

OptiBuilder 软件使用手册

声 明

- 本说明书版权属于浙江源创智控技术有限公司，事先未经明确的书面许可不得复制或修改本说明书中的任何内容，否则将追究相关法律责任。
- 本说明书中如有难以理解的地方或错误、遗漏等问题，敬请联系我们。

电话：0571-86667887

统一热线：400-0571-669

网址：www.sctechsmart.com

商 标

源创智控、Yuntrol、OptiSys、OptiBuilder、OptiVIEW 以上商标或标识均是浙江源创智控技术有限公司已经注册或已经申请注册或正在使用的商标和标识，拥有以上商标的所有权，未经浙江源创智控技术有限公司的书面授权，任何个人及企业不得擅自使用上述商标对于非法使用我司商标的行为，我司将保留依法追究行为人及企业的法律责任的权利。

1 术语和缩写	6
2 概述	6
2.1 主要功能	7
2.2 性能指标	8
2.3 结构概述	9
3 控制站组态	9
3.1 启动软件	9
3.1.1 窗口移动	10
3.1.2 窗口复位	11
3.2 新建工程	11
3.3 硬件组态及位号组态	14
3.3.1 开放通讯服务	14
3.3.2 Modbus 服务	15
3.3.3 站间通讯	17
3.3.4 BACnet/IP 服务	21
3.3.5 以太网总线	24
3.3.6 串行总线	29
3.3.7 扩展功能	39
3.4 基本组态	43
3.5 本地 IO	45
3.6 变量管理	46
3.6.1 硬件变量	46
3.6.2 自定义变量	48
3.6.3 变量分组	50
3.7 自定义数据类型和自定义功能块	52
3.7.1 自定义数据类型	52

3.7.2 自定义功能块	56
3.7.3 分组功能	59
3.7.4 复用配置	59
3.8 函数	62
3.9 任务配置（控制方案组态）	64
3.9.1 添加任务	64
3.9.2 添加程序	67
3.9.3 周期执行任务和中断执行任务触发条件设置	68
3.9.4 调整程序执行顺序	70
3.10 查找和替换	70
3.11 变量声明	73
3.12 布局排布	74
3.12.1 对齐	74
3.12.2 重排	75
3.13 联机	75
3.13.1 编译	75
3.13.2 选择联机设备	76
3.13.3 联机选项	77
3.13.4 联机调试	78
3.13.5 多工程调试	81
3.13.6 注意事项	82
4 设备管理	83
4.1 IP 扫描功能	83
4.2 MAC 扫描	85
4.3 时间设置	85
4.3.1 设备时间设置	85

4.3.2 时钟同步设置	86
4.4 冗余切换	87
5 实现监控	87
5.1 通过 OptiVIEW 实现监控	87
5.2 通过其他监控软件实现监控	87
5.2.1 使用 ModbusRtu 驱动	88
5.2.2 使用 ModbusTcp 驱动	89
6 编程指南	90
6.1 FBD 语言编程	90
6.1.1 FBD 编辑器主要命令一览表	91
6.1.2 功能块	91
6.1.3 自定义功能块的引脚定义	92
6.1.4 程序的私有变量	93
6.1.5 链接	93
6.1.6 执行次序	93
6.1.7 编程原则	93
6.2 ST 语言编程	94
6.2.1 ST 语言语法	94
6.2.2 编程助手	100
6.2.3 ST 可调用函数	100
6.3 LD 语言编程	101
6.3.1 LD 编辑器主要命令一览表	101
6.3.2 触点	102
6.3.3 线圈	105
6.3.4 执行次序	109
6.3.5 编程原则	109

6.4 自由 LD 语言编程	109
6.4.1 自由 LD 编辑器主要命令以及元素说明	109
6.4.2 快捷操作	109
6.4.3 执行次序	111
6.4.4 编程原则	111
6.5 载入示例	111
7 资料版本说明	113

OptiBuilder 软件

1 术语和缩写

- RCP (Routable Communication Protocol) : 路由通信协议。
- POU (Programming Organisation Unit) : 程序组织单元。
- SNTP (Simple Network Time Protocol) : 简单网络时间协议。
- RTC (Real-Time Clock) : 实时时钟。

2 概述

OptiBuilder 2.0 组态编程软件具有直观的图形化界面、集成的开发环境、多种编程语言支持、通信协议支持、强大的调试工具等特点,可以帮助用户快速开发和调试 PLC 程序,提高工作效率和程序质量。系统结构示意图如下图所示。

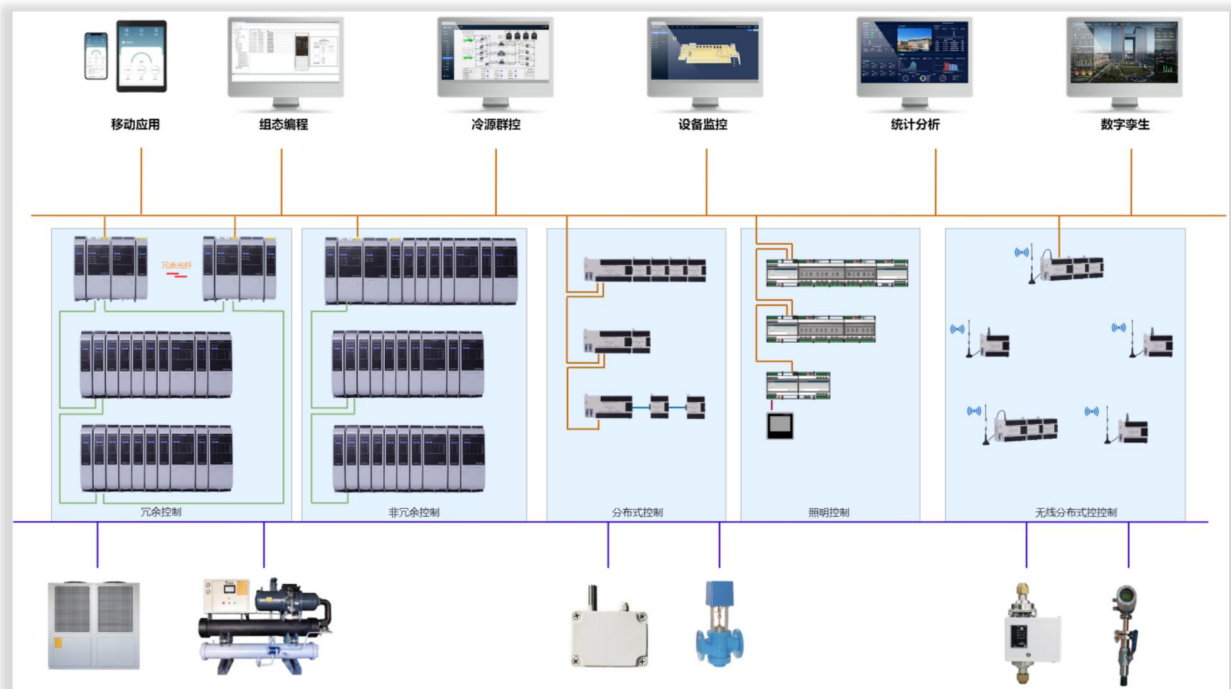


图 2-1 系统结构示意图

2.1 主要功能

◆ 硬件组态功能

- 支持系统结构的配置
- 支持设备参数的配置
- 支持硬件位号的配置
- 支持异构通信设备（MODBUS RTU/TCP/自由串口）的配置

◆ 数据类型

- 基本数据类型，支持
BOOL/SINT/INT/DINT/LINT/USINT/UINT/UDINT/ULINT/REAL/LREAL/BYTE/WORD/DWORD/LWORD 等
- 复杂数据类型，支持自定义数组和自定义结构（自定义结构允许嵌套，不允许递归）

◆ 位号组态功能

- 自动生成硬件变量
- 支持全局变量的配置
- ◆ 多任务的控制策略组态功能
 - 支持 IEC61131-3 标准功能块
 - 支持 ST、LD 和 FBD 编写的自定义功能块组态
 - 支持 ST、LD 和 FBD 编写的用户程序组态
 - 支持用户任务的组态
- ◆ 工程管理功能
 - 支持工程组态的配置和编译
 - 支持组态的离线下载和在线下载
 - 支持组态上载
 - 支持设备在线诊断
 - 支持变量调试，支持硬点强制
 - 支持程序诊断
- ◆ 设备管理功能
 - 支持扫描在线设备
 - 支持对设备进行详细诊断
 - 支持网络配置
 - 支持对控制器时间的设置
 - 支持搜索发现本地网卡连接的设备
 - 支持固件升级

2.2 性能指标

- ◆ 采用多任务，64 位程序
- ◆ 良好的开放性能

- ◆ 系统组态结构清晰，界面操作方便
- ◆ 控制算法组态采用国际标准，实现图形组态与语言组态结合，功能强大
- ◆ 系统安全、可靠，长期运行稳定
- ◆ 支持 ModbusTcp/Rtu
- ◆ 支持自定义串口协议
- ◆ 支持在线下载功能
- ◆ 支持组态上载

2.3 结构概述

用户在安装 OptiBiulder 软件后，电脑桌面上只有一个 OptiBuilder2.0 图标。该软件中包含硬件组态、设备管理、用户程序编译及仿真功能。

3 控制站组态

3.1 启动软件

控制站组态软件实现对单个控制站进行硬件组态、变量管理、任务配置等操作，本章节主要介绍控制站组态软件（OptiBuilder2.0）的使用说明。

双击  “OptiBuilder2.0”，进入界面，如下图所示。

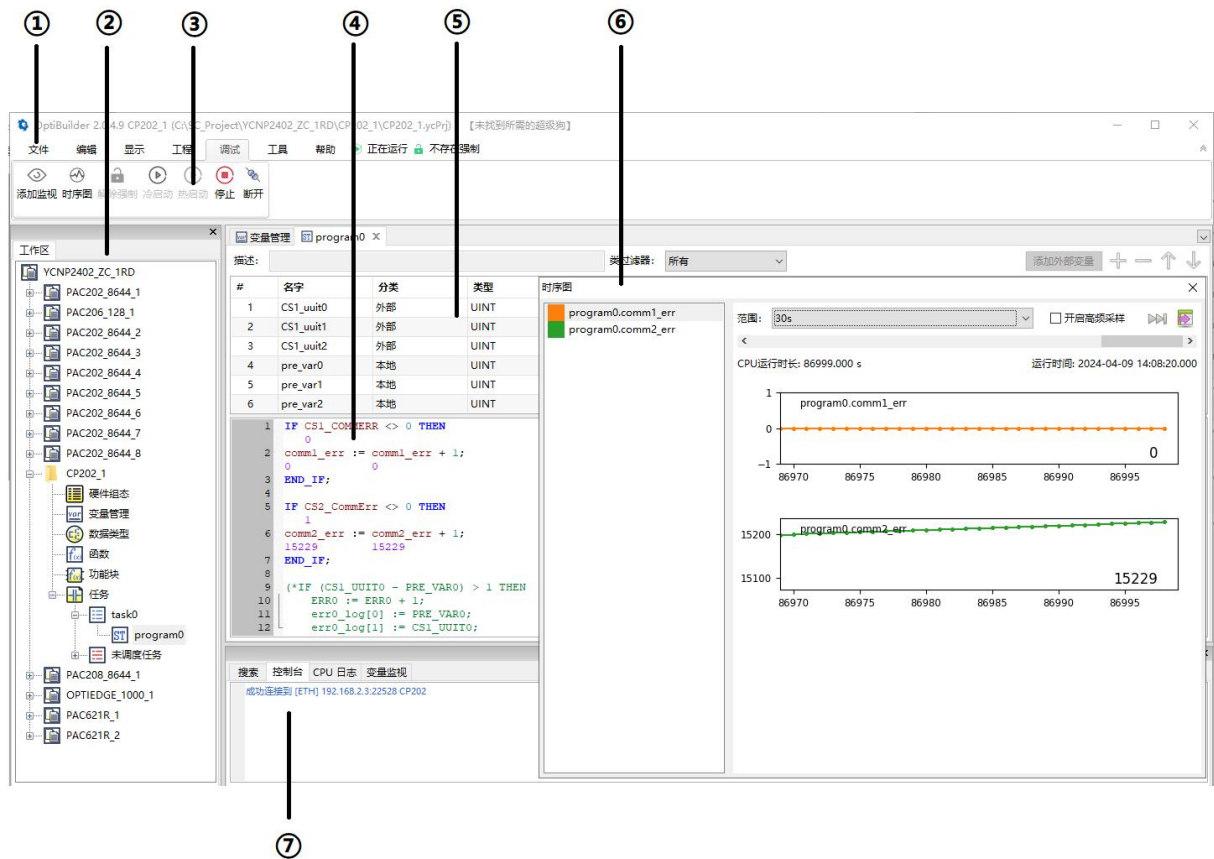


图 3- 1 OptiBuilder2.0 软件界面

- ① 菜单栏
- ② 工程项目树
- ③ 工具栏
- ④ 程序编辑器
- ⑤ 程序变量信息表
- ⑥ 变量监视时序图
- ⑦ 输出窗口

3. 1. 1 窗口移动

要移动窗口，请单击窗口框架，然后开始拖动。拖动时会以阴影指示将其拖放到该位置时该窗口的位置。



图 3- 2 拖动阴影

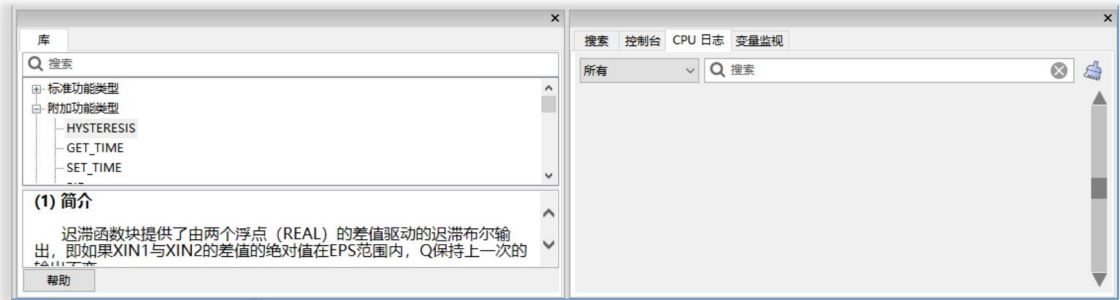


图 3- 3 拖动释放吸附

3. 1. 2 窗口复位

通过复位布局按钮一键复位组态软件窗口布局



图 3- 4 复位布局

3. 2 新建工程

点击工具栏上的“文件” -> “新建按钮” 弹出图 3- 5 所示的新建工程对话框

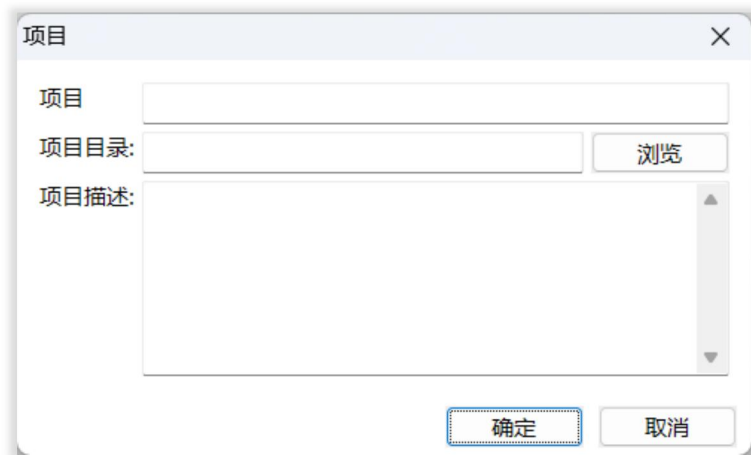


图 3- 5 新建工程

- 项目：新建项目的项目名。项目名称长度不超过 32 个字节。
- 项目目录：新建项目的存放路径。
- 项目描述：为新建的项目添加描述。项目描述长度不超过 100 个字节。

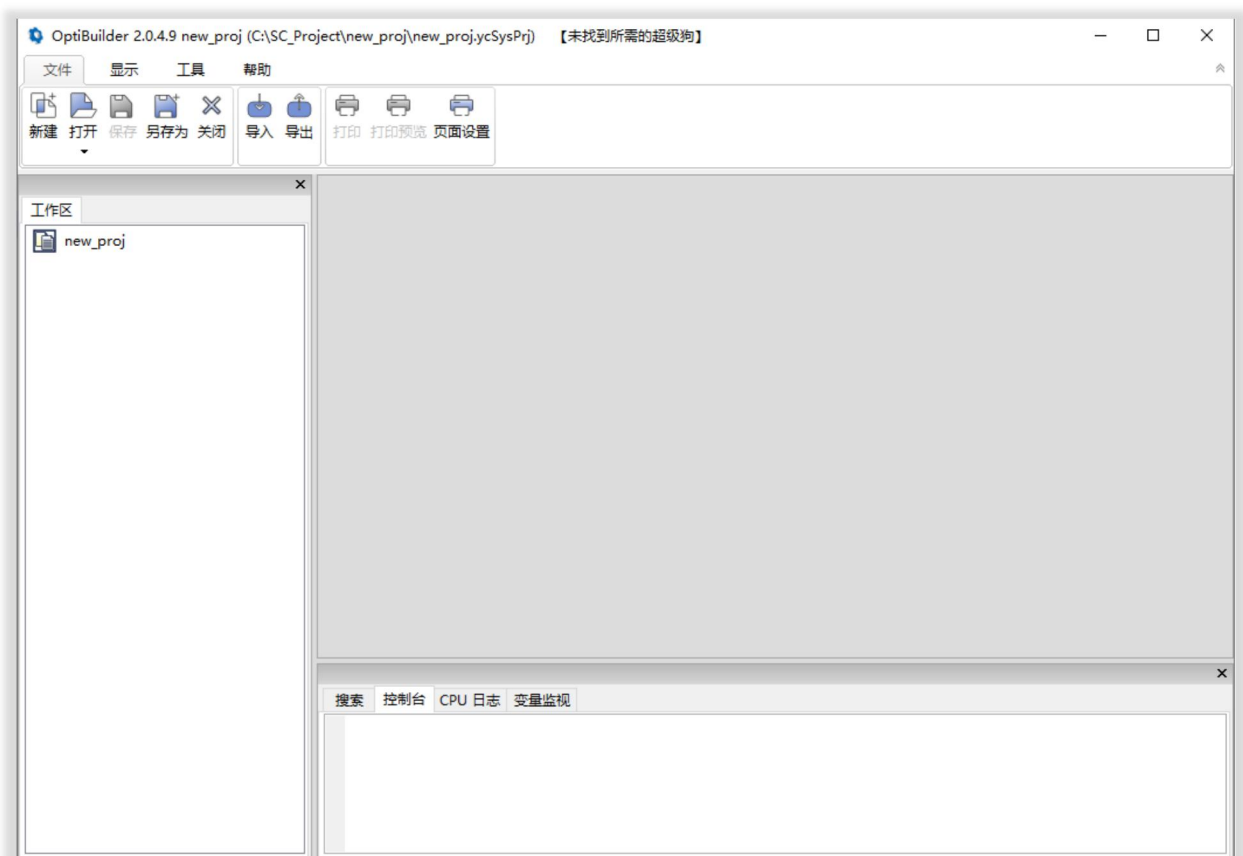


图 3- 6 新建项目完成

在新建的项目 “T_MOTER_FORWARD_REVERSE” 上鼠标 “右击” ->添加工程

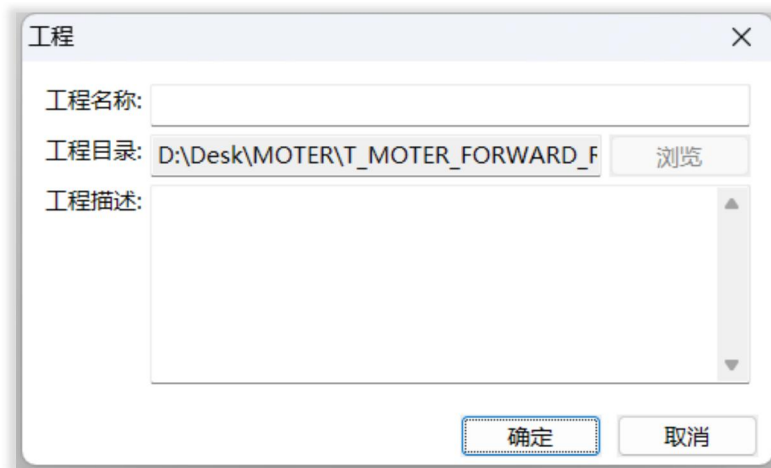


图 3- 7 新建工程

- 工程名称：新建工程的工程名。项目名称长度不超过 32 个字节。
- 工程目录：新建工程的存放路径。。
- 工程描述：为新建的工程添加描述。项目描述长度不超过 100 个字节

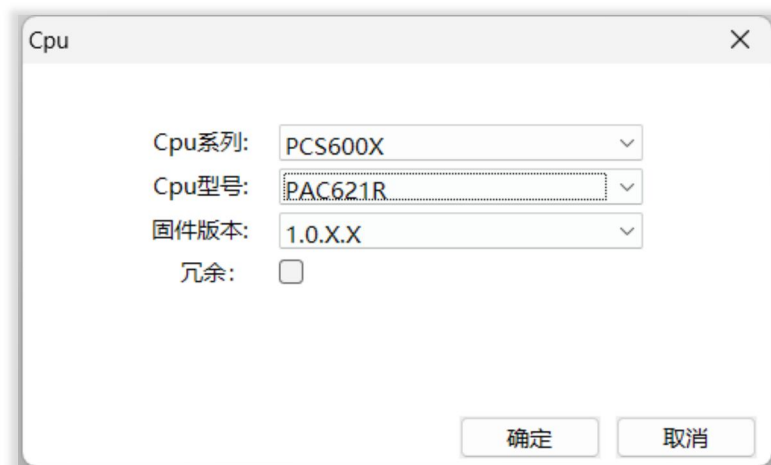


图 3- 8 选择控制器

- Cpu 系列：PCS-200、PCS-600X、LCS-300、Gateway
- Cpu 型号：选择具体 Cpu 系列中的 Cpu
- 固件版本：选择具体 Cpu 的固件版本
- 冗余： 选择当前工程对应的控制器系统是否需要冗余。

提示:

- 一个工程下可以创建多个工程。工程上限为 32 个
- 右击创建的工程“配置”重新配置工程名称、工程目录、工程描述
- 右击创建的工程“在文件资源管理器中打开项目”，可以直接打开创建的工程
- 右击创建的工程“复制”，可以复制一个当前选中的工程
- 右击创建的工程“删除”，可以删除一个当前选中的工程

新建完工程以后，可以在工程下完成硬件组态、变量管理、数据类型、函数、功能块、任务操作

3.3 硬件组态及位号组态

本节详细介绍硬件组态、位号组态等组态方法。

3.3.1 开放通讯服务

在工程目录下鼠标双击“硬件组态”，弹出如下图所示的硬件组态软件界面。界面右侧包含了位号组态软件显示区。

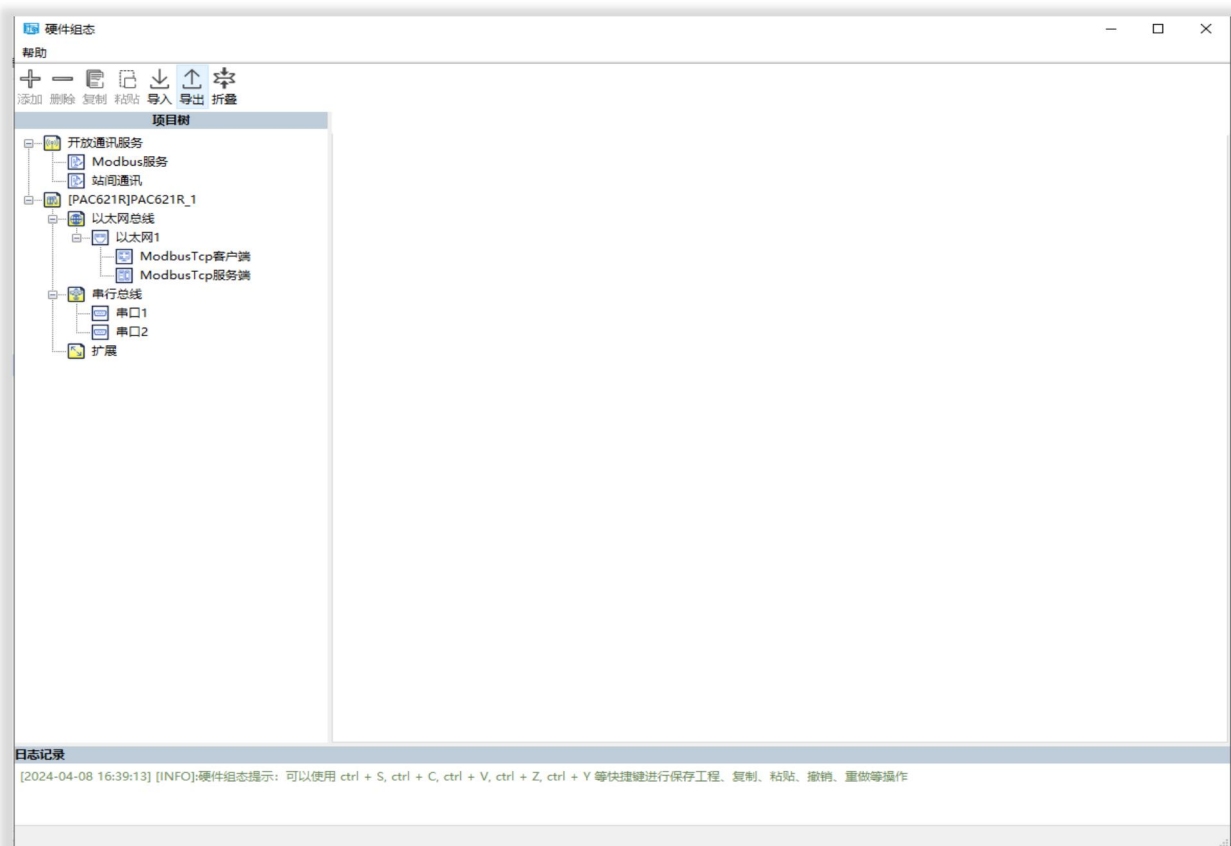


图 3- 9 硬件组态界面

3.3.2 Modbus 服务

此 Modbus 服务是指 PLC 控制器开放自身的变量位号数据给第三方设备进行读写使用，在 Modbus RTU 从站和 Modbus TCP 服务端上都能使用。Modbus 服务中的点位表可以通过点击“导出位号表”导出成“Modbus 服务位号表.csv”，方便第三方设备点位配置。

使用方法：点击 Modbus 服务，在右侧窗口空白处右键鼠标选择要开放的位号数据。

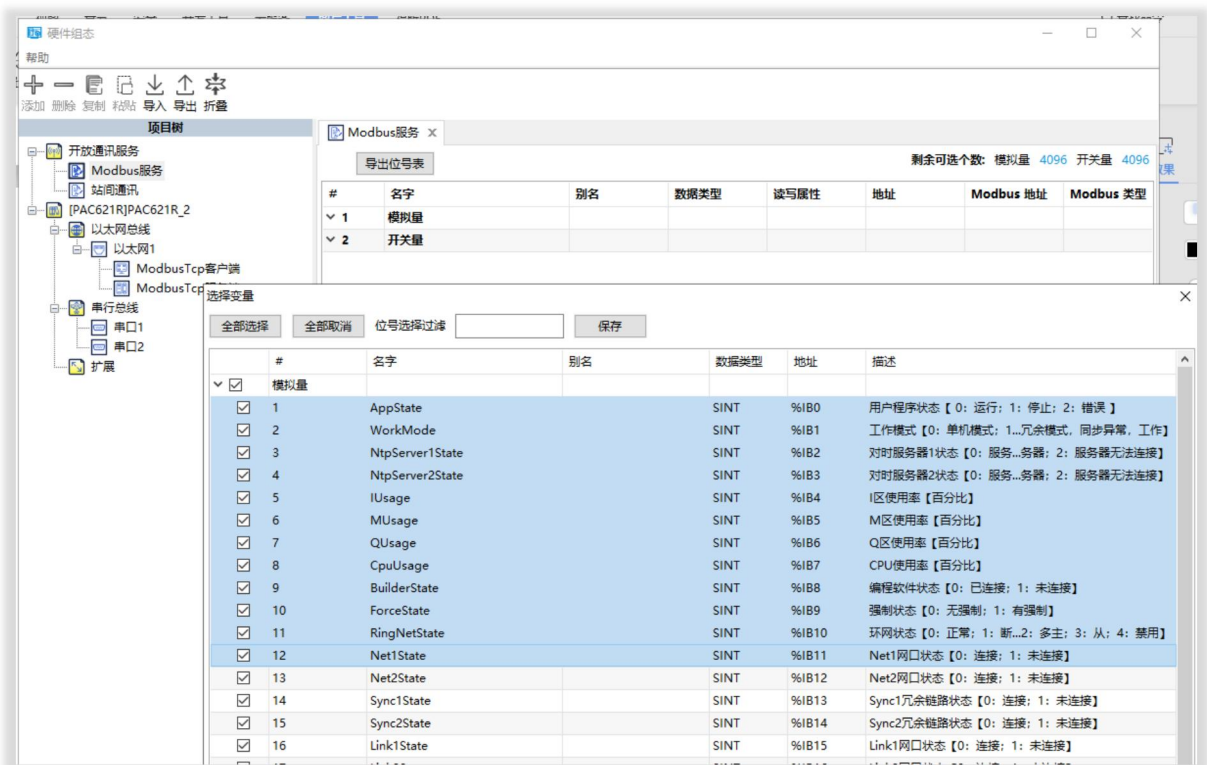


图 3- 10 Modbus 服务选择变量界面

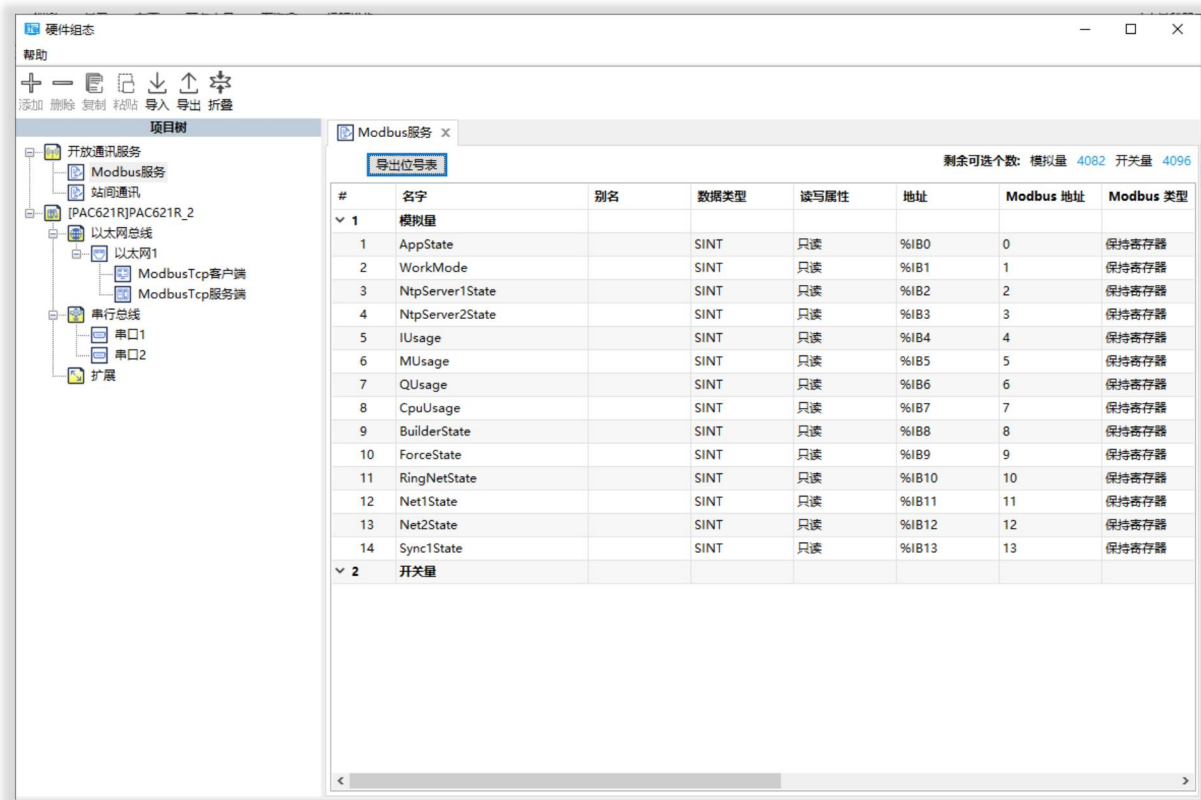


图 3- 11 Modbus 服务选择完成界面

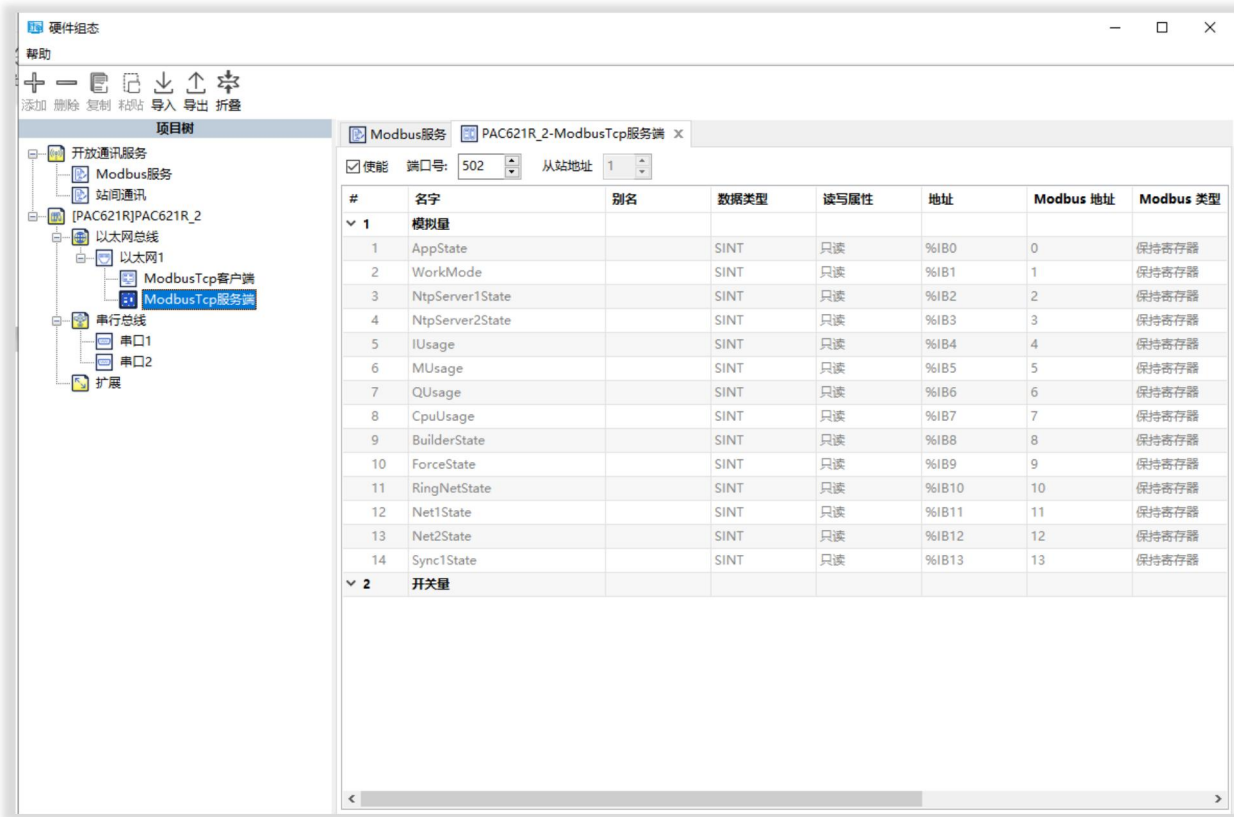


图 3- 12 ModbusTcp 服务端界面

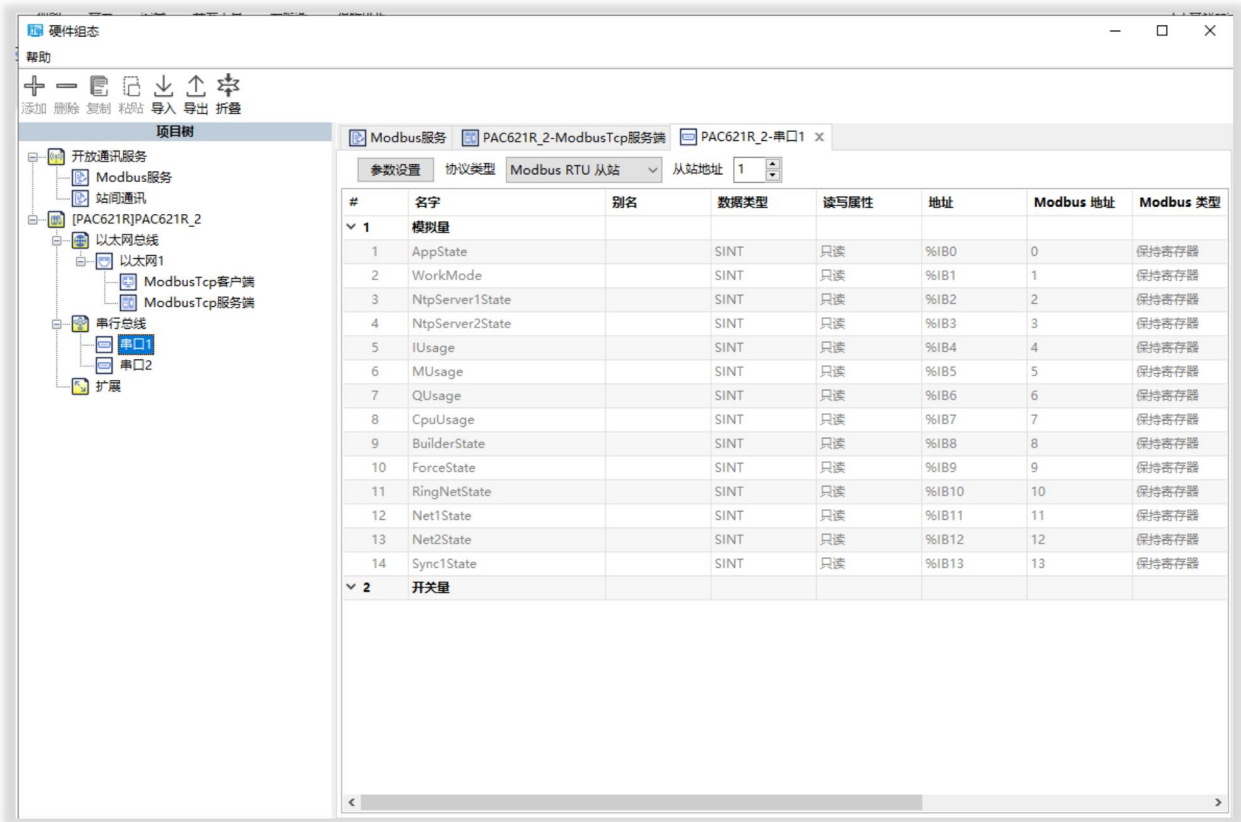


图 3- 13 ModbusRtu 从站界面

3. 3. 3 站间通讯

站间通讯可以提供 PLC 控制器之间的数据通讯服务，支持广播发送和点对点发送方式，广播发送是指一个 PLC 发送的数据在同一局域网内的 PLC 都能接收到，点对点发送是指 PLC 控制器往指定的 PLC 控制器进行发送。

◆ 发送配置

- **广播发送方式：**选择发送方式为广播模式，右键发送窗口空白处选择要发送的位号数据。

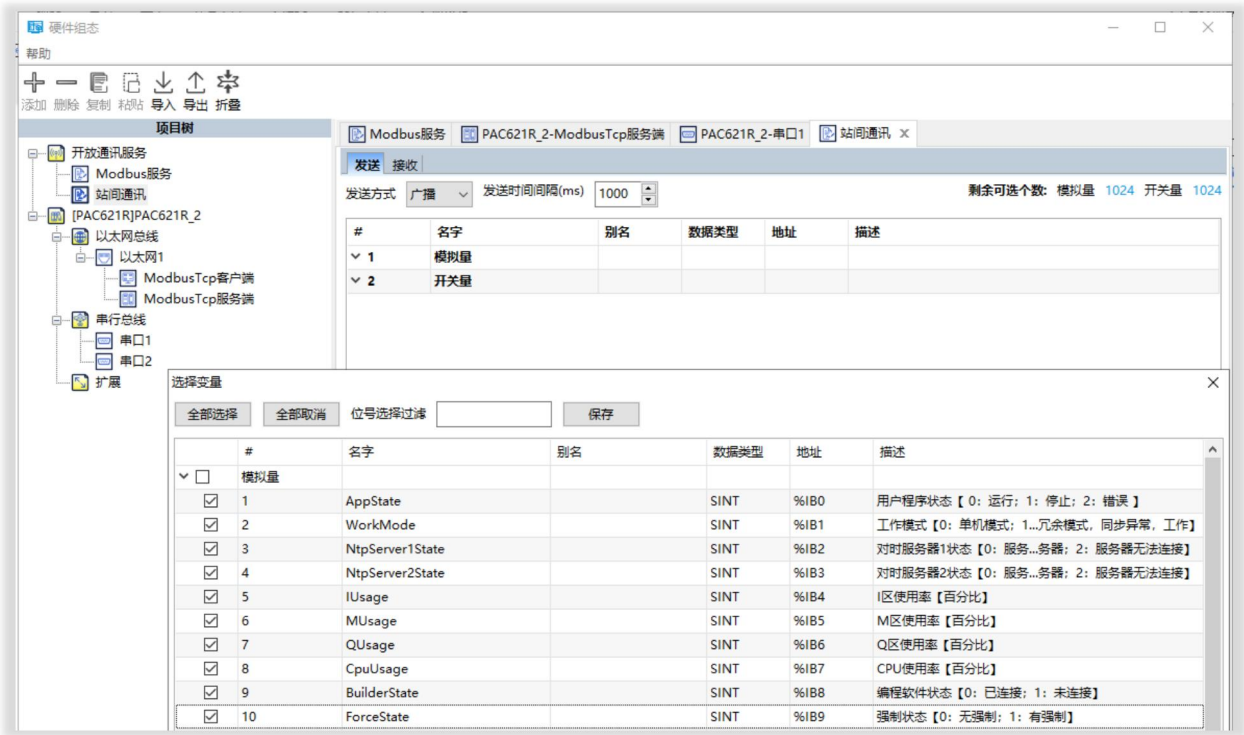


图 3- 14 站间通讯选择变量界面

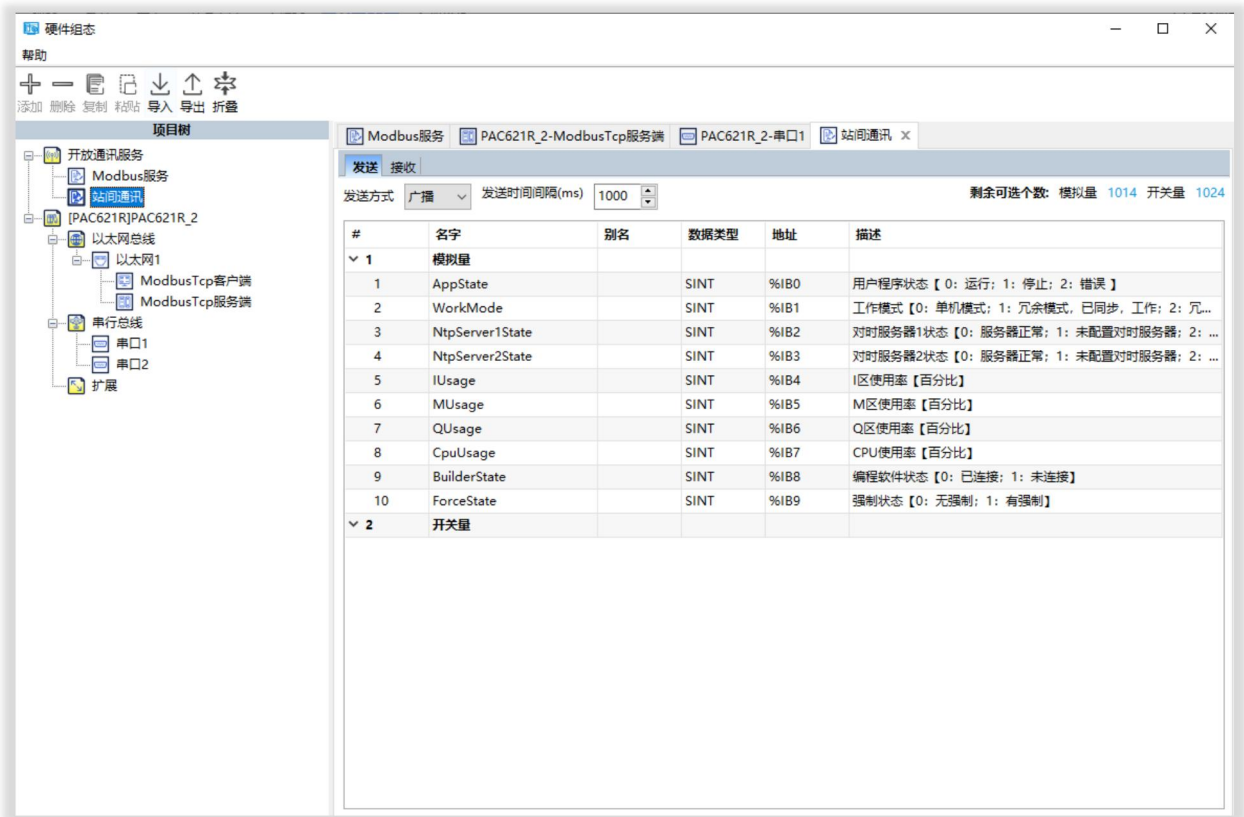


图 3- 15 站间通讯发送变量界面

- **点对点发送方式：**选择发送方式为点对点模式，填入要指定 PLC 控制器的 IP 地址。

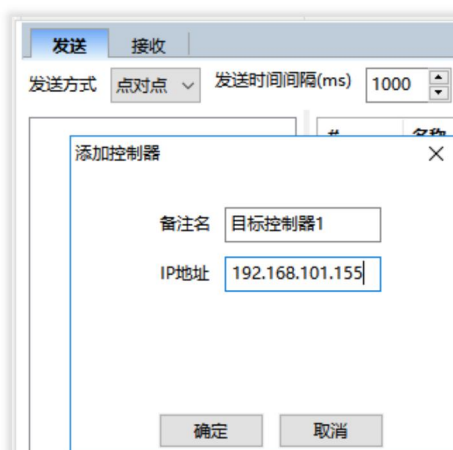


图 3- 16 点对点发送配置界面

右键发送窗口空白处选择要发送的位号数据。



图 3- 17 点对点发送界面

◆ 接收配置

打开待接收 PLC 控制器的硬件组态，选择接收窗口，在接收下方空白处右键鼠标，选择要接收的控制器，并手动填入对方控制器的 IP 地址和时间间隔，时间间隔表示接收控制器接收的超时时间基准配置，超时位号报警时间是此基准上再乘以 3，比如配置 1000ms，则接收超时时间 3000ms。

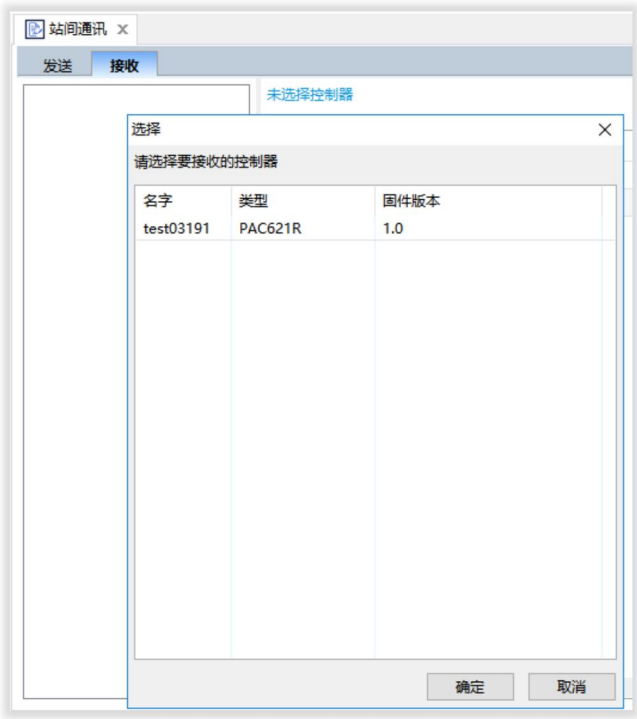


图 3- 18 选择接收控制器界面

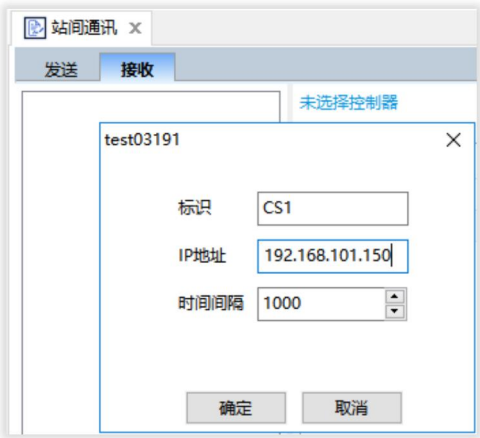


图 3- 19 配置接收控制器界面

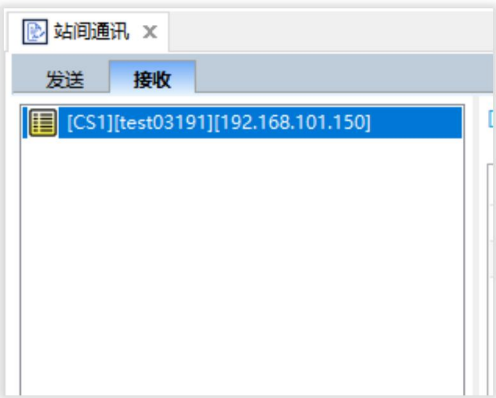


图 3- 18 配置完成界面

接收位号窗口右键空白处选择要接收的位号数据。



图 3- 19 选择接收位号界面

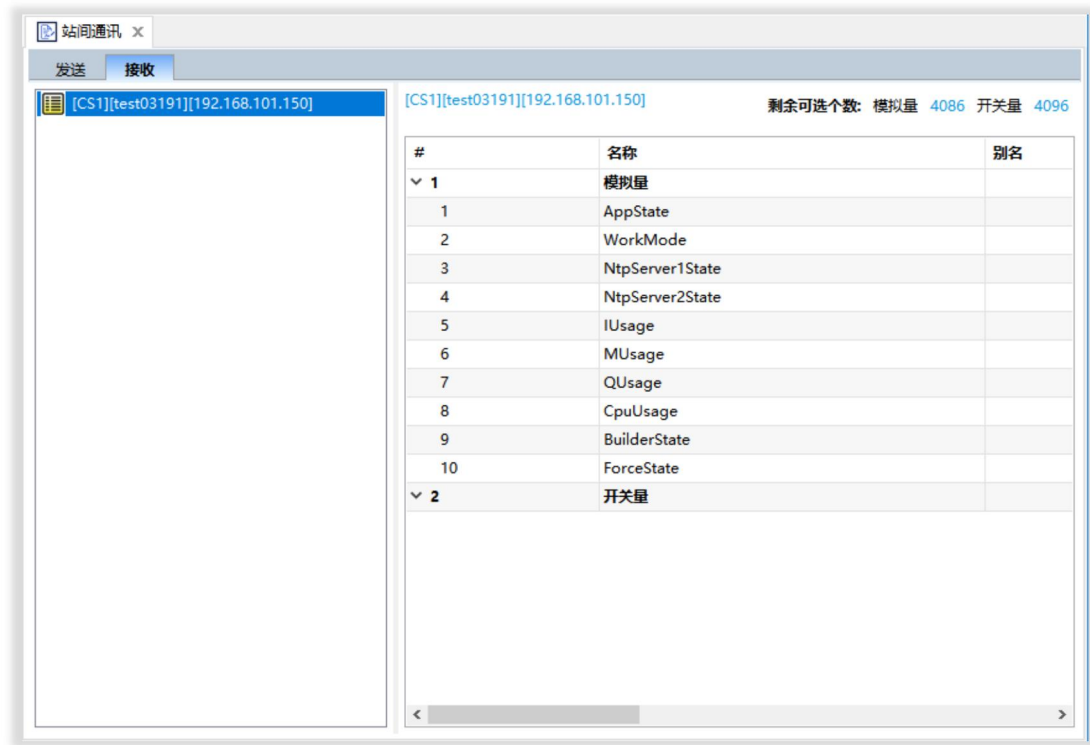


图 3- 20 选择接收变量完成界面

3. 3. 4 BACnet/IP 服务

BACnet/IP 服务可以配置相关控制器，使其具备 BACnet 标准通讯协议，用以向 BACnet 客户端提供数据服务。支持自定义端口号配置，默认配置端口号为 47808，可配置范围[1024~65535]，可将

控制器自带的数字量位号和模拟量位号通过 OptiBuilder 组态软件配置成 BI、BO、BV、AI、AO、AV、MSV、Date Value、Time Value、Date Time Value、CharacterString Value 十一种数据类型，方便客户端读取，也支持 BACnet/IP 服务的位号表导出，方便查看位号属性，以及支持导出的位号表修改寄存器地址后重新导入生成新的位号表。

◆ 选择变量

在 BACnet/IP 服务窗口栏空白处中右键选择变量，可自行选择需要添加到 BACnet/IP 服务的位号，支持控制器所有硬件变量和自定义变量的添加。

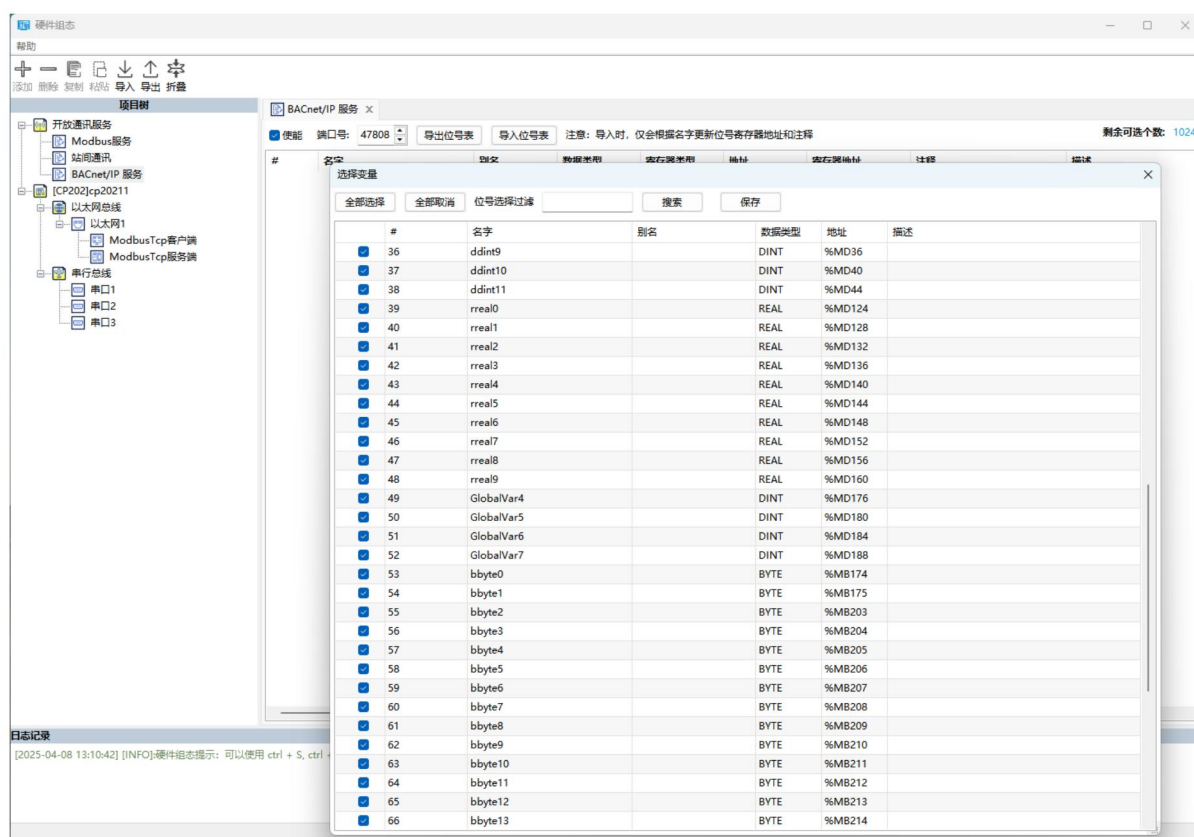


图 3- 21 BACnet/IP 服务选择变量界面

变量选择完成之后，组态软件会按照变量类型自动进行 BACnet 寄存器类型的匹配，将各变量按照匹配的寄存器类型进行配置。

BACnet/IP 服务 x

☒ 使能

端口号: 47808

导出位号表

导入位号表

注意: 导入时, 仅会根据名字更新位号寄存器地址和注释

剩余可选个数: 942

#	名字	别名	数据类型	寄存器类型	地址	寄存器地址	注释	描述
1	bbaa0		BOOL	BV	%MX164	0		
2	bbaa1		BOOL	BV	%MX165	1		
3	bbaa2		BOOL	BV	%MX166	2		
4	bbaa3		BOOL	BV	%MX167	3		
5	bbaa4		BOOL	BV	%MX168	4		
6	bbaa5		BOOL	BV	%MX169	5		
7	bbaa6		BOOL	BV	%MX170	6		
8	bbaa7		BOOL	BV	%MX171	7		
9	bbaa8		BOOL	BV	%MX172	8		
10	bbaa9		BOOL	BV	%MX173	9		
11	AppState		SINT	AI	%IB0	0		用户程序状态【0: 运行;
12	NtpServer1State		SINT	AI	%IB1	1		对时服务器1状态【0: 服
13	NtpServer2State		SINT	AI	%IB2	2		对时服务器2状态【0: 服
14	IUsage		SINT	AI	%IB3	3		I区使用率【百分比】
15	MUsage		SINT	AI	%IB4	4		M区使用率【百分比】
16	QUsage		SINT	AI	%IB5	5		Q区使用率【百分比】
17	CpuUsage		SINT	AI	%IB6	6		CPU使用率【百分比】
18	BuilderState		SINT	AI	%IB7	7		编程软件状态【0: 已连接
19	ForceState		SINT	AI	%IB8	8		强制状态【0: 无强制; 1:
20	yyy_CommErr		SINT	AI	%IB9	9		通信状态【0: 通信正常 1
21	yyy_rrre0_R		UINT	AI	%IW10	10		
22	yyy_rrre1_R		UINT	AI	%IW12	11		
23	yyy_rrre2_R		UINT	AI	%IW14	12		
24	yyy_rrre3_R		UINT	AI	%IW16	13		
25	yyy_rrre4_R		UINT	AI	%IW18	14		
26	yyy_rrre5_R		UINT	AI	%IW20	15		
27	yyy_rrre6_R		UINT	AI	%IW22	16		
28	yyy_rrre7_R		UINT	AI	%IW24	17		

图 3- 22 BACnet/IP 服务选择变量完成界面

◆ 修改寄存器地址

组态软件支持用户自定义寄存器地址，可在相关位号的寄存器地址处双击输入需要的地址。

BACnet/IP 服务 x

☒ 使能

端口号: 47808

导出位号表

导入位号表

注意: 导入时, 仅会根据名字更新位号寄存器地址和注释

剩余可选个数: 94

#	名字	别名	数据类型	寄存器类型	地址	寄存器地址	注释	描述
1	bbaa0		BOOL	BV	%MX164	0		
2	bbaa1		BOOL	BV	%MX165	1		
3	bbaa2		BOOL	BV	%MX166	2		
4	bbaa3		BOOL	BV	%MX167	3		
5	bbaa4		BOOL	BV	%MX168	4		
6	bbaa5		BOOL	BV	%MX169	5		
7	bbaa6		BOOL	BV	%MX170	6		
8	bbaa7		BOOL	BV	%MX171	7		
9	bbaa8		BOOL	BV	%MX172	8		
10	bbaa9		BOOL	BV				
11	AppState		SINT	AI				用户程序状态【0: 运行;
12	NtpServer1State		SINT	AI				对时服务器1状态【0: 服
13	NtpServer2State		SINT	AI				对时服务器2状态【0: 服
14	IUsage		SINT	AI				I区使用率【百分比】
15	MUsage		SINT	AI				M区使用率【百分比】
16	QUsage		SINT	AI				Q区使用率【百分比】

输入

请输入一个数字

范围是 0 到 1023

确定

取消

图 3- 23 BACnet/IP 服务修改变量寄存器地址界面

◆ 修改寄存器类型

如果 BACnet/IP 服务内的变量寄存器类型是 AV 且数据类型是 BYTE、SINT 或者 USINT 中的一种，那么此变量寄存器类型可以被用户修改为 MV 类型，双击寄存器类型处进行修改。



图 3- 24 BACnet/IP 服务修改变量寄存器类型界面

96	uusint5		USINT	AV	%MB236	61	
97	uusint6		USINT	AV	%MB237	62	
98	uusint7		USINT	AV	%MB238	63	
99	bbyte16		BYTE	MSV	%MB217	42	
100	ssint8		SINT	MSV	%MB229	54	
101	uusint8		USINT	MSV	%MB239	64	
102	uusint9		USINT	MSV	%MB240	65	

图 3- 25 BACnet/IP 服务修改变量寄存器类型完成界面

3. 3. 5 以太网总线

以太网总线支持标准 Modbus TCP 协议，可单独配置 Modbus TCP 客户端和服务端。

3. 3. 5. 1 Modbus TCP 客户端

选择以太网总线的 Modbus TCP 客户端，在 TCP 客户端空白处右键可以选择添加、粘贴或者导入服务端选项，在新增的服务端中根据实际需要通讯的对象填写名称、IP 地址、端口号、从站地址等参数，这样就完成了服务端的添加。

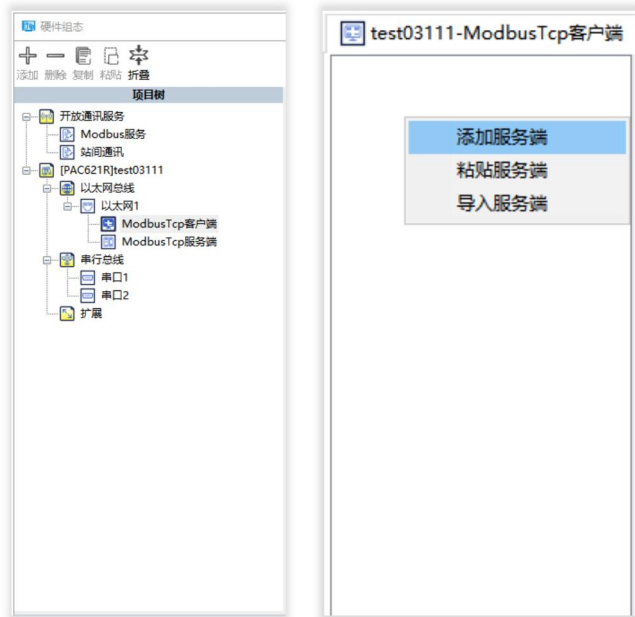
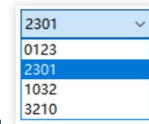


图 3-26 添加服务端界面



图 3-27 服务端参数界面

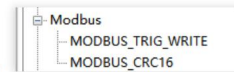
- **15 功能码**：可以支持单个线圈写或者多个线圈写功能。
- **16 功能码**：可以支持单个寄存器写或者多个寄存器写功能。
- **浮点字节/长整型顺序**：如果通讯的寄存器类型是浮点型或者长整型，即超过 1 个寄存器的



数据，则需要选择相应的字节顺序，支持四种字节顺序配置。

- **读周期(ms)**：默认配置 1000ms，即 Modbus TCP 客户端读取服务端的通讯周期是 1000ms，可根据用户需求自行配置。

- **触发写**：如果勾选了触发写，则 PLC 控制器不会在 Modbus 通讯中自动执行写行为，需要



用户在编程环境中调用系统库中的触发写功能块。反之如果未勾选触发写功能，则 Modbus 配置成周期写行为。

- **优化读取配置**：如果勾选优化读取配置，则 PLC 控制器会根据用户 Modbus 地址配置最大数据长度进行读取，如果未配置优化读取，则会根据用户真实配置的数据长度进行读取。举例：用户配置了 Modbus 地址 0 到 9 的 10 个寄存器，又配置了 20 到 29 的 10 个寄存器。则如果勾选了优化读取配置，PLC 控制器会发送从 0 地址读取 30 个寄存器的一包数据，如果未勾选优化读取配置，则 PLC 控制器会发送 0 地址到 9 地址的一包数据，接着马上又发送地址 20 到地址 29 的一包数据。

- **写周期**：只有在未勾选触发写的前提下才能进行配置，默认配置 1000ms，即 PLC 控制器会以 1000ms 周期发送写命令，用户可根据需求进行配置。

☐ 触发写	☒ 优化读取配置
写周期(ms)	1000

- **添加变量**

在右侧变量窗口空白处鼠标右键选择添加变量，可以选择保持寄存器、输入寄存器、线圈和离散输入四种 Modbus 类型，同时可以指定 Modbus 起始地址和位号数量，提供了 USINT、UINT、UDINT、SINT、INT、DINT、REAL、BYTE、WORD、DWORD 总共 10 种数据类型。



图 3- 28 添加客户端变量界面

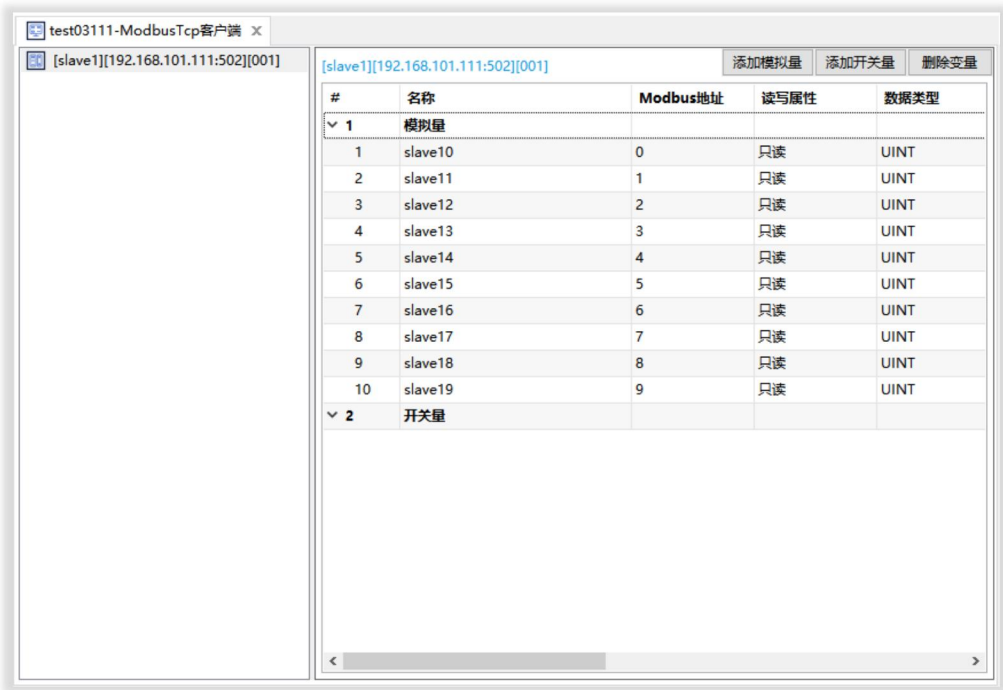


图 3- 29 客户端变量添加完成界面

至此，Modbus TCP 客户端的功能配置已经完成，可以在变量管理里面的硬件变量查看对应的变量信息。

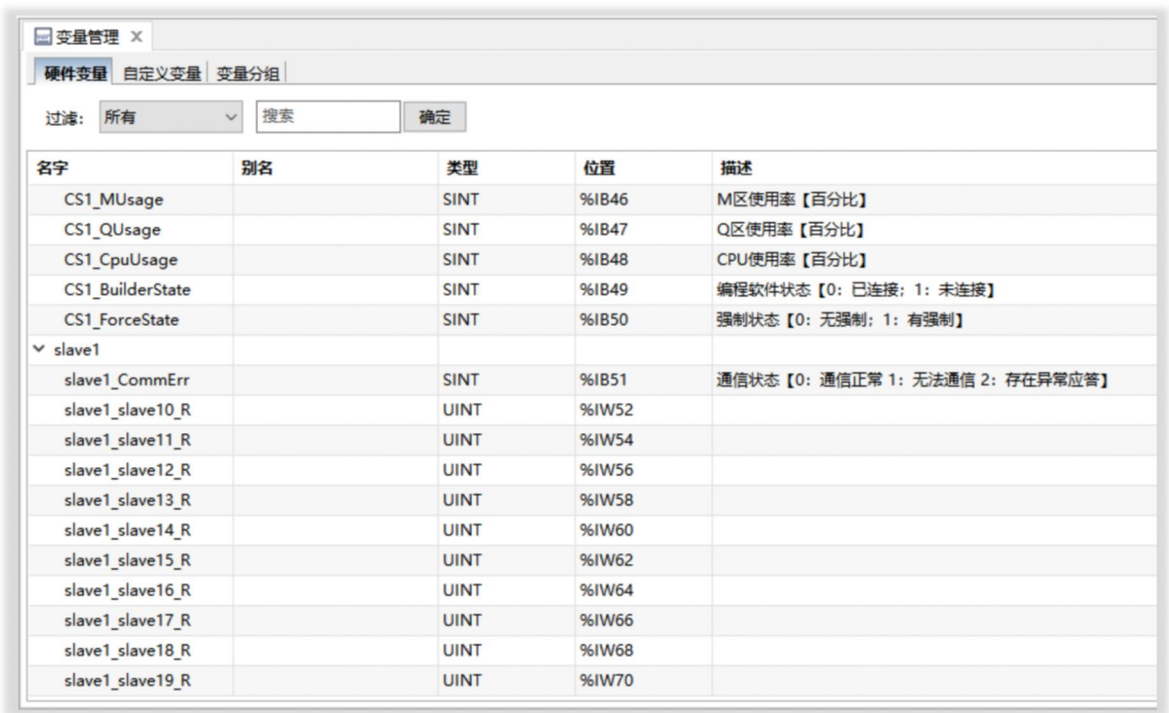


图 3- 30 客户端变量在变量管理中显示

3.3.5.2 Modbus TCP 服务端

Modbus TCP 服务端功能的开放点位配置在 Modbus 服务中进行，在服务端中只需要配置好本服务端的端口号即可。

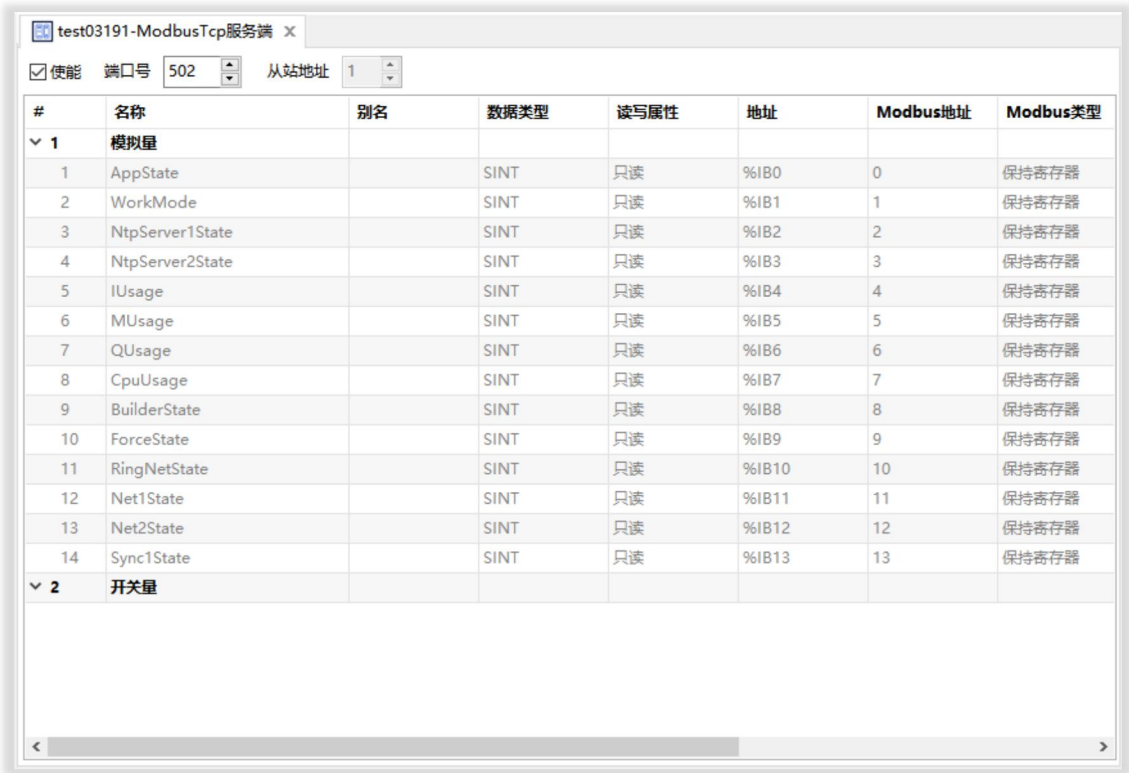


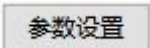
图 3- 31 服务端变量界面

3. 3. 6 串行总线

串行总线配置可支持标准 Modbus RTU 协议、自由串口主站协议、DLT645 电表协议以及 CJ188 水表协议，通过 PLC 控制器的 485 口实现第三方设备的数据通讯能力。根据 PLC 硬件的规格参数完成不同协议传感器的数据通讯。

3. 3. 6. 1 Modbus RTU 主站

● 串口参数设置

可通过参数设置按钮  进行串口的的基本参数配置，可配置波特率、数据位、停止位、奇偶校验位。

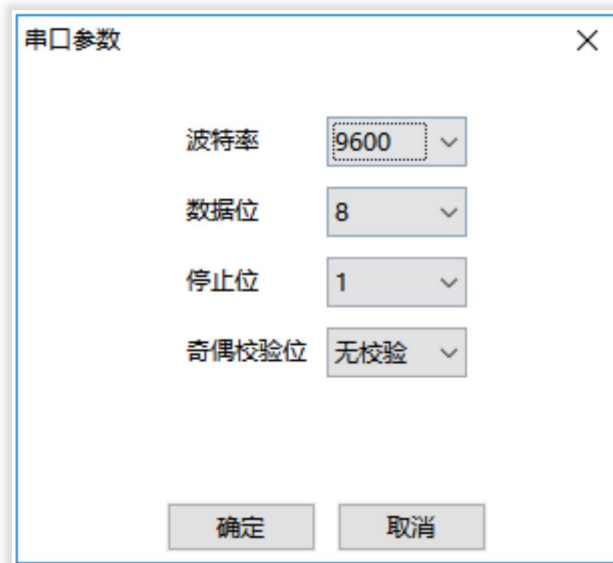
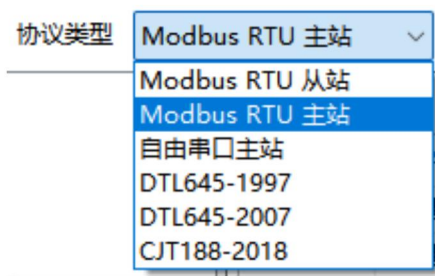


图 3- 32 串口参数设置

- 协议类型设置

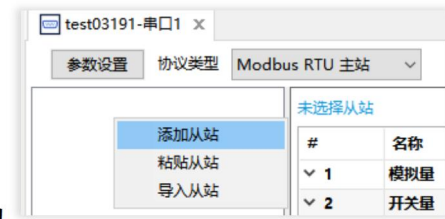
可通过协议类型下拉框选择串口的协议，支持 Modbus RTU 主站、Modbus RTU 从站、自由串口主站协议、DLT645-1997 电表协议、DLT645-2007 电表协议、CJT188-2018 水表协议



- 命令发送间隔

此参数用于配置串口在 Modbus RTU 主站情况下发送多条功能码的最小发送间隔，比如配置一个从站即有寄存器又有线圈，则默认配置命令发送间隔 1ms，PLC 的串口主站发送第一条读线圈命令之后，收到回复会立马发送第二条读寄存器命令，如果配置命令发送间隔 1000ms，则 PLC 的串口主站发送第一条读线圈命令之后，收到回复后要等 1000ms 才会发送第二条读寄存器命令。

- 添加从站



在串口主站窗口空白处右键选择添加从站按钮，填写名字、从站地址等必要参数，其他配置项可按照用户需求进行配置，功能跟以太网总线 Modbus TCP 客户端一样。



图 3- 33 从站参数配置

● 添加变量

在右侧变量窗口空白处鼠标右键选择添加变量，可以选择保持寄存器、输入寄存器、线圈和离散输入四种 Modbus 类型，同时可以指定 Modbus 起始地址和位号数量，提供了 USINT、UINT、UDINT、SINT、INT、DINT、REAL、BYTE、WORD、DWORD 总共 10 种数据类型。添加的从站变量可以“设为模板”，在添加从站的时候进行复用。

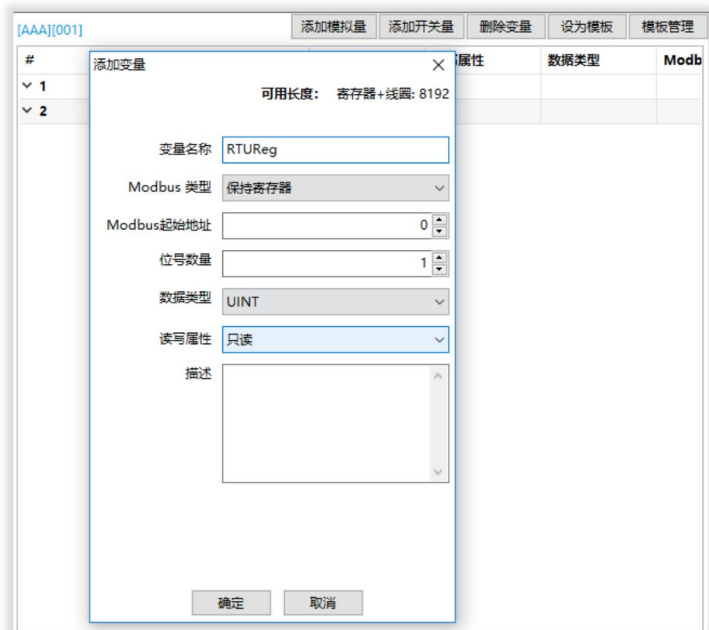


图 3- 34 添加变量

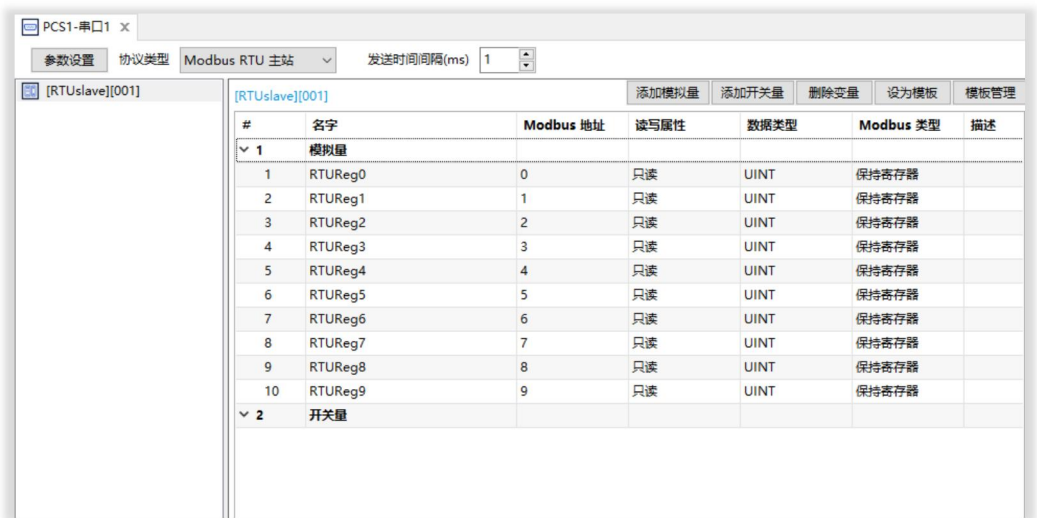


图 3- 35 添加变量显示界面

至此，Modbus RTU 主站的功能配置已经完成，可以在变量管理里面的硬件变量查看对应的变量信息。

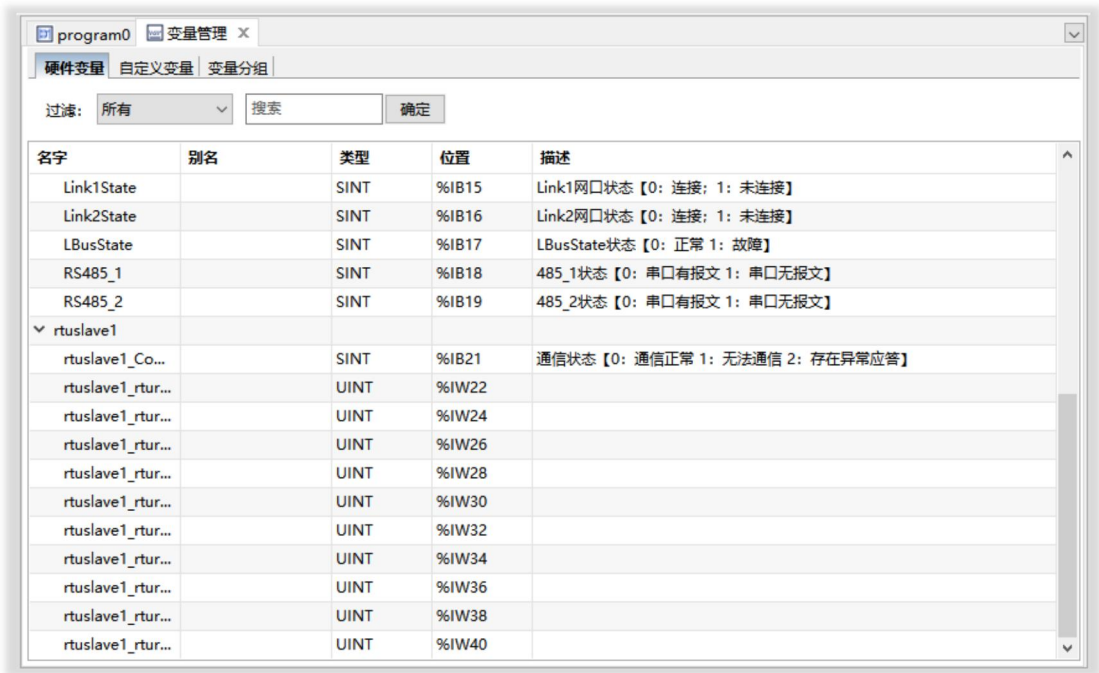


图 3- 36 变量信息查看界面

3. 3. 6. 2 Modbus RTU 从站

Modbus RTU 从站功能的开放点位配置在 Modbus 服务中进行，在从站中只需要配置好本站的地址即可

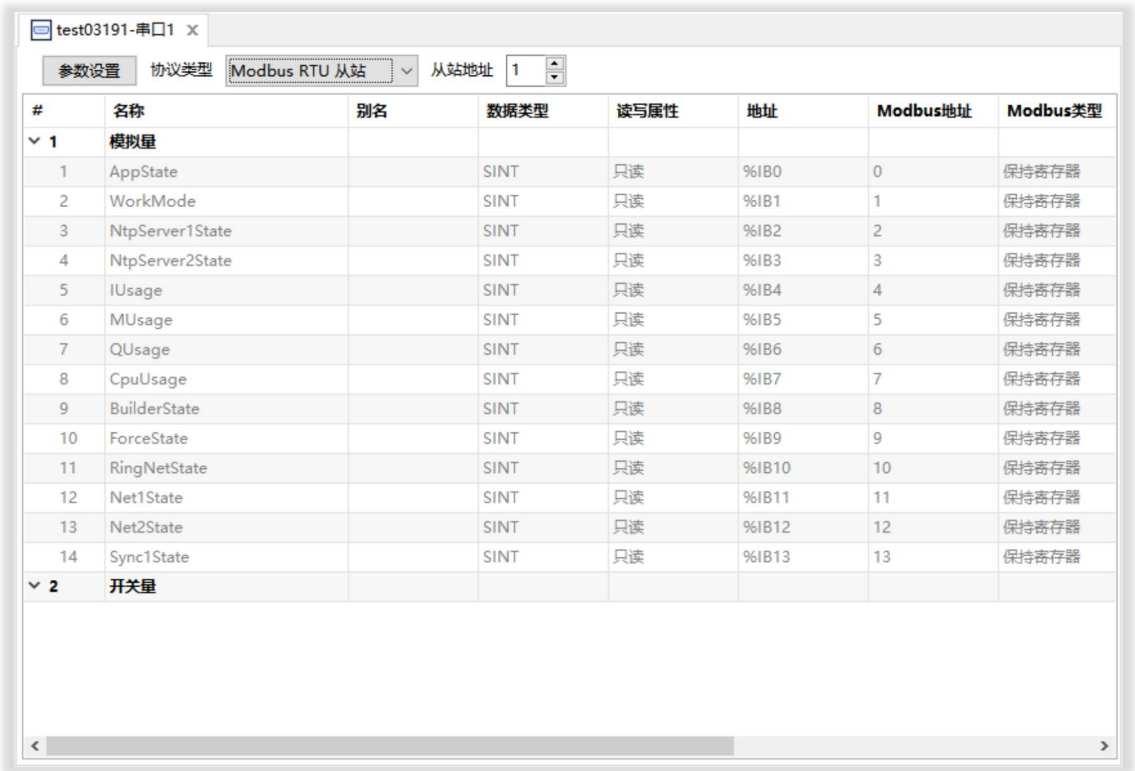
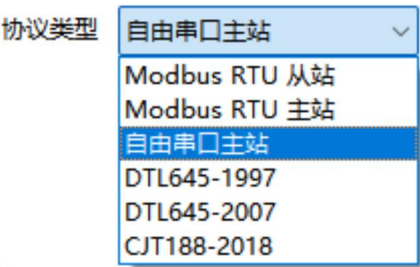


图 3- 37 从站变量界面

3.3.6.3 自由串口主站功能

自由串口主站功能是指 PLC 控制器具备自由配置和使用串口的能力，即串口可由用户自由发送和接收数据，主要应用于一些非标准协议的第三方设备对接，如传感器、执行器、人机界面等。



通过串口协议类型的下拉选框，选择自由串口主站功能



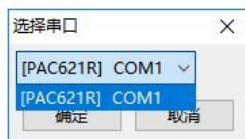
图 3- 38 串口参数设置及协议选择界面



用户通过系统库自带的串口发送和串口接收函数功能，来进行数据的发送和接收。

● 串口发送功能

在 POU 程序图中选择 “COM_SEND” 串口发送功能，选择要发送的串口



，梯形图 LD 程序和功能块图 FBD 程序调用如下

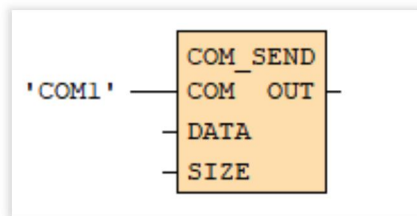


图 3- 39 串口发送 FBD 功能块

结构化文本 ST 程序调用如下

```
OUT(*USINT*) := COM_SEND( COM := 'COM1', DATA := (*ANY*), SIZE := (*UDINT*)) ;
```

图 3- 40 串口发送 ST 功能块

具体引脚定义请参考帮助栏里的《功能块手册》说明文档。

● 串口接收功能

在 POU 程序图中选择 “COM_RECV” 串口接收功能，选择要接收的串口



梯形图 LD 程序和功能块图 FBD 程序调用如下

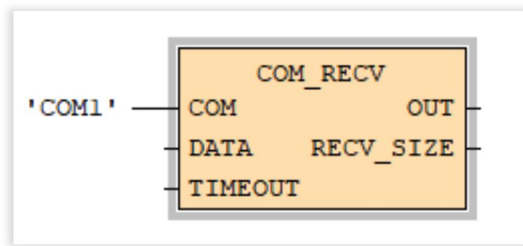


图 3- 41 串口接收 FBD 功能块

结构化文本 ST 程序调用如下

```
OUT(*USINT*) := COM_RECV( COM := 'COM1', DATA := (*ANY*), TIMEOUT := (*UDINT*), RECV_SIZE => (*UDINT*) );
```

图 3- 42 串口接收 ST 功能块

具体引脚定义请参考帮助栏里的《功能块手册》说明文档。

3.3.6.4 DTL645-1997/2007 电表协议功能

根据要通讯的电表类型选择 DTL645-1997 或者 DTL645-2007 协议，然后设置好相关串口参数，可参考 Modbus RTU 主站设置步骤。

● 添加电表



在串口页面窗口左边空白处右键选择添加电表按钮，然后填写设备名称和对应的实际表号，以及读取的周期。

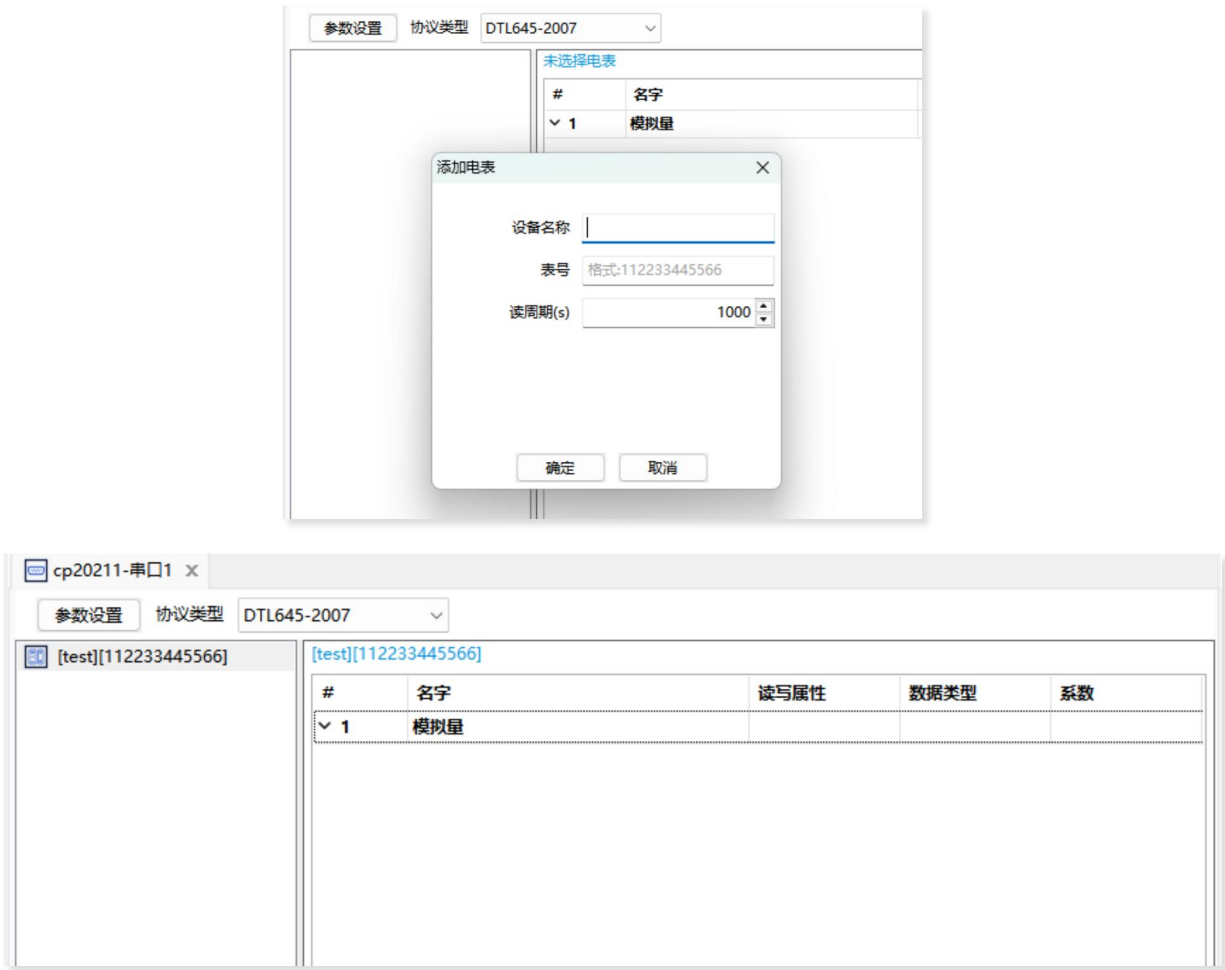


图 3- 43 添加电表

● 添加变量

在右侧变量窗口空白处鼠标右键选择添加变量，可根据电表需求选择总有功电能(kWh)、正向总有功电能(kWh)、反向总有功电能(kWh)、正向总无功电能(kWh)、反向总无功电能(kWh)、A 相电压(V)、B 相电压(V)、C 相电压(V)、A 相电流(A)、B 相电流(A)、C 相电流(A)、瞬时总有功(kW)、瞬时总无功(kW)、总功率因数这些数据标识，可根据需求自行修改相关变量系数。

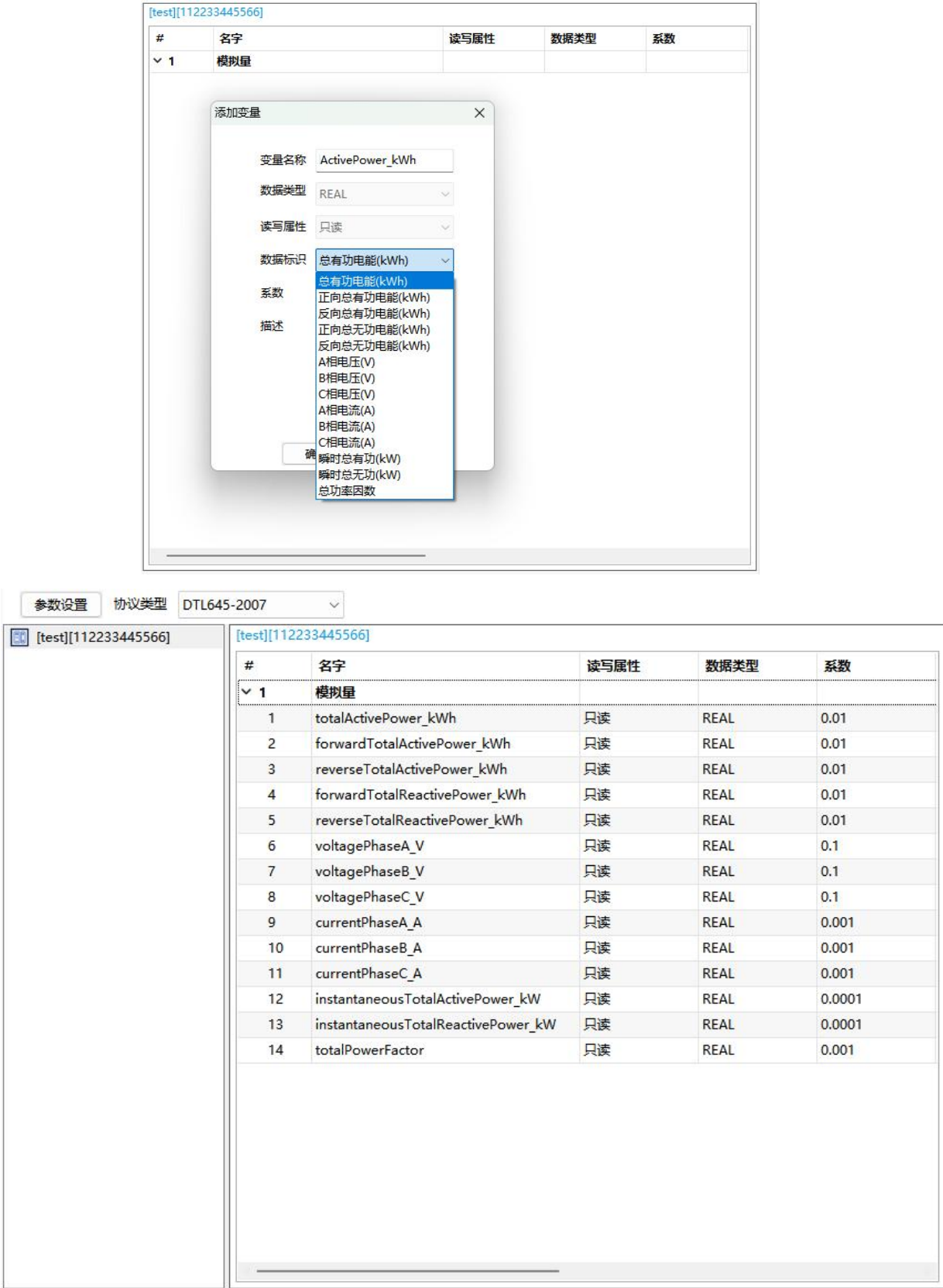


图 3- 44 添加电表变量

3. 3. 6. 5 CJT188-2018 水表协议功能

根据要通讯的电表类型选择 CJT188-2018 协议，其他配置如添加水表以及添加位号功能跟上述电表协议功能类似，可自行参考电表协议功能相关配置步骤，这里不再赘述。

3.3.7 扩展功能

扩展功能应用于 PLC 控制器可以通过扩展总线的方式来外扩模块，满足在各种应用场景。可支持添加扩展串口通讯模块、扩展以太网通讯模块、扩展 IO 模块等，可灵活自由配置 PLC 控制系统。

3.3.7.1 添加模块

进入硬件组态界面，右键选择“扩展”按钮 ，选择添加模块，可以添加相应的模块，不同的 PLC 控制器可添加扩展模块的类型也不一样。第一层级扩展可以添加远程子站和异构扩展模块，第二层级扩展（远程子站下属层级）可添加 IO 模块。

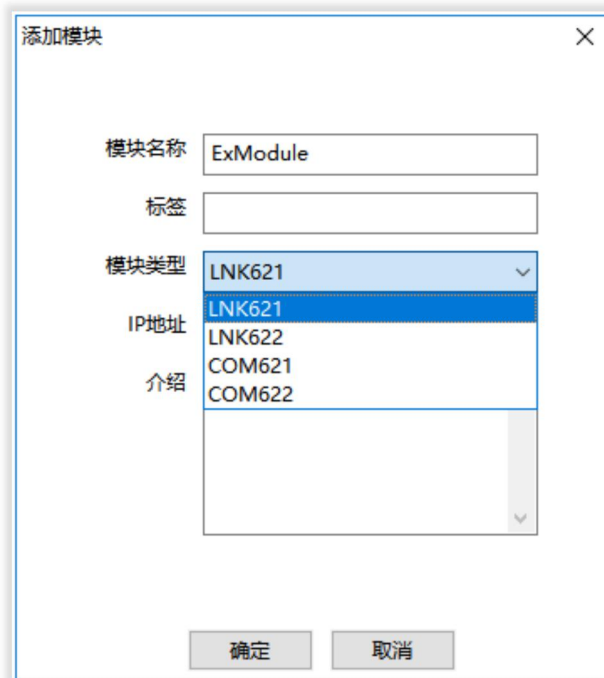


图 3- 45 添加远程子站及异构扩展模块模块

- 模块名称：为添加的模块命名，且只能以英文字母或下划线开头，由字母、数字和下划线组成，长度不超过 32 个字节。
- 标签：可为模块进行分类，添加标签后自动在模块前生成 xxx_模块名称。且只能以英文字母或下划线开头，由字母、数字和下划线组成，长度不超过 32 个字节。
- 模块类型：不可修改，根据相应控制器在下拉框内选择添加。
- IP 地址：遵循互联网通信协议第四版。
- 介绍：描述当前选择模块主要信息，不可修改。

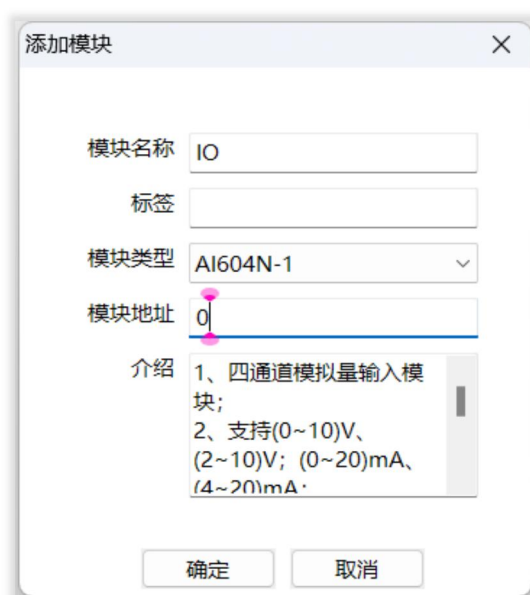


图 3- 46 添加 IO 模块

- 模块名称：为添加的模块命名，且只能以英文字母或下划线开头，由字母、数字和下划线组成，长度不超过 32 个字节。
- 标签：可为模块进行分类，添加标签后自动在模块前生成 xxx_模块名称。且只能以英文字母或下划线开头，由字母、数字和下划线组成，长度不超过 32 个字节。
- 模块类型：不可修改，根据相应控制器选择添加。
- 模块地址：可以手动输入模块地址，支持 0-31。连续添加模块地址自动+1。
- 介绍：描述当前选择 IO 模块主要信息，不可修改。

3.3.7.2 删除设备

3.3.7.2.1 单设备删除

在模块列表选择一个模块（设备），点击工具栏上的“删除”按钮或者右击选中的模块（设备）->选择“删除模块”。控制器模块不能被删除。

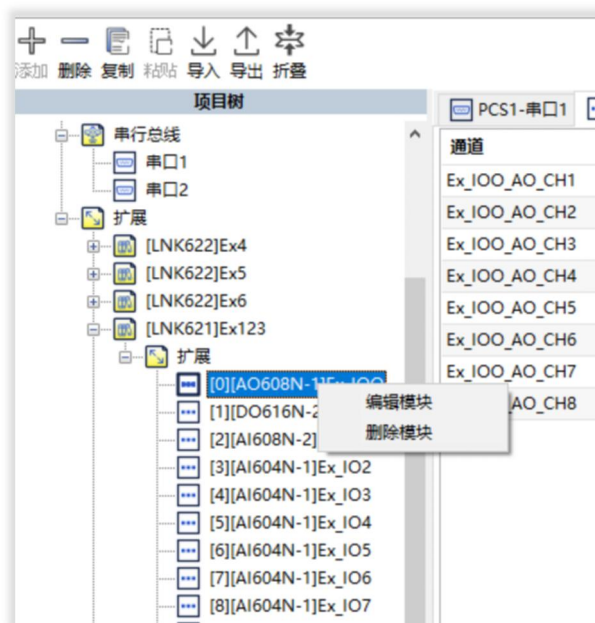


图 3- 47 单设备删除

3.3.7.2.2 批量删除

在模块列表选中“扩展”，右击鼠标选择“批量删除”，选择需要删除的模块进行删除。



图 3- 48 批量删除

3. 3. 7. 3 复用设备

3. 3. 7. 3. 1 导出设备

在工具栏选择导出



，可将硬件组态导出成 CSV 格式的接线表。

3.3.7.3.2 导入设备

在工具栏选择导入 ，可将导出的接线表导入生成硬件组态。

3.4 基本组态

对模块的属性、地址、通道信息等进行设置

- 远程子站

选择远程子站（LNK621/LNK622）鼠标“双击”，在右侧界面中显示该模块可以修改的属性，修改完成后保存生效。



图 3- 49 远程子站组态

- 异构通信

选择异构通信模块（COM621/COM622）鼠标“双击”，在右侧界面中显示该模块可以修改的属性，修改完成后保存生效。

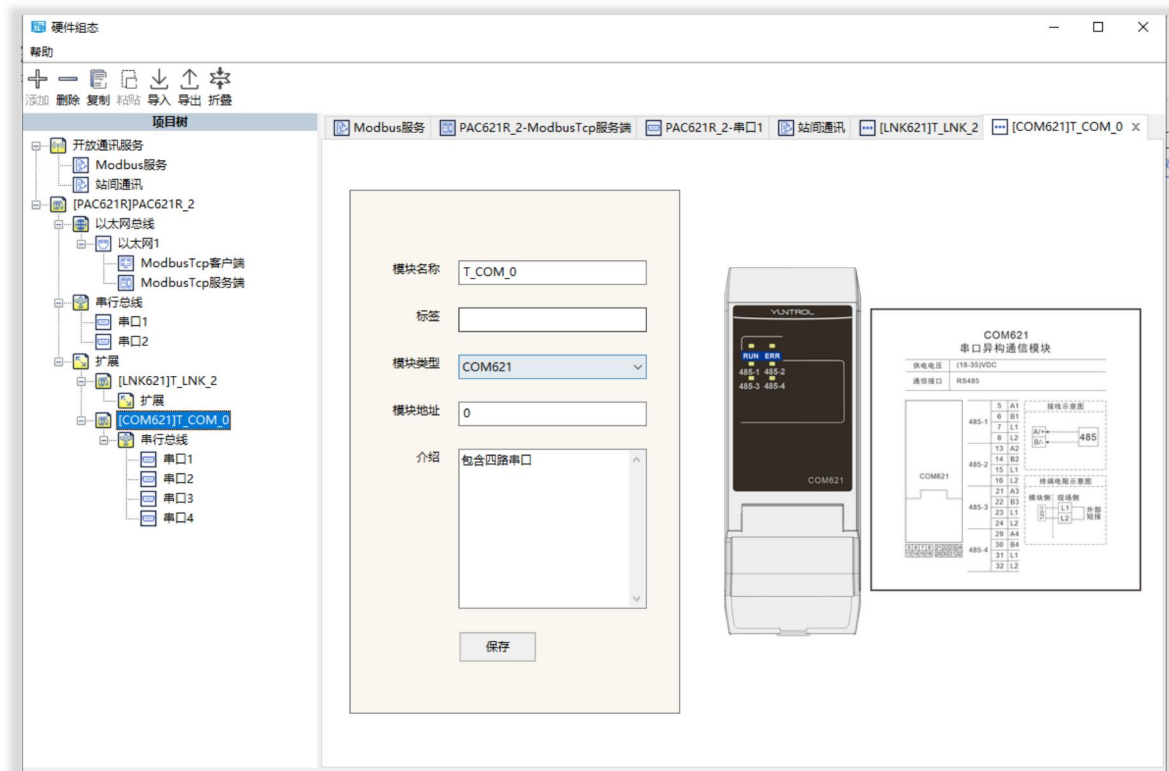


图 3- 50 异构模块组要

- IO 模块

选择 IO 模块，鼠标“双击”，在右侧界面中可以配置模块通道信息。

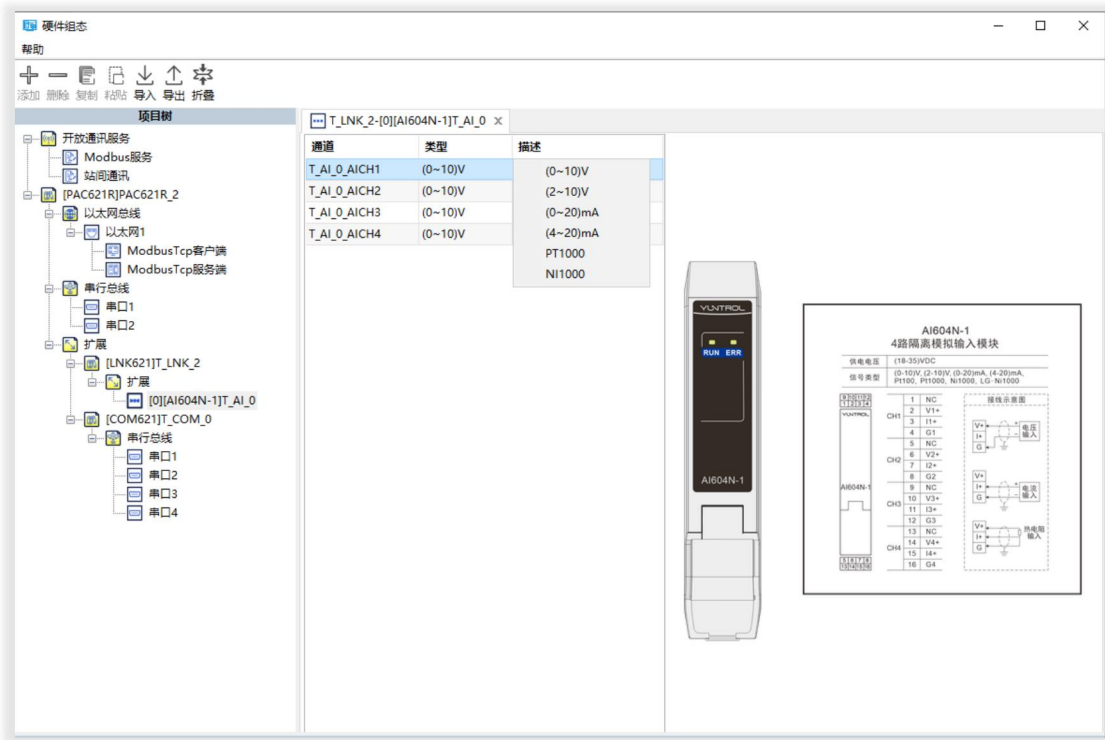


图 3- 51 IO 通道组态

3.5 本地IO

对模块本身自带的 I/O 点位进行配置（如果存在）。

通道	类型	滤波时间	描述
Local_AI_CH1	AI-(4~20)mA	/	LocalIO UI通道[1]信号类型
Local_AI_CH2	AI-(4~20)mA	/	LocalIO UI通道[2]信号类型
Local_AI_CH3	AI-(4~20)mA	/	LocalIO UI通道[3]信号类型
Local_AI_CH4	AI-(4~20)mA	/	LocalIO UI通道[4]信号类型
Local_AI_CH5	AI-(4~20)mA	/	LocalIO UI通道[5]信号类型
Local_AI_CH6	AI-(4~20)mA	/	LocalIO UI通道[6]信号类型
Local_AI_CH7	AI-(4~20)mA	/	LocalIO UI通道[7]信号类型
Local_AI_CH8	AI-(4~20)mA	/	LocalIO UI通道[8]信号类型
Local_DI_CH9	DI	100	LocalIO DI通道[1]信号极性
Local_DI_CH10	DI	100	LocalIO DI通道[2]信号极性
Local_DI_CH11	DI	100	LocalIO DI通道[3]信号极性
Local_DI_CH12	DI	100	LocalIO DI通道[4]信号极性
Local_DI_CH13	DI	100	LocalIO DI通道[5]信号极性
Local_DI_CH14	DI	100	LocalIO DI通道[6]信号极性
Local_AO_CH15	AO-(4~20)mA	/	LocalIO AO通道[1]信号类型
Local_AO_CH16	AO-(4~20)mA	/	LocalIO AO通道[2]信号类型
Local_AO_CH17	AO-(4~20)mA	/	LocalIO AO通道[3]信号类型
Local_AO_CH18	AO-(4~20)mA	/	LocalIO AO通道[4]信号类型
Local_DO_CH19	DO	/	LocalIO DO通道[1]信号极性
Local_DO_CH20	DO	/	LocalIO DO通道[2]信号极性
Local_DO_CH21	DO	/	LocalIO DO通道[3]信号极性
Local_DO_CH22	DO	/	LocalIO DO通道[4]信号极性

图 3- 52 本地 IO 通道组态

3.6 变量管理

3.6.1 硬件变量

硬件变量是由组态软件根据硬件组态的配置自动生成的位号变量，位号地址是由系统自动分配的，用户不可以进行更改，但是支持每个硬件变量都能单独修改别名和描述。从硬件变量的结构树中可以直观地观察当前 PLC 控制器的自身位号及诊断信息、扩展模块的位号及诊断信息、第三方通讯位号及诊断信息等，并且当处于调试状态时，用户可在硬件变量界面观察每个变量的数据信息，以及可通过写值或者强制功能去进行操作。

● 未处于调试状态模式

显示变量名称、别名、数据类型、地址、描述

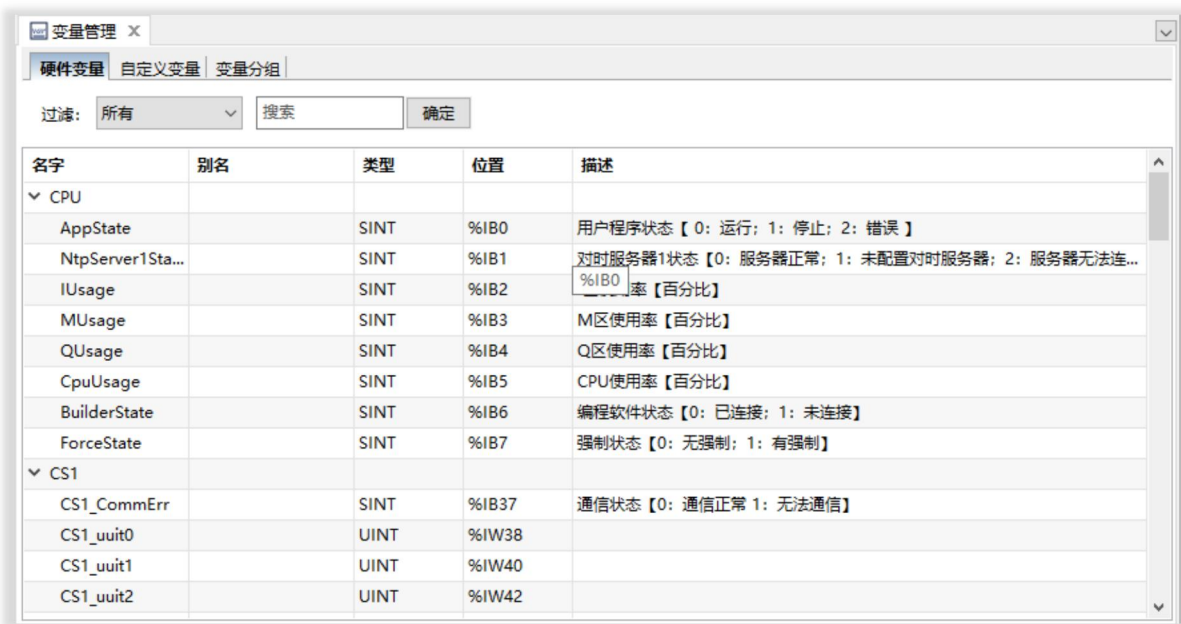


图 3- 53 未调试状态界面

● 处于调试状态模式

除了显示变量信息之外，还可以显示当前变量的实时值，并且通过双击鼠标左键来设置变量实时值以及强制当前变量的值。强制变量值指的是勾选强制功能后，这个变量的值就固定当前值，

不会再随着运算结果的变化而变化，通常这个功能可用于调试验证用户程序的某个功能，模拟实际变量的值来验证是否符合程序预期。

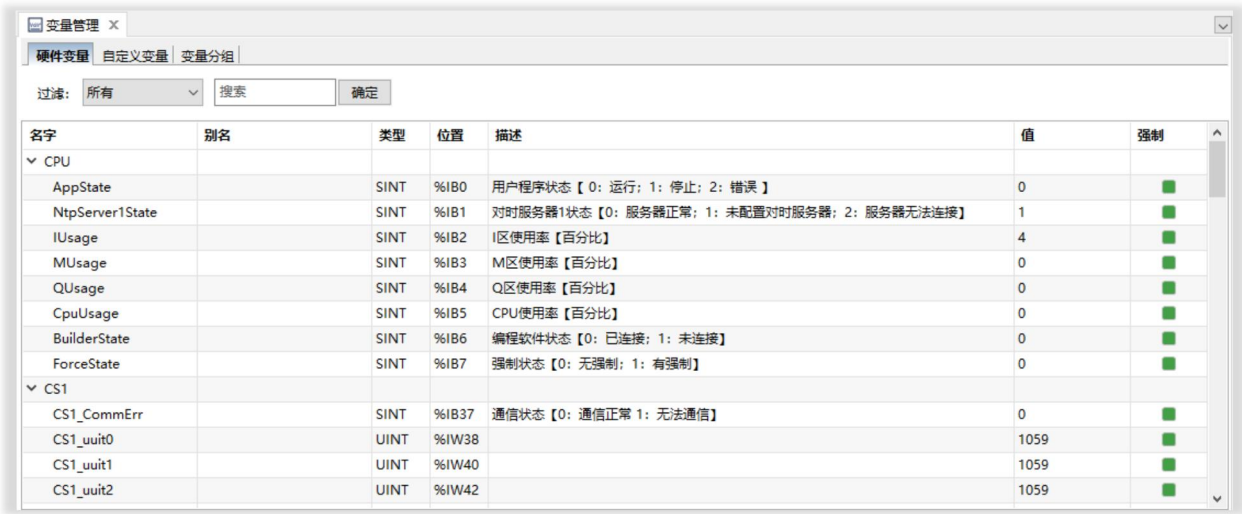


图 3- 54 调试状态界面

● 设置变量值



图 3- 55 调试状态变量修改

● 强制变量值

LNK621						
LNK621_Mo...		SINT	%IB...	模块匹配性 【0: 模块匹配 1: 模块不匹配】	1	☒
LNK621_Cor...		SINT	%IB...	通信状态 【0: 通信正常 1: 无法通信】	1	☒
LNK621_Cfg...		SINT	%IB...	组态状态 【0: 组态匹配 1: 组态不匹配】	1	☒

图 3- 46 强制变量界面

● 修改别名

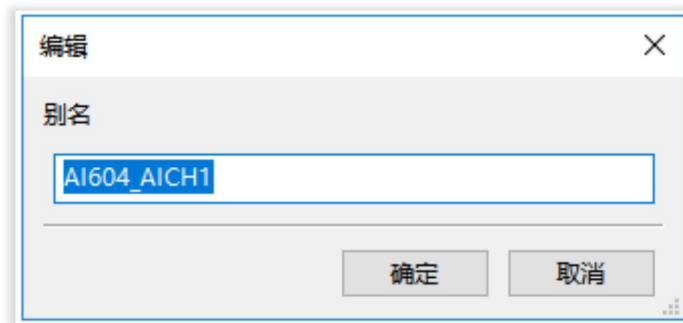


图 3- 56 编辑别名界面

- **修改描述**

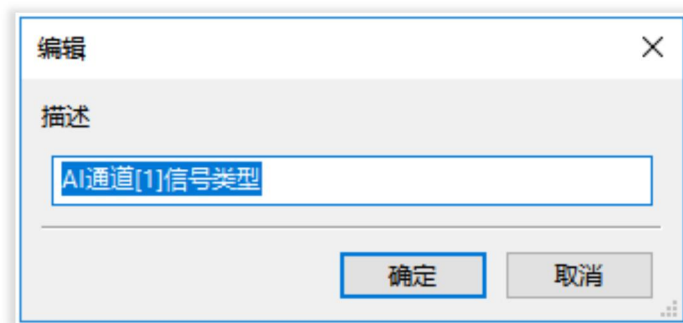


图 3- 57 修改描述界面

3. 6. 2 自定义变量

自定义变量功能允许用户创建和使用自定义变量来存储和处理数据。这些自定义变量定义在 M 区，具有以下功能：

- **数据存储**：可以用于存储各种类型的数据，如Bool、INT、REAL等20种系统自带数据类型。用户可以根据需要定义变量的数据类型和初始值。
- **数据处理**：可以用于进行各种数据处理操作，如算术运算、逻辑运算、比较操作等。用户可以使用自定义变量来执行复杂的计算和逻辑判断。
- **状态跟踪**：可以用于跟踪系统或设备的状态。例如，可以使用布尔型变量来表示设备的开关状态，或使用计数器变量来记录事件的发生次数。
- **逻辑控制**：可以用于控制程序的执行流程。用户可以使用条件语句和循环结构来根据自定义变量的值来决定程序的执行路径。

- **数据交换：**可以用于在不同的程序块之间传递数据。用户可以在一个程序块中定义变量，并在其他程序块中使用该变量来共享数据。
- **灵活性和可扩展性：**自定义变量功能使得PLC程序更加灵活和可扩展。用户可以根据需要创建任意数量的自定义变量，并根据实际情况进行调整和修改。

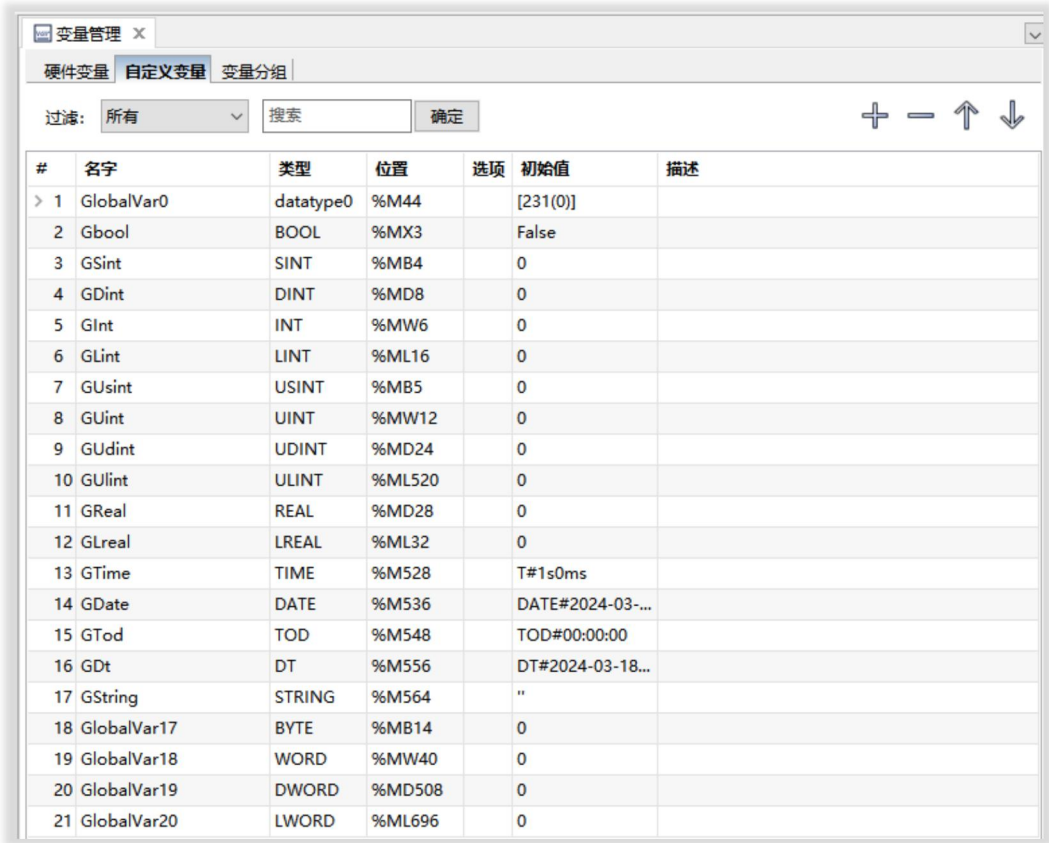


图 3- 58 自定义变量界面

● **过滤功能**

支持通过文本信息输入来过滤得到相应的位号



图 3- 59 过滤功能

3. 6. 3 变量分组

变量分组功能允许功能允许用户将相关的变量组织在一起，以便更好地管理和使用这些变量。通过将相关的变量放在同一个组中，用户可以更轻松地查找和访问这些变量，不需要在整个程序中搜索变量，只需查看特定的变量组即可。这样可以节省时间和精力，并提高编程效率。

通过空白处右键可以新建组，以及在组内可以添加组变量

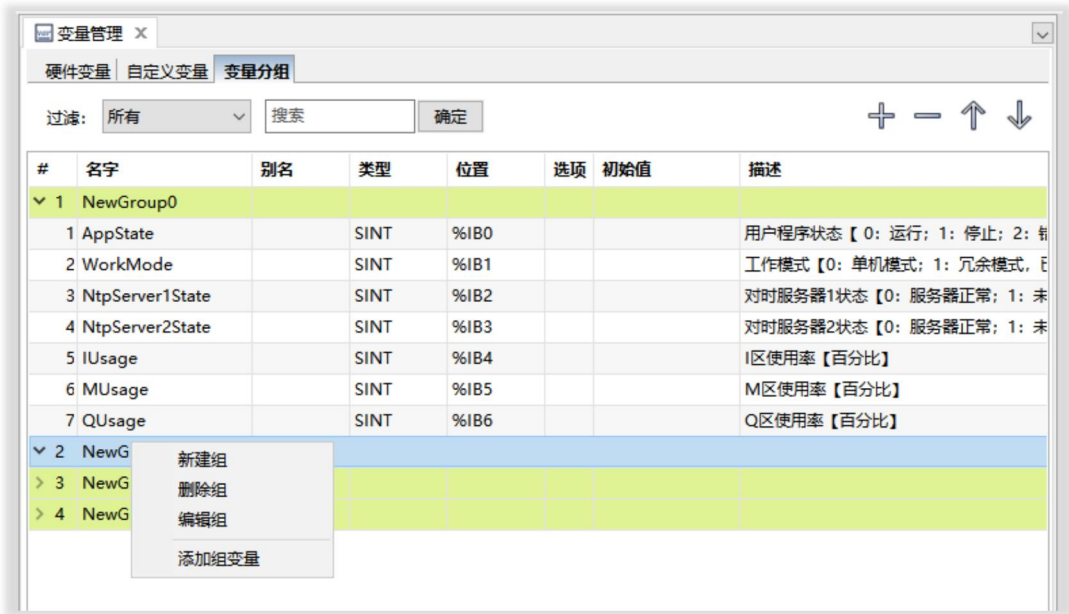


图 3- 60 新建/编辑变量分组

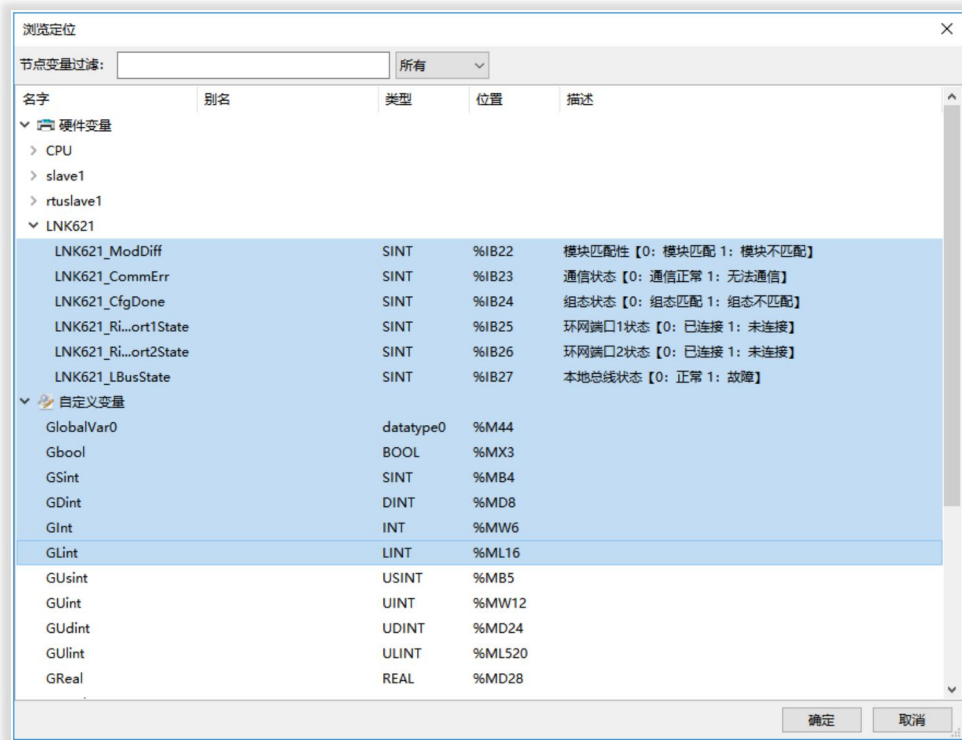


图 3- 61 选择分组变量

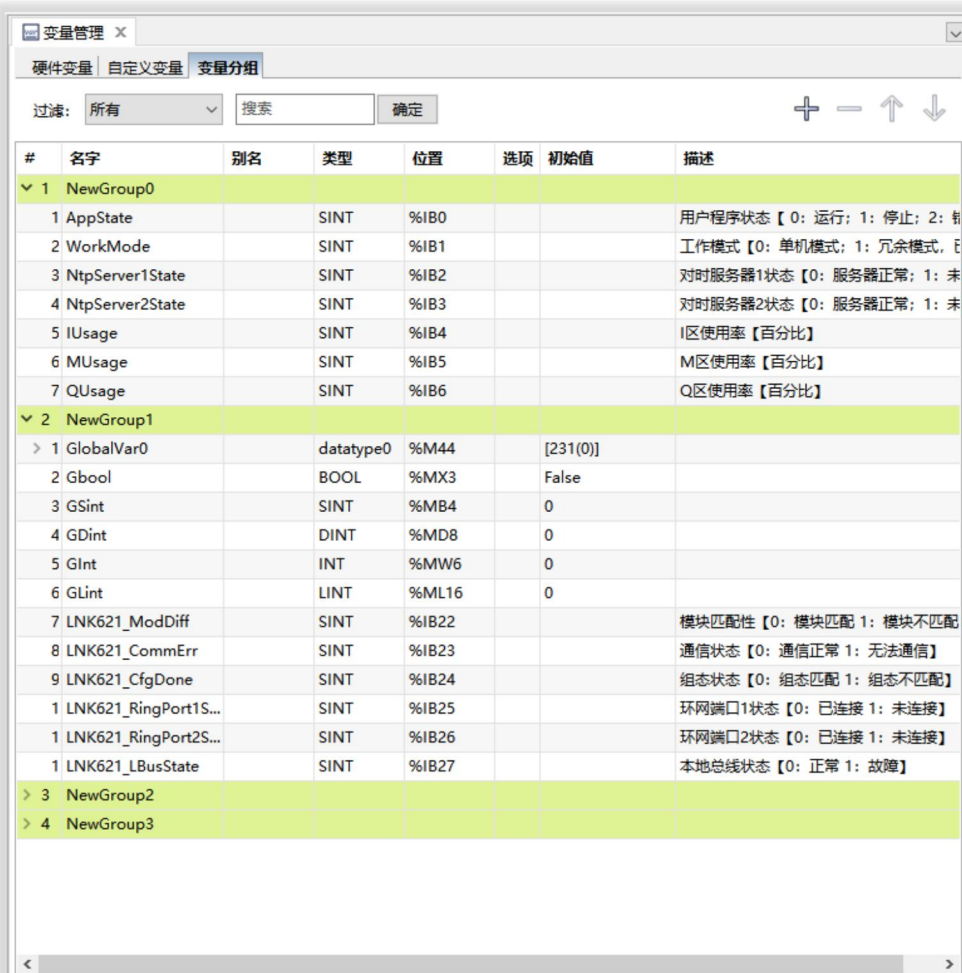


图 3- 62 分组完成界面

3.7 自定义数据类型和自定义功能块

本节主要介绍自定义数据类型和自定义功能块的配置方式。

3.7.1 自定义数据类型

组态软件可支持除了系统自带的 20 种数据类型之外，还可以用户自定义，方便用户进行对象化的数据传输和解析。在下图所示的界面中选择“数据类型”→右键鼠标选择“添加”，在右侧界面中显示自定义数据类型的配置界面。通过推导类型栏中的下拉菜单框来选择自定义数据类型的分类：结构的、数组。

通过按钮“   

”，“添加”、“删除”、“上移”、“下移”4 个按钮增加自定义数据类型（数组结构）的成员。


图 3- 63 添加自定义数据类型

成员：该自定义类型数据的内部成员。

- 名字：自定义数据类型的名称。名称需以英文字母或下划线开头，只能包含字母、数字和下划线并且长度不超过 32 个字节。

- 类型：内部成员数据类型。数据类型可以左击鼠标选择：“基类型”、“用户数据类型”、“数组”。
- 初始值：表示该变量在程序刚开始执行时的默认值。
- 描述：对该内部成员进行注解。

提示：

自定义数据类型配置成功后，即可在变量管理、函数、功能块、用户程序中添加选择使用该数据类型。

➤ 添加数据类型

导航到工程树下面的数据类型功能，右键可以选择添加、导入、导出、粘贴四种功能。选择添加功能，则会新增一个默认名称为 datatype0 的数据类型，用户可以重命名。

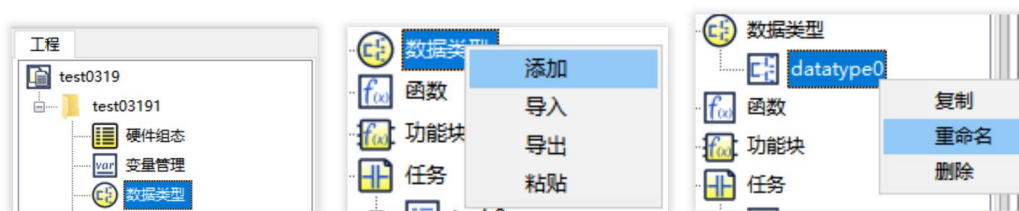


图 3- 64 添加数据类型步骤

➤ 自定义数据类型功能

用户可以通过推导类型下拉框选择新增的数据类型是数组还是结构体

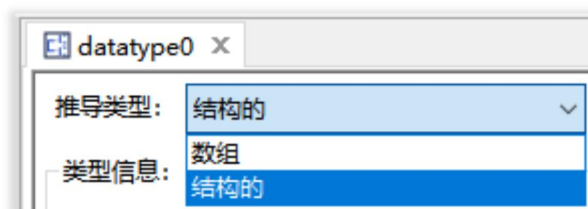


图 3- 65 自定义数据类型选择

➤ 结构体数据类型

通过数据类型窗口右侧的+号可以添加结构体成员数量

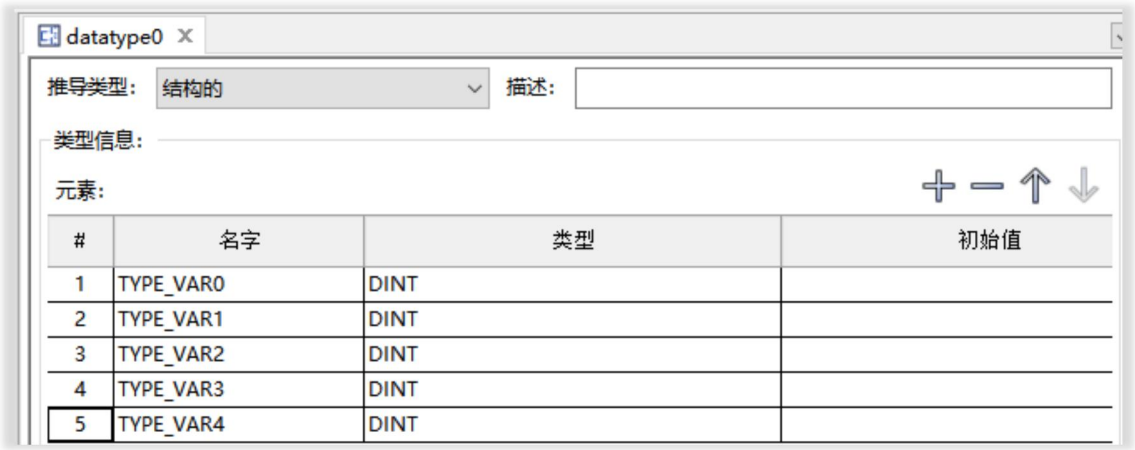


图 3- 66 添加数据类型成员

同时支持修改结构体成员的类型，可支持组态软件自带的 20 种基本类型、用户自定义的数据类型以及支持嵌套数组类型

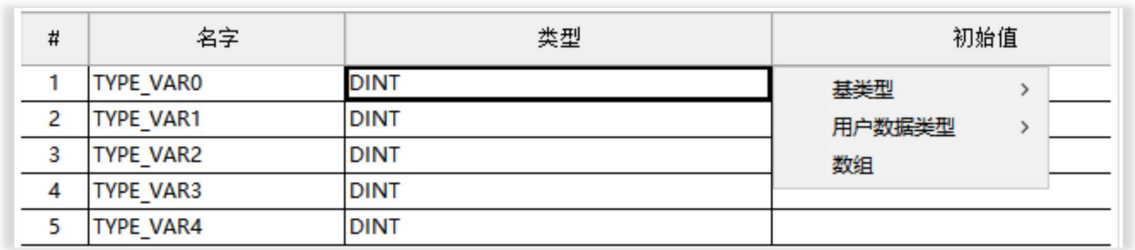


图 3- 67 修改成员类型

➤ 数组数据类型

首先选择数组的基本类型，支持系统自带的 20 种基本类型

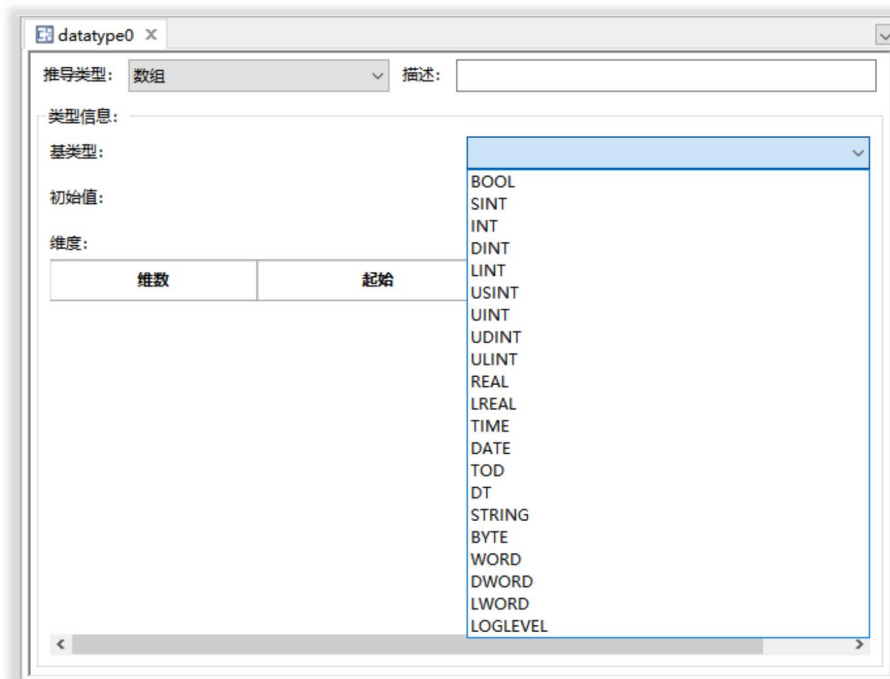


图 3- 68 添加数组形式的数据类型

通过数组类型右侧的+号添加数组可支持的维度，并且可以设置每个维度数组的长度

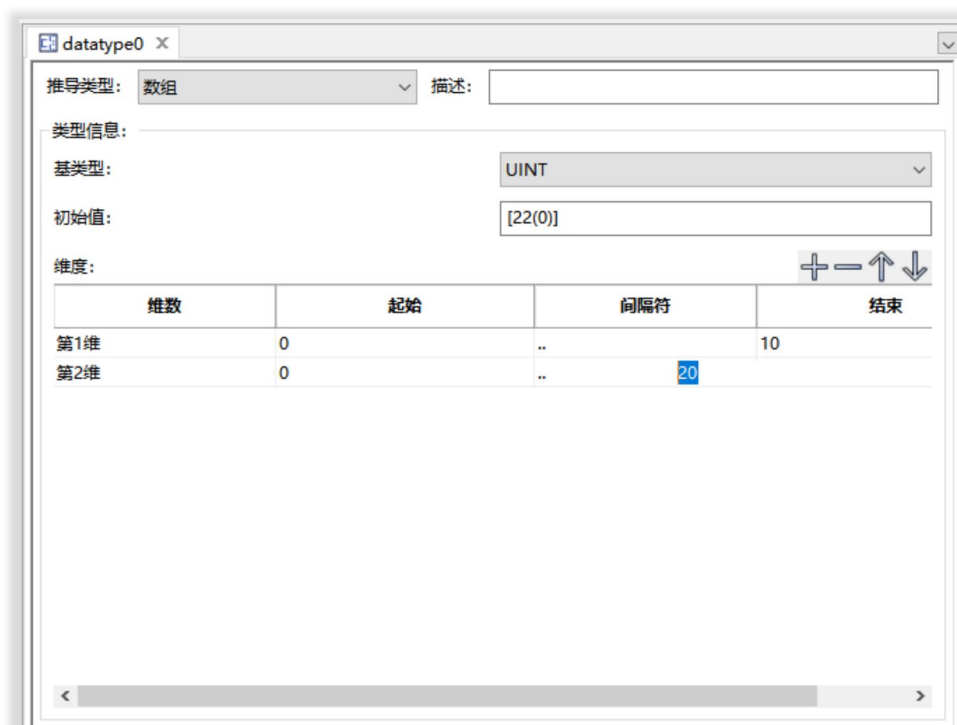


图 3- 69 编辑数组形式的数据类型

设置完以后点击窗口右上方的应用按钮进行应用保存，在变量应用中可使用用户自定义的数据类型

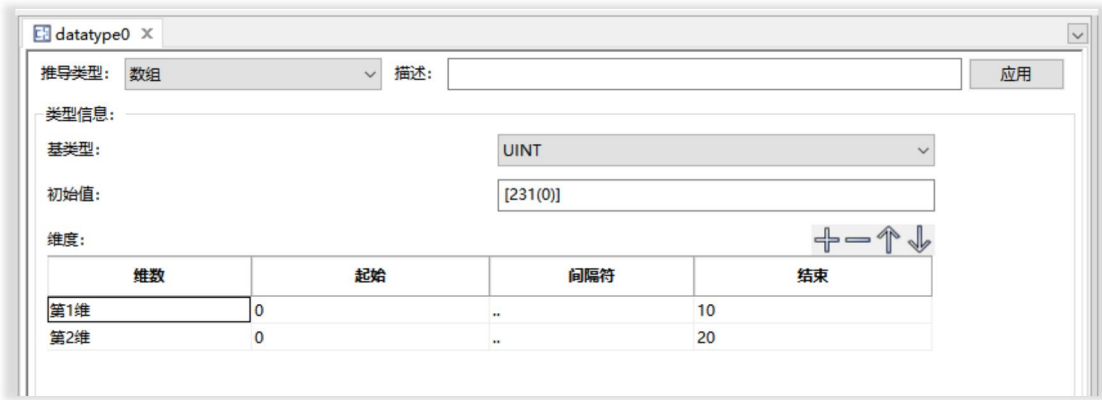


图 3- 70 编辑完成数据类型

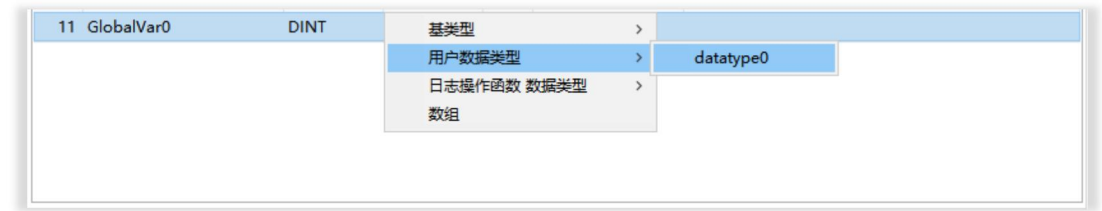


图 3- 71 将自定义变量选择为自定义数据类型

11	GlobalVar0	datatype0	%M44	[231(0)]	
-	GlobalVar0[0, 0]	UINT	%MW44	0	
-	GlobalVar0[0, 1]	UINT	%MW46	0	
-	GlobalVar0[0, 2]	UINT	%MW48	0	
-	GlobalVar0[0, 3]	UINT	%MW50	0	
-	GlobalVar0[0, 4]	UINT	%MW52	0	
-	GlobalVar0[0, 5]	UINT	%MW54	0	
-	GlobalVar0[0, 6]	UINT	%MW56	0	
-	GlobalVar0[0, 7]	UINT	%MW58	0	
-	GlobalVar0[0, 8]	UINT	%MW60	0	
-	GlobalVar0[0, 9]	UINT	%MW62	0	
-	GlobalVar0[0, 10]	UINT	%MW64	0	
-	GlobalVar0[0, 11]	UINT	%MW66	0	
-	GlobalVar0[0, 12]	UINT	%MW68	0	
-	GlobalVar0[0, 13]	UINT	%MW70	0	
-	GlobalVar0[0, 14]	UINT	%MW72	0	
-	GlobalVar0[0, 15]	UINT	%MW74	0	
-	GlobalVar0[0, 16]	UINT	%MW76	0	
-	GlobalVar0[0, 17]	UINT	%MW78	0	
-	GlobalVar0[0, 18]	UINT	%MW80	0	
-	GlobalVar0[0, 19]	UINT	%MW82	0	
-	GlobalVar0[0, 20]	UINT	%MW84	0	
-	GlobalVar0[1, 0]	UINT	%MW86	0	
-	GlobalVar0[1, 1]	UINT	%MW88	0	
-	GlobalVar0[1, 2]	UINT	%MW90	0	
-	GlobalVar0[1, 3]	UINT	%MW92	0	
-	GlobalVar0[1, 4]	UINT	%MW94	0	
-	GlobalVar0[1, 5]	UINT	%MW96	0	

图 3- 72 展开自定义数据类型的变量

3. 7. 2 自定义功能块

OptiBuilder 2.0 组态软件除了提供上述函数功能之外，还提供了功能块封装功能。功能块与函数有很多共同点，都是可以进行代码封装、参数传递、模块化设计和结构化编程。但是函数和功能块之间也存在一些区别：

- ◆ **语法和语义：**函数通常使用结构化文本（如ST）或其他高级编程语言来定义和实现。它们遵循特定的语法和语义规则。而功能块通常使用图形化的符号表示法（如梯形图LD、功能块图FBD）来定义和实现。
- ◆ **可见性和作用域：**函数的作用域通常是全局的，可以在整个程序中调用。而功能块的作用域通常是局部的，只能在定义它们的程序块内部调用。
- ◆ **可重用性：**函数通常具有更高的可重用性，因为它们可以在不同的程序中多次调用。而功能块的可重用性通常限于定义它们的程序块内部。

在图 3-5 所示的界面中选择“功能块”→右击鼠标“添加”，弹出对话框“新建一个 POU”，菜单，可选择添加自定义功能块的 POU 名字、编程语言（结构化文本 ST、梯形图 LD、功能块图 FBD）。

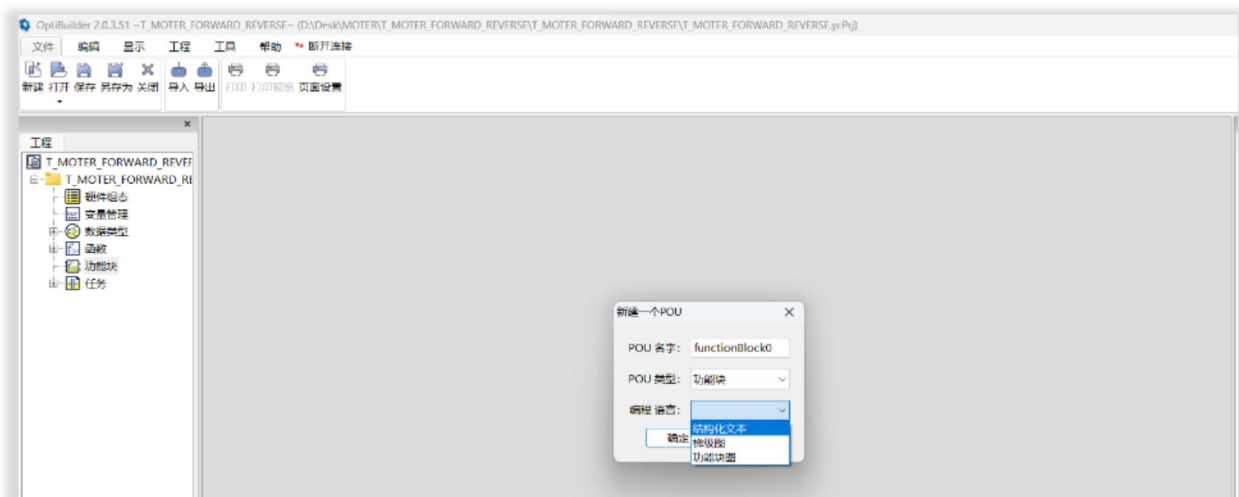


图 3- 73 新建自定义功能块

👉 **POU 名字：**自定义功能块的名称，名称需以字母开头，只能包含字母、数字和下划线并且长度 不超过 32 个字节。

👉 **编程语言：**选择自定义功能块编程语言。目前支持结构化文本、梯形图、功能块。

功能块可以接收输入参数，并根据参数的值执行相应的操作。输入参数可以是变量、常量或其他数据类型。功能块还可以生成输出结果，可以是变量、常量或其他数据类型。



图 3- 74 设置自定义功能块变量

用户可以将复杂的控制逻辑和操作封装成可重用的代码块并且进行应用

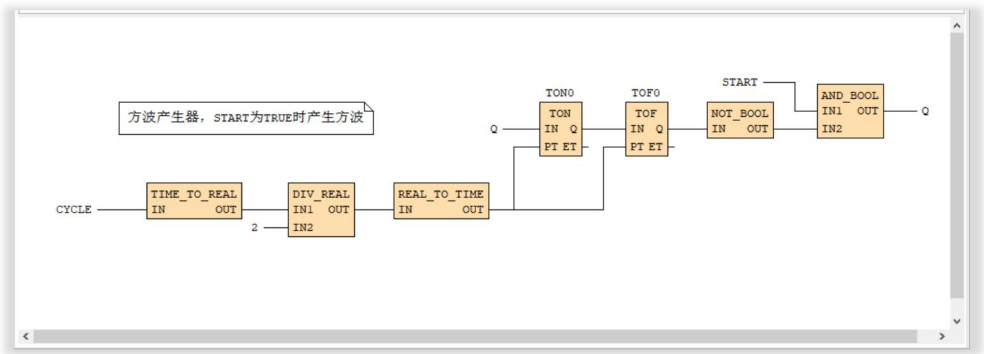


图 3- 75 封装自定义功能块

功能块封装好后，在用户程序中可对自定义的功能块进行调用

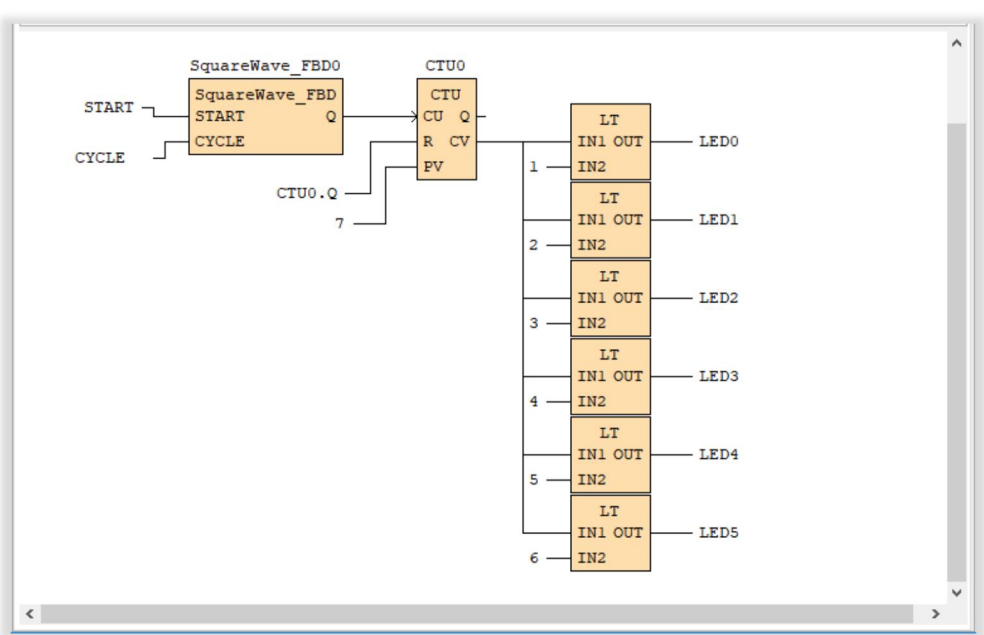


图 3- 76 调用封装好的自定义功能块

3. 7. 3 分组功能

自定义数据类型、自定义功能块以及自定义函数都支持添加分组，通过拖动的方式进行按组分类。

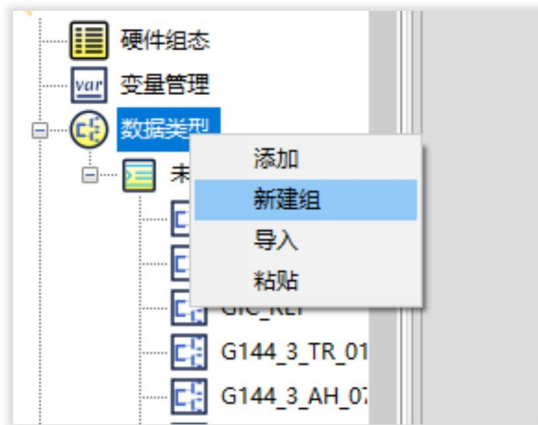


图 3- 77 新建分组



图 3- 78 分组显示

3. 7. 4 复用配置

3.7.4.1 导出配置

自定义数据类型、自定义功能块、自定义函数以及程序都支持导出配置实现复用，在需要导出的文件所在的分组处，右键点击导出，勾选需要导出的文件内容，即可导出至下方选择的目标位置。

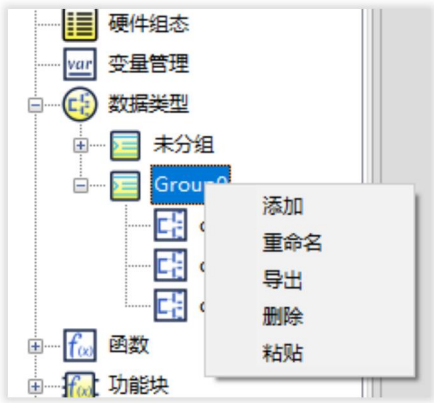


图 3- 79 自定义数据类型导出

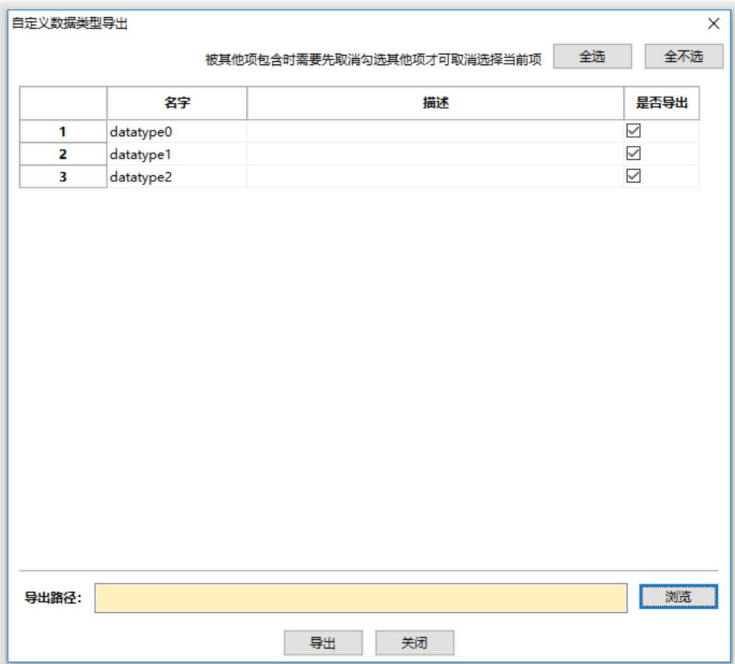


图 3- 80 导出选择界面

3.7.4.2 导入配置

导出的文件可以在同类项目录下点击右键进行导入。例如导出的自定义数据类型可以在数据类型点击右键进行导入。

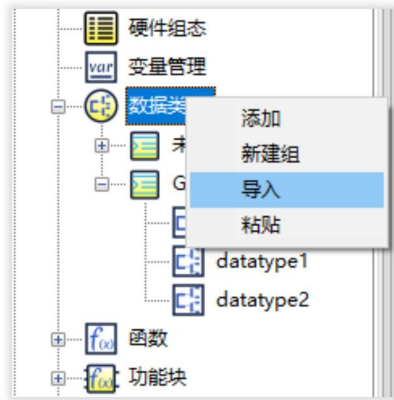


图 3- 81 导入选择界面

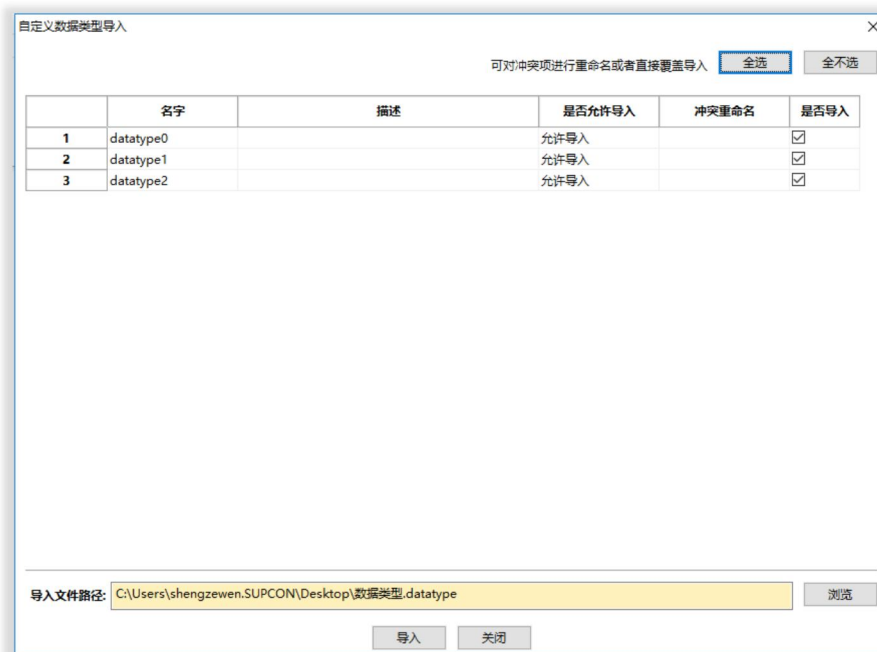


图 3- 82 导入选择界面

3.7.4.3 全局导入

当同一项目中含有多个工程时，可以选择右键项目，对项目中的所有工程进行导入操作。

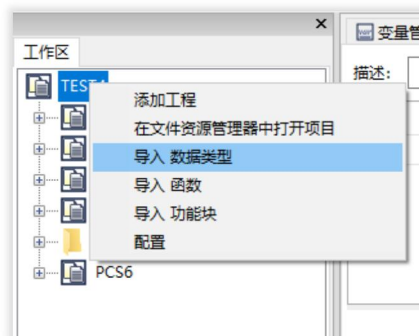


图 3- 83 导入选择界面

3.8 函数

OptiBuilder 2.0 组态软件提供了函数封装功能，它是一种用于执行特定任务或操作的可重用代码块。用户可以提高编程效率、代码的可读性和可维护性。函数功能还可以促进程序的模块化设计和结构化编程，使得程序更易于理解和调试。函数功能是一种重要的编程工具，对于实现复杂的控制任务和提高程序的可重用性具有重要意义。

在工程树中选择“函数” -> 右键鼠标“添加”，在右侧界面中弹出对话框“新建一个 POU”，可选择添加自定义功能块的 POU 名字、编程语言（结构化文本 ST、梯形图 LD、功能块图 FBD）。

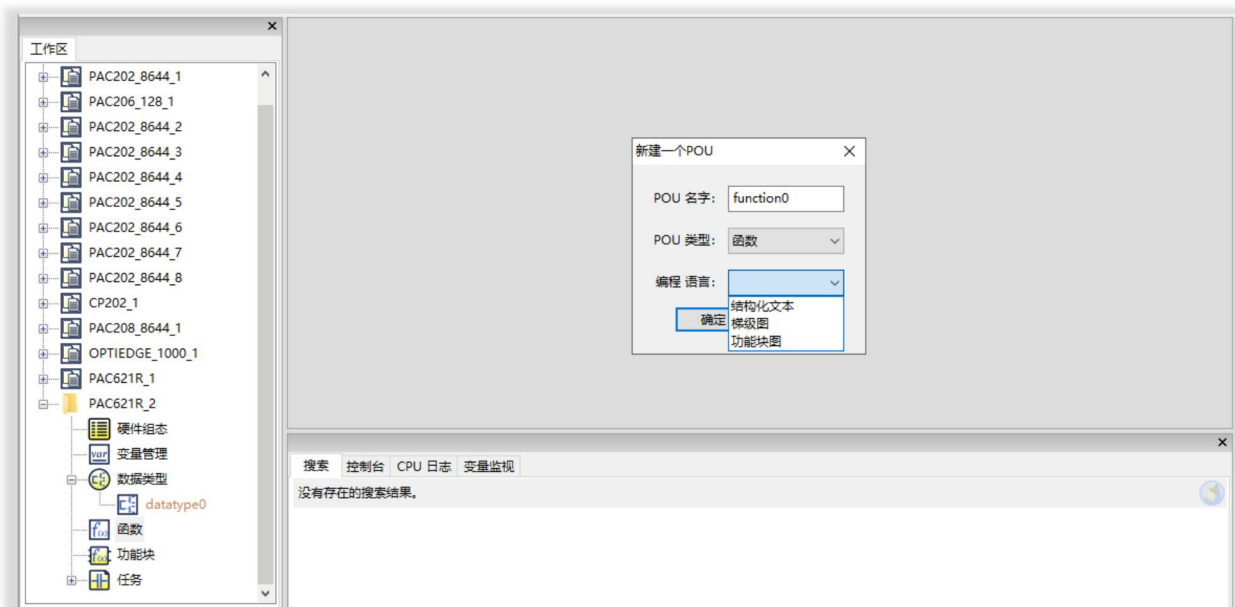


图 3- 84 添加自定义函数

- POU 名字：自定义功能块的名称，名称需以字母开头，只能包含字母、数字和下划线并且长度 不超过 32 个字节。

- 编程语言：选择自定义功能块编程语言。目前支持结构化文本、梯形图、功能块。

函数功能可以返回一个值作为结果，这个返回值默认支持系统自带的 20 种数据类型，也可以支持用户自定义的数据类型。通过返回值，函数可以将计算结果或处理结果传递给调用函数的地方。

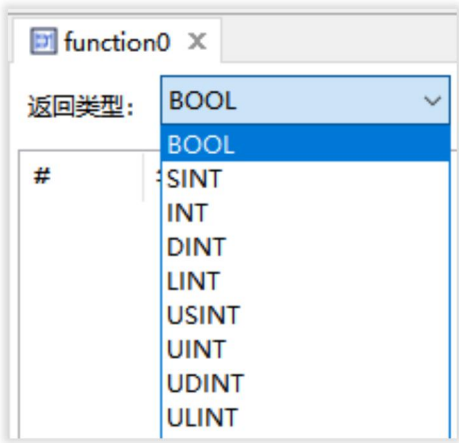


图 3- 85 选择函数返回值类型

函数功能可以定义局部变量，这些变量只在函数内部可见和使用。局部变量的作用域仅限于函数内部，不会与其他函数或全局变量发生冲突。

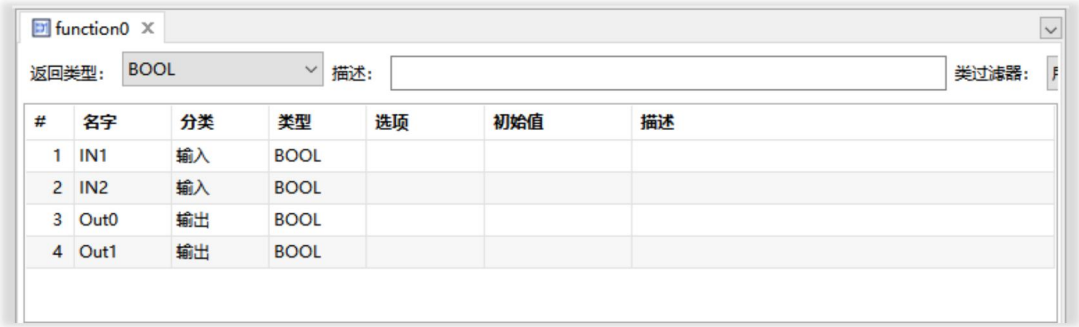


图 3- 86 添加函数所需变量

函数功能可以将一段代码封装成一个独立的单元。这样可以隐藏代码的实现细节，提高代码的安全性和可维护性。

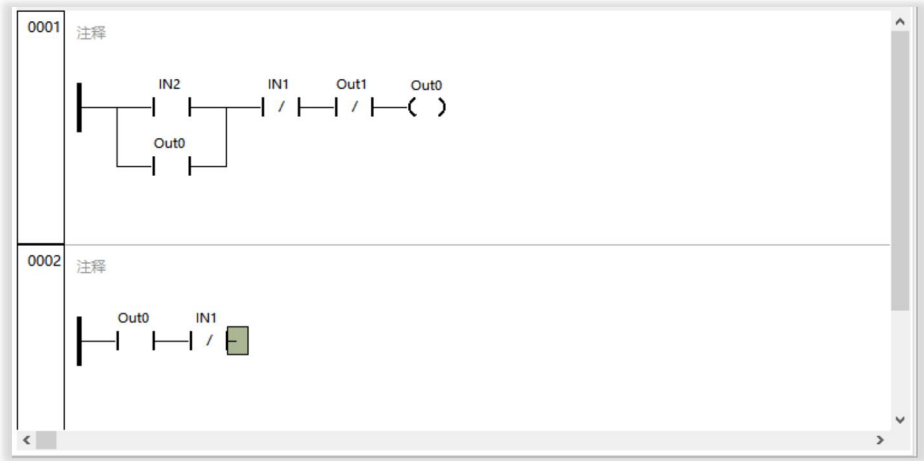


图 3- 87 封装成单元调用

函数功能封装好后，在用户程序中可对自定义的函数进行调用。

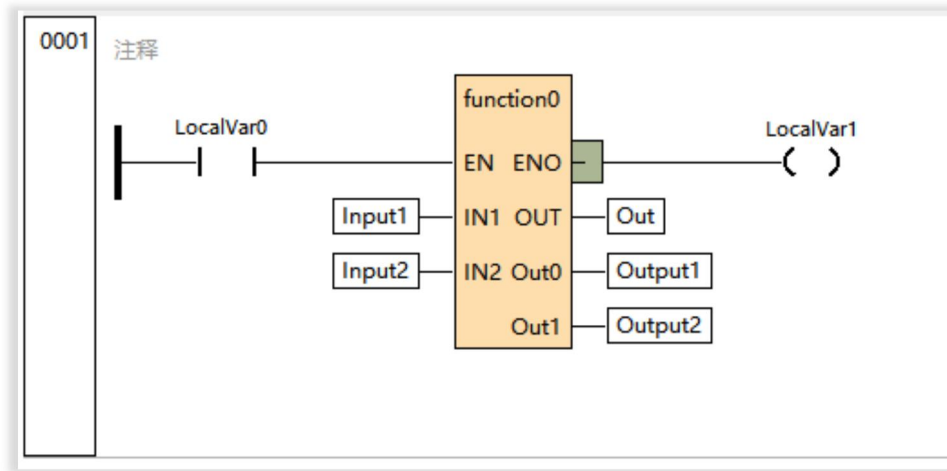


图 3- 88 封装成单元调用

3.9 任务配置（控制方案组态）

用户可定义不同执行模式的任务来实现不同的控制方式。可定义的执行模式为周期执行和中断执行。在不同模式的任务下可添加多个程序，实现控制逻辑。

软件还提供了“未调度任务”节点，在该节点下的程序不被运行，用户在调试程序时可将暂时不希望运行的程序拖动到该节点下，仅让希望运行的程序运行。

各类型模式的定义和特点说明如下表所示。

表 3-4 各类型任务定义、特点说明

任务模式	定义	特点	备注
周期任务	按照固定周期运行的用户任务	1. 当周期到时，中断所有低优先级的任务，执行一次周期任务。 2. 周期时间可组态范围为 10ms~2000ms，默认 20ms	适用于需要按照固定周期运行的控制任务。 如：每 100ms 控制步进电机位移 10cm
事件任务	当触发源产生上升沿时立即执行的用户任务	当触发源产生上升沿时，执行一次当前的事件任务。	适用于需要高速响应的控制任务。 如：储罐压力值超过安全值后立即进行紧急停车操作

3.9.1 添加任务

用户可在工程项目树中选择“任务”右键鼠标“添加”创建相应的任务来进行编程操作，可根据需求添加一个或者多个任务，并且可在不同任务中分别添加不同的 POU 程序，系统支持添加功能块图（FBD）、梯形图（LD）、结构化文本（ST）三种 POU 程序。如果创建多个任务，则执行顺序由上至下进行执行。

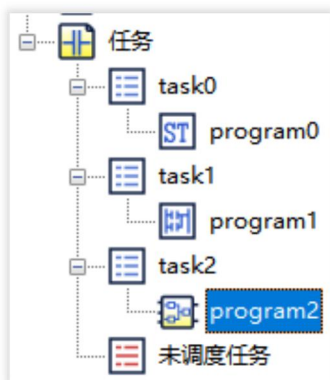


图 3- 89 多种类型的任务

如果用户在编写完 POU 程序后想暂时不启用又不想删除，可将相应的任务拖到未调度任务中进行屏蔽，不会删除用户程序代码。

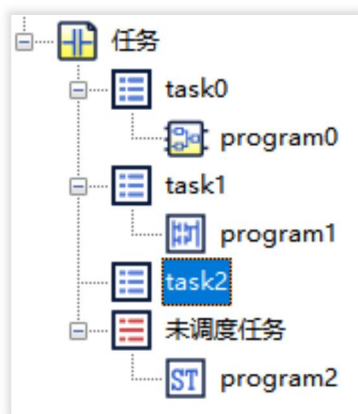


图 3- 90 未调度任务

每个添加的任务都支持单独配置，可在创建的任务重右键选择配置

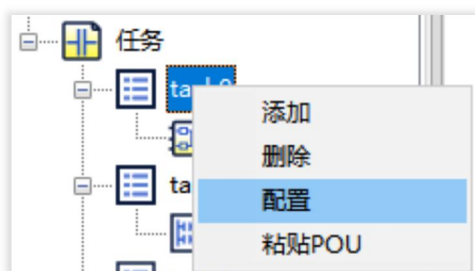


图 3- 91 配置任务

配置界面包括信息、配置、管理三个菜单栏。用户可以在信息栏中修改创建的任务名称，添加对任务的描述；

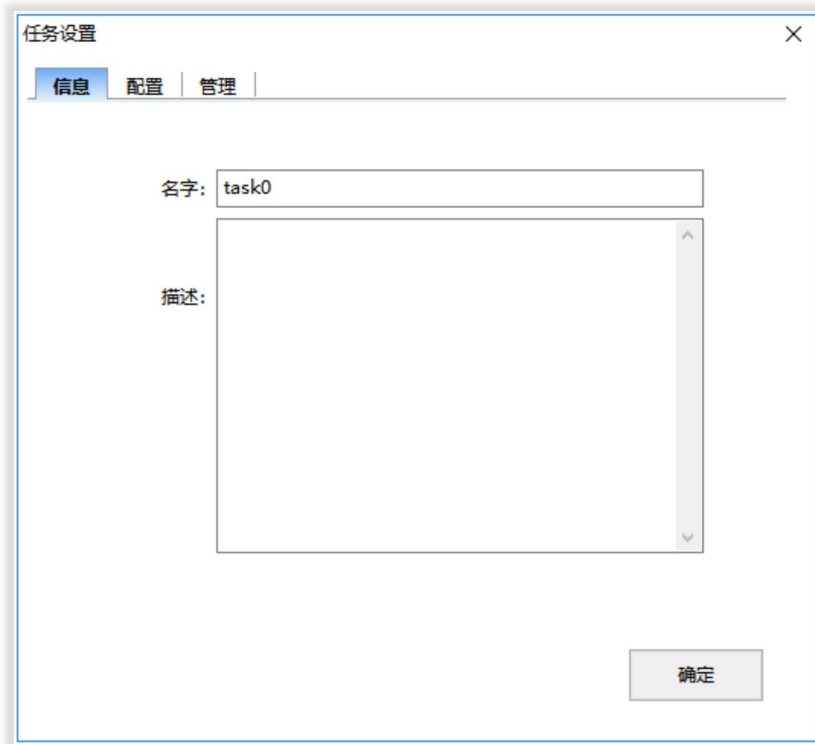


图 3- 92 任务信息

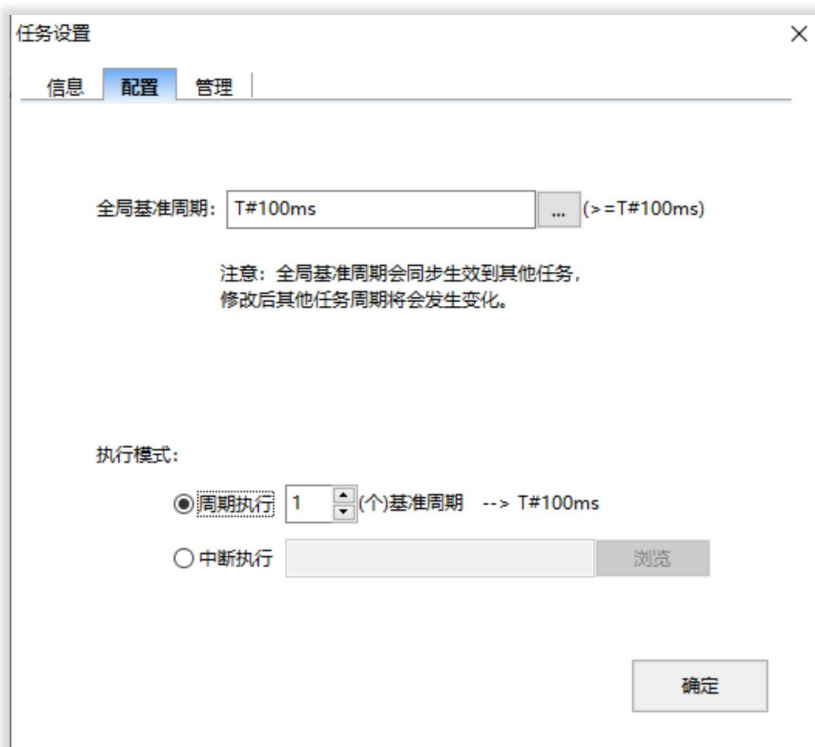


图 3- 93 任务配置

- 描述：对该任务进行注解。
- 执行模式：任务的执行模式，各类型任务的特点可参见表 3-4 中的说明。
 - 周期执行：每个周期任务可定义不同的执行周期，任务在每个周期中执行一次。
 - 中断执行：当触发源产生上升沿时，执行一次当前的事件任务。
- 周期时间：任务类型选择为周期任务时，需设置任务的全局基准周期值和任务执行时需要几个基准周期，全局基准周期可设置的范围为 100ms~730 天，执行周期个数可设置的范围为 1~100。

3.9.2 添加程序

选中工程项目树中“任务”节点下添加的一个程序，右键鼠标选择“添加”程序，弹出对话框。

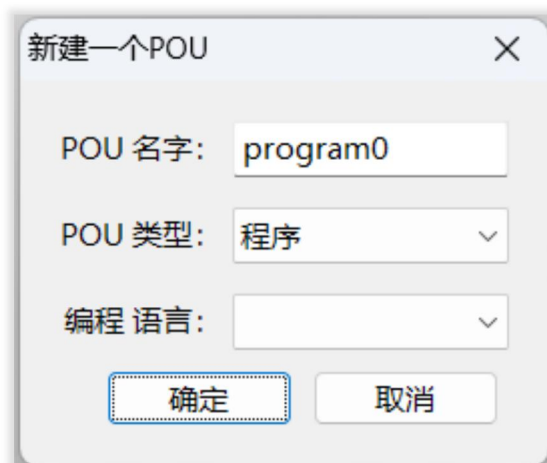


图 3- 94 添加程序

- POU 名称：自定义功能块的名称，名称需以字母开头，只能包含字母、数字和下划线并且长度不超过 32 个字节。
- 编程语言：选择自定义功能块编程语言。目前支持结构化文本、梯形图、功能块。

组态软件支持程序加密功能，通过右键要加密的 POU 程序，点击属性来设置密码

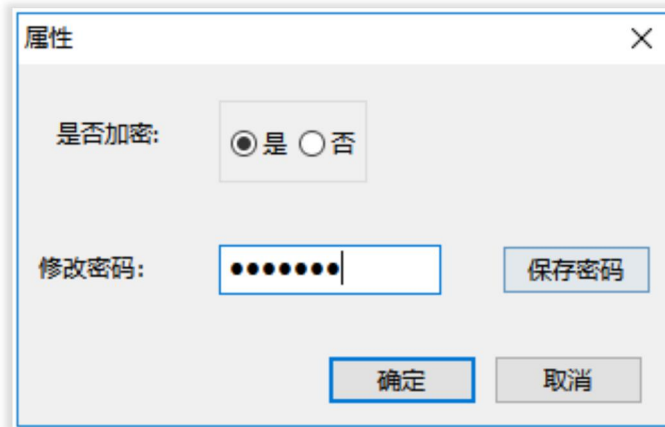


图 3- 95 程序加密

当使用加密功能后，需要重新打开组态软件才能生效

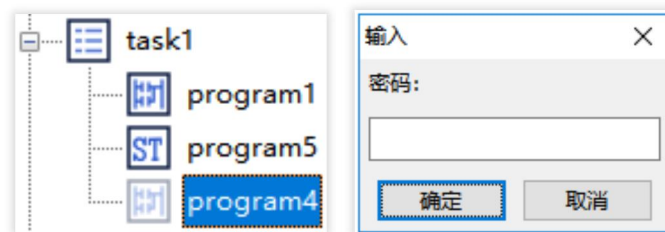


图 3- 96 需输入密码打开加密程序

3.9.3 周期执行任务和中断执行任务触发条件设置

用户可在配置栏中对单个任务周期进行配置，并且可按照执行模式配置成周期执行或者是中断执行。

周期执行：配置成周期执行就是这个任务按照配置的任务周期顺序执行，比如全局基准周期设置成 100ms，执行模式里的周期执行设置成 1 个基准周期，则此任务会按照 100ms 的周期进行运行，时间周期到了之后就一直循环执行下去。



图 3- 97 任务全局基准和执行基准周期

中断执行: 配置成中断执行, 则此任务需要绑定一个外部布尔类型的变量, 通过这个变量的上升沿触发来决定当前任务是否执行, 类似于绑定一个外部引脚来触发任务执行。



图 3- 98 任务中断模式

3.9.4 调整程序执行顺序

如果一个任务中存在多个 POU 程序，那么用户可在任务管理页面中对 POU 程序的执行顺序进行调整，通过选中程序后的上升和下降箭头来调整执行顺序。

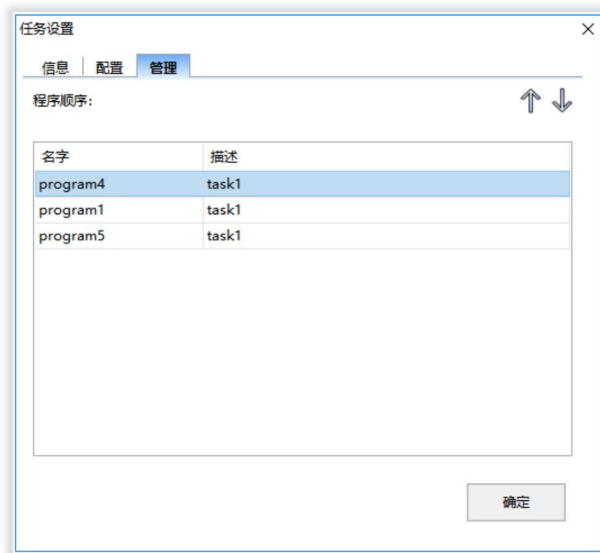


图 3- 99 调整任务执行顺序

3.10 查找和替换

在菜单栏中有“查找”、“查找下一个”、“查找前一个”、“在程序中替换”、“在项目中搜索”，5 个按钮。

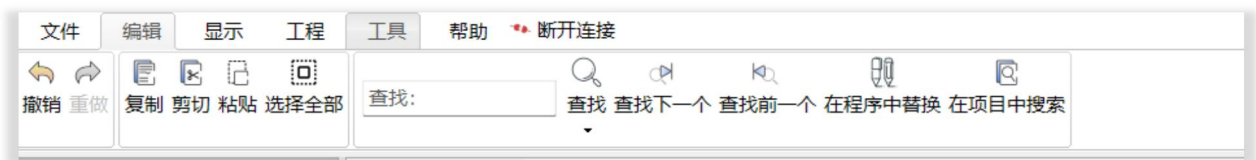


图 3- 100 查找和替换

查找指在当前界面中根据输入“关键字”进行查找，鼠标左击按钮  的下拉框来选择查找的方式。当查找到符合输入的关键字时，当前界面上会高亮显示。

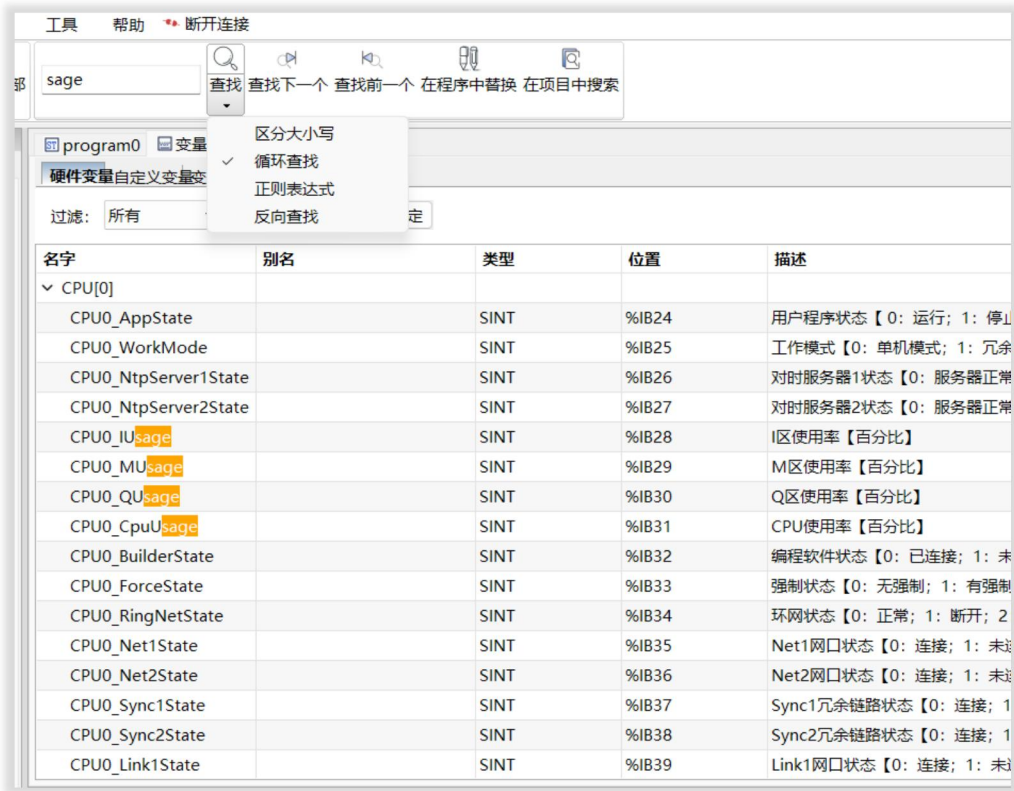


图 3- 101 菜单栏查找功能

查找下一个与查找前一个指在当前界面中根据输入“关键字”单个进行查找，当查找到符合输入的关键字时，当前界面上会高亮显示。



图 3- 102 高亮显示查找内容

在程序中替换指在用户程序中可以进行批量替换。



图 3- 103 替换查找内容

在项目中搜索指全局搜索，会在整个项目下根据检索内容进行匹配，匹配结果在界面最下方显示。

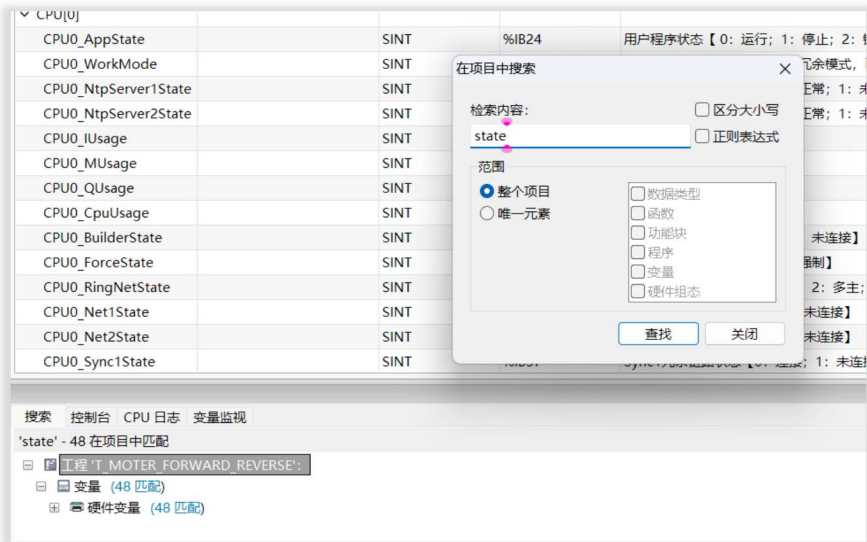


图 3- 104 全局搜索

展开匹配结果后，对搜索的结果双击，会自动跳转至对应结果。



图 3- 105 在搜索栏中高亮显示

3.11 变量声明

在自定义函数、自定义功能块和程序中调用的变量需要在上方的变量表中先声明，否则会编译失败。可以在变量表中添加本地变量，也可以添加外部变量。在图形化编程语言梯级图、功能块图中支持直接在输入/输出双击绑定未声明的变量，若该变量为已存在的外部变量，则会自动在变量表中生成，若该变量不存在，则会提示是否创建此变量。

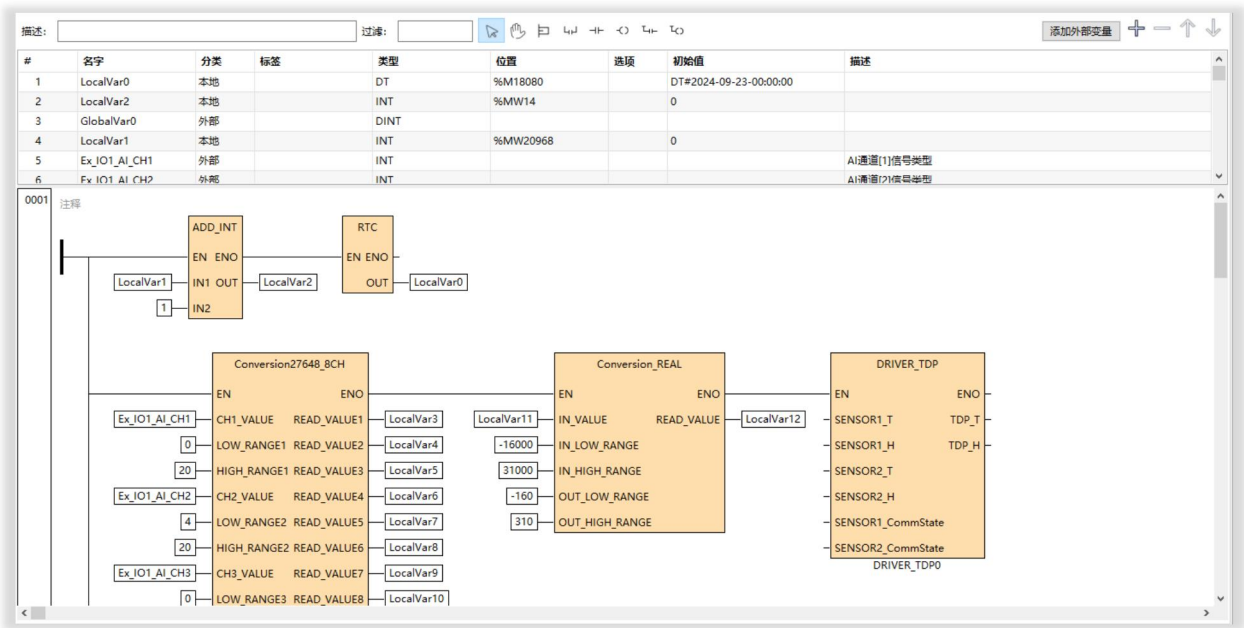


图 3- 106 编程界面变量表

3. 12 布局排布

本节介绍功能块图布局整理的两种方式。

3.12.1 对齐

仅在功能块图中该功能生效，可对已布局的功能块、变量位置进行对齐排布。多选需要对齐的功能块图或者变量，右键“对准”，选择期望的对齐方式，即可完成选中对象的对齐操作。

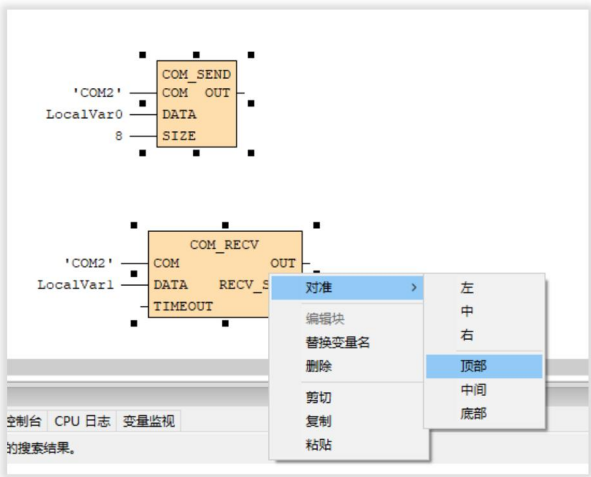


图 3- 107 功能块图对齐

3.12.2 重排

仅在功能块图中该功能生效，可对当前页面已布局的变量位置进行合理化排布，重新排布至与功能块输入/输出连线最短位置。

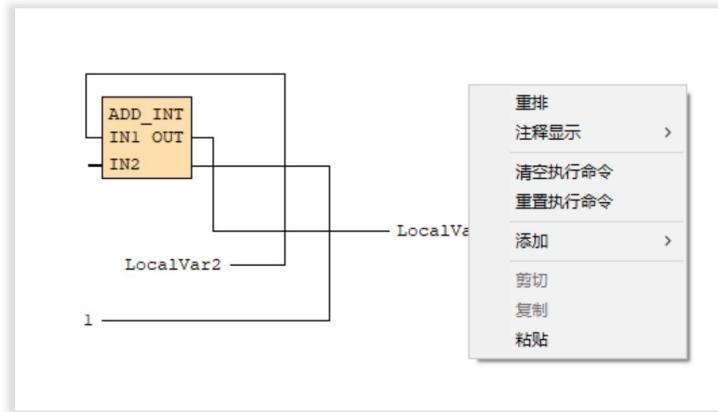


图 3- 108 功能块图重排前

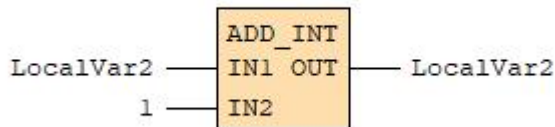


图 3- 109 功能块图重排后

3.13 联机

本节主要介绍编译、联机调试、组态备份及注意事项等。

3.13.1 编译

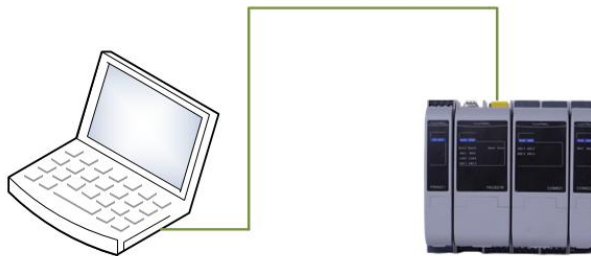
在联机前需要先保存并且编译工程，选择工程菜单栏的“构建”按钮进行程序编译工程。编译完成后输出窗口可查看编译成功等信息。同时选择菜单的“编译信息”按钮可查看最后一次成功编译时间及各内存区的使用情况，如下图所示。



图 3- 110 查看编译信息

3. 13. 2 选择联机设备

- ◆ **建立以太网硬件通讯连接：**通过网线连接操作站与 PLC 控制器。



- ◆ **设置操作站IP地址：**与PLC控制器设置成同一网段。
- ◆ **连接选择对应网段：**在工程菜单栏中点击连接，选择对应网卡的对应网段进行连接，此时出现控制器型号和IP地址说明网络连接配置通讯正常。

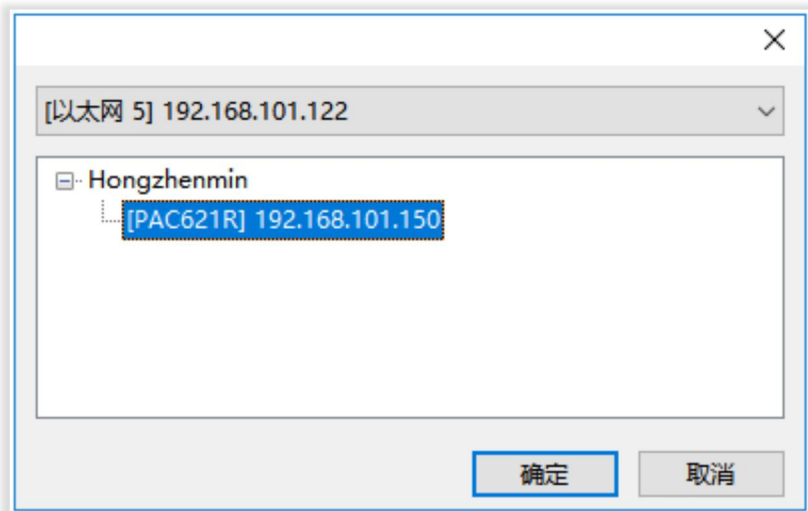


图 3- 111 选择需要连接的设备

3. 13. 3 联机选项

点击连接按钮，支持在线或者离线下载

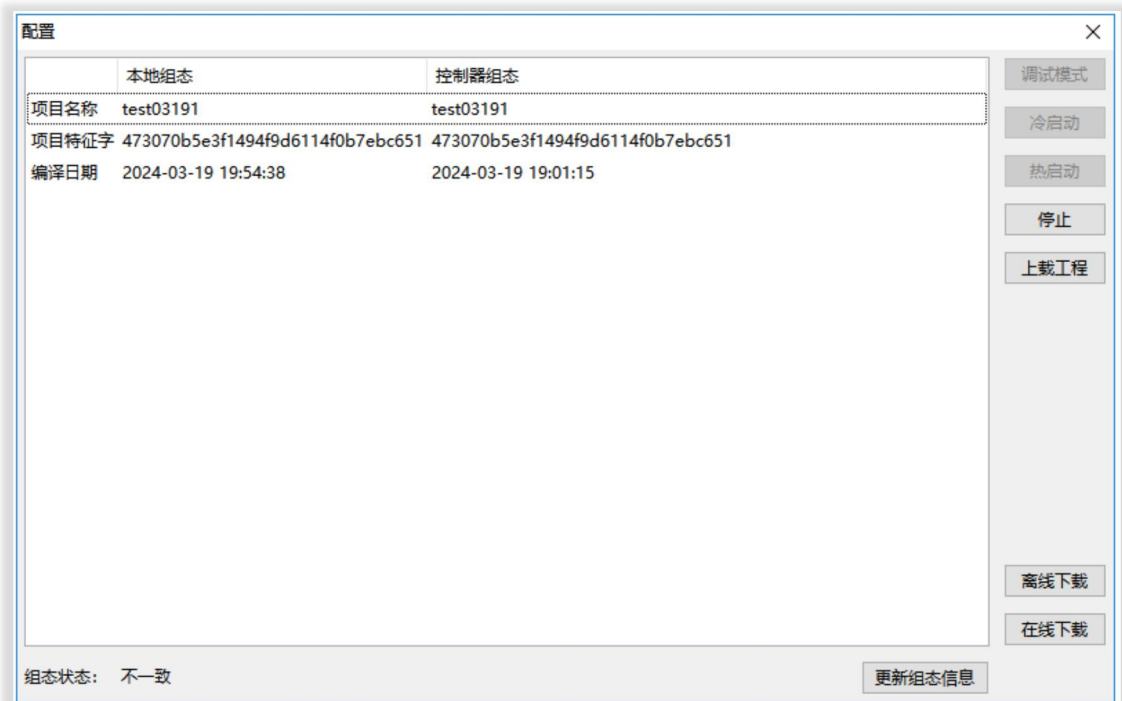


图 3- 112 联机配置

- **更新组态信息**

刷新界面中显示的组态信息。

- **调试模式**

详见3.11.4 联机调试章节。

- **冷启动**

所有变量都按照组态中设置的初始值，程序按照组态中配置的顺序，从初始开始，类似于上电第一次启动。

- **热启动**

所有变量都按照暂停前值，程序从暂停处开始，继续执行。

- **离线下载**

离线下载时，软件将组态完整的下载到控制器中，下载完成后，控制器中的各变量初始化为初值。

- **在线下载**

在线下载时，软件将控制器中的组态上载，然后将上载的内容与将要下载的内容进行对比，只下载差异部分的组态到控制器中。下载完成后，控制器中发生修改的变量从初值开始运行，未修改的变量将继续在实时值上运行。

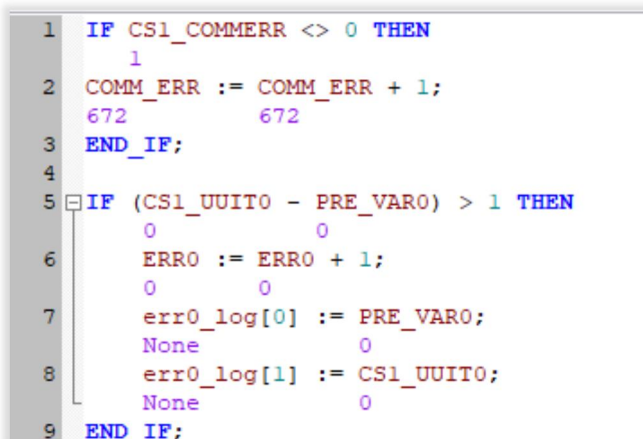
- **上载工程**

可将控制器中的组态上载到本机进行保存（上载的组态为完整的工程组态）。

3.13.4 联机调试

当上下位机组态一致时，在界面中点击“联机调试”按钮即可进行联机调试。

联机调试状态下，可以实时显示变量的数据。



```

1 IF CS1_COMMERR <> 0 THEN
2   1
3   COMM_ERR := COMM_ERR + 1;
4   672      672
5   END_IF;
6
7 IF (CS1_UUIT0 - PRE_VAR0) > 1 THEN
8   0      0
9   ERR0 := ERR0 + 1;
10  0      0
11  err0_log[0] := PRE_VAR0;
12  None    0
13  err0_log[1] := CS1_UUIT0;
14  None    0
15 END IF;

```

图 3- 113 ST 语言程序显示实时值

在工具栏中点击“添加监视”按钮弹出添加变量监视窗口，在窗口中选择需要监视的变量，双击添加后可在底部“变量监视”页签中进行观察和调试。

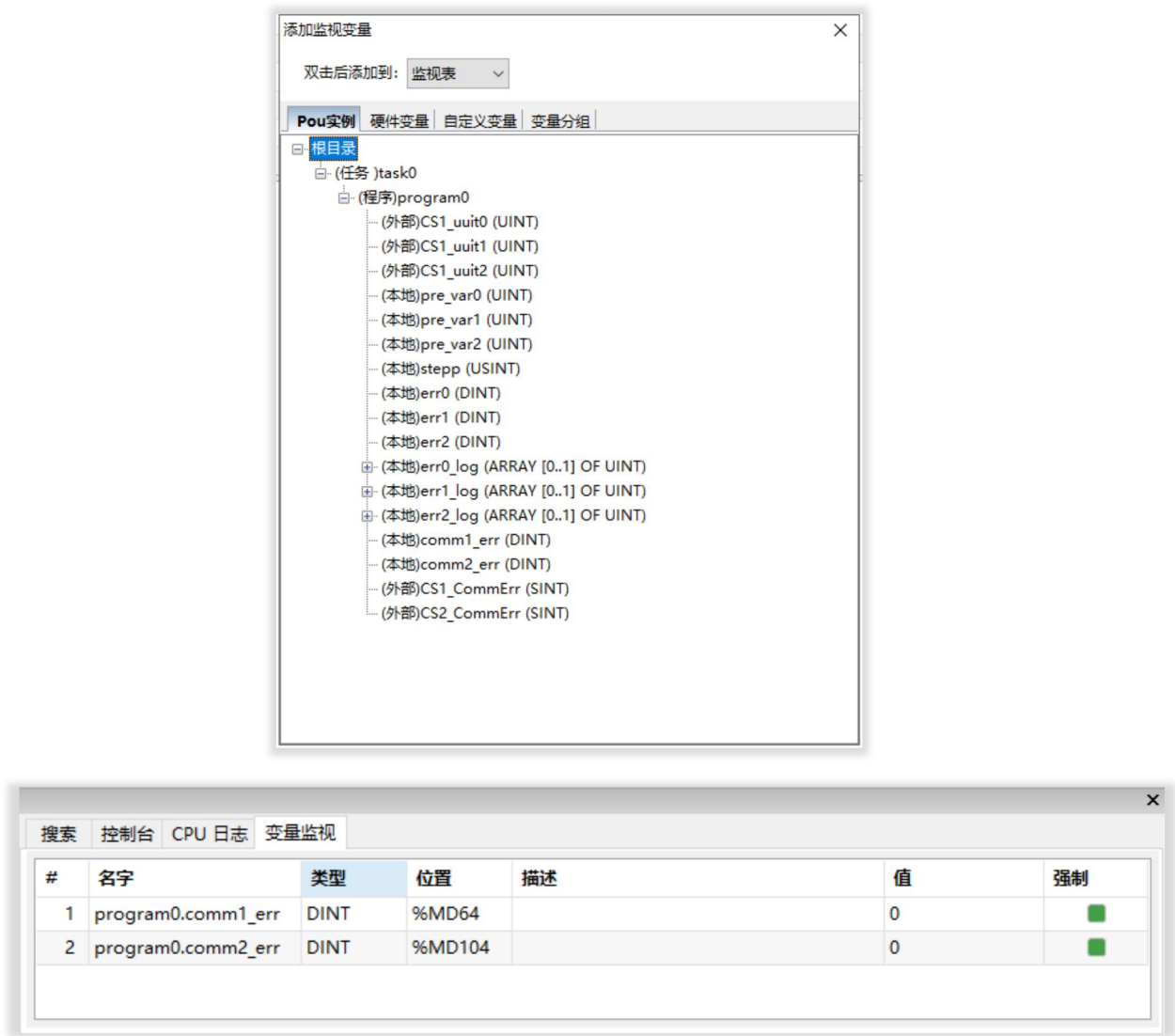


图 3- 114 添加变量至变量监视栏

联机调试状态下，双击“变量管理”节点，进入变量管理界面，可修改自定义变量的值，进行调试。

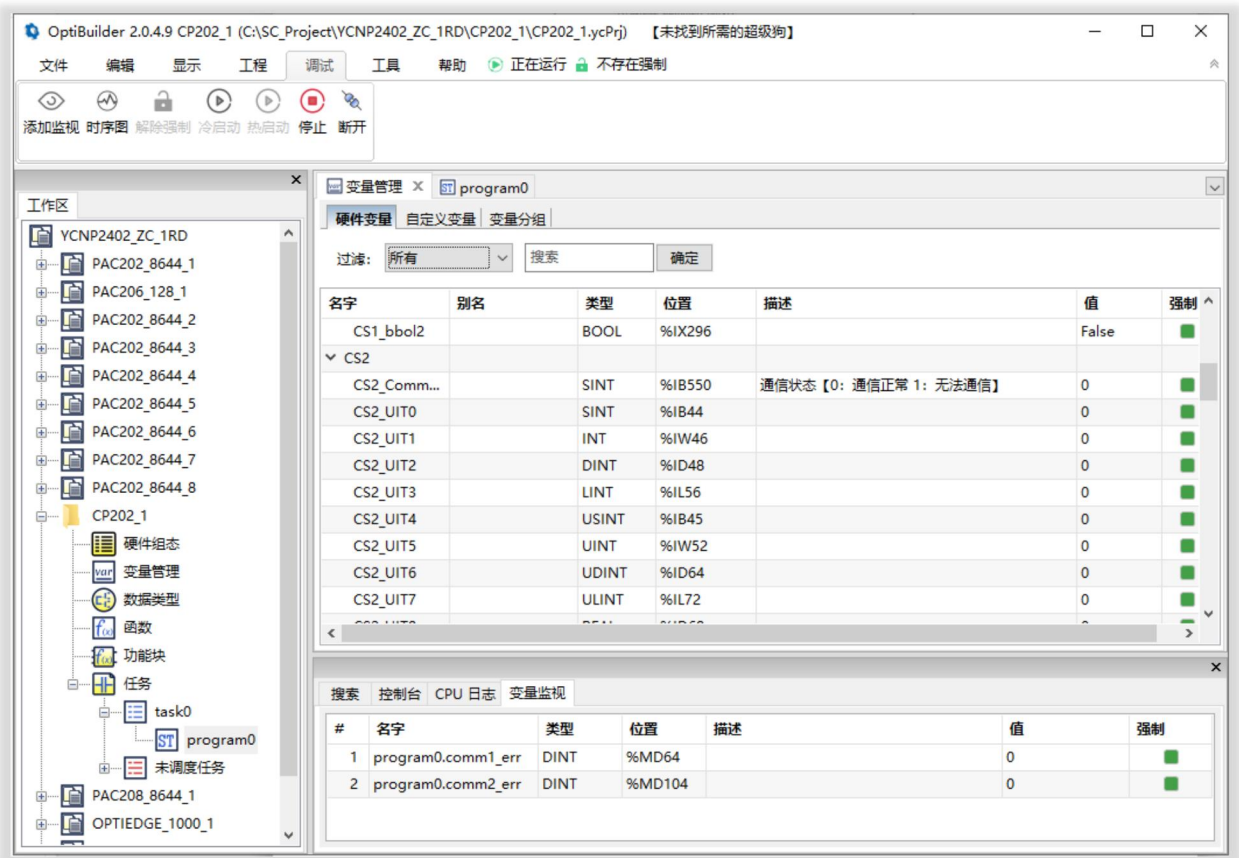


图 3- 115 查看变量实时值

选择硬件变量页面，双击强制标志可以对硬件位号进行强制置值。 同时在调试工具栏中提供“解除强制”按钮，可取消对所有变量的强制设置。

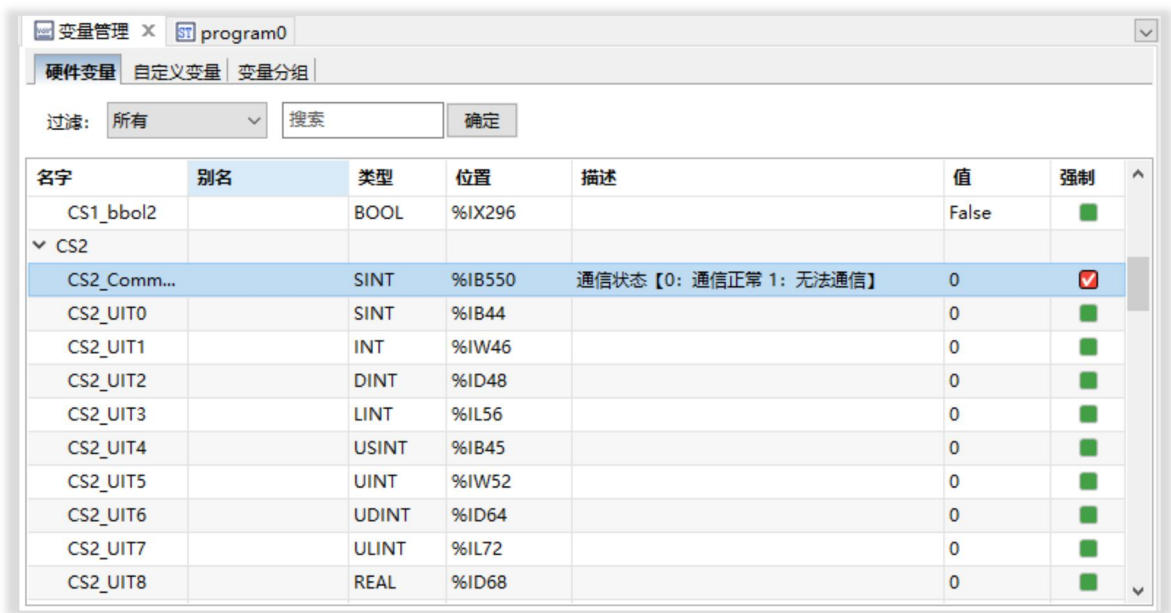


图 3- 116 强制变量值



图 3- 117 解除强制

在工具栏中点击  “时序图” 按钮弹出时序图窗口， 在窗口中选择需要监控时序的变量， 双击添加后可右侧进行观察。

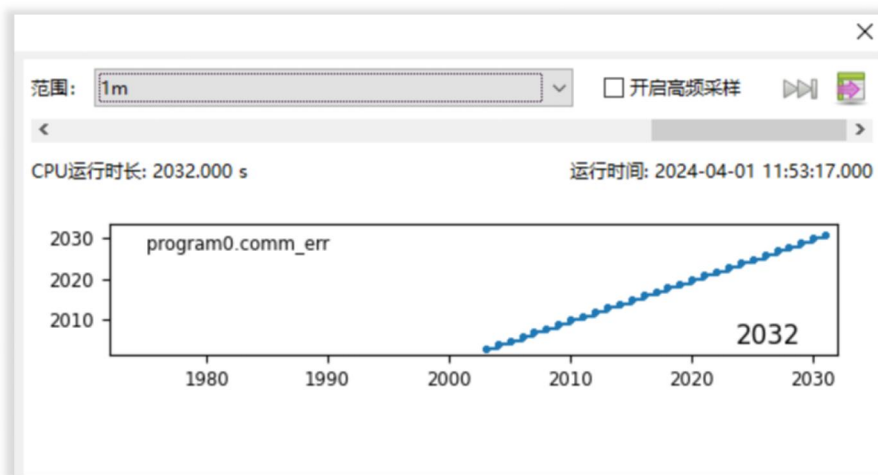


图 3- 118 时序图

3. 13. 5 多工程调试

当项目中含有多个工程时，可将当前联机调试中的工程添加至新窗口监视，关闭该工程调试，打开其他工程进入联机调试后也添加至新窗口监视，即可在全局调试面板对多个工程中的变量进行同步调试。



图 3- 119 新窗口监视

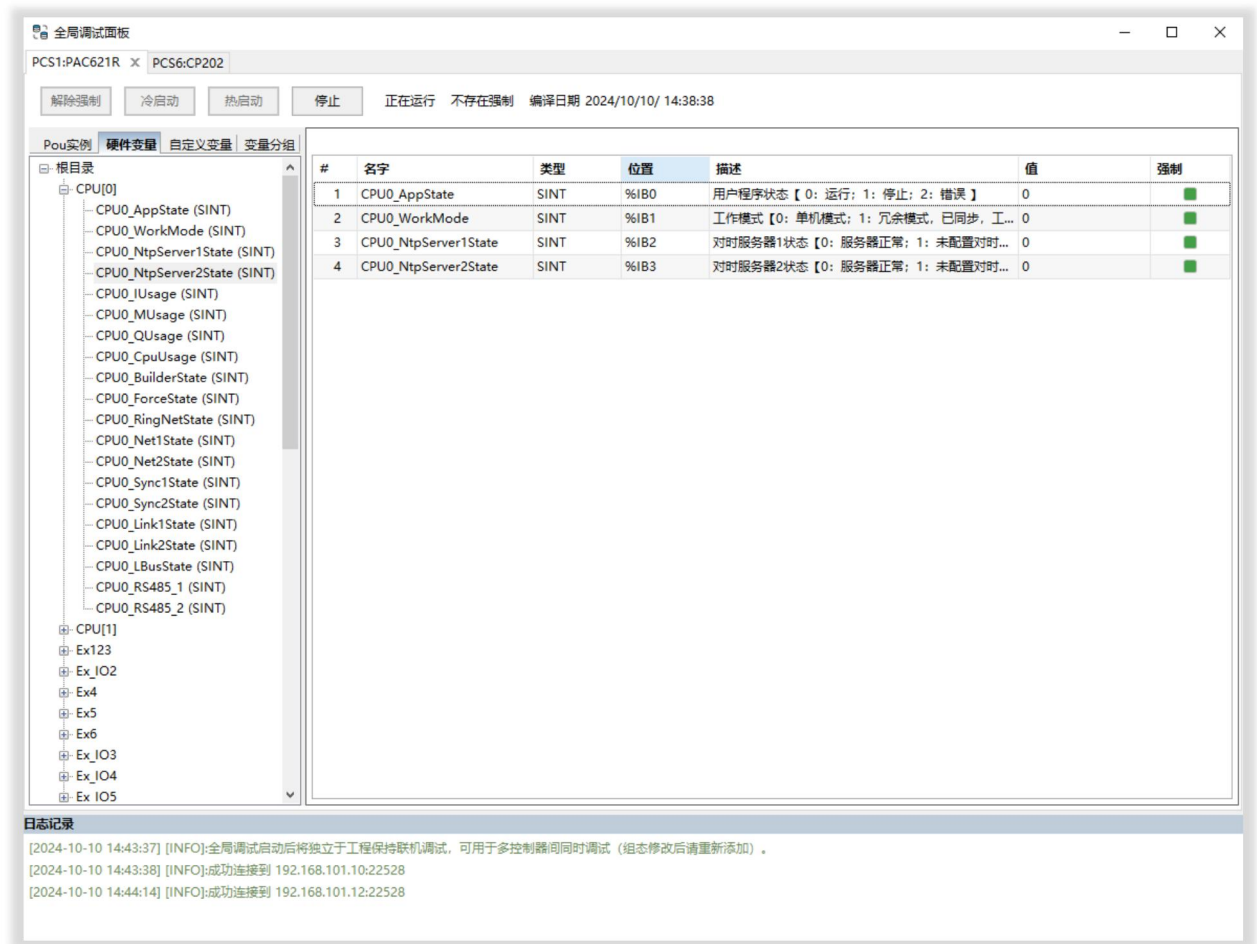


图 3- 120 全局调试面板

3. 13. 6 注意事项

联机后各功能可使用的必要条件如下表所示。表中：

- 工程特征字：工程创建的时候即创建工程特征字，工程的修改编译均不影响工程特征字。
- 组态特征字：工程组态的每次编译即生产新的组态特征字。

功能	必要条件
离线下载	本地组态已编译
在线下载	本地组态已编译 本地组态与控制器组态工程特征字相同 本地组态与控制器组态组态特征字不同
组态上载	无
联机调试	本地组态已编译 本地组态与控制器组态工程特征字相同 本地组态与控制器组态组态特征字相同

4 设备管理

OptiBuilder 软件自带设备管理功能，可以实现 IP 扫描和 MAC 扫描，方便用户快速获取设备的信息，能对设备的工作状态进行准确的判断。本节主要介绍 IP 扫描和 MAC 扫描。

在菜单栏目中的工具目录下，鼠标点击“设备管理”按钮，弹出新的设备管理界面

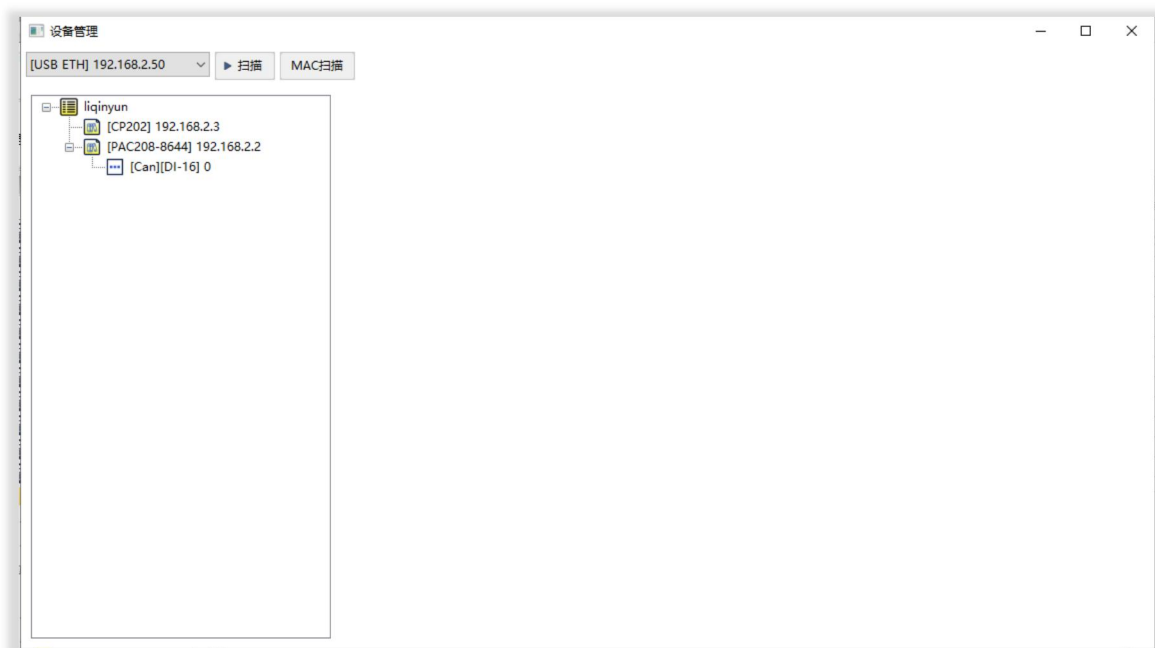


图 4- 1 设备管理界面

4.1 IP 扫描功能

在设备管理界面中选择 PLC 控制器相同网段网卡，可以直接将已连接网络的控制器显示出来，也可以点击“扫描”按钮，组态软件会持续进行扫描

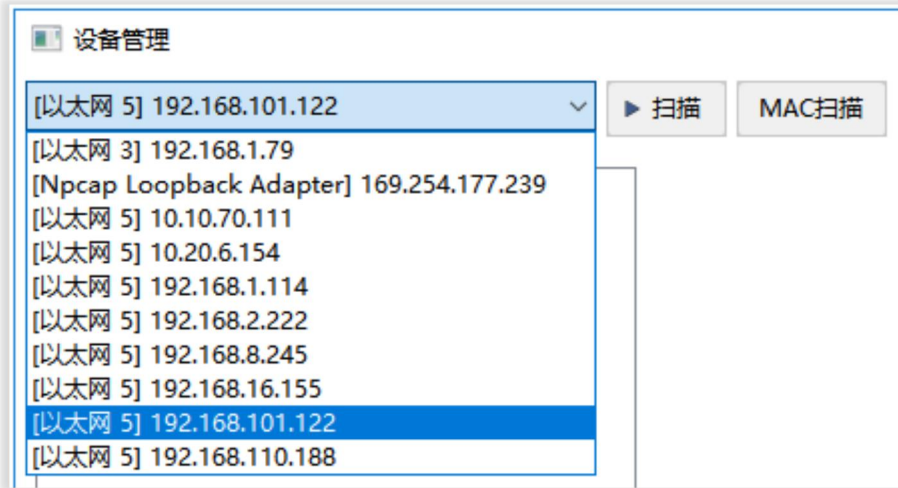


图 4- 2 选择网卡

选择与控制器处于同一局域网 IP 地址的网卡，可以显示扫描到的控制器

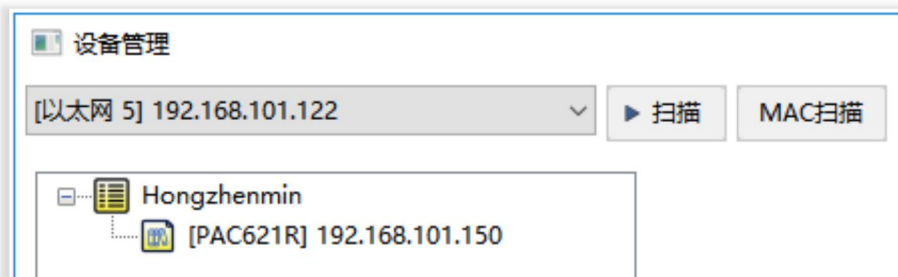


图 4- 3 扫描到设备

双击扫描到的控制器模块可以显示 PLC 控制器的名称、设备厂商、固件版本等信息

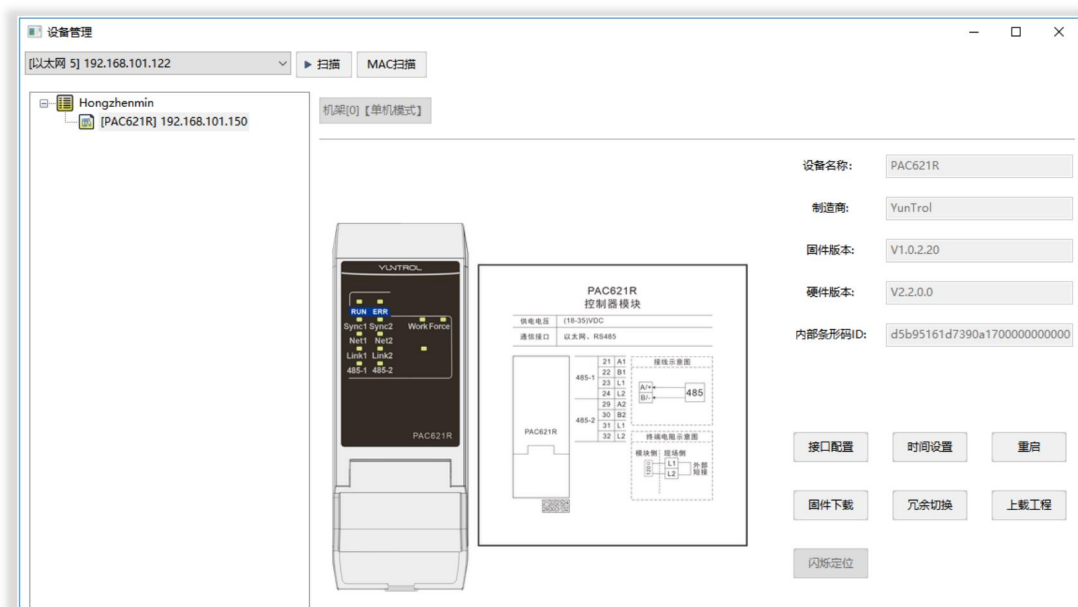


图 4- 4 设备信息

4.2 MAC 扫描

可通过网线直连的方式扫描系统内的所有以太网相关设备。点击设备管理界面上的“MAC 扫描”按钮，可以对 MAC 设备查看目标网卡下所有 MAC 设备情况，双击扫描到的设备，可以修改设备的 IP 地址，子网掩码和网关地址。

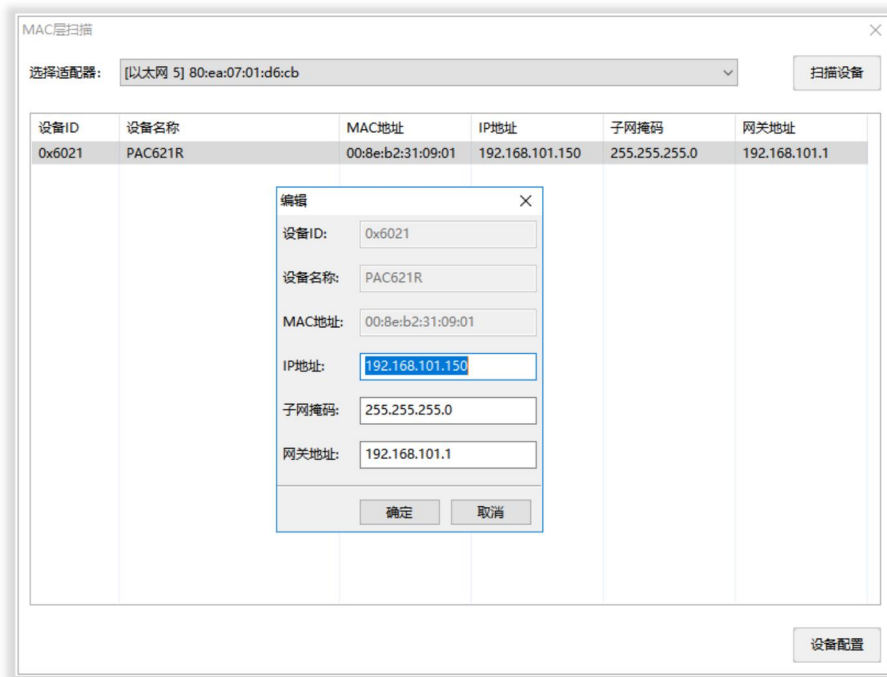


图 4- 5 MAC 扫描界面

4.3 时间设置

本章包含设备时间（RTC）设置和时间同步设置的说明。

控制器模块RTC 时钟默认时间为 2016.3.1 日 00：00：00，现场应用的控制器上电后须重新校对设备时间并进行时钟同步设置。

时钟同步设置过程中，只能在控制系统中设置一个操作节点作为时钟服务器。

4.3.1 设备时间设置

通过 IP 扫描功能扫描到控制器模块后，点击扫描到的控制器模块，点击右下方“时间设置”按钮，弹出时间设置对话框。点击“计算机时间”可以与本地计算机时间同步，点击“保存”设置设备时间。也可以在“设备时间”框中手动修改控制器模块时间，保存生效。

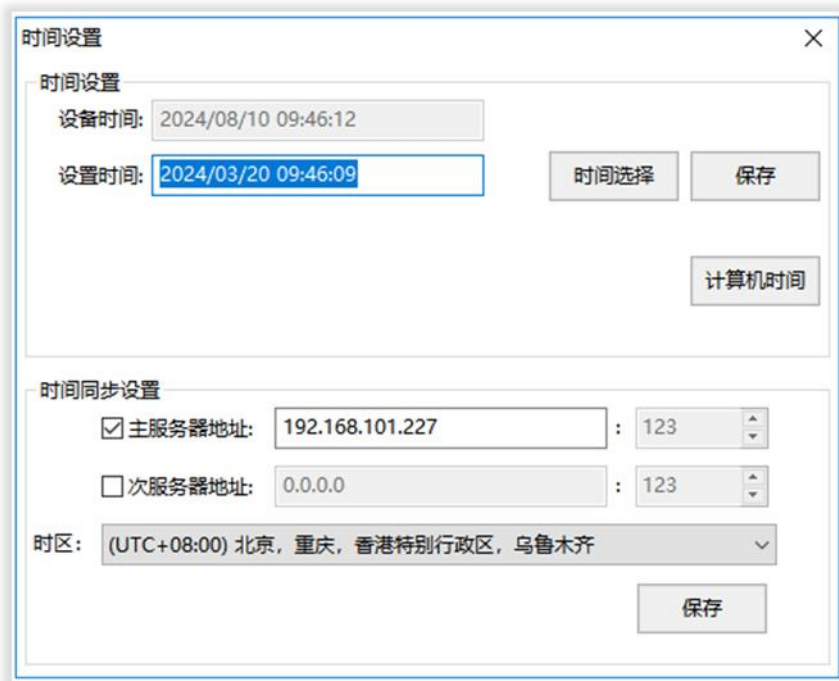


图 4- 6 时间设置

4. 3. 2 时钟同步设置

通过 IP 扫描功能扫描到控制器模块后，点击扫描到的控制器模块，点击右下方“时间设置”按钮，弹出时间设置对话框。在时间同步设置栏目下主服务器地址前打勾，在主服务器地址框内填写同步的服务器 IP 地址



图 4- 7 时钟同步

4.4 冗余切换

通过 IP 扫描功能扫描到控制器模块后，点击扫描到的控制器模块，点击右下方“冗余切换”按钮，可以实现 2 个控制器主备切换，切换后主控制器 Work 灯常亮。

5 实现监控

在 OptiBuilder 组态完成后可通过 OptiVIEW 软件或者通过支持 MODBUS 协议的设备实现数据的监控。

5.1 通过 OptiVIEW 实现监控

OptiVIEW 软件可通过添加/配置 RCP 驱动后直接读取相关控制器中的数据。

详情请参阅 OptiVIEW 相关说明书。

5.2 通过其他监控软件实现监控

本节主要介绍如何通过使用 ModbusRtu 驱动和 ModbusTcp 驱动实现监控。

5.2.1 使用 ModbusRtu 驱动

5.2.1.1 ModbusRtu 通信组态

添加完控制器后，在硬件组态中选择需要通讯的串口，对串口的各项参数进行具体的组态。

1) 控制器通信端口设置

例如设置串口 1 为“从站”模式，并设置通信参数与监控端的串口通信参数匹配，如下图所示。

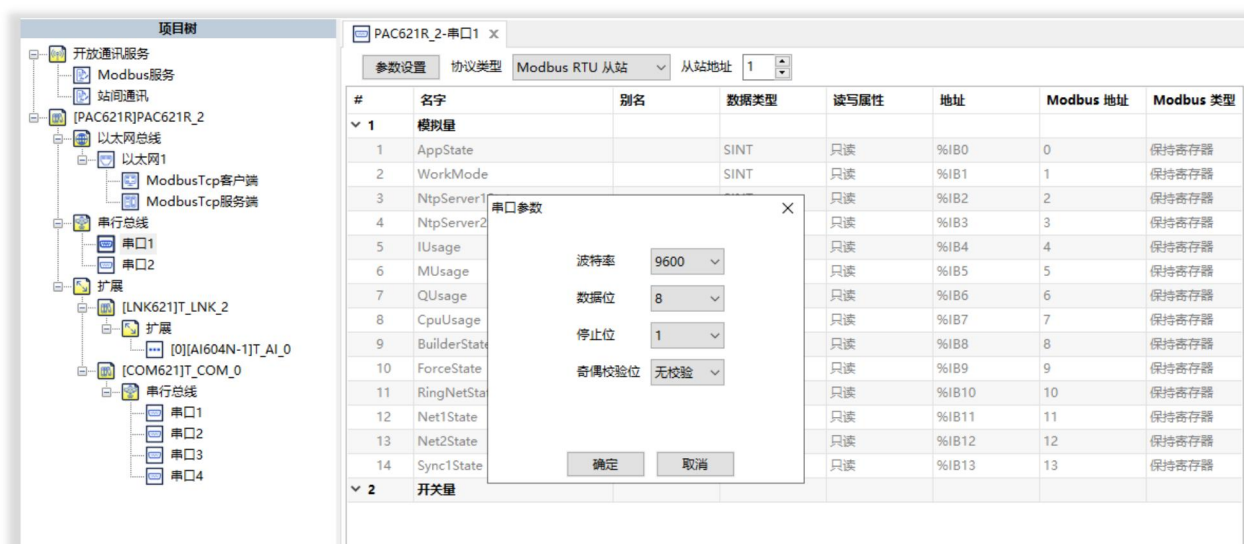


图 5-1 参数设置示意图

2) ModbusRtu 开放位号配置

模块作为从站使用时，支持的 ModbusRtu 命令有：FC01（读线圈状态）、FC03（读保持寄存器）、FC05（写单个线圈）、FC06（写单个寄存器）、FC15（写多个线圈）、FC16（写多个寄存器）。

5.2.1.2 通过 ModbusRtu 协议访问控制器数据

支持 ModbusRtu 协议的设备可以通过 ModbusRtu 命令访问对应 Modbus 地址的控制器内存数据。

5.2.2 使用 ModbusTcp 驱动

5.2.2.1 ModbusTcp 通信组态

添加完控制器后，在硬件组态中选择 ModbusTcp 服务端，设置服务端的端口号（默认 502）。

1) 控制器通信端口设置

设置通信端口号与监控端的客户端通信参数匹配，如下图所示。

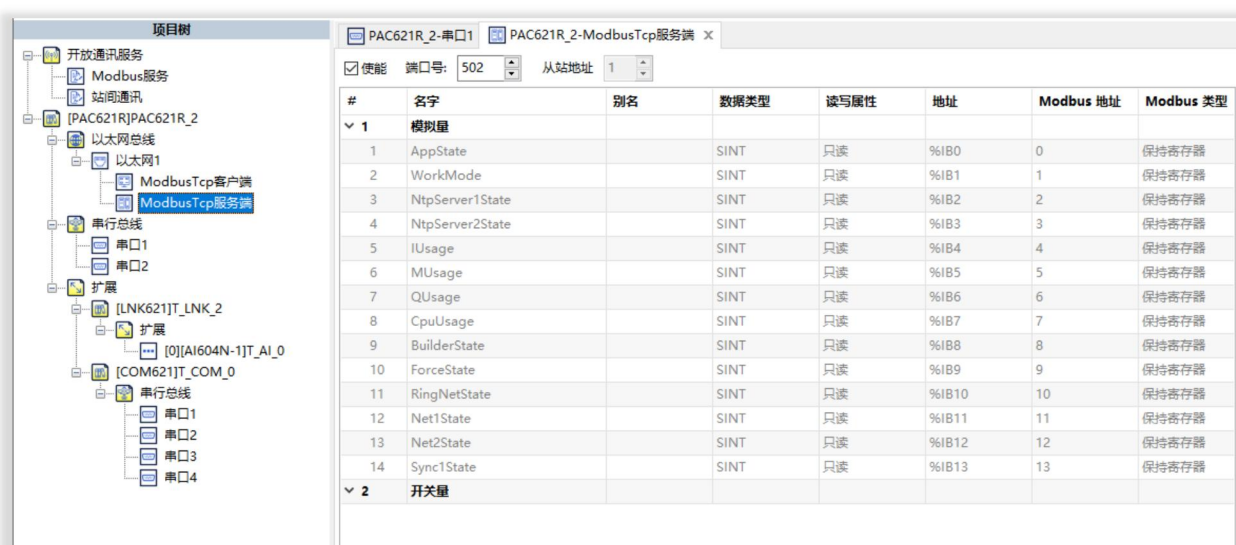


图 5-2 参数设置示意图

2) ModbusTcp 开放位号配置

模块作为服务端使用时，支持的 ModbusTcp 命令有：FC01（读线圈状态）、FC03（读保持寄存器）、FC05（写单个线圈）、FC06（写单个寄存器）、FC15（写多个线圈）、FC16（写多个寄存器）。

5.2.2.2 通过 ModbusTcp 协议访问控制器数据

支持 ModbusTcp 协议的设备可以通过 ModbusTcp 命令访问对应 Modbus 地址的控制器内存数据。

6 编程指南

本章节介绍 FBD（功能块图）、ST（结构化文本）、LD（梯形图）语言编程的数据类型、编程规则等内容。

6.1 FBD 语言编程

FBD 编辑器将基本的功能块和信号（变量、位号）组成功能块图。图形内可以自由放置基本元素和文本。

软件提供了部分预定义的功能块库，功能块库包括标准功能类型、附加功能类型：

- 标准功能块包括：触发器、计数器、计时器等。
- 附加功能块包括：获取/设置时间、PID、斜坡、微分、积分等。

除了预定义的功能块外，还包括多种函数：

类型转换、取绝对值、赋值运算、算术运算、移位运算、按位运算、对数运算、三角运算、比较函数、选择函数、取极值、限幅函数、时间运算、字符串操作、日志操作、自由串口、Modbus 等。此外用户还可以添加自定义功能块。

用户编辑的自定义功能块被放在自定义功能块库中。自定义功能块被放入自定义功能块库后，可以被 ST、FBD 和 LD 语言编辑器使用。

6.1.1 FBD 编辑器主要命令一览表

表 6-1 FBD 编程界面主要命令一览表

图标	描述	备注
	选择对象	无
	移动视图	无
	新建一个备注	无
	新建一个变量	无
	新建一个块	无
	新建一个连接	无

6.1.2 功能块

功能块用带有输入和输出的图形框来描绘。输入在图形框的左边，输出在图形框的右边。功能块的名称在图形框的中间显示。功能块的实例名在图形框上方显示。在同一工程内，模块的实例名是唯一的。

所有功能块都可以用一个 EN 输入和一个 ENO 输出控制功能块的运行，默认隐藏，需要打开可以双击功能块，勾选“执行控制”来打开 EN/ENO 功能。

当调用的功能块 EN 值等于 OFF，则不执行该功能块，ENO 值自动设置为 OFF；如果 EN 值等于 ON，则执行该功能块，执行完后，ENO 值等于 ON。

在不需要 EN 的时候，可以隐藏 EN 和 ENO 引脚；EN，ENO 引脚隐藏后功能块始终运行。

以下分别是显示有 EN、ENO 与隐藏了 EN 和 ENO 的模块示意图：

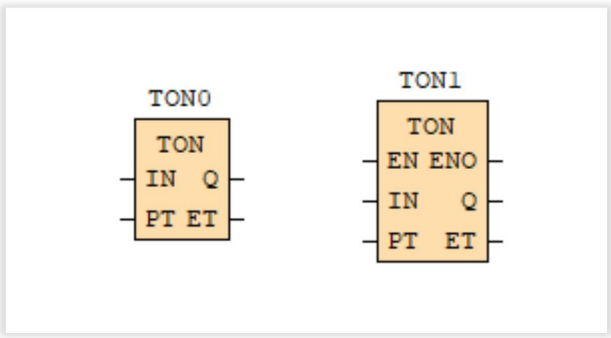


图 6-1 模块示意图

功能块的输入引脚和输出引脚可以连接变量和常数，或者与其他功能块的引脚相连，相连的引脚的数据类型必须相同，并且必须是输入引脚与输出引脚相连。

功能块的输入输出引脚只能连接变量。

功能块的内部参数可以设置变量和常数。

各个功能块的使用说明请参见《功能块使用手册》。

6.1.3 自定义功能块的引脚定义

使用 FBD 自定义功能块时，点击自定义功能块。支持添加 4 种 类型的变量，点击“添加”按钮添加变量，通过双击类型选项选择变量类型，如下图所示的。

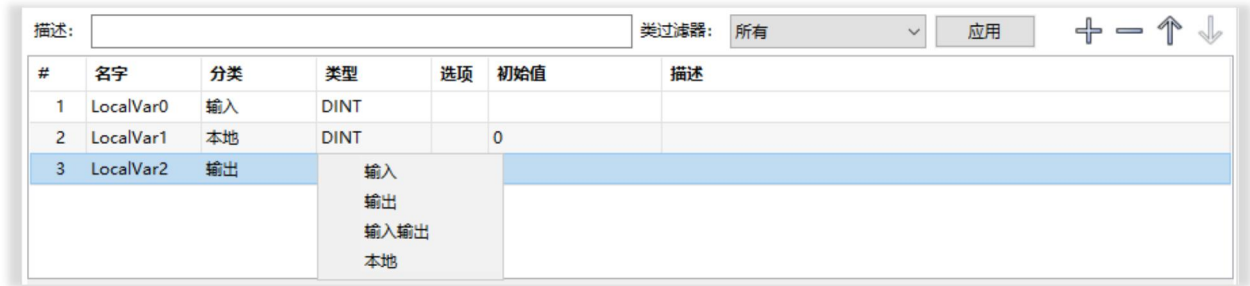


图 6-2 自定义功能块时可添加的变量

- 输入变量：作为自定义功能块的输入引脚。
- 输出变量：作为自定义功能块的输出引脚。
- 输入输出变量：既可以作为自定义功能块的输入引脚又可以作为输出引脚。
- 内部变量：自定义功能块的内部程序使用变量。

6.1.4 程序的私有变量

在需要编写的程序内，点击  添加变量，该变量即作为程序内使用的私有变量（分类为本地）。



#	名字	分类	类型	选项	初始值	描述
1	LocalVar0	本地	DINT		0	
2	LocalVar1	本地	DINT		0	
3	LocalVar2	本地	DINT		0	
4	LocalVar3	本地	DINT		0	

图 6-3 添加程序私有变量

6.1.5 链接

链接是功能块之间的连接。一个功能块输出可以连接多个功能块的输入，相连的输入/输出的数据类型必须匹配。链接允许与其他目标重叠，链接不能形成环路。

6.1.6 执行次序

FBD 区段内的执行次序由区段内的数据流决定。

FBD 的执行次序由上到下、由左到右。

6.1.7 编程原则

- 变量必须先声明再使用；
- 输入输出类型必须一致；
- 功能块和变量可添加注释；
- 不允许通过链路构成环路。
- 对于比较复杂或较大的 FBD 编程，宜先将程序分为简单的程序段，然后再逐段编程。

6.2 ST 语言编程

结构化文本编程是一种灵活、强大且可扩展的编程方法，适用于复杂的控制任务和大规模的 PLC 程序。它提供了更高级的编程功能和更大的自由度，使工程师能够更精确地控制和操作 PLC 系统。结构化文本编程使用类似于高级编程语言的文本语言，如 Structured Text (ST)，这些语言通常基于标准的编程语法和语义规则。

结构化文本编程使用各种语法结构来表示逻辑和控制关系，常见的语法结构包括条件语句（如 IF-THEN-ELSE）、循环语句（如 FOR、WHILE）、函数和过程等。允许定义和使用各种变量和数据类型，支持各种运算和逻辑操作，如算术运算、逻辑运算、比较操作等，支持模块化和可重用性的编程风格，提供了调试和错误处理的机制。用户可以使用调试工具来检查程序的执行过程和变量的值。同时，可以使用错误处理机制来处理程序中的错误和异常情况

以下是一个正弦波发生器的示例工程。

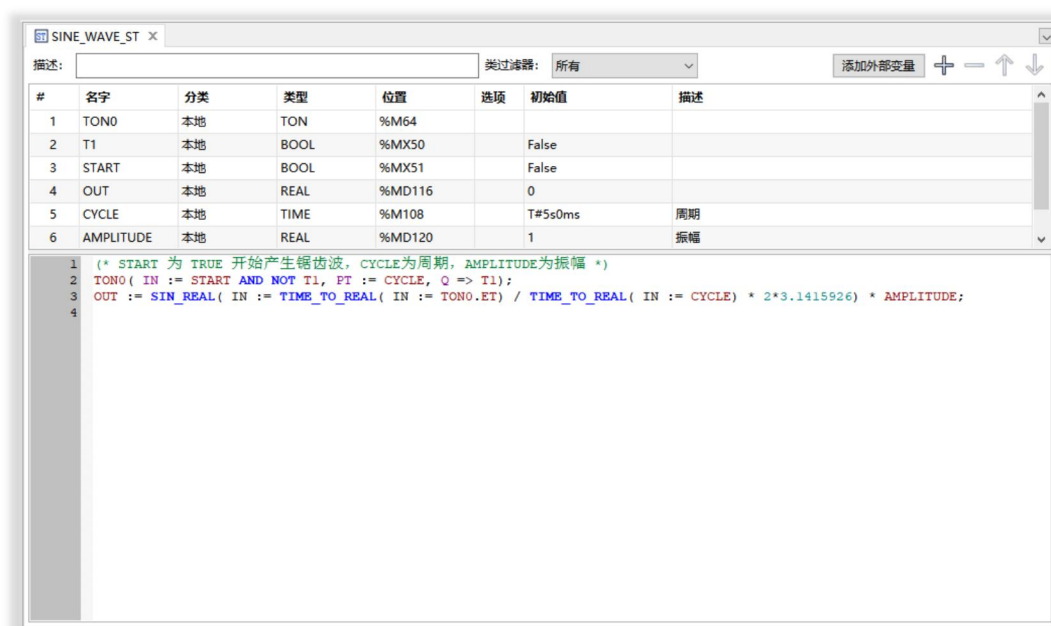


图 6-4 ST 语言编程

6.2.1 ST 语言语法

1. 表达式

表达式为变量、操作符、常量、函数的组合，计算结果为单个值。

表达式的求值按运算符的优先级进行，优先级高的运算符先被处理。相同优先级的运算符按从左到右的顺序执行。

以下为几个合法的表达式举例：

$A + B * (C - 3) + \text{FUNC1}(2, D)$

B1

FUNC1()

变量、函数等标识符的命名必须满足下列条件：

- 以英文字母开头，由英文字母、数字或下划线组成；
- 长度最多为 32 个字节。标识符包括变量、函数、功能块、常数。

2. 关键字

表 6-2 关键字

关键字	描述
CASE...OF...ELSE...END_CASE	CASE 语句
EXIT	终止循环
FOR...TO...BY...DO...END_FOR	FOR 语句
IF...THEN...ELSEIF...ELSE...END_IF	IF 语句
REPEAT...UNTIL...END_REPEAT	REPEAT 语句
RETURN	函数返回
WHILE...DO...END_WHILE	WHILE 语句

3. 运算符

下表中的运算符按运算优先级从高到低列表（9 表示优先级高，1 表示优先级低）

表 6-3 运算符

运算符	描述	类型	优先级
()	表达式运算	无	9
.	取结构成员	无	8
[]	取数组成员	无	8
-	单目负	无	7
NOT	取反	逻辑运算	7
*(MUL)	乘	算术运算	6
/(DIV)	除	算术运算	6

MOD	取余	算术运算	6
+ (ADD)	加	算术运算	5
- (SUB)	减	算术运算	5
>	大于	比较运算	4
>=	大于等于	比较运算	4
<=	小于等于	比较运算	4
<	小于	比较运算	4
=	等于	比较运算	4
<>	不等于	比较运算	4
AND	与	逻辑运算	3
XOR	异或	逻辑运算	2
OR	或	逻辑运算	1

4. 语句

下表为允许的语句列表。赋值语句将“:=”右边表达式的值赋给左边的变量，RETURN 仅用于判断语句或循环语句中。

表 6-4 语句和例子

No.	语句	例子
1	赋值语句	A := B; A := B + 1;
2	函数调用	A := FUNC(P1,P2);
3	RETURN	IF FLAG THEN RETURN; ELSE FLAG := ON; END_IF;
4	IF	IF A > 0 THEN B := 1; ELSIF A > -5 THEN B := 2; ELSE B := 3; END_IF;

5	CASE	<pre> TW = FUNC1(); CASE TW OF 1 : I := 1; 2 : I := 2; ELSE I = 3; END_CASE; </pre>
6	FOR	<pre> J := 10; FOR I := 0 TO 100 BY 2 DO IF I > 50 THEN J := 1; EXIT; END_IF; END_FOR; </pre>
7	WHILE	<pre> J := 1; WHILE J <= 100 AND B1 DO J := J + 2; END_WHILE; </pre>
8	REPEAT	<pre> J := 1; REPEAT J := J + 2; UNTIL J = 101 OR B1 END_REPEAT; </pre>
9	EXIT	<pre> J := 1; WHILE J <= 100 AND B1 DO J := J + 2; IF J >= 50 THEN EXIT; END_IF; END_WHILE; </pre>
10	EMPTY	<pre> FOR I := 1 TO 100 BY 2 DO ; END_FOR; </pre>

5. 常数及注释

常数

ST 语言中常数有以下几种表示方式，见下表。

表 6-5 常数表示

序号	类型	表示 (举例)
----	----	---------

1	布尔型	ON, OFF, TRUE, FALSE
2	浮点类型	1.3 或 1.3E5
3	整型	10 或 8#10 (8 进制的 10) 或 16#EF (16 进制的 EF)

注释

注释由(**)表示，注释内的内容全部采用绿色，注释不允许有嵌套，但是注释内允许有括号，允许多行注释，如下图所示：

```

1  (*注释从这里开始
2  LocalVar0 = 1;
3  到这里结束()*)

```

6. 函数调用

使用 ST 语言时可以调用函数。

函数定义：只有一个输出（变量类型可以自行确定），根据输入可以唯一确定输出。

函数的调用包括函数名加小括号对，括号内为参数，参数间由逗号隔开。

函数的调用规则：

OUT(*BOOL*) := EQ(IN1 := (*ANY*), IN2 := (*ANY*)); (*作为表达式返回值*)
MODBUS_CRC16(DATA := (*ANY*), SIZE := (*UDINT*), CRC16_H => (*BYTE*), CRC16_L => (*BYTE*)); (*作为子程序处理*)。

7. ST 语言程序举例

赋值语句

赋值语句将“:=”右边表达式的值赋给左边的变量。

```

A := B; A := B + C; A := C * D;
A := AND_DWORD( B,C);

```

◆ IF 语句

```
IF (E > (F + 1)) THEN
    AA := BB;
ELSIF (B > G) THEN
    AA := CC;
ELSE
    AA := DD;
END_IF;
```

IF 语句规定了一组语句在规定的逻辑表达式为 TRUE 时执行。当逻辑表达式为 FALSE 时，这些语句不被执行，或在 ELSE 中规定的另一组语句被执行。

◆ CASE 语句

```
CASE A OF
    1:
        AA := BB;
    2:
        AA := CC;
    3:
        AA := DD;
ELSE
    AA := EE;
END_CASE;
```

CASE 语句规定了整数类型的选择项，以及选择项在不同的值时的几组语句组。当选择项等于某个规定的值时，相应的语句组被执行，当没有规定的值符合时在 ELSE 中的语句组将被执行（在 CASE 语句中定义了 ELSE 分支）。

◆ FOR 语句

```
FOR I := 1 TO 100 BY 2 DO
    DOSOMETHING();
END_FOR;
```

在以上 FOR 语句中，I 为控制变量，1 为初始值，100 为终止值，2 为步进值。在 FOR 语句中控制变量的初始值、终止值、步进值必须是相同的整型。步进值缺省为 1。终止条件的判断一开始就进行，当初始值大于终止值时，规定的语句组一次都不会执行。

◆ WHILE 语句

WHILE 条件 DO

语句组

END_WHILE;

条件的判断一开始就进行，如条件一开始就变 FALSE 时，规定的语句组一次都不会执行。

◆ REPEAT 语句

REPEAT

语句组

UNTIL 终止条件

END_REPEAT;

终止条件的判断在语句组执行一次后才进行，所以规定的语句组至少会执行一次。当终止条件成 TRUE 时，循环被终止。

6.2.2 编程助手

ST 语言的编程助手可以对所有的变量、关键字、功能块名称等进行首字母定位。如在进行 ST 语言编程中，输入字母 c 后，即跳出如下图所示的编程助手。

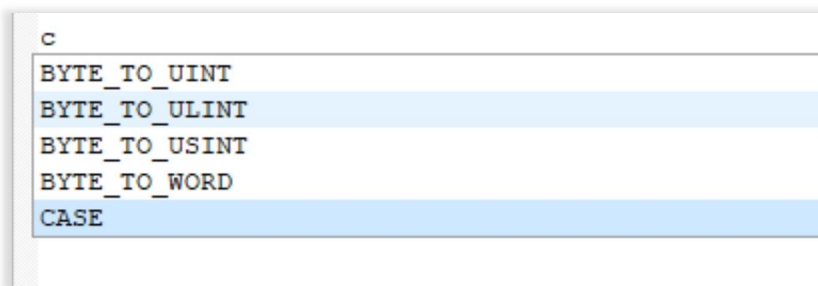


图 6-5 ST 语言的编程助手

OptiBuilder 软件中支持编程助手的有：

- ST 自定义功能块
- ST 用户程序中的代码

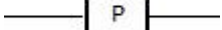
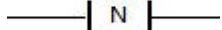
6.2.3 ST 可调用函数

所有支持的库函数以及自定义函数。

6.3 LD 语言编程

6.3.1 LD 编辑器主要命令一览表

◆ 触点：

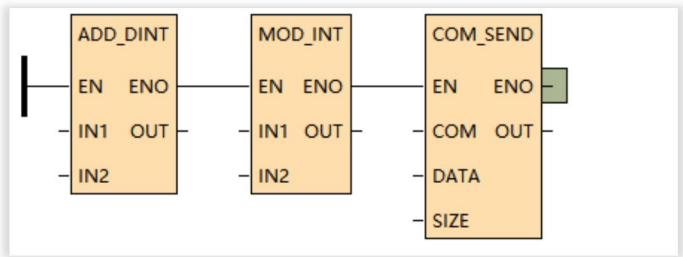
触点类型	标识
 （常开触点）	LocalVar0 
 （常闭触点）	LocalVar0 
 （上升沿触点）	LocalVar0 
 （下降沿触点）	LocalVar0 

◆ 线圈：

线圈类型	标识
LocalVar0 -()- （常开线圈）	LocalVar0 ——()
LocalVar0 -(/)- （常闭线圈）	LocalVar0 ——(/)
LocalVar0 -(S)- （置位线圈）	LocalVar0 ——(S)
LocalVar0 -(R)- （复位线圈）	LocalVar0 ——(R)
LocalVar0 -(P)- （上升沿线圈）	LocalVar0 ——(P)
LocalVar0 -(N)- （下降沿线圈）	LocalVar0 ——(N)

◆ 块：

梯形图编程中支持功能块的插入



6.3.2 触点

触点是 LD 元素，它将状态传送至其右侧的水平链路。这一状态是其左侧水平链路状态与相关变量状态进行布尔操作的结果。触点不改变相关变量的值。

触点类型分为正常、否定、上升沿、下降沿。

正常触点

在正常触点中，如果和触点相关的变量(LocalVar0)为 ON 时，左链路的状态复制到右链路；否则右链路的状态为 OFF。

下图用梯形图和功能块图的方法描述了正常触点：

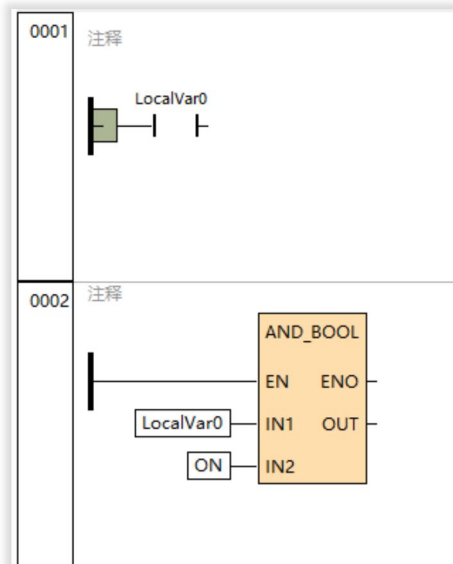


图 6-6 正常触点梯形图和功能块图

否定触点

在否定触点中，如果和触点相关变量(LocalVar0)的状态为 OFF 时，左链路的状态复制到右链路；否则右链路的状态为 OFF。

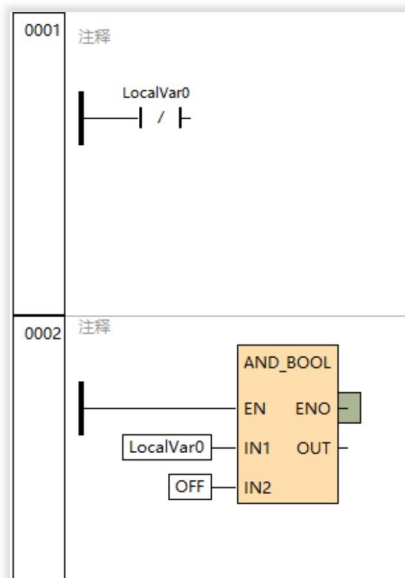


图 6-7 否定触点

否定触点对应于含两个输入的 AND_BOOL 功能，其中一个输入是反相的。

上升沿触点

在上升沿触点中，如果和触点相关 BOOL 变量(LocalVar0)的状态从 OFF 跳变为 ON 时，同时左链路的状态为 ON 的话，则右链路在下一个程序周期为 ON；否则右链路的状态为 OFF。

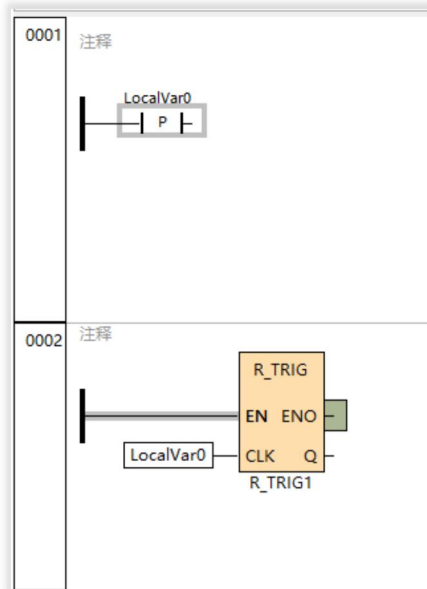


图 6-8 上升沿触点

下降沿触点

在下降沿触点中，如果和触点相关 BOOL 变量(LocalVar0)的状态从 ON 跳变为 OFF 时，同时左链路的状态为 ON 的话，则右链路在下一个程序周期为 ON；否则右链路的状态为 OFF。

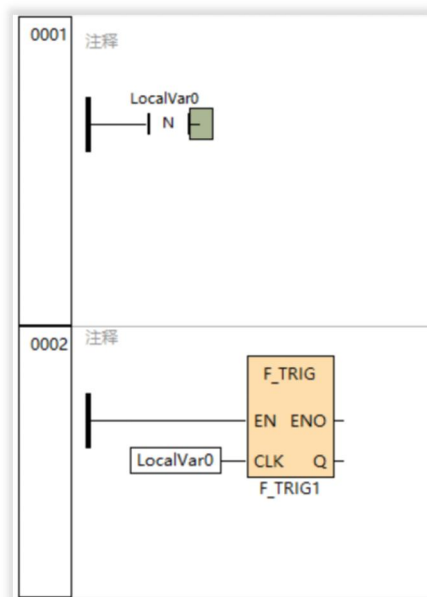


图 6-9 下降沿触点

比较触点

比较触点本质上是普通触点针对比较的一个定向优化，可以选择两个相同类型的变量以及比较符号进行比较，在比较结果为 True 时左链路的状态复制到右链路，否则右链路的状态为 OFF。

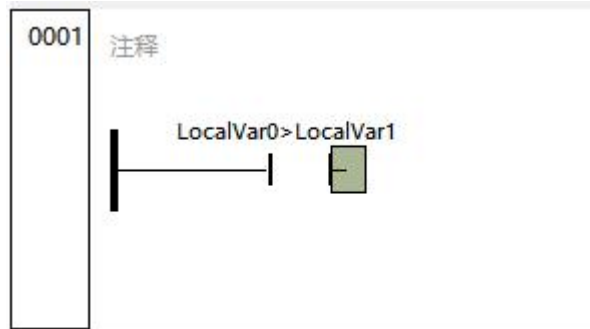


图 6-10 比较触点

6.3.3 线圈

线圈是 LD 元素，它将其左侧的水平链路状态传送至其右侧的水平链路，相关变量的状态将保存。

线圈的类型有正常、否定、设置、重置、上升沿、下降沿。

正常线圈

将左链路的状态复制到相关的布尔变量和右链路。线圈通常跟在触点之后，但线圈之后也可以接触点单元。正常线圈对应于 MOVE 功能。

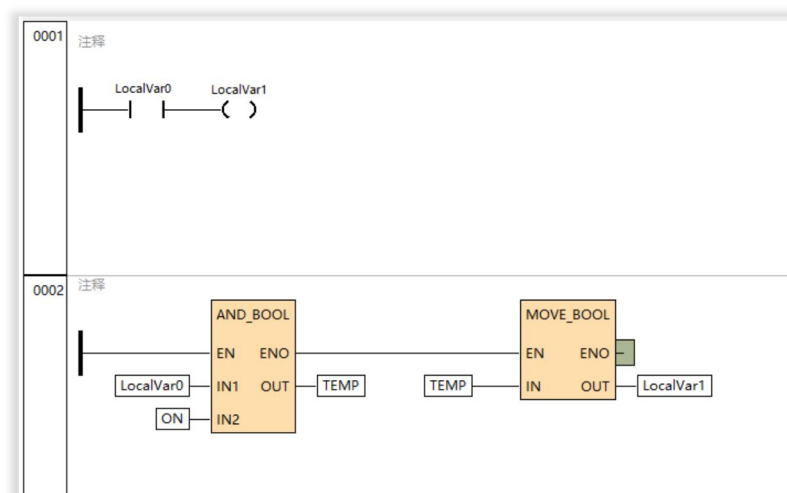


图 6-11 正常线圈

否定线圈

将左链路的状态复制到右链路。左链路的取反状态复制至相关的布尔变量(LocalVar1)。如果左链路为 OFF，则右链路将为 OFF，而相关变量将为 ON。

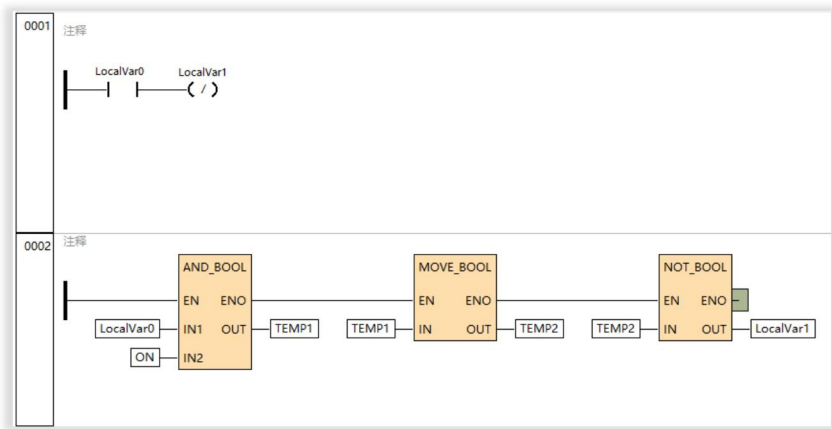


图 6-12 否定线圈

否定线圈对应于带有反相输出的 MOVE 功能。

设置线圈

将左链路的状态复制至右链路。如果左链路为 ON，则相关的布尔变量(LocalVar1)置为 ON；否则相关的布尔变量保持不变直至程序或手动修改其值。相关布尔变量能够借助复位线圈复位。设置线圈 对应于输入固定为 ON 的 MOVE 功能。

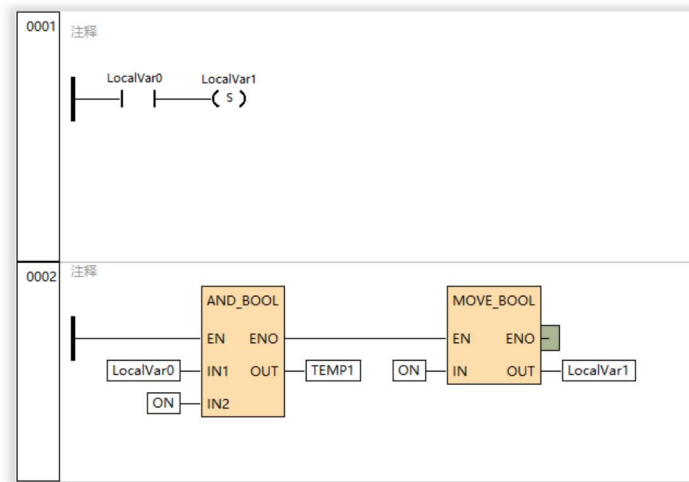


图 6-13 设置线圈

重置线圈

将左链路的状态复制至右链路。如果左链路为 ON，则相关的布尔变量(LocalVar1)置为 OFF；否则 保持不变直至程序或手动修改其值。相关布尔变量能够借助置位线圈置位。重置线圈 对应于输入固 定为 OFF 的 MOVE 功能。

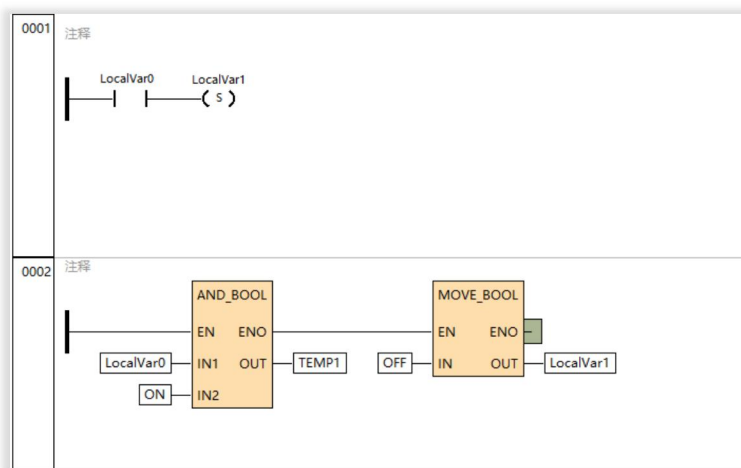


图 6-14 重置线圈

正跳变线圈

将左链路的状态复制至右链路。如果左链路从 OFF 跳变为 ON，则相关的布尔变量(LocalVar1)将在 下一个程序周期内为 ON。正跳变线圈对应于 R_TRIG 功能块。

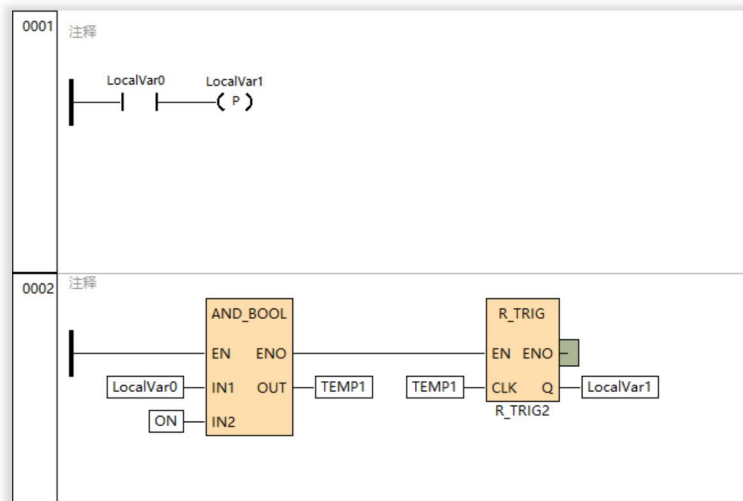


图 6-15 正跳变线

负跳变线圈

将左链路的状态复制至右链路。如果左链路从 ON 跳变为 OFF，则相关的布尔变量(LocalVar1)将在 下一个程序周期内为 ON。负跳变线圈对应于 F_TRIG 功能块。

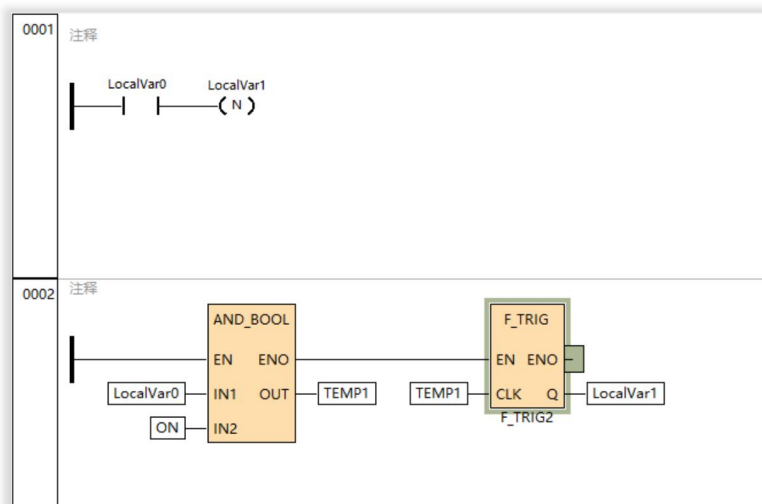


图 6-16 负跳变线圈

6.3.4 执行次序

LD 区段内的执行次序由区段内的数据流决定。

LD 的执行次序从上到下、从左到右。

6.3.5 编程原则

- 变量必须先声明再使用；
- 梯形图的每一逻辑行必须从左边汇流条连接点输入开始；
- 母线连接点的使用次数不受限制；
- 对于比较复杂或较大的 LD 编程，宜先将程序分为简单的程序段，然后再逐段编程。

6.4 自由 LD 语言编程

6.4.1 自由 LD 编辑器主要命令以及元素说明

自由 LD 编辑命令与元素与普通 LD 编辑器基本一致，此小节主要介绍自由 LD 的特点与差异项。

6.4.2 快捷操作

- 相比于普通 LD, 自由 LD 支持对元素自由连线；

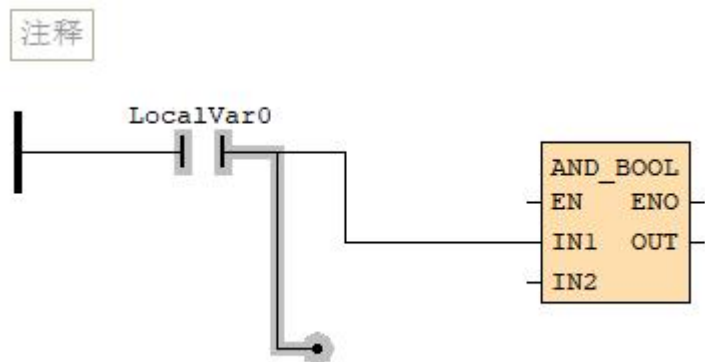


图 6-17 自由连线操作图示

- 左右并联操作，支持将触点或导轨等元素快速左右并联连接；

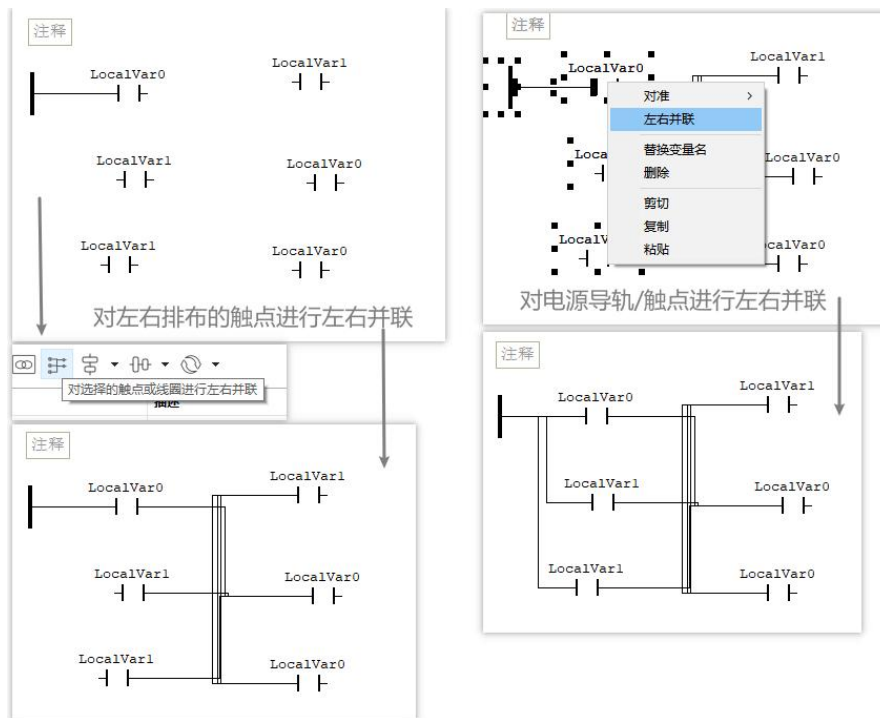
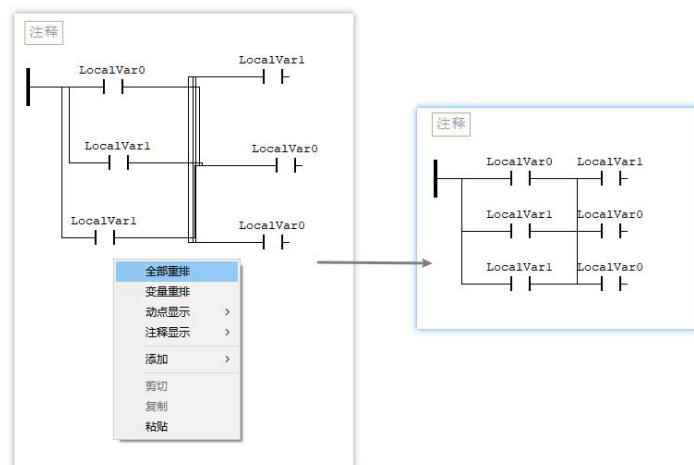


图 6-18 左右并联操作图示

- 更完善的重排功能，全部重排将对未锁定的导轨进行重排，变量重排将仅对块相连的变量进行重排；



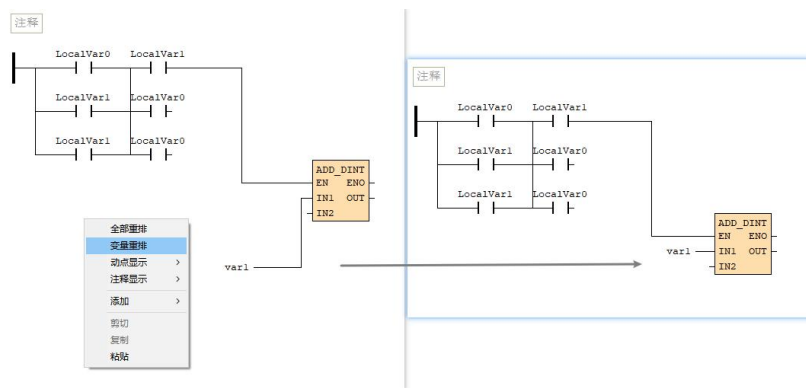


图 6-19 重排操作图示

6.4.3 执行次序

自由 LD 基于下面原则执行元素，排序靠前的优先：

1. 从电源导轨出发，按照能流线顺序执行。
2. 遇到岔路按照从上到下、从左到右的顺序执行。
3. 遇到块的优先执行块的输出变量赋值，再执行 ENO 以及其它连接线。
4. 对于未连接到电源导轨的悬空块，将按照位置顺序执行。

6.4.4 编程原则

- 变量必须先声明再使用；
- 现在支持未连接到电源导轨的悬空无执行控制块的编译，但是为了保证执行顺序，不推荐这样使用，如果需要使用，务必做好测试工作；
- 重排只辅助布局，如果遇到线条分布不理想的，可以手动微调后锁定该段落；
- 对于比较复杂或较大的 LD 编程，宜先将程序分为简单的程序段，然后再逐段编程。

6.5 载入示例

系统库中自带的功能块都分别有用结构化文本、梯级图、功能块图编写的示例，点击工具栏中帮助目录下的“载入示例”，选择需要查看的示例工程，确定后即可打开一个含有示例功能块应用方式的工程。



图 6-20 载入示例按键

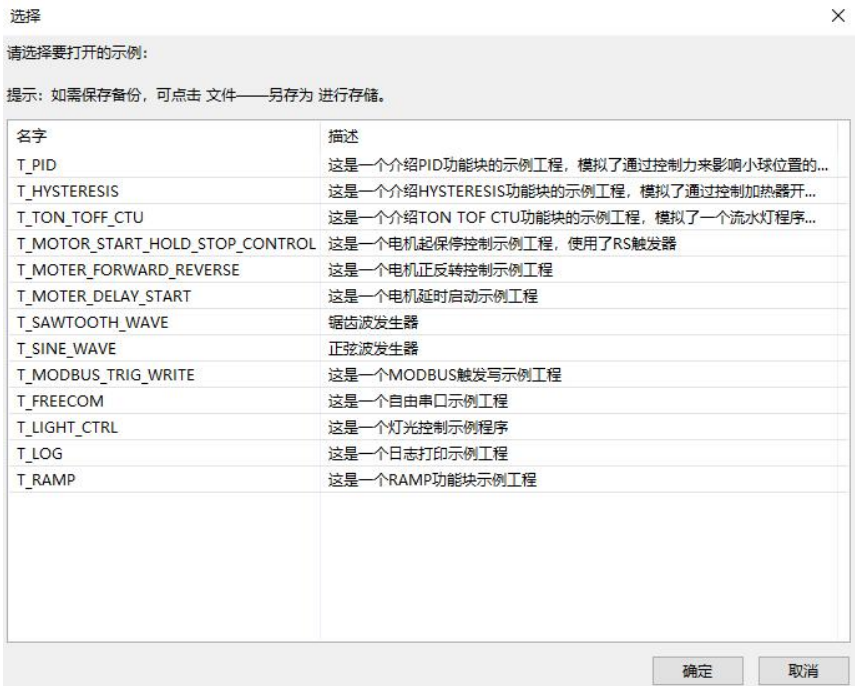


图 6-21 载入示例功能块选择

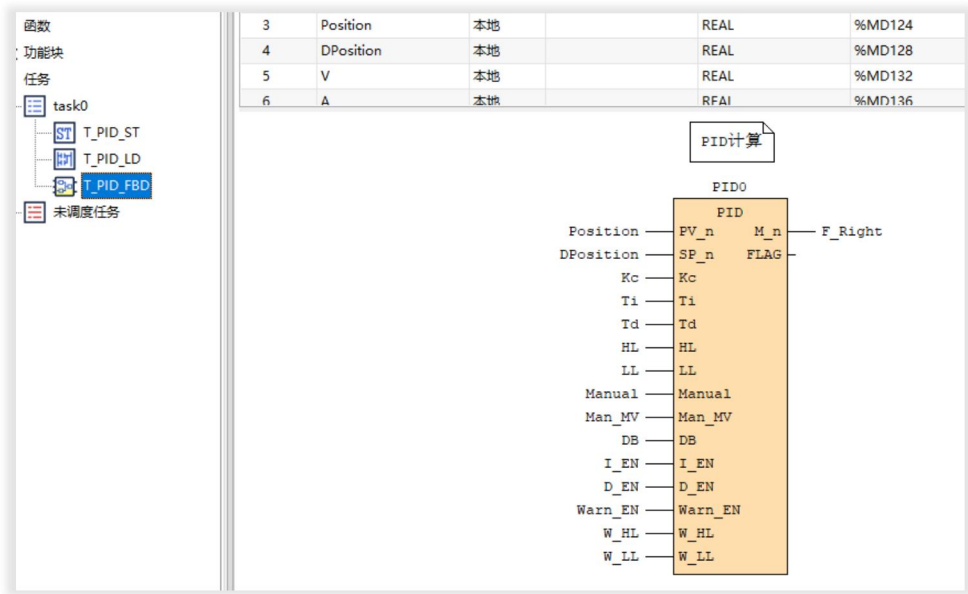


图 6-22 包含示例功能块的工程

7 资料版本说明

表 7-1 版本升级更改一览表

资料版本号	适用产品型号	更改说明
V1.0 (20240401)	PCS-600X 系列	第一版本编写
V1.1 (20241216)	PCS-200 系列、PCS-600X 系列、LCS-300 系列、GateWay 系列	第二版本编写