



成就客户 | 共创共赢



SC-C系列产品应用手册

雷赛控制技术PLC产品部

2024年4月



雷赛智能
Leadshine

资料获取

- 1、雷赛官网/公众号/联系雷赛人员 (<http://www.leisai.com>)
- 2、各种手册，软件下载，示例程序...

雷赛智能官网

官方对外展示平台



雷赛智能公众号

获取更多应用案例和公司资讯



雷赛智能在线型录

获取更多产品资料





雷赛智能
Leadshine

目录

Contents

01 SC-C 产品介绍

02 Leadsys Studio 软件介绍

03 编程基础知识

04 功能块与函数

05 运动控制轴

06 高速计数器

07 中断任务

08 插补运动

09 同步运动

10 Modbus 通讯示例

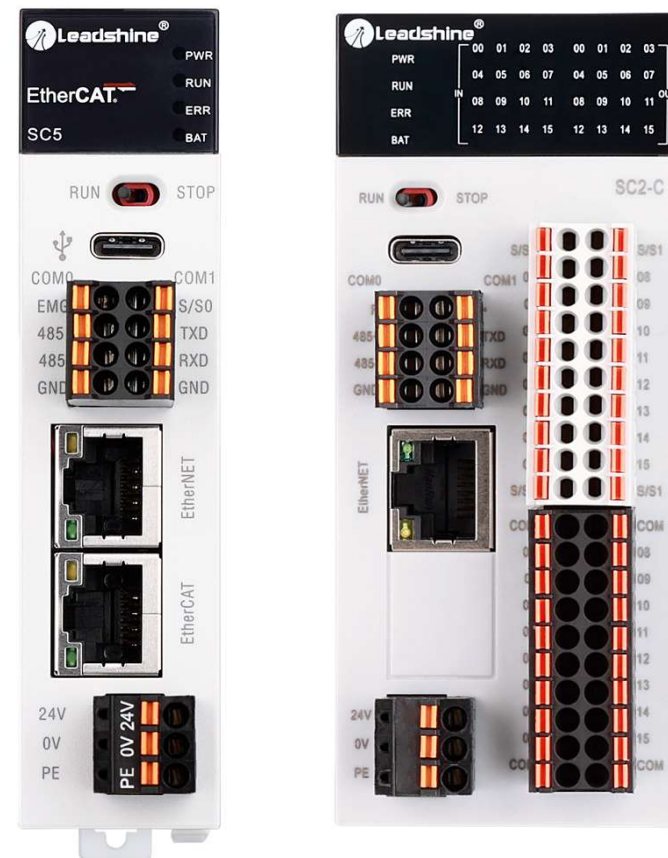
11 实用功能

12 其他资料

01

SC-C产品介绍

- ◆ SC5:4/8EtherCAT总线控制，SC2:4/6/8轴脉冲轴控制
- ◆ 运动控制能力强：支持定位/速度/回原控制；支持电子齿轮/电子凸轮/追剪/飞剪；支持直线/圆弧/连续插补；在线变速变位/速度连续不停顿
- ◆ 支持IEC61131-3标准6种编程语言，一个工程下ST、LD、SFC、CFC、FBD、IL多语言混合编程；
- ◆ 功能库符合PLCopen规范，标准易用；
- ◆ 本机自带以太网、Type C、RS232和RS485接口
- ◆ 主机本地可带16个R1系列扩展模块



SC-C系列经济版小型PLC

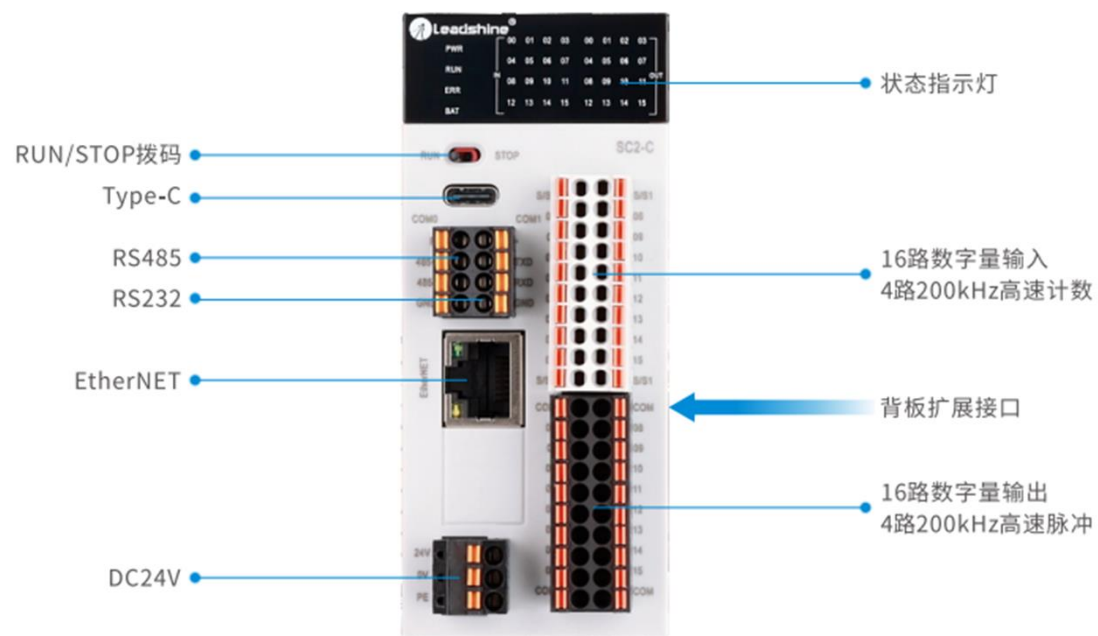
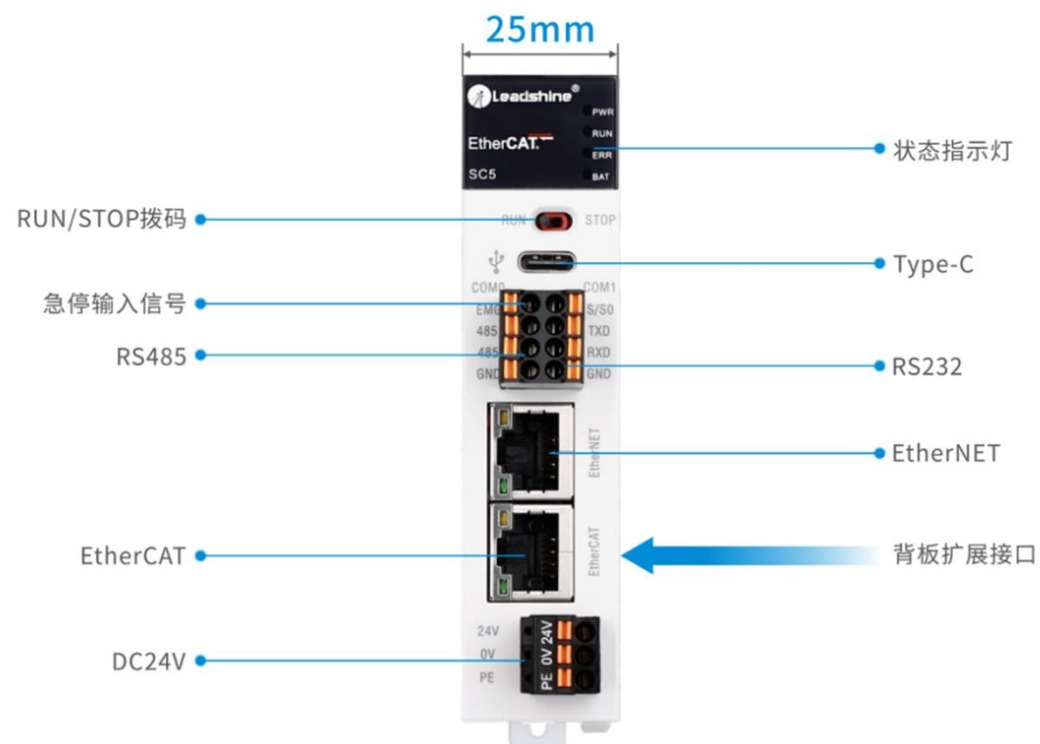
产品规格



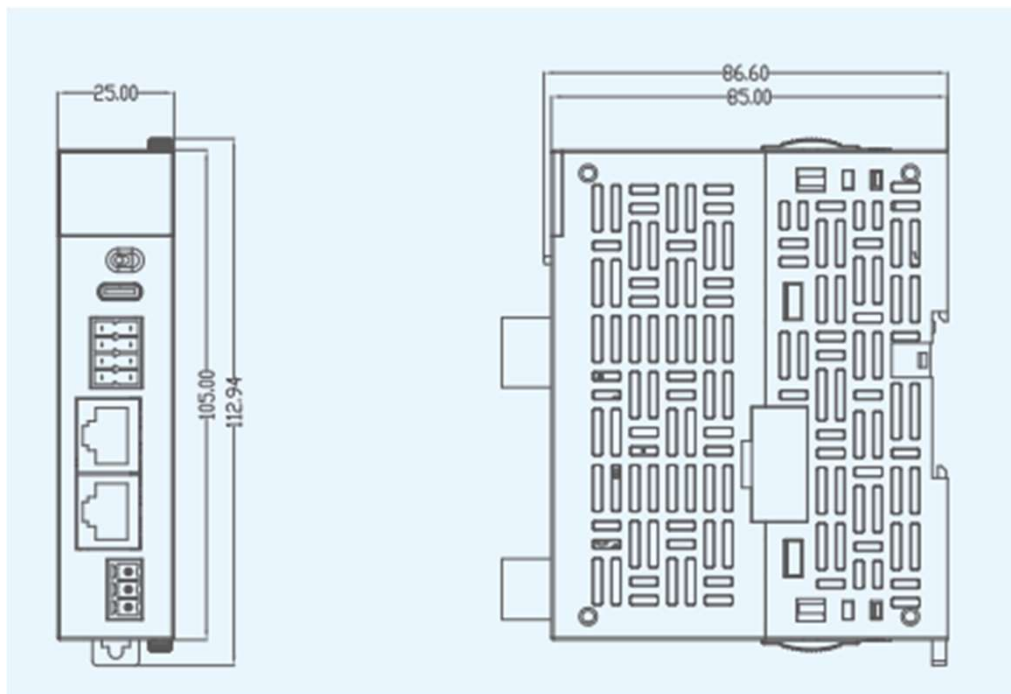
成就客户 | 共创共赢

型号		SC2-C32A4DS	SC2-C32A6DS	SC2-C32A8DS	SC5-C0A4	SC5-C0A8
轴数		4轴200KHz脉冲	6轴200KHz脉冲	8轴200KHz脉冲	4轴EtherCAT总线	8轴EtherCAT总线
高速脉冲输出	脉冲模式	A/B相、脉冲+方向、CW/CCW	A/B相、脉冲+方向		不支持	
	PWM功能	2路	不支持			
	探针	每个脉冲轴1路	不支持			
高速计数	通道数	4路200KHz			不支持	
	计数器模式	A/B相1/2/4倍频、单相、脉冲+方向、CW/CCW	A/B相1/2/4倍频、脉冲+方向			
	高速探针	每个计数器1路	不支持			
	比较输出	每个计数器1路	不支持			
	计数中断任务	每个计数器1路	不支持			
	预置值	外部信号、软件逻辑电平触发预置值				
外部中断	6个 (IN0-IN5) 高速口, 上升沿和下降沿	不支持		不支持		
运控能力	电子齿轮\电子凸轮/追剪/飞剪; 直线/圆弧/连续插补; 定位/回原/速度/转矩控制;在线变速变位, 速度连续不停顿					
以太网	1个, 支持Modbus-TCP主从站; EtherNet/IP主从站; SOCKET自由协议					
串口通信	RS232*1, RS485*1, 支持Modbus-RTU主从站、串口自由协议					
程序容量	8MByte			16MByte		
数据容量	16MByte, 其中256KByte掉电保持空间			30MByte, 其中256KByte掉电保持空间		
数字量规格	16 IN (NPN/PNP)/16 OUT(NPN)			1 IN (急停任务)		
其他接口	Type-C直接供电调试 (程序上下载、监控、固件升级); 支持Type-C接口U盘 (FAT32格式) 文件读写、固件升级、程序更新; RUN/STOP (连续拨动5次以上, 恢复默认IP)					
右扩展模块	16个R1系列IO扩展模块					
编程语言	LD、ST、CFC、FBD、SFC、IL					

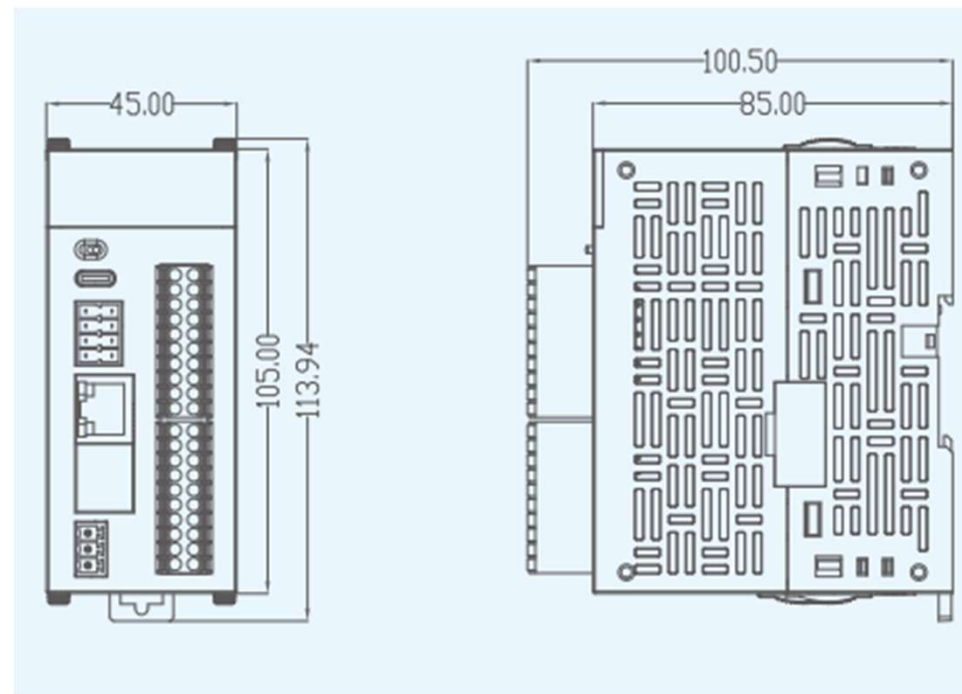
硬件接口介绍



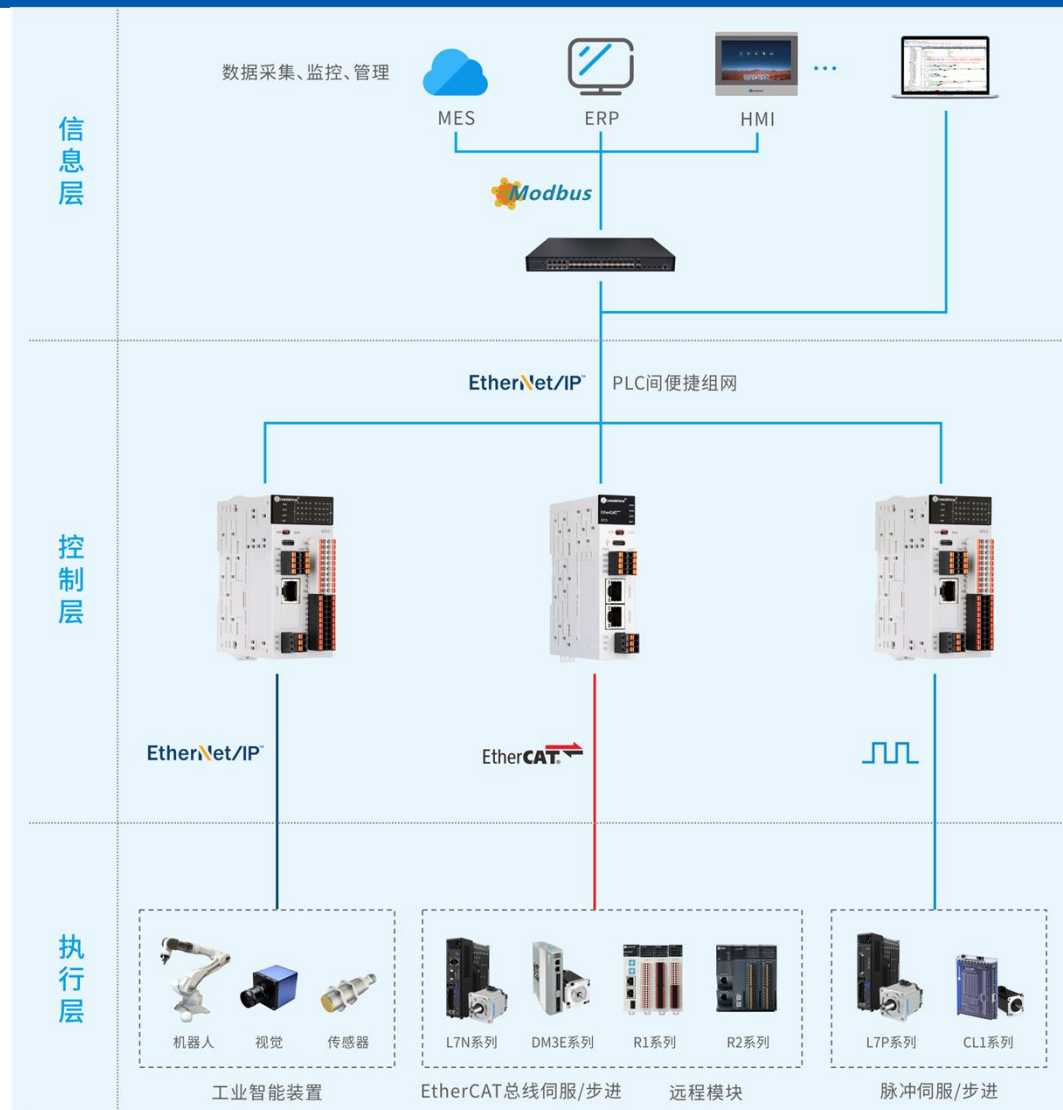
SC5-C系列主机 适用于SC5-C0A4/SC5-C0A8



SC2-C系列主机 适用于SC2-C32A4DS/SC2-C32A6DS/SC2-C32A8DS



全系统拓扑结构图



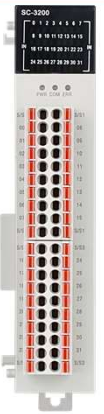
SC系列扩展模块



EtherCAT
耦合器
R1EC



数字量输入
SC-1600



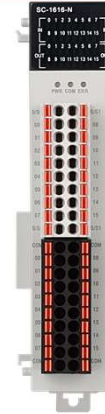
数字量输入
SC-3200



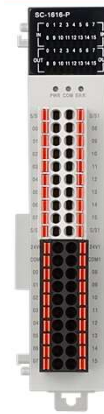
数字量输入
SC-3200-1



数字量输入
输出
SC-0808-N



数字量输入
输出
SC-1616-N



数字量输入
输出
SC-1616-P



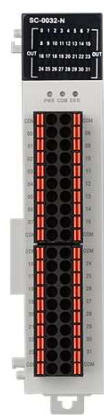
数字量
输出
SC-
0016-N



数字量
输出
SC-
0016-P



数字量
输出
SC-
0016-R



数字量
输出
SC-
0032-N



数字量
输出
SC-0032-
N-1



模拟量
输入
SC-
A004-IV

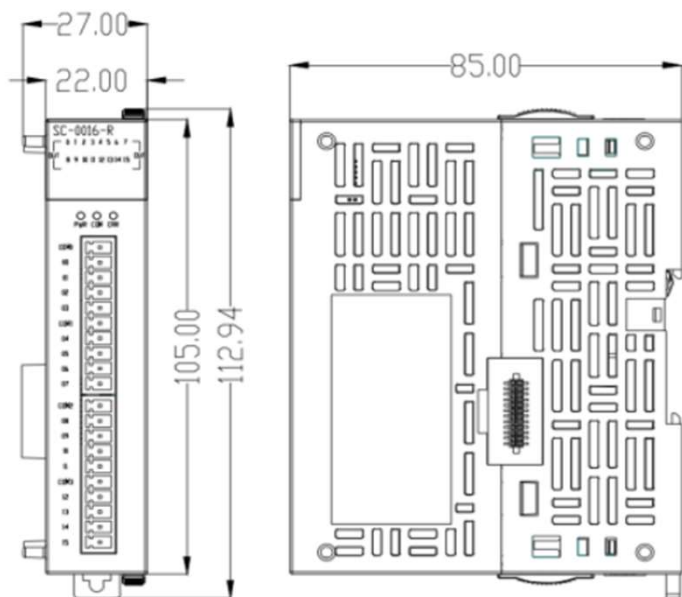


模拟量
输出
SC-A0400-
IV

SC系列扩展模块安装尺寸

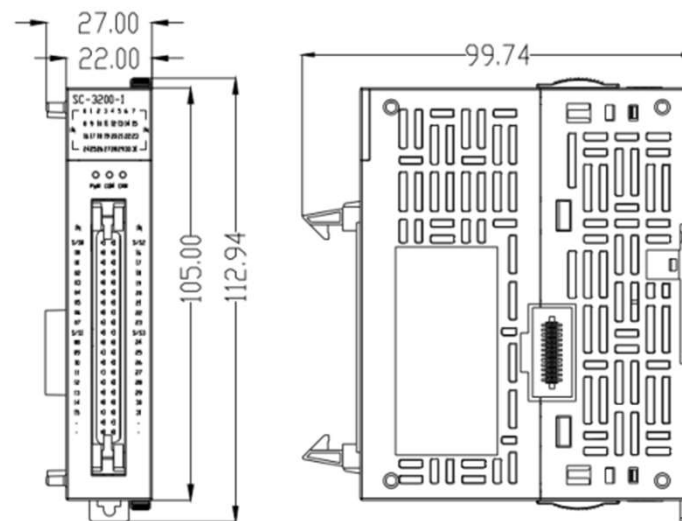
扩展模块尺寸

适用于SC-1600/SC-3200/SC-0016-N/SC-0016-P/SC-0032-N/SC-0016-R/SC-0808-N/SC-1616-N/SC-1616-P/SC-A0400-IV/SC-A0400-IV



扩展模块尺寸

适用于SC-3200-1/SC-0032-N-1

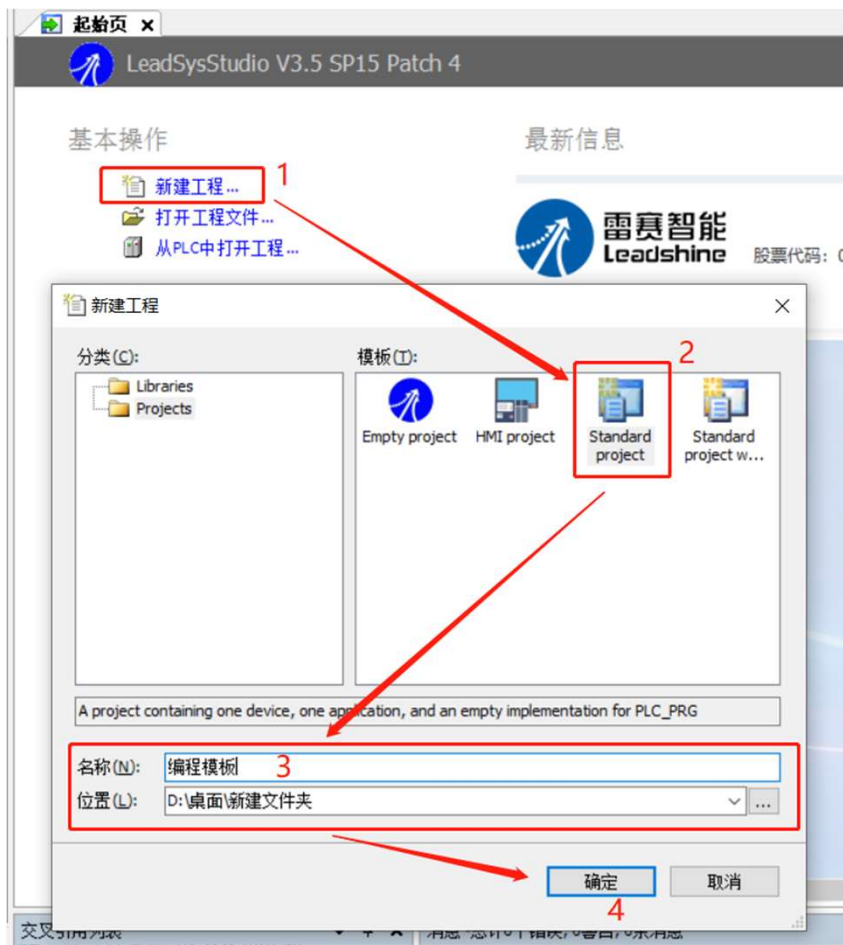


02

Leadsys Studio 软件介绍

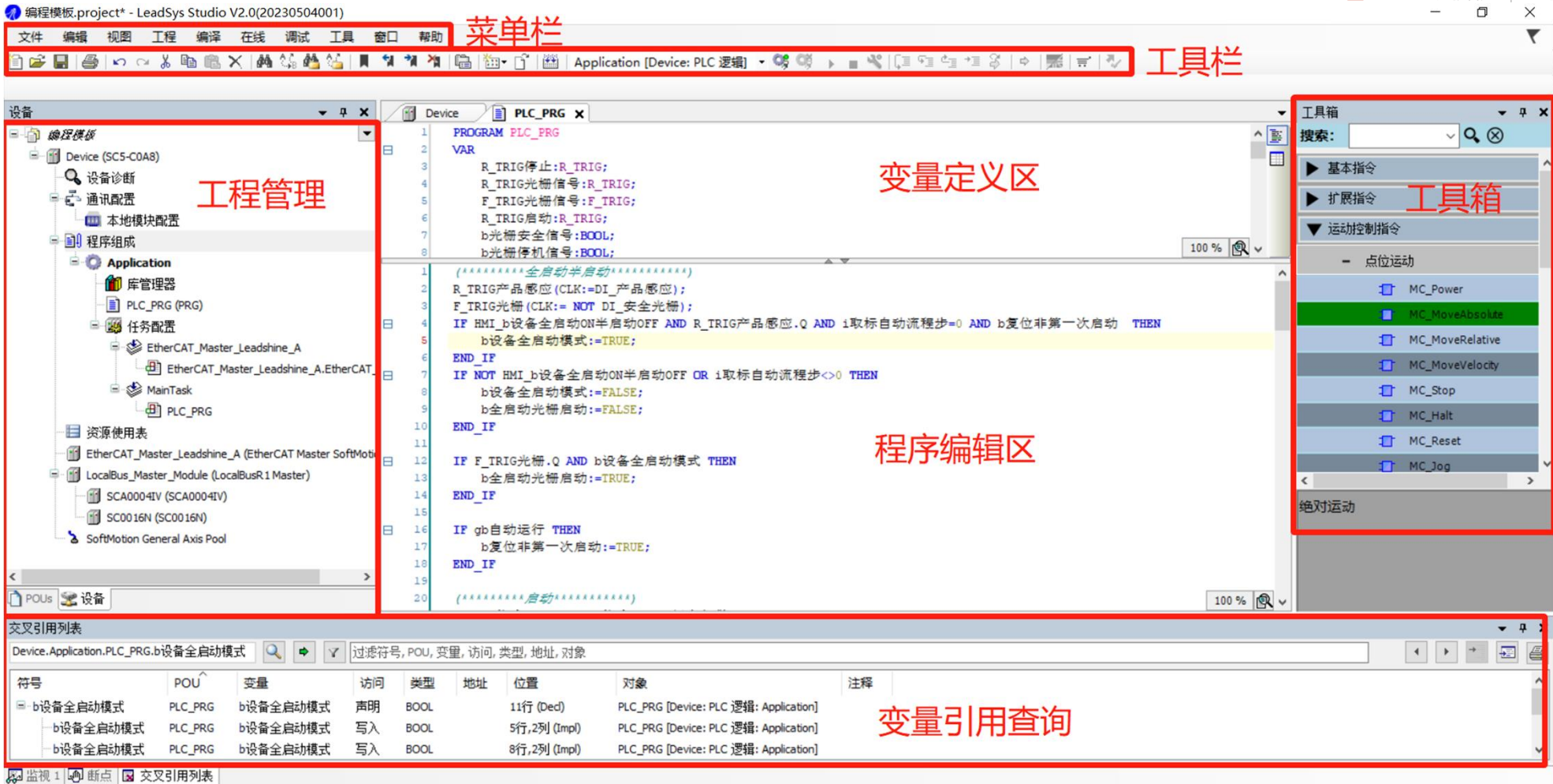
新建工程

1. 打开软件，新建一个标准工程，设定工程名称及保存位置。



2. 选择对应的PLC型号SCx，选择初始POU的编程语言，点击确定后完成工程新建。



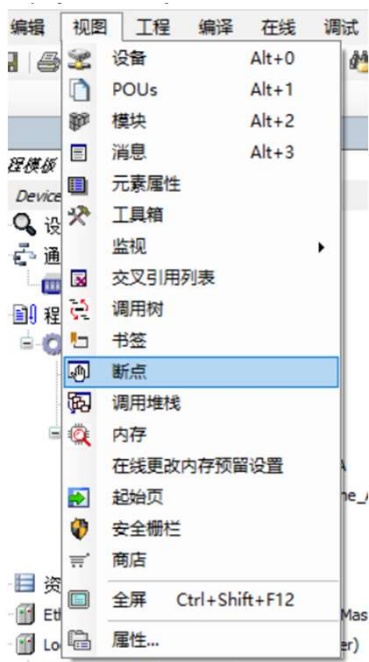


The screenshot shows the LeadSys Studio V2.0 interface with several key components highlighted by red boxes and labels:

- 菜单栏 (Menu Bar):** Located at the top, containing options like 文件 (File), 编辑 (Edit), 视图 (View), 工程 (Project), 编译 (Compile), 在线 (Online), 调试 (Debug), 工具 (Tools), 窗口 (Windows), and 帮助 (Help).
- 工具栏 (Toolbar):** Located below the menu bar, containing various icons for file operations and development tasks.
- 工程管理 (Project Management):** A tree view on the left side showing the project structure, including Device (SC5-C0A8), 设备诊断 (Device Diagnosis), 通讯配置 (Communication Configuration), 本地模块配置 (Local Module Configuration), 程序组成 (Program Structure), Application, 库管理器 (Library Manager), PLC_PRG (PRG), 任务配置 (Task Configuration), EtherCAT_Master_Leadshine_A, EtherCAT_Master_Leadshine_A.EtherCAT, MainTask, PLC_PRG, 资源使用表 (Resource Usage Table), EtherCAT_Master_Leadshine_A (EtherCAT Master SoftMotion), LocalBus_Master_Module (LocalBusR1 Master), SCA0004IV (SCA0004IV), SC0016N (SC0016N), and SoftMotion General Axis Pool.
- 变量定义区 (Variable Definition Area):** The top section of the central code editor, containing variable declarations such as R_TRIG停止:R_TRIG; R_TRIG光栅信号:R_TRIG; F_TRIG光栅信号:F_TRIG; R_TRIG启动:R_TRIG; b光栅安全信号:BOOL; and b光栅停机信号:BOOL;.
- 程序编辑区 (Code Editing Area):** The main area of the code editor containing PLC ladder logic code, including comments like {*****全启动半启动*****} and {*****启动*****}.
- 工具箱 (Toolbox):** Located on the right side, containing a search bar and a list of motion control instructions under the category 运动控制指令 (Motion Control Instructions), such as MC_Power, MC_MoveAbsolute, MC_MoveRelative, MC_MoveVelocity, MC_Stop, MC_Halt, MC_Reset, and MC_Jog.
- 变量引用查询 (Variable Reference Query):** A table at the bottom of the interface showing cross-references for the variable b设备全启动模式.

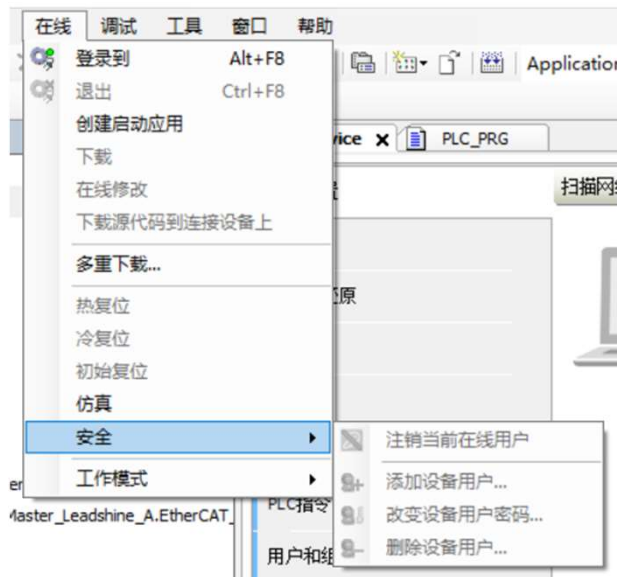
符号	POU	变量	访问	类型	地址	位置	对象	注释
b设备全启动模式	PLC_PRG	b设备全启动模式	声明	BOOL		11行 (Ded)	PLC_PRG [Device: PLC 逻辑: Application]	
b设备全启动模式	PLC_PRG	b设备全启动模式	写入	BOOL		5行,2列 (Impl)	PLC_PRG [Device: PLC 逻辑: Application]	
b设备全启动模式	PLC_PRG	b设备全启动模式	写入	BOOL		8行,2列 (Impl)	PLC_PRG [Device: PLC 逻辑: Application]	

使用LeadSys Studio 软件，主界面中包含菜单栏、工具栏、工程管理区域、变量定义区，代码编辑区以及工具箱6个部分。



视图：功能窗口打开，程序断点调试...

在线：PLC复位，加密设置



调试：断点，设置值，强制值

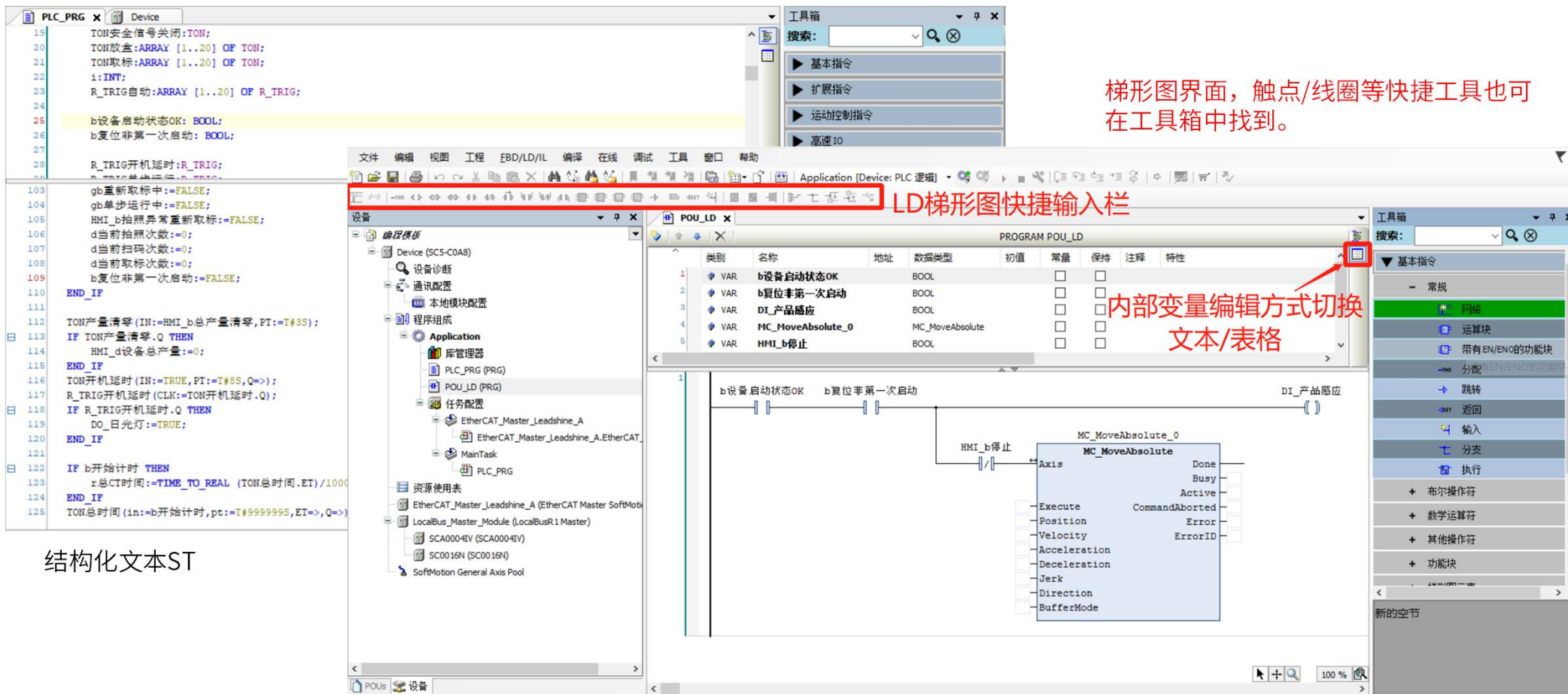
工具：库，设备存储库，固件升级



菜单栏常用功能：文件（上传下载、保存）、视图（打开功能界面）、在线（热/冷/初始复位、仿真）、工具（工艺库、设备库、固件/软件升级）。

软件布局---编程区域

LeadSys Studio支持6种编程语言，用户可根据自己需要选择语言进行编程，这里介绍ST跟LD梯形图编程方式的界面。



The screenshot displays the LeadSys Studio software interface. On the left, the 'PLC_PRG' window shows structured text (ST) code for a PLC program. The main workspace is split into two views: 'POU_ID' (top) and 'LD' (bottom). The 'POU_ID' view shows a table of variables:

类别	名称	地址	数据类型	初值	常量	保持	注释	特性
VAR	b设备启动状态OK		BOOL		<input type="checkbox"/>	<input type="checkbox"/>		
VAR	b复位非第一次启动		BOOL		<input type="checkbox"/>	<input type="checkbox"/>		
VAR	DI_产品感应		BOOL		<input type="checkbox"/>	<input type="checkbox"/>		
VAR	MC_MoveAbsolute_0		MC_MoveAbsolute		<input type="checkbox"/>	<input type="checkbox"/>		
VAR	HMI_b停止		BOOL		<input type="checkbox"/>	<input type="checkbox"/>		

The 'LD' view shows a ladder logic diagram with a network containing a normally open contact for 'b设备启动状态OK', a normally closed contact for 'b复位非第一次启动', and a coil for 'DI_产品感应'. Below the coil is an 'MC_MoveAbsolute_0' block with various parameters like 'Axis', 'Done', 'Busy', etc. The interface includes a menu bar, a toolbar, and a right-hand toolbox with a search function. A red box highlights the 'LD梯形图快捷输入栏' (LD Ladder Logic Quick Input Bar) in the toolbar. A red arrow points to a button in the 'POU_ID' window labeled '内部变量编辑方式切换 文本/表格' (Internal Variable Editing Mode Switch Text/Table).

梯形图界面，触点/线圈等快捷工具也可在工具箱中找到。

内部变量编辑方式切换
文本/表格

结构化文本ST



工程管理区域

资源使用列表：I、Q、M区容量使用情况

EtherCAT_Master_Leadshine: EtherCAT主站配置

L7EC_400: EtherCAT从站配置

Axis: CIA 402轴配置

LocalBus_Master_Module: 本地背板总线

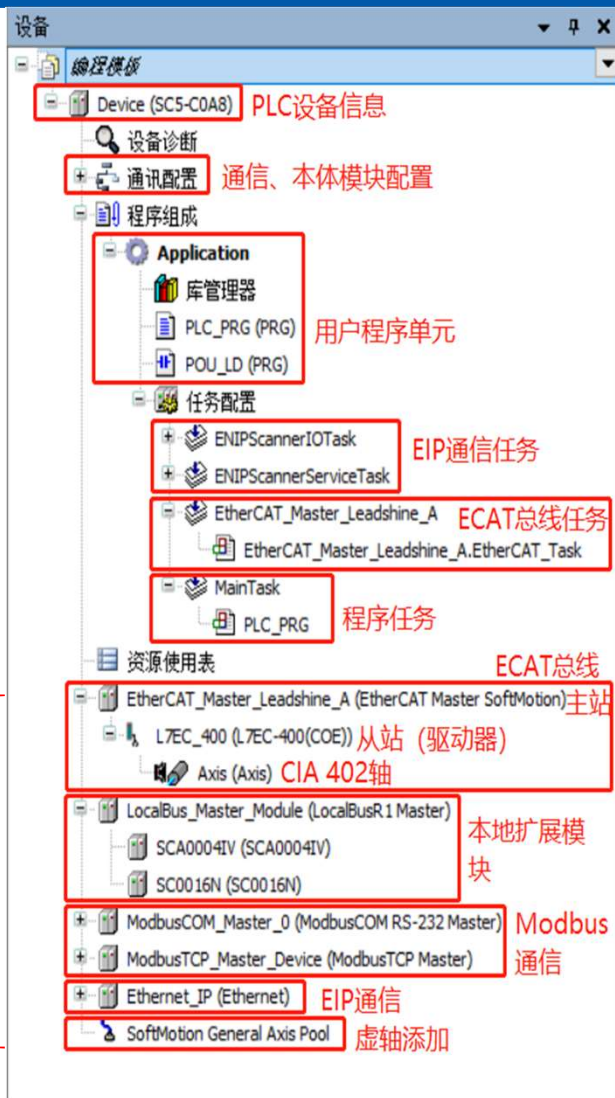
ModbusCOM RS-232 Master: RS232通讯PLC主站配置

ModbusTCP Master: Modbus通讯PLC主站配置

EtherNet_IP: EtherNet IP适配器、扫描器配置

SoftMotion General Axis Pool: 驱动器/编码器虚轴配置

设备树



设备及组态区域

程序区域

Device: 通讯设置, 日志, 系统设置

设备诊断: 读取故障信息

通讯配置: RS232、RS485、EtherCAT、

EtherNet IP、Modbus TCP

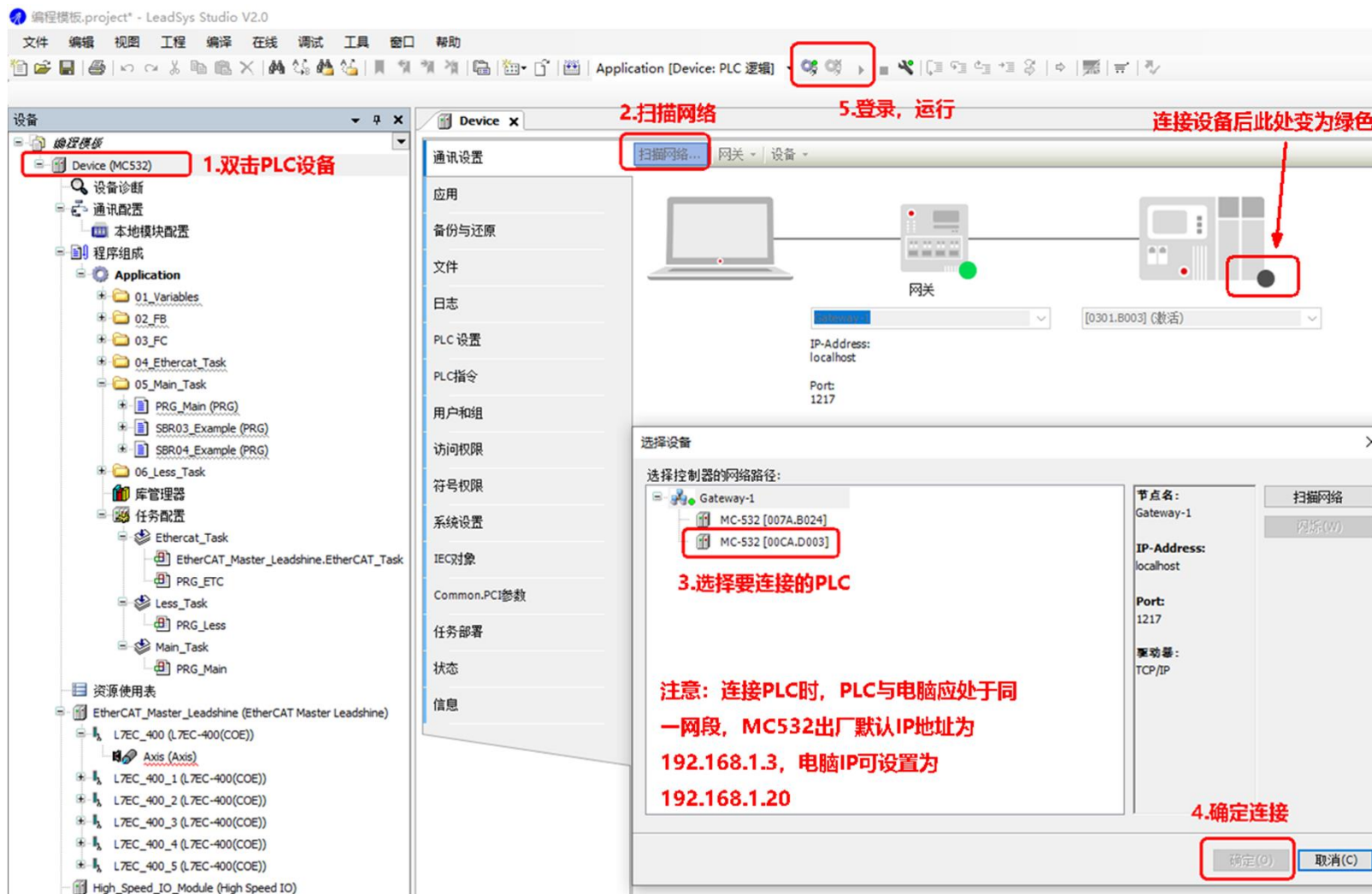
本地模块配置: 本地IO模块配置

Application: 用户程序管理单元

库管理器: 指令库

任务配置: 创建任务执行POU

Device---连接PLC



1. 双击PLC设备

2. 扫描网络

3. 选择要连接的PLC

4. 确定连接

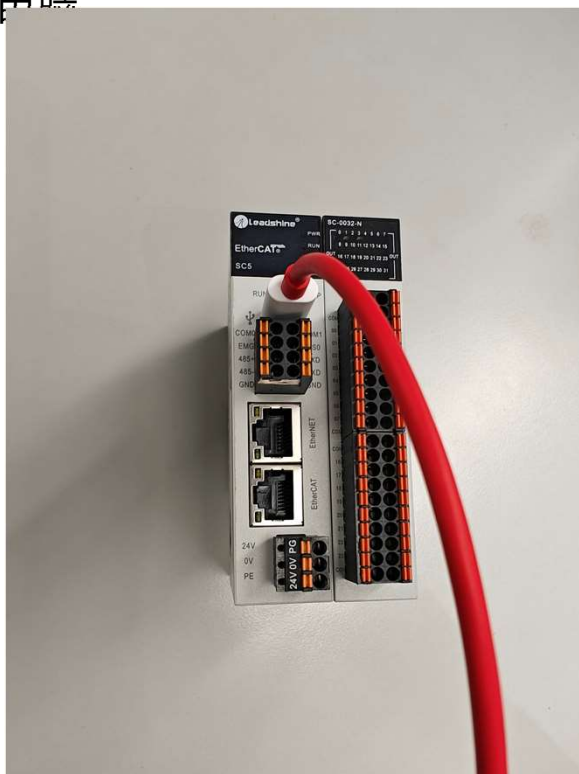
5. 登录, 运行

连接设备后此处变为绿色

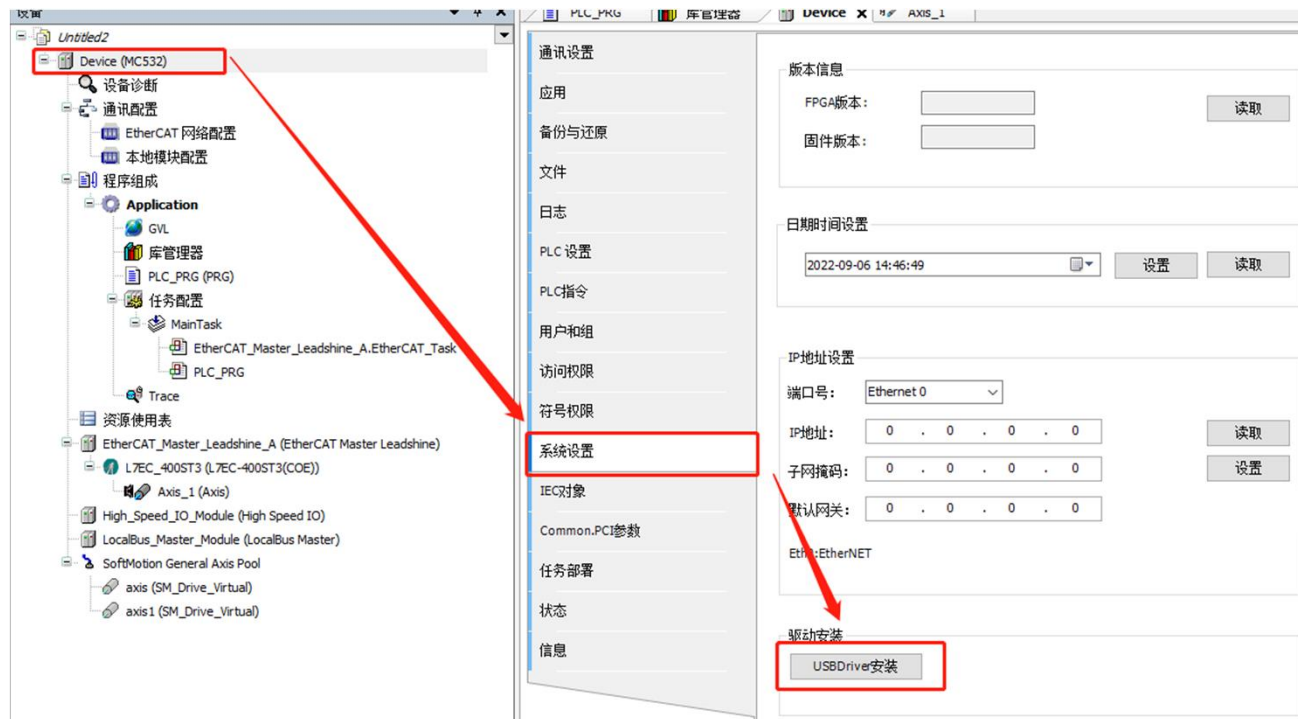
注意: 连接PLC时, PLC与电脑应处于同一网段, MC532出厂默认IP地址为192.168.1.3, 电脑IP可设置为192.168.1.20

Type-C连接PLC使用和安装方法

第一步：准备一台电脑和一根Type-C线，将Type-C头插到PLC上的Type-C接口,如下图所示，另一头插到电脑



第二步：打开软件点击device---选择系统设置---点击驱动安装，如下图所示

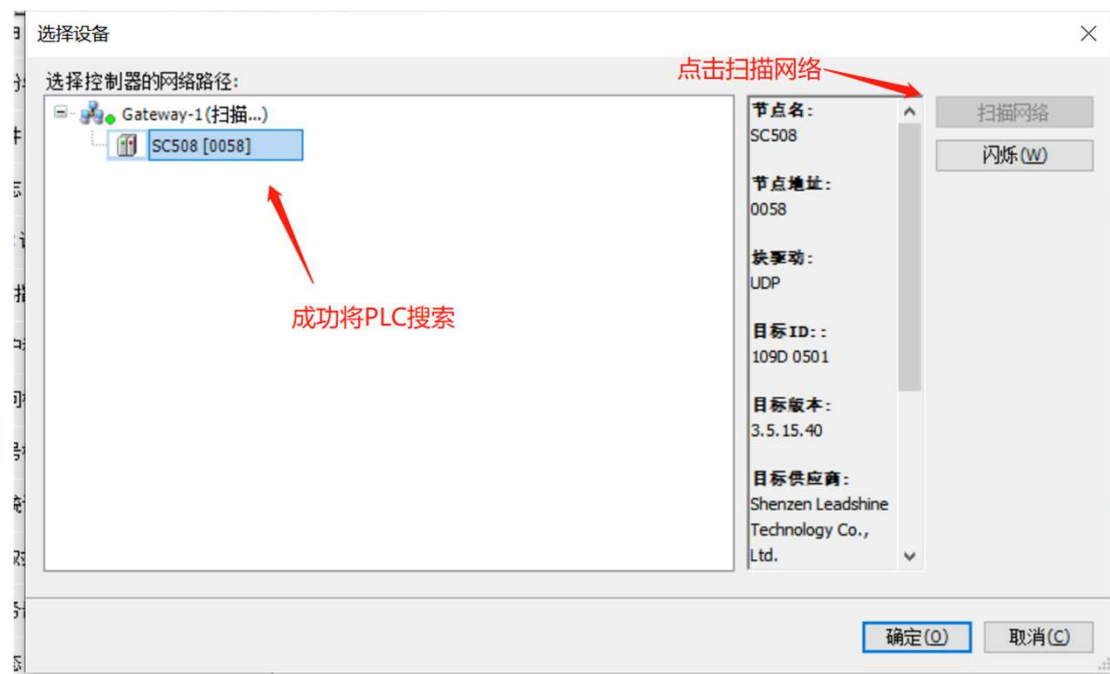


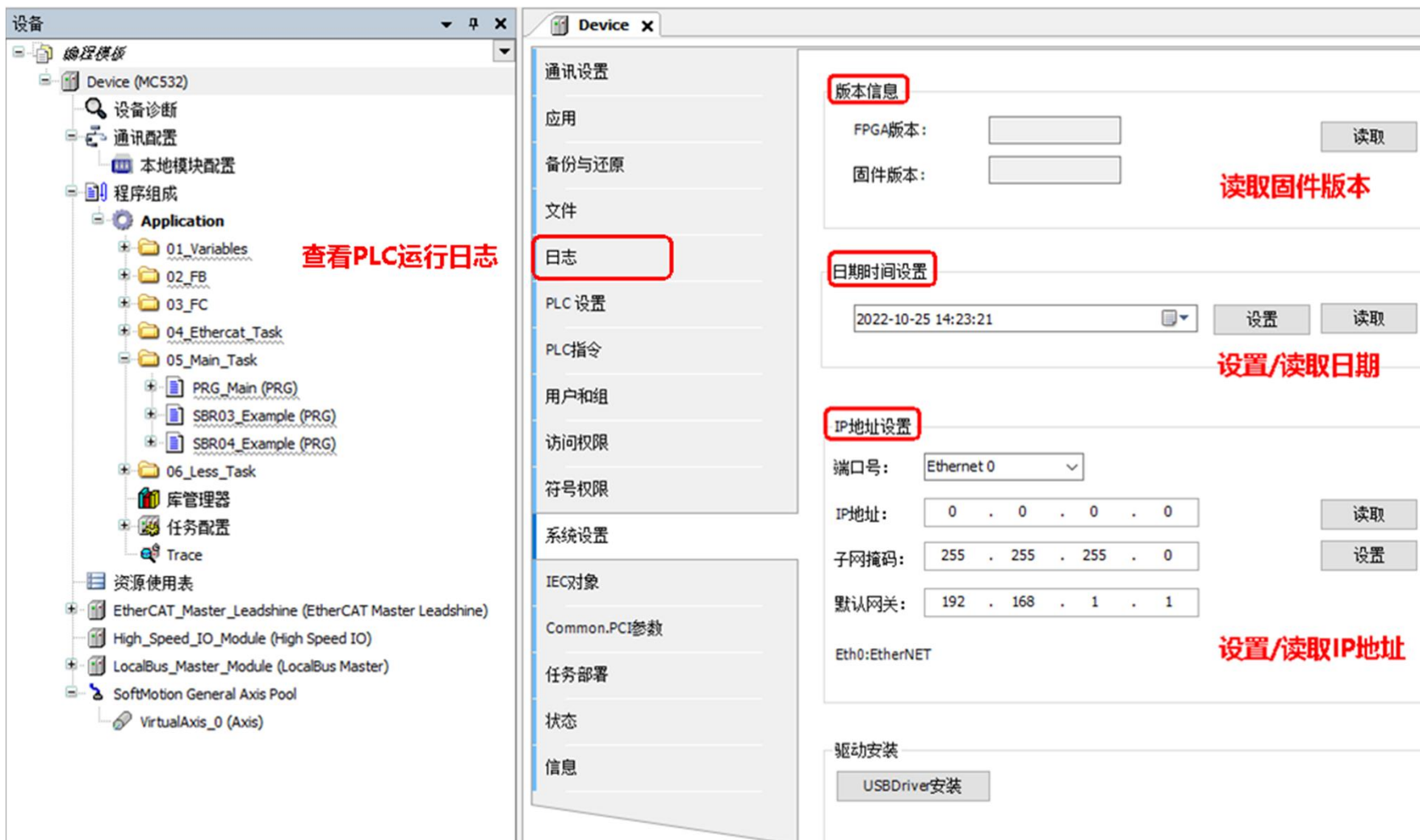
Typc-C连接PLC使用和安装方法

第三步：驱动安装完后，电脑会创建一个虚拟网口通道，如下图所示（PLC的默认ip地址为“192.168.88.88”）

第四步：打开电脑控制面板\网络和 Internet\网络连接（如下图所示），重新设置一下LeSai USB网口的IP(注意：重新设置的IP不要是192.168.88.88)

最后一步，扫描PLC





The screenshot displays the 'Device' configuration window with the '日志' (Log) tab selected in the left-hand menu. The main area is divided into three sections:

- 版本信息 (Version Information):** Contains input fields for 'FPGA版本' and '固件版本', with a '读取' (Read) button and a red label '读取固件版本'.
- 日期时间设置 (Date/Time Settings):** Shows a date/time field set to '2022-10-25 14:23:21', with '设置' (Set) and '读取' (Read) buttons, and a red label '设置/读取日期'.
- IP地址设置 (IP Address Settings):** Includes a '端口号' (Port) dropdown set to 'Ethernet 0', and input fields for 'IP地址' (0.0.0.0), '子网掩码' (255.255.255.0), and '默认网关' (192.168.1.1). It features '读取' (Read) and '设置' (Set) buttons, and a red label '设置/读取IP地址'. Below these is the text 'Eth0:EtherNET'.

At the bottom, there is a '驱动安装' (Driver Installation) section with a 'USB Driver安装' button.

查看PLC运行日志

读取固件版本

设置/读取日期

设置/读取IP地址

通讯配置---EtherCAT组态



1. 双击通讯配置

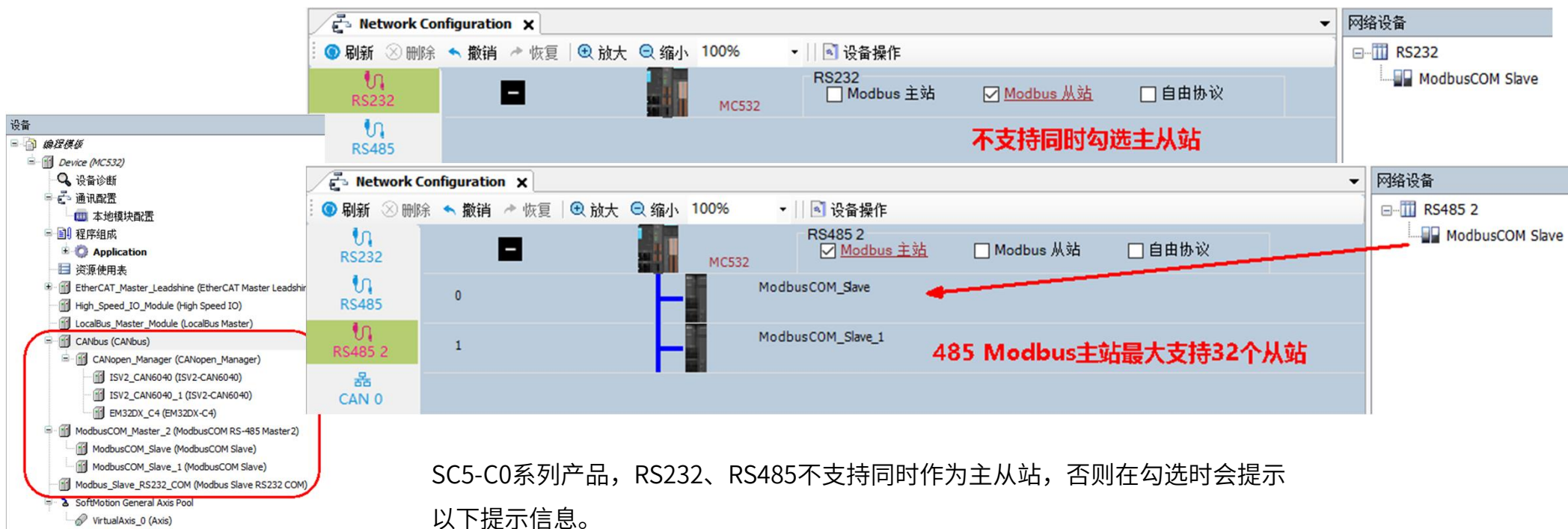
2. 点击EtherCAT A, 勾选EtherCAT主站后, 可在右侧网络设备中拖动/双击伺服, 步进, 耦合器, 添加到EtherCAT从站

3. 添加的总线及设备会显示在设备树下面

Port	Device
0	L7EC_400 Axis
1	L7EC_400_1 Axis_1
2	L7EC_400_2 Axis_2
3	L7EC_400_3 Axis_3
4	L7EC_400_4 Axis_4
5	L7EC_400_5 Axis_5

- 设备树 (Device Tree): EtherCAT_Master_Leadshine (EtherCAT Master Leadshine)
 - L7EC_400 (L7EC-400(COE))
 - Axis (Axis)
 - L7EC_400_1 (L7EC-400(COE))
 - L7EC_400_2 (L7EC-400(COE))
 - L7EC_400_3 (L7EC-400(COE))
 - L7EC_400_4 (L7EC-400(COE))
 - L7EC_400_5 (L7EC-400(COE))
 - High_Speed_IO_Module (High Speed IO)

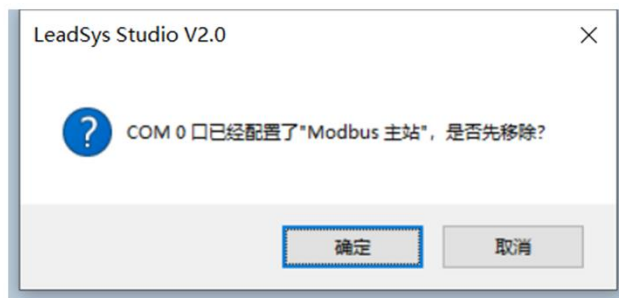
通讯配置---RS232/RS485主从站组态



不支持同时勾选主从站

485 Modbus主站最大支持32个从站

SC5-C0系列产品，RS232、RS485不支持同时作为主从站，否则在勾选时会提示以下提示信息。



通讯配置---EtherNet/EtherNet IP组态



The screenshot displays the 'Network Configuration' software interface. On the left, a tree view shows the device structure, with 'ModbusTCP_Master_Device' and its sub-slaves highlighted by a red circle. The main area shows two configuration windows for 'MC532'.

Top Window: ModbusTCP Configuration

- Device: MC532
- Protocol: ModbusTCP
- Options: ModbusTCP 主站, ModbusTCP 从站
- Table:

Index	Device Name
0	ModbusTCP_SubSlave
1	ModbusTCP_SubSlave_1

Bottom Window: EtherNet/IP Configuration

- Device: MC532
- Protocol: EtherNet IP1
- Options: 扫描器, 适配器
- Table:

Index	Device Name
0	EtherNetIP_Adapter
1	EtherNetIP_Adapter_1
2	EtherNetIP_Adapter_2
3	EtherNetIP_Adapter_3

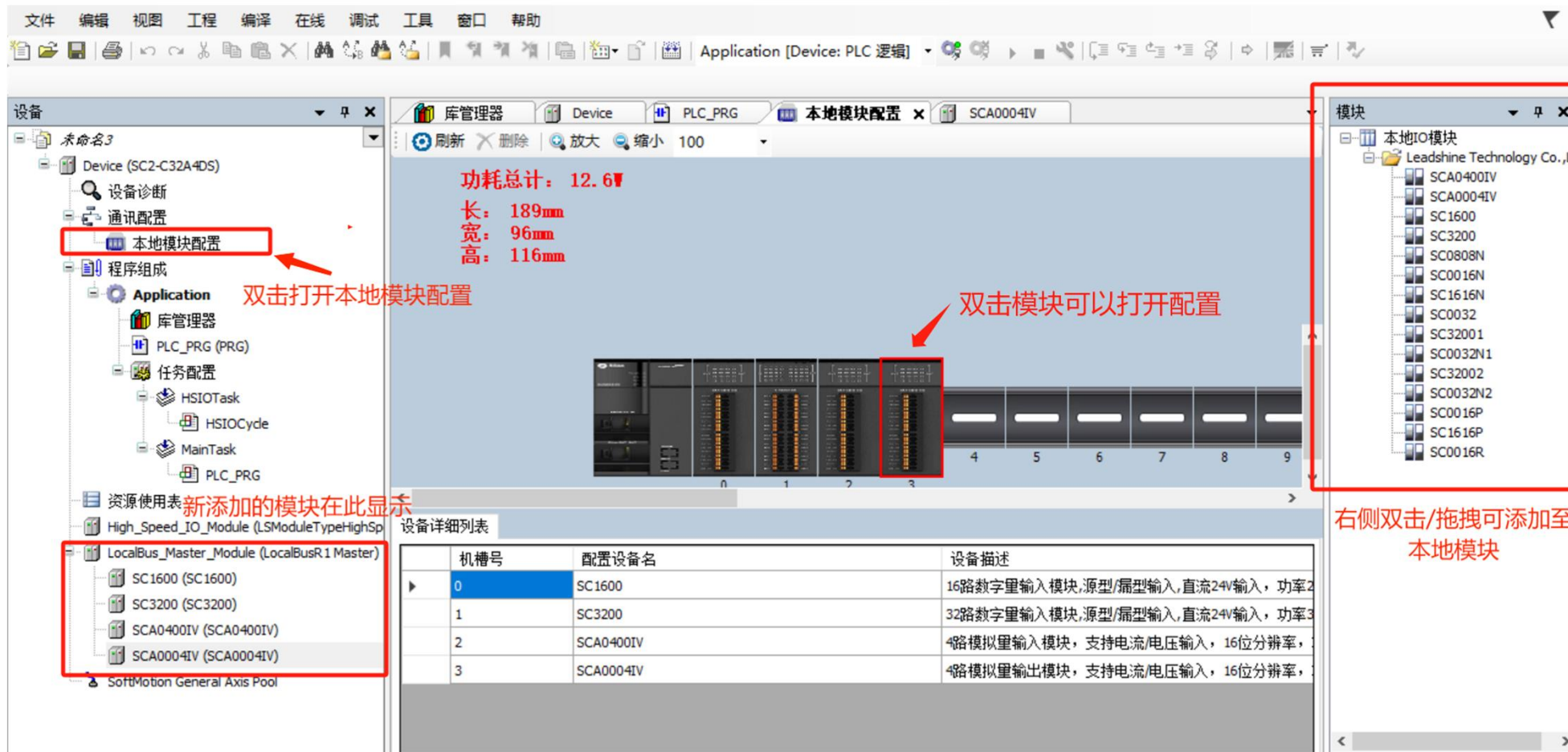
Annotations:

- Red text: **可同时做Modbus TCP主从站** (Can simultaneously be Modbus TCP Master/Slave)
- Red text: **暂不支持同时勾选** (Temporarily does not support simultaneous selection)

Right Panel: Network Devices

- Network Equipment: EtherNet, ModbusTCP SubSlave
- Network Equipment: EtherNet IP1, Generic EtherNet/IP device, EtherNetIP Adapter

本地模块配置---本地模块组态



文件 编辑 视图 工程 编译 在线 调试 工具 窗口 帮助

Application [Device: PLC 逻辑]

设备

库管理器 Device PLC_PRG 本地模块配置 SCA0004IV

刷新 删除 放大 缩小 100

功耗总计: 12.6W
长: 189mm
宽: 96mm
高: 116mm

本地IO模块

- Leadshine Technology Co.,L
- SCA0400IV
- SCA0004IV
- SC1600
- SC3200
- SC0808N
- SC0016N
- SC1616N
- SC0032
- SC32001
- SC0032N1
- SC32002
- SC0032N2
- SC0016P
- SC1616P
- SC0016R

双击打开本地模块配置

双击模块可以打开配置

新添加的模块在此显示

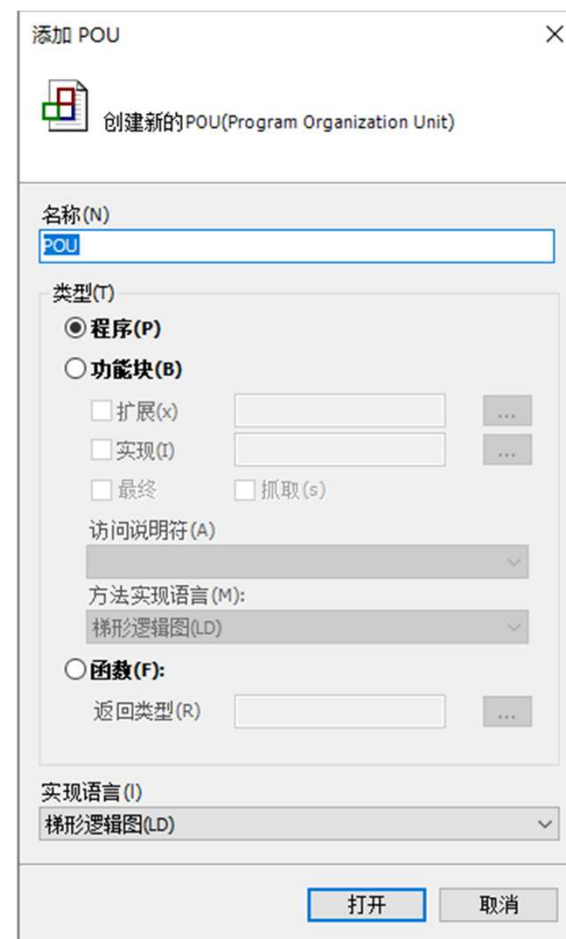
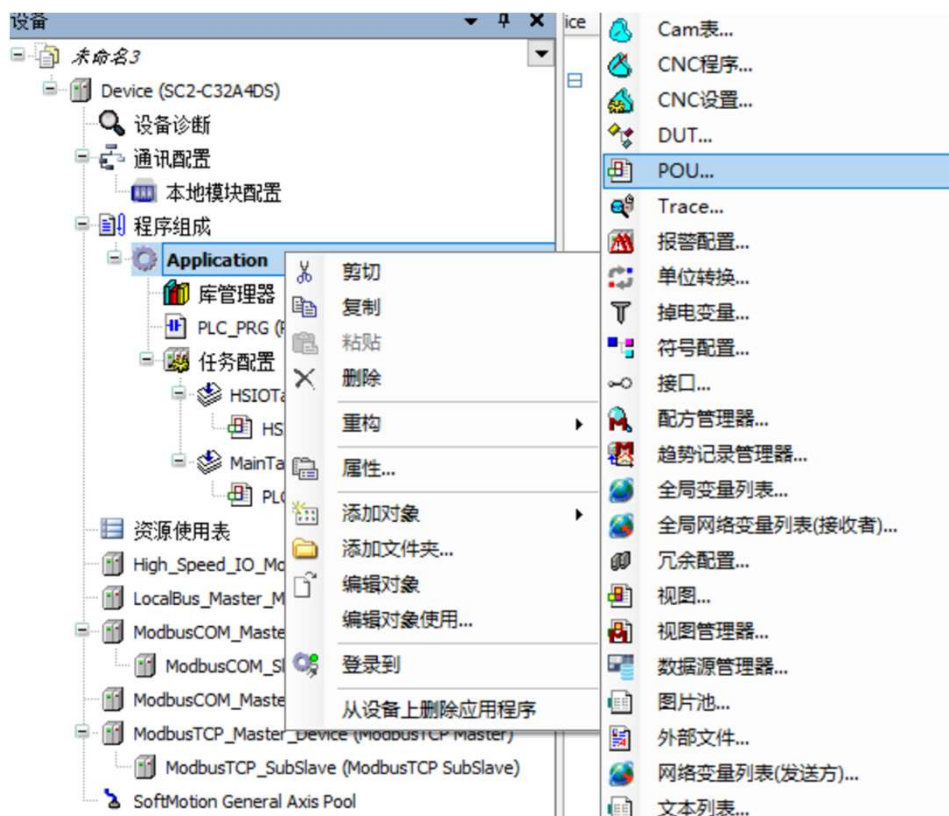
机槽号	配置设备名	设备描述
0	SC1600	16路数字量输入模块,源型/漏型输入,直流24V输入,功率2
1	SC3200	32路数字量输入模块,源型/漏型输入,直流24V输入,功率3
2	SCA0400IV	4路模拟量输入模块,支持电流/电压输入,16位分辨率,
3	SCA0004IV	4路模拟量输出模块,支持电流/电压输入,16位分辨率,

右侧双击/拖拽可添加至本地模块

SC-C系列产品本体右侧支持扩展16个SC系列扩展模块（具体型号见10页的模块说明）

用户程序管理单元---新增POU/FB/FC

右键Application-“添加对象”-“POU”，可添加POU/FB/FC



命名

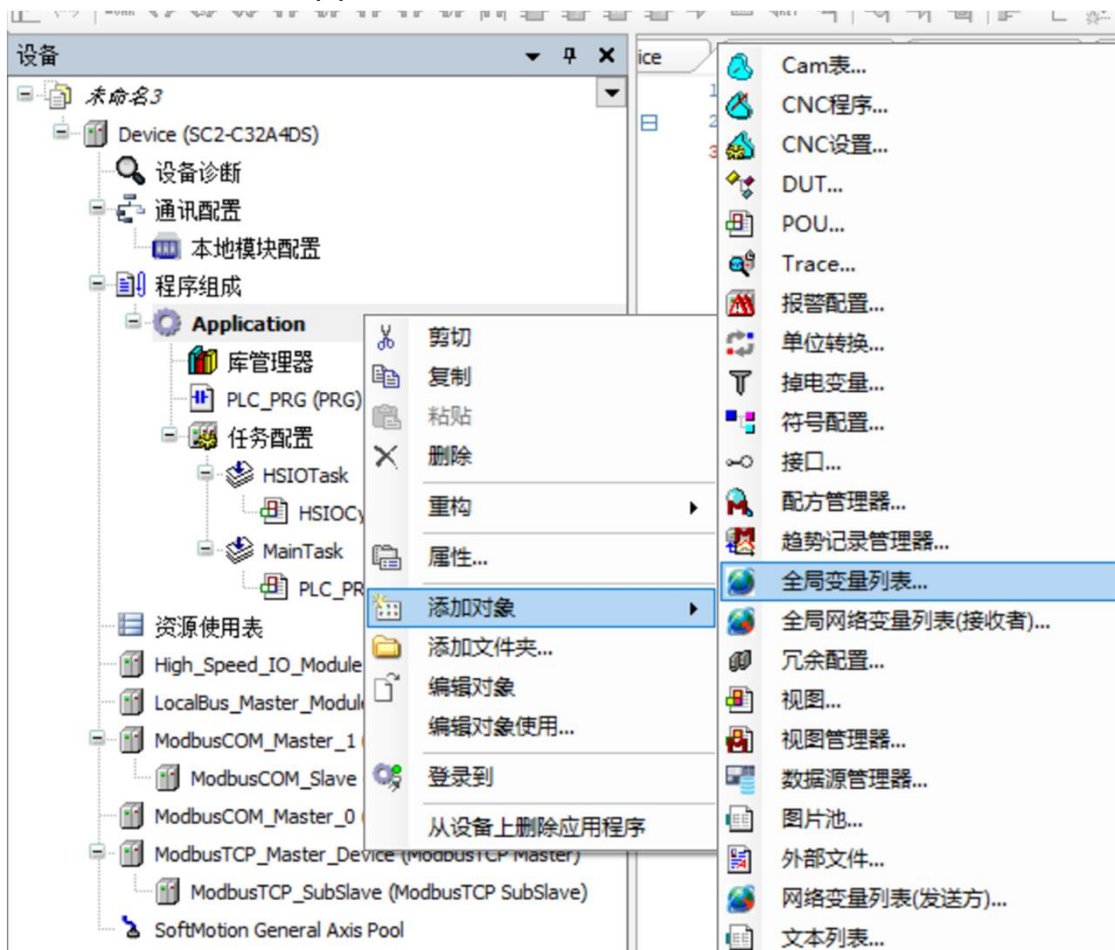
POU

FB功能块

FC

编程语言选择

右键Application-“添加对象”-“全局变量列表”，可添加全局变量表



```
GVL x
1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3   xTest : BOOL;
4 END_VAR
```

上图举例声明了一个Bool型变量xTest，声明之后该变量即可在程序中使用。

其中，首行{attribute 'qualified_only'}可以根据需要选择是否注释(注释使用“//”)。

如果注释，在程序中使用全局变量时，直接调用xTest即可。

如不注释，则在程序中调用时需要添加变量表名，如GVL.xTest。

用户程序管理单元---新增跟踪

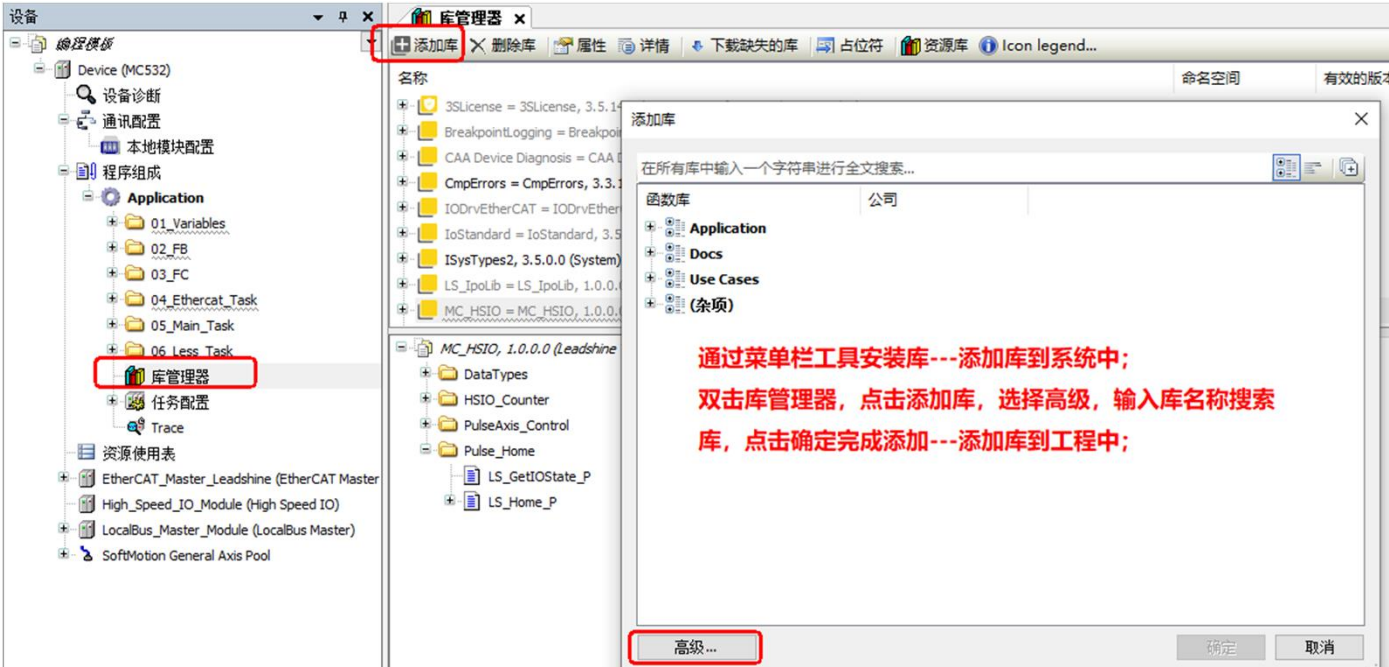
右键Application-
“添加对象” - “跟
踪”，可添加Trace；
在Trace中添加变量
后右键下载跟踪。

注：需要在Trace配
置中关联任务。如
果需要采集轴运动
状态，则关联ECAT
总线任务。

The screenshot displays the LeadSys Studio V2.0 interface. On the left, the '设备' (Device) tree shows the project structure, including 'Application' and various tasks like '01_Variables', '02_FB', '03_FC', '04_Ethercat_Task', '05_Main_Task', and '06_Less_Task'. The main window shows a 'Trace' window with five stacked plots over a 30-second period. The top plot shows a blue signal (Axis.dwPosOffsetForResiduals) with a sawtooth pattern and step changes. The second plot shows a green signal (Axis.fActPosition) with a high-frequency sawtooth. The third plot shows a red signal (Axis.dIActPosition) with a high-frequency sawtooth. The fourth plot shows a grey signal (Axis.dISetPosition) with a linear ramp. The bottom plot shows a blue signal (Axis.fSetPosition) with a high-frequency sawtooth. A red box highlights the '配置' (Configure) button in the top right of the Trace window. A '跟踪配置' (Trace Configuration) dialog is open in the foreground, showing settings for 'Main_Task' and '任务(T)' (Task) selected in a dropdown menu. The dialog includes fields for '触发器变量' (Trigger Variable), '触发器边沿(E)' (Trigger Edge), '后触发器(采样)' (Post-trigger (Sampling)) set to 200ms, '触发器水平(L)' (Trigger Level), and '记录条件' (Recording Conditions). The '分辨率(R)' (Resolution) is set to 'ms' and '自动重启' (Auto-restart) is unchecked. Buttons for '添加变量' (Add Variable), '高级...' (Advanced...), '确定(O)' (OK), and '取消(C)' (Cancel) are visible.

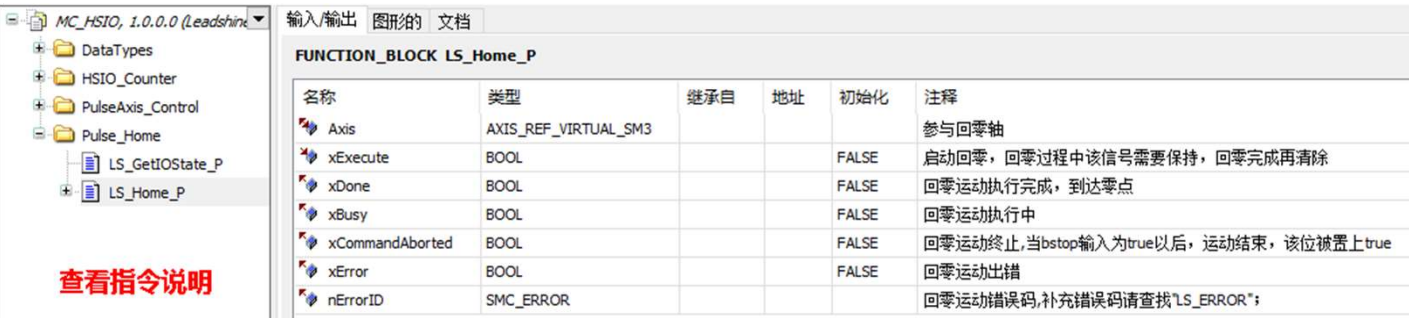


用户程序管理单元---库管理器



通过菜单栏工具安装库---添加库到系统中;
双击库管理器, 点击添加库, 选择高级, 输入库名称搜索库, 点击确定完成添加---添加库到工程中;

库文件一般是由工艺算法或其他编程人员封装好的包含数据结构、函数、功能块等内容的文件, 导入后可直接使用封装好的内容。



查看指令说明

名称	类型	继承自	地址	初始化	注释
Axis	AXIS_REF_VIRTUAL_SM3				参与回零轴
xExecute	BOOL			FALSE	启动回零, 回零过程中该信号需要保持, 回零完成再清除
xDone	BOOL			FALSE	回零运动执行完成, 到达零点
xBusy	BOOL			FALSE	回零运动执行中
xCommandAborted	BOOL			FALSE	回零运动终止, 当bstop输入为true以后, 运动结束, 该位被置上true
xError	BOOL			FALSE	回零运动出错
nErrorID	SMC_ERROR				回零运动错误码, 补充错误码请查找"LS_ERROR";

用户程序管理单元---任务配置

任务界面中，可以设置任务的优先级（0最高）、任务类型、任务周期。优先级越高，任务会优先执行，循环任务可以设定循环周期。一个Application可以有多个任务，一个任务可以执行多个POU。POU只有被添加到任务下，才会执行。



数值越大，优先级越低

优先级 (0..31): 0

类型: 循环 间隔(如t#200ms): 2000 μs

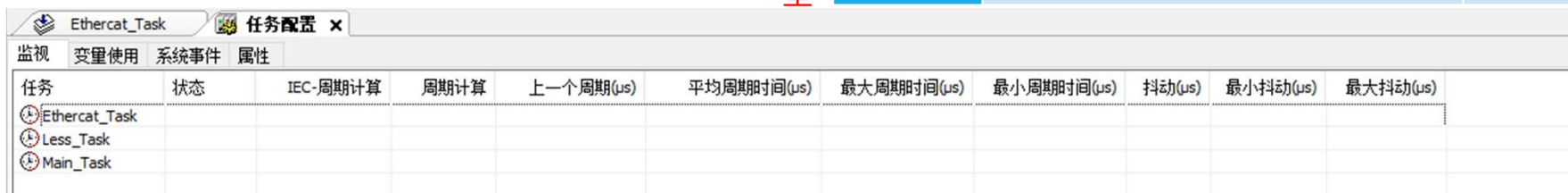
看门狗 超时报错

使能 时间(如t#200ms): t#5000ms 灵敏度: 1

调用该任务执行的POU
运动控制程序调用一定要与对应 EtherCAT 程序调用在同一任务下!

任务类型

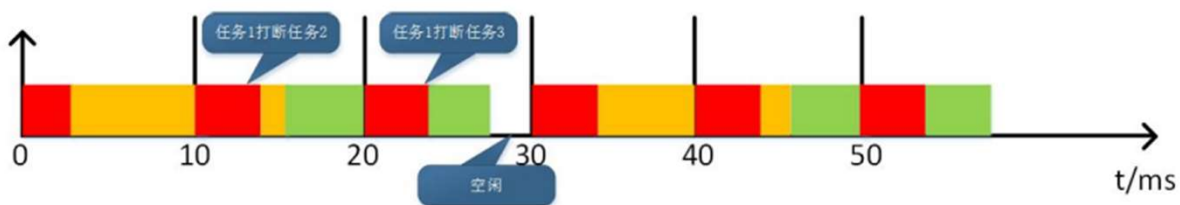
类型	执行逻辑	设置项
循环	按照设定的间隔时间循环执行	时间间隔
事件	关联的变量触发上升沿后，任务开始执行	触发变量
外部的	外部DI输入边沿信号触发	/
惯性滑行	在程序开始时和完整过程结束后以连续循环任务的方式自动开始处理任务	外部DI信号
状态	关联的变量值为True时，任务开始执行	触发变量



任务	状态	IEC-周期计算	周期计算	上一个周期(μs)	平均周期时间(μs)	最大周期时间(μs)	最小周期时间(μs)	抖动(μs)	最小抖动(μs)	最大抖动(μs)
⌚ Ethercat_Task										
⌚ Less_Task										
⌚ Main_Task										

在线模式下，可以在“任务配置”里的“监视”选项卡对任务的状态、周期、抖动 等数据进行监视

- 任务1: 优先级0 循环周期: 10ms
- 任务2: 优先级15 循环周期: 30ms
- 任务3: 优先级20 循环周期: 40ms



任务的时序图

0~10ms: 先执行任务 1, 当任务 1 执行完成后执行任务2;

10~20ms: 任务 1 的循环周期到达, 由于任务 1 的优先级较高, 故直接打断任务 2 的执行, 待任务 1 执行完成后, 继续执行任务 2 剩余的程
序, 待任务 2 也执行完成, 接着执行任务 3;

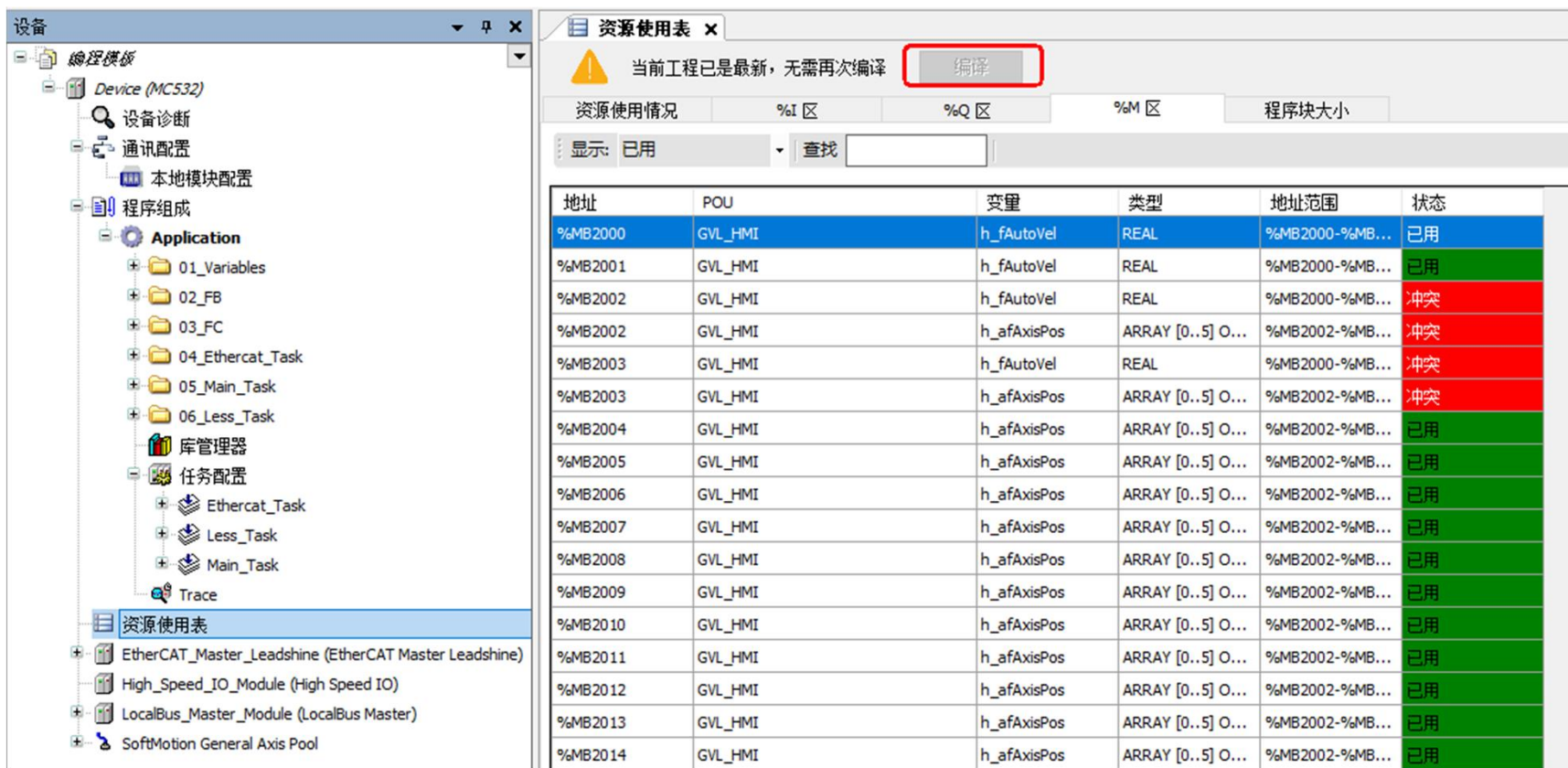
20~30ms: 同上一周期, 由于任务 1 的循环周期已到达, 故任务 1 打断任务 3 执行, 任务 1 执行完成后, 接着执行任务 3 剩余程序, 任务 3 执行完成后, 控制器无其他任务需要执行, 故处于空闲状态;

外部紧急任务: 具备最高优先级, 可以打断优先级0的ECAT总线任务。

资源使用表

双击打开资源使用表，点击编译后，可以查看直接地址是否冲突。

注：地址冲突可能引起数据异常，将冲突的变量地址范围进行更改。



资源使用表

当前工程已是最新，无需再次编译 [编译]

资源使用情况 %I 区 %Q 区 %M 区 程序块大小

显示: 已用 查找

地址	POU	变量	类型	地址范围	状态
%MB2000	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	已用
%MB2001	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	已用
%MB2002	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	冲突
%MB2002	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	冲突
%MB2003	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	冲突
%MB2003	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	冲突
%MB2004	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2005	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2006	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2007	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2008	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2009	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2010	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2011	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2012	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2013	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2014	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用

LocalBus_Master_Module---背板扫描

The screenshot displays the software's device tree on the left, where the 'LocalBus_Master_Module (LocalBus Master)' is highlighted with a red box. A context menu is open over this item, with '背板扫描' (Backplane Scan) selected. The main window shows the configuration for the 'LocalBus_Master_Module' and 'PM0016'. A '扫描设备' (Scan Devices) dialog box is overlaid, featuring a table with columns for '设备名' (Device Name) and '设备类型' (Device Type). The dialog includes a '扫描设备' button, a '复制所有设备到工程中' button, and a '关闭' button. A red text box in the dialog provides instructions: '右键 LocalBus_Master_Module, 选择背板扫描, 复制所有设备到工程中, 自动完成组态。' (Right-click LocalBus_Master_Module, select Backplane Scan, copy all devices to the project, and automatically complete the configuration.)

除手动配置外，还可通过背板扫描，将设备复制到工程中

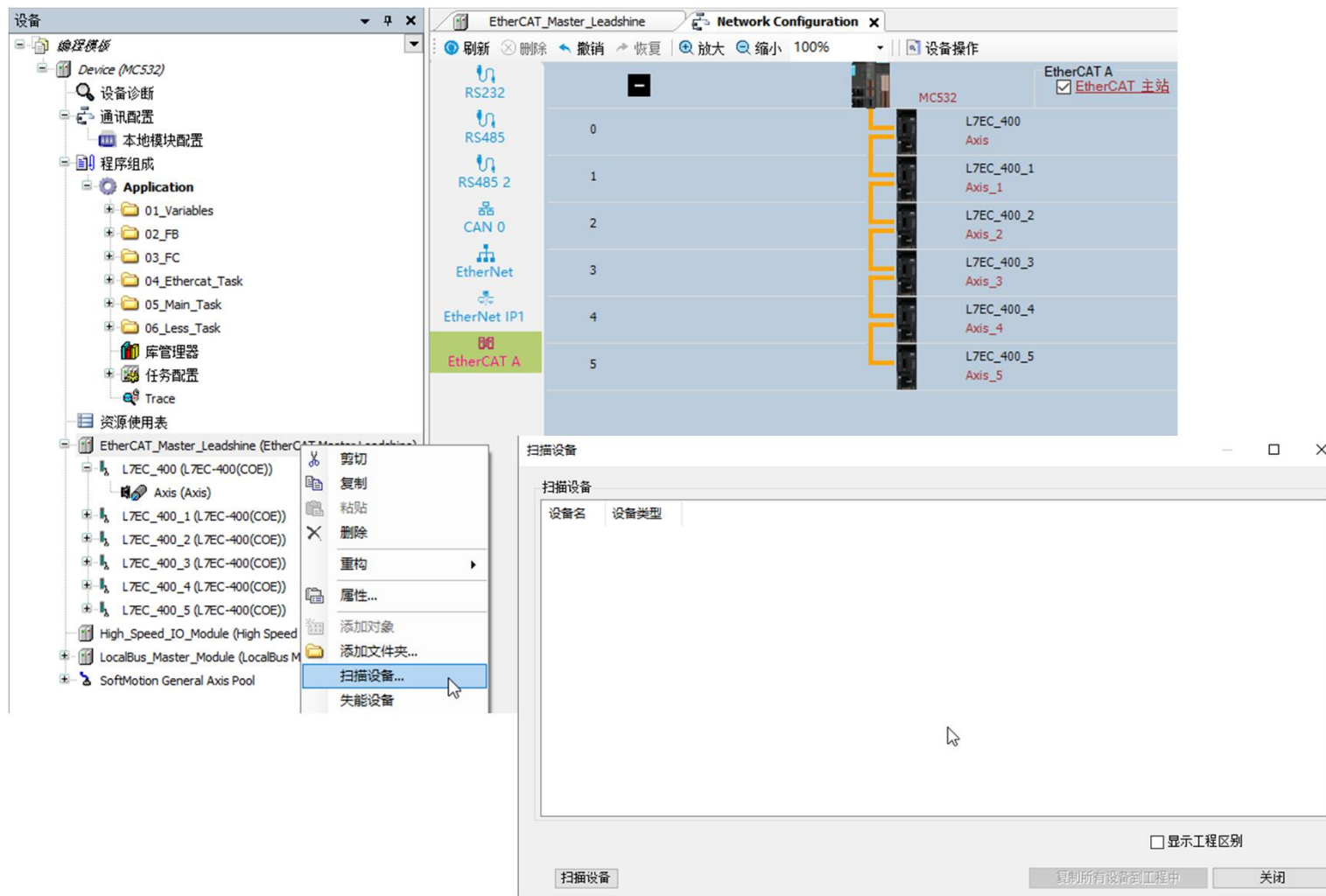
右键 LocalBus_Master_Module, 选择背板扫描, 复制所有设备到工程中, 自动完成组态。

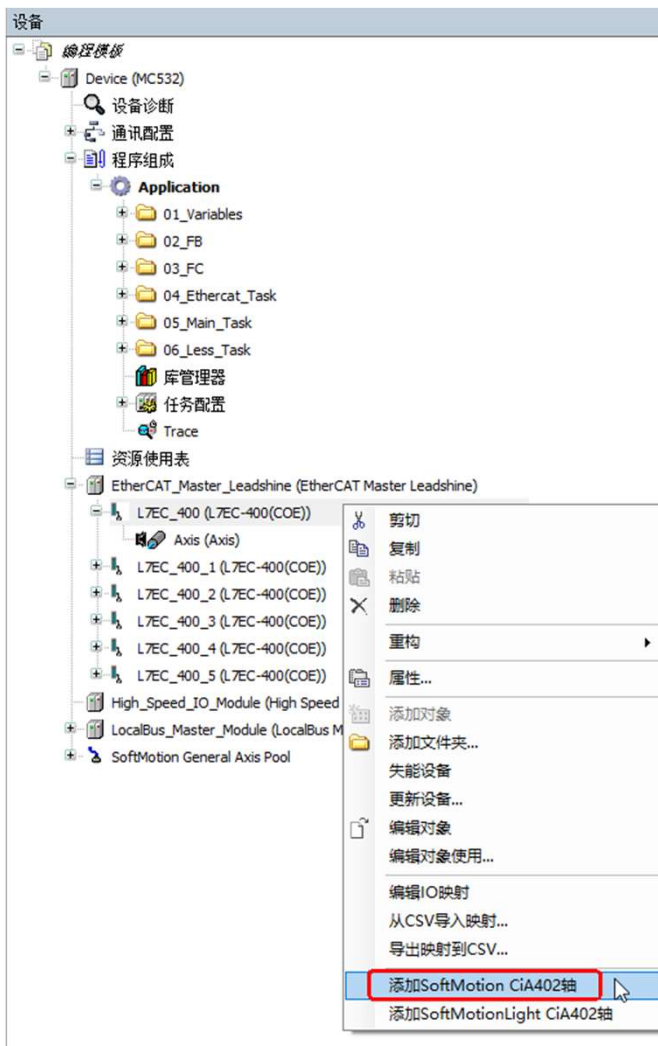
EtherCAT配置---扫描设备

右键

EtherCAT_Master_Leadshine_A, 扫描设备, 复制所有设备到工程中, 自动完成组态。

注: PLC 和 EtherCAT 设备上电, 且 EtherCAT 总线已将伺服电机、IO 模块等 EtherCAT 设备硬件连接正常





右键选择没有 402 轴的伺服电机，如选择“L7EC_400”，点击右键，添加“softMotion CiA402 轴”。

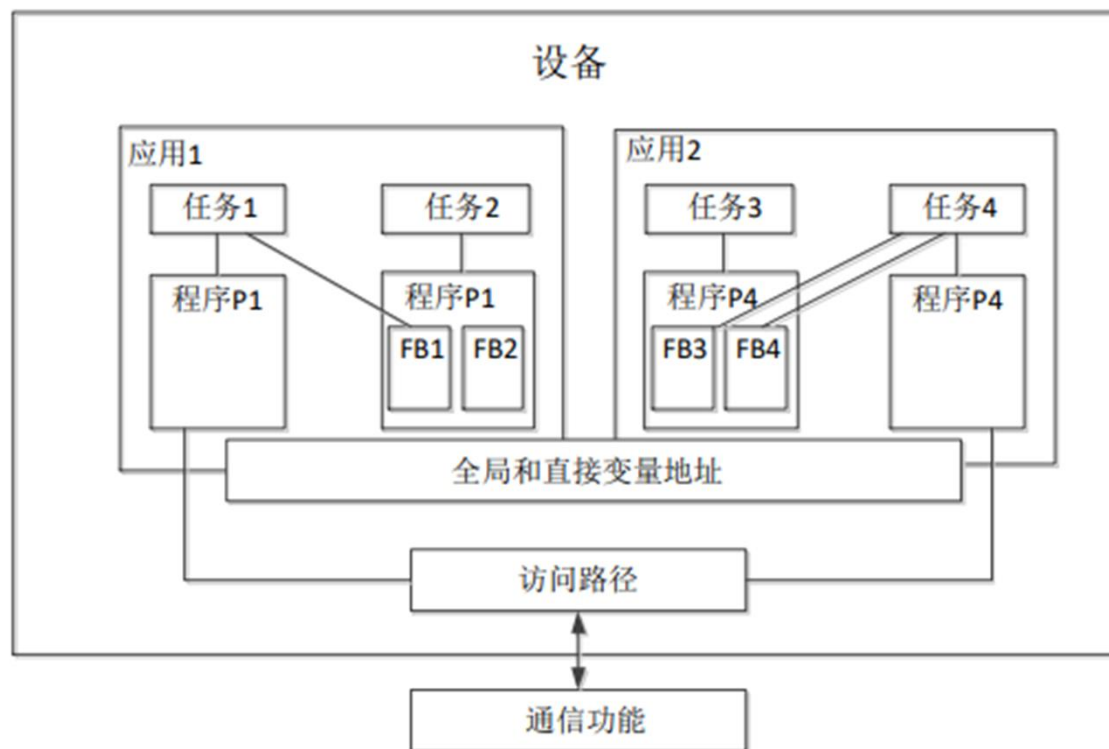
注：

- 1.添加雷赛伺服从站不需要手动加轴，第三方伺服从站需要手动添加402轴；
2. 402 轴默认自动命名为 Axis、Axis_1、Axis_2 等轴变量名，用户可根据需要更改，支持中文变量名；

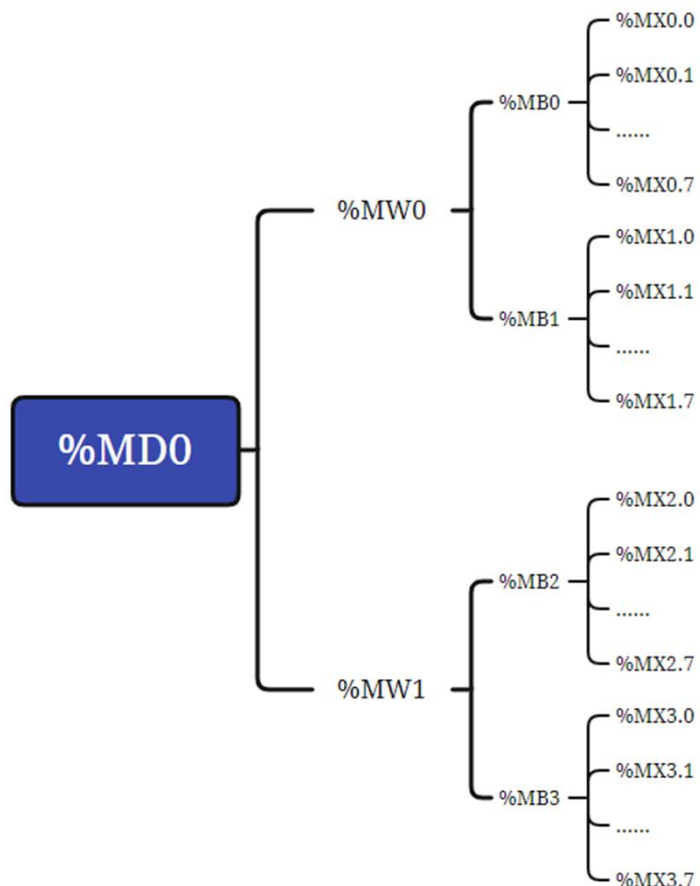
03

编程基础知识

软件模型描述了基本的软件元素相互关系，这些软件元素包含设备、应用、任务、全局变量、访问路径和应用对象，它们是现代 PLC 的软件基础，其内部结构如下图所示。



SC-C0系列产品的直接地址分为I（输入）、Q（输出）、M（存储区）三个区域，下面以M区的MD0为例子，说明地址分配的方式：



D: Double Word (32位)

W: Word (16位)

B: Byte (8位)

X: Bit (1位)

注意：在使用I、Q区域时，尽量空出前面的地址，因为添加设备（比如：EtherCAT从站）时会默认按照顺序编址，容易出现地址冲突的情况。

使用语法：%+地址区域+地址大小前缀+序号.<位号>

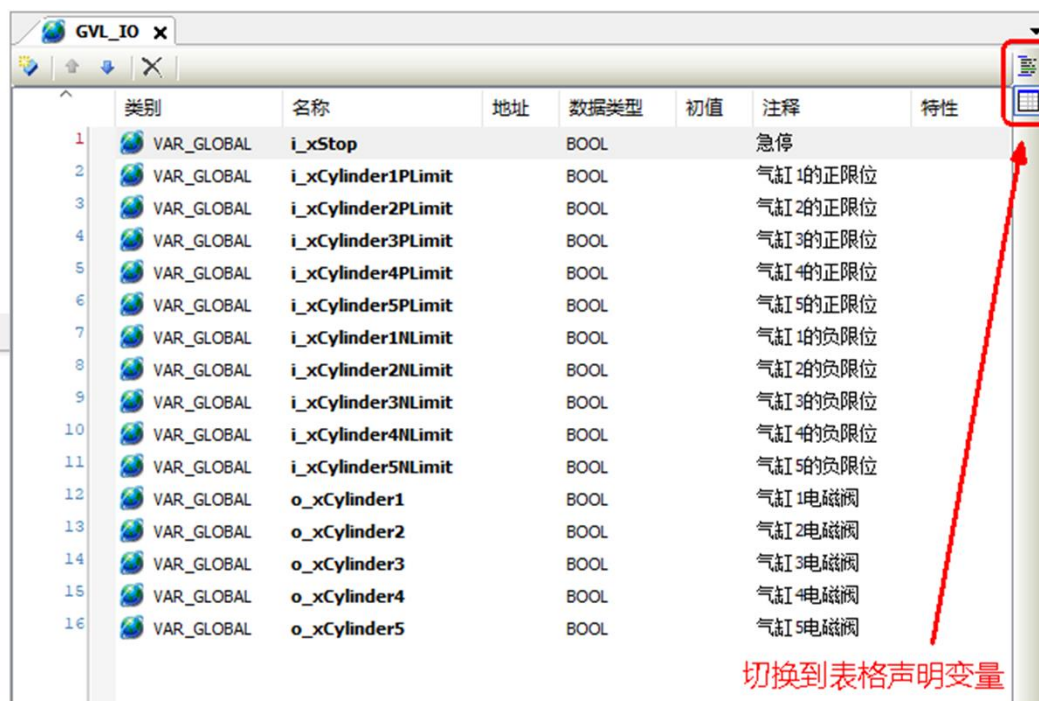
区域	大小	地址范围
I区 (128KByte)	64KWords	IW0~IW65535
Q区 (128KByte)	64KWords	QW0~QW65535
M区 (512KByte)	256KWords	MW0~MW262143

变量分为局部变量和全局变量，局部变量在POU/FB/FC中定义，全局变量在全局变量表中定义，局部变量只能在对应的POU或功能块内部使用，全局变量则可以在所有POU中使用：

变量可由用户自己定义，定义的语法为：**变量名：数据类型；**

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3     xTest : BOOL;
4     iTTest : INT;
5     diTest : DINT;
6     byTest : BYTE;
7 END_VAR
8
```

局部变量定义



类别	名称	地址	数据类型	初值	注释	特性
VAR_GLOBAL	i_xStop		BOOL		急停	
VAR_GLOBAL	i_xCylinder1PLimit		BOOL		气缸1的正限位	
VAR_GLOBAL	i_xCylinder2PLimit		BOOL		气缸2的正限位	
VAR_GLOBAL	i_xCylinder3PLimit		BOOL		气缸3的正限位	
VAR_GLOBAL	i_xCylinder4PLimit		BOOL		气缸4的正限位	
VAR_GLOBAL	i_xCylinder5PLimit		BOOL		气缸5的正限位	
VAR_GLOBAL	i_xCylinder1NLimit		BOOL		气缸1的负限位	
VAR_GLOBAL	i_xCylinder2NLimit		BOOL		气缸2的负限位	
VAR_GLOBAL	i_xCylinder3NLimit		BOOL		气缸3的负限位	
VAR_GLOBAL	i_xCylinder4NLimit		BOOL		气缸4的负限位	
VAR_GLOBAL	i_xCylinder5NLimit		BOOL		气缸5的负限位	
VAR_GLOBAL	o_xCylinder1		BOOL		气缸1电磁阀	
VAR_GLOBAL	o_xCylinder2		BOOL		气缸2电磁阀	
VAR_GLOBAL	o_xCylinder3		BOOL		气缸3电磁阀	
VAR_GLOBAL	o_xCylinder4		BOOL		气缸4电磁阀	
VAR_GLOBAL	o_xCylinder5		BOOL		气缸5电磁阀	

切换到表格声明变量

全局变量定义

变量关联---直接地址

变量关联直接地址，语法为：变量名+AT+%+地址区域+地址大小前缀+序号.<位号>

如：iTest AT %MW0 :int; //定义int型变量iTest关联直接地址MW0

The screenshot displays two windows from a PLC programming environment. The left window, titled 'GVL_IO', shows a table of global variables. The right window, titled 'PLC_PRG', shows a program with local variable declarations.

类别	名称	地址	数据类型	初值	注释
VAR_GLOBAL	i_xStop		BOOL		急停
VAR_GLOBAL	i_xCylinder1PLimit	%QX2.0	BOOL		气缸1的正限位
VAR_GLOBAL	i_xCylinder2PLimit	%QX2.1	BOOL		气缸2的正限位
VAR_GLOBAL	i_xCylinder3PLimit	%QX2.2	BOOL		气缸3的正限位
VAR_GLOBAL	i_xCylinder4PLimit	%QX2.3	BOOL		气缸4的正限位
VAR_GLOBAL	i_xCylinder5PLimit		BOOL		气缸5的正限位
VAR_GLOBAL	i_xCylinder1NLimit		BOOL		气缸1的负限位
VAR_GLOBAL	i_xCylinder2NLimit		BOOL		气缸2的负限位
VAR_GLOBAL	i_xCylinder3NLimit		BOOL		气缸3的负限位
VAR_GLOBAL	i_xCylinder4NLimit		BOOL		气缸4的负限位
VAR_GLOBAL	i_xCylinder5NLimit		BOOL		气缸5的负限位
VAR_GLOBAL	o_xCylinder1		BOOL		气缸1电磁阀
VAR_GLOBAL	o_xCylinder2		BOOL		气缸2电磁阀
VAR_GLOBAL	o_xCylinder3		BOOL		气缸3电磁阀
VAR_GLOBAL	o_xCylinder4		BOOL		气缸4电磁阀
VAR_GLOBAL	o_xCylinder5		BOOL		气缸5电磁阀

```
PROGRAM PLC_PRG
VAR
    xTest AT%QX3.0 : BOOL ;
    iTest AT%MW10 : INT ;
    diTest AT%MD10 : DINT ;
    byTest AT%MB10 : BYTE ;
END_VAR
```

上图中xTest属于局部变量，且关联直接地址%QX3.0，在其他POU中使用该变量时需要加上程序名称，即PLC_PRG.xText

变量关联---IO

变量关联IO，可直接修改IO映射中的设置，关联后数据直接写入到对应的变量中，不再占用绝对地址；

本地IO、
远程IO皆
可采用IO
映射方式，
程序中直
接使用变
量编程。

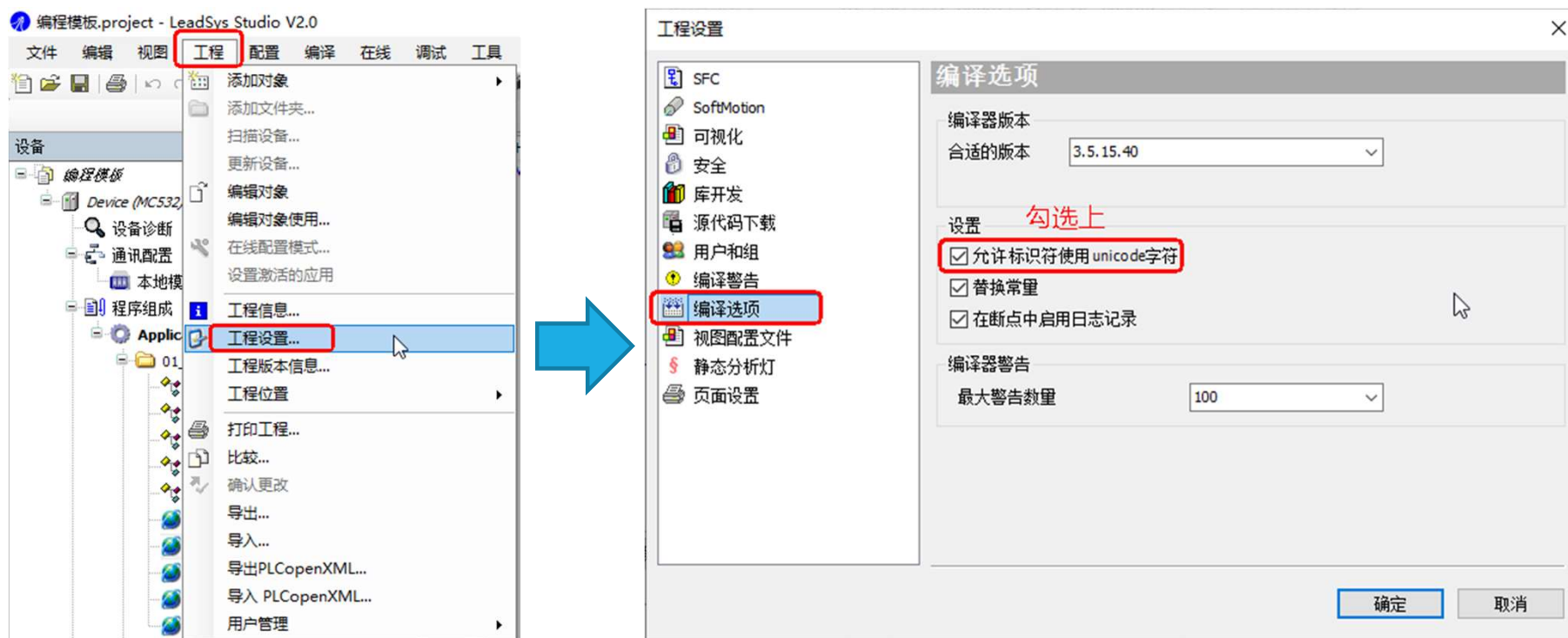
添加IO映射

变量	映射	通道	地址	类型	单元	描述
Application.h_axJogP.0		Bit0	%QX76.0	BOOL		
		Bit1	%QX76.1	BOOL		
		Bit2	%QX76.2	BOOL		
		Bit3	%QX76.3	BOOL		
		Bit4	%QX76.4	BOOL		
		Bit5	%QX76.5	BOOL		
		Bit6	%QX76.6	BOOL		
		Bit7	%QX76.7	BOOL		
		Bit8	%QX77.0	BOOL		
		Bit9	%QX77.1	BOOL		
		Bit10	%QX77.2	BOOL		
		Bit11	%QX77.3	BOOL		
		Bit12	%QX77.4	BOOL		
		Bit13	%QX77.5	BOOL		
		Bit14	%QX77.6	BOOL		
		Bit15	%QX77.7	BOOL		

默认使能1

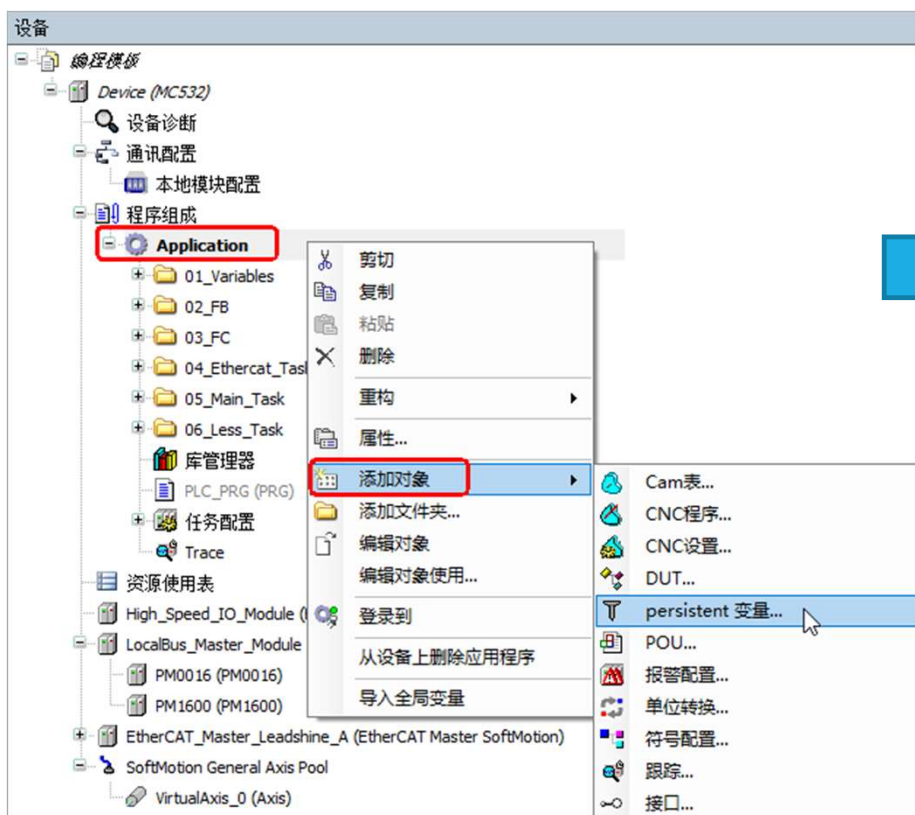
使能1(如果未在任何任务中使用则使用总线循环在)

菜单栏-工程-工程设置，选择编译选项，勾选“允许标识符使用unicode字符”



注意：OPU UA/标签通讯不支持中文变量。

右键Application- “添加对象” - “Persistent变量”，可添加保持型变量表



```
PersistentVars x
1  //{attribute 'qualified_only'}
2  VAR_GLOBAL PERSISTENT RETAIN
3      iTTest : BOOL ;
4      fTest : REAL ;
5  END_VAR
```

上图中，首行{attribute 'qualified_only'}可以根据需要选择是否注释(注释使用“//”)。

如果注释，则在程序中使用直接调用iTTest即可。

如不注释，则在程序中调用时需要添加变量表名，如 PersistentVars.iTest。

注意：I区/Q区无掉电保持，仅M区有掉电保持

```

PersistentVars x
1  //{attribute 'qualified_only'}
2  VAR_GLOBAL PERSISTENT RETAIN
3      GVL.iTest : BOOL ;
4      GVL.fTest : REAL ;
5  END_VAR

GVL
1  //{attribute 'qualified_only'}
2  VAR_GLOBAL PERSISTENT RETAIN
3      iTTest AT%MX10.2: BOOL ;
4      fTest AT%MD2: REAL ;
5  END_VAR
    
```

在线命令	VAR	VAR_RETAIN	VAR PERSISTENT RETAIN
掉电	×	√	√
热复位	×	√	√
冷复位	×	×	√
初始值复位	×	×	×
程序下载	×	×	√
在线修改	√	√	√

注意：保持型变量表中不支持变量关联直接地址，先在全局变量表中声明变量并关联直接地址，再在保持型变量表中调用。

```

GVL x
1  //{attribute 'qualified_only'}
2  VAR_GLOBAL PERSISTENT RETAIN
3  END_VAR
4  VAR_GLOBAL CONSTANT
5      diTest: DINT := 100 ;
6  END_VAR
7
8
    
```

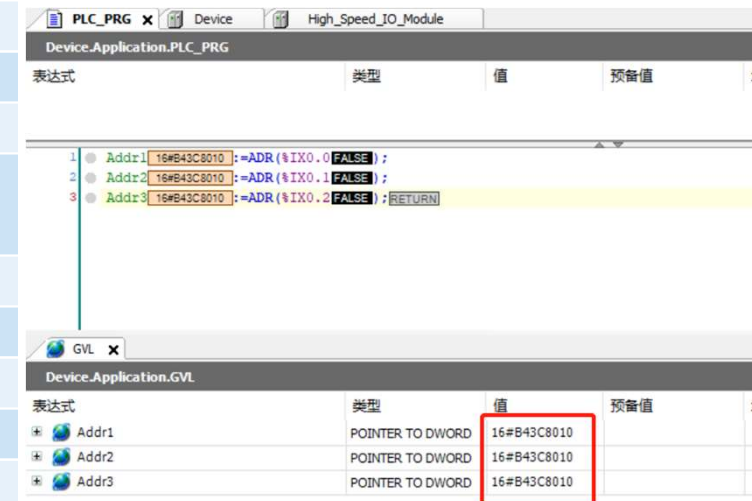
常量关键字：CONSTANT

数据类型——标准

标准数据类型如左图所示：

注意：布尔类型默认占用内存1个BYTE，即系统最小单位是BYTE。比如只用一个%IX3.0，系统会自动分配%IX3.0~3.7。用指针取输入点地址时，获取的地址值都一样。

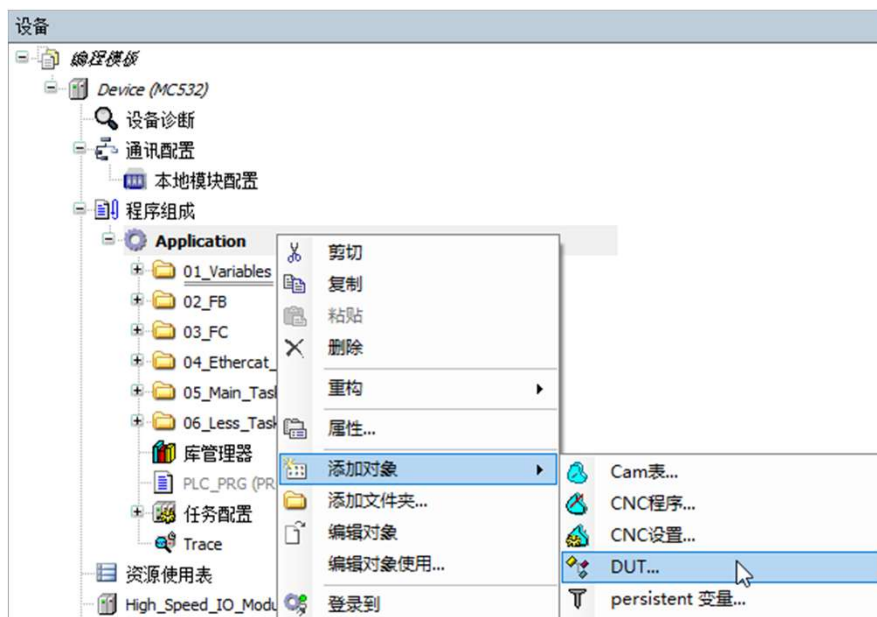
数据类型	关键字	占用内存	取值范围
位	BIT	1bit	0和1，仅定义结构体和功能库内部可使用
布尔	BOOL	1Byte	FALSE(0)或TRUE(1)
整型	BYTE	8bit	0~255
	WORD	16bit	0~65535
	DWORD	32bit	0~4294967295
	LWORD	64bit	0~(264-1)
	SINT	8bit	-128~127
	USINT	8bit	0~255
	INT	16bit	-32768~32767
	UINT	16bit	0~65535
	DINT	32bit	-2147483648~2147483647
	UDINT	32bit	0~4294967295
	LINT	64bit	-263~(263-1)
实数	REAL	32bit	1.175494351e-38~3.402823466e+38
	LREAL	64bit	2.2250738585072014e-308~1.7976931348623158e+308
字符串	STRING	8×N bit	
	WSTRING	16×N bit	
时间	TIME	32bit	T#0ms~T#71582m47s295ms
	TIME_OF_DAY		TOD#0:0:0~TOD#1193:02:47.295
	DATE		D#1970-1-1~D#2106-02-06
	DATE_AND_TIME		DT#1970-1-1-0:0:0~DT#2106-02-06-06:28:15



右键Application- “添加对象” - “DUT”，可添加结构体/枚举体/联合体

注意：是POU中调用时需要
进行实例化

数据类型	关键字	占用内存	取值范围
结构体	STRUCT	根据定义	根据定义
枚举	ENUM	根据定义	根据定义
联合	UNION	根据定义	根据定义
指针	POINTER TO	根据定义	根据定义
数组	ARRAY [?.?] OF	根据定义	根据定义



```
St_PointAxis x Enum_AxisName Union_Example
1 TYPE St_PointAxis ://轴结构体参数
2 STRUCT
3     //输入
4     i_iAxisMode      : INT;//轴模式 默认0 Ethercat,1:CanOpen,2:脉冲
5     i_iGoAbs_dir     : INT;//轴运动方向
6     i_xReset         : BOOL;//轴复位
7     i_xSerOn         :
8 END_STRUCT
9 END_TYPE
```

结构体

```
St_PointAxis Enum_AxisName x Union_Example
1 TYPE Enum_AxisName : //轴枚举举例
2 (
3     A轴 := 0,
4     B轴 := 1,
5     C轴 := 2,
6     D轴 := 3,
7     E轴 := 4,
8     F轴 := 5,
9     虚拟轴:=6
10 );
11 END_TYPE
```

枚举体

```
St_PointAxis Enum_AxisName Union_Example x
1 TYPE Union_Example ://数据的联合, 里面的成员共用一个内存地址
2 UNION
3     byxxx :BYTE;
4     xxxxx :BOOL;
5     ixxx  :INT;
6 END_UNION
7 END_TYPE
```

联合体

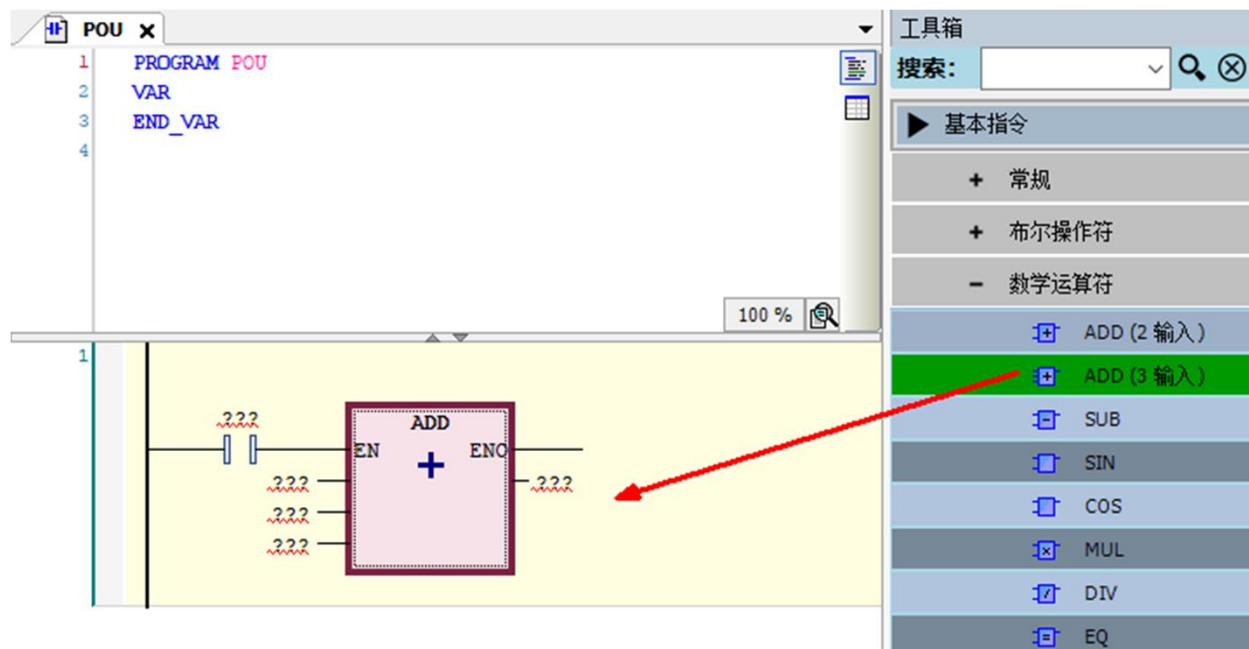
```
GVL x
1 //{attribute 'qualified_only'}
2 VAR_GLOBAL
3     pbyData : POINTER TO ARRAY [0..9] OF BYTE ; //由10个BYTE组成数组的地址
4     Data : ARRAY [0..9] OF BYTE ; //数组, 数组里有10个BYTE
5 END_VAR
```

数组：同一类型的N个数据组成的集合（一维/二维/三维）

指针：变量的内存地址，pbyData^：取指针指向的变量值，ADR（Data）：取变量地址



LeadSys Studio 软件中的梯形图可以通过拖拽直接完成编程，也可以通过键盘快捷键输入，各功能指令的快捷键可以在菜单栏 LBD/LD/IL选项卡中查找。



操作符	执行的操作	优先级
(表达式)	括号	高 ↑ 低
函数名 (参数列表, 由逗号分隔)	调用函数	
EXPT	指数运算	
-, NOT	取反	
*	乘	
/	除	
MOD	取余数	
+	加	
-	减	
<, >, <=, >=	比较运算	
=	等于	
<>	不等于	
AND	与	
XOR	异或	
OR	或	

表达式中包括操作符和操作数，操作数按照操作符指定的优先级规则进行运算，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。例如： $a+b+c+d/2*3.14*R$

如果在表达式中有若干个操作符，则操作符会按照规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。

赋值指令用作将变量、表达式或FUN/FB的值赋给另一个变量。

语法：

变量A := 变量、表达式或FUN/FB的值；将操作符“:=”右边变量、表达式或FUN/FB的值赋给左边的变量A。

例程：

V1的值加1，再将得到的结果赋给V2

```
V2:=V1+1;
```

使用IF指令可以检查条件，并根据此条件执行相应的指令。

语法：

常用的IF指令结构有以下3种：

(1) //当条件A满足时，执行语句A

```
IF 条件A THEN  
    语句A;  
END_IF
```

(2) //当条件A满足时，执行语句A;否则，执行语句B

```
IF 条件A THEN  
    语句A;  
ELSE  
    语句B;  
END_IF
```

(3) //当条件A满足时，执行语句A;

```
IF 条件A THEN  
    语句A;  
ELSIF 条件B THEN //当条件B满足时，执行语句B;  
    语句B;  
...  
ELSIF 条件N-1 THEN //当条件N-1满足时，执行语句N-1;  
    语句N-1;  
ELSE //如果以上条件都不满足，则执行语句N;  
    语句N;  
END_IF //指令结束
```

例程：

若变量a的值小于0，则变量b的值为0;
若变量a的值为0，则变量b的值为1;
若变量a 的值不在上述范围内，则变量b的值为2。

```
IF a<0 THEN  
    b:=0;  
ELSIF a=0 THEN  
    b:=1;  
ELSE  
    b:=2;  
END_IF
```

CASE指令用于将控制变量和若干个操作数进行比较，再根据控制变量的值，从多个语句中选择要执行的语句。

语法：

CASE 控制变量 OF

选择值:

语句;

选择值:

语句;

...

ELSE

语句;

END_CASE

- 如果控制变量与其中一个值相同，则执行该值对应的语句;
- 如果与任何一个值都不相同，则执行ELSE指令的语句。

例程：

若变量a = 1或2时，则变量b的值为10。若变量a的值为3，则变量b的值为20。

若变量a 的值不在上述范围内，则变量b的值为30。

CASE a OF

1,2:

b:=10;

3:

b:=20;

ELSE

b:=30;

END_CASE

FOR循环指令是有限制的循环指令,当限制条件满足(变量值等于“循环结束时变量值”)时,程序就将退出FOR循环,执行下一条指令。

语法:

```
FOR 控制变量 := 循环起始值 TO 循环结束值 { BY 递增步长} DO
```

```
    语句;
```

```
END_FOR
```

其中, {BY 递增步长}内语句可根据需要省略,省略时步长默认为1。若要中断循环指令,请执行EXIT指令。

例程:

循环控制变量为Counter,循环开始时控制变量初值为1,每一次循环Counter+1;当Counter等于5时,执行完FOR循环内容后,退出循环,执行下一条语句。

语句Var1:=Var1*2一共执行5次;若Var1的初始值是1,那么循环结束后,Var1的值为32。

```
FOR Counter:=1 TO 5 BY 1 DO
```

```
    Var1:=Var1*2;
```

```
END_FOR
```

若上例中所使用的计数变量Counter的类型是SINT(范围从-128到127),如果语句为“FOR Counter:=1 TO 127 BY 1 DO”则会进入死循环。编程时应避免此类情况的发生。

WHILE循环的结束条件不是指定的循环次数，而是任意的逻辑表达式。当该表达式叙述的条件满足时，执行循环。

语法：

WHILE 循环条件 DO

 语句;

END_WHILE

WHILE循环执行前先检查是否为TRUE，如果为TRUE，则执行；

当执行完一次后，再次检查，如果仍为TRUE，则再次执行，直到为FALSE。

如果一开始就为FALSE，则不会执行WHILE循环里的指令。

若要中断循环处理，请执行EXIT指令。此时，将不执行从EXIT指令到END_WHILE指令之间的处理。

例程：

变量Counter大于0，则一直会执行WHILE循环中的指令，直到Counter小于等于0为止。每执行一次循环，通过指令“Counter := Counter-1”使Counter的值减1，当Counter小于等于0时，循环结束。

```
WHILE Counter>0 DO
```

```
    Counter := Counter-1;
```

```
END_WHILE
```

REPEAT循环在指令执行以后，才检查结束条件。因此无论结束条件怎样，循环至少执行一次。

语法：

REPEAT

语句;

UNTIL 循环结束条件

END_REPEAT

语句一直执行，直到为TRUE时，REPEAT循环结束。如果一开始就为TRUE，则循环只执行一次。

例程：

变量Counter大于0，则一直会执行 REPEAT 循环中的指令，直到Counter小于等于0为止。每执行一次循环，通过指令“Counter := Counter-1”使Counter的值减1，当Counter小于等于0时，循环结束。

REPEAT

 Counter := Counter-1;

UNTIL Counter < 0

END_REPEAT。

调用功能块：

语法：

<Name of FB instance>

(FB input variable := ,

further FB input variable := ,

… ,

FB output variables => ,

further FB output variables => , …);

例程：

在ST中调用TON定时器，假设其实例名TON_1：

TON_1

(IN := btrue ,

PT := T#500ms ,

Q => T1_OUT ,

ET => T1_ET);

ST中的注释：

语法：

ST语言中，有两种注释方法：

(1) 注释以 (*) 开始，以 *) 结束。这种注释方法允许多行注释。

例如：

```
(* a:=inst.out; (* to be checked *) b:=b+1; *)
```

(2) 注释以 “//” 开始，一直到本行结束。这是单行注释的方法。

例如：

```
// This is a comment
```


指令表IL

与汇编语言类似，是一种助记符编程语言。

涉及到算法部分的程序请选择**ST**语言，编写的程序往往简洁而高效；
涉及到逻辑控制部分的程序，请选择**LD**语言，编写的联锁、互锁等功能逻辑简单易懂；

功能块图FBD

涉及到功能块部分的程序，请选择**FBD**，编写的程序会形成一个脉络清晰的网状电路图，容易读懂。

连续功能图CFC

涉及到功能块部分的程序，请选择**CFC**，编写的程序会形成一个脉络清晰的网状电路图，容易读懂。

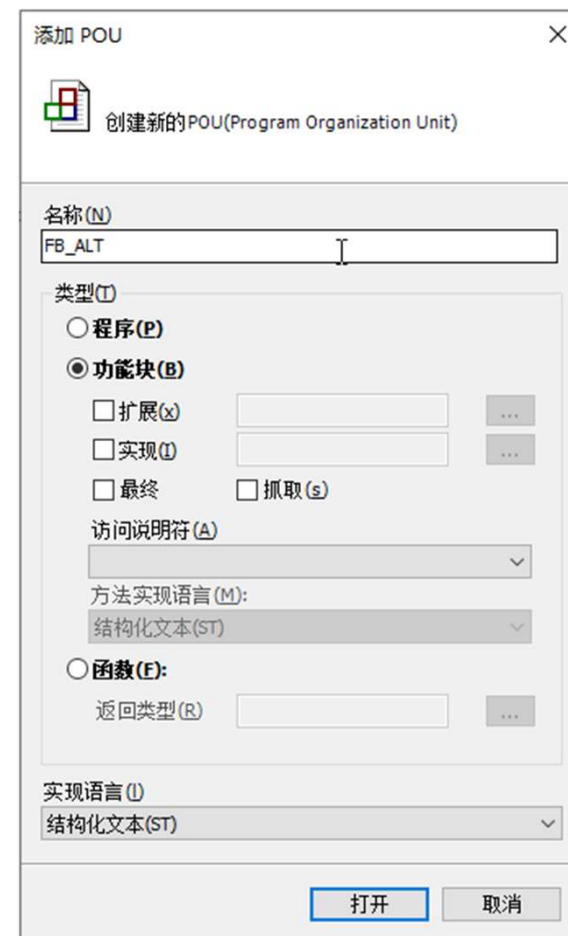
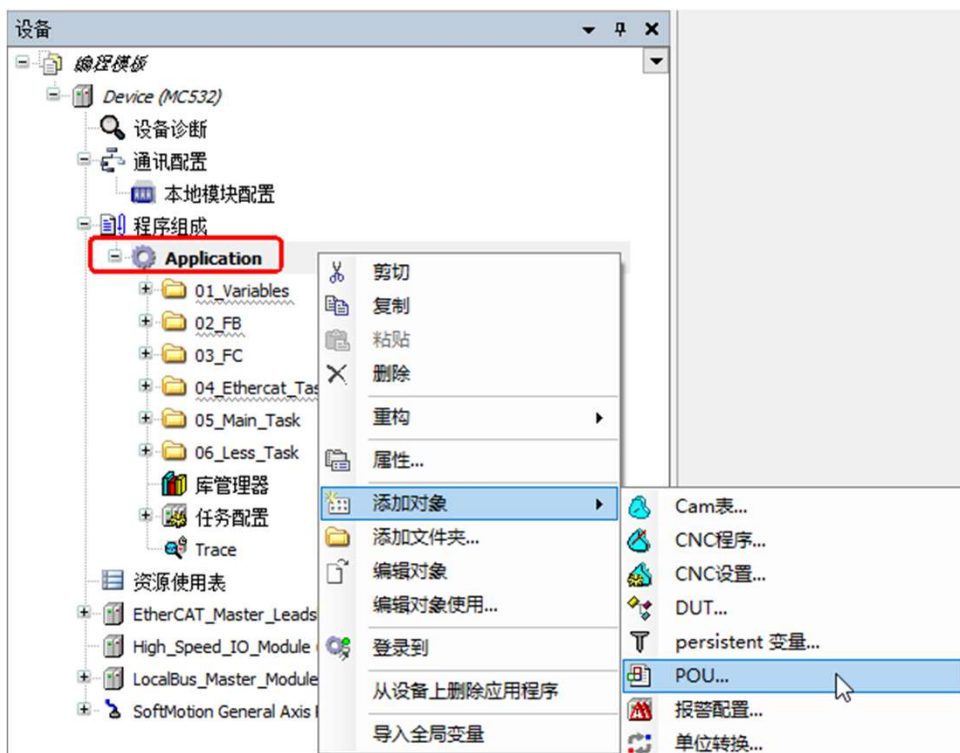
顺序功能图SFC

涉及到流程控制部分的程序，请选择**SFC**语言，编写的程序会条理清晰，逻辑关系不会混乱；

04

功能块与函数

右键Application- “添加对象” - “POU”，新建功能块FB_ALT



```
ALT x
1 {attribute 'hide_all_locals'}
2 //当ALT指令xExecute检测到上升沿时, xOut将反转其状态
3 FUNCTION_BLOCK ALT
4 VAR_INPUT
5     xExecute : BOOL := FALSE ; //上升沿触发有效
6 END_VAR
7 VAR_OUTPUT
8     xOut : BOOL := FALSE ; //输出变量
9 END_VAR
10 VAR
11     R_TRIG0: R_TRIG;
12     ProState: INT; //步数
13 END_VAR
14
15
16 R_TRIG0(CLK:=xExecute , Q=> );
17
18 IF R_TRIG0.Q THEN//上升沿触发一周期
19     ProState:=1;
20 END_IF
21
22 CASE ProState OF
23 0:
24     ;
25 1:
26     IF xOut = TRUE THEN
27         xOut := FALSE ;
28         ProState:=0;
29     ELSE
30         xOut := TRUE ;
31         ProState:=0;
32     END_IF
33 2:
34     ;
35 END_CASE
```

输入输出与局部变量定义窗口

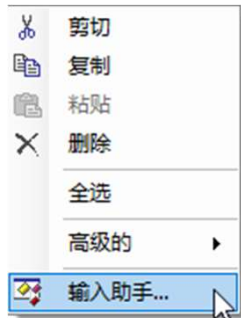
功能块编程案例——反转输出信号状态程序



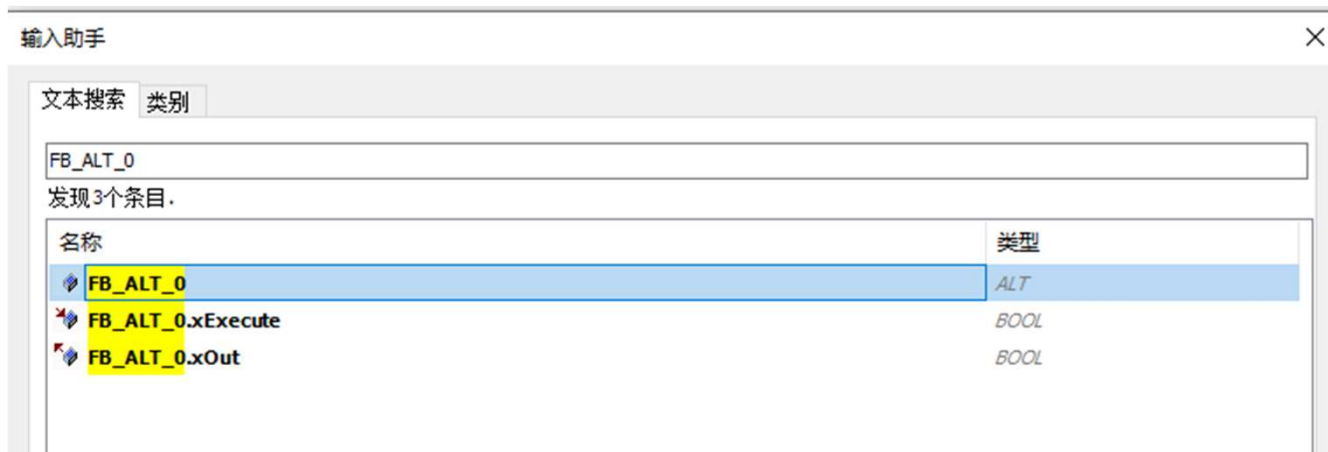
```
PLC_PRG x
2
3 PROGRAM PLC_PRG
4 VAR
5     FB_ALT_0 : ALT ;
6     xIn: BOOL;    //输入
7     xOut: BOOL;   //输出
8 END_VAR
9
10
```

在变量声明区实例化功能块

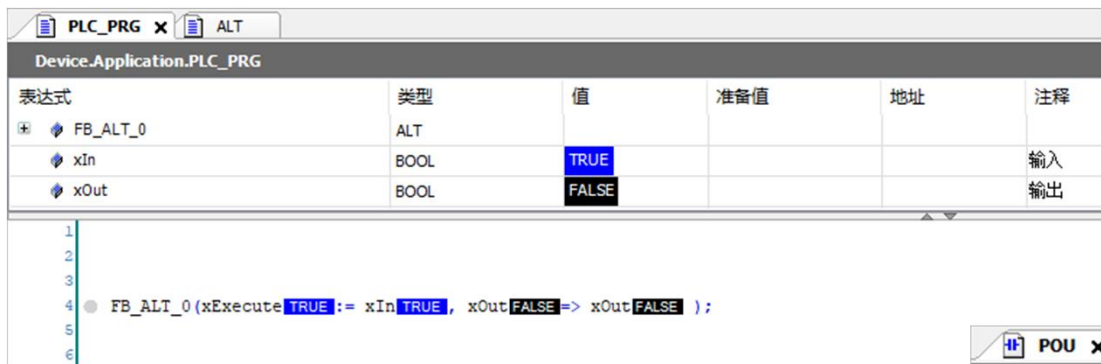
```
1
2
3 FB_ALT_0(xExecute:= xIn, xOut=> xOut );
4
5
6
7
8
```



在代码编辑区点击鼠标右键→选择输入助手→搜索并调用功能块实例，分别给FB功能块输入输出引脚xIn、xOut关联变量



ST编程环境下调用方式:



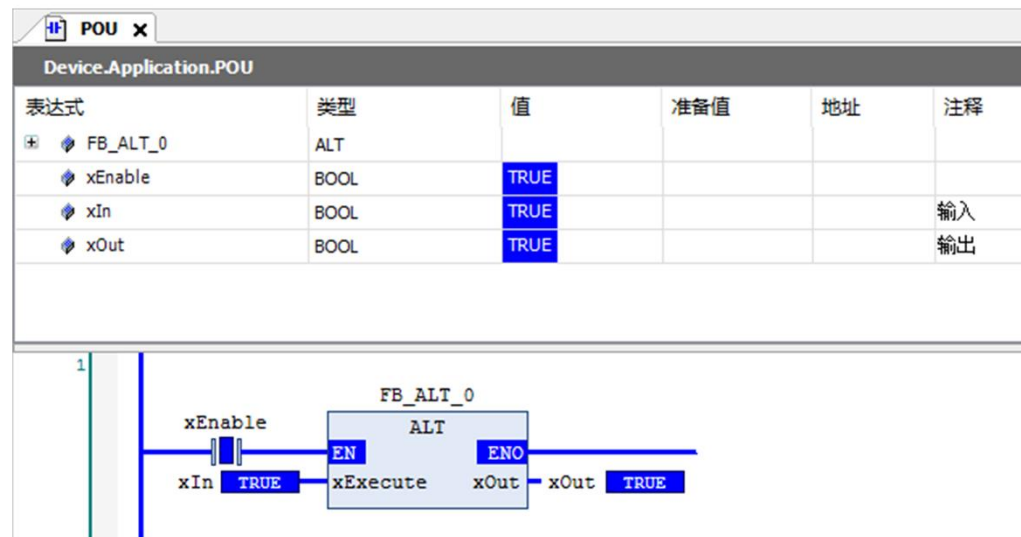
Device.Application.PLC_PRG

表达式	类型	值	准备值	地址	注释
FB_ALT_0	ALT				
xIn	BOOL	TRUE			输入
xOut	BOOL	FALSE			输出

```
1  
2  
3  
4 ● FB_ALT_0(xExecute TRUE := xIn TRUE, xOut FALSE => xOut FALSE );  
5  
6
```

LD编程环境下调用方式:

EN引脚变量为ON时功能块程序执行，
反之则功能块程序不扫描刷新

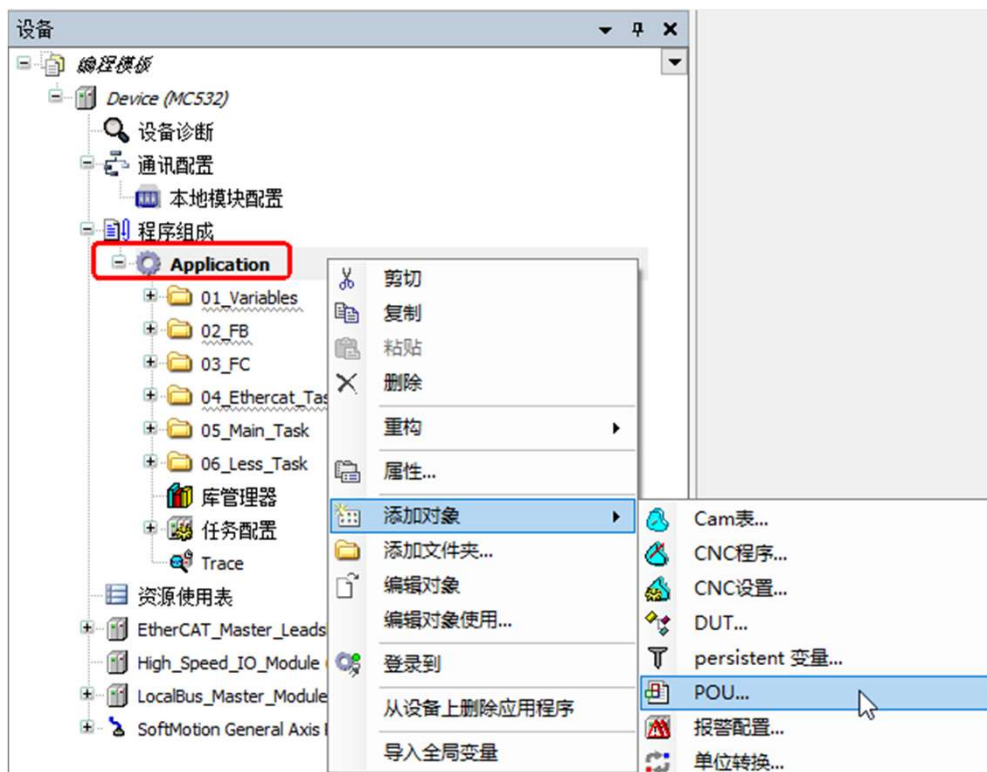


Device.Application.POU

表达式	类型	值	准备值	地址	注释
FB_ALT_0	ALT				
xEnable	BOOL	TRUE			
xIn	BOOL	TRUE			输入
xOut	BOOL	TRUE			输出

```
1  
|  
| xEnable [ ] --- EN ---> FB_ALT_0 (ALT)  
| xIn [ TRUE ] --- xExecute ---> FB_ALT_0  
| xOut [ TRUE ] --- ENO ---> FB_ALT_0  
| xOut [ TRUE ] ---> xOut
```

右键Application- “添加对象” - “POU”，新建函数FUN_XCH，给定返回类型



```
XCH x
1 {attribute 'hide_all_locals'}
2 //数据交换指令, 两个浮点数据互换
3
4 FUNCTION XCH : BOOL //返回值, TRUE: 执行成功, FALSE: 执行错误
5 VAR_IN_OUT
6     fDataSrc1 : REAL ; //交互数据
7     fDataSrc2 : REAL ; //交互数据
8 END_VAR
9
10 VAR
11     fDataSrc_relay : REAL ; //交互数据中继
12
13 END_VAR
14

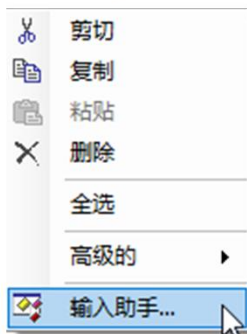
1
2
3 fDataSrc_relay := fDataSrc1 ; //数据1中转
4 fDataSrc1 := fDataSrc2 ;
5 fDataSrc2 :=fDataSrc_relay ;
6 XCH := TRUE ;
```

输入、输出和局部变量定义窗口

函数编程案例——浮点数据交换程序

FC函数—调用

```
PLC_PRG x
1
2
3 PROGRAM PLC_PRG
4 VAR
5     fDataSrc1: REAL; //浮点数据1
6     fDataSrc2: REAL; //浮点数据2
7 END_VAR
8
9
10
11
12 XCH(fDataSrc1:= fDataSrc1, fDataSrc2:= fDataSrc2);
13
14
15
```



在代码编辑区点击鼠标右键→选择输入助手→搜索并调用函数，给定输入输出变量fDataSrc1、fDataSrc2

输入助手

文本搜索 类别

XCH

发现3个条目.

名称	类型
XCH	FUNCTION
GVL_PLC.FB_FreeTcp_Server_0.v_xCheck	BOOL
GVL_PLC.FB_FreeTcp_Server_0.v_xcheck2	BOOL



ST编程环境下调用方式及执行

PLC_PRG x

Device.Application.PLC_PRG

表达式	类型	值	准备值	地址	注释
fDataSrc1	REAL	0			浮点数据1
fDataSrc2	REAL	12			浮点数据2

```
1  
2  
3 XCH(fDataSrc1:= fDataSrc1 0, fDataSrc2:= fDataSrc2 12);  
4  
5
```

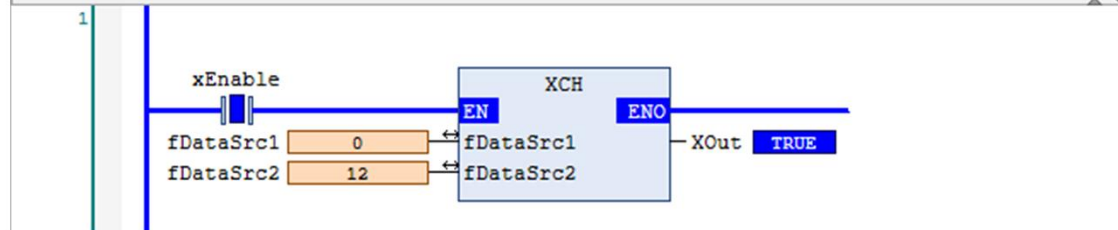
LD编程环境下调用方式及执行

EN引脚变量为ON时函数执行扫描刷新

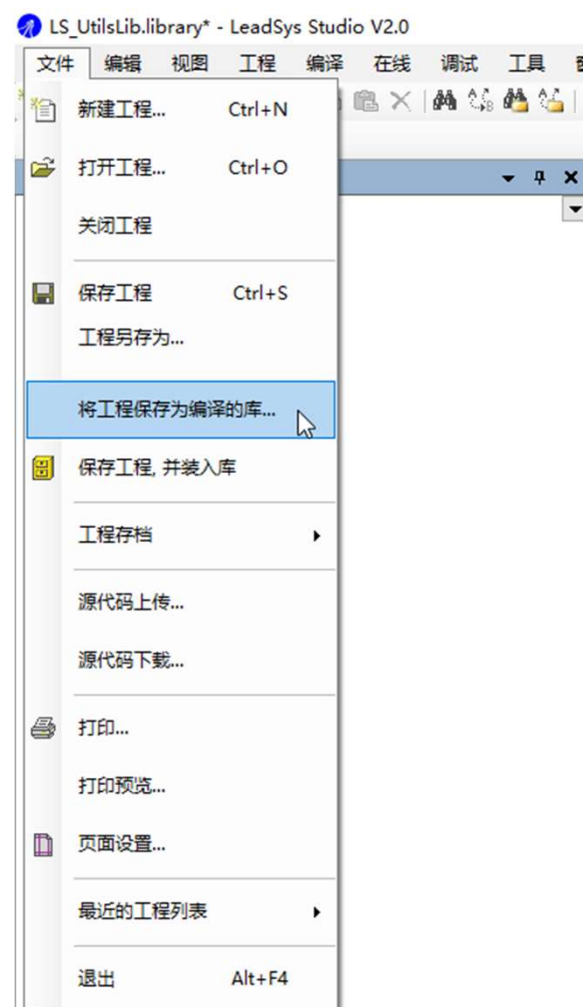
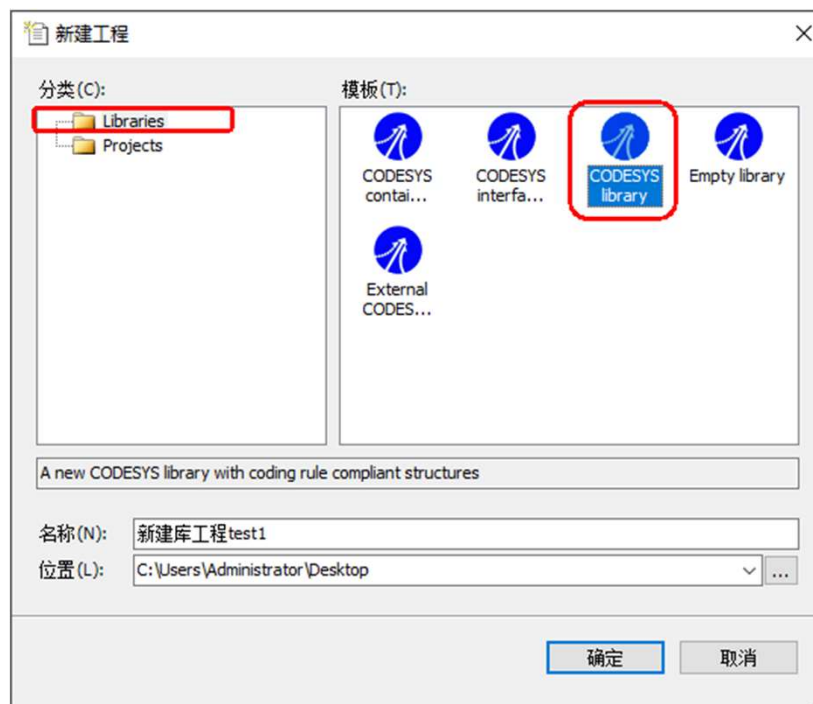
PLC_PRG POU x

Device.Application.POU

表达式	类型	值	准备值	地址	注释
xEnable	BOOL	TRUE			
fDataSrc1	REAL	0			浮点数据1
fDataSrc2	REAL	12			浮点数据2
XOut	BOOL	TRUE			



将FB功能块或FC函数封装成库



新建库工程，将功能块与函数复制到工程中，单击文件，选择“将工程保存为编译的库”；

将保存到的库安装调用到任一工程中，即可在工程中调用相关功能块与函数。

- 每个实例化后的FB功能块具有独立的内部变量，控制器执行系统给功能块内部状态变量分配内存；
- FB功能块对于相同参数的输入变量值，可能存在有不同的内部状态变量，会得到不同的计算结果；
- FC函数没有内部状态存储，相同的输入参数能得到相同的输出结果；
- FC函数是没有输出函数的，因为它的名字就是输出。

FB块优点：

1. 易于移植性，对于相同控制逻辑不同参数的被控对象，只要实例化多次同一个FB块就可以解决；
2. 多重调用，减少重复工作，提高效率；
3. 多次调用时，参数修改方便；
4. 有独立的存储区。

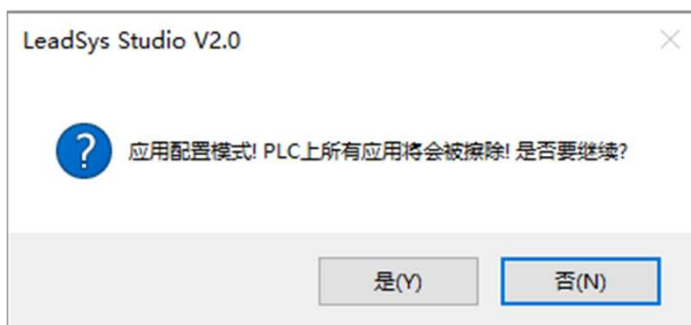
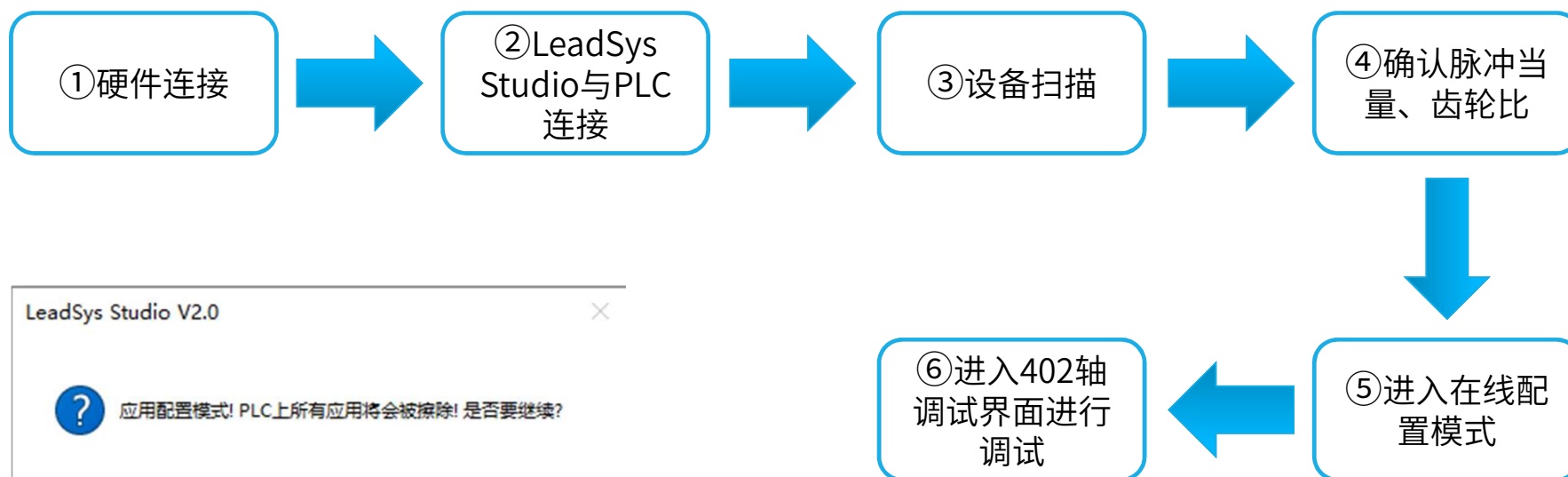
FC块优点：

1. 小巧灵活，对于非多次调用的程序更易理解；
2. 不占用额外的存储资源。

05

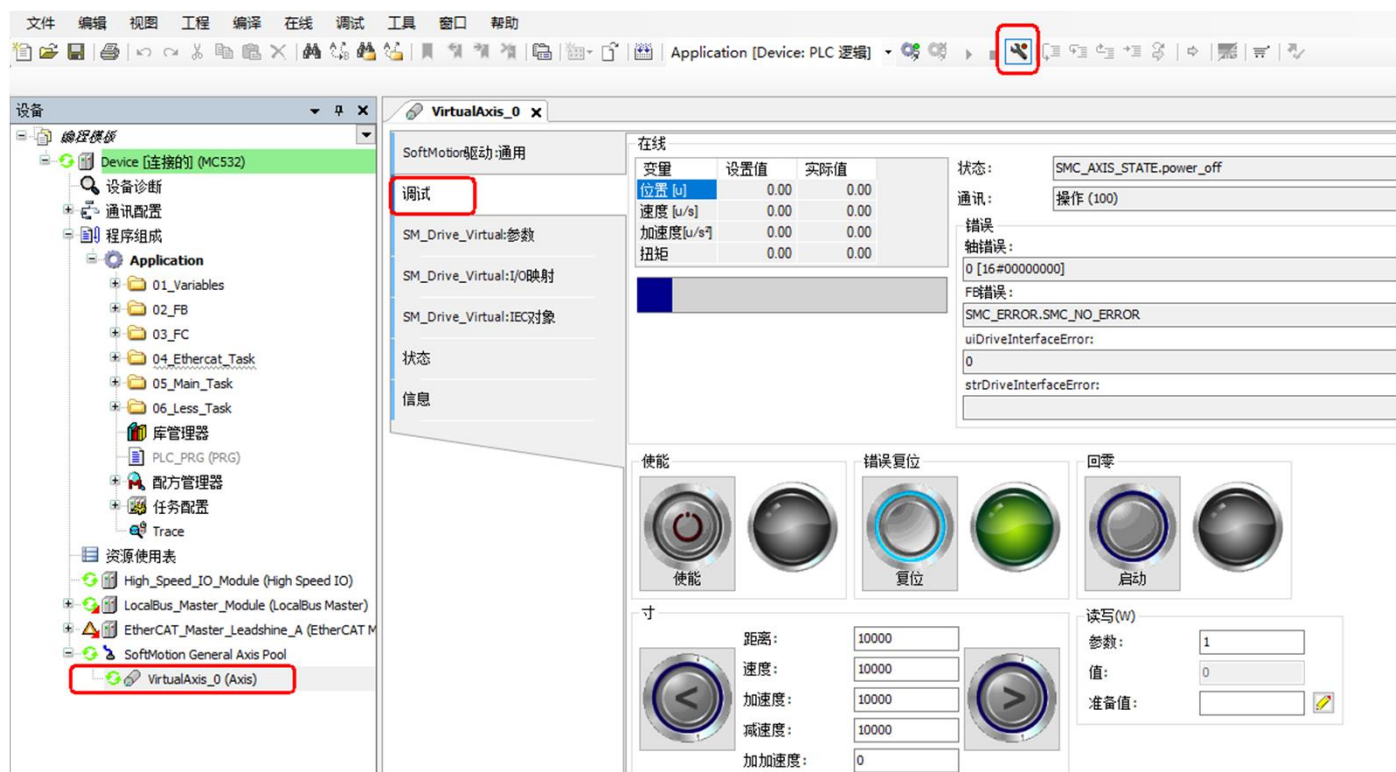
运动控制轴

调试流程



注意：进入在线配置模式
会擦除PLC中的应用程序

用户可以通过轴的在线调试功能在无需编写用户程序的情况下，直接驱动轴动作，减少用户调试所需要花费时间。（目前在线配置功能只适用于总线轴）



①监控区：分为两个部分

- (1) 轴的位置、速度、加速度、扭矩
- (2) 轴的状态机、通讯状态、及错误监控

控

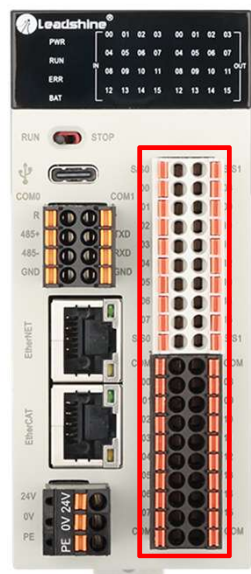
②调试区：分为三个部分

- (1) 设置距离、速度、加速度、减速度等
- (2) 使能、复位及回零功能
- (3) 寸动功能

本地轴及总线轴：

SC系列的轴根据控制方式分为本地轴及总线轴两种类型；

本地轴跟总线轴在使用运动控制指令时是一样的，但组态方式略有不同；



本地脉冲轴（SC2系列）：
直接通过PLC本体的DO输出脉冲给到步进或者伺服带动电机进行运动



总线轴（SC5系列）：
通过Ethercat总线控制步进或者伺服带动电机进行运动

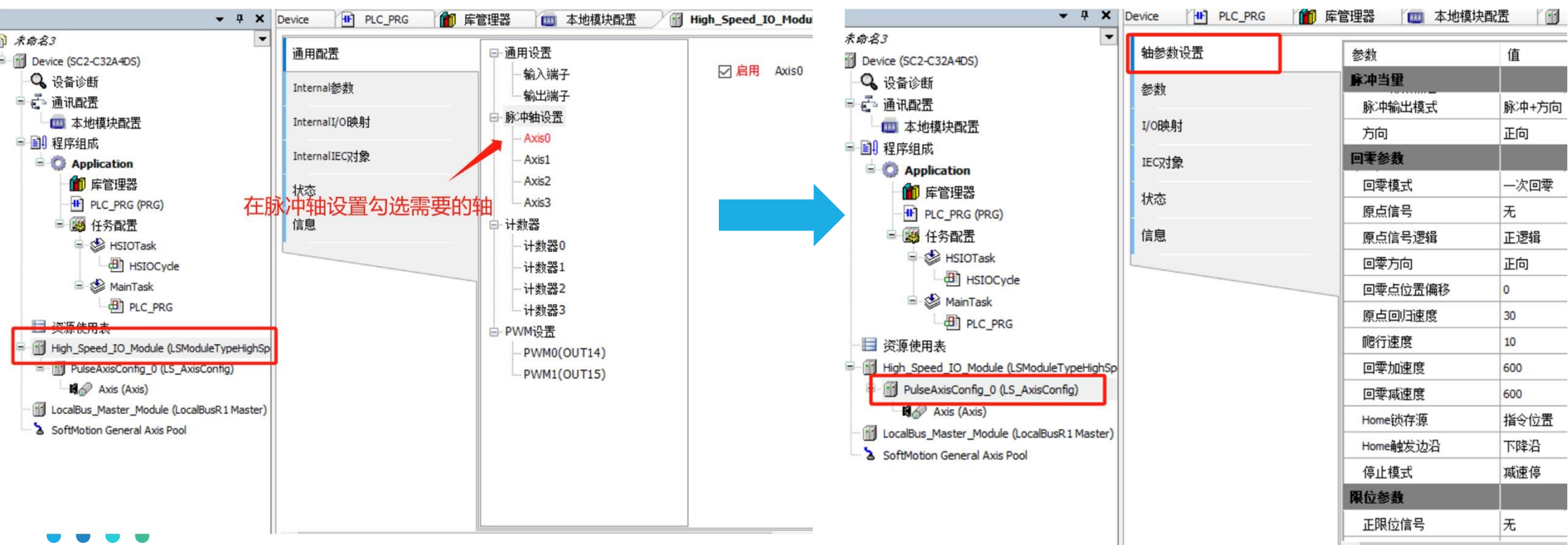
本地脉冲轴

本地轴的组态需要打开High_Speed_IO_Module，在“通用配置”中，勾选需要组态的轴，在新建轴参数设置轴的“脉冲当量”及“回零参数”。

轴0输出端口：默认Y0脉冲/Y1方向，轴1输出端口：默认Y2脉冲/Y3方向...

SC系列本地轴组态完成后，直接程序里调用轴名即可，**无需重复调用功能块绑定本地轴**

轴控指令与高速IO功能POU应放HSIOTask任务下，读轴状态和逻辑控制放在其他优先级任务



在脉冲轴设置勾选需要的轴

参数	值
脉冲当量	
脉冲输出模式	脉冲+方向
方向	正向
回零参数	
回零模式	一次回零
原点信号	无
原点信号逻辑	正逻辑
回零方向	正向
回零点位置偏移	0
原点回归速度	30
爬行速度	10
回零加速度	600
回零减速度	600
Home锁存源	指令位置
Home触发边沿	下降沿
停止模式	减速停
限位参数	
正限位信号	无

通过LS_MTHSIO本地高速IO库，可以查阅轴高速计数指令、中断使能、设置计数器预置值指令、PWM输出、脉冲轴回零指令等。

名称	命名空间	有效的版本
IoStandard = IoStandard, 3.5.15.0 (System)	IoStandard	3.5.15.0
LC_FileManage = LC_FileManage, 3.5.15.40 (Leadshine Technology Co.Ltd)	LC_FileManage	3.5.15.40
LS_MTHSIO = LS_MTHSIO, 1.0.0.17 (Leadshine Technology Co.Ltd)	LS_MTHSIO	1.0.0.17
LS_SysLib = LS_SysLib, 1.0.0.6 (Leadshine Technology Co.Ltd)	LS_SysLib	1.0.0.6
SM3_Basic = SM3_Basic, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Basic	4.10.0.0
SM3_CNC = SM3_CNC, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_CNC	4.10.0.0



The screenshot shows the documentation for the LS_MTHSIO library. On the left, a tree view displays the library structure: DataTypes, DriveInterface, Globals, POU's, Counter (containing LS_CompareFIFO, LS_Counter, LS_EnableInterrupt, LS_PresetValue, LS_PWM, LS_TouchProbe), and MC_HOME_P. The main area shows the library metadata:

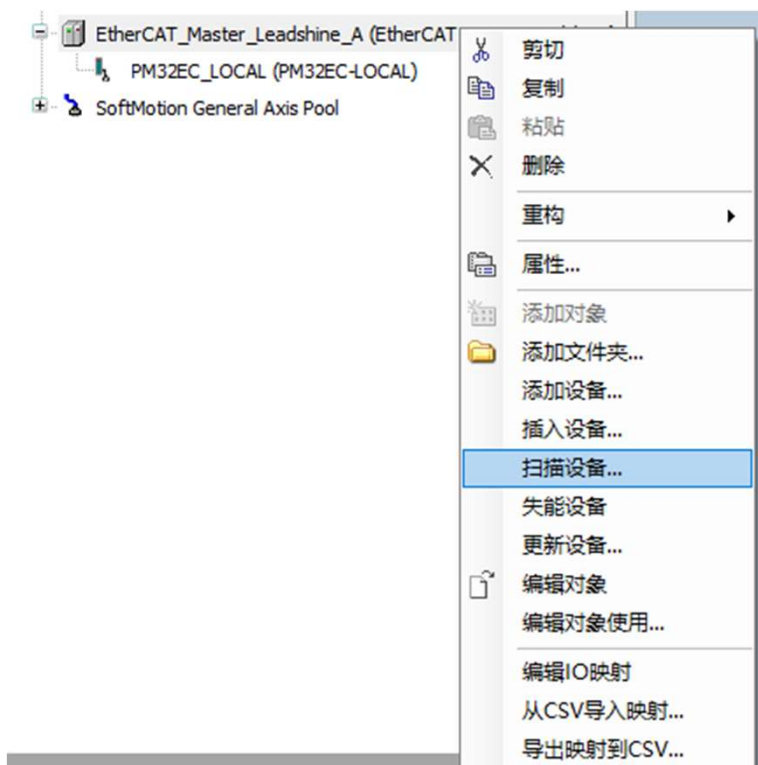
LS_MTHSIO Library Documentation

Company: Leadshine Technology Co.Ltd
Title: LS_MTHSIO
Version: 1.0.0.17
Categories: LeadSys|Public
Namespace: LS_MTHSIO
Author: Leadshine
Placeholder: LS_HSIO

Description [1]
高速IO通用库 (目前支持SC系列控制器)

Contents:

总线轴组态可以通过手动添加 / 扫描设备的方式组态。



需要该功能需要先连接到PLC，保证PLC与从站设备的硬件接口连接正常，即可进行扫描。

轴的单位换算：

由于在程序中应用时，使用的都是用户单位，故需要对单位换算进行设置。本地轴在“轴参数设置”里设置，总线轴在402轴“脉冲当量换算：传动参数”里设置。

此处以总线轴为例，假设电机每转脉冲数为10000，电机与工作台的齿轮比为20:1，工作台转一圈，工件移动17mm，设置如右图所示：

1. 每转脉冲数，与伺服或步进的分辨率需对应

2. 是否使用变速装置，及相关参数设置，需与实际机械机构对应



脉冲当量换算

原点参数设置

SoftMotion驱动:通用

SoftMotion:自动映射

调试

SM_Drive_ETC_GenericDSP402:参数

SM_Drive_ETC_GenericDSP402:I/O映射

SM_Drive_ETC_GenericDSP402:IEC对象

状态

信息

传动参数

反向

电机旋转一圈的指令脉冲数 10000 指令脉冲/转

不使用变速装置

工作台旋转一圈的工作行程 17 单位/圈

参考：单位换算公式

$$\text{脉冲数 (pulse)} = \frac{\text{电机旋转一圈的指令脉冲数 [DINT]}}{\text{工作台旋转一圈的工作行程 [LREAL]}} \times \text{移动距离 [用户单位]}$$

使用变速装置

工作台旋转一圈的工作行程 17 单位/圈

(如果轴类型是旋转模式，请参考轴基本设置界面的旋转周期值)

齿轮比分子 (下图中 (5) 的齿数) 1

齿轮比分母 (下图中 (4) 的齿数) 20

轴类型为线性模式

参考：单位换算公式

$$\text{脉冲数 (pulse)} = \frac{\text{电机旋转一圈的指令脉冲数 [DINT]} \times \text{齿轮比分子 [DINT]}}{\text{工作台旋转一圈的工作行程 [LREAL]} \times \text{齿轮比分母 [DINT]}} \times \text{移动距离 [用户单位]}$$

M:电机, W:工作台



轴类型为旋转模式

参考：单位换算公式

$$\text{脉冲数 (pulse)} = \frac{\text{电机旋转一圈的指令脉冲数 [DINT]} \times \text{齿轮比分子 [DINT]}}{\text{工作台旋转一圈的工作行程 [LREAL]} \times \text{齿轮比分母 [DINT]}} \times \text{移动距离 [用户单位]}$$

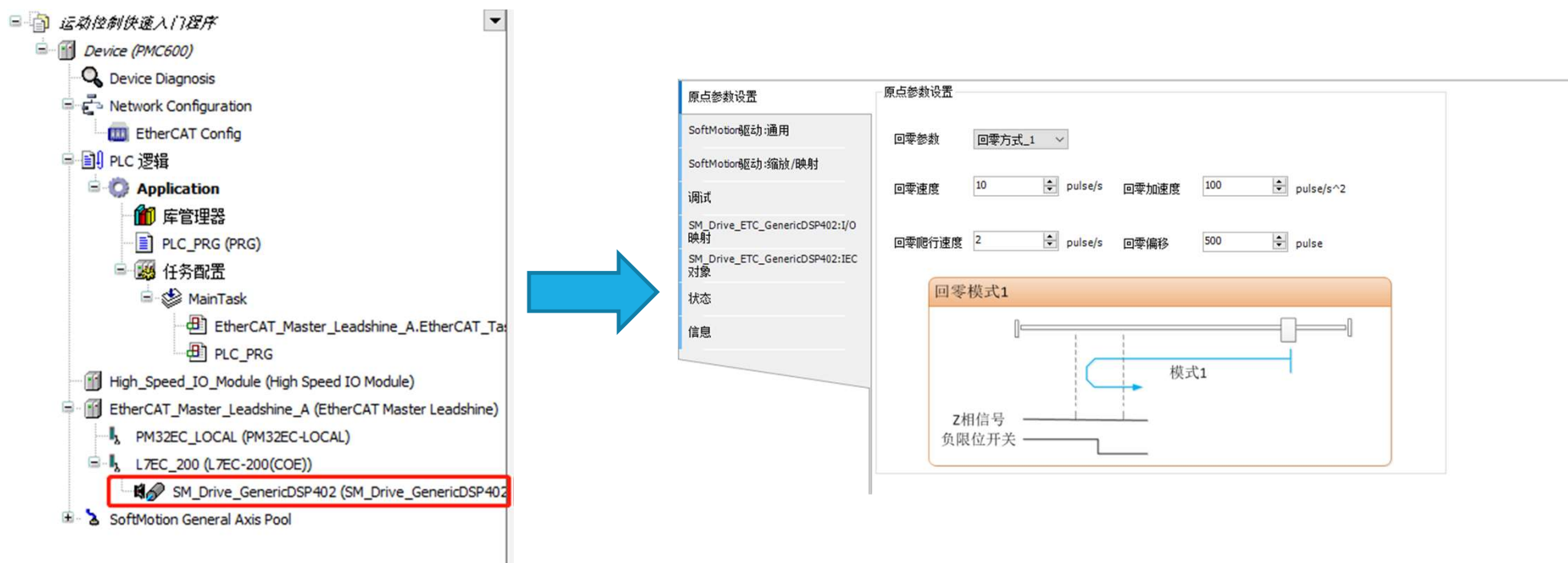
设置每转脉冲数 10000

设置使用变速装置 工作台每转移动量 17Unit 齿轮比为20:1

回原参数设置：

本地轴与总线轴关于回原方式设置有所不同；

总线轴设置：在对应CIA402轴的“原点参数设置”里设置回原方式及回原参数。

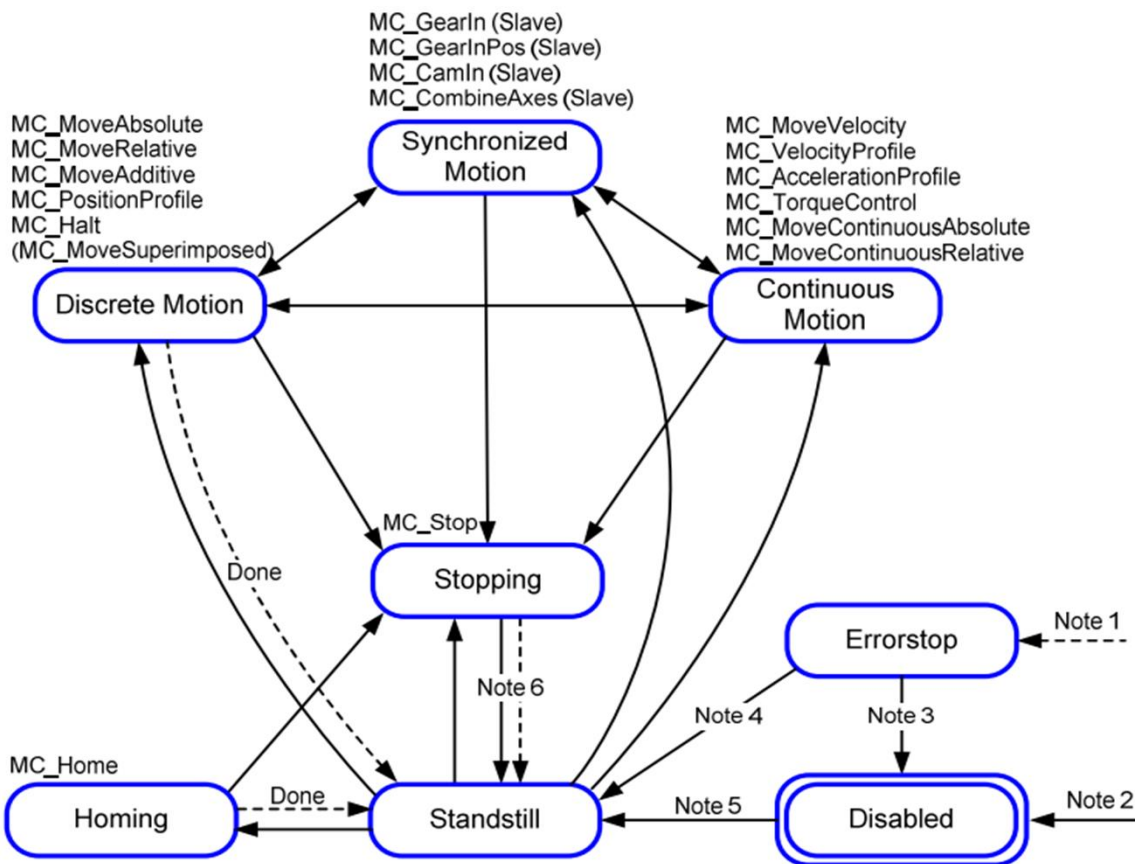


The screenshot displays the software interface for configuring a bus axis. On the left, a tree view shows the project structure, with the axis **SM_Drive_GenericDSP402 (SM_Drive_GenericDSP402)** selected and highlighted in red. A blue arrow points from this selection to the right-hand configuration window.

The configuration window is titled "原点参数设置" (Origin Parameter Setting) and contains the following settings:

- 回零参数: 回零方式_1
- 回零速度: 10 pulse/s
- 回零加速度: 100 pulse/s²
- 回零爬行速度: 2 pulse/s
- 回零偏移: 500 pulse

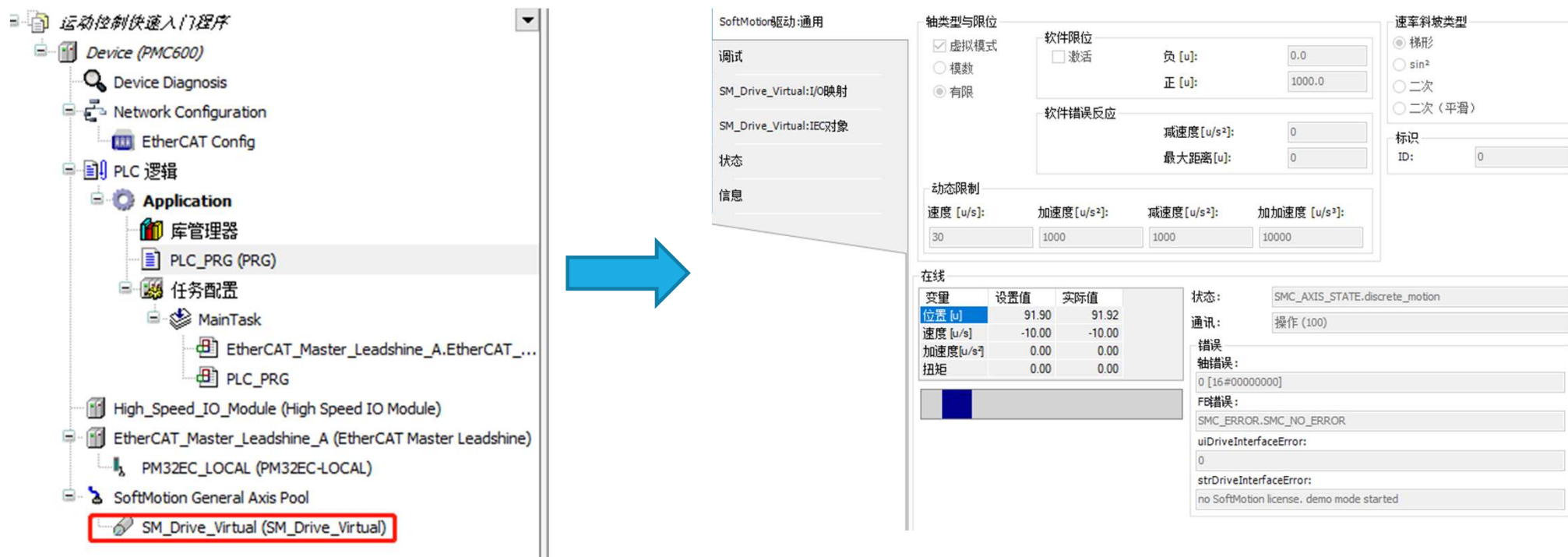
Below the settings is a diagram titled "回零模式1" (Homing Mode 1). The diagram shows a horizontal axis with a slider. A blue arrow labeled "模式1" indicates the homing direction. Below the axis, two signals are shown: "Z相信号" (Z-phase signal) and "负限位开关" (Negative limit switch). The Z-phase signal is a square wave, and the limit switch signal is a step function that transitions from high to low.



排查问题有效工具

名称	值	注释
Power_off (Disabled)	0	轴未使能, 需执行 MC_Power 指令
Errorstop	1	故障停止状态, 需先执行 MC_Reset/MC_Power 指令
Stopping	2	停止中, 等待停机操作完成
Standstill	3	使能状态, 轴已停止运行
Discrete_Motion	4	轴处于离散运行状态
Continuous_Motion	5	轴处于连续运行中
Synchronized_Motion	6	轴处于同步运行中
Homing	7	轴处于回零运行中, 等待归零操作执行完成

调试时可以在轴信息界面监控当前运行状态。



运动控制快速入门程序

- Device (PMC600)
 - Device Diagnosis
 - Network Configuration
 - EtherCAT Config
 - PLC 逻辑
 - Application
 - 库管理器
 - PLC_PRG (PRG)
 - 任务配置
 - MainTask
 - EtherCAT_Master_Leadshine_A.EtherCAT_...
 - PLC_PRG
 - High_Speed_IO_Module (High Speed IO Module)
 - EtherCAT_Master_Leadshine_A (EtherCAT Master Leadshine)
 - PM32EC_LOCAL (PM32EC-LOCAL)
 - SoftMotion General Axis Pool
 - SM_Drive_Virtual (SM_Drive_Virtual)**

SoftMotion驱动:通用

调试

SM_Drive_Virtual:I/O映射

SM_Drive_Virtual:IEC对象

状态

信息

轴类型与限位

虚拟模式
 模数
 有限

软件限位

激活

负 [u]: 0.0

正 [u]: 1000.0

软件错误反应

减速度 [u/s²]: 0

最大距离 [u]: 0

速率斜坡类型

梯形
 sin²
 二次
 二次 (平滑)

标识

ID: 0

动态限制

速度 [u/s]: 30 加速度 [u/s²]: 1000 减速度 [u/s²]: 1000 加加速度 [u/s³]: 10000

在线

变量	设置值	实际值
位置 [u]	91.90	91.92
速度 [u/s]	-10.00	-10.00
加速度 [u/s²]	0.00	0.00
扭矩	0.00	0.00

状态: SMC_AXIS_STATE.discrete_motion

通讯: 操作 (100)

错误

轴错误: 0 [16#00000000]

F球错误: SMC_ERROR.SMC_NO_ERROR

uiDriveInterfaceError: 0

strDriveInterfaceError: no SoftMotion license. demo mode started

轴结构体：

在LeadSys Studio平台中，每个轴创建后，其对应的结构体也会被创建，用户可以在编程中直接调用。常用的结构体成员如下：

结构体成员	注释
nAxisState	对应PLCopen状态机 0: power_off 1: errorstop 2: stopping 3: standstill 4: discrete_motion 5: continuous_motion 6:synchronized_motion 7: homing
wCommunicationState	通讯状态 等于100时通讯正常
fSetPosition	命令位置 (unit)
fSetVelocity	命令速度 (unit)
fSetTorque	命令转矩 (unit)
fActPosition	反馈位置 (unit)
fActVelocity	反馈速度 (unit)
fActTorque	反馈转矩 (unit)

用法：

轴名.结构体成员

例：

Axis_0.fActposition

单轴控制是指通过位置控制、速度控制、转矩控制这3种模式控制轴运动。

在单轴控制中，脉冲型、EtherCAT总线型电机的常用单轴控制指令如表所示，指令的详细说明请参考以下手册



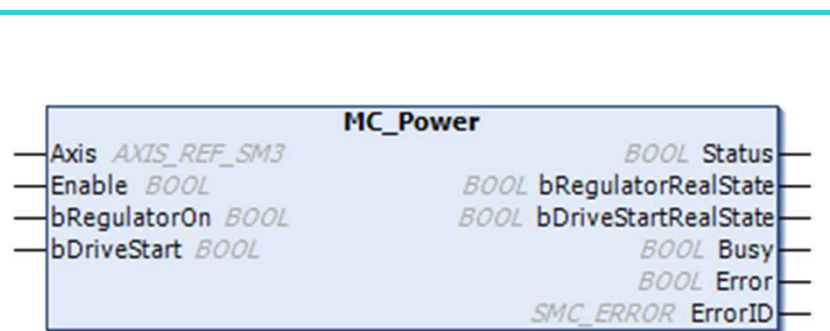
SC系列运动
控制指令-20:



雷赛大中型P
LC指令手册.r

单轴控制指令	指令名称	说明
MC_Power	轴使能	将驱动器切换为可运行状态
MC_SetPosition	设置轴位置	修改当前轴位置寄存器的位置。该操作电机不会移动，仅产生坐标偏移
MC_ReadStatus	读取轴的状态	读取轴的状态数据
MC_Jog	点动	以指定速度运动，常用于点动控制
SMC_Inch	寸动	手动控制轴一段一段的朝指定方向运动
MC_MoveRelative	相对点位运动	以当前位置为起点，移动指定距离
MC_MoveAbsolute	绝对点位运动	以绝对坐标控制目标位置
MC_MoveAdditive	叠加相对点位运动	在当前运行指令的基础上，再做相对点位运动
MC_MoveVelocity	速度控制	使用伺服驱动器的位置模式，控制伺服电机以指定速度运行
LS_Home_P	脉冲型电机回零	控制脉冲型电机回原点
MC_Home	总线型电机回零	控制总线型电机回原点
MC_Halt	暂停	命令电机暂停运行，可再次调用MC_Movexxx指令触发运行
MC_Stop	停止	可命令电机以指定减速度减速后停止
MC_Reset	轴报警复位	当伺服电机出现报警停止后，可运行该指令复位。如轴状态 Axis.nAxisState 为 errorstop 需要切换到 StandStill 要用该指令复位

使能、复位指令



轴使能指令

使用MC_Power指令，可以将轴从Disable状态切换到StandStill状态轴。运动之前，必须先使用MC_Power指令，使能轴。



轴复位指令

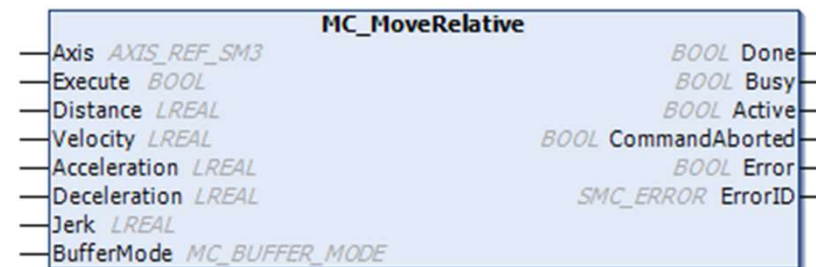
使用MC_Reset指令，可以复位轴控指令运行异常导致的故障，即从ErrorStop状态恢复到Disable或者Standstill状态。复位后轴才可以正常执行动作指令。

绝对、相对定位指令



绝对定位指令

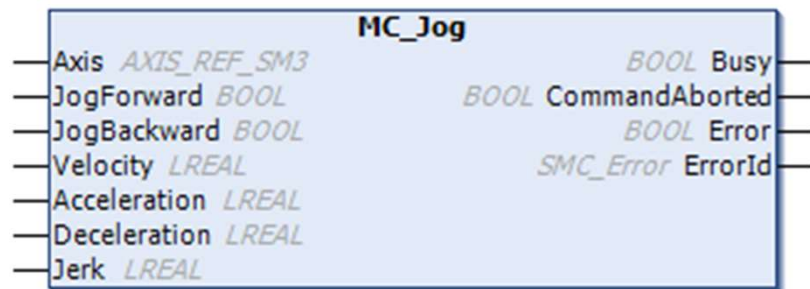
使用MC_MoveAbsolute指令，可以让运动控制轴实现绝对定位。当轴的ActPosition与指令的Position参数一致时，Done信号输出，定位完成。



相对定位指令

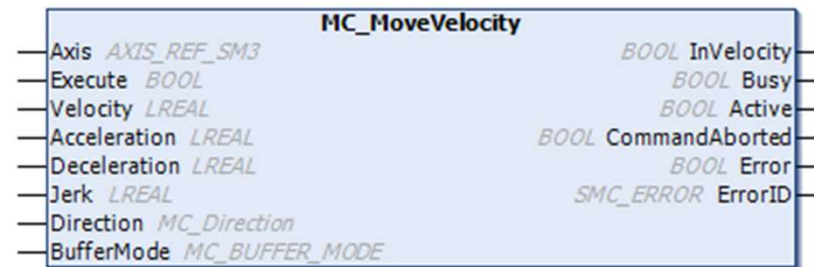
使用MC_MoveRelative指令，可以让运动控制轴实现相对定位。定位完成之后，轴的ActPosition等于当前位置+指令中Distance设定的值。

点动、速度指令



点动指令

使用MC_Jog指令，可以让运动控制轴实现点动。导通Enable之后，触发JogForward及JogBackward可进行正向点动、反向点动。



速度指令

使用MC_MoveVelocity指令，可以让运动控制轴以设定速度运动。触发Execute之后，需要用MC_Stop指令才能停止轴的运行。

停止、暂停指令



停止指令

触发MC_Stop之后，锁存左侧输入参数，按照输入参数停止轴的运行。此时，轴的状态变成Stopping状态。当Execute有效时，无法使轴运动。



暂停指令

触发MC_Halt指令，锁存左侧输入参数，按照输入参数停止轴的运行。与MC_Stop不同的是，此时轴状态变为Discrete motion状态。Execute有效时，使用其他指令可以打断该指令，使轴运动。

回原、设定位置指令

MC_Home

Axis	AXIS_REF_SM3	BOOL Done
Execute	BOOL	BOOL Busy
Position	LREAL	BOOL CommandAborted
		BOOL Error
		SMC_ERROR ErrorID

回原指令

MC_Home 指令用于启动**总线运动控制轴**的回原功能，触发指令之后，轴根据**CIA402轴→原点参数设置**里的模式进行原点回归，回原完成之后Done信号输出。

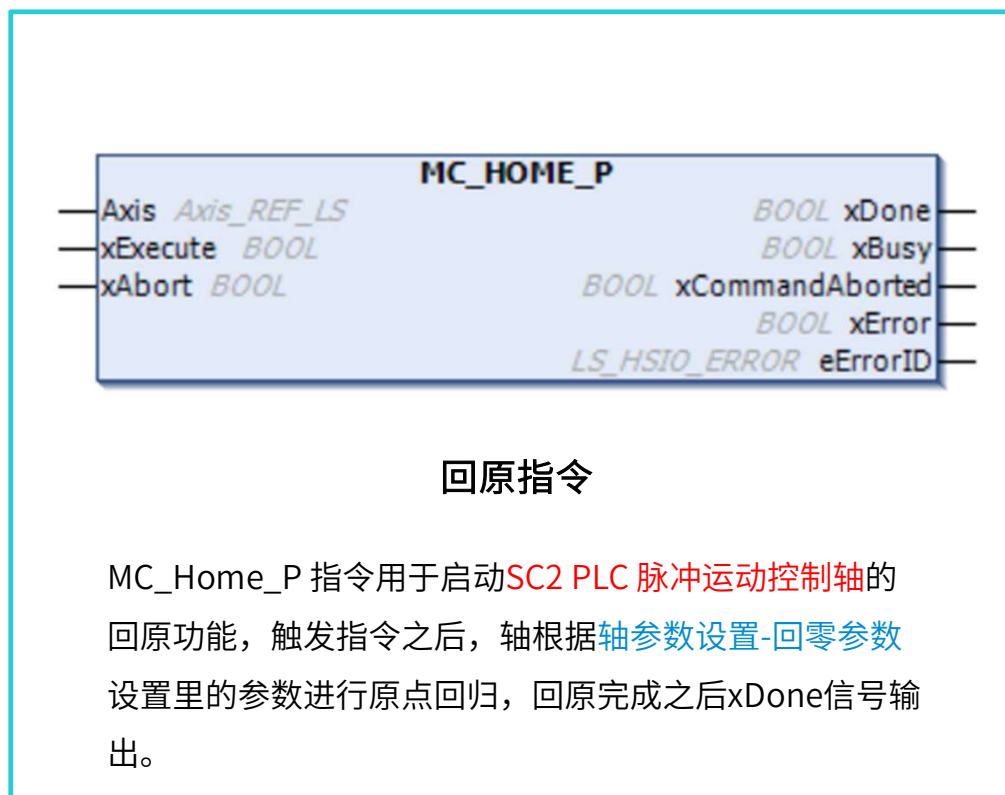
MC_SetPosition

Axis	AXIS_REF_SM3	BOOL Done
Execute	BOOL	BOOL Busy
Position	LREAL	BOOL Error
Mode	BOOL	SMC_ERROR ErrorID

设定位置指令

MC_SetPosition指令用于设定轴的当前位置。该指令有两种模式，当输入Mode=0时，执行指令后fActposition=指令设置位置，当输入Mode=1时，执行指令后fActposition=当前位置+指令设置位置。

脉冲轴回原指令



主站控制：位置、速度曲线规划由PLC完成，规划周期是分布式时钟的同步周期。

周期同步位置模式 (csp)

周期同步位置模式下，上位控制器完成位置指令规划，然后将规划好的目标位置 607Ah 上以周期性同步的方式发送给伺服驱动器，位置、速度、转矩控制由伺服驱动器内部完成。

周期同步速度模式 (csv)

周期同步速度模式下，上位控制器将计算好的目标速度 60FF 周期性同步的发送给伺服驱动器，速度、转矩调节由伺服内部执行。

周期同步转矩模式 (cst)

此模式下，上位控制器将计算好的目标转矩 6071h 周期性同步的发送给伺服驱动器，转矩调节由伺服内部执行。当速度达到限幅值后将进入调速阶段

原点回归模式 (hm)

原点回零模式用于寻找机械原点，并定位机械原点与机械零点的位置关系。

从站自主控制：主站通过对象字典给定目标值，由驱动器内部自动完成曲线规划。

轮廓位置模式 (pp)

此模式主要用于点对点定位应用。此模式下，上位机给目标位置 (绝对或者相对)、位置曲线的速度、加减速及减速度，伺服内部的轨迹发生器将根据设置生成目标位置曲线指令，驱动器内部完成位置控制、速度控制。

轮廓速度模式 (pv)

此模式下，上位控制器将目标速度、加速度、减速度发送给伺服驱动器，速度、转矩调节由伺服内部执行。

轮廓转矩模式 (pt)

此模式下，上位控制器将目标转矩 6071h、转矩斜坡常数 6087h 发送给伺服驱动器，转矩调节由伺服内部执行。当速度达到限幅值将进入调速阶段。

程序中切换控制模式：

程序中切换控制模式可以使用指令SMC_SetControllerMode：



nControllerMode支持模式：

- SMC_nocontrol : 控制器停止周期发送轴数据；
- SMC_torque : 力矩控制，CST；
- SMC_velocity : 速度控制，CSV；
- SMC_position : 位置控制，CSP；
- SMC_current : 保持当前的控制模式。

注：CST模式需要添加对应的PDO映射，详情参考例程

06

高速计数器

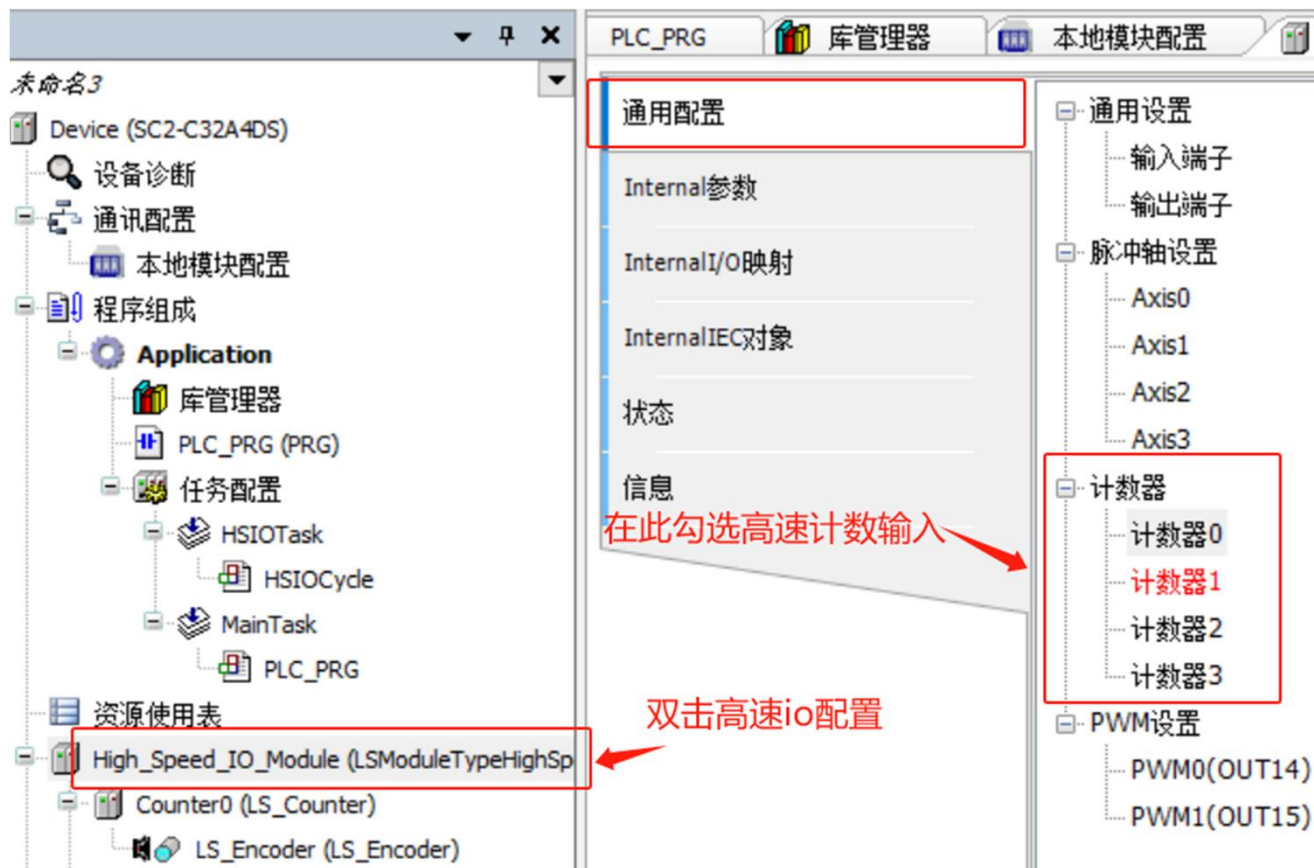
高速计数器—设置

SC2-C系列支持4路高速计数器，支持探针、比较输出、计数中断功能

SC2-C输入口 IN0~IN7 为高速输入口，可复用定义为高速计数器，其输入信号有 4 种类型

配置高速计数器：

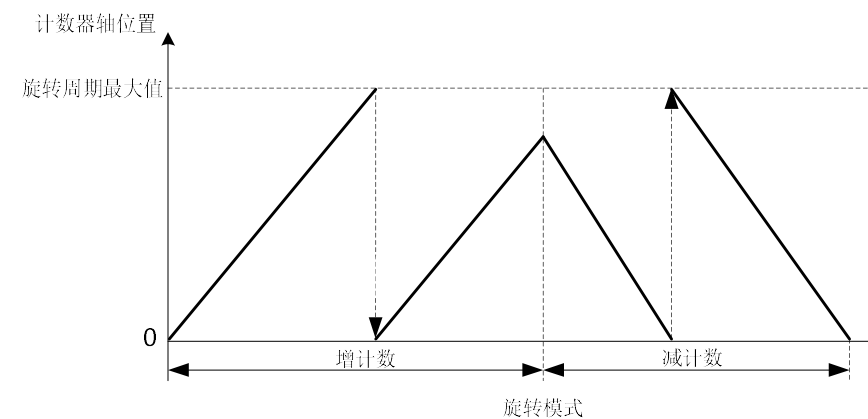
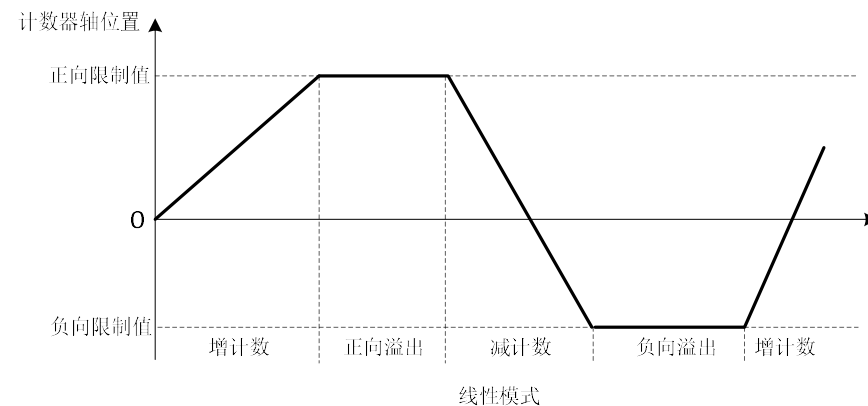
打开High_Speed_IO_Module，在“通用配置”中，勾选需要计数器。新建高速计数器后，下方会新建一个计数器轴，可在参数配置高速计数器的模式与量纲转换



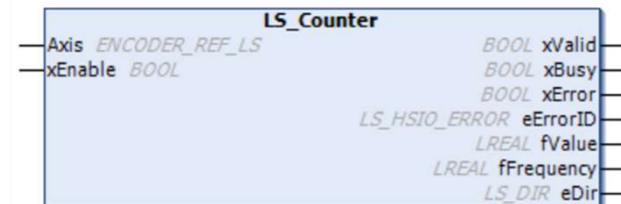
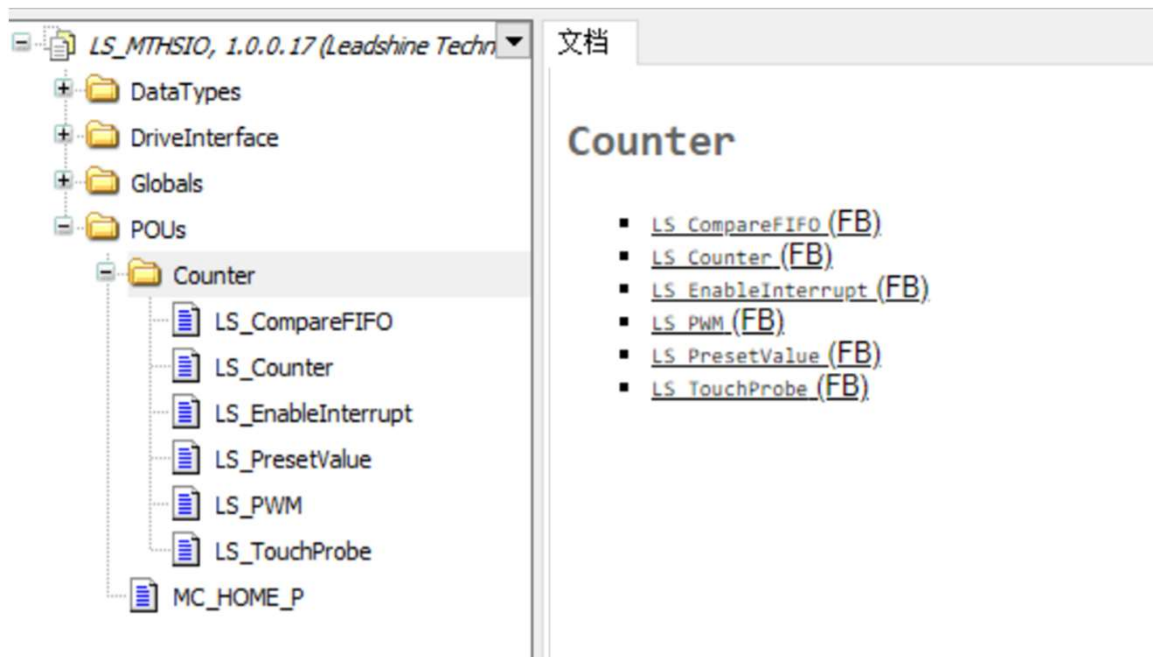
高速计数器—模式选择

参数配置		计数器配置	
参数	参数	值	备注
I/O映射	工作模式	A/B相4倍频	计数器工作模式设置
IEC对象	计数方向	正向	计数方向设置,只对A/B相模式及脉冲
状态	计数模式	线性	计数模式设置
信息	上限值	2147483647	上限值设置,范围-2147483648~21
	下限值	-2147483648	下限值设置,范围-2147483648~21
	预置输入口	无	预置输入端子设置
	探针0	无	探针0输入口设置
	比较输出信号	无	比较输出信号设置
	信号源	IN0-A相,IN1-B相	端口设置

- 计数器工作模式：支持AB相4倍频、CW/CCW、脉冲方向和单相计数
- 计数模式支持线性模式和环形模式



高速计数器支持的指令集合在LS_MTHSIO库中。



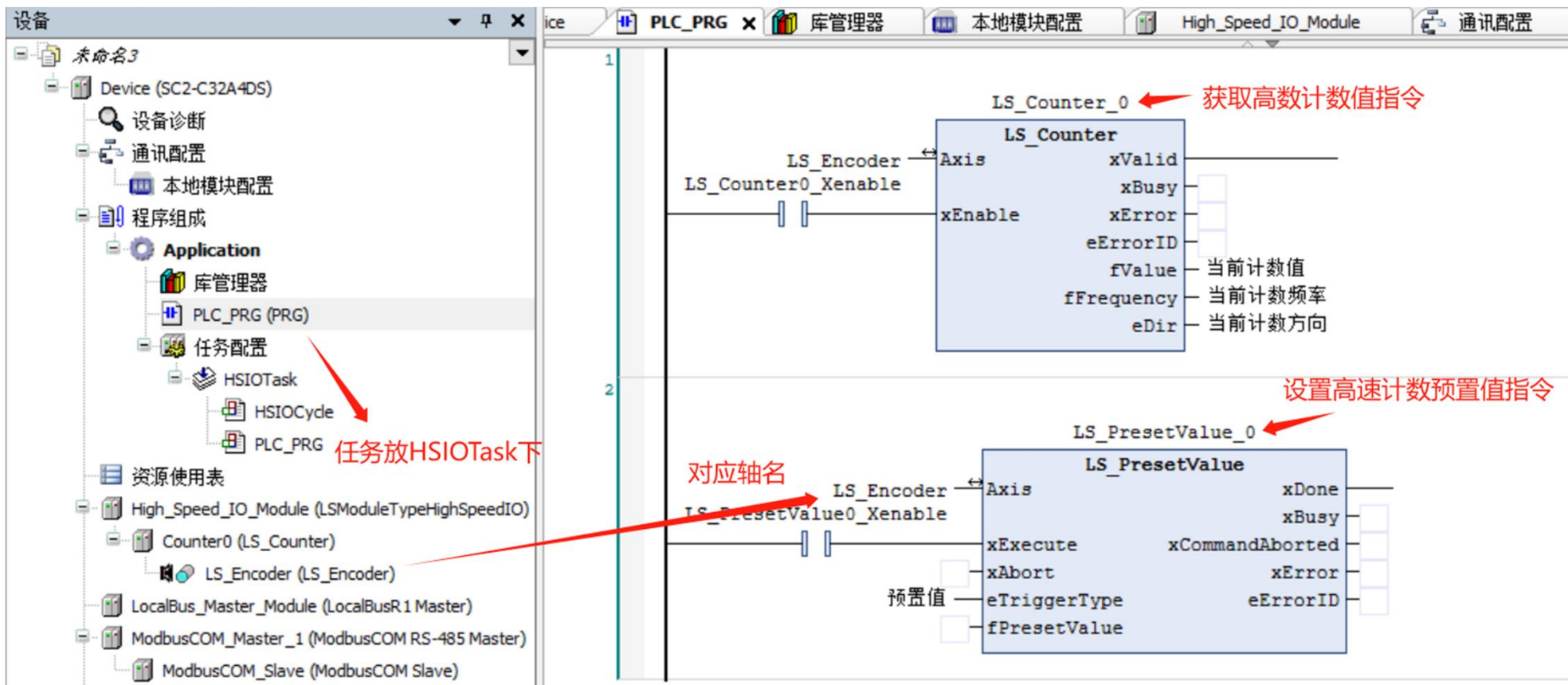
获取高速计数模块的设置参数以及计数值，需要先在Device下High_Speed_IO_Module>Counter中配置对应的硬件口功能及参数，否则该功能无效；使能为TRUE时计数器才有效



设置计数器的预置值，需要先在Device下的High_Speed_IO_Module->Counter中配置对应的预置输入口参数，否则该功能无效

功能块	名称	说明
LS_Counter	计数器使能	获取计数器的计数值、频率、方向
LS_PresetValue	计数器预置	支持内部触发、IN0-IN7外部输入信号
LS_TouchProbe	高速锁存（探针）	支持单次和连续锁存
LS_CompareStep	步长比较输出	线性模式，支持设定比较步长，输出控制支持时间方式和脉冲方式
LS_CompareFIFO	一维比较输出	FIFO队列比较，单次触发，输出控制支持时间方式和脉冲方式，最大FIFO数为1000个，可动态压入比较点
LS_Sample	计数器采样	统计指定时间内的脉冲个数
LS_PWM	PWM功能	动态脉冲宽度调试输出，与高速脉冲输出功能互斥
LS_EnableInterrupt	打开外部中断	用于接收外部输入信号触发中断任务

高速计数器指令编程实例



设备

- 未命名3
 - Device (SC2-C32A4DS)
 - 设备诊断
 - 通讯配置
 - 本地模块配置
 - 程序组成
 - Application
 - 库管理器
 - PLC_PRG (PRG)
 - 任务配置
 - HSIOTask
 - HSIOCycle
 - PLC_PRG
 - 资源使用表
 - High_Speed_IO_Module (LSModuleTypeHighSpeedIO)
 - Counter0 (LS_Counter)
 - LS_Encoder (LS_Encoder)
 - LocalBus_Master_Module (LocalBusR1 Master)
 - ModbusCOM_Master_1 (ModbusCOM RS-485 Master)
 - ModbusCOM_Slave (ModbusCOM Slave)

PLC_PRG x 库管理器 本地模块配置 High_Speed_IO_Module 通讯配置

1

LS_Counter_0 ← 获取高数计数值指令

LS_Encoder ↔ Axis

LS_Counter0_Xenable → xEnable

xValid

xBusy

xError

eErrorID

fValue 当前计数值

fFrequency 当前计数频率

eDir 当前计数方向

2

设置高速计数预置值指令

LS_PresetValue_0 ← 设置高速计数预置值指令

LS_Encoder ↔ Axis

LS_PresetValue0_Xenable → xExecute

预置值 → eTriggerType

xDone

xBusy

xCommandAborted

xError

eErrorID

fPresetValue

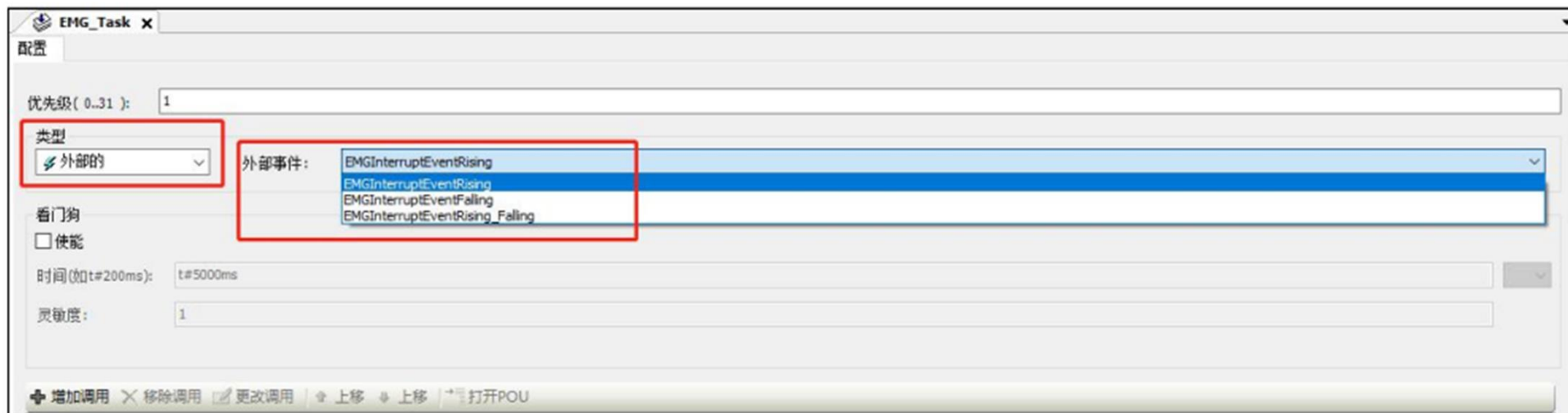
任务放HSIOTask下

对应轴名

07

中断任务

SC5-C 带 EMG 中断输入口，支持 NPN/PNP 输入类型，可用于接收外部中断信号并以最高的任务优先级执行中断任务。



1. 外部中断任务的添加

新建任务时，选择任务类型为“外部的”，外部事件可选择上升沿中断、下降沿中断和双边沿中断。当选择上升沿中断时，EMG 输入口上升沿执行中断任务调用的 POU。

外部中断任务可用于执行紧急停止，也可当普通中断使用（即在中断任务关联的 POU 中的内部变量）。

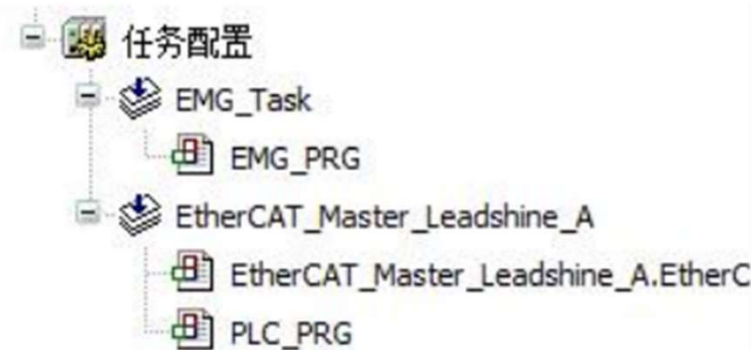
外部中断任务具有最高优先级，不受用户配置的优先级影响。触发外部时，中断任务执行 POU 中的运控指令会直接打断优先级 0 的 EtherCAT 总线任务中正在执行的运动，达到紧急控制的目的。

2. 中断程序示例

以急停程序为例，分别用EMG作普通急停和作中断任务急停，对比两种方式的急停效果。

两种急停程序的任务配置方式相同。

任务配置如下：



2.1 使用 EMG 输入口传递变量实现普通轴急停

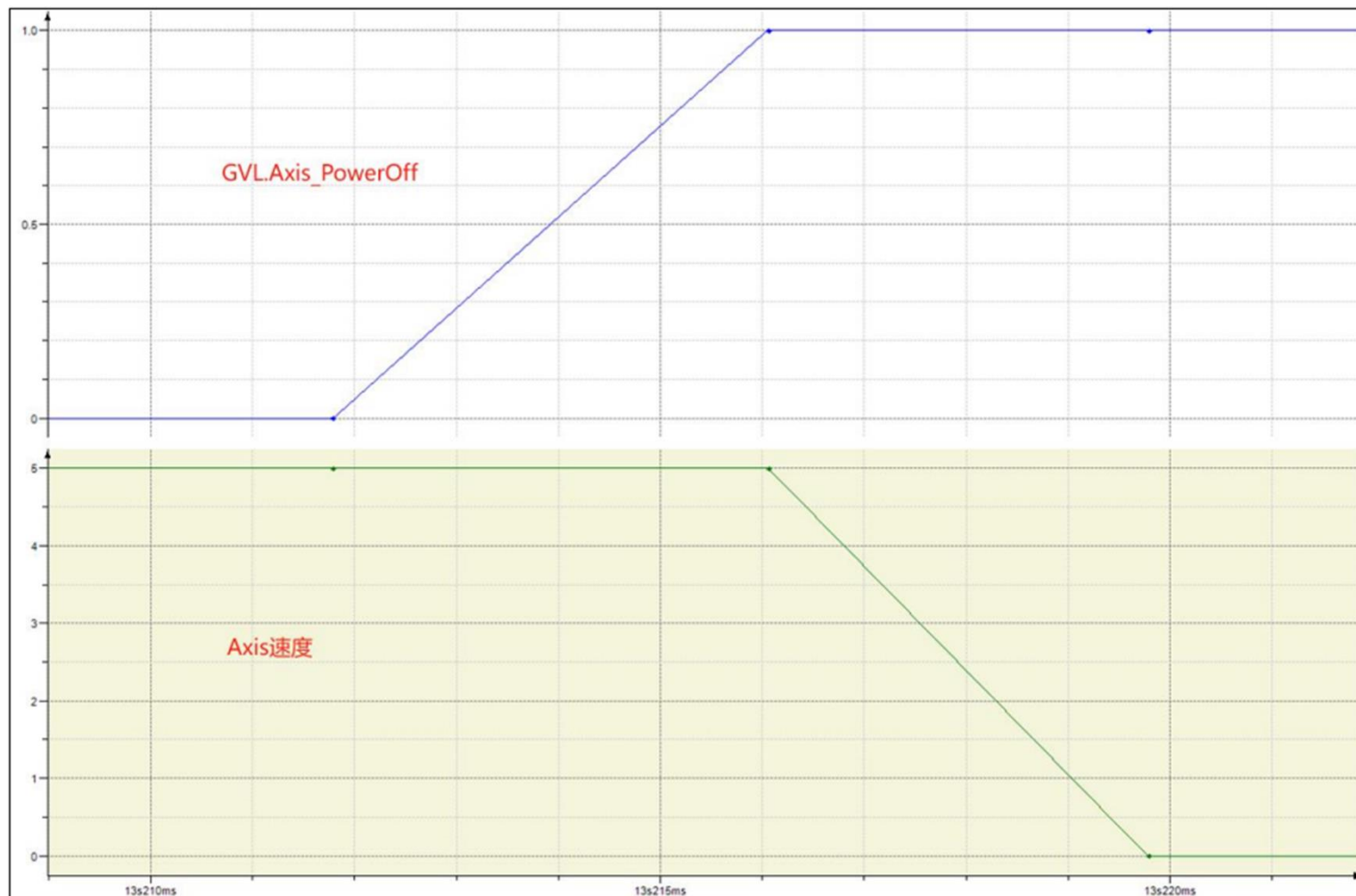
EMG_PRG程序代码：

```
1 GVL.Axis_PowerOff:=TRUE;  
2
```

PLC_PRG 程序代码：

```
PLC_PRG x  
1 PROGRAM PLC_PRG  
2 VAR  
3   MC_Power_1: MC_Power;  
4   MC_Jog_1: MC_Jog;  
5   JogF: BOOL;  
6   JogB: BOOL;  
7 END_VAR  
8  
9 MC_Power_1(Axis:= Axis, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= NOT GVL.Axis_PowerOff,  
10   Status=>, bRegulatorRealState=>, bDriveStartRealState=>, Busy=>, Error=>, ErrorID=> );  
11  
12 MC_Jog_1(Axis:= Axis, JogForward:= JogF, JogBackward:= JogB, Velocity:= 5,  
13   Acceleration:= 20, Deceleration:= 20, Jerk:= 200, Busy=>, CommandAborted=> ,  
14   Error=>, ErrorID=> );
```

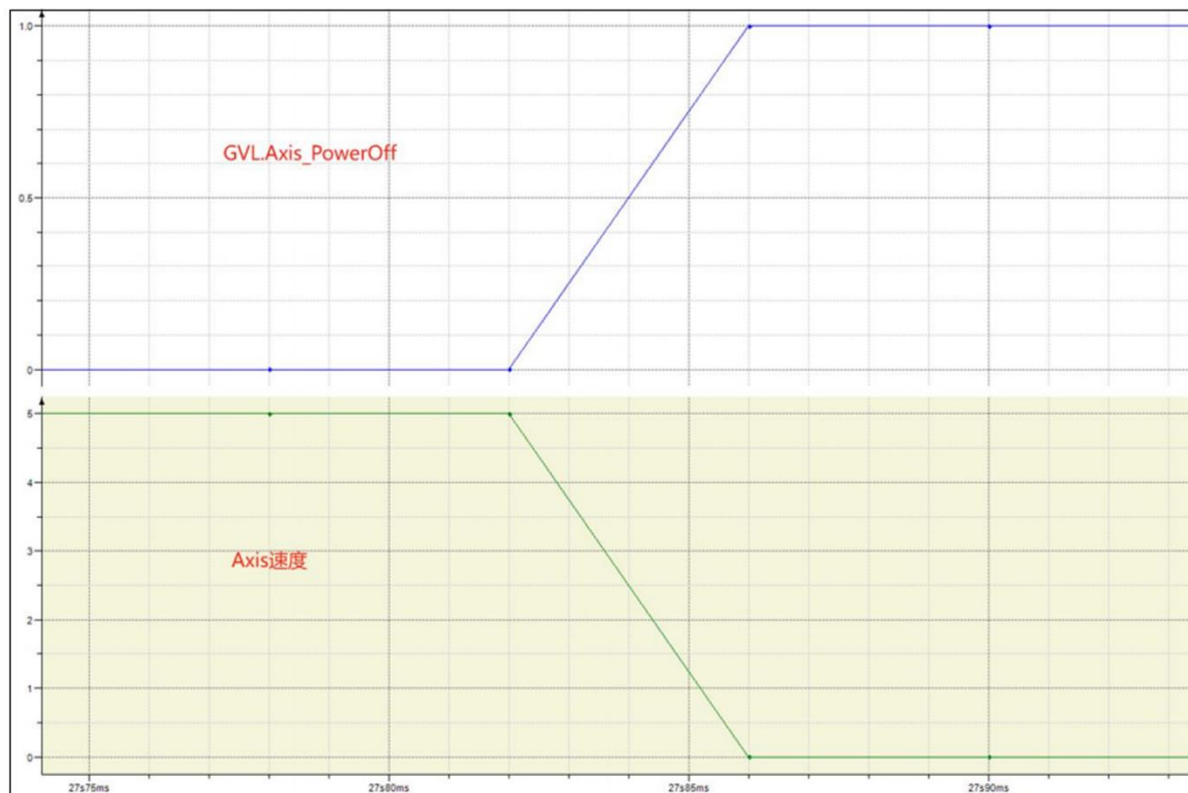
当 EMG 输入端口触发上升沿时，变量GVL.Axis_PowerOff和轴Axis速度的变化如下。变量 GVL.Axis_PowerOff 变为 TRUE 和轴速度降为 0 之间有一个总线周期（4ms）的时间间隔。



2.2使用EMG输入口实现中断任务轴急停 EMG_PRG程序代码：

```
EMG_PRG x
1 PROGRAM EMG_PRG
2 VAR
3   MC_Power_1: MC_Power;
4 END_VAR
5
1 GVL.Axis_PowerOff:=TRUE;
2 MC_Power_1(Axis:= Axis, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= NOT GVL.Axis_PowerOff,
3   Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );
```

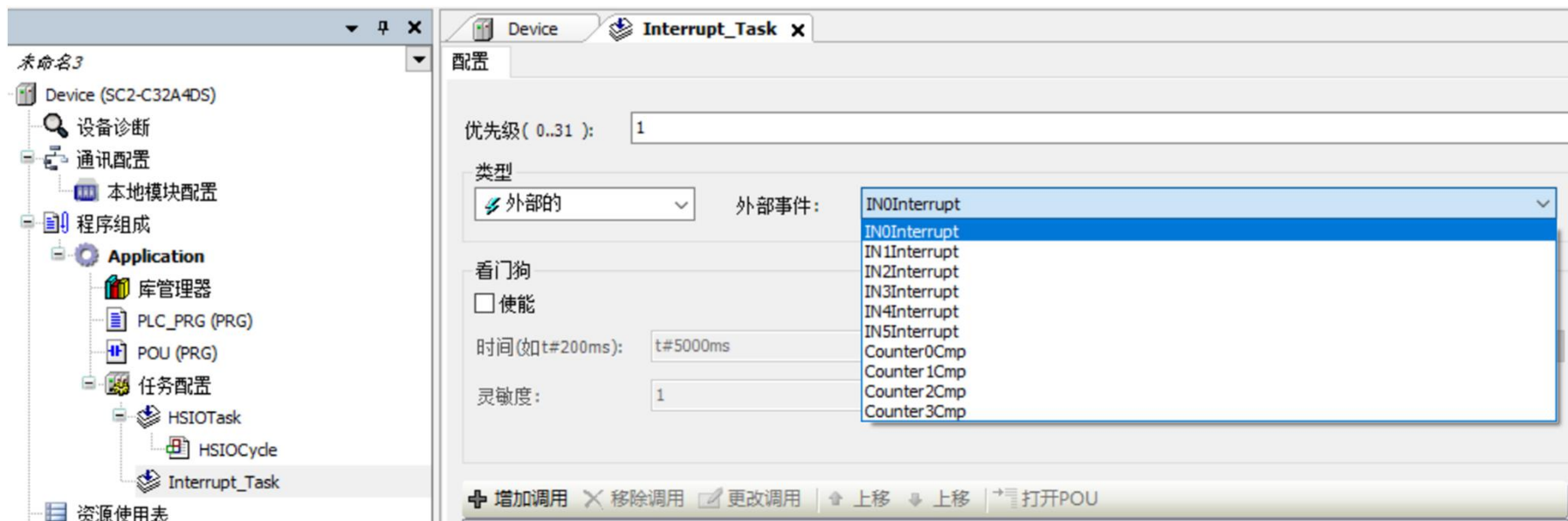
当 EMG 输入端口触发上升沿时，变量 GVL.Axis_PowerOff 和轴 Axis 速度的变化如下。变量 GVL.Axis_PowerOff 变为 TRUE 和轴速度降为 0 同时发生，实现了急停的即时生效。



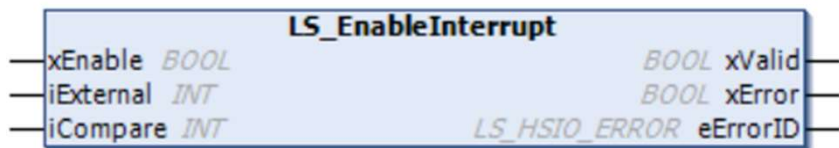
SC2-C 支持6路 (IN0-IN5) 中断输入口, 支持 上升沿/下降沿 输入类型, 4路高速计数数值比较中断, 可用于最高的任务优先级执行中断任务。

1.外部中断任务的添加

右击任务配置——添加一个新任务——选择类型-外部的——选择事件——增加调用



2.程序编写——功能块调用



输入/输出 图形的 文档

FUNCTION_BLOCK LS_EnableInterrupt

名称	类型	继承自	地址	初始化	注释
xEnable	BOOL				使能中断 True:使能 FALSE:无效
iExternal	INT				打开外部输入中断, 比如3, 二进制为2#11,
iCompare	INT				打开比较中断, SC2系列暂不支持
xValid	BOOL				
xError	BOOL				
eErrorID	LS_HSIO_ERROR				

中断使能, 本功能块用于打开外部输入中断; 当发生中断时, 执行关联的中断任务



```
1 PROGRAM PLC_PRG
2 VAR
3     LS_EnableInterrupt_0:LS_EnableInterrupt;
4     VAR1:BOOL;
5 END_VAR

1 LS_EnableInterrupt_0(
2     xEnable:=VAR1 ,
3     iExternal:=2#001011 ,
4     iCompare:=2#0101 ,
5     xValid=> ,
6     xError=> ,
7     eErrorID=> );
```

表示IN0、IN1、IN3的中断输入打开

表示第1、3路高速计数值比较中断打开

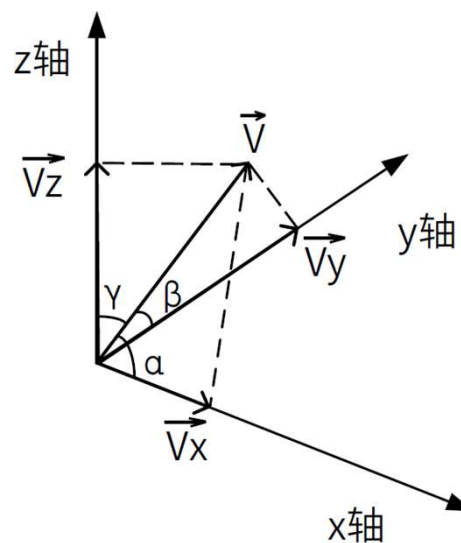
新建一个PRG调用中断使能功能块, 将其放入其他任务下

08

插补

什么是插补?

插补就是输入基本数据(如直线的起点、终点坐标, 圆弧的起点、终点、圆心坐标等), 运用一定的算法计算, 根据计算结果向相应的坐标发出进给指令。对应着每一进给指令, 控制对象在相应的坐标方向上移动一定的距离, 从而加工出需要的轮廓形状。



SC系列PLC提供了一套雷赛专用插补指令，包括：直线插补运动、平面和空间圆弧插补运动、两轴椭圆插补、三轴螺旋线插补运动、多轴G代码插补指令等。其中通用多轴插补运动指令包含在LS_IpoLib库之中，程序若用到插补相关运动功能，须在工程中添加LS_IpoLib库。



The screenshot shows the LS_IpoLib library structure on the left and the details of the LS_2AxisCircle (FB) function block on the right. The library structure includes folders for circular interpolation, fixed-point speed, data types, elliptical interpolation, linear interpolation, helical interpolation, and tracking motion. The details for LS_2AxisCircle (FB) include its function signature, a description of its modes, and a list of its InOut parameters.

LS_2AxisCircle (FB)

FUNCTION_BLOCK PUBLIC LS_2AxisCircle

两轴圆弧插补指令，采用绝对坐标。参数IpoCycle、Jerk、Velocity、Acceleration、Deceleration不可以设置为0。圆弧插补有4种模式：模式0，三点圆弧模式，辅助点输入圆弧经过的点坐标；模式1，圆心与终点模式，辅助点输入圆心的坐标；模式2，终点半径模式，需要输入圆弧的半径，半径正值表示大圆弧，负值表示小圆弧；模式3，目标角度模式，停止位置为相对于起点的位置，需要输入圆弧的方向和圆心坐标；

InOut:

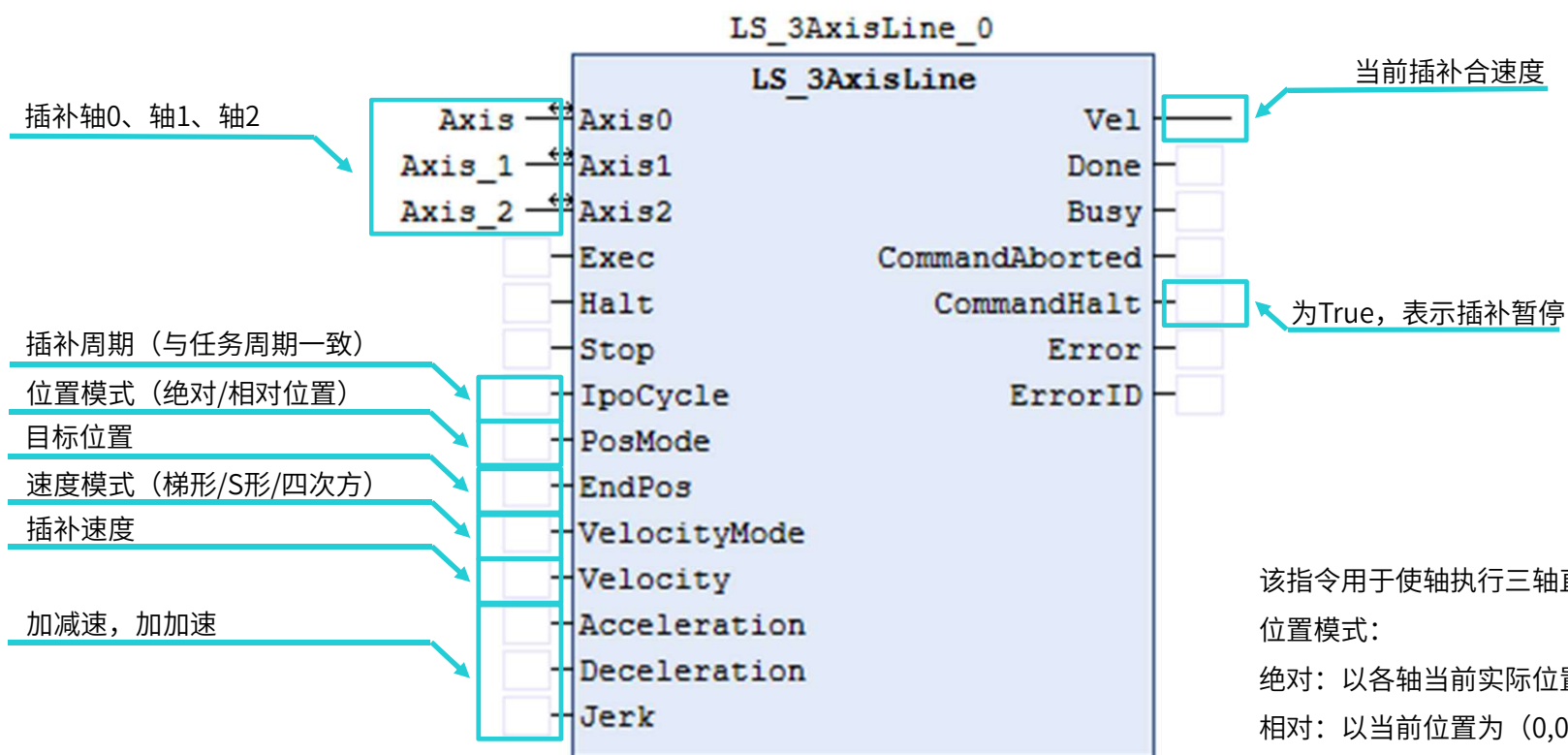
- LS_2AxisCircle.ACT Halt (ACT)
- LS_2AxisCircle.ACT IPO (ACT)
- LS_2AxisCircle.ACT Launch (ACT)
- LS_2AxisCircle.ACT Path (ACT)
- LS_2AxisCircle.ACT Stop (ACT)
- LS_2AxisCircle.ACT buffer (ACT)
- LS_2AxisCircle.clear_gcode (ACT)

- 支持脉冲轴与EtherCAT轴混合插补

指令名	功能说明
LS_2AxisLine	两轴直线插补运动
LS_3AxisLine	三轴直线插补运动
LS_4AxisLine	三轴直线插补、一轴跟随运动
LS_5AxisLine	三轴直线插补、两轴跟随运动
LS_6AxisLine	三轴直线插补、三轴跟随运动

SC5系列支持2~6轴直线插补运动，其中4~6轴直线插补运动中前三轴做空间直线插补运动，剩余对应的轴做跟随运动，和参与插补运动的轴同时启停。

LS_3AxisLine指令



该指令用于使轴执行三轴直线插补动作。

位置模式：

绝对：以各轴当前实际位置，计算目标位置

相对：以当前位置为 (0,0)，计算目标位置

SC5系列PLC的雷赛插补指令支持平面圆弧插补（两轴圆弧插补）和空间圆弧插补（三轴圆弧插补）。

指令名	功能说明
LS_2AxisCircle	两轴圆弧插补运动
LS_3AxisCircle	三轴圆弧插补运动

平面圆弧插补有4种模式：

模式0：三点圆弧模式，辅助点输入圆弧经过的点坐标；

模式1：圆心与终点模式，辅助点输入圆心的坐标；

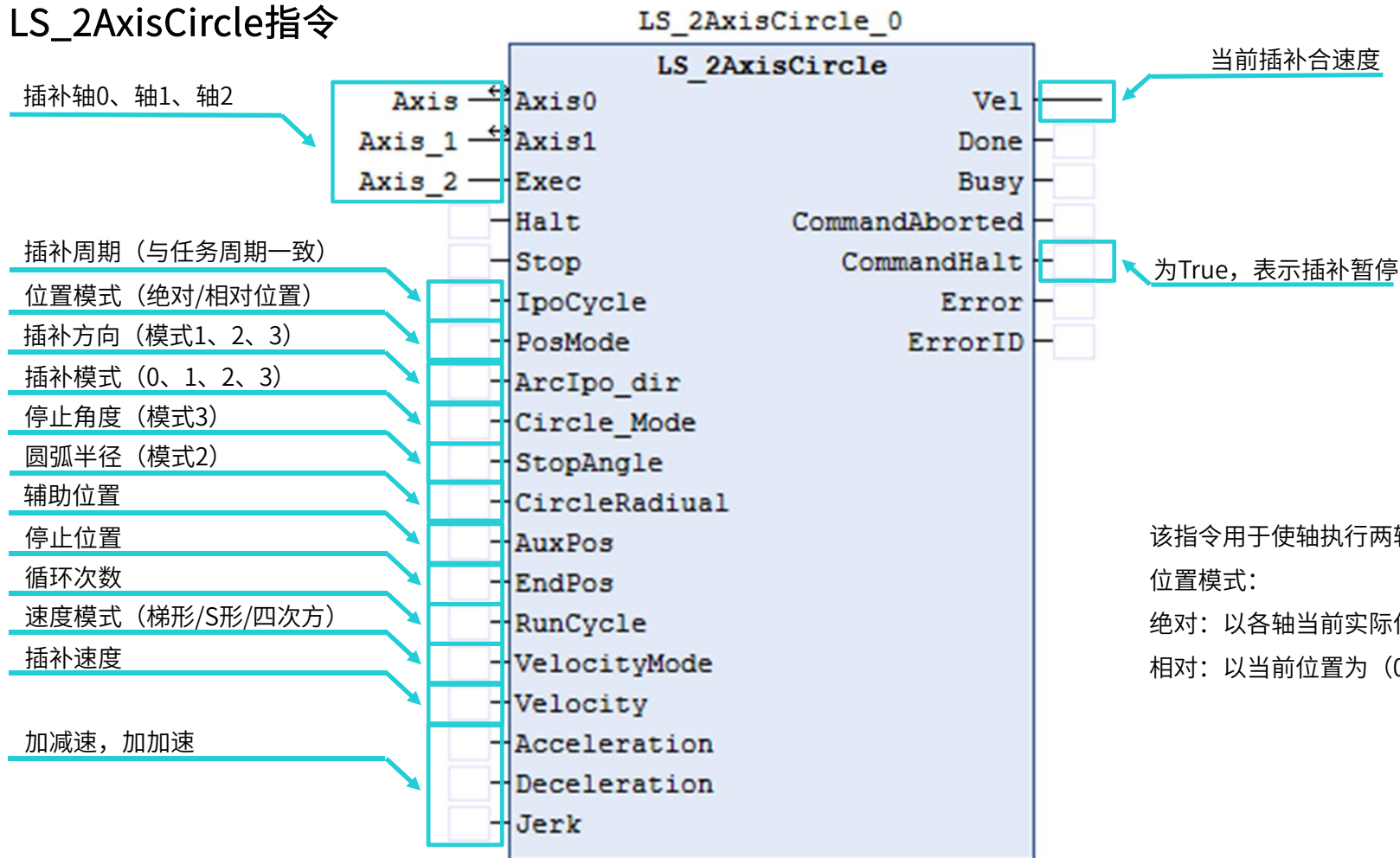
模式2：终点半径模式，需要输入圆弧的半径；

模式3：目标角度模式，停止位置为相对于起点的位置，需要输入圆弧的方向和圆心坐标。

空间圆弧插补只支持模式0：三点圆弧模式，具体参照运动指令说明。

- 各个模式中没有用到的参数就不要填。

LS_2AxisCircle指令



该指令用于使轴执行两轴圆弧插补动作。

位置模式：

绝对：以各轴当前实际位置，计算目标位置

相对：以当前位置为 (0,0)，计算目标位置

圆弧插补模式

指令的输入辅助位置AuxPos、终点位置Endpos都为数组，分别对应不同轴的各个关键点。

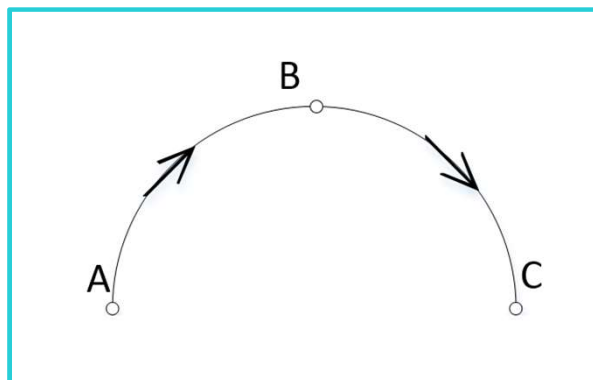
点类型	X轴	Y轴
当前位置	px	py
辅助点 (AuxPos)	AuxPos[0]	AuxPos[1]
终点 (Endpos)	Endpos[0]	Endpos[1]

圆弧插补模式

1、Circle_Mode = 0 (三点圆弧模式)

在使用三点圆弧模式时，需要在圆弧轨迹上确定3个不重合的点，如下图中的A、B、C三点。在指令中，A代表圆弧插补轴0和1的当前位置，不需要设置；B代表圆弧经过的点，需要设置为参数AuxPos；C代表圆弧的终点位置，需要设置为参数Endpos。

当起点、通过点与终点在同一条直线上时不能构成圆，指令报错，停止插补指令的执行。

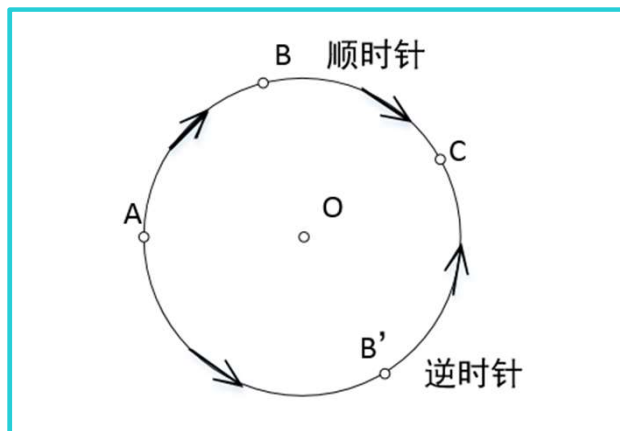


圆弧插补模式

1、Circle_Mode = 1（圆心与终点圆弧模式）

在使用圆心与终点圆弧模式时，需要设置圆心O、终点C和运动方向等参数。在指令中O代表圆心坐标，需要设置为参数AuxPos；C代表圆弧的终点坐标，需要设置为参数Endpos；运动方向需要设置参数ArcIpo_dir（0：逆时针；1：顺时针）。

注意：设置圆心坐标时要保证圆心到初始点A与终点C的距离相等。



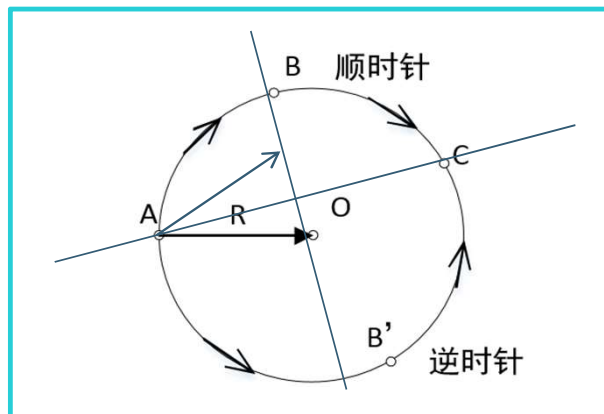
圆弧插补模式

1、Circle_Mode = 2 (终点半径圆弧模式)

在使用终点半径圆弧模式时，需要设置终点C，半径R和运动方向等参数。在指令中C代表圆弧的终点位置，需要设置为参数Endpos；半径R需要设置为参数CircleRadiual；运动方向需要设置为参数ArcIpo_dir。

注意：

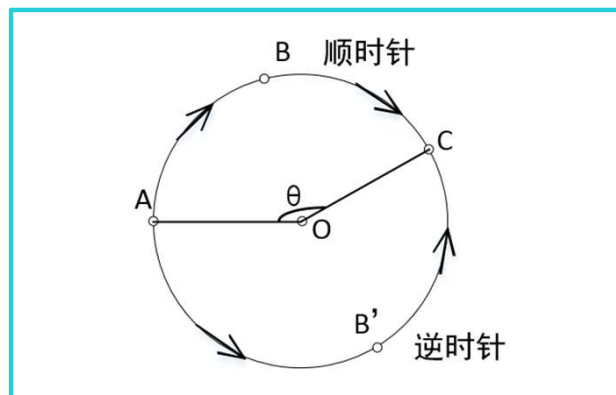
- ◆ 半径取值最小为“起始点-终点”距离的一半；
- ◆ 终点半径圆弧插补模式确定的圆有两个；半径为负表示圆弧角小于180度，半径为正表示圆弧角大于180度。



圆弧插补模式

1、Circle_Mode = 3 (目标角度圆弧模式)

在使用目标角度圆弧模式时，需要设置圆心O，角度 θ 和运动方向等参数。在指令中O代表圆心位置，需要设置为参数Auxpos；目标角度 θ 需要设置为参数StopAngle；运动方向需要设置为参数ArcIpo_dir。



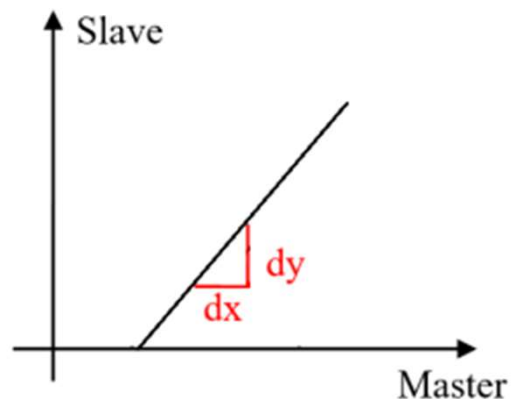
09

同步运动

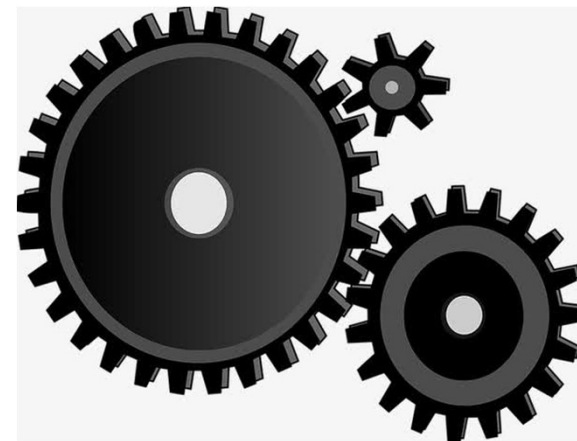
SC-C系列PLC均支持电子齿轮功能，在CPU负载不高于80%时可任意配置编码器轴与脉冲轴为主轴。

电子齿轮：

电子齿轮用来建立主轴和从轴的齿轮比的线性关系，电子齿轮可以大大地简化机械设计，而且可以根据用户需要快速变换齿轮比，大大提高了效率。



如上图所示：水平轴显示连接的主轴位置，垂直轴显示连接的从轴位置。当主轴信号有规律的运动（匀速运动），从轴的速度按照指定也是匀速的（固定的位置改变）。线性曲线的斜率决定电子齿轮比率。



简单应用如：
传送带，保证主从转送带速度同步。

齿轮啮合：

通过MC_GearIn指令设置主从轴齿轮比并启动电子齿轮。



RatioNumerator: 齿轮比分子

RatioDenominator: 齿轮比分母

VAR_IN_OUT	名称	类型	有效范围	初始化	注释
Master	主轴	AXIS_REF	-	-	映射到主轴, 参阅“AXIS_REF_SM3”
Slave	从轴	AXIS_REF	-	-	映射到主轴, 参阅“AXIS_REF_SM3”
VAR_INPUT					
Execute	启动	BOOL	TRUE FALSE	FALSE	此输入的一个上升沿将启动功能块的处理
RatioNumerator	分子	INT	正数	1	齿轮比分子
RatioDenominator	分母	UINT	正数	1	齿轮比分母
Acceleration	目标加速度	LREAL	负数, 0, 正数	0	电子齿轮比加速度
Deceleration	目标减速度	LREAL	负数, 0, 正数	0	电子齿轮比减速度
Jerk	目标加加速度	LREAL	负数, 0, 正数	0	加加速度
BufferMode	缓存模式	MC_BUFFER_MODE			
VAR_OUTPUT					
InGear	齿轮啮合	BOOL	TRUE FALSE	FALSE	TRUE 表示齿轮比的处理已经完成
Busy	执行中	BOOL	TRUE FALSE	FALSE	TRUE 表示功能块的处理没有完成
Active		BOOL			该轴在工作
CommandAborted	指令被中断	BOOL	TRUE FALSE	FALSE	TRUE 表示命令被另一个命令中断
Error	错误	BOOL	TRUE FALSE	FALSE	功能块内部发生错误信号
ErrorID	错误码	SMC_ERROR	-	0	错误 ID, 参阅“SMC_ERROR”

齿轮脱耦：

通过MC_GearOut指令解除主从轴的齿轮耦合。



VAR_IN_OUT	名称	类型	有效范围	初始化	注释
Slave	从轴	AXIS_REF	-	-	映射到主轴，参阅“AXIS_REF_SM3”
VAR_INPUT					
Execute	启动	BOOL	TRUE FALSE	FALSE	此输入的一个上升沿将启动功能块的处理
VAR_OUTPUT					
Done	执行完成	BOOL	TRUE FALSE	FALSE	TRUE 表示电子凸轮断开
Busy	执行中	BOOL	TRUE FALSE	FALSE	TRUE 表示功能块的处理没有完成
Error	错误	BOOL	TRUE FALSE	FALSE	功能块内部发生错误信号
ErrorID	错误代码	SMC_ERROR	-	0	错误 ID，参阅“SMC_ERROR”

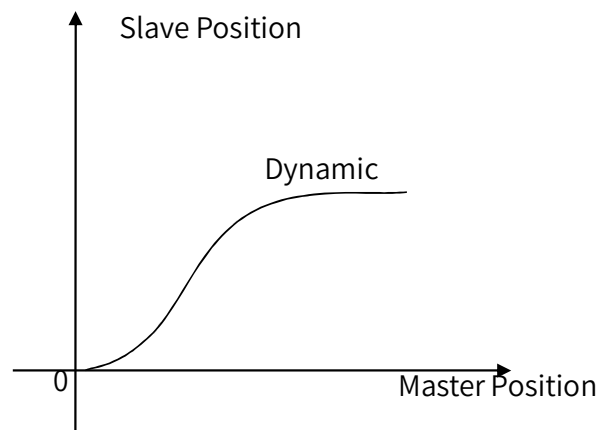
- ◆ 切出电子齿轮动完成后此时从轴的速度为切出前的速度，所以需配合以 MC_Stop 指令停止从轴。
- ◆ 本指令对MC_GearIn指令的主轴动作没有影响。

SC-C系列PLC均支持电子凸轮功能，在CPU负载不高于80%时可任意配置编码器轴与脉冲轴为主轴。

电子凸轮：

电子凸轮（ECAM）是利用构造的凸轮曲线来模拟机械凸轮，以达到与机械凸轮系统相同的凸轮轴与主轴之间相对运动的软件功能。

电子凸轮相当于无数个各种类型的机械凸轮的集合体，只需进行参数配置，即可适用于各种方案。

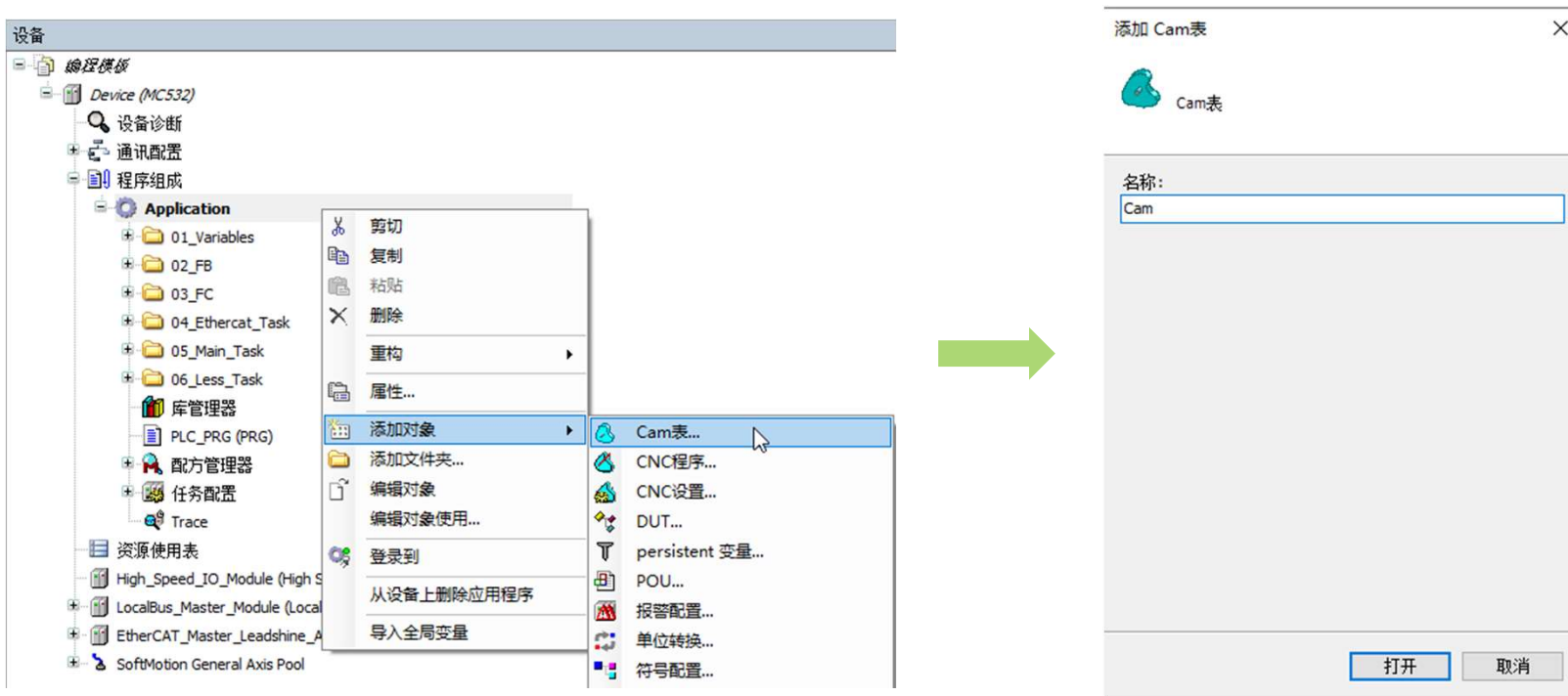


上图中，水平轴显示主轴位置值，垂直轴显示从轴位置值。凸轮在特定的区域内（凸轮控制周期）针对每个主轴相位有一个独立的从轴位置值，当凸轮驱动连接激活时，从轴必须遵循此轮廓。



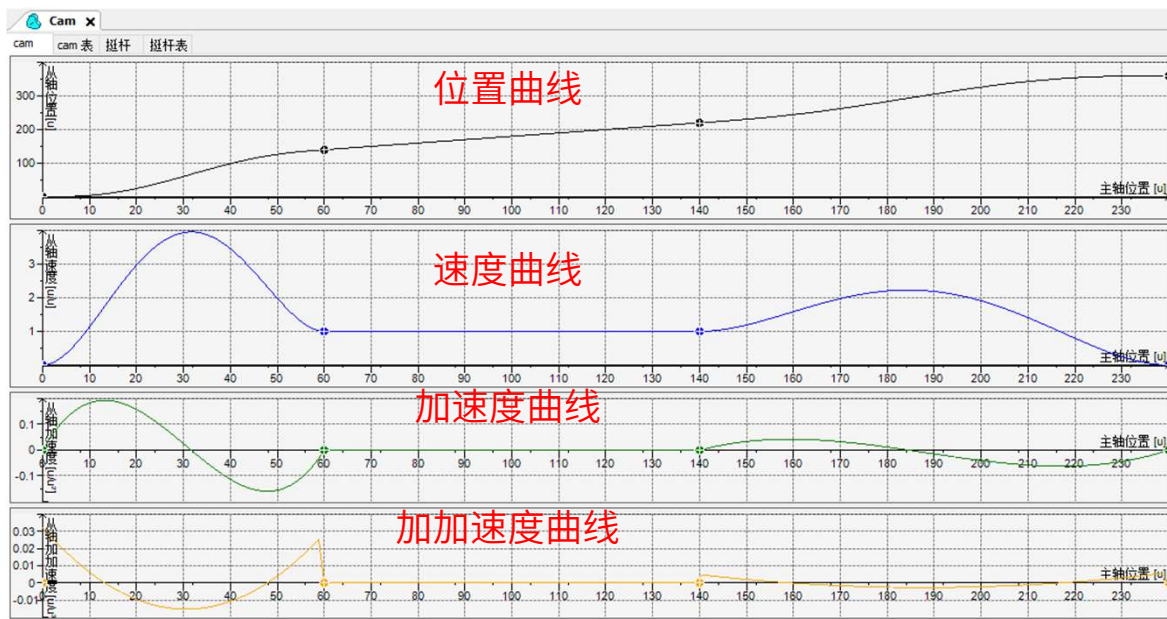
新建凸轮表：

凸轮表通过LeadSys软件进行添加及配置。右键“Application—添加对象—Cam表”。



凸轮编辑器：

可以修改凸轮表关键点参数。



- ◆ 在凸轮功能中，将由主轴相位和从轴位移组成的一对数据称之为凸轮关键点，凸轮表即为多个凸轮关键点数据组成的数据组合；
- ◆ 凸轮动作中，根据主轴相位和设定的曲线类型计算出从轴的位移，从而控制从轴的动作；
- ◆ 可以对关键点进行拖动，修改关键点的数据。

凸轮编辑器：

可以修改凸轮表关键点参数。

Cam x											
cam	cam 表	挺杆	挺杆表								
		X	Y	V	A	J	段类型	最小(位置)	最大(位置)	最大(速度)	最大(加速度)
		0	0	0	0	0					
							Poly5	0	140	3.9579444027047339	0.19333623627273949
		60	140	1	0	0					
							Line	140	220	1	0
		140	220	1	0	0					
							Poly5	220	360	2.2290809327846368	0.062268886719375117
		240	360	0	0	0					

选项	功能
	删除关键点
	插入关键点
X	主轴相位
Y	从轴位移
V	连接点速度比
A	连接点加速度比
J	连接点加加速度比
段类型	凸轮曲线类型, Poly5 (五次曲线)、Line (直线)
最小 (位置)	该段中从轴的最小位置, 不可修改
最大 (位置)	该段中从轴的最大位置, 不可修改
最大 (速度)	从轴速度比的最大值
最大 (加速度)	从站加速度比的最大值

- ◆ 通过软件生成CAM表时, Leadsys 会自动生成 MC_Cam_REF 类型的凸轮表数据,命名为“CAM表定义名称”及凸轮表关键点 SMC_CAMXYVA 类型的数组,命名为“CAM表定义名称+_A”;
- ◆ 用户可通过自动生成的 CAM_A 在程序中修改关键点数据;
- ◆ 如将第三个关键点位置修改为 (200,200)
CAM_A[2].dx:=200
CAM_A[2].dy:=200;

MC_CAM_REF (凸轮表数据结构)

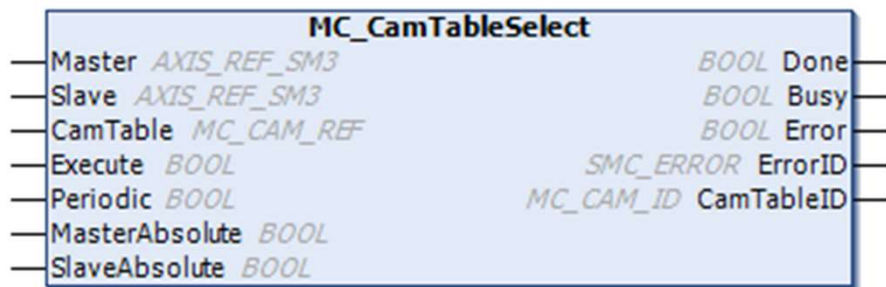
变量	数据类型	初始值	描述
pce	POINTER TO BYTE	0	指针指向类型取决于 byType ; 1: pce指向SMC_CAMTable_..._1例如: SMC_CAMTable_LREAL_128_1 2: pce指向SMC_CAMTable_..._2例如: SMC_CAMTable_UDINT_256_2 3: pce指向一个SMC_CAMXYVA数组
pt	POINTER TO SMC_CAMTableTapet	0	指向挺杆
dwTapetActiveBits	DWORD		内部变量
strCAMName	STRING	“	凸轮表名称
byInterpolationQuality	BYTE	1	1: 线性插值 3: 三次插值
byCompatibilityMode	BYTE	0	兼容模式: Bit0: TRUE, 定期执行具有主宽度的表
bChangedOnline	BOOL	FALSE	内部变量
xPartofLM	BOOL		内部变量

SMC_CAMXYVA (凸轮关键点数据结构)

变量	数据类型	初始值	描述
dY	LREAL	0	主轴相位
dX	LREAL	0	从轴位移
dV	LREAL	0	一阶导数 dY/dX , 从轴速度
dA	LREAL	0	二阶导数 d^2Y/dX^2 , 从轴加速度

凸轮表选择指令：

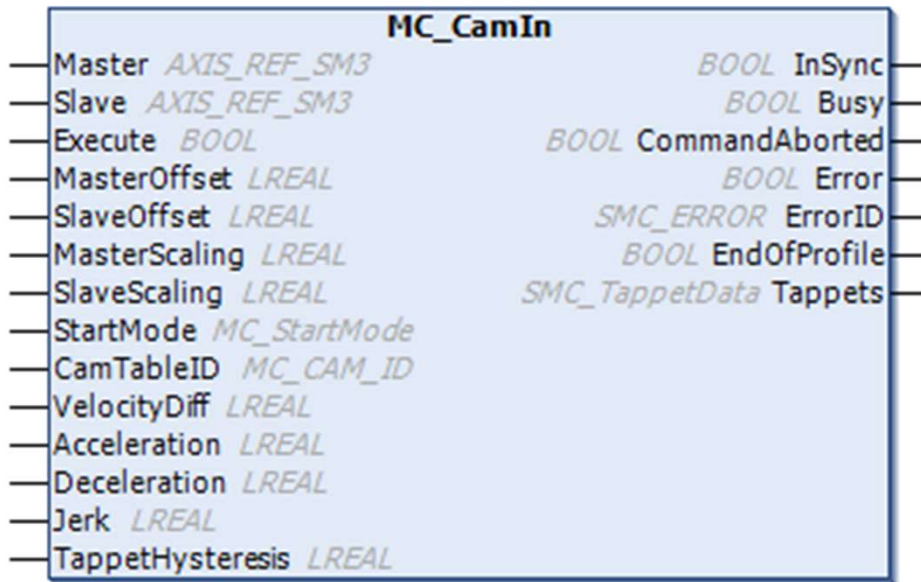
通过MC_CamTableSelect指令选择需要执行的凸轮表。



VAR_IN_OUT	名称	类型	有效范围	初始化	注释
Master	主轴	AXIS_REF	-	-	映射到主轴，参阅“AXIS_REF_SM3”
Slave	从轴	AXIS_REF	-	-	映射到从轴，参阅“AXIS_REF_SM3”
CamTable	凸轮表	MC_CAM_REF	-	-	映射为 cam 表格描述，参阅 MC_CAM_REF
VAR_INPUT					
Execute	启动	BOOL	TRUE FALSE	FALSE	此变量的一个上升沿将会启动功能块的处理
Periodic	重复模式	BOOL	TRUE	TRUE	True: 周期、重复地执行所指定的凸轮表
			FALSE		FALSE: 仅执行一次凸轮表
MasterAbsolute	主轴绝对模式	BOOL	TRUE	TRUE	TRUE 表示绝对坐标，FALSE 表示相对坐标
			FALSE		
SlaveAbsolute	从轴绝对模式	BOOL	TRUE	TRUE	TRUE 表示绝对坐标，FALSE 表示相对坐标
			FALSE		
VAR_OUTPUT					
Done	执行完成	BOOL	TRUE FALSE	FALSE	TRUE 表示 cam 表格选择结束
Busy	执行中	BOOL	TRUE FALSE	FALSE	TRUE 表示功能块的处理没有完成
Error	错误	BOOL	TRUE FALSE	FALSE	功能块内部发生错误信号
ErrorID	错误代码	SMC_ERROR	-	0	错误 ID，参阅 SMC_Error
CamTableID	挺杆 ID	MC_CAM_ID	-	-	cam 表格的定义，用于功能块 MC_CamID

电子凸轮耦合：

通过MC_CamIn指令使主轴和从轴按照凸轮表进行同步凸轮动作。



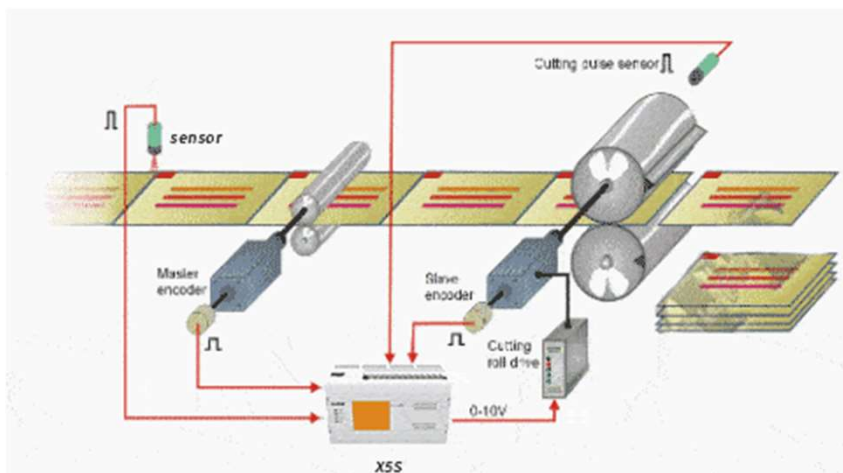
VAR_IN_OUT	名称	类型	有效范围	初始化	注释
Master	主轴	AXIS_REF	-	-	映射到主轴，参阅“AXIS_REF_SM3”
Slave	从轴	AXIS_REF	-	-	映射到主轴，参阅“AXIS_REF_SM3”
VAR_INPUT					
Execute	启动	BOOL	TRUE FALSE	FALSE	一个上升沿将启动功能块的处理
MasterOffset	主轴偏移	LREAL	负数， 0，正数	0	主轴表格的偏移
SlaveOffset	从轴偏移	LREAL	负数， 0，正数	0	从轴列表的偏移
MasterScaling	主轴比例	LREAL	正数	1	主轴缩放比例
SlaveScaling	从轴比例	LREAL	正数	1	从轴缩放比例
StartMode	从轴啮合方式	MC_StartMode	0-4	0: absolute	0: absolute 绝对位置；1: relative 相对位置；2: ramp_in(斜坡切入)；3: ramp_in_pos(正向斜坡切入)；4: ramp_in_neg 反向斜坡切入
CamTableID	凸轮表	MC_CAM_ID	-	-	定义 cam 表格的使用，是 MC_CamID 的输出
VelocityDiff	速度	LREAL	负数，0 正数	0	与 ramp_in 不同的最大速度
Acceleration	加速度	LREAL	0，正数	0	对 ramp_in 的最大加速度
Deceleration	减速度	LREAL	0，正数	0	对 ramp_in 的最大减速度
Jerk	目标加速度	LREAL	负数， 0， 正数	0	加速度
TappetHysteresis	挺杆阻尼	LREAL	0 正数	0	挺杆滞后的尺寸大小
VAR_OUTPUT					
InSync	同步中	BOOL	TRUE FALSE	FALSE	TRUE 表示从轴根据 cam 表格与主轴同步
Busy	执行中	BOOL	TRUE FALSE	FALSE	TRUE 表示功能块的处理没有完成
CommandAborted	指令被中断	BOOL	TRUE FALSE	FALSE	TRUE 表示命令被另一个命令中断
Error	错误	BOOL	TRUE FALSE	FALSE	功能块内部发生错误
ErrorID	错误代码	SMC_ERROR	-	0	错误 ID，参阅 SMC_Error
EndOfProfile	曲线完成	BOOL	TRUE FALSE	FALSE	在 cam 编译周期结束输出的脉冲信号
Tappets	挺杆表	SMC_TappetData	-	-	用于 SMC_GetTappetValue 处理的挺杆信号

电子凸轮脱离：

通过 MC_CamOut 指令解除指定的从轴与其对应主轴的凸轮耦合关系。



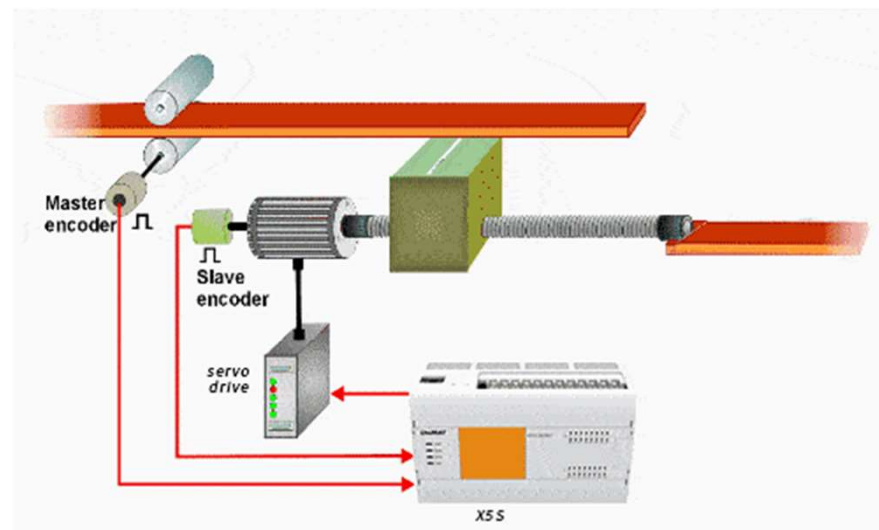
VAR_IN_OUT	名称	类型	有效范围	初始化	注释
Slave	从轴	AXIS_REF	-	-	映射到主轴，参阅“AXIS_REF_SM3”
VAR_INPUT					
Execute	启动	BOOL	TRUE FALSE	FALSE	此变量的一个上升沿将会启动功能块的处理
VAR_OUTPUT					
Done	执行完成	BOOL	TRUE FALSE	FALSE	TRUE，如果 cam 已经被分离
Busy	执行中	BOOL	TRUE FALSE	FALSE	TRUE，如果功能块的处理没有完成
Error	错误	BOOL	TRUE FALSE	FALSE	功能块内有错误信号发生
ErrorID	错误码	SMC_ERROR	-	0	错误 ID，参阅 SMC_Error



飞剪：剪切机构一般为圆周运动，与被剪切物体同向运动，通过改变剪切机构运行中的速度，达到改变剪切长度的目的。

共同点：飞剪与追剪都分为非同步区与同步区的概念，要求同步区剪切机构与被剪切机构速度相同。

不同点：飞剪是圆周运动，同步区小，但是可以做高速运动；追剪是往复运动，同步区大，可以完成较为复杂的剪切、冲压等动作。



追剪：剪切机构平行于被剪切物体，剪切机构做往复运动，通过改变在非同步区的速度达到改变剪切长度的目的。

10

Modbus 通讯示例

Modbus从站



SC作为Modbus从站，只需要在通讯配置里勾选Modbus从站，配置完成后会在设备树里生成对应的设备



波特率：表示每秒钟传送的符号的个数。例如300波特表示每秒钟发送300个符号。

奇偶校验位：是一种检错方式。

数据位：衡量通信中实际数据位的参数。

停止位：是一帧数据结束的标志。

标准的Modbus协议可以访问的PLC内部I、Q区范围

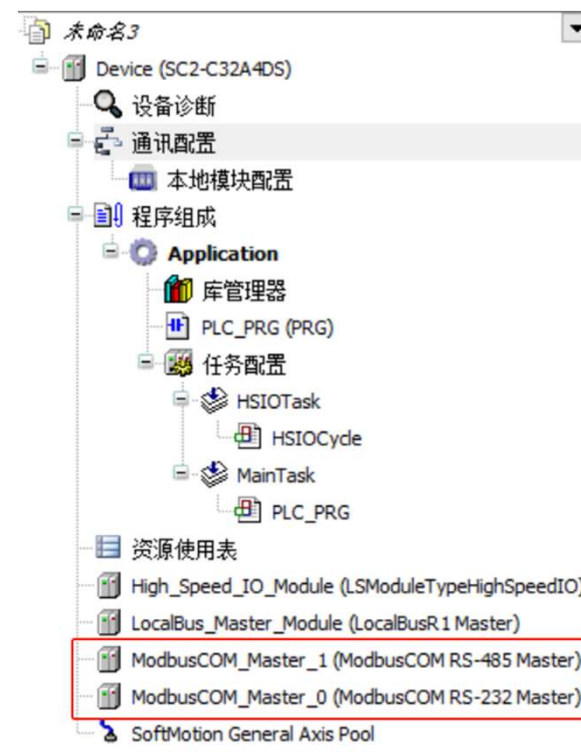
地址范围	功能码	起始地址	线圈数量	说明
QW0~QW4095 (QX0.0~QX8191.7)	0X01,0x05, 0x0f	0	65536	通用标准 Modbus 协议都可以访问
IW0~IW4095 (IX0.0~IX8191.7)	0X02	0	65536	通用标准 Modbus 协议都可以访问

标准的Modbus协议可以访问的PLC内部M区范围

地址范围	功能码	起始地址	寄存器数量	说明
MW0~MW65535	0x03,0x06, 0x10	0	65536	通用标准 Modbus 协议都可以访问



SC作为串口Modbus主站，需在通讯配置里勾选Modbus主站，并添加从站设备,配置完成后会在设备树里生成对应的设备



双击设备树添加的ModbusCOM_RS485_Master，在Modbus主站配置中可以设置串口通讯参数。

Device (SC2-C32A4DS)

- 设备诊断
- 通讯配置
 - 本地模块配置
- 程序组成
 - Application
 - 库管理器
 - PLC_PRG (PRG)
 - 任务配置
 - HSIOTask
 - HSIOCycle
 - MainTask
 - PLC_PRG
- 资源使用表
- High_Speed_IO_Module (LSModuleTypeHighSpeedIO)
- LocalBus_Master_Module (LocalBusR1 Master)
- ModbusCOM_Master_1 (ModbusCOM RS-485 Master)
- ModbusCOM_Master_0 (ModbusCOM RS-232 Master)
- SoftMotion General Axis Pool

PCI-BusIEC对象

Modbus主站配置

Internal参数

状态

信息

串口配置

波特率 9600

奇偶校验 偶校验

数据位 8

停止位 1

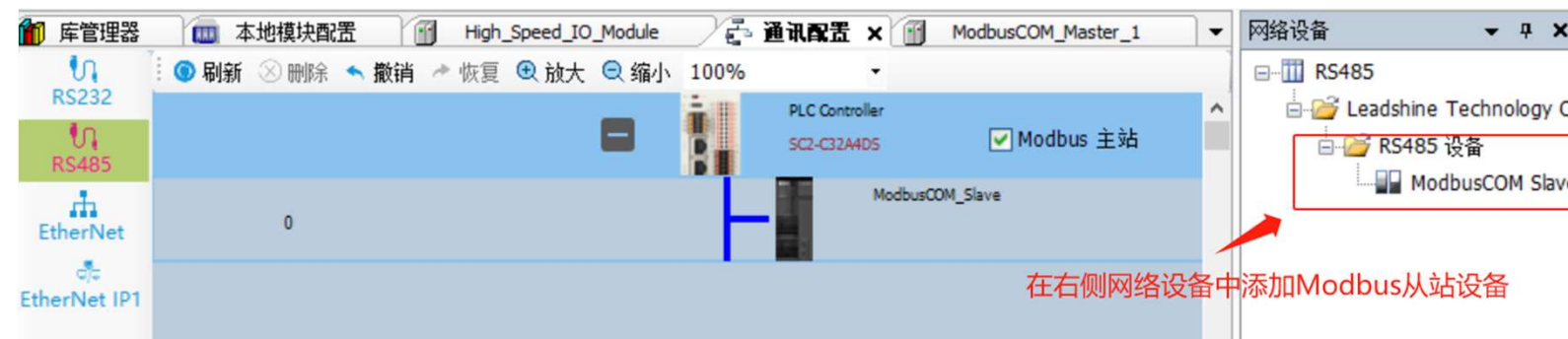
传输模式 RTU ASCII

Modbus主站配置

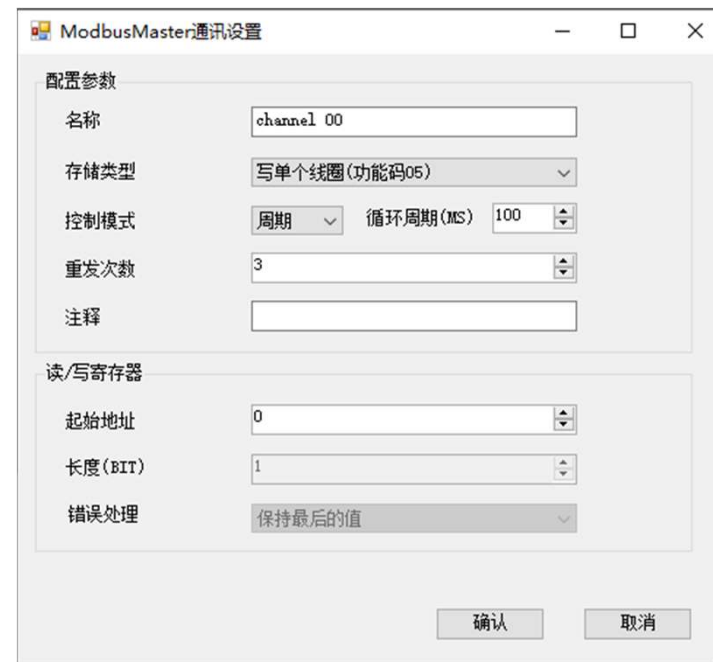
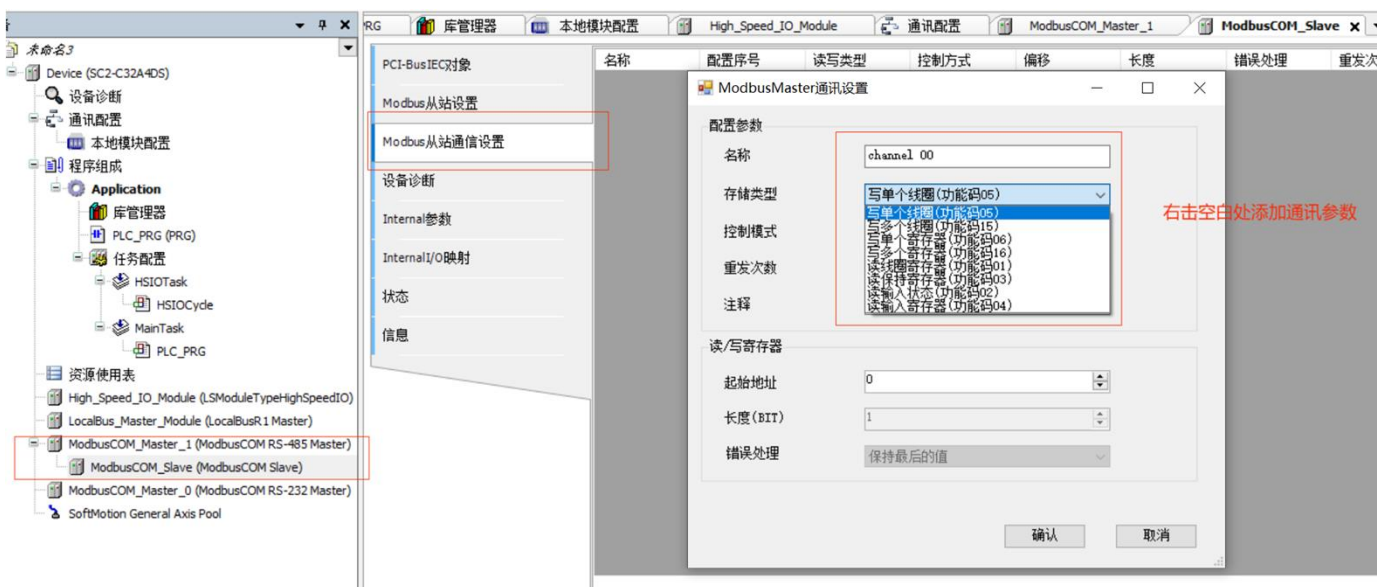
帧间隔 (ms) 5

双击Modbus主站设备打开配置

Modbus主站-串口



双击设备树添加的 ModbusCOM_Slave，在Modbus从站通讯设置中可以添加通讯项，读写从站数据。



在I/O映射中可以监视数据值与读写状态。

The screenshot shows the 'ModbusCOM_Slave' configuration window. The left sidebar has the following menu items: PCI-Bus IEC对象, Modbus从站设置, Modbus从站通信设置, 设备诊断, Internal参数, Internal I/O映射 (selected), 状态, and 信息. The main area displays a table of I/O mappings with columns: 变量 (Variable), 映射 (Mapping), 通道 (Channel), 地址 (Address), 类型 (Type), 单元 (Unit), and 描述 (Description). The table contains the following data:

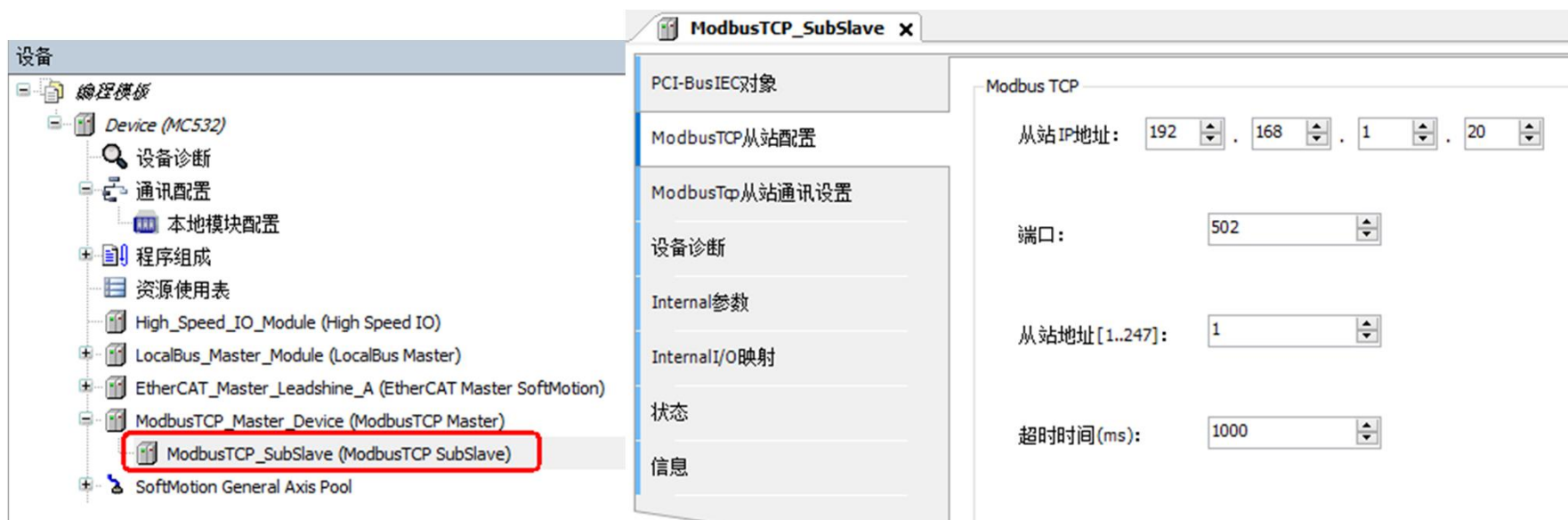
变量	映射	通道	地址	类型	单元	描述
错误码		channel 00	%IW78	WORD		ModBus_RTU通讯错误码
channel 00		channel 00	%QW39	ARRAY [0..0] OF WORD		写单个线圈
channel 00[0]		channel 00[0]	%QW39	WORD		
Bit0		Bit0	%QX78.0	BOOL		Write #0000
Bit1		Bit1	%QX78.1	BOOL		
Bit2		Bit2	%QX78.2	BOOL		
Bit3		Bit3	%QX78.3	BOOL		
Bit4		Bit4	%QX78.4	BOOL		
Bit5		Bit5	%QX78.5	BOOL		
Bit6		Bit6	%QX78.6	BOOL		
Bit7		Bit7	%QX78.7	BOOL		
Bit8		Bit8	%QX79.0	BOOL		
Bit9		Bit9	%QX79.1	BOOL		
Bit10		Bit10	%QX79.2	BOOL		
Bit11		Bit11	%QX79.3	BOOL		
Bit12		Bit12	%QX79.4	BOOL		
Bit13		Bit13	%QX79.5	BOOL		
Bit14		Bit14	%QX79.6	BOOL		
Bit15		Bit15	%QX79.7	BOOL		
channel 00		channel 00	%IX158.0	BOOL		处理完成状态

At the bottom of the window, there are controls for '复位映射' (Reset Mapping), '一直更新变量:' (Always Update Variable:), and '使能1(如果未在任何任务中使用则使用总线循环):' (Enable 1 (if not used in any task use bus cycle)). There are also icons for '创建新变量' (Create New Variable) and '映射到现有变量' (Map to Existing Variable).

Modbus TCP主站

SC作为Modbus TCP主站，需在通讯配置里勾选Modbus TCP主站，并添加从站设备，配置完成后会在设备树里生成对应的设备

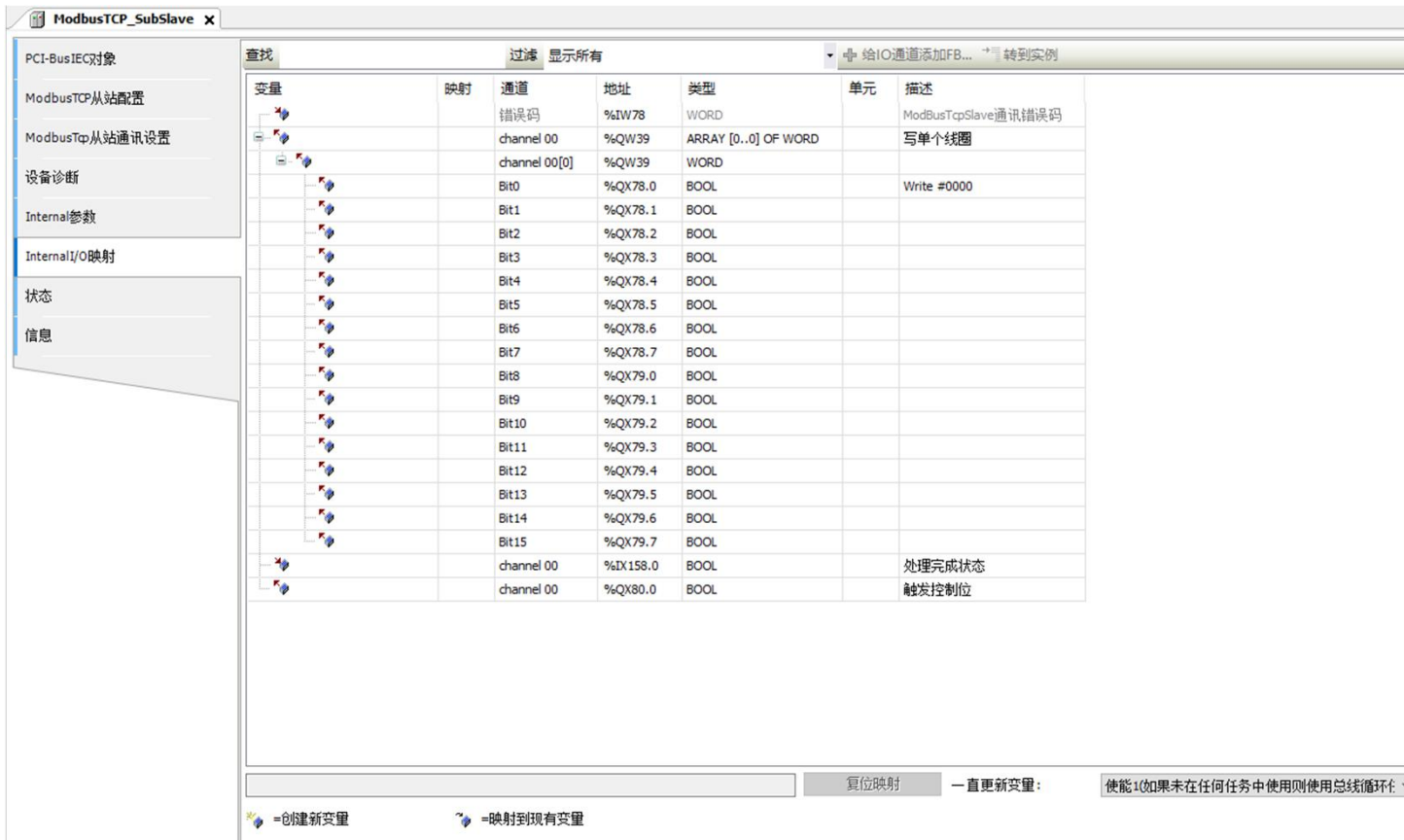
双击设备树添加的ModbusTCP_SubSlave，在Modbus TCP从站配置中可以设置通讯参数。



双击设备树添加的ModbusTCP_SubSlave，在Modbus TCP从站通讯设置中可以添加通讯项，读写从站数据。



在I/O映射中可以触发读写功能，监视数据值与读写状态。



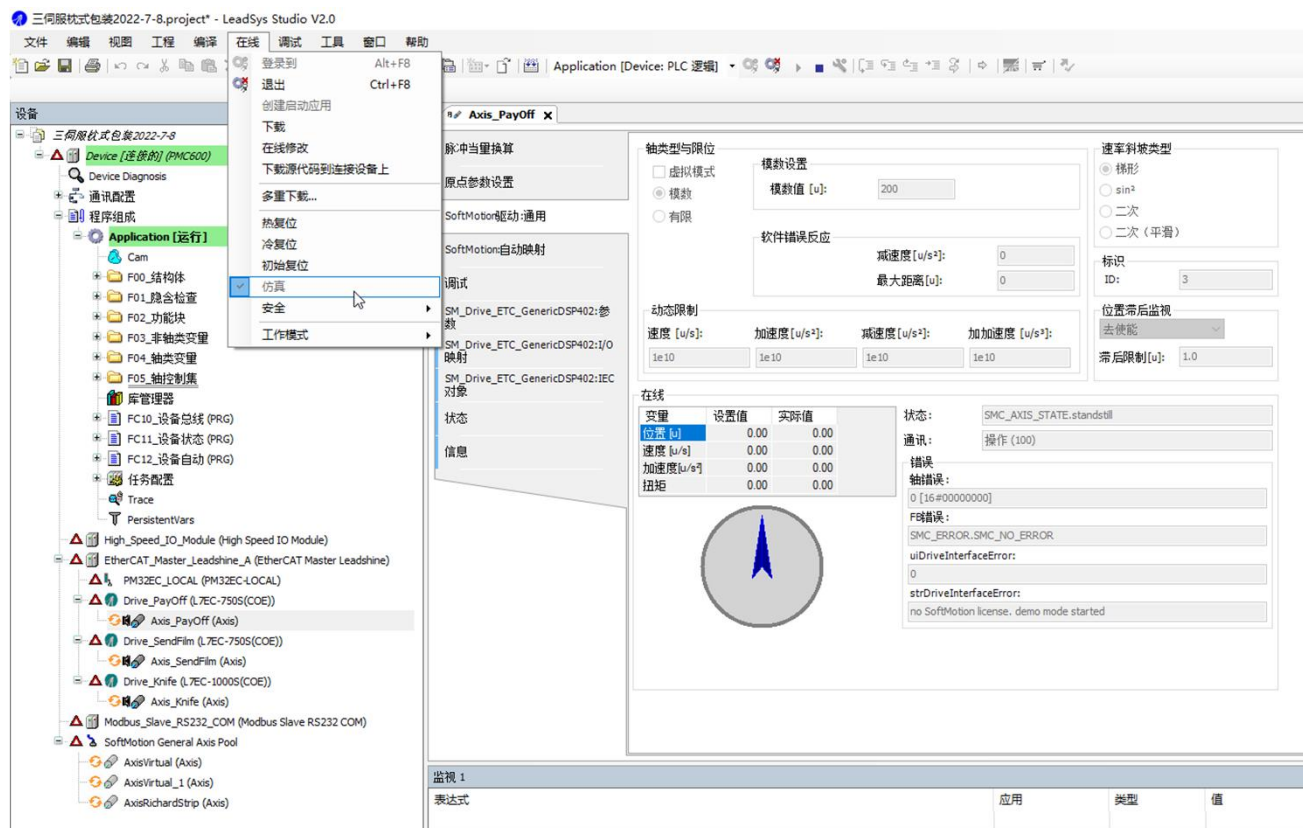
The screenshot displays the 'ModbusTCP_SubSlave' configuration window. On the left, a sidebar lists various configuration options, with 'Internal I/O映射' (Internal I/O Mapping) selected. The main area features a table for mapping variables to Modbus addresses. The table has columns for '变量' (Variable), '映射' (Mapping), '通道' (Channel), '地址' (Address), '类型' (Type), '单元' (Unit), and '描述' (Description). The table lists several bit variables (Bit0 to Bit15) and two channel variables (channel 00) with their respective addresses and data types. At the bottom, there are buttons for '创建新变量' (Create new variable) and '映射到现有变量' (Map to existing variable), along with a status bar containing '复位映射' (Reset mapping), '一直更新变量:' (Always update variable:), and '使能1(如果未在任何任务中使用则使用总线循环)' (Enable 1 (if not used in any task, use bus cycle)).

变量	映射	通道	地址	类型	单元	描述
错误码			%IW78	WORD		ModbusTcpSlave通讯错误码
channel 00			%QW39	ARRAY [0..0] OF WORD		写单个线圈
channel 00[0]			%QW39	WORD		
Bit0			%QX78.0	BOOL		Write #0000
Bit1			%QX78.1	BOOL		
Bit2			%QX78.2	BOOL		
Bit3			%QX78.3	BOOL		
Bit4			%QX78.4	BOOL		
Bit5			%QX78.5	BOOL		
Bit6			%QX78.6	BOOL		
Bit7			%QX78.7	BOOL		
Bit8			%QX79.0	BOOL		
Bit9			%QX79.1	BOOL		
Bit10			%QX79.2	BOOL		
Bit11			%QX79.3	BOOL		
Bit12			%QX79.4	BOOL		
Bit13			%QX79.5	BOOL		
Bit14			%QX79.6	BOOL		
Bit15			%QX79.7	BOOL		
channel 00			%IX158.0	BOOL		处理完成状态
channel 00			%QX80.0	BOOL		触发控制位

11

实用功能

“菜单栏--在线—仿真—编译—登录—运行”



通过LeadSys读写总线型伺服电机驱动器和步进电机驱动器的参数，简化调试。

L7EC_400 x

通用

在线

CoE在线

功能码

过程数据 (PDO)

启动参数 (SDO)

EtherCAT参数

EtherCATI/O映射

EtherCATIEC对象

状态

信息

全部参数

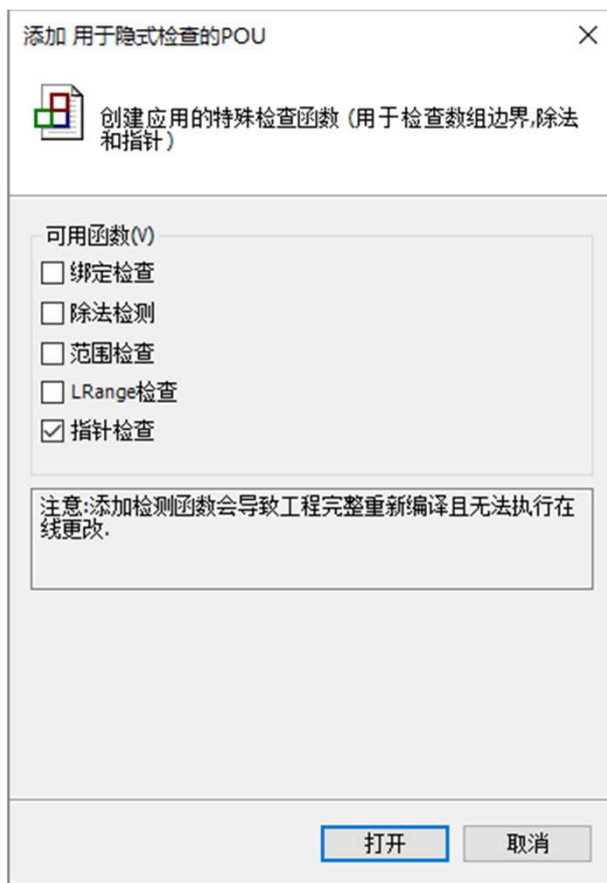
- Pr0. 基本设定
- Pr1. 增益调整
- Pr2. 振动抑制
- Pr3. 速度转矩控制
- Pr4. 监视器设定
- Pr5. 扩展设定
- Pr6. 特殊设定
- Pr7. 出厂设定
- Pr8. 控制参数
- Pr9. 路径参数

全部选择
本页全选
读取选中参数
写入选中参数
导出所选功能码
导入功能码
恢复出厂设置

功能码编号	名称	当前值	写入值	出厂值	范围	读写权限	有效类型
<input checked="" type="checkbox"/> Pr0.00	模型跟随带宽			0	0~2000	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.01	控制模式			9	0~9	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.02	实时自动调整模式			2	0~2	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.03	实时自动调整刚性			11	0~31	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.04	惯量比			250	0~10000	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.06	指令极性设置			0	0~1	R/W	断电有效
<input checked="" type="checkbox"/> Pr0.07	探针信号极性设置			3	0~3	R/W	断电有效
<input checked="" type="checkbox"/> Pr0.08	每转指令脉冲数			0	0~2147483647	R/W	断电有效
<input checked="" type="checkbox"/> Pr0.13	第1转矩限制			300	0~500	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.14	位置偏差过大设置			200	0~500	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.15	绝对式编码器设定			0	0~32767	R/W	断电有效
<input checked="" type="checkbox"/> Pr0.16	再生放电电阻阻值			100	40~500	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.17	再生放电电阻功率			50	20~5000	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.19	保留参数			0	0~1000	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.23	从站别名			2	0~32767	R/W	断电有效
<input checked="" type="checkbox"/> Pr0.24	从站别名来源			1	0~1	R/W	断电有效
<input checked="" type="checkbox"/> Pr0.25	同步补偿时间1			10	1~100	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.26	同步补偿时间2			50	1~2000	R/W	立即有效
<input checked="" type="checkbox"/> Pr0.27	保留参数			0	0~50	R/W	立即有效



“右键Application—添加对象—用于隐式检查的POU”



绑定检查--排查数组越界

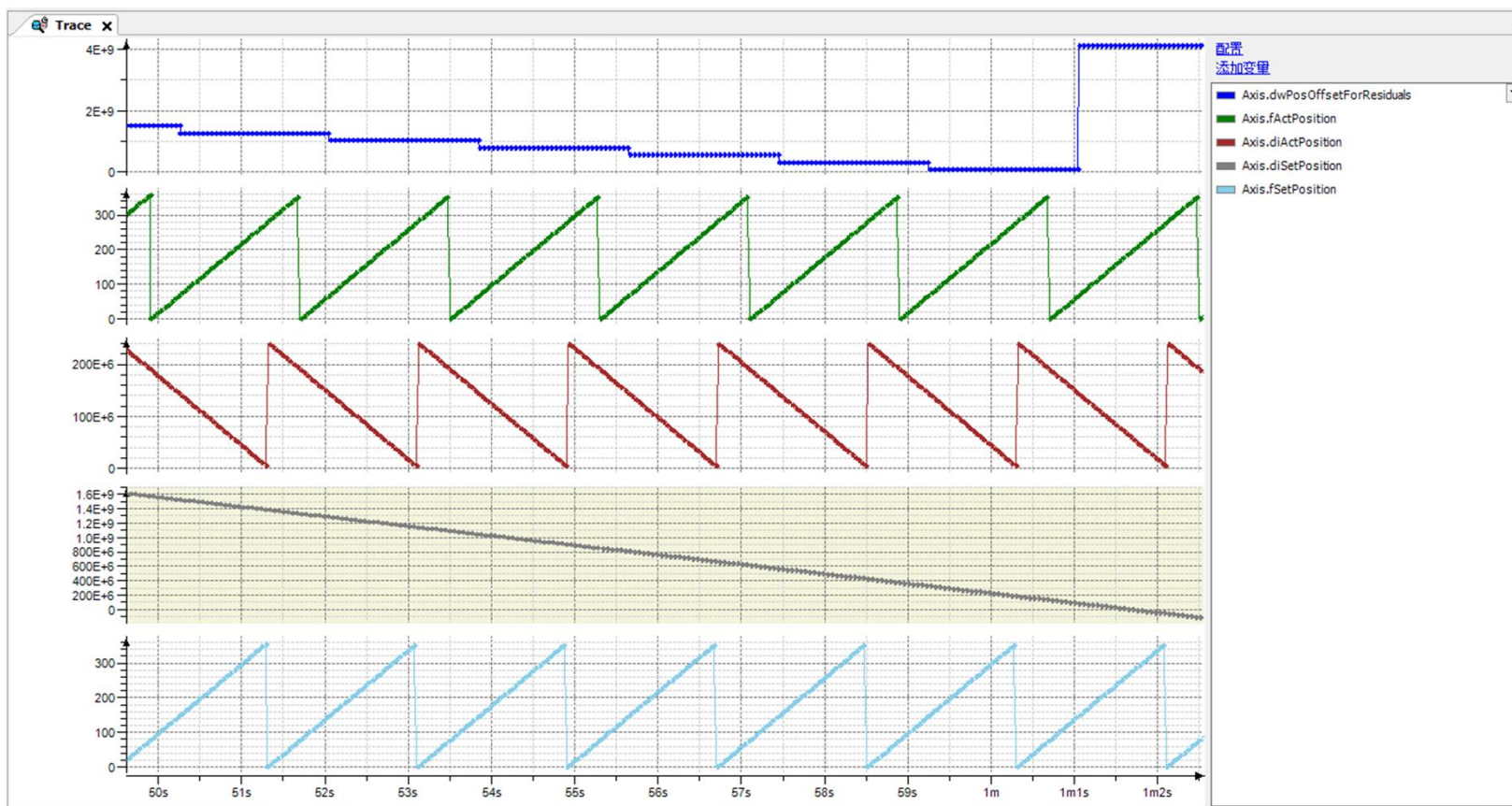
除法检测--排查除数为0

范围检查/Lrange检查--排查数据超过数据类型

允许范围

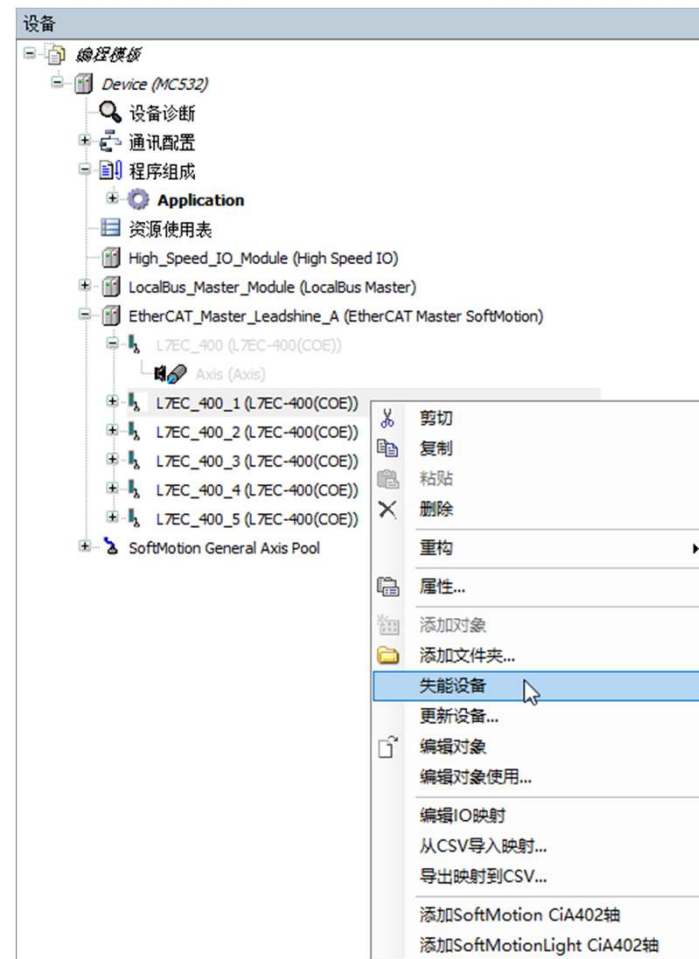
指针检查--排查空指针

“右键Application—添加对象—跟踪”



“右键ETC从站—使能设备/失能设备”

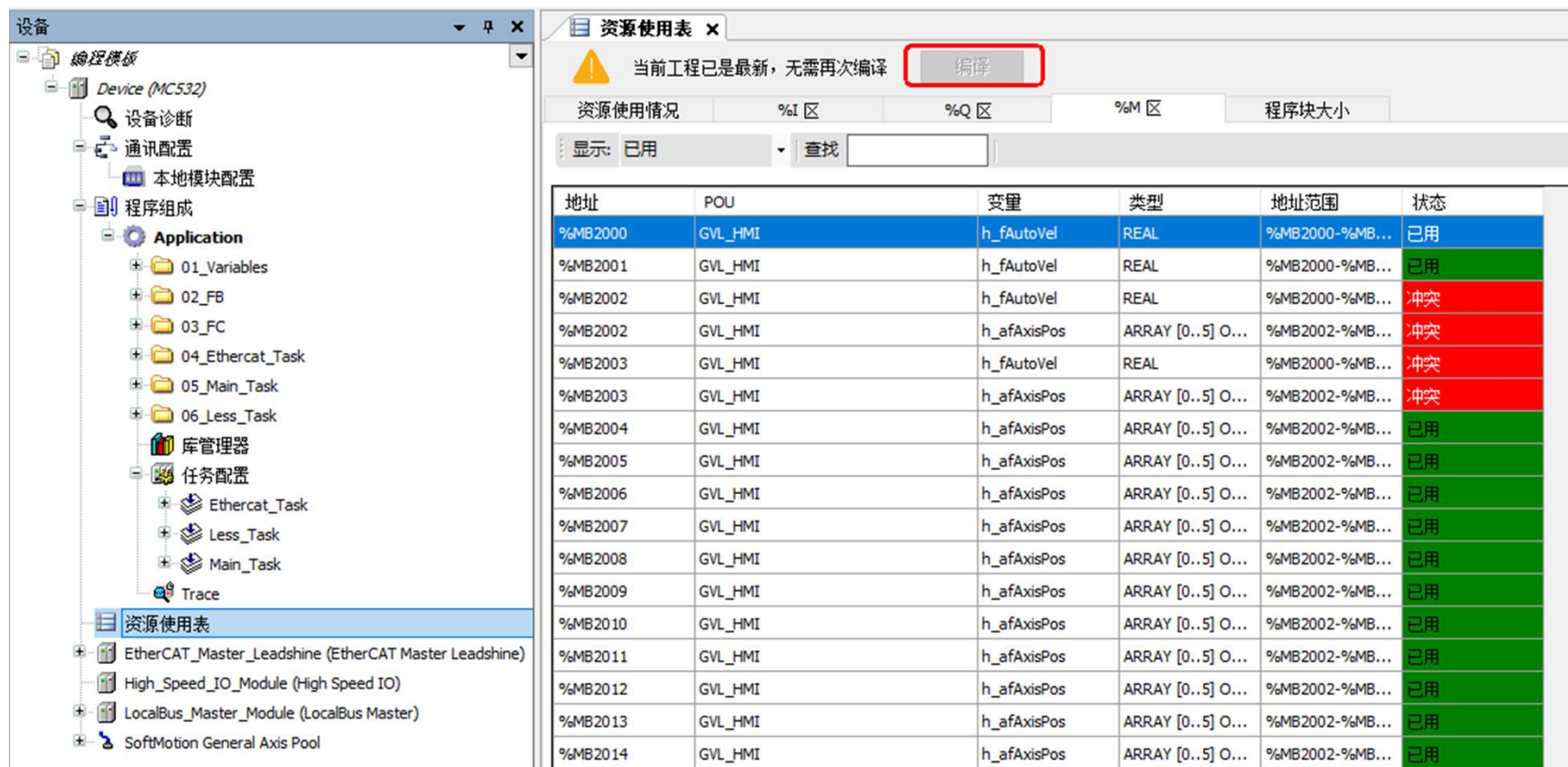
- ◆在实际应用中，用户可能会遇到组态与实际硬件连接不同的情况，导致总线无法正常运转
- ◆针对这种情况，LeadSys提供了一种解决方案，用户可以失能没有连接的从站，以保证总线能正常运转。
- ◆程序中使用到设备相关的变量时，采用失能设备的方式禁用设备，也不会导致编译报错
- ◆被失能的设备在设备树中以浅灰色字体显示
- ◆需要重新使能设备时，点击“使能设备”即可



资源使用表

双击打开资源使用表，点击编译后，可以查看直接地址是否冲突。

注：地址冲突可能引起数据异常



地址	POU	变量	类型	地址范围	状态
%MB2000	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	已用
%MB2001	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	已用
%MB2002	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	冲突
%MB2002	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	冲突
%MB2003	GVL_HMI	h_fAutoVel	REAL	%MB2000-%MB...	冲突
%MB2003	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	冲突
%MB2004	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2005	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2006	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2007	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2008	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2009	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2010	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2011	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2012	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2013	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用
%MB2014	GVL_HMI	h_afAxisPos	ARRAY [0..5] O...	%MB2002-%MB...	已用

12

其他资料

轴控、EIP、OPC UA、文件操作功能等

名称	修改日期	类型	大小
EtherNet IP	2022/9/22 17:01	文件夹	
HMI与PLC通信	2022/9/22 17:01	文件夹	
ModBus TCP	2022/9/22 16:42	文件夹	
OPC UA	2022/9/22 17:01	文件夹	
RS232_RS285自由协议	2022/9/22 17:01	文件夹	
TCPIP自由协议	2022/9/22 16:42	文件夹	
UDP协议	2022/9/22 17:01	文件夹	
点位运动控制功能	2022/9/22 17:01	文件夹	
电子齿轮	2022/9/22 16:42	文件夹	
电子凸轮	2022/11/11 14:59	文件夹	
高速输出、高速输入	2022/11/23 19:38	文件夹	
恒速控制及JOG功能	2022/9/22 16:42	文件夹	
文件操作功能	2022/9/22 16:42	文件夹	
周期同步速度模式(CSV)控制	2022/9/22 17:01	文件夹	

雷赛产品资料获取途径



成就客户 | 共创共赢

编程手册

指令手册

编程软件

2D/3D图纸

选型样本

升级包

……更多产品内容可以上雷赛官网下载
或者联系技术支持



雷赛智能官网

官方对外展示平台



雷赛智能公众号

获取更多应用案例和公司资讯



雷赛智能在线型录

获取更多产品资料

