

EC系列中型PLC 控制器 使用手册



前 言

感谢您购买并使用软控(深圳)电气有限公司EC系列中型PLC控制器!

EC系列控制器是软控(深圳)电气有限公司开发的中型可编程逻辑控制器, 使用CODESYS 3.5 SP19 编程环境, 支持逻辑控制和运动控制功能。可以通过FB/FC 功能实现工艺的封装和复用。通过RS485、以太网、EtherCAT和CANOpen接口可以实现多层次网络通信。PLC 本体集成数字量输入和数字量输出功能, 支持扩展8个锐特 IO模块。

您可以根据自己的需求选择是否支持CODESYS Softmotion功能、高速脉冲输入输出的型号。EC 系列控制器目前有 EC300一个型号, 请参照本手册第2.2 节具体 型号规格简述进行选型。

本手册为指导软控 EC系列控制器使用的综合资料, 介绍产品的安装和接线, 包括产品信息、机械安装、电气安装等。在使用本产品之前, 请仔细阅读手册, 并在充分理解手册内容的前提下进行接线和编程调试。只有具备一定的电气知识的操作人员才可以对本产品进行接线和编程调试等操作。使用该产品时, 请首先自行确认是否符合要求以及安全。在使用中如有不明白的地方, 请咨询本公司的技术人员以获得帮助。

由于致力于PLC控制器的不断改善, 因此本公司提供的资料如有变更, 恕不另行通知。

手册版本变更记录

日期	变更后版本	变更内容
2025年 5月	V1.01	首次发行，支持EC300型号说明

目 录

1. 1. 安全注意事项	5
1. 2. 确认产品到货时的注意事项	5
2. 产品信息	6
2. 1. 产品型号说明	6
2. 2. 具体型号规格简述	6
2. 3. 产品结构图	7
2. 3. 1. 正视图	7
2. 3. 2. 侧视图	7
3. 规格参数	8
3. 1. 一般规格	8
3. 2. 端口说明	9
3. 3. 性能规格	10
3. 4. 电气参数	11
3. 5. 安装规格	12
3. 5. 1. 安装尺寸	12
3. 5. 2. 安装方法	12
3. 6. 接线规格	13
3. 6. 1. 电源端子信号定义	13
3. 6. 2. I/O 输入端子信号定义	13
3. 6. 3. 高速输入口复用功能表	13
3. 6. 4. I/O 输出端子信号定义	14
3. 6. 5. 高速输出口复用功能表	14
3. 6. 6. 漏型数字输入端子接线	15
3. 6. 7. 源型数字输入端子接线	15
3. 6. 8. 数字输出端子接线	16
3. 6. 9. 数字量输入内部示意图	16

3. 6. 10. 数字量输出内部示意图	
3. 6. 11. RS485 及 CAN 端子信号定义	18
3. 7. 通信规格	18
3. 7. 1. EtherCAT 通信规格	18
4. 产品的运行、调试、维护	19
4. 1. 运行与调试	19
4. 1. 1. 产品的检查	19
4. 1. 2. 程序的编写和下载	19
4. 1. 3. 程序的调试	19
4. 1. 4. PLC 的指示灯	20
4. 2. 日常维护	20
4. 2. 1. 产品的定期检查	20
4. 2. 2. 关于电池	20
5. Codesys 软件安装	21
5. 1. 下载并安装 codesys 安装包	21
5. 2. 安装 PLC 的设备描述文件	22
6. 创建、编译并运行一个 CODESYS 项目	23
6. 1. 快速创建第一个项目	23
6. 1. 1. 新建项目	23
6. 1. 2. 创建程序并定义任务	24
6. 1. 3. 登录控制器并运行项目	27
7. IO 配置	31
7. 1. PLC 本机 IO 配置	31
7. 2. PLC 扩展 IO 模块配置	31
8. RT Modbus 库安装使用说明	32
8. 1. 库安装	32
8. 2. 工程内使用	33

8.3. Modbus 主站接口描述	
8.3.1. RT_Modbus_Client	34
8.3.2. RT_Modbus_ClientMaskWriteRegister	34
8.3.3. RT_Modbus_ClientReadBits	35
8.3.4. RT_Modbus_ClientReadInputBits	35
8.3.5. RT_Modbus_ClientReadInputRegisters	36
8.3.6. RT_Modbus_ClientReadWriteRegisters	36
8.3.7. RT_Modbus_ClientSerial	37
8.3.8. RT_Modbus_ClientTCP	38
8.3.9. RT_Modbus_ClientWriteBit	38
8.3.10. RT_Modbus_ClientWriteBits	39
8.3.11. RT_Modbus_ClientWriteRegister	39
8.3.12. RT_Modbus_ClientWriteRegisters	40
8.4. Modbus 从站接口描述	40
8.4.1. RT_ServerDataTable	40
8.4.2. RT_ModbusServer	41
8.4.3. RT_Modbus_ServerSerial	41
8.4.4. RT_Modbus_ServerTCP	42
9. Rt_PulseControlFB_脉冲控制库使用 说明	43
9.1. 库安装	43
9.2. 工程内使用	44
9.3. 功能块接口描述	45
9.3.1.2. RT_PulseControlFree	46
9.3.2.2. RT_JogP	47
9.3.2.3. RT_MoveAbsoluteP	48
9.3.2.4. RT_MoveRelativeP	48
9.3.2.5. RT_MoveStopP	49

9.3.2.6. RT_MoveVelocityP	
9.3.2.7. RT_PowerP	50
9.3.2.8. RT_StopCrdP	51
9.3.2.9. RT_2AxisLineP	52
9.3.2.10. RT_3AxisLineP	53
9.3.2.11. RT_2AxisCircleC	54
9.3.2.12. RT_2AxisCircleR	55
9.3.2.13. RT_2AxisCircle3P	56
9.3.3. 脉冲轴参数设置功能块	57
9.3.3.2. RT_SetAxisIoMapP	57
9.3.3.3. RT_GetEncoderModeP	58
9.3.3.4. RT_SetEncoderModeP	58
9.3.3.5. RT_GetIoConfigP	59
9.3.3.6. RT_SetIoConfigP	60
9.3.3.7. RT_GetPulseModeP	62
9.3.3.8. RT_SetPulseModeP	62
9.3.3.9. RT_ResetAxisIoMap	63
9.3.3.10. RT_SetPositionP	63
9.3.3.11. RT_SetEncoderPositionP	64
9.3.4. 脉冲轴信息获取功能块	64
9.3.4.1. RT_ReadActualPositionP	64
9.3.4.2. RT_ReadActualVelocityP	65
9.3.4.3. RT_ReadEncoderPositionP	65
9.3.4.4. RT_ReadStatusP	66
10. 附录	67
10.1. 电工中的3种接地是什么意思?	67
10.2. 如何查看和修改EC系列PLC 的 IP 地址	68

1.安全提醒

1. 1. 安全注意事项

1. 在安装控制器前，请务必断开所有外部电源。否则有触电的危险。
2. 对控制器上电后，请勿触摸端子，请勿带电对端子进行接线、拆线等操作。否则有触电的危险。
3. 请在手册的规格规定的环境条件下，安装和使用本产品。请勿在潮湿、高温、有灰尘、烟雾、导电性粉尘、腐蚀性气体、可燃性气体、以及有振动、冲击的场所中使用。 否则有可能引起触电、火灾、误动作、产品损坏等。
4. 请在控制器的外部设计安全回路，确保控制器运行异常时，整个系统也能安全运行。否则有引起误动作、故障的危险。
5. 请将DC 24V 电源正确连接到控制器的专用电源端子上。接错电源，可能会烧毁控制器。
6. 请勿将控制接线与动力接线捆绑在一起，原则上要分开 10cm。否则有可能引起误动作、产品损坏。
7. 请勿直接触摸产品的导电部位。否则有可能引起误动作、故障。
8. 请使用 DIN46277 导轨或 M3 螺丝固定本产品，并安装在平整的表面。错误的安装可能引起误动作、产品损坏。
9. 进行螺丝孔的加工时，请切勿使切割粉末、电线碎屑掉入产品外壳内。否则有可能引起误动作、故障。
10. 连接或拆卸外围设备、扩展设备、电池等设备时，请务必断电操作。否则有可能引起误动作、故障。
11. 请使用 2mm² 的电线对控制器的接地端子进行第三种接地，不可与强电系统公共接地。 否则有可能造成故障、产品损坏等。
12. 使用电线连接端子时，请注意务必拧紧，且不可使导电部分接触到其他电线或端子。否则有可能引起误动作、产品损坏。
13. 对控制器中的程序进行更改之前，请务必先对其 STOP。否则有可能引起误动作。
14. 请勿擅自拆卸、组装本产品。否则有可能造成产品的损坏。
15. 请在断电的情况下，插拔连接电缆。否则有可能造成电缆的损坏、引起误动作。
16. 请绝对不要对本产品进行改造，否则可能会导致受伤或机械损伤。
17. 产品废弃时，请按工业废弃物处理或者按当地环境保护规定处理。

1. 2. 确认产品到货时的注意事项

确认项目	说明
到货产品是否与您订购的产品型号相符?	包装箱内含有您订购的产品，请通过控制器的标签型号进行确认。
产品是否有损坏的地方?	请查看包装外表，产品在运输过程中是否有破损现象。若发现遗漏或损坏，请与本公司或您的供货商联系。

2. 产品信息

2.1. 产品型号说明

EC 310-1616 T
① ② ③ ④

①	产品系列 EC系列中型可编程逻辑控制器
②	机型代号 3: 中型 300 系列支持 Softmotion 功能; 1: EtherCAT 总线型 8: 脉冲轴数, 0 代表不支持脉冲轴
③	输入输出点数 1616 表示: 本体 IO 16 点输入 16 点输出
④	输出类型 R: 继电器输出, T: 晶体管 NPN 输出, P: 晶体管 PNP 输出

2.2. 具体型号规格简述

型号	描述
EC310-1616T	中型EtherCAT 总线型可编程逻辑控制器, 支持EtherCAT 总线 1ms 周期8 轴同步、支持 CODESYS Softmotion, 支持 EtherNET、CANOPEN 主站、2 路 RS485, 本体带 16 点输入 16 点晶体管NPN 输出, 可扩展 8 个软控EC系列 IO 模块
EC518-1616T	中型EtherCAT 总线型可编程逻辑控制器, 支持EtherCAT 总线 1ms 周期8 轴同步、支持 CODESYS Softmotion, 支持 EtherNET、CANOPEN 主站、2 路 RS485, 本体带 16 点输入 16 点晶体管NPN 输出、8 路高速脉冲输出、8 路高速计数, 可扩展 8 个软控EC系列 IO 模块

2.3. 产品结构图

2.3.1. 正视图



2.3.2. 侧视图



3. 规格参数

3.1. 一般规格

项目	标准与规范
工作环境	
温度	0℃~55℃
湿度	5%~95%（无凝露）
海拔	-1000m~+2000 米
空气	防尘、非腐蚀性，较低盐雾、潮湿、尘雾等环境， SO ₂ <0.5ppm，相对湿度<60%，无结露 H ₂ S<0.1ppm，相对湿度<60%，无结露
绝缘电压 /耐高压绝缘性	DC 500V 2MΩ 以上/ 端对地 2200VDC，I/O 接口端对其它端 1500VAC，持续 1 分钟
存储环境	
温度	-25~70℃
电磁兼容性 - 抗扰度	
抗电气干扰	脉宽 50ns，重复频率 5kHz，2,000V 电压峰值 / 噪声电压 1000Vp-p 1us 脉冲 1 分钟
静电放电 IEC61000-4-2	接触放电：±4KV 空气放电：±8KV
电快速瞬变脉冲群 IEC61000-4-4	电源线：2KV，5KHz 信号线：2KV，5KHz（I/O 耦合夹） 1KV，5KHz（通信耦合夹）
抗振性	频率：10~57Hz，幅度：0.1mm，频率：57~150Hz，加速度：1.0g，三 维 方向各 10 次
抗冲击性	15g，持续 11ms，3 维方向冲击 3 次
接地（FG）	第三种接地（不可与强电系统公共接地）

3.2. 端口说明

序号	端口类型	接口标识	定义	指示灯颜色	说明
1	运行状态指示灯	PWR	电源正常	黄绿色	● 常亮：表示供电正常 ● 熄灭：表示供电异常
		RUN	运行正常	黄绿色	● 常亮：表示用户程序正在运行 ● 熄灭：表示用户程序停止
		ERR	运行错误	红色	● 熄灭：表示无严重错误 ● 闪烁：表示有严重错误发生
		ENET	EtherNet 通信状态指示	黄绿色	● 常亮：表示链接建立成功 ● 熄灭：表示链接未建立
		ECAT	EtherCAT 通信状态指示	黄绿色	● 常亮：表示链接建立成功 ● 熄灭：表示链接未建立
		CAN	CAN 通信状态指示	黄绿色	● 常亮：表示链接建立成功 ● 熄灭：表示链接未建立
2	IO 指示灯	IN/OUT	IO 状态显示	黄绿色	● 常亮：表示输入或输出有效 ● 熄灭：表示输入或输出无效
3	拨码开关	RUN/STOP	控制主机运行 / 停止	—	—
4	Type-C 接口	—	用于 USB 扩展	—	使用 type-C 接口的 U 盘
5	SD 卡插槽	SD	用于 SD 存储扩展，用于存储用户程序与用户数据	—	使用 micro SD 卡
6	RS485 PORT1	485A+	485 通信信号+	—	—
		485A-	485 通信信号-	—	—
		GND	485 通信地	—	RS485 PORT1 和 PORT2 共用
	RS485 PORT2	485B+	485 通信信号+	—	—
		485B-	485 通信信号-	—	—
	CAN	CAN L	CAN Low	—	—
		CAN H	CAN High	—	—
		CGND	CAN 通信地	—	—
7	以太网口	EtherNet	以太网通信 RJ45 接口	—	—
8	EtherCAT 接口	EtherCAT	用于 EtherCAT 通信	—	—
9	电源接口	+24V	直流 24V 电源正	—	—
		0V	直流 24V 电源负	—	—

		PE	PE	—	—
10	I/O 输入端子	—	16 路输入	—	3.6.2-3.6.3 节详细定义
11	I/O 输出端子	—	16 路输出	—	3.6.4-3.6.5 节详细定义

3.3. 性能规格

项目		规格
基本项目	程序容量	20M Bytes
	数据容量	20MByte, 其中EC300 支持4K Bytes 掉电保持, RM518/418 支持16K Bytes 掉电保持
	X 区 (%I)	128Byte
	Y 区 (%Q)	128Byte
	M 区 (%M)	128KByte
	带轴性能	1ms 周期8 轴同步 (运动控制计算的执行时间)
	电子凸轮、插补	支持
	本地扩展 I/O 模块	最多可支持8 个本地扩展模块
	实时时钟	纽扣电池保持 (可以自行更换)
编程	编程平台	CODESYS V3.5 SP19
	编程语言	IEC 61131-3 编程语言 (LD、ST、SFC、CFC)
通信	EtherCAT	● 传输速度 100Mbps (100Base-TX) ● 支持协议, EtherCAT 主站 ● 支持最多 128 个EtherCAT 从站, 最小同步周期: 500 μs ● 从站支持禁用和扫描
	EtherNet	● 传输速度 100Mbps (100Base-TX) ● 支持 Modbus-TCP 主站/从站: 做主站, 支持最多63个从站; 做从站, 支持最多 16 个主站 ● TCP/UDP 自由协议, 支持最多 16 个连接 ● Socket, 最大连接数量: 4, 支持 TCP/UDP ● IP 地址初始值: 192.168.1.3
	CAN	● 通信波特率: 125000bit/s、250000bit/s、500000bit/s、800000bit/s、1000000bit/s ● 支持 CANOPEN 协议 ● 终端电阻, 内置 120 Ω ● 最大传送距离 100m (125000bit/s 时)
	RS485	● 支持路数: 2 路 ● 隔离方式: 不隔离 ● 可以做 Modbus 主站或者从站 (ASCII/RTU) ● 接从站数量: 最多支持 31 个Modbus-RTU 从站 ● 通信波特率: 9600bit/s、19200bit/s、38400bit/s、

		57600bit/s、115200bit/s ● 支持串口自由协议 ● 终端电阻，外接 120 Ω ● 最大传送距离 500m (9600bit/s 时)
	USB	● USB 线距离：1.5m ● USB 通信版本：USB2.0，全速 ● USB 接口：Type-C ● 主/从：只能做主、不能做从
用户程序升级	EtherNet	支持EtherNet 监控PLC，上下载用户程序
	TF 卡	不支持通过存储扩展卡下载用户程序
	Type-C	不支持 Type-C 监控PLC、上下载用户程序

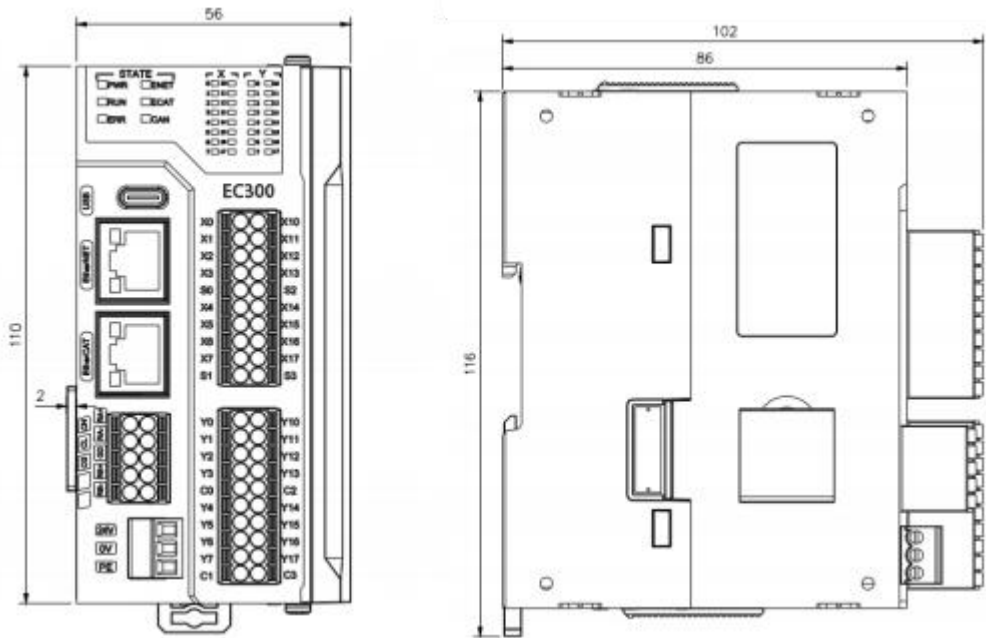
3.4. 电气参数

项目	电气参数
电源输入电压	DC 24V
允许电源电压波动范围	DC20.4V~28.8V (-15%~+20%)
24V 输入电源保护	支持短路保护和反接保护
主机开关量输入	
数字量输入点数	16 点双极性输入
隔离方式	光电耦合
输入阻抗	2.4K Ω
输入为ON	高速输入端输入电流大于 5.8mA/24V，普通输入端输入电流大于 9.9mA/24V
输入为OFF	高速输入端输入电流小于 4.5mA/19V，普通输入端输入电流小于 4mA/17V
滤波功能	有滤波功能： ● 滤波参数：1ms~1000ms
高速脉冲计数功能	无
输入公共端方式	4 点/公共端（输入电源的极性+/-均可变更）
输入电平	漏型/源型，S/S 接 24V 为NPN，S/S 接 GND 为PNP。
隔离	现场与逻辑分组隔离
主机开关量输出	
数字量输出点数	16 点NPN输出
允许最大电流	0.5A/点
回路电源电压	DC 24V
电路绝缘	光电绝缘
ON 响应时间	0.5ms
输出公共端方式	4 点/公共端（输出电源的极性-）
输出电平	低电平NPN，COM接负
短路保护功能	每路支持短路保护，断电后恢复

3.5. 安装规格

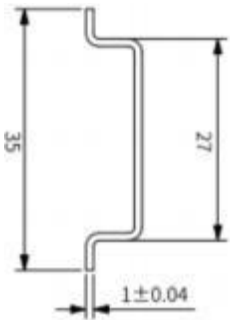
3.5.1. 安装尺寸

安装尺寸信息如下图所示，单位为毫米（mm）。



3.5.2. 安装方法

采用DIN导轨安装，DIN导轨需符合 IEC 60715标准（35mm宽，1mm厚），尺寸信息如下图所示，单位为毫米（mm）。



3.6. 接线规格

3.6.1. 电源端子信号定义

端子序号	电源接线
1	直流 24V 电源正
2	直流 24V 电源负
3	PE

3.6.2. IO 输入端子信号定义

左侧信号	左侧端子	右侧端子	右侧信号
X0 信号输入	1A	1B	X10 信号输入
X1 信号输入	2A	2B	X11 信号输入
X2 信号输入	3A	3B	X12 信号输入
X3 信号输入	4A	4B	X13 信号输入
输入公共端 SS0	5A	5B	输入公共端 SS2
X4 信号输入	6A	6B	X14 信号输入
X5 信号输入	7A	7B	X15 信号输入
X6 信号输入	8A	8B	X16 信号输入
X7 信号输入	9A	9B	X17 信号输入
输入公共端 SS1	10A	10B	输入公共端 SS3

3.6.3. 高速输入口复用功能表

左侧端子	复用信号 1	复用信号 2	复用信号 3	复用信号 4
1A	X0 信号输入	单相高速计数 0	AB 相计数器 0 A 相	
2A	X1 信号输入		AB 相计数器 0 B 相	
3A	X2 信号输入	单相高速计数 1	AB 相计数器 1 A 相	
4A	X3 信号输入		AB 相计数器 1 B 相	
5A	输入公共端 SS0			
6A	X4 信号输入	单相高速计数 2	AB 相计数器 2 A 相	
7A	X5 信号输入		AB 相计数器 2 B 相	
8A	X6 信号输入	单相高速计数 3	AB 相计数器 3 A 相	
9A	X7 信号输入		AB 相计数器 3 B 相	
10A	输入公共端 SS1			

右侧端子	复用信号 1	复用信号 2	复用信号 3	
1B	X10 信号输入	单相高速计数 4	AB 相计数器 4 A 相	
2B	X11 信号输入		AB 相计数器 4 B 相	
3B	X12 信号输入	单相高速计数 5	AB 相计数器 5 A 相	
4B	X13 信号输入		AB 相计数器 5 B 相	
5B	输入公共端 SS2			
6B	X14 信号输入	单相高速计数 6	AB 相计数器 6 A 相	AB 相计数器 0 Z 相
7B	X15 信号输入		AB 相计数器 6 B 相	AB 相计数器 1 Z 相
8B	X16 信号输入	单相高速计数 7	AB 相计数器 7 A 相	AB 相计数器 2 Z 相
9B	X17 信号输入		AB 相计数器 7 B 相	AB 相计数器 3 Z 相
10B	输入公共端 SS3			

3.6.4. IO 输出端子信号定义

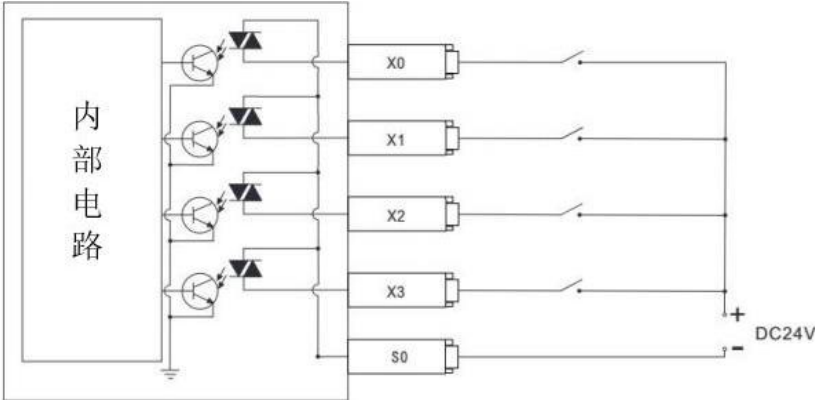
左侧信号	左侧端子	右侧端子	右侧信号
Y0 信号输出	1A	1B	Y10 信号输出
Y1 信号输出	2A	2B	Y11 信号输出
Y2 信号输出	3A	3B	Y12 信号输出
Y3 信号输出	4A	4B	Y13 信号输出
输出公共端 COM0	5A	5B	输出公共端 COM2
Y4 信号输出	6A	6B	Y14 信号输出
Y5 信号输出	7A	7B	Y15 信号输出
Y6 信号输出	8A	8B	Y16 信号输出
Y7 信号输出	9A	9B	Y17 信号输出
输出公共端 COM1	10A	10B	输出公共端 COM3

3.6.5. 高速输出口复用功能表

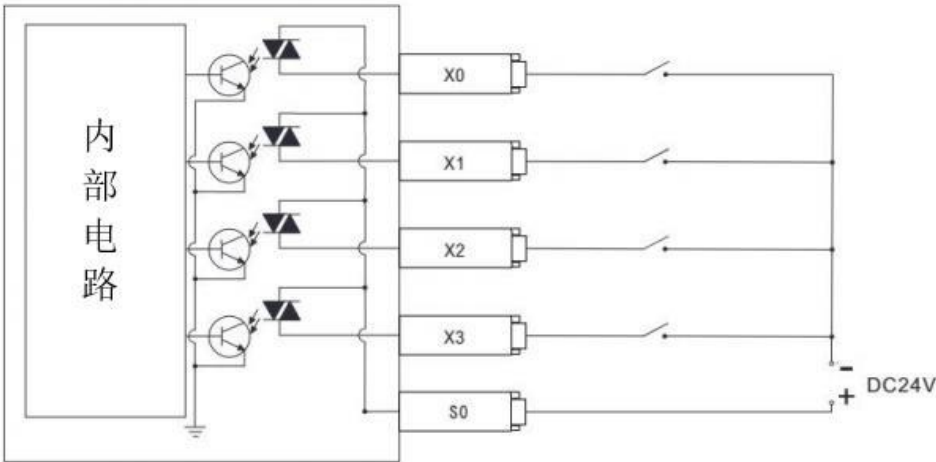
左侧端子	复用信号 1	复用信号 2	复用信号 3
1A	Y0 信号输出	轴 0 PULSE 输出	轴 0 CW 输出
2A	Y1 信号输出	轴 0 DIR 输出	轴 0 CCW 输出
3A	Y2 信号输出	轴 1 PULSE 输出	轴 1 CW 输出
4A	Y3 信号输出	轴 1 DIR 输出	轴 1 CCW 输出
5A	输出公共端 COM0		
6A	Y4 信号输出	轴 2 PULSE 输出	轴 2 CW 输出
7A	Y5 信号输出	轴 2 DIR 输出	轴 2 CCW 输出
8A	Y6 信号输出	轴 3 PULSE 输出	轴 3 CW 输出
9A	Y7 信号输出	轴 3 DIR 输出	轴 3 CCW 输出

10A	输出公共端 COM1		
右侧端子	复用信号 1	复用信号 2	复用信号 3
1B	Y10 信号输出	轴 4 PULSE 输出	轴 4 CW 输出
2B	Y11 信号输出	轴 4 DIR 输出	轴 4 CCW 输出
3B	Y12 信号输出	轴 5 PULSE 输出	轴 5 CW 输出
4B	Y13 信号输出	轴 5 DIR 输出	轴 5 CCW 输出
5B	输出公共端 COM2		
6B	Y14 信号输出	轴 6 PULSE 输出	轴 6 CW 输出
7B	Y15 信号输出	轴 6 DIR 输出	轴 6 CCW 输出
8B	Y16 信号输出	轴 7 PULSE 输出	轴 7 CW 输出
9B	Y17 信号输出	轴 7 DIR 输出	轴 7 CCW 输出
10B	输出公共端 COM3		

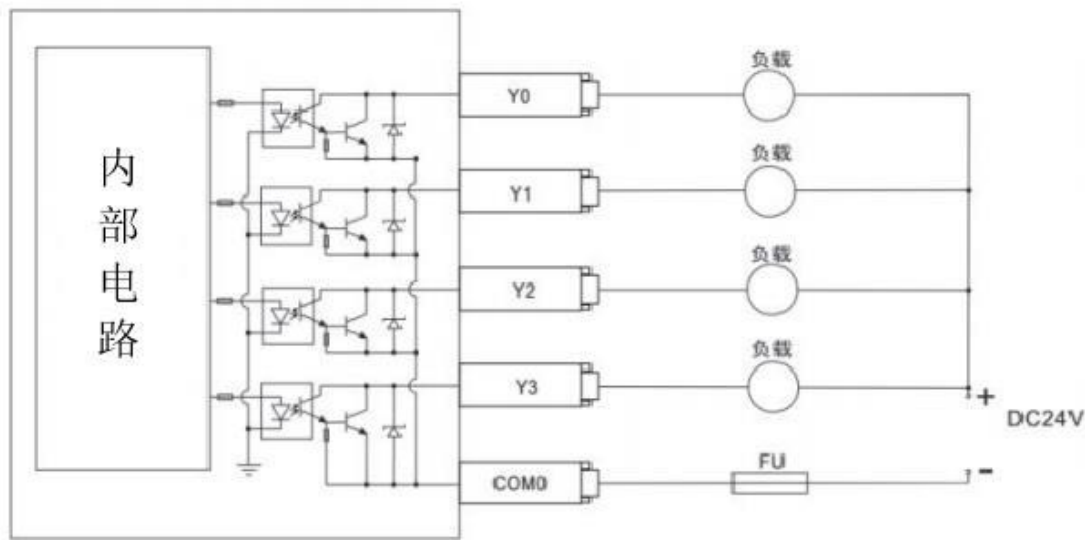
3.6.6. 漏型数字输入端子接线



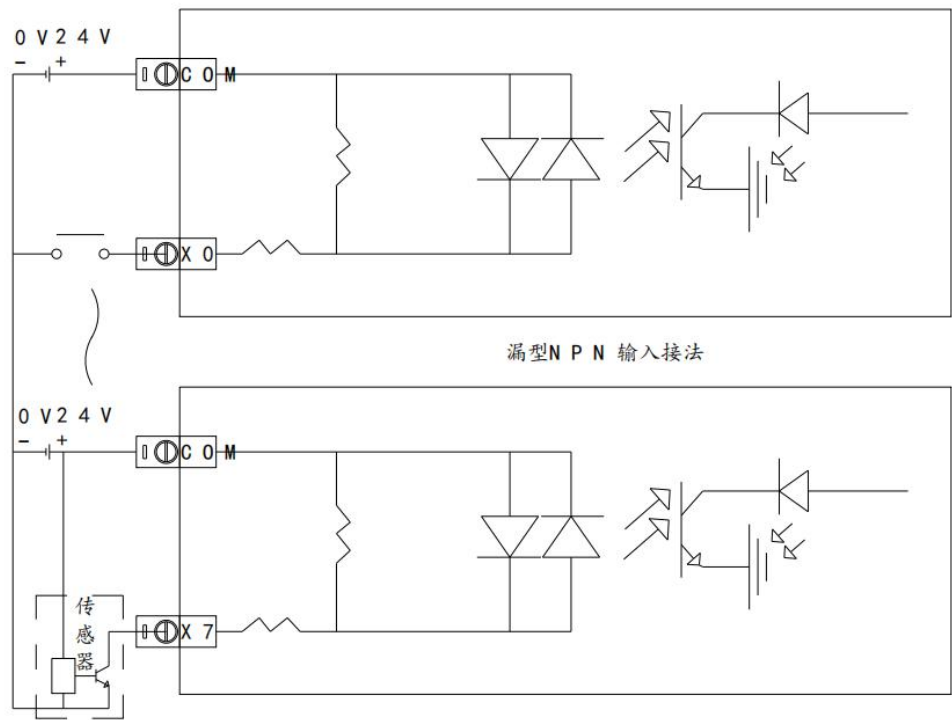
3.6.7. 源型数字输入端子接线

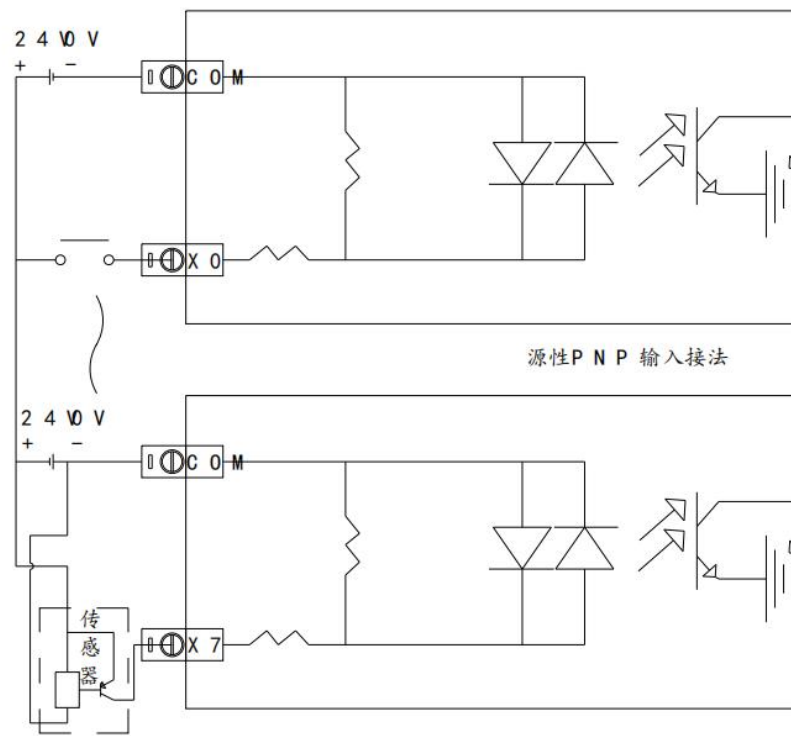


3.6.8. 数字输出端子接线

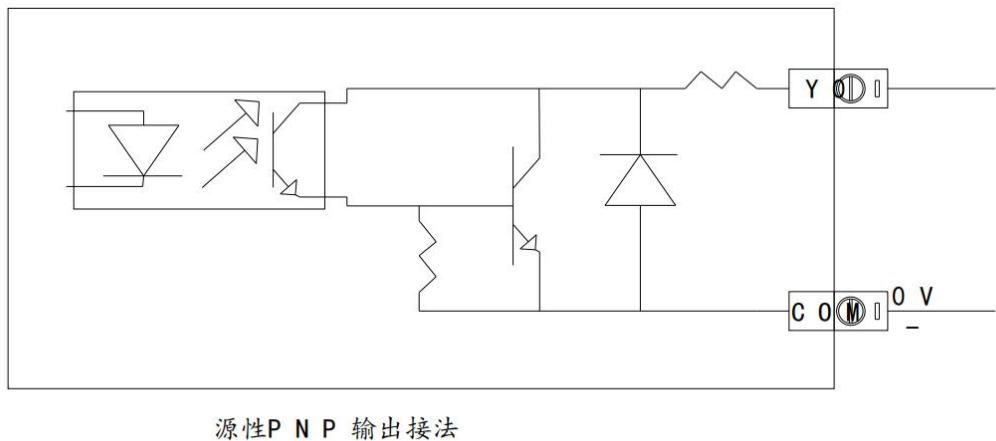


3.6.9. 数字量输入内部示意图





3.6.10. 数字量输出内部示意图



3. 6. 11. RS485 及 CAN 端子信号定义

左侧端子	左侧信号	右侧信号	右侧端子
1A	CAN-H	485A+	1B
2A	CAN-L	485A-	2B
3A	CGND	GND	3B
4A	保留	485B+	4B
5A	保留	485B-	5B

3. 7. 通信规格

3. 7. 1. EtherCAT 通信规格

项目	描述
通信协议	EtherCAT 协议
支持服务	CoE (PDO、SDO)
同步方式	DC-分布式时钟
物理层	100Mbit/s (100Base-TX)
双工方式	全双工
拓扑结构	线形拓扑结构
传输媒介	AWG26 超 5 类双绞屏蔽线
传输距离	两节点间小于 100m
从站数	最多可带 128 个
EtherCAT 帧长度	44 字节 ~1498 字节
过程数据	单个以太网帧最大 1486 字节

4. 产品的运行、调试、维护

本章介绍EC300系列 PLC 从编程到投入使用的过程，这其中涉及 PLC 的运行、调试以及日常维护等内容。

4.1. 运行与调试

4.1.1. 产品的检查

拿到产品之后，请首先检查产品的输入输出端子是否完好，是否缺少部件。一般而言，此时的 PLC 可以直接连接电源线进行上电检查，PWR 和 RUN 指示灯应常亮。

4.1.2. 程序的编写和下载

确认产品完好之后，就可以对 PLC 编写程序了，程序的编写在个人电脑中进行。编写完成的程序可以下载到 PLC 中了。一般操作步骤如下：



4.1.3. 程序的调试

理想情况下，PLC 处于正常运行状态，但如果发现 PLC 中的程序有误，需要修改时，就需要对运行中的 PLC 重新写入程序。

- (1) 使用编程电缆连接 PLC 与电脑；
- (2) 上载 PLC 中的程序；
- (3) 修改上载后的程序，修改过的程序建议另存；
- (4) 暂停 PLC 的运行，将修改后的程序下载到 PLC 中；
- (5) 对 PLC 进行监控；
- (6) 如仍旧不满足要求，可继续修改程序并下载到 PLC 中，直到满足要求。

4.1.4. PLC 的指示灯

PLC 处于正常运行时，指示灯 PWR 和 RUN 应常亮。

指示灯 ERR 常亮时，表示 PLC 程序运行出现问题，请及时更正程序。
指示灯 PWR 不亮，则电源出现问题，应检查电源接线。

4.2. 日常维护

4.2.1. 产品的定期检查

尽管可编程控制器具有一定抗干扰以及较强的稳定性，但也应该养成定期对控制器检查保养的习惯。检查的项目包括：

PLC 的输入输出端子、电源端子是否松动不牢固

； 通讯端口是否完好无损；

电源指示灯、输入输出指示灯是否可以点亮；

扫除 PLC 外部积压的灰尘，避免灰尘、导电尘埃落到 PLC 内部

； 尽量使 PLC 的运行和存储环境符合本手册 3-1 节中所述的标准。

4.2.2. 关于电池

可编程控制器内部并无严重缩短其寿命的元器件，可一直使用下去。但如果使用了 PLC 的实时时钟（RTC）功能，则需定期为其更换电池。

电池的使用寿命一般为 3~5 年。

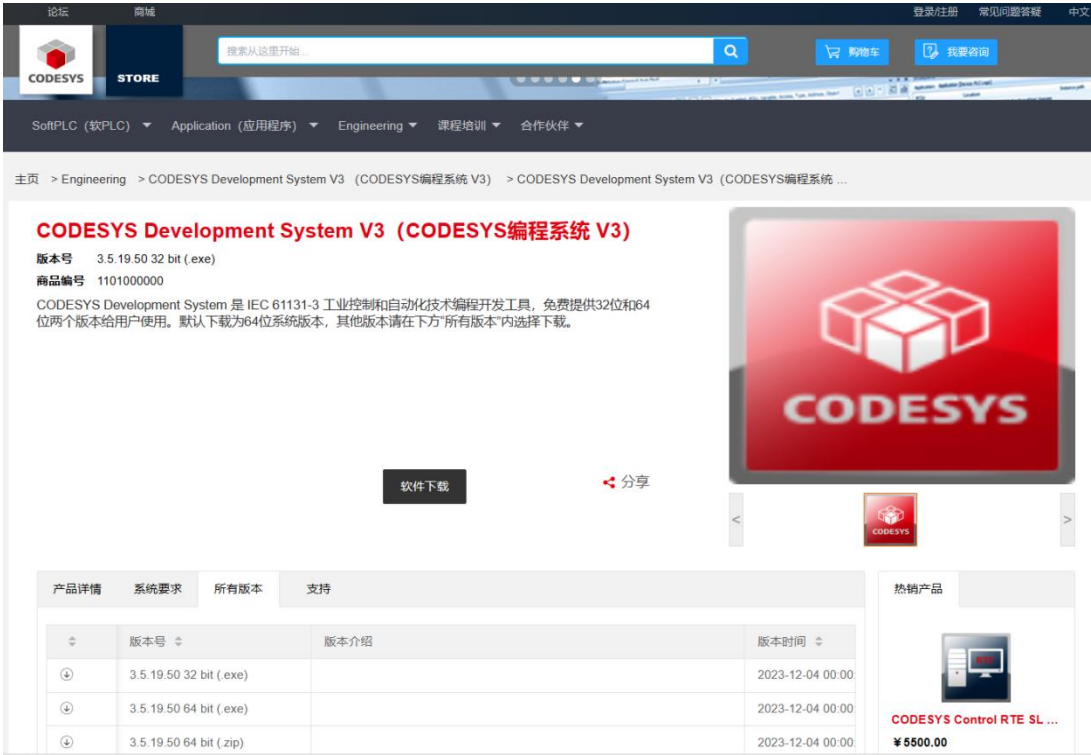
发现电池电量下降后，请尽早更换电池。

更换电池后的 PLC，请尽早上电，否则可能导致电池耗尽。

5. Codesys 软件安装

5.1. 下载并安装 codesys 安装包

1、找到 codesys 官网（下载专区-CODESYS-控制器开发平台软件）如下图，下载并安装 codesys 安装包。



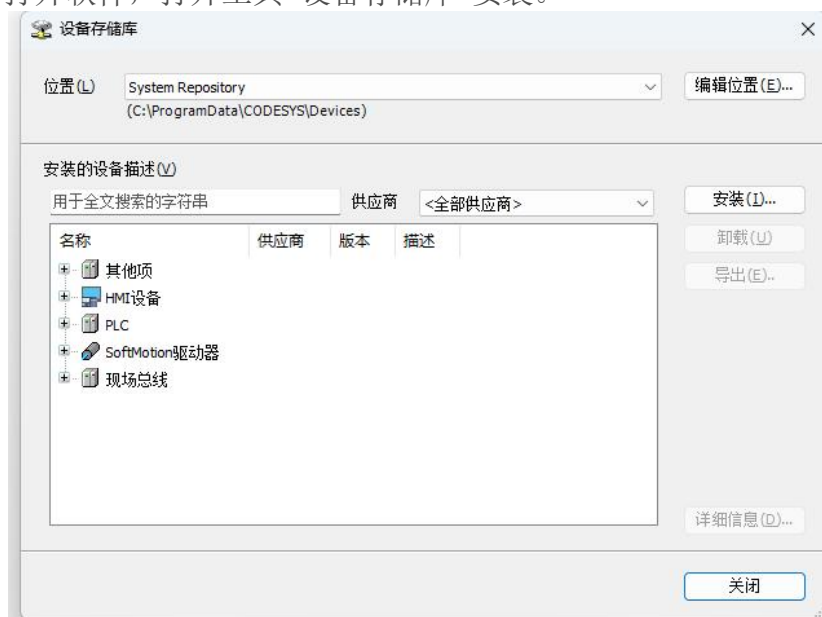
2、下载完成之后进行安装，默认配置直接安装。请注意需以管理员身份运行文件CODESYS.3.5.19.20，以启动 CODESYS V3 的安装。

3、安装完成之后状态栏将有以下三个图标，分别是权限管理（可以不管，但是要有）、网关以及软PLC。

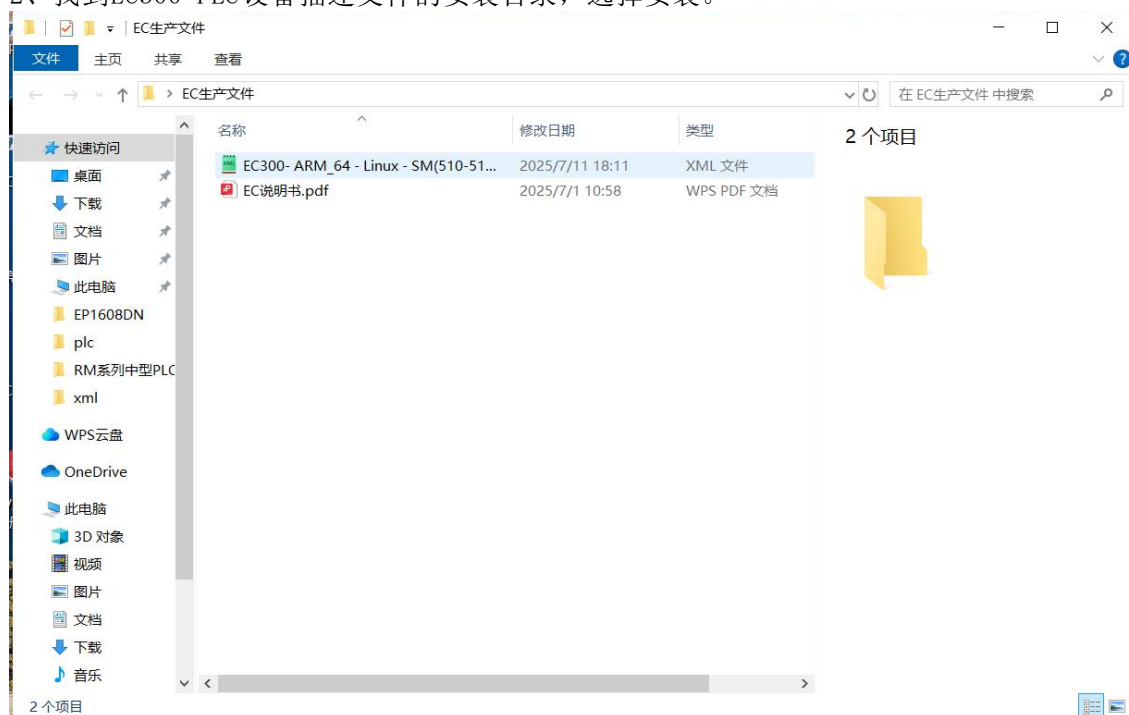


5.2. 安装PLC 的设备描述文件

1、打开软件，打开工具-设备存储库-安装。



2、找到EC300 PLC设备描述文件的安装目录，选择安装。



6. 创建、编译并运行一个 CODESYS 项目

上一章，我们已经完成CODESYS编程环境的安装；本章将详细介绍如何创建、编译并运行第一个CODESYS V3项目。在此之前，需再次确认以下要求已被满足：

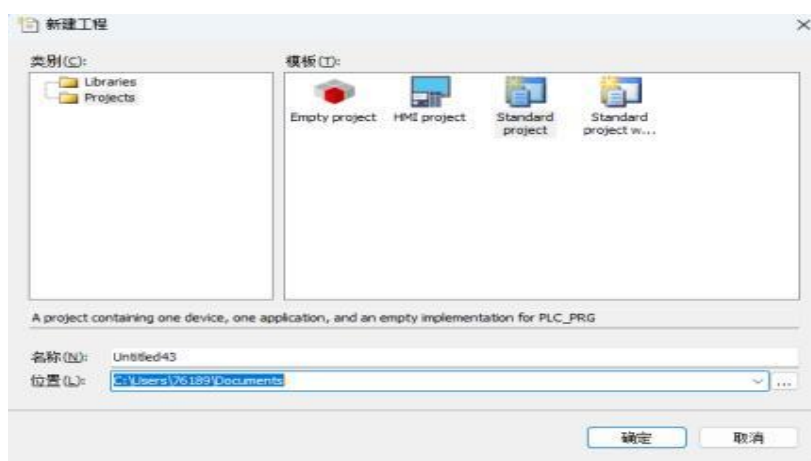
- 控制器的网络设置必须正确配置，以便通过以太网访问控制器。
- 必须在开发人员的计算机上安装与控制器固件版本相对应的CODESYS V3版本和 设备库和PLC的设备描述文件。

6.1. 快速创建第一个项目

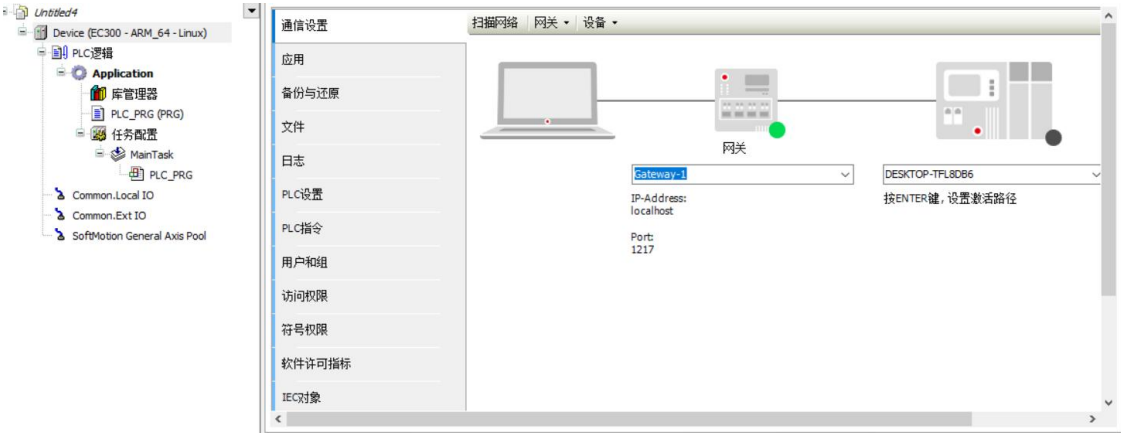
确认无误后，我们可以逐步创建一个CODESYS V3 的示例项目，并加载到控制器上，执行的步骤如下：

6.1.1. 新建项目

点击“新建项目”，选择“标准项目”，自定义存储路径和项目名称，注意选择文件设备类型。



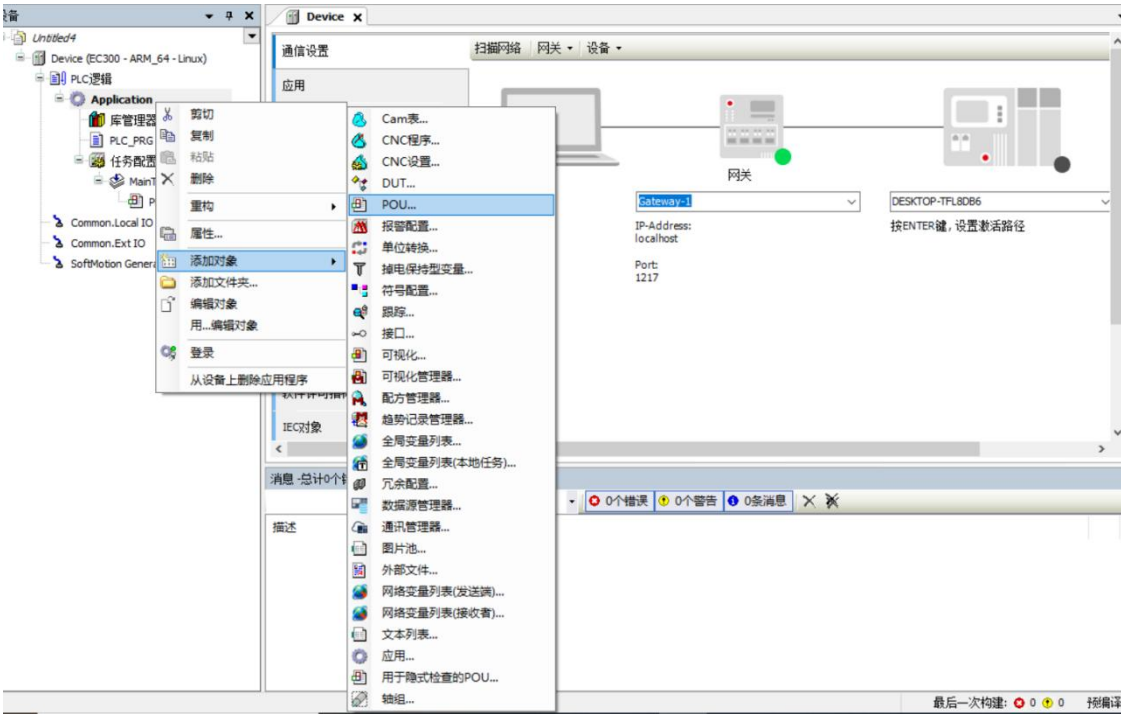
创建成功后，工程界面如下图所示：



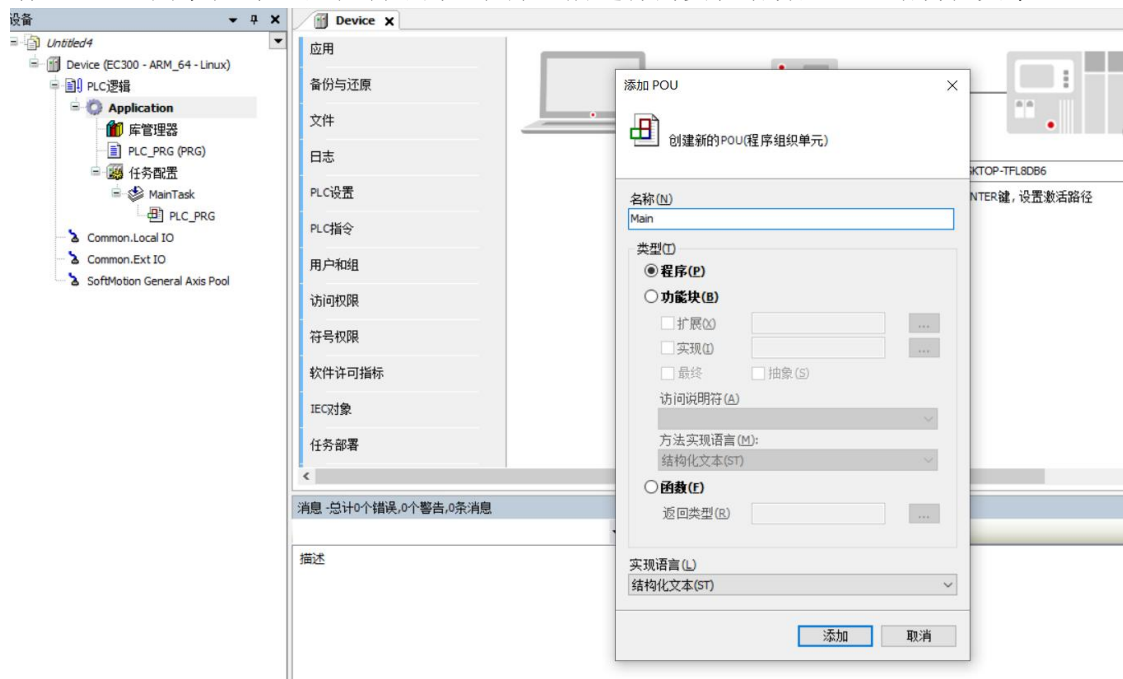
6.1.2. 创建程序并定义任务

1、创建程序：

选择您的应用程序对象，然后单击右键以打开上下文菜单，选择“添加对象”，然后在子菜单中选择“POU...”。

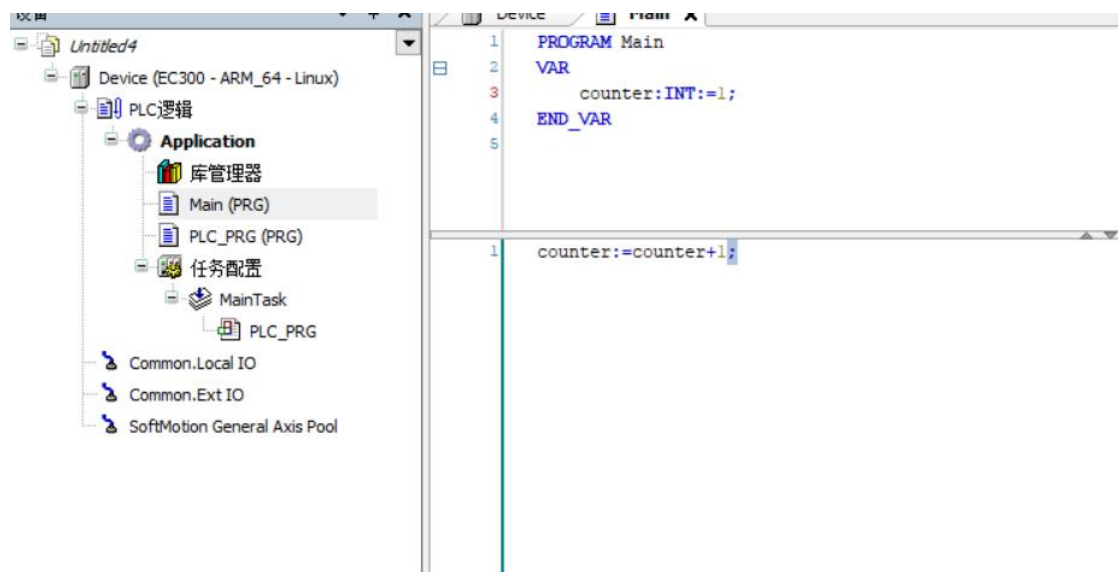


选择POU的类型为“程序”，并给该程序起一个名称。在本例中，名称“Main”用来表示这是控制器的主程序。所选择的实现语言是ST（结构化文本）。



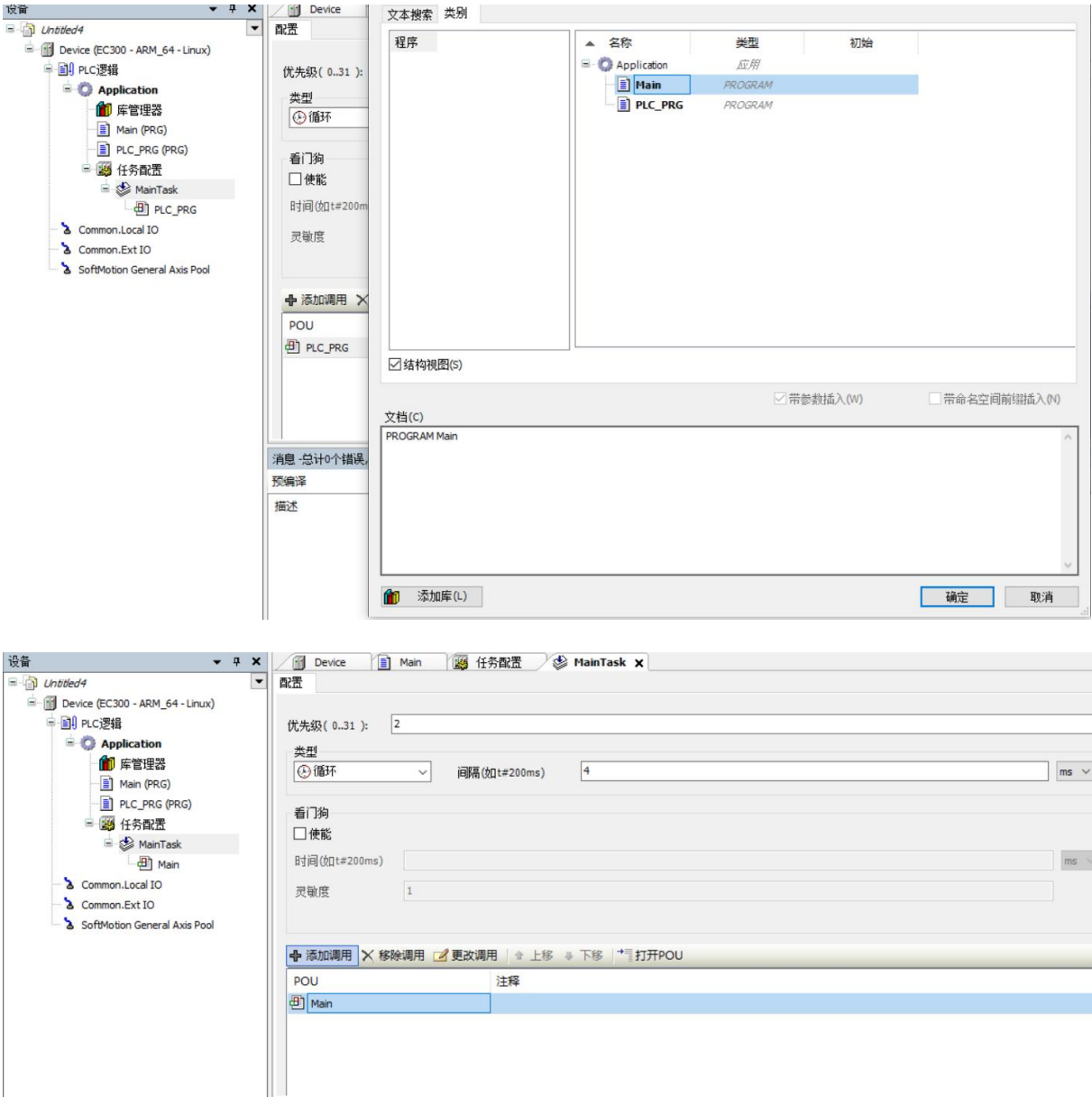
2、定义任务：

通过双击编辑模块，在编辑器窗口的上半部分中定义我们的第一个变量，名称为“counter”，数据类型为INT，并用值“0”初始化这个变量。就此我们实现了一个简单的程序：每当您调用“main”对象时，该程序都会对变量“counter”实现+1。



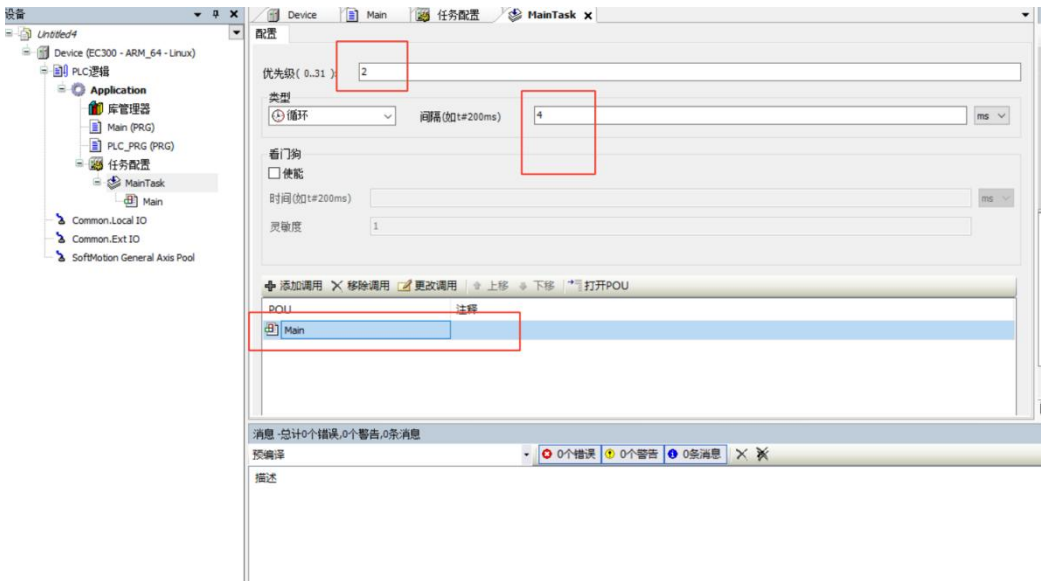
3、选择任务调用对象：

“任务配置”对象会自动创建一个类型为“MainTask”的子对象。您可以通过双击“MainTask”对象来配置它。选择“添加调用”，然后选择您已经创建的“程序”对象“Main”，点击确定。



4、任务时间间隔：

默认情况下，任务时间间隔设置为4 ms，这意味着控制器将每4 ms调用并运行你的“程序”对象。对于几个已定义的任务，当一个程序正在执行时，优先级高的任务优先于优先级低的任务，能中断同一资源中优先级低的程序执行，使较低优先级的程序执行被放缓。

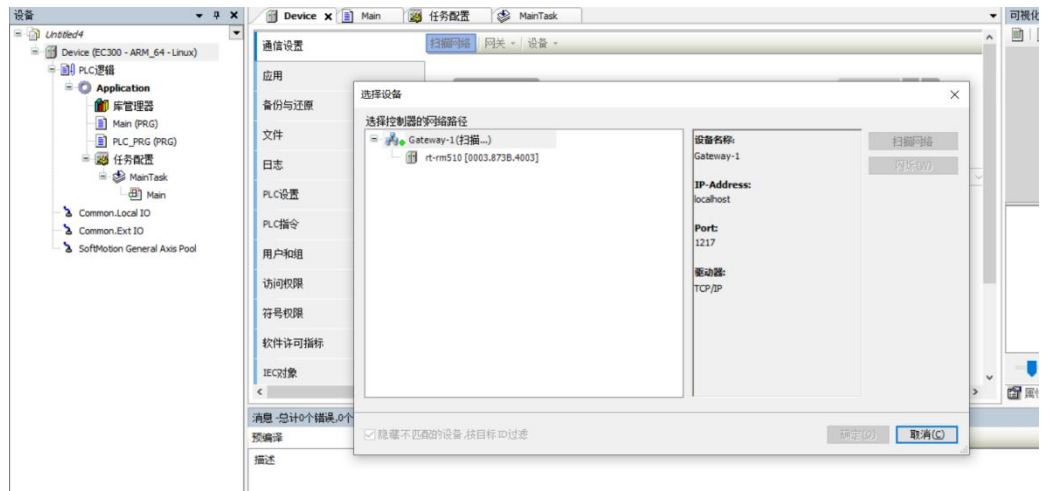


6.1.3. 登录控制器并运行项目

要将应用程序加载到控制器上时，CODESYS V3 不会自动知悉项目应该加载到哪一控制器上。这就要求用户自行行为CODESYS V3 项目分配控制器。除了分配控制器外，还必须确认应用程序没有错误。

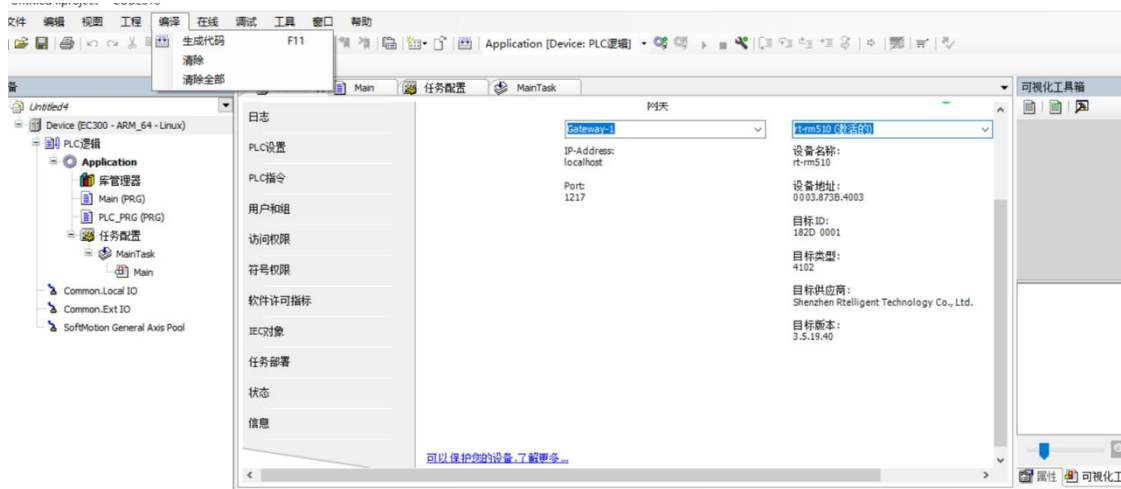
1、扫描控制器：

在设备窗口中双击“Device”打开通信设置，显示尚未分配设备。点击“扫描网络”按钮，搜索网络中已连接的控制器。



2、检查程序错误：

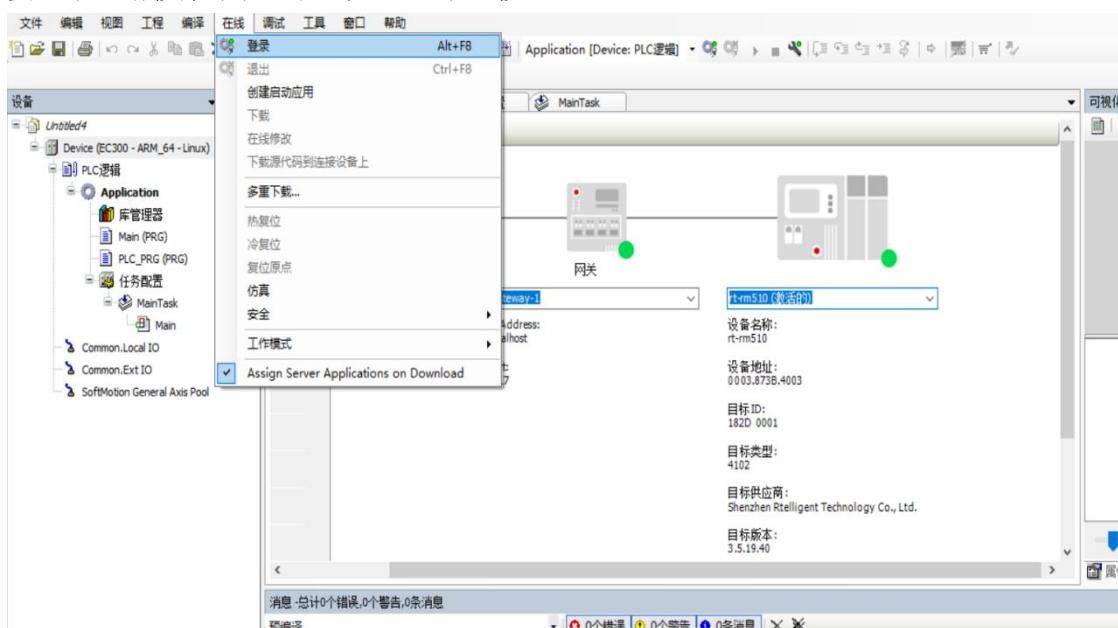
按下“编译”菜单栏下的“生成代码”项，或通过“F11”快捷键，构建程序并检查代码、可视化和设置中的错误。



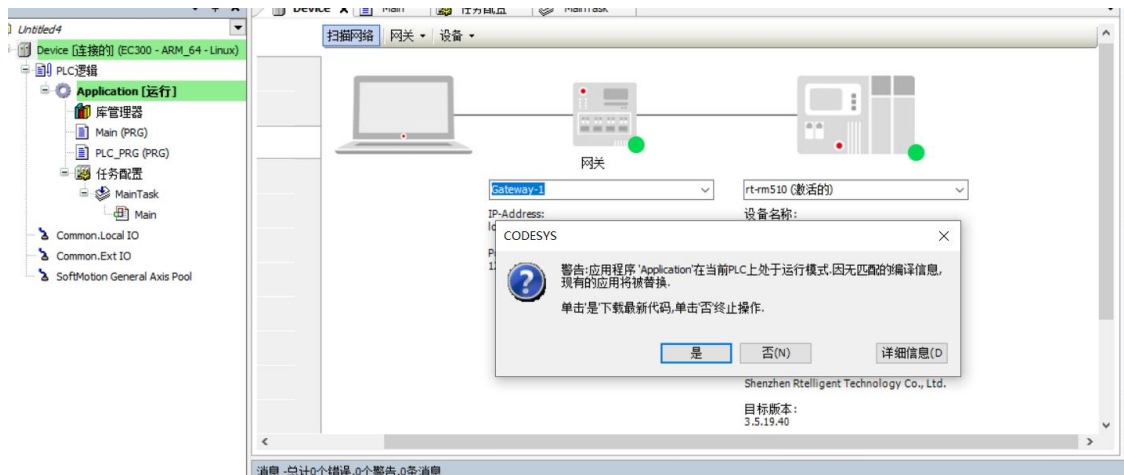
经过短暂的等待后，结果将显示在消息窗口中。如果您在创建此示例时没有出错，则应该显示“0错误”和“0警告”；如果确实发生了错误，它将显示在消息中，通过双击错误消息，CODESYS V3 将自动跳转到错误的位置，帮助您有效且轻松地修复错误。如果该项目已经完全没有错误，并且已分配了一个控制器，那么您可以将该程序加载到该控制器上。

3、加载应用：

要登录，请按菜单栏“在线->“登录”按钮。



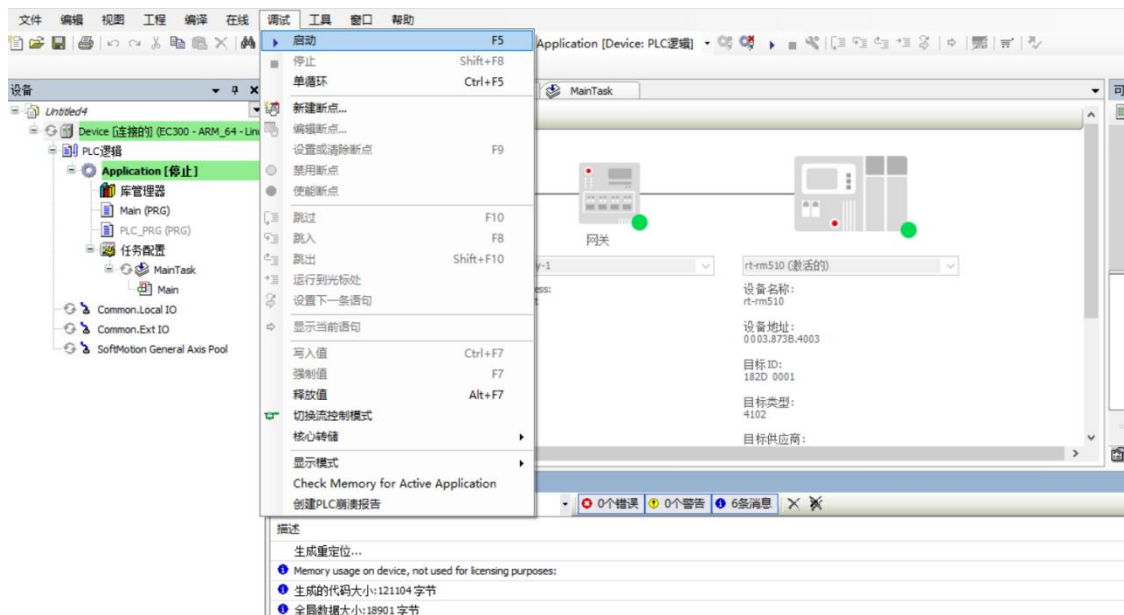
如果到目前为止在控制器上没有应用程序，则会出现如图显示的消息；如果一个应用程序已经加载到控制器上，则会出现一条消息，说明该控制器上有一个未知的应用程序。此消息可能会根据现有的应用程序是否正在运行而有所不同。



在所有情况下，请按下“是”来确认。但如果消息显示程序中仍然有错误，您需要取消登录，首先找到程序中的错误并纠正它们。然后，再以 CODESYS V3 将应用程序加载到控制器上。

4、运行应用：

当“设备”和“应用程序”后出现绿色背景，并且在后面出现“[已连接]”或“[停止]”字样时，加载过程就完成了。此时应用程序已完全加载到控制器上，但仍处于 Idle 状态，尚未运行。要启动程序，请点击菜单栏中的“调试->启动”或按“F5”键。



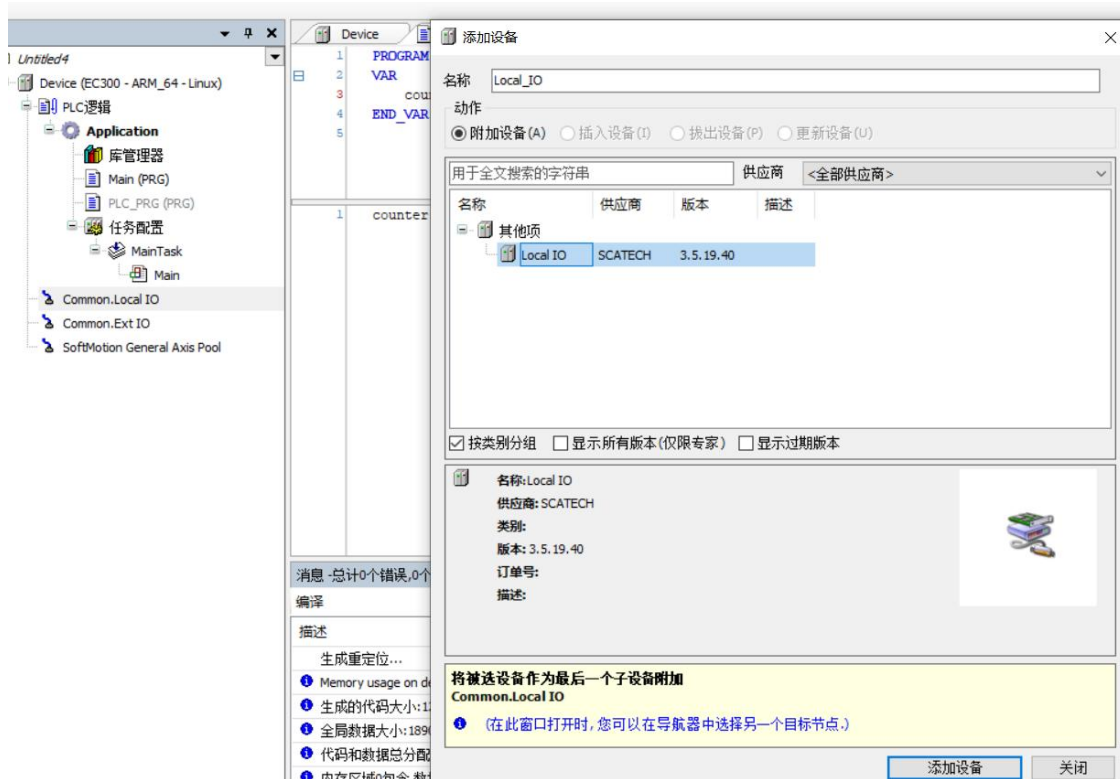
当设备窗口中的“应用程序”状态从“[停止]”更改为“[运行]”时，将在控制器上执行该程序；此时，点开之前创建的程序“Main”，可以看到计数器“counter”中的值在累加。



7. IO 配置

7.1. PLC 本机 IO 配置

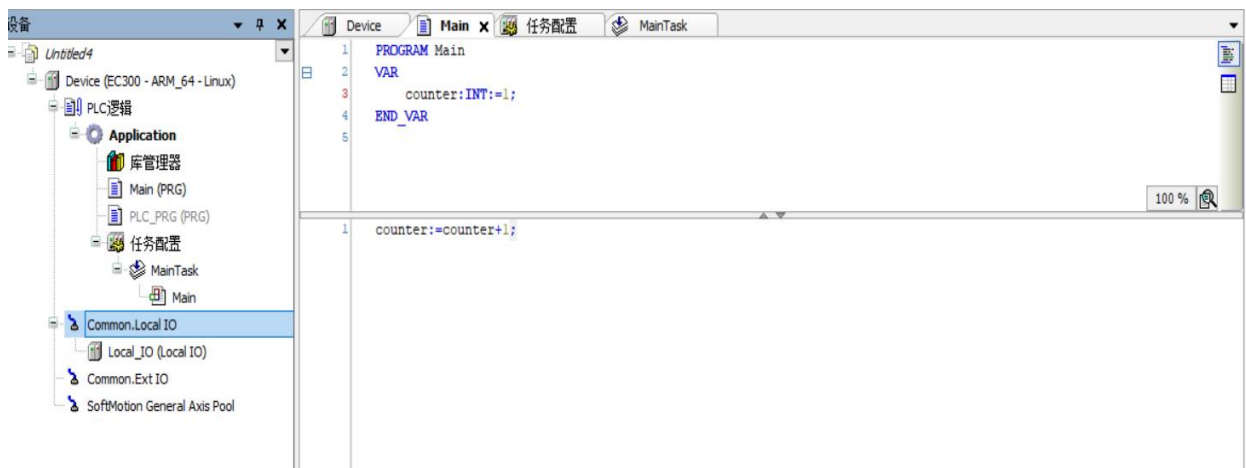
右键点击工程中的“Common.Local IO”，添加设备，选择“Local IO”，如下：



7.2. PLC 扩展 IO 模块配置

PLC 扩展 IO 模块需要用户另行购买，这里讲述如何配置使用。

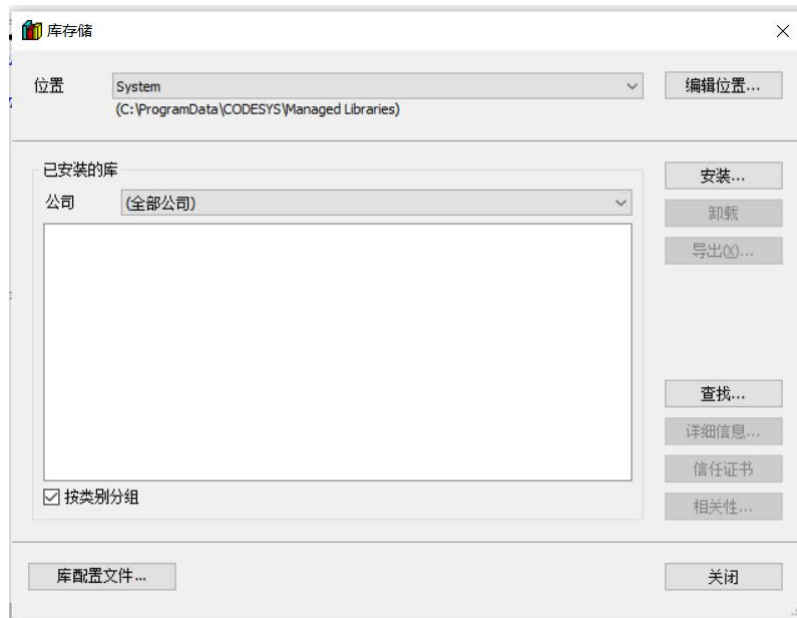
右键点击工程中的“Common.Ext IO”，添加设备，选择需要添加的扩展 IO 模块（最多可以添加8个 IO 模块），如下所示：



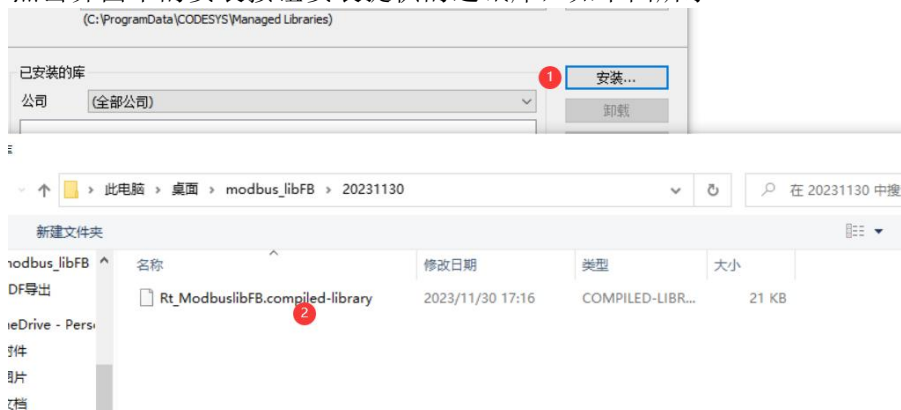
8. RT Modbus 库安装使用说明

8.1. 库安装

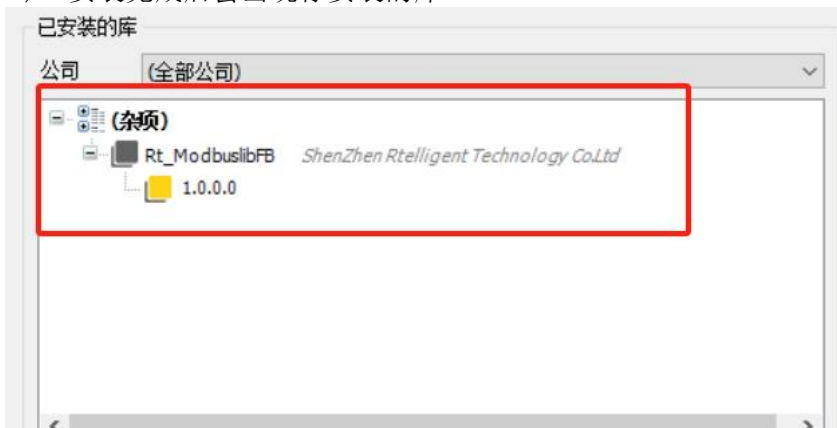
- 1) 在CodeSys工程环境上方找到 工具->库储存，点击出现下图界面



- 2) 点击界面中的安装按钮安装提供的通讯库，如下图所示



- 3) 安装完成后会出现你安装的库

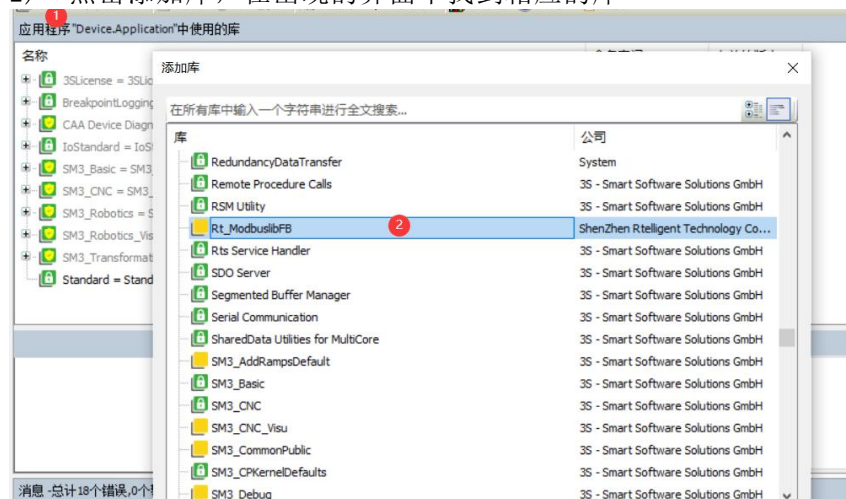


8.2. 工程内使用

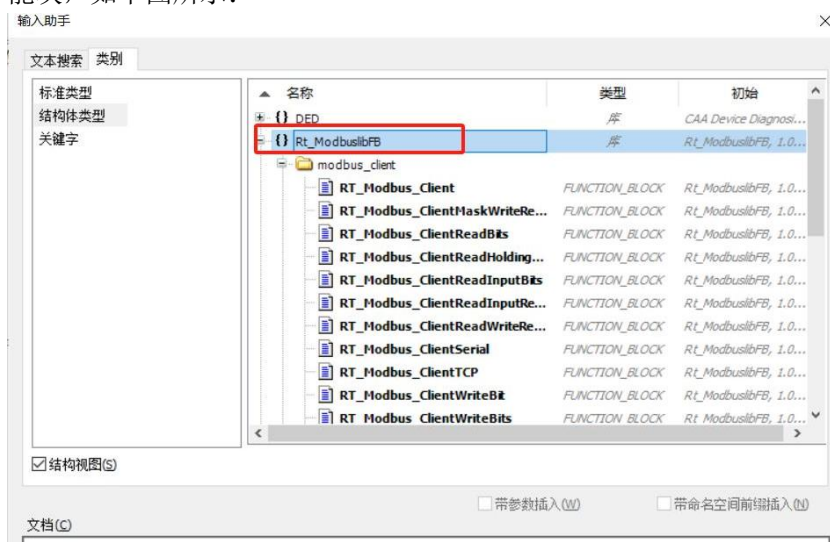
1) 在项目中找到“库管理器”，双击出现下图界面



2) 点击添加库，在出现的界面中找到相应的库



3) 至此相关库已能够在程序中使用，在程序中可以通过 F2（输入助手）来输入库中的功能块，如下图所示：

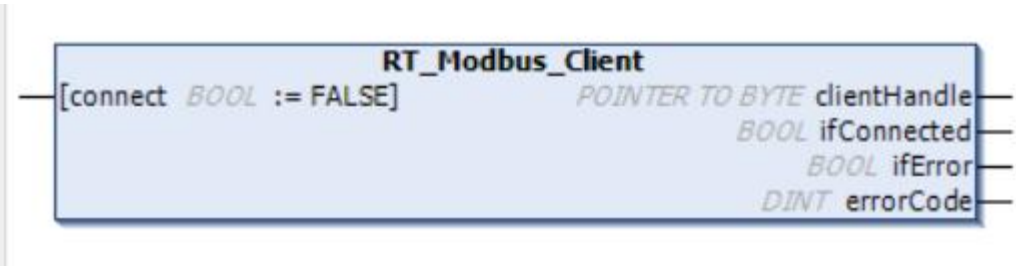


8.3. Modbus 主站接口描述

8.3.1. RT_Modbus_Client

Modbus 主站抽象基类，RTU和TCP主站的父类，子类共享主站的状态成员。

FUNCTION_BLOCK ABSTRACT RT_Modbus_Client						
	名称	类型	继承自	地..	初始化	注释
INPUT	connect	BOOL			FALSE	由False变为True时发起连接请求，由True变为False时关闭现有连接
OUTPUT	clientHandle	POINTER TO B...				modbus从站句柄
OUTPUT	ifConnected	BOOL				反应从站连接状态，True表示连接中，False表示断开
OUTPUT	ifError	BOOL				反应从站是否出错，True表示从站出错，False表示从站正常
OUTPUT	errorCode	DINT				错误码，正常情况下为0



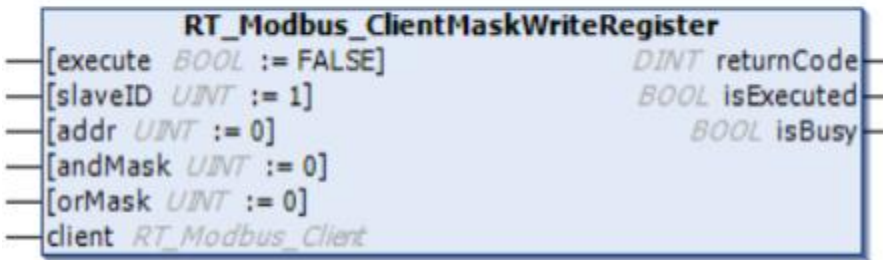
8.3.2. RT_Modbus_ClientMaskWriteRegister

屏蔽写寄存器，在指定的寄存器上进行位操作(功能码0x16)。
该功能使用“与字(andMask)”、“或字(orMask)”以及寄存器当前内容进行运算来修改指定保持寄存器的内容。可用于设置或清除寄存器中的某个位。
结果=(寄存器当前内容 && andMask) || (orMask && (~andMask))

实例如图:

For example:		
Current Contents=	Hex	Binary
And_Mask =	12	0001 0010
Or_Mask =	F2	1111 0010
(NOT And_Mask)=	25	0010 0101
Result =	0D	0000 1101
	17	0001 0111

FUNCTION_BLOCK RT_Modbus_ClientMaskWriteRegister						
	名称	类型	继承自	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行，从False变为True时读取参数
INPUT	slaveID	UINT			1	从站号，ModbusTCP连接可忽略该参数
INPUT	addr	UINT			0	寄存器地址
INPUT	andMask	UINT			0	与操作掩码
INPUT	orMask	UINT			0	或操作掩码
IN_OUT	client	RT_Modbus_Client				主站句柄，传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量，
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中



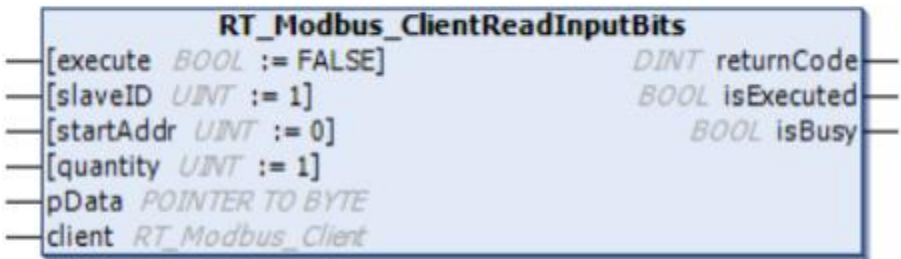
8.3.3. RT_Modbus_ClientReadBits

读取线圈或者离散量输出状态(功能码 0x1)。

8.3.4. RT_Modbus_ClientReadInputBits

读取离散量输入值(功能码 0x2)

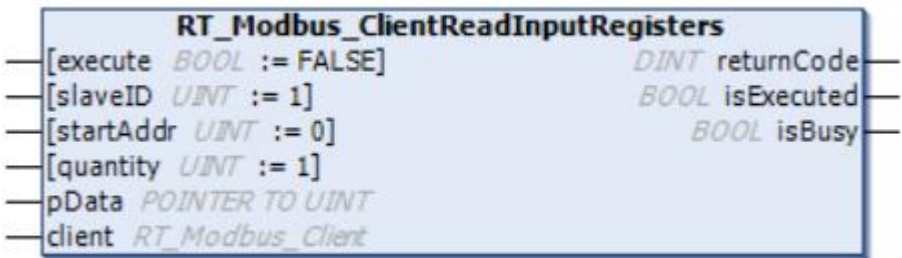
FUNCTION_BLOCK RT_Modbus_ClientReadInputBits						
	名称	类型	继承...	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
INPUT	slaveID	UINT			1	从站号，ModbusTCP连接可忽略该参数
INPUT	startAddr	UINT			0	起始地址
INPUT	quantity	UINT			1	操作数量
INPUT	pData	POINTER TO BYTE				返回读取到的数据
IN_OUT	client	RT_Modbus_Client				主站句柄，传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量，失败返回错误码。
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中



8.3.5. RT_Modbus_ClientReadInputRegisters

读取输入寄存器(功能码 0x4)

FUNCTION_BLOCK RT_Modbus_ClientReadInputRegisters						
	名称	类型	继承...	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
INPUT	slaveID	UINT			1	从站号，ModbusTCP连接可忽略该参数
INPUT	startAddr	UINT			0	起始地址
INPUT	quantity	UINT			1	操作数量
INPUT	pData	POINTER TO UINT				返回读取到的数量
IN_OUT	client	RT_Modbus_Client				主站句柄，传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量，失败返回错误码。
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中



8.3.6. RT_Modbus_ClientReadWriteRegisters

同时执行读写多个寄存器操作

FUNCTION_BLOCK RT_Modbus_ClientReadWriteRegisters						
	名称	类型	继承...	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
INPUT	slaveID	UINT			1	从站号，ModbusTCP连接可忽略该参数
INPUT	readStart...	UINT			0	需要读操作的寄存器起始地址
INPUT	readQuan...	UINT			1	需要读操作地址数量
INPUT	pReadData	POINTER TO UINT				返回读取到的数据,空间长度应该>=操作地址数量
INPUT	writeStart...	UINT			0	需要写操作的寄存器起始地址
INPUT	writeQuan...	UINT			1	需要写操作地址数量
INPUT	pWriteData	POINTER TO UINT				写入数据，数据长度应该等于操作地址数量
IN_OUT	client	RT_Modbus_Client				主站句柄，传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量，失败返回错误码。
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中

```
RT_Modbus_ClientReadWriteRegisters
[execute BOOL := FALSE]
[slaveID UINT := 1]
[readStartAddr UINT := 0]
[readQuantity UINT := 1]
pReadData POINTER TO UINT
[writeStartAddr UINT := 0]
[writeQuantity UINT := 1]
pWriteData POINTER TO UINT
client RT_Modbus_Client
```

8.3.7. RT_Modbus_ClientSerial

ModbusRTU初始化连接操作：继承自RT_Modbus_Client

FUNCTION_BLOCK RT_Modbus_ClientSerial EXTENDS RT_Modbus_Client						
	名称	类型	继承自	地址	初始化	注释
INPUT	connect	BOOL	RT_Modbus_Client		FALSE	由False变为True时发起连接请求，由True变为False时关闭现有连接
INPUT	comPort	UINT			0	需要请求连接的本地端口号，范围[0,1]
INPUT	baudrate	UDINT			9600	串口波特率
INPUT	dataBits	UDINT			8	串口数据位
INPUT	parity	UDINT			0	校验位，0=无校验，1=奇校验，2=偶校验
INPUT	stopBit	UDINT			1	停止位
INPUT	timeOut	TIME			TIME#1s0ms	通讯超时等待时间
OUTPUT	clientHandle	POINTER TO BYTE	RT_Modbus_Client			modbus从站句柄
OUTPUT	ifConnected	BOOL	RT_Modbus_Client			反应从站连接状态，True表示连接中，False表示断开
OUTPUT	ifError	BOOL	RT_Modbus_Client			反应从站是否出错，True表示从站出错，False表示从站正常
OUTPUT	errorCode	DINT	RT_Modbus_Client			错误码，正常情况下为0

```
RT_Modbus_ClientSerial
[connect BOOL := FALSE]
[comPort UINT := 0]
[baudrate UDINT := 9600]
[dataBits UDINT := 8]
[parity UDINT := 0]
[stopBit UDINT := 1]
[timeOut TIME := TIME#1s0ms]
POINTER TO BYTE clientHandle
BOOL ifConnected
BOOL ifError
DINT errorCode
```

8.3.8. RT_Modbus_ClientTCP

ModbusTCP初始化连接操作，继承自RT_Modbus_Client

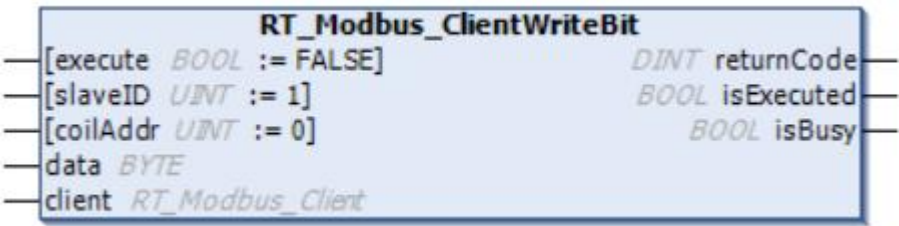
FUNCTION_BLOCK RT_Modbus_ClientTCP EXTENDS RT_Modbus_Client						
	名称	类型	继承自	地址	初始化	注释
INPUT	connect	BOOL	RT_Modbus_Client		FALSE	由False变为True时发起连接请求，由True变为False时关闭现有连接
INPUT	ipAddr	STRING			*	IP地址
INPUT	port	UDINT			502	端口号，ModbusTCP默认使用端口号为502
INPUT	timeOut	TIME			TIME#1s0ms	通讯超时等待时间
OUTPUT	clientHandle	POINTER TO BYTE	RT_Modbus_Client			modbus从站句柄
OUTPUT	ifConnected	BOOL	RT_Modbus_Client			反应从站连接状态，True表示连接中，False表示断开
OUTPUT	ifError	BOOL	RT_Modbus_Client			反应从站是否出错，True表示从站出错，False表示从站正常
OUTPUT	errorCode	DINT	RT_Modbus_Client			错误码，正常情况下为0



8.3.9. RT_Modbus_ClientWriteBit

写单个线圈或者单个离散量(功能码 0x5)

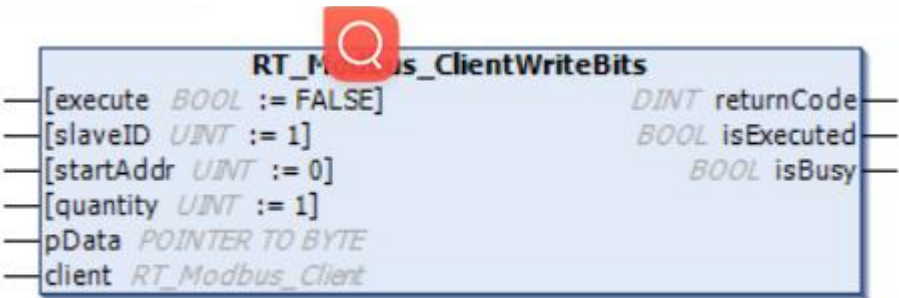
FUNCTION_BLOCK RT_Modbus_ClientWriteBit						
	名称	类型	继承...	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
INPUT	slaveID	UINT			1	从站号，ModbusTCP连接可忽略该参数
INPUT	coilAddr	UINT			0	操作地址
INPUT	data	BYTE				写入数据
IN_OUT	client	RT_Modbus_Client				主站句柄，传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量，失败返回错误码。
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中



8.3.10. RT_Modbus_ClientWriteBits

写多个线圈(功能码 0x5)

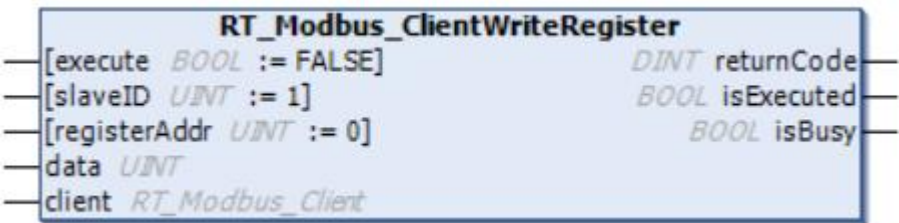
FUNCTION_BLOCK RT_Modbus_ClientWriteBits						
	名称	类型	继承...	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
INPUT	slaveID	UINT			1	从站号，ModbusTCP连接可忽略该参数
INPUT	startAddr	UINT			0	起始地址
INPUT	quantity	UINT			1	操作数量
INPUT	pData	POINTER TO BYTE				写入数据，数据长度应该等于操作数量
IN_OUT	client	RT_Modbus_Client				主站句柄，传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量，失败返回错误码。
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中



8.3.11. RT_Modbus_ClientWriteRegister

写单个保持寄存器(功能码 0x6)

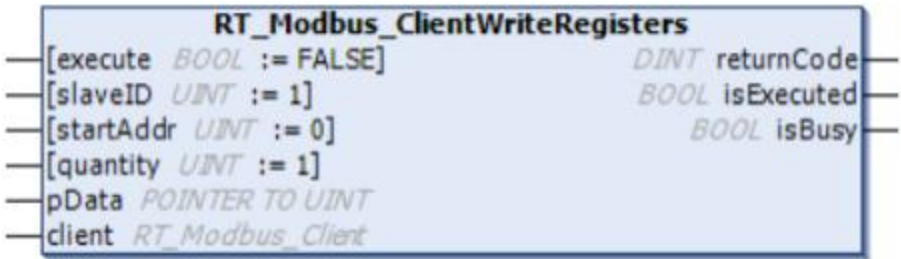
FUNCTION_BLOCK RT_Modbus_ClientWriteRegister						
	名称	类型	继承...	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
INPUT	slaveID	UINT			1	从站号，ModbusTCP连接可忽略该参数
INPUT	registerAddr	UINT			0	寄存器地址
INPUT	data	UINT				需要写入的数据
IN_OUT	client	RT_Modbus_Client				主站句柄，传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量，失败返回错误码。
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中



8.3.12. RT_Modbus_ClientWriteRegisters

写多个保持寄存器(功能码 0x10)

FUNCTION_BLOCK RT_Modbus_ClientWriteRegisters						
	名称	类型	继承...	地址	初始化	注释
INPUT	execute	BOOL			FALSE	是否执行, 从False变为True时读取参数并执行
INPUT	slaveID	UINT			1	从站号, ModbusTCP连接可忽略该参数
INPUT	startAddr	UINT			0	寄存器起始地址
INPUT	quantity	UINT			1	操作数量
INPUT	pData	POINTER TO UINT				写入数据, 数据长度应该等于操作数量
IN_OUT	client	RT_Modbus_Client				主站句柄, 传入主站功能块的实例
OUTPUT	returnCode	DINT				执行返回。执行成功返回操作的数量, 失败返回错误码。
OUTPUT	isExecuted	BOOL			FALSE	操作是否执行完成
OUTPUT	isBusy	BOOL			FALSE	操作是否在执行中

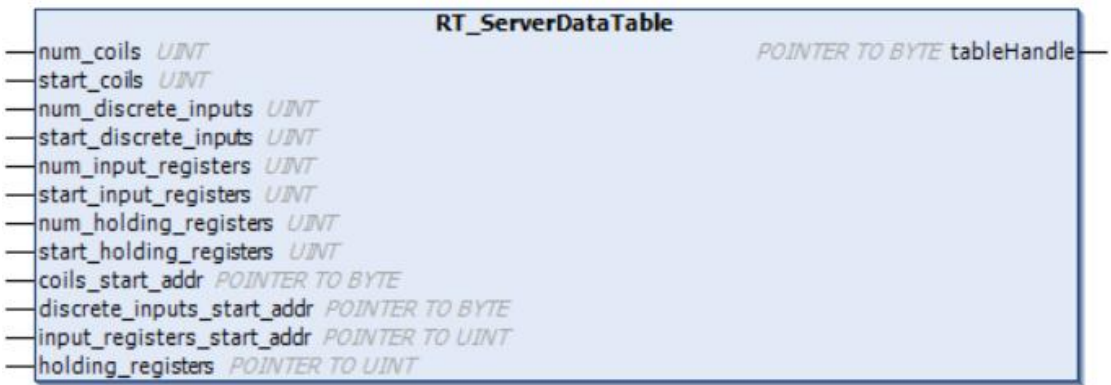


8.4. Modbus 从站接口描述

8.4.1. RT_ServerDataTable

Modbus从站数据映射表, 寄存器数量应该等于传入指针指向的数组长度。

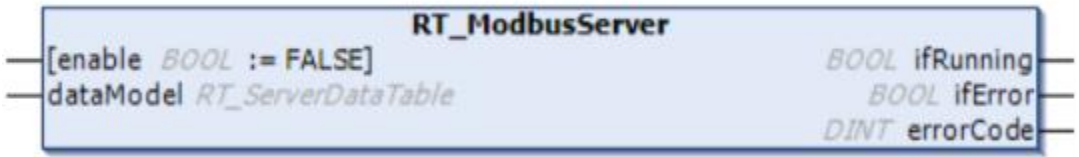
FUNCTION_BLOCK RT_ServerDataTable						
	名称	类型	继承自	地址	初始化	注释
INPUT	num_coils	UINT				线圈寄存器的数量
INPUT	start_coils	UINT				线圈寄存器的起始地址
INPUT	num_discrete_inputs	UINT				离散输入寄存器数量
INPUT	start_discrete_inputs	UINT				离散输入寄存器的起始地址
INPUT	num_input_registers	UINT				输入寄存器的数量
INPUT	start_input_registers	UINT				输入寄存器的起始地址
INPUT	num_holding_registers	UINT				保持寄存器的数量
INPUT	start_holding_registers	UINT				保持寄存器的起始地址
INPUT	coils_start_addr	POINTER TO BYTE				线圈寄存器起始地址
INPUT	discrete_inputs_start_addr	POINTER TO BYTE				离散输入寄存器起始地址
INPUT	input_registers_start_addr	POINTER TO UINT				输入寄存器起始地址
INPUT	holding_registers	POINTER TO UINT				保持寄存器起始地址
OUTPUT	tableHandle	POINTER TO BYTE				地址空间句柄



8. 4. 2. RT_ModbusServer

Modbus服务端基类，TCP和RTU服务端继承自此类。主要包含服务端的运行状态和共同输入参数。

FUNCTION_BLOCK ABSTRACT RT_ModbusServer						
	名称	类型	继承自	地址	初始化	注释
INPUT	enable	BOOL			FALSE	是否使能从站，有False变为True时读取输入参数并使能从站；由True变为False时关闭从站。
IN_OUT	dataModel	RT_ServerDataTable				从站运行的输入参数，包含寄存器和线圈的数量及起始地址
OUTPUT	ifRunning	BOOL				返回从站是否运行，True表示运行中，False表示未运行
OUTPUT	ifError	BOOL				返回从站是否出错，True表示从站出错，False表示从站正常
OUTPUT	errorCode	DINT				从站运行错误码，0表示正常



8. 4. 3. RT_Modbus_ServerSerial

ModbusRTU从站初始化，继承自RT_ModbusServer

FUNCTION_BLOCK RT_Modbus_ServerSerial EXTENDS RT_ModbusServer						
	名称	类型	继承自	地址	初始化	注释
INPUT	enable	BOOL	RT_ModbusServer		FALSE	是否使能从站，有False变为True时读取输入参数并使能从站；由True变为False时关闭从站。
IN_OUT	dataModel	RT_ServerDataTable	RT_ModbusServer			从站运行的输入参数，包含寄存器和线圈的数量及起始地址
INPUT	slaveID	UINT			1	从站号
INPUT	comPort	UINT			0	从站使用的串口号,范围[0,1]
INPUT	baudrate	UDINT			9600	从站串口波特率
INPUT	dataBits	UDINT			8	从站串口数据位
INPUT	parity	UDINT			0	从站串口校验位，0=无校验，1=奇校验，2=偶校验
INPUT	stopBit	UDINT			1	从站串口停止位
OUTPUT	ifRunning	BOOL	RT_ModbusServer			返回从站是否运行，True表示运行中，False表示未运行
OUTPUT	ifError	BOOL	RT_ModbusServer			返回从站是否出错，True表示从站出错，False表示从站正常
OUTPUT	errorCode	DINT	RT_ModbusServer			从站运行错误码，0表示正常

RT_Modbus_ServerSerial

[enable *BOOL* := FALSE]

[dataModel *RT_ServerDataTable*]

[slaveID *UINT* := 1]

[comPort *UINT* := 0]

[baudrate *UDINT* := 9600]

[dataBits *UDINT* := 8]

[parity *UDINT* := 0]

[stopBit *UDINT* := 1]

8. 4. 4. RT_Modbus_ServerTCP

ModbusTCP从站初始化，继承自RT_ModbusServer

FUNCTION_BLOCK RT_Modbus_ServerTCP EXTENDS [RT_ModbusServer](#)

	名称	类型	继承自	地址	初始化	注释
INPUT	enable	BOOL	RT_ModbusServer		FALSE	是否使能从站，有False变为True时读取输入参数并使能从站；由True变为False时关闭从站。
IN_OUT	dataModel	RT_ServerDataTable	RT_ModbusServer			从站运行的输入参数，包含寄存器和线圈的数量及起始地址
INPUT	netInterfaceName	STRING			'eth0'	从站网卡名称
INPUT	port	UINT			502	从站监听端口
OUTPUT	ifRunning	BOOL	RT_ModbusServer			返回从站是否运行，True表示运行中，False表示未运行
OUTPUT	ifError	BOOL	RT_ModbusServer			返回从站是否出错，True表示从站出错，False表示从站正常
OUTPUT	errorCode	DINT	RT_ModbusServer			从站运行错误码，0表示正常

RT_Modbus_ServerTCP

[enable *BOOL* := FALSE]

[dataModel *RT_ServerDataTable*]

[netInterfaceName *STRING* := 'eth0']]

[port *UINT* := 502]

BOOL ifRunning

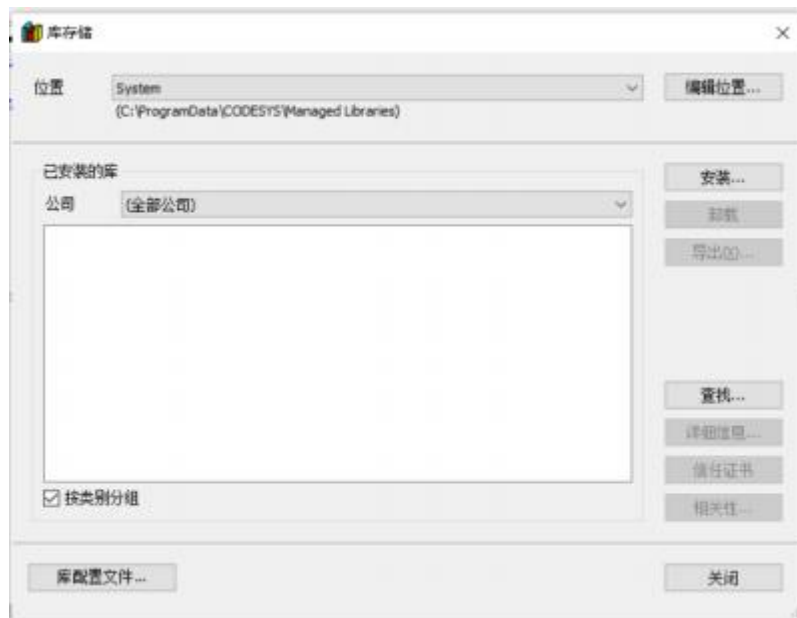
BOOL ifError

DINT errorCode

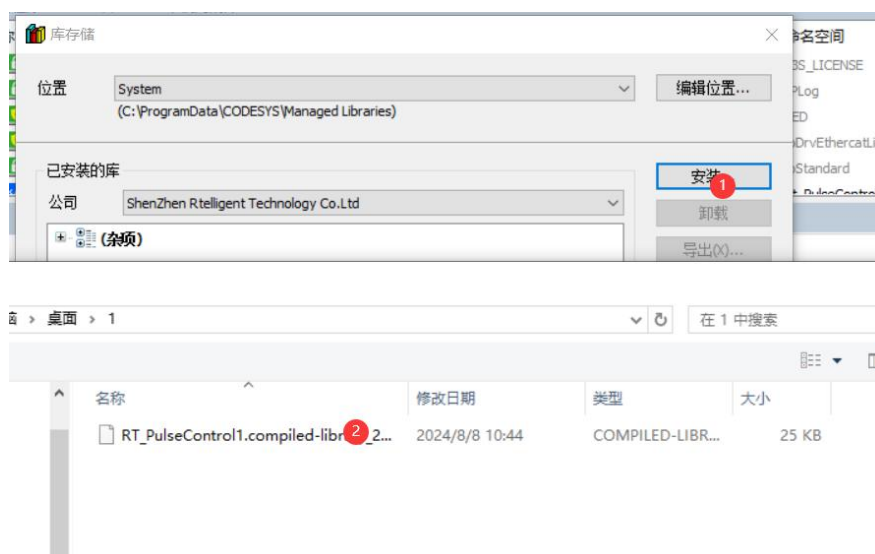
9. Rt_PulseControlFB_脉冲控制库使用 说明

9.1. 库安装

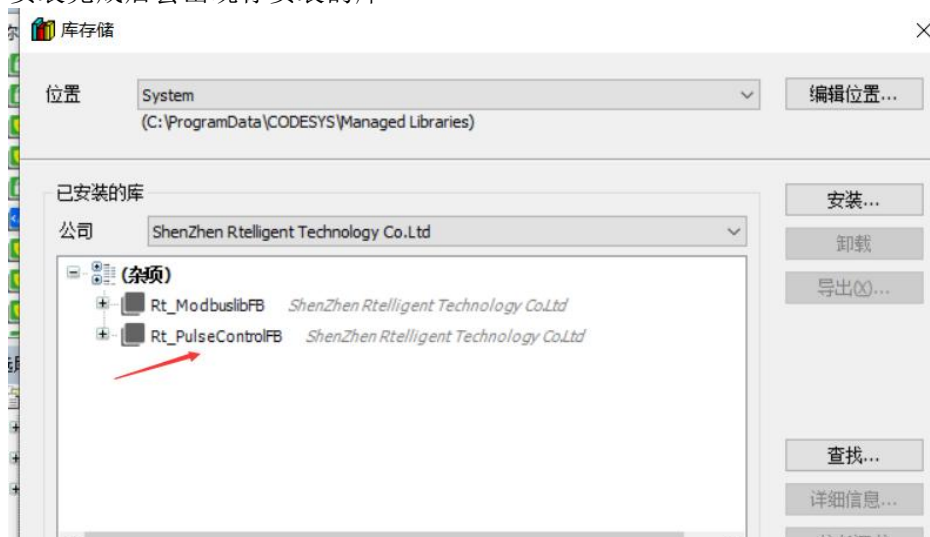
- 1) 在CodeSys工程环境上方找到 工具->库储存，点击出现下图界面



- 2) 点击界面中的安装按钮安装提供的通讯库，如下图所示



3) 安装完成后会出现你安装的库

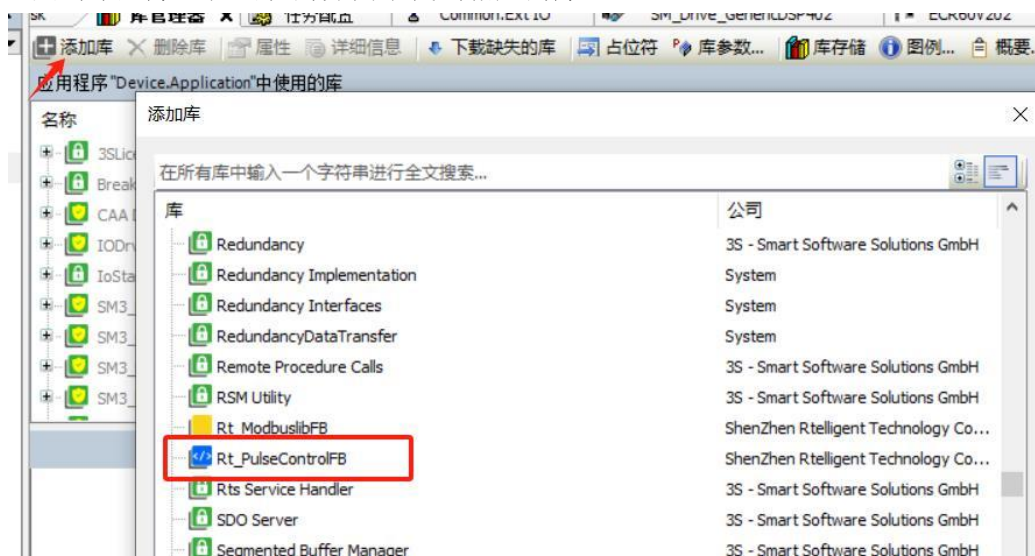


9.2. 工程内使用

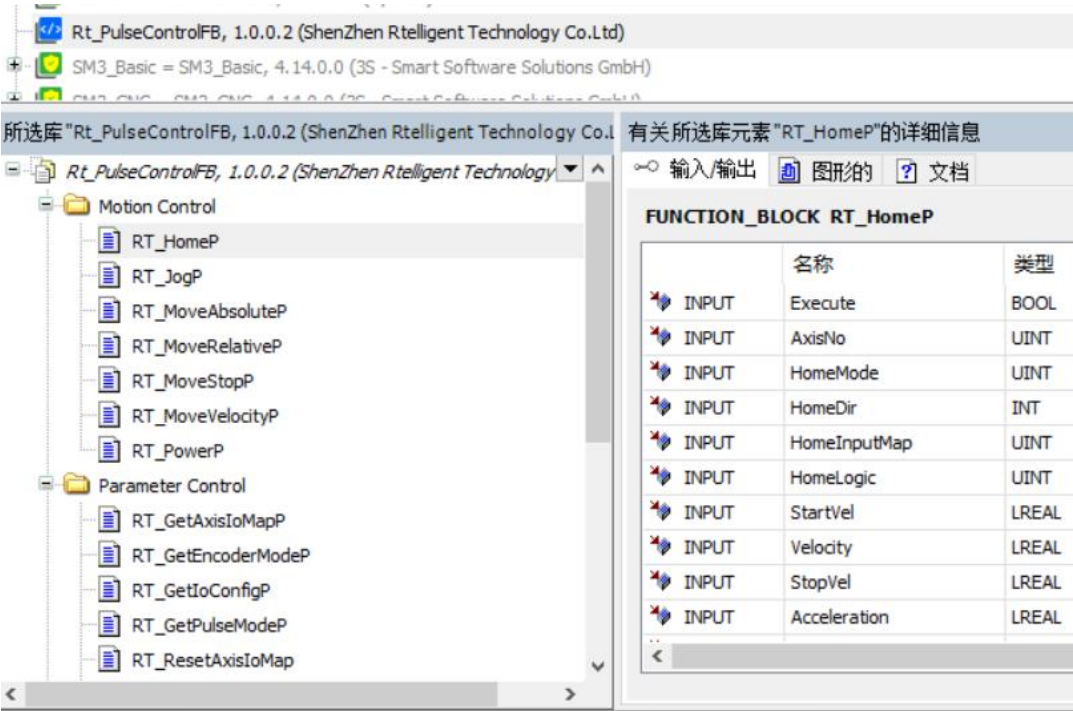
1) 在项目中找到“库管理器”，双击出现下图界面



2) 点击添加库，在出现的界面中找到相应的库



- 3) 至此相关库已能够在程序中使用，在程序中可以通过 F2（输入助手）来输入库中的功能块，如下图所示：



9.3. 功能块接口描述

9.3.1.脉冲轴初始化函数

9.3.1.1.RT_PulseControlInit



9.3.1.2. RT_PulseControlFree

FUNCTION RT_PulseControlFree

脉冲模块资源释放
一般无需主动调用

名称	类型	继承自	地址	初始化	注释
RT_PulseControlFree	DWORD				返回值

RT_PulseControlFree

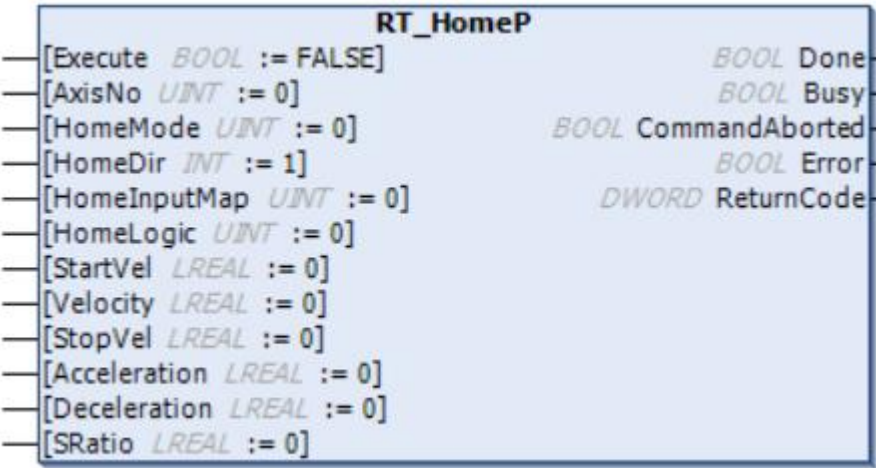
DWORD RT_PulseControlFree

9.3.2. 运动控制功能块

9.3.2.1. RT_HomeP

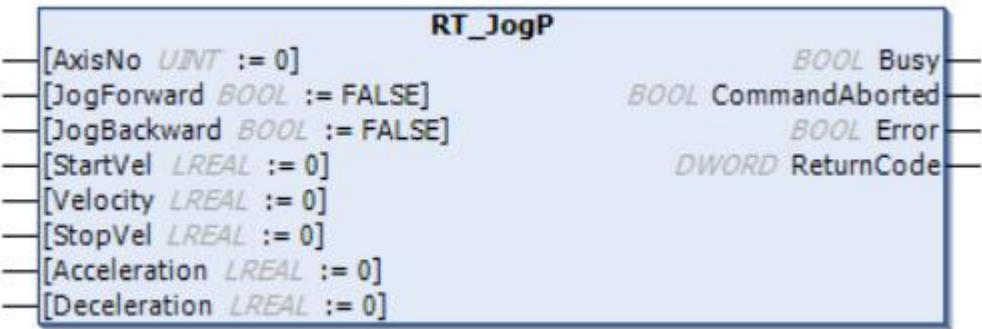
本指令用于实现脉冲轴回零
Excuse由FALSE变为TRUE触发启动
回零模式 (HomeMode) : 0——次回零
1——次回零+反找
2——两次回零
原点信号映射(HomeInputMap):取值范围[0,15]对应通用输入信号Input0——Input15

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行, 从False变为True时读取参数并执行
AxisNo	UINT			0	轴号, 范围[0,7]
HomeMode	UINT			0	回零模式
HomeDir	INT			1	回零方向, 1——正方向, -1——负方向
HomeInputMap	UINT			0	原点信号映射,取值范围[0,15]对应通用输入信号Input0——Input15
HomeLogic	UINT			0	原点信号有效电平,0——低电平有效,1——高电平有效
StartVel	LREAL			0	回零起始速度(Pulse/s)
Velocity	LREAL			0	回零最大速度(Pulse/s)
StopVel	LREAL			0	回零停止速度(Pulse/s)
Acceleration	LREAL			0	回零加速度(Pulse/s^2)
Deceleration	LREAL			0	回零减速度(Pulse/s^2)
SRatio	LREAL			0	加速减速段的S段比例,范围[0,1],当取值为0时为梯形加速减速曲线
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9. 3. 2. 2. RT_JogP

FUNCTION_BLOCK RT_JogP					
手动控制轴朝指定方向运动					
名称	类型	继承自	地址	初始化	注释
AxisNo	UINT			0	轴号, 范围[0,7]
JogForward	BOOL			FALSE	如果JogForward为TRUE, 轴将按给定参数朝正向运动, 如果JogBackward同时为TRUE, 轴不动
JogBackward	BOOL			FALSE	如果JogBackward为TRUE, 轴将按给定参数朝负向运动, 如果JogForward同时为TRUE, 轴不动
StartVel	LREAL			0	起始速度(Pulse/s)
Velocity	LREAL			0	最大速度(Pulse/s)
StopVel	LREAL			0	停止速度(Pulse/s)
Acceleration	LREAL			0	加速度(Pulse/s^2)
Deceleration	LREAL			0	减速度(Pulse/s^2)
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

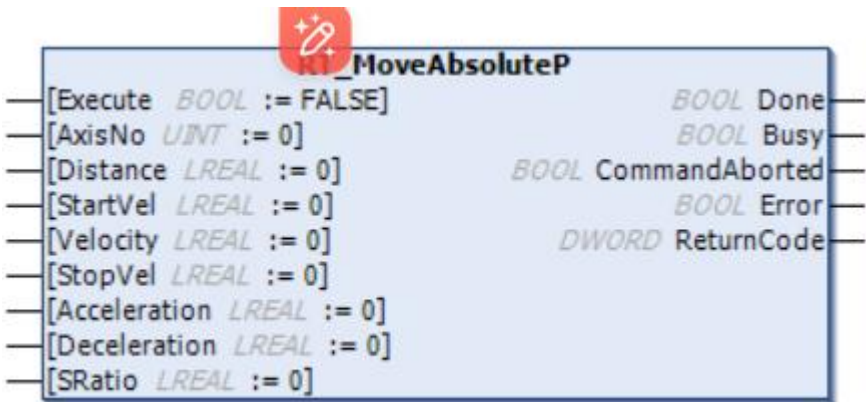


9.3.2.3. RT_MoveAbsoluteP

FUNCTION_BLOCK RT_MoveAbsoluteP

脉冲轴单轴绝对位置点位运动功能块
Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行, 从False变为True时读取参数并执行
AxisNo	UINT			0	轴号, 范围[0,7]
Distance	LREAL			0	目标位置(Pulse)
StartVel	LREAL			0	起始速度(Pulse/s)
Velocity	LREAL			0	最大速度(Pulse/s)
StopVel	LREAL			0	停止速度(Pulse/s)
Acceleration	LREAL			0	加速度(Pulse/s ²)
Deceleration	LREAL			0	减速度(Pulse/s ²)
SRatio	LREAL			0	加减速段的S段比例,范围[0,1],当取值为0时为梯形加减速曲线
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

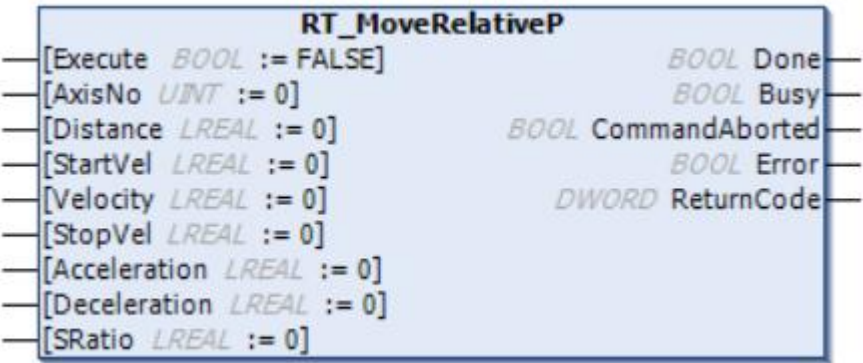


9.3.2.4. RT_MoveRelativeP

FUNCTION_BLOCK RT_MoveRelativeP

脉冲轴单轴相对位置点位运动功能块
Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行, 从False变为True时读取参数并执行
AxisNo	UINT			0	轴号, 范围[0,7]
Distance	LREAL			0	目标位置(Pulse)
StartVel	LREAL			0	起始速度(Pulse/s)
Velocity	LREAL			0	最大速度(Pulse/s)
StopVel	LREAL			0	停止速度(Pulse/s)
Acceleration	LREAL			0	加速度(Pulse/s ²)
Deceleration	LREAL			0	减速度(Pulse/s ²)
SRatio	LREAL			0	加减速段的S段比例,范围[0,1],当取值为0时为梯形加减速曲线
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9.3.2.5. RT_MoveStopP

FUNCTION_BLOCK RT_MoveStopP

脉冲运动停止功能块
Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
StopMode	UINT			0	停止模式，0——减速停止；1——立即停止
StopTime	LREAL			0.1	停止时间(s)，对应停止模式0的时候的减速停止时间
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

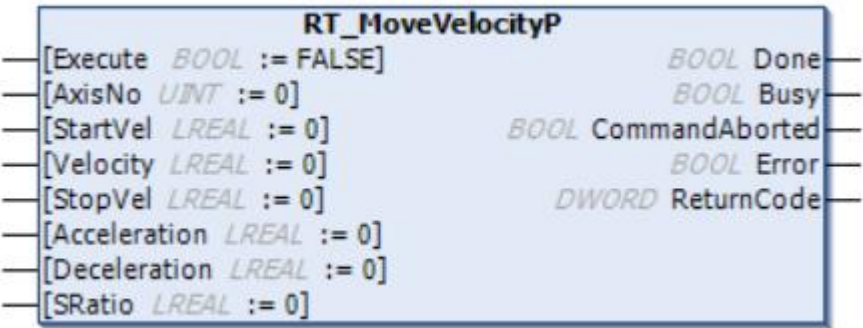


9.3.2.6. RT_MoveVelocityP

FUNCTION_BLOCK RT_MoveVelocityP

本指令用于实现脉冲轴按照设定参数，保持设定的最大速度运动
Excuse由FALSE变为TRUE触发启动

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数
AxisNo	UINT			0	轴号，范围[0,7]
StartVel	LREAL			0	起始速度(Pulse/s)
Velocity	LREAL			0	最大速度(Pulse/s)(符号代表运行方向)
StopVel	LREAL			0	停止速度(Pulse/s)
Acceleration	LREAL			0	加速度(Pulse/s^2)
Deceleration	LREAL			0	减速度(Pulse/s^2)
SRatio	LREAL			0	加减速段的S段比例,范围[0,1],当取值
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRU
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

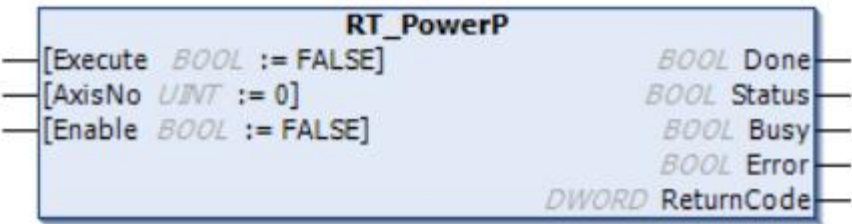


9. 3. 2. 7. RT_PowerP

FUNCTION_BLOCK RT_PowerP

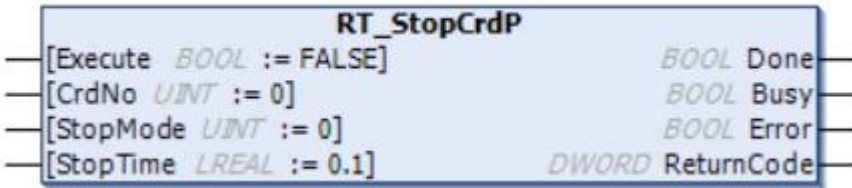
脉冲运动轴启用功能块
当想要控制相应轴脉冲时必须确保改轴已启用，否则将会返回错误码
Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号,范围[0,7]
Enable	BOOL			FALSE	脉冲轴是否启用
Done	BOOL				功能块是否执行完成
Status	BOOL				当前轴是否处于使能状态
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9.3.2.8. RT_StopCrdP

FUNCTION_BLOCK RT_StopCrdP					
停止插补系运动功能块					
Execute由False变为True时读取参数并执行一次					
名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
CrdNo	UINT			0	插补系索引[0,3],不同的插补系可以独立执行插补运动
StopMode	UINT			0	停止模式，0——减速停止；1——立即停止
StopTime	LREAL			0.1	停止时间(s)，对应停止模式0的时候的减速停止时间
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

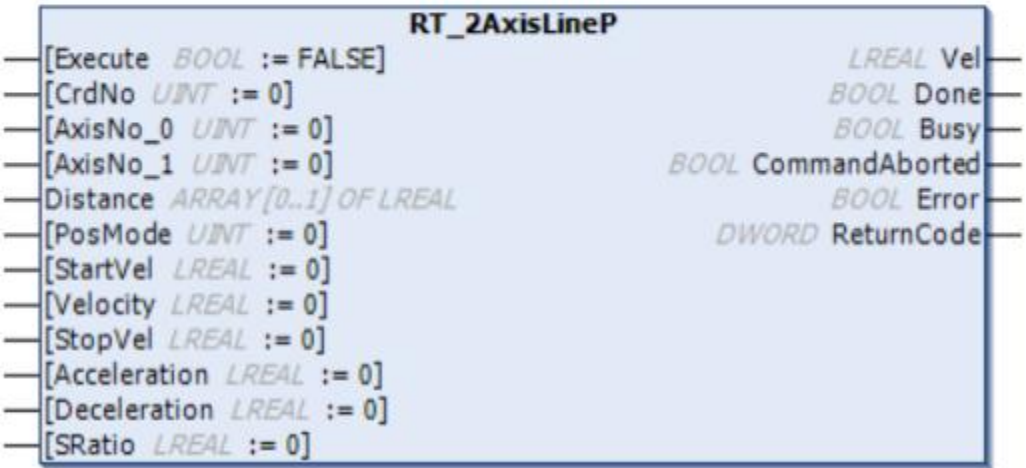


9.3.2.9. RT_2AxisLineP

FUNCTION_BLOCK RT_2AxisLineP

脉冲轴两轴直线插补功能块
Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行, 从False变为True时读取参数并执行
CrdNo	UINT			0	插补系索引[[0,3],不同的插补系可以独立执行插补运动
AxisNo_0	UINT			0	参与插补的0轴轴号, 范围[0,7]
AxisNo_1	UINT			0	参与插补的1轴轴号, 范围[0,7]
Distance	ARRAY [0..1] OF LREAL				插补轴的目标位置(Pulse)
PosMode	UINT			0	位置模式, 0表示相对运动, 1表示绝对运动
StartVel	LREAL			0	插补起始速度(Pulse/s)
Velocity	LREAL			0	插补最大速度(Pulse/s)
StopVel	LREAL			0	插补停止速度(Pulse/s)
Acceleration	LREAL			0	插补加速度(Pulse/s^2)
Deceleration	LREAL			0	插补减速度(Pulse/s^2)
SRatio	LREAL			0	插补加减速段的S段比例,范围[0,1],当取值为0时为梯形加减速曲线
Vel	LREAL				当前插补系运动的速度(Pulse/s)
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9.3.2.10. RT_3AxisLineP

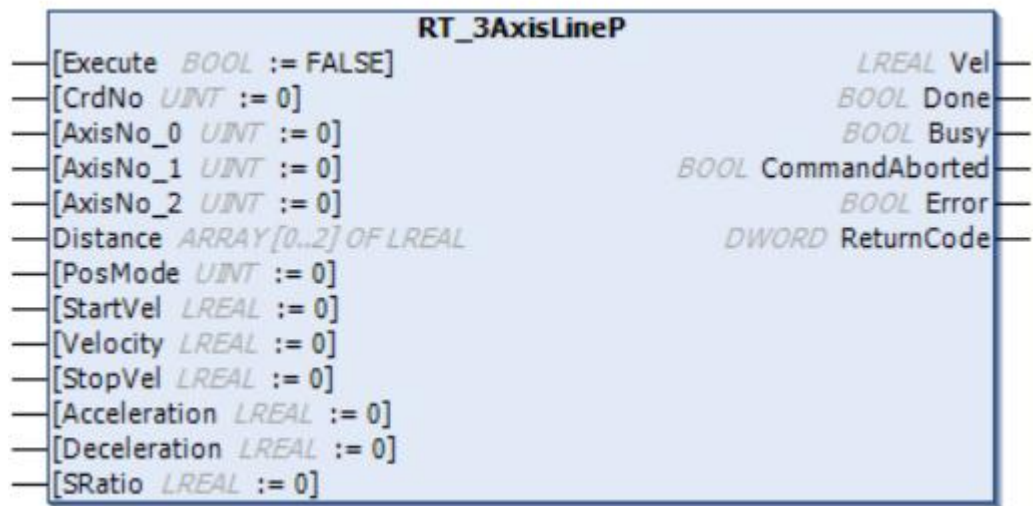
FUNCTION_BLOCK RT_3AxisLineP



脉冲轴两轴直线插补功能块

Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
CrdNo	UINT			0	插补索引[0,3],不同的插补系可以独立执行插补运动
AxisNo_0	UINT			0	参与插补的0轴轴号，范围[0,7]
AxisNo_1	UINT			0	参与插补的1轴轴号，范围[0,7]
AxisNo_2	UINT			0	参与插补的2轴轴号，范围[0,7]
Distance	ARRAY [0..2] OF LREAL				插补轴的目标位置(Pulse)
PosMode	UINT			0	位置模式，0表示相对运动，1表示绝对运动
StartVel	LREAL			0	插补起始速度(Pulse/s)
Velocity	LREAL			0	插补最大速度(Pulse/s)
StopVel	LREAL			0	插补停止速度(Pulse/s)
Acceleration	LREAL			0	插补加速度(Pulse/s ²)
Deceleration	LREAL			0	插补减速度(Pulse/s ²)
SRatio	LREAL			0	插补加减速段的S段比例,范围[0,1],当取值为0时为梯形加减速曲线
Vel	LREAL				当前插补系运动的速度(Pulse/s)
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9.3.2.11. RT_2AxisCircleC

FUNCTION_BLOCK RT_2AxisCircleC



脉冲轴两轴圆弧插补（终点圆心）功能块
Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
CrdNo	UINT			0	插补系索引[0,3],不同的插补系可以独立执行插补运动
AxisNo_0	UINT			0	参与插补的0轴轴号，范围[0,7]
AxisNo_1	UINT			0	参与插补的1轴轴号，范围[0,7]
EndPos	ARRAY [0..1] OF LREAL				插补轴的目标位置(Pulse)
CenPos	ARRAY [0..1] OF LREAL				插补圆弧的圆心位置
ArcDir	UINT			0	插补方向，0--顺时针，1--逆时针
RunCycle	UINT			0	圆弧旋转的圈数
PosMode	UINT			0	位置模式，0表示相对运动，1表示绝对运动
StartVel	LREAL			0	插补起始速度(Pulse/s)
Velocity	LREAL			0	插补最大速度(Pulse/s)
StopVel	LREAL			0	插补停止速度(Pulse/s)
Acceleration	LREAL			0	插补加速度(Pulse/s^2)
Deceleration	LREAL			0	插补减速度(Pulse/s^2)
SRatio	LREAL			0	插补加减速段的S段比例,范围[0,1],当取值为0时为梯形加减速曲线
Vel	LREAL				当前插补系运动的速度(Pulse/s)
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

RT_2AxisCircleC	
[Execute <i>BOOL</i> := FALSE]	<i>LREAL</i> Vel
[CrdNo <i>UINT</i> := 0]	<i>BOOL</i> Done
[AxisNo_0 <i>UINT</i> := 0]	<i>BOOL</i> Busy
[AxisNo_1 <i>UINT</i> := 0]	<i>BOOL</i> CommandAborted
EndPos <i>ARRAY [0..1] OF LREAL</i>	<i>BOOL</i> Error
CenPos <i>ARRAY [0..1] OF LREAL</i>	<i>DWORD</i> ReturnCode
[ArcDir <i>UINT</i> := 0]	
[RunCycle <i>UINT</i> := 0]	
[PosMode <i>UINT</i> := 0]	
[StartVel <i>LREAL</i> := 0]	
[Velocity <i>LREAL</i> := 0]	
[StopVel <i>LREAL</i> := 0]	
[Acceleration <i>LREAL</i> := 0]	
[Deceleration <i>LREAL</i> := 0]	
[SRatio <i>LREAL</i> := 0]	

9.3.2.12. RT_2AxisCircleR

FUNCTION_BLOCK RT_2AxisCircleR

脉冲轴两轴圆弧插补（终点半径）功能块

Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行。从False变为True时读取参数并执行
CrdNo	UINT			0	插补索引[0,3],不同的插补系可以独立执行插补运动
AxisNo_0	UINT			0	参与插补的0轴轴号, 范围[0,7]
AxisNo_1	UINT			0	参与插补的1轴轴号, 范围[0,7]
EndPos	ARRAY [0..1] OF LREAL				插补轴的目标位置(Pulse)
ArcRadius	LREAL			0	圆弧半径
ArcDir	UINT			0	插补方向, 0--表示劣弧, 1--表示优弧
RunCycle	UINT			0	圆弧旋转的圈数
PosMode	UINT			0	位置模式, 0表示相对运动, 1表示绝对运动
StartVel	LREAL			0	插补起始速度(Pulse/s)
Velocity	LREAL			0	插补最大速度(Pulse/s)
StopVel	LREAL			0	插补停止速度(Pulse/s)
Acceleration	LREAL			0	插补加速度(Pulse/s ²)
Deceleration	LREAL			0	插补减速度(Pulse/s ²)
SRatio	LREAL			0	插补加减速段的S段比例,范围[0,1],当取值为0时为梯形加减速曲线
Vel	LREAL				当前插补系运动的速度(Pulse/s)
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



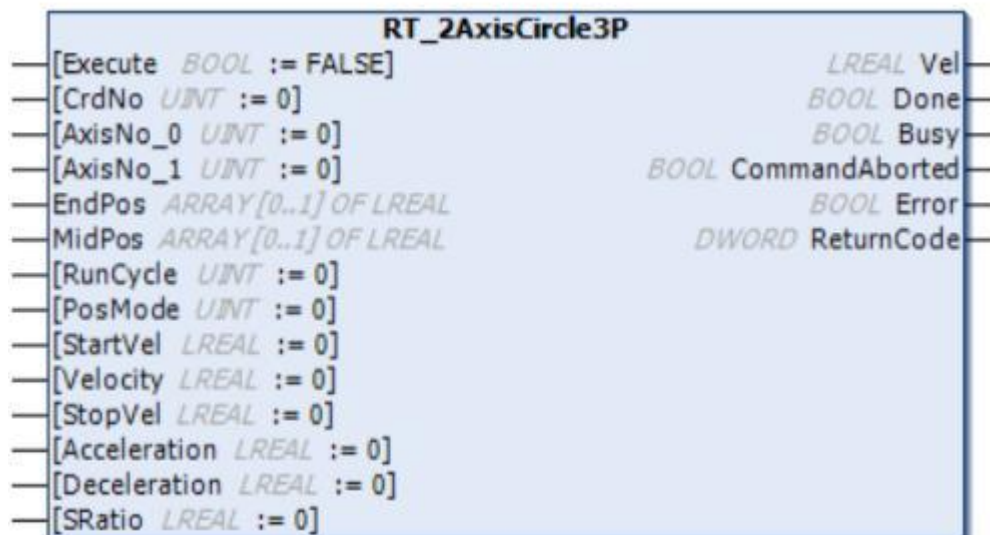
9.3.2.13. RT_2AxisCircle3P

FUNCTION_BLOCK RT_2AxisCircle3P

脉冲轴两轴圆弧插补（三点圆弧）功能块

Execute由False变为True时读取参数并执行一次

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
CrdNo	UINT			0	插补系数索引[0,3],不同的插补系可以独立执行插补运动
AxisNo_0	UINT			0	参与插补的0轴轴号，范围[0,7]
AxisNo_1	UINT			0	参与插补的1轴轴号，范围[0,7]
EndPos	ARRAY [0..1] OF LREAL				插补轴的目标位置(Pulse)
MidPos	ARRAY [0..1] OF LREAL				圆弧中点，由起点位置、中点位置、目标位置三点确定一段圆弧
RunCycle	UINT			0	圆弧旋转的圈数
PosMode	UINT			0	位置模式，0表示相对运动，1表示绝对运动
StartVel	LREAL			0	插补起始速度(Pulse/s)
Velocity	LREAL			0	插补最大速度(Pulse/s)
StopVel	LREAL			0	插补停止速度(Pulse/s)
Acceleration	LREAL			0	插补加速度(Pulse/s ²)
Deceleration	LREAL			0	插补减速度(Pulse/s ²)
SRatio	LREAL			0	插补加减速段的S段比例,范围[0,1],当取值为0时为梯形加减速曲线
Vel	LREAL				当前插补系运动的速度(Pulse/s)
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
CommandAborted	BOOL				如果该命令已被其他命令终止则为TRUE
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



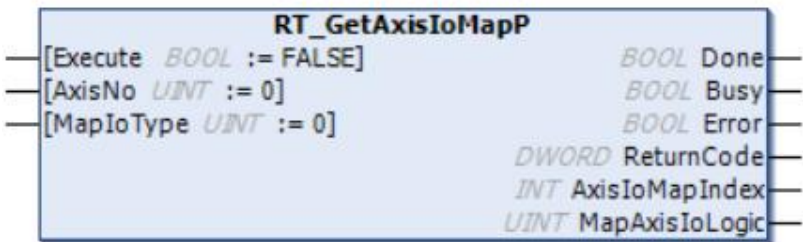
9.3.3. 脉冲轴参数设置功能块

9.3.3.1. RT_GetAxisIoMapP

FUNCTION_BLOCK RT_GetAxisIoMapP

该功能块用于获取脉冲轴的专用信号映射状态

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
MapIoType	UINT			0	IO类型，0——负限位，1——正限位
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值
AxisIoMapIndex	INT				轴专用信号映射索引，范围[0,15]对应通用输入信号Input0——Input15；-1则表示未被映射。
MapAxisIoLogic	UINT			0	轴专用信号逻辑，0——低电平有效；1——高电平有效

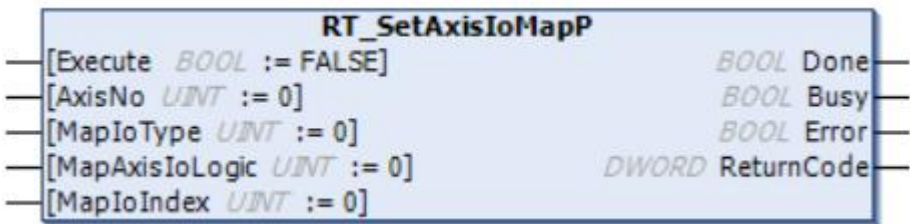


9.3.3.2. RT_SetAxisIoMapP

FUNCTION_BLOCK RT_SetAxisIoMapP

该功能块用于设置脉冲轴的专用信号映射
可以灵活配置通用输入信号作脉冲轴的限位信号使用

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
MapIoType	UINT			0	IO类型，0——负限位，1——正限位
MapAxisIoLogic	UINT			0	轴专用信号逻辑，0——低电平有效；1——高电平有效
MapIoIndex	UINT			0	映射索引，取值范围[0,15]对应通用输入信号Input0——Input15
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

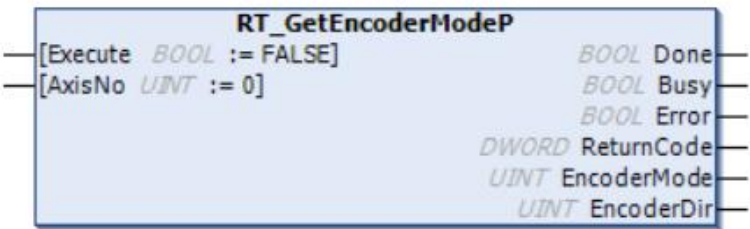


9.3.3.3. RT_GetEncoderModeP

FUNCTION_BLOCK RT_GetEncoderModeP

该功能块用于获取编码器计数器模式
编码器模式：
0--AB相四倍频计数
1--单相计数
编码器计数器方向：
0--正向计数
1--负向计数

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值
EncoderMode	UINT				编码器计数模式
EncoderDir	UINT				编码器计数方向

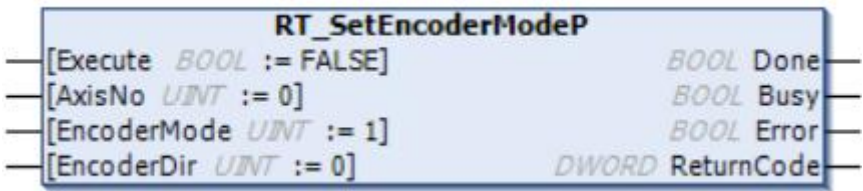


9.3.3.4. RT_SetEncoderModeP

FUNCTION_BLOCK RT_SetEncoderModeP

该功能块用于设置编码器计数器模式
编码器模式：
0--AB相四倍频计数
1--单相计数
编码器计数器方向：
0--正向计数
1--负向计数

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
EncoderMode	UINT			1	编码器计数模式，默认为1--单相计数
EncoderDir	UINT			0	编码器计数方向，默认为0--正向计数
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9.3.3.5. RT_GetIoConfigP

该功能块用于获取Output 口是否使能为脉冲输出功能

AxisOutput_0	对应输出口 Y0&Y1;为TRUE 则该输出口作轴 0 的脉冲输出用, 为FALSE 则该输出口作普通输出用
AxisOutput_1	对应输出口 Y2&Y3;为TRUE 则该输出口作轴 1 的脉冲输出用, 为FALSE 则该输出口作普通输出用
AxisOutput_2	对应输出口 Y4&Y5;为TRUE 则该输出口作轴 2 的脉冲输出用, 为FALSE 则该输出口作普通输出用
AxisOutput_3	对应输出口 Y6&Y7;为TRUE 则该输出口作轴 3 的脉冲输出用, 为FALSE 则该输出口作普通输出用
AxisOutput_4	对应输出口 Y10&Y11;为 TRUE 则该输出口作轴 4 的脉冲输出用, 为 FALSE 则该输出口作普通输出用
AxisOutput_5	对应输出口 Y12&Y13;为 TRUE 则该输出口作轴 5 的脉冲输出用, 为 FALSE 则该输出口作普通输出用

AxisOutput_6	对应输出口 Y14&Y15;为 TRUE 则该输出口作轴 6 的脉冲输出用, 为 FALSE 则该输出口作普通输出用
AxisOutput_7	对应输出口 Y16&Y17;为 TRUE 则该输出口作轴 7 的脉冲输出用, 为 FALSE 则该输出口作普通输出用

该功能块用于获取 Input 口是否使能为编码器计数功能

AxisInput_0	对应输入口 X0&X1;为TRUE则该输入口作轴 0 的编码器输入用, 为FALSE 则该输入口作普通输入用
AxisInput_1	对应输入口 X2&X3;为TRUE则该输入口作轴 1 的编码器输入用, 为FALSE 则该输入口作普通输入用
AxisInput_2	对应输入口 X4&X5;为TRUE则该输入口作轴 2 的编码器输入用, 为FALSE 则该输入口作普通输入用
AxisInput_3	对应输入口 X6&X7;为TRUE则该输入口作轴 3 的编码器输入用, 为FALSE 则该输入口作普通输入用
AxisInput_4	对应输入口 X10&X11;为TRUE则该输入口作轴4 的编码器输入用, 为FALSE 则该输入口作普通输入用
AxisInput_5	对应输入口 X12&X13;为TRUE则该输入口作轴5 的编码器输入用, 为FALSE 则该输入口作普通输入用

AxisInput_6	对应输入口X14&X15;为TRUE则该输入口作轴6 的编码器输入用，为FALSE 则该输入口作普通输入用
AxisInput_7	对应输入口X16&X17;为TRUE则该输入口作轴7 的编码器输入用，为FALSE 则该输入口作普通输入用



9. 3. 3. 6. RT_SetIoConfigP

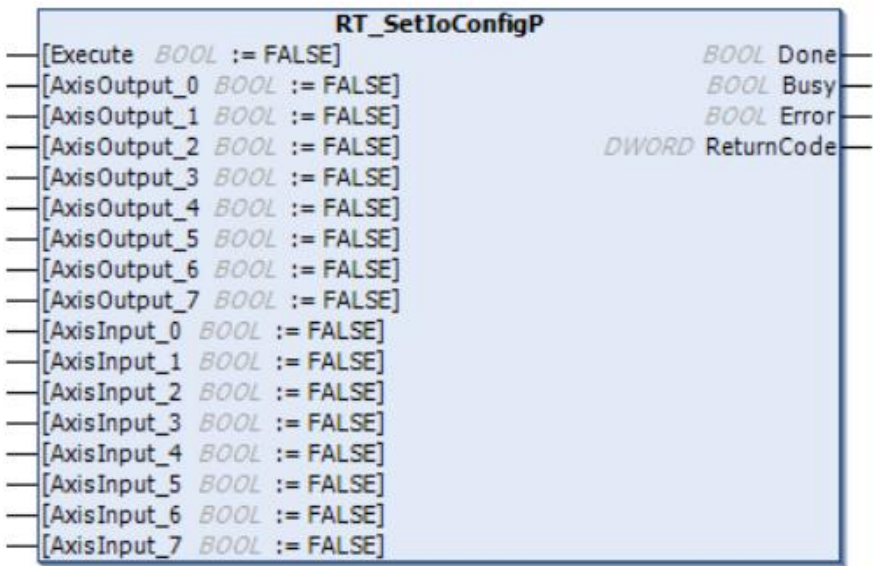
该功能块用于设置Output 口是否使能为脉冲输出功能

AxisOutput_0	对应输出口Y0&Y1;默认为FALSE该输出口作普通输出用，为TRUE时作轴0 的脉冲输出用
AxisOutput_1	对应输出口 Y2&Y3;默认为FALSE该输出口作普通输出用，为TRUE时作轴 1 的脉冲输出用
AxisOutput_2	对应输出口Y4&Y5;默认为FALSE该输出口作普通输出用，为TRUE时作轴2 的脉冲输出用
AxisOutput_3	对应输出口Y6&Y7;默认为FALSE该输出口作普通输出用，为TRUE时作轴3 的脉冲输出用
AxisOutput_4	对应输出口Y10&Y11;默认为FALSE该输出口作普通输出用，为TRUE时作轴 4 的脉冲输出用
AxisOutput_5	对应输出口Y12&Y13;默认为FALSE该输出口作普通输出用，为TRUE时作轴 5 的脉冲输出用
AxisOutput_6	对应输出口Y14&Y15;默认为FALSE该输出口作普通输出用，为TRUE时作轴 6 的脉冲输出用

AxisOutput_7	对应输出口Y16&Y17;默认为FALSE该输出口作普通输出用，为TRUE时作轴7的脉冲输出用
--------------	---

该功能块用于设置 Input 口是否使能为编码器计数功能

AxisInput_0	对应输入口X0&X1;默认为FALSE该输入口作普通输入用, 为TRUE时作轴0的编码器输入用
AxisInput_1	对应输入口 X2&X3;默认为FALSE该输入口作普通输入用, 为TRUE时作轴1的编码器输入用
AxisInput_2	对应输入口X4&X5;默认为FALSE该输入口作普通输入用, 为TRUE时作轴2的编码器输入用
AxisInput_3	对应输入口X6&X7;默认为FALSE该输入口作普通输入用, 为TRUE时作轴3的编码器输入用
AxisInput_4	对应输入口 X10&X11;默认为 FALSE 该输入口作普通输入用, 为 TRUE 时作轴4的编码器输入用
AxisInput_5	对应输入口 X12&X13;默认为 FALSE 该输入口作普通输入用, 为 TRUE 时作轴5的编码器输入用
AxisInput_6	对应输入口 X14&X15;默认为 FALSE 该输入口作普通输入用, 为 TRUE 时作轴6的编码器输入用
AxisInput_7	对应输入口 X16&X17;默认为 FALSE 该输入口作普通输入用, 为 TRUE 时作轴7的编码器输入用



9.3.3.7. RT_GetPulseModeP

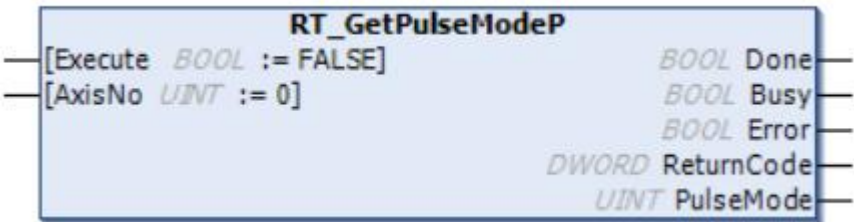
FUNCTION_BLOCK RT_GetPulseModeP

该功能块用于获取脉冲轴的脉冲模式

脉冲输出模式:

- 0--脉冲高+方向高
- 1--脉冲低+方向高
- 2--脉冲高+方向低
- 3--脉冲低+方向低
- 4--双脉冲高
- 5--双脉冲低
- 6--AB相

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值
PulseMode	UINT				脉冲轴的当前脉冲模式



9.3.3.8. RT_SetPulseModeP

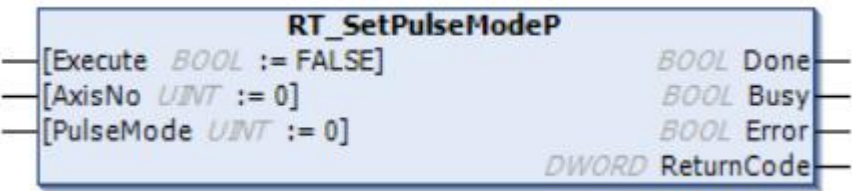
FUNCTION_BLOCK RT_SetPulseModeP

该功能块用于设置脉冲轴的脉冲模式

脉冲输出模式:

- 0--脉冲高+方向高
- 1--脉冲低+方向高
- 2--脉冲高+方向低
- 3--脉冲低+方向低
- 4--双脉冲高
- 5--双脉冲低
- 6--AB相

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
PulseMode	UINT			0	脉冲输出模式，默认为0--脉冲高+方向高
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值

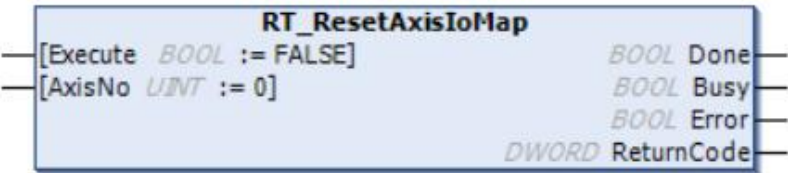


9.3.3.9. RT_ResetAxisIoMap

FUNCTION_BLOCK RT_ResetAxisIoMap

该功能块用于复位脉冲轴的专用信号映射，复位之后则轴的所有专用信号未配置且无效

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9.3.3.10. RT_SetPositionP

FUNCTION_BLOCK RT_SetPositionP

该功能块用于设置脉冲轴当前位置（编码器位置会被同步设置）

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
Position	LREAL			0	输入位置(Pulse)
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



9.3.3.11. RT_SetEncoderPositionP

FUNCTION_BLOCK RT_SetEncoderPositionP

该功能块用于设置编码器当前位置

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，从False变为True时读取参数并执行
AxisNo	UINT			0	轴号，范围[0,7]
Position	LREAL			0	输入位置(Pulse)
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值



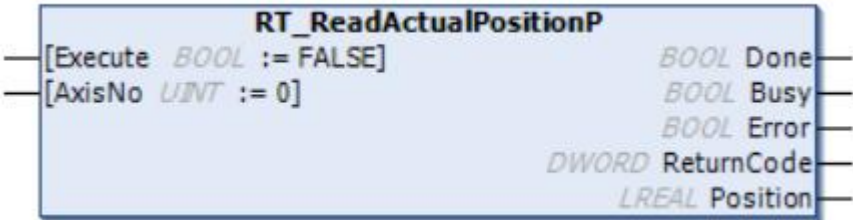
9.3.4. 脉冲轴信息获取功能块

9.3.4.1. RT_ReadActualPositionP

FUNCTION_BLOCK RT_ReadActualPositionP

该功能块用于读取脉冲轴当前实际位置值

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，为TRUE时将持续读取轴的实际位置
AxisNo	UINT			0	轴号，范围[0,7]
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值
Position	LREAL				轴当前实际位置(Pulse)

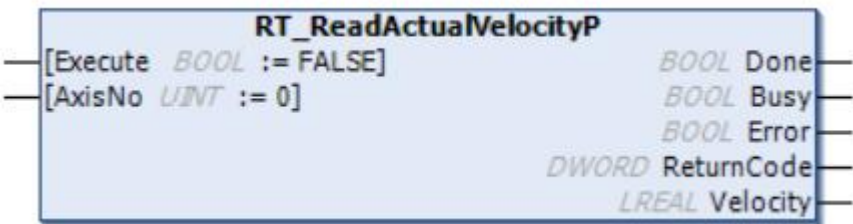


9.3.4.2. RT_ReadActualVelocityP

FUNCTION_BLOCK RT_ReadActualVelocityP

该功能块用于读取脉冲轴的实际速度值

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，为TRUE时将持续读取轴的当前速度
AxisNo	UINT			0	轴号，范围[0,7]
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值
Velocity	LREAL				轴当前实际速度(Pulse/s)

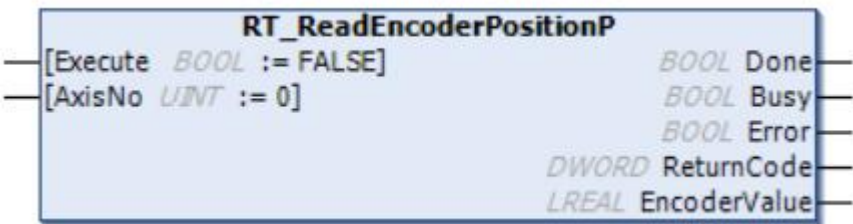


9.3.4.3. RT_ReadEncoderPositionP

FUNCTION_BLOCK RT_ReadEncoderPositionP

该功能块用于读取编码器计数功能

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行，为TRUE时将持续读取编码器计数值
AxisNo	UINT			0	轴号，范围[0,7]
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值
EncoderValue	LREAL				当前编码器的计数值(Pulse)



9.3.4.4. RT_ReadStatusP

FUNCTION_BLOCK RT_ReadStatusP

该功能块用于读取脉冲轴的状态
轴的回零状态(HomeStatus):0——回零未完成
1——回零完成, 找到原点位置
2——回零过程报错, 没有原点信号
3——回零过程发现设备有两个原点信号
4——回零过程没有发现EZ信号
5——回零配置的方向错误
6——回零过程限位停止
7——超出软件限位范围
100——没有正常回零完成

名称	类型	继承自	地址	初始化	注释
Execute	BOOL			FALSE	是否执行, 为TRUE时将持续读取轴的当前状态
AxisNo	UINT			0	轴号, 范围[0,7]
Done	BOOL				功能块是否执行完成
Busy	BOOL				功能块是否运行中
Error	BOOL				功能块是否发生错误
ReturnCode	DWORD				功能块返回值
Enable	BOOL				如果轴状态为Enable, 则为TRUE
Running	BOOL				如果轴处于运动中, 则为TRUE
HomeStatus	UINT				轴的回零状态



10. 附录

10.1. 电工中的 3 种接地是什么意思？

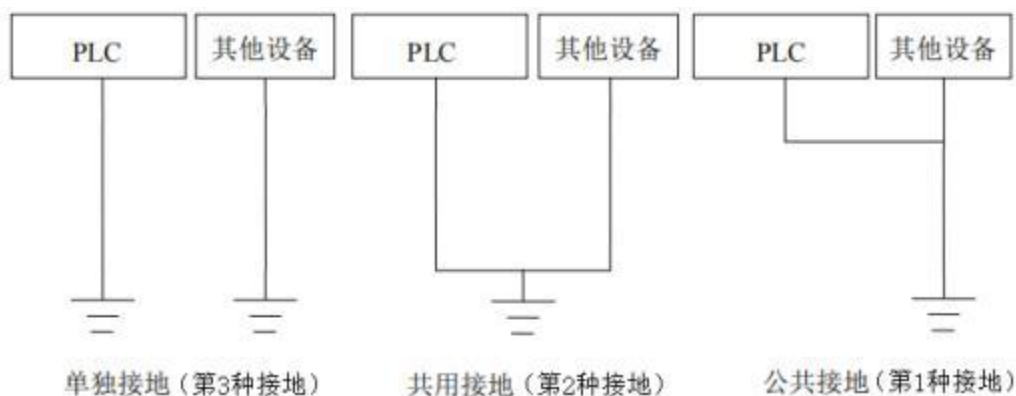
低压电气设备的单点接地方式可分为：串联式单点接地、并联式单点接地、多分支单点接地。

串联式单点接地：也就是第 1 种接地方式。接地方法：将多个低压电气设备的接地端子在设备的就近处与同一根接地线连上，然后通过这根接地线与接地装置连接。这种接地的好处在于：节省人力、物力；而坏处在于：当公用的接地线出现断路时，如果接地系统中有一台设备漏电，就会引起其它设备的外壳上均出现电压，对人员安全造成威胁。

并联式单点接地：也就是第 2 种接地方式。接地方法：设备的接地端子都引出一根接地线，然后将这若干条线同时接到接地装置上。这种接地方式的好处在于：当接地系统中的其中一台接地设备出现断路时，不会造成其它设备的外壳出现电压，对保障人身安全有好处。而这种接地方式的不完美之处在于：如果是电子设备或其它对高频干扰高度敏感的电气设备，来自于其它设备的高频干扰（例如变频器、中频炉等晶闸管变流器件）将会从共地点串入，造成设备工作不正常。

多点分支接地：也就是第 3 种接地方式。接地方法：将每个设备的接地端子单独接到接地装置上。接地方法和第 2 种接地的区别在于：设备有单独的接地体（或者变通一下：直接接到离接地体最近的接地装置上（或者接地源处），每个设备在电气接地回路上的距离是比较远的（例如超过 50 米）），这有效避免了设备之间的相互电磁干扰。但这种方式费时、费力而且单独接地源不一定好取。

在平常施工中，如果有条件，建议使用第 3 种接地方式。但是，实际上 PLC 的接地方式采用第 2 种接地方式的比较多，至于电磁干扰方面：如果柜内有多个大功率的变频器，可以在 PLC 电源的前端加装一个单相电源滤波器。



10.2. 如何查看和修改EC系列PLC 的 IP 地址

1、RM 系列 PLC 的出厂默认IP 地址是 192.168.1.3，子网掩码 255.255.255.0，通过 PLC 指令 的 `ipv4-get` 指令，就可以获取如下图所示 PLC 设备的 IP 地址



2、如果该改变 PLC 设备的 IP 地址，通过 PLC 指令的 `ipv4-set` 指令，就可以一个新的 IP 地址，如下图所示：`ipv4-set 192.168.1.4`



3、如果不知道 PLC 的 IP 地址，搜索不到 EC 系列 PLC 了怎么办？

EC 系列 PLC 有一个不可修改的备用 IP 地址 10.0.0.10，子网掩码 255.0.0.0，如果忘记了 PLC 的 IP 地址，您可以把电脑的 IP 地址修改成 IP 地址 10.0.0.11，子网掩码 255.0.0.0，则可以通过 CODESYS 搜索到 PLC；搜索到 PLC 以后请立即 设定自己的 IP 地址。

联系我们 SCATECH