

# E300 系列可编程控制器

## 用户手册【硬件篇】



宁波海天驱动技术有限公司

发布版本：V1.10

发布日期：2024-08-12

## 权利声明

---

### 版权声明

宁波海天驱动有限公司版权所有，并保留对本手册及本声明的最终解释权和修改权。未得到宁波海天驱动有限公司的书面许可，任何单位及个人不得以任何方式或形式对本手册内的任何部分进行复制、摘录、备份、修改、传播、汇编、翻译成其它语言、将其全部或部分用于商业用途。

### 商标声明

 为宁波海天驱动有限公司或其关联公司的注册商标，受法律保护，侵权必究。未经宁波海天驱动有限公司或其关联公司的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、永久下载、修改、传播、抄录或与其他产品捆绑销售，或注册为其域名或无线网址名称，或域名或无线网址的主要部分。

### 免责声明

本手册依据现有信息制作，内容如有更改，恕不另行通知。宁波海天驱动有限公司在编写该手册时，已尽最大努力保证其内容准确可靠，但不对本手册中的遗漏、不准确或印刷错误导致的损失和损害承担责任。

## 前言

感谢您购买宁波海天驱动有限公司的 E300 系列 PLC 产品！

在使用我公司 E300 系列 PLC 产品前，请您仔细阅读本手册，以便更清楚地掌握产品的特性，更安全地应用，充分利用本产品丰富的功能。

### 安全注意事项

负责产品安装、操作的人员必须经过严格培训，遵守相关行业的安全规范，严格遵守该手册提供的相关设备注意事项和特殊安全指示，按正确的操作方法进行设备的各项操作。

为防止人身伤害或设备损坏，对务必遵守的事项特做一下说明。错误使用本产品而可能带来的危害和损害程度，见相关符号说明。

	<b>警告</b> 表示由于没有按要求操作造成的危险，可能导致人身伤亡。
	<b>注意</b> 表示由于没有按要求操作造成的危险，可能会导致人身轻度或中度伤害和设备损坏。
	<b>提示</b> 表示对操作的描述进行必要的补充或说明。

## 版本履历

---

将手册版本升级所更新的内容罗列出来并插入超链接，可从此处直接跳转到特定修改处，便于查看修改内容。

发行日期	版本	修订内容
2023 年 6 月	V1.00	• 首版发布
2024 年 6 月	V1.10	• 修复了一些问题，相关组件修改

# 目 录

权利声明 .....	II
前 言 .....	III
版本履历 .....	IV
目 录 .....	V
<b>1 产品概述 .....</b>	<b>- 1 -</b>
1.1 PLC 介绍 .....	- 2 -
1.1.1 PLC 外观 .....	- 2 -
1.1.2 PLC 规格参数 .....	- 2 -
1.1.3 CPU 输入功能 .....	- 5 -
1.1.4 PLC 接口定义 .....	- 7 -
1.1.5 系统架构 .....	- 9 -
1.1.6 数据存储器 .....	- 9 -
1.2 PLC 安装 .....	- 10 -
1.2.1 安装注意事项 .....	- 10 -
1.2.2 安装尺寸 .....	- 11 -
1.2.3 接地和布线 .....	- 11 -
1.2.4 安装方法 .....	- 11 -
1.2.5 抑制电路 .....	- 13 -
<b>2 扩展模块规格 .....</b>	<b>- 15 -</b>
2.1 电源模块 .....	- 16 -
2.2 数字量模块 .....	- 18 -
2.2.1 数字量输入模块 .....	- 18 -
2.2.2 数字量输出模块 .....	- 20 -
<b>3 简易工程的组态步骤 .....</b>	<b>- 22 -</b>
3.1 CODESYS 与 PLC 硬件连接 .....	- 23 -
3.2 新建工程 .....	- 23 -
3.3 安装设备描述文件和库到 CODESYS 中 .....	- 27 -
3.3.1 安装设备描述文件 .....	- 27 -
3.3.2 安装库 .....	- 28 -
3.4 设置通信 .....	- 30 -
3.5 配置任务 .....	- 32 -
3.6 编写程序 .....	- 32 -
3.7 编译和下载 .....	- 34 -
3.8 监控和调试 .....	- 36 -
<b>4 模块的使用和状态描述 .....</b>	<b>- 38 -</b>
4.1 PLC 本机高速 IO 的使用说明 .....	- 39 -
4.1.1 本机 IO 作为普通输入使用 .....	- 39 -

4.1.2	本机 IO 作为高速输入使用 .....	- 40 -
4.1.3	ExtBus 库指令说明 .....	- 44 -
4.2	数字量模块的使用 .....	- 49 -
4.3	模拟量模块的使用 .....	- 56 -
<b>5</b>	<b>附录 .....</b>	<b>- 59 -</b>
5.1	修改 PLC 的 IP 及网关 .....	- 60 -
5.2	读取 PLC 版本 .....	- 62 -
5.3	U 盘固件更新 .....	- 62 -
5.4	掉电数据保存 .....	- 62 -
5.5	写入 PLC 时间 .....	- 64 -
5.6	采样跟踪的使用(Trace) .....	- 65 -
5.7	源程序上下载 .....	- 70 -
5.8	同一网络下存在多个 PLC 时如何更改 PLC 名称 .....	- 74 -
5.9	如何将文件写入 U 盘 .....	- 76 -
5.10	添加库文件到工程中 .....	- 83 -
5.11	复位功能 .....	- 84 -

# 产品概述

1

1.1

PLC 介绍

1.2

PLC 安装

# 1.1 PLC 介绍

**E357:** CODESYS-SP19 版本编程环境，PLC 本机自带 10 路 DI，6 路高速计数器，EtherNET×1，EtherCAT×1，CANopen/RS485×1，USB×1，32MB 程序数据空间，支持可视化 WebVisu、支持 EtherNet/IP 主站、支持 Modbus (RTU/TCP) 主从官方授权，支持双独立 IP，支持 OPC UA 从站，支持 Softmotion 和 CNC。

## 1.1.1 PLC 外观

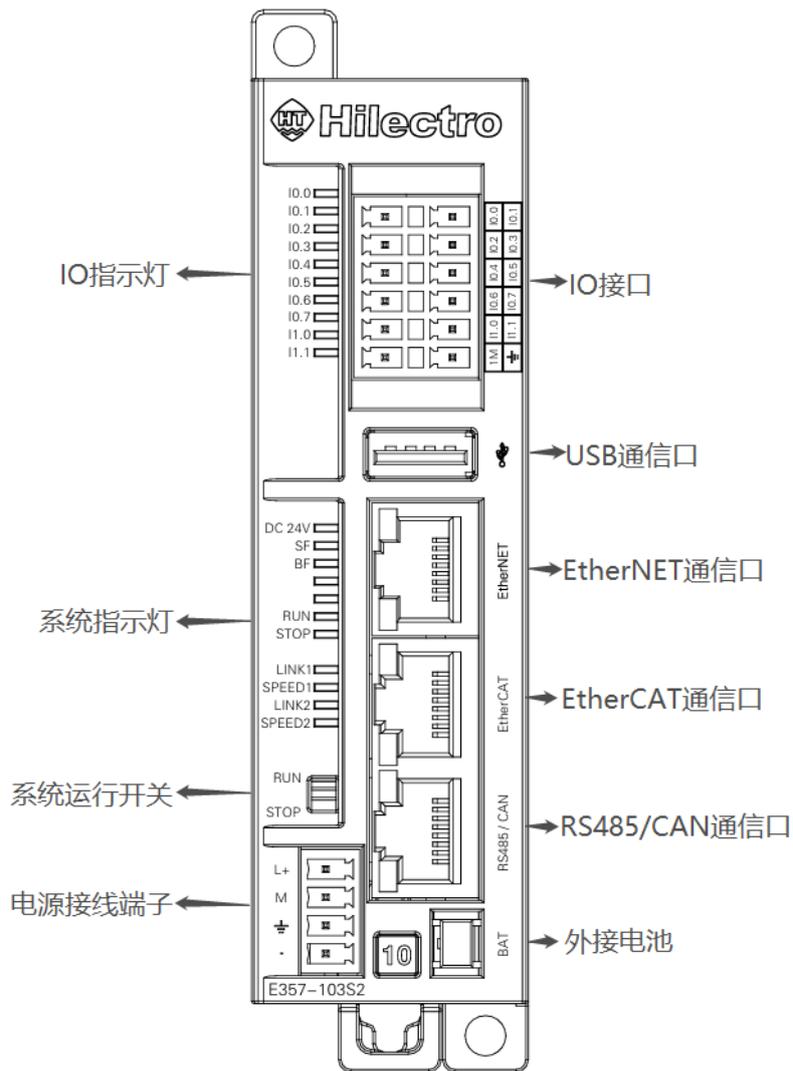


图 1-1 E357 外观接口示意图

## 1.1.2 PLC 规格参数

表1-1 常规规格

尺寸 (W×H×D)	34×115×101.6 mm
功耗	19.2W
<b>内存</b>	
用户程序空间	32MB

掉电保持		
掉电保持空间	总共 <b>64KB</b> ，分为： 内存 1（ <b>32KB</b> ）： <b>GVL</b> 全局变量里面的 <b>RETAIN</b> （保留型变量） 内存 2（ <b>32KB</b> ）： <b>PersistentVars</b> （持续型变量）	
系统时钟	掉电保持时间约 112 小时（典型值）每月偏差 < 60s，E357 可以外接电池，外接电池后掉电保持时间为 2 年以上（典型值）。	
电源特性		
额定输入电压	24V DC	
输入电压范围	20.4V~28.8V DC	
输入电流	0.8A	
极性反接保护	有	
总线电源电压	+5V DC	
总线电源电流	1.6A	
隔离	外部电源与系统电源之间隔离	
LED 指示灯特性		
24V 电源指示灯	绿色	亮起：24VDC 供电正常，熄灭：无 24VDC 供电
SF 指示灯	红色	亮起：系统故障，熄灭：无错
BF 指示灯	红色	亮起：总线故障，熄灭：无错
RUN 指示灯	绿色	亮起：系统运行，熄灭：系统停止
STOP 指示灯	橙色	亮起：系统停止，熄灭：系统运行
RJ45 网口指示灯	绿色	闪烁：连接，熄灭：未连接
	黄色	亮起： <b>100Mbps</b> ，熄灭：10Mbps
CAN 口指示灯	绿色	数据接收指示灯
	黄色	数据发送指示灯
网口通信指示灯		
Link1 指示灯	绿色	亮起：连接；熄灭：未连接
SPEED1 指示灯	黄色	亮起：100Mbps；熄灭：10Mbps
Link2 指示灯	绿色	亮起：连接；熄灭：未连接
SPEED2 指示灯	黄色	亮起：100Mbps；熄灭：10Mbps
I0.0~I1.1	绿色	亮起：有信号输入；熄灭：无信号输入
保护功能		
电源保护	供电电源端提供反接保护功能及浪涌吸收功能	
接口保护	通讯端口防雷保护	

扩展 I/O 能力	
1个 CPU 支持 INT-00机架数	4
每机架支持最多模块数	主机架：11个（电源模块，CPU，中继模块，8个信号模块）， 从机架：10个（电源模块，中继模块，8个信号模块）
1个 CPU 可支持的 EtherCAT 从站数	最多 128 个
每个 EtherCAT 从站支持的模块数	最多支持 8 个模块
隔离	
电源隔离	外部电源与系统电源之间隔离
通讯隔离	以太网、RS485 隔离、CAN 隔离
编程软件	
编程软件包	CODESYS V3.5 SP19 Path5
编程语言	符合 IEC61131-3 的编程语言：CFC, FBD, LD, IL, ST, SFC
位指令执行速度	0.015us
运动控制器功能	
支持的运动控制器功能	SoftMotion 和 CNC

表1-2 通讯口规格

EtherNET 通讯口	
通讯接口	1个以太网口
协议类型	支持一路 EtherNet/IP 主站；支持一路 Modbus TCP 通信，可以同时做 Modbus TCP 主站和从站；socket；可视化 WebVisu。
每段最大电缆长度	100m
一个站点最大连接数	EtherNet/IP 主站最多支持64个从站；Modbus TCP 做主站时，最多支持32个从站，Modbus TCP 做从站时，最多支持32个主站；socket 最大连接数1000。
用户数据	EtherNet/IP 主站单个连接最多支持1400个字节数据读写，最小循环通讯周期（RPI）为5ms。 Modbus TCP 主站单个连接最多支持240个字节数据读写，最小循环通讯周期（RPI）为1ms。 socket 单帧数据512个字节，可多帧发送，最小循环通讯周期根据程序扫描规划。
波特率	10/100Mbps 自适应
隔离	通信口隔离
EtherCAT 通讯口	
通讯接口	1个 EtherCAT 接口
最大站点数	每个主站最多支持128个从站
协议类型	EtherCAT 接口协议
支持功能	支持分布时钟配置、启动参数配置、配置 PDO 参数和映射、配置总线循环周期。
每段最大电缆长度	100m

波特率	10/100Mbps 自适应						
隔离	通信口隔离						
<b>RS485通讯口</b>							
通讯接口	1个 RS485通讯口						
最大站点数	每个主站最多支持32个 Modbus RTU 从站						
协议类型	Modbus 主站/从站协议						
Modbus 波特率	1200、4800、9600、19200、38400、57600、115200bps						
隔离	通信口隔离						
<b>CANopen 通讯口</b>							
通讯接口	1 个 CAN 通讯主站接口						
最大从站点数	1 个主站后面最多可连接 32 个从站						
协议类型	CANopen DS301 标准协议						
支持功能	支持自动启动 CANopen Manager						
	支持可选从站轮询						
	支持启动从站						
	支持 NMT						
	支持同步生产						
	支持同步消耗						
	支持心跳产生						
	支持激活时间创建						
传输速率 (Kbit/s)	1000	800	500	250	125	50	20
最大长度 (m)	25	50	100	250	500	1000	2500
隔离	通信口隔离						
<b>USB 接口</b>	USB 主设备接口，更新固件，下载程序						

### 1.1.3 CPU 输入功能

E357 本机集成 10 路数字量输入，其输入特性如下表所示。

表1-3 E357 数字量输入特性

数字量输入特性	
本机集成 IO 点数	10
输入类型	漏型/源型
额定电压	24V DC
输入电压范围	20.4~28.8V DC

浪涌电压	35V DC, 持续0.5s	
逻辑1信号 (最小)	15 VDC, 2.5mA	
逻辑0信号 (最大)	5 VDC, 1mA	
连接2线接近开关传感器 (BERO)	1mA (允许的最大漏电流)	
输入滤波	可配置, 支持0.2us, 0.4us, 0.8us, 1.6us, 3.2us, 6.4us, 12.8us、0.2ms, 0.4ms, 0.8ms, 1.6ms, 3.2ms, 6.4ms, 12.8ms, 默认为6.4ms	
隔离(现场与逻辑)	500V AC, 1分钟	
隔离组	见接线图	
同时接通的输入	10	
最大电缆长度	500米(标准输入)	
屏蔽	50米(高速计数器输入)	
非屏蔽	300米(标准输入)	
<b>高速计数器输入特性</b>		
脉冲捕捉输入	10	
高数计数器	总计	6
	单相	6×500kHz
	两相	4×250kHz

表1-4 E357 本机输入作为高速计数器时, 高速计数器输入点及计数模式如下

模式	计数器	描述		
	HSC0	I0.0	I0.1	I0.2
	HSC1	I0.3	I0.4	I0.5
	HSC2	I0.6	I0.7	--
	HSC3	I1.0	I1.1	--
	HSC4	I0.2	--	--
	HSC5	I0.5	--	--
0	带有内部方向控制的单相计数器	时钟	--	--
1		时钟	--	复位
2		不支持该计数模式		
3	带有外部方向控制的单相计数器	时钟	方向	--
4		时钟	方向	复位
5		不支持该计数模式		
6	带有增减计数时钟的两相计数器	增时钟	减时钟	--
7		增时钟	减时钟	复位
8		不支持该计数模式		
9	A/B 相正交计数器	时钟 A	时钟 B	--
10		时钟 A	时钟 B	复位

11	不支持该计数模式
----	----------

E357 本机 IO 的具体使用说明请参考章节 4.1 [PLC 本机高速 IO 的使用说明](#)。

### 1.1.4 PLC 接口定义

E357 本机 IO 输入接线如下：

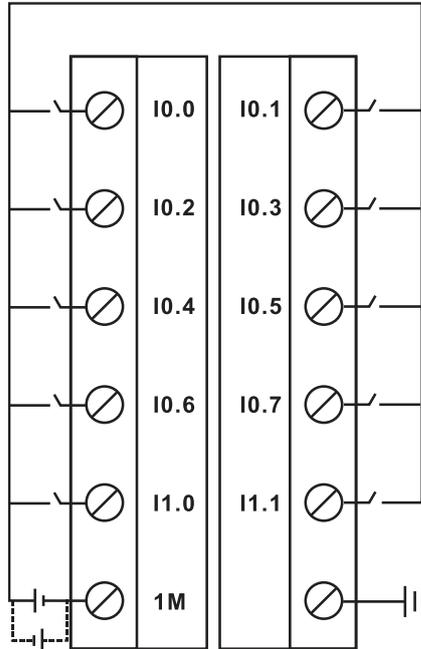


表1-5 电源接口（X4）信号定义

4 位可拆卸端子 (X4)	符号	信号定义
L+  M  ⚡  • 	L+	24V 电源正
	M	24V 电源负
	⚡	大地
	--	--

表1-6 EtherNET/EtherCAT 通信接口信号定义

RJ45 网口	位号	信号	信号定义
	1	TX+	数据发送正端
	2	TX-	数据发送负端
	3	RX+	数据接收正端
	4	TERM	--
	5	TERM	--
	6	RX-	数据接收负端
	7	TERM	--
	8	TERM	--
	连接器外壳	PE	机壳接地
	绿色指示灯	作为 EtherNET/EtherCAT 连接状态指示	
黄色指示灯	作为 EtherNET/EtherCAT 速度状态指示		

表1-7 RS485/CAN 接口定义（E357 的 RS485 与 CAN 接口为同一个接口）

RJ45 网口	位号	信号	信号定义
	1	CAN_H	数据发送正端
	2	CAN_L	数据发送负端
	3	--	--
	4	A0	RS485 信号 A
	5	B0	RS485 信号 B
	6	--	--
	7	CAN_GND	CAN/RS485 地
	8	--	--
	连接器外壳	PE	机壳接地

表1-8 系统运行开关拨码定义

系统运行开关	信号	拨码方向	信号定义
RUN	RUN	向上	系统运行
STOP	STOP	向下	系统停止

表1-9 USB 接口定义

USB 接口	位号	信号	信号定义
	1	V_BUS	+5V 电源
	2	Data-	数据-
	3	Data+	数据+

	4	GND	地
--	---	-----	---

### 1.1.5 系统架构

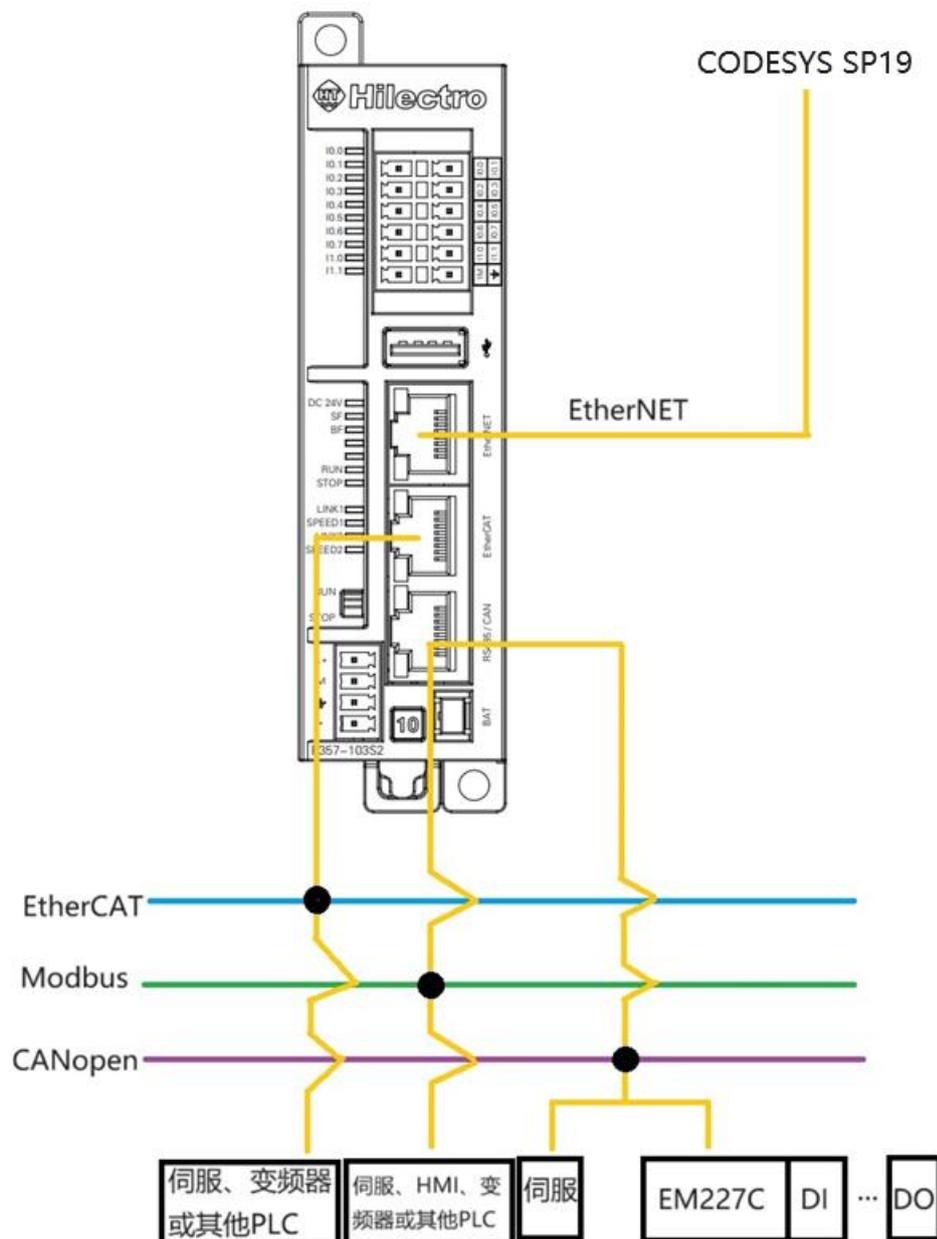


图 1-3 E357 网络架构图

### 1.1.6 数据存储

表1-10 内存区域

区域	描述	大小	寻址方式示例
I	输入区（通过输入驱动的物理输入）	32KB	%X3.5 and %Q3.5: 输入区的地址 3, 第 5 位
Q	输出区（通过输出驱动的物理输出）	32KB	%QB2: 输出区的地址 2, 1 个字节

M	内存区	512KB	%MD30: 内存区的地址 30, 双字
---	-----	-------	----------------------

语法: %<范围前缀><长度前缀><数字|.数字|.数字...>

## 1.2 PLC 安装

PLC 可通过安装孔或 DIN 夹子进行固定。

### 1.2.1 安装注意事项

PLC 的安装须注意以下事项:

#### □ 将 PLC 与加热装置、高电压和电子噪声隔离开

在安装设备器件时,总是把产生高电压和高电子噪声的设备与 PLC 这样的低压电子型的设备分隔开。

在控制柜的背板上安排 PLC 时,应考虑把电子器件安排在控制柜中温度较低的区域。电子器件长期在高温环境下工作会缩短其无故障时间。

要考虑控制柜的背板布线,尽量避免把交流供电线、高能量、开关频率很高的直流信号线与低压信号线、通讯电缆设计在同一个线槽中。

#### □ 为散热和接线留出适当的空间

PLC 的设计采用自然对流散热方式,在模块的上下方都必须留有至少 80mm 的空间,以便于正常的散热。



#### 注意

在垂直安装时,其允许的最高环境温度要比水平安装时低 10℃,而且 CPU 应安装在所有扩展模块的下方。

在安装 PLC 时,应留出足够空间用于接线和连接通讯电缆。

下图显示的是安装在多个机架上的 PLC,其中显示了各机架与相邻组件、电缆槽、机柜之间的间距。通过电缆槽为模块接线时,屏蔽连接元件底部与电缆槽间的最小间距为 40mm。

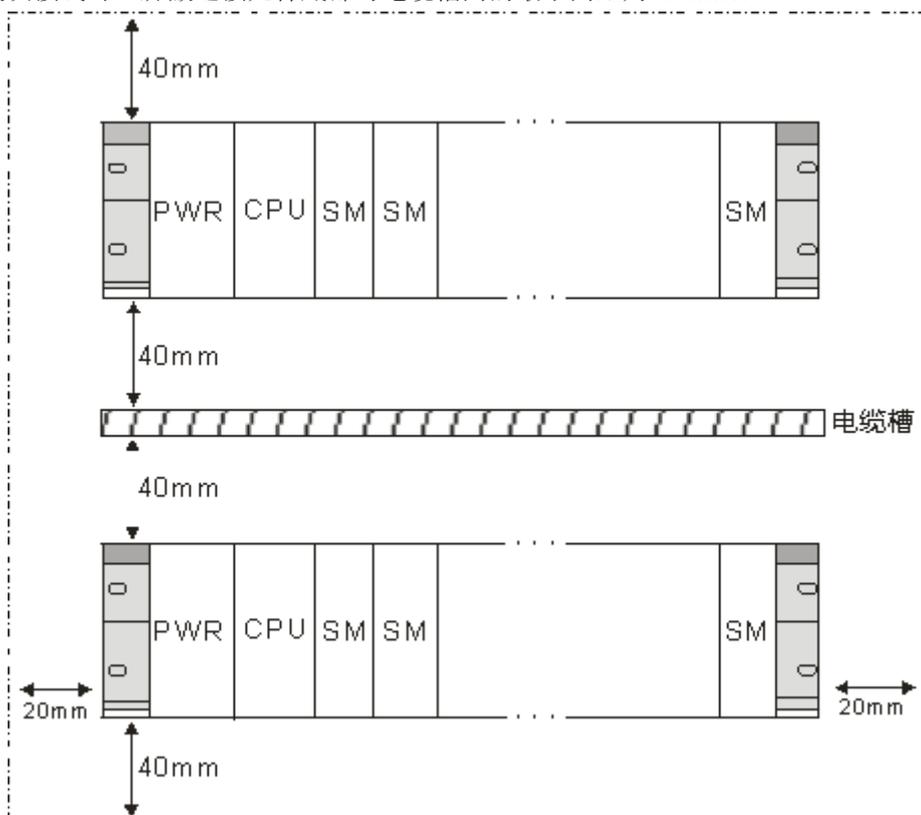


图 1-2 安装示意图

### □ 电源预算

内部总线电压为 5VDC，电流由 CPU（无中继模块时）或中继模块提供。每个机架上扩展模块总线电流消耗之和不能超过 CPU 或中继模块允许的最大总线电流。

## 1.2.2 安装尺寸

CPU 及其扩展模块都有安装孔，便于安装在背板上，安装尺寸见下表。

表1-11 模块安装尺寸表

型号	尺寸图（单位：mm）
E357	

## 1.2.3 接地和布线

### PLC 接地和接线指南

合理的接地和接线对于所有的电器设备是至关重要的，它能够确保您的系统具备最优的操作特性，同时能够为您的系统提供更好的电子噪声保护。

在接地和接线之前，必须先确保设备的电源已被切断，也要保证与该设备相关的设备电源已被切断。

在对 PLC 及其相关设备接线时，必须确保遵从所有有效的电气编码规则。安装和操作所有设备要符合所有有效的国家或地区标准。同地区的权威保持联系，以确定哪些标准符合您的特殊需要。



#### 警告

试图在带电情况下进行接地或接线，有可能造成死亡或严重的人身伤害和设备损坏。

在设计 PLC 系统的接地和接线时必须考虑安全因素，否则有可能造成设备的误动作。因此，您应该执行所有的安全规定以避免人员伤害和设备损坏。



#### 警告

控制设备有可能造成它所控制设备的误操作。这种误操作有可能导致死亡或者严重的人身伤害和设备损坏。因此系统中必须具备独立于 PLC 的急停功能、机电互锁或者其它冗余的安全设施。

## 1.2.4 安装方法

### □ 先决条件

在安装和拆卸 PLC 及其相关设备时，必须预先采取适当的安全措施并且确认其供电被切断。

**警告**

试图在带电情况下安装或拆卸 PLC 及其相关设备有可能导致电击或者设备误动作，在安装和拆卸 CPU 及其相关设备时，如果未能切断所有电源，有可能造成死亡或严重的人身伤害和设备损坏。

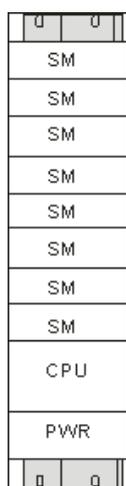
在更换或安装 PLC 时，要确定使用了正确或等同的模块，除了要使用相同的模块外，还要确保安装的方向和位置是正确的。

**注意**

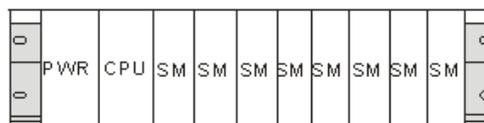
- 如果安装了不正确的模块，PLC 的程序可能会产生错误的功能。
- 如果未能使用相同的模块按照相同的方向和顺序替换 PLC，有可能导致死亡或者严重的人身伤害和设备损坏。

**□ 安装方式**

PLC 既可以安装在控制柜背板上，也可以安装在标准 DIN 导轨上；既可以水平安装，也可以垂直安装，安装时始终将 CPU 和电源模块安装在左侧或底部。



垂直安装



水平安装

**□ 安装与拆除操作**

请按照以下方法安装或拆除 PLC。

**● 安装面板**

- 1) 按照尺寸要求定位打孔；
- 2) 用合适的螺钉将模块固定在背板上；
- 3) 如果使用了扩展模块，将扩展模块的扁平电缆连到前盖下面的扩展口。

**● DIN 导轨安装**

- 1) 将导轨固定在背板上，保持间距 80mm；
- 2) 打开模块底部的 DIN 夹子，将模块背部卡在 DIN 导轨上；
- 3) 如果使用了扩展模块，将扩展模块的扁平电缆连到前盖下面的扩展口；
- 4) 旋转模块贴近 DIN 导轨，合上 DIN 夹子；
- 5) 仔细检查模块上 DIN 夹子与 DIN 导轨是否紧密固定好；
- 6) 为避免模块损坏，不要直接按压模块正面，而要按压安装孔的部分。

**注意**

当 PLC 在振动比较大的使用环境或者采用垂直安装方式时，应该使用 DIN 导轨挡块。如果系统处于高振动环境中，使用背板安装方式可以得到较高的振动保护等级。

**● 拆卸 CPU 或者扩展模块**

- 1) 拆除 PLC 的电源；
- 2) 拆除模块上的所有连线和电缆；
- 3) 如果有其它扩展模块连接在您所拆卸的模块上，请打开上盖，拔掉相邻模块的扩展扁平电缆；
- 4) 拆掉安装螺钉或者打开 DIN 夹子；
- 5) 拆下模块，拆卸端子排。

● 扩展模块的端子排拆卸

- 1) 打开端子排安装位置的上盖；
- 2) 用螺丝刀将安装螺钉逆时针旋转；
- 3) 拆卸端子排。

● 扩展模块端子排的安装

- 1) 打开端子排安装位置的上盖；
- 2) 确保模块上的插针与端子排边缘的小孔对正；
- 3) 将端子排向下压入模块，螺丝刀顺时针将螺丝钉紧。

### 1.2.5 抑制电路

在使用感性负载时，要加入抑制电路来限制输出关断时电压的升高。抑制电路可以保护输出点不至于因为高感抗开关电流而过早的损坏。另外，抑制电路还可以限制感性负载开关时产生的电子噪声。



**注意**

抑制电路的有效性取决于应用，您应该调整其参数以适应您的特殊应用。要确保所有器件参数与实际应用相符合。

□ 晶体管输出和控制直流负载的继电器输出

晶体管输出有内部保护，可以适应多种应用。由于继电器型输出既可以连接直流负载，又可以连接交流负载，因而没有内部保护。

图 1-5 给出了直流负载抑制电路的一个实例。在大多数的应用中，用附加的二极管 A 即可，但如果您的应用中要求更快的关断速度，则推荐您加上齐纳二极管 B。确保齐纳二极管能够满足输出电路的电流要求。

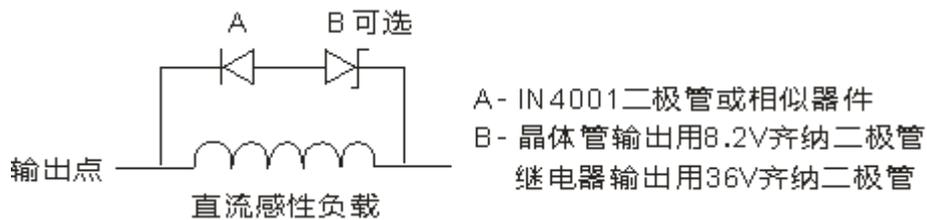


图 1-3 直流负载的抑制电路

□ 交流输出和控制交流负载的继电器输出

交流输出对大部分应用都有内部保护，因为继电器可用于 DC 或 AC 负载所以不提供内部保护。

图 1-6 给出了交流负载抑制电路的一个实例。在大多数的应用中，附加的金属氧化物可变电阻(MOV)可以限制峰值电压，从而保护 PLC 内部电路。要确保 MOV 的工作电压比正常的线电压至少高出 20%。

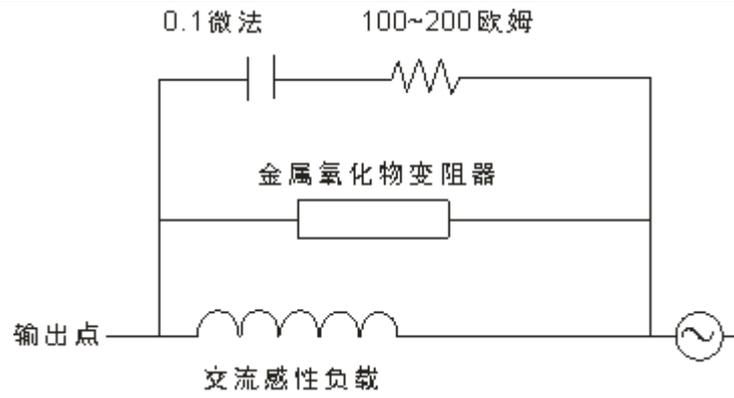


图 1-4 交流负载的抑制电路

# 扩展模块规格

2

2.1

电源模块

2.2

数字量模块

表2-1 扩展模块信息表

订货号	规格描述
<b>电源模块</b>	
EM300-PWR	PWR-02 电源模块, 输入 85~264V AC, 输出 24V DC/2A
<b>数字量模块</b>	
EM300-DI16TN	DIT-16 数字量模块, 数字量输入, 16DI x 24VDC
EM300-DO16TN	DQT-16 数字量模块, 数字量输出, 16DQ x 24VDC

## 2.1 电源模块

EM300-PWR 电源模块为 CPU 及扩展模块（除数字量模块外）提供 24V DC 电源。每个机架请选配一个电源模块，数字量输入输出电源和传感器电源请选择其他供电电源。

表2-2 电源模块 EM300-PWR 基本规格

规格描述	订货号
输入：85~264V AC, 输出：24V DC/2A	EM300-PWR

表2-3 电源模块技术参数

<b>物理特性</b>	
尺寸 (W×H×D)	34×115×101.6 mm
<b>LED 指示灯特性</b>	
24V 电源指示灯 (绿色)	亮起：有 24V DC 输出, 熄灭：无 24V DC 输出
<b>开关特性</b>	
开关	控制 24V DC 电源输出 ON：有 24V DC 输出, OFF：无 24V DC 输出
<b>输入电压特性</b>	
电压范围	85~264V AC, 宽电压输入
额定频率 频率范围	50Hz/60Hz 47Hz~63Hz
效率	75%
交流电流	0.9A/110V、0.5A/220V
浪涌电流 (25℃最大)	≤20A/110V、≤35A/220V
泄露电流	≤5mA/220V AC
<b>输出电压特性</b>	
直流电压/额定电流	24V DC/2A

额定功率	48W
纹波和噪声（最大）	150mVp-p
电压输出范围	±5%
启动/上升/保持时间	≤2.5s/≤50ms/≥20ms
隔离（电源输入与输出）	110V/220V AC 与 24V DC 之间隔离
<b>保护功能</b>	
过载保护	105%~130%的额定输出功率，切断输出，故障排除自动恢复
过压保护	115%~135%U <sub>e</sub> ；保护方式：打嗝模式，故障排除自动恢复
浪涌保护	供电电源端提供浪涌吸收功能
过流保护	电源输出端提供过流保护
<b>安全电磁兼容</b>	
耐电压	输入~输出：1.5kVDC，输入-PE：1.5kV DC，输出-PE：500V DC
隔离电阻	输入~输出，输入-PE，输出-PE：100MΩ/500V DC
依据标准	安全参照UL60950和UL1950，电磁兼容参照EN55022

### 电源模块展示

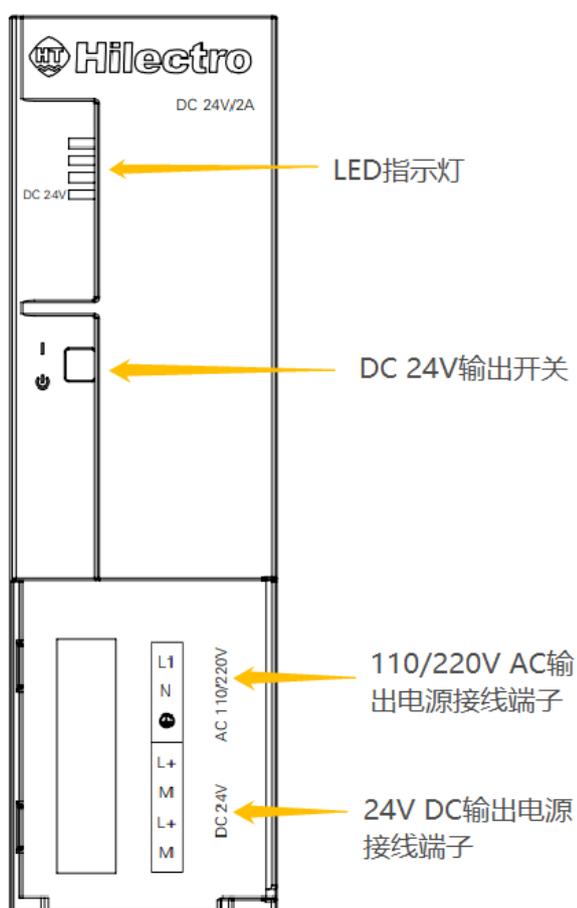


表2-4 220V AC 输入电源接口定义

3 位可拆卸端子	信号	信号定义
	L	火线
	N	零线
		大地

表2-5 24V DC 输出电源接口定义

4 位可拆卸端子	信号	信号定义
	L+	24V 电源正
	M	24V 电源负
	L+	24V 电源正
	M	24V 电源负

表2-6 拨码开关定义

两态开关	位号	拨码方向	信号定义
	ON	向上	有 24V DC 输出
	OFF	向下	无 24V DC 输出

## 2.2 数字量模块

数字量扩展模块用于扩展数字量输入输出。

表2-7 数字量模块基本规格

规格描述	CODESYS 中对应的名称
数字量输入 16 x 24VDC	EM300-DI16T

### 2.2.1 数字量输入模块

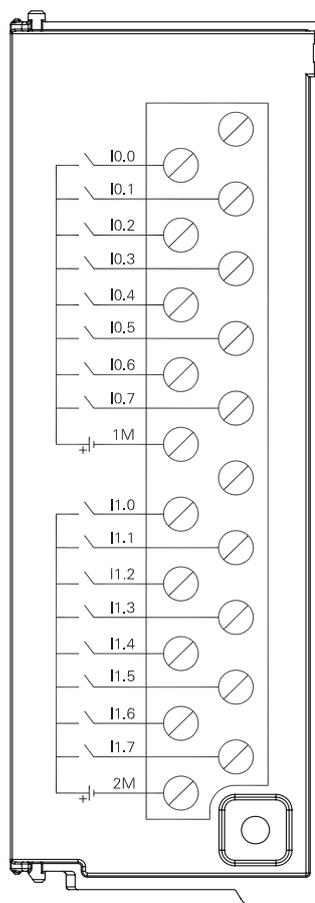
表2-8 数字量输入模块技术参数

特性	EM300-DI16T	
尺寸 (W×H×D)	34×115×100 mm	
输入点数	16	
电流消耗	24V DC	4mA/通道
	+5V 总线	80mA
输入类型	漏型/源型 (IEC 1 类漏型)	
输入额定电压	24V DC, 6mA	

输入电压范围	20.4~28.8V DC		
浪涌电压	35V DC, 持续 0.5s		
逻辑 1 (最小)	15V DC, 2.5mA, 翻转电平: 10.5V DC±15%		
逻辑 0 (最大)	5V DC, 1mA		
连接 2 线接近开关传感器 (BERO) 允许的漏电流 (最大)	1mA		
输入滤波	可配置, 支持 0.2ms、0.4ms、0.8ms、1.6ms、3.2ms、6.4ms (默认)、12.8ms		
输入频率	1.5kHz, 占空比 50%		
输入阻抗	6.6kΩ		
隔离	500V AC, 持续 1min		
每组隔离点数	8	8	8
同时 ON 点数	8	16	32
电缆长度	屏蔽	500m	
	非屏蔽	300m	

### 接线规格

EM300-DI16T 接线图



### 数字量输入模块通道配置

每组（8 通道）配置参数格式为：

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
保留	通道 4-7 滤波时间			保留	通道 0-3 滤波时间		

滤波时间：

0: 0.20ms; 1: 0.40ms; 2: 0.80ms; 3: 1.60ms; 5: 3.20ms; 6: 6.40ms（默认值）；7: 12.8ms。

### 2.2.2 数字量输出模块

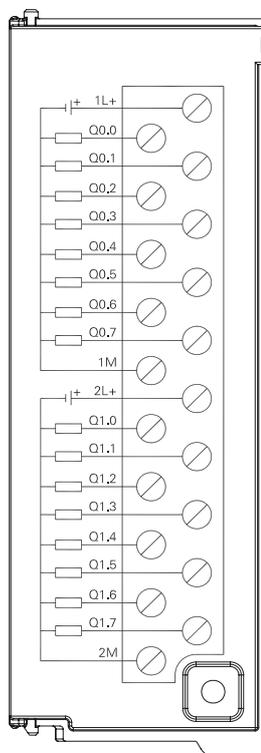
表2-9 数字量输出模块技术参数

特性	EM300-D016TN	
尺寸 (W×H×D)	34×115×100 mm	
输出点数	16	
电流消耗	24V DC	95mA
	+5V 总线	120mA
输出类型	固态-MOSFET, 源型	
输出额定电压	24V DC	
输出电压范围	20.4~28.8V DC	
逻辑 1 (最小)	20V DC	
逻辑 0 (最大)	0.1V DC, 10kΩ 负载	
输出电流 (最大)	0.5A	
每个公共端电流	最大 4A	
输出漏电流 (最大)	15uA	
浪涌电流	8A, 100ms	
灯负载	5W	
接触电阻	0.3Ω, 最大 0.6Ω	
输出延迟	断开到接通 (最大): 50us 接通到断开 (最大): 200us	
输出频率 (最大)	1kHz	
机械寿命 (无负载)	-	
触点寿命 (额定负载)	-	
隔离 现场到逻辑 线圈到触点	500V AC, 持续 1min -	
每组隔离点数	8	
同时 ON 点数	16	
两个输出并联	支持同组内两个输出并联	

电缆长度	屏蔽	500m
	非屏蔽	150m

接线规格

EM300-DO16TN 接线图



数字量输出模块通道配置

每组（8 通道）配置参数格式为：

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
保留							

# 简易工程的组态步骤

## 3

3.1 CODESYS 与 PLC 硬件连接

3.2 新建工程

3.3 安装设备描述文件和库到 CODESYS 中

3.4 设置通信

3.5 配置任务

3.6 编写程序

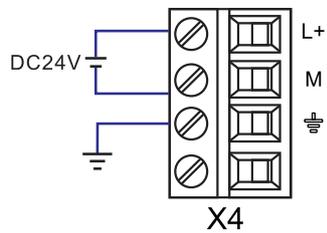
3.7 编译和下载

3.8 监控和调试

### 3.1 CODESYS 与 PLC 硬件连接

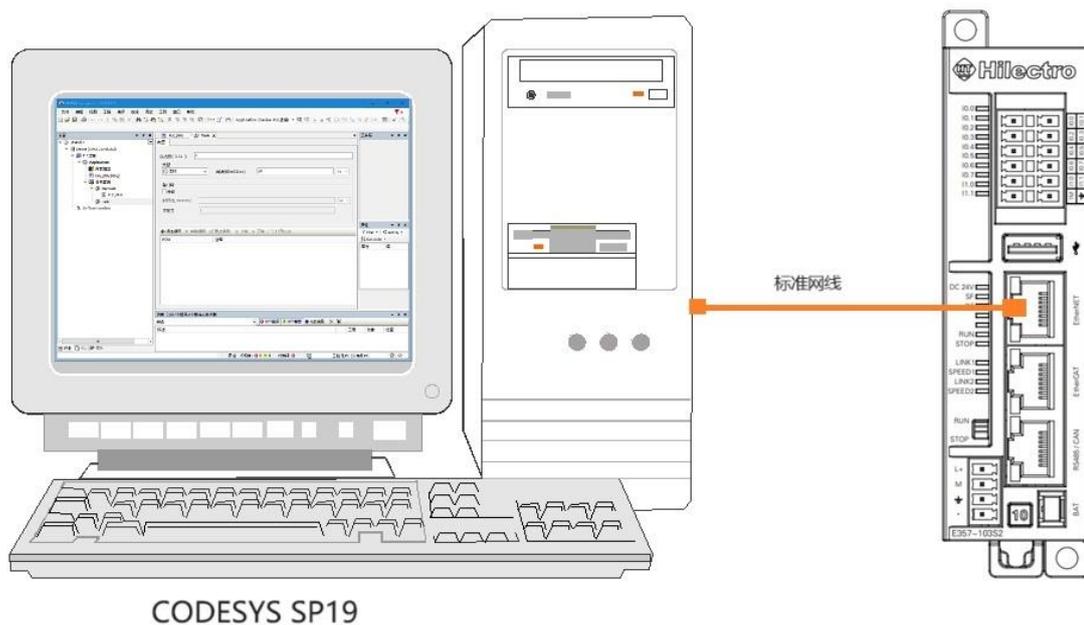
#### 给 E357 供电

下图为 E357 的直流供电接线示意图：



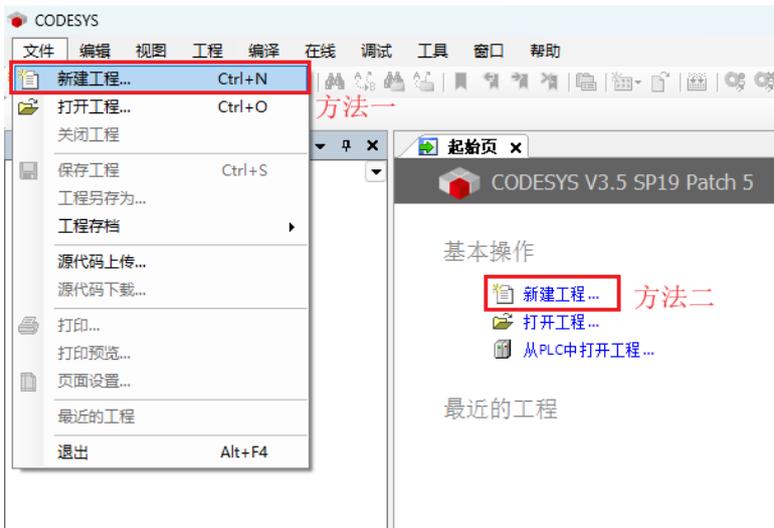
#### 通过以太网口与 E357 相连

使用标准网线连接编程设备 PG/PC 与 E357（EtherNET 通信口）

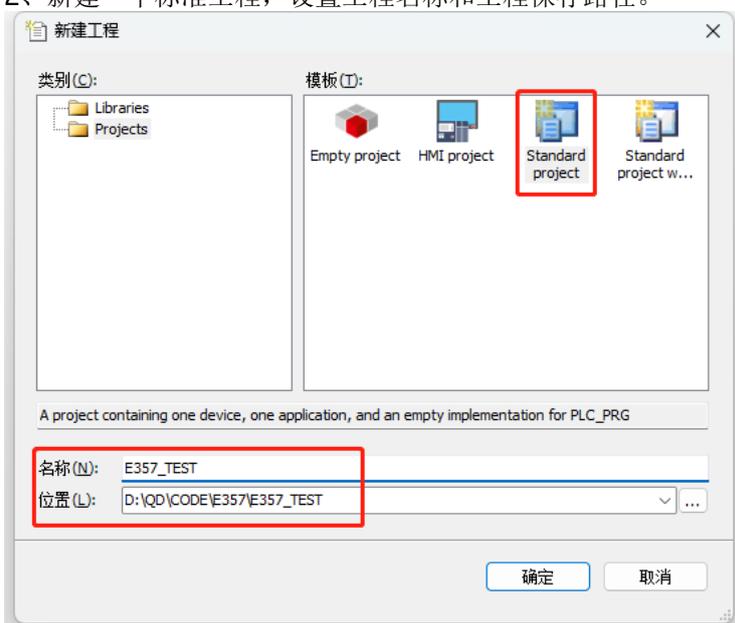


### 3.2 新建工程

1、启动 CODESYS SP19，起始界面如下，点击新建工程。

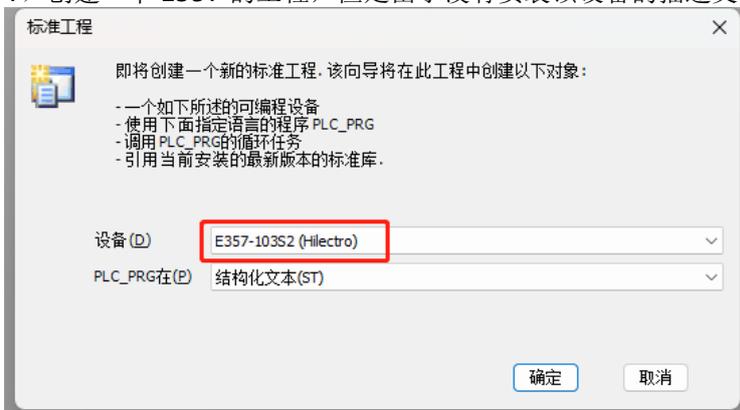


2、新建一个标准工程，设置工程名称和工程保存路径。



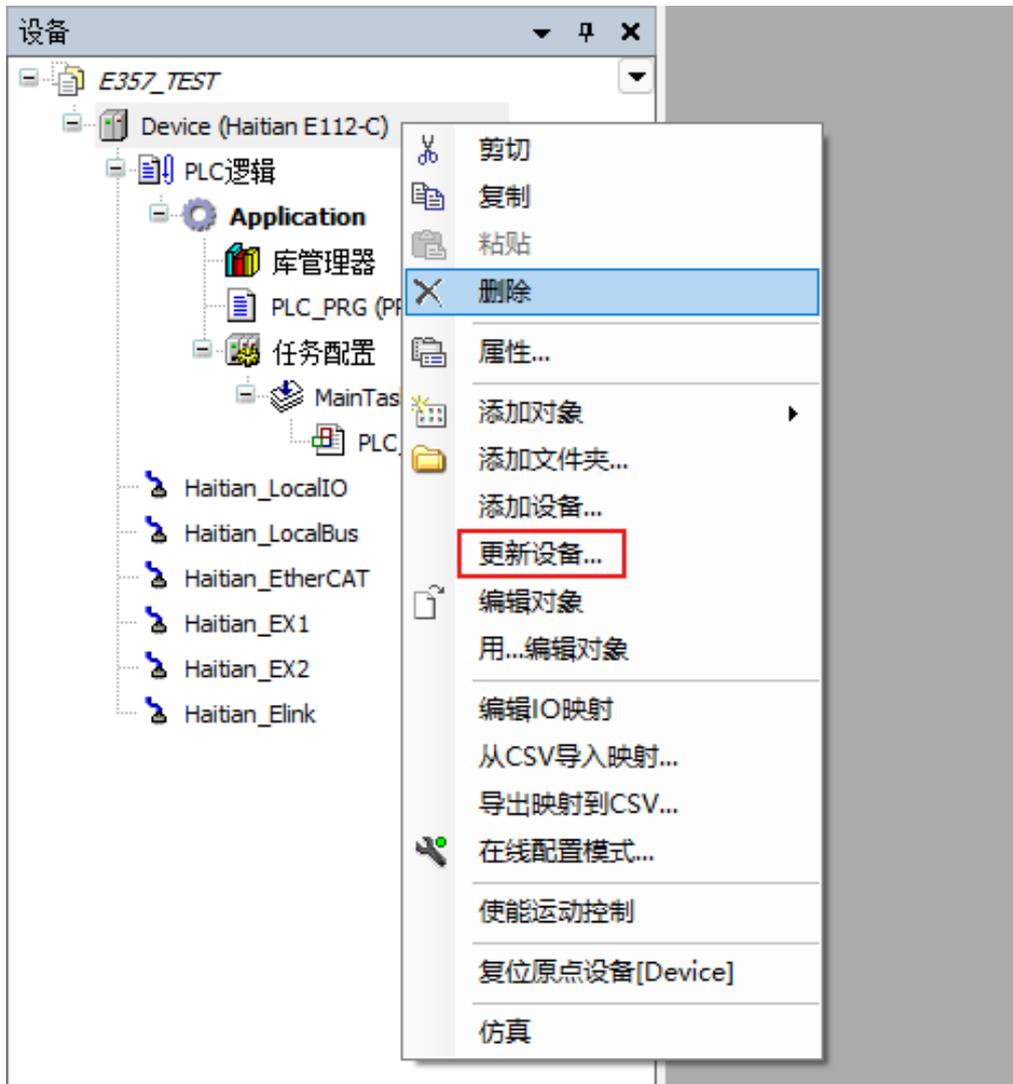
3、选择设备以及编程语言，若没有所需设备，可以选择默认设备创建工程，然后在工程中安装所需设备的描述文件，安装完成后直接更新工程设备即可。

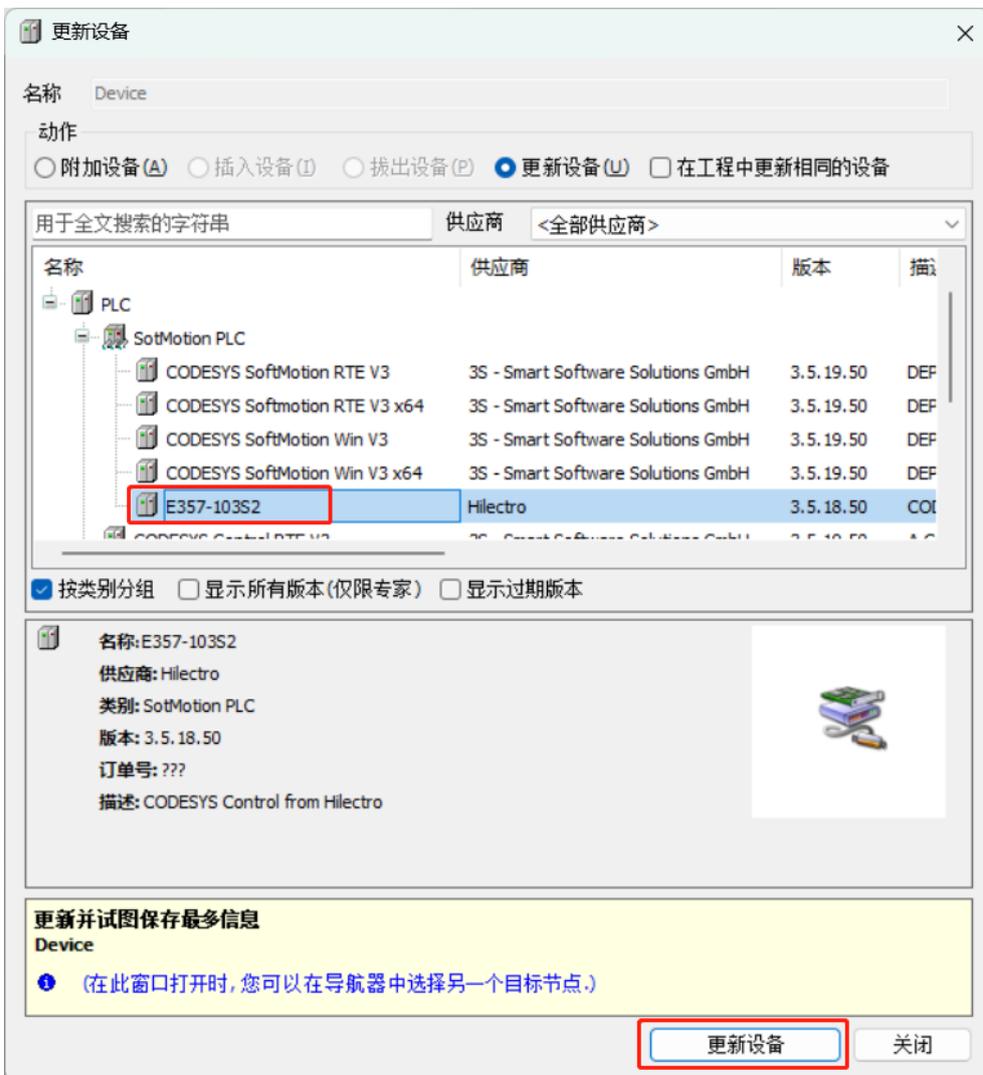
1) 创建一个 E357 的工程，但是由于没有安装该设备的描述文件，我们先创建一个默认工程。



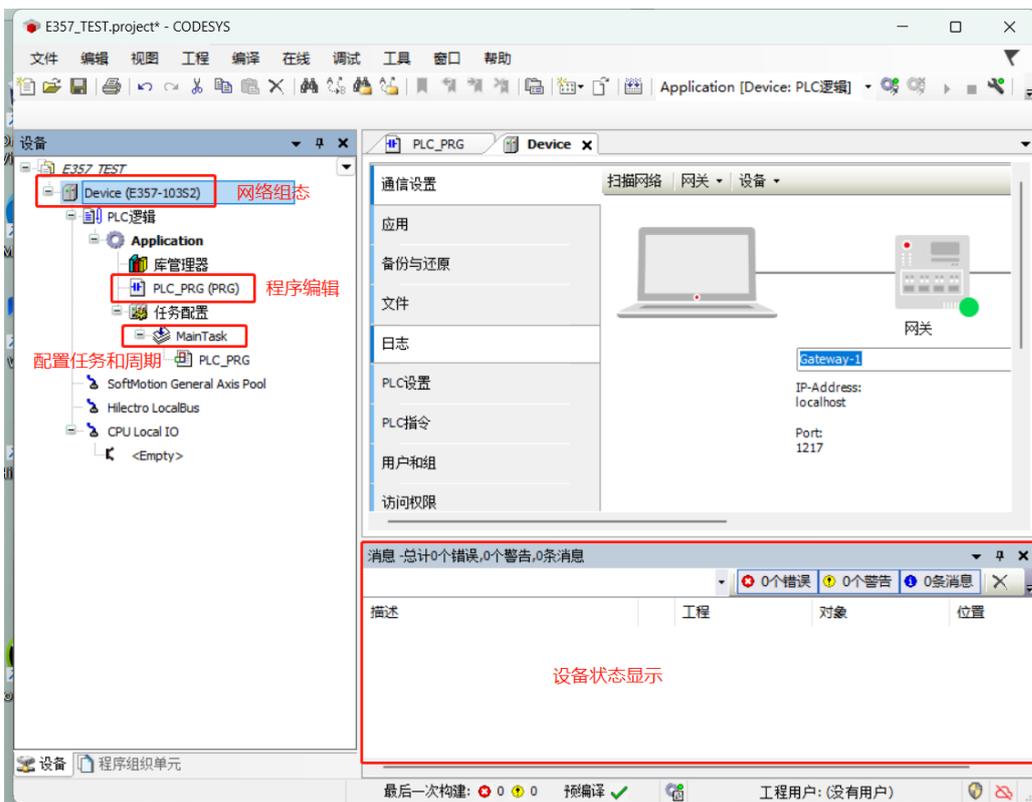
2) 进入工程界面，参考 3.3.1 [安装设备描述文件](#) 安装 E357 的设备描述文件。

3) 更新工程设备





4、工程基本界面如下:

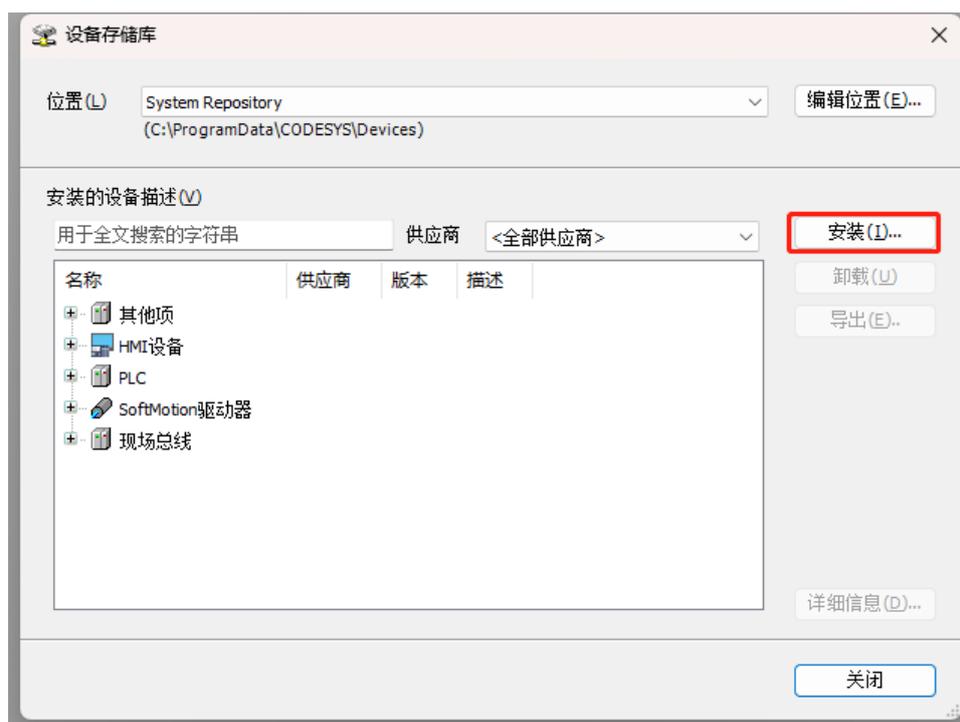
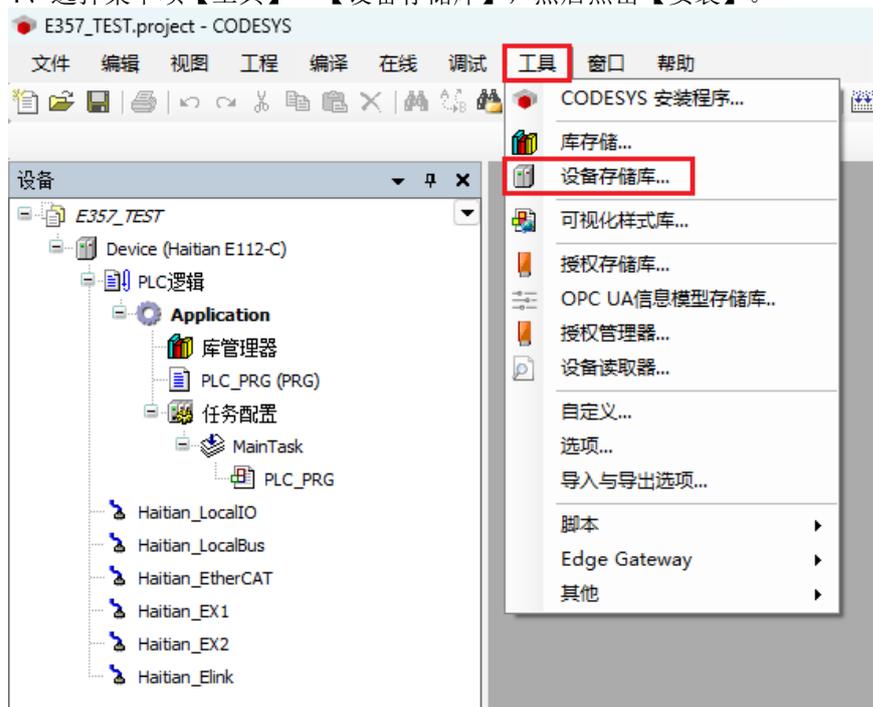


### 3.3 安装设备描述文件和库到 CODESYS 中

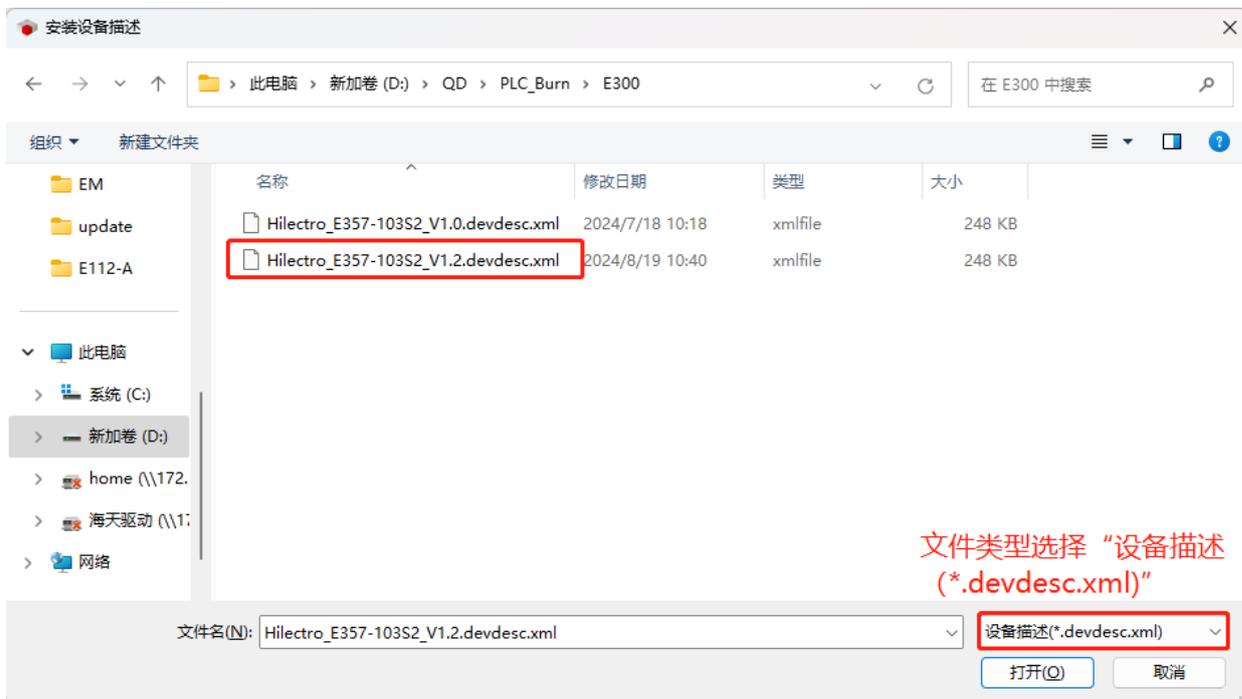
#### 3.3.1 安装设备描述文件

具体添加步骤如下：

1、选择菜单项【工具】→【设备存储库】，然后点击【安装】。

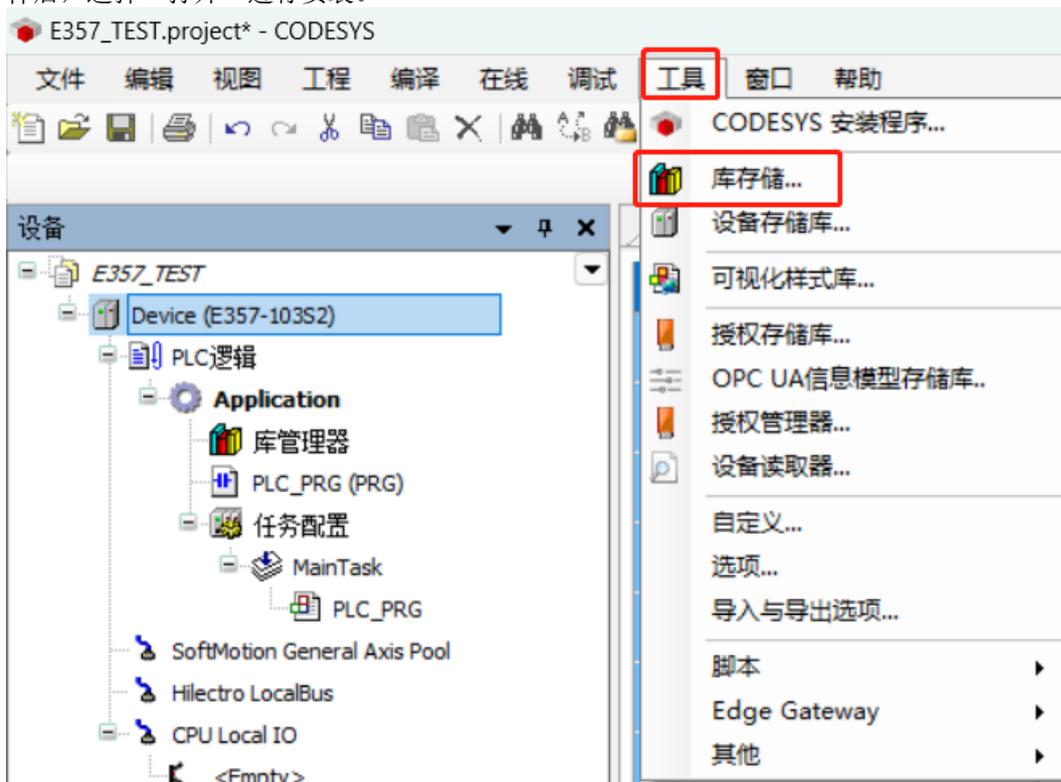


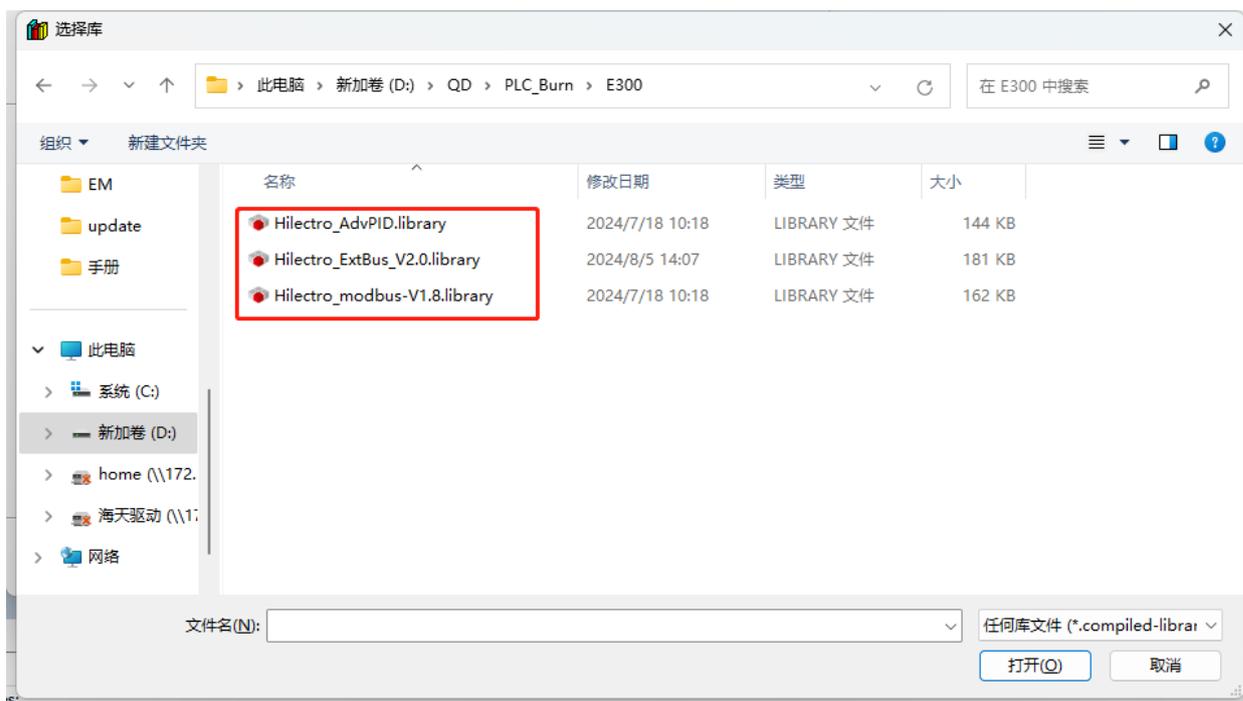
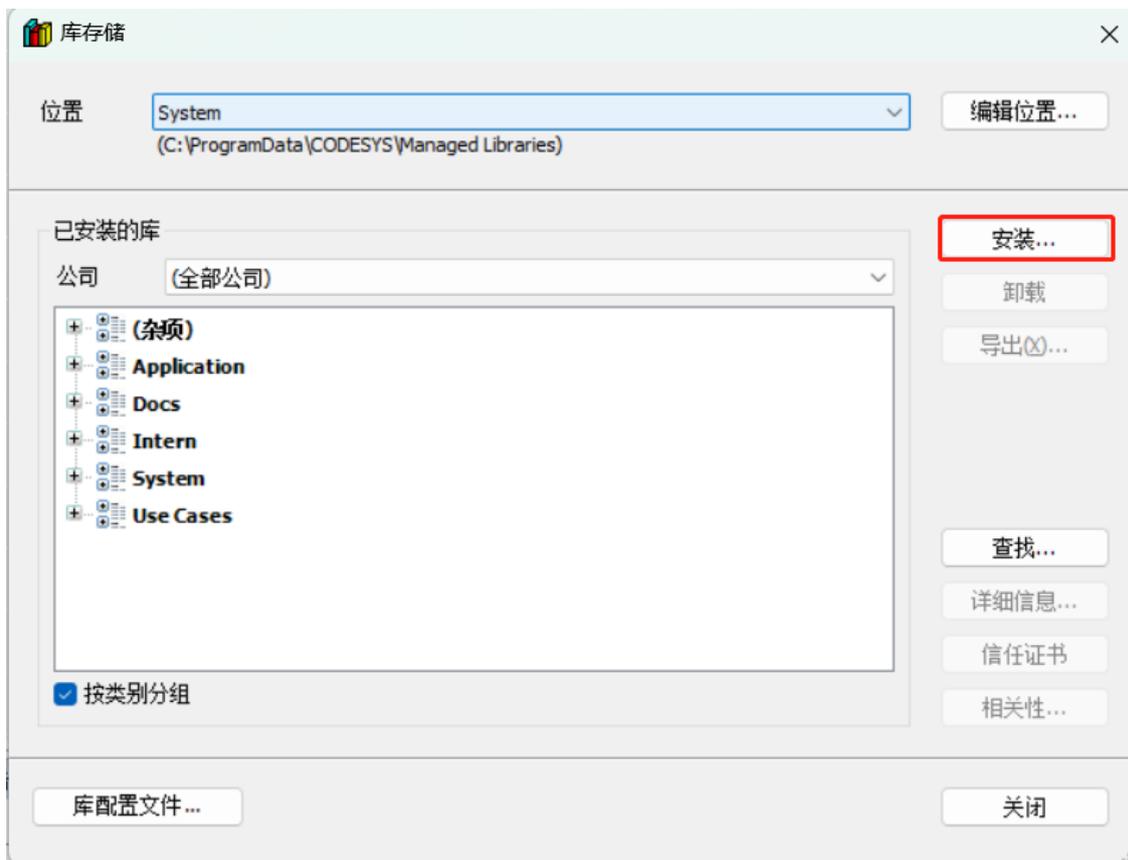
2、找到描述文件所在的路径后，可根据所需要安装的设备选择文件类型，选择设备后直接安装即可。



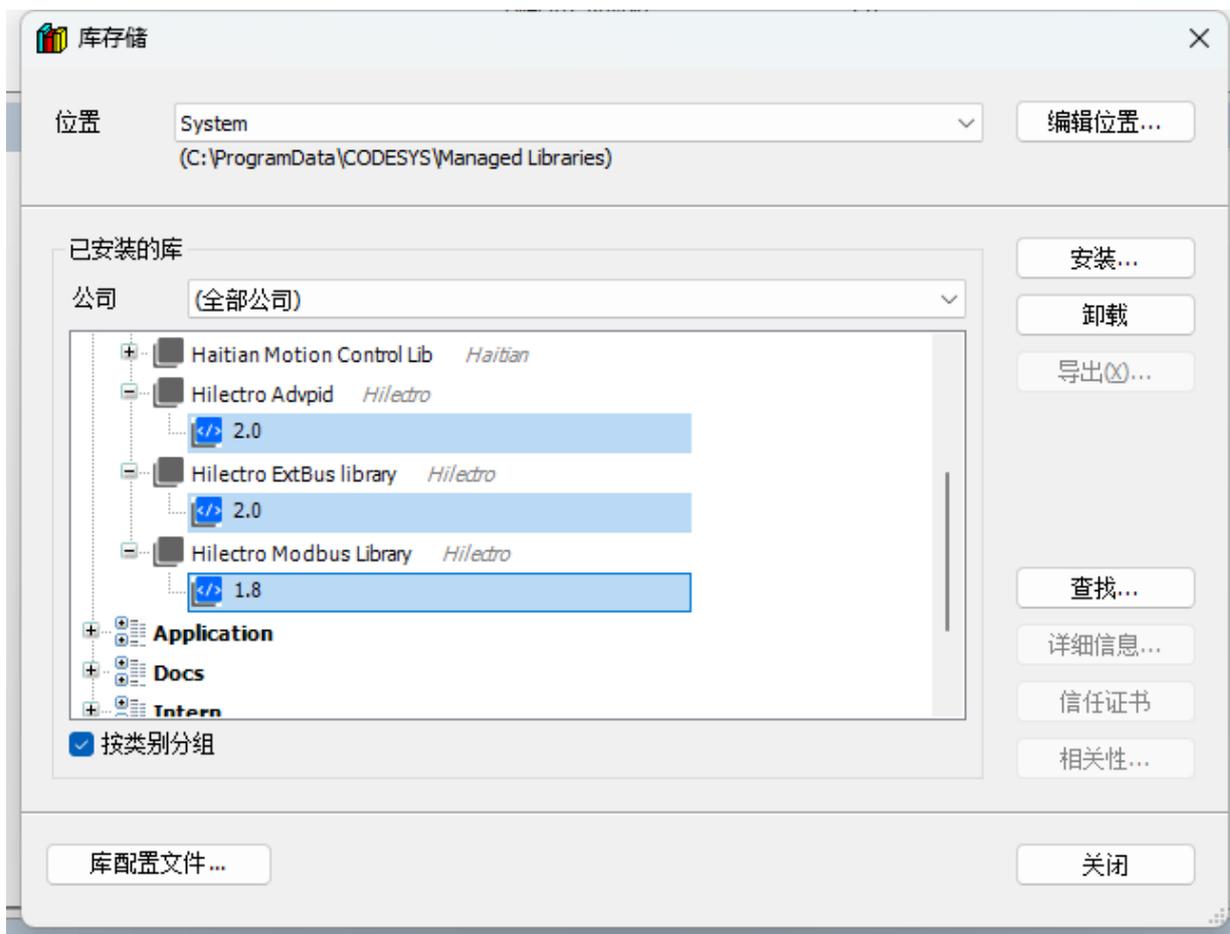
### 3.3.2 安装库

1、选择菜单栏【工具】下的【库存储】，在弹出对话框右侧选择【安装】，选择需要安装到系统中的库文件后，选择“打开”进行安装。





2、安装完毕可看到已安装库目录下的 HSC 库：



### 3.4 设置通信

#### 1、将编程设备的 IP 设为与 PLC 同一个网段

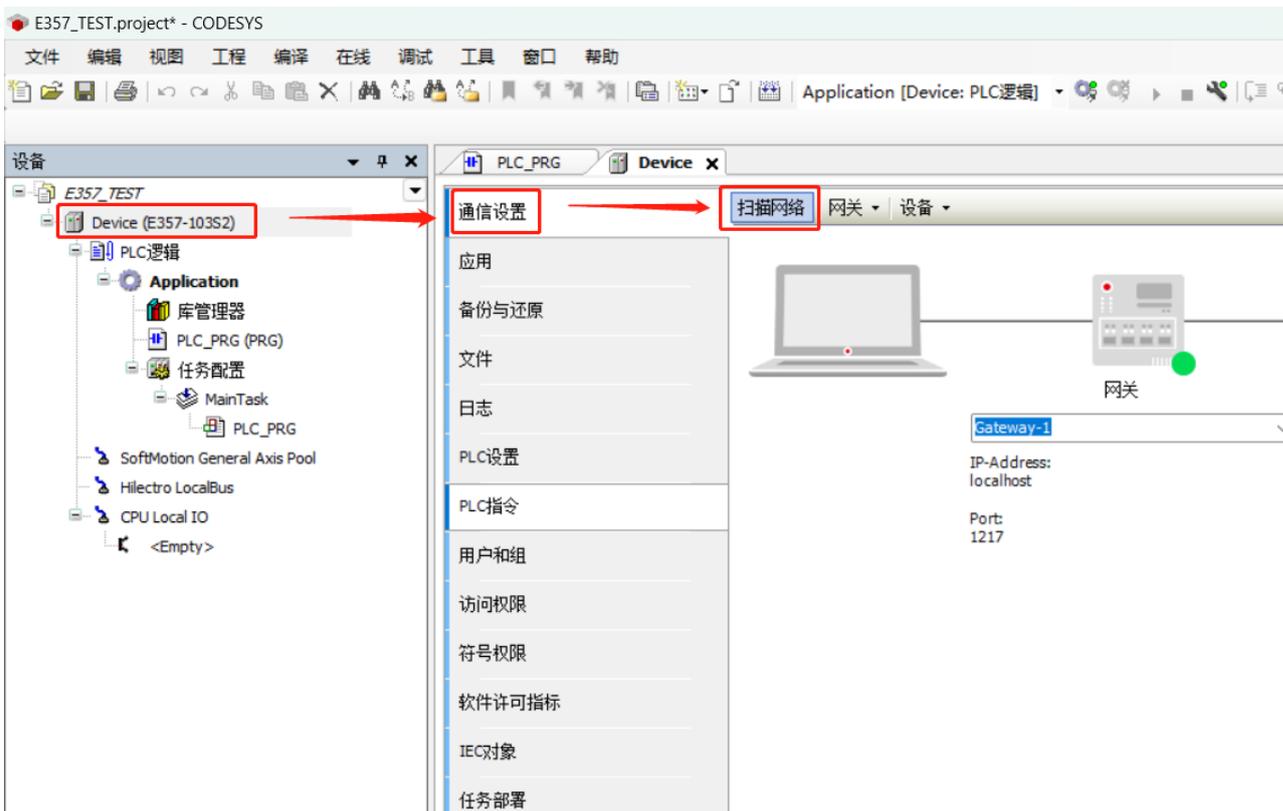
设置通信前，需要将编程 PC 的 IP 设置与 PLC（IP: 192.168.0.X）同一个网段。

设置方法：

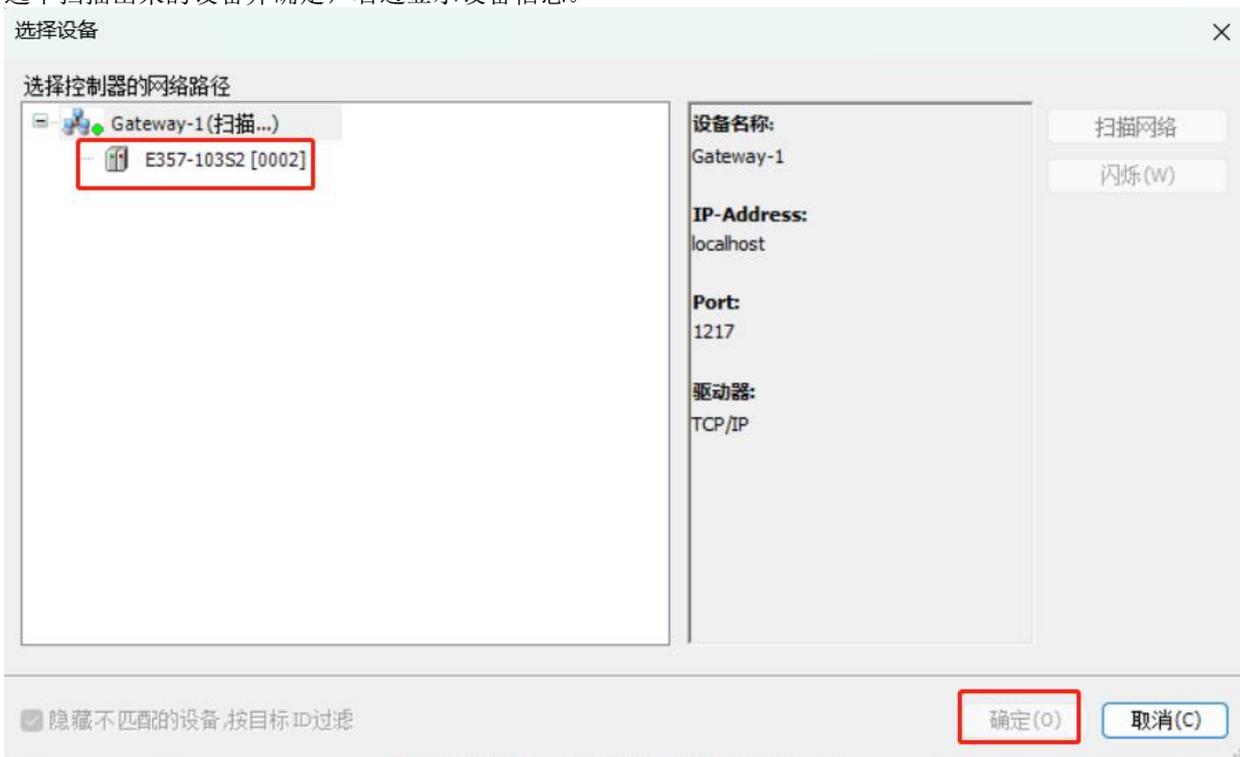
- 1) 将 PC 的以太网属性打开
- 2) 双击 Internet TCP/IP 协议
- 3) 将“自动获得 IP 地址”更改为“使用下面的 IP 地址”，然后在 IP 地址中填写“192.168.0.X”即可。

#### 2、在 CODESYS 设备视图中执行“通讯设置”

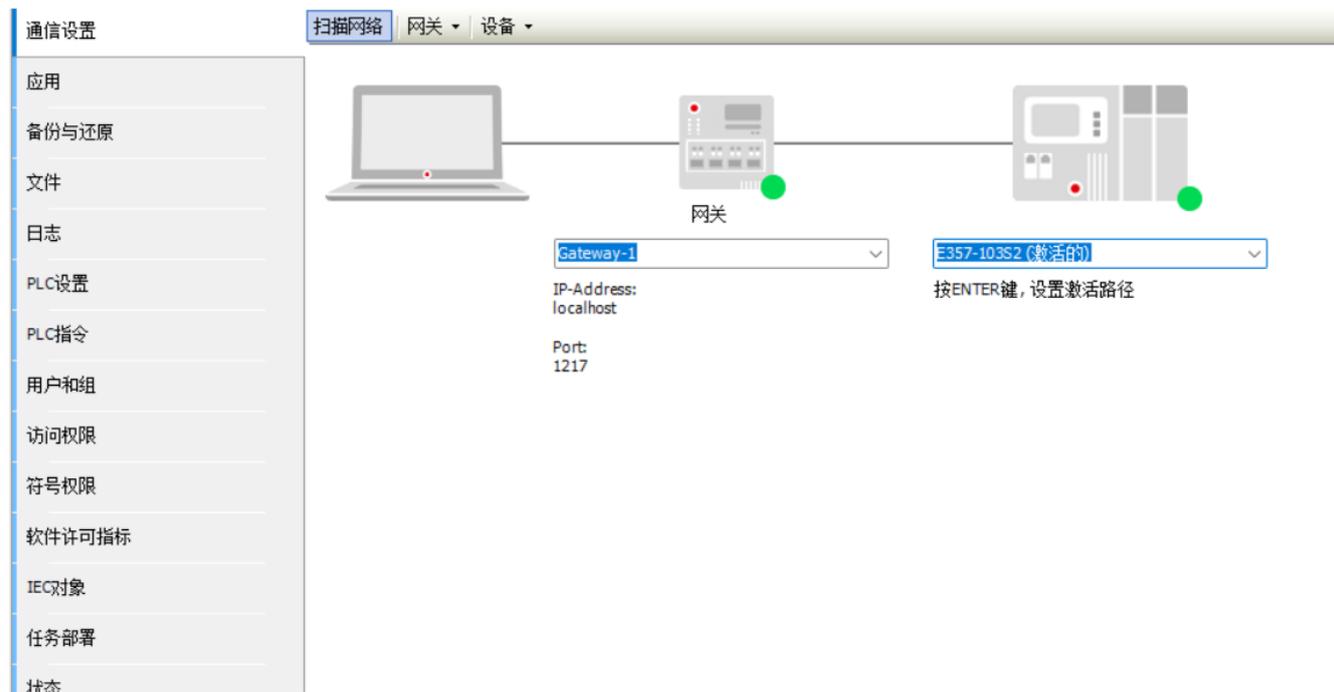
在设备视图中双击【Device】，然后在选项卡【通信设置】中点击【扫描网络】。



选中扫描出来的设备并确定，右边显示设备信息。

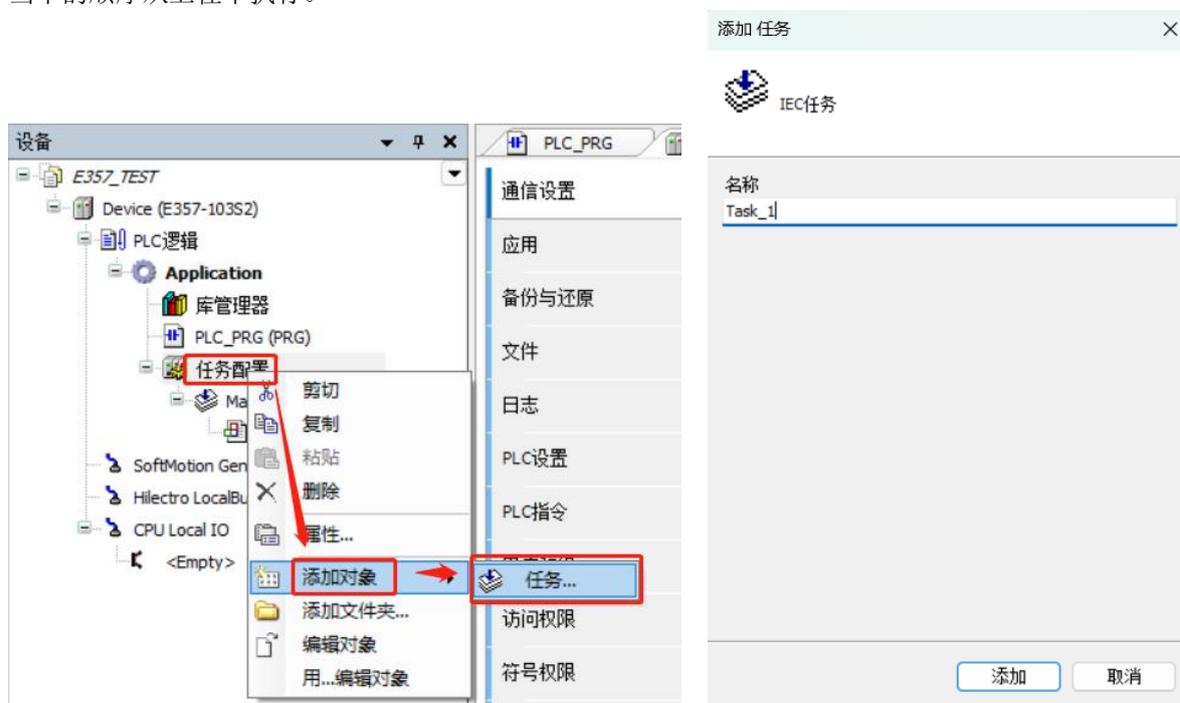


回到通讯设置界面可看到两个绿色指示灯，表示通讯通道激活，也就是所有与通讯相关的操作与该通道关联，设备成功通信。



### 3.5 配置任务

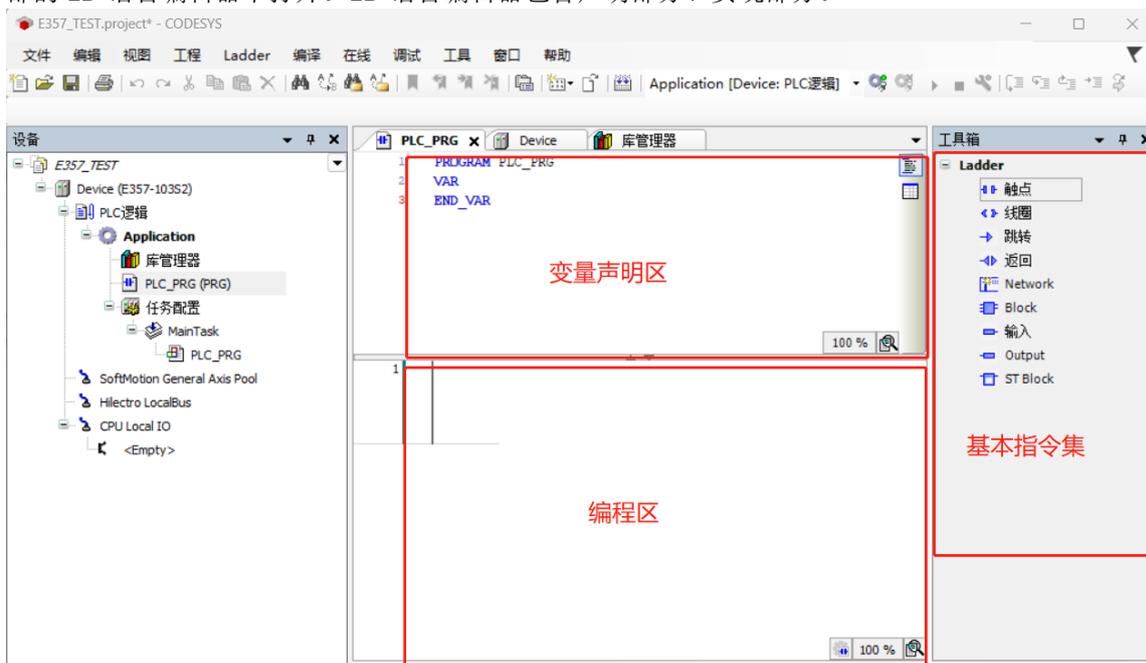
在树形菜单任务配置中可以进行管理，新建一个标准的 PLC 项目会自动生成一个循环执行的任务，该任务自动关联 PLC\_PRG，任务周期默认是 4ms，优先级为 1，PLC 程序只有被任务调用才会参与编译和实际执行。右击任务配置→添加对象→任务，定义任务名称后完成新任务的创建，不同类型的任务最多可以创建 100 个，按照用户设定的优先级顺序执行，数字越小优先级越高，优先级相同的情况下，按照在任务配置当中的顺序从上往下执行。



对于用户新建的 PLC 程序,需要手动进行任务配置和调用，否则程序不执行，双击 MainTask→添加调用→需要调用的 PLC 程序点击确定完成调用。

### 3.6 编写程序

在设备视窗中，缺省 POU 为“PLC\_PRG”，双击设备视图中“PLC\_PRG”，自动在 CODESYS 用户界面中部的 LD 语言编辑器中打开。LD 语言编辑器包含声明部分、实现部分。

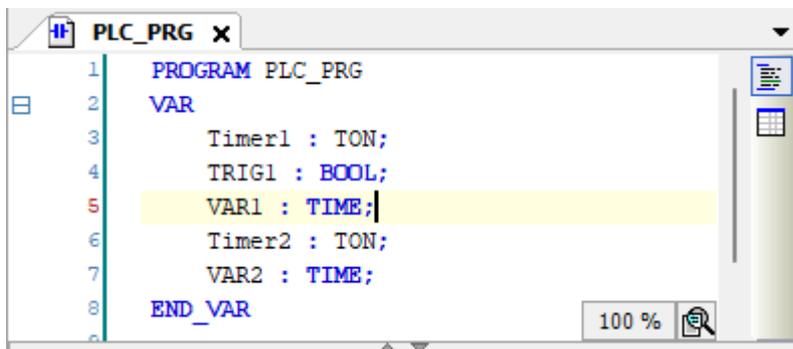


声明部分包括：显示在左侧边框中的行号、POU 类型和名称（如“PROGRAM PLC\_PRG”），以及在关键字“VAR”和“END\_VAR”之间的变量声明。实现部分为空，仅显示行号 1。

本例程序实现目的：定时器 1 和定时器 2 在 5S 间隔中周而复始的进行置位和复位操作。

### 1、在 PLC\_PRG 中声明变量

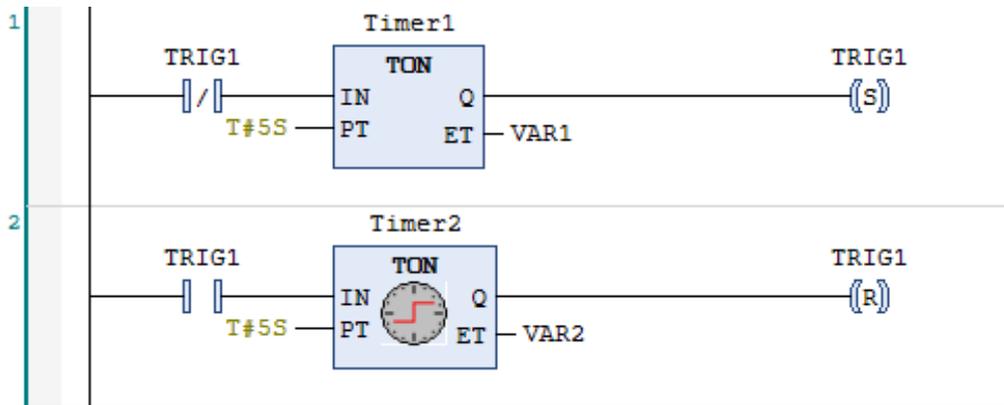
在编辑器的声明部分，将光标移到 VAR 后，点击回车，插入新的空行，声明需要使用的变量。或者在程序的实现部分，使用自动声明功能：在程序的实现部分输入指令，点击回车，如果在新行中有未声明的变量，则系统打开自动声明对话框，在此可以进行声明设置。本例声明部分内容如下图：



### 2、在 PLC\_PRG 的实现部分输入指令

在 LD 语言编辑器右侧的工具箱中展开“梯形图元素”，使用鼠标将“通电延时定时器 TON”拖曳到语言编辑器的实现部分，然后从工具箱的“梯形图元素”中拖曳一个复位线圈到 TON 指令输出点 Q 后方。使用同样的操作，在网络 2 中创建一个 TON 指令。

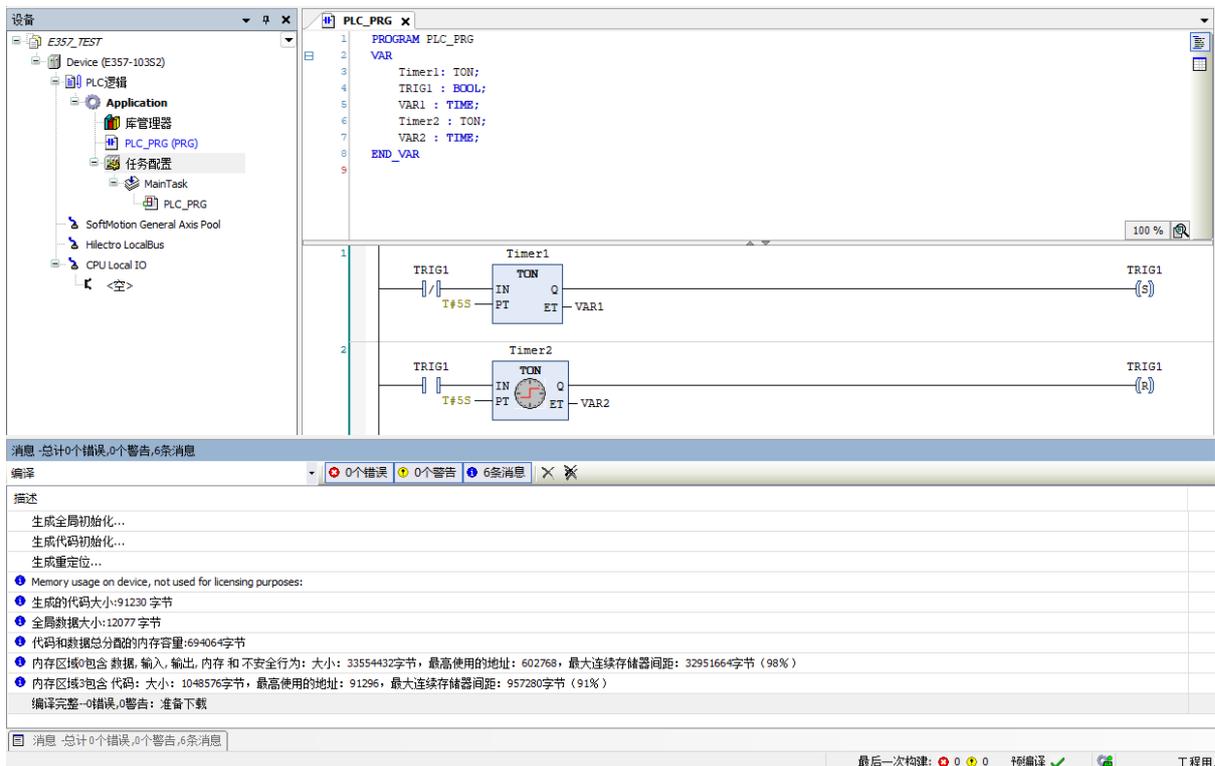
本例最终完成的 LD 程序如下：



## 3.7 编译和下载

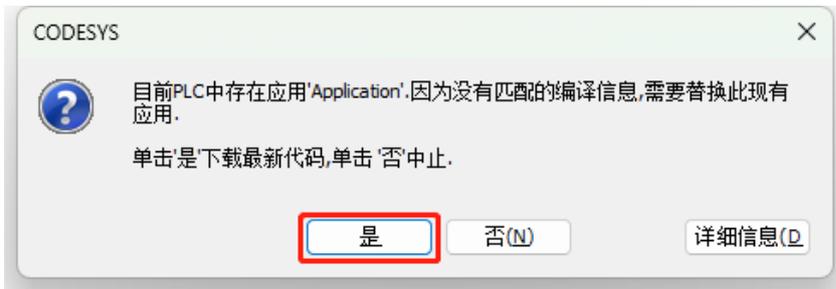
### 1、保存当前工程并进行编译

程序编写完成后，保存当前工程，然后选择菜单项“编译”即可对当前对象进行语法检查，语法检查完成时，任何错误消息和警告会在“编译”类的消息窗口中显示出来。若无错误，则表示编译成功。

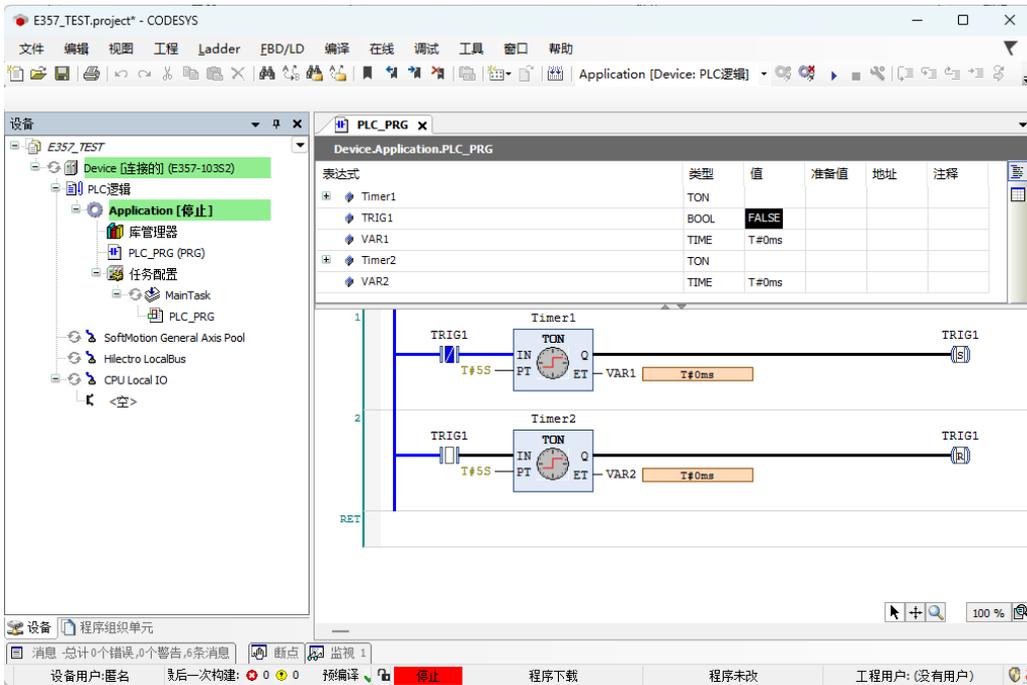


### 2、登录、下载和启动程序

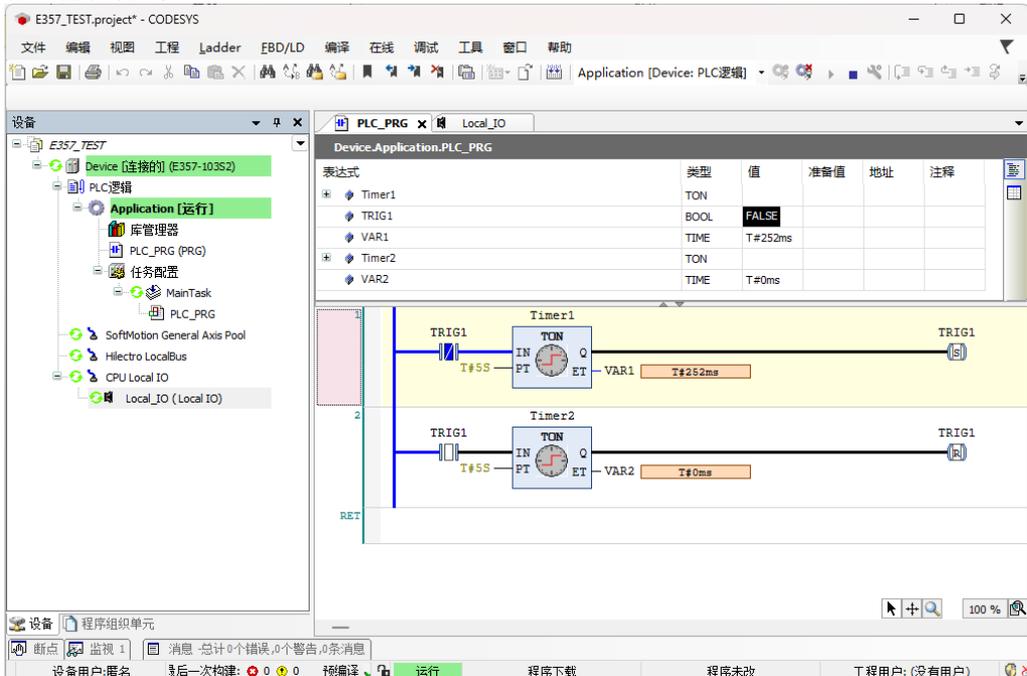
选择菜单项“在线” → “登录...”或直接点击工具栏图标，使应用程序与 PLC 建立起连接，并进入在线状态，如果已经进行了通讯设置，会弹出如下对话框提示：



点击“是”启动程序的下载，登录成功后的界面如下：



登录成功后，选择菜单项“调试”→“启动”或直接点击运行图标，运行 PLC 程序，此时可对当前工程进行监控和调试。



## 3.8 监控和调试

使用以下三种方法，可以监控应用程序中的变量和地址：

- 包含已定义监视列表的监视窗口
- 写入和强制变量
- 特殊 POU 的在线视图

### 1、打开程序的示例窗口

双击打开 PLC\_PRG，出现如下在线视图：上半部分显示对应于 PLC\_PRG 实现部分视图的程序主体，由每一个变量后的内部监视窗口显示实际的值。

The screenshot shows the 'Device.Application.PLC\_PRG' monitoring window. It features a table at the top listing variables and their current values, and a ladder logic diagram below.

表达式	类型	值	准备值	Address	注释
Timer1	TON				
TRIG1	BOOL	TRUE			
VAR1	TIME	T#0ms			
Timer2	TON				
VAR2	TIME	T#3s20ms			

The ladder logic diagram below the table shows two rungs. Rung 1 contains a normally open contact labeled 'TRIG1' connected to the 'IN' input of a 'Timer1' (TON) block. The 'PT' (preset time) of Timer1 is set to 't#5s'. The 'Q' output of Timer1 is connected to another 'TRIG1' coil. The 'ET' (elapsed time) output of Timer1 is connected to a variable 'VAR1', which is shown with a value of 'T#0ms'. Rung 2 contains a normally open contact labeled 'TRIG1' connected to the 'IN' input of a 'Timer2' (TON) block. The 'PT' of Timer2 is set to 't#5s'. The 'Q' output of Timer2 is connected to another 'TRIG1' coil. The 'ET' output of Timer2 is connected to a variable 'VAR2', which is shown with a value of 'T#3s20ms'.

### 2、写入和强制变量

通过写入或强制的方式将某一“准备值”赋给变量 TRIG1，在下一周期开始，变量将显示为该值。在“准备值”的输入框输入所需的整数值，单击回车或者该区域外部，然后执行命令“写入值”或者“强制值”，即可将该值写入或强制到 PLC。

### 3、使用监视视窗

选择菜单项“视图”→“监视”→“监视 1”打开监视窗口。然后，鼠标点击表达式列第一行，打开编辑框，输入要监视的变量 TRIG1 的完整路径：“Device.Application.PLC\_PRG.TRIG1”，随后即可对变量进行写入和强制值。

The screenshot shows the CODESYS interface for project E357\_TEST. The 'View' (视图) menu is open, and 'Monitoring' (监视) is selected. The main editor shows a ladder logic program with two TON timers. The monitoring table below shows the status of the monitored expression.

监视 1								
表达式	应用	类型	值	准备值	执行点	地址	注释	
PLC_PRG.TRIG1	Device.Application	BOOL	TRUE		循环监测			

# 模块的使用和状态描述

## 4

4.1 PLC 本机高速 IO 的使用说明

4.2 数字量模块的使用

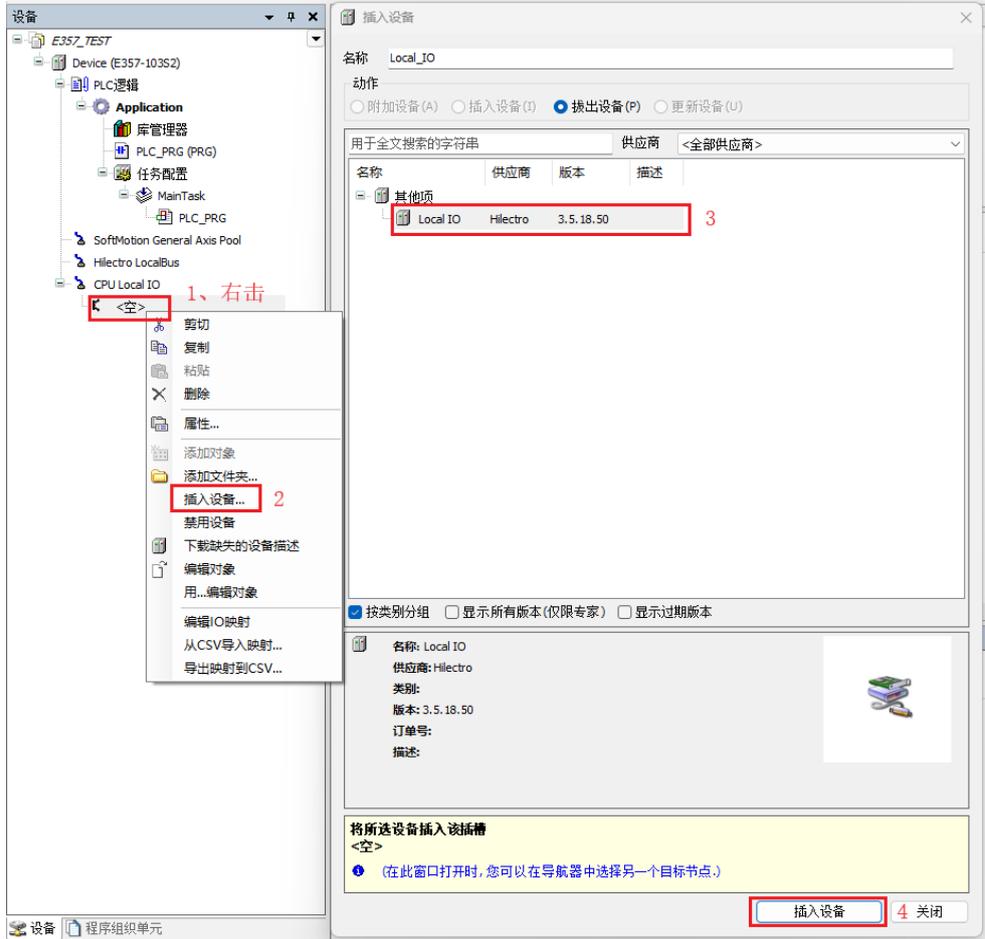
4.3 模拟量模块的使用

## 4.1 PLC 本机高速 IO 的使用说明

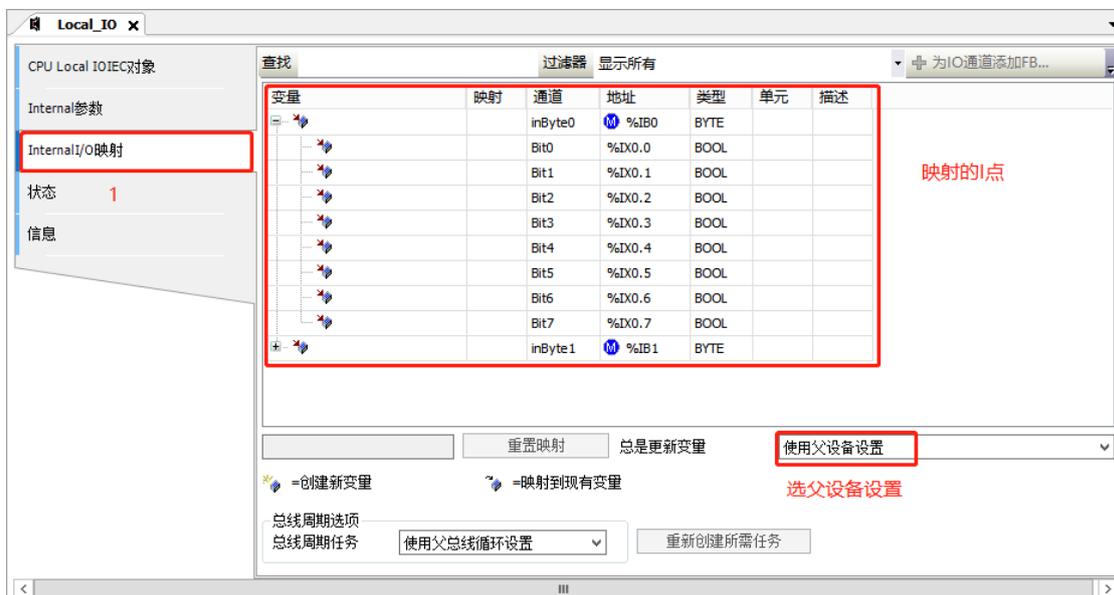
E357 自带 10 路 IO 口，下面将介绍本机 10 路 IO 的使用。

### 4.1.1 本机 IO 作为普通输入使用

作为普通输入 IO 时的接线与数字量输入模块的接线相同，在软件上的组态步骤如下：



2、添加完成后在右侧 General I/O 映射下就可以监控当前本地 I 点的状态。



## 4.1.2 本机 IO 作为高速输入使用

E357 本身集成 6 路高速计数器（HSC0~HSC5），也可搭载高速计数模块使用，本节介绍本机 IO 作为高速输入时的使用。

表4-1 E357 本机输入作为高速计数器时，高速计数器输入点及计数模式如下：

模式	计数器	描述		
	HSC0	I0.0	I0.1	I0.2
	HSC1	I0.3	I0.4	I0.5
	HSC2	I0.6	I0.7	--
	HSC3	I1.0	I1.1	--
	HSC4	I0.2	--	--
	HSC5	I0.5	--	--
0	带有内部方向控制的单相计数器	时钟	--	--
1		时钟	--	复位
2		不支持该计数模式		
3	带有外部方向控制的单相计数器	时钟	方向	--
4		时钟	方向	复位
5		不支持该计数模式		
6	带有增减计数时钟的两相计数器	增时钟	减时钟	--
7		增时钟	减时钟	复位
8		不支持该计数模式		
9	A/B 相正交计数器	时钟 A	时钟 B	--
10		时钟 A	时钟 B	复位
11		不支持该计数模式		

### 一、计数模式说明

1、E357 本身集成 6 路高速计数器，高速计数器的输入点不可自由分配，需按照上表的输入点进行配置。同一个输入点不能作用于多个高速计数器，只能作用于一个高速计数器。

2、HSC0 有三个输入，除了不支持上表中列出的 2、5、8、11 模式，HSC0 和 HSC1 支持余下所有的计数模式。当 HSC0 采用模式 4、7、10 时，I0.2 被占用，则 HSC4 不可用。同理，当 HSC1 采用模式 4、7、10 时，I0.5 被占用，HSC5 没有输入。

3、内部计数方向与外部计数方向的区别在于：内部计数方向是通过控制字直接计数，不需要接线和占用 I 点；外部计数方向则是通过外部输入信号启动计数。

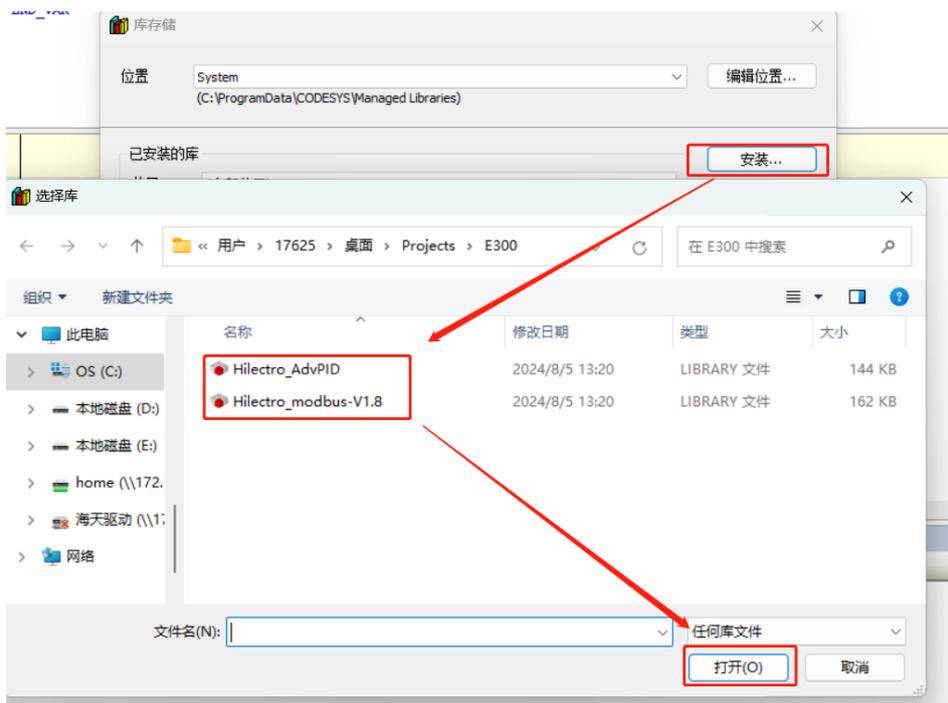
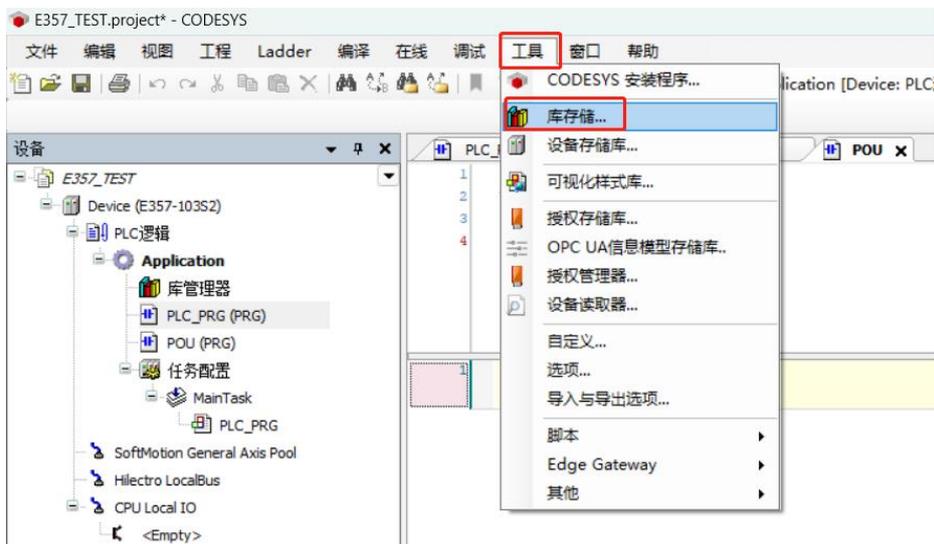
4、上表中的“时钟，增、减时钟，A、B 时钟”指的是脉冲输入信号，不是字面上的时钟。“复位”信号既可以复位计数器，也可以做一个捕获信号，具体可以通过 HSC\_SETMODE 进行设置。

5、计数单位：脉冲个数。

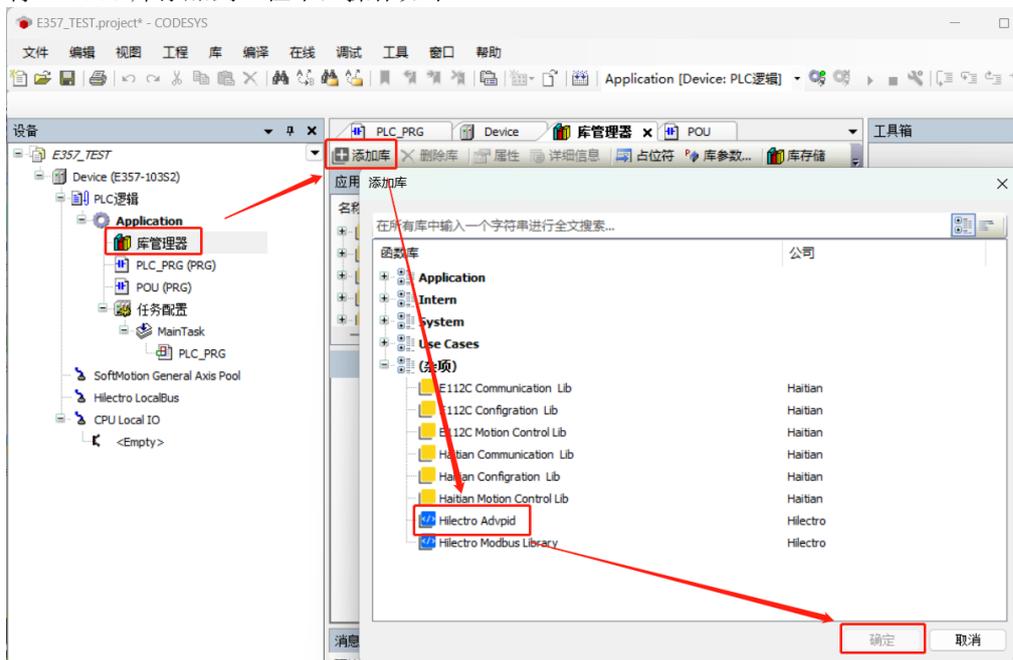
### 二、在 CODESYS 中组态高速计数器

1、在组态高速计数器之前，需要先将 Extbus 库安装到 CODESYS 中，然后添加该库到工程中方便调用。

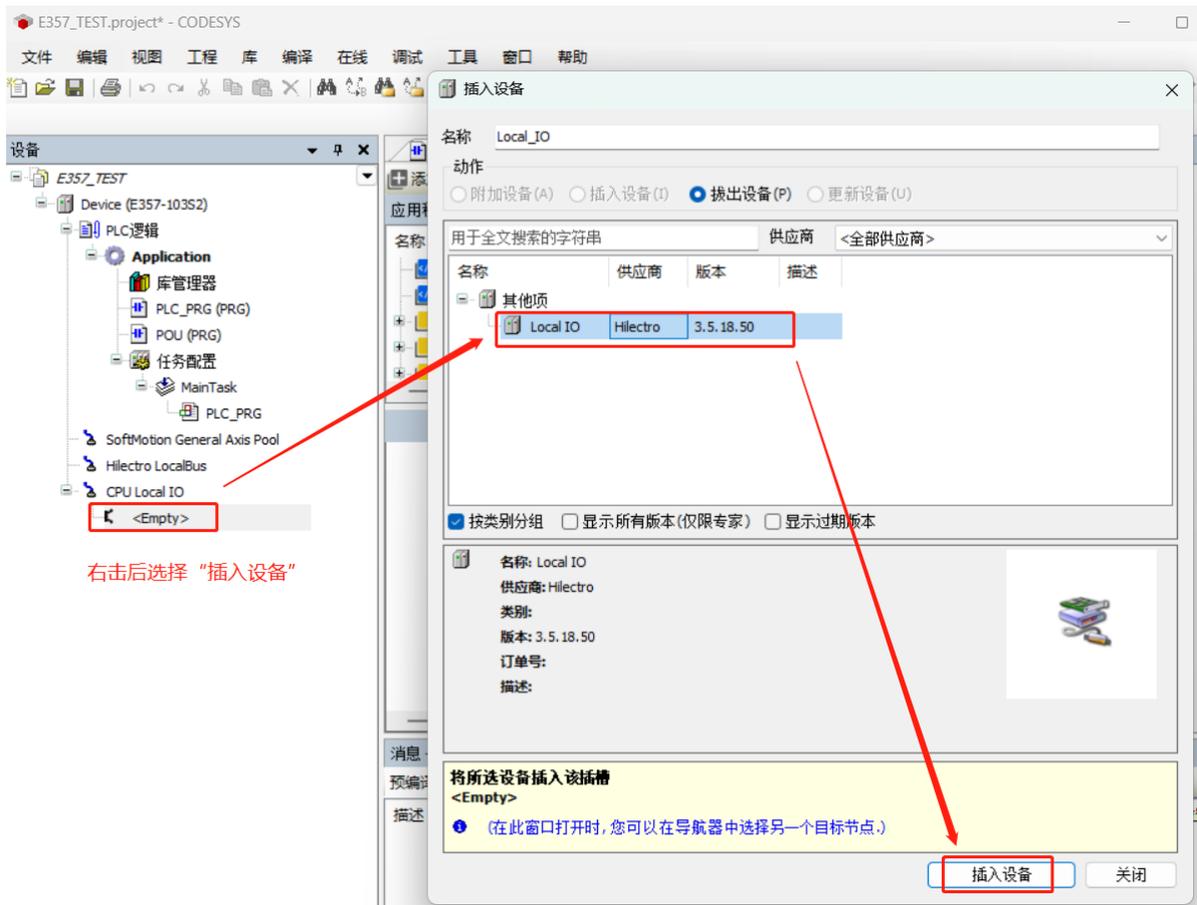
首先将 Extbus 库添加到 CODESYS 中，操作如下：



将 Extbus 库添加到工程中，操作如下：



2、组态高速计数输入，插入 PLC 自带的 IO。



3、在高速计数器的 **Internal** 参数中，可以配置高速计数器的参数，包括通道滤波、计数模式、控制字等，如下图所示。如果在 **Internal** 参数中配置了控制字、模式等参数，则无需调用 HSC\_300、HSC\_SETMODE 指令配置高速计数器参数以及计数模式。

假如既在 **Internal** 参数中设置了 HSC 模式和 HSC 控制字，又在程序中调用了 HSC\_300、HSC\_SETMODE 设置 HSC 模式和 HSC 控制字，则以 HSC\_300、HSC\_SETMODE 指令为准，因为 **Internal** 参数中的 HSC 模式和 HSC 控制字会被 HSC 300、HSC\_SETMODE 指令覆盖。

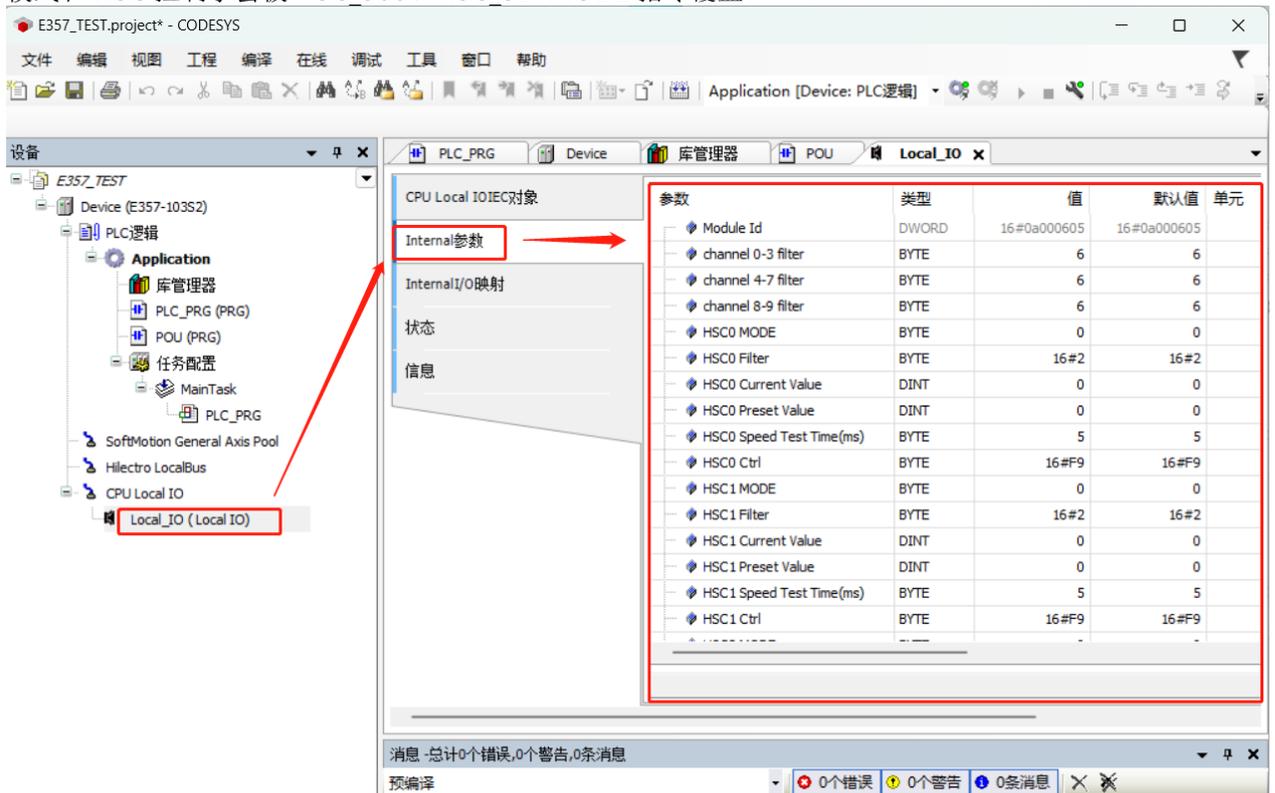
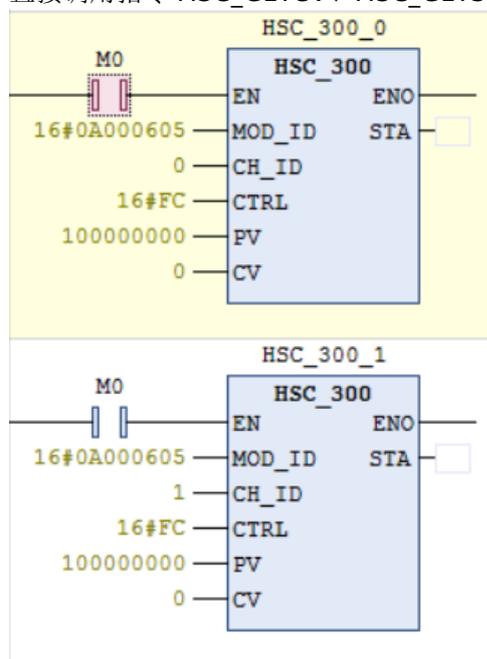


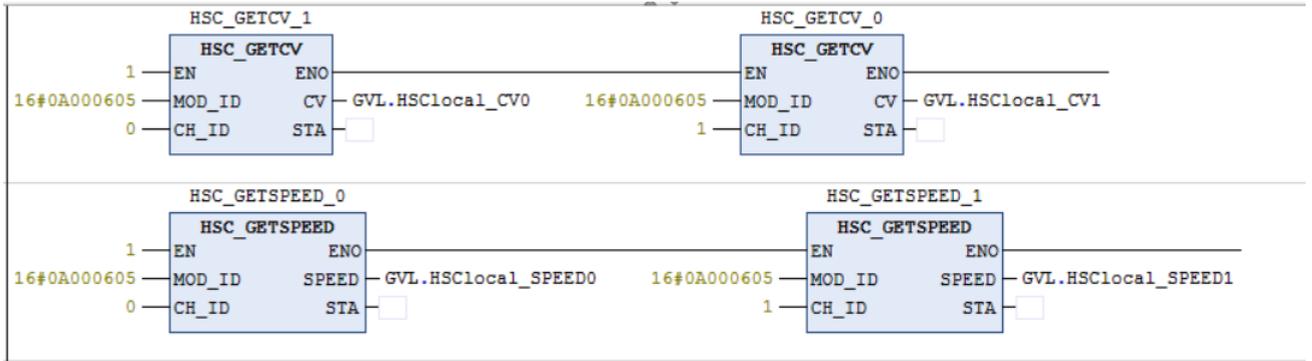
表4-2 高速计数器参数

参数	类型	值	默认值	描述
Module Id	DWORD	16#0a000605	16#0a000605	Module Id, Extbus 库指令中的 Module Id 指的就是 16#0a000605。
channel 0~9 filter	BYTE	6	6	滤波时间 1: 0.2ms, 2: 0.4ms, 3: 0.8ms, 4: 1.6ms, 5: 3.2ms 6: 6.4ms, 7: 12.8ms, 8: 0.2us, 9: 0.4us, 10: 0.8us 11: 1.6us, 12: 3.2us, 13: 6.4us, 14: 12.8us
HSCO MODE	BYTE	0	0	bit0~bit3: HSC 模式 (0, 1, 3, 4, 6, 7, 9, 10) bit4: Z lock disable, 0: enable, 1: disable bit5: Z clear disable, 0: enable, 1: disable bit6: reserve bit7: clear lock data 0:clear 1:not clear
HSC 0~5 Filter	BYTE	16#2	16#2	Bit0~Bit3: HSC Filter(Hz) 1: 750k, 2: 500k, 3: 375k, 4: 250k 5: 125k, 6: 100k, 7: 75k
HSC0~5 Current Value	DINT	0	0	
HSC0~5 Preset value	DINT	0	0	
HSC0~5 Speed Test Time(ms)	BYTE	5	5	
HSC0~5 Ctrl	BYTE	16#F9	16#F9	bit0: Reset Level, 0: Low, 1: High. bit1~bit2: quad Rate, 0: 4x, 1: 2x, 2: 1x. bit3: Direction, 0: Decrease, 1: Increase. bit4: Direction Update, 0: Not Update, 1: Update. bit5: Preset Value Update, 0: Not update, 1: Update. bit6: Current Value Update, 0: Not update, 1: Update. bit7: HSC Enable, 0: Disable, 1: Enable.

#### 4、调用 Extbus 库指令进行编程

直接调用指令 HSC\_GETCV、HSC\_GETSPEED、HSC\_STA 读取计数值、计数速度及计数状态等信息。



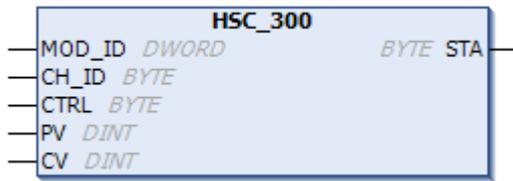


### 4.1.3 ExtBus 库指令说明

表4-3 高速计数器支持的指令库（ExtBus），各指令的描述参考如下说明。

指令	名称
HSC_300	设置计数参数指令
HSC_CLEARLOCK	清除锁存值
HSC_GETLOCK	获取当前锁存值
HSC_GETCV	获取当前计数值
HSC_GETSPEED	获取当前计数速度
HSC_GETSPEED-AVG	使用平均值获取当前速度
HSC_GETSTA	获取当前计数状态
HSC_SETMODE	设置计数器模式

## 1、设置计数参数指令



功能：设置计数参数。

参数

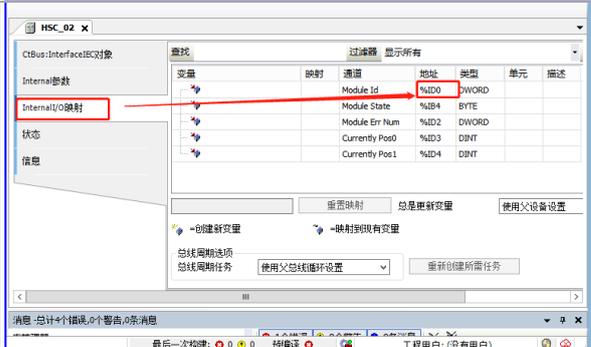
参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	<ul style="list-style-type: none"> <li>使用 PLC 本机高速计数，MOD_ID 为16#0a000605</li> <li>使用高速计数模块，MOD_ID 则对应高速计数模块 internal/0 映射中的 Module Id</li> </ul>
CH_ID	IN	通道地址	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
CTRL	IN	控制字	BYTE	16#F9	<b>bit0:</b> 复位电平，1-高电平复位，0-低电平复位。 <b>bit1~2:</b> 正交计数选择，00表示4倍正交模式，01表示2倍正交模式，10表示1倍正交模式。 <b>bit3:</b> 计数方向，0-减计数，1-增计数。 <b>bit4:</b> 计数方向更新，0-不更新，1-更新。 <b>bit5:</b> 预设值更新，0-不更新，1-更新。 <b>bit6:</b> 当前值更新，0-不更新，1-更新。 <b>bit7:</b> 计数使能，0-不使能，1-使能。
PV	IN	预设值	DINT	0	
CV	IN	当前值	DINT	0	
STA	OUT	返回状态	BYTE	0	模块状态字 0: OK; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层校验错误。

## 2、获取当前锁存值

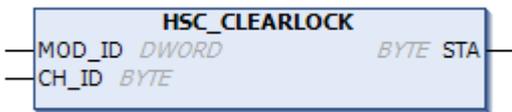


参数

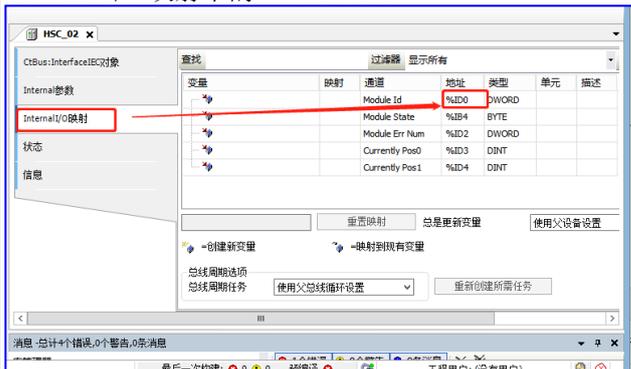
参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	<ul style="list-style-type: none"> <li>使用 PLC 本机高速计数，MOD_ID 为16#0a000605</li> <li>使用高速计数模块，MOD_ID 则对应高速计数模块 internal/0 映射中的 Module Id</li> </ul>

					
CH_ID	IN	通道地址	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
LOCK	OUT	锁存值	DINT	0	锁存值
STA	OUT	模块状态字	BYTE	0	模块状态字 0: OK; 2: 参数非法; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层校验错误。

### 3、清除锁存值



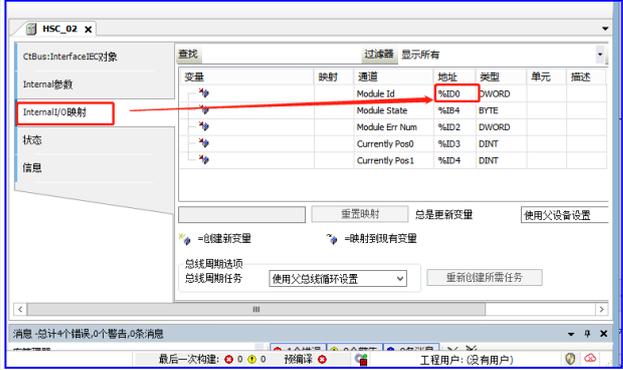
#### 参数

参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	<ul style="list-style-type: none"> <li>使用 PLC 本机高速计数，MOD_ID 为16#0a000605。</li> <li>使用高速计数模块，MOD_ID 则对应高速计数模块 internal/I/O 映射中的 Module Id。</li> </ul> 
CH_ID	IN	通道地址	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
STA	OUT	模块状态字	BYTE	0	模块状态字 0: OK; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层校验错误。

### 4、获取当前计数值



参数

参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	<ul style="list-style-type: none"> <li>使用 PLC 本机高速计数，MOD_ID 为16#0a000605</li> <li>使用高速计数模块，MOD_ID 则对应高速计数模块 internal/O 映射中的 Module Id</li> </ul> 
CH_ID	IN	通道	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
CV	OUT	当前计数值	DINT	0	当前值
STA	OUT	模块状态字	BYTE	0	模块状态字 0: OK; 2: 参数非法; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层较验错误。

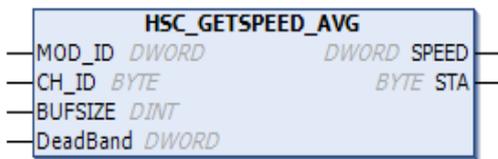
5、获取当前计数速度



参数

参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	<ul style="list-style-type: none"> <li>使用 PLC 本机高速计数，MOD_ID 为16#0a000605</li> <li>使用高速计数模块，MOD_ID 则对应高速计数模块 internal/O 映射中的 Module Id。</li> </ul>
CH_ID	IN	通道地址	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
SPEED	OUT	计数速度	DWORD	0	Hz
STA	OUT	模块状态字	BYTE	0	模块状态字 0: OK; 2: 参数非法; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层较验错误。

6、使用平均值获取当前速度



参数

参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	<ul style="list-style-type: none"> <li>使用 PLC 本机高速计数，MOD_ID 为16#0a000605。</li> <li>使用高速计数模块，MOD_ID 则对应高速计数模块 internal/O 映射中的 Module Id。</li> </ul>
CH_ID	IN	通道地址	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
BUFSIZE	IN	平均值缓冲区	DINT	16	平均值缓冲区大小大于0小于64
DeadBand	IN	死区值	DWORD	20000	平均值与现有值的差值小于死区值，以平均值为准，大于死区平均值为现有值
SPEED	OUT	计数速度	DWORD	0	Hz
STA	OUT	模块状态字	BYTE	0	模块状态字 0: OK; 2: 参数非法; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层校验错误。

7、获取当前计数状态



参数

参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	<ul style="list-style-type: none"> <li>使用 PLC 本机高速计数，MOD_ID 为16#0a000605。</li> <li>使用高速计数模块，MOD_ID 则对应高速计数模块 internal/O 映射中的 Module Id。</li> </ul>
CH_ID	IN	通道地址	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
HSC_STA	OUT	计数状态	BYTE	0	bit0~bit3: 当前模式 bit4: 预留 bit5: HSC0当前计数方向: 1=增计数 bit6=1: 当前值等于预设值位 bit7=1: 当前值大于预设值位
STA	OUT	模块状态字	BYTE	0	模块状态字 0: OK; 2: 参数非法; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层校验错误。

8、设置计数器模式



参数

参数名	输入输出	参数描述	类型	初始化	备注
MOD_ID	IN	模块地址	DWORD	0	• 使用 PLC 本机高速计数，MOD_ID 为16#0a000605。使用高速计数模块，MOD_ID 则对应高速计数模块 internal/0 映射中的 Module Id。
CH_ID	IN	通道地址	BYTE	0	通道地址，PLC本机取值0~5，高速计数模块取值0~1。
MODE	IN	计数模式	BYTE	0	bit0~bit3: HSC 计数模式（本机支持 0, 1, 3, 4, 6, 7, 9, 10; HSC-02 支持 0~11 模式） 模式 0~2: 具有内部方向控制的单相计数器。 模式 3~5: 具有外部方向的单相计数器。 模式 6~8: 具有 2 个时钟输入的双相计数器。 模式 9~11: A\B 相正交计数器。 bit4: Z 信号锁存功能，0: 锁存，1: 不锁存 bit5: Z 信号清零功能，0: 清零，1: 不清零 bit6: 预留 bit7: 锁存值清零，0: 无效，1: 有效
STA	OUT	模块状态字	BYTE	0	模块状态字 0: OK; 2: 参数非法; 5: 模块参数错误; 7: 模块没响应; 8: 模块链路层校验错误。

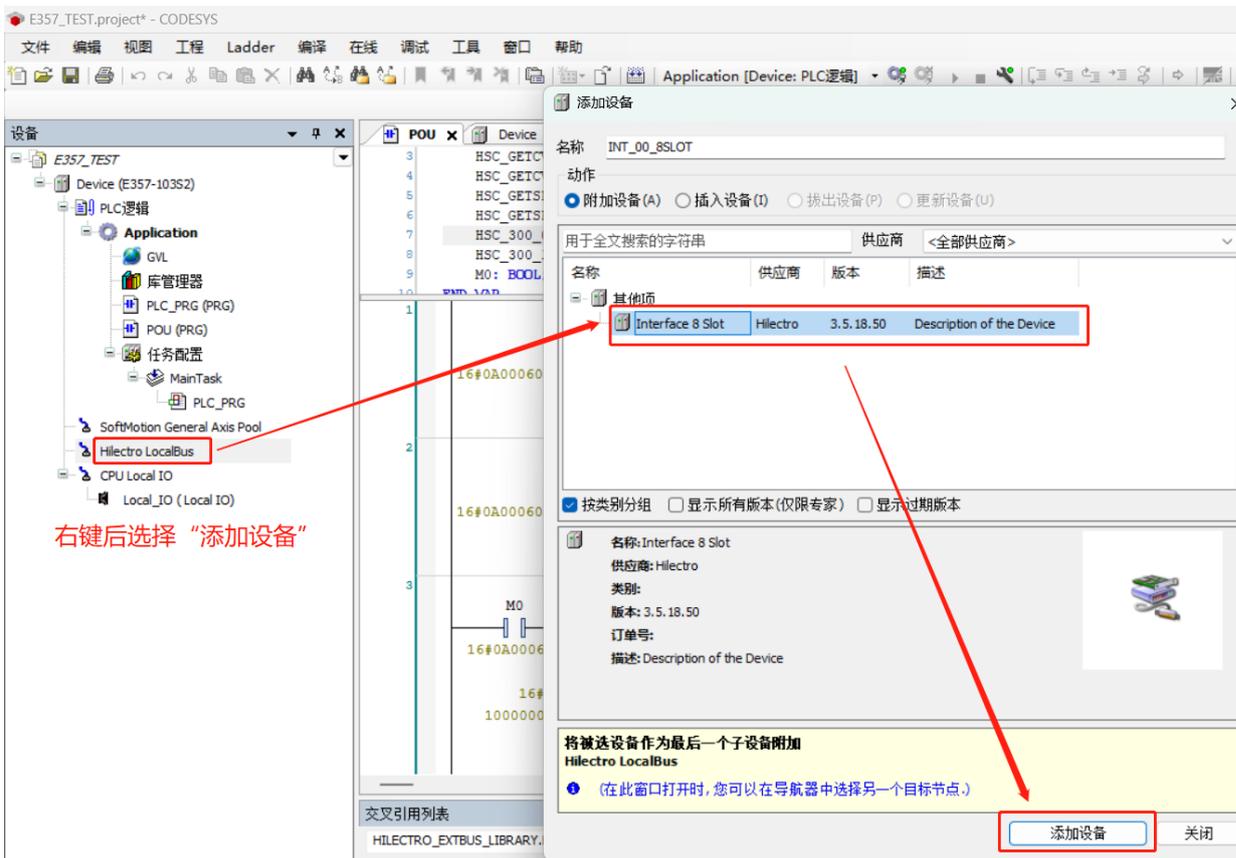
## 4.2 数字量模块的使用

本节做一个跑马灯工程组态示例，E357 做主站，简单说明数字量模块的使用。实现的功能如下：**让中继模块上的输出模块端口进行跑马灯输出，每 2 秒由低向高移动 1 位，在 8bit 范围内循环移动。**

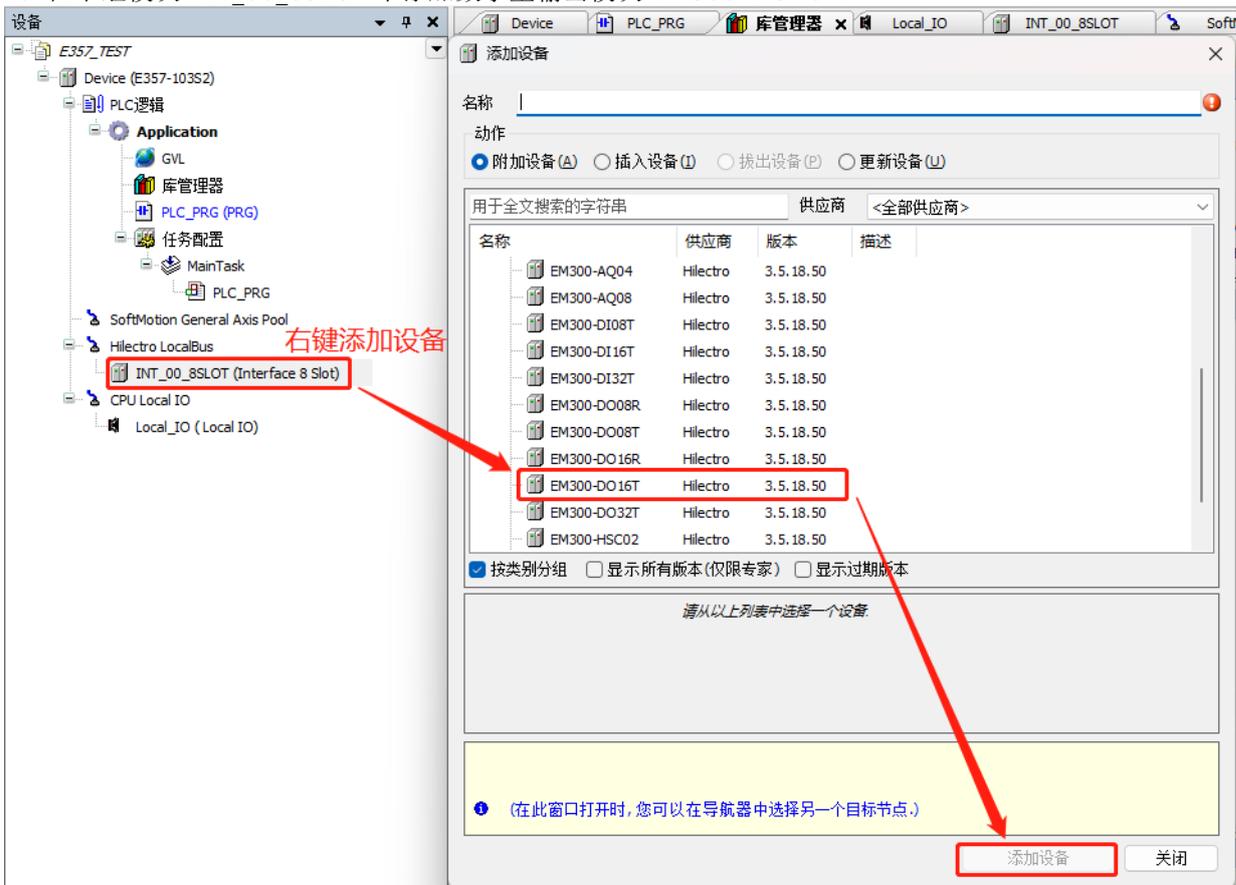
整个工程组态的大概步骤如下：

### 步骤 1、新建工程，添加中继模块和数字量输出模块至工程中

1) 打开 CODESYS SP19 新建一个用户工程，在设备视图右键点击“Hilectro LocalBus”并选择“添加设备”，即可在弹出的对话框中选择并添加中继模块，参考如下示意图：



2) 在中继模块 INT\_00\_8SLOT 下添加数字量输出模块 EM300-DO16T。



## 步骤 2、编写程序

在 PLC\_PRG 中编写跑马灯程序, 实现每隔 2s 就由低位向高位移动。

变量声明:

```

1 PROGRAM PLC_PRG
2 VAR
3     ton1:Standard.TON;
4     q00:BOOL;
5     q01:BOOL;
6     q02:BOOL;
7     q03:BOOL;
8     q04:BOOL;
9     q05:BOOL;
10    q06:BOOL;
11    q07:BOOL;
12    index:INT;
13

```

主程序:

```

1 ton1(IN:=NOT ton1.Q,PT:=T#2S);
2 CASE index OF
3     0:
4     q00:=1;q01:=0;q02:=0;q03:=0;q04:=0;q05:=0;q06:=0;q07:=0;
5     IF ton1.Q THEN
6         index:=1;
7     END_IF
8     1:
9     q00:=0;q01:=1;q02:=0;q03:=0;q04:=0;q05:=0;q06:=0;q07:=0;
10    IF ton1.Q THEN
11        index:=2;
12    END_IF
13    2:
14    q00:=0;q01:=0;q02:=1;q03:=0;q04:=0;q05:=0;q06:=0;q07:=0;
15    IF ton1.Q THEN
16        index:=3;
17    END_IF
18    3:
19    q00:=0;q01:=0;q02:=0;q03:=1;q04:=0;q05:=0;q06:=0;q07:=0;
20    IF ton1.Q THEN
21        index:=4;
22    END_IF
23    4:
24    q00:=0;q01:=0;q02:=0;q03:=0;q04:=1;q05:=0;q06:=0;q07:=0;
25    IF ton1.Q THEN
26        index:=5;
27    END_IF
28    5:
29    q00:=0;q01:=0;q02:=0;q03:=0;q04:=0;q05:=1;q06:=0;q07:=0;
30    IF ton1.Q THEN
31        index:=6;
32    END_IF
33    6:
34    q00:=0;q01:=0;q02:=0;q03:=0;q04:=0;q05:=0;q06:=1;q07:=0;
35    IF ton1.Q THEN
36        index:=7;
37    END_IF
38    7:
39    q00:=0;q01:=0;q02:=0;q03:=0;q04:=0;q05:=0;q06:=0;q07:=1;
40    IF ton1.Q THEN
41        index:=0;
42    END_IF
43 END_CASE
44
45

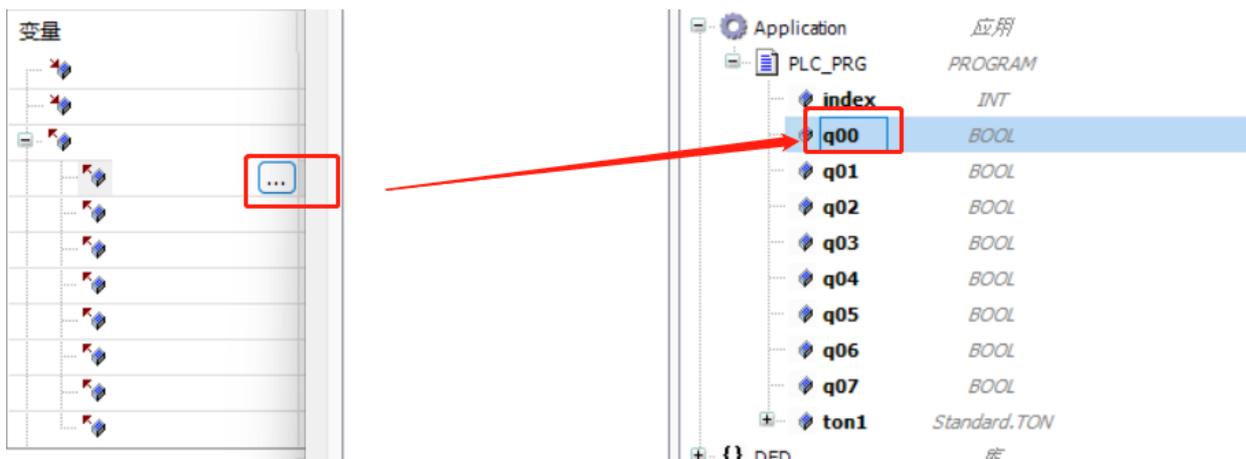
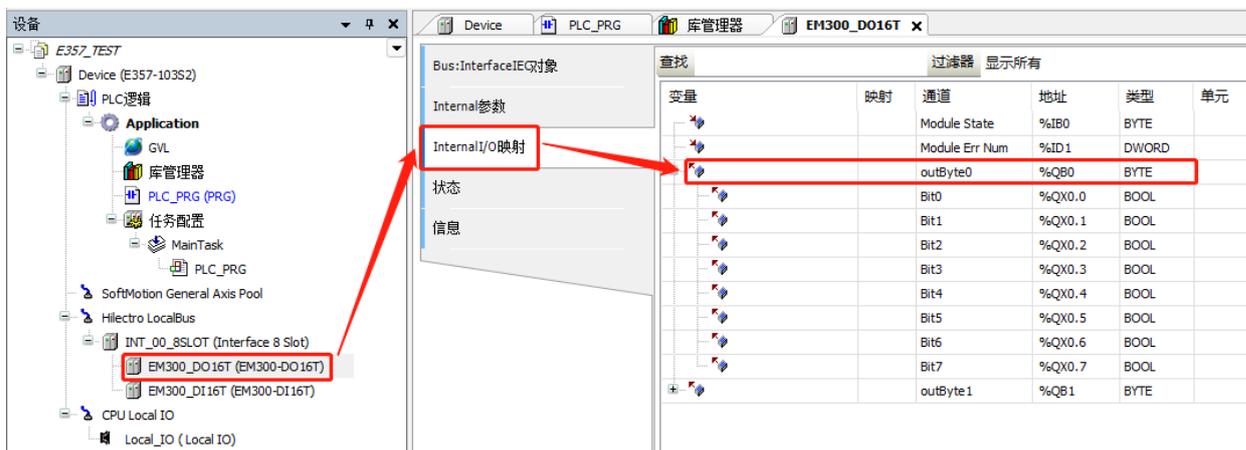
```

### 步骤 3、设置通信

参考 3.4 章节[设置通信](#)设置 PLC 与 CODESYS SP19 通信。

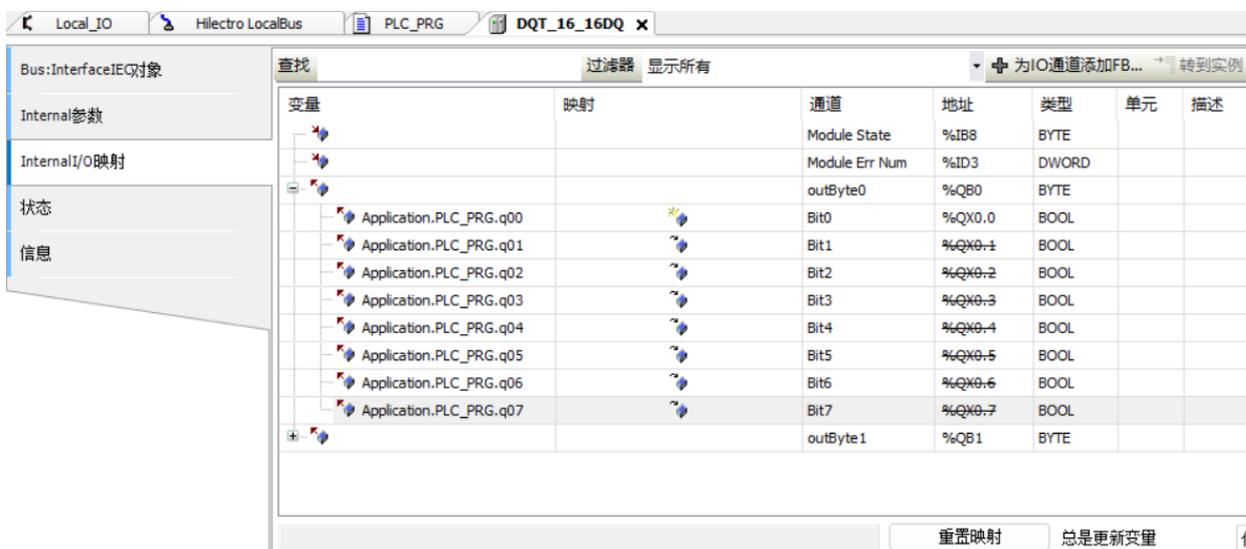
### 步骤 4、将用户程序变量与数字量输出模块端口进行关联

1) 双击 EM300\_DO16T, 进入 Internal I/O 映射, 展开 outByte0 下的变量, 然后给每一个 bit 分配一个变量。



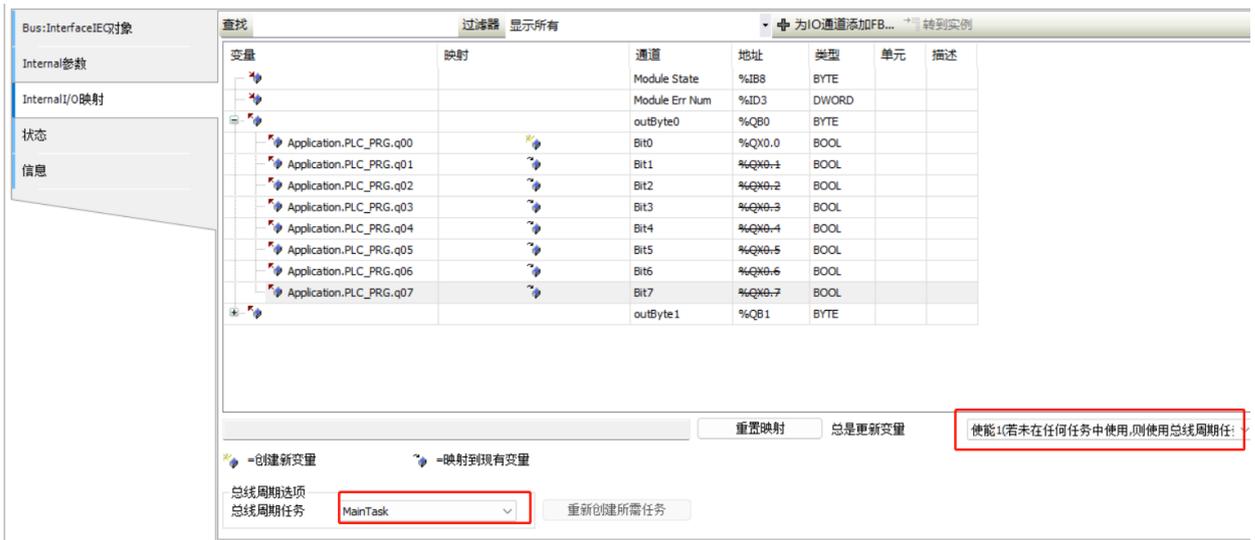
2

）变量与数字量输出模块端口关联完毕如下：



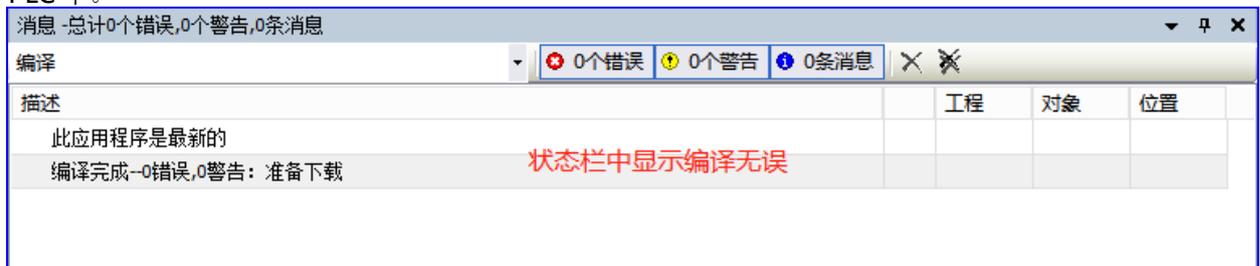
3) 总线周期任务选择“Main Task”，总线循环采用“启用 1（如果未在任何任务中使用，则使用总线循环任务）”。

- 使用父设备设置：根据上位设备的设置进行更新。
- 启用 1（如果未在任何任务中使用，则使用总线循环任务）：CODESYS 如果总线循环任务中的 I/O 变量未用于任何其他任务，则更新它们。



### 步骤 5、编译，登录

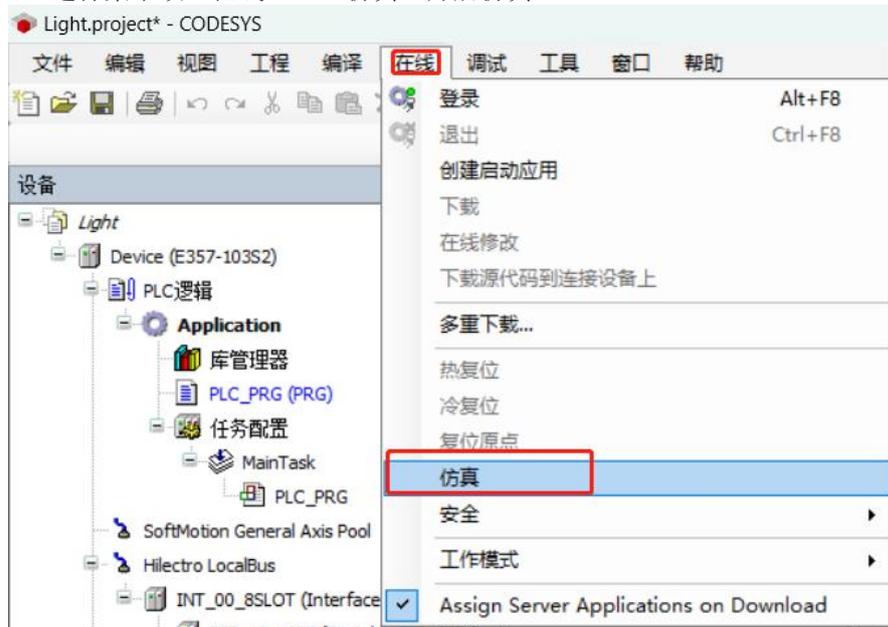
首先点击 编译工程，或者直接按 F11 键编译生成代码，编译无错后点击登录图标 就可以将工程下载到 PLC 中。



### 步骤 6、运行，监控

下面通过仿真演示跑马灯程序的执行结果，首先开启仿真功能，开启仿真后再登录至 PLC。开启仿真就不需要进行步骤 3，直接登录运行就可以在数字量输出扩展模块的 IO 映射中查看执行结果。

1) 选择菜单项“在线” → “仿真”开启仿真



2) 点击登录图标 就可以将应用程序与虚拟 E357 建立起连接，并进入在线状态。然后点击“启动”图标 使 E357 中的应用程序开始运行，进入 EM300-DO16T 的 Internal I/O 映射即可看到程序运行如下：

变量	映射	通道	地址	类型	当前值	预备值	单元	描述
		Module State	%IB0	BYTE	0			
		Module Err Num	%ID1	DWORD	0			
		outByte0	%QB0	BYTE	仅更新子元素			
Application.PLC_PR...		Bit0	%QX0.0	BOOL	TRUE			
Application.PLC_PR...		Bit1	%QX0.1	BOOL	FALSE			
Application.PLC_PR...		Bit2	%QX0.2	BOOL	FALSE			
Application.PLC_PR...		Bit3	%QX0.3	BOOL	FALSE			
Application.PLC_PR...		Bit4	%QX0.4	BOOL	FALSE			
Application.PLC_PR...		Bit5	%QX0.5	BOOL	FALSE			
Application.PLC_PR...		Bit6	%QX0.6	BOOL	FALSE			
Application.PLC_PR...		Bit7	%QX0.7	BOOL	FALSE			

变量	映射	通道	地址	类型	当前值	预备值	单元	描述
		Module State	%IB0	BYTE	0			
		Module Err Num	%ID1	DWORD	0			
		outByte0	%QB0	BYTE	仅更新子元素			
Application.PLC_PR...		Bit0	%QX0.0	BOOL	FALSE			
Application.PLC_PR...		Bit1	%QX0.1	BOOL	TRUE			
Application.PLC_PR...		Bit2	%QX0.2	BOOL	FALSE			
Application.PLC_PR...		Bit3	%QX0.3	BOOL	FALSE			
Application.PLC_PR...		Bit4	%QX0.4	BOOL	FALSE			
Application.PLC_PR...		Bit5	%QX0.5	BOOL	FALSE			
Application.PLC_PR...		Bit6	%QX0.6	BOOL	FALSE			
Application.PLC_PR...		Bit7	%QX0.7	BOOL	FALSE			

变量	映射	通道	地址	类型	当前值	预备值	单元	描述
		Module State	%IB0	BYTE	0			
		Module Err Num	%ID1	DWORD	0			
		outByte0	%QB0	BYTE	仅更新子元素			
Application.PLC_PR...		Bit0	%QX0.0	BOOL	FALSE			
Application.PLC_PR...		Bit1	%QX0.1	BOOL	FALSE			
Application.PLC_PR...		Bit2	%QX0.2	BOOL	TRUE			
Application.PLC_PR...		Bit3	%QX0.3	BOOL	FALSE			
Application.PLC_PR...		Bit4	%QX0.4	BOOL	FALSE			
Application.PLC_PR...		Bit5	%QX0.5	BOOL	FALSE			
Application.PLC_PR...		Bit6	%QX0.6	BOOL	FALSE			
Application.PLC_PR...		Bit7	%QX0.7	BOOL	FALSE			

表4-4 I/O 映射涉及的参数信息

名称	说明
Module State	模块状态 0x00: 没有错误 0x01: 模块忙 0x02: 超时未响应 0x03: 模块类型匹配 0x04: 模块版本不匹配 0x05: 软件错误 0x06: 等待标志超时 0x07: 总线应答错误 0x08: 总线 CRC 校验错误 0x0E: 地址没有配置上 0x10: 内存偏移量超范围 0x11: 模块没有准备好 0x12: 模块组态错误 0x13: 模块不支持本条指令

	0x15: 模块内部诊断 0x16: 模块没有电源 0x17: 校验出错
outByte0	数字量输出通道值 outByte0 代表模块的输出通道字节 0
Module Err Num	模块错误次数

### Internal 参数

Internal 参数包含了模块启动时所需的一些基本配置参数，设备启动时，将以 Internal 参数中设置的值执行操作。

参数	类型	值	默认值	单元	描述
Vendor	STRING	'Systeme Electric'	'Systeme Electric'		Vendor of the device
ModuleName	STRING	'DIT08 8DI'	'DIT08 8DI'		Module name of the device
ModuleSignature	DWORD	16#120000	16#120000		Module Signature of the device
channel 0-3 filter	BYTE	6	6		filter time,0:0.2ms, 1:0.4ms, 2:0.8ms, 3:1.6ms, 5:3.2ms, 6:6.4ms, 7:12.8ms
channel 4-7 filter	BYTE	6	6		filter time,0:0.2ms, 1:0.4ms, 2:0.8ms, 3:1.6ms, 5:3.2ms, 6:6.4ms, 7:12.8ms

表4-5 Internal 参数信息

名称	说明																
vender	设备供应商																
ModuleName	设备模块名称																
Modulesignature	设备模块签名																
channel 0~3 filter channel 4~7 filter	数字量输入滤波配置 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>bit 3</th> <th>bit 2</th> <th>bit 1</th> <th>bit 0</th> </tr> </thead> <tbody> <tr> <td>保留</td> <td colspan="3">通道 0-3 滤波时间</td> </tr> <tr> <th>bit 7</th> <th>bit 6</th> <th>bit 5</th> <th>bit 4</th> </tr> <tr> <td>保留</td> <td colspan="3">通道 4-7 滤波时间</td> </tr> </tbody> </table> 滤波时间: 0: 0.20ms 1: 0.40ms 2: 0.80ms 3: 1.60ms 5: 3.20ms 6: 6.40ms (默认值) 7: 12.8ms	bit 3	bit 2	bit 1	bit 0	保留	通道 0-3 滤波时间			bit 7	bit 6	bit 5	bit 4	保留	通道 4-7 滤波时间		
bit 3	bit 2	bit 1	bit 0														
保留	通道 0-3 滤波时间																
bit 7	bit 6	bit 5	bit 4														
保留	通道 4-7 滤波时间																

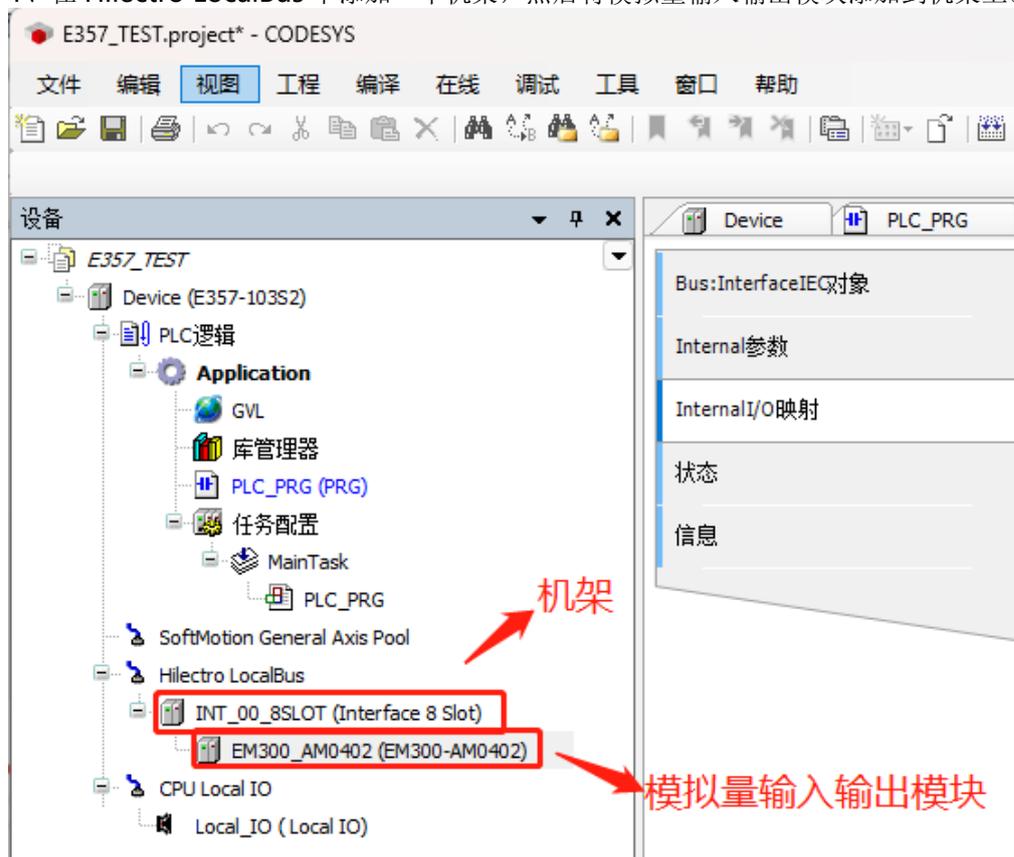
## 4.3 模拟量模块的使用

硬件：E357，模拟量输入输出模块（电流电压输出）。

软件：CODESYS V3.5 SP19 Path5

工程组态：

1、在 Hillectro LocalBus 下添加一个机架，然后将模拟量输入输出模块添加到机架上。



2、双击模拟量输入模块可进入模块配置界面，在 **Internal** 参数中可以查看模块名称，模块控制字等信息。**Internal** 参数包含了模块启动时所需的一些基本配置参数，设备启动时，将以 **Internal** 参数中设置的值执行操作。

下面是模拟量输入输出模块的配置界面，在 **Internal** 参数中需要设置控制字的值，通过输入控制字设置模拟量输入类型、输入量程、采样周期；通过输出控制字设置输出量程。

参数	类型	值	默认值	单元	描述
Vendor	STRING	'Systeme El...	'Systeme Electric'		Vendor of the device
ModuleName	STRING	'AMS-06 4...	'AMS-06 4AI2AQ 12...		Module name of the device
ModuleSignature	DWORD	16#101100	16#101100		Module Signature of the device
Input 0-1 Ctrlword	WORD	16#4041	16#4041		bit0-4 input range 0:0~5V, 1:0~10V, 2:0~20mA, 3:4~20mA, 0x10: +2.5V, 0x11: +5
Input 2-3 Ctrlword	WORD	16#4041	16#4041		bit0-4 input range 0:0~5V, 1:0~10V, 2:0~20mA, 3:4~20mA, 0x10: +2.5V, 0x11: +5
Output 0-1 Ctrlword	WORD	16#0000	16#0000		bit0-4, Output range 16#00:-10~10V, 16#11:0~20mA, 16#12:4~20mA

表4-6 模拟量输入控制字（默认为 16#4041，表示输入为电压信号，其量程为 0~10V，采样周期为 50Hz；通过 bit4 来区分单极性和双极性，bit4 为 1 表示双极性；为 0 表示单极性）

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
设置采样周期			设置输入类型和输入量程					
模块类型	更新频率 (采样周期)	采样周期编码 (BIT7~5)						
AI 模块 4 通道	200·Hz	000	电压	0~5V		00000		
	100·Hz	001		0~10V (默认值)		00001		
	50·Hz (默认值)	010		±2.5V		10000		
	20·Hz·····	011		±5V		10001		
	10·Hz	100		电流	0~20mA (默认值)		00010	
AI 模块 8 通道	50·Hz	000	4~20mA		00011			
	20·Hz	001						
	10·Hz (默认值)	010						
	5·Hz	011						
	2·Hz···	100						
bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
保留	必须是 1	保留	输入量程为 4~20mA 时，设置断线方向。 0：正方向（默认） 1：负方向 其他量程时该保留	保留	保留	保留		
<b>备注：</b> <ul style="list-style-type: none"> <li>• 使用的通道出现断线时，通道值显示一个固定的值（断线值），而且 SF 红灯闪烁报警；</li> <li>• 两个断线方向，分别为一正一负，选择 bit12 的值，当模拟量通道断线后，会在对应的通道显示其断线值，bit12=0 为正断线值，bit12=1 为负断线值。具体的断线值可以参考章节 2.8 <a href="#">模拟量模块</a> 查看对应模块的断线值。</li> <li>• 未使用通道上电显示值为 0（备注：若通道有使用，但有断线情况或输入超量程时，通道上电初始为 0，待接线恢复正常且输入在正常范围以内时，通道值恢复正常）。</li> </ul>								

表4-7 模拟量输出控制字（默认为 16#0000，表示输出量程为-10V~10V）

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
保留			设置输出量程 16#00: -10V~10V 16#11: 0~20mA 16#12: 4~20mA				
bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
保留							

**备注>** 模拟量模块有输入模块，输出模块，输入输出模块，模块控制字不是一致的，具体的控制字按照上述步骤在对应模块的 **Internal** 参数查看。

### 3、Internal I/O 地址

模拟量扩展模块添加到机架上以后，在 **Internal I/O** 地址中，系统会自动分配模块的通道地址。在 **Internal I/O** 地址中还可以查看模块的状态，以及模块错误次数。

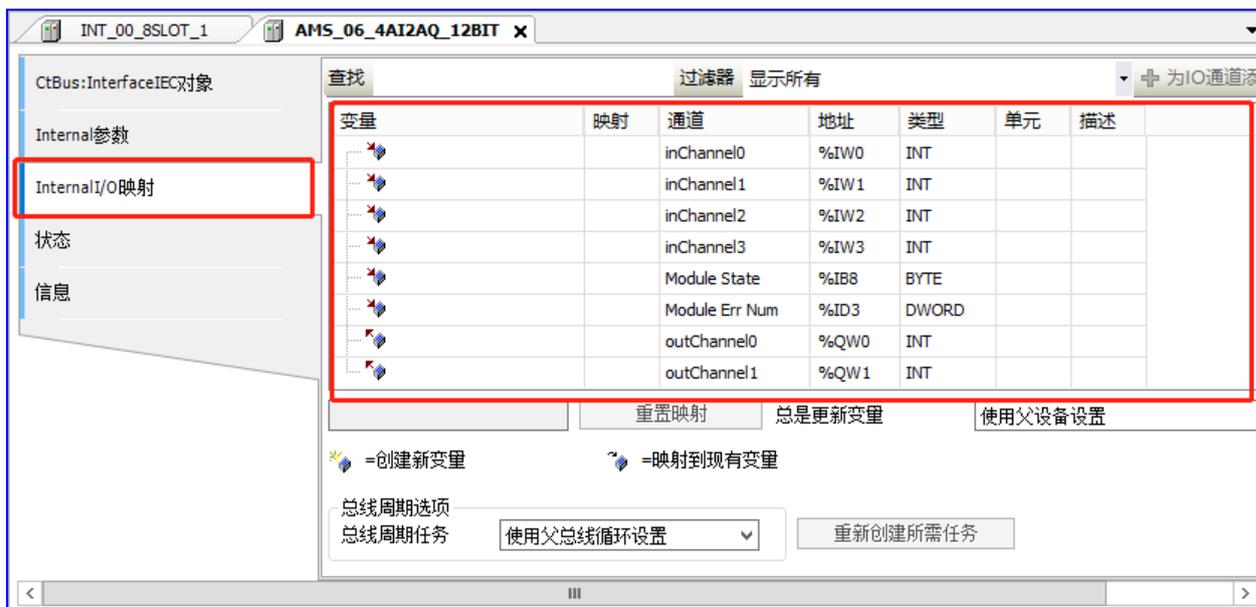
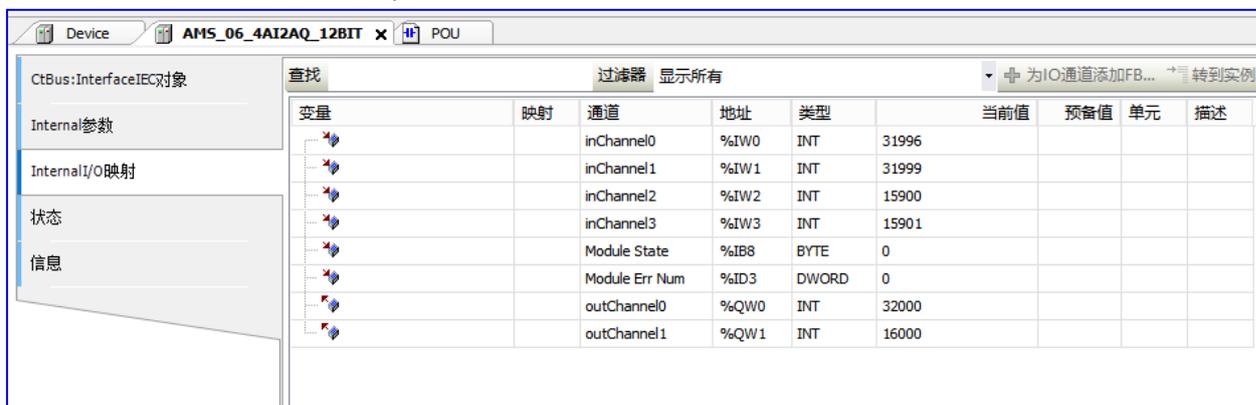


表4-8 Internal I/O 信息

名称	说明																																								
inChannel	工程运行后，将会在通道显示检测到输入值，断线的时候则显示断线值。																																								
Module State	<p>模块状态</p> <table border="1"> <thead> <tr> <th>地址</th> <th>说明</th> <th>地址</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>没有错误</td> <td>0x0E</td> <td>地址没有配置上</td> </tr> <tr> <td>0x01</td> <td>模块忙</td> <td>0x10</td> <td>内存偏移量超范围</td> </tr> <tr> <td>0x02</td> <td>超时未响应</td> <td>0x11</td> <td>模块没有准备好</td> </tr> <tr> <td>0x03</td> <td>模块类型匹配</td> <td>0x12</td> <td>模块组态错误</td> </tr> <tr> <td>0x04</td> <td>模块版本不匹配</td> <td>0x13</td> <td>模块不支持本条指令</td> </tr> <tr> <td>0x05</td> <td>软件错误</td> <td>0x15</td> <td>模块内部诊断</td> </tr> <tr> <td>0x06</td> <td>等待标志超时</td> <td>0x16</td> <td>模块没有电源</td> </tr> <tr> <td>0x07</td> <td>总线应答错误</td> <td>0x17</td> <td>校验出错</td> </tr> <tr> <td>0x08</td> <td>总线 CRC 校验错误</td> <td>0xF</td> <td>模拟量输入控制字写错（仅限模拟量输入模块）</td> </tr> </tbody> </table>	地址	说明	地址	说明	0x00	没有错误	0x0E	地址没有配置上	0x01	模块忙	0x10	内存偏移量超范围	0x02	超时未响应	0x11	模块没有准备好	0x03	模块类型匹配	0x12	模块组态错误	0x04	模块版本不匹配	0x13	模块不支持本条指令	0x05	软件错误	0x15	模块内部诊断	0x06	等待标志超时	0x16	模块没有电源	0x07	总线应答错误	0x17	校验出错	0x08	总线 CRC 校验错误	0xF	模拟量输入控制字写错（仅限模拟量输入模块）
地址	说明	地址	说明																																						
0x00	没有错误	0x0E	地址没有配置上																																						
0x01	模块忙	0x10	内存偏移量超范围																																						
0x02	超时未响应	0x11	模块没有准备好																																						
0x03	模块类型匹配	0x12	模块组态错误																																						
0x04	模块版本不匹配	0x13	模块不支持本条指令																																						
0x05	软件错误	0x15	模块内部诊断																																						
0x06	等待标志超时	0x16	模块没有电源																																						
0x07	总线应答错误	0x17	校验出错																																						
0x08	总线 CRC 校验错误	0xF	模拟量输入控制字写错（仅限模拟量输入模块）																																						
Module Err Num	模块错误次数																																								
outChannel	工程运行后，通过给该通道输出值来控制输出																																								

4、将工程下载到 PLC 后可以在 I/O 映射中看到输入输出通道的值。



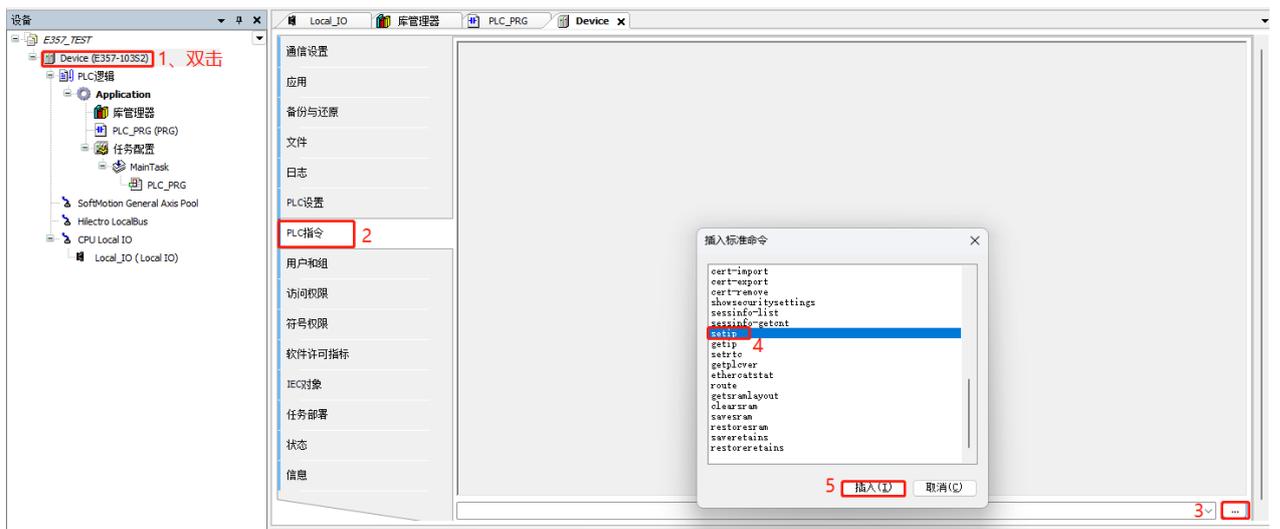
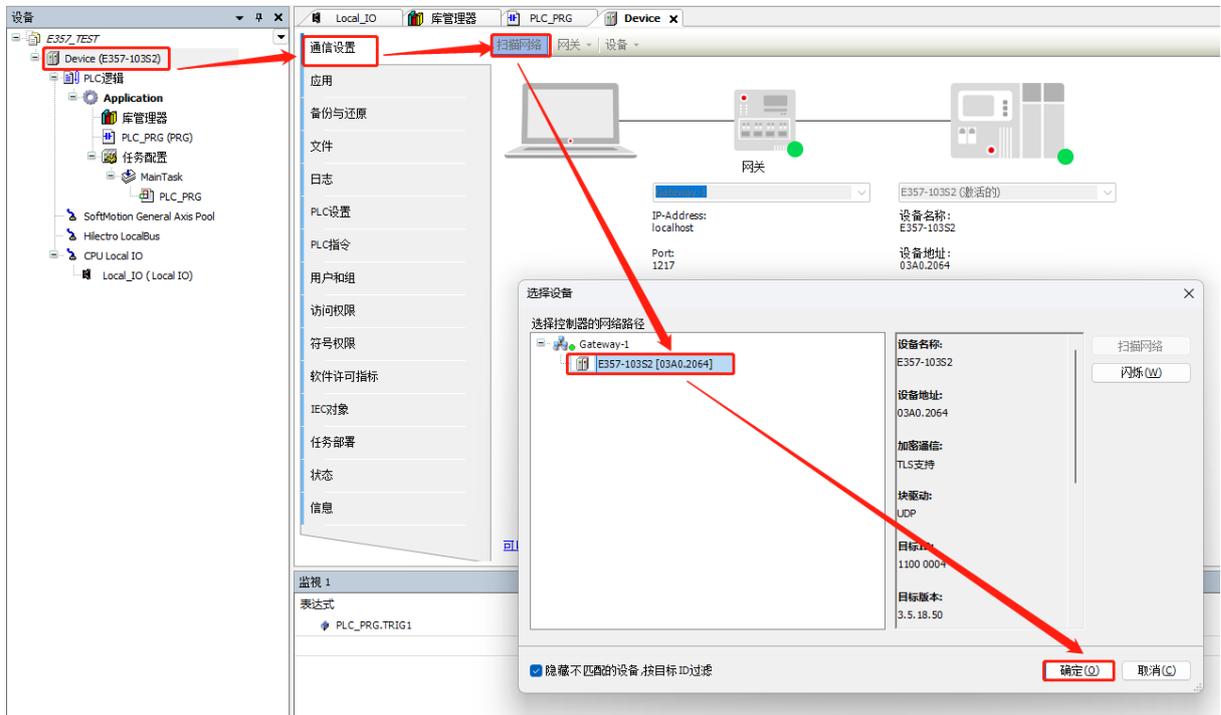
# 附录

# 5

5.1	修改 PLC 的 IP 及网关
5.2	读取 PLC 版本
5.3	U 盘固件更新
5.4	掉电数据保存
5.5	写入 PLC 时间
5.6	采样跟踪的使用 (Trace)
5.7	源代码上下载
5.8	同一网络下存在多个 PLC 时如何更改 PLC 名称
5.9	如何将文件写入 U 盘
5.10	添加库文件到工程中
5.11	复位功能
5.12	订货信息

## 5.1 修改 PLC 的 IP 及网关

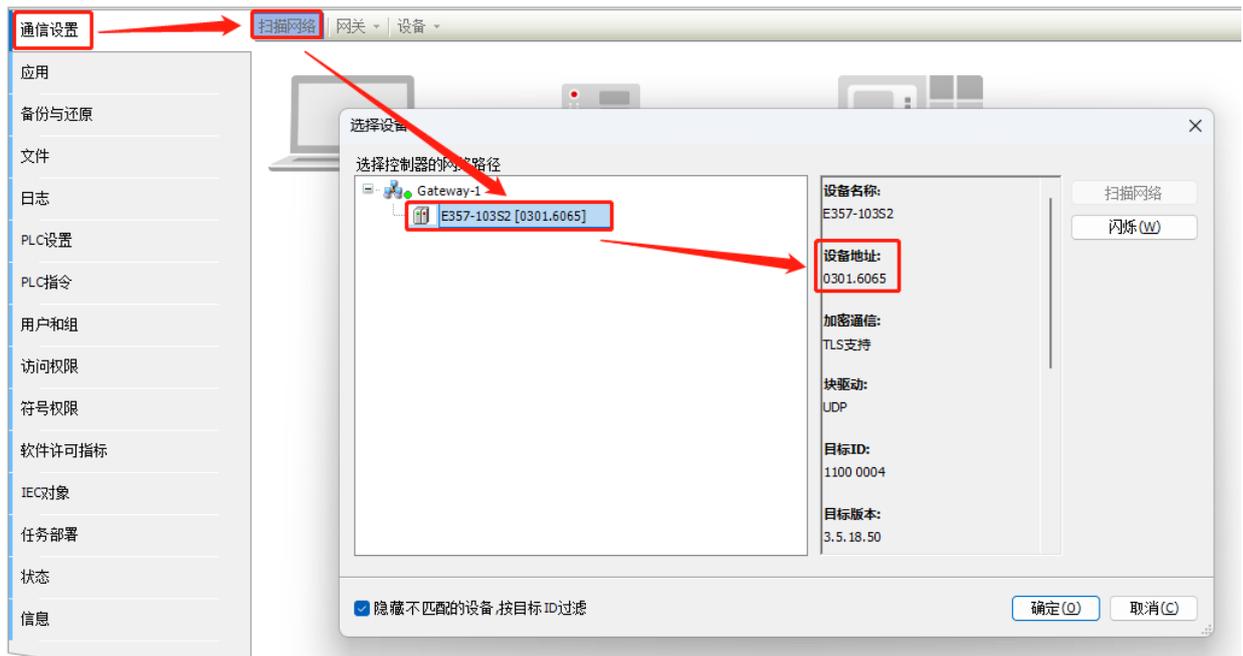
IP 地址需通信成功后，在 PLC 指令中修改，具体操作如下：





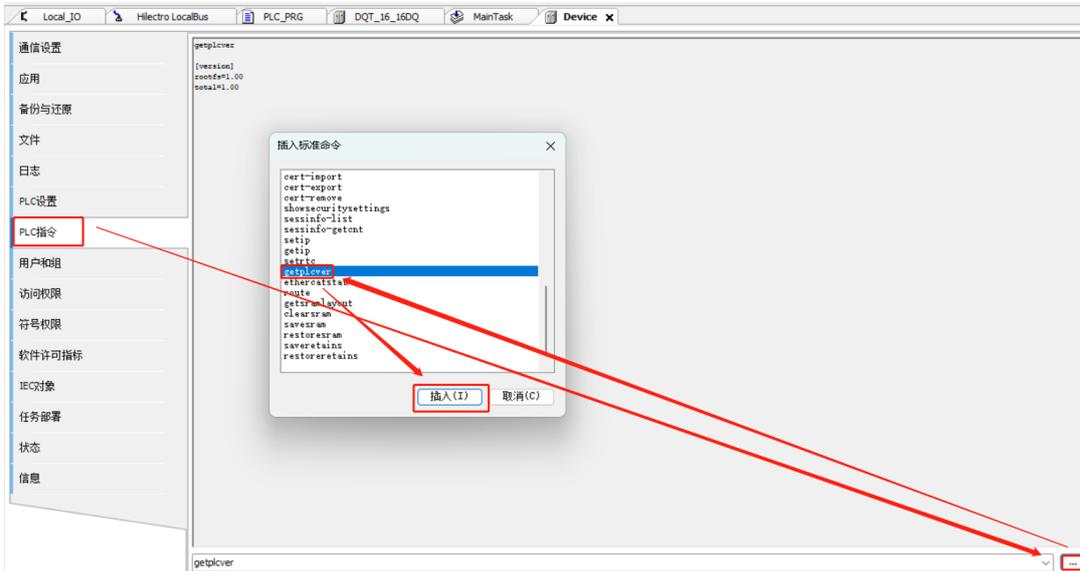
也可以忽略前面的 3, 4, 5 步, 直接在输入框里输入 setip eth0 ip:192.168.0.X

修改 IP 后, 可以在通信设置中看到修改成功的 IP 地址。



## 5.2 读取 PLC 版本

1、PLC 在线情况下，双击 Device，选择 PLC 指令，点击右下角...，在出现的窗口里选择 `getplcver`，或者直接输入 `getplcver`，执行。在弹出窗口中出现的数字即为版本号，如下图，PLC 固件版本为 1.00。



```
getplcver

[version]
rootfs=1.00
total=1.00
```

## 5.3 U 盘固件更新

CPU 断电，开关拨到停机状态后，将 U 盘插入到 PLC 的 USB 口后再上电，几秒后除了电源灯外，其他灯常亮代表加载过程。加载完成后除了电源灯外其他灯闪烁，之后将 CPU 断电重启。

## 5.4 掉电数据保存

掉电保持变量，就是具有 `persistent(persistent remain)` 属性的变量，该变量除了掉电保持和热复位之外，还具有冷复位保留值和程序下载保留值的特性，掉电保持变量只有在初始复位才被初始化。

使用掉电保持变量的目的：当设备掉电，冷热复位，下载程序，在线修改程序时，某些与实际应用关联的参

数能够保持当前值，不会被初始化到原始状态。

表5-1 掉电保持变量掉电一览表

在线命令	VAR	VAR RETAIN	VAR PERSISTENT	VAR PERSISTENT RETAIN
初始值复位	初始化	初始化	初始化	初始化
下载	初始化	初始化	初始化	保持
冷复位	初始化	初始化	初始化	初始化
热复位	初始化	保持	初始化	保持
掉电	初始化	保持	初始化	保持
在线修改	保持	保持	保持	保持

**注意事项：**

I/O 区不能设为掉电保持变量

功能模块中的局部变量定义为掉电保持变量，功能块的整个实例占用保留区内存，但只有定义的掉电保持变量具有掉电保持功能。

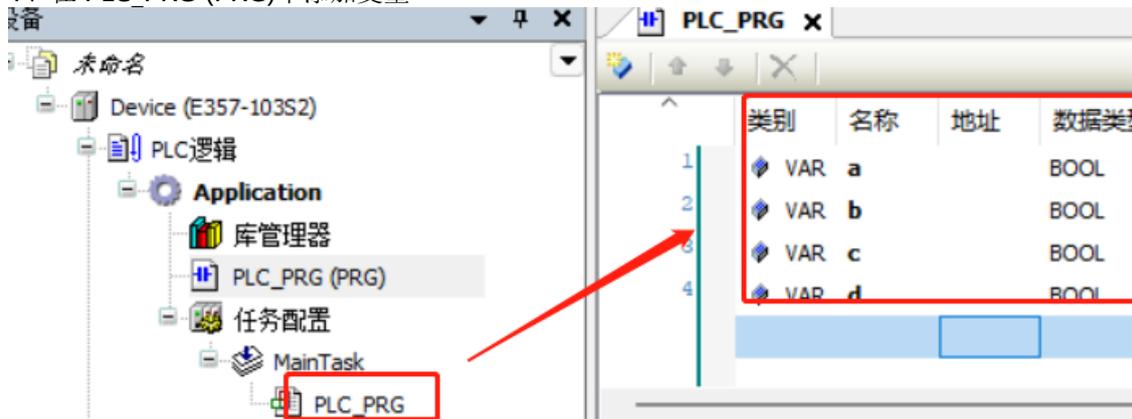
如果程序中的局部变量定义掉电保持变量，该变量也占用保留区内存。

**掉电保持变量使用示例：**

通过创建两个断电保持变量和两个常规变量来解释断电保持特性

具体操作如下：

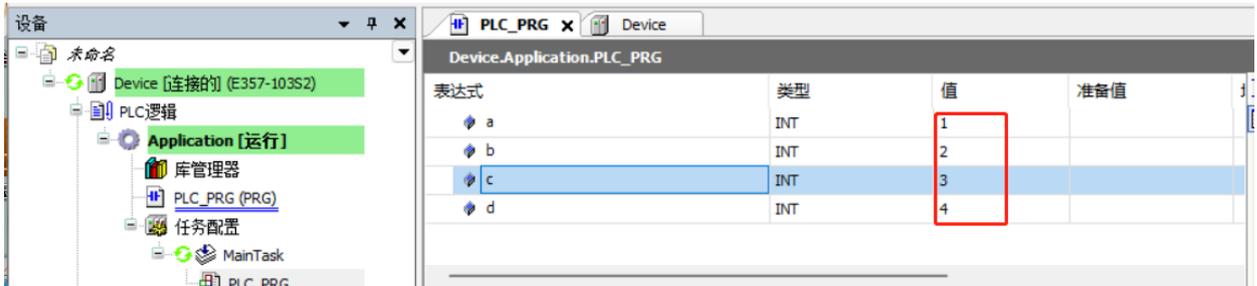
**1、在 PLC\_PRG (PRG)中添加变量**



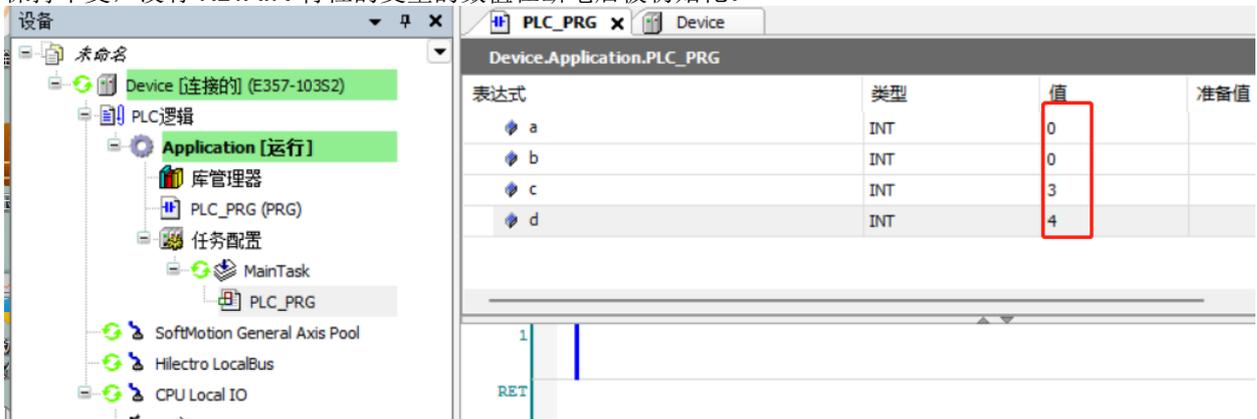
**2、设置变量类别如下**



**3、登录 PLC，然后为四个变量赋值**



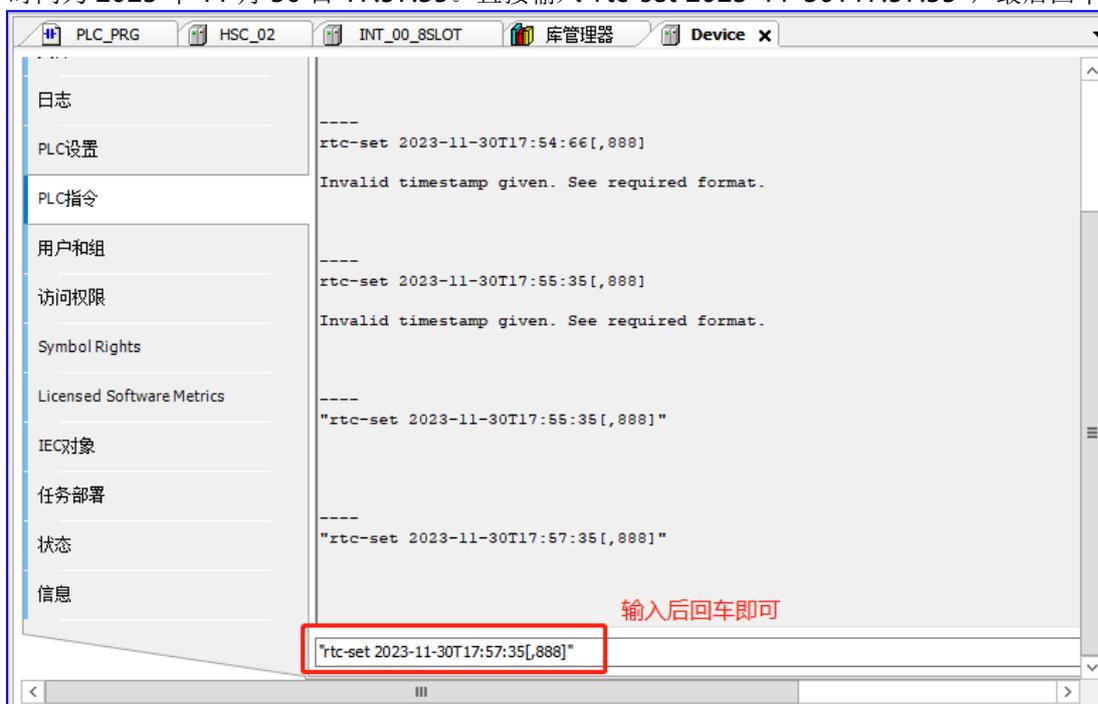
4、退出登录，关闭 PLC 电源并重新启动，然后观察四个变量的值。具有 RETAIN 属性的变量的值在断电后保持不变，没有 RETAIN 特性的变量的数值在断电后被初始化。



## 5.5 写入 PLC 时间

给 PLC 写入时间的操作如下：

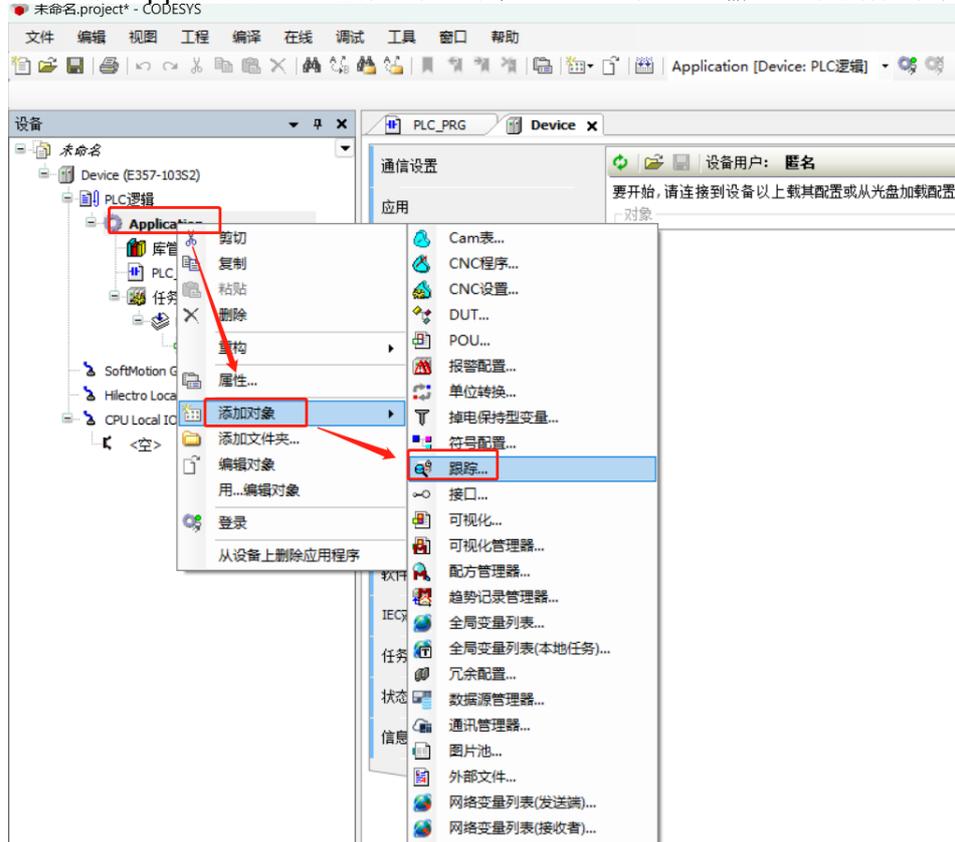
电脑与 PLC 通信，然后双击 device，进入 PLC 指令，输入"rtc-set YYYY-MM-DDThh:mm:ss"。比如设置时间为 2023 年 11 月 30 日 17:57:35。直接输入"rtc-set 2023-11-30T17:57:35"，最后回车即可。



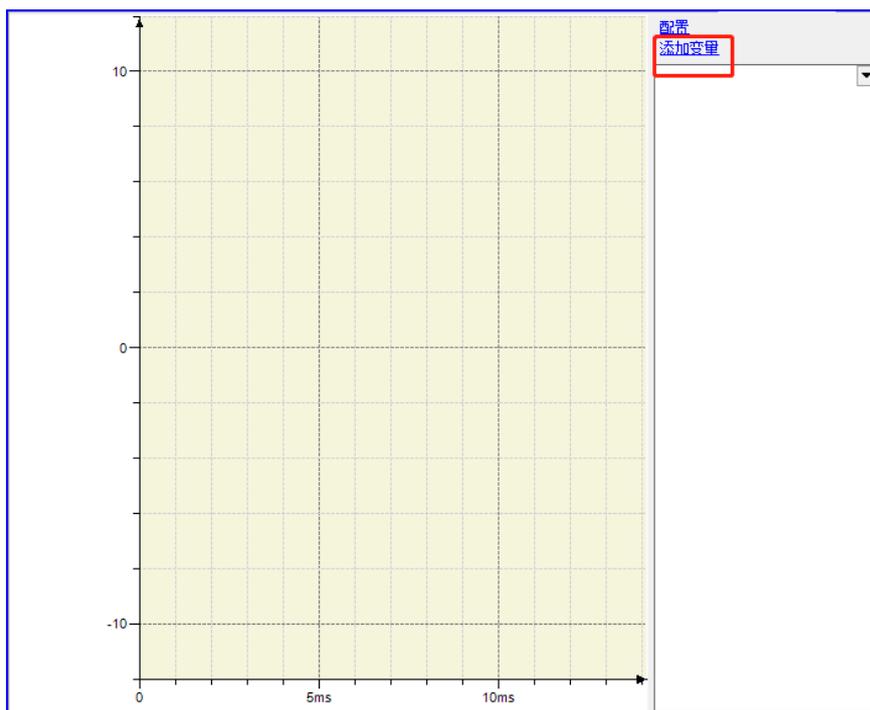
## 5.6 采样跟踪的使用(Trace)

采样跟踪是 CODESYS 提供的图形化数据监控软件，类似示波器。在程序运行过程中产生的数据都是转瞬即逝的，无法根据实时产生的数据进行分析，通过采样跟踪，可以把程序执行过程中产生的过程数据全部记录下来，例如运动控制过程中的电机当前位置、速度、加速度等。通过对采集到的数据分析，就可以清楚的观察到系统运行的整个过程。采样跟踪可以采集 PLC 运行过程中产生的数据波形，用户可以在同一个 PLC 应用下创建多个跟踪配置文件，如有需要用户也可以设置采样触发条件、采样周期以及对采样数据进行保存。

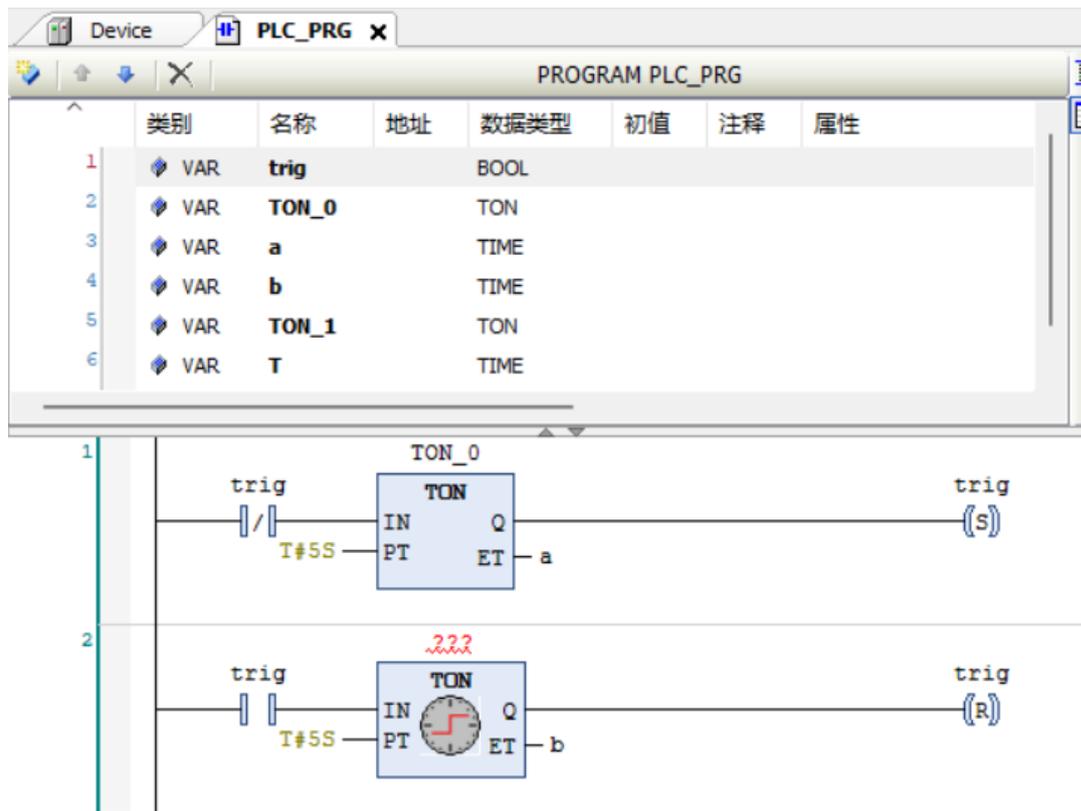
1、右击“Application”，选择“添加对象”→“跟踪...”，输入跟踪名称后确认添加。



2、打开新建的跟踪后，选择界面右侧“添加变量”，进行跟踪变量的添加

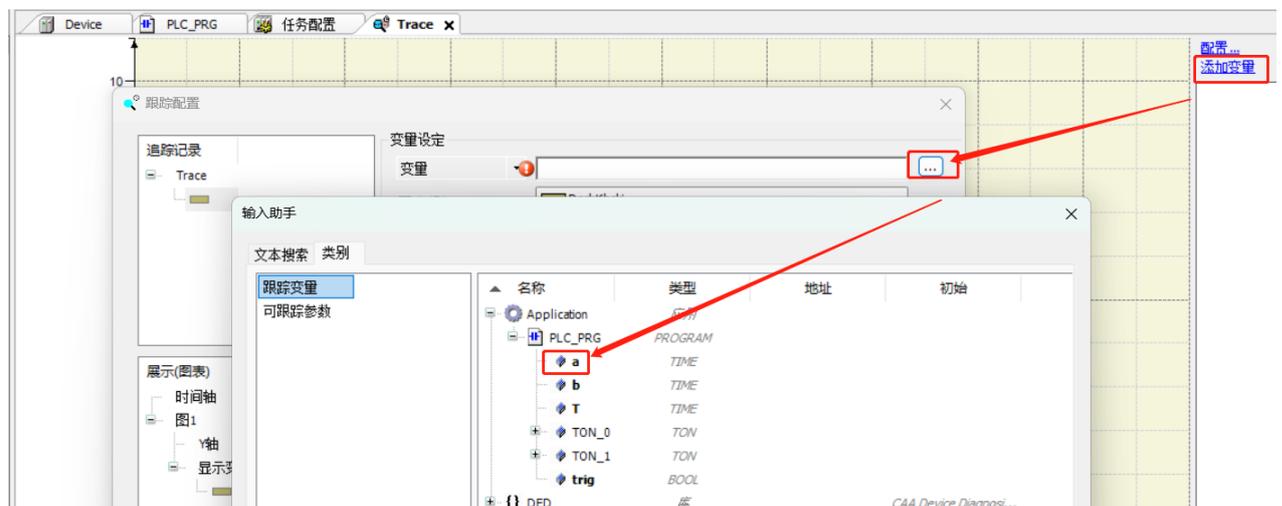


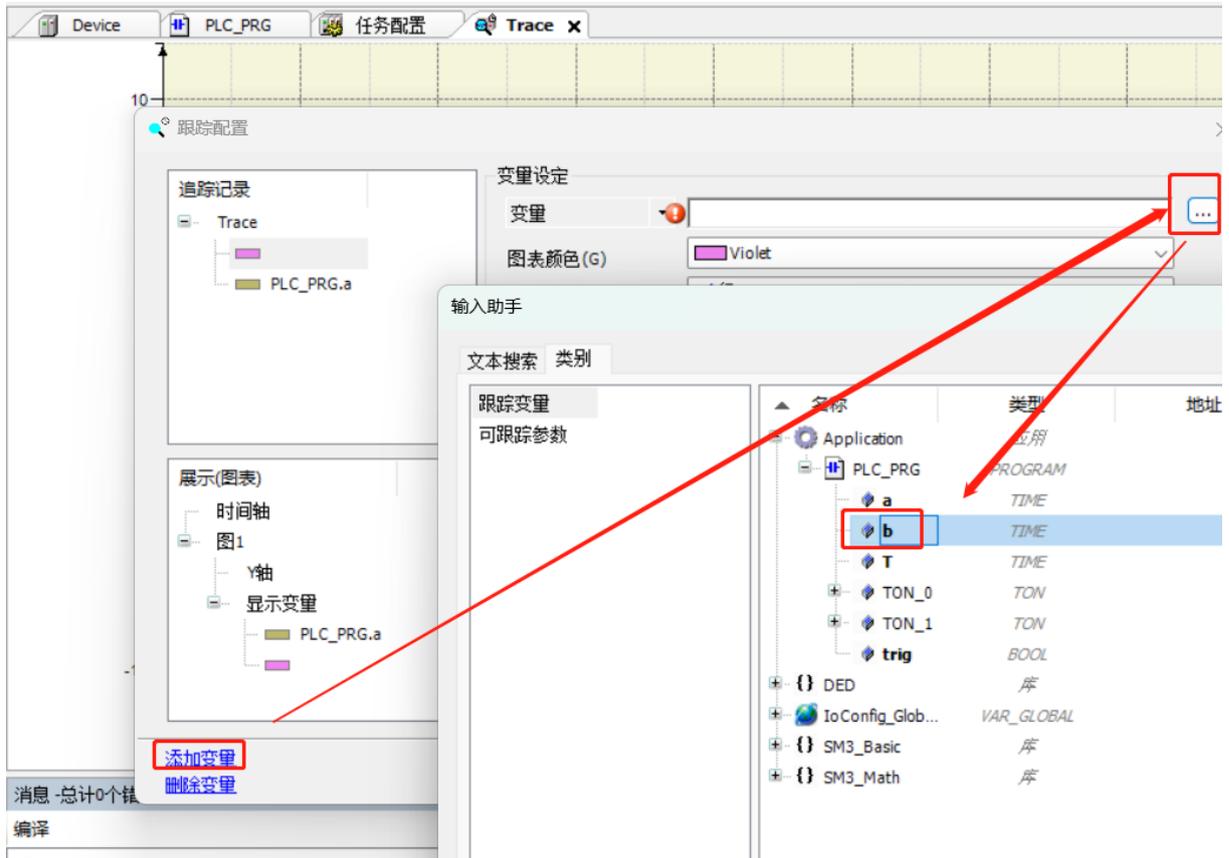
以下面的变量 var1 和 var2 为例，监测它们的变化趋势。



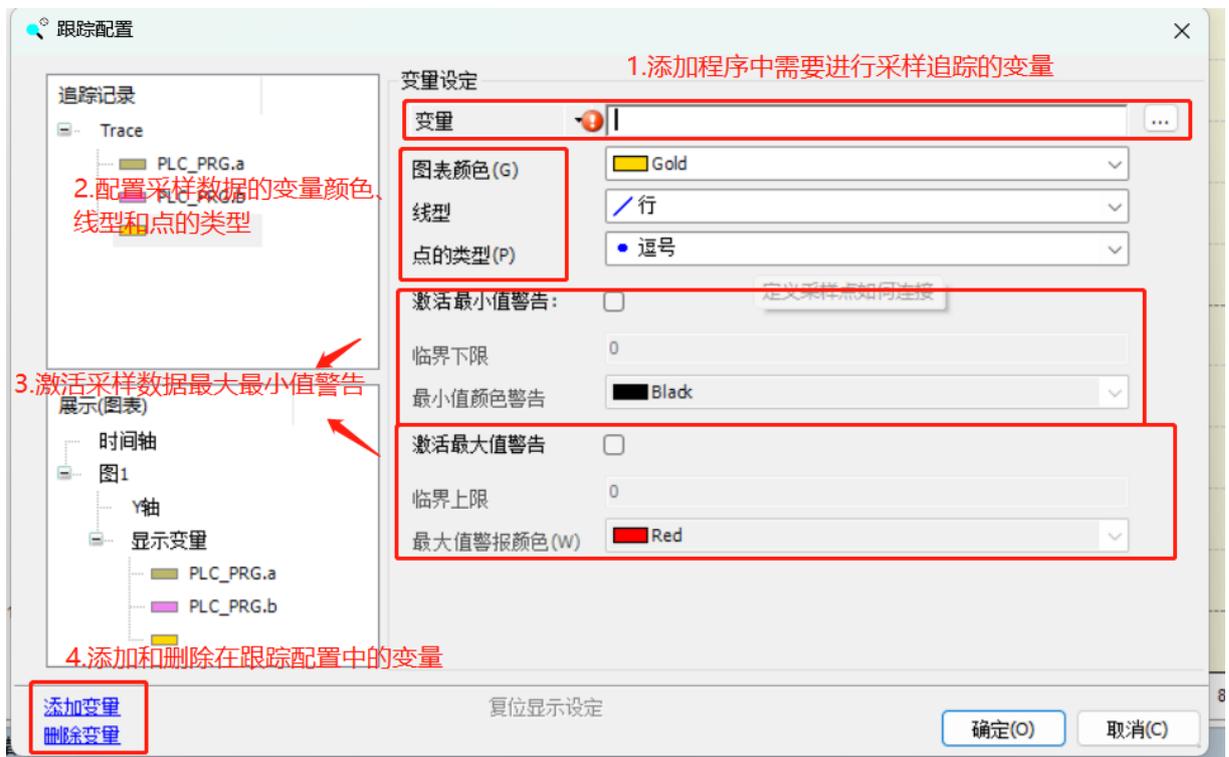
3、登录 PLC，运行并监控上述程序

4、将监控变量 a 和 b 添加到跟踪中

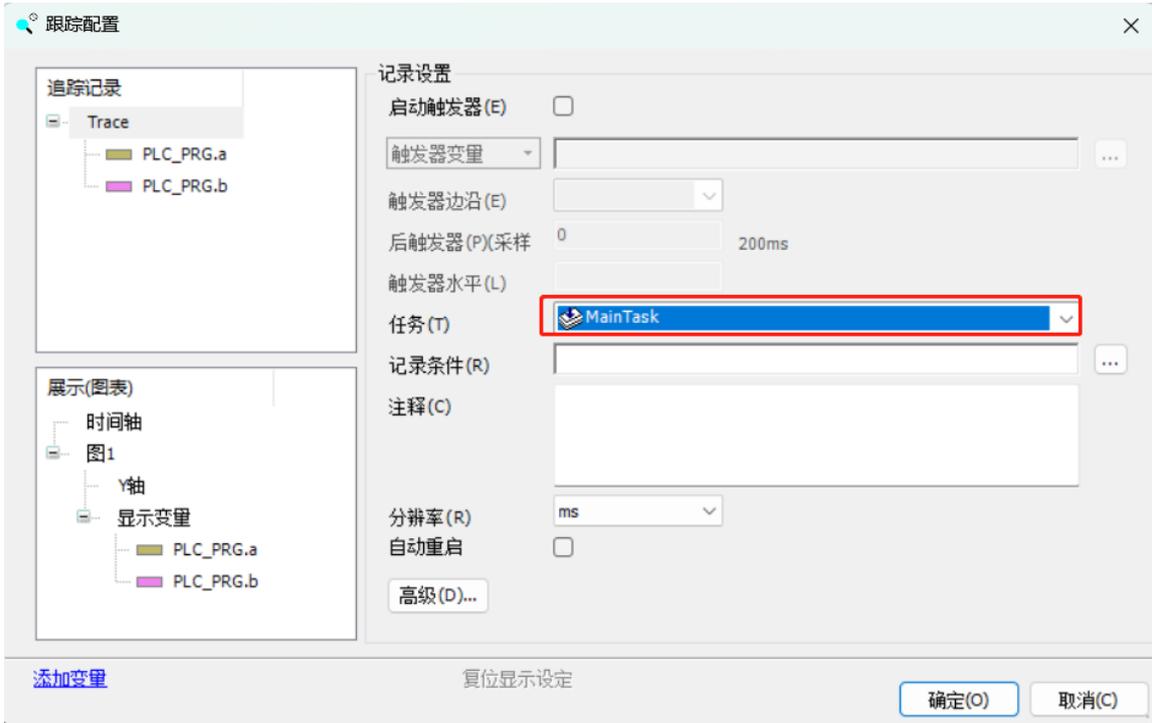




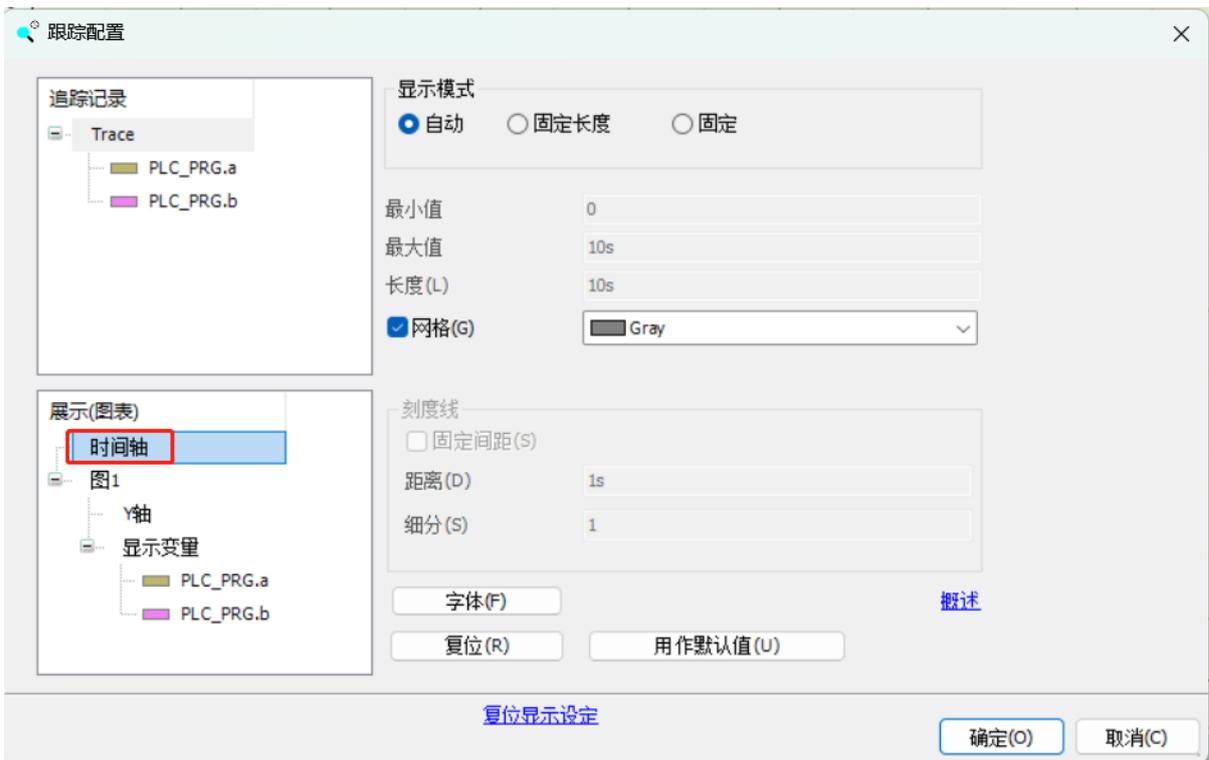
5、变量配置



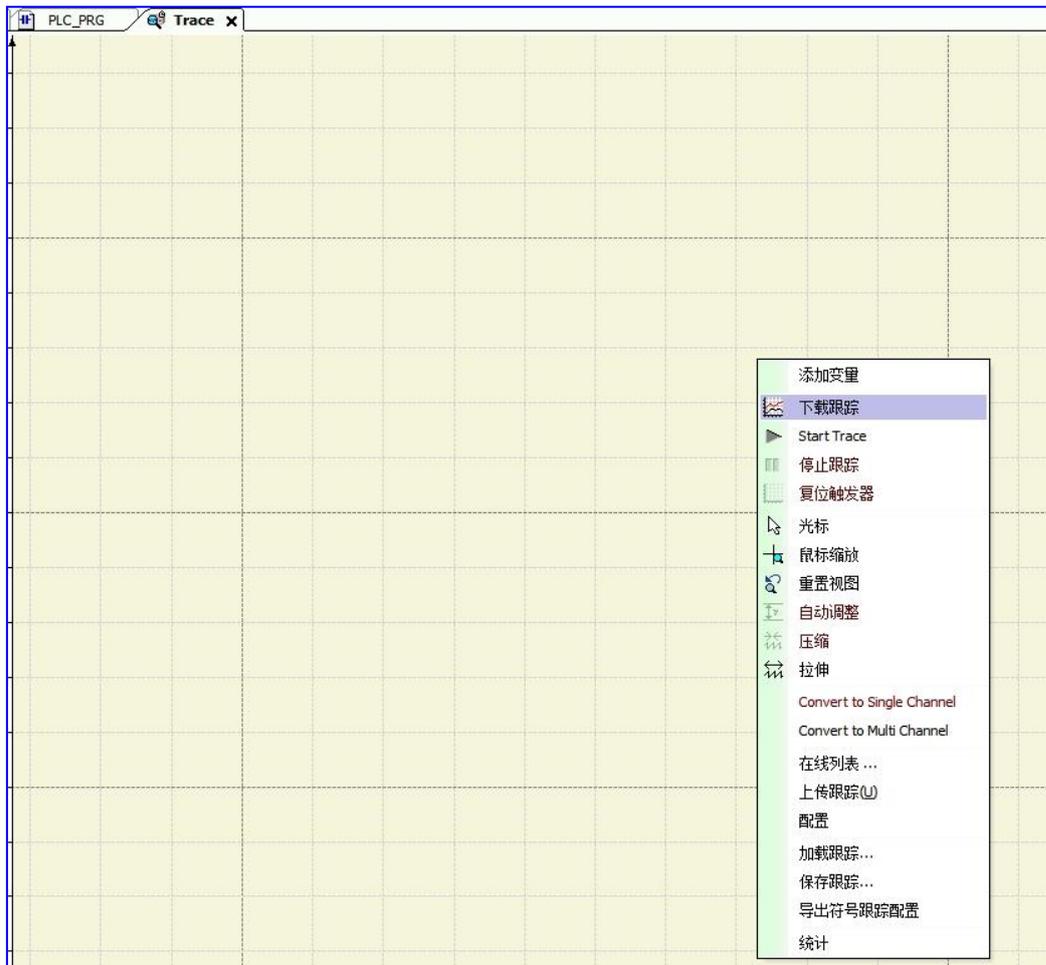
6、在配置中选择 main task



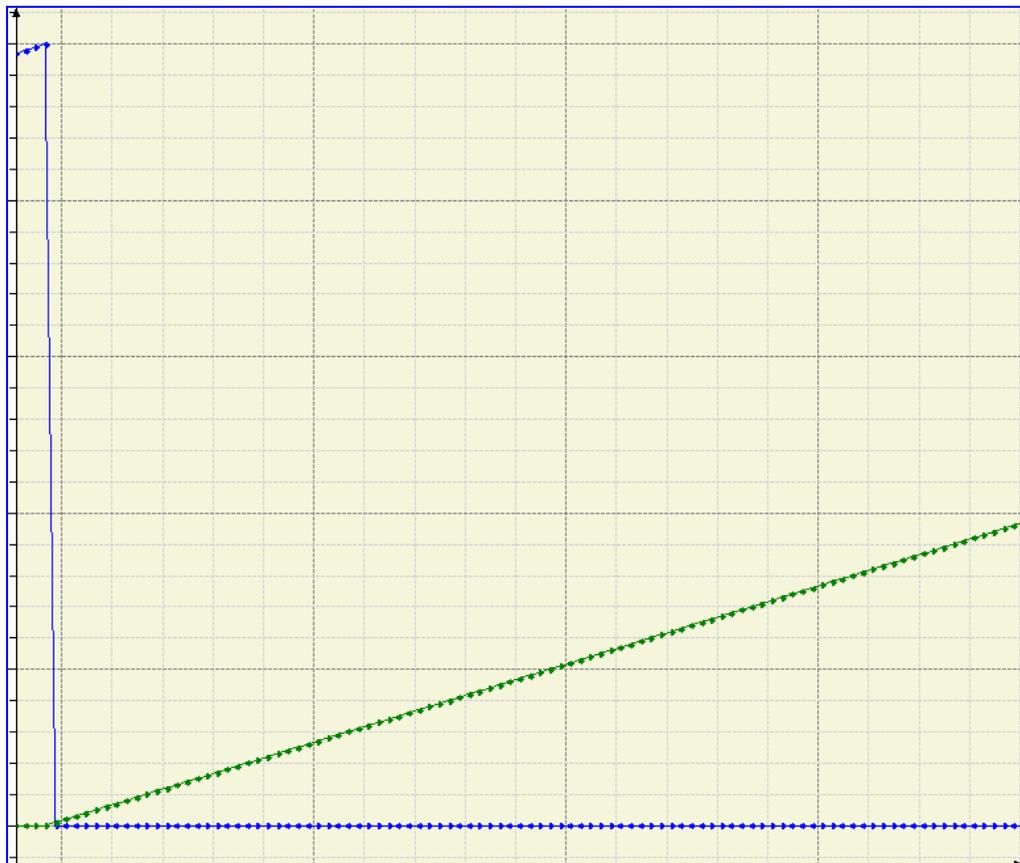
7、在“配置”中还可以对追踪的背景和坐标轴进行设置，在“TimeAxis”时间轴中可以配置显示模式为自动、固定长度、固定，设置成自动后，采样过程中，时间轴会自动调整；选择固定长度，则由用户自定义显示时长，采样窗口将只显示长度内的采样结果；最后，设置成固定，将会由用户指定显示采样过程中某一段的曲线。在同一页面中还可以对字体、网格等进行配置，选择Y轴之后可以观察到类似的页面，按照需求进行配置即可。



8、右键单击下面的视图并下载跟踪



9、跟踪变量视图如下：

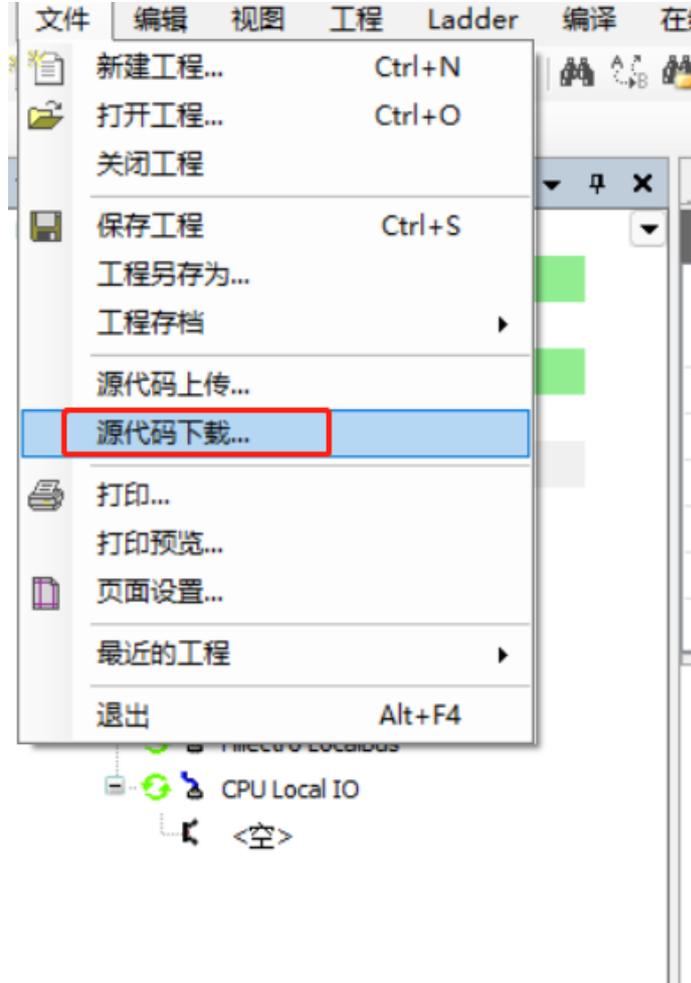


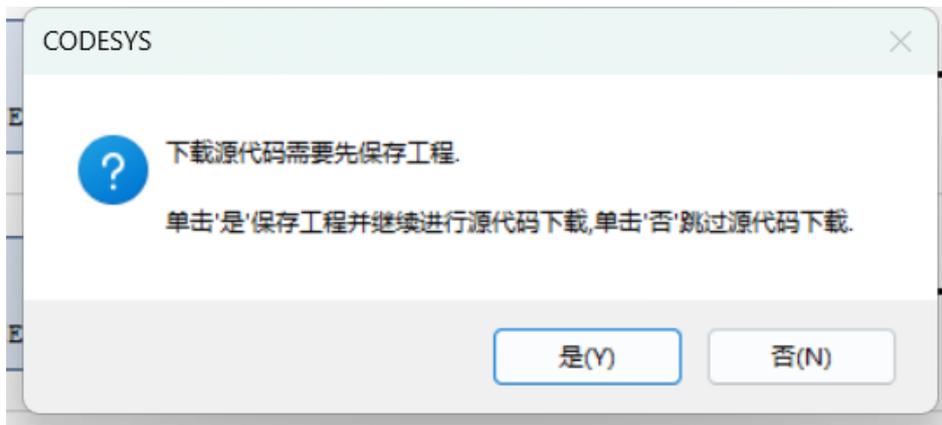
## 5.7 源程序上下载

程序下载与源程序下载不同。登录并下载程序时，下载到 PLC 的文件是执行文件，不能以这种方式上传程序。程序只能在下载源程序后才能上传。以下介绍了程序下载的操作：

### 源程序下载

在下载源程序之前，请确保其编译正确，登录 PLC 后即可进行源程序下载。

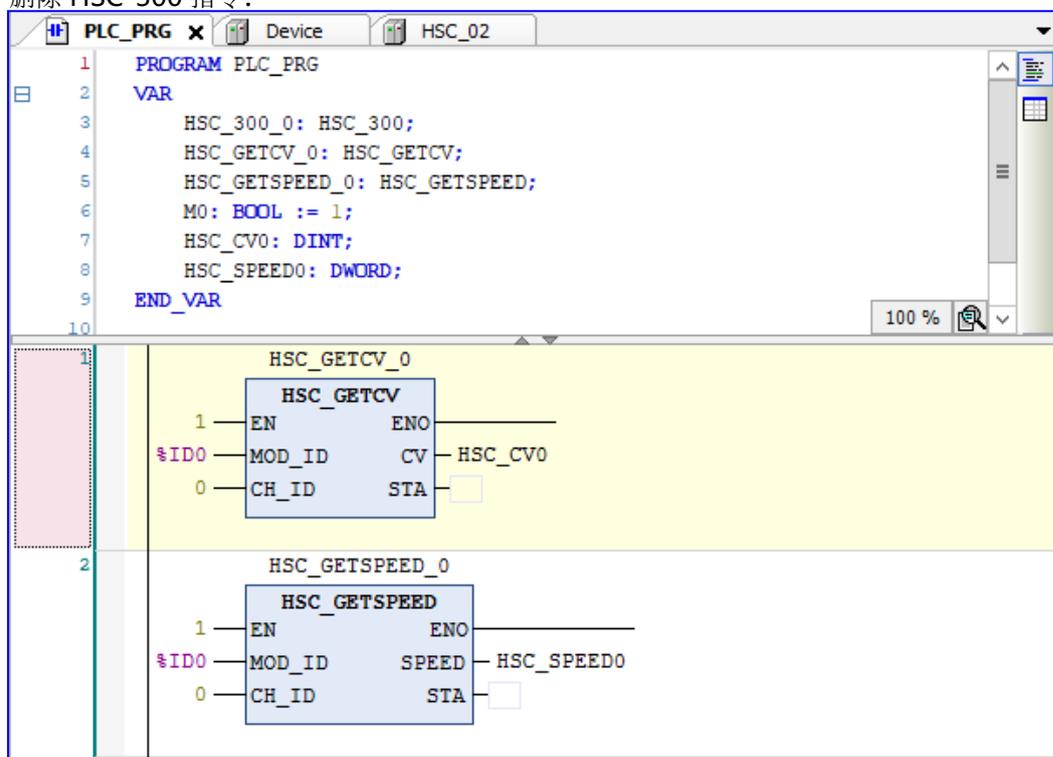




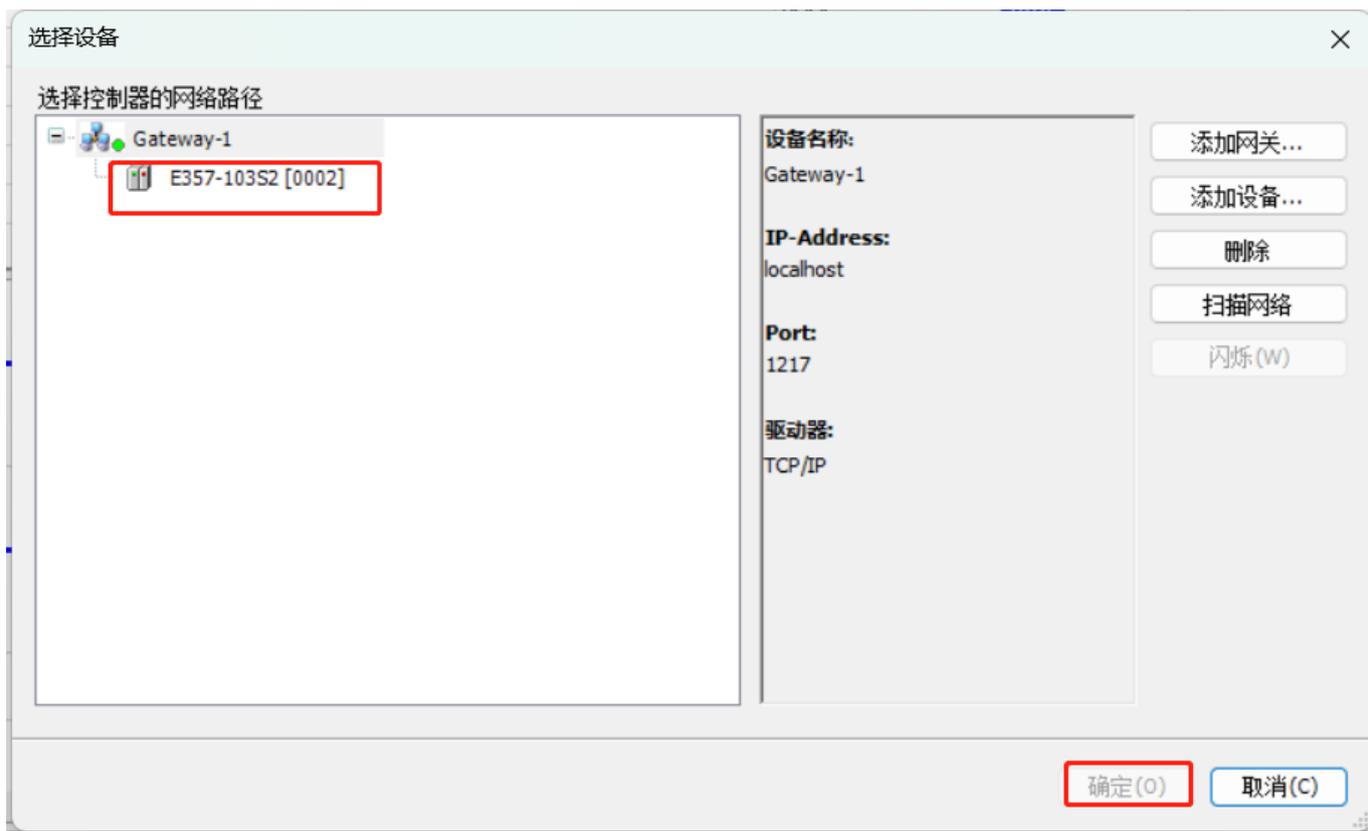
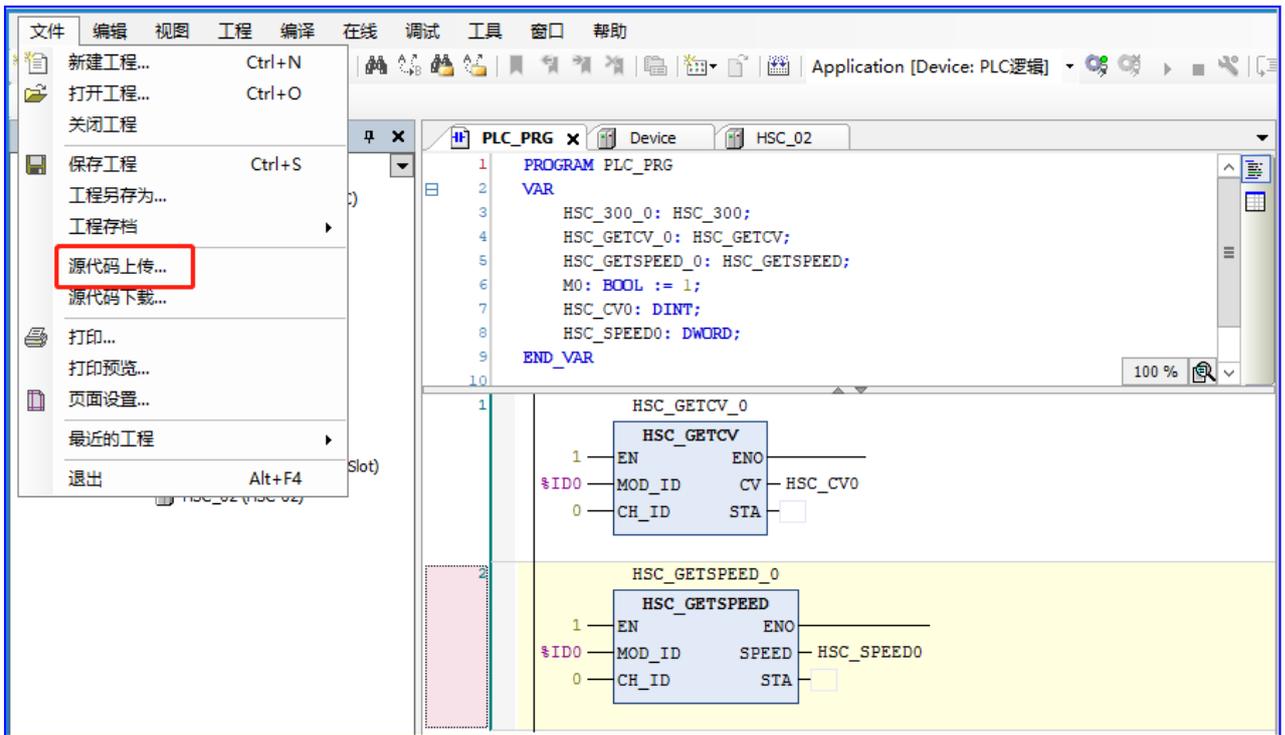
### 源程序上传

如果用户在调试后需要从 PLC 上传程序，请登录 PLC 并选择源程序上传，然后选择要上传源程序的设备。  
为了验证程序是否上传成功，在上传源程序之前，我们在登录 PLC 之前删除 HSC\_300 指令，然后最后上传程序，看看它是否与之前下载的源程序相同。

删除 HSC 300 指令：



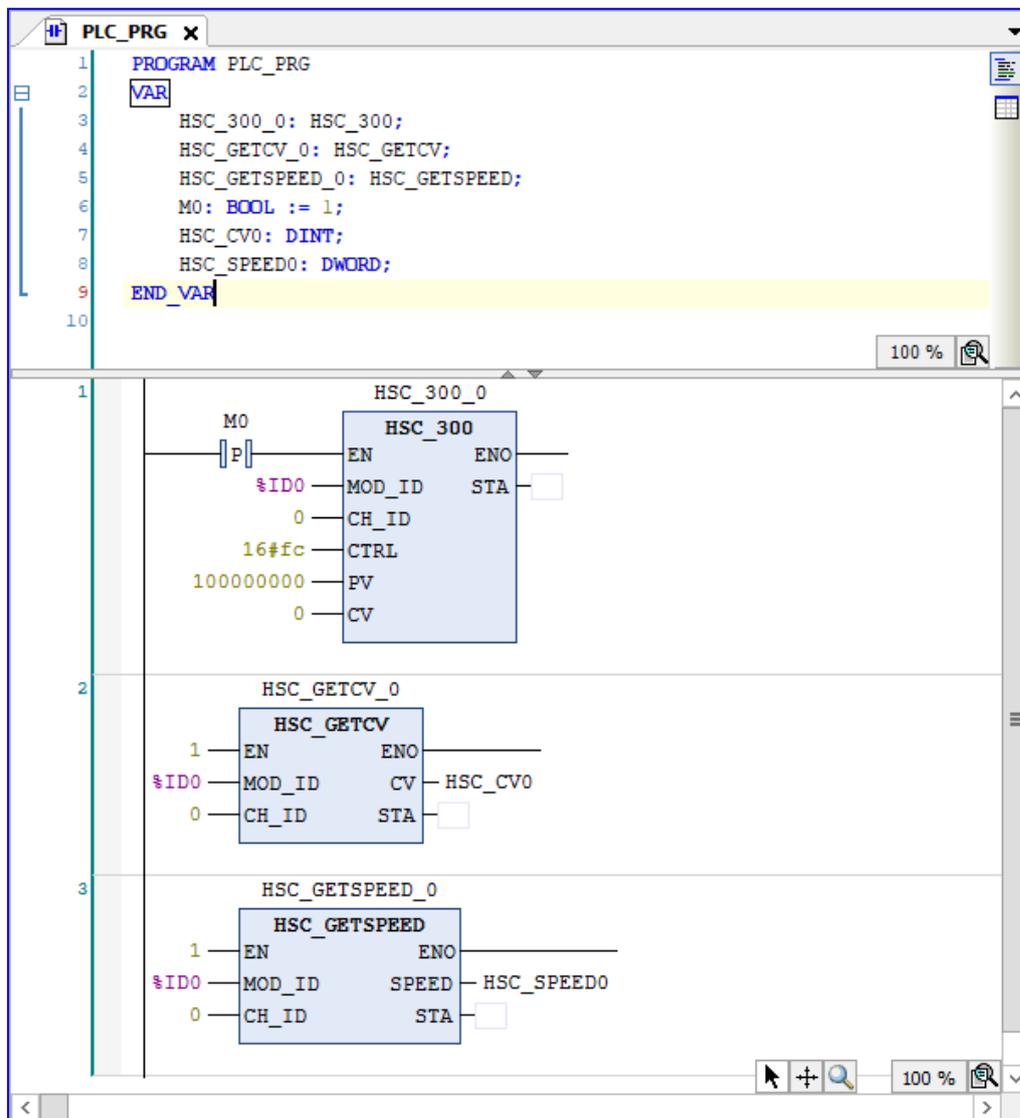
上传源程序：



为以下所有选项选择“是”。



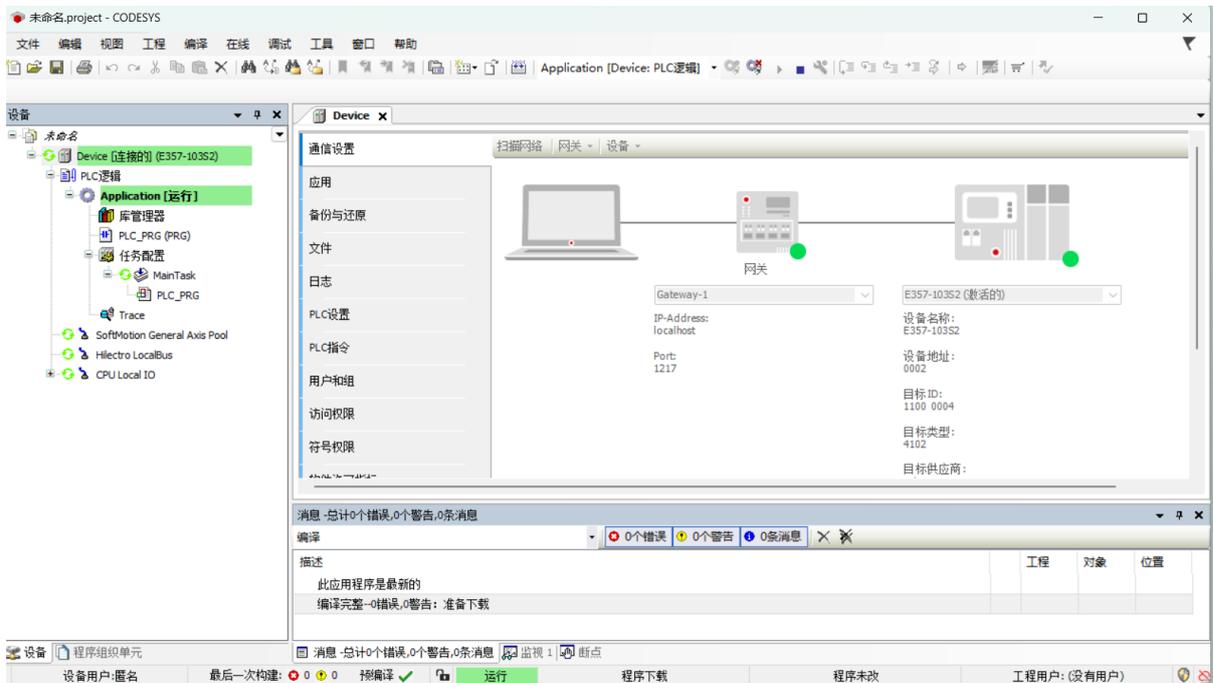
成功上传的程序如下:



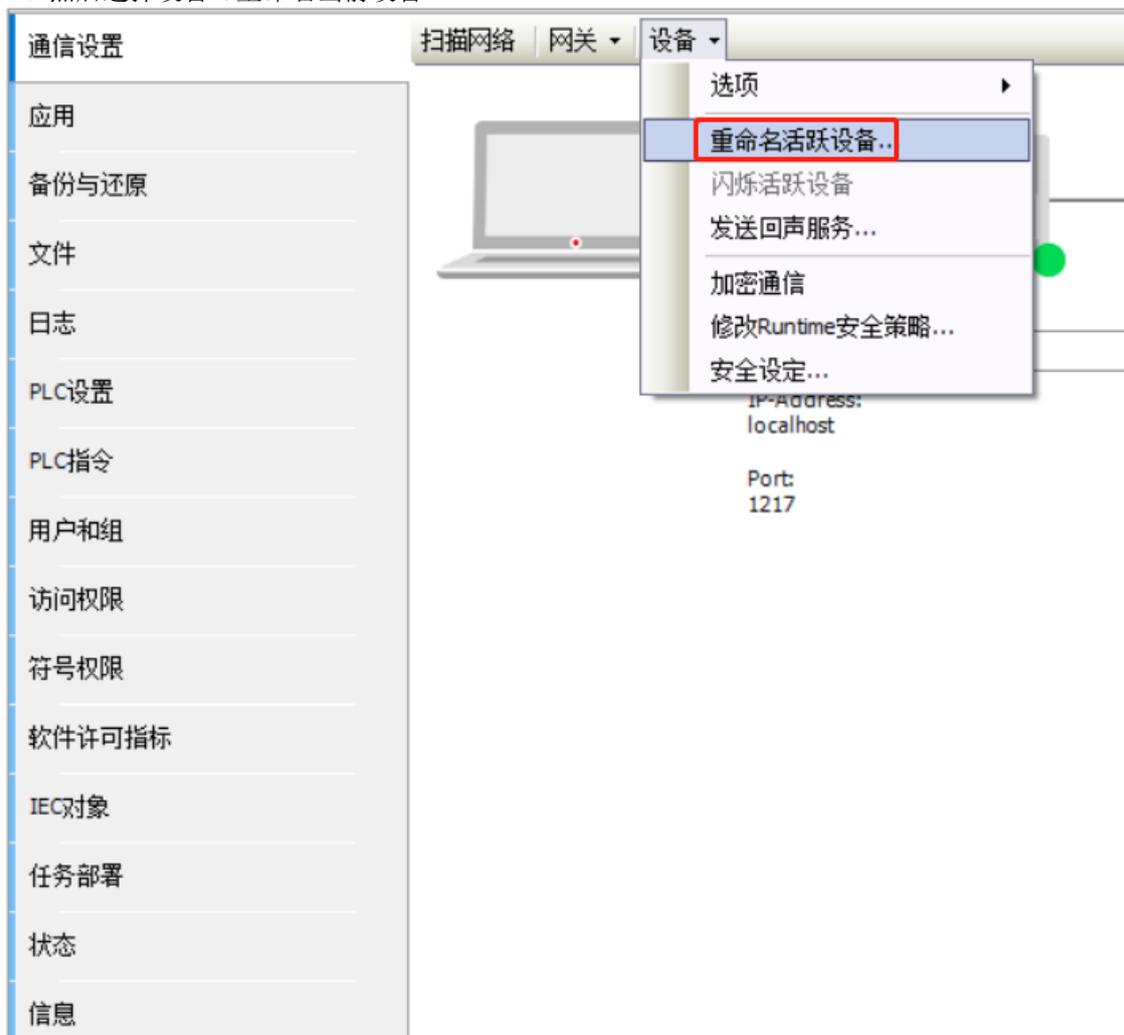
## 5.8 同一网络下存在多个 PLC 时如何更改 PLC 名称

当有多个相同 PLC 处于同一个网络下时，不太好区分哪一个是要操作的 PLC，可以通过下述方式给 PLC 重命名来进行区分。

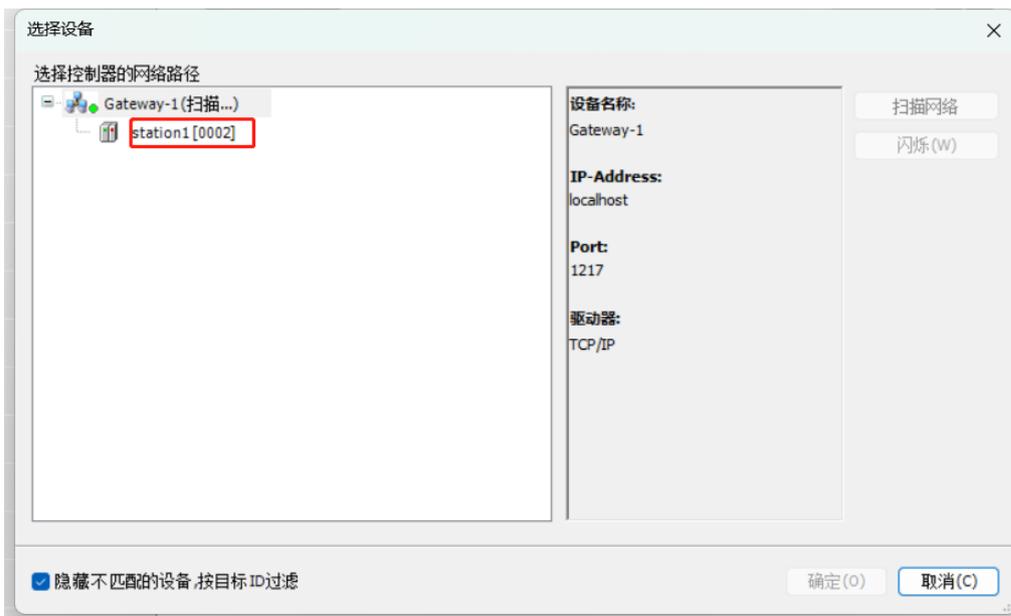
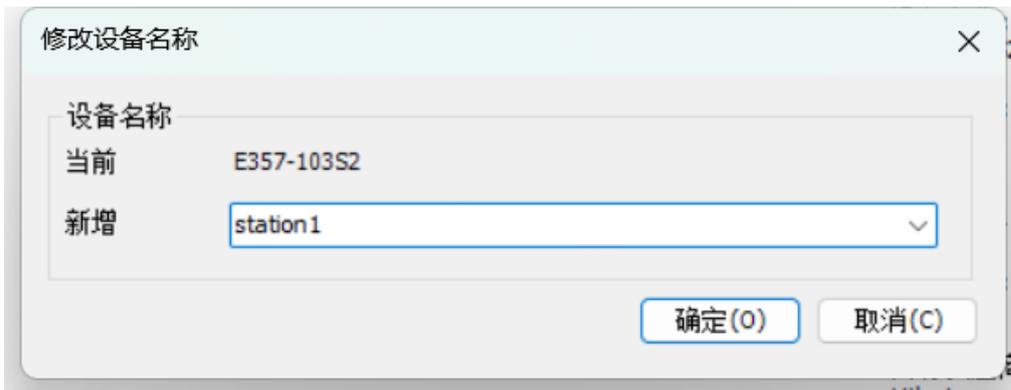
- 1、首先和 PLC 建立通讯



2、然后选择设备→重命名当前设备



3、输入新名称后点击确定即可完成重命名

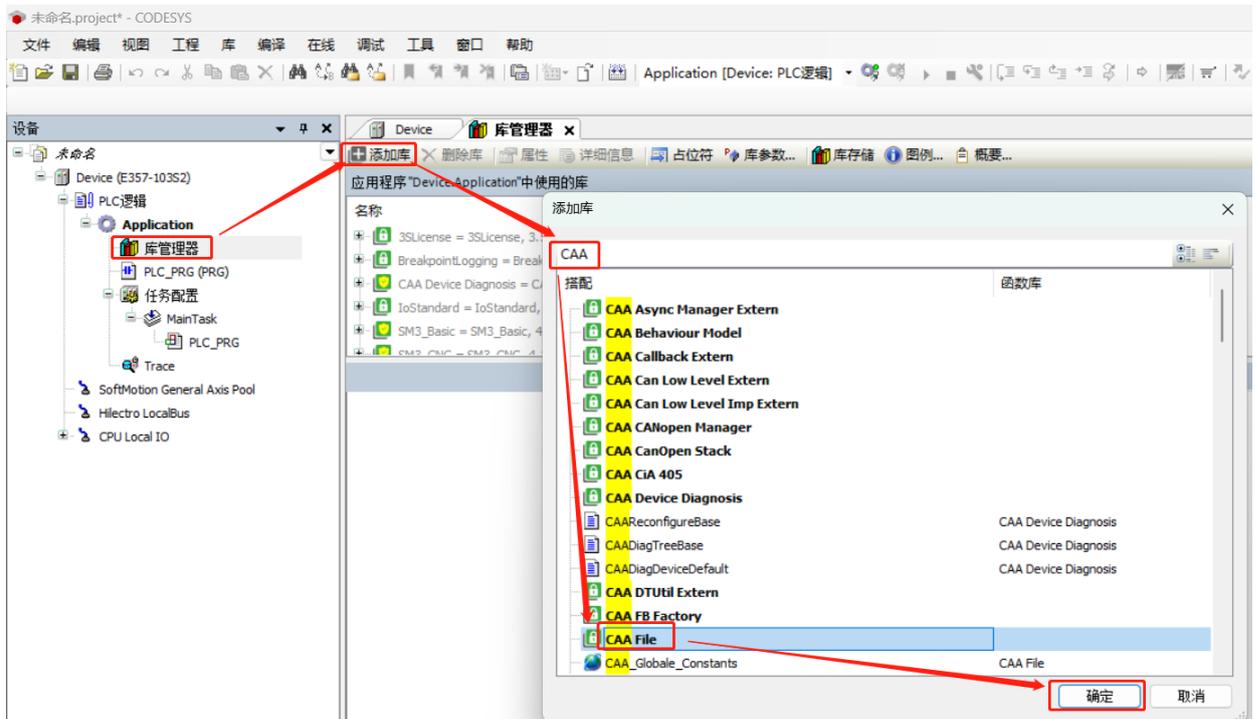


## 5.9 如何将文件写入 U 盘

使用场景：随着运动控制的发展，我们需要处理的数据越来越多，在 CNC 应用中，G 代码的文件通常为几 K 到几百 K 不等；在电子凸轮采样数据应用中，我们的数据也有几万个。PLC 永久保存只有 64KB，但 CNC 文件都需要 100K 以上的数据需要掉电保存。

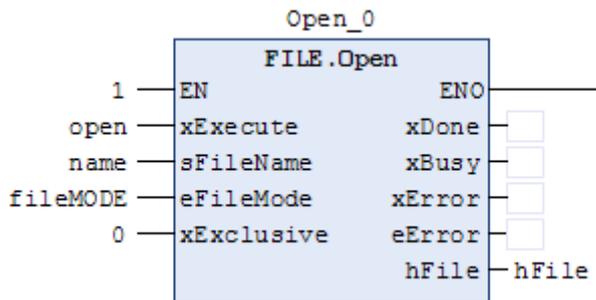
解决方案：CODESYS 集成 CAA Libraries 中的 CAA\_FILE.library 提供了文件读写的函数库，此库提供的功能块用于访问文件目录系统和文件。下面本文将介绍该库的使用办法。

本库中包含了变量类型的定义、模块，但本文主要是介绍关键模块的使用，包括 FILE.Open、FILE.Close、FILE.Write、FILE.Read 和 FILE.Flush，以及变量类型 FILE.MODE、FILE.ERROR。其他模块和变量类型请自行查阅 CODESYS 帮助文档。



## 1、FILE.Open

此功能块打开已存在的文件或创建一个新文件。



输入输出	参数名	数据类型	说明
IN	xExecute	BOOL	触发位，在处理文件时应一直导通。
IN	sFileName	string	文件名称，建议用英文。
IN	eFileMode		访问文件的模式，类型为 FILE.MODE。一共有四种。 FILE.MODE.MWRITE：写访问，文件将被覆盖或创建。 FILE.MODE.MREAD：读访问，文件将被打开以供阅读。 FILE.MODE.MRDWR：读写访问，该文件将被覆盖或创建。 FILE.MODE.MAPPD：文件将以 WRITE 模式打开，但写入的数据将附加在文件的末尾。
IN	xExclusive	BOOL	文件访问数量。 TRUE：独占数据访问，只有一个 FILE.Open 模块可以访问该文件，其他模块不能访问该文件，其他 FILE.Open 模块访问该文件的时候会出错，xError 为 TRUE。 FALSE：可以进行多个 FILE.Open 模块的数据访问。
OUT	xDone	BOOL	模块执行完成将置 1
OUT	xBusy	BOOL	模块正在执行将置 1
OUT	xError	BOOL	模块执行发生错误时将置 1
OUT	eError		错误代码，类型为 FILE.ERROR

OUT	hFile	返回值是一个文件句柄，可以用作函数块 FILE.Read, FILE.Write, FILE.Close 中的输入“hFile”。
-----	-------	---

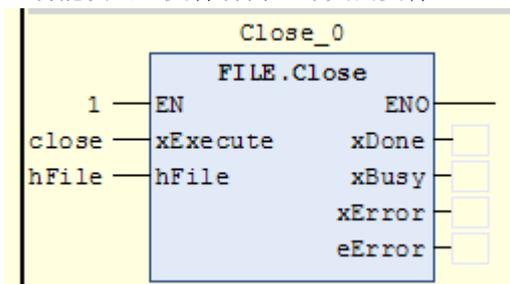
FILE.ERROR 类型此数据结构描述了在处理 CAA\_File.library 的函数时可能发生的错误。错误范围 5100~5199 保留用于 CAA.File.library。

### 错误码

错误	错误代码	注释
NO_ERROR	0	无错误
FIRST_ERROR	5100	第一个库特定的错误
TIME_OUT	5101	超过时间限制
ABORT	5102	通过激活输入 xAbort 中止了操作
HANDLE_INVALID	5103	无效句柄
NOT_EXIST	5104	目录或文件不存在
EXIST	5105	目录或文件已经存在
NO_MORE_ENTRIES	5106	没有其他条目可用
NOT_EMPTY	5107	目录或文件不为空
READ_ONLY_CAA	5108	文件或目录是写保护的
WRONG_PARAMETER	5109	错误参数
ERROR_UNKNOWN	5110	未知错误
WRITE_INCOMPLETE	5111	并非所有数据都已写入
NOT_IMPLEMENTED	5112	未实现
FIRST_MF	5150	第一个制造商特定错误
LAST_ERROR	5199	最新的库特定错误

## 2、FILE.Close

此功能块终止文件访问，即关闭文件。

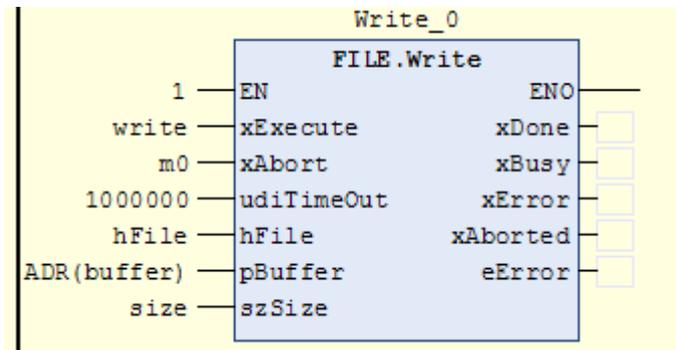


输入输出	参数名	数据类型	说明
IN	xExecute	BOOL	触发位，在处理文件时应一直导通。
IN	hFile		文件句柄，由 FILE.Open 的执行结果中得到
OUT	xDone	BOOL	模块执行完成将置 1

OUT	xBusy	BOOL	模块正在执行将置 1
OUT	xError	BOOL	模块执行发生错误时将置 1
OUT	eError		错误代码，类型为 FILE.ERROR

### 3、FILE.Write

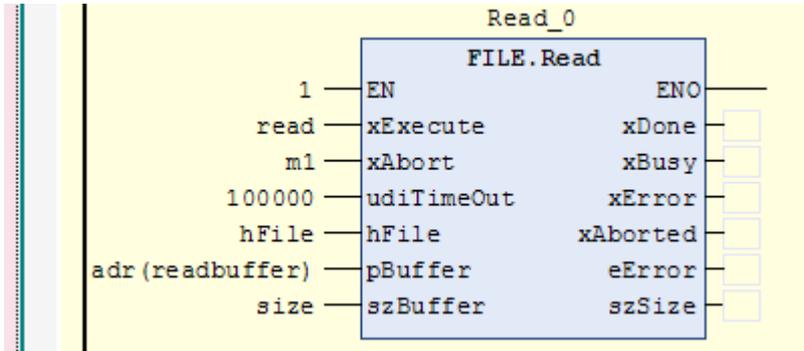
此功能块将数据写入文件，该文件以前应由 FILE.Open 打开。指针“pBuffer”指示的存储器区域的内容在写入操作期间不可修改！不检查包含要写入的字节的数据结构的大小以及要写入的字节数。



输入输出	参数名	数据类型	说明
IN	xExecute	BOOL	触发位，在处理文件时应一直导通。
IN	xAbort	BOOL	终止
IN	udiTimeOut		超时时间
IN	hFile		文件句柄，由 FILE.Open 的执行结果中得到
IN	pBuffer		获得数据的地址，可以使用 adr 指令
IN	szSize		要写入的字节数，可以通过 sizeof 操作检索
OUT	xDone	BOOL	模块执行完成将置 1
OUT	xBusy	BOOL	模块正在执行将置 1
OUT	xError	BOOL	模块执行发生错误时将置 1
OUT	xAborted		写入文件被中断
OUT	eError		错误代码，类型为 FILE.ERROR

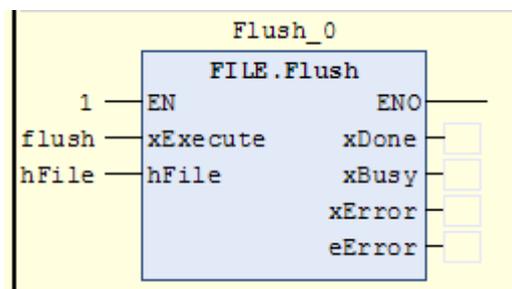
### 4、FILE.Read

此功能块读取以前通过 FILE.Open 打开的文件。如果读取的字符少于“szBuffer”中指定的字符数，则功能块返回一个活动的“xDone”，并指示“szSize”中当前的字符数。不检查要读取的字节的目标存储器结构的大小和要读取的字节数。



输入输出	参数名	数据类型	说明
IN	xExecute	BOOL	触发位，在处理文件时应一直导通。
IN	xAbort	BOOL	终止
IN	udiTimeOut		超时时间
IN	hFile		文件句柄，由 FILE.Open 的执行结果中得到
IN	pBuffer		存入数据的地址，可以使用 adr 指令
IN	szBuffer		要读取的最大字节数，可以通过 sizeof 操作得到
OUT	xDone	BOOL	模块执行完成将置 1
OUT	xBusy	BOOL	模块正在执行将置 1
OUT	xError	BOOL	模块执行发生错误时将置 1
OUT	xAborted		写入文件被中断
OUT	eError		错误代码，类型为 FILE.ERROR
OUT	szSize		读取到的数据字节数

## 5、FILE.Flush



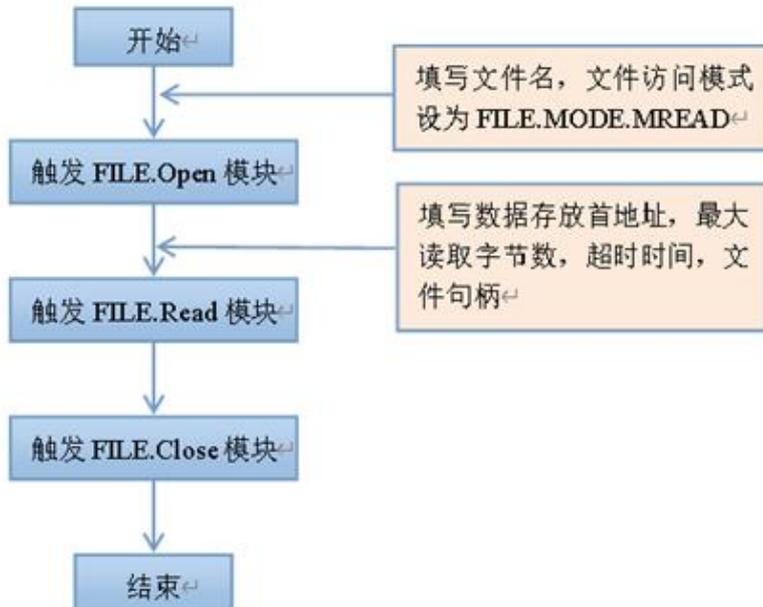
此功能块将数据从文件系统的高速缓存拷贝到磁盘，调用此功能块可确保数据将存储在磁盘上。

输入输出	参数名	数据类型	说明
IN	xExecute	BOOL	触发位
IN	hFile		文件句柄，由 FILE.Open 的执行结果中得到
OUT	xDone	BOOL	模块执行完成将置 1

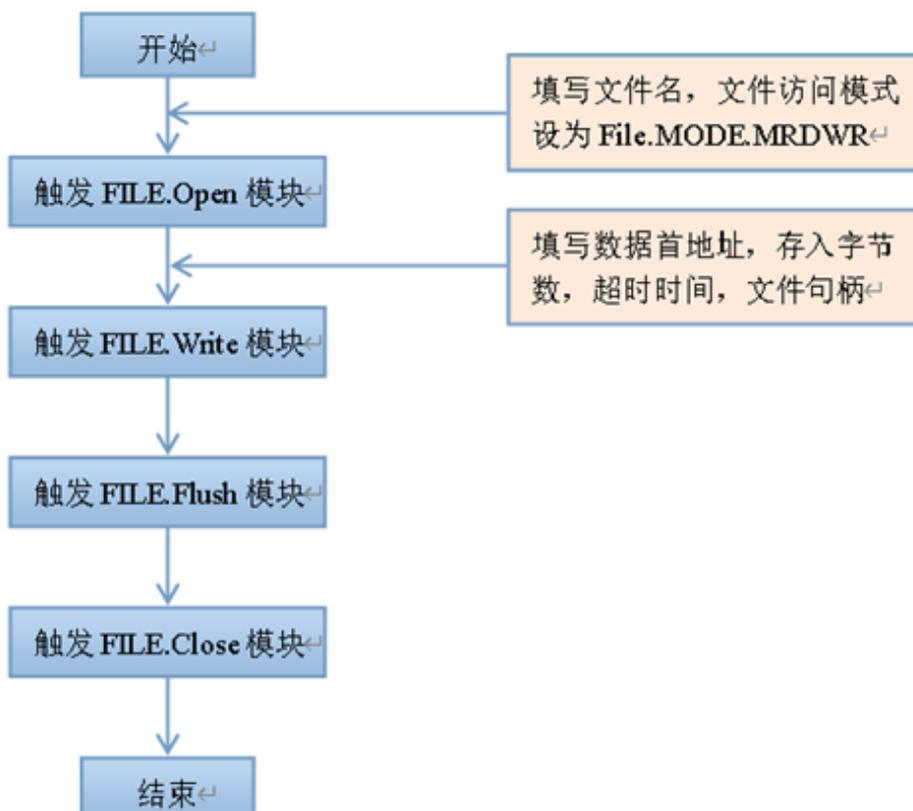
OUT	xBusy	BOOL	模块正在执行将置 1
OUT	xError	BOOL	模块执行发生错误时将置 1
OUT	eError		错误代码，类型为 FILE.ERROR

下面是具体使用流程

读操作：打开文件（FILE.MODE.MREAD 模式）→-读取数据→关闭文件

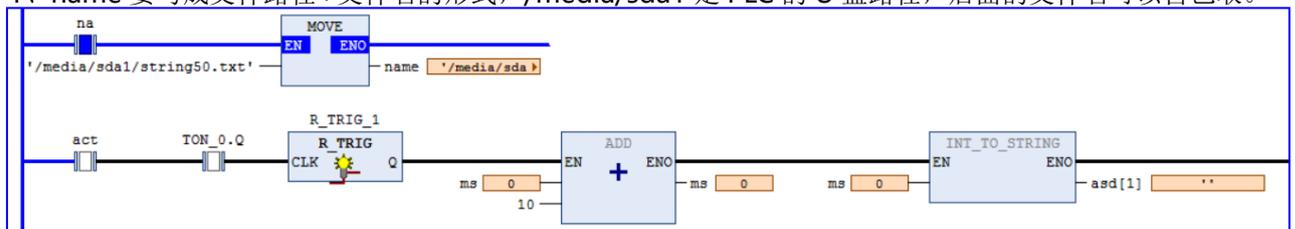


写操作：打开或新建文件（FILE.MODE.MRDWR 模式）→写入数据→保存数据→关闭文件



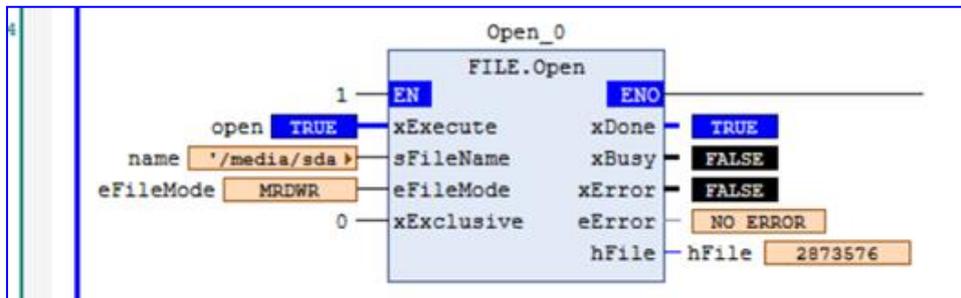
根据读写文件的流程，本文提供一个编程示例，具体如下：

1、name 要写成文件路径+文件名的形式， /media/sda1 是 PLC 的 U 盘路径， 后面的文件名可以自己取。

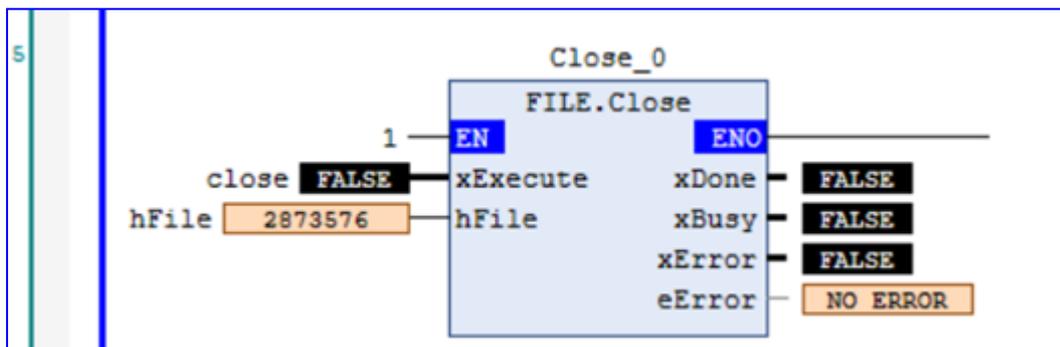


2、此功能用来创建或打开文件。

eFileMode 需要改成 MRDWR 的类型， 然后 open 置 1， hFile 自动生成数字则表示文件创建成功。

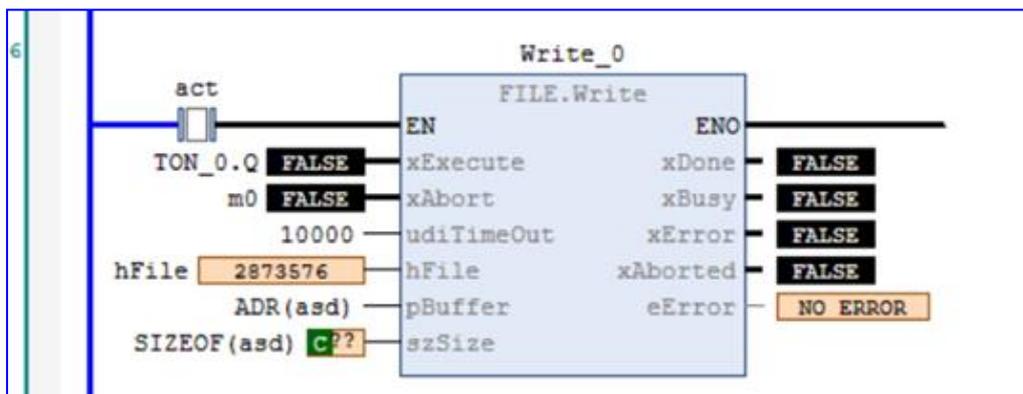


3、此功能用来关闭文件。



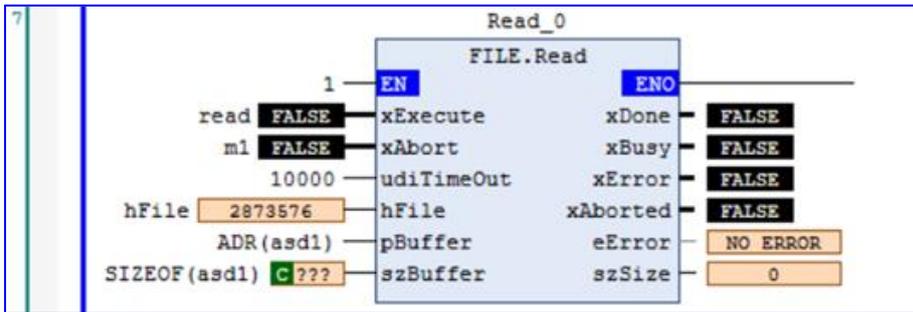
4、此功能用来写入文件。

pBuffer 获得数据的地址， 可以使用 adr 指令。（把需要写入到 U 盘的数据， 写入该地址）

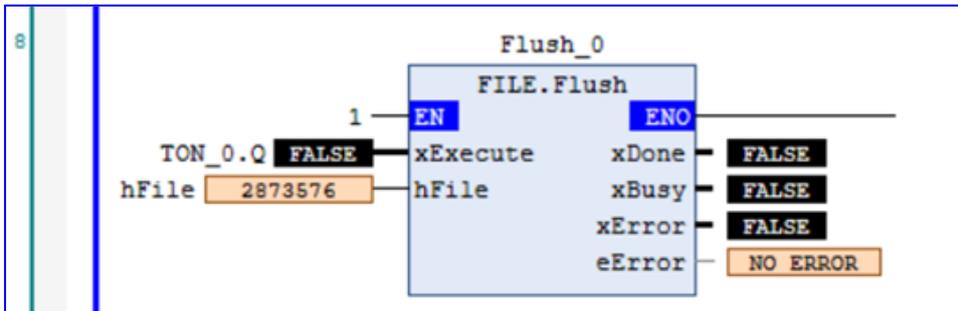


5、此功能用来写读文件。

pBuffer 获得数据的地址， 可以使用 adr 指令。



6、此功能可将文件夹的数据拷贝到 U 盘里。



若写入的数据类型是 STRING 则可以直接用.txt 的格式打开并查看。

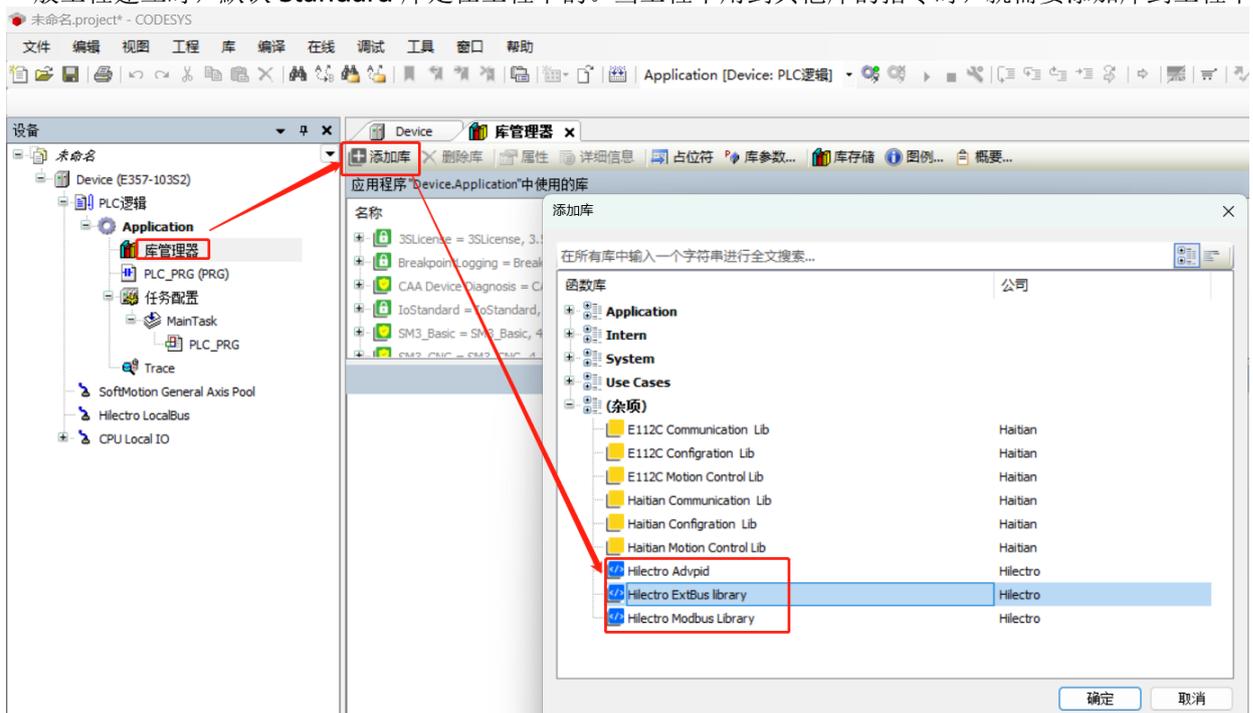
若写入的数据类型是 DINT 直接打开会出现乱码，可以使用 LabVIEW 软件将 DINT 类型改为 STRING 类型。（其他数据类型同理）

使用 LabVIEW 软件还可以修改 U 盘里的数据格式，用 CODESYS 的 Trace 打开并显示。

## 5.10 添加库文件到工程中

编写程序时，经常用到一些指令或功能块，比如触发器、功能块、通信库、计数功能块、PID 控制功能块等。CODESYS 将这些指令和功能块进行分类，然后建立专门的库。

一般工程建立时，默认 Standard 库是在工程中的。当工程中用到其他库的指令时，就需要添加库到工程中。



**<注意>** 添加库文件到工程中和添加库到 CODESYS 中是不一样的，首先我们需要添加外部库到 CODESYS 中，如此在 CODESYS 中有了外部库，然而建立工程时这些外部库是没有包含在工程中的，当工程需要用到这些外部库的时候就需要将库添加到工程中。

## 5.11 复位功能

重置应用程序会停止程序并将变量复位为其初始化值，CODESYS 有三种复位方式：热复位、冷复位和初始值复位。三种方式可以在“在线”菜单中进行选择，单击执行后会弹出提示对话框进行确认。

**热复位：**属于在线命令，在线模式下有效。热复位后，除了保持型变量(**retain** 和 **persistent** 变量)外，其它当前活动的变量都被重新初始化。如果设置了初始值的变量，热复位后变量值为设定的初始值，其它变量都设置为标准初始值（例如：设置为 0）。

**冷复位：**属于在线命令，只在线模式下有效。跟热复位命令不同的是，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** 和 **persistent** 变量）的值也设置为初始值。冷复位发生在程序下载到 PLC 之后，运行之前（冷启动）。一般在总线中断后，可以采取该方式重新启动总线。

**初始值复位：**将所有变量（包括剩余变量）都复位为其初始化值。擦除控制器上的所有用户文件，将控制器置于“空”状态。