，可启动扫描功能，将扫描到的设备显示在界面中，如图所示：

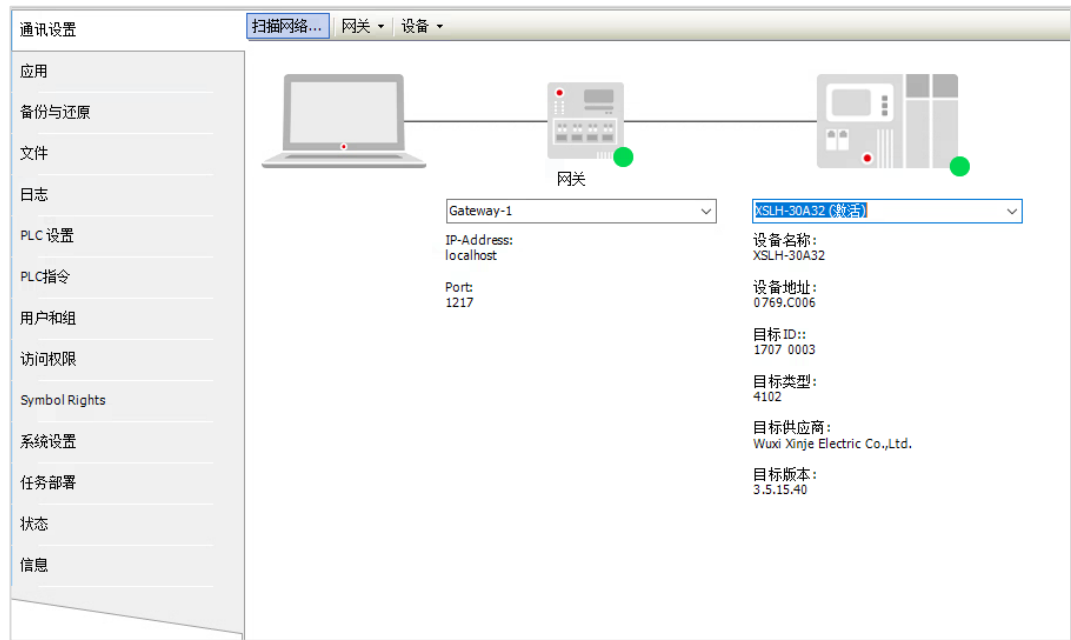


- 此时，可以扫描到绿色标识显示的同网段设备 192.168.6.6，通过选中当前设备双击连接设备，或选中设备点击确定按钮即可建立连接。
- 也可以扫描到无绿色标识且斜体字显示的跨网段设备 192.168.10.6。选中该设备，可在右侧查看相关设备信息，但不可建立连接。

扫描设备前，可在当前界面内的“设备”菜单下，打开“Options”菜单项，取消勾选“通过目标 ID 过滤网络扫描”。取消后再扫描，既可扫描同工程机型设备，也可以扫描与工程不同机型的设备。如下图所示：

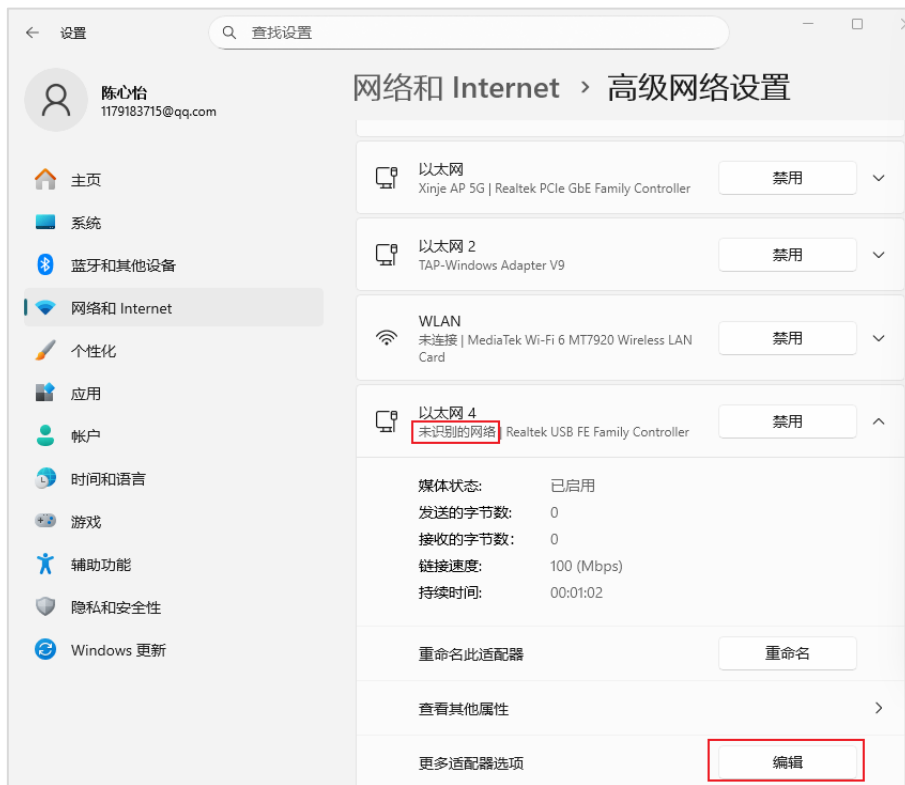


选择绿色图标显示的同网段设备，此处示例：XSLH-30A32（192.168.6.6）设备，选中该设备后双击，连接成功。如下图所示：

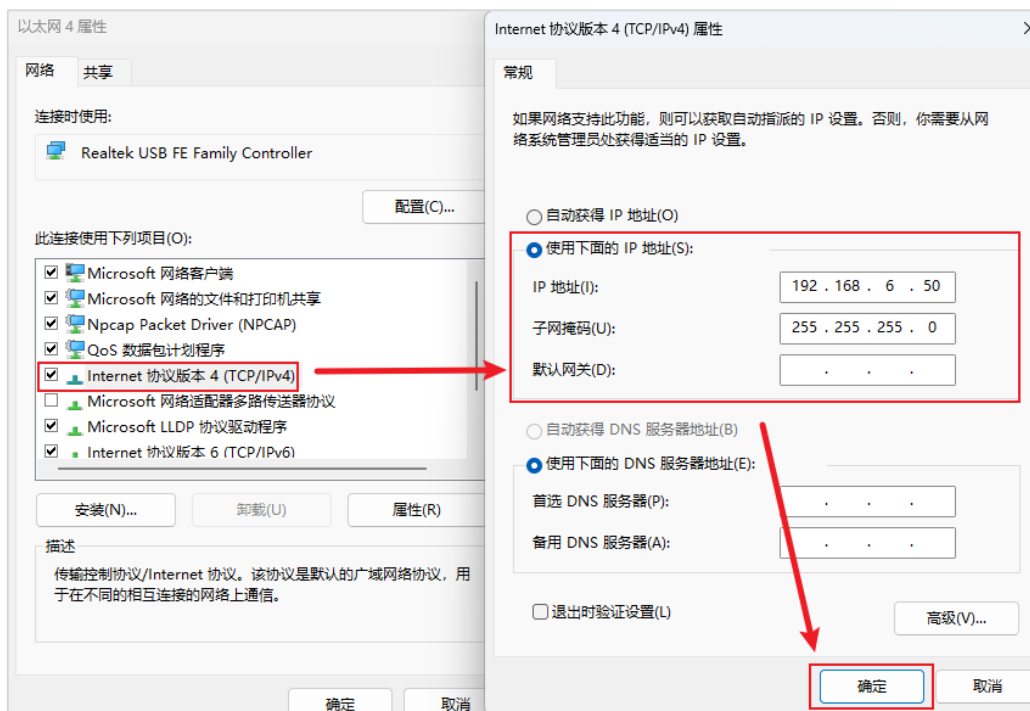


2-2-3-2. Ethernet 口联机

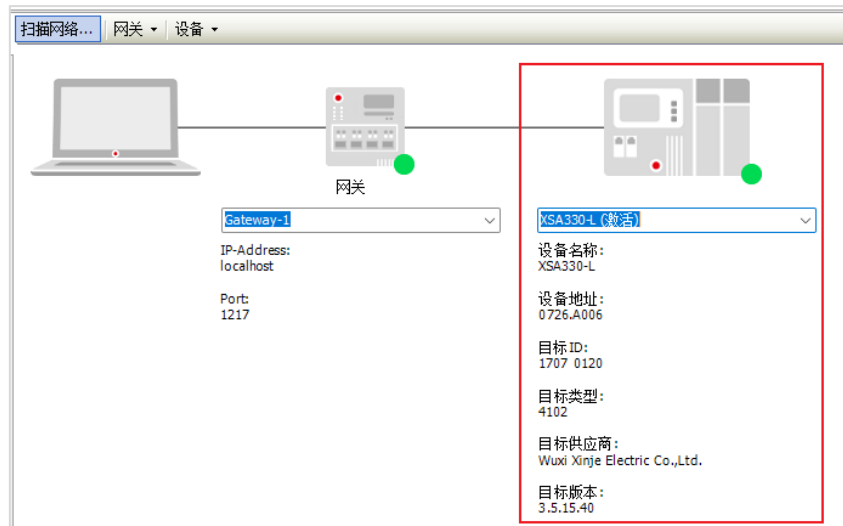
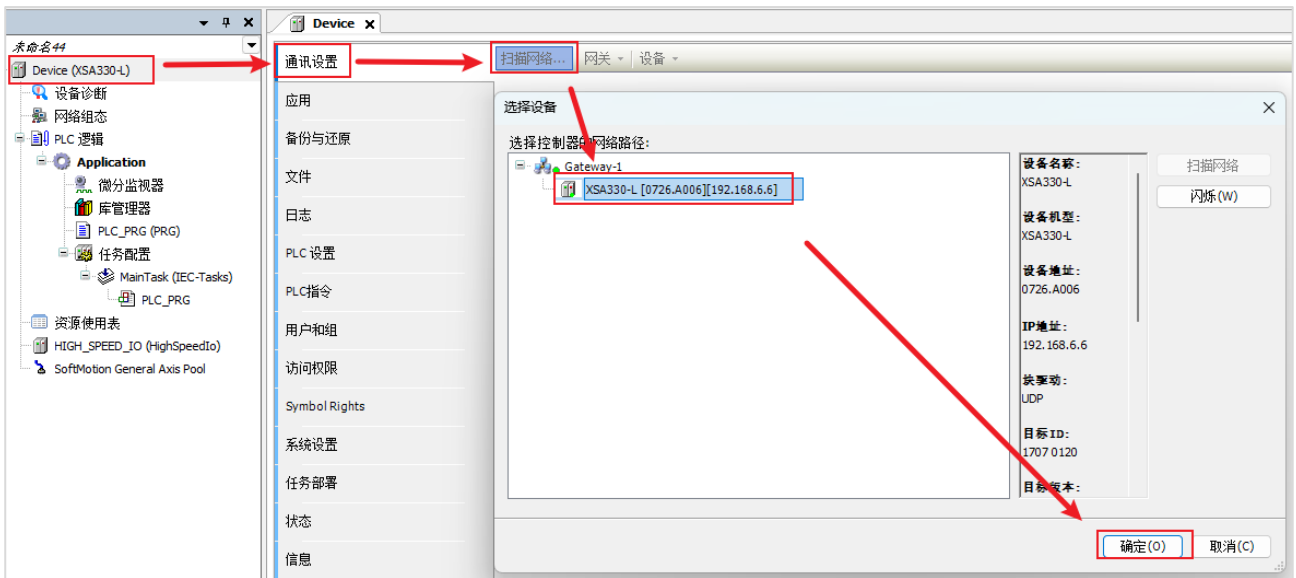
1、用网线将 XS 系列 PLC 的 Ethernet 口连接至电脑，并打开电脑的“网络适配器”页面。



2、设置电脑的 IP 地址。将电脑对应的网卡 IP 设为与 PLC 同网段的 IP。



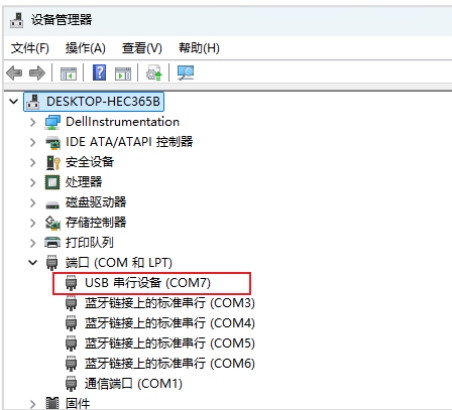
- 3、打开 XS Studio 软件，点击“Device”-“通讯设置”-“扫描网络”扫描出 PLC 后进行联机即可。



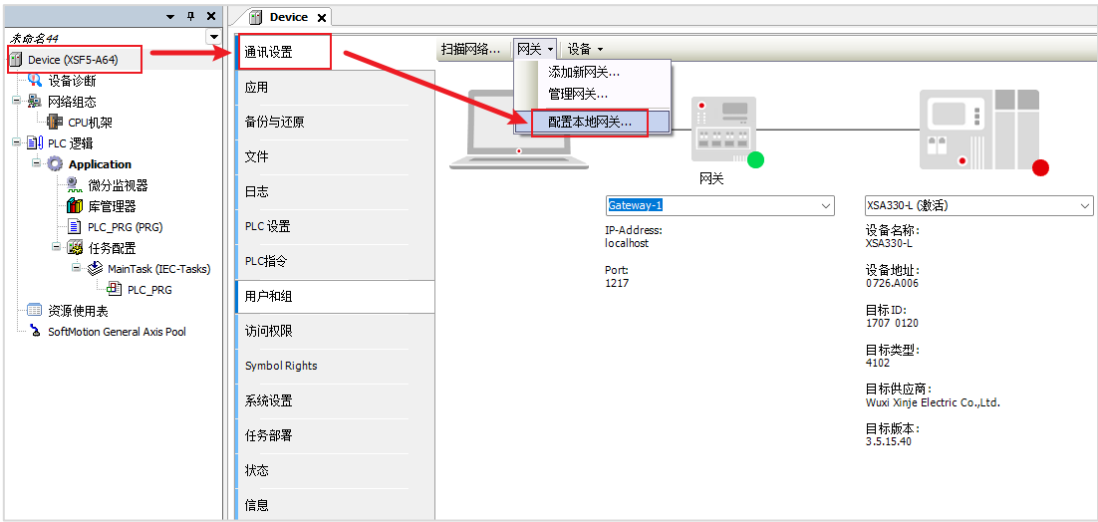
2-2-3-3. Tpye-C 口联机

- 1、用 USB 数据线将 XSF 系列 PLC 的 Type-C 口连接至电脑。

2、打开电脑的设备管理器，查看识别的 COM 端口号。当前案例中识别到的是 COM7。



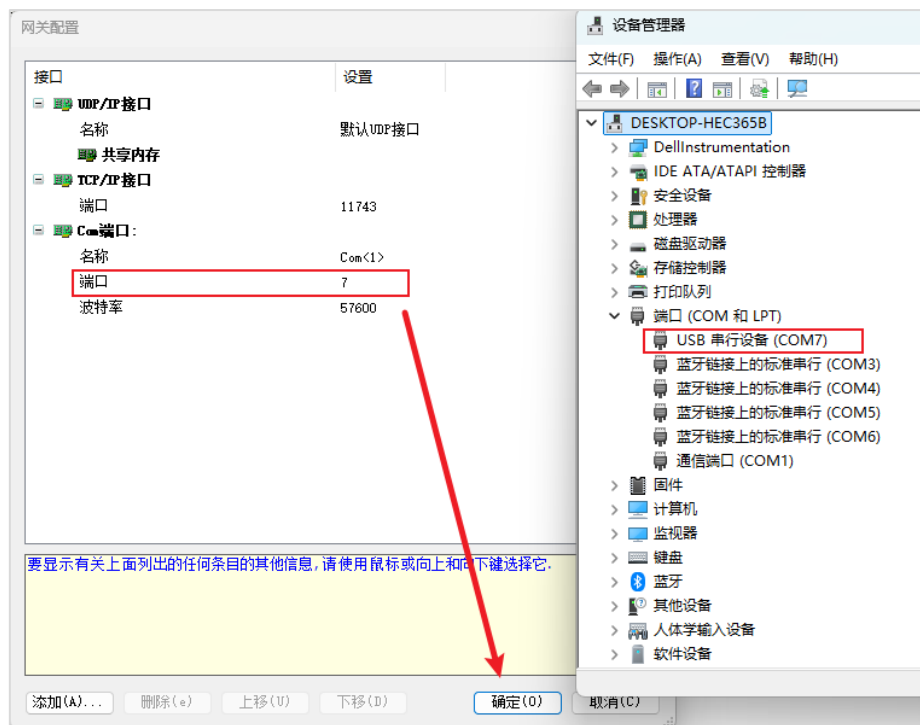
3、打开 XS Studio 软件，点击 “Device”-“通讯设置”- “网关” -“配置本地网关”。



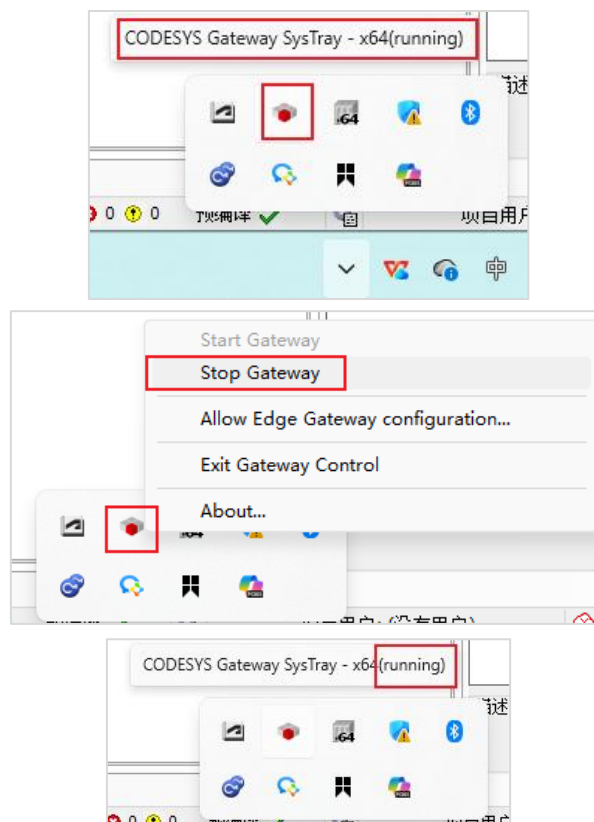
4、添加顶级接口。



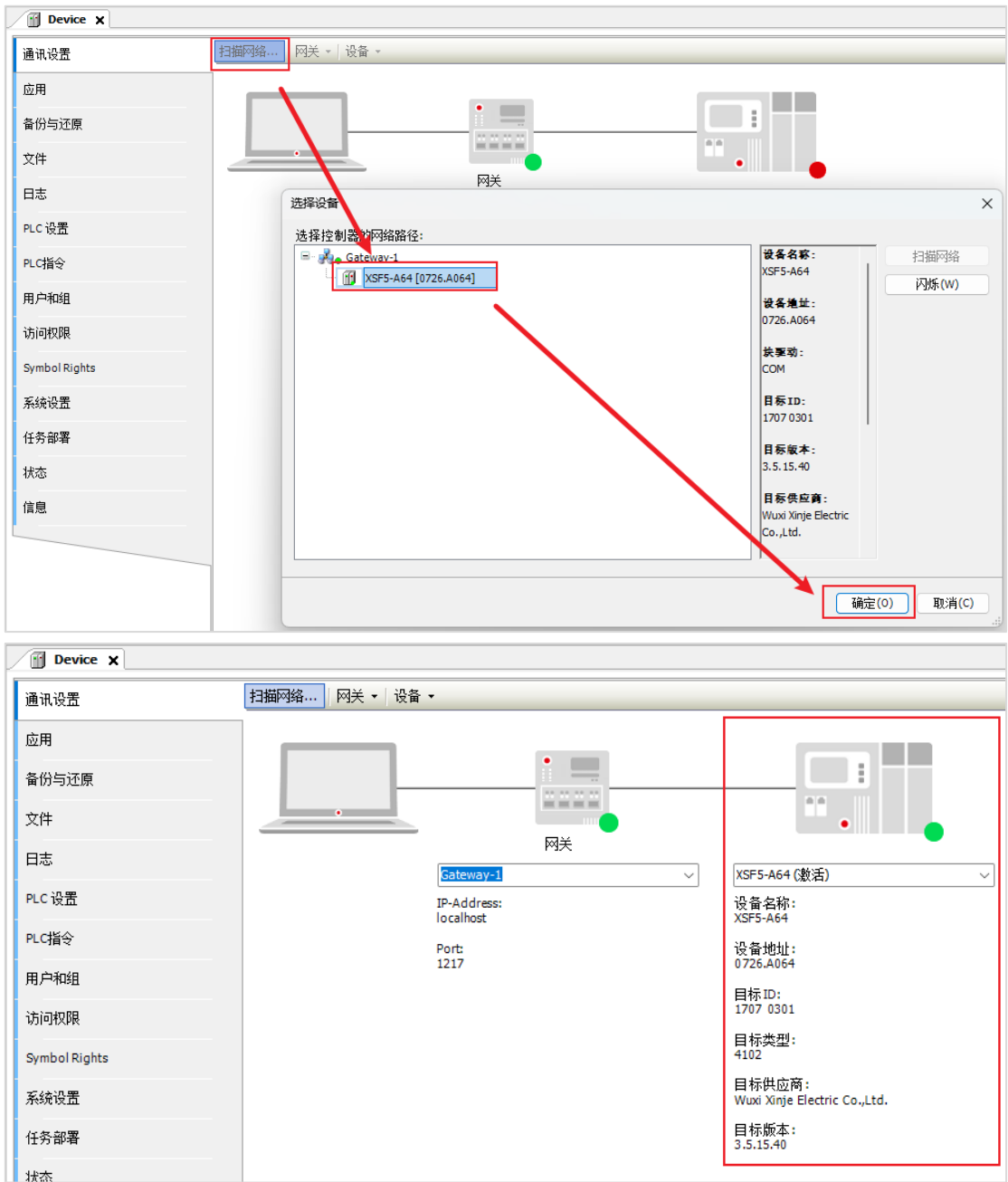
- 5、“Com 端口”的“端口”设为和电脑设备管理器识别的一样的端口号，点击“确定”。此处设置为端口 7。



- 6、在电脑右下角应用中找到“CODESYS Gateway SysTray - x64”，选中该图标并点击鼠标右键操作“Stop Gateway”，此时该图标会变灰并很快又自动运行起来，显示 running。



7、 点击“Device”-“通讯设置”-“扫描网络”扫描出 PLC 后进行联机即可。

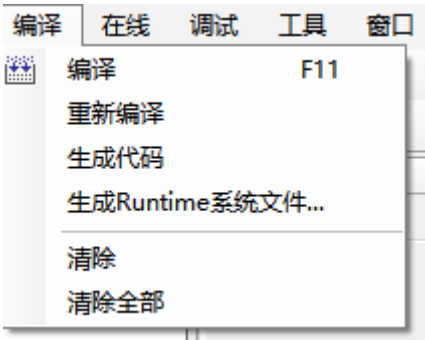


2-2-4. 程序下载/读取

2-2-4-1. 编译

程序编写完成后，在下载之前需要对程序进行编译。编译命令对编写的程序进行语法检查，并且只编译添加到任务中的程序。如果创建的 POU 没有添加到任务中，编译命令不对该 POU 进行语法检查。

编译指令不生成任何代码，只针对 POU 的语法进行检查。直接执行设备登录命令，系统也将默认执行编译指令（等同于先手动执行编译命令），在编译检查没有语法错误后执行连接登录指令。同样，在编译中不对没有添加到任务中的 POU 进行语法检查。执行登录命令同时会生成代码。



项目	说明
编译	对当前的应用进行编译。
重新编译	如果需要对已经编译过的应用再次编译，可以通过重新编译进行。
生成代码	执行此命令后生成当前应用的机器代码，执行登录命令时，生成代码默认执行。
清除	删除当前应用的编译信息，如果再次登录设备时需要重新生成编译信息。
清除全部	删除工程中所有的编译信息。

执行编译命令后可以看到，添加到任务里面的“PLC_PRG”显示为蓝色，没有添加到任务里面的则显示为灰色。编译指令不会对灰色的 POU 进行语法检查，因为该程序单元没有处于活动状态，编译指令只针对于处于活动状态的 POU 进行语法检查。如果在编译的过程中发现需要运行的程序单元显示为灰色，可以检查该程序单元是否已经被成功的添加到了所需要运行的任务当中。

编译命令执行完成之后可以在消息栏看到编译生成的信息，其中可以看到编译的程序是否有错误或者警告，以及错误和警告的数量。如果有错误和警告产生可以通过消息窗口进行查看和查找，根据提示信息对程序进行修改。



2-2-4-2. 登录下载

1、登录

登录使应用程序与目标设备建立起连接，并进入在线状态。能正确登录的前提条件是要正确配置设备的通讯设置并且应用程序必须是无编译错误的。

对于以当前活动应用登录，生成的代码必须没有错误和设备通信设置必须配置正确。登入后，系统会自动选择程序下载。

2、下载

下载命令，在在线模式下有效。它包括对当前应用程序的编译和生成目标代码两部分。除了语法检查（编译处理）外，还生成应用目标代码并装载到 PLC。



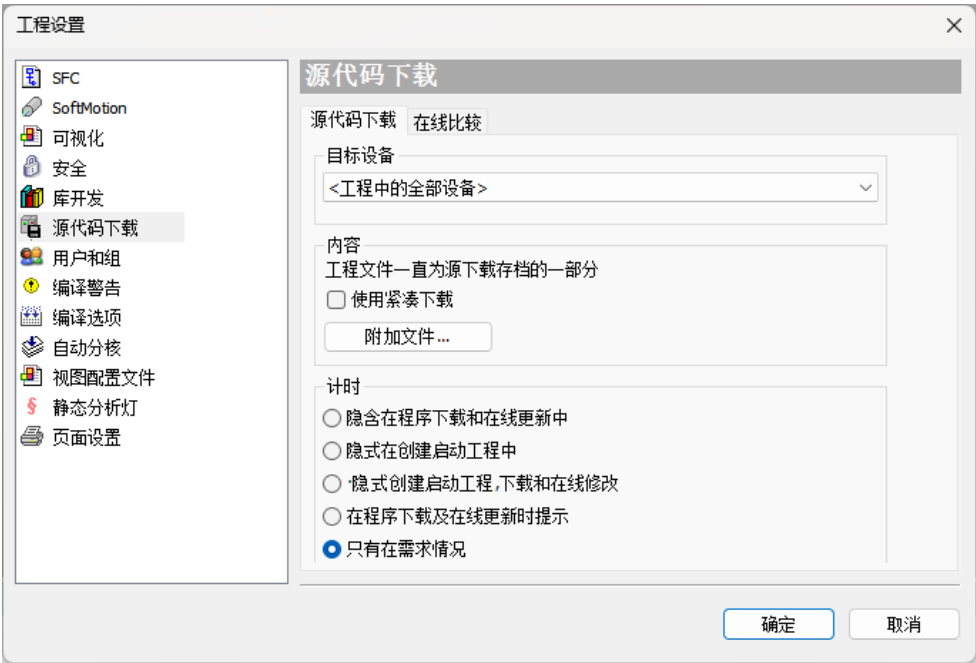
项目	说明
登录-在线修改 ^注	用户选择此选项后，项目的更改部分被装载到控制器中。使用“登录-在线修改”操作，可以防止控制器进入 STOP 状态。建议同时勾选“更新自动启动程序”，防止前面修改程序的内存导致数据程序等丢失。
登录并下载	选择“登录并下载”后，将整个项目重新装载到控制器中。与“登录-在线修改”最大的区别是当完成下载后，控制器会停留在 STOP 模式，等待用户发送 RUN 指令，或重启控制器程序才会运行。
登录-不做任何修改	登录时，不更改上次装载到控制器中的程序。



- 用户之前至少执行过一次完整的下载。
- 指针数据会更新最近一个周期的数值，如果改变了原变量的数据类型，无法确保数据的准确性，此时，需要重新分配指针数据。
- XS Studio2.3.1 及以上时，标准保持掉电模式需要勾选“更新自动启动程序”，否则可能在结构体长度变化时，效果异常。
- XS Studio2.3.2 下载时默认勾选更新自动启动程序。

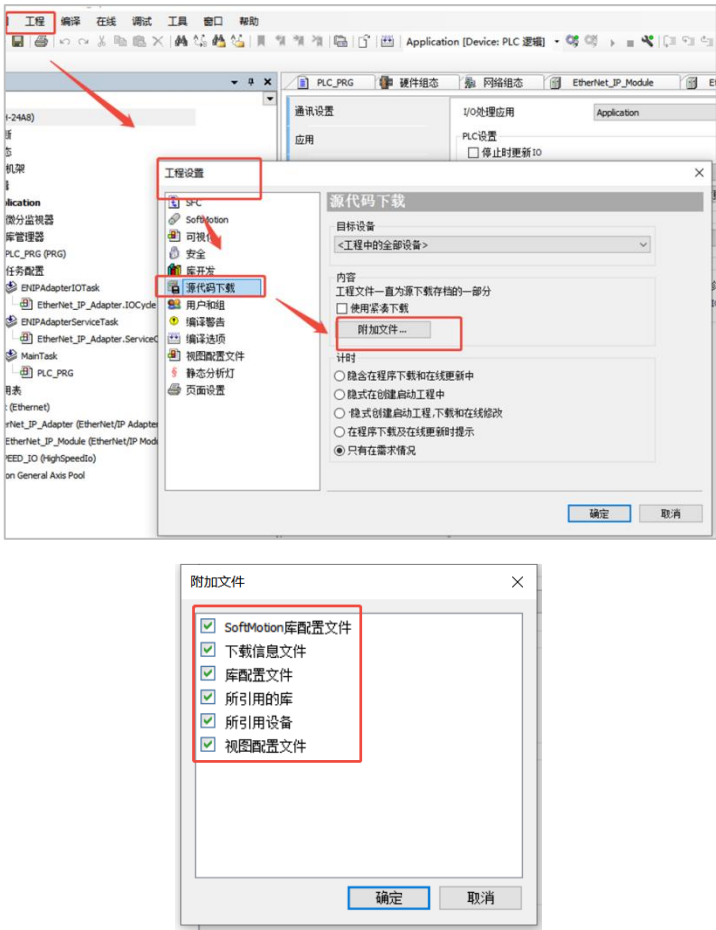
2-2-4-3. 源代码下载

出于对程序员源代码的保护，下载时默认不自动下载源代码，如需下载源代码则需要进行手动设置，点击“在线”→“下载源代码到连接设备上”。用户也可以在“工程”→“工程设置”→“源代码下载”→“计时”选项中对该属性进行设置。



注意：

若多台电脑联合编程，则需要在附件文件中配置库相关，否则其它电脑上传时没有库信息会引用软件中的最高版本，若和工程中的不一致，会导致无法在线下载或者无法直接登录调试。

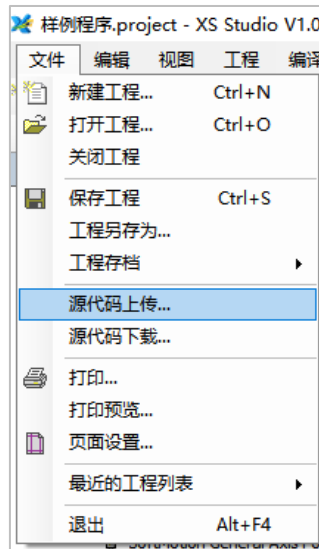


2-2-4-4. 读取程序

在“文件”>“源代码上传”命令打开一个设备选择对话框，用户选择连接 PLC 的网络路径，点

击“确定”按钮。如果用户选择的路径下已经存在存档文件，给出是否覆盖提示。

这里需要注意的是，在读取程序之前需要确保在前一次的下载过程中已经做过“下载源代码到连接设备上”。否则不能读出控制器中的数据。



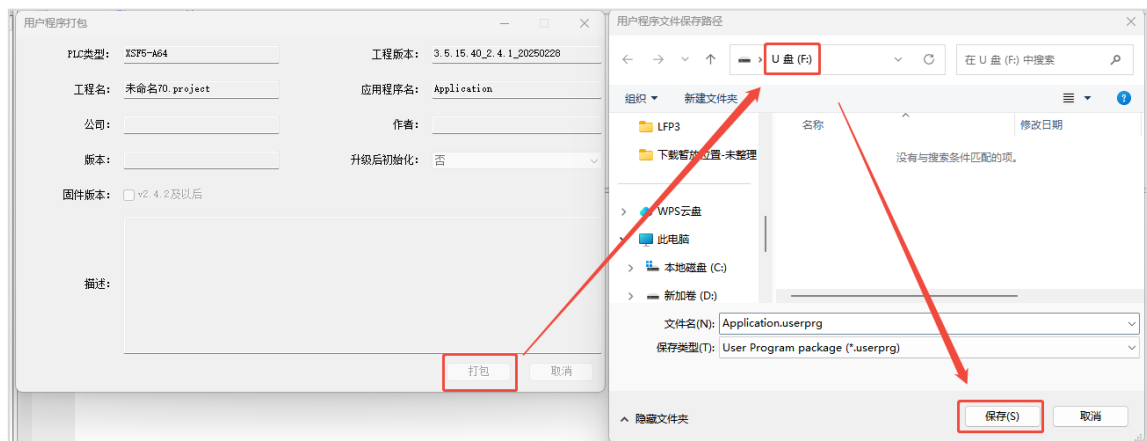
2-2-4-5. SD 卡/U 盘下载程序

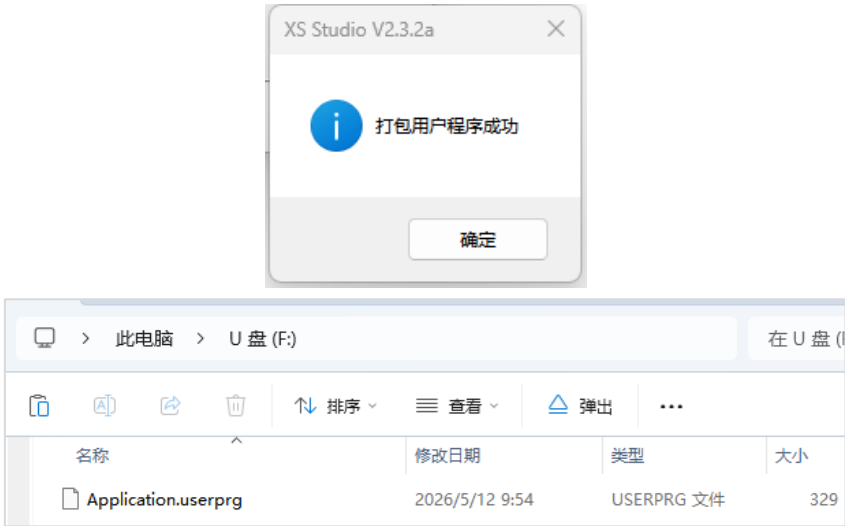
此处以 XSF5 插 SD 卡下载程序举例：

1、 确认 SD 卡格式为 FAT32 或 NTFS，将其插到电脑上；



2、 XS Studio 程序编辑好后点击“编译”确认程序无误，再点击“编译”-“打包用户程序”，将程序打包文件保存在 SD 卡里；





3、 将 XSF5 断电，SD 卡插入 XSF5 后上电，观察指示灯。等待 ERR 灯闪烁结束、RUN 灯常亮可以正常运行逻辑后就可以将 XSF5 断电并取出 SD 卡，此时已完成了程序下载的操作。



- 使用 SD 卡下载程序，SD 卡里除了本次打包的程序外，不可以有其他的打包程序文件、固件包、恢复出厂设置文件等。
- 打包的工程 PLC 机型 XML 版本不得高于需下载程序的目标 PLC 固件版本，否则会导致无法成功将 SD 卡里的用户工程下载进 PLC。

2-2-5. 程序调试

2-2-5-1. 复位

XS Studio 程序复位有如下三种方式，在“在线”菜单中进行选择。



项目	说明
热复位	热复位后，除了保持型变量（PERSISTENT 和 RETAIN 变量）或映射到 M 掉电存储区变量外，其它当前的应用的变量都被重新初始化。如果设置了初始值的变量，热复位后这些变量值还原为设定的初始值，否则变量都会被置为标准初始值 0。
冷复位	与“热复位”不同的是，冷复位命令不但将普通变量的值设置为当前活动应用程序的初始值，而且将保持型变量（RETAIN 变量）的值也设置为初始值 0。保持型变量（PERSISTENT）或映射到 M 掉电存储区变量保持不变。
初始复位	当在设备树中选择一个可编程设备时，无论是在离线还是在在线状态下都可以使用此命令。使用此命令将使设备复位到初始状态，即设备中的任何应用、引导工程和剩余变量都将被清除。 由于所有工程信息被清除，重新登入后，需要重新“下载”程序，并“启动”运行。

2-2-5-2. 断点

在 XS Studio 中“调试”菜单的视图如图所示。主要的操作涉及断点设置及单循环。



1、断点

断点是程序内处理停止的功能，当程序停止后，程序研发人员可以借此观察程序到断点位置时其变量及 I/O 等相关变量的内容，有助于深入了解程序运行的机制，发现及排除程序故障。

在 XS Studio 中所有的编程语言都可以设置断点。在文本编辑器 ST 语言中，断点设置在行上；在 FBD 和 LD 编辑器中设置在网络号上；而在 SFC 中，设置在步上。

2、单步执行

设置断点后，可以进行单步执行程序，该功能可以让程序一步一步的运行，方便编程人员进行调试，以便检查程序中的逻辑错误。

项目	说明
跳过	该命令会执行程序中当前的一条指令，执行完停止。在不调用 POU 时，跳过和跳入命令效果是一样的。但如果是调用 POU，那么跳过不会进入这个 POU，而是把这个 POU 调用当作完整的一步，一次执行完；跳入则会进入该 POU。如果用的是 SFC 语言，那么跳过会把一个动作当作完整的一步，一次执行完。如果想要实现跳到被调用的 POU 中单步调试，就必须用跳入。
跳入	执行时，当前的指令位置由一个黄色箭头表示。如果当前指令没有调用 POU，那么使用该命令和使用跳过命令的效果一样。
跳出	当正在一个 POU 中单步调试时，用跳出会把这个 POU 剩下的指令一次执行完，然后返回到这个 POU 被调用处的下一条指令。所以，如果是一层一层向下调用 POU，那么跳出就会一层一层向上返回，每次返回一层。如果程序中不包含任何 POU 调用，那么跳出无法向上层返回，就会返回到程序的开头处。

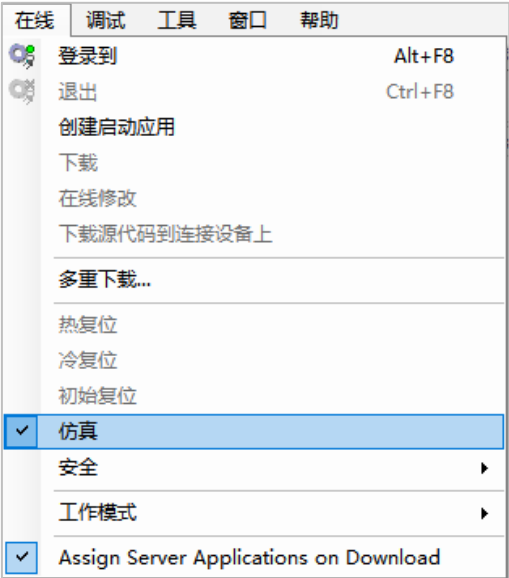
3、单循环

在“调试”—>中选择“单循环”，这样程序进行单步运行。即按一次运行，程序执行一个周期即停止并等待下一次运行指令。

2-2-6. 仿真

1、离线仿真

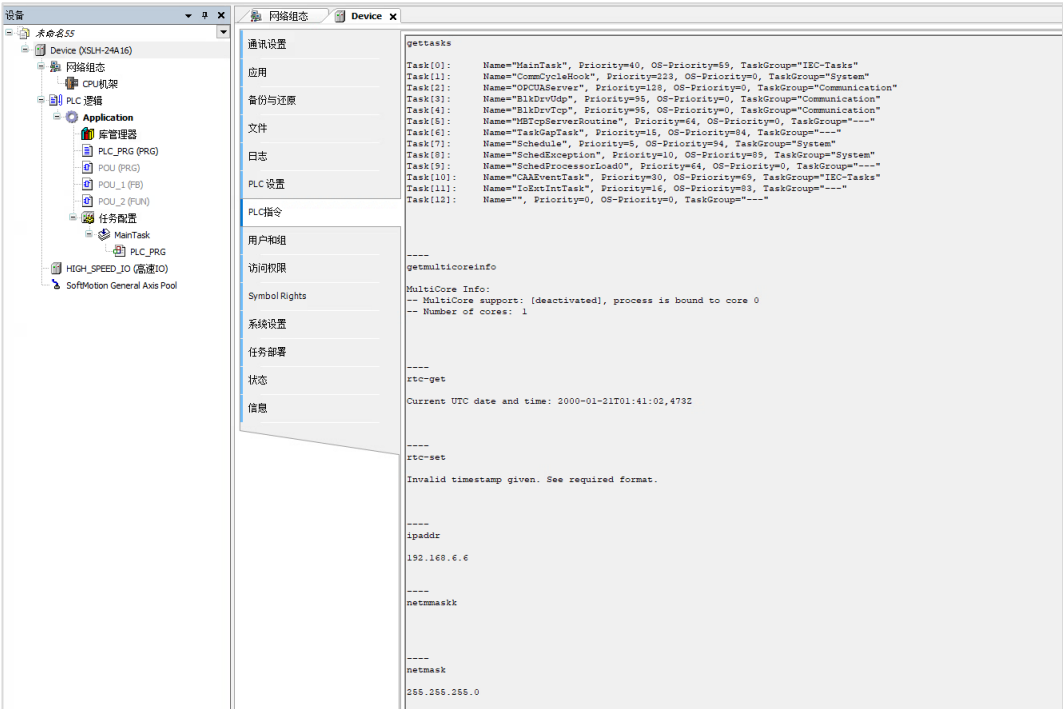
在菜单“在线”中选择“仿真”，便进入了仿真模式下的程序运行过程。确认选项“仿真”前已打上“√”后，编译程序，没有错误后，即可进入仿真模式。



2-2-7. PLC 脚本功能

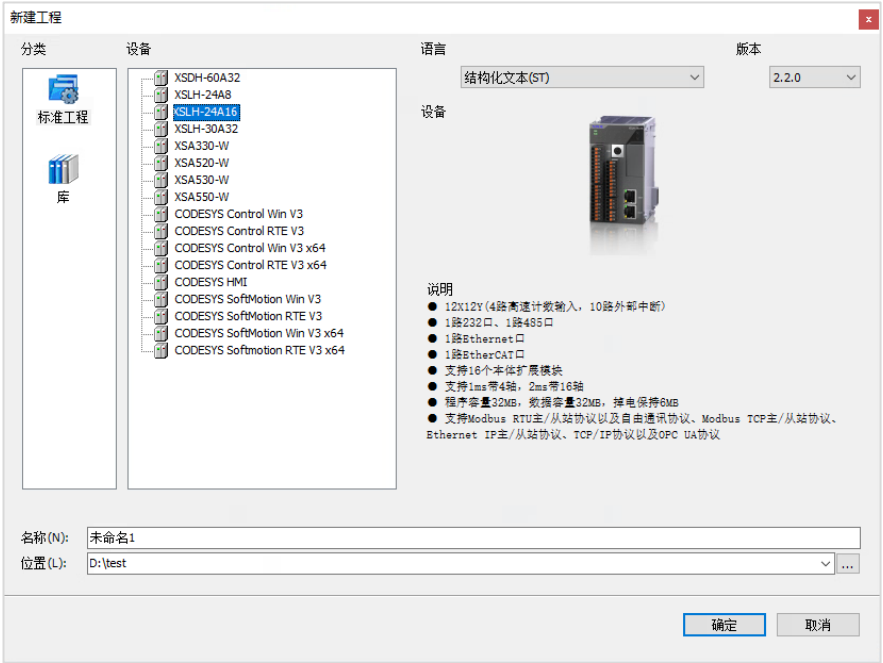
该 PLC 脚本是一个基于文本的控制监视（终端）。该功能从控制器中得到具有特定信息的命令，以一个输入行进行输入并且作为一个字符串发送到控制器，返回相关的字符串在浏览窗口中的结果显示，这个功能用于诊断和调试。

鼠标双击选中“Device”，在右边视图找到“PLC 指令”，在下方的命令输入框中输入相应指令即可。输入“？”，按回车键即可显示所有该控制器支持的命令。

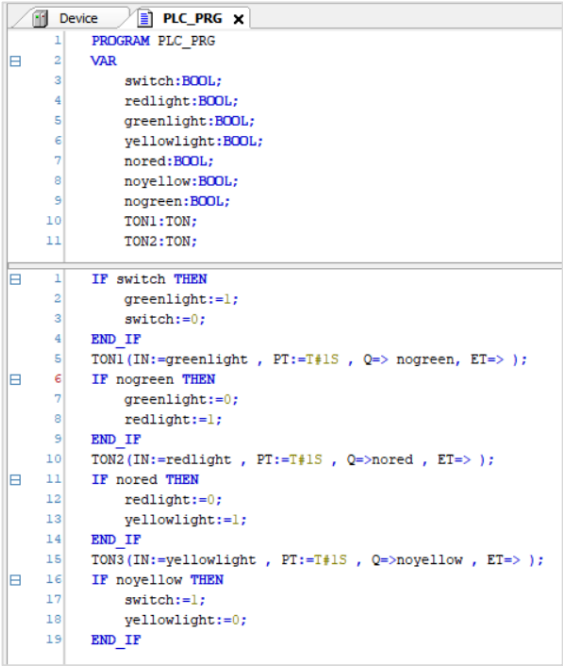
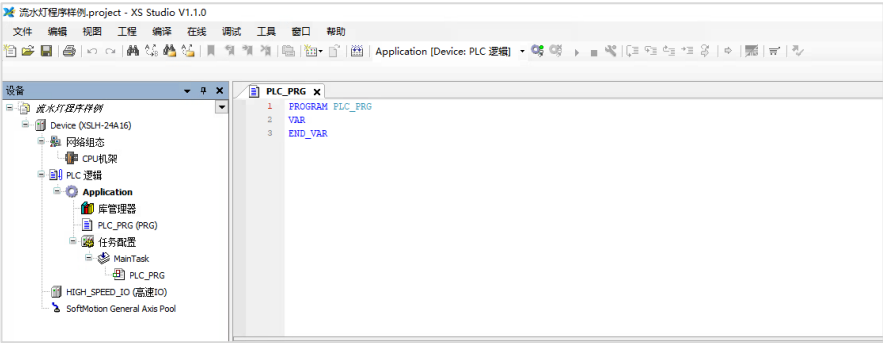


2-3. XS Studio 编写流水灯程序样例

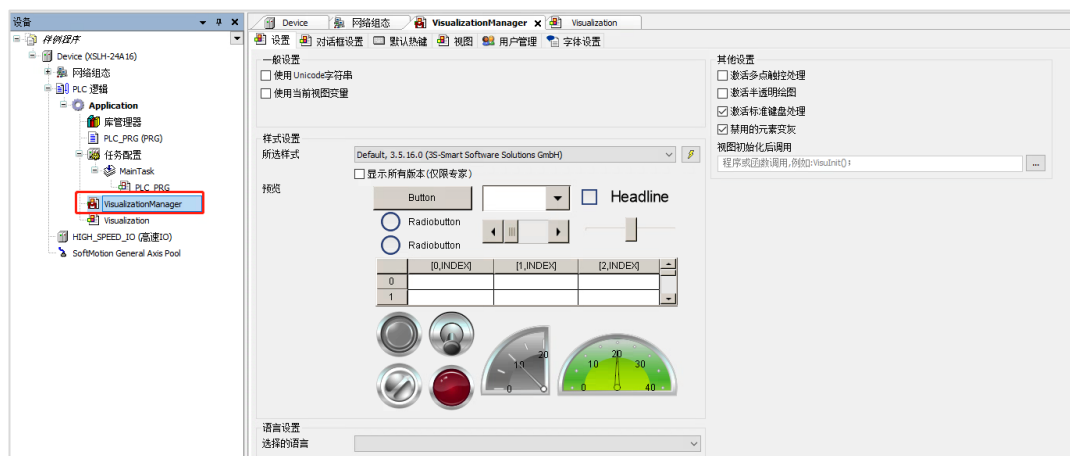
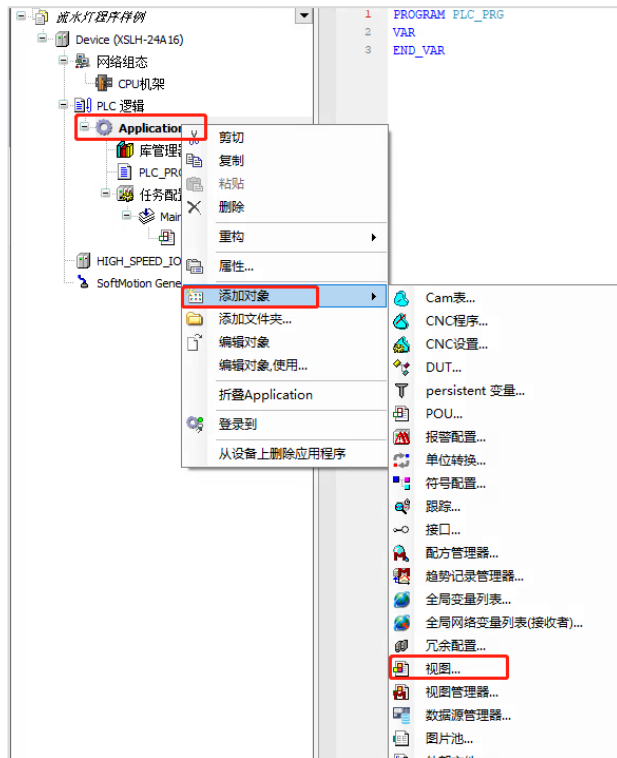
1、新建工程；



2、编写样例程序；

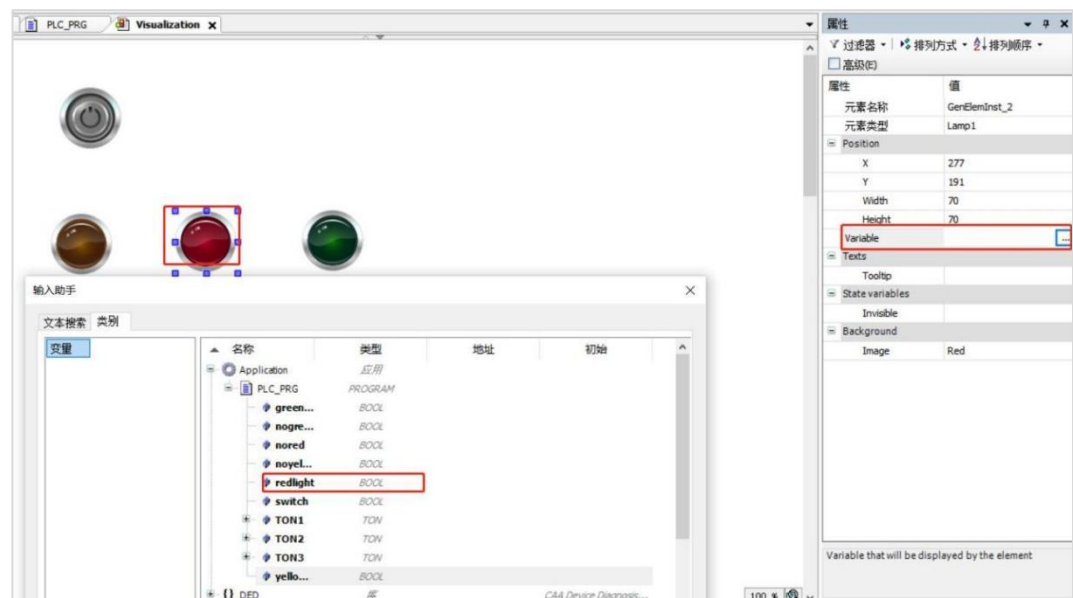


3、“application” – “添加对象” – “视图”；

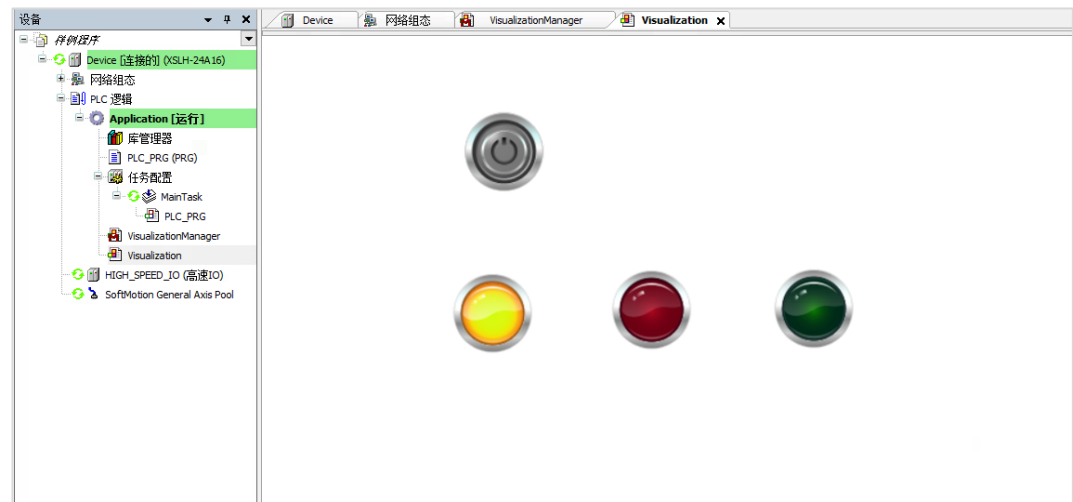


4、添加可视化对象，映射变量；





5、 登录设备，启动。



2-4. 如何登录设备

2-4-1. 登录设备的必备条件和操作简介

“登录设备”是指在 PC 上运行的 XS Studio 与 XS 系列控制器建立通讯，从而进行用户程序的下载、监控调试等操作。

PC 与 XS 系列控制器之间可以通过网线直连；也可以通过路由器、集线器连接 PLC，此时，一台 PC 可以与多台 XS 系列控制器联机，也可以多台 PC 访问同一个 XS 系列控制器。

PC 与 XS 系列控制器两者的 IP 地址必须在同一个网段且网关正常工作才能登录，否则 XS Studio 将无法扫描到 XS 系列控制器。比如，XSDH 的出厂默认 IP 地址为 192.168.6.6，若 PC 的 IP 地址为 192.168.6.xxx（这里 xxx 范围为 1~254，但不得与 XSDH 的 IP 相同），那么 XS Studio 就可以扫描到 XSDH 并与其进行连接，进行用户程序下载、监控调试等。若 XSDH 的 IP 被人为修改过，其地址与 PC 不在同一 IP 网段，二者将无法建立通讯，若客户知道其 IP 地址可以将 PC 的 IP 修改为与 XSDH 同一网段再进行连接。若用户此时不知道该设备的 IP 地址，此时需要将 XSDH 的 IP 地址恢复为出厂地址 192.168.6.6，再将 PC 本机的地址修改为 192.168.6.xxx 进行连接。

2-4-2. 扫描不到设备或者连接不上设备的解决方法

1、XS 系列

1) 不记得 IP

方法一：PLC 断电，只把拨码 1 置 ON，再重新上电恢复默认 IP 为 192.168.6.6 再联机。

方法二：XS Studio V2.2.0 开始（PLC 固件版本 V2.2.0 及以上版本），支持跨网段扫描设备功能，进行跨网段扫描来查看设备 IP。

2) 若 IP 确认无误还是无法连接设备，可能是 PLC 程序死机（程序里有死循环或超出 PLC 的负载能力），此时可将拨码 2 置 ON（上电不加载用户程序），再次扫描连接设备；若能扫描连接，此时下载一个空程序，进行抹除异常的程序之后，再恢复拨码状态，同时检查异常程序（是否有超长循环或任务周期时间过小）。

3) 修改了 IP 网段，但 PLC 的网关没有同时设置也会连不上，可以如下图直接输入 IP 地址联机，然后在 Device-PLC 指令里使用 gateway 命令修改 PLC 的网关。

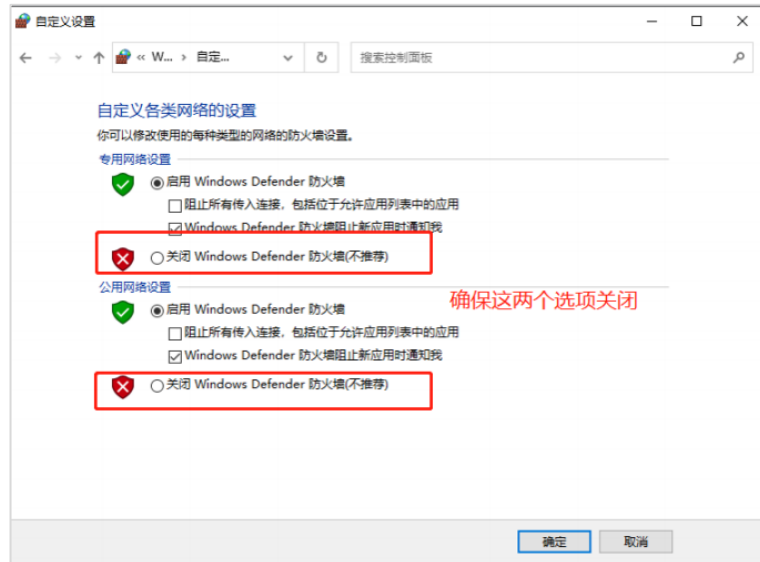


4) 如果勾选了“通过目标 ID 过滤网络扫描”，需要确认上位机工程设备型号与目标设备一致，否则会无法扫描到设备。

5) 若以上步骤还是无法连接设备, 请反馈技术支持, 并同时提供出现此问题之前做了什么操作, RUN 和 ERR 灯的状态。

2、XSA 系列

- 1) 确认上位机工程设备与目标机设备一致。
- 2) 连接显示器确认 IP、子网掩码、网关无误, 确认 PLC 与电脑双方 IP 是否为同一网段, 能否 ping 通; 若 IP 无误, 但是无法 ping 通, 则可能是防火墙的问题, 需关闭工控机防火墙之后再进行连接。



若能 ping 通, 则直接输入 IP 地址进行连接, 排除子网掩码问题。

3) 若能 ping 通且输入 IP 地址也无法连接设备, 此时有以下两种可能:

- ① PLC 程序死机, 删除 D:\CODESYS\Application 文件夹(删除用户程序), 然后重启工控机。
- ② 设备信息丢失, 可通过 RTE 配置界面查看目标 ID, 该系列产品的高 16 位 ID 为 1707。可联系技术支持进行恢复处理。





此步骤查询，需外接显示器，否则无法处理。

3、XSF 系列

1) 不记得 IP:

方法一：三段式拨码，拨到 RESET 保持 11s-20s，立刻恢复默认 IP 并释放对应连接（ENET0 192.168.6.6、ENET1：自动获取 IP 地址）；

方法二：使用 TypeC 口连 XS 软件。

2) 若 IP 确认无误还是无法连接设备，可能是 PLC 程序死机（程序里有死循环或超出 PLC 的负载能力），此时可将三段式拨码拨到 STOP（上电不加载用户程序），再次扫描连接设备；若能扫描连接，此时下载一个空程序，进行抹除异常的程序之后，再恢复拨码状态，同时检查异常程序（是否有超长循环或任务周期时间过小）。

3) 修改了 IP 网段，但 PLC 的网关没有同时设置也会连不上，可以如下图直接输入 IP 地址联机，然后在 Device-PLC 指令里使用 gateway 命令修改 PLC 的网关。



4) 如果勾选了“通过目标 ID 过滤网络扫描”，需要确认上位机工程设备型号与目标设备一致，否则会无法扫描到设备。

5) 若以上步骤还是无法连接设备，请反馈技术支持，并同时提供出现此问题之前做了什么操作，RUN 和 ERR 灯的状态。

3. 网络配置

3-1. 设备组态

组态是用户进行 PLC 编程的第一步，可通过“网络组态”和“硬件组态”界面来添加 PLC 可支持的功能。

网络组态：是组态设备的入口，可通过使能窗口及右侧网络设备连接列表对主从站设备进行布局，并以总线型的网络拓扑视角在界面内显示。

硬件组态：可添加 PLC 的右扩模块（XSA 机型不支持右扩模块）。

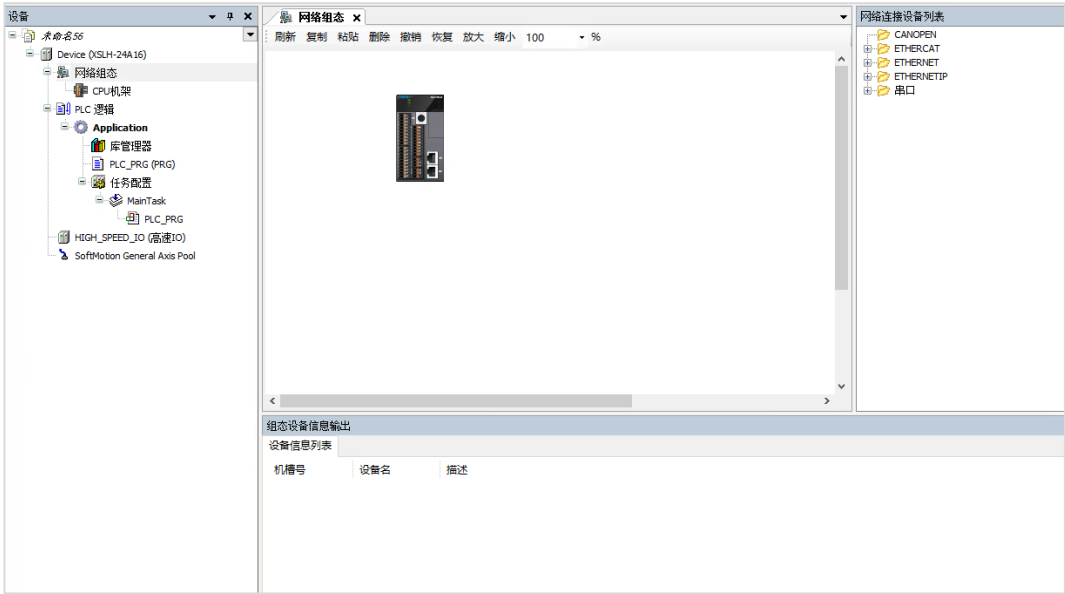
3-1-1. 网络组态

1、打开组态界面

新建 XS Studio 工程后，可通过双击软件左侧设备树内的“网络组态”节点打开组态界面。



双击“网络组态”节点后可打开网络组态界面、右侧网络连接设备列表以及组态设备信息输出界面。网络组态界面显示用户工程当前所使用的 PLC 设备，而网络连接设备列表显示当前 PLC 所支持的所有设备，设备信息输出界面则显示当前组态界面设备的名称及相关描述信息。

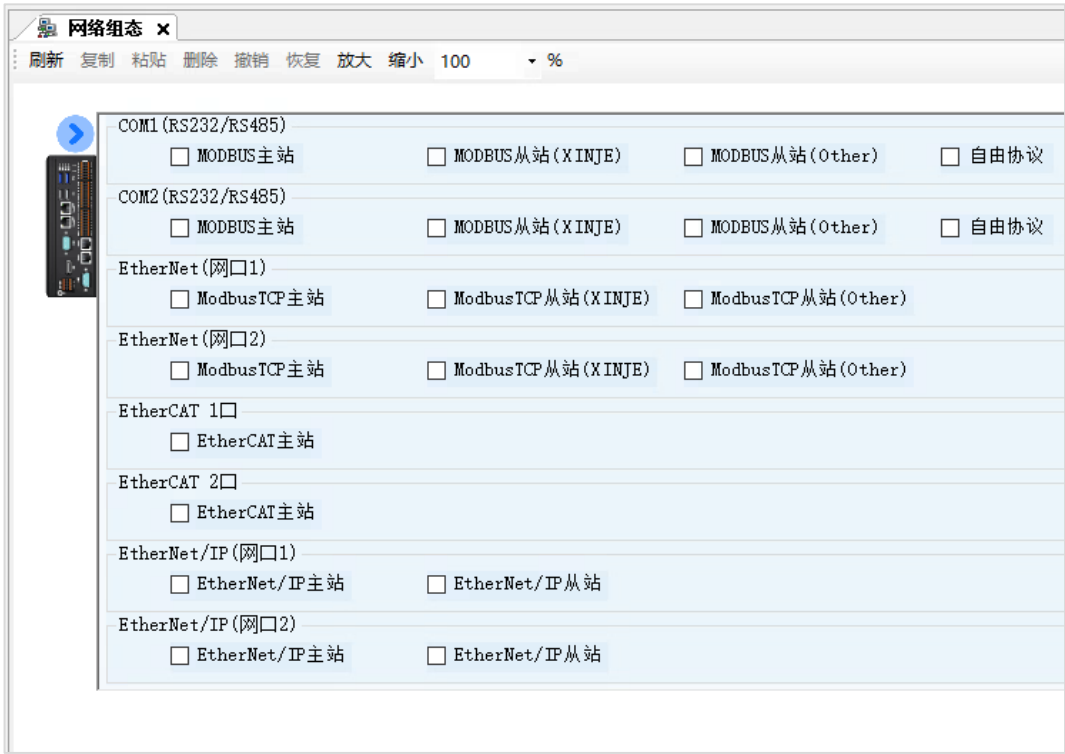


- 网络连接设备列表默认为折叠状态;
- 默认情况下, 设备信息列表为空。选中网络组态界面内的设备时会动态显示当前选中设备的相关信息描述。

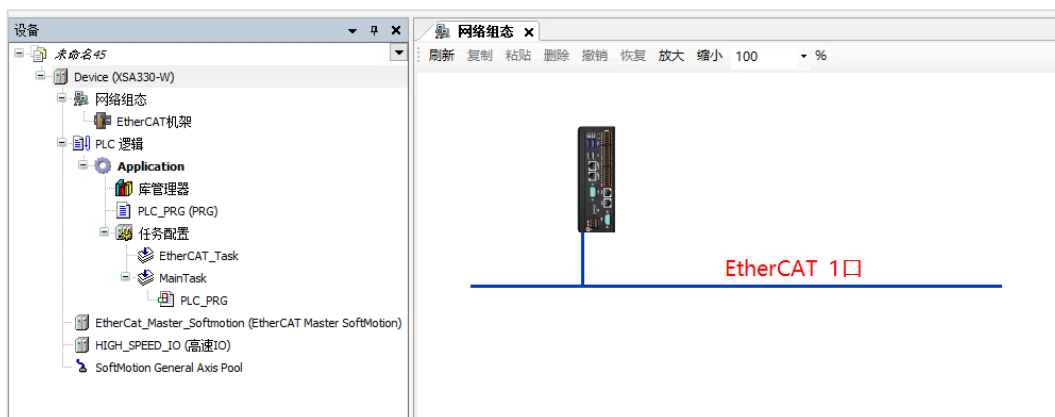
2、设置 PLC 作为主站或从站设备

1) 使能主站

单击网络组态中的 PLC 本体设备, 会显示 PLC 支持的主/从站使能窗口。如下图所示, 根据应用需要选中窗口中的复选框按钮即可使能 CPU 所支持的主/从站功能。以 XSA330-W 机型为例:



使能 CPU 的主站功能时, 都会显示总线型的拓扑界面, 并在左侧生成对应的设备节点。下图所示为使能 EtherCAT 主站:



● 失能设备

在原来勾选的基础上再次点击，则会有弹框询问是否确定移除当前设备，用户可选择确定或取消当前移除操作。

● 互斥规则

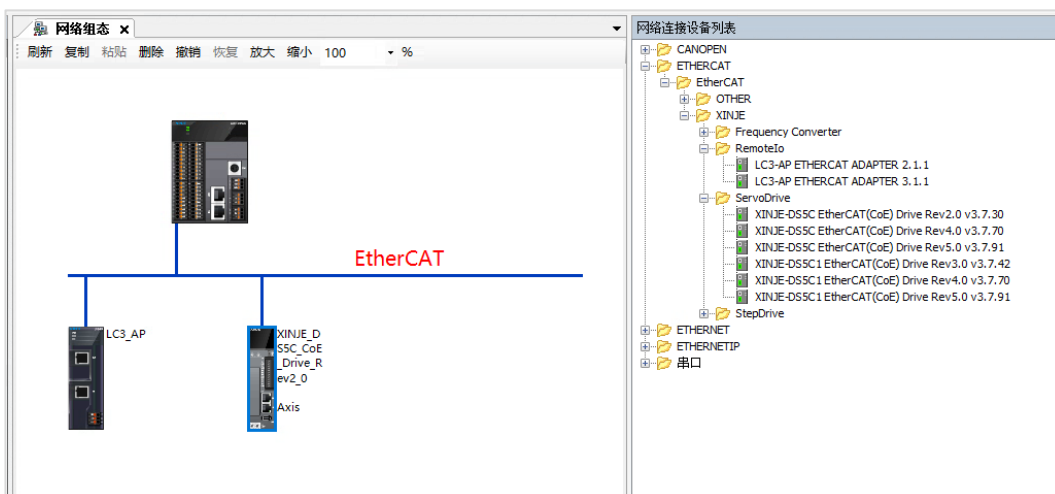
- ◆ COM 口：在对已有硬件接口上面做协议修改时，会有弹窗提示，点击确定则替换为新添加的设备，点击取消则取消当前操作；
- ◆ EtherNet: ModbusTCP 信捷从站与官方从站互斥，同时点击勾选会有弹框询问；
- ◆ EtherCAT：无互斥；
- ◆ CANopen：无互斥；
- ◆ EtherNet/IP：EtherNet/IP 主站/从站，可以同时使用，无互斥。

2) 添加从站

使能 CPU 中的某个特定的主站后，即可添加其相对应的总线下的从站设备，添加从站设备有三种方式（以 EtherCAT 总线为例）：

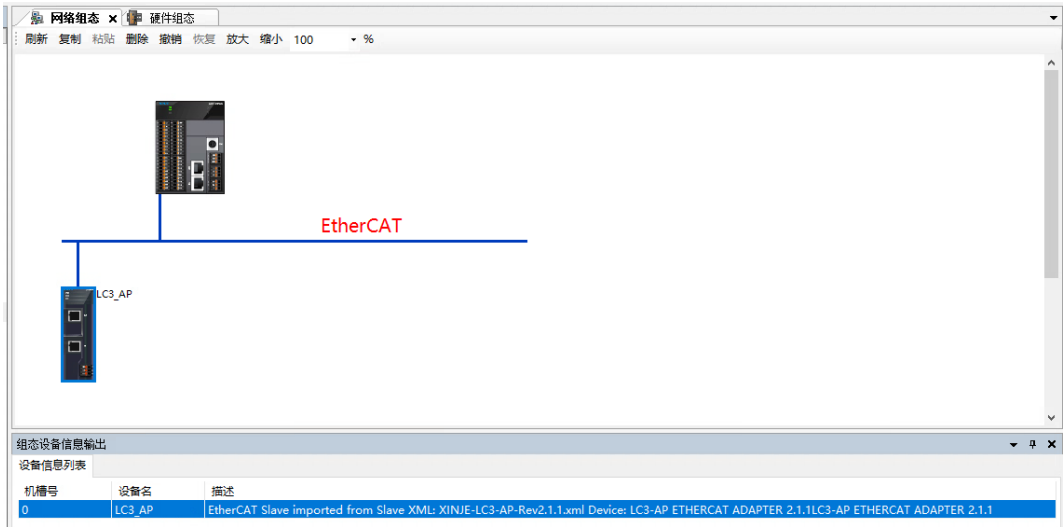
- ① 先使能 EtherCAT 主站功能，然后从网络连接设备列表中的“EtherCAT 口”节点下选中一个从站设备节点，按住鼠标左键不松拖拽到网络组态界面中。
- ② 先使能 EtherCAT 主站功能，然后网络连接设备列表中的“EtherCAT 口”节点下双击一个从站设备节点即可。
- ③ 直接在网络设备列表中的“EtherCAT 口”节点下双击一个从站设备节点添加，此种方式会默认先使能 CPU 内部的特定主站功能。

若添加的从站为 EtherCAT 远程 IO 设备，需要配置从站后的 IO 模块，可双击设备进入“EtherCAT 机架”界面进行配置。添加从站后的网络组态界面如下所示：



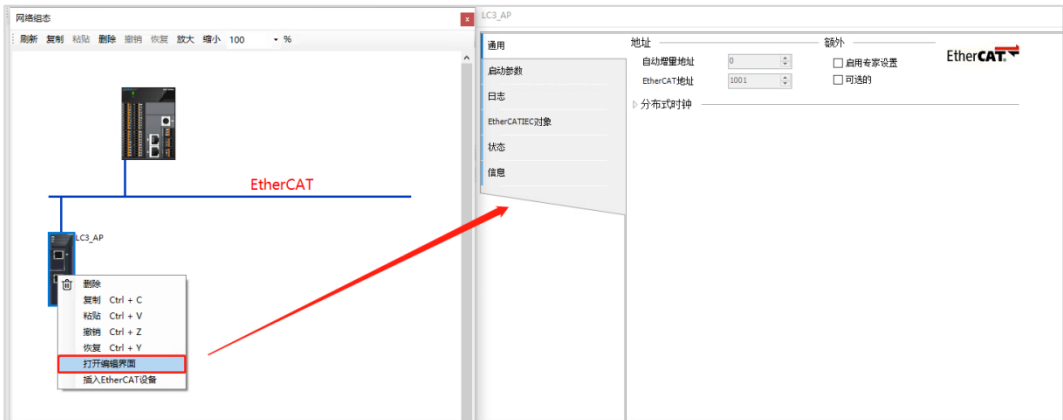
3) 查看设备基本信息

选中网络组态界面内的设备后，可在“组态设备信息输出”框内的设备信息列表内查看设备对应的基本信息。



4) 打开设备配置界面

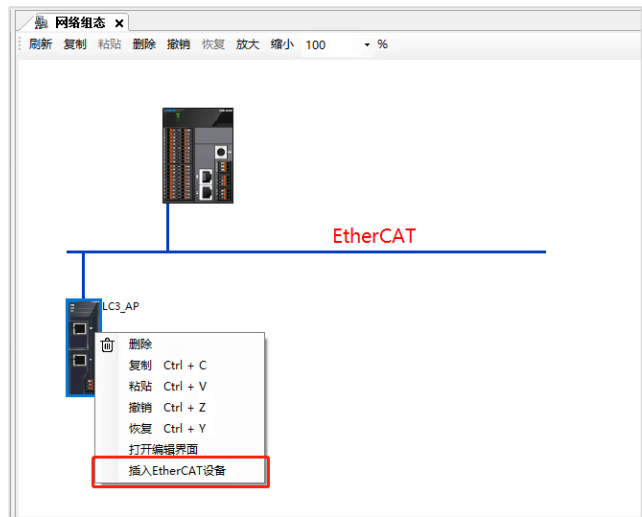
右键单击网络组态界面内的从站设备，通过“打开编辑界面”菜单项进入设备的参数配置界面。以 EtherCAT 为例，如图所示：



- 网络组态界面内双击 EtherCAT 或 CANopen 设备图标可跳转到该设备模块对应的硬件组态界面，点击其他的设备图标则跳转到模块参数配置界面；
- 双击扩展模块或从站后的 IO 模块，可打开模块配置界面。

5) 插入设备

右键单击网络组态内的从站设备，可通过“插入 XX 设备”菜单项打开插入设备弹框添加从站设备。以插入 EtherCAT 设备为例，如下图所示：



组态设备可进行复制、粘贴、删除等操作，具体情况请参见组态的基础操作说明。

6) 设备信息列表

设备信息列表通过软件菜单栏“视图”下的“组态设备信息列表视图”打开。设备信息列表显示组态设备的基本信息，主要包括机槽号、设备名及对应的信息描述。若设备信息列表处于隐藏状态，则需要手动点击  组态设备信息输出 打开列表界面。

组态设备信息输出		
设备信息列表		
机槽号	设备名	描述
0	LC3_AP	EtherCAT Slave imported from Slave XML: XINJE-LC3-AP-Rev2.1.1.xml Device: LC3-AP ETHERCAT ADAPTER 2.1,1LC3-AP ETHERCAT ADAPTER 2.1,1
1	XL_E16X_V3	扩展16点直流输入,供电电源DC24V,NPN输入
2	XL_E32X_V3	扩展32点直流输入,供电电源DC24V,NPN 输入
3	XL_E32X_V4	扩展32点直流输入,供电电源DC24V,NPN 输入
4	XL_E16Y_V2	扩展16点继电器输出,此模块无需供电电源
5	XL_E16Y_V3	扩展16点继电器输出,此模块无需供电电源

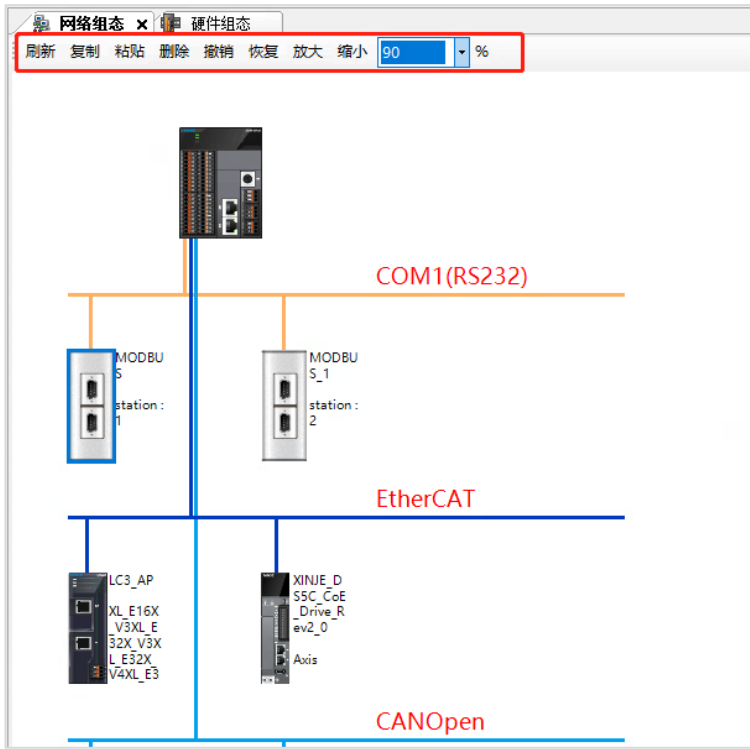
项目	说明
机槽号	对应硬件组态中的设备插槽，无论是主机架 CPU 上的模块还是通讯从站后的模块，其槽号都是从 1 开始。其中，在硬件组态界面内的通讯从站本体槽号默认为 0。主机架 CPU 内第一个 1 号机槽对应左扩展模块，第二个 1 号机槽对应中扩展模块，第三个 1 号机槽对应第一个右扩展模块。如下方图片所示。
设备名	与软件左侧设备树中所示的设备名称一致。
描述	设备的基本，包括设备基本工作指标及功能。



- 组态基础操作说明
- 组态设备基本操作包括设备的刷新、复制、粘贴、删除、撤销、恢复、放大、缩小功能：



项目	说明
刷新	点击刷新按钮，硬件组态界面或 CPU 机架界面内两个右扩模块之间有空槽的，刷新之后可将空槽右边的模块左移将空槽替换。 新增 XML、EDS、DCF 等描述文件的设备，刷新之后在网络连接设备列表中更新。
复制	点击新添加的设备后，鼠标点击对应设备图标，设备高亮显示，点击“复制”即可对相应设备进行复制。未点击对应设备图标时，复制按钮不可用。支持快捷键 Ctrl + C 进行复制操作。
粘贴	对所添加的设备点击复制后可进行粘贴。未复制的情况下，粘贴按钮不可用。支持快捷键 Ctrl + V 进行粘贴操作。
删除	选中设备后可点击“删除”来删除相应的设备。未选中状态下，删除按钮不可用。支持快捷键 Del 进行删除操作。
撤销	可撤销针对界面组态上一步的操作步骤，可连续撤销多次。支持快捷键 Ctrl + Z 进行撤销操作。
恢复	针对已撤销的内容，点击“恢复”后可恢复到上一次撤销前的界面。无撤销操作时，恢复按钮不可用。支持快捷键 Ctrl + Y 进行恢复操作。
放大/缩小	可通过放大/缩小下拉框设置界面缩放比例，也支持快捷键 Ctrl+鼠标放大或缩小当前界面。如下方图片所示。



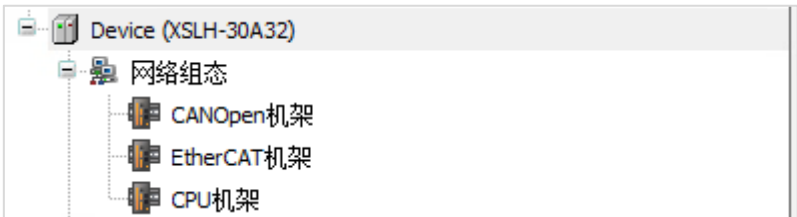
- 设备复制、粘贴、删除、撤销、恢复操作在硬件组态 EtherCAT 机架与 CPU 机架界面内只针对 IO 模块的操作，CANOpen 机架界面内禁用，而网络组态界面仅针对从站设备；
- 如果在网络组态界面中对从站设备进行复制、粘贴、删除，则后面的模块也会被相应操作；
- 导入设备文件：支持通过软件菜单栏的“工具”、“设备存储库”菜单项导入需要的设备文件，可导入 XML、EDS、DCF 等类型的设备描述文件。

3-1-2. 硬件组态

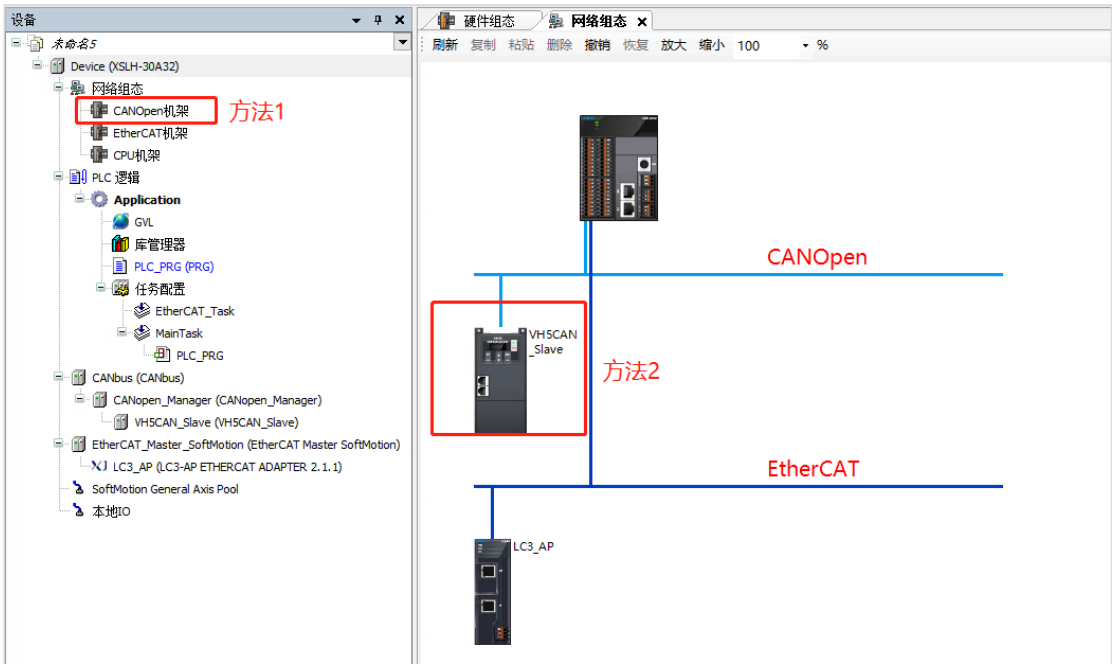
硬件组态引入了实际设备组态中的机架和机槽概念，用来模拟现场设备的模块化配置。硬件组态主要面向 PLC 系列产品的 IO 模块。
从组态流程来讲，如果是添加远程 IO 模块，应该在网络组态中完成通讯模块组态之后，再在硬件组态中进行 IO 模块的配置。

1、硬件组态界面入口

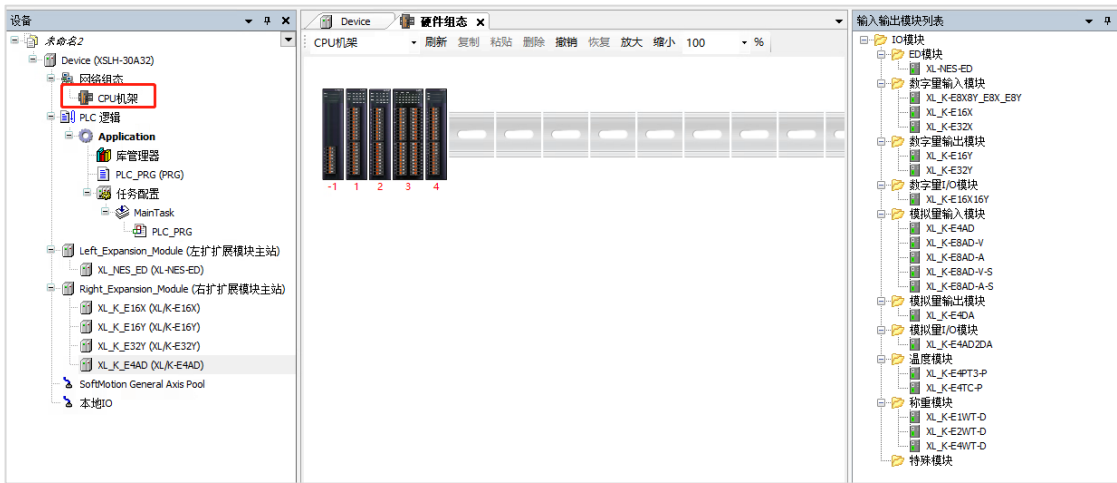
目前总线型设备 EtherCAT 与 CANOpen 有对应的硬件组态界面。



- 有两种方式可进入硬件组态界面：
- ① 双击设备树网络组态节点下的某个总线节点；
 - ② 在网络组态界面内双击一个设备。



ARM 系列控制器默认有“CPU 机架”总线配置节点（X86 系列控制器不支持），双击左侧设备树“CPU 机架”即可进入本地模块的配置界面，同时软件右侧显示“输入/输出模块列表”。以 XSLH-30A32 机型为例，具体如下图所示：

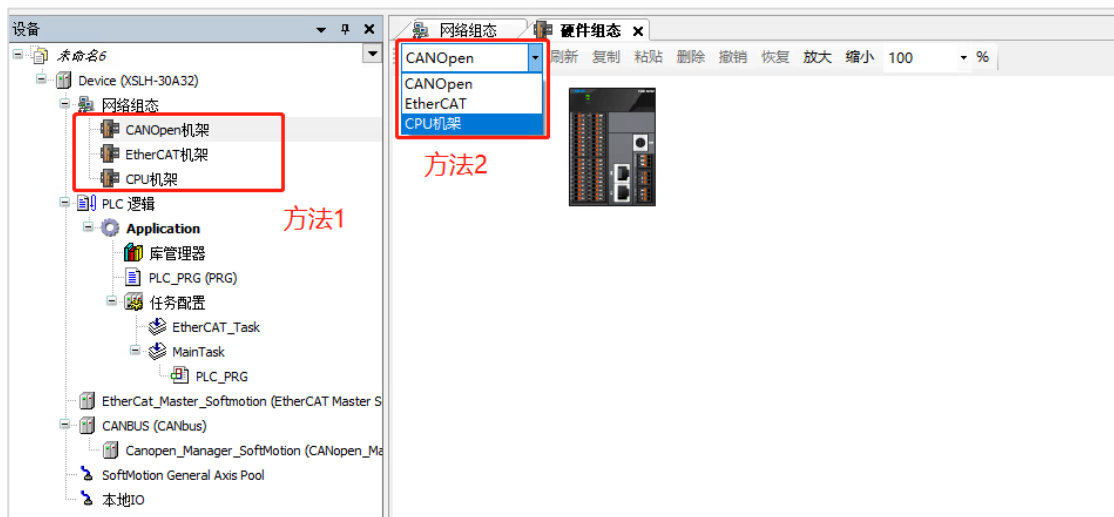


- XSDH 系列支持添加 1 个 ED 和 1 个 BD，16 个 XD 系列扩展模块；
- XSLH 系列支持添加 1 个 ED 和 16 个 XL 系列扩展模块。
- XSF 系列支持添加 32 个 XF 系列扩展模块

2、总线切换

有两种方式可实现硬件组态总线之间的切换。

- ① 双击软件左侧设备树“网络组态”节点下的某个总线节点，进入相应的配置界面；
- ② 在当前硬件组态界面选中下拉框内的其他总线类型进行切换。

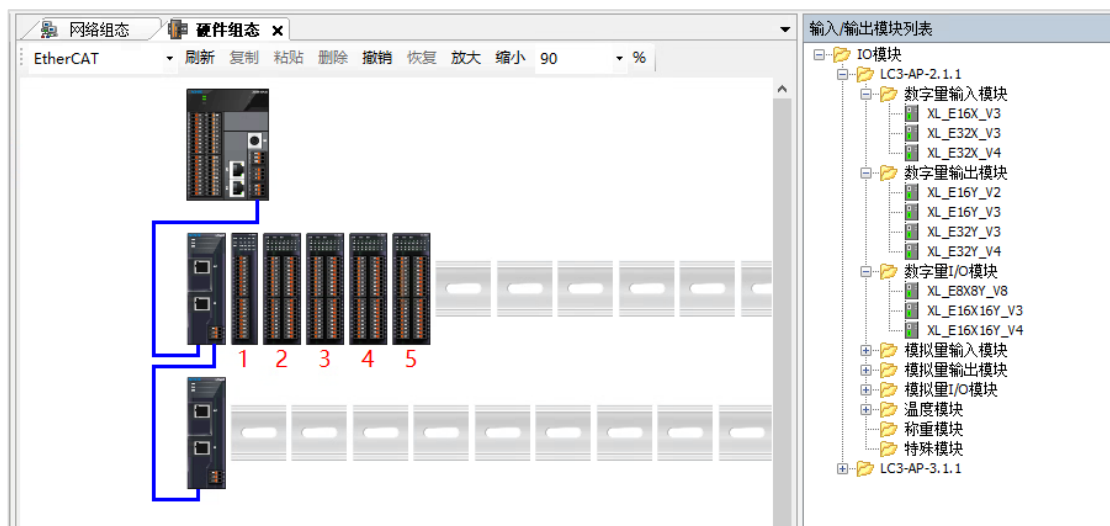


3、添加模块

有三种方式来添加 IO 模块。

- ① 双击打开机架上的一个空机槽，在弹出的“插入远程 IO 模块”中双击特定模块即可添加；
- ② 在右侧视图的“输入/输出模块列表”中，选中一个设备节点，按住鼠标左键将其拖放到空机槽上；
- ③ 从站后添加 IO 模块则需选中一个设备，双击视图右侧“输入/输出模块列表”中的某个设备，即可将设备按顺序自动添加到机架上的空槽中。而如果选中某个空槽，则会添加到该空槽中。CPU 机架界面内添加 IO 模块则不需要在添加之前先选中本体，直接双击即可。

如下图所示：



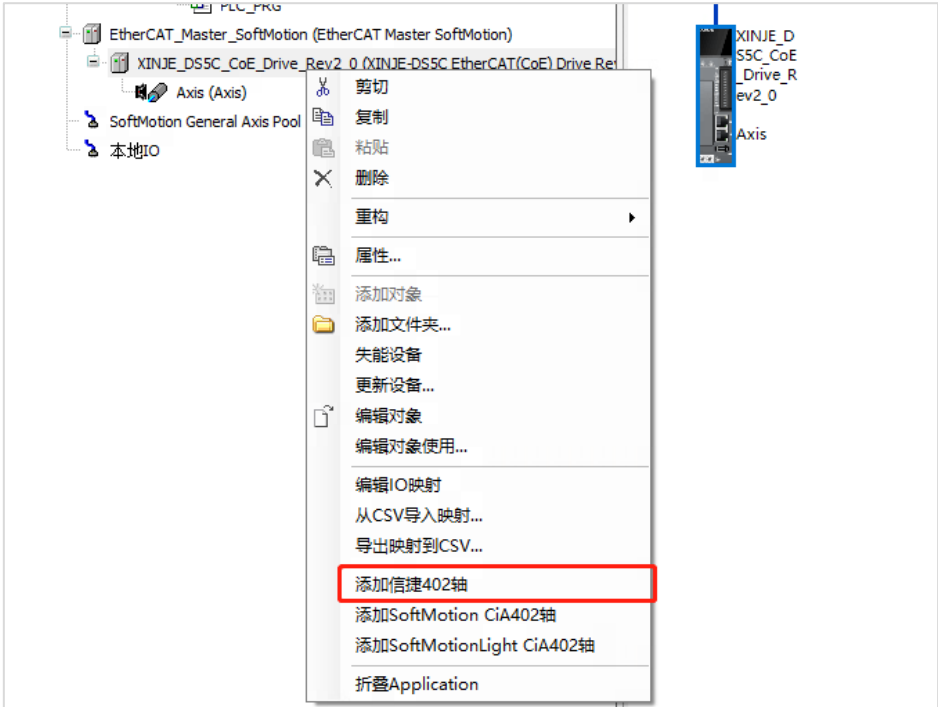
4、拖拽模块

通过选中一个模块按住鼠标左键进行模块的拖拽操作，拖拽到目标槽位置松开鼠标即可。拖拽操作包括两个模块之间的位置交换，或将一个模块拖动到空槽中，但不支持两个扩展机架之间模块的拖拽操作。

3-1-3. 设备树操作

1、信捷 402 轴

EtherCAT 伺服支持右键添加“信捷 402 轴”；

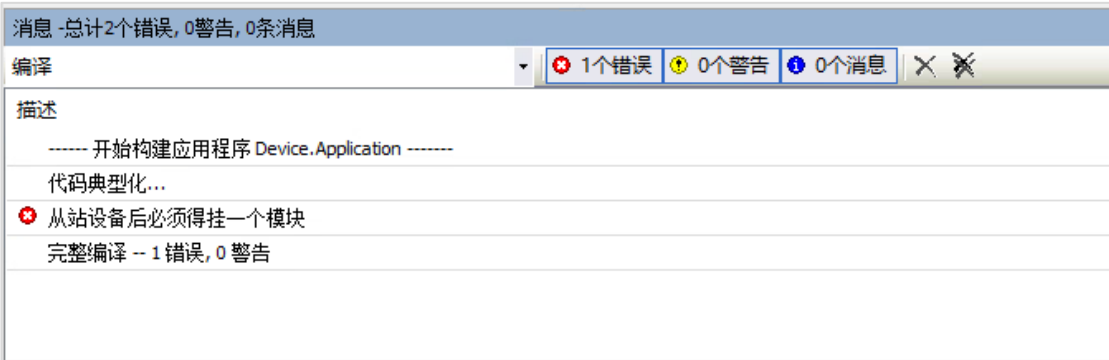


信捷 402 轴支持 Home 回原界面配置。

3-1-4. 组态编辑错误定位

组态设备制定了一些配置规则及错误检测机制。比如网络组态中两个 MODBUS 设备的站号相同，或是 TCP 设备的 IP 地址相同；硬件组态设备中扩展机架上的从站设备后没有接 IO 模块；EtherCAT 挂载的非失能轴数超出可支持的范围等都会导致组态编译报错。

在编译工程时，如果出现组态错误，XS Studio 消息输出框中会显示，双击其中的错误列表可以自动定位到对应的组态界面。



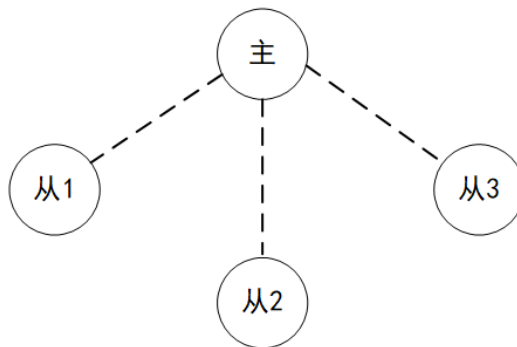
3-2. MODBUS 通讯

XS Studio 支持 Modbus 协议通讯主、从机形式。

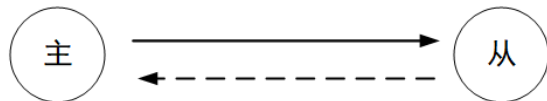
主站形式：可编程控制器作为主站设备时，可与其它使用 Modbus 协议的从机设备通讯；与其他设备进行数据交换。例：信捷 XS 系列 PLC，可以通过通讯来控制变频器。

从站形式：可编程控制器作为从站设备时，只能对其它主站的要求作出响应。

主从的概念：在 RS485 网络中，某一时刻，可以有一主多从（如下图），其中主站可以对其中任意从站进行读写操作，从站之间不可直接进行数据交换，主站需编写通讯程序，对其中的某个从站进行读写，从站无需编写通讯程序，只需对主站的读写进行响应即可。（接线方式：所有的 485+ 连在一起，所有的 485- 连在一起）



在 RS232 网络中（如下图），只能一对一通讯，某一时刻只有一主一从。

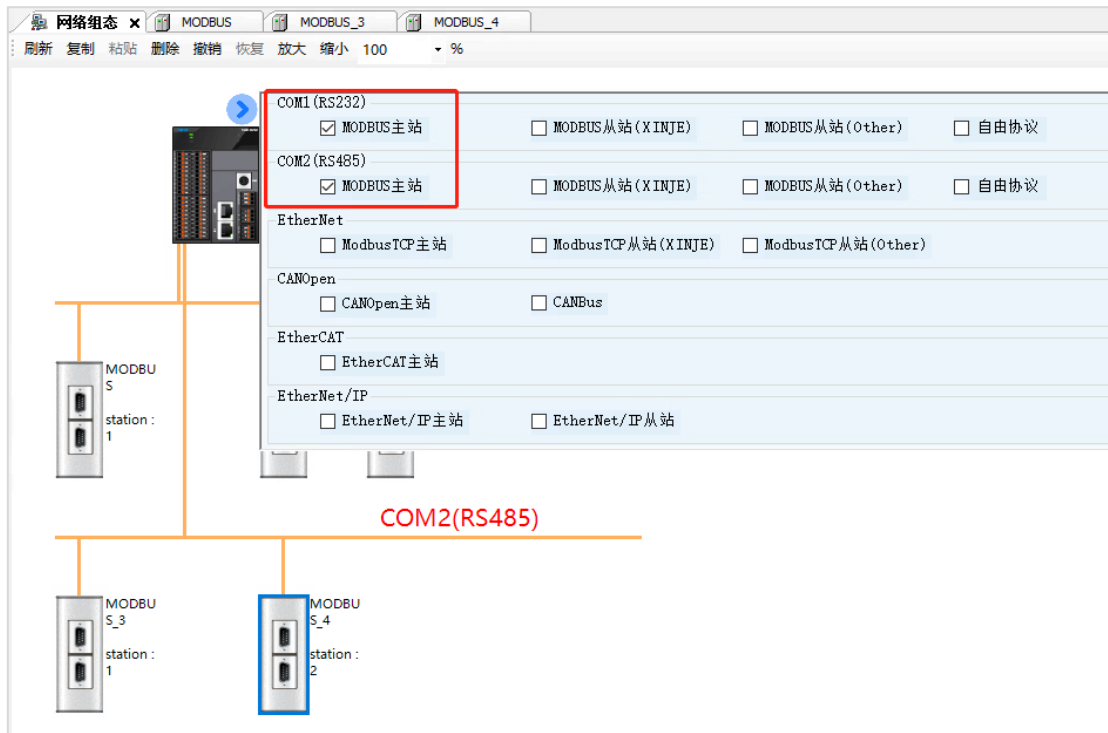


之所以图中有虚线箭头（包括 RS485 网络中），是因为理论上在两个网络中，只要各个 PLC 不发数据，网络中任意 PLC 都可以用来作为主站，其它 PLC 作为从站；但是由于多个 PLC 之间没有一个统一的时钟基准，容易出现在同一时刻有多个 PLC 发送数据，会导致通讯冲突失败，因此不建议这样使用。

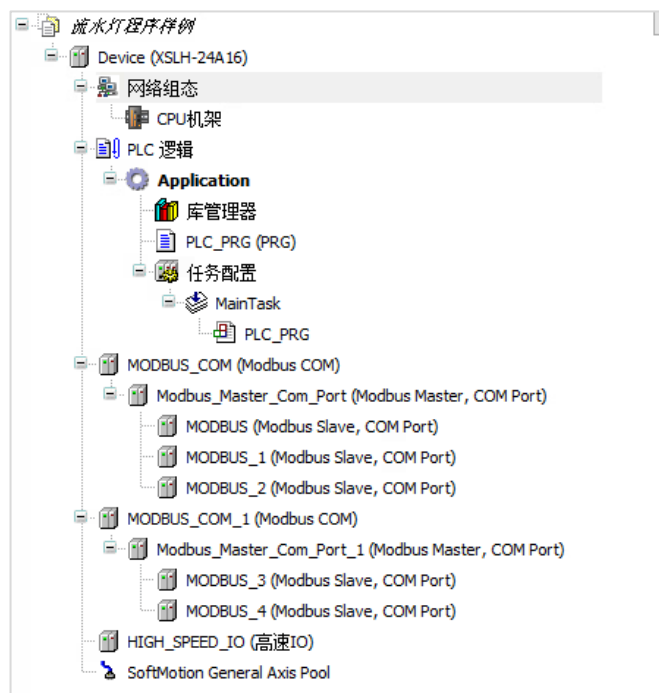
3-2-1. MODBUS 主站配置

1、使能主站、添加主站

单击网络组态中的 PLC 设备，会显示 PLC 内部所支持的主/从站的使能窗口。如下图所示：单击窗口中的复选框按钮来使能 CPU 所支持的主/从站功能，再从视图右侧的“网络连接设备列表”中单击“MODBUS”将从站添加到网络图中。



此时，在界面左侧视图中将出现 Modbus 组态配置对应设备树，如下图所示：



2、主站通信配置

PLC 做 Modbus 主站时，双击左侧设备树内的“Modbus_COM”节点，打开 Modbus 通讯配置界面，相关配置界面如下：



COM 端口	该主站物理连接选择串口 0 或串口 1
波特率	通信时的速率
奇偶校验	通信帧的校验方式
数据位	通信帧包含的实际数据位
停止位	通信时表示单个包的最后位

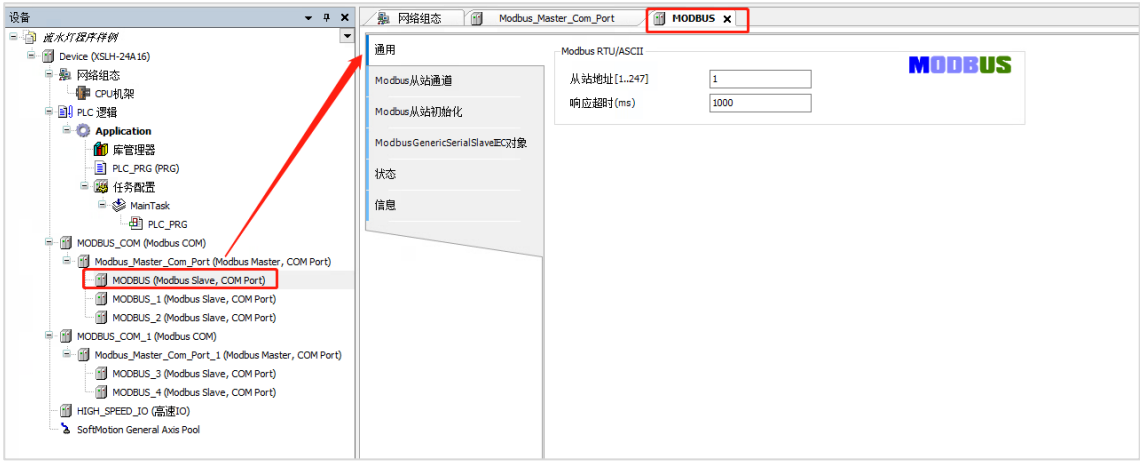
双击设备树中的主站设备，打开 Modbus 主站通讯参数配置窗口，配置界面如图：



传输模式	选择 RTU 或 ASCII 码
响应超时（ms）	指主站等待从站响应的时间间隔。如果在这段时间中从站没有发出响应，主站将会请求下一个从站。此时输入的值会认为是每个从站的缺省值。在从站配置页面，每个从站可单独设置合适的时间间隔
帧之间的时间（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

至此，主站的配置结束，接下来，需要对主站连接的从站做相应的配置。

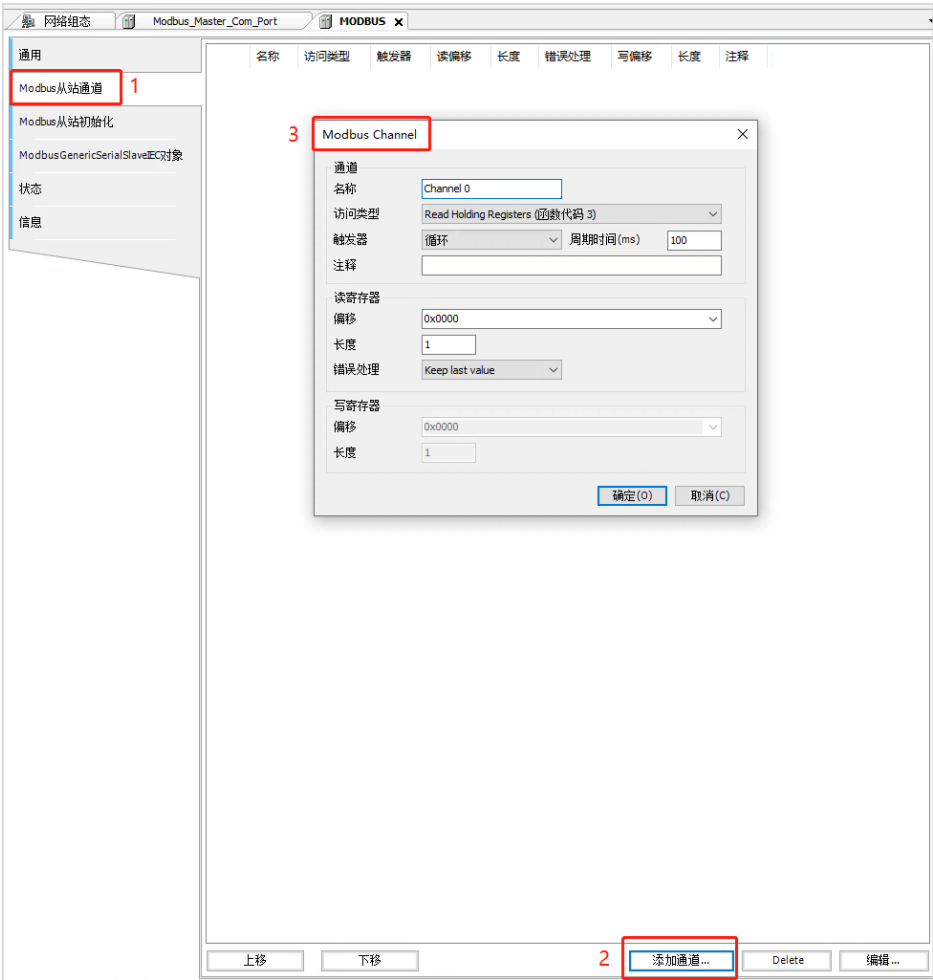
主站配置完成后，双击 MODBUS(Modbus Slave,COM Port)节点打开从站配置界面，如图所示：



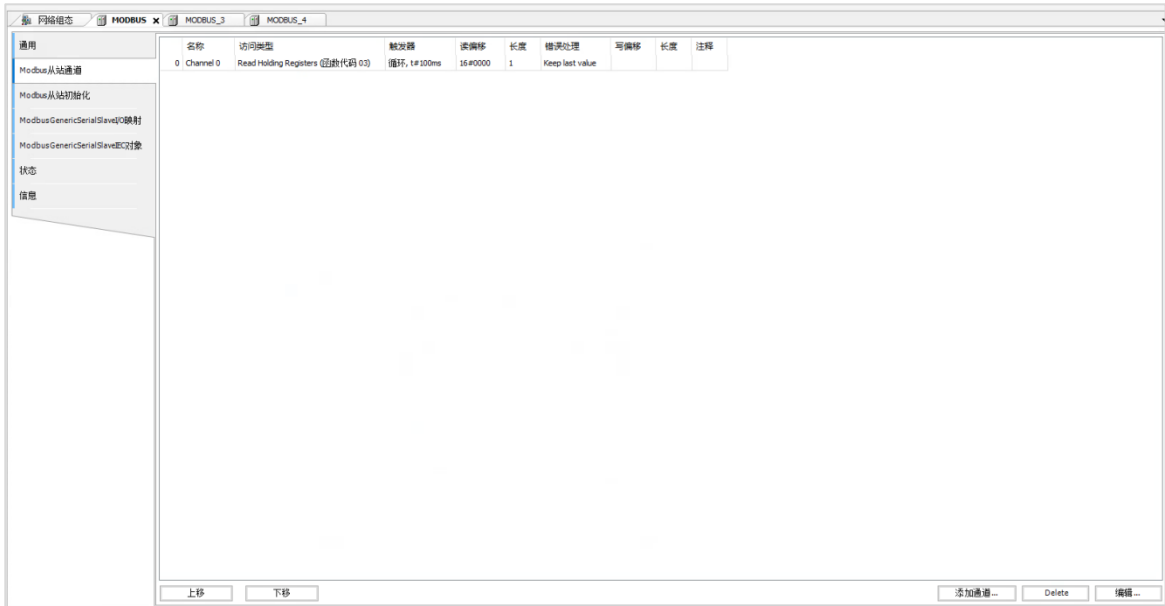
从站地址：设置从站的站地址，1~247 有效。

响应超时：设置从站的响应超时时间，如果超过该时间从站还没有响应主站，则主站认为该从站有通讯故障。

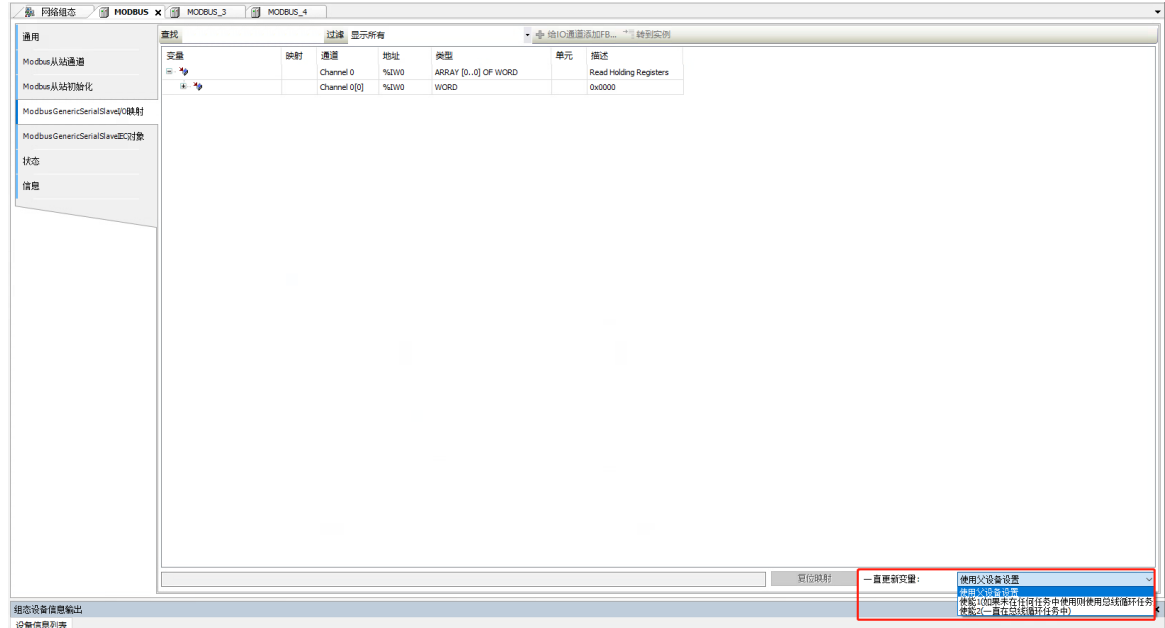
设置从站的通讯通道如图所示，在该设置选项中，用户可以自定义从站的 Modbus 通讯通道，但必须与实际的从站硬件相匹配，点击“添加通道”后，系统会自动弹出 Modbus Channel 对话框，用户可以直接选择访问类型、地址偏移、数据长度及通讯周期时间等。



通道添加成功后，如下图：



此处用户需根据实际需求设置“一直更新变量”，可从下拉框下拉选择使能 1 或使能 2。如下图：



3-2-2. MODBUS 从站（Other）配置

可通过网络组态界面内的使能窗口使能从站设备，左侧视图会相应的生成从站设备节点，如下图所示：

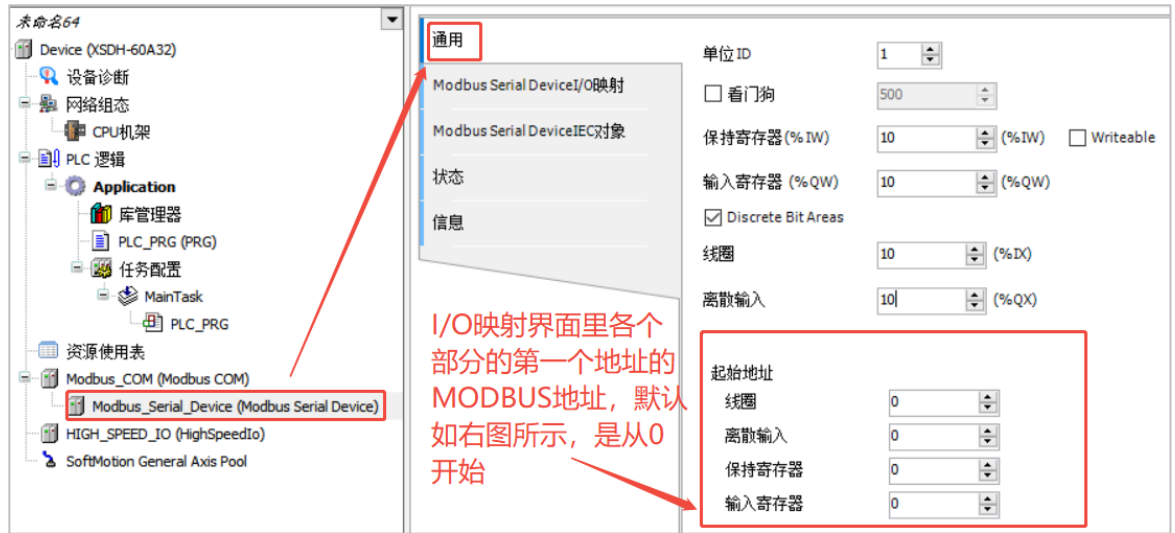


配置 PLC 的串口参数，具体如下图：



COM 端口	该主站在网络组态中选择的串口号
波特率	通信时的速率
奇偶校验	通信帧的校验方式
数据位	通信帧包含的实际数据位
停止位	通信时表示单个包的最后位

点击左侧设备树内“Modbus_Serial_Device (Modbus Serial Device)”节点，打开 Modbus 从站通讯数据配置界面。具体如下图：

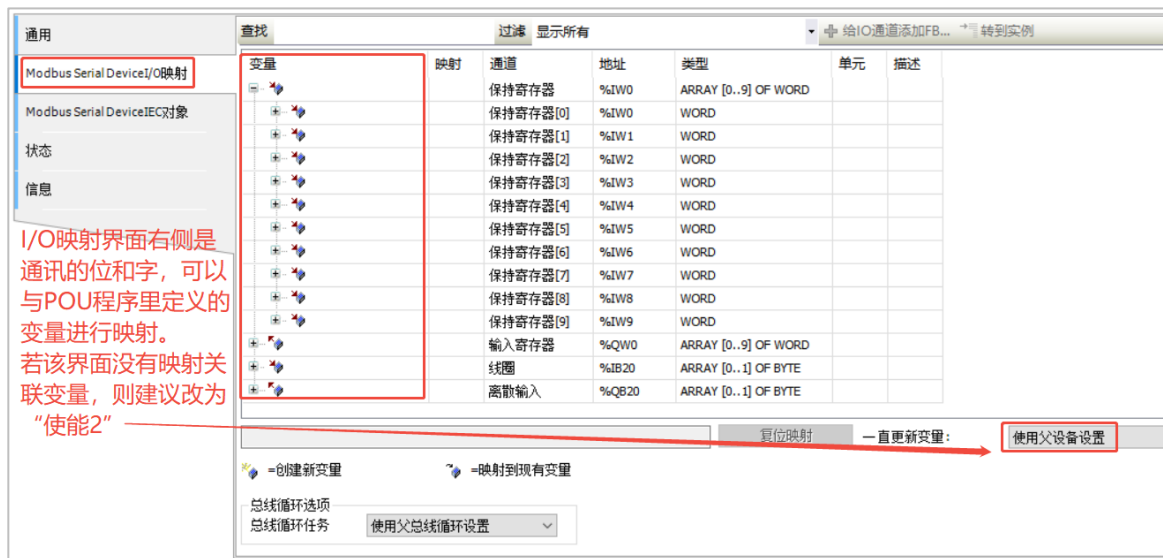




可以配置通讯的字和位的数量，注意 MODBUS 地址分配。

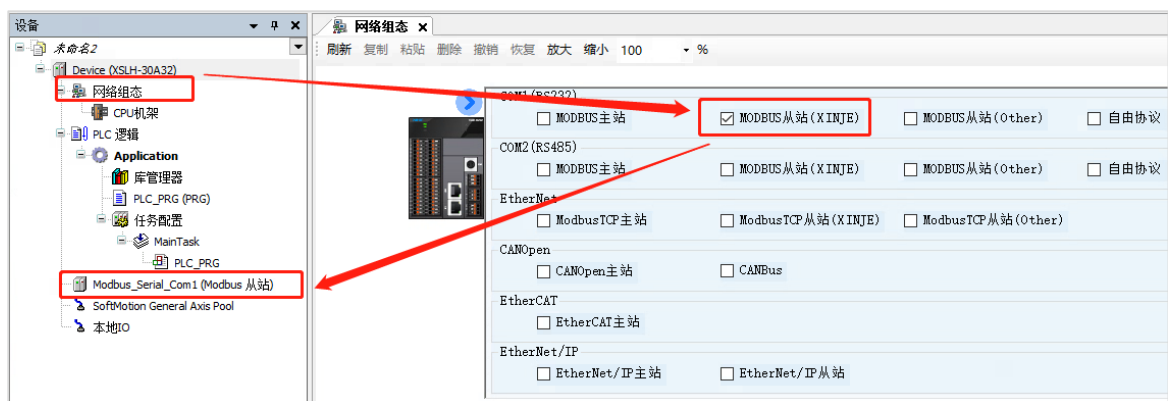
- 保持寄存器对应触摸屏上的 4X（可读可写）。
- 输入寄存器对应触摸屏上的 3X（只读不写）。
- 线圈对应触摸屏上的 0X（可读可写）。
- 离散输入对应触摸屏上的 1X（只读不写）。

在该窗口内切换到“Modbus Serial DeviceI/O 映射”，用户需要按照实际需求设置“一直更新变量”等参数，具体如下图：



3-2-3. MODBUS 从站 (XINJE) 配置

从左侧设备树双击“网络组态”节点，打开网络组态界面，通过使能窗口使能 Modbus 从站(XINJE)设备，同时会在左侧设备树内生成“Modbus_Serial_Com1 (Modbus 从站)”节点。具体如下图：



左侧设备树内双击“Modbus_Serial_Com1 (Modbus 从站)”节点打开 Modbus 参数配置界面。可根据实际需求设置相关的串口参数。默认情况如下图所示：



Modbus 从站配置参数如下：

COM 端口	该主站在网络组态中选择的串口号
波特率	通信时的速率
数据位	通信帧包含的实际数据位，当模式选择为 RTU，模式数据为 8 位
校验位	通信帧的校验方式
停止位	通信时表示单个包的最后位
模式	RTU
站号	本设备站号，范围 1~247

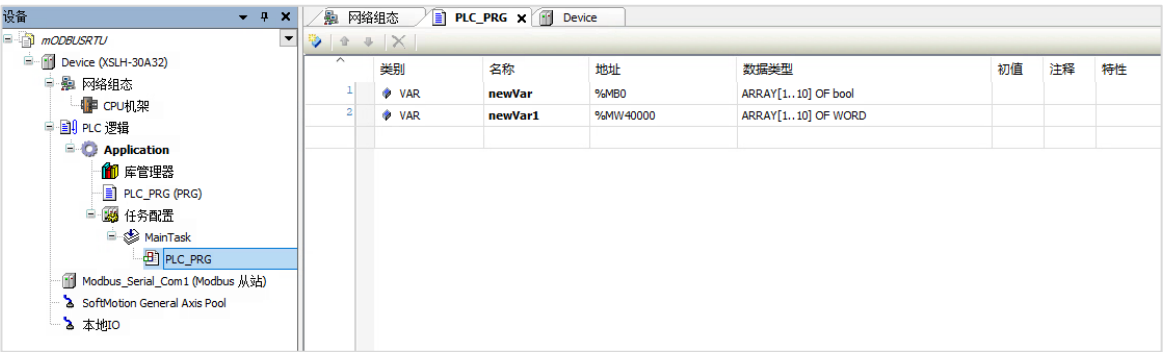
PLC 做 Modbus RTU (X1NJE) 从站设备时，可以被主站设备访问的地址范围定义如下：

- ◆ 所有的线圈操作（功能码 0x01、0x02、0x05、0x0F）可读写地址为：%MB0-%MB65534；
- ◆ 所有的寄存器操作（功能码 0x03、0x04、0x06、0x10）可读写地址为：%MW40000-%MW105534。

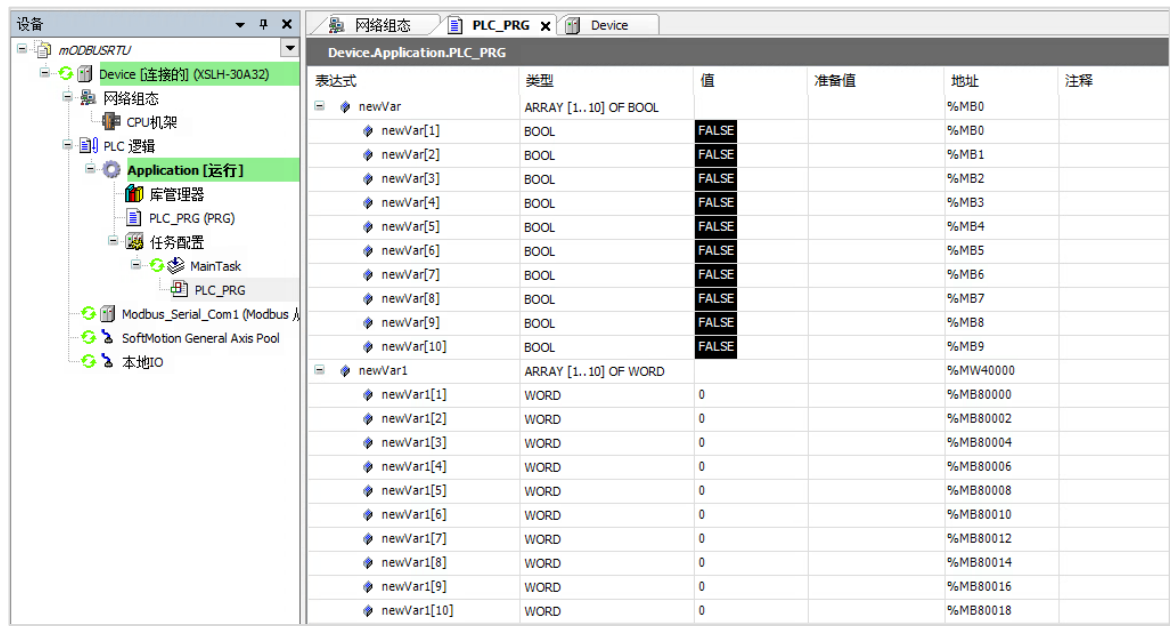
 应用举例：

此处 XS Studio 软件作为从站，使用第三方调试工具 Modbus Poll 作为主站建立连接并进行串口通讯，实现接收或发送寄存器或线圈数据。

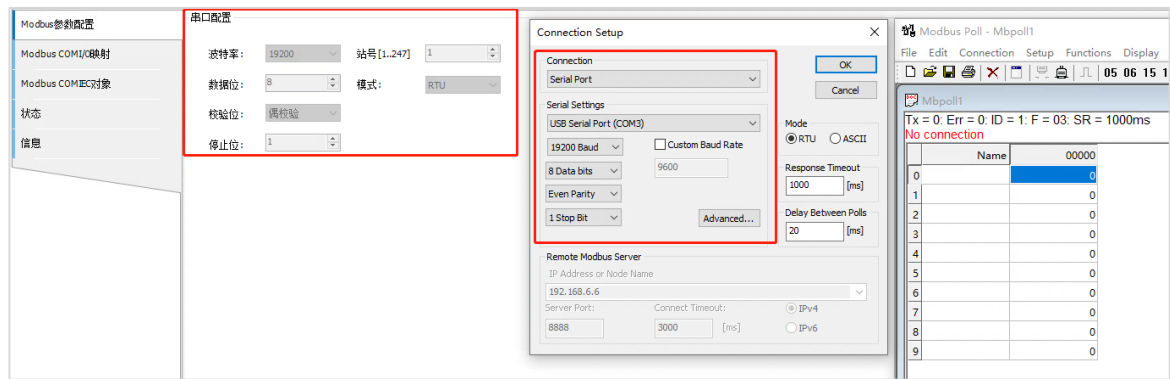
1) 在“PLC-PRG”编辑器内申明两个变量，分别接收与发送寄存器或线圈数据。如下图：



2) 与 XSLH-30A32 设备建立连接，并登录后运行。如下图：

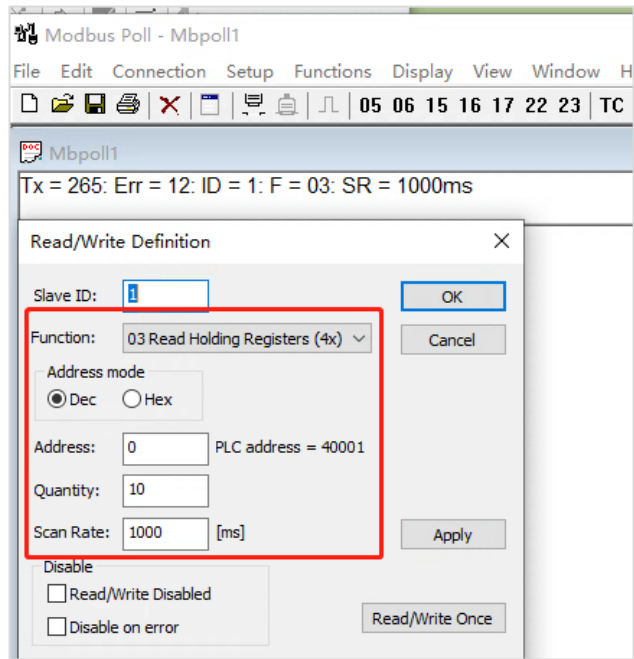


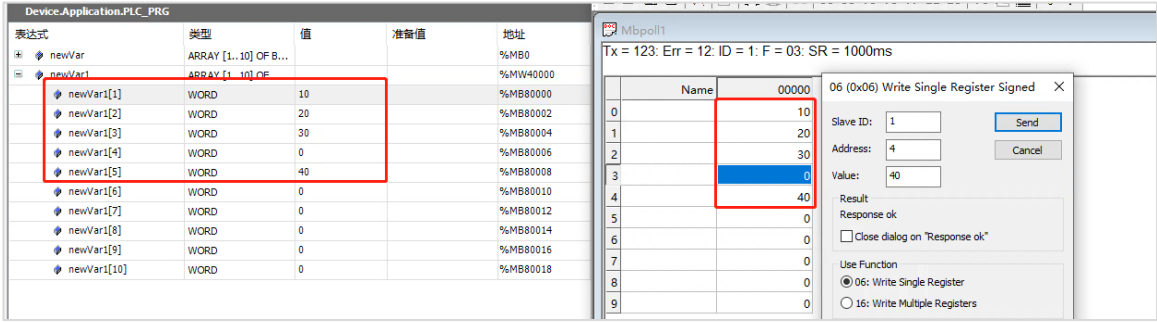
3) 配置 Modbus Poll 的相关参数，与从站的串口配置信息保持一致，否则无法成功建立连接。如下图所示：



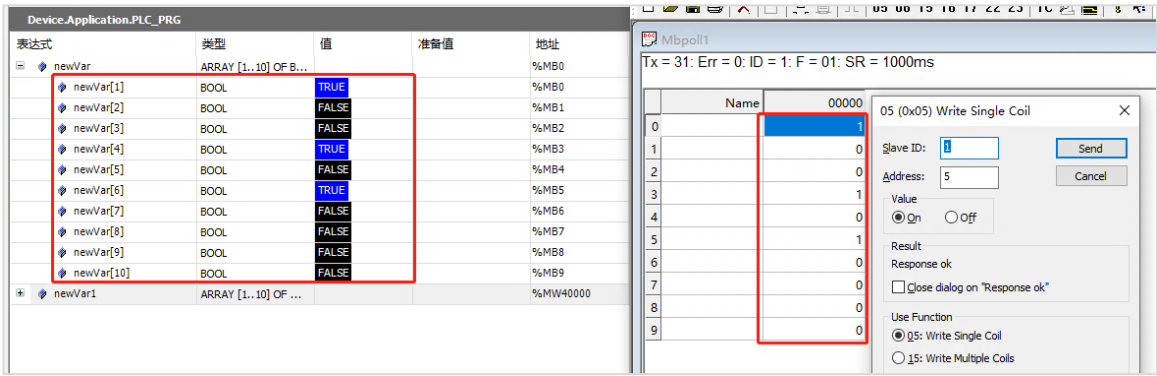
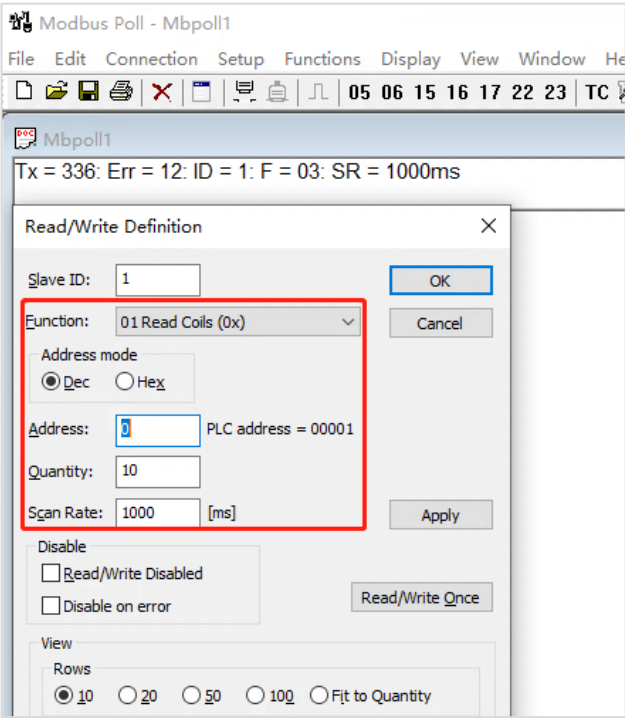
4) 配置主站设备的相关读写参数，即可与从站进行相关的读/写寄存器或线圈操作。

◆ 读/写寄存器操作，如下图所示：





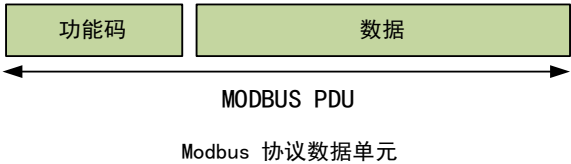
◆ 读/写线圈操作。如下图所示：



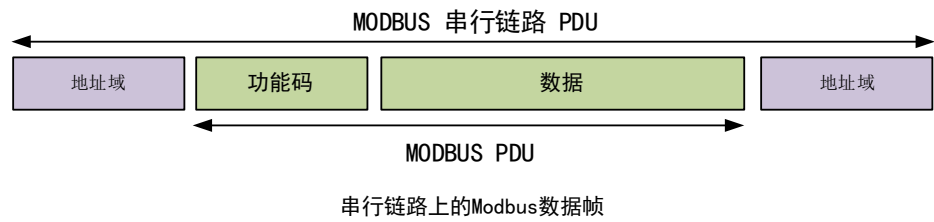
此时主站与从站已成功通信。

3-2-4. MODBUS 通讯帧格式说明

《Modbus 应用协议》定义了简单的协议数据单元 (PDU - Protocol Data Unit)：



发起 Modbus 事务处理的客户端构造 Modbus PDU，然后添加附加的域以构造通信 PDU。



1、在 Modbus 串行链路，地址域只含有子节点地址。

如前文所述，合法的子节点地址为十进制 0~247，每个子设备被赋予 1~247 范围中的地址，主节点通过将子节点的地址放到报文的地址域对子节点寻址。当子节点返回应答时，它将自己的地址放到应答报文的地址域以让主节点知道哪个子节点在回答。

2、功能码指明服务器要执行的动作。功能码后面可跟表示含有请求和响应参数的数据域。

3、错误检验域是对报文内容执行“冗余校验”的计算结果。

根据不同的传输模式 (RTU or ASCII) 使用两种不同的计算方法。

有两种串行传输模式被定义：RTU 模式和 ASCII 模式。所有设备必须实现 RTU 模式，ASCII 传输模式是选项。Modbus RTU 一般采用串口 RS232C 或 RS485/422，而 Modbus TCP 一般采用以太网口。

1) RTU 传输模式

当设备使用 RTU (Remote Terminal Unit)模式在 Modbus 串行链路通信，报文中每个 8 位字节含有两个 4 位十六进制字符。这种模式的主要优点是较高的数据密度，在相同的波特率下比 ASCII 模式有更高的吞吐率。每个报文必须以连续的字符流传送。

RTU 模式每个字节（11 位）的格式为：

编码系统：8 位二进制，报文中每个 8 位字节含有两个 4 位十六进制字符 (0~9，A~F)

每字节的 bit 流：

- ◆ 1 起始位
- ◆ 8 数据位，首先发送最低有效位
- ◆ 1 位作为奇偶校验
- ◆ 1 停止位

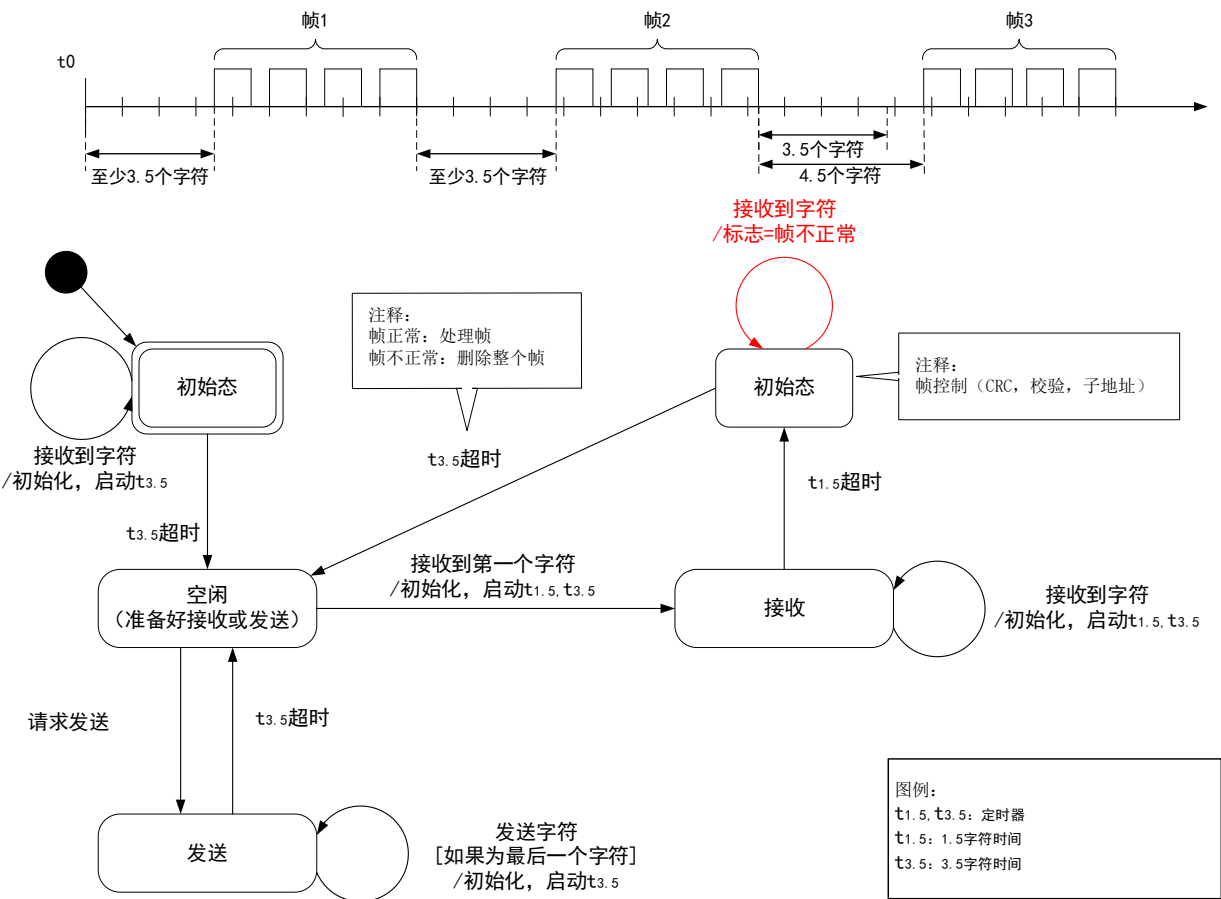
偶校验是要求的，其它模式 (奇校验，无校验) 也可以使用。为了保证与其它产品的最大兼容性，同时支持无校验模式是建议的。默认校验模式必须为偶校验。



使用无校验要求 2 个停止位。

地址码	功能码	数据	校验码
1 字节	1 字节	n 节	2 字节 (CRC)

在 RTU 模式，报文帧由时长至少为 3.5 个字符时间的空闲间隔区分。在后续的部分，这个时间区间被称作 t3.5。



- ◆ 从“初始”态到“空闲”态转换需要 t3.5 定时超时：这保证帧间延迟；
- ◆ “空闲”态是没有发送和接收报文要处理的正常状态；
- ◆ 在 RTU 模式，当没有活动的传输的时间间隔达 3.5 个字符长时，通信链路被认为在“空闲”态。
- ◆ 当链路空闲时，在链路上检测到的任何传输的字符被识别为帧起始。链路变为“活动”状态。然后，当链路上没有字符传输的时间间隔达到 t3.5 后，被识别为帧结束
- ◆ 检测到帧结束后，完成 CRC 计算和检验。然后，分析地址域以确定帧是否发往此设备，如果不是，则丢弃此帧。为了减少接收处理时间，地址域可以在一接到就分析，而不需要等到整个帧结束。这样，CRC 计算只需要在帧寻址到该节点（包括广播帧）时进行。

2) ASCII 传输模式

当 Modbus 串行链路的设备被配置为使用 ASCII 模式通信时,报文中的每个 8 位字节以两个 ASCII 字符发送。当通信链路或者设备无法符合 RTU 模式的定时管理时使用该模式。



由于一个字节需要两个字符，此模式比 RTU 效率低。

ASCII 模式每个字节（11 位）的格式为：

编码系统：十六进制，ASCII 字符 0~9，A~F。报文中每个 ASCII 字符含有 1 个十六进制字符。

每字节的 bit 流：

- ◆ 1 起始位；
- ◆ 8 数据位，首先发送最低有效位；
- ◆ 1 位作为奇偶校验；
- ◆ 1 停止位。

偶校验是要求的，其它模式（奇校验、无校验）也可以使用。为了保证与其它产品的最大兼容性，

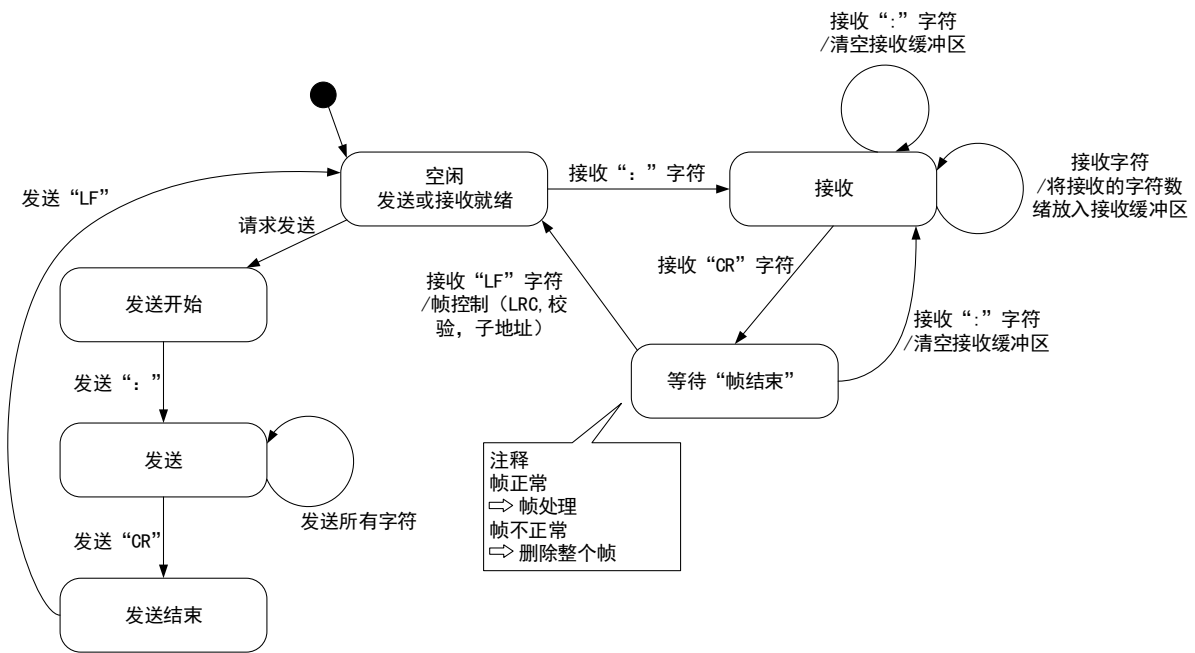
同时支持无校验模式是建议的。默认校验模式必须为偶校验。



使用无校验要求 2 个停止位。

起始	地址码	功能码	数据	校验码	回车换行
字符 “:” (冒号)	2 字节	2 字节	0 到 2*252 字节	2 字节 (LRC 校验)	2 字节 (CR, LF)

报文帧的地址域含有两个字符。
在 ASCII 模式, 报文用特殊的字符区分帧起始和帧结束。一个报文必须以一个“冒号”(ASCII 十六进制 3A) 起始, 以“回车-换行”对应 (ASCII 十六进制 0D 和 0A) 结束。
对于所有的域, 允许传送的字符为十六进制 0~9, A~F (ASCII 编码)。设备连续的监视总线上的“冒号”字符。当收到这个字符后, 每个设备解码后续的字符一直到帧结束。
报文中字符间的时间间隔可以达一秒。如果有更大的间隔, 则接受设备认为发生了错误。



ASCII 传输模式状态图

- “空闲”态是没有发送和接收报文要处理的正常状态。
- 每次接收到“:”字符表示新的报文的开始。如果在一个报文的接收过程中收到该字符, 则当前地报文被认为不完整并被丢弃。而一个新的接收缓冲区被重新分配。
- 检测到帧结束后, 完成 LRC 计算和检验。然后, 分析地址域以确定帧是否发往此设备, 如果不是, 则丢弃此帧。为了减少接收处理时间, 地址域可以在一接到就分析, 而不需要等到整个帧结束。

3-3. 串口自由协议通讯

3-3-1. 概要

信捷 PLC 在与其它设备通讯的时候，如果作为下位机，则上位机必须按照 MODBUS RTU 的数据格式与其进行数据交换；如果信捷 PLC 作为上位机，当下位机也支持 MODBUS RTU 协议的时候，可直接使用相关通讯指令进行通讯，使程序编写更简单，效率更高，如果下位机不直接支持 MODBUS RTU 协议，则可使用自由格式通讯。

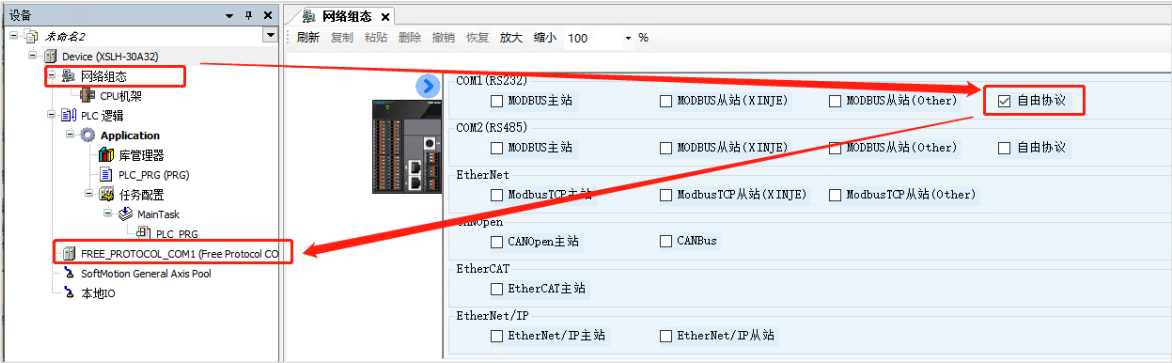
所谓自由格式，就是当下位机通讯协议与 PLC 协议不匹配时候，PLC 内部自定义数据格式来发送数据，这样可以和很多下位机进行通讯。

自由格式通讯是以数据块的形式进行数据传送，每块最大可传送 1024 字节，同时每块可设置一个起始符和终止符，也可以不设。

3-3-2. 串口配置

以 XSLH-30A32 机型设备为例，进行串口自由协议配置。

从左侧设备树双击“网络组态”节点，打开网络组态界面，从使能窗口将“COM1”配置成“自由协议”，配置完成后，在左侧设备树内生成“FREE_PROTOCOL_COM1(Free Protocol COM)”节点。如下图所示。



3-3-3. 通讯配置

1) 从左侧设备树内双击“FREE_PROTOCOL_COM1(Free Protocol COM)”节点打开自由协议参数配置界面。参数配置界面如下：



串口配置参数如下：

COM 端口	该主站物理连接的串口号
波特率	通信时的速率
数据位	通信帧包含的实际数据位
校验位	通信帧的校验方式
停止位	通信时表示单个包的最后位
帧超时	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
起始符	设置起始符后 PLC 在发送数据时，自动加上起始符，在接收数据时，自动去掉起始符，可以看成是协议中数据帧头
终止符	设置终止符后 PLC 在发送数据时，自动加上终止符，在接收数据时，自动去掉终止符，可以看成是协议中数据帧尾
缓冲位数	缓存位数可设置为 8 位或 16 位，当缓存位数为 8 位时，只发送寄存器的低字节数据；当缓存位数为 16 位时，寄存器的高低字节数据都将被发送，低字节在前，高字节在后

自由协议参数配置如下：

最大接收长度：一次可接收的最大数据字节数。默认 256 字节，最大允许设置为 1024 字节；
最大发送长度：一次可发送的最大数据字节数。默认 256 字节，最大允许设置为 1024 字节。

2) 从当前自由协议参数配置窗口切换到 FreeProtocolI/O 映射界面。发送或接收数据的通道具体如下：

自由协议配置	查找 过滤 显示所有 给IO通道添加FB... 转到实例						
FreeProtocolI/O映射	变量	映射	通道	地址	类型	单元	描述
FreeProtocolI/O对象	+		send data size	%QW0	WORD		发送数据长度
			send data buffer	%QW1	ARRAY [0..4] OF WORD		发送数据缓存
状态			actual send data size	%IW0	WORD		实际发送数据长度
			actual receive data size	%IW1	WORD		实际接收数据长度
信息	+		receive data	%IW2	ARRAY [0..4] OF WORD		接收数据缓存

自由协议 I/O 映射参数如下表：

通道	描述
send data size	发送数据长度；以填写非零数据为发送字节数进行触发发送，发送完数据后自动将该寄存器进行复位位置 0，发送数据时需要打断接收数据的准备状态，待发送数据结束后恢复接收数据的准备状态。
Send data buffer	发送数据缓存；将待发送的数据缓存在发送数据缓存区中，等待发送信号发送数据。
actual send data size	实际发送数据长度；以实际发送出去字节数进行显示。
actual recive data size	实际接收数据长度；以实际接收数据的字节数进行显示。
receive data	接收数据；将接收到的数据按照缓存所选的模式进行存放到对应的地址中，接收数据处于一直准备接收数据状态，除发送数据进行打断准备状态外。

3-3-4. 应用举例

此处 XS Studio 软件作为从站，触摸屏作为主站建立连接并进行 Modbus 串口自由协议通讯，实现接收或发送数据。

1、配置 XS Studio 软件内的串口及自由协议参数配置。具体如下图；

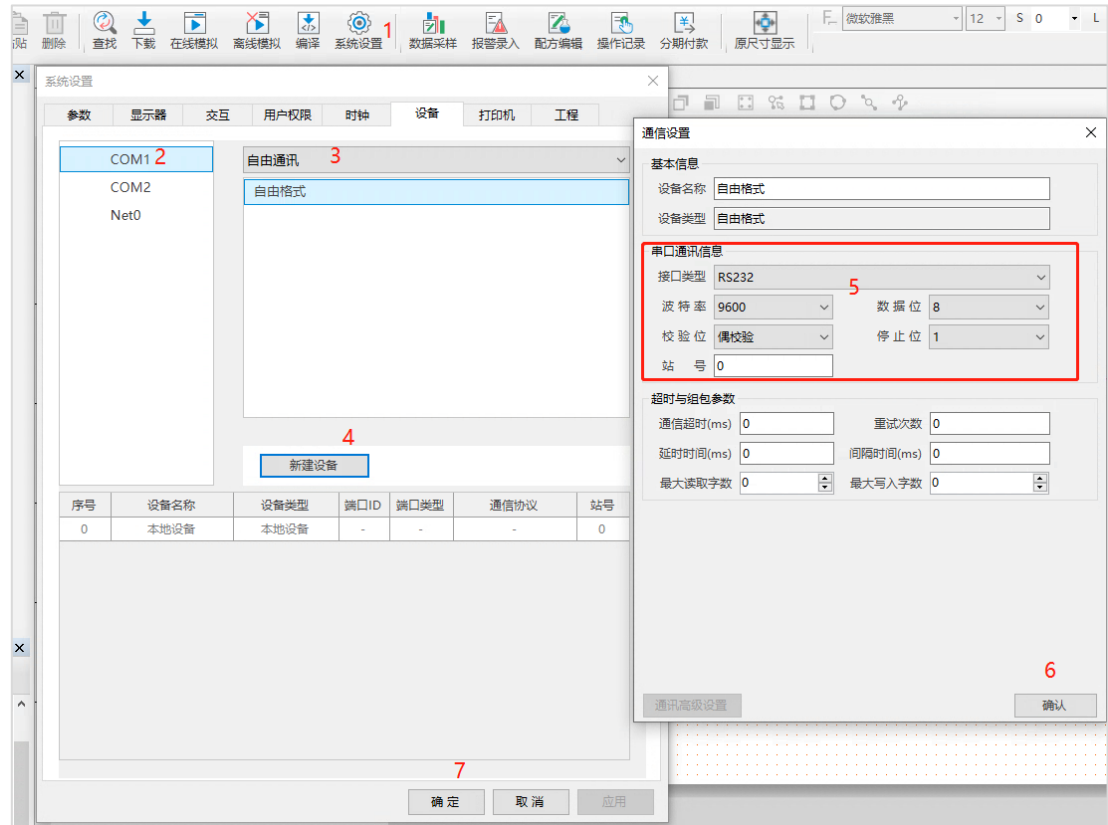


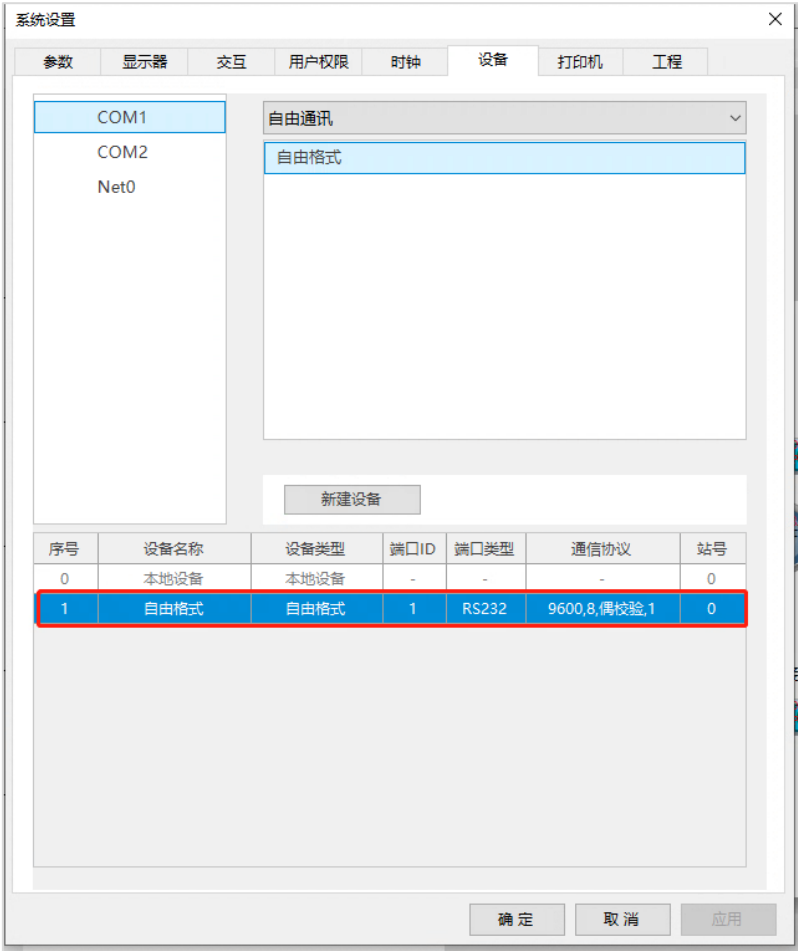
2、与 PLC 建立连接，登录并运行。如下图；



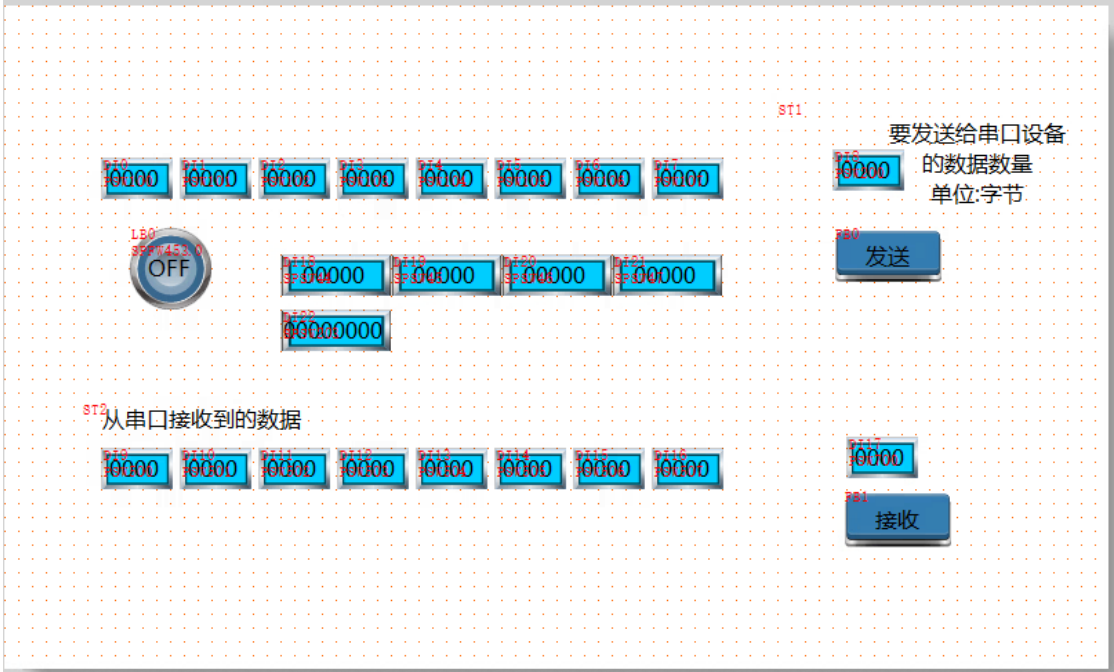
3、设置触摸屏的通信参数。

从“系统设置”窗口打开“设备”功能界面,按照实际 PLC 设备的 COM 端口选择 COM1 -> 自由通讯下的自由格式, 然后新建设备, 并进行通信设置。此处串口信息要求与 XS Studio 保持一致, 否则无法进行正确的数据交互。具体如下图所示：

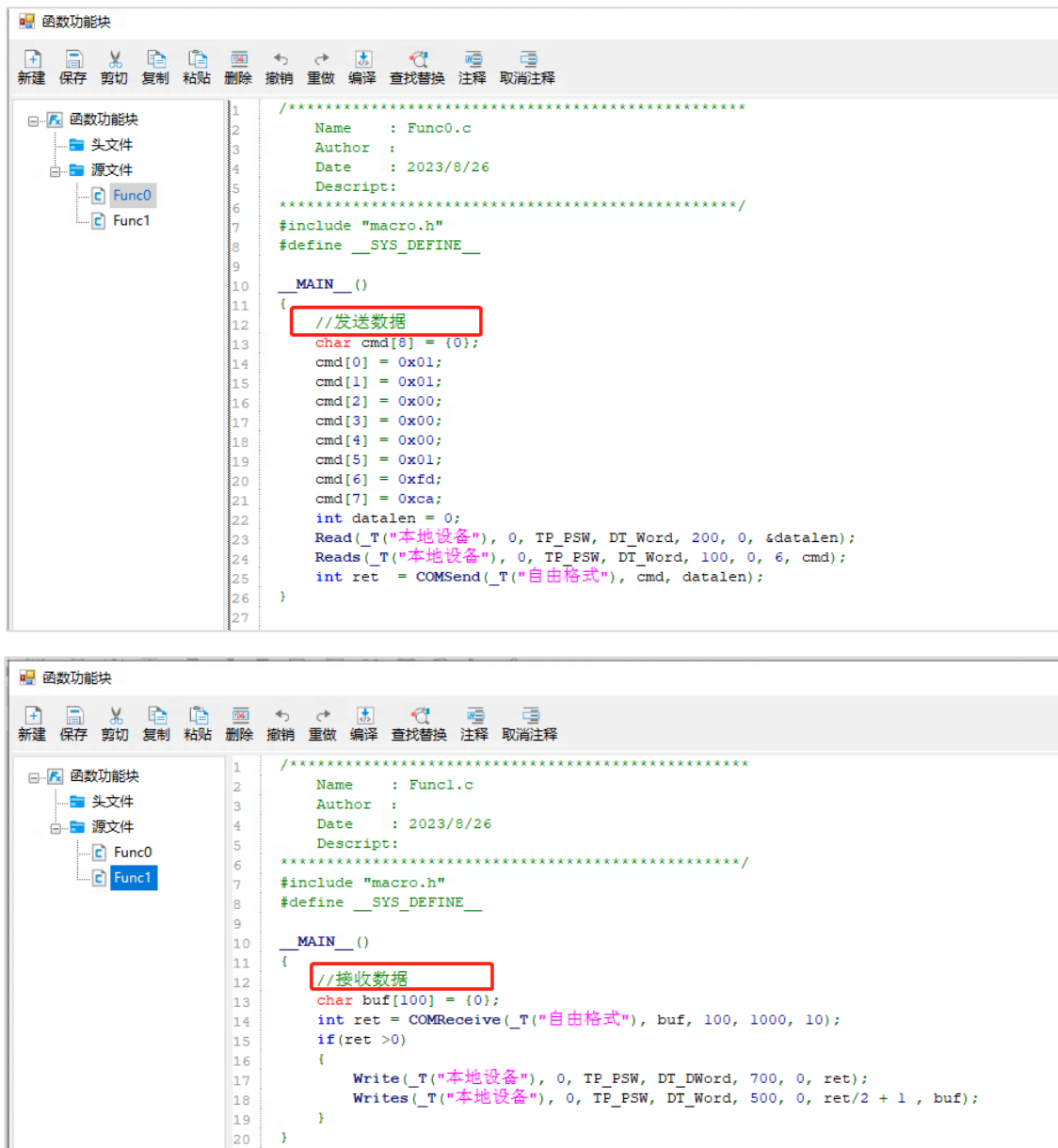




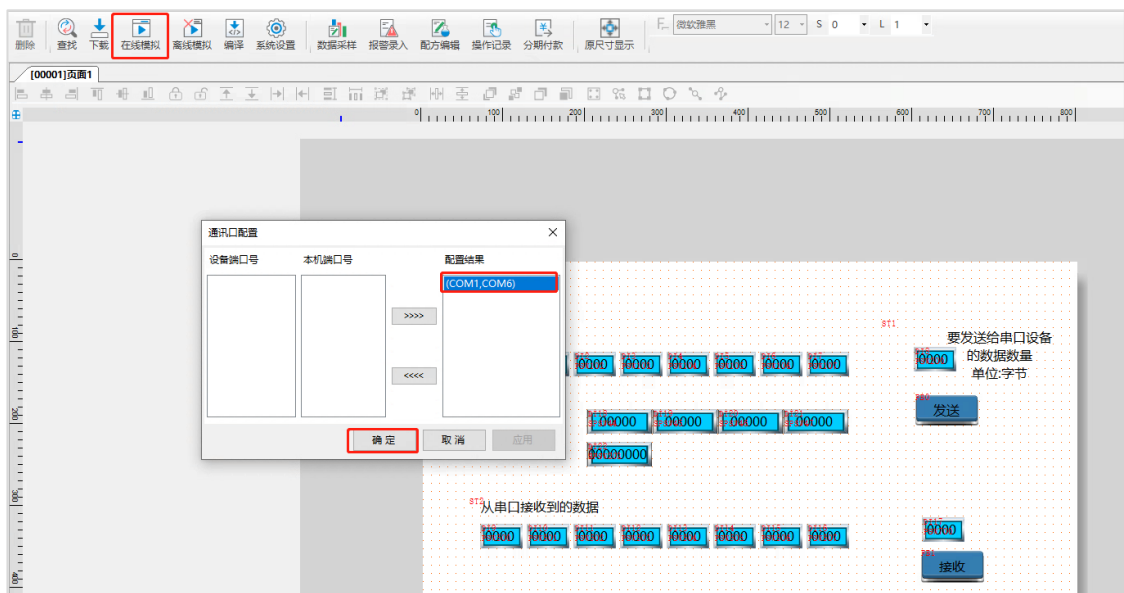
4、 在触摸屏内新建工程，添加部件。如下图：

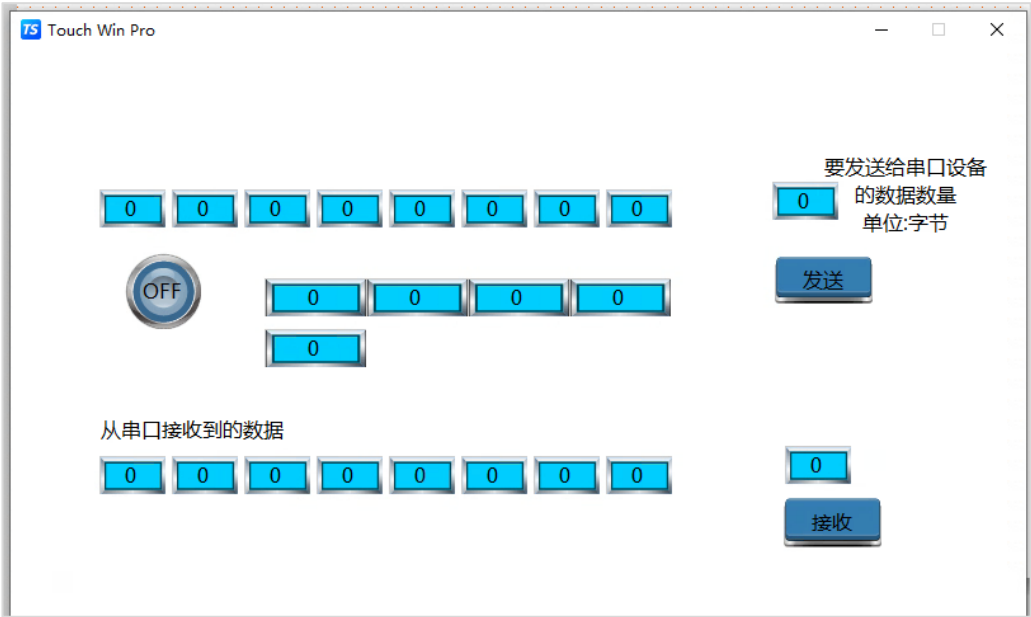


5、 建立 C 函数功能块，来接收或发送数据。如下图：



6、 接收/发送函数编辑完成后，调用函数，并选择在线模拟，与 XS Studio 建立连接。如下图：



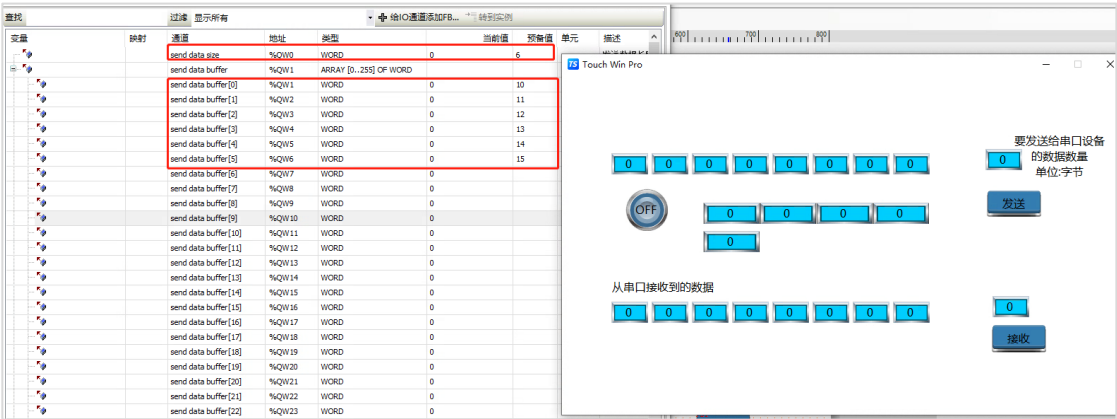


如上图，配置结果模块内的 COM1 端口为 PLC 设备实际串口连接的端口。COM6 是指实际 PC 上 USB 虚拟串口。

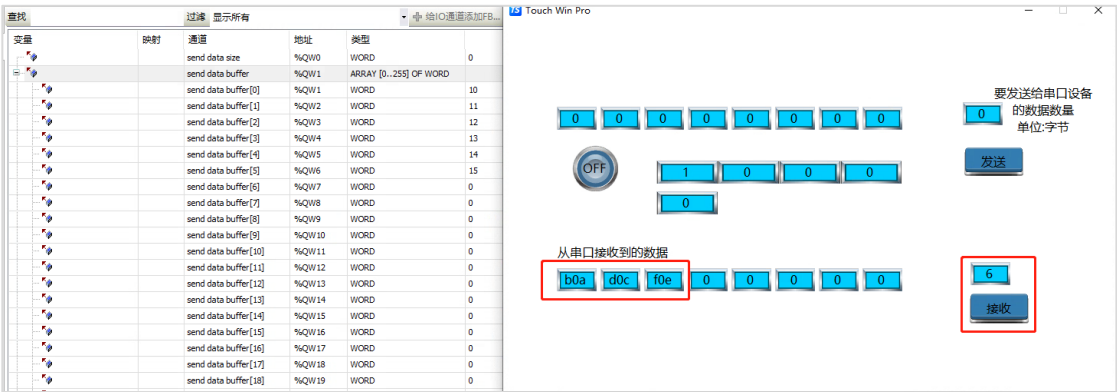
7、 XS 从站与触摸屏主站进行数据交互。具体如下图：

● XS Studio 发送数据到触摸屏

在“发送数据缓存”通道内输入自定义预备值，并设置“%QW0 发送数据长度”；如果不设置，则在写入后无法发送数据。如下图：



发送完成后，发送数据长度“%QW0”内的数据自动清零，此时触摸屏接受区域即可接收到 PLC 发送的对应数据。如下图：

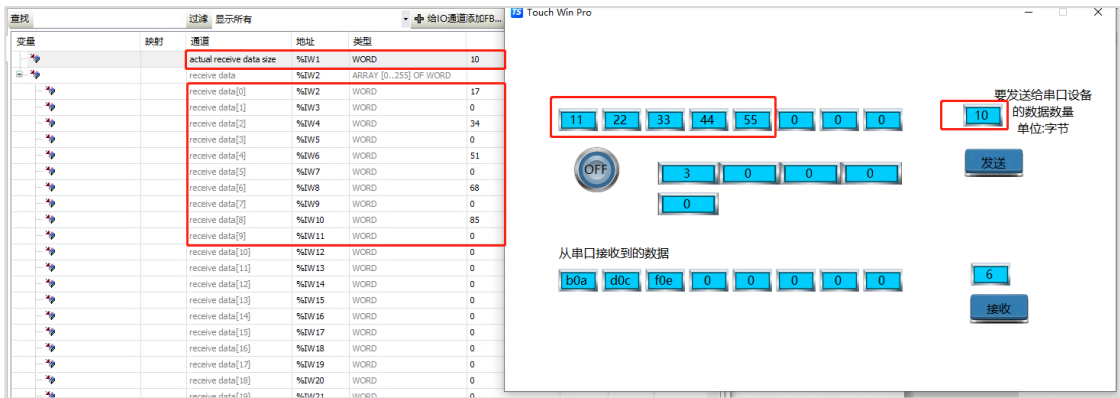


XS Studio 发送数据后，触摸屏点击接收按钮，此时按钮上方显示接收的数据长度，按钮左侧部件显示已接受的数据，此处十六进制高低字节显示。



如果 PLC 设备写入的数据长度大于设置的最大发送长度 10 个字节时，数据将按照设置的数据长度 10 发送。

● 触摸屏发送数据到 XS Studio



此时，实际接收数据长度为%IW1 = 10，接收数据缓存在“%IW2-%IW11”。



- 当主站再次向 PLC 发送数据时，会覆盖原来的数据继续从首个地址“%IW2”开始接收缓存。
- 当主站发送的数据长度超过设置的最大接收长度 256 个字节时，以 256 进行写入。

3-4. ModbusTCP 通讯

Modbus TCP 使用 TCP/IP 在站点间传送 Modbus 报文，Modbus TCP 结合了 TCP/IP 协议以及以 Modbus 协议作为应用协议标准的数据表示方法。Modbus TCP 通信报文被封装于以太网 TCP/IP 数据包中。与传统的串口方式，Modbus TCP 插入一个标准的 MODBUS 报文到 TCP 报文中，不再带有数据校验和地址。

XS 系列可编程控制器本体支持 Modbus TCP 协议通讯主、从机形式。

主站形式：可编程控制器作为主站设备时，可与其它使用 Modbus TCP 协议的从机设备通讯。一个主站最多可以连接 64 个从站。

从站形式：可编程控制器作为从站设备时，只能对其它主站的要求作出响应。

3-4-1. MODBUS TCP 主站配置



在 XSA-W 工控机桌面安装并运行上位机通讯软件时，内部通讯 IP 需按机型区分配置：

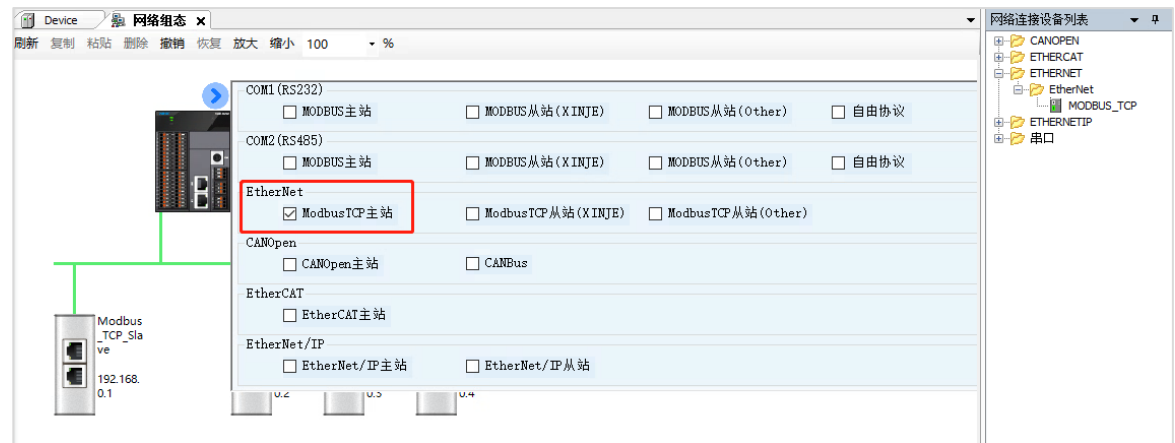
1、XSA330-W、XSA500-W 系列工控机：
设备内部通讯统一采用回送地址 127.0.0.1。

2、XSA230-W、XSA620-W 支持虚拟化的工控机：

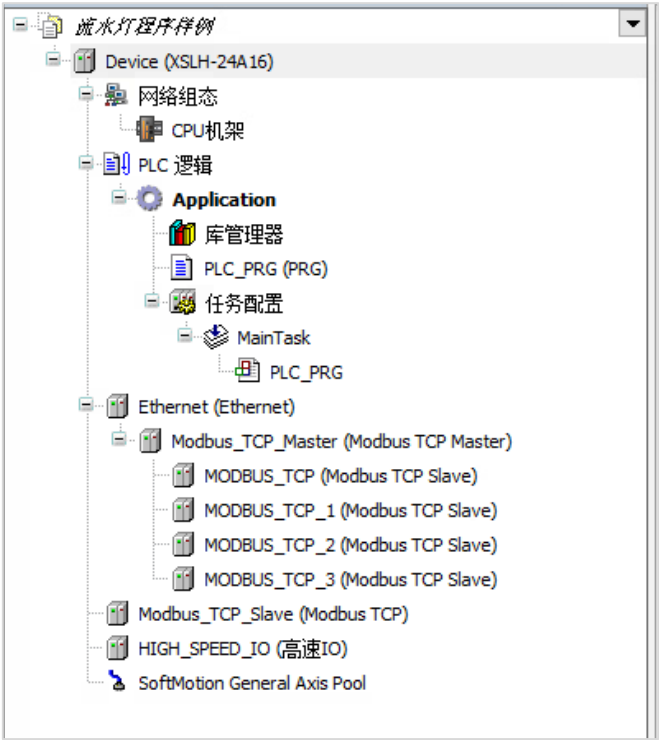
- ①内部通讯目标 IP 固定为 192.168.2.2；
- ②本机虚拟网卡默认 IP 为 192.168.2.1，可连接显示器查看；
- ③若用户需要修改虚拟网卡的 IP，则仅支持修改为 192.168.2 网段的 IP 地址，不可更换为其他网段。

1、使能主站、添加从站

单击网络组态中的 PLC 设备，会显示 PLC 内部所支持的主/从站的使能窗口。如下图所示：单击窗口中的复选框按钮来使能 CPU 所支持的主/从站功能，再从视图右侧的“网络连接设备列表”中单击“MODBUS_TCP”将从站添加到网络中。

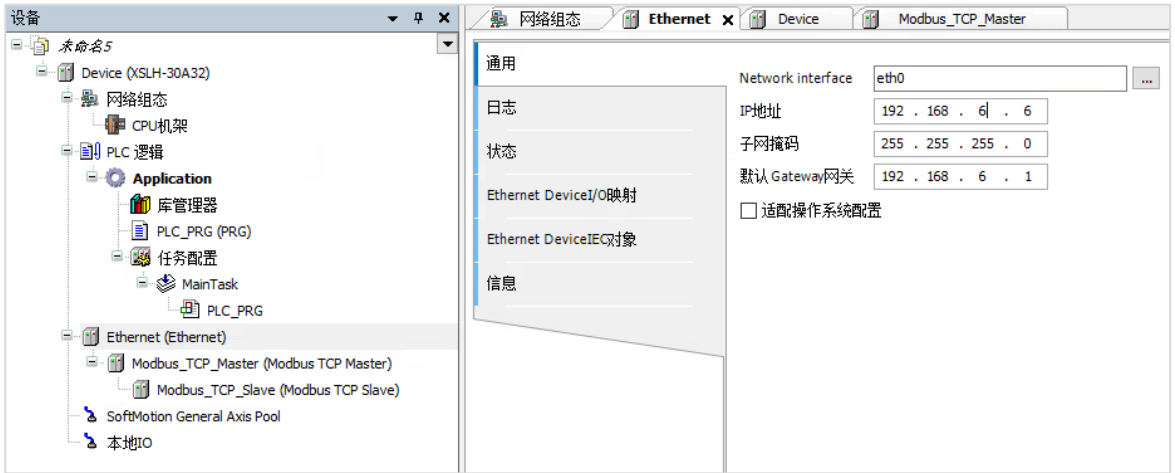
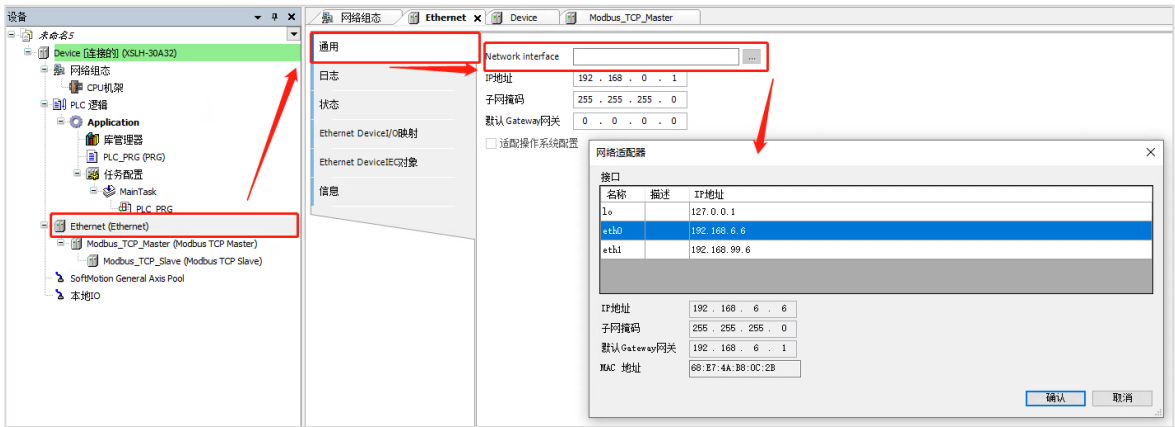


此时，在界面左侧视图中将出现 ModbusTCP 组态配置对应设备树，如下图所示：



2、主站通讯配置

PLC 做 Modbus TCP 主站时，双击设备树中的“Ethernet”，打开 Ethernet 主站配置窗口并进行配置，点击“通用”选择主站网口，配置 IP 地址等参数，具体如下图所示：



双击 MODBUS_TCP_Slave(MODBUS TCP Slave)节点打开配置界面配置从站信息，如图所示：



3-4-2. MODBUS TCP 从站（Other）配置

在 XSA-W 工控机桌面安装并运行上位机通讯软件时，内部通讯 IP 需按机型区分配置：

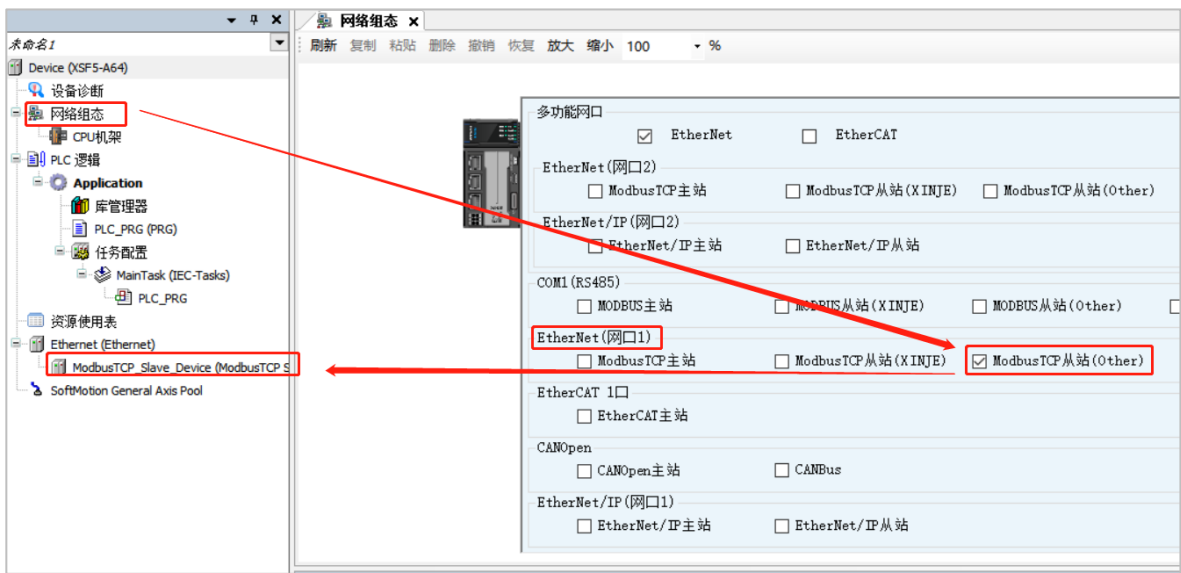
1、XSA330-W、XSA500-W 系列工控机：
设备内部通讯统一采用回送地址 127.0.0.1。

2、XSA230-W、XSA620-W 支持虚拟化的工控机：

- ①内部通讯目标 IP 固定为 192.168.2.2；
- ②本机虚拟网卡默认 IP 为 192.168.2.1，可连接显示器查看；
- ③若用户需要修改虚拟网卡的 IP，则仅支持修改为 192.168.2 网段的 IP 地址，不可更换为其他网段。

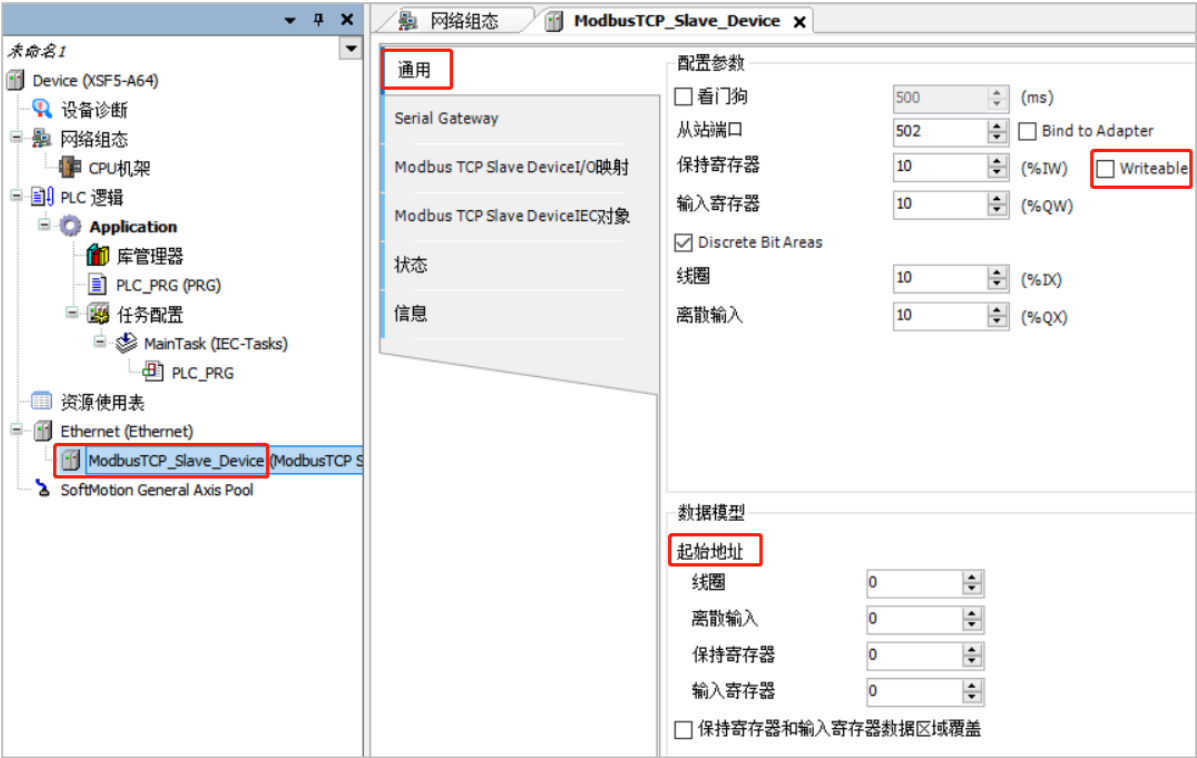
1、使能从站

可通过网络组态界面内的使能窗口使能从站设备，左侧视图会相应的生成从站设备节点，如下图所示：



2、从站通讯配置

添加从站设备后，在设备树节点下双击“Modbus_TCP_Slave_Device”节点打开配置界面，可切换至 Modbus TCP 从站配置界面。

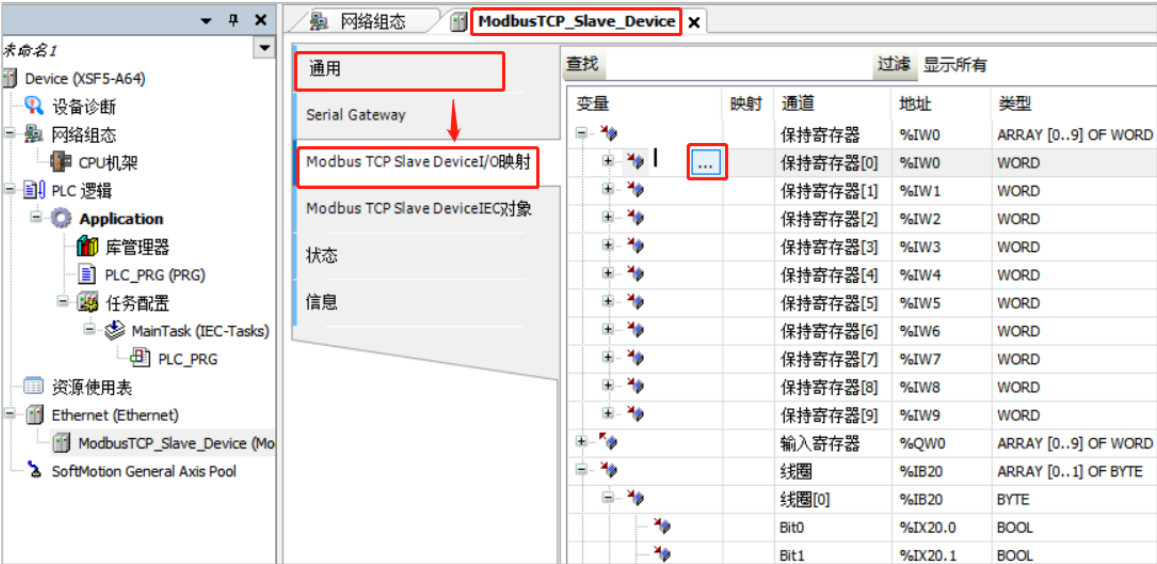


该界面可以配置 PLC 作为从站时可以通讯的地址数量。通讯的 WORD 字变量分为保持寄存器、输入寄存器，通讯的 BOOL 布尔变量分为线圈和离散输入。



- 上图“起始地址”是保持寄存器、输入寄存器、线圈、离散输入各自的第一个 WORD 或 BOOL 的 MODBUS 地址，地址依次排序，默认各自的 MODBUS 地址都是从 0 开始的。
- 上图“Writeable”默认不勾选，若勾选上则保持寄存器的地址类型变为%QW，PLC 作为服务器也可以通过 codesys 程序编程给保持寄存器强制赋值，但要注意 PLC 给值和其他客户端给值不要冲突。

“通用”界面配置好参数后再点击“I/O 映射”界面就可以看到配置的通讯地址数量。程序里定义的变量可映射在下图对应地址前的“ | ...”中，或者是使用 AT 地址来关联变量与通讯的地址。

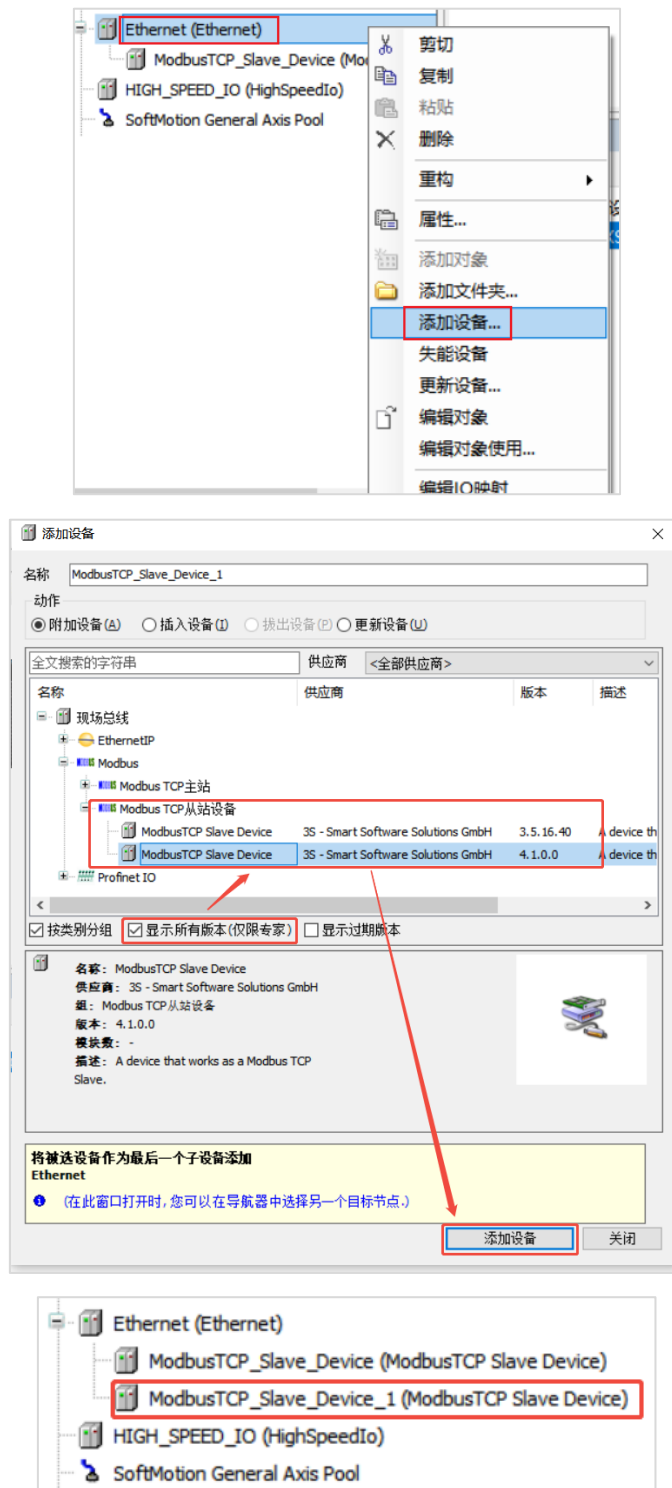


 注意:

- XS Studio2.3.2 以下时,软件里配置为 ModbusTCP 从站(Other)时仅支持添加一个从站设备。
- XS Studio2.3.2 及以上时,软件里配置为 ModbusTCP 从站(Other)时可支持添加多个从站设备。添加方式如下:

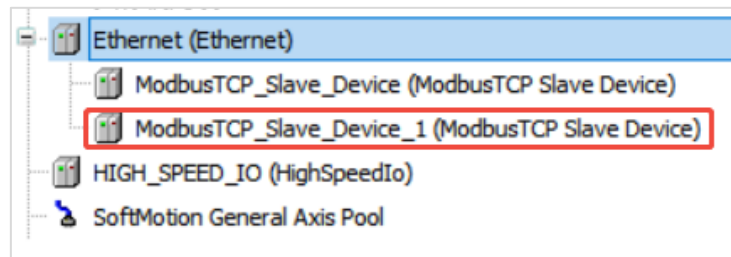
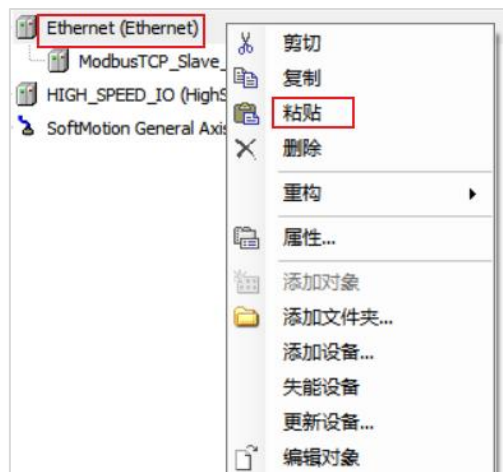
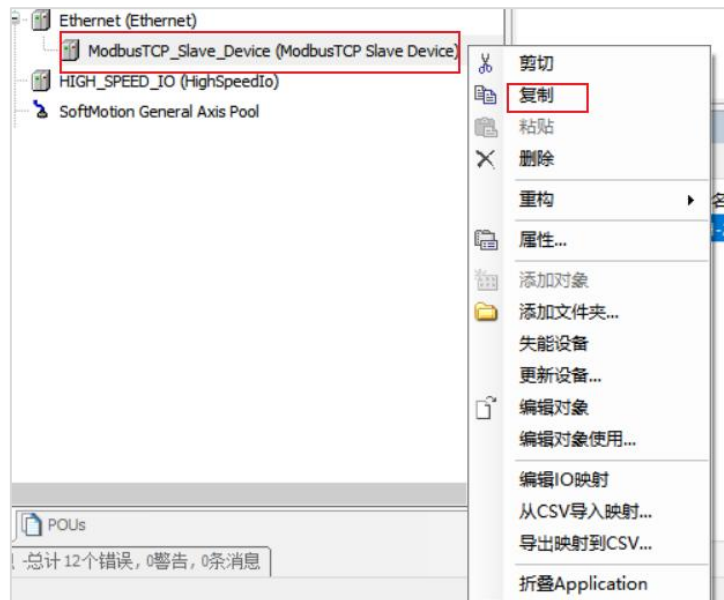
方法一:

选中“Ethernet”后点击鼠标右键,点击“添加设备”,在“添加设备”界面选择 ModbusTCP Slave Device 后添加即可。



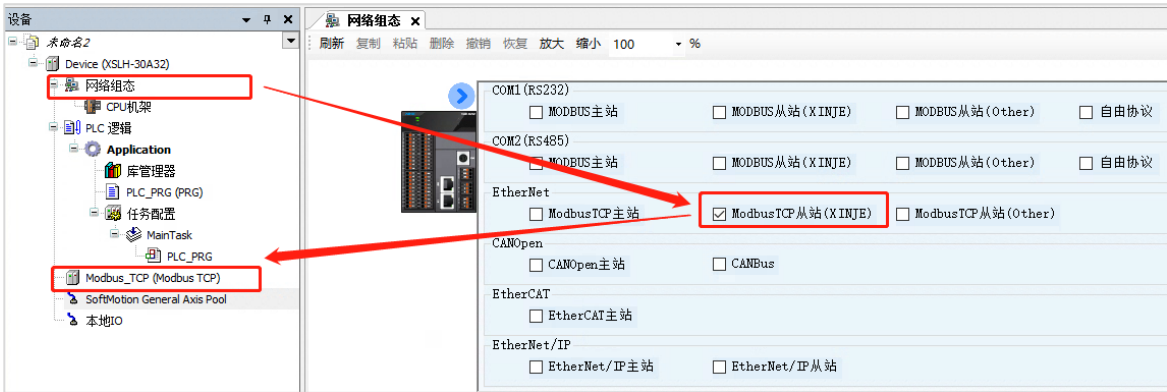
方法二:

选中“ModbusTCP_Slave_Device”后点击鼠标右键,点击“复制”,然后再选中“Ethernet”后点击鼠标右键,点击“粘贴”即可。

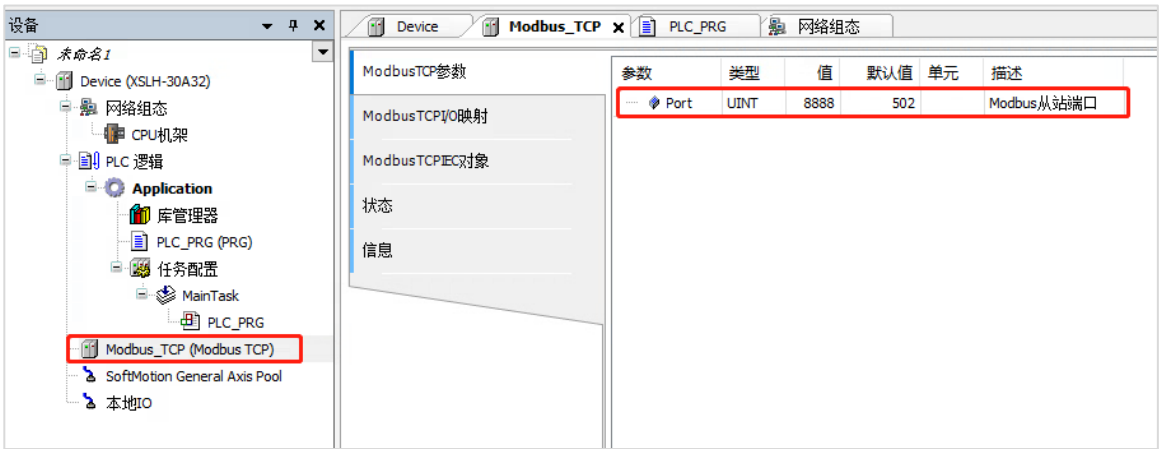


3-4-3. MODBUS TCP 从站 (XINJE) 配置

- 1、从左侧设备树双击“网络组态”节点，打开网络组态界面，通过使能窗口使能 ModbusTCP 从站(XINJE)设备，同时会在左侧设备树内生成“Modbus_TCP(Modbus TCP)”节点。具体如下图：



2、 左侧设备树内双击“Modbus_TCP(Modbus TCP)”节点打开 ModbusTCP 参数配置界面。可根据实际需求设置 ModbusTCP 端口。此处按照实际需求设置为：8888，如下图所示：



注意：

- 允许用户对端口号进行配置，端口号配置范围：1-65535，默认端口号 502。
- PLC 做 Modbus TCP(XINJE)从站设备时，可以被主站设备访问的地址范围定义如下：
 - ◆ 所有的线圈操作（功能码 0x01、0x02、0x05、0x0F）可读写地址为：%MB0-%MB65534；
 - ◆ 所有的寄存器操作（功能码 0x03、0x04、0x06、0x10）可读写地址为：%MW40000-%MW105534。
- XS 机型不同固件版本与不同 Modbus TCP（XINJE）库版本可支持的掉电保持属性如下方表格所示。

Modbus TCP (XINJE) 库版本 固件版本 (XS)	Modbus TCP (XINJE) 库版本		
	V1.0.0.0	V2.0.0.0	V3.0.0.0
V1.0.2a	M 区固定掉电保持	M 区固定掉电保持	M 区固定掉电不保持
V1.1.0	M 区固定掉电保持	M 区固定掉电不保持	M 区固定掉电不保持
V2.2.0 及以上	M 区固定掉电保持	M 区可配掉电保持	M 区可配掉电保持

- XSF5 机型不同固件版本与不同 Modbus TCP（XINJE）库版本可支持的掉电保持属性如下方表格所示。

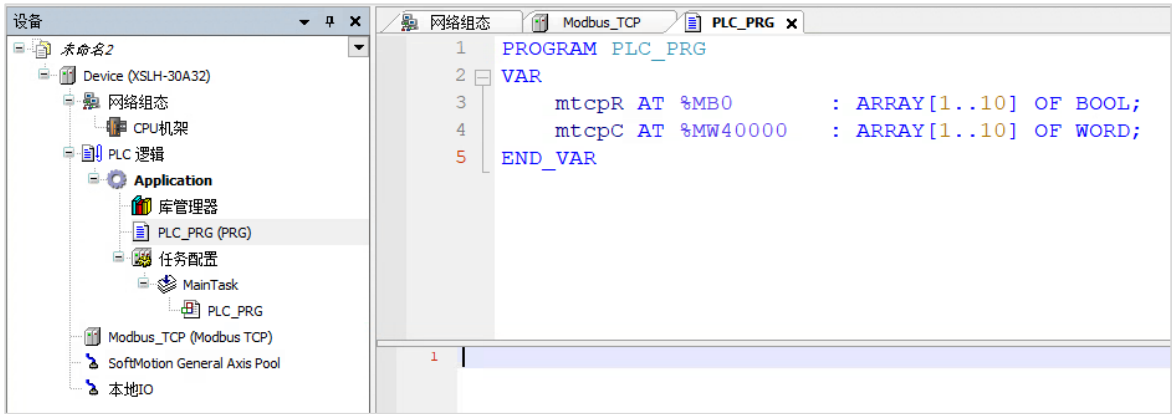
Modbus TCP (XINJE) 库版本	V1.0.0.0	V2.0.0.0	V3.0.0.0
	固件版本 (XSF5)		
V2.3.0 以下	M 区固定掉电保持	M 区固定掉电不保持	M 区固定掉电不保持
V2.3.0 及以上	M 区固定掉电保持	M 区可配掉电保持	M 区可配掉电保持

XSA 系列机型只支持 V3.0.0.0 的 Modbus TCP (XINJE) 库版本, 不支持 V1.0.0.0 和 V2.0.0.0 库版本。XSA 固件为 2.2.0 及以上支持在“Device” - “属性” - “目标内存设置”里修改 M 区掉电保持范围, 默认不保持、可配置保持地址范围。

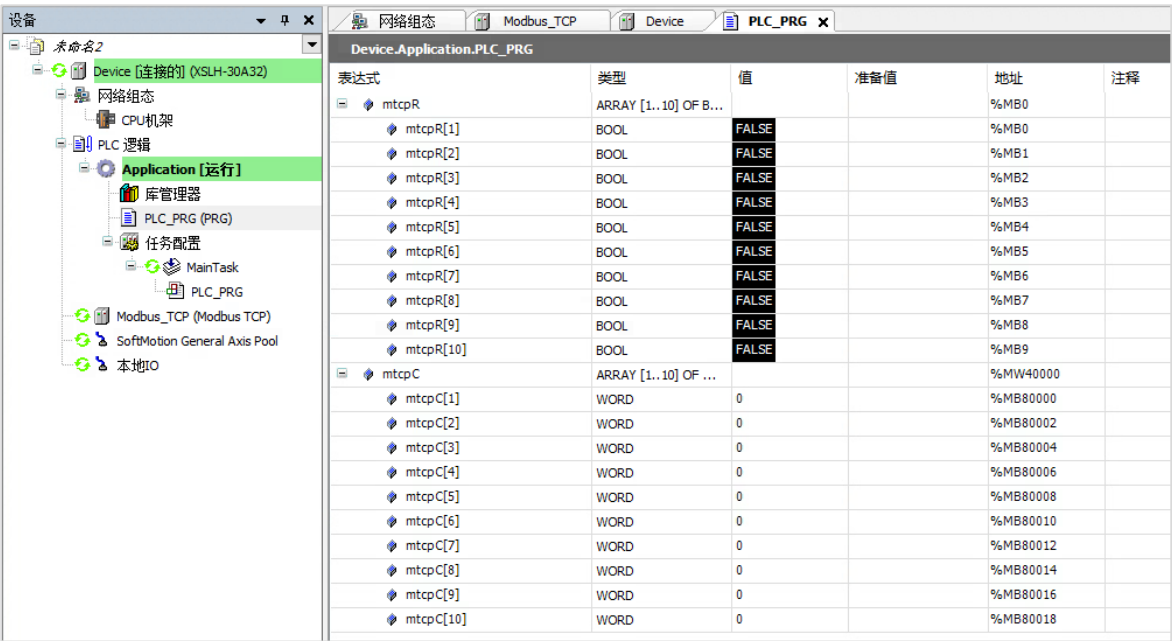
 应用举例:

示例一: 此处 XS Studio 软件作为从站, 使用第三方调试工具 Modbus Poll 作为主站建立连接并进行 Modbus TCP 通讯, 实现接收或发送寄存器或线圈数据。

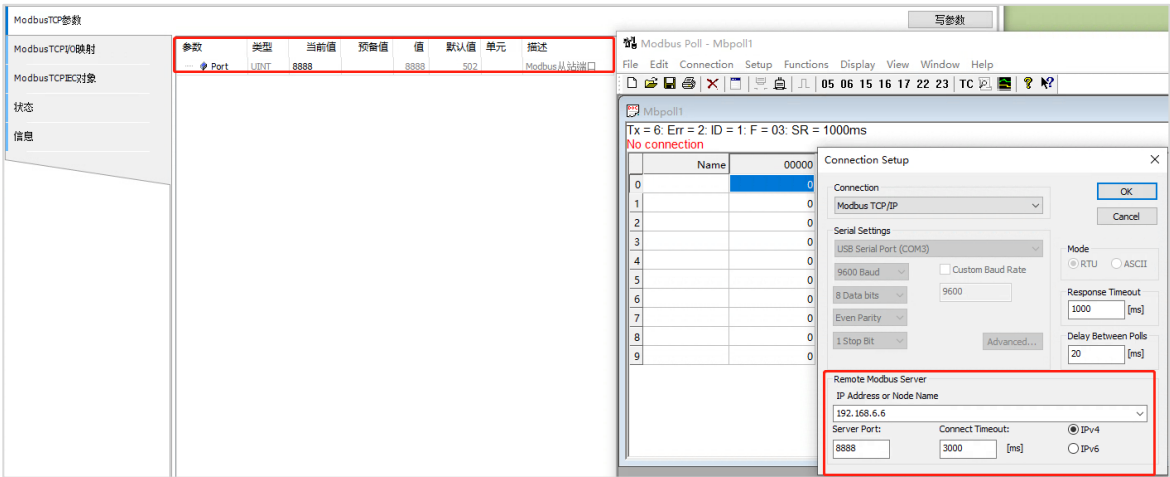
1) 在“PLC-PRG”编辑器内申明两个变量, 分别接收与发送寄存器或线圈数据。如下图:



2) 与 XSLH-30A32 设备建立连接, 并登录后运行。如下图:

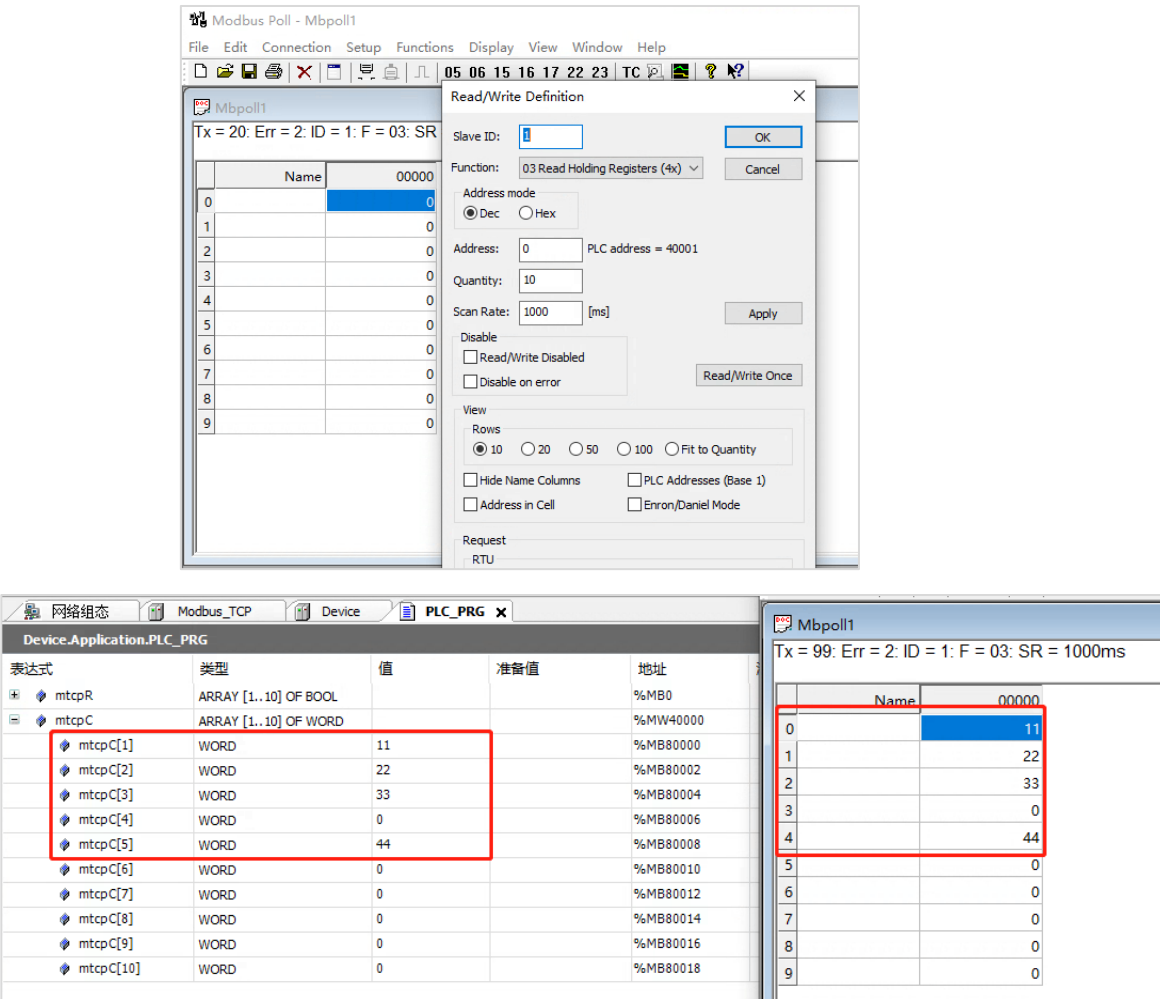


3) 配置 Modbus Poll 的相关参数，与从站设备的 IP 保持一致，与软件内的 Port 端口号保持一致，否则无法成功建立连接。如下图所示：

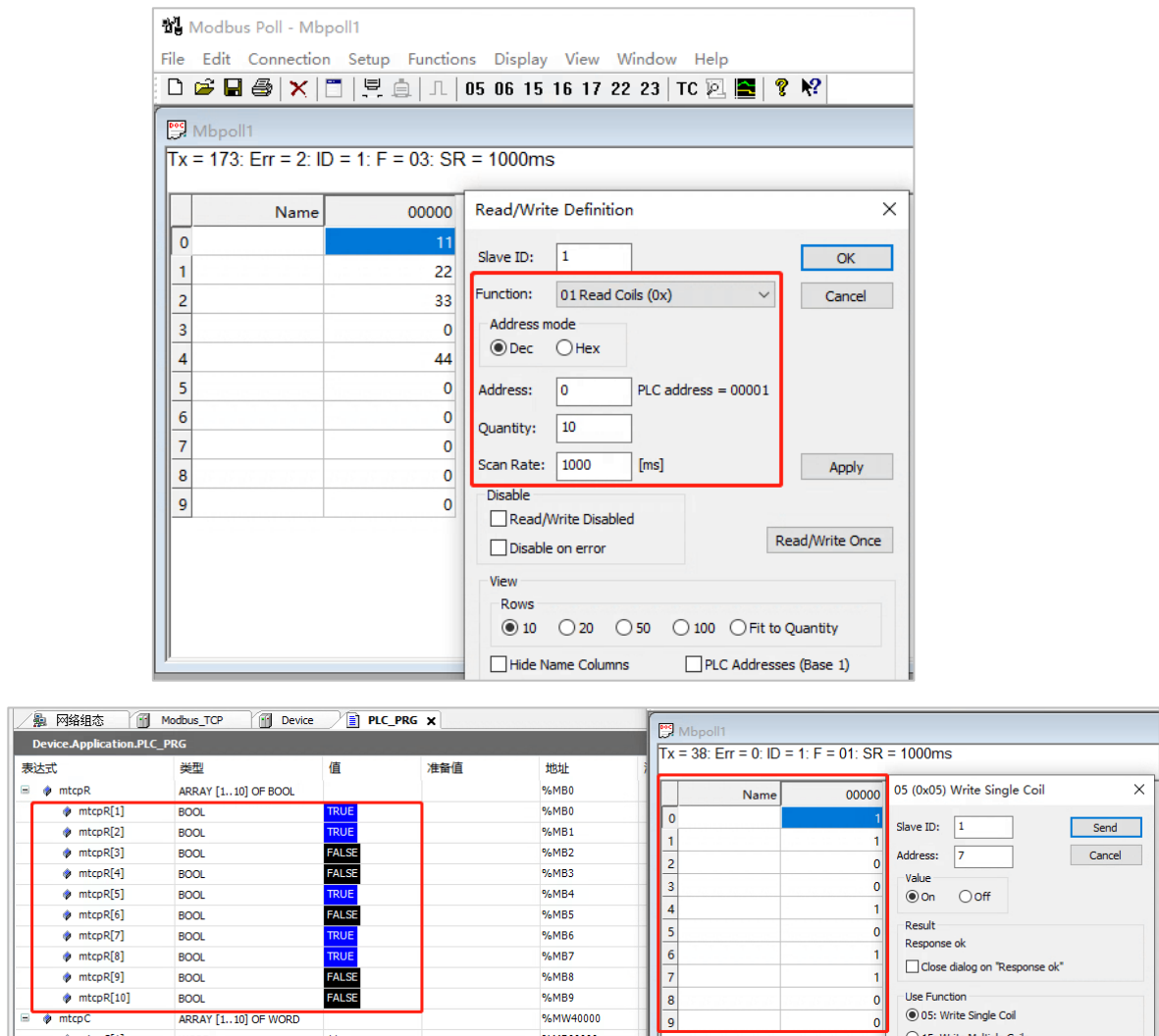


4) 配置主站设备的相关读写参数，即可与从站进行相关的读/写寄存器或线圈的操作：

- ◆ 读/写寄存器操作。如下图所示：



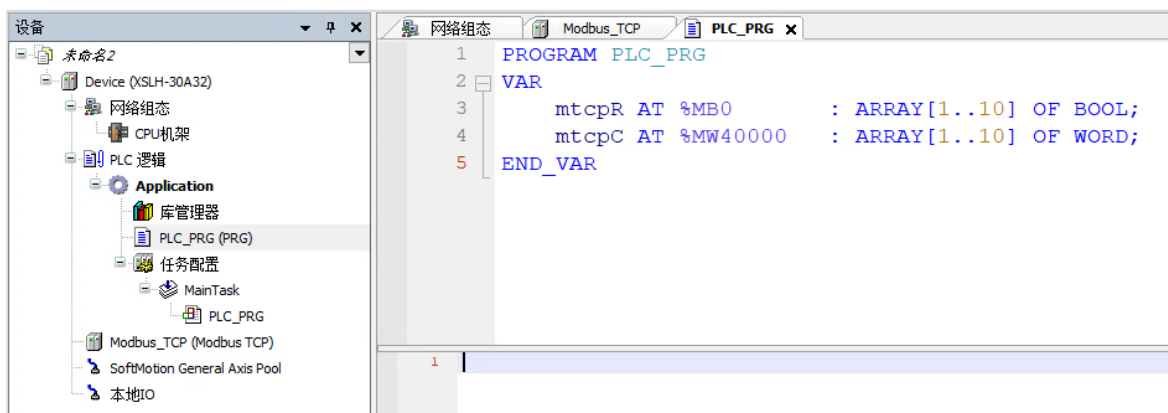
- ◆ 读/写线圈操作。如下图所示：



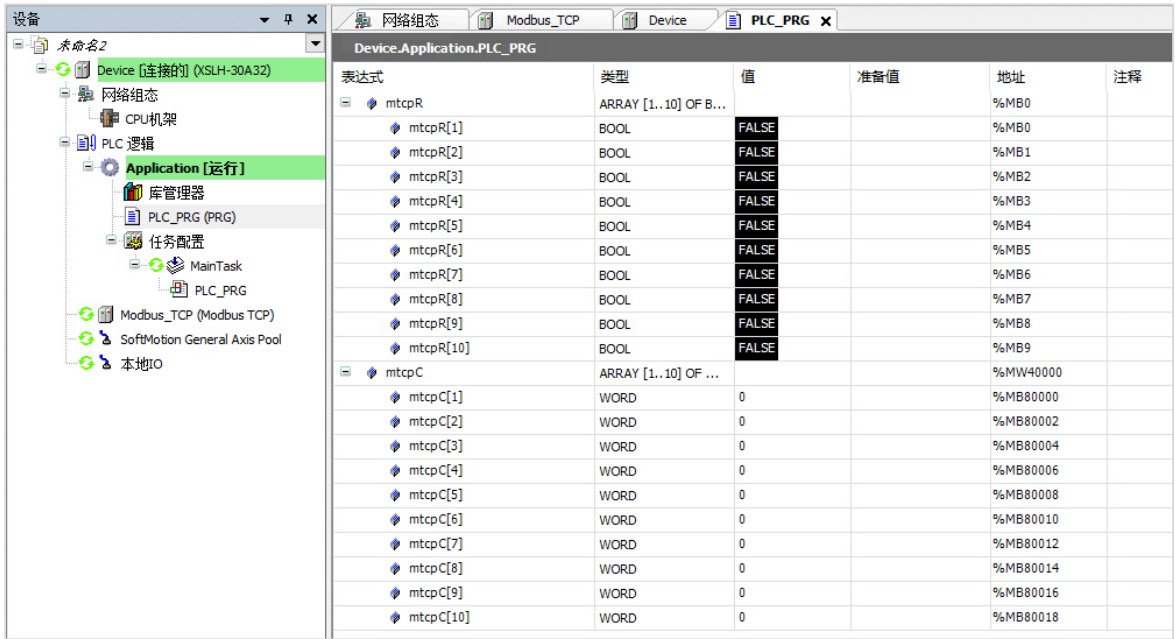
此时主站与从站已成功通信。

示例二：此处 XS Studio 软件作为从站，触摸屏作为主站，与触摸屏建立连接并进行 Modbus TCP 通讯，实现数据交互。

- 1) 在“PLC-PRG”编辑器内申明两个变量，分别接收与发送寄存器或线圈数据。如下图：

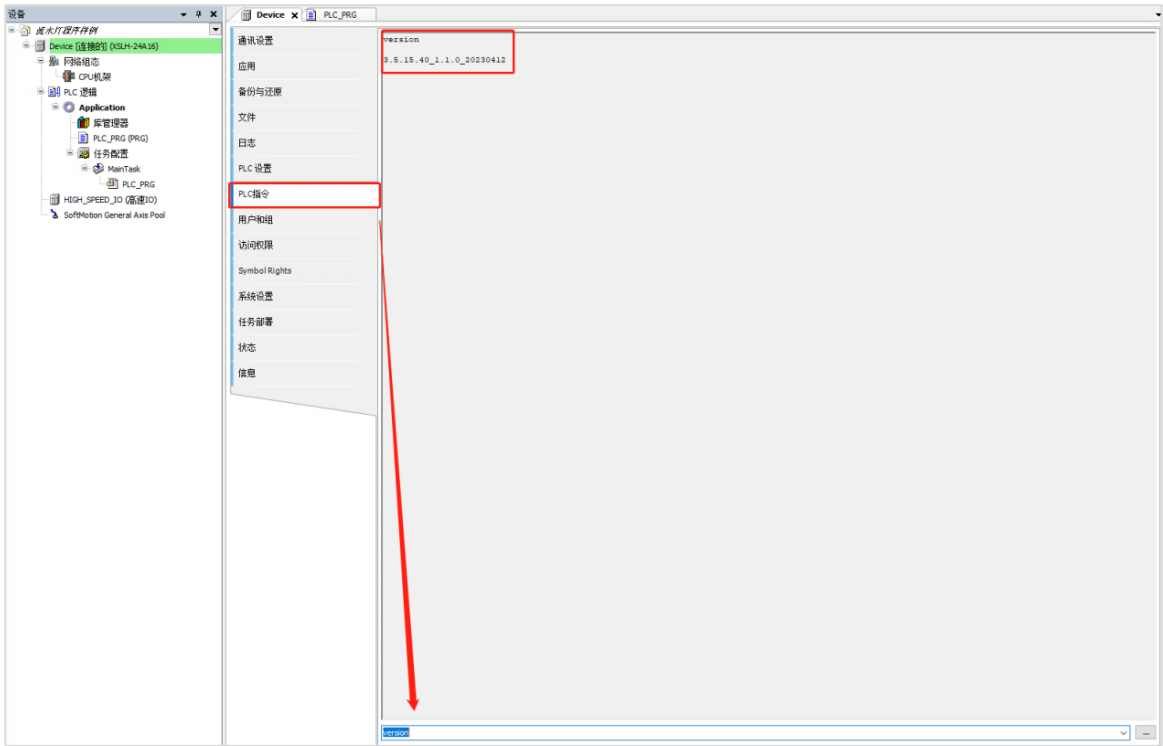


2) 与 XSLH-30A32 设备建立连接，并登录后运行。如下图：



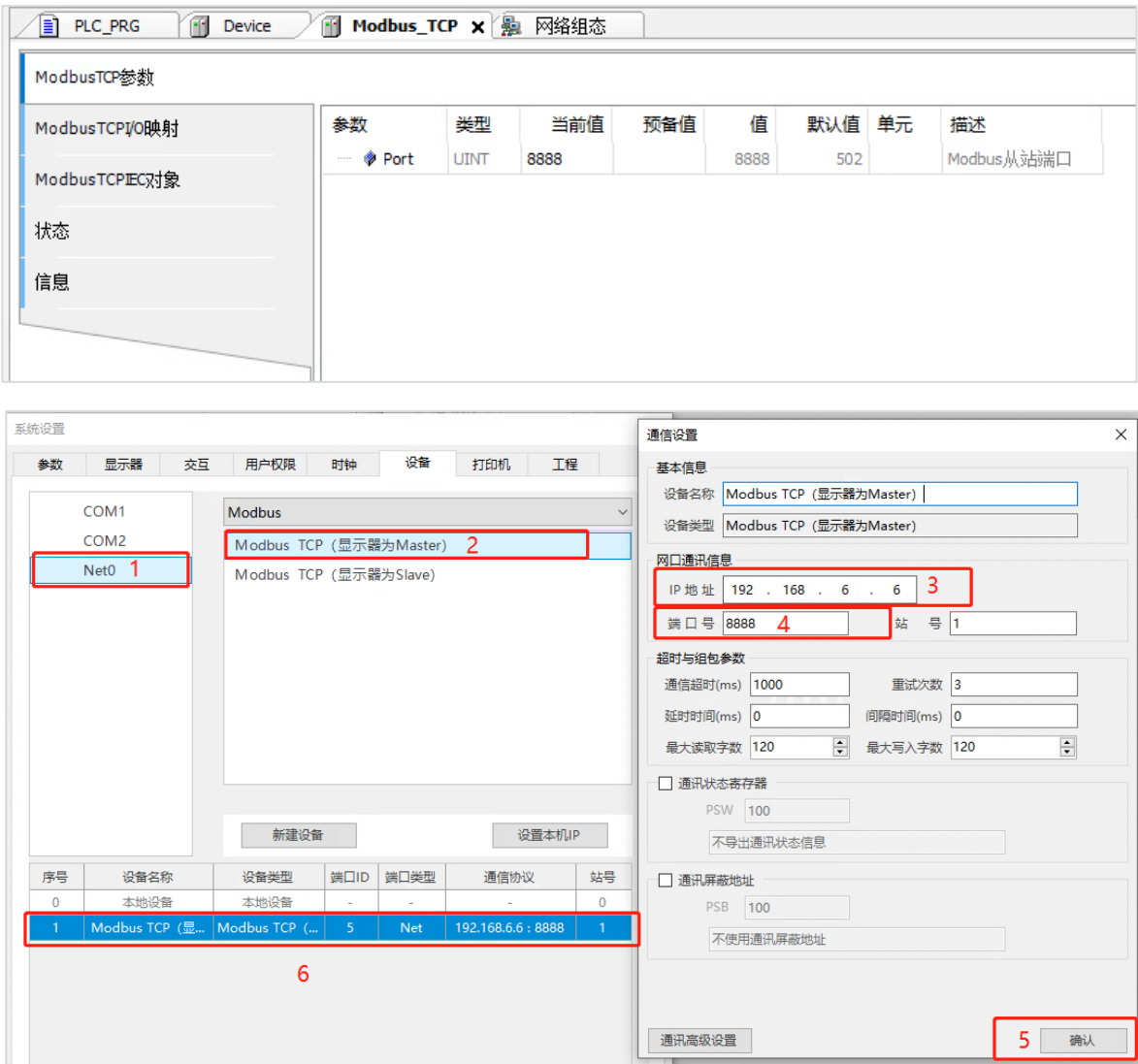
3) 配置触摸屏相关参数。

在使用此通讯功能时，请先检查 PLC 的固件版本是否为 3.5.15.40_1.0.0_P1_20211028 及其以上，如果不是此版本，请先升级固件。

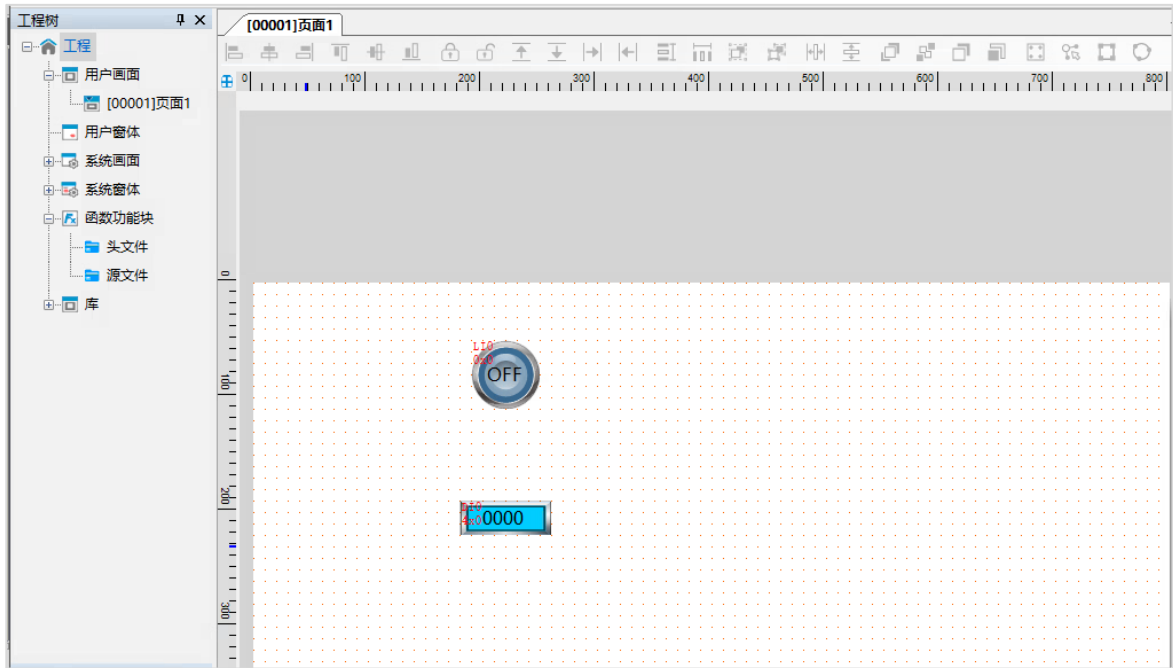


4) 设置触摸屏的通信参数。

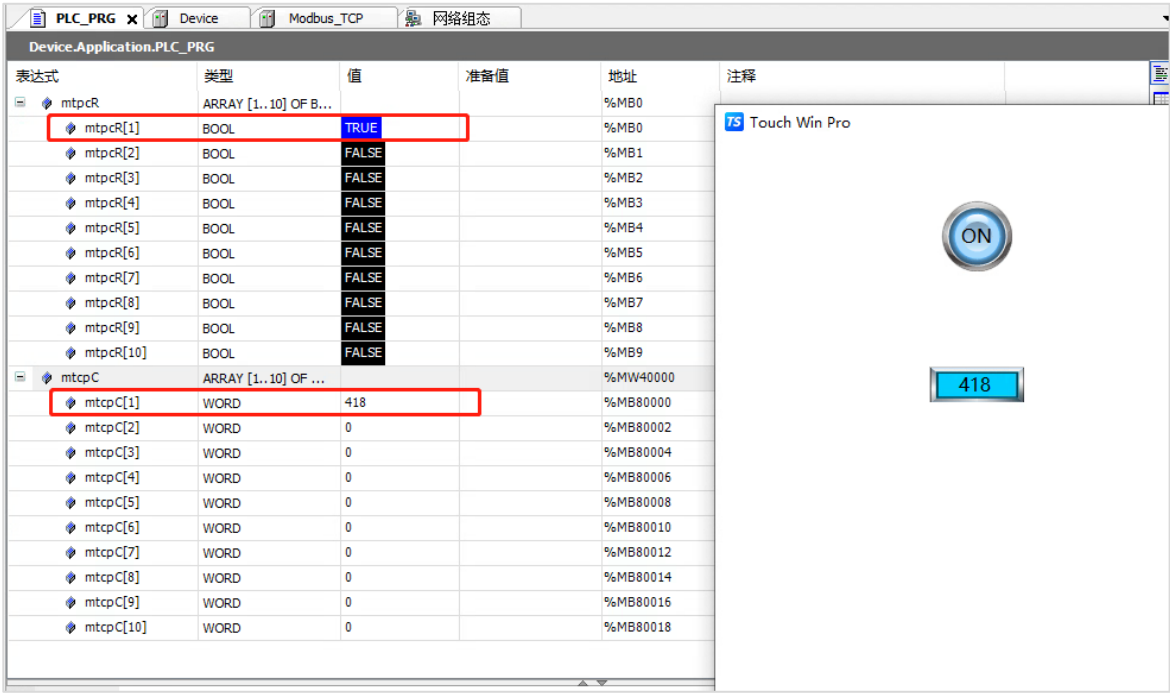
从“系统设置”窗口打开“设备”功能界面，选择 Net0 -> Modbus TCP(显示器 Master)。此处 IP 地址需要与从站设备的 IP 保持一致，与软件内的 Port 端口号保持一致，否则无法成功建立连接。如下图所示：



5) 在工程界面内添加相关部件，如下图：



6) 在线模拟，与 XS Studio 软件建立连接即可与从站进行相关的读/写寄存器或线圈的操作。如下图所示：



- 添加所需要的触摸屏元素，设备选择 Modbus_通用，站点号必须设为 0！
- 按钮或者指示灯的对象类型选择 0X（可读可写）或者 1X（只读），0X0 和 1X0 都对应 MB0，然后依次类推；
- 数据输入或者数据显示的对象类型选择 3X（只读）或者 4X（可读可写），3X0 和 4X0 都对应%MW40000，然后依次类推；
- 如果数据类型输入或者显示的是 DWORD，那么 3X0 和 4X0 占用%MW40000、%MW40001 这两个寄存器,然后依次类推。

3-4-4. MODBUS TCP 常见故障

故障现象	处理方式
主站无法对作为从站的信捷 XS 控制器进行读写	主站方配置从站参数时，站号要填写 0。
信捷 XS 控制器作为主站，与从站通讯不上	(1) 为客户端去访问这个地址，首先得保证服务器里有这个地址，不然客户端无法访问不存在的地址，会报错； (2) 查看从站是否有这些访问的数据类型、首地址及通讯个数； (3) 需注意一下功能码，功能码对不上，也不能通讯上。

3-4-5. MODBUS TCP 通讯帧格式说明

Modbus 设备可分为主站 (poll) 和从站 (slave)。主站只有一个，从站有多个，主站向各从站发送请求帧，从站给予响应。在使用 TCP 通信时，主站为 client 端，主动建立连接；从站为 server 端，等待连接。

- ◆ 主站请求：功能码+数据；
- ◆ 从站正常响应：请求功能码+响应数据；
- ◆ 从站异常响应：异常功能码+异常码，其中异常功能码即将请求功能码的最高有效位置 1，异常码指示差错类型。



需要超时管理机制，避免无期限的等待可能不出现的应答。

IANA (Internet Assigned Numbers Authority, 互联网编号分配管理机构) 给 Modbus 协议赋予 TCP 端口号为 502，这是目前在仪表与自动化行业中唯一分配到的端口号。

ModbusTCP 的数据帧可分为两部分：MBAP+PDU。

● 报文头 MBAP

MBAP 为报文头，长度为 7 字节，组成如下：

事务处理标识	协议标识	长度	单元标识符
2 字节	2 字节	2 字节	1 字节

事务处理标识	可以理解为报文的序列号，一般每次通信之后就要加 1 以区别不同的通信数据报文。
协议标识符	00 00 表示 ModbusTCP 协议。
长度	表示接下来的数据长度，单位为字节。
单元标识符	可以理解为设备地址。

● 帧结构 PDU

PDU 由功能码+数据组成。功能码为 1 字节，数据长度不定，由具体功能决定。

通信过程：

- 1、connect 建立 TCP 连接；
- 2、准备 Modbus 报文；
- 3、使用 send 命令发送报文；
- 4、在同一连接下等待应答；
- 5、使用 recv 命令读取报文，完成一次数据交换；
- 6、通信任务结束时，关闭 TCP 连接。

3-5. CAN 总线

CAN 是 Controller Area Network 的缩写（以下称为 CAN），是 ISO 国际标准化的串行通信协议。在北美和西欧，CAN 总线协议已经成为汽车计算机控制系统和嵌入式工业控制局域网的标准总线，并且拥有以 CAN 为底层协议专为大型货车和重工机械车辆设计的 J1939 协议。

3-5-1. 参数配置

CANBus 抽象含义是控制器局域网。实则是一条具有高低电平差位的双绞线。它起着传输数据的作用，基于它串行通信的实时性、可靠性、有效性，被广大工程师们青睐。最早是德国 Bosch 公司为汽车电子的应用而开发的，现已推广至机械制造、工业自动化、伺服电机制造、大型医疗机械、楼宇安防监控等领域。目前 CANBus 已经成为工业通信首选的现场总线。

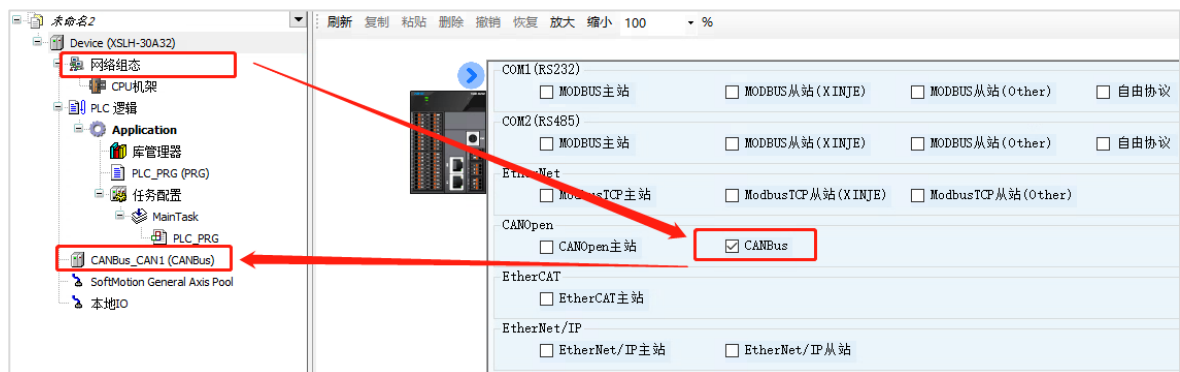
目前 XSLH-30A32 和 XSF5 机型支持 CAN 通讯，可以与 CAN 网络中的其他支持 CAN2.0B 或 CAN2.0A 协议设备进行自由格式的数据交换。



XSF5 固件 V2.3.0 及以上支持 CAN2.0B 即 CANBUS，之前版本仅支持 CAN2.0A 即 CANOPEN。

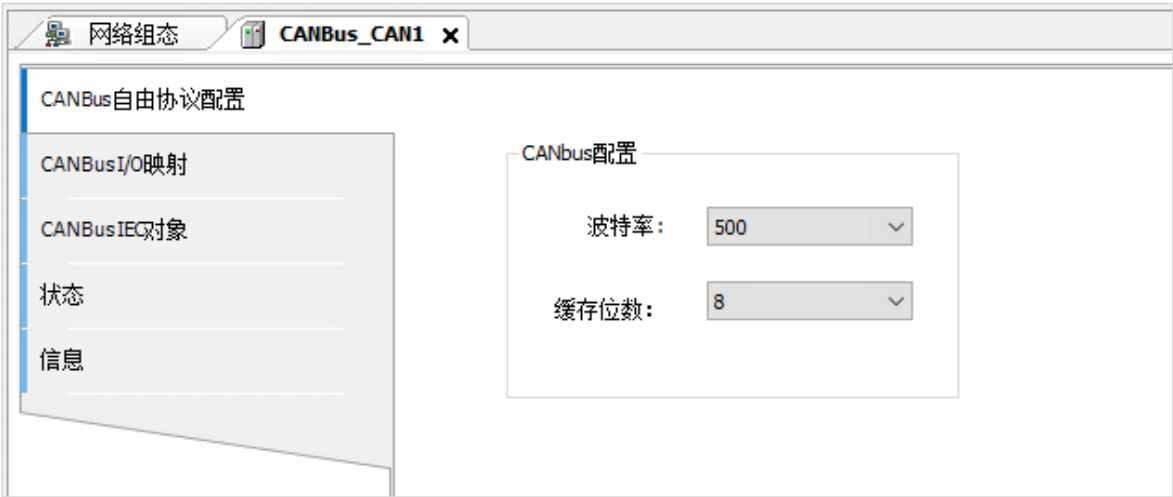
1、使能 CANBus 设备

左侧设备树内双击“网络组态”节点打开网络组态界面，通过使能窗口使能“CANBus”，在左侧设备上生成“CANBus_CAN1(CANBus)”节点。如下图所示。



2、配置 CANBus 参数

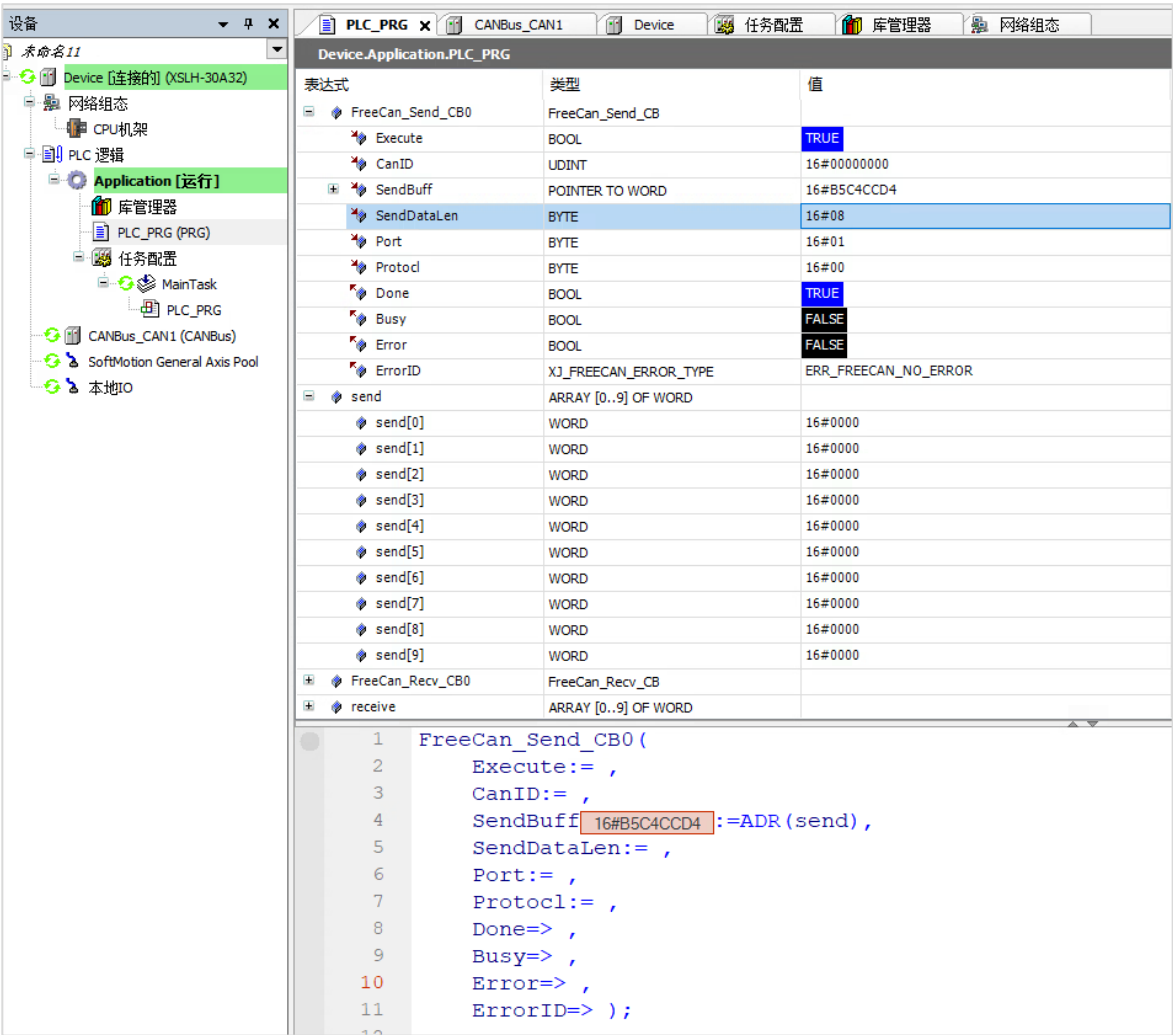
在左侧设备树双击“CANBus_CAN1(CANBus)”节点打开 CANBus 自由协议配置界面。可按照实际需求设置 CANBus 的波特率及缓存位数。如下图所示。



CANbus 参数配置如下：

波特率	通信时的速率。
缓存位数	缓存位数可设置为 8 位或 16 位，当缓存位数为 8 位时，只发送寄存器的低字节数据；当缓存位数为 16 位时，寄存器的高低字节数据都将被发送，低字节在前，高字节在后。

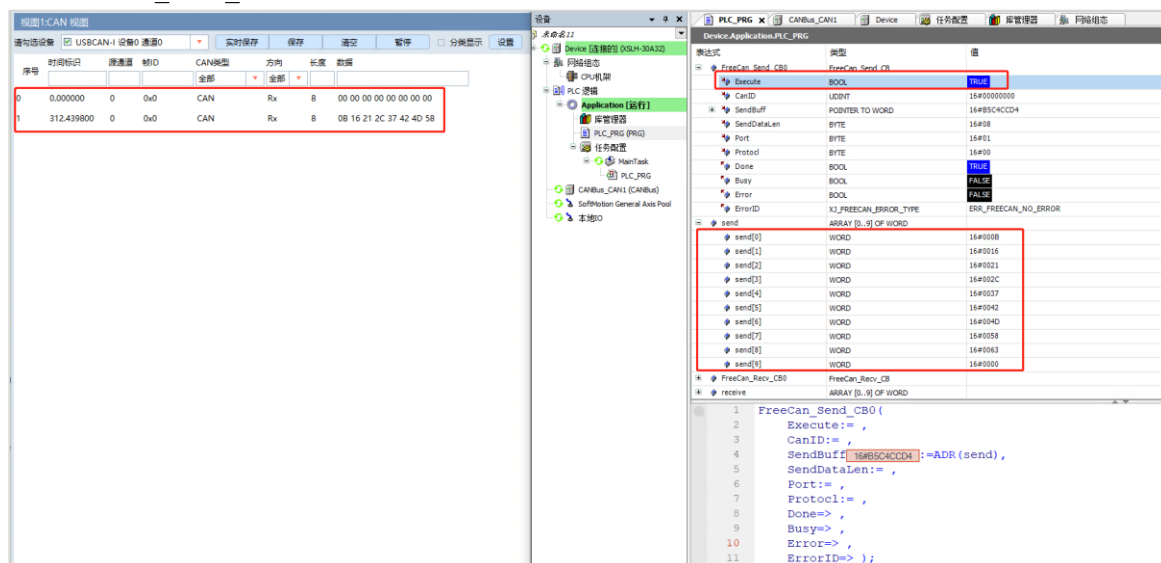
3、编写 CANBus 自由格式通讯指令并配置相关参数后与 PLC 设备建立连接。具体如下图：



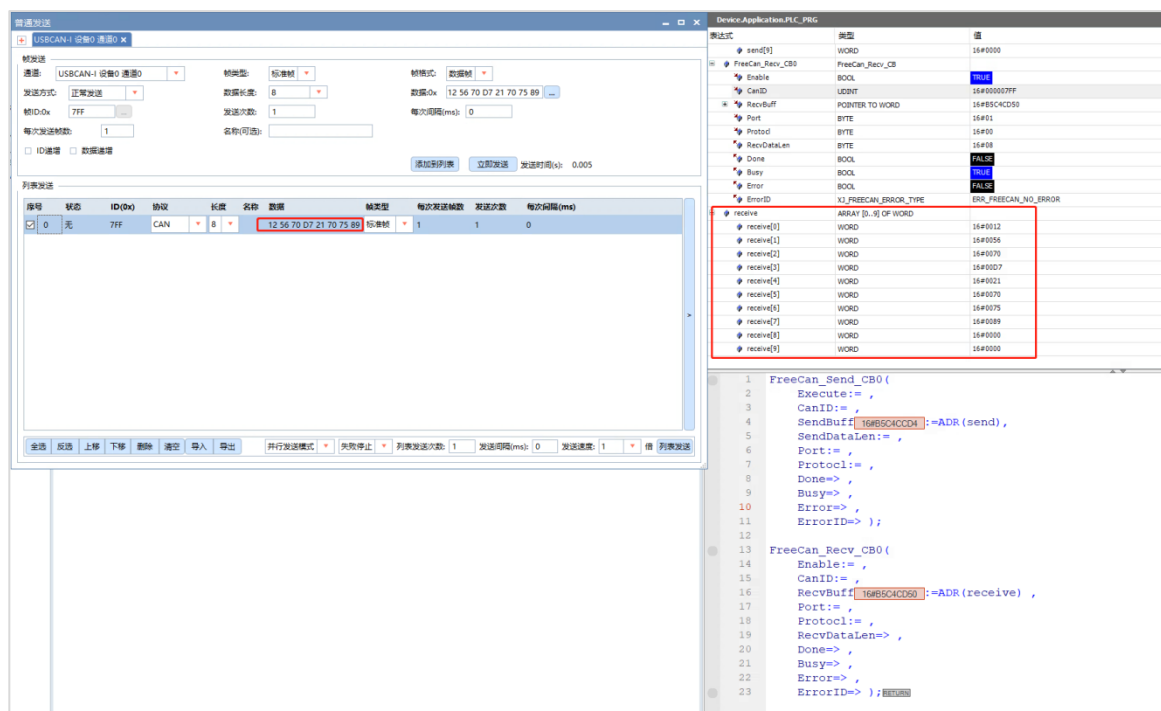
相关指令不在此手册描述，请参考《XS 系列 PLCopen 标准控制器用户手册【基本指令篇】》。

4、XS Studio 软件与第三方调试工具 ZCANPRO 进行数据交互。具体如下图所示：

通过 FreeCan_Send_CB 发送指令发送数据到调试工具；如下图：



调试工具 ZCANPRO 发送数据，通过接收指令 FreeCan_Recv_CB 接收数据并显示；如下图：

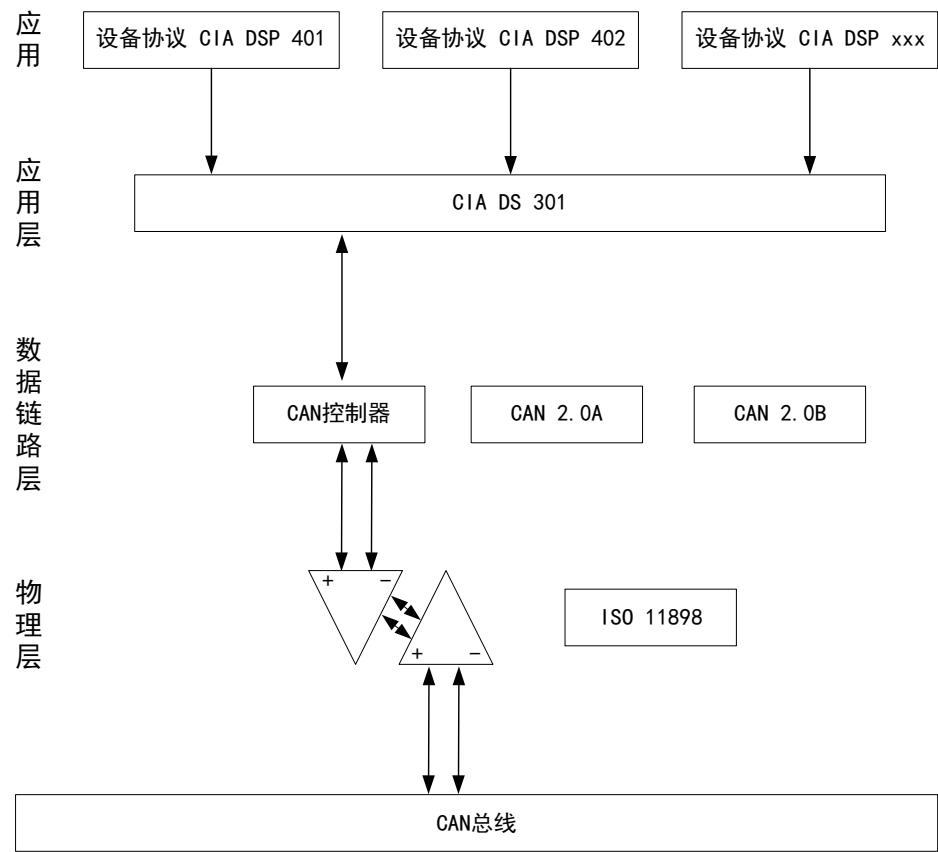


3-5-2. CANopen 网络

CANopen 协议是在 20 世纪 90 年代末，由总部位于德国纽伦堡的 CiA 组织(CAN-in-Automation) 在 CAL (CAN Application Layer) 的基础上发展而来。

CANopen 是一个基于 CAN 串行总线的网络传输系统的应用层协议，遵循 ISO/OSI 标准模型。其中基础协议是 CANopen 应用层协议 (CANopen Application Layer and Communication Profile) (DS 301)，规定了 CANopen 协议层及通信结构描述。在基础协议之上，各个行业具备设备子协议。所谓子协议，就是针对不同行业的应用对象，对 CANopen 内部的数据含义进行重新定义或添加新的控制逻辑。

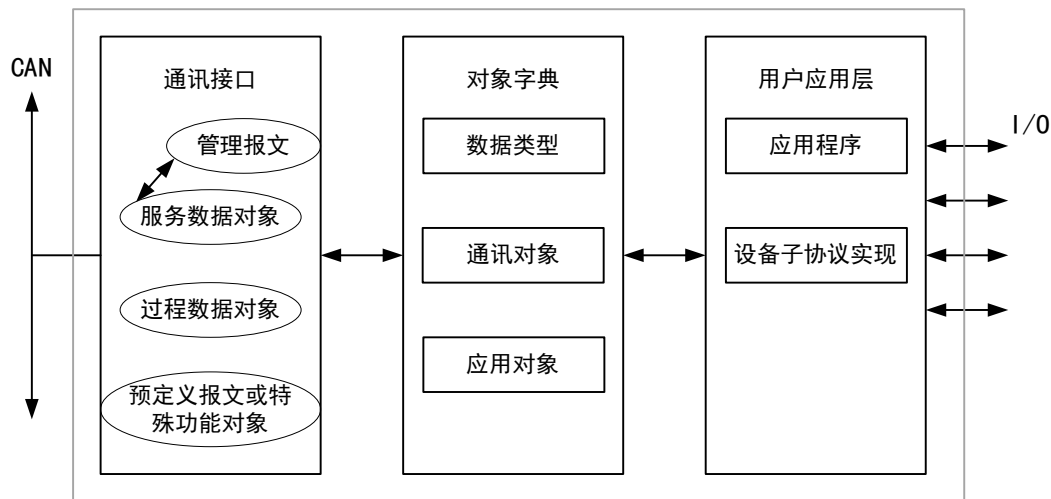
在 OSI 模型中，CAN 标准、CANopen 协议之间的关系如下图所示：



OSI 模型是一个概念模型，用于标准化各种通信技术之间的通信功能。较低的层描述基本的通信（例如原始比特流），而较高的层描述诸如长消息的分段之类的东西以及诸如消息的发起、指示、响应和确认之类的服务。

CANopen 协议通常分为用户应用层、对象字典以及通信三个部分。其中最核心的是对象字典，CANopen 通讯通过对象字典（OD）能够访问驱动器的所有参数。

CANopen 设备结构如下图所示：



1、通信对象

CANopen 协议中常用的通信对象包含如下几点：

1) 网络管理对象 (NMT)

网络管理对象包括 Boot-up 消息, Heartbeat 协议及 NMT 消息, 基于主从通信模式, NMT 用于管理和监控网络中的各个节点, 主要实现三种功能: 节点状态控制、错误控制和节点启动。

2) 服务数据对象 (SDO)

服务数据对象主要用于主节点对从节点的参数配置。服务确认是 SDO 最大的特点, 为每个消息都生成一个应答, 以确保数据传输的准确性。在一个 CANopen 系统中, 通常 CANopen 从节点作为 SDO 服务器, CANopen 主节点作为客户端。客户端通过索引和子索引能够访问数据服务器上的对象字典, 所以 CANopen 主节点可以访问从节点的任意对象字典项的参数, 并且 SDO 可以传输任何长度的数据 (当数据长度超过 4 字节时, 拆分成多个报文来传输)。

3) 过程数据对象 (PDO)

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 1~8 个字节。每个 CANopen 设备包含 8 个缺省的 PDO 通道, 4 个发送 PDO 通道和 4 个接收 PDO 通道。PDO 包含同步和异步两种传输方式, 由该 PDO 对应的通信参数决定。

4) 同步对象 (SYNC)

同步对象是由 CANopen 主站周期性地广播到 CAN 总线的报文, 用来实现基本的网络时钟信号, 每个设备可以根据自己的配置, 决定是否使用该事件来跟其它网络设备进行同步通信。

5) 紧急报文 (EMCY)

设备内部通信故障或者应用故障错误时发送的报文。

2、对象字典

CANopen 对象字典 (Object Dictionary, OD) 是 CANopen 协议最为核心的概念。所谓的“对象字典”, 就是一个有序的对象组; 每个对象采用一个 16 位的索引值来寻址。为了访问数据结构中的元素, 同时定义了一个 8 位的子索引。

CANopen 网络中每个节点都有一个对象字典。对象字典包含了描述这个设备和它的网络行为的所有参数。

CANopen 对象字典中的项由一系列子协议来描述。子协议描述对象字典中每个对象的功能、名字、索引、子索引、数据类型、读/写属性, 以及这个对象是否必需, 从而保证不同厂商的同类型设备兼容。

CANopen 协议的核心描述子协议是 DS301, 包括 CANopen 协议应用层及通信结构描述, 其他子协议都是对 DS301 协议描述文本的补充与扩展。

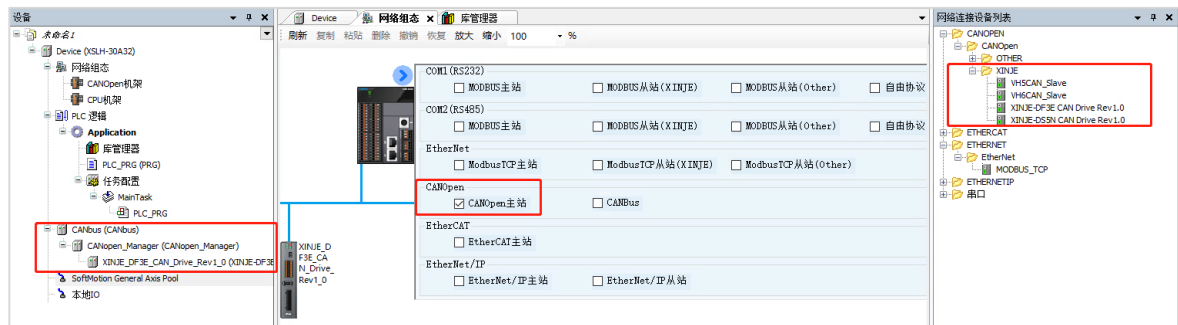
3-5-3. CANopen 主站配置

1、硬件端口

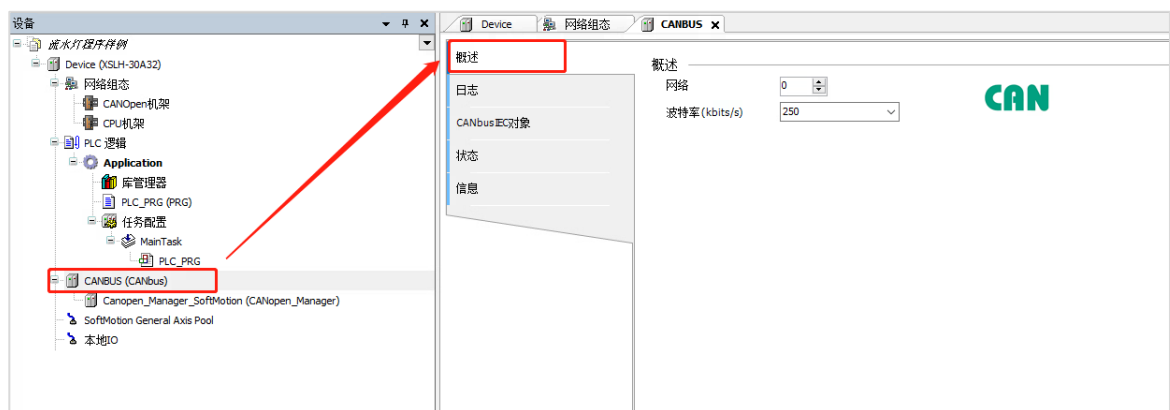
设备接入 CAN 总线时, 需要将 CAN+ 连接 CAN+, CAN- 连接 CAN-, 若从站为伺服, 需要将网线一端的第一根 (TX+) 和第二根 (TX-) 线分别接到 CAN+ 和 CAN- 上, 另外一端插入到伺服的网口。同时, PLC 上的拨码 3 和 4 为自带的终端电阻, 需要将其置 ON, 为了增强 CAN 通讯的可靠性, 消除 CAN 总线终端信号反射干扰, CAN 总线网络最远的两个端点通常要加入终端电阻, 若其它 CANopen 设备未自带终端电阻, 需用户自行安装。

2、软件配置

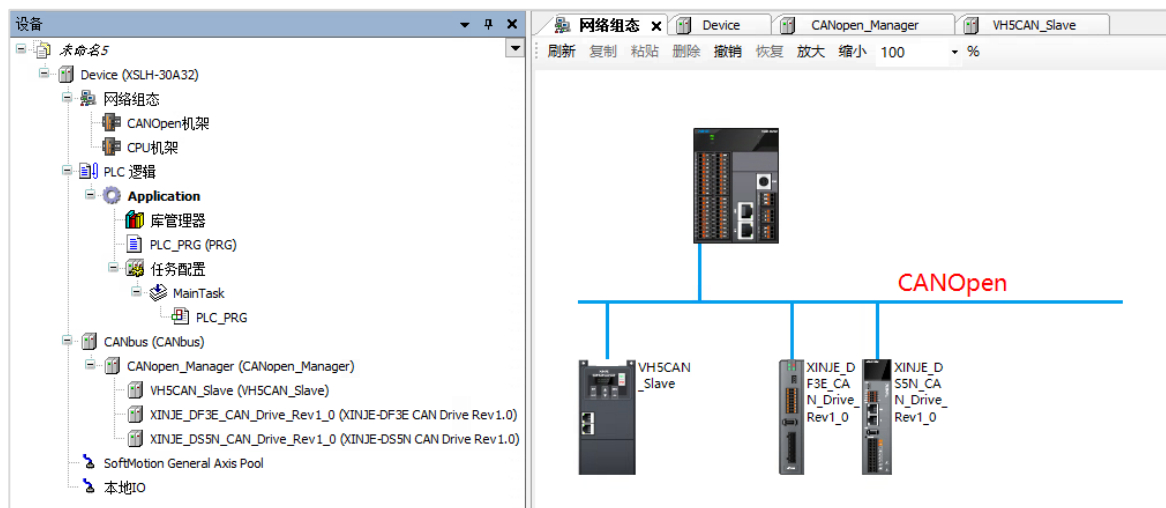
1) 在网络组态中激活 CANopen 总线。激活 CANopen 总线后，会自动添加 CANopen 主站。



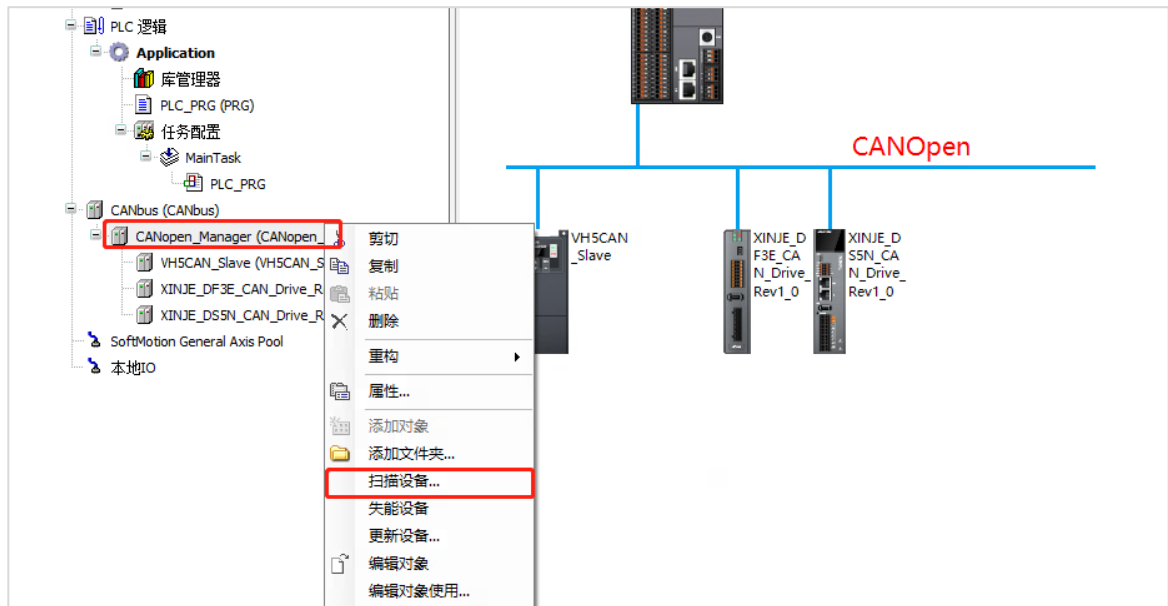
2) 添加成功后可看到设备栏下的“CANBus”，双击“CANBus”，在界面“概述”的波特率设置成和从站一致。



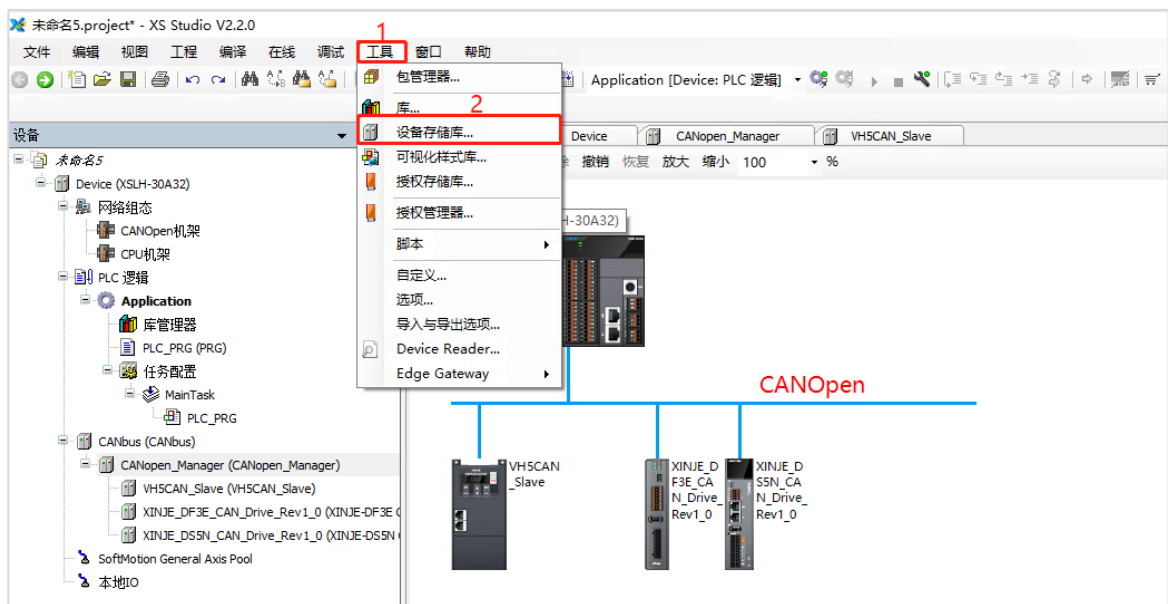
通过右侧“网络设备连接列表”可添加 CANopen 从站模块，并在界面左侧视图中将出现组态配置对应设备树。如图所示：

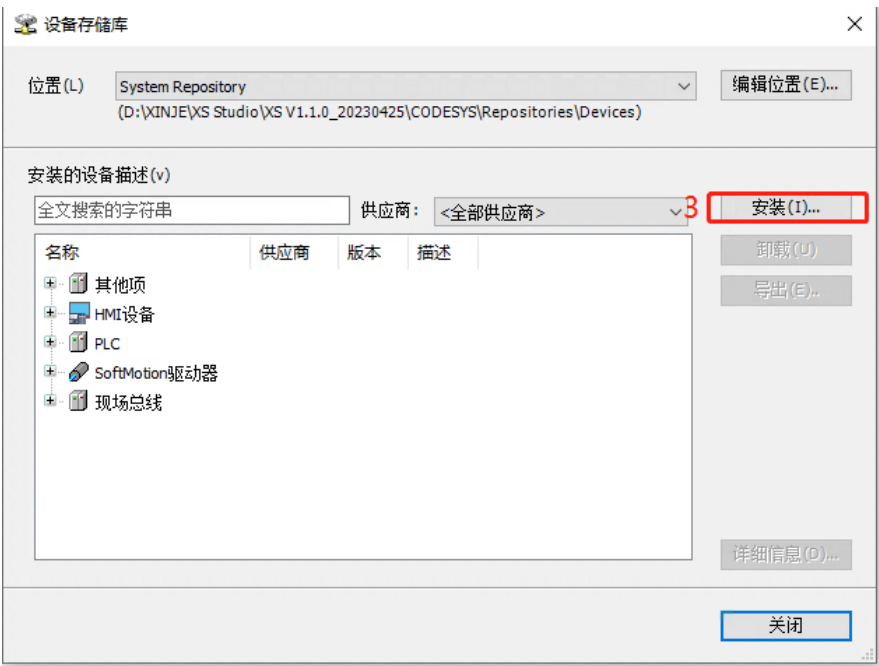


3) 添加“CANopen Manager SoftMotion”后，需要先下载程序，下载后右击“CANopen_Manager_SoftMotion”扫描设备，扫描成功后复制到工程中。

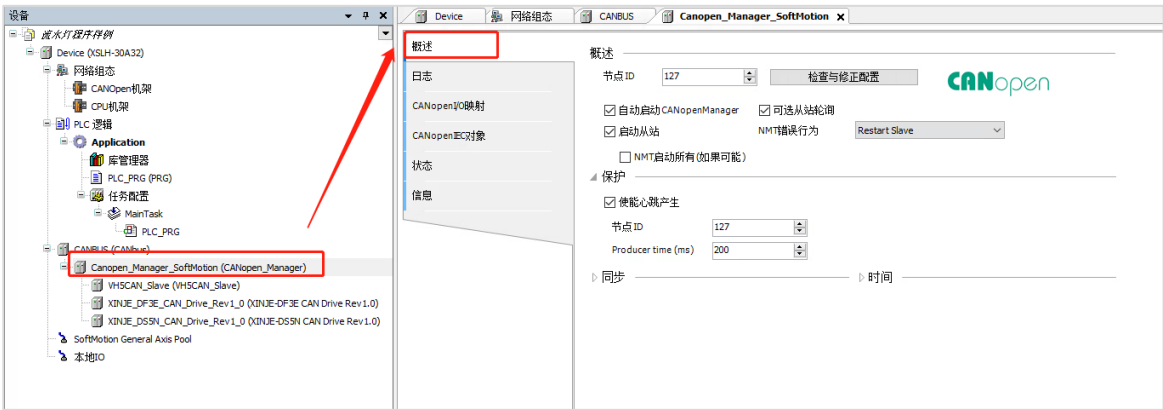


若没有扫描成功, 可检查是否导入 EDS 文件, 在工具-设备存储库-安装, 将从站的 EDS 文件导入进来。扫描进来后, 从站的节点 ID 会自动识别。若是手动添加从站设备, 需要手动修改从站 ID。





4) 在“CANopen_Manager_SoftMotion”中，需要对 CANopen 主站进行设置。



节点 ID	主站在 CANopen 网络唯一标识号，默认 127，范围 1~127。
检查与修正配置	点击进入后，若有错误可点击“自动修复”。
自动启动 CANopenManager（默认开启）	勾选后在所有从站准备就绪后 CANopen 管理器自动重启。
可选从站轮询 （默认开启）	当从站在引导序列中没有响应时，CANopen 管理器每秒询问一次它，直到它做出响应。不断轮询从站会增加总线周期时间，这会干扰应用（尤其是运动应用）。您可以停用轮询以避免此行为。如果轮询被停用，则在发送启动消息时会再次检测到从属服务器。
启动从站（默认开启）	CANopen 管理器负责启动从站。
NMT 启动所有 （如果可能）	如果激活了启动从站选项（默认：禁用），则 CANopen 管理器使用“NMT 全部启动”命令启动所有从站。只要可选从站尚未准备好启动，就不会执行“NMT 全部启动”命令。在这种情况下，CANopen 管理器单独启动每个从站。“NMT 全部启动”命令只能在没有可选从属设备的项目中得到保证。
NMT 错误行为	Restart Slave-如果在从机监控期间发生错误（NMT 错误事件），则堆栈会自动重新启动从站（NMT 复位+ SDO 配置+ NMT 启动）；Stop Slave-如果在从站监控期间发生错误（NMT 错误事件），则从站将停止。然后，您必须使用 CiA405 NMT 功能块从应用程序重置从机。

● 保护

使能心跳产生	如果启用这个选项（默认：禁用），主站将发送心跳信息。
节点 ID	发送心跳信息的唯一标识符，默认为主站节点 ID，范围 1~127（十进制）。
生产时间 (ms)	心跳信息发送的时间间隔，单位为毫秒，范围为 1ms~65535ms，并且是总线任务时间的整数倍。

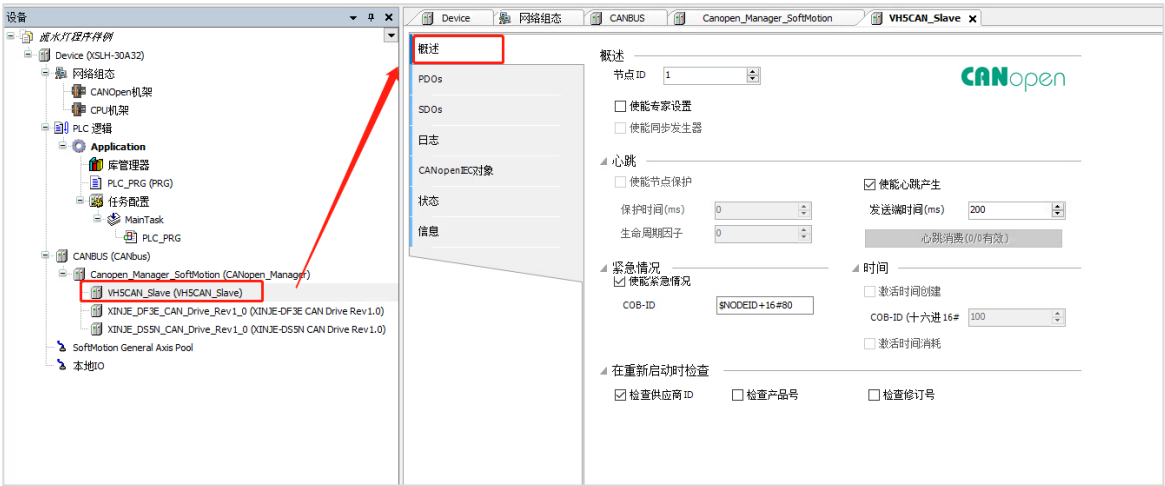
● 同步

启动同步生成	如果启用这个选项（默认：禁用），主站将发送同步信息。一个 CANopen 总线系统只能有一个站启用同步生产。同步类型 PDO 在同步信息发送后根据设置类型发送信息。
COB-ID	通信对象标识，此设置用于标识同步消息 ID。值不能修改，为 16#80。
循环周期 (us)	同步信息以同步周期定义的时间间隔发送，同步周期的单位为微秒，范围为 100-4294967295us，并且是总线任务时间的整数倍。
窗口长度 (us)	用于同步 PDO，以微秒为单位的时间窗长度。
启用同步消费	如果启用这个选项（默认：禁用），另一个设备必须生成由 CANopen 管理器接收的 SYNC 报文。

● 时间

启动时间生成	如果启用这个选项（默认：禁用），CANopen 管理器发送 TIME 消息。
COB-ID	通信对象标识符，标识消息的时间戳。默认值：[0...2047]，预设 16#100。
Producer time (ms)	发送时间戳时的时间间隔，此值必须是任务周期时间的倍数，范围 [0..65535]。

5) 双击从站设备，在 CANopen 从站里配置从站基本参数、PDO 配置、SDO 配置。



● 基本参数

节点 ID	从站在 CANopen 网络唯一标示号范围 1~127(十进制)，需要和从站本身一致。
使能专家设置	如果启用这个选项（默认：禁用），显示由设备的设备描述（EDS 文件）预定义的所有设置。
使能同步发生器	如果启用这个选项（默认：禁用），此从站将发送同步信息。一个 CANopen 总线系统只能有一个启用同步生产。同步发送参数使用主站的同步配置参数。
使能同步发生器	仅在“CANopen_Manager_SoftMotion”中选择“启用同步生成”选项时可用。如果启用这个选项（默认：禁用），I/O 传输在总线上同步。从站充当同步生产者。同步间隔的参数在“CANopen_Manager_SoftMotion”的设置中定义。

- 心跳

使能节点保护	如果启用这个选项（默认：禁用），CANopen 管理器在保护时间（ms）间隔内向从站发送消息。如果从站没有使用给定的防护 COB-ID（通信对象标识符）进行响应，则 CANopen 管理器会按照生存期因子中定义的次数重新发送此消息，或者直到从站响应为止。如果从站没有响应，则将其标记为“不可用”。
保护时间（ms）	发送消息的间隔（默认：200 毫秒）。
生命周期因子	当从站没有响应时，将根据生命周期因子时间乘以保护时间建立节点保护错误。
使能心跳产生	模块按照生产者时间（ms）中给出的时间间隔发送检测信号。
发送端时间（ms）	看设备描述文件里设置的时间。

- 紧急情况

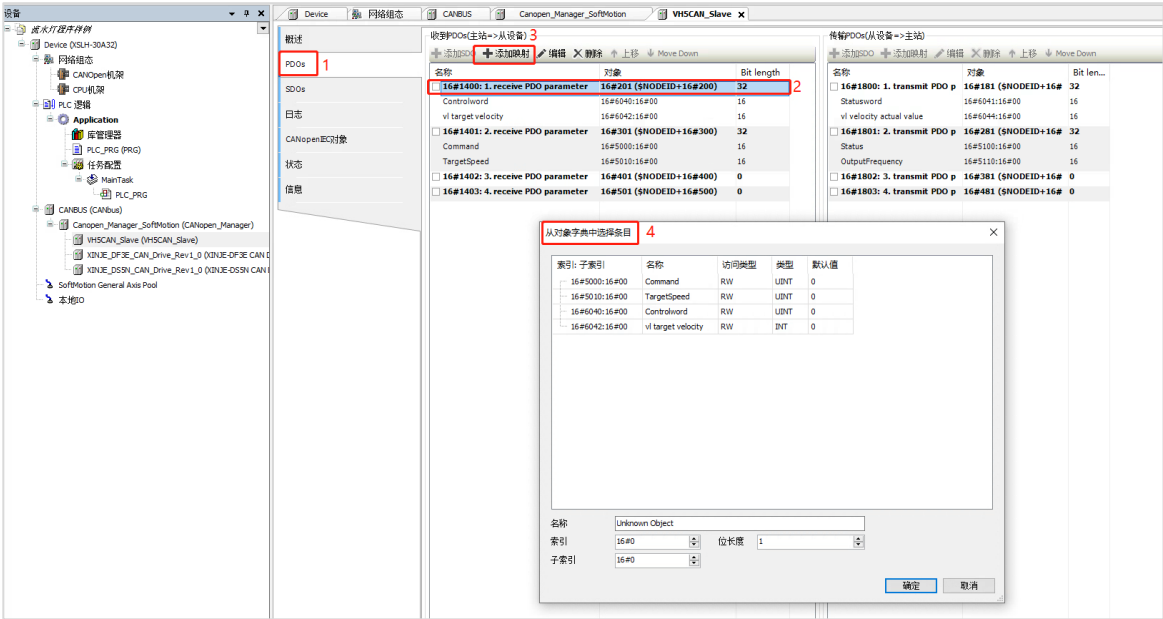
使能紧急情况	当发生内部错误时，从站发送具有唯一 COB-ID 的紧急消息。
COB-ID	从站发送紧急报文的 COB-ID，默认为\$NODEID+16#80。
时间	此功能的可用性取决于设备描述。
激活时间创建	设备发送时间消息。
COB-ID（十六进制）	通信对象标识符，标识消息的时间戳。
激活时间消耗	设备处理时间消息。

- 在重新启动时检查

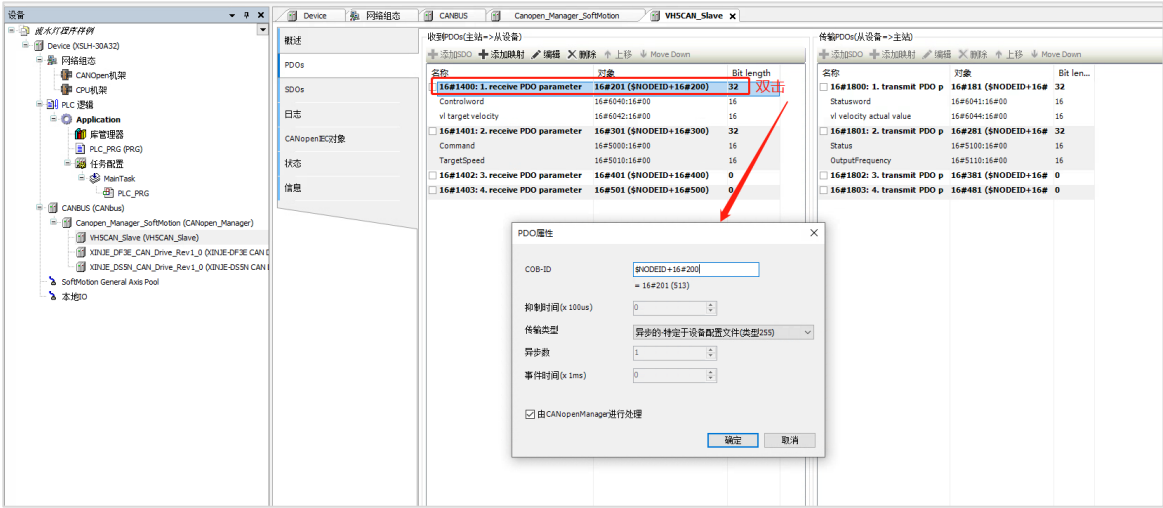
从 CANopen 从站（0x1018 身份对象）的固件中读取相应的信息，并与 EDS 文件中的信息进行比较。如果出现差异，则停止配置，不启动从站。

检查供应商 ID	在启动时检查供应商 ID。
检查产品号	在启动时检查产品编号。
检查修订号	启动时检查修订版本号。
PDOs	PDO（过程数据对象）对于主站和从站之间的实时数据传输，接收 PDO 为主站向从站发送的实时数据。

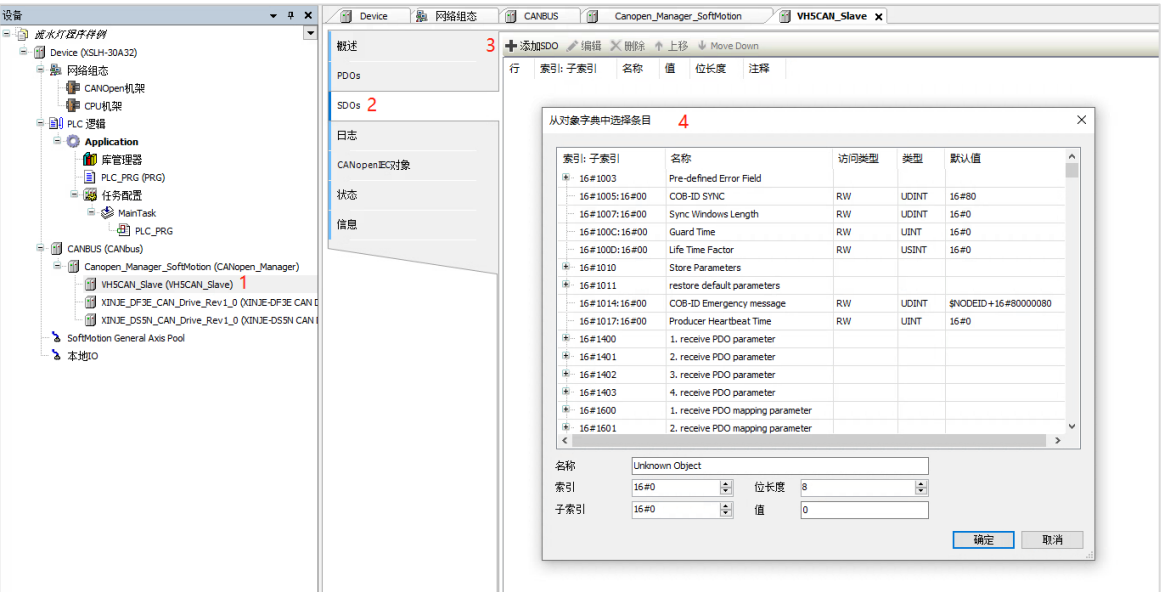
通过在 PDOs 界面，接收 PDO 来自于对象字典中从索引 1400~1403，发送 PDO 来自于对象字典中从索引 1800~1803，点击索引添加需要的通信参数，选择索引和子索引，点击“确定”。如用户需要添加/删除/修改映射地址，则需要在“接收 PDO”和“发送 PDO 映射”中设置。



双击加粗字体-索引，可以对具体 PDO 进行设置，可以设置其 COB-ID，传输类型的方式。

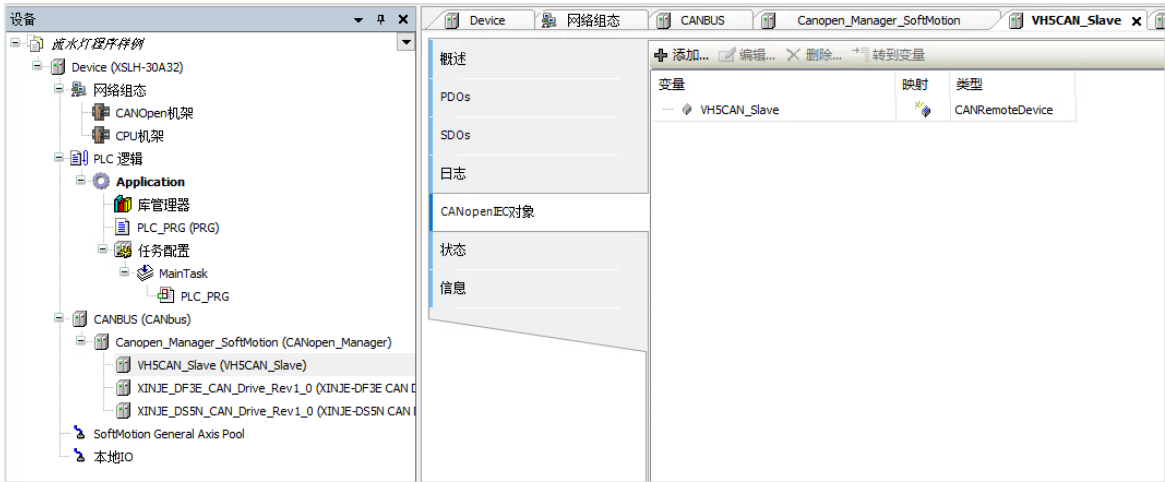


SDOs:



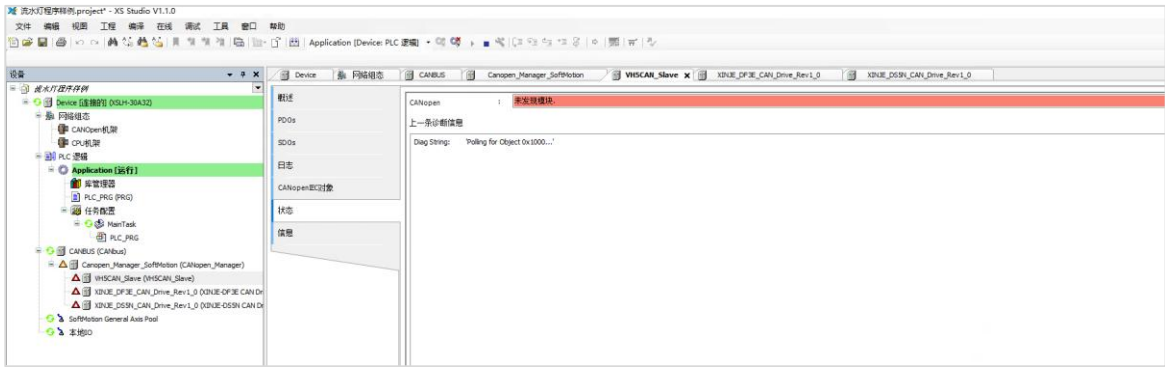
CANopen/IO 映射：

可以查看 CANopen/IO 映射关系、功能描述、实际地址以及映射变量的类型。



状态：

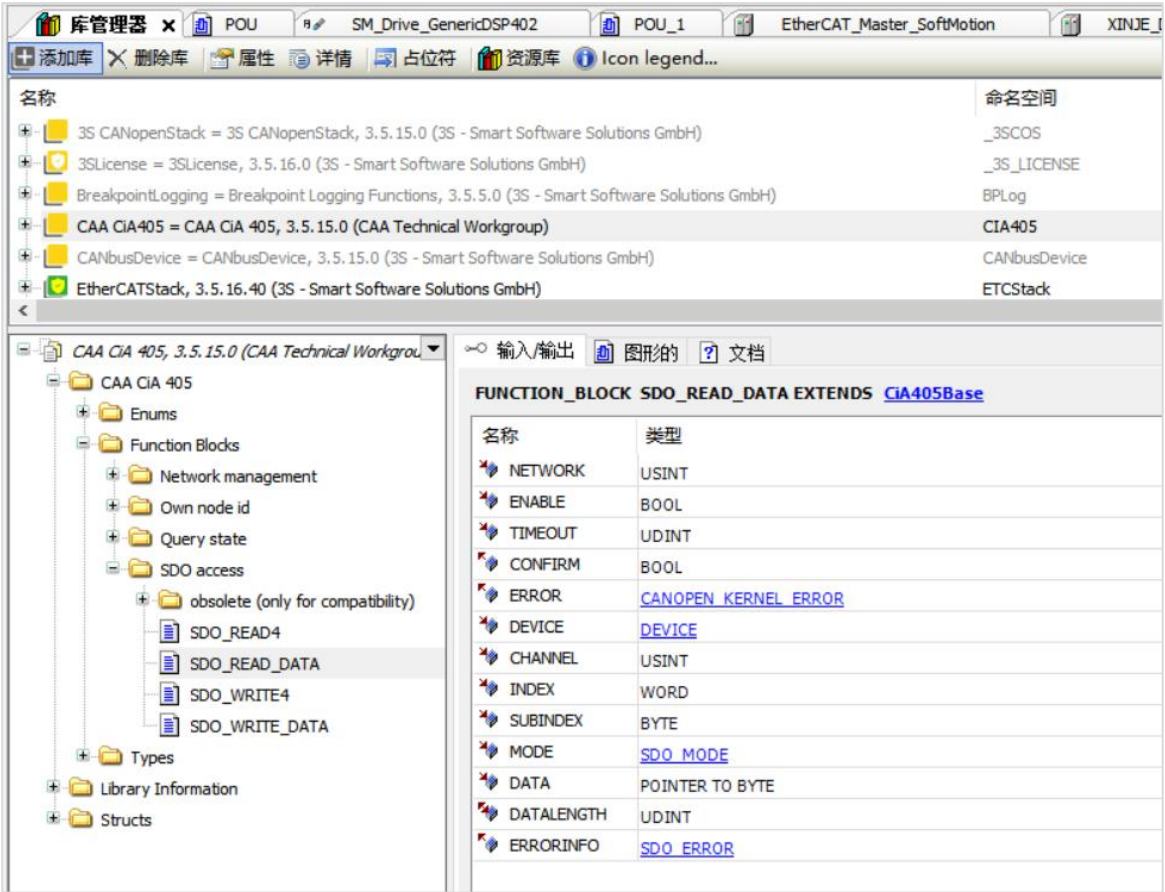
可以给用户提供设备状态（如“运行”，“未运行”）以及设备的诊断信息。



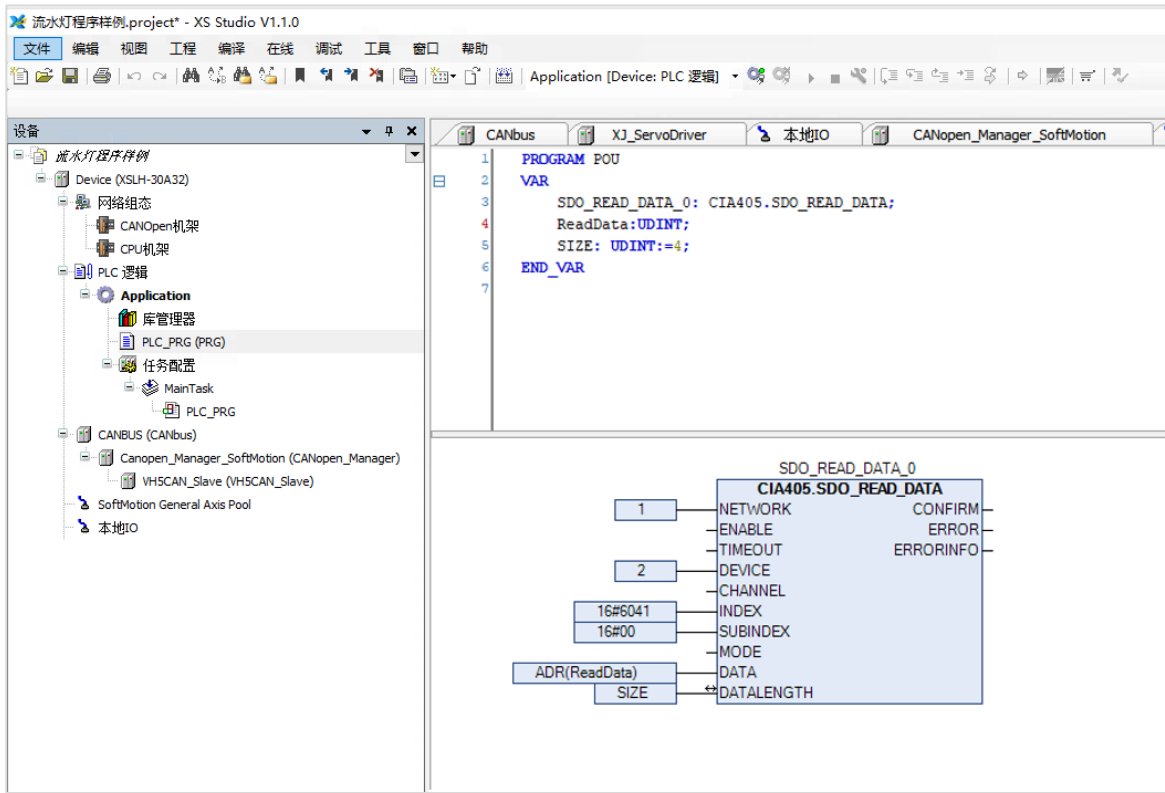
6) SDO 通信功能块

采用 PDO 方式做数据交换，简单且直接。但受到数量限制，而且这些数据都会占用总线，会造成连接在总线上的设备不能太多。SDO 通信主要是用于主节点对从节点的参数配置，用来设备之间传输的低优先级数据。

若使用 SDO 通信方式，需要添加库“CAA CIA405”，添加后即可在该库文件中看到“SDO access”文件夹。



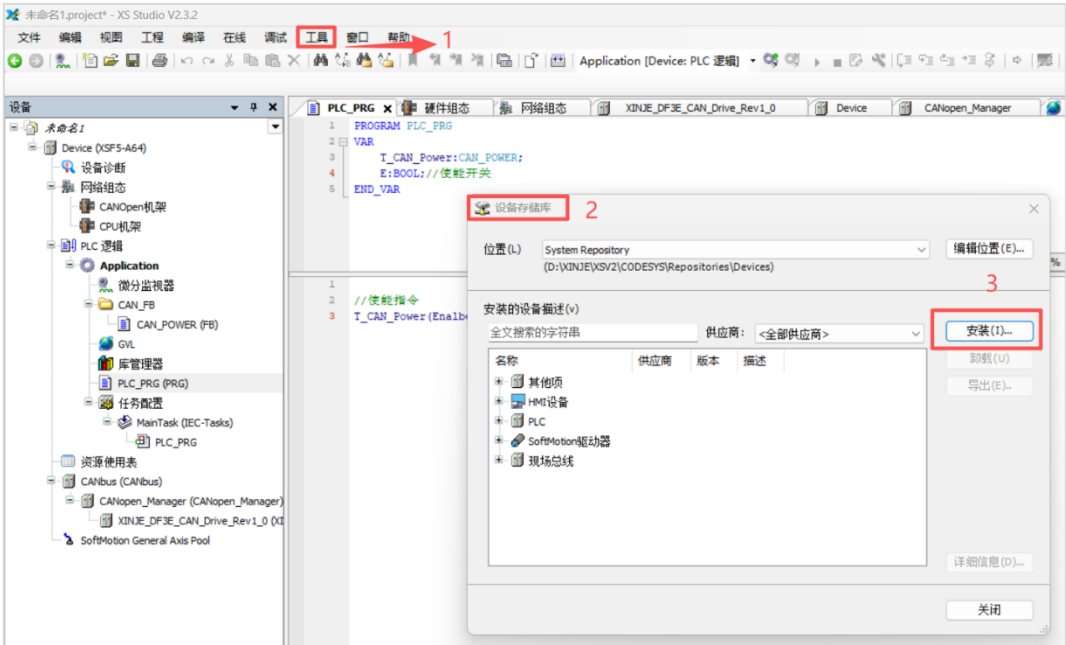
比如：添加功能块“CIA405.SDO_READ_DATA”，可以通过程序功能块并读取该参数。



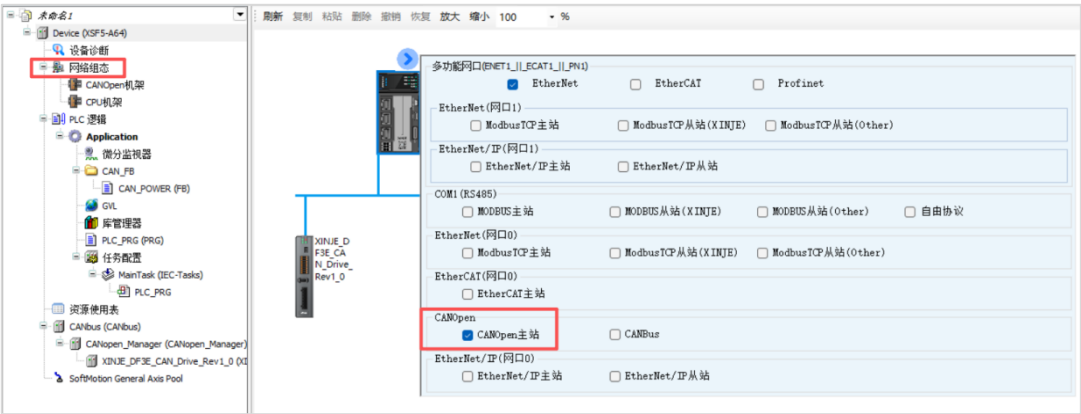
3-5-4. 应用举例

以 XSF5 主站连接信捷 DF3E 低压伺服从站为列。

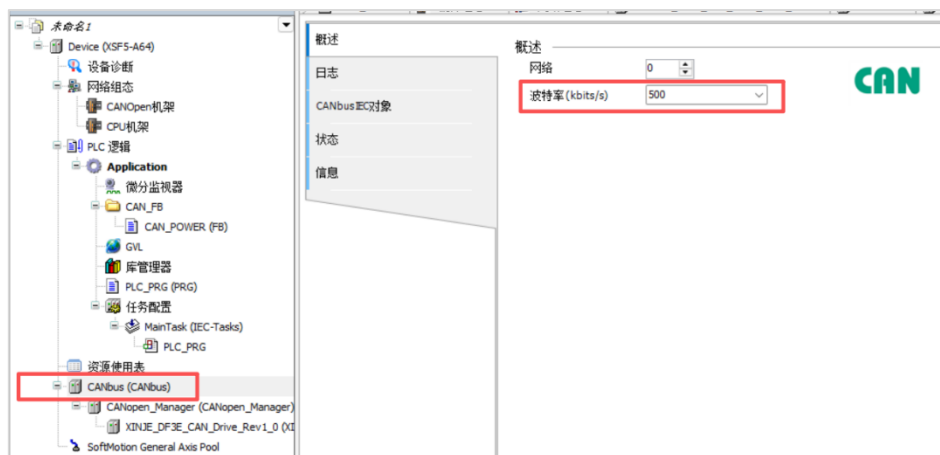
1、添加 CAN 设备对应的“.EDS”文件。打开 XS Studio 编程软件，点击“工具”下拉框中“设备存储库”，添加对应从站的“.EDS”文件。



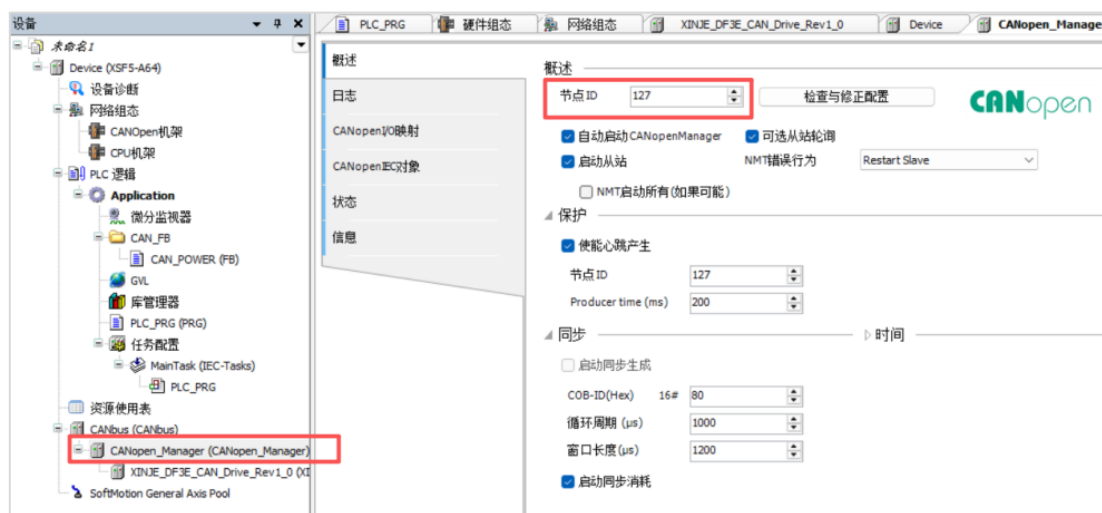
2、将网线一端的第一根（TX+）和第二根（TX-）线分别接到 PLC 上的 CAN+和 CAN-，另外一端插入到伺服的网口，完成 CAN 通讯接线。点击“网络组态”选择“CANOpen 主站”。



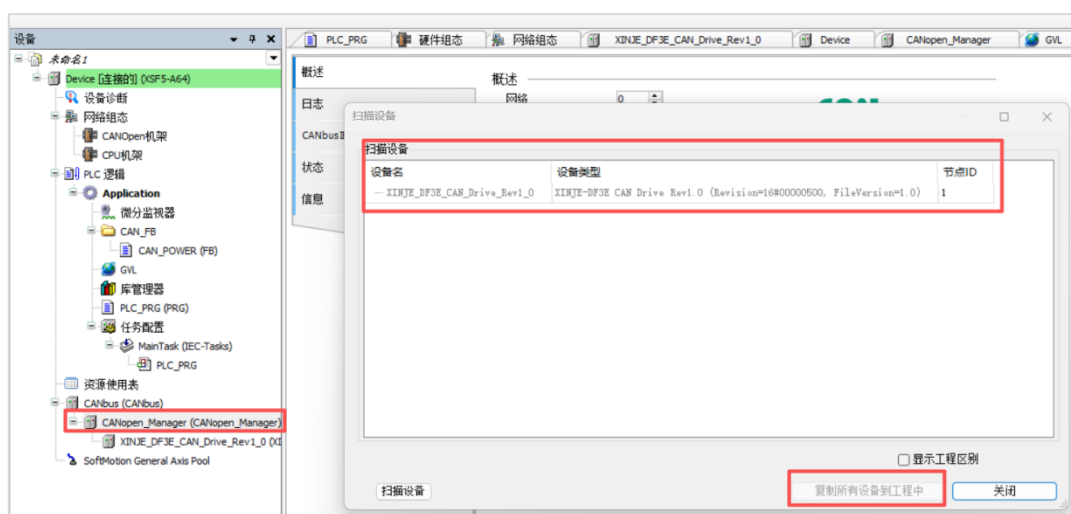
- 3、双击“CANbus”设置波特率，需与从站波特率一致。网络参数为本体设备的第几个 CAN 通讯口，XSF5 设备仅有一个 CAN 通道，故为 0。



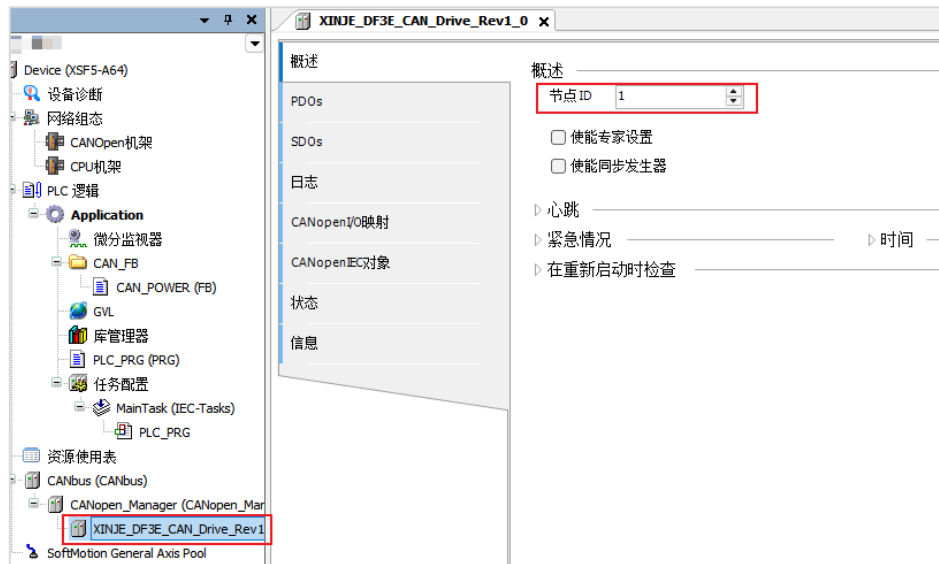
- 4、确定主站站号，在“CANopen_Manager”中的节点 ID 处设置主站站号，注意该站号不能与从站设备一致。



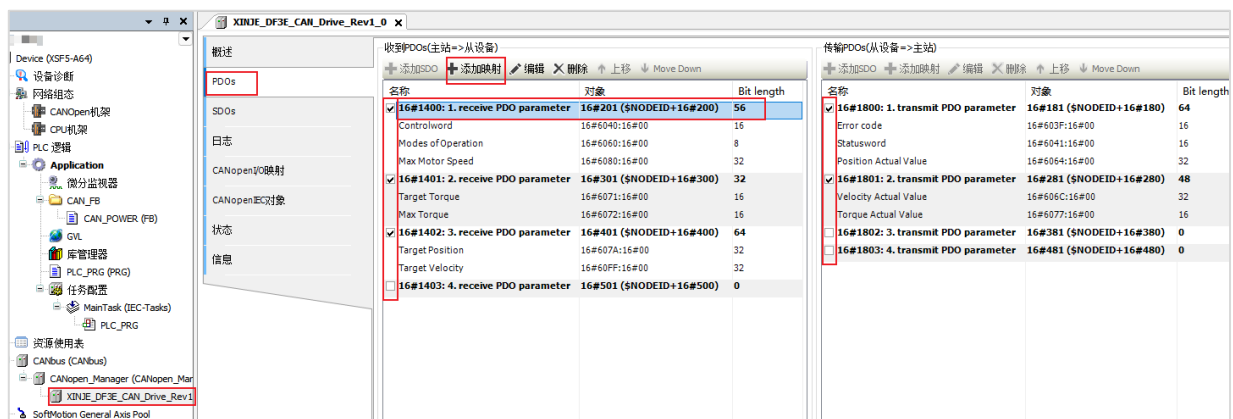
- 5、先登录下载程序，将设备信息同步至 PLC 端，然后退出登录，右键“CANopen_Manager”，点击扫描设备，确认扫描设备拓扑与实际拓扑无误后，点击“复制所有设备到工程中”。



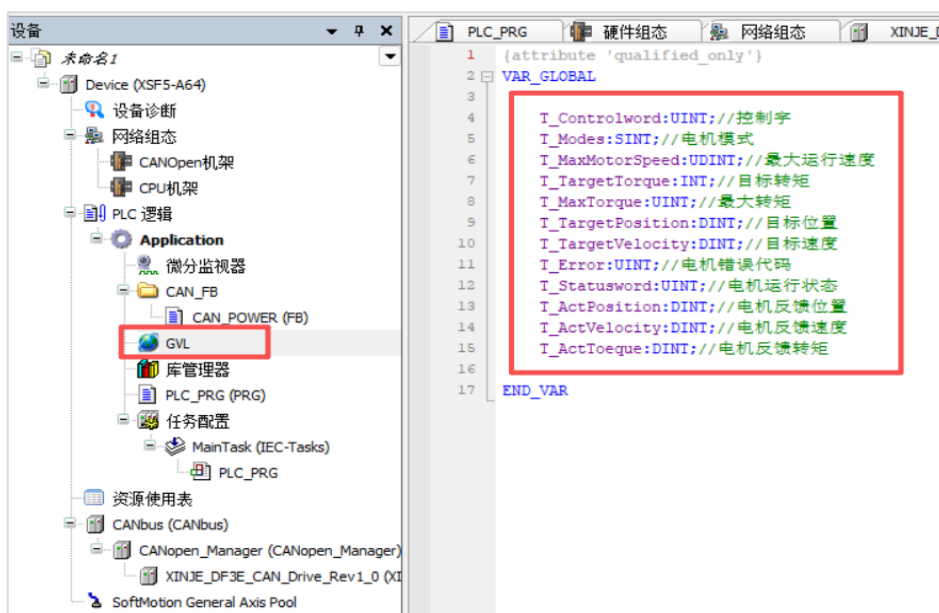
- 6、复制完成后，点击对应的设备，确定当前节点 ID 是否与设备一致。扫描进来后，从站的节点 ID 会自动识别。若是手动添加从站设备，需要手动修改从站 ID。



- 7、添加完 CAN 设备组态之后，在 CAN 从站界面点击“PDOs”，即可配置 PDO 参数，配置好 PDO 参数后要注意勾选列表框。



- 8、在 Application 右键-添加对象-全局变量表，然后在全局变量表中根据 CANOpen I/O 映射中的数据类型新建全局变量。如下图实例：



3-6. OPC UA 通讯

3-6-1. 通讯概述

OPC UA (OPC Unified Architecture, 基于 OPC 统一架构的时间敏感网络技术), 建立支持网络间互操作的时间敏感机制, 突破性实现信息技术 (IT) 与操作技术 (OT) 在物理层、数据链层、网络层、传输层、会话层、表达层和应用层全面融合的技术。该技术基于国际电工委员会 (IEC) 和电气和电子工程师协会 (IEEE) 国际标准搭建, 可为工业互联网网络体系构建提供标准化模块, 是建立从传感器到云端大带宽、高同步、广兼容通讯的关键技术。

OPC UA 实质上是一种抽象的框架, 是一个多层架构, 其中的每一层完全是从其相邻层抽象而来。这些层定义了线路上的各种通信协议, 以及能否安全地编码/解码包含有数据、数据类型定义等内容的讯息。利用这一核心服务和数据类型框架, 人们可以在其基础上 (继承) 轻松添加更多功能。

3-6-2. 参数配置



在 XSA-W 工控机桌面安装并运行上位机通讯软件时, 内部通讯 IP 需按机型区分配置:

1、XSA330-W、XSA500-W 系列工控机:

设备内部通讯统一采用回送地址 127.0.0.1。

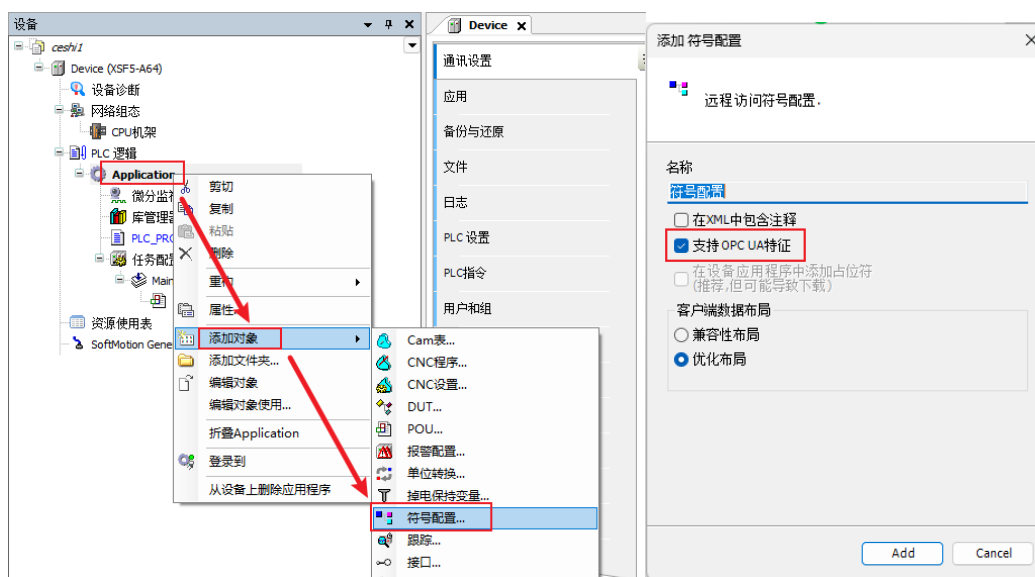
2、XSA230-W、XSA620-W 支持虚拟化的工控机:

①内部通讯目标 IP 固定为 192.168.2.2;

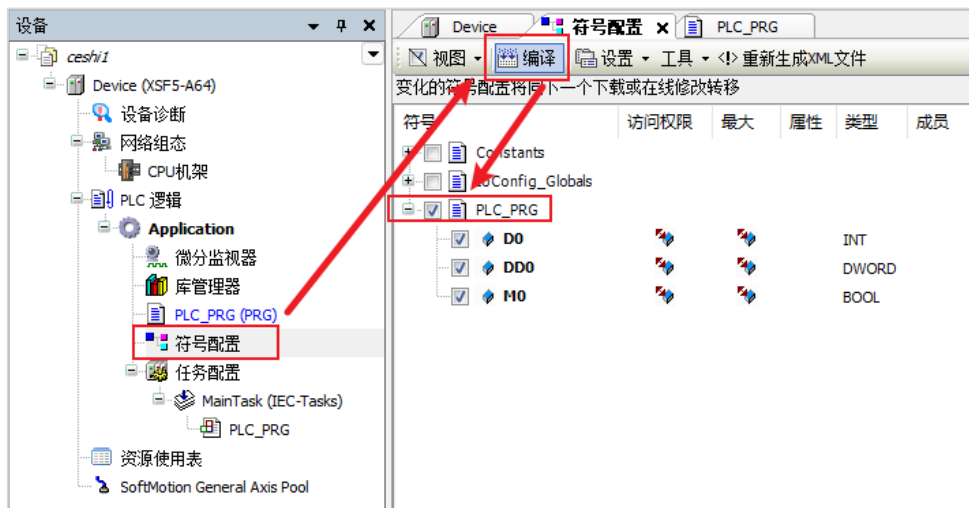
②本机虚拟网卡默认 IP 为 192.168.2.1, 可连接显示器查看;

③若用户需要修改虚拟网卡的 IP, 则仅支持修改为 192.168.2 网段的 IP 地址, 不可更换为其他网段。

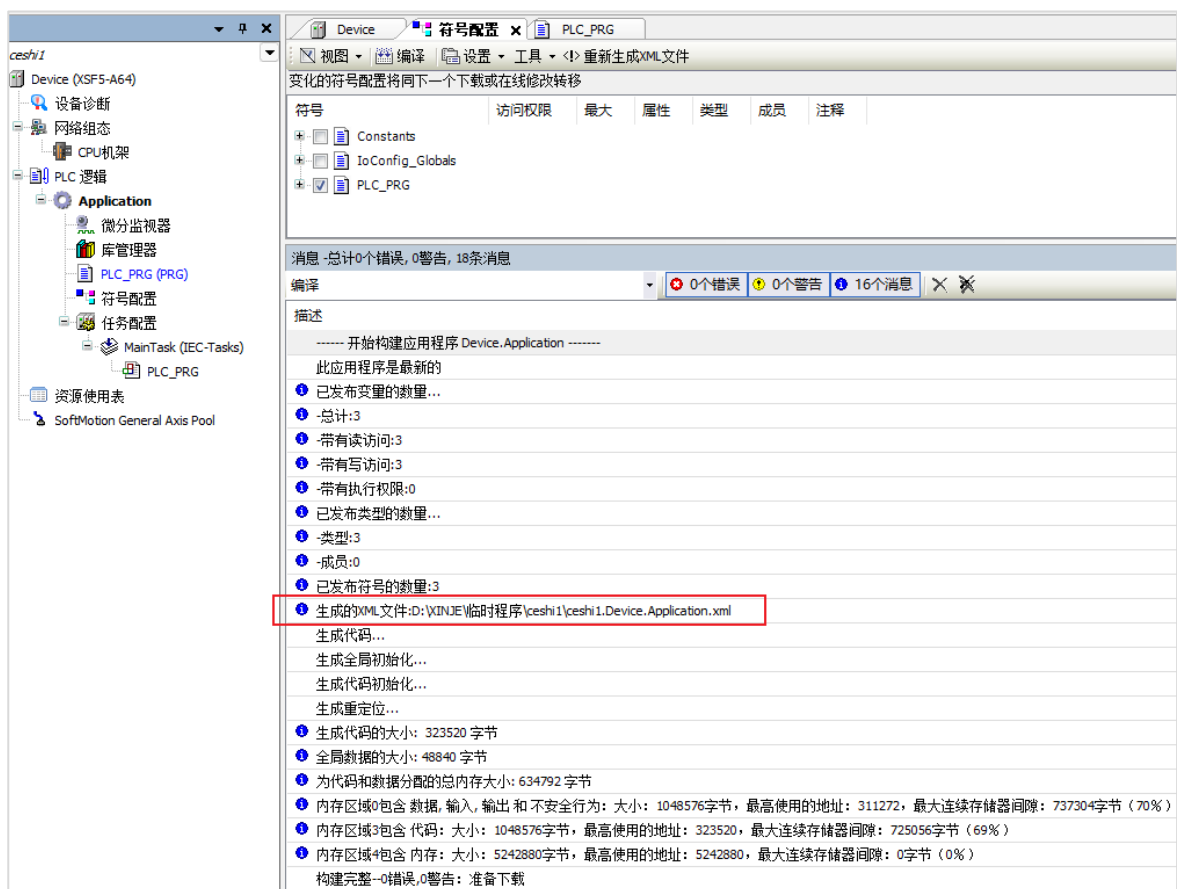
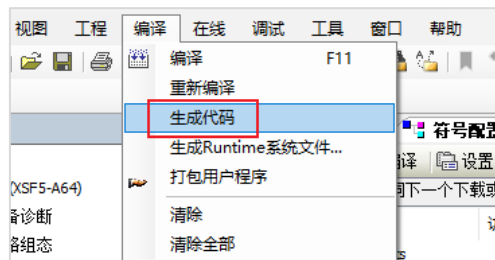
- 1、在应用的工程中, 右击 “Application”, 选择 “添加对象” - “符号配置”, 在弹出的对话框中勾选 “支持 OPC UA 特征”, 则开启 OPC UA 的功能。

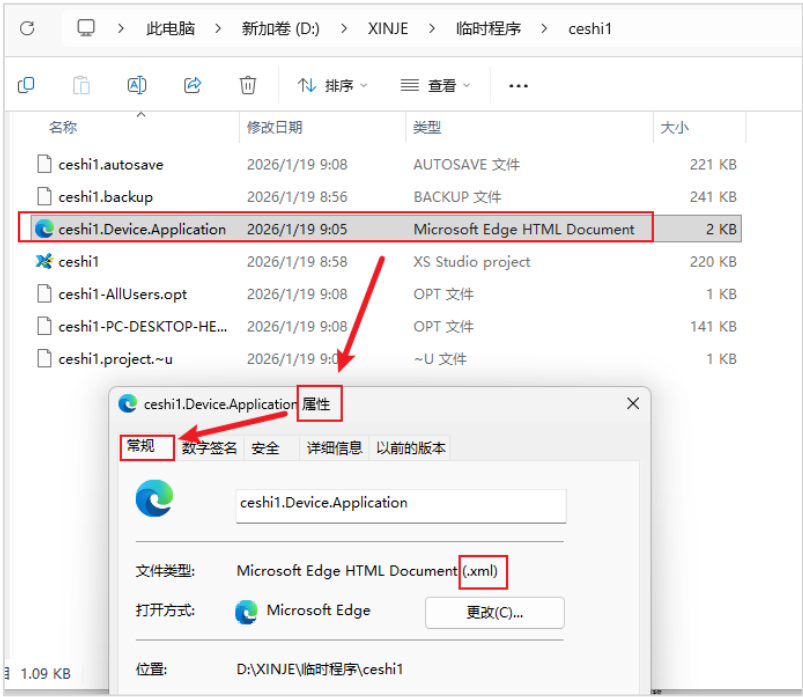


- 2、双击 “符号配置”, 在弹出的界面里点击 “编译” 后, 勾选需要监控的参数。

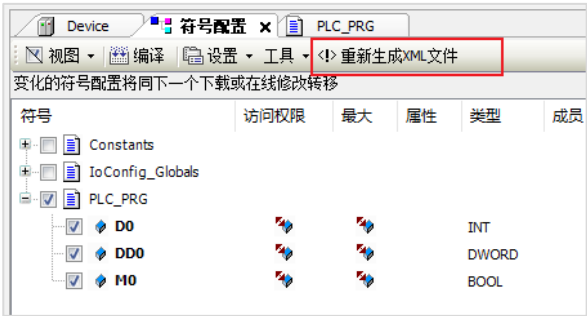


- 3、在菜单栏的“编译”中选择“生成代码”，生成所需要的 xml 文件，路径为该工程所在的路径。





- XS StudioV2.3.2 及以上版本软件有“重新生成 XML 文件”按钮，其功能是若误删除了之前生成的 XML 文件操作时，可以无需修改程序重新编译、直接点击该按钮就可以再次生成对应 XML 文件。

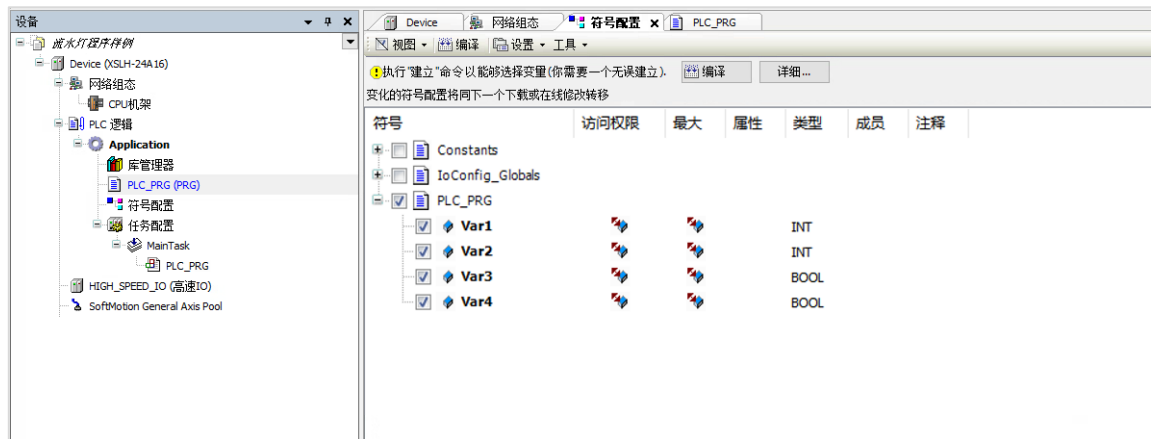


3-6-3. OPC UA 样例

例一：使用信捷 XSLH-24A16 和威纶通（型号为 CMT3105X）的触摸屏进行 OPC UA 通讯。

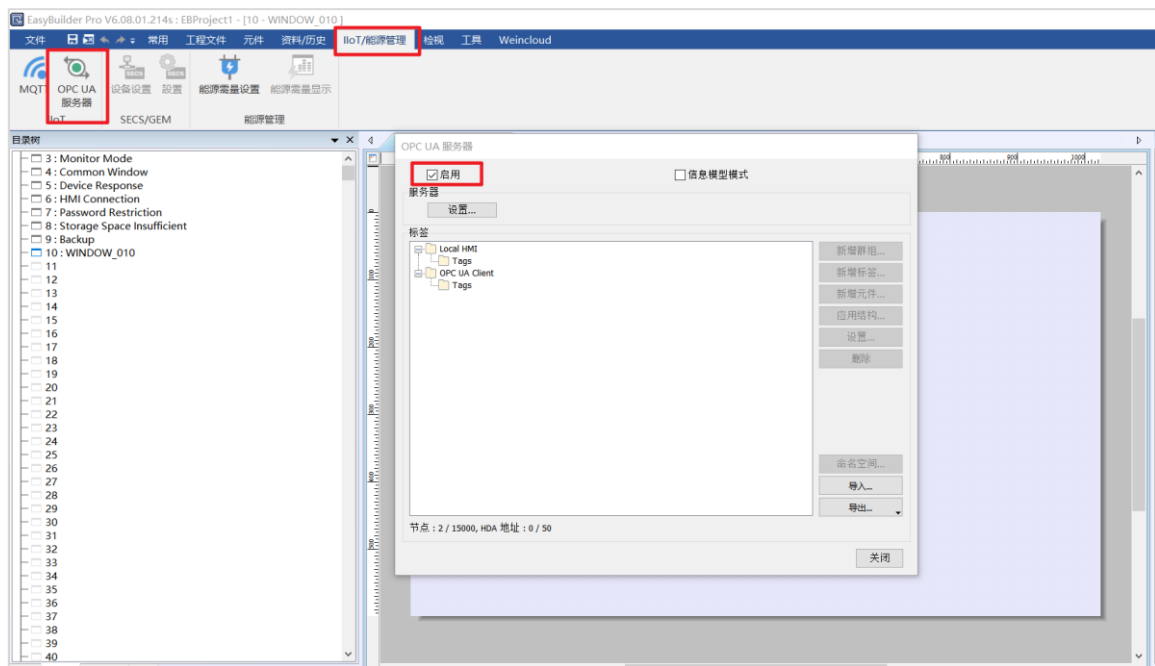
程序编写：

1) 在 XSLH-24A16 中写几个参数，并在 OPC UA 界面中勾选登录下载。

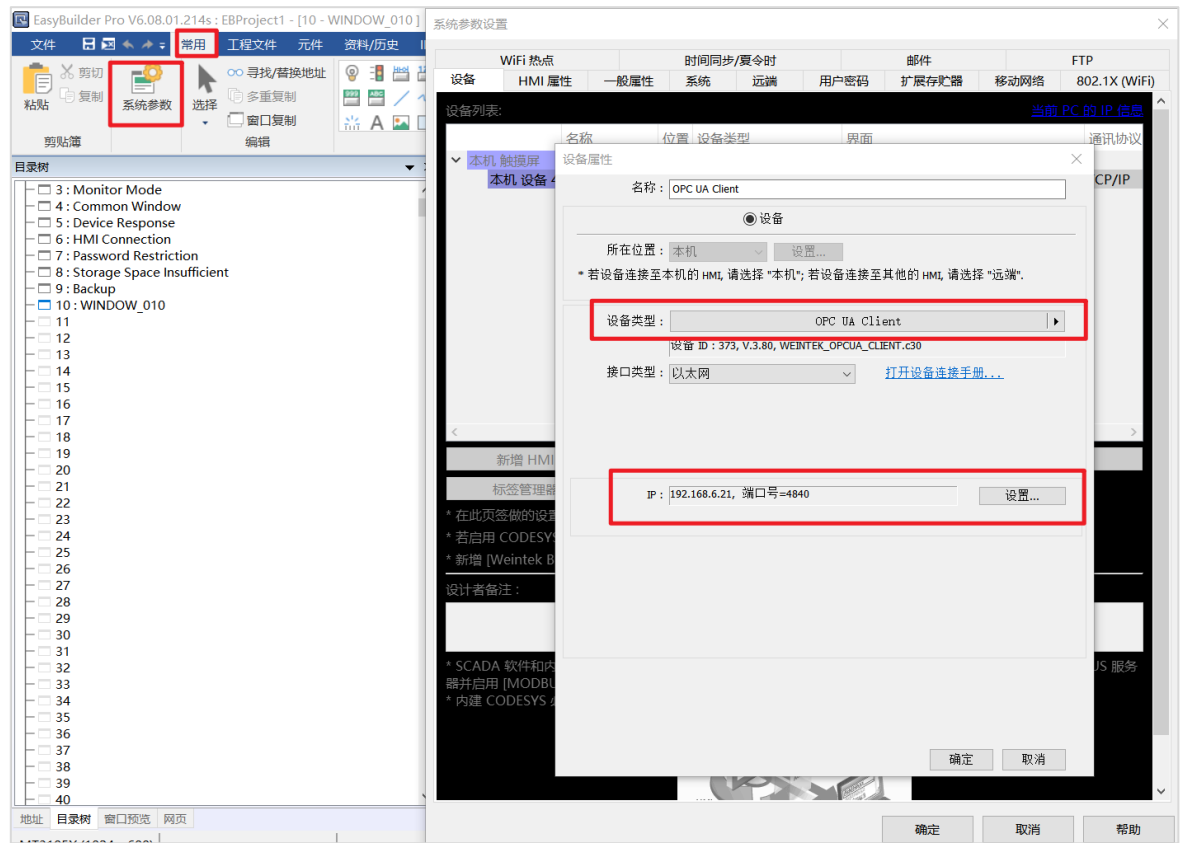


2) 威纶通触摸屏配置

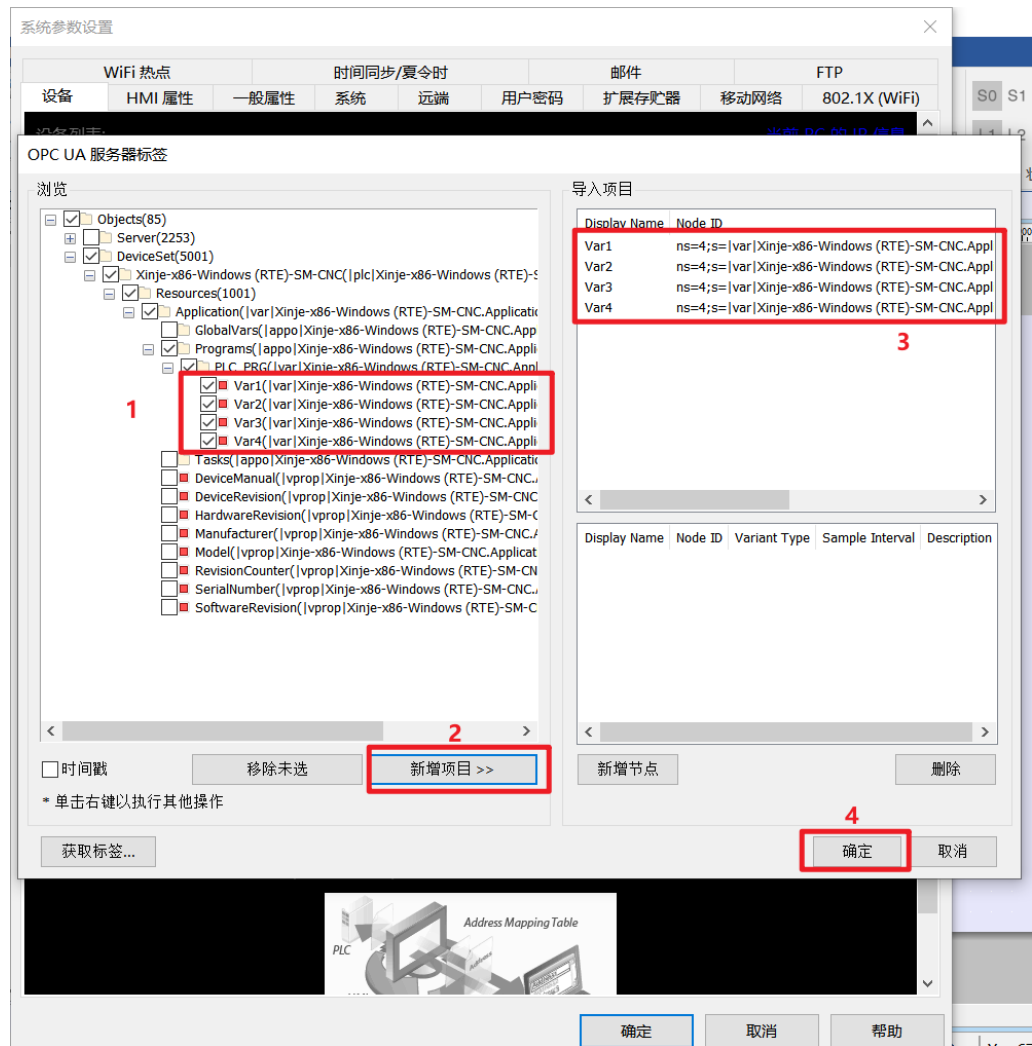
① 在“IIOT 能源管理”界面中选择“OPC UA 服务器”，在点开的“OPC UA 服务器”界面中勾选“启用”，弹出的界面直接点击确定，在弹出相关标签后，将界面关闭。



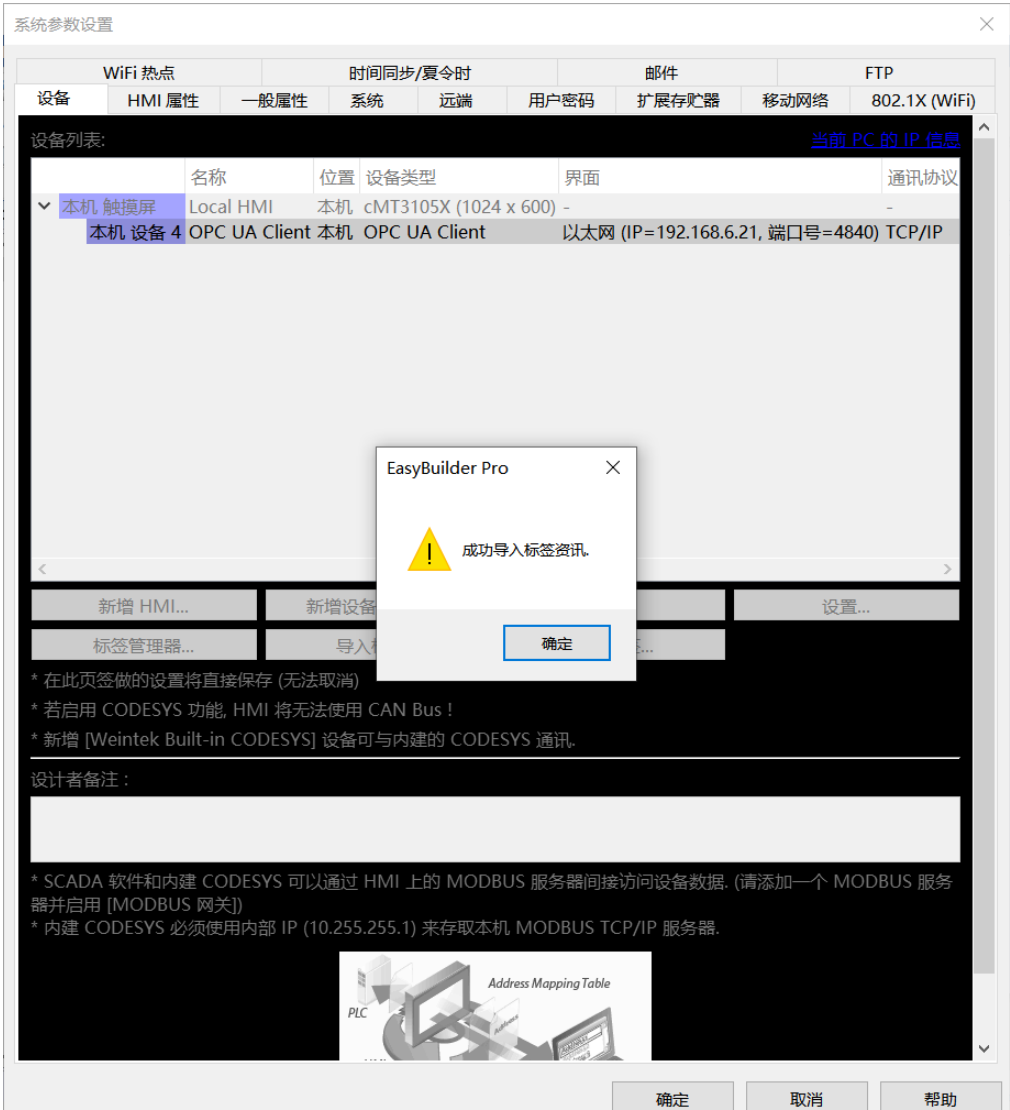
② 在“系统参数设置”界面中，点击“新增设备/服务器”，在弹出来的“设备属性”界面中选择设备类型为“OPC UA Client”，并设置 IP 为 XSLH-24A16 的 IP 地址,设置好后点击确定。



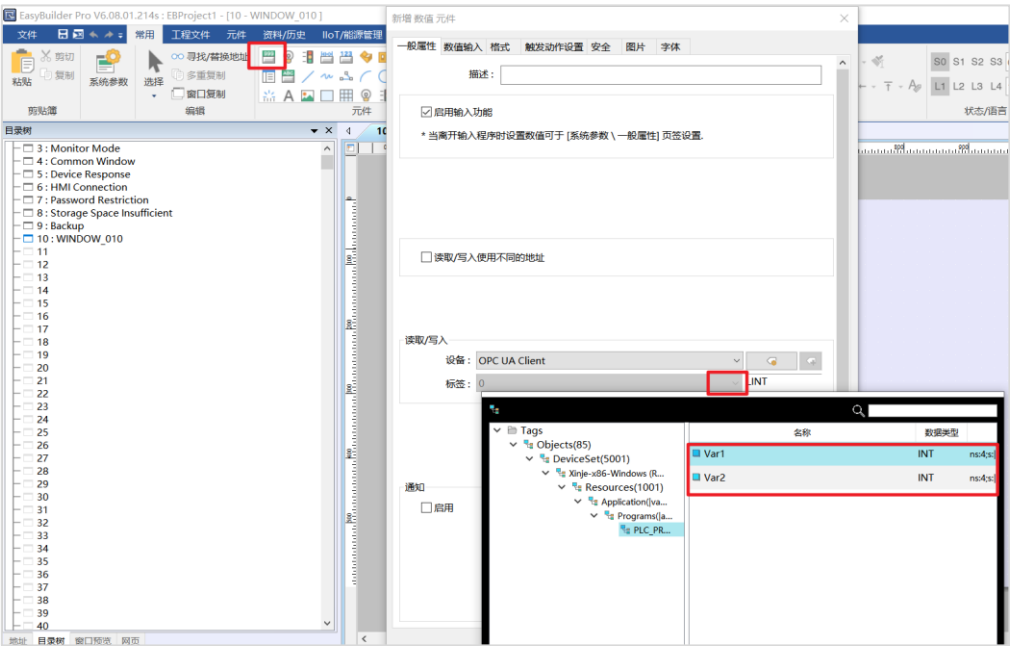
③ 点击“导入标签”，在弹出来的界面中选择确定后，会出现“OPC UA”服务器标签界面，可在此界面中选择 PLC 的相关数据。

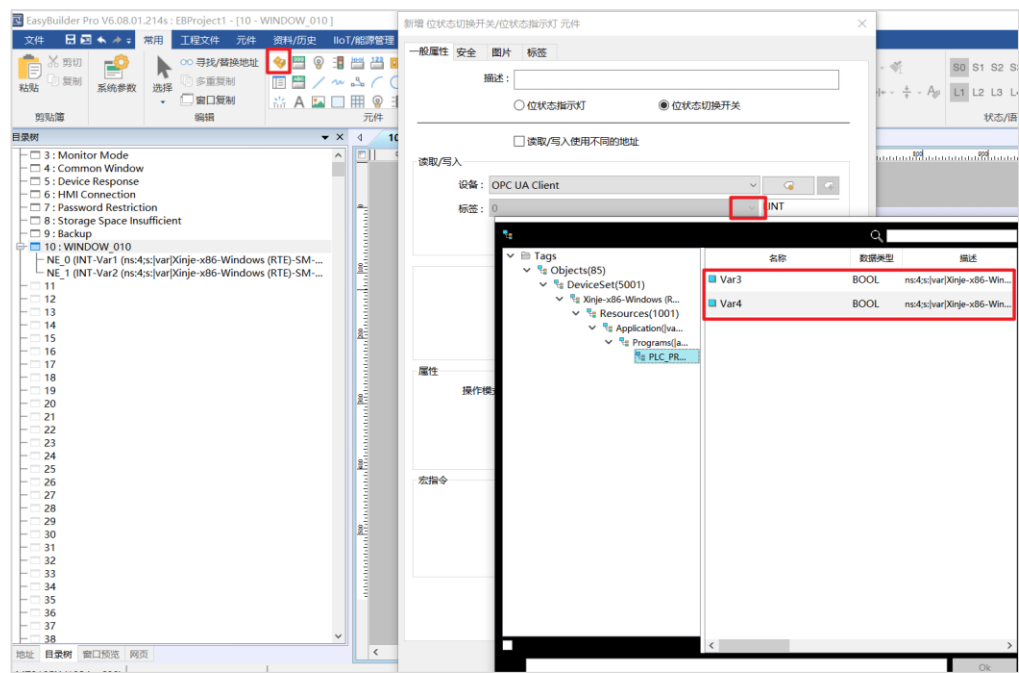


④ 点击确定，出现“成功导入标签资讯”。

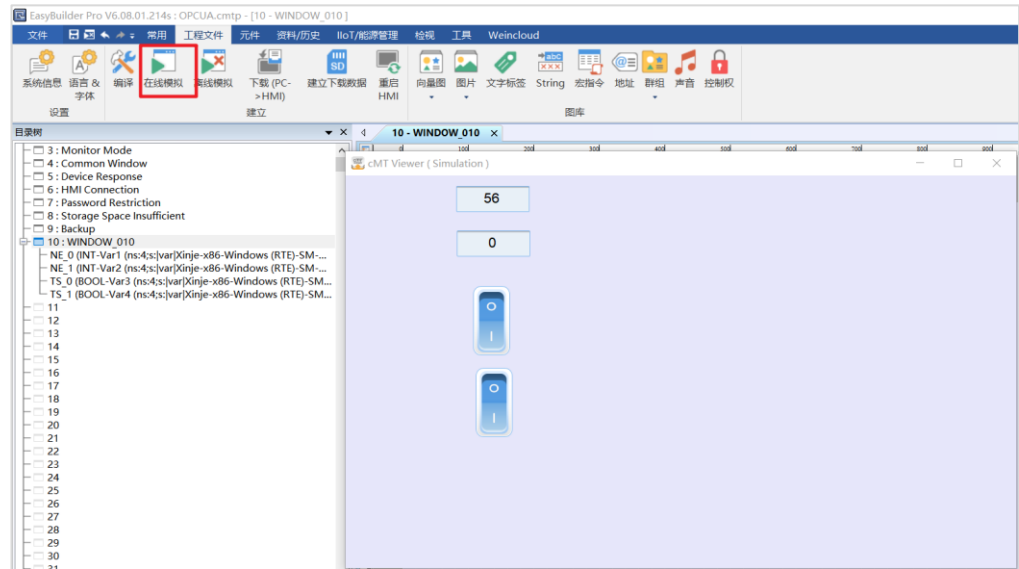


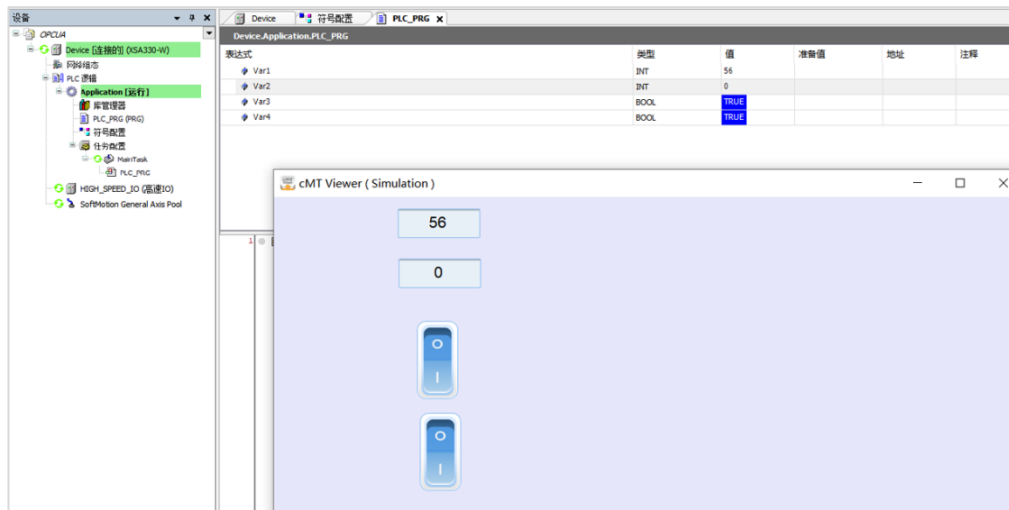
⑤ 在“元件”里选择相关类型的元件，在弹出的界面里的“标签”点击下拉图标，出现相关的参数。依次选择所有参数。





⑥ 在“工程文件”里选择“在线模拟”，可实现触摸屏和 PLC 之间通讯。

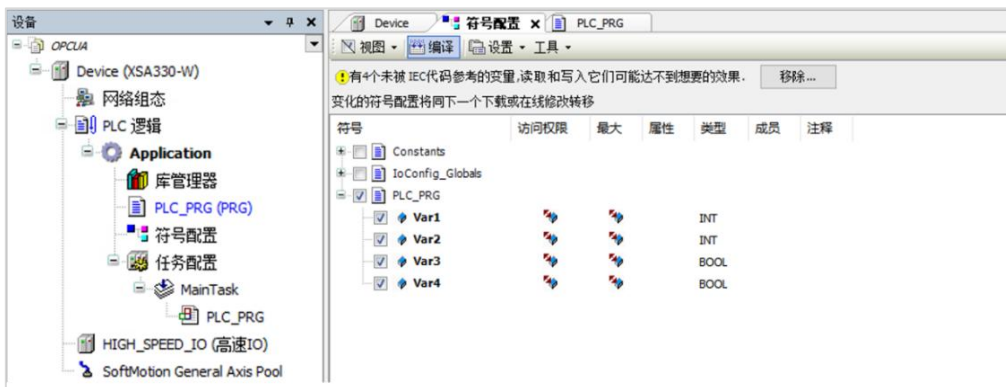




例二：使用信捷 XSA330-W 和威纶通（型号为 CMT3105X）的触摸屏进行“codesys v3”通讯。

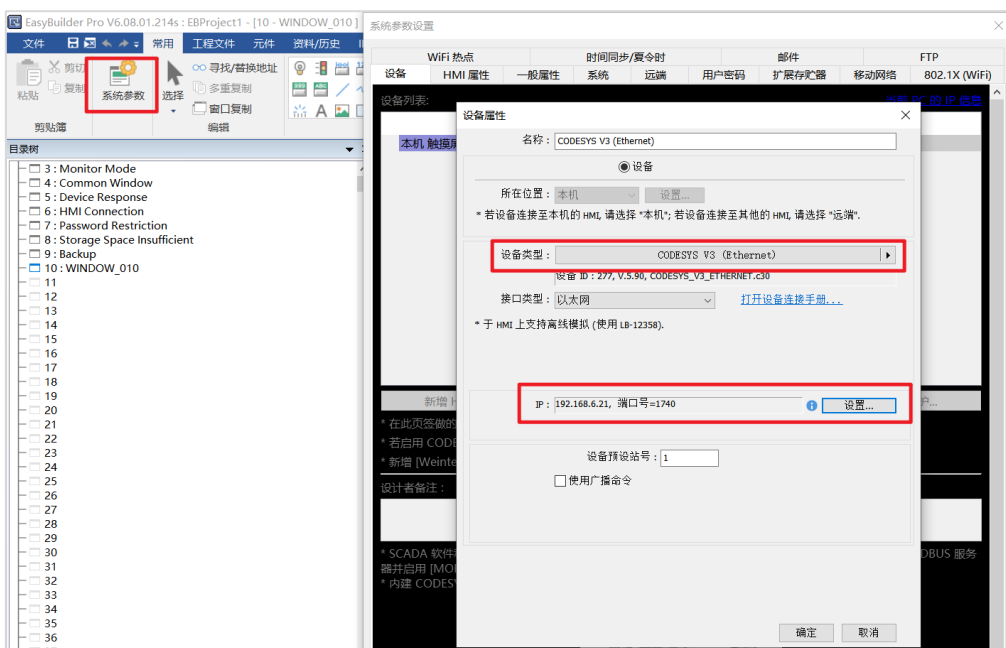
程序编写：

1) 在 XSA330-W 中写了几个参数，并在 OPC UA 界面中勾选登录下载。



2) 威纶通触摸屏配置

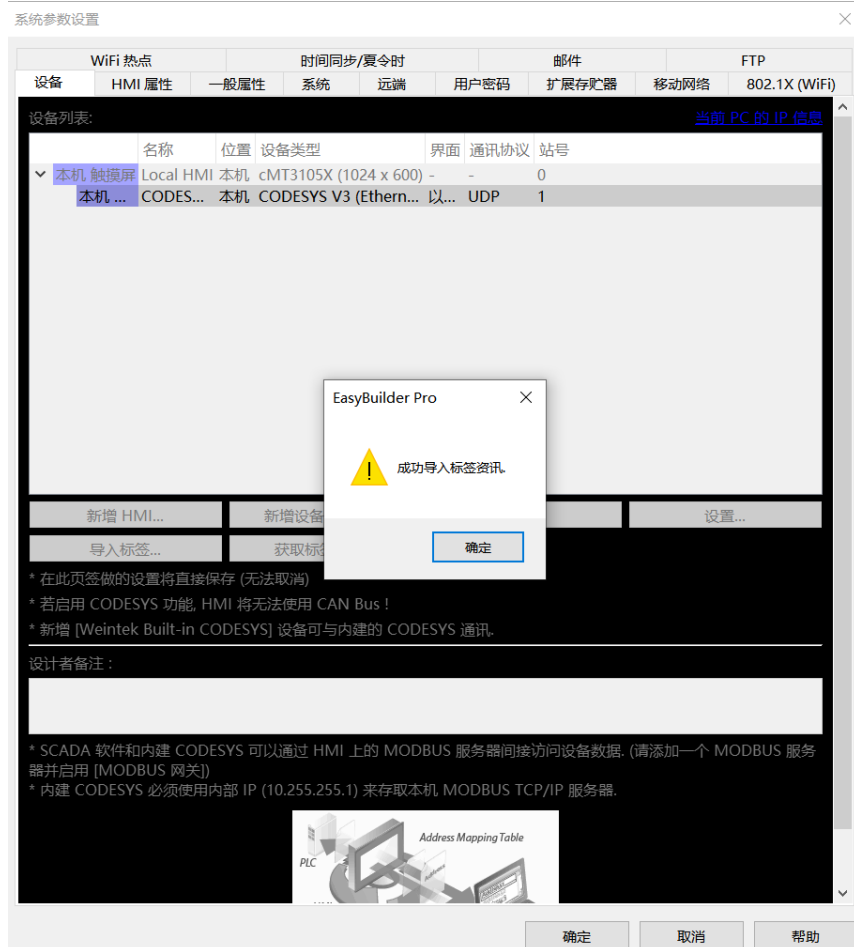
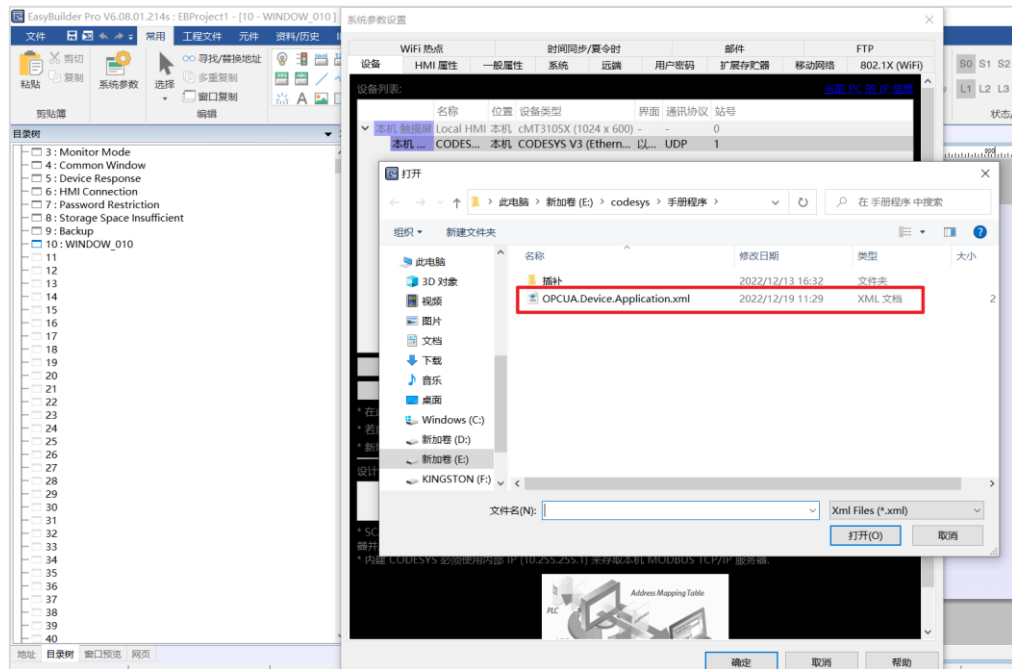
① 在“系统参数设置”界面中，点击“新增设备/服务器”，在弹出来的“设备属性”界面中选择设备类型为“CODESYS Automation Alliance”中的“CODESYS V3”，并设置 IP 为 XSA330-W 的 IP 地址，设置好后点击确定。



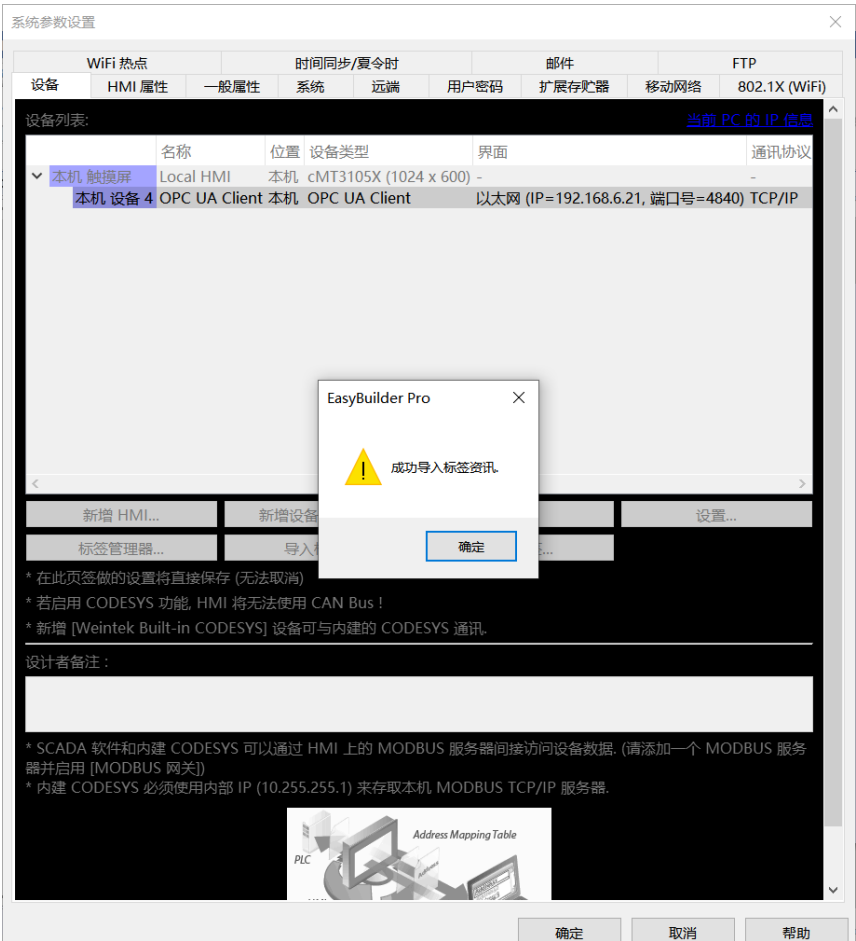
- ② 点击“导入标签”，导入生成的 XML 文件，导入成功后会出现“成功导入标签资讯”。



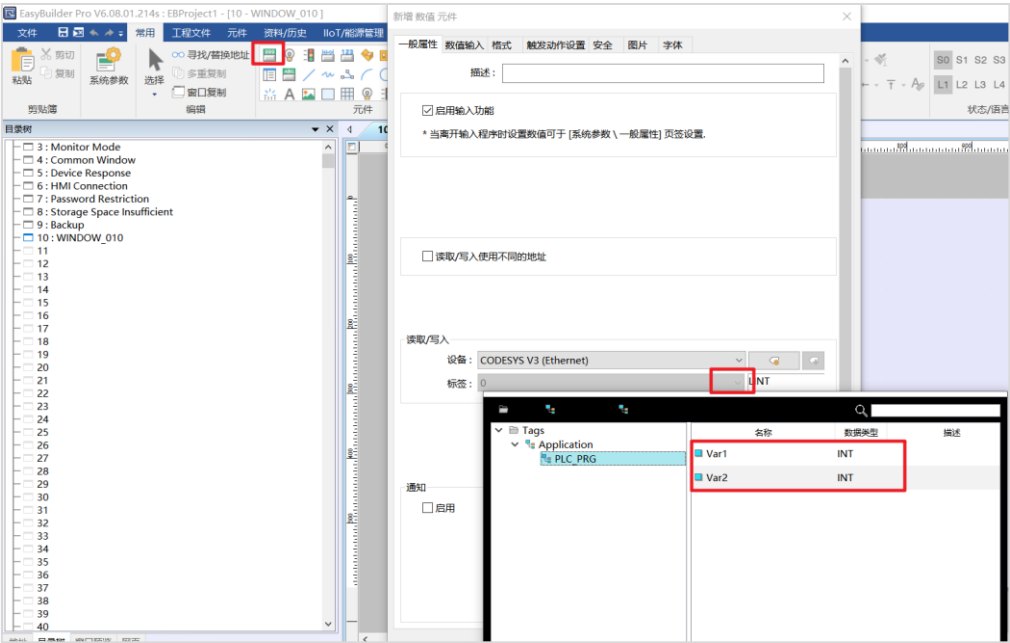
生成的 XML 文件是自动生成在程序保存目录中。

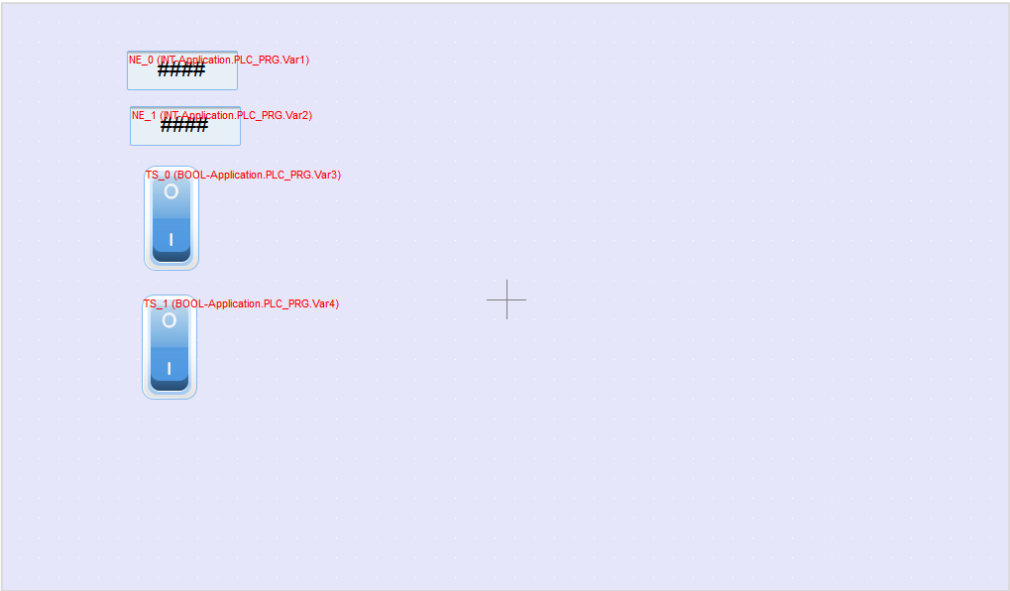


- ③ 点击确定，出现“成功导入标签资讯”。

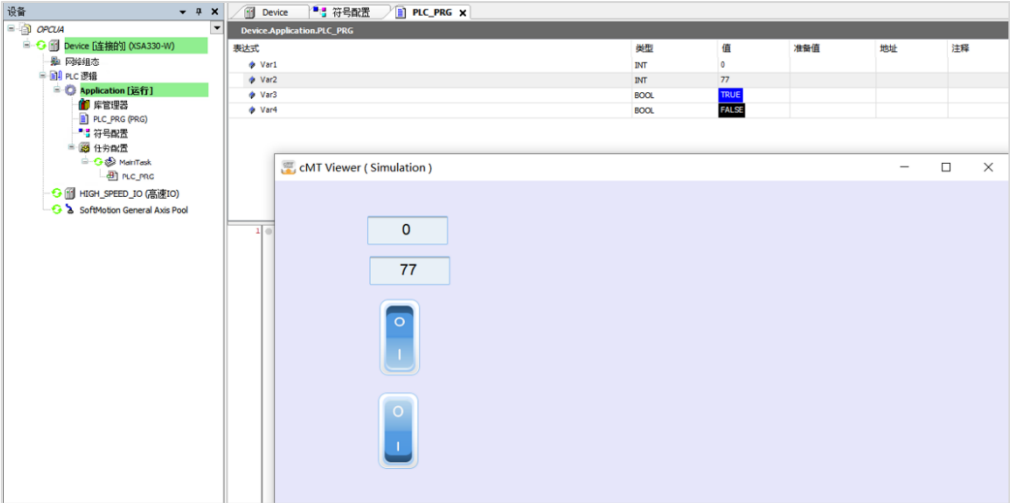


④ 在“元件”里选择相关类型的元件，在弹出的界面里的“标签”点击下拉图标，出现相关的参数。依次选择所有参数。同例一的步骤。





⑤ 在“工程文件”里选择“在线模拟”，可实现触摸屏和 PLC 之间通讯。



3-7. PROFINET 通信

3-7-1. 通讯概述

PROFINET 由 PROFIBUS 国际组织（PROFIBUS International, PI）推出，是新一代基于工业以太网技术的自动化总线标准。

PROFINET 为自动化通信领域提供了一个完整的网络解决方案，囊括了诸如实时以太网、运动控制、分布式自动化、故障安全以及网络安全等当前自动化领域的热点话题，并且，作为跨供应商的技术，可以完全兼容工业以太网和现有的现场总线（如 PROFIBUS）技术，保护现有投资。

3-7-2. 通讯规格

项目	规格
适用机型	XSF 系列
通信模式	RT 模式
最小通讯周期	1ms（RT 模式）
模式	仅支持从站
介质冗余（MRP）	不支持
介质路径规划冗余（MRPD）	不支持
网络接口	1 个 RJ45 口
单个 PN Device 支持模块的数量	64 个 IO 模块
单个 PN Device 支持的发送 Byte 数	1440Byte
单个 PN Device 支持的接收 Byte 数	1440Byte
支持端口	ENET1 ECAT1
物理层	100BASE-TX
连接速率	10/100Mbps，自适应，全双工
传输距离	两节点之间小于等于 100m
拓扑结构	支持线型、星型、树型等
传输介质	超五类及以上

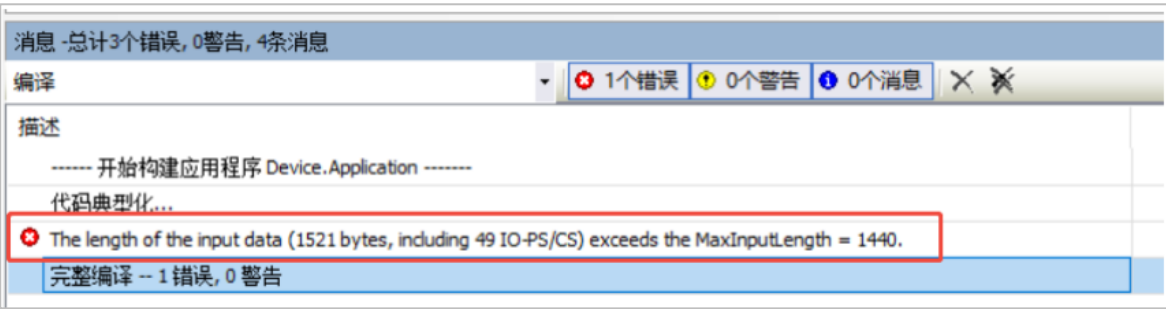
3-7-3. PROFINET 通讯说明

1、1440byte 计算大小说明

- 超出 64 个模块报错信息提示如下：

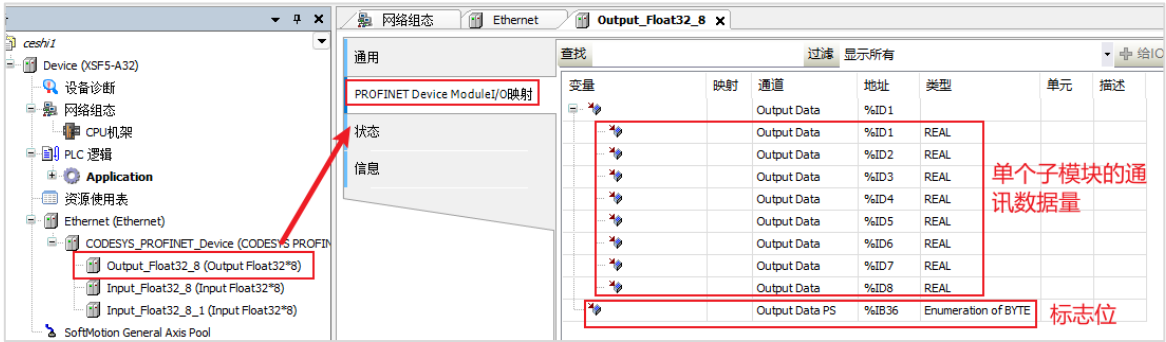


- 数据传输长度不能超出 1440Byte，否则会报错提示如下：



- Byte 计算方式

所有子模块下 I/O 映射：总数据量 = 所有子模块的“I/O 映射”界面参数数量总和 + 标志位总和，通讯传输最大不能超过 1440Byte。



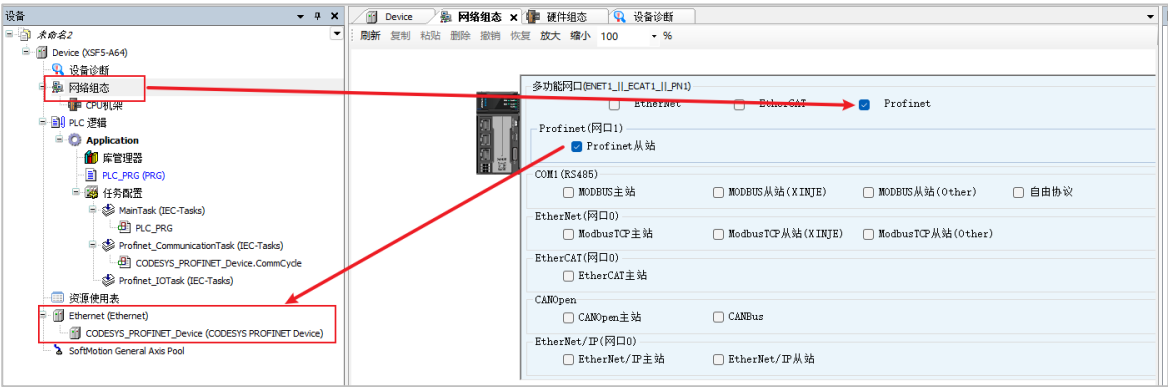
2、主站更改 IP 说明

不管主站使用 CODESYS RUNTIME PLC 还是西门子 PLC,主站都有一个对从站 IP 配置的选项，在 PN 从站与主站建立连接后，主站会通过从站 IP 参数的配置来设置从站，从而 PN 从站 PLC 的 IP 发生改变。

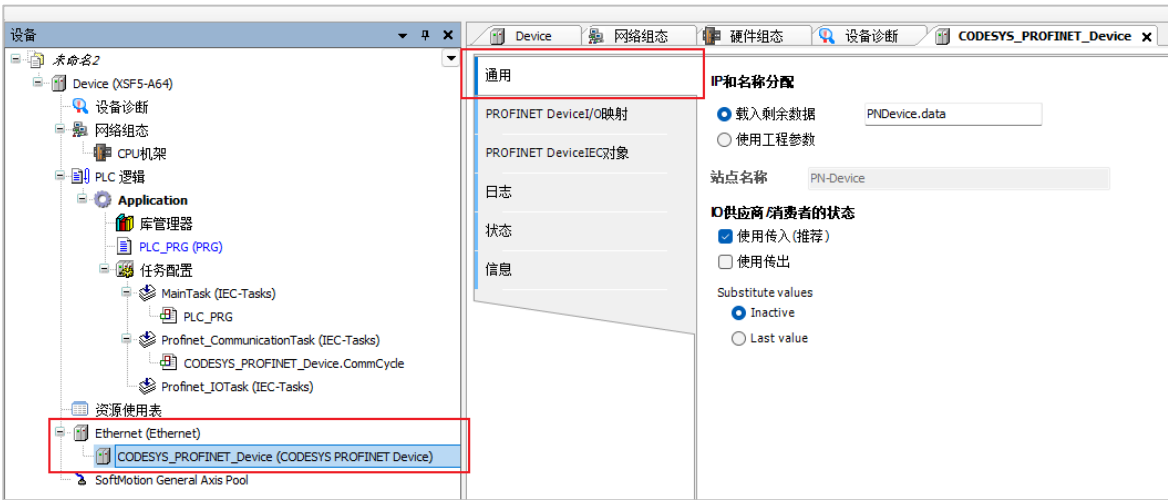


3-7-4. PROF INET 从站参数配置

- 1、鼠标单击网络组态内的使能窗口, 多功能网口 (ENET1_II_ECAT1_II_PN1), 勾选“profinet”。在 “profinet (网口 1)” 勾选 “profinet 从站” 左侧设备树节点下会自动添加 “CODESYS_PROFINET_Device”。



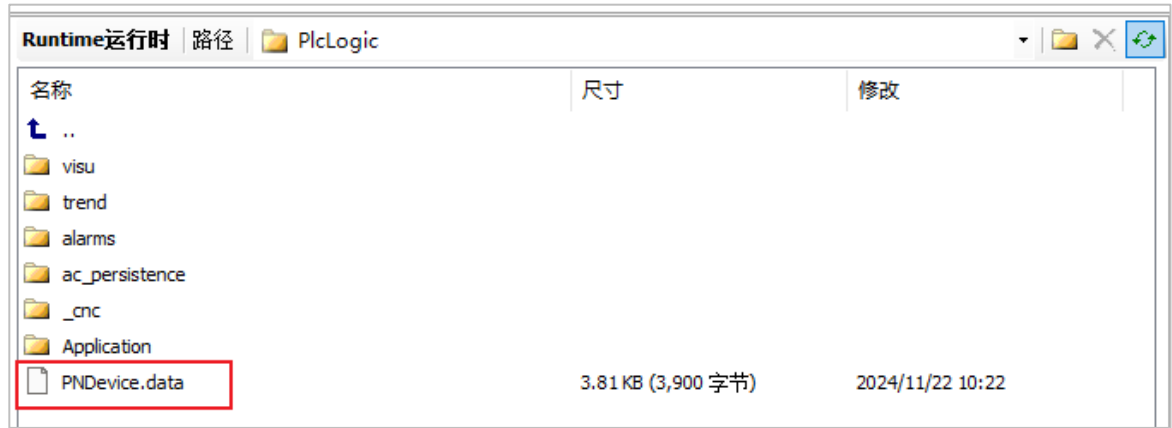
- 2、添加从站设备后, 在设备树节点下双击“CODESYS_PROFINET_Device”节点打开配置界面, 可切换至 profinet 从站配置界面。



3-7-4-1. 通用

1、IP 和名称分配

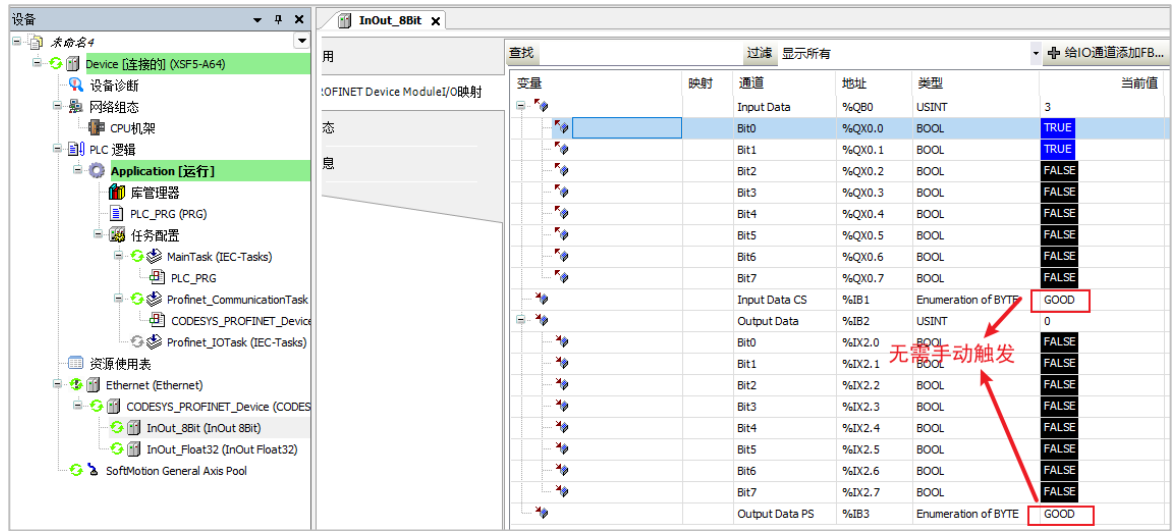
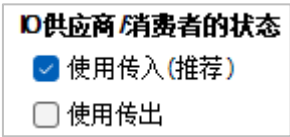
载入剩余数据：使用文件的 IP 设置和站名，这个文件存储在下位机中。



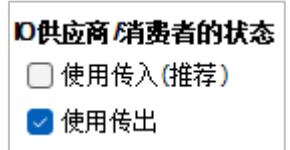
使用项目参数（推荐）：使用项目的 IP 设置和站名，站名：从站名称 Profinet 设备。

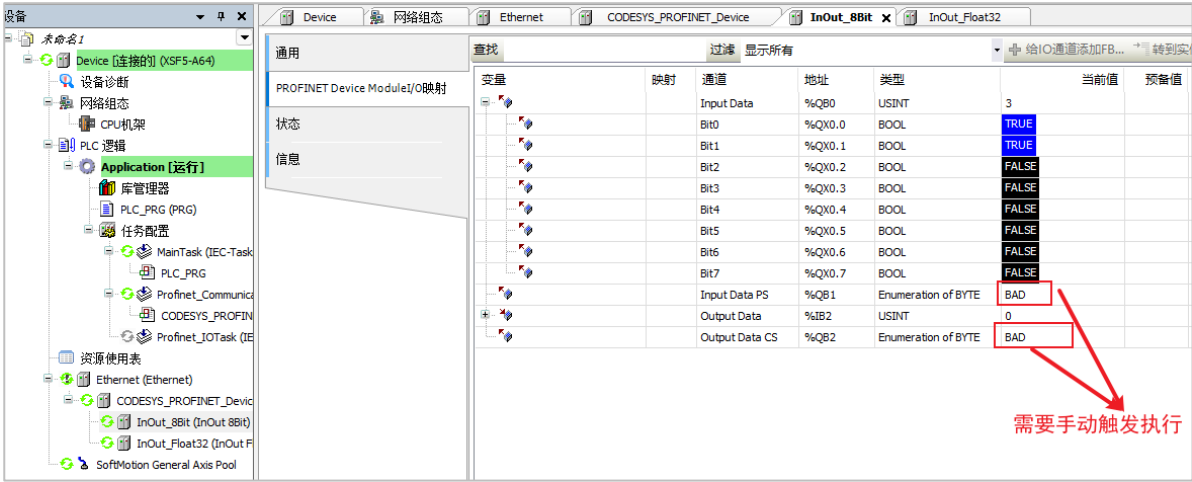
2、IO 供应商和消费者的状态

- 使用传入：无需手动控制状态。



- 使用传出：此时可以手动控制 input Data PS 和 output Data CS 状态进行通讯。

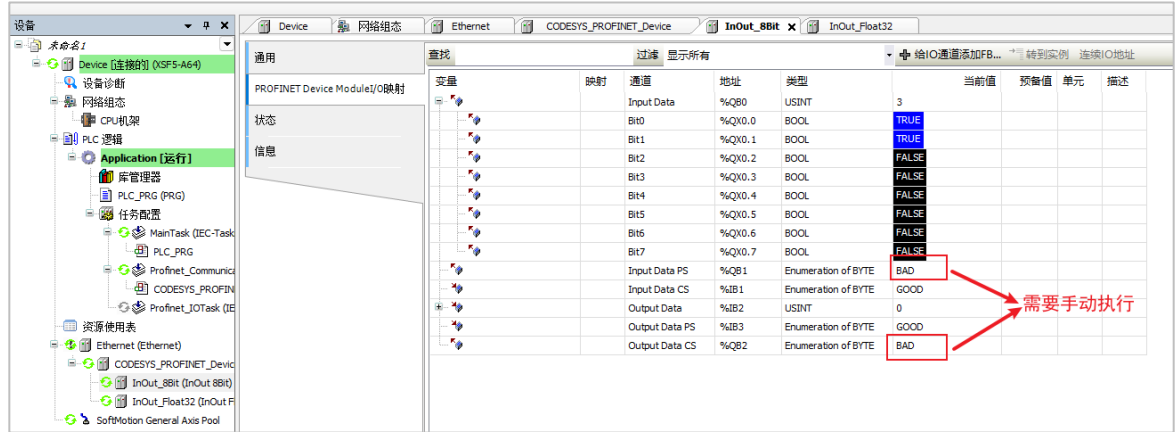




- 两个都选：此时可以手动控制 input Data PS 和 output Data CS 状态进行通讯。

IO 供应商/消费者的状态

- ☒ 使用传入 (推荐)
- ☒ 使用传出



3、Substitute values (代替值)

Inactive (不活跃)：当主站停止后，从站接收变为 0。
Last value (最后一个值)：当主站停止后，从站数据会保留为最后一个值。

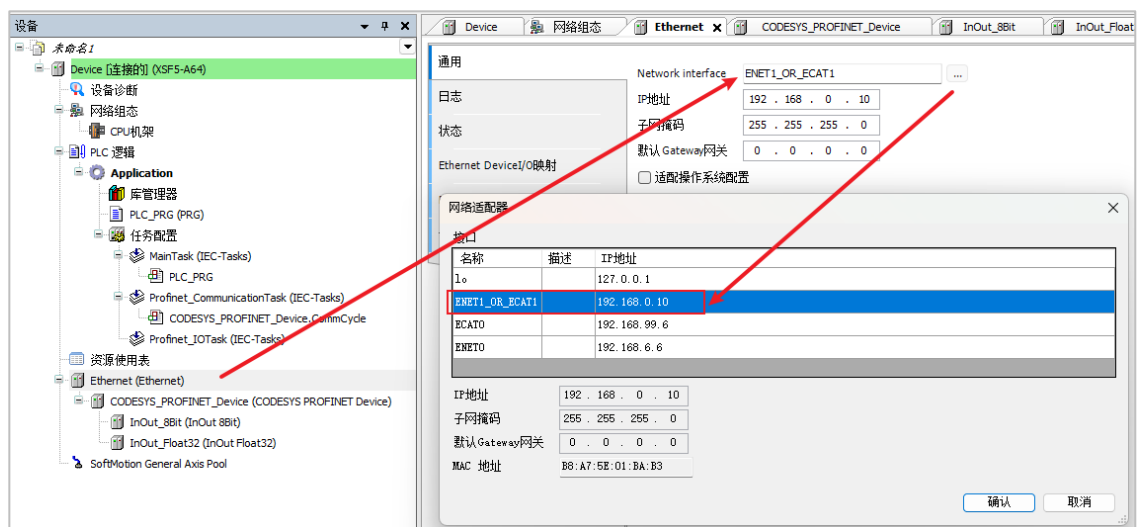
3-7-5. PROFINET 样例

3-7-5-1. 西门子 S7-200SMART 使用

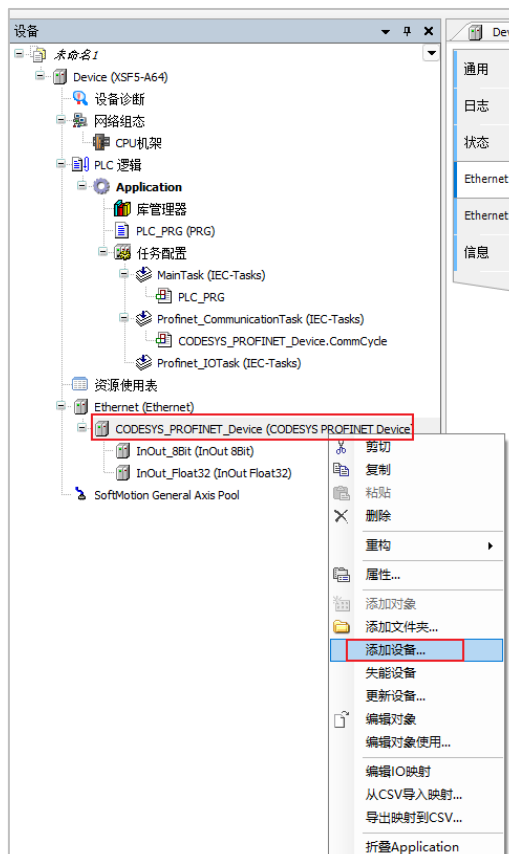
例：使用信捷 XSF5-A64（从站）和西门子 S7-200SMART（主站）进行 PROFINET 通讯。

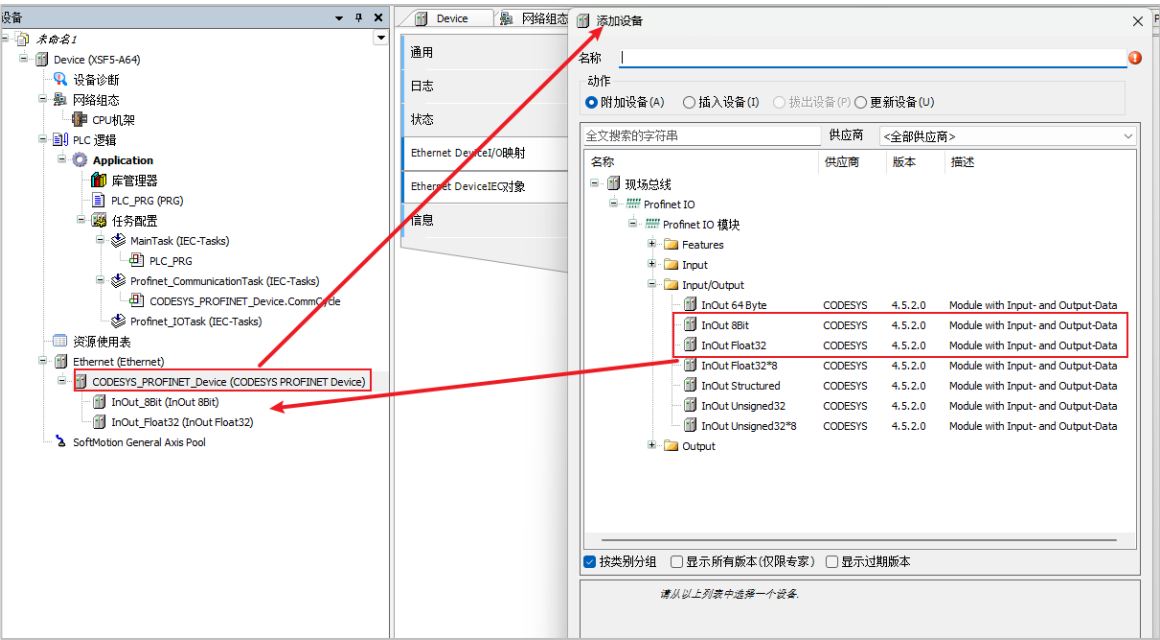
■ 信捷 XSF5-A64（从站）配置

- 1、多功能网口选择“profinet”并选择网口 1 为 profinet 从站设备后，在生成的 ethernet 中选择相应网口的 IP 地址；

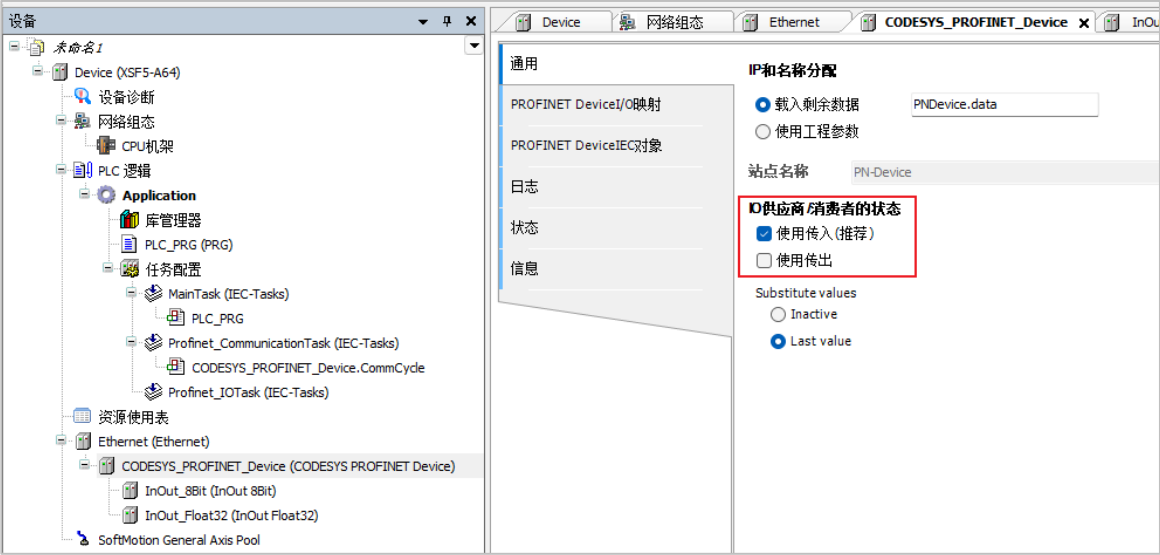


- 2、右击“CODESYS_PROFNET_Device”添加通讯子模块（以 8bit 和 float32 为例）；

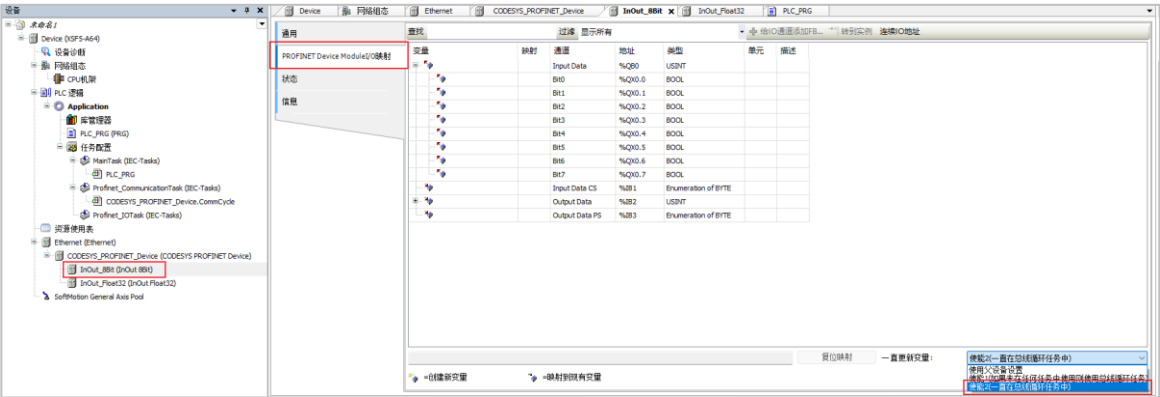




3、IO 供应商和消费者的状态选择“使用传入”；

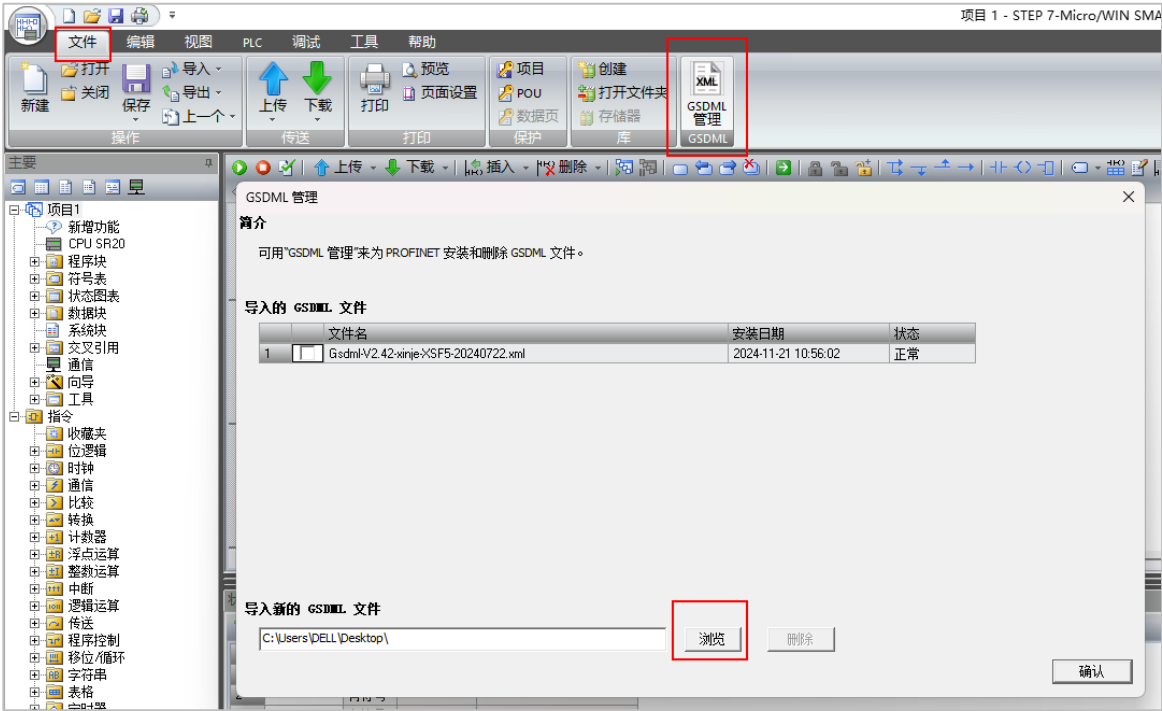


4、通讯子模块的 PROFINET Device Module/o 映射选择：使能 2。

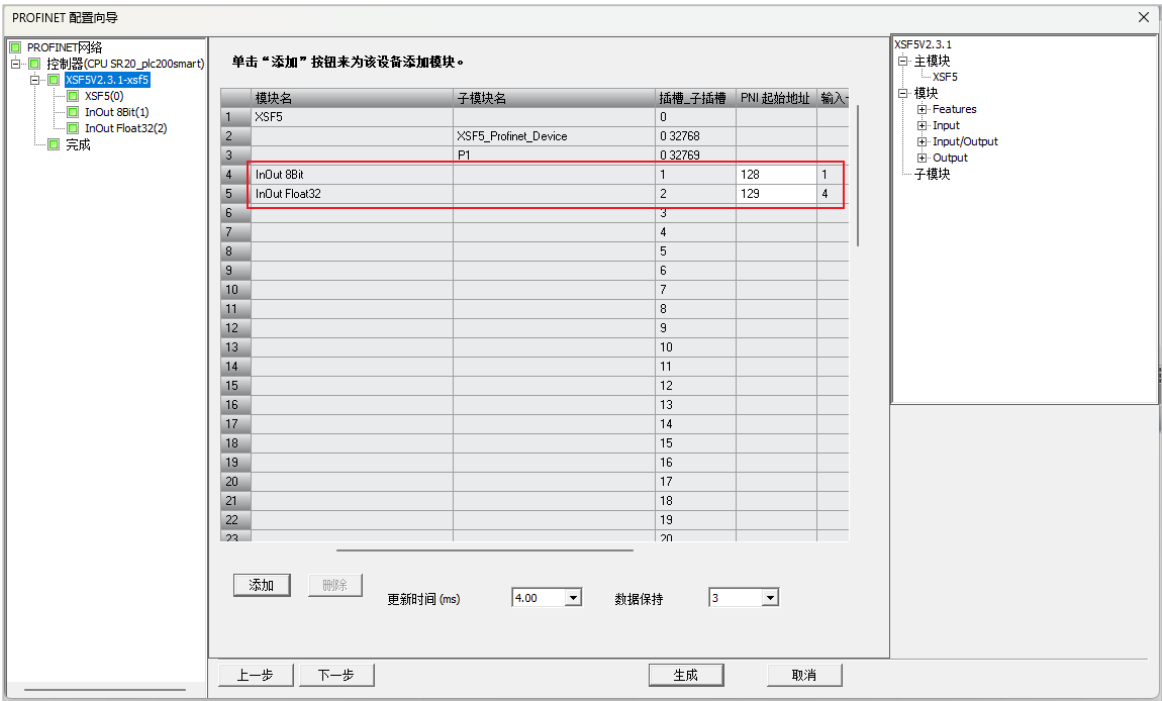


■ 西门子 S7-200SMART（主站）配置

1、 导入 XSF5 的 GSDML 文件至工程中。打开西门子 STEP 7-MicroWIN SMART 软件。在“文件”菜单下，点击 GSDML 管理，在弹出的窗口，点击浏览找到 XSF5 的 GSDML 文件点击打开，完成 GSDML 文件的安装。



2、 配置从站信息以及通讯子模块；



3、添加监控通讯地址。

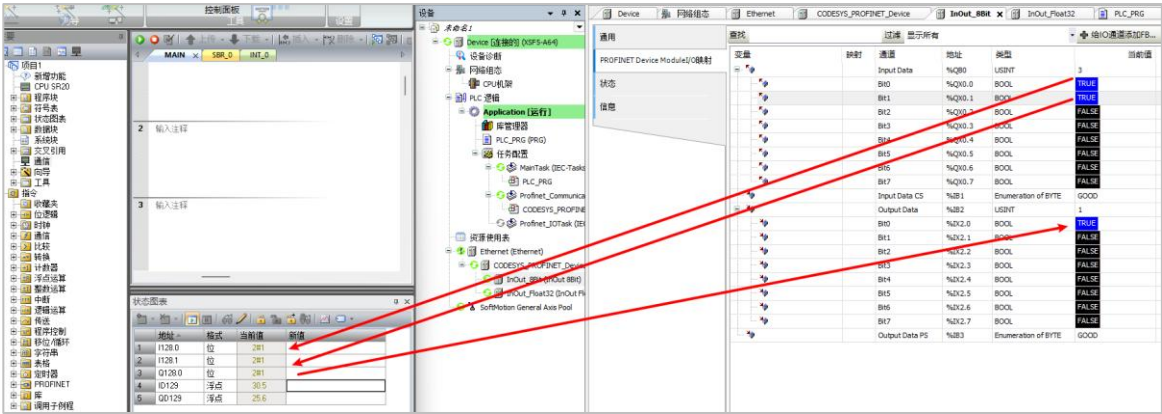
状态图表			
地址	格式	当前值	新值
1 I128.0	位	2#1	
2 I128.1	位	2#0	
3 Q128.0	位	2#0	
4 ID129	浮点	0.0	
5 QD129	浮点	0.0	

此时：主站的 I128.0 与从站通讯子模块中 InOut_8bit 的 Input Data 中的第一个 BOOL 对应，以此类推。

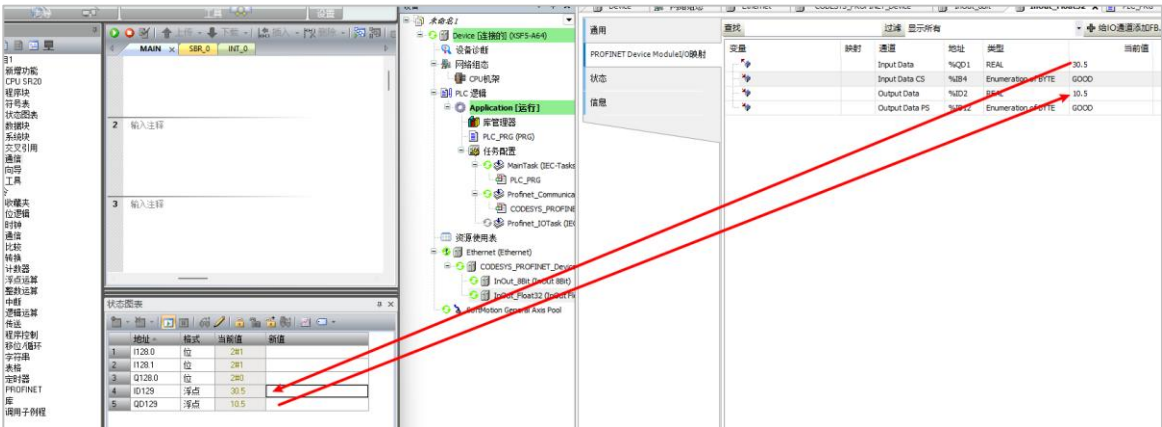
主站的 ID129 与从站通讯子模块中 InOut_Float32 的 Input Data 中的 REAL 对应。

结果如下：

当置位从站的通讯子模块下面的 Input Data 的 BOOL 时，监控主站的 I 地址会被置位。同理，置位主站的 Q 地址则从站的通讯子模块下面的 Output Data 的 BOOL 会被置位。



当给从站的通讯子模块下面的 Input Data 的 REAL 数值时，监控主站的 ID 地址会有相应的数值显示。同理，给主站的 QD 地址数值则从站的通讯子模块下面的 Output Data 的 REAL 会有相应的数值显示。

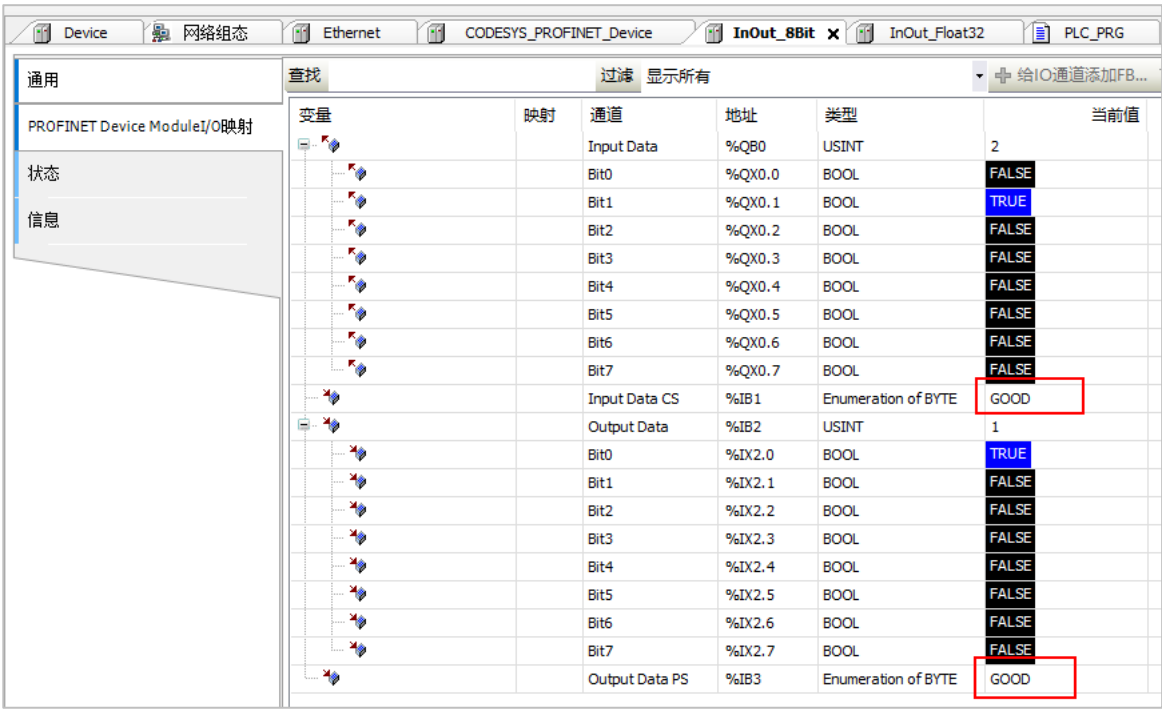


■ 检查通讯状态

C 和 P 代表提供者（Consumer）和消费者（Producer），S 表示状态。

GOOD：标志通讯正常

若 IO 供应商和消费者的状态选择“使用传出”，则需手动更改状态“BAD”->“GOOD”即可通讯。

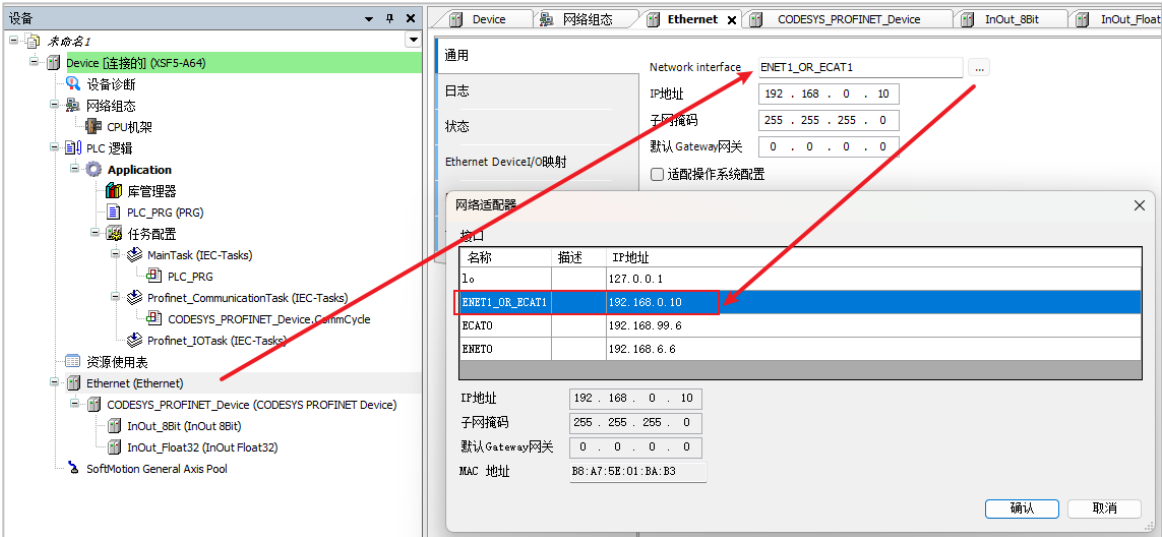


3-7-4-2 西门子 S7-1200/1500 使用

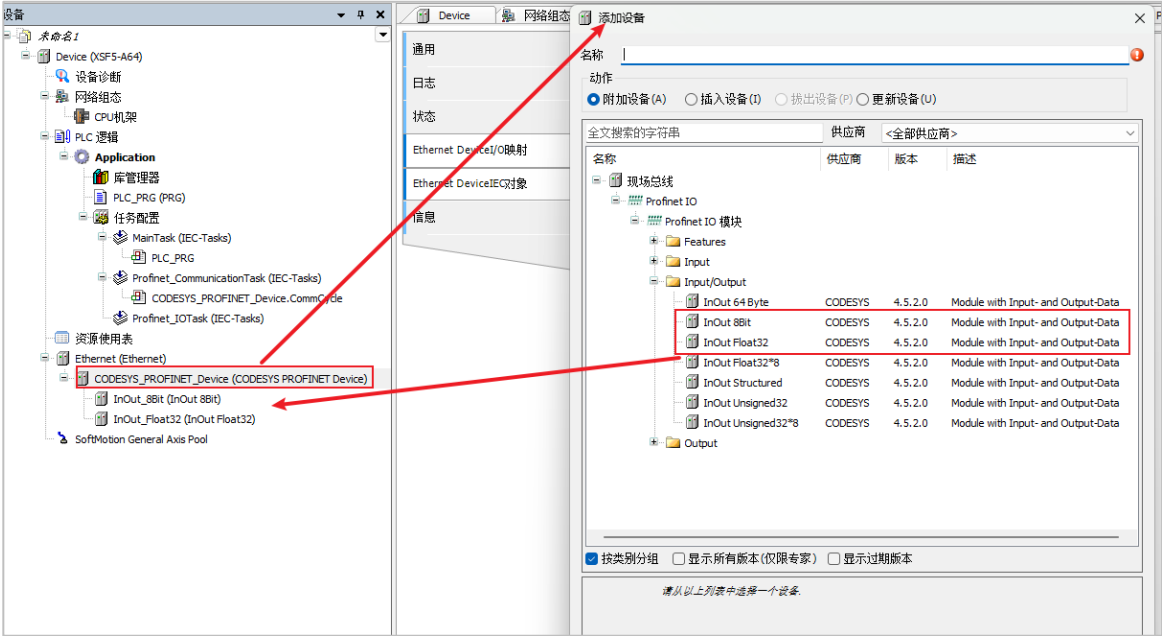
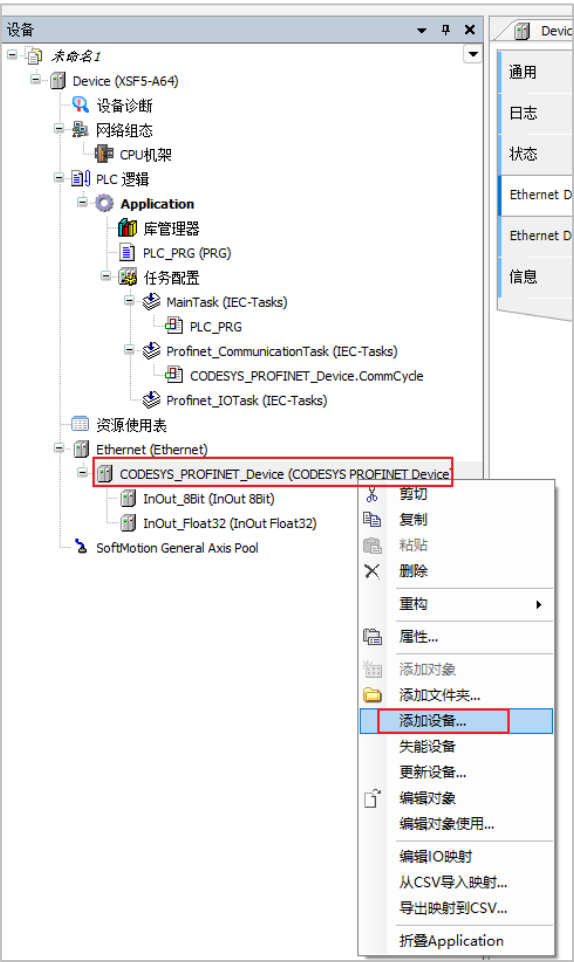
例：使用信捷 XSF5-A64（从站）和西门子 S7-1200（主站）进行 PROFINET 通讯。

■ 信捷 XSF5-A64（从站）配置

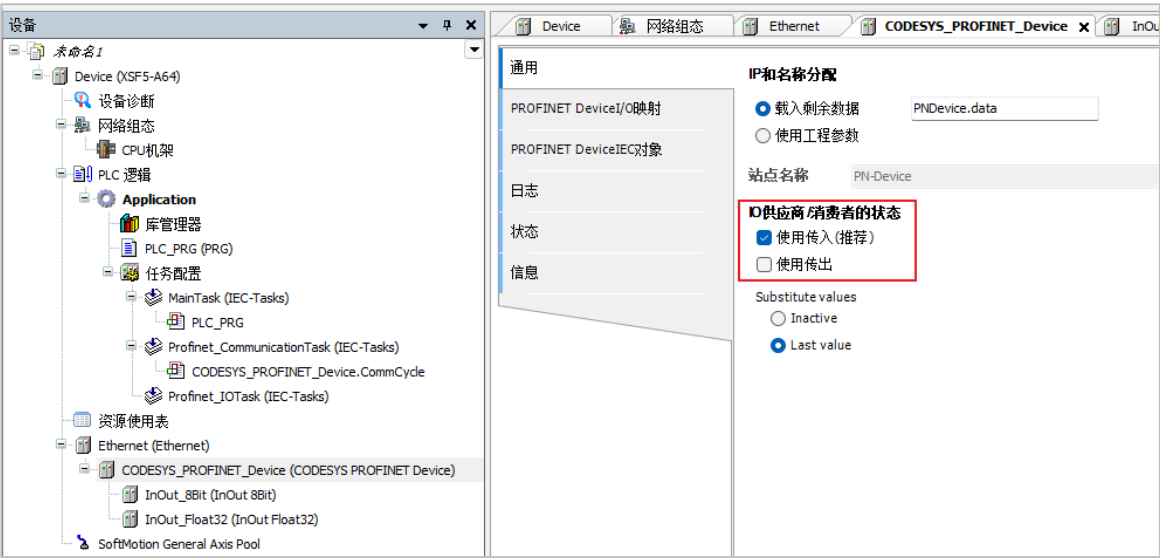
1、多功能网口选择“profinet”并选择网口 1 为 profinet 从站设备后，在生成的 ethernet 中选择相应网口的 IP 地址；



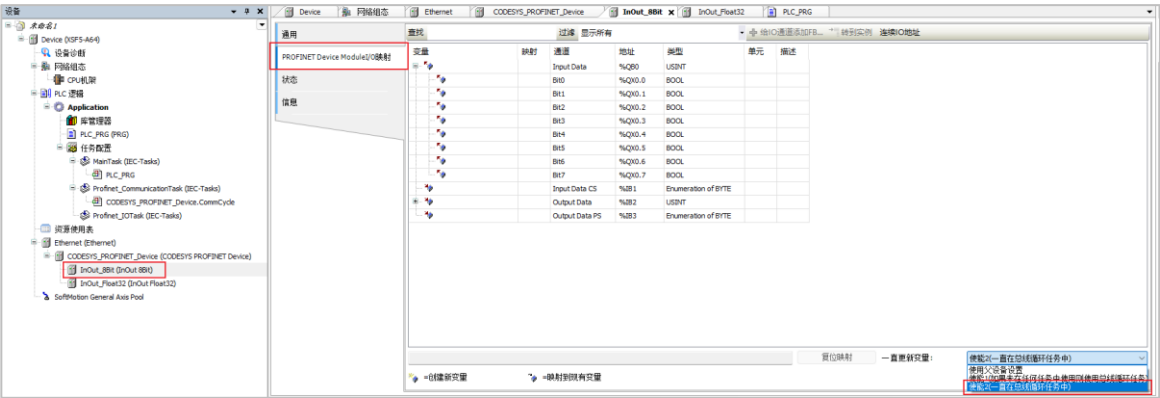
2、右击“CODESYS_PROFINET_Device”添加通讯子模块（以 8bit 和 float32 为例）；



3、IO 供应商和消费者的状态选择“使用传入”；



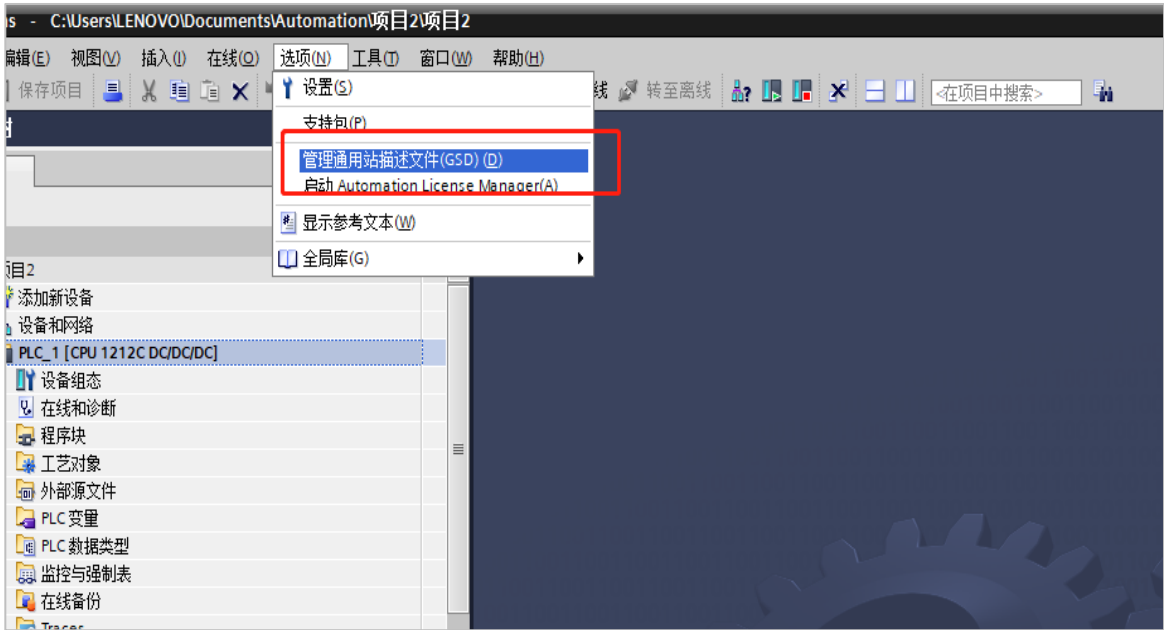
4、通讯子模块的 PROFINET Device ModuleI/O 映射选择：使能 2。



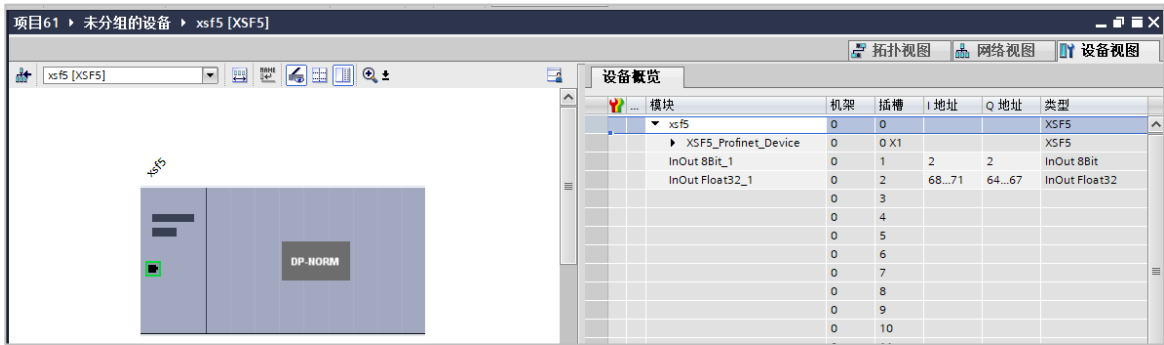
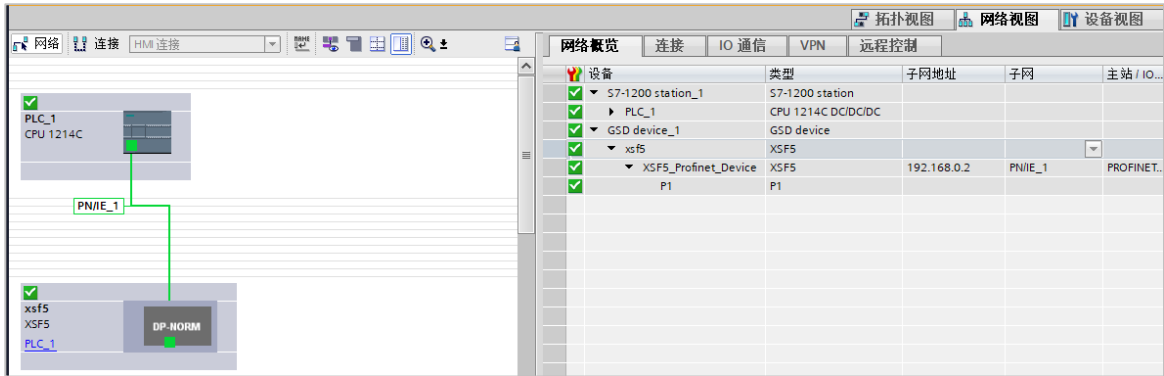
■ 西门子 S7-1200（主站）配置

1、导入 XSF5 的 GSDML 文件至工程中；

在菜单栏“选型”中，点击“管理通用站描述文件（GSD）”，添加 XSF5 的 GSD 文件。在源路径中选择 GSD 文件所在文件夹，选择对应的 GSD 文件，点击安装。



2、配置从站信息以及通讯子模块；



3、添加监控通讯地址。

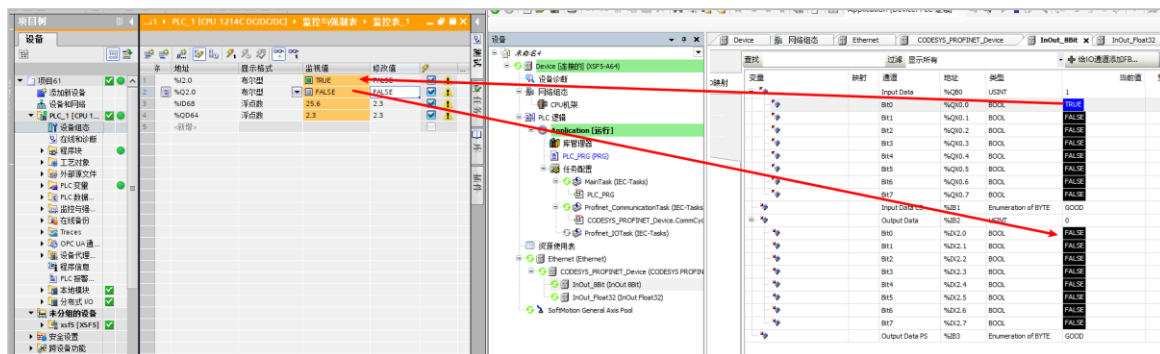
	名称	地址	显示格式	监视值	修改值	注释	变量注释
1		%I2.0	布尔型		FALSE	Y0端子输出状态	
2		%Q2.0	布尔型		TRUE		
3		%ID68	浮点数		2.3		
4		%QD64	浮点数		2.3		
5		<新增>					

此时：主站的 I2.0 与从站通讯子模块中 InOut_8bit 的 Input Data 中的第一个 BOOL 对应，以此类推。

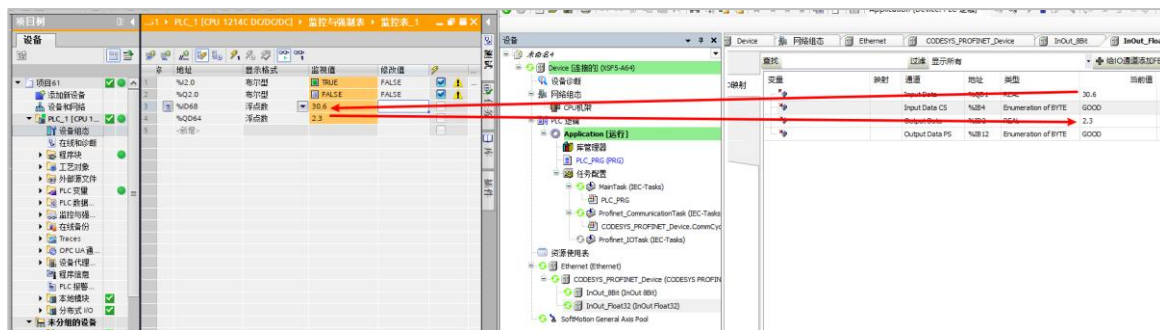
主站的 ID68 与从站通讯子模块中 InOut_Float32 的 Input Data 中的 REAL 对应。

结果如下：

当置位从站的通讯子模块下面的 Input Data 的 BOOL 时，监控主站的 I 地址会被置位。同理，置位主站的 Q 地址则从站的通讯子模块下面的 Output Data 的 BOOL 会被置位。



当给从站的通讯子模块下面的 Input Data 的 REAL 数值时，监控主站的 ID 地址会有相应的数值显示。同理，给主站的 QD 地址数值则从站的通讯子模块下面的 Output Data 的 REAL 会有相应的数值显示。

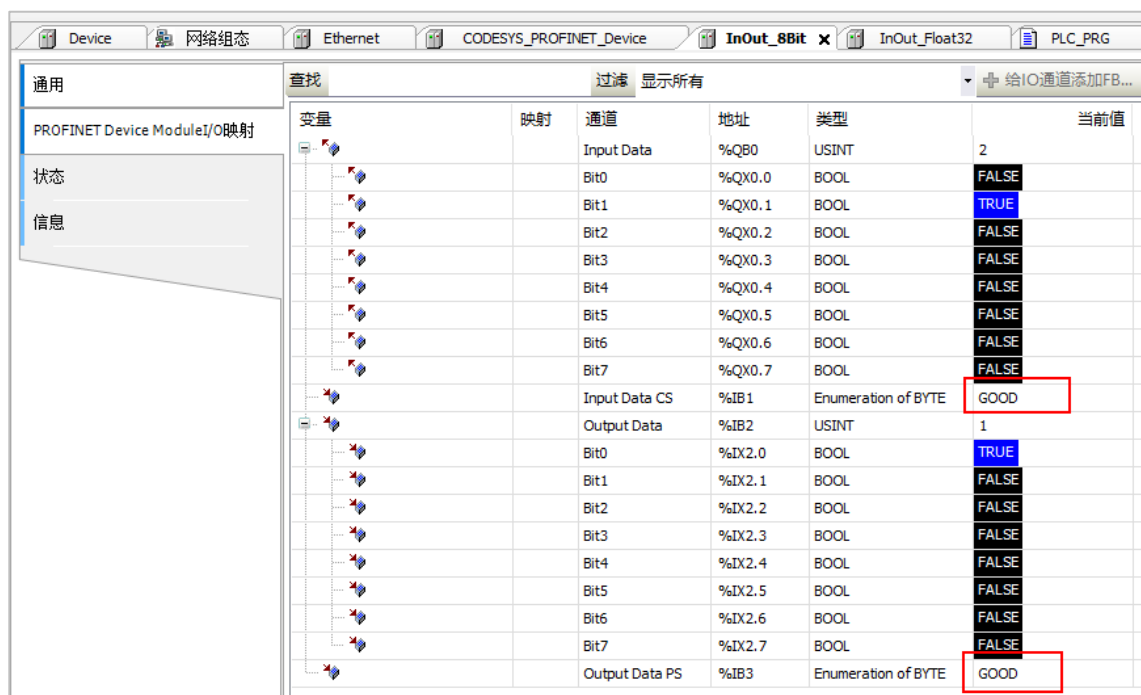


■ 检查通讯状态

C 和 P 代表提供者（Consumer）和消费者（Producer），S 表示状态。

GOOD：标志通讯正常

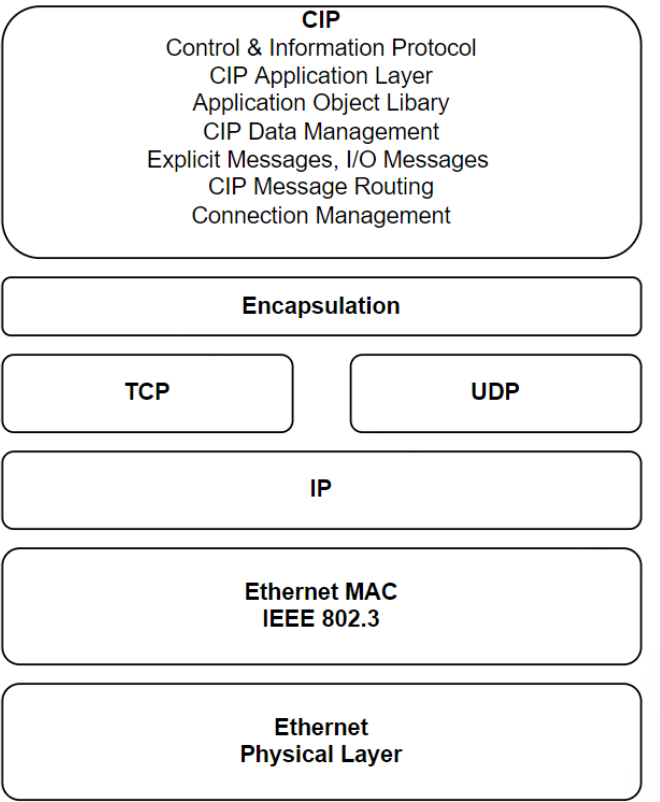
若 IO 供应商和消费者的状态选择“使用传出”，则需手动更改状态“BAD”->“GOOD”即可通讯。



4. EtherNet/IP 通讯

4-1. EtherNet/IP 概要

Ethernet/IP 是一个面向工业自动化应用的工业应用层协议。是 2000 年 3 月由 Control Net International 和 ODVA（Open DeviceNet Vendors Association）共同开发的工业以太网标准。它建立在标准 UDP/IP 与 TCP/IP 协议之上，利用固定的以太网硬件和软件，为配置、访问和控制工业自动化设备定义了一个应用层协议。各层结构如图所示：



Ethernet/IP 实现实时性的方法是在 TCP/IP 层之上增加了用于实时数据交换和运行实时应用的 CIP 协议（Common Industrial Protocol）。

EtherNet/IP 协议的技术特点：

- ◆ EtherNet/IP 实现实时性的方法是在 TCP/IP 层之上增加了用于实时数据交换和运行实时应用的 CIP 协议(Common Industrial Protocol)。
- ◆ EtherNet/IP 在物理层和数据链路层采用标准的以太网技术，在网络层和传输层使用 IP 协议和 TCP、UDP 协议来传输数据。UDP 是一种非面向连接的协议，它能够工作在单播和多播的方式，只提供设备间发送数据报的能力。对于实时性很高的 I/O 数据、运动控制数据和功能行安全数据，使用 UDP/IP 协议来发送。而 TCP 是一种可靠的、面向连接的协议。对于实时性要求不是很高的数据(如参数设置、组态和诊断等)采用 TCP/IP 协议来发送。
- ◆ EtherNet/IP 采用生产者/消费者数据交换模式。生产者向网络中发送有唯一标识符的数据包。消费者根据需要通过标识符从网络中接收需要的数据。这样数据源只需一次性地把数据传到网上，其它节点有选择地接收数据，这样提高了通信的效率。

Ethernet/IP 是在 CIP 这个协议的控制下实现非实时数据和实时数据的传输。CIP 是一个提供工业设备端到端的面向对象的协议，且独立于物理层及数据链路层，这使得不同供应商提供的设备能够很好的交互。另外，为了获得更好的时钟同步性能，2003 年 ODVA 将 IEEE 1588 引入 Ethernet/IP，并制定了 CIPsync 标准以提高 Ethernet/IP 的时钟同步精度。

4-2. EtherNet/IP 隐式通讯

4-2-1. EtherNet/IP 作为从站的样例



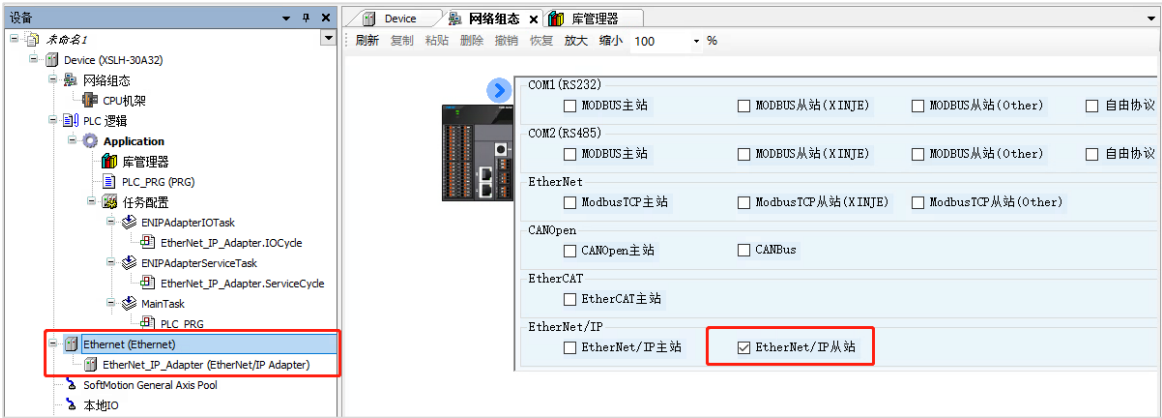
在 XSA-W 工控机桌面安装并运行上位机通讯软件时，内部通讯 IP 需按机型区分配置：

1、XSA330-W、XSA500-W 系列工控机：
设备内部通讯统一采用回送地址 127.0.0.1。

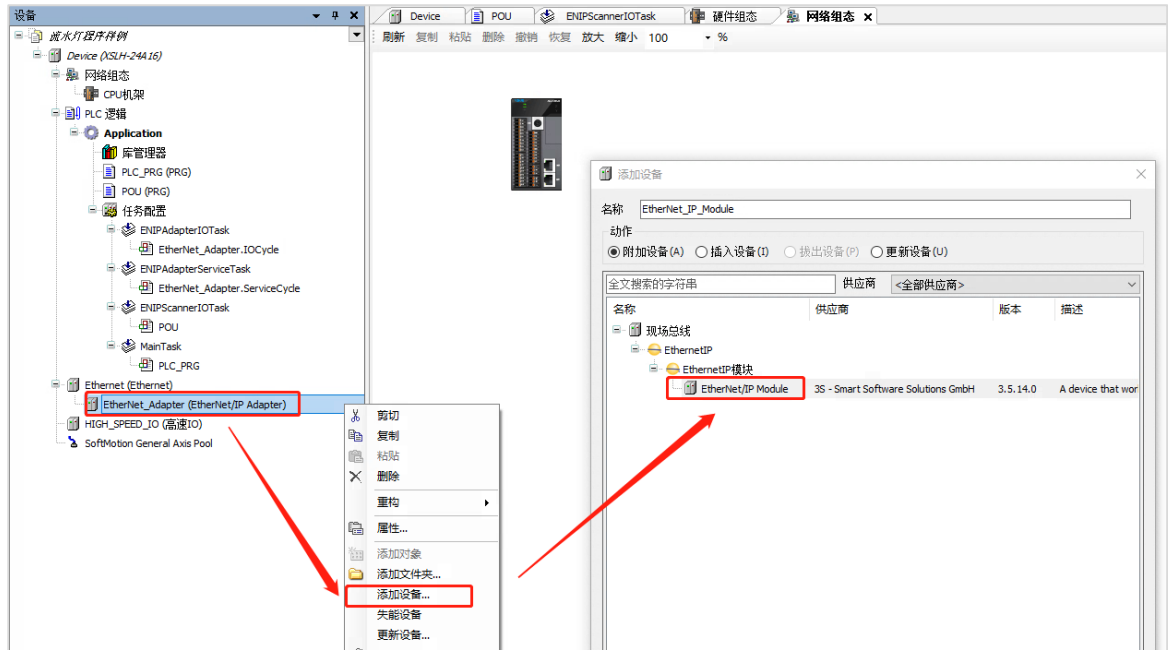
2、XSA230-W、XSA620-W 支持虚拟化的工控机：

- ①内部通讯目标 IP 固定为 192.168.2.2；
- ②本机虚拟网卡默认 IP 为 192.168.2.1，可连接显示器查看；
- ③若用户需要修改虚拟网卡的 IP，则仅支持修改为 192.168.2 网段的 IP 地址，不可更换为其他网段。

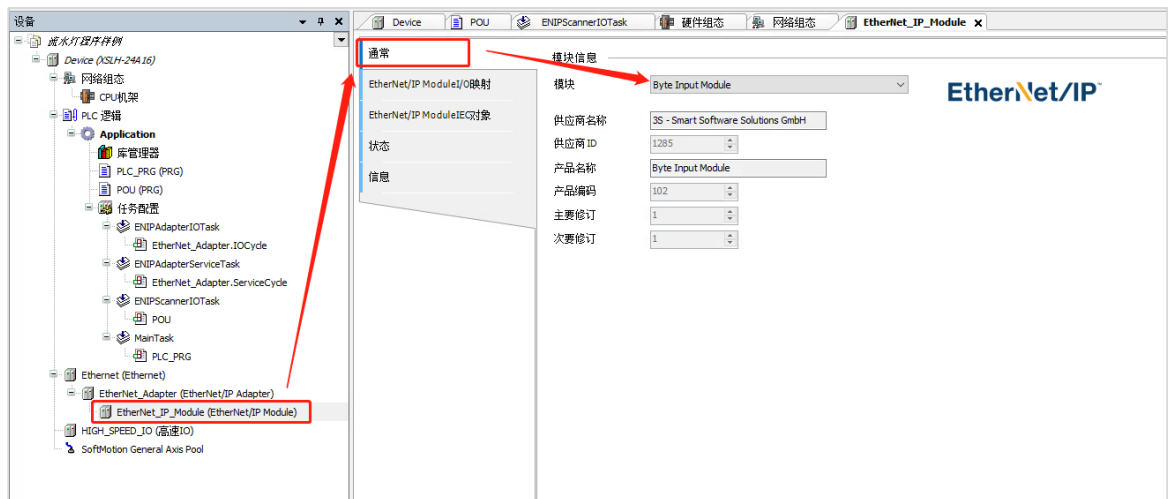
1、鼠标单击网络组态内的使能窗口，勾选“EtherNet IP 从站”。左侧设备树节点下会自动添加“EtherNet/IP Adapter”。

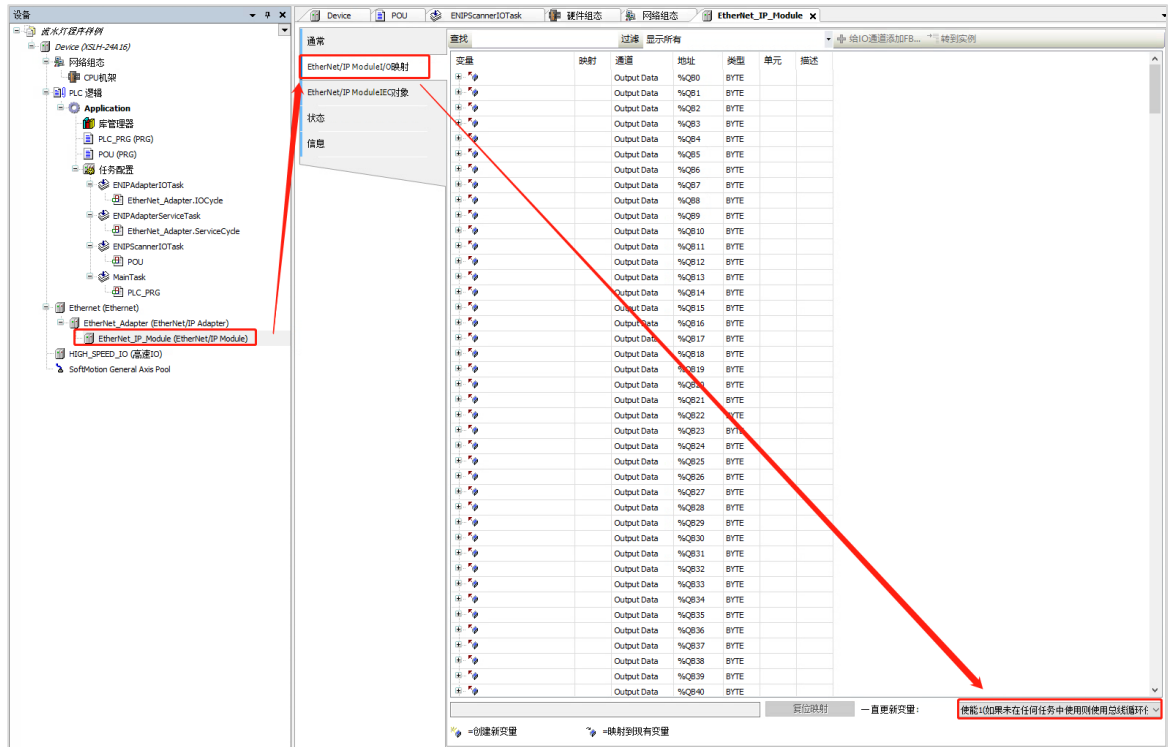


2、右击“EtherNet/IP Adapter”→添加“EtherNet/IP Module”。

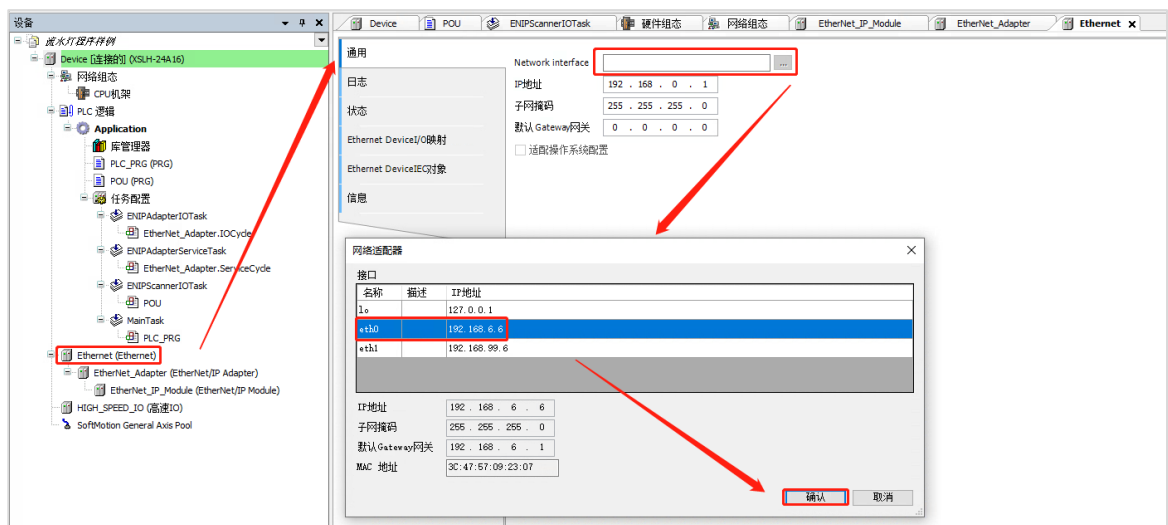


3、添加想要访问的数据类型用来测试。

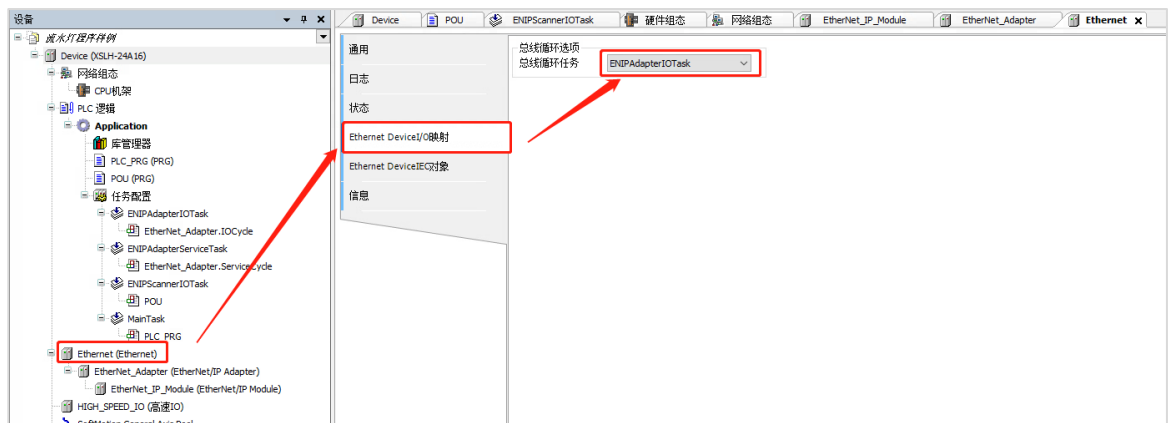




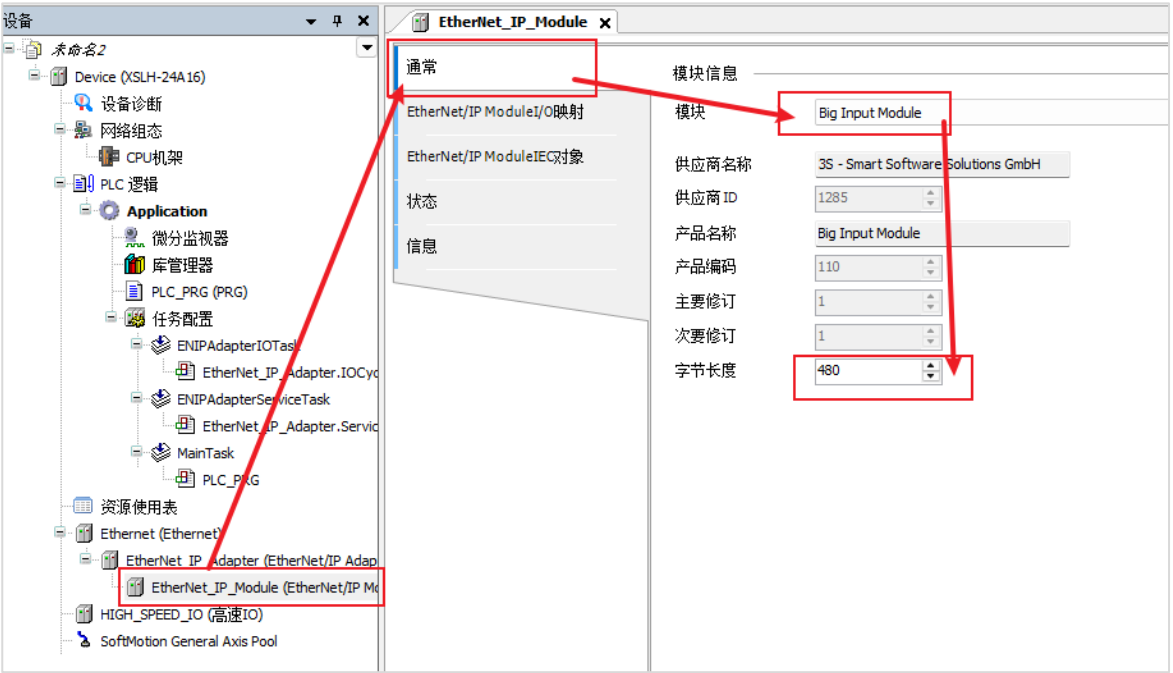
4、选择用于 Ethernet IP 通讯的网络接口。



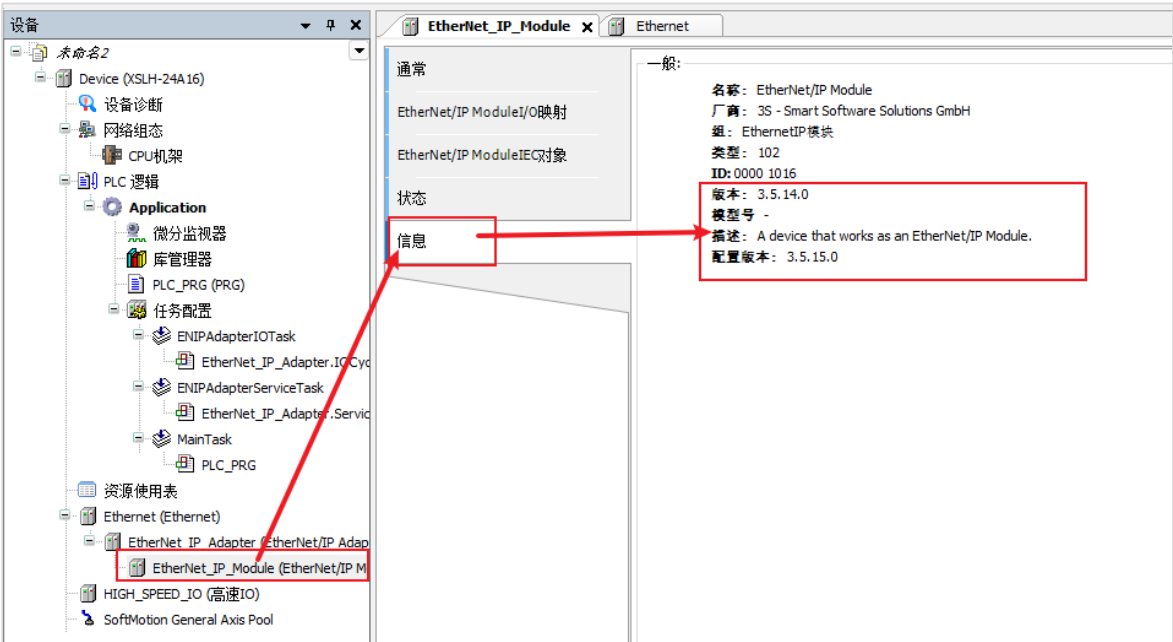
5、总线循环设置改为“ENIPAdapterIOTask”：



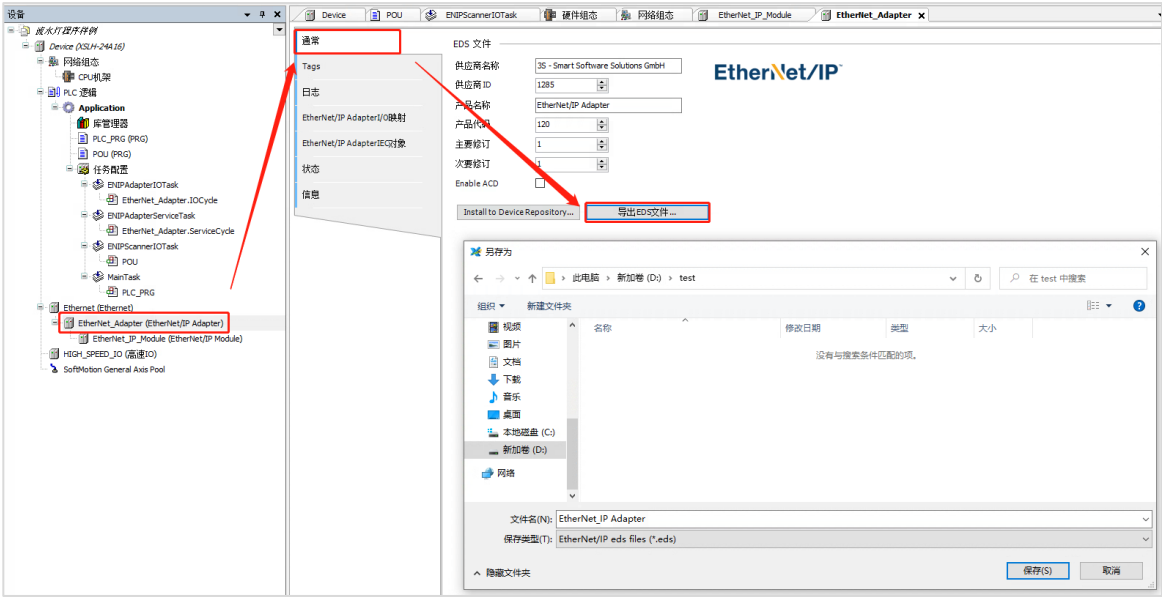
6、当 EIP 模块的模式选择 Big Input module 或 Big Output module 时，在“通常”配置页面会出现字节长度的设置框。字节长度默认是 480，最小值是 1。



7、上述功能仅在 XS 自带的 EIP 版本支持。



8、添加完通讯的数据类型之后导出 eds 文件。



4-2-2. EtherNet/IP 作为主站的样例



在 XSA-W 工控机桌面安装并运行上位机通讯软件时，内部通讯 IP 需按机型区分配置：

1、XSA330-W、XSA500-W 系列工控机：

设备内部通讯统一采用回送地址 127.0.0.1。

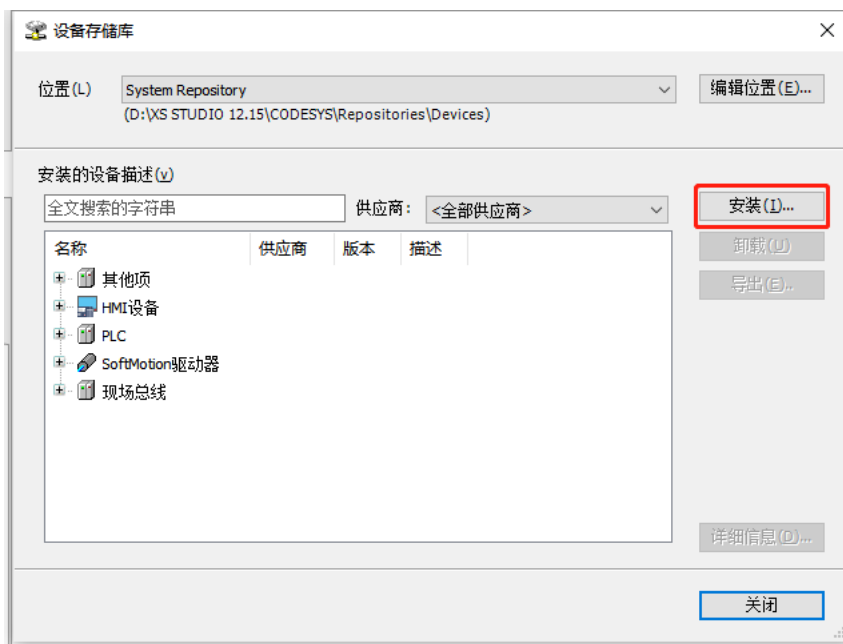
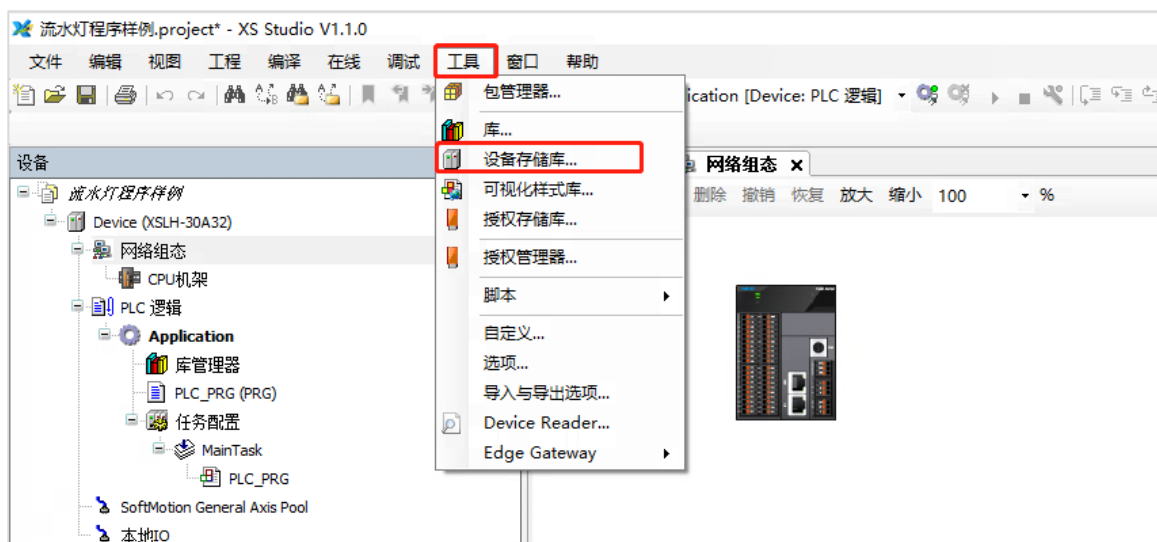
2、XSA230-W、XSA620-W 支持虚拟化的工控机：

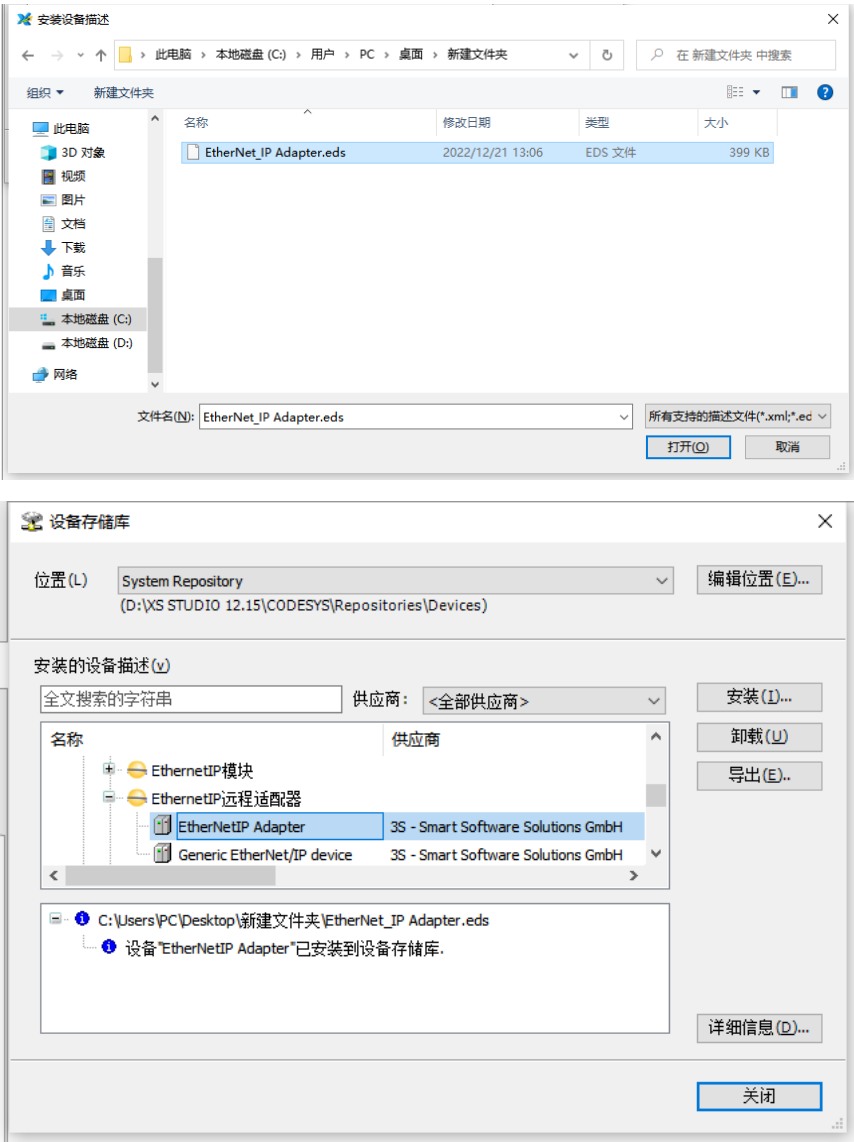
①内部通讯目标 IP 固定为 192.168.2.2；

②本机虚拟网卡默认 IP 为 192.168.2.1，可连接显示器查看；

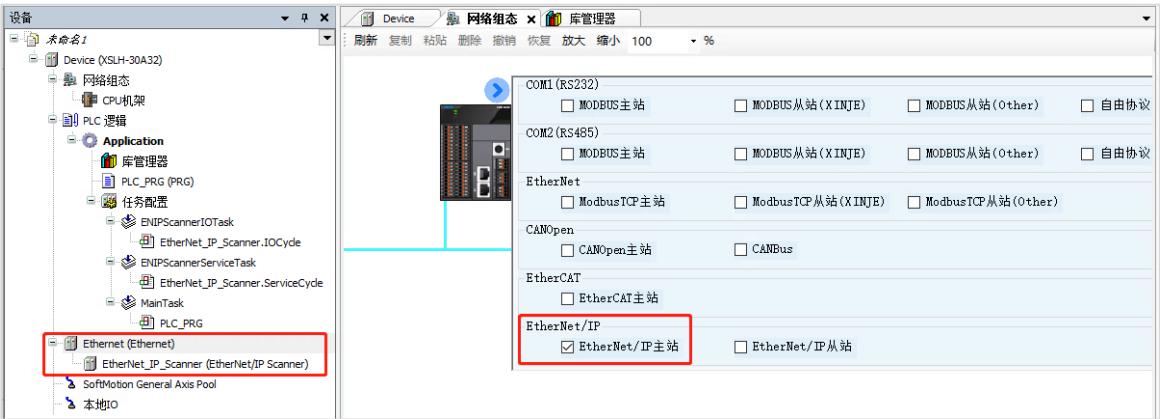
③若用户需要修改虚拟网卡的 IP，则仅支持修改为 192.168.2 网段的 IP 地址，不可更换为其他网段。

1、 工具→设备存储库→安装→打开刚才导出的 EDS 文件→如图，添加完成。

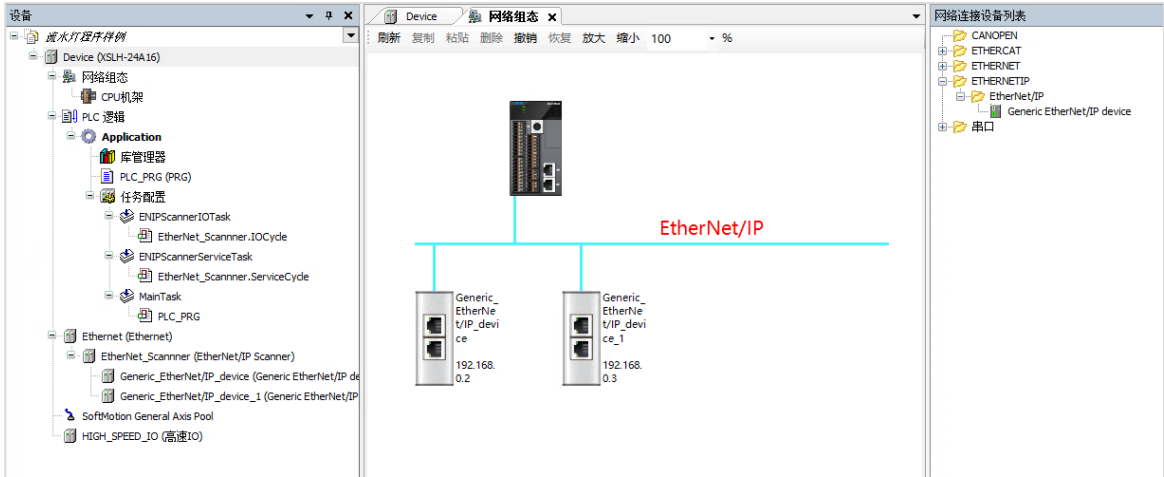




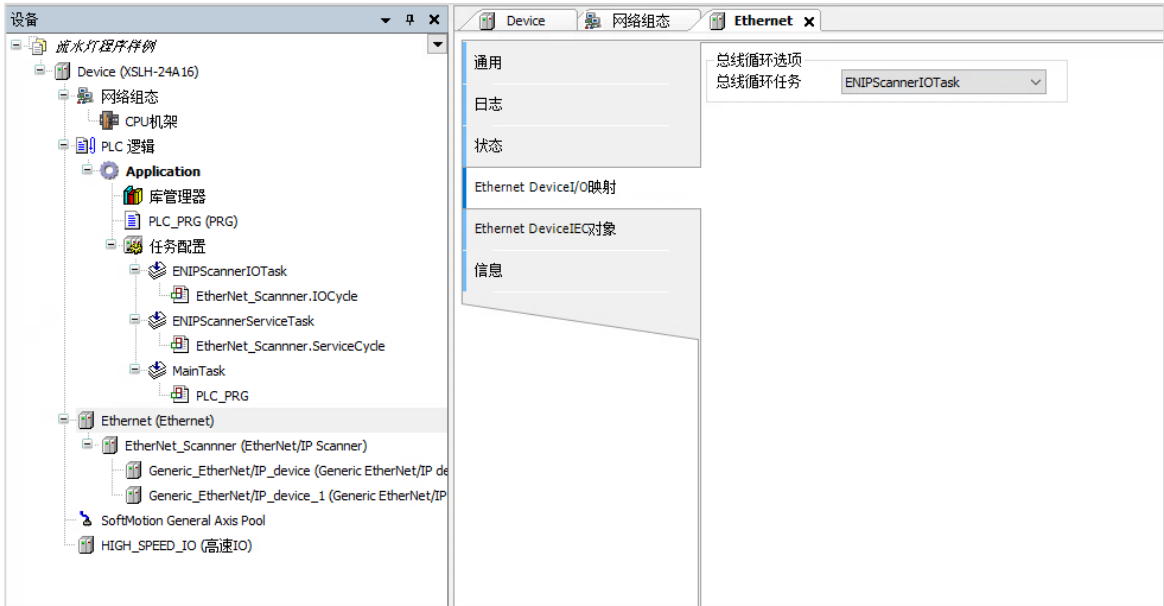
2、鼠标单击网络组态内的使能窗口，勾选“EtherNet IP 主站”。



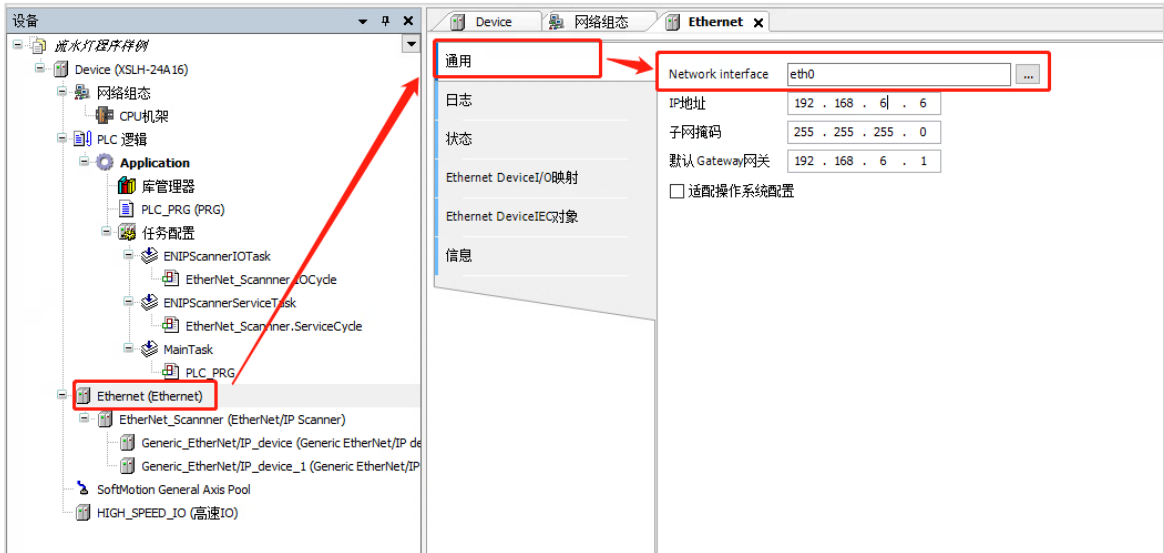
3、从右侧“网络连接设备列表”添加从站设备，如图所示：



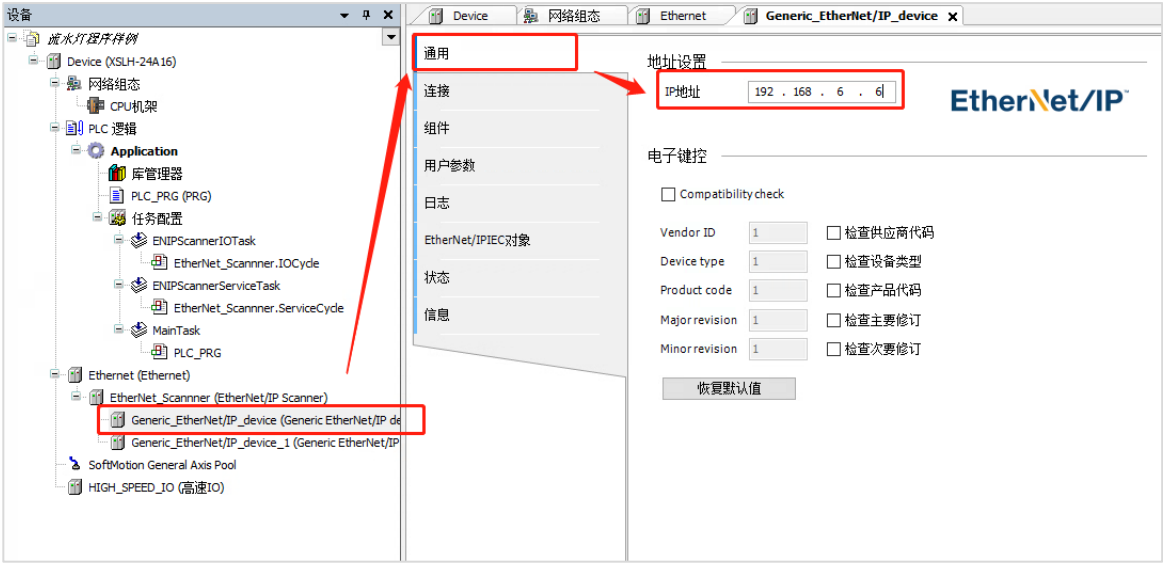
4、将总线循环任务设置为统一的任务配置：



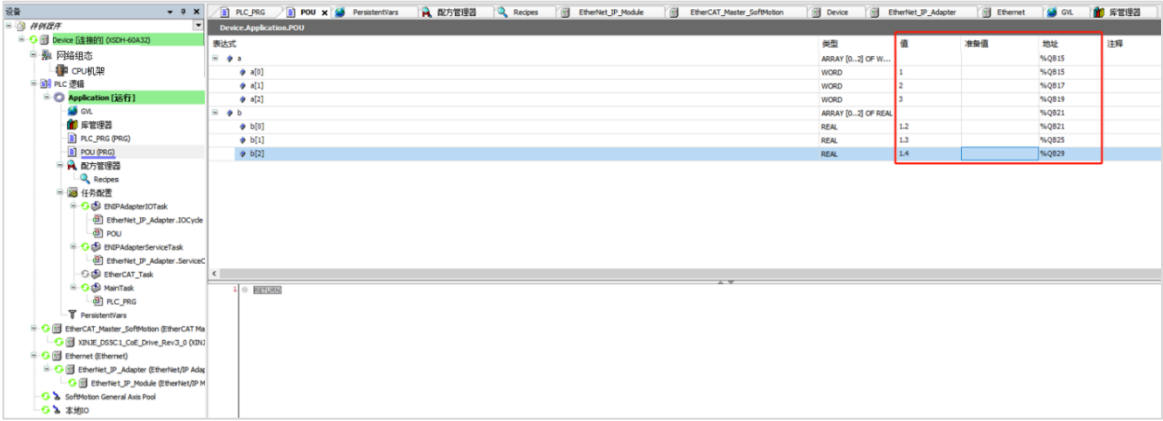
5、选择主站网口：



6、设置从站 IP:



7、通讯测试：定义并关联变量，将程序下载进 PLC 中，此时通讯成功。



4-3. EtherNet/IP 显式通讯

4-3-1. 功能介绍

为了支持 MES、HMI 等客户端通过显示通讯的方式获取 PLC 的数据，全局变量列表增加了 EIP 标签功能。客户端可以通过标签通讯的连接方式读写网络公开的全局变量。



目前仅支持作为 EtherNet/IP 显式通讯的从站。

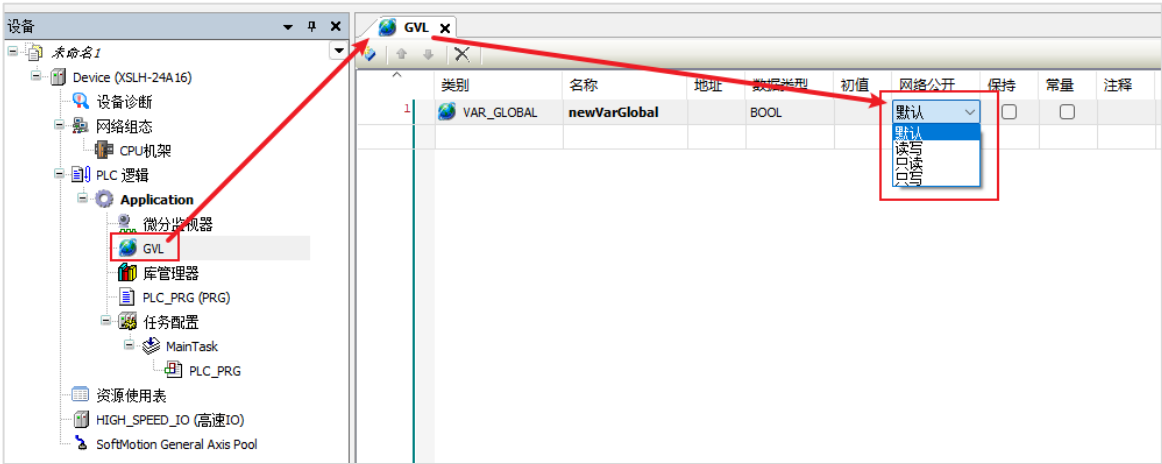
全局变量 EIP 标签支持的数据说明如下：

基础数据类型			
整型	SINT	浮点型	REAL
	INT		LREAL
	DINT	位类型	BYTE
	LINT		WORD
	USINT		DWORD
	UINT		LWORD
	UDINT		BOOL
	ULINT		STRING

类型	维度	成员
数组	一维	基础类型
		结构体
	二维	基础类型
		结构体

类型	成员	说明
结构体	基础类型	—
	结构体	最多支持两层结构体嵌套

4-3-2. EIP 标签全局变量表定义

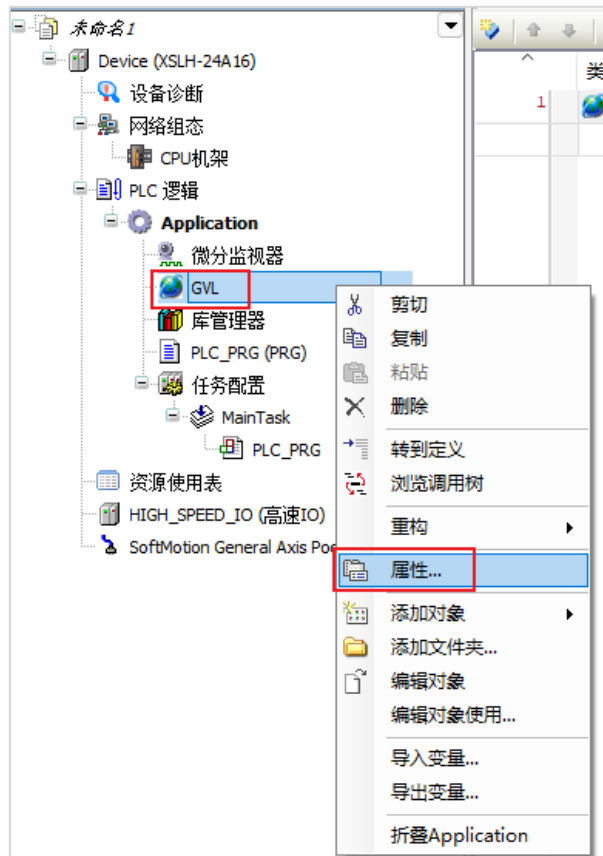


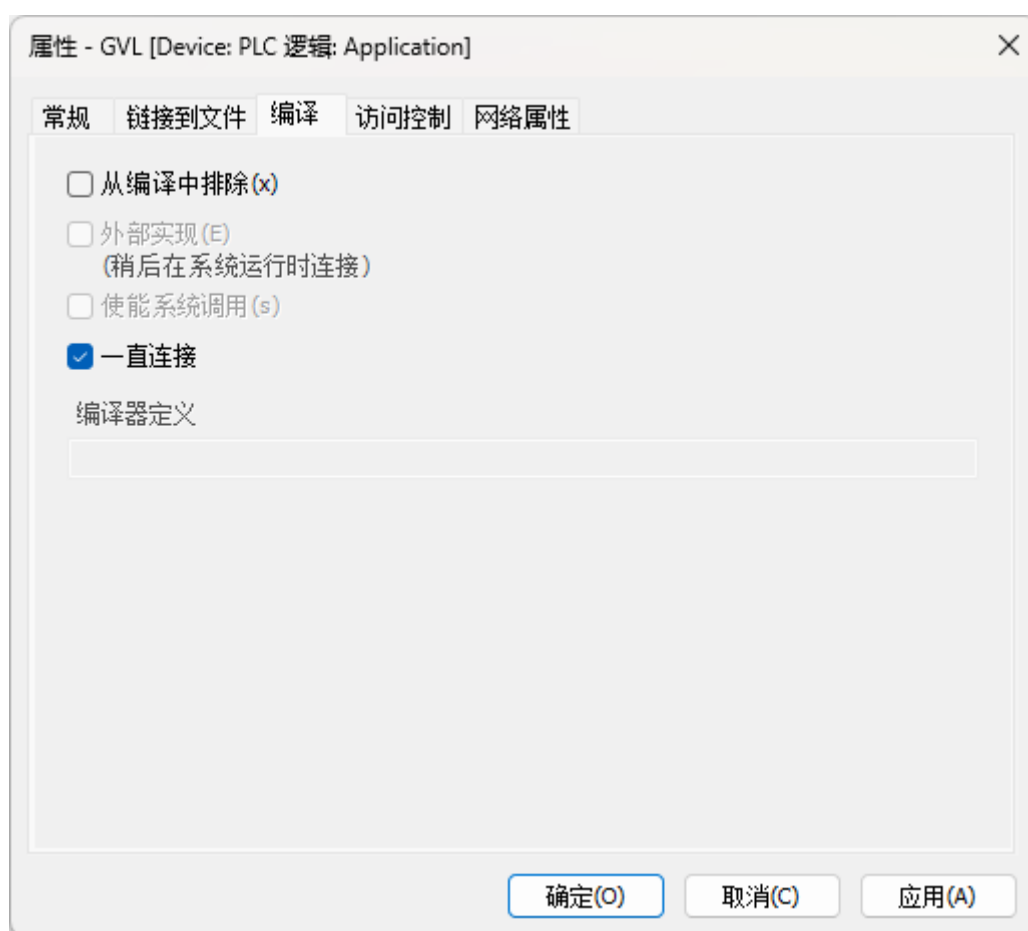
- “默认”表示该变量不作为标签，仅在本工程中使用；
- “读写”表示该变量作为标签且可读可写；
- “只读”表示该变量作为标签只可读；
- “只写”表示该变量作为标签只可写。



对网络属性为“只读”的标签进行写请求会返回失败，同理，对网络属性为“只写”的标签进行读请求时也会返回失败。

进行通讯时需要在全局变量表的属性中勾选“一直连接”。

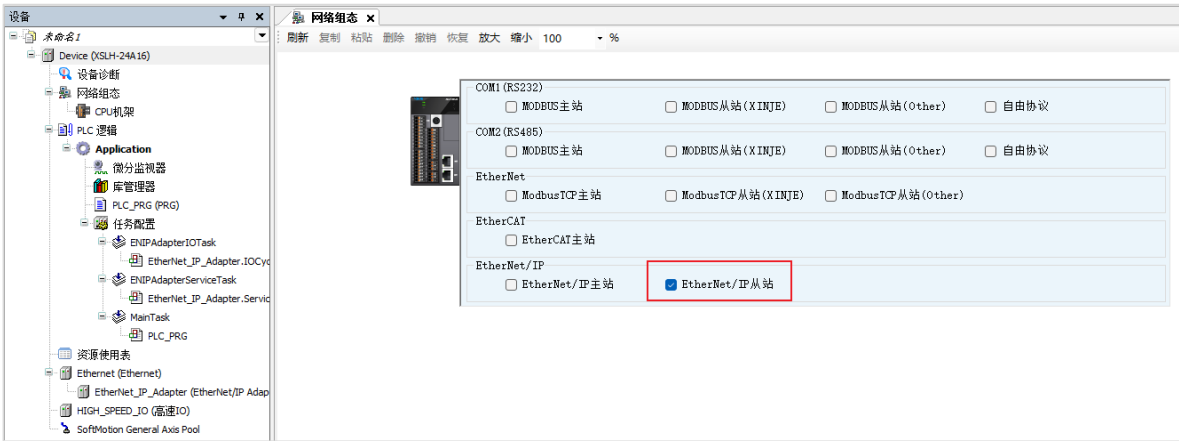




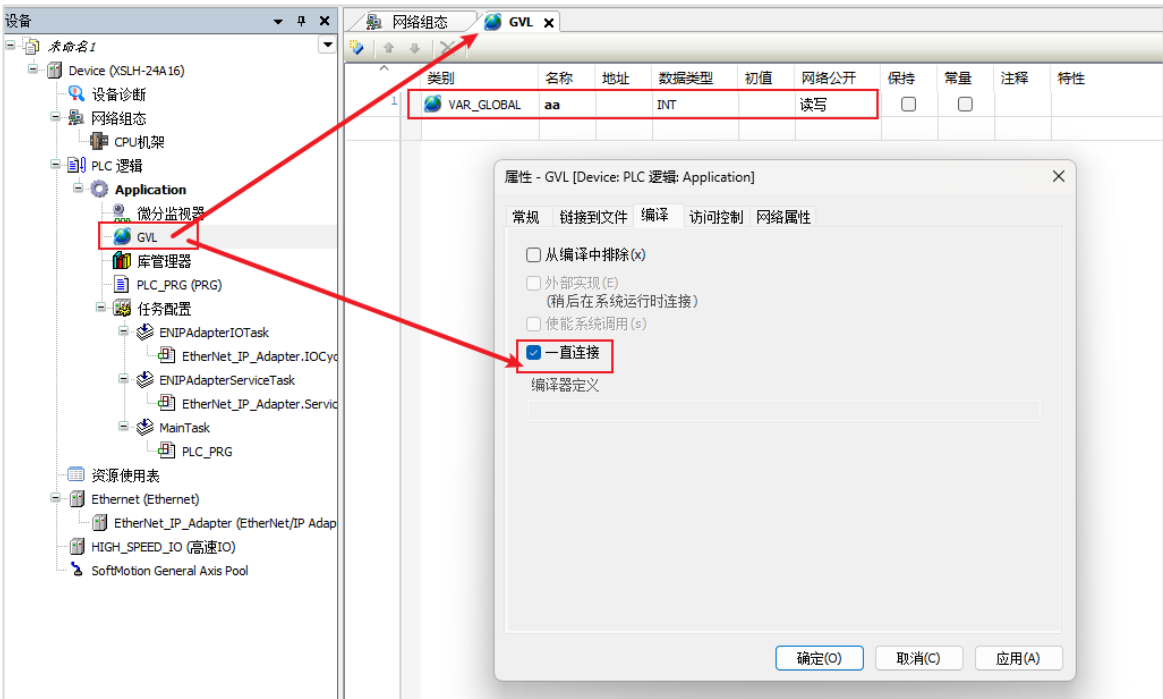
4-3-3. EtherNet/IP 显式通讯样例

使用 XSLH-24A16 和 XDH-60T4 进行 EtherNet/IP 显式通讯。XSLH-24A16 作为 EtherNet/IP 显式通讯的从站，XDH-60T4 作为 EtherNet/IP 显式通讯的主站。

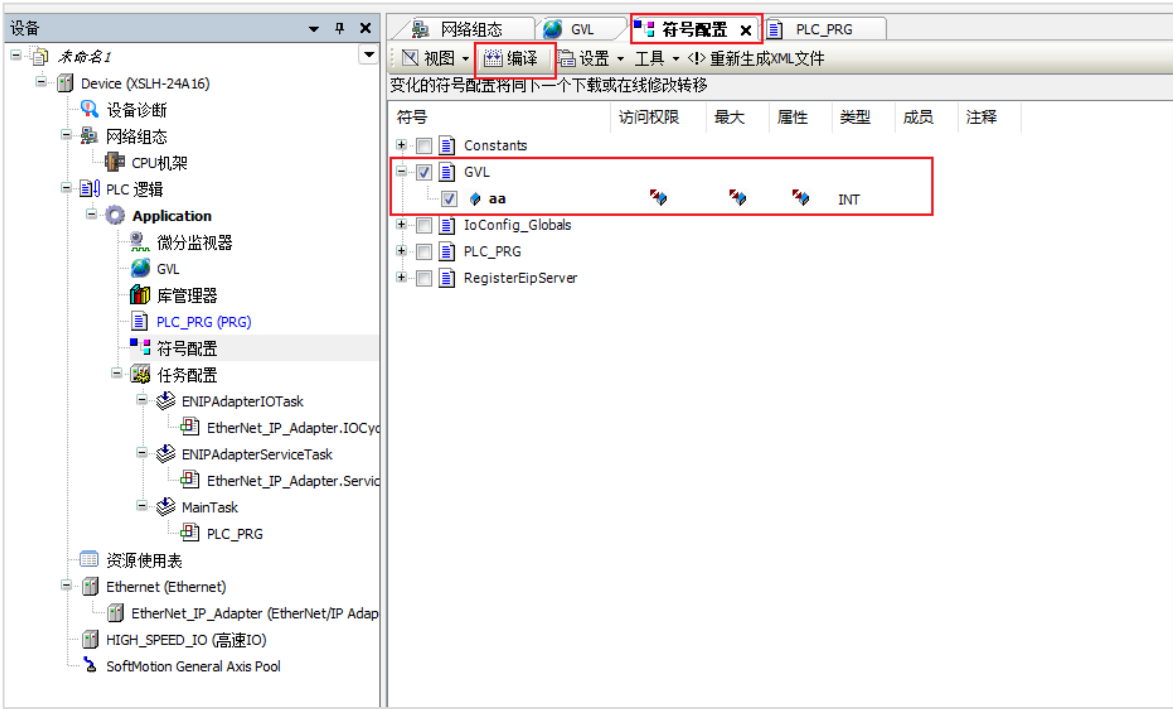
1、 打开 XS Studio 新建工程，以 XSLH-24A16 为例，打开网络组态选择“Ethernet/IP 从站”。



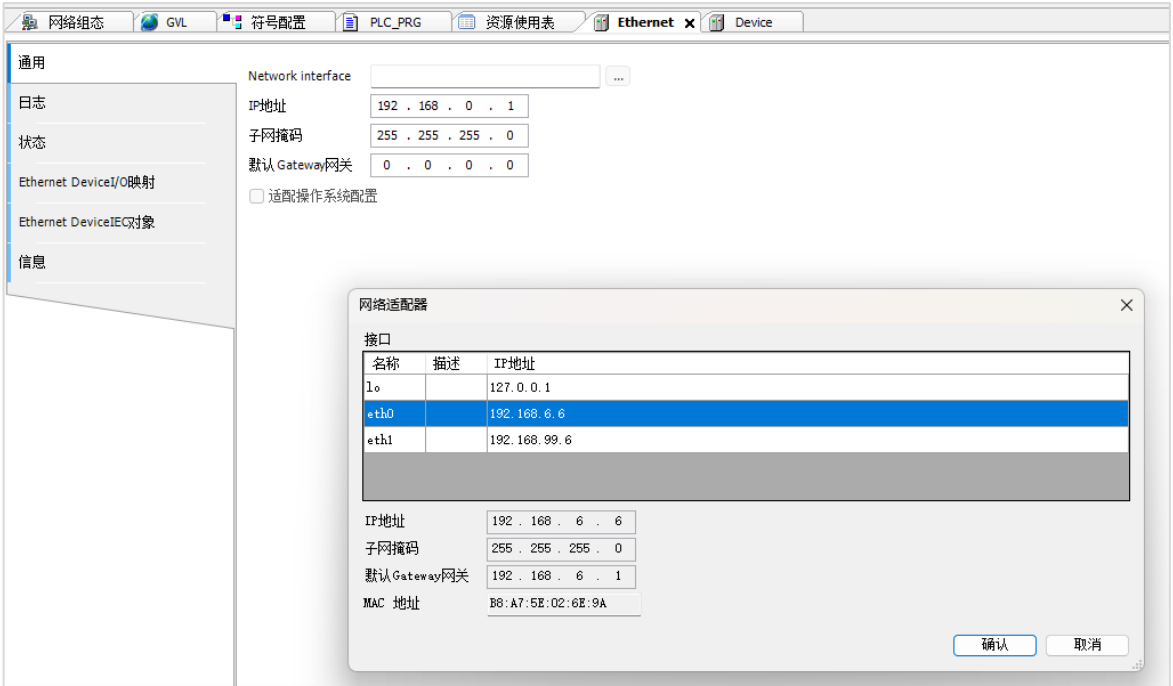
2、 在设备树添加全局变量列表，并创建一个通讯变量，网络属性设置为“读写”，在全局变量属性界面勾选“一直连接”。



3、 在设备树添加“符号配置”列表，并勾选该变量。



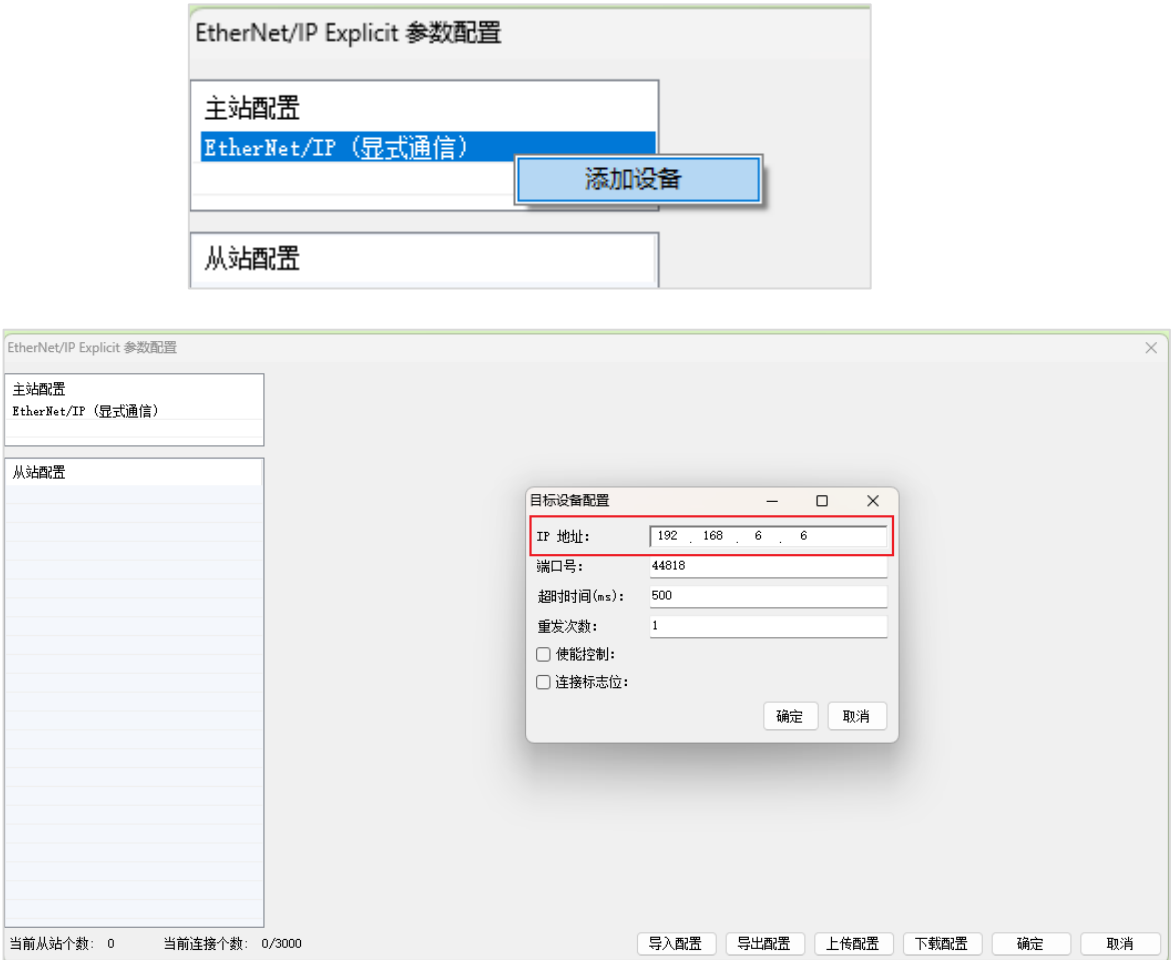
4、在 Ethernet 配置界面中选择使用网口的 IP 地址。



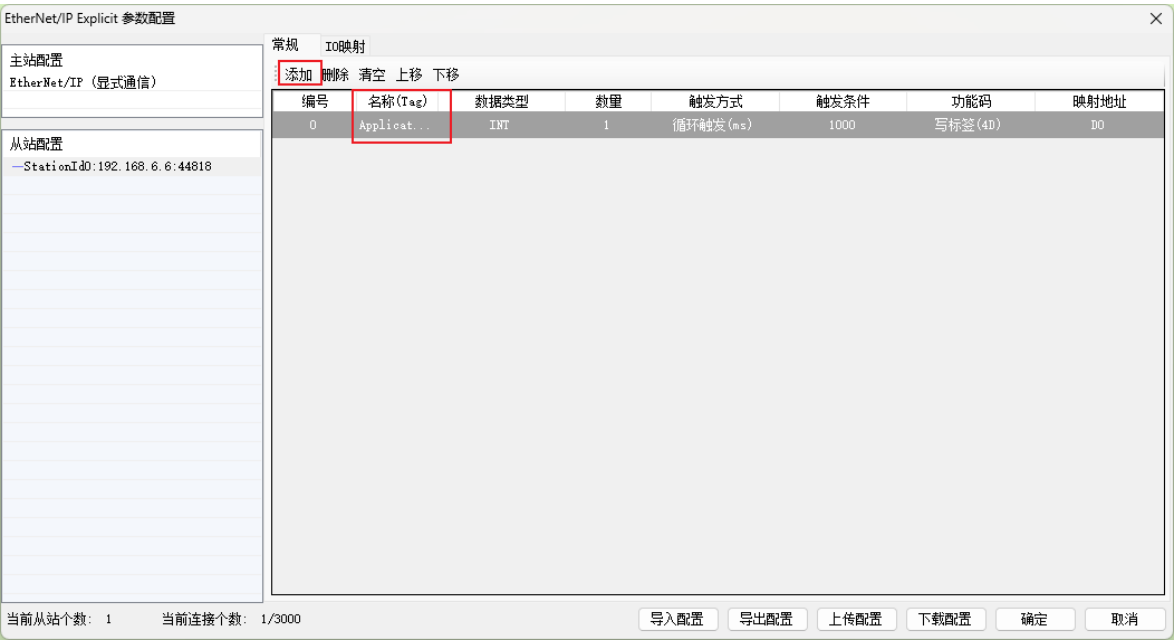
在 XSA-W 工控机桌面安装并运行上位机通讯软件时，内部通讯 IP 需按机型区分配置：

- 1、XSA330-W、XSA500-W 系列工控机：
设备内部通讯统一采用回送地址 127.0.0.1。
- 2、XSA230-W、XSA620-W 支持虚拟化的工控机：
①内部通讯目标 IP 固定为 192.168.2.2；
②本机虚拟网卡默认 IP 为 192.168.2.1，可连接显示器查看；
③若用户需要修改虚拟网卡的 IP，则仅支持修改为 192.168.2 网段的 IP 地址，不可更换为其他网段。

- 5、登录并下载运行。
- 6、打开 XDDPRO 编程软件，连接 XDH-60T4，在工程栏中添加 EthernetIP 显式通讯配置，并添加从站 IP 地址。



- 7、添加通讯标签变量，变量名称为 Application.GVL.变量名(全局变量列表名)，案例中通讯的变量为 aa，故此处通讯标签名称为 Application.GVL.aa。



EtherNet/IP Explicit 参数配置

主站配置	常规				
	IO映射				
EtherNet/IP (显式通信)	标签	通道	类型	映射地址	数值
	Application.GVL aa	写标签(4D)	INT	D0	0
从站配置					
StationId:192.168.6.6:44818					

8、登录并下载程序，可实现 EtherNet/IP 显式通讯。

设备

Device [连接的] (XSLH-24A16)

设备诊断

网络组态

CPU机架

Device.Application.GVL

表达式	类型	值	准备值	地址	注释
aa	INT	10			

窗口(W) 帮助(H)

PLC1-自由监控1

监控窗口

添加 修改 删除 全部删除 上移

名称	监控值	类型	映射地址/字
D0	10	INT	单字

Explicit 参数配置

常规 IO映射

添加 删除 清空 上移 下移

编号	名称(Tag)	数据类型	数量	触发方式	触发条件	功能码	映射地址
0	Applicat...	INT	1	循环触发(ms)	1000	写标签(4D)	D0

5. EtherCAT 配置

5-1. EtherCAT 概要

5-1-1. 概述

EtherCAT 是 Ethernet for Control Automation Technology 的简称。是 Beckhoff Automation GmbH 开发的实时以太网用的主站和从站间的开放网络通信，由 ETG（EtherCAT Technology Group）进行管理。

5-1-2. 系统构成

EtherCAT 的连接形态是，线型连接主站（FA 控制器）和多个从站的网络系统。
从站可连接的节点数取决于主站处理或者通信周期、传送字节数等。

5-1-3. 通讯规格

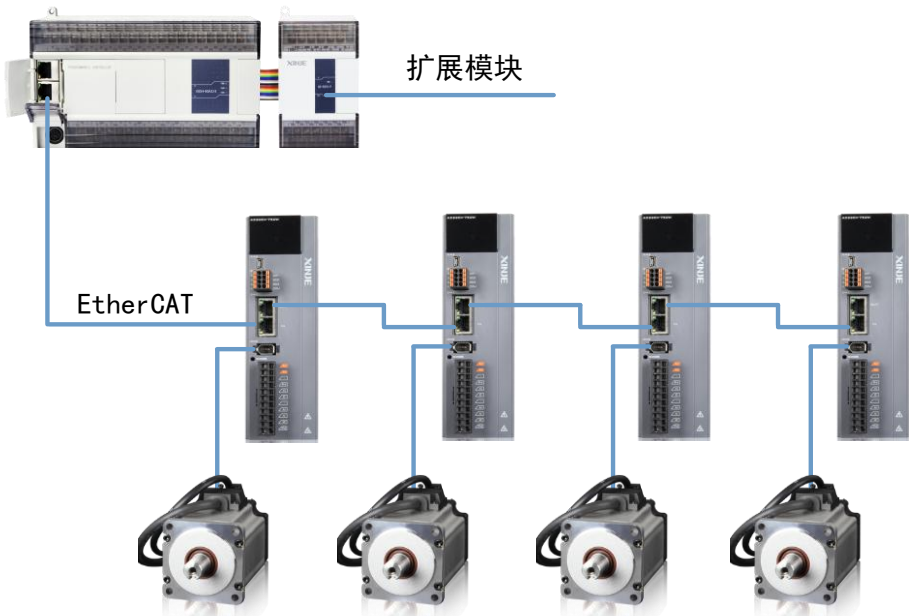
项目	规格
物理层	100BASE-TX（IEEE802.3）
波特率	100[Mbps]（full duplex）
拓扑	Line
连接电缆	JC-CA 双绞线（屏蔽双绞线）
电缆长	节点间最长 100m
通信口	2 Port（RJ45）
EtherCAT Indicators（LED）	[Run] RUN Indicator [L/A IN] Port0 Link/Activity Indicator（Green） [L/A OUT] Port1 Link/Activity Indicator（Green）
Station Alias（ID）	设定范围：0～65535 设定地址：2700h
Explicit Device ID	不支持
邮箱协议	COE（CANopen Over EtherCAT）
SyncManager	4
FMMU	3

项目	规格		
Modes of operation 控制模式	Modes of operation		
	位置	csp	Cyclic synchronous position mode (Cyclic 位置控制模式)
		PP	Profile position mode (Profile位置控制模式)
		hm	Homing mode (原点复位位置控制模式)
	速度	csv	Cyclic synchronous velocity mode (Cyclic 速度控制模式)
		pV	Profile velocity mode (Profile速度控制模式)
	转矩	cst	Cyclic synchronous torque mode (Cyclic 转矩控制模式)
		tq	Torque profile mode (Profile 转矩控制模式)
同期模式	DC (SYNCO 事件同期), SM (SM 事件同步)		
Cyclic time (DC 通信周期)	500, 1000, 2000, 4000[μs]		
通信对象	SDO[服务数据对象], PDO[过程数据对象]		
PreOP 模式下邮箱通讯间隔	1ms		
电子邮箱	SDO 请求和 SDO 信息		

5-1-4. EtherCAT 通讯连接说明

EtherCAT 运动控制系统的接线十分简单，得益于 EtherCAT，Ethernet 的星型拓扑结构可以被简单的线型结构所替代。以信捷 DS5C1 系列伺服为例，由于 EtherCAT 无需集线器和交换机，DS5C1 系列伺服均自带 EtherCAT 通讯网口，因而电缆、桥架的用量大大减少，连线设计与接头校对的工作量也大大减少，便于节省安装费用。

EtherCAT 总线接线建议使用线型接法。XSDH 系列接线方式如下图所示：

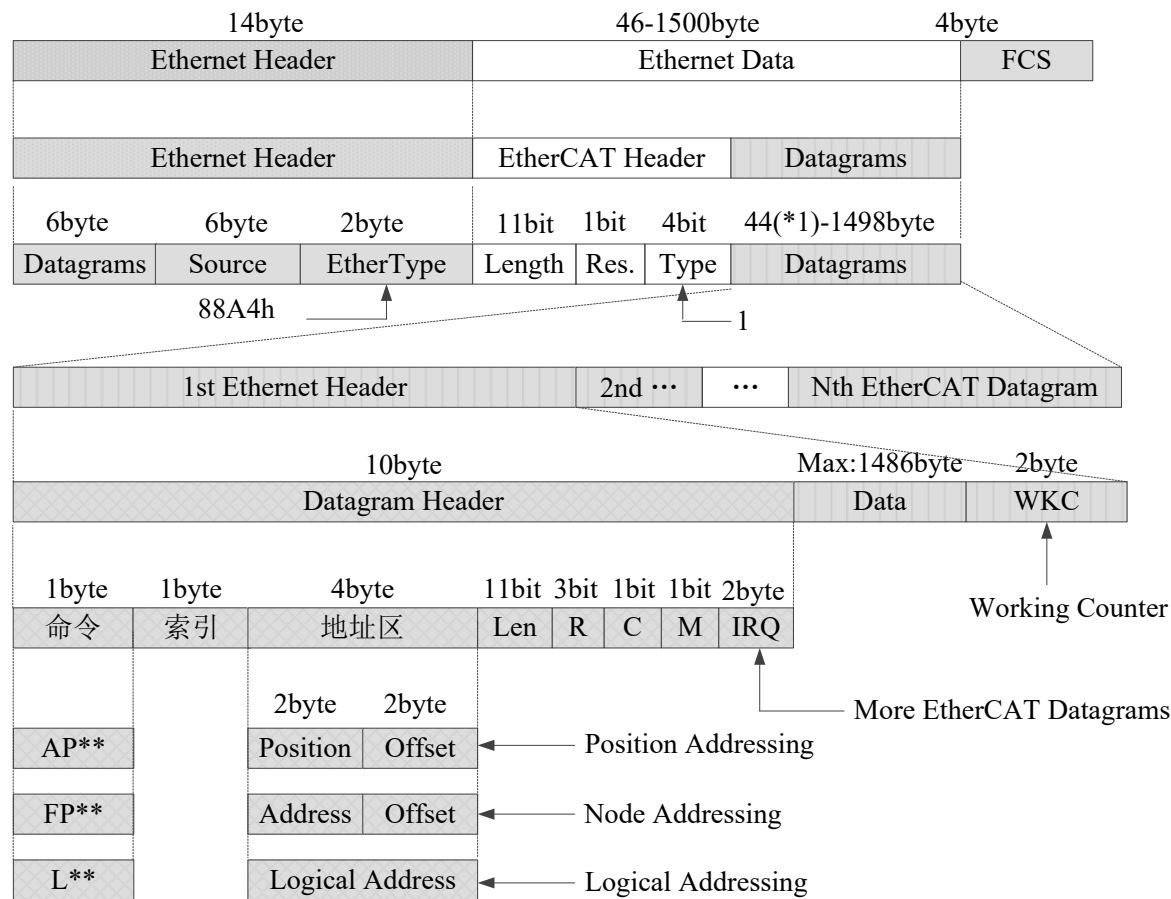


整个总线网络为线型结构，其中 XSDH 系列控制器为主站，信捷 DS5C1 系列总线控制型伺服为从站。XSDH 系列 PLC 带有上下两个网口，上面的网口为 Ethernet/IP，用于连接 XS Studio 上位机；下面的网口为 EtherCAT 接口，用于连接信捷 DS5C1 系列伺服实现 EtherCAT 通讯。信捷 DS5C1 系列伺服驱动器的两个通讯网口则需遵循“下进上出”的原则。

5-2. EtherCAT 通讯规格

5-2-1. EtherCAT 帧结构

EtherCAT 是基于 Ethernet 可实时控制的工业用通信协议，只是对 IEEE 802.3Ethernet 规格进行扩充，并未对基本结构进行任何变更，所以可以转送标准的 Ethernet 帧内的数据。

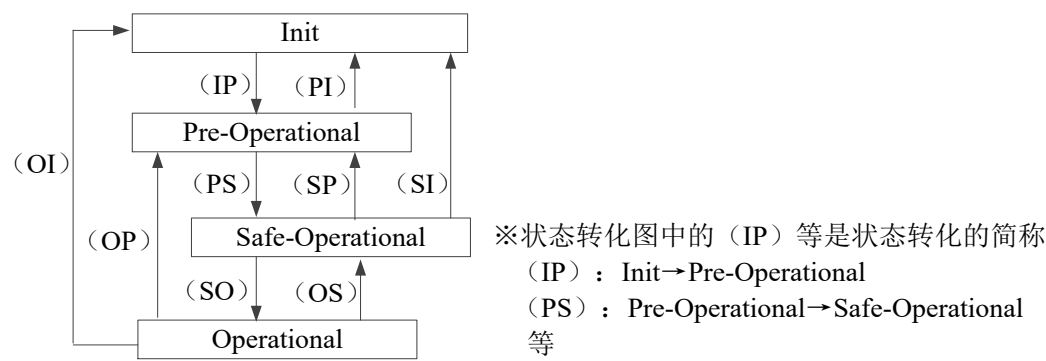


因为 Ethernet Header 的 EtherType 为「88A4h」，所以将之后的 Ethernet Data 作为 EtherCAT 帧来处理。“EtherCAT 帧”是以某种协议来进行定义和解析，只要主站和从站都遵守这个协议，就可以实现数据通信了。一般采用的协议有 CANopen Over EtherCAT (CoE)，还有 Sercos Over EtherCAT (SoE) 等。

5-2-2. 状态机 ESM

EtherCAT 状态机 (ESM) 负责协调主站和从站应用程序在初始化和运行时的状态关系。状态改变请求由主站执行，主站向应用层服务提出控制请求，后者在从站中产生应用层控制事件，从站在状态改变请求成功或失败后通过本地的应用层状态写服务来响应应用层控制服务。如状态改变失败，从站保持状态并置出错误标志。

下图为 ESM 的状态转化图：



Init: 初始化状态；
Pre-Operational: 预运行状态；
Safe-Operational: 安全运行状态；
Operational: 运行状态；

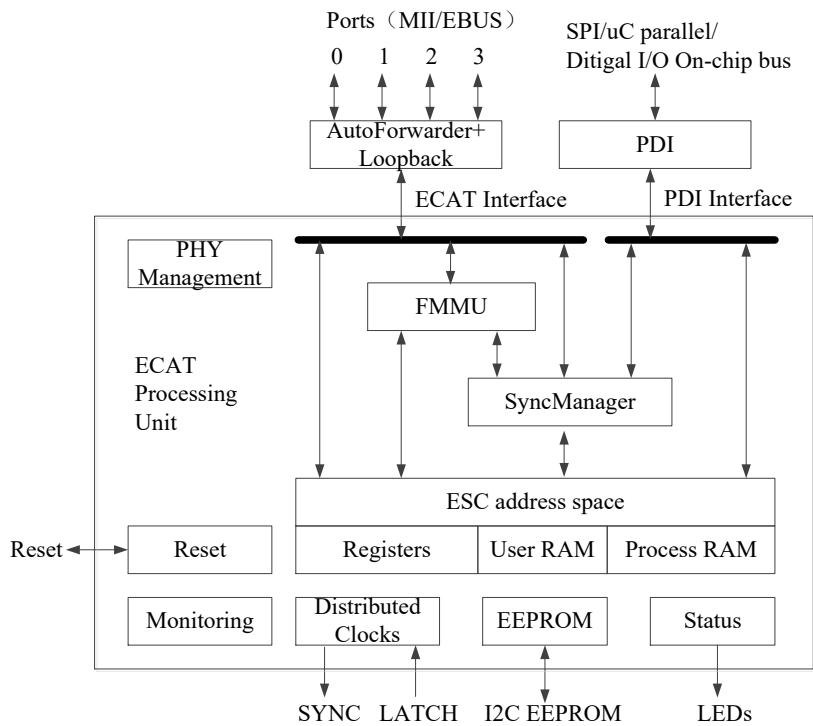
从站状态	各状态下的动作	通讯动作		
		SDO（邮箱）收发信	PDO 发信	PDO 收信
Init	通信初始化，SDO、PDO 无法收发信的状态	-	-	-
Pre-Operational（简称 PreOP）	仅 SDO 收发信的状态	Yes	-	-
Safe-Operational（简称 SafeOP）	仅 SDO 收发信，PDO 发信的状态	Yes	Yes	-
Operational（简称 OP）	SDO 收发信，PDO 收发信全部可行的状态	Yes	Yes	Yes

从主站到 ESC 寄存器的访问与上表无关，随时都可以。
PDO（Process Data Object）过程数据对象用来传输周期性通讯数据。
SDO（Service Data Object）服务数据对象用来传输非周期性通讯数据。
ESM 状态切换时进行指令或界面操作可能造成通信异常报错。

5-2-3. 从站控制器 ESC

5-2-3-1. 原理概述

ESC 指的是 EtherCAT 从站控制器（EtherCAT Salve Controller）。通信过程完全由 ESC 处理，它具有四个数据收发端口，每个端口具有一个 TX 和 RX。每个端口都可以收发以太网数据帧，ESC 中的数据流向是固定的：端口 0——>端口 3——>端口 1——>端口 2——>端口 0 的顺序依次传输。如果 ESC 检测到某个端口没有外接 PHY，则自动闭合这个端口，通过内部回环自动转发到下一个端口。



5-2-3-2. 地址空间

DS5C1 系列持有 8Kbyte 的物理地址空间。
最初的 4Kbyte（0000h~0FFFh）是作为寄存器空间使用，另外 4Kbyte（1000h~1FFFh）是过程数据 PDO 作为 RAM 领域使用。寄存器的详细内容请参考 IP（ET1810/ET1811/ET1812）的数据表。

ESC 寄存器字节地址	长度 (Byte)	说明	初始值
ESC Information（从站控制器信息）			
0000h	1	Type	04h
0001h	1	Revision	02h
0002h~0003h	2	Build	0040h
0004h	1	FMMUs supported	03h
0005h	1	SyncManagers supported	04h
0006h	1	RAM Size	08h
0007h	1	Port Descriptor	0Fh
0008h~0009h	2	ESC Features supported	0184h
Station Address			
0010h~0011h	2	Configured Station Address	-
0012h~0013h	2	Configured Station Alias	-
...			
Data Link Layer			
...			
0100h~0103h	4	ESC DL Control	-
...			
0110h~0111h	2	ESC DL Status	-
Application Layer			

ESC 寄存器字节地址	长度 (Byte)	说明	初始值
0120h~0121h	2	AL Control	-
0130h~0131h	2	AL Status	-
0134h~0135h	2	AL Status Code	-
...			
PDI			
0140h	1	PDI Control	08h
0141h	1	ESC Configuration	0Ch
0150h	1	PDI Configuration	-
0151h	1	SYNC/LATCH PDI Configuration	66h
0152h~153h	2	Extend PDI Configuration	-
...			
Watchdogs			
0400h~0401h	2	Watchdog Divider	-
0410h~0411h	2	Watchdog Time PDI	-
0420h~0421h	2	Watchdog Time Process Data	-
0440h~0441h	2	Watchdog Status Process Data	-
0442h	1	Watchdog Counter Process Data	-
0443h	1	Watchdog Counter PDI	-
...			
FMMU			
0600h~062Fh	3x16	FMMUs[2:0]	-
+0h~3h	4	Logical Start Address	-
+4h~5h	2	Length	-
+6h	1	Logical Start bit	-
+7h	1	Logical Stop bit	-
+8h~9h	2	Physical Start Address	-
+Ah	1	Physical Start bit	-
+Bh	1	Type	-
+Ch	1	Activate	-
+Dh~Fh	3	Reserved	-
...			
Distributed Clocks (DC) -SYNC Out Unit			
0981h	1	Activation	-
...			
0984h	1	Activation Status	-
098Eh	1	SYNCO Status	-
...			
0990h~0993h	4	Start Time Cyclic Operation/Next SYNCO Pulse	-
...			
09A0h~09A3h	4	SYNCO Cycle Time	-
...			

5-2-4. SII 区域

ESC 配置区域（EEPROM 字地址 0000h~0007h）内，Configured Station Alias 在驱动器电源启动后，根据 ESC 自动读取，写入 ESC 寄存器。将 SII EEPROM 变更后的值反映到 ESC 寄存器时，需要再次启动电源。除此之外 IP 核（ET1810/ET1811/ET1812）的初始值被设定。详细内容请参考 IP 核（ET1810/ET1811/ET1812）的数据表。

5-2-5. SDO

DS5C1系列支持SDO（Service Data Object）。SDO的数据交换使用Mailbox通信，所以SDO的数据刷新时间变得不稳定。

主站侧在对象字典内的记录中读写数据，可进行对象设定以及从站的各种状态的监测。到SDO的读写动作的响应需要花费时间。用PDO刷新的对象请不要用SDO来刷新，用PDO的值覆盖。

5-2-5-1. Mailbox 帧结构

Mailbox/SDO 的帧结构如下所示。详细请参照 ETG 规格书（ETG1000-5 及 ETG1000-6）。



帧部	数据区域	数据类型	功能
	Service	Unsigned4	信息型
Cmd specific	Size Indicator	Unsigned1	Data Set Size 使用许可
	Transfer Type	Unsigned1	Normal 转送/Expedited 转送择
	Data Set Size	Unsigned2	指定数据大小
	Complete Access	Unsigned1	对象的访问方法的选择（未对应）
	Command Specfier	Unsigned3	上传/下载 要求/响应等的选择
	Index	WORD	对象的 Index
	Subindex	BYTE	对象的 Subindex
			对象的数据或者 Abort message 等

5-2-5-2. Mailbox 超时

本伺服驱动器在Mailbox通信中进行下述超时设定。

Mailbox请求的超时时间：100ms

主站向从站（驱动器）发出请求，请求帧的发信数据的WKC如果被更新，从站则被认为正常接收请求。直到WKC被更新为止，反复重试，然而直到此设定时间WKC仍未被更新则主站侧超时。

Mailbox响应的超时时间：10s

主站接收来自从站（驱动器）请求的响应，如果此WKC被更新则认为是正常接收响应。直到此设定时间为止，如果无法接收WKC被更新的响应，则主站侧超时。

从站（驱动器）的响应完成所需的最大时间。

5-2-5-3. 异常报警时信息

1、Error code

Error code返回和603Fh（Error code）相同的值。

0000h～FEFFh根据IEC61800-7-201进行定义。

FF00h～FFFFh由制造商定义的，下述内容所示：

Index	Sub-Index	Name/Description	Range	Date Type	Access	PDO	Op-mode
603Fh	00h	Error code	0-65535	U16	ro	TxPDO	All
		现在伺服驱动器发生的报警（只有主编号）。 报警未发生时，显示 0000h。 报警发生时，显示报警。 FF**h 报警（主）编号（00h～FFh） （例）FF03h ... 03h=3d E-030（过压保护）发生 FF55h ... 55h=85d E-850（TxPDO 配置异常保护）、E-851（RxPDO 配置异常保护），其中任意一个发生 作为例外，E-817（SyncManager2/3 设定异常）的情况下，显示 A000h。					

2、Error register

Error register返回和1001h（Error register）相同的值。

Index	Sub-Index	Name/Description	Range	Date Type	Access	PDO	Op-mode
1001h	00h	Error register	0-65535	U16	ro	TxPDO	All
		显示伺服驱动器正发生的报警种类（状态）。 报警未发生时，显示 0000h。 不显示警告。					
		Bit	内容				
		0	不支持				
		1					
		2					
		3					
		4	AL status code 定义的报警 生*1				
		5	不支持				
		6	保留				
		7	AL status code 未定义的报警发生*2				
		*1: 所谓“AL status code 定义的报警”，指 EtherCAT 通信关联异常 E-800～7、E-810～7、E-850～7。					
*2: 所谓“AL status code 未定义的报警”，指 EtherCAT 通信关联异常 E-880～7 和 EtherCAT 通信关联以外的异常。							

5-2-6. PDO

DS5C1 系列支持 PDO（Process Data Object）。
基于 EtherCAT 的实时数据转送通过 PDO（Process Data Object）的数据交换进行。
PDO 有从主站到从站转送的 RxPDO 和从从站到主站转送的 TxPDO。

	发信侧	收信侧
RxPDO	主站	从站
TxPDO	从站	主站

5-2-6-1. PDO 映像对象

PDO 映射是指，从对象字典到 PDO 的应用对象的映射。
DS5C1 系列 PDO 映射用的表，可以使用 RxPDO 用 1600h～1603h、TxPDO 用 1A00h～1A03h 的映射对象。
一个映射对象可以映射的应用对象的最大数如下所示：
RxPDO: 24 [byte] ， TxPDO: 24 [byte]
以下表示的是 PDO 映射的设定示例。
< 设定示例 >
分配应用对象 6040h, 6060h, 607Ah, 60B8h 到映射对象 1600h(Receive PDO mapping 1: RxPDO_1) 的情况。

Index	Sub	Object contents	
1600h	00h	04h	
	01h	6040 00 10 h	
	02h	6060 00 08 h	
	03h	607A 00 20 h	
	04h	60B8 00 10 h	
	05h	0000 00 00 h	
	...		
	18h	0000 00 00 h	
6040h	00h	Controlword	U16
6060h	00h	Mode of operation	I8
607Ah	00h	Target Position	I32
60B8h	00h	Touch probe function	U16

5-2-6-2. PDO 分配对象

为了 PDO 数据交换, 必须分配 PDO 映射用的表到 SyncManager。PDO 映射用的表和 SyncManager 的关系记述到 PDO 分配对象。DS5C1 系列, 作为 PDO 分配对象, 可以使用 RxPDO(SyncManager2) 用 1C12h、TxPDO (SyncManager3) 用 1C13h。

一个映射对象可以映射的应用对象的最大数如下所示:

RxPDO: 4 [Table] (1600h~1603h)。

RxPDO: 4 [Table] (1A00h~1A03h)。

通常、因为映射对象 1 个就足够了, 所以默认的不需要变更。

PDO 分配对象的设定示例:

分配映射对象 1600h 到分配对象 1C12h (Sync manager channel 2) 的情况。

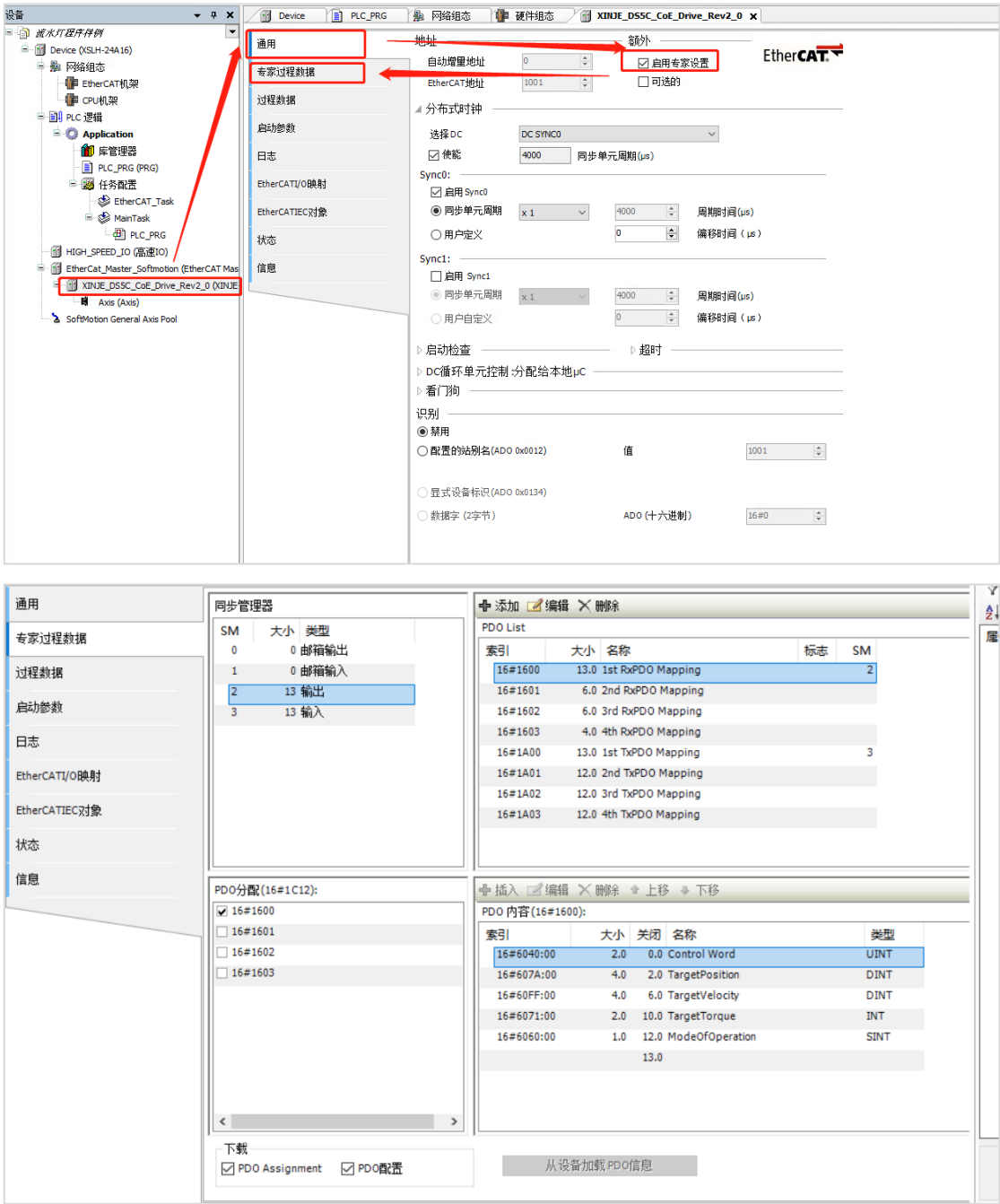
Index	Sub	Object contents
1C12h	00h	01h
	01h	1600h
	02h	0000h
	03h	0000h
	04h	0000h

分配映射对象 1600h 到分配对象 1C13h (Sync manager channel 3) 的情况。

Index	Sub	Object contents
1C13h	00h	01h
	01h	1A00h
	02h	0000h
	03h	0000h
	04h	0000h

5-2-6-3. PDO 配置

双击 EtherCAT 从站设备-“通用”-勾选“启用专家设置”-“专家过程数据”



5-2-7. 通信同步模式

DS5C1 系列可以选择以下的同步模式。

同步模式	内容	同步方法	特征
DC	SYNC0 事件同步	以第 1 轴的时间为基准同步 其他从站的时刻信息	高精度 需要在主站侧进行补偿处理
SM2	SM2 事件同步	根据 RxPDO 的收信时间进行同步	无传送延迟补偿，精度差 需要在控制器侧保持传送时间（专用

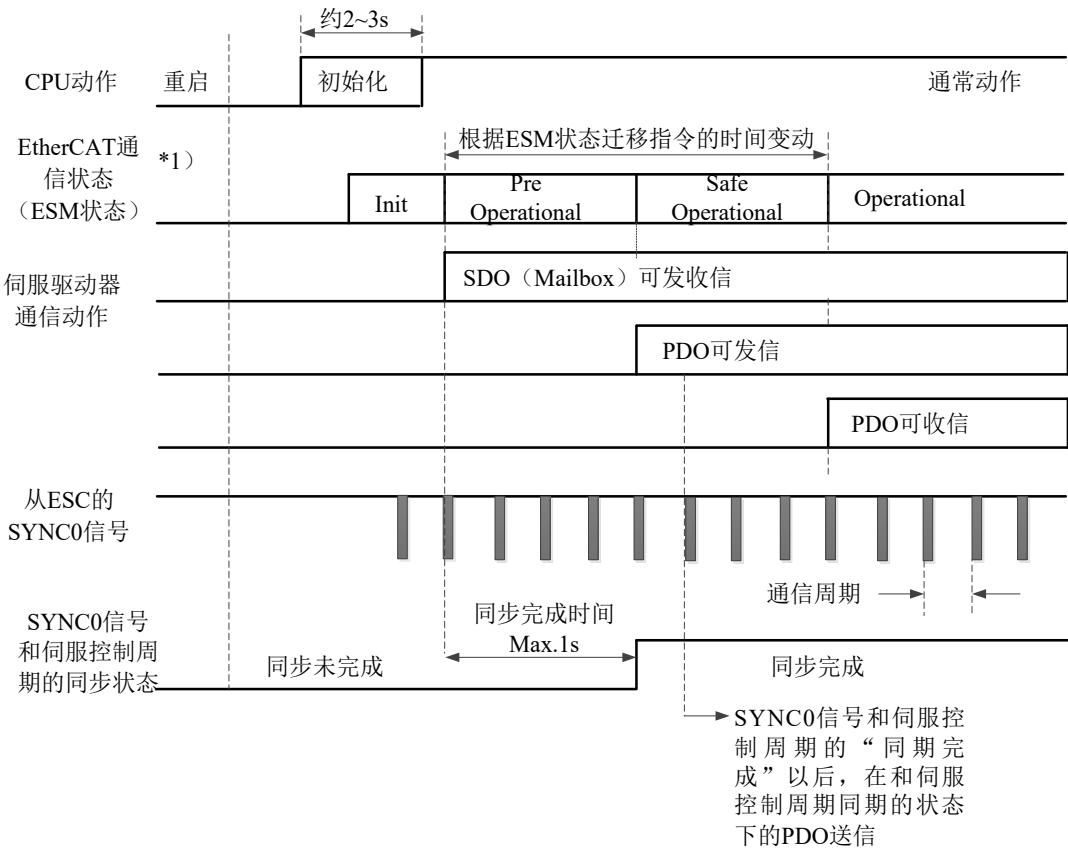
同步模式	内容	同步方法	特征
			硬件等)
FreeRun	非同步	非同步	处理简单 实时性差

5-2-7-1. DC (SYNC0 事件同步)

DS5C1 系列有 64bit 的 DC (Distributed Clock)。

EtherCAT 通信的同步是基于此 DC 进行的。依据 DC 从站通过共有相同基准的时钟(System Time)实现同步。从站的本地周期开始于 SYNC0 事件。因为从站的处理(伺服处理)是开始于 SYNC0 事件周期，所以总是与 SYNC0 事件同步。

主站在通信初始化时需要进行传输延时补偿(偏移量补偿)，还有定期的偏差补偿。下图表示从控制电源投入到 SYNC0 事件和从站的处理(伺服处理)的同步完成的过程。



5-2-7-2. SM2 (SM2 事件同步)

从站的本地周期开始于 SM2 事件。

因为从站的处理开始于 SM2 事件周期，所以总是与 SM2 事件同步。

因为 SM2 事件发生在 PDO 的收信完成时，所以一定要确保上位(主站)侧定时送信。如果送信时间的波动(偏差)太大，同步无法完成，或者发生报警。

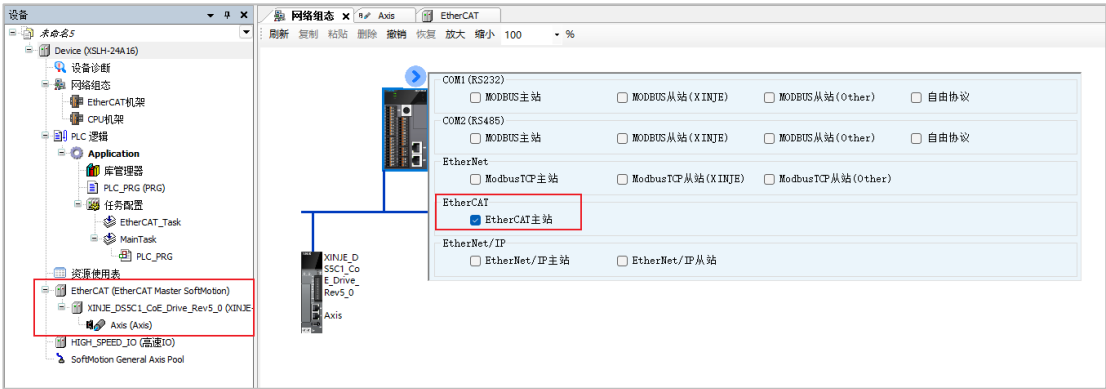
如果发生上述问题，请使用 DC (SYNC0 事件同步)。

5-3. EtherCAT 参数配置

5-3-1. EtherCAT 主站

5-3-1-1. 添加主站

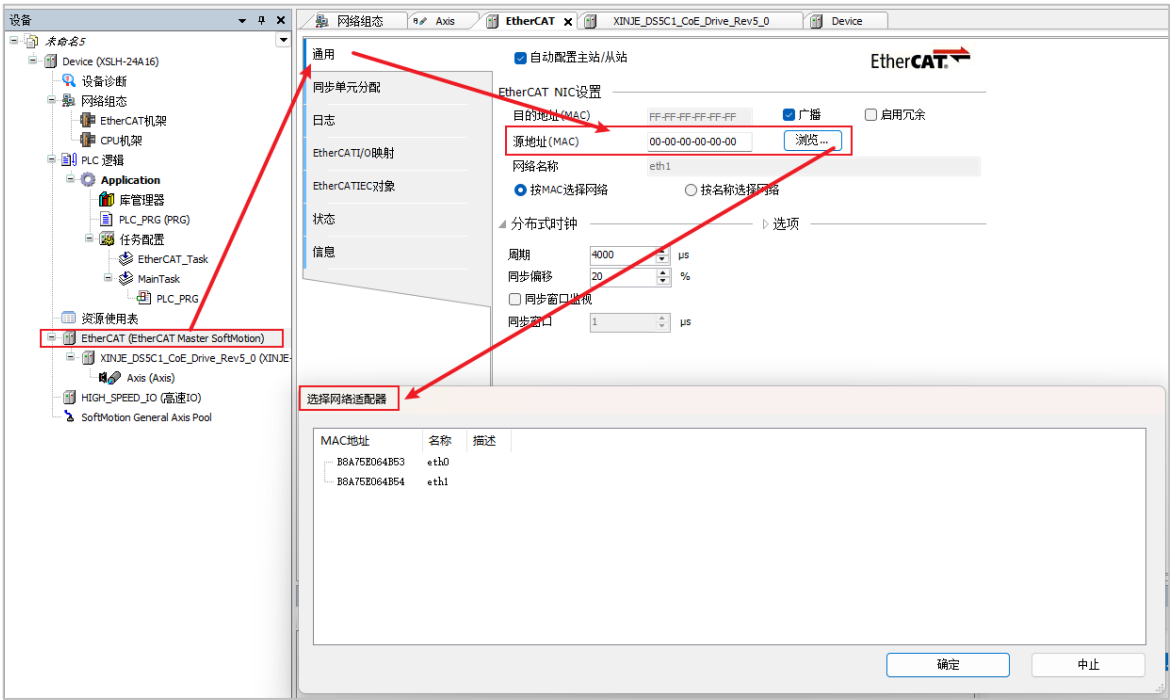
单击网络组态界面内的使能窗口，通过勾选“EtherCAT 主站”添加主站设备，具体如图所示：



- 只支持一个 ECAT 网口的设备，主站名称默认名称为“EtherCAT”；
- 支持两个及以上 ECAT 网口的设备，主站名称默认名称为“EtherCAT_0/EtherCAT_1/EtherCAT_2-----”，且 0/1/2 和网口顺序挂钩，与添加顺序无关。

5-3-1-2. 通用

1、EtherCAT NIC 设置



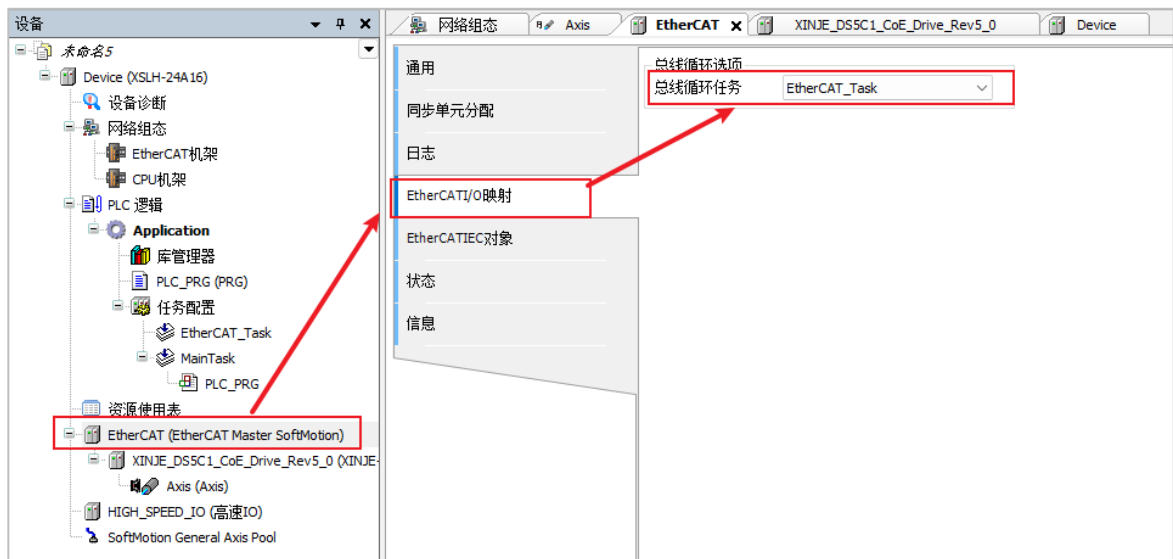
目的地址 Destination address (MAC)	为接受EtherCAT报文的目标地址，如果“广播”选项被激活，则不需要输入目标地址，系统会自动进行通过广播搜索目标地址。
激活冗余 (Redundancy)	该选项使能后，则正式开启EtherCAT冗余模式，其支持环形拓扑结构。
源地址Source address (MAC)	PLC网络接口的MAC地址，可以选择“根据MAC选择网络”或者选择“根据名称选择网络”。用户可以选择“浏览（Browse）”鼠标选择想要设置的源地址。选择“按名称选择网络”，默认网络名称根据上述机型网口配置显示。 注：XS StudioV2.3.1及以上时，软件支持根据机型自动设置对应的ECAT网络名称，无需用户手动配置：

2、分布式时钟

周期时间 (Cycle time)	如果分布式时钟功能被激活，主站发送将会根据该循环时间向从站发送相应数据报文。因此数据交换可以实现精确同步，在分布式过程中要求同步动作时（例如几个伺服轴执行同时联动任务），此功能尤为重要的。可以在网络范围内提供信号抖动小于1微秒的主时钟。
同步偏移 (Sync offset)	通常当PLC任务开始20%后，同步报文开始影响从站，这就意味着PLC的任务周期可以有80%的延迟，在此延迟内不会有数据会丢失。
同步窗口监控 (Sync window monitoring)	如果该选项打开，可以监控从站的同步状态。
同步窗口 (Sync window)	用于监控同步窗口的时间。如果所有的从站在同步窗口时间内，则变量xSyncInWindow (IoDrvEtherCAT)将会被置为TRUE，否则为FALSE。

5-3-1-3. EtherCAT I/O 映射

在建立EtherCAT主站时，会自动建立EtherCAT_Task任务，在EtherCAT I/O映射中设置总线循环任务，默认为EtherCAT_Task。

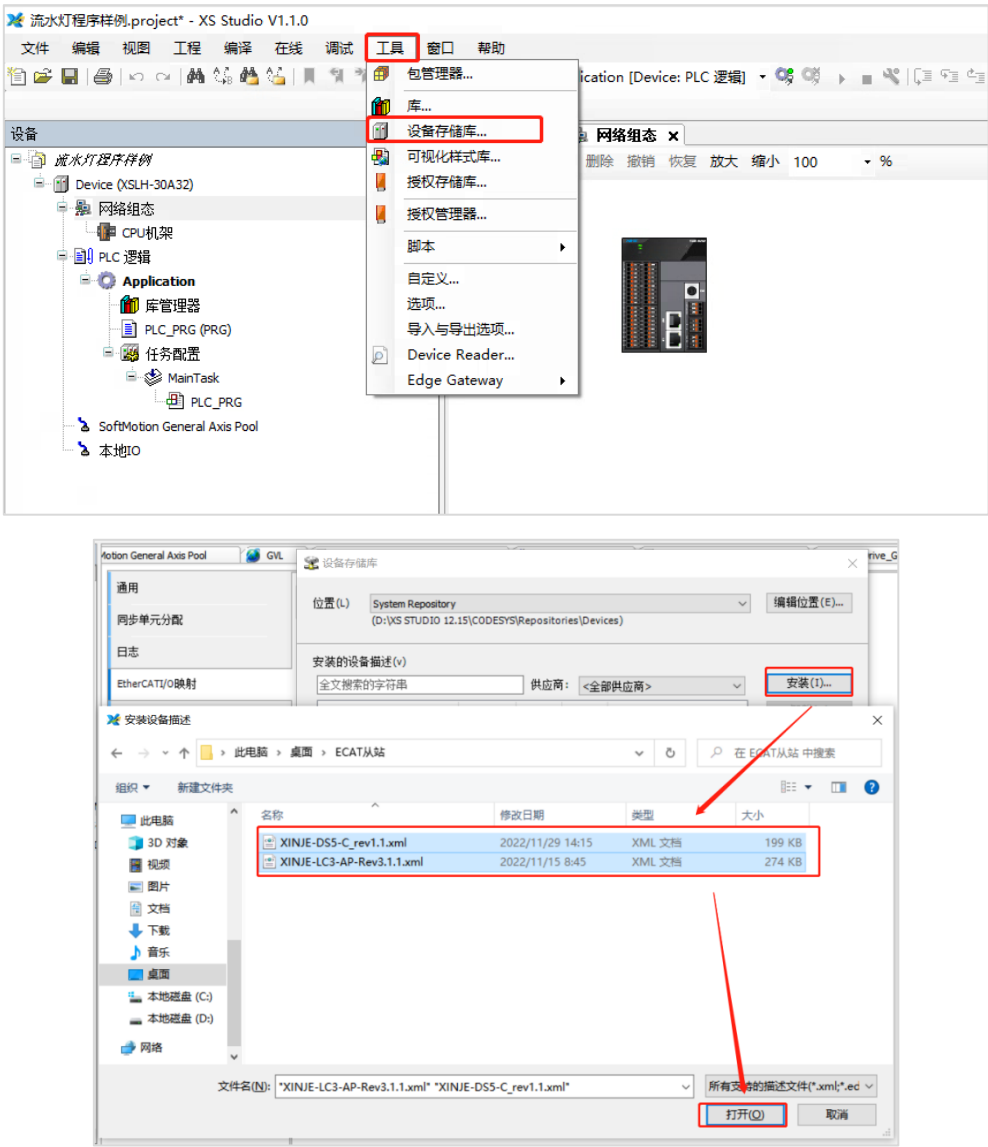


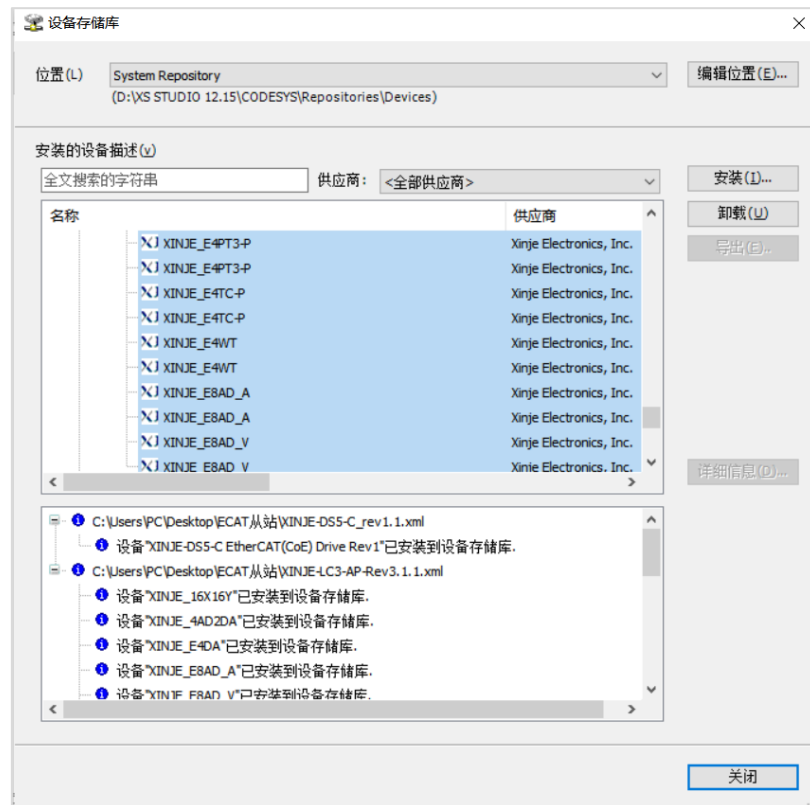
5-3-2. EtherCAT 从站

5-3-2-1. 添加从站

1、添加xml文件

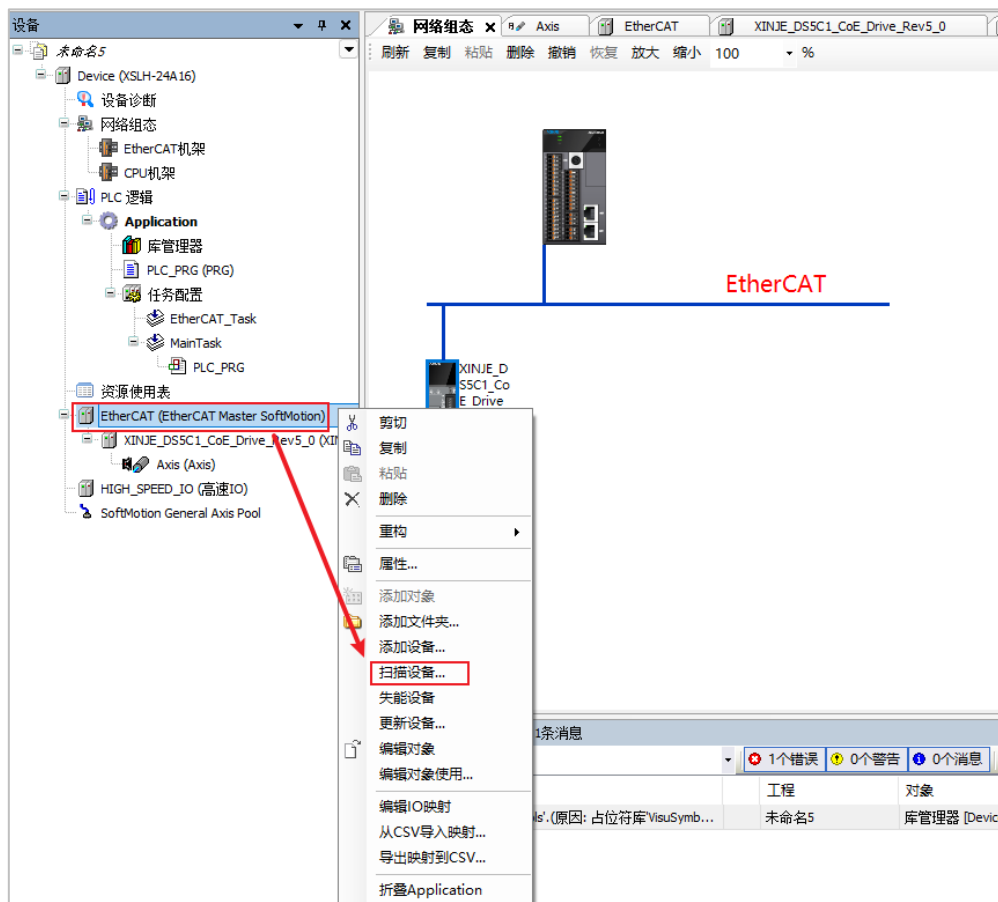
打开工具设备库，添加从站设备的XML文件。添加从站设备的XML文件。依次点击“工具”--“设备存储库...”，在弹出的对话框中点击“安装”，选择XML文件所在的路径找到XML文件，选中后点击打开。



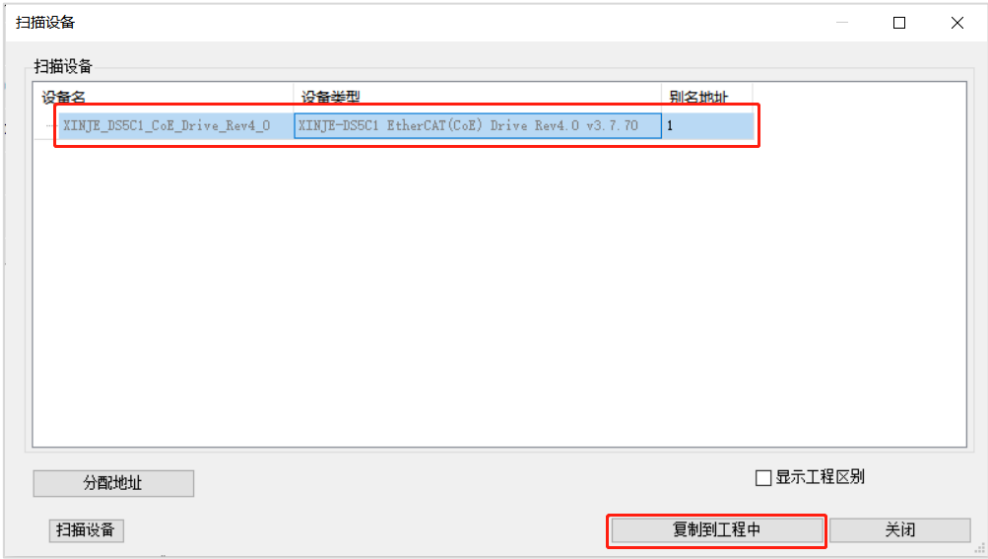


2、扫描从站

在Device工程栏中，右击EtherCAT_Master_SoftMotion，点击“扫描设备”扫描EtherCAT从站设备，或者右击EtherCAT_Master_SoftMotion，点击“添加设备”，手动添加设备。



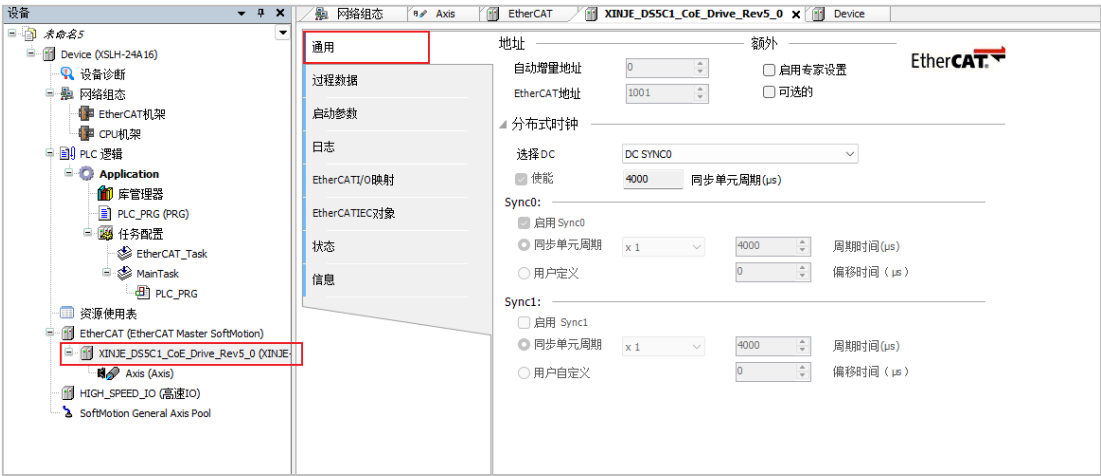
本例中连接了1台DS5C1系列的伺服，扫描结果如下图所示，单击Copy All to Project将扫描到的所有从站添加到项目中去。





使用“扫描设备”功能之前，必须确保从站的 EtherCAT 设备描述文件已经安装在调试 PC 的 XS Studio 内，否则无法使用此功能。

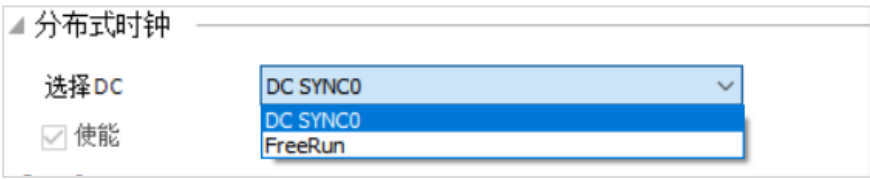
5-3-2-2. 通用



1、地址

自动配置地址	由从站在网络中的位置确定。这个地址仅仅在启动时使用，主站需要向从站分配EtherCAT地址。当用于此目的的首个报文通过从站时，每一个经过的从站将自身的自动增量地址加1。
EtherCAT地址	从站的最终地址，由主站在启动时分配。

2、分布式时钟



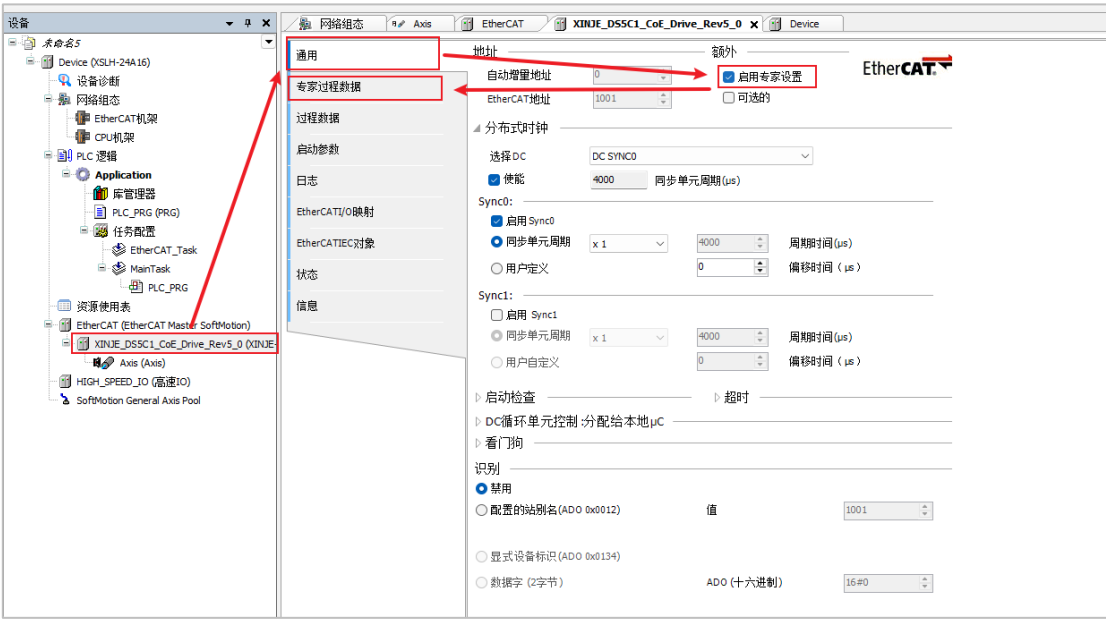
选择DC	下拉菜单提供了由设备描述文件提供的所有关于分布时钟的设置，可以选择同步或者 freerun 非同步模式。
------	--

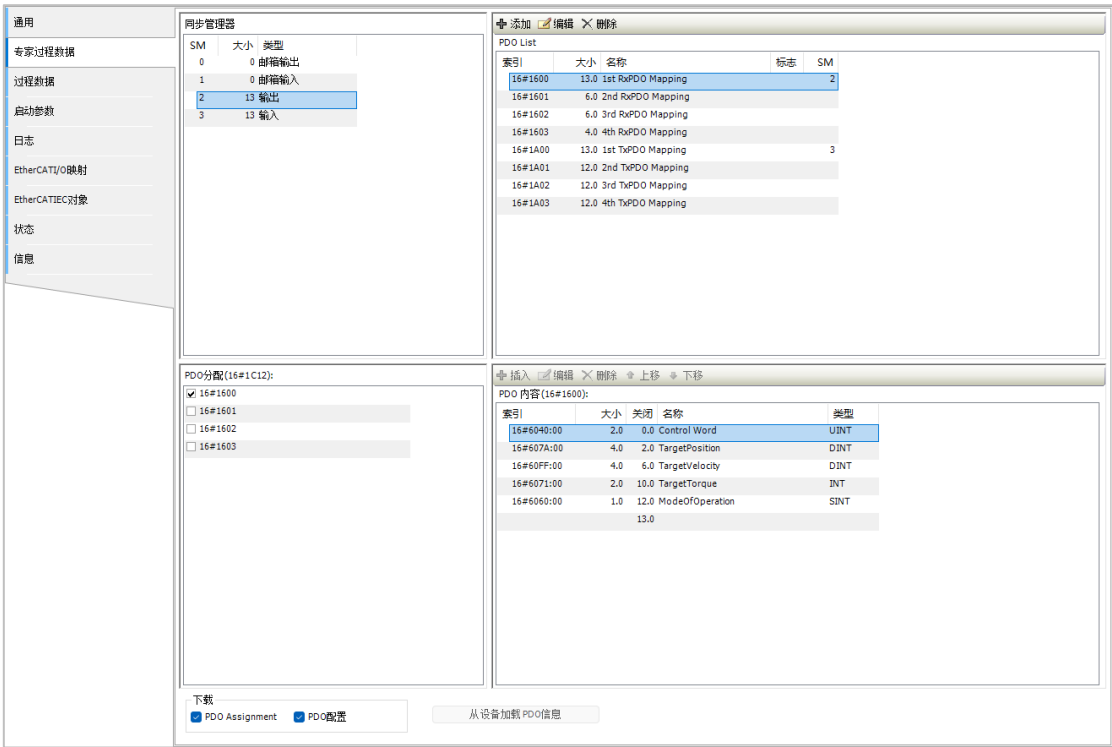
3、同步0/1

sun同步0/1使能	如果该选项被选中，使用“sync0/1”同步单元。一个同步单元描述了一套同步交换的过程数据。
同步单元循环	主站周期的时间乘以所选择的系数，将被用作从站的同步周期时间。循环时间（us）栏显示当前设置的周期时间。

5-3-2-3. 专家过程数据

在通用界面，勾选专家设置，则会出现专家过程数据的配置界面。



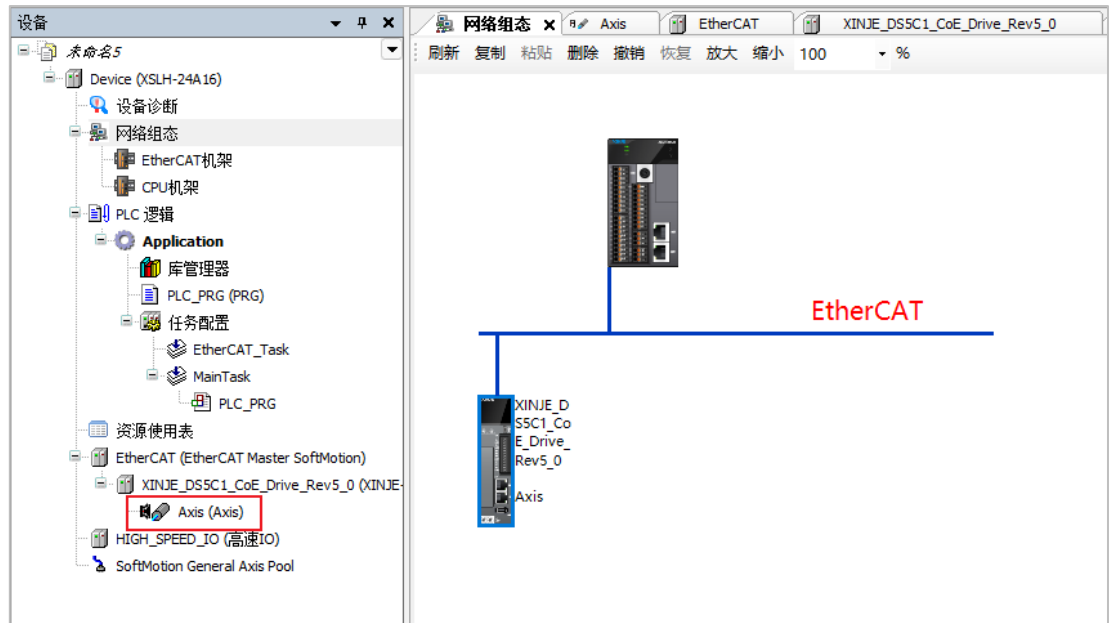


5-3-3. 轴配置

5-3-3-1. 信捷 402 轴

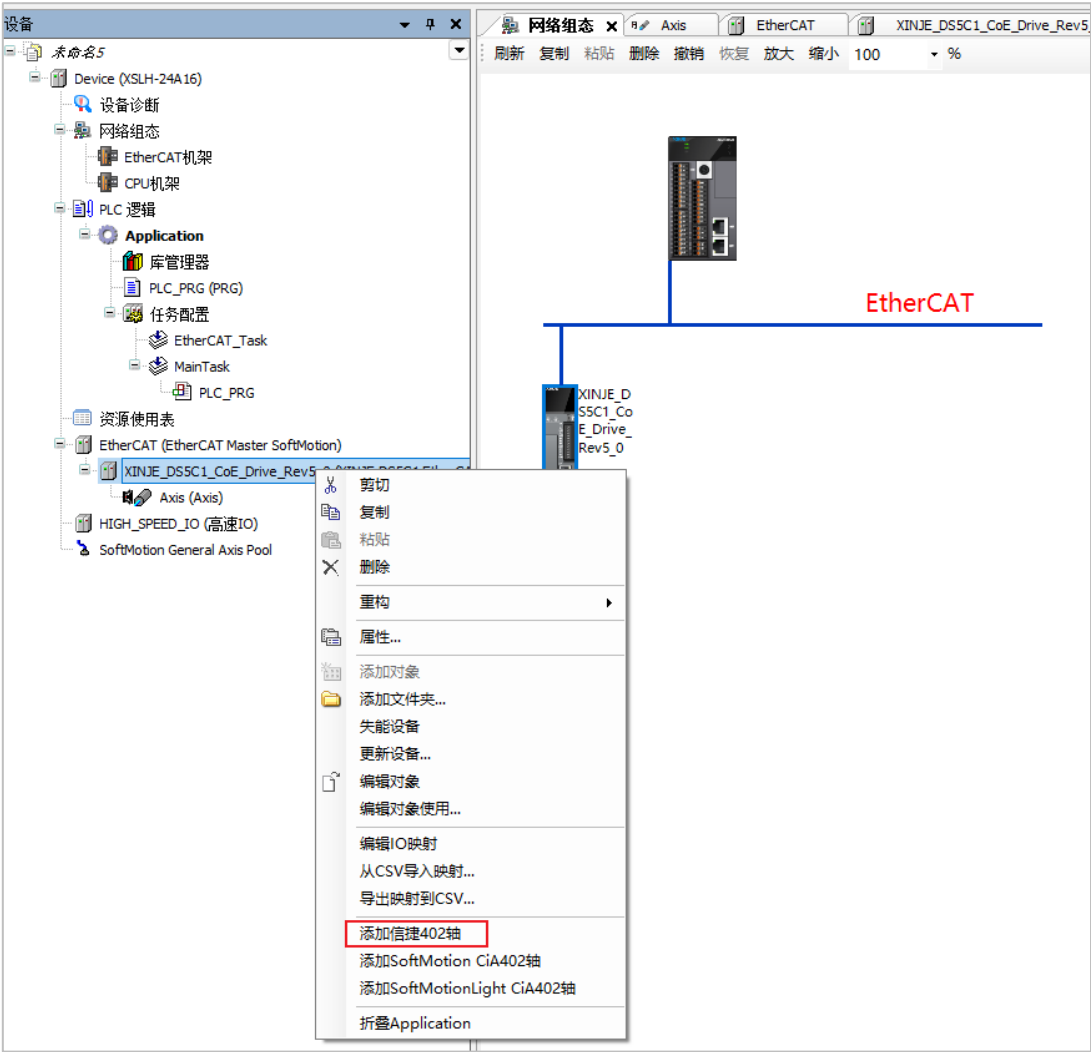
1、添加“信捷402轴”

方法一：在网络组态界面内使能 EtherCAT 主站，添加信捷的伺服从站设备时，会自动添加信捷 402 轴。如下图所示：

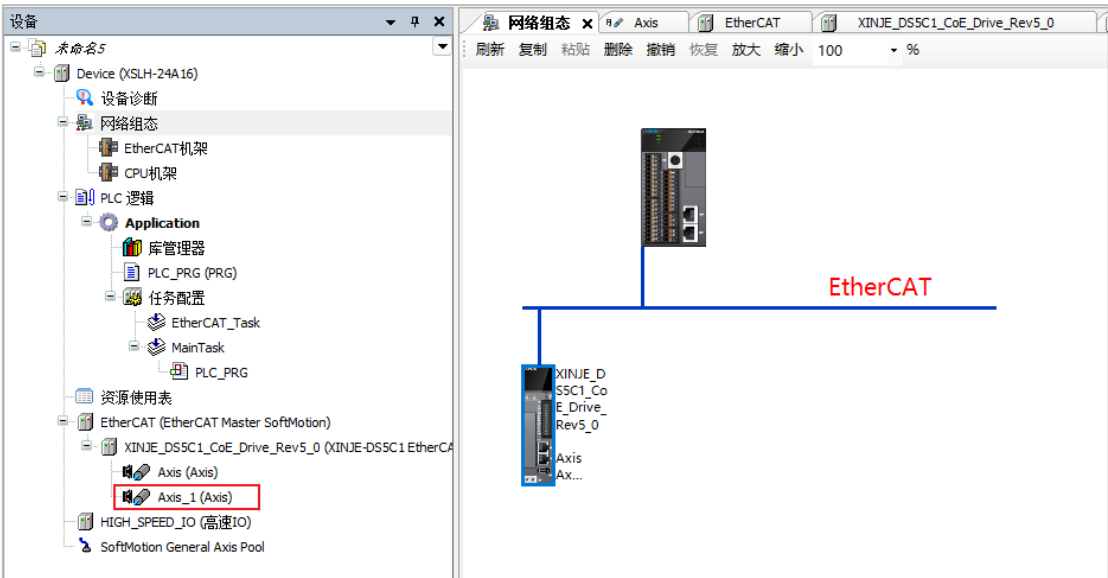


方法二：添加伺服从站后，右键菜单可添加“信捷 402 轴”。如下图所示：

添加前：



添加信捷 402 轴后：

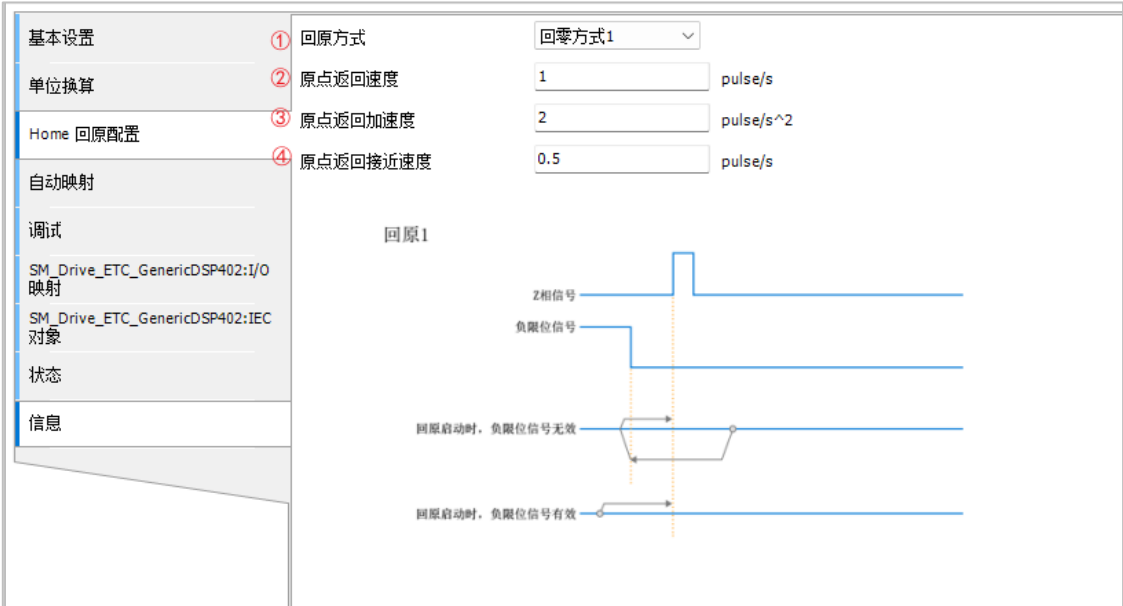


双击“轴”可打开对应轴的配置界面，如下图所示：



2、HOME回原配置

HOME 回原参数设置，主要用于轴回零图形化参数配置。提供了图形化配置指导，无需另外查阅伺服手册便可通过配置界面中的下拉菜单直接选择所需的回零方式，方便用户更加直观、便捷地完成参数配置过程。



图中的主要选项及其功能如下所述：

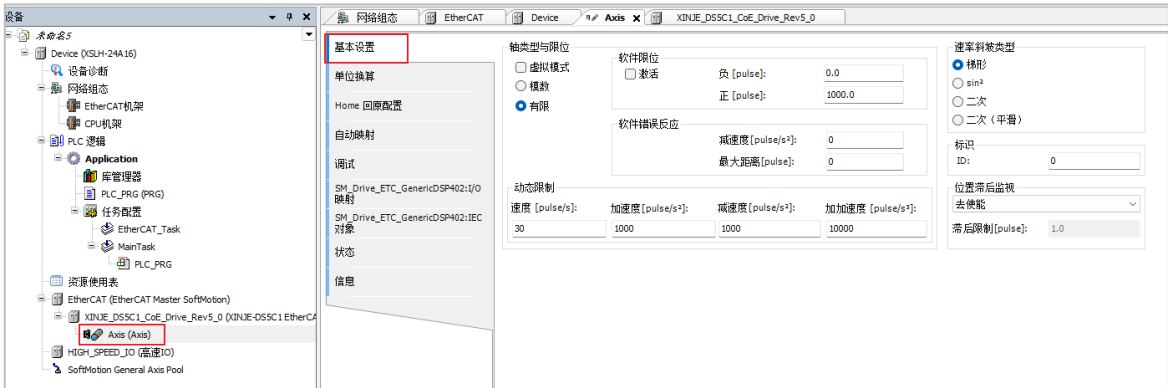
序号	功能	说明
①	回原方式 (#6098h: Home method)	配置驱动器回原点的方式，总共支持 35 种选项（实际的支持回零方式由驱动器决定）。每种不同的回零方式，下边的示例图会有所不同（示例图参考 DS5C 系列的伺服回原方式），根据需要选择不同的回零方式。
②	原点返回速度 (#6099h 子索引: 01h)	设定到 Switch 信号检出的动作的速度。
③	原点返回加速度 (#609Ah)	设定原点复位时的加速度以及减速度。
④	原点返回接近速度 (#6099h 子索引: 02h)	设定到原点检出的动作速度。



若原点返回速度 $\leq$ 原点返回接近速度则，在两个输入框的右边进行感叹号报警及信息提示。如下图所示：



5-3-3-2. 基本设置



轴类型

为了精确地控制运动位置，控制器必需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，选择“轴类型与限制”，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的错误。

没有接入实际伺服电机的场合，则选择“虚轴模式”；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，此时选择“线性模式”比较好；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，则选择“周期模式”比较好。

5-3-3-3. 单位换算

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。

基本设置

单位换算

Home 回原配置

自动映射

调试

SM_Drive_ETC_GenericDSP402:I/O 映射

SM_Drive_ETC_GenericDSP402:IEC 对象

状态

信息

用户单位

pulse

mm

°

单位换算

反转

每圈脉冲数

16#20000

脉冲/转

每圈移动量

1

pulse转

启用减速机

启用

不启用

减速机工件侧系数

1

减速机电机侧系数

1

计数类型为线性计数：

脉冲数 {pulse} = $\frac{\text{每圈脉冲数 (1)} * \text{减速机电机侧系数 (3)}}{\text{每圈移动量 (2)} * \text{减速机工件侧系数 (4)}} * \text{移动距离}$

(1)

(3)

(4)

(2)

计数类型为旋转计数：

脉冲数 {pulse} = $\frac{\text{每圈脉冲数 (1)} * \text{减速机电机侧系数 (3)}}{\text{每圈移动量 (2)} * \text{减速机工件侧系数 (4)}} * \text{移动距离}$

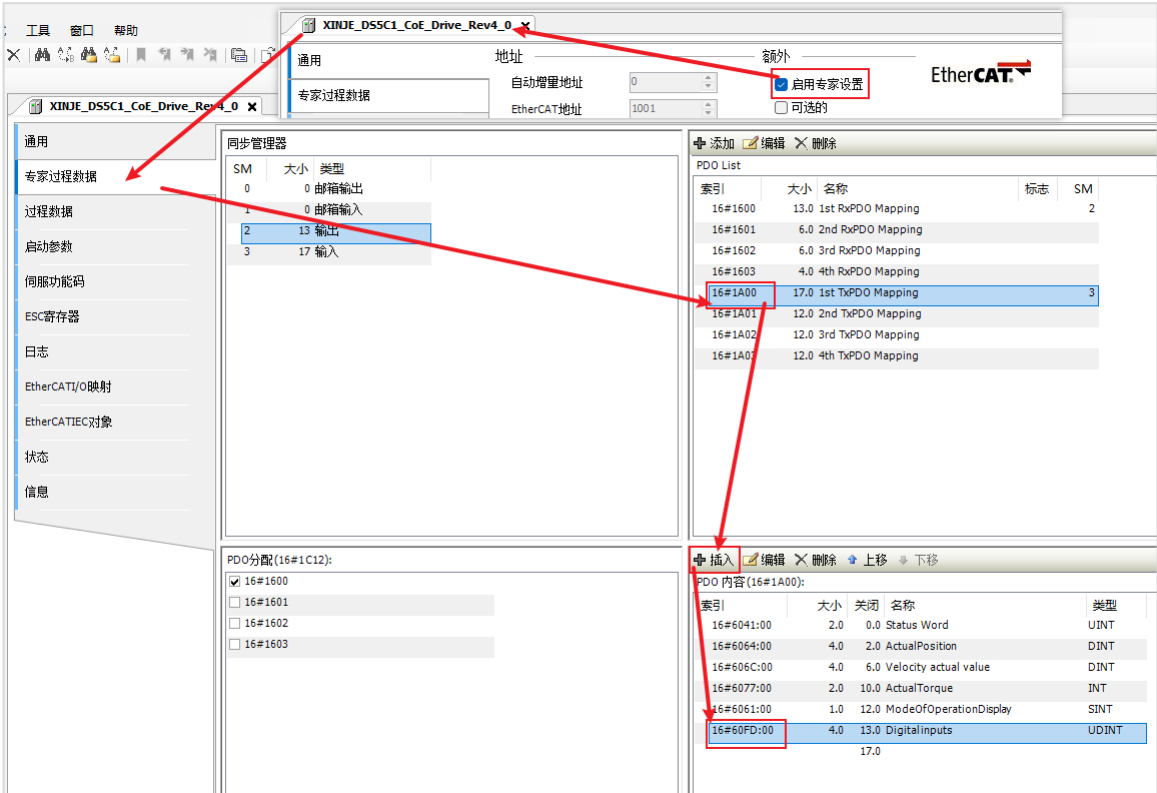
参数说明：

- ◆ 用户单位：默认值为 pluse
- ◆ 每圈脉冲数：参数范围 1-2147483647；
- ◆ 每圈移动量：默认值 1，参数范围：(0-3.4e+38]（单精度浮点数最大值）；
- ◆ 减速机：默认不启用；
- ◆ 减速机工件侧系数：默认值 1，参数范围 1-2147483647；
- ◆ 减速机电机侧系数：默认值 1，参数范围 1-2147483647。

173

5-3-3-4. 自动映射

在相应的伺服中添加 60FD 之后，在对应的轴的自动映射界面有“驱动器数字输入”的地址分配。



基本设置		Mapping																																																					
单位换算		☒ 自动映射																																																					
Home 还原配置		输入:																																																					
自动映射		<table><thead><tr><th>周期性对象</th><th>对象数</th><th>地址</th><th>类型</th></tr></thead><tbody><tr><td>status word (in.wStatusWord)</td><td>16#6041:16#00</td><td>%IW1'</td><td>UINT</td></tr><tr><td>actual position (diActPosition)</td><td>16#6064:16#00</td><td>%ID1'</td><td>DINT</td></tr><tr><td>actual velocity (diActVelocity)</td><td>16#606C:16#00</td><td>%ID2'</td><td>DINT</td></tr><tr><td>actual torque (wActTorque)</td><td>16#6077:16#00</td><td>%IW6'</td><td>INT</td></tr><tr><td>Modes of operation display (OP)</td><td>16#6061:16#00</td><td>%IB14'</td><td>SINT</td></tr><tr><td>digital inputs (in.dwDigitalInputs)</td><td>16#60FD:16#00</td><td>%ID4'</td><td>UDINT</td></tr><tr><td>Touch Probe Status</td><td>16#60B9:16#00</td><td>"</td><td>"</td></tr><tr><td>Touch Probe 1 rising edge</td><td>16#60BA:16#00</td><td>"</td><td>"</td></tr><tr><td>Touch Probe 1 falling edge</td><td>16#60BB:16#00</td><td>"</td><td>"</td></tr><tr><td>Touch Probe 2 rising edge</td><td>16#60BC:16#00</td><td>"</td><td>"</td></tr><tr><td>Touch Probe 2 falling edge</td><td>16#60BD:16#00</td><td>"</td><td>"</td></tr><tr><td>Following error (A632)</td><td>16#60F4:16#00</td><td>"</td><td>"</td></tr></tbody></table>		周期性对象	对象数	地址	类型	status word (in.wStatusWord)	16#6041:16#00	%IW1'	UINT	actual position (diActPosition)	16#6064:16#00	%ID1'	DINT	actual velocity (diActVelocity)	16#606C:16#00	%ID2'	DINT	actual torque (wActTorque)	16#6077:16#00	%IW6'	INT	Modes of operation display (OP)	16#6061:16#00	%IB14'	SINT	digital inputs (in.dwDigitalInputs)	16#60FD:16#00	%ID4'	UDINT	Touch Probe Status	16#60B9:16#00	"	"	Touch Probe 1 rising edge	16#60BA:16#00	"	"	Touch Probe 1 falling edge	16#60BB:16#00	"	"	Touch Probe 2 rising edge	16#60BC:16#00	"	"	Touch Probe 2 falling edge	16#60BD:16#00	"	"	Following error (A632)	16#60F4:16#00	"	"
周期性对象	对象数	地址	类型																																																				
status word (in.wStatusWord)	16#6041:16#00	%IW1'	UINT																																																				
actual position (diActPosition)	16#6064:16#00	%ID1'	DINT																																																				
actual velocity (diActVelocity)	16#606C:16#00	%ID2'	DINT																																																				
actual torque (wActTorque)	16#6077:16#00	%IW6'	INT																																																				
Modes of operation display (OP)	16#6061:16#00	%IB14'	SINT																																																				
digital inputs (in.dwDigitalInputs)	16#60FD:16#00	%ID4'	UDINT																																																				
Touch Probe Status	16#60B9:16#00	"	"																																																				
Touch Probe 1 rising edge	16#60BA:16#00	"	"																																																				
Touch Probe 1 falling edge	16#60BB:16#00	"	"																																																				
Touch Probe 2 rising edge	16#60BC:16#00	"	"																																																				
Touch Probe 2 falling edge	16#60BD:16#00	"	"																																																				
Following error (A632)	16#60F4:16#00	"	"																																																				
调试		输出:																																																					
SM_Drive_ETC_GenericDSP402:I/O 映射		<table><thead><tr><th>周期性对象</th><th>对象数</th><th>地址</th><th>类型</th></tr></thead><tbody><tr><td>ControlWord (out.wControlWord)</td><td>16#6040:16#00</td><td>%QW0'</td><td>UDINT</td></tr><tr><td>set position (diSetPosition)</td><td>16#607A:16#00</td><td>%QD1'</td><td>DINT</td></tr><tr><td>set velocity (diSetVelocity)</td><td>16#60FF:16#00</td><td>%QD2'</td><td>DINT</td></tr><tr><td>set torque (wSetTorque)</td><td>16#6071:16#00</td><td>%QW6'</td><td>INT</td></tr><tr><td>Modes of operation (OP)</td><td>16#6060:16#00</td><td>%QB14'</td><td>SINT</td></tr><tr><td>Touch Probe Function</td><td>16#60B8:16#00</td><td>"</td><td>"</td></tr><tr><td>Add velocity value</td><td>16#60B1:16#00</td><td>"</td><td>"</td></tr><tr><td>Add torque value</td><td>16#60B2:16#00</td><td>"</td><td>"</td></tr><tr><td>Digital outputs (A637)</td><td>16#60FE:16#01</td><td>"</td><td>"</td></tr></tbody></table>		周期性对象	对象数	地址	类型	ControlWord (out.wControlWord)	16#6040:16#00	%QW0'	UDINT	set position (diSetPosition)	16#607A:16#00	%QD1'	DINT	set velocity (diSetVelocity)	16#60FF:16#00	%QD2'	DINT	set torque (wSetTorque)	16#6071:16#00	%QW6'	INT	Modes of operation (OP)	16#6060:16#00	%QB14'	SINT	Touch Probe Function	16#60B8:16#00	"	"	Add velocity value	16#60B1:16#00	"	"	Add torque value	16#60B2:16#00	"	"	Digital outputs (A637)	16#60FE:16#01	"	"												
周期性对象	对象数	地址	类型																																																				
ControlWord (out.wControlWord)	16#6040:16#00	%QW0'	UDINT																																																				
set position (diSetPosition)	16#607A:16#00	%QD1'	DINT																																																				
set velocity (diSetVelocity)	16#60FF:16#00	%QD2'	DINT																																																				
set torque (wSetTorque)	16#6071:16#00	%QW6'	INT																																																				
Modes of operation (OP)	16#6060:16#00	%QB14'	SINT																																																				
Touch Probe Function	16#60B8:16#00	"	"																																																				
Add velocity value	16#60B1:16#00	"	"																																																				
Add torque value	16#60B2:16#00	"	"																																																				
Digital outputs (A637)	16#60FE:16#01	"	"																																																				
EtherCAT I/O 映射		驱动器数字输入:																																																					
EtherCAT IEC 对象		<table><thead><tr><th>周期性对象</th><th>对象数</th><th>地址</th><th>类型</th></tr></thead><tbody><tr><td>Negative limit switch</td><td>16#60FD:16#00-Bit0</td><td>%ID4.0'</td><td>Bit</td></tr><tr><td>Positive limit switch</td><td>16#60FD:16#00-Bit1</td><td>%ID4.1'</td><td>Bit</td></tr><tr><td>Home switch</td><td>16#60FD:16#00-Bit2</td><td>%ID4.2'</td><td>Bit</td></tr><tr><td>Touch probe 1</td><td>16#60FD:16#00-Bit3</td><td>%ID4.3'</td><td>Bit</td></tr><tr><td>Touch probe 2</td><td>16#60FD:16#00-Bit4</td><td>%ID4.4'</td><td>Bit</td></tr><tr><td>zero impulse signal output</td><td>16#60FD:16#00-Bit5</td><td>%ID4.5'</td><td>Bit</td></tr></tbody></table>		周期性对象	对象数	地址	类型	Negative limit switch	16#60FD:16#00-Bit0	%ID4.0'	Bit	Positive limit switch	16#60FD:16#00-Bit1	%ID4.1'	Bit	Home switch	16#60FD:16#00-Bit2	%ID4.2'	Bit	Touch probe 1	16#60FD:16#00-Bit3	%ID4.3'	Bit	Touch probe 2	16#60FD:16#00-Bit4	%ID4.4'	Bit	zero impulse signal output	16#60FD:16#00-Bit5	%ID4.5'	Bit																								
周期性对象	对象数	地址	类型																																																				
Negative limit switch	16#60FD:16#00-Bit0	%ID4.0'	Bit																																																				
Positive limit switch	16#60FD:16#00-Bit1	%ID4.1'	Bit																																																				
Home switch	16#60FD:16#00-Bit2	%ID4.2'	Bit																																																				
Touch probe 1	16#60FD:16#00-Bit3	%ID4.3'	Bit																																																				
Touch probe 2	16#60FD:16#00-Bit4	%ID4.4'	Bit																																																				
zero impulse signal output	16#60FD:16#00-Bit5	%ID4.5'	Bit																																																				
状态																																																							
信息																																																							

5-3-3-5. SM_Drive_ETC_GenericDSP402:I/O 映射



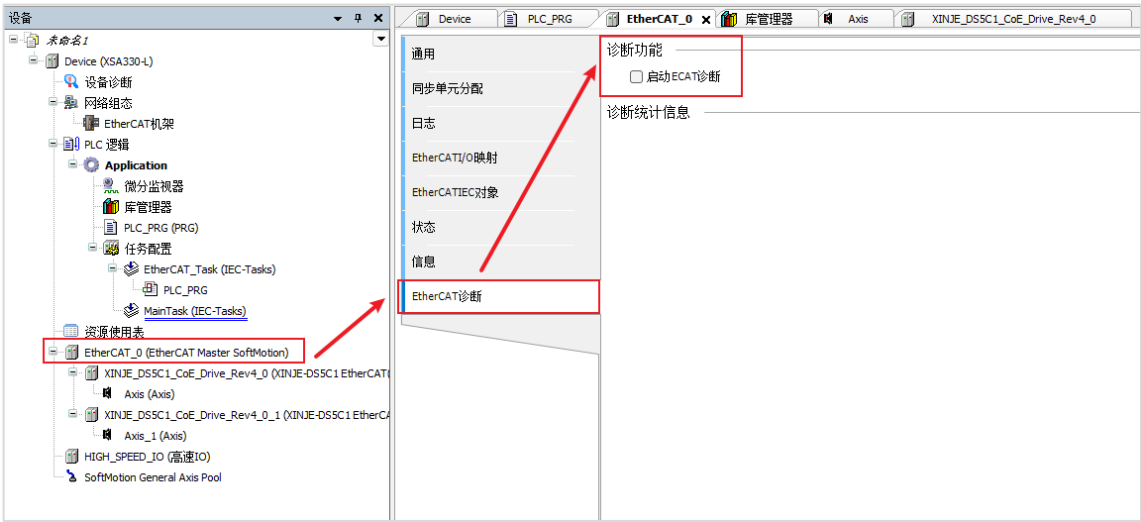
5-3-4. EtherCAT 诊断

5-3-4-1. EtherCAT 诊断功能适用设备

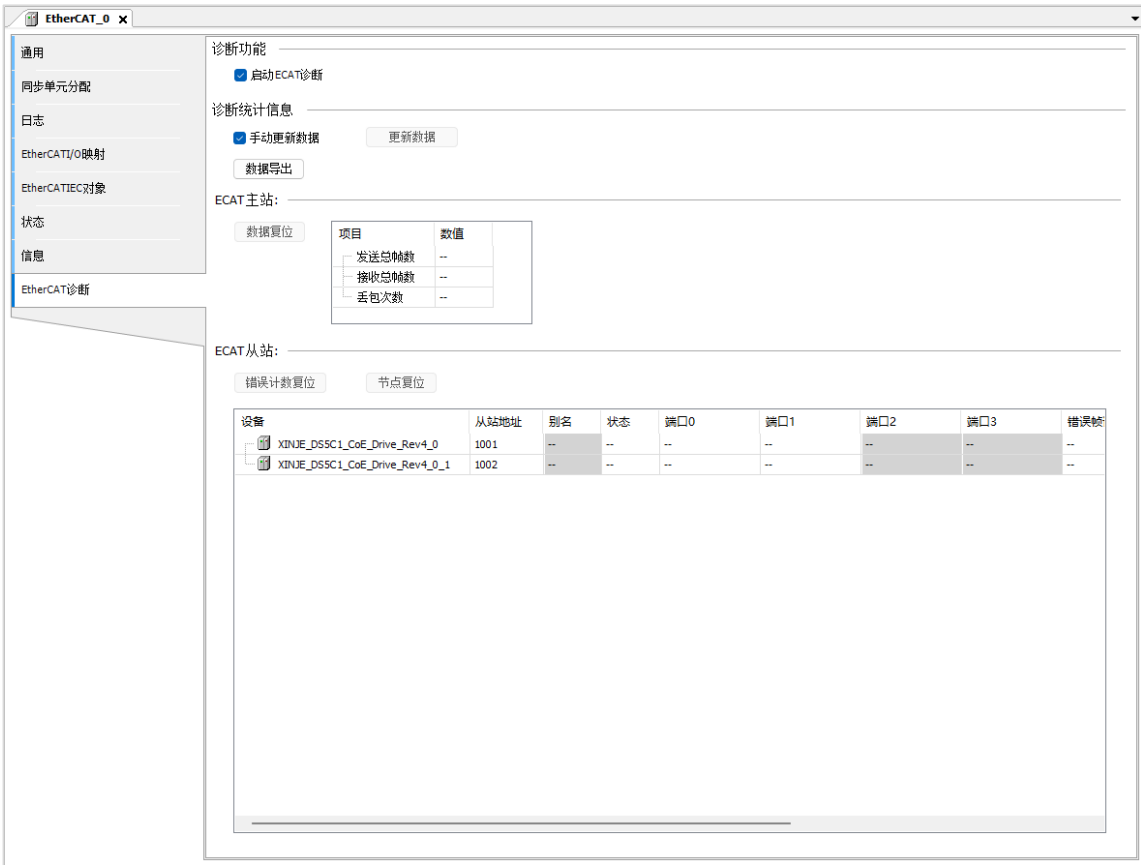
XS 机型及版本	功能	EtherCAT 诊断 (XS StudioV2. 3. 2 及以上支持)
XSDH		固件版本 V2.4.2c 及以上支持
XSLH		固件版本 V2.4.2c 及以上支持
XSF5		固件版本 2.4.2 及以上支持
XSA-W		不支持
XSA-L		固件版本 2.4.2 及以上支持

5-3-4-2. EtherCAT 诊断功能设置

在EtherCAT主站界面可选择启用EtherCAT诊断功能。



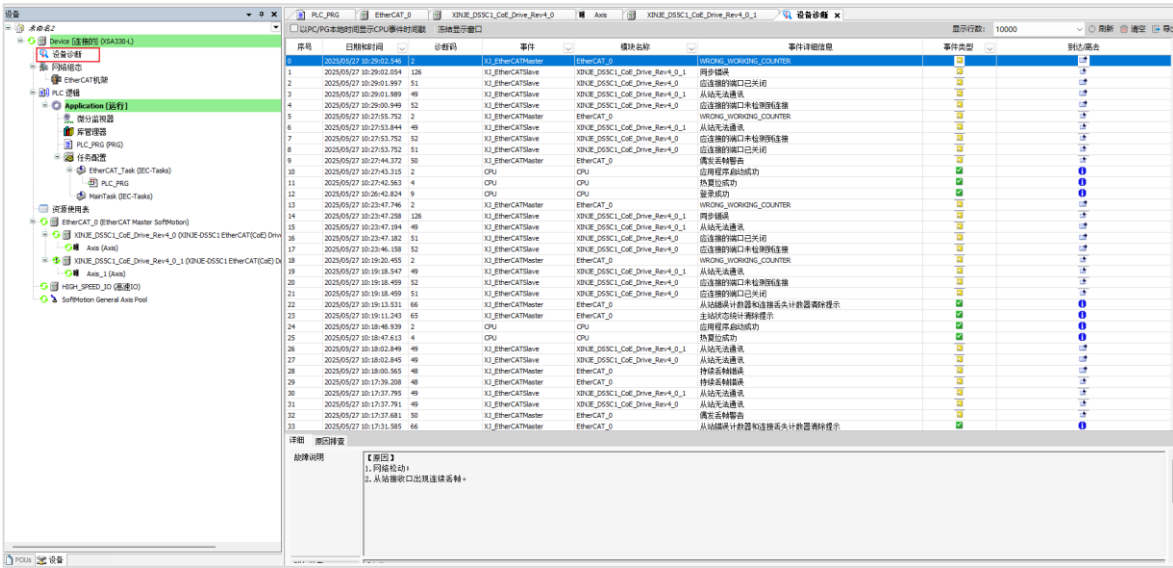
5-3-4-3. EtherCAT 诊断界面参数介绍



部分	参数名称	参数含义
诊断功能	启动ECAT诊断	是否启动ECAT诊断（默认：空） 打勾：启动ECAT诊断； 空：不启动ECAT诊断（诊断统计信息不显示）
	手动更新数据	是否选择手动更新数据（默认：空） 打勾：手动更新数据（“更新数据”按钮显示）； 空：诊断统计信息自动更新
诊断数据信息	更新数据	对诊断数据刷新一次

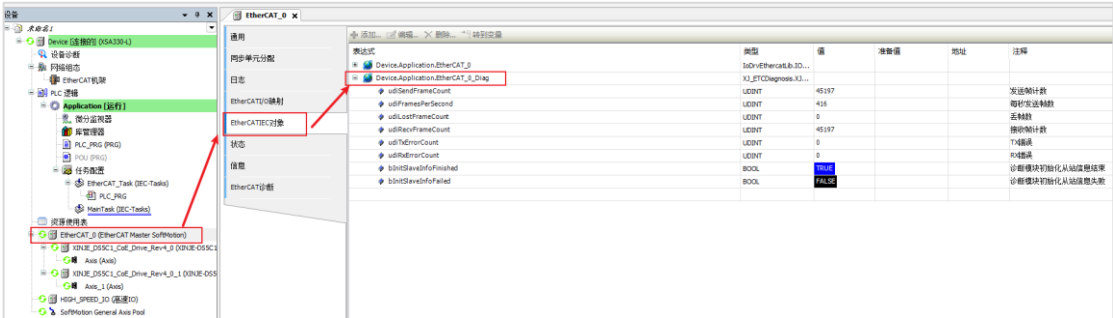
部分	参数名称	参数含义
	数据导出	导出表格（主站，从站）信息，以表格方式
ECAT主站	数据复位	清除ECAT主站所有统计数据
ECAT从站	错误计数复位	清除“错误帧计数”和“连接丢失计数”
	节点复位	复位节点状态（重启ECAT链路）
	设备	显示主站下所有从站的名称
	从站地址	显示从站的地址信息
	别名	显示（只读）从站别名信息（没有设定的显示灰色）
	状态	初始化：从站在初始化状态 预运行：从站在预运行状态； 安全运行：从站在安全运行状态； 运行：从站在运行状态； 引导：从站在引导状态； 无通信：无法与从站通信
	错误帧计数	端口0错误计数
	连接丢失计数	端口0链接断开计数
	故障信息	从站（节点）故障简单描述

在勾选“启动 ECAT 诊断”后相关节点报错信息显示在“设备诊断”界面中。



5-3-4-4. EtherCAT 主站诊断界面参数调用

在Ethercat主站中的Ethercat IEC对象中的ETHerCAT0_Diag中可以看到诊断的错误信息，并可在程序中调用监控。



参数如下：

参数名称	参数类型	参数含义
udiSendFrameCount	UDINT	发送帧计数
udiFramesPerSecond	UDINT	每秒发送帧数
udiLostFrameCount	UDINT	丢帧数
udiRecvFrameCount	UDINT	接收帧计数
udiTxErrorCount	UDINT	TX错误
udiRxErrorCount	UDINT	RX错误
bInitSlaveInfoFinished	BOOL	诊断模块初始化从站信息结束
bInitSlaveInfoFailed	BOOL	诊断模块初始化从站信息失败

 举例：

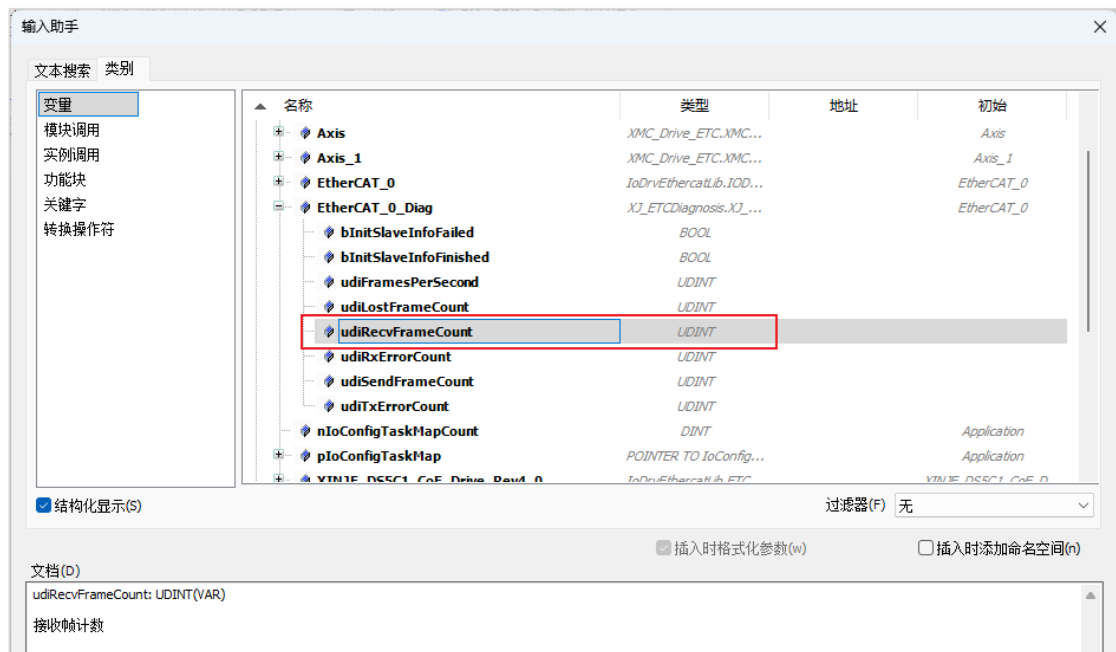
在程序中读取 Ethercat 主站诊断信息中的“接收帧计数”参数，首先定义一个 UDINT 变量，将 ETHERCAT0_Diag 结构体中的 udiRecvFrameCount 参数和定义的变量关联起来：

```

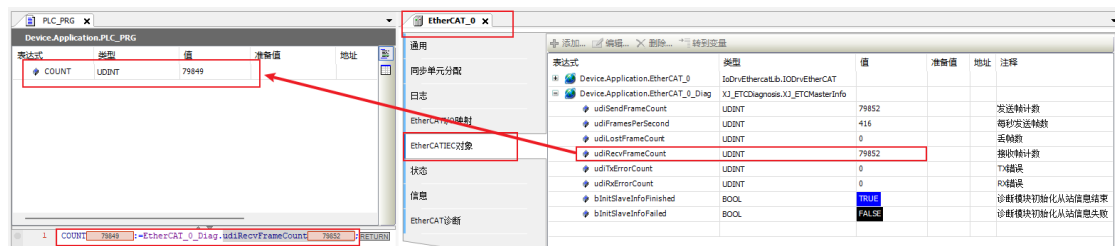
1  PROGRAM PLC_PRG
2  VAR
3      COUNT:UDINT;
4  END_VAR

1  COUNT:=EtherCAT_0_Diag.udiRecvFrameCount;

```

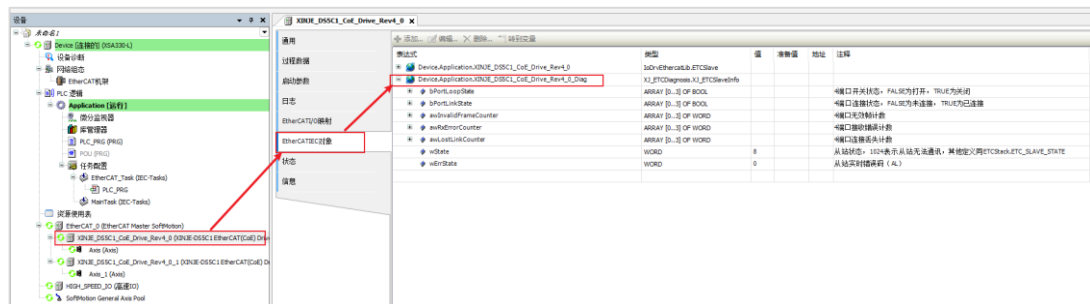


执行效果如下，可以在设置的变量中读取到 ETHERCAT0_Diag 结构体中的 udiRecvFrameCount 的值：



5-3-4-5. EtherCAT 从站诊断界面参数调用

以信捷的DS5C1为例，在XINJE_DS5C1_CoE_Drive_Rev4_0中Ethercat IEC对象中的XINJE_DS5C1_CoE_Drive_Rev4_0_Diag中可以看到诊断的错误信息，并可在程序中调用监控。



参数如下:

参数名称	参数类型	参数含义
awInvalidFrameCounter	ARRAY [0..3] OF BOOL	4端口开关状态，FALSE为打开，TRUE为关闭
awLostLinkCounter	ARRAY [0..3] OF BOOL	4端口连接状态，FALSE为未连接，TRUE为已连接
awRxErroCounter	ARRAY [0..3] OF WORD	4端口无效帧计数
bPortLinkState	ARRAY [0..3] OF WORD	4端口接收错误计数
bPortLoopState	ARRAY [0..3] OF WORD	4端口连接丢失计数
wState	WORD	从站状态，1024表示从站无法通讯等
wErrState	WORD	从站实时错误码

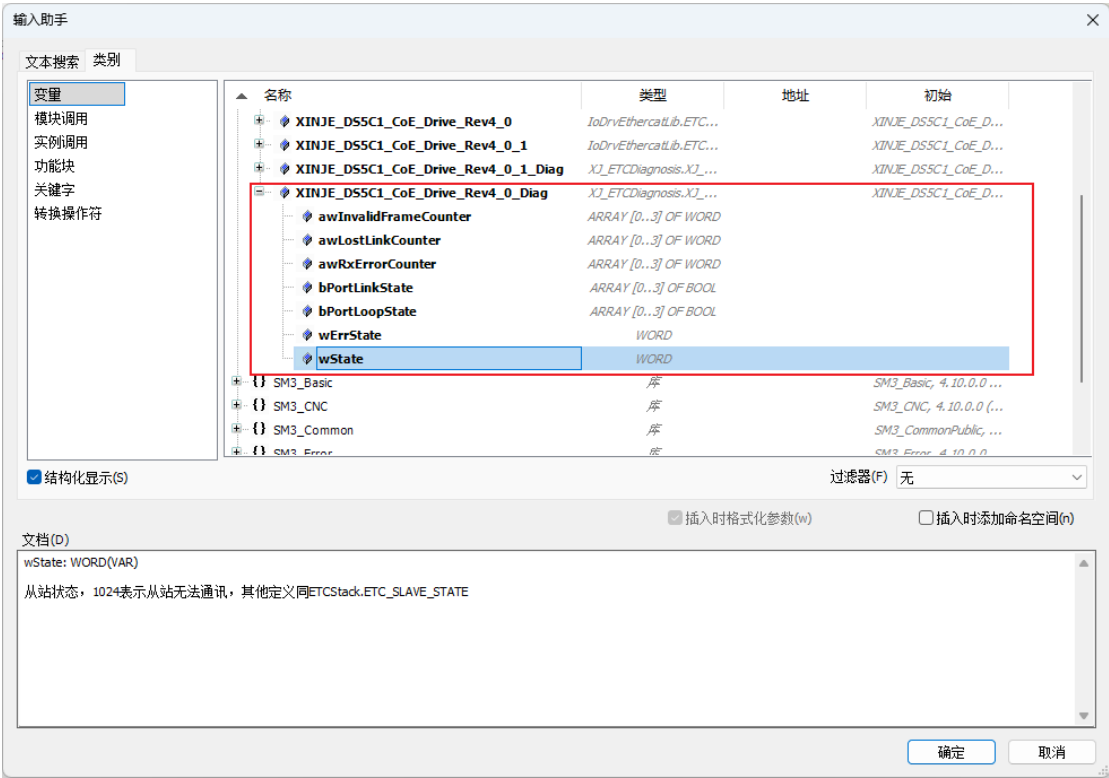


举例:

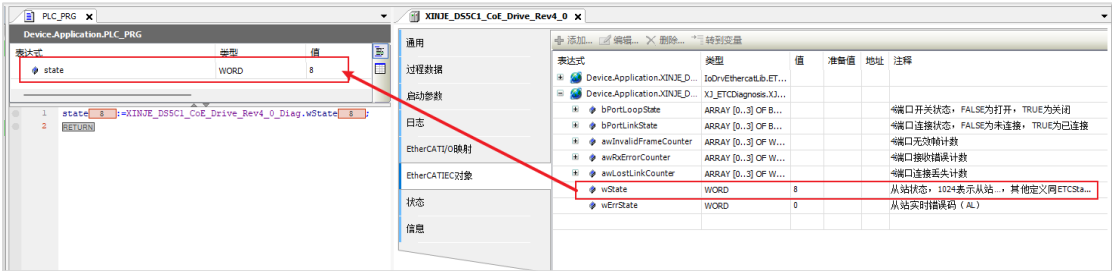
在程序中读取 Ethercat 从站诊断信息中的“从站状态”参数，首先定义一个 WORD 变量，将 XINJE DS5C1 CoE Drive Rev4 0 Diag 结构体中的 wState 参数和定义的变量关联起来：

```
PROGRAM PLC_PRG
VAR
    state:WORD;
END_VAR

state:=XINJE_DS5C1_CoE_Drive_Rev4_0_Diag.wState;
```



执行效果如下，可以在设置的变量中读取到 XINJE_DS5C1_CoE_Drive_Rev4_0_Diag 结构体中的 wState 的值：



5-3-4-6. 报错代码

诊断码		错误描述
1	0x01	无通讯
2	0x02	工作计数器错误
3	0x03	总线DC时钟错误
4	0x04	打开第一个网络适配器失败
5	0x05	打开第二个网络适配器失败
6	0x06	两个网络适配器MAC重复
7	0x07	初始化所有从站失败
8	0x08	从站供应商ID与配置不一致
9	0x09	从站产品ID与配置不一致
10	0x0A	从站数量与配置不一致
11	0x0B	写SDO错误
12	0x0C	写SDO超时
13	0x0D	设备告警

诊断码		错误描述
14	0x0E	SOE下载失败
15	0x0F	SOE下载超时
16	0x10	请求状态机超时
17	0x11	从站IN/OUT连接错误
48	0x30	持续丢帧故障
49	0x31	从站无法通讯
50	0x32	偶发丢帧警告
51	0x33	应连接的端口已关闭
52	0x34	应连接的端口未检测到连接
53	0x35	从站站号重复
101	0x65	从站未知错误
102	0x66	从站邮箱申请内存失败
117	0x75	无效的状态机请求
118	0x76	未知的状态机请求
119	0x77	从站不支持引导模式
120	0x78	固件程序无效
121	0x79	引导状态下检测到邮箱配置错误
122	0x7A	预操作状态下检测到邮箱配置错误
123	0x7B	同步管理器配置无效
124	0x7C	输入数据无效
125	0x7D	输出数据无效
126	0x7E	同步错误
127	0x7F	同步管理器看门狗超时
128	0x80	同步管理器类型无效
129	0x81	输出PDO配置无效
130	0x82	输入PDO配置无效
131	0x83	看门狗配置无效
132	0x84	从站需要冷启动
133	0x85	从站需要初始化状态
134	0x86	从站需要预操作状态
135	0x87	从站需要安全操作状态
136	0x88	输入PDO映射无效
137	0x89	输出PDO映射无效
138	0x8A	从站不一致的设置
139	0x8B	从站不支持Free-Run模式
140	0x8C	从站不支持同步运行模式
141	0x8D	Free-Run模式需要配置3个缓冲区
142	0x8E	后台程序看门狗超时
143	0x8F	输入和输出无效
144	0x90	重要同步错误
145	0x91	未检测到同步信号
148	0x94	DC同步配置无效
149	0x95	DC锁存配置无效
150	0x96	PLL错误
151	0x97	DC IO错误

诊断码		错误描述
152	0x98	DC超时错误
153	0x99	DC同步周期无效
154	0x9A	Sync0周期错误
155	0x9B	Sync1周期错误
165	0xA5	从站AOE邮箱错误
166	0xA6	从站EOE邮箱错误
167	0xA7	从站COE邮箱错误
168	0xA8	从站FOE邮箱错误
169	0xA9	从站SOE邮箱错误
179	0xB3	从站VOE邮箱错误
180	0xB4	从站EEPROM无法访问
181	0xB5	从站EEPROM错误
196	0xC4	从站已在本地重新启动

5-3-4-7. 应用举例

以 XSA330-L 带两个信捷 DS5C1 伺服为例，不同情况下诊断信息的提示如下：

情况一：在正常运行过程中断开 XSA330-L 的 ethercat 口的网线或断开第一台 DS5C1 的 IN 口的网线

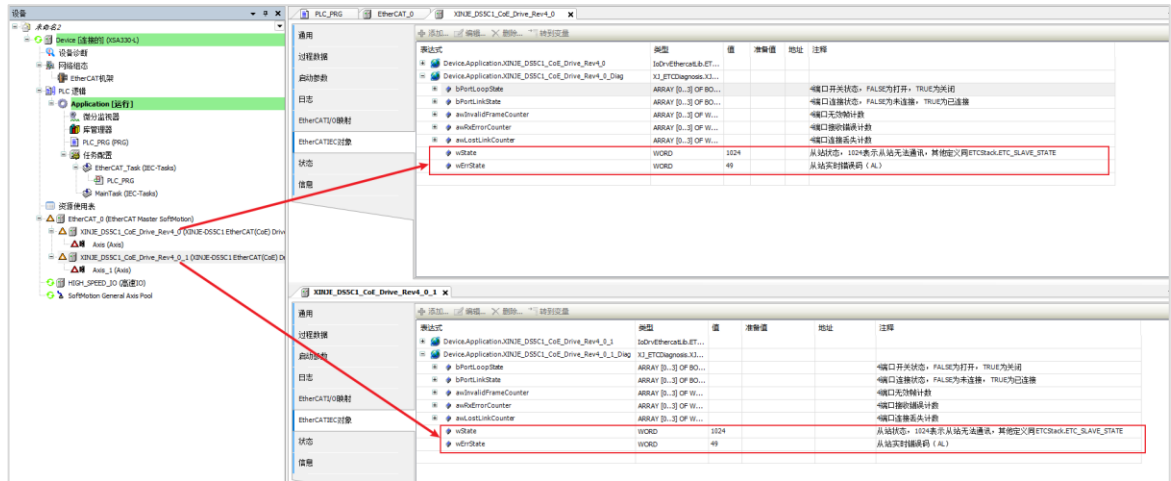
正常运行时，在 ethercat 主站的诊断信息中显示两个从站通信正常：



此时断开 XSA330-L 的 ethercat 口的网线，在 ethercat 主站的诊断界面中 ETH 主站中丢包数累加，ETH 从站显示两个从站状态为无连接，故障信息为从站无法通讯。



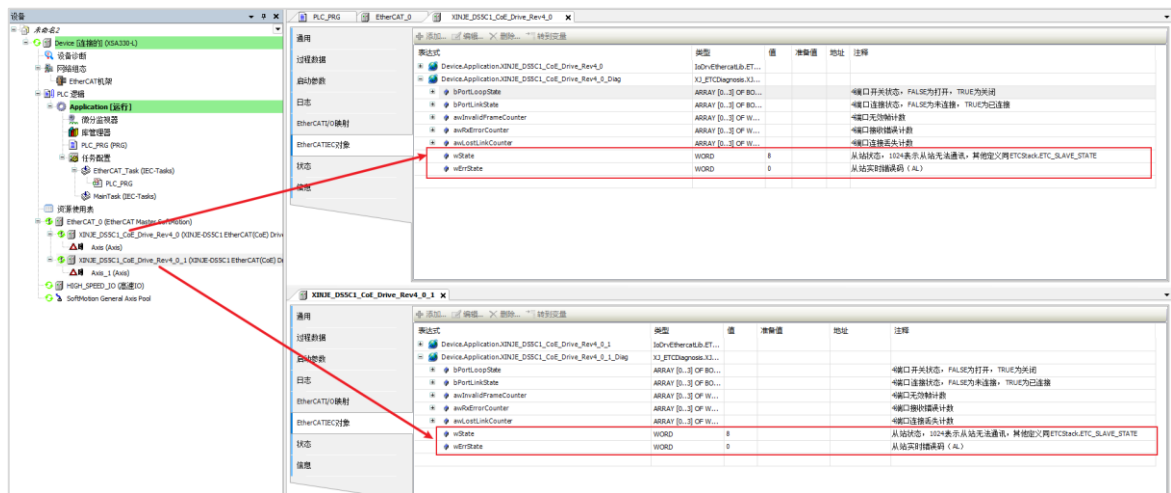
在两个 ethercat 从站的诊断界面显示从站状态为 1024（无法通讯），从站实时错误码为 49（从站无法通讯）。



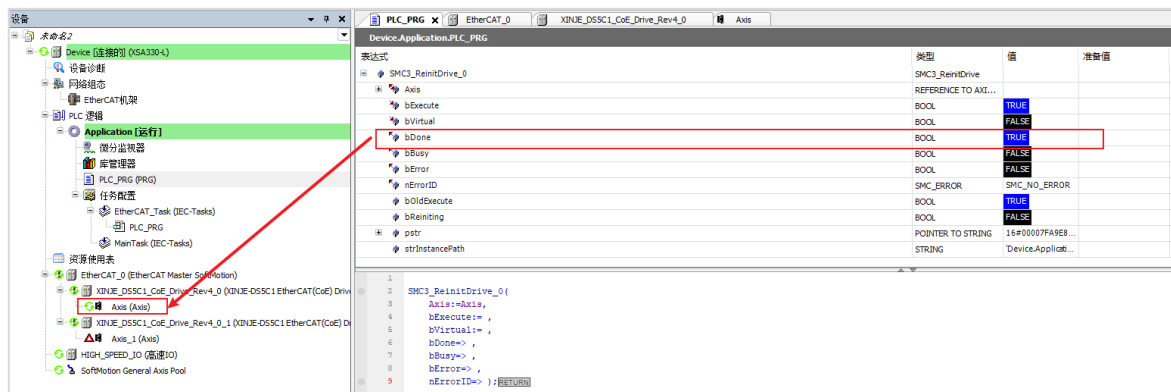
勾选自动重连之后, 把网线插回, 此时 ethercat 主站诊断信息中显示两个从站状态为运行, 故障信息为通信正常。



两个 ethercat 从站的诊断信息显示从站状态为 8, 从站实时错误码为 0。



此时轴处于 errstop 的状态, 需要对轴重新调用后正常使用。



情况二：在正常运行过程中断开第一台 DS5C1 的 OUT 口的网线或断开第二台 DS5C1 的 IN 口网线。

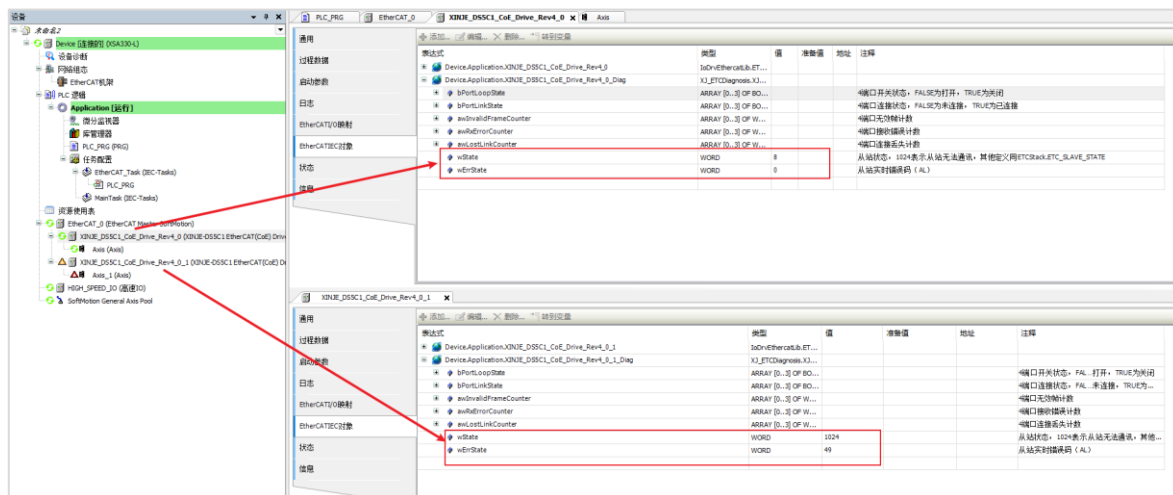
正常运行时，在 ethercat 主站的诊断信息中显示两个从站通信正常：



此时断开第一台 DS5C1 的 OUT 口的网线，在 ethercat 主站的诊断界面中 ETH 主站中丢包数不变，ETH 从站显示第一个从站状态为运行，第二个从站状态显示无通信，故障信息为从站无法通讯。



在两个 ethercat 从站的诊断界面，第一个从站显示从站状态为 8（正常通讯），第二个从站状态为 49（从站无法通讯）

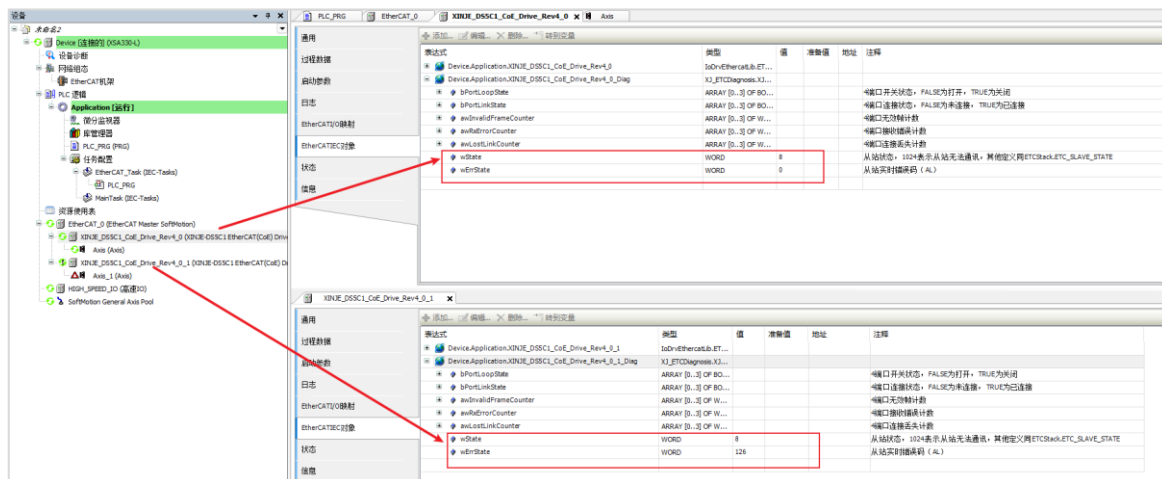


勾选自动重连之后，把网线插回，此时 ethercat 主站诊断信息中显示第一个从站状态为运行，第二个从站状态为运行，故障信息为同步错误。

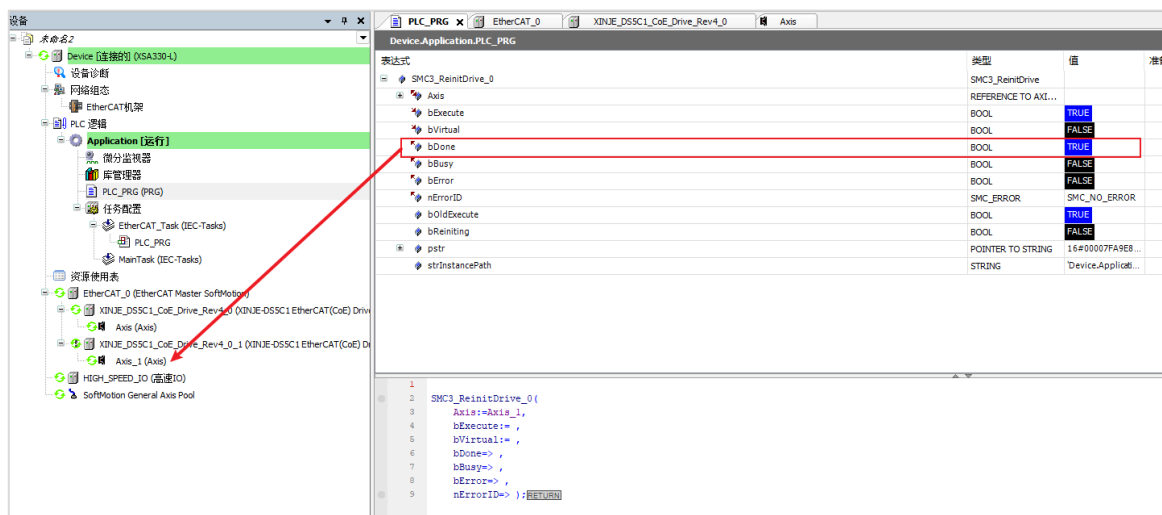


第一个 ethercat 从站的诊断信息显示从站状态为 8，从站实时错误码为 0；

第二个 ethercat 从站的诊断信息显示从站状态为 8，从站实时错误码为 126（同步错误）。



此时轴处于 errstop 的状态，需要对轴重新调用后正常使用。



5-3-5. EtherCAT 控制工程

5-3-5-1. 运动工程控制

在一个工程中，程序中所用到的指令，都需要来自文件库的支持。而每一个 POU，如果不在某个任务中调用的话，便不会执行。用户可以选择直接配置到某一个任务之中去执行，或者选择在另外一个已经在任务中的 POU 对该为配置任务的 POU 进行调用。POU 中执行的程序，如果需要和外部的 IO 或总线进行交互，则需要在程序中另外配置相应的高速 IO 模块或 EtherCAT 总线及从站设备。

5-3-5-2. 多个 POU 用法

在编写应用程序时，不同执行周期的程序功能，应放到不同的 POU 中进行编写，并配置到不同优先级和周期时间的任务之中，以便于后续程序的查看和优化。

- 合理分配 CPU 资源，按每个功能需要的周期时间去进行周期的分配；
- 程序结构清晰，各功能区分清楚，相较于将所有程序堆叠在一起，多个 POU 用法可通过不同的命名区分功能，体现在工程栏上，程序的逻辑结构一目了然；
- 调试方便，在调试时，可以很方便地对某部分需要屏蔽的功能进行屏蔽；
- 可以实现在不同工程之间，对 POU 进行直接的引用，直接将 POU 从工程 1 复制到工程 2；
- 将程序规划清楚后，可以分给多个人进行编程开发，提高编程的效率；
- 可以在不同的 POU 使用不同的编程语言，只要保证接口清晰，对 POU 内的编程语言没有统一要求。

5-3-5-3. 运动功能的调用方式

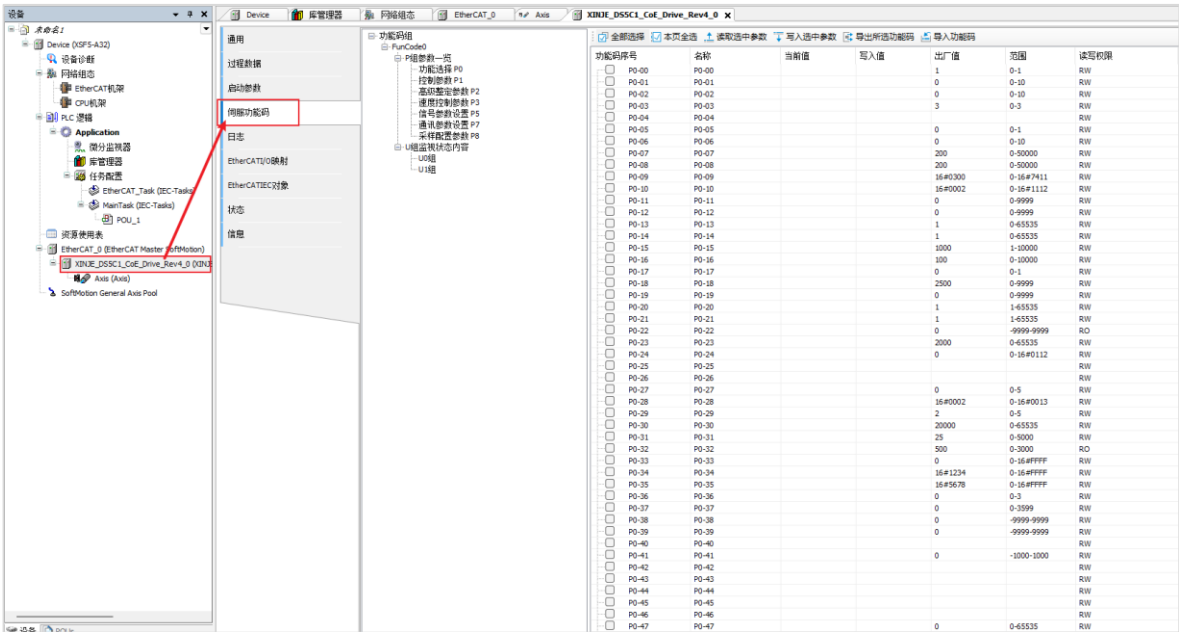
在一个工程中，为了更合理地分配 CPU 的资源，编程时会将不同周期的程序放到不同的 POU 和任务之中。

运动功能需要最高优先级的任务，而逻辑功能一般不需要那么高优先级的任务配置，因此，在实际工程中，通常这两块是放在两个不同的 POU 与任务之中的。那么，要怎么实现即使运动功能与逻辑功能分开，依然能够在逻辑功能之中去控制一个运动功能的执行呢？一般都是在运动功能中定义一些输入变量和输出变量，给其它功能进行调用，如在逻辑 POU 中，需要调用运动功能，就往运动 POU 的输入变量写入控制数据，运动 POU 将运动状态放到输出变量中，给到逻辑 POU，用以判断运动状态，判断程序逻辑的执行。

5-4. 伺服参数调试

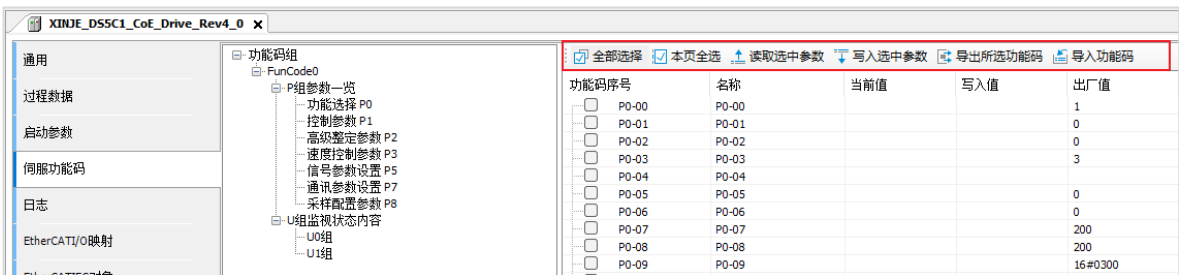
5-4-1. 伺服功能码

在“Ethercat 从站”---“伺服功能码”中，针对伺服从站提供读写、导入和导出厂家参数调试界面。

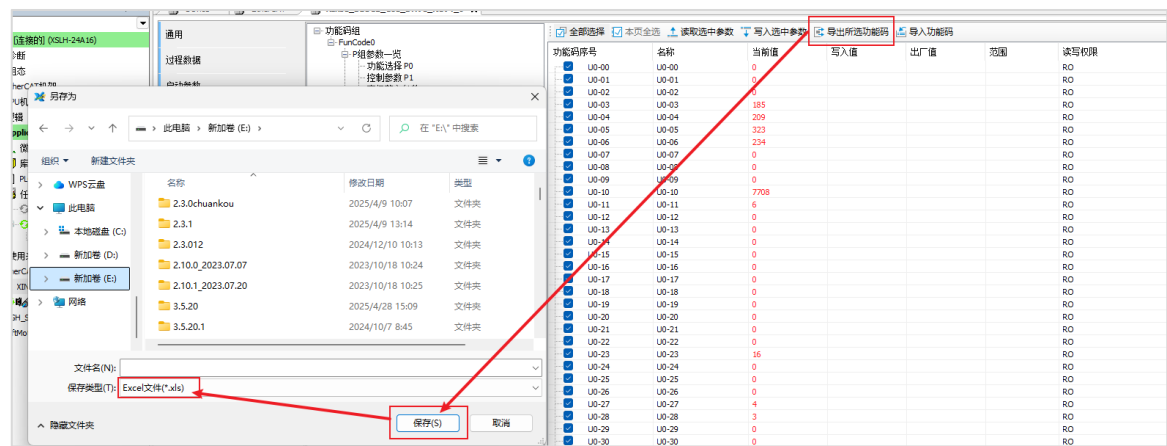


伺服功能码从伺服 XML 解析而来，且该功能仅所有信捷伺服支持。

5-4-1-1. 界面功能



- “全部选择”：可以全部选择伺服功能码和“取消”全部选择项；
- “本页全选”：只针对当前页有效，全选和取消全选；
- “读取选中参数”：只有从站在运行状态时，且项目属性支持读属性时，才可以读取；
- “写入选中参数”：只有从站在运行状态时，且项目属性支持写属性时，才可以写入；
- “导出所选功能码”：导出选择的功能码。



以.xls 格式导出展示，表格内容参考如下：

	A	B	C	D	E	F	G
1	功能码序号	名称	当前值	写入值	出厂值	范围	读写权限
2	P0-00	P0-00	1	1	1	0-1	RW
3	P0-01	P0-01	8	8	0	0-10	RW
4	P0-02	P0-02	8	8	0	0-10	RW
5	P0-03	P0-03	3	3	3	0-3	RW
6	P0-04	P0-04	20	20			RW
7	P0-05	P0-05	0	0	0	0-1	RW
8	P0-06	P0-06	0	0	0	0-10	RW
9	P0-07	P0-07	0	0	200	0-50000	RW
10	P0-08	P0-08	200	200	200	0-50000	RW
11	P0-09	P0-09	3840	3840	16#0300	0-16#7411	RW
12	P0-10	P0-10	258	258	16#0002	0-16#1112	RW
13	P0-11	P0-11	0	0	0	0-9999	RW

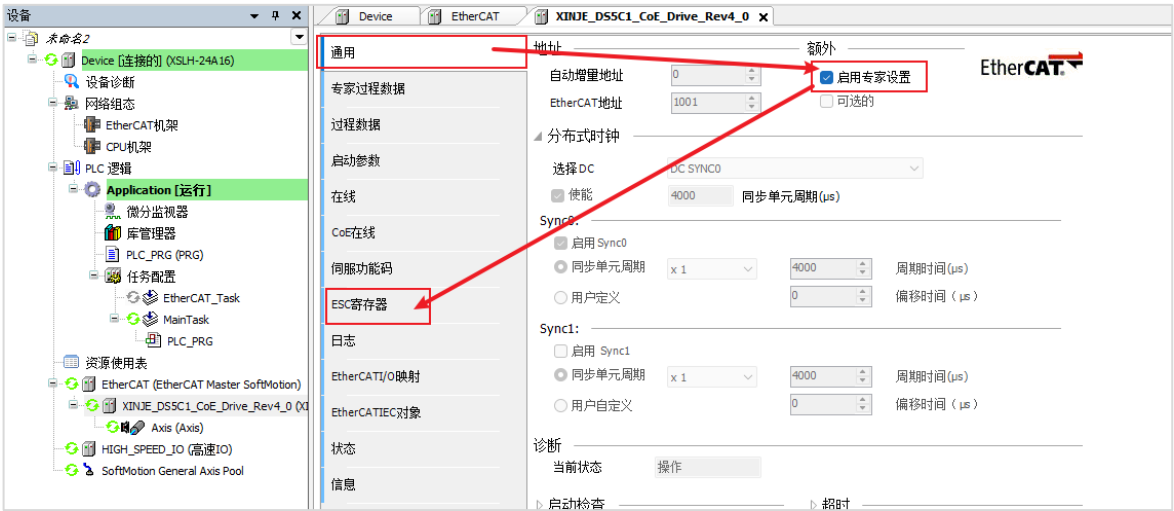
- “导入功能码”：从外部导入功能码。

导入时根据“功能码序号”识别，仅导入“写入值”，用户根据需求是否覆盖到写入值。

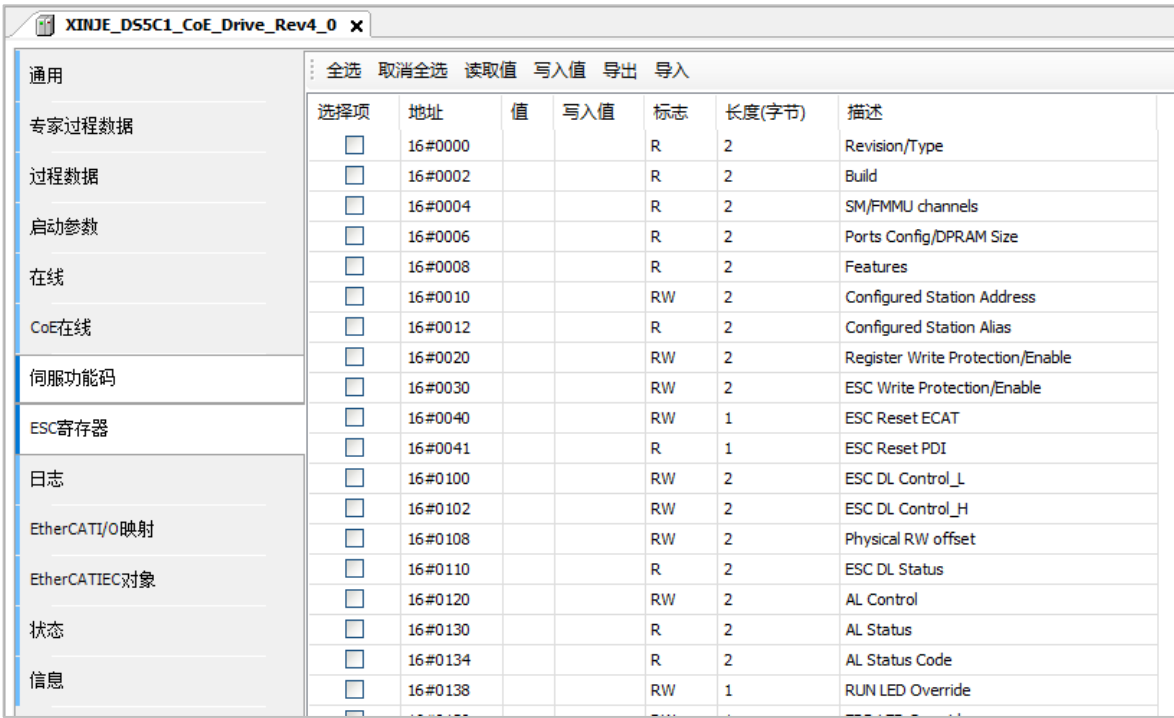


5-4-2. ESC 寄存器

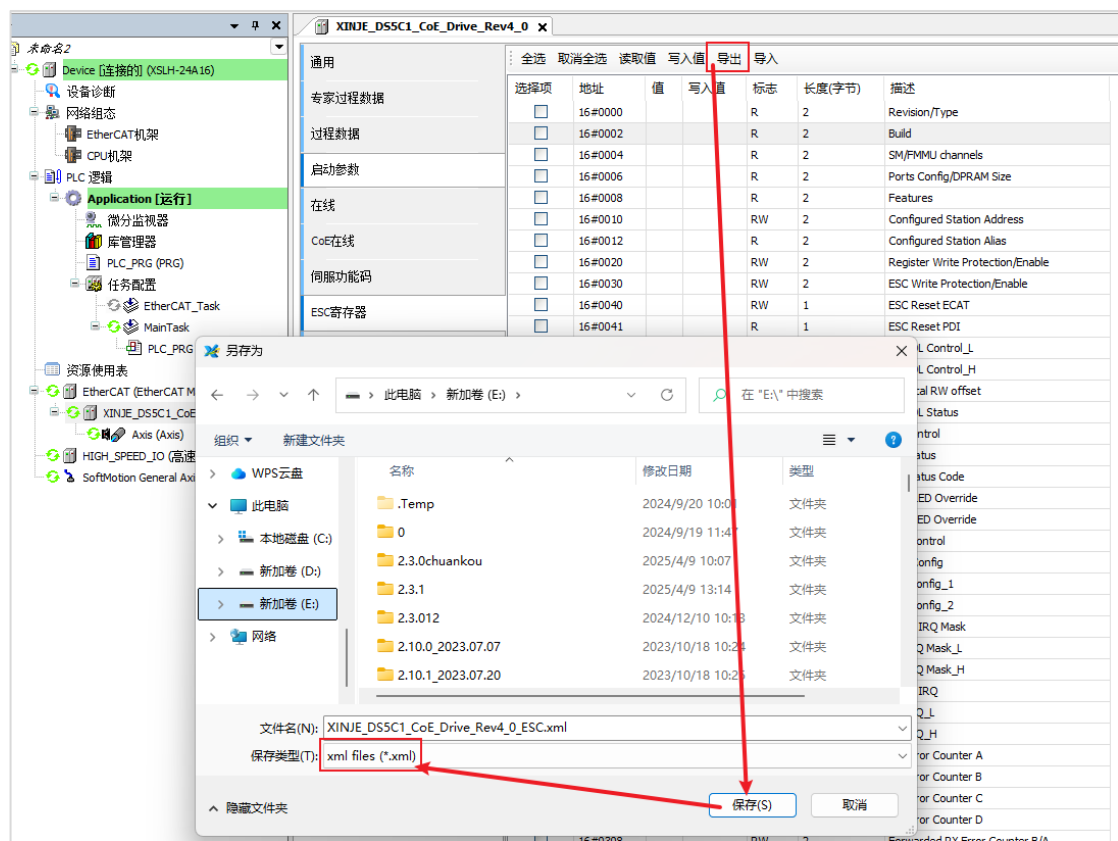
在勾选“启用专家设置”后显示“ESC 寄存器”界面。



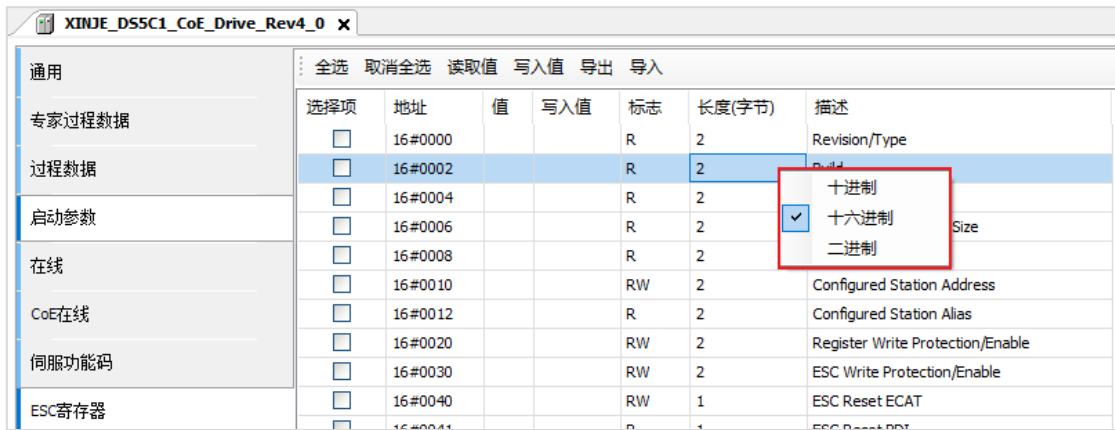
5-4-2-1. 界面功能



- “全选”：全部选择。
- “取消全选”：取消全部的选择项。
- “读取值”：在运行状态下，可以读取选择项的值。
- “写入值”：在运行状态下，可以写入具有写入属性的值。
- “导出”：以 XML 格式导出选择的项。



- “导入”：导入外部符合格式要求的 XML 文件，只显示导入 XML 项。
- 数值显示形式：“右键” --- 可以进行十六进制、十进制和二进制切换。



- 打开老工程，新增显示伺服功能码，ESC 寄存器界面在勾选“使能专家设置”时显示。
- 新建工程，显示伺服功能码，ESC 寄存器界面在勾选“使能专家设置”时显示。
- “伺服功能码”需要信捷设备及支持上述功能的 XML 才显示。
- “ESC 寄存器”对于所有厂家都支持。

6. 编程基础

XS Studio 也有与其他高级编程语言相似的常量、变量概念。常量具有和变量一样的概念和性质，不同的是，常量不可修改，是固定的，而变量可以多次修改。变量的存储位置可由用户指定为%I 区、%Q 区、%M 区的特定地址，或不指定地址，由系统自行分配，用户可以不关注这些变量的存储位置。

6-1. 直接地址

6-1-1. 定义语法

在 XS Studio 应用中，当需要和可编程逻辑控制器的 I/O 模块进行变量映射或和外部设备进行网络通讯时，需要采用此声明方法。

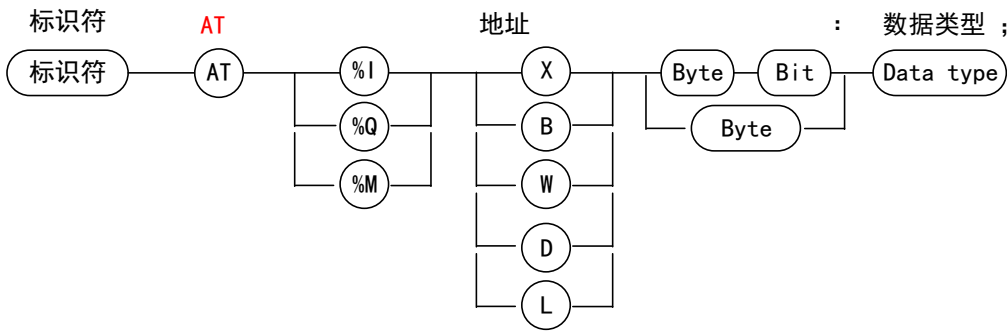
使用关键字 AT 把变量直接联结到确定地址，直接变量须符合以下规则：

AT<地址>：

<标识符>AT<地址>：<数据类型>{：=<初始化值>}；

{ } 中为可选部分。

使用“%”开始，随后是位置前缀符号和大小前缀符号，如果有分级，则用整数表示分级，并用小数点符号“.”表示，如%IX0.0，%QW0。直接变量声明的具体格式如下图所示：



位置前缀的定义：

- ◆ I：表示输入单元；
- ◆ Q：表示输出单元；
- ◆ M：表示存储区单元。

大小前缀的定义如下表所示：

前缀符号	定义	约定数据类型
X	位（bit）	BOOL
B	字节（byte）	BYTE
W	字（WORD）	WORD
D	双字（DWORD）	DWORD
L	长字（LWORD）	LWORD
*	未特定位置的内部变量，系统自动分配	

此区域可根据实际硬件资源调整大小。

 示例:

%IX3.2 输入区域偏移 3 个字节的第三位 (bit2)
%QW10 输出区域偏移 10 个字
%MB20 内存区域偏移 20 个字节
Var1 AT%ID48:DWORD; //Var1 变量是双字类型，映射到输入区域偏移 48 双字位置

6-1-2. PLC 直接地址存储区域

区域	用途	大小	地址范围
I 区 (%I) 128KB	用户使用区域	64KWords	%IW0-%IW65535
Q 区 (%Q) 128KB	用户使用区域	64KWords	%QW0-%QW65535
M 区 (%M) 256KB	用户使用区域	128KWords	%MW0-%MW131070

6-2. 变量

6-2-1. 概述

变量可以在 POU 的定义部分或者通过自动声明对话框定义,也可以在全局变量列表编辑器定义,通过变量类型关键字来标识变量类型,例如通过 VAR 和 END_VAR 来标识之间定义的变量为本地变量。

变量类型包括本地变量 (VAR), 输入变量 (VAR_INPUT), 输出变量 (VAR_OUTPUT), 输入输出变量 (VAR_IN_OUT), 全局变量 (VAR_GLOBAL), 临时变量 (VAR_TEMP), 静态变量 (VAR_STAT), 常量 (VAR_CONSTANT), 保持变量 (VAR_RETAIN), 持久变量 (VAR_PERSISTENT)。

6-2-2. 变量定义

1、变量定义

1) 文本声明

```
1 {attribute 'qualified only'}
2 VAR_GLOBAL PERSISTENT RETAIN
3   M AT %MB0: BOOL:=TRUE;
4   D AT %MW40000:ARRAY [0..5] OF WORD;
5   HD AT %MW40006:ARRAY [0..1] OF LREAL;
6 END_VAR
```

2) 表格声明

	类别	名称	地址	数据类型	初值	注释	特性
1	VAR_GLOBAL RETAIN PERSISTENT	M	%MB0	BOOL	TRUE		
2	VAR_GLOBAL RETAIN PERSISTENT	D	%MW40000	ARRAY[0..5] OF WORD			
3	VAR_GLOBAL RETAIN PERSISTENT	HD	%MW40006	ARRAY[0..5] OF LREAL			

表格声明中,可以对变量的各类属性进行编辑设置。下表是对表格声明的具体说明:

类别	变量的类别 (如本地变量 (VAR)、输入变量 (VAR_INPUT)、输出变量 (VAR_OUTPUT) 等)
名称	变量的名称
地址	变量映射的地址
数据类型	变量的数据类型 (如 BOOL, INT 等)
初值	变量的初始值
注释	变量的注释
特性	变量的特性

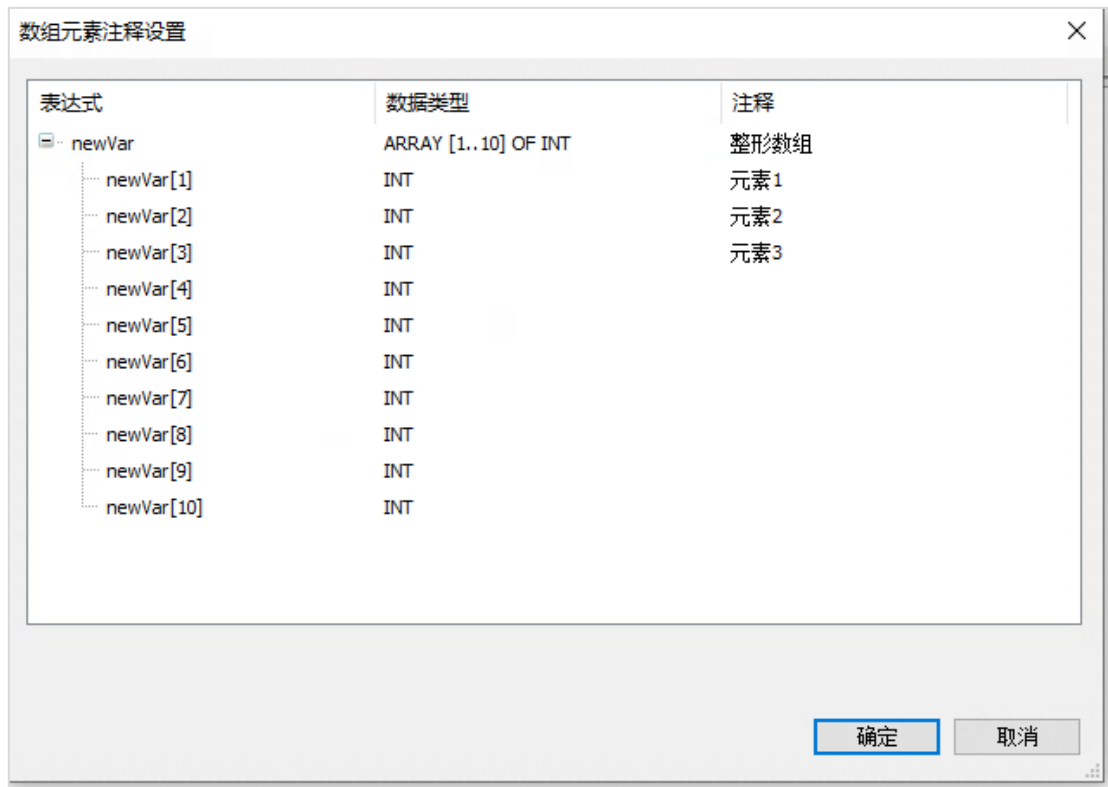
2、变量定义支持数组元素注释和实例注释

1) 数组元素注释

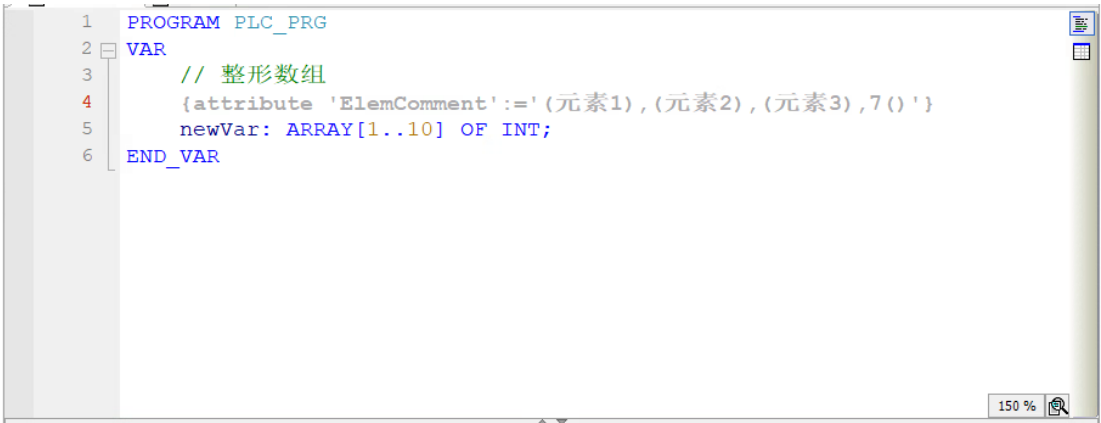
表格声明注释设置界面如下所示：



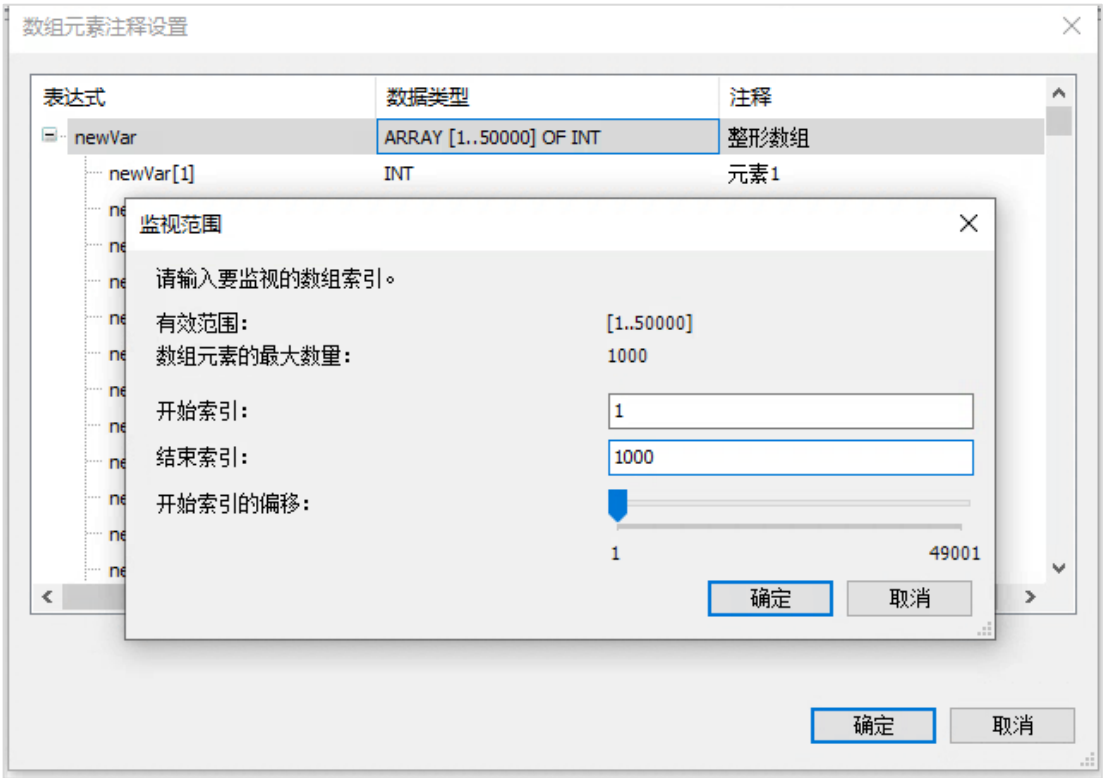
双击注释空白处后点击



设置完成后文本声明效果如图所示（同样也可以用文本直接声明）：



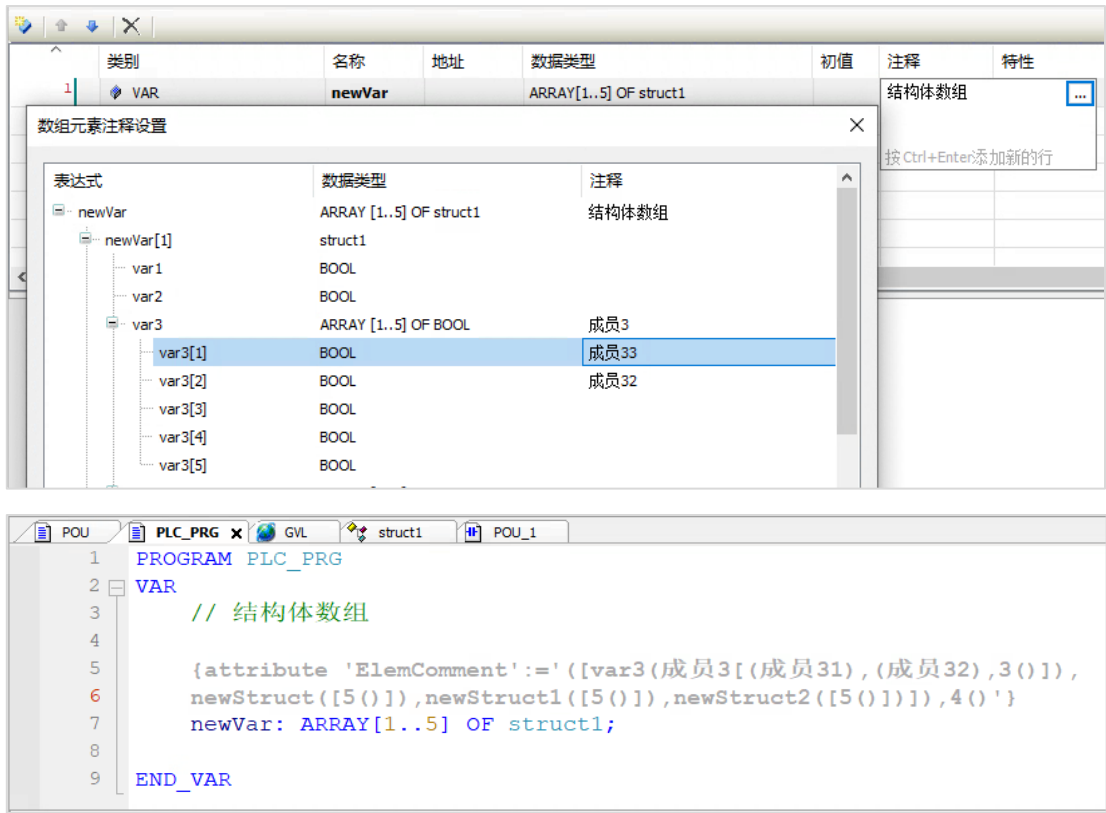
- 可以通过表格和文本进行数组元素注释编辑。
- 表格在注释列，弹框（同初始值操作）显示当前元素及子元素注释编辑界面。
- 文本编辑器编辑格式如下：
 - ◆ 数组本身：使用标准的注释编辑方式。
 - ◆ 数组元素：{attribute 'ElemComment' := ' 1(子元素 1 注释),1(子元素 2 注释),n(相同子元素注释)' }。
- 如果是表格模式，声明数组类型变量时，默认注释为空（默认增加特性格式）。
- 表格模式的数组元素注释，只显示数组本身注释，不显示元素注释。
- 特性列中，去掉元素注释特性显示(数组元素注释是用特性来实现的，特性是标记在变量上的信息)。
- 表格视图数组长度变更时，存量数组元素的注释也会跟随保存。
- 表格视图中数组维度变更时，数组元素注释按扩展维度的最小下标对注释进行平移保存。
 - ◆ 如数组 INT_ARRAY:ARRAY[1..2,2..3]维度变更为 ARRAY[1..2,2..3,3..4],原来的数组元素 INT_ARRAY[1,2]注释会迁移到新数组元素 INT_ARRAY[1,2,3]中。
 - ◆ 如 INT_ARRAY:ARRAY[1..2,2..3] 维度变更为 ARRAY[1..2]，原来的数组元素 INT_ARRAY[1,2]注释会迁移到新数组元素 INT_ARRAY[1]中。
- 表格视图中，数据类型由数组类型变更为非数组类型时，会清空数组元素注释。
- 数组元素注释编辑界面最大显示 1000 个元素，双击数组所在的行中‘数据类型’列，可调节编辑显示范围。



2) 实例注释

在 PRG（程序）和 GVL（全局变量表）中声明的变量或申明为 VAR_STAT（静态）类型的变量，可以不受限制地展开变量内部成员编辑注释，保存注释时所有内部成员的注释会标记在该变量上，这种注释称为变量的实例注释。

如下图所示，数据结构体内的成员注释都可以标记并保存在变量数组结构体上。

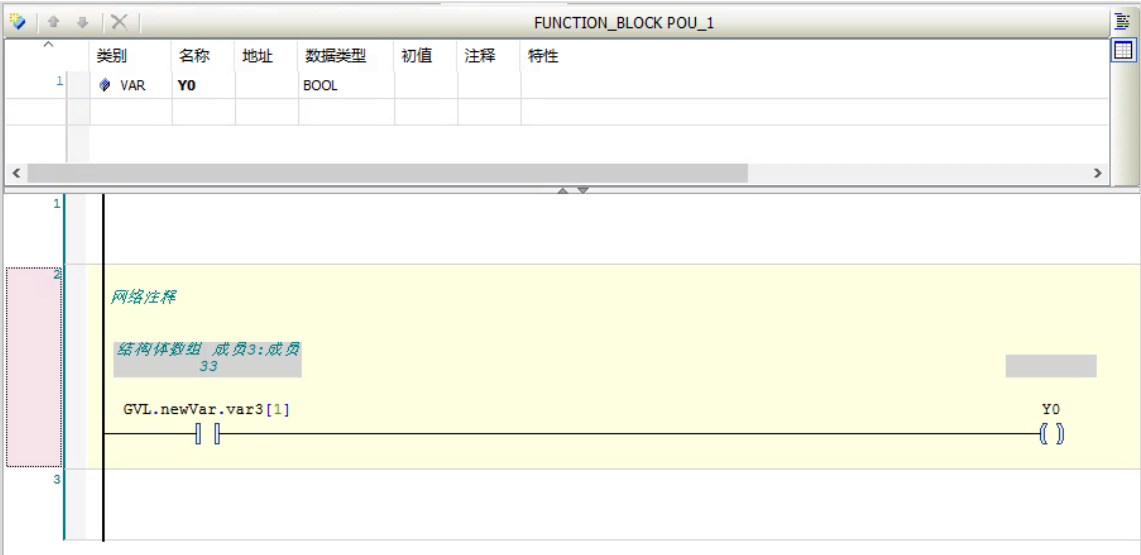


- ◆ 内部成员为 FB 类型时，只会显示输入、输出、输入输出变量，其他类型的变量不会展示。
- ◆ 表格中，数据类型由非数组类型变更为数组类型时，会清空实例注释。
- ◆ 变量的数组类型成员最大显示 1000 个元素，可调节编辑显示范围。

3) 注释的显示

在初始值编辑界面，监控变量表界面，梯形图，鼠标悬停显示注释等涉及到变量注释显示的功能，注释显示以实例注释优先，如果变量没有实例注释，则显示变量的类型注释。

梯形图中涉及到数组元素的注释显示的，以数组注释元素注释合并显示；但同样采用上面的优先级顺序规则进行显示。如下图所示：



实例注释：



3、变量的初始值设置

1) 初始值设置

在变量声明的初始值一列中点击 打开设置窗口。



2) 也可以用文本形式直接声明

```
1 {attribute 'qualified only'}
2 VAR_GLOBAL PERSISTENT RETAIN
3 // 开关
4 M AT %MB0: BOOL:=TRUE;
5 // 长度
6 D:ARRAY [0..2] OF WORD := [1, 2, 3];
7 // 距离
8 HD AT %MW40006:LREAL;
9 END_VAR
```

4、变量定义的子元素显示地址信息

在变量声明的地址一列中点击，打开变量地址设置窗口。
地址显示界面如下：

PROGRAM PLC_PRG

	类别	名称	地址	数据类型	初值	注释	特性
1	VAR	newVar1	%MB2000	ARRAY[0..2] OF ARRAY [1..3] OF BYTE			

变量地址

表达式	地址	数据类型
PLC_PRG.newVar1	%MB2000	ARRAY [0..2] OF ARRAY [1..3] OF BYTE
PLC_PRG.newVar1[0]	%MB2000	ARRAY [1..3] OF BYTE
PLC_PRG.newVar1[0][1]	%MB2000	BYTE
PLC_PRG.newVar1[0][2]	%MB2001	BYTE
PLC_PRG.newVar1[0][3]	%MB2002	BYTE
PLC_PRG.newVar1[1]	%MB2003	ARRAY [1..3] OF BYTE
PLC_PRG.newVar1[1][1]	%MB2003	BYTE
PLC_PRG.newVar1[1][2]	%MB2004	BYTE
PLC_PRG.newVar1[1][3]	%MB2005	BYTE
PLC_PRG.newVar1[2]	%MB2006	ARRAY [1..3] OF BYTE



- 地址列文本框只读；
- 最大显示元素个数，数组中最大显示 1000 个元素，可调节显示范围。

6-2-3. 变量类型

变量类型包括本地变量（VAR），输入变量（VAR_INPUT），输出变量（VAR_OUTPUT），输入输出变量（VAR_IN_OUT），全局变量（VAR_GLOBAL），临时变量（VAR_TEMP），静态变量（VAR_STAT），配置变量（VAR_CONFIG）。

变量类型声明语法：<TYPE> | Attribute

variable1;

variable2;

...

END_VAR

TYPE：类型关键字，包括 VAR（本地变量），VAR_INPUT（输入变量），VAR_OUTPUT（输出变量），VAR_IN_OUT（输入输出变量），VAR_GLOBAL（全局变量），VAR_TEMP（临时变量），VAR_STAT（静态变量），VAR_CONFIG（配置变量）。

Attribute：属性关键字，包括 RETAIN，PERSISTENT，CONSTANT，用于明确变量的范围。

1、变量类型

变量类型关键字	变量属性	外部读写	内部读写
VAR	局部变量	-	R/W
VAR_INPUT	输入变量，由外部提供	R/W	R
VAR_OUTPUT	输出变量，有内部变量提供给外部	W	R/W
VAR_IN_OUT	输入-输出变量	R/W	R/W
VAR_GLOBAL	全局变量，能在所有配置、资源内使用	R/W	R/W
VAR_TEMP	临时变量，程序和功能块内部存储使用的变量	-	R
VAR_STAT	静态变量		
VAR_EXTERNAL	外部变量，能在程序内部修改，但需由全局变量提供	R/W	R/W

VAR、VAR_INPUT、VAR_OUTPUT 和 VAR_IN_OUT 是在程序组织单元（POU）中被应用的最多的几种变量类型。

VAR_GLOBAL 全局变量在实际的工程项目中也需要被大量的使用。

2、变量属性

变量附加属性关键字	变量附加属性
RETAIN	保持型变量，用于掉电保持
PERSISTENT	保持型变量
VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN	两者功能一样，皆为保持性变量，用于掉电保持
CONSTANT	常量

● RETAIN

以关键字 RETAIN 声明类型变量。RETAIN 型变量在控制器正常关闭、打开（或收到在线命令“热复位”），甚至意外关闭之后这类变量仍然能保持原来的值。随着程序重新开始运行，存储的值能继续发挥作用。

RETAIN 类型变量声明格式如下：

VAR RETAIN

<标识符>:<数据类型>;

END_VAR

但 RETAIN 变量在“初始值位”、“冷复位”和程序下载之后将会重新初始化。

内存存储位置：RETAIN 型变量仅仅被存储在一个单独的内存区中。

在实际的工程应用中，如生产线上的计件器便是一个典型的例子：电源被切断之后，它仍然可以在再次启动时继续计数。而其它所有变量此时都将被重新初始化，变为指定初始值或标准初始化的值。

● PERSISTENT

目前只有少数 PLC 还保留独立的内存区域用于存放 PERSISTENT 类型数据，在 XS Studio 中，取消了其原掉电保持的功能，取而代之的是通过 VAR RETAIN PERSISTENT 或 VAR PERSISTENT RETAIN 来实现，两者从功能上完全一样。

PERSISTENT 类型变量声明格式如下：

VAR GLOBAL PERSISTENT RETAIN

<标识符>:<数据类型>;

END_VAR

内存存储位置：与 RETAIN 变量一样，RETAIN PERSISTENT 和 PERSISTENT RETAIN 变量也存储在一个独立的内存区中。

● CONSTANT

常量，在程序运行过程中，只能对其读取数据而不能进行修改的量称之为常量，关键字为 CONSTANT。可以将常量声明为局部常量，也可以为全局常量。

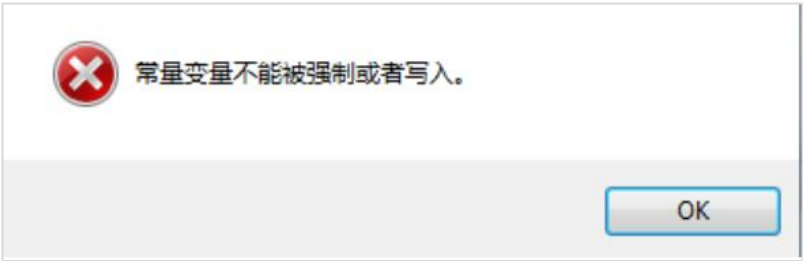
CONSTANT 常量声明格式如下：

```
VAR_GLOBAL CONSTANT
<标识符>:<数据类型> := <初始化值>;
END_VAR
```

在实际应用中，通常可以将一些重要参数或系数设为常量，这样可以有效的避免其他变量对其修改最终影响系统整体稳定性及安全性。举例如下。

```
VAR CONSTANT
pi:REAL:= 3.1415926;
END_VAR
```

程序一旦开始运行，通过 CONSTANT 声明的变量在程序运行过程中，是不允许被修改的，如强制修改系统会出现如下所示的系统错误：



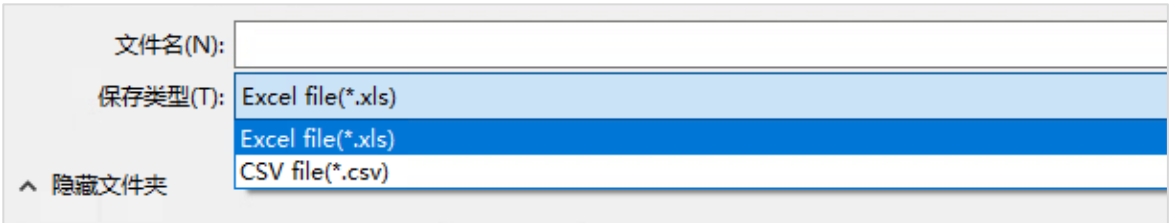
6-2-4. 变量导入与导出

支持变量导入与导出，导出文件类型为“XLS 工作表.xls)”，呈现为 Excel 表格形式，可在外部进行增加、删减或其他对变量的编辑后再导入到 XS Studio 编程软件。

如下图所示：

	类别	名称	地址	数据类型	初值	注释	特性
1	VAR_GLOBAL	newVarGlobal		BOOL			
2	VAR_GLOBAL CONSTANT	newVarGlobal1		INT			
3	VAR_GLOBAL CONSTANT RETAIN	newVarGlobal2		byte			
4	VAR_GLOBAL PERSISTENT	newVarGlobal3		WORD	100		
5	VAR_GLOBAL RETAIN PERSISTENT	newVarGlobal4		DWORD		TEST	
6	VAR_GLOBAL	newVarGlobal5		UINT			

在变量表中增加部分变量，并右键选择导出变量表类型 Excel/CSV（CSV 是纯文本文件，excel 包含有格式信息。CSV 文件较小，创建分发读取较方便，适合存放结构化信息。CSV 文件在 windows 平台默认的打开方式是 excel，本质是文本文件），两种格式文件对变量的编辑没有区别。



打开导出的文件，并编辑（添加新变量 VAR1、VAR2、VAR3、VAR4）后导入，效果如下图所示：

Type	Name	Address	DataType	InitValue	Comment	Attribute
VAR_GLOBAL	newVarGlobal		BOOL			
VAR_GLOBAL CONSTANT	newVarGlobal1		INT			
VAR_GLOBAL CONSTANT RETAIN	newVarGlobal2		BYTE			
VAR_GLOBAL PERSISTENT	newVarGlobal3		WORD	100		
VAR_GLOBAL RETAIN PERSISTENT	newVarGlobal4		DWORD		TEST	
VAR_GLOBAL	newVarGlobal5		UINT			
VAR_GLOBAL	VAR1		INT	10		
VAR_GLOBAL	VAR2		BYTE		导入	
VAR_GLOBAL	VAR3		WORD			
VAR_GLOBAL	VAR4		REAL			

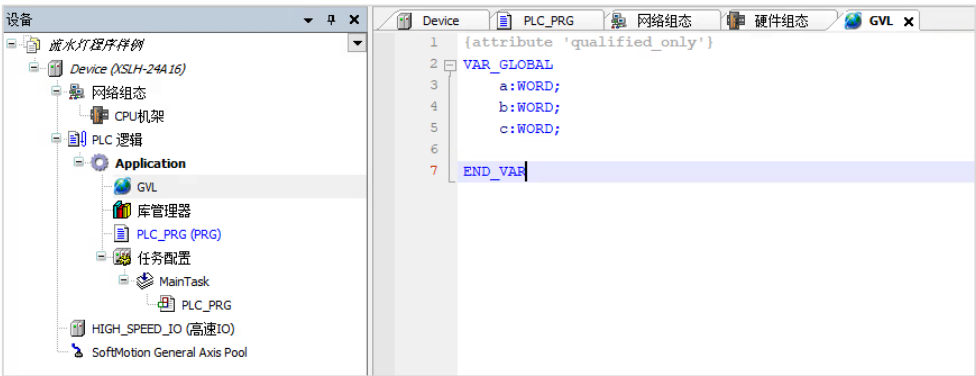
类别	名称	地址	数据类型	初值	注释	特性
1 VAR_GLOBAL	newVarGlobal		BOOL			
2 VAR_GLOBAL CONSTANT	newVarGlobal1		INT			
3 VAR_GLOBAL CONSTANT RETAIN	newVarGlobal2		BYTE			
4 VAR_GLOBAL PERSISTENT	newVarGlobal3		WORD	100		
5 VAR_GLOBAL RETAIN PERSISTENT	newVarGlobal4		DWORD		TEST	
6 VAR_GLOBAL	newVarGlobal5		UINT			
7 VAR_GLOBAL	VAR1		INT	10		
8 VAR_GLOBAL	VAR2		BYTE		导入	
9 VAR_GLOBAL	VAR3		WORD			
10 VAR_GLOBAL	VAR4		REAL			

6-3. 配方操作

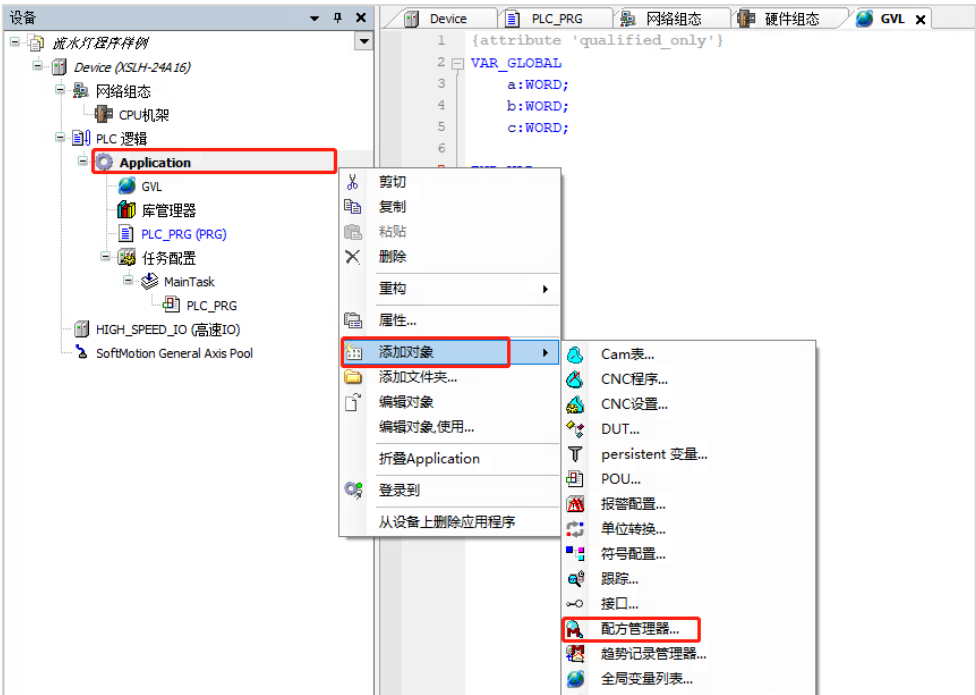
配方管理器的功能是提供用于维护用户定义的变量列表（配方定义）。用户通过配方管理器可以对存储位置、存储方式、存储类别进行配置，如下图所示。配方管理器配置成功后，用户即可实现对配方定义的上传与下载。

6-3-1. 应用举例

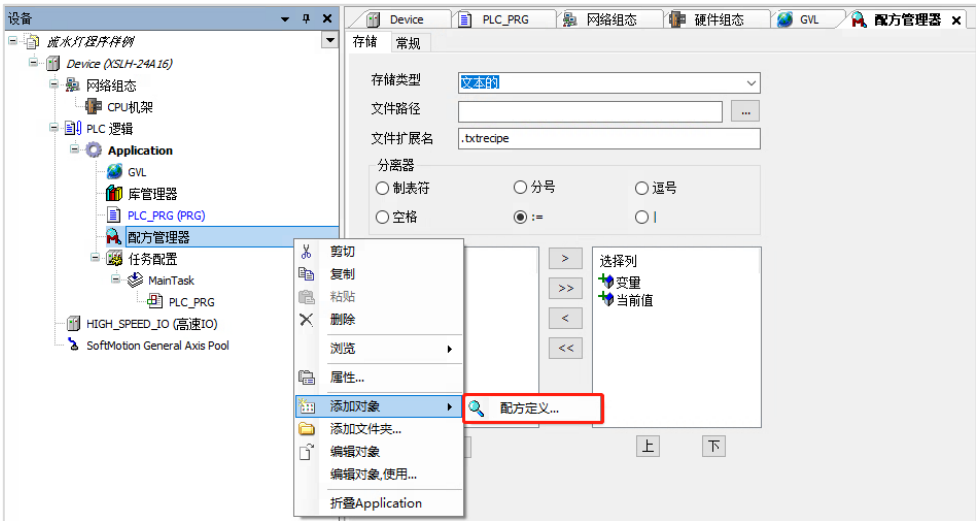
1、 创建变量：



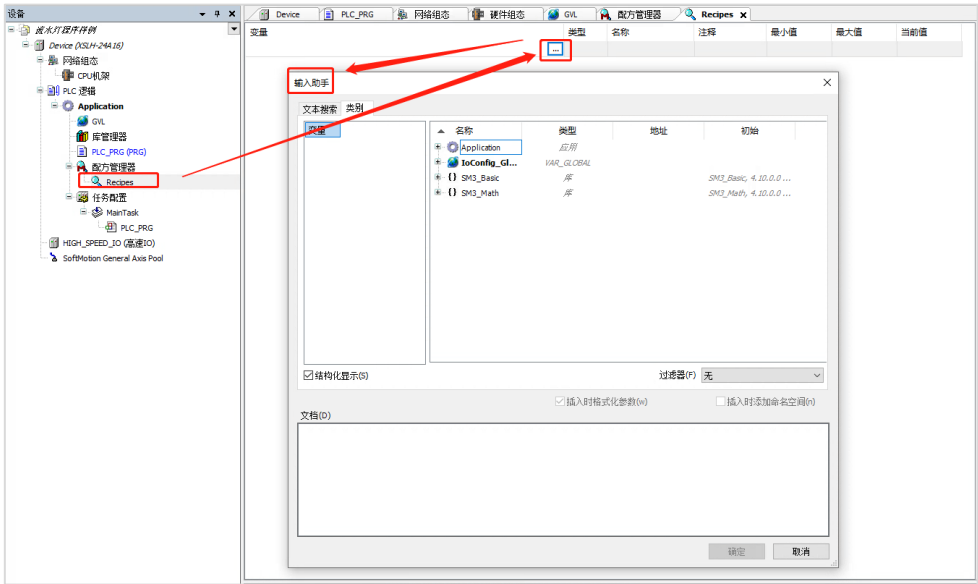
2、 建立配方管理器：



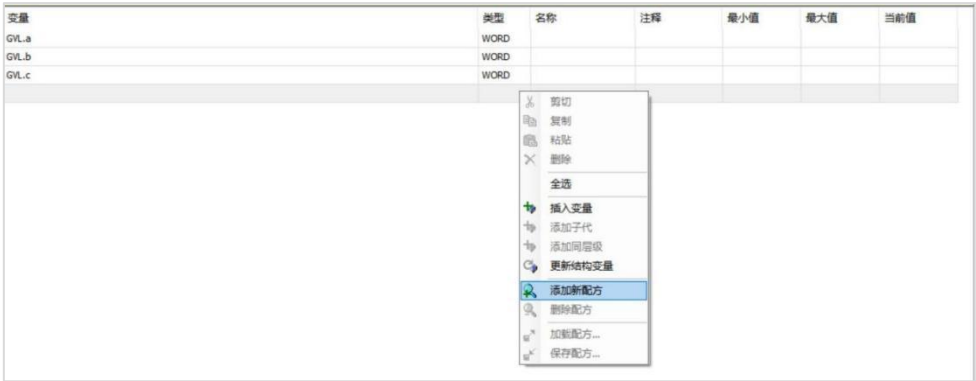
3、 创建配方定义：

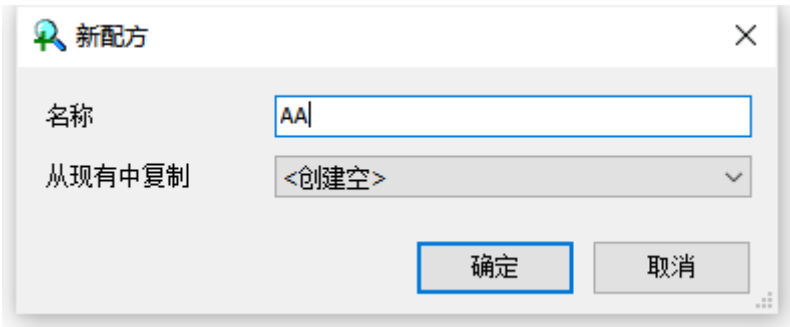


4、 选择需要使用配方功能的变量：

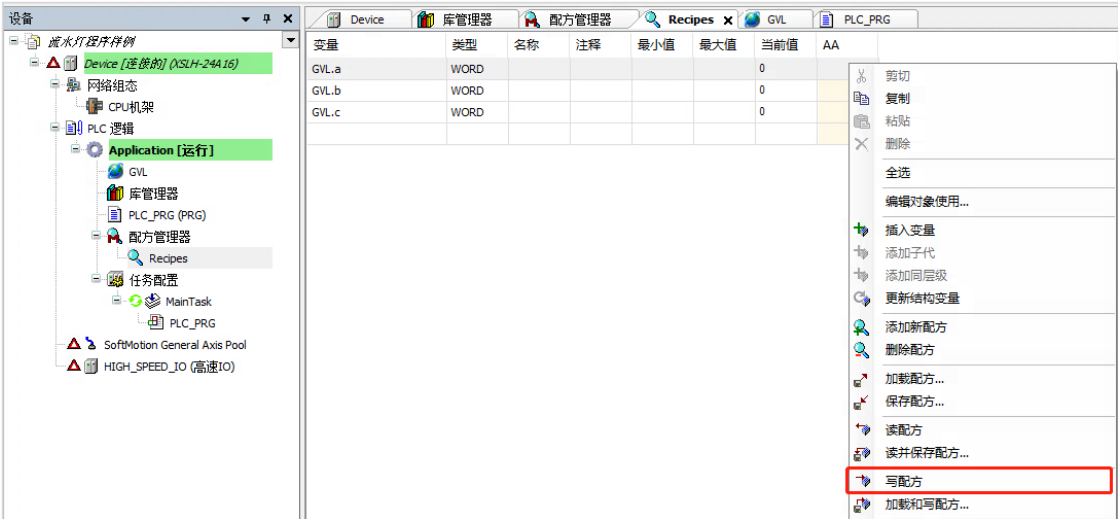


5、 添加配方：





6、 登录设备在对应配方最下方表格右击可对配方进行写入读取等操作；



7、 也可使用指令进行配方创建、读取、写入等操作。

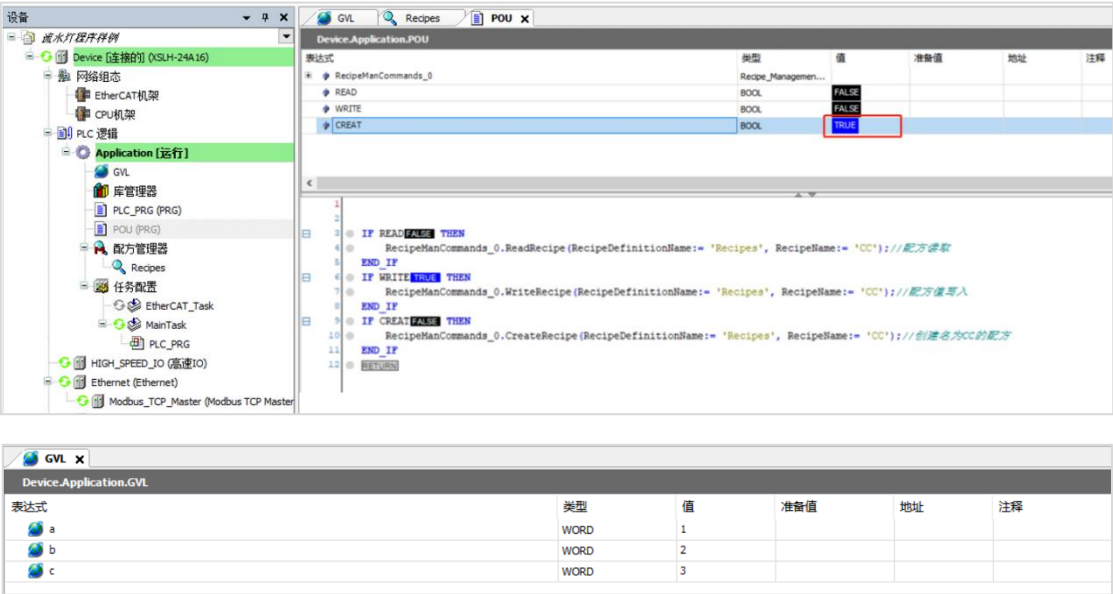
① 编写配方创建、读取写入的程序；

 示例：

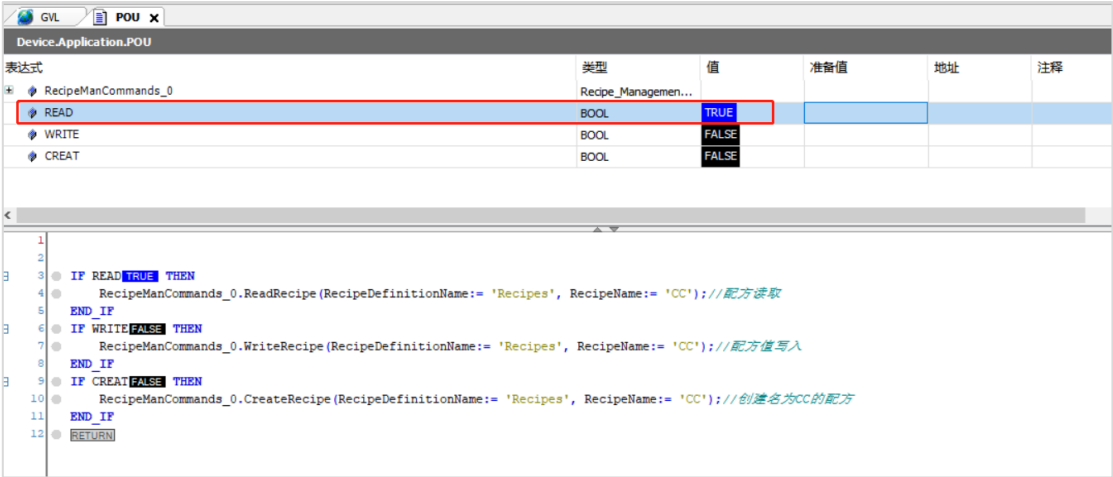
```
VAR
    RecipeManCommands_0:Recipe_Management.RecipeManCommands;
    READ:BOOL;
    WRITE:BOOL;
    CREAT:BOOL;
END_VAR

IF READ THEN
    RecipeManCommands_0.ReadRecipe(RecipeDefinitionName:= 'Recipes', RecipeName:= 'CC');
    //配方读取
END_IF
IF WRITE THEN
    RecipeManCommands_0.WriteRecipe(RecipeDefinitionName:= 'Recipes', RecipeName:= 'CC');
    //配方值写入
END_IF
IF CREAT THEN
    RecipeManCommands_0.CreateRecipe(RecipeDefinitionName:= 'Recipes', RecipeName:= 'CC');
    //创建名为 CC 的配方
END_IF
```

登录创建配方；



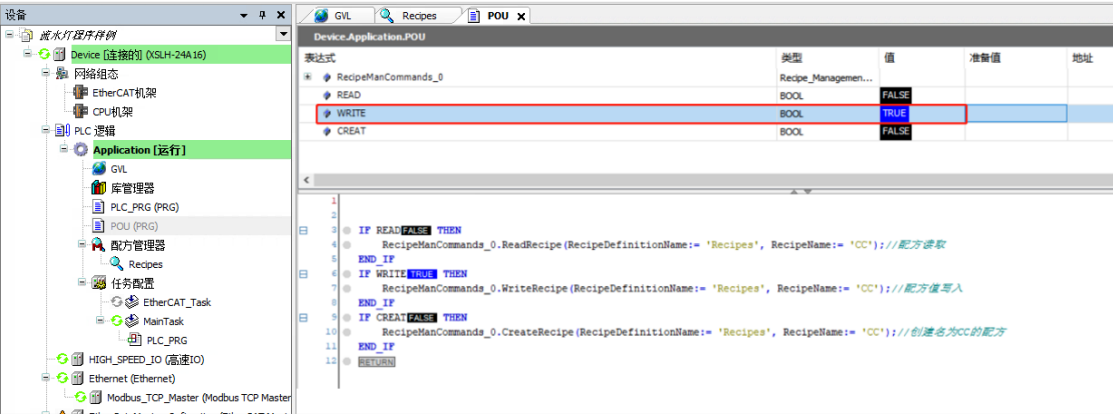
② 读取值到配方；



③ 将当前值修改为其他值；



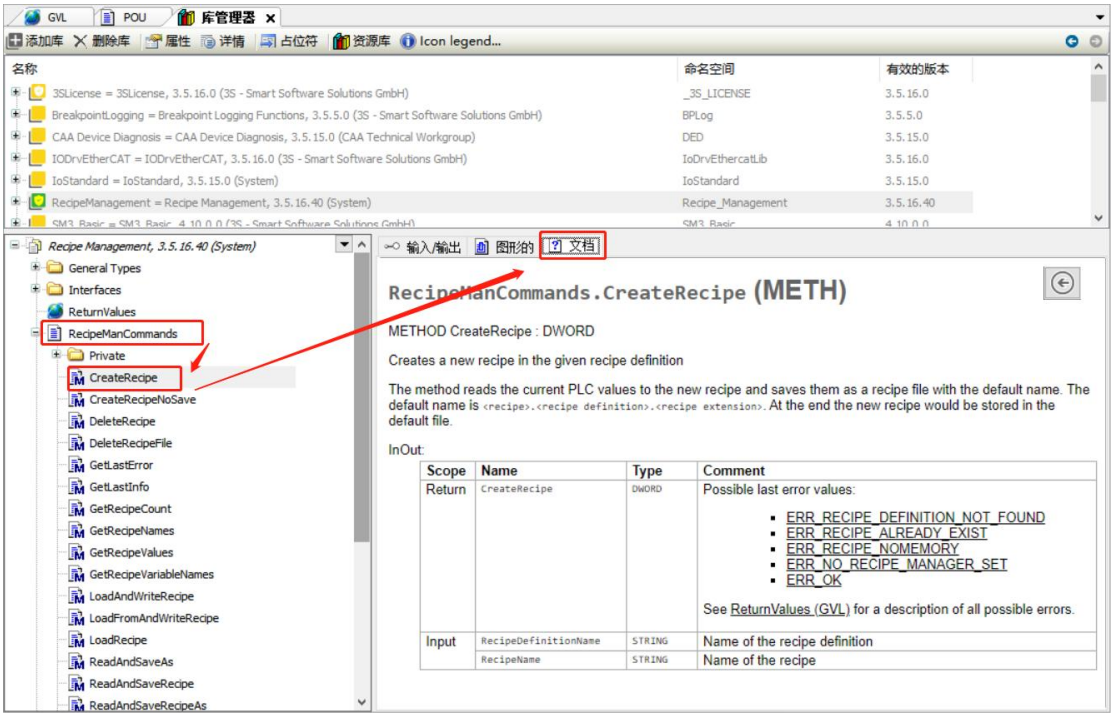
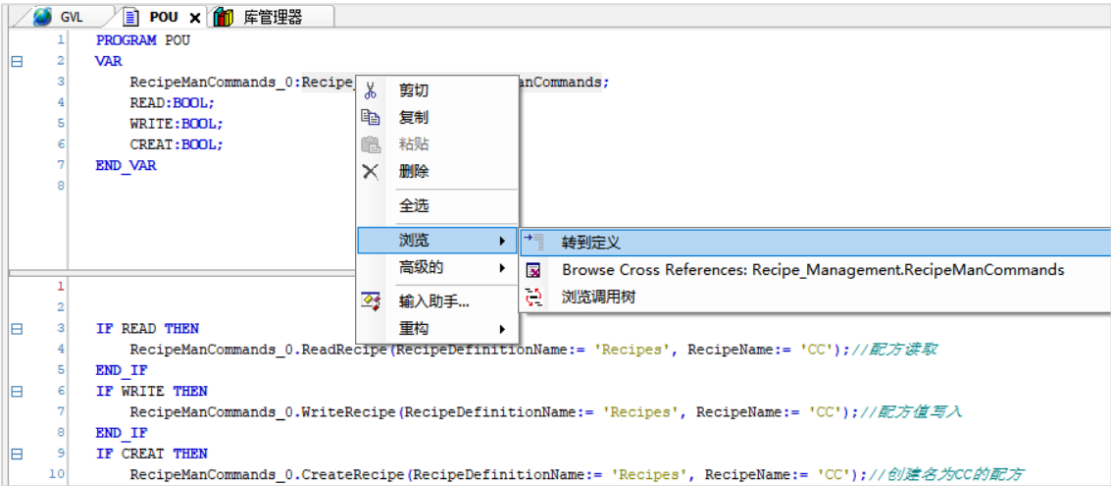
④ 写入配方值。



Device.Application.GVL				
表达式	类型	值	准备值	地址
a	WORD	1		
b	WORD	2		
c	WORD	3		



配方功能其他功能可以在定义中查找使用。如下方图片所示。



7. 编程语言

7-1. XS Studio 支持的编程语言

XS Studio 编程软件支持的 PLC 编程语言：

- ◆ Ladder diagram(LD) 梯形图
- ◆ Function block diagram(FBD) 功能块图
- ◆ Structured text (ST) 结构化文本
- ◆ Sequential function chart (SFC) 顺序功能图
- ◆ Continuous function chart (CFC) 连续功能图

以上所有语言，在编辑器界面均支持标准的 Ctrl、Shift 编辑器功能快捷键。如复制（Ctrl+C）、粘贴（Ctrl+V）和“撤销”（Ctrl+Z）等快捷键；同时支持快捷键<F2>启动输入助手，系统会根据当前编程环境提供相对应的输入提示或选择。

7-2. 结构化文本语言（ST）

7-2-1. 简介

结构化文本（ST）是一种高级的文本语言，可以用来描述功能，功能块和程序的行为，还可以在顺序功能流程图中描述步、动作和转变的行为。结构化文本编程语言是一种高级语言，类似于 Pascal，是一种特别为工业控制应用而开发的一种语言，也是在 XS Studio 中最常用的一种语言，对于熟悉计算机高级语言开发的人员来说，结构化文本语言更是易学易用，它可以实现选择、迭代、跳转语句等功能。此外，结构化文本语言还易读易理解，特别是当用有实际意义的标识符、批注来注释时，更是这样。在复杂控制系统中，结构化文本可以大大减少其代码量，使复杂系统问题变得简单，缺点是调试不直观，编译速度相对较慢。

例如：

```
FOR a:=0 TO 0 BY 1 DO
  D_温度显示值[a] := TO_REAL(D_温度实际值[a]) / 10;
  D_温度最终值[a] := D_温度显示值[a] + D_温度补偿值[a];
END_FOR
IF M_自整定开关 THEN
  M_温控模式[0] := 1;
END_IF
```

7-2-2. ST 程序执行顺序

1、程序执行顺序

使用结构化文本的程序执行顺序根据“行号”依次从上至下开始顺序执行，如下图所示：

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

SMC3_ETC_WriteParameter_CoE_0(
 xExecute:= WRITE, //上升沿触发
 xAbort:= ,
 uiIndex:=16#607D, //对象索引例如16#6060
 usiSubIndex:=1, //对象的子索引, 例如0
 usiDataLength:=4, //写入数据的长度, 以字节为单位(1 ~ 4)
 dwValue:= 16#8000, //写入值 DWORD
 Axis:=SM_Drive_GenericDSP402, //"SoftMotion轴。
 xDone=> ,
 xBusy=> ,
 xError=> ,
 dwErrorCode=> ,
 eError=>);
SMC3_ETC_WriteParameter_CoE_1(
 xExecute:= WRITE,
 xAbort:= ,
 uiIndex:=16#607D ,
 usiSubIndex:=2 ,
 usiDataLength:=4 ,
 dwValue:= 16#8000,
 Axis:=SM_Drive_GenericDSP402 ,
 xDone=> ,
 xBusy=> ,
 xError=> ,
 dwErrorCode=> ,
 eError=>);

2、表达式执行顺序

表达式中包括操作符和操作数，操作数按照操作符指定的规则进行运算，得到结果并返回。操作数可以为变量、常量、寄存器地址、函数等。

a+b+c;
3.14*R*R;
ABS(-10)+var1;

如果在表达式中有若干个操作符，则操作符会按照约定的优先级顺序执行：先执行优先级高的操作符运算，再顺序执行优先级低的操作符运算。如果在表达式中具有优先级相同的操作符，则这些操作符按照书写顺序从左至右执行。操作符的优先级如下表所示：

操作符	符号	优先级
小括号	()	最高
函数调用	Function name (Parameter list)	
求幂	EXPT	
取反	NOT	
乘法	*	
除法	/	
取模	MOD	
加法	+	
减法	-	
比较	<,>,<=,>=	
等于	=	
不等于	<>	
逻辑与	AND	
逻辑异或	XOR	
逻辑或	OR	最低

7-2-3. 语句

结构化文本语句如下表所示：

指令类型	指令语句	举例
赋值语句	:=	bFan:= TRUE;
功能块/函数调用	功能块/函数调名();	
选择语句	IF	IF <布尔表达式> THEN <语句内容>; END IF
	CASE	
迭代语句	FOR	
	WHILE	
	REPEAT	
跳转语句	EXIT	
	CONTINUE	
	JMP	
返回语句	RETURN	
NULL 语句	;	

1、赋值语句

是结构化文本中最常用的语句之一，作用是将其右侧的表达式产生的值赋给左侧的操作数（变量或地址），使用“:=”表示。

< 变量>:=< 表达式>;

 示例:

分别给两个布尔型变量赋值，bFan 置 TRUE，bHeater 置 FALSE。

```
VAR
bFan: BOOL;
bHeater:BOOL;
END_VAR

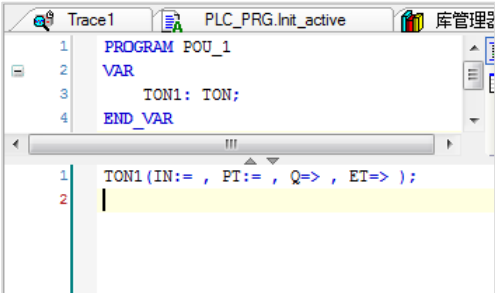
bFan:= TRUE;
bHeater:= TRUE;
```

2、函数及功能块调用

功能块调用采用将功能块名进行实例化实现调用，如 Timer 为 TON 功能块的实例名，具体格式如下：

功能块实例名:(功能块参数);

如果需要在 ST 中调用功能块，可直接输入功能块的实例名称，并在随后的括号中给功能块的各项参数分配数值或变量，参数之间以逗号隔开；功能块调用以分号结束。
例如，在结构化文本中调用功能块 TON 定时器，假设其实例名为 TON1，具体实现如图所示：



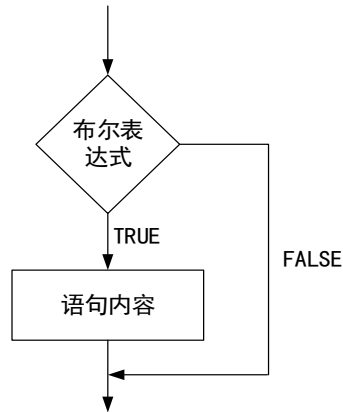
3、选择语句

1) IF语句

用 IF 语句实现单分支选择结构，基本格式如下：

```
IF < 布尔表达式> THEN
< 语句内容>;
END_IF
```

如果使用上述格式，只有当<布尔表达式>为 TRUE 时，才执行语句内容，否则不执行 IF 语句的<语句内容>。语句内容可以为一条语句或者可以为空语句，也可以并列多条语句，该语句表达式执行流程图如图所示：

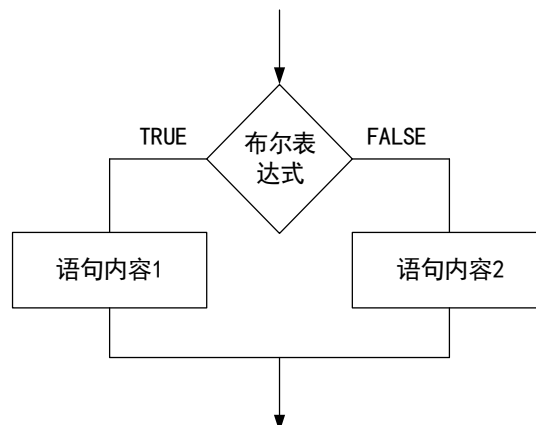


2) IF...ELSE语句

用 IF 语句实现双分支选择机构，基本格式如下：

```
IF <布尔表达式> THEN
<语句内容 1>;
ELSE
<语句内容 2>;
END_IF
```

如上表达式先判断<布尔表达式>内的值，如果为 TRUE，则执行<语句内容 1>，如为 FALSE，则执行<语句内容 2>，程序执行流程图如图所示：



当程序的条件判断式不止一个时，此时，需要再一个嵌套的 IF...ELSE 语句，即多分支选择结构，基本格式如下：

```
IF <布尔表达式 1> THEN
IF <布尔表达式 2> THEN
<语句内容 1>;
ELSE
```

```

<语句内容 2>;
END_IF
ELSE
<语句内容 3>;
END_IF

```

如上，在 IF...ELSE 中有放入了一个 IF...ELSE 语句，实现嵌套，下面通过一个例子说明嵌套的使用。如上表达式先判断<布尔表达式 1>内的值，如果为 TRUE，则继续判断<布尔表达式 2>的值，如果<布尔表达式 1>的值为 FALSE，则执行<语句内容 3>，回到<布尔表达式 2>判断，如果<布尔表达式 2>为 TRUE，则执行<语句内容 1>，反之则执行<语句内容 2>。

3) IF..ELSIF..ELSE 语句

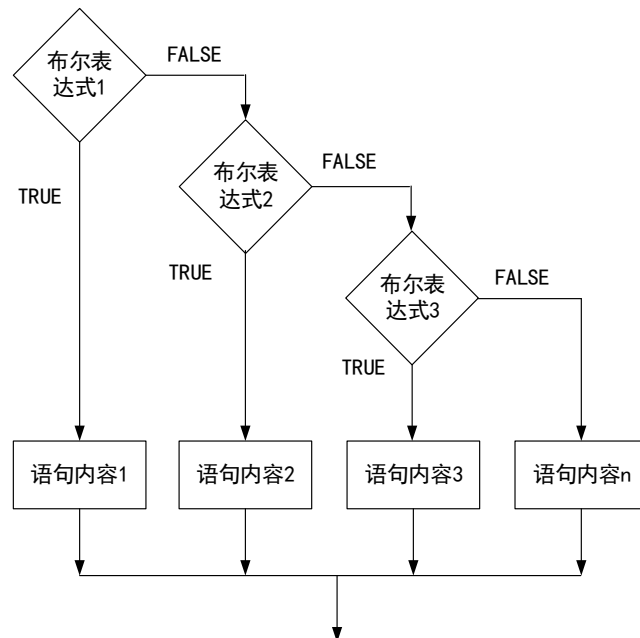
此外，多分支选择结构还能通过如下方式来呈现。具体格式如下：

```

IF < 布尔表达式 1> THEN
< 语句内容 1>;
ELSIF < 布尔表达式 2> THEN
< 语句内容 2>;
ELSIF < 布尔表达式 3> THEN
< 语句内容 3>;
...
...
ELSE
< 语句内容 n>;
END_IF

```

如果表达式<布尔表达式 1>为 TRUE，那么只执行指令<语句内容 1>，不执行其它指令。否则，从表达式<布尔表达式 2>开始进行判断，直到其中的一个布尔表达式为 TRUE，然后执行与此布尔表达式对应语句内容。如果布尔表达式的值都不为 TRUE，则只执行指令<语句内容 n>，程序执行流程图如图所示：



4) CASE语句

CASE 语句是多分支选择语句，他根据表达式的值来使程序从多个分支中选择一个用于执行的分支，基本格式如下：

```

CASE < 条件 变量> OF
< 数值 1>: < 语句内容 1>;
< 数值 2>: < 语句内容 2>;
< 数值 3, 数值 4, 数值 5>: < 语句内容 3>;

```

```

< 数值 6.. 数值 10>: < 语句内容 4>;
...
< 数值 n>: < 语句内容 n>;
ELSE
<ELSE 语句内容>;
END_CASE;

```

CASE 语句按照下面的模式进行执行：

- 如果<条件变量>的值为<数值 i>，则执行指令<语句内容 i>。
- 如果<条件变量>没有任何指定的值，则执行指令< ELSE 语句内容>。
- 如果条件变量的几个值都需要执行相同的指令，那么可以把几个值相继写在一起，并且用逗号分开。这样，共同的指令被执行，如上程序第四行。
- 如果需要条件变量在一定的范围内执行相同的指令，可以通过写入初、终值，以两个点分开。这样，共同的指令被执行，如上程序第五行。

4、迭代语句

迭代语句主要用于重复执行的程序，在 XS Studio 中，常见的迭代语句有 FOR, REPEAT 及 WHILE 语句，下面对这种语句做详细解释：

1) FOR 循环

FOR 循环语句用于计算一个初始化序列，当某个条件为 TRUE 时，重复执行嵌套语句并计算一个迭代表达式序列，如果为 FALSE，则终止循环，具体格式如下：

```

FOR < 变量> := < 初始值> TO < 目标值> {BY < 步长>} DO
< 语句内容>
END_FOR;

```

FOR 循环的执行顺序如下：

- 计算<变量>是否在<初始值>与<目标值>的范围内。
- 当<变量>小于<目标值>，执行<语句内容>。
- 当<变量>大于<目标值>，则不会执行<语句内容>。
- 当每次执行<语句内容>时，<变量>总是按照指定的步长增加其值。步长可以是任意的整数值。

如果不指定步长，则其缺省值是 1。当<变量>大于<目标值>时，退出循环。

从某种意义上理解，FOR 循环的原理就像复印机，在复印机上先预设要复印的份数，在此即循环的条件，当条件满足时，即复印的张数等于设置的张数，停止复印。

FOR 循环是循环语句中最常用的一种，FOR 循环体现了一种规定次数、逐次反复的功能，但由于代码编写方式不同，也可以实现其他循环功能，下面通过一个实例，演示如何使用 FOR 循环。



示例：使用 FOR 循环实现 2 的五次方计算。

```

VAR
Counter: BYTE; (*循环计数器*)
Var1: WORD; (*输出结果*)
END_VAR

```

```

FOR Counter:=1 TO 5 BY 1 DO
Var1:=Var1*2;
END_FOR;

```

假设 Var1 的初始值是 1，那么循环结束后，Var1 的值为 32。



如果<目标值>等于<变量>的极限值，则会进入死循环。假设上例中的计数变量 Counter 的类型为 SINT (-128~127)，将<目标值>设定为 127 时，控制器会进入死循环。故不能对<目标值>设极限值。

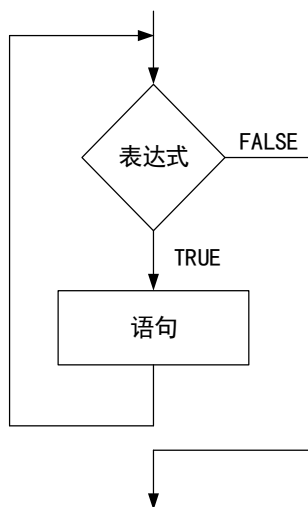
2) WHILE 循环

WHILE 循环与 FOR 循环使用方法类似。二者的不同之处是，WHILE 循环的结束条件可以是任意的逻辑表达式。即可以指定一个条件，当满足该条件时，执行循环，具体格式如下：

```
WHILE <布尔表达式>
<语句内容>;
END_WHILE;
```

WHILE 循环的执行顺序如下：

- ◆ 计算<布尔表达式>的返回值。
- ◆ 当<布尔表达式>的值为 TRUE 时，重复执行<语句内容>。
- ◆ 当<布尔表达式>初始值为 FALSE，那么指令<语句内容>不会被执行，跳转至 WHILE 语句的结尾。其流程图如下所示：



如果<布尔表达式>的值始终为 TRUE，那么将会产生死循环，应当避免死循环的产生。可以通过改变循环指令的条件来避免死循环的产生。例如：利用可增减的计数器避免死循环的产生。

WHILE 语句在像在工程中控制一台电机，当按下“启动”按钮时（布尔表达式为 TRUE 时），电机不停的旋转，当按下“停止”按钮后（布尔表达式为 FALSE 时），电机也立即停止。下面通过一个实例，演示如何使用 WHILE 循环。



示例：只要计数器不为零，则始终执行循环体内的程序。

```
VAR
Counter: BYTE; (*计数器*)
Var1: WORD;
END_VAR
```

```
WHILE Counter <> 0 DO
Var1 := Var1 * 2;
Counter := Counter - 1;
END_WHILE
```

在一定的意义上，WHILE 循环比 FOR 循环的功能更加强大，这是因为在执行循环之前，WHILE 循环不需要知道循环的次数。因此，在有些情况下，只使用这两种循环就可以了。然而，如果清楚地知道了循环的次数，那么 FOR 循环更好，因为 FOR 循环可以避免产生死循环。

3) REPEAT 循环

REPEAT 循环与 WHILE 循环不同，因为只有在指令执行以后，REPEAT 循环才检查结束条件。这就意味着无论结束条件怎样，循环至少执行一次。

```
REPEAT
< 语句内容>
UNTIL
< 布尔表达式>
END_REPEAT;
```

REPEAT 循环的执行顺序如下：

- ◆ 当<布尔表达式>的值为 FALSE 时，执行<语句内容>。
- ◆ 当<布尔表达式>的值为 TRUE 时，停止执行<语句内容>。
- ◆ 在第一次执行<语句内容>后，如果<布尔表达式>的值为 TRUE，那么<语句内容>只被执行一次。



如果<布尔表达式>的值始终为 TRUE，那么将会产生死循环，应当避免死循环的产生。可以通过改变循环指令部分的条件来避免死循环的产生。例如：利用可增减的计数器避免死循环的产生。



示例：REPEAT 循环举例，当计数器为 0 时，则停止循环。

```
VAR
Counter: BYTE;
END_VAR

REPEAT
Counter := Counter+1;
UNTIL
Counter=0
END_REPEAT;
```

此例的结果为，每个程序周期都进入该 REPEAT 循环，Counter 为 BYTE (0-255)，即每个周期内都进行了 256 次自加一计算。

因为之前提到的“这就意味着无论结束条件怎样，循环至少执行一次”，所以每当进入该 REPEAT 语句时，Counter 先为 1，每个周期内都执行了 256 遍的 Counter := Counter+1 指令，直到将 Counter 变量累加至溢出为 0，跳出循环。再被加到溢出，如此往复。

5、跳转语句

1) EXIT语句

如果 FOR、WHILE 和 REPEAT 三种循环中使用了 EXIT 指令，那么无论结束条件如何，内循环立即停止，具体格式如下：

```
EXIT;
```



示例：使用 EXIT 指令避免当使用迭代语句时中出现除零。

```
FOR Counter:=1 TO 5 BY 1 DO
INT1:= INT1/2;
IF INT1=0 THEN
EXIT; (* 避免程序除零*)
END_IF
Var1:=Var1/INT1;
END_FOR
```

当 INT1 等于 0 时，则 FOR 循环结束。

2) CONTINUE语句

该指令为 IEC 61131-3 标准的扩展指令，CONTINUE 指令可以在 FOR、WHILE 和 REPEAT 三种循环中使用。

CONTINUE 语句中断本次循环，忽略位于它后面的代码而直接开始一次新的循环。当多个循环嵌套时，CONTINUE 语句只能使直接包含它的循环语句开始一次新的循环，具体格式如下：

CONTINUE;



示例：使用 CONTINUE 指令避免当使用迭代语句时中出现除零。

```
VAR
Counter: BYTE; (*循环计数器*)
INT1,Var1: INT; (*中间变量*)
Erg: INT; (*输出结果*)
END_VAR

FOR Counter:=1 TO 5 BY 1 DO
INT1:= INT1/2;
IF INT1=0 THEN
CONTINUE; (*为了避免除零*)
END_IF
Var1:=Var1/INT1; (*只有当 INT1 不等于 0 的情况下才执行*)
END_FOR;
Erg:=Var1;
```

3) JMP 语句

跳转语句，跳转指令可以用于无条件跳转到使用跳转好标记的代码行，具体格式如下：

```
< 标识符>:
.
JMP < 标识符>;
```

<标识符>可以是任意的标识符，它被放置在程序行的开始。JMP 指令后面为跳转目的地，即一个预先定义的标识符。当执行到 JMP 指令时，将跳转到标识符所对应的程序行。



必须避免制造死循环，可以配合使用 IF 条件控制跳转指令。



示例：使用 JMP 语句实现计数器在 0..10 范围内循环。

```
VAR
nCounter: BYTE;
END_VAR

Label1:nCounter:=0;
Label2:nCounter:=nCounter+1;
IF nCounter<10 THEN
JMP Label2;
ELSE
JMP Label1;
END_IF
```

上例中的 Label1 和 Label2 属于标签，不属于变量，故在程序中不需要进行变量声明。通过 IF 语句判断计数器是否在 0-10 的范围内，如果在范围内，则执行语句 JMP Label2，程序会

在下一个周期跳转到至 Label2，执行程序 `nCounter:=nCounter+1`，将计数器进行自加 1，反之，则会跳转至 Label1，执行 `nCounter:=0`，将计数器清零。
此例子中的功能同样可以通过使用 FOR，WHILE 或者 REPEAT 循环来实现。通常情况下，应该避免使用 JMP 跳转指令，因为这样降低了代码的可读性和可靠性。

4) RETURN 指令

RETURN 指令是返回指令，用于退出程序组织单元 (POU)，具体格式如下：

```
RETURN;
```



示例：使用 IF 语句作为判断，当条件满足时，立即终止执行本程序。

```
VAR
nCounter: BYTE;
bSwitch: BOOL; (*开关信号*)
END_VAR
```

```
IF bSwitch=TRUE THEN
RETURN;
END_IF;
nCounter:= nCounter +1;
```

当 bSwitch 为 FALSE 时，nCounter 始终执行自加 1，如 bSwitch 为 TRUE 时，nCounter 保持上一周期的数值，立刻退出此程序组织单元 (POU)。

6、空语句

即什么内容都不执行。
具体格式如下：

7、注释

1) 添加注释

注释是程序中非常重要的一部分，它使程序更加具有可读性，同时不会影响程序的执行。在 ST 编辑器的声明部分或执行部分的任何地方，都可以添加注释。
在 ST 语言中，有两种注释方法：

方法一：多行注释以 (*) 开始，以 *) 结束。这种注释方法允许多行注释，如下图所示：

```
(*
    bOperationActive:=FALSE;
    bOrderActive:=FALSE;
    bRecipeActive:=FALSE;
    bInfoActive:=FALSE;
    bServiceActive:=FALSE;
    bSimulationActive:=FALSE;
*)
IF iMainAreaIndex = 0 THEN
    bOperationActive:=TRUE;
ELSIF iMainAreaIndex = 1 THEN
    bOrderActive:=TRUE;
```

方法二：单行注释以 “//” 开始，一直到本行结束。这是单行注释的方法，如下图所示：

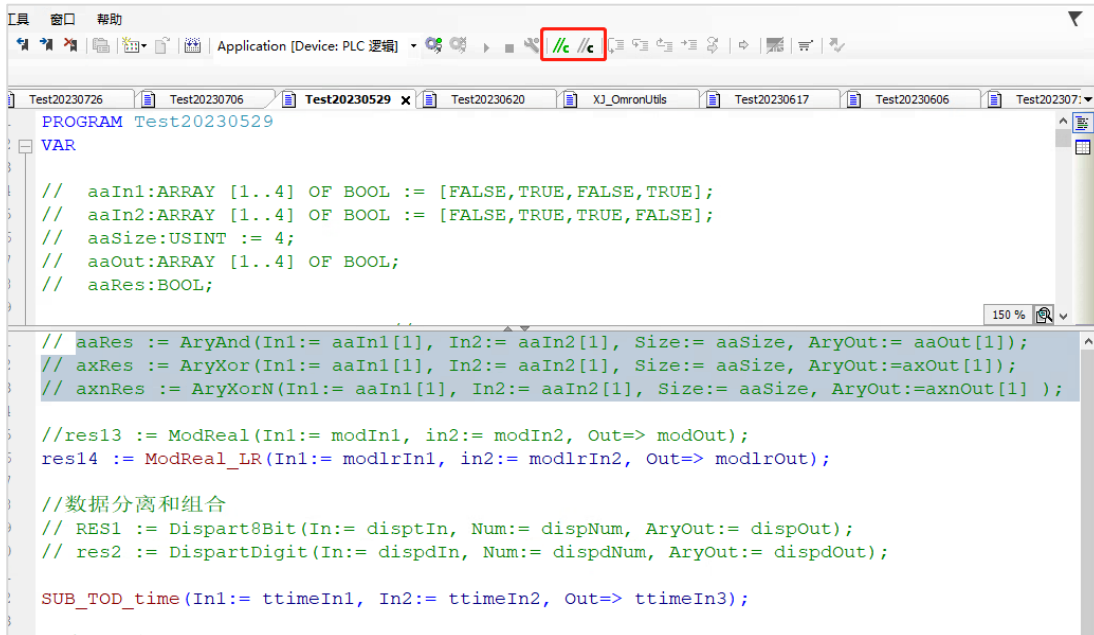
```

// gesture handling:
// only when mouseup was done
IF xRight AND bDragCanStart = FALSE THEN
    xRight := FALSE;
    IF iMainAreaIndex < MAX_MODULES-1 THEN
        iMainAreaIndex := iMainAreaIndex + 1;
        bIndexChanged := TRUE;
    END_IF

```

2) 注释切换

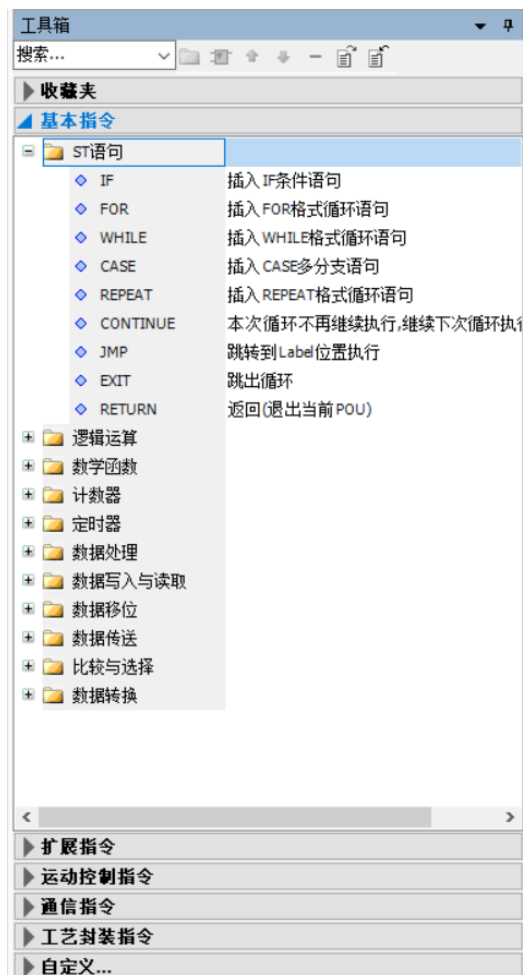
通过快捷键 Ctrl + U、Ctrl + M 或从菜单栏点击实现快速注释或取消注释选择代码。



7-2-4. ST 编辑

1、ST 工具箱

工具类别界面，如下：



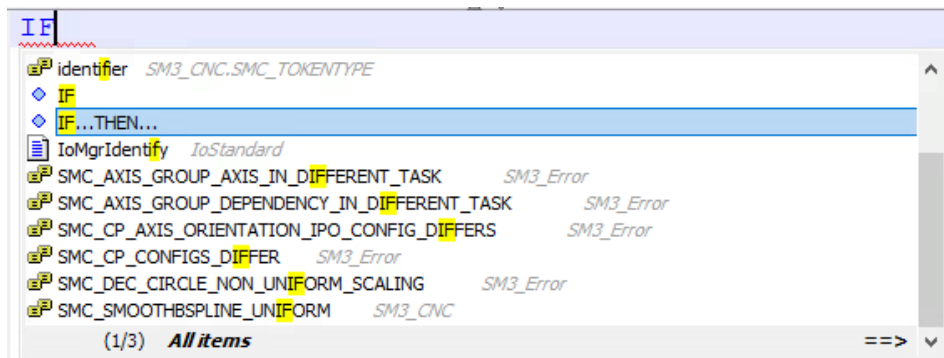
工具箱中包含基本指令、扩展指令、运动控制指令、通信指令、工艺封装指令，可通过拖拽或双击的形式拖入程序编写区，例如ST语句、IF语句、WHILE语句、REPEAT、CASE语句、CONTINUE、JMP、EXIT、RETURN，插入时自动插入语句模板。

2、智能提示

1) 关键字匹配

输入 ST 语句类型关键字，能够自动匹配，语句包含 IF 语句、WHILE、FOR、CASE、REPEAT，格式化模板见附录语句模板。

如下图所示，输入 IF 能够弹出响应联想语句：



2) TAB键快捷功能

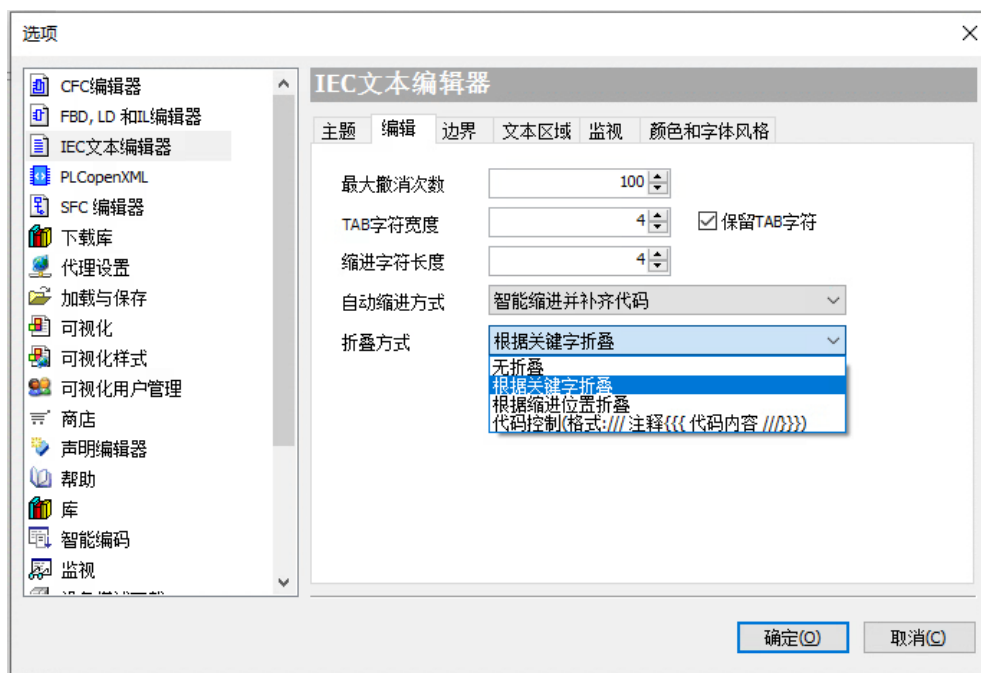
- 能够对功能块、函数、方法、动作、程序自动格式化输入、输出；
- 能够对功能块实例及实例的方法、动作自动格式化输入、输出；
- 能够对 IF、WHILE、FOR、CASE、REPEAT 语句自动格式化，格式化模板见附录语句模板，

如功能类型名、功能块实例等输入后按 Tab 键自动格式化。

3、折叠缩放

- 折叠方式支持关键字：关键字包含 VAR、VAR_INPUT、VAR_GLOBAL、VAR_OUTPUT、VAR_IN_OUT、VAR_TEMP、VAR_STAT、VAR_EXTERNAL、CASE、FOR、REPEATED、IF/ELSE/ELSIF、WHILE、STRUCT、UNION、TYPE、__TRY、__CATCH、__FINALLY。
- 如果选择自动缩进功能中的智能缩进，根据上述的关键字，自动加 Tab 长度，如果选择智能缩进并自动补齐，自动给关键字补齐结尾，如 VAR，FOR，WHILE 自动补齐结束标记，支持嵌套。
- 智能缩进时，如果上行是关键字，换行后自动增加 Tab 字符，如果非关键字，和上行同缩进。块高亮显示。括号、中括号、WHILE、FOR、IF、ELSE、CASE、REPEAT、STRUCT、UNION、TYPE、
- TRY 等类别之间显示块高亮信息，在文本边界和文本区域都有高亮显示标记。

此处，可通过点击菜单栏内的“工具”->“选项”菜单打开“IEC 文本编辑器”设置界面，用户可根据实际的需求设置折叠方式。如下图：



7-3. 梯形图

7-3-1. 简介

梯形图来源于美国，它基于图形表示的继电器逻辑，是 PLC 编程中被最广泛使用一种图形化语言。梯形图程序的左、右两侧有两垂直的电力轨线，左侧的电力轨线名义上为功率流从左向右沿着水平梯级通过各个触点、功能、功能块、线圈等提供能量，功率流的终点是右侧的电力轨线。每一个触点代表了一个布尔变量的状态，每一个线圈代表了一个实际设备的状态，功能或功能块与 IEC 1131-3 中的标准库或用户创建的功能或功能块相对应。

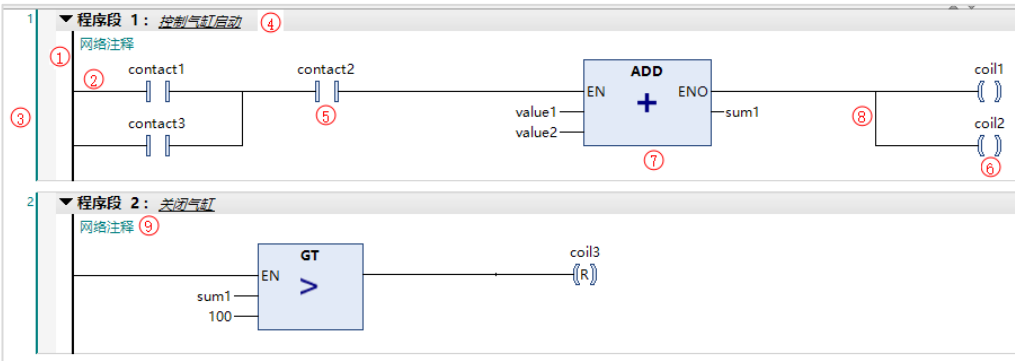
梯形图是国内应用最为广泛的编程语言，它也是 IEC 1131-3 的三种图形化编程语言种一种，梯形图是传统 PLC 使用得最多的图形编程语言，也被称为 PLC 的第一编程语言。根据梯形图中各触点的状态和逻辑关系，求出与图中各线圈对应的编程元件的状态，称为梯形图的逻辑解算。

梯形图中的某些编程元件沿用了继电器这一名称，如线圈、触点等，但是它们不是真实的物理继电器，而是一些存储单元（软继电器），每一软继电器与 PLC 存储器中映像寄存器的一个存储单元相对应。该存储单元如果为“TRUE”状态，则表示梯形图中对应软继电器的线圈“通电”，其常开触点接通，常闭触点断开，称这种状态是该软继电器的“TRUE”或“ON”状态。如果该存储单元为“FALSE”状态，对应软继电器的线圈和触点的状态与上述的相反，称该软继电器为“FALSE”或“OFF”状态。使用中也常将这些“软继电器”称为编程元件。

7-3-2. LD 术语解释

梯形图的执行过程是按照从左至右，从上到下的顺序进行执行。

LD 程序由在程序段电源线上串联或并联的各个元素组成。至少有一个梯级从电源线引出。程序段编程从梯级左边缘开始。可以通过使用多个梯级和分支来延伸电源线。

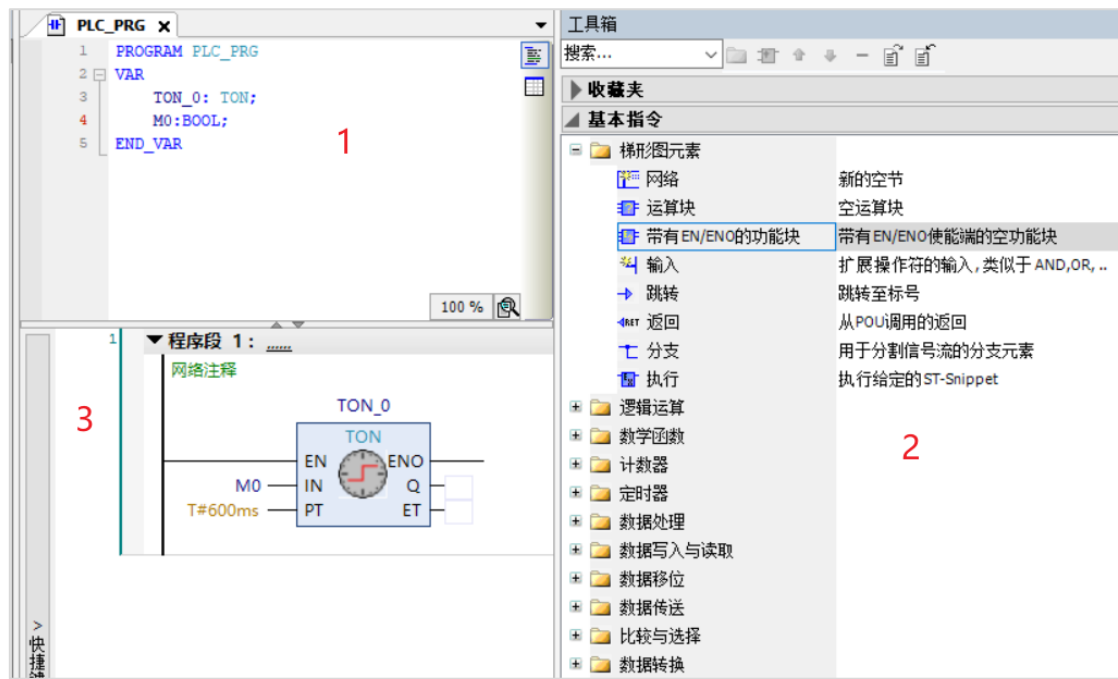


编号	术语	说明
①	电源线	每个 LD 程序段都包含至少有一个梯级的电源线。通过添加其它梯级可扩展程序段。可以使用分支在特定梯级中创建并联结构。
②	梯级	梯形图网络结构的最小单位，每个梯级是由若干个输入、输出节点串并联而成。两个相邻梯级之间不允许有垂直连接线。
③	程序段	程序段主要用于对多段不同工艺的梯形图程序进行分段标识；一个程序段可以包含多个梯级。
④	程序段标题	简要标识程序段用途，折叠开始时，程序段标题可以直接帮助定位程序。
⑤	触点	可以使用触点创建或中断两个元素之间的载流连接。 电流从左向右传递。可以使用触点查询操作数的信号状态或值，并根据电流的结果对其进行控制。
⑥	线圈	可以使用线圈控制二进制操作数。线圈可根据逻辑运算结果的信号状态置位或复位二进制操作数。
⑦	运算块	函数、功能块、程序、执行块、动作、方法
⑧	分支	可以使用分支来设计并联电路。分支插在主梯级中。可以将多个触点插入分支中，从而实现串联的并联电路。这样便能设计复杂的梯

编号	术语	说明
		形逻辑。
⑨	网络注释	可以详细描述程序段功能及程序设计注释。

7-3-3. 编辑器显示

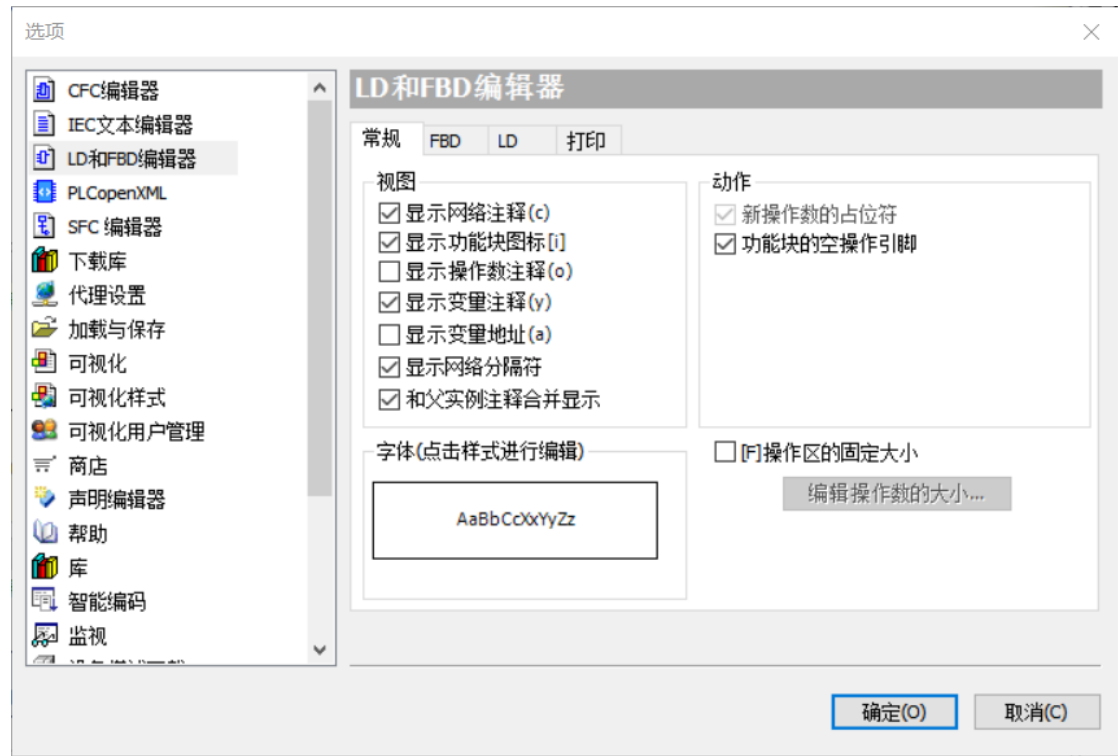
1、默认视图



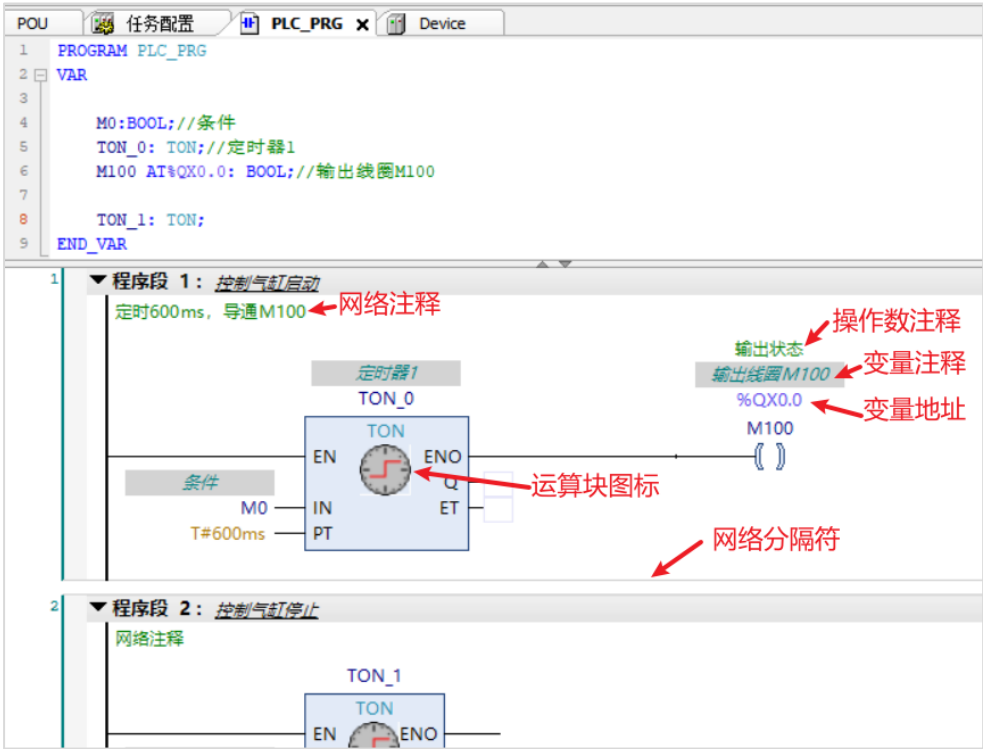
上图中 1 处为变量定义区，2 处为工具箱，3 处为梯形图编程区。
梯形图语言中支持嵌入 EXECUTE 执行块，用于插入 ST 语言程序段。

2、梯形图设置

可以通过【工具】-【选项】-【LD 和 FBD 编辑器】设置梯形图界面的字体、操作数及注释显示。



LD 设置对象显示如下图：



属性	默认	说明
显示网络注释	勾选	如果激活此选项，梯形图每个网络可以编辑网络注释。如果增加了网络注释，在网络标题下增加一行显示网络注释，如果网络注释不存在或被删除，不会产生空白行，保持显示“网络注释”。 可以通过双击网络注释进行编辑，或通过菜单栏命令【LD/FBD】-【编辑网络注释】进行编辑。
显示功能块图标	勾选	如果激活此选项，如果运算块定义了图标，运算块中间会显示图标。标准操作符（如 ADD、SUB）和功能块（如 TON、TOF）都定义了图标，用户自定义的函数、功能块或者程序可以通过在左侧“设备”界面选中对象，点击鼠标右键，【属性】-【位图】-【点击此处选择与工程有关的位图】来添加图片，形成运算块图标。
显示操作数注释	不勾选	如果激活此选项，梯形图界面每个操作数上都可以编辑和显示操作数注释。操作数是一个编程概念，如变量、常量、地址都是操作数。由于梯形图中不一定都使用变量，如常量或者地址，这时可以通过操作数注释对它们进行注释。 当激活“显示操作数注释”选项后，鼠标选中操作数的名称，右键点击“编辑操作数注释”菜单，可以手动输入操作数的注释。如果未激活此选项，不显示空白行。
显示变量注释	勾选	如果激活此选项，梯形图界面变量上会显示变量声明时的注释。变量注释来自于变量声明，不能编辑。如果未激活此选项，不显示空白行
显示变量地址	不勾选	如果激活此选项，且一个地址被分配给一个符号（变量），那么这个地址就会显示在变量名的上方。如果没有变量或没有映射地址，则不显示空白行如果未激活此选项，不显示空白行
显示网络分隔符	勾选	如果激活此选项，各个网络之间会显示一个分隔符。
和父实例注释合并显示	勾选	如果激活此选项，不仅显示实例成员注释，还会显示实例中的成员的注释。如实例化定时器名字为 TON1，注释设为“定时器 1”，对其引脚 Q 添加注释名“输出”，此时程序里调用该 TON1.Q 会显示注释“定时器 1 输出”，否则只会显示“定时器 1”。

属性	默认	说明
新操作数的占位符	勾选	保留 Codesys 功能
功能块的空操作引脚	勾选	如果激活此选项，新增加运算块时，运算块输入和输出引脚用空字符，如果未激活此选项，运算块输入和输出引脚用“???”字符
操作区的固定大小	不勾选	如果激活此选项，可编辑操作数大小。打开对话框设置字符数和行数，并以操作数固定大小设置对话框参数配置显示。

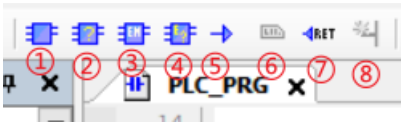
3、菜单栏 LD/FBD



4、工具栏



编号	功能
①	插入触点（Ctrl+K） 如果该触点对应当前布尔变量值为 True 时，则该触点吸合。反之，当布尔变量值为 False 时，右侧连接元素状态为 False。
②	插入常闭触点 如果该触点对应当前布尔变量值为 False 时，则该常闭触点处于吸合状态。反之，当布尔变量值为 True 时，触点断开，则右侧连接元素状态为 False。
③	插入并联触点（向下）（Ctrl+R） 可以进行多个触点的并联，在触点下侧并联插入常开触点。两个并联触点中只需一个触点为 True，则平行线传输 True。
④	插入并行运算块（下方） 可以进行多个触点的并联，在触点上侧并联插入常开触点。两个并联触点中只需一个触点为 True，则平行线传输 True。
⑤	插入并联触点（向上）（Ctrl+P）
⑥	插入线圈（Ctrl+Shift+A） 左侧连接元素的状态被传递到有关的布尔变量和右侧连接元素，如果线圈左侧连接元素的状态为 TRUE，则线圈的布尔变量为 TRUE，反之线圈为 FALSE。
⑦	插入置位线圈 线圈中有一个 S。当左侧连接元素的状态为 TRUE 时，该线圈的布尔变量被置位并且保持，直到由 Reset（复位）线圈的复位。
⑧	插入复位线圈 线圈中有一个 R。当左侧连接元素的状态为 TRUE 时，该线圈的布尔变量被复位并且保持，直到由 Set（置位）线圈的置位。



编号	功能
①	插入运算块（Ctrl+B） 插入函数或功能块，根据弹出的对话框通过鼠标选择想要使用的函数及功能块。
②	插入空运算块（Ctrl+Shift+B） 直接插入矩形块，在“???”处直接输入函数或功能块名。
③	插入带 EN/ENO 的运算块（Ctrl+Shift+E） 只有当 EN 为 True 时，才执行函数或功能块并允许将状态传递至下游。
④	插入带 EN/ENO 的运算块 在“???”处直接输入函数或功能块名，只有当 EN 为 True 时，才执行函数或功能块并允许将状态传递至下游。
⑤	插入跳转（Ctrl+L）
⑥	插入标号
⑦	插入返回
⑧	插入输入（Ctrl+Q）

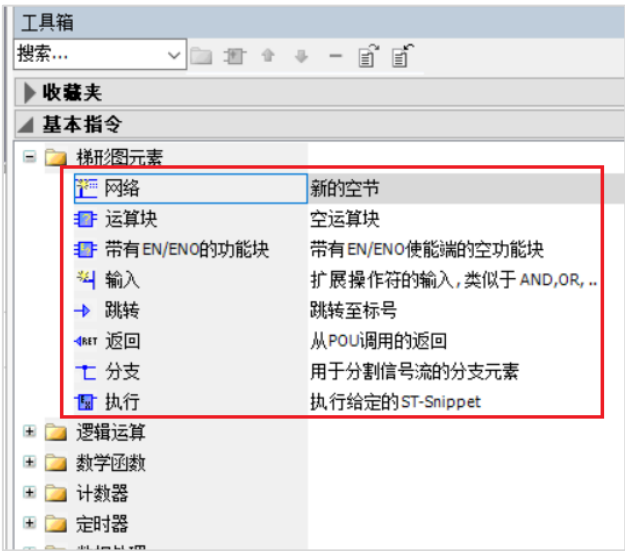


编号	功能
①	取反（Ctrl+N）
②	边沿检测（Ctrl+E）
③	置位/复位（Ctrl+M）



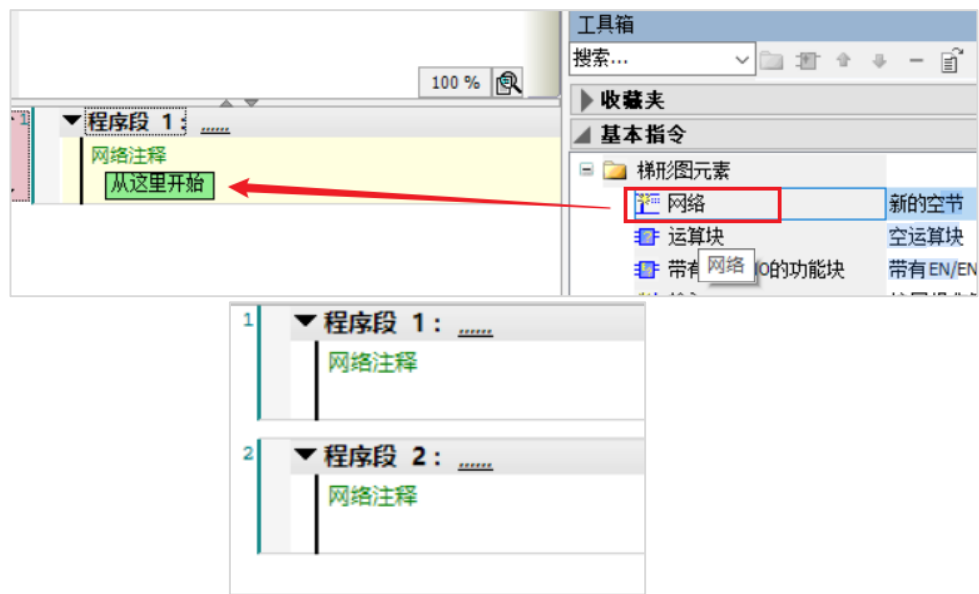
编号	功能
①	插入分支（Ctrl+Shift+V）
②	在上面插入分支
③	设置分支起点
④	设置输出连接（Ctrl+W）
⑤	展开所有网络（Ctrl+Shift+M）
⑥	折叠所有网络（Ctrl+Shift+N）
⑦	更新参数（Ctrl+U） 表示批量更新当前梯形图编辑器中所有运算块的输入和输出参数。如果运算块的输入/输出参数发生变化时，通过执行“更新参数”命令，更新运算块的输入/输出参数。 例如 FB 成员名称、类型等改变，或者新增、删除输入输出参数，可以点击“更新引脚”，此时外部实例化调用了该 FB 块的功能块引脚会自动更新。
⑧	添加网络节点（Ctrl+Shift+J）
⑨	删除网络节点（Ctrl+Shift+K）

7-3-4. 梯形图元素



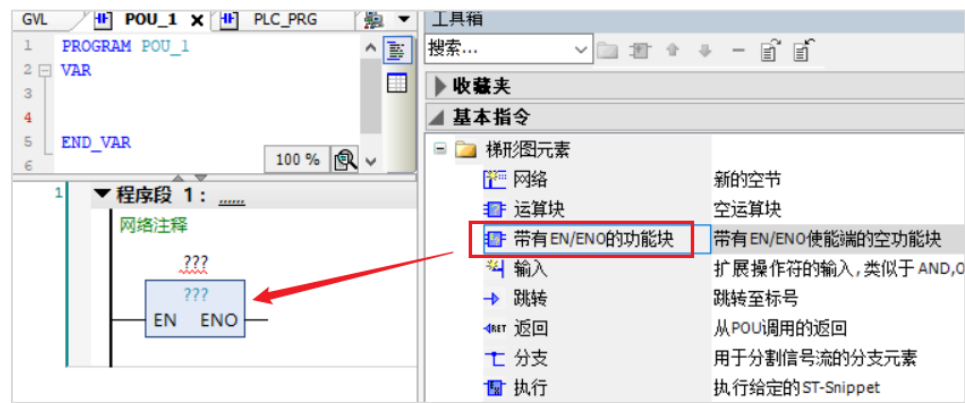
1、网络

右侧工具箱，选中“网络”拖拽到空白程序段里的“从这里开始”，可以新建一个程序段。



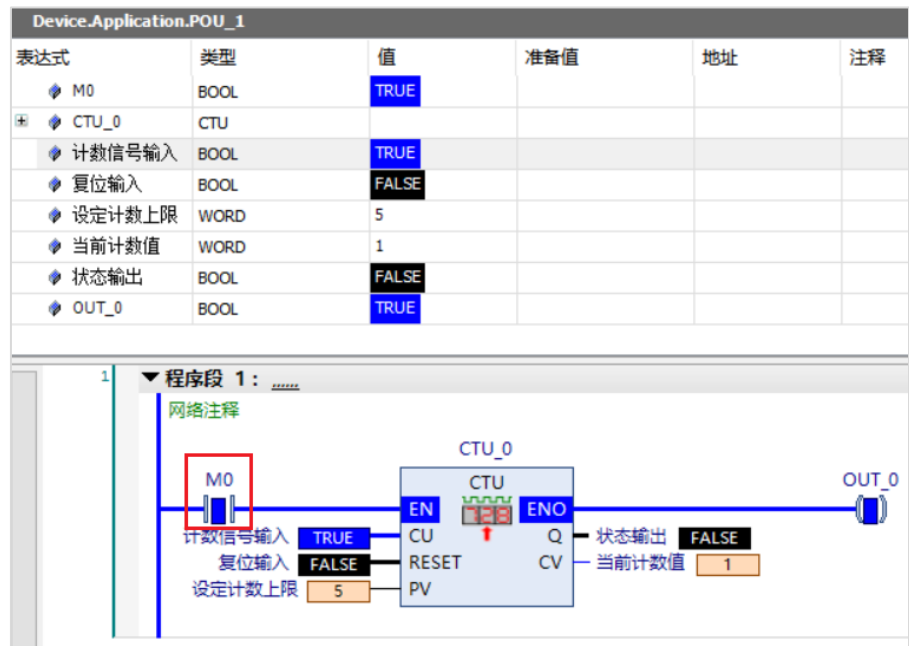
2、带有 EN/ENO 的功能块

在“???”处直接输入函数或功能块名，只有当 EN 为 TRUE 时，才执行函数或功能块并允许将状态传递至下游。



函数和功能块的名称显示在框内的上中部，功能块需要将其实例化，实例名列于框外的上中部。用功能块的实例名作为其在项目中的唯一标识。

 示例：功能块调用实参的设置。



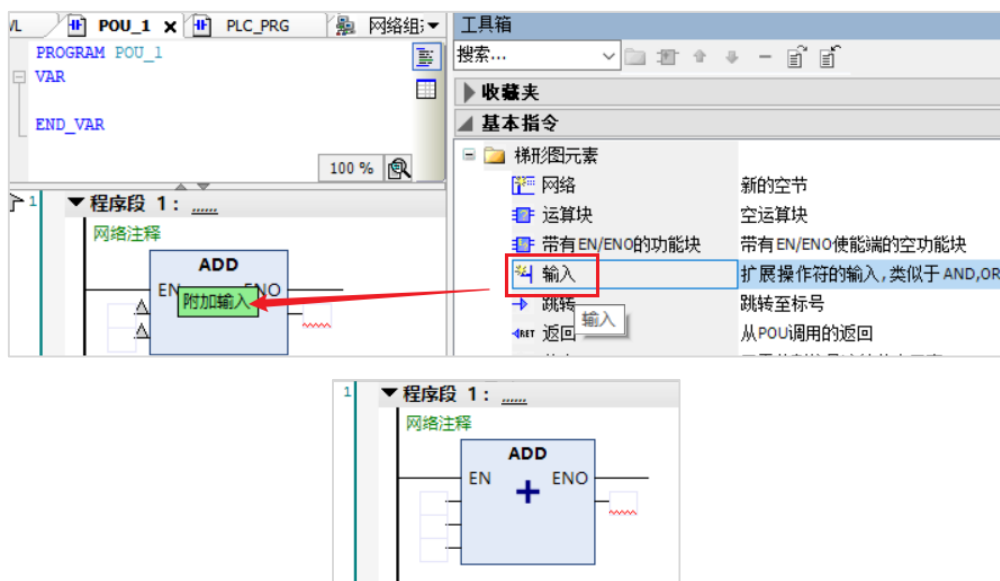


- 当 M0 为 FALSE 时, 增计数器功能块 CTU_0 本体定义的操作不被执行, ENO 的值也相应为 FALSE。
- 当 M0 为 TRUE 时, 增计数器功能块 CTU_0 允许执行, 此时 ENO 的值为 TRUE, OUT_0 为被导通为 TRUE。
- 当计数信号输入由 FALSE->TRUE 时, CTU_0 会计数加 1, 当前计数值显示加 1。

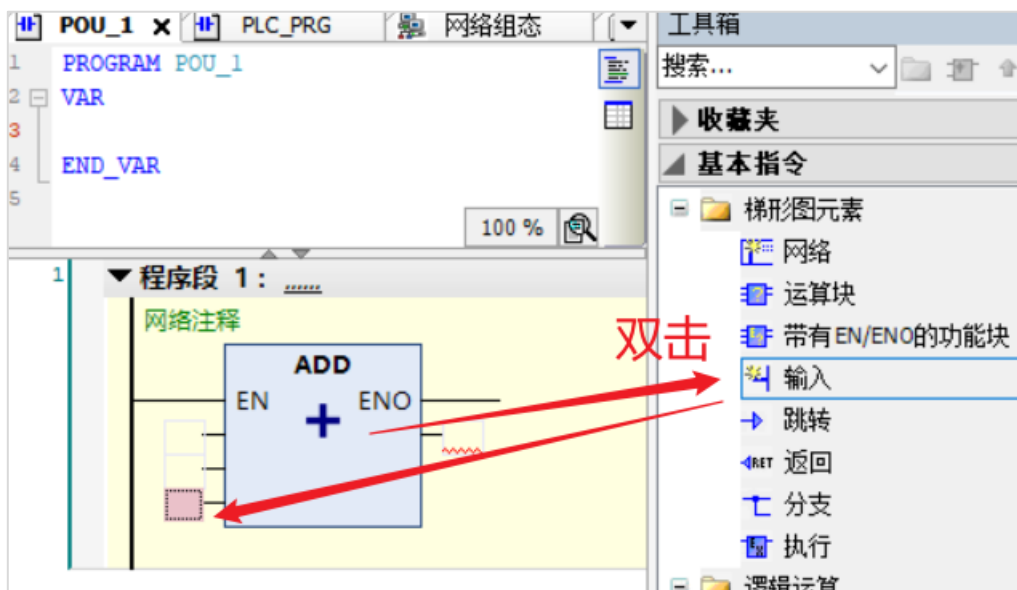
3、输入

扩展操作符的输入。

方法一：点击右侧“输入”拖拽到操作符块界面里显示的“附加输入”，可以看到下图中 ADD 的输入引脚会多 1 个。



方法二：先点击选中操作符块，再在右侧工具箱界面双击“输入”，可以看到下图中 ADD 的输入引脚会多 1 个。

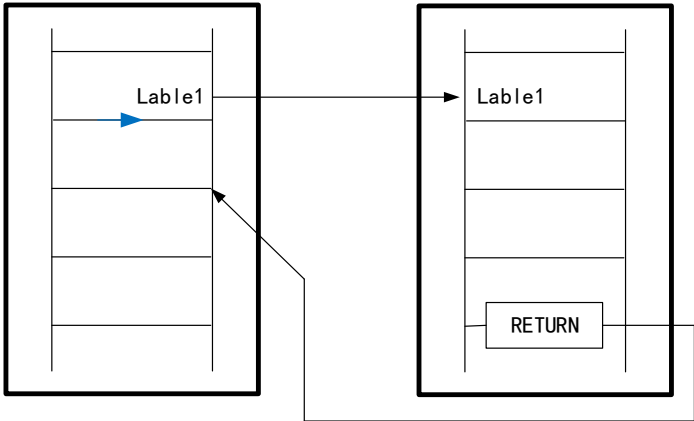


4、跳转、返回

跳转的标签是一个可选的标识符且当定义跳转时可以确定其地址。它可以包含任何字符。

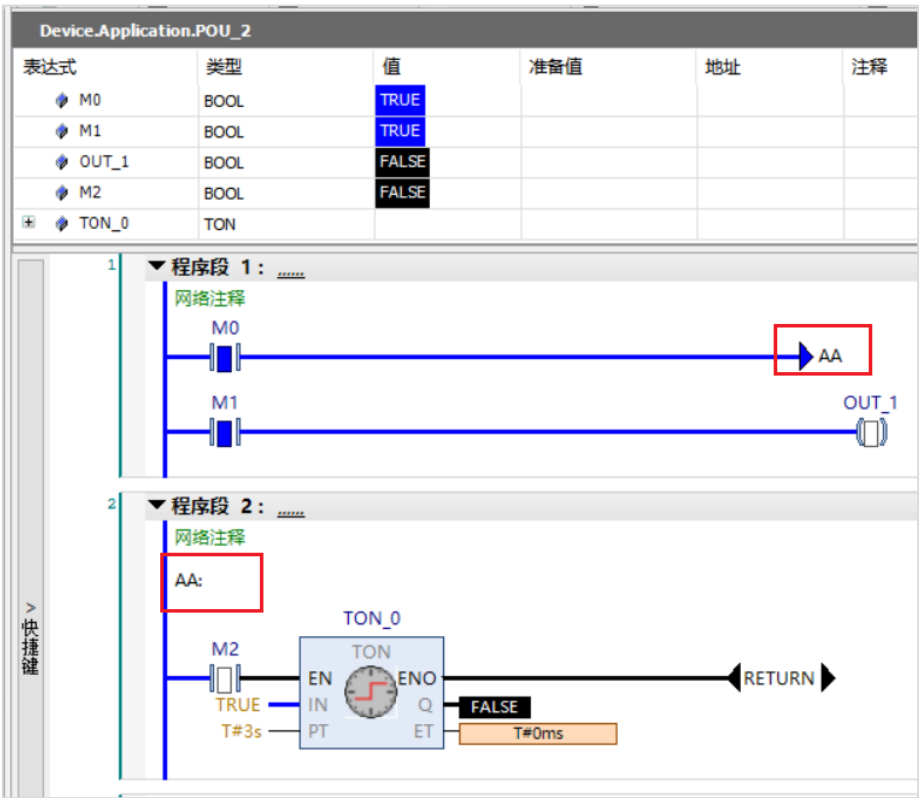
当满足跳转条件时, 程序跳转到标号的节开始执行, 直至该部分程序执行到 RETURN 时, 返回

原来的节并继续执行。其结构图如下图所示：



XS Studio 中使用梯形图的跳转指令和返回指令。

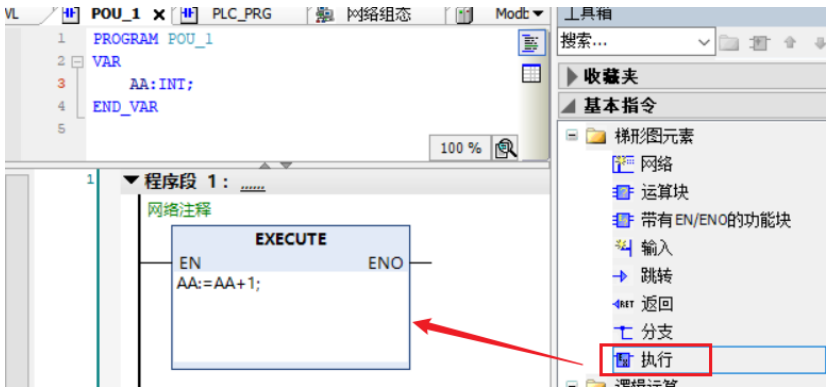
 **示例：**使用跳转指令执行程序的示例。



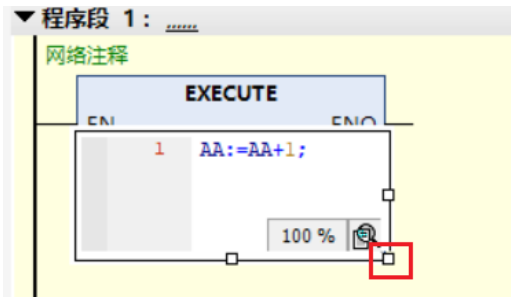
如图所示，M0 被置为 True，故执行跳转语句，根据标签 AA，程序跳转至程序段 2 的 AA，所以尽管后来 M1 被置为 ON，OUT_1 始终不会被置为 True，因为程序直接跳过了该语句。只有当 M0 为 False 且 M1 为 True 时，OUT_1 才会 True。

5、执行

执行块是一个可以编辑 ST 语句的块，单键文本区域进行编辑；执行块只有 EN 输入引脚、ENO 输出引脚。



在编辑状态可以拖拽执行块边框，实现执行块大小控制。



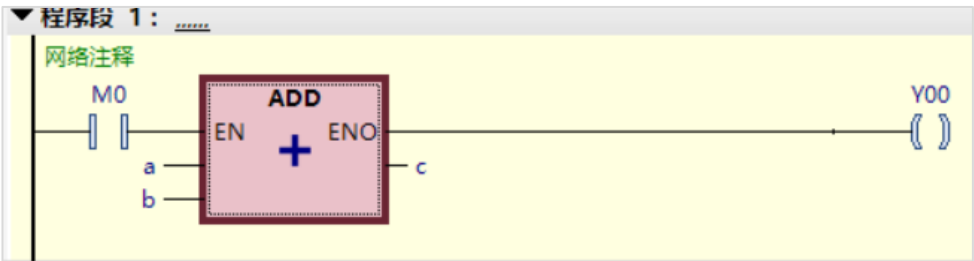
7-3-5. 梯形图操作

1、选择

选择对象可以是元件或者连线，选择可以单选也可以按下 Ctrl、Shift 多选，可以连续选择，也可以非连续选择。支持鼠标框选、 Ctrl 和 Shift 多选、全选。

1) 单选

如下图，选择 ADD 的运算块，然后复制再粘贴时会包含 ADD 运算块本身、输入 a、输入 b、输出 c。



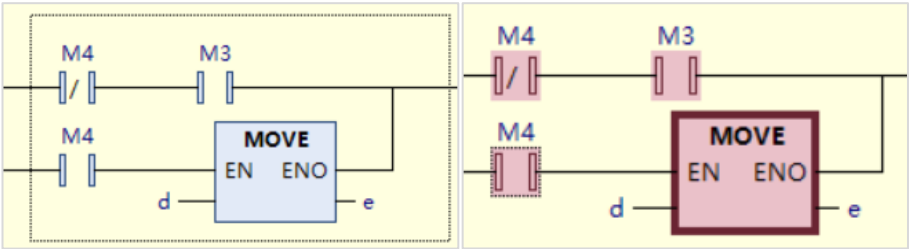
2) 多选

Ctrl 和 Shift 多选方式：

- Ctrl 多选：按下 Ctrl 时，如果当前元素没有选择则把当前元素增加到选择列表中；如果已经选择，则从选择列表中去除。
- Shift 多选：从上次选择元素到本次选择元素矩形内的元素进行选择。

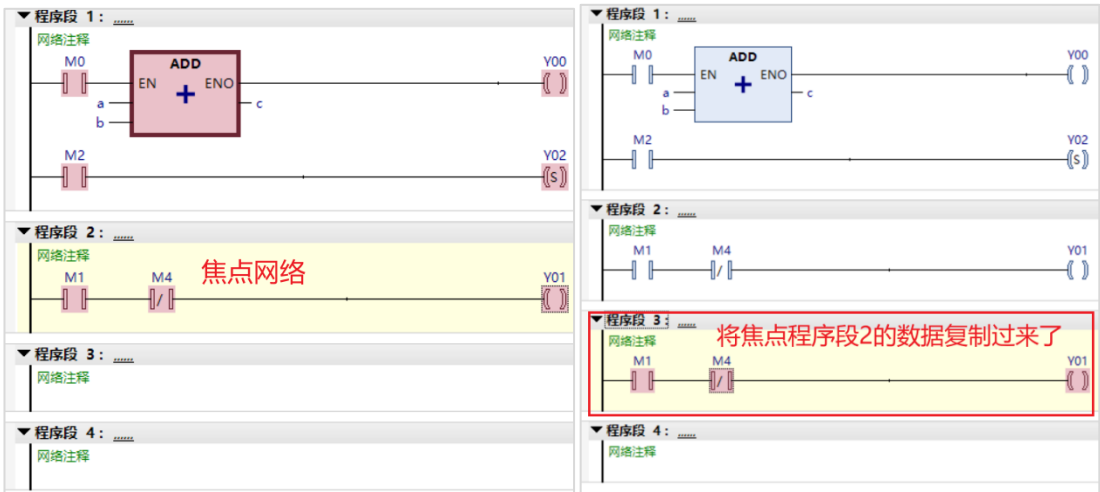
3) 框选

从鼠标按下位置到鼠标松开位置之间矩形内的元素被选择（只能单网络，不能跨网络）。如下图：



2、复制

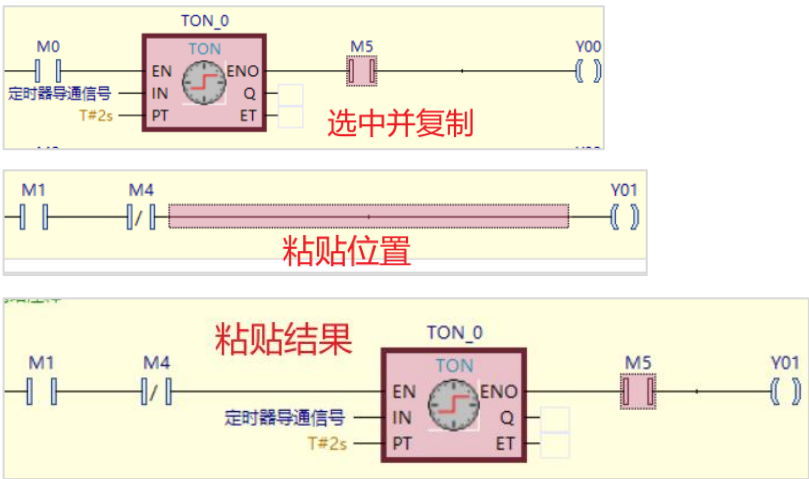
梯形图中可以复制的元素包括网络、触点、线圈、运算块、字符串、分支连线。
复制可以复制连续选择的元素，也可以复制非连续的，和选择有关。但是复制的元素只能单个网络内，如果跨网络选择，只能复制焦点网络选择的数据。



3、粘贴

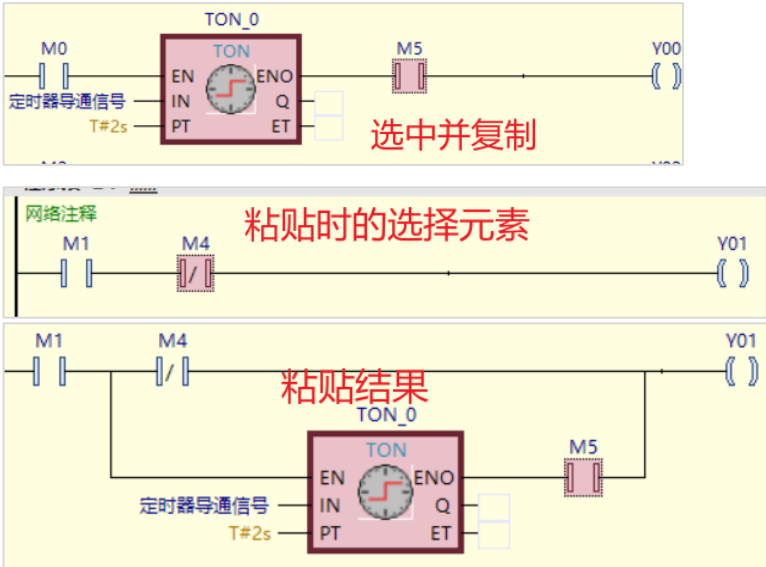
1) 连线上粘贴

在选择元件上粘贴时，和所选择的批量元件生成并联关系。选择的元件必须满足：在一条线上、必须连续、不能跨越分支、开始元素和结束元素不能再并联分支内部和外部。



2) 元素上粘贴

在选择元素上粘贴时，和所选择的批量元件生成并联关系。

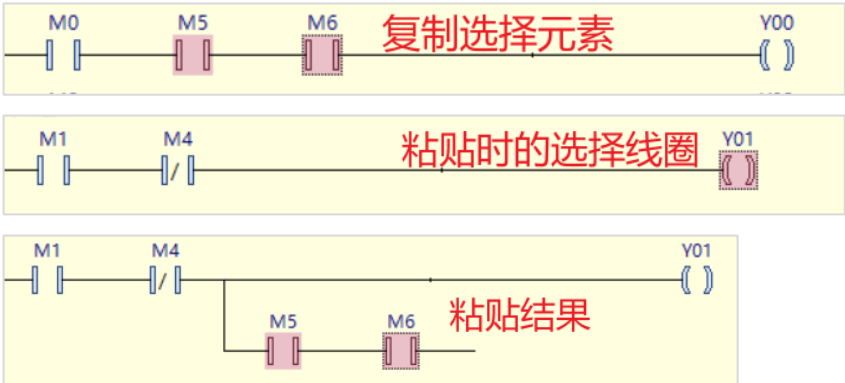


3) 线圈位置粘贴

由于线圈右侧不能存在元素，所以粘贴时有些特殊规则。跳转元素、返回元素规则和线圈相同，下面仅说明线圈。

- 粘贴时选择的是线圈

如果选择为线圈（选择元素并联），粘贴的元素位于一个新分支上，此分支位于线圈下面，如下图所示（选中 Y01 粘贴 M5 和 M6，M5 和 M6 在一个新分支上，分支位于 Y01 线圈下面）。

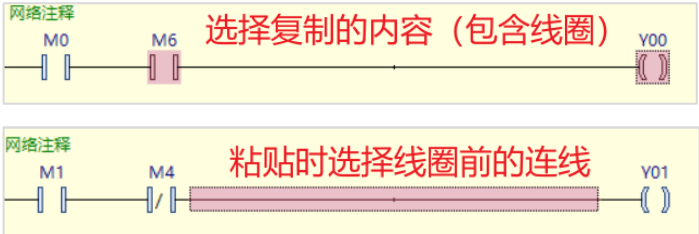


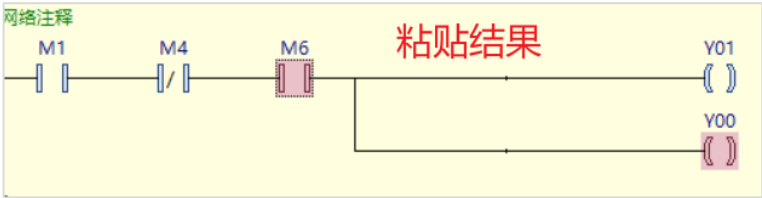
- 粘贴时选择的是线圈前的连线

如果选择为线圈之前的连线（选择连线串联），粘贴时，根据复制的元素内容，分为复制内容仅包含一个分支和复制内容包含多个分支。

- 复制内容仅包含一个分支

如果复制内容不包含线圈，复制元素直接和线圈串联。
如果复制内容包含线圈，则复制内容（除了线圈外）插入到连线选择位置，复制内容中的线圈和所选连线之后的线圈形成一个非闭合并联，如下图（选择 Y01 之前连线粘贴 M6、Y00，粘贴后，Y00 之前 M6 串联在 Y01 之前连线上，Y00 和 Y01 通过分支并联）。



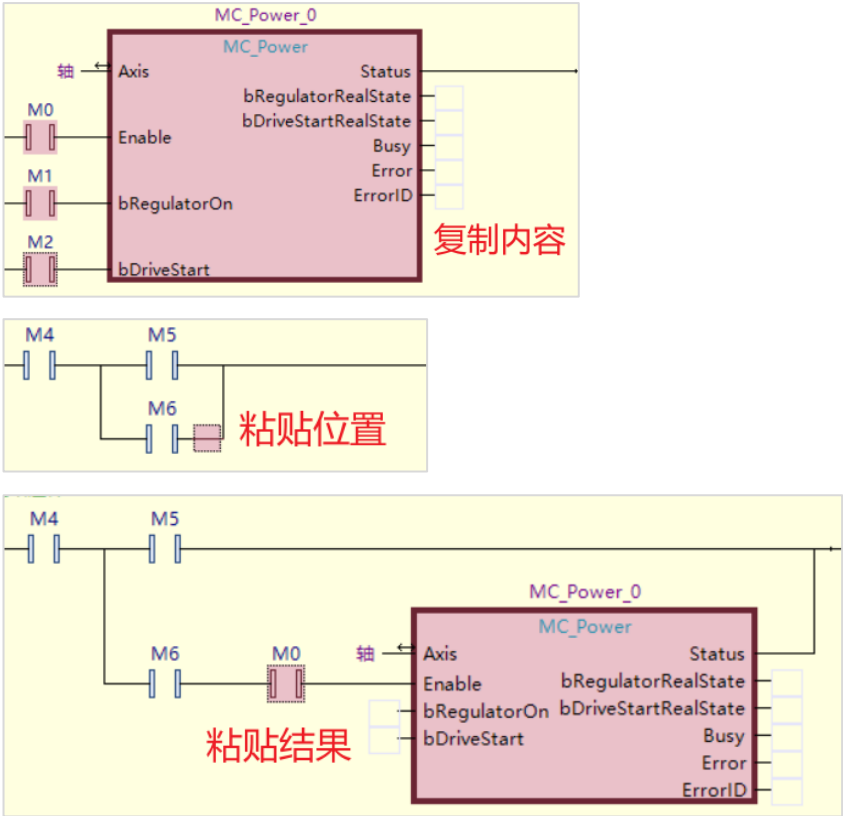


- 复制内容包含多分支

如果复制内容水平最后一个分支不包含线圈，在选择连线位置插入复制内容。
如果复制内容水平最后一个分支包含线圈，则把复制内容水平最后一个分支的线圈和选择连线之后的线圈并联，其它数据串并联不变。

注意：

- 复制元素为多输入连线运算块时的粘贴注意点。
- 多输入连线运算块只能在第一个分支非并联位置，所以粘贴位置如果不能包含多连线运算块时，会删除所复制运算块的第二个输入连线开始的连线元素。如下图，复制的运算块输入 M0、M1、M2 触点，粘贴时 M1、M2 被删除。



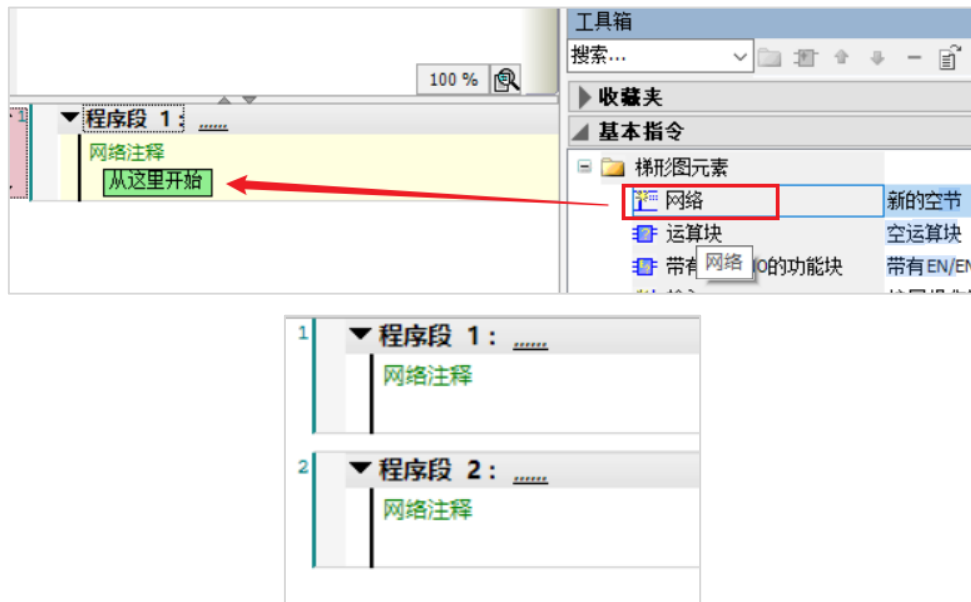
7-3-6. 网络/程序段功能

1、新建程序段

方法一：在梯形图空白编辑区（程序段外）点击鼠标右键，选择“插入网络”。



方法二：右侧工具箱，选中“网络”拖拽到空白程序段里的“从这里开始”。



2、展开/折叠

(1) 通过程序段标题栏“倒三角”按钮，折叠/展开当前选中程序段。



(2) 梯形图编辑区右键快捷菜单栏

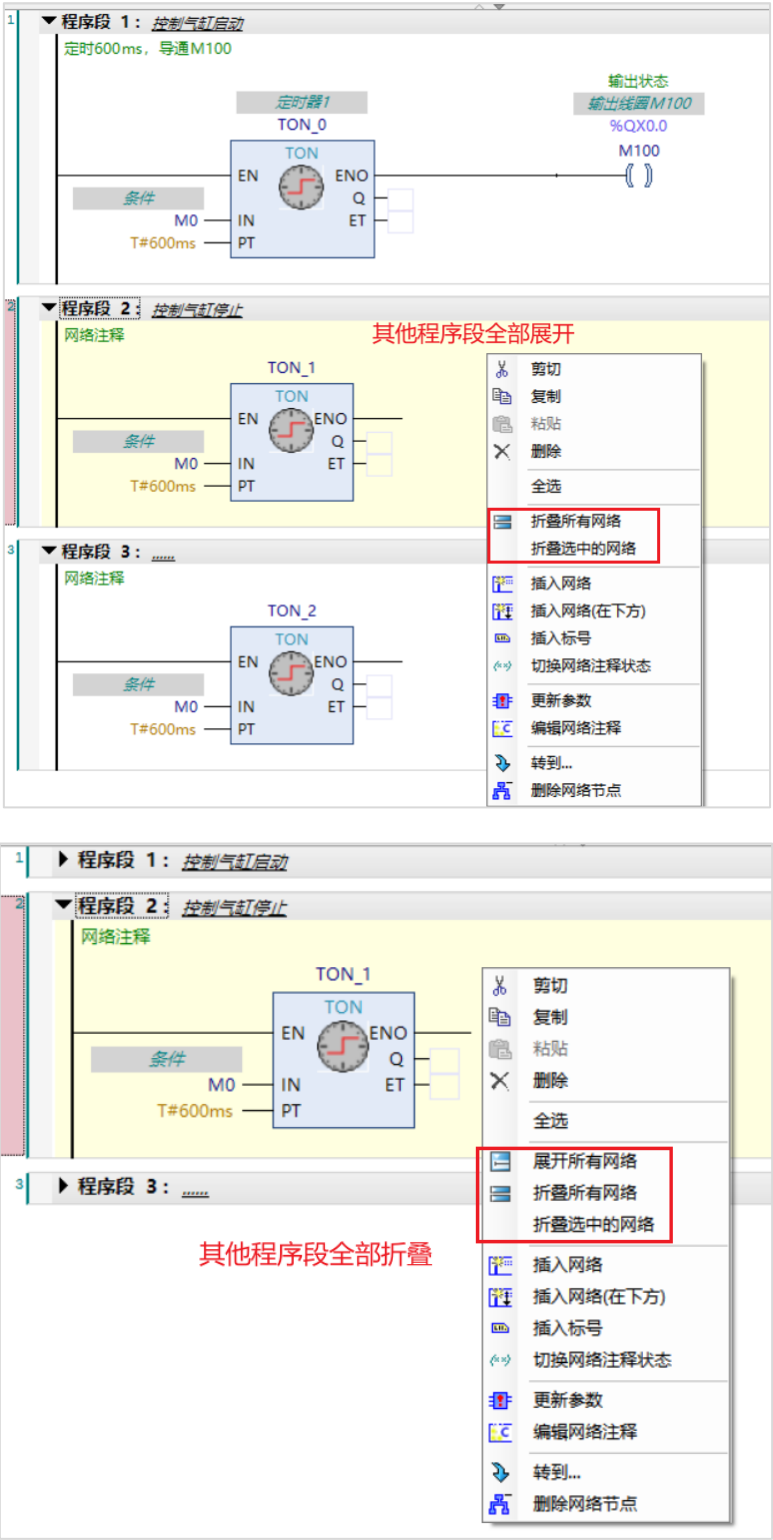
- 未选择程序段，右键快捷菜单栏的“展开/折叠程序段”情况如下：

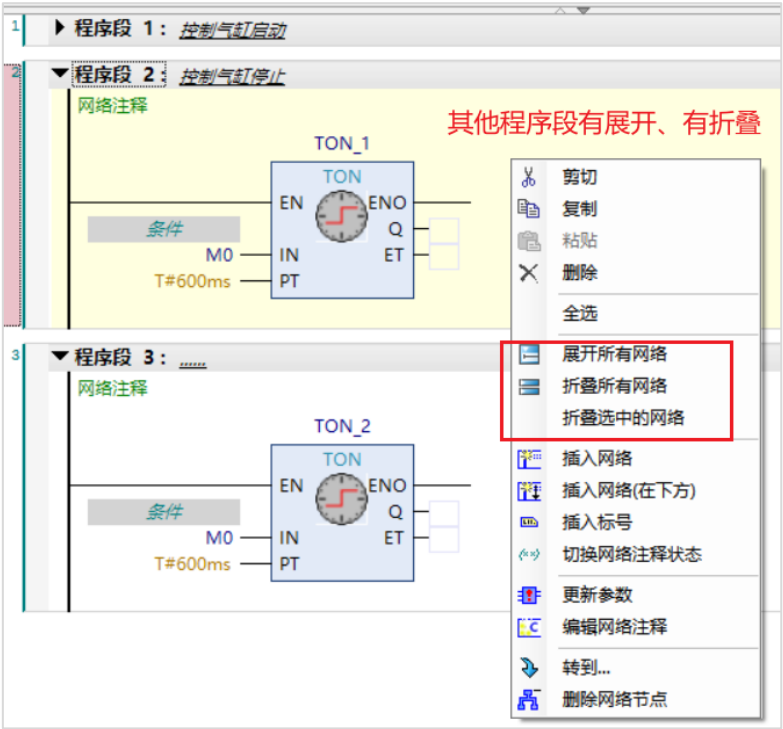
所有程序段状态	右键快捷菜单
全部展开	只有“折叠所有网络”
全部折叠	只有“展开所有网络”
有展开、有折叠	“展开所有网络” “折叠所有网络”



- 选择程序段时（当前选择的程序段是展开的），右键快捷菜单栏的“展开/折叠程序段”情况如下：

其他程序段状态	右键快捷菜单
全部展开	“折叠所有网络” “折叠选中的网络”
全部折叠 有展开、有折叠	“展开所有网络” “折叠所有网络” “折叠选中的网络”



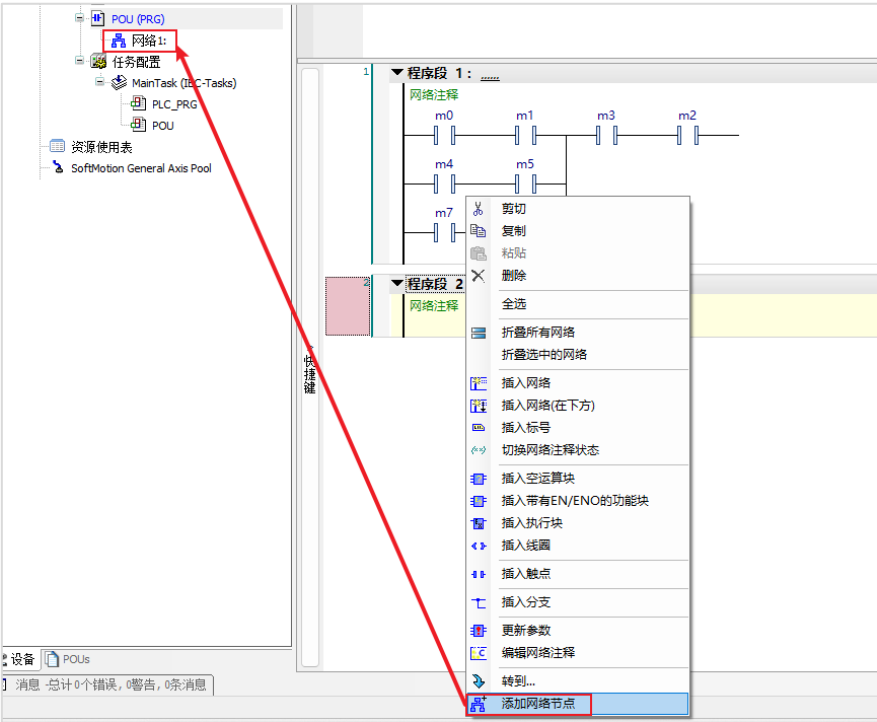


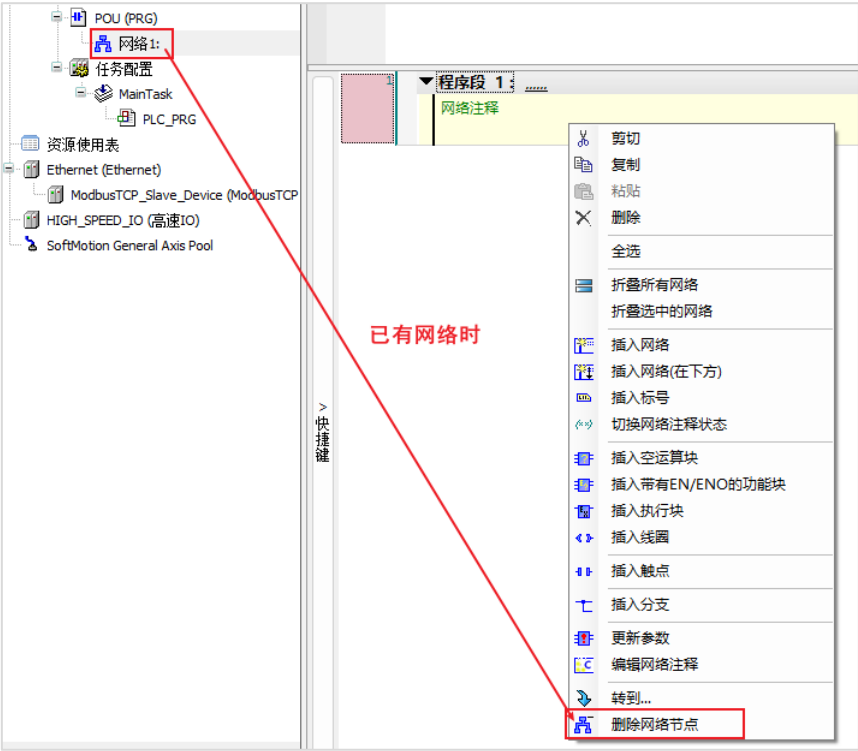
3、网络列表

1、添加/删除网络节点

选择网络后，右键菜单可选【添加网络节点】和【删除网络节点】两个功能，可将梯形图中网络节点添加至左侧工程树中，双击工程树节点可以定位到相应的网络。

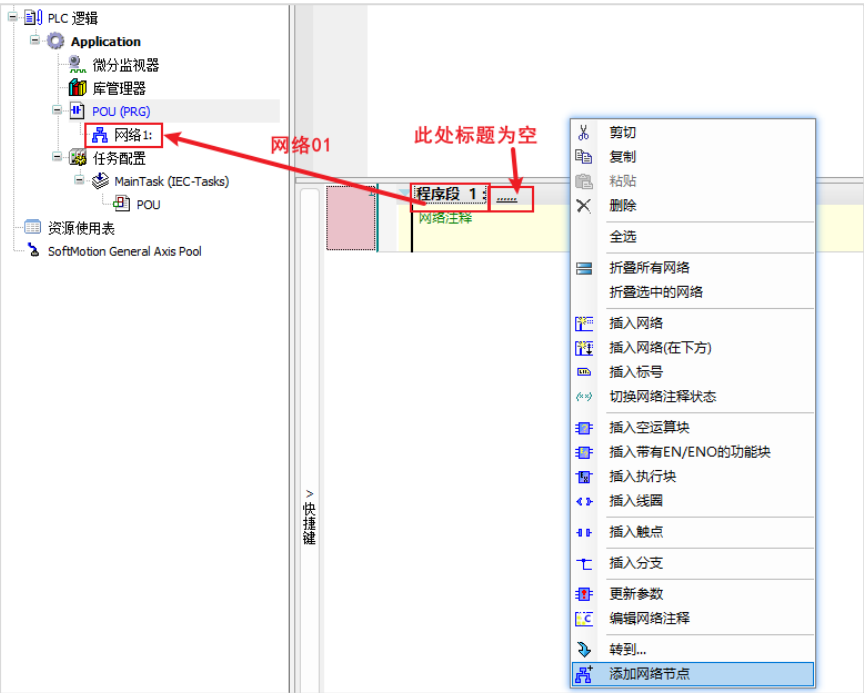
- 若选中网络中，都未添加网络节点，则右键菜单仅显示【添加网络节点】。
- 若选中网络中，都已添加网络节点，则右键菜单仅显示【删除网络节点】。
- 若选中网络中，包含已添加节点的网络和未添加节点的网络，则右键菜单显示【添加网络节点】【删除网络节点】两个功能。选择【添加网络节点】，仅操作未添加节点的网络；选择【删除网络节点】，仅操作已添加节点的网络。

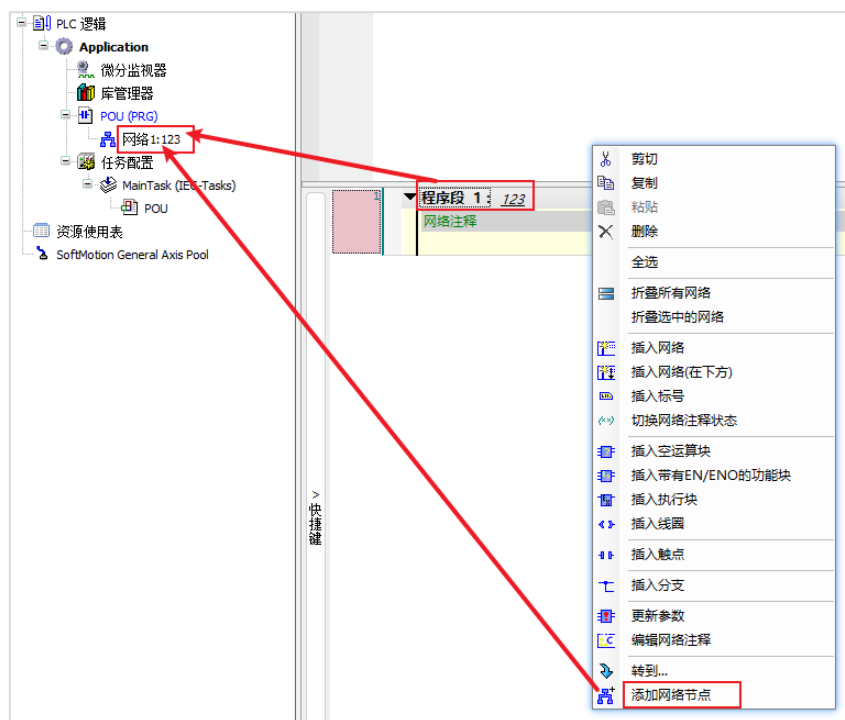




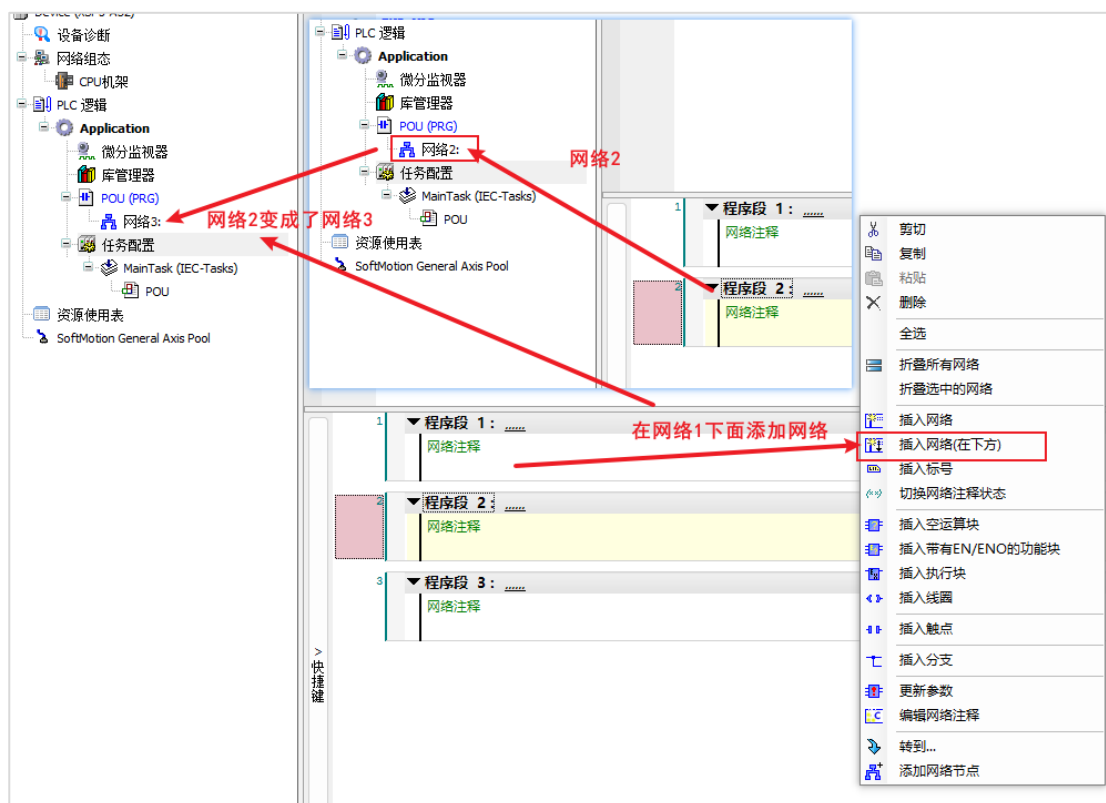
2、网络节点命名规则

【添加网络节点】时的节点命名规则为：网络序号：网络标题。





若已添加网络节点的网络序号或标题发生变化时，网络节点的名字也随之变化。



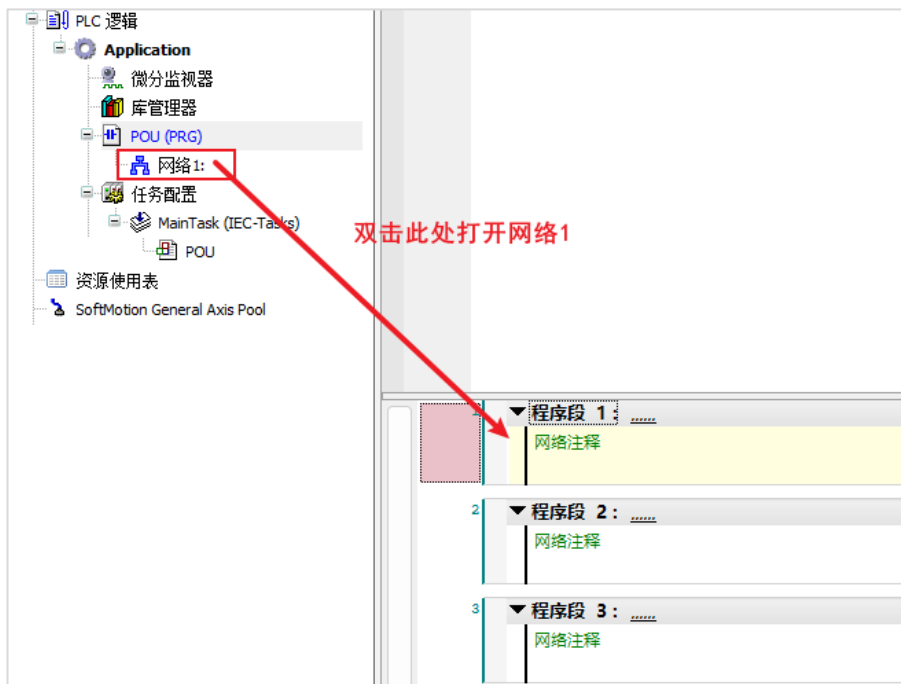
3、多选添加网络节点

支持多选添加网络节点。在工程树网络节点中的位置根据网络序号进行升序排序排列。



4、定位网络

鼠标双击工程树网络节点，界面跳转定位网络。若程序未打开，则自动打开相应程序，跳转定位网络。



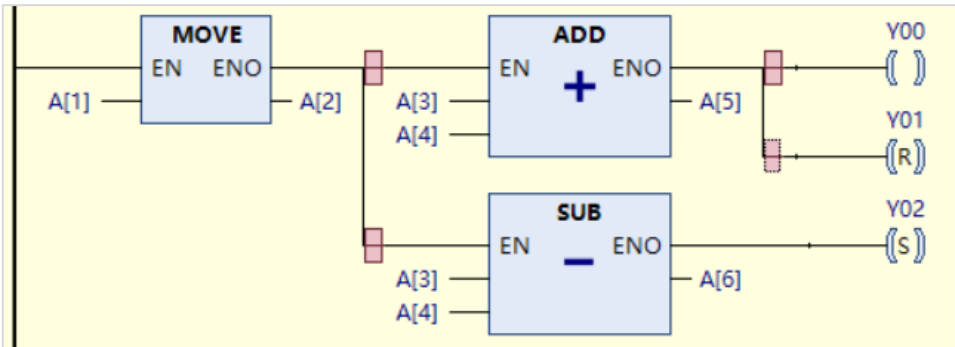
5、删除节点网络

选择已添加网络节点网络，右键仅显示【删除网络节点】，单击此删除功能，移除工程树网络节点。



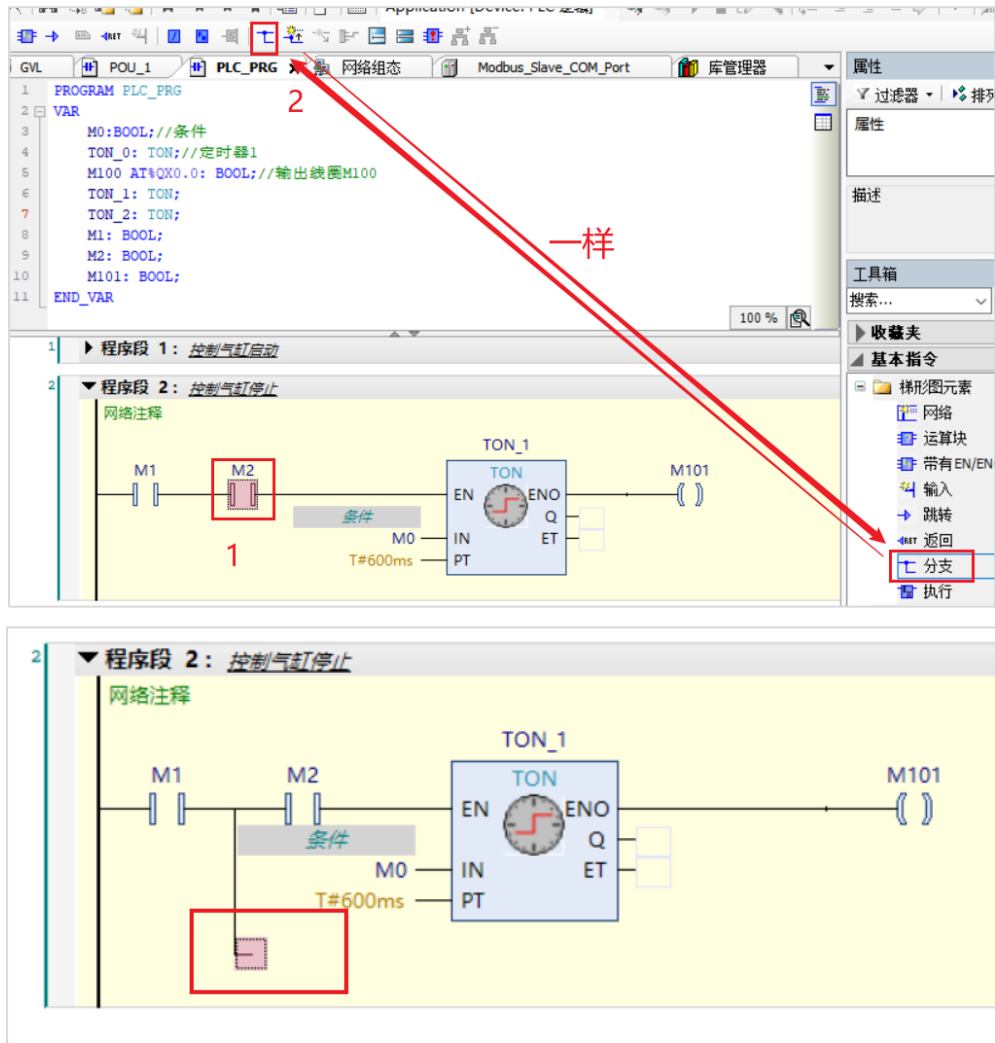
7-3-7. 分支功能

下图中，每个选择位置都表示一个分支。

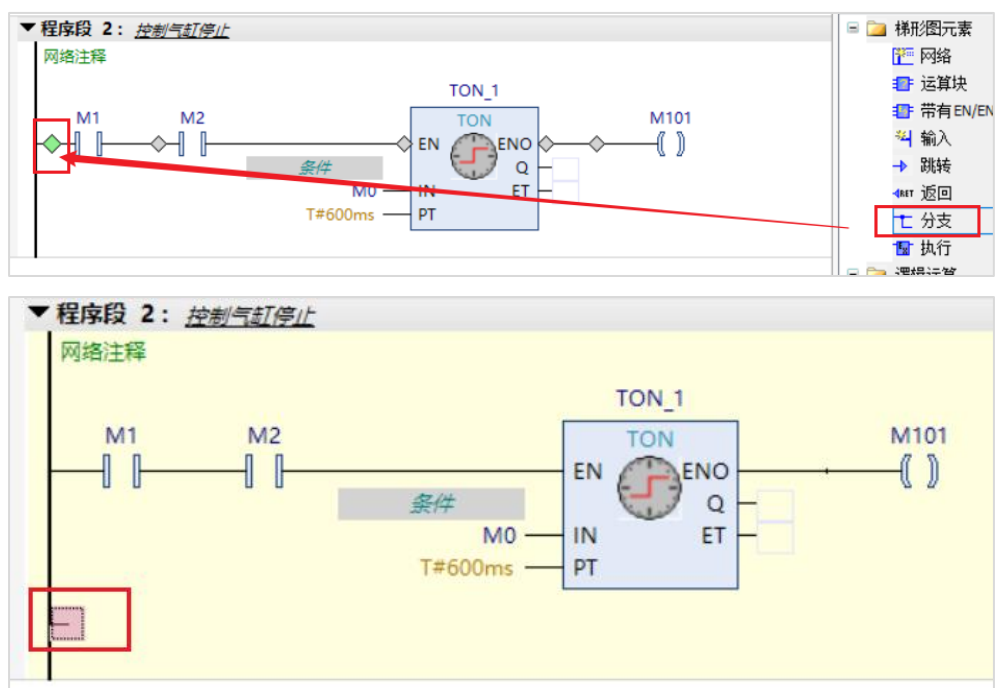


1、插入分支

方法一：选中对象，点击插入分支：



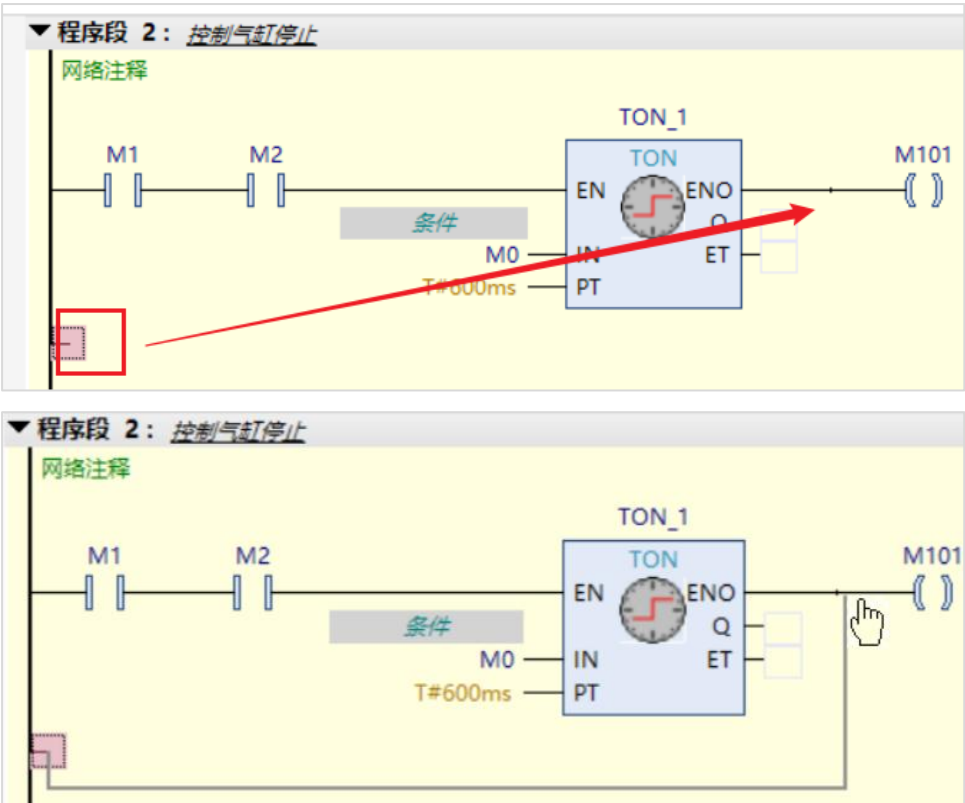
方法二：长按住右侧工具箱里的“插入分支”，进行拖拽：



还有“在上面插入分支”、“设置分支起点”功能都可以操作分支，可自行点击对梯形图进行编辑。

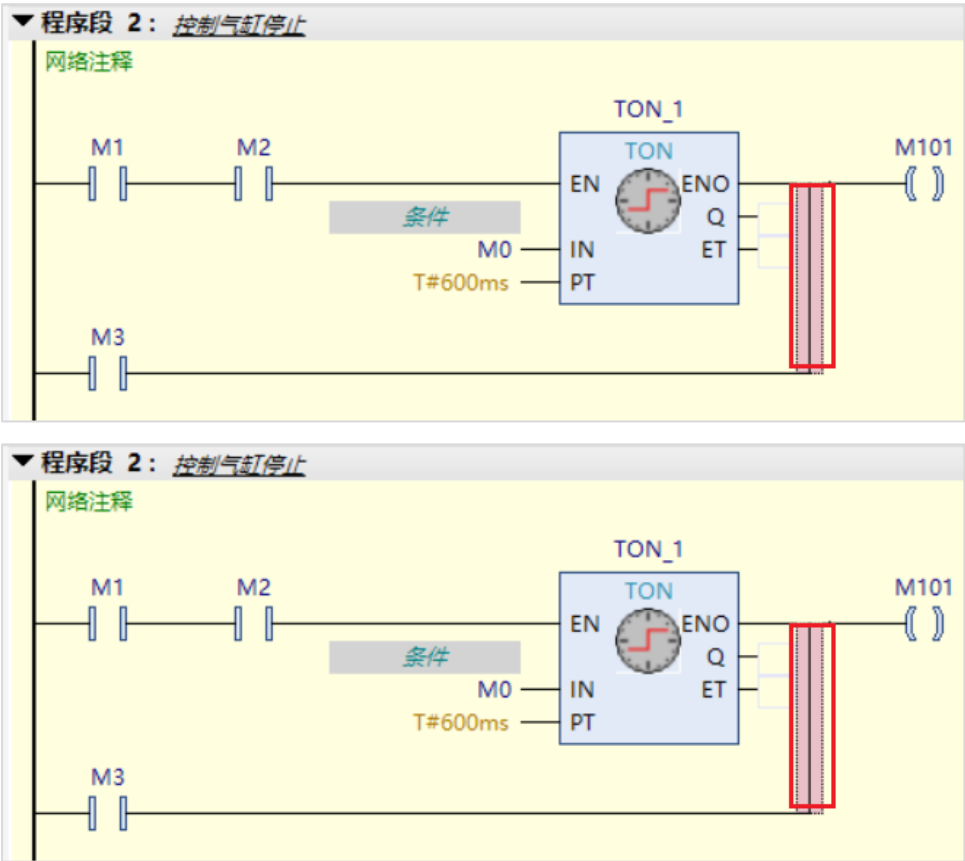
2、合并分支

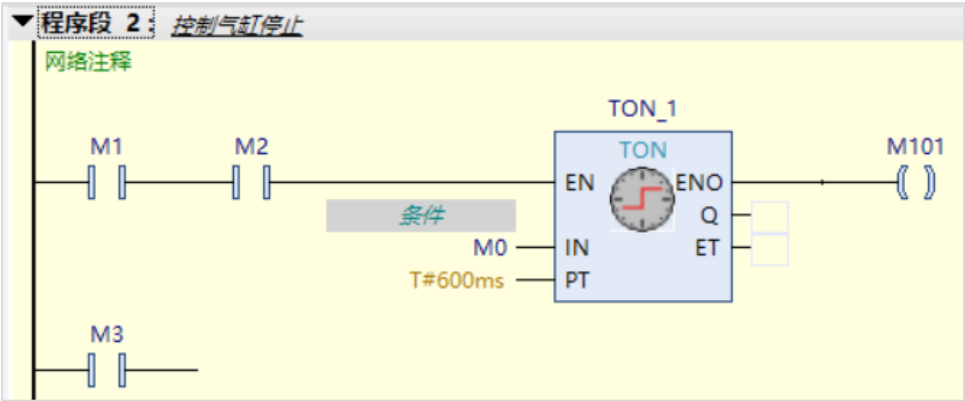
可以鼠标绘线。允许鼠标绘线到此处时，鼠标会显示手势图标。



3、打开分支

通过选中并按键盘 Delete 键实现打开分支的功能：

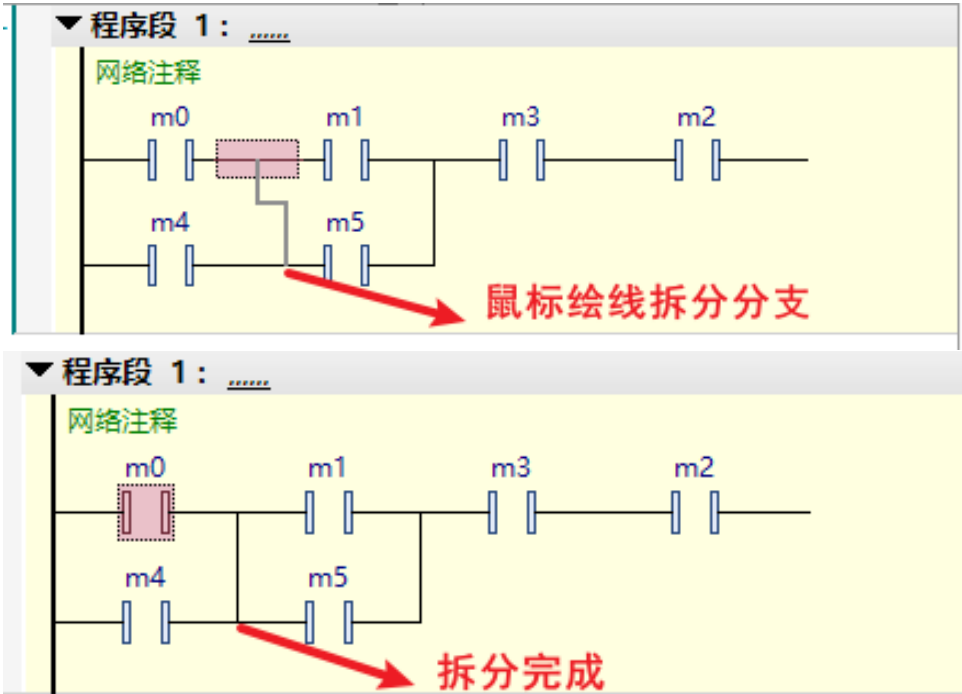




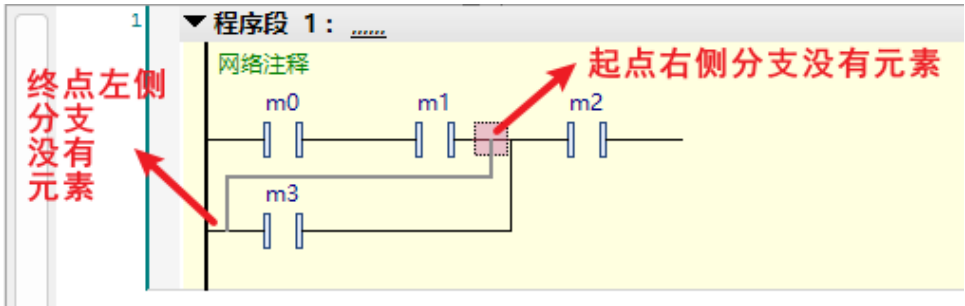
4、拆分分支

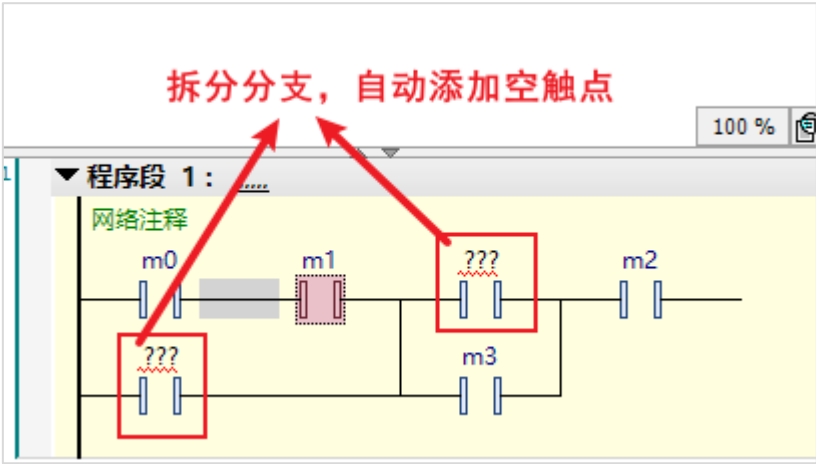
支持在两个相邻的分支中间绘制垂线以拆分分支。

- 当划线起点和终点为并联两相邻分支时，划线时会把起点、终点连线两侧拆分为两个并联。

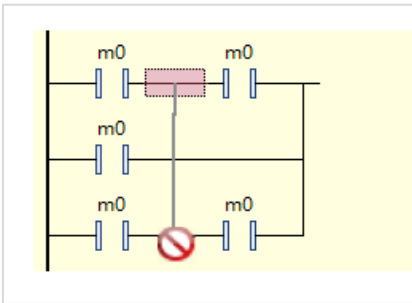


- 当起点或者终点左侧或者右侧没有元素，自动在没有元素的一侧增加空触点。



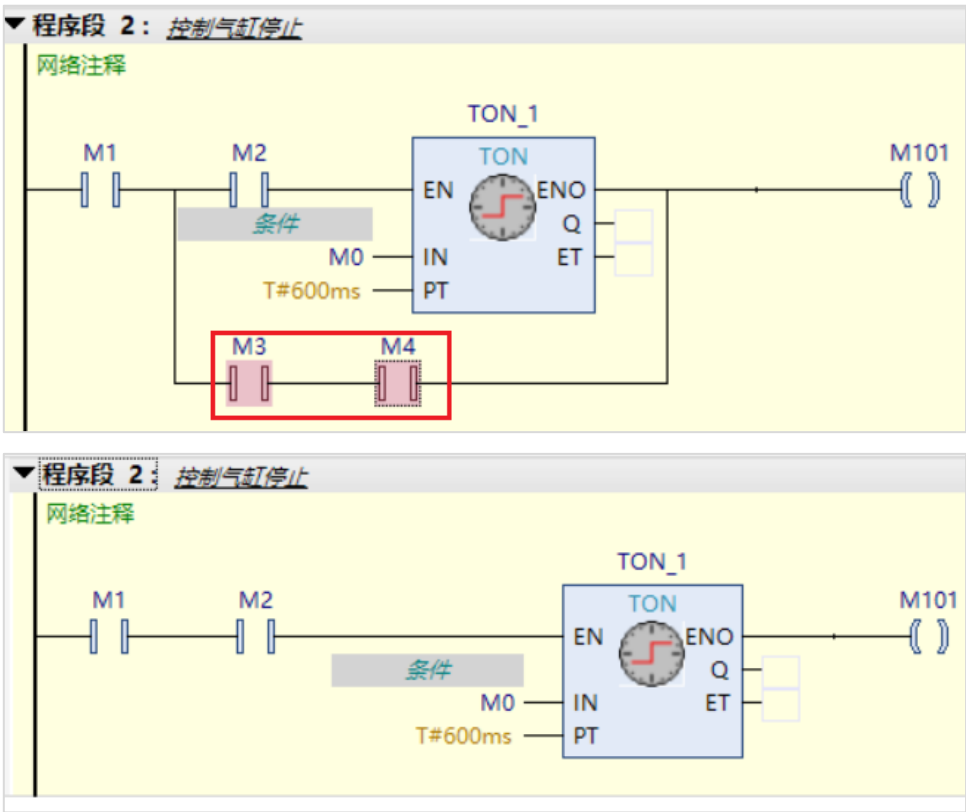


- 拆分分支不支持跨越打开分支。



5、删除分支

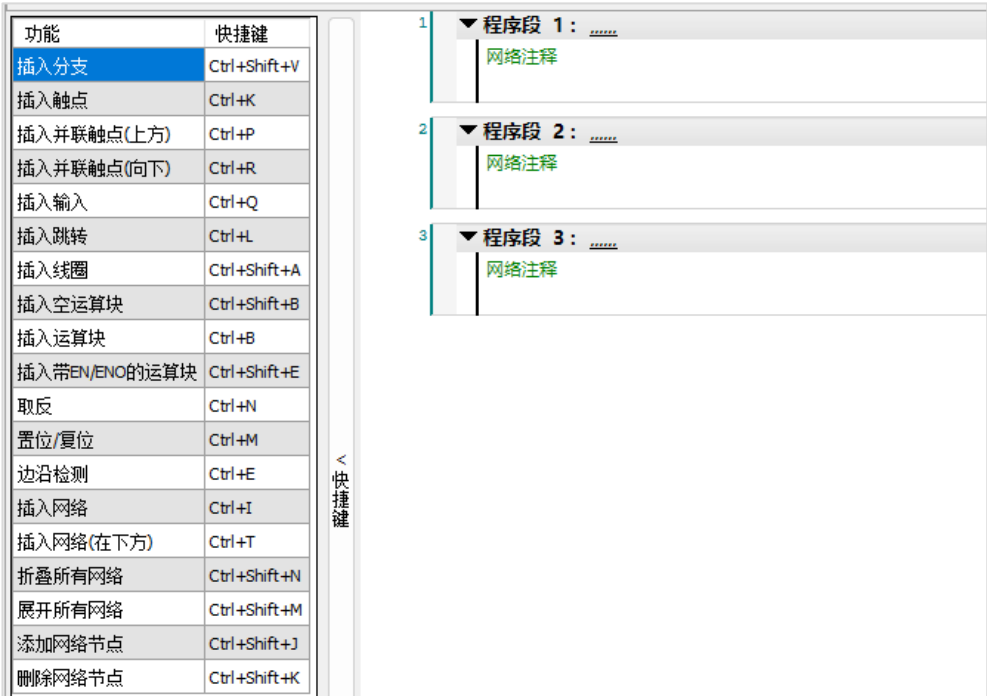
选中分支上的所有节点并删除，即可删除当前分支：



7-3-8. 快捷键提示栏

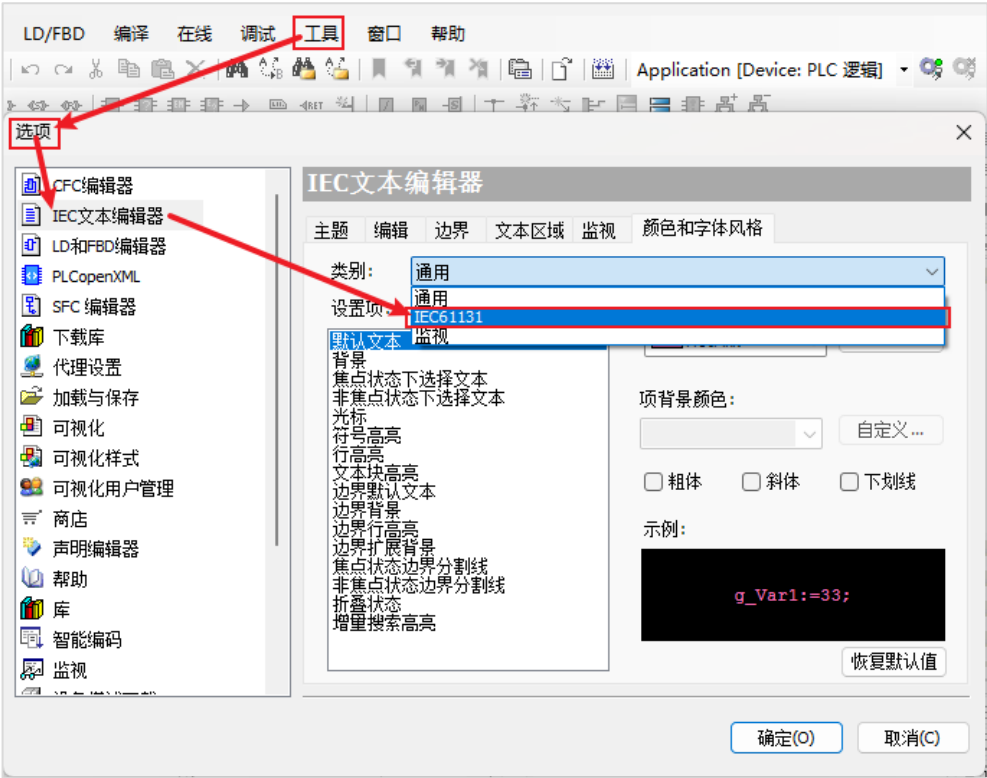
梯形图编辑器左侧显示编辑梯形图程序的快捷键列表，默认隐藏。可通过以下两种方式显示、隐藏列表：

- 在梯形图编辑器中触发快捷键“H”；
- 在梯形图编辑器中鼠标单击快捷键列表按钮。



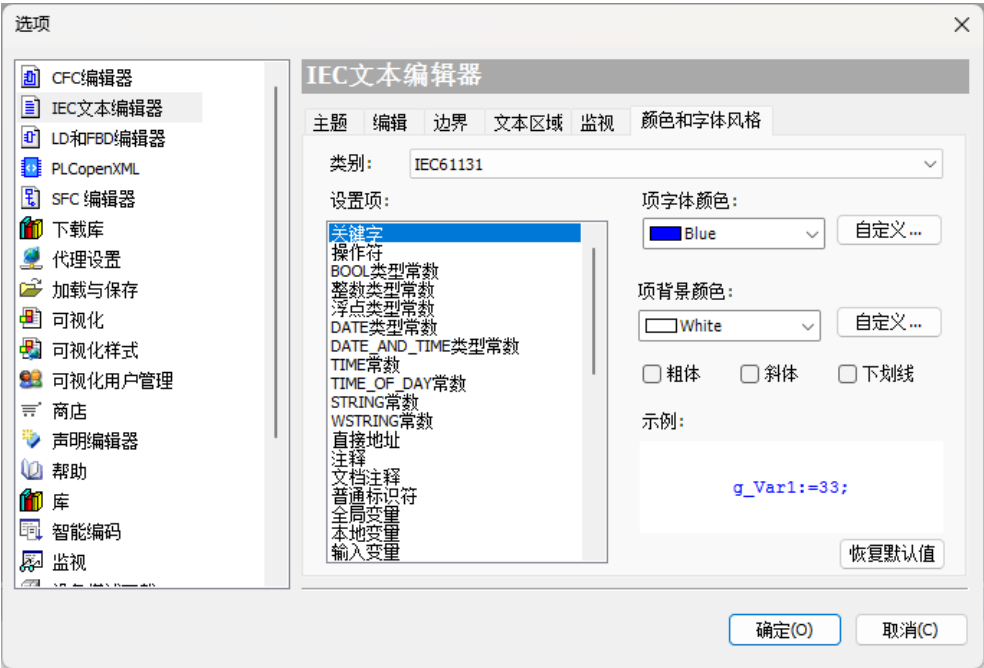
7-3-9. 编辑器字体和颜色设置

支持设置梯形图编辑器数据颜色，在“工具”——“选项”——“IEC 文本编辑器”选择“ISC61131”中更改。



支持数据如下：

常数类型：



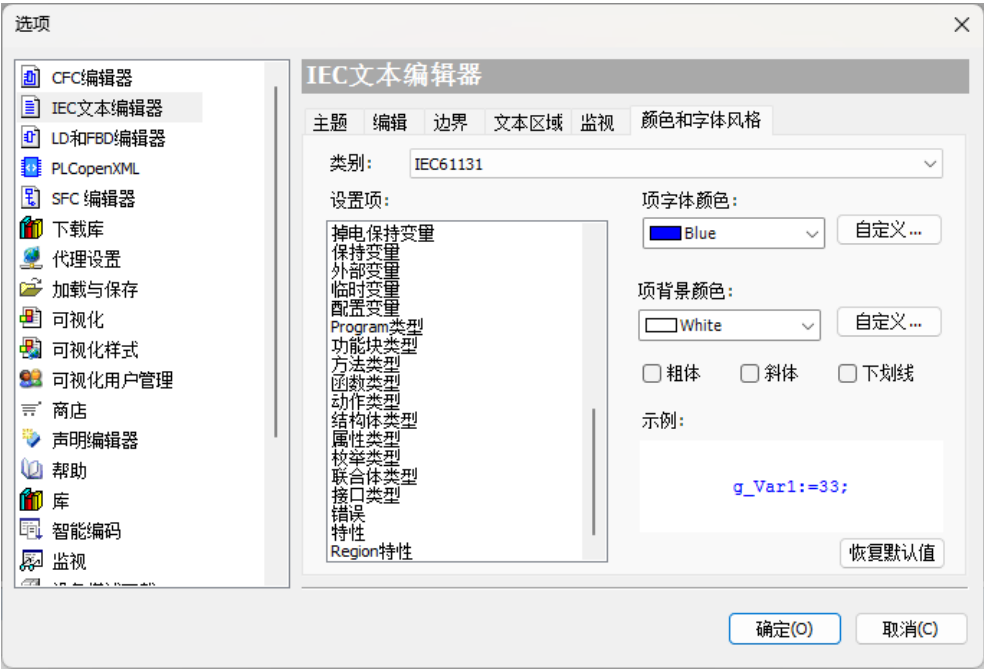
直接地址

注释

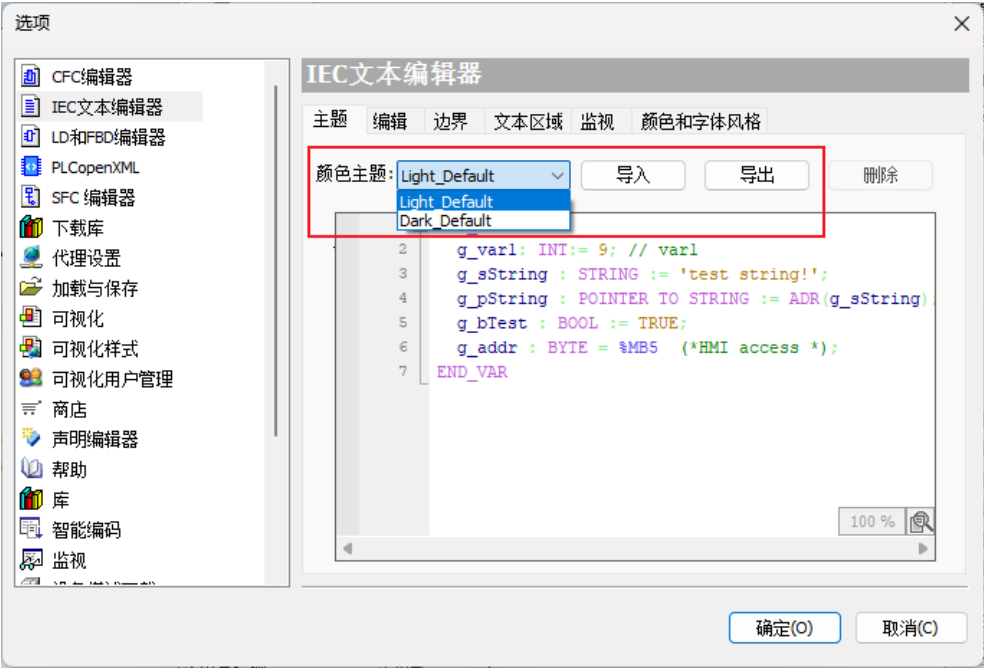
变量类型



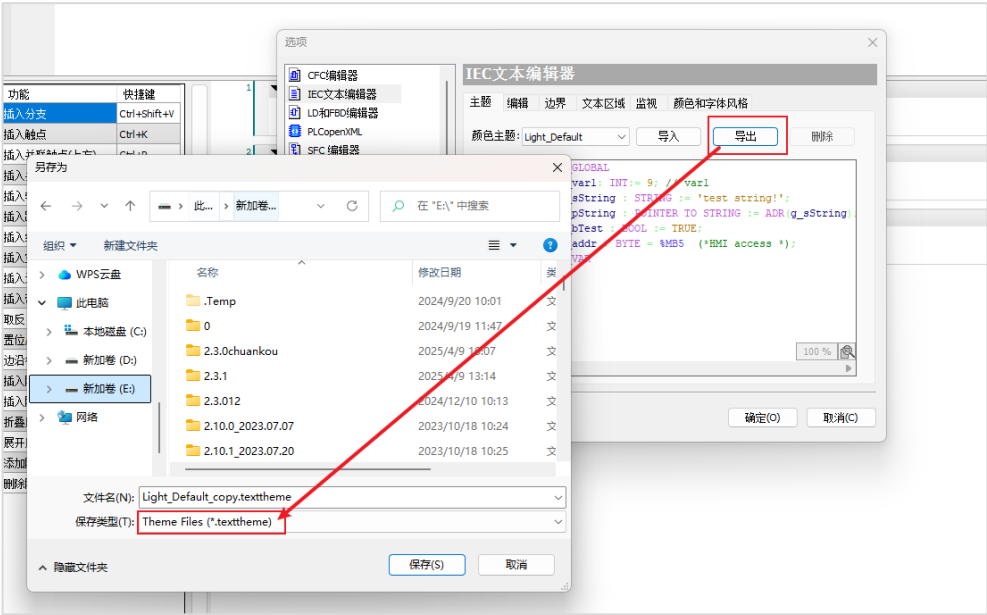
其他类型



在“IEC 文本编辑器”中支持导入，导出，删除功能。



- 主题：可选“Light_Default”和“Dark_Default”两种主题形式。
- 导出：可将设置的主题导出。导出文件名称默认为：主题名称_copy，导出文件名称对应主题名称，可根据需求修改。文件格式不做限制。



- 导入：将选择主题文件导入，导入成功新增主题。若导入文件名称与窗口已有主题名称重复，则提示：主题已存在，是否覆盖。可根据需求选择是否覆盖主题。
- 删除：选择主题删除。

7-3-10. 单键操作

单键命令通过单个字符快捷键进行快速编辑。可以在连线上执行单键命令，也可以在元素上执行单键命令。

连线上执行单键插入串行元素；元素上单键命令用于插入并行元素或者元素功能切换。

每个命令对应的字符，可以在“工具-选项 - LD 和 FBD 编辑器-LD 页面设置，默认设置参数如下。

单键设置

连线/空网络上执行的单键

插入触点:

C

插入常闭触点:

/

插入线圈:

Q

插入复位线圈:

R

插入置位线圈:

S

插入空运算块:

F

插入空 EnEno 运算块:

E

Set/Rset/沿信号切换:

空格

添加网络节点(网络):

J

删除网络节点(网络):

K

元素上执行的单键

插入并行触点:

C

插入并行空运算块:

F

插入并行空 EnEno 运算块:

E

插入线圈:

Q

元素取反切换:

/

元素 Set/Rset/沿信号切换:

空格

其他

快捷键显示/隐藏:

H

- 单键支持设置 26 个字母、空格和 “/” 格式；
- 互斥原则：

连线/空网络中的单键互斥；

元素上执行的单键互斥；

其他中的快捷键和连线/空网络、元素上的互斥。

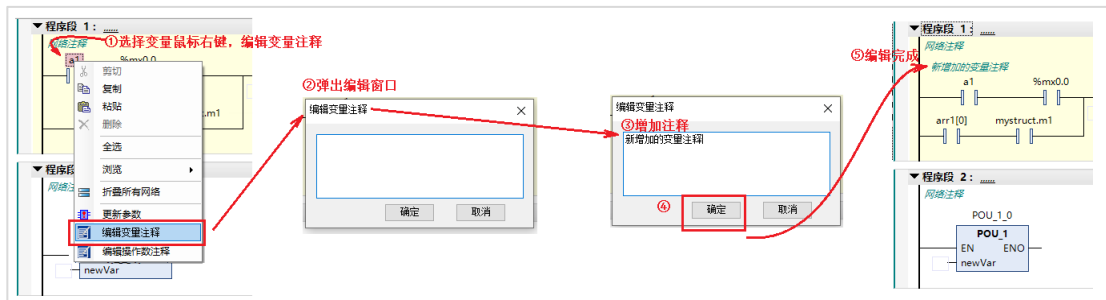
7-3-11. 变量注释

支持在梯形图编辑器中修改变量、实例的注释。

1、右键菜单

选中梯形图编辑器中变量或实例，右键选择【编辑变量注释】，在提示的变量注释编辑窗口中进入编辑界面。编辑完成后，确认并关闭编辑窗口后编辑内容生效。

● 编辑注释



● 编辑复杂类型变量注释



● 编辑复杂类型变量成员注释

若编辑复杂类型成员注释时，直接编辑其成员注释。



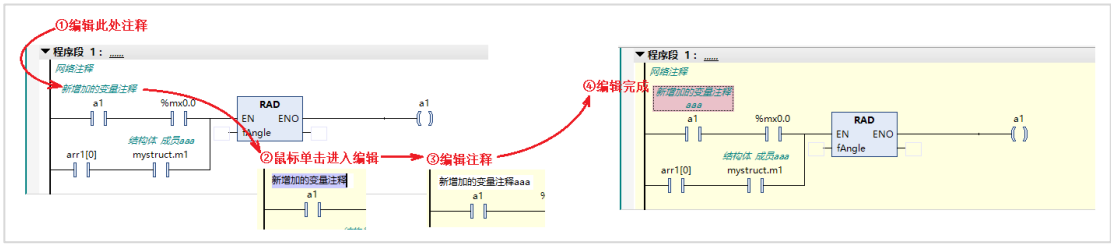
2、直接编辑

选择梯形图中变量注释，支持以下两种方式直接编辑：

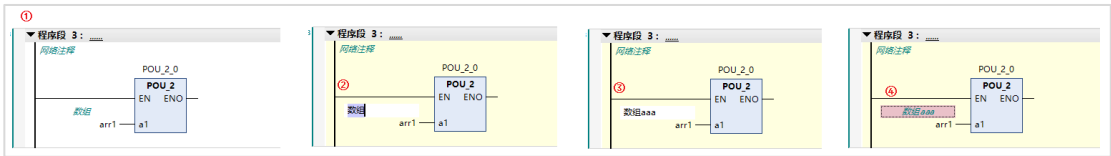
- ① 鼠标单击进入编辑
- ② 键盘选中，enter 后进入编辑。

注释修改完成后，触发 enter 或鼠标点击其他区域，编辑完成，修改内容生效。若在编辑过程中触发 ESC，则退出编辑，注释不做变化。

● 编辑注释

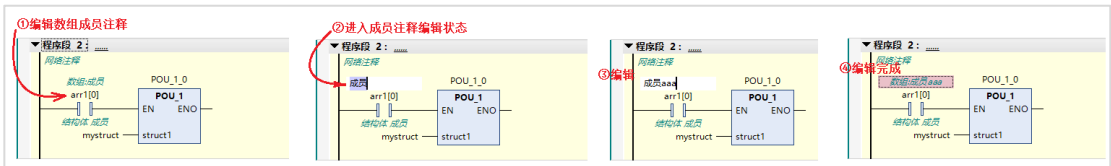


● 编辑复杂类型变量注释



● 编辑复杂类型变量成员注释

若编辑复杂类型成员注释时，直接编辑其成员注释。



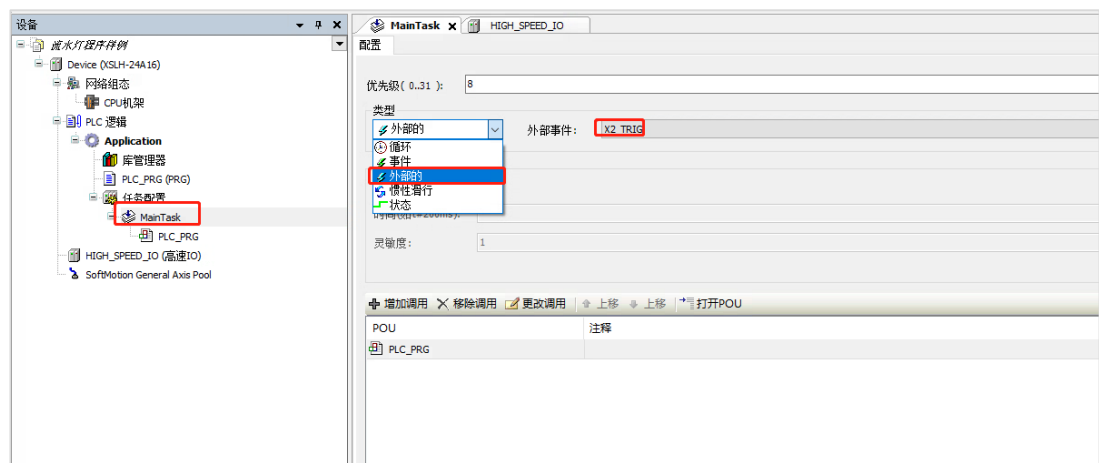
8. 特殊功能

8-1. 外部中断

1、XSDH 和 XSLH-30A32（3 款 XS 机型固件 2.4.0 以下）：

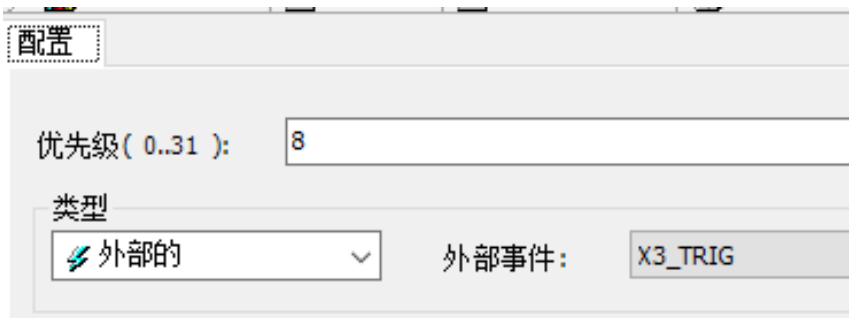
支持 X 端子中断，同一个端子支持上升沿和下降沿中断，在 XS Studio 中通过任务类型中的外部事件形式来使用中断。如 X2R_TRIG 代表 X2 上升沿中断，X2F_TRIG 代表下降沿中断，各机型支持的中断数量和类型见外部事件“外部的”选项。

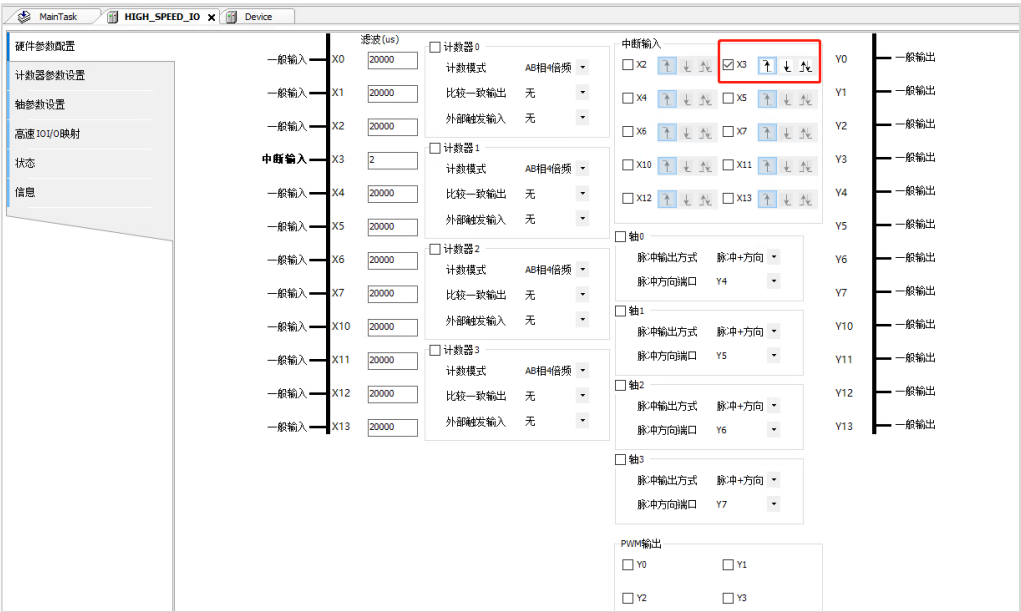
双击“MainTask”，在弹出界面中设置为外部事件“外部的”-外部中断使用端子 X，也可设置外部中断事件的优先级。



2、XSLH-24、XSA 及其他 XS 机型（除了 XSLH-24 和 XSA 外，固件要求 2.4.0 及以上）：

需要使用【XJ_Interrupt】和【XJ_WriteInterruptParameter】指令和界面（功能详见指令篇手册）。设置 X3 为外部中断输入，取其双边沿信号，可以在硬件参数配置界面配置，也可以使用 XJ_WriteInterruptParameter 指令配置，给一次 X3 的边沿信号，就执行一次另一个 task（配置为外部的、X3_TRIG）下的 POU 程序里的自加 1 指令，参数配置和指令如下图：





Device.Application.PLC_PRG						
表达式	类型	值	准备值	地址	注释	
XJ_EnableInterrupt_0	XJ_EnableInterrupt	TRUE			使能	
xEnable	BOOL	8			打开外部输入中断，例如	
udiExternal	UINT	0			打开比较一致中断，例如	
uiCompare	UINT	0				
xValid	BOOL	TRUE			中断生效	
xBusy	BOOL	FALSE			正在运行	
xError	BOOL	FALSE			错误标志	
eErrorID	HSIO_ERROR	ERR_OK			错误代码	
XJ_WriteInterruptParameter_0	XJ_WriteInterruptParameter					
Port	UINT	8			端口号，例如（外部输入中	
xExcute	BOOL	TRUE			触发	
byValue	BYTE	2			值(0: 上升沿, 1 为下降沿,	
xDone	BOOL	TRUE			完成标志	
xBusy	BOOL	FALSE			正在运行	
xError	BOOL	FALSE			错误代码	
eErrorID	HSIO_ERROR	ERR_OK			错误标志	

XJ_EnableInterrupt_0

XJ_EnableInterrupt

TRUE

8

0

uiCompare

xValid

FALSE

FALSE

ERR_OK

XJ_WriteInterruptParameter_0

Port

xExcute

byValue

xDone

FALSE

FALSE

FALSE

ERR_OK

8-2. 高速计数

1、XSDH 和 XSLH-30A32（3 款 XS 机型固件 2.4.0 以下）：

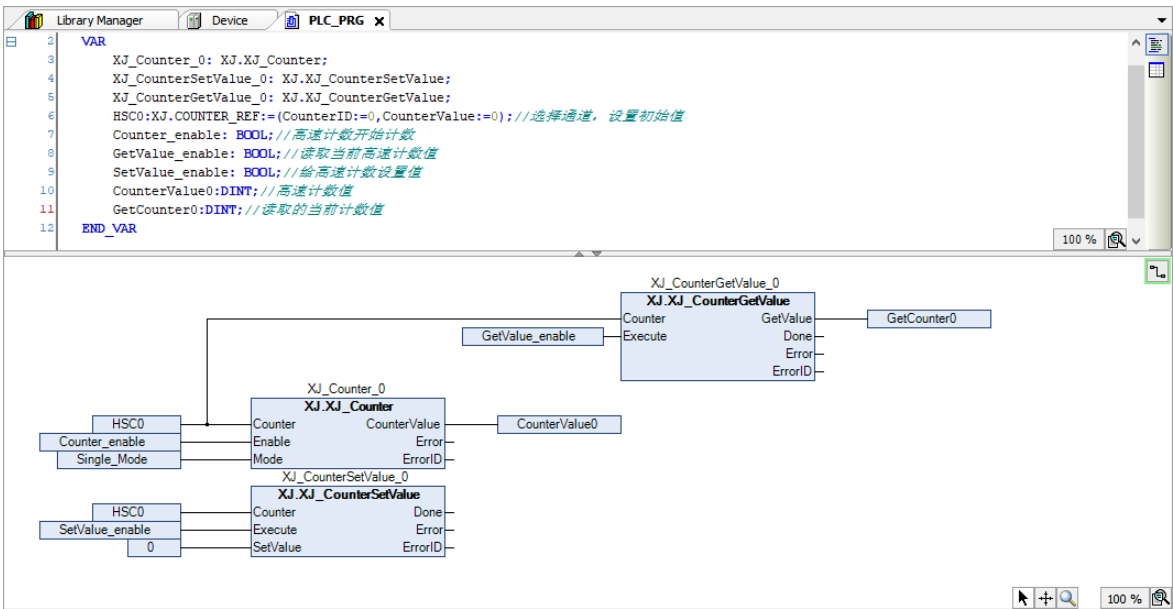
 **示例：**使用第一路高速计数，并可在计数中读取当前计数值，修改当前高速计数值。

程序操作：

- （1）按照 2-1-3 节中的步骤安装需要使用的库；
- （2）编写高速计数程序。

编写程序：

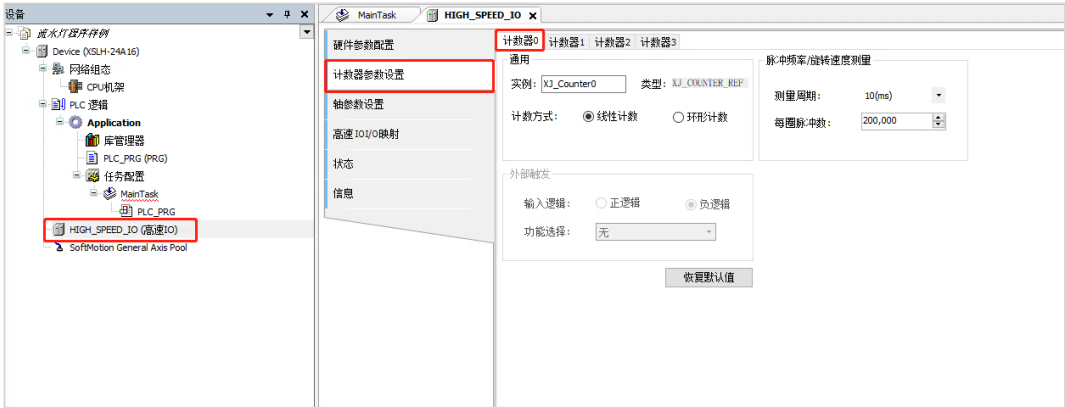
使用功能块“XJ.XJ_Counter”、“XJ.XJ_CounterGetValue”、“XJ.XJ_CounterSetValue”，可在程序中设置使用的高速计数端口、高速计数模式、设定的高速计数值。



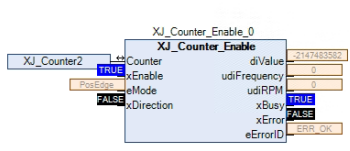
2、XSLH-24、XSA 及其他 XS 机型（除了 XSLH-24 和 XSA 外，固件要求 2.4.0 及以上）：

 示例：使用【XJ_Counter_Enable】指令，测量外部的高速信号输入，配置如下图：





表达式	类型	值	准备值	地址	注释
XJ_Counter_Enable_0	XJ_Counter_Enable				
Counter	REFERENCE TO XJ_COUNTER...				数据类型XJ_COUNTER_REF
xEnable	BOOL	TRUE			使能
eMode	HSC_EDGE_MODE	PosEdge			PosEdge: 上升沿计数; NegEdge: 下降沿计数; BothEdge: 双边沿计数
xDirection	BOOL	FALSE			false:加计数; true: 减计数;
diValue	DINT	-2147483582			高速计数值
udiFrequency	UDINT	0			脉冲频率测量值 (单位: Hz),若为低频可通过界面测量周期配合使用
udiRPM	UDINT	0			每分钟旋转速度 (单位: r/min), 与界面配置中的每圈脉冲数配合使用
xBusy	BOOL	TRUE			忙碌中
xError	BOOL	FALSE			错误标志
eErrorID	HSIO_ERROR	ERR_OK			错误代码



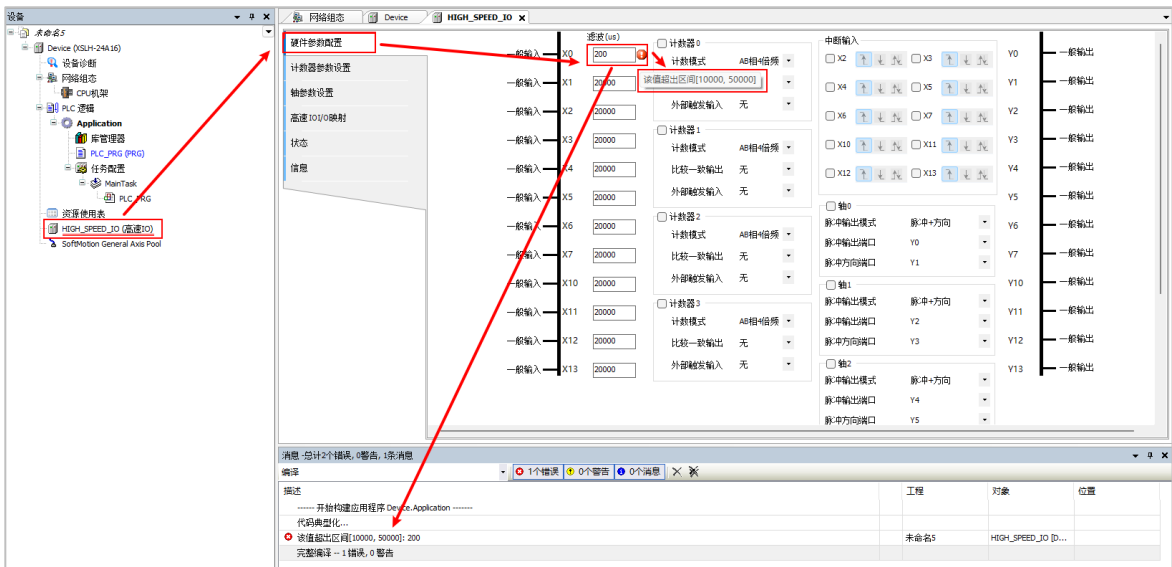
8-3. 高速 IO 配置

双击设备树中的 HIGH-SPEED-IO（高速 IO）选项，可打开高速 IO 的硬件参数配置界面。在该界面可对高速脉冲输出功能及 PWM 输出功能进行配置(XS Studio 1.1.0 及以上版本仅 XSLH-24A16、XSLH-24A8 支持)。默认参数配置界面如下图所示：

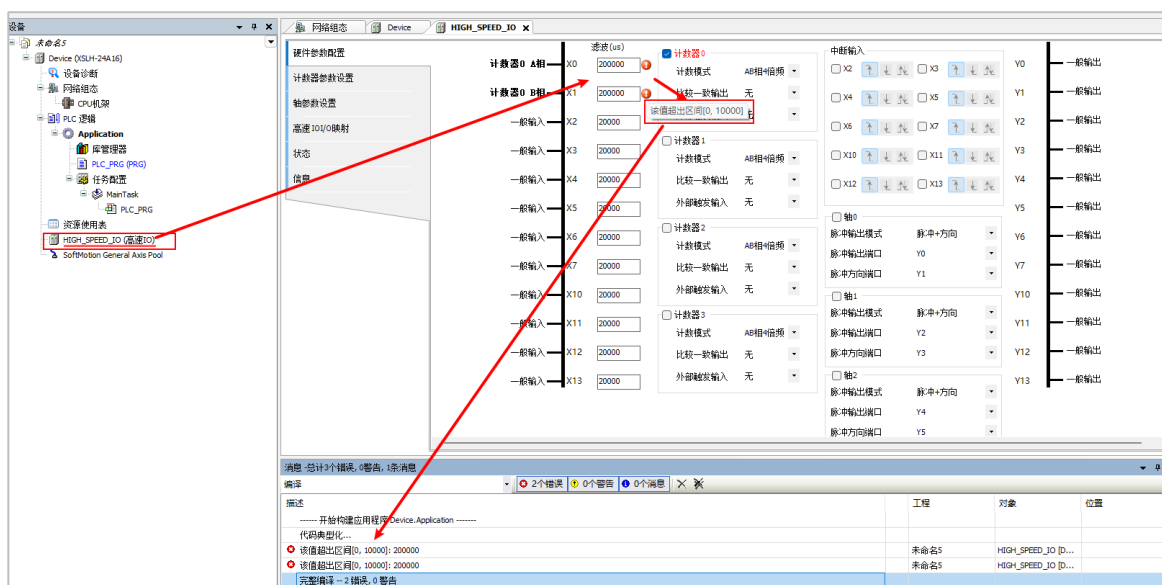


■ 滤波

一般输入时：单位 us，默认值 20000us，范围 10000us-50000us，异常输入时做叹号提示，编译后在信息栏展示错误信息，如图所示：

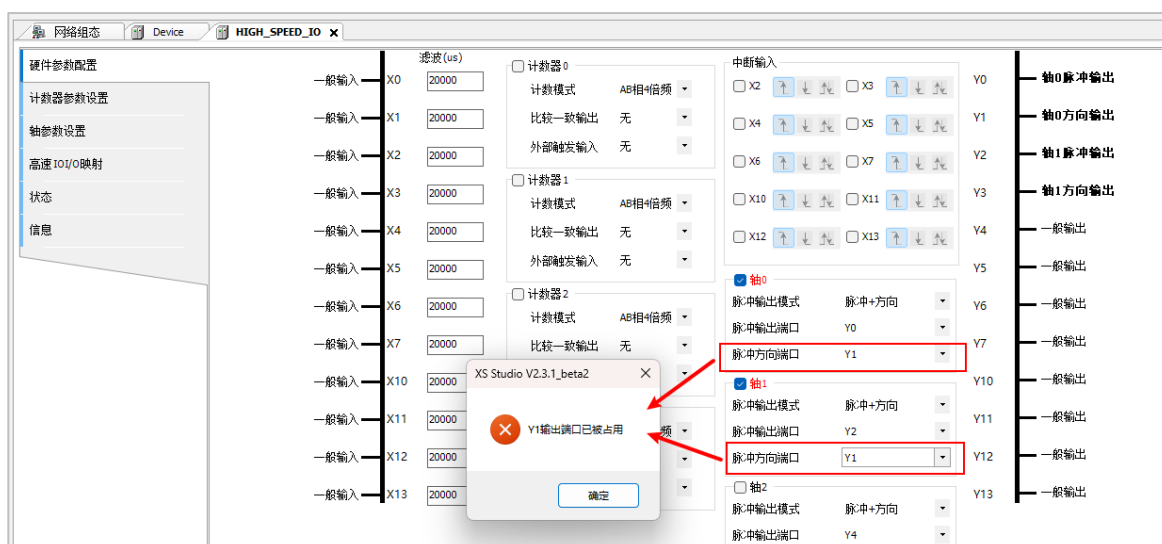


AB 相/中断输入时：单位 us，默认值 2us，范围 0us-10000us，异常输入时做叹号提示，编译后在信息栏展示错误信息，如图所示：



■ X/Y 端口

端口冲突时，配置提示用户端口占用，确定后返回上一次的配置。



8-3-1. 高速脉冲输出功能

1、脉冲指令

指令名	功能说明
MC_Power	使轴开始可运行状态
MC_Reset	复位轴内部相关错误
MC_Jog	点动
MC_Stop	停止控制器运动
MC_MoveAbsolute	实现一个控制轴达到指定绝对运动位置
MC_MoveRelative	从当前轴的位置将轴移动一个相对位置
MC_MoveVelovity	轴以一个指定速度持续运行下去

指令名	功能说明
MC_SetPosition	设置轴的当前位置
MC_ReadStatus	读取轴状态
MC_ReadSetPosition	读取当前轴的设置位置
MC_ReadActualPosition	读取当前相关轴的当前位置
XMC_ZRN	脉冲回原

以上指令中，XMC_ZRN 为 XJ_HSIO 库内的指令（该指令下位机 V2.2.0 以上固件支持），其余均为 SM3_Basic 库内的指令。具体指令说明见《XS 系列 PLCopen 标准控制器用户手册【运动控制篇】》。

2、配置高速脉冲输出功能

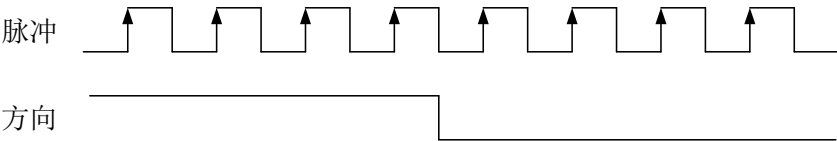
- 脉冲输出方式：下拉菜单可选（脉冲+方向、AB 相、CW/CCW）
默认参数：脉冲+方向
- 脉冲输出端口（以轴 0 配置为例，轴号支持 0-3），以及脉冲方向端口（Y0-Y13 可选）。
选择不同的输出端口可选参数见下表：

轴名称	脉冲输出方式选择	脉冲输出端口 默认参数/可设参数	脉冲方向端口 默认参数/可设参数
轴 0	脉冲+方向	Y0	Y1/Y1-Y13、无方向
	AB 相	Y0	Y1/无法选择改变此参数
	CW/CCW	Y0	Y1/无法选择改变此参数
轴 1	脉冲+方向	Y2	Y3/Y0、Y1、Y3-Y13、无方向
	AB 相	Y2	Y3/无法选择改变此参数
	CW/CCW	Y2	Y3/无法选择改变此参数
轴 2	脉冲+方向	Y4	Y5/Y0-Y3、Y5-Y13、无方向
	AB 相	Y4	Y5/无法选择改变此参数
	CW/CCW	Y4	Y5/无法选择改变此参数
轴 3	脉冲+方向	Y6	Y7/Y0-Y5、Y7-Y13、无方向
	AB 相	Y6	Y7/无法选择改变此参数
	CW/CCW	Y6	Y7/无法选择改变此参数

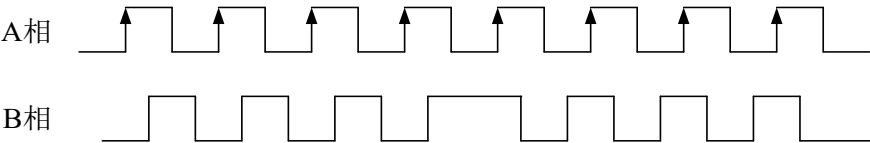
- 勾选使用的轴 0；勾选后配置情况如图所示：



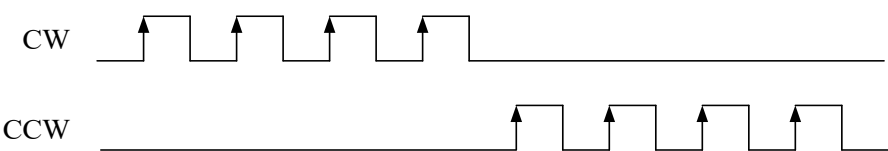
- 配置高速脉冲输出的工作方式：
 - (1) 脉冲+方向
- “脉冲输出方式”选择“脉冲+方向”，输出脉冲+方向信号，占用两个硬件输出端口。



- (2) AB 相
- “脉冲输出方式”选择“A/B 相”，输出 AB 相信号，占用两个硬件输出端口。



- (3) CW/CCW
- “脉冲输出方式”选择“CW/CCW”，输出 CW/CCW 信号，占用两个硬件输出端口。



3、脉冲轴参数配置

在轴参数设置界面可对已配置的轴进行实例化等操作。以轴 0 为例，配置界面如下图所示：

- 高速 IO 版本：1.0.1.0 对应界面如下：

硬件参数配置

计数器参数设置

轴参数设置

高速 IO/O 映射

状态

信息

轴0轴1轴2轴3

轴名称

实例：

类型：

软件限位

☐ 模数

☒ 有限

☐ 激活

正限位 (pulse):

负限位 (pulse):

电机类型

脉冲端口

☒ 旋转

☐ 线性

比例缩放

☐ 反转方向

增量 <=> 电机转

电机转动 <=> 齿轮输出转

减速电机输出转 <=> 应用的单位

动态限制

速度 [u/s]:

加速度 [u/s²]:

减速度 [u/s²]:

加加速度 [u/s³]:

软件错误反应

减速度 [u/s²]:

最大距离 [u]:

速度斜坡类型

☒ 梯形

☐ sin²

☐ 二次方

☐ 二次 (平滑)

恢复默认值

- 高速 IO 版本：1.1.0.0 对应界面如下（新增正负极限及脉冲回原配置参数）：

Device

PLC_PRG

GVL

HIGH_SPEED_IO

硬件参数配置

计数器参数设置

轴参数设置

高速 IO/O 映射

状态

信息

轴0轴1轴2轴3

轴名称

实例：

类型：

软件限位

☐ 模数

☒ 有限

☐ 激活

正限位 [u]:

负限位 [u]:

原点正负限位信号选择

原点端口

未分配

正极限端口

未分配

负极限端口

未分配

Z相端口

未分配

原点端口极性

极性不反转

负极限端口极性

极性不反转

正极限端口极性

极性不反转

Z相端口极性

极性不反转

Z相个数

电机类型

☒ 旋转

☐ 线性

比例缩放

☐ 反转方向

增量 <=> 电机转

电机转动 <=> 齿轮输出转

减速电机输出转 <=> 应用的单位

回零参数

回零速度 [u/s]:

爬行速度 [u/s]:

回零加速度 [u/s²]:

回零减速度 [u/s²]:

回零加加速度 [u/s³]:

回零方向: ☒ 正向 ☐ 负向

零点位置: u

软件错误反应

减速度 [u/s²]:

最大距离 [u]:

动态限制

速度 [u/s]:

加速度 [u/s²]:

减速度 [u/s²]:

加加速度 [u/s³]:

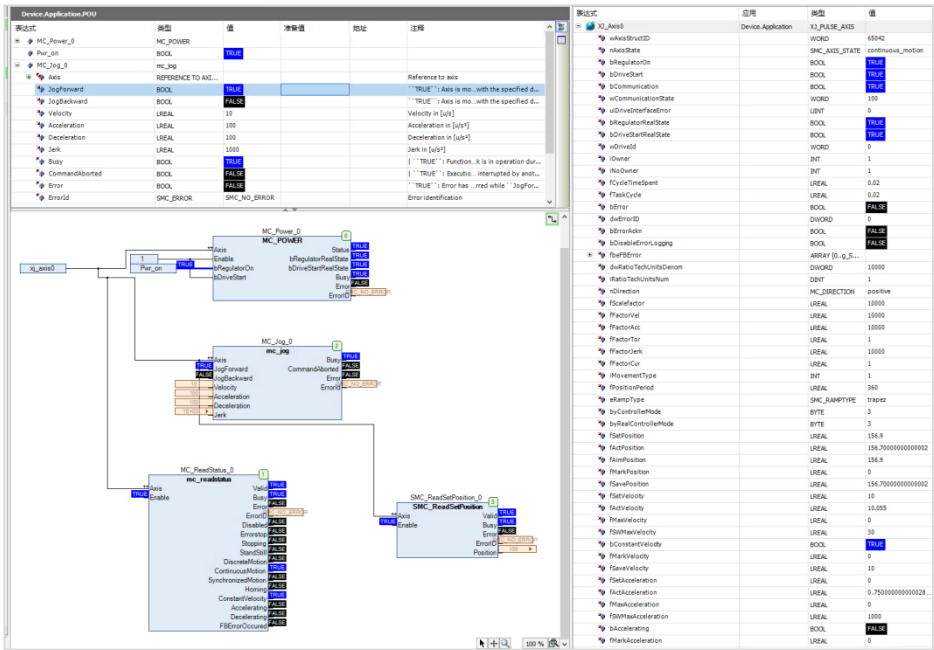
恢复默认值

- 轴 0 实例化名称默认为：XJ_Axis0；支持用户手动修改。
- 软件限位：正限位（默认 1000.0）、负限位（默认 0.0）输入范围为 LREAL 类型的范围，负限位需小于等于正限位。
- 模数值：模数值需大于 0
- Z 相个数：范围[0,32767]，取消分配时，Z 相个数清零。
- 回零参数：参数类型为 LREAL，其中回零速度、爬行速度、回零加速度、回零减速度、回零加加速度输入范围为（0，LREAL 类型最大值），零点位置为 LREAL 类型的范围。

259

4、指令使用示例

使用【MC_POWER】、【MC_JOG】等指令实现脉冲轴点动、位置以及轴状态获取功能，配置如下图：



8-3-2. PWM 输出功能

在硬件参数配置界面可对 PWM 输出进行配置。结合【XJ_PWM】指令使用时需先在硬件参数界面勾选对应的端口。配置如图所示：



可结合脉宽调制【XJ_PWM】指令使用，以 Y0 端子为例进行 PWM 脉宽调制输出，配置如下图所示：

Device.Application.POU

表达式	类型	值	准备值	地址	注释
XJ_PWM_0	XJ_PWM				
xEnable	BOOL	TRUE			TRUE: 输出PWM波形, FALSE: 停止输出
ePort	Y_PORT	Y0			输出端口
byDuty	WORD	32767			占空比, 范围为 1~65535
udiFrequency	UDINT	100			输出频率, 单位为 0.1Hz, 范围为 1~200KHz
xValid	BOOL	TRUE			是否有效
xBusy	BOOL	TRUE			是否正在执行
xError	BOOL	FALSE			错误标记
eErrorID	HSIO_ERROR	ERR_OK			错误代码
cycleUs	REAL	100000			

XJ_PWM_0

XJ_PWM

0

TRUE

TRUE

FALSE

ERR_OK

Y0

32767

100

xEnable

ePort

byDuty

udiFrequency

xValid

xBusy

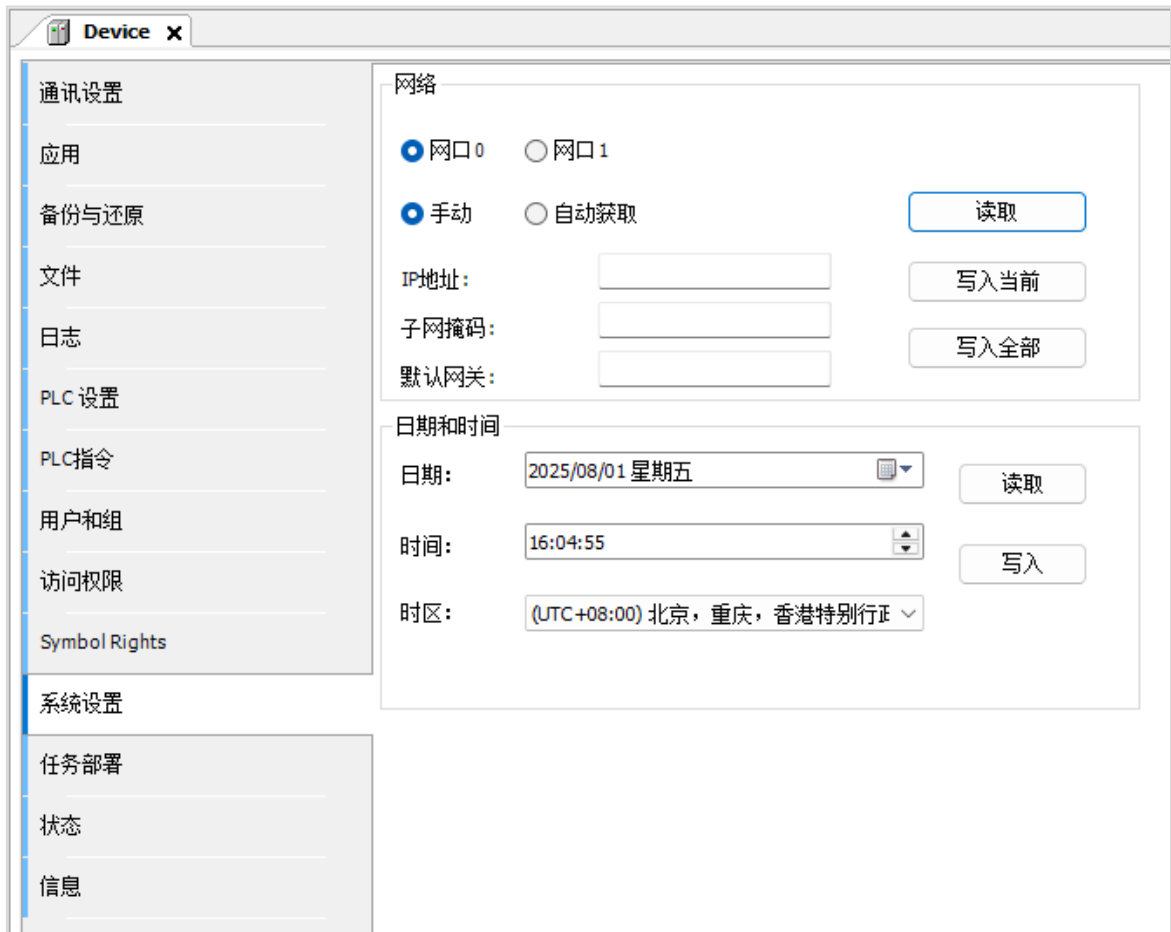
xError

eErrorID

8-4. 系统设置

8-4-1. 固件 1.1.0 及以上版本系统设置应用举例

在应用的工程中，双击设备“Device”，找到“系统设置”，在系统设置的界面中可读取/设置网口 IP、系统时间。

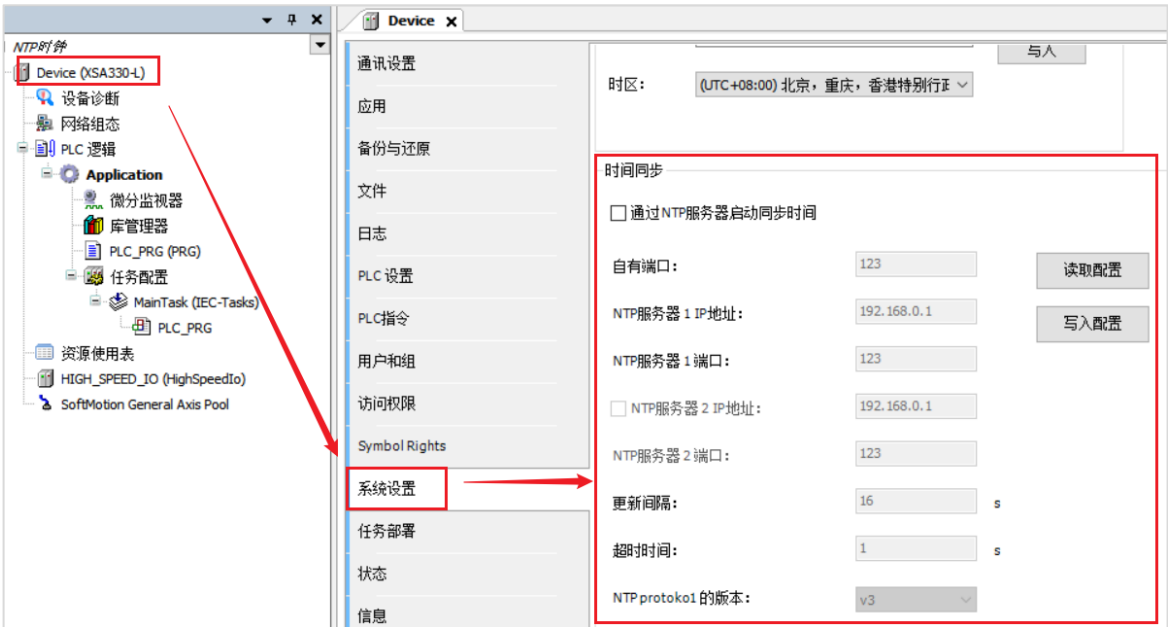


- 读取 IP：若该网口没有插网线，则无法获取该网口的全部 IP 信息；
- 写入/读取日期和时间：同时读取/写入日期、时间、时区信息。

8-4-2. NTP 同步时间

使控制器与网络中的 NTP 服务器(如电脑)保持时间一致,确保设备系统时间的准确性和统一性。

1、参数界面



2、参数介绍

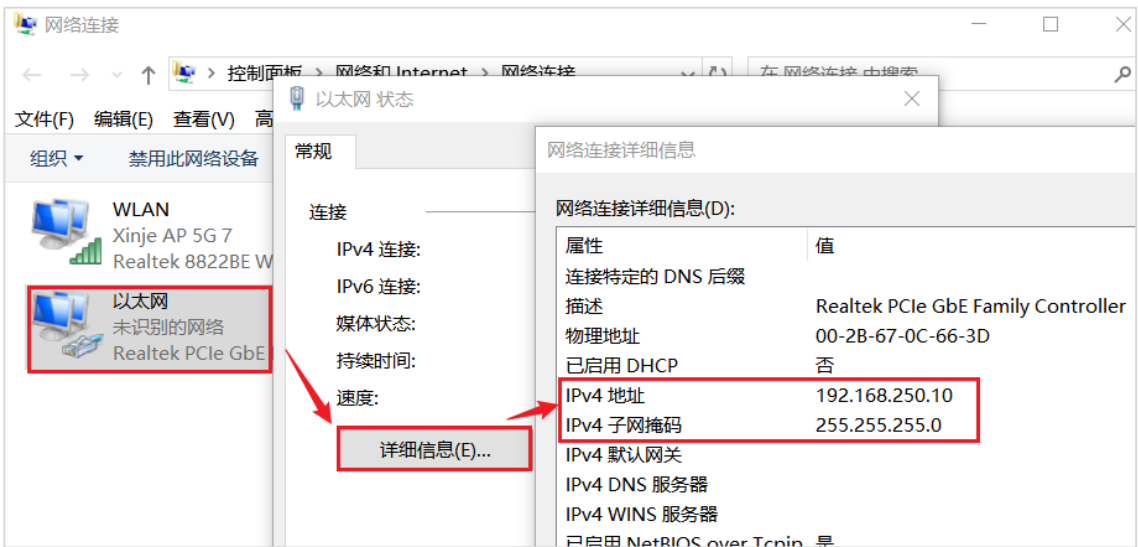
命令	默认值	功能
通过 NTP 服务器启动同步时间	(未勾选)	勾选并写入后,控制器可以与网络中的 NTP 服务器保持时间一致。
自有端口	123	(不可更改)
NTP 服务器 1 IP 地址	192.168.0.1	配置 NTP 服务器 1 的 IP 地址,控制器的时间与该服务器的时间保持一致。
NTP 服务器 1 端口	123	(不可更改)
NTP 服务器 2 IP 地址	192.168.0.1	若需启用冗余 NTP 服务器功能,则勾选“NTP 服务器 2 IP 地址”前的方框,输入 NTP 服务 2 的 IP 地址。NTP 服务器 1 和 NTP 服务器 2 的 IP 地址不能设置成一样的。
NTP 服务器 2 端口	123	(不可更改)
更新间隔	16s	根据“更新间隔”设置的时间,控制器会周期得与服务器时间进行校验同步。 给值范围: 16-86400s
超时时间	1s	控制器与服务器的 NTP 通讯断开超过“超时时间”后,会弹窗提示超时错误。 给值范围: 1-10s
NTP protokol 的版本	v3	v3: 仅支持 IPv4 网络,应用较为广泛 v4: 是对 v3 的扩展,同时支持 IPv4 和 IPv6 网络(暂不支持 Ipv6)

命令	默认值	功能
		可以手动切换 v3、v4 版本
读取配置	/	在点击“读取配置”时，若下位机中有时间同步配置信息，则将下位机中存储的所有时间同步配置信息读到上位机界面中。
写入配置	/	在点击“写入配置”时，若此时没勾选“通过 NTP 服务器启动同步时间”，则关断 NTP 时间同步，且弹窗提示“NTP 时间同步关闭成功”；

3、操作步骤

举例：控制器连一台电脑，该电脑作为 NTP 服务器。

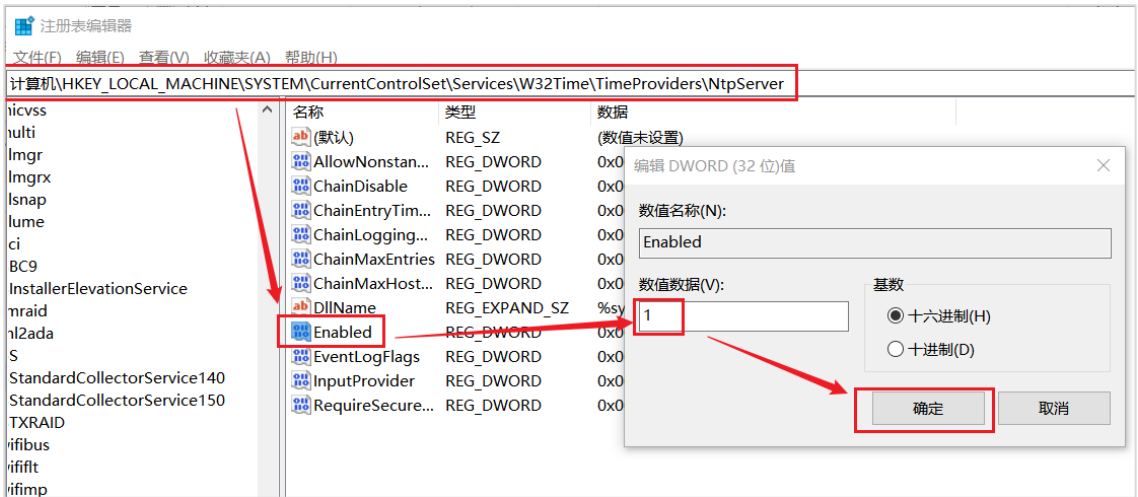
1、电脑与控制器网线连接，设置好并查看电脑网卡的 IP 地址。



2、开启 NtpServer 服务

Win+R 组合键在“运行”窗口中输入 regedit，打开注册表编辑器，然后复制如下项目路径到编辑器界面的路径上再回车：

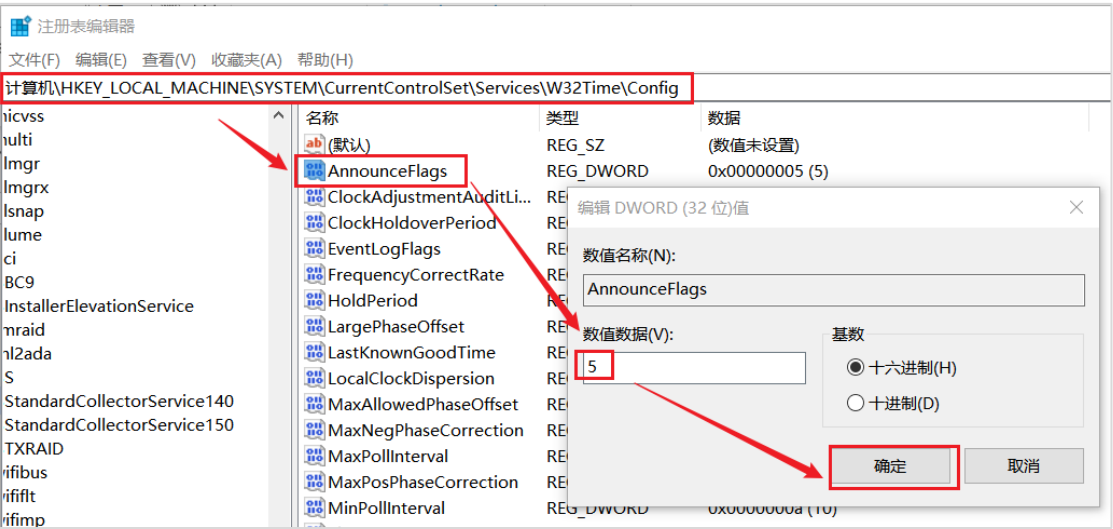
计算机\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer，将 Enabled 设置为 1，开启 NTP 服务。



然后再复制如下项目路径到编辑器界面的路径上再回车：

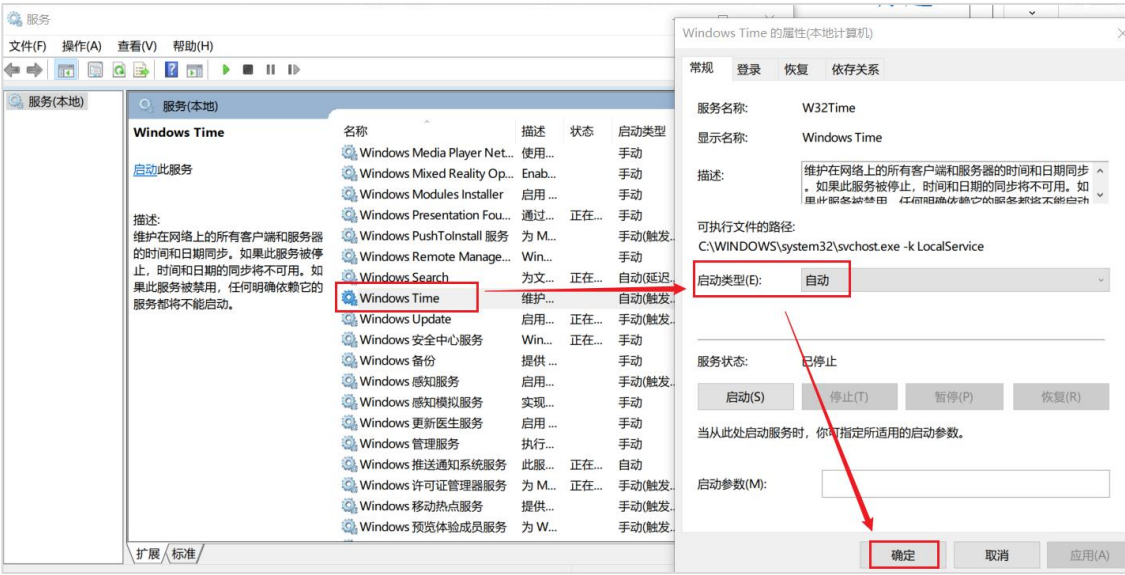
计算机\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config

将 AnnounceFlags 的值设置为 5（系统默认为 a）。这个设置是允许 NTP 服务器向其客户端广播其时间服务，并且允许客户端请求同步时间。

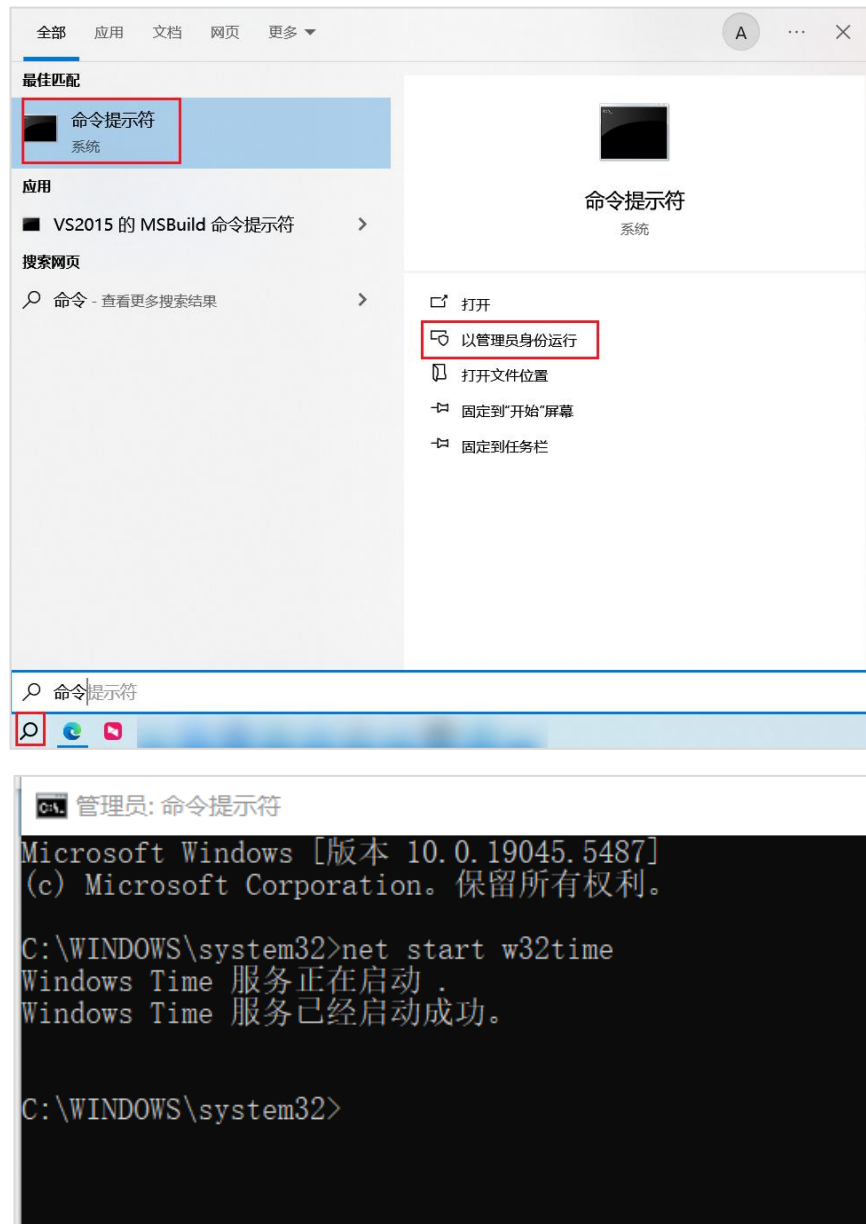


3、启动 NtpServer 服务

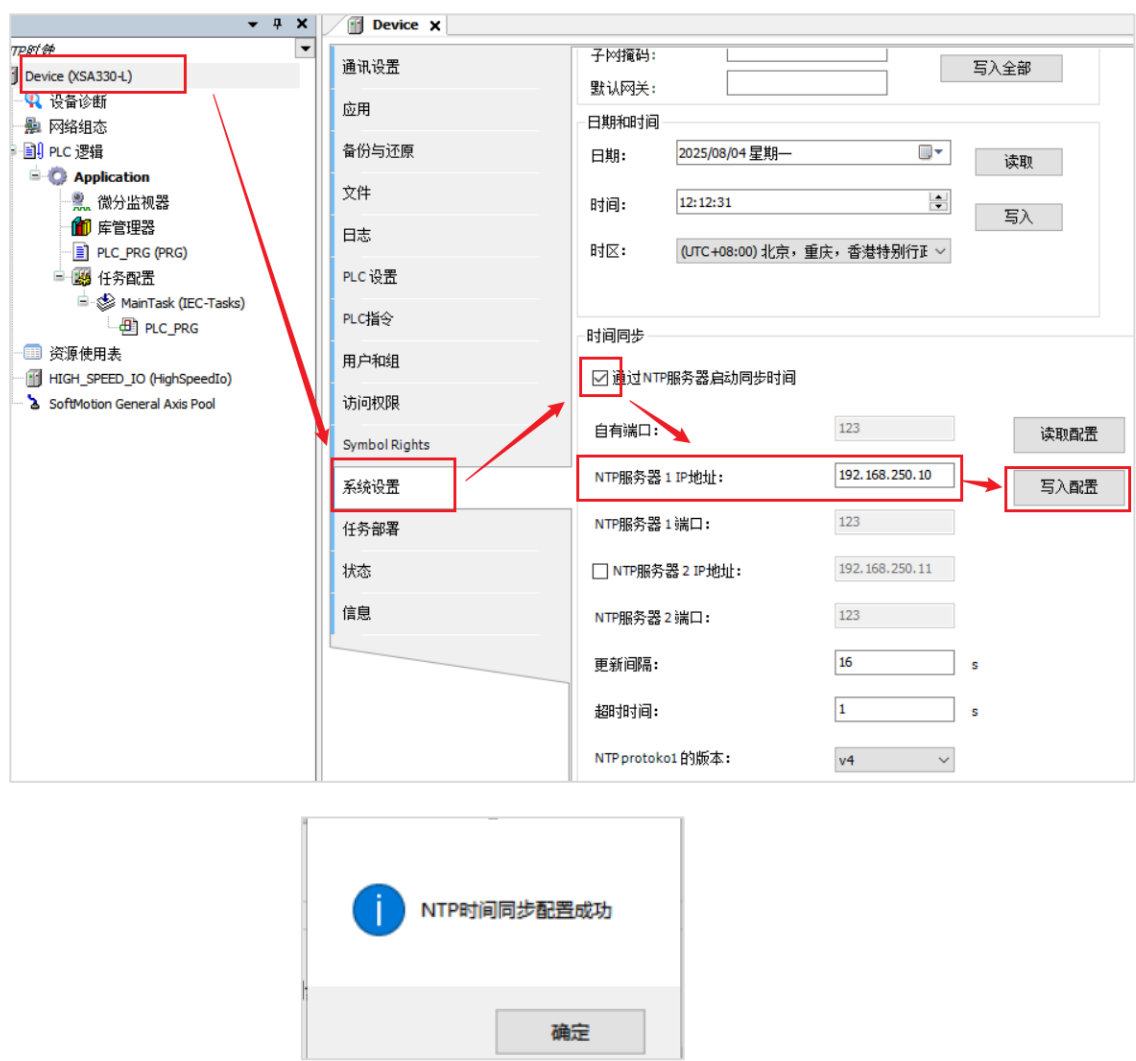
Win+ R 组合键在“运行”窗口中输入 services.msc，打开服务。将 Windows Time 服务的启动类型设置为自动。



在电脑搜索栏搜索“cmd”或“命令提示符”找到命令提示符，点击“以管理员身份运行”，输入“net start w32time”后回车，即可启动 windows NTP（此时该命令提示符界面可关、也可不关）。



- 4、控制器与 XS Studio 联机，打开“Device”-“系统设置”界面，勾选“通过 NTP 服务器启动同步时间”，在“NTP 服务器 1 IP 地址”里填写电脑的 IP，然后点击“写入配置”。



5、时间同步配置成功后，点击日期和时间旁边的“读取”按钮，即可将 NTP 服务器中的时间读取到界面中。



 注意:

- 若用户想启用冗余 NTP 服务器功能，则勾选“NTP 服务器 2 IP 地址”前的方框，勾选后，将局域网内另一台用作 NTP 服务器的电脑的 IP 填写进去，并写入配置即可（两个 NTP 服务器电脑的 IP 不要重复冲突）。此时若 NTP 服务器 1 的 NTP 服务被关闭了，控制器会与 NTP 服务器 2 的时间进行同步。
- 新机或控制器刷机后，未操作“写入配置”而直接点击“读取配置”时，会提示“未配置任何 NTP/SNTP 时间同步信息”。
- 设置 NTP 时间同步后，作为服务器的电脑的时间不可设置为手动，必须是自动设置时间的配置。



8-5. PLC 指令

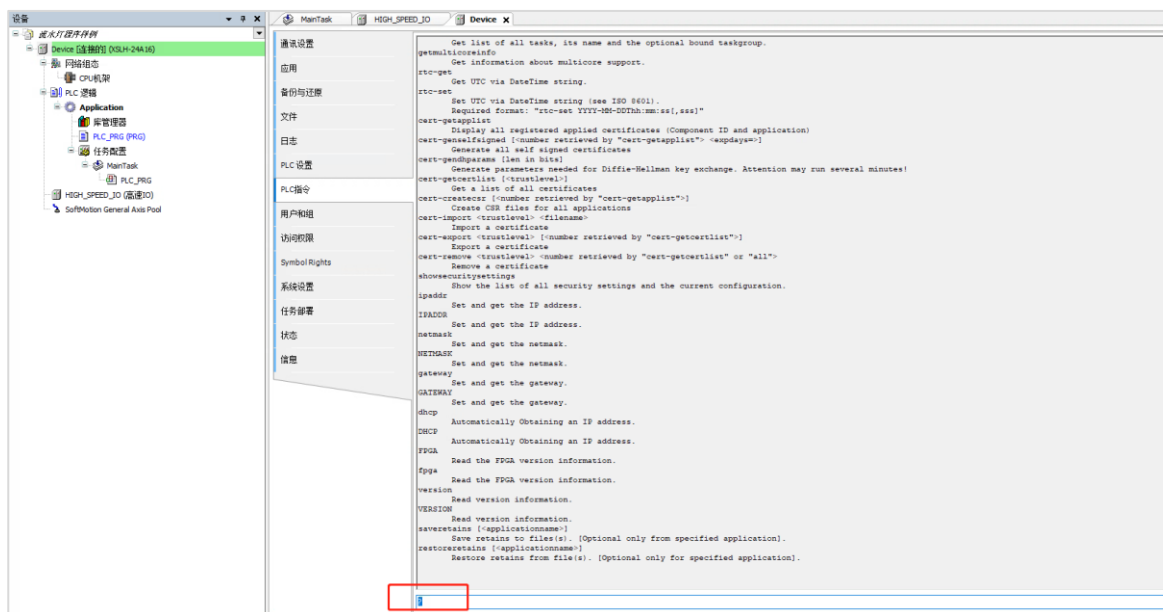
PLC 指令功能是一个基于文本的控制监视器，可以用于查询控制器的特定信息，在输入窗口中输入指定命令，结果窗口中接收来自控制器的响应。

指令列表

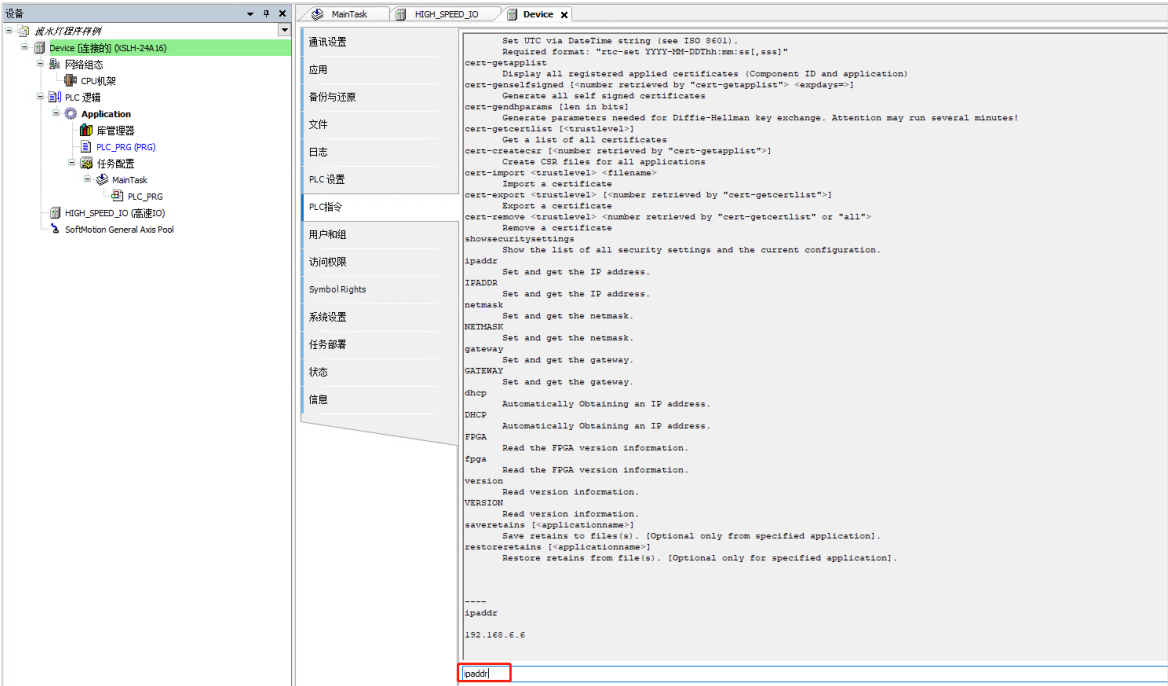
指令名称	功能介绍
ipaddr / IPADDR	获取/设置 PLC 的 IP 地址（仅支持 XSDH、XSLH）
netmask / NETMASK	获取/设置 PLC 的子网掩码（仅支持 XSDH、XSLH）
gateway / GATEWAY	获取/设置 PLC 的网关（仅支持 XSDH、XSLH）
dhcp / DHCP	设置 IP 为自动获取（仅支持 XSDH、XSLH）
version / VERSION	获取 PLC 的固件版本
rtc-get / RTC-GET	获取当前 UTC 时间
rtc-set / RTC-SET	设置 UTC 时间

8-5-1. 应用举例

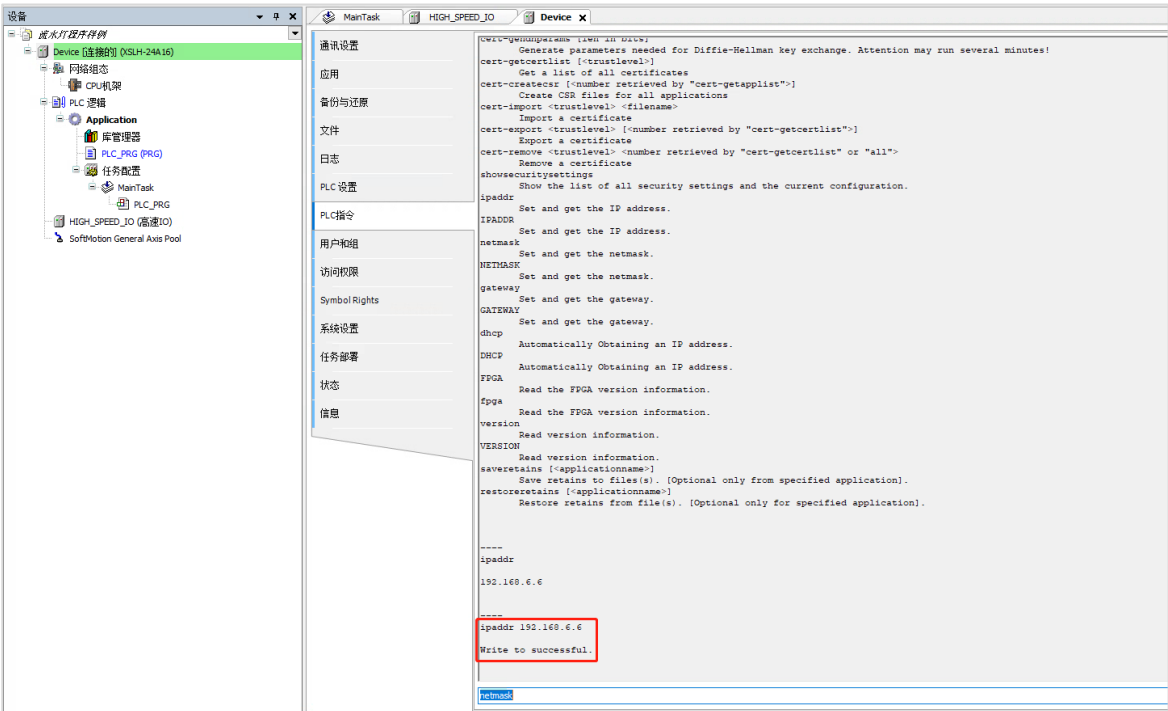
双击“Device”，在“PLC 指令”里输入“？”，可以出来全部功能。可以在这里修改 IP，获取固件版本，设置/读取时钟信息等等。



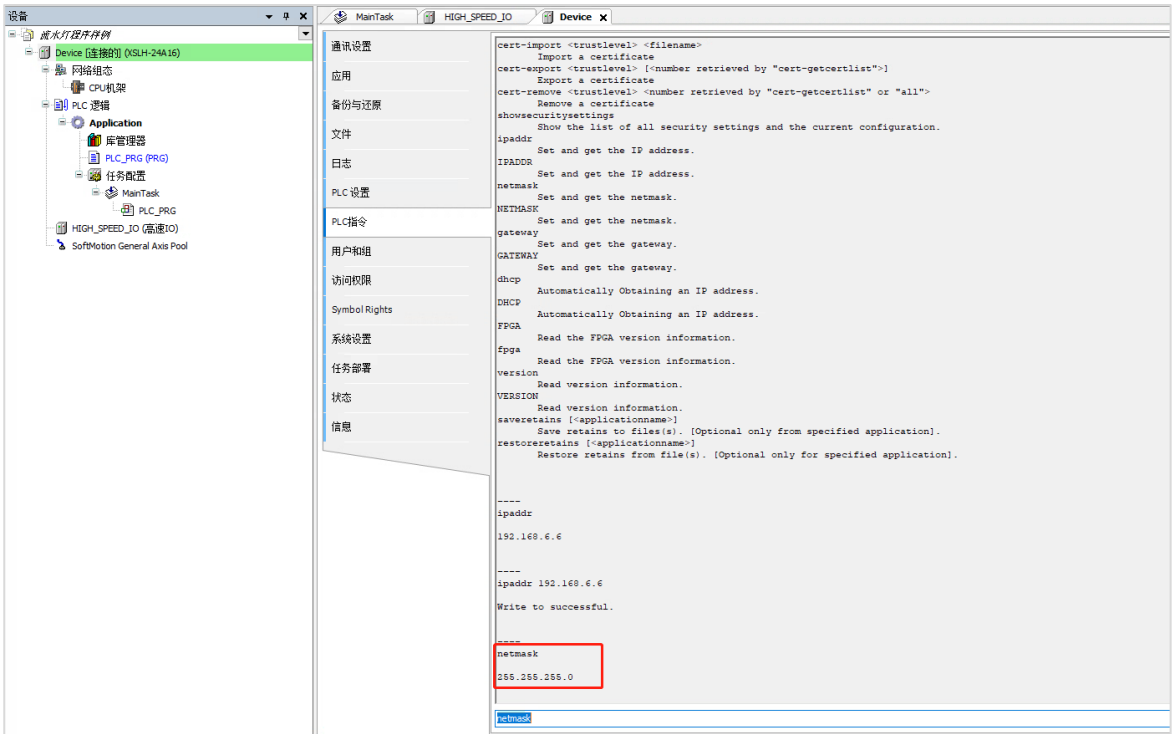
比如：输入“ipaddr”就可以获取到 PLC 的当前的 IP 地址。



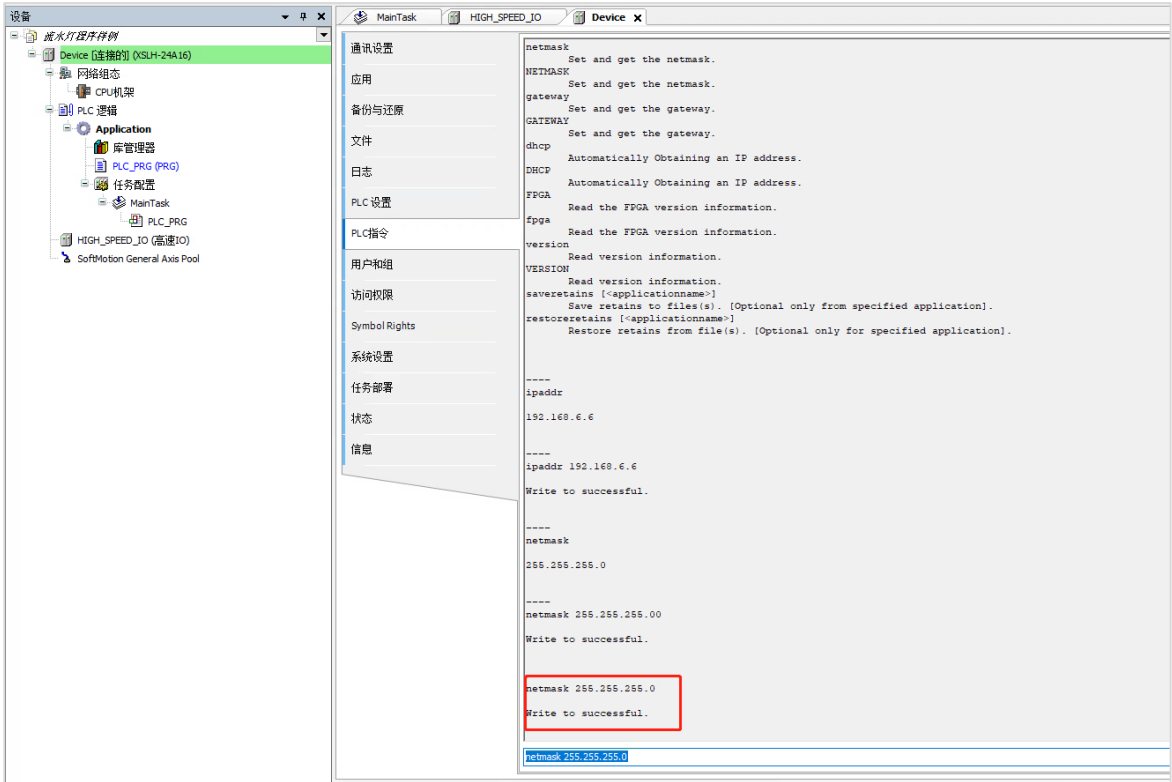
输入“ipaddr 192.168.6.10”，设置 PLC 的 IP 地址，显示“Write to successful”则写入成功，重新上电生效。



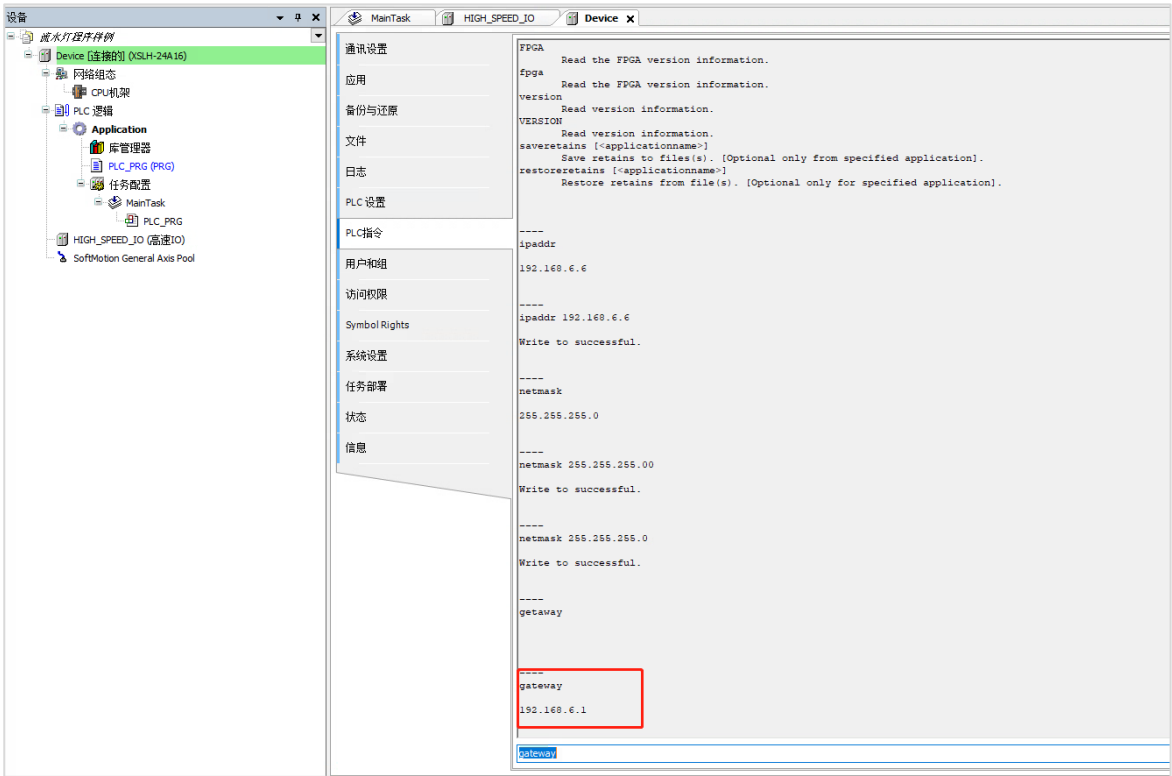
输入“netmask”，就可以获取到 PLC 的当前的子网掩码。



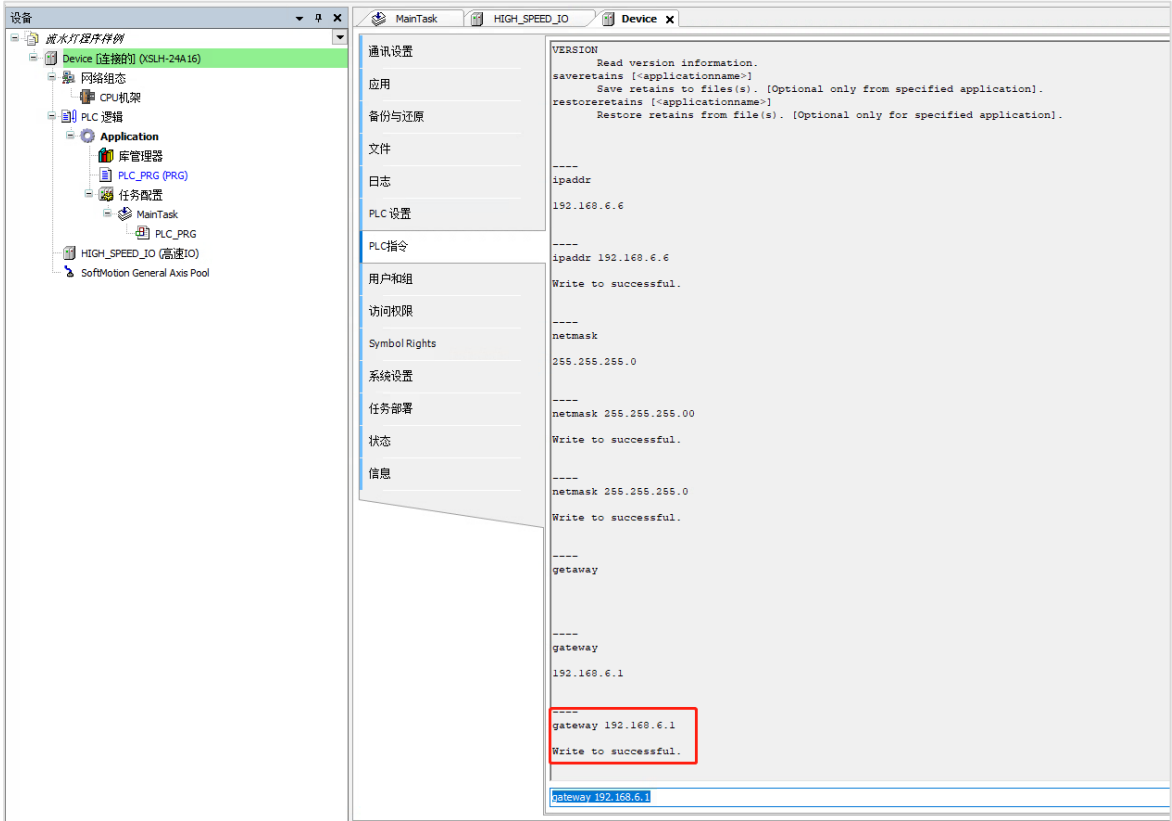
输入“netmask 255.255.254.0”，设置 PLC 的子网掩码，显示“Write to successful”则写入成功。



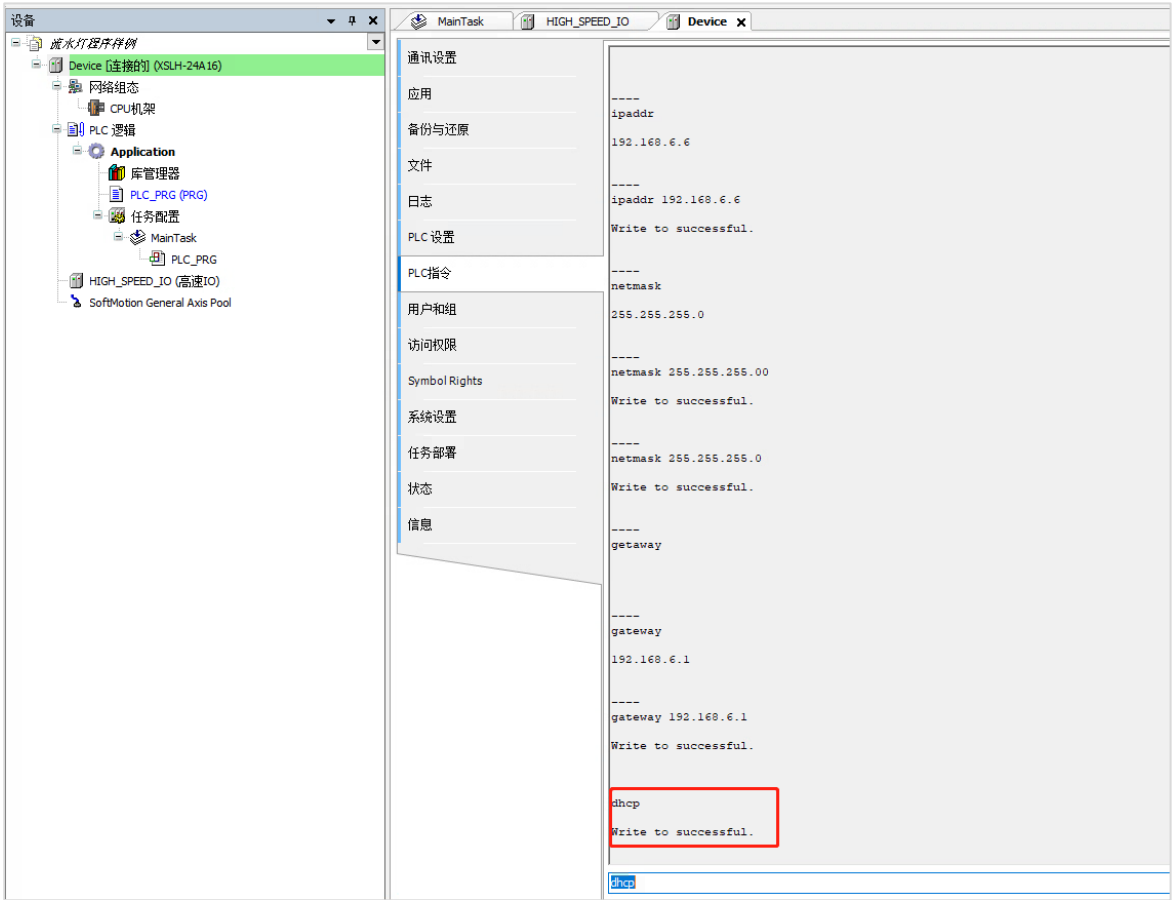
输入“gateway”，就可以获取到 PLC 的当前的默认网关。



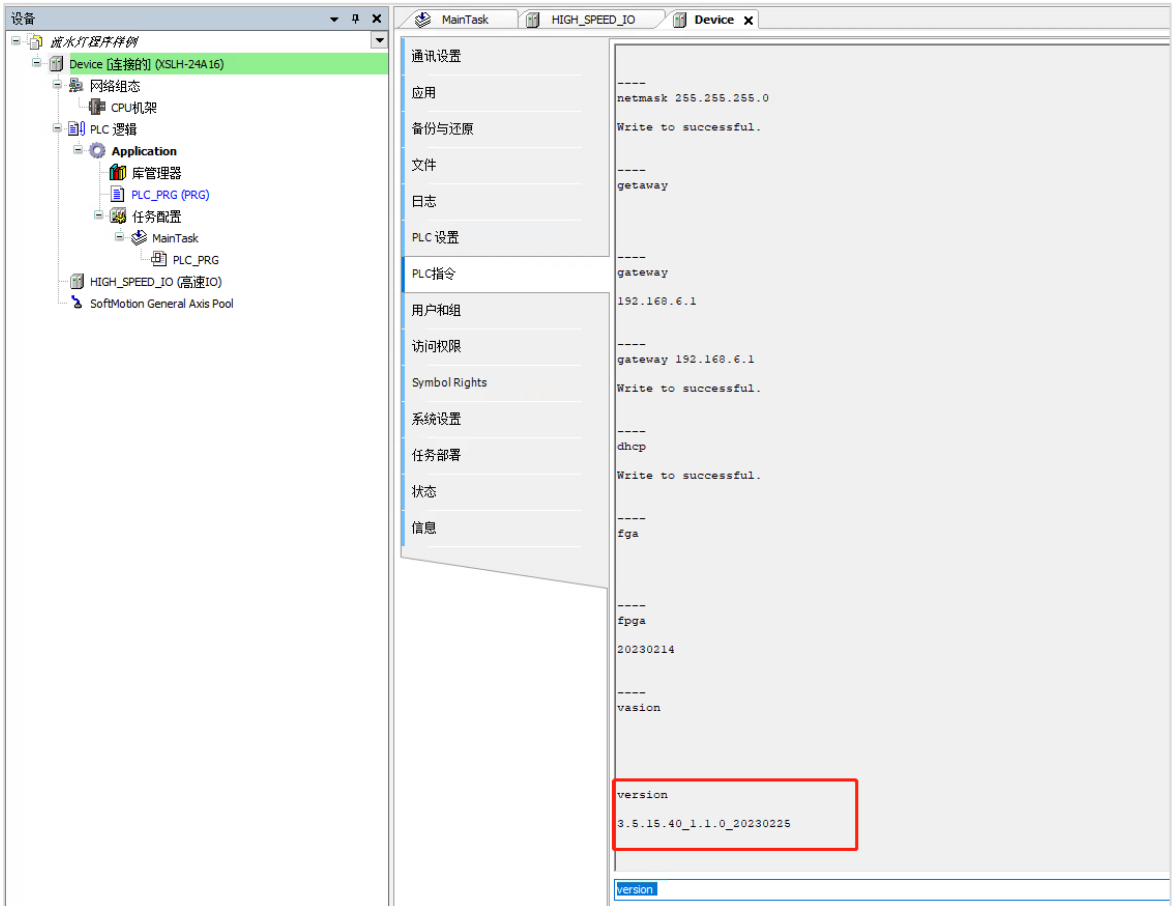
输入“gateway 192.168.6.1”，设置 PLC 的网关，显示“Write to successful”则写入成功。



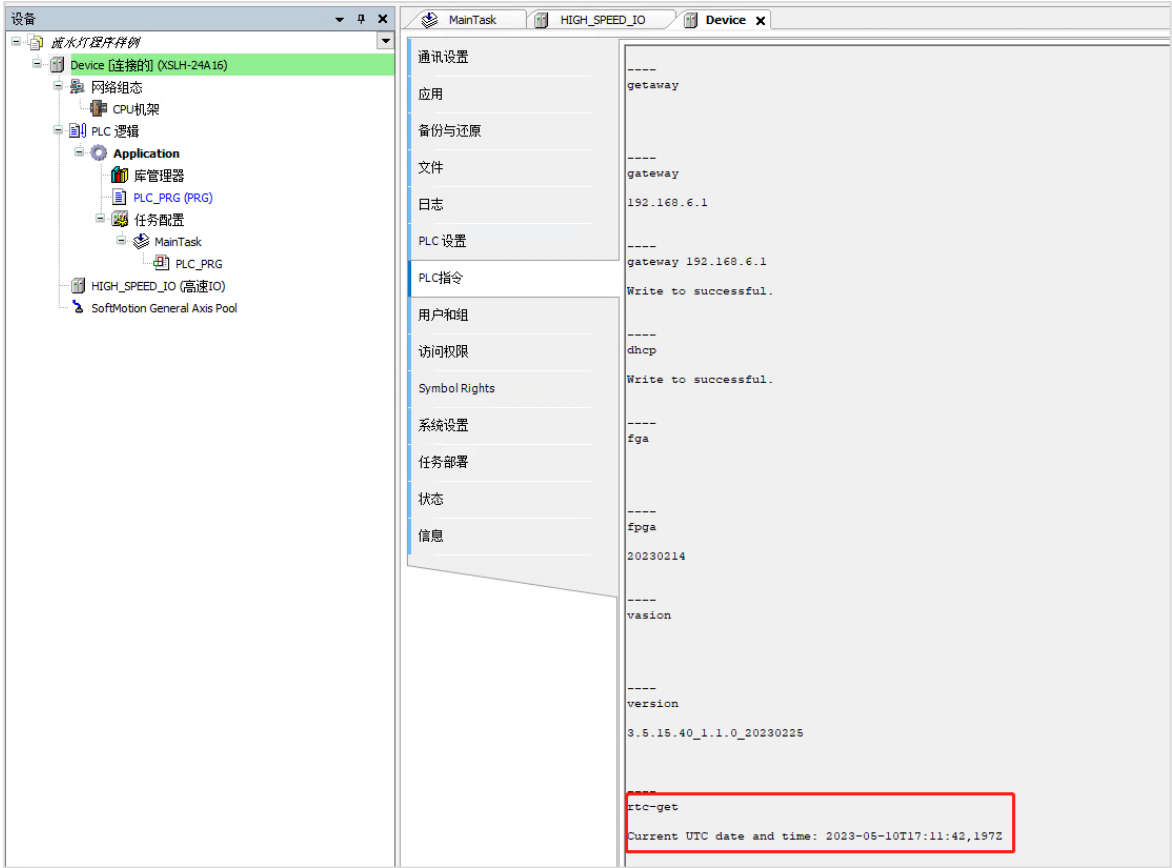
输入“dhcp”，设置 PLC 的 IP 获取方式为自动获取，显示“Write to successful”则写入成功。当 IP 的获取方式为自动获取时，需要保证网络环境良好。



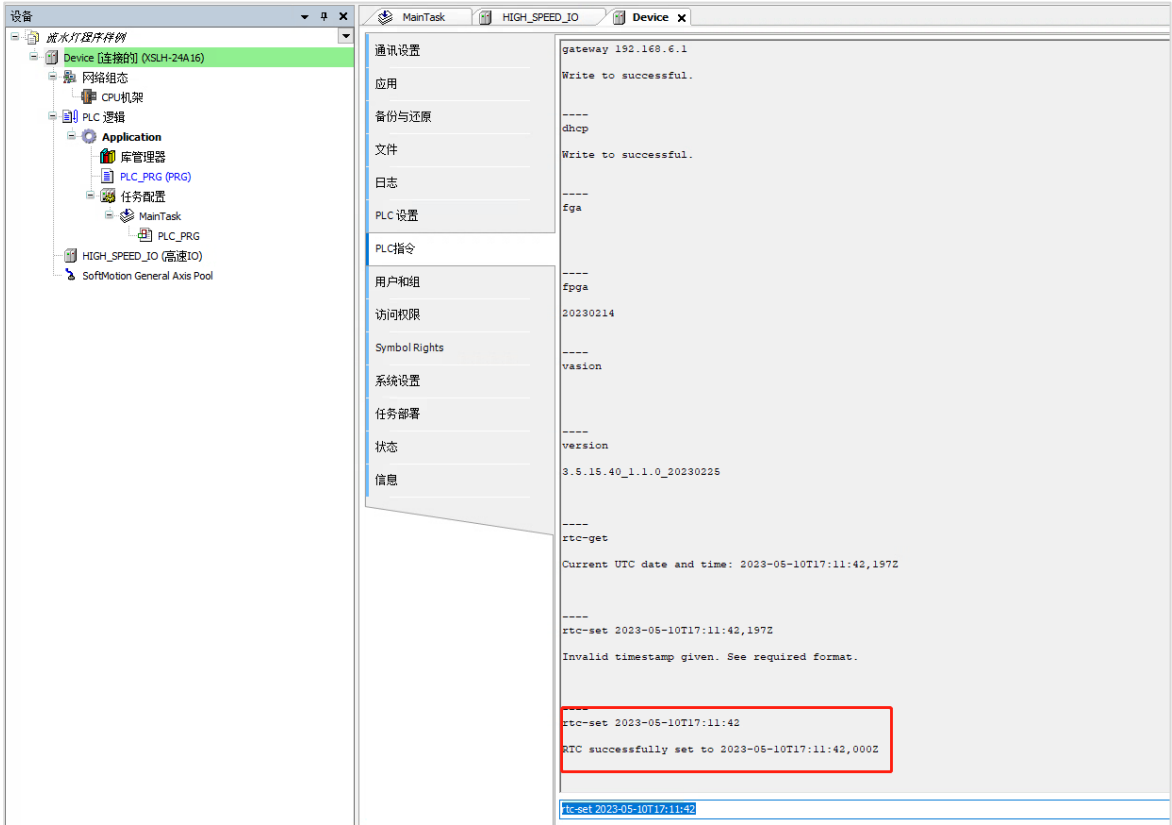
输入“version”，就可以获取到 PLC 当前的固件版本。



输入“rtc-get”，就可以获取当前的 UTC 时间。



输入“rtc-set 2021-10-25T18:24:30”，可设置 UTC 时间。显示“RTC successfully set to 2021-10-25T18:24:30,000Z”则写入成功。其中“000Z”显示不定。

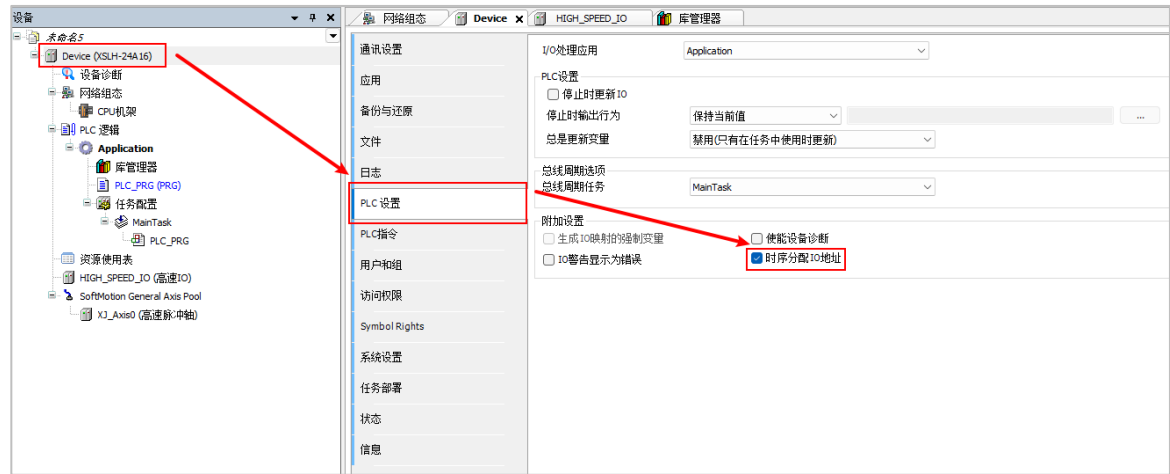


8-6. 固定地址分配

8-6-1. 设置地址分配方式

1、操作界面

在左侧设备树中双击 PLC 设备[例如“Device(XSLH-24A16)”],在打开的界面中单击“PLC 设置”,设置“其他设置”区域中“时序分配 IO 地址”参数。

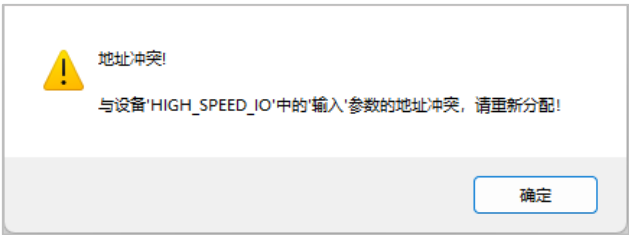


- 时序地址分配：按照设备添加的时间先后顺序分配 IO 映射变量的地址。
- 顺序地址分配：按照设备在设备树中的位置分配 IO 映射变量的地址。
- 默认勾选（勾选：按照时序地址分配；不勾选：按照顺序地址分配）。

2、功能说明

时序地址分配：新增设备时，在不改变当前其余设备地址的基础上自动分配“空闲地址”；

- 当手动分配地址时，如果输入地址与当前已用的地址冲突，则提示用户“地址冲突！与 xxx 参数的地址冲突，请重新分配”，点击确定后返回上一次正确的值。如图所示：



- 当手动分配地址后，变量会出现 M 标志，当取消标志时地址不会重新分配(仅取消 M 标志)，如图例所示：

通用 过程数据 启动参数 日志 EtherCAT/I/O映射 EtherCATIEC对象 状态 信息	查找		过滤 显示所有		给IO通道添加FB... 转到实例 连续		
	变量	映射	通道	地址	类型	单元	描述
			Control Word	%QW1	UINT		Control Word
			TargetPosition	%QD1	DINT		TargetPosition
			TargetVelocity	%QD2	DINT		TargetVelocity
			TargetTorque	%QW6	INT		TargetTorque
			ModeOfOperation	%QB14	SINT		ModeOfOperation
			Status Word	%IW1	UINT		Status Word
			ActualPosition	%ID1	DINT		ActualPosition
			Velocity actual value	%ID2	DINT		Velocity actual value
			ActualTorque	%IW6	INT		ActualTorque
			ModeOfOperationDisplay	M %IB14	SINT		ModeOfOperationDisplay

- 当复制粘贴时，按照“时序地址分配”功能分配设备地址（若存在 M 标志，粘贴的设备会取消 M 标志重新分配）；
- 空闲地址：即从起始地址开始查找符合条件的连续地址，而不是根据上一个设备的结束地址来查找地址；
- 无论删除、移动等操作，已使用的设备地址都不会改变。

8-6-2. 连续 IO 地址

所有设备的 IO 地址界面，可启用“连续 IO 地址”功能。

点击“连续 IO 地址”时，对该界面 IO 全部重新分配地址（带有 M 标志的也会被重新分配），并将设备的 IO 映射地址重新分配为连续地址。

以本体高速 IO 界面为例：

The screenshot shows the SIMATIC Manager HW Config interface. The left sidebar displays the project tree with the following structure:

- 设备 (Device) - SIMATIC 300 (16)
- 设备树 (Device Tree)
- 网络组态 (Network Configuration)
- EtherCAT 机架 (EtherCAT Rack)
- CPU 机架 (CPU Rack)
- PLC 逻辑 (PLC Logic)
- Application
 - 库管理器 (Library Manager)
 - PLC_PRG (PRG)
 - 任务配置 (Task Configuration)
 - EtherCAT_Task
 - MainTask
 - PLC_PRG

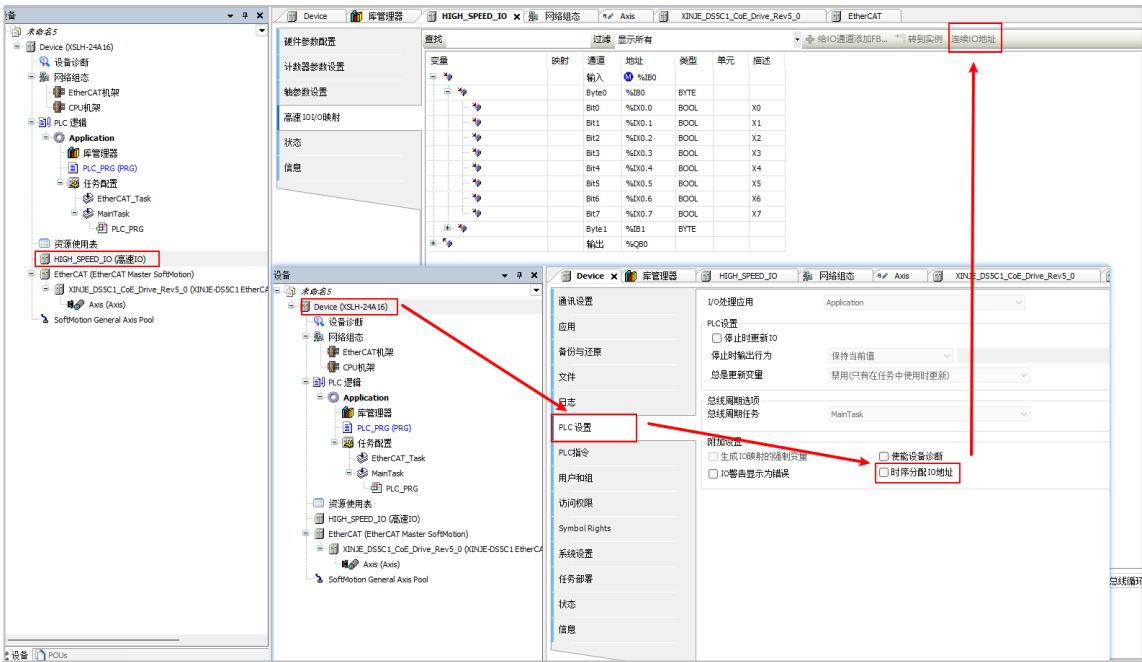
- 资源使用表 (Resource Usage Table)
- HIGH_SPEED_IO (高速IO)** (Selected)
- EtherCAT (EtherCAT Master SoftMotion)
- XINUE_DSSC1_CoE_Drive_Rev5_0 (XINUE-DSSC1 EtherCAT)
- Axis (Axis)
- SoftMotion General Axis Pool

The main window displays the 'HIGH_SPEED_IO' configuration table. The table has the following columns: 变量 (Variable), 映射 (Mapping), 通道 (Channel), 地址 (Address), 类型 (Type), 单元 (Unit), and 描述 (Description). The '连续IO地址' (Continuous IO Address) column is highlighted in red.

变量	映射	通道	地址	类型	单元	描述
		输入	%B0			
		Byte0	%B0	BYTE		
		B00	%D0.0	BOOL	X0	
		B01	%D0.1	BOOL	X1	
		B02	%D0.2	BOOL	X2	
		B03	%D0.3	BOOL	X3	
		B04	%D0.4	BOOL	X4	
		B05	%D0.5	BOOL	X5	
		B06	%D0.6	BOOL	X6	
		B07	%D0.7	BOOL	X7	
		Byte1	%B1	BYTE		
		输出	%Q0			



只在时序地址分配模式下生效，若当前模式为“顺序地址分配”时，“连续 IO 地址”按钮显示灰色，不可点击使用。



8-7. 时钟

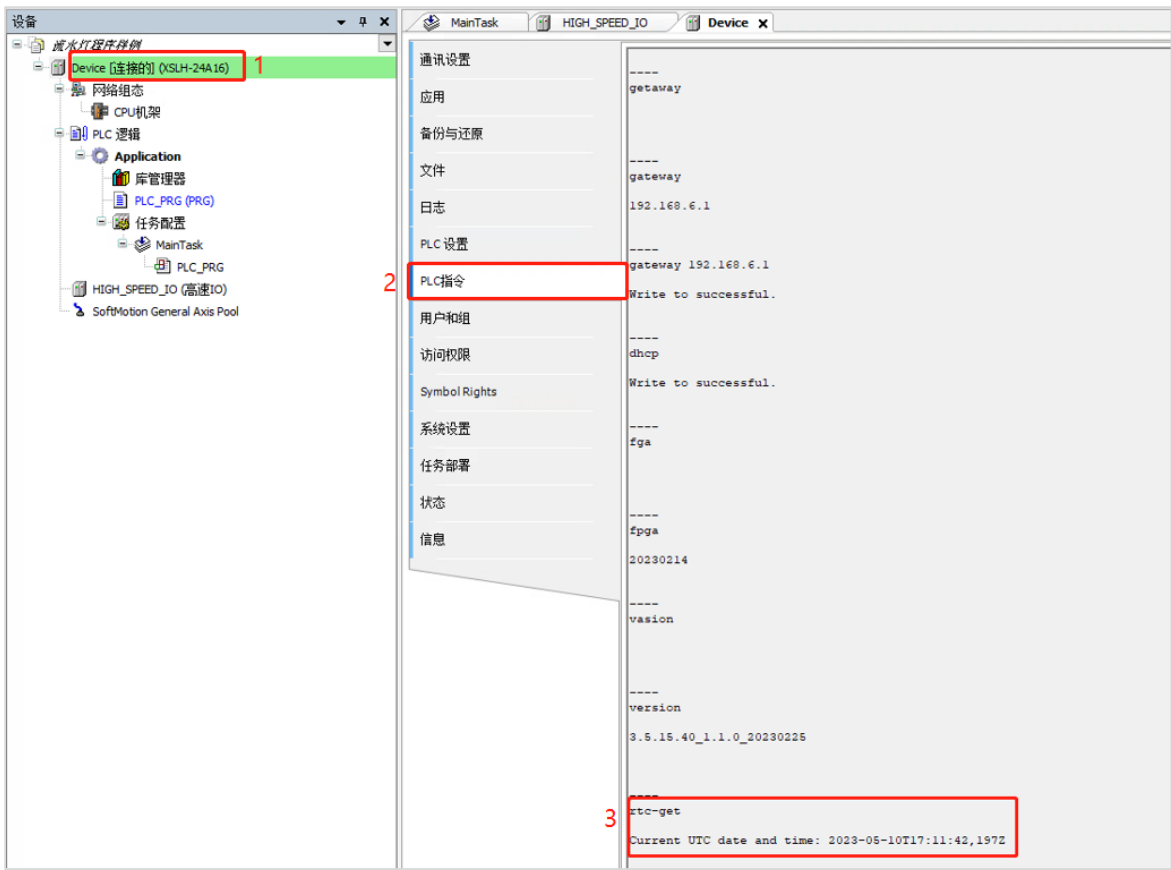
8-7-1. 功能概述

XS 系列 PLC 集成了 RTC，用于记录当前的系统时间，该时钟由电池供电，可保证时间的精确性，同时也支持用户手动修改 RTC 时间。

8-7-2. 应用举例

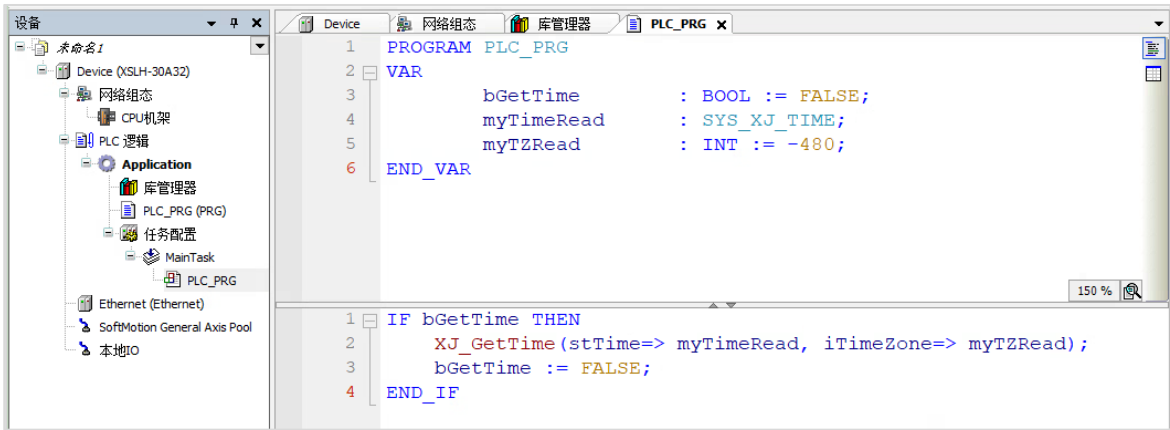
获取时间的方式：

1、双击“Device”，在“PLC 指令”里输入“rtc-get”，获取当前时间。

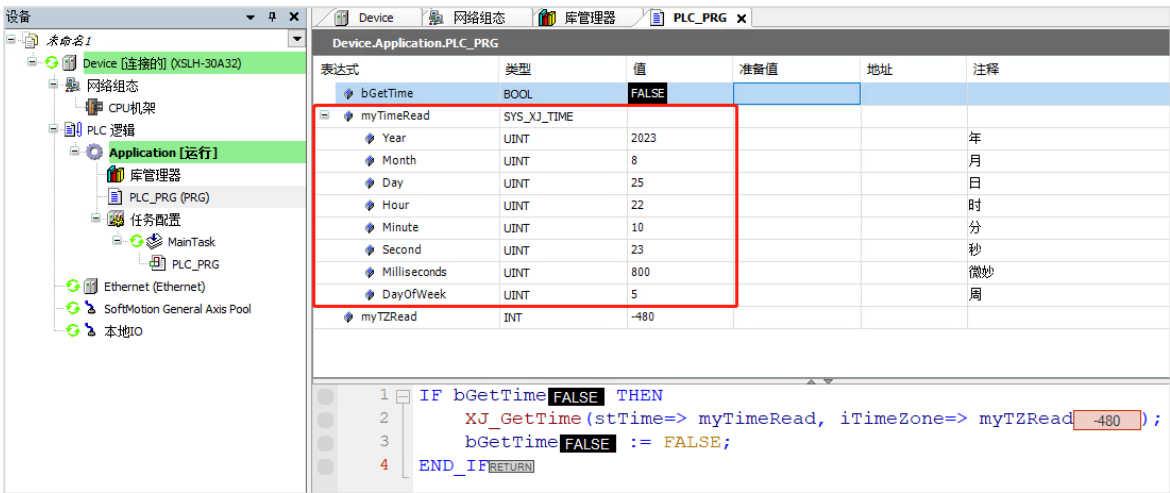
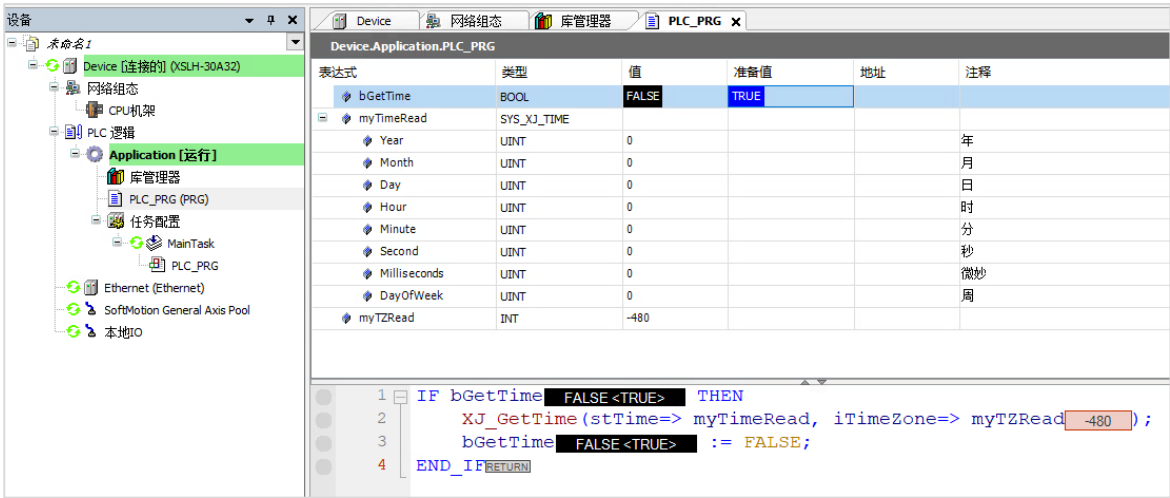


2、通过时钟指令

1) 打开软件，在 PLC-PRG 编辑器内编写 XJ_GetTime 指令。如下图：



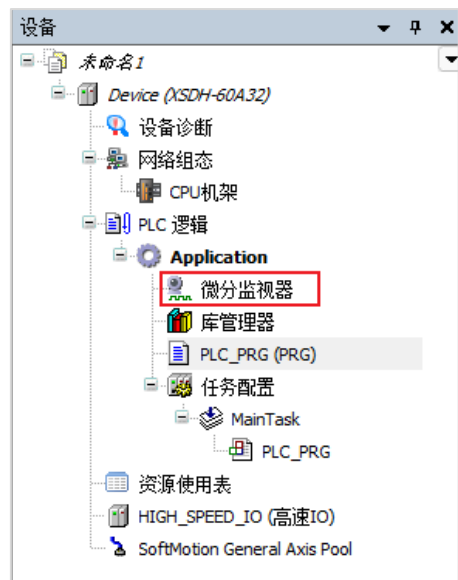
2) 与 PLC 设备建立连接，登录后运行。如下图：



运行后，已正确读取时间并显示。

8-8. 微分监视器

8-8-1. 功能概述

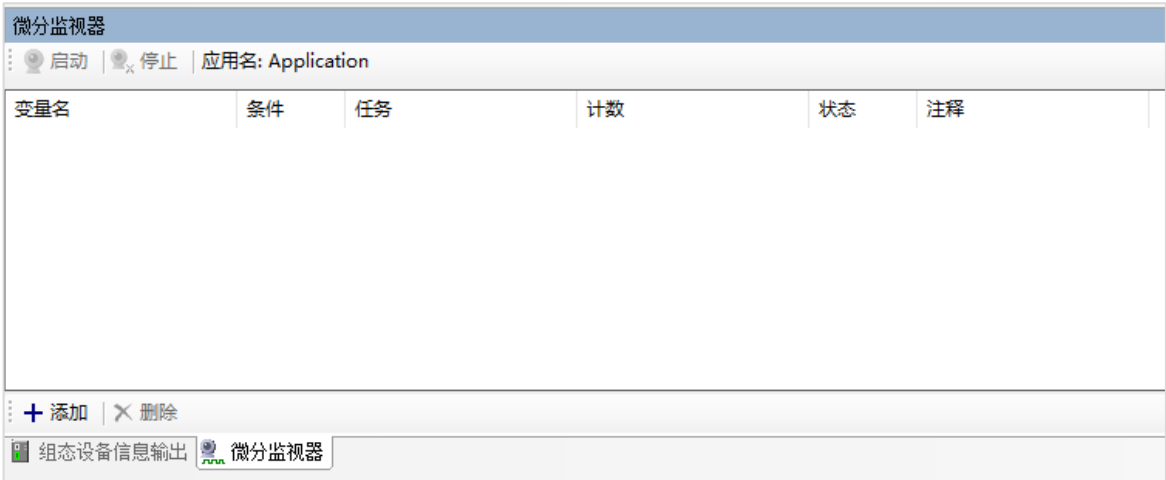


微分监视器用于检测边沿信号的触发次数。可用于监视设定任务周期内 BOOL/BIT 变量的变化次数，最多只能监控 8 个变量。



- 仅标准工程支持微分监视器（库工程不支持）。
- Application 里只允许添加一个微分监视器，可以删除，删除后可以通过右键“Application” - “添加对象” - “微分监视器”进行添加。

8-8-2. 参数说明



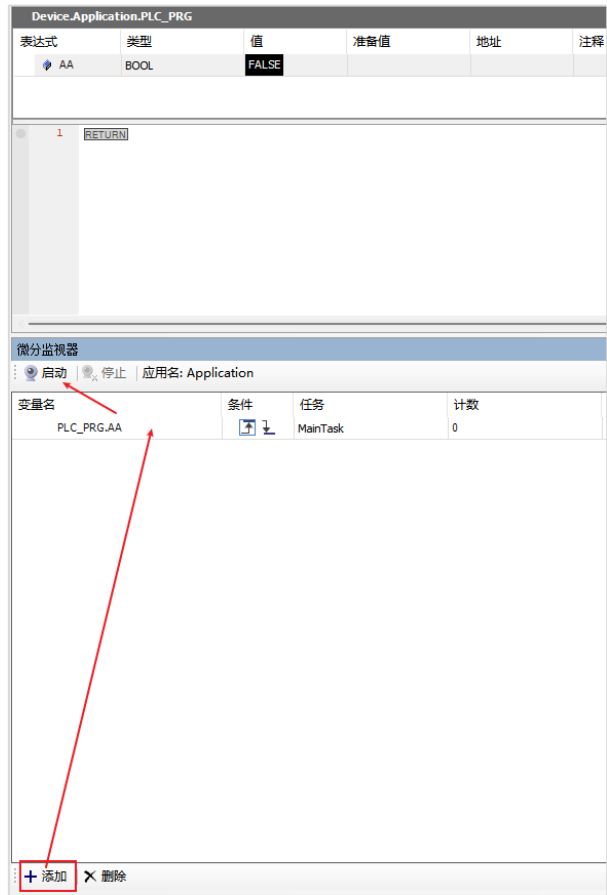
参数名称	参数说明
应用名	应用名称
变量名	监视变量的名称 监视精准场景：监视变量只在任务 A 中进行读写，监视任务选择任务 A，可准

参数名称	参数说明
	确监视变量的状态；其他情况下监视可能存在遗漏计数
条件	监视变量的条件，单击图标选择监视的条件。分为上升沿和下降沿 默认监视上升沿
任务	监视变量挂载的任务，可单击选择挂载的任务 默认为 MainTask
计数	统计触发监视变量条件的次数 在监视变量所在的任务周期内触发监视变量条件，即使在周期内发生了多次翻转，也只计数一次； 变量计数条件是与上一个周期最后一次记录的布尔类型值进行比较，所以在启动微分监视器监视时，程序第一次运行的第一个周期中不进行计数操作
状态	监视变量的状态指示 不触发监视变量：空白； 触发监视变量：会显示绿色的点位图标并闪烁，触发一次闪烁 6s 左右时长
注释	监视变量的注释内容 获取该变量的注释信息，最多显示 100 个字符串
添加/删除	添加/删除监视变量

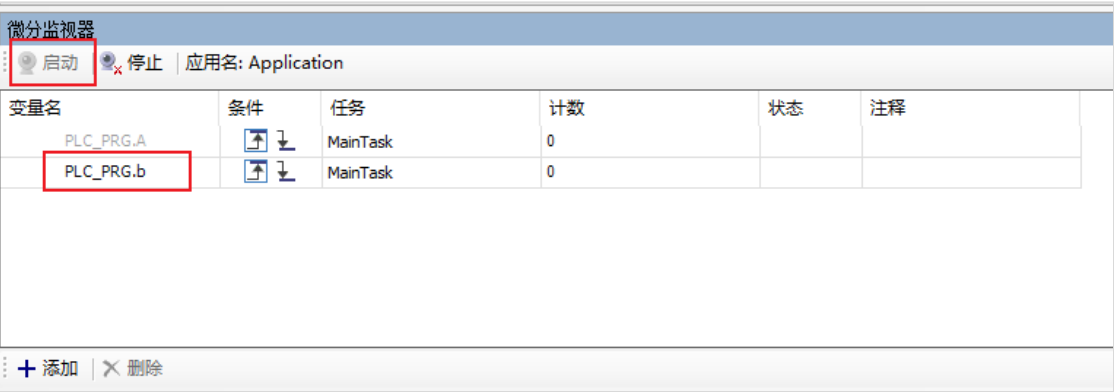
8-8-3. 启动/停止功能说明

1、启动微分监视器

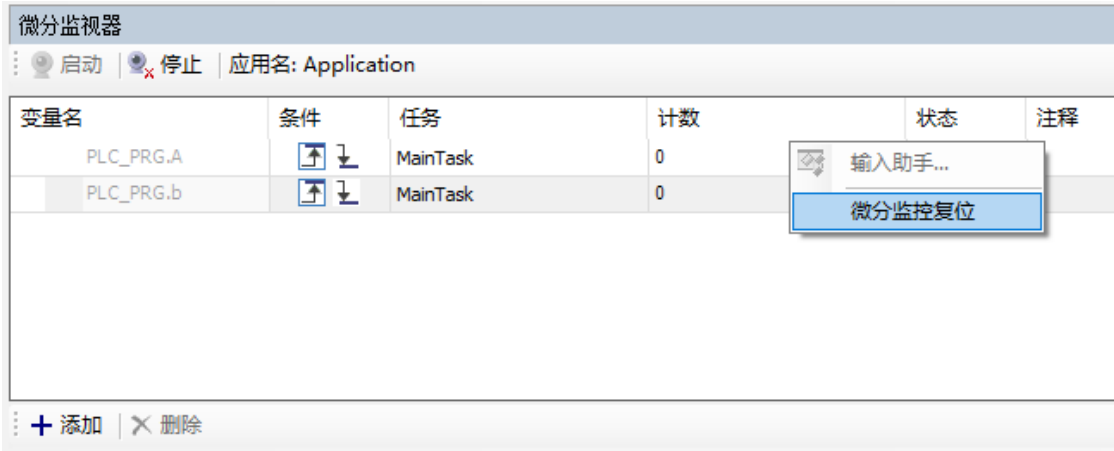
添加变量后单击“启动”，可以启动微分监视器，根据配置记录对应变量的边沿次数。



启动监视后，可以正常监控中的参数，但不支持删除，可以新增变量。但由于新增的变量未启动，故无法监控。

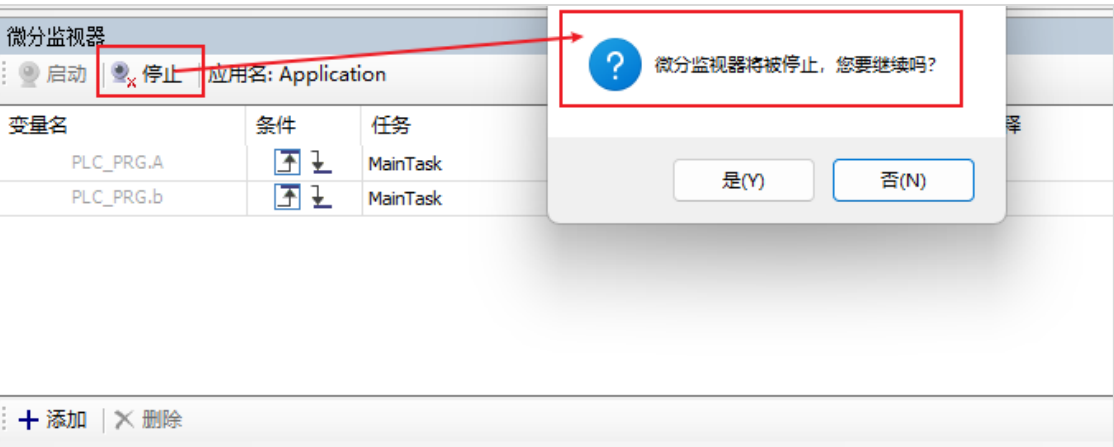


启动微分监视器后，变量的计数值和监控状态可通过“微分监控复位”来清零和复位。



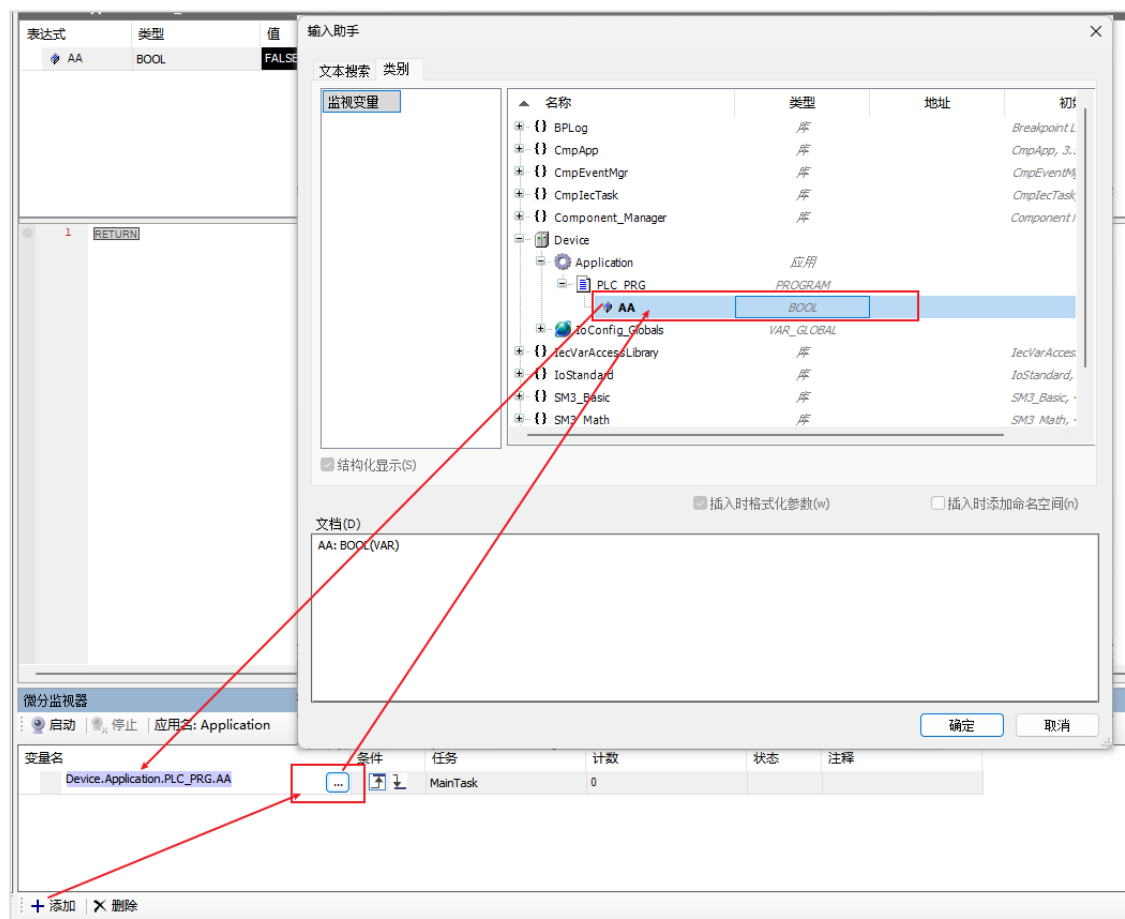
2、停止微分监视器

单击微分监视器中“停止”，会提示是否需要停止，选择“是”后，微分监视器停止监控。PLC 停止运行后，微分监视器也会停止计数。

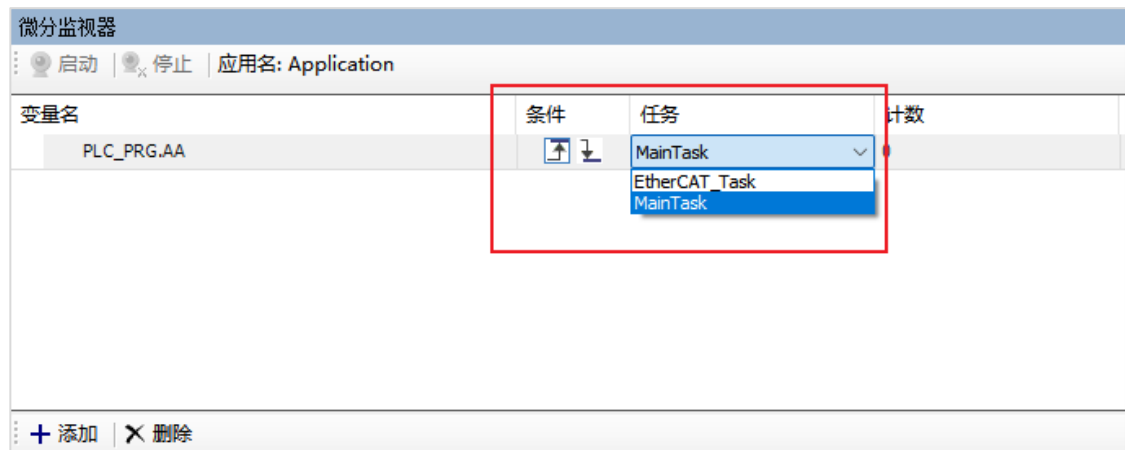


7-8-4. 应用举例

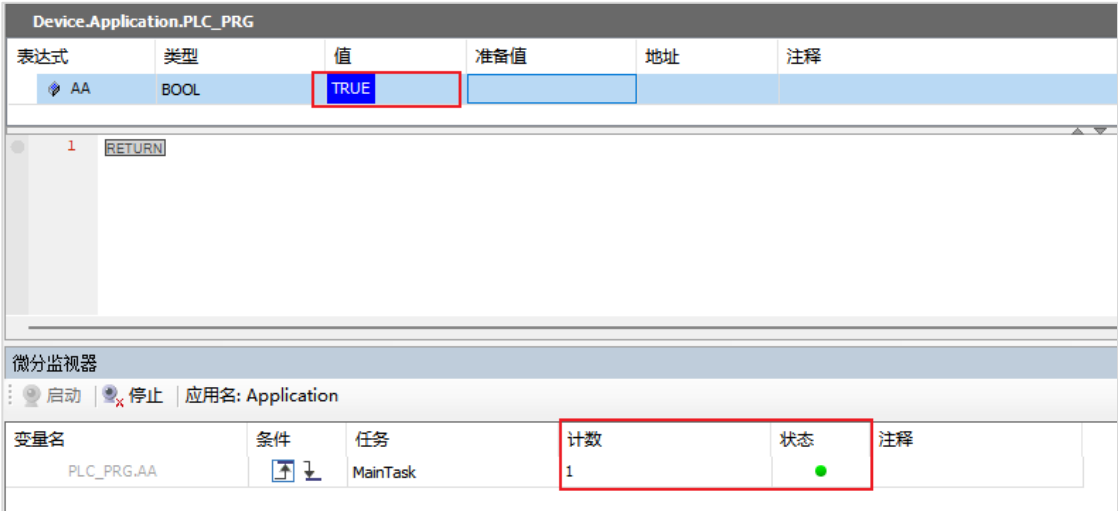
1、双击打开“微分监视器”界面，点击“添加”后界面会显示一行空白数据栏，通过点击 可以选择要监视的 BOOL/BIT 变量。



2、然后在“条件”处选择上升沿或下降沿，在“任务”处配置监视变量挂载的任务。



3、程序登录下载后，点击“启动”后便可以监视变量的上升沿或下降沿次数。



4、若想要将某个变量的当前边沿累计计数清零，可以通过单击选择该变量，点击鼠标右键快捷菜单里的“微分监控复位”来将计数值清零。



8-9. 隐式 POU 检查

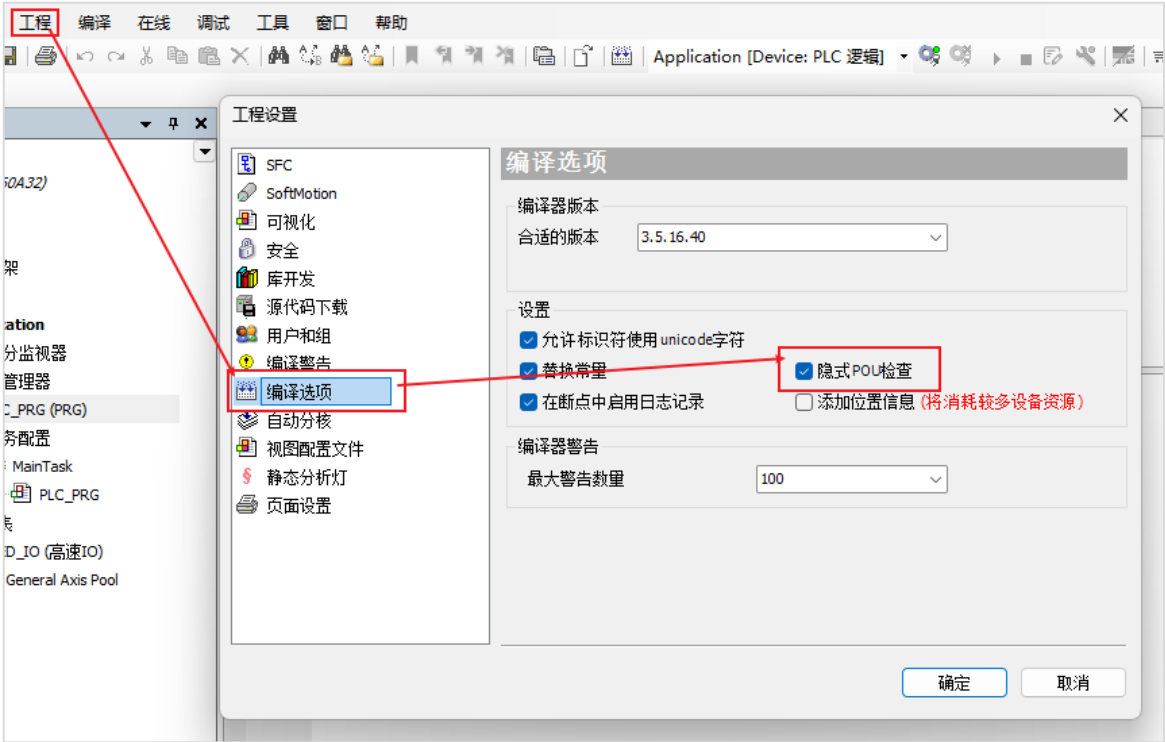
8-9-1. 适用设备

XS 机型及版本 \ 功能	隐式 POU 检查 (XS StudioV2.3.2 及以上支持)
XSDH	固件版本 V2.4.2c 及以上支持
XSLH	固件版本 V2.4.2c 及以上支持
XS F5	固件版本 2.4.2 及以上支持
XSA-W	不支持
XSA-L	固件版本 2.4.2 及以上支持

8-9-2. 功能概述

在“工程”--“编译选项”界面，可开启“隐式 POU 检查”功能，开启此功能后，在程序运行时且在上位机登录的状态下，如果程序中存在数组越界、除零情况时，PLC 不会异常停止，在设备诊断界面提示具体的报错信息，同时做如下处理：

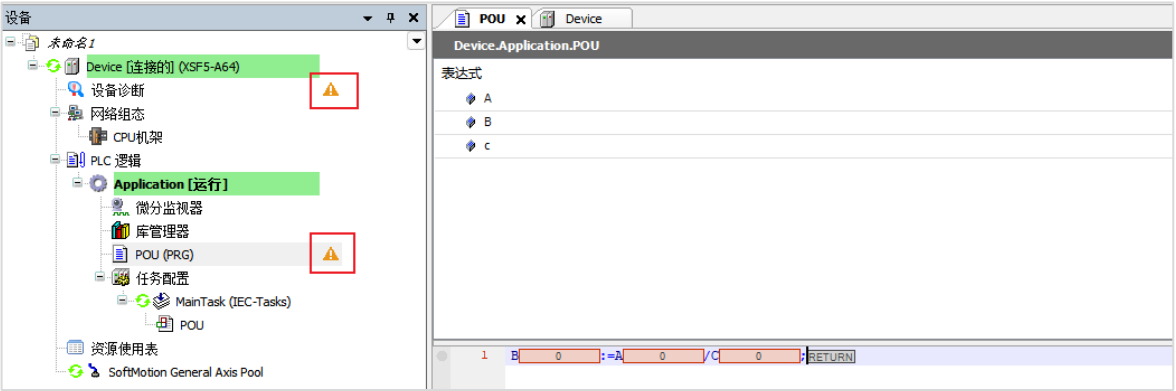
- 当小于数组下标最小值时返回下标最小值，大于数组下标最大值时返回下标最大值。
- 当出现除数为零时，提示除零异常，且返回结果是被除数的值。



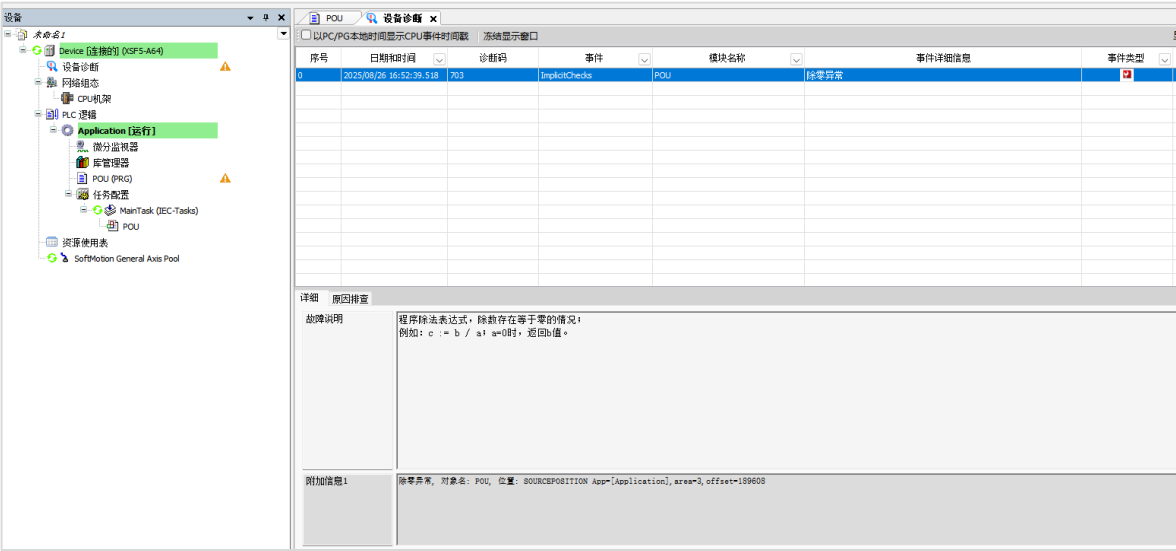
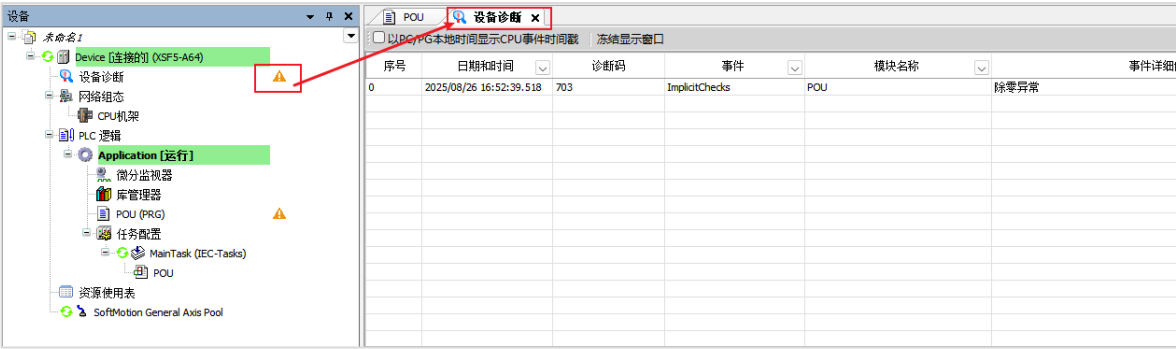
8-9-3. 故障诊断

8-9-3-1. 故障诊断节点显示

出现数组越界和除零异常报错后，设备诊断节点会提示黄色三角感叹号。



双击设备诊断节点或黄色三角感叹标志，会打开故障诊断界面，在此界面显示故障说明以及错误代码位置



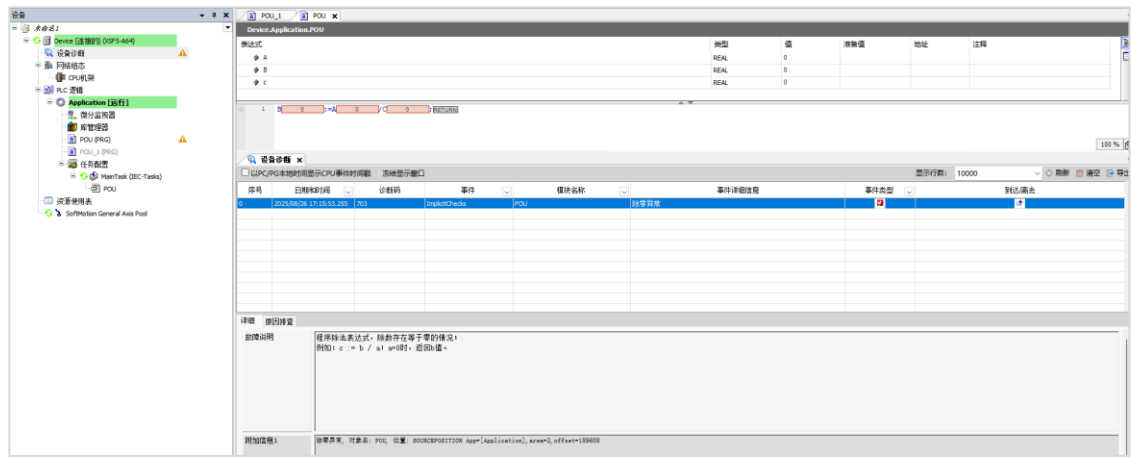
8-9-3-2. 故障诊断界面

故障诊断界面可查看数组越界、除零异常和空指针报错。这 3 个错误默认为 。

在运行过程中如出现上述错误时，如下处理机制：

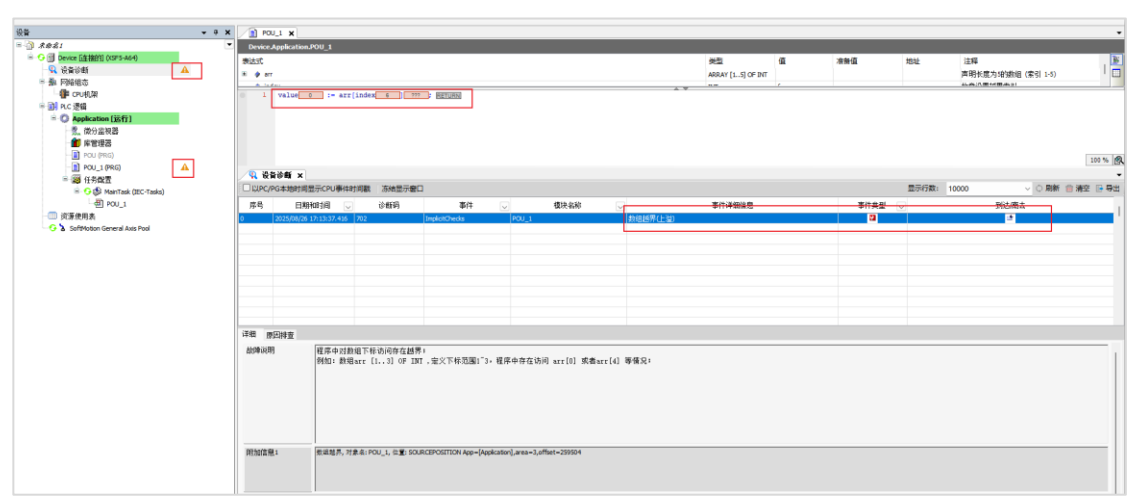
- a：出现除零错误时，可以在设备诊断和日志中查看相关报错信息。

除零报错如下示例，此时设备处于运行状态：

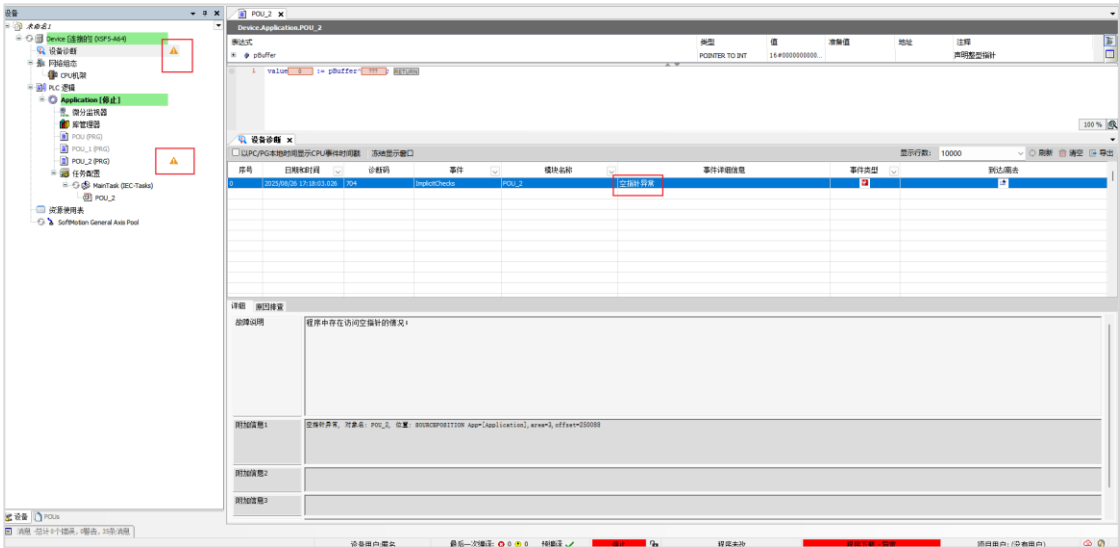


- b：出现数组越界时，当数组索引小于数组最小索引时，会将最小索引返回给数组，使其获取最小索引对应的数据；当数组索引大于数组最大索引时，会将最大索引返回给数组，使其获取最大索引对应的数据。

数组越界报错如下示例，此时设备处于运行状态：



- c：空指针报错时直接停机，可以在设备诊断和日志中查看相关报错信息。



9. 资源使用表

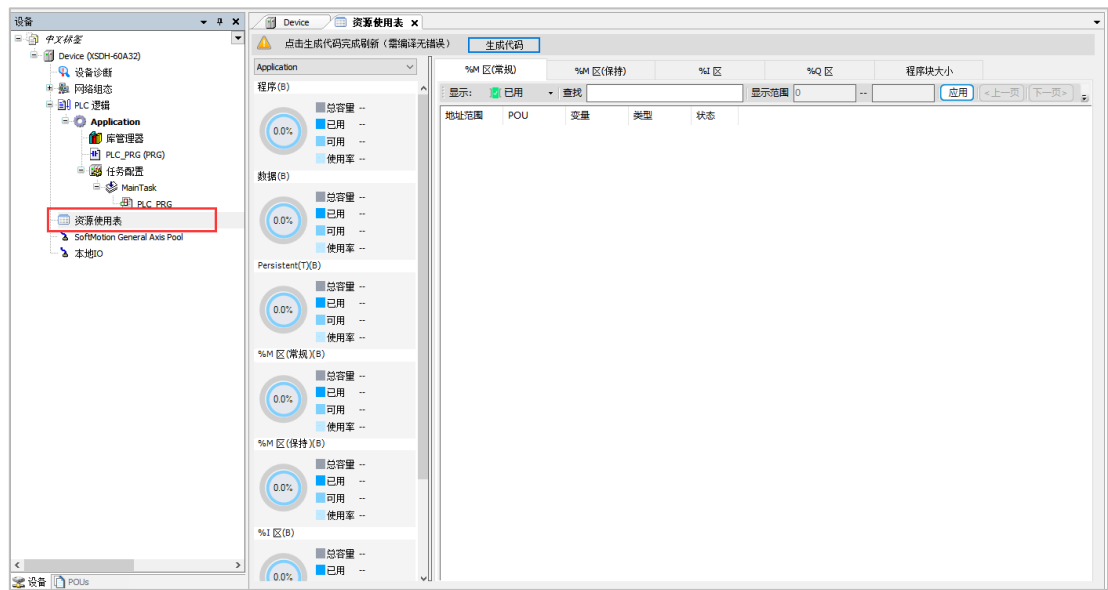
9-1. 资源使用表

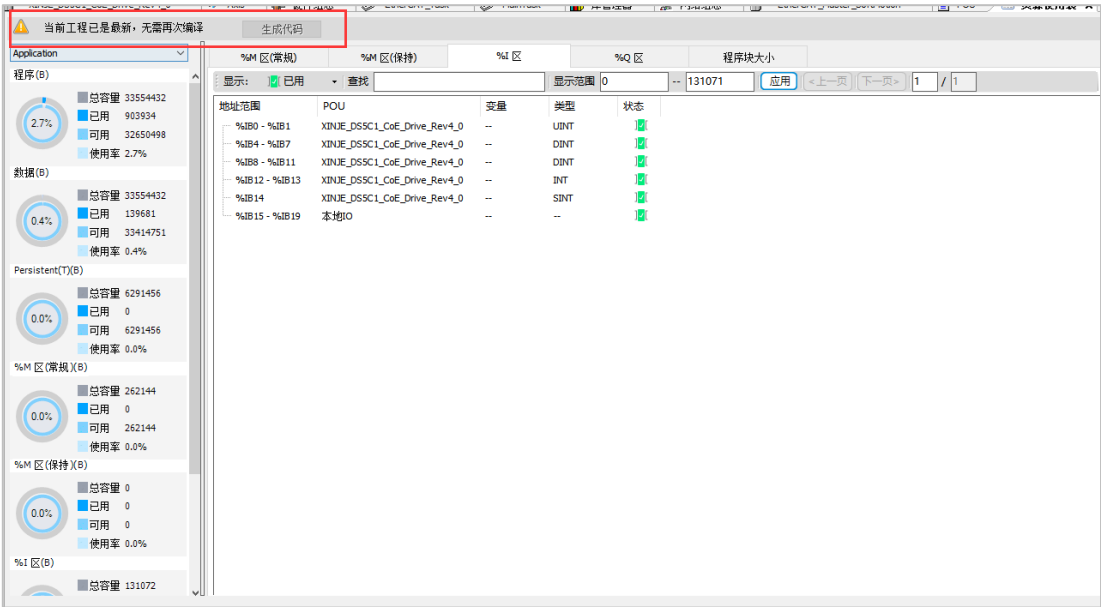
资源使用表用于显示 I/Q/M 区地址使用情况，程序区、数据区、掉电保持区、M（常规）/M（保持）/I/Q 区域的使用大小、可用大小及使用率大小。

9-2. 功能介绍

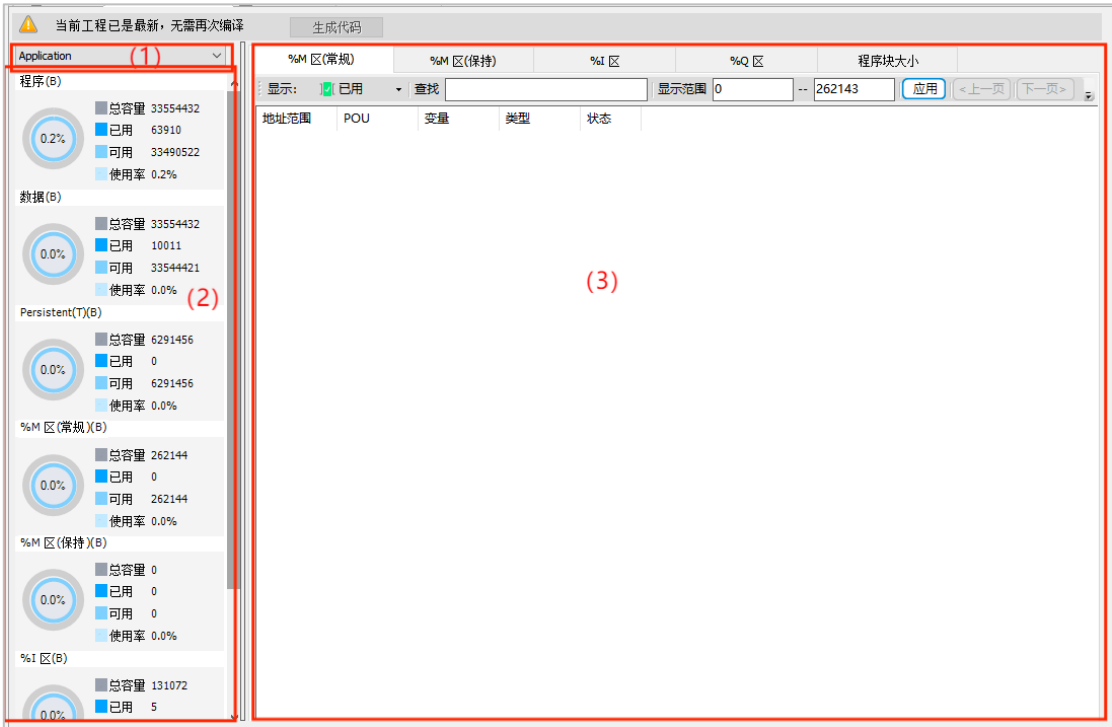
9-2-1. 更新资源使用表

新建 XS studio 工程后，可通过双击软件左侧设备树上资源使用表节点，打开资源使用表界面，点击界面上生成代码，会根据生成的数据刷新界面。若工程发生变更刷新资源使用时，需要重新生成代码。





9-2-2. 界面功能介绍

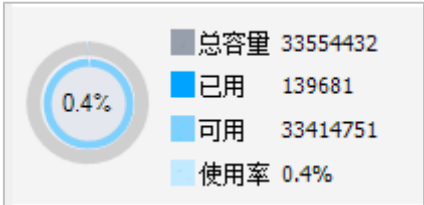


1、应用程序选择（即界面的 Application）

用于工程中存在多个应用时，查看不同应用程序所占用资源，默认显示激活的应用。

2、区域使用信息

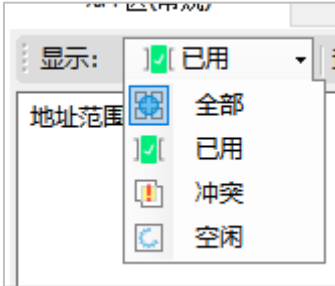
主要显示程序区、数据区、掉电保持区、M(常规)/M(保持)/I/Q 区的基本信息，包含区域总容量、已用区大小、可用区大小及区域使用率大小。



3、I/Q/M 详细使用信息及程序块信息显示

主要显示 M(常规)/M(保持)/I/Q 区地址的使用情况；程序块大小信息包括程序中调用的功能块类型及大小信息。
该界面一共显示分为 5 个显示选项，分别是%M 区（常规）、%M 区（保持）、%I 区、%Q 区和程序块大小，分别显示对应区域的使用信息。其菜单选项中包含：显示、查找、显示范围、应用按钮和翻页选项。

① 【显示选项】：在 M 区、I 区、Q 区中，选项内容为：全部、已用、冲突、空闲；



在程序块大小显示界面，选项内容为：全部、结构体、联合体和功能块。



当程序工程中存在占用内存空间的结构体、联合体或功能块时，显示窗口才可选择，否则下拉框中不存在该选项。

② 【查找】：M 区、I 区、Q 区可对地址范围、POU、变量、类型进行查找；程序块大小可对名称、类型进行查找。

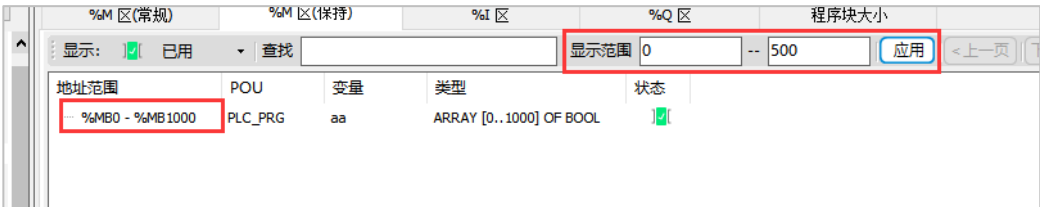


查找按照字符进行非全字匹配，不区分大小写，查找选项中内容会改变筛选显示。

③ 【显示范围】：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。显示需要点击应用选项，才可生效。程序块界面没有显示范围设置。



如果显示范围区域小于使用的变量范围，地址范围会显示实际使用的范围。



④ 【应用】：系统会根据显示范围进行数据筛选显示。



⑤ 【翻页】：表格默认最多添加 1000 个地址段单元，如果数据超过 1000 个地址段，采用分页显示，可通过点击上/下页或是输入指定页码项，跳转到对应页面。

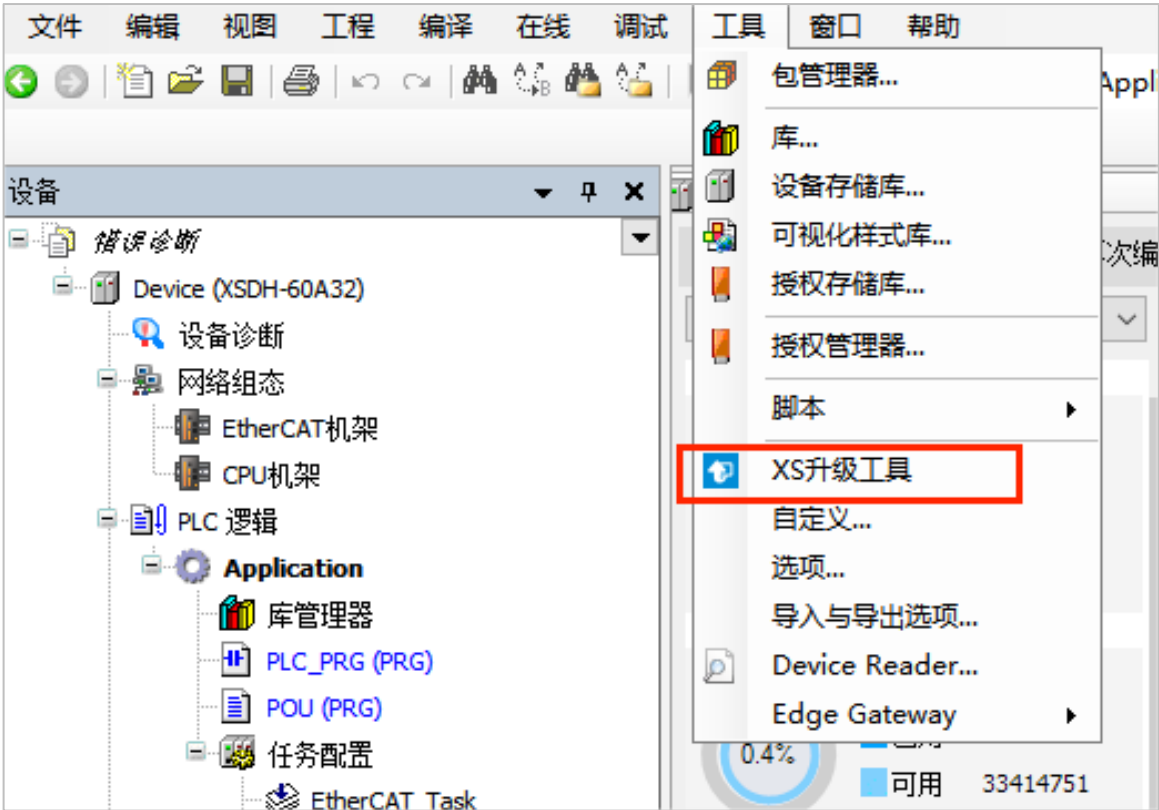


⑥ M(常规)/M(保持)/I/Q 区表格显示：

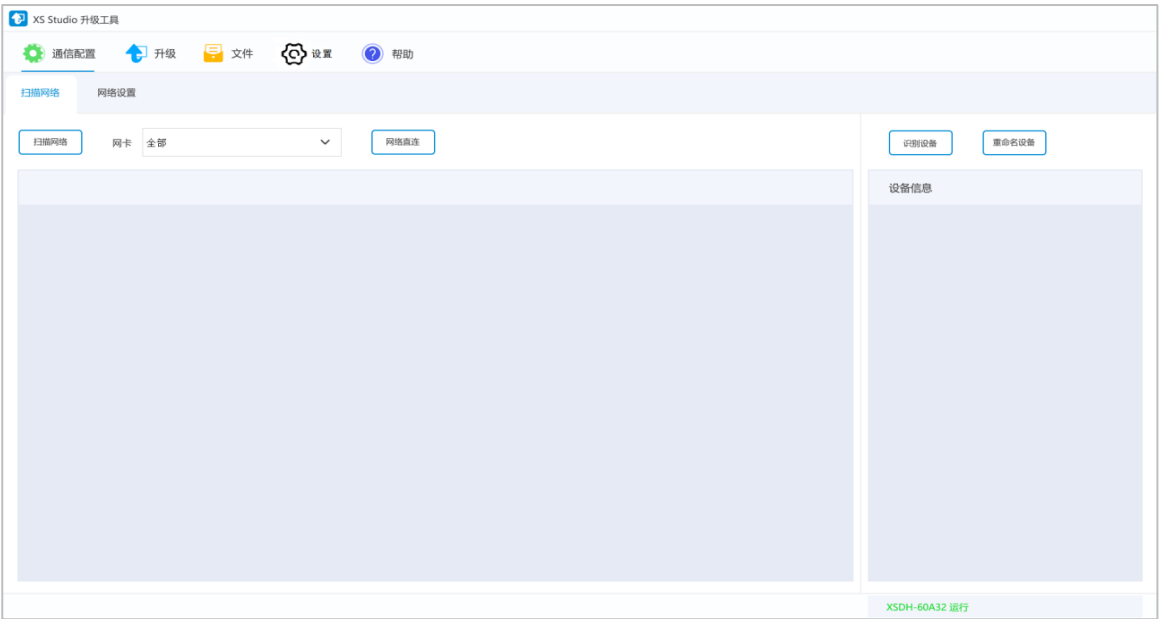
表格中主要根据筛选条件显示符合条件的地址段使用信息。

地址范围	显示变量的整个地址区间。
POU	显示变量所属的 POU 名称。
变量	显示该区域关联的变量名称。
类型	显示变量类型。
状态	显示地址区的使用情况，已用时图标为  ，冲突为  ，空闲下为  。 【注】冲突按照 bit 检测。当用户定义变量与设备分配 IO 地址冲突时，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问题点。

10. XS Studio 升级工具



XS 升级工具包含通信配置、升级、文件、设置、帮助五大功能。打开编程软件后，在工具菜单栏里打开 XS 升级工具。

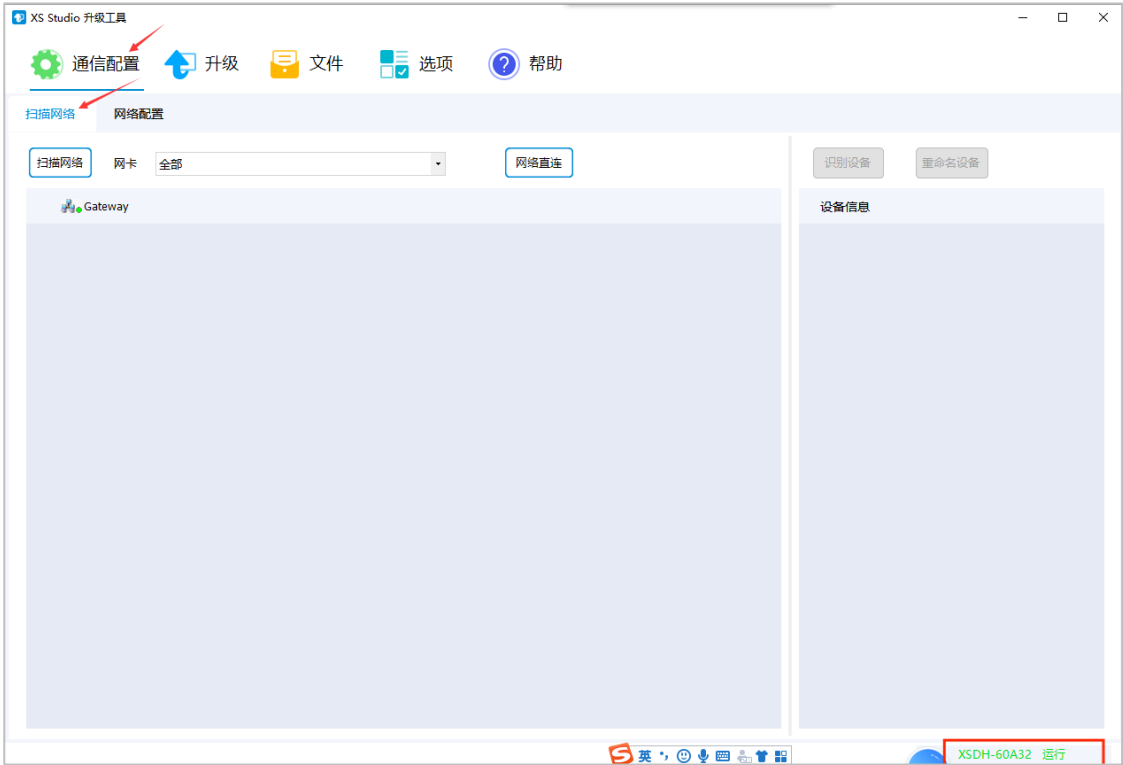


10-1. 通信配置

通信配置是用户使用工具软件的第一步，具体包括扫描网络和网络配置两个功能。用户可以通过这两个功能连接主机所有同网段下的 PLC 设备，并对 PLC 设备进行 IP 设置。

10-1-1. 联机

点击“通信配置”->“扫描网络”，可以扫描和连接所在网段下的 PLC 设备，如下图所示：

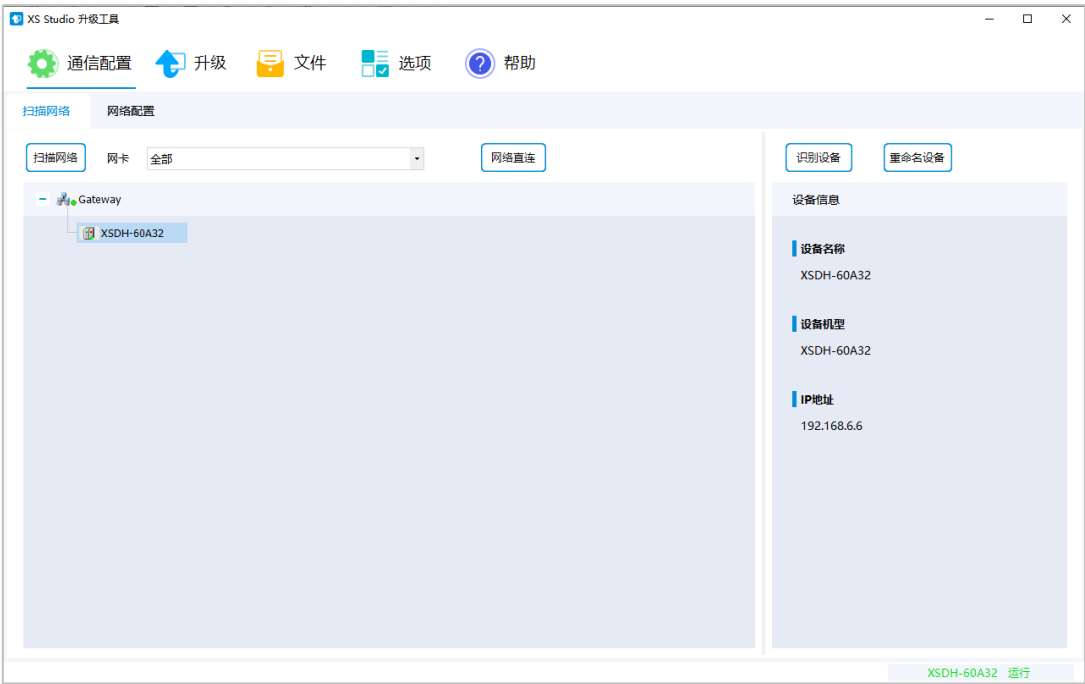


设备连接之后会在右下角状态栏显示设备连接信息，包括当前连接的设备名称及状态。

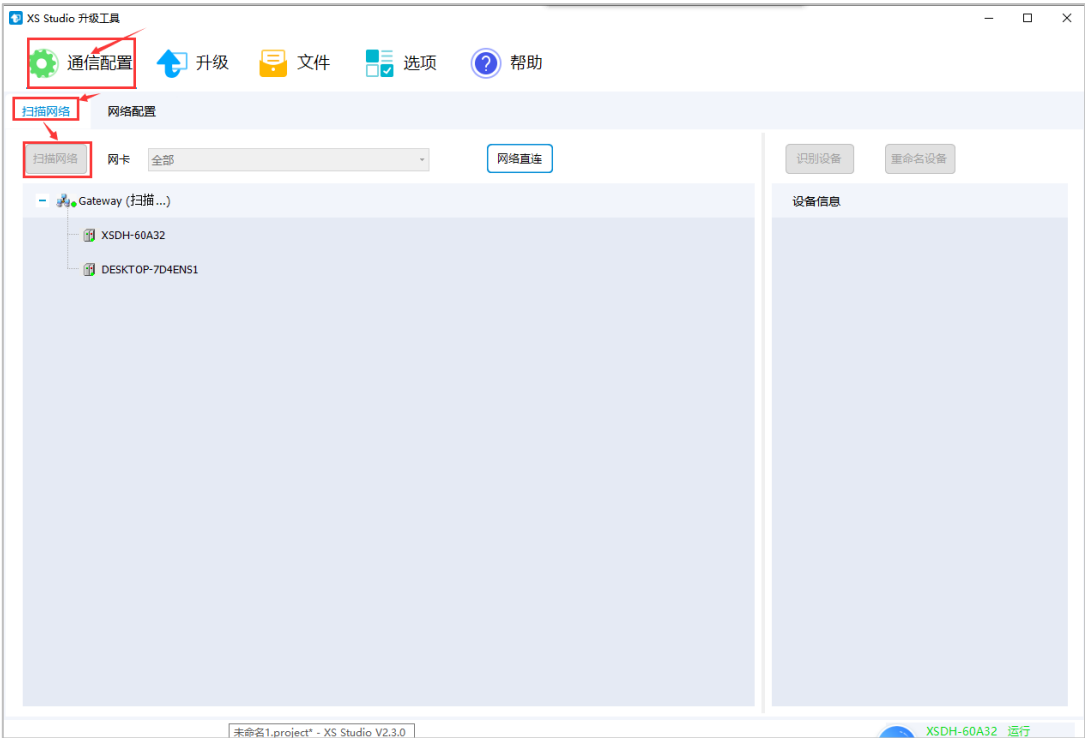
连接设备状态	状态栏显示	
未连接	红色字符标识：未连接设备	未连接设备
运行状态	绿色字符标识：设备名称，绿色字符标识：运行	XSDH-60A32 运行
停止状态	绿色字符标识：设备名称，红色字符标识：停止	XSDH-60A32 停止
err 状态	绿色字符标识：设备名称，红色字符标识：异常	XSDH-60A32 异常

10-1-1-1. 扫描网络

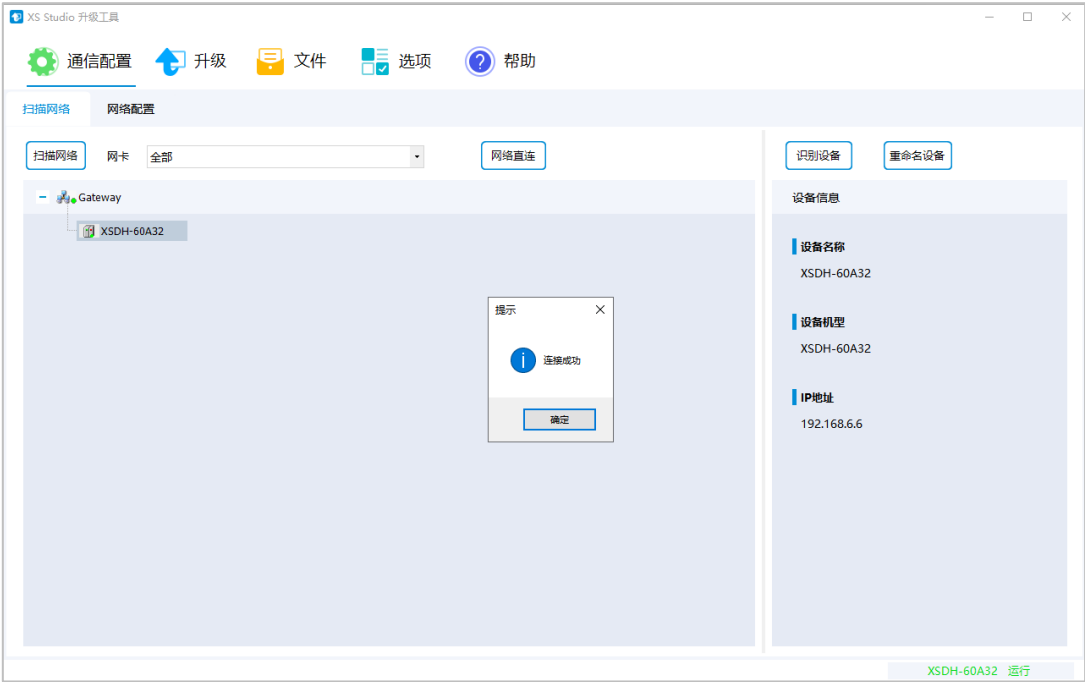
点击“扫描网络”，可以根据网卡选择显示联机的所有网段或者单个网卡下的 PLC 设备。



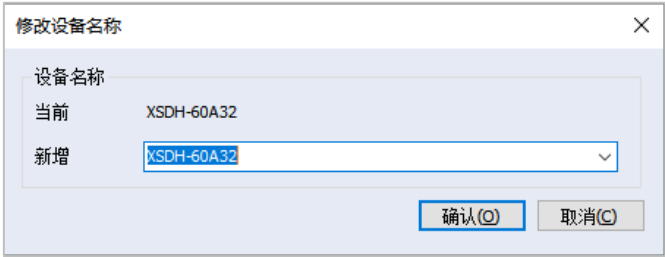
在设备树里单击 PLC 设备，该机型会变成选中状态，右侧设备信息会显示对应设备的基本信息，包括设备名称、设备机型、IP 地址。



双击被选中设备，上位机和设备成功连接。点击“识别设备”，已连接机型的 err 灯会闪烁显示。

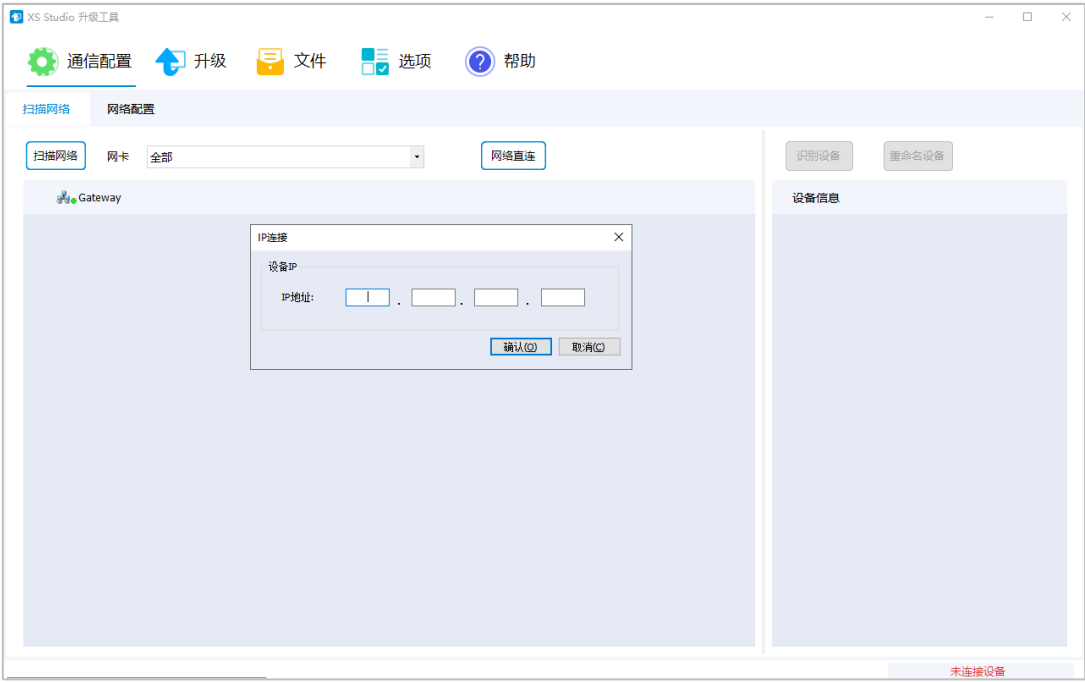


点击“重命名设备”可以修改对应的设备名称，用于区分不同的 PLC 设备。



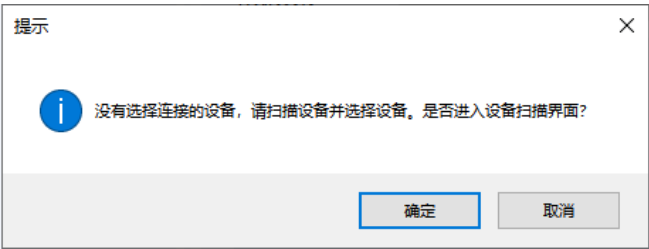
10-1-1-2. 网络直连

点击网络直连，输入 IP 地址直接连接对应设备。



注意：

未连接设备时，点击扫描网络、网络直连、选择 xxx 升级包之外的其他按钮，会提示报错。



10-1-2. 网络设置

点击“通信配置”->“网络配置”，可以读写所连接 PLC 设备的 IP 地址、网关等信息，设置界面如下：



选择对应的网口，点击“读取”，读取所连接设备的 IP 地址、子网掩码和网关，同时提示“读取成功”。



也可以通过该界面写入需要设置的 IP 地址，并将设备断电重启后生效。



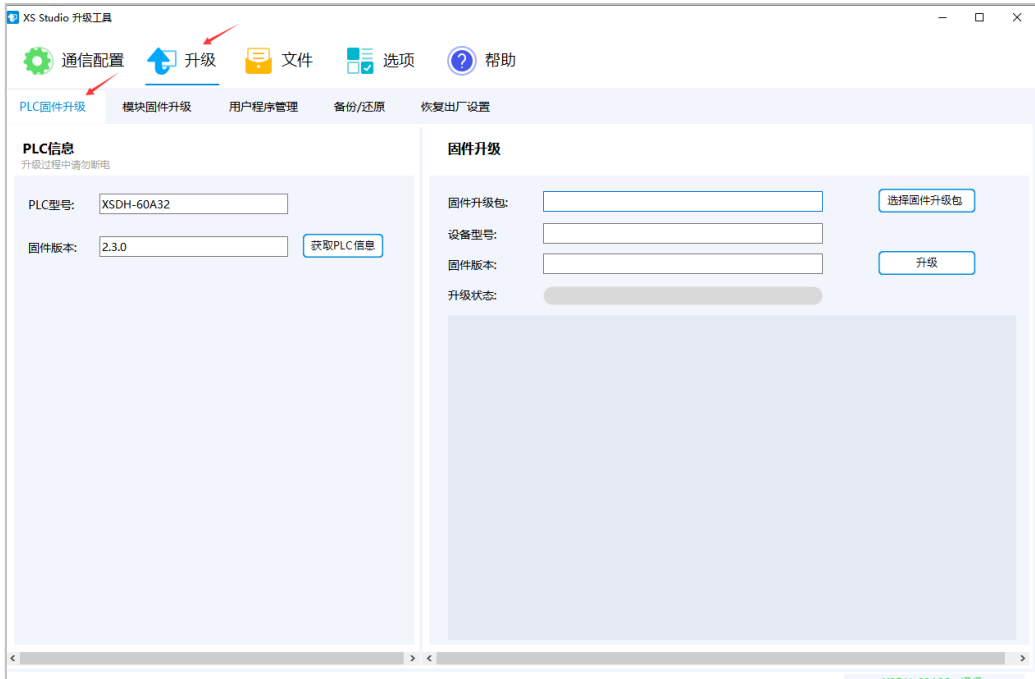
10-2. 升级

“升级”菜单可以对 XSDH/XSLH/XSF5 系列 PLC 进行 PLC 固件升级、模块固件升级、用户程序管理、备份/还原和恢复出厂设置五个功能。

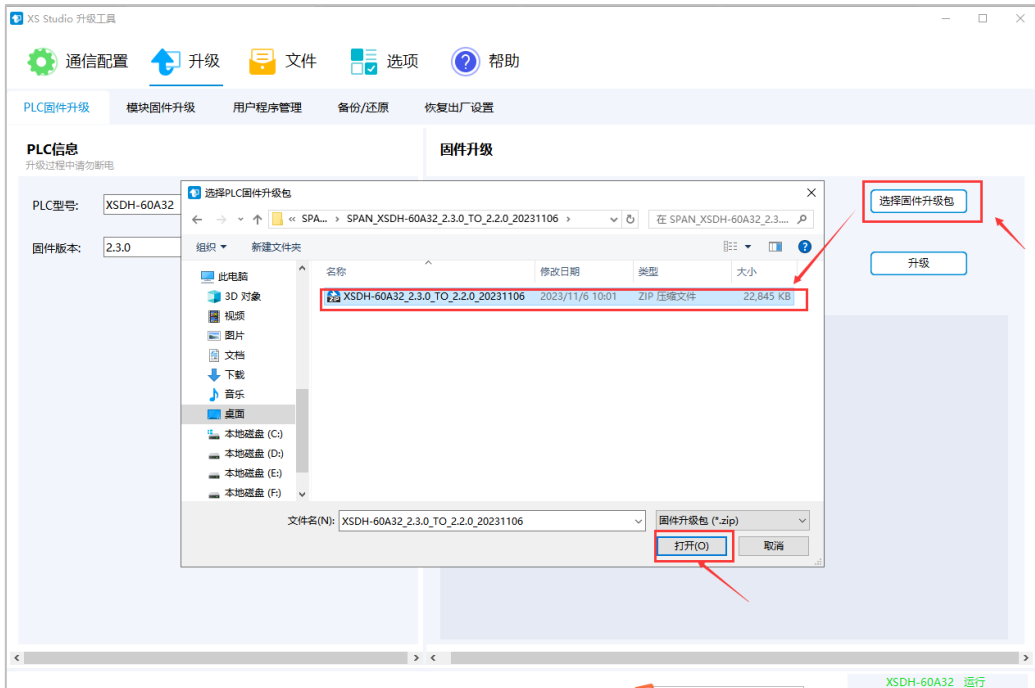
10-2-1. PLC 固件升级

该功能可以实现 PLC 的固件升级。成功连接 PLC 设备之后，单击“升级”→“PLC 固件升级”，进入 PLC 固件升级界面。

点击“获取 PLC 信息”，左侧窗口将显示所连接设备的 PLC 型号和固件版本，如下图所示：

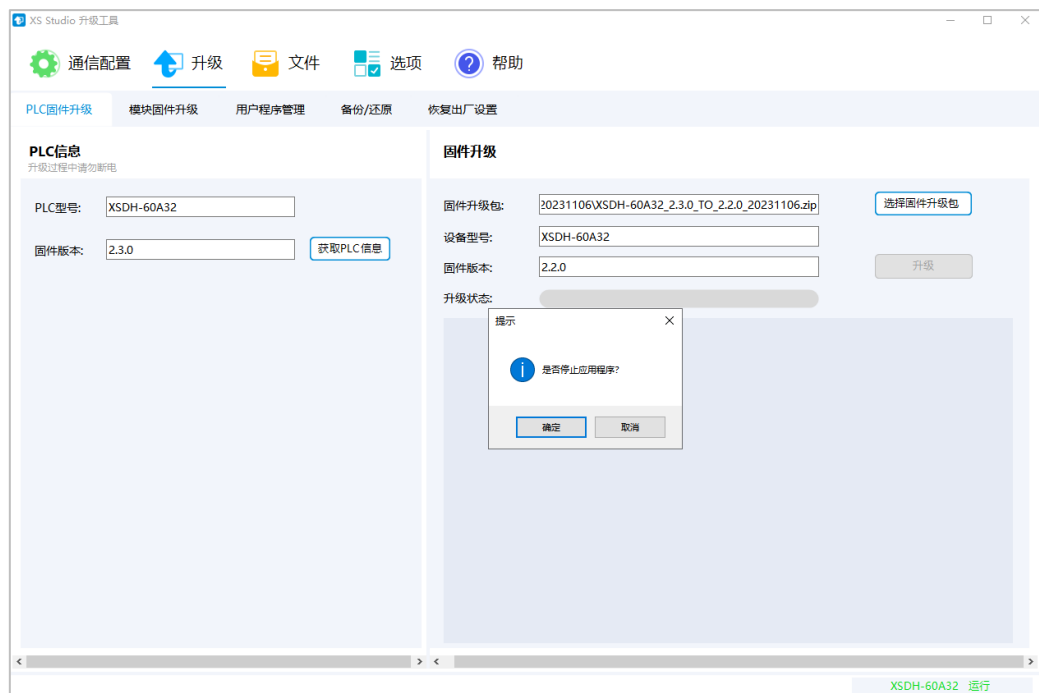


单击“选择固件升级包”，弹出文件选择窗口，选择相应的升级包文件，点击“打开”，如下图所示：





单击“升级”，首先确认连接机型是否处于停止状态，如果是运行状态会提示是否停止应用程序，如果点击取消则停止升级；如果点击确认继续升级。



同时会确认升级包与在线设备兼容性，若不兼容则停止升级，信息框输出具体信息，若兼容则会对所连接设备进行固件升级。

固件升级

固件升级包:

C:\Users\XinJe\Desktop\SPAN_XSDH-60A32_2.3.0_TO_

选择固件升级包

设备型号:

XSDH-60A32

固件版本:

2.2.0

升级

升级状态:

[2023.11.21 14:31:16] 停止应用程序

[2023.11.21 14:31:16] 所选固件升级包和当前机型版本相匹配

[2023.11.21 14:31:16] 下载中...

[2023.11.21 14:31:27] 下载成功!

[2023.11.21 14:31:28] 升级中...

[2023.11.21 14:31:41] 升级成功!

[2023.11.21 14:31:41] 设备将会自动重启, 请不要断电。

===== End =====

-  **注意:**
- 升级过程中, 设备状态同 U 盘/SD 卡升级固件相同, **err** 灯闪烁代表升级中, **err** 灯熄灭代表升级成功, 升级成功后设备处于 **run** 状态, 并以升级后的固件版本运行 (见下表)。
 - 设备升级过程中, 不可将软件切换至其他界面。

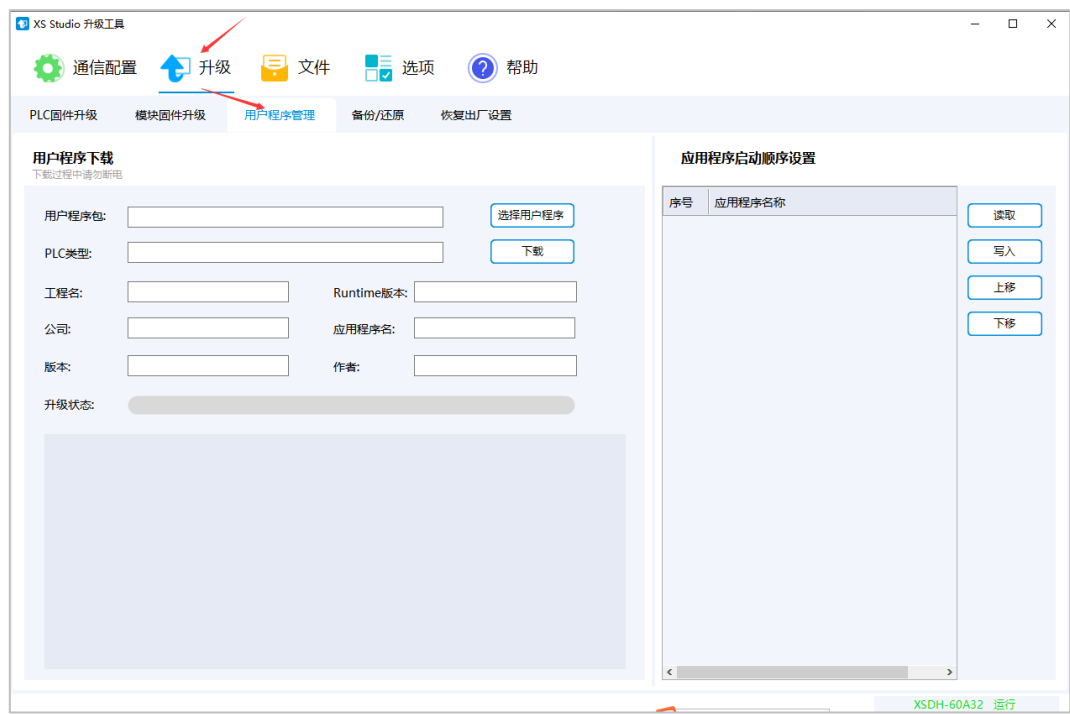
指示灯	灯状态	状态描述
ERR	闪烁	固件升级中
	灭	升级成功, 设备以新固件运行
	常亮	升级失败, 设备以刷机前固件运行

10-2-2. 模块固件升级

暂不支持此功能。

10-2-3. 用户程序管理

该功能可实现用户程序的下载; 成功连接 PLC 设备之后, 单击“升级”→“用户程序管理”, 进入用户程序管理界面, 如下图所示:



单击“选择用户程序”，弹出文件选择窗口，选择相应的用户程序包（需要在软件编译-打包用户程序生成对应文件），获取对应用户程序包的路径信息和用户程序包对应的 PLC 类型、Runtime 版本、工程名、应用程序名等信息，并显示在左侧窗口中，工程包中的描述信息显示在左下方的信息框中；



之后单击“下载”按钮，确认用户程序与在线设备是否兼容，若不兼容则停止下载，信息框中输出具体信息；若兼容则将应用程序包下载到所连接设备中；下载成功后需要重新连接该设备。



右侧“应用程序启动顺序设置”窗口中的“读取”按钮可以读取当前在线 PLC 中的应用程序。例如 PLC 里运行的工程存在多个 **Application** 应用程序时会读取到多个，上移/下移用于操作上下移动应用程序的顺序，修改顺序后可写入。



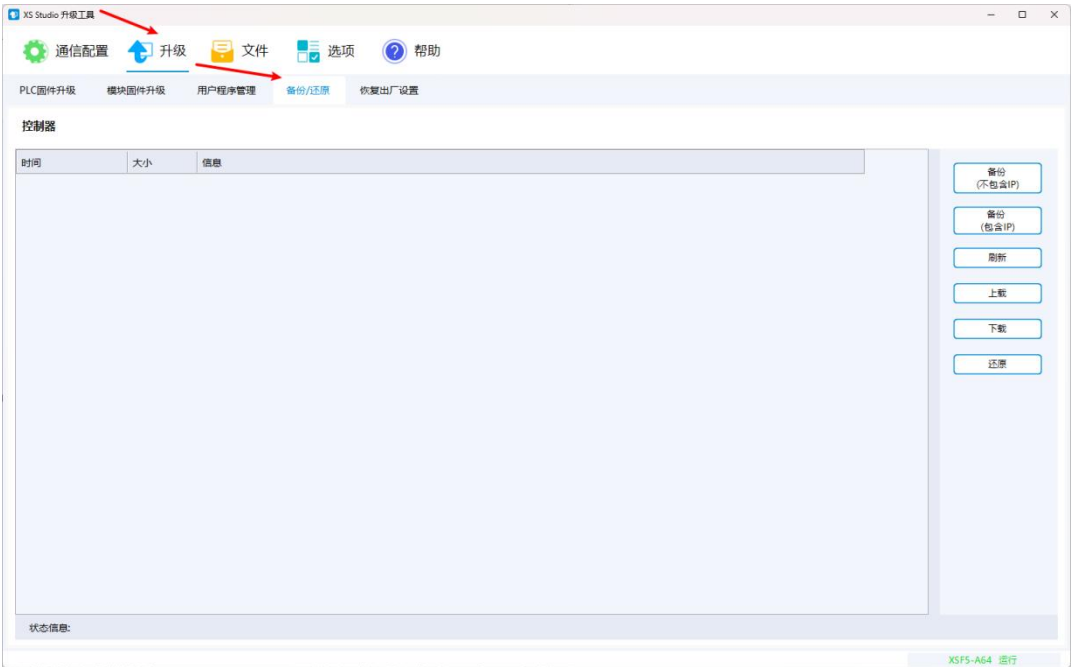
10-2-4. 备份/还原

该功能是对当前控制器中用户程序和数据进行备份和还原。

备份：设备中只允许存在一份备份文件，若已经备份文件，点击备份会替换掉之前的备份文件。

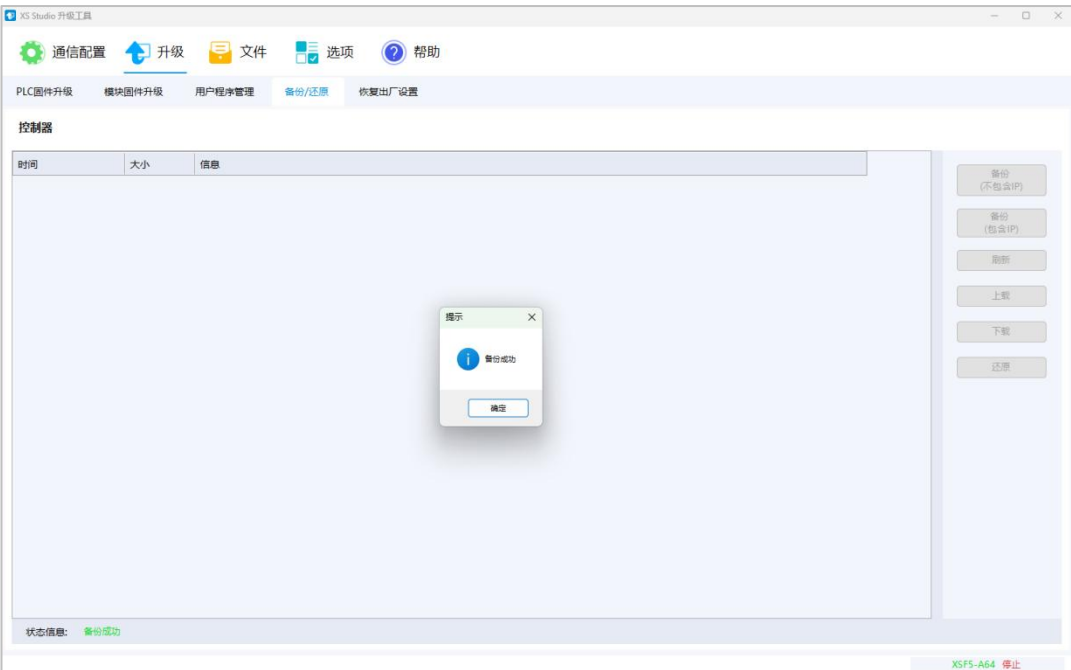
还原：可以还原所选用户程序备份文件类型；需要在先获取备份文件并选择备份文件方可实现。

成功连接设备之后，单击功能菜单框里升级，选择备份/还原，进入备份/还原界面。

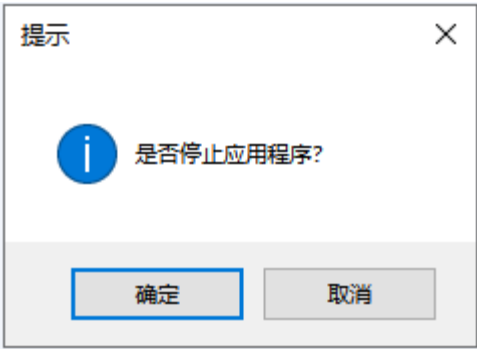


1、备份

点击“备份”，可对当前控制器中的用户程序和数据进行备份。界面和状态信息将提示“备份成功”。可选择是否包含 IP 进行备份。

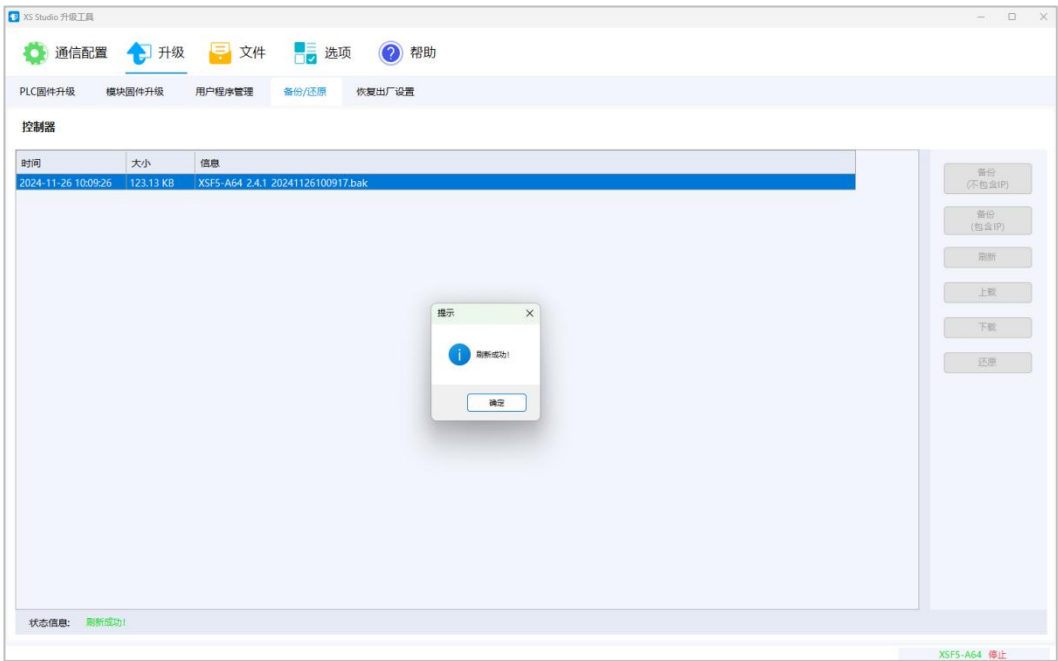


备份需要在 PLC 停止状态下进行，如果在 PLC 运行状态下点击“备份”，会提示“是否停止应用程序？”，点击“确定”，PLC 停止并进行备份，点击“取消”，则取消备份。



2、刷新

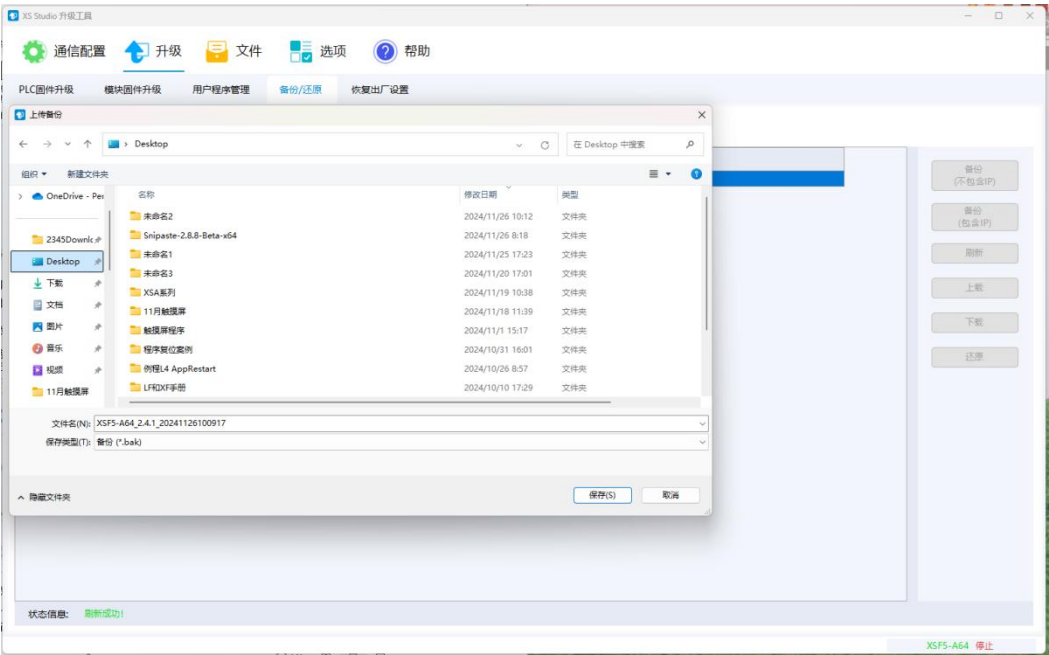
备份之后，点击“刷新”，控制器窗口中将显示连接设备的备份文件名称、大小等信息，同时状态栏显示“刷新成功!”，如下图所示：



刷新会显示该升级工具生成的备份文件，Codesys 原生备份生成的文件不支持显示。

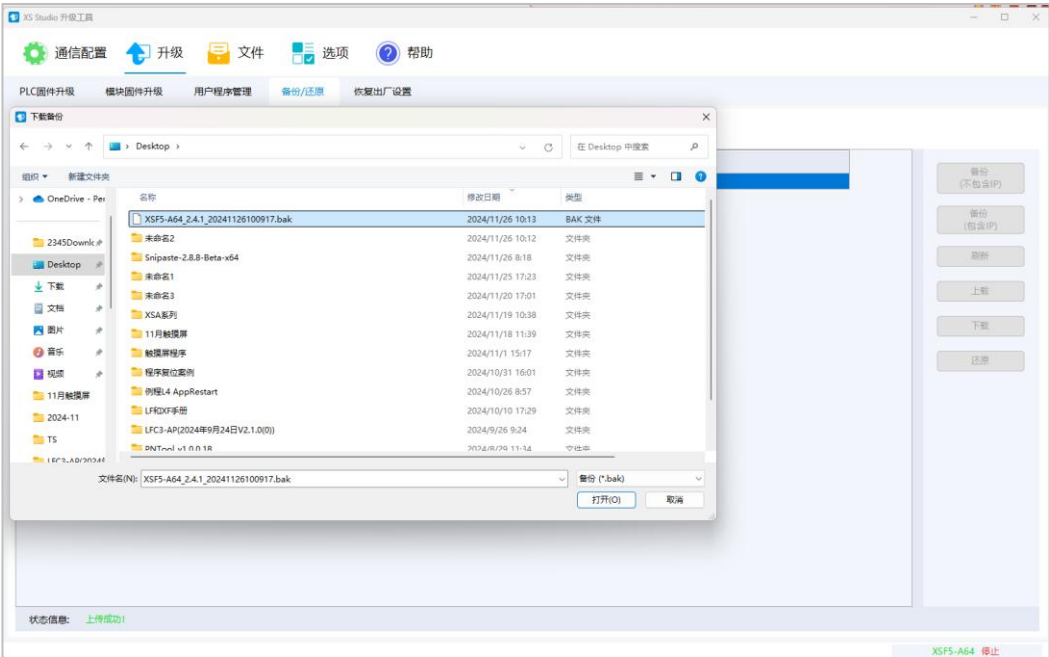
3、上载

刷新之后，选中对应的备份文件，点击“上载”，弹出文件保存窗口，选择文件需要保存的地址，点击“保存”，显示“上载成功”。

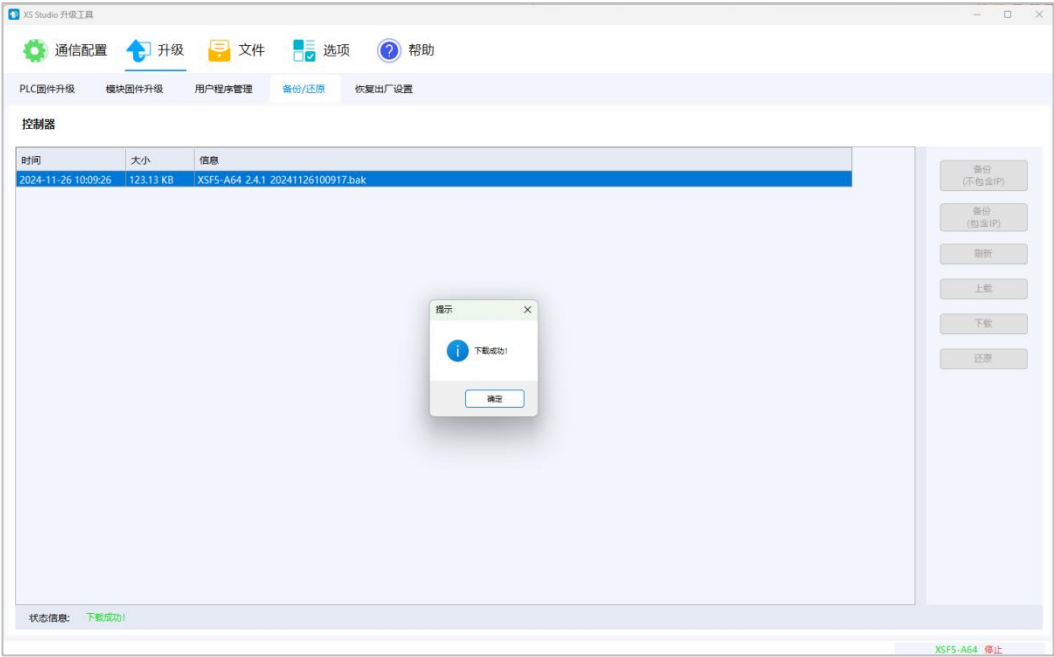


4、下载

“下载”功能可将备份文件下载到所连接的设备中。点击“下载”，弹出文件选择窗口，选择需要的备份文件，并进行下载。选择的备份文件需要与所连接设备类型相匹配，否则会提示下载失败。

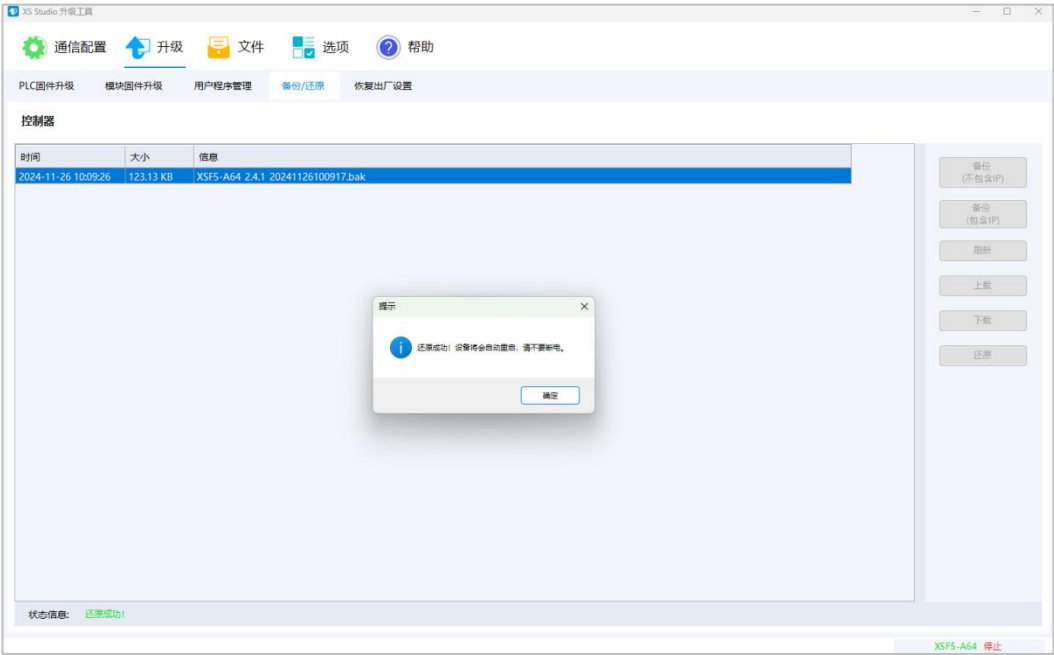


下载成功之后，提示“下载成功!”，并在状态栏中显示。导入下载的备份文件需点击刷新显示。

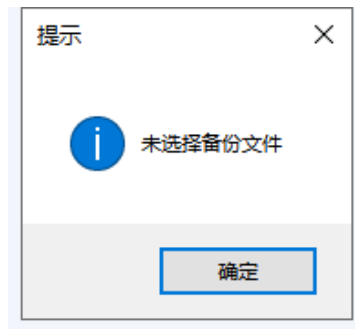


5、还原

文件下载刷新之后，勾选对应的备份文件，点击“还原”，设备将运行选择还原的备份程序，还原成功之后，提示还原成功。

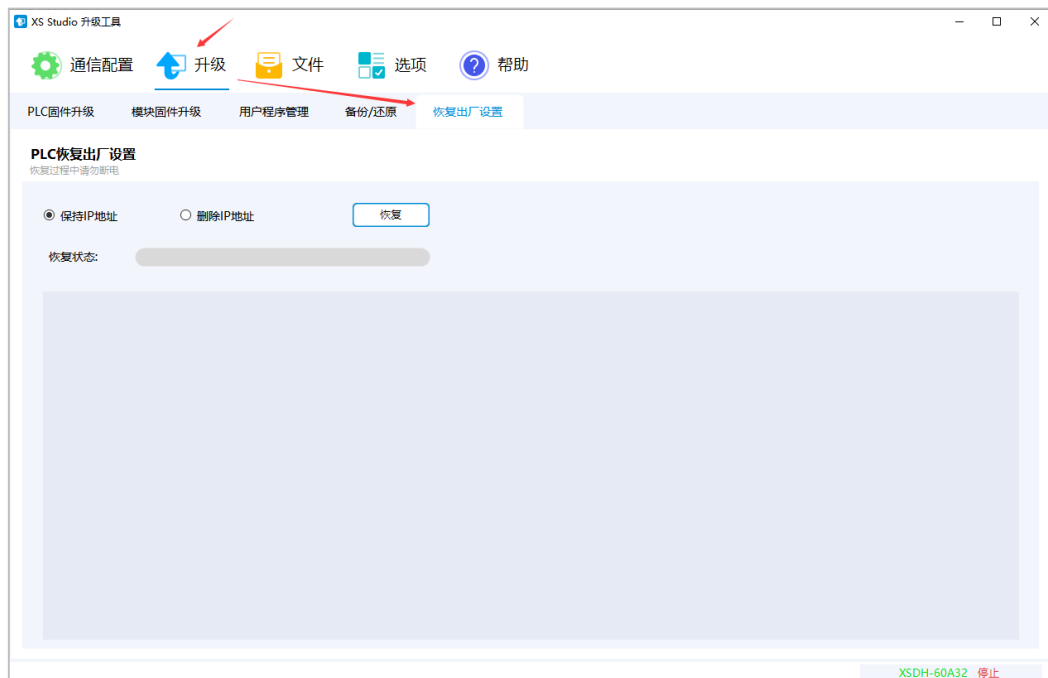


- 如果信息界面为空，会提示“未选择备份文件”，需要点击刷新显示备份文件。
- PLC 的固件在 2.3.0 以上支持备份/还原功能。
- 备份文件自动运行
 - ①通过 U 盘/SD 卡存储备份文件->U 盘/SD 卡插到控制器->控制器断上电运行备份文件；
 - ②将备份文件放到 PLC 的 APP 目录下->控制器断上电运行备份文件。



10-2-5. 恢复出厂设置

恢复出厂设置功能可以将控制器初始化，包括删除用户工程、数据、文件等。
成功连接设备之后，单击“升级”→“恢复出厂设置”，进入恢复出厂设置界面。



勾选“保持 IP 地址”，再点击“恢复”，设备 IP 不清除；选择“删除 IP 地址”，点击“恢复”，删除设备 IP，并恢复为默认 IP（192.168.6.6）。恢复成功之后，需要重新连接设备。



- 恢复过程中，PLC 不能断电。
- 2.3.0 以下固件版本不支持保持 IP 地址功能。

10-3. 文件

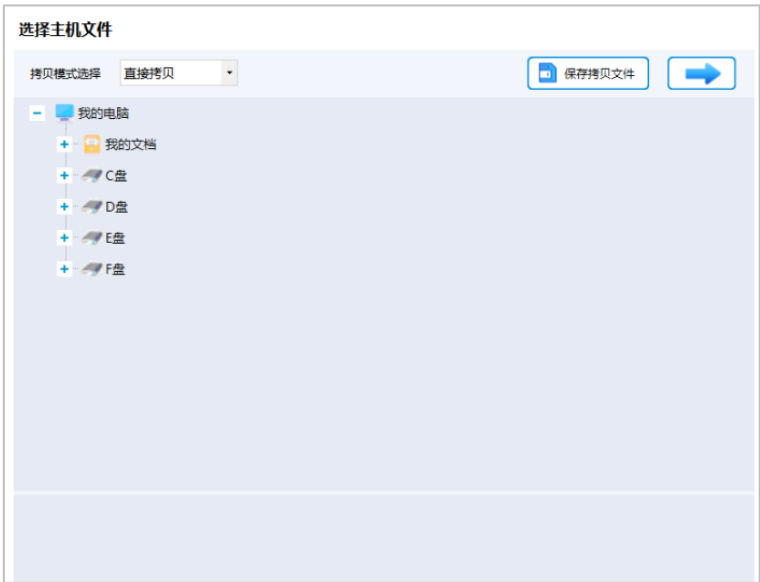
文件功能实现电脑客户端和 PLC 设备端进行一些文件拷贝操作。

10-3-1. 文件传输

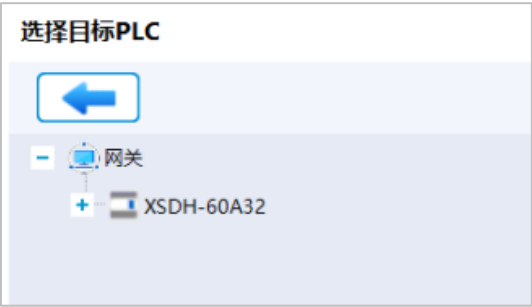
文件传输功能可以实现电脑和控制器之间的文件拷贝互传。
单击“文件”→“文件传输”，进入文件功能界面。



“选择主机文件”窗口中，可以将文件或文件夹传输到 PLC 里。



“选择目标 PLC”窗口中，可以显示网关中所有在线的设备型号。

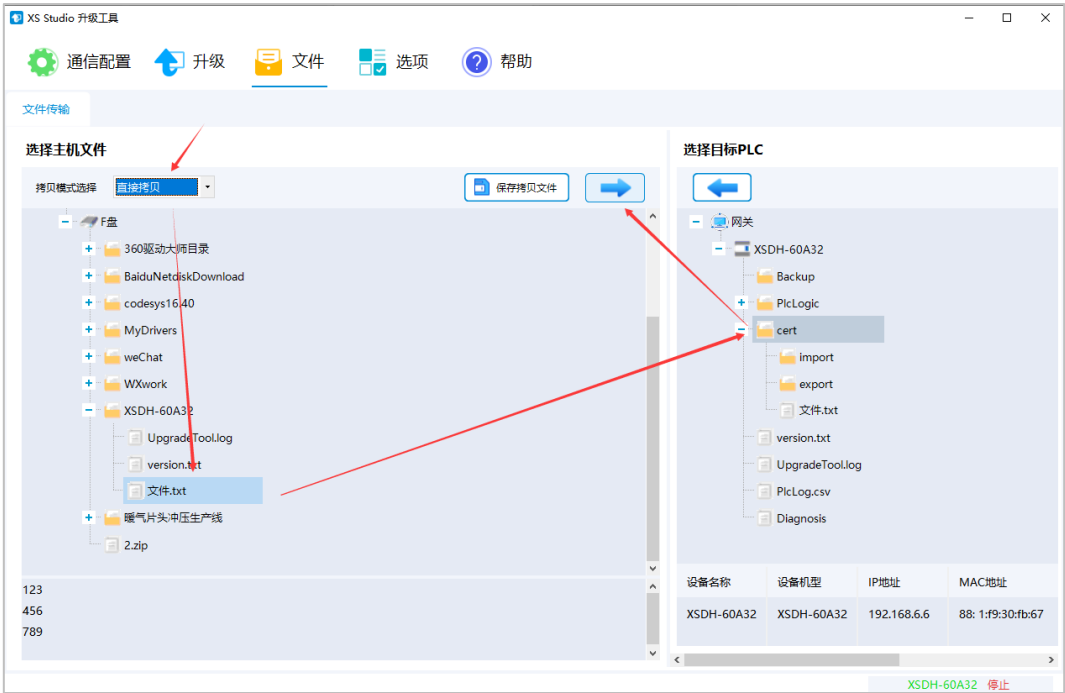


在主机传输时，选择相应的 PLC，在界面右下角 PLC 信息里会显示该设备名称、设备机型、IP 地址和 MAC 地址。

设备名称	设备机型	IP地址	MAC地址
XSDH-60A32	XSDH-60A32	192.168.6.6	88: 1:f9:30:fb:67

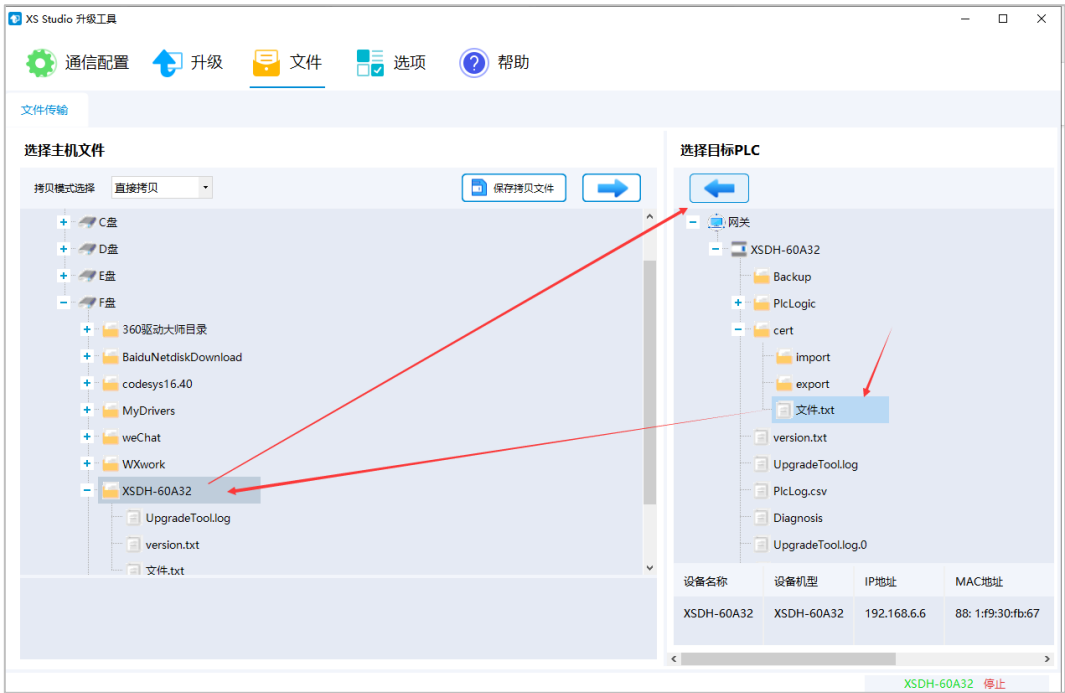
1、主机文件传输到 PLC

拷贝模式选择“直接拷贝”，在主机文件里选择需要拷贝的文件，在目标 PLC 中选择保存拷贝文件的文件夹名称，点击或者右键选择“拷贝到 PLC”即可将文件传送到目标 PLC 里。

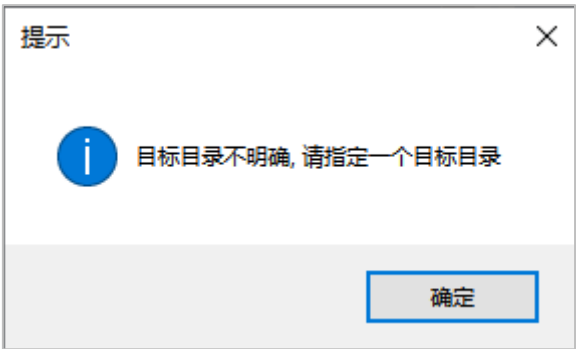


2、PLC 文件传输到主机

在右侧“选择目标 PLC”窗口中选中需要拷贝的文件，在主机文件里选择保存拷贝文件的文件夹名称，点击或者右键选择“拷贝到主机”即可传送文件到主机。



主机文件界面或者目标 PLC 界面未选择任何目录，则弹窗提示目标目录不明确，需要指定一个目标目录。

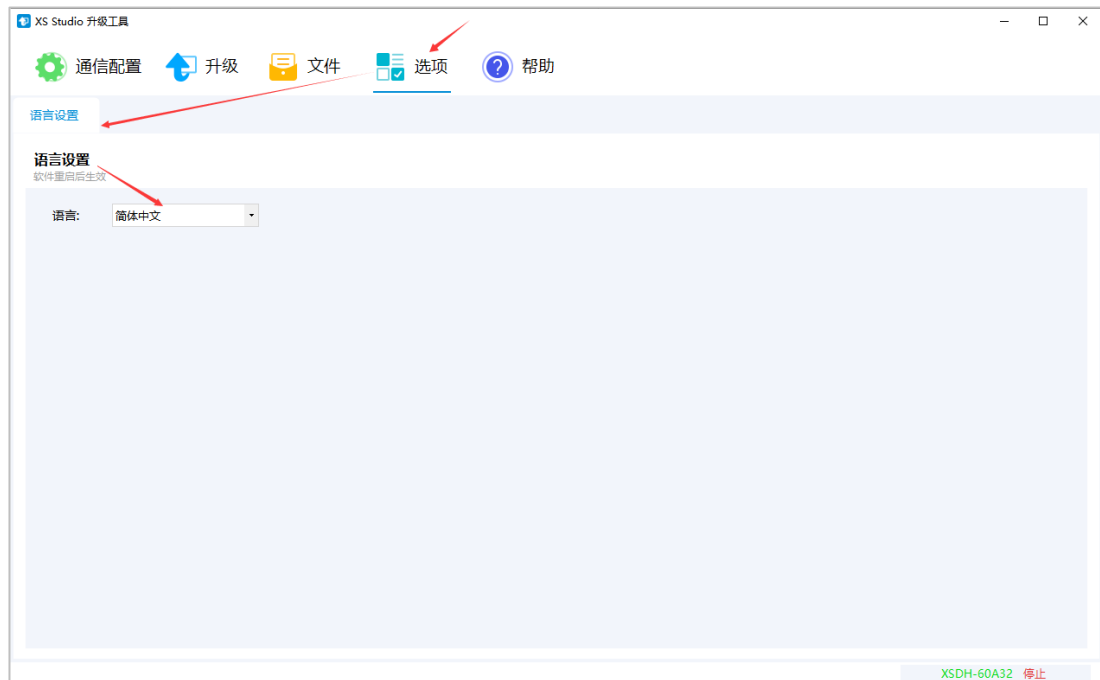


- 若传输时检测设备存储空间不足，则弹框提示用户：设备存储空间不足，文件“xxx”传输失败！
- 目前支持传输到一台选择的 PLC，不同时传多台 PLC。
- 保存拷贝文件可以将拷贝文件保存至主机任意位置比如 U 盘内。

10-4. 选项

选项功能用于配置软件语言。点击功能菜单栏的选项，可以进入语言设置界面。

在“语言设置”里可以通过下拉框选择简体中文或 English。选择需要的语言后，重启软件即可实现语言的切换。



10-5. 帮助

点击“帮助”可以打开对应手册，帮助用户更好地使用该升级工具。

11. 设备诊断

11-1. 设备诊断

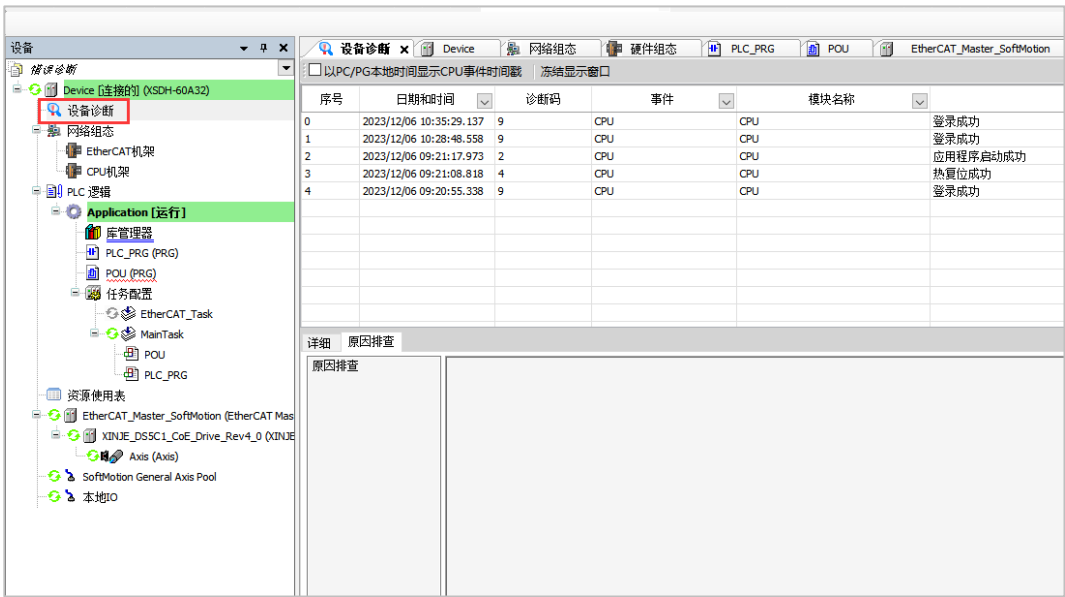
故障诊断是为了快速定位 PLC 运行过程中出现的错误,帮助用户能够通过诊断错误信息或状态找出对应的解决方案。此功能适用信捷 codesys 平台全系列机型 (2.3.0 及以上版本)。

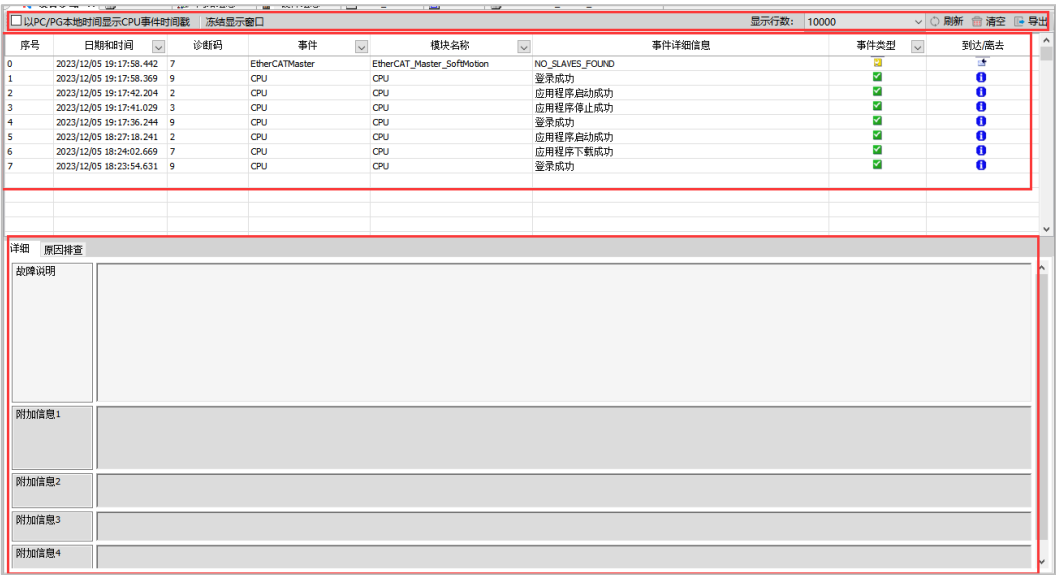
通过诊断码可以诊断 CPU、MODBUS (RTU/ASCII)、MODBUS TCP、EtherCat、Ethernet/IP、本体扩展模块 (右扩展模块、BD 板扩展、ED 扩展模块) 的错误信息。

11-2. 具体功能说明

设备诊断用于显示所有设备的诊断信息,并提供相关故障信息的详细说明、原因和排查方法;同时能够针对特殊情况提供更详细的诊断信息。

登录 PLC 后双击设备树“设备诊断”获取和显示设备的诊断信息;界面自动保存该设备上一次登录后的诊断信息,再次登录且运行后进行刷新可以显示最新的诊断信息。

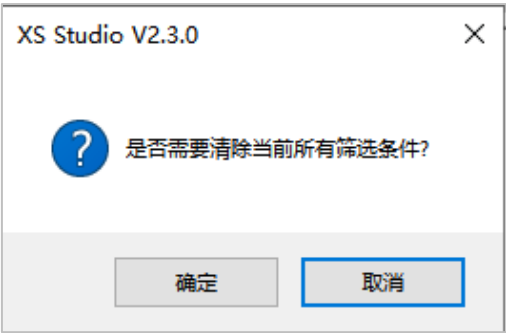




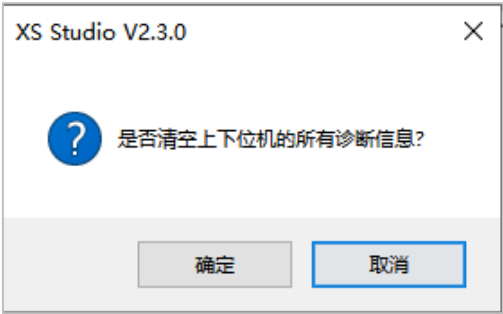
1、按钮说明

以 PC/PG（编程控制器）本地时间显示 CPU 事件时间戳	不勾选，诊断信息的日期和时间以电脑的时间进行显示；勾选后以控制器的时间进行显示； 注：该功能需要 2.3.0 及以上版本支持。
冻结显示窗口	点击“冻结窗口显示”，会变成“解冻窗口显示”，此时诊断缓冲区条目保持当前数据不会被刷新。冻结后 CPU 会继续在诊断缓冲区中接收诊断信息的输入
显示行数	该界面默认的显示行数为 10000 条，显示行数选用下拉框选择，可选项 1000、2000、5000、10000、20000、50000、100000、200000；诊断信息缓存条数最大支持 20 万条。
刷新	用于清除所有筛选条件
清空	清空下位机所有诊断的所有信息，同时清空上位机表格中的诊断信息
导出	导出表格中缓存的所有诊断信息

【刷新】：用户也可以在对应支持筛选项中进行下拉选项自行手动改变筛选条件。点击刷新时会提示“是否需要清除当前所有筛选条件”，点击“确定”即清除所有筛选条件。



【清空】：点击清空按钮会提示“是否要清空下位机的所有诊断信息”，点击“确定”即清空诊断信息。



【导出】：导出文件格式以.CSV 文件的形式导出；导出文件中包含序号、时间和日期、错误码、事件、模块名称、事件详细信息、事件类型、到达/离去；

在导出故障信息最后一行输出当前设备的设备信息，包含该设备的 PLC 类型、上位机软件版本、对应 PLC 固件版本，文件信息大致如下：

序号	日期和时间	错误码	事件	模块名称	事件详细信息	事件类型	到达/离去
1	2023/7/4 17:25:03.89	101	Ethercat	ETHERCAT.C	NO_COMM	错误	到达
2	2023/7/4 17:25:03.100	203	Ethercat	XINJE_DSSC1_CoE_Drive_Rev4_0	模块掉线或通信失败	错误	到达
3	2023/7/4 17:25:03.200	406	Ethercat	Axis	IC_DL_ETC_DEVICE_NOT_IN_OP_MO	错误	到达
4	2023/7/4 17:30:03.567	0	Ethercat	XINJE_DSSC1_CoE_Drive_Rev4_0	模块通讯正常	正常	离去
5	2023/7/4 17:30:03.345	0	Ethercat	Axis	轴通讯正常	正常	离去
6	2023/7/4 20:08		PLC类型: XSDH-60A32-E	XS Studio 软件版本: xxxxxx	PLC固件版本: xxxxxx		

2、诊断信息表格说明

序号	对诊断信息条目以时间轴发生的先后顺序进行排序，当使用筛选功能后对应事件的序号与未筛选对应事件的序号保持一致
时间与日期	按照年/月/日 时：分：秒：毫秒的形式显示
诊断码	当前诊断信息对应的诊断码
事件	显示该诊断信息属于哪种通讯类型
模块名称	显示该诊断信息属于事件下的特定设备名称
事件详细信息	显示该事件详细诊断信息
事件类型	分为正常、错误、严重错误三种类型
到达/离去	故障事件错误的产生与恢复，分为到达事件、离去事件、离去事件与到达事件无关

① 【时间类型】：时间类型详细说明如下：

图标	操作	说明
	正常	代表设备连接或用户操作对应的命令正常，例如：连接建立成功、启动-停止 PLC、退出-登录、添加组态且运行成功
	错误	代表设备连接或用户操作对应的命令异常，例如：建立的通讯连接断开、设备异常等
	严重错误	代表设备连接或用户操作对应的命令出现严重错误，例如：PLC 出现死机、掉电保持数据丢失、有野指针数据的使用等比较严重的错误类型

② 【到达/离去】:

图标	操作	说明
	到达	故障事件的产生，对应的事件为到达事件
	离去	对应故障事件产生的恢复
	离去事件与到达事件无关	操作正常

③ 诊断信息按照先进先出的处理方式，后进入的诊断信息排在列表的第一行，先进入的诊断信息排在列表的后面。超过最大的缓存条数后会覆盖先进入的诊断信息。

④ 右击选中某一故障信息可以跳转到事件，最近一条离去/到达。

跳转到事件：可以跳转到该事件设备配置界面，以使用户能够快速修改或定位对应的事件。

序号	日期和时间	诊断码	事件	模块名称	事件详细信息	事件类型	到达/离去
0	2023/12/06 09:21:17.990	2	CPU	CPU	应用程序启动成功	✓	到达
1	2023/12/06 09:21:08.835	4	CPU	CPU	热复位成功	✓	到达
2	2023/12/06 09:21:09.398	0	EtherCATMaster	EtherCAT_Master_SoftMotion	成功	✓	到达
3	2023/12/06 09:20:56.387	1	EtherCATMaster	EtherCAT_Master	成功	✓	到达
4	2023/12/06 09:20:55.355	9	CPU	CPU		✓	到达

跳转到最近一件离去：可以跳转到该故障信息最近一条离去即恢复正常的诊断信息。

序号	日期和时间	诊断码	事件	模块名称	事件详细信息	事件类型	到达/离去
0	2023/12/06 09:21:17.990	2	CPU	CPU	应用程序启动成功	✓	到达
1	2023/12/06 09:21:08.835	4	CPU	CPU	热复位成功	✓	到达
2	2023/12/06 09:21:09.398	0	EtherCATMaster	EtherCAT_Master_SoftMotion	成功	✓	到达
3	2023/12/06 09:20:56.387	1	EtherCATMaster	EtherCAT_Master	成功	✓	到达
4	2023/12/06 09:20:55.355	9	CPU	CPU		✓	到达

跳转到最近一件到达：可以跳转到该诊断信息最近一条到达即产生故障的诊断信息。

序号	日期和时间	诊断码	事件	模块名称	事件详细信息	事件类型	到达/离去
0	2023/12/06 09:21:17.990	2	CPU	CPU	应用程序启动成功	✓	到达
1	2023/12/06 09:21:08.835	4	CPU	CPU	热复位成功	✓	到达
2	2023/12/06 09:21:09.398	0	EtherCATMaster	EtherCAT_Master_SoftMotion	成功	✓	到达
3	2023/12/06 09:20:56.387	1	EtherCATMaster	EtherCAT_Master	成功	✓	到达
4	2023/12/06 09:20:55.355	9	CPU	CPU		✓	到达

3、详细/原因排查窗口

① 信息详细说明窗口：当在诊断信息列表中选中了某一条诊断信息，在信息详细说明窗口会显示该故障的详细信息。

附加信息：增加 4 个附加信息，主要用于对故障提供更多的信息说明，展示多节点组网设备中故障设备的节点地址、站号、IP 地址。

例如：ethercat 通讯过程中网线异常断开，详细说明如下图所示：

详细	原因排查
故障说明	【原因】 1. PLC 与第一个从站硬件连接断开；
附加信息1	
附加信息2	
附加信息3	
附加信息4	

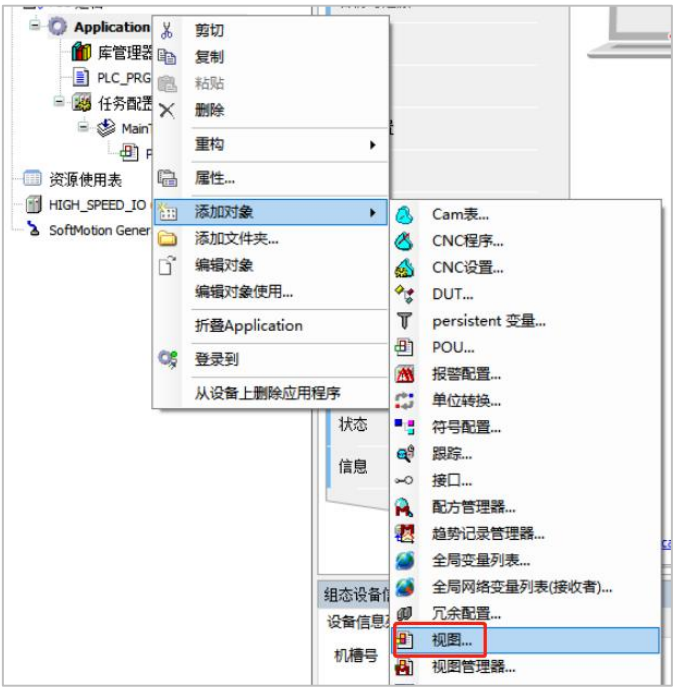
② 原因排查：用于提供原因排查的具体操作方法。

例如：ethercat 通讯过程中网线异常断开，原因排查如下图所示，提供原因排查的具体操作方法。



12. 可视化

采用可视化功能，可以把内部变量的数值动态地显示在界面中，以使用户更加直观地看到所需功能的执行情况。



12-1. 视图

所有 codesys 机型都支持在工程里添加视图。



12-1-1. 视图管理器

1、“使用 Unicode 字符串”

统一码 (Unicode)，也叫万国码、单一码，是为了解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。

若需要在视图中使用中文等其它语言的字符，则需要勾选“使用 Unicode 字符串”才能正常显示。



2、“使用当前视图变量”

勾选“使用当前视图变量”后才可以使用 STRING 类型的全局变量 VisuElems.CurrentVisu，该变量可以显示应用程序运行中的视图的名称。支持读取该变量的值来获取当前活动的视图名称或者可以修改该变量的值来切换视图。

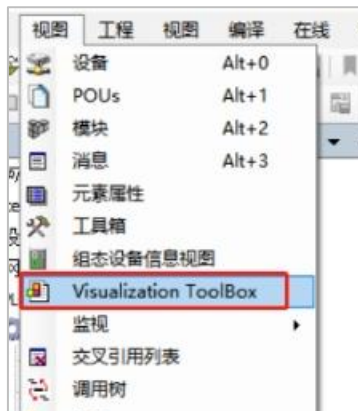


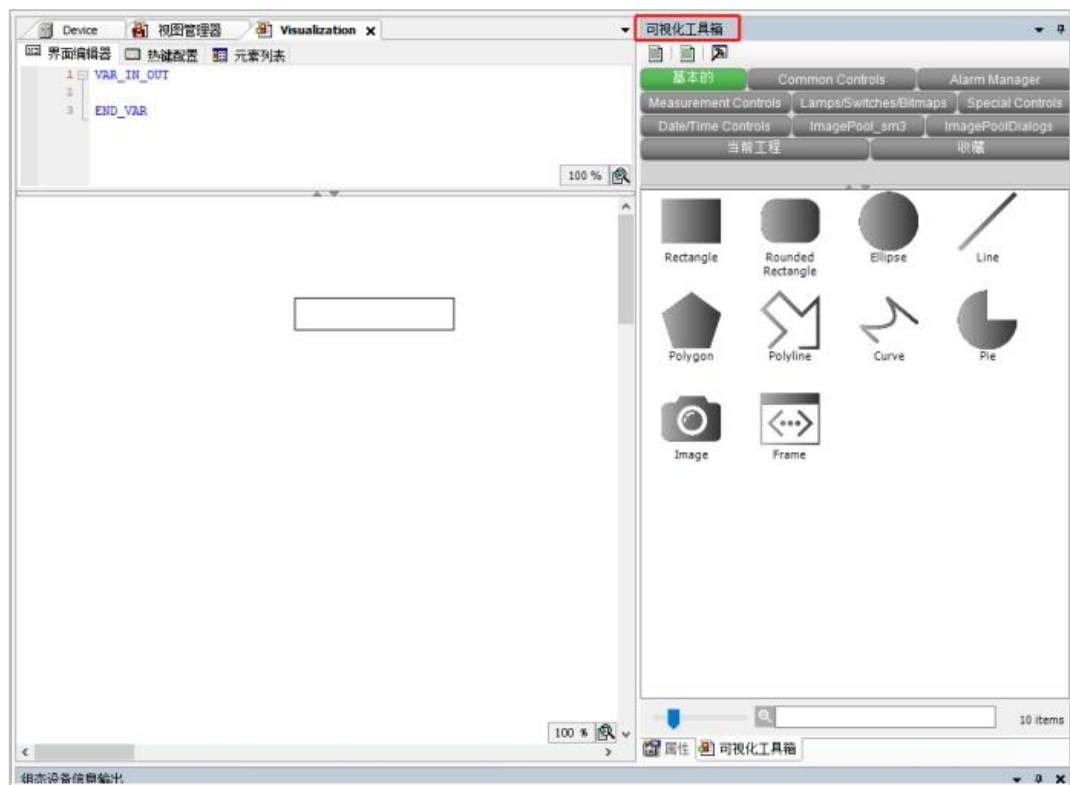
视图管理器更多介绍可见 codesys 在线帮助：

https://content.helpme-codesys.com/zh-CHS/CODESYS Visualization/_visu_obj_manager.html

12-1-2. 可视化工具箱

提供可视化元素，用户可以将可视化元素插入到视图中。通过点击“视图” - “可视化工具箱 (Visualization ToolBox)”可以关闭和打开该界面。

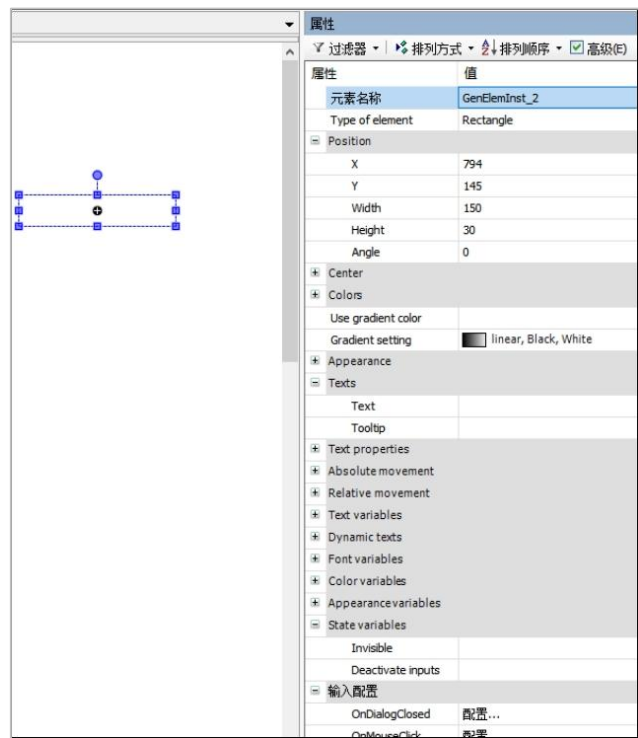




可视化工具箱更多介绍可见 codesys 在线帮助：
https://content.helpme-codesys.com/zh-CHS/CODESYS%20Visualization/_visu_view_toolbox.html

12-1-3. 元素属性

添加可视化元素后，点击该元素可看到该元素的“属性”标签页。可以配置元素的位置、颜色、映射变量等。



可视化元素的更多介绍可见 codesys 在线帮助：
https://content.helpme-codesys.com/zh-CHS/CODESYS%20Visualization/_visu_f_reference_prog_visualization_elements.html

12-2. 本地可视化

只有 XSA-W 支持本地可视化。

12-2-1. 配置步骤

添加“目标视图”和视图画面后，可以配置“目标视图”中“起始视图”等参数，然后将程序下载到 XSA-W 中，此时使用 DP 转 VGA 转换线将 XSA-W 的 DP 接口与显示器的 VGA 接口连接起来就可以在显示器上看到视图画面。





12-2-2. 远程进入桌面

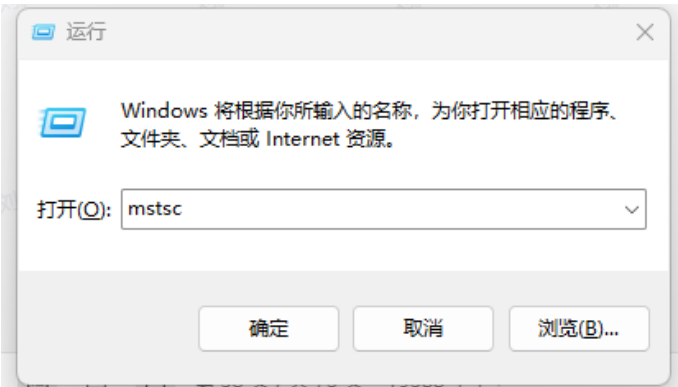


虚拟化机型 XSA230-W、XSA620-W 不支持远程桌面功能，其他 XSA-W 机型支持远程桌面功能。

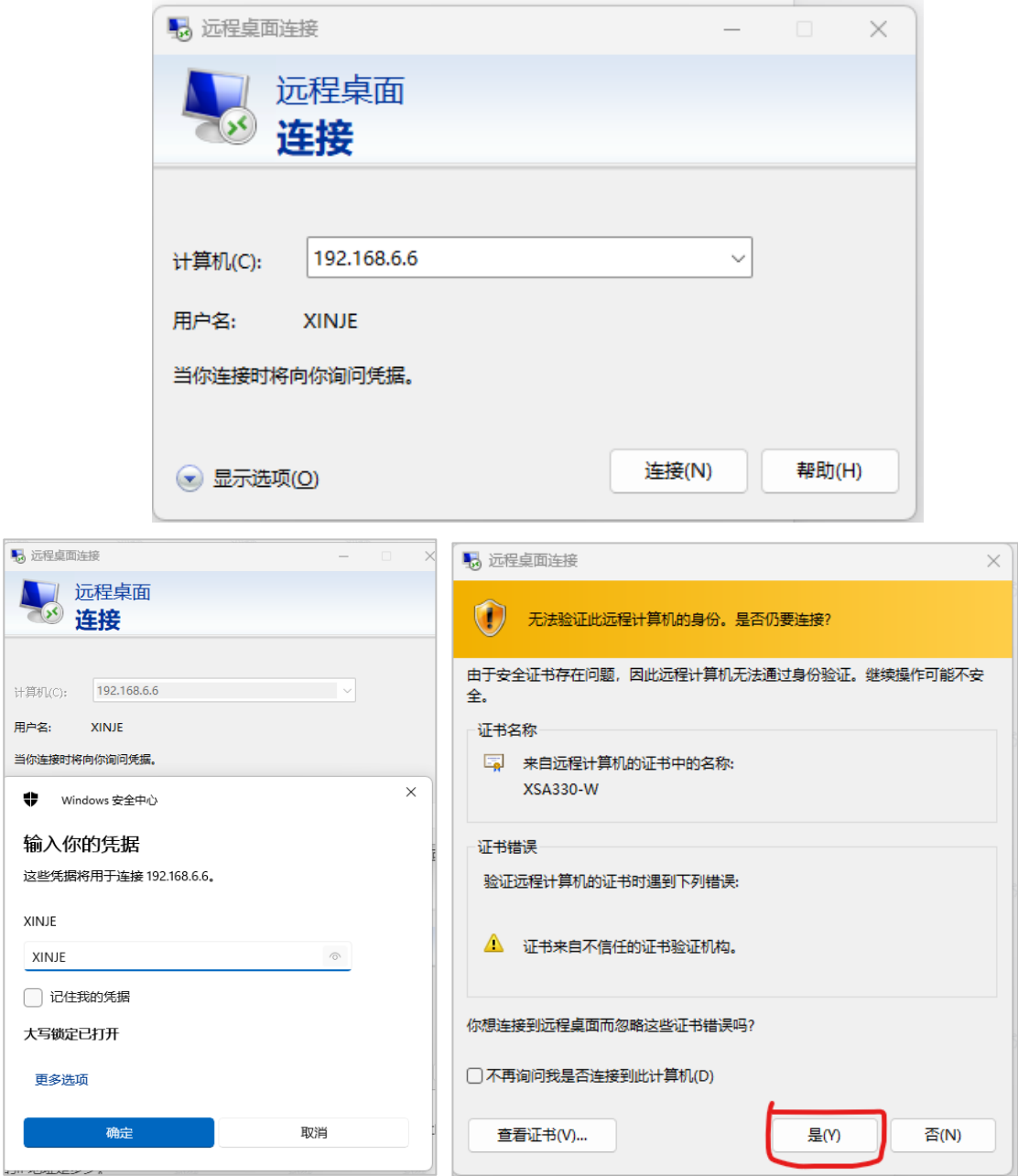
连接桌面

工控机直连电脑。

- 1、电脑打开“远程桌面”功能：
- 方式一：左下角输入远程桌面；
- 方式二：键盘按下 WIN+ R，输入 mstsc 回车弹出远程桌面。



- 2、输入 XSA-W 的 IP 地址，如果不确定 IP 地址，可以程序加一条读取 IP 地址的指令，或者键盘按下 WIN+R 键后输入 cmd，再输入 arp -a (“-”前有个空格)，看下工控机的 IP 地址是多少。



一般工控件出厂均已设置密码，有密码就可以直接远程连接，用户名和密码都为“XINJE”，若客户已更改，可以参照以下方式打开远程设置密码。

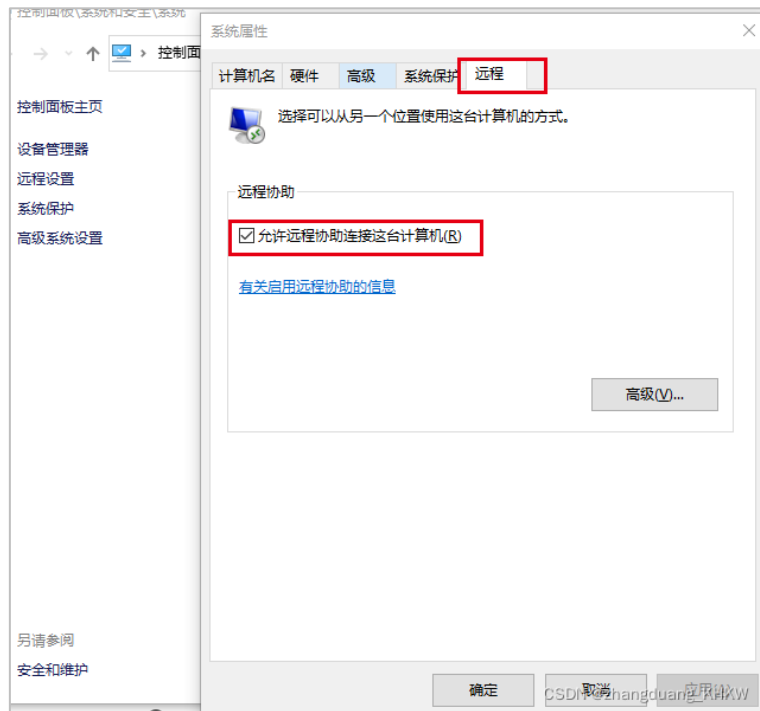
使用 DP 线将 XSA-W 与显示器连起来，进入 XSA-W 的桌面，打开远程控制：桌面找到我的电脑，右键选择“属性”；



点击“远程设置”；

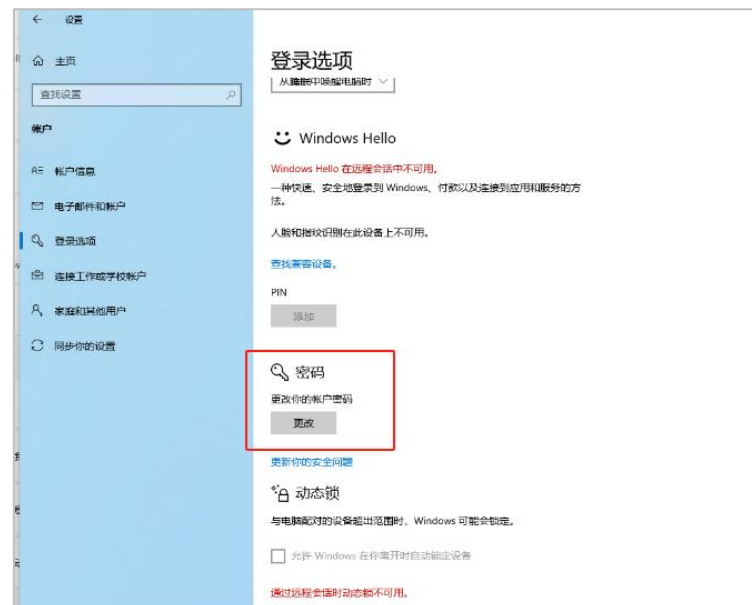


点击“远程”->把“允许远程协助连接这台计算机”前面的框点击选中，然后点击下面的“确定”；



设置密码。

win10:



12-3. 网络可视化

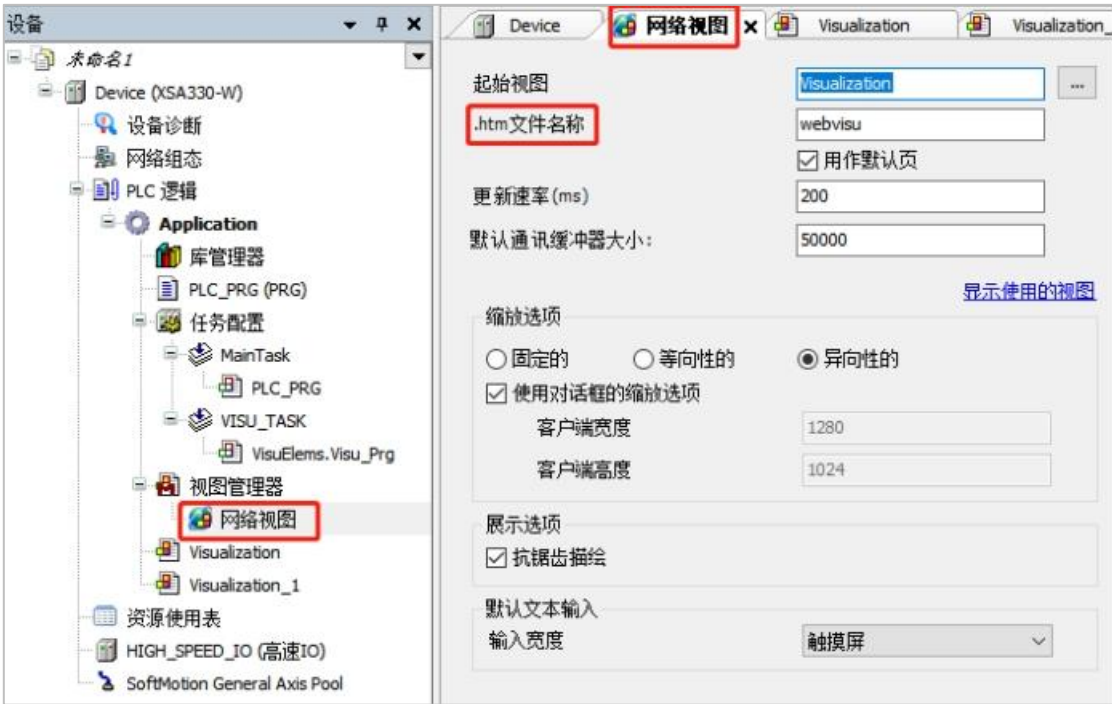
目前 XSA-W（所有固件版本）、XSA-L（2.3.0 及以上固件）和 XSF5（2.3.0 及以上固件）支持网络可视化功能。

与控制器或工控机处于同一局域网内的电脑、手机等设备都可以登录网络可视化的网址来访问和监控视图画面。浏览器网址：**http://控制器 IP:端口号 8080/网络可视化文件名称.htm**。



网址里的“网络可视化文件名称.htm”可写可不写（举例：**http://192.168.6.11:8080**）。

优点：可以大屏看可视化画面、手机连接进行点位调试等。





13. 任务分核

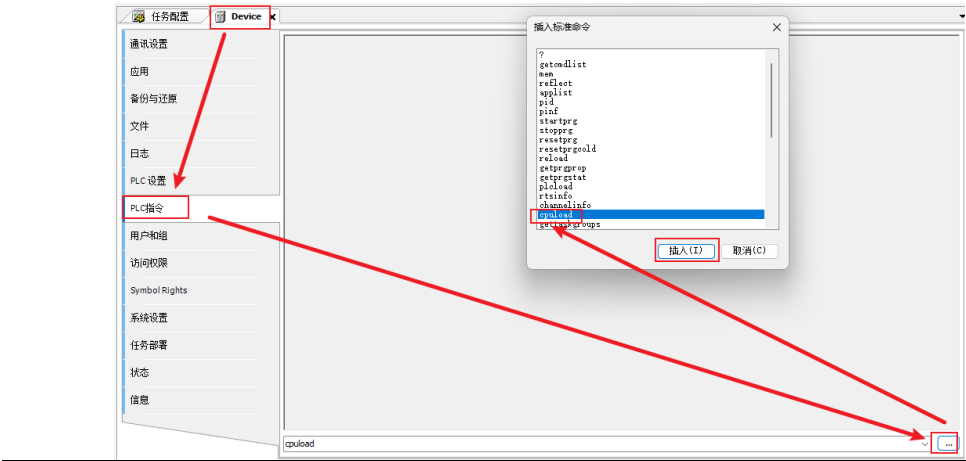
核心数为一个物理 CPU 拥有的运算单元数量。使用多核的 codesys 控制器比如使用 XSF5 或 XSA-L 时，若 codesys 程序很大或运行 cpuload 很高，则可以考虑将工程中的多个任务放在不同核下运行来平衡控制器的运行负载。

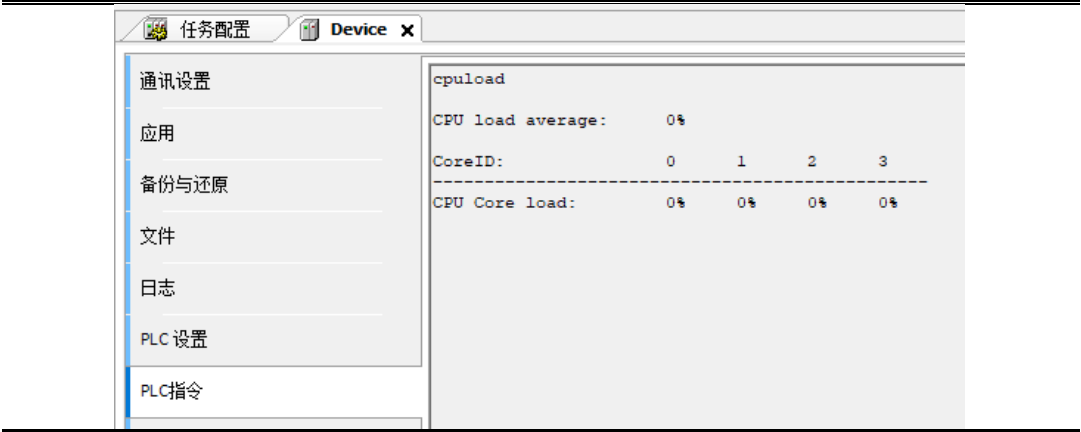
13-1. 适用设备

功能 XS 机型及版本	上位机自动分核 ^{注2}		下位机自动分核 ^{注3}		手动分核	
	支持情况	核心数	支持情况	核心数	支持情况	核心数
XSDH	不支持	1 核	不支持	1 核	不支持	1 核
XSLH	不支持		不支持		不支持	
XSF5	支持	2 核	固件版本 2.4.2 及以上支持	2 核	支持	2 核
XSA-W	不支持	1 核	不支持	(见下方 ^{注1})	不支持	1 核
XSA-L	支持	2 核	固件版本 2.4.2 及以上支持		支持	2 核



- 注 1：XSA330、XSA520、XSA550 是 4 核；XSA520、XSA230、XSA620 是 2 核。
- 注 2：上位机自动分核：
XS Studio 软件版本达到 V2.3.2 及以上时，若多核 PLC 固件版本低于 2.4.2 或者 XML 机型版本低于 2.4.2 时，则开启自动分核后使用的是上位机自动分核策略。
- 注 3：下位机自动分核：
多核 PLC 固件版本达到 2.4.2 及以上且配合使用 2.4.2 及以上 XML 机型文件时，若开启自动分核则使用的是下位机自动分核策略。
- 查看 CPU 负载率的方法：
 - ◆ XSF5 和 XSA-L 系列机型支持在“任务配置”-“核负载”界面查看负载率；
 - ◆ 所有 XS 系列支持使用 XJ_GetCpuUsage 查看负载率，指令说明详见《XS 系列 PLCopen 标准控制器用户手册【基本指令篇】(XS Studio)》
 - ◆ “Device” - “PLC 指令”界面点击“...” 若有 cpuload 则可以插入并回车，可以查看负载率。





13-2. 自动分核

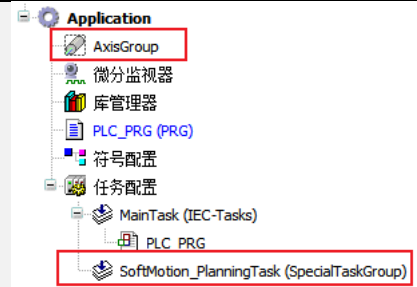
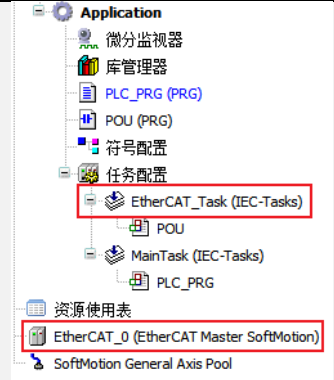
用户无需手动指定每个任务的CPU核心，系统可依据预设策略自动完成核心分配，提高系统整体性能与配置效率。

分为上位机自动分核和下位机自动分核两个功能。

13-2-1. 上位机分核策略

XS Studio 软件版本达到 V2.3.2 及以上时，若多核 PLC 固件版本低于 2.4.2 或者 XML 机型版本低于 2.4.2 时，则开启自动分核后使用的是上位机自动分核策略。

任务模型		分核策略
存在任务类型	任务截图	
可视化任务		该任务绑定至核心 0
报警管理任务		
趋势图任务		

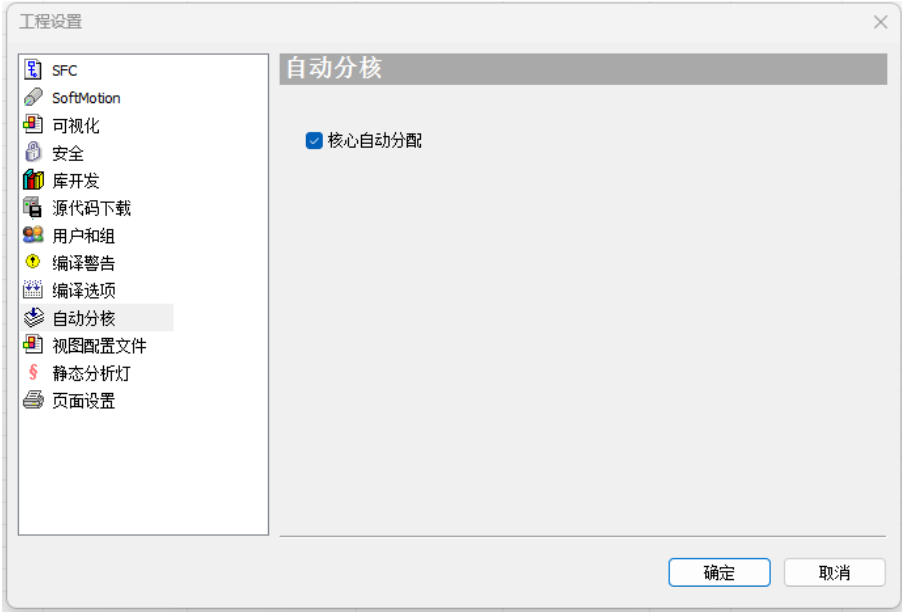
轴组任务			
不存在 EtherCAT 功能时 (包括其他总线通信任务)		/	所有任务绑定至核心 1
有 EtherC AT 功 能	只 有 一 个 EtherCAT 主站		该任务绑定至核心 1
	有 两 个 EtherCAT 主站		两个任务分别绑定一个核心(①先比轴数,轴数多的任务放在核心 1;②若轴数相同再比同步周期,周期小的放在核心 1;③若同步周期相同再比优先级,优先级小即优先级高的放在核心 1)。
	除了 EtherCAT 任务外的其他任务(包括其他总线通信任务)	/	非 EtherCAT 的其他任务绑定核心 0

13-2-2. 下位机分核策略

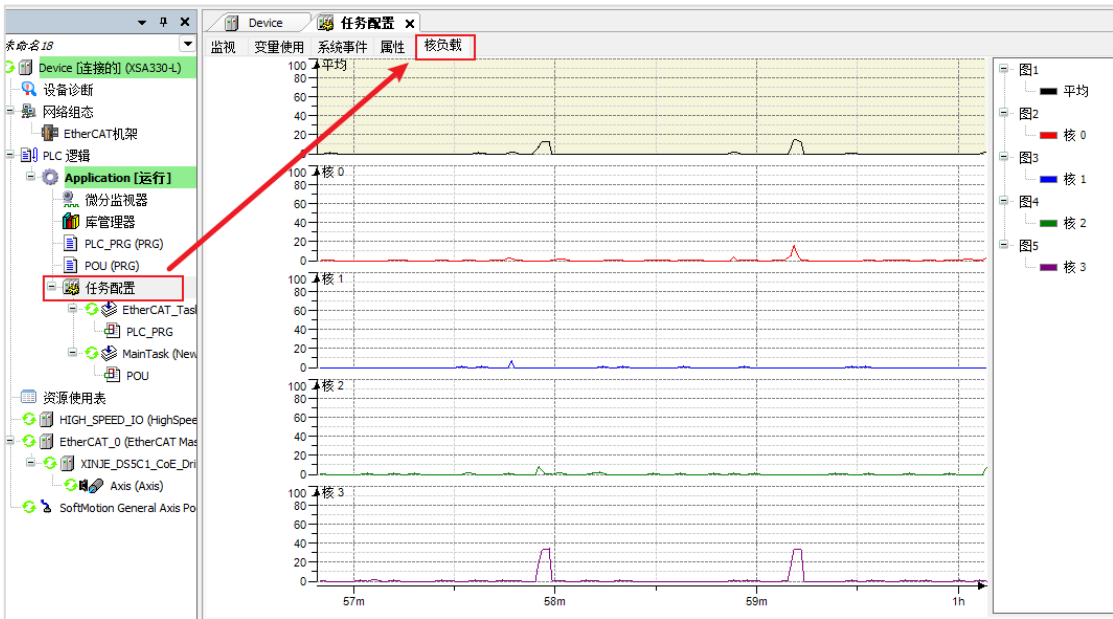
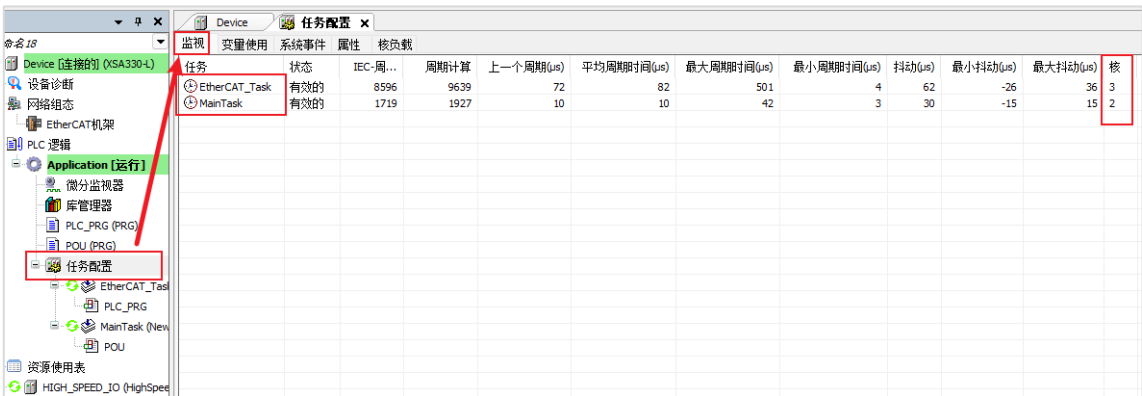
多核PLC固件版本达到2.4.2及以上且配合使用2.4.2及以上XML机型文件时，若开启自动分核则使用的是下位机自动分核策略。该策略是多核PLC有多少核心就会被系统配置多少核心，系统自动完成核心分配，提高系统整体性能与配置效率。

13-2-3. 配置步骤

- 1、打开 XS Studio 软件，点击“工程”-“工程设置”-“自动分核”，勾选“核心自动分配”。

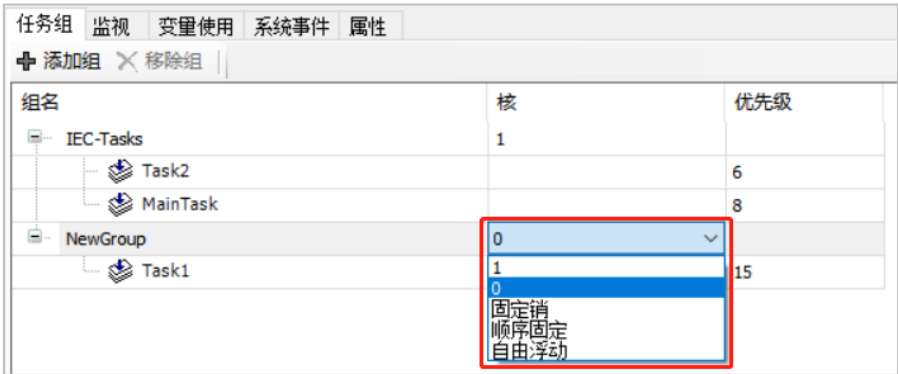


- 2、此时便已配置好了自动分核，程序登录在线后可以通过“任务配置”-“监视”页面查看系统给各个任务自动分配的所在核心，同时也可以从“任务配置”-“核负载”页面查看各个核心的负载情况。



13-3. 手动分核

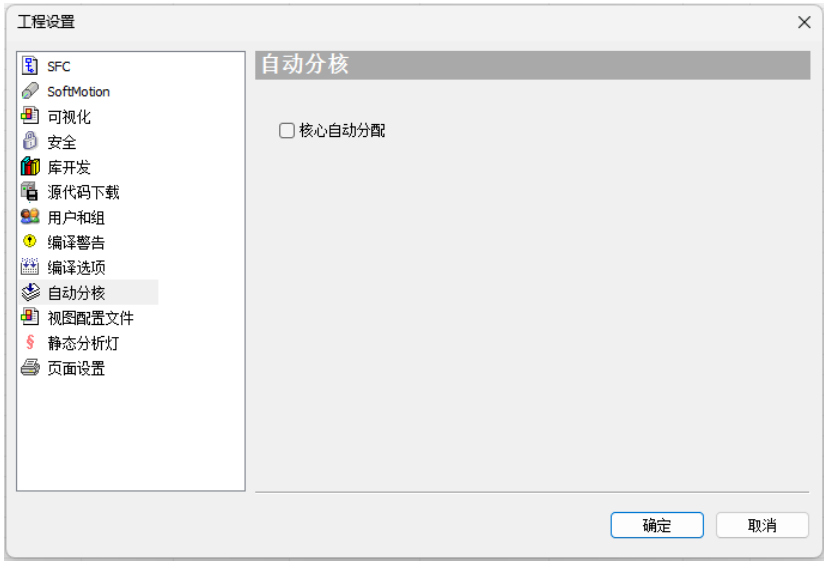
13-3-1. 参数说明



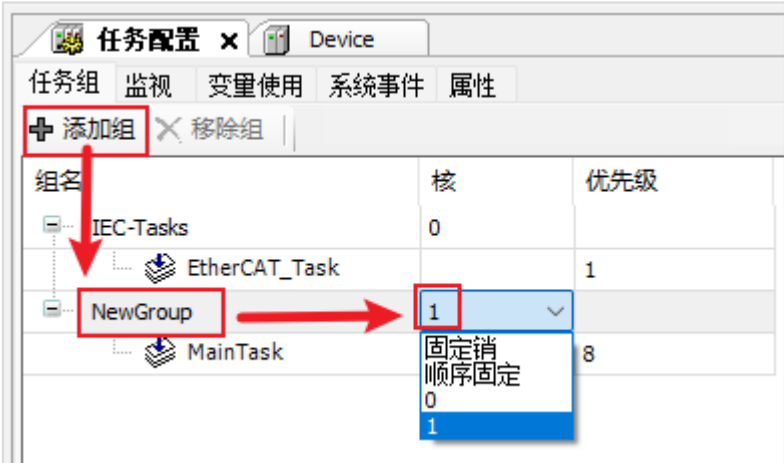
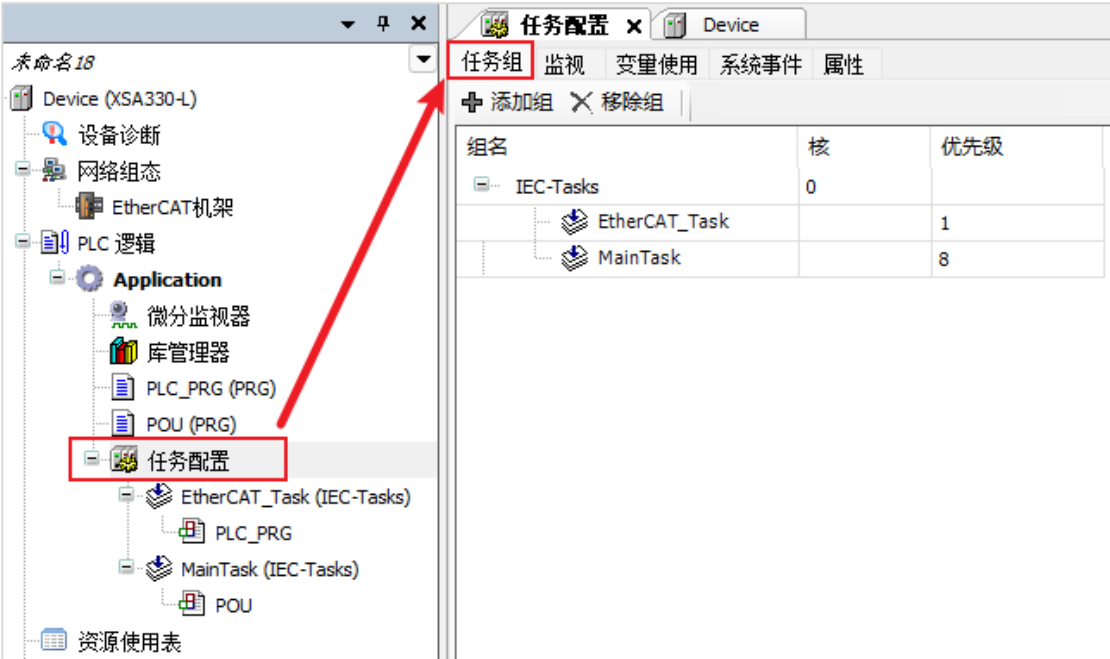
核	选项
0	绑定到处理器内核 0 运行
1	绑定到处理器内核 1 运行
固定销	由 codesys (Runtime) 决定绑定运行在处理器的哪个核上
顺序固定	所有任务都绑定并固定到不同的处理器内核。 比如任务组里有 2 个组为“IEC-Tasks”、“NewGroup”，都设为顺序固定的情况下， 则 2 个组依次被绑定到核 0、核 1 上。
自由浮动	XSF5: 选择自由浮动是固定绑定在核 0 上 XSA-L: 没有自由浮动的选项

13-3-2. 配置步骤

1、打开 XS Studio 软件，点击“工程”-“工程设置”-“自动分核”，取消勾选“核心自动分配”。



2、“任务配置”-“任务组”界面点击“添加组”，可以修改组名、修改绑定核心策略，比如添加组后将新添加组的核设为 1。此时可以将核 0 的 IEC-Tasks 里的某些任务拖拽到新组里，如下图是将 MainTask 拖拽到新组核 1 里。



3、任务分核后，登录下载程序，此时在“任务配置”-“核负载”页面可以查看各个核心的负载情况，“任务配置”-“监视”页面可以查看所有任务目前运行的最大周期等参数。

任务	状态	IEC-周期计算	周期计算	上一个周期(us)	平均周期时间(us)	最大周期时间(us)	最小周期时间(us)	抖动(us)	最小抖动(us)	最大抖动(us)	核
EtherCAT_Task	有效的	186306	187873	87	90	559	4	95	-47	48	0
MainTask	有效的	37262	37575	10	10	31	3	88	-44	44	1



- XS StudioV2.3.2 软件：
多核 PLC 工程里“任务配置”-“任务组”页面会一直存在，但只有将“工程”-“工程设置”-“自动分核”界面里的“核心自动分配”取消勾选后，该“任务组”界面的手动分核配置才会生效。
- XS StudioV2.3.2 及以上版本软件：
多核 PLC 工程里只有将“工程”-“工程设置”-“自动分核”界面里的“核心自动分配”取消勾选后，“任务配置”里才会显示“任务组”界面并支持手动分核配置。

14. Codesys 加密

14-1. 工程加密

14-1-1. 功能说明

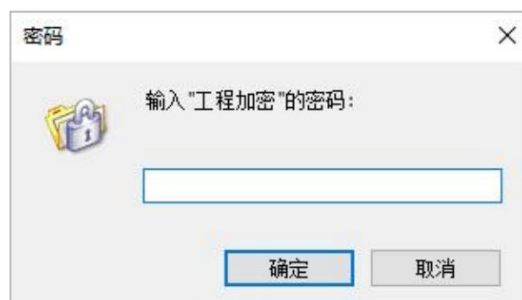
打开加了密的工程后会提示输入密码，不输密码或密码错误则无法打开工程、无法下载工程和无法看到工程的内容。

14-1-2. 配置步骤

“工程”-“工程设置”-“安全”，可以点选加密，比如点选密码后，设置密码后，将整个工程保存关闭。



然后再打开该工程，就需要输入密码才能打开。



14-2. POU 加密

14-2-1. 功能说明

POU 加密就是设置 POU 的访问控制。



项目	说明
窗口	能否看到该 POU 及其添对象的程序
修改	能否修改程序
删除	能否删除该 POU 及其添加对象的程序
添加/删除子	能否添加或删除该 POU 的对象（动作、方法、属性、转移）

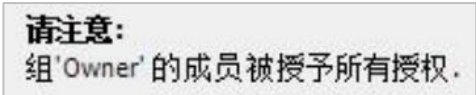
这就需要配置哪些组的成员被限制其访问控制。
配置前提：需确认能以组'Qwner'的成员登录，因为组'Qwner'的成员被赋予所有授权，也就是拥有最高权限，可以配置所有用户权限。

14-2-2. 配置步骤

- 1、在软件右下角可看到当前用户登录情况，可进行用户登录或退出登录。新建的程序默认是无用户登录状态。

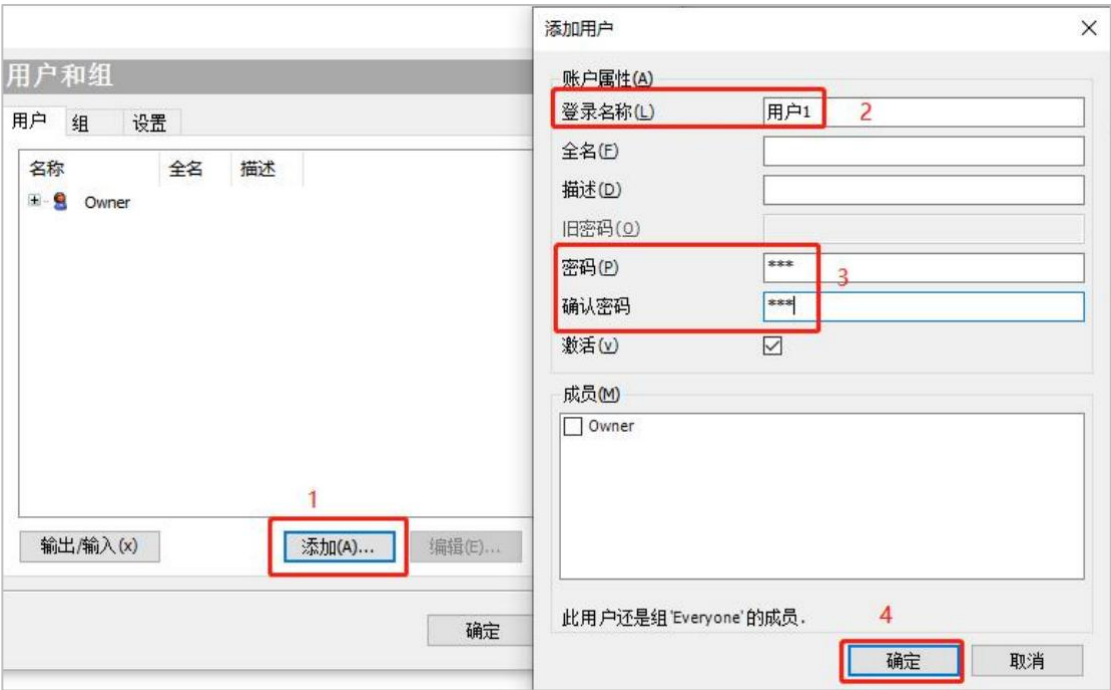


- 2、工程-工程设置-用户和组，可以看到组'Owner'的成员，默认有个用户成员'Owner'（初始密码是空白没有的），用户成员'Owner'的登录名称、密码等可以重新设置。





3、举例添加一个登录名称为“用户 1”的用户。没有改过工程授权的前提下，默认需以组‘Owner’的成员登录才能进行配置，见下图。



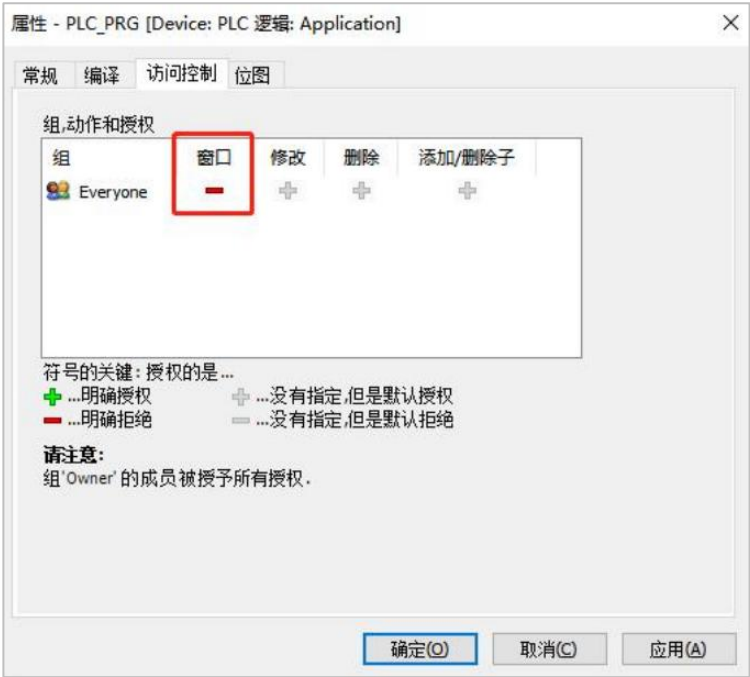


4、添加好“用户 1”后，可以在组面板里看到默认有两个组，分别为组’Owner’ 和组’Everyone’， “用户 1” 默认已经添加到组’Everyone’里了。



5、选中要加密的 POU，点击鼠标右键，选择“属性”-“访问控制”，可以配置除了组’Qwner’ 外其他组对该 POU 的访问权限。举例将组’Everyone’的 “窗口” 设置为拒绝，如下图：

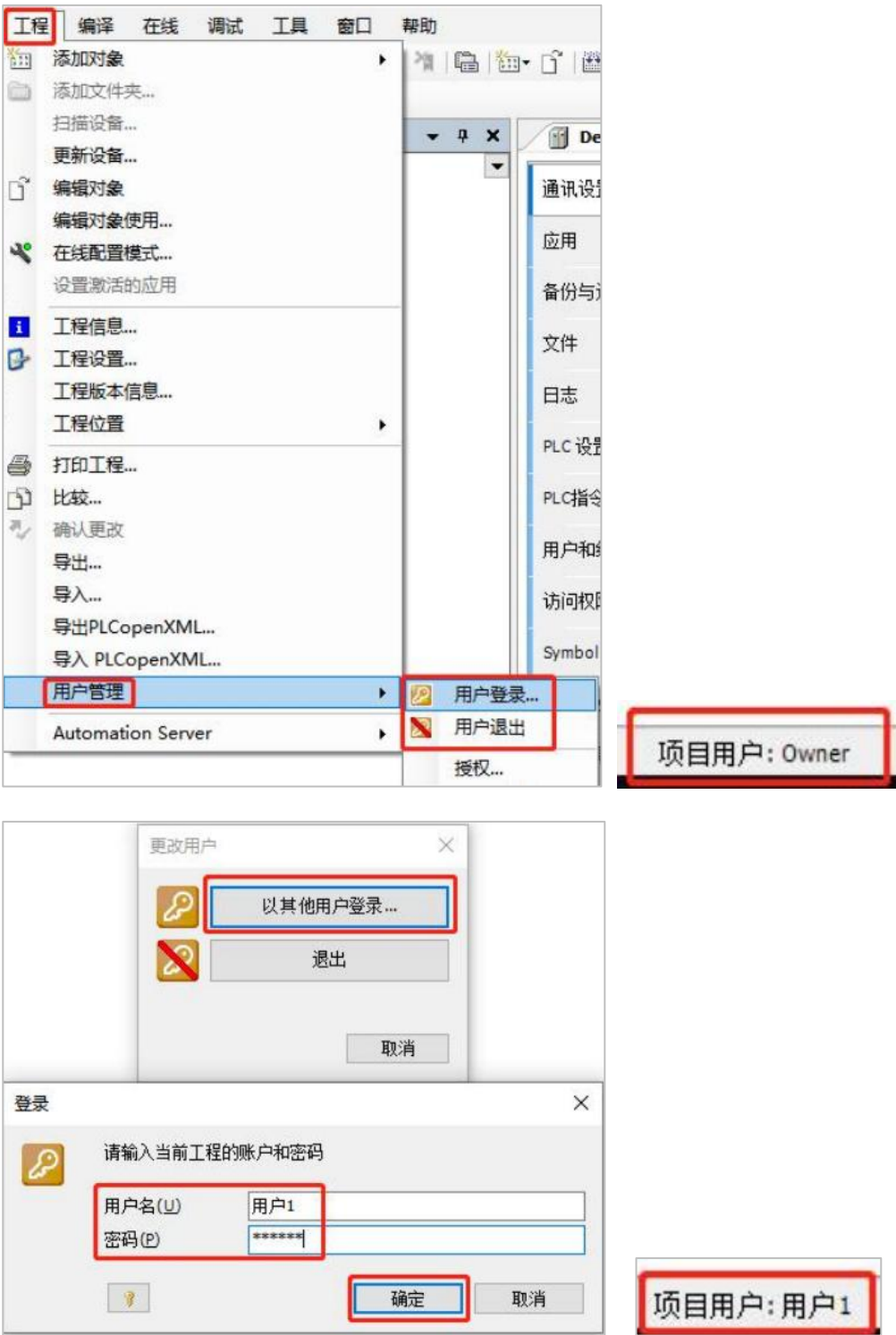
- 窗口：能否看到该 POU 及其添加对象的程序；
- 修改：能否修改程序；
- 删除：能否删除该 POU 及其添加对象的程序
- 添加/删除子：能否添加或删除该 POU 的对象（动作、方法、属性、转移）。



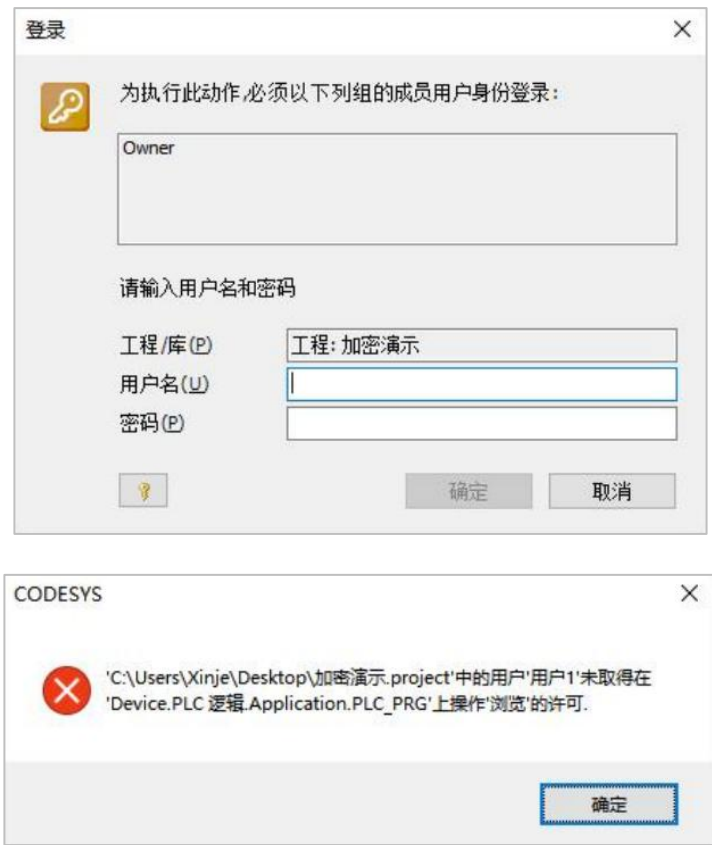
6、然后用登录名称为“用户 1”的用户登录。

有两处地方可以操作用户登录和退出功能：

- ① 点击“工程” - “用户管理”；
- ② 双击软件右下角“项目用户”。

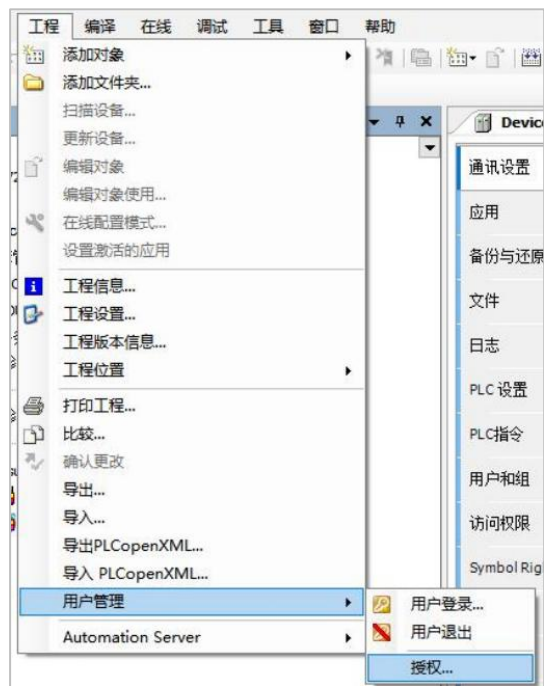


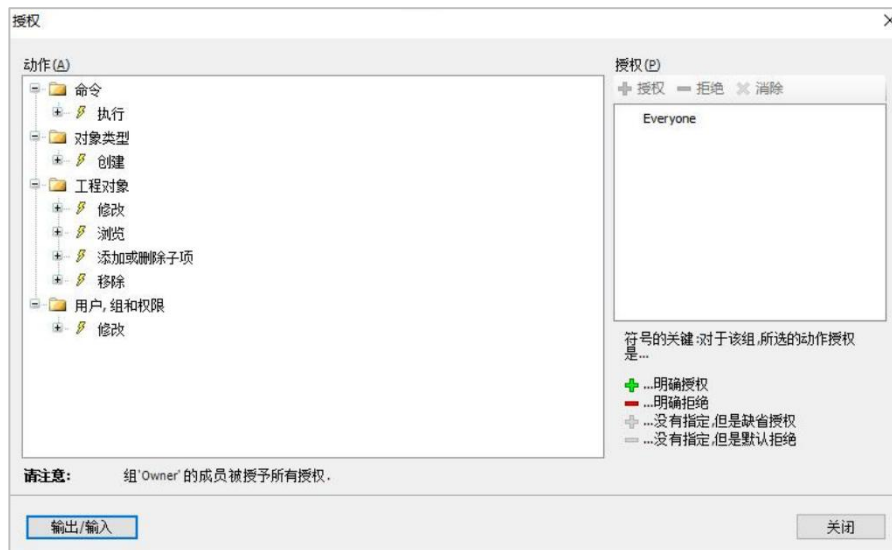
7、当前是“用户 1”在登录，双击加密的 POU 是无法查看程序的，提示需要以组'Owner'的成员用户登录才行。这就实现了 POU 加密的功能。



14-2-3. 注意点

- 组'Everyone'包含所有用户（包括没有用户登录的情况），所以一般配置组'Everyone'的访问控制权限即可（这就配置了除组'Owner'的用户成员外其他成员的访问控制权限）。若组'Everyone'和组'Owner'会有相同的用户成员，这些成员登录不会受到访问限制，因为组'Owner'的成员被赋予所有授权，也就是拥有最高权限。
- POU 加密在此简易说明了使用，但实际不同用户组对于工程各个功能的权限都是可以设置的，详细可见“工程”-“用户管理”-“授权...”。





- 对要 POU 程序设置“访问控制”权限时，建议对还未放在 Task 里的 POU 进行鼠标右键设置“属性”-“访问控制”，如下图只有一个 POU 程序，名为“PLC_PRG”，设置“访问控制”权限时需对框出来的“PLC_PRG”进行设置而非 MainTask 里的“PLC_PRG”。（能保证无论该 POU 是否放在 Task 里都被设置了权限。若只对放在 Task 里的 POU 设置访问控制权限则 Task 外同一个 POU 是没有受到限制的）。



14-3. PLC 加密

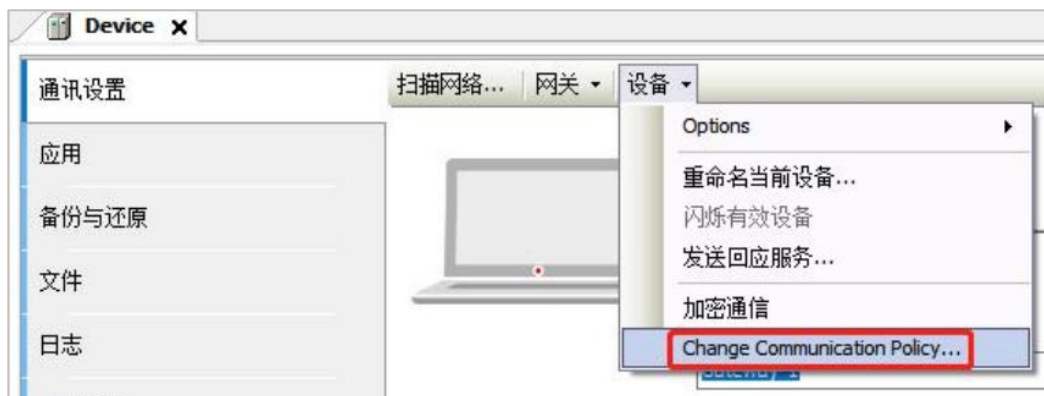
14-3-1. 功能说明

对 PLC 进行加密后，打开工程扫描到该 PLC 需要联机时，会弹出一个“设备用户登录”的对话框，需要输入正确的用户名和密码后，codesys 软件才能与该 PLC 联机进行控制。联机以后除非重新打开工程或新工程，否则无需再次输入用户名和密码，可以直接联机。

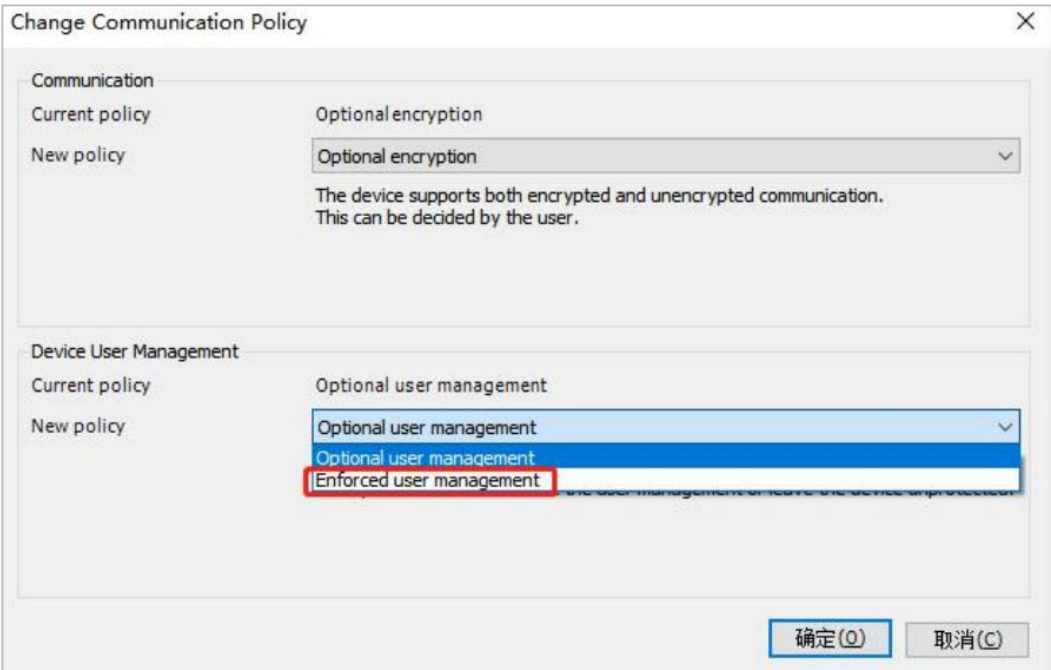


14-3-2. 配置方法一

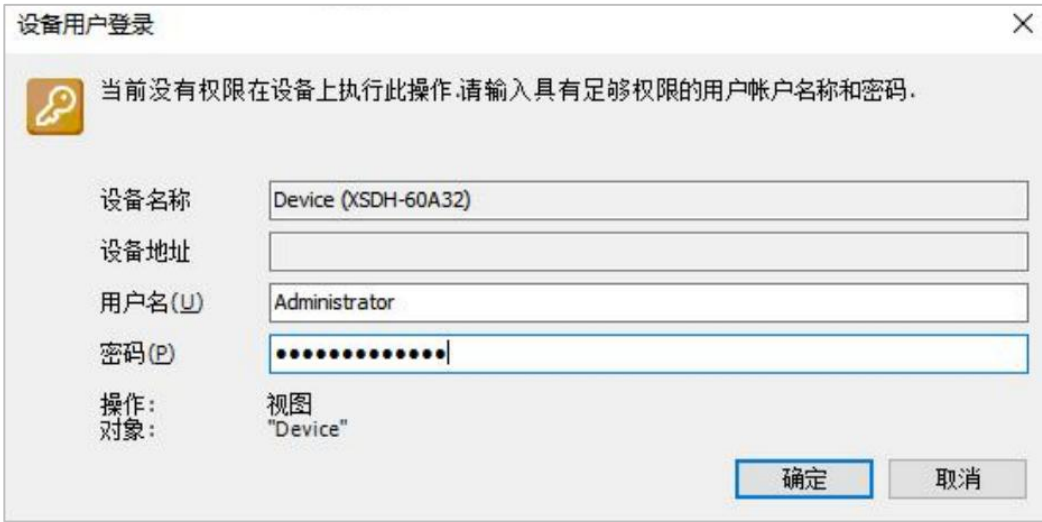
1、点击“Device” - “通讯设置” - “设备” - “Change Communication Policy...”。



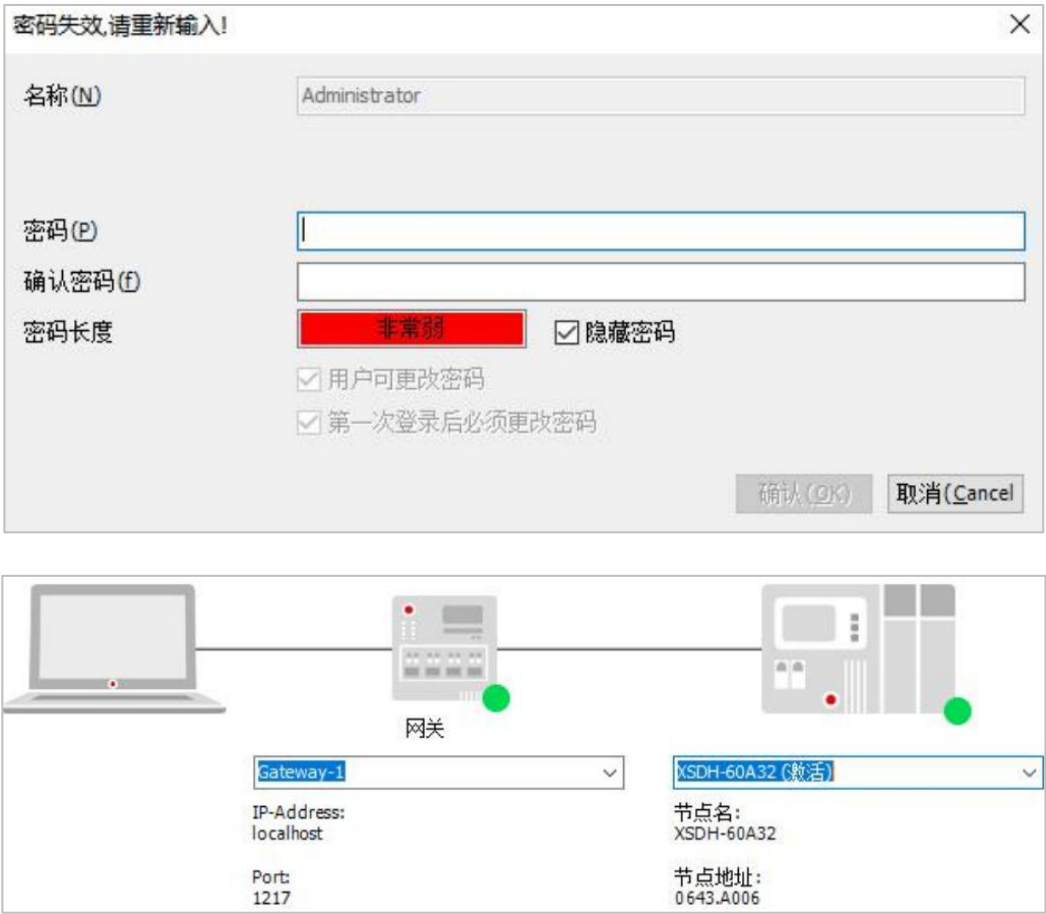
2、弹出的对话框“Current policy”默认是“Optional user management”，此时将“New policy”改为“Enforced user management”，点击确定。



3、此时点击“扫描网络...”，对扫描到的 PLC 进行联机时就会弹出“设备用户登录”的对话框，默认的用户名和密码都是 Administrator，输入好后点击确定。



- 4、名称默认是 Administrator ，可以更改密码，输入自己设置的密码后点击确认，此时是联机的状态。



- 5、只要输入一次正确的用户名和密码联机，当前工程就无需重复输入密码联机，但若是该工程关闭再打开、或打开其他工程，则需要输入用户名和密码才能联机。

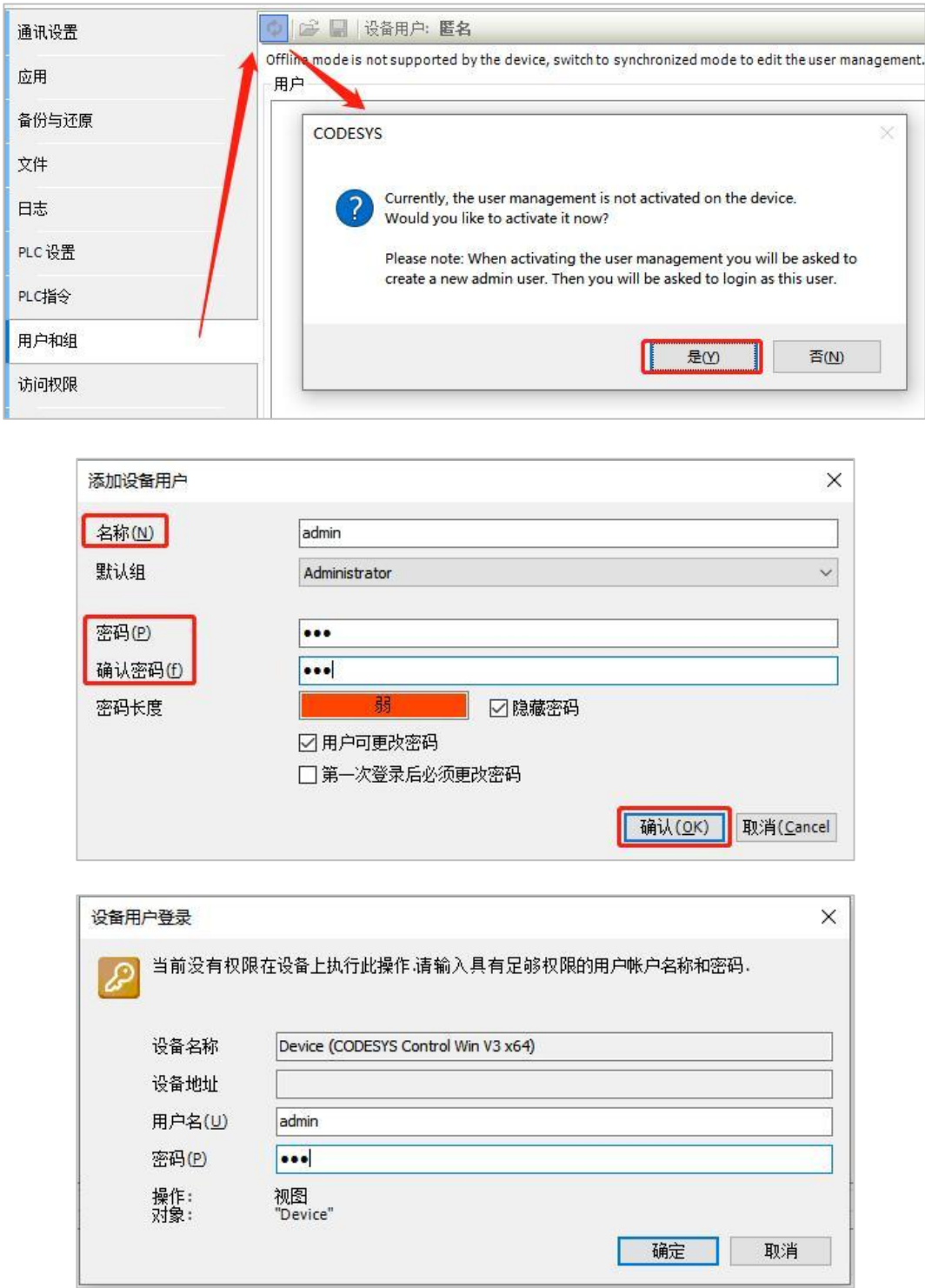
14-3-3. 配置方法二

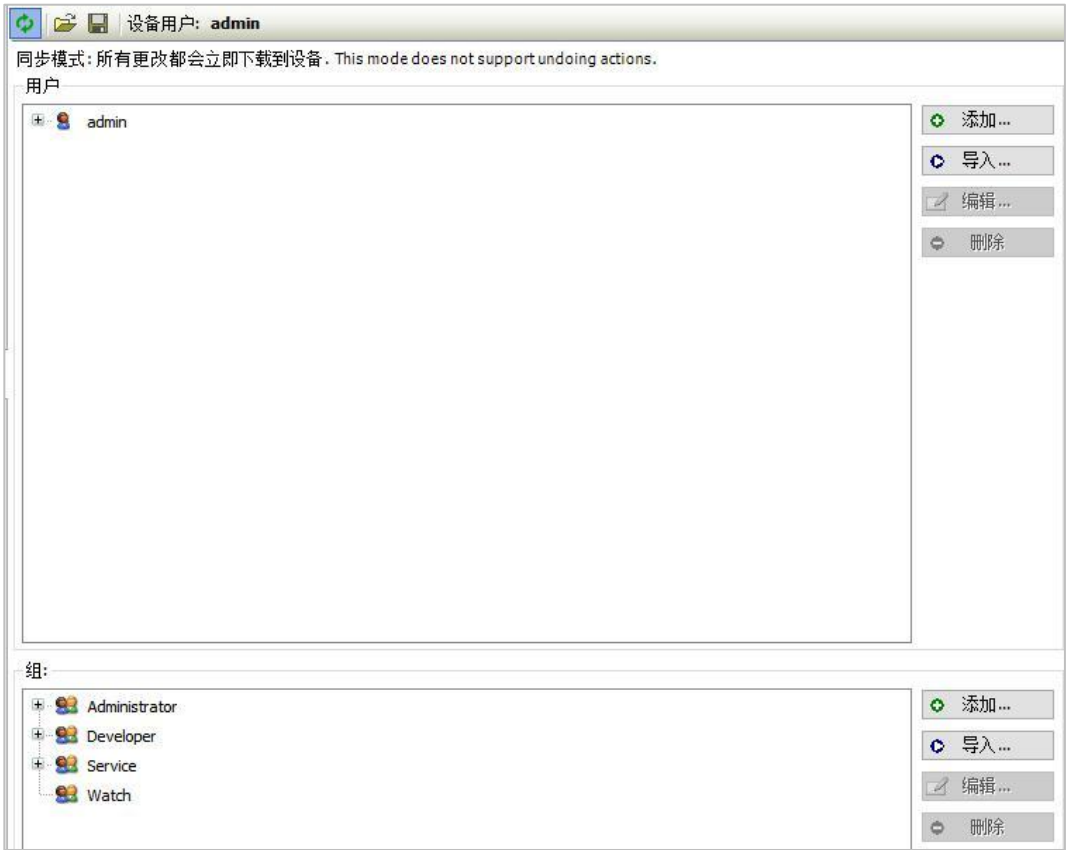
“Device” - “用户和组”：使用软 PLC 举例：

1、codesys 软件与软 PLC 联机。

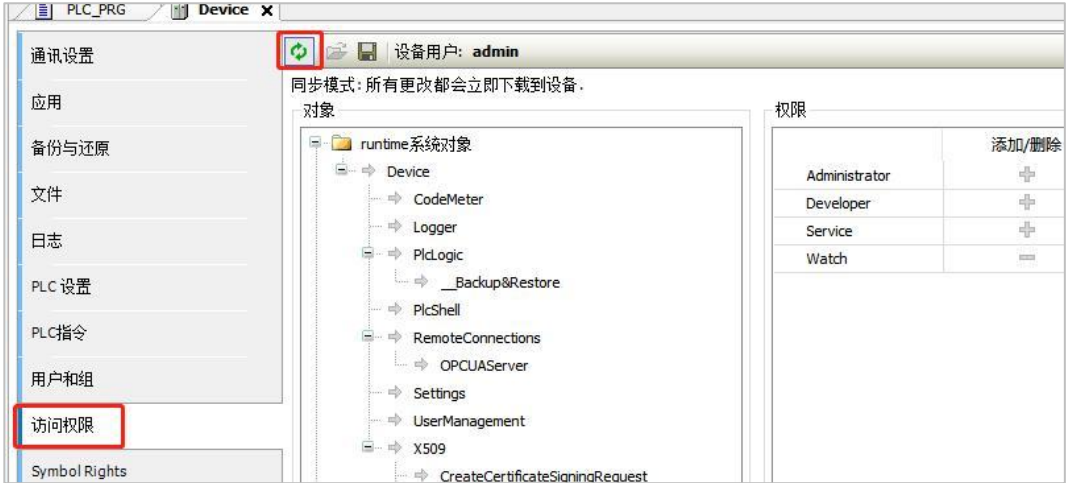


2、 点击 “Device” - “用户和组”，点击刷新图标，提示需要新建管理用户。





3、组 Administrator 拥有最高的权限，具体各个组的权限可在“Device”-“访问权限”里查看，访问权限里的内容一般不建议修改。

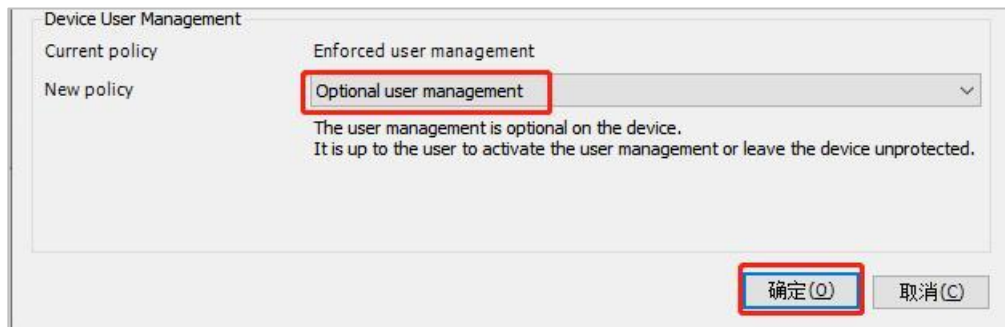


4、只要输入一次正确的用户名和密码联机，当前工程就无需重复输入密码联机，但若是该工程关闭再打开、或打开其他工程，则需要输入用户名和密码才能与该 PLC 联机。

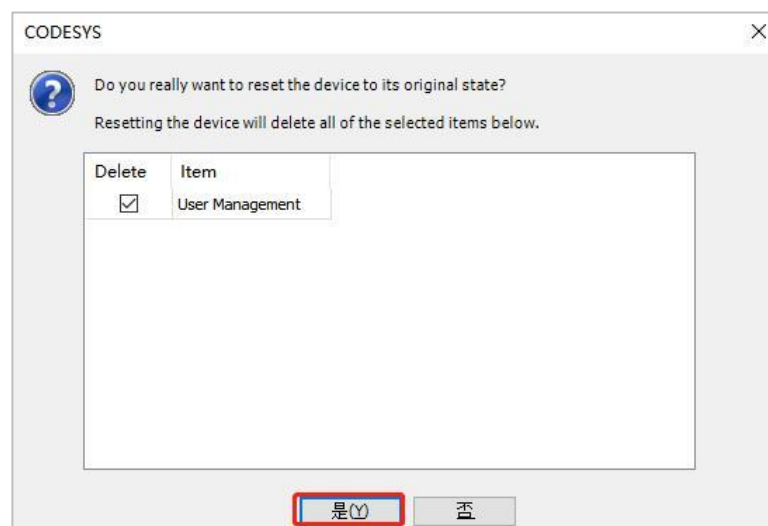
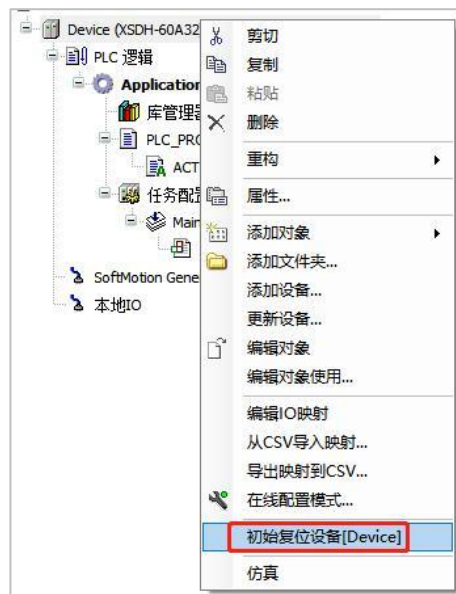
14-3-4. 取消设备加密

1、实体 PLC

- 1、codesys 软件需与该 PLC 联机上，然后点击“Device” - “通讯设置” - “设备” - “Change Communication Policy...”。 “New policy” 内容改为“Optional user management”，点击确定。

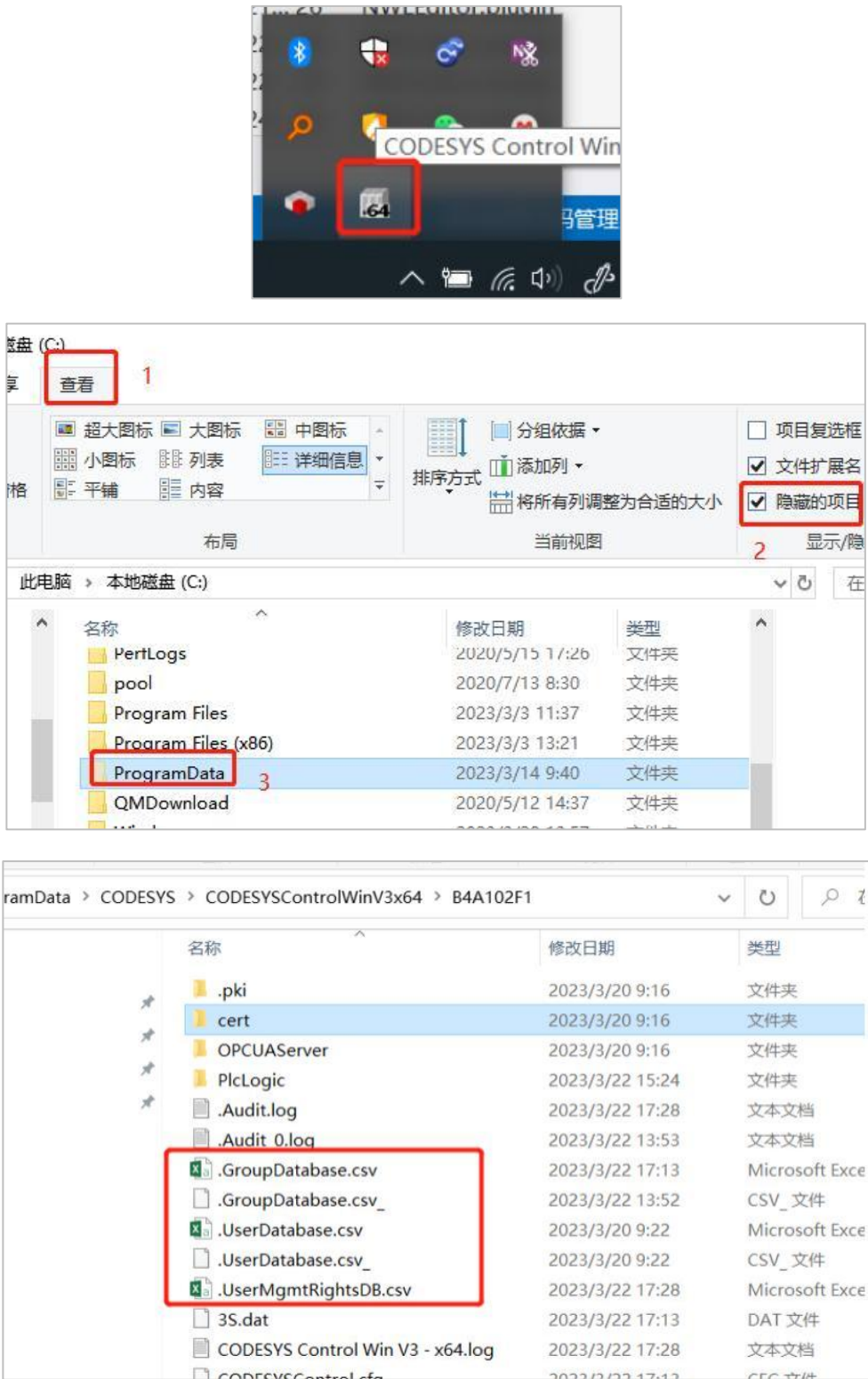


- 2、点击“Device”后点击鼠标右键，再点击“初始复位设备”，弹出的对话框点击是(Y)，此时就完成了取消设备加密即禁用用户管理操作了。

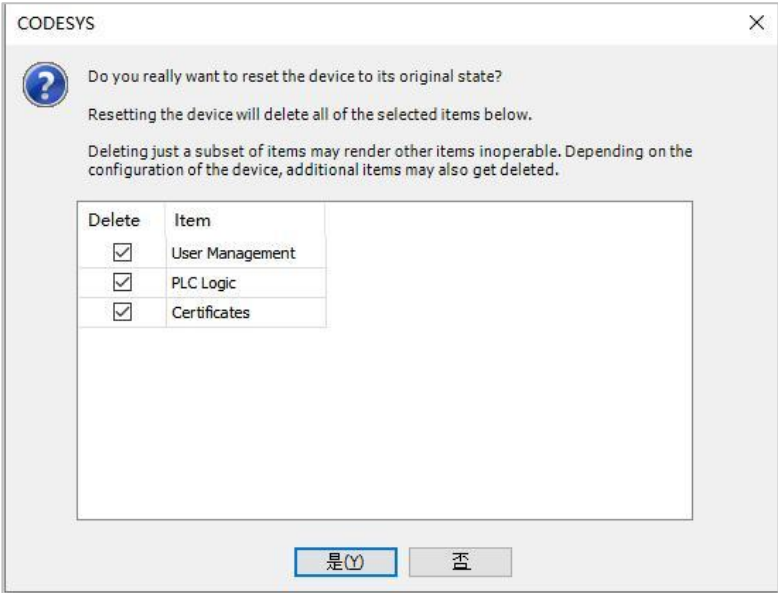


2、软 PLC

方法一：右键 Stop PLC，打开 C:\ProgramData\CODESYS\CODESYSControlWinV3x64（即找当前设备名称的文件夹）\XXXXXX 文件夹，将.csv 文件删除，重新启动 PLC，连接后重新设置用户名和密码。若上面路径没有找到.csv 文件，则可以直接在电脑里搜索文件名比如“.UserDatabase”，找到后选中右键打开其所在位置，将几个.csv 文件删除。



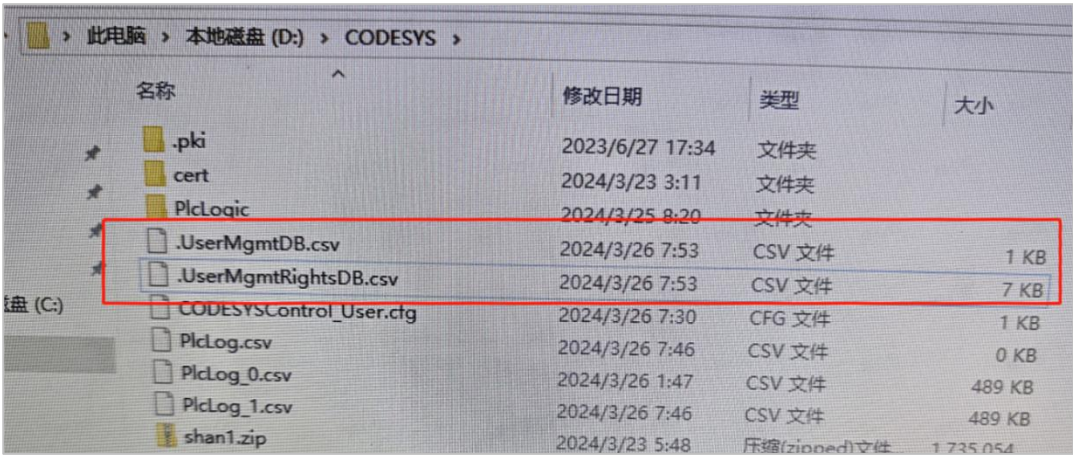
方法二：点击“Device”后点击鼠标右键，再点击“初始复位设备”，弹出的对话框点击是(Y)就可以了。



14-3-5. 注意点

1、激活管理员账户之后，若忘记账户和密码

- ARM 系列：XSDH、XSLH 刷固件可恢复。
- XSA-W 工控机系列：stop RTE 之后，删除 D:/CODESYS 下 2 个.csv 文件。如需设置新密码，输入默认账号密码：Administrator，Administrator，再重新设个密码就可以；若没有忘记密码，仅想要去掉密码，在调试模式下右击 Device，使用初始复位设备即可，会自动将后台中的这两个 CSV 文件删掉。



- XSA-L 工控机系列：刷固件可以恢复。
- XSF5：先联系技术支持沟通好后，再寄回给技术支持处理。

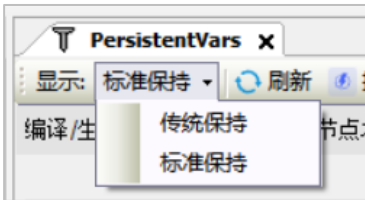
2、产品出厂密码设置

- 若需使用到账户，权限等操作需激活管理员账户，激活账户为：Administrator；密码：Administrator；激活完成之后，需设置新的管理员账户和密码。

15. 掉电保持

掉电保持变量表包含两种掉电保持方式，分为“传统保持”和“标准保持”。

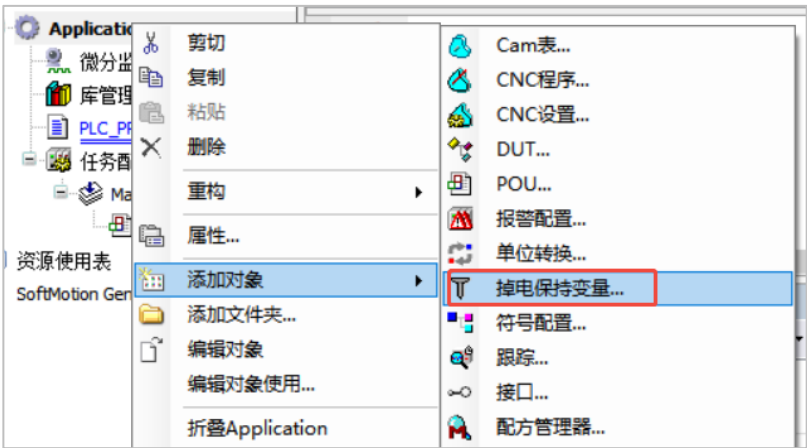
两种掉电保持方式互斥，切换时会弹出对话框提示变量表会被清空是否继续执行，用户按照自己所需选择是否继续即可。



15-1. 传统保持

15-1-1. 配置步骤

1、在左侧设备树内右击“Application”，点击“添加对象”，从菜单项点击“persistent 变量”，添加后在设备树内生成 persistent 变量节点。如下如图所示



2、选择“传统保持”模式：



3、在 PLC_PRG 编辑器内声明类别列选择标志为“PERSISTENT”的变量，如图所示；

PROGRAM PLC_PRG										
	类别	名称	地址	数据类型	初值	网络公开	保持	常量	注释	特性
1	VAR PERSISTENT	newVar		BOOL		默认	☒	☐		

4、编译信息提示如下：

消息 -总计0个错误, 1警告, 0条消息	
编译	0个错误 1个警告 0个消息
描述	
----- 开始构建应用程序 Device.Application -----	
此应用程序是最新的	
C0244: 在变量PLC_PRG.newVar中找到VAR_PERSISTENT列表中没有匹配的实例路径。 使用命令“添加所有实例路径”将所有实例路径添加到VAR_PERSISTENT列表。	
完整编译 -- 0 错误, 1 警告	

5、在掉电保持列表界面，点击鼠标右键，选择“添加所有实例路径”，变量的保持属性方可生效。之后即可下载程序，给变量赋值，断电重启后数值还在，实现了掉电保持的功能。

显示: 传统保持 取消RETAIN标志

	类别	名称	地址	数据类型	初值	网络公开	保持	常量	注释	特性

剪切

复制

粘贴

删除

全选

重构

导入变量...

导出变量...

编辑变量声明表标题...

插入

下移

上移

添加所有实例路径

重新排序并清除差异

保存当前值到配方

消息 -总计0个错误, 1警告, 0条消息

编译

错误 1个警告 0个消息

显示: 传统保持 取消RETAIN标志

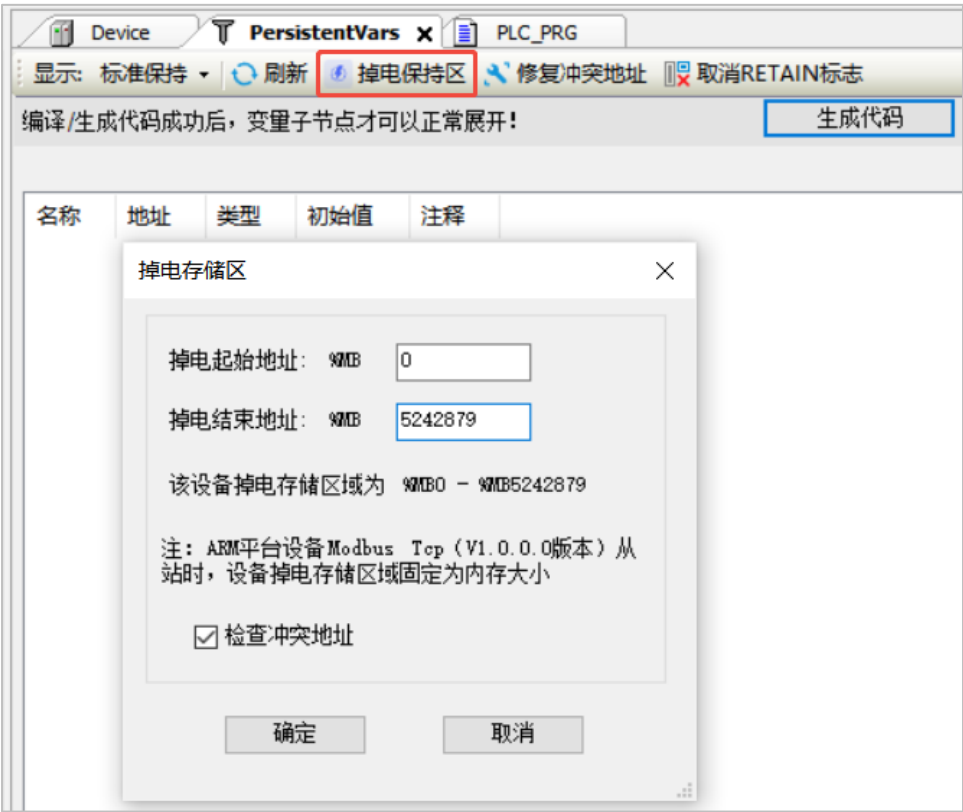
	类别	名称	地址	数据类型	初值	网络公开	保持	常量	注释	特性
1	VAR_GLOBAL RETAIN PERSISTENT	PLC_PRG.newVar		BOOL		默认	☒	☐		掉电保持变量生成的实例路径

15-1-2. 保持范围

所有机型最大支持设置 512K（支持动态调整，例如 0-524287 或者 10-524297）。

15-2. 标准保持

地址映射到 M 掉电存储区的变量可防止受到电源故障的影响。变量值在重启后仍然可以保留。



注意：支持标准保持功能的 XS 机型、固件版本及 XML 版本列举如下：

固件及 XML 版本		固件版本	XML 版本
XS 机型			
XSDH	2.4.1 及以上	2.4.1 及以上	2.4.1 及以上
XSLH			
XSF5			
XSA	2.4.2c 及以上	2.4.2 及以上	2.4.2 及以上

15-2-1. 变量定义

可通过表格和文本两种方式进行定义，以在程序（PRG）中定义为例。

15-2-1-1. 表格形式

在“PLC_PRG(PRG)”编程界面表格方式下，勾选“保持”列中的复选框，定义该变量为“掉电保持变量”，编译或生成代码后系统自动为其分配地址。



- 如果勾选了“保持”后，将地址映射到 I/Q 区时，会提示“I 区和 Q 区变量不支持保持”；

如果先映射 I/Q 地址，再勾选“保持”也是同样的效果，不支持此操作。点击确认后返回上一次正确的值。

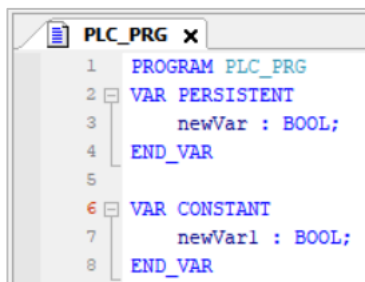
- 如果直接分配 M 区地址，则程序编译时会校验地址：
 - ◆ 如果变量分配的 M 区地址在 M 保持区，且没有勾选“保持”，则编译报错“变量占用掉电保持区地址，该区变量必须包含 PERSISTENT 特性!”；
 - ◆ 如果变量分配的 M 区地址不在 M 保持区，则不要勾选“保持”，否则编译报错“地址超限”；
 - ◆ 如果变量分配的 M 区地址既在 M 保持区又在非 M 保持区，则编译报错“地址超限”。



勾选“保持”后，会自动在变量类别中设置 PERSISTENT 属性；勾选“常量”后，会自动在变量类别中设置 CONSTANT 属性。

15-2-1-2. 文本形式

在“PLC_PRG(PRG)”编程界面文本方式下，通过对变量增加“PERSISTENT”关键字，定义该变量为掉电保持变量。(CONSTANT 表示常量)。



15-2-2. 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，否则该变量不具备掉电保持功能。

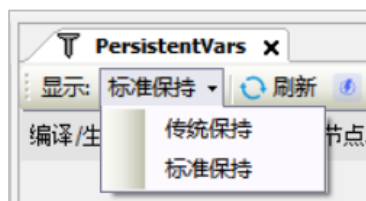
掉电保持变量表可通过以下两种方式生成：

- 手动添加：通过右键“Application” - “添加对象” - “PERSISTENT 变量”来添加掉电保持变量表；
- 自动添加：当声明了掉电保持变量，编译或生成代码时会自动创建掉电保持变量表（标准模式下支持）。

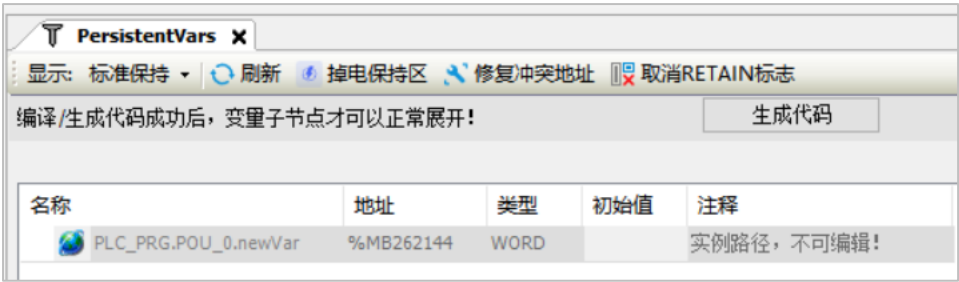
掉电保持变量表包含两种掉电保持方式，分为“传统保持”和“标准保持”。

💡 注意：

两种掉电保持方式互斥，切换时会弹出对话框提示变量表会被清空是否继续执行，用户按照自己所需选择是否继续即可。



可以通过下拉框切换到“标准保持”界面，此时工具栏会新增一些功能配置：刷新、掉电保持区、修复冲突地址、取消RETAIN标志

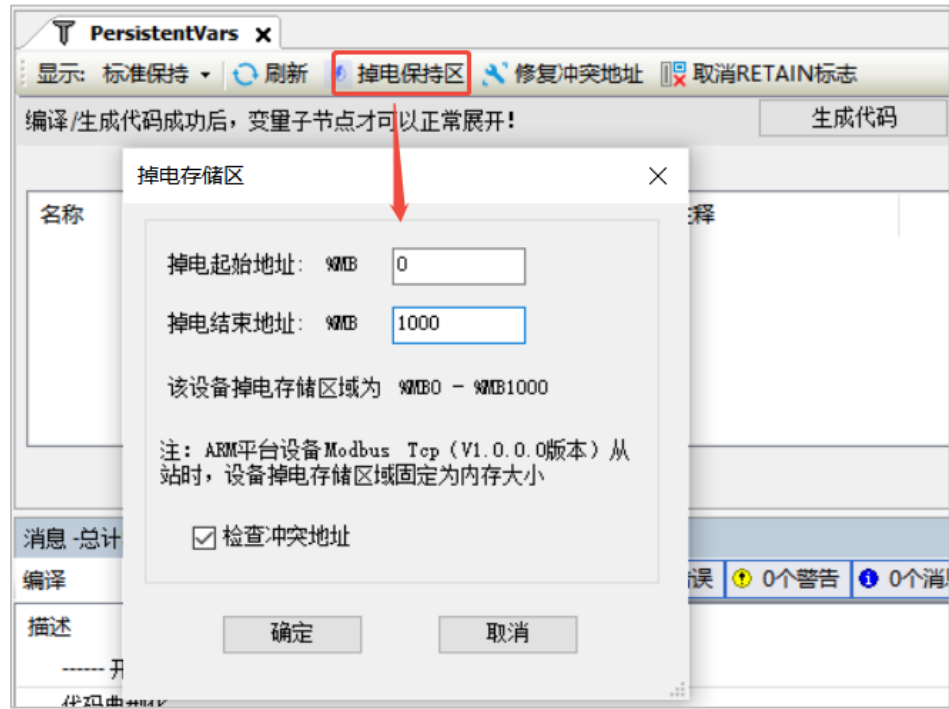


(1) 工具栏选项的功能

选项	功能
显示	可以通过下拉框方式选择“传统保持”或“标准保持”对应的变量表（默认选择“标准保持”）
刷新	1、可多次点击，登录在线后显示灰色不可点击 2、将外部工程的保持变量添加到标准保持模式变量表中 3、检测所有掉电保持变量地址的合法性
掉电保持区	掉电保持存储区域地址范围设置。登录在线后显示灰色不可点击
修复冲突地址	可以对掉电保持变量的冲突地址进行重新分配。登录在线后显示灰色不可点击
生成代码	1、可多次点击，登录在线后显示灰色不可点击 2、生成代码后会将外部工程的持久性变量添加到持久性变量表中 3、生成代码过程中，不支持操作
取消 RETAIN 标志	一键取消程序中所有变量的 retain 标志

(2) “掉电保持区”选项

参数配置界面如下图所示：



- 用于配置 M 区掉电存储区域起始、终止地址，默认掉电起始地址为 0、掉电结束地址为“内存大小-1”（即内存大小若为 262144，则范围为 0-262143）

- 掉电存储区地址修改时需注意结束地址不得小于起始地址，且起始、终止地址不能大于内存大小，否则会有错误提示。
- “检查冲突地址”默认是勾选的。勾选后会对所有掉电保持变量进行 M 区地址重复检查，当两个及以上变量的 M 区地址发生冲突，编译后会在消息窗口进行错误提示。

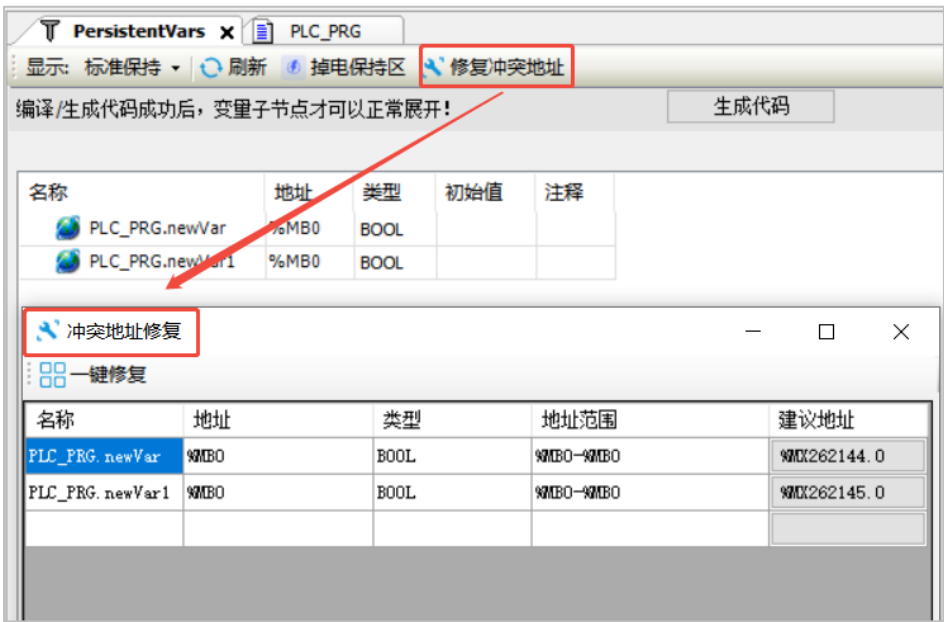


使用标准保持功能时，为了保证变量数据保存的正确性，原则上每个变量应该对应唯一的 M 区地址。若用户需要变量地址可以重复，可在“掉电存储区”选项中取消勾选“检查冲突地址”。

(3) “修复冲突地址”选项

当掉电变量地址发生地址冲突时，用户可以通过手动修改变量地址的方法，自助对冲突地址进行调整。

除此之外，也可以通过“修复冲突地址”选项来对冲突地址进行修复。参数配置界面如下图所示：



- 界面包括“一键修复”以及变量表格，无论是否勾选了“检查冲突地址”，表格都会展示所有存在地址冲突的变量。表格中包括变量的名称、地址、类型、地址范围和建议地址。
- 如果 M 保持区地址不够时，建议地址会填为空，点击“一键修复”或“建议地址”里的空白地址，会将空的地址写入（后续编译后，会对空地址进行地址分配，此时再报错地址不足，见地址分配）。

项目	功能	属性
名称	显示掉电保持变量的来源和变量名	不可编辑
地址	显示掉电保持变量的地址	不可编辑
类型	显示掉电保持变量的类型	不可编辑
地址范围	显示掉电保持变量当前的地址范围	不可编辑
建议地址	1、显示每个变量系统自动分配的地址 2、点击单个变量的建议地址把建议地址写入该变量，成功后将该变量从表格中删除 3、若点击空白栏的建议地址处，无任何效果	可点击
一键修复	1、点击后修复所有变量的地址，写入成功后将所有变量从表格中删除 2、若变量表为空，点击无任何效果	可点击

15-2-3. 掉电保持模式

可通过表格和文本两种方式进行定义，以在程序（PRG）中定义为例。

15-2-3-1. 模式对比

目前掉电保持变量包含传统保持和标准保持两种模式，二者的差异对比如下表所示：

选项	标准保持	传统保持
数据来源	全部来源于掉电保持变量表外部带有 Persistent 特性或映射到 M 区掉电保持范围内的变量	掉电保持变量表内部定义变量和外部带 Persistent Retain 特性变量
地址映射	所有的变量均需强制关联到 M 区掉电保存区	不需要强制关联
中间插入变量	由于变量是通过地址映射,因此插入或删除变量对其它变量保存值无影响	意外操作可能引起数据丢失（例：清除全部、PLC 设备更新，编译选项修改等）
配方功能	支持配方功能	只能使用外部独立配方功能

15-2-4. 掉电保持规则

- 变量名变化时，掉电保持变量的值将被初始化为初始值。
- 变量地址、初始值、注释、变量类别和特性发生变化时，掉电保持变量的值将保持原来的值。
- 变量类型变化，包括变量的类型名称变化、变量的类型成员变化、变量的类型成员类型变化。
 - ◆ 掉电保持变量的类型名称变化时，掉电保持变量的值将被初始化为初始值。
 - ◆ 掉电保持变量的类型名称未发生变化，但类型成员发生变化时：
 - ◆ 结构体/功能块：没有变化的成员的值将保持原来的值，变化的成员的值将被初始化为初始值。
 - ◆ 联合体/枚举：当其类型大小减少时将被初始化为初始值。
 - ◆ 数组：当数组类型变量的基类型发生变化时，数组变量将被初始化为初始值（数组类型变量的基类型为结构体且仅增删改结构体成员时，没有变化的结构体成员将保持原来的值）；当数组类型变量的基类型未发生变化时，相同索引的元素值将保持原来的值。

15-2-5. 地址分配

15-2-5-1. 自动分配地址规则

若用户程序中变量勾选了保持属性，但并未分配地址，只需单击工具栏中的“编译”或掉电保持变量表界面的“刷新”，系统会自动为其分配地址。掉电保持范围默认设置为 0-6291456(6M)，变量起始地址计算方法举例说明如下：

- 考虑用户使用 MODBUS 地址（线圈 MB0-MB65534，寄存器:MW40000-MW105534。），初次分配时选取“%MB262144”为起始地址;只有当尾部地址使用完或者无法再为变量分配地址，再从%MB0 开始从头向后分配。

- 再与现有分配的地址进行冲突检测，若地址冲突则抛弃该地址，从下一个可用地址重新计算，再次检测直至地址不再冲突；若未找到合法地址，则变量保持属性配置失败。
- 只有当所有变量都能分配到合法地址时，才将地址同步到工程中。
- 在对变量进行地址分配时，有几种数据类型需要满足四字节对齐原则，即变量初始地址能够被 4 整除。变量类型如包括：
 - ◆ 自定义数据结构；
 - ◆ 枚举类型；
 - ◆ REAL 或 LREAL 类型；
 - ◆ 基类型数据为上述三种类型的数组 Array；

如果变量不满足四字对齐原则，则输出信息提示用户“变量地址不与“4”字节对齐”。

15-2-5-2. 地址异常处理

1、地址冲突时

- 可通过“修复冲突地址”功能，系统自动对地址冲突得变量进行地址重新分配，用户手动确认修复后生效。
- 可通过在掉电保持变量表右击菜单中，选择“清除全部变量地址”来清除所有变量的地址，再通过编译对所有变量进行地址重新分配。

2、掉电保持区不足时

- 可在掉电保持变量表中的“掉电保持区”中修改起始、结束地址扩大范围。
- 可在 Application-目标内存属性中的内存大小，来增大“掉电保持区”可设范围。

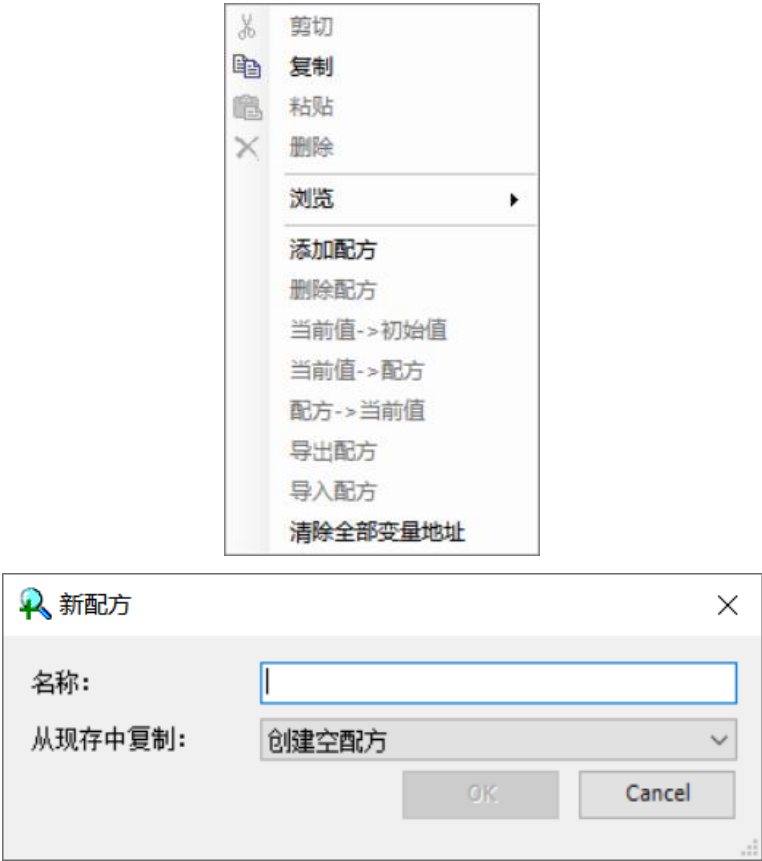
15-2-6. 配方操作

标准模式下，掉电保持变量表可以新增配方列，用来保存一组变量的数据，作为掉电保持变量写入值。

在掉电保持标准模式下，用户可以在掉电保持变量表中右键单击“配方”条目来执行“添加配方”、“删除配方”、“当前值->配方”、“配方->当前值”、“导出配方”和“导入配方”等操作。

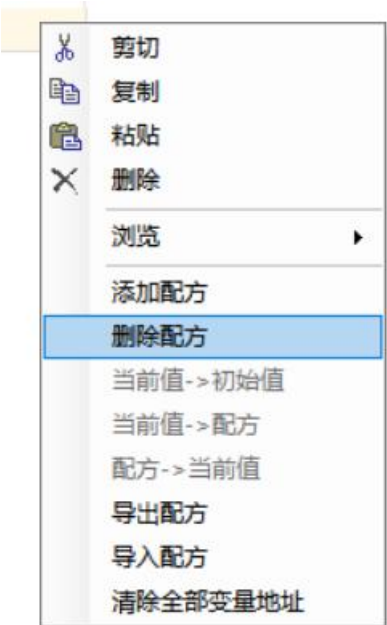
15-2-6-1. 添加配方

可在掉电保持变量表中右键菜单中选择“添加配方”，此操作后在掉电保持变量表最后一列后添加新的配方列。在弹出的“新配方”设置对话框中设置新配方列的名称，也可选择从已有的配方列中复制。



15-2-6-2. 删除配方

选择配方列，通过右击菜单中的“删除配方”可删除对应配方列。



15-2-6-3. 配方列表功能

选项	功能
添加配方	可以新增一行配方列
删除配方	右击对应配方列的时候，可以选择删除该配方列
当前值->初始值	仅登录在线时支持配置
当前值->配方	仅登录在线时支持配置
配方->当前值	仅登录在线时支持配置
导入、导出配方	与配方功能一样，可以导出为.txtrecipe 文件
清除全部变量地址	可以将当前界面中变量对应的地址全部清除，并同步到变量定义处，再通过编译对所有变量进行地址重新分配

15-2-7. 掉电保持说明

- 当掉电保持变量数据结构发生变化时，系统会保存原结构掉电保持变量已存在数据，对新增变量则会进行初始化操作。

例 1：结构体成员新增

```
TYPE DUT :  
STRUCT  
    newstruct: LREAL;  
    newstruct1: LREAL;  
    newstruct2: LREAL;  
END_STRUCT  
END_TYPE
```

→

```
TYPE DUT :  
STRUCT  
    newstruct: LREAL;  
    newstruct1: LREAL;  
    newstruct2: LREAL;  
    newstruct3: LREAL;  
END_STRUCT  
END_TYPE
```

原参数数值掉电保持

新增参数数值默认初始化

例 2：数组成员新增

GVL_AA	%MB262144	ARRAY [0..9] OF INT
GVL_AA[0]	%MB262144	INT
GVL_AA[1]	%MB262146	INT
GVL_AA[2]	%MB262148	INT
GVL_AA[3]	%MB262150	INT
GVL_AA[4]	%MB262152	INT
GVL_AA[5]	%MB262154	INT
GVL_AA[6]	%MB262156	INT
GVL_AA[7]	%MB262158	INT
GVL_AA[8]	%MB262160	INT
GVL_AA[9]	%MB262162	INT

→

GVL_AA	%MB262144	ARRAY [0..10] OF INT
GVL_AA[0]	%MB262144	INT
GVL_AA[1]	%MB262146	INT
GVL_AA[2]	%MB262148	INT
GVL_AA[3]	%MB262150	INT
GVL_AA[4]	%MB262152	INT
GVL_AA[5]	%MB262154	INT
GVL_AA[6]	%MB262156	INT
GVL_AA[7]	%MB262158	INT
GVL_AA[8]	%MB262160	INT
GVL_AA[9]	%MB262162	INT
GVL_AA[10]	%MB262164	INT

原数组成员数值掉电保持

新增数组成员首次下载，数值初始化

- 当掉电保持变量仅地址发生变化时，系统会将原地址数据拷贝到新地址内存中，不会影响掉电保持变量数据变化。

```
PROGRAM PLC_PRG  
VAR RETAIN_PERSISTENT  
    a AT %MB262144 : ARRAY [0..9] OF INT;  
END_VAR
```

→

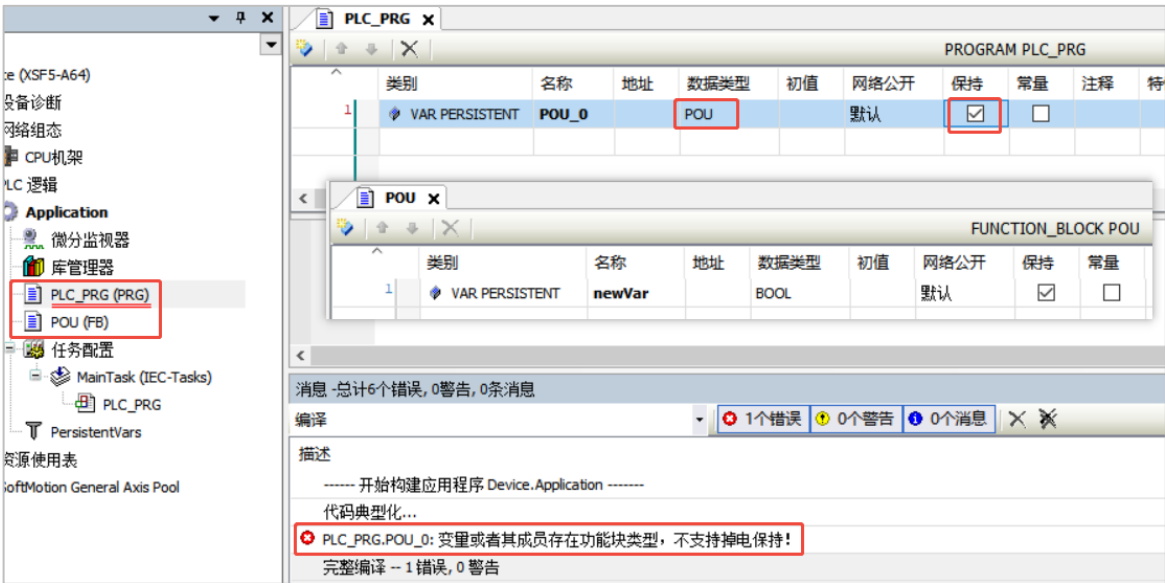
```
PROGRAM PLC_PRG  
VAR RETAIN_PERSISTENT  
    a AT %MB302144 : ARRAY [0..9] OF INT;  
END_VAR
```

15-2-8. 变量支持情况说明

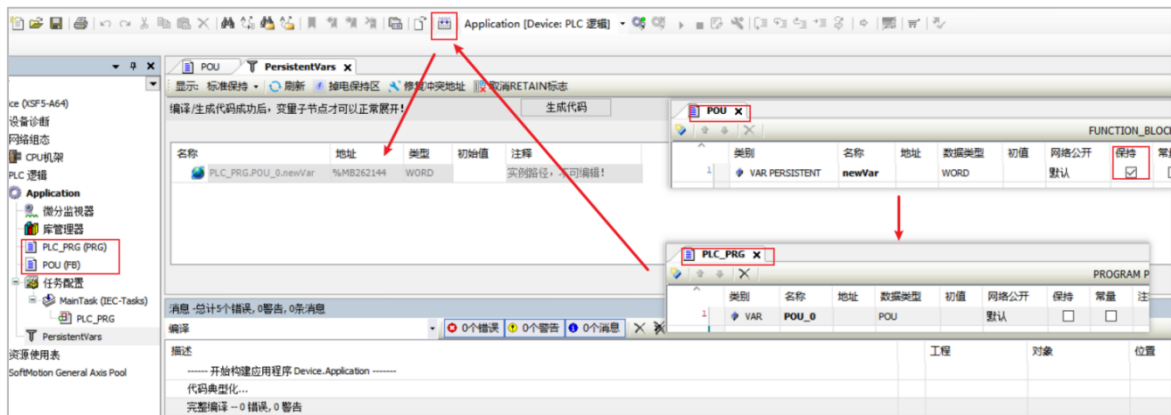
掉电保持变量支持在全局变量表 (GVL)、程序 (PRG)、功能块 (FB) 中进行勾选“保持”实现，不支持在函数、方法 (METH)、属性 (Prop)、结构体 (STRUCT)、联合体 (Union)、枚举 (Enum)、别名 (Alias) 中进行勾选“保持”。以下类别需要特殊处理：

1、功能块 FB

只能在 FB 中对其成员设置保持属性，对声明成 FB 实例无法设置保持属性。
对于声明成 FB 实例的变量，勾选保持属性/分配到 M 区保持地址编译后，报错提示如下图（可以通过双击错误信息定位到错误处）。



编译等操作后，会将 FB 类型的变量中具有掉电保持属性的变量生成实例路径，并添加到掉电保持变量表中；



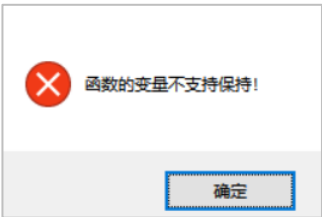
FB 内部变量勾选保持属性编译后，内部成员不会自动分配地址，只会在编译后将每个实例自动生成实例路径，然后系统自动对实例路径分配地址，该实例在掉电保持变量表中以灰色显示，不可修改地址；



FB 块里有的变量映射保持的 M 区（或勾选“保持”），然后数据就可以保持住；有的变量既没有映射保持的 M 区（也没有勾选“保持”），则无法保持。

2、函数 FUN

FC 内部不支持勾选保持，勾选时报错提示如下图：



3、不支持保持功能的内容

方法（METH）、属性（Prop）、结构体（STRUCT）中定义的变量不支持设置 M 区保持。

- 结构体（STRUCT）、方法（METH）以表格形式展示时，不存在保持属性列，属性（Prop）维持原状不支持表格形式；
- 以文本形式展示时，功能与官方保持一致；
联合体（Union）、枚举（Enum）、别名（Alias）中定义变量形式不支持设置 M 区保持。

15-2-9. M 区范围说明

XS F5 系列最大为 10M 掉电保持大小,XSDH/XSLH 系列、XSA 系列最大为 6M 掉电保持大小。

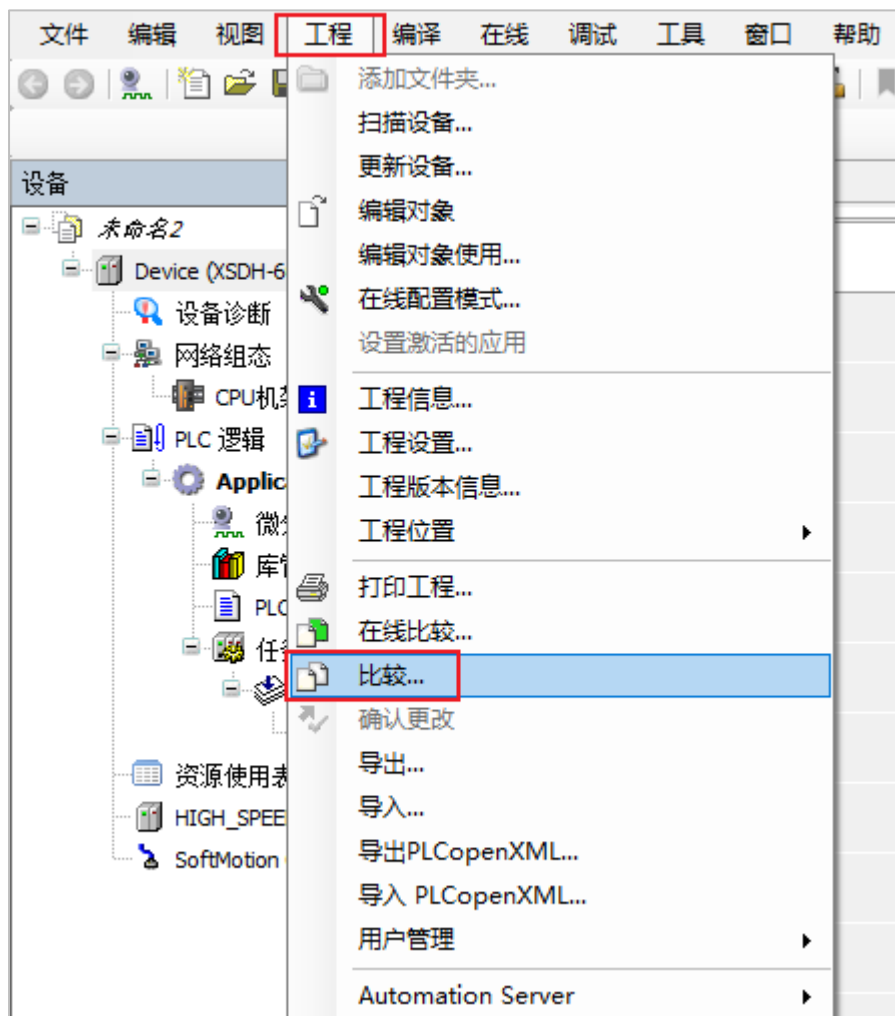
机型	M 区默认值	M 区最小值	M 区最大值
XSDH、XSLH	3M	0	6M
XS F5	5M	0	10M
XSA	3M	0	6M

16. 工程比较

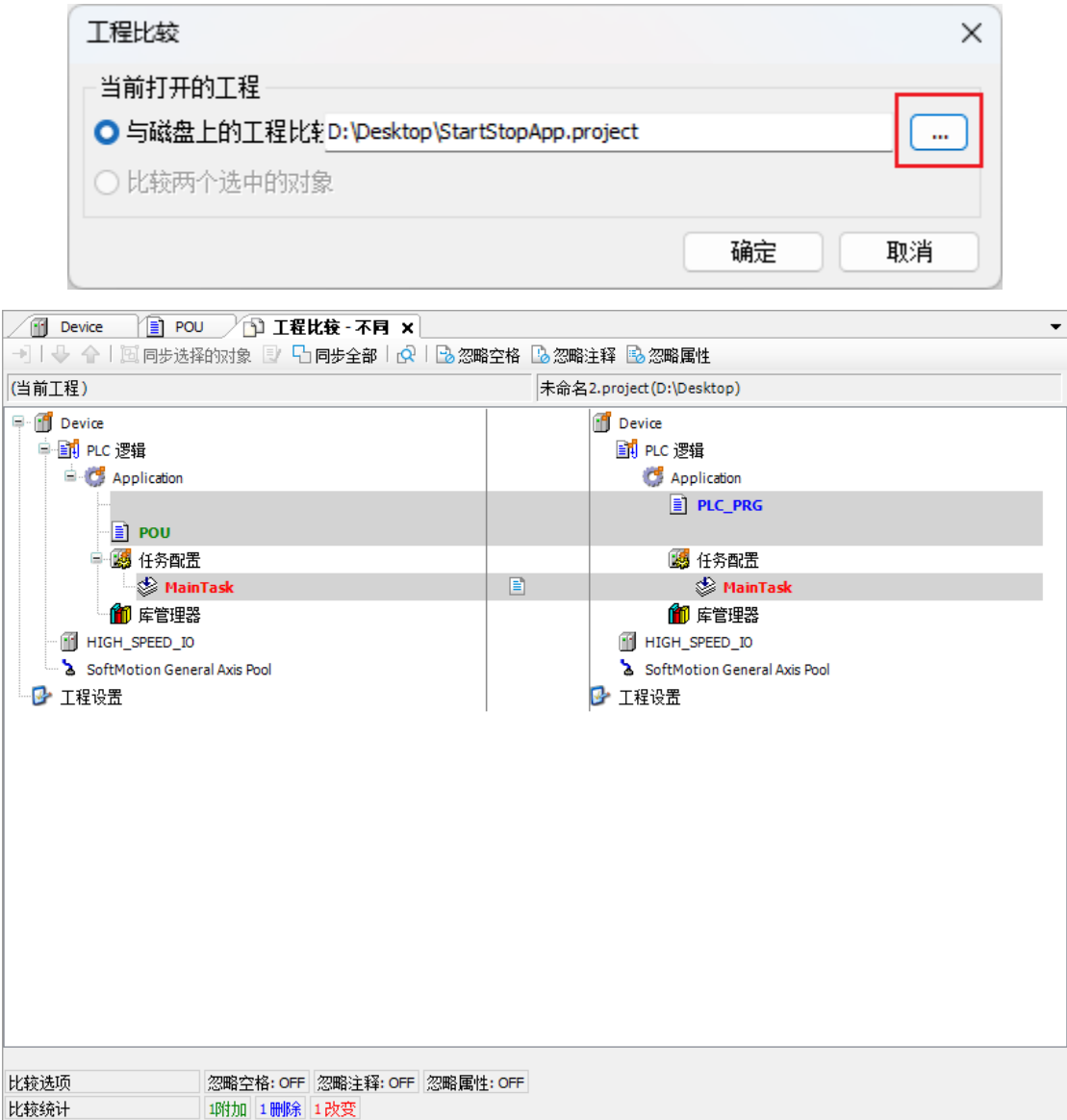
16-1. 离线工程比较

16-1-1. 工程比较

- 1、打开 XS 软件并打开本地工程
- 2、在菜单栏选择“工程”-“比较”，打开“工程比较”对话框。



- 3、单击“...”，选择磁盘上的工程，单击“确定”，打开“工程比较-不同”界面。左侧显示当前工程，右侧显示磁盘上的工程；界面底部显示比较选项开启情况和比较统计结果。



工程比较界面工具栏图标功能及含义如下表所示：

图标	含义
	打开对象的详细比较视图 使能条件：选中的对象内容不同
	返回至工程对象的比较视图 使能条件：当前视图是对象详细比较视图
	选择下一个存在差异的对象/块 使能条件：选中的对象/块下面存在差异的对象/块
	选择上一个存在差异的对象/块。 使能条件：选中的对象/块上面存在差异的对象/块
	同步选择的对象（支持多选） 使能条件：选中的对象包含未同步的差异
	恢复选择对象至同步前状态（支持多选） 使能条件：选中的对象所有差异部分处于同步状态
	接受对象的属性和权限访问 使能条件：选中的对象具有不同的属性和访问权限。
	同步所有差异对象 使能条件：对象列表里包含未同步差异的对象

图标	含义
	恢复所有差异对象至同步前状态 使能条件：对象列表里所有差异对象的差异部分处于同步状态
	切换对象比较模式，选中后，可以选择左右两边任意两个对象做比较 说明：选中对象的图标右上角将显示一个放大镜标识
	选中后，语义相关的空格（例如字符串文字）不忽略外，其他空格都将忽略
	选中后，将忽略注释的差异
	选中后，将忽略对象的属性或访问权限的差异
	确认修改，同步后点击该按钮，可保存同步后结果
	接受改变区域 使能条件：当前鼠标焦点所在区域是未同步的差异区域。
	恢复改变区域 使能条件：当前鼠标焦点所在区域是同步后的差异区域。

说明：

- 同步过程即将磁盘上的工程对象差异部分同步至当前打开的工程对象中
- 上述图标原生部分维持现状即可，差异部分参照图标文件夹。工程比较界面差异部分使用图标符号和字体颜色区分，具体说明如下表所示：

图标&字体颜色	含义
黑色字体	对象相同，无差异
比较对象图标右上角显示 	对象的子对象不同，存在差异。
蓝色字体	对象仅在右侧磁盘工程中存在。
绿色字体	对象仅在左侧当前打开的工程中存在。
	对象具有不同的属性。
	对象具有不同的访问权限。
	对象的实现不同。
	将右侧磁盘工程中对象同步至当前打开工程中。
	在当前打开工程中删除对象。
	同步右侧磁盘工程中对象的属性。
	同步右侧磁盘工程中对象的访问权限。
	同步右侧磁盘工程中对象。

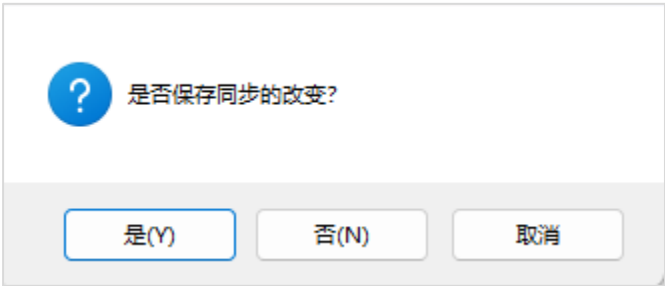
4、根据不同使用场景执行相应操作。

使用场景	操作步骤
同步选择对象	1、单击同步对象：鼠标单击选中待同步对象，或通过   选中，单击  同步选择对象 2、多选同步对象：支持【Ctrl】【SHIFT】多选待同步对象，单击  同步选择对象 说明：需要恢复选择对象至同步前状态，单击  恢复选择对象
同步全部对象	单击  同步全部

使用场景	操作步骤
	说明：如果如需恢复所有差异对象至同步前状态，单击  恢复全部
同步对象的属性和权限访问	1、选中具有不同属性和权限访问的对象，单击  2、在打开的对话框中单击“确定”
比较任意两个对象	1、单击  ，切换对象比较模式。 2、在左侧工程选择待比较对象，在右侧工程选择待比较对象。 3、在工具栏单击“比较”，进入对象的详细比较视图。 4、选中待详细比较的内容，单击  同步块 同步块，同步该差异部分。 说明：如需恢复改变区域，单击  恢复块 恢复块
开启忽略空格	单击  说明：如需关闭忽略空格，再次单击该图标即可。
开启忽略注释	单击  说明：如需关闭忽略注释，再次单击该图标即可。
开启忽略属性	单击  说明：如需关闭忽略属性，再次单击该图标即可。
进行对象详细比较	1、选中待详细比较的对象，单击  或双击该对象，进入对象详细比较视图 2、选中待详细比较的内容，单击  同步块 同步块，同步该差异部分。说明：如需恢复改变区域，单击  恢复块 恢复块

5、保存同步结果。

1) 关闭工程比较视图或关闭工程详细比较视图，提示“是否保存同步结果”。



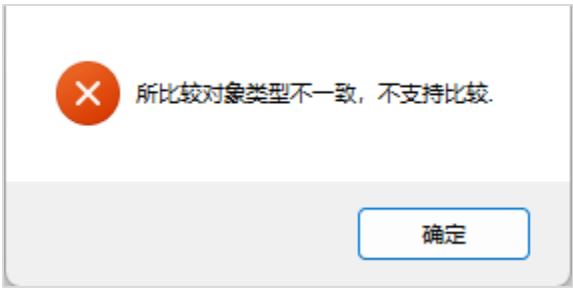
2) 单击“是”，保存同步结果。

16-1-2. 对象比较

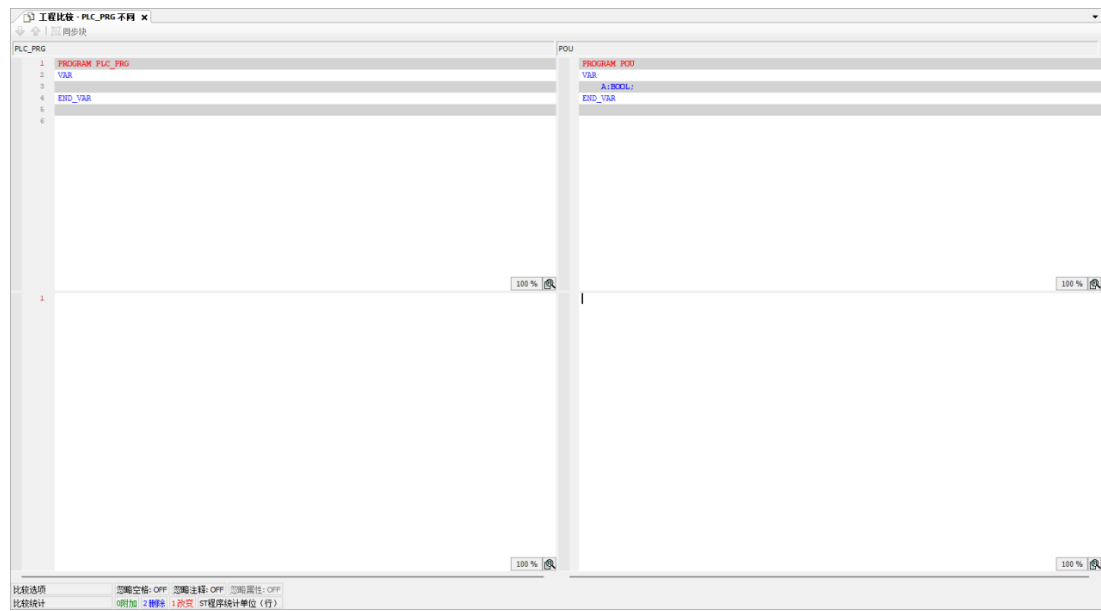
- 1、打开 XS 软件并打开本地工程
- 2、在设备树中选择需要比较的两个对象，然后在菜单栏选择“工程” - “比较”，打开“工程比较”对话框。



注意：选择不同类型对象时无法比较，提示以下信息。



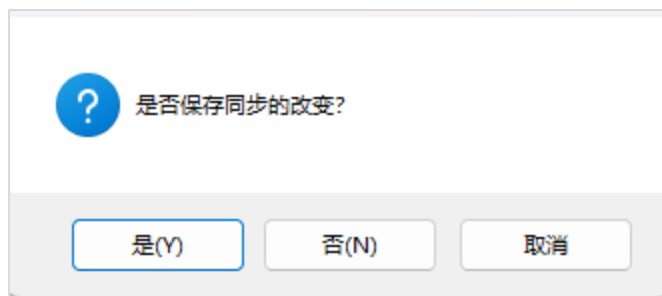
- 3、单击“确定”，打开“工程比较”界面。



- 4、选中待详细比较的内容，单击  同步块，同步该差异部分。

如需恢复改变区域，单击  恢复块。

5、关闭工程详细比较视图，打开是否保存同步结果的对话框。



6、单击“是”，保存同步结果。

16-2. 在线工程比较

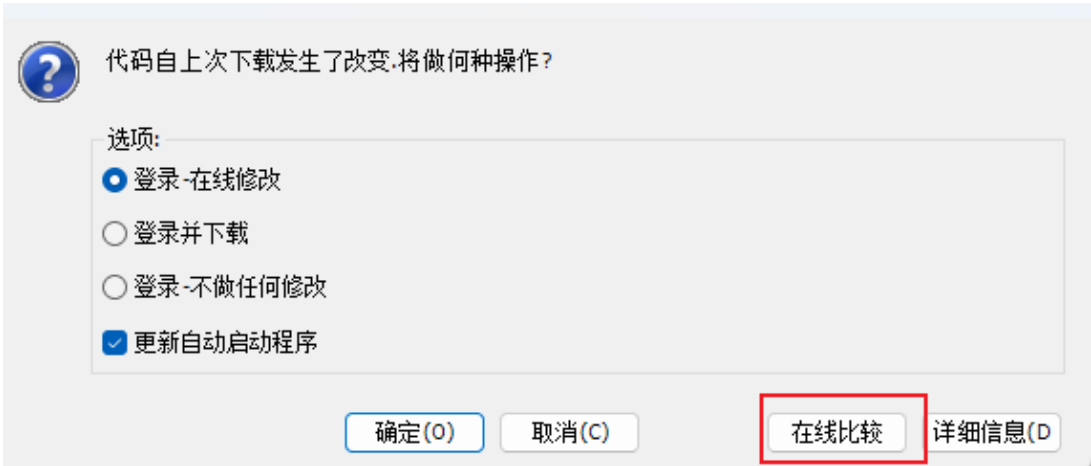
16-2-1. 适用设备

功能 XS 机型及版本	在线工程比较 (XS StudioV2. 3. 2 及以上支持)
XSDH	固件版本 V2.4.2c 及以上支持
XSLH	固件版本 V2.4.2c 及以上支持
XSF5	固件版本 2.4.2 及以上支持
XSA-W	不支持
XSA-L	固件版本 2.4.2 及以上支持

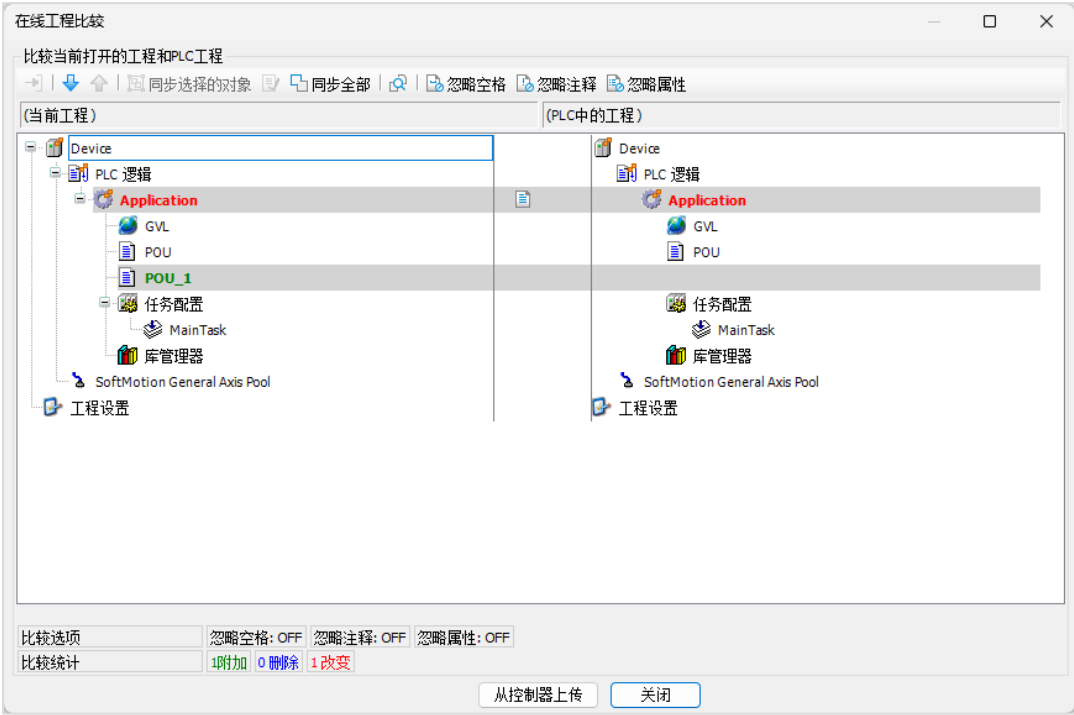
16-2-2. 操作说明

1、打开在线工程比较界面

方式一：用户程序下载时进入
下载程序时打开如下图所示对话框。



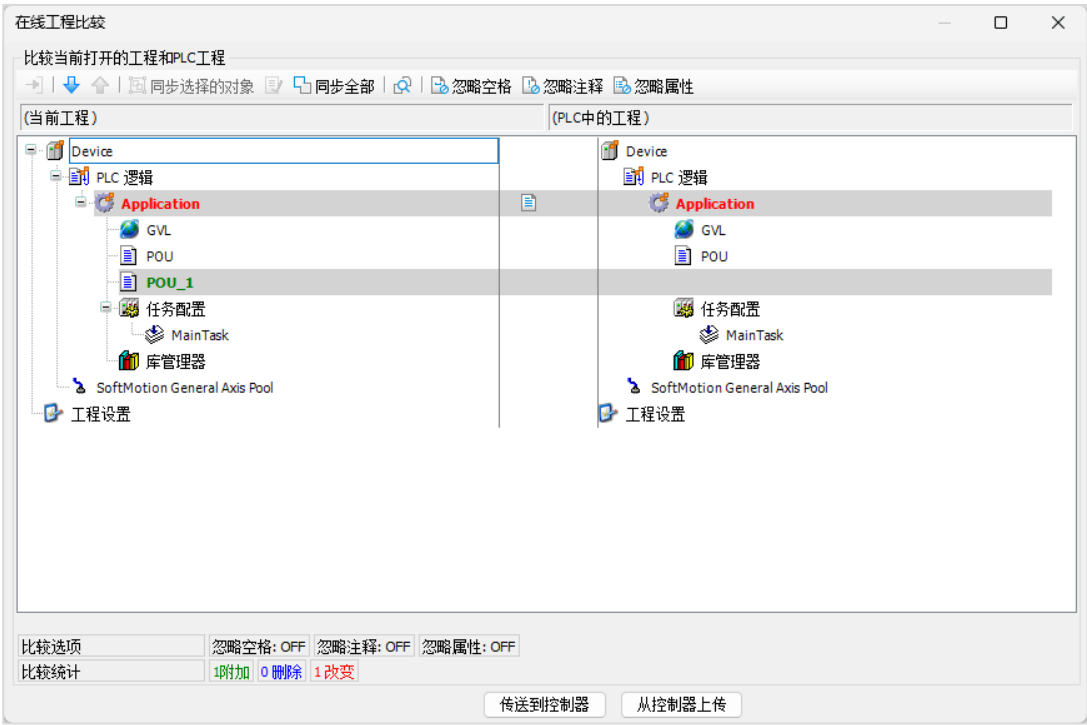
单击“在线比较”，打开“在线工程比较”对话框。



在线工程比较操作具体请参见离线工程比较操作部分内容。

方式二：菜单进入。

在菜单栏选择“工程”-“在线比较”，打开“在线工程比较”对话框。



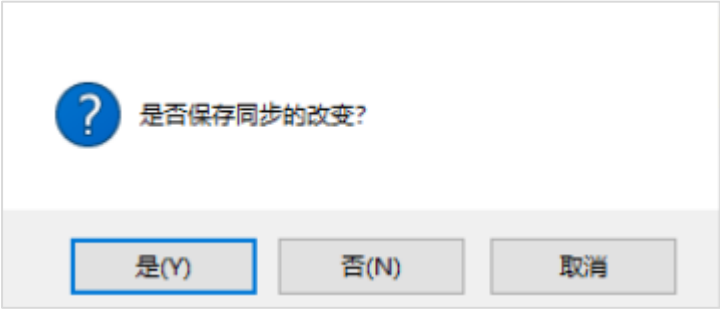
在线工程比较操作具体请参见离线工程比较操作部分内容。

根据不同使用场景执行相应操作。

使用场景	操作步骤
当连接的 PLC 工程有更新且需要比较最新的 PLC 工程时	单击“从控制器上传”，将 PLC 的工程上传至“在线工程比较”右侧
当需要将当前工程下载到 PLC 时(下载规则同多重下载默认参数功能)	单击“传送到控制器”，将当前工程下载至 PLC（通过菜单进入界面时显示功能键，下载流程不显示）

2、保存同步结果。

关闭工程比较视图或关闭工程详细比较视图，打开是否保存同步结果的对话框。单击“是”，保存同步结果。



17. 常见问题及解决方法

17-1. Package

17-1-1. Package 命名规则

XSDH-60A32 3.5.15.40 1.0.0 P1 20211027
① ② ③ ④ ⑤

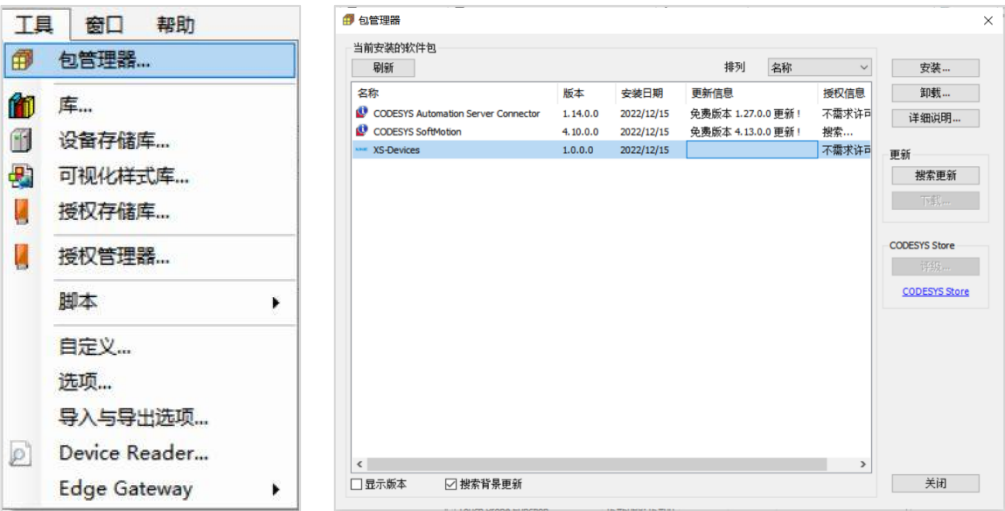
序号	命名	备注
①	XSDH-60A32	PLC 型号
②	3.5.15.40	Runtime 版本
③	1.0.0	Package 投产版本
④	P1	投产后第 1 次在线升级包
⑤	20211027	Package 包升级日期

17-1-2. Package 的获取

请至官网获取或联系技术支持，官网网址：www.xinje.com；技术服务热线：400-885-0136。

17-1-3. Package 的安装

选择“工具”——“包管理器”，在弹出的界面里安装 Package 包，选择“Install”，找到 Package 所在位置进行安。例如：想要安装 XSLH-24A16 的 Package，最好先将之前的 Package 进行卸载后再安装新的。



17-2. XS 系列 PLC 固件升级

17-2-1. 固件命名规则

XSDH-60A32_3.5.15.40_1.0.0_P1_20211027

①②③④⑤

序号	命名	备注
①	XSDH-60A32	PLC 型号
②	3.5.15.40	Runtime 版本
③	1.0.0	固件投产版本
④	P1	投产后第 1 次在线固件升级
⑤	20211027	固件升级日期

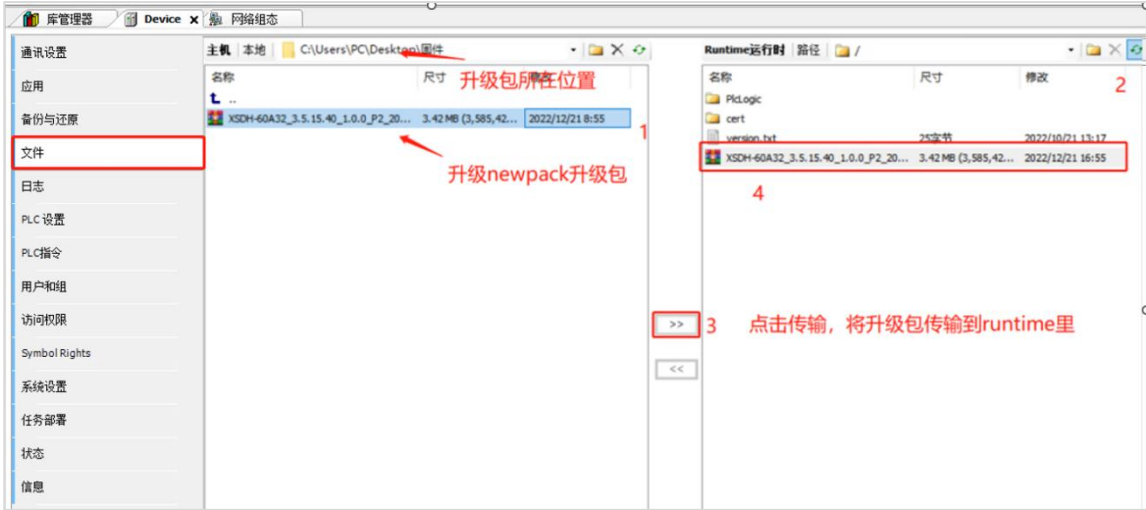
17-2-2. 固件获取

请至官网获取或联系技术支持，官网网址：www.xinje.com；技术服务热线：400-885-0136。

17-2-3. 固件安装及其注意事项

方式一：通过 newpack 包升级固件

创建设备标准工程，连接设备，在主设备目录下选择“文件”选项，点击右上角的刷新后，将 newpack 升级包传输到 runtime 中，等待传输完成，重启设备，设备在升级时 ERR 灯会常亮，更新结束后 ERR 会熄灭，此时就可以扫到设备。



方式二：固件版本 V1. 0. 2a 或 V1. 1. 0 升级至 V2. 2. 0



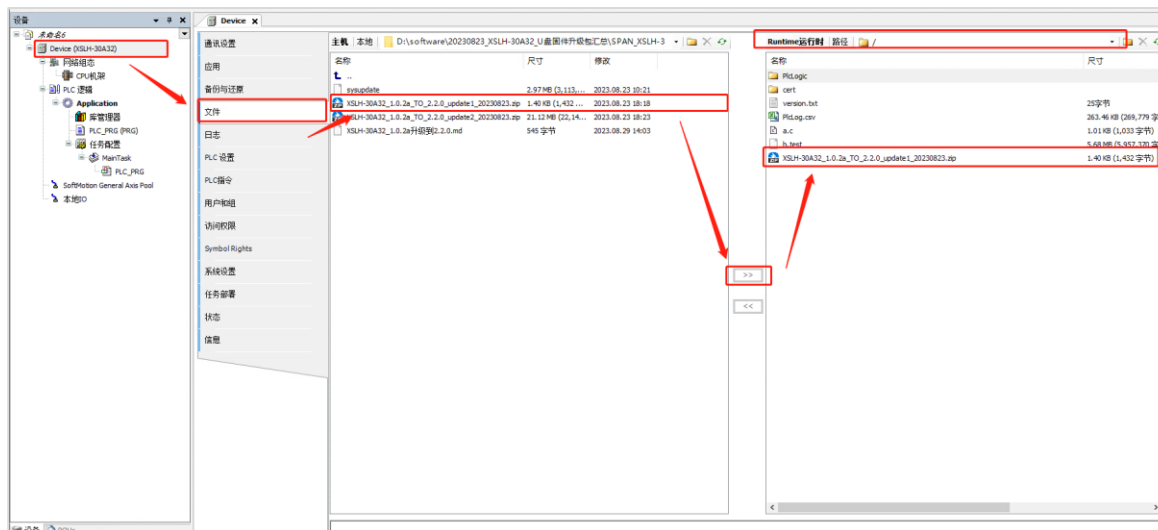
该方式仅适用于 ARM 系列机型。

此处，以升级 XSLH-30A32 机型设备为例，其他类型设备的操作步骤均相同。

1、上位机与 PLC 设备建立连接，如下图：

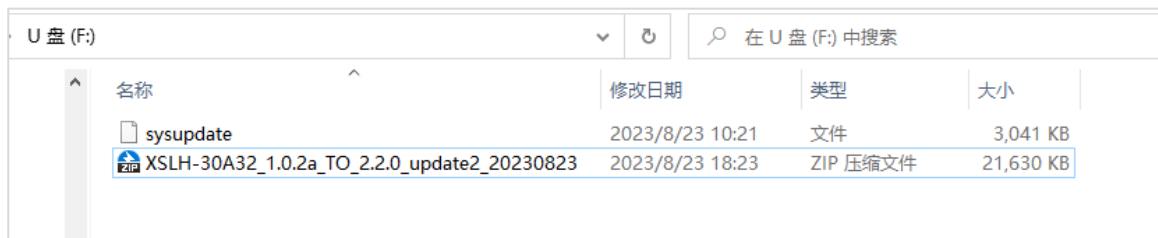


2、在“Device”的“文件”窗口内，选择本地文件“XSLH-30A32_1.0.2a_TO_2.2.0_update1_20230823.zip”传送到 PLC runtime 根目录(/)，如下图：



3、PLC 断电后重启。在升级过程中，ERR 灯闪烁，大约 1~2 秒；升级完成后，RUN 灯亮：

4、把 XSLH-30A32_1.0.2a_TO_2.2.0_update2_20230823.zip 和 sysupdate 复制到 SD 盘根目录；如下图：



5、PLC 断电，插入 SD 卡，上电；上电后 PWR 灯常亮，ERR 灯闪烁后熄灭，此时只有 PWR 灯亮。

6、断电，取出 SD 卡，上电；即可扫描连接。

说明：

- 用户不能擅自修改 PLC 固件升级包名称；
- U 盘或 SD 卡为 FAT32 或者 NTFS 格式；
- PLC 固件升级包只能放置于 U 盘/SD 卡根目录下，不得放置于其他子目录下。只能放置一份升级文件不允许放置多份。否则不执行；
- U 盘/SD 卡固件升级完成前，建议不要拔掉 U 盘/SD 卡，ERR 灯闪烁（至少两秒）代表升级中，此时 ERR 灯灭掉代表升级完成且只有 PWR 灯亮。此时可拔 U 盘/SD 卡。若此时 ERR 灯常亮代表更新失败；
- 固件升级完成之后，原有的程序会被初始化，若想运行该程序，需要用户重新下载；
- 固件升级过程中请勿断电；
- 若升级失败拔出 U 盘/SD 卡，重新上电后按照原来程序运行。

方式三：固件版本 V2.2.0 及以上的升级方式（后续固件升级均用此方式）

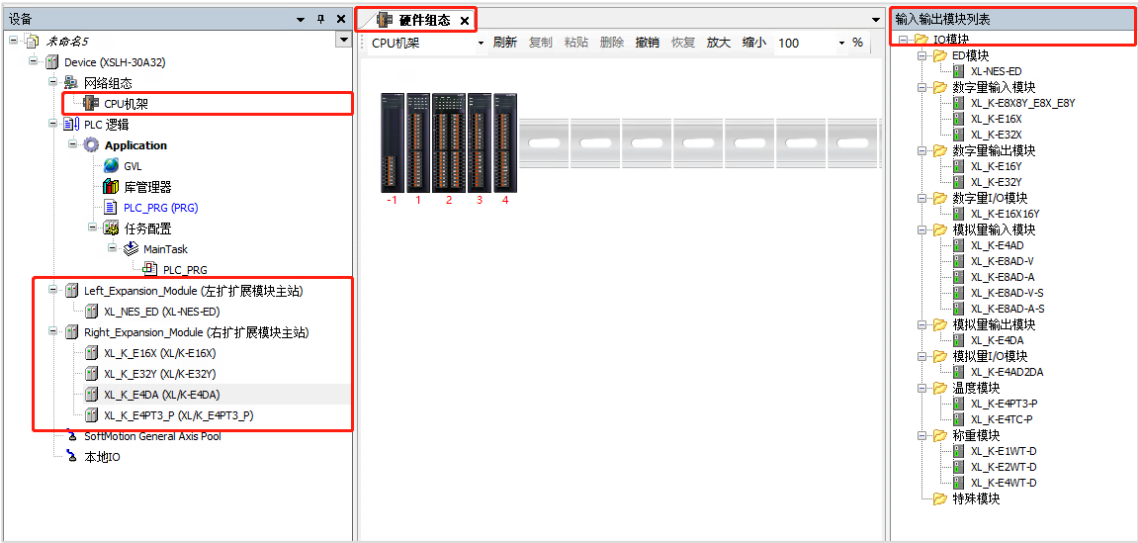


- 下位机固件 V2.2.0 及以上版本支持通过 U 盘或 SD 卡升级固件；
- 目前 U 盘用来升级 X86 工控机型设备，SD 卡用来升级 ARM 机型设备。

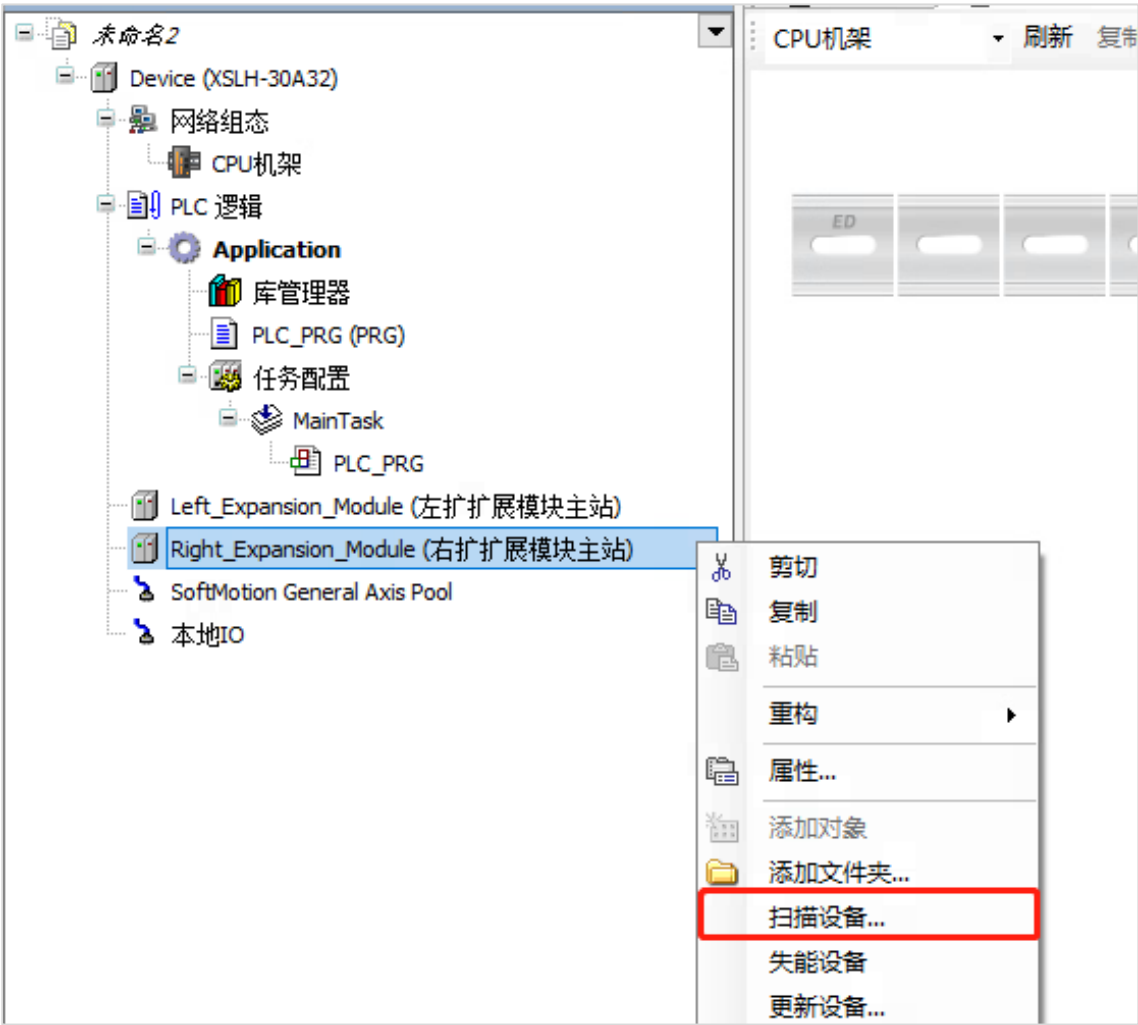
其他刷机方式请咨询技术支持。

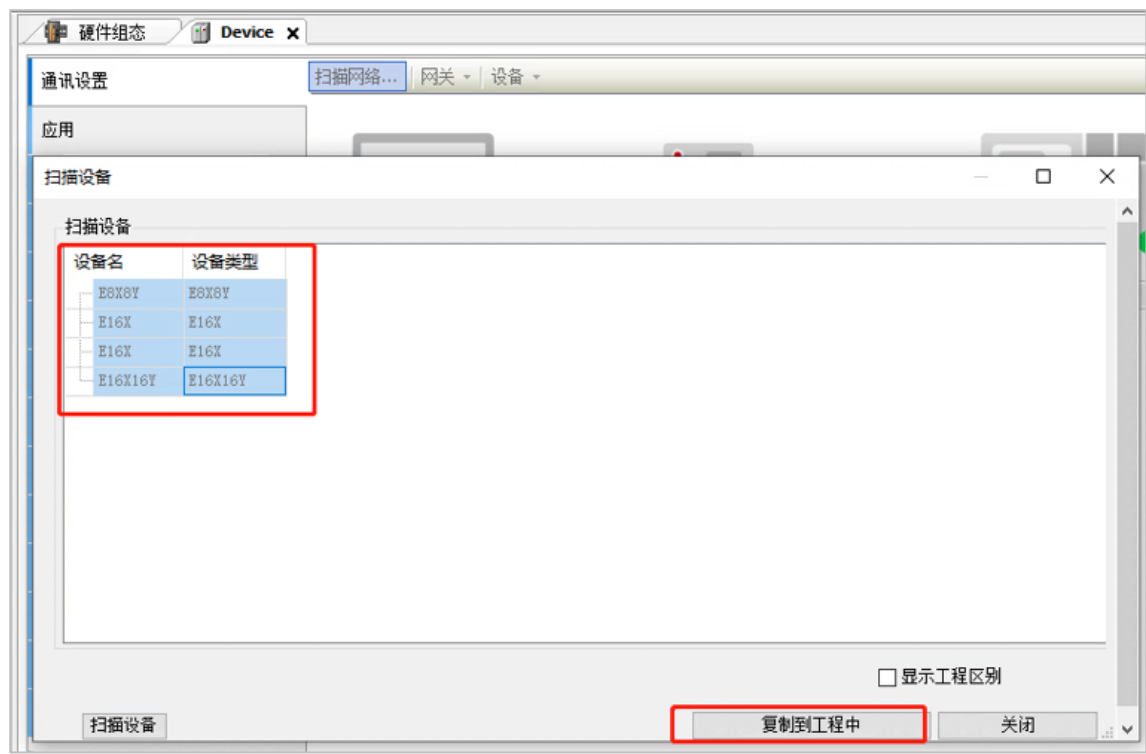
17-3. XS 系列的本地扩展模块

- 1、 双击网络组态节点下的 CPU 机架总线节点，可打开本地硬件组态配置界面与右侧“输入/输出模块列表”界面。可通过“输入/输出模块列表”添加本地 IO 模块。如图所示：

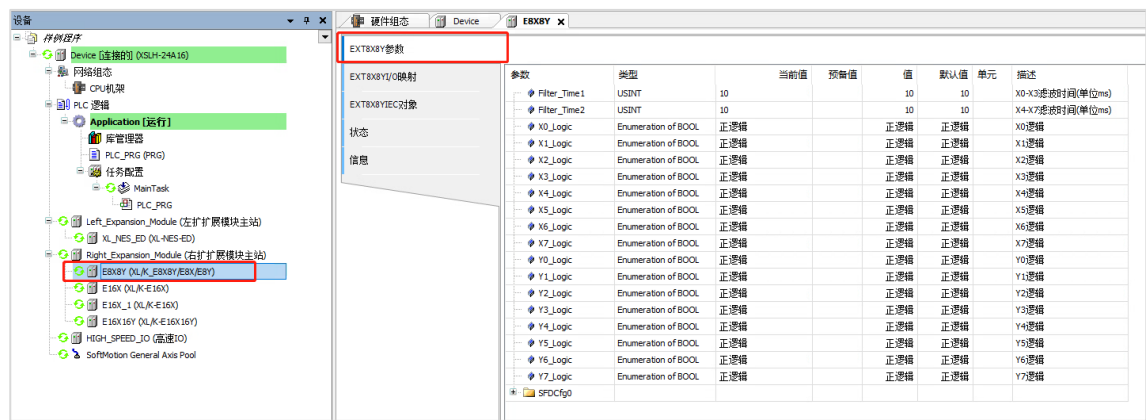


- 2、 可通过点击“Right_Expansion_Module(右扩展模块主站)”右键选择“扫描设备”菜单项来添加扩展模块。



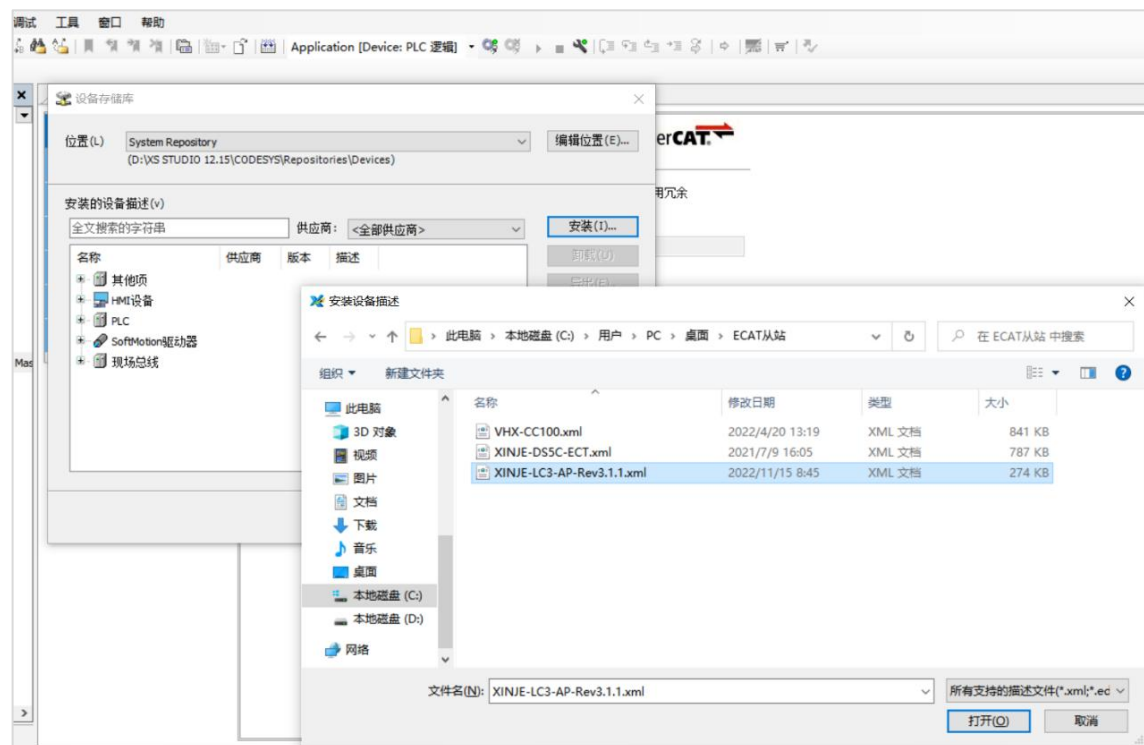
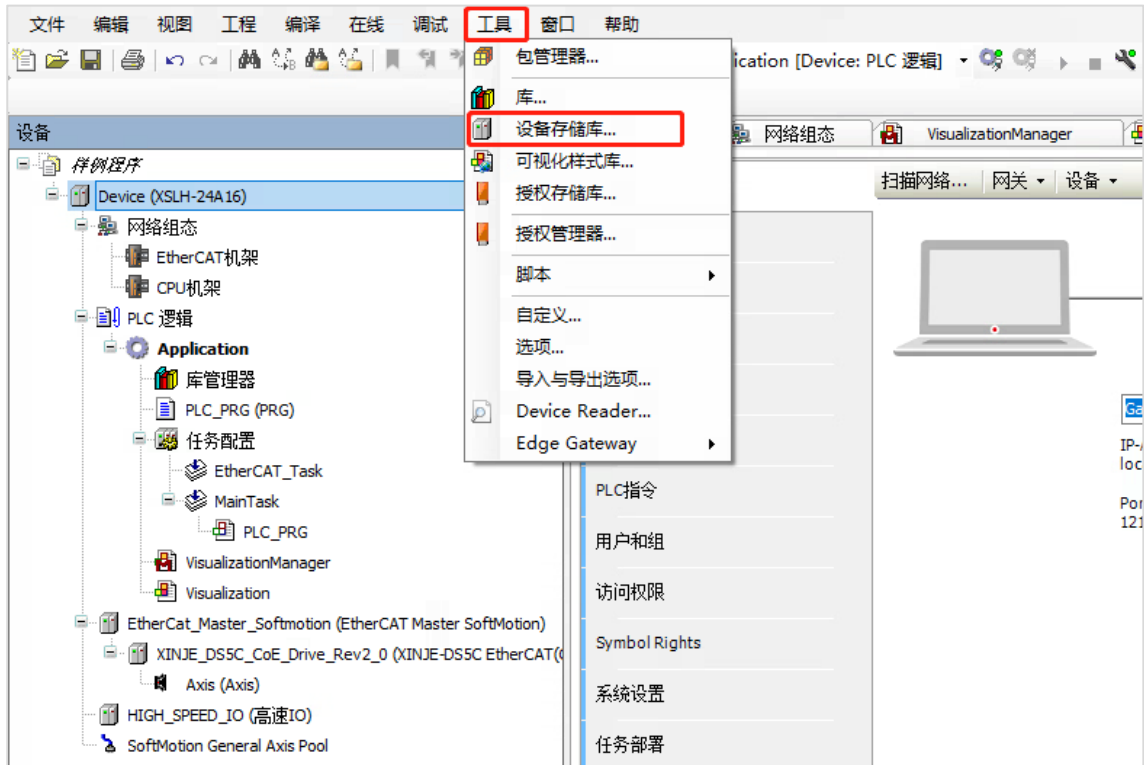


3、扫描添加完成后，连接 PLC 设备，登录后运行。如下图所示：

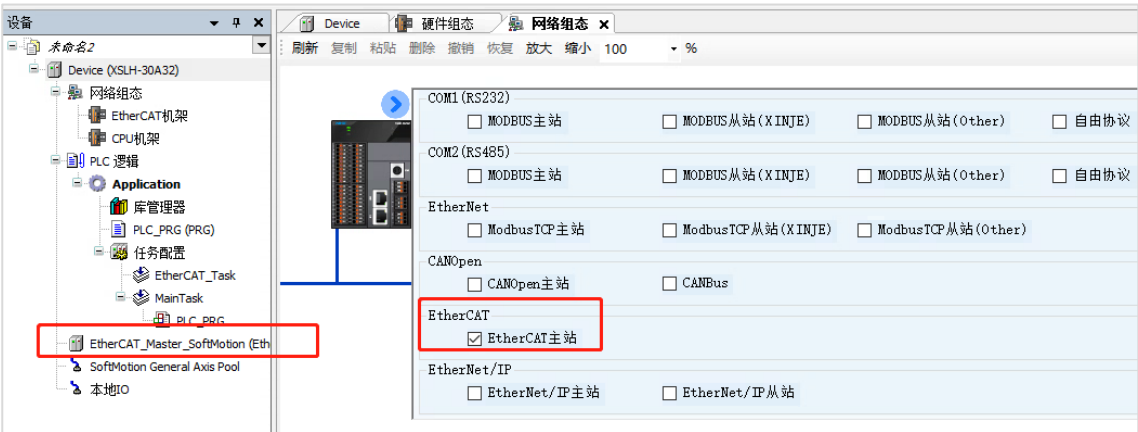


17-4. XS 系列的远程扩展模块

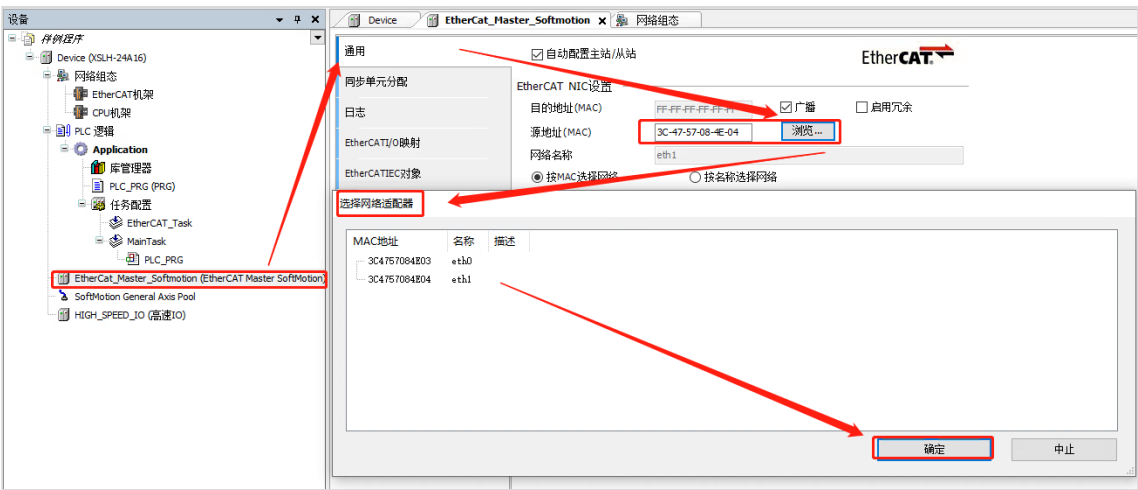
- 1、将 LC3-AP 远程模块接上 24V 电源供电。
- 2、XS Studio 上位机添加 LC3-AP 描述文件。



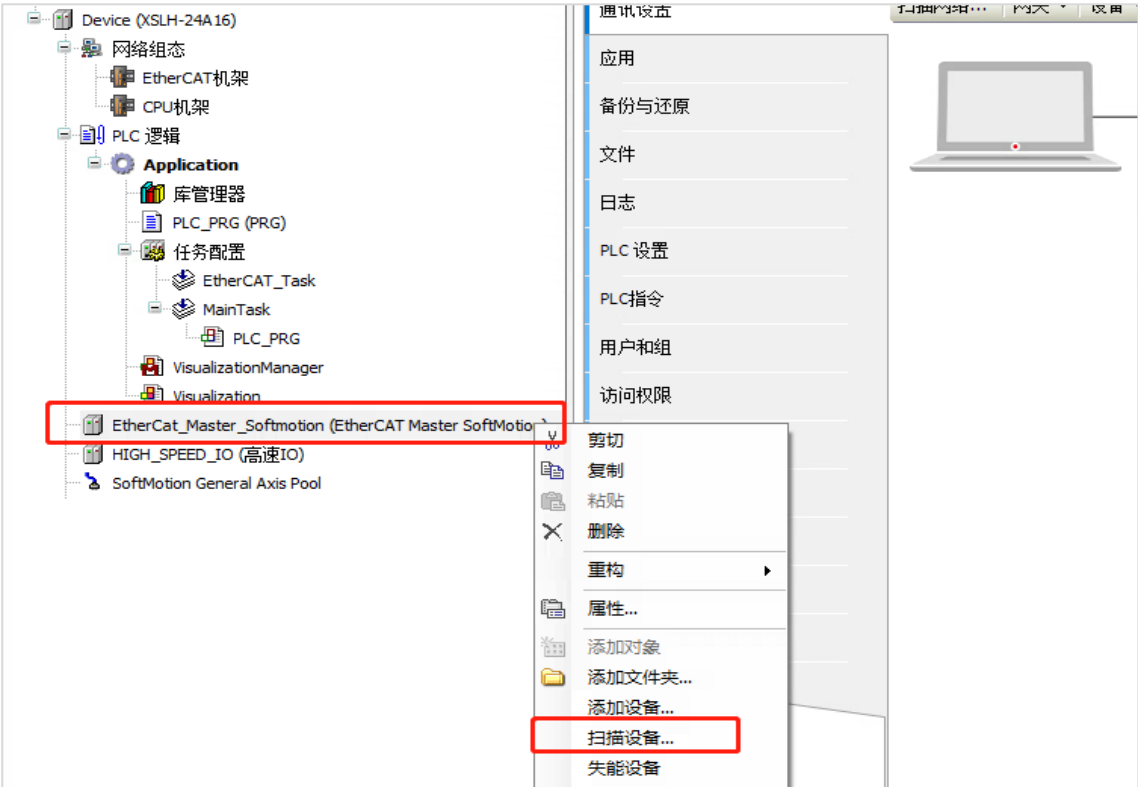
3、添加 EtherCAT 主站。



4、选择用于通讯的网口。



5、扫描添加 LC3-AP 远程模块。



6、复制所有设备到工程中。



17-5. 拨码

- XSDH-60A32-E 支持拨码功能，其具体功能如下所示：
 - ◆ 00：正常启动，无特殊处理，加载用户程序；
 - ◆ 10：初始化 IP；
 - ◆ 01：上电不加载用户程序。
- XSLH 系列 PLC 支持拨码功能，其具体功能如下所示：
 - ◆ 00：正常启动，无特殊处理，加载用户程序；
 - ◆ 10：初始化 IP；
 - ◆ 01：上电不加载用户程序。
- 系统拨码
 - ◆ 拨到 RUN，PLC 切到运行状态；
 - ◆ 拨到 STOP，PLC 切到停止状态；
 - ◆ 拨到 RESET 保持 11s-20s，立刻恢复默认 IP 并释放对应连接（ENET0：192.168.6.6、ENET1：自动获取 IP 地址）。

17-6. 用户新安装 XS Studio 软件，打开后进行编译，出现了很多报错

一般来说是由于缺失库导致的，在工程栏双击打开库管理器-点击下载缺失的库-等待下载缺失库。

17-7. 网关显示红点

可能是网关服务被关闭。可以去任务管理器中打开服务“Codesys Gateway V3”或者重启电脑。

17-8. 添加多个 EtherCAT 从站后，有 warning 提示

是由于伺服站号重复，不会影响使用，若想清除 warning 和双蓝色下划线，重新扫描一下伺服，然后修改重复站号即可。

17-9. EtherCAT 轴一运行通讯就断了

EtherCAT 相关 POU 必须要放在 EtherCAT 任务下，它是有位置同步周期的。

17-10. 用户怎么取消密码登录

分为两步：

- 1、在扫描设备的界面的“设备”里--点击 change communication Policy--在弹出来的界面里的 Device User Management 里的 New Policy，改成 Optional User Management。
- 2、在 Devices 界面里选择“Device”--右击选择“初始复位设备[Device]”。这样操作后，就不需要每次登录都需要密码。

若客户想登录时输入密码，将在扫描设备的界面的“设备”里--点击 change communication Policy--在弹出来的界面里的 Device User Management 里的 New Policy，改成“Enforced User Management”。

17-11. 为什么无法连接到 PLC 设备

无法连接 PLC 一般归纳为以下几点：

- 请确认为 XS 系列产品（出现过很多将 XD、XG 系列产品错当成 XS 系列产品的情况）。
- 在未取消勾选“通过目标和 ID 过滤网络扫描”菜单项的情况下，确认上位机工程设备与目标设备一致，否则也会无法扫描到设备。
- 通过取消勾选“通过目标和 ID 过滤网络扫描”菜单项进行网络扫描来确认双方 IP 是否为同一网段，若扫描到的设备没有绿色标识显示，则为跨网段设备，可通过右侧信息栏查看该设备的 IP 地址；也可通过 ping 指令确认可否 ping 通；若无法确认 IP 地址，可尝试将拨码 1 置 ON 之后重启设备（上电初始化 IP 为 192.168.6.6），再次进行扫描连接；若网段相同但子网掩码不同，会无法扫描到设备，但可直接输入 IP 地址进行连接设备。



- 若 IP 确认无误还是无法连接设备，可能是 PLC 程序死机（程序里有死循环或超出 PLC 的负载能力），此时可将拨码 2 置 ON（上电不加载用户程序），再次扫描连接设备；若能扫描连接，此时下载一个空程序，进行抹除异常的程序之后，再恢复拨码状态，同时检查异常程序（是否有超长循环或任务周期时间过小）。
- 若以上步骤还是无法连接设备，请反馈技术支持。

17-12. IP 地址修改不成功

修改 IP 之后若网段不同，同时需要修改网关。修改成功之后重新上电生效。

17-13. 提示“没有可用于此对象的源码，是否浏览原始库以显示源代码”



- ① 指针非法访问：空指针、指针指向不合法区域（指针指向的地址与操作系统内部地址冲突）；
- ② 数组越界；
- ③ 除数为 0；
- ④ 有符号和无符号变量之间赋值运算；
- ⑤ for、while、repeat 循环条件使用不当。

17-14. setposition 清完位置之后断电上电绝对值伺服位置发生变化

- ① 使用掉电保持程序进行中转；
- ② 信捷伺服固件 3792 版本，使用 MC_Home，35 模式。

17-15. PLC 死机

- ① ARM 机型：拨码 1 置 ON 断电上电，不加载程序。重新下载程序，拨码拨回原位；
- ② X86 机型：D 盘--CODESYS 文件夹--Plclogic--Application 删除。

17-16. 在线下载程序丢失

在线下载勾选如下图：



17-17. 不同电脑在同一局域网有时会连接到其他设备

关闭网络再连接 PLC 或在连接时使用闪烁判断扫描到的设备是否为实际连接设备。

17-18. 添加隐含检查功能

在程序的编写过程中，可能会发生如下几种情况：

- ① 除法运算的被除数在某些情况下会为零；
- ② 指针在赋值的过程中可能不小心指向空地址；
- ③ 调用数组时数组边界溢出了。

针对上述情况，XS Studio 有专门的解决方案，在一个应用下可以添加特殊的 POU，但是这个 POU 程序必须存在应用下，如果隐式检查能够检查函数的阵列和界限，以及除数为零和运行系统中的指针。

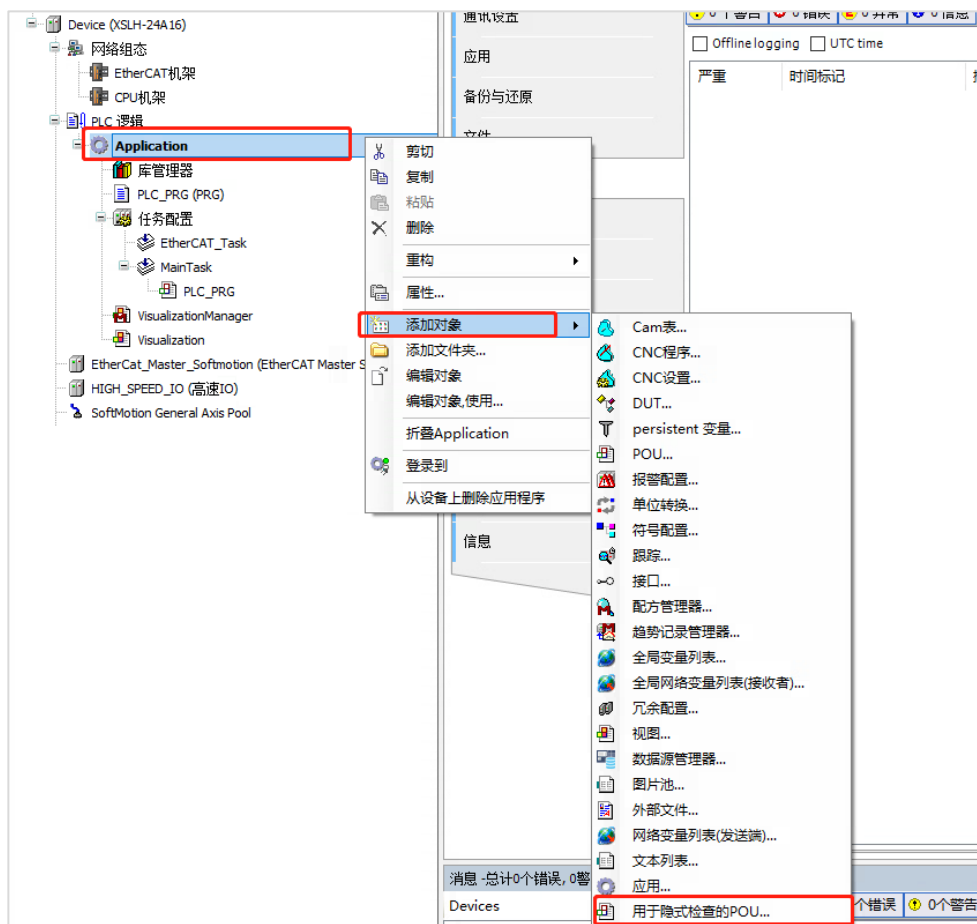
 注意:

如果设备的校验功能是由一个特殊的库提供的，那么这个功能可以被停用。

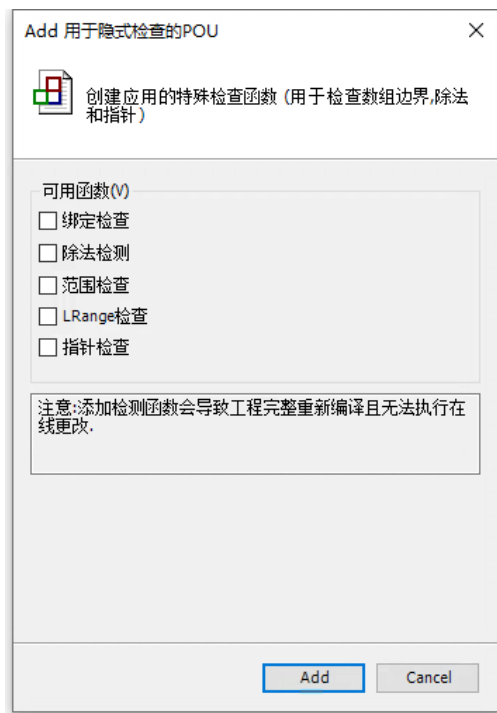
因为这个原因“添加对象”菜单中的 POU 的“隐式检查”属性中提供了以下函数功能：

- CheckBounds
- CheckDivInt
- CheckDivLInt
- CheckDivReal
- CheckDivLreal
- CheckRange
- CheckRangeUnsigned
- CheckPointer

当在 POU 中添加一个检查之后，将会按照选定的编程语言方式打开。默认的编程环境是 ST 语言编辑器。用户可以在应用下鼠标右键选择“添加对象”，选择“用于隐含检查的 POU”，随后，系统则会弹出对话框，如图所示：



如下，会对这几种常用的函数做介绍。



隐含检查的 POU 添加选项

1) CheckBounds

此函数检查是否对数组的边界有侵犯（例如，通过检测到的错误标志设置或更改索引）。一个可变的数组类型被分配这个函数被称作隐藏的函数。

当调用该函数参照以下输入参数：

- 索引：字段元素的索引；
- 下限：字段部分的下限；
- 上限：字段部分的下限。

只要索引的范围内，则返回值是索引本身。否则将被返回对应违反上限或下限范围内的字段。

例如，在程序的数组中“a”超出上限，其程序如下：

```
PROGRAM PLC_PRG
VAR a: ARRAY[0..7] OF BOOL;
b: INT:=10;
END_VAR
```

```
a[b]:=TRUE;
```

在程序之初，数组 a 只有 0~7 八个成员，而在实际的程序中，a 数组的第 b 个成员为 TRUE，而 b 在程序定义时却为 10，实际已经超出了数组 a 的定义范围。

当使用了该 CheckBound 函数后，将影响的分配前的影响，被改变的索引值从“10”到上限“7”。因此值 TRUE 将会分配给数组元素 a[7]。

2) Check+数据类型

为了检查除数的值，避免除数为零，可以使用检查函数 CheckDivInt, CheckDivLint, CheckDivReal 以及 CheckDivLReal。在将它们包含在应用中之后每个相关代码发生的除法过程都将产生一个此函数调用的预处理。

例如，使用除法指令，具体程序如下：

```
PROGRAM PLC_PRG
VAR
```

```

erg:REAL;
v1:REAL:=799;
d:REAL;
END_VAR

```

```

erg:= v1 / d;

```

在上例中，`erg` 等于 `v1` 除以 `d`，而 `d` 在变量定义之初并没有赋予初值，故其初值为 0，在程序中如果直接将数除以 0，系统则会出错。但是，如果在指令中经过了指向除法运算检查的 `CheckDivReal` 函数之后除数“`d`”的值在初始化的时候变为“1”。因此除法最终结果为 799，能够有效的避免控制器异常出错。

3) CheckRange(Un)Signed

为了在运行过程中检查域的限制，可以使用函数 `CheckRangeSigned` 或 `CheckRangeUnsigned`。此检查功能的目的是恰当的子集违规处理（如设置一个检测到的错误旗帜或改变值）。当一个变量的子集类型被确认后这个功能将被隐藏访问。

当访问这个功能时得到以下输入参数：

- 值：域类型被分配的值；
- 低：域的下限；
- 高：域的上限。

如果被分配的值是在有效的域内，他将作为功能中的返回值被使用。否则，对应超过范围的数值，要么上限或下边界的范围将被返回。

例如，分配 `i:=10*y` 将被隐性替代为
`i := CheckRangeSigned(10*y, -4095, 4095);`

如果 `y` 的值为 1000，变量 `i` 将不会分配到原始执行提供的 $10 \times 1000 = 10000$ ，取而代之的是 4095，因为函数设定的上限值最大为 4095。

例如，死循环的例子：

```

VAR
ui : UINT (0..10000);
END_VAR
FOR ui:=0 TO 10000 DO
...
END_FOR

```

FOR 循环将永远不会离开，因为检查功能制止了将 `ui` 超过 10000。



使用 `CheckRangeSigned` 指令和 `CheckRangeUnsigned` 的功能，可能会导致在一个无限循环，例如：如果一个子界类型被用作循环不匹配子范围的增量。

4) CheckPointer

函数 `CheckPointer` 检查地址的指针引用是否都在有效的内存范围内。在运行过程中，用户可能在每个指针操作都可以使用 `CheckPointer` 来进行指针访问的检查。

17-19. 实现掉电保持的注意事项

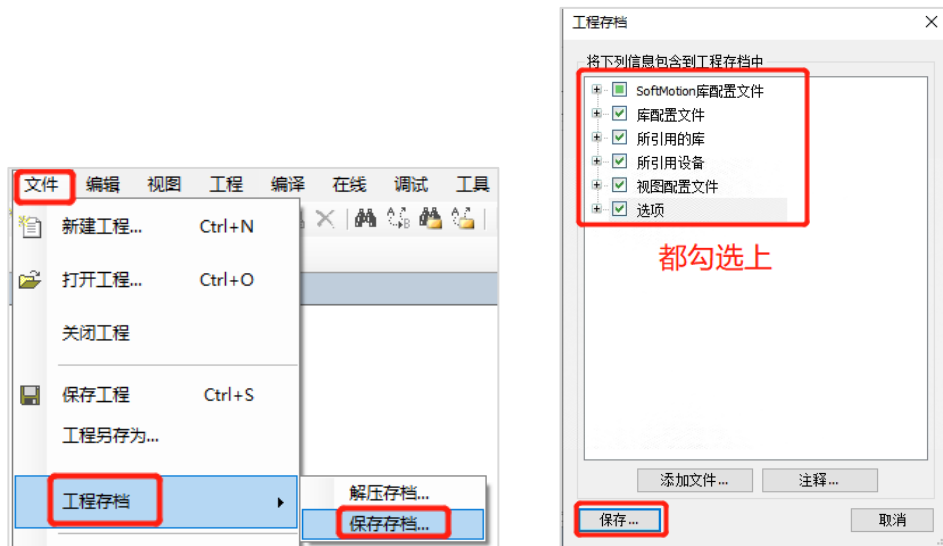
- 掉电保持新增或删减需要将程序登录并下载或在线修改时勾选更新自动启动程序。
- 修改了掉电保持区域，内存分配重新排列了。数据会被清 0。
- X86 机型（XSA330、XSA500 系列）内部有 UPS 可以实现掉电保持，其他 X86 机型（原

M210) 需要确定是否配有 UPS。

- 确定程序中是否有其他地方进行赋值。

17-20. 打开工程报错，工程保存必须以存档的方式保存

project 格式工程是没有包含全部信息的，当打开他人的工程或以不同的版本打开时，会丢失信息，要以打包格式存储，这样才不会丢失信息。



17-21. 设置可以添加行注释和节点注释



手册更新日志

本手册的资料编号记载在手册封面的右下角，关于手册改版的信息汇总如下：

序号	资料编号	章节	更新内容
1	PS06 20230109 1.0	-	1、第一版手册发布
2	PS06 20230504 1.1	-	1、增加设备组态章节； 2、增加信捷 402 轴，替换原本 CIA402 轴； 3、新增特殊功能中的高速 IO 配置； 4、更新手册相应配图。
3	PS06 20230906 1.2	-	1、增加指令工具箱； 2、增加 M 区掉电保持； 3、新增变量表的定义、导入导出等功能； 4、增加 Modbus RTU、串口自由格式等协议； 5、增加 CANBus 自由格式通讯功能； 6、更新手册相应功能及配图。
4	PS06 20240126 1.2.1	-	1、修改 3-4-3、5-2-3、14-19 章节内容。
5	PS06 20240814 1.3	-	1、新增第 8、9、10、11、12、13 章内容。
6	PS06 20250402 1.4	-	1、修改 3-2、3-4、4-3、第 7 章、9-2-4 章节内容； 2、新增 3-8 PROFINET 通信、第 14 章掉电保持功能说明。
7	PS06 20250904 1.5	-	1、增加显示特殊机型说明； 2、新增 ModbusTCP 添加多从站说明； 3、变更第四章内容，新增 EtherNet/IP 通讯显示通讯功能说明； 4、变更 7-3 章节梯形图说明； 5、新增 1-1-1、5-3-4、5-4、8-4-2、8-8、8-9、13-3、16-2 章节内容； 6、修改 15-2-9 节 M 区范围说明
8	PS06 20260227 1.6	-	1、新增 5-3-4-1 EtherCAT 诊断功能适用设备、8-9-1 隐式 POU 检查适用设备、16-2-1 在线工程比较适用设备，表格列出 XS 机型版本及 XS Studio 版本要求。 2、第 13 章 任务分核 章节内容及架构重新调整，增加了相关说明。 3、手册错误修正
9	PS06 20260819 1.7	-	1、新增 “2-2-4-5.SD 卡/U 盘下载程序”； 2、新增 XSA-W 内部网口通讯时的 IP 设置注意点； 3、手册错误修正。



微信扫一扫，关注我们



无锡信捷电气股份有限公司

WUXI XINJE ELECTRIC CO., LTD.

地址：江苏省无锡市滨湖区建筑西路 816 号

总机：0510-85134136

传真：0510-85111290

网址：www.xinje.com

邮箱：xinje@xinje.com

全国技术服务热线：400-885-0136



扫码咨询智能客服