



www.hollysys.com



 Version 3.1.4

Auto *Think*





AutoThink V3.1

用户手册_工程组态

3.5 版

2016 年 7 月

版权声明

本手册内容，包括文字、图表、标志、标识、商标、产品型号、软件程序、版面设计及其它内容等，均受《中华人民共和国著作权法》、《中华人民共和国商标法》、《中华人民共和国专利法》及与之适用的国际公约中有关著作权、商标权、专利权或其他财产所有权法律的保护，为北京和利时智能技术有限公司专属所有或持有。

由于本手册中所描述的设备有多种使用方法，用户及设备使用责任人必须保证每种方法的许可性。对由使用或错误使用这些设备造成的任何直接或间接损失，北京和利时智能技术有限公司将不负法律责任。

由于实际应用时的不确定因素，北京和利时智能技术有限公司不承担直接使用本手册中提供的数据的责任。

本手册仅供商业用户阅读，在未得到北京和利时智能技术有限公司书面授权的情况下，无论出于何种目的和原因，不得以任何形式（包括电子、机械或其它形式）传播或复制本手册的任何内容。违者我公司将依法追究其相关责任。

已核对本手册中的内容、图表与所述硬件设备相符，但误差难以避免，并不能保证完全一致。同时，会定期对手册的内容、图表进行检查、修改和维护，恕不另行通知。

HOLLiAS MACS、HollySys、和利时、 **HollySys** 的字样和徽标均为北京和利时智能技术有限公司的商标或注册商标。

手册中涉及到的其他商标或注册商标属于它们各自的拥有者。

北京和利时智能技术有限公司版权所有。

地址：北京经济技术开发区地盛中路 2 号院

邮编：100176

电话：010-5898 1588

产品咨询热线：4008-111-999

技术支持电话：010-58981514

传真：010-5898 1558

网址：<http://www.hollysys.com/>

Email：PLC@hollysys.com

新浪微博：<http://weibo.com/hollysysplc>

目录

第 1 章 关于本书	1
1.1 文档更新	1
1.2 文档用途	1
1.3 阅读对象	2
1.4 使用约定	2
1.4.1 菜单命令	2
1.4.2 鼠标指令	2
1.4.3 键盘指令	2
1.4.4 重要信息	2
1.5 产品文档目录	3
1.6 快捷键说明	4
1.6.1 菜单命令快捷键	4
1.6.2 非菜单命令快捷键	7
第 2 章 软件概述	9
第 3 章 安装卸载	13
3.1 安装软件	13
3.1.1 安装环境	13
3.1.2 安装步骤	14
3.2 卸载软件	22
第 4 章 快速入门	25
4.1 软件启动	25
4.1.1 启动软件	25
4.1.2 退出软件	26
4.2 创建工程	26
4.3 工程组态	29
4.3.1 硬件配置	29

4.3.2	添加任务	38
4.3.3	创建 POU.....	40
4.3.4	变量定义.....	42
4.3.5	编写程序.....	44
4.3.6	调用 POU.....	46
4.4	编译下装	47
4.4.1	编译.....	47
4.4.2	通讯设置.....	48
4.4.3	下装.....	50
4.4.4	调试.....	50
第 5 章	AutoThink 软件组态介绍.....	55
5.1	软件界面	55
5.1.1	标题栏.....	56
5.1.2	菜单栏.....	56
5.1.3	工具栏.....	57
5.1.4	工程管理.....	57
5.1.5	信息输出窗口.....	60
5.1.6	状态栏.....	60
5.1.7	设备库.....	61
5.1.8	工作区域.....	61
5.2	工程管理	63
5.2.1	创建工程.....	63
5.2.2	打开工程.....	68
5.2.3	工程属性.....	70
5.2.4	保存工程.....	71
5.2.5	关闭工程.....	73
5.2.6	最近工程列表.....	74
5.2.7	查找.....	74
5.2.8	替换.....	75
5.2.9	工程权限.....	75
5.2.10	工程设置.....	77
5.2.11	下载/上传工程源文件.....	81
5.2.12	打印预览.....	81
5.2.13	打印(P).....	83
5.2.14	窗口(W).....	83
5.2.15	多机互联.....	84
5.2.16	帮助(H).....	86
5.2.17	编辑(E).....	87
5.3	程序组织单元	88

5.3.1	POU 的类型	89
5.3.2	添加 POU	90
5.3.3	POU 的操作	94
5.3.4	POU 的调用	98
5.3.5	配方函数.....	105
5.3.6	文件夹.....	118
5.3.7	任务组态.....	120
5.4	编程语言	130
5.4.1	LD 编辑器.....	130
5.4.2	ST 编辑器.....	156
5.4.3	CFC 编辑器.....	179
5.4.4	SFC 编辑器.....	198
5.4.5	相关操作.....	248
5.5	数据和变量.....	248
5.5.1	数据.....	249
5.5.2	数据类型.....	267
5.5.3	变量.....	279
5.6	硬件配置	315
5.6.1	硬件配置文件.....	315
5.6.2	硬件配置窗口.....	317
5.6.3	MC/LE 硬件组态.....	318
5.6.4	LK 硬件组态.....	361
5.6.5	平台切换.....	388
5.6.6	设备库.....	390
5.7	编译下装	392
5.7.1	编译.....	392
5.7.2	通讯设置.....	396
5.7.3	下装.....	398
5.7.4	在线调试及操作.....	401
5.7.5	查看操作日志.....	420
5.8	工具.....	421
5.8.1	PID 指令向导	421
5.8.2	串口通讯指令向导.....	429
5.8.3	轴参数显示设置.....	444
5.8.4	查看轴参数.....	446
5.8.5	示波器.....	447
5.8.6	下传用户文件.....	454
5.8.7	辅助工具.....	455

第 6 章 库管理	471
6.2 库管理器	473
6.2.1 查找	473
6.2.2 库配置	474
6.2.3 删除库	476
6.2.4 查看	477
6.2.5 库的更新	478
第 7 章 常见问题	485
7.1 工程升级	485
7.1.1 不切换目标配置	485
7.1.2 切换目标配置	485
索引	487

第1章 关于本书

1.1 文档更新

本手册以下章节内容进行了更新，如表 1.1-1 所示。

表 1.1-1 手册内容更新列表

序号	变更章节	新功能	备注
1	全文涉及的窗口图片	信息输出窗口位置调整	
2	5.8.4 查看轴参数	AT 中图标形象化	
3	5.1.6 状态栏	AT 状态栏明显标识离线 在线状态	
4	5.5.3.5 全局变量的 PWM 功能监视变量组	PWM 组态相关功能	
5	5.5.3.3 全局变量声明的方法	变量排序	
6	5.4.1.10 块元件的【高级】选项设置 5.4.3.4 块元件的设置引脚属性	块引脚更新	
7	5.4.1.1 LD 元素的注释说明	LD 网络紧凑排布	
8	5.4.4.5 步属性 5.4.4.27 选项	SFC 步、转换文本及注释 显示调整	
9	5.4.2.6 调试模式	ST 支持单步调试	
10	5.6.3.7 LE5405 组态	LE5405 配置组态	
11	第 6 章 库管理	算法库分类显示	
12	5.8.7.2 控制器操作	新增 IP 修改 (LE5405)	
13	5.6.3.8 的配置 RTEX 主从站下的复位协议主站和复位所有驱动器 5.6.3.8 的配置 EtherCAT 主从站下的复位协议主站	主协议栈复位命令	

1.2 文档用途

本手册帮助使用 AutoThinkV3.1.4 软件的用户，了解在 MC、LK、LE 平台中工程组态的方法，以及软件各功能的使用，并指导用户正确、有效的使用该软件。

1.3 阅读对象

本手册适用于以下人员：

- 负责系统工程实施的工程人员；
- 负责系统维护的技术人员；
- 现场系统运行时的操作人员。

1.4 使用约定

1.4.1 菜单命令

菜单项的名称均用【】表示，如【删除】、【下装】、【添加设备】等。

窗口、对话框的名字均用“”表示，如“设备库”、“库管理器”、“新增任务”等。

1.4.2 鼠标指令

指向：移动鼠标指针到某个项目上。

按下：按下鼠标左键一次并且不松开。

单击：按下鼠标左键一次并松开。

右击：按下鼠标右键一次并松开。

双击：快速按下鼠标左键两次并松开。

拖动：移动鼠标时按住鼠标左键不放。

1.4.3 键盘指令

键盘上的按键名称均用粗体字表示，如 **Shift**、**Enter**、**Shift+F2** 等。

1.4.4 重要信息

在本手册中，使用以下标识明确相应信息：



- 危险图标，标识该操作有造成物理伤害或人身伤亡的潜在威胁。



- 电击图标，标识该操作有造成电击伤害的潜在威胁。



- 警告图标，标识该操作有造成软硬件设备故障或损坏的潜在威胁。



- 重要图标，标识需要理解的操作或功能的重要信息。



- 操作图标，标识该对象的打开或操作方法。

SEE ALSO

- 参考图标，标识理解该功能需要参考的页面或内容。

1.5 产品文档目录



AutoThink V3.1 用户手册_工程组态



和利时可编程控制器 PLC 指令手册



LK 系列可编程控制器系统手册



MC1000 系列可编程控制器系统手册



1.6 快捷键说明

本节对 AutoThink 组态软件中的快捷键进行说明，帮助用户更好的掌握 AutoThink 软件的功能，便于使用。

在 AutoThink 中，快捷键分为菜单命令快捷键及非菜单命令快捷键，此外，不同的功能模块又有所不同。下面进行详细的介绍。

1.6.1 菜单命令快捷键

1. 通用菜单命令快捷键

表 1.6-1 离线编辑快捷键

功能	快捷键
文件\新建	Ctrl + N
文件\打开	Ctrl + O
文件\保存	Ctrl + S
文件\另存为	F12
文件\打印	Ctrl + P
编辑\撤销	Ctrl + Z
编辑\恢复	Ctrl + Y
编辑\剪切	Ctrl + X
编辑\复制	Ctrl + C
编辑\粘贴	Ctrl + V
编辑\删除	Del
编辑\查找	Ctrl + F
编辑\替换	Ctrl + H
编辑\输入助手	F2
工程\编译	F11

表 1.6-2 在线编辑快捷键

功能	快捷键
在线\监视\退出监视	Alt + F8
在线\单步调试	F10
在线\跳出	Ctrl+Shift+F8
在线\跳进	Ctrl+F11
在线\断点运行	Ctrl+F5
在线\运行	F5
在线\停止	Shift + F8
在线\写入	Ctrl + F7
在线\强制	F7
在线\全部释放	Shift + F7
在线\显示强制变量表	Ctrl + Shift + F7

2. LD 菜单命令快捷键

表 1.6-3 LD 编辑器菜单命令快捷键

功能	快捷键
插入\触点（追加）	F3
插入\前节	Ctrl + M
插入\后节	Ctrl + W
插入\并联触点	F10
插入\线圈	F6
插入\块元件	Alt + F9
插入\置反	Ctrl + G
置位/复位	Ctrl + T
多输入（增加块引脚）	Ctrl + D

3. CFC 菜单命令快捷键

表 1.6-4 CFC 编辑器菜单命令快捷键

功能	快捷键
插入\输入元件	Ctrl + I

功能	快捷键
插入\输出元件	Ctrl + U
插入\块元件	Ctrl + B
插入\跳转元件	Ctrl + J
插入\返回元件	Ctrl + R
插入\标签元件	Ctrl + L
插入\注释元件	Ctrl + K
插入\置反	Ctrl + G
插入\置位/复位	Ctrl + T
插入\使能	Ctrl + E
插入\多输入	Ctrl + D

4. SFC 菜单命令快捷键

表 1.6-5 SFC 编辑器菜单命令快捷键

功能	快捷键
插入\步-转换（前）	Ctrl + T
插入\步-转换（后）	Ctrl + E
插入\选择分支（右）	Ctrl + A
插入\选择分支（左）	Ctrl + D
插入\并行分支（右）	Ctrl + L
插入\并行分支（左）	Ctrl + M
插入\跳转	Ctrl + J
插入\转换-跳转	Ctrl + R
插入\添加入口动作	Ctrl + I
插入\添加出口动作	Ctrl + G
附加\增加并行分支标号	Ctrl + Q
附加\添加动作\转换	Ctrl + W
附加\移除动作\转换	Ctrl + U
附加\关联动作	Ctrl + K

1.6.2 非菜单命令快捷键

1. 通用非菜单命令快捷键

表 1.6-6 通用非菜单命令快捷键

功能	快捷键
帮助	F1
IEC 视图区修改变量属性	Shift + F2
退出监视	Ctrl + F8
高亮显示下一条错误信息	F9
高亮显示前一条错误信息	Shift + F9
变量自动声明	Enter
全选	Ctrl + A

2. ST 键盘命令

表 1.6-7 ST 键盘命令快捷键

功能	快捷键
向前删除一个字符	Backspace
向后删除一个字符	Delete
下一行	↓
上一行	↑
滚动到下一行	Ctrl + ↓
滚动到上一行	Ctrl + ↑
向下逐行选中	Shift + ↓
向上逐行选中	Shift + ↑
光标跳到行尾	End
光标跳到程序结尾	Ctrl + End
光标后整行选中	Shift + End
光标后整个程序选中	Ctrl + Shift + End
光标跳到本行开始位置	Home
光标跳到程序开始位置	Ctrl + Home
光标前整行选中	Shift + Home

功能	快捷键
光标前整个程序选中	Ctrl + Shift + Home
插入\覆盖切换	INSERT
光标跳到左侧一个字符前	←
光标跳到左侧一个词前	Ctrl + ←
选中光标左侧一个字符	Shift + ←
选中光标左侧一个词	Ctrl + Shift + ←
光标跳到右侧一个字符前	→
光标跳到右侧一个词前	Ctrl + →
选中光标右侧一个字符	Shift + →
选中光标右侧一个词	Ctrl + Shift + →
插入制表标符	Tab
光标回退一个制表符	Shift+ Tab

第2章 软件概述

AutoThink 软件是和利时公司开发的基于 Windows 平台的编程工具，该软件用于控制站算法编程，是控制方案的开发平台，是硬件配置和软件编程的标准软件包。

AutoThink 主要具有以下功能及特点：

- 组态简单直观：采用树形结构进行硬件配置、任务配置、程序块组态。
- 组态语言丰富标准：支持 LD、ST、CFC 和 SFC 四种编程语言，完全符合 IEC61131-3¹标准。
- 组态指令丰富，支持多种数据类型。
- 支持图形化硬件模块配置和参数设置。
- 支持用户自定义库。
- 具有仿真调试功能。
- 具有在线调试及程序检查等功能。
- 具有日志功能。
- 具有密码保护功能。
- 用工程来管理一个自动化系统的硬件和软件。

实际工程中，每一个控制站中要对应一个控制站算法工程，控制站算法工程包含了所要运行的控制策略，即控制算法。在控制站算法中主要进行任务配置、硬件配置、创建主程序(Main)以及通过添加 POU(程序组织单元)来编写算法。

本手册是和利时公司自主研发的 MC、LK 和 LE 智能控制器编程软件的指导手册，由于 AutoThink 组态软件，包含了 MC、LK、LE 三个目标平台，三个平台的某些功能不通用，请用户在查阅具体内容前，先了解三种产品在 AutoThink 软件中的功能差异，如表 2-1 所示。帮助您快速的了解各个平台具有的功能。

表 2-1 功能适用范围列表

功能	MC 目标平台	LK 目标平台	LE 目标平台	说明
----	---------	---------	---------	----

¹IEC61131-3: 由国际电工委员会 (IEC) 于 1993 年 12 月所制定 IEC 61131 标准的第 3 部分-编程语言的标准，用于规范可编程控制器 (PLC, DCS, IPC, CNC 和 SCADA) 的编程语言及其基本元素，应用 IEC 61131-3 标准已经成为工业控制领域的趋势。编辑软件只需符合 IEC 61131-3 国际标准规范，便可借由符合各项标准的语言架构，来建立人人皆可了解的程序。

功能	MC 目标平台	LK 目标平台	LE 目标平台	说明
任务配置	√	√		
多任务编辑	√			
配方/配方函数			√	
Table 变量组	√			
轴参数变量组	√			
LEIODiagGroup	√			
ModbusOrderDiagVar	√	√		
LE5405DiagGroup	√			
PWM 功能监视变量组	√			
中断配置			√	
多机互联			√	
增量编译	√	√		
仿真			√	
添加配置文件		√	√	
串口通讯指令向导			√	
配方数据使用向导			√	
轴参数显示设置	√			
查看轴参数	√			
示波器	√			
终端	√			
全部释放		√	√	
强制变量表		√	√	
写入存储卡			√	
清空存储卡			√	
急停按钮	√			
使能按钮	√			
下传用户文件	√	√		
设置控制器 IP	√			

功能	MC 目标平台	LK 目标平台	LE 目标平台	说明
控制器校时	√	√		
读取控制器信息	√	√		
控制器锁	√	√		
网络配置		√		
固件升级	√	√		
工程升级	√	√		
日志读取	√	√		
IP 修改 (LE5405)	√			
强制		√	√	
支持 string 类型	√	√		
热复位		√		
断点调试	√			
64 位整型数据类型	√			

第3章 安装卸载

AutoThink V3.1.4 软件是目前发布的专门用于 PLC 编程使用的 AutoThink 版本。该软件属于控制站算法组态软件。本章主要讲述 AutoThink V3.1.4 软件的安装与卸载，在下文中使用 AutoThink 作为该版本软件的简称。

3.1 安装软件

用户在使用 AutoThink 软件进行编写应用程序之前，必须安装 AutoThink 软件。本节介绍该软件的安装环境及步骤。

3.1.1 安装环境

计算机配置如表 3.1-1 所示。

表 3.1-1 推荐使用的计算机配置

环境	类型	型号
硬件环境	显示器	彩色 CRT 或液晶屏
	输入输出	标准键盘、鼠标
	USB 接口	至少 1 个 USB 2.0 接口
	显卡	分辨率支持 1280×720
	CPU	Intel Pentium 2.4 GHz 以上
	内存	512 MB 以上
软件环境	硬盘	10 GB 以上
	操作系统	WindowsXP Professional+SP3 中文版、 Windows 7 Professional 32bit 中文版
	工具软件	MS Excel 2003, 2007, 2010

工程中使用的计算机配置应与上述配置相当，或高于上述配置。

3.1.2 安装步骤

如若之前已安装过本软件，需完全卸载，并删除安装目录文件夹..\AutoThink 后，方可重新安装。注意在删除软件时做好工程备份。

软件主要安装步骤：

第1步 选择安装语言

将本软件安装光盘插入光盘驱动器，弹出“选择安装语言”对话框，如图 3.1-1 所示。



图 3.1-1 安装语言选择

第2步 启动安装向导

语言选择为中文（简体），单击确定。

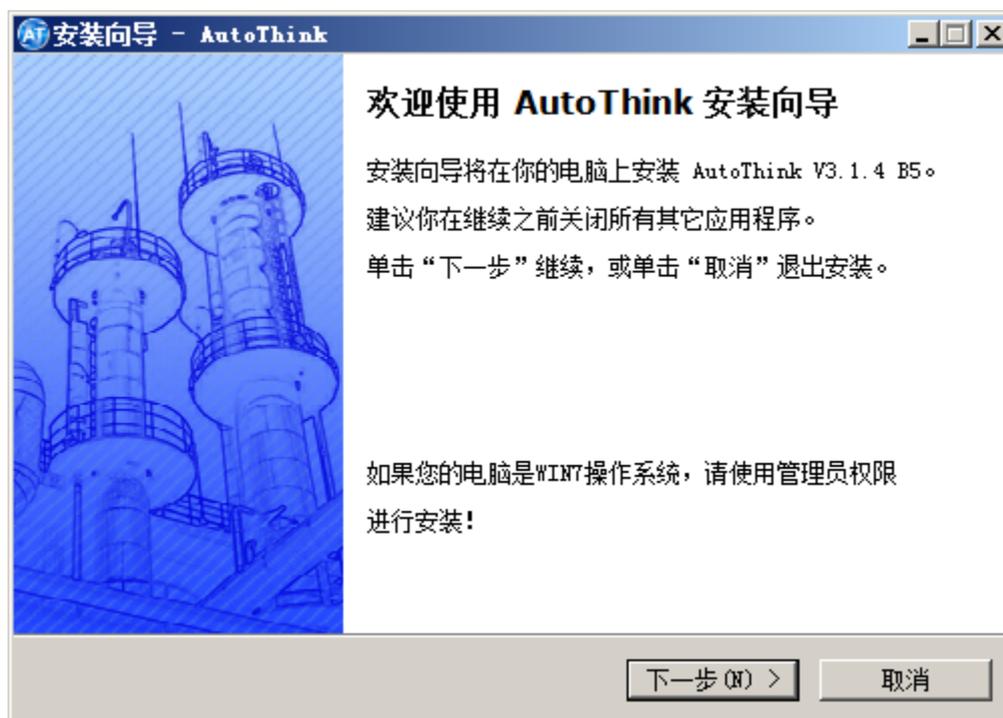


图 3.1-2 安装向导

第3步 许可协议

单击下一步(N)，如图 3.1-3 所示。选择我接受协议(A)，然后单击下一步(N)。



图 3.1-3 许可协议

第4步 选择安装路径

如图 3.1-4 所示，通过单击**浏览(R)...**选择安装路径，单击**下一步(N)**。



图 3.1-4 选择安装路径

第5步 选择开始菜单文件夹

如图 3.1-5 所示, 通过单击**浏览(R)...**选择文件夹, 单击**下一步(N)**。

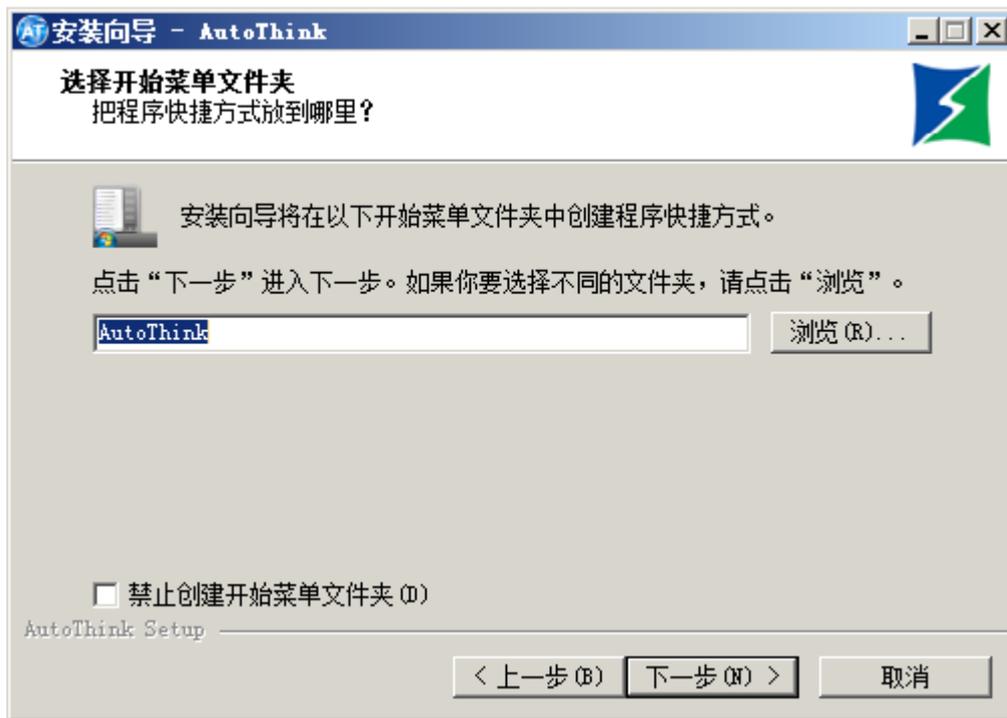


图 3.1-5 开始菜单文件夹名称

第6步 选择附加任务

如图 3.1-6 所示，在**附加图标**下勾选需要创建的图标项，单击**下一步(N)**。

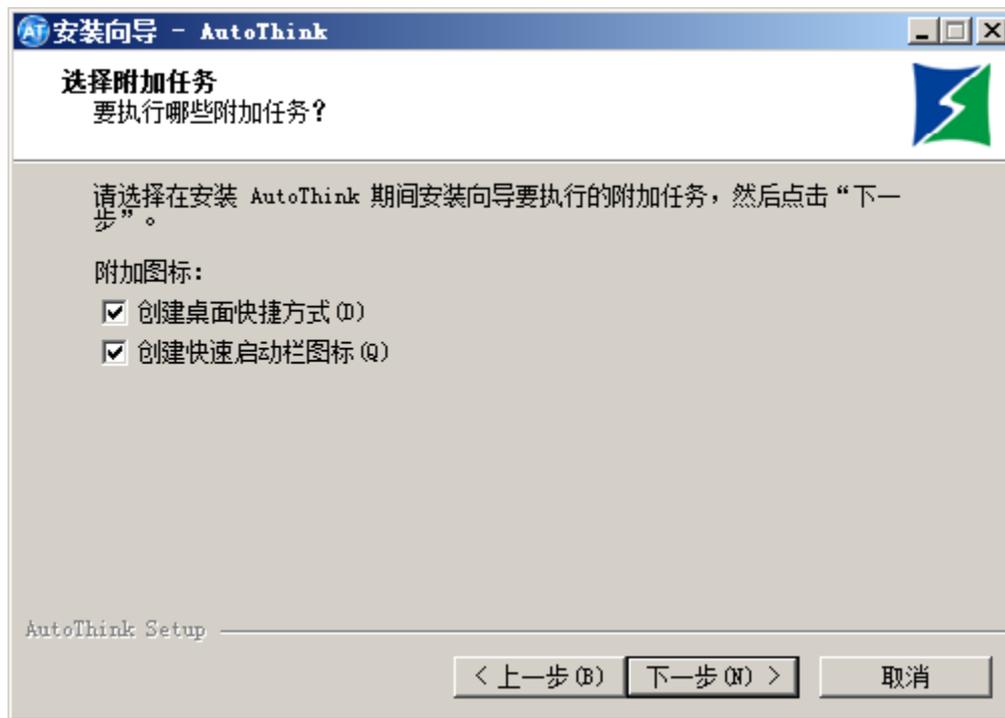


图 3.1-6 快捷方式创建

第7步 准备安装

如图 3.1-7 所示，确认安装信息。如果需要进行修改，则单击上一步(B)。无误后单击**安装**，开始软件安装。



图 3.1-7 准备安装

第8步 安装过程

如图 3.1-8 所示，会显示安装进度；若要取消安装，则单击**取消**。

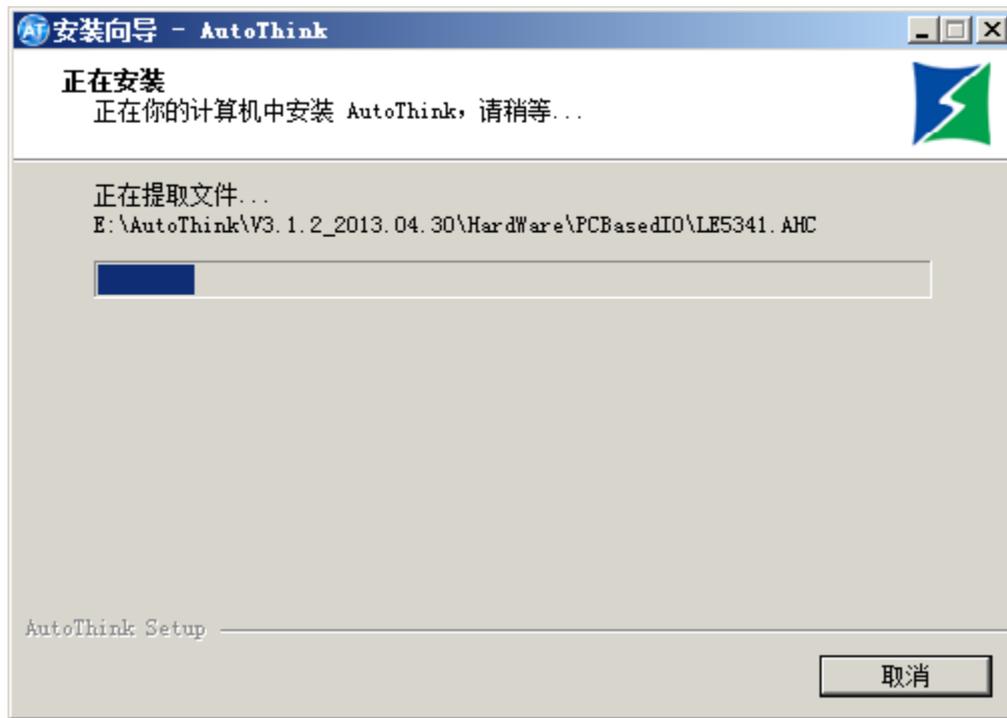


图 3.1-8 安装过程

第9步 安装完成提示

安装完毕，弹出“完成 AutoThink 安装”窗口。如图 3.1-9 所示。



图 3.1-9 完成安装

当前工程为 LE 时，**安装 USB 驱动**选项必须勾选，否则无法下装工程。

安装完成后需要立即运行 AutoThink 软件，则勾选**立即启动 AutoThink**。单击**完成**，安装 AutoThink 软件完毕。

3.2 卸载软件

可通过控制面板或启动卸载程序卸载 AutoThink 软件。

通过启动卸载程序进行卸载操作：



- 桌面：单击【开始】菜单—【所有程序】—【AutoThink】—【卸载 AutoThink】。



图 3.2-1 卸载确认

单击**是(Y)**，如图 3.2-2 所示，开始卸载。



图 3.2-2 开始卸载

卸载完毕，弹出提示窗口，如图 3.2-3 所示。单击**确定**，完成软件的卸载。



图 3.2-3 卸载完成

第4章 快速入门

4.1 软件启动

4.1.1 启动软件



- 桌面：单击【开始】菜单—【所有程序】—【AutoThink】文件夹—【AutoThink】软件；

- 快捷图标：双击桌面 AutoThink 软件图标。

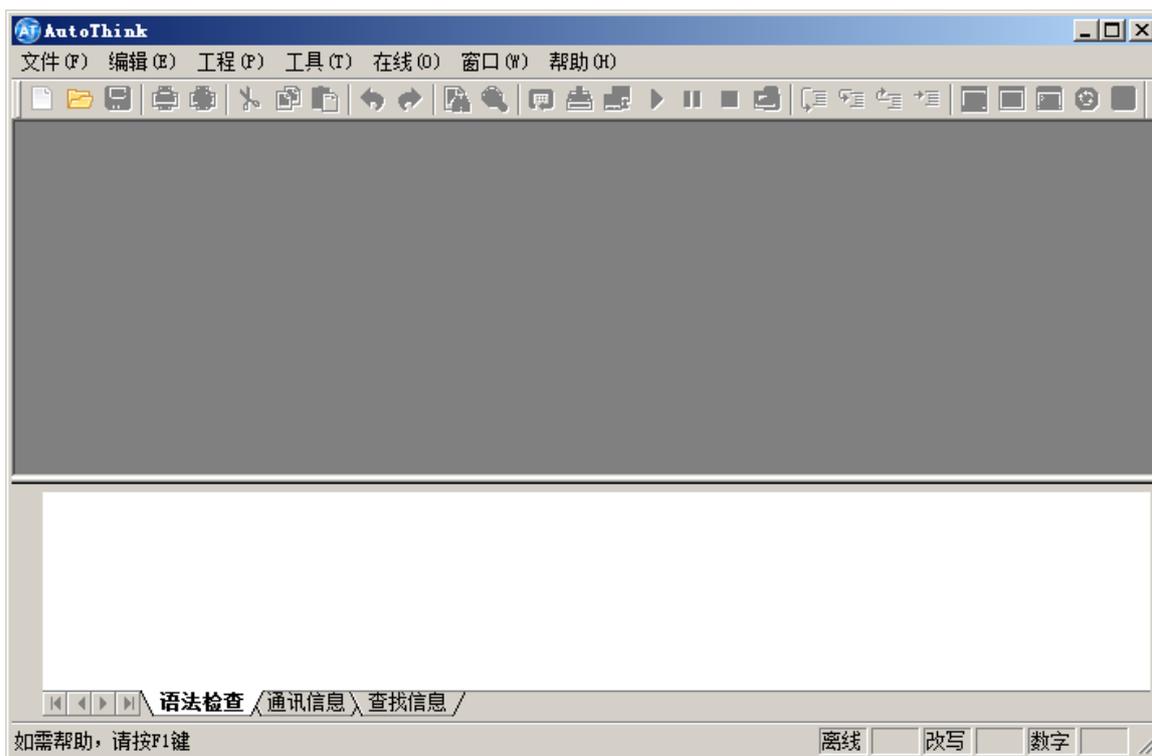


图 4.1-1 AutoThink 软件界面

如果已存在工程，打开 **AutoThink** 时，默认加载上一次打开的工程。打开时是否加载工程可在【选项】中设置，请详见 [5.2.10.1 配置](#)。

4.1.2 退出软件



- 菜单栏：单击【文件】—【退出】；
- 快捷键：**Alt+F4**。

退出 **AutoThink** 组态软件，若当前打开工程做过修改，系统提示保存。

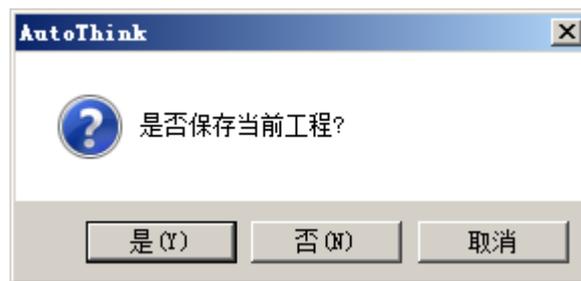


图 4.1-2 保存工程提示



- 仿真状态下无法退出软件，请先退出仿真再执行软件退出操作。

4.2 创建工程

打开软件后首先需要创建工程。



- 菜单栏：单击【文件】—【新建】；
- 工具栏：。

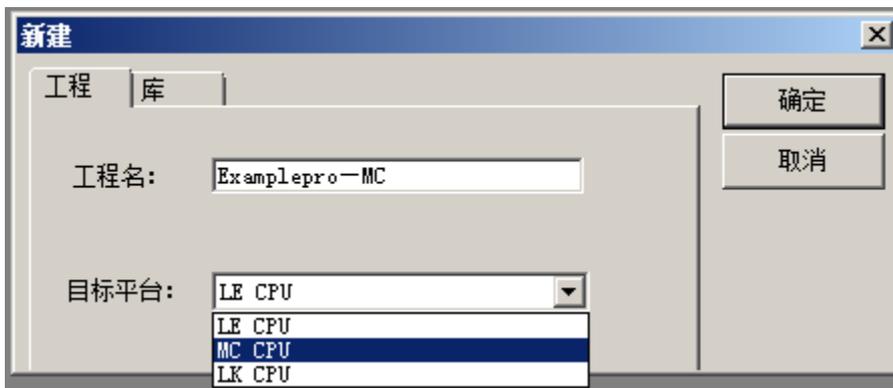


图 4.2-1 新建工程对话框

输入工程名，选择目标平台，完成后单击确定。

新建工程后，在“工程管理”窗口下将自动添加以工程名称为根节点的树状图。选择的目标平台不同，工程管理树的节点内容也不同。MC、LK、LE 的工程管理树如图 4.2-2、图 4.2-3、图 4.2-4 所示。

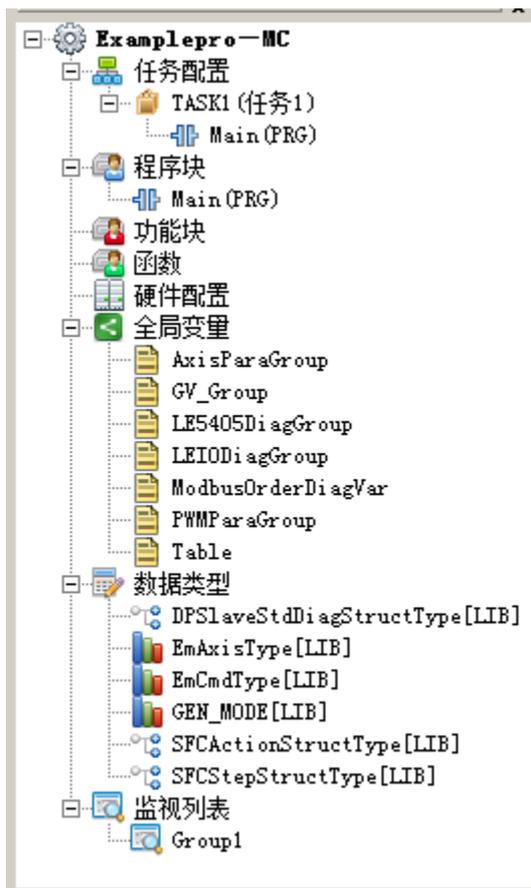


图 4.2-2 MC 工程树状图

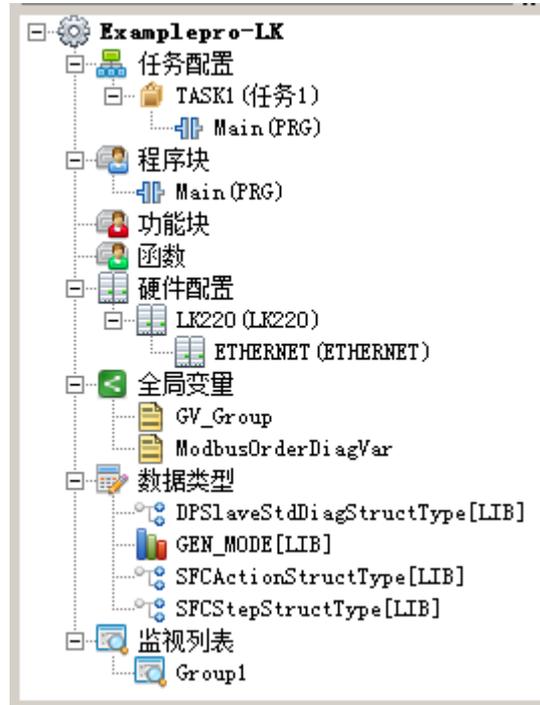


图 4.2-3 LK 工程树状图

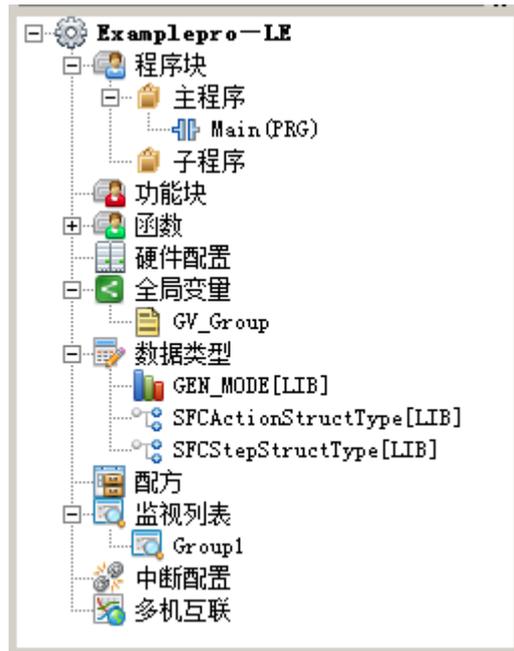


图 4.2-4 LE 工程树状图

4.3 工程组态

4.3.1 硬件配置

4.3.1.1 MC/LE 平台

硬件配置操作是对硬件结构进行组态，生成一个与实际的硬件系统完全一样的系统。需要添加 CPU 与 IO 模块、组态通道参数、通讯参数。从“设备库”中拖拽需要配置的硬件模块到组态区域，如图 4.3-1 所示。



图 4.3-1 硬件配置操作

对 LE5230 DI 模块和 LE5220 DO 模块进行通道组态，修改通道名称和通道说明，如图 4.3-2 所示。



(a)

模块参数信息 通道参数信息 通讯参数信息				
通道基本参数				
通道号	通道名称	通道类型	通道地址	通道说明
1	P101	BOOL	%QX28.0	P101启停输出
2	M3_CH_1	BOOL	%QX28.1	第1位输出(晶体管)

(b)

图 4.3-2 通道组态

硬件配置的详细内容请参见章节 5.6 硬件配置。

4.3.1.2 LK 平台

LK 平台通过在【硬件配置】节点上添加设备的方式进行硬件组态。

创建 LK 工程后，硬件配置节点下缺省有一个控制器模块 LK220，可以在控制器模块下配置 DP 协议和 MODBUS TCP 协议。如图 4.3-3 所示，LK220 缺省添加以太网适配器，通过右键菜单的【添加设备】命令，可以添加 DP 通信设备。

1. 配置 DP 协议

如图 4.3-3 所示，单击【添加设备】命令。



图 4.3-3 CPU 模块右键菜单

在弹出的“添加”对话框中，选择 LK249 模块进行添加。如图 4.3-4 所示。

LK220 模块最多支持一个以太网适配器和一个 LK249 设备。

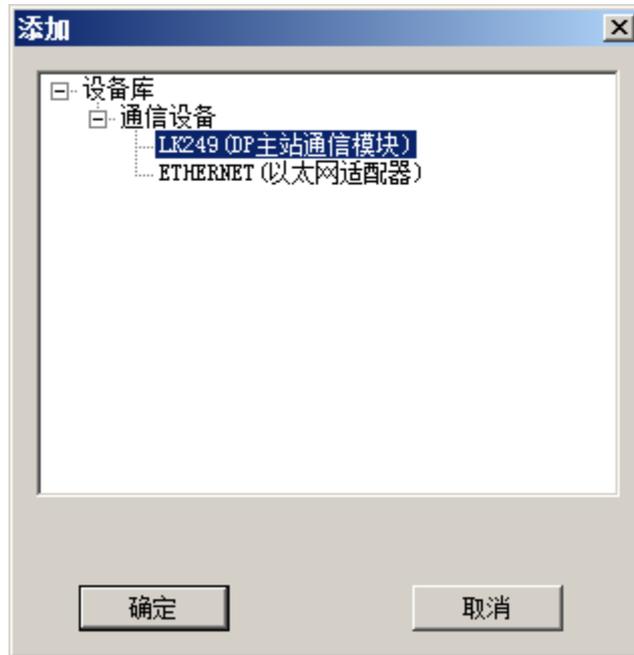


图 4.3-4 选择通信设备

(1) 添加协议

为 LK249 模块添加 DP 协议。如图 4.3-5 所示，选择【添加协议】进行添加。



(a)



(b)

图 4.3-5 添加协议

(2) 配置端口

为协议配置 COM 端口。双击 DP_MASTER 协议，打开“设备信息”窗口，如图 4.3-7 所示。



项目	内容
协议名称	DP_MASTER
端口	COM口
波特率	500.00kBits/s
TSL (0~65535)	200
最小站延时 (0~65535)	11
最大站延时 (0~65535)	100
发送器失败/中继器切换时间 (0~255)	0
建立时间 (0~255)	1
目标循环时间	3416
GAP更新因数	10
重试最大次数 (0~255)	1
最小从站间隔 (0~65535)	50
主站对主站最大请求时间	500
全局总线状态在双口RAM中保存所需最小时间	1200
协议	PROFIBUS-DP
从站地址范围	2~125
主/从站位置	Profibus-DP 主站

图 4.3-6 配置端口

在端口项中配置端口为 COM 口。

(3) 添加 IO 设备

右击协议，选择【添加设备】项，可以添加从站 IO 设备，如图 4.3-7 所示。

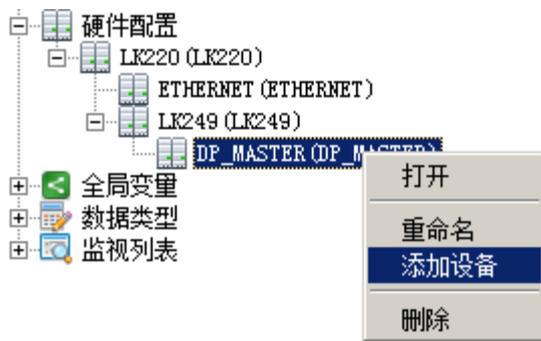


图 4.3-7 添加 IO 设备

选择模块类型，设置从站地址，单击**确定**，在协议节点下生成从站设备，如图 4.3-8 所示。

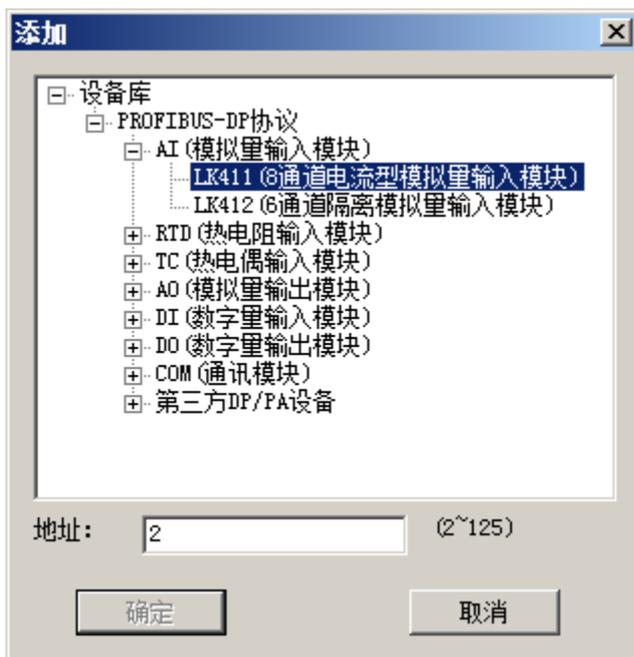


图 4.3-8 选择 IO 模块

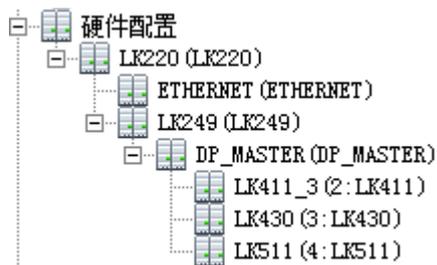


图 4.3-9 IO 模块添加完成

在从站的设备信息窗口中，可以设置从站地址如图 4.3-10 所示。



图 4.3-10 LK411 设备信息窗口

通过双击**设备属性**可以打开设备属性对话框，查看已添加的子模块信息和设置【用户参数】。从站设备是第三方设备或通信模块时，可以在“设备信息”窗口中添加子模块。

2. 配置 MODBUS TCP 协议

在【ETHERNET】节点下，可以配置 MODBUS TCP 主从协议。

(1) 添加协议

当前 CPU 模块作主站时，选择 MODBUSTCP_MASTER 主站协议；当前 CPU 模块作从站时，选择 MODBUSTCP_SLAVE 从站协议。通过【添加协议】命令进行协议的添加。



(a)



(b)

图 4.3-11 添加协议

(2) 端口配置

打开 MODBUSTCP 主/从协议的设备信息窗口，配置端口为以太网口 1。如图 4.3-12 所示，以 MODBUS TCP 主站为例进行说明。



图 4.3-12 配置端口

(3) 用户参数设置

在 MODBUS TCP 主/从站的设备信息窗口，双击设备属性，进行用户参数的设置。如图 4.3-13 所示，以 MODBUS TCP 主站为例进行说明。

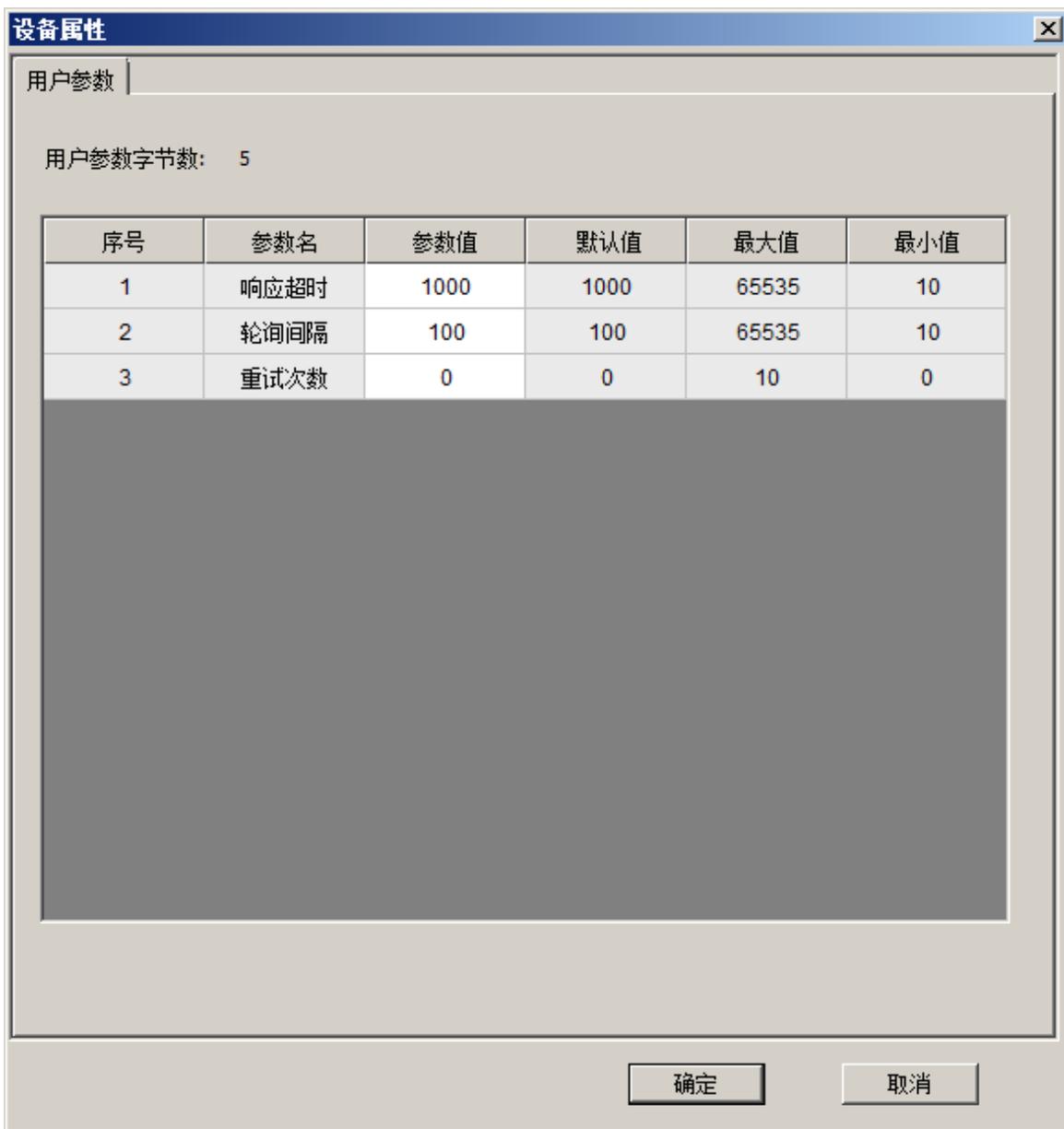


图 4.3-13 用户参数设置

(4) MODBUS 主站配置从站

在 MODBUSTCP 主站下可配置从站。如图 4.3-14 所示，进行从站的添加。

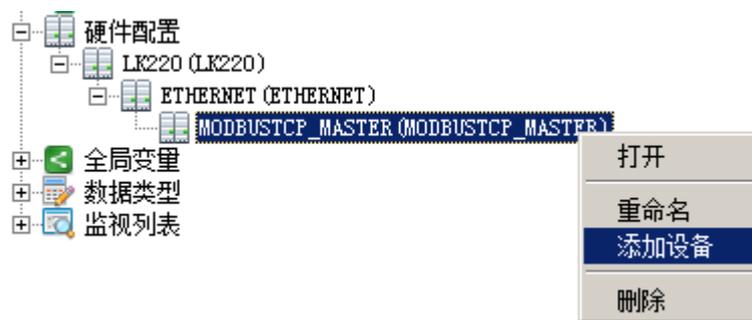


图 4.3-14 从站配置

■ 添加从站指令

在从站的设备属性窗口，可以添加需要的通信数据。如图 4.3-15 所示，选择指令添加到已选指令下。



图 4.3-15 配置从站指令

在【用户参数】中设置从站与主站通信的数据参数。

详细内容请参见章节 [5.6.4 LK 硬件组态](#)。

4.3.2 添加任务

工程管理树的【任务配置】节点下缺省有任务 TASK1，如图 4.3-16 所示。



图 4.3-16 任务配置节点



- LK 只有一个缺省的任务，不能进行【添加任务】操作。

右击【任务配置】节点，单击【添加任务】命令，进行任务添加。



图 4.3-17 添加任务

弹出“新增任务”对话框，勾选启动高级任务配置，创建一个时间片为 3、启动类型为自动、50 ms 运行一次的周期性任务 TEST_01，如图 4.3-18 所示。



图 4.3-18 任务参数设置

参数设置完成，单击**确定**，在【任务配置】节点下生成新任务 TEST_01，如图 4.3-19 所示。

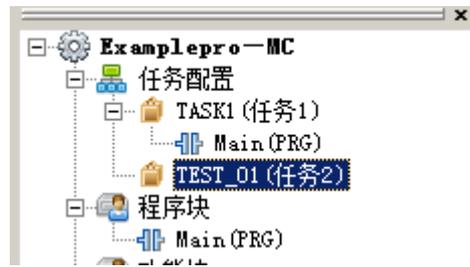


图 4.3-19 任务添加完成

4.3.3 创建 POU

4.3.3.1 MC/LK 平台

【程序块】节点下缺省有程序【Main(PRG)】，如图 4.3-20 所示。



- 工程管理树：右击【程序块】节点，单击【添加程序】。



图 4.3-20 添加程序



图 4.3-21 添加程序对话框

输入名称，选择编程语言，单击确定，程序块添加完成，【程序块】节点下生成新的程序【LDprg (PRG)】，如图 4.3-22 所示。

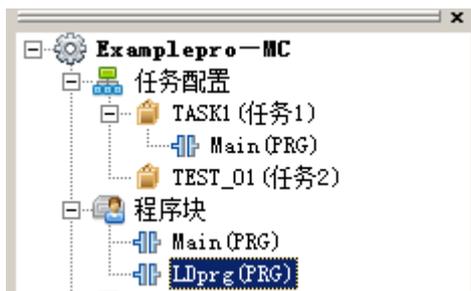


图 4.3-22 程序块添加完成

4.3.3.2 LE 平台

如图 4.3-23 所示，LE 工程的【程序块】节点下有【主程序】和【子程序】节点。在新建工程时，已经有一个缺省的主程序，名为【Main(PR)】。



图 4.3-23 LE 工程程序块节点

这里的主程序等效于 MC 工程的【任务】节点，但是前者的子节点不支持用户自定义，而后者可以。【子程序】等效于 MC 工程的【程序块】节点。



- 工程管理树：右击【子程序】节点，单击【添加子程序】。

如图 4.3-24 所示，具体操作同 4.3.3.1 MC/LK 平台。这里不再赘述。



图 4.3-24 添加子程序

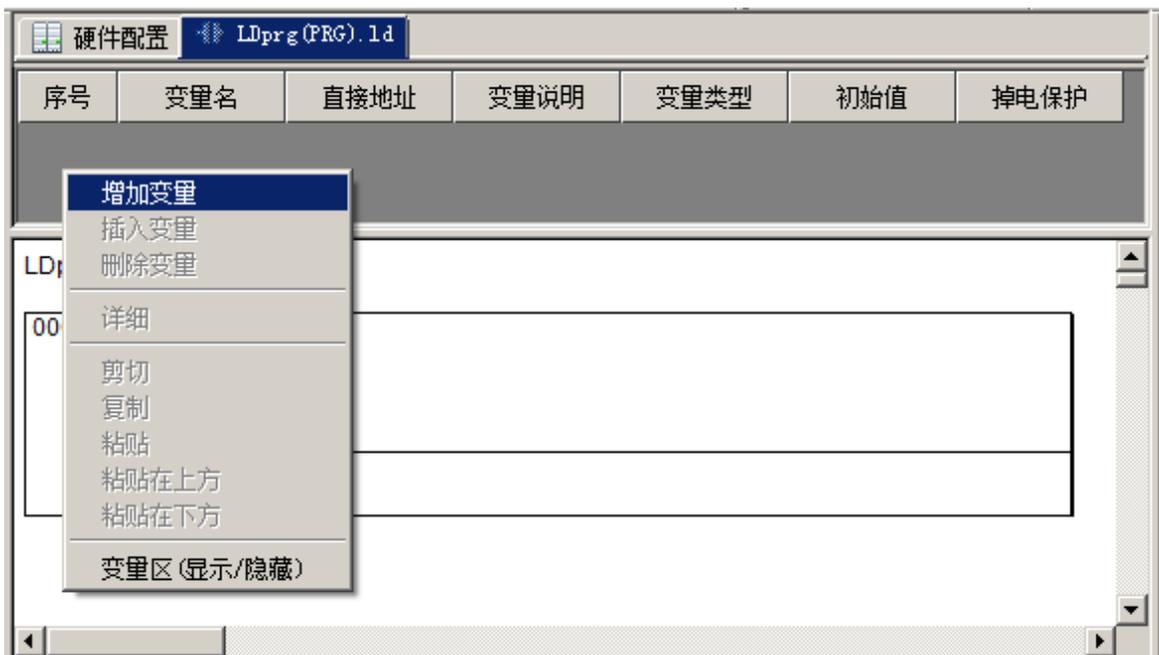
4.3.4 变量定义

双击【程序块】节点下已建好的程序【LDprg (PRG)】，打开程序编辑器，如图 4.3-25 所示。



图 4.3-25 程序编辑器

在程序区上部，右击空白处，选择【增加变量】命令，进行变量的添加。如图 4.3-26 所示。



(a)



序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	p1			BOOL	FALSE	FALSE
0002	p2			BOOL	FALSE	FALSE
0003	p3			BOOL	FALSE	FALSE

(b)

图 4.3-26 添加变量

根据所组态的通道，定义变量，更改变量名，在直接地址栏输入通道地址，变量类型选择与通道类型一致。如图 4.3-27 所示。



序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	K101	%IX16.0	P101启动按钮	BOOL	FALSE	FALSE
0002	K102	%IX16.1	P101停止按钮	BOOL	FALSE	FALSE
0003	P101_C	%QX28.0	P101启停输出	BOOL	FALSE	FALSE

图 4.3-27 变量定义完成

4.3.5 编写程序

完成 P101 泵的起保停逻辑。当启动按钮 K101 按下时，P101 泵启动，当停止按钮 K102 按下时，P101 泵停止。

在节点 001 上一次添加【插入触点】、并联触点、【线圈】，如图 4.3-28 所示。

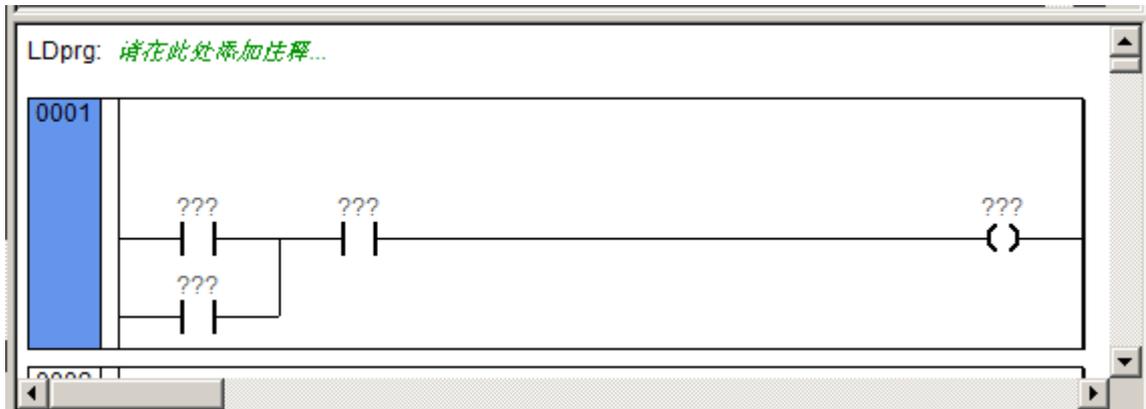
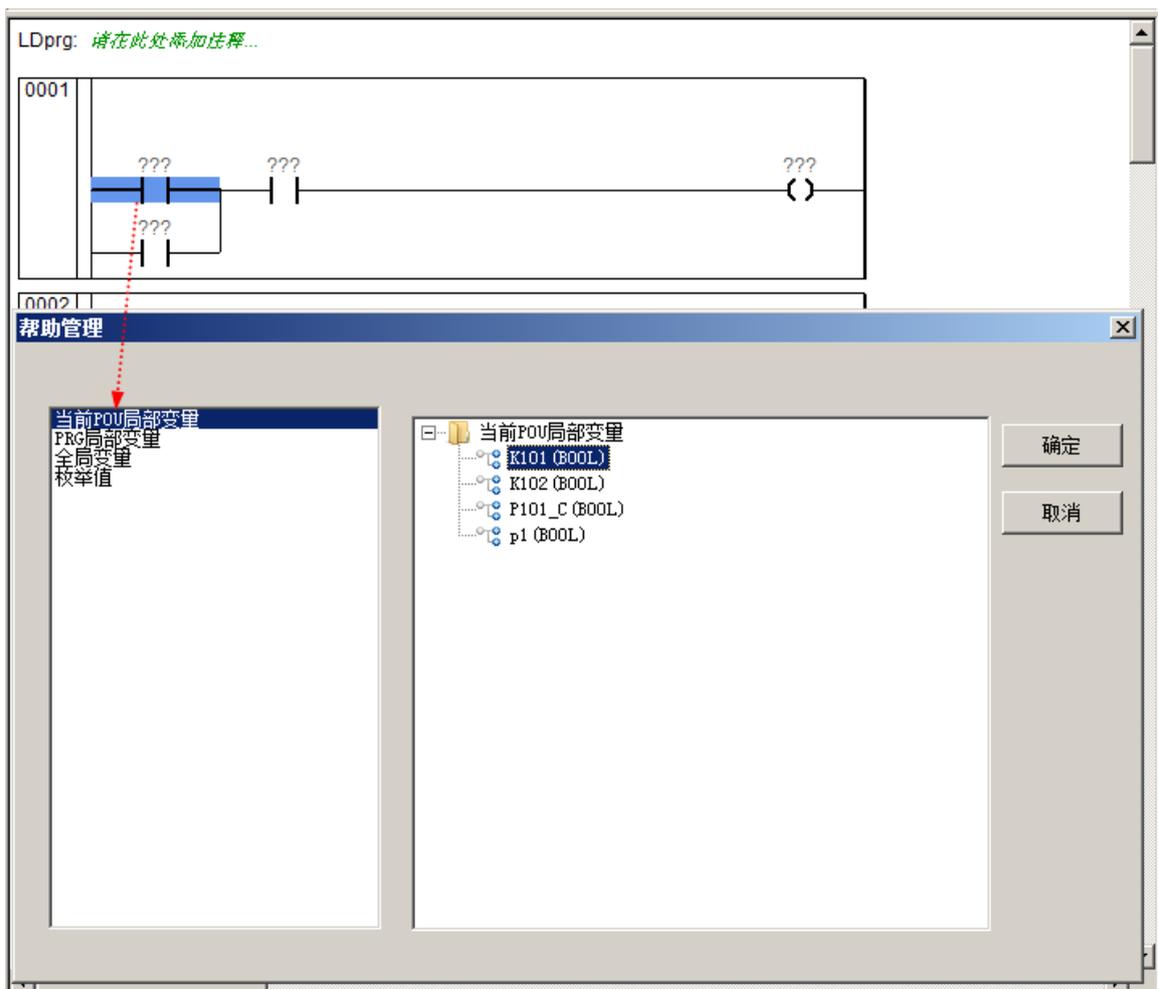
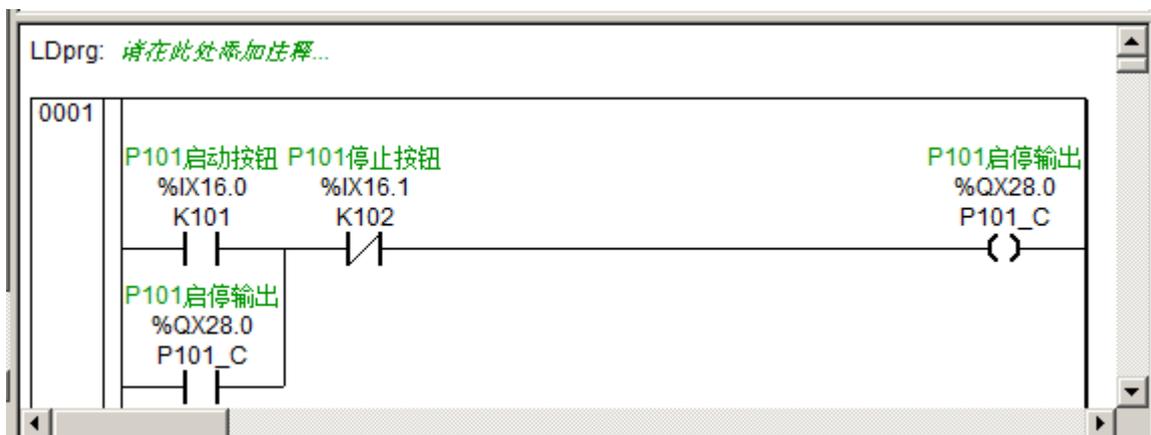


图 4.3-28 添加元件

选中触点或线圈，按 **F2**，在弹出的“帮助管理”对话框中选择已定义的变量，如图 4.3-29 所示。



(a)



(b)

图 4.3-29 引用变量

4.3.6 调用 POU

1. MC/LK 平台

将编写完成的程序块 LDprg (PRG) 拖动到已建好的任务 TEST_01 下面，如图 4.3-30 所示。

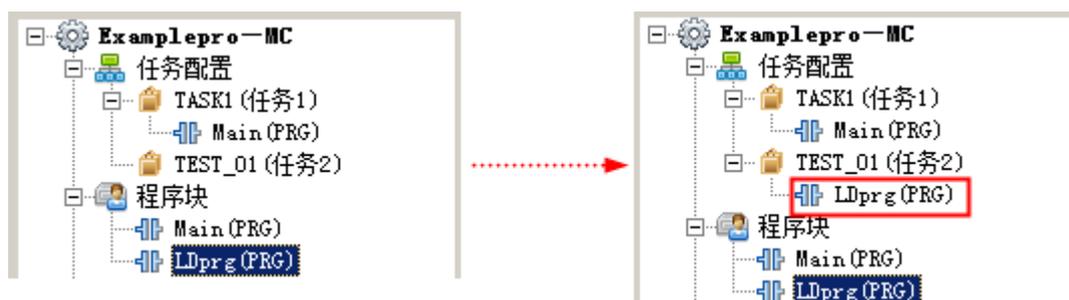


图 4.3-30 任务调用 POU

2. LE 平台

LE 工程的主程序 Main(PRG)调用【子程序】时，将子程序节点下的 POU 拖动到主程序的节上，进行程序调用，如图 4.3-31 所示。

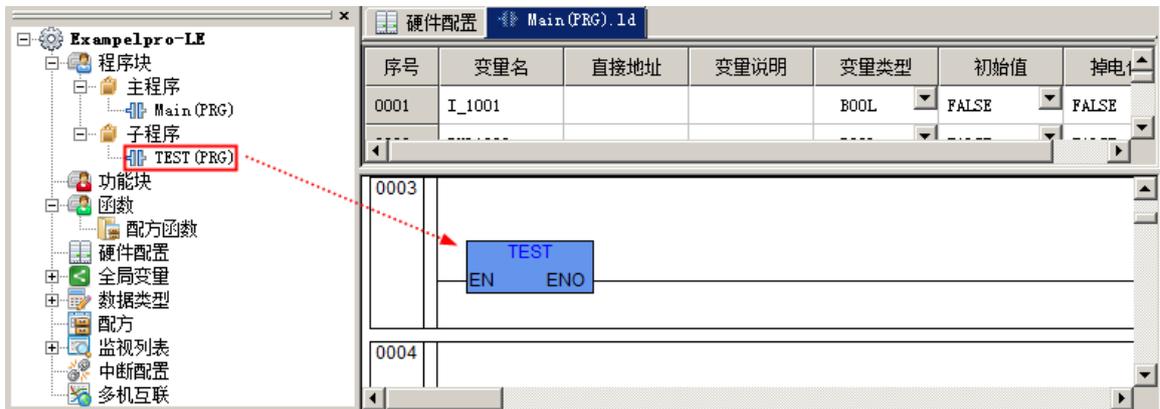


图 4.3-31 主程序调用子程序

4.4 编译下装

4.4.1 编译

程序编写完毕，对其进行编译，编译的目的有两个，一是进行语法检查，二是形成可下装到控制器的执行文件。



- 菜单栏：单击【工程】—【编译】。

执行编译操作。编译结果显示在信息输出窗口的【语法检查】标签页内。

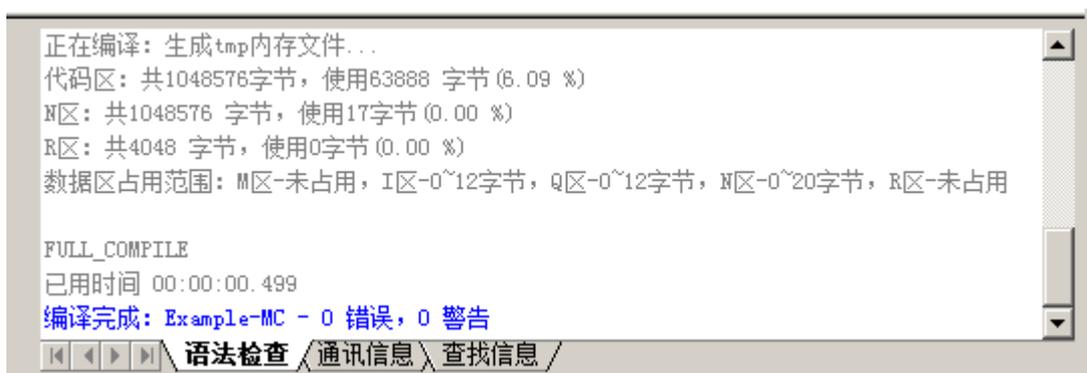


图 4.4-1 编译成功提示

4.4.2 通讯设置

编译无误后，将编译生成的程序文件下载到控制器中，再进行监视等操作。进行下载操作前，需要对当前连接的控制器进行通讯设置。



- 菜单栏：单击【在线】—【通讯设置】。

1. MC/LK 平台

设置当前控制器的 IP 地址，如图 4.4-2 所示。



图 4.4-2 MC 工程通讯设置对话框

通讯设置完成后，需要设置计算机网卡的本地连接 IP 地址与控制器在同一网段中，如控制器地址为 192.168.0.250，则本地连接 IP 地址可以为 192.168.0.10，如图 4.4-3 所示。

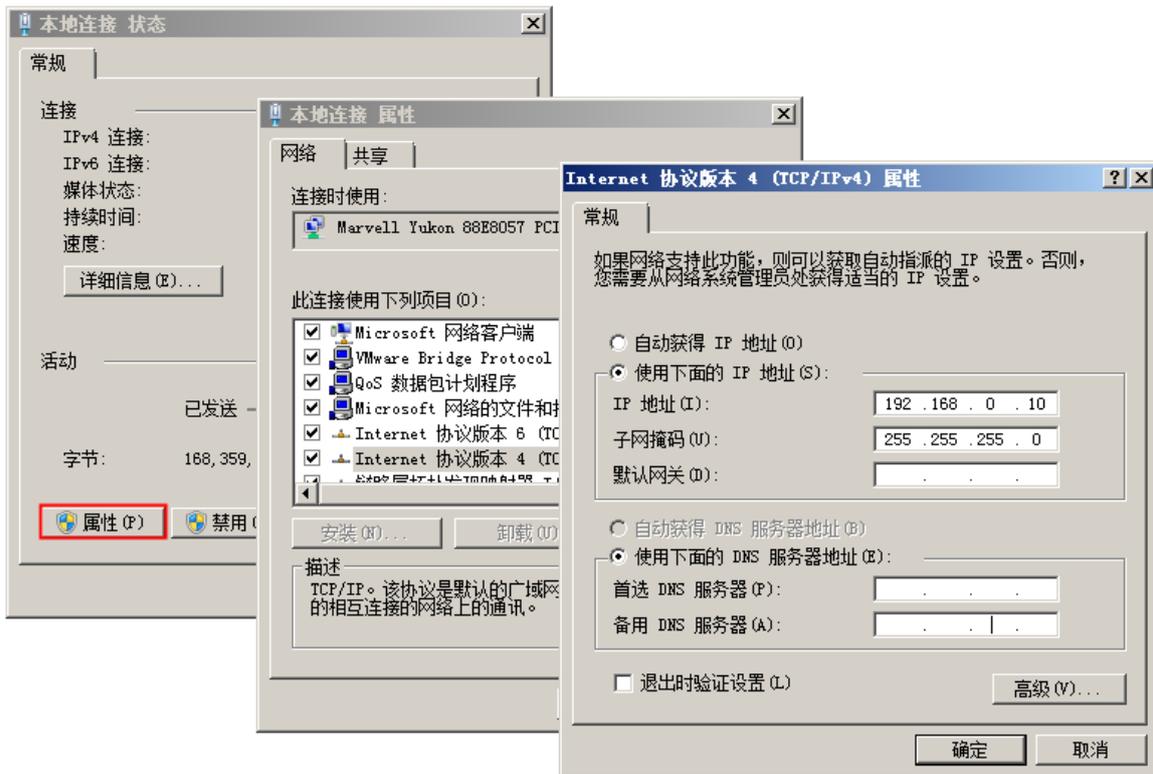


图 4.4-3 计算机 IP 地址设置

2. LE 平台

LE 工程“通讯设置”对话框多一个【PC 串口】标签页，进行串口通讯的参数设置。使用 PC 串口进行通讯的设置方法可参见章节 5.7.2 通讯设置的内容。



图 4.4-4 LE 工程通讯设置对话框

4.4.3 下装

通讯设置完成后，通过以下操作，将程序下装到控制器中。



- 菜单栏：单击【在线】—【下装】。

4.4.4 调试

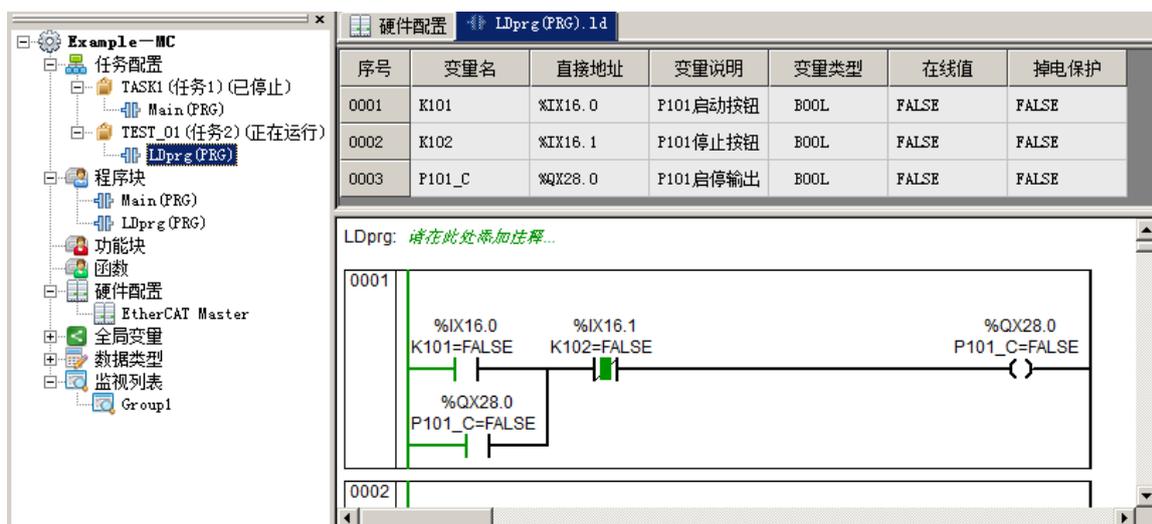
下装完成后，进行程序的调试。



- 菜单栏：单击【在线】—【监视】；
- 快捷键：**Alt+F8**；
- 工具栏：。

监视状态下，用户可以查看变量的在线值，可进行写入或强制操作。通过【在线】菜单下的【退出监视】可退出在线监视。

进入在线调试状态，【TEST_01】任务节点显示**正在运行**，双击该任务节点下的程序块 LDprg，观察数据变化或进行调试操作。如图 4.4-5 所示。



序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	K101	%IX16.0	P101启动按钮	BOOL	FALSE	FALSE
0002	K102	%IX16.1	P101停止按钮	BOOL	FALSE	FALSE
0003	P101_C	%QX28.0	P101启停输出	BOOL	FALSE	FALSE

LDprg: 请在此处添加注释...

0001

```

    %IX16.0          %IX16.1          %QX28.0
    K101=FALSE      K102=FALSE      P101_C=FALSE
    |               |               |
    ( )             ( )             ( )
  
```

0002

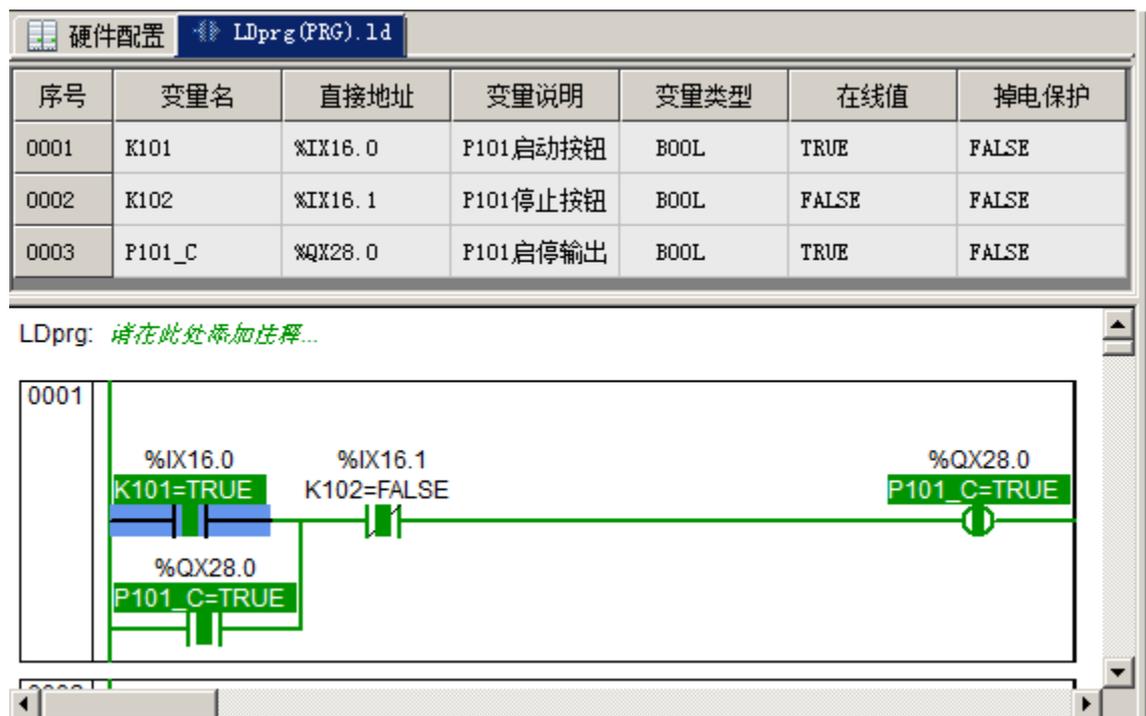
0003

图 4.4-5 程序调试界面

双击 K101 触点，弹出“调试变量”对话框，输入变量值为 TRUE，单击写入按钮。P101_C 变为 TRUE 输出到通道，启动 P101 泵。如图 4.4-6 所示。



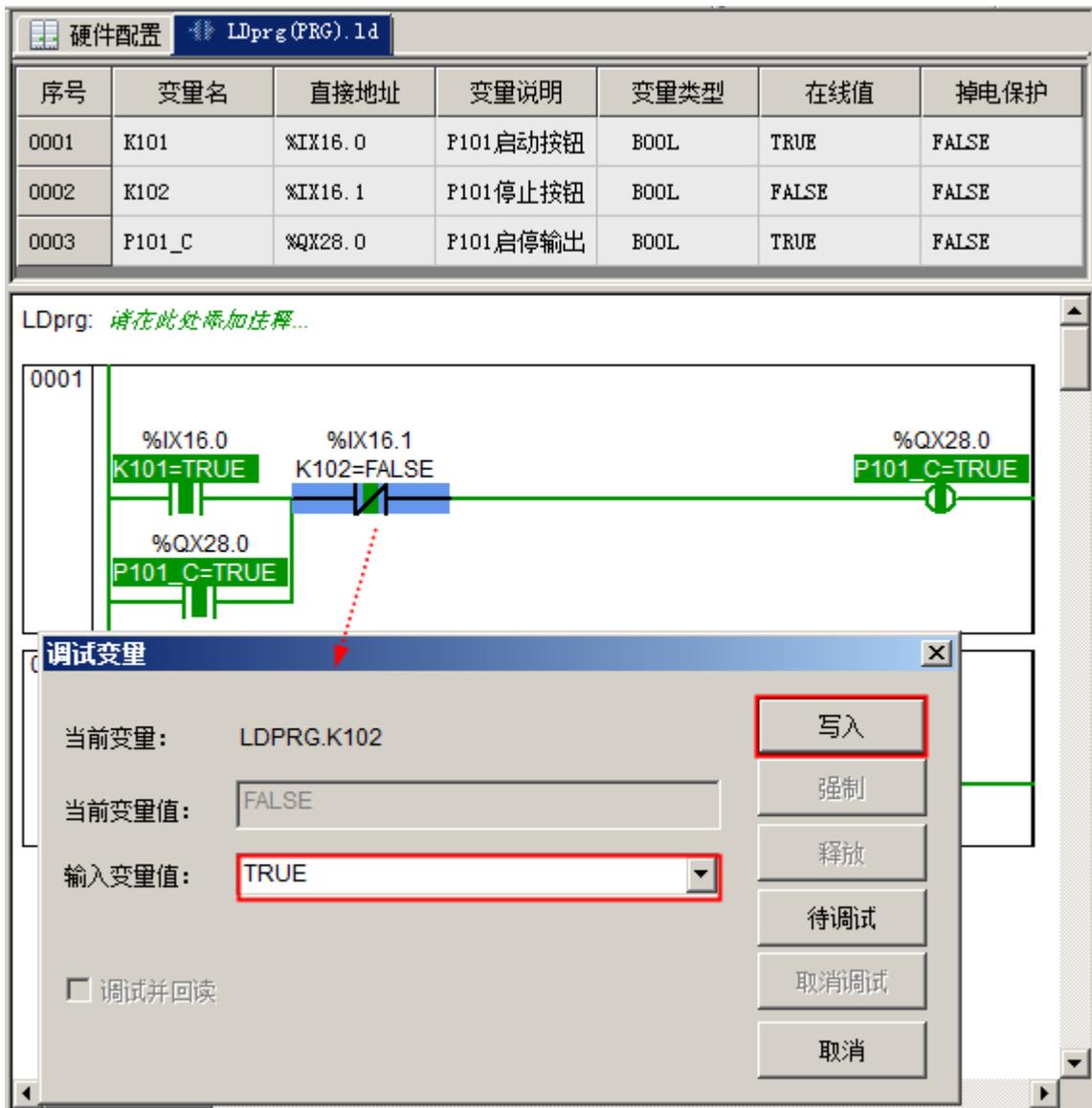
(a)



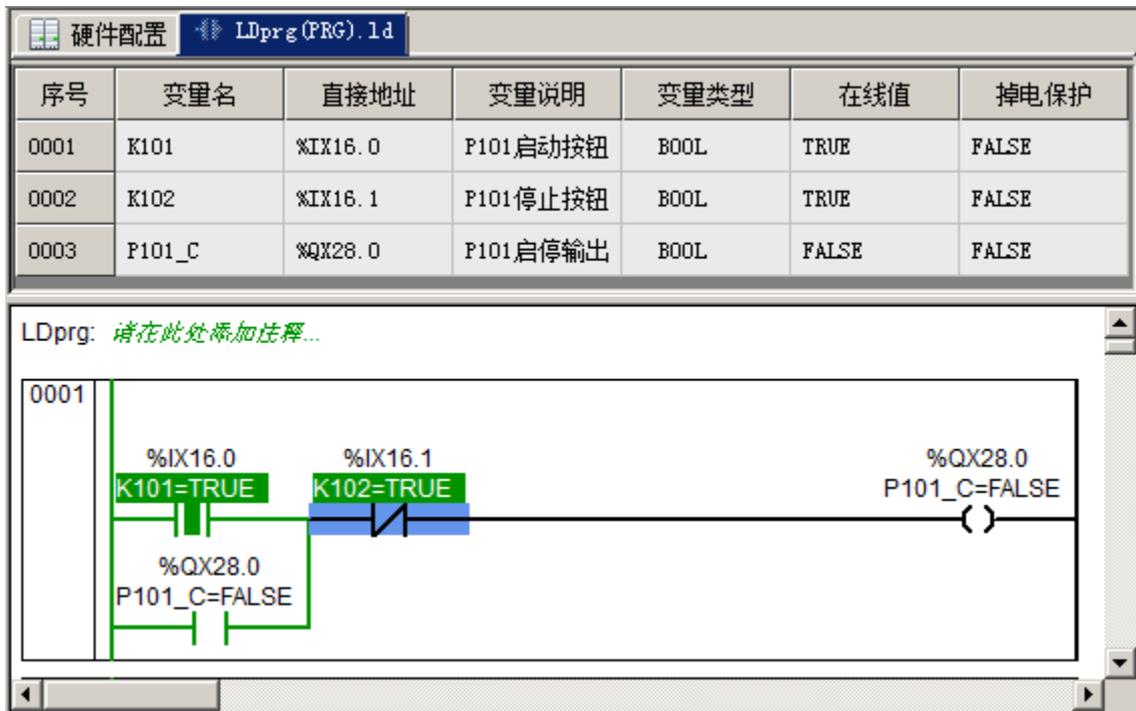
(b)

图 4.4-6 启动程序调试

同理，可调试 K102 触点为 TRUE，P101_C 线圈断开，向通道输出 FALSE 值，停 P101 泵。如图 4.4-7 所示。



(a)



(b)

图 4.4-7 停止程序调试

可以对任务进行【运行】、【停止】、【暂停】操作。使其分别进入正在运行、已停止、已暂停等任务状态，如图 4.4-8 所示。



图 4.4-8 任务的在线操作

第5章 AutoThink 软件组态介绍

5.1 软件界面

AutoThink 软件的编程界面组成如下：

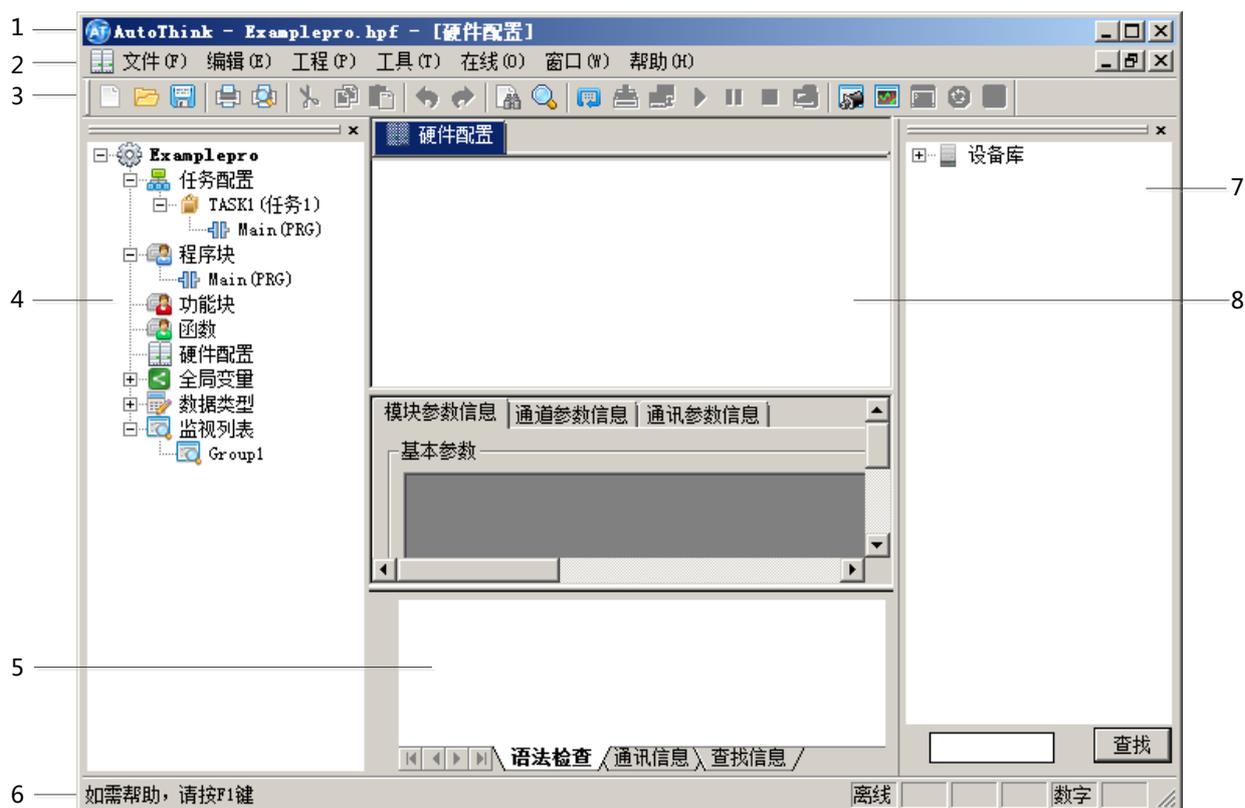


图 5.1-1 软件界面

1. 标题栏
2. 菜单栏
3. 工具栏
4. 工程管理树
5. 信息输出窗口
6. 状态栏
7. 设备库窗口

8. 工作区

5.1.1 标题栏

“标题栏”从左到右依次显示为：软件标识（软件图标和软件名称）、工程名称（工程状态）、正在编辑的对象（对象名称、类型）、窗口控制按钮（最小化、最大化、关闭按钮），如图 5.1-2 所示。



图 5.1-2 标题栏

1. 软件标识
2. 工程名称
3. 未保存标识
4. 编辑对象名称
5. 窗口控制按钮

当工程进行了修改之后，会在工程名称后面显示为“工程名.hpf*”，工程状态“*”表示修改后未保存，保存后“*”消失。

5.1.2 菜单栏

在标题栏下方的是菜单栏。通过单击各个菜单名称，在下拉菜单中选择各个菜单命令项进行算法组态的相关操作。菜单栏根据工作区域编辑内容的不同，显示的菜单也略有不同，一般包括以下几组菜单，如图 5.1-3 所示。



图 5.1-3 菜单栏

- 图标：显示当前打开的界面类型，包括：（CFC 语言）、（LD 语言）、（ST 语言）、（SFC 语言）、（硬件配置）、（全局变量）等。
- 文件：执行与工程文件有关的操作，例如打开、保存、关闭等。
- 编辑：对已有内容进行编辑、替换、查找等操作。
- 工程：对当前正在编辑的工程进行程序编译、设置、日志查看等操作。
- 插入：在组态程序块时显示该菜单，可以插入图形化编程语言的元件。
- 工具：提供各种指令向导和使用向导。

- 在线：提供工程在实际系统或仿真系统等操作的相关命令，如下装、在线、仿真等。
- 窗口：提供对当前打开窗口的排列方式、切换显示或关闭等操作的命令。
- 帮助：提供该软件的帮助信息。

5.1.3 工具栏

工具栏从左到右依次为主工具栏、在线工具栏、ST 工具栏（只有 ST 编辑器中可用）、MC 工具栏（只有在 MC 工程中可用）、编程语言工具栏。编程语言工具栏根据编程语言的不同，分别显示为 LD 工具栏、ST 工具栏、CFC 工具栏。如图 5.1-4 所示。鼠标放到工具栏图标上时，显示图标功能。

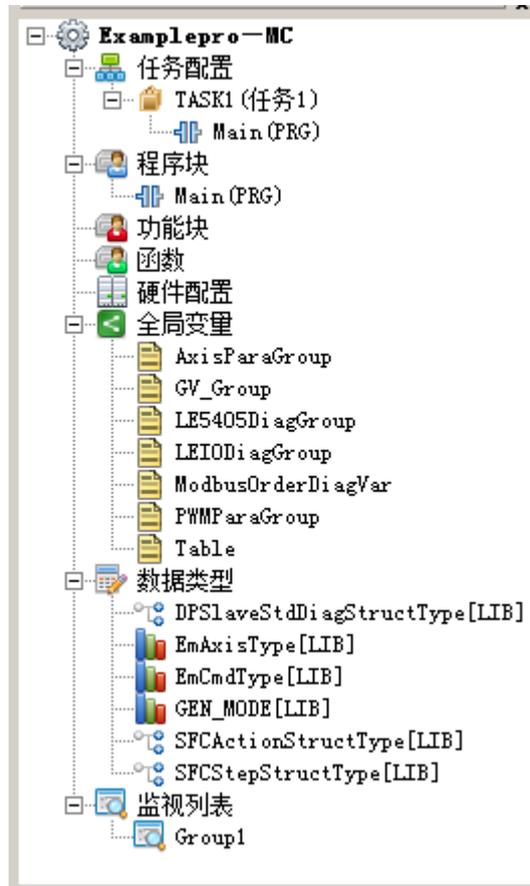


图 5.1-4 工具栏

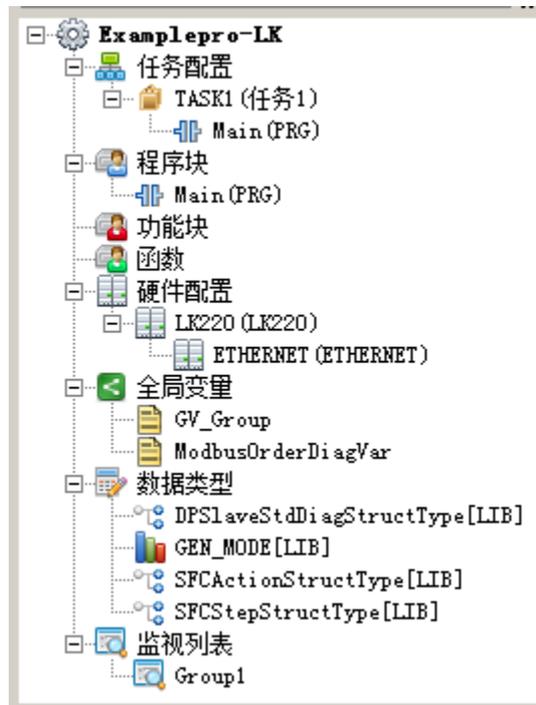
各工具栏的使用请参见对应的菜单命令，这里只列举 MC 工具栏。

5.1.4 工程管理

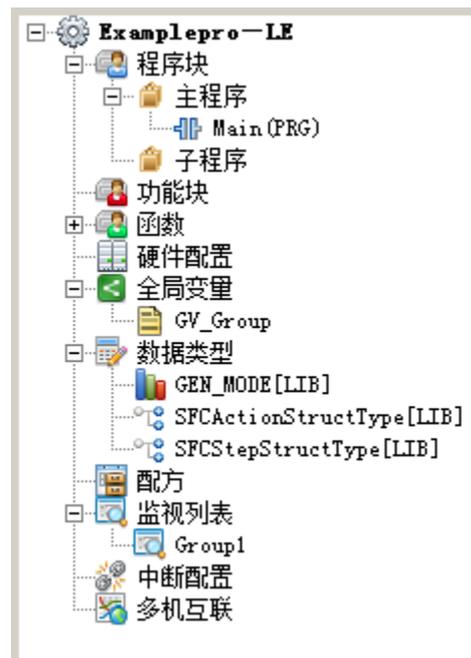
工程管理采用树状结构图进行相应内容的管理，该种管理方式使整个工程的管理层次清楚、结构清晰、管理方便。工程管理树状图的根节点名称为当前工程名称，下面包括各种子节点：任务配置、程序块、功能块、函数、硬件配置、全局变量、数据类型、监视列表、中断配置、多机互联等。MC、LK、LE 工程的工程管理树状图如图 5.1-5 所示。



(a)



(b)



(c)

图 5.1-5 工程管理树状图

5.1.5 信息输出窗口

信息输出窗口分为语法检查、通讯信息、查找信息 3 个选项卡，用于显示有关编译、错误、警告、通讯或查找的消息，如图 5.1-6 所示。

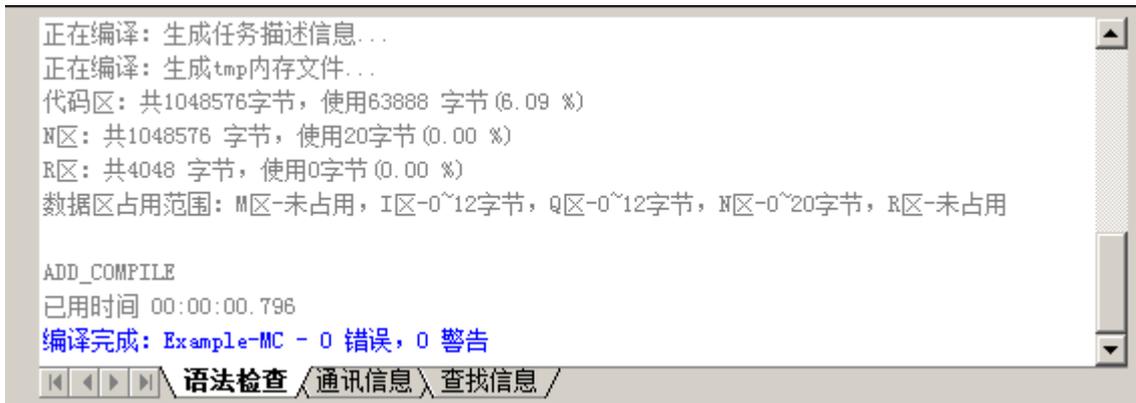


图 5.1-6 信息窗口

通过双击信息输出窗口中显示的编译错误、警告信息或者查找结果的所在行，可以自动定位到编辑器中的相关位置，以便查找相关信息。当选中信息窗口中的一个错误，按下 **Shift+F9** 可以快速定位上一个错误或警告。按下 **F9** 可以快速定位下一个错误或警告。

在操作过程中出现异常情况时，会在信息输出窗口中以红色字体提示异常信息。

在消息区，单击鼠标右键，可弹出菜单项 **清空所有信息**：清除当前正在显示的标签页中的全部信息。

隐藏/显示信息窗口：通过【窗口】菜单中的【信息窗口】命令控制是否显示信息窗口。信息窗口隐藏后，执行编译、查找操作均会自动加载显示信息窗口。

5.1.6 状态栏

状态栏显示与当前编辑窗口相关的信息，如图 5.1-7 所示。



图 5.1-7 状态栏窗口

- 1: 显示帮助提示信息；
- 2: 当编辑 ST 语言的 POU 时，状态栏显示当前光标所在的行号、列号信息；

- 3: 显示工程的在线、离线、仿真状态。仿真状态只有 LE 平台会显示;
- 4: LE 平台中, 显示工程的运行、停止状态;
- 5: 预留;
- 6: 预留;
- 7: 显示“数字”;
- 8: 预留。

5.1.7 设备库

“设备库”窗口中显示当前目标配置支持的所有模块型号, 供用户硬件配置时选择添加。

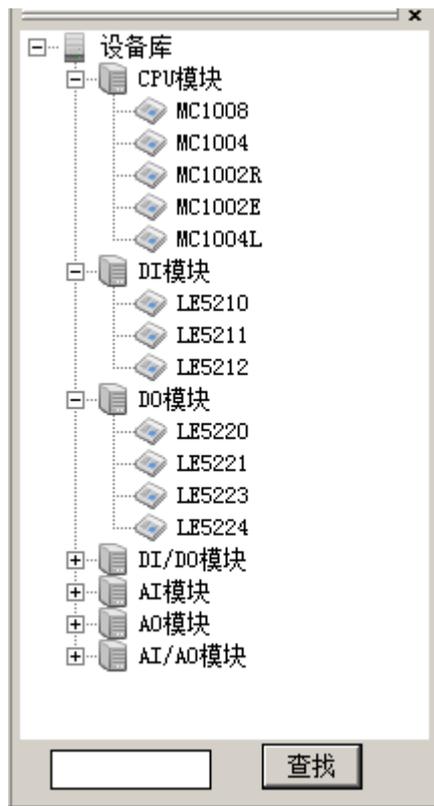


图 5.1-8 设备库窗口

5.1.8 工作区域

工作区域是指具体进行变量定义、程序编辑、硬件配置等操作的一块矩形区域, 如图 5.1-9 所示中红色矩形框标识的位置。集中显示各种编辑窗口, 在工作区域顶端以标签页的方式显示, 通过单击标签页进入相应的编辑窗口。

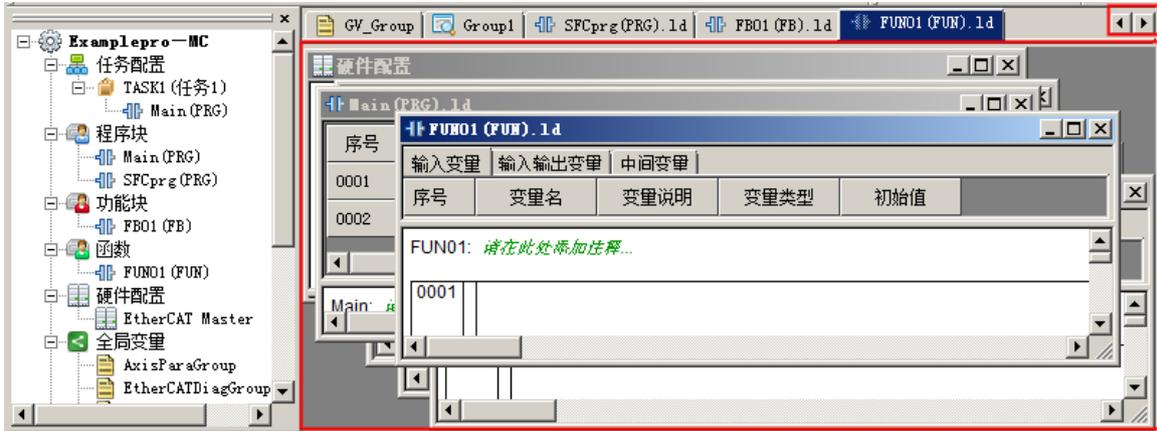


图 5.1-9 “工作区域”-各种编辑窗口

工作区域中的各个窗口的右上角都有一组窗口控制按钮，双击各个窗口的标题栏可以使窗口在浮动和固定状态间进行切换。

当编辑窗口为活动窗口时，单击窗口标题栏左侧的窗口图标，可以显示窗口控制下拉菜单，如图 5.1-10 所示，同样可以切换和控制窗口。

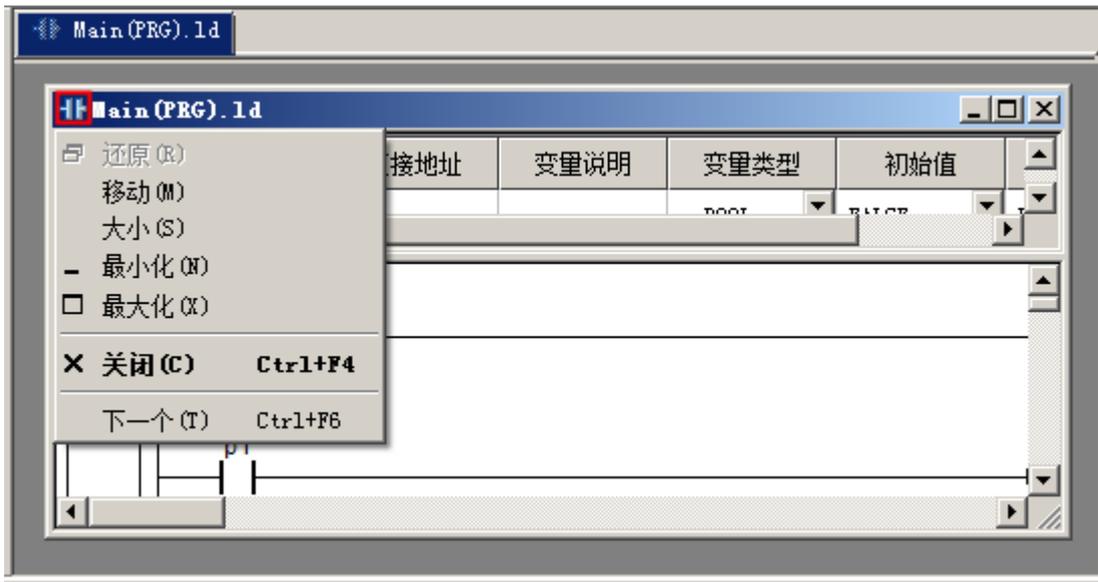


图 5.1-10 活动窗口的窗口控制菜单

5.2 工程管理

5.2.1 创建工程

此命令创建一个新的工程或库。



- 菜单栏：单击【文件】—【新建】；
- 工具栏：；
- 快捷键：**Ctrl+N**。

5.2.1.1 工程

选择【新建】命令后，系统弹出“新建”对话框，如图 5.2-1 所示。

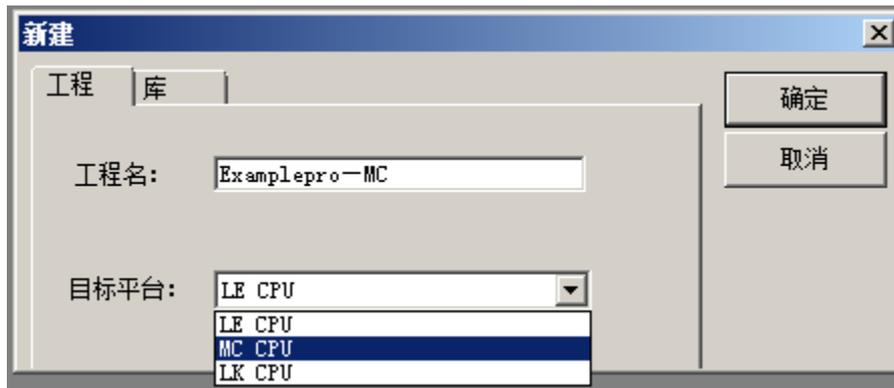


图 5.2-1 “新建”对话框—工程

- 工程名：缺省为“Untitled”。当保存此工程的时候，建议改变此文件名。
 - 工程名称中不能使用“\”、“/”、“:”、“*”、“?”、““”、“<”、“>”、“|”、“空格”这 10 种字符。
 - 工程名称不可为空。
 - 工程名称长度最多不超过 32 字节，超出范围的部分无法输入。工程名无效时，**确定**按钮不可操作。
 - 目标平台：可根据使用的 CPU 类型选择相应的平台，工程创建后将无法更改。
- 输入**工程名**，选择目标平台，单击**确定**按钮完成【工程】新建。

5.2.1.2 库

鼠标单击【库】标签页，对话框内容如图 5.2-2 所示。



图 5.2-2 “新建”对话框—库

- 库名：新建的库名称缺省为“Untitled”，在库名框中指定库的名称，命名规则与工程名的命名规则一致。
- 库版本：采用四段式版本号命名方式，每段设置的数字范围为 0~255。用户借助它可以记录库变更的版本（记录变更发生的日期，如“20.15.01.20”），也可以直接采用默认值。
- 目标配置：是一个配置集合，这里列举了目前所有的配置版本。选择不同的目标配置会影响工程的硬件配置信息、S 区参数和库文件。
- 目标平台：与【工程】中目标平台选择一致。

通常每发布一个软件版本都会升级一次目标配置，目前已有的目标配置与 AutoThink 软件版本的对应关系如表 5.2-1 所示。

表 5.2-1 目标配置版本与 AutoThink 软件版本的对应关系

序号	目标配置版本	AutoThink 软件版本
1	V3.1.1B1	V3.1.1B1
2	V3.1.1B2	V3.1.1B2、V3.1.1 正式版
3	V3.1.2B1	V3.1.2B1
4	V3.1.2B2	V3.1.2B2、V3.1.2 正式版、V3.1.2SP1
5	V3.1.3B1	V3.1.3B1
6	V3.1.3B1Patch1	V3.1.3B1Patch1
7	V3.1.4B1	V3.1.4B1
8	V3.1.4B2	V3.1.4B2
9	V3.1.4B3	V3.1.4B3

序号	目标配置版本	AutoThink 软件版本
10	V3.1.4B3	V3.1.4B4
11	V3.1.4B5	V3.1.4B5

在新建工程、工程升级时需要设置该项。由于目标配置与工程的硬件配置等信息有关，所以用户需要根据实际的控制器型号选择目标配置。选择时可以参考表 5.2-2。

表 5.2-2 目标配置版本与支持的控制器型号

序号	目标配置版本	控制器型号
1	V3.1.1B1	LE5104 LE5105 LE5106-A03 及以下版本 LE5107 LE5108-A06 及以下版本 LE5109
2	V3.1.1B2	LE5104 LE5105 LE5106-A03 及以下版本 LE5107 LE5108-A06 及以下版本 LE5109
3	V3.1.2B1	LE5104 LE5105 LE5106-A03 及以下版本 LE5107 LE5108-A06 及以下版本 LE5109 LE5107E LE5107L LE5109L LE5128 LE5706 LE5707
4	V3.1.2B2	LE5104 LE5105 LE5106-A03 及以下版本 LE5107 LE5108-A06 及以下版本 LE5109 LE5107E LE5107L LE5109L LE5128 LE5706 LE5707 LE5710
5	V3.1.3B1	LE5104 LE5105 LE5106-A03 及以下版本 LE5107 LE5108-A06 及以下版本 LE5109

序号	目标配置版本	控制器型号
		LE5107E LE5107L LE5109L LE5128 LE5706 LE5707 LE5710 MC1008-A01 (软件版本 V3.1.4B1 及以上不兼容)
6	V3.1.3B1Patch1	LE5104 LE5105 LE5106-A04 及以上版本 LE5107 LE5108-A07 及以上版本 LE5109 LE5107E LE5107L LE5109L LE5128 LE5706 LE5707 LE5710 MC1008-A01 (软件版本 V3.1.4B1 及以上不兼容)
7	V3.1.4B1	LE5104 LE5105 LE5106-A04 及以上版本 LE5107 LE5108-A07 及以上版本 LE5109 LE5107E LE5107L LE5109L LE5128 LE5706 LE5707 LE5710 LE5708 MC1008-B01 MC1004-A01
8	V3.1.4B2	LE5104 LE5105 LE5106-A04 及以上版本 LE5107 LE5108-A07 及以上版本 LE5109 LE5107E LE5107L LE5109L LE5128 LE5706 LE5707 LE5710 LE5708 MC1008-B01 MC1004-A01 MC1002R-A01
9	V3.1.4B3	LE5104 LE5105

序号	目标配置版本	控制器型号
		LE5106-A04 及以上版本 LE5107 LE5108-A07 及以上版本 LE5109 LE5107E LE5107L LE5109L LE5128 LE5706 LE5707 LE5710 LE5708 MC1008-C01 MC1004-B01 MC1002R-A02 MC1002E-A01 MC1004L-A01 LK220
10	V3.1.4B5	LE5104 LE5105 LE5106-A04 及以上版本 LE5107 LE5108-A07 及以上版本 LE5109 LE5107E LE5107L LE5109L LE5128 LE5706 LE5707 LE5710 LE5708 MC1008-C02 MC1004-B02 MC1002R-A02 MC1002E-A01 MC1004L-A01 LK220

同一工程在 AutoThink 软件版本、目标配置发生变化时，该工程的应用情况也随之变化，变化的具体内容请参见表 5.2-3。

表 5.2-3 目标配置版本与软件版本变更时对工程的影响

目标配置	AutoThink 软件版本	结果
不变	变化	工程可以使用当前软件版本的所有功能（与硬件模块无关的）
变化	不变	工程需要配合对应版本兼容的硬件模块使用
变化	变化	同时具备以上描述

新建库完成后，在库节点上单击右键，选择【属性】，如图 5.2-3 所示，可以查看该库的属性信息。



图 5.2-3 库管理节点



- LE 平台创建的库中不支持使用 STRING 类型常量。

5.2.2 打开工程

此命令打开一个已存在的工程或库。



- 菜单栏：单击【文件】—【打开】；
- 工具栏：；
- 快捷键：**Ctrl+O**。

系统弹出“打开”对话框，选择工程或库的目标文件，如图 5.2-4、图 5.2-5 所示，单击**打开**按钮后软件加载该文件，同时在状态栏显示打开进度。

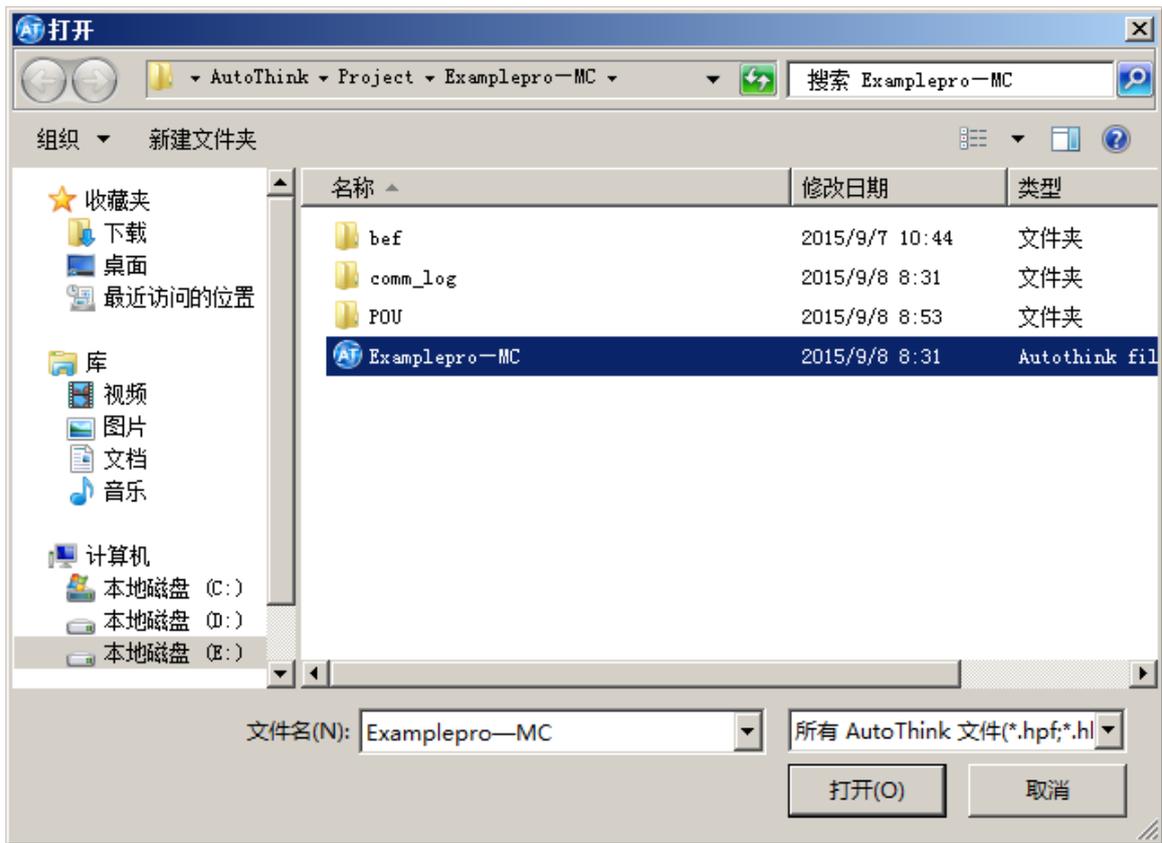


图 5.2-4 “打开”工程对话框



图 5.2-5 “打开”库对话框

5.2.3 工程属性



- 工程管理树：右击根节点（工程名称），单击【属性】。

打开当前工程的配置信息，如图 5.2-6 所示。

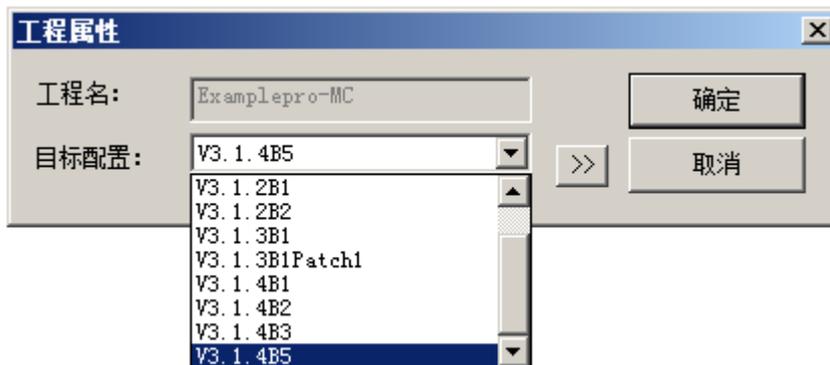


图 5.2-6 工程管理根节点-工程属性

目标配置：指当前 CPU 模块支持的 AutoThink 软件版本，用户可以在不同的版本之间进行切换。每个软件版本支持的功能会有所区别，请切换前了解版本的相关信息。

通过  按钮可以查看该工程的具体内容，如图 5.2-7 所示。



图 5.2-7 工程配置信息

数据区详细信息参见章节 [5.5.1.2 数据的存储的数据存储区](#)。

5.2.4 保存工程

用此命令保存在工程或库中所做的更改。



- 菜单栏：单击【文件】—【保存】；
- 工具栏：；
- 快捷键：**Ctrl+S**。

如果工程或库以 **Untitled** 保存，请务必更改名字。如图 5.2-8、图 5.2-9 所示，缺省保存到安装路径下的对应文件夹下，保存时会在状态栏显示保存进度。

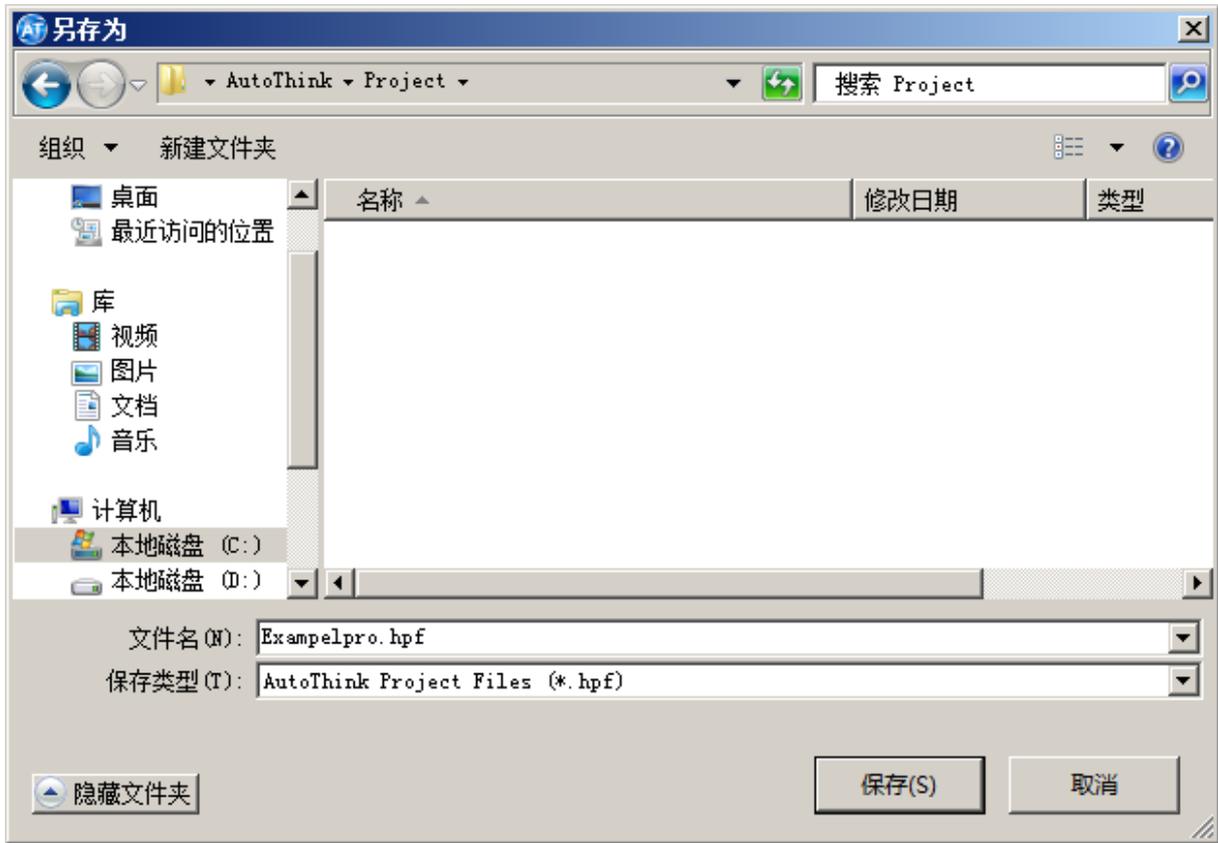


图 5.2-8 保存工程



图 5.2-9 保存库

用户变更当前工程或库的保存路径时，可以通过【另存为】命令实现。



- 菜单栏：单击【文件】—【另存为】；
- 快捷键：**F12**。

当选择此命令时，弹出“另存为”对话框。如图 5.2-8、图 5.2-9 所示。选择一个已存在的文件名或输入一个新文件名，选择文件类型，单击**确定**按钮完成另存操作。

5.2.5 关闭工程



- 菜单栏：单击【文件】—【关闭】。

使用此命令关闭当前打开的工程或库，并不退出 AutoThink 软件。

如果工程或库已经被更改，该系统会提示是否保存当前工程或库。

5.2.6 最近工程列表



- 菜单栏：单击【文件】—【最近工程列表】。

显示最近打开过的工程的列表，包括工程路径和工程名称。通过此命令可以快速切换工程。

5.2.7 查找



- 菜单栏：单击【编辑】—【查找】；
- 工具栏：；
- 快捷键：**Ctrl+F**。

此命令可以在当前编辑器或整个工程中查找某一文本，打开“查找”对话框，如图 5.2-10 所示，在**查找内容**输入框中输入要查找的字符序列，此外，可以设置搜索选项：是否全字匹配、是否区分大小写，搜索整个工程或是当前页面。



图 5.2-10 “查找”对话框

单击**确定**按钮后开始查找，查找均从当前页面开始。查找结果显示在信息窗口的【查找信息】栏中，如图 5.2-11 所示，双击查找结果项，将定位到查找文本所在的位置。

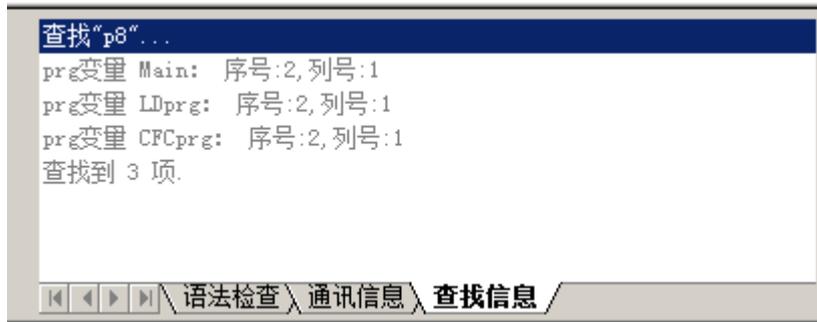


图 5.2-11 “查找”结果

5.2.8 替换



- 菜单栏：单击【编辑】—【替换】；
- 快捷键：**Ctrl+H**。

使用此命令可以如同【查找】命令一样查找某一个文本，且用另一个代替它。打开“替换”对话框，如图 5.2-12 所示。



图 5.2-12 “替换”对话框

在**查找内容**输入框中输入要查找的字符序列，在**替换为**输入框中输入替换内容，设置**搜索选项**和**搜索范围**后，单击**替换**按钮，则在当前页面或整个工程中替换符合查找条件的字段，每次替换一处，再次单击**查找下一个**，继续向下查找替换；若单击**全部替换**，则替换当前页面或整个工程中所有符合查找条件的内容，并在替换完成后，弹出提示，显示完成的替换总次数。

5.2.9 工程权限

用户可以对工程的操作权限进行设置，权限分为读写和只读两种。

- 读写权限：指用户对工程进行的有效编辑和操作不受约束，该权限打开工程的标题栏样式如图 5.2-13 所示。



图 5.2-13 读写权限的标题栏

- 只读权限：指用户只可以对工程进行查看或监视，该权限打开工程的标题栏上有 **ReadOnly** 标识，如图 5.2-14 所示，只读权限下，工程离线状态时不支持编辑，已下装成功的工程在线状态时只允许监视。



图 5.2-14 只读权限的标题栏



- 菜单栏：单击【工程】—【权限设置】。

需要进行权限设置时，单击此命令，弹出权限设置对话框，如图 5.2-15 所示。



图 5.2-15 “权限设置”对话框

新密码和确认新密码位置输入内容必须完全相同，密码长度为 6~12 个字符，且区分大小写。

设置成功后，在下次打开该工程时，会提示用户输入相关密码，如图 5.2-16 所示，可输入读写权限或只读权限密码，以不同权限打开工程，如果连续 3 次密码验证错误，则打开工程失败。



图 5.2-16 密码输入框

5.2.10 工程设置



- 菜单栏：单击【工程】—【选项】。

打开“选项”对话框，该选项被分成几大类。每个选项会对应右边不同的设置，如图 5.2-17 所示。

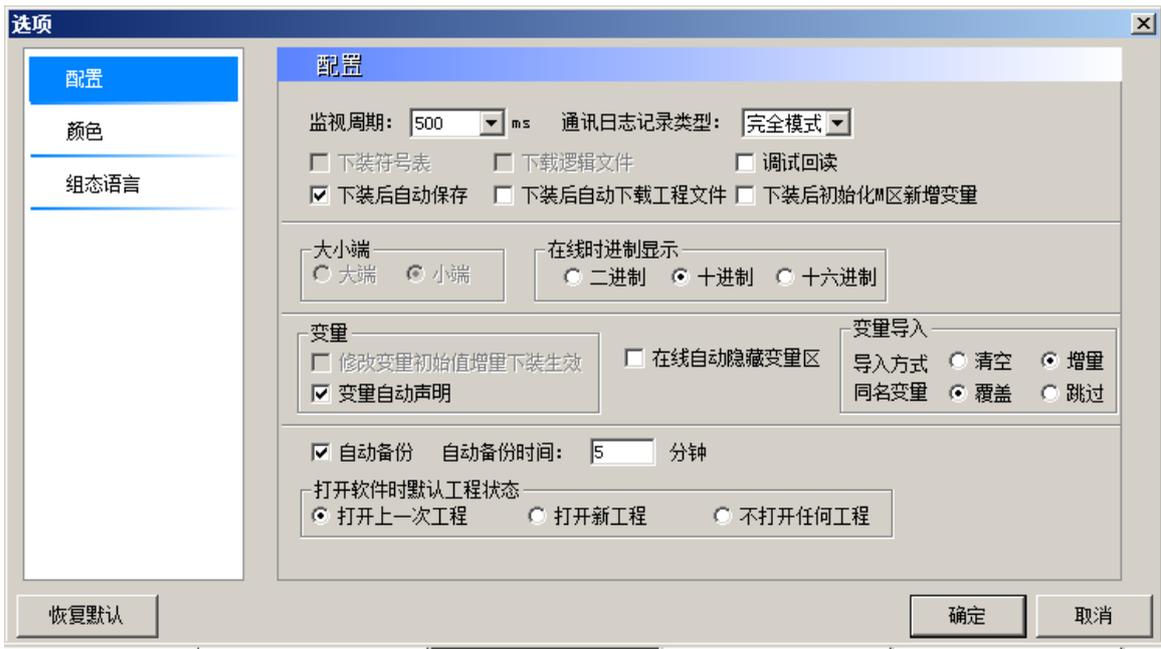


图 5.2-17 “选项”对话框

选项包括：配置、颜色、组态语言。下面依次介绍：

5.2.10.1 配置

配置对话框显示工程相关的监视周期、日志、在线显示、变量、保存等设置情况，具体如下：

- 监视周期：缺省为 500 ms，250 ms、500 ms、1000 ms 可选，可根据需要进行选择。

- 通讯日志记录类型：设置通讯日志是否进行记录以及记录的格式，缺省为**完全模式**。**完全模式**下可记录详细的通讯信息。选择**精简模式**只记录通讯信息名字，不记录码值等信息。
- 下装符号表：将工程编译生成的符号表下装指定的通讯站，缺省未选中，该设置不可以修改。
- 下载逻辑文件：将编译通过的工程文件下装指定的通讯站，缺省未选中，该设置不可以修改。
- 下装后初始化 M 区新增变量：设置下装后是否初始化 M 区的新增变量，MC、LK 工程可选，编译后生效。
- 下装后自动保存：设置工程下装操作后是否进行自动保存，缺省不勾选**下装后自动保存**，该设置可以修改。
- 下装后自动下载工程文件：设置工程下装操作后是否下载工程文件到控制器，缺省为不勾选，该设置可以修改，勾选后，**下装后自动保存**被强制勾选。
- 调试回读：设置在调试时是否可以进行数据回读到控制站工程中，缺省不勾选**调试回读**，该设置可以修改。
- 大小端选择：指数据存储模式，缺省为**小端模式**，该设置不可以修改。
- 在线时进制显示：在线或仿真状态下，设置在线显示的数据格式，可以选择：**二进制**、**十进制**或**十六进制**，缺省设置为**十进制**显示，该设置可以修改。
- 变量：设置变量的增量下装和自动声明设置。
 - 9: 修改变量初始值增量下装生效：当修改了变量的初始值时，触发增量下装。缺省为未选中，该设置不可修改。
 - 10: 变量自动声明：设置在程序区新建变量时，是否自动弹出声明对话框。选中状态表示会自动弹出，反之，不弹出。
 - 11: 在线自动隐藏变量区：在线情况下，设置是否自动隐藏 POU（PRG）中的变量区。缺省不勾选该项。
- 变量导入：设置变量的导入方式，以及增量导入时同名变量的导入方式。
 - 1: 导入方式：分**清空**和**增量**两种导入方式。选择**清空**方式，导入变量时，先清空对应 POU 或全局变量组中的变量，再执行导入。选择**增量**方式，导入变量时，仅导入增量变量，同名变量进行覆盖或跳过两种方式导入。
 - 2: 同名变量：当导入方式为**增量**时可设。选择**覆盖**，导入变量时，会覆盖工程中的同名变量，并在信息输出窗口提示“XX 组：X 被覆盖！”。选择**跳过**，导入变量时，同名变量不做导入，并在信息输出窗口提示“XX 组：X 被跳过！”。
- 自动备份：离线状态下，设置工程自动备份的时间和时间间隔，用于软件异常关闭时的工程备份。
- 打开软件时默认工程状态：设置在启动软件时打开的工程，可以选择：打开上一次工程、打开新工程或不打开任何工程，缺省设置为**打开上一次工程**，该设置可以修改。

5.2.10.2 颜色

选择该项后，显示与程序中颜色相关的参数设置，如图 5.2-18 所示。在此对话框中显示了当前软件中缺省定义的各种标准颜色，用于区分在程序定义时使用的不同对象颜色，可以通过下拉颜色菜单进行修改，使组态内容更易区分。

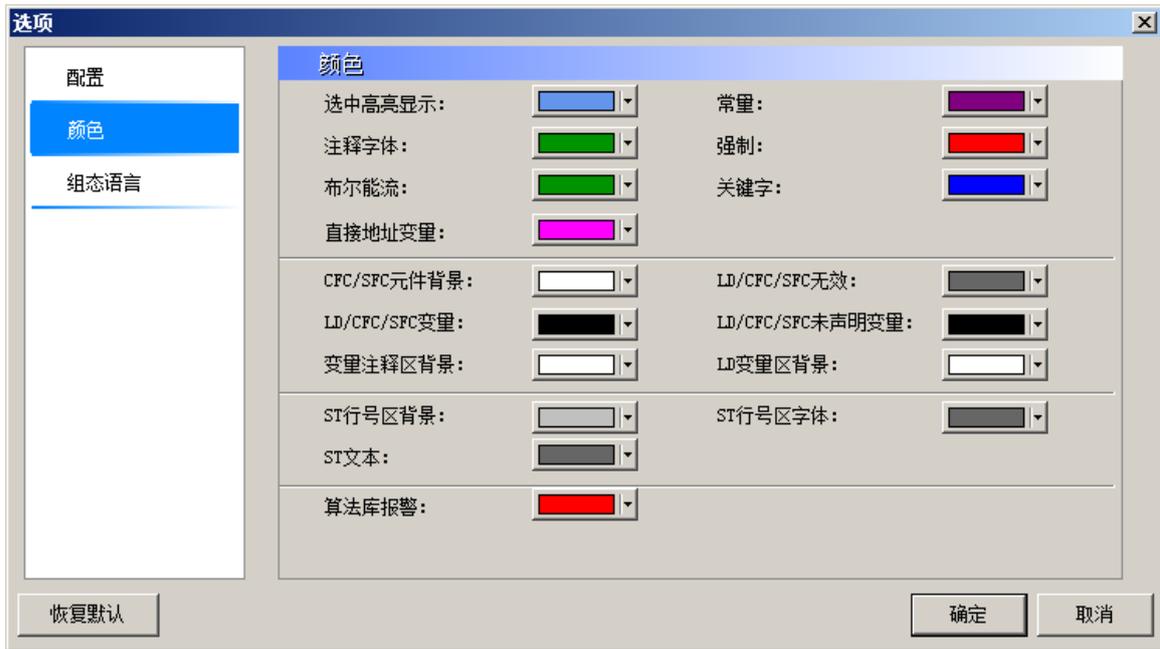


图 5.2-18 “颜色”的对话框



- 工程中实际使用的各种颜色标识均可以根据实际工程需要进行调整，本文中涉及的颜色描述均以缺省设置的颜色为准，不再赘述。

5.2.10.3 组态语言

该项设置各种 IEC 组态语言的显示特性，以及各个语言特有的显示特性，如图 5.2-19 对话框所示。



图 5.2-19 “IEC 编辑”的对话框

■ IEC

- 编译时保护：设置是否进行**除零保护**和**数组越界保护**。**除零保护**，缺省为选中状态，控制器中出现除零运算时，运算结果保持上一周期的值。勾选**数组越界保护**，当数组的下标超上限或下限时，运算结果保持上一周期的值。
- 显示注释：显示普通类型、功能块类型变量的**变量说明**的内容。
- 打印方式：设置打印方向：**分页打印**、**单页打印**。缺省设置为**分页打印**。

■ LD：显示与 LD 语言相关的参数设置。

- 显示条颜色：设置 LD 语言组态的程序块中的地址名的背景颜色。
- 显示直接地址：默认设置添加一个触点或线圈后，显示地址。
- 输入地址后由变量替代：在变量区定义变量 A1，直接地址为%MX100.0，在程序区定义一个触点，关联%MX100.0，回车（单击）后，显示 A1。

■ CFC：显示与 CFC 语言相关的参数设置。

- 碰撞检测：检测在使用 CFC 语言编写程序块时，各个元件是否允许互相重叠。当不检测碰撞时，如果两个或两个以上元件重叠后，后放置的元件将部分或完全遮盖掉先放置的元件。
- 使能：选中该项，则当使用 CFC 语言编写程序块，将添加带使能输入输出端的块元件，缺省该项为不选中状态。
- 显示网络：选中该项，则当使用 CFC 语言编写程序块，在编程区显示网格标记，否则编辑区显示为空白区域，缺省该项为不选中状态。

- ST: 显示与 ST 语言相关的参数设置。
 - 在线显示字符数: 设置在线时, 变量名及运算结果显示的字符数。缺省设置为 24 个, 显示的长度范围为 20~100 个字符。
- 恢复默认

单击**恢复默认**按钮, 弹出提示框如图 5.2-20 所示。



图 5.2-20 恢复默认选项设置提示框

单击**确定**按钮, 选项中的参数恢复系统缺省设置值。

5.2.11 下载/上传工程源文件



- 菜单栏: 单击【在线】—【下载工程源文件】/【上传工程源文件】。

下载工程源文件是将当前打开的控制站工程的工程文件(*.hpf), 从当前操作员站所在的计算机下载备份到相应站号的 CPU 模块上。

上传工程源文件将当前打开的控制站工程的工程文件(*.hpf)备份, 从 CPU 模块上传至当前操作员站所在计算机的指定路径下。

【上传工程源文件】和【下载工程源文件】命令, 即是对当前打开的工程文件执行在本地和相应的 CPU 模块上进行拷贝的一对相互操作, 目的是将下装后的工程文件(*.hpf)备份到相应的 CPU 模块上, 以便留有备份, 在本地文件丢失, 或不确定与当前 CPU 模块中运行的工程文件版本是否一致时, 保持工程文件的一致性、准确性。

5.2.12 打印预览

使用此命令可以预览当前活动窗口的打印效果。



- 菜单栏：单击【文件】—【打印预览】；
- 工具栏：。

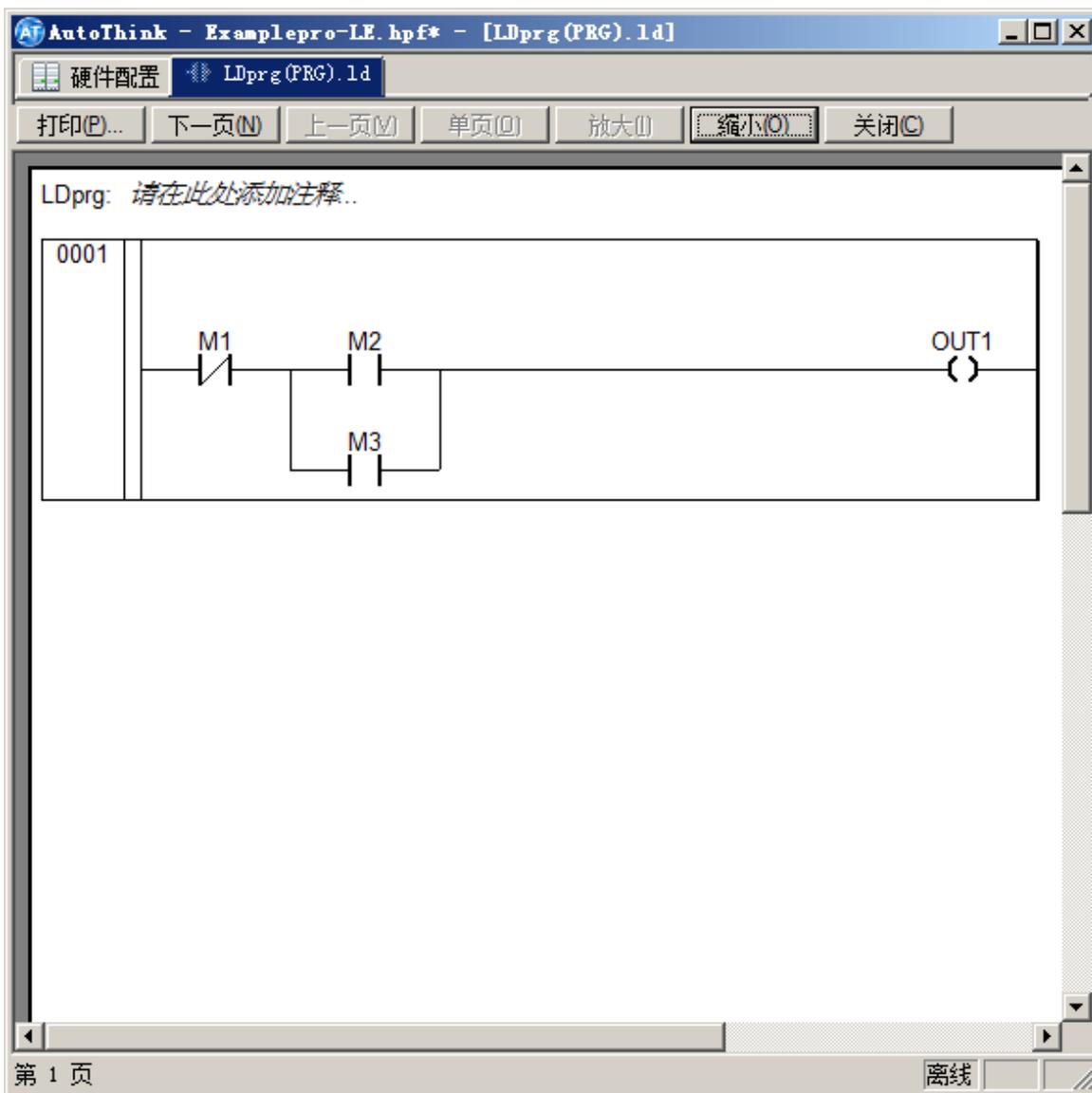


图 5.2-21 “打印预览”窗口

该打印预览页面可以对当前页面进行放大、缩小，当多页时，单击上一頁或下一頁向前向后翻页。

5.2.13 打印(P)

此命令可以打印活动窗口的内容。



- 菜单栏：单击【文件】—【打印】；
- 工具栏：；
- 快捷键：**Ctrl+P**。

弹出“打印”对话框，如图 5.2-23 所示。选择打印机名称，设置页面范围、打印份数，单击确定按钮，活动窗口被打印。

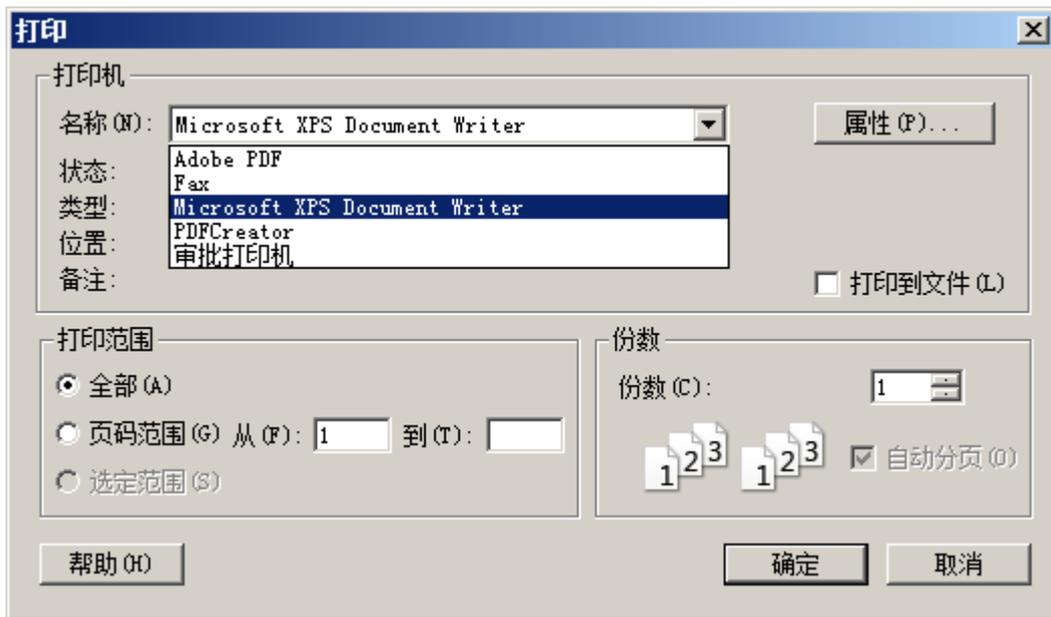


图 5.2-22 打印对话框

打印份数超过 1 页时，可设置自动分页。还可以将打印内容输出到文件。

5.2.14 窗口(W)

【窗口 (W)】菜单提供用于设置工作区各个窗口的布局方式的命令，隐藏或显示指定的窗口，且提供可以切换显示或最近打开的窗口命令，如图 5.2-23 所示，下面分别介绍具体功能。



图 5.2-23 “窗口”菜单

窗口菜单中的操作命令及其功能如表 5.2-4 所示。

表 5.2-4 窗口菜单项说明

菜单命令	说明
层叠	所有窗口在工作区串联排列，一个叠一个，一次错开
水平	所有窗口在工作区水平排列，不重叠，且充满整个工作区
垂直	所有窗口在工作区垂直排列，不重叠，且充满整个工作区
全部关闭	关闭工作区域当前打开的所有窗口

5.2.14.1 常用窗口

该区域列出了常用的辅助窗口：**【工程信息窗口】**、**【信息输出窗口】**、**【库管理窗口】**、**【设备库窗口】**、**【状态栏】**，通过勾选来显示相应窗口。

在**【窗口】**下拉菜单的底端显示当前工作区域打开的各个编辑窗口名称，单击窗口名称可以切换为当前的活动窗口。

5.2.15 多机互联

仅 LE 特有功能。

多机互联用于设置整个工程中网络变量的可用范围、地址范围以及是否启动网络功能。

通过配置多机互联，可以建立多个控制站间的数据互访。

在 LE 工程管理树中，双击【多机互联】节点，显示“多机互联”对话框，如图 5.2-24 所示。



图 5.2-24 “多机互联”对话框

- 网络总节点数：选择整个工程支持的网络站点个数，范围为 2~16。
- 当前节点号：选择当前工程所占用的网络节点号，该节点号的可设置范围与所选的**网络总节点数**关联，设置数值必须小于等于**网络总节点数**。
- 扫描时间：设置站之间发送广播的延迟时间，单位为毫秒。
- 启用多机互联：设置是否启动多机互联功能。

单击 ，展开设置的各个站号占用的内存区域范围，如图 5.2-25 所示。



图 5.2-25 “多机互联”详细参数对话框

每个网络站点为网络变量分配 30 个字节的存储区域。内存区域范围是 WORD 型地址，每个 WORD 字占用两个字节，如图 5.2-25 所示，1 号站范围为%SW300~%SW328，2 号站范围为%SW330~%SW358，2 号站的起始字节为%SW330，即%SW328+2。

所有已配置多机互联的工程，可以通过访问其他站号的内存地址来读取数据。%SW300~%SW778 内存数据为只读数据。%SW270~%SW298 内存数据为可读写数据，多机互联的网络节点可进行数据的读写。

5.2.16 帮助(H)

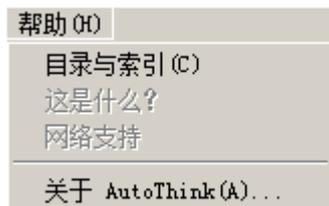


图 5.2-26 帮助菜单

- 目录与索引：该命令可以打开 AutoThink 在线帮助。
- 关于 AutoThink：该命令可以查看软件的版本和版权信息。

5.2.17 编辑(E)

5.2.17.1 撤销



- 菜单栏：单击【编辑】—【撤销】；
- 工具栏：；
- 快捷键：**Ctrl+Z**。

此命令取消在当前编辑器中最近执行的动作。重复执行此命令，所有的动作能被执行到窗口打开那一次，这适用于当前打开的编辑器中的所有的动作。

5.2.17.2 恢复(E)



- 菜单栏：单击【编辑】—【恢复】；
- 工具栏：；
- 快捷键：**Ctrl+Y**。

此命令可以在当前打开的编辑窗口中恢复已经撤销的动作。

5.2.17.3 剪切(T)



- 菜单栏：单击【编辑】—【剪切】；
- 工具栏：；
- 快捷键：**Ctrl+X**。

此命令把当前选中的对象从编辑器传送到剪贴板，并把选中的对象从编辑器中删去。

5.2.17.4 复制(C)



- 菜单栏：单击【编辑】—【复制】；
- 工具栏：；
- 快捷键：**Ctrl+C**。

此命令把当前选中的对象从编辑器拷贝到剪贴板，不会改变编辑器窗口的内容。对于工程树此命令只适用于被选中的对象，然而并不是所有的对象都可被拷贝，例如硬件配置和任务。

5.2.17.5 粘贴(P)



- 菜单栏：单击【编辑】—【粘贴】；
- 工具栏：；
- 快捷键：**Ctrl+V**。

把剪贴板的内容粘贴到编辑器的当前位置。

5.2.17.6 删除



- 菜单栏：单击【编辑】—【删除】；
- 快捷键：**Delete**。

从编辑窗口中删除选中的区域，这不会改变剪贴板的内容。在工程树中，这同样适用于选中的对象，然而不是所有的对象都可被删除的，例如：硬件配置。

5.3 程序组织单元

程序组织单元（Program Organization Unit，简称 POU）是 AutoThink 软件的基本构成元素，用户基于程序组织单元编写应用程序，POU 包括程序（Program）、功能块（Function Block）和函数（Function）三种类型。程序可以调用功能块和函数，构成用户的应用程序。

任务主要进行程序的调度。所有的程序在任务节点下调用才可以执行。用户可以根据实际需要配置任务的属性。

控制站算法支持所有符合 IEC 标准的 POU。

本章主要介绍任务组态、POU 的类型、POU 的添加、POU 的调用及操作。

5.3.1 POU 的类型

POU 有三种类型：程序、功能块、函数。

每个 POU 都由变量声明和算法编程两部分组成。

声明部分：局部变量声明在 POU 编辑器中的变量区进行变量的定义。

算法部分：即 POU 主体，在编辑器中的程序区选择 LD、ST、CFC，SFC 四种不同的语言，进行算法编程操作。

有关各种语言编辑器，请详见章节 5.4 编程语言。

POU 结构示意图如图 5.3-1 所示。

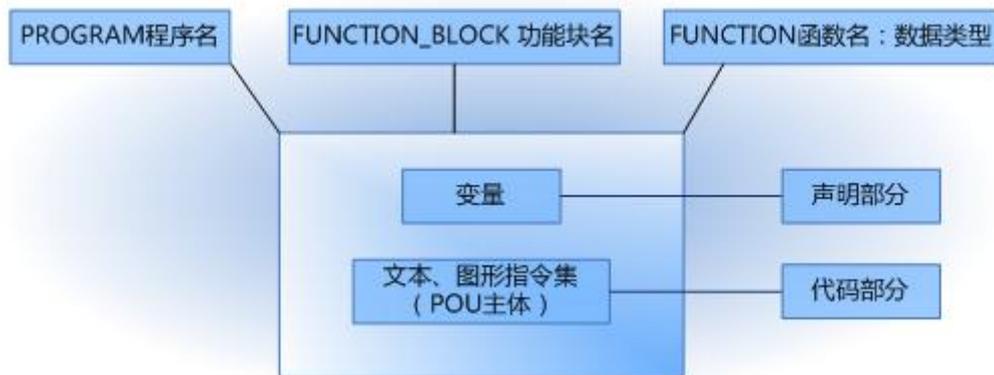


图 5.3-1 POU 的结构图

■ 程序 (Program)

程序是为了完成某项任务而编写的语句序列，或一组指令的集合。

AutoThink 软件中将程序分为子程序（也称为程序块）和主程序（也称为任务）。前者用来实现现场的控制逻辑；后者用来调用前者，使之运行。

■ 功能块 (Function Block)

功能块可产生一个或多个数值可执行的逻辑元件。功能块可保留值于下一运算，输入一组数据后，可能得到不同的输出数据。

功能块不能单独运行，必须被程序调用后，才能在程序中运行。

■ 函数 (Function)

函数是产生确切结果的逻辑元件。与功能块不一样，值不能保留到下一次求值中。

当定义函数时，函数必须接受一个数据类型作为返回值（返回数据类型）。函数的计算结果赋给函数本身，及函数的输出变量就是函数名本省。

函数执行时，会对一系列特定的输入产生唯一数据类型的输出结果。相对于功能块而言，函数只有一个输出结果，没有任何内部条件。也就是说，只要给定相同的输入参数，调用函数必定得到相同的运算结果。平时所使用的各种数学运算，例如 $\text{SIN}(X)$ 等就是典型的函数类型。



- 在下面的描述中，程序类 POU 简称为程序；功能块类 POU 简称为功能块；函数类 POU 简称为函数。

5.3.2 添加 POU

AutoThink 软件中按照 POU 组态内容的不同，对其进行了划分，但是用户在添加 POU 时，打开的对话框是相同的；对话框中的可操作内容会根据 POU 的类型有所差异。

一个工程中最多可添加 512 个 POU。



- 工程管理树：右击【程序块】节点，单击【添加程序】。

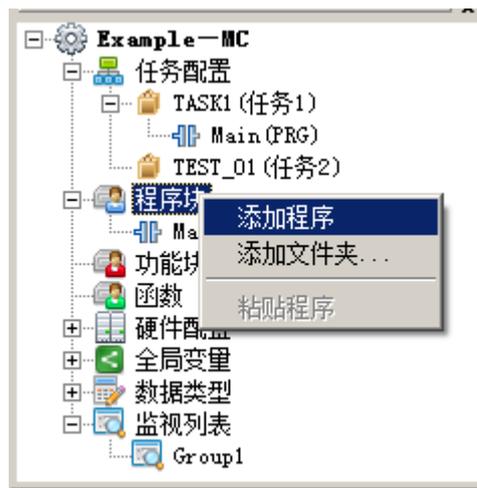


图 5.3-2 选择添加 POU 的命令



图 5.3-3 添加 POU 对话框

- 名称：输入 POU 名称，如果名字不合法，**确定**按钮不可用。

POU 名只能包含字母、数字、下划线“_”，第一个字符必须是字母或者下划线，且遵循如下原则：

- 不能与变量名、变量组名、POU 文件夹名、任务名、工程名、数据类型（自定义或系统缺省的）、关键字、指令库名或功能块名重名。
 - POU 名不可为空。
 - POU 名称长度最多不超过 32 字节，超出范围的部分无法输入。
 - POU 名称不能定义为 Windows 预留的关键字，关键字包括：CON、PRN、AUX、NUL、COM0、COM1、COM2、COM3、COM4、COM5、COM6、COM7、COM8、COM9、LPT0、LPT1、LPT2、LPT3、LPT4、LPT5、LPT6、LPT7、LPT8、LPT9。
 - 纸张类型：选择纸张类型，即打开编程窗口的大小，缺省为 A0，可选择 A3、A4、B5、Ax、A0。
 - 类型：不可修改，根据用户所选的 POU 节点，自动缺省 POU 类型。POU 类型请参见章节 5.3.1 POU 的类型。
 - 语言：选择编程语言，提供 CFC（连续功能图）、LD（梯形图）、ST（结构化文本）、SFC（顺序功能图）。
- 关于 POU 中各类语言的组态方法可参见章节 5.4 编程语言。
- 返回类型：当添加函数 POU 时，该按钮可操作。

在弹出的“类型管理器”对话框中选择函数返回的数据类型。单击**确定**结束类型设置。

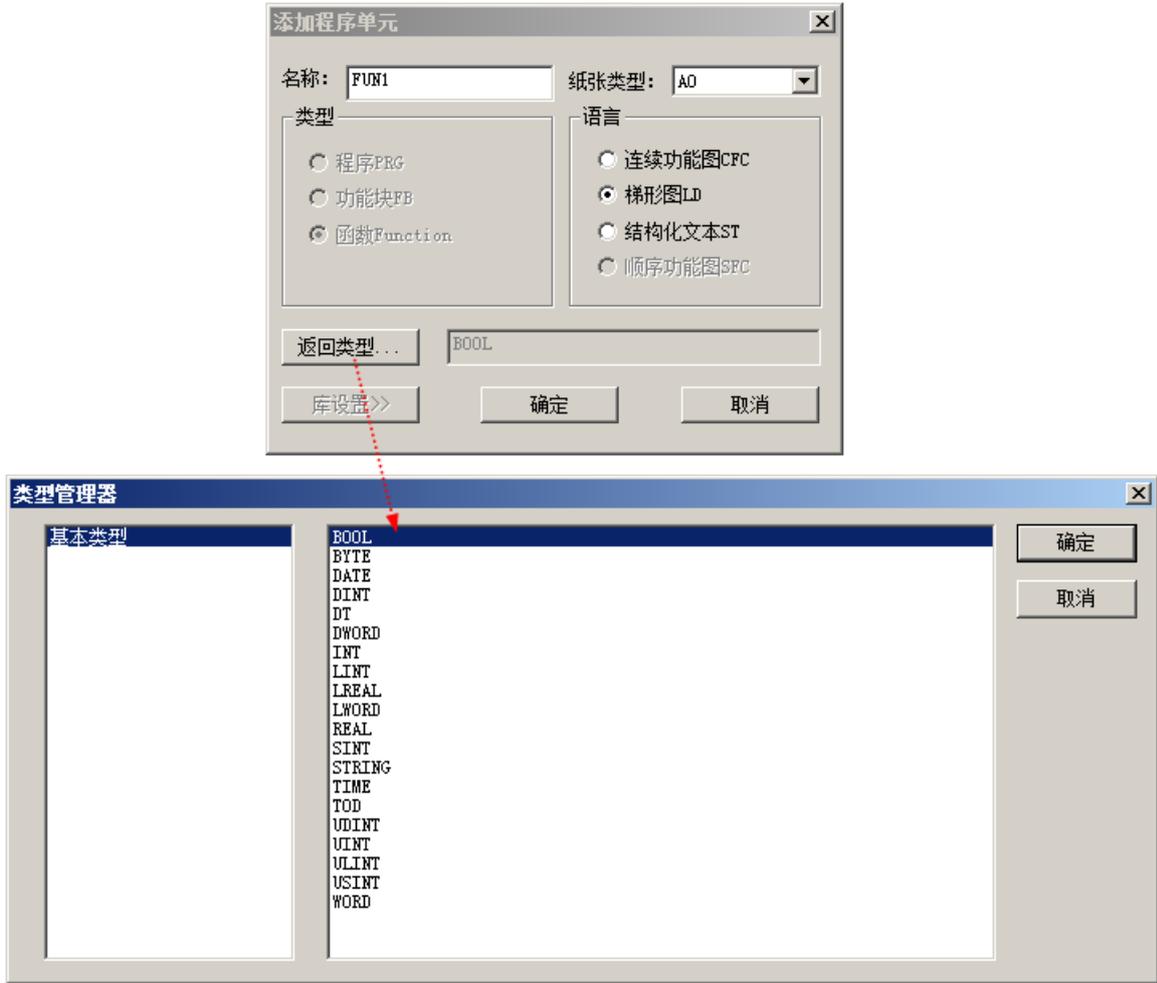


图 5.3-4 添加函数 POU

单击**确定**完成 POU 的添加，同时该 POU 以子节点的形式出现在【工程管理】树中相应节点下。

下面是不同类型 POU 的添加结果，如图 5.3-5、图 5.3-6、图 5.3-7 所示。以 LE 工程为例。

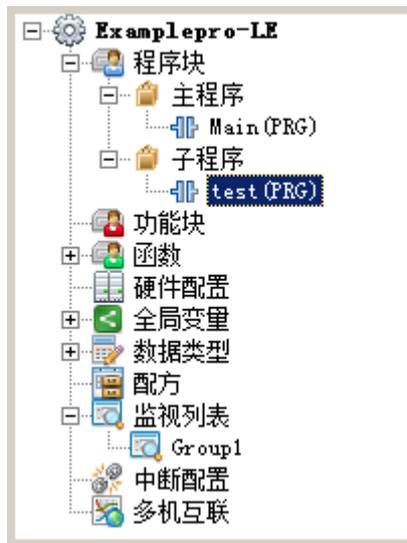


图 5.3-5 添加子程序



图 5.3-6 添加功能块



图 5.3-7 添加函数

5.3.3 POU 的操作

以 MC 工程为例，右击 POU，在它的快捷菜单中可以对其进行一系列的操作，如图 5.3-8 所示。



图 5.3-8 POU 程序的操作

5.3.3.1 删除

右键菜单中选择【删除】命令，将该 POU 删除。



- POU 删除后不能恢复。

5.3.3.2 编辑



- 工程管理树：右击 POU 名，单击【编辑】。

打开该 POU 程序进行编辑。

5.3.3.3 重命名

右键菜单中选择【重命名】命令，弹出“重命名”对话框，如图 5.3-9 所示。



图 5.3-9 POU 重命名对话框

在新名称框中输入新的 POU 名称，命名规则参见 5.3.2 添加 POU。

5.3.3.4 复制-粘贴

选择【复制程序单元】命令，进行子程序的复制粘贴操作。操作如图 5.3-10 所示。



图 5.3-10 复制和粘贴 POU

5.3.3.5 禁止调用/允许调用

该命令用来阻止用户程序参与运算。选择【禁止调用】命令，该 POU 图符变为灰色，该 POU 即使添加到其他程序或任务中也不会被调用，如图 5.3-11 所示。



图 5.3-11 禁止调用

禁止调用后可以再次选中该 POU，通过选择【允许调用】命令，使该 POU 恢复可调用状态，如图 5.3-12 所示。

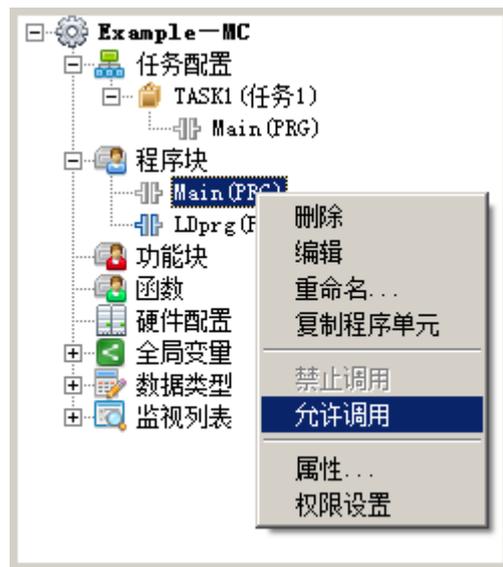


图 5.3-12 允许调用



- 禁止调用的 POU，以及编译成功后未被任务调用的 POU，其图符显示为灰色。
- 设置禁止调用属性的 POU 被调用时，软件编译报错。

5.3.3.6 属性

选择【属性】命令，弹出 POU 属性窗口，如图 5.3-13 所示，除纸张类型设置外，其他设置只能查看，不可修改。



图 5.3-13 POU 属性窗口

5.3.3.7 权限设置

选择【权限设置】命令，弹出权限设置对话框，如图 5.3-14 所示，用户可以对 POU 分别设置读写密码和只读密码。



图 5.3-14 POU 权限设置对话框

其设置方法参见章节 5.2.9 工程权限。



- 用户对 SFC 语言的 POU 设置权限后，该 POU 下添加的所有动作也会被赋予相应权限。

5.3.4 POU 的调用

任何一个 POU 只有被调用才能够开始运算。这个触发在 MC、LK 工程中由任务以时间片周期扫描的方式实现。在 LE 工程中，由主程序以周期扫描的方式实现，也可以中断的方式实现。

5.3.4.1 POU 的调用原则

POU 的调用要遵循以下原则，如图 5.3-15 所示。

- 程序可以调用函数、功能块和其他程序。
- 功能块可以调用函数和其他功能块。
- 函数可以调用功能块或函数。

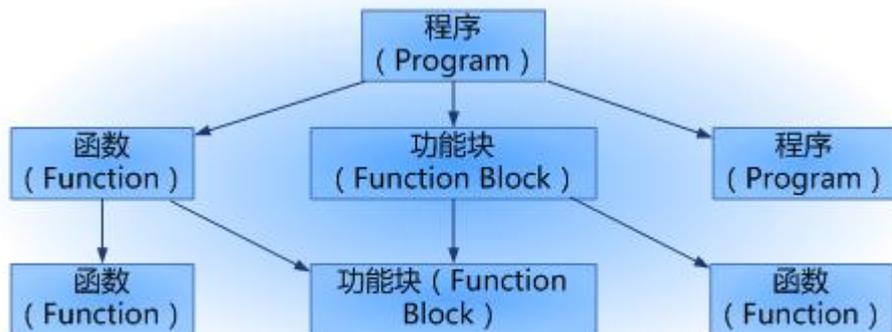


图 5.3-15 POU 的调用

5.3.4.2 POU 的调用方式

POU 调用：一个 POU（主调）调用另一个 POU（被调）（不允许递归调用）。这里的主调 POU 可以是任意一个用户自定义 POU，也可以是缺省的主程序 Main。

缺省情况下，当一个主调 POU 下有多个被调用的 POU 时，按照从上到下的顺序依次进行调用。

中断调用：调用中断程序，使系统对特殊的内部或外部事件作出响应。系统响应中断时自动保存当前子程序的执行状态，执行中断程序；中断处理完成时，又自动恢复至子程序中断前的状态。

5.3.4.3 程序调用程序

从【程序块】节点下拖动被调用的程序到调用的 POU 中，如图 5.3-16 所示。

SFC 语言在动作中调用程序。

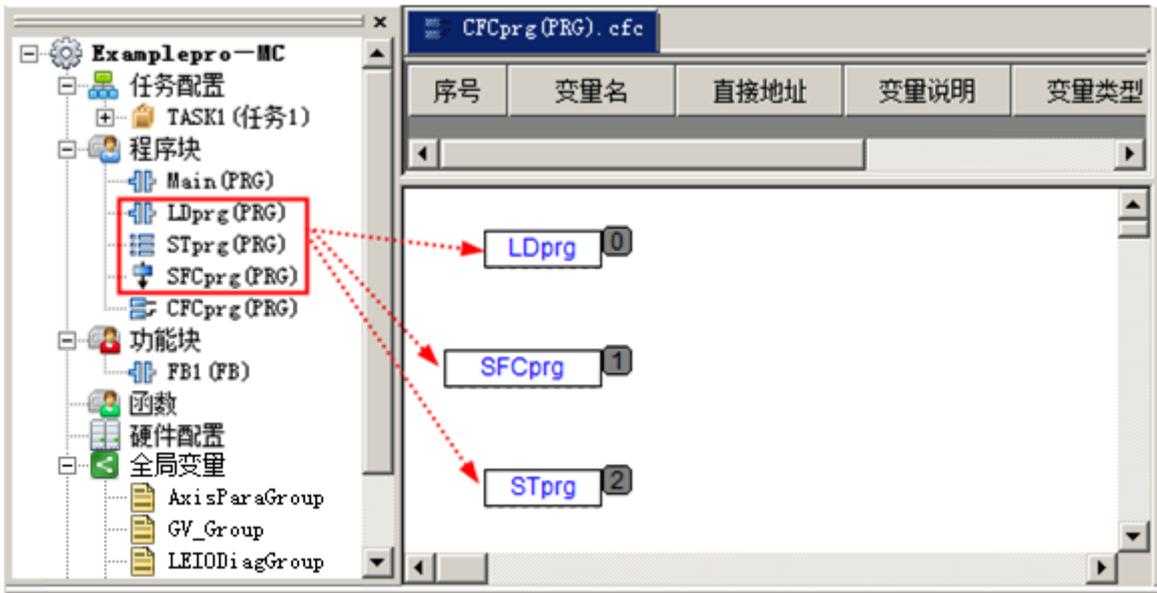


图 5.3-16 MC 工程程序调用程序

LE 工程中，【子程序】节点下的 POU 必须被主程序 Main (PRG) 调用才可以被运算。如图 5.3-17 所示，将【子程序】下的 POU 拖动到主程序 Main (PRG) 节上进行调用。

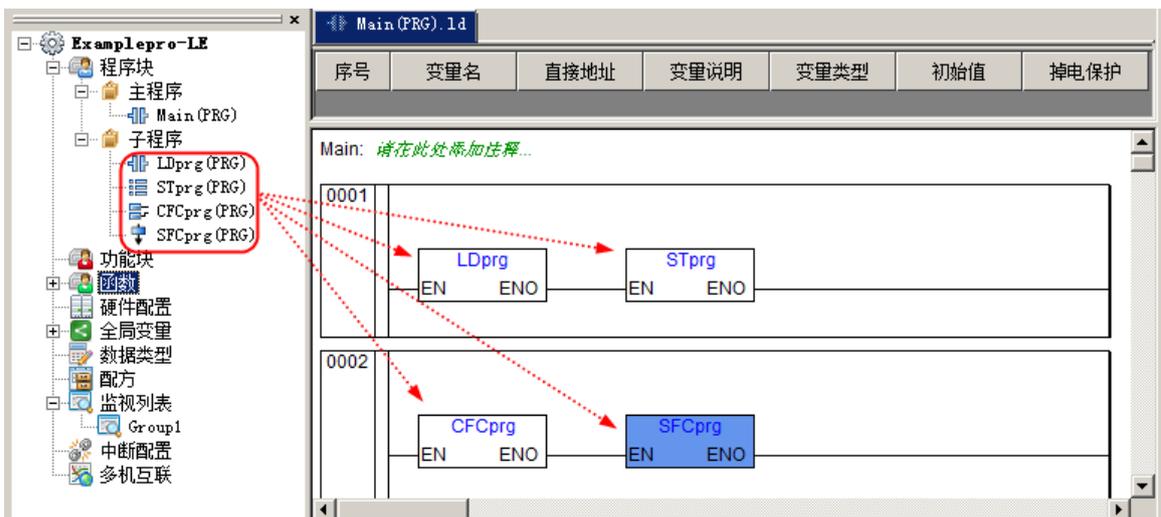


图 5.3-17 LE 工程主程序调用子程序

5.3.4.4 程序调用功能块

程序调用用户自行创建的功能块的方法与调用系统本身提供的标准功能块指令的方法相同。

程序调用功能块时，必须先对功能块进行实例声明。对功能块进行实例声明后，可通过“实例名.参数名”，调用功能块中的参数。

实例针对功能块而言，每个功能块实例就是一个独立的、可完成特定逻辑功能的对象。不同的程序、不同的任务都可以定义和调用功能块的应用实例，每个调用实例都占用独立的内存，保留独立的逻辑状态。

以 MC 工程为例进行介绍。

1. 在 LD、CFC 以及 SFC 的动作中调用功能块方式一致。（以 LD 为例说明）

■ 拖拽方式进行调用

将库管理器中的功能块拖拽到程序区节点上。如图 5.3-18 所示。

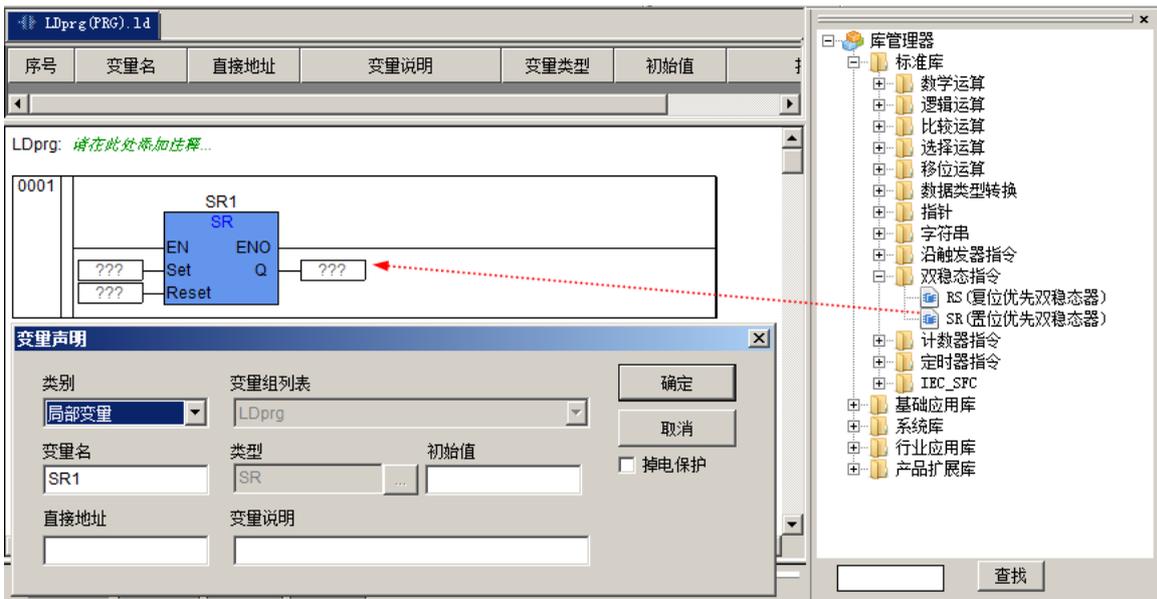


图 5.3-18 拖动方式调用功能块

弹出“变量声明”对话框，在**变量名**中输入变量名，单击**确定**，实例声明完成。

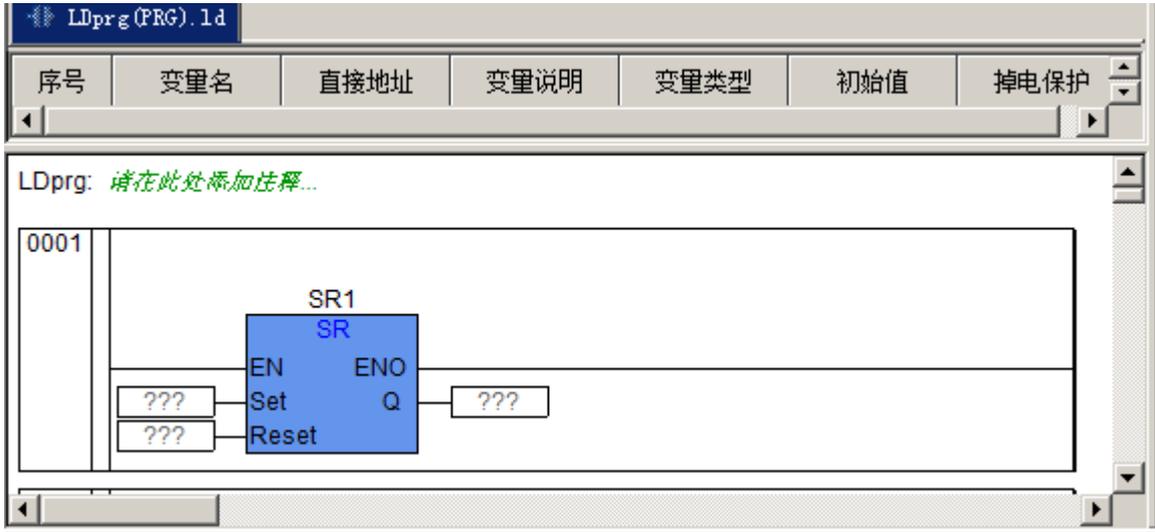


图 5.3-19 在节上添加功能块

也可直接将【功能块】节点下自定义的功能块 FB1 拖到选中的“节”上，过程同上。如图 5.3-20 所示。

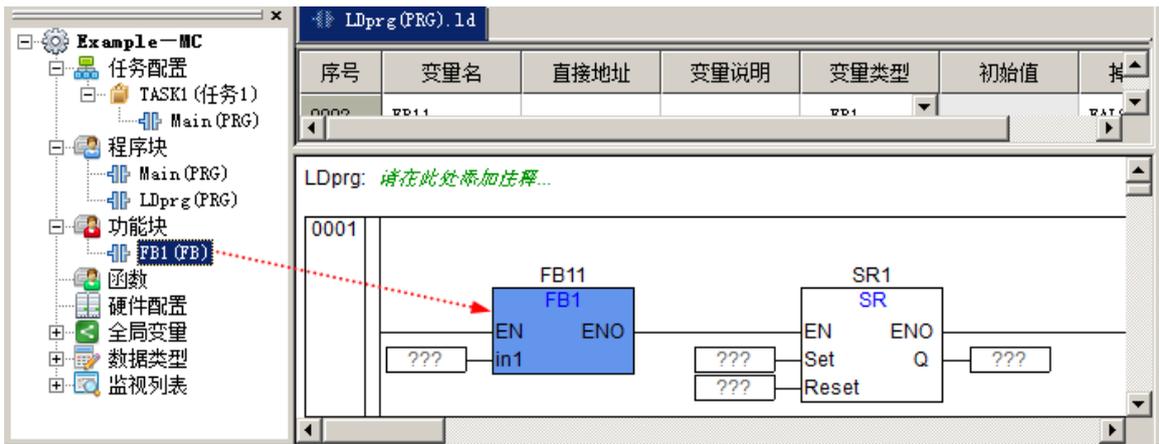


图 5.3-20 使用程序块节点下定义的功能块

■ 添加块元件方式

在程序区插入【块元件】，修改缺省的“AND”为要添加的功能块类型。如图 5.3-21 所示。

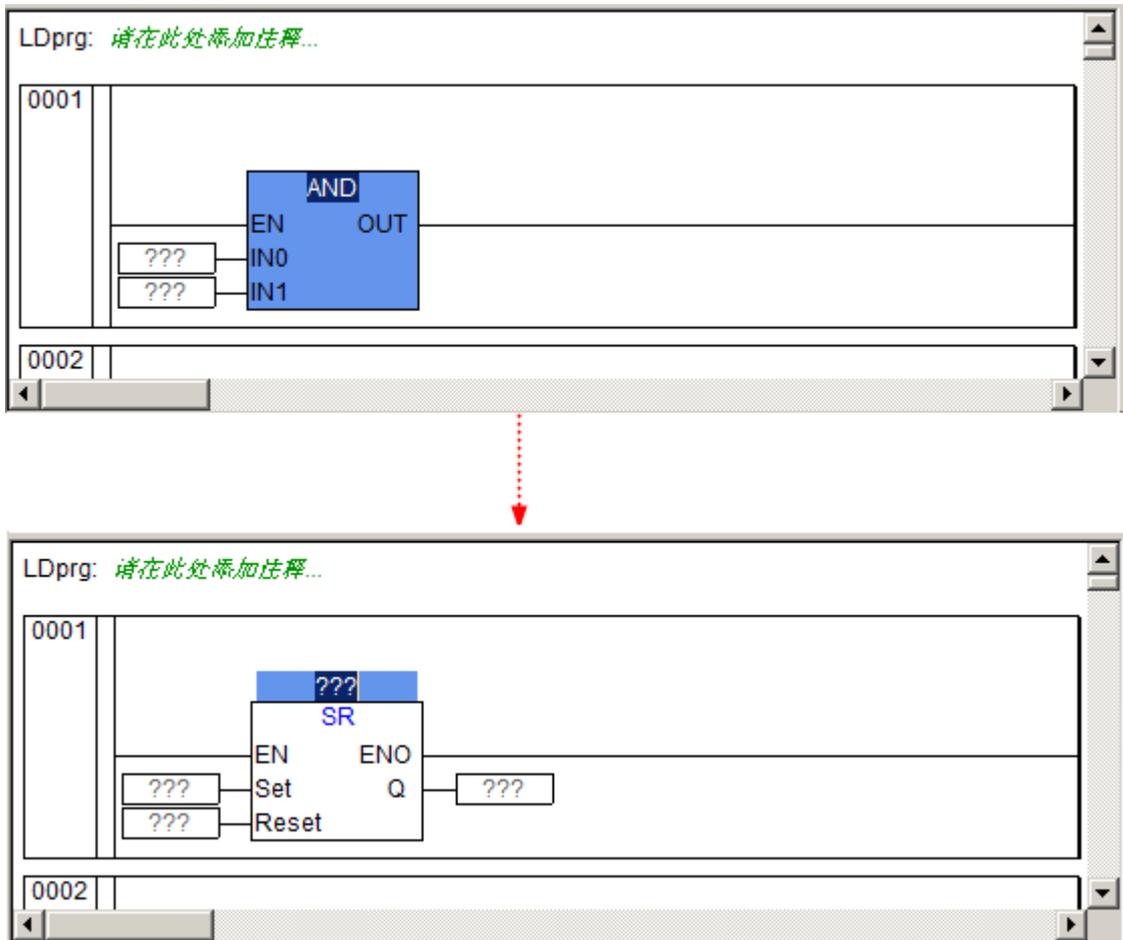


图 5.3-21 添加块元件方式调用

选中文本???, 输入实例名, 弹出“变量声明”对话框, 单击**确定**, 完成实例声明。

2. ST 程序调用功能块

ST 调用功能块有拖拽和输入添加方式两种, 拖拽与 LD 相同, 将库管理器中的功能块拖拽到 ST 编辑器的当前光标位置处。

■ 输入添加方式

在编程区写入功能块的实例名, 按 **Enter** 键, 弹出“变量声明”对话框, 单击**类型**按钮, 在类型管理器中选择要调用的功能块。如图 5.3-22 所示。

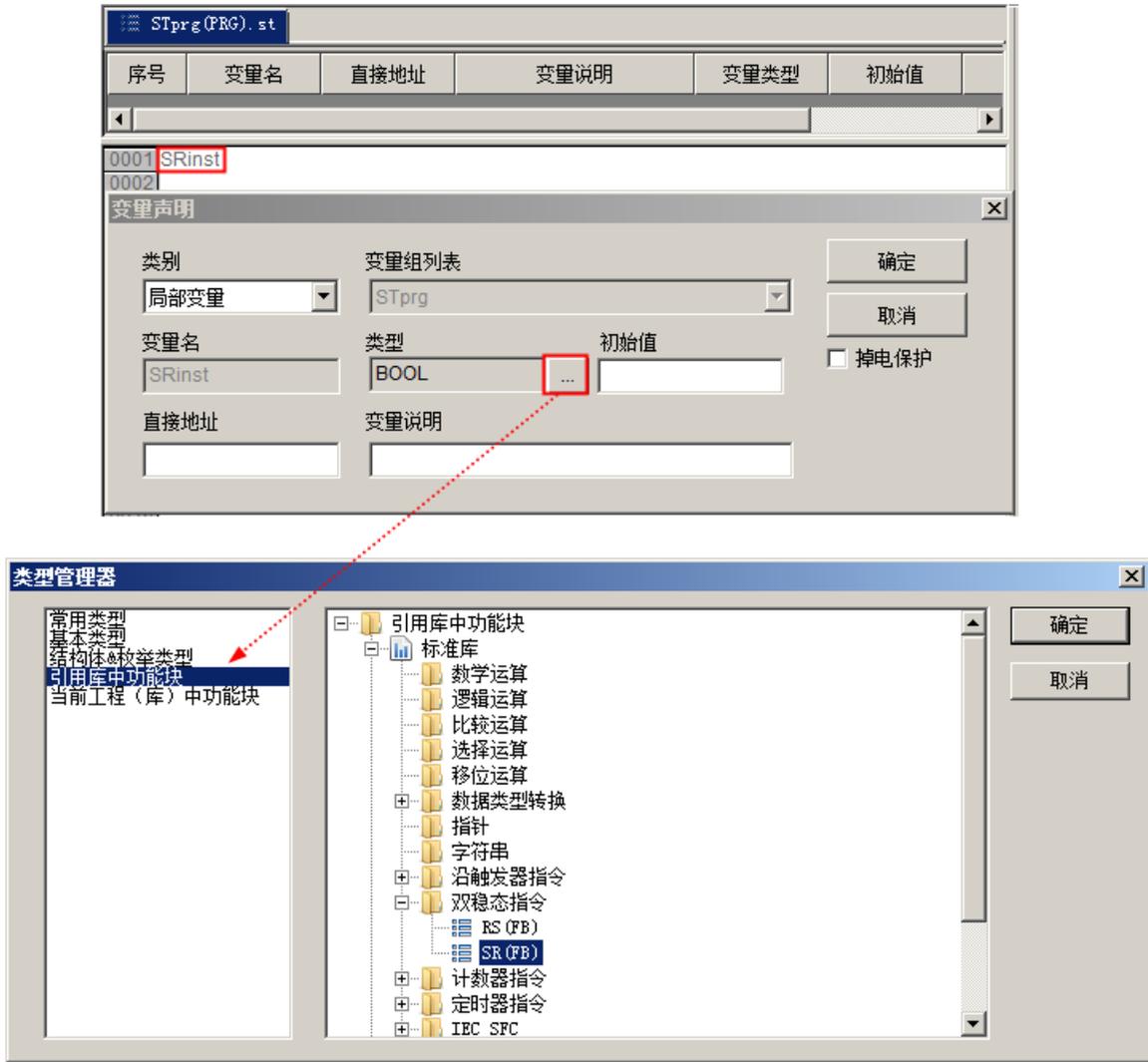


图 5.3-22 输入添加调用方式



- 手动输入添加功能块时需在功能块实例名后加上一对括号，例如“SRinst();”。

5.3.4.5 程序调用函数

程序调用函数的方法与调用功能块一样，只是函数没有实例名。调用方法参见章节 5.3.4.4 程序调用功能块。

5.3.4.6 中断配置

仅 LE 支持此功能。

中断是指在程序的执行过程中，能引起某些程序优先执行的特殊触发条件。

中断配置是指配置中断信号以引起相应的子程序调用的操作，这些不同功用的中断信号均由系统提供，中断在执行过程中按照其缺省的优先级关系顺次执行，从优先级最高的向优先级低的中断命令执行。

在程序运行的过程中，当中断发生时，会停止主程序，进而调用该中断关联的程序。

在 LE 工程管理树中，双击【中断配置】节点，显示【中断配置】编辑窗口，如图 5.3-23 所示。



图 5.3-23 “中断配置”编辑窗口

需要启用某条中断信号时，请勾选对应行。但是，必须为该中断设置中断程序名后，方可选择是否启动该中断。

双击中断程序名字段，弹出“添加中断程序”对话框，如图 5.3-24 所示。



图 5.3-24 “添加中断程序”对话框

在名称框中输入【程序块】中的 POU 名称，例如 TEST1 程序，语言自动默认该 POU 的语言类型，单击确定关闭对话框。这时，勾选中断名称，如图 5.3-25 所示。中断触发后，执行 TEST1 程序。

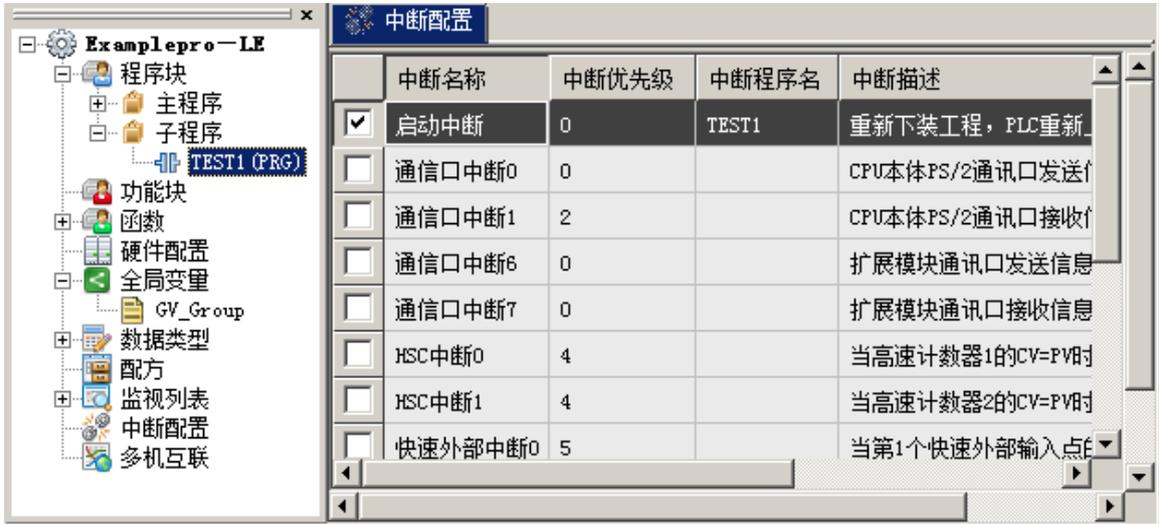


图 5.3-25 启动“中断”

5.3.4.7 查看调用树



- 菜单栏：单击【工程】—【查看调用树】。

选择该命令，显示“调用树”窗口，如图 5.3-26 所示，显示当前工程中的调用树状结构图，即任务与 POU、POU 与库指令之间的调用关系，可以层层展开或关闭。工程编译成功后，才能查看调用树。

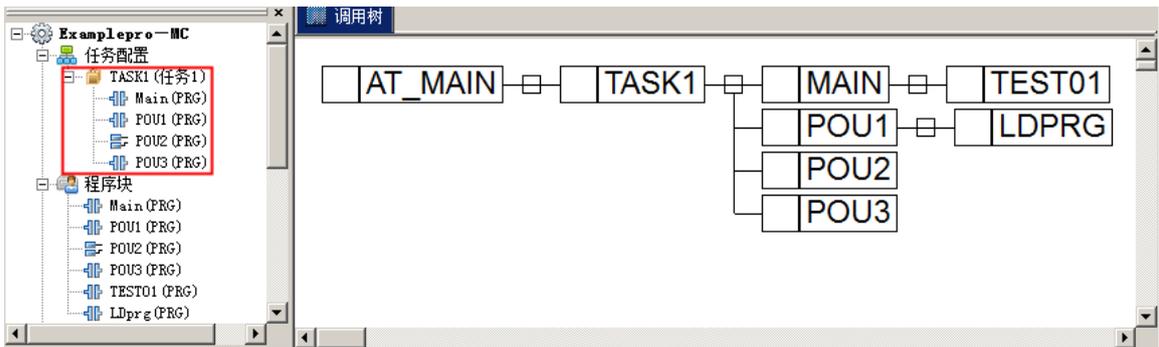


图 5.3-26 “调用树”窗口

5.3.5 配方函数

仅 LE 特有功能。

配方函数是配方新建完成后自动添加的函数。

配方的作用是用户按照预先定义的结构来组织若干组数据，然后通过直接对内存块操作的方式来读写这批数据。一个配置对应一个配方模板，相当于一个结构体类型实例。一个配方模板中可包含多个域，

相当于一个结构体中可定义多个成员变量。一个配置包含多个配方（至少一个配方，最多可包含 32 个配方），相当于结构体实例中成员变量的多种赋值方案。配置、配方、域的关系，如图 5.3-27 所示。

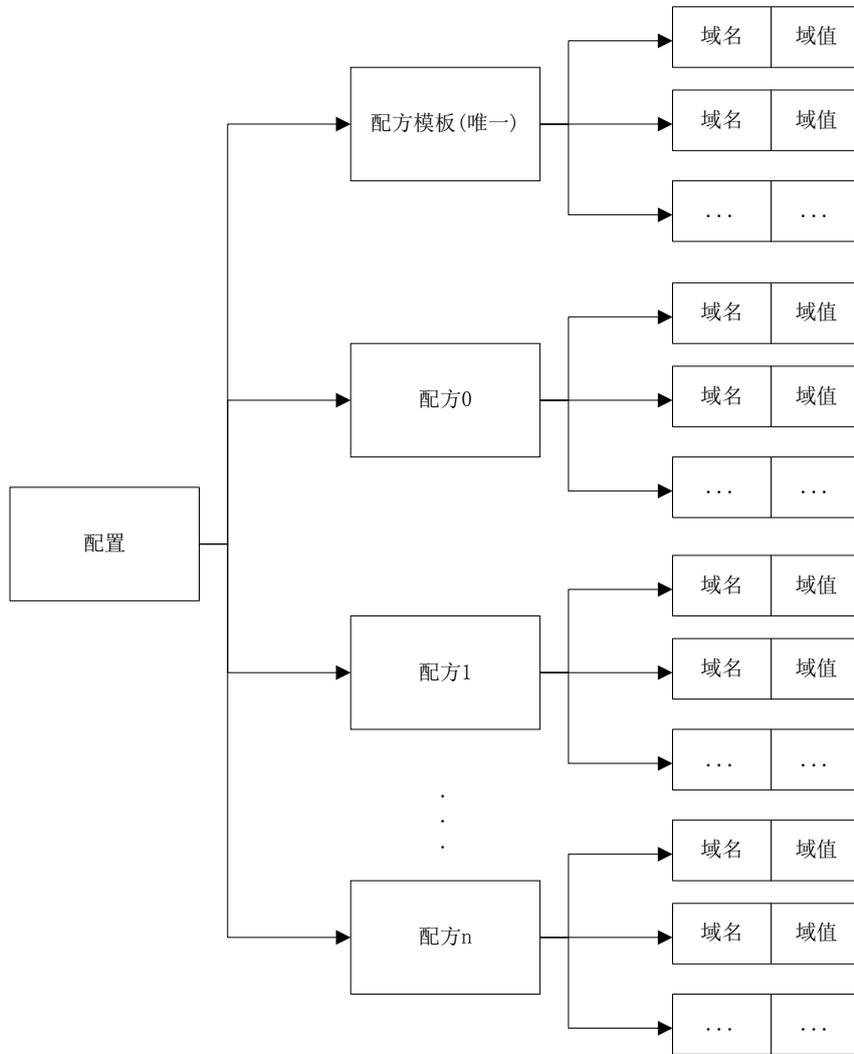


图 5.3-27 配方、配置、域的关系图 ($n \leq 31$)

5.3.5.1 创建配方数据

第1步 创建一个配置



- 工程管理树：右击【配方】节点，单击【新建配置】；
- 菜单栏：单击【工具】—【配方数据使用向导】。

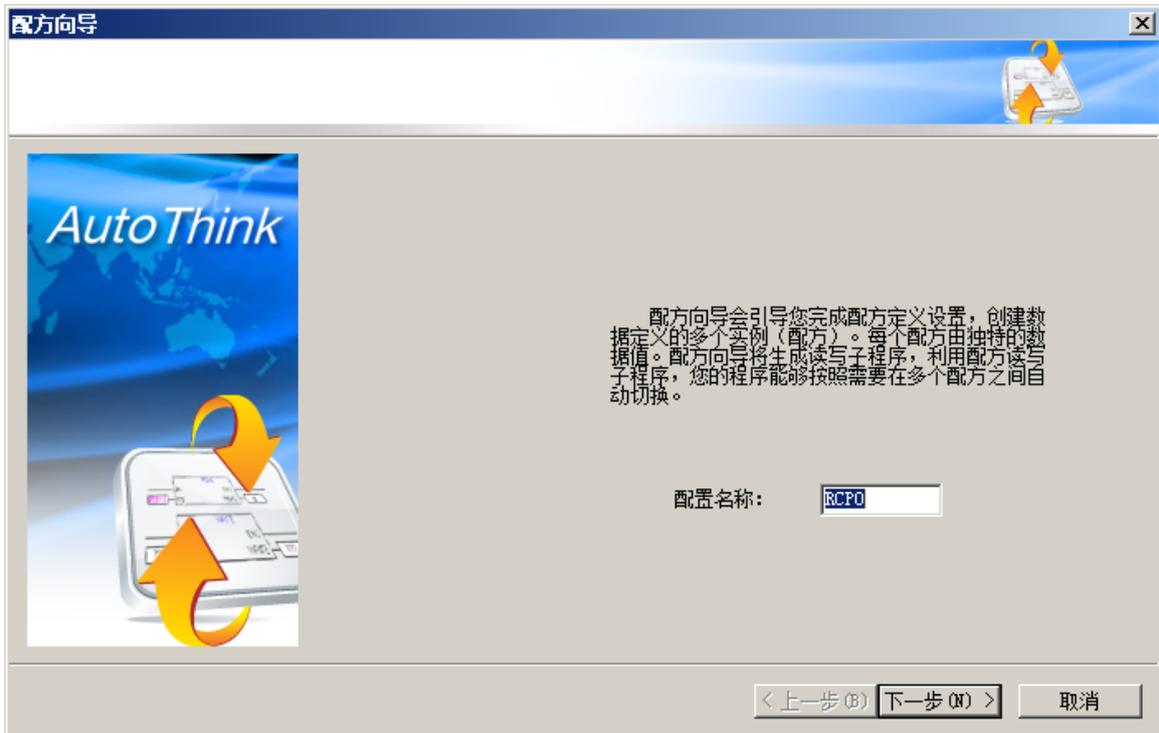


图 5.3-28 配方向导

- 配置名称可以是字母、数字、下划线，不能以数字开头，不能以“AT_”开头，不能是关键字，且遵循如下原则：关于重名：不能与全局变量名、配方名、枚举常量名、数据类型（自定义或系统缺省的）、POU 名、通道名称重名。
- 配置名称不可为空。
- 配置名称长度最多不超过 25 字节，超出后无法继续输入。

在此对话框中输入符合系统要求的**配置名称**，然后单击**下一步(N)**。

第2步 定义配方模板

为该配置 **RCF0** 添加配方模板内容，即输入域名、数据类型、默认值以及注释信息，如图 5.3-29 所示。定义配方模板后，单击**下一步(N)**，如图 5.3-30 所示。



图 5.3-29 定义配方模板



图 5.3-30 配方模板示例

域名：一个配方模板中至少包含一个域名。域名可以是字母、数字、下划线，不能以数字开头，不能以“AT_”开头，不能是关键字，长度最多不超过 32 字节。域名允许在不同配置中重名，但不允许在同一个配置中重名。

数据类型：采用下拉列表框的形式进行选择，主要包括 BYTE、BOOL、WORD、INT、REAL、DINT、DWORD 等。当域名输入完，如果没有选择数据类型，数据类型自动选择 BYTE。

在域名和数据类型输入完后，系统自动根据不同的数据类型生成默认值。用户可根据需要进行修改，如果修改值不符合相应的数据类型，则系统自动恢复该类型的默认值。

注释长度最多不超过 64 字节，该项可以为空。

第3步 为配置分配存储区域

为配置分配存储区域，可直接使用系统自动计算生成的建议地址。如需修改，则输入的地址不能为空，必须以“%MB”开头，并且起始地址必须是 4 的倍数。然后单击**下一步(N)**，如图 5.3-31 所示。

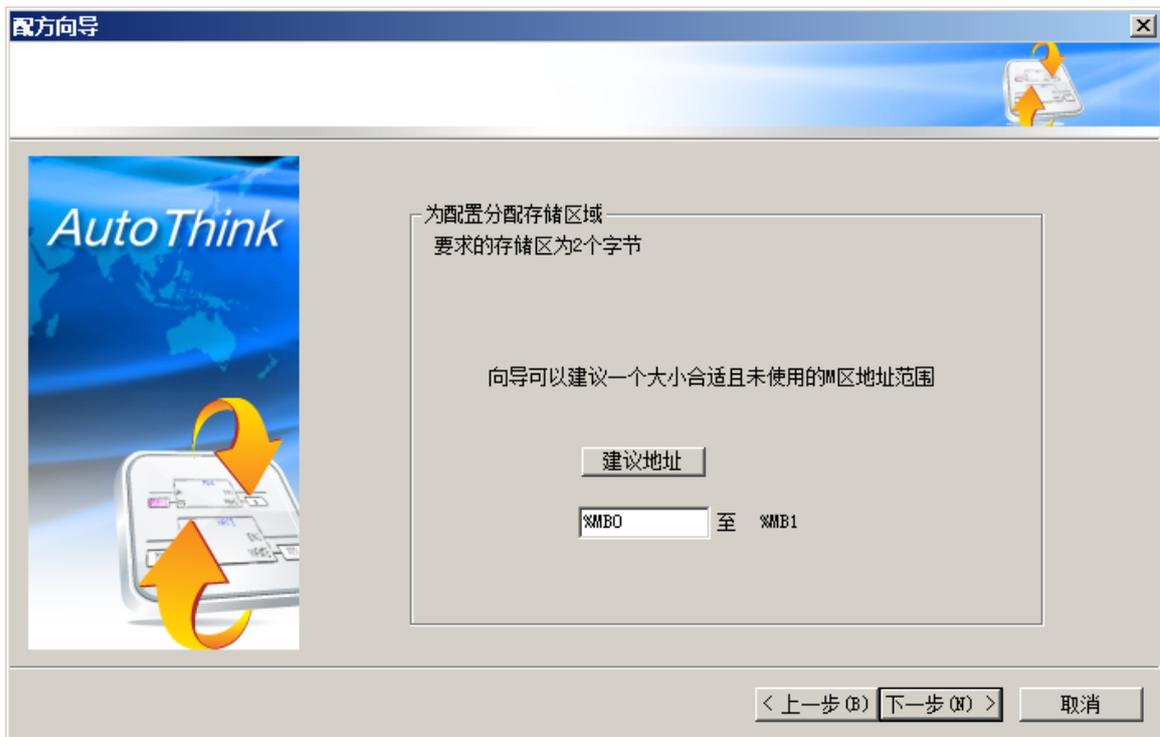


图 5.3-31 为配置分配存储区域

第4步 为配方模板分配地址空间

根据前面的输入，系统自动为每个域名分配地址。用户可以查看，但不可对该区域进行编辑和修改，然后单击**下一步(N)**，如图 5.3-32 所示。



图 5.3-32 为每个域名分配存储区

上述步骤完成一个配方模板的定义，下述步骤为各个配方实例的定义。不同的配方实例可以定义不同的域值。

第5步 创建和编辑配方

如图 5.3-33 所示，此步中可以修改已有配方的域值和配方名称、增加新配方、删除配方，但配方个数不得少于一个。

每个配方名所在列下的数值为该配方的域对应的值。如图 5.3-34 所示，a 域在配方 0 中的值为 0，在配方 1 中的值为 1。单击**还原默认值**，可将选中配方对应的域值设为域名的默认值。完成此步后，单击**下一步(N)**。



图 5.3-33 创建和编辑配方



图 5.3-34 创建和编辑配方示例

第6步 为配方分配地址空间

系统计算所有新增加的配方所占的存储区字节，自动生成每个新增加配方的存储区范围，然后单击下一步(N)，如图 5.3-35 所示。



图 5.3-35 配方实例地址分配

第7步 创建配方读写子程序

如图 5.3-36 所示，单击完成，系统自动生成两个配方读写子程序，如果配置名称为 **RCP0**，则两个子程序名称为 **RCP0_READ** 和 **RCP0_WRITE**。配方的所有设置完成后，系统会自动在工程管理树中的【配方】和【配方函数】下添加相应的内容。

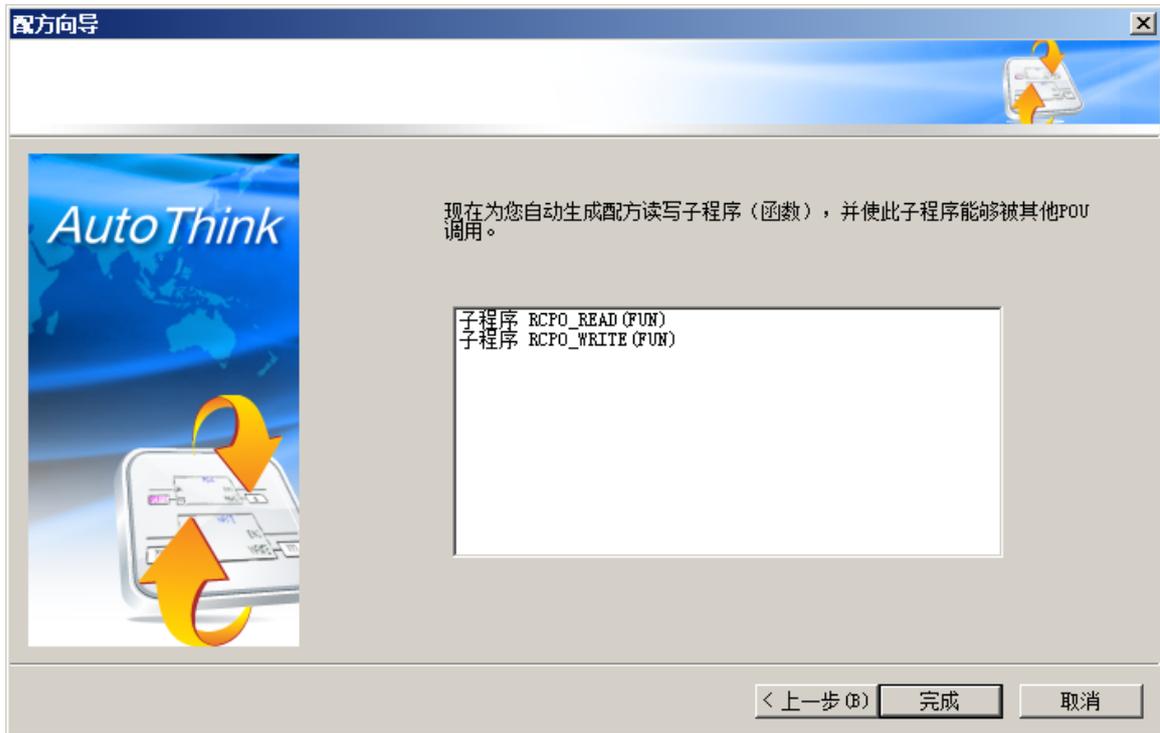


图 5.3-36 创建配方读写子程序

配方设置完成后，在配方的树状图中，可以查看相应的配方函数，配置或查看子配置项（配方、配方地址和配方定义），如图 5.3-37 所示。双击或右击子配置项选择【打开】，可以进入相应的配方向导对话框进行查看和设置。右击配置名（如图中的【RCP0】），选择【删除】，则删除该配方的相关内容。

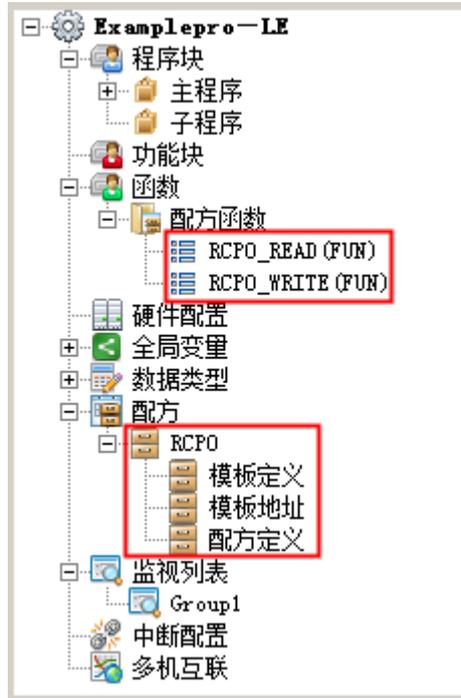


图 5.3-37 “配方”树状图

5.3.5.2 配方函数的使用

通过调用配方的读写子程序函数，就能使用配方向导功能实现内存的拷贝。

■ 索引号

每个实例对应一个索引号，通过索引号实现对不同配方的引用。



- 工程管理树：右击配方【RCPO】节点下的【配方定义】，单击【打开】。



图 5.3-38 打开配方向导

在“配方向导”对话框中，索引号从 0 开始，从左到右依次增加，如图 5.3-39 所示。

		0	1	2	3
域名	数据类型	配方3	配方2	配方1	配方0
a	BYTE	0	0	0	1
b	BYTE	0	0	2	0

图 5.3-39 配方索引示意

用户在读写函数输入端的变量编辑框中，只需要输入配方名（如“配方 0”）就能自动获得其索引号（如“配方 0: 1”）。也可以直接输入索引号，进行配方数据的读写。

如图 5.3-40 所示，当网路 0001 导通时，第 0 组配方实例(配方 1)对应的内存数据(%MB8 到%MB9 中的数据)就会读取到配方模板内存(%MB0 到%MB1)。运行后的结果，如图 5.3-41 所示。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	p1	%MB0		BYTE	0	FALSE
0002	p2	%MB1		BYTE	2	FALSE
0003	p3	%MB8		BYTE	1	FALSE
0004	p4	%MB9		BYTE	3	FALSE
0005	b			BOOL	FALSE	FALSE

PLC_PRG: 请在此处添加注释...

图 5.3-40 配方读子程序（运行前）

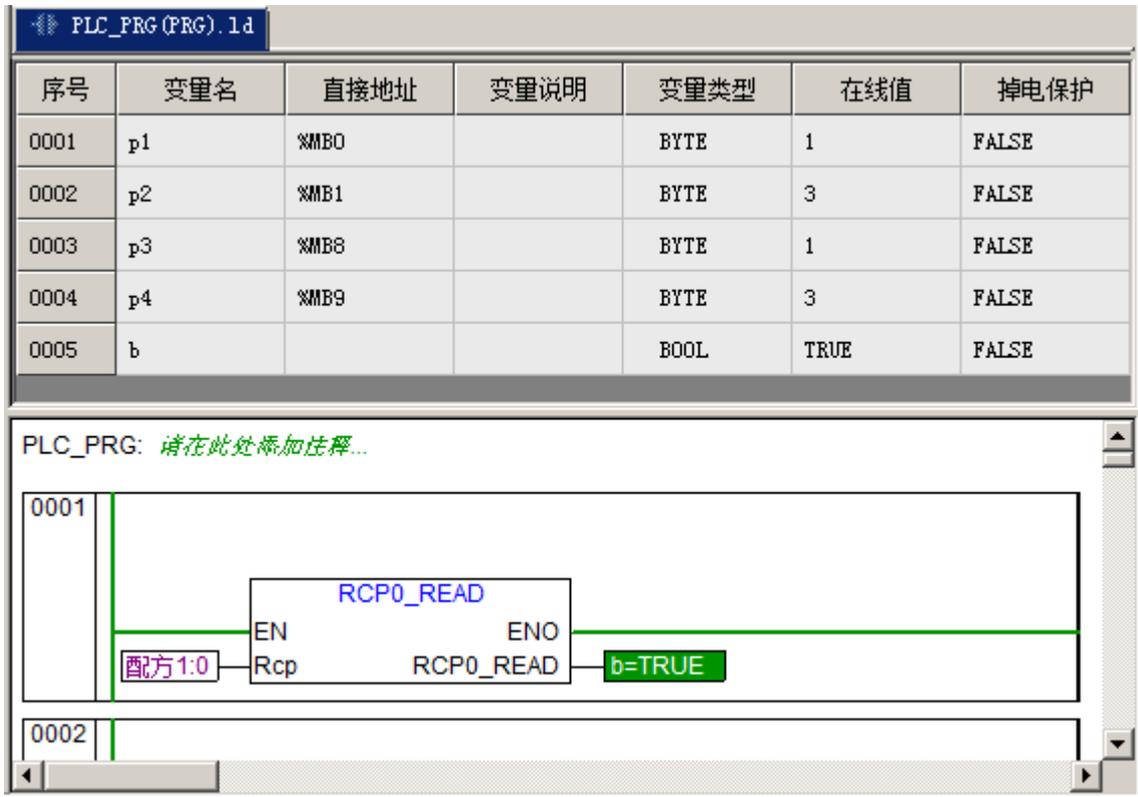


图 5.3-41 配方读子程序（运行后）

如图 5.3-42 所示，当网络 0001 导通时，配方模板内存数据（%MB0 到%MB1 中的数据）就会写入到第 0 组配方（配方 1）实例对应的内存（%MB8 到%MB9）。运行后的结果，如图 5.3-43 所示。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	p1	%MB0		BYTE	0	FALSE
0002	p2	%MB1		BYTE	2	FALSE
0003	p3	%MB8		BYTE	1	FALSE
0004	p4	%MB9		BYTE	3	FALSE
0005	b			BOOL	FALSE	FALSE

PLC_PRG: 请在此处添加注释...

0001

配方1:0

Rcp

RCP0_WRITE

EN

ENO

b=FALSE

0002

图 5.3-42 配方写子程序（运行前）

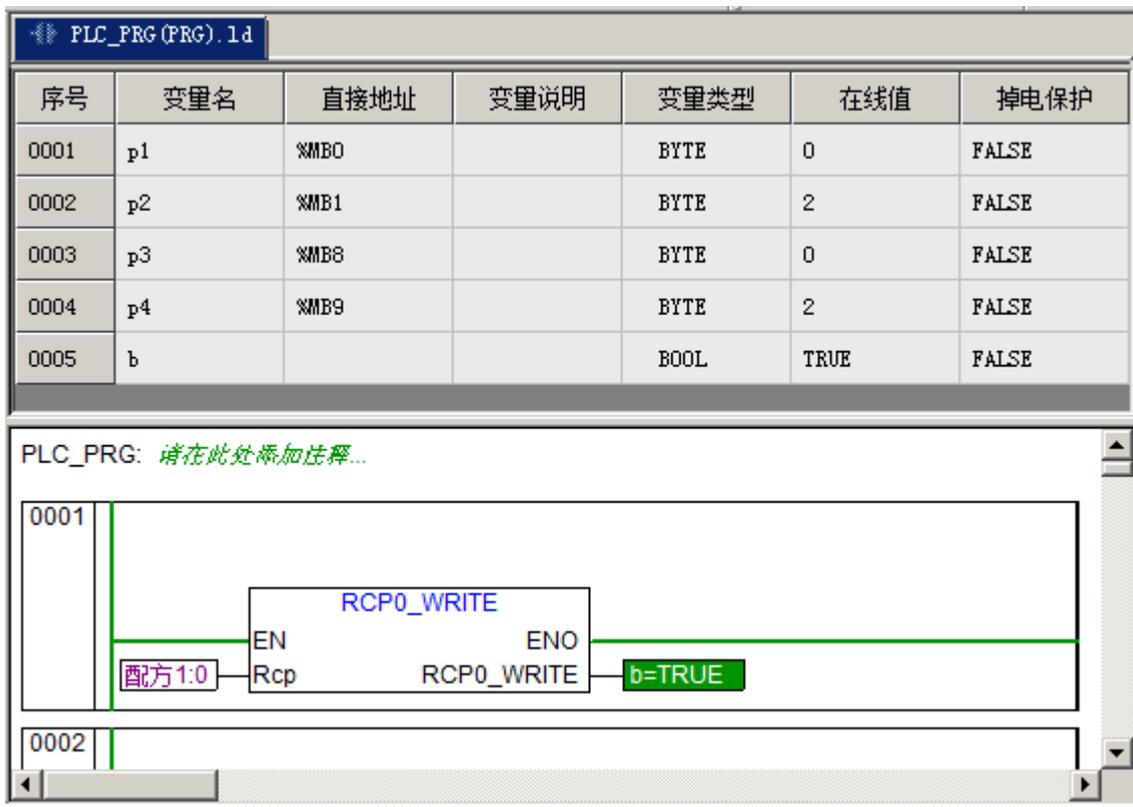


图 5.3-43 配方写子程序（运行后）



- ST 语言中调用配方函数只支持输入配方索引，不支持输入配方实例名。

5.3.6 文件夹

用户可以通过分层的“文件夹”对添加的 POU 进行分类管理，方便查找与操作。

5.3.6.1 添加文件夹

- 选中 POU 节点添加



- 工程管理树：右击 POU 节点，单击【添加文件夹】。

以 LE 工程中的【功能块】节点为例，如图 5.3-44 所示。

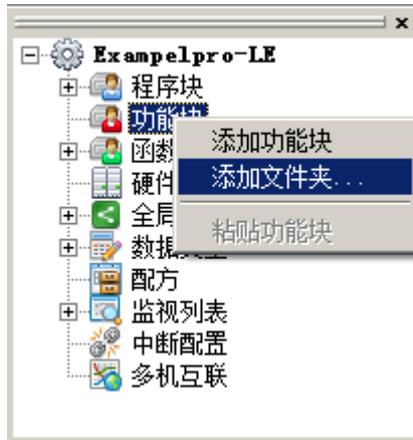


图 5.3-44 “功能块”节点的右键菜单

2. 在已存在文件夹下添加



图 5.3-45 已存在文件夹的右键菜单

弹出“添加文件夹”对话框如图 5.3-46 所示。



图 5.3-46 文件夹命名

输入文件夹名称，文件夹名只能包含字母、数字、中文字符、下划线“_”，长度不超过 32 字节。单击**确定**按钮完成文件夹的添加。

5.3.6.2 删除文件夹



- 工程管理树：右击 POU 节点下的【文件夹】，单击【删除】。

会弹出系统提示，以确认删除操作。删除后，不可恢复，请操作时引起注意。

5.3.6.3 文件夹重命名



- 工程管理树：右击 POU 节点下的文件夹名，单击【重命名】。

对已建的文件夹可以进行重命名操作，在需要更名的文件夹的右键菜单中，选择相应命令。

5.3.7 任务组态

5.3.7.1 添加任务

MC、LK 工程管理树的【任务配置】节点下，缺省有 TASK1（任务 1）节点，该任务节点缺省调用【程序块】节点下已配置的程序 Main（PRG）。在 MC 工程的【任务配置】节点下可以创建多个任务，通过设置每个任务的**时间片**来调度任务被执行的时间。所有的 POU 只有被任务调用，才能执行。



- MC 工程最多支持 8 个任务。
- LK 仅支持单任务。

以 MC 工程为例进行说明。



- 工程管理树：右击【任务配置】节点，单击【添加任务】。



图 5.3-47 添加任务

弹出“新增任务”对话框，如图 5.3-48 所示。勾选启动高级任务配置，进行设置。

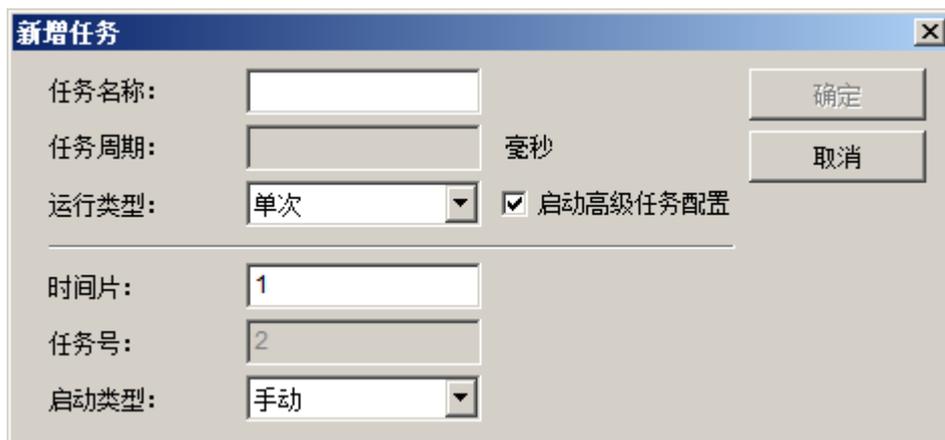


图 5.3-48 新增任务对话框

- 任务名称：名字由数字、字母、下划线组成，不能以数字开头。长度不超过 32 个字符。
- 任务周期：周期性执行任务的时间间隔。当运行类型选择为周期时，可以进行设置。默认为 50 毫秒，设置范围为 1~100,000 毫秒。LK 工程的任务周期设置为 50~100000 之间的整数，且能被 10 整除。
- 运行类型：可选项为单次和周期。选择单次，则该任务只执行一次。选择周期，任务根据设置的时间片和任务周期循环执行。LK 只支持周期类型。
- 时间片：是由系统划分的最小执行时间单元。用户可以设置时间片个数，默认为 1，可设置为 1~50 之间的整数。每个时间片默认为 62.5 μ s。
- 任务号：任务 ID，任务按照任务号顺序执行。不可编辑，新建任务时自动分配。
- 启动类型：默认为手动，可以选择手动和自动。选择手动，工程下装后进入监视状态，任务是停止状态，需要手动单击运行按钮，任务才可以运行。选择为自动，工程下装后进入监视状态，任务自动运行。

添加一个任务名称为 TEST_01，时间片设置为 3，启动方式为自动，以 10 ms 为周期循环运行的任务，如图 5.3-49 所示。



图 5.3-49 新增任务节点



- 对 V3.1.4B3 之前版本 AutoThink 创建的 MC 工程，在升级后可正常使用，原工程的主程序生成一个新的任务节点，名称为主程序名+序号，在任务节点下挂接原主程序节点。同时原主程序和子程序都显示在程序块。

5.3.7.2 添加程序调用

(1) 通过任务右键菜单调用

选中任务节点，在右键菜单中选择【添加程序调用】命令，如图 5.3-50 所示。



图 5.3-50 任务节点右键菜单

弹出“添加程序调用”对话框，显示【程序块】节点下所有的 POU，如图 5.3-51 所示。勾选需要调用的 POU，单击**确定**，完成对已勾选 POU 的调用。

也可通过**全选**进行快速全选。单击**反选**，取消对 POU 的勾选。单击**取消**，取消对勾选 POU 的调用，并关闭“添加程序调用”对话框。



图 5.3-51 选择调用的 POU

任务节点下生成调用的 POU，如图 5.3-52 所示。

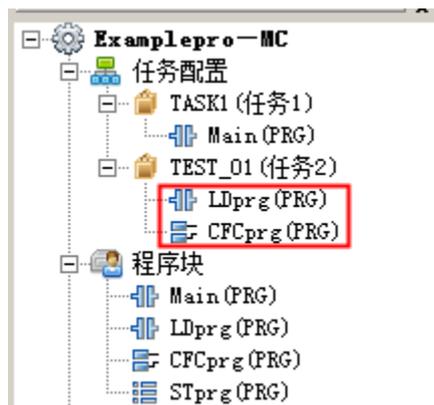


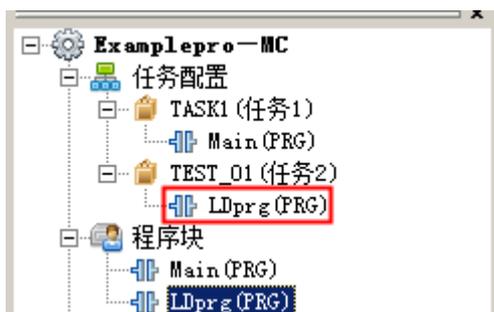
图 5.3-52 调用 POU 完成

(2) 通过拖拽方式调用

选中【程序块】节点下的 POU LDprg，拖拽到任务节点 TEST_01 下进行调用，在任务节点下生成一个同名的 POU，如图 5.3-53 所示。



(a)



(b)

图 5.3-53 拖动方式调用 POU

任务中调用的 POU 按照从上到下的顺序被执行。

被调用的 POU 可通过拖拽来调整调用顺序。

■ 重命名被调用的 POU 名

修改【程序块】节点下 POU 的名字，任务节点下的同名 POU 同步被更新。



- 工程管理树：右击【程序块】节点下被调用的 POU，单击【重命名】。



图 5.3-54 被调用 POU 的重命名操作

■ 删除被调用的 POU



- 工程管理树：右击任务节点下被调用的 POU，单击【删除】。



(a)



(b)

图 5.3-55 删除被调用的 POU

单击**确定**，删除任务节点下被调用的 POU 节点。

如果删除【程序块】节点下被调用的 POU，则任务节点下的同名 POU 同时被删除。

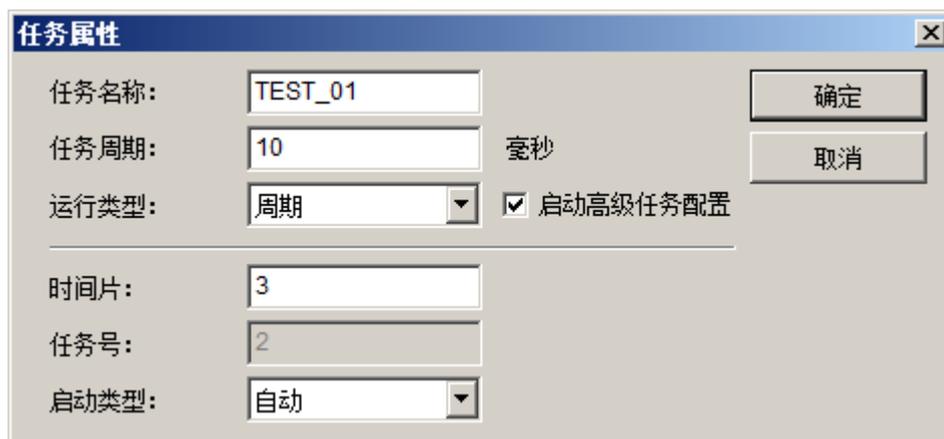
5.3.7.3 修改任务属性



- 工程管理树：右击任务节点 TEST_01（任务 2），单击【属性】。



(a)



(b)

图 5.3-56 任务属性

修改任务属性时，可参考各项的相关信息 [5.3.7.1 添加任务](#)。

5.3.7.4 删除任务



- 工程管理树：右击任务节点 TEST_01，单击【删除】。



图 5.3-57 删除任务提示框

单击**确定**，将删除任务节点 TEST_01 及其下的 POU。

5.3.7.5 任务的运行

编译下装工程后，单击【在线】—【监视】命令，进入在线。TEST_01 任务启动类型组态的是自动，则进入在线后任务自动运行。任务状态实时显示在任务 POU 上，如图 5.3-58 所示。



图 5.3-58 在线运行

右击任务节点，在弹出的右键菜单中显示任务的相关命令。如图 5.3-59 所示。

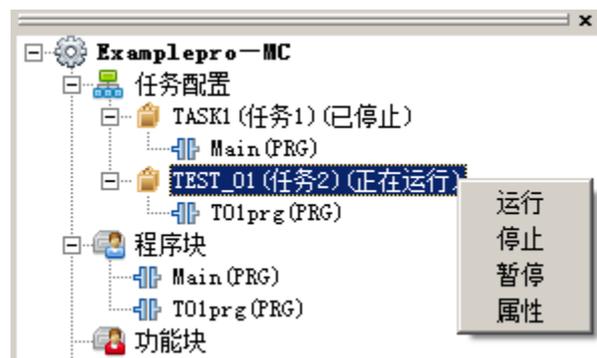


图 5.3-59 在线时任务的右键菜单

如图 5.3-60 所示为任务在线操作按钮，从左到右依次为运行、暂停、停止，对所有任务有效。

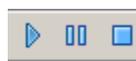


图 5.3-60 工具栏按钮

单击【运行】、【停止】、【暂停】命令，使任务进入相应的运行状态。命令与任务状态的关系如图 5.3-61 图所示。

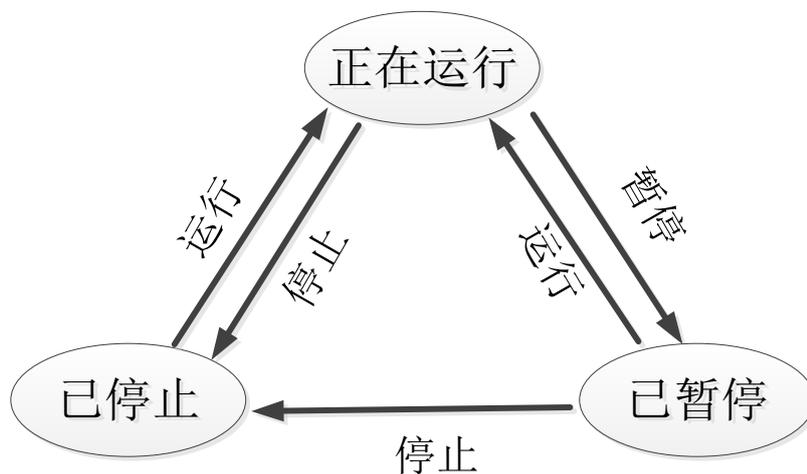


图 5.3-61 命令与任务状态关系

单击【属性】命令可查看任务的属性。

5.3.7.6 程序结构

任务的执行应该遵循以下原则：

- 一旦任务的条件被满足，该任务就执行；
- 如果多个任务的条件都有效，那么将按照任务号从小到大的顺序执行任务；
- 程序将按照它们在任务编辑器中的顺序（上下）被调用。

任务与程序的调用关系及执行顺序如图 5.3-62 所示。

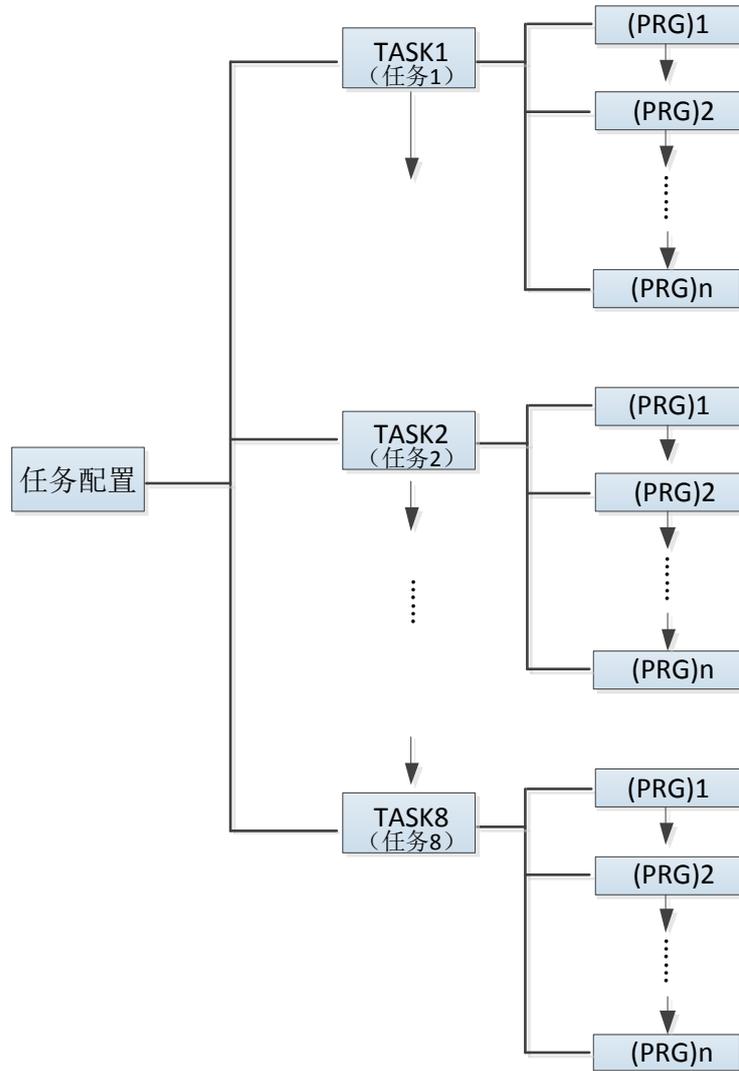


图 5.3-62 任务与程序的调用关系示意图

例如，【任务配置】节点下创建有 6 个任务，如图 5.3-63 所示。任务设置的运行类型均为周期型，启动类型为自动，进入在线执行状态时，任务按照任务号 1~6 依次被执行。当任务 5 在当前周期被执行时，PRG 程序从 Pass_Steel(PRG)开始依次向下顺序执行。如任务 5 执行完毕则立即切换到任务 6；如果任务 5 下的 PRG 没有被执行完，下个周期，任务 5 从上次中断位置继续依次向下执行。每个周期任务严格按照各自周期时间，循环执行。

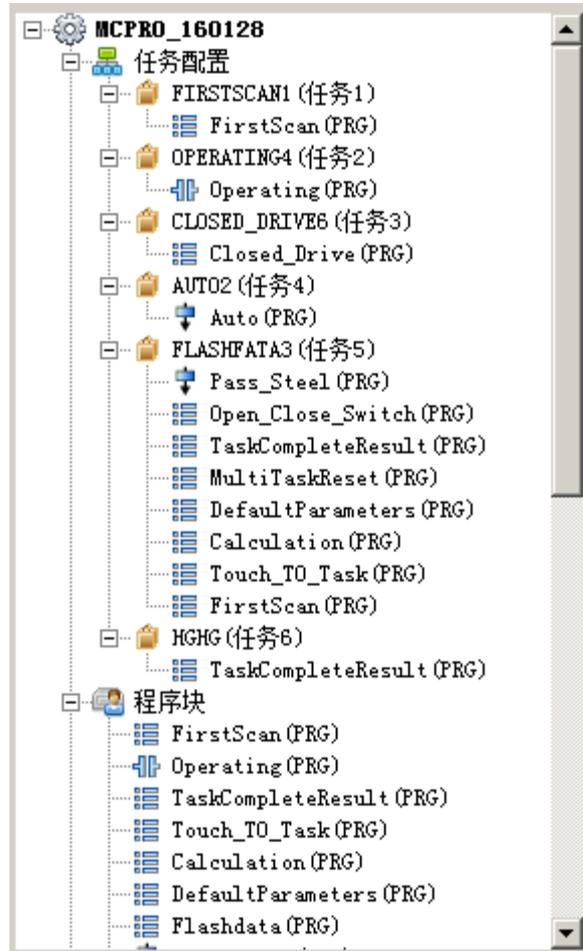


图 5.3-63 任务调用程序

5.4 编程语言

POU 可以用 4 种语言编写：LD、ST、CFC、SFC，下面具体讲述各种语言的编程窗口的使用。

5.4.1 LD 编辑器

LD 是梯形图（Ladder Diagram）的简称，是一种图形化的编程语言。

LD 主要由触点、线圈、功能块和连接线等编程元素组成，通过水平线和垂直线连接成平面网状图，如图 5.4-1 所示。

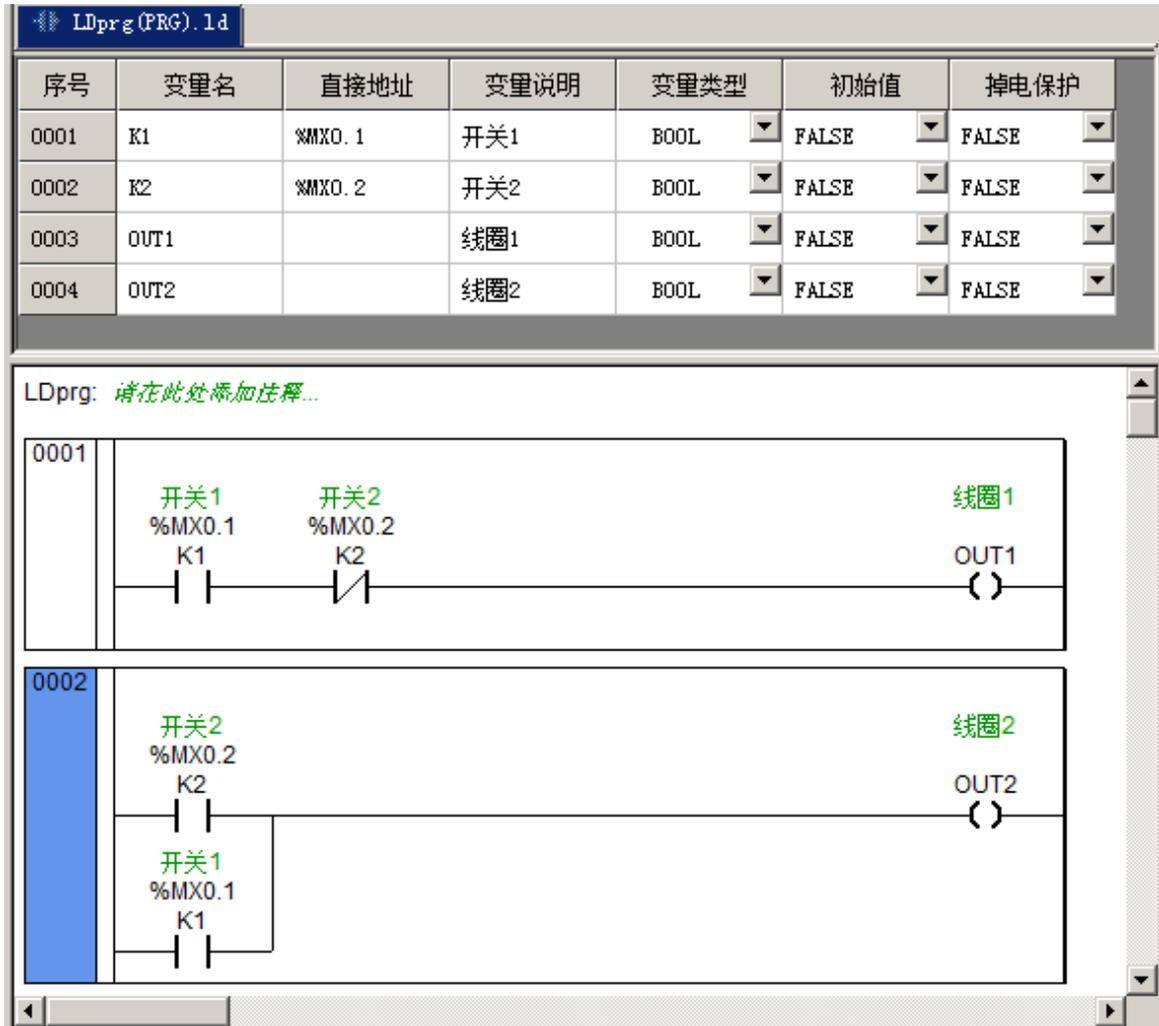


图 5.4-1 LD 编程窗口

一般称最左边的垂直线为“能量线”，其状态永远是真（TRUE）。各编程元件以一定的规则互相结合，最终连接到这条能量线上，形成一个个“节”、“段”或“网络”，完成特定的逻辑运算。每个节上的线称为母线，在母线可以添加触点、线圈、功能块。

5.4.1.1 LD 元素

1. 节

节是 LD 程序的基本单元，每个 LD 语言编写的 POU 都是由“节”组成。节号缺省从 0001 开始，每个 POU 中最多能添加 999 个节。



- 在 LD 语言编程区的每个节的网络中，横向不超过 32 列，纵向不超过 16 行。

2. 触点

每个触点代表一个 BOOL 型变量，可以是通道输入变量，如开关、按钮，也可以是内部变量。触点从左向右传递值。

| | : 常开触点，变量值为 TRUE 时，触点接通，能量流向右传递。否则，能量流在触点处断开。

3. 线圈

线圈代表逻辑运算的输出，用 () 表示，传递从左到右的能量流，可以接通道输出变量，控制设备的启停，马达的开、停、关等。

4. 块元件

块元件包括功能块、函数、程序这三种类型。详细说明参见章节 5.3.1 POU 的类型。

5. 图例说明

LD 程序由一系列的节组成，左右两边的能量线，以及中间母线上的元素组建了整个程序网络。

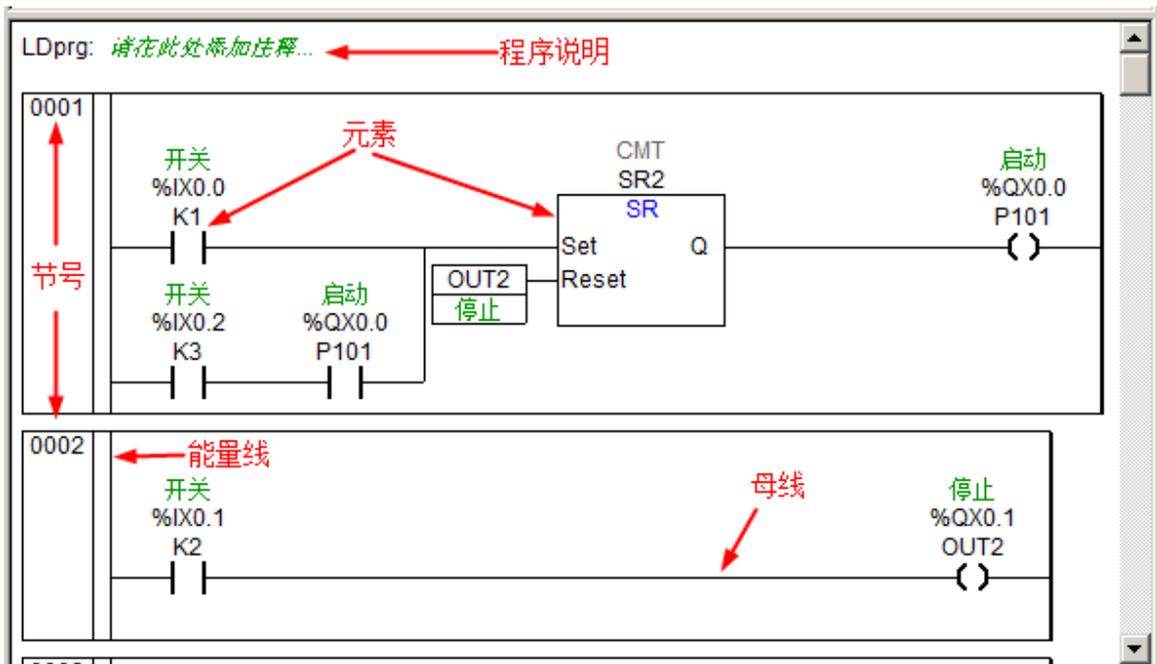


图 5.4-2 LD 语言编程元素图例说明

直接地址和变量说明的显示需要在菜单【工程】—【选项】中进行设置。

6. 注释说明

- 程序注释：对 POU 程序进行说明，如“泵的启动/停止”。
- 跳转标签注释：输入【跳转】命令的标号。默认不显示，需要在节点右键菜单中选择【跳转标号】命令才能显示标号区。最多显示 32 个字符，可通过右键菜单【删除】命令删除跳转标签注释。

- 网络节点注释：对当前节程序进行说明。默认不显示，需要在节点右键菜单中选择【修改注释】命令才能显示注释区。

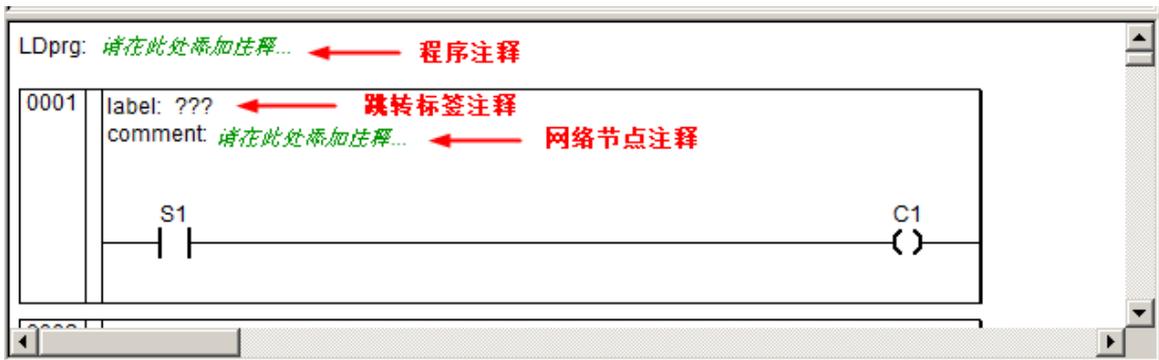


图 5.4-3 LD 语言编程注释说明

5.4.1.2 光标位置

光标位置即指当前的可编辑对象焦点，一般会以鲜明的颜色进行提示，如表 5.4-1 所示。

表 5.4-1 光标位置

光标位置	图例
文本域上	
触点上	
功能块上	
线圈上	
节上	
网络节点的注释上	



- 在 LD 语言编程区，当添加块元件时，缺省添加的均为 AND 功能块，且带有使能输入端子 EN 和使能输出端子 ENO（ENO 默认不显示），ENO 输出值等于 EN 端的输入值。

5.4.1.3 前节/后节

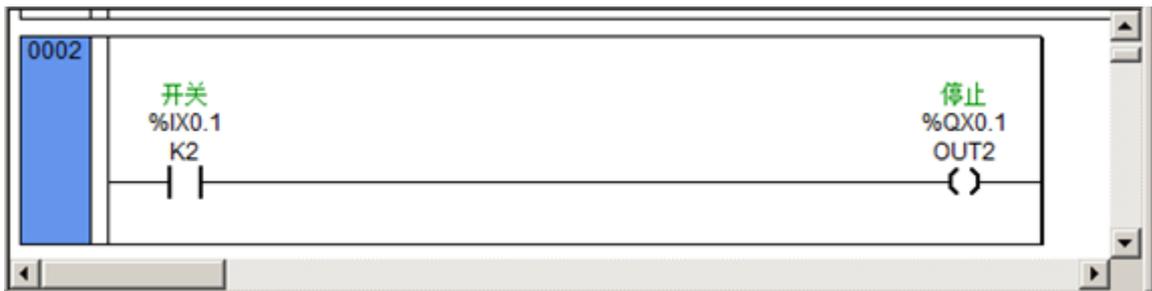
通过【前节】/【后节】命令，在光标所在当前节的前面或后面增加一节空节。

以【前节】为例进行说明。

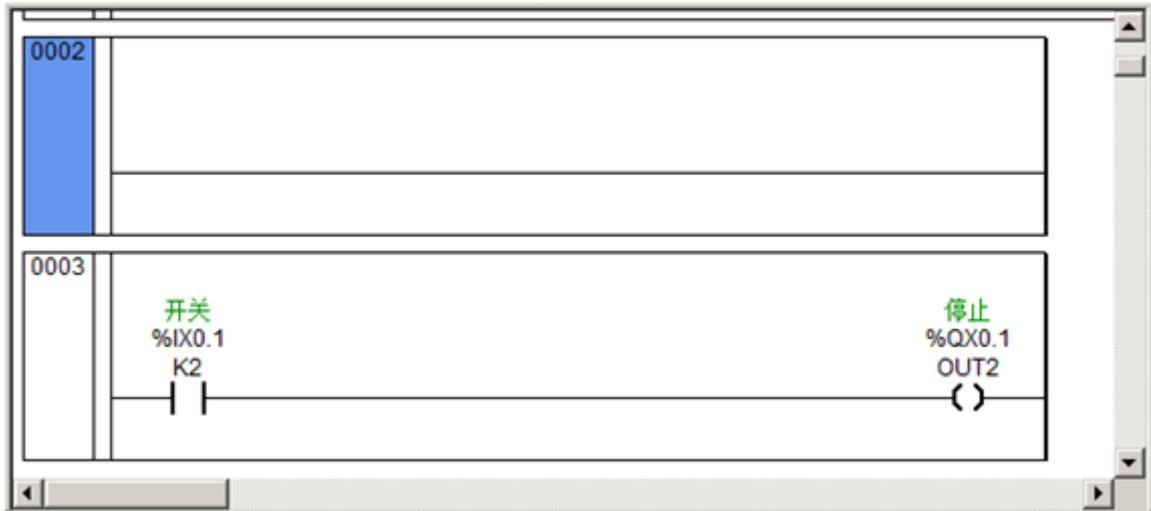


- 菜单栏：单击【插入】—【前节】；
- 程序区：右击节点，单击【前节】；
- 快捷键：**Ctrl+M**。

此时在选中节前面插入一个新节，如图 5.4-4 所示。



(a)



(b)

图 5.4-4 前节操作

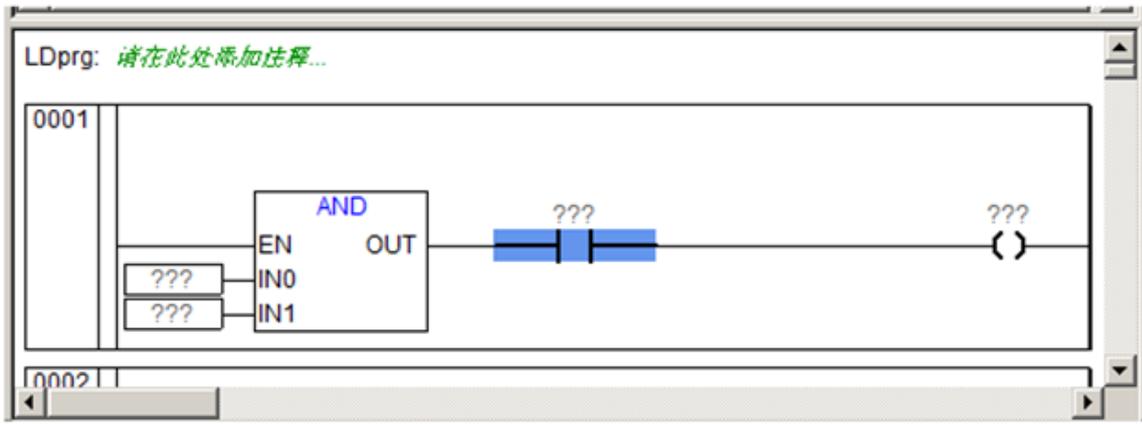
5.4.1.4 串联触点

选中触点、块元件、线圈元素，执行以下操作，插入串联触点。

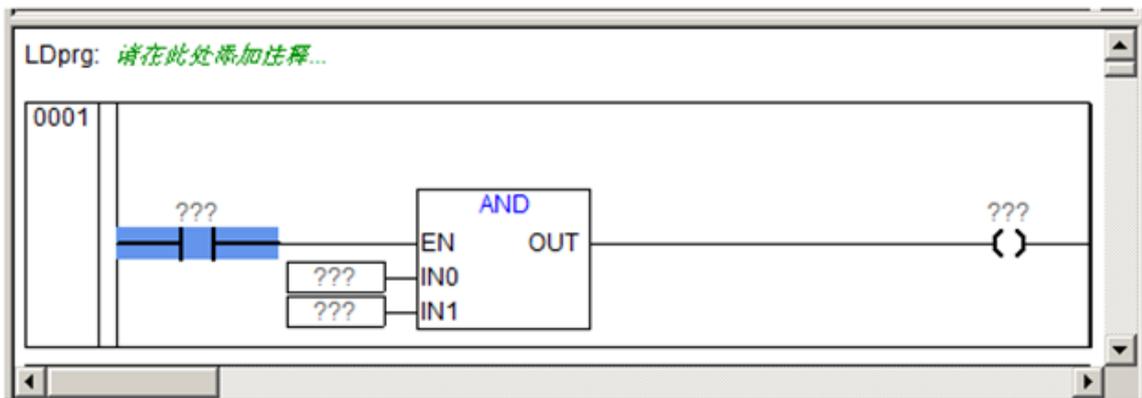


- 菜单栏：单击【插入】—【串联触点】；
- 程序区：右击元素，单击【追加触点】；
- 程序区：右击元素，单击【插入触点】；
- 工具栏：；
- 快捷键：**F3**。

选中功能块“AND”，执行【串联触点】或【追加触点】命令，插入触点的位置如图 5.4-5 (a) 所示。执行【插入触点】命令，插入触点的位置如图 5.4-5 (b) 所示。在线圈元素上执行此操作时，在线圈前面插入新触点。



(a)



(b)

图 5.4-5 插入“串联触点”的位置

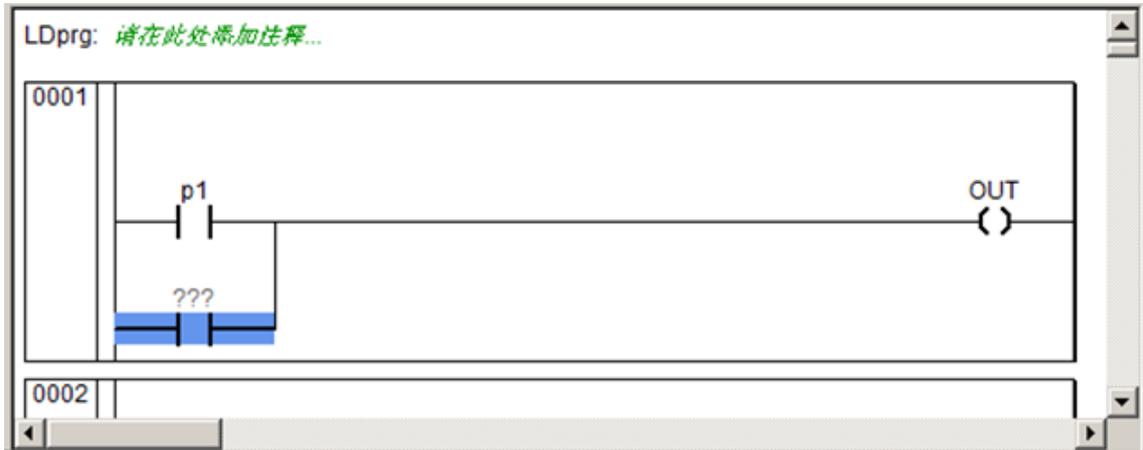
5.4.1.5 并联触点

选中触点或块元件执行以下操作，插入并联触点。

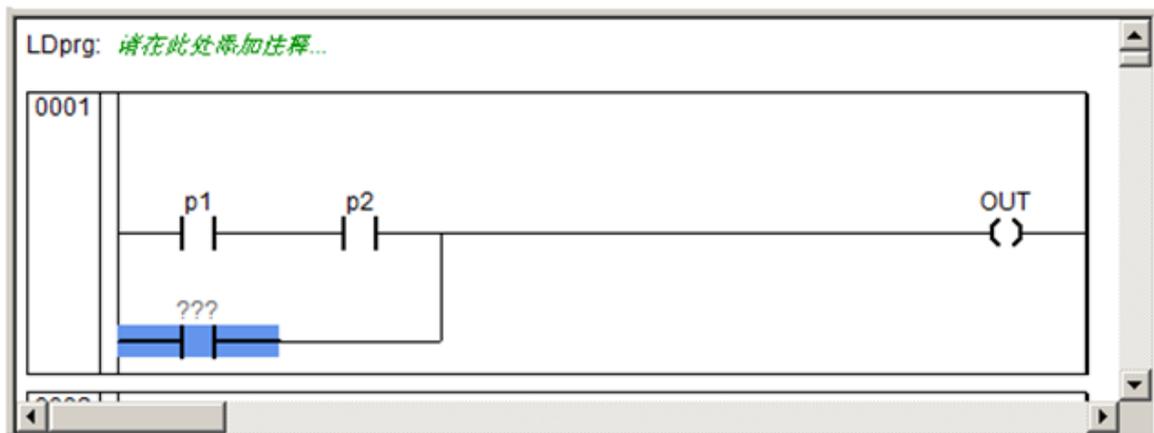


- 菜单栏：单击【插入】—【并联触点】；
- 工具栏：；
- 快捷键：**F10**。

单触点上执行以上操作，如图 5.4-6 (a) 所示。按住 **Shift** 键，选中多个串联触点，执行以上操作，如图 5.4-6 (b) 所示。



(a)



(b)

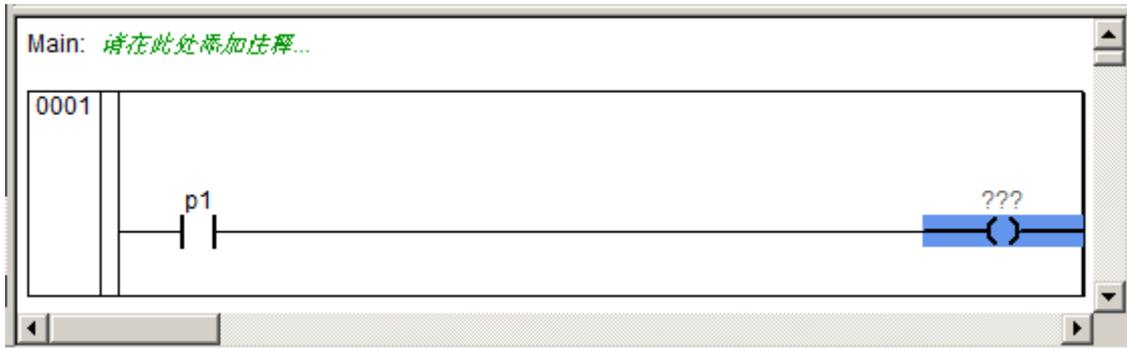
图 5.4-6 插入“并联触点”的位置

5.4.1.6 线圈

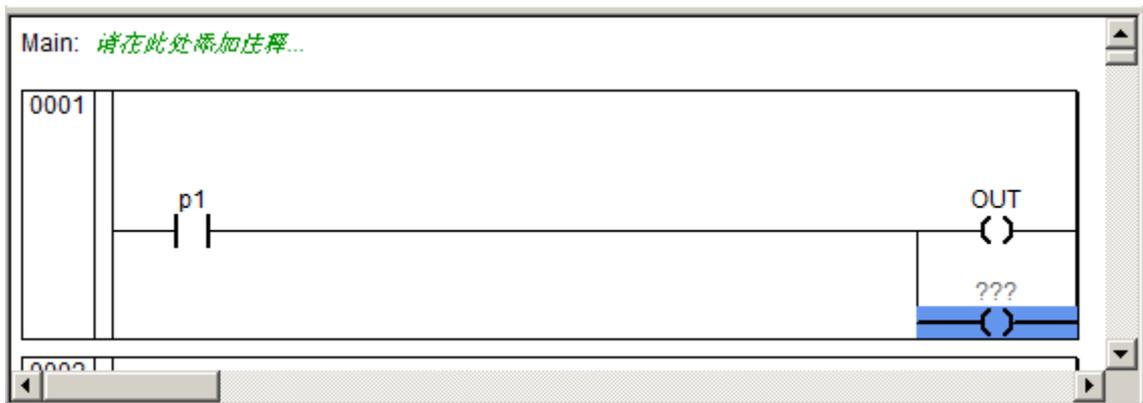
执行以下操作，在选中的当前节母线尾端上插入一个线圈，或者在选中的已有线圈下方插入一个新的线圈。如图 5.4-7 所示。



- 菜单栏：单击【插入】—【线圈】；
- 程序区：右击元素，单击【线圈】；
- 工具栏：；
- 快捷键：F6。



(a)



(b)

图 5.4-7 插入“线圈”的位置

5.4.1.7 并联块元件

LD 语言，组态时可以添加并行输出的块元件。



- 程序区：右击线圈，单击【并联块元件】。

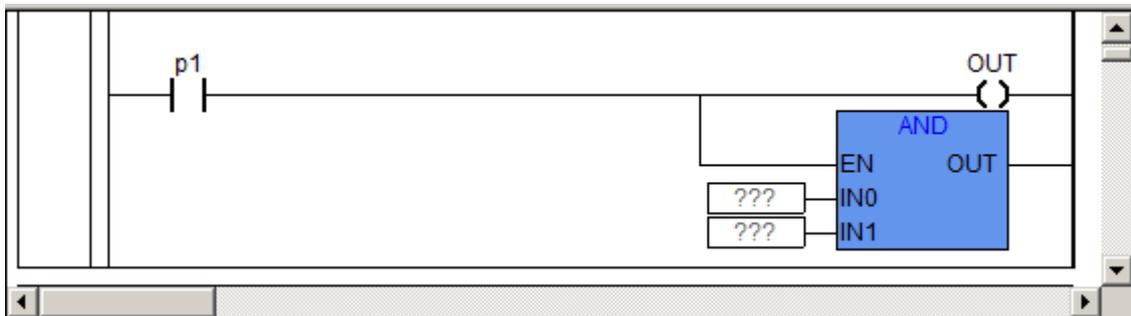


图 5.4-8 添加并联块元件

5.4.1.8 置反

选中触点、线圈或者块元件的引脚，执行以下操作，可以进行取非。



- 菜单栏：单击【插入】—【置反】；
- 程序区：右击菜单，单击【置反】；
- 工具栏：；
- 快捷键：**Ctrl+G**。

对触点 S1 置反，当 S1=FALSE 时，P101=TRUE；否则，当 S1=TRUE 时，P101=FALSE。

对线圈 P102 置反，当 S2=FALSE 时，P102=TRUE；否则，当 S2=TRUE 时，P102=FALSE。

对块元件引脚 IN0 置反，当 p1=FALSE 时，IN0=TRUE；否则，当 p1=TRUE 时，IN0=FALSE。

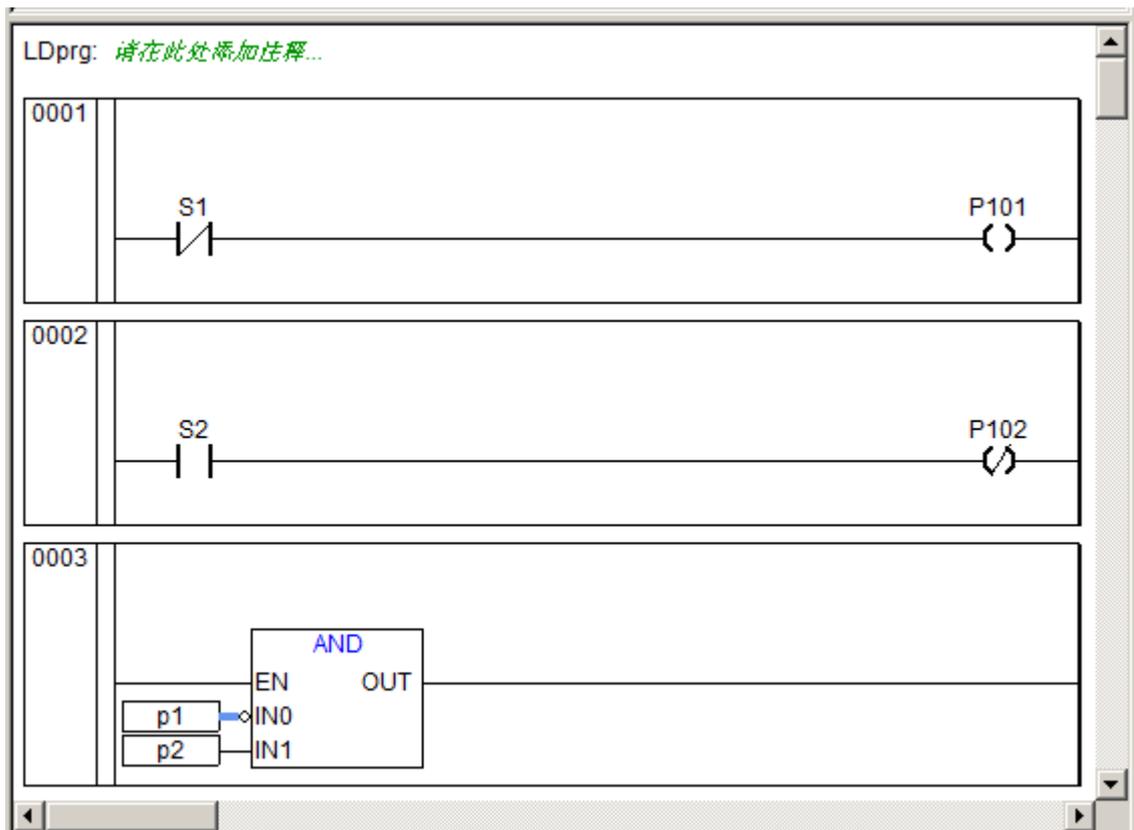


图 5.4-9 置反操作

5.4.1.9 置位/复位

置位(S) ：当运算条件满足，将线圈变量置为 TRUE，线圈将一直保持为 TRUE，即使运算条件不满足也保持为 TRUE，直到该变量被重新复位才变为 FALSE。

复位(R) : 当运算条件满足, 将线圈变量置为 FALSE, 该复位线圈将一直保持为 FALSE, 即使运算条件不满足也保持为 FALSE, 直到该变量被重新置位才变为 TRUE。



- 程序区: 右击线圈, 单击【置位/复位】;
- 快捷键: **Ctrl+T**;
- 工具栏: 。

线圈置位、复位状态如图 5.4-10 所示。

只有 S1 为 FALSE, 线圈 C1 被置位为 TRUE; 只有当 S2 变为 TRUE, C1 线圈被置位为 FALSE。

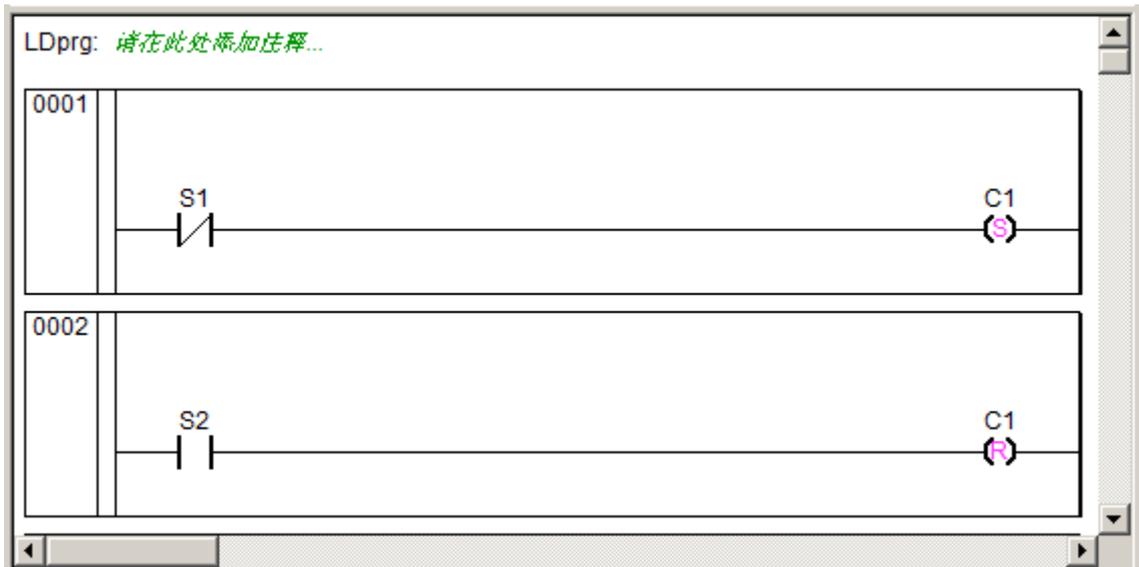


图 5.4-10 “置位/复位”操作

5.4.1.10 块元件



- 菜单栏: 单击【插入】—【块元件】;
- 程序区: 右击空白区域或元素, 单击【块元件】;
- 工具栏: ;
- 快捷键: **Alt+F9**。

在选中的当前位置插入一个块元件, 缺省为“AND”功能块, 且该块元件具有使能输入 EN, 使能输入端 EN 为 BOOL 类型。当使能输入端 EN 为 TRUE 值时, 运算才被执行。

1. 【高级】选项设置

如图 5.4-11 所示，块元件的【高级】菜单项可以进行多输入和引脚的显示/隐藏设置。

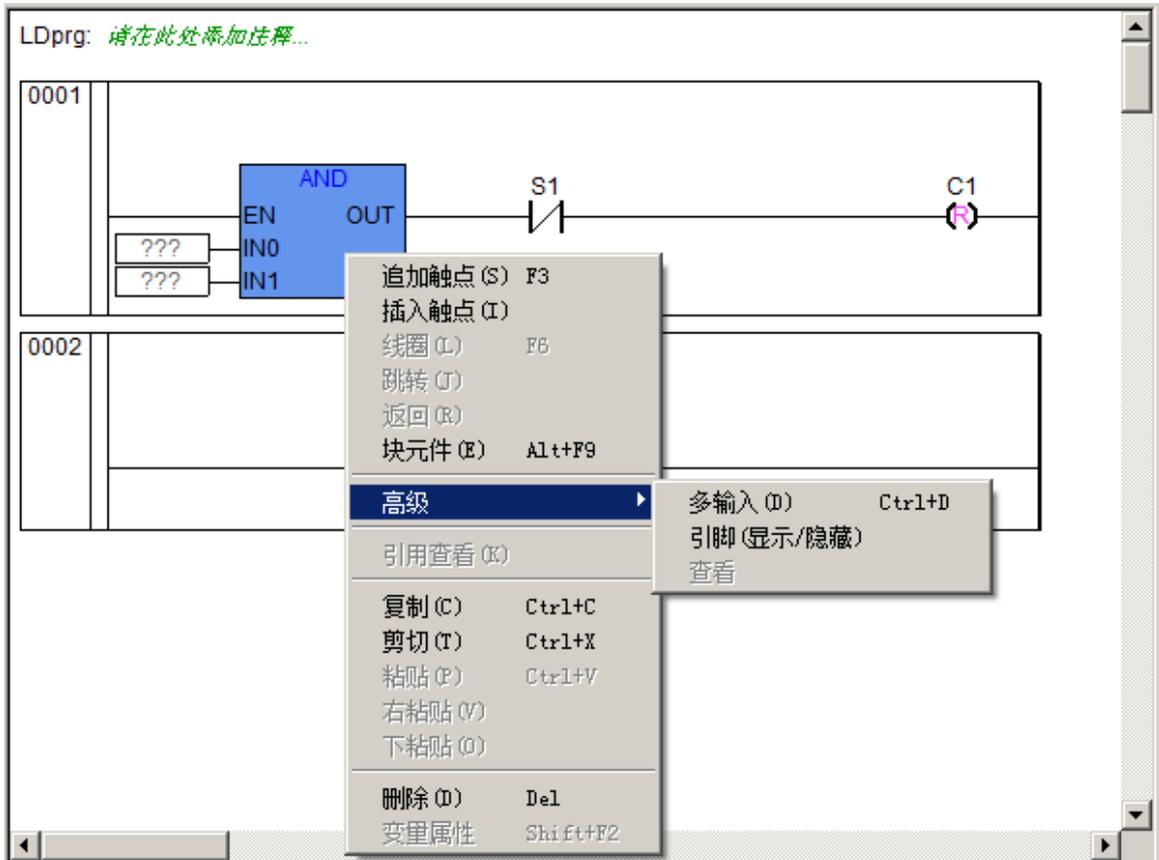


图 5.4-11 块元件“高级”菜单项

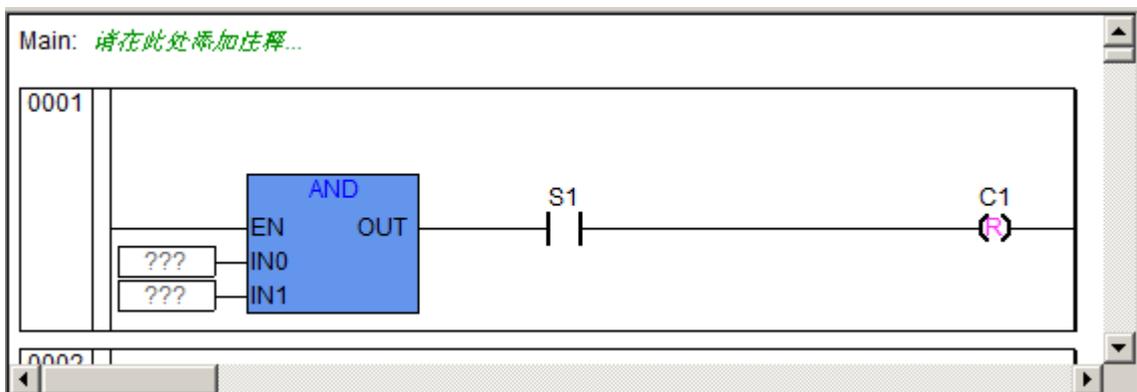
(1) 多输入

对选中的块元件或块元件引脚执行此命令，可以增加块元件的输入引脚。每执行一次，输入引脚增加一个。

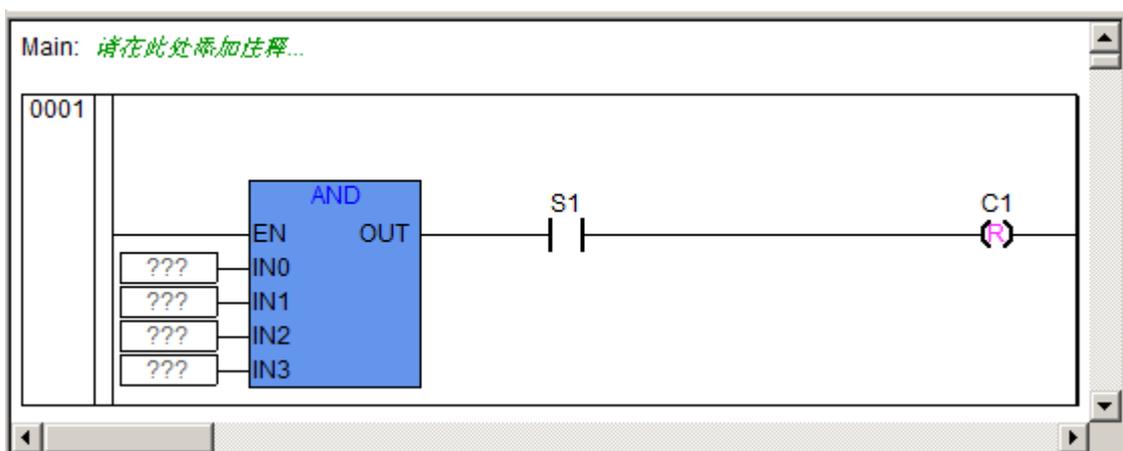


- 程序区：右击块元件，单击【高级】—【多输入 (D)】；
- 程序区：右击块元件引脚，单击【多输入 (D)】；
- 快捷键：**Ctrl+D**。

如图 5.4-12 (a) 所示，对选中的块元件执行【多输入】命令后，增加了输入引脚，如图 5.4-12 (b) 所示。



(a)



(b)

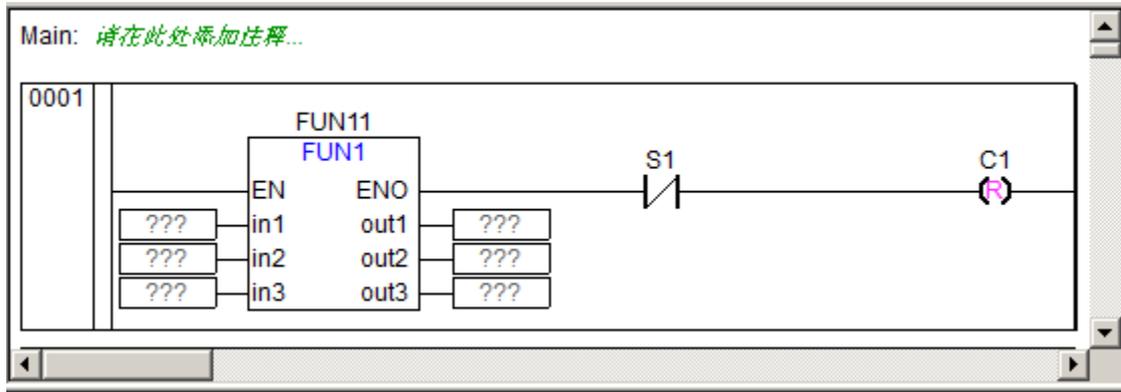
图 5.4-12 执行多输入命令

(2) 引脚（显示/隐藏）

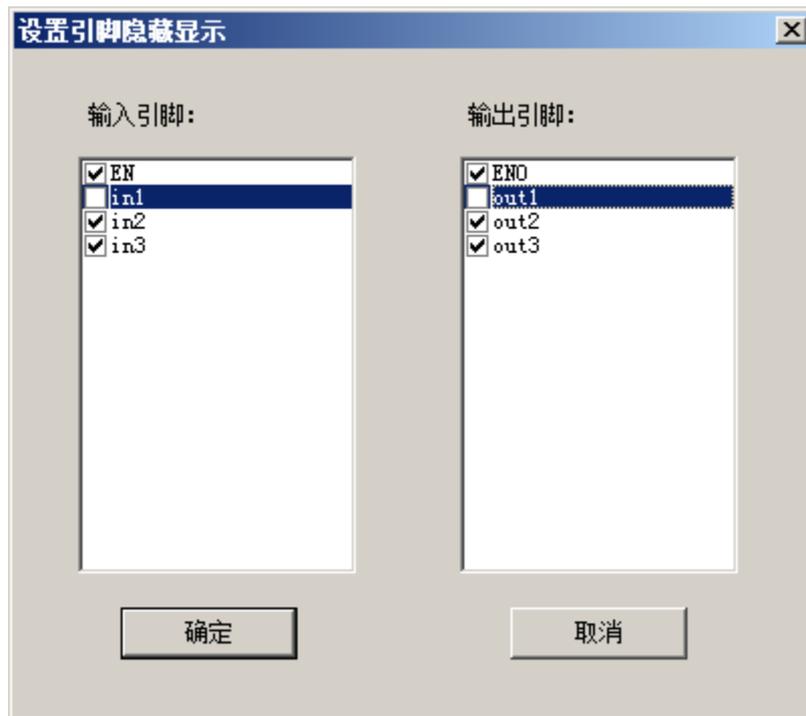
【引脚（显示/隐藏）】选项设置输入、输出引脚的显示与隐藏。

如 POU 中调用自定义功能块 FUN1，执行【引脚（显示/隐藏）】命令，可进行引脚显示/隐藏设置。当功能块引脚变化时，自动更新 POU 中已调用功能块的引脚。

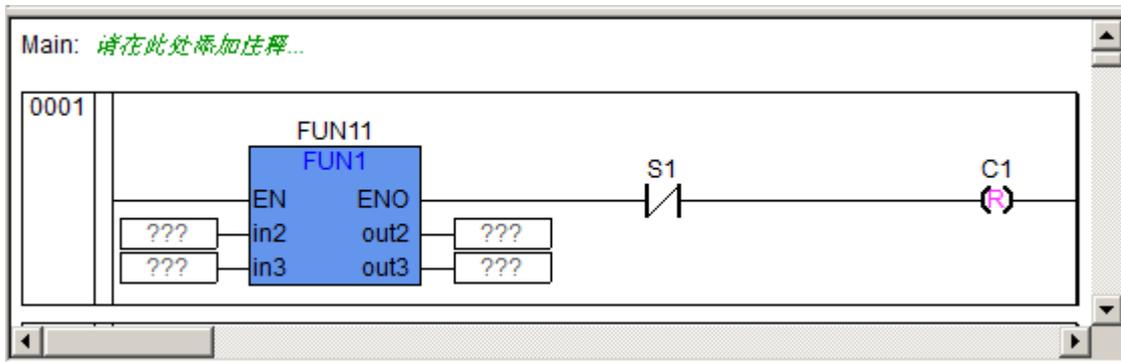
如图 5.4-13 (a) 所示，输入、输出引脚全部显示。如图 5.4-13 (c) 所示，隐藏输入引脚 in1、输出引脚 out1。



(a)



(b)



(c)

图 5.4-13 引脚的显示与隐藏设置

当自定义功能块 FUN1 增加引脚输入输出引脚 in4、out4 时，POU 中已调用的该功能块自动更新引脚。如图 5.4-14 所示。

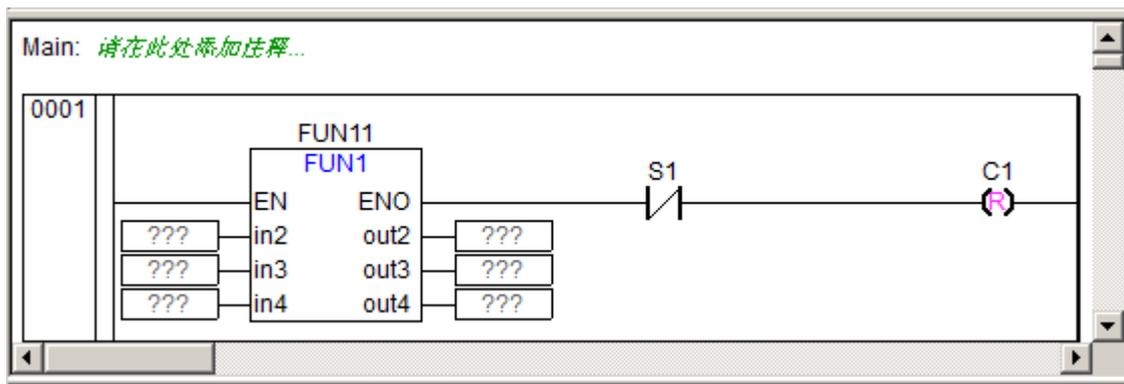


图 5.4-14 引脚自动更新

2. “功能块”与“使能运算符”的区别

功能块与使能运算符的根本区别在于执行方式不同。对于功能块，不论是否使能，在程序运行时，均会执行该功能块。然而对于使能运算符，只有在使能端 EN 有效时，才会执行。这里用 LD 语言编写一个简单程序进行介绍，如图 5.4-15 所示。

从图中可以看到，在 0001 节和 0002 节中分别以功能块和使能运算符调用通电延时定时器。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	M1_1			BOOL	FALSE	FALSE
0002	M1_2			BOOL	FALSE	FALSE
0003	T1			TON		FALSE
0004	ET1			TIME	T#0MS	FALSE
0005	M2_1			BOOL	FALSE	FALSE
0006	M2_3			BOOL	FALSE	FALSE
0007	M2_2			BOOL	FALSE	FALSE

Main: 请在此处添加注释...

图 5.4-15 功能块与使能运算符

程序开始运行时，首先使 M1_1、M2_1 和 M2_3 同时接通，此时两个定时器运行状态一致。当断开 M1_1 和 M2_1，M2_3 保持接通，此时 ET1 清零，而 ET2 不清零。

然后再断开 M2_3，ET2 时间仍不清零。只有当 M2_1 保持状态接通，检测到 M2_3 为断开，ET2 才会清零。这是因为只有使能端有效，才会进入“TON”内部执行代码。

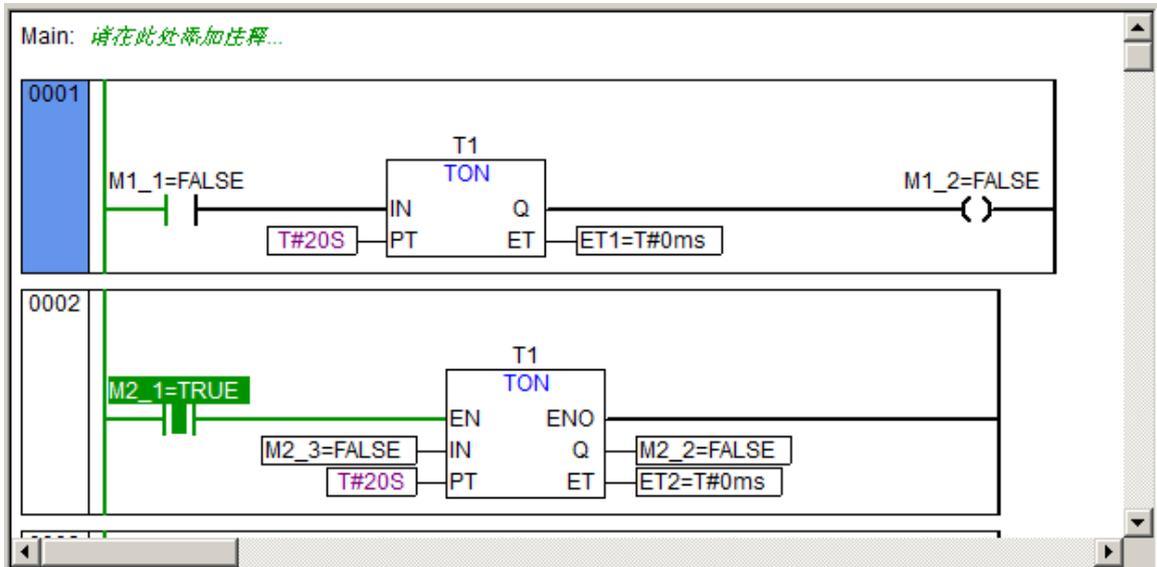


图 5.4-16 功能块与使能运算符—运行分析图

采用使能运算符调用子程序，当使能端无效时，子程序不会执行。采用功能块调用子程序，功能块使能时，运行使能部分；功能块不被使能时，便运行非使能分支部分。

3. 级联组态

级联组态指块元件的输入引脚可以连接其它块元件和赋值，右击块元件的输入引脚，可以选择【赋值】和【块元件】命令，如图 5.4-17 所示。

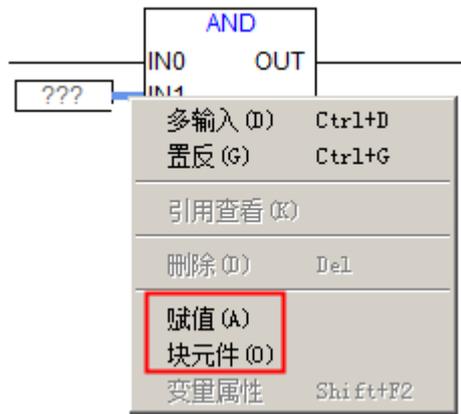


图 5.4-17 块元件输入引脚的右键菜单

(1) 赋值

块元件的输入引脚级联对象为【赋值】]时，会同时添加两个元件，级联结果如图 5.4-18 中的位置 ①②所示。

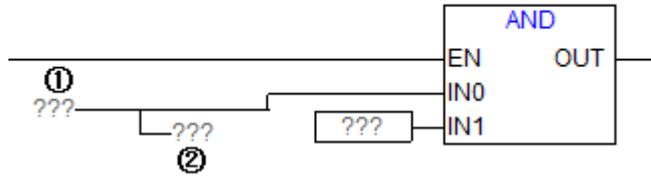


图 5.4-18 级联“赋值”元素的结果

其中，位置①是“赋值”元件的输入端，位置②是“赋值”元件的输出端；并且位置①同时给位置②以及“AND”块的 IN0 项赋值。

“赋值”元件不支持复制、粘贴、剪切操作。

可以通过节点的【删除】命令移除已级联的“赋值”元件；删除的范围与选取的节点位置有关，如图 5.4-19 所示。

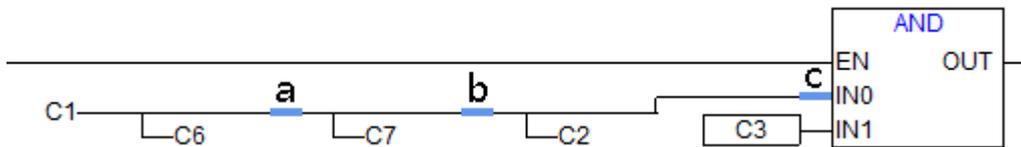


图 5.4-19 级联节点

通过图 5.4-19 节点(a)可以删除端点 C1 和 C6；

通过图 5.4-19 节点(b)可以删除端点 C1、C6 和 C7；

通过图 5.4-19 节点(C)可以删除端点 C1、C6、C7 和 C2；

通过位置①右键菜单中的【赋值】或【块元件】命令，可以对位置①进行下一级的扩展。

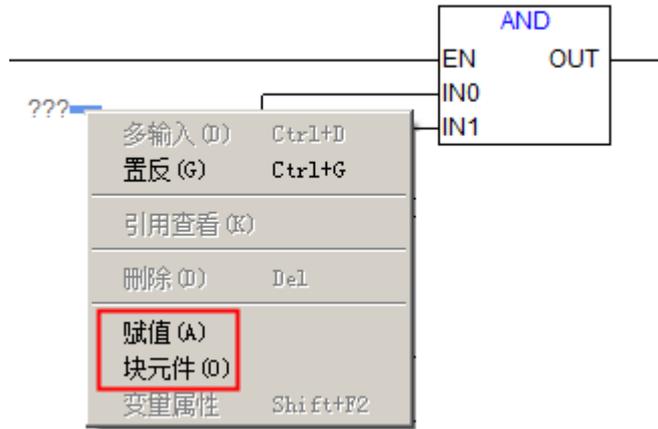


图 5.4-20 输入端的右键菜单

用户通过位置②右键菜单中的【多输出】命令，可以增加输出端，对位置②进行并行扩展。如图 5.4-21 所示。



图 5.4-21 输出端的右键菜单

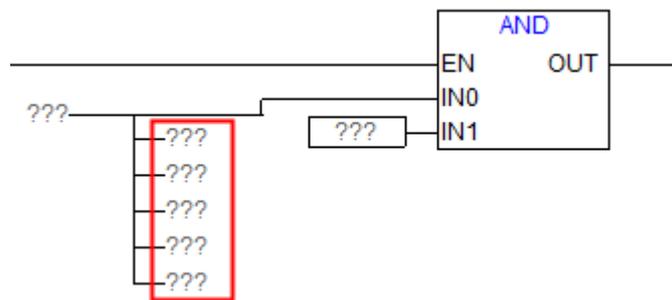


图 5.4-22 多个输出端

通过节点的【删除】命令可以移除输出端。删除的范围与选取的节点位置有关，如图 5.4-23 所示。

通过图 5.4-23 节点(a)可以删除端点 C61~C64；

通过图 5.4-23 节点(b)可以删除端点 C7;

通过图 5.4-23 节点(C)可以删除端点 C2;

通过图 5.4-23 节点(d)可以删除端点 C61/C62/C63/C64。

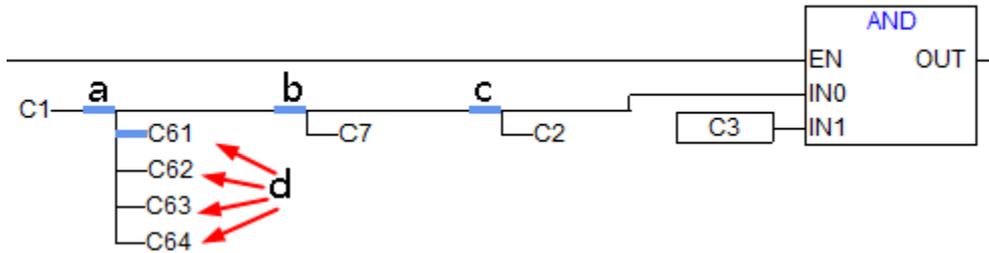


图 5.4-23 输出端节点

(2) 块元件

块元件的输入引脚级联对象为【块元件】时，会缺省添加一个“AND”块，级联结果如图 5.4-24 所示。

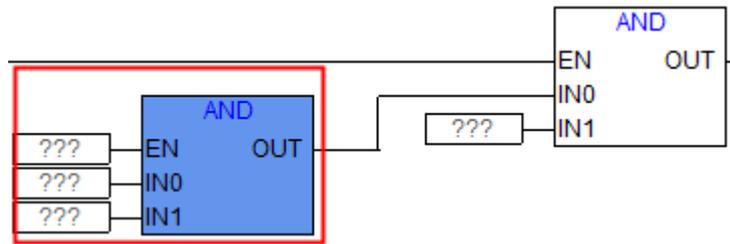


图 5.4-24 级联“块元件”的结果

级联【块元件】时，缺省连接块元件的第一个输出引脚（如图 5.4-24 中的 OUT 端），当用户需要变更连接的输出端时，可以通过【引脚（显示/隐藏）】命令调整。

删除块元件时的操作方法及影响范围，可参考“赋值”元件的相关内容，这里不再赘述。

5.4.1.11 跳转和返回

跳转和返回都是对程序执行顺序的改变。在正常情况下，CPU 模块将根据主程序中节的顺序进行执行。

(1) 跳转：

当条件满足后，则跳转至指定标号的相应节。

光标位于节，通过以下方式插入跳转指令。



- 菜单栏：单击【插入】—【跳转】；
- 程序区：右击触点或线圈，单击【跳转】。

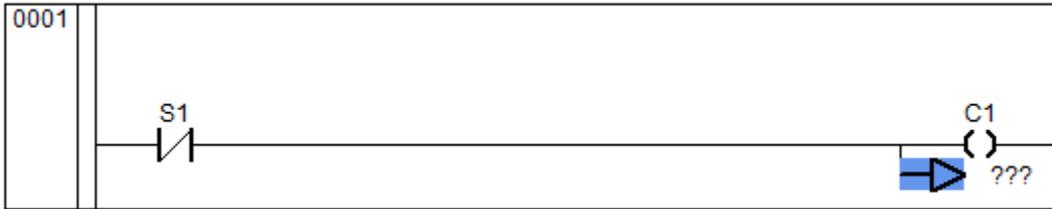


图 5.4-25 插入跳转

插入跳转后，需要输入跳转标号（缺省为“???”），跳转标号用于识别跳转目的地。输入跳转标号“lable”，如图 5.4-26 所示，则跳转条件满足后，直接跳转至 0003 节，不再执行 0002 节的程序。

每个网络都有一个标号，位于网络的首行，缺省为空。鼠标单击该节的首行位置，即显示为亮蓝色的矩形输入框，能且只能输入跳转标签，如图 5.4-26 中 0003 节的跳转标签“lable”所在位置。

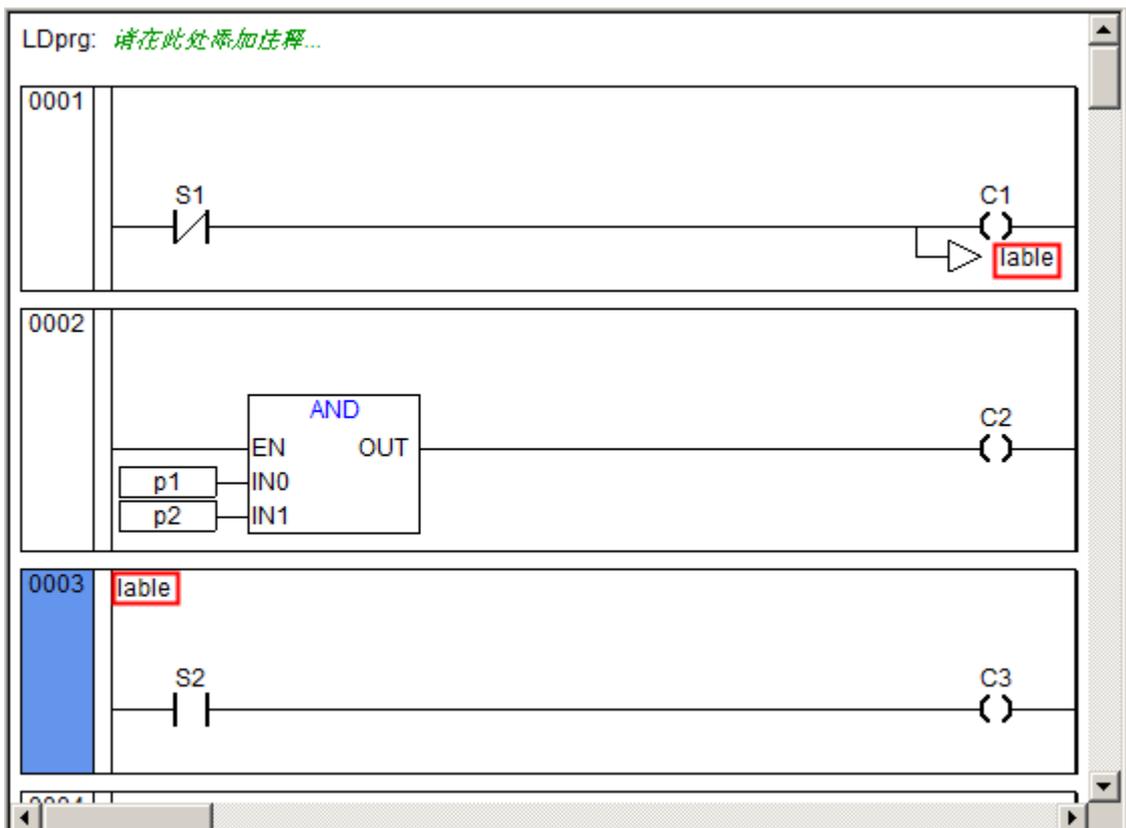


图 5.4-26 跳转

(2) 返回

当调用 POU 时，可以利用【返回】命令，当条件满足后，被调用的 POU 不再继续执行，而返回调用的 POU 中。

光标位于节，通过以下操作插入【返回】命令。



- 菜单栏：单击【插入】—【返回】；
- 程序区：右击元素，单击【返回】。

插入返回符，如图 5.4-27 中的 0002 节所示，默认为 Return。当 0002 节的触点 O 闭合接通时，跳出程序，返回到调用它的 POU，不执行 0003 节及以下的程序。

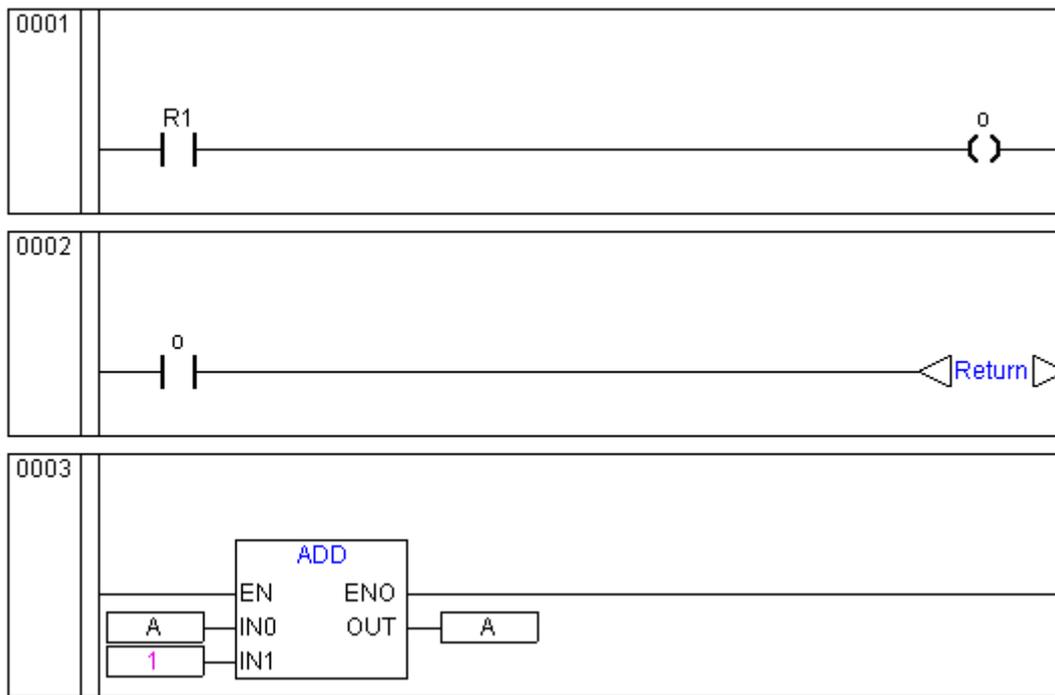


图 5.4-27 返回实例

5.4.1.12 立即输出

无论程序周期为多少，只要运行至该立即输出线圈位置，即将该线圈值直接输出到关联的物理输出信号进行输出。这里，设置为【立即输出】属性的线圈必须关联物理的开关量输出变量。



- 程序区：右击线圈，单击【立即输出】。



图 5.4-28 设置立即输出线圈



- MC、LK 平台不支持此功能。

5.4.1.13 移动

选中元素拖动，此时所有可以移动到的元件前后（线圈只有左侧的移动矩形框）处有亮蓝标识■，如图 5.4-29 所示，移动元件时，离鼠标最近的方框显示红色高亮■，松开鼠标元件移动到该位置。

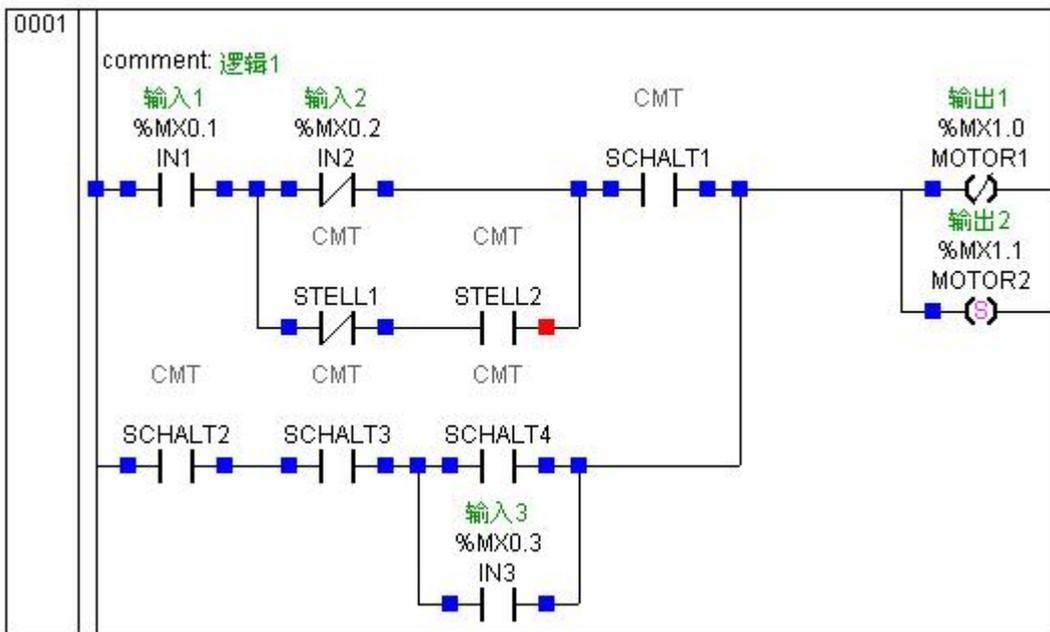


图 5.4-29 移动元素

5.4.1.14 复制/粘贴/剪切/删除

(1) 复制/剪切

在程序区选中所要进行复制操作的节、触点、线圈、块元件，执行以下操作：



- 程序区：右击元素，单击【复制】/【剪切】。

(2) 粘贴

将复制、剪切的元素粘贴到目标位置。

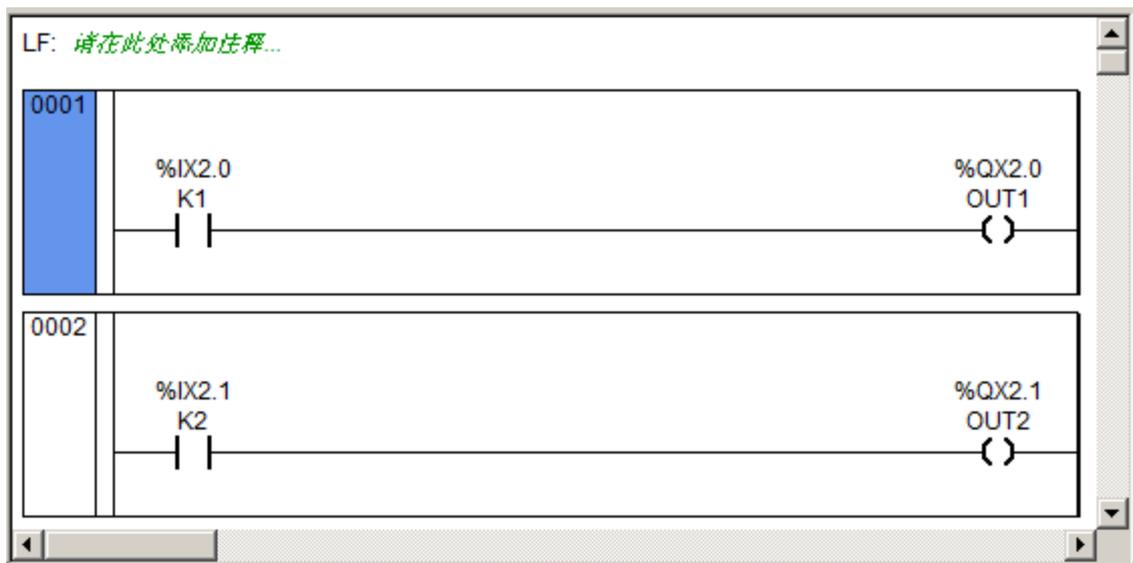


- 程序区：右击节的空白区域，单击【粘贴】；
- 快捷键：**Ctrl+V**。

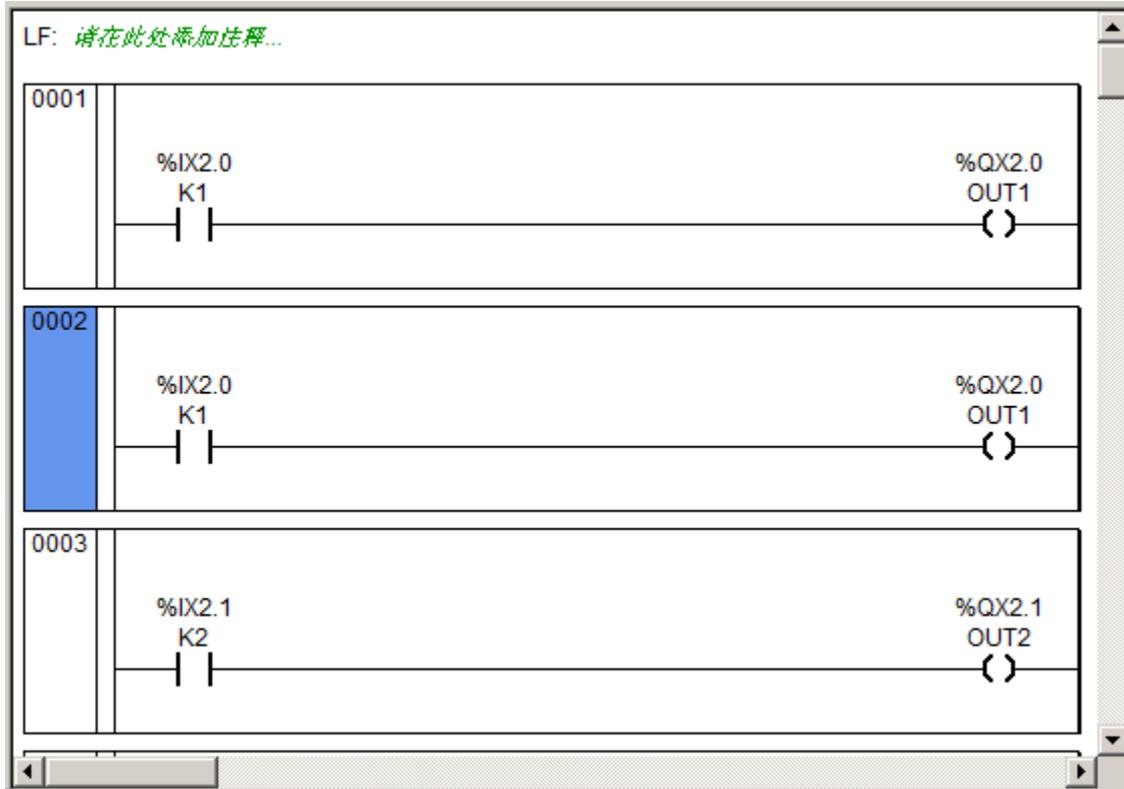
■ 节的粘贴

选中目标节，执行【粘贴】操作后，在目标节位置插入已复制、剪切的节。

【复制】0001 节，右击 0002 节执行【粘贴】操作，则在 0002 节插入已复制的 0001 节，如图 5.4-30 所示。



(a)



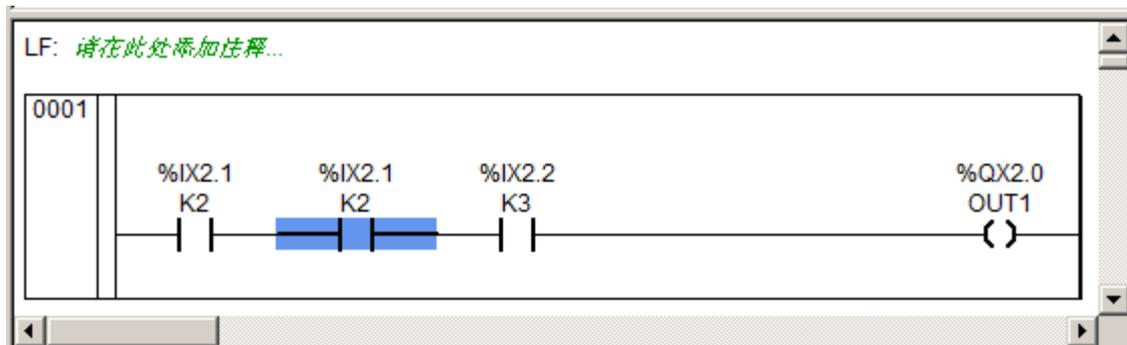
(b)

图 5.4-30 节的粘贴

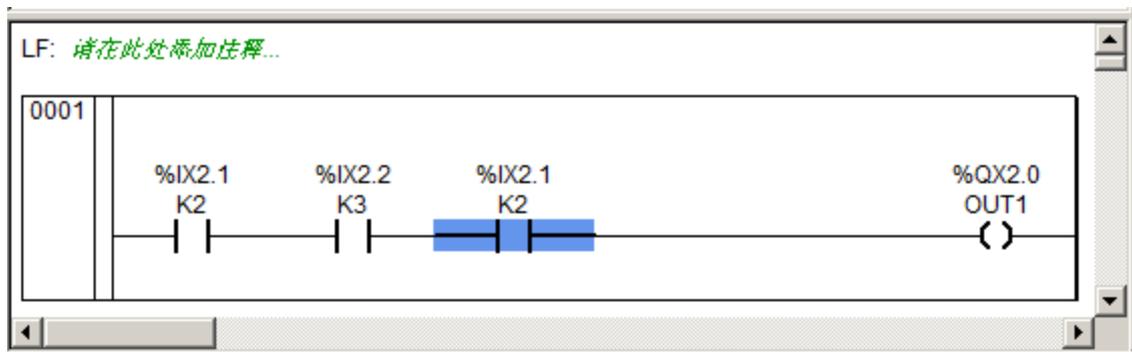
■ 触点、块元件的粘贴

对已复制、剪切的触点、块元件可以进行【粘贴】、【右粘贴】、【下粘贴】操作，以触点为例进行说明。

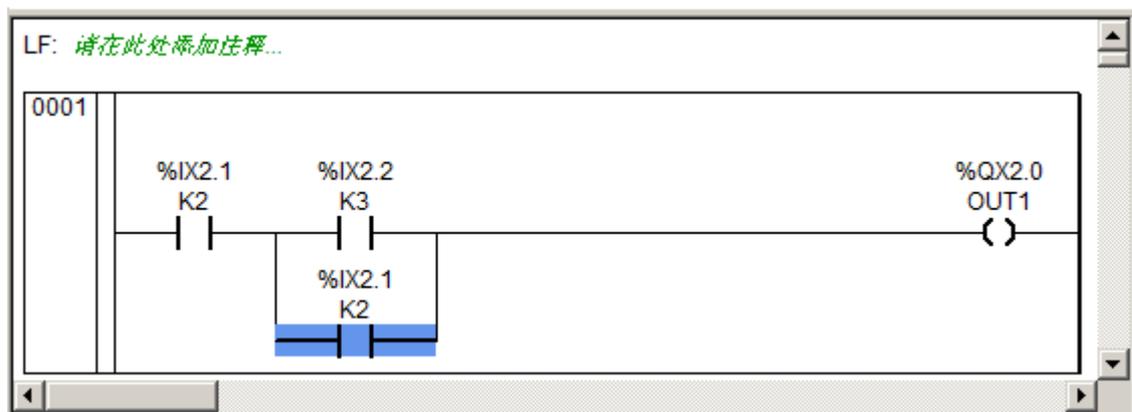
【复制】触点 K2，在触点 K3 上右键执行【粘贴】、【右粘贴】、【下粘贴】操作，分别如图 5.4-31 (a)、(b)、(c) 所示。



(a)



(b)



(c)

图 5.4-31 触点的粘贴

■ 线圈的粘贴

对已复制、剪切的线圈进行【粘贴】操作时，若目标节点上已有线圈，粘贴后的线圈将并联插入到已存在线圈的下面。

复制线圈 P101，在节 0002 上执行【粘贴】操作，如图 5.4-32 所示。

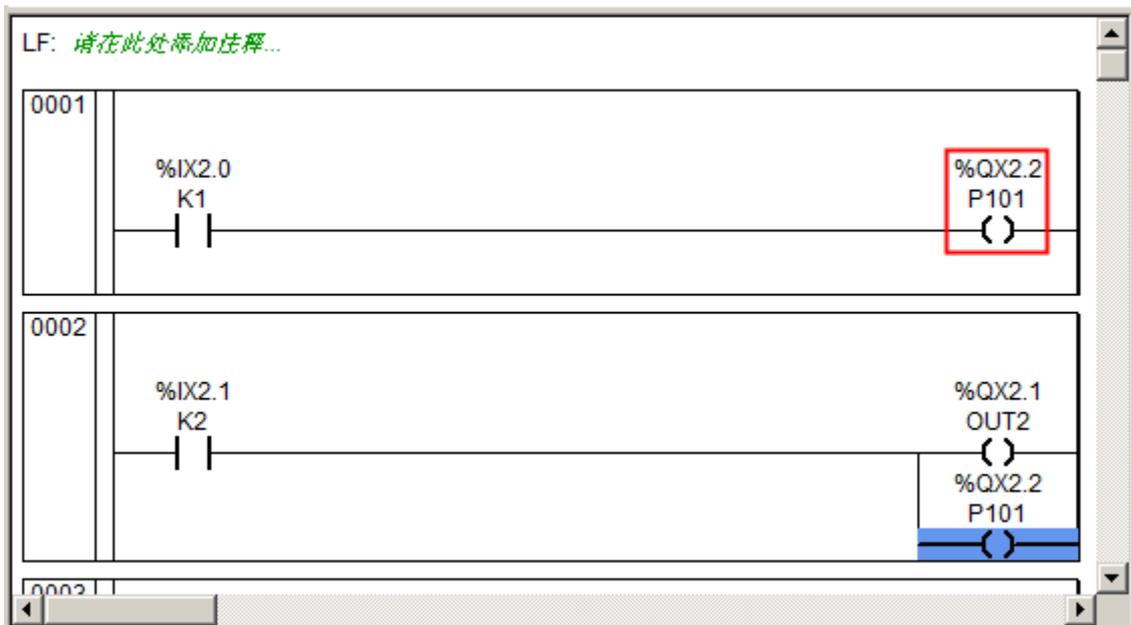


图 5.4-32 线圈的粘贴

(3) 删除

选中要删除的元素，执行以下操作：



- 程序区：右击元素，单击【删除】；
- 快捷键：**Delete**。

5.4.2 ST 编辑器

ST 是结构化文本（Structured Text）的简称，是一种文本化的编程语言。

POU 运算过程是由语句决定的。

库管理器中所有的函数和功能块都可以在 ST 语言中被调用。函数被调用时的部分显示范围被操作符遮盖了，功能块在 ST POU 中声明后可以被使用。他们被参数化的方法与在 LD 或 CFC 语言中一样。

执行顺序可以通过编排在 ST 编辑器中的语句顺序得到，从左到右，从上到下。这个顺序只能通过插入循环语句而改变。

ST 编程窗口如图 5.4-33 所示。

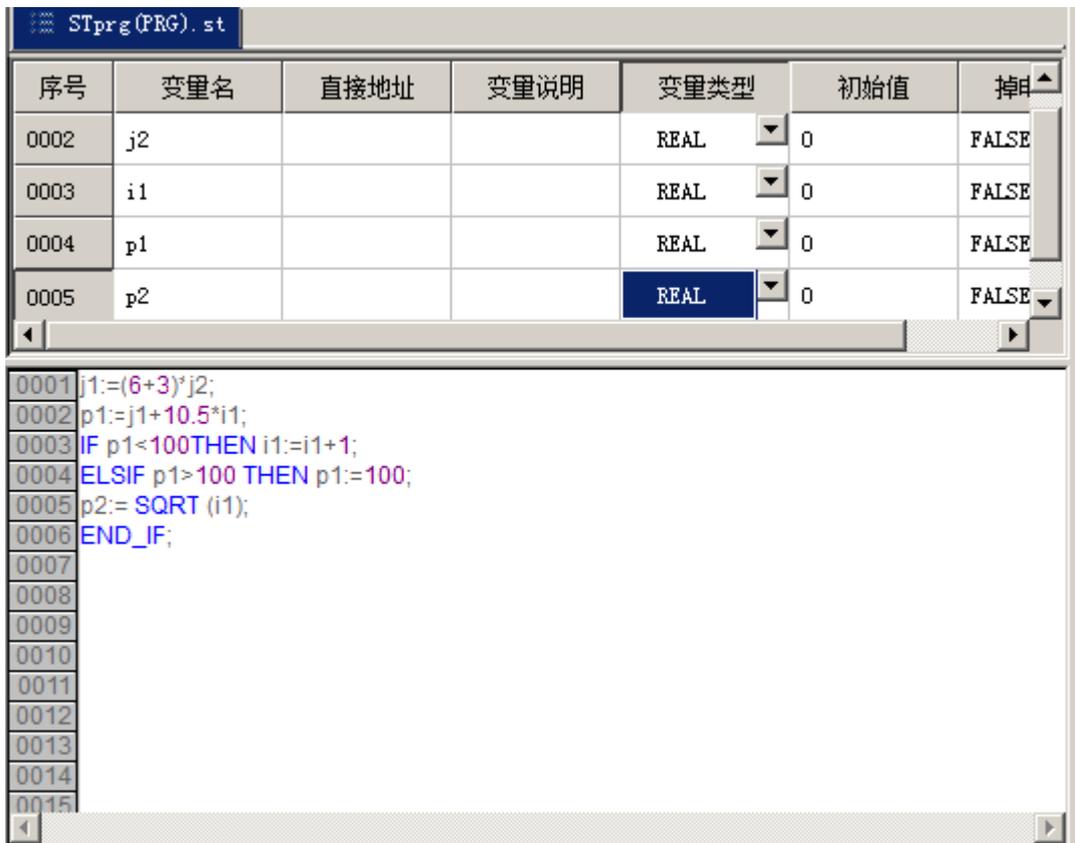


图 5.4-33 ST 编程窗口

ST 语言的工具栏除基本的按钮外，还提供 **FOR** 和 **IF** 按钮，在编辑区光标处单击这两个按钮，则可实现模板语句的调用，方便用户进行编程。

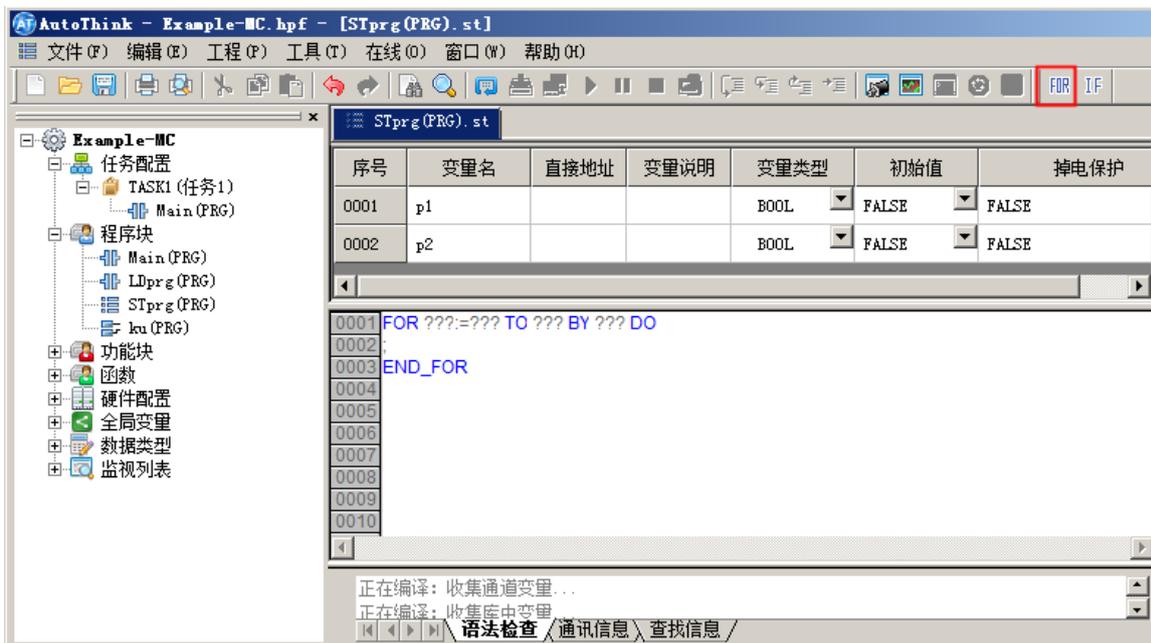


图 5.4-34 插入 FOR 语句

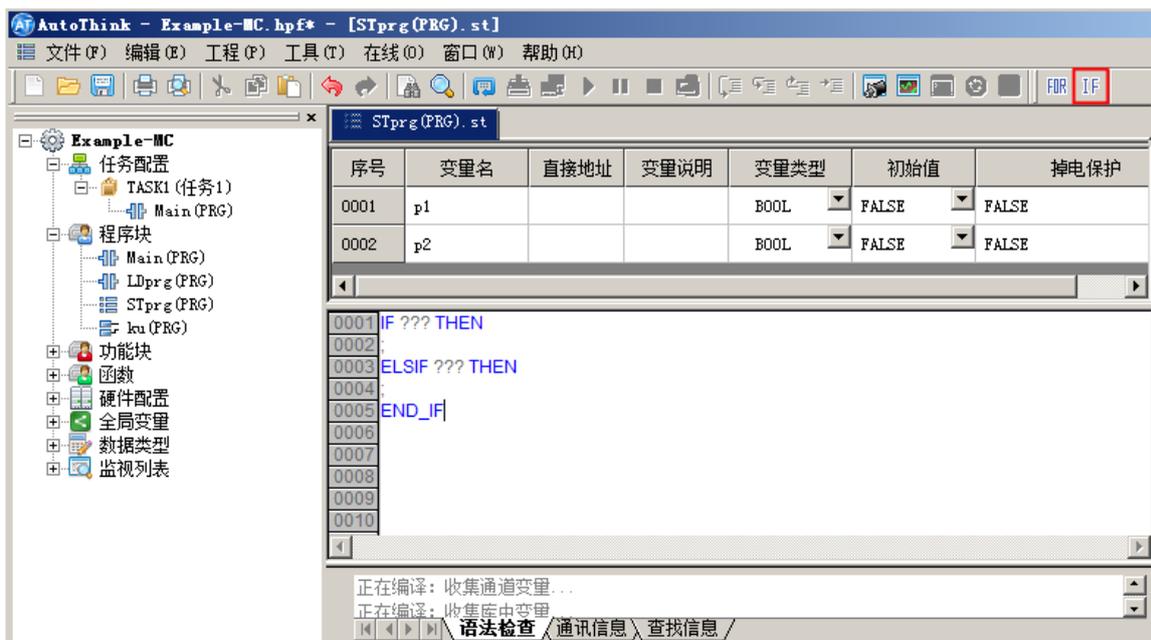


图 5.4-35 插入 IF 语句

5.4.2.1 ST 元素

使用结构化文本（ST）的编程语言，可以执行多种操作，例如调用功能块、赋值和有条件地执行指令和重复任务。

ST 的编程语言由各种元素组成，具体如下：

- 表达式：由操作数和操作符组成的结构，在执行表达式时会返回值。
- 操作数：操作数表示变量，数值，地址，功能块等。
- 操作符：操作符是执行运算过程中所用的符号。
- 语句：语句用于将表达式返回的值赋给实际参数，并构造和控制表达式。

各元素的图例说明，如图 5.4-36 所示。



图 5.4-36 ST 语言表示形式



- ST 语言编辑时，单个 POU 的编辑内容不能超过 9999 行。

5.4.2.2 表达式

表达式是返回变量评估值的结构。在语句中需要使用该返回值。表达式由操作符和操作数组成。操作数可以是常量、变量、函数调用的返回值或其他表达式。举例：

- 12 (*常量*)
- Var1 (*变量*)
- Fun(a,b,c) (*函数调用*)
- a + b (*操作符调用*)
- l+ (x*y*z) (*操作符调用*)

计算表达式时将根据操作符的优先级所定义的顺序将操作符应用于操作数表。首先执行表达式中具有最高优先级的操作符，接着执行具有次优先级的操作符；以此类推，直到完成整个计算过程。优先级相同的操作符将根据它们在表达式中的书写顺序从左至右执行。可使用括号更改此顺序。

例如，如果 A、B、C 和 D 的值分别为 1、2、3 和 4，并按以下方式计算： $A-B-C*D$ ，结果为-13。 $(A-B-C)*D$ ，结果则为-16。

如果操作符包含两个操作数，则先执行左边的操作数，例如在表达式 $SIN(x)*COS(y)$ 中，先计算表达式 $SIN(x)$ ，后计算 $COS(y)$ ，然后计算二者的乘积。

5.4.2.3 ST 操作符

操作符是一种符号，它表示要执行的算术运算、要执行的逻辑运算、功能编辑调用。操作符是泛型的，即它们自动适应操作数的数据类型。

常见 ST 操作符及其执行的操作如表 5.4-2 所示。

表 5.4-2 常见 ST 操作符及其执行的操作

操作符	执行的操作	优先级
(表达式)	括号	高
函数名 (参数列表)	调用函数	
EXPT	指数运算	
-, NOT	取反	
*	乘	
/	除	
MOD	取余数	
+	加	
-	减	
<, >, <=, >=	比较运算	
=	等于	
<>	不等于	

操作符	执行的操作	优先级
AND	与	
XOR	异或	
OR	或	

5.4.2.4 操作数

操作数可以是：地址、数值、变量、数组变量、结构体变量、数组/结构体变量的元素、功能调用、功能块输出等。处理操作数的语句中的数据类型必须相同。如果需要处理不同类型的操作数，则必须预先执行类型转换。

在下面示例中，整数变量 Var1 在添加到实数变量 R1 中之前会先转换为实数变量。

```
R2:=R1+SIN (INT_TO_REAL (Var1));
```

5.4.2.5 语句

1. 赋值语句

(1) 语义

赋值语句用表达式的求值结果替代单元素或多元素变量的当前值。

赋值表达式的结构为：语句 $A := B$ ；（A 是变量名称；B 是求值的表达式）

两个变量（分别位于赋值操作符的左侧和右侧）的数据类型必须相同。数组是个特例。显式启用后，也可对长度不同的两个数组执行赋值操作。赋值用于将一个变量的值赋给另一个变量。

(2) 示例

■ 将数值赋给变量。

如果 A 和 B 是基本数据类型，则 B 的单个值会传递给 A。如果 A 和 B 是导出的数据类型，则所有 A 的所有元素的值都由变量 B 对应元素的当前值替代。

□ 语句 $C := 25$;

用于将值 25 赋给变量 C。

■ 将运算值赋给变量

□ 语句 $X := (A+B-C)*D$;

用于将 $(A+B-C)*D$ 的运算结果赋给变量 X。

■ 将 FUN/FB 的值赋给变量

赋值用于将功能或功能块返回的值赋给变量。

□ 语句 A := MY_TON.Q;

用于将 MY_TON 功能块 (TON 功能块的实例) 的 Q 输出值赋给变量 A。(这不是功能块调用)

□ 语句 B := C MOD A;

用于调用 MOD (模数) 功能并将计算结果赋给变量 B。

2. 选择语句 IF...THEN...END_IF

(1) 语义

IF 语句只有确定其相关布尔表达式的值为 1(真)时,才会执行语句或一组语句。如果条件为 0(假),将不会执行该语句或语句组。THEN 语句标识条件的结尾和语句的开头。END_IF 关键字标记语句的结尾。



- 可以嵌套任何数量的 IF...THEN...END_IF 语句,以生成复杂的选择语句。

(2) 示例

■ IF...THEN...END_IF

该条件可以使用布尔变量表达。

如果 FLAG 为 1,将执行语句;如果 FLAG 为 0,则不会执行。如图 5.4-37 所示。

```

0001 IF FLAG THEN
0002 C:=SIN(A)*COS(B);
0003 B:=C-A;
0004 END_IF;
0005

```

图 5.4-37 示例 1

该条件可使用返回布尔结果的操作表达。如果 A 大于 B,将会执行语句;如果 A 小于或等于 B,则不会执行。

■ IF NOT...THEN...END_IF

该条件可使用 NOT 反转 (为 0 时执行这两个语句)。如图 5.4-38 所示。

```

0001 IF NOT FLAG THEN
0002 C:=SIN(A)*COS(B);
0003 B:=C-A;
0004 END_IF;
0005

```

图 5.4-38 示例 2

3. 选择语句 ELSE

(1) 语义

ELSE 语句始终出现在 IF...THEN、ELSIF...THEN 或 CASE 语句后面。

如果 ELSE 语句出现在 IF 或 ELSIF 语句后面,则仅当 IF 和 ELSIF 语句的关联布尔表达式为 0(假)时,才会执行该语句或语句组。如果 IF 或 ELSIF 语句的条件为 1(真),则不会执行该语句或语句组。

如果 ELSE 语句出现在 CASE 后面,则仅当所有范围都不包含选择符的值时,才会执行 ELSE 语句或语句组。如果某个范围包含选择符的值,则不会执行 ELSE 语句或语句组。



- 可以嵌套任何数量的 IF...THEN...ELSE...END_IF 语句,以生成复杂的选择语句。

(2) 示例

```
0001 IF A>B THEN
0002 C:=SIN(A)*COS(B);
0003 B:=C-A;
0004 ELSE
0005 C:=A+B;
0006 B:=C*A;
0007 END_IF;
0008
0009
```

图 5.4-39 示例 3

4. 选择语句 ELSIF...THEN

(1) 语义

ELSE 语句始终出现在 IF...THEN 语句后面。ELSIF 语句确定仅当 IF 语句的关联布尔表达式的值为 0(假)并且 ELSIF 语句的关联布尔表达式的值为 1(真)时,才会执行语句或语句组。如果 IF 语句的条件为 1(真)或者 ELSIF 语句的条件为 0(假),则不会执行该命令或命令组。THEN 关键字标识 ELSIF 条件的结尾和语句的开头。



- 可以嵌套任何数量的 IF...THEN...ELSIF...THEN...END_IF 语句,以生成复杂的选择语句。

(2) 示例

■ ELSIF...THEN

```

0001 IF A>B THEN
0002 C:=SIN (A) * COS (B);
0003 B:=SUB(C, A);
0004 ELSIF A=B THEN
0005 C:=ADD (A, B);
0006 B:=MUL(C, A);
0007 END_IF;
0008
0009

```

图 5.4-40 示例 4

■ 嵌套语句

```

0001 IF A>B THEN
0002 IF B=C THEN
0003 C:=SIN (A) * COS (B);
0004 ELSE
0005 B:=SUB(C, A);
0006 END_IF;
0007 ELSIF A=B THEN
0008 C:=ADD (A, B);
0009 B:=MUL(C, A);
0010 ELSE
0011 C:=DIV (A, B);
0012 END_IF;
0013

```

图 5.4-41 示例 5

5. 选择语句 CASE...OF...END_CASE

(1) 语义

CASE 语句包含一个 INT 数据类型或枚举数据类型（选择符）变量求值的表达式和一个语句组列表。每组都标记可应用的一个或多个整数（INT、DINT、UINT 或 UDINT）或枚举值或整数值的范围。它规定执行第一组语句，其范围之一包含选择符的计算值。否则，将不执行任何语句。

OF 语句指示范围的开头。

所有范围都不包含选择符的值时，才会在 **CASE** 语句内执行 **ELSE** 语句。

END_CASE 关键字标记语句的结尾。

(2) 语句结构

CASE 表达式 **OF**

值 1...

值 2...

ELSE

END_CASE

(3) 示例

```

0001 CASE INT1 OF
0002 1: BOOL1 := TRUE;
0003 2: BOOL2 := TRUE;
0004 ELSE
0005 BOOL1 := FALSE;
0006 BOOL2 := FALSE;
0007 END_CASE;
0008

```

图 5.4-42 示例 6

6. 重复语句 FOR...TO...BY...DO...END_FOR

(1) 语义

FOR 语句用于在发生次数可预先确定的情况下。否则可使用 WHILE 或 REPEAT。

FOR 语句会重复执行语句序列，直到遇到 END_FOR 语句为止。发生次数由起始值、结束值和控制变量决定。

控制变量、起始值和结束值必须具有相同的数据类型（DINT 或 INT），不可被重复语句改变。

FOR 语句以控制变量值为步幅递增或递减起始值，直到达到结束值。增量值的缺省值为 1。如果要使用其他值，则可以指定增幅值（变量或常量）。每个新的循环之前都要检查控制变量值。如果它位于起始值和结束值的范围之外，则将离开循环。

DO 语句标识重复定义的结尾和语句的开头。可以使用 EXIT 提前终止循环。

END_FOR 关键字标记语句的结尾。

(2) 语句结构

FOR (表达式 1) TO (表达式 2) BY (表达式 3) DO

表达式 1: 循环变量的初始化（初始值）

表达式 2: 循环条件（终止值）

表达式 3: 循环变量自增

END_FOR

(3) 示例

```

0001 J:=101;
0002 FOR I:=1 TO 100 BY 2 DO
0003 IF ARR[I] = 70 THEN
0004 J:=I;
0005 EXIT;
0006 END_IF;
0007 END_FOR;
0008

```

图 5.4-43 示例 7

7. 重复语句 WHILE...DO...END_WHILE

(1) 语义

WHILE 语句可使一个语句序列重复执行，直到其相关布尔表达式为 0（假）。如果从一开始该表达式就为假，则根本不会执行该语句组。

DO 语句标识重复定义的结尾和语句的开头。可以使用 EXIT 提前终止循环。

END_WHILE 关键字标记语句的结尾。

下列情况下不应使用 WHILE，因为它可能导致无限循环，从而造成程序崩溃：

- WHILE 不能用于过程之间的同步；

例如，不能用作具有外部定义的结束条件的“等待循环”。

- WHILE 不能用在算法中；

因为无法确保完成循环结束条件或执行 EXIT 语句。

(2) 示例

```
0001 x:= 1;  
0002 WHILE  
0003 x<=100 DO x:=x+4;  
0004 END_WHILE  
0005  
0006  
0007  
0008
```

图 5.4-44 示例 8

8. 重复语句 REPEAT...UNTIL...END_REPEAT

(1) 语义

REPEAT 语句可使一个语句序列重复执行（至少执行一次），直到相关布尔条件为 1（真）。UNTIL 语句标记结束条件。可以使用 EXIT 提前终止循环。END_REPEAT 语句标记语句的结尾。

下列情况下不应使用 REPEAT，因为它可能导致无限循环，从而造成程序崩溃：

- REPEAT 不能用于过程之间的同步；

例如，不能用作具有外部定义的结束条件的“等待循环”。

- REPEAT 不能用在算法中；

因为无法确保完成循环结束条件或执行 EXIT 语句。

(2) 示例

```
0001 X:=-1;  
0002 REPEAT  
0003 X:= X + 2;  
0004 UNTIL X >= 101  
0005 END_REPEAT  
0006  
0007
```

图 5.4-45 示例 9

9. 退出循环语句 EXIT

(1) 语义

EXIT 语句用于在满足结束条件前终止重复语句 (FOR、WHILE 或 REPEAT)。如果 EXIT 语句位于嵌套的重复语句内,则会离开最里面的循环(EXIT 所在的循环)。接下来,将执行循环结尾(END_FOR、END_WHILE 或 END_REPEAT) 后的第一个语句。

(2) 示例

如果 FLAG 的值为 0, 执行语句后 SUM 将为 15。

如果 FLAG 的值为 1, 执行语句后 SUM 将为 6。如图 5.4-46 所示。

```
0001 SUM:= 0;  
0002 FOR I:= 1 TO 3 DO  
0003 FOR J:= 1 TO 2 DO  
0004 IF FLAG=1 THEN EXIT;  
0005 END_IF;  
0006 SUM:= SUM + J;  
0007 END_FOR;  
0008 SUM:= SUM + I;  
0009 END_FOR
```

图 5.4-46 示例 10

10. 函数和功能块控制语句

函数和功能块控制语句由调用功能块机制,以及在物理结束函数或功能块以前把控制返回到调用实体机制组成。

函数 (FUN) 求值应作为表达式求值部分被调用。

功能块 (FB) 应通过语句调用,而语句由功能块实例名称和跟随其后带括号的变元表组成。

在下面的例子中,通过给两个参数 IN 和 PT 赋值来调用一个定时器,然后结果变量 Q 的值赋予变量 OUT。

结果变量被表示为功能块名称后跟一个小点和变量的名字。如图 5.4-47 所示。

```

0001 CMD_TMR(IN:=DM01,PT:=T#300MS);
0002 OUT:=CMD_TMR.Q;
0003

```

图 5.4-47 调用功能块示例

(1) 函数的文本调用特性

函数的文本调用应由函数名称及其后的变元表组成。变元应该用逗号来分隔，而该表的左右两边应该用括号界定。

(2) 函数调用应遵守以下规则：

- 函数输出变量的赋值应是空值或给变量赋值。
- 对 VAR_IN_OUT 变元的赋值应是变量。
- 对 VAR_INPUT 变元的赋值可以是空值、常量、变量或函数调用。在后一种情形，函数结果用作实际变元。

11. 返回语句 RETURN

(1) 语义

提供从函数、功能块或程序中提前退出的功能。

(2) 示例

当 P1 为 TRUE 时，P2 进行自加运算；当 P1 为 FALSE 时，P2 停止自加，退出 IF 语句。

```

0001 IF p1=TRUE
0002 THEN p2:=p2+1
0003 ;
0004 ELSIF p1=FALSE
0005 THEN RETURN
0006 ;
0007 END_IF;

```

图 5.4-48 RETURN 语句示例



- 用户在 ST 编辑器中调用自定义的函数时，可以不连接返回值直接调用，如“FUN1(KKS1);”。

5.4.2.6 调试模式

在离线状态，可勾选【调试模式】。编译下装后，进入在线，可对 ST 语言的 POU 进行在线调试。该功能只对 ST 语言有效。



- 菜单栏：单击【在线】—【调试模式】。

调试模式下，可执行的操作包含：单步执行、设置调试任务、断点对话框、跳出、跳进、断点运行。以上操作在线时可用。【断点对话框】在离线模式下也可以使用，勾选【调试模式】并编译后可操作。

下面以调试 TASK1 中的 STprg 程序为例进行说明。



图 5.4-49 调试模式

1. 设置调试任务

调试前，首先要设置需要调试的任务，在线状态可操作。设置调试任务下拉框中显示所有可调试任务，如图 5.4-50 所示。选择 TASK1 为调试任务。

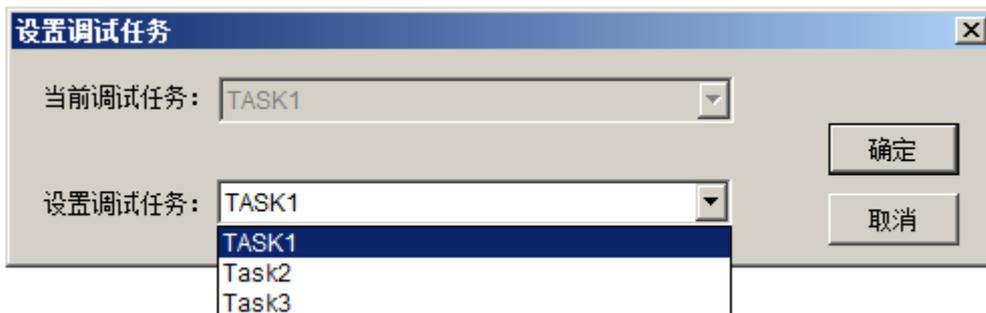


图 5.4-50 选择调试任务

2. 断点设置

为需要调试的 POU 程序设置断点，以方便用户调试。离线或在线状态均可设置。在“断点”对话框中可以添加断点、删除断点、以及设置断点使能等。



- 菜单栏：单击【在线】—【断点对话框】。

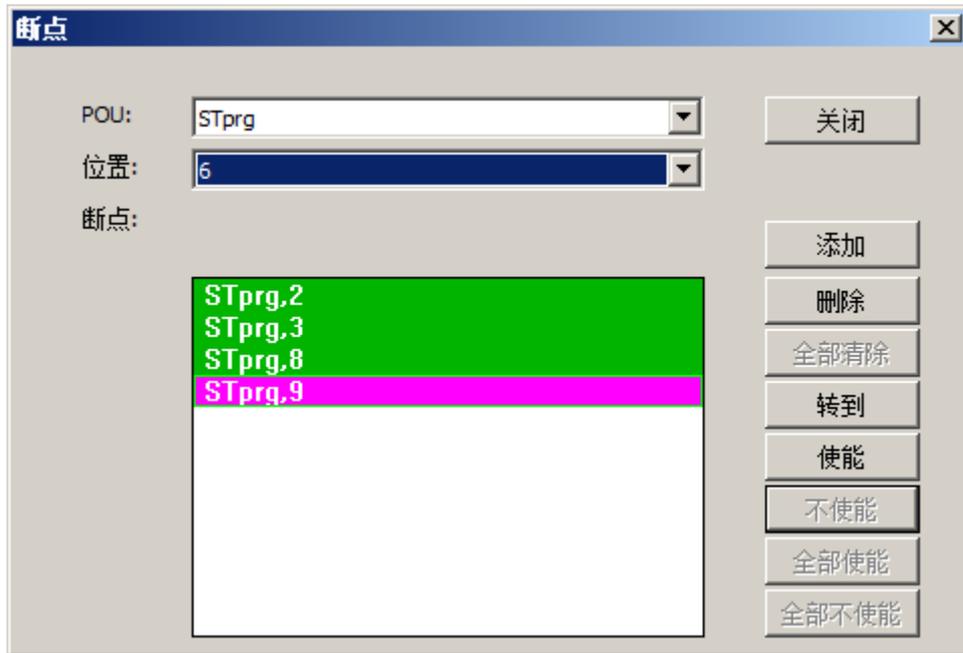


图 5.4-51 断点对话框

- POU：下拉框中显示当前工程中所有 ST 类型的 POU，选择需要调试的 POU 即可。
- 位置：在下拉框中显示已选 POU 下可打断点行的位置，该位置即语句行标号。不可打断点行的位置不显示。
- 断点：在断点列表框中显示所有已添加的使能断点行和非使能断点行。

如图 5.4-52 所示，STprg 中，行号 0007 为不可打断点行，颜色为灰色。可打断点行显示蓝色。行号 0002 为当前执行行，颜色为红色。设置行 2、3、8、9 为 STprg 的断点行，其中，行 2、3、8 为使能断点行，颜色为绿色；行 9 为非使能断点行，颜色为玫红色。

不可打断点行

系统自动识别不可打断点行，无需设置。

可打断点行

除不可打断点行之外的所有语句行均为可打断点行。可打断点行又可设置为使能点行和非使能点行。

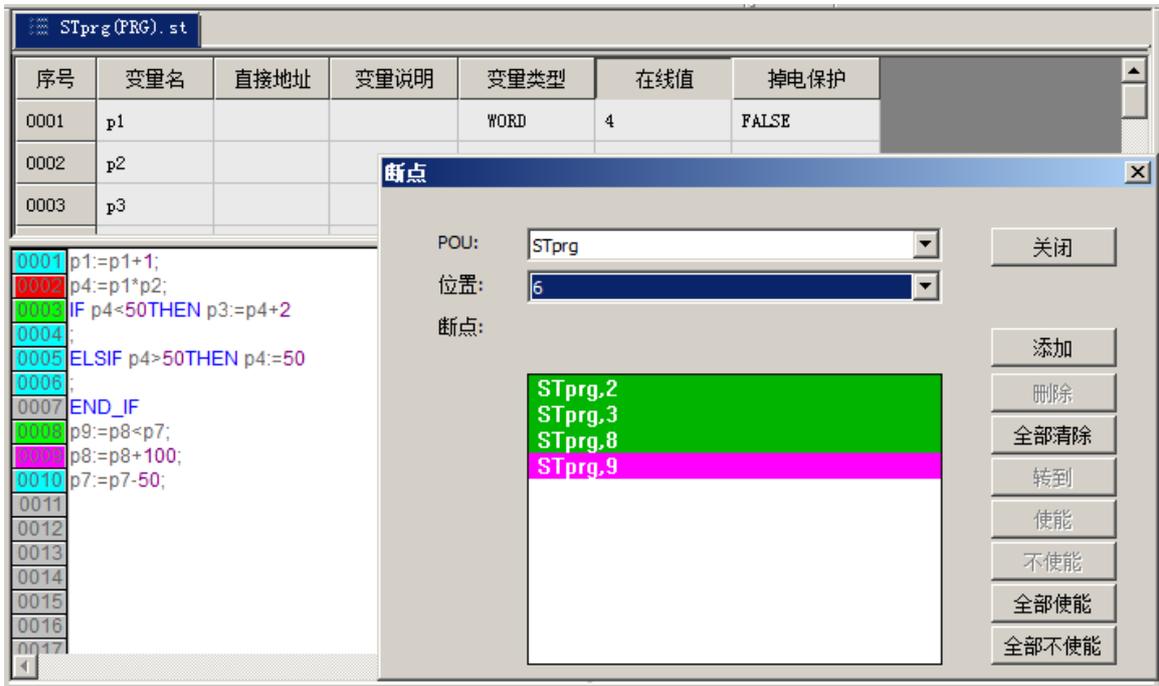


图 5.4-52 添加断点

■ 断点

断点列表框中显示当前工程中所有 ST 类型 POU 中设置的断点。

■ 添加

选择好 POU、断点位置后，单击**添加**，将可打断点行添加到断点列表下。

■ 删除

选中已添加的断点，单击**删除**，从断点列表中删除该断点。

■ 全部清除

单击**全部清除**，将删除断点列表中所有断点。

■ 转到

选中某一断点，单击**转到**，则光标跳转到该断点行位置，并选中当前行。

■ 使能

选中某一非使能断点，单击**使能**，将该断点使能。添加的断点默认使能。

■ 不使能

选中某一使能断点，单击**不使能**，将该断点取消使能。不使能的断点不作为一个断点行，执行【断点运行】时，不会执行到该行。

- 全部使能

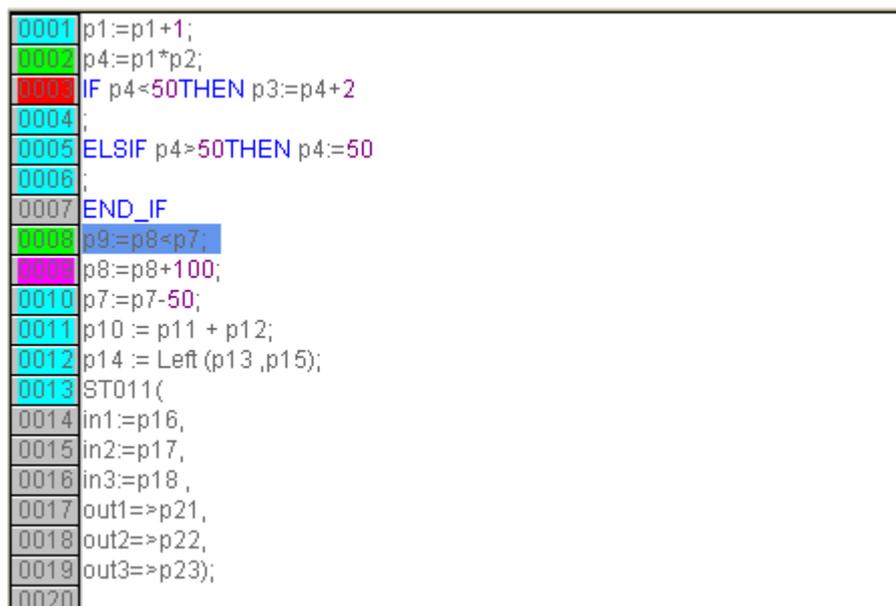
对所有添加的断点全部使能。

- 全部不使能

对所有添加的断点都不使能。

在线状态下，也可通过单击可打断点行的行号，将该行设置为使能断点行。也可通过单击断点行取消该行的断点设置。同时，“断点”对话框中同步更新使能断点行的设置。

如图 5.4-53 (a) 所示，单击使能断点行的行号 0008，取消该行的使能断点设置，如图 5.4-53 (b) 所示。



```
0001 p1:=p1+1;
0002 p4:=p1*p2;
0003 IF p4<=50THEN p3:=p4+2
0004 ;
0005 ELSIF p4>50THEN p4:=50
0006 ;
0007 END_IF
0008 p9:=p8<p7;
0009 p8:=p8+100;
0010 p7:=p7-50;
0011 p10 := p11 + p12;
0012 p14 := Left (p13 ,p15);
0013 ST011(
0014 in1:=p16,
0015 in2:=p17,
0016 in3:=p18 ,
0017 out1=>p21,
0018 out2=>p22,
0019 out3=>p23);
0020
```

(a)

```

0001 p1:=p1+1;
0002 p4:=p1*p2;
0003 IF p4<50THEN p3:=p4+2
0004 ;
0005 ELSIF p4>50THEN p4:=50
0006 ;
0007 END_IF
0008 p9:=p8<p7;
0009 p8:=p8+100;
0010 p7:=p7-50;
0011 p10 := p11 + p12;
0012 p14 := Left (p13 ,p15);
0013 ST011(
0014 in1:=p16,
0015 in2:=p17,
0016 in3:=p18 ,
0017 out1=>p21,
0018 out2=>p22,
0019 out3=>p23);

```

(b)

图 5.4-53 单击行号添加使能断点

3. 单步执行

ST 程序以可打断点行为步，执行程序语句。单击【单步执行】，程序执行完当前行，同时跳转到下一行。

如果当前调试任务为单次任务，在 pou 最后一步执行两次【单步执行】，本次任务停止；如果当前调试任务为周期任务，在 pou 最后一行执行两次【单步执行】，则跳转到本 pou 的第一步。

【单步执行】以语句末尾的分号为步单位执行用户程序。



- 菜单栏：单击【在线】—【单步执行】；
- 工具栏：；
- 快捷键：**F10**。

在图 5.4-54 中，当前执行行为 0003，需要执行两次【单步执行】，即 0004 行的分号被执行后，IF 语句才被执行，同时跳转到 0008 行。第一次单步执行结果如图 5.4-54 (b) 所示。第二次单步执行结果如图 5.4-54 (c) 所示，IF 语句被执行，P3 被赋值 12。

0001	p1:=p1+1;	p1 = 5		
0002	p4:=p1*p2;	p4 = 10		p2 = 2
0003	IF p4<50THEN p3:=p4+2	p4 = 10	p1 = 5	p3 = 10
0004	;			
0005	ELSIF p4>50THEN p4:=50	p4 = 10		
0006	;			
0007	END_IF			
0008	p9:=p8<p7;	p9 = TRUE	p8 = 90	p7 = 100
0009	p8:=p8+100;	p8 = 90		
0010	p7:=p7-50;	p7 = 100		
0011				
0012				

(a)

0001	p1:=p1+1;	p1 = 5		
0002	p4:=p1*p2;	p4 = 10		p2 = 2
0003	IF p4<50THEN p3:=p4+2	p4 = 10	p1 = 5	p3 = 10
0004	;			
0005	ELSIF p4>50THEN p4:=50	p4 = 10		
0006	;			
0007	END_IF			
0008	p9:=p8<p7;	p9 = TRUE	p8 = 90	p7 = 100
0009	p8:=p8+100;	p8 = 90		
0010	p7:=p7-50;	p7 = 100		
0011				
0012				

(b)

0001	p1:=p1+1;	p1 = 5		
0002	p4:=p1*p2;	p4 = 10		p2 = 2
0003	IF p4<50THEN p3:=p4+2	p4 = 10	p1 = 5	p3 = 12
0004	;			
0005	ELSIF p4>50THEN p4:=50	p4 = 10		
0006	;			
0007	END_IF			
0008	p9:=p8<p7;	p9 = TRUE	p8 = 90	p7 = 100
0009	p8:=p8+100;	p8 = 90		
0010	p7:=p7-50;	p7 = 100		
0011				
0012				

(c)

图 5.4-54 单步执行示例

如果调用了函数、功能块或其他 ST 程序，执行【单步执行】操作，则会进入被调用程序内部的断点行。

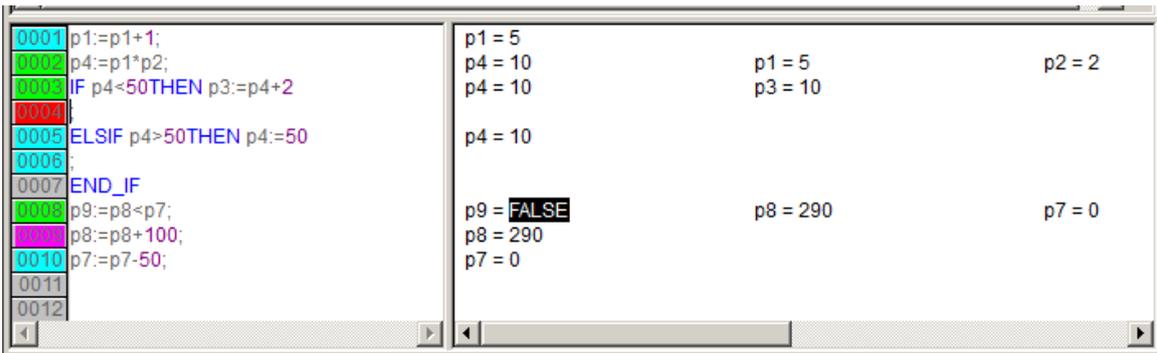
4. 断点运行

以设置的使能断点为步，执行程序语句。执行【断点运行】，程序从当前行开始执行，直到遇到下一个使能断点，才停止运行。再次执行【断点运行】，则从当前使能断点行开始执行，直到下一个使能断点停止运行。

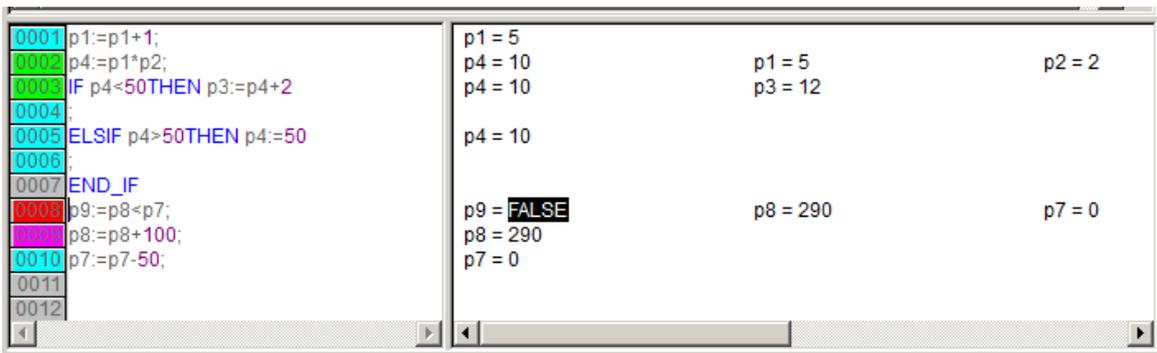


- 菜单栏：单击【在线】—【断点运行】；
- 工具栏：；
- 快捷键：**Ctrl+F5**。

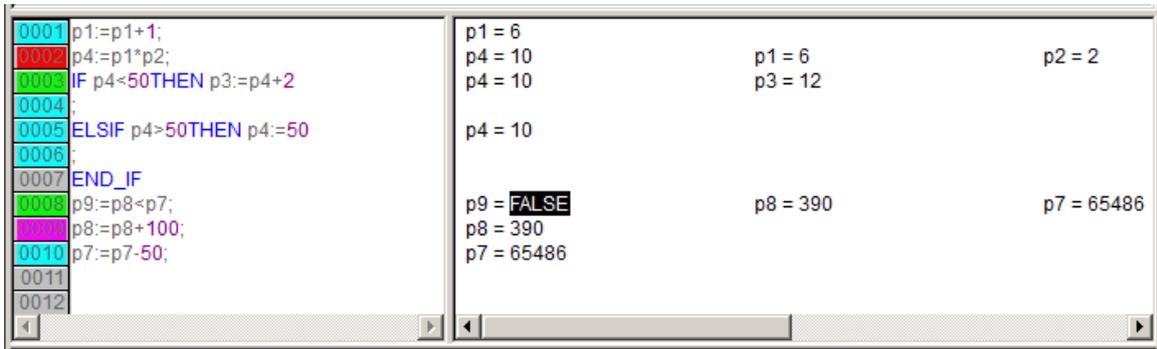
在图 5.4-55 中，当前执行行为 0004，执行【断点运行】，程序从该行开始执行，直到遇到使能断点行 0008 才停止执行，如图 5.4-55 (b) 所示。再次执行【断点运行】，程序从使能断点行 0008 开始执行，直到下一个使能断点行 0002 停止，如图 5.4-55 (c) 所示。非使能断点行 0009 不作为断点运行的步，处理方式与一般可打断点行一致。



(a)



(b)



(c)

图 5.4-55 断点运行示例

若是调用了函数、功能块或其他 ST 程序，即使该调用行没有设置使能断点，但是函数、功能块、或 ST 程序内部设置有使能断点行，则执行【断点运行】时，会跳入到相应的使能断点处。

5. 跳进

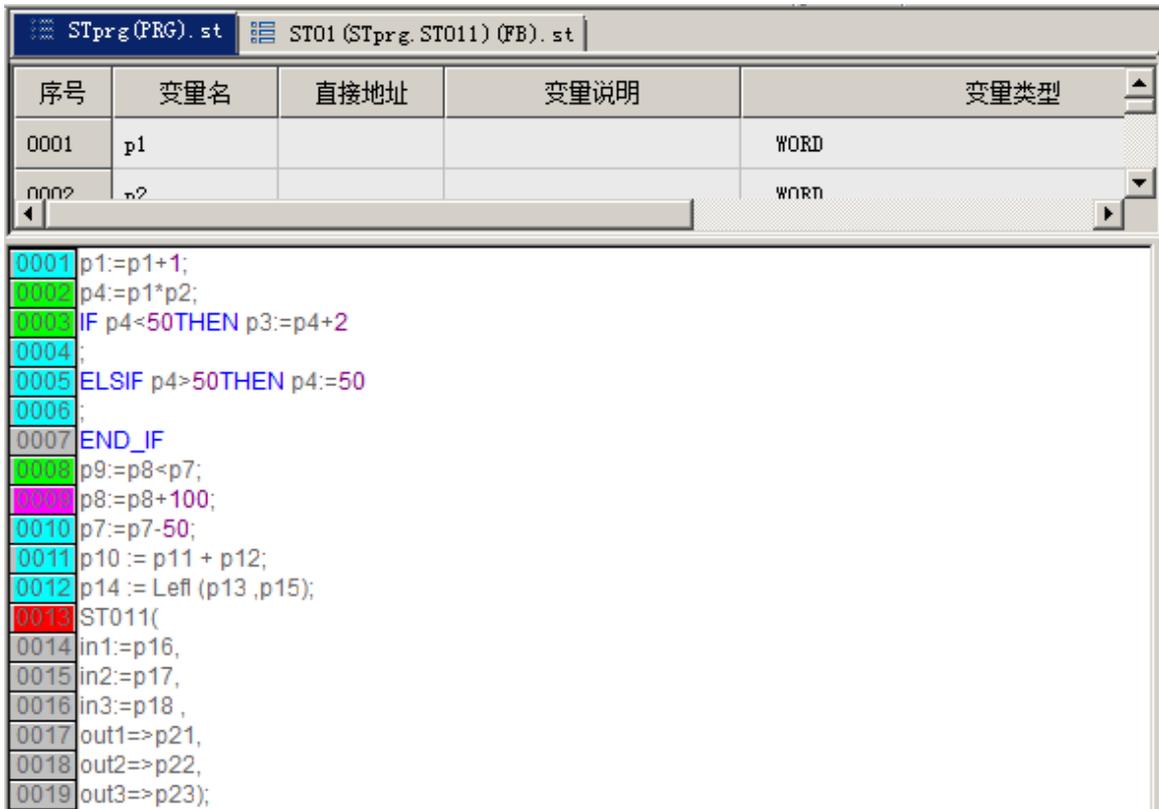
执行该命令，当前行被执行，同时跳到下一行。若当前行是函数、功能块或已调用的其他 ST 程序，则进入该函数、功能块或程序逻辑内部，待执行语句为第一行可执行的语句。

对于普通语句行（非函数、功能块或已调用的其他 ST 程序），【跳进】与【单步执行】的作用一致。



- 菜单栏：单击【在线】—【跳进】；
- 工具栏：
 
- 快捷键：Ctrl+F11。

程序 STprg 中调用功能块 ST011，并设置 ST011 的位置 1、2 为使能断点行。当前执行行为 0013，如图 5.4-56 (a) 所示。执行【跳进】操作，程序进入被调用的功能块内部，如图 5.4-56 (b) 所示。在功能块最后一步执行【单步执行】，则程序执行该行，并返回到图 5.4-56 (a) 所示的当前功能块调用位置。



(a)



(b)

图 5.4-56 跳进示例

6. 跳出

在函数、功能块或已调用的其他 ST 程序内部的最后一个使能断点行，执行该命令，则该函数、功能块或程序执行完并返回到调用处。

对于普通语句行（非函数、功能块或已调用的其他 ST 程序），【跳出】与【断点运行】的作用一致。

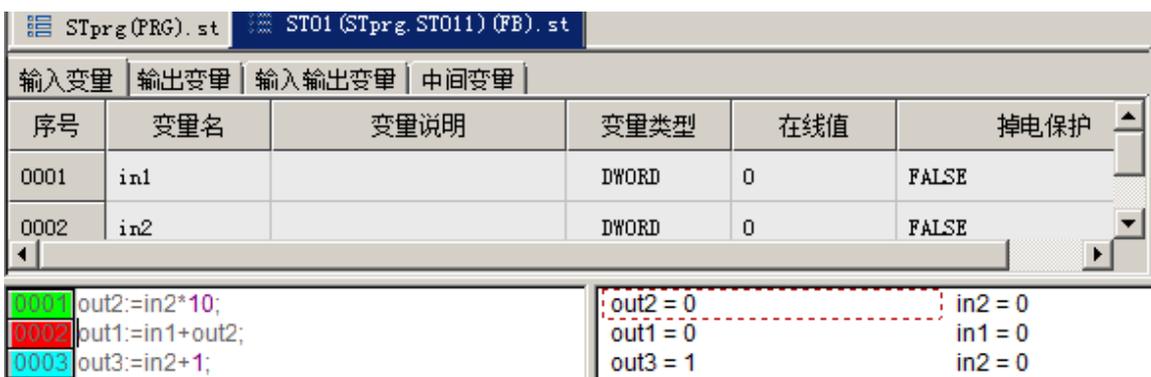


- 菜单栏：单击【在线】—【跳出】；
- 工具栏：；
- 快捷键：**Ctrl+Shift+F8**。

程序 STprg 中调用功能块 ST011，并设置 ST011 的位置 1、2 为使能断点行。功能块 ST011 内部当前执行行为 0001，如图 5.4-57 (a) 所示，执行【跳出】命令，程序执行完 0001 行并跳转到下一个使能断点行 0002，如图 5.4-57 (b) 所示。再次执行【跳出】命令，程序返回到功能块调用处，如图 5.4-57 (c) 所示。



(a)



(b)

```
0001 p1:=p1+1;
0002 p4:=p1*p2;
0003 IF p4<50THEN p3:=p4+2
0004 ;
0005 ELSIF p4>50THEN p4:=50
0006 ;
0007 END_IF
0008 p9:=p8<p7;
0009 p8:=p8+100;
0010 p7:=p7-50;
0011 p10 := p11 + p12;
0012 p14 := Left (p13 ,p15);
0013 ST011(
0014 in1:=p16,
0015 in2:=p17,
0016 in3:=p18 ,
0017 out1=>p21,
0018 out2=>p22,
0019 out3=>p23);
```

(c)

图 5.4-57 跳出示例

5.4.3 CFC 编辑器

CFC 是连续功能图 (Continuous Function Chart)的简称。是一种可视化图形编辑语言，具有操作方便、显示直观的特点，适合用于连续过程控制的组态。

如图 5.4-58 所示，CFC 编辑器是一种图形编辑器，通过菜单栏、工具栏或右键菜单命令进行编辑。

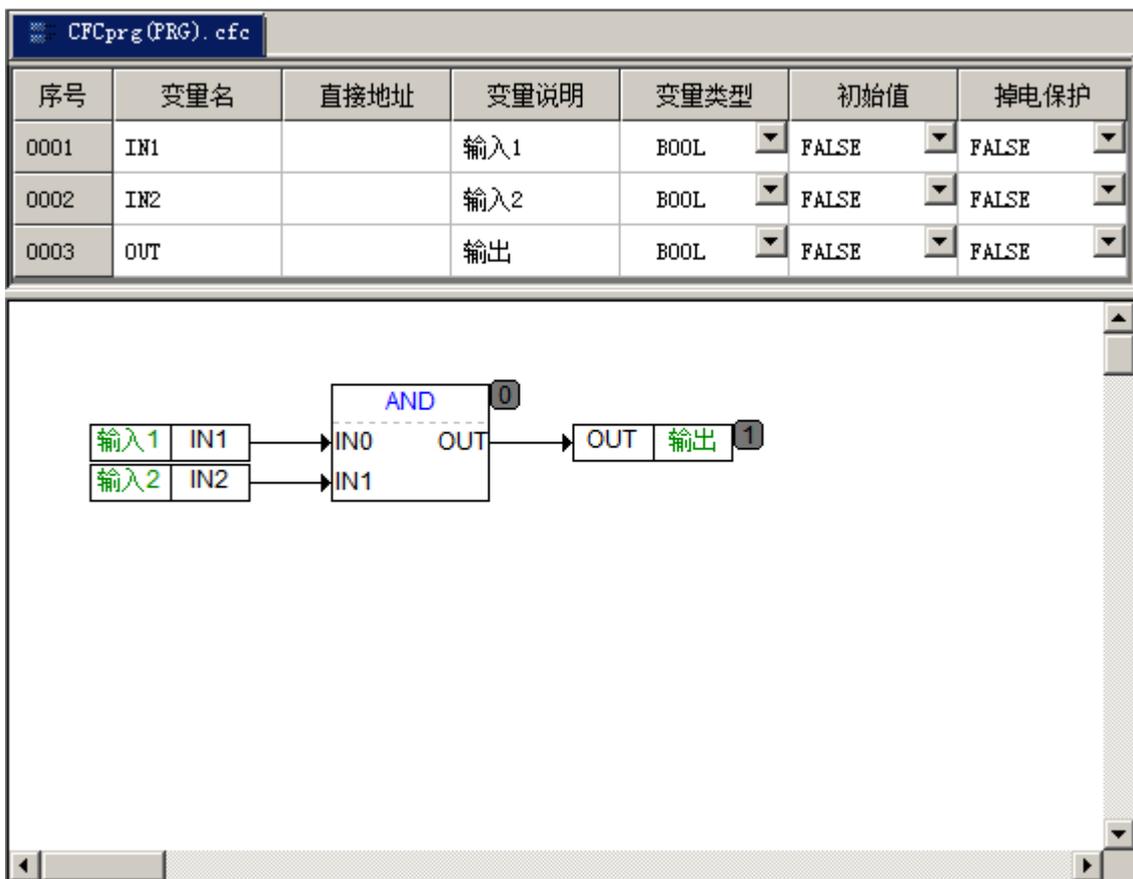


图 5.4-58 “CFC” 编程窗口

5.4.3.1 光标位置

每个文本都是光标可能的位置，选中的文本渐变为蓝色并且可以被修改。下面是光标可能位置的示例。

- (1) 选中输入、输出、跳转、标签、返回、块元件和注释元件

单击文本区域，即可选中，如图 5.4-59 所示。

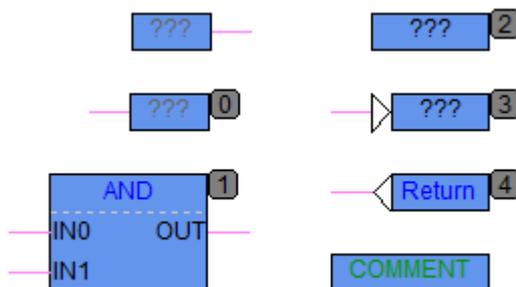


图 5.4-59 选中元件

(2) 选中输入、输出、跳转、标签、返回、块元件和注释元件。

双击文本区域，即可选中，如图 5.4-60 所示。

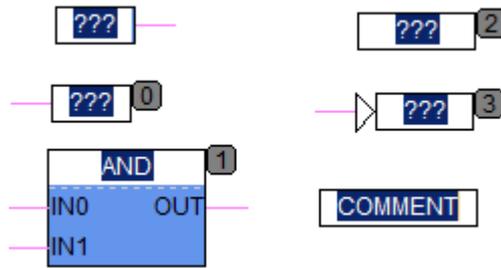


图 5.4-60 选中元件的文本区域

(3) 选中输出、跳转、块元件、返回元件的输入端

单击输入端，即可选中（呈蓝色），一次只能选中一个。如图 5.4-61 所示。

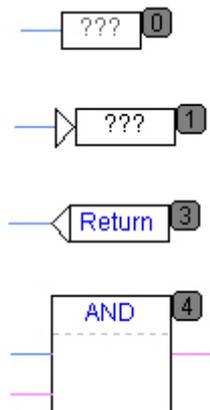


图 5.4-61 选中输入端

(4) 选中输入元件、块元件的输出端

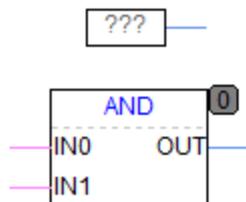


图 5.4-62 选中输出端

元件选中状态标识的颜色可以在【工程】—【选项】—【颜色】中进行修改。

5.4.3.2 输入元件



- 菜单栏：单击【插入】—【输入元件】；
- 程序区：右击空白区域，单击【输入元件】；
- 工具栏：；
- 快捷键：**Ctrl+I**。

这个命令用来插入一个输入，出现的文本“???”可以被选中并被变量或常量替代。

新插入的输入元件随鼠标移动，移动到合适的位置单击，插入成功。如图 5.4-63 所示。

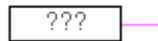


图 5.4-63 输入元件

5.4.3.3 输出元件



- 菜单栏：单击【插入】—【输出元件】；
- 程序区：右击空白区域，单击【输出元件】；
- 工具栏：；
- 快捷键：**Ctrl+U**。

这个命令用来插入一个输出，出现的文本“???”可以被选中并被变量替代。

新插入的输出元件随鼠标移动，移动到合适的位置单击，插入成功。如图 5.4-64 所示。

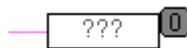


图 5.4-64 输出元件

输出元件只有一个输入引脚，用来接收其他元件输出引脚的数据。输出元件关联一个执行序号，用来标示执行顺序。



- 输出元件右上方的序号①，初始添加时表示添加顺序，之后可进行按数据流排序。详细内容请参章节 5.4.3.18 调整元件执行顺序。

5.4.3.4 块元件

通过以下方式插入块元件。



- 菜单栏：单击【插入】—【块元件】；
- 程序区：右击空白区域，单击【块元件】；
- 工具栏：；
- 快捷键：**Ctrl+B**。

在程序编辑区插入块元件如图 5.4-65 所示。

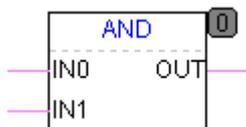


图 5.4-65 块元件

块包括操作符、函数、功能块和程序。本身包含 BOOL 类型的使能端 EN 和 ENO，且 EN 和 ENO 处于相同状态。如果块处于隐藏 EN 和 ENO 端状态，则 EN 和 ENO 状态为 TRUE；否则视输入状态而定。

(1) 操作符

库中定义并提供给用户使用的块，主要完成算术运算、赋值运算、逻辑运算、移位、选择、比较、数据类型转换、初等数学运算等等功能。操作符块基本是一些常用功能，直接提供给用户用来组态复杂程序逻辑。

一个位运算操作符块如图 5.4-66 所示。

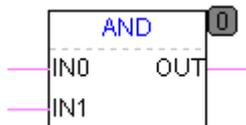


图 5.4-66 运算操作符块

(2) 函数

完成一定功能的函数，以块的形式参与 CFC 组态（供 CFC 用户调用）。函数块可以是库提供也可以是用户用 IEC 编程语言自定义的。

函数存在若干输入变量和唯一返回值，一个典型函数块 FUNC 如图 5.4-67 所示。



图 5.4-67 函数块

(3) 功能块

完成一定功能，以块的形式参与 CFC 组态（供 CFC 用户调用）。功能块可以是标准库提供也可以是用户用 IEC 编程语言自定义的。

功能块存在若干输入和若干输出，一个典型功能块 FB 如图 5.4-68 所示。

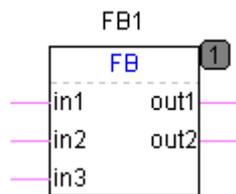


图 5.4-68 功能块

(4) 程序

程序块是完成一定功能的独立单元，以块的形式参与 CFC 组态（供 CFC 用户调用），能够直接被程序调用。程序块存在一个输入和一个使能输出，一个典型程序块 PRG1 如图 5.4-69 所示。



图 5.4-69 程序块

1. 多输入

使用此命令增加块元件的输入端。

选中块元件执行以下操作：



- 菜单栏：单击【插入】—【多输入】；
- 程序区：右击块元件，单击【高级】—【多输入】；
- 工具栏：；
- 快捷键：**Ctrl+D**。

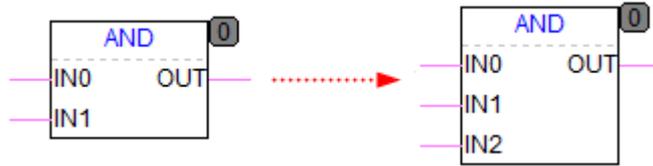


图 5.4-70 执行多输入命令

选中已添加的输入端，执行【删除】命令，即可删除引脚。

2. 修改块元件类型

用户双击块元件类型的文本区域，弹出编辑框，输入新的块元件类型。如图 5.4-71 所示。

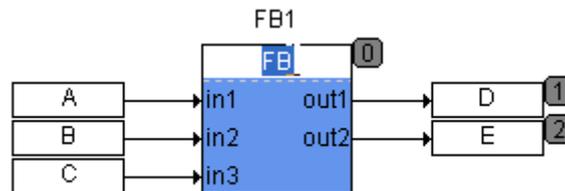


图 5.4-71 双击块类型文本，弹出编辑框

3. 块元件使能

这个命令用来显示、隐藏选中块元件的使能输入端和使能输出端。如图 5.4-72 所示，重复激活这个命令可以使块元件在使能、正常状态之间转换。



- CFC 程序默认情况下隐藏块元件使能端。

选中块元件，通过以下方式执行【使能】命令。



- 菜单栏：单击【插入】—【使能】；
- 程序区：右击块元件，单击【高级】—【使能】；
- 工具栏：
 ；
- 快捷键：**Ctrl+E**。

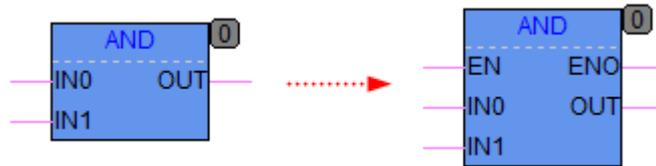


图 5.4-72 显示使能端

4. 设置引脚属性

【设置引脚属性】选项设置块元件的输入、输出引脚的显示与隐藏。



- 程序区：右击块元件，单击【高级】—【设置引脚属性】。

详细内容请参见章节 5.4.1.10 块元件的【高级】选项设置。

5.4.3.5 标签元件

该元件的功能是标示程序的执行位置信息，是跳转指令跳转的目的地。标号指令不存在任何输入和输出引脚。



- 菜单栏：单击【插入】—【标签元件】；
- 程序区：右击空白区域，单击【标签元件】；
- 工具栏：
 ；
- 快捷键：**Ctrl+L**。

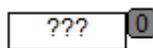


图 5.4-73 标签元件

文本域“???”中输入标号的标签名。该名称必须与跳转名一致。

5.4.3.6 跳转元件

该元件可以实现程序的转移执行。当执行该指令时跳转到该指令引用的标签处。跳转指令存在唯一的 BOOL 值输入引脚。



- 菜单栏：单击【插入】—【跳转元件】；
- 程序区：右击空白区域，单击【跳转元件】；
- 工具栏：；
- 快捷键：**Ctrl+J**。

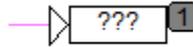


图 5.4-74 跳转元件

文本域“???”中输入要跳转到的标号名称，该名称必须与标签定义的名称一致。

跳转条件必须为 **BIT** 或 **BOOL** 类型（包括可以转换成 **BOOL** 的类型）的变量。

如图 5.4-75 所示，当跳转条件满足，CFC 程序将直接跳转到标签所在的红色标记区域，按照信号流方向执行程序。跳转标签之前的信号流②、③所在的程序行不执行。

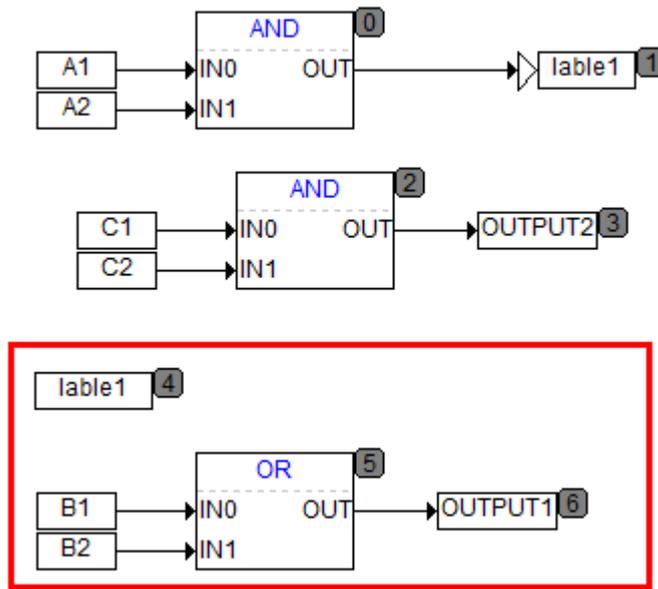


图 5.4-75 跳转与标签指令举例

5.4.3.7 返回元件

执行该指令后当前 POU 直接返回到调用该 POU 程序处。返回指令存在唯一输入引脚，输入值为 BOOL 类型。



- 菜单栏：单击【插入】—【返回元件】；
- 程序区：右击空白区域，单击【返回元件】；
- 工具栏：；
- 快捷键：**Ctrl+R**。



图 5.4-76 返回元件

5.4.3.8 注释元件

注释元件不参与程序的执行，其作用就是在 CFC 程序界面插入，表示说明的文本信息。相当于其他语言中的注释信息，该指令不存在任何输入输出信息。



- 菜单栏：单击【插入】—【注释元件】；
- 程序区：右击空白区域，单击【注释元件】；
- 工具栏：；
- 快捷键：**Ctrl+K**。

COMMENT

图 5.4-77 注释元件

5.4.3.9 删除元件

该命令用于删除一个或多个 CFC 元件。



- 程序区：右击块元件，单击【删除】；
- 快捷键：**Delete**。

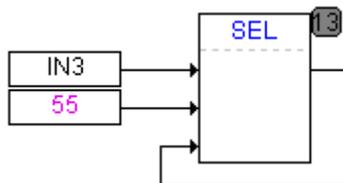
删除元件时，连同连线一起被删除。

5.4.3.10 增加连线

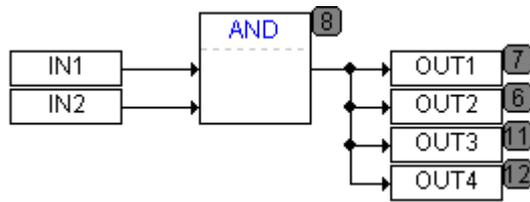
通过连线将块元件的引脚连接起来，组态用户逻辑。

拖动鼠标，连接元件的输入和输出，会自动创建连线。当元件移动的时候，连线线也会自动调整。

一个元件的输入引脚只能连一个变量（本元件的输出引脚或其他元件的输出引脚）如图 5.4-78 中 (a) 所示，而一个元件的输出引脚可以连多个变量（本元件的输入变量或其他元件的输入变量），如图 5.4-78 (b) 所示。



(a)

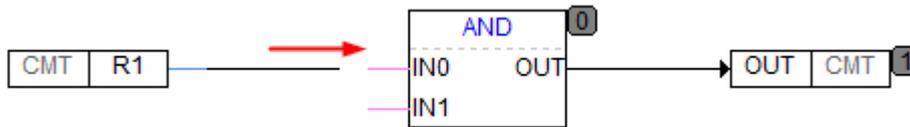


(b)

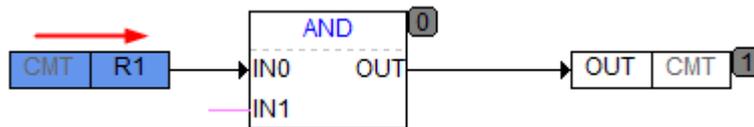
图 5.4-78 引脚信号连线方式

有两种方式连接 R1、AND 和 OUT 元件之间的连线。

- 单击 R1 的输出引脚，拖拽到 AND 的输入引脚上，释放左键，如图 5.4-79 (a) 所示。
- 拖动 R1，将其输出引脚线与 AND 的输入引脚线重合，释放左键。如图 5.4-79 (b) 所示。



(a)



(b)

图 5.4-79 连线

5.4.3.11 删除连线

若要删除连线，可以选中引脚然后执行右键【删除】或键盘 **Delete**，连线即可被删除。

删除连线两种方式，如图 5.4-80 所示。选中 A 的输出引脚执行【删除】命令，将删除 A 输出引脚所连的全部连线。选中 C 出入引脚，执行【删除】命令，只删除 C 输入引脚与 A 输出引脚间的连线，A 与 B 之间的连线不受影响。

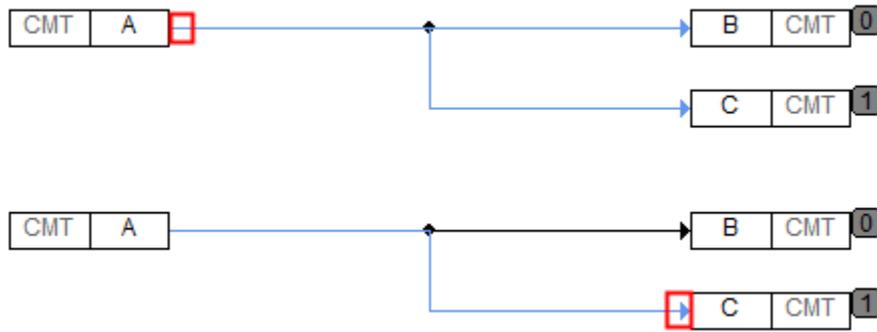


图 5.4-80 删除连线的选中位置

5.4.3.12 修改元件文本

每一个 CFC 元件文本都占据一定区域，双击文本区域就会弹出编辑元件文本的编辑框。通过编辑框用户可以修改元件的文本。举例，如图 5.4-81 所示的 CFC 元件，现在要修改功能块文本 FB1。

双击功能块文本 FB1，在原来文本处弹出编辑框如图 5.4-81 所示，用户即可在编辑框内输入新的文本。

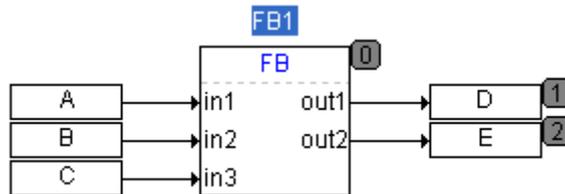


图 5.4-81 文本编辑

5.4.3.13 输出元件的置位复位

CFC 输出元件具有置位复位功能。被置位的输出元件可被复位，复位的输出元件能被还原。

选中输出元件，连续执行以下操作，输出元件在置位、复位、还原状态之间转换。如图 5.4-82 所示。



- 菜单栏：单击【插入】—【置位/复位】；
- 程序区：右击输出元件，单击【高级】—【置位/复位】；
- 工具栏：；
- 快捷键：**Ctrl+T**。

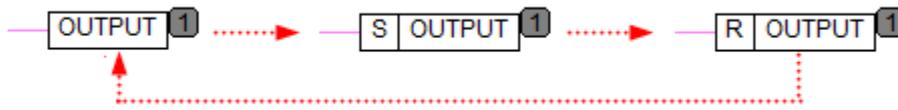


图 5.4-82 连续执行置位复位操作

如图 5.4-83 所示，对于置位输出，AND 块元件的 OUT 为 TRUE，则 OUTPUT1 被置位为 TRUE，并保持置位状态不变，即使 OUT 变为 FALSE。只有当输入元件 P2 变为 FALSE，OUTPUT1 才被复位为 FALSE，并且保持复位状态不变，直到置位条件满足，再次被置位。

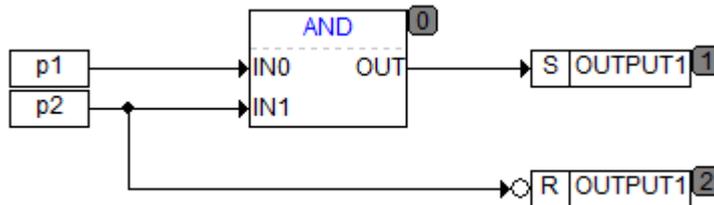


图 5.4-83 置位复位示例

5.4.3.14 引脚的取反和还原

置反操作是对 BOOL 值进行取反操作。选中引脚，执行【置反】命令，则被选中的引脚被置反。再次选中被取反的引脚，执行【置反】命令后，选中的引脚被还原。如图 5.4-84 所示。



- 菜单栏：单击【插入】—【置反】；
- 程序区：右击元件的引脚，单击【置反】；
- 工具栏：；
- 快捷键：**Ctrl+G**。



图 5.4-84 执行置反操作

如图 5.4-85 所示，若 p2 为 TRUE，则经过取反操作后，OUTPUT1 的值为 FALSE。若 p2 为 FALSE，则 OUTPUT1 的值为 TRUE。

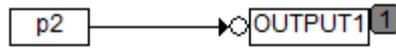
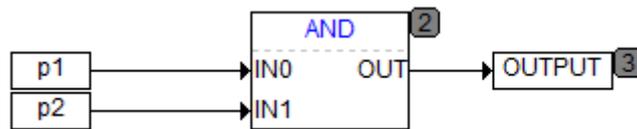


图 5.4-85 置反操作示例

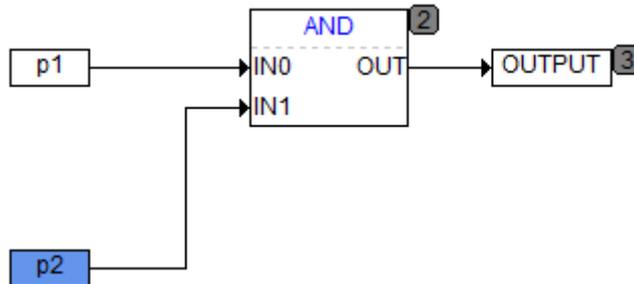
5.4.3.15 移动元件

CFC 中可在编辑区自由拖动元件。选中元件拖动，可实现元件的移动。如果选项中开启默认碰撞检测，即移动到的新位置已存在元件，则移动失败，恢复到移动前状态。可以自由拖动元件到目标位置。

如图 5.4-86 所示，选中元件 P2，拖动到一个新位置，实现元件的移动。



(a)



(b)

图 5.4-86 移动元件

5.4.3.16 剪切、复制和粘贴

复制：选中要复制的元件，执行复制命令或键盘指令 **Ctrl+C**，即可把选中元件数据放到剪贴板上。如果选择多个元件，并且元件间存在连线，则保持原来连线。

剪切：选中要剪切的元件，执行剪切命令即可把选中元件数据放到剪贴板上，同时选中的元件被删除。

粘贴：将已复制或剪切在剪贴板上元件数据以元件形式添加到 CFC 编辑界面中。

通过以下方式执行复制/剪切操作。



- 程序区：右击元件，单击【复制】/【剪切】。

执行以上操作后，进行粘贴操作。



- 程序区：右击空白，单击【粘贴】。

5.4.3.17 撤销和恢复

撤销是取消上一次命令的执行，恢复到上一次执行操作命令之前。

恢复命令将操作恢复到执行撤销命令前。

所有 CFC 程序组态操作命令分为两类：不可以撤销恢复操作和可以撤销恢复操作。这里不包括对 POU 本身的操作比如 POU 重命名、禁止调用和允许调用等等。



- 不可以撤销恢复命令包括：选中元件、选中输入端、选中输出端、复制元件、查找。

5.4.3.18 调整元件执行顺序

CFC 元件除了输入元件和注释都存在一个标示程序执行顺序的数字，显示在元件的右上角。

CFC 程序按照各个元件的执行序号，从 0 号开始，依次完成赋值、计算、跳转、返回等程序流程。执行顺序编号默认按照元件输入顺序排序。元件执行序号在组态过程中始终保持从小到大并按自然数连续排列。



图 5.4-87 执行顺序选项

(1) 显示执行顺序

此命令用于显示隐藏元件的执行顺序。如图 5.4-88 所示。

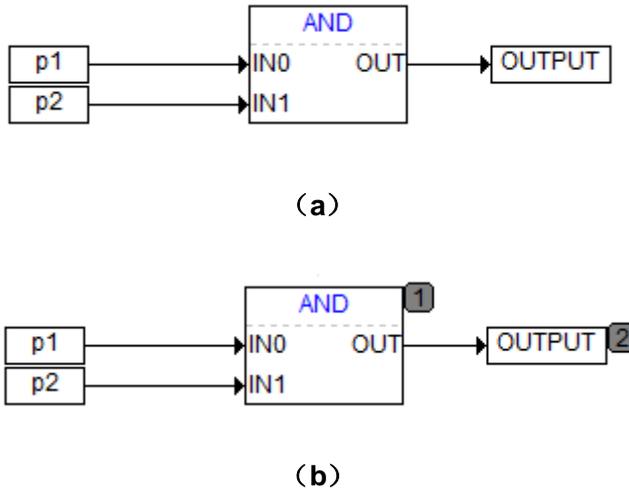


图 5.4-88 显示隐藏执行顺序

(2) 前移一位

选中带有执行顺序的元件，执行该命令，选中元件与执行编号在其前面一位的元件进行执行编号交换。选中最小编号的元件执行该命令后执行顺序保持不变。

如图 5.4-89 所示，对输出元件 c 执行【前移一位】命令，其执行编号与其前一位执行编号互换。

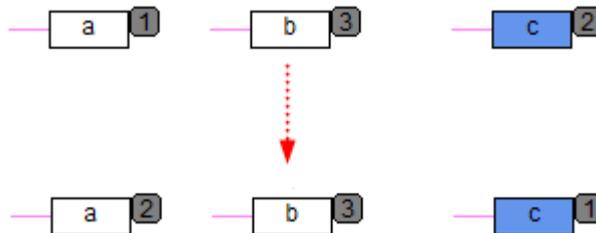


图 5.4-89 执行前移一位命令

(3) 后移一位

选中带有执行顺序的元件，执行该命令，选中元件与执行编号在其后一位的元件进行执行编号交换。选中最大编号的元件执行该命令后执行顺序保持不变。

如图 5.4-90 所示，对输出元件 c 执行【后移一位】命令，其执行编号与其后一位执行编号互换。

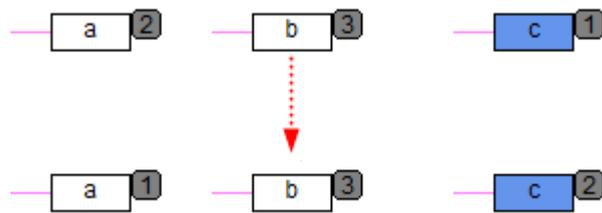


图 5.4-90 执行后移一位命令

(4) 移至首

将选中元件编号设置为 0，原元件执行编号前面的带执行编号元件的编号分别加 1。选中元件编号为 0 则执行该命令后无影响。

如图 5.4-91 所示，对输出元件 b 执行【移至首】，其执行编号变为最小编号 0，其原执行编号前的编号 2、1 分别加 1。

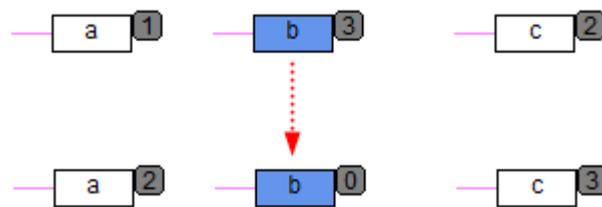


图 5.4-91 执行移至首命令

(5) 移至尾

将选中元件编号设置为现有编号范围中的最大值，原来元件执行编号后面的带执行编号元件的编号分别减 1。选中元件编号为最大则执行该命令后无影响。

如图 5.4-92 所示，对输出元件 b 执行【移至尾】命令，其执行编号变为最大执行编号，其原执行编号后的编号 2、3 分别减 1。

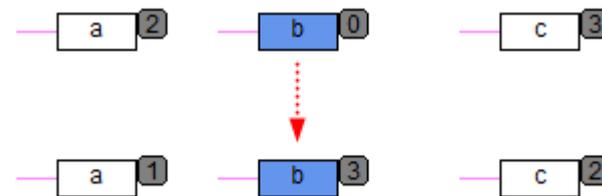


图 5.4-92 执行移至尾命令

(6) 按数据流排序

按数据流排序原则：以输入为基础，执行编号按照逆波兰式顺序排序。

举例：如图 5.4-93 所示，按照数据流排序的 CFC 程序。

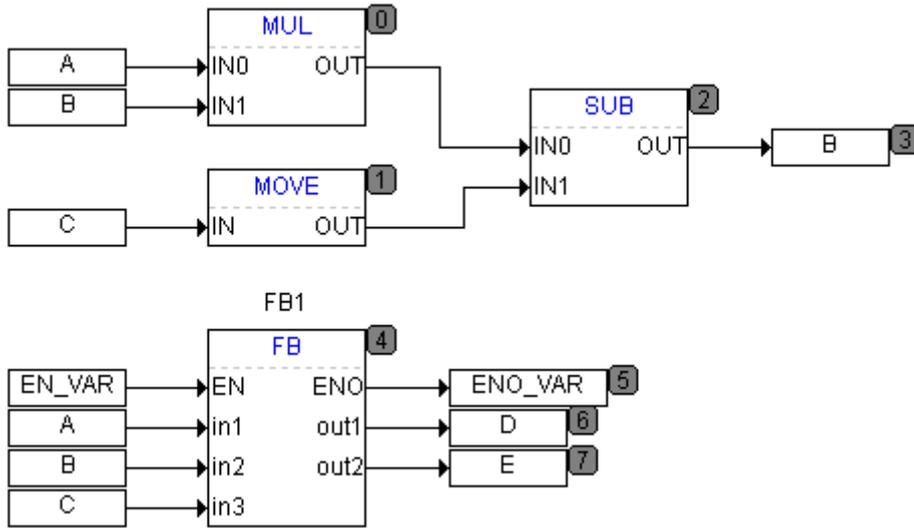


图 5.4-93 按数据流排序

5.4.3.19 元件的对齐

对块元件进行水平方向上的左右对齐，以及垂直方向上的间距最小距离排布。

选中两个以上块元件，执行对齐操作。



- 程序区：右击选中的元件，单击【元件对齐】—【水平左对齐】/【水平右对齐】/【垂直对齐】。



图 5.4-94 元件的对齐

水平左/右对齐：以选中元件中最左/右边位置的元件为基准，选中的其他元件在水平方向上与基准对齐。

垂直对齐：以选中元件中最上边位置的元件为基准，选中的其他元件在垂直方向上相互间隔最小距离排布，水平方向不做移动。

5.4.3.20 变量区（显示/隐藏）

此命令，用来显示或隐藏变量区。

通过以下方式执行该命令。



- 程序区：右击空白区域，单击【变量区（显示/隐藏）】。

5.4.4 SFC 编辑器

SFC 是顺序功能图（Sequential Function Chart）的简称，是 IEC61131-3 标准中的一种编程语言，用于建立和修改顺序控制，如图 5.4-95 所示。SFC 能使单任务到复杂任务用一种简单的配置方法完成。这个结构用独立的元素表示用户的子程序，类似网络节点元素。

子任务是关联到转换和步的，通过程序的形式体现的。这些程序可以使用 CFC、LD、ST 语言编辑。转换表示激活后继步的使能条件。步周期地处理直到下一个转换条件满足。

转换通过线、分支和控制过程的独立元素连接起来。选择分支和并行分支有区别。如果是选择分支，一次只能是一个过程在运行；反之，过程同时有分支时，几个步并行处理。

当 POU 文件设置为分页打印时，每个用 SFC 语言编辑的 POU 的工作区由若干个页面组成。通过水平和垂直的滚动条可以接触到每个独立的页面。一个隐形的栅格覆盖了整个工作区。一个 POU 文件打印输出的每一页内容和独立页的是完全对应的。

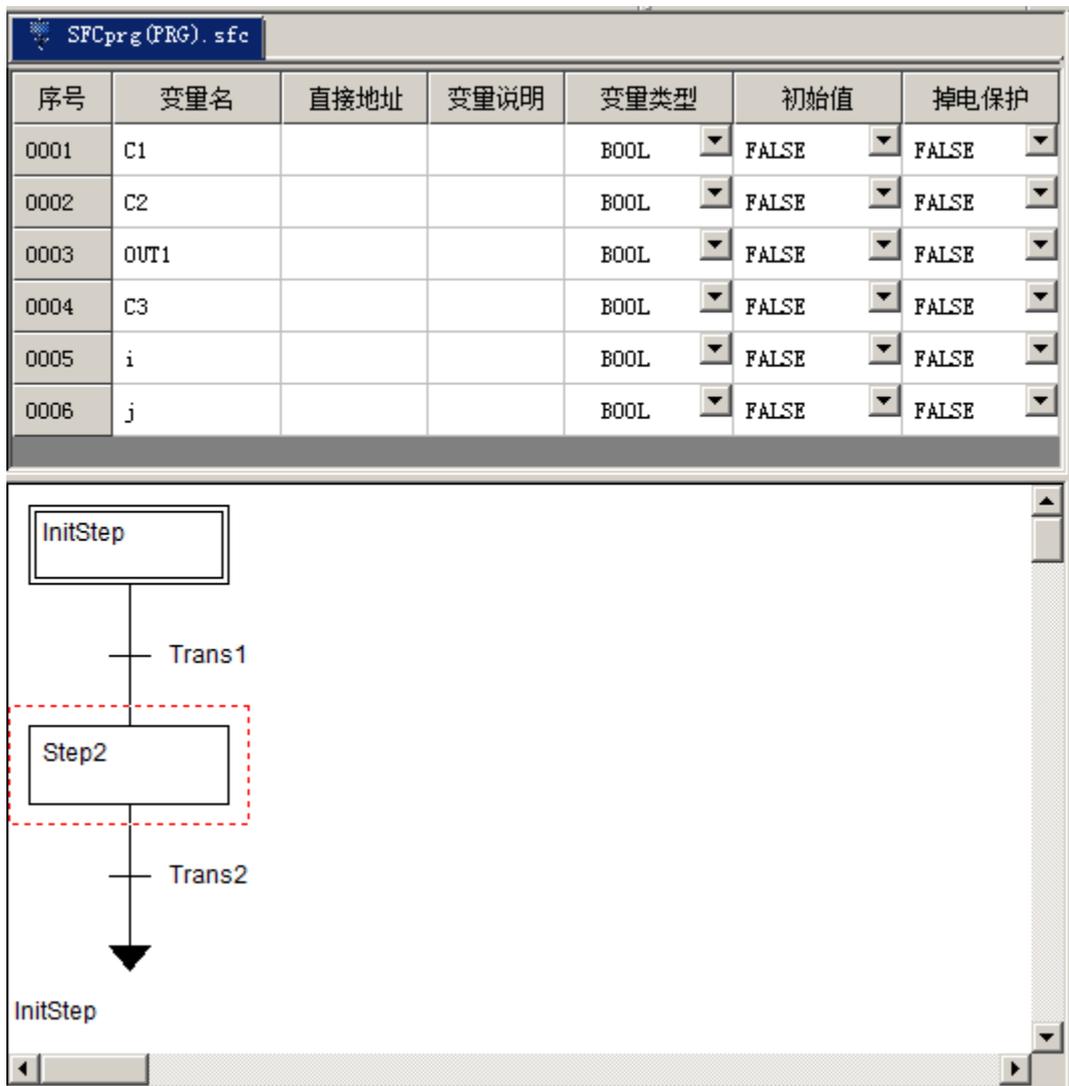


图 5.4-95 SFC 编程窗口

5.4.4.1 SFC 步元素

一个步代表一种状态，一个步要么是活动的要么是不活动的。任何时刻，程序组织单元（POU）的状态由一些活动步的设置及其内部变量的值来定义。

1. 初始步

每个 SFC 编辑的 POU 以初始步开始。这个步是进入 POU 的入口。

只有初始步总是在 SFC 的 POU 中存在的。

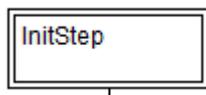


图 5.4-96 初始步

2. 步

步说明了在这个工艺步控制了什么，步的动作描述了配置的程序，这些程序可以用 ST、LD 或 CFC 编辑。

插入一个步，步名需要在 SFC 的 POU 中明确。一个步最多可以关联 9 个动作。

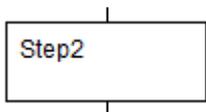


图 5.4-97 步元素

StepName.t(TIME): 步运行时间，当一个步被解除激活时，步所经历的时间值应保持在此步被解除激活时所具有的值，当步被激活时，步所经历的时间值应复位为 **t#0s**，重新开始计时。如图 5.4-98 所示，**T#10s** 为设置的步最小时间，**T#4s259ms** 为 **Step2.t** 的值。当设置了步最小时间时，该步被激活时，**StepName.t** 显示如图 5.4-98 所示位置。

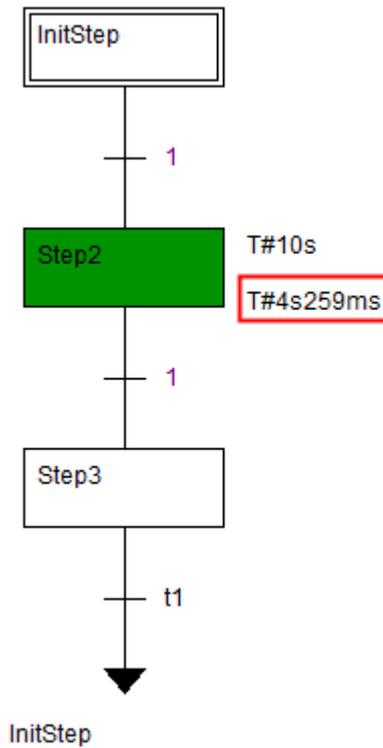


图 5.4-98 StepName.t 步运行时间在线值

StepName.x(BOOL): 步标志, 是一个 BOOL 型元素, 标示步当前的活动状态, 1: 步活动; 0: 步停用。如图 5.4-98 所示, 当 Step2 之前的转换条件变为 True 时, Step2 被激活, 此时 StepName.x 为 1, 该步为活动状态。



- 步名称、步运行时间、步标志仅在当前 POU 有效, 不支持其他 POU 引用。

5.4.4.2 SFC 动作元素

每个步应关联若干动作, 具有零个相关动作的步应该被认为具有等待功能, 即等待一个后继转换条件变为真。动作可以是 ST 语言的一组语句、LD 语言的一组梯级、或 CFC 语言的一组程序。

5.4.4.3 SFC 组态操作举例

下面给出的示例能够帮助用户更多的了解 SFC 控制。示例中以行号, 列号表示具体元素的位置。

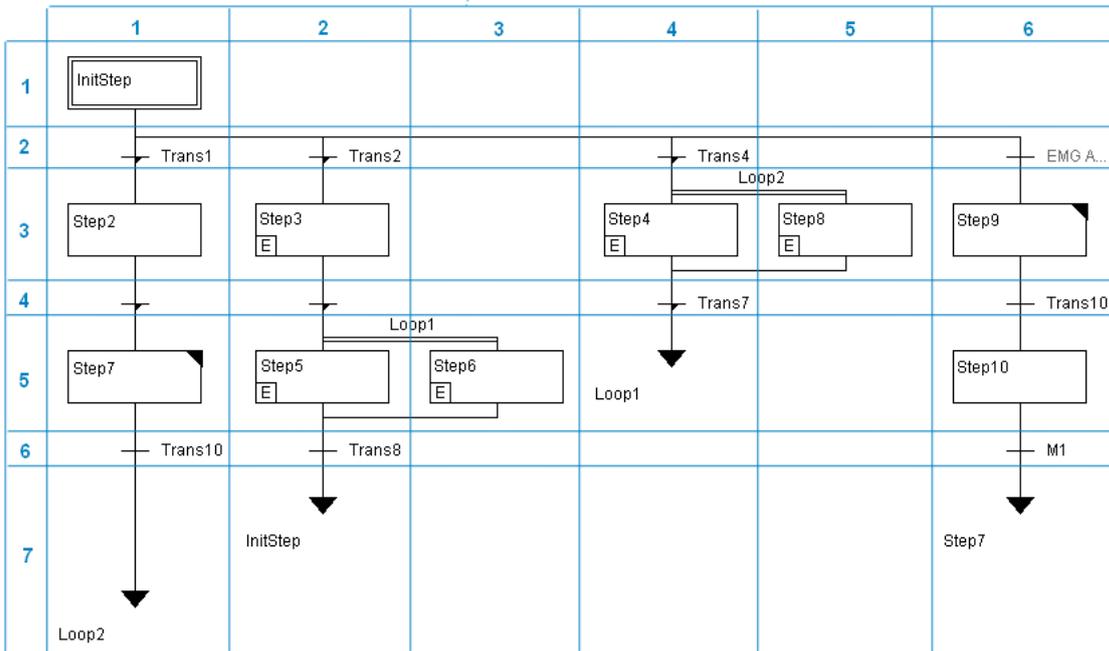


图 5.4-99 SFC 组态操作示例

一个 SFC 控制是以初始步开始（第 1 行，第 1 列）。

初始步后面可以连接转换-步，也可以像这里连接选择分支。

选择分支包含 4 个子分支，分布于第 2 行的第 1、2、4、6 列。每个子分支都是以转换开始。当初始步执行完成后，按照从左到右的优先级执行某一子分支。

转换和步是成对添加或删除的。转换（步）可以添加转换、平行分支（动作、选择分支）。例如转换可以添加转换条件（如第 2 行，第 1 列、第 2 行，第 6 列、以及第 4 行，第 1 列）、平行分支等。

步 Step4（第 3 行，第 4 列）和 Step5（第 5 行，第 2 列）分别添加平行分支，依次分布在第 3 行，第 4、5 列和第 5 行，第 2、3 列，分支标号分别是 Loop2 和 Loop1。

平行分支用双水平线连接子分支。平行线上可以输入分支标号，主要用来标记跳转位置；每个子分支以步开始，表示当转换满足时，同时激活所有子分支的第一步。

如果前一步 InitStep（第 1 行，第 1 列）处于激活状态，此时 Trans4（第 2 行，第 4 列）条件满足，会同时激活步 Step4 和 Step8。

如果前一步 Step3（第 3 行，第 2 列）处于激活状态，此时转换条件（第 4 行，第 2 列）条件满足，会同时激活步 Step5 和 Step6。

所有子分支都支持以跳转结束，跳转目标可以是步、并行分支标号、初始步（默认）。

这里的目标跳转位置有：

并行分支标号：第 7 行，第 1 列和第 5 行，第 4 列；

初始步：第 7 行，第 2 列；

步：第 7 行，第 6 列。

5.4.4.4 光标位置

通过单击来选中元素，选中的元素是用封闭的虚线框来标识，如图 5.4-100 所示。

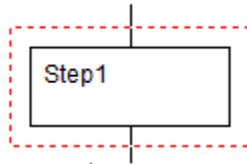


图 5.4-100 被选中的步

移动鼠标到元素上并单击，可以选择一个元素（步、转换或跳转）。

为了选择一组中多个元素，在选中的元素上按 **Shift**，并选择组的左下角或右下角的元素，这样就选中了包括这些元素的最小的组。

如图 5.4-101 所示，选择不同的元素组合，选择的结果可能会不同。

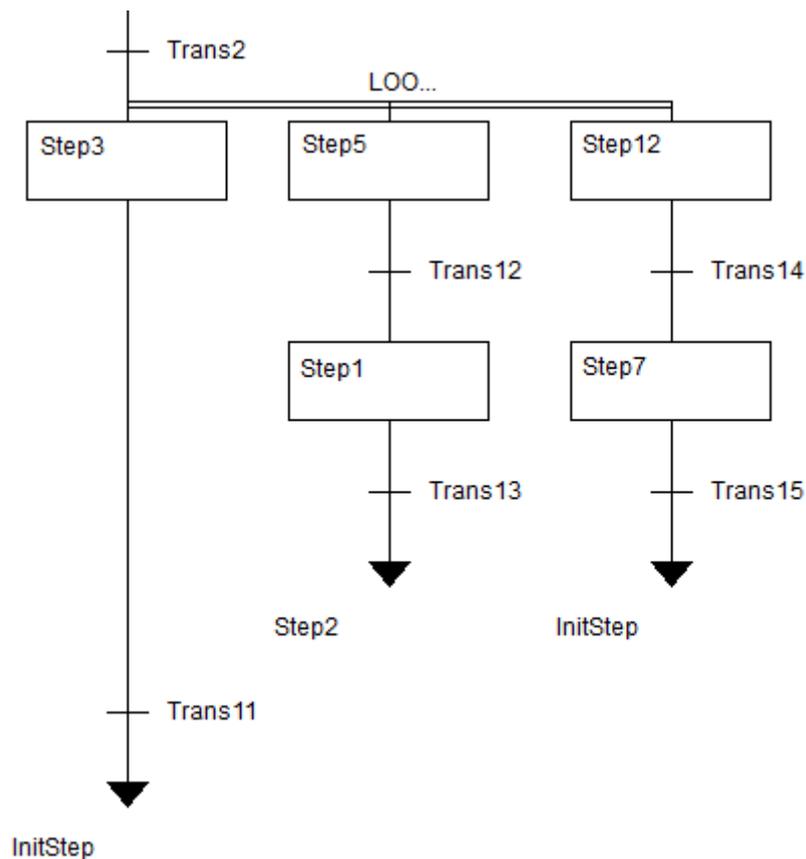


图 5.4-101 选择元素前

先选择元素 Step5，**Shift** 键按下时选择跳转元素 Step2，它们所在分支的所有内容被选中，如图 5.4-102 所示。

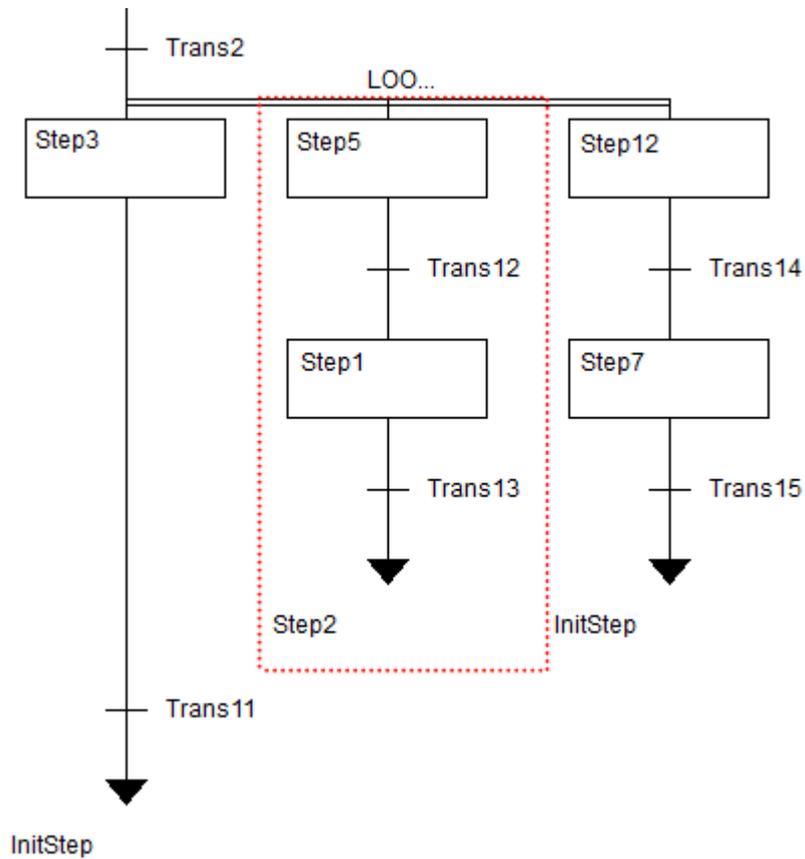


图 5.4-102 选中一组分支

先选择元素 Step3，按下 **Shift** 键的同时选择 Step3 所属分支（Trans11 和跳转 InitStep）外的其他任何元素，最终会得到同一种结果，如图 5.4-103 所示。

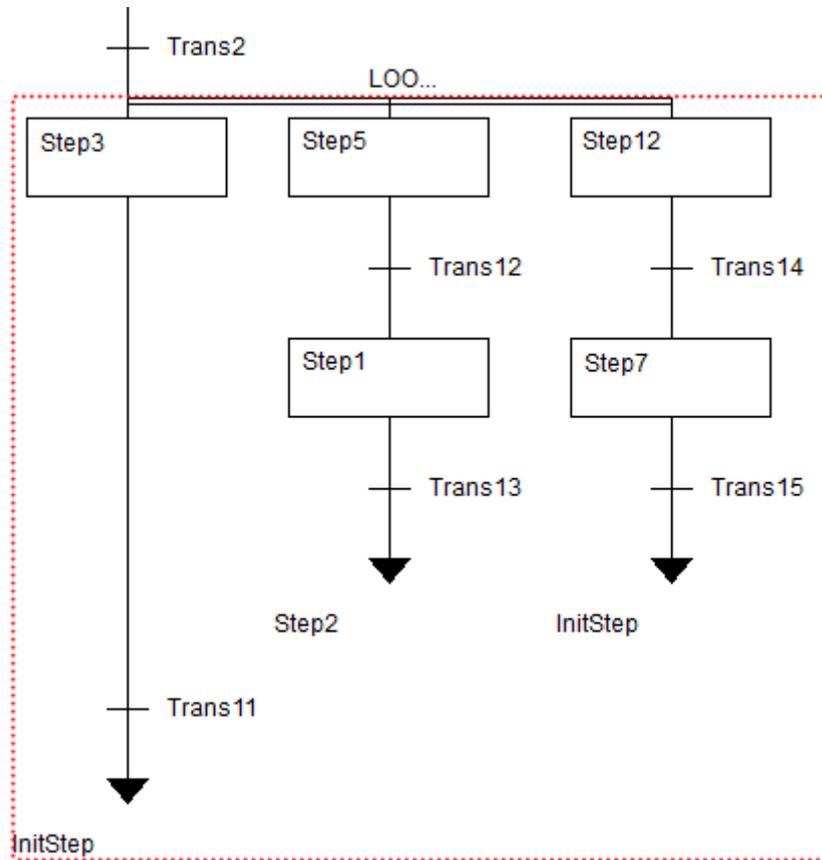


图 5.4-103 选中一组并行分支



- 步只能和它之前或随后的转换一起删除！

5.4.4.5 步属性

选中任意步：



- 菜单栏：单击【附加】—【步的属性 (B)】；
- 程序区：右击步，单击【步的属性 (B)】。



图 5.4-104 “步属性”窗口

最小时间：步运行时间的下限值，当步的运行时间小于**最小时间**时，即使其出口转移条件满足，程序也不会执行转移，一直到该步的执行时间超过**最小时间**后才执行转移。建议进行设置。

注释：步的相关说明，字符数不设限。确定设置内容后，本步的右边会显示设置的注释信息，如图 5.4-105 所示。显示的注释信息多少与设置的步高度和是否显示最大、最小时间有关。

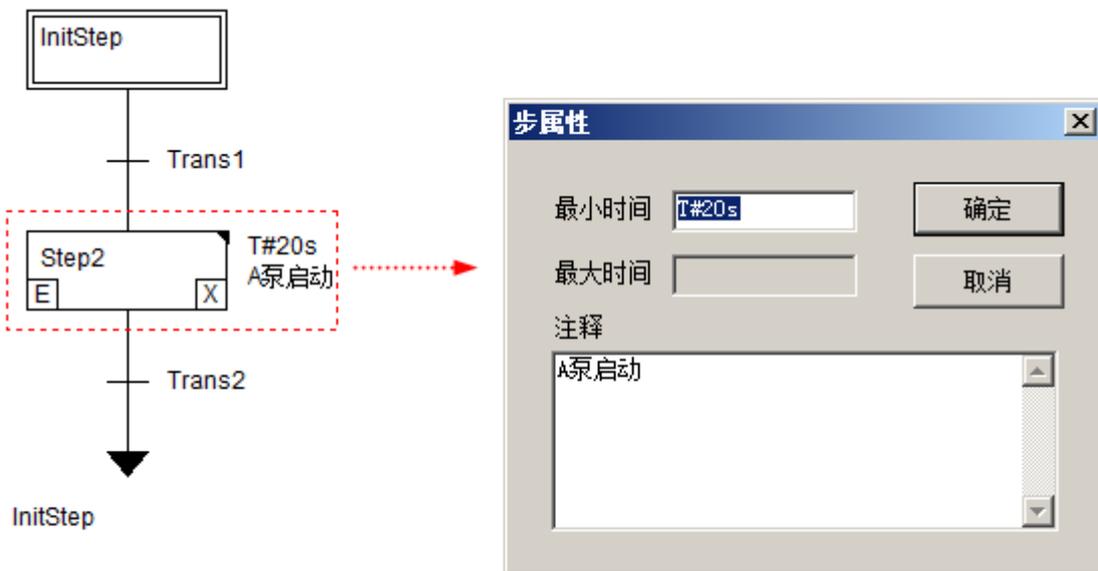


图 5.4-105 步属性设置示例



- 时间可以由 TIME 类型的变量表示，也可以用时间数据表示，常量的书写格式：T+“#”+时间值。

5.4.4.6 标志符

标识符用来控制 SFC POU 中的程序流，它在工程运行期间隐含创建。为了能读这些标识符，必须定义合适的局部变量。

SFCInit: BOOL 类型变量。当这个变量值是 TRUE 时，顺序功能图返回到初始步，同时其它的 SFC 标志符也会被复位。初始步保持激活，但是不被执行。只有当 SFCInit 被重设置为 FALSE，顺序功能图才可以恢复正常执行。

SFCPause: BOOL 类型变量。只有这个变量是 TRUE，顺序功能图停止执行。如图 5.4-106 所示。

SFCReset: BOOL 类型变量。当该变量为 TRUE，则 SFC 程序被复位，返回到初始步，同时其它的 SFC 标志符也会被复位。初始步保持激活。与 SFCInit 不同之处在于 SFCReset 变量值为 TRUE 时，初始步的动作不会受影响，继续执行。同时，SFCReset 标识符可以在初始步中被复位到 FALSE。

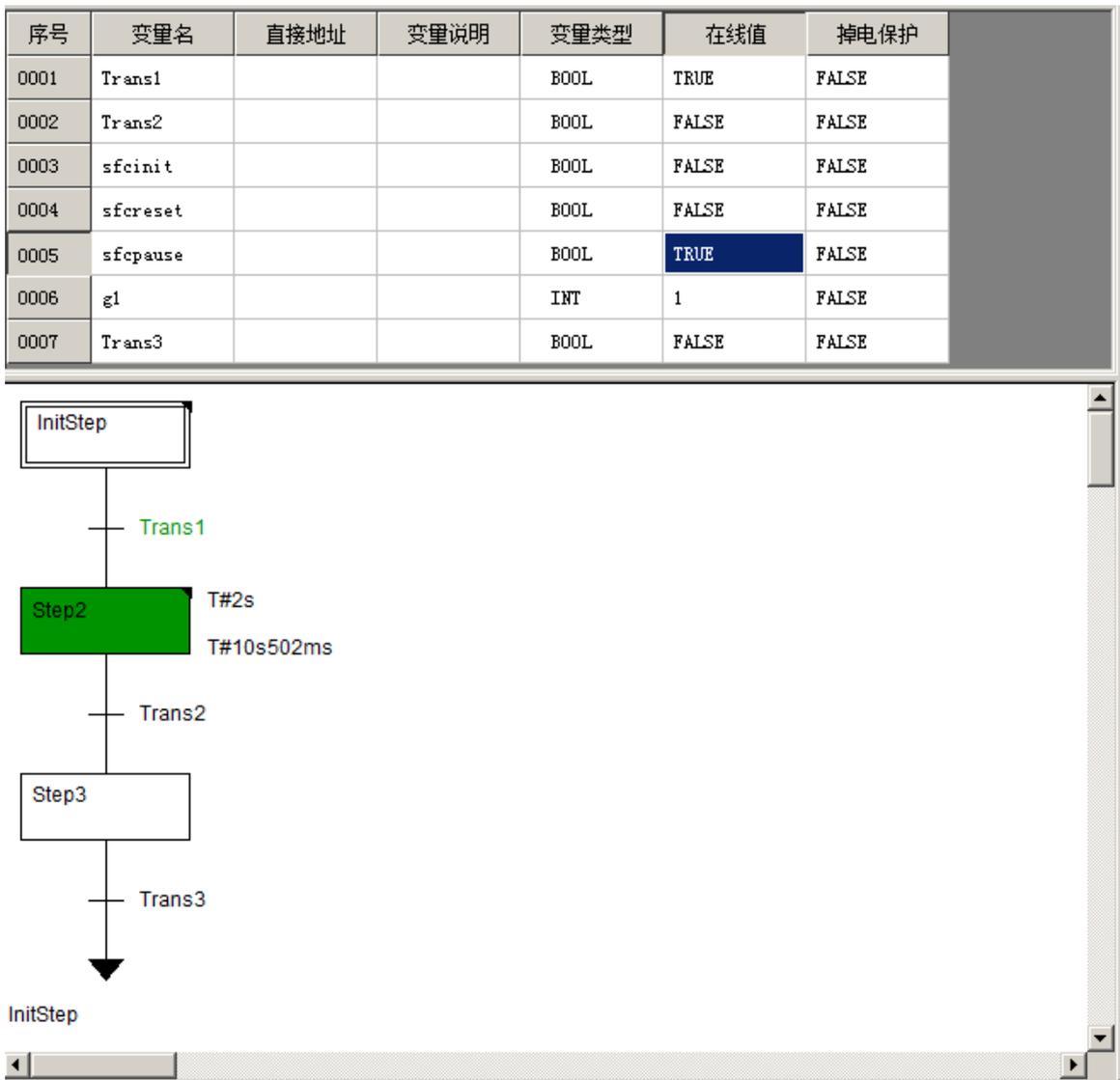


图 5.4-106 SFCPause 变量为 TRUE

5.4.4.7 插入步

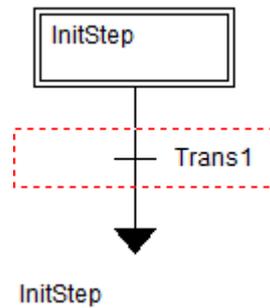
(1) 向前插入步

选中转换或步，执行以下操作：

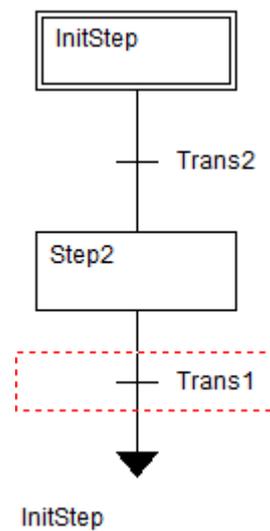


- 菜单栏：【插入】—【步—转换（前）】；
- 程序区：右击转换或步，单击【步—转换（前）】；
- 工具栏：；
- 快捷键：**Ctrl+T**。

选中转换时，在转换前插入一个新转换和一个新步，如图 5.4-107 所示。



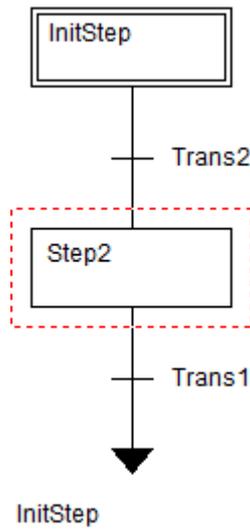
(a)



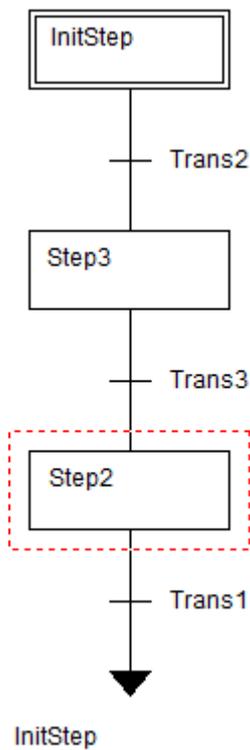
(b)

图 5.4-107 向前插入步—选中转换

选中步时，在步前插入一个新步和一个新转换，初始步（InitStep）不能向前插入步，如图 5.4-108 所示。



(a)



(b)

图 5.4-108 向前插入步—选中步

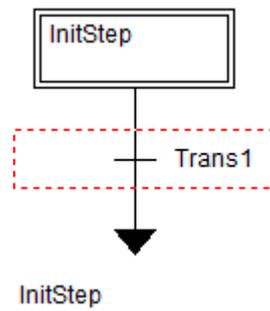
(2) 向后插入步

选中步和转换，执行以下操作：

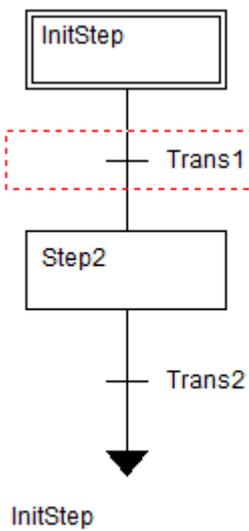


- 菜单栏：【插入】—【步—转换（后）】；
- 程序区：右击转换或步，单击【步—转换（后）】；
- 工具栏：
 ；
- 快捷键：Ctrl+E。

选中转换时，在转换后插入一个新步和一个新转换，如图 5.4-109 所示。



(a)



(b)

图 5.4-109 向后插入步—选中转换

选中步时，在步后插入一个新转换和一个新步，如图 5.4-110 所示。

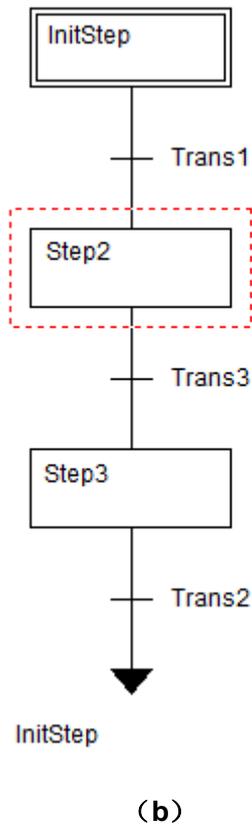
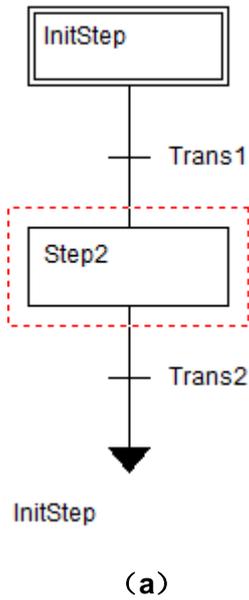


图 5.4-110 向后插入步—选中步



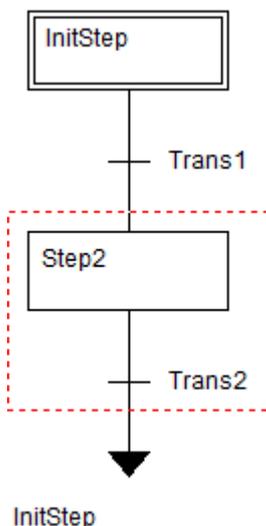
- 实际上，插入步是插入“步-转换”对，使原有的 SFC 程序保持基本正确的逻辑流程。
- 每个 SFC POU 添加“步”的最大数是 2000 个。

5.4.4.8 删除步

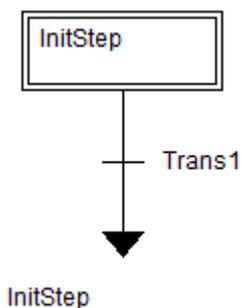
删除步也是删除“步-转换”对，使原有的 SFC 程序保持基本正确的逻辑流程。初始步不能被删除。

(1) 删除单个“步-转换”对

选中需要删除的“步-转换”对：单击鼠标左键选中步 Step2 后，按住 **Ctrl** 或 **Shift** 键，同时单击转换 Trans2，同时选中 Step2 和 Trans2，执行删除命令，如图 5.4-111 所示。



(a)

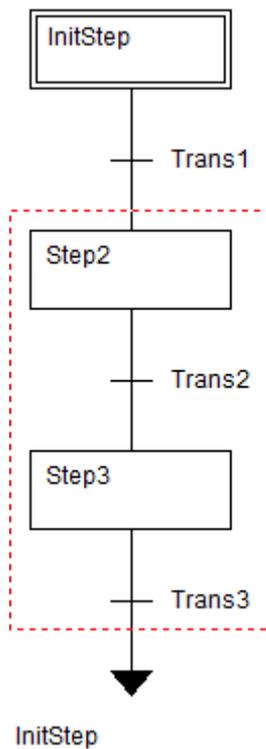


(b)

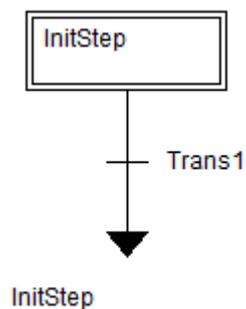
图 5.4-111 删除单个“步-转换”对

(2) 删除多个“步-转换”对

选中需要删除的“步-转换”对: 单击鼠标左键选中转换 Trans3 后, 按住 **Ctrl** 键, 依次单击步 Step3、转换 Trans2 和步 Step2, 或按住 **Shift** 键, 选中转换 Trans3 和步 Step2, 执行删除命令, 如图 5.4-112 所示。



(a)



(b)

图 5.4-112 删除多个“步-转换”对

5.4.4.9 修改步名称

新建步时系统会自动为其分配名称，如需对其修改，新名称需要按照普通变量命名规则进行设置，并确保在工程中不重名。

双击步的名称文本所在的矩形区域，步名变为可编辑状态，输入新步名即可，如图 5.4-113 所示。

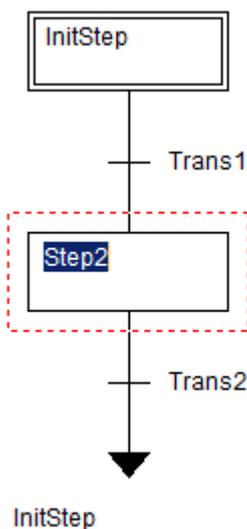


图 5.4-113 修改步名称

输入新名称后，在任意空白处单击或者 **Enter** 键确定，则编辑结束。

5.4.4.10 添加入口动作

入口动作，指本步刚进入激活状态时执行的动作。

选中步，通过以下方式添加入口动作：



- 菜单栏：单击【插入】—【添加入口动作】；
- 程序区：右击步，单击【插入】—【添加入口动作】；
- 快捷键：**Ctrl+I**。

弹出选择动作语言类型的对话框，如图 5.4-114 所示。



图 5.4-114 入口动作语言类型

选择动作描述语言后，单击**确定**，完成了添加入口动作操作，在步的左下角会出现[E]标识，如图 5.4-115 所示，双击该标识，可以编辑步的入口动作。

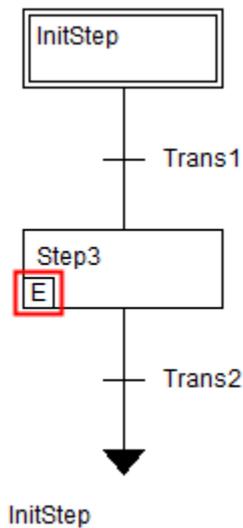


图 5.4-115 入口动作标识



- 初始步（InitStep）不能添加入口动作。

5.4.4.11 添加出口动作

出口动作，指本步即将退出激活状态时，执行的动作。

选中步，通过以下方式添加出口动作：



- 菜单栏：单击【插入】—【添加出口动作】；
- 程序区：右击步，单击【插入】—【添加出口动作】；
- 快捷键：**Ctrl+G**。

弹出选择动作语言类型的对话框，如图 5.4-116 所示。

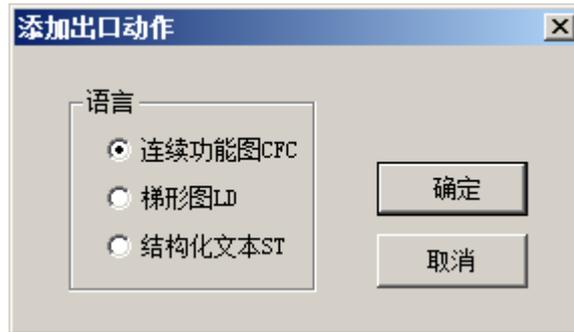


图 5.4-116 出口动作语言类型

选择动作描述语言后，单击**确定**，完成添加出口动作操作，在步的右下角会出现  标识，如图 5.4-117 所示，双击该标识，可以编辑步的出口动作。

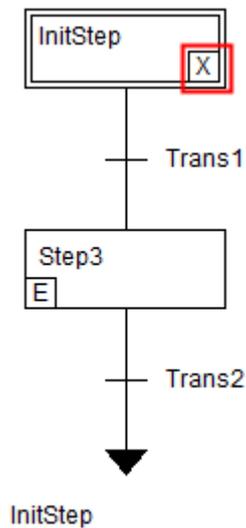


图 5.4-117 出口动作标识

5.4.4.12 关联动作

通过【关联动作】操作，将 SFC POU 中增加的动作关联到某一步中。

动作与步的关联关系是由动作限定符决定的。动作限定符如图 5.4-118 所示。



图 5.4-118 动作限定符

A 区域为动作限定符，自带下拉框，可以选择所需要的限定符；如果 A 区域选择与时间相关的限定符可在 B 区域对时间进行编辑，默认值为 T#1S；C 区域为一个动作名称，当在 POU（SFC）中增加动作后（请参见[增加动作](#)），则动作名会在 C 区域的下拉框中显示，可以对其进行调用；该动作是否被激活与 A 区域的限定符有密切关系。如表 5.4-3 所示，列出了不同限定符和动作的执行关系。

表 5.4-3 动作限定符

序号	限定符	类型说明	说明
1	N	非存储	
2	R	存储复位	
3	S	设置存储	
4	L	限时	

序号	限定符	类型说明	说明
5	D	延时	
6	P	脉冲	
7	SD	存储和延时	
8	DS	延时和存储	

序号	限定符	类型说明	说明
9	SL	存储和限时	
10	P1	上升沿脉冲	
11	P0	下降沿脉冲	

选中步：



- 菜单栏：单击【附加】—【关联动作】；
- 程序区：右击步，单击【关联动作】。
- 快捷键：**Ctrl+K**。

即为步关联了一个新的动作。如图 5.4-119 中所示，步 Step3 关联了 3 个动作。

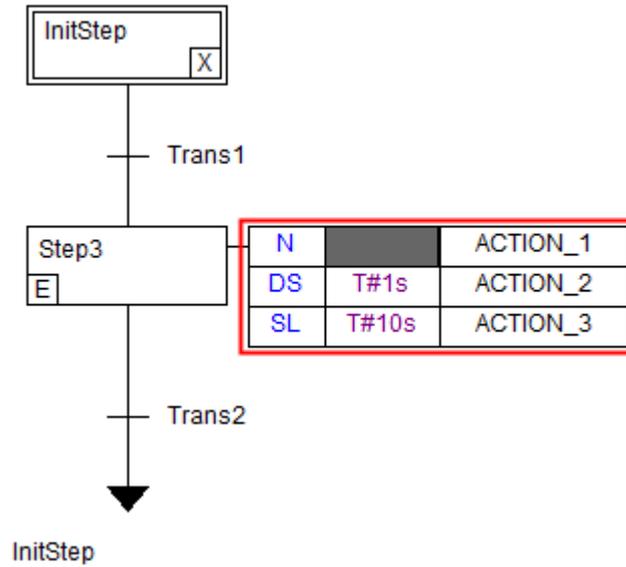


图 5.4-119 关联动作

5.4.4.13 SFC 步动作执行时序

如图 5.4-120 所示，Step1 添加了出口动作，Step2 添加了入口动作、步动作、IEC 关联动作。当前扫描周期内，步动作的执行如表 5.4-4 所示。

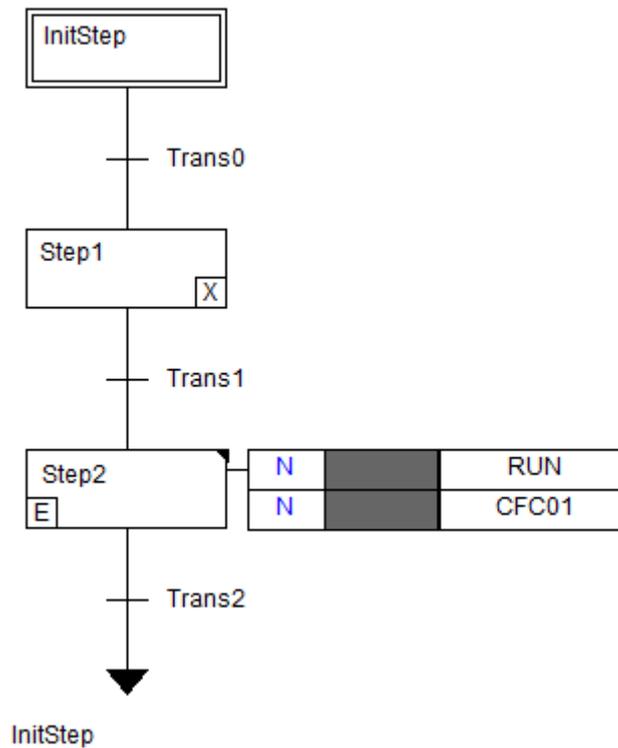


图 5.4-120 步执行动作示例

表 5.4-4 动作执行顺序

执行时序	动作	动作处理
1	Step1 出口动作	当 Step1.x=TRUE，且 Trans1=TRUE，该动作被执行。（它的入口动作和步动作已经在上一个周期被执行了）
2	Step2 入口动作	如果 Trans1=TRUE，且 Step2 被激活（即 Step2.x=TRUE），则入口动作被执行。
3	Step2 步动作	Step2 被激活后，且入口动作已执行完，则执行步动作。
4	Step2 IEC 关联动作	步动作执行完毕，IEC 关联动作按照字母顺序执行。
5	转移，激活下一步	如果 Step2 是激活状态，Trans2 为 TRUE，且 Step2.t 大于最小时间，则后续步被激活。

5.4.4.14 SFC 演变规则

序号	举例	规则
1		<p>单个分支: 连续地重复进行步—转换的交替。</p> <p>举例: 只有当步 S3 处于活动状态且转换条件 c 为真时, 从 S3 到 S4 的转移才会发生。</p>
2		<p>选择一支: 几个分支之间的选择可以用水平线下面的若干转换符号表示, 有几个转换符号, 就有几个可能的不同转移。</p> <p>举例: 仅当 S5 活动且转换条件 e 为真时, 从 S5 到 S6 的转移才会发生; 或仅当 S5 活动且 f 为真、e 为假时, 从 S5 到 S8 的转移才会发生。</p>
3		<p>选择分支的汇集: 选择分支的结束由水平线上面的若干转换符号表示, 有几个转换符号, 就有几个结束的选择路径。</p> <p>举例: 仅当 S7 活动且转换条件 h 为真时, 从 S7 到 S10 的转移才会发生; 或仅当 S9 活动且 j 为真时, 从 S9 到 S10 的转移才会发生。</p>
4		<p>并列一支: 在并列分支的双水平线上面只可能有一个公用的转换符号。</p> <p>举例: 仅当 S11 活动且公用转换相关的转换条件 b 为真时, 从 S11 到 S12、S14、.....的转移才会发生。S12、S14 等同时激活后, 每个分支独立地执</p>

序号	举例	规则
5		<p>行。</p> <p>并列分支—汇集: 在并列分支的水平线下面只可能有一个公用的转换符号。</p> <p>举例:</p> <p>仅当与水平线上面连接的所有步都活动, 且与公用转换相关的转换条件 d 为真时, 从 S13、S15、.....到 S16 的转移才会发生。</p>
6		<p>分支跳变: “分支跳变”是选择分支的一种特殊情况, 在其中的一个或多个分支不包含步。</p> <p>举例:</p> <p>当 a 为假且 d 为真时, S30 到 S33 的演变才会发生, 即跳过序列 S31 和 S32。</p>

序号	举例	规则
7		<p>分支跳转：是选择分支或并列分支中的一种特殊情况，在其中的一个或多个分支跳转到其它步。</p> <p>举例：当 S33 活动且 d 为真时，S33 到 S31 的转移才会发生。</p>

5.4.4.15 添加动作/转换

1. 添加步动作

选中步：



- 菜单栏：单击【附加】—【添加动作\转换】；
- 程序区：右击步—【添加动作\转换】；
- 快捷键：**Ctrl+W**。



图 5.4-121 步动作语言类型

选择动作描述语言后，单击**确定**，完成添加步动作操作。在步的右上角会出现 标识，如图 5.4-122 所示，双击该标识，打开相应语言环境的步编辑页面，可以编辑步动作。

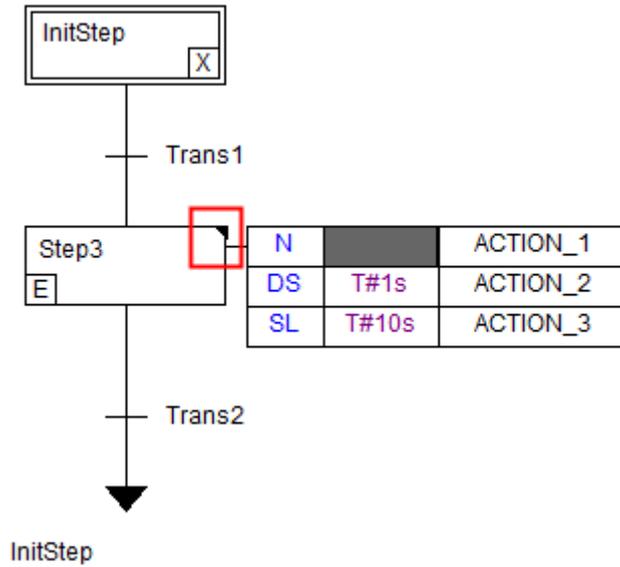


图 5.4-122 步动作标识

2. 转换添加转换

选中转换条件，执行【添加动作\转换】命令：



- 菜单栏：单击【附加】—【添加动作\转换】；
- 程序区：右击转换，单击【添加动作\转换】；
- 快捷键：**Ctrl+W**。



图 5.4-123 转换语言

选择转换条件描述语言，点击**确定**即可。

转换在添加转换后，转换位置会出现  标识，如图 5.4-124 所示。

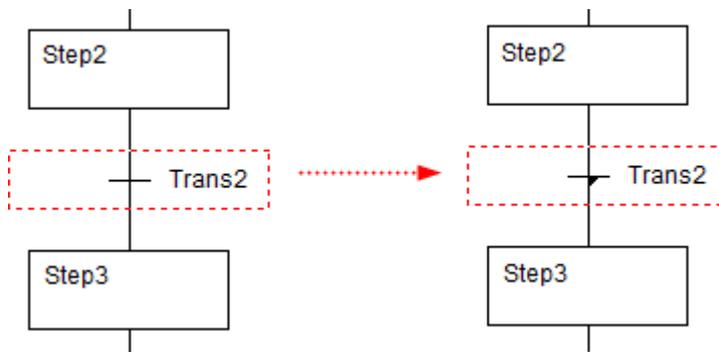


图 5.4-124 已添加转换的标识



- 当转换语言选择 ST 编辑时，转换内容的书写格式必须是赋值语句，以“Trans:=”开始。
- 当添加转换后，就有了两个转换条件，这时只有用 CFC/LD/ST 编辑的转换条件有效。

5.4.4.16 移除动作/转换

1. 移除步动作

对已添加动作的步，按实际需要进行移除。

选中步：



- 菜单栏：单击【附加】—【移除动作\转换】；
- 程序区：右击步，单击【移除动作\转换】；
- 快捷键：**Ctrl+U**。

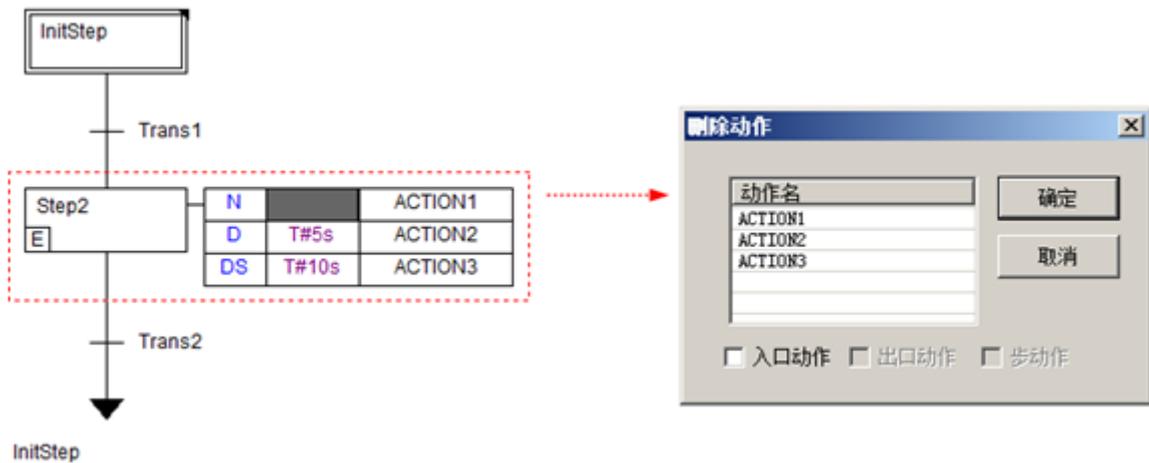


图 5.4-125 移除动作



- 当步只有一个动作时，执行【移除动作\转换】命令后，会直接将动作删除，不会弹出“删除动作”对话框。

选择后，单击**确定**，完成移除操作。关联动作的移除也可以直接右击需要删除的动作，选择【删除】命令，如图 5.4-126 所示。

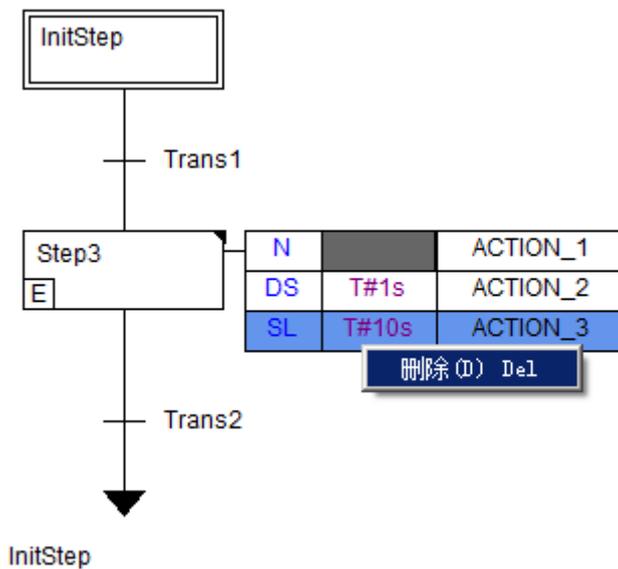


图 5.4-126 删除关联动作

2. 移除转换

选中转换，右键菜单选择【移除动作\转换】命令，已添加的转换被删除。

5.4.4.17 增加动作

右击 SFC 语言的 POU, 选择弹出菜单的【增加动作】项, 可以为该 POU 增加一个动作, 如图 5.4-127 所示。

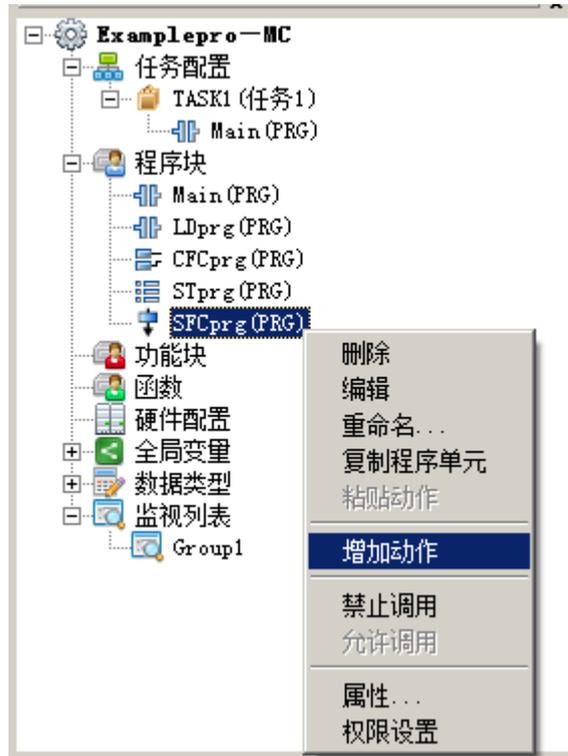


图 5.4-127 增加动作操作



图 5.4-128 添加动作

输入**动作名**, 不超过 30 个字符, 选择编辑语言后, 单击**确定**, 添加动作完成。在 POU 子目录会出现动作名。如图 5.4-129 中所示, SFCprg(PRG)添加了一个编辑语言是 CFC, 动作名是 CFC01(ACTION) 的动作。

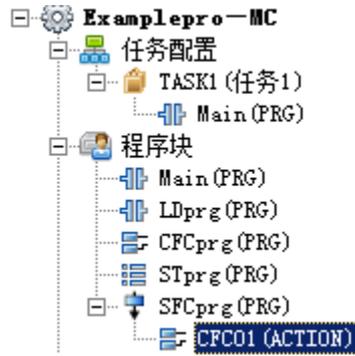


图 5.4-129 添加的动作节点

5.4.4.18 复制动作

右击动作名，在快捷菜单中可以选择对该动作进行相关操作，如图 5.4-130 所示。

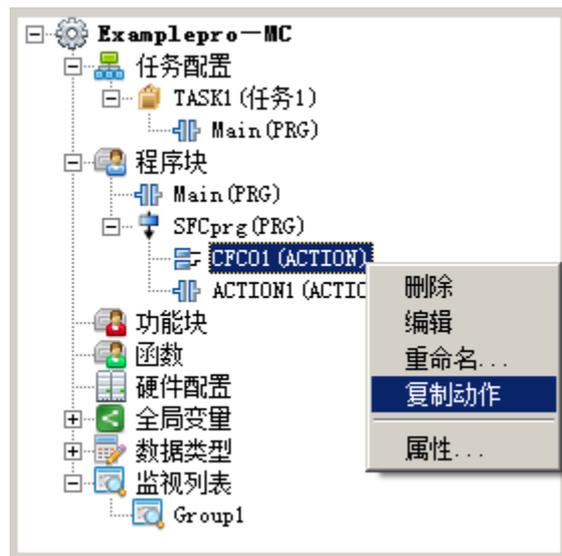
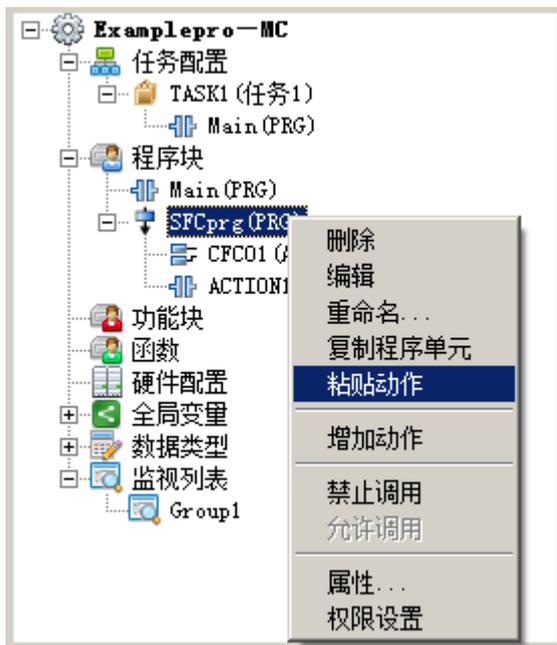


图 5.4-130 动作相关操作

执行其中的【复制动作】命令后，用户可以将复制的内容粘贴给对多个 SFC POU。

如图 5.4-131 所示，复制 SFCprg 的 IEC 动作 CFC01，执行【粘贴动作】命令后，系统会自动将动作名变为“NewAction*”。



(a)



(b)

图 5.4-131 复制、粘贴动作操作



- 每个“步”最多能关联 9 个动作。
- 每个 SFC 关联的动作最大数是 1024 个，其中包含“步”动作（出入口动作、步动作）、“转换”动作和 SFC POU 的 IEC 动作。
- 复制动作时，复制的内容不包含动作中局部变量的定义。

5.4.4.19 转换元素

转换表示控制从一个或多个前驱步沿相应的有向连接转换到一个或多个后继步所依据的条件。转换方向是从前驱步底部到后继步顶端。

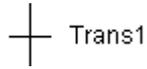


图 5.4-132 转换



- 每个 SFC POU 添加“转换”的最大数是 2000 个。

每个转换应有一个转换条件，它是一个单 **BOOL** 表达式的求值结果。总之，为真的转换条件应由符号“1”或关键字“**TRUE**”来表示。转换的求值可以由 **LD**、**CFC** 和 **ST** 得出。在转换条件求值期间，如果给变量赋值而不是给转换，是一个错误情况。转换相关描述如表 5.4-5 所示。

表 5.4-5 转换相关描述

序号	举例	说明
1		前驱步 ST 语言描述的转换条件, 转换条件为单 BOOL 求值表达式 后继步
2		前驱步 转换名 后继步

序号	举例	说明
3		LD 语言描述的转换条件
4		CFC 语言描述的转换条件
5	<pre>TRANS:=A&B;</pre>	ST 描述的转换条件

(1) 单 BOOL 表达式做转换条件

直接在转换名称处写入单 BOOL 表达式作为转换条件。如图 5.4-133 所示。

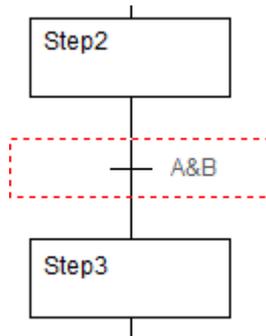


图 5.4-133 转换条件—单 BOOL 表达式

(2) 编辑已添加的转换

无论是转换条件还是已添加的转换，在需要对转换内容进行编辑时，双击转换条件或已添加转换的标识即可。

(3) 剪切、复制、粘贴

可以剪切和复制的单元包括：步与转换组成的序列对、并行分支和选择分支。复制的单元内容中不能包括初始步。

粘贴，剪贴板上的数据是以下几种单元的任意一种时，可以执行粘贴命令：

- 以步开始并且以步结束的单元；
- 以转换开始并且以转换结束的单元；
- 以步开始并且以转换结束的单元；

- 以转换开始并且以步结束的单元。

5.4.4.20 并行分支

两个或者两个以上以步起始并且以步结束的 SFC 单元，通过相互平行的双实线并行起来形成并行分支。并行分支如图 5.4-134 所示，当转换 Trans1 为真并且初始步 (InitStep) 活动时，同时激活 STEP2 和 STEP4。

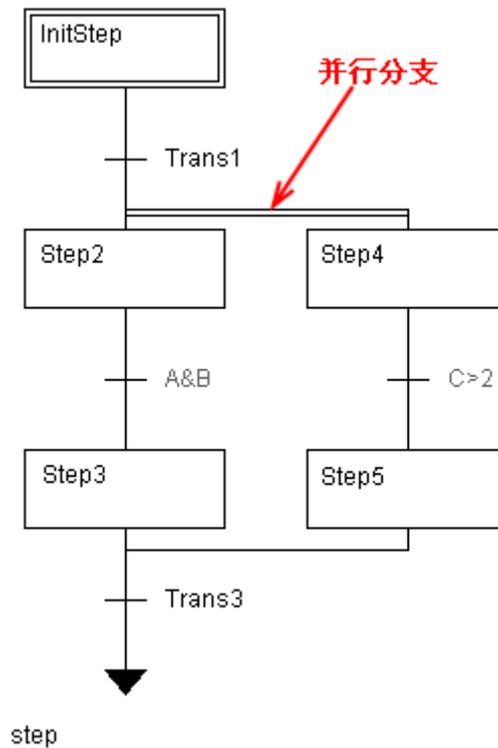


图 5.4-134 并行分支

(1) 增加并行分支

选中单个步或者一个以步开始且以步结束的单元，通过以下方式增加并行分支：



- 菜单栏：单击【插入】—【并行分支（右）/（左）】；
- 工具栏：、；
- 程序区：右击选中的步或单元，单击【并行分支（右）/（左）】。

只介绍向右增加并行分支，向左增加并行分支类似。

选中单个步，增加向右并行分支，如图 5.4-135 所示。

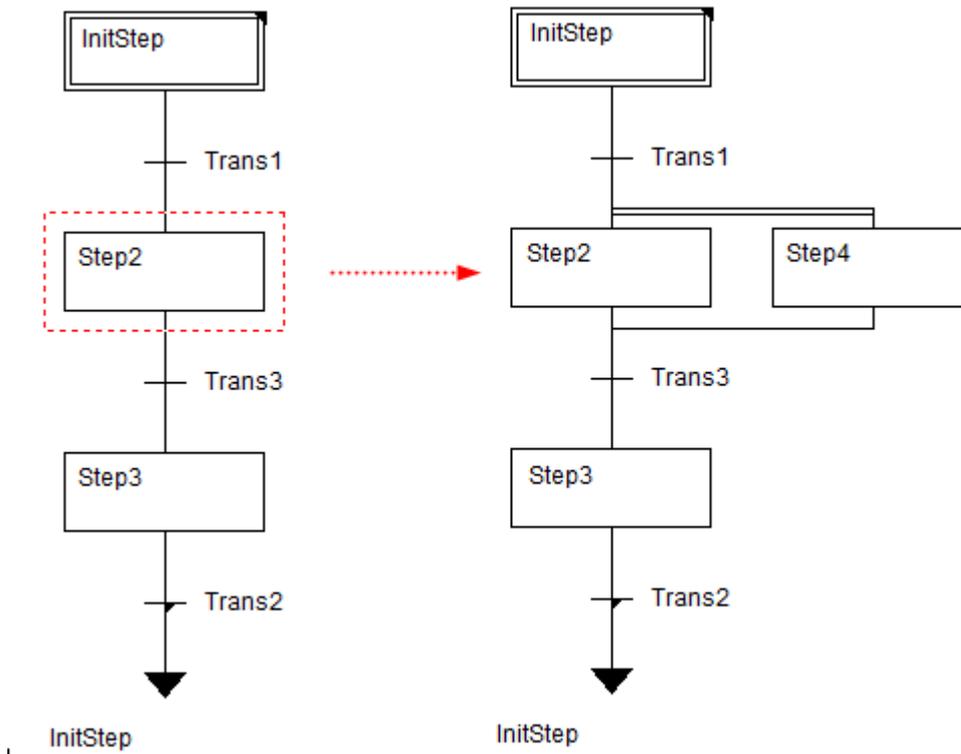


图 5.4-135 增加右并行分支一单个步

选中一个以步开始且以步结束的单元，如果是复选，需要按住 **Ctrl** 键或者 **Shift** 键选中步单元。向右增加并行分支，如图 5.4-135 所示。

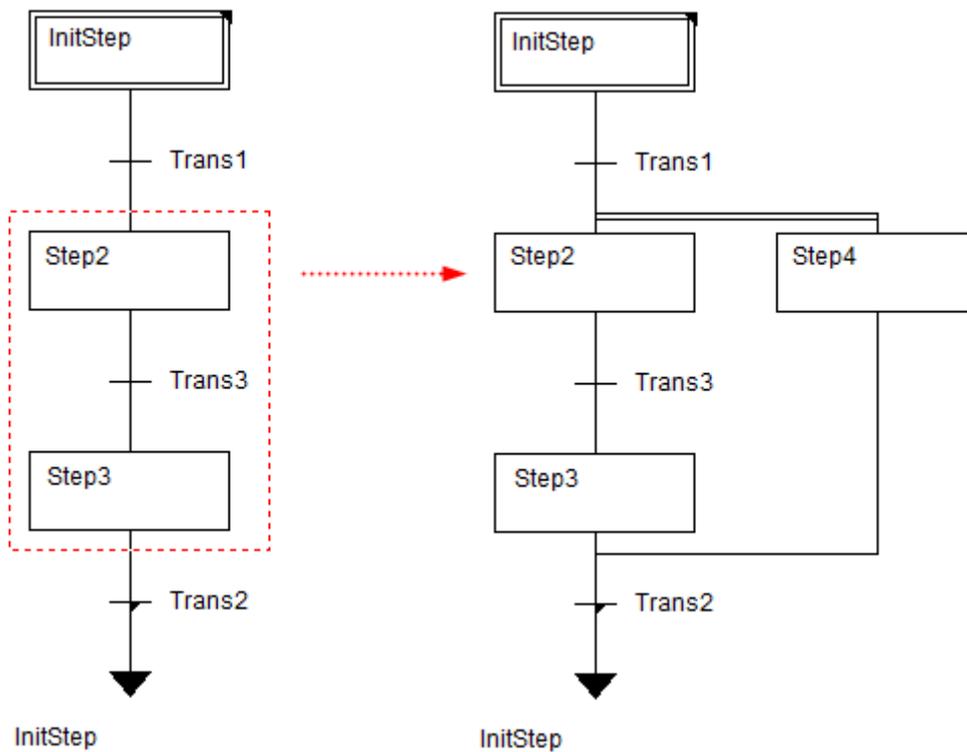


图 5.4-136 增加右并行分支一步单元

(2) 并行分支标号

可以为整个并行分支添加一个标号，作为并行分支位置的标示，可以跳转到该位置。

选中并行分支前的转换：



- 菜单栏：单击【附加】—【增加并行分支标号】；
- 程序区：右击转换，单击【增加并行分支标号】；
- 快捷键：**Ctrl+Q**。

如图 5.4-137 所示，缺省标号名 **LOOP**，可以对其进行修改，双击选中后，输入实际标号名（转换-跳转位置），在任意空白处单击鼠标左键或按 **Enter** 键完成重命名操作。

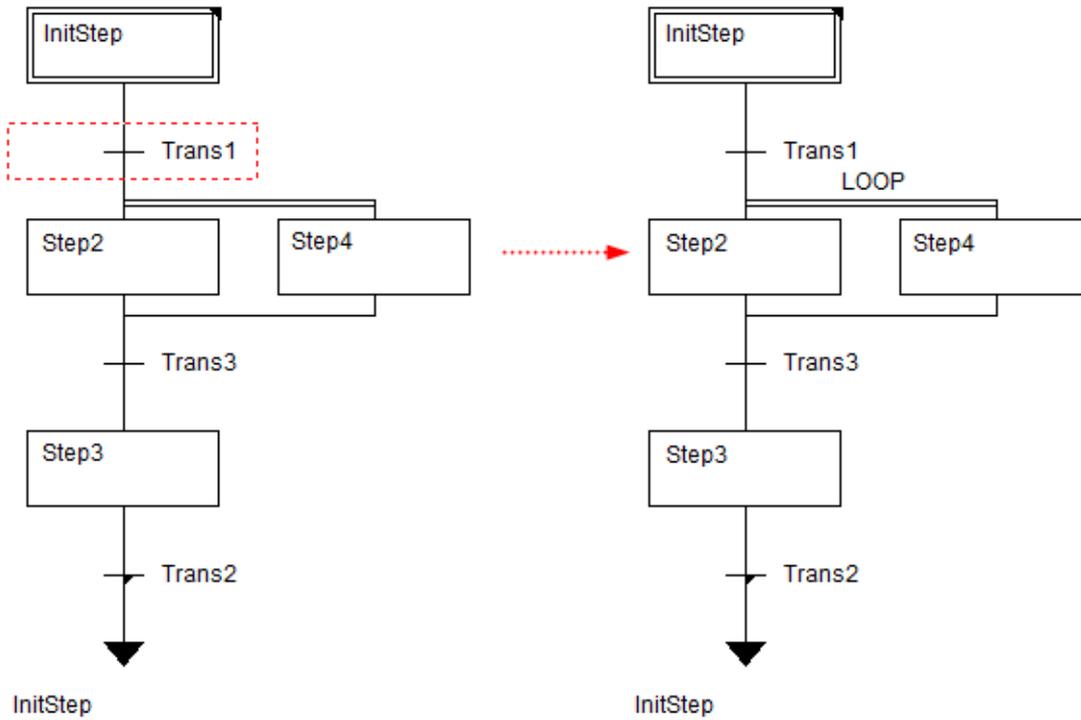


图 5.4-137 并行分支标号

(3) 删除并行分支

选中并行分支（左或右）的所有元素，执行删除命令即可。

5.4.4.21 选择分支

两个或者两个以上以转换起始并且以转换结束的 SFC 单元，通过相互平行的单实线并行起来形成选择分支，如图 5.4-138 所示。初始步（InitStep）活动时，如果转换 Trans4 为真且 Trans1 为假，则激活 Step4，或者初始步活动，Trans1 为真则激活 Step2。

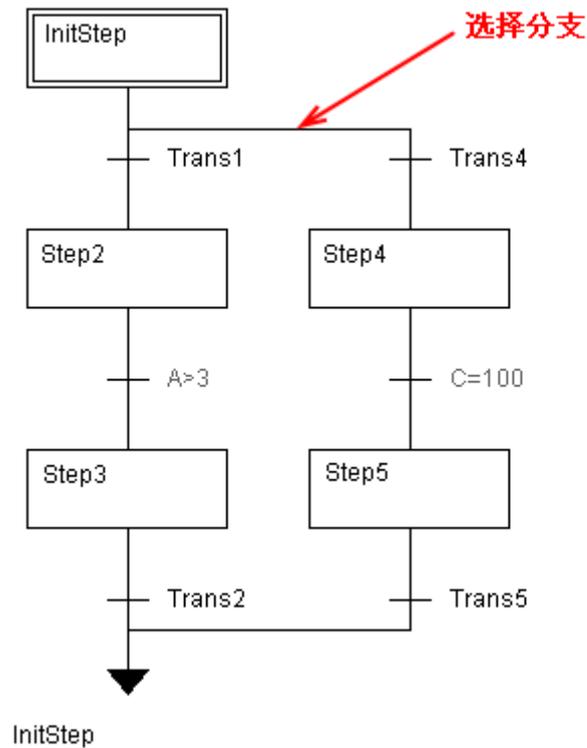


图 5.4-138 选择分支

(1) 增加选择分支

选中单个转换或者一个以转换开始并且以转换结束的单元，通过以下方式添加选择分支：



- 菜单栏：单击【插入】—【选择分支（右）/（左）】；
- 工具栏：、；
- 程序区：右击选中的步或单元，单击【选择分支（右）/（左）】。

下面仅介绍向右增加选择分支，向左增加类似。

选中单个转换“Trans1”，增加选择分支前后如中图 5.4-139 所示。

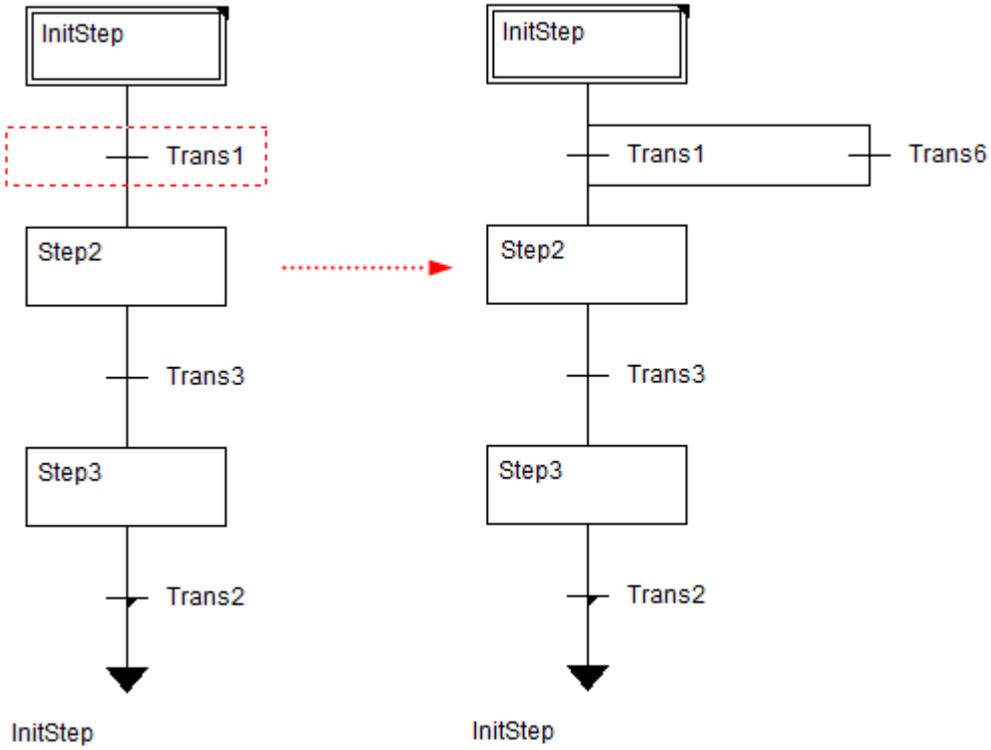


图 5.4-139 单个转换增加选择分支

选中一个以转换为起始并且以转换为结束的单元，如果是复选需要按住 **Ctrl** 键或者 **Shift** 键，执行增加选择分支前后对比如图 5.4-140 所示。

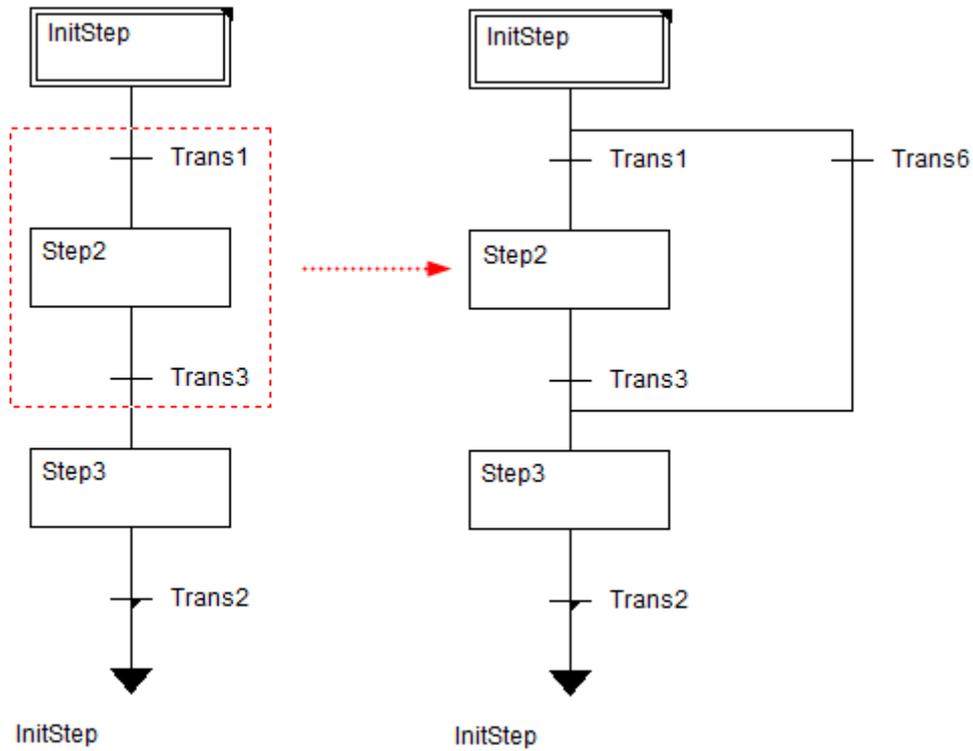


图 5.4-140 单元增加选择分支



- 如果分支的转换条件（如图中的 Tran1 和 Trans6）同时满足，则程序执行的优先级是从左到右。

(2) 删除选择分支

选中选择分支的所有元素，执行【删除】命令。

5.4.4.22 跳转

SFC 程序中的转移，通过该元素使 SFC 实现循环，SFC 语言编辑的 POU 至少包含一个跳转。

跳转是一种转移方法，可以从某一步直接转移到当前分支内部或外部的其他步。跳转用来代替一个步，并且由它所在位置的转换来执行，转换条件满足时，跳转才执行。



图 5.4-141 跳转

(1) 增加跳转

只能在选择分支上插入跳转元素。跳转元素的跳转位置必须是步名或者并行分支标号名。

选中选择分支上的一个元素，通过以下方式增加跳转：



- 菜单栏：单击【插入】—【跳转 (J)】；
- 程序区：右击选择分支上的步或转换，单击【跳转 (J)】；
- 工具栏：；
- 快捷键：**Ctrl+J**。

如图 5.4-142 所示，在选择分支上插入跳转，跳转位置标示在跳转元素的左下角，默认显示 **InitStep** 表示默认跳转到初始步，选中跳转元素，双击跳转标号，可以写入需跳转到的步名或分支名称。

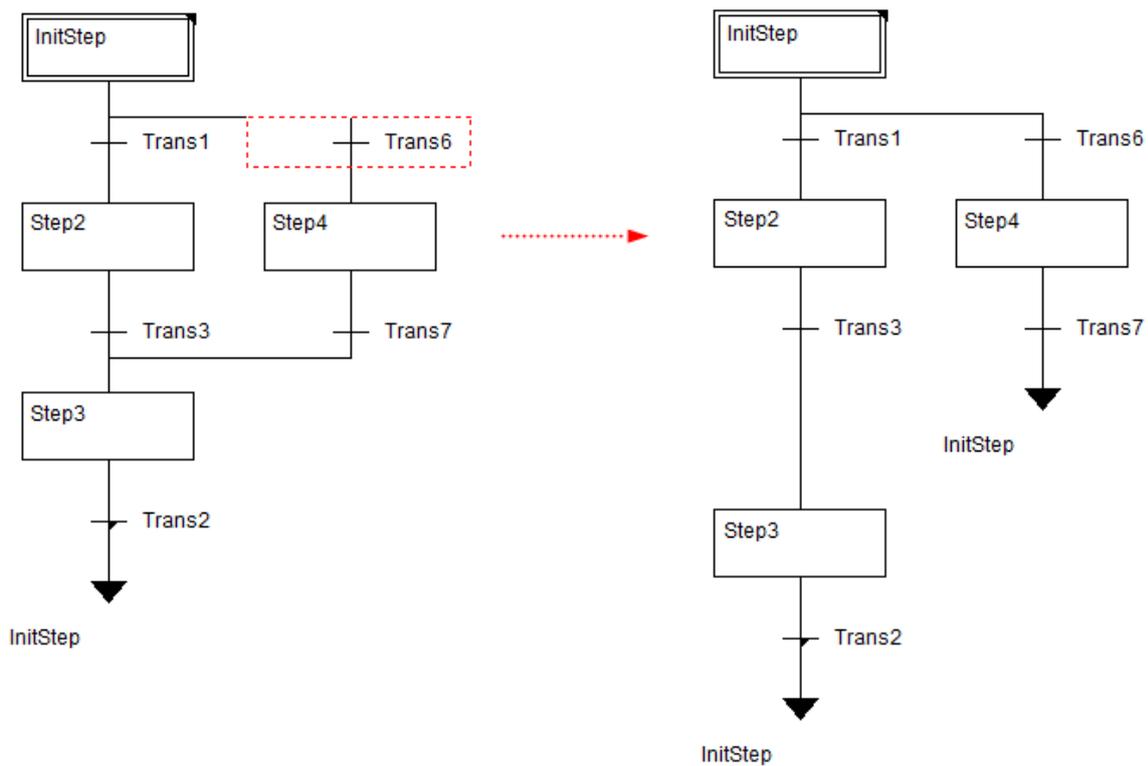


图 5.4-142 选择分支上插入跳转



- “跳转”必须接在“转换”之后，所以在“步”后插入跳转时，选择【转换-跳转】命令，而在转换后可以直接插入“跳转”。

(2) 删除跳转

如图 5.4-143 所示，选中跳转分支，执行【删除】命令。

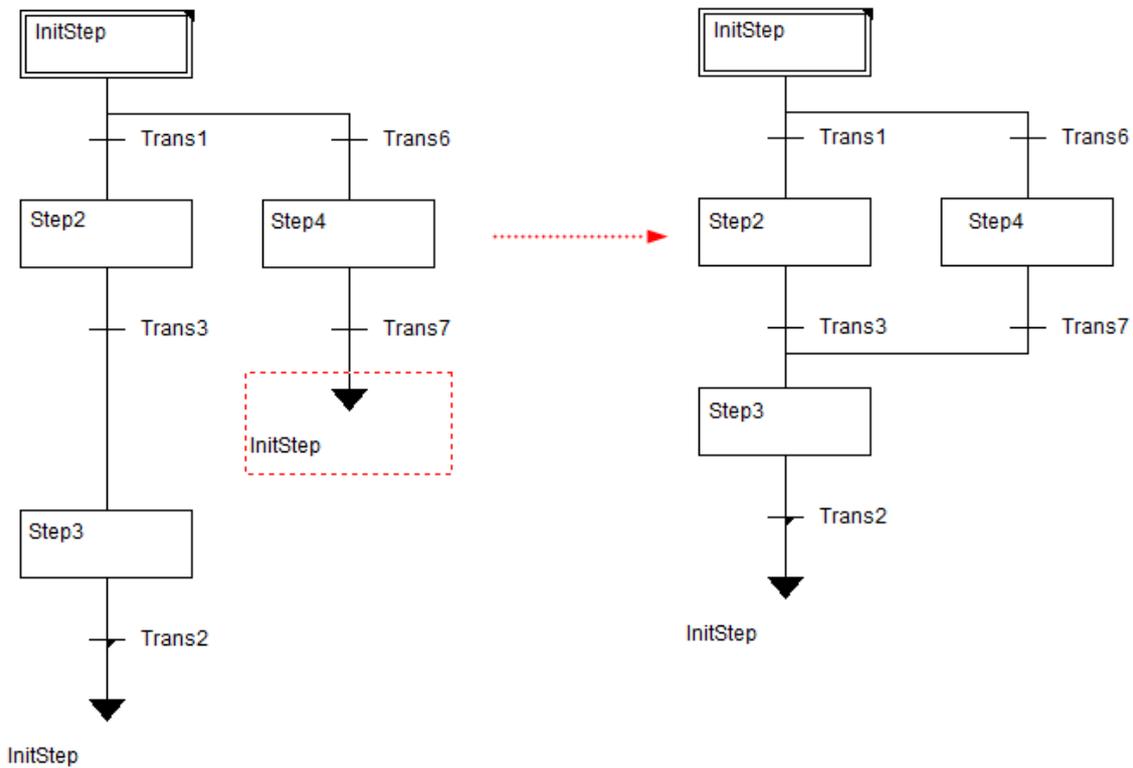


图 5.4-143 选择分支删除跳转

5.4.4.23 转换跳转

只能在并行分支上插入转换—跳转元素。跳转元素的跳转位置必须是步名或者并行分支标号名。转换—跳转元素插入到步之后。

(1) 增加转换—跳转

选中并行分支上的一个元素，通过以下方式增加转换—跳转：



- 菜单栏：单击【插入】—【转换—跳转 (R)】；
- 程序区：右击并行分支上的步或转换，单击【转换—跳转 (R)】；
- 工具栏：；
- 快捷键：**Ctrl+R**。

如图 5.4-144 所示，跳转位置标示在转换—跳转元素的左下角，默认显示 **InitStep**，表示默认跳转到初始步，选中跳转元素，双击跳转标号，可以写入需跳转到的步名或分支名称。

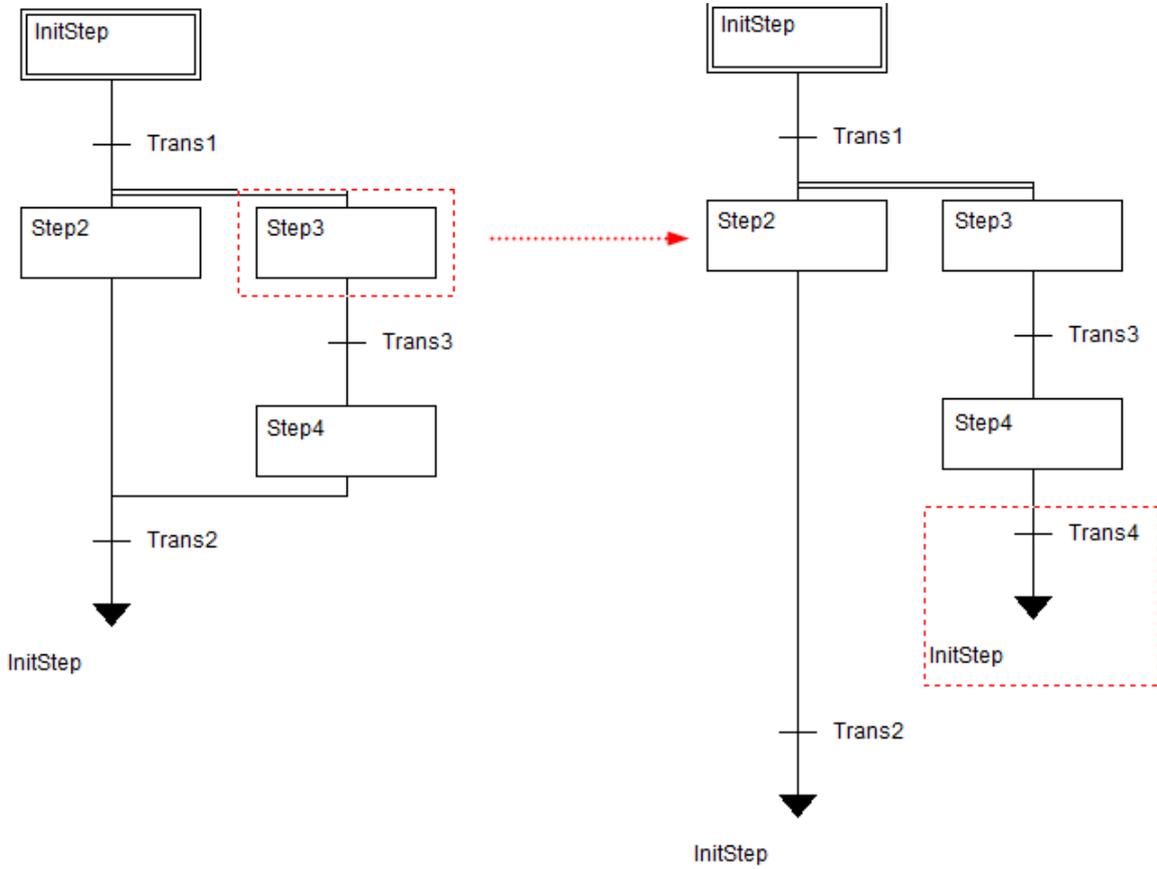


图 5.4-144 并行分支插入转换—跳转

(2) 删除转换—跳转

按下 **Shift** 键，选中转换—跳转，执行【删除】命令，转换—跳转被删除。如图 5.4-145 所示。

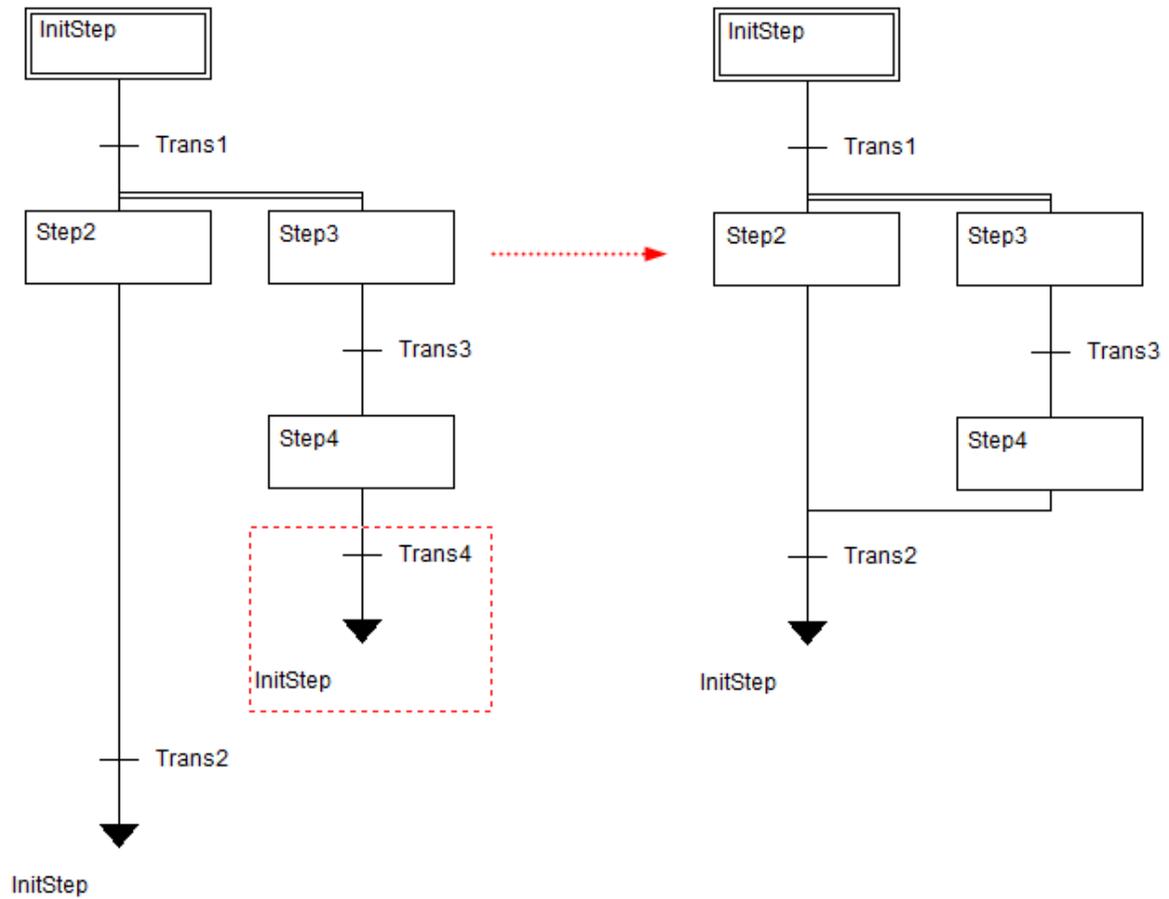


图 5.4-145 删除转换—跳转

5.4.4.24 粘贴到右边

将复制的单个步或转换，粘贴到某一步或转换的右边。



- 菜单栏：单击【附加】—【粘贴到右边】；
- 程序区：右击步或转换，单击【粘贴到右边】；
- 快捷键：**Ctrl+V**。

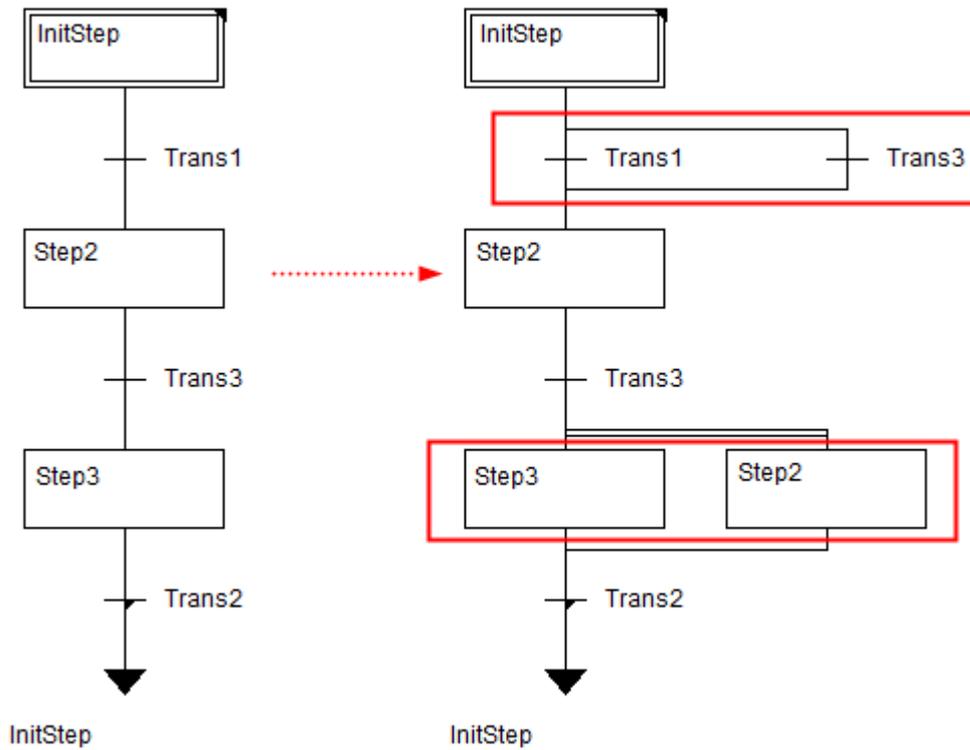


图 5.4-146 粘贴到右边

5.4.4.25 粘贴到后面

将复制的步-转换对，粘贴到某一转换的后面。

选中转换：



- 菜单栏：单击【附加】—【粘贴到后面】；
- 程序区：右击转换—【粘贴到后面】；
- 快捷键：**Ctrl+V**。

粘贴完成如图 5.4-147 所示。

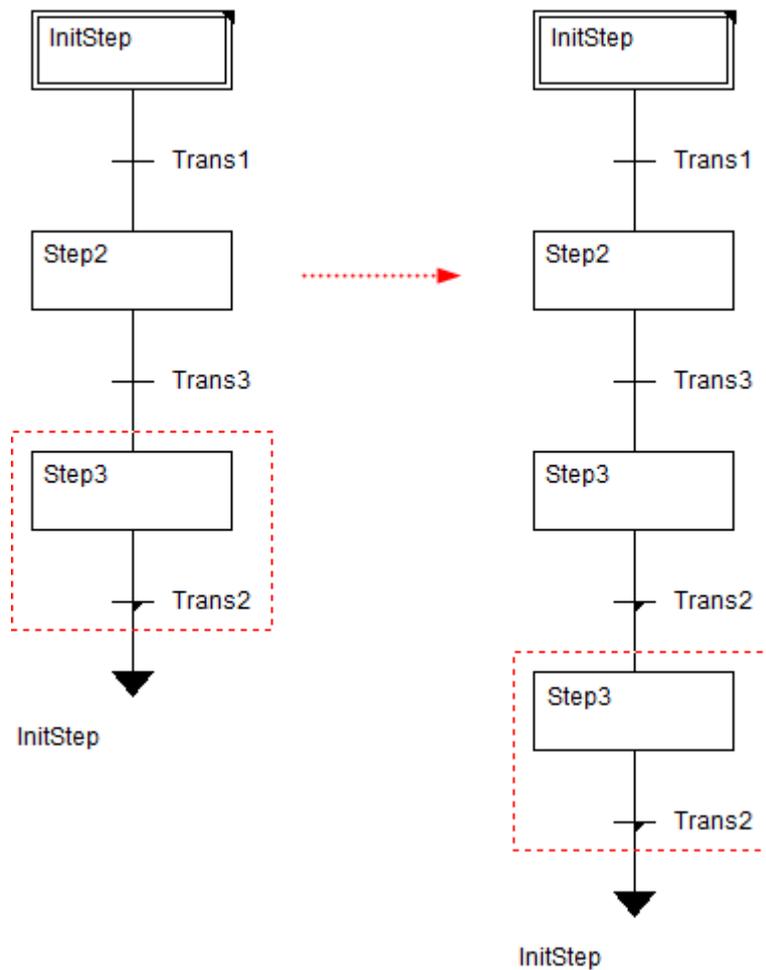


图 5.4-147 粘贴到后面

5.4.4.26 时间总览



- 菜单栏：单击【附加】—【时间总览 (T)】；
- 程序区：右击空白区域，单击【时间总览 (T)】。

在该窗口中，可以查看并修改本 SFC (PRG) 程序页中的所有步时间的设置，如图 5.4-148 所示。

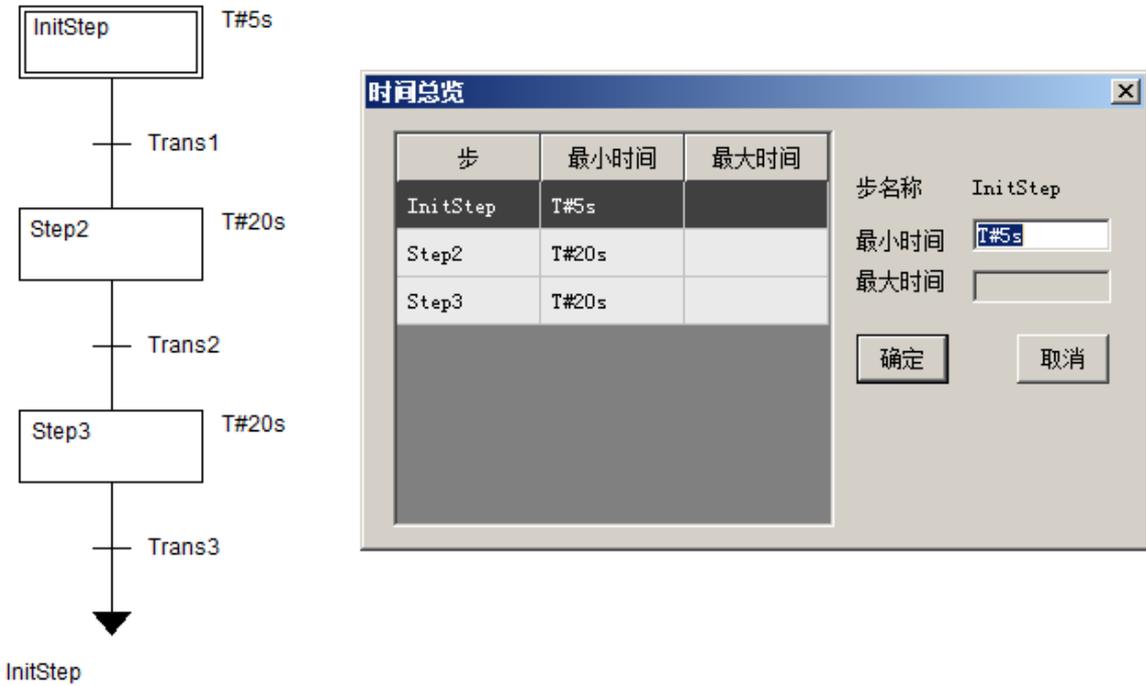


图 5.4-148 时间总览

5.4.4.27 选项



- 菜单栏：单击【附加】—【选项 (O)】；
- 程序区：右击空白区域，单击【选项 (O)】。

在该窗口中，可以设置步显示（标识框）的高度、长度、注释宽度等相关内容，如图 5.4-149 所示。

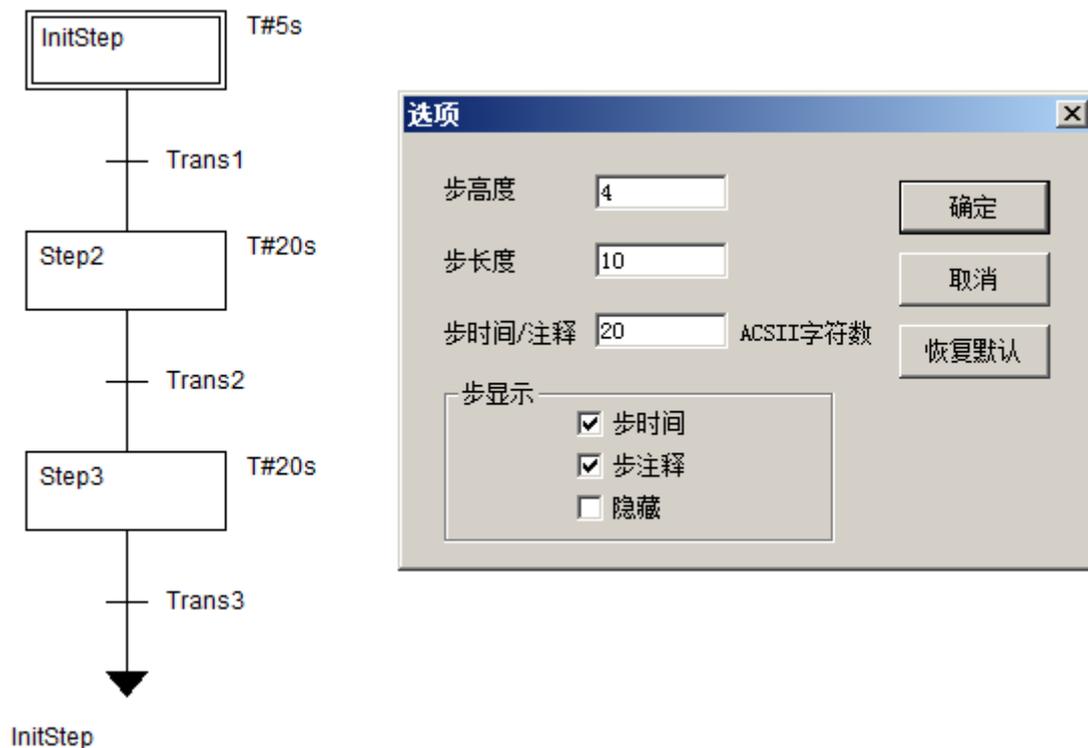


图 5.4-149 “选项”窗口

表 5.4-6 步参数有效范围

项名	有效设置范围
步高度	4~16 个单位长度 (10 个像素)
步长度	8~16 个单位长度 (10 个像素)
步时间/注释	每行最大可显示 4~20 字符

在【步的属性】中设置步的**最小时间**和**注释**后，在“选项”对话框的**步显示**中可以选择步的显示内容，缺省情况下，选择显示**步时间**；在需要显示每步的注释时，可以选择**步注释**，或者**隐藏**显示内容。如图 5.4-150 所示。

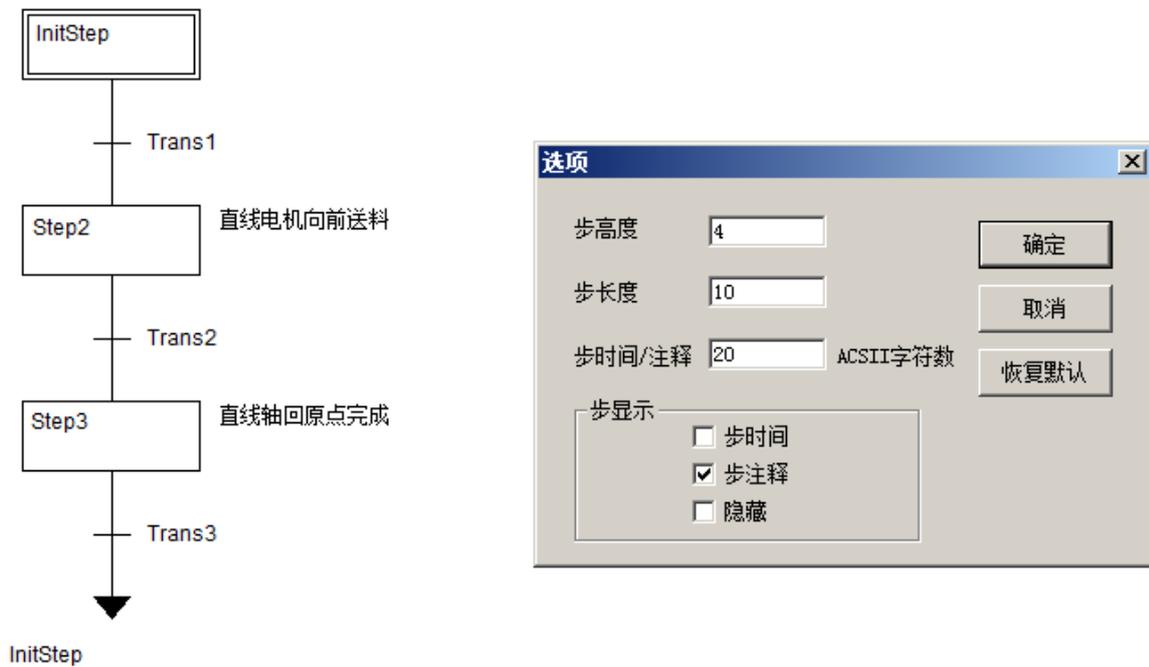


图 5.4-150 选择“步注释”

恢复默认：选择该按钮，可以将选项内容设置的当前值恢复为缺省值。

5.4.5 相关操作

5.4.5.1 字体缩放

四种语言编辑器中的变量字体支持放大与缩小，通过按住 **Ctrl+鼠标滚轮**，即可放大缩小。

5.5 数据和变量

控制和运算的操作对象是数据。数据有两个来源，一是来自生产过程或设备的物理参数(开关、按键、温度、压力、流量、长度和速度等等)，二是来自程序运算结果和人为设置的参数。在程序中，以变量的方式引入这些数据，每个变量都有相应的变量名，在程序中通过变量名进行引用，参与运算和控制。

本章叙述数据的含义、数据的存储、数据的类型、变量的类型、变量声明的格式、变量声明的规定、局部变量声明的方法和全局变量声明的方法。

5.5.1 数据

数据是工业系统中控制和运算的操作对象，数据存于 CPU 模块的内存储器。本节叙述数据的含义和数据的存储。

5.5.1.1 数据的含义

可编程控制器用于生产过程或设备的控制，其控制要求必须通过程序来实现，程序的基本元素是指令，指令由操作符和操作数构成。其中操作数分两类，一类是来自 I/O 模块的实际数据，例如行程开关的通(ON)或断(OFF)、温度的高低和流量的大小等；另一类是来自内部功能模块和程序的运算结果或中间结果。

数据的外部表现形式是变量名，例如行程开关的变量名为 SW1，当行程开关接通时，SW1=ON(或 1 或 TRUE)；当其断开时，SW1=OFF(或 0 或 FALSE)。数据的内部存储形式是 CPU 模块的存储器某位(bit)、字节(byte)或字(word)状态的变化。例如当行程开关接通时，相应的存储器某位(bit)为 1；当行程开关断开时，相应的存储器某位(bit)为 0；如图 5.5-1 所示。当行程开关 SW1、SW2 接通时，其状态通过 DI 模块送到 CPU 模块存储器，例如输入区 I321 字节的位 0=1、位 1=1；再通过逻辑梯形图(LD)运算，其运算结果 LA1 存入 CPU 模块存储器，例如输出区 Q321 字节的位 0=1；最后通过 DO 模块输出，使灯 LA1 亮。

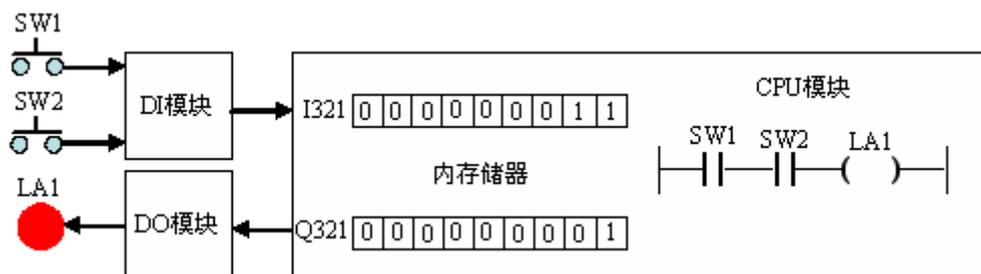


图 5.5-1 PLC 中的 DI 和 DO 数据

5.5.1.2 数据的存储

数据存储在 PLC 的 CPU 模块的存储器中，该存储器分为程序区和数据区，前者用于存储程序，后者用于存储数据。

1. 数据存储区

存储器中的数据区用于存储数据，该区分为输入区 (I)、输出区 (Q)、中间区 (M)、特殊寄存器区 (S)、随机区 (N) 和保持区 (R)。

右击工程管理树的根节点，选择【属性】命令，在弹出的“工程属性”对话框中单击红色圈出区域，可以查看该工程所能使用的各种区域的大小限制，以 MC 工程为例进行说明。如图 5.5-2 所示。

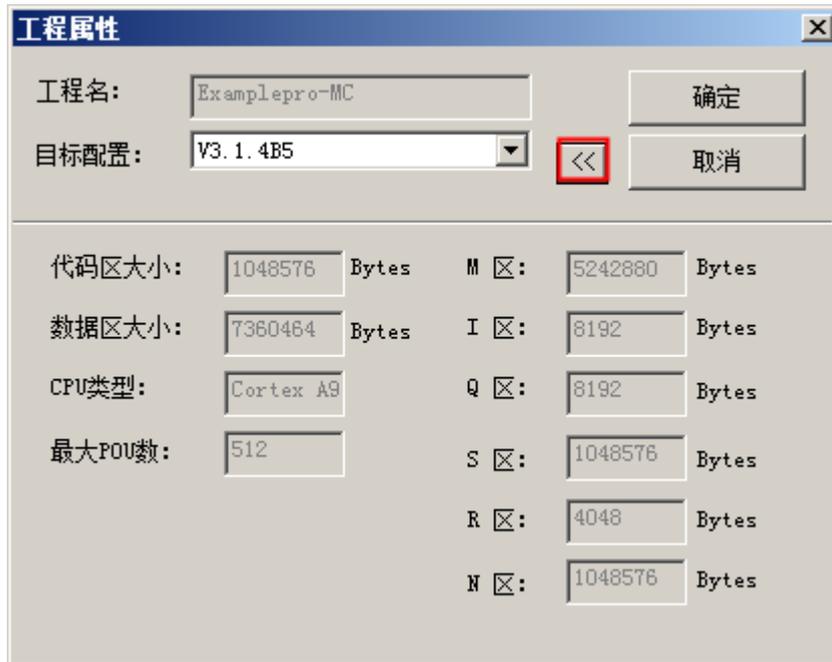


图 5.5-2 工程中数据区大小

各个区的主要功能如下：

(1) 输入区(I)

在每个扫描周期的首端，CPU 对输入点进行采样，并将采样值存储到内存储器的输入区。可按位(BIT)、字节(BYTE)、字(WORD)、双字(DWORD)来存取输入区。

(2) 输出区(Q)

在每个扫描周期的末端，CPU 将内存储器的输出区的数据传送到物理输出点上。可按位(BIT)、字节(BYTE)、字(WORD)、双字(DWORD)来存取输出区。

(3) 中间区(M)

中间区用来存储程序的中间结果、工作状态或其他控制信息。可按位(BIT)、字节(BYTE)、字(WORD)、双字(DWORD)来存取中间区。

(4) 随机区(N)

随机区用来存储用户所定义的未指定地址的变量。

N 存储区也属于 PLC 的中间寄存器区，用于存储和管理中间过程产生的数据和状态。与 M 存储区不同的是，N 存储区只能通过变量的方式来访问和调用。

N 存储区中的变量地址，是系统自动分配而用户无法指定的。N 区中的变量数据类型不仅有字节、字和双字，还有 REAL、TIME、INT 等其他众多数据类型。另外，除了数据变量外，定义的功能块变量也存储在 N 存储区。

N 存储区可以读写，可以被输入和强制。但 N 存储区的数据是不能掉电保持的。

(5) 保持区(R)

保持区存储掉电保持变量，若将数据或变量定义为保持型（掉电保护值为 TRUE），则当 CPU 模块掉电时，系统会自动保存该数据，待重新上电后，又会自动恢复该数据。

该区调用方式与 N 区一致，也是通过变量的方式访问，无法指定地址。

R 存储区变量可以读写，可以被输入和强制。

变量定义时，假如没有选择保持功能，则该变量存储在 N 区；若选择了保持功能，则该变量存储于 R 区，具有掉电保持功能。

(6) 特殊寄存器区 (S)

a. LE 平台特殊寄存器分配

表 5.5-1 特殊寄存器列表 (1)

地址	名称	属性	描述
SX0.0	First_Scan_On	只读	仅第一个扫描周期中接通为 ON，支持仿真模式
SX0.1	Clock_60s	只读	在 1 分钟的循环周期内，接通为 ON 30 秒，关断为 OFF 30 秒，支持仿真模式
SX0.2	Clock_1s	只读	在 1 秒钟的循环周期内，接通为 ON 0.5 秒，关断为 OFF 0.5 秒，支持仿真模式
SX0.3	Clock_Scan	只读	扫描循环时钟，一个周期接通为 ON，下一个周期关断为 OFF，支持仿真模式
SX0.4	Mode_Switch	只读	表明模式开关的当前位置：0=STOP（停止），1=RUN（运行）
SX0.5	Divide_By_0	只读	当尝试用零除时，置位为 1
SX0.6	Xmit0_Idle	只读	当发送器（本体 PS/2 口）处于空闲状态时置为 0，忙时置位为 1
SX0.7	Xmit1_Idle	只读	当发送器（本体接线端子）处于空闲状态时置为 0，忙时置位为 1
SX1.0	Xmit2_Idle	只读	当发送器（功能扩展板通讯口）处于空闲状态时置为 0，忙时置位为 1
SX1.1	/	只读	保留
SX1.2	I/O_Err	只读	当有任何 I/O 错误时，置为 1
SX1.3	/	只读	保留
SX1.4	/	只读	保留
SX1.5 SX1.7	~ /	不可读 写	保留
SB2~SB7	/	不可读 写	保留

表 5.5-2 特殊寄存器列表（2）

寄存器地址	名称	读写标志	描述
SB8~SB23	/	只读	多机互联 1~16 号站诊断信息

表 5.5-3 多机互联每个字节的具体含义

Bit7~Bit3	Bit2	Bit1	Bit0
故障站站号	保留	CRC 出错	超时

表 5.5-4 特殊寄存器列表（3）

寄存器地址	名称	读写标志	描述
SB24	/	读写	485-1\485 圆形接口--通讯参数
SB25	/	读写	485-1\485 圆形接口--Modbus 从站地址
SB26	/	读写	485-1\485 圆形接口--Modbus 从站应答延时时间
SB27	/	读写	保留
SB28	/	读写	保留
SB29	/	读写	485-2\485 接线端子--通讯参数
SB30	/	读写	485-2\485 接线端子--Modbus 从站地址
SB31	/	读写	485-2\485 接线端子--Modbus 从站应答延时时间
SB32	/	读写	485-2\485 接线端子--多机互联当前节点号
SB33	/	读写	485-2\485 接线端子--多机互联总节点数
SB34	/	读写	485-3\功能扩展板--通讯参数
SB35	/	读写	485-3\功能扩展板--Modbus 从站地址
SB36	/	读写	485-3\功能扩展板--Modbus 从站应答延时时间
SB37	/	读写	保留
SB38	/	读写	保留
SB39~SB95	/	不可读写	保留

备注：

S 区设置三个通讯口的通讯参数说明如下：

- LE 通用控制器的三个通讯端口分别命名为：485 圆形接口、485 接线端子、功能扩展板；LE 空调控制器的三个通讯端口分别命名为：485-1、485-2、485-3。
- 当 S 区参数为 0 时，【工程管理】—【硬件配置】编辑区的【通讯参数信息】中的设置有效；当 S 区参数不为 0 时，S 区设置参数有效，并且支持在线修改。

- SB24、SB29、SB34 分别设置三个通讯口的波特率、校验位、数据位、停止位，具体含义参见表 5.5-5。
- 485-1 口和上位机 AutoThink 通讯过程中,通过 S 区设置 COM1 口通讯参数时,退出 AutoThink 监视后设置的通讯参数生效。

表 5.5-5 通讯参数设置

校验		数据位			波特率			停止位	保留	设置说明
7	6	5	4	3	2	1	0			
0	0	0	0	0	0	0	0	0	以 AutoThink 软件界面设置为准	
0	1								偶校验	
1	0								无校验	
1	1								奇校验	
		0							8 位	
		1							7 位	
			0	0	0				115200bps	
			0	0	1				57600bps	
			0	1	0				38400bps	
			0	1	1				19200bps	
			1	0	0				9600bps	
			1	0	1				4800bps	
			1	1	0				2400bps	
			1	1	1				1200bps	
						0			1 位	
						1			2 位	

- SB25、SB30、SB35 表示 Modbus 从站地址，设置参数范围为 1~247。
- SB26、SB31、SB36 表示 modbus 从站应答延时时间，参数范围为 1~10,255，设置 1~10 对应含义为 Modbus 从站应答延时时间 1~10ms，设置为 255 含义为不延时。
- SB32、SB33 表示多机互联当前节点号和总结点数，参数范围为 1~16。
- 空调控制器 LE57XX 系列中 RS485-2 口，通过拨码开关设置通讯参数、modbus 从站地址、多机互联当前站号，不支持通过 S 区设置。

表 5.5-6 特殊寄存器列表（4）

寄存器地址	名称	读写标志	描述
-------	----	------	----

寄存器地址	名称	读写标志	描述
SD96	Last_Scan	只读	最后一次扫描周期时间
SD100	Minimum_Scan	只读	自运行以来记录的最短扫描时间
SD104	Maximum_Scan	只读	自运行以来记录的最长扫描时间

表 5.5-7 特殊寄存器列表（5）

寄存器地址	读写标志	描述
SD108	只读	LE5128（轴 1）绝对位置
SD112	只读	LE5128（轴 2）绝对位置
SD116	只读	LE5128（轴 3）绝对位置
SD120	只读	LE5128（轴 4）绝对位置
SD140	只读	LE5128（轴 5）第一路脉冲+方向高速输出/第一路 CW/CCW 高速输出绝对范围 计数范围（ $-2^{30}-1 \sim 2^{30}$ ） LE5108/LE5106/LE5104 第一路脉冲+方向高速输出/第一路 CW/CCW 高速输出绝对范围 计数范围（ $-2^{31}-1 \sim 2^{31}$ ） 注意：单方向长时间运行会导致绝对位置寄存器溢出
SD144	只读	LE5128（轴 6）第二路脉冲+方向高速输出绝对范围 计数范围（ $-2^{30}-1 \sim 2^{30}$ ） LE5108/LE5106/LE5104 第二路脉冲+方向高速输出绝对范围 计数范围（ $-2^{31}-1 \sim 2^{31}$ ）
SD148	只读	LE5128（轴 7）第三路脉冲+方向高速输出或第二路 CW/CCW 高速输出绝对范围 计数范围（ $-2^{30}-1 \sim 2^{30}$ ） LE5108 第三路脉冲+方向高速输出或第二路 CW/CCW 高速输出绝对范围 计数范围（ $-2^{31}-1 \sim 2^{31}$ ）
SD152	只读	LE5128（轴 8）第四路脉冲+方向高速输出绝对范围 计数范围（ $-2^{30}-1 \sim 2^{30}$ ） LE5108 第四路脉冲+方向高速输出绝对范围 计数范围（ $-2^{31}-1 \sim 2^{31}$ ）
SB140~SB155	不可读写	保留
SB156	只读	LE5108\LE5106 第一路脉冲+方向输出的方向：0 为正向脉冲，1 为负向脉冲
SB157	只读	LE5108\LE5106 第二路脉冲+方向输出的方向：0 为正向脉冲，1 为负向脉冲
SB158	只读	LE5108 第三路脉冲+方向输出的方向：0 为正向脉冲，1 为负向脉冲
SB159	只读	LE5108 第四路脉冲+方向输出的方向：0 为正向脉冲，1 为负向脉冲

寄存器地址	读写标志	描述
SB160~SB269	不可读写	保留

备注：

SD140、SD144、SD148、SD152 为输出轴的绝对位置。PWM、PTO_H 两个指令只输出脉冲，不修改轴的绝对位置；其他高速输出指令在输出脉冲的同时修改轴的绝对位置。

表 5.5-8 特殊寄存器列表（6）

寄存器地址	读写标志	描述
SW270~SW298	读写	多机互联当前节点数据
SW300~SW328	只读	多机互联 1 号站数据
SW330~SW358	只读	多机互联 2 号站数据
SW360~SW388	只读	多机互联 3 号站数据
SW390~SW418	只读	多机互联 4 号站数据
SW420~SW448	只读	多机互联 5 号站数据
SW450~SW478	只读	多机互联 6 号站数据
SW480~SW508	只读	多机互联 7 号站数据
SW510~SW538	只读	多机互联 8 号站数据
SW540~SW568	只读	多机互联 9 号站数据
SW570~SW598	只读	多机互联 10 号站数据
SW600~SW628	只读	多机互联 11 号站数据
SW630~SW658	只读	多机互联 12 号站数据
SW660~SW688	只读	多机互联 13 号站数据
SW690~SW718	只读	多机互联 14 号站数据
SW720~SW748	只读	多机互联 15 号站数据
SW750~SW778	只读	多机互联 16 号站数据

表 5.5-9 特殊寄存器列表（7）

寄存器地址	读写标志	描述
SB780~SB789	只读	背板概要诊断信息
SB790~SB809	只读	CPU 模块诊断信息
SB810~SB819	只读	第 1 个扩展模块诊断信息
SB820~SB829	只读	第 2 个扩展模块诊断信息

寄存器地址	读写标志	描述
SB830~SB839	只读	第 3 个扩展模块诊断信息
SB840~SB849	只读	第 4 个扩展模块诊断信息
SB850~SB859	只读	第 5 个扩展模块诊断信息
SB860~SB869	只读	第 6 个扩展模块诊断信息
SB870~SB879	只读	第 7 个扩展模块诊断信息
SB880~SB889	只读	第 8 个扩展模块诊断信息
SB890~SB899	只读	第 9 个扩展模块诊断信息
SB900~SB909	只读	第 10 个扩展模块诊断信息
SB910~SB919	只读	第 11 个扩展模块诊断信息
SB920~SB929	只读	第 12 个扩展模块诊断信息
SB930~SB939	只读	第 13 个扩展模块诊断信息
SB940~SB949	只读	第 14 个扩展模块诊断信息
SB950~SB959	只读	第 15 个扩展模块诊断信息
SB960~SB969	只读	第 16 个扩展模块诊断信息
SB970~SB979	只读	第 17 个扩展模块诊断信息
SB980~SB989	只读	第 18 个扩展模块诊断信息
SB990~SB999	只读	第 19 个扩展模块诊断信息
SB1000~SB1009	只读	第 20 个扩展模块诊断信息
SB1010~SB1019	只读	第 21 个扩展模块诊断信息
SB1020~SB1023	不可读写	保留

备注：

系统为 CPU 模块分配了 20 个字节，为每个扩展模块分配了 10 个字节，用于存放该模块的详细诊断信息。

■ CPU 模块

第 1 字节：模块状态，如图 5.5-3 所示。

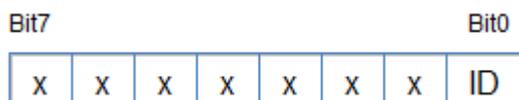


图 5.5-3 模块状态信息

Bit0=0，模块 ID 配置正确；Bit0=1，模块 ID 配置有误。



- 诊断信息中“x”表示保留位，统一置为“0”，下同。

若 CPU 模块有模拟量通道，从第 2 个字节开始为模拟量通道的诊断信息，参见模拟量模块诊断信息。

■ 数字量扩展模块

数字量模块包含 1 个字节的诊断信息，详细信息如图 5.5-4 所示。

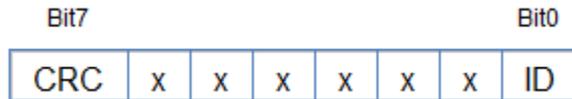


图 5.5-4 模块诊断信息

Bit0=0，模块 ID 配置正确；Bit0=1，模块 ID 配置有误。

Bit7=0，背板通讯数据 CRC 校验正确；Bit7=1，背板通讯数据 CRC 校验有误。

第 2~10 字节，保留。

■ 模拟量输入模块

第 1 字节：模块状态，如图 5.5-5 所示。

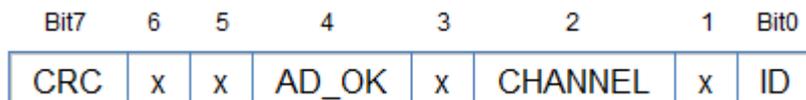


图 5.5-5 模块状态信息

Bit0=0，模块 ID 配置正确；Bit0=1，模块 ID 配置有误。

Bit2=0，无通道故障；Bit2=1，有通道故障。（所有通道都无故障时，则认为无通道故障。）

Bit4=0，AD 芯片正常；Bit4=1，AD 芯片异常。

Bit7=0，背板通讯数据 CRC 校验正确；Bit7=1，背板通讯数据 CRC 校验有误。

第 2 字节以后为通道状态：

每 4 位表示一个通道的状态信息，如图 5.5-6 所示。

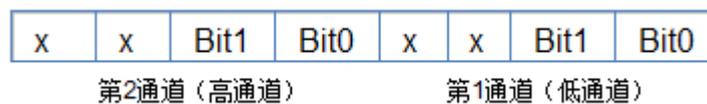


图 5.5-6 通道状态信息

Bit0=0, 正常; Bit0=1, 超量程下线。(仅对 4-20mA 有效, 当检测到输入电流值小于 4mA, 认为超量程下线)

Bit1=0, 正常; Bit1=1, 超量程上线。(对 4-20mA, 0~20mA, 0~10V 均有效)。

■ 模拟量输出模块

第 1 字节: 模块状态, 如图 5.5-7 所示。

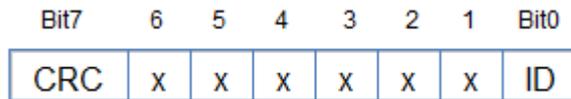


图 5.5-7 模块状态信息

Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置有误。

Bit7=0, 背板通讯数据 CRC 校验正确; Bit7=1, 背板通讯数据 CRC 校验有误。

第 2 字节以后为通道状态, 每 4 位表示一个通道的状态信息, 如图 5.5-8 所示。

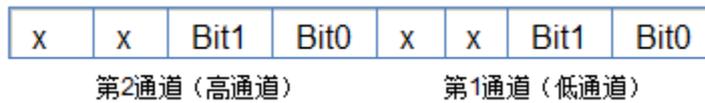


图 5.5-8 通道状态信息

Bit0=0, 正常; Bit0=1, 对地短路 (电压)。

Bit1=0, 正常; Bit1=1, 对地断路 (电流)。

■ 热电偶模块

第 1 字节: 模块状态, 同模拟量输入模块。

第 2 字节以后为通道状态, 每 4 位表示一个通道的状态信息, 如图 5.5-9 所示。



图 5.5-9 通道状态信息

Bit0=0, 正常; Bit0=1, 断偶或超量程。

■ 热电阻模块

第 1 字节: 模块状态, 同模拟量输入模块。

第 2 字节以后为通道状态, 每 4 位表示一个通道的状态信息, 如图 5.5-10 所示。

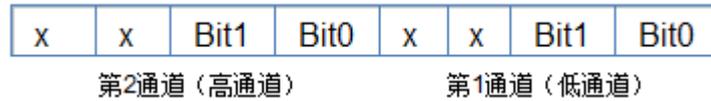


图 5.5-10 通道状态信息

Bit0=0, 正常; Bit0=1, 短路。

Bit1=0, 正常; Bit1=1, 断线。

■ DP 通讯模块

DP 模块只包含 1 个字节的模块状态信息, 无通道状态诊断信息, 详细信息如下:

第 1 字节: 模块状态, 如图 5.5-11 所示。

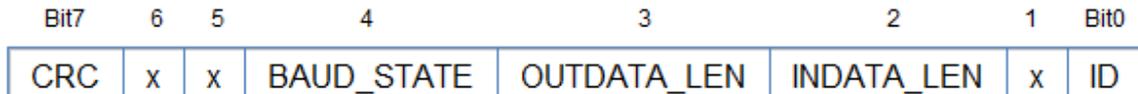


图 5.5-11 模块状态信息

Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置有误。

Bit2=0, AutoThink 中对 DP 模块配置的输入数据字节数与 DP-Master 组态软件中对 DP 模块配置的输出数据字节数匹配; Bit2=1, 上述字节数不匹配。

Bit3=0, AutoThink 中对 DP 模块配置的输出数据字节数与 DP-Master 组态软件中对 DP 模块配置的输入数据字节数匹配; Bit3=1, 上述字节数不匹配。

Bit4=0, 检测到波特率; Bit4=1, 未检测到波特率。

Bit7=0, 背板通讯数据 CRC 校验正确; Bit7=1, 背板通讯数据 CRC 校验有误。

■ 以太网通讯模块

以太网模块只包含 1 个字节的模块状态信息, 无通道状态诊断信息, 详细信息如下:

第 1 字节: 模块状态, 如图 5.5-12 所示。

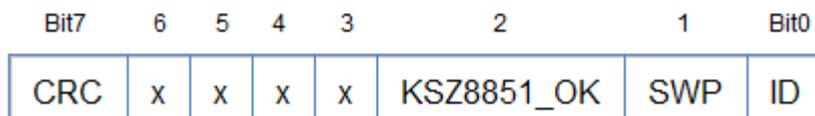


图 5.5-12 模块状态信息

Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置有误。

Bit1=0, 有网络数据交换; Bit1=1 无网络数据交换。

Bit2=0, KSZ8851 正常; Bit2=1, KSZ8851 异常。

Bit7=0, 背板通讯数据 CRC 校验正确; Bit7=1, 背板通讯数据 CRC 校验有误。

■ GPRS 通讯模块

GPRS 模块只包含 1 个字节的模块状态信息, 无通道状态诊断信息, 详细信息如下:

第 1 字节: 模块状态, 如图 5.5-13 所示。

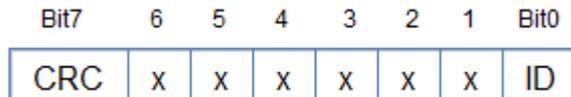


图 5.5-13 模块状态信息

Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置有误。

Bit7=0, 背板通讯数据 CRC 校验正确; Bit7=1, 背板通讯数据 CRC 校验有误。

SB780~SB789 为模块概要诊断信息, 目前只用了前四个字节 SB780~SB783, 具体含义如表 5.5-10 所示。

表 5.5-10 特殊寄存器列表 (8)

寄存器地址	描述
SB780.Bit0	CPU 是否异常
SB780.Bit1~7	第 1~7 个扩展模块是否异常
SB781.Bit0~7	第 8~15 个扩展模块是否异常
SB782.Bit0	第 16 个扩展模块是否异常
SB782.Bit1	第 17 个扩展模块是否异常
SB782.Bit2	第 18 个扩展模块是否异常
SB782.Bit3	第 19 个扩展模块是否异常
SB782.Bit4	第 20 个扩展模块是否异常
SB782.Bit5	保留
SB782.Bit6	保留
SB782.Bit7	存储卡是否异常
SB783.Bit0~6	保留
SB783.Bit7	组态模块个数大于定义空间

b. MC 平台特殊寄存器分配



- MC 平台模块诊断信息中除概要诊断信息外, 其余暂不支持。

MC 平台中该区为轴参数变量预定义区及诊断信息，详细诊断信息如表 5.5-11 所示。

表 5.5-11 特殊寄存器列表（9）

寄存器地址	名称	读写标志	描述
SB10240~SB10249	/	只读	背板概要诊断信息
SB10250~SB10259	/	只读	第 1 个扩展模块诊断信息
SB10260~SB10269	/	只读	第 2 个扩展模块诊断信息
SB10270~SB10279	/	只读	第 3 个扩展模块诊断信息
SB10280~SB10289	/	只读	第 4 个扩展模块诊断信息
SB10290~SB10299	/	只读	第 5 个扩展模块诊断信息
SB10300~SB10309	/	只读	第 6 个扩展模块诊断信息
SB10310~SB10319	/	只读	第 7 个扩展模块诊断信息
SB10320~SB10329	/	只读	第 8 个扩展模块诊断信息
SB10330~SB10339	/	只读	第 9 个扩展模块诊断信息
SB10340~SB10349	/	只读	第 10 个扩展模块诊断信息
SB10350~SB10359	/	只读	第 11 个扩展模块诊断信息
SB10360~SB10369	/	只读	第 12 个扩展模块诊断信息
SB10370~SB10379	/	只读	第 13 个扩展模块诊断信息
SB10380~SB10389	/	只读	第 14 个扩展模块诊断信息
SB10390~SB10399	/	只读	第 15 个扩展模块诊断信息
SB10400~SB10409	/	只读	第 16 个扩展模块诊断信息
SB10410~SB10419	/	只读	第 17 个扩展模块诊断信息
SB10420~SB10429	/	只读	第 18 个扩展模块诊断信息
SB10430~SB10439	/	只读	第 19 个扩展模块诊断信息
SB10440~SB10449	/	只读	第 20 个扩展模块诊断信息
SB10450~SB10459	/	只读	第 21 个扩展模块诊断信息
SB10460~SB10469	/	只读	第 22 个扩展模块诊断信息
SB10470~SB10479	/	只读	第 23 个扩展模块诊断信息
SB10480~SB10489	/	只读	第 24 个扩展模块诊断信息
SB10490~SB10499	/	只读	第 25 个扩展模块诊断信息
SB10500~SB10509	/	只读	第 26 个扩展模块诊断信息
SB10510~SB10519	/	只读	第 27 个扩展模块诊断信息
SB10520~SB10529	/	只读	第 28 个扩展模块诊断信息
SB10530~SB10539	/	只读	第 29 个扩展模块诊断信息
SB10540~SB10549	/	只读	第 30 个扩展模块诊断信息

寄存器地址	名称	读写标志	描述
SB10550~SB10559	/	只读	第 31 个扩展模块诊断信息

备注:

■ 数字量扩展模块诊断信息

数字量模块包含 1 个字节的诊断信息，详细信息如图 5.5-14 所示。

Bit7	6	5	4	3	2	1	Bit0
CRC	×	×	×	×	×	×	ID

图 5.5-14 诊断信息

第 0 位: Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置错。(指本模块 ID 与 CPU 模块收到后返回的模块 ID 是否相同。)

第 7 位: Bit7=0, 扩展模块读取到的数据 CRC 正确; Bit7=1, 扩展模块读取到的数据 CRC 错误。

其它: 保留位, 统一置为“0”。

说明: 诊断信息中“×”表示保留位, 下同。

说明: 诊断信息中“CRC”位与“ID”位的诊断工作均由主控完成, 下同。

■ 模拟量扩展模块诊断信息

□ 模拟量输入模块

模拟量输入模块包含 1 个字节的“模块状态”诊断信息与多个字节（具体字节数与通道数有关，每 4 位表示一个通道的状态信息）的“通道状态”诊断信息，详细信息如下：

第 1 字节: 模块状态, 如图 5.5-15 所示。

Bit7	6	5	4	3	2	1	Bit0
CRC	×	×	AD_OK	×	CHANNEL	×	ID

图 5.5-15 模块状态信息

第 0 位: Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置错。(指本模块 ID 与 CPU 模块收到后返回的模块 ID 是否相同。)

第 2 位: Bit2=0, 无通道错误; Bit2=1, 有通道错误。(所有通道都无错误时, 则认为无通道错误。)

第 4 位: Bit4=0, AD 芯片正常; Bit4=1, AD 芯片异常。

第 7 位: Bit7=0, 扩展模块读取到的数据 CRC 正确; Bit7=1, 扩展模块读取到的数据 CRC 错误。

其它: 保留位, 统一置为“0”。

第 2 字节以后为通道状态:

每 4 位表示一个通道的状态信息, 如图 5.5-16 所示。不同的模块类型有不同的通道状态信息。

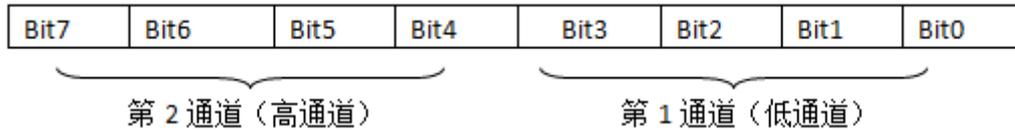


图 5.5-16 通道状态信息

第 0 位: Bit0=0, 正常; Bit0=1, 超量程下线。(仅对 4-20mA 量程有效, 当检测到输入电流值小于 4mA, 认为超量程下线)

第 1 位: Bit1=0, 正常; Bit1=1, 超量程上线。(对 4-20mA, 0~20mA, 0~10V 均有效)

其它: 保留, 保留位置为“0”。

□ 模拟量输出模块

模拟量输入模块只包含 1 个字节的“模块状态”诊断信息, 与多个字节(具体字节数与通道数有关, 每 4 位表示一个通道的状态信息)的“通道状态”诊断信息, 详细信息如下:

第 1 字节: 模块状态, 如图 5.5-17 所示。

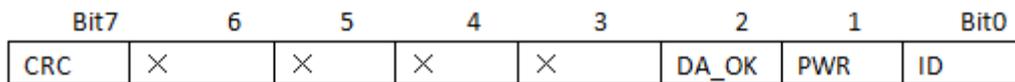


图 5.5-17 模块状态信息

第 0 位: Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置错。

第 1 位: Bit1=0, 外供电 DC 24V 电源正常, Bit1=1, 外供 DC 24V 电源异常。

第 2 位: Bit2=0, DA 正常; Bit2=1, DA 异常。(根据实际情况确定是否添加此诊断位)

第 7 位: Bit7=0, 扩展模块读取到的数据 CRC 正确; Bit7=1, 扩展模块读取到的数据 CRC 错误。

其它: 保留位, 统一置为“0”。

第 2 字节以后为通道状态:

每 4 位表示一个通道的状态信息，如图 5.5-18 所示。不同的模块类型有不同的通道状态信息。

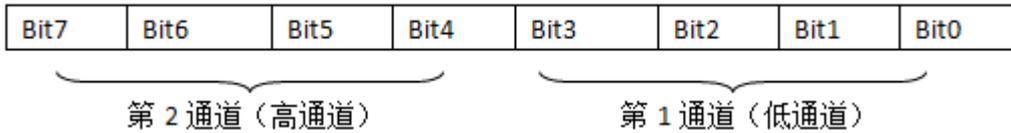


图 5.5-18 通道状态信息

第 0 位：Bit0=0，正常；Bit0=1，对地短路（电压）。

第 1 位：Bit1=0，正常；Bit1=1，对地断路（电流）。

其它：保留，保留位置为“0”。

□ 模拟量混合模块

模拟量混合模块的诊断信息即模拟量输入模块与模拟输出模块的诊断信息的整合。

□ 热电偶模块

热电偶模块包含 1 个字节的“模块状态”诊断信息与多个字节（具体字节数与通道数有关，每 4 位表示一个通道的状态信息）的“通道状态”诊断信息，详细信息如下：

第 1 字节：模块状态，如图 5.5-19 所示。



图 5.5-19 模块状态信息

第 0 位：Bit0=0，模块 ID 配置正确；Bit0=1，模块 ID 配置错。（指本模块 ID 与 CPU 模块收到后返回的模块 ID 是否相同。）

第 2 位：Bit2=0，无通道错误；Bit2=1，有通道错误。（所有通道都无错误时，则认为无通道错误。）

第 4 位：Bit4=0，AD 芯片正常；Bit4=1，AD 芯片异常。

第 7 位：Bit7=0，扩展模块读取到的数据 CRC 正确；Bit7=1，扩展模块读取到的数据 CRC 错误。

其它：保留位，统一置为“0”。

第 2 字节以后为通道状态：

每 4 位表示一个通道的状态信息，如图 5.5-20 所示。



图 5.5-20 通道状态信息

第 0 位: Bit0=0, 正常; Bit0=1, 断偶或超量程。

其它: 保留位, 统一置为“0”。

□ 热电阻模块

热电阻模块包含 1 个字节的“模块状态”诊断信息与多个字节(具体字节数与通道数有关, 每 4 位表示一个通道的状态信息)的“通道状态”诊断信息, 详细信息如下:

第 1 字节: 模块状态, 如图 5.5-21 所示。

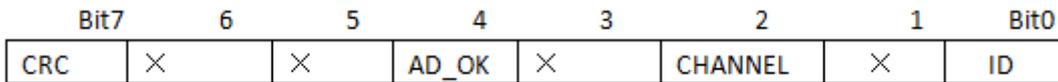


图 5.5-21 模块状态信息

第 0 位: Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置错。(指本模块 ID 与 CPU 模块收到后返回的模块 ID 是否相同。)

第 2 位: Bit2=0, 无通道错误; Bit2=1, 有通道错误。(所有通道都无错误时, 则认为无通道错误。)

第 4 位: Bit4=0, AD 芯片正常; Bit4=1, AD 芯片异常。

第 7 位: Bit7=0, 扩展模块读取到的数据 CRC 正确; Bit7=1, 扩展模块读取到的数据 CRC 错误。

其它: 保留位, 统一置为“0”。

第 2 字节以后为通道状态:

每 4 位表示一个通道的状态信息, 如图 5.5-22 所示。



图 5.5-22 通道状态信息

第 0 位: Bit0=0, 正常; Bit0=1, 短路。

第 1 位: Bit0=0, 正常; Bit0=1, 断线。

其它: 保留位, 统一置为“0”。

SB10240~SB10249 为模块背板诊断信息, 目前只用了前四个字节 SB10240~SB10243, 具体含义如表 5.5-12 所示。

表 5.5-12 特殊寄存器列表 (10)

寄存器地址	描述
SB10240.Bit0	预留
SB10240.Bit1~7	第 1~7 个扩展模块是否异常
SB10241.Bit0~7	第 8~15 个扩展模块是否异常
SB10242.Bit0~7	第 16~23 个扩展模块是否异常
SB10243.Bit0~7	第 24~31 个扩展模块是否异常

2. 数据存储格式-地址

数据存储格式分为字节、字、双字、位这 4 种方式, 以中间区 (M) 为例, 叙述数据存储格式。

(1) 字节数据

字节数据的存储格式如图 5.5-23 所示。其中 MSB 表示数据的最高位(位 7), LSB 表示数据的最低位(位 0); “%”表示地址, “M”为存储器标识, “B”表示字节数据, 占 8 位。

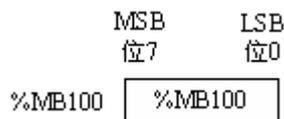


图 5.5-23 字节数据

(2) 字数据

字数据的存储格式如图 5.5-24 所示。其中 MSB 表示数据的最高位(位 15), LSB 表示数据的最低位(位 0); “%”表示地址, “M”为存储器标识, “W”表示字数据, 占 16 位。

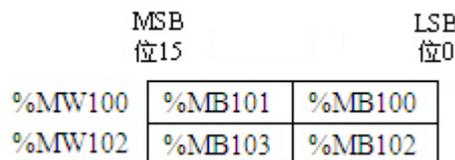


图 5.5-24 字数据

(3) 双字数据

双字数据的存储格式如图 5.5-25 所示。其中 MSB 表示数据的最高位(位 31)，LSB 表示数据的最低位(位 0)；“%”表示地址，“M”为存储器标识，“D”表示双字数据，占 32 位。

	MSB 位31			LSB 位0
%MD100	%MB103	%MB102	%MB101	%MB100
%MD104	%MB107	%MB106	%MB105	%MB104

图 5.5-25 双字数据

(4) 位数据

位数据的存储格式如图 5.5-26 所示。“%”表示地址，“M”为存储器标识，“X”表示单个位，即占 1 位。

%MX1.0 位0
⋮
%MX1.7 位7

图 5.5-26 位数据

5.5.2 数据类型

在数据类型节点下，显示系统定义的数据类型和自定义数据类型，以树状图的方式对数据类型进行管理，可以添加自定义数据类型，如图 5.5-27 所示。

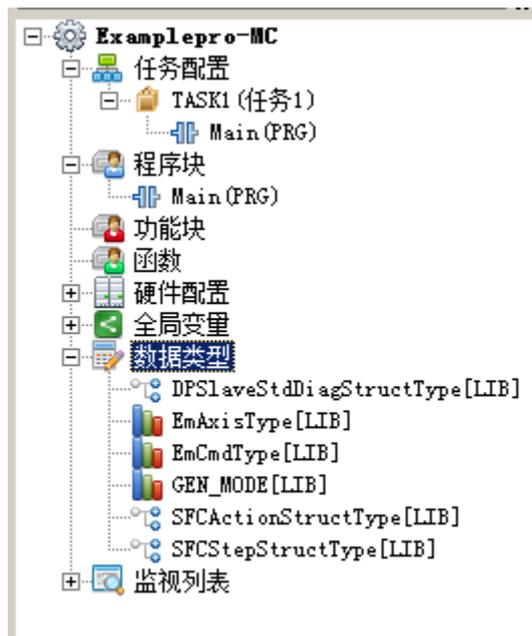


图 5.5-27 数据类型树状图

5.5.2.1 添加数据类型



- 工程管理树：右击【数据类型】—【添加类型】。



图 5.5-28 “添加类型”对话框

在**数据类型名称**框中输入自定义的数据类型名称，该名称定义原则同变量组名定义，选择数据类型，单击**确定**按钮，确认定义完成。

结构体关联的图符为，枚举型关联的图符为.

软件已定义的数据类型的变量名后标识为“LIB”，即提供给库使用的数据类型；自定义的数据类型的变量名后无标识符。



图 5.5-29 数据类型节点

鼠标双击变量名，在工作区域展开同名的编辑窗口，如图 5.5-30 所示。



图 5.5-30 数据类型编辑窗口

该软件支持的数据类型(或变量类型)分为标准数据类型和自定义数据类型。其中标准数据类型又分为布尔型、字节型、字型、双字型、长字型、短整型、无符号短整型、整型、长整型、无符号整型、无符号长整型、32 位整型、无符号 32 位整型、实型、时间型、数组和指针。自定义数据类型分为结构体和枚举。

5.5.2.2 标准数据类型

该软件支持的标准数据类型如表 5.5-13 所示，此外还有实型、时间型、数组、指针、长实型、字符串。

表 5.5-13 数据类型描述

类型	类型名称	数据下限	数据上限	存储位数
BOOL	布尔型	0	1	1bit
BYTE	字节型	0	255	8bit
WORD	字型	0	65535	16bit
DWORD	双字型	0	4294967295	32bit
LWORD	长字型	0	$2^{64}-1$	64bit
SINT	短整型	-128	127	8bit
USINT	无符号短整型	0	255	8bit
INT	整型	-32768	32767	16bit
LINT	长整型	-2^{63}	$2^{63}-1$	64bit

类型	类型名称	数据下限	数据上限	存储位数
UINT	无符号整型	0	65535	16bit
ULINT	无符号长整型	0	$2^{64}-1$	64bit
DINT	32 位整型	-2147483648	2147483647	32bit
UDINT	无符号 32 位整型	0	4294967295	32bit
REAL	实型	-3.402823466E+38	-3.402823466E+38	32bit
		1.175494351E-38	3.402823466E+38	
LREAL	长实型	-1.7976931348623158e+308	-2.2250738585072014e-308	64bit
		-1.175494351E-38	-1.175494351E-38	

1. 布尔(BOOL)型数据

布尔型数据为逻辑量，占用存储器 1 位(bit)，其状态为“0”或“1”、“TRUE”或“FALSE”、“真”或“假”。

2. 整型数据

整型数据分为 BYTE、WORD、LWORD、DWORD、SINT、USINT、INT、LINT、UINT、ULINT、DINT 和 UDINT。

整型数据的特点是无小数位。

■ 整数常量识别规则

AutoThink 中整数常量识别的顺序如下：

BYTE(0~255)->SINT(-128~127)->USINT(0~255)->WORD(0~65535)->INT(-32768~32767)->UINT(0~65535)->DWORD(0~4294967295)->DINT(-2147483648~2147483647)->UDINT(0~4294967295)->LWORD(0~ $2^{64}-1$)->LINT(- 2^{63} ~ $2^{63}-1$)->ULINT(0~ $2^{64}-1$)

即正数优先被识别为无符号型数据，图 5.5-31 更形象的说明了这一点。

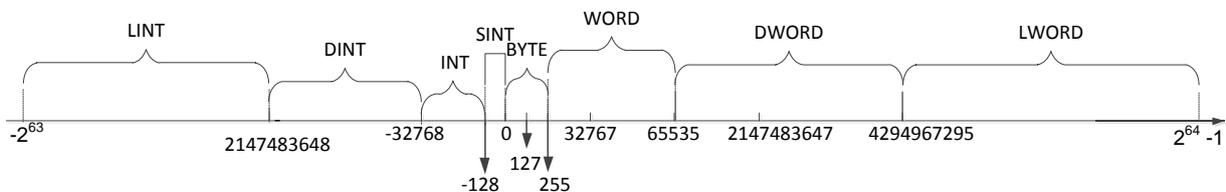


图 5.5-31 整数常量识别规则

■ 类型优先级

AutoThink 中整型数据类型优先级由低到高如下：

SINT(4)->BYTE(5)->USINT(6)->INT(7)->WORD(8)->UINT(9)->DINT(11)->DWORD(12)->UDINT(13)->LWORD(14)->LINT(15)->ULINT(16)

应用：

G1:real;

G1:=300+(-350);

300 识别为 WORD 型，-350 识别为 INT 型。WORD 型优先级高，结果按 WORD 行处理，运算结果 G1 为 65486。

■ 类型提升

LE 系列 PLC 单片机型号为 XE164，机器字长 16 位。因此，16 位数据，如 WORD、INT、UINT 不提升，8 位数据如 BYTE、SINT、USINT 提升为 INT 型。

如：

G1:real;

G1:=100 - 200;

100，200 为 BYTE 型数，提升为 INT 型运算再转给 G1，结果为-100。



- 当范围较大的类型转换为较小的类型时，可能丢失信息。



用户组态时注意以下几点：

- AutoThink 中正数整型常量被识别为无符号型数据，无符号型优先级高。
- 不要使用有符号数与无符号数共同参与一个基本运算，除非运算结果符合预期。
- 对于多个输入的基本指令，输入不要全为常量，除非结果符合预期。
- 对于结果可能为负数的基本运算，若使用 16 位整数作为输入，最好使用 INT 型变量，若使用 32 位整数作为输入，最好使用 DINT 型变量。

3. 实型(REAL)数据

实型数据的特点是有小数位，既可表示成整数和小数形式，也可表示成指数形式。表示为 REAL(实型)。

REAL 取值范围：负数取值范围为-3.402823466E+38 到-1.175494351E-38，正数取值范围为 1.175494351E-38 到 3.402823466E+38。

例: R1:REAL:=23.34、R2:REAL:=1.64e+009。



当有实型数据参与运算时, 有时会出现 1#INF 和 1#QNAN0 的值, 含义如下:

- 1#INF 是表示无穷大数, 分为正无穷和负无穷, 它由非 0 的有限数除 0 得到, 或者超过了实型的数据上、下限。实型除零不报错。
- 1#QNAN0 即 not a number, 表示这不是一个数。它是由一些无意义的运算引起的。

4. 时间型数据

时间型数据用于定义时间(TIME)、时刻(TOD)、日期(DATE)、日期时刻(DT)。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	TimeVar1		时间	TIME	T#0MS	FALSE
0002	TimeVar2		日期	DATE	D#1970-01-01	FALSE
0003	TimeVar3		日期时刻	DT	DT#1970-01-01-00:00:00	FALSE
0004	TimeVar4		时刻	TOD	TOD#00:00:00	FALSE

图 5.5-32 时间数据

■ 时间(TIME)

时间数据的标识符为 TIME, 用于定义操作时间。定义时间的格式有 4 种: TIME#, time#, T#, t#, 格式符#后为时间数据, 依次为时、分、秒和毫秒。如图 5.5-32 中 TimeVar1 所示。

例:

正确的定义:

TIME1 := T#14ms;

TIME1 := T#100S12ms; (*允许最高位超过其数据上限*)

TIME1 := t#13d12h34m15s;

错误的定义:

TIME1 := t#5m68s; (*低位超限*)

TIME1 := 15ms; (*缺少提示符 T#*)

TIME1 := t#4ms13d; (*输入顺序有误*)



- 定义时间的最高位允许超过其上限, 而低位不允许超过其上限。

■ 日期(DATE)

日期的标识符为 DATE，用于存储日期。定义日期的格式有 4 种：D#，d#，DATE#，date#，格式符#后为日期数据，依次为年、月、日。如图 5.5-32 中 TimeVar2 所示。

■ 日期时刻(DT)

日期时刻的标识符为 DT，用于存储日期和时刻。定义日期时刻的格式有 4 种：DT#，dt#，DATE_AND_TIME#，date_and_time#，格式符#后为日期和时刻数据，依次为年、月、日、时、分、秒。如图 5.5-32 中 TimeVar3 所示。

■ 时刻(TOD)

时刻数据的标识符为 TOD，用于存储时刻。定义时刻的格式有 4 种：TOD#，tod#，TIME_OF_DAY#，time_of_day#，格式符#后为时刻数据，依次为时、分、秒，其中秒为实数，即秒可有小数。如图 5.5-32 中 TimeVar4 所示。



- 定义时刻的秒为实数，即秒可有小数。

5. 数组(ARRAY)

有时需要将同种类型的数据元素有序的组合成一个有机的整体，以便于引用，数组(ARRAY)就具有此功能。

根据标准数据类型来定义一维、二维数组，数组是将同一类型的数据有序的组合在一起。数组内每个元素有固定的编号(或下标)，例如 2×2 二维数组内 4 个元素的编号(或下标)分别为 11、12、21、22。

可以初始化数组中所有元素或者不初始化。

例如：为一维数组 ARR1 赋初始值，ARR1 数组下标从 1~6，变量类型为 REAL，如图 5.5-33 所示。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	ARR1		数组1	ARRAY[1..6] OF REAL		FALSE

Main.ARR1						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	ARR1[1]			REAL	0	FALSE
0002	ARR1[2]			REAL	0	FALSE
0003	ARR1[3]			REAL	0	FALSE
0004	ARR1[4]			REAL	0	FALSE
0005	ARR1[5]			REAL	0	FALSE
0006	ARR1[6]			REAL	0	FALSE

图 5.5-33 为数组元素赋初始值

在 ST 语言编辑器中对上述数组元素赋值，采用的格式为：“ARR1[1]:=10;”，或者定义另外一个数组变量 ARR2，且保证该数组维数（下标）以及数据类型与 ARR1 完全一样，这时，可以采用的格式为：“ARR1:=ARR2;”。

(1) 定义数组

在变量声明对话框中，我们可以定义一维、二维数组型变量。添加变量后，在该变量行中选择**变量类型**单元，单击下拉箭头▼，从数据类型的下拉列表中选择 **ARRAY** 项，弹出“数组”定义对话框，如图 5.5-34 所示。



图 5.5-34 定义数组

Dim 列表示定义的数组维数，本软件支持多维数组；**Start** 列表示数组起始下标；**End** 列表示终止下标。在**类型**下拉菜单中可选择数组元素的类型，单击**确定**按钮完成定义。



- 数组下标缺省从 0 开始，也可以根据需要修改该起始下标的开始位数。

选中变量表中的数组变量点所在行，双击**序号**或选择右键菜单中的【详细】命令，弹出该变量各个子项的定义窗口，如图 5.5-35 所示，包括：**子项点名**、**变量说明**、**变量类型**、**初始值**、**掉电保护**等属性，在该窗口中修改各个子项的详细信息。



图 5.5-35 数组元素的各个子项

(2) 引用数组元素

程序中可引用数组元素，例如引用二维数组元素 ArrayVar[1,2]，以 ST 语言为例：

```
Var1: =ArrayVar[1,2] (*二维数组元素[1,2]*)
```

6. 指针

指针是一个用来指示一个计算机内存地址的变量。在定义变量时，选择变量类型为 **POINTER**，弹出“设置指针指向的类型”对话框，如图 5.5-36 所示，在 **POINTER TO** 下拉列表中选择指针的指向类型。



图 5.5-36 指针指向的类型定义对话框

这里，指针支持如下两种操作：

$P:=ADR(VAR1);$ (*P 为指针变量，VAR1 为普通变量，这里为将 VAR1 的地址赋给变量 P*)

$VAR1:=VAL(P);$ (*P 为指针变量，VAR1 为普通变量，这里为将变量 P 的值赋给 VAR1*)

说明：

LE PLC 不支持对奇地址取值，如 $p1: \text{pointer to word}$, $p1:=12548637$ ，对 p1 取值即 VAL 运算会引起控制器复位。

变量或功能块地址在程序运行时被存储在指针中。

指针可指向任何数据类型或功能块，甚至可用于自定义类型。地址操作符 ADR 的功能是分配变量或功能块的地址到指针。

取值操作符 VAL 的功能是对指针指向内存进行取值。



- 指针以字节为单位计算！可以通过使用说明 $p=p+SIZEOF(p)$ (p 指向的变量)像在 C_Compiler 一样进行计算。

7. 长实型 (LREAL) 数据



- LREAL 数据类型仅在 MC、LK 工程中支持。

LREAL 负数取值范围为 $-1.7976931348623158e+308$ 到 $-2.2250738585072014e-308$ ，正数取值范围为 $2.2250738585072014e-308$ 到 $1.7976931348623158e+308$ 。

8. 字符串 (STRING) 型数据



- 字符串类型数据仅在 MC、LK 工程中支持。

(1) 字符串变量

字符串类型变量可包含任何一串字符。声明时变量的长度就决定为变量保留多大的存储空间。

用户选择变量的数据类型为 STRING 时，要求用户设置字符串数据的宽度要求用户设置字符串数据的长度，如图 5.5-37 所示。



图 5.5-37 字符串长度设置

字符串变量的长度范围是 1~1000，表示可以输入 1~1000 个字符，每个字符占一个字节（8Bit）的地址空间。若没有指定，系统默认长度为 80 个字符。

用户在设置字符串类型变量的初始值时，需要在内容的始末添加英文单引号。如'this is a string'。

（2）字符串常量

字符串常量的宽度范围是 1~1000 个字符。书写格式同字符串变量的初始值。



- 用户批量导入变量时，请使字符串变量的宽度不超过 1000 个字符，否则会导入失败。
- 字符串型数据只能由 ASCII 码字符组成。

5.5.2.3 自定义数据类型

AutoThink 软件支持的自定义数据类型分为结构体(STRUCT)和枚举(ENUM)。

1. 结构体(STRUCT)

有时需要将不同类型的数据组合成一个有机的整体，以便于引用，结构体(STRUCT)就具有此功能。结构体由多个成员(或分量)组成，每个成员有成员名及类型，类型可不同或相同。在【数据类型】节点下添加结构体，主要步骤如下：

第1步 添加类型

右击【数据类型】节点，显示的树形结构如图 5.5-38 所示。

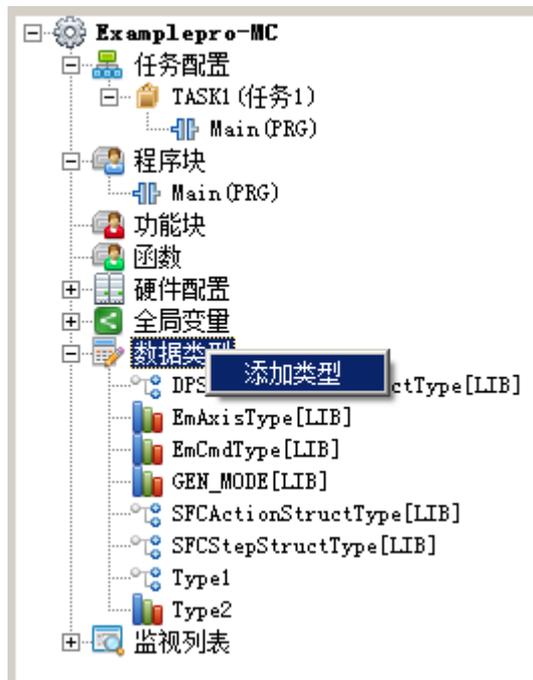


图 5.5-38 数据类型节点树状图



图 5.5-39 数据类型对话框

在**数据类型名称(N)**框内填写结构体名，在【类型】项选择数据类型，单击**确定**，结构体添加完成。

第2步 增加成员

在结构体窗口中右击，选择【增加成员】命令，在定义的数据表中添加一行，缺省的结构体成员名为“s*”，修改结构体的成员名、成员说明、成员类型和初始值，如图 5.5-40 所示。

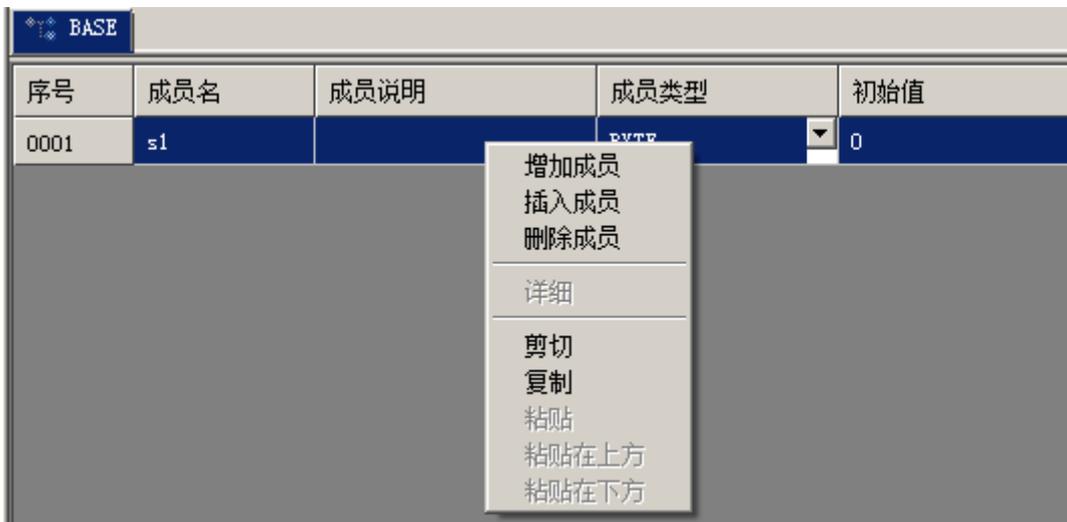


图 5.5-40 定义结构体的方法-3

可以通过右键菜单对已添加的成员，进行其他操作。对结构体进行实例声明后，可通过“实例名.成员名”调用结构体中的成员。

2. 枚举(ENUM)

添加枚举变量的方法与添加结构变量的步骤一致，只是【类型】项选择**枚举**类型。

一个工程中可定义多个枚举，枚举的成员名必须唯一，即不同枚举的成员名中不允许有重名。



- 程序中可直接引用枚举的成员，不必在变量区声明。
- 程序中引用结构体(STRUCT)的成员，必须在变量区声明结构体。
- 结构体成员不支持 STRING 类型。
- 枚举值范围是-32768~32767，枚举成员个数上限 65536。

5.5.3 变量

数据是 AutoThink 中控制和运算的操作对象，人们把变化的数据称为变量，变量存于 CPU 模块的内存储器。为了在数据运算过程中使用、标识这些变量，每个变量都需要有一个名字，即变量名。控制算法组态中所用到的变量的名称和数据类型必须是事先明确的，这就需要首先对用到的变量进行定义，这个定义过程叫做变量声明。

本节叙述变量的类型、变量声明的格式、变量声明的规定、局部变量声明的方法和全局变量声明的方法。

5.5.3.1 变量的类型

1. 按照变量数据类型划分

分为标准类型和自定义类型。

(1) 标准类型

包括布尔型(BOOL)、字节型(BYTE)、字型(WORD)、双字型(DWORD)、长字型(LWORD)、短整型(SINT)、无符号短整型(USINT)、整型(INT)、长整型(LINT)、无符号整型(UINT)、无符号长整型(ULINT)、32位整型(DINT)和无符号32位整型(UDINT);另外还有实型(REAL)、长实型(LREAL)、字符串(STRING)、数组(ARRAY)和时间型,其中时间型包括时间(TIME)、时刻(TOD)、日期(DATE)、日期时刻(DT)。

(2) 自定义类型

分为结构体(STRUCT)和枚举(ENUM)。

2. 从结构形式划分

分为简单型变量和功能块实例两种。

(1) 简单型变量

简单型变量指单一的变化量,能被赋予一个明确的数值。一个简单型变量仅代表一个意义,例如 BOOL、REAL 类型的变量,如图 5.5-41 所示。



序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	VAR1		变量1	BOOL	FALSE	FALSE
0002	VAR2		变量2	REAL	0	FALSE

图 5.5-41 简单变量

(2) 功能块实例

一个“功能块实例”由一组特定的变量组成,具体是怎样的一组特定变量,则取决于该功能块实例的功能块类型。所以功能块实例接近于数据结构的概念。

功能块实例的格式如下,如图 5.5-42 所示,双击功能块实例定义的信息行,展开功能块各个内部项窗口。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	VAR1		变量1	BOOL	FALSE	FALSE
0002	VAR2		变量2	REAL	0	FALSE
0003	BLOCK1		功能块1	RS		FALSE

BLOCK1						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Set		置位	BOOL	FALSE	FALSE
0002	Reset		复位	BOOL	FALSE	FALSE
0003	Q		输出	BOOL	FALSE	FALSE

图 5.5-42 功能块实例



- 添加系统提供的功能块类型的实例名称时，请务必确认该类型功能块所在的库已经添加到当前工程中，否则在“变量类型”下拉列表中不能找到相应类型的功能块名称。特别地，由于在函数的输入变量区和局部变量区不支持功能块类型，所以其相应的下拉列表中不显示功能块类型名。

3. 按照变量有效范围（使用范围）划分

全局变量 (VAR_GLOBAL): 是使用范围最广的变量，一个全局变量可以在工程的多个地方同时使用，但一个工程中的全局变量不允许重名。

局部变量: 相对全局变量而言，使用范围比较有限，只在所属的 POU 中使用有效；在一个工程中可以重名。

局部变量有效范围按照用途可分为输入变量、输出变量、输入输出变量和局部变量。

- **输入变量 (Input_Variable):** 变量是从外部输入的，本功能块（或函数）对该变量只可读。
- **输出变量 (Output_Variable):** 变量是向外部输出的，本功能块对该变量可读写；其它功能块只可读。
- **输入输出变量 (Input/Output_Variable):** 兼有输入变量和输出变量的特性，其它功能块和本功能块对该变量都可读写。
- **局部变量 (Local_Variable):** 是对具体的功能块（或函数）而言，是块实现逻辑运算时所需要的临时变量；该变量不与外部连接，只在它所属的块中有效。在功能块查看信息中被称为参数 (VAR_TEMP)。



- 如果需要在全局范围内引用局部变量，必须是 PRG 类型的 POU 以“POU 名称.局部变量名”的形式使用。

5.5.3.2 变量的声明规范

变量的声明必须符合 AutoThink 软件的语法规则，否则编译出错，对所定义的变量名、变量类型和声明有具体的规定。

1. 变量名

定义变量的名称只能由字母、数字、下划线组成，名称长度最多不超过 32 字节。

变量名的命名遵循如下规则：

- 变量名必须以字母或者下划线开头，不能以数字开头。
- 变量名识别下划线，例如 AB_CD 和 ABC_D 被认为是两个不同的变量名。
- 变量名不区分大小写。例如 VAR1、Var1 和 var1 表示相同的变量。
- 变量名不能为空，且不能包含空格。例如 AB CD 是错误的变量名。
- 变量名中不能包含特殊字符，如中划线“-”和加号“+”等。例如 AB-CD 和 AD+CD 是错误的变量名。
- 变量名不能与类型名（包括自定义类型）、POU 名、枚举名、任务名或类型转换函数名重名，不能以 AT_ 开头。全局变量名、通道名称、配方不能重名。
- 变量名中不能与关键字相同。

2. 变量地址

为一个变量指定一个变量地址，是指把一个变量连接到确定的内存储器地址，即是对输入区（I）、输出区（Q）和中间区（M）地址的映射。

若直接在用户组态区输入直接地址，输入完毕后显示为大写。

示例：在输出区(Q)定义的变量-OUT_SV2，变量地址设置为：%QX2.1>(*Q 输出区，X 位寻址，位变量*)，变量访问的输入格式符含义如表 5.5-14 所示。

表 5.5-14 输入格式符含义

含义	标识
地址标识符	%
存储区位置	输入区-I、输出区-Q、中间区-M

含义		标识
寻址方式		位寻址-X
		字寻址-W
地址	字号	存储区的字号
	位号	存储区的位号。根据数据寻址方式的不同，位号可省略，如果有位号时，位号前必须加分隔符“.”，位号 0~7

使用 M 区地址时建议关联变量名，否则编译时不做处理，同时该地址数据不进行数据同步。如，变量定义区 M 区地址已使用到%MB100，用户组态时直接只用地址%MB101 而没有为其关联变量名，则在编译时，M 区的地址使用显示为%MB100，同时，%MB101 的数据不进行冗余同步。

5.5.3.3 全局变量声明的方法

全局变量在整个工程中有效，可被任意一个程序引用。本小节叙述全局变量声明的方法和操作步骤。



- 工程管理树：右击【全局变量】节点，单击【添加组】。



图 5.5-43 添加组

弹出“新建全局变量组”对话框，如图 5.5-44 所示，写入变量组的名称。



图 5.5-44 “添加组”对话框

变量组名可以是字母、汉字、数字、下划线“_”，但必须以字母、汉字或下划线开头，且遵循如下原则：

- 不能与 POU 名、任务名、关键字、指令库名或功能块名重名。

- 变量组名不可为空，为空时，“确定”按钮显示为灰色，不可用。
- 变量组名称长度最多不超过 32 字节，超出范围的部分无效。
- 变量的命名规则与变量组的命名规则一致，不再赘述。

1. 手动声明变量

输入符合命名规则的变量组名，如 `var_1`，单击**确定**按钮完成添加，在【全局变量】节点下生成新添加的变量组分支，鼠标双击该分支，在工作区域打开该变量组的编辑窗口，如图 5.5-45 所示，在该窗口中单击鼠标右键菜单各项命令添加或删除变量。

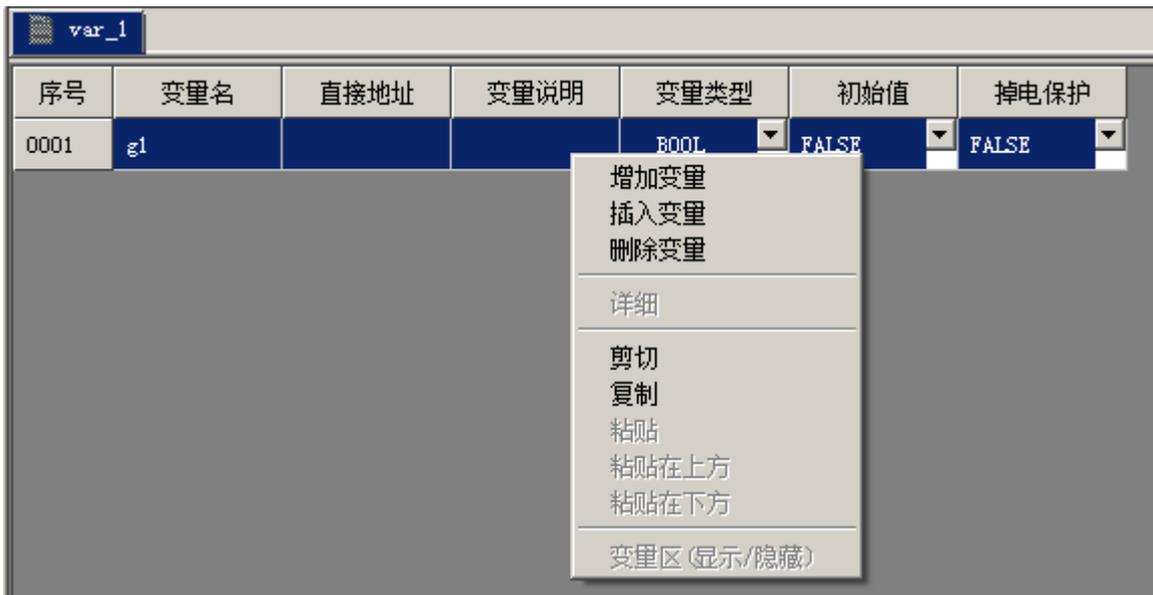


图 5.5-45 添加全局变量

通过单击第一行的变量属性项对变量按属性进行排序。如果序号不是升序，则在右键菜单、【编辑】菜单、工具栏、以及快捷键对于变量的【插入】、【删除】命令不可用。

2. 自动声明变量

通过在编程窗口引用变量，由系统自动提示对未定义过的变量进行声明，从而添加新变量。注意，这里要确认有关**变量自动声明**的设置被启用。启用方法选择菜单【工程】—【选项】，在设置窗口中勾选**变量自动声明**，如图 5.5-46 所示。

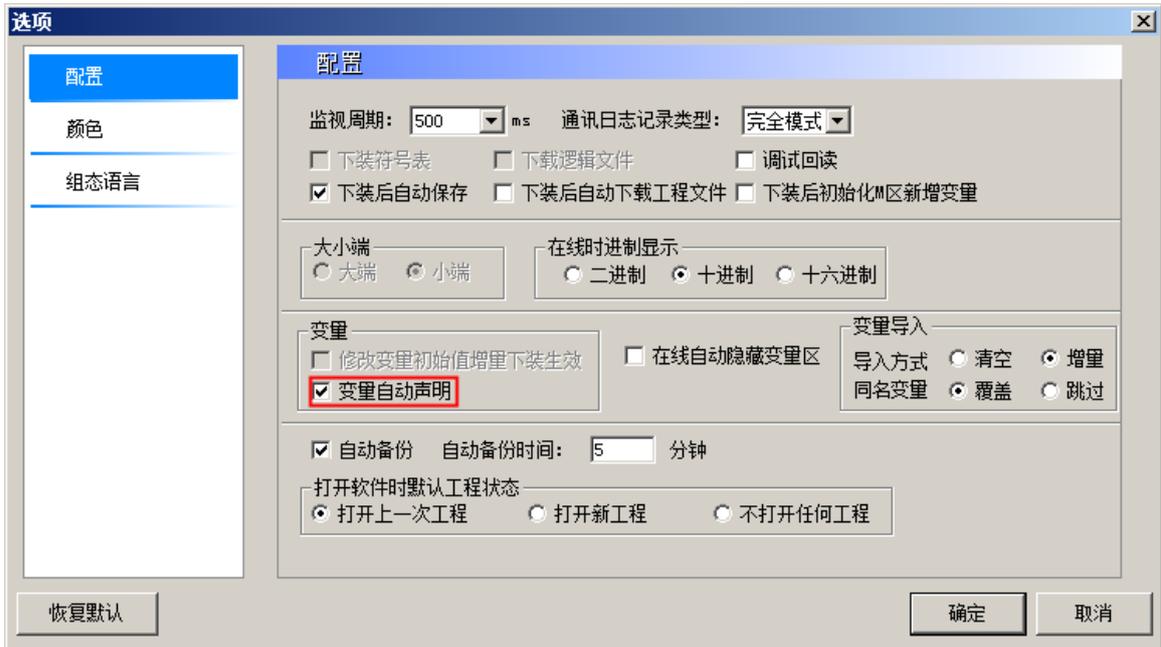


图 5.5-46 选项设置窗口

在编程界面的程序区，若写入未定义的变量，按 **Enter** 键后，则自动跳出变量声明窗口，可对变量进行定义，如图 5.5-47 所示。



图 5.5-47 自动声明变量

在类别下拉菜单可以选择定义该变量为局部变量还是全局变量，如果选择定义为全局变量，则在变量组列表下拉菜单中选择该变量保存到的变量组名称，如图 5.5-48 所示。输入直接地址、变量说明，设置掉电保护属性后，单击确定后该变量被添加至该变量组之中，如图 5.5-49 所示。



图 5.5-48 声明为全局变量



序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	g1			BOOL	FALSE	FALSE
0002	S1			BOOL	FALSE	FALSE

图 5.5-49 保存至自定义全局变量组中



- 在工程中定义的变量类别注意选择为**全局变量**类别。全局变量只能存放在【全局变量】节点下的相应变量组中。

5.5.3.4 局部变量声明的方法

目前，在 POU 程序的变量定义区域，以及在功能块（FB）和函数（Function）类型的 POU 中定义的变量属于局部变量，该变量只在当前 POU 中有效，其他程序不能直接引用，可以通过“POU 程序名.变量名”的格式进行全局引用。

局部变量只能在各自程序相应的变量区中声明。

1. 手动声明变量

在编辑函数和功能块类型的 POU 时，通过编程界面的变量区进行变量的声明或变量的定义。

当编辑【功能块】类型的 POU 时，每个变量定义区分为：输入变量、输出变量、输入输出变量、中间变量这四个标签页，单击标签页名称进入各个类型的变量定义区。

单击鼠标右键，通过右键菜单中的各项命令添加或删除变量。每添加一个变量，相应地变量表中添加一条记录，序号按照添加的顺序依次增加如图 5.5-50 所示。可以修改变量名、变量地址、变量类型、初始值、掉电保护属性。

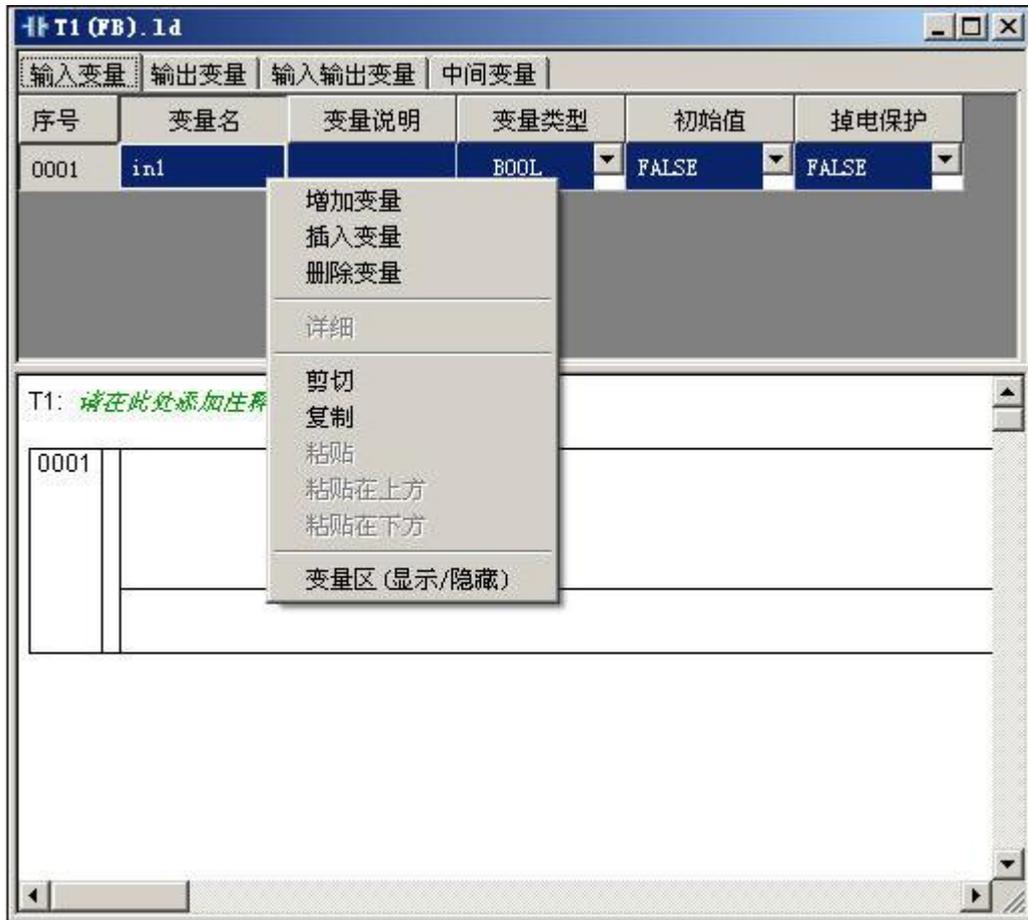


图 5.5-50 手动声明变量-功能块

当编辑【函数】类型的 POU 时，每个变量定义区分为：输入变量、输入输出变量、中间变量这三个标签页，如图 5.5-51 所示，单击标签页名称进入各个类型的变量定义区。变量的操作方法与【功能块】中的一样。

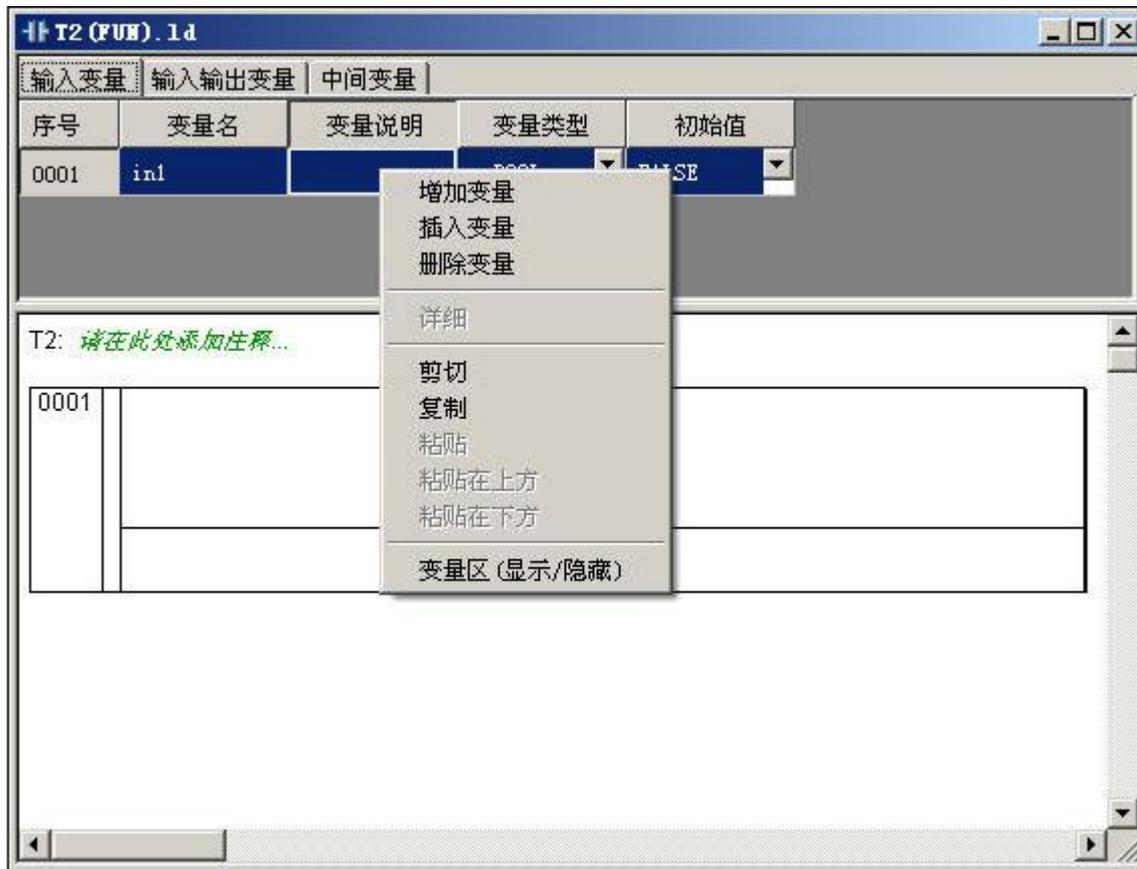


图 5.5-51 手动声明变量-函数

2. 自动声明变量

声明方法参见章节 5.5.3.3 全局变量声明的方法的自动声明变量，只是类别可选项不同，如图 5.5-52 所示。



图 5.5-52 自动声明变量—类别项

5.5.3.5 全局变量

以 MC 工程的【全局变量】节点为例进行介绍。

【全局变量】节点下缺省包含了 AxisParaGroup 轴参数组、GV_Group 变量组、LEIODiagGroup 模块诊断组、ModbusOrderDiagVar 诊断变量组、Table 变量组。当添加 MC1004 或 MC1008 时，全局变量节点下自动生成 LE5405DiagGroup 诊断变量组和 PWM 功能监视变量组。LE5405DiagGroup 诊断变量组的内容请参见章节 5.6.3.7 LE5405 组态的 LE5405DiagGroup。

1. 轴参数组



- 此功能为 MC 工程特有。

双击【全局变量】节点下的【AxisParaGroup】节点，打开轴参数变量表，共有轴参数 64 个，用户可修改变量名和变量说明。双击变量列表行的序号，打开详细参数设置列表框，可以对初始值列的白色可编辑区域的值进行修改。如图 5.5-53 所示。

AxisParaGroup								
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护		
0001	Axis0	%SD32768	Axis0	AXIS		FALSE		
0002	Axis1	Axis0						
0003	Axis2	序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0004	Axis3	0001	AxisID	%SD32768	轴号	USINT	0	FALSE
0005	Axis4	0002	AxisType	%SB32770	轴类型	EMAXISTYPE	Virtual	FALSE
0006	Axis5	0003	AxisStatus	%SD32772	轴当前状态	UDINT	0	FALSE
0007	Axis6	0004	AxisErrMask	%SD32776	轴状态异...	UDINT	1	FALSE
0008	Axis7	0005	Units	%SD32784	单位转换...	LREAL	1	FALSE
0009	Axis8	0006	Speed	%SD32792	运动时设...	LREAL	2.00E+03	FALSE
0010	Axis9	0007	CreepSpeed	%SD32800	爬行速度...	LREAL	2.00E+02	FALSE
0011	Axis10	0008	EndPos	%SD32808	当前运动...	LREAL	0	FALSE

图 5.5-53 轴参数列表

2. GV_Group 变量组

GV_Group 变量组中进行全局变量的定义，供用户组态逻辑时引用。详见章节 5.5.3.3 全局变量声明的方法。

对于全局变量组还可以进行删除或重命名操作，如图 5.5-54 所示。



图 5.5-54 变量组的操作命令

单击【删除组】命令，弹出提示框如图 5.5-55 所示，单击**确定**按钮，变量组被删除。



图 5.5-55 删除变量组提示



- 删除的变量组不能恢复，请在删除前确认是否删除。
- 轴参数组、LEIODiagGroup 模块诊断组、Table 变量组不支持删除和重命名操作。

单击【重命名...】命令，弹出“重命名”对话框，如图 5.5-56 所示，在**新名称**框中输入符合要求的变量组名称，单击**确定**按钮完成重命名操作。

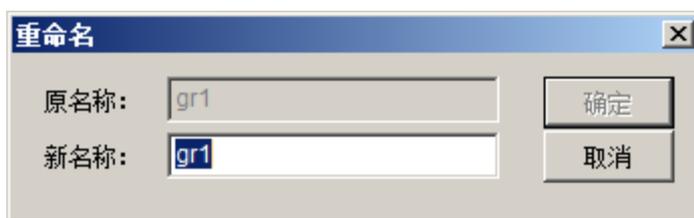


图 5.5-56 重命名变量组

3. IO 模块诊断组

通过 IO 模块诊断组，可以查看已组态 IO 模块的工作状态是否异常。打开该变量组，窗口如图 5.5-57 所示。

LEIODiagGroup						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	ModuleState	%SB10240	第1~7个扩展模块是否...	USINT	0	FALSE
0002	ModuleState1	%SB10241	第8~15个扩展模块是否...	USINT	0	FALSE
0003	ModuleState2	%SB10242	第16~23个扩展模块是...	USINT	0	FALSE
0004	ModuleState3	%SB10243	第24~31个扩展模块是...	USINT	0	FALSE

图 5.5-57 IO 模块诊断窗口

窗口中显示%SB10240~%SB10243 四个地址的诊断信息，每个地址为一个字节，字节的 0~7 位分别代表了 8 个 IO 扩展模块的状态。%SB10240 的 bit0 忽略不计，bit1~ bit7 分别代表硬件配置窗口中 CPU 右侧的第 1~7 个模块的状态。

在线值是 8 个模块 0~7bit 的加权值，8 个模块全部正常时，在线时显示 255，ModuleState 变量的在线值为 254（第 0 位忽略）。模块状态的在线值如图 5.5-58 所示。用户可根据**在线值**去判断哪个模块异常。

LEIODiagGroup						
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	ModuleState	%SB10240	第1~7个扩展模块是否...	USINT	254	FALSE
0002	ModuleState1	%SB10241	第8~15个扩展模块是否...	USINT	0	FALSE
0003	ModuleState2	%SB10242	第16~23个扩展模块是...	USINT	0	FALSE
0004	ModuleState3	%SB10243	第24~31个扩展模块是...	USINT	0	FALSE

图 5.5-58 模块状态在线值

4. ModbusOrderDiagVar

当 LK、MC 工程中组态了 Modbus 主站指令，则编译后，在该变量组中生成变量的诊断信息，如图 5.5-59 所示。该变量的参数不支持编辑。变量名由 SYS_OrderDiag_偏移地址组成，如变量 SYS_OrderDiag_0，“0”为直接地址%SW0 的偏移量 0，用户可以根据**变量说明**列显示的诊断信息描述来查找具体指令。如变量 SYS_OrderDiag_0 是 MC1002E 设备，COM1 端口下的地址为 1 的从站的第一条指令的诊断信息。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	SYS_OrderDiag_0	%SW0	设备MC1002E 端口号COM1 地址1 第1条指令的状态诊断	WORD	0	FALSE
0002	SYS_OrderDiag_2	%SW2	设备MC1002E 端口号COM1 地址1 第2条指令的状态诊断	WORD	0	FALSE
0003	SYS_OrderDiag_4	%SW4	设备MC1002E 端口号COM1 地址1 第3条指令的状态诊断	WORD	0	FALSE
0004	SYS_OrderDiag_6	%SW6	设备MC1002E 端口号COM1 地址1 第4条指令的状态诊断	WORD	0	FALSE
0005	SYS_OrderDiag_8	%SW8	设备MC1002E 端口号COM1 地址1 第5条指令的状态诊断	WORD	0	FALSE
0006	SYS_OrderDiag_10	%SW10	设备MC1002E 端口号COM1 地址1 第6条指令的状态诊断	WORD	0	FALSE
0007	SYS_OrderDiag_12	%SW12	设备MC1002E 端口号COM1 地址1 第7条指令的状态诊断	WORD	0	FALSE
0008	SYS_OrderDiag_14	%SW14	设备MC1002E 端口号COM1 地址1 第8条指令的状态诊断	WORD	0	FALSE
0009	SYS_OrderDiag_16	%SW16	设备MC1002E 端口号COM1 地址1 第9条指令的状态诊断	WORD	0	FALSE
0010	SYS_OrderDiag_18	%SW18	设备MC1002E 端口号COM1 地址1 第10条指令的状态诊断	WORD	0	FALSE
0011	SYS_OrderDiag_20	%SW20	设备MC1002E 端口号COM1 地址1 第11条指令的状态诊断	WORD	0	FALSE

图 5.5-59 Modbus 变量诊断信息

在线时，通过指令状态的码值可以诊断指令状态。

表 5.5-15 Modbus TCP 主站指令状态码值定义

指令状态码值	含义	支持的平台
0	无错误	LK、MC
2	超时	LK、MC
4	功能码错误	LK
64	发送的数据与接收的数据不匹配	LK、MC
128	单元 ID 不匹配	MC
16	单元 ID 不匹配（从站地址错误）	LK
32	TCP 连接失败	LK
33	发送请求报文失败	LK
34	接收确认报文失败	LK
128+1	从站不支持的功能码	LK
128+2	数据地址溢出	LK
128+3	数据范围溢出	LK
128+4	从站设备故障	LK
128+6	从站设备忙	LK
128+15	从站其它故障	LK

表 5.5-16 Modbus RTU 主站指令状态码值定义

指令状态码值	含义
0	无错误
1	数据 CRC 错误
2	超时
4	功能码错误
8	数据长度错误
16	从站地址错误

5. PWM 功能监视变量组

当添加控制器 MC1004、MC1008 后，全局变量节点下生成【PWMParaGroup】变量组。该变量组配合 PWM 功能块使用，用于 PWM 功能块参数的在线监视和部分参数的修改。PWM 功能块的详细信息请参见《MC1000 系列多轴运动控制器指令手册》中第 4 章的 PWM 激光控制。

该变量组已定义两个变量，双击变量名可查看详细的参数信息，如图 5.5-60 所示。

PWMParaGroup						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	HMC_PWM0	%SD92952	HMC_PWM0	HMC_PWM		FALSE
0002	HMC_PWM1	%SD93096	HMC_PWM1	HMC_PWM		FALSE
HMC_PWM0						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	PWMChan1	%SD92952	PWM通道号	USINT	0	FALSE
0002	ControlMode	%SB92953	控制模式...	USINT	0	FALSE
0003	OutputMode	%SB92954	输出方式...	USINT	0	FALSE
0004	OutputID	%SB92955	DO通道号...	USINT	0	FALSE
0005	DOCompens...	%SD92956	DO输出补...	DINT	0	FALSE
0006	ArrLinkAxis	%SD92964	自动随动...	POINTER T...	0	FALSE
0007	LinkAxisNum	%SD92968	速度关联...	USINT	0	FALSE
0008	AxisSpeedSrc	%SB92969	速度计算...	USINT	0	FALSE
0009	Enable	%SD92976	PWM输出使...	USINT	0	FALSE
0010	ManualFreq	%SD92984	手动输出...	LREAL	0	FALSE
0011	ManualDut...	%SD92992	手动输出...	LREAL	0	FALSE
0012	FixedFreq	%SD93000	设定频率...	LREAL	0	FALSE
0013	FixedPuls...	%SD93008	设定脉宽...	UDINT	0	FALSE
0014	FixedDuty...	%SD93016	设定占空...	LREAL	0	FALSE

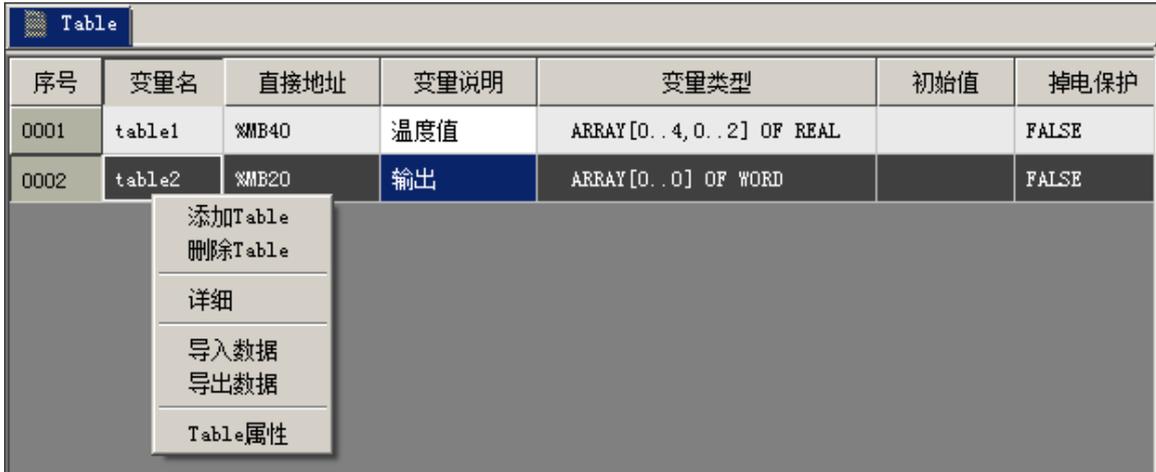
图 5.5-60 PWMParaGroup 变量组

6. Table 数据区管理



- 此功能为 MC 工程特有。

在【Table】变量组视图中，支持用户新建、删除 Table 表、导入导出 Table 表数据等，如图 5.5-61 所示。下面对 Table 表支持的功能分别进行介绍。



序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	table1	%MB40	温度值	ARRAY[0..4, 0..2] OF REAL		FALSE
0002	table2	%MB20	输出	ARRAY[0..0] OF WORD		FALSE

图 5.5-61 Table 变量组视图

(1) 添加 Table 表

在 Table 变量组视图中，打开右键菜单，选择【添加 Table】。打开对话框后，填写 Table 属性，如图 5.5-62 所示。



图 5.5-62 添加 Table 表对话框

组名与全局变量命名要求一致，变量类型从列表中选择，起始地址只能为 M 区。维数可以是一维或者二维，维尺度定义 Table 表的行数和列数。定义完成后，Table 变量组中可以查看到新添加的 Table 表变量，如图 5.5-63 所示。

Table						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	table1	%MB40	温度值	ARRAY[0..4,0..2] OF REAL		FALSE
0002	table2	%MB20	输出	ARRAY[0..0] OF WORD		FALSE
0003	table3	%MB80	压力值	ARRAY[0..3,0..7] OF LREAL		FALSE

图 5.5-63 Table 表视图

(2) 删除 Table 表

在【Table】变量组中，选中一个或多个 Table 表变量后，按 DEL 键或者使用右键菜单中的【删除 Table】命令，即可删除选中的 Table 表。

(3) 查看 Table 表元素

在【Table】变量组中，选中一个 Table 表变量后，双击该变量序号或者使用右键菜单中的【详细】命令，即查看选中 Table 表的数据元素。Table 表元素以表格形式显示，其中第一行为行号，第一列为列号。以 table3 为列，打开的对话框如图 5.5-64 所示。数据元素的初值（离线时）和在线值（在线时）可在此处修改。

talbe3								
序号	0	1	2	3	4	5	6	7
0000	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0

图 5.5-64 Table 元素视图

(4) 导入/导出 Table 数据

在【Table】变量组中，选中一个 Table 表变量后，使用右键菜单中的【导入数据】或者【导出数据】命令，即可完成相应操作。

选择导出数据，导出文件打开后格式如图 5.5-65 所示。Table 表变量按行为单位导出，每行依次为本行各元素的值。



	A	B	C	D	E	F	G	H	I
1	0	1	2	3	4	5	6	7	
2	0	0	22	33	0	0	0	0	
3	0	111	0	0	0	0	0	0	
4	0	0	0	0	4444	0	0	0	
5									
6									
7									
8									

图 5.5-65 Table 数据文件内容

(5) 查看 Table 表属性

在【Table】变量组中，选中一个 Table 表变量后，使用右键菜单中的【Table 属性】命令，即可完成相应操作。在图 5.5-66 对话框中，只可查看 Table 表属性，不可修改。



图 5.5-66 Table 表属性

5.5.3.6 监视列表

将重要变量添加到各个监视列表中，以便在监视模式下跟踪和观察各个变量的值变化情况。



- 工程管理树：右击【监视列表】，单击【添加列表】。



图 5.5-67 添加监视列表命令



图 5.5-68 “添加监视列表”对话框

在名称框中输入监视列表的名称，单击确定在【监视列表】节点下添加列表分支，同时在工作区域自动加载该列表的编辑窗口，如图 5.5-69 所示。



图 5.5-69 “监视列表”编辑窗口

监视列表子节点进行可以删除或者重命名操作。

在监视列表的编辑窗口中，可以添加、修改或者删除监视变量内容。在监视列表编辑窗口中右击，选择【增加变量】命令，如图 5.5-70 所示，在变量名中输入程序中定义的变量名或者变量地址，则添加后，在监视模式下，自动读取显示该变量的变量类型、变量说明、在线值（显示该变量的实时值）。

可以创建多个监视列表以便对重要的变量进行分类管理，便于监视模式下对其数值变化的掌握和跟踪。



图 5.5-70 “监视列表”窗口中的操作

离线状态下在监视列表编辑窗口中单击鼠标右键，如图 5.5-70 所示，可以选择【批量监视】菜单项，批量监视一段连续地址（100 个字以内）的内容。如图 5.5-71 所示，通过确定监视的内存区，输入起始地址和监视变量的个数，完成批量监视。其中，起始地址包含以下信息：要监视变量的区域（IQMS），类型（位\字节\字\双字），起始地址。一次批量监视设置中所有变量的类型必须一致。



图 5.5-71 批量监视对话框

单击**确定**之后，软件会检查输入的起始地址、变量个数是否合法，此段连续地址的长度是否超过 100 个字等。

在线监视状态下，可以通过右键菜单选择监视变量在线值的进制，表示监视列表中所有变量在线值的进制显示情况。在线状态下监视列表编辑窗口的操作命令如图 5.5-72 所示。



图 5.5-72 在线状态下监视列表编辑窗口的操作命令



图 5.5-73 选择二进制表示变量



图 5.5-74 选择十六进制表示变量



- 监视列表中可以添加全局变量，或者添加程序块（必须为 PRG 类型）中的局部变量，添加局部变量时，格式为“程序名.变量名”，例如“TEST1.VAR1”。

5.5.3.7 变量访问

“访问”即“使用”，运算过程中对变量的访问包括：读取变量值和给变量赋值。

- 访问一个简单型变量，直接写它的变量名即可

如：AM01:=AM02+AM03。

这个例子表示：读取变量 AM02 和 AM03 的值，将二者相加，并将加的结果赋值给变量 AM01。

- 访问一个功能块实例的项，书写格式为：功能块变量名.项名

如：PID01.SP（取 PID01 的设定值项，前提已经声明了 PID01 的类型为 HSPID）。

- 访问一个局部变量（其他程序进行引用），书写格式为：程序名.变量名

如：VAR1(POU1 中定义的局部变量)，在 POU2 中的引用格式为：POU1.VAR1。

■ 快速访问变量

在程序编辑区选中某一变量名称，按下 **Shift+F2** 功能键，弹出该变量对应的“变量声明”对话框，快速显示当前选中的变量的属性窗口，如图 5.5-75 所示。可以修改白色背景的编辑框。



图 5.5-75 “变量声明”对话框

在程序编辑区选中某一变量名称，按下 **F2** 功能键，弹出“帮助管理”对话框，如图 5.5-76 所示，用户可以在这个窗口中选择想要使用的变量。

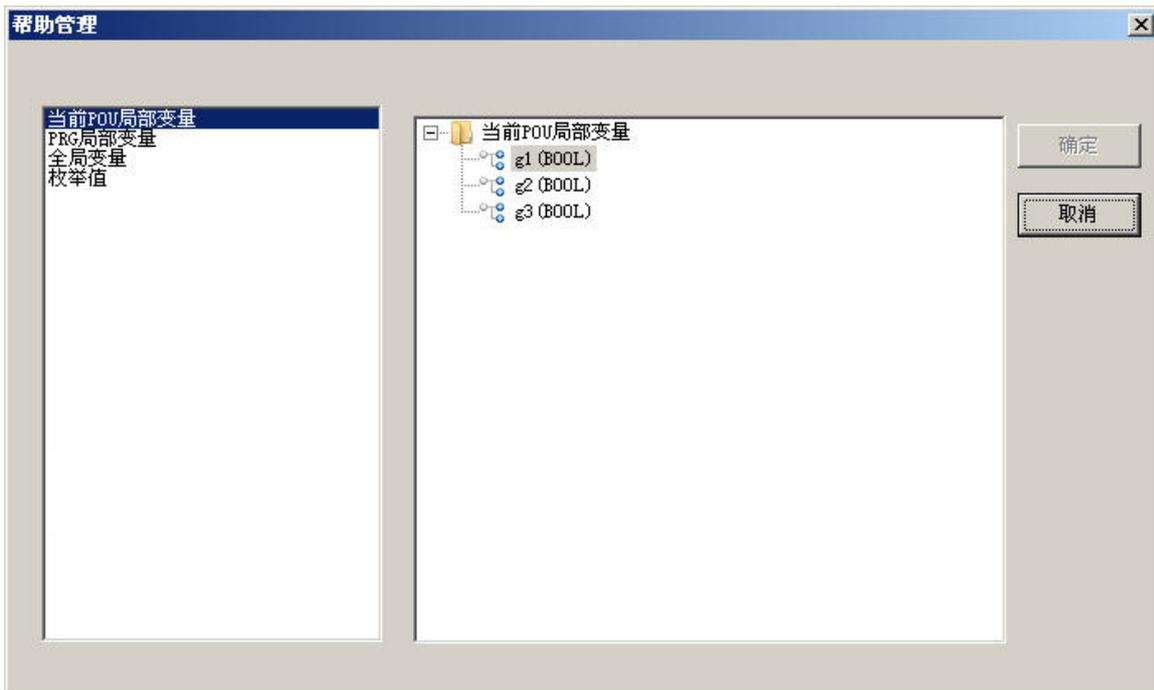


图 5.5-76 “变量编辑管理”对话框



- 在函数、功能块、程序类型的 POU 变量定义区中定义的变量为局部变量，允许定义同名的全局变量和局部变量，但注意局部变量具有优先权，会覆盖全局变量。
- 在程序中使用变量时一定要注意数据类型前后一致！即一个信号连线两端的变量的数据类型应严格相同！

5.5.3.8 变量显示

对于定义在 I 区、Q 区、M 区、S 区的变量，当在 POU 方案页中组态逻辑时，若是写入直接地址，如 %IX0.0，如果有与其直接地址相同的变量，则按照以下优先级显示，全局变量>PRG 局部变量>通道变量。即若定义了一个全局变量 g1:BOOL:%IX0.0，则优先关联 g1，以此类推。



- 同名变量的情况除外。

举例说明：组态时定义了一个全局变量 g1，如图 5.5-77 所示。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	g1	%IX0.0		BOOL	FALSE	FALSE

图 5.5-77 全局变量 g1

定义一个 PRG 局部变量 g2，如图 5.5-78 所示。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	g2	%IX0.0		BOOL	FALSE	FALSE

图 5.5-78 PRG 局部变量 g2

通道变量，如图 5.5-79 所示。

通道号	通道名称	通道类型	通道地址	通道说明
1	M1_CH_1	BOOL	%IX0.0	第0位输入

图 5.5-79 通道变量

组逻辑时，在菜单【工程】—【选项】—【组态语言】—【显示】中勾选输入地址后由变量替代，在线圈或触点处输入 %IX0.0，则显示优先级高的变量 g1，如图 5.5-80 所示。

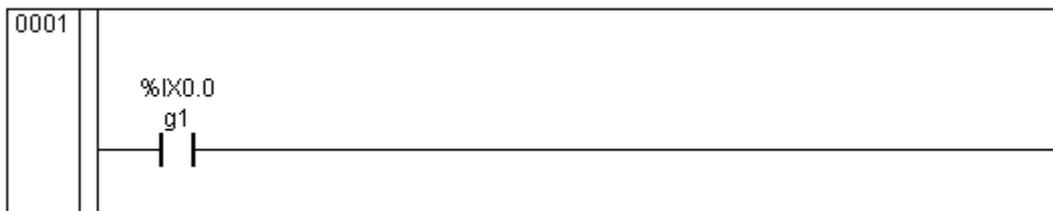


图 5.5-80 变量显示

5.5.3.9 导入变量

将指定的变量表格导入到工程中，数据表格式如图 5.5-81 所示。

变量名	直接地址	变量说明	变量类型	初始值	掉电保护
p1			DWORD	0	FALSE
p2			DWORD	0	FALSE

图 5.5-81 导入的数据表样式

可导入的数据分为两类：全局变量组中的变量和程序（PRG）中的变量。

(1) 全局变量组中的变量

该表单名称无要求，表格第一行格式必须为：“变量组名” + (COMMON)。执行导入操作时，如果工程中没有相应的变量组，则工程自动创建与导入表单同名的变量组，并且加入相应的数据。

(2) 程序（PRG）中的局部变量

该表单名称无要求，表格第一行格式必须为：“程序名” + (PRG)。执行导入操作时，如果工程中没有相应的程序，则相应表单中的变量不能被导入，在信息窗口中会提示“不存在 PRG 局部变量组：****导入失败**”。

导入变量前可选择变量的导入模式，缺省为**增量覆盖**导入。导入模式的选择请参见章节 5.2.10 工程设置的配置。

变量各个项的数值定义与工程中的保持一致。执行该命令，将弹出导入模式确认提示框，如图 5.5-82 所示。



图 5.5-82 导入变量提示框

单击**确定**，选择的表格中的数据导入到相应的程序中或者全局变量组中。在信息输出窗口提示导入信息。如图 5.5-83 所示。



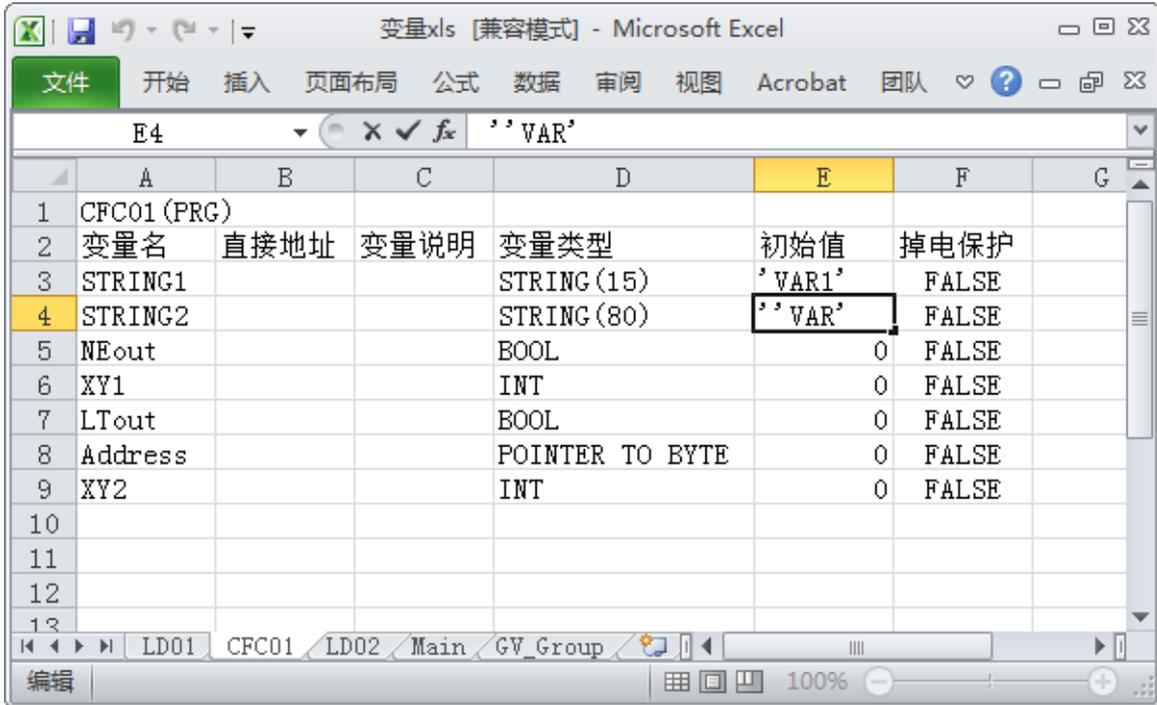
图 5.5-83 导入后信息输出窗口显示



- 用户批量导入变量时，请使字符串变量的宽度不超过 1000 个字符，否则会导入失败。
- 表格中除需要修改的变量列外，其余列禁止修改，包括调整列宽等，否则导入失败，失败后请重新导出表单使用。
- 执行导入操作时，会导入表格中的隐藏工作表，不使用的的工作表请删除。

5.5.3.10 导出变量

将工程文件中全局变量组和 POU（PRG 类型的）中的局部变量导出到指定的表格文件中。导出的表格如图 5.5-84 所示。



	A	B	C	D	E	F	G
1	CFC01 (PRG)						
2	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	
3	STRING1			STRING (15)	' VAR1'	FALSE	
4	STRING2			STRING (80)	' VAR'	FALSE	
5	NEout			BOOL	0	FALSE	
6	XY1			INT	0	FALSE	
7	LTout			BOOL	0	FALSE	
8	Address			POINTER TO BYTE	0	FALSE	
9	XY2			INT	0	FALSE	
10							
11							
12							
13							

图 5.5-84 导出的数据表样式



- 当用户在表中编辑 STRING 类型的初始值时，会出现 3 个单引号，第一个引号为 Excel 对'的转义字符，如图 5.5-84 中的 E4 单元格，用户只能在后两个单引号之间编辑初始值。

5.5.3.11 未使用变量

通过该命令可以对当前工程中未使用变量进行回收。



- 菜单栏：单击【工程】—【查看】—【未使用变量】。

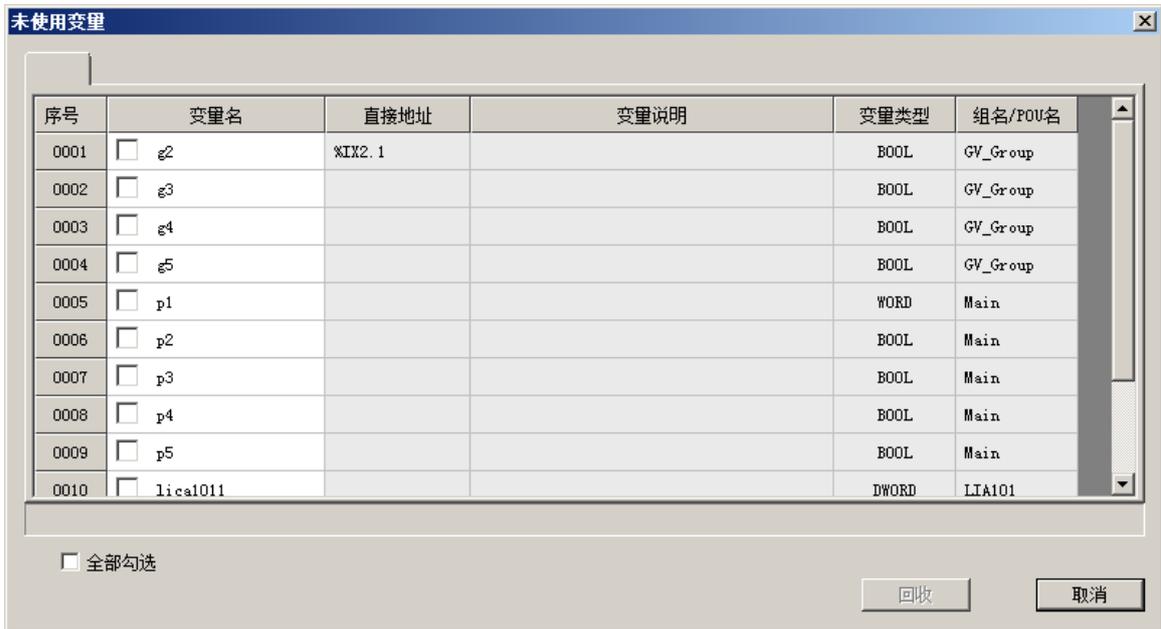


图 5.5-85 “未使用变量”对话框

如图 5.5-86 所示。在该窗口中显示全局变量组和 PRG 类型的 POU 中已定义但未使用的变量。窗口中记录了未使用的变量的变量名、及该变量所在的全局变量组或 POU 名。可以选择变量名进行回收。

手动勾选需要删除的变量，或通过单击**全部勾选**框快速全选，单击**回收**。

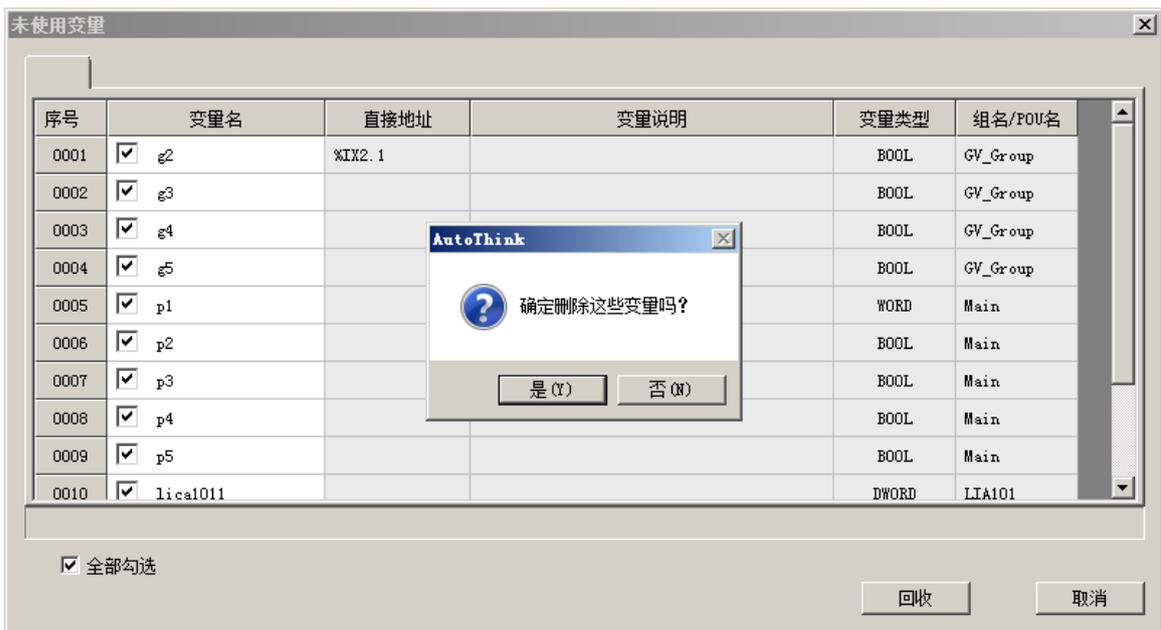


图 5.5-86 变量回收确认



- 工程编译成功后，才能使用该命令。
- 如果 POU 属性为只读或者 POU 有密码但是没有被打开，则该 POU 变量不能进行回收。

5.5.3.12 地址重叠

仅 LE 支持此功能。



- 菜单栏：单击【工程】—【查看】—【地址重叠】。

选择该命令，检查工程中内存分配的重叠情况。检查范围包括在全局变量组、PRG 类型的 POU 中已定义直接地址的变量、配方中分配的存储区域及硬件配置中的通道地址。除配方外，全局变量组和 PRG 类型 POU 中的变量需在组态逻辑中引用才能检测到。检查结果可以查看信息窗口的【语法检查】标签页的内容，如图 5.5-87 所示。检查结果采用分组显示，列出每处内存重叠信息，每组中包括内存重叠位置地址及重叠的各变量所在的内存起始地址。

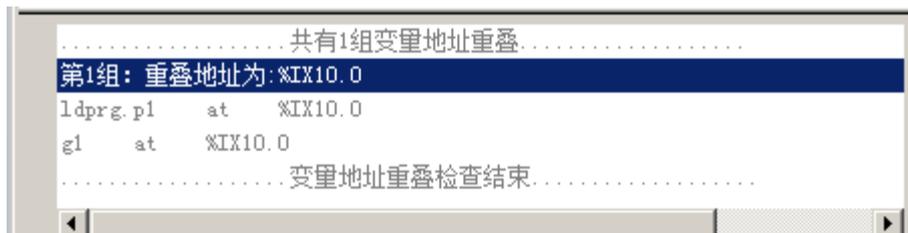


图 5.5-87 重叠内存区



- 工程编译成功后，才能使用该命令。

5.5.3.13 多路写输出情况

仅 LE 支持此功能。



- 菜单栏：单击【工程】—【查看】—【多路写输出】。

选择该命令，显示“多路写输出”窗口，查看多次作为输出使用的变量直接地址，包括全局变量或 PRG 类型的 POU 中的变量，如图 5.5-88 所示。

结果信息中包括多路写输出的变量名、使用该变量的 POU 名及该变量在 POU 中所在节的编号。【多路写输出】只能对 Q 区的直接地址变量进行检测。

多路写输出				
序号	变量名	直接地址	POU名	位置
0001	g2	%QX6.0	test01 (PRG).ld	0001
0002	ldprg.p3	%QX6.0	ldprg (PRG).ld	0001
0003	ldprg.p3	%QX6.0	ldprg (PRG).ld	0002

图 5.5-88 “多路写输出”窗口



- 工程编译成功后，才能使用该命令。

5.5.3.14 内存使用情况

仅 LE 支持此功能。



- 菜单栏：单击【工程】—【查看】—【内存使用】。

用户在编译完工程后，可以选择该命令查看控制器内存使用情况。软件按照 I/Q/M/S 的顺序，以字节为单位，如果某一字节的内存已经被组态，则显示给用户该字节的使用情况，如第几位被使用，或者该字节是作为字节、字还是双字被组态的。如图 5.5-89 窗口所示，结果信息中显示该内存是以什么形式被组态，对应的以对号标记。

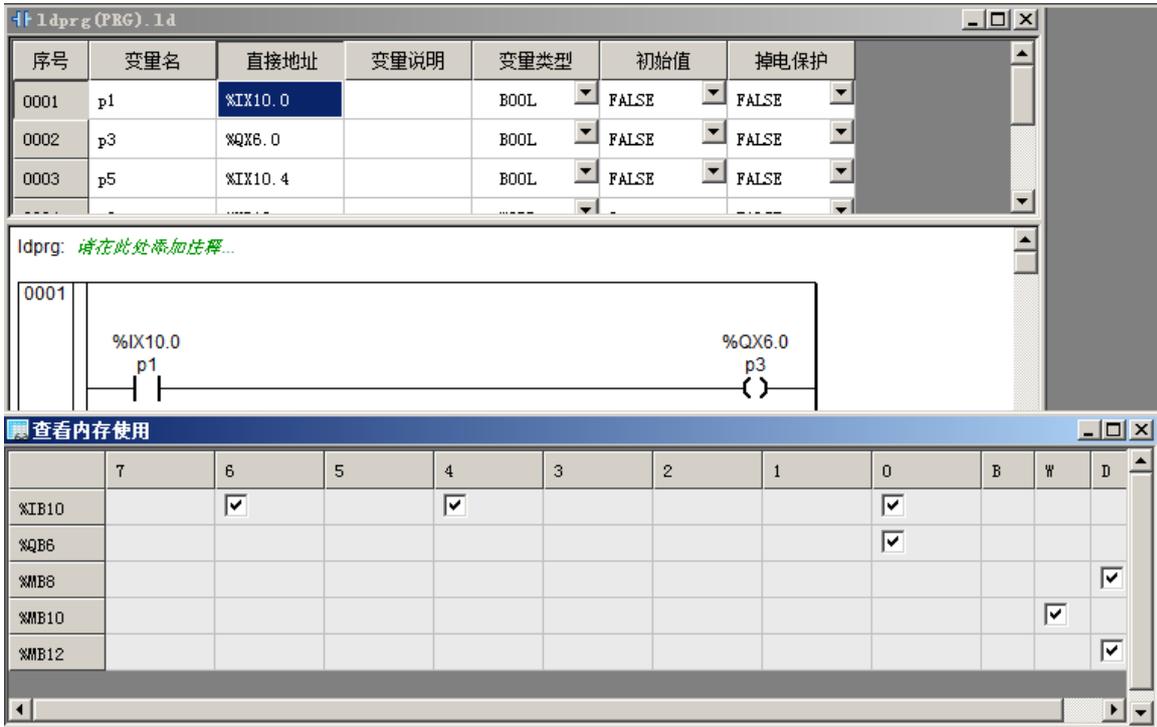


图 5.5-89 查看内存使用的窗口



- 工程编译成功后，才能使用该命令。

5.5.3.15 查看交叉引用表



- 菜单栏：单击【工程】—【查看交叉引用表】。

使用该命令可以查看当前工程中交叉引用的变量。交叉引用表中显示引用变量的 POU 名、变量被引用的位置以及该变量在被引用位置的读写属性。若该变量被赋值，则读写属性为写；否则，读写属性为读。查看交叉引用表的窗口如图 5.5-90 所示。



图 5.5-90 “查看交叉引用表”窗口

交叉引用表默认**精确查找**，在**变量名**中输入要查看的变量名称或变量地址，变量名不区分大小写。**POU 名**默认为**全部**，可以在下拉框中选择已编译的 POU 名。单击**查找**，在交叉引用表中查找出**变量名**完全符合的所有数据，如图 5.5-90 所示。若不勾选**精确查找**，则将查找出所有包含**变量名字段**的数据。双击交叉引用表的某一行，将定位到 POU 中该变量被引用的位置。单击各列表项的**名称**，可以对各列表项进行升降序排序。



- 工程编译成功后，才能使用该命令。
- 【查看交叉引用表】中显示的信息为最近一次工程编译通过所生成的信息。如果新增交叉引用变量，没有进行编译，则不会更新【查看交叉引用表】中显示的内容。

5.5.3.16 引用查看

变量【引用查看】功能是本软件对查看交叉引用功能的细化，主要功能包括三个方面：查找当前选中变量的前一个被引用的地方并跳转定位；查找后一个被引用的地方并跳转定位；查找全部被引用的信息并显示。

在查找跳转过程中，根据如下原则显示被定位信息的位置：

- POU 名：根据工程管理树中 POU 节点的前后顺序(从上至下)排列查找的引用结果。
- POU 中的位置：同一个 POU 内部的变量顺序是按照从上到下，从左到右的，可直接从交叉引用表的数据中可以获取、显示。CFC 语言由于其没有明确上下左右的顺序，故 CFC 中的变量的顺序由元件的 ID 的顺序（即 CFC 元件被添加的先后顺序）决定。



- 程序区：右击某一关联了变量的元素，单击【引用查看】。

在弹出的对话框中显示此变量对应的 POU 名及引用位置。

1. 查询简单全局变量

变量 a 在 Main(PRG)和 st_prg 都被引用，编译通过后，选中此变量关联的触点，单击右键在弹出菜单中选择【引用查看】，如图 5.5-91 所示。

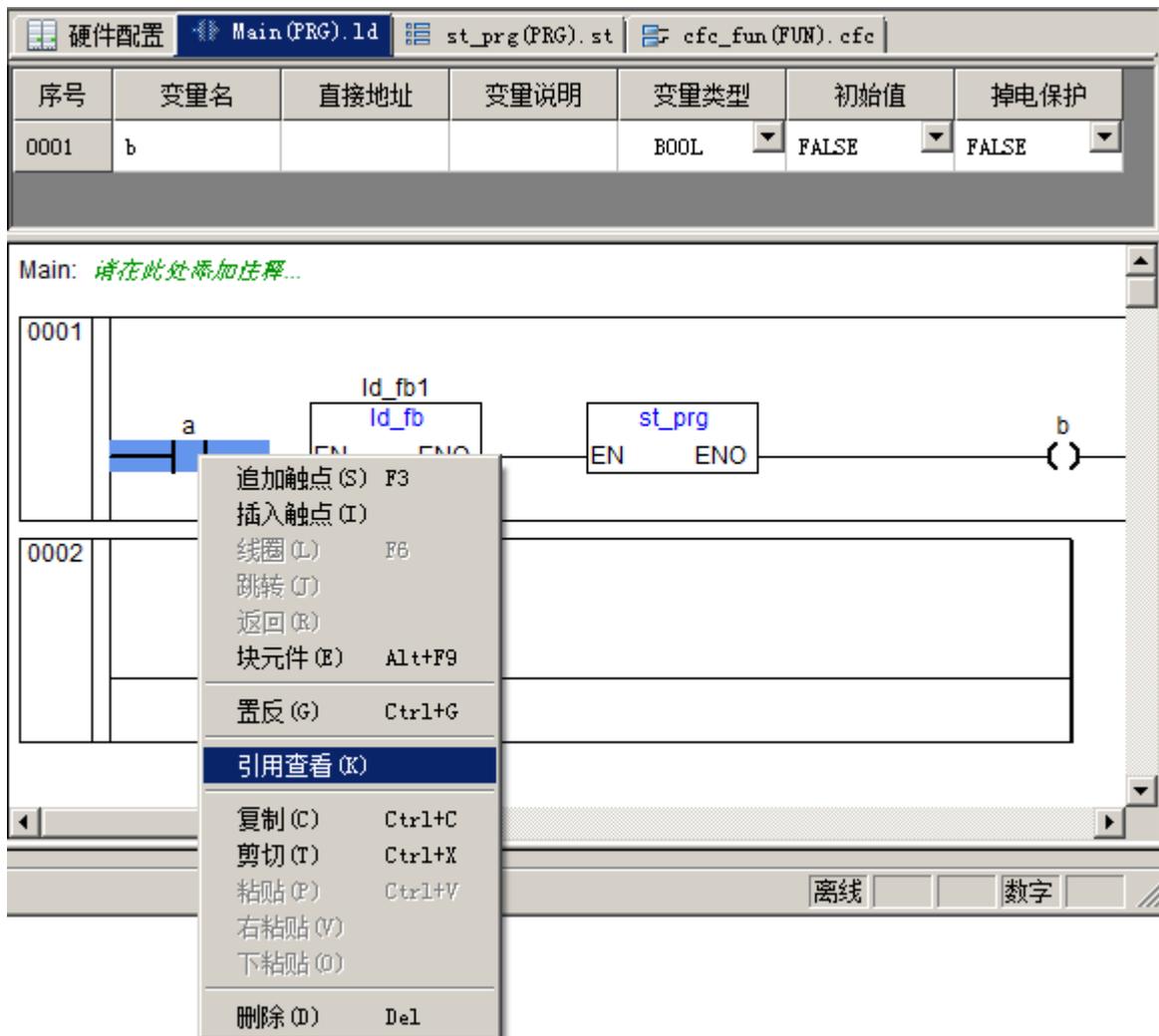


图 5.5-91 选择触点，进行变量引用查看

在弹出的引用查看结果界面中，可以看到该变量 a 被引用的所有信息，包括 POU 名和位置。点击前一个、后一个可以跳转到相应的 POU 界面。如图 5.5-92 和图 5.5-93 所示。

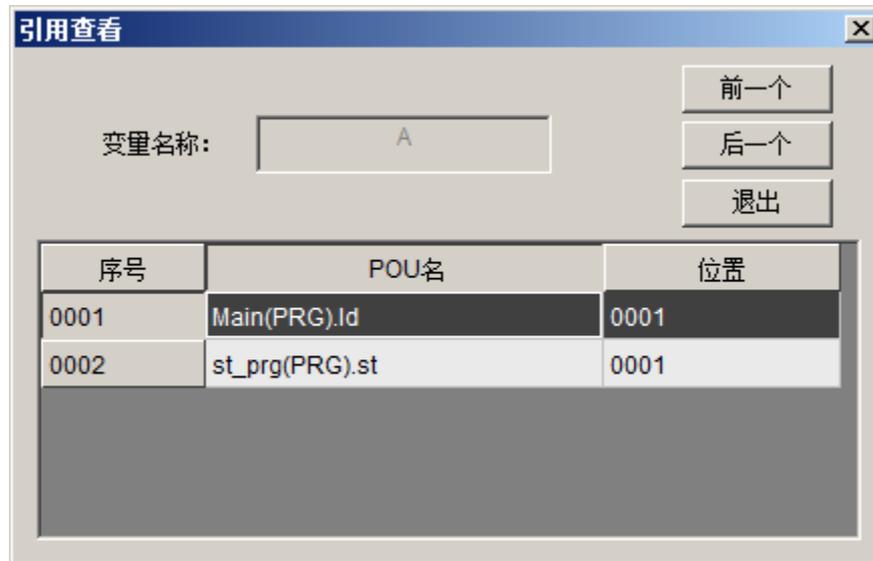


图 5.5-92 显示引用查看结果：所在 POU 名和位置

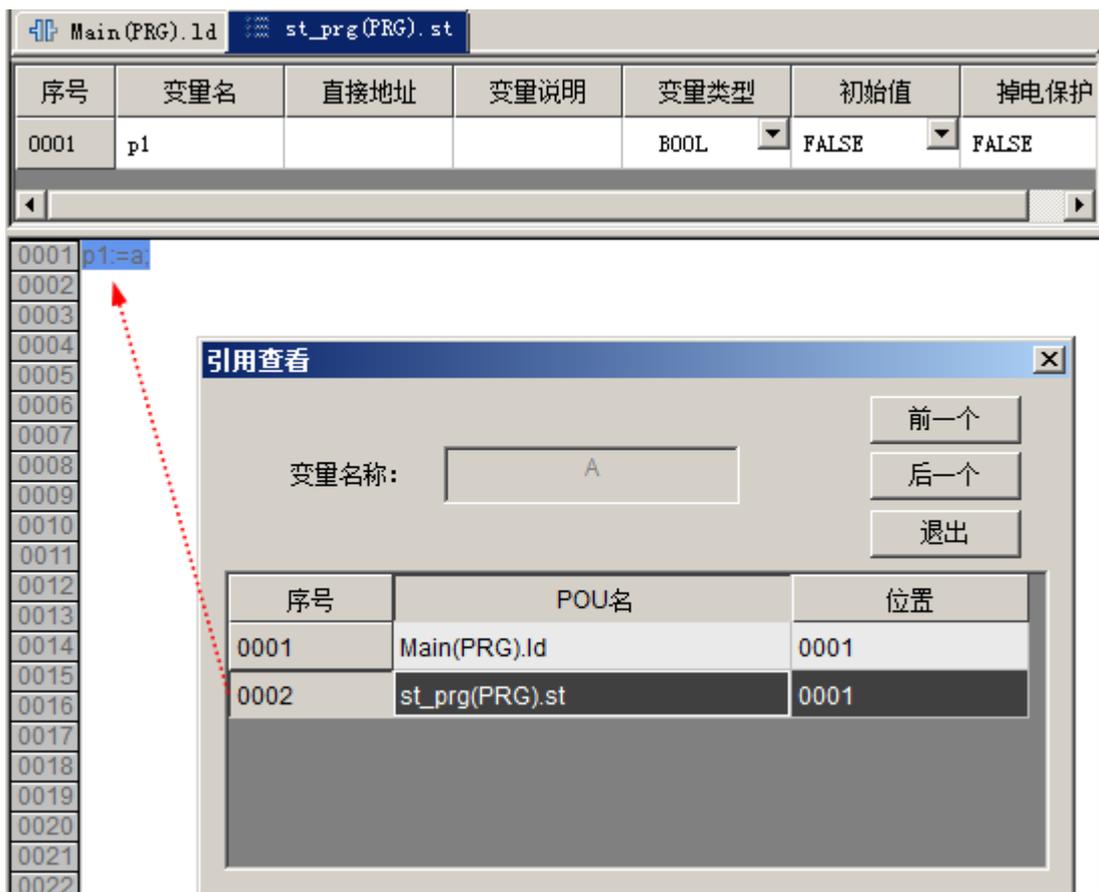


图 5.5-93 根据引用查看结果，进行定位

2. 查询功能块实例变量

同样的操作步骤，也可以对功能块实例变量进行查看，功能块可以是工程（库）中的功能块，也可以是自定义的。如图 5.5-94 和图 5.5-95 所示，查看功能块实例变量 LD_FB1。

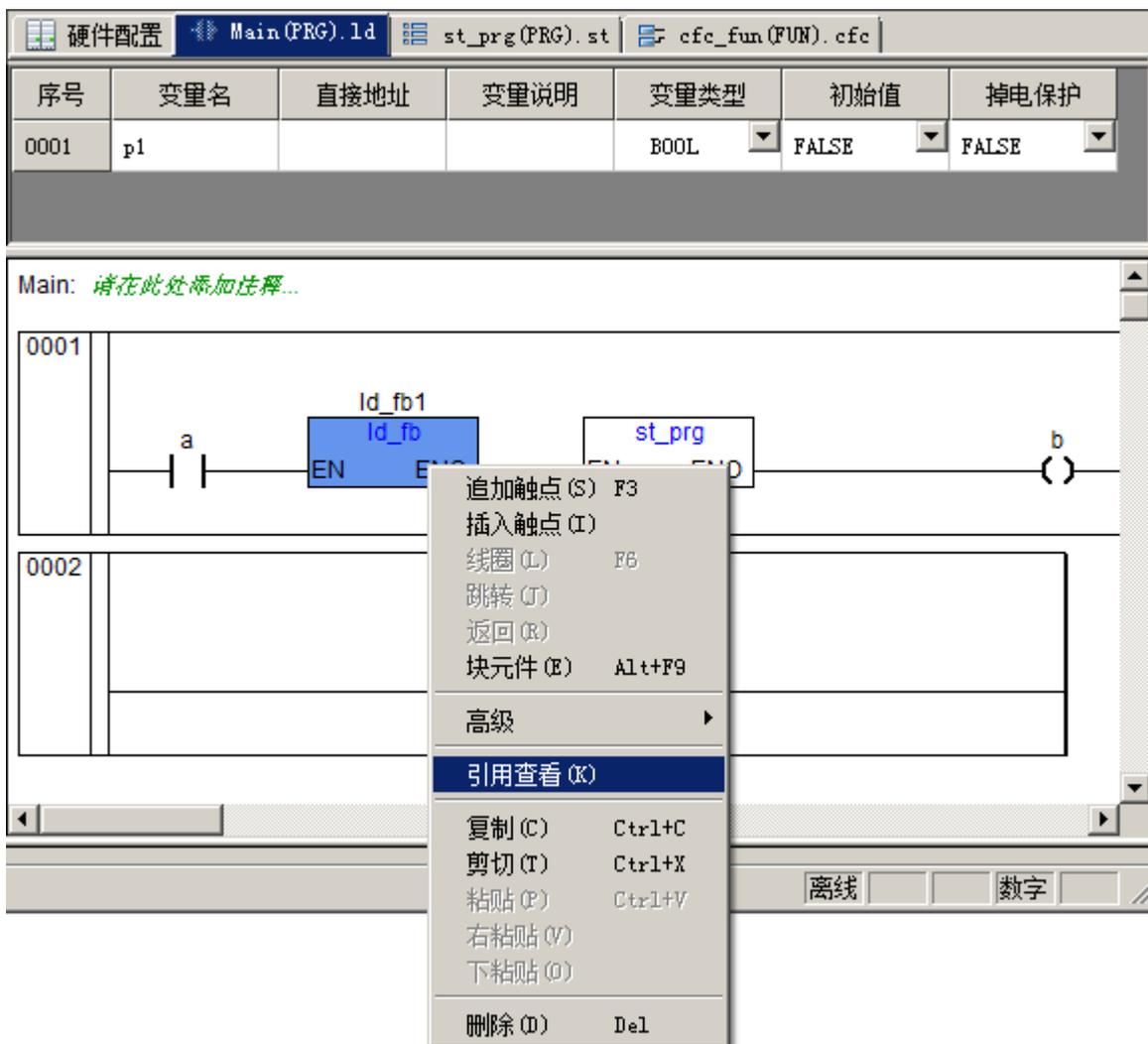


图 5.5-94 查看功能块实例变量的引用

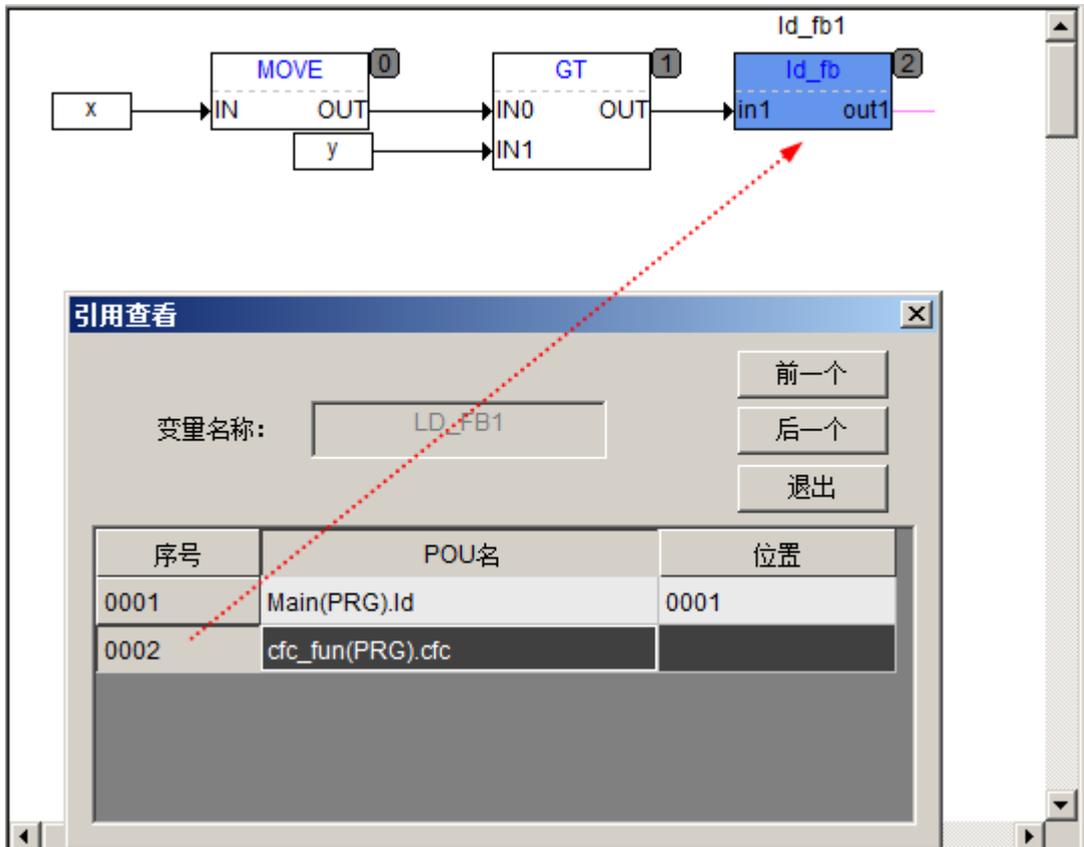


图 5.5-95 选择查看结果的一项，跳转到相应 POU

5.5.3.17 输入助手



- 菜单栏：单击【编辑】—【输入助手】；
- 快捷键：F2。

使用输入助手可以为元素或功能块快速选择需要关联的变量名或功能块名，以及导入 ST 语言的操作符、关键字。

使用时，在编辑区或监视列表中，选中一个变量或功能块名，按下 F2，在弹出的帮助管理器中选择目标变量或功能块，将其导入，替换原变量或功能块。

如果选择的是 PRG 局部变量，则返回文本为：POU 名.局部变量名的形式。

也可以选中块元件名 AND，或在监视列表中选中变量名，使用输入助手，方法同上。

在 ST 编辑区使用输入助手时，按 F2 时，如果没有选中文本，则在当前光标所在处追加在帮助管理窗口中选中的文本；如果选中了文本，则用帮助管理窗口中选中的文本替换选中的文本。

表 5.5-17 返回文本列表

引用类型	返回文本
引用库中定义函数或工程（库）中自定义函数	返回“???:=FUN(???,???,???)”的形式，其中 FUN 是函数名，???表示该函数的输入参数及返回值
工程中定义程序	返回“PRG_POU()”的形式
引用库中定义功能块或工程（库）中自定义功能块	返回“FB1(in1:=???,in2:=???)”的形式，FB1 为实例名，in1、in2 为输入及输入输出引脚名

5.5.3.18 变量属性

通过【变量属性】菜单项可以快速显示当前选中变量的属性窗口。

在程序编辑区，选中变量元素、变量名，或在 SFC 语言中选中转换，通过以下方式查看变量属性。



- 程序区：右击元素或变量名，单击【变量属性】；
- 菜单栏：单击【编辑】—【变量属性】；
- 快捷键：**Shift+F2**。

弹出该变量对应的“变量声明”对话框，如图 5.5-96 所示。可以修改类别、类型、初始值、变量说明，全局变量也可修改直接地址和变量组列表。



图 5.5-96 变量声明对话框

5.6 硬件配置

系统通过硬件模块采集和处理现场的数据，为了完成采集和控制任务，需要根据具体的工程，对系统的硬件模块进行相应的配置，硬件配置以模块化图形方式显示。

5.6.1 硬件配置文件

AutoThink 软件安装路径下，缺省路径为“C:\AutoThink\V3.1.4_2015.09.15 (V3.1.4B3 版本对应的配置文件)\“目标文件名称 (配置文件)”配置文件描述的是设备相关信息，与硬件模块的型号相关联。



- 上述路径中的部分内容与软件安装路径有关，在此仅作举例参考，请根据工程的实际情况查看。本文中有关路径问题同理，不再赘述。

5.6.1.1 添加配置文件

在硬件组态前，导入需要的硬件配置文件，则在设备库中生成相应的硬件模块供用户组态使用。

模块对应的配置文件均作为系统文件，用户不可任意修改或删除。

导入新的配置文件时，需要保证所导入文件的格式和内容正确无误。



- MC 工程不支持此功能。



- 菜单栏：单击【工程】—【添加配置文件】。

1. LK 平台

在 LK 工程中选择此命令，弹出添加配置文件对话框，如图 5.6-1 所示。

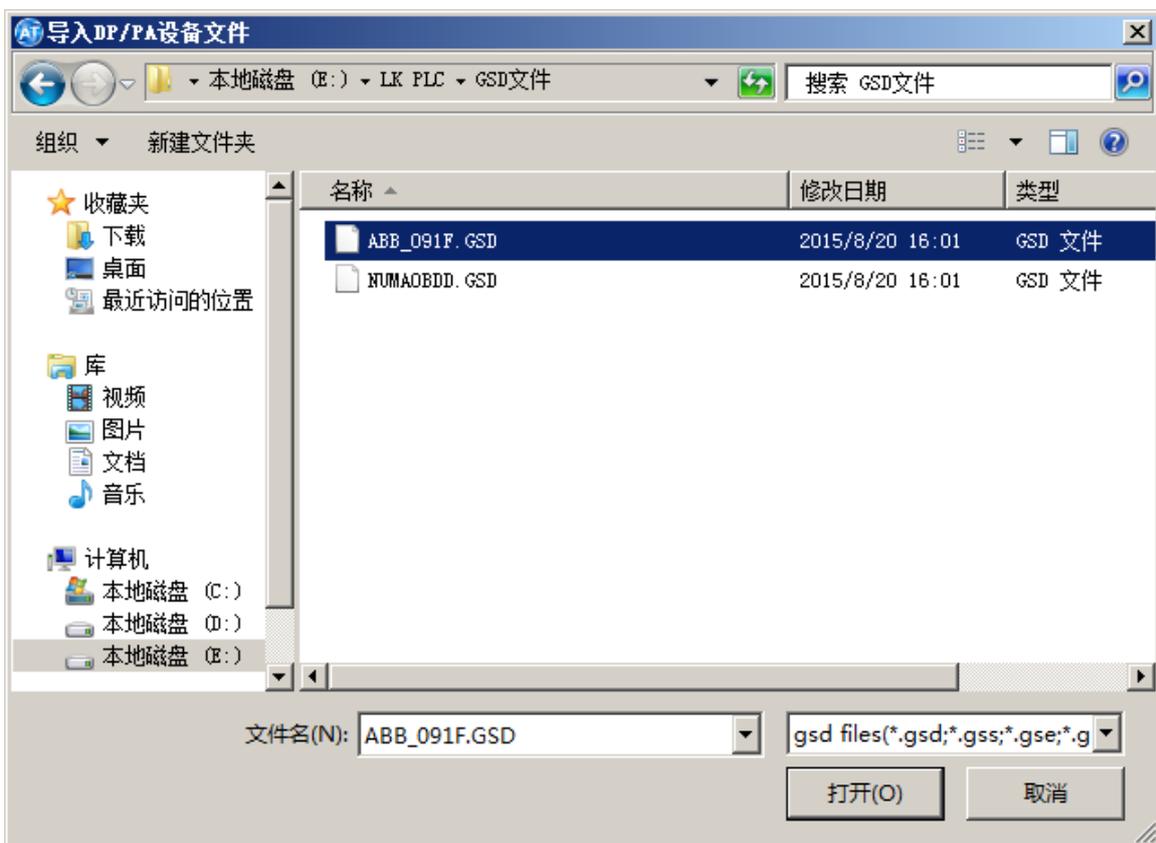


图 5.6-1 添加配置文件对话框

在对话框中选择需要导入的配置文件，单击**打开**，弹出“选择 DP/PA 分类”对话框，如图 5.6-2 所示。

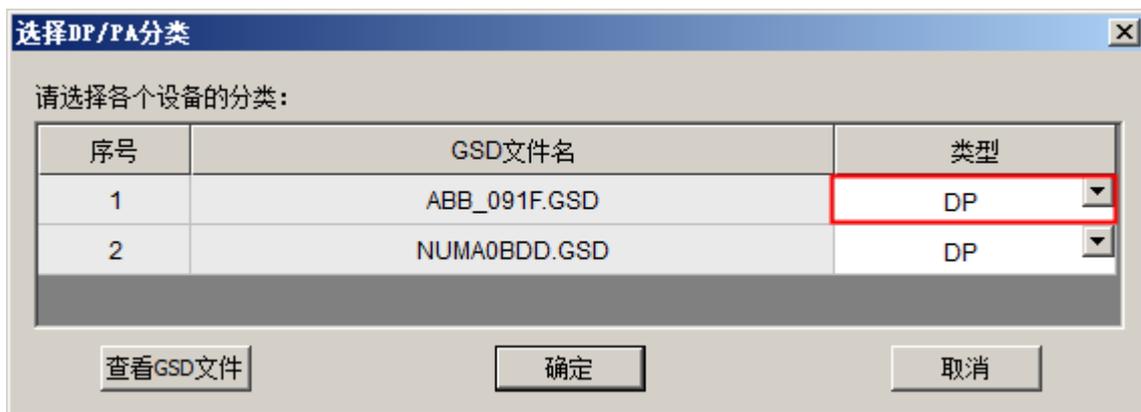


图 5.6-2 选择设备类型

在**类型**项的下拉框中选择设备类型，单击**确定**，则对应的模块配置文件会自动放到配置文件目录下，同时设备库更新。在进行硬件配置时，可以对导入的第三方设备进行组态。

2. LE 平台

LE 工程中选择此命令，弹出“添加配置文件”窗口，如图 5.6-3 所示。在硬件配置文件路径下，选择需要导入模块的.AHC 文件，单击**打开**按钮，设备库下生成导入的模块。

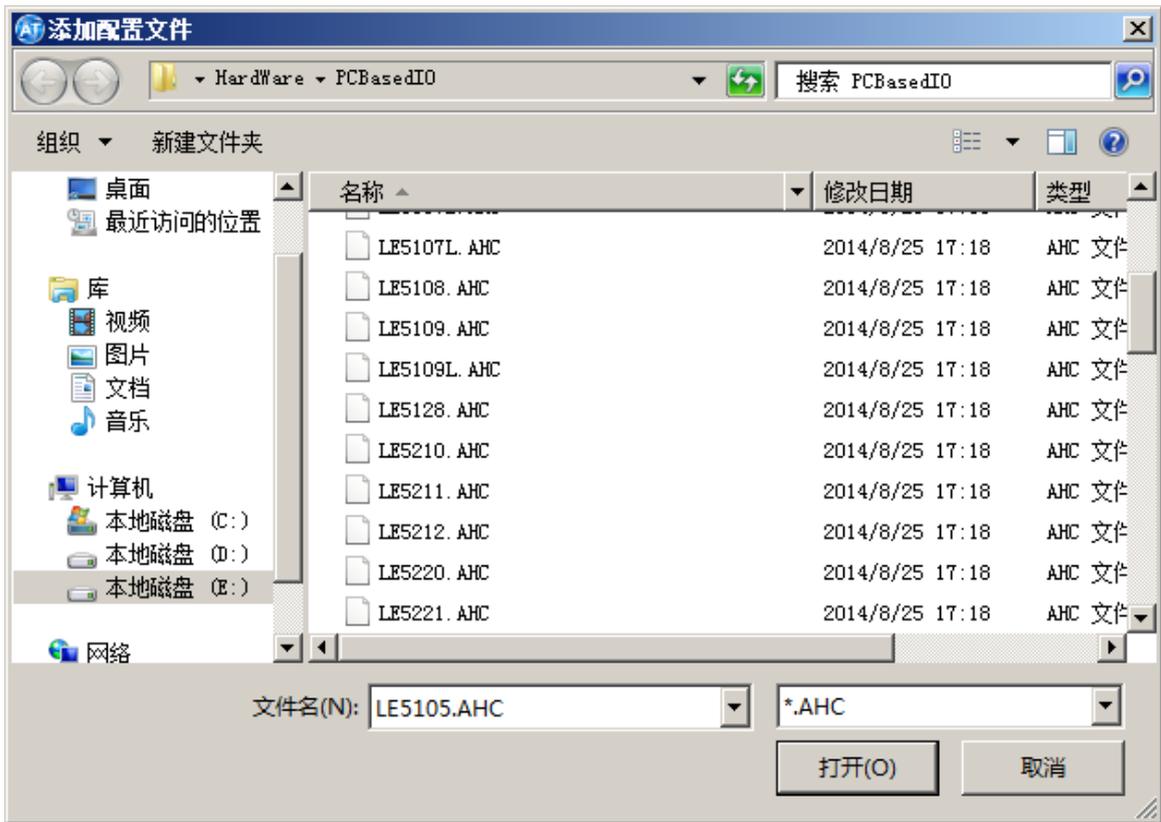


图 5.6-3 添加配置文件对话框



- LE 设备库中已导入所有的 LE 平台支持的硬件模块，用户不需要单独导入。

5.6.2 硬件配置窗口

MC、LE 工程的“硬件配置”窗口通过【硬件配置】节点打开。LK 工程的硬件配置是通过【硬件配置】节点完成。以 MC 工程为例说明硬件配置窗口。

双击工程管理树的【硬件配置】节点，在工作区域打开“硬件配置”编辑窗口，如果是首次打开“硬件配置”编辑窗口会同时自动加载“设备库”窗口，如图 5.6-4 所示。



图 5.6-4 MC 平台“硬件配置”窗口

“硬件配置”的编辑窗口的上部为模块布置区域，可以配置工程的硬件模块；下部为参数信息区域（模块参数信息、通道参数信息、通讯参数信息这 3 个标签页），该参数信息区域各标签页内容对应显示上部选中的模块信息。

5.6.3 MC/LE 硬件组态

MC、LE 工程采用图形模块化的硬件组态方式，从“设备库”中直接拖动方式添加硬件模块，组态方式简单方便、直观形象。以 MC 为例进行说明。

5.6.3.1 添加模块

工程组态时必须配置 CPU 模块，并且一个控制站中只能添加一个 CPU 模块。

拖动 CPU 模块 MC1008、AI 模块 LE5310/LE5311 到“硬件配置”窗口中，如图 5.6-5 所示。

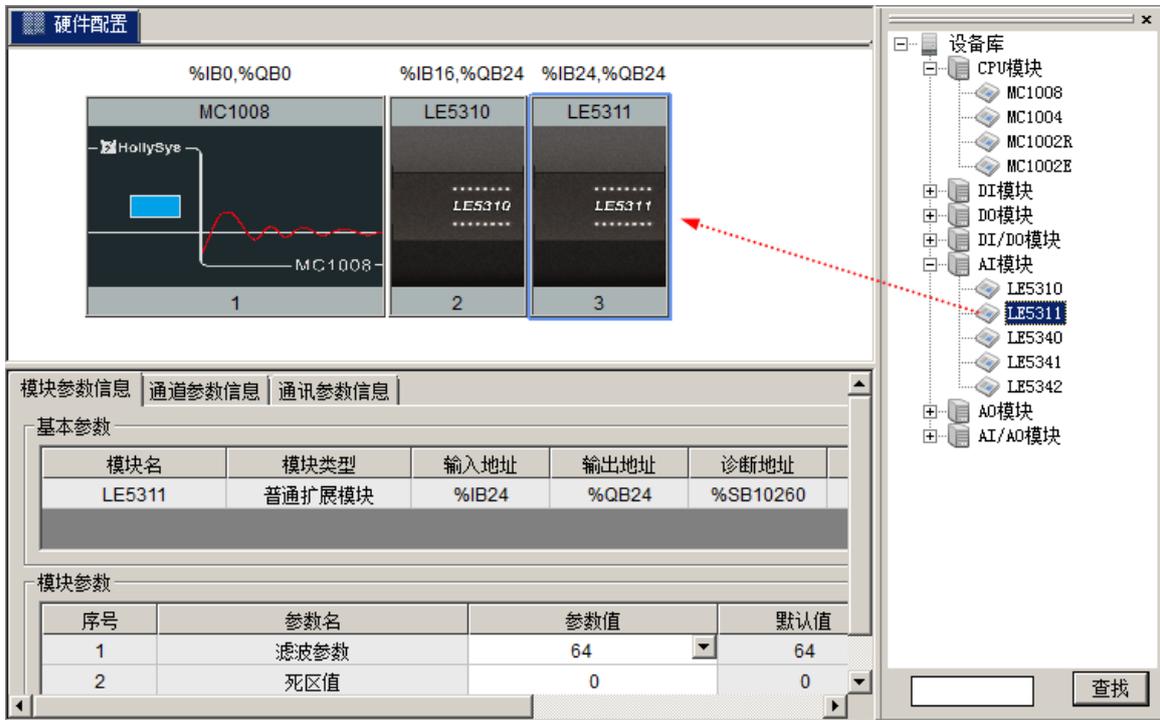


图 5.6-5 添加模块

CPU 自动添加到硬件配置的图形显示区的最左侧，在窗口下端显示该模块的【模块参数信息】、【通道参数信息】、【通讯参数信息】标签页，以表格形式显示相应的信息内容。

MC 工程支持添加 LE 类型的扩展 I/O 模块，支持的类型如表 5.6-1 所示，添加时需要在【通讯参数信息页】中将 COM3 或 COM4 的协议方式设置为 LE IO_Master。COM4 协议方式默认为 LE IO_Master，使用时可直接添加。

表 5.6-1 MC 工程支持添加的 LE 类型扩展 I/O 模块

模块类型	产品型号
数字量 I/O 模块	LE5210、LE5211、LE5212、LE5220、LE5221、LE5223、LE5224、LE5230、LE5231
模拟量 I/O 模块	LE5310、LE5311、LE5320、LE5321、LE5330、LE5340、LE5341、LE5342



- 使用 LE 扩展 I/O 模块时，必须配合 MC1901 模块使用，该模块的详细介绍请参见《MC1000 系列可编程控制器系统手册》第 5 章的 MC1901。

5.6.3.2 模块替换

可以对已添加的 CPU 模块或 IO 模块进行更换。选中 CPU 模块或 IO 模块后，选择右键菜单【模块替换】命令，选择要替换的 CPU 类型或 IO 模块类型进行替换，如图 5.6-6、图 5.6-7 所示，则“硬件配置”编辑窗口中的模块被替换。



图 5.6-6 替换 CPU 模块



- 将 MC1002E 替换为其它 CPU 模块时，EtherCAT Master 节点下的从站组态全部删除，再替换回 MC1002E 时，EtherCAT Master 节点下信息已被删除。
- CPU 模块替换时，已添加 LE 扩展模块的地址会重新分配。

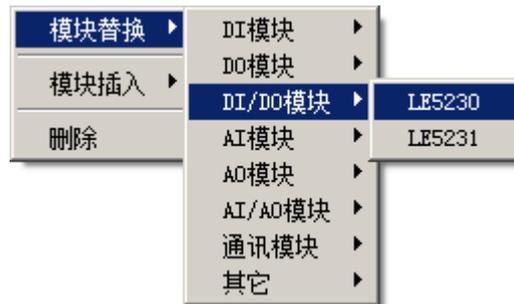


图 5.6-7 IO 模块的替换操作

5.6.3.3 模块插入

IO 模块可以进行模块插入操作。当执行【模块插入】操作时，新的模块添加到当前选中模块的左侧。如图 5.6-8 所示，选中模块 LE5310，右键菜单执行【模块插入】命令，插入 LE5320 模块。

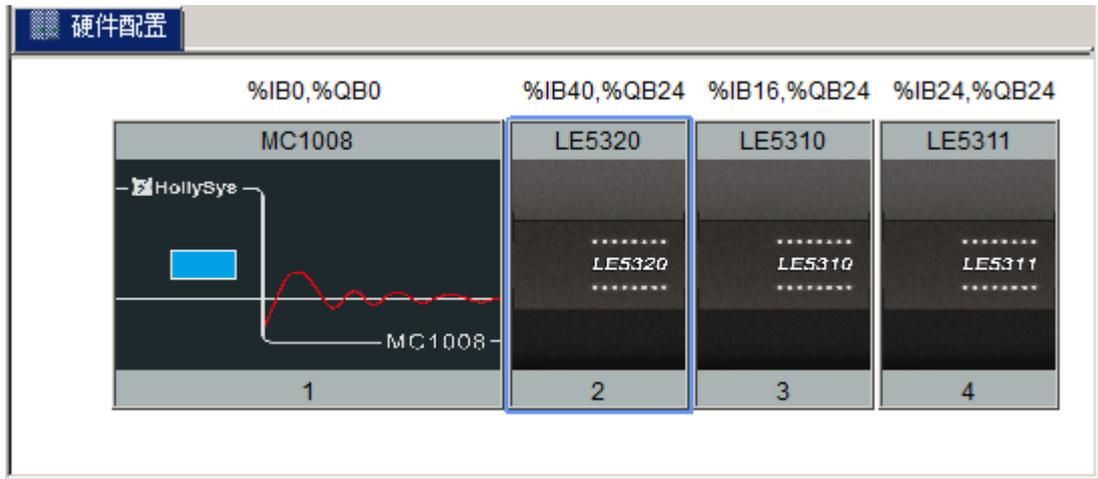


图 5.6-8 插入模块

通过【删除】命令，或者按 **Delete** 键可以删除当前选中的模块。



- IO 模块的增删改只影响本模块，不会引起所有模块的数据区地址重新分配。

5.6.3.4 模块参数信息

在“硬件配置”窗口下部显示当前选中模块的【模块参数信息】。以 LE5310 模块为例，【模块参数信息】如图 5.6-9 所示。

模块参数信息					
基本参数					
模块名	模块类型	输入地址	输出地址	诊断地址	规格说明
LE5310	普通扩展模块	%IB12	%QB16	%SB10260	4通道模拟量输入
模块参数					
序号	参数名	参数值	默认值	最大值	最小值
1	滤波参数	64	64		
2	死区值	0	0	4080	0

图 5.6-9 模块参数信息

在该属性页中显示**基本参数**、**模块参数**。

(1) 基本参数

每个硬件模块都有**基本参数**，显示该模块的模块名、模块类型、输入地址、输出地址、诊断地址、规格说明。

其中**输入地址**和**输出地址**分别表示该模块对应 I/O 存储区的起始地址，添加模块时为其分配输入和输出地址；某个模块被删除后，则占用的输入输出区的内存会被回收，可以重新分配给其他新添加的模块。

(2) 模块参数

模块参数对模块的基本参数进行设置，包括**滤波参数**、**死区值**等。根据模块类型的不同，模块参数也各不相同。各模块参数的详细说明请参见各模块使用说明书，此处不再赘述。

5.6.3.5 通道参数信息

添加模块时，已经按照系统命名规则自动添加了通道名称，为了便于访问 I/O 变量，可以根据实际工程的需要修改其变量定义信息。

在【通道参数信息】标签页，单击每一通道，下面显示相应的通道参数信息，如图 5.6-10 所示。

模块参数信息 通道参数信息 通讯参数信息				
通道基本参数				
通道号	通道名称	通道类型	通道地址	通道说明
1	M3_CH_0	WORD	%IW12	通道1
2	M3_CH_1	WORD	%IW14	通道2
3	M3_CH_2	WORD	%IW16	通道3
通道M3_CH_0的参数				
序号	参数名	参数值	默认值	最大值
1	通道输入信号	4-20mA	4-20mA	
2	通道使能	使能	使能	

图 5.6-10 模块的“通道编辑”窗口

通道基本参数显示了该模块的通道号、通道名称、通道类型、通道地址、通道说明。每个通道又包含参数名、参数值、默认值、最大值、最小值属性。

- 通道号：由系统自动生成，从上到下，依次从小到大排序，根据模块类型可分为 6 通道、8 通道、16 通道等，从 1 开始编号。
- 通道名称：根据该模块所处的控制站中的位置自动生成，格式为“M 模块号_CH_通道号”，可进行修改。
- 通道类型：根据模块的通道数据类型自动生成，常见的有 WORD、BOOL、REAL。
- 通道地址：通道分配的地址。
- 通道说明：对通道进行描述说明，可进行编辑。
- 参数名：通道可配置的参数名称。
- 参数值：设置通道接收的信号类型以及通道的使能与禁止。
- 默认值：通道参数的缺省值。

用户在变量声明时，在**直接地址**栏输入**通道地址**，**变量类型**选择必须与**通道类型**一致。可以根据需要在**通道基本参数**中修改通道名称和通道说明。如图 5.6-11 所示。



图 5.6-11 模块的通道参数组态

如图 5.6-12 所示，在**参数值**项组态信号的类型和通道的使能/禁止，信号类型根据现场实际的一次仪表输出信号进行组态。对使用的通道组态为**使能**，通道没有接线或没有使用的可以组态为**禁止**。

通道M3_CH_2的参数

序号	参数名	参数值	默认值
1	通道输入信号	4-20mA	4-20mA
2	通道使能	4-20mA 0-20mA 0-10V	使能

(a)

通道M3_CH_2的参数

序号	参数名	参数值	默认值
1	通道输入信号	4-20mA	4-20mA
2	通道使能	使能 禁止	使能

(b)

图 5.6-12 通道参数值设置

5.6.3.6 通讯参数信息

【通讯参数信息】为 CPU 模块特有属性,对模块进行通讯时的相关参数进行设置。以 MC 工程 CPU 模块 MC1008 为例,【通讯参数信息】如图 5.6-13 所示。



图 5.6-13 MC 工程 CPU 模块通讯参数信息

该页中可对协议方式、通讯端口的波特率、校验位、停止位、数据位、Modbus 从站地址、Modbus 从站应答延时进行设置。通过单击参数值列中数据右侧的下拉列表箭头进行选择。

硬件配置的详细参数设置请参见各模块硬件使用说明书。

5.6.3.7 LE5405 组态

LE5405 模块是 MC TO LE-IO 网关通讯模块，它一端是 Modbus-TCP 从，一端是 LE BUS 主，实现 MODBUS-TCP 从站与 LE BUS 主站的数据交换。MC1004、MC1008 可作为 LE BUS 主站，LE5405 作为 ModbusTCP 从站，需要在 LE5405 从站下添加 IO 模块。

1. 添加 LE5405

控制器下最多可添加 32 个 LE5405 模块。添加操作如图 5.6-14 所示。



(a)



(b)



(c)

图 5.6-14 添加 LE5405

图 5.6-14 (b) 中的**添加个数**范围为当前所能添加的最大个数。添加范围根据已添加的模块个数进行自动调整。输入的个数超过该范围或没有选中 LE5405 模块，则**确定**按钮不可用。

添加的 LE5405 组成：模块名称(:模块型号)，如果名称重复，添加时，默认的模块名称为“LE5405_序号”。

■ 模块参数信息



- 硬件配置：右击 LE5405 节点，选择【打开模块】。
- 硬件配置：双击 LE5405 节点。

打开模块信息窗口，如图 5.6-15 所示。

LE5405 (LE5405)					
模块参数信息 通道参数信息 通讯参数信息					
基本参数					
模块名	模块类型	输入地址	输出地址	诊断地址	规格说明
LE5405	普通扩展模块	%IB12	%QB16	%SB20000	网关模块
模块参数					
序号	参数名	参数值	默认值	最大值	最小值
1	IP地址				
2	单元ID	1	1	247	1
3	端口号	502	502	65535	1

图 5.6-15 LE5405 模块参数设置

- IP 地址：设置 LE5405 的 IP 地址。IP 地址与主站在同一网段即可。设置的 IP 地址显示在 LE5405 节点名称的后边。

LE5405 IP 修改的内容请参见章节 [5.8.7.28 IP 修改 \(LE5405\)](#)。

- 单元 ID：MODBUS TCP 协议单元 ID。
- 端口号：MODBUS TCP 协议端口号。

■ 重命名



- 硬件配置：右击 LE5405 节点，选择【重命名】。



图 5.6-16 重命名对话框

输入新名称，**确定**即可。

重命名规则请参见章节 5.5.3.2 变量的声明规范的变量命名规则。

■ 删除模块



- 硬件配置：右击 LE5405 节点，选择【删除模块】。

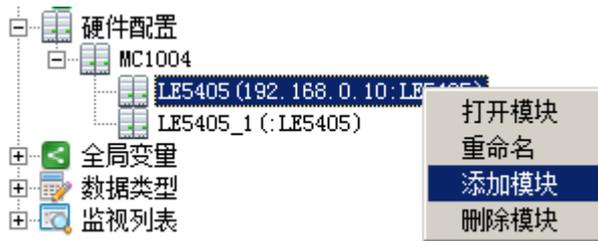
删除选中的 LE5405 模块。

2. IO 模块添加

右击 LE5405 模块，选择【添加模块】进行 IO 模块的添加。如图 5.6-17 所示。

LE5405 下最多能添加 30 个 IO 模块。各 IO 模块能够添加的最大值与模块本身的功耗有关。

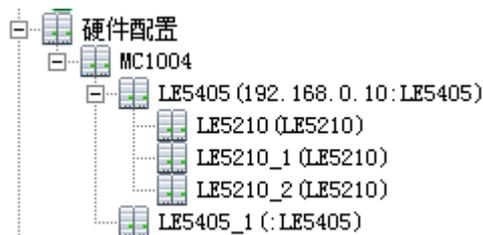
图 5.6-17 (b) 中的**添加个数**范围为当前所能添加的最大个数。输入的个数超过该范围或没有选中 IO 模块，则**确定**按钮不可用。



(a)



(b)



(c)

图 5.6-17 添加 IO 模块

添加的 IO 模块组成：模块名称(:模块型号)，如果名称重复，添加时，默认模块名称为“模块名称_序号”。

可通过拖拽来调整模块在 LE5405 节点下的位置。

在添加 IO 模块时，请确保 IO 模块的顺序与实际硬件连接顺序一致。

■ IO 模块参数设置



- 硬件配置：右击 IO 模块，选择【打开模块】。
- 硬件配置：双击 IO 模块。

打开模块信息窗口，可设置模块参数信息和通道参数信息。在【模块参数信息】标签页可修改参数值。在【通道参数信息】标签页可修改通道名称和通道说明。

默认通道名称组成：MODULE_模块 ID_CHANNEL_序号，其中模块 ID 在添加模块时自动分配。

LE5210 (LE5210)					
模块参数信息 通道参数信息 通讯参数信息					
基本参数					
模块名	模块类型	输入地址	输出地址	诊断地址	规格说明
LE5210	普通扩展模块	%IB12	%QB16	%SB20030	8通道数字量输入模块，DI8×DC24V输入
模块参数					
序号	参数名	参数值	默认值	最大值	最小值
1	通道1滤波参数	不滤波	不滤波		
2	通道2滤波参数	不滤波	不滤波		
3	通道3滤波参数	不滤波	不滤波		
4	通道4滤波参数	不滤波	不滤波		
5	通道5滤波参数	不滤波	不滤波		
6	通道6滤波参数	不滤波	不滤波		
7	通道7滤波参数	不滤波	不滤波		

图 5.6-18 IO 模块参数设置

■ 重命名



- 硬件配置：右击 IO 模块，选择【重命名】。



图 5.6-19 重命名对话框

输入新名称，**确定**即可。

重命名规则请参见章节 5.5.3.2 变量的声明规范的变量名命名规则。

■ 插入模块

在当前选中 IO 模块的前面插入一个新模块。



- 硬件配置：右击 IO 模块，选择【插入模块】。

弹出“添加”对话框，详细的添加过程请参见本节的 IO 模块添加。

■ 删除模块



- 硬件配置：右击 IO 模块，选择【删除模块】。

删除选中的 IO 模块。

3. LE5405DiagGroup

当添加 MC1004 或 MC1008 时，【全局变量】节点下自动生成诊断变量组 LE5405DiagGroup。该变量组显示 LE5405 从站下 IO 模块的诊断信息。变量组内的变量个数与所添加的 LE5405 个数有关，变量名与【硬件配置】节点下 LE5405 的模块名称一致。变量信息在编译后生成。

LE5405DiagGroup						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	LE5405	%SB20000	第 1 个 LE5405 诊断信息	ARRAY [0. ...		FALSE
0002	LE5405_1	%SB20350	第 2 个 LE5405 诊断信息	ARRAY [0. ...		FALSE

图 5.6-20 LE5405 诊断组

双击变量名可查看详细的诊断信息，如图 5.6-21 所示。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	LE5405[0]	%SB20000	LE5405运行状态	BYTE	0	FALSE
0002	LE5405[1]	%SB20001	组态配置的LEIO模块个数	BYTE	0	FALSE
0003	LE5405[2]	%SB20002	在线的LEIO模块个数	BYTE	0	FALSE
0004	LE5405[3]	%SB20003	预留	BYTE	0	FALSE
0005	LE5405[4]	%SB20004	预留	BYTE	0	FALSE
0006	LE5405[5]	%SB20005	预留	BYTE	0	FALSE
0007	LE5405[6]	%SB20006	预留	BYTE	0	FALSE
0008	LE5405[7]	%SB20007	预留	BYTE	0	FALSE
0009	LE5405[8]	%SB20008	预留	BYTE	0	FALSE
0010	LE5405[9]	%SB20009	预留	BYTE	0	FALSE
0011	LE5405[10]	%SB20010	预留	BYTE	0	FALSE
0012	LE5405[11]	%SB20011	预留	BYTE	0	FALSE
0013	LE5405[12]	%SB20012	预留	BYTE	0	FALSE
0014	LE5405[13]	%SB20013	预留	BYTE	0	FALSE
0015	LE5405[14]	%SB20014	预留	BYTE	0	FALSE
0016	LE5405[15]	%SB20015	预留	BYTE	0	FALSE

图 5.6-21 诊断信息

表 5.6-2 诊断变量说明

变量名	变量说明	诊断信息说明
LE5405[0]	LE5405 运行状态	1: 系统初始化开始 2: 空载运行状态 3: 配置文件正确但不带 LEIO 的运行状态 4: 配置文件正确且带 LEIO 的运行状态 5: 配置错误, 包括背板配置错误 (如组态和实际硬件连接不一致) 和加载配置文件错误 6: 通讯过程发生错误的状态 254: 代表 LE5405 配置失败 255: 代表 LE5405 与控制器通讯过程发生错误的状态
LE5405[1]	组态配置的 LEIO 模块个数	显示 LE5405 下组态了的 IO 模块个数
LE5405[2]	在线的 LEIO 模块个数	显示实际连接的模块个数
LE5405[3]~ LE5405[19]	预留	
LE5405[20]	第 1~7 个扩展模块是否异常, 第 0 位忽略	该字节的 Bit0 预留, Bit1~ Bit7 分别代表扩展模块 1~7 的状态值, 0 为正常, 1 为异常

变量名	变量说明	诊断信息说明
LE5405[21]	第 8~15 个扩展模块是否异常	该字节的 Bit0~ Bit7 分别代表扩展模块 8~15 的状态值, 0 为正常, 1 为异常
LE5405[22]	第 16~23 个扩展模块是否异常	该字节的 Bit0~ Bit7 分别代表扩展模块 16~23 的状态值, 0 为正常, 1 为异常
LE5405[23]	第 24~30 个扩展模块是否异常	该字节的 Bit0~ Bit6 分别代表扩展模块 24~30 的状态值, Bit7 预留, 0 为正常, 1 为异常
LE5405[24]~ LE5405[29]	预留	
LE5405[30]~ LE5405[329]	30 个扩展模块的诊断信息, 每个模块占 10 Byte	详细内容请参见下面的扩展模块诊断信息
LE5405[330]~ LE5405[349]	预留	

扩展模块诊断信息:

每个扩展 IO 模块的诊断信息占用 10 个字节, 实际的有效诊断信息字节数与模块类型有关。

(1) 数字量模块诊断信息

数字量模块诊断信息有效字节为 1 Byte, 详细的诊断信息如图 5.6-22 所示。

Bit7	6	5	4	3	2	1	Bit0
CRC	×	×	×	×	×	×	ID

图 5.6-22 数字量诊断信息

第 0 位: Bit0=0, 模块 ID 配置正确; Bit0=1, 模块 ID 配置错 (指本模块 ID 与 CPU 模块收到后返回的模块 ID 是否相同)。

第 7 位: Bit7=0, 扩展模块读取到的数据 CRC 正确; Bit7=1, 扩展模块读取到的数据 CRC 错误。

其它位: 为保留位, 诊断信息中用“×”表示, 统一置为“0”。

(2) 模拟量模块诊断信息

模拟量模块诊断信息由两部分组成: 模块状态信息和通道状态信息。模拟量诊断信息的字节数与模块的类型有关。

■ 模块状态信息

诊断信息的第一个字节为模块状态信息, 详细诊断信息如图 5.6-23 所示。

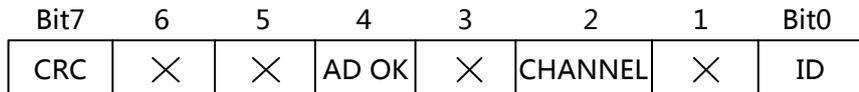


图 5.6-23 模块状态信息

第 0 位：Bit0=0，模块 ID 配置正确；Bit0=1，模块 ID 配置错（指本模块 ID 与 CPU 模块收到后返回的模块 ID 是否相同）。

第 2 位：Bit2=0，无通道错误（所有通道都无错误时，则认为无通道错误）；Bit2=1，有通道错误。

第 4 位：Bit4=0，AD 芯片正常；Bit4=1，AD 芯片异常。

第 7 位：Bit7=0，扩展模块读取到的数据 CRC 正确；Bit7=1，扩展模块读取到的数据 CRC 错误。

其它位：为保留位，诊断信息中用“×”表示，统一置为“0”。

■ 通道状态信息

每 4 位表示一个通道的状态信息，如图 5.6-24 所示。模块型号不同，通道状态信息字节数也不同。

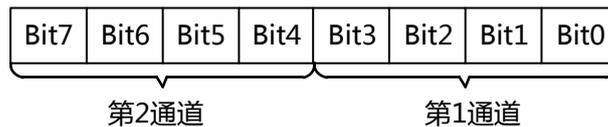


图 5.6-24 通道状态信息

第 0 位：Bit0=0，正常；Bit0=1，超量程下线（仅对 4-20mA 量程有效，当检测到输入电流值小于 4mA，认为超量程下线）。

第 1 位：Bit1=0，正常；Bit1=1，超量程上线（对 4-20mA，0~20mA，0~10V 均有效）。

其它位：保留，保留位置为“0”。

5.6.3.8 通讯协议配置

1. 配置 Modbus 从站

对于 MC 工程，当前控制器作为主站进行 MODBUS 通讯时，需要对从站地址进行配置，在硬件配置窗口中控制器模块上右击，弹出菜单中选择【配置 Modbus 从站】，如图 5.6-25 所示。



图 5.6-25 配置从站命令



- 进行从站配置前，需要在【通讯参数信息】中设置进行通讯的 COM 端口协议方式为 **MODBUS RTU_Master**。

打开配置从站窗口，选择当前进行通讯的端口，并右击**添加**，添加从站至列表当中，如所图 5.6-26 所示，选中列表中的从站名称，可对从站进行**删除**和**查看**操作。



图 5.6-26 配置从站窗口



- 每个端口最多可添加 32 个从站。

选中添加的从站名称，单击**查看**，打开“查看从站”窗口，如图 5.6-27 所示。

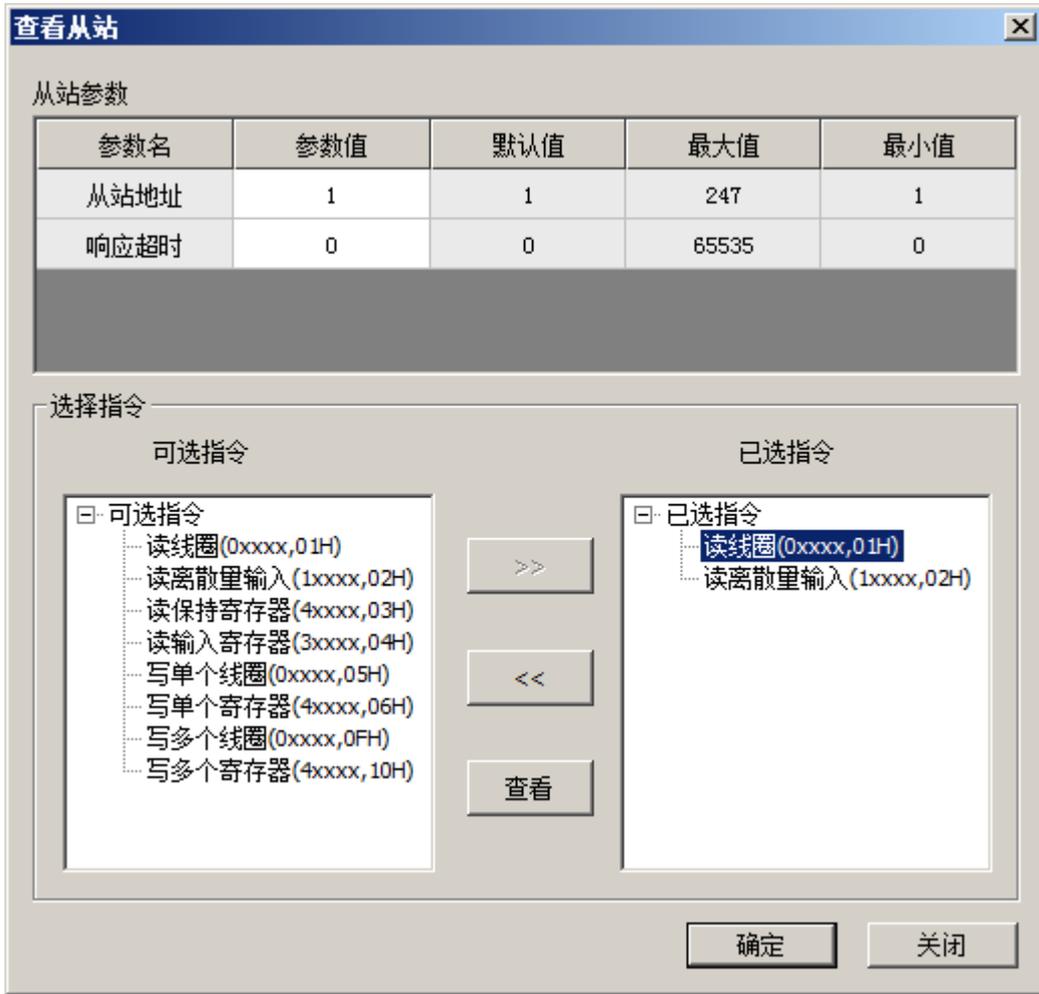


图 5.6-27 查看从站窗口

窗口中双击从站参数列表中的参数值项，可对**从站地址**和**响应超时**时间进行修改。在下方的可选指令列表中选中需要添加的指令类型，单击“>>”按钮，将指令添加至已选指令列表中，或单击“<<”按钮对已选指令进行删除。



- 最多可添加 128 个指令到已选指令列表当中。

指令添加完成后，选中已选指令，单击**查看**按钮，打开查看指令窗口，如图 5.6-28 所示。



图 5.6-28 查看指令窗口

在窗口中,可对**指令参数**和**通道信息**进行设置,可修改的指令参数包括**周期**、**错误处理**、**读写起始地址**、**读写开关量长度**。修改读写开关量长度后,通道信息中将自动生成该长度的通道数量,生成的通道信息中仅**变量名**允许修改。

具体参数设置请参见《MC1000 系列可编程控制器系统手册》。



- Modbus 指令和通道的增删改只影响本模块,不会引起所有模块的数据区地址重新分配。

2. 配置 RTEX 主从站

(1) 离线设置

对于 MC 工程，当添加的 CPU 模块支持 RTEX 协议时，如 MC1002R。在工程管理树窗口的【硬件配置】节点下将出现【RTEX Master】主站模块节点，如图 5.6-29 所示。

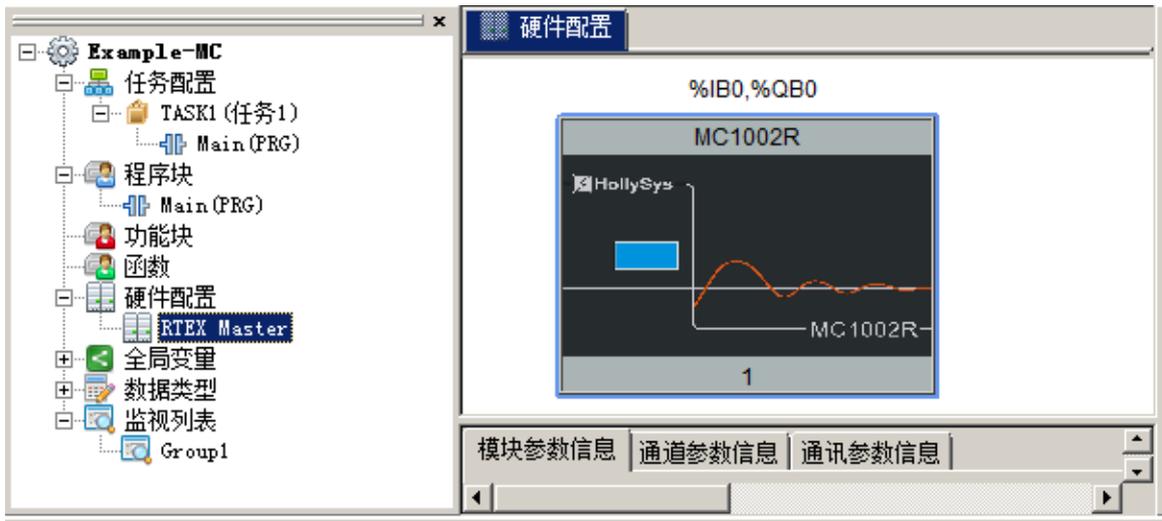


图 5.6-29 RTEX 主站

双击该节点，在工作区打开“RTEX Master”窗口，可对主站模块的通信周期、命令更新周期、通信数据大小、内部轴同步模式等参数信息进行查看和设置，如图 5.6-30 所示。



图 5.6-30 主站模块参数信息



- 松下 MINAS-A5N 驱动器的参数 Pr7.20 RTEX 通讯周期必须设定为 0.5 ms。(目前 MC1002R 的 RTEX Master 配置通讯周期只支持 0.5 ms，如果 A5N 和 MC1002R 的通讯周期不一致会报错)。

与从站进行通讯时，在该节点上单击右键，在弹出菜单中选择【添加从站】，添加的从站节点如图 5.6-31 所示。



图 5.6-31 添加 RTEX 从站

从站节点默认以 A5N_0(0:A5N)形式显示，其中括号前的字符串代表从站别名，括号中的数字代表从站号，冒号后面的字符串代表从站类型。



- 最多可以添加 32 个 RTEX 从站，站号为 0~31。

双击添加的从站节点，可以对从站模块的参数信息进行查看，如图 5.6-32 所示。用户可根据实际从站的设备型号和轴号设置**模块参数**的**参数值**。



图 5.6-32 从站模块参数信息

参数设置的详细内容请参见《MC1000 系列可编程控制器系统手册》。

在添加的 RTEX 从站节点上右击，在弹出菜单中可对从站进行删除、修改站号或名称操作，如图 5.6-33 所示。



图 5.6-33 RTEX 从站操作

设置完成后，执行编译操作，对主从站配置信息的合法性进行检查。

(2) 在线调试

下装完成后，在线监视状态下，可对 RTEX 从站的参数进行调试。在添加的 RTEX 从站节点上右击，在弹出菜单中选择【参数调试】，如图 5.6-34 所示。



图 5.6-34 RTEX 从站参数调试

打开的“参数调试”对话框中包含【状态】、【监视】、【参数】三个标签页。

a. 状态

【状态】标签页如图 5.6-35 所示。

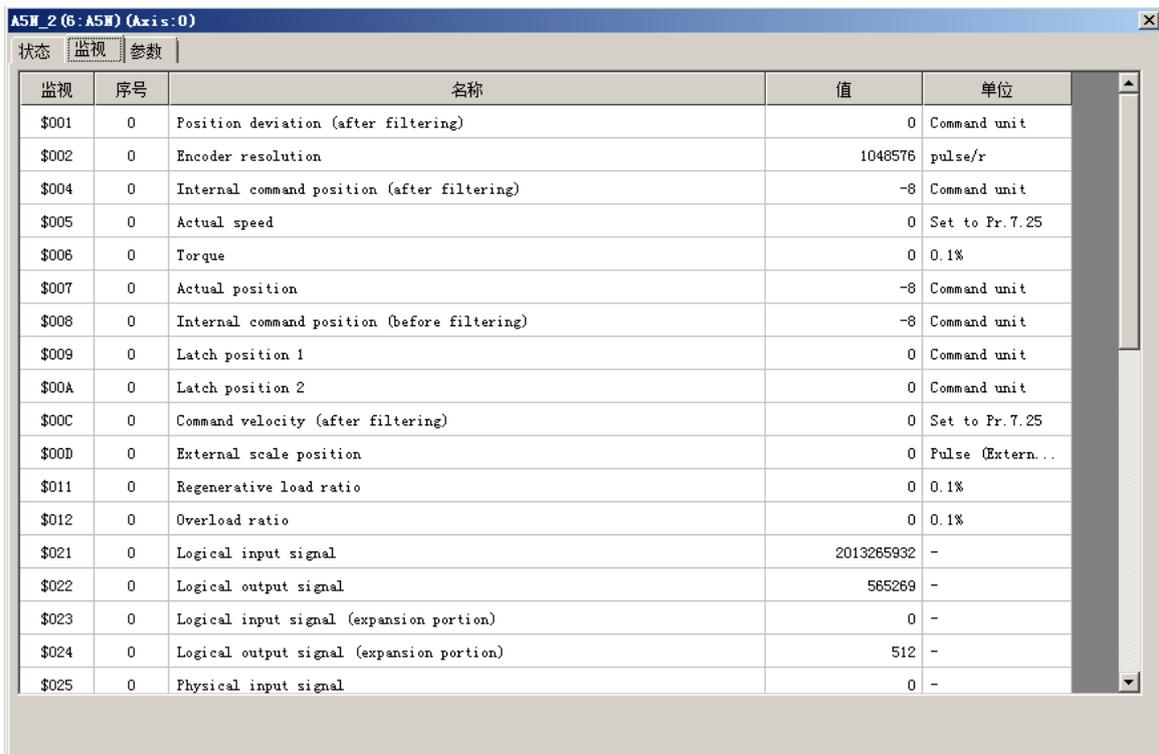


图 5.6-35 参数调试一状态

- 驱动器信息和电机信息：包含驱动器模块型号、序列号、固件版本信息、电机的模块型号和序列号信息。
- 状态：显示当前驱动器状态。
- 运动过程数据：显示位置、速度和力矩信息。
- 报警：显示最近的 15 个历史报警信息。单击下方的按钮，可以实现清除当前报警、清除历史报警和清除外部报警。

b. 监视

【监视】标签页中显示了 RTEX 协议规定的所有监视信息，包括名称、(实际)值和单位，如图 5.6-36 所示。



监视	序号	名称	值	单位
	\$001	0	Position deviation (after filtering)	0 Command unit
	\$002	0	Encoder resolution	1048576 pulse/r
	\$004	0	Internal command position (after filtering)	-8 Command unit
	\$005	0	Actual speed	0 Set to Pr. 7.25
	\$006	0	Torque	0 0.1%
	\$007	0	Actual position	-8 Command unit
	\$008	0	Internal command position (before filtering)	-8 Command unit
	\$009	0	Latch position 1	0 Command unit
	\$00A	0	Latch position 2	0 Command unit
	\$00C	0	Command velocity (after filtering)	0 Set to Pr. 7.25
	\$00D	0	External scale position	0 Pulse (Extern...
	\$011	0	Regenerative load ratio	0 0.1%
	\$012	0	Overload ratio	0 0.1%
	\$021	0	Logical input signal	2013265932 -
	\$022	0	Logical output signal	565269 -
	\$023	0	Logical input signal (expansion portion)	0 -
	\$024	0	Logical output signal (expansion portion)	512 -
	\$025	0	Physical input signal	0 -

图 5.6-36 参数调试—监视

其中，（实际）值实时刷新，不支持修改。

c. 参数

【参数】标签页中显示了 RTEX 协议规定的所有参数信息，如图 5.6-37 所示。

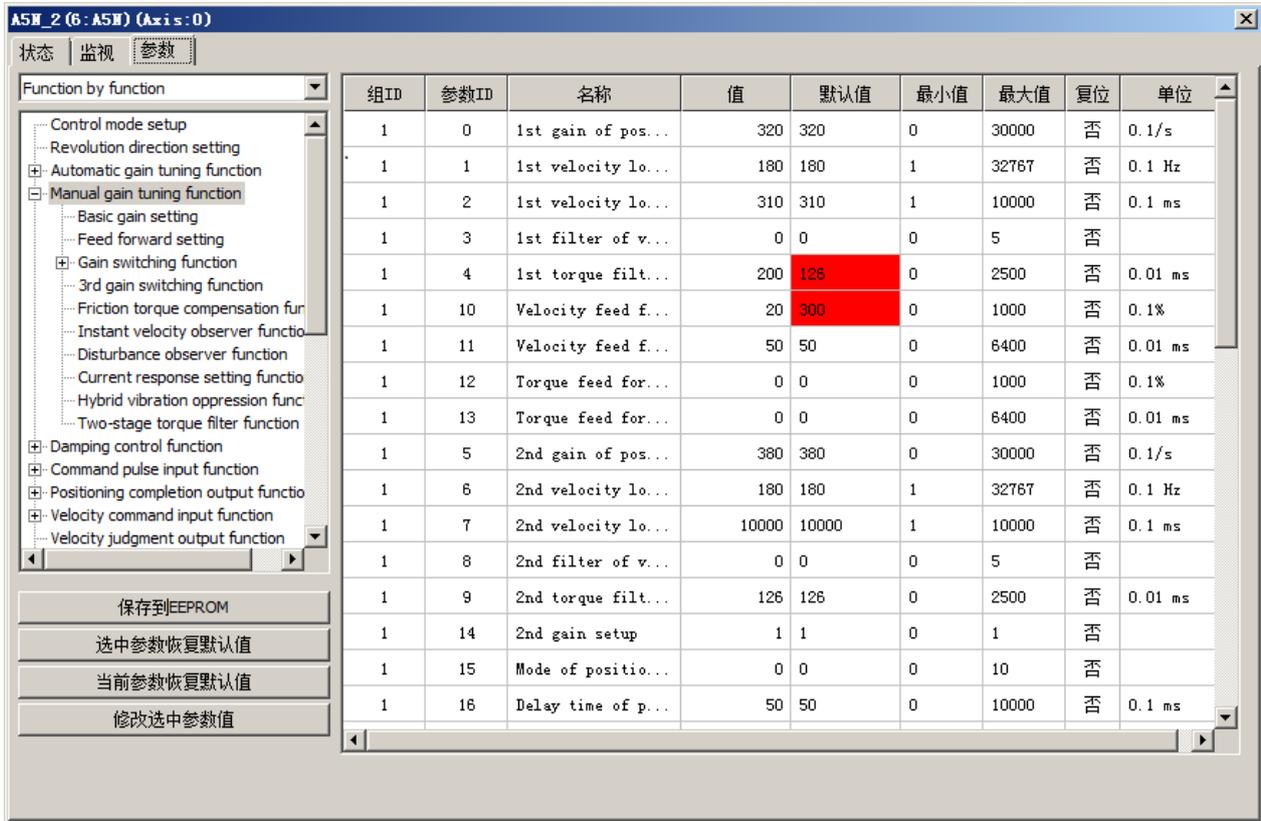


图 5.6-37 参数调试—参数

单击标签页左上角的下拉列表，可对参数项进行分组显示。其中值可进行修改，当默认值与值不同时，默认值单元格以红色显示。

- 保存到 EEPROM：设置参数驱动器掉电保持。
- 选中参数恢复默认值：将选中行中参数的值设置为默认值。
- 当前参数恢复默认值：将当前页面中所有参数的值设置为默认值。
- 修改选中参数值：选中参数中的值，单击该按钮，弹出“写入”对话框，如图 5.6-38 所示。



图 5.6-38 参数调试—写入

在写入值一栏输入数值后，单击确定，完成修改。修改完成后，单击保存到 EEPROM 按钮设置参数驱动器掉电保持。如该参数复位列为否，则修改后立即生效；如复列为是，则需要复位控制器才能使参数生效。

(3) 复位协议主站

使用【复位协议主站】命令，可以复位协议主站及其所有从站。使用该复位命令会触发调用功能块 RTEX_BusReset（复位协议主栈），实现协议主站复位的目的。功能块的详细内容请参见《MC1000 系列多轴运动控制器指令手册》中第 5 章的 RTEX_BusReset（复位协议主栈）。

当总线断线或从站驱动器断电等原因导致协议栈异常时，通过该命令对协议主站进行复位。



- 工程管理树：右击【硬件配置】节点下的 RTEX Master 主站，单击【复位协议主站】。

弹出确认框，如图 5.6-39 所示。

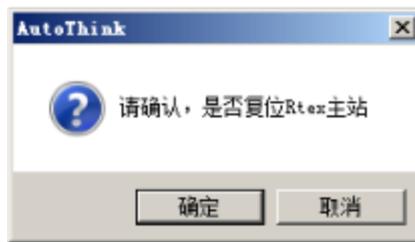


图 5.6-39 复位 RTEX 主站确认框

单击**确定**，对协议主站进行复位。

(4) 复位所有驱动器

在线状态，使用【复位所有驱动器】命令，可以复位所有从站驱动器。使用该复位命令会触发调用功能块 RTEX_DriveResetAll（复位所有驱动器），实现所有从站驱动器复位的目的。功能块的详细内容请参见《MC1000 系列多轴运动控制器指令手册》中第 5 章的 RTEX_DriveResetAll（复位所有驱动器）。



- 工程管理树：右击【硬件配置】节点下的 RTEX Master 主站，单击【复位所有驱动器】。

弹出确认框，如图 5.6-40 所示。



图 5.6-40 复位 RTEX 从站确认框

单击**确定**，对所有的驱动器进行复位。

(5) RTEX 诊断信息

当添加 MC1002E 时，在工程管理树的【全局变量】节点下将出现【RTEXDiagGroup】节点，如图 5.6-41 所示。



图 5.6-41 RTEXDiagGroup 变量组

该变量组包含了 RTEX 主站模块的所有诊断信息。双击该节点，在右侧工作区域中显示了包含的所有变量内容，如图 5.6-42 所示。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	StackStatus	%SB10560	协议栈状态码	USINT	0	FALSE
0002	MNMSStatus	%SB10561	专有芯片状态码	USINT	0	FALSE
0003	ResStatus0	%SB10562	预留	USINT	0	FALSE
0004	ResStatus1	%SB10563	预留	USINT	0	FALSE
0005	MainErrcode	%SB10564	协议栈主错误码	UINT	0	FALSE
0006	Errcode0	%SB10566	协议栈子错误码0	UINT	0	FALSE
0007	Errcode1	%SB10568	协议栈子错误码1	UINT	0	FALSE
0008	Errcode2	%SB10570	协议栈子错误码2	UINT	0	FALSE
0009	Errcode3	%SB10572	协议栈子错误码3	UINT	0	FALSE
0010	Errcode4	%SB10574	协议栈子错误码4	UINT	0	FALSE
0011	SendFrameSum	%SD10576	总线发送帧总数	UDINT	0	FALSE

图 5.6-42 RTEXDiagGroup 变量

在线状态下，可对变量的实际值进行查看，了解主站模块状态。

3. 配置 EtherCAT 主从站

MC 工程中，CPU 配置为 MC1002E 时，可以作为主站与其它从站通过 EtherCAT 总线协议进行通讯。在硬件配置节点下生成【EtherCAT Master】节点，在该节点下添加从站。如图 5.6-43 所示。

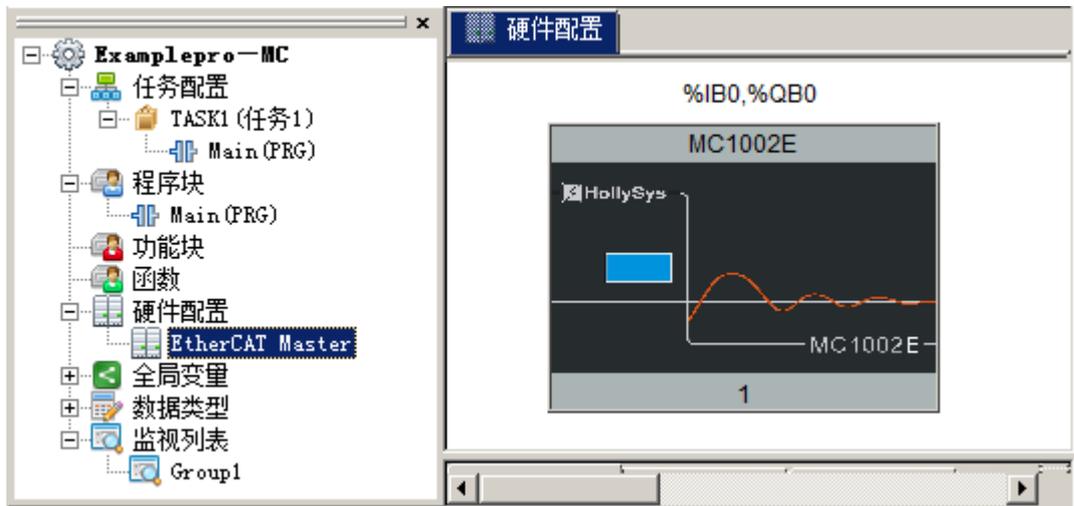


图 5.6-43 EtherCAT Master 节点

(1) EtherCAT 主站配置

双击【EtherCAT Master】主站节点，在右侧区域打开主站配置界面，如图 5.6-44 所示。此界面包括【模块参数】和【信息】两个标签页。

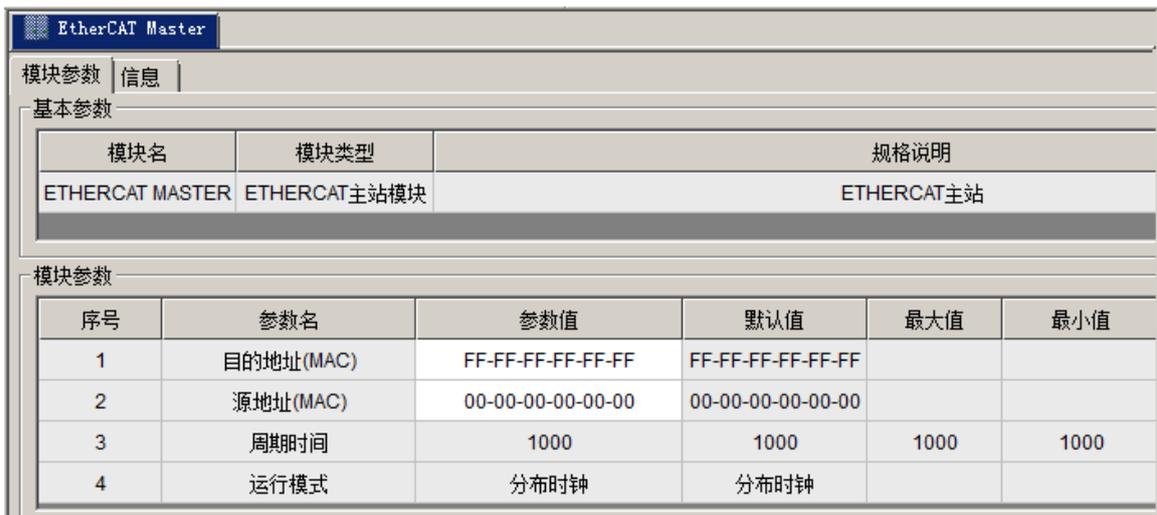


图 5.6-44 EtherCAT 主站配置界面

- 模块参数：该标签页中可设置目的地址、源地址。周期时间和运行模式不可编辑。
- 信息：显示主站设备的名称、设备型号、厂商等信息。

(2) EtherCAT 从站配置

a. 导入 DDF 文件

在添加从站之前，先要导入 DDF 文件。导入后，在添加从站时，“添加从站”对话框中显示已导入的设备名称，用户可根据实际从站的型号进行选择。以伺服器从站配置为例进行阐述。



- 工程管理树，右击【EtherCAT Master】主站节点，单击【导入 DDF 文件】。



图 5.6-45 导入 DDF 文件操作

弹出“打开”对话框，选择需要导入的 DDF 文件，如图 5.6-46 所示。

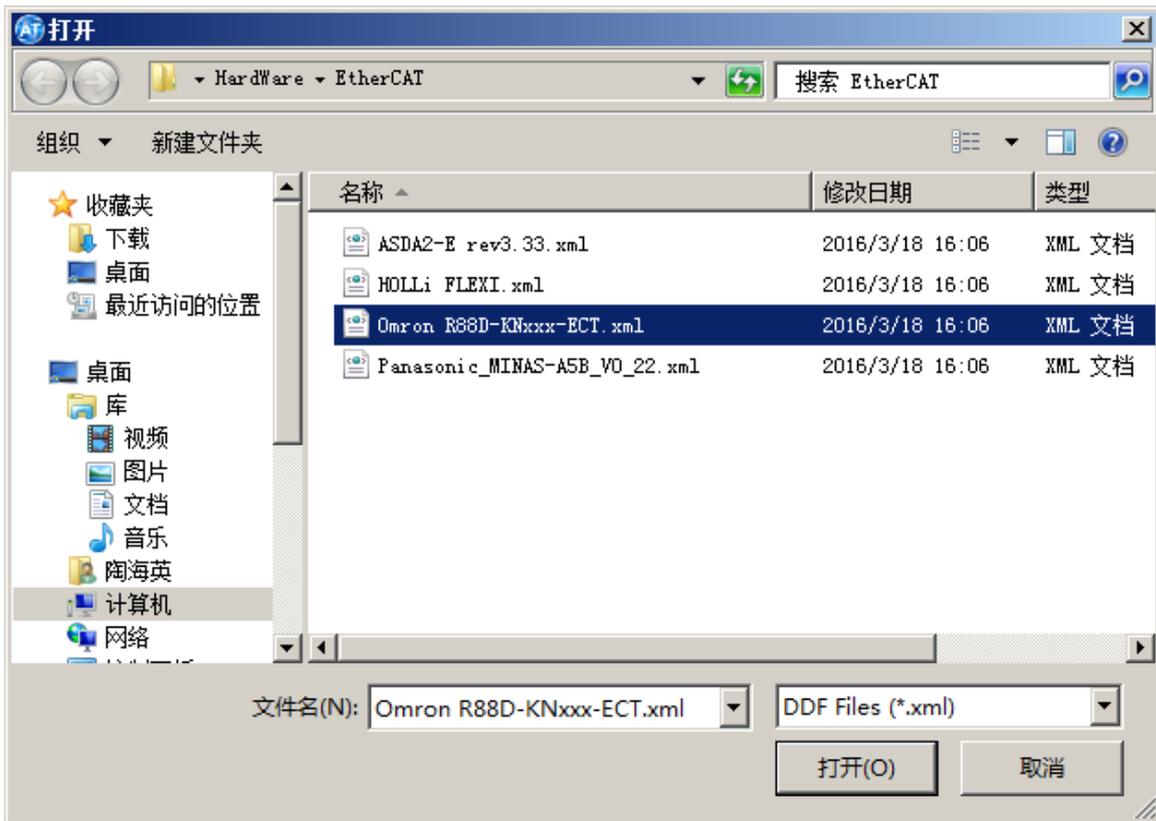


图 5.6-46 选择 DDF 文件

单击打开按钮，弹出文件导入成功的提示框，如图 5.6-47 所示。



图 5.6-47 导入成功提示框

单击**确定**，完成 DDF 文件导入。

b. 添加从站



- 工程管理树：右击【硬件配置】节点下的【EtherCAT Master】，单击【添加从站】。



图 5.6-48 添加从站操作

弹出“添加从站”对话框，如图 5.6-49 所示。在**供应商**下拉框中显示已导入的设备所属的**供应商**，用户可以进行选择，同时设备列表框中显示该**供应商**支持的**设备型号**。

如果在添加从站之前没有导入 DDF 文件，单击“添加从站”对话框左下角的**导入 DDF 文件**按钮进行导入操作。

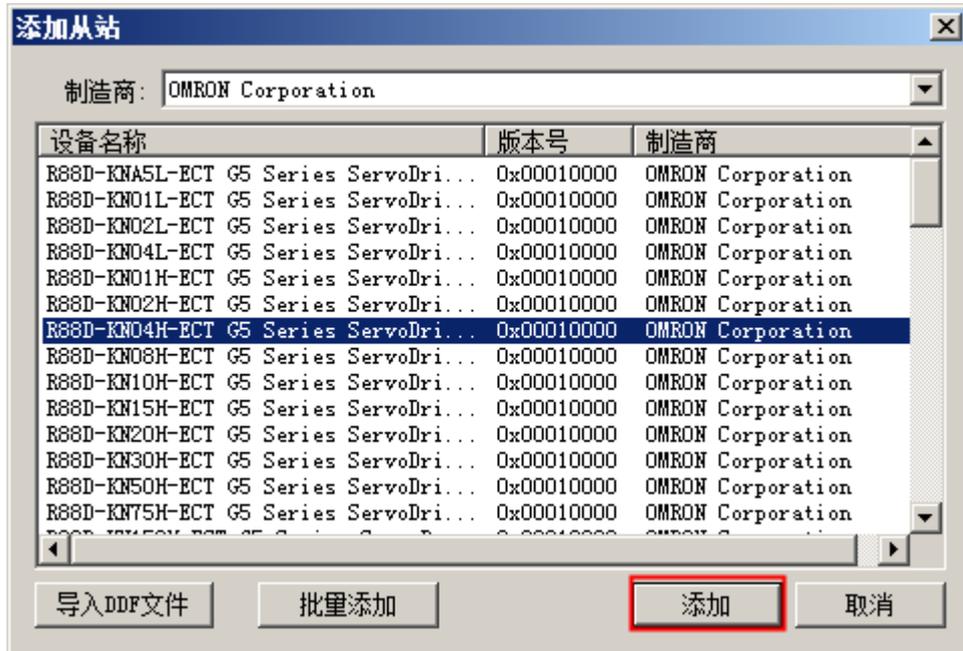


图 5.6-49 选择从站设备

根据实际工程连接的从站设备，选择设备型号，单击**添加**，完成从站的添加。



- EtherCAT 从站最多添加 64 个。

如图 5.6-50 所示，在 EtherCAT Master 节点下生成一个对应的从站，从站名称组成：（从站号：型号：轴号）+从站名_序号。从站号由 256+站号组成，添加从站设备时自动分配站号。可通过从站的右键菜单进行设置，设置范围为 0~63。

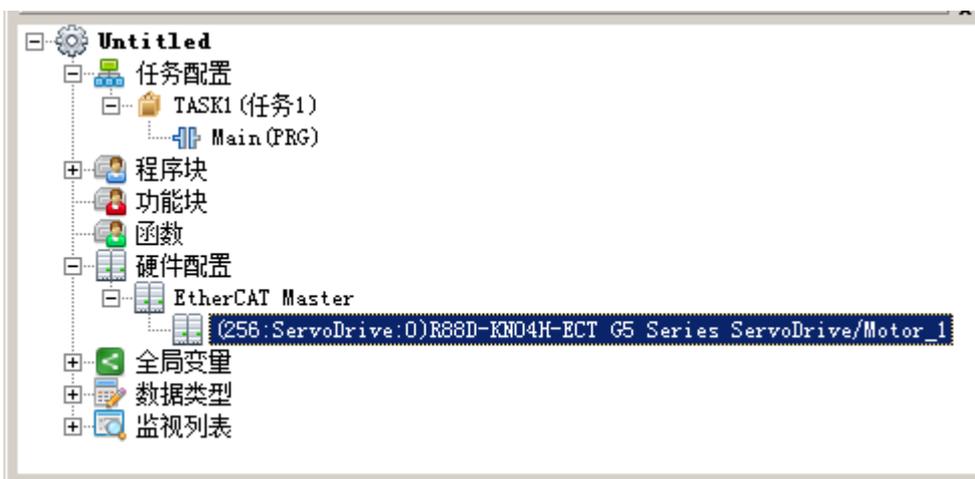


图 5.6-50 从站设备节点一伺服器

c. 批量添加

通过批量添加可添加设定个数的从站。如图 5.6-49 所示，选择要添加的设备，单击**批量添加**，打开“批量添加”对话框，如图 5.6-51 所示。

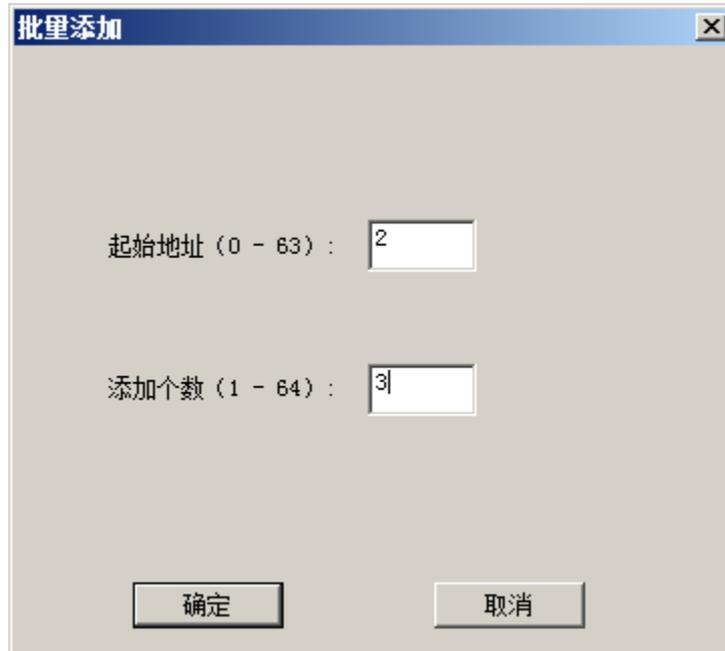


图 5.6-51 批量添加对话框

输入起始地址（站号）和添加个数，单击**确定**，将添加从站号从 258（256+起始地址）开始的 3 个从站，如图 5.6-52 所示。

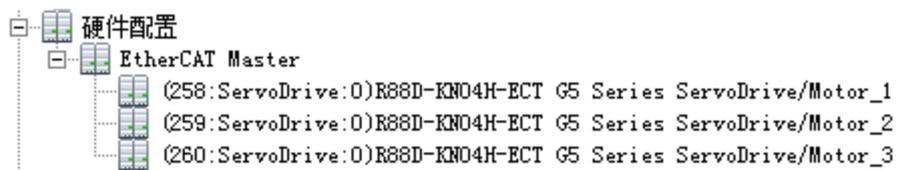


图 5.6-52 批量添加完成

如果输入的起始地址已被占用，则顺序检测未被占用的起始地址作为站号，依次添加从站。

d. 从站的操作

通过从站的右键菜单可以修改从站名称和站号、插入从站、以及删除从站操作，如图 5.6-53 所示。

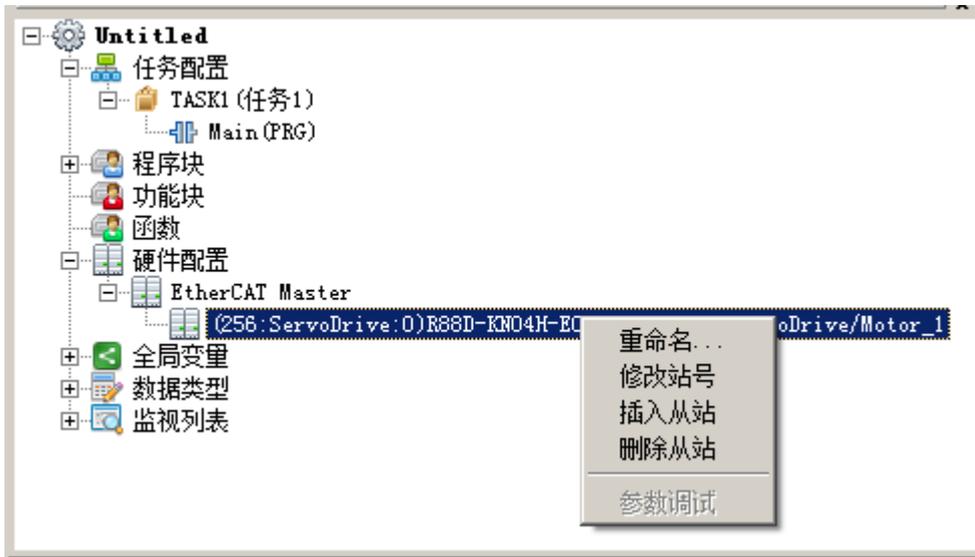
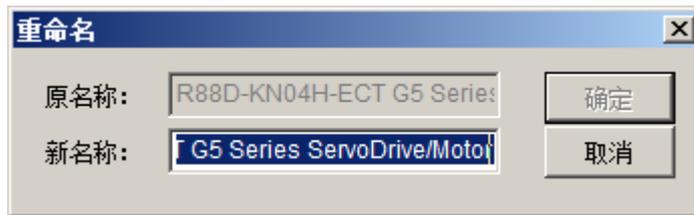


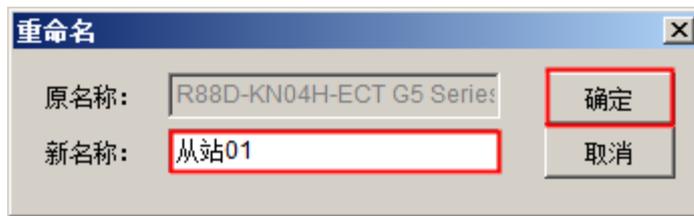
图 5.6-53 从站设备的操作

■ 修改名称

选择【重命名】操作，弹出“重命名”对话框，如图 5.6-54 (a) 所示，新名称输入框中默认显示原名称。如图 5.6-54 (b) 所示，输入新名称从站 01，单击确定，完成名称修改。



(a)



(b)

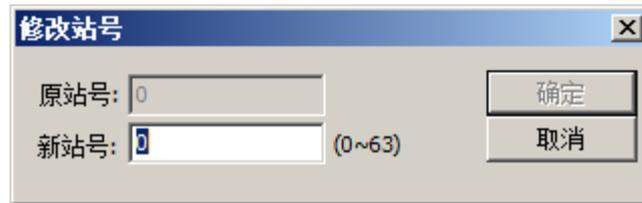


(c)

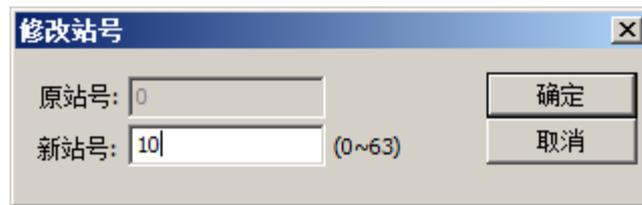
图 5.6-54 修改从站名称

■ 修改站号

选择【修改站号】命令，弹出“修改站号”对话框如图 5.6-55（a）所示。**新站号**输入框中默认显示**原站号**。如图 5.6-55（b）所示，输入新站号 10，单击**确定**按钮，从站号显示修改后的十进制数 266（256+10），如图 5.6-55（c）所示。



(a)



(b)



(c)

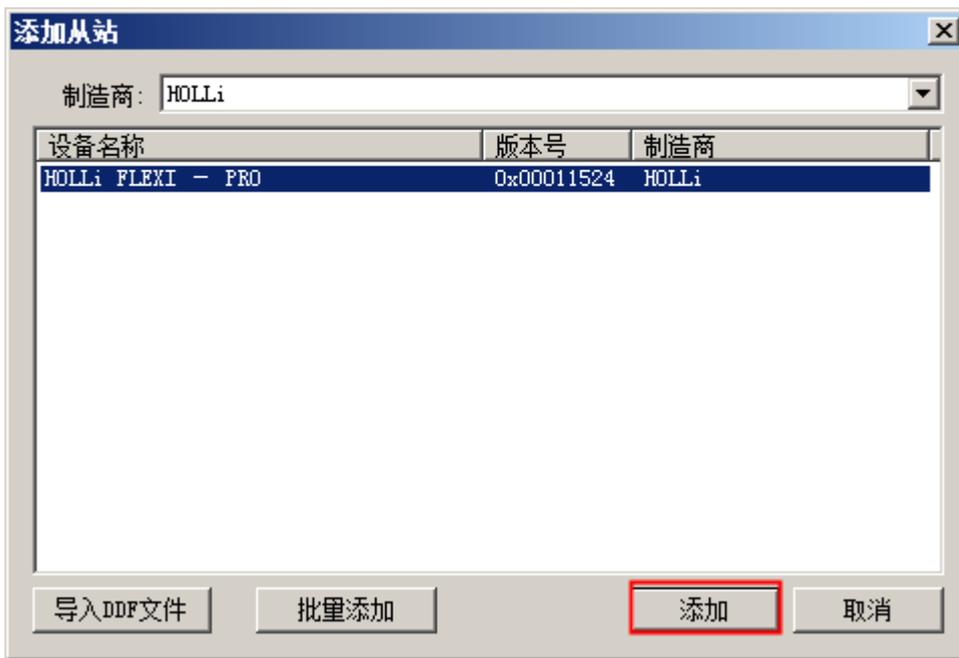
图 5.6-55 修改从站站号

■ 插入从站

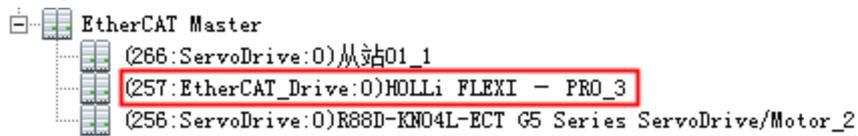
选中要插入从站的节点，右键选择【插入从站】命令，弹出“添加从站”对话框，如图 5.6-56（b）所示。选择要插入的从站设备型号，单击**添加**，在当前选中从站的前面插入新的从站。批量插入时，单击**批量添加**，添加方法请参见本节的批量添加。



(a)



(b)



(c)

图 5.6-56 插入从站

可通过拖拽调整从站的上下位置。

■ 删除从站

选择【删除从站】命令，弹出删除确认提示框，单击**确定**，将从站删除。



图 5.6-57 删除从站提示框

e. EtherCAT 从站参数设置

双击从站节点，在右侧打开从站参数配置界面，如图 5.6-58 所示。此界面包括【模块参数】、【过程参数】、【EtherCAT IO 映射】、【信息】四个标签页。可通过每个标签页进行相关参数的设置。

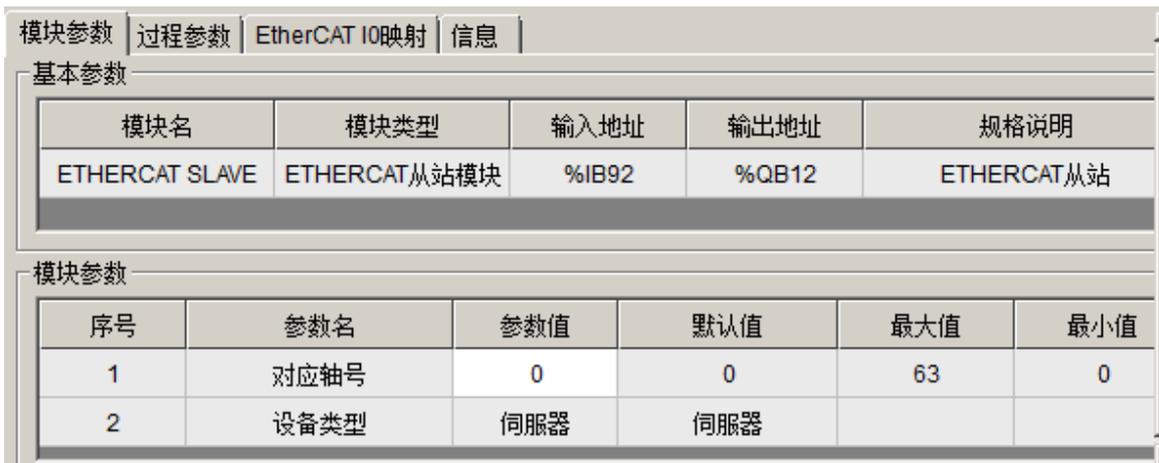


图 5.6-58 从站参数配置界面

- 模块参数：模块参数标签页包含基本参数和模块参数两部分。基本参数显示模块名称、类型、输入/输出地址等信息。模块参数部分显示从站的对应轴号、设备类型。可设置**对应轴号**的参数值。
- 过程参数：在此标签页中显示从站设备支持的所有模块，如图 5.6-59 所示。勾选需要的 PDO Mapping，在【EtherCAT IO 映射】标签页中自动映射已勾选模块的通道地址。

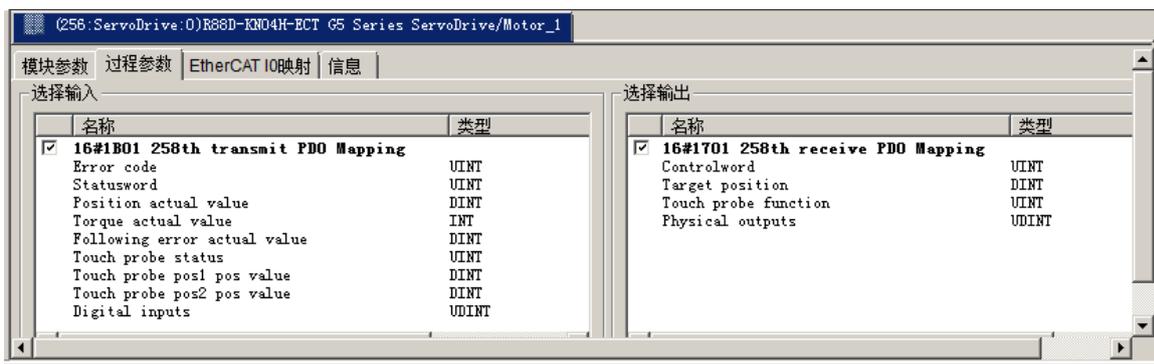


图 5.6-59 过程参数配置

- EtherCAT IO 映射：该标签页显示配置的【过程参数】与 I、O 区的映射关系。修改【过程参数】的配置，则 EtherCAT IO 映射重新分配。

通道号	通道名称	通道类型	字节数	通道地址	通道说明
1	E1_CH_1	UINT	2	%IW92	Error code
2	E1_CH_2	UINT	2	%IW94	Statusword
3	E1_CH_3	DINT	4	%ID96	Position actual value
4	E1_CH_4	INT	2	%IW100	Torque actual value
5	E1_CH_5	DINT	4	%ID104	Following error actual value
6	E1_CH_6	UINT	2	%IW108	Touch probe status
7	E1_CH_7	DINT	4	%ID112	Touch probe pos1 pos value

图 5.6-60 EtherCAT IO 映射

- 信息：显示从站设备的名称、设备型号、厂商等信息。

主从站配置的详细参数设置请参见《MC1000 系列可编程控制器系统手册》。

(3) 在线调试

在线状态时，通过从站右键菜单的【参数调试】命令，可以进行参数的调试。例如：修改值、读取在线值、保存修改值到 EEPROM。如图 5.6-61 所示。

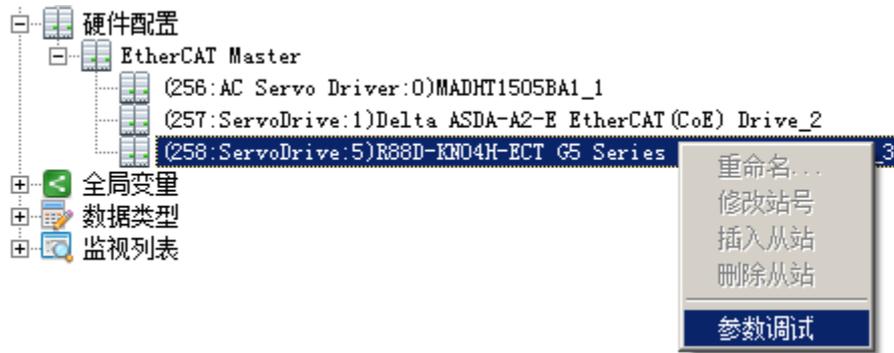


图 5.6-61 选择参数调试命令

打开“参数调试”对话框，如图 5.6-62 所示。



图 5.6-62 参数调试对话框

单击**选择参数**，在弹出的“选择参数”对话框中勾选需要调试的参数，如图 5.6-63 所示。勾选索引项时，下面的子参数默认都被选择。所选的参数（包含子参数）不能超过 255，否则单击**确定**时会弹出超限的提示框。



图 5.6-63 选择调试参数

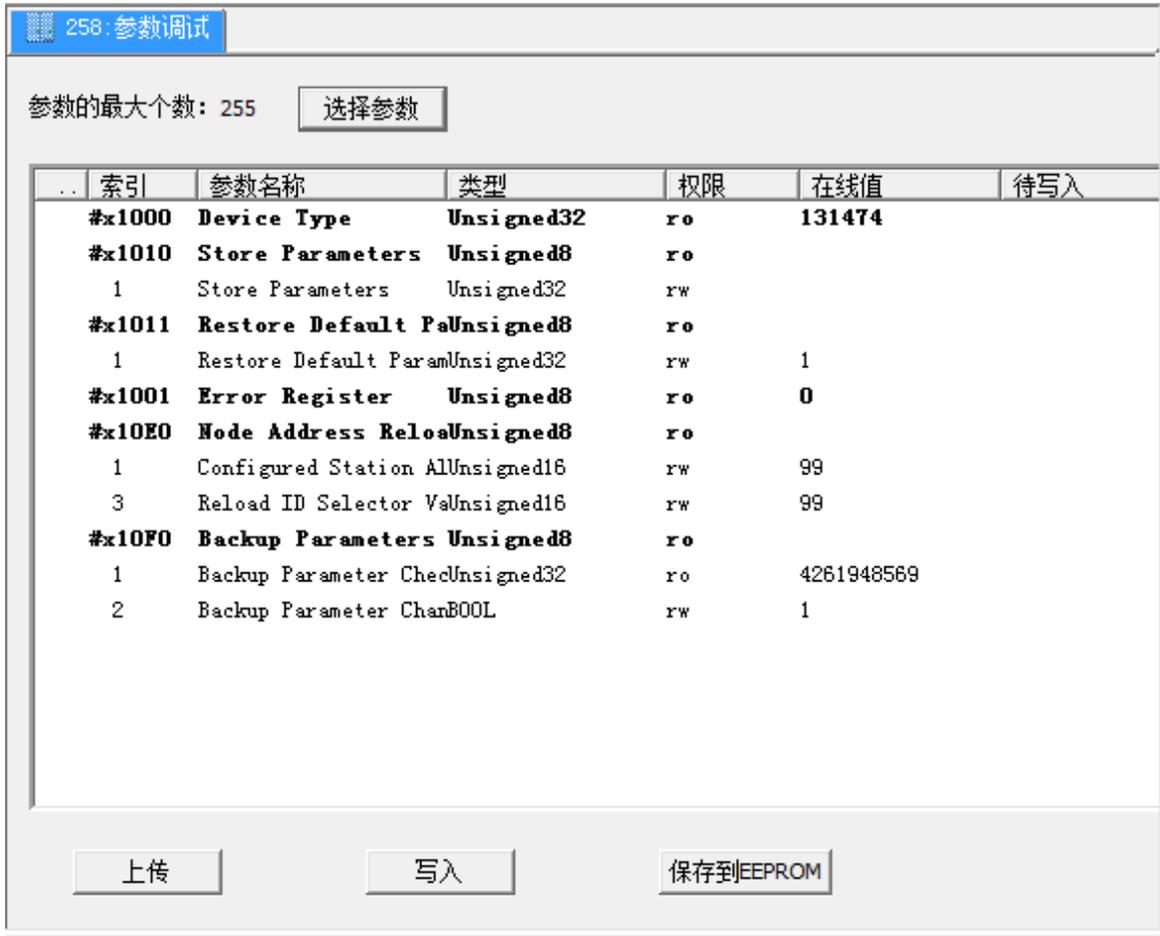


图 5.6-64 调试参数添加完成

■ 上传

单击**上传**，可以从伺服器设备中读取参数的当前值，在线值被更新。

■ 写入

在**待写入**列输入需要写入的值，单击**写入**，该参数值被修改。单击**上传**后，**在线值**被刷新为修改的值。当参数为只读权限时，待写入项不可编辑。

■ 保存到 EEPROM

写入参数值后，单击**保存到 EEPROM**，该值被备份到 EEPROM 中，掉电后该修改值不会丢失。否则，写入的参数值掉电后恢复为初始值。EtherCAT 诊断信息。

(4) 复位协议主站

在线状态，使用【复位 EtherCAT 主站】命令，可以复位整个 EtherCAT 总线。复位过程包括复位 comX 协议板卡、重新配置从站、初始化协议栈。使用该复位命令会触发调用功能块 EtherCAT_BusReset（复位 EtherCAT 总线），实现 EtherCAT 总线复位的目的。功能块的详细内容可参见《MC1000 系列多轴运动控制器指令手册》中第 6 章的 EtherCAT_BusReset（复位 EtherCAT 总线）。

菜单命令复位操作如下：



- 工程管理树：右击【硬件配置】节点下的 EtherCAT Master 主站，单击【复位 EtherCAT 主站】。

弹出确认框，如图 5.6-65 所示。



图 5.6-65 复位 EtherCAT 总线确认框

单击**确定**按钮，进行 EtherCAT 总线复位。

(5) EtherCAT 诊断信息

当添加 MC1002E 时，在工程管理树的【全局变量】节点下将出现【EtherCATDiagGroup】节点，如图 5.6-66 所示。



图 5.6-66 EtherCATDiagGroup 变量组

该变量组包含了 EtherCAT 主站模块的所有诊断信息。双击该节点，在右侧工作区域中显示了包含的所有变量内容，如图 5.6-67 所示。

EtherCATDiagGroup						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	StackState	%SB10560	协议栈状态机状态码	USINT	0	FALSE
0002	BusState	%SB10561	总线状态码	USINT	0	FALSE
0003	CfgSlaveNum	%SB10562	配置从站个数	USINT	0	FALSE
0004	ActiveSla...	%SB10563	激活从站个数	USINT	0	FALSE
0005	StackReady	%SB10564	协议栈准备状态	USINT	0	FALSE
0006	StackRunning	%SB10565	协议栈运行状态	USINT	0	FALSE
0007	StackComm...	%SB10566	协议栈通信状态	USINT	0	FALSE
0008	StackError	%SD10568	协议栈错误码	UDINT	0	FALSE
0009	StackSubE...	%SD10572	协议栈子错误码	UDINT	0	FALSE
0010	SlaveState	%SB10576	从站状态码	ARRAY[0...		FALSE

图 5.6-67 EtherCATDiagGroup 变量组

离线状态下，可对**变量说明**项进行修改。在线状态下，可对变量的实际值进行查看，了解主站模块状态。

5.6.4 LK 硬件组态

LK 采用右键菜单添加模块的方式进行硬件组态。

5.6.4.1 控制器模块

LK 工程硬件组态方式与 MC、LE 不同。LK 硬件组态是在硬件配置节点下完成。

LK 工程新建完成后，在【硬件配置】节点下缺省有 LK220 控制器模块，控制器默认添加以太网适配器，如图 5.6-68 所示。



图 5.6-68 LK 工程硬件配置节点

通过右键菜单相关命令，可以查看控制器设备信息、进行重命名和设备添加。如图 5.6-69 所示。



图 5.6-69 CPU 模块右键菜单

■ 设备信息

选择【打开】命令，或者双击 LK220 模块，在右侧区域弹出设备信息窗口，同时自动加载设备库。如图 5.6-70 所示。



图 5.6-70 LK220 CPU 模块的设备信息窗口

控制器默认配置为冗余。

■ 重命名

选择【重命名】可以修改控制器的名称，如图 5.6-71 所示。输入新名称。单击**确定**即可。

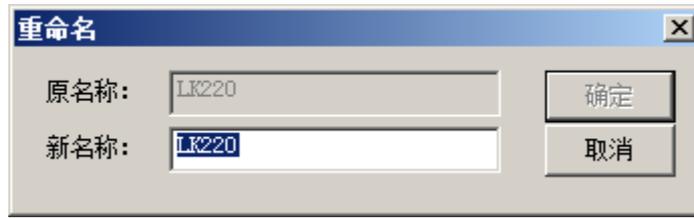


图 5.6-71 重命名控制器

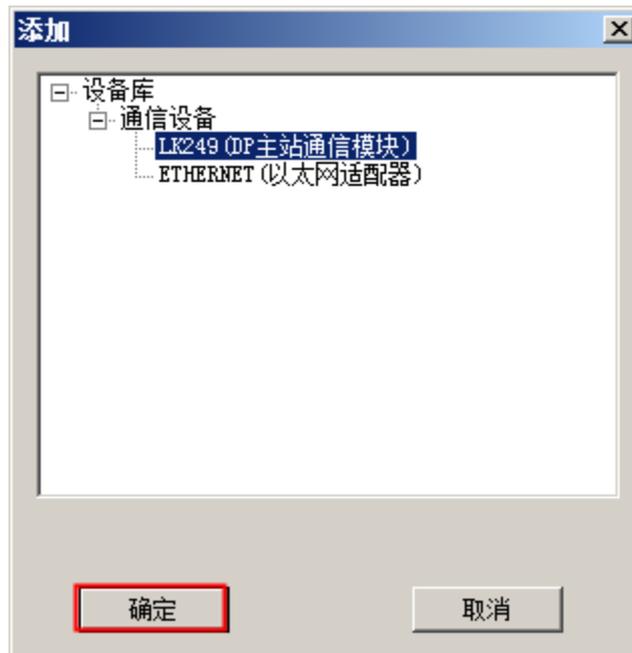
5.6.4.2 添加设备

通过【添加设备】命令，进行通信设备的添加，如图 5.6-72 所示。目前支持的通信设备包括 DP 主站通信模块和以太网适配器。以太网适配器新建工程时默认添加，用户不需要添加。



图 5.6-72 选择添加设备命令

弹出“添加”对话框，图 5.6-73 (a) 所示。选择 LK249 模块进行 IO 设备组态。选择通信设备，单击**确定**，在 CPU 模块下生成所选的通信模块。如图 5.6-73 (b) 所示。



(a)



(b)

图 5.6-73 通信设备的添加



- LK220 CPU 模块目前最多支持 1 个 DP 主站通信模块和 1 个以太网适配器。

通过通信模块的右键菜单，可以查看模块的设备信息、进行重命名、协议的添加和模块删除操作。如图 5.6-74 所示。

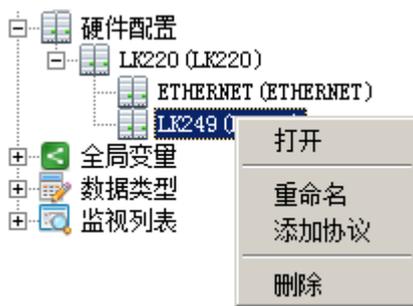


图 5.6-74 LK249 右键菜单

1. 查看设备信息

选择【打开】命令，或双击模块，在右侧区域打开该模块的设备信息窗口，同时自动加载“设备库”窗口。如图 5.6-75 所示。



图 5.6-75 设备信息窗口

在设备信息窗口中，显示该模块的基本设备信息和模块图片。

2. 重命名

参见章节 5.6.4.1 控制器模块的重命名操作。

3. 删除

选择【删除】命令，弹出删除确认提示框，如图 5.6-76 所示，单击**是**，删除所选模块。



图 5.6-76 删除确认提示框

5.6.4.3 添加协议

LK220 模块支持 PROFIBUS—DP 协议和 MODBUS TCP 协议，用户可在相应的通信设备下进行协议的添加。如图 5.6-77 所示，在 LK249 模块的右键菜单中选择【添加协议】命令，进行 DP 协议的添加。MODBUS TCP 协议的添加请详见 5.6.4.6 配置 MODBUS TCP 协议。

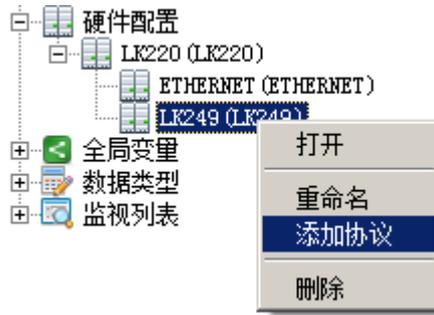


图 5.6-77 选择添加协议命令

弹出“添加”对话框，如图 5.6-78 所示。选择所需的协议，单击**确定**，完成协议添加。后续介绍以 DP_MASTER 协议为例进行阐述。



(a)



(b)

图 5.6-78 添加 DP 协议过程



- 一个 DP 主站通信设备下最多支持一个 DP MASTER 协议。

通过 DP_MASTER 节点的右键菜单，可以查看模块的设备信息、重命名、添加设备以及模块删除操作。如图 5.6-79 所示。



图 5.6-79 DP_MASTER 右键菜单

1. 查看设备信息

选择【打开】命令，或双击 DP_MASTER 节点，在右侧区域打开该模块的设备信息窗口，同时自动加载“设备库”窗口。



图 5.6-80 DP_MASTER 信息窗口

在 DP_MASTER 模块的设备信息窗口中，显示 PROFIBUS—DP 通讯的参数信息。可以设置端口和波特率。端口选择为 **COM** 口。设置波特率时，选择不同的波特率，其它总线参数自动切换为最佳值。

2. 重命名

可以修改 DP_MASTER 协议的名称。详细操作参见章节 5.6.4.1 控制器模块的重命名操作。

3. 删除

使用删除命令，将协议删除。详细内容参见章节 5.6.4.2 添加设备的相关内容。

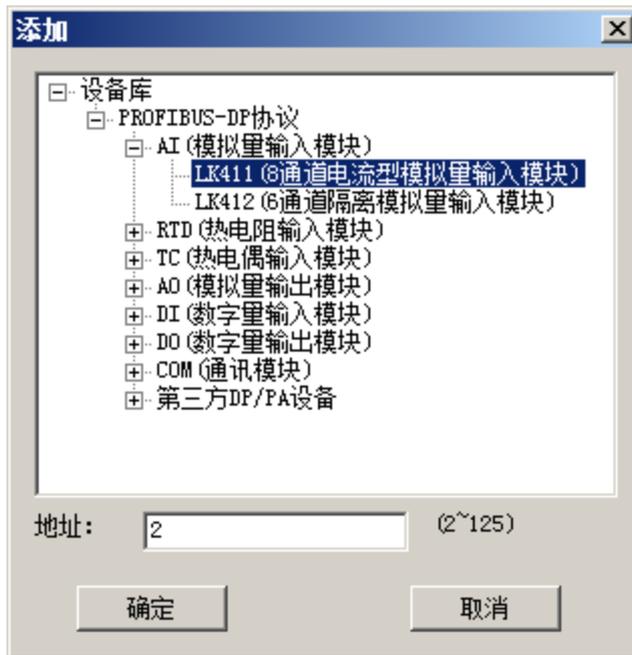
5.6.4.4 添加从站设备

选择【添加设备】命令，进行从站设备的添加，如图 5.6-81 所示操作。

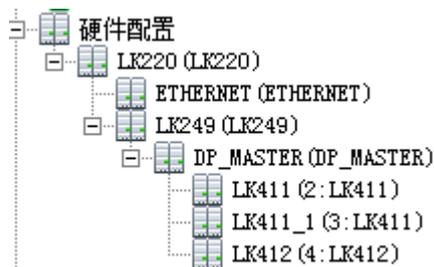


图 5.6-81 选择添加设备命令

弹出“添加”对话框，如图 5.6-82 所示。选择模块类型，在地址框中显示默认的从站地址，可进行修改。单击确定，在 DP_MASTER 节点生成新添加的模块。



(a)



(b)

图 5.6-82 添加从站设备

添加的从站模块的命名组成：从站名称（从站号：模块名称），对于同一模块重复添加时，会在从站名称的后面显示添加的序号，如图 5.6-82 所示的 LK411_1（3：LK411），1 模块添加的序号，3 为从站号。用户可通过右键菜单的【重命名】项修改从站名。

组态 DP 从站规模时，整个 DP 轮询周期不应超过 150ms。参见表 5.6-3 可大致估算出当前组态规模下的 DP 轮询周期（注：下表为单个从站的轮询周期的参考值）。

DP 轮询周期指轮询完所有的从站的总时间。

表 5.6-3 单个从站的轮询周期

波特率	LK239 模块用时 (ms)	非 LK239 模块用时 (ms)
187.5Kbps	29	2
500kbps	11	1
1500kbps	4	0.3

1. 添加第三方设备

在添加第三方设备之前，首先要将第三方设备的 GSD 文件导入到工程中。

(1) 导入 GSD 文件



- 菜单栏：单击【工程】—【添加配置文件...】。

参见章节 5.6.1.1 添加配置文件。

(2) 添加 DP/PA 设备

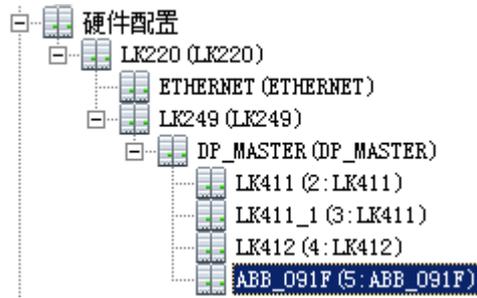
在“添加”对话框中单击**第三方 DP/PA 设备**前的, 在 DP 设备和 PA 设备节点下显示导入的设备，如图 5.6-83 (a) 所示。选择设备，单击**确定**，在 DP_MASTER 节点下生成添加的 DP 设备，如图 5.6-83 (b)、(c) 所示。



(a)



(b)



(c)

图 5.6-83 添加 DP/PA 设备

5.6.4.5 模块的操作

右击添加好的从站模块，弹出右键菜单，如图 5.6-84 所示。通过右键菜单命令，可以打开模块的设备信息、重命名、进行模块删除操作、查看设备属性。（以通讯模块 LK239-MASTER 为例进行介绍）。

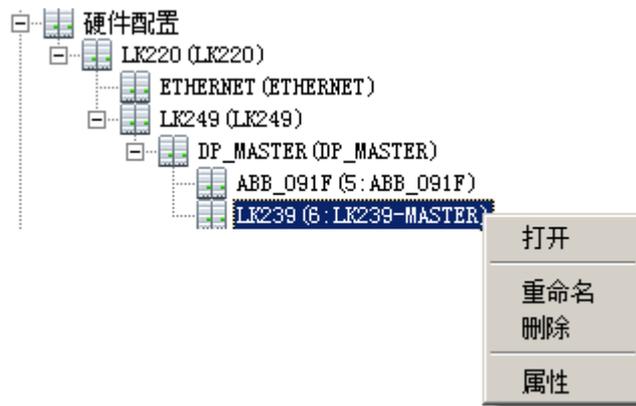


图 5.6-84 从站设备右键菜单

1. 打开

选择【打开】命令，或双击模块名称，在右侧区域打开模块的设备信息窗口，同时加载设备库。如图 5.6-85 所示。



图 5.6-85 模块设备信息窗口

(1) 修改从站地址

双击**设备地址**项的蓝色区域，弹出“修改地址”对话框，如图 5.6-86 所示。



图 5.6-86 修改从站地址

输入新地址，地址范围 2~125，单击**确定**，修改完成。



- 输入的地址不合法或地址重复时，**确定**按钮不可用。

(2) 设备属性

在【设备信息】列表中，双击**设备属性**项的**双击配置**（图 5.6-85 中蓝色区域），打开“设备属性”对话框，如图 5.6-88 所示。

该对话框包含【输入/输出选择】标签页和【用户参数】两部分内容。

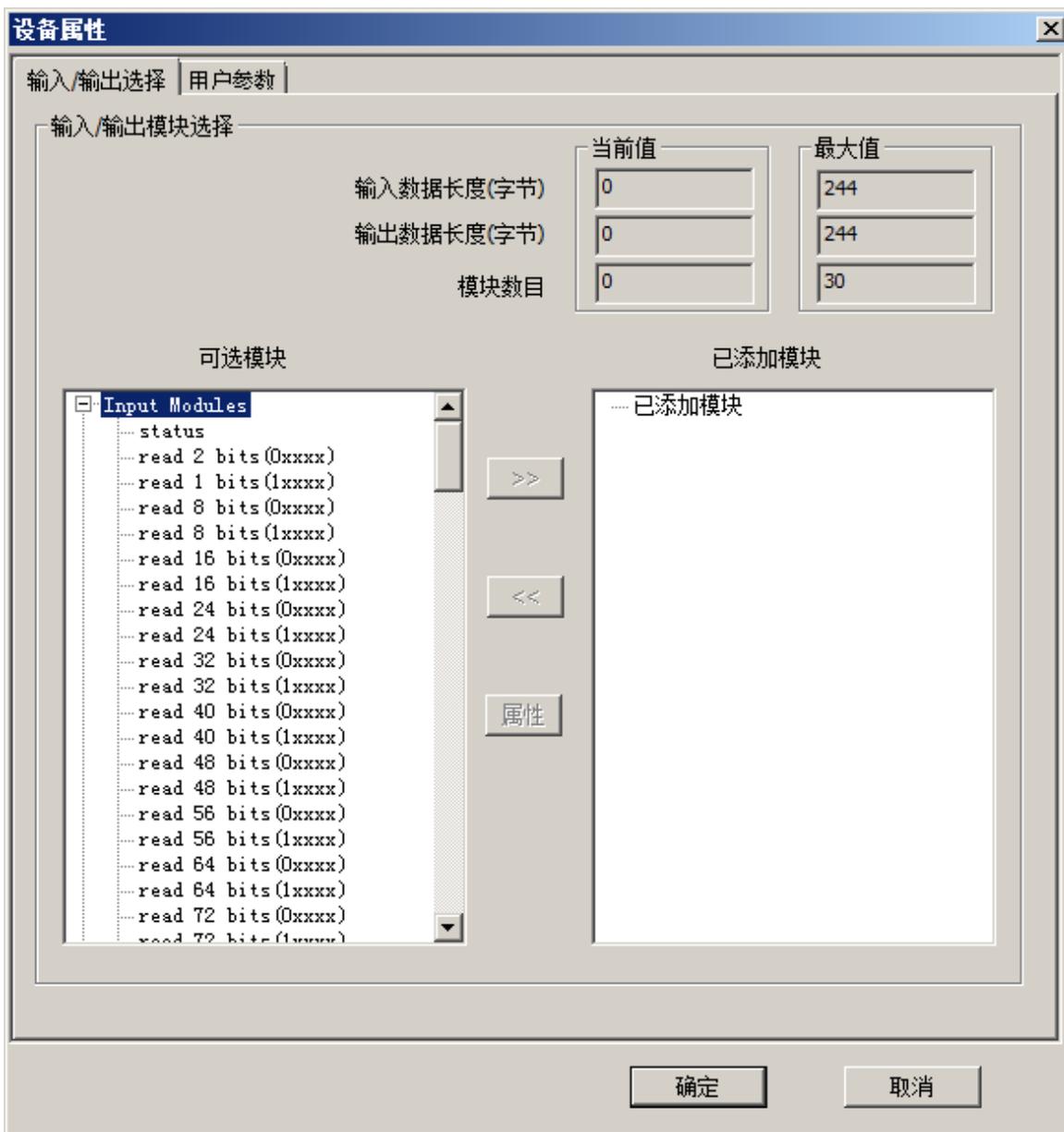


图 5.6-87 输入/输出选择标签页

【输入/输出选择】标签页进行子模块的添加以及子模块的参数设置。输入数据长度、输出数据长度、模块数目分别是当前已添加子模块的输入数据字节数、输出数据字节数、模块的个数。当前值不能超过最大值，否则提示操作失效。



- 只有通讯模块和第三方设备，可以进行在【输入/输出选择】页面进行子模块的添加和删除操作。自产模块只能查看。



图 5.6-88 用户参数标签页

【用户参数】标签页显示用户参数信息，用户可对参数进行配置。

(3) 子模块的添加

在【可选模块】列表框中选中模块，单击按钮 ，选中的模块被添加到【已添加模块】的节点下，如图 5.6-89 所示。

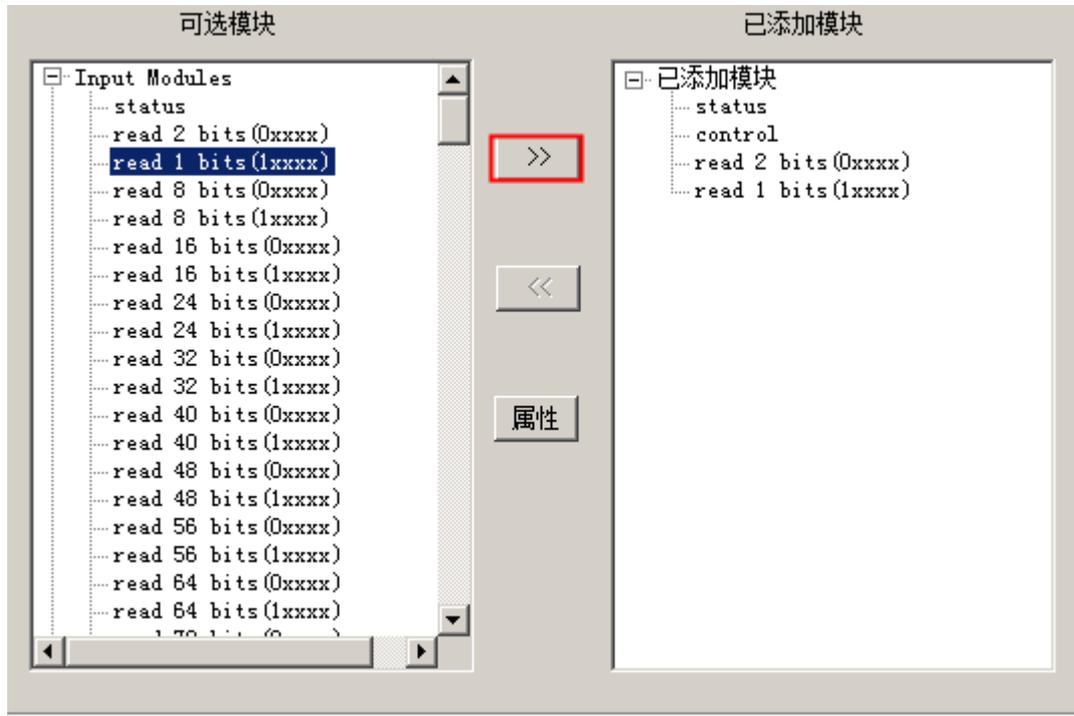


图 5.6-89 添加模块

选中【已添加模块】节点下的模块，单击按钮 ，模块从该节点下删除。

选中【已添加模块】节点下的模块，单击**属性**，弹出“子模块属性”对话框，如图 5.6-90 所示。可修改模块的**参数值**。

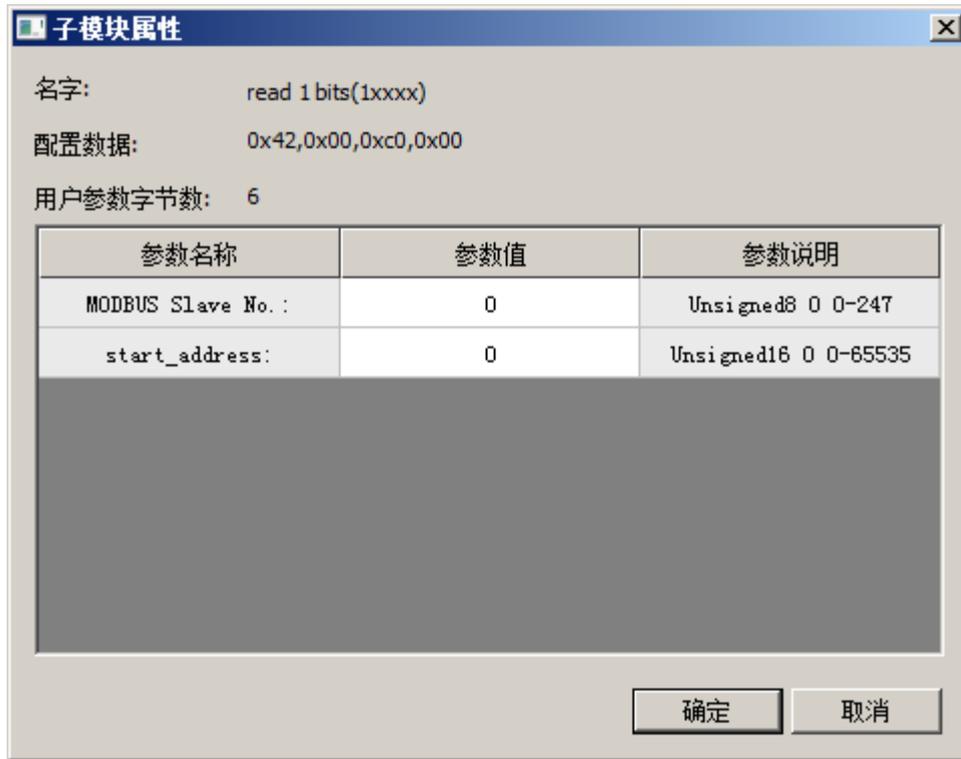


图 5.6-90 子模块属性对话框

设置好参数值后，单击**确定**按钮，关闭“子模块属性”对话框，同时在“设备属性”对话框中单击**确定**按钮，完成子模块的添加。

(4) 通道信息

子模块添加完成后，在“设备信息”窗口中以标签页的形式显示已添加子模块的通道信息，如图 5.6-91 所示。

单击子模块名称，显示该模块的所有通道信息。在线时会显示通道的**在线值**。

通道名称组成：**DPIO__串行通信模块序号_协议序号_从站地址_通道号**。串行通信模块序号缺省为 2，协议序号缺省为 1。

通道号	通道名称	通道类型	通道地址	通道说明
1	DPIO_2_1_6_1	BYTE	%I60	

图 5.6-91 子模块通道信息

2. 重命名

可以修改 IO 从站的名称。参见章节 5.6.4.1 控制器模块的重命名操作。

3. 删除

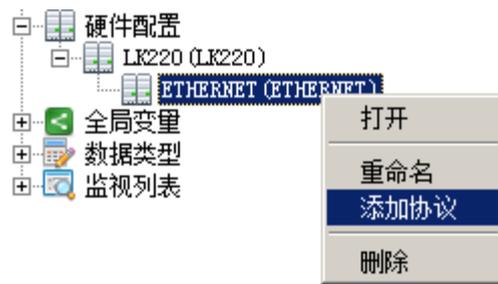
进行模块删除操作。详细内容参见章节 5.6.4.2 添加设备的相关内容。

4. 属性

单击模块右键菜单中的【属性】命令，打开“设备属性”对话框，详细内容参见本小节的【打开】命令设备属性内容。

5.6.4.6 配置 MODBUS TCP 协议

LK220 模块支持 MODBUS TCP 协议，在【ETHERNET】节点的右键菜单中选择【添加协议】命令，弹出“添加”对话框，如图 5.6-92 (b) 所示。共有 MODBUS TCP 主站和 MODBUS TCP 从站两种协议类型。当前 CPU 模块作主站时，选择 MODBUSTCP_MASTER 主站协议；当前 CPU 模块作从站时，选择 MODBUSTCP_SLAVE 从站协议。选择协议类型后，单击**确定**，添加协议完成，如图 5.6-92 (c) 所示。



(a)



(b)



(c)

图 5.6-92 添加 MODBUS TCP 协议

1. MODBUS TCP 主站协议配置

通过双击 MODBUSTCP_MASTER 节点或在右键菜单中选择【打开】命令，进入设备信息窗口，如图 5.6-93 所示。

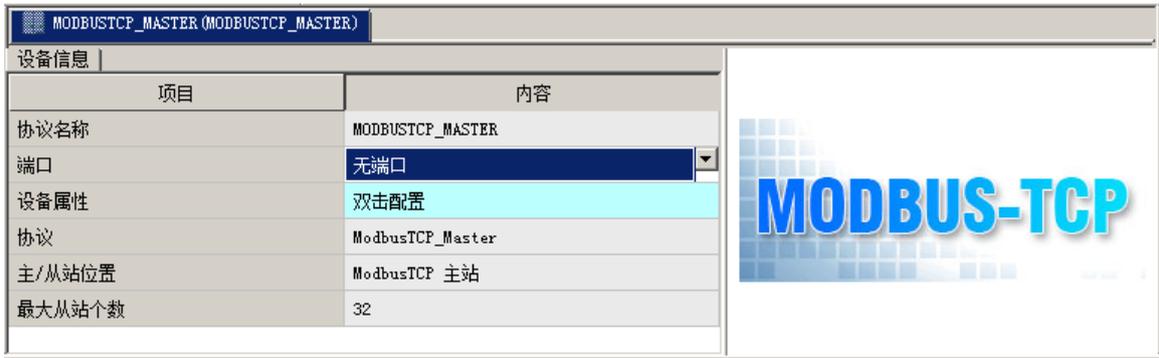


图 5.6-93 MODBUS TCP 主站“设备信息”窗口

在端口项中配置 MODBUS TCP 主站的端口，选择以太网口 1。

通过双击配置打开“设备属性”窗口，如图 5.6-94 所示。

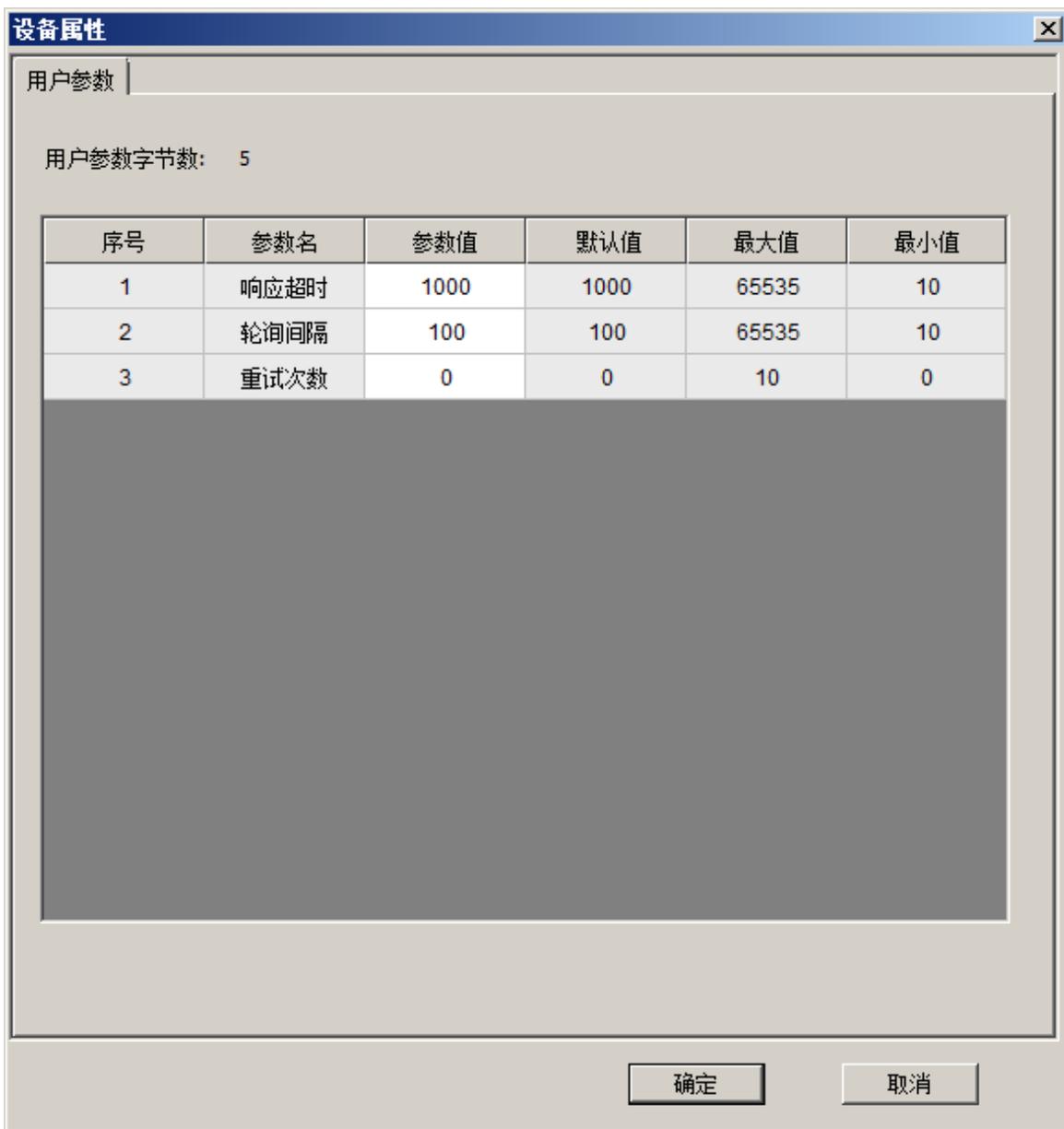


图 5.6-94 MODBUS TCP 主站“设备属性”对话框

设备属性对话框中显示可配置的 MODBUS TCP 主站协议参数，用户可修改参数值。

- 响应超时：MODBUS TCP 主站发送请求帧后所允许的从站延时应答时间。默认值 1000。
- 轮询间隔：MODBUS TCP 主站接收到从站应答帧后到发送下一个请求帧的间隔时间，如果上一帧从站应答超时，则主站可忽略该间隔时间直接发送请求帧。
- 重试次数：从站应答异常后主站重新发送请求的次数。

2. MODBUS 主站协议配置从站

当前 CPU 为主站时，可以配置一个或多个从站进行数据通信。在 MODBUSTCP_MASTER 节点的右键菜单中选择【添加设备】命令，如图 5.6-95 所示。

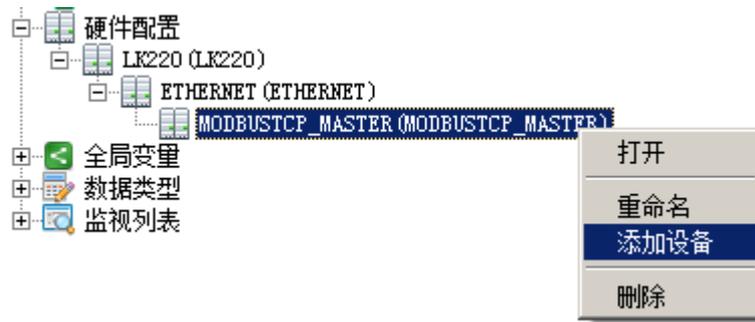
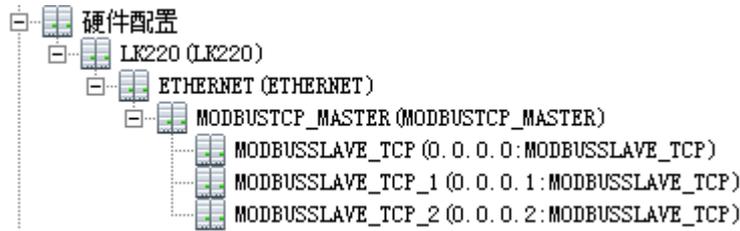


图 5.6-95 选择添加设备

弹出“添加”对话框，如图 5.6-96 (a) 所示，选择从站设备，在地址框中显示从站默认的 IP 地址 0.0.0.0，可进行修改，输入地址后，单击确定，MODBUSTCP 从站被添加到主站节点下，如图 5.6-96 (b) 所示。



(a)



(b)

图 5.6-96 MODBUSTCP 主站配置从站

MODBUS TCP 主站下可添加 32 个从站，添加的从站缺省显示：从站名（从站地址：设备名称），用户可以对从站地址进行修改。

(1) 修改从站地址

双击从站，打开“设备信息”窗口，如图 5.6-97 (a) 所示。双击设备地址栏，弹出“修改地址”对话框，如图 5.6-97 (b) 所示，输入新地址，单击**确定**，修改地址完成。

MODBUSSLAVE_TCP (0.0.0.0:MODBUSSLAVE_TCP)	
设备信息	
项目	内容
模块型号	MODBUSSLAVE_TCP
设备地址	0.0.0.0 (双击配置)
输入起始地址	未配置
输出起始地址	未配置
设备属性	双击配置
最大指令个数	32
主/从站位置	ModbusTCP 从站

(a)

修改地址 ✕

原地址

新地址

(b)

图 5.6-97 修改从站地址

(2) MODBUS 从站设备指令配置

如图 5.6-97 (a) 所示, 在**设备属性**栏通过**双击配置**, 打开“设备属性”对话框, 如图 5.6-98 所示。对话框包含【配置指令】和【用户参数】两个标签页。在【配置指令】标签页中进行指令的配置以及指令参数的设置。在【用户参数】标签页中显示从站的参数信息, 用户可对参数值进行修改。

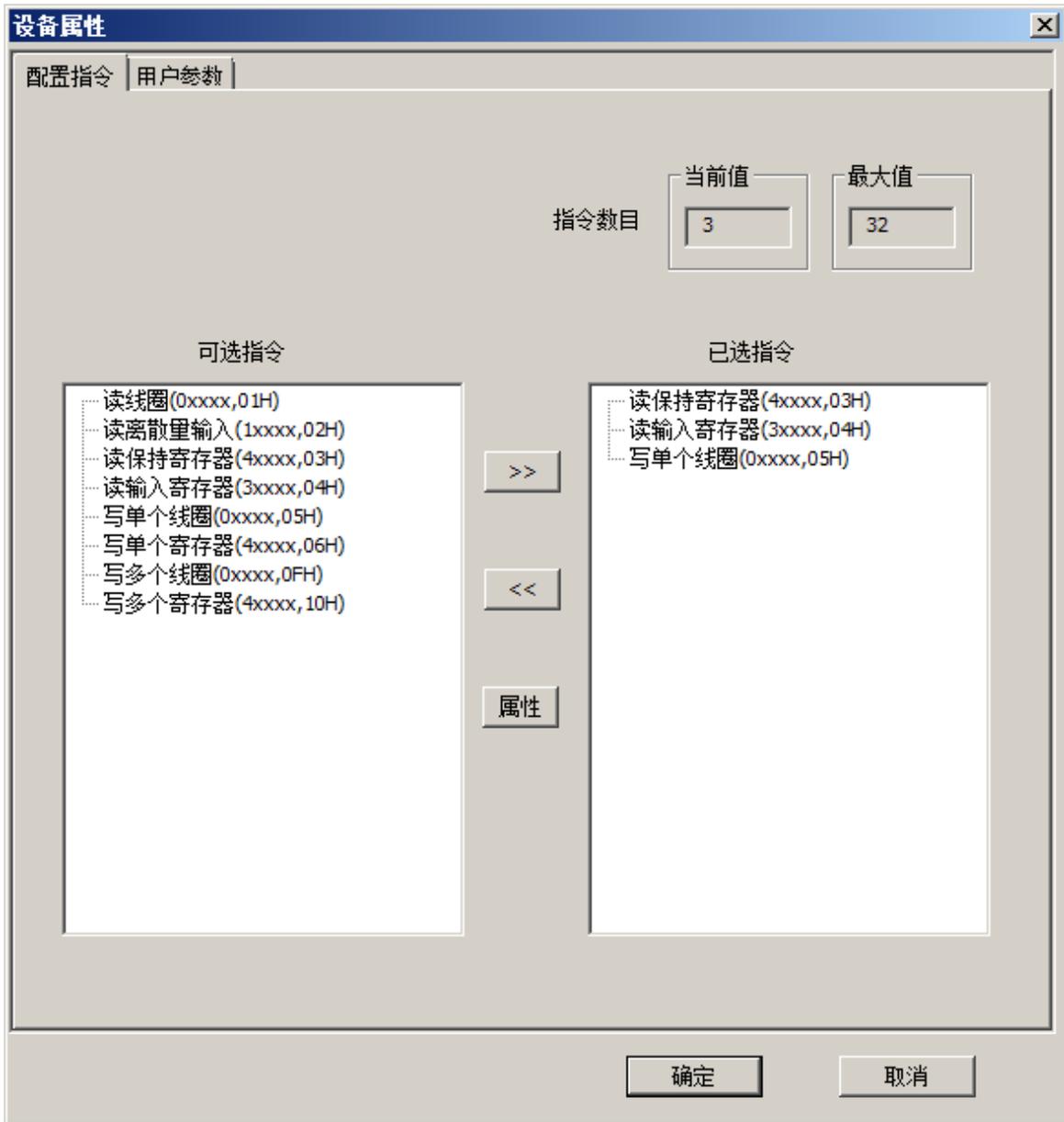


图 5.6-98 MODBUS TCP 从站设备属性

可选指令列表框中显示可以选择的指令集, 选择指令后, 单击 **>>** 按钮, 将选中的指令添加到**已选指令**列表框中。同时, 每增加一条指令, **指令数目**当前值增加 1, 添加的指令数目不能超过**最大值**。

通过 **<<** 按钮，可以将已选指令列表框中选中的指令进行删除。选中**已选指令**列表框中的指令，单击**属性**，打开“指令参数”对话框，如图 5.6-99 所示。可对白色区域的**参数值**项进行修改。

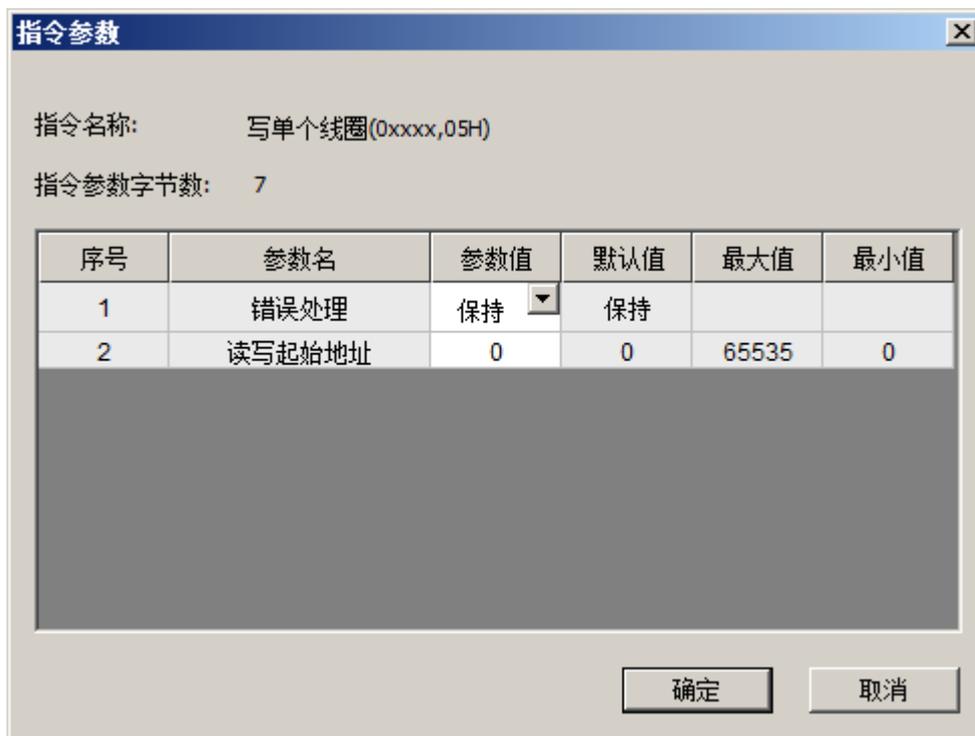


图 5.6-99 MODBUS TCP 从站指令参数设置

当修改“指令参数”对话框中的读/写模拟量、开关量数据长度时，会弹出修改确认的提示框。

单击【用户参数】标签页，显示用户可组态的参数信息，可对**参数值**进行修改。



图 5.6-100 MODBUS TCP 从站用户参数设置

- 单元 ID: MODBUS TCP 协议单元 ID。
- 端口号: MODBUS TCP 协议端口号。
- 响应超时: MODBUS TCP 主站发送请求帧后所允许的从站延时应答时间。

指令配置好后, 单击**确定**, 在 MODBUS_SLAVE_TCP 从站设备信息窗口下面自动更新所配置的指令信息, 用户可以修改**通道名称**、**通道说明**, 通道名称初始值为 MODBUS_TCPIO_设备序号_协议序号_四位从站 IP 地址_通道号。



图 5.6-101 MODBUS TCP 从站指令添加完成

3. MODBUS TCP 从站协议配置

LK220 作从站时,需要添加 MODBUSTCP_SLAVE 协议,添加方法参见 5.6.4.6 配置 MODBUS TCP 协议。

(1) 配置端口

双击 MODBUSTCP_SLAVE 从站节点,打开“设备信息”窗口,如图 5.6-102 所示。



图 5.6-102 MODBUS TCP 从站设备信息窗口

在端口项中配置 MODBUSTCP 从站的端口,选择以太网口 1。

(2) MODBUSTCP_SLAVE 从站参数设置

双击图 5.6-102 中的设备属性项,打开“设备属性”对话框,如图 5.6-103 所示。



图 5.6-103 MODBUS TCP 从站“设备属性”对话框

在“设备属性”对话框中显示可配置的 MODBUS TCP 从站协议参数，用户可修改参数值。

- 单元 ID: MODBUS TCP 协议单元 ID。
- 端口号: MODBUS TCP 协议端口号。
- 超时时间: 从站没有接收到主站发送数据的时间间隔，超过设定的时间，则从站断开与主站的通讯链接。

5.6.5 平台切换

通过平台切换，可以把工程从一个平台转换到另一个平台，用户无需重新组态，就能在新平台中使用之前编写的程序。

5.6.5.1 平台切换的触发

平台切换是通过更换 CPU 模块触发的。在已添加 CPU 模块的工程中，选中 CPU 模块后，选择右键菜单【模块替换】—【CPU 模块】，选择要替换的 CPU 类型，如果选择的 CPU 模块与当前工程的目标平台不对应，则触发平台切换。这里以 14 点平台切换为 24 点为例进行介绍。目标配置版本与支持的 CPU 型号的列表请参见表 5.2-2。

打开“硬件配置”视图，拖动一个 24 点平台的 CPU 模块 LE5106（或 LE5107）到“硬件配置”窗口中。由于添加的 CPU 模块和当前使用的 14 点的目标平台不同，平台切换立即触发。根据目标配置的不同，平台切换触发后有以下两种表现形式：

在 V3.1.1B1 的目标配置下，平台切换触发后弹出提示框，如图 5.6-104 所示，提示用户将更新工程数据。

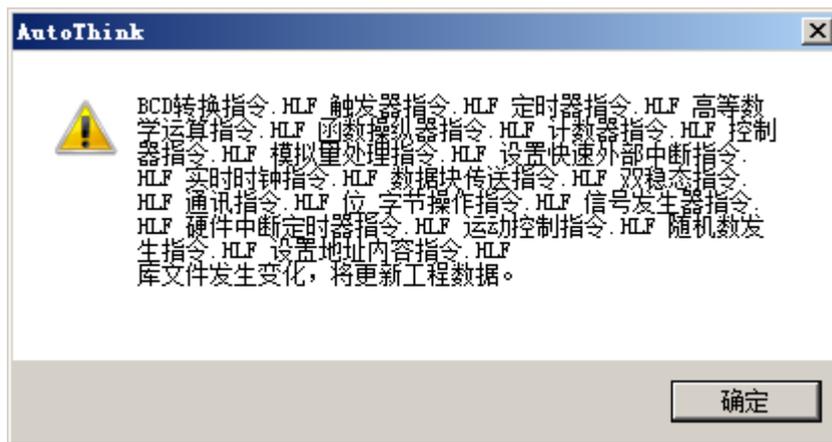


图 5.6-104 工程数据更新提示窗口

单击**确定**，可以看到 LE5106 添加成功，平台切换到了 24 点，如图 5.6-105 所示。

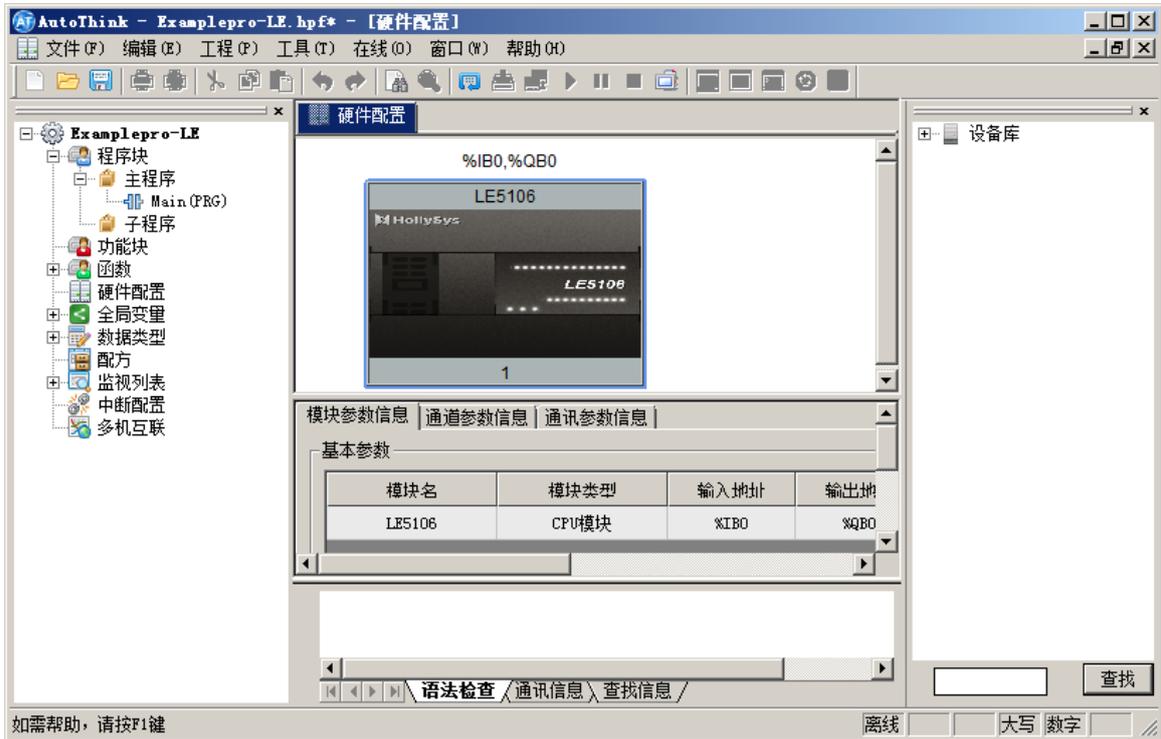


图 5.6-105 平台切换完成

在最新的目标配置上，平台切换触发后，不弹出工程数据更新提示窗口，直接将 14 点的目标配置切换为 24 点。



- 在最新的目标配置上平台切换，不弹出工程数据更新提示窗口，是因为最新的目标配置上各个平台的库文件合一。

5.6.5.2 平台切换后工程的变化

平台切换后工程会有相应变化，下面是一些需要注意的问题。

1. 变量的可强制属性

变量的可强制属性有两种：可强制和不可强制。一个变量是可强制的，是指在程序运行时变量的值可以被强制为一个固定的值，保持不变。如果一个变量不可强制，那么在程序运行时变量的值无法被强制不变。这就是变量的可强制属性。

变量的可强制属性由平台决定，用户无法进行修改。

变量的强制详见章节 [5.7.4.1 在线的强制](#)。

平台切换后，变量的可强制属性由切换后的平台置确定：平台从 14 点切换到 24 或 40 点后，变量由不可强制变成可强制；从 24 点或 40 点切换到 14 点后，变量由可强制变成不可强制；24 点和 40 点之间切换，变量的可强制属性不变。

2. 平台切换后的其它变化

- 库文件更新：在较早的目标配置下，更换库文件为切换后平台的库文件版本；最新的目标配置下无变化。
- 直接地址变量或配方可能越界：由于涉及到直接地址分配内存，所以平台切换后内存区域分配上的变化可能会导致越界。
- 工程状态的改变：平台切换后相当于一个新工程的建立，需要重新编译、下装等。
- 硬件配置模块的变化：14 点平台支持的硬件配置模块，与其它平台的不同，所以切换后，工程中的硬件模块随着平台切换改变。例如 14 点的硬件配置是没有功能扩展板，而 24 点和 40 点的是支持功能扩展板的。

5.6.6 设备库

MC、LE 工程中打开【硬件配置】编辑界面时自动加载“设备库”窗口。LK 工程打开【硬件配置】节点下模块的“设备信息”窗口时，自动加载“设备库”窗口。手动关闭“设备库”窗口后，需要手动加载设备库。

“设备库”窗口与“硬件配置”窗口之间的记忆关联关系参见章节第 6 章 库管理。



- 菜单栏：单击【窗口】—【设备库窗口】。



图 5.6-106 手动加载设备库窗口

在该窗口中显示了当前工程可以使用的各种类型的设备。主要分为 CPU、采集不同类型信号的 IO 模块、其它模块等类型。展开各个子节点可以看到具体的模块分支，鼠标移动到模块名称上显示该模块的模块提示框。

在该窗口底端还提供查找功能框，可以对设定类型的模块进行快速查找定位配置。

在**查找**条件输入框中输入要查找的条件，搜索条件不区分字母大小写，单击**查找**进行检索，搜索完成后，会在搜索结果列表中显示符合搜索条件的模块类型，双击搜索结果，将定位到设备列表中并将模块标识为亮蓝色以供使用，如图 5.6-107 所示。



图 5.6-107 设备库窗口的查找功能

5.7 编译下装

5.7.1 编译



- 菜单栏：单击【工程】—【编译】；
- 工具栏：；
- 快捷键：**F11**。

程序编写完成后，通过编译来验证程序的正确性和可执行性，并在“信息”窗口中显示编译结果。MC、LK 工程中编译分为全编译和增量编译两种方式。

默认情况下单击【编译】将执行增量编译操作，当增量编译条件不满足时，单击【编译】将执行全编译操作。编译过程中在状态栏显示编译进度。



- LE 工程只支持全编译方式，执行【编译】命令时，将进行全编译操作，且无需进行全编译确认。
- 使用 V3.1.3B1 之前版本建的工程，在升级之后需先手动清除工程文件下 comm_log 文件夹，再编译、下装、监视运行操作。

5.7.1.1 全编译



- 菜单栏：单击【工程】—【全编译】。

全编译时硬件配置信息重新生成，所有变量重新分配，POU 重新翻译等，生成新的组态逻辑信息和硬件配置信息。以下条件将触发全编译操作：

- 新建工程后首次执行【编译】操作。
- 编译时，.iec 或.tmp 文件不完整或者损坏。
- 切换目标配置，执行【编译】操作。
- 切换 CPU 型号，执行【编译】操作。
- 打开旧版本工程文件引起 TrgVersion 文件升级。
- 下装控制器时，.iec 或者.tmp 文件不完整。

- 直接执行【全编译】命令。
- 硬件配置改变，执行【编译】操作。

除新建工程首次执行【编译】操作外，其他情况下触发全编译操作时，均需要进行全编译确认。

直接执行【全编译】命令时，弹出全编译确认窗口如图 5.7-1 所示。

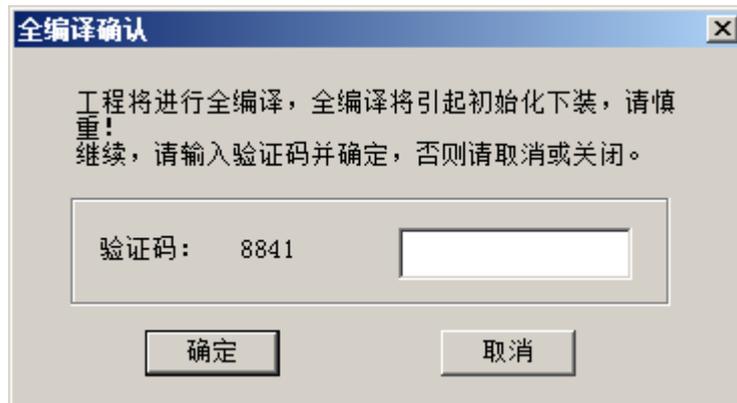


图 5.7-1 全编译确认

其他条件下触发全编译操作时，弹出全编译确认窗口，如图 5.7-2 所示。

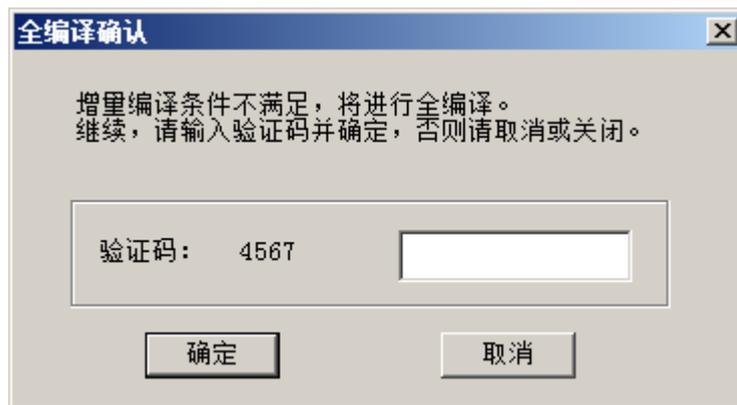


图 5.7-2 全编译确认

在弹出的窗口中输入正确的验证码之后，单击**确定**，执行全编译操作。

反复修改变量时删除的 G 区和 R 区变量不释放，全编译后可减少变量占用的内存数量。



- 全编译将会引起 AutoThink 初始化下装，请慎重操作！

5.7.1.2 增量编译

增量编译只对修改和追加的内容进行编译。

以下任一情况发生都进行增量编译操作：

- POU 的修改：增加或删除 POU、修改已有 POU 中的逻辑。
- 变量的修改：增加或删除变量定义、修改已有变量定义。
- 任务属性修改：增加调用 POU、任务属性值改变。

编译成功后，相关的文件参见表 5.7-1。

表 5.7-1 工程的文件类型表

文件扩展名	例子	描述	格式
.hpf	Project01.hpf	控制站算法工程文件	二进制文件
.tmp	Project01.tmp	编译和增量下装信息文件	二进制文件
.iec	Project01.iec	IEC 运算逻辑文件	二进制文件
.at	Project01.at	编译生成的逻辑组态信息文件	二进制文件
.sim	Project01.sim	仿真时编译生成的逻辑组态信息文件	二进制文件

5.7.1.3 编译结果

编译完成后，会在 AutoThink 下方信息窗口显示编译结果，有以下两种可能：

一是编译全部正确，如图 5.7-3(b)所示，这表明工程全部正确。

二是编译有错误或警告，如图 5.7-3(a)所示，这表明工程有部分错误，并用红色字显示错误信息。双击编译错误可以直接定位到该处错误，进行修改。

```

开始编译...
正在编译: 硬件配置...
正在编译: 引用库...
正在编译: POU的错误检查...
正在编译: Main检查...
E2501: LD程序组织单元 Main: 0002节 关联数据:标识符'TGG'未定义
正在编译: 类型收集...
正在编译: 类型表...
正在编译: 收集工程变量...
正在编译: 收集通道变量...
正在编译: 收集库中变量...
正在编译: 收集配方中变量...
正在编译: 调用树...
正在编译: 偏移分配...
正在编译: 工程POU翻译...
正在编译: 库文件POU翻译...
库翻译结束...
正在编译: 生成iec内存文件...
正在编译: 生成任务描述信息...
正在编译: 生成tmp内存文件...

ADD_COMPILE
已用时间 00:00:00.500
编译完成: Untitled - 1 错误, 0 警告

```

(a)

```

正在编译: 调用树...
正在编译: 偏移分配...
正在编译: 工程POU翻译...
正在编译: Main翻译...
正在编译: 库文件POU翻译...
库翻译结束...
正在编译: 生成iec内存文件...
正在编译: 生成任务描述信息...
正在编译: 生成tmp内存文件...
代码区: 共1048576字节, 使用64092 字节 (6.11 %)
N区: 共1048576 字节, 使用20074字节 (1.91 %)
R区: 共4048 字节, 使用2字节 (0.05 %)
数据区占用范围: M区-未占用, I区-0~12字节, Q区-0~12字节, N区-0~20076字节, R区-0~2字节

ADD_COMPILE
已用时间 00:00:00.562
编译完成: Untitled - 0 错误, 0 警告

```

(b)

图 5.7-3 编译结果对照

5.7.2 通讯设置

下装前，需要建立工程师站和主控单元间的通讯参数。每个控制站算法工程针对一个现场控制站，必须为算法工程设置正确畅通的通讯端口参数。MC、LK 工程仅支持 TCP/IP 通讯，LE 除 TCP/IP 通讯外还支持 PC 串口通讯。

设置本地计算机与控制器 CPU 模块之间的通讯信息，以 LE 工程为例进行介绍。



- 菜单栏：单击【在线】—【通讯设置】。

如使用 TCP/IP 进行通讯，选择【TCP/IP】标签页，设置当前下装控制器的 IP 地址，心跳时间、重连时间和端口号均采用默认值，不允许修改，最后单击确定按钮，完成通讯设置，如图 5.7-4 所示。



图 5.7-4 通讯设置—“TCP/IP”

设置完成后，还需设置计算机网卡的本地连接 TCP/IP 地址，需保证该地址与控制器地址在同一网段内，如控制器地址为 192.168.1.40，则计算机本地连接 TCP/IPv4 地址为 192.168.1.1，如图 5.7-5 所示。

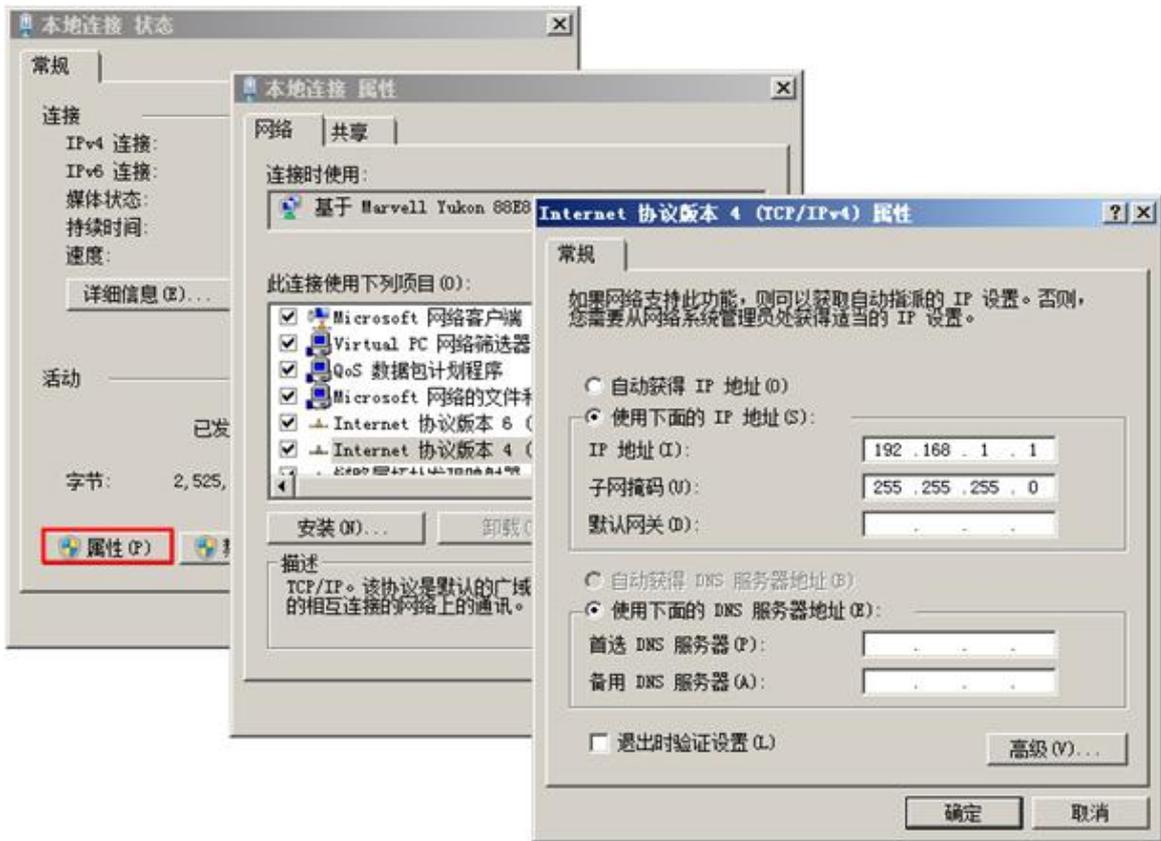


图 5.7-5 计算机 IP 地址设置

如使用 PC 串口通讯，选择【PC 串口】页面，设置自动搜索串口或者手动选择串口，如图 5.7-6 所示。

当选中自动搜索串口方式时，系统自动查找可用的通讯端口；选中手动选择串口，在选择串口下拉列表中选择下装的端口号，单击确定，完成通讯设置。



图 5.7-6 通讯设置—“PC 串口”



- 下装前，请设置正确的待下装控制器 IP 地址或可用的下装端口。
- MC、LK 工程不支持 PC 串口通讯。

5.7.3 下装

程序编译成功后，就可以下装到主控单元了。而且根据现场实际情况的不同，触发不同的下装方式。

AutoThink 中下装分为全下装和增量下装两种方式。

程序编译通过后，选择该命令，将工程相关文件下装到 CPU 模块中。



- 菜单栏：单击【在线】—【下装】；
- 工具栏：。



- AutoThink 软件中对于 LE 工程仅支持全下装方式，对于 MC、LK 工程支持全下装和增量下装两种方式。

5.7.3.1 全下装

即初始化下装，下装全部组态逻辑信息和硬件配置信息到控制器，下装时，对应控制器停止运算、输出，当出现以下情况（中的一种）时，将进行初始化下装操作：

- 工程初次下装。
- 当前工程与控制器工程不一致或版本不连续。
- 执行全编译操作后。

单击【下装】命令，首先弹出初始化下装确认窗口，如图 5.7-7 所示。

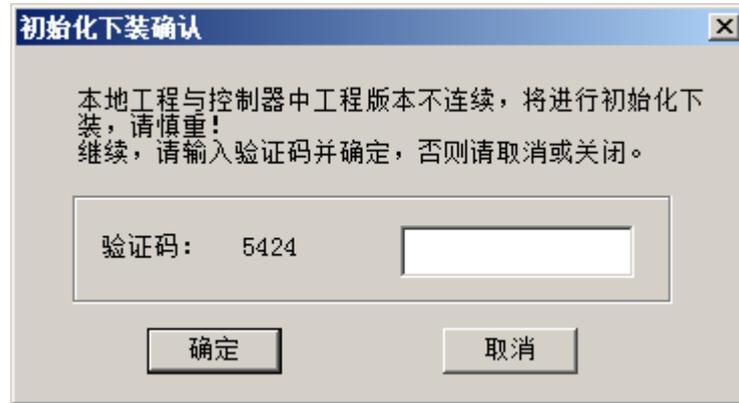


图 5.7-7 初始化下装确认



- LE 工程仅支持全下装方式，执行下装操作时，无需输入验证码，直接进行下装确认。

在窗口中输入正确的验证码后，单击**确定**，弹出下装提示，如图 5.7-8 和图 5.7-9 所示。



图 5.7-8 LE 工程下装提示信息



图 5.7-9 MC 工程下装提示信息

在提示窗口中选择**是**，开始下装程序。下装完成后，会在 AutoThink 的“通讯信息”窗口中显示下装结果。

5.7.3.2 增量下装

增量下装是指工程下装完成后，修改工程文件编译通过后，再次执行【下装】操作时触发的下装命令。增量下装时任务停止、算法运算停止、指令缓存清空、控制器中的变量地址和值不受影响、算法模块的轴参数信息不变、而新变量增量初始化等。

单击【下装】命令执行增量下装时，如果新增全局变量，或者 PRG 类型 POU 中新定义变量或者原有变量类型修改，则首先会弹出“待初始化新增变量”对话框提示新增变量，单击**确定**继续下装过程，单击**取消**取消下装，如图 5.7-10 所示。



图 5.7-10 待初始化新增变量窗口

待初始化新增变量确认完成后，弹出下装提示框，如图 5.7-11 所示，单击**是**继续下装过程，选择**否**取消下装。



图 5.7-11 MC 工程下装提示信息

选择**是**，开始下装程序，如图 5.7-12 所示。

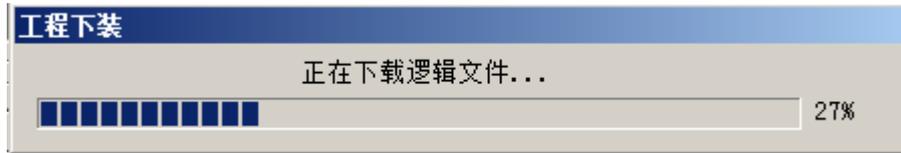


图 5.7-12 MC 工程增量下装

下装完成后，会在 AutoThink 的“通讯信息”窗口显示提示信息。

当下装前后的工程未发生变动时，单击【下装】命令，在下装过程中将弹出系统提示，如图 5.7-13 所示。



图 5.7-13 下装提示信息-重复下装

单击**确定**按钮后，系统退出下装，并在“通讯信息”窗口显示提示信息。

下装完成后，可进入在线调试状态，观察数据变化。

5.7.4 在线调试及操作

5.7.4.1 在线

1. 运行(R)



- 菜单栏：单击【在线】—【运行】；
- 工具栏：；
- 快捷键：**F5**。

监视状态下，使用该命令进入在线运行。在 MC、LK 工程中，如果任务属性的**启动类型**选择为**自动**，编译下装后进入监视状态，任务自动运行，无需单击**运行**。

2. 停止(S)



- 菜单栏：单击【在线】—【停止】；
- 工具栏：；
- 快捷键：**Shift+F8**。

停止程序的运算状态，各个变量保持运算停止前的值，当再次运行时，在该值的基础上继续运算。

3. 复位(H)



- 菜单栏：单击【在线】—【复位】。

复位是模拟掉电再上电后的控制器状态，控制器需要将掉电保持区的变量恢复为掉电前的数值，而其他内存区的变量恢复为初始化值。当变量本身处于强制状态时，则复位后，将强制状态取消，对应的内存值根据存储区域的不同恢复为对应的数值。



- 在监视/仿真模式下有效。
- LK 工程，复位后变量强制状态保持。

4. 冷复位(T)



- 菜单栏：单击【在线】—【冷复位】。

冷复位是模拟工程初始化下装后的状态，当现场控制器下装程序后，需要把各内存区的数据恢复到初始值时启用冷复位功能。当变量处于被强制状态时，冷复位也起作用，将被强制的变量恢复至初始值，并将强制状态取消。



- 在监视/仿真模式下有效。

5. 热复位 (W)



- 菜单栏：单击【在线】—【热复位】。

热复位是模拟工程重启后各数据区保持重启前的状态。启用热复位功能后，重新加载用户工程逻辑，各内存区的数据保持不变。



- 仅 LK 平台支持此功能。

6. 清空控制器(C)

在离线模式下有效，清空 PLC 中的工程，重新初始化 PLC 系统。



菜单栏：单击【在线】—【清空控制器】。

在 MC、LK 工程中，此功能用于清空控制器指定目录，弹出对话框如图 5.7-14 所示。



图 5.7-14 清空控制器 (MC)



- MC、LK 工程执行清空控制器后，需要对控制器进行复位或重启。

在 LE 工程中执行【清空控制器】命令：

弹出对话框如图 5.7-15 所示，单击**确定**，清空控制器。

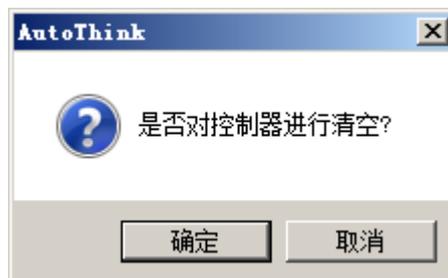


图 5.7-15 清空控制器

7. 写入存储卡



菜单栏：单击【在线】—【写入存储卡】。

使用该命令可以将当前程序文件写入到存储卡中。



- MC、LK 工程不支持此功能。

8. 清空存储卡



菜单栏：单击【在线】—【清空存储卡】。

使用该命令可以清空存储卡。



- MC、LK 工程不支持此功能。

9. 控制器校时



菜单栏：单击【在线】—【控制器校时】。

该功能在线状态下可用。在弹出的窗口中，上方显示的是控制器当前时间，下方为设置控制器新时间。新时间可手动输入，然后单击**设置**按钮生效。或者直接单击**同步 PC 时钟**按钮，将控制器时间设置为 PC 时间。控制器校时界面如图 5.7-16 所示。

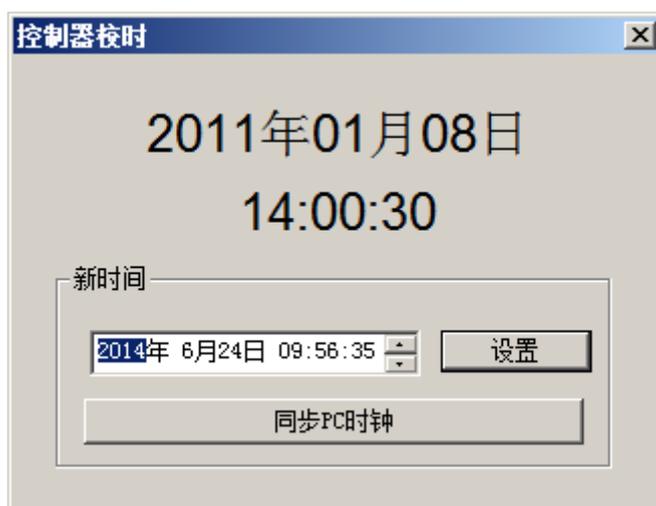


图 5.7-16 控制器校时



- LE 不支持此功能。

10. 设置控制器 IP



菜单栏：单击【在线】—【设置控制器 IP】。

该功能在线状态下可用，弹出的窗口如图 5.7-17 所示。设置控制器新的 IP 后，控制器在重启后生效。



图 5.7-17 设置控制器 IP



- LE、LK 不支持此功能。

11. 监视(O)



- 菜单栏：单击【在线】—【监视】；
- 工具栏：；
- 快捷键：**Alt+F8**。

把编程系统与监视 CPU 模块运行结合起来，使系统进入在线监视运行状态。再次单击则退出调试状态，切换到程序编辑状态。

在线监视时，双击功能块可查看内部运行逻辑。



- 在线监视时，系统功能块内部逻辑不可查看，工程或自定义库中增加的功能块可查看逻辑。
- 函数块在离线或者在线状态下，显示的内部算法均是离线状态。
- 当监视变量过多时，POU 或监视列表视图中部分监视变量**在线值**会显示为“???”，请拖动右侧滚动条使该变量处于变量定义区域的下部区域或者缩小变量定义可视区域，减少该可视区域中变量个数，使其**在线值**正常显示。

12. 仿真(U)



- 菜单栏：单击【在线】—【仿真】；
- 工具栏：。

通过该命令进入仿真模式，程序块运行在操作系统平台下的本地计算机的相应的仿真 CPU 模块内。此模式可用于检查工程。



- MC、LK 平台不支持此功能。

退出仿真模式：



- 菜单栏：单击【在线】—【退出仿真】；
- 工具栏：。

13. 写入



- 菜单栏：单击【在线】—【写入】；
- 快捷键：**Ctrl+F7**。

调试时修改变量值。双击处于运行状态的程序中的变量，弹出窗口如图 5.7-18 示。

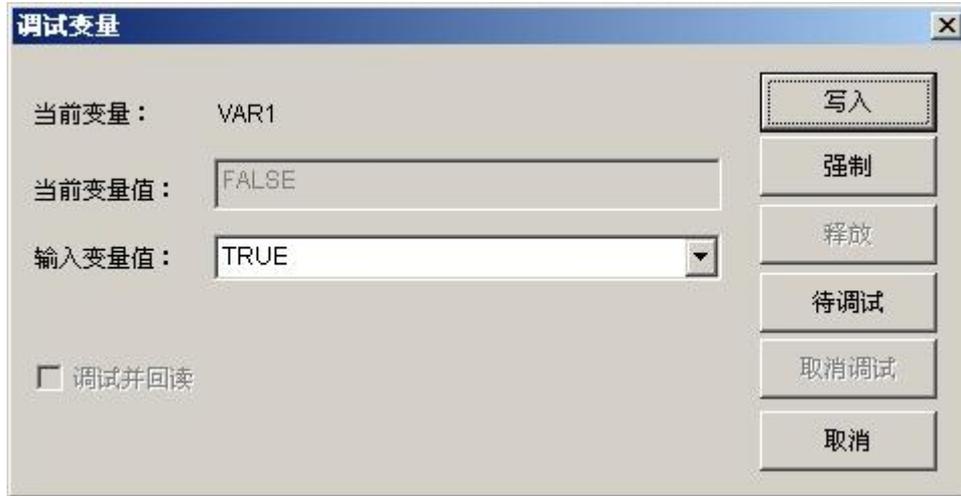


图 5.7-18 调试变量—写入

当前变量栏中显示当前被调试的变量名，当前变量值栏中显示当前值，在输入变量值中输入要修改后的值，单击写入按钮，变量值被修改。与强制操作一样，【写入】命令用于对待调试状态的多个变量进行批量写入操作，如果要单独对某一变量进行写入操作，直接双击该变量，使用“调试变量”对话框中的写入按钮进行写入操作即可。



- 当在工程选项中勾选了“调试回读”选项时，如图 5.7-19 所示，则调试对话框上，调试回读可用并默认被勾选。在写入（强制）后，变量的初始值被修改为写入（强制）的值，退出监视后，工程需要重新编译、下装。

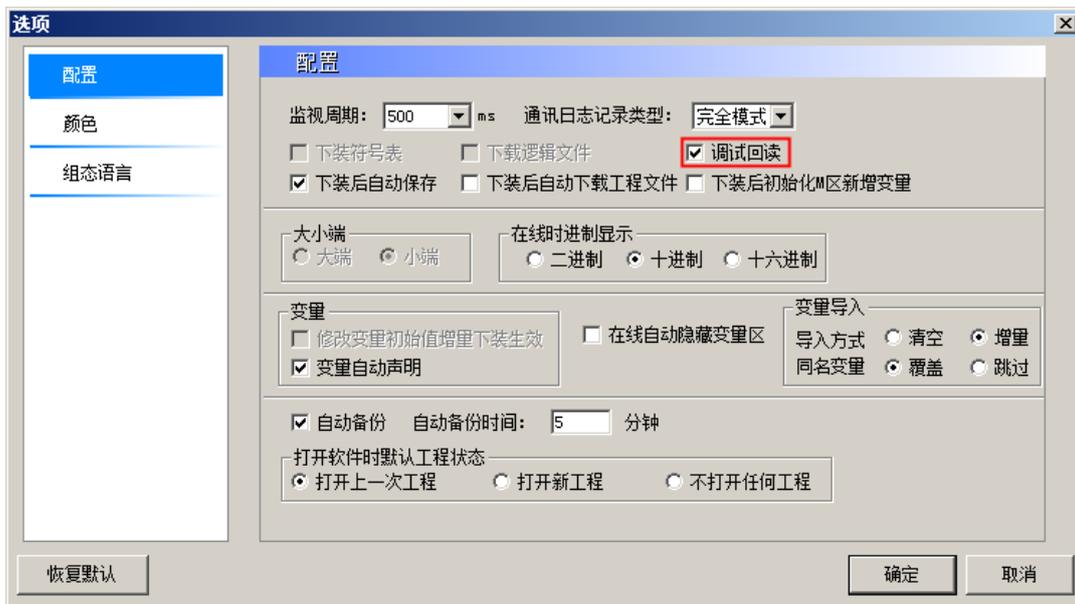


图 5.7-19 工程选项-调试回读

14. 强制



- 菜单栏：单击【在线】—【强制】；
- 快捷键：F7。

用于调试时对变量赋值。在每个循环结束之后，被强制的变量都被写入强制值，直到执行【解除强制】指令为止。

可以通过“调试变量”窗口对单个变量实施强制，也可以通过【在线】—【强制】命令批量强制多个变量。



- MC 工程不支持此功能。

■ 强制

- 单个变量强制：双击要进行强制操作的变量，弹出“调试变量”对话框，在输入变量值中输入要修改后的值，确定后，单击**强制**按钮，则强制完成，被强制后的变量显示为红色字体 **VAR1=TRUE**。

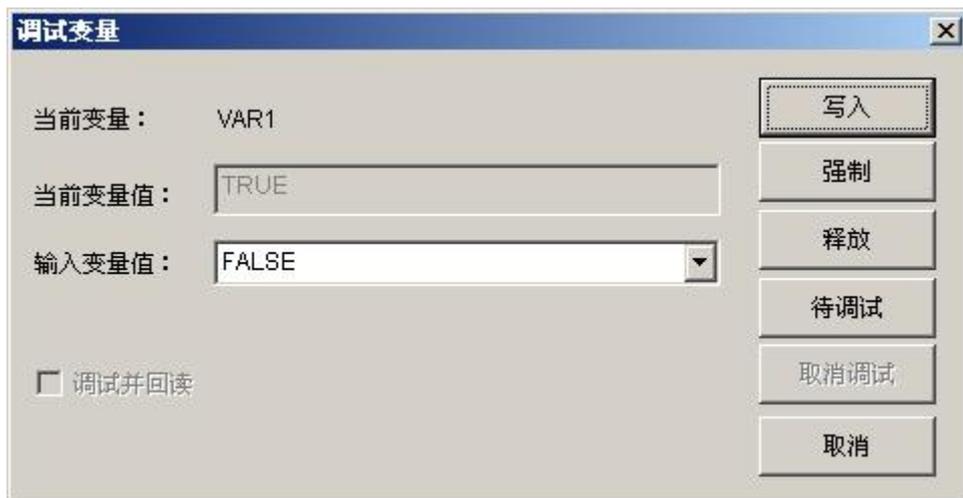


图 5.7-20 “调试变量”对话框

- 多个变量强制：在“调试变量”对话框，修改输入变量值后，单击**待调试**按钮，该变量的“调试变量”对话框关闭，变量值显示为 **VAR1=TRUE < FALSE**，前面的为该变量的当前值，后面为目标值。对多个变量执行上述操作后，均显示为待调试状态，这时执行【在线】—【强制】，则刚刚所选的所有的待调试的变量均被强制为各自设置的目标值，完成多变量强制。

■ 释放

通过“调试变量”对话框中的**释放**按钮可以对处于强制状态的变量进行强制释放。“调试变量”窗口只能对单个变量进行强制解除，如果要批量解除强制，请在【在线】—【强制变量表】中进行操作。

■ 待调试与取消调试

在“调试变量”对话框后，单击**待调试**按钮，该对话框关闭，且变量显示为待调试状态；再次双击该变量，按下“调试变量”对话框中的**取消调试**按钮，变量待调试状态取消。

15. 全部释放



- 菜单栏：单击【在线】—【全部释放】；
- 快捷键：**Shift+F7**。

用于释放工程中所有被强制的变量。



- MC 工程不支持此功能。

16. 强制变量表(L)



- 菜单栏：单击【在线】—【强制变量表】；
- 快捷键：**Ctrl+Shift+F7**。

在线或仿真状态，允许对多个变量进行强制写值，变量在被写入强制值后，直到**释放**操作后该强制值才解除。

选择该命令后，弹出窗口如图 5.7-21 所示。



图 5.7-21 强制变量列表

在“强制变量列表”窗口中列举当前所有的强制点，在此窗口可以集中处理各个强制点。逐个选中要释放的点，单击**释放**按钮，则只有选中的点解除强制操作；单击**全部释放**按钮，则该列表中所有显示的点解除强制操作。解除强制后的点从该列表中删除。



- MC 工程不支持此功能。

17. 急停

仅 MC 支持此功能。



工具栏: .

在线状态下可用。单击**急停**按钮，控制器停止运行、断开伺服使能端口、停止 IEC 任务、清空运动控制指令队列。

18. 使能

仅 MC 支持此功能。



工具栏：。

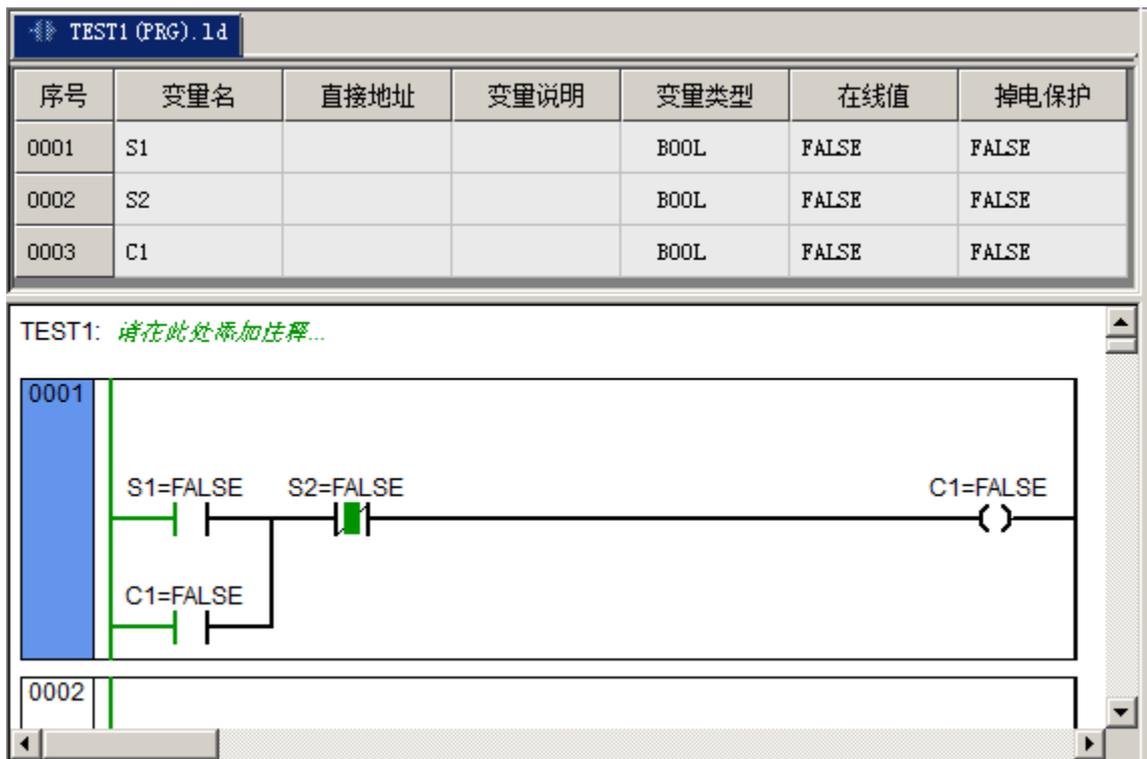
在线状态下可用。通过此按钮开启或关闭伺服使能端口。

5.7.4.2 在线调试

此模式下必须连接硬件设备，并将编译成功的程序通过正确的通道地址下装到主控中，然后才能进行程序调试。

调试步骤如下：

第1步 将程序下装到 CPU 模块中后，选择【监视】命令进入监视状态。



序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	S1			BOOL	FALSE	FALSE
0002	S2			BOOL	FALSE	FALSE
0003	C1			BOOL	FALSE	FALSE

TEST1: 请在此处添加注释...

0001 | S1=FALSE S2=FALSE C1=FALSE | C1=FALSE

0002 |

图 5.7-22 运行程序监视模式

第2步 双击触点 S1，在弹出的“调试变量”窗口中，将 S1 预置成“TRUE”，单击强制按钮确定强制值，如图 5.7-23 所示。



图 5.7-23 在线强制（1）

强制后的 S1 显示为红色，其相应的变量区内将显示在线值为 TRUE。如图 5.7-24 所示。

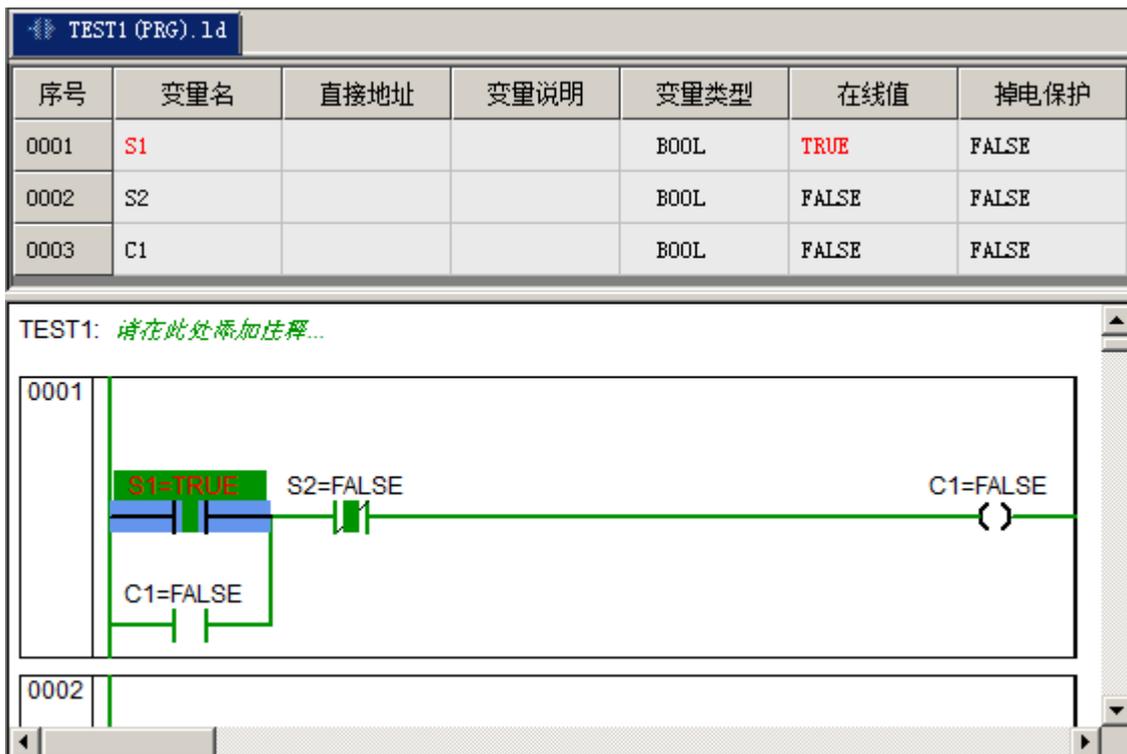


图 5.7-24 在线强制（2）

第3步 再次双击触点“S1”，在弹出的窗口中单击**释放**按钮，解除对“S1”触点的强制。如图 5.7-25 所示。



图 5.7-25 解除强制（1）

解除强制后，“S1”触点上将由强制时的红色变为黑色，且全局变量区内的赋值项也将恢复为黑色，如图 5.7-26 所示。

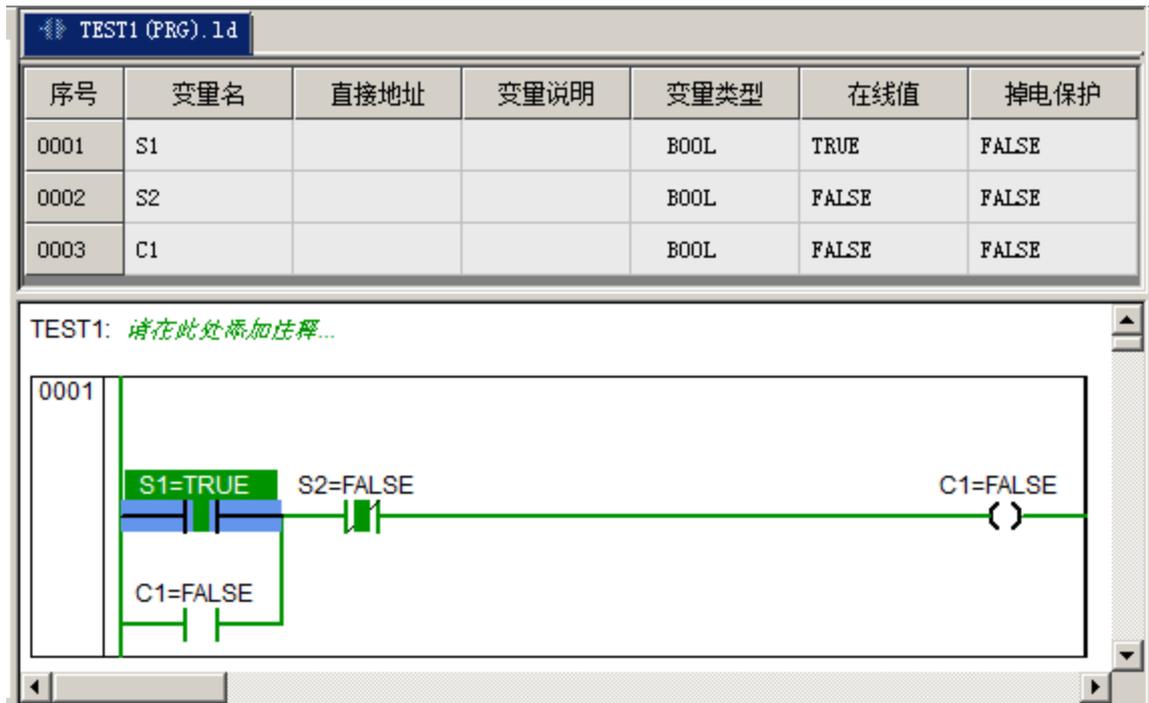


图 5.7-26 解除强制（2）

5.7.4.3 仿真调试

此模式下不需要连接硬件设备。工程编译通过成功后，执行【仿真】命令，进入仿真状态。

仿真调试过程的操作与 CPU 模块中在线调试的操作一致。



- MC、LK 工程不支持此功能。

在线/仿真对变量进行调试时，可写入二进制及十六进制数。写入值支持以下两种形式：“变量值类型#进制#变量值”和“进制#变量值”。如图 5.7-27 所示，仿真时变量调试写入的值为“int#16#10”，int#16#10 代表十进制数值为 16。

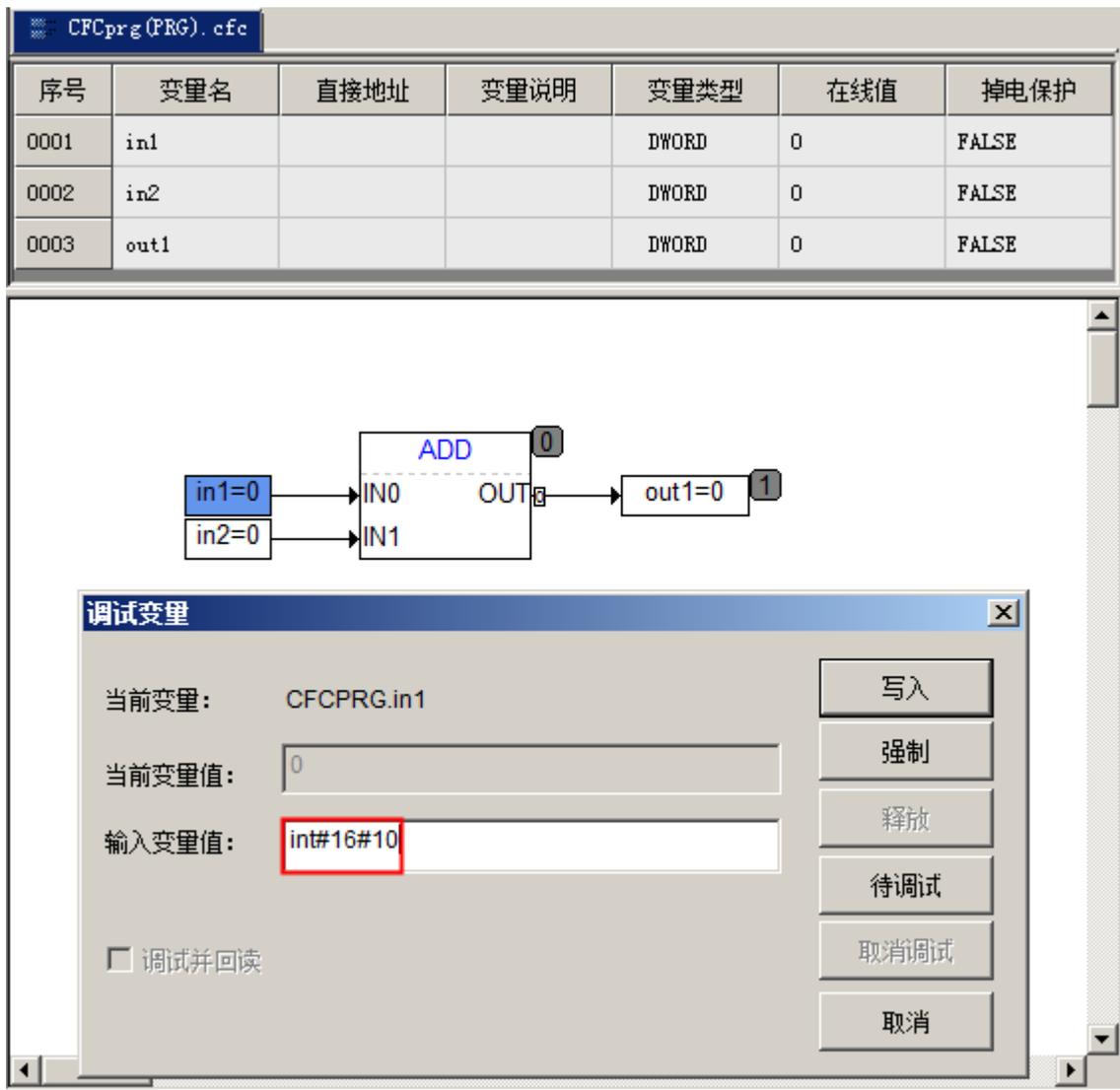


图 5.7-27 调试变量时写入十六进制数字

如图 5.7-28 所示的为写入变量值之后程序的运行结果。

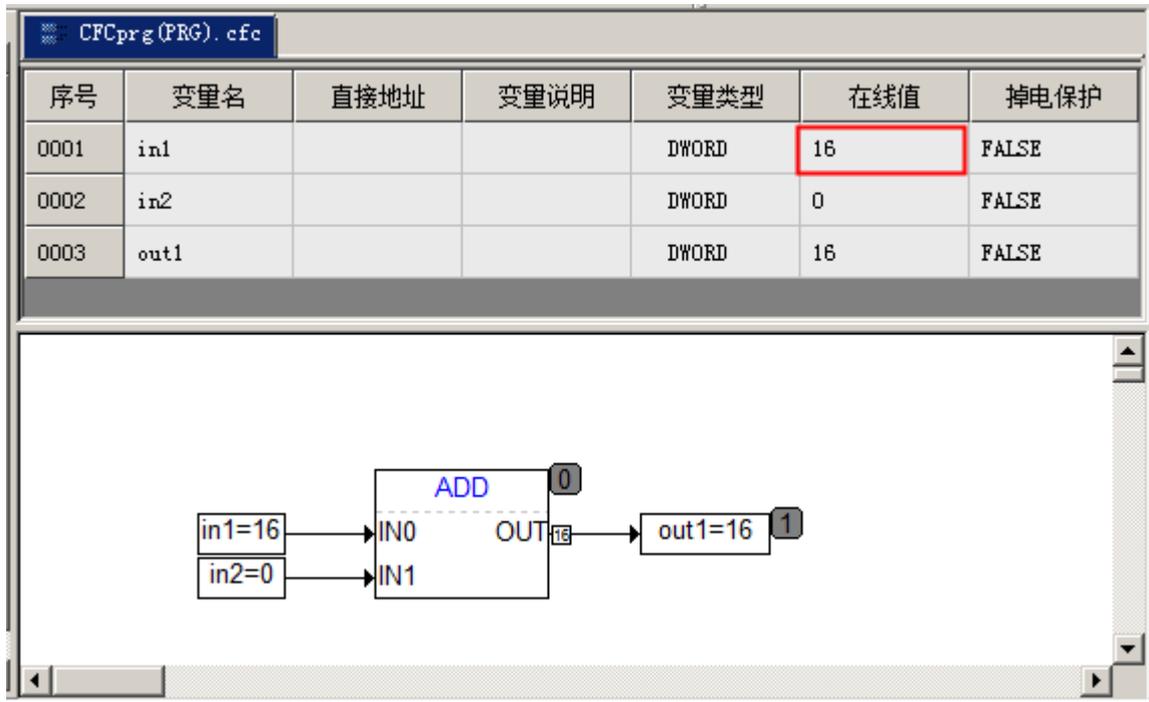


图 5.7-28 变量值写入之后的结果 (1)

除此之外还可以以“进制#变量值”的形式写入数值，如图 5.7-29 所示：写入“16#11”，代表十进制数值为 17，如图 5.7-30 所示的为写入变量值之后的运行结果。



图 5.7-29 以进制#变量值形式写入变量值

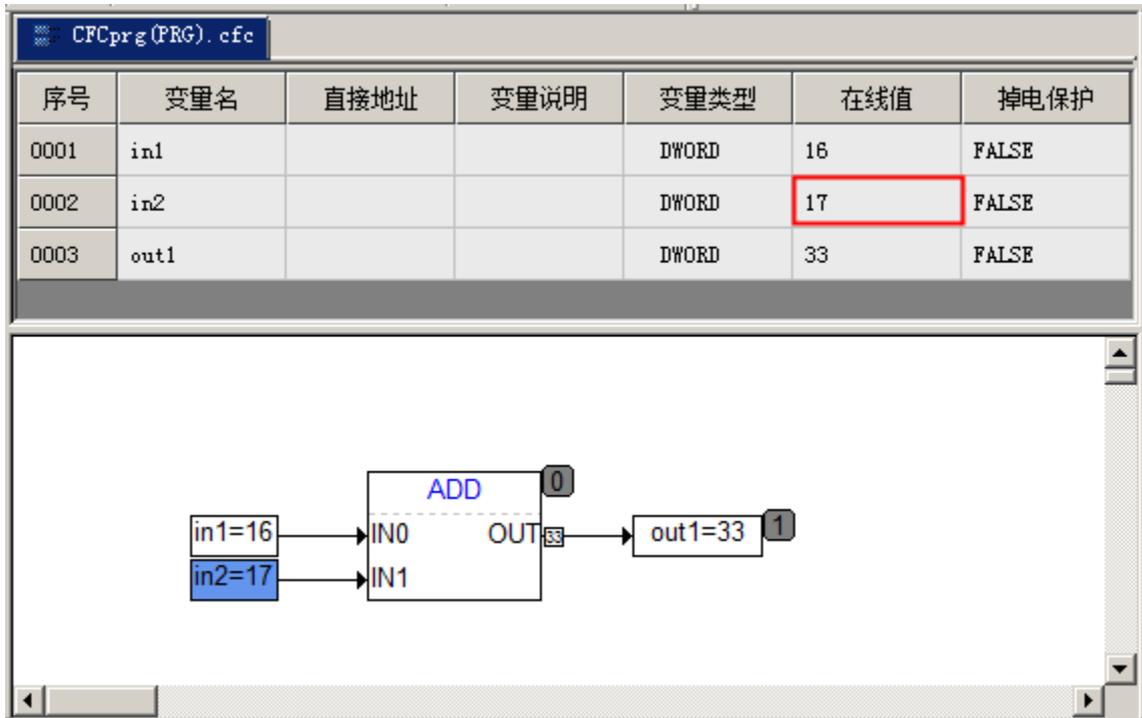


图 5.7-30 变量值写入之后的结果 (2)

5.7.4.4 终端

监视状态下，通过以下方式打开“终端”对话框。



- 菜单栏：单击【工具】—【终端】；
- 工具栏：。

打开“终端”对话框如图 5.7-31 所示。

在线状态下，用户借助这个工具可以与控制器进行直接交互，用户用 ST 语言将相关命令直接输入“终端”窗口，**Enter** 后，输入的命令会即时生效。

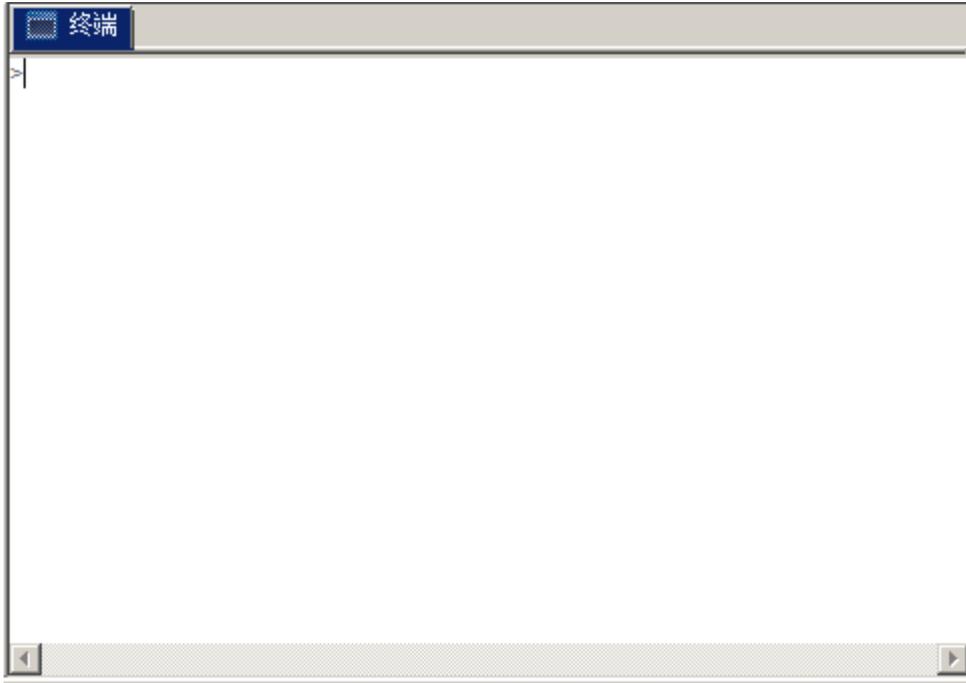


图 5.7-31 终端对话框

在该窗口中用户可以输入命令行，每个有效的命令行都以 `>` 开始，输入相关命令后，按 **Enter** 键，将内容按照从左到右的顺序下装到控制器，下装完成后会自动切换到下一个命令行。

如图 5.7-32 所示，在终端中设置轴 `axis0` 的速度为 `2000`，则轴参数列表中 `axis0` 的速度自动更新被终端写入的值。

```

>HMC_FORWARD(2);
>hmc_move(0,1.50,50);
>在函数'hmc_move'中参数个数多于其所定义的参数个数
>IF Main.p1=1 THEN Main.p3:=0;END_IF;
>在'='前需要':='
>IF Main.p1=1 THEN Main.p3:=0;END_IF;
>g1:=hmc_reverse(1);
>Main.p1:=1;
>axis0.peed:=1000;
>'peed'不是'axis0'的一个成员
>axis0.speed:=1000;
>axis0.speed:=2000;
>
    
```

(a)

轴参数	类型	Axis0	Axis1
StopFETol	LREAL	10	10
RepMode	SINT	2	2
RepDist	LREAL	10000000000	10000000000
EndPos	LREAL	0	-2
Remain	LREAL	0	-2
Velocity Profile			
Speed	LREAL	2000	2000
Accel	LREAL	200000	200000

(b)

图 5.7-32 终端应用示例

命令行中使用的变量必须是工程中已声明的全局变量和 PRG POU 中的变量，其中局部变量的书写格式：POU.变量名。用户可以使用输入助手 **F2** 键，打开帮助管理器，快速选择变量、关键字或 **ST** 操作符。也可以手动输入变量名，终端支持变量名联想功能。在命令行中输入 **POU**，则自动显示该 **POU** 下的所有变量名，如图 5.7-33 所示，选择变量名后，按 **Enter** 键即可。

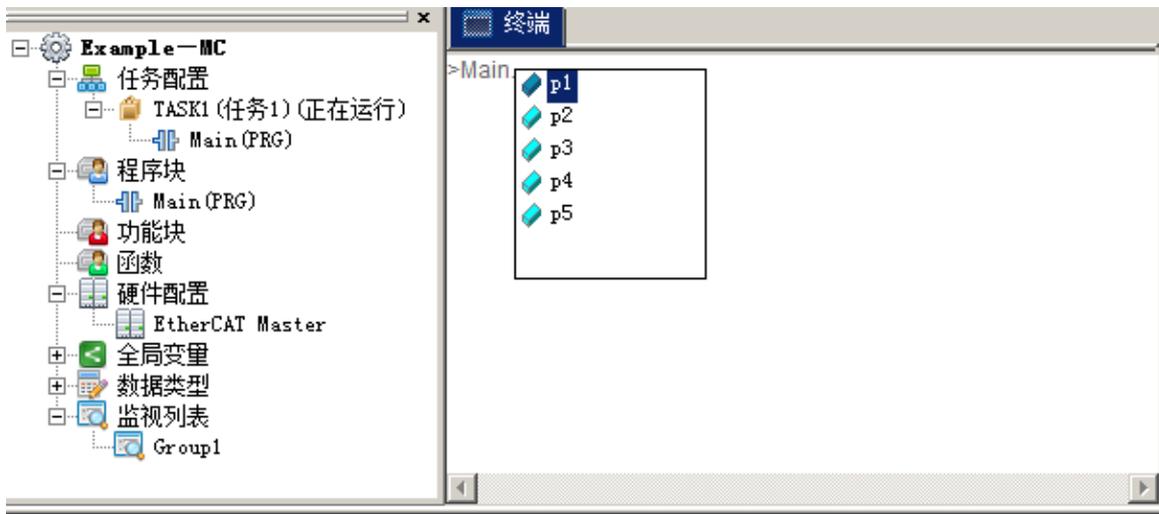


图 5.7-33 终端联想变量



- 此功能为 MC 工程特有。
- 在终端编辑命令函数时，允许不添加函数返回值，直接输入命令；如需返回值，输入即可。
- 用户通过组合键 **Ctrl+C** 可以使终端退出死循环。
- 一个命令行中最多可以输入 999 个字符，达到最大字符数时光标停止移动。

无论下装结果是成功还是失败，只要下装完成后，相关命令只允许复制。对当前正在编辑的命令行可以进行复制、剪切等操作。关闭“终端”窗口后再次打开时，不再显示之前的输入信息。

用户可以通过 ↑ 和 ↓ 键将已下装的一个命令复制到当前命令行。按 ↑ 键会根据从下至上的顺序对已下装的命令行进行选择、复制；↓ 键会根据从上向下的顺序进行。↑ 和 ↓ 键可以在最新的 50 个命令行中进行选择。

用户通过右键菜单中的【全部清除】命令可以快速删除“终端”窗口中的所有内容。

“终端”窗口中命令行的下装操作，会记录到操作日志中，方便用户查看。

5.7.5 查看操作日志



菜单栏：单击【工程】—【查看操作日志】。

该命令离线状态下可用。选择此命令，打开“操作日志”窗口，窗口中可以记录程序的在线操作信息：操作时间、操作内容、记录的操作条数。在操作日志中，可展开的节点前有“+”标记，可收缩的节点前有“-”标记。可通过单击该标记展开或收缩操作日志节点。如图 5.7-34 所示。

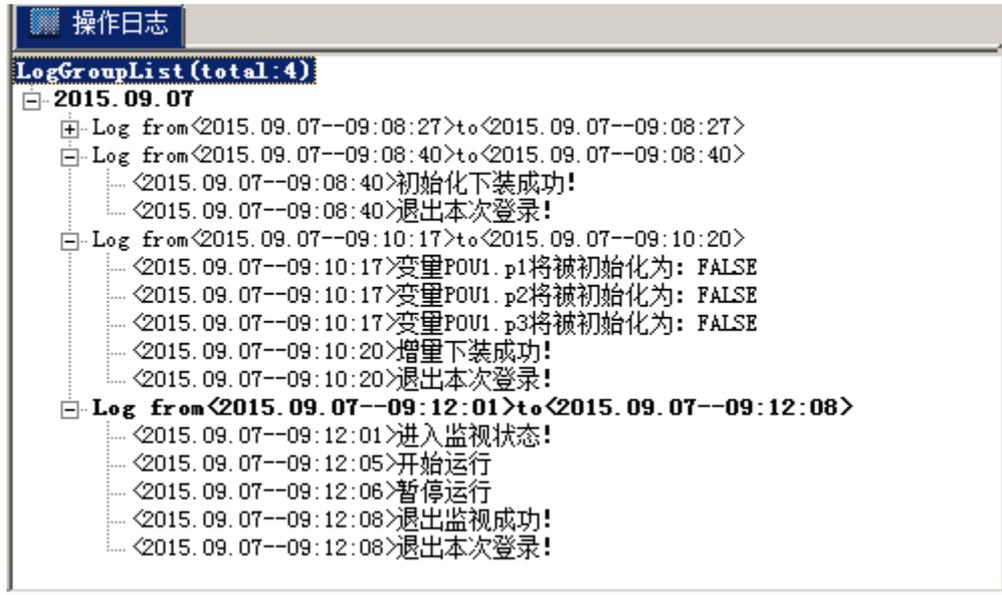


图 5.7-34 “操作日志”窗口



- 操作日志信息数量上限为 4000，记满后将生成备份文件，清空后重新开始记录，再次记满后生成的备份将覆盖掉之前的备份文件。
- 操作日志文件记录在安装目录下名称为“工程名.log”，备份日志文件为“工程名_bak.log”。

5.8 工具

5.8.1 PID 指令向导



- 菜单栏：单击【工具】—【PID 指令向导】。

PID 指令向导主要用来帮助用户配置 PID 指令的输入参数，最终在 LD 语言编辑区中自动生成一个 PID 实例(该实例的所有输入参数均已经配置完成)，其中 PID 指令的所有输入参数如表 5.8-1 所示。

表 5.8-1 PID 指令的输入参数

参数名称	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	FALSE: 无效 TRUE: 上升沿设定、高电平有效	FALSE
MVManual	REAL	手动输入值		0
Auto	BOOL	“自动”模式选择	FALSE: 手动 TRUE: 自动	FALSE
SP	REAL	设定值		0
PV	REAL	过程值		0
Derection	BOOL	“正作用”方式选择	FALSE: 反作用 (EV=SP-PV) TRUE: 正作用 (EV=PV-SP)	FALSE
TS	REAL	运算周期 (S)	$T_s > 0$ 并且应与调用该功能块的任务的间隔时间一致	0.05
KP	REAL	比例增益		1
TI	REAL	积分时间 (S)	$T_i \geq 0$, 如果积分时间为 0, 表示没有积分环节	1
TD	REAL	微分时间 (S)	$T_d \geq 0$, 如果微分时间为 0, 表示没有微分环节	0
DeadBand	REAL	偏差死区限	1) $DeadBand \geq 0$ 2) 若 $DeadBand = 0$, 表示没有死区 3) 偏差绝对值大于该值时有效, 否则偏差为 0	0
MVBias	REAL	前馈控制量		0.0
MVMax	REAL	输出值上限		+100
MVMin	REAL	输出值下限		-100

第1步 手/自动模式选择

可以选择 PID 指令为自动模式(Auto)或者手动模式(MVManual)。在选择手动模式时, 需要设置手动输入值; 自动模式下手动输入值为不可编辑状态, 如图 5.8-1 所示。

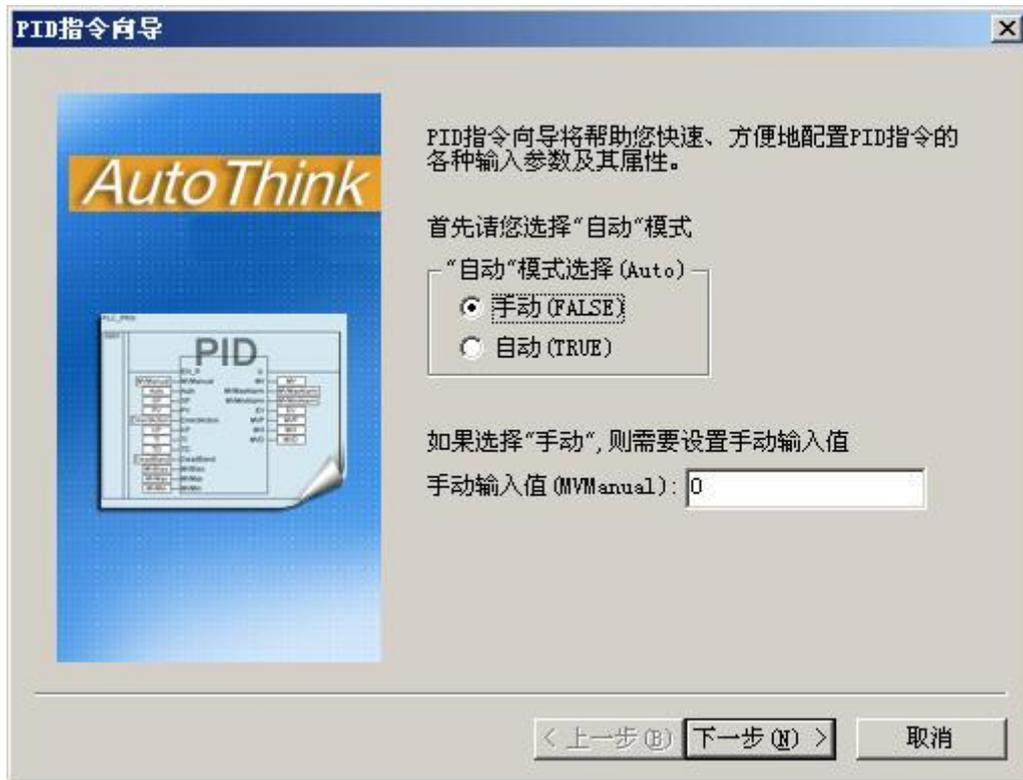


图 5.8-1 PID 指令的“自动”模式选择

第2步 配置 PID 指令的设定值(SP)、过程值(PV)以及选择正作用方式，如图 5.8-2 所示。

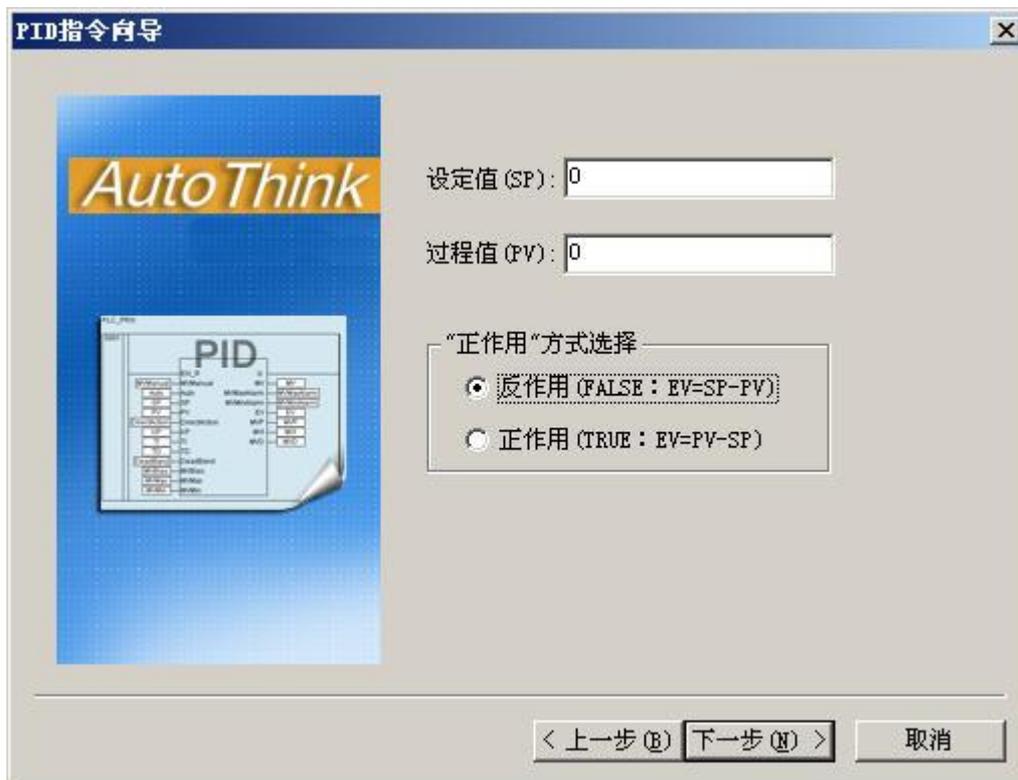


图 5.8-2 PID 指令的设定值、过程值及“正作用”方式选择

第3步 配置 PID 指令的积分时间(TI)和微分时间(TD)，如图 5.8-3 所示。

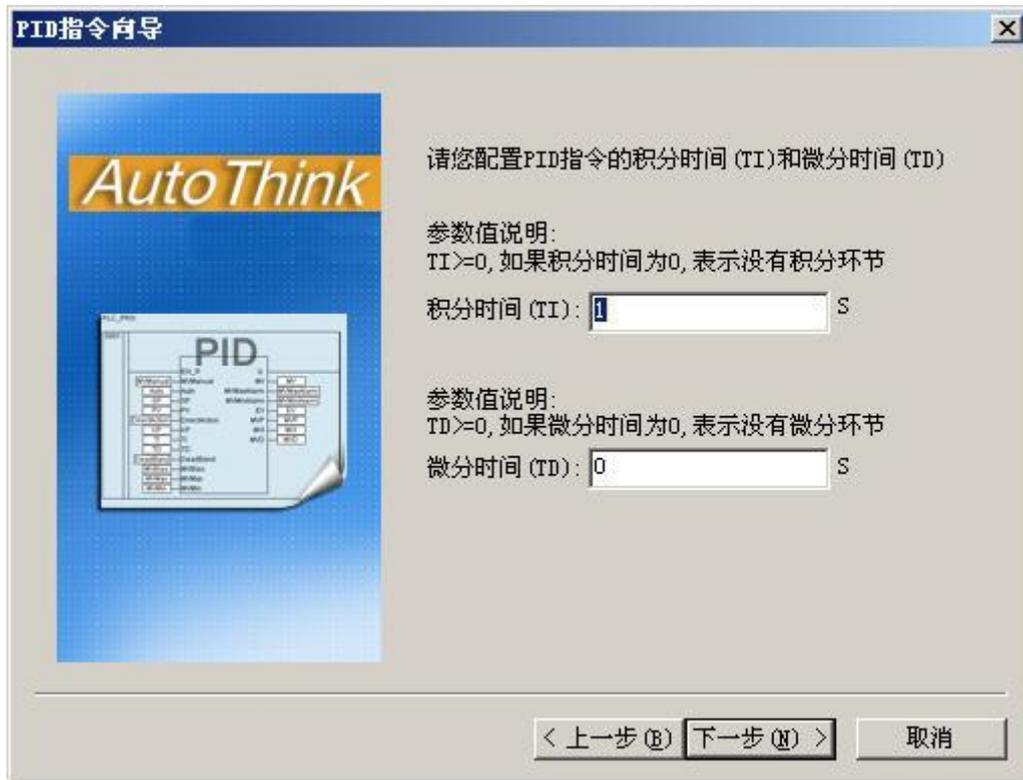


图 5.8-3 PID 指令的积分时间和微分时间

第4步 配置 PID 指令的运算周期(TS)和偏差死区限(DeadBand)，如图 5.8-4 所示。

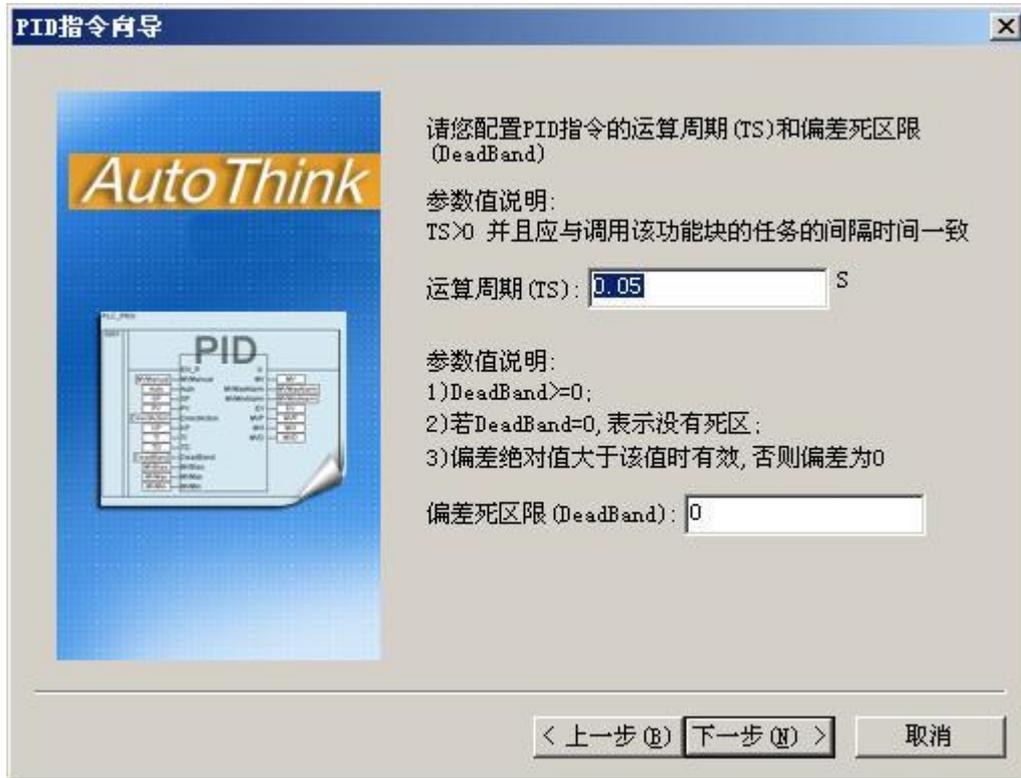


图 5.8-4 指令的运算周期和偏差死区限

第5步 配置 PID 指令的比例增益(KP)、前馈控制量(MVBias)、输出值上限(MVMax)及输出值下限(MVMin)，最终完成 PID 指令所有输入参数的配置，如图 5.8-5 所示。

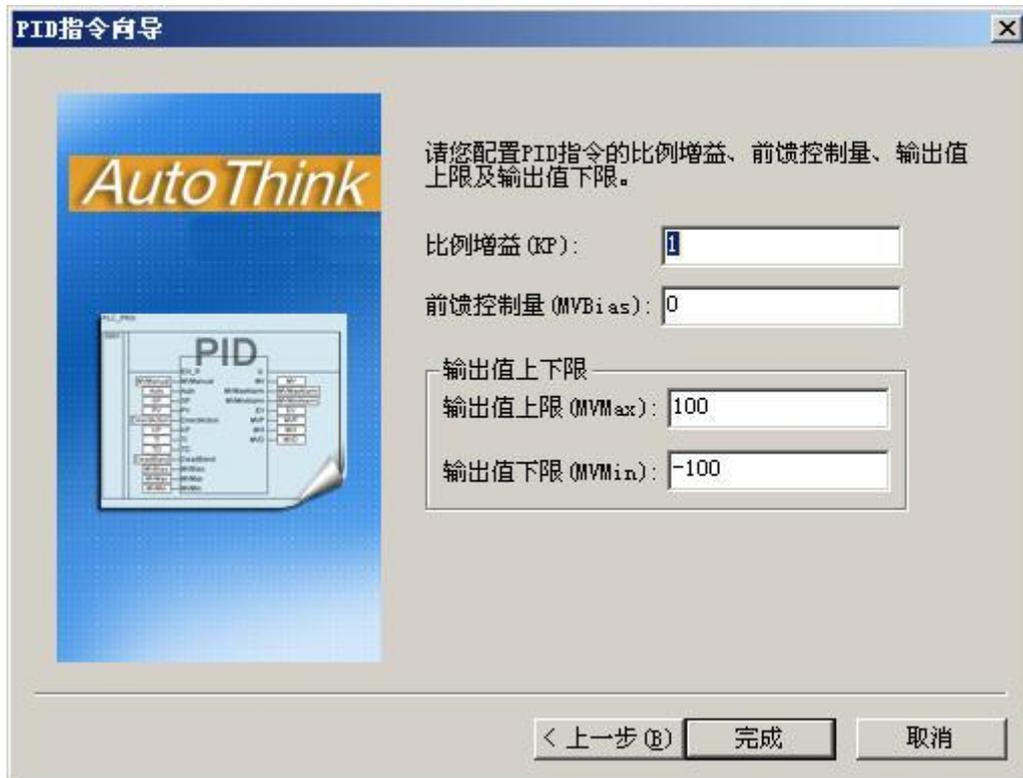


图 5.8-5 PID 指令的比例增益、前馈控制量及输出值上下限

单击**完成**后，弹出 PID 指令变量自动声明对话框，如图 5.8-6 所示，完成变量声明后，PID 指令就插入完成了，如图 5.8-7 所示。



图 5.8-6 PID 指令向导配置完成后弹出变量自动声明对话框

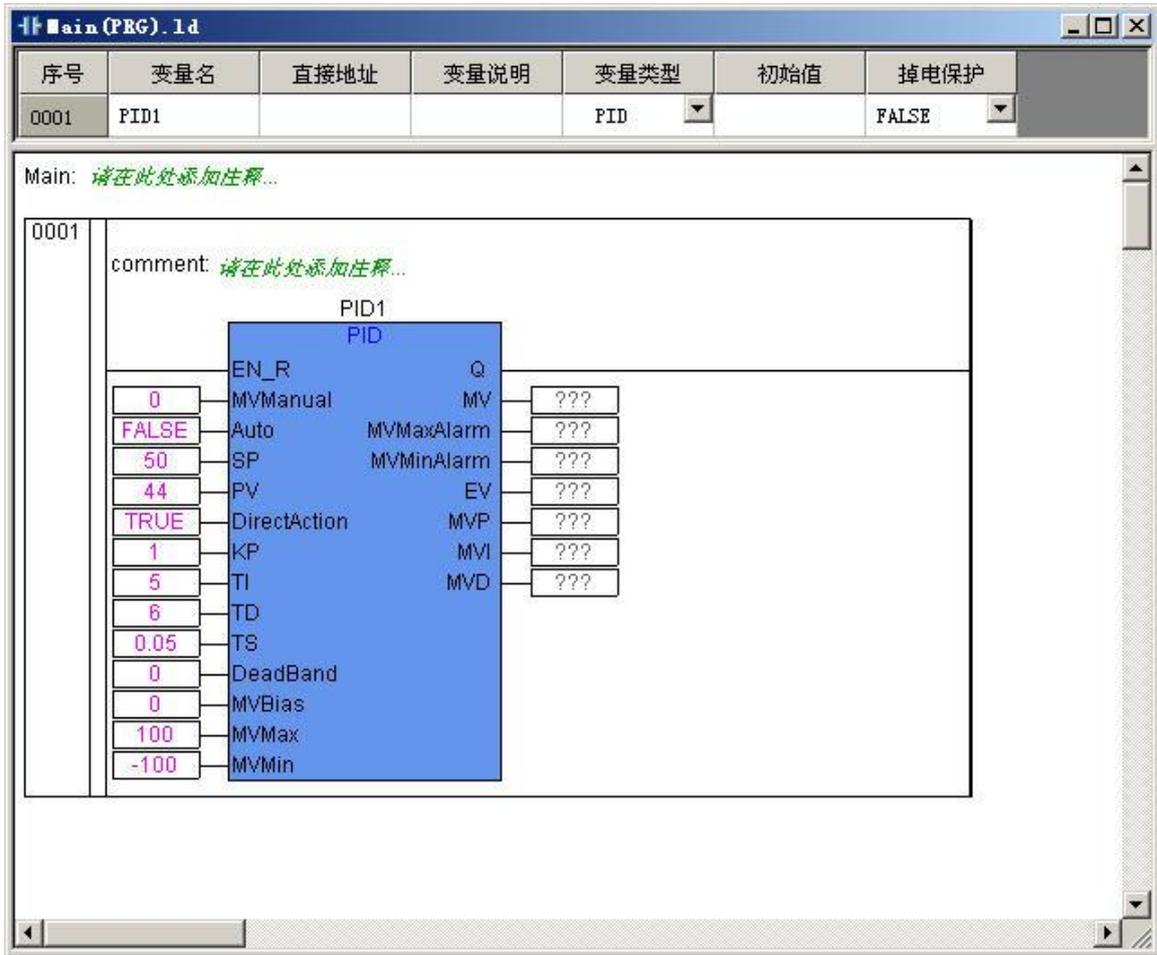


图 5.8-7 PID 指令向导完成

5.8.2 串口通讯指令向导



- 菜单栏：单击【工具】—【串口通讯指令向导】。

使用此向导，可以帮助用户在 LD 语言编辑区中配置生成串口通讯指令。



- MC、LK 工程不支持此功能。

串口通讯指令主要包括 Modbus 从站通讯、Modbus 主站通讯及自由口通讯等 3 种类型的应用，因此，其指令向导的参数含义、配置界面等将分别根据用户选择的应用类型的不同而不同。其中，串口通讯指令向导的通讯口类型选择和应用类型选择的显示界面如图 5.8-8 所示。



图 5.8-8 串口通讯指令向导通讯口类型和应用类型选择

下面针对用户选择的应用类型（Modbus 从站通讯、Modbus 主站通讯、自由口通讯），分别介绍其指令参数配置步骤。

5.8.2.1 Modbus 从站通讯

当用户选择 Modbus 从站通讯应用时，只需要提示用户到硬件配置管理窗口的通讯参数信息属性页中设置 Modbus 从站地址的相关属性即可，而并不需要在 LD 语言编辑区中生成具体的实例。其信息提示界面如图 5.8-9 所示。



图 5.8-9 Modbus 从站通讯指令的信息提示界面

5.8.2.2 Modbus 主站通讯

Modbus 主站通讯指令(MODBUS_MASTER)主要用于实现串口 Modbus 协议的主站数据读写操作，当用户选择该应用时，则需要分五步来完成其所有参数的配置。其所有输入参数见表 5.8-2。

表 5.8-2 Modbus 主站通讯指令的输入参数

参数名称	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0: 无效 1: 上升沿使能，且高电平有效	0
Port	BYTE	通讯口选择	0: CPU 本体的 PS/2 1: CPU 本体端子通讯口（14 点不支持） 2: 扩展功能板通讯口（14 点不支持） 3: 第一个通讯扩展模块通讯口 4~N: 第二个到第 N 个通讯扩展模块通讯口	0
Slave	BYTE	从站地址	Modbus 从站地址，1-247	1
RW	BOOL	读/写选择	0: 读取数据 1: 写入数据	0
DataAddress	DWORD	从站存放数据的地址	详见表 5.8-3	1
DataLength	BYTE	数据长度	1-100. 对于开入/开出为所需要传输的总比特数。对于模入/模出为所要传输的总通道数	1

参数名称	数据类型	功能描述	参数值说明	默认值
TBL	WORD	主站存放数据的首字地址	诸如 200，表示存放地址为%MW400 开始的一段空间。如果是读指令，读回的数据值存放到这个数据区中；例如 3 号从站 3050 地址的数据为 1000，则%MW400 存放 1000。如果是写指令，要写出的数据值放到这个数据区中；例如向 3 号从站的 3050 地址写入 500，则 %MW400 存放 500。	0
Timeout	WORD	超时时间设置 (ms)	最小时间单位为 50 ms	0
INT_EN	BOOL	中断使能	0: 禁止中断 1: 中断使能 通讯口发送信息完成和接收信息完成各产生一个中断。中断列表中的通信口中断 0 和通信口中断 1。	0

表 5.8-3 DataAddress 参数详细说明

参数说明	
最高位	最高位
0: 开出	0: 开出 1: 开入 3: 模入 4: 模出
1: 开入	
3: 模入	
4: 模出	
从站的地址采用标准 Modbus RTU 驱动的地址填写方式，即采用六位数（十进制），第六位只能选择 0、1、3、4，分别表示 Modbus 不同的数据类型，低五位表示寄存器地址（地址范围 0-65535）。 例如：000001，表示 Modbus 地址为 0001 的开出点 403050，表示 Modbus 地址为 3050 的模出点	

下面分五步来详细叙述 Modbus 主站通讯指令的参数配置情况。

第1步 选择 Modbus 主站通讯指令的读写属性(RW)，如图 5.8-10 所示。



图 5.8-10 Modbus 主站通讯指令的读写属性

第2步 配置从站的数据存放地址(DataAddress)，如图 5.8-11 所示。其中 Modbus 数据类型下拉框中将有 4 个选项，它们分别是 0：开出、1：开入、3：模入、4：模出，默认选项为 0：开出。



图 5.8-11 Modbus 主站通讯指令的从站数据存放地址

第3步 配置 Modbus 从站地址及主站发送或者接收数据的长度(DataLength), 其单位为 BYTE, 如图 5.8-12 所示。



图 5.8-12 Modbus 从站地址及主站通讯指令发送或者接收数据的长度

第4步 配置 Modbus 主站存放数据的首地址(TBL), 如图 5.8-13 所示。

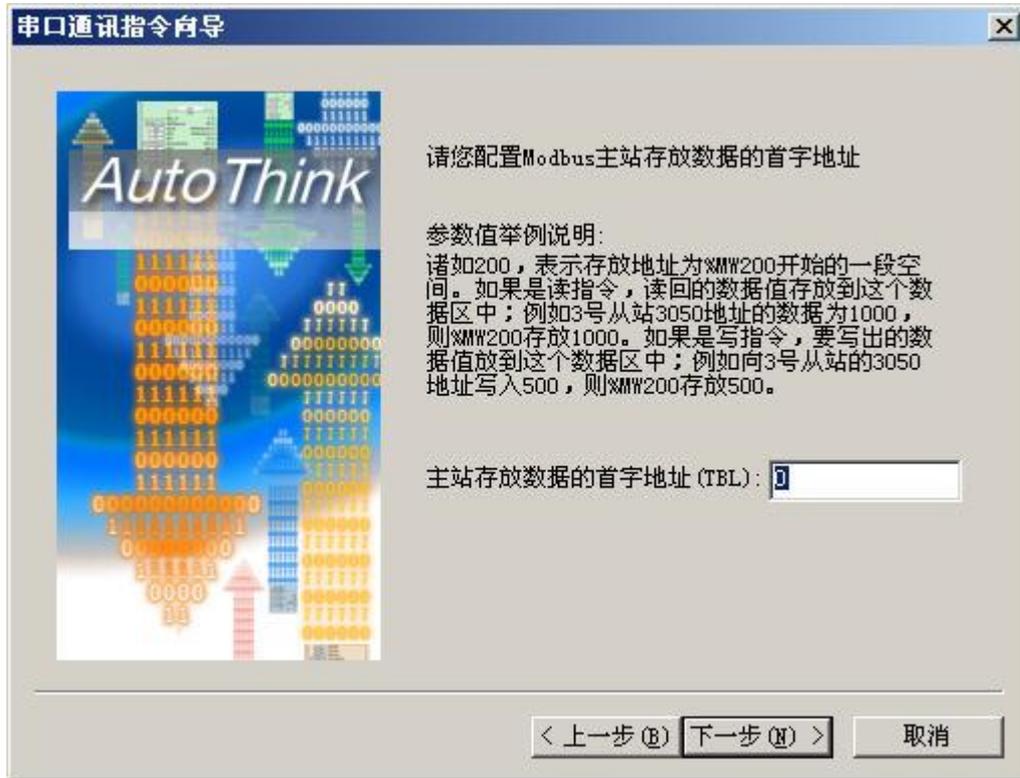


图 5.8-13 Modbus 主站存放数据的首地址

第5步 配置超时时间(Timeout), 如果选择**禁止超时**, 则其超时时间默认配置为 0 ms, 否则将其配置为大于或等于 50 ms 的时间值; 配置中断使能标识(Interrupt_EN), 最终完成 Modbus 主站通讯指令的所有参数配置, 如图 5.8-14 所示。



图 5.8-14 超时时间

单击**完成**后，弹出 Modbus 主站通讯变量自动声明对话框，完成变量声明后，MODBUS_MASTER 指令会自动出现在指定的插入地点，如图 5.8-15 所示。

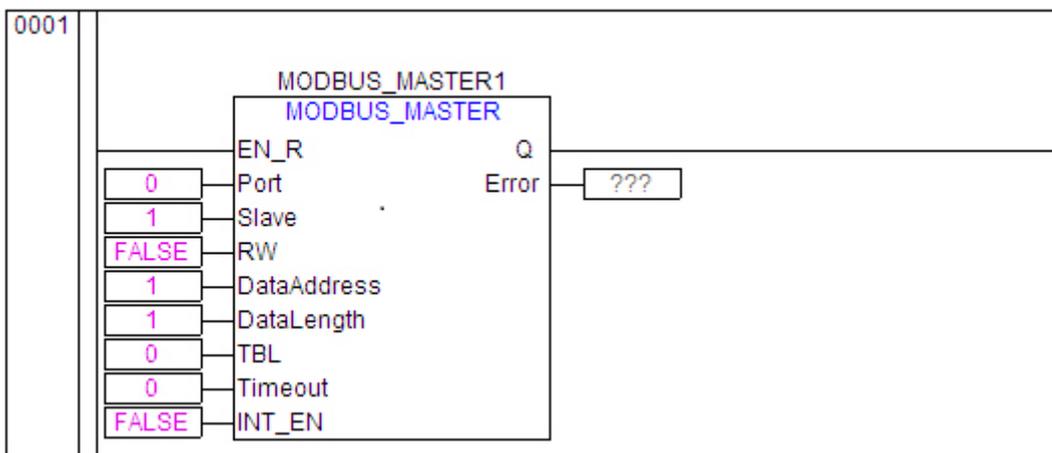


图 5.8-15 Modbus 主站配置完成

5.8.2.3 自由口通讯

自由口通讯主要包括只发送数据、只接收数据、发送与接收数据等三个方面的应用，如图 5.8-16 所示。当用户选择该指令类型时，将分三种情形进行参数的配置，下面分别进行介绍。



图 5.8-16 自由口通讯指令三个方面的应用



- 如果应用中存在接收数据指令，则系统会自动在生成的接收数据指令实例前添加一个置反的触点，其触点名称为“接收数据指令名称.Q”，如图 5.8-17 所示。

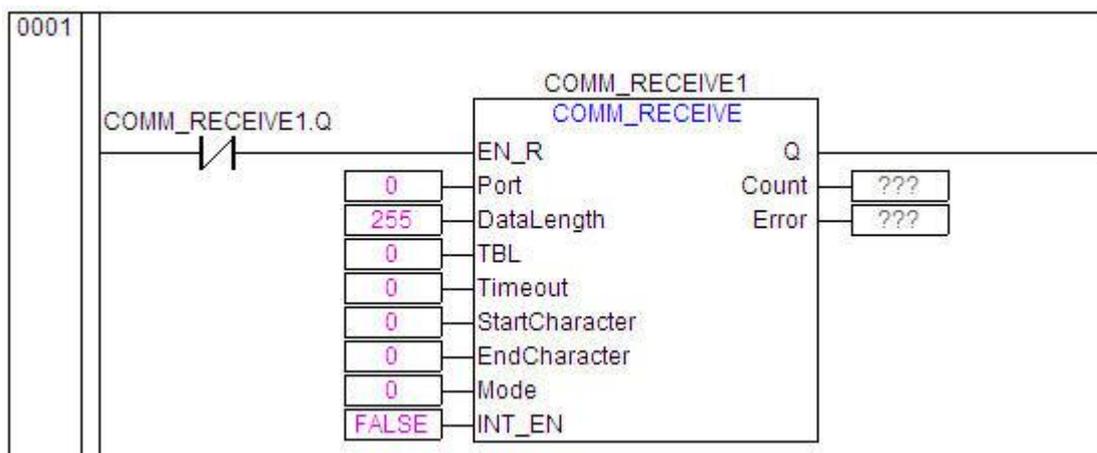


图 5.8-17 接收数据指令实例前的置反触点

1. 只发送数据

自由口通讯指令只应用于发送数据时，总共可分为两步完成其参数的配置，发送数据指令 (COMM_SEND)所有的输入参数如表 5.8-4 所示。

表 5.8-4 发送数据指令的输入参数

参数名称	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0: 无效 1: 上升沿使能，且高电平有效	0
Port	BYTE	通讯口选择	0: CPU 本体的 PS/2 1: CPU 本体端子通讯口 (14 点不支持) 2: 扩展功能板通讯口 (14 点不支持) 3: 第一个通讯扩展模块通讯口 4~N: 第二个到第 N 个通讯扩展模块通讯口	0
DataLength	BYTE	发送字节数	发送的字节数，1-255 个字节。	255
TBL	WORD	数据存放地址	指向 M 区数据存放的首字节地址。诸如 200，表示存放地址为%MB200 开始的一段空间。	0
INT_EN	BOOL	中断使能输入	0: 发送完数据后不产生中断 1: 发送完数据后产生中断 中断程序名称通过中断配置界面配置	0

第1步 配置其发送数据的存放地址(TBL)，如图 5.8-18 所示。



图 5.8-18 发送数据的存放地址

第2步 配置发送数据的字节数(DataLength)及中断使能输入(Interrupt_EN)属性, 最终完成发送数据指令的所有参数配置, 如图 5.8-19 所示。



图 5.8-19 发送数据的字节数和中断使能输入属性

2. 只接收数据

自由口通讯指令只用于接收数据应用时, 总共可分为三步完成其所有参数的配置, 接收数据指令 (COMM_RECEIVE)的所有输入参数如表 5.8-5 所示。

表 5.8-5 接收数据指令的输入参数

参数名称	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0: 无效 1: 上升沿使能, 且高电平有效	0
Port	BYTE	通讯口选择	0: CPU 本体的 PS/2 1: CPU 本体端子通讯口 (14 点不支持) 2: 扩展功能板通讯口 (14 点不支持) 3: 第一个通讯扩展模块通讯口 4~N: 第二个到第 N 个通讯扩展模块通讯口	0
DataLength	BYTE	接收字节数	接收的字节数, 0-255 个字节, 设为 0 是不定长接收模式	255
TBL	WORD	数据存放地址	指向 M 区数据存放的首字节地址, 诸如 200, 表示存放地址为 %MB200 开始的一段空间	0
Timeout	WORD	帧超时时间	从启动接收过程开始计算, 在规定的时间内没有再接收到	0

参数名称	数据类型	功能描述	参数值说明	默认值
		(ms)	完整的数据包，中止接收过程。最小值为 50ms，最小时间单位为系统调度时间	
StartCharacter	WORD	开始字符	当启用开始字符时，只有接收到开始字符才表征一帧数据的开始，否则被丢弃	0
EndCharacter	WORD	结束字符	当启用结束字符时，只有接收到结束字符才表征一帧数据的结束，否则继续接收直到缓冲区满（255 字节）	0
Mode	BYTE	控制模式参数设置	详见表 5.8-6	0
INT_EN	BOOL	中断使能输入	0: 接收完数据后不产生中断 1: 接收完数据后产生中断 中断程序名称通过中断配置界面配置	0

表 5.8-6 Mode 参数详细说明

Mode (BYTE)							
7	6	5	4	3	2	1	0
未定义				开始、结束字符数据长度控制	结束字符控制	开始字符控制	超时控制
超时控制							
0: 禁止超时控制 1: 启动超时控制							
开始字符控制							
0: 禁止开始字符控制 1: 启动开始字符控制							
结束字符控制							
0: 禁止结束字符控制 1: 启动结束字符控制							
开始、结束字符数据长度控制							
0: 开始、结束字符为字 1: 开始、结束字符为字节							

第1步 配置接收数据的存放地址(TBL)，如图 5.8-20 所示。



图 5.8-20 接收数据的存放地址

第2步 设置接收数据指令的控制模式，如图 5.8-21 所示。

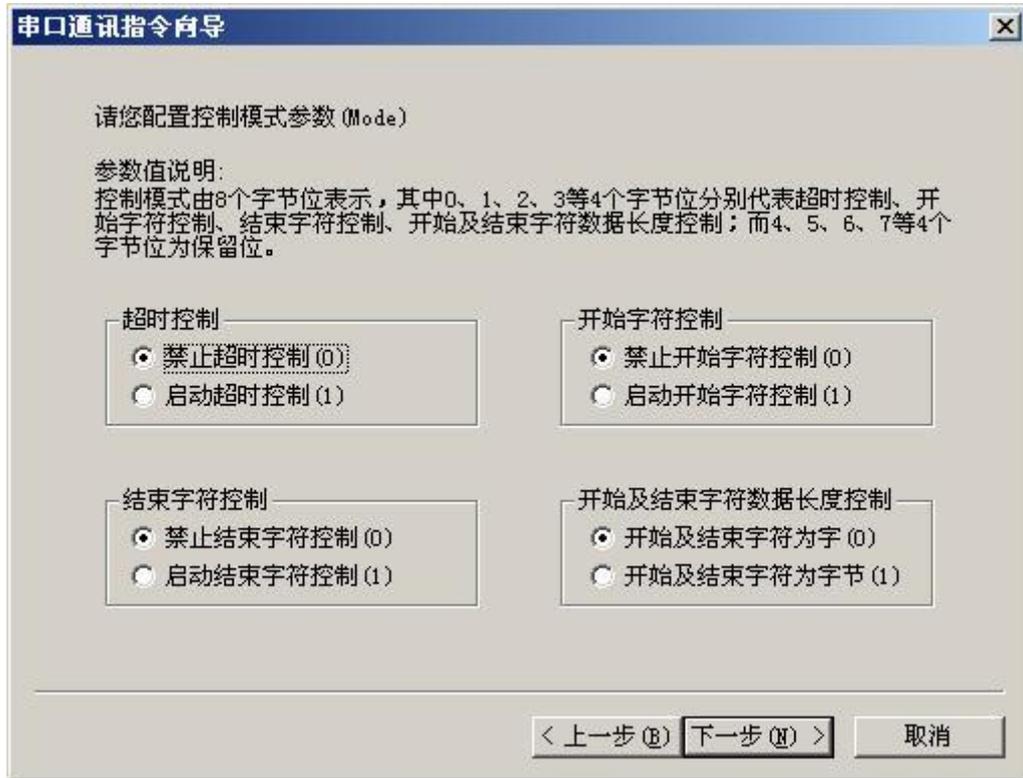


图 5.8-21 接收数据指令的控制模式

第3步 配置接收数据的字节数(DataLength)、帧超时时间(Timeout)、开始字符(StartCharacter)、结束字符(EndCharacter)及中断使能输入(Interrupt_EN)属性；如果第二步中选择了**禁止超时控制**、**禁止开始字符控制**、**禁止结束字符控制**等三项，则在第三步中将**帧超时时间**、**开始字符**、**结束字符**等编辑区置为灰色禁用状态，其值均默认为0。通过该步最终完成接收数据指令的所有参数配置，如图 5.8-22 所示。



图 5.8-22 接收数据的字节数、帧超时时间、开始字符、结束字符及中断使能输入

3. 发送与接收数据

自由口通讯指令同时应用于发送与接收数据时，其参数的配置步骤由上面两部分（只发送数据和只接收数据）所述的步骤结合完成。

5.8.3 轴参数显示设置



- 此功能为 MC 工程特有。



- 菜单栏：单击【工具】—【轴参数显示设置】。

打开“轴参数显示设置”对话框，如图 5.8-23 所示。MC CPU 支持 64 轴，默认只显示前几轴（具体的显示项与当前 CPU 型号有关）。轴参数显示设置支持用户自定义显示的轴号和轴参数。此处配置的轴号和轴参数同步在轴参数列表中显示。



图 5.8-23 选择轴号



图 5.8-24 选择轴参数

5.8.4 查看轴参数



- 此功能为 MC 工程特有。



- 菜单栏：单击【工具】—【查看轴参数】；
- 工具栏：。

打开“轴参数”窗口如图 5.8-25 所示。

轴参数表中显示的轴和轴参数是用户在【轴参数显示设置】中配置的。第一行显示的是各轴轴号，第一列显示的是轴参数名称，第二列为轴参数对应的类型。轴参数以分组的形式显示，方便用户查看。单击轴参数组前的 □、田 图标可以对轴参数组进行折叠和展开。

轴参数的读写属性由系统预先定义。轴参数列表中白色区域是可写属性的轴参数，可以修改它的初值和在线值。灰色区域是只读属性的轴参数，只能查看，无法修改参数值。

轴参数						
轴参数	类型	Axis0	Axis1	Axis2	Axis3	
Basic						
AxisID	USINT	0	1	2	3	
AxisType	EmAxisType	Virtual	Virtual	Virtual	Virtual	
Units	LREAL	1	1	1	1	
Status						
Positions						
MPos	LREAL	0	0	0	0	
DPos	LREAL	0	0	0	0	
FE	LREAL	0	0	0	0	
StopFETol	LREAL	1.00E+01	1.00E+01	1.00E+01	1.00E+01	
RepMode	SINT	2	2	2	2	
RepDist	LREAL	1.00E+10	1.00E+10	1.00E+10	1.00E+10	
EndPos	LREAL	0	0	0	0	
Remain	LREAL	0	0	0	0	
Velocity Profile						
Speed	LREAL	2.00E+03	2.00E+03	2.00E+03	2.00E+03	

图 5.8-25 轴参数表



- MC 工程中，轴参数作为系统变量已在全局变量组【AxisParaGroup】中预先定义。64 个轴的轴参数变量名默认为 Axis0~Axis63，支持用户对轴参数变量重新命名。
- 轴参数变量为功能块类型，引脚是该轴的所有参数。比如设置轴 1 的 Speed 为 1000，可以使用以下语句：“Axis01.Speed := 1000;”。
- 在轴参数表的右键菜单中集成了【轴参数显示配置】功能。

5.8.5 示波器



- 此功能为 MC 工程特有。



- 菜单栏：单击【工具】—【示波器】；
- 工具栏：

打开“示波器”窗口如图 5.8-26 所示。

组态软件对用户出示波器功能，类似真实的示波器，用户在示波器界面里进行配置，运行后，示波器根据控制器采集的通道数据进行波形或轨迹绘制，同时支持将数据导出至文件中。

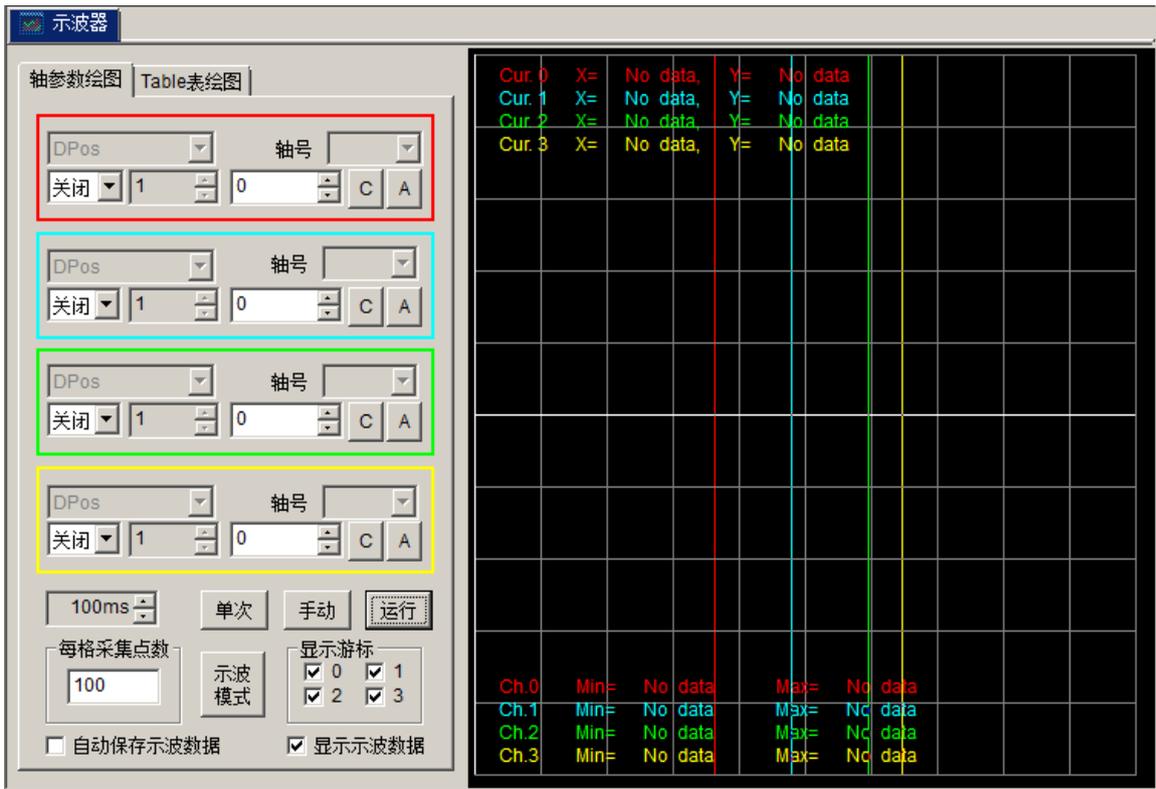


图 5.8-26 示波器界面

5.8.5.1 轴参数绘图设置

1. 示波显示设置

示波器有 4 路示波通道。每路通道都有一个相应的控制组，如图 5.8-27 所示。



图 5.8-27 示波通道控制组

■ 轴参数



图 5.8-28 轴参数设置

轴参数列表框位于控制组的左上角，示波器可记录和显示的参数在这个列表框中选择。

■ 轴号



图 5.8-29 轴号设置

轴号从此列表框中选择。

■ 运行模式



图 5.8-30 运行模式设置

通道的运行模式控制波形（轨迹）如何显示和缩放。

表 5.8-7 通道的运行模式列表

模式	说明
关闭	不采集数据，不显示波形（轨迹）
自动	采集数据，自动缩放到合适的比例显示
手动	采集数据，用户指定比例
冻结	不采集数据，显示冻结时的波形（轨迹）

■ 纵向比例



图 5.8-31 比例显示框

自动模式下，示波器计算出合适的显示比例，并显示在比例文本框中。

手动模式下，用户指定单格比例。纵向比例通过文本框右侧的上下调节按钮  修改。

■ 纵向偏移



图 5.8-32 纵向偏移设置

表 5.8-8 控制纵向偏移的按钮

按钮	说明
	纵向偏移按钮，在竖直方向上移动波形（轨迹）。特别是，当两个或多个波形（轨迹）重合，只显示最上方的波形（轨迹）时，使用此按钮可以将它们分开
	零偏移按钮，将纵向偏移置为零
	自动偏移按钮，将波形（轨迹）居中显示。此按钮激活时，纵向偏移按钮和零偏移按钮不可用（置灰），由示波器计算偏移。此按钮未激活时，用户手动设置的偏移生效，纵向偏移按钮和零偏移按钮处于可用状态

2. 采集参数设置

示波器每屏有 10 格，用户可设置的采集参数包括时间基准和单格数据点数。如图 5.8-33 所示。



图 5.8-33 采集参数设置

■ 时间基准

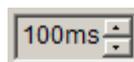


图 5.8-34 时间基准设置

时间基准通过文本框右侧的上下比例按钮调节。选择的值作为单格数据采集时间使用。

■ 单格数据点数



图 5.8-35 单格数据点数设置

单格数据点数由用户直接在文本框输入。单格数据点数越多，绘图精度越高。每格采集点数取值范围为 1 到 100。



- 采集周期必须为整数，即 1ms 的整数倍。当单格数据点数无法被时间基准整除时，示波器会对数据点数自动纠正。

3. 示波组态与配置

示波器组态与配置按钮位于示波器显示控制组下方，如图 5.8-36 所示。

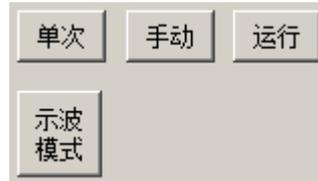


图 5.8-36 示波组态与配置

■ 单次/重复模式



图 5.8-37 模式设置按钮

单击此按钮，在**单次**模式和**重复**模式之间切换。

单次模式下，示波器在触发后开始运行，采集一屏数据后停止。**重复**模式下，示波器每采集完一屏，等再次触发后继续采集。**重复**模式下，示波器一直工作，直到用户手动单击**停止**按钮后才停止。

■ 触发模式



图 5.8-38 触发模式按钮

单击此按钮，在**手动**模式和**程序**模式之间切换。

手动模式下，控制器一直处于触发状态，按下**运行**按钮后立即采集数据。**程序**模式下，示波器在按下**运行**按钮后开始运行，但控制器要等到 IEC 程序执行完触发指令 **HMC_TriggerScope** 才开始采集数据。

■ 触发按钮



图 5.8-39 触发按钮

按下此按钮后，示波器开始运行，按钮变成**停止**。如果示波器处于单次模式，示波器采集完一屏后自动停止。示波器运行时，随时可以按下**停止**按钮停止示波。

■ 显示模式



图 5.8-40 显示模式设置按钮

单击此按钮，在**示波模式**（X/T）和**轨迹模式**（X/Y）之间切换。

示波器默认工作在**示波模式**下，横轴表示时间，纵轴表示采集的数据，用户可以观察到轴参数随时间的变化情况。**轨迹模式**下，用户可观察到一个示波通道数据相对另一个的变换情况。



- 轨迹模式下，前两个通道为一组，后两个通道为一组，不支持交叉使用。

4. 示波器数据查询

示波器数据查询控制按钮位于示波器控制界面底部，如图 5.8-41 图所示。



图 5.8-41 示波器数据查询

■ 示波器游标



图 5.8-42 游标显示设置

勾选一个单选框，则显示对应通道的游标，否则不显示。示波器的游标数据显示在示波绘图界面顶部。

■ 示波关键数据

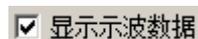


图 5.8-43 示波数据显示设置

勾选此单选框，则在示波器绘图界面底部显示每个波形（轨迹）的最大值和最小值，以及示波器游标当前位置。

■ 示波数据自动保存

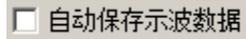


图 5.8-44 自动保存设置

勾选此单选框，示波器每采集完一屏数据后，自动保存示波数据到工程目录下。

5. 示波器数据导入导出

此功能集成在示波器绘图界面的右键菜单中，如图 5.8-45 所示。

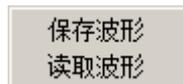


图 5.8-45 示波器数据导入导出

保存波形，将当前示波界面显示的波形（轨迹）导出到文件中。读取波形，示波器从磁盘文件中读取波形（轨迹）数据进行绘制。

5.8.5.2 Table 表绘图设置

Table 表绘图设置包括指定关联的 Table 表，选择关联的行/列等，如图 5.8-46 所示。



图 5.8-46 Table 表绘图设置

■ 关联 Table 表

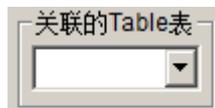


图 5.8-47 关联的 Table 表的选择

列表框中显示所有在工程中定义的 Table 表，关联的 Table 表从此列表框中选择。

■ 关联行/列



图 5.8-48 选择关联 Table 表的行或列

指定关联 Table 表的行或者列，绘图时按此方式取数据。

■ 选择数据来源



图 5.8-49 示波通道控制组

示波器 4 个通道均可用于 Table 表绘图。以红色通道 0 为列，从图中选中的列表框中选择序号。如果上一步指定关联行，此处的是选择行号；如果上一步指定关联列，则此处选择的是列号。

■ 刷新视图



图 5.8-50 刷新视图按钮

按下此按钮，绘制波形。

■ 示波器状态字

示波状态字是 MC 示波器内部定义的变量，与控制器通信时使用，用于获取波形数据。新建工程时系统会以全局变量的形式对示波状态字进行预定义。

5.8.6 下传用户文件

该功能在离线状态下可用，用于备份用户文件到控制器中。



- 菜单栏：单击【工具】—【下传用户文件】。



图 5.8-51 下传用户文件界面

在**路径**下拉框中选择文件在控制器中的存放路径，单击按钮 **..** 从本地路径下选择需要备份到控制器的文件，单击**确定**，开始传输文件，下传用户文件过程如图 5.8-52 所示。



图 5.8-52 下传用户文件过程



- LE 不支持此功能。

5.8.7 辅助工具



- LE 不支持辅助工具功能。

5.8.7.1 工程目标文件加密

此功能可以进行工程目标文件加密。

加密通过和控制器中的唯一 SN（或者 ID）进行关联。后续在下装或者传输.key 文件到控制器时，加密的原 SN(或者 ID)和自身的一样，则进行解密、装载程序等操作；如果 SN(或者 ID)不一致，则控制器拒绝解析和加载此文件。

通过该功能实现对 .at 文件进行加密，加密方式分为单独加密和批量加密两种方式。



- 菜单栏：单击【工具】—【辅助工具】—【1 工程目标文件加密】。

1. 单独加密

可对当前控制器的 SN 号和 .at 文件关联，加密后的工程目标文件只能下装或传输到此 SN 号对应的控制器中。

打开加密工具，对话框如图 5.8-53 所示。



图 5.8-53 单独加密

在 IP 和端口正确的情况下，单击**读取**按钮获取当前控制器的 SN 号，或直接输入控制器的 SN 号。单击**打开**选择对应的 .at 文件路径，单击**加密**，对该 .at 文件进行加密操作。加密完成后弹出生成文件成功的提示框。加密后的 .at 文件名按照“原文件名_sn 号.key”命名，并放置到 .at 文件所在目录的 KEY 文件夹下。

2. 批量加密

可对多个控制器的 SN 号与 .at 文件绑定，加密后生成多个 .key 文件。

用户需要创建一个 .csv 文件。新建一个 .txt 文件，在文档中输入需要批量加密的控制器 SN 号，SN 号之间用英文逗号隔开（如 1,2,3）或直接换行均可。保存后将 .txt 的后缀名改为 .csv，则包含有批量 SN 信息的 .csv 文件生成。

单击【批量】标签页，打开加密窗口如图 5.8-54 所示。



图 5.8-54 批量加密

单击 SN 后的**打开**，选择要加载的.csv 文件。选择完成后，单击**加密**按钮，对.at 文件按照批量的.sn 文件进行加密。加密完成后提示“生成加密文件成功”。加密后的.at 文件命名方式与单独加密文件一致，在 KEY 文件下生成多个 SN 对应的.key 文件。用户可通过【工程升级】工具将加密的.key 文件传输到对应 SN 号的控制器中。控制器 SN 号与.key 文件的 SN 号不一致会导致传输失败。



- 重复为同一工程的.at 文件（工程名与 SN 号均相同）进行加密时，单独加密会出现替换提示，批量加密将不出现提示而直接进行替换操作。

5.8.7.2 控制器操作

控制器操作包含【控制器 IP】、【控制器信息】、【控制器锁】、【网络配置】、【工程升级】、【固件升级】、【日志读取】、【IP 修改（LE5405）】共八个标签页。



- 在各标签页进行相关操作前，请先在【控制器 IP】标签页中输入当前控制器的 IP 地址，连接到控制器。否则在各标签页进行操作时会提示设置控制器 IP。

1. 控制器 IP

在对控制器进行相应操作之前，需要连接控制器。“控制器 IP”设置界面如图 5.8-55 所示。

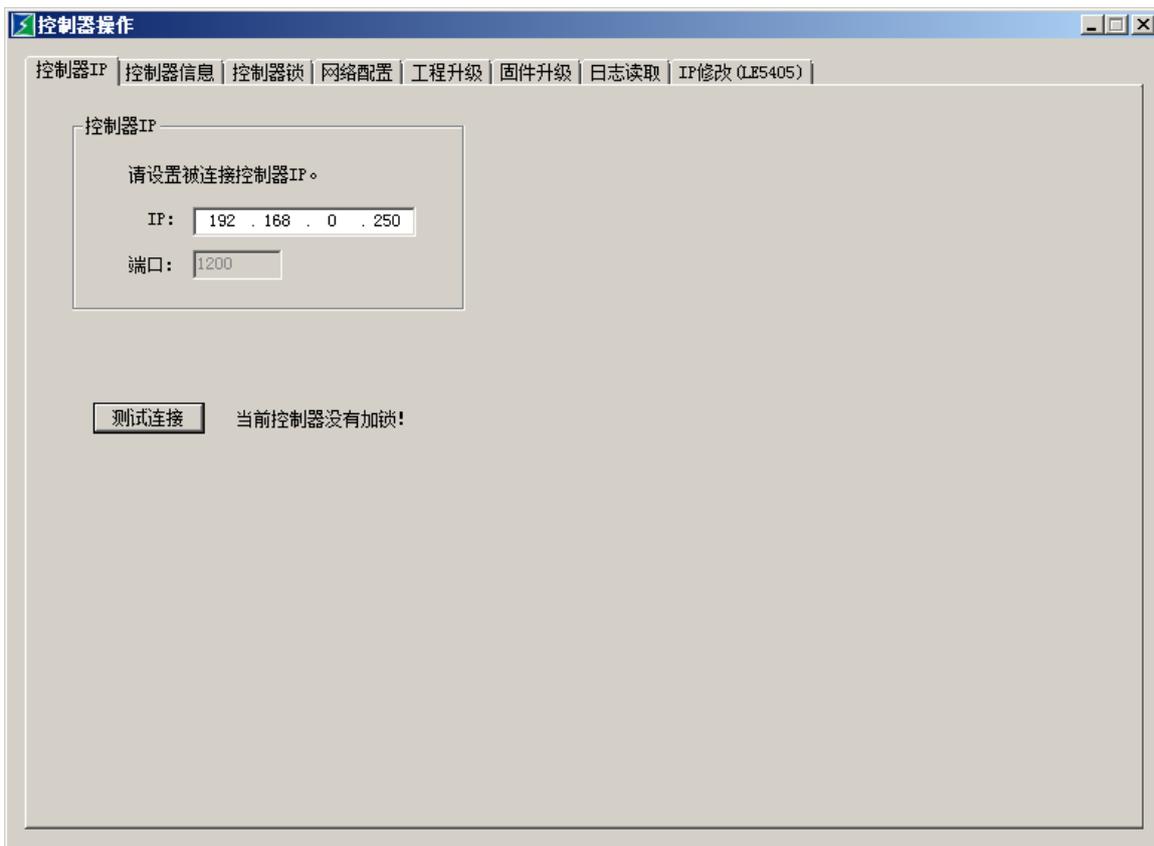


图 5.8-55 控制器 IP 标签页

在 IP 地址编辑框中输入当前控制器的 IP 地址，单击**测试连接**按钮，弹出“控制器连接成功!”的提示框，单击**确定**后，显示当前控制器的加锁状态。连接控制器后，则可以在各标签页进行相关操作。

2. 控制器信息

在【**控制器信息**】标签页中可以读取控制器的版本信息，对于 LK 冗余控制器，可以获取当前控制器的冗余状态。

在【**控制器信息**】标签页中，单击**读取信息**按钮，可以获取当前控制器的版本信息，显示在白色窗口中。同时弹出“读取版本信息成功!”的提示框，如图 5.8-56 所示。



图 5.8-56 读取控制器版本信息

对于 LK 冗余控制器，在【本机状态】和【对方机状态】栏中显示控制器冗余状态信息。各状态的显示信息如表 5.8-9 所示。

表 5.8-9 控制器冗余状态信息说明

控制器状态	状态信息	说明
主从机状态	未知态	只有当本机的主从机状态为不合格主机时，需要单击清除按钮进行清除
	初始态	
	硬件 Ready 态	
	双机 Ready 态	
	单机 Ready 态	
	故障态	
	错误态	
	工程冗余	
	工程验证	
	从机	
主机		
不合格主机		

控制器状态	状态信息	说明
单双机状态	未知态 单机 双机	若【本机状态】为单机， 则【对方机状态】为 unknown
AB 机状态	未知态 A 机 B 机	显示当前系为 A 机还是 B 机
冗余工作网状态	未知态 第一路光纤 第二路光纤	此状态值显示当前工作 的是 A 网还是 B 网
工程运行状态	未知态 正在运行 已停止 挂起	显示工程的运行状态
钥匙开关状态	UNKNOWN RUN REMOTE PRG	显示当前 A、B 机的钥匙 开关状态

3. 控制器锁

在【控制器锁】标签页中，实现对当前控制器的加解锁操作。

通过对控制器进行加锁操作，离线状态下禁止对控制器进行相关操作，包括升级控制器、下传用户文件、下装、上传工程文件、下载工程文件、监视、清空控制器。

控制器锁窗口如图 5.8-57 所示。

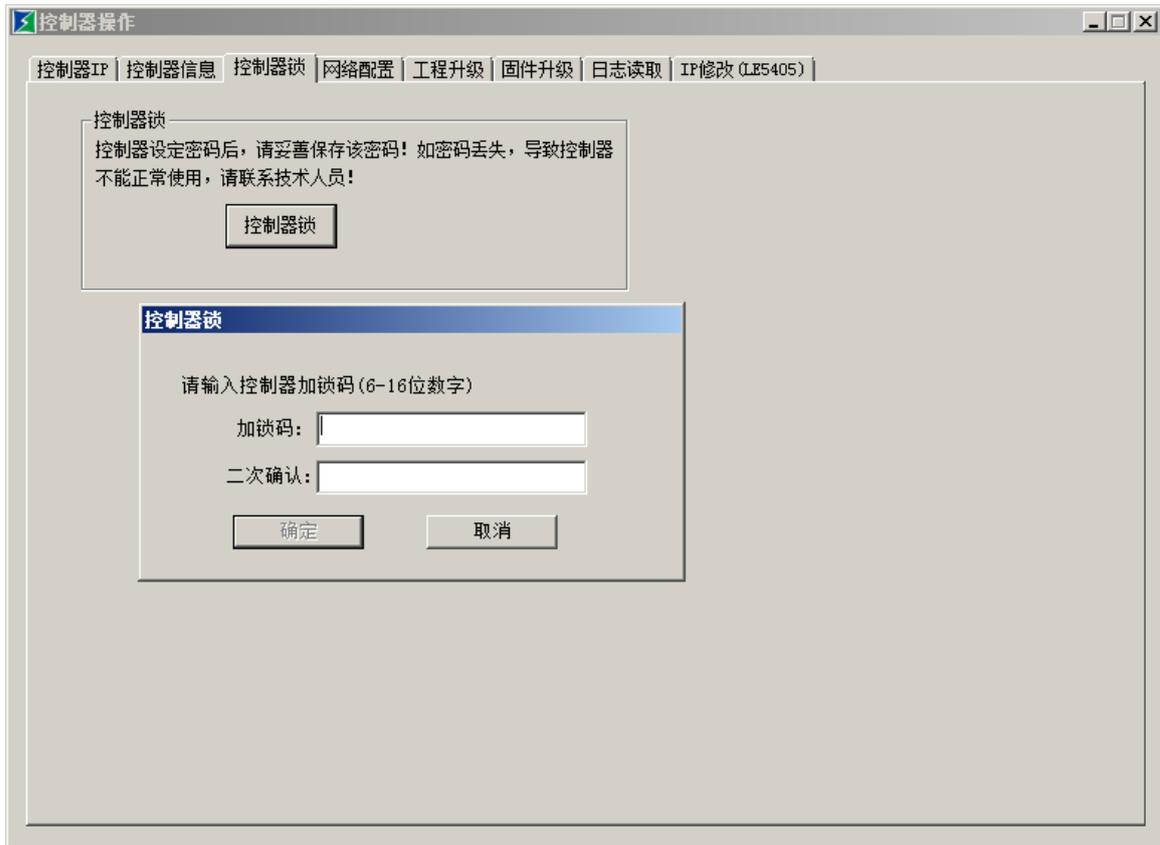


图 5.8-57 控制器加锁

单击**控制器锁**按钮，弹出控制器加锁密码输入对话框。输入两次相同的 6-16 位数字**加锁码**后，单击**确定**，完成对控制器加锁操作，并提示控制器加锁成功。

加锁后对控制器进行相关操作时，操作无法完成，并在信息栏出现“当前控制器已加锁，请解锁后重试”。

加锁后，再次单击**控制器锁**按钮可进行解锁。解锁对话框如图 5.8-58 所示。输入之前设置的**加锁码**，单击**确定**，完成控制器解锁操作，并提示控制器解锁成功。

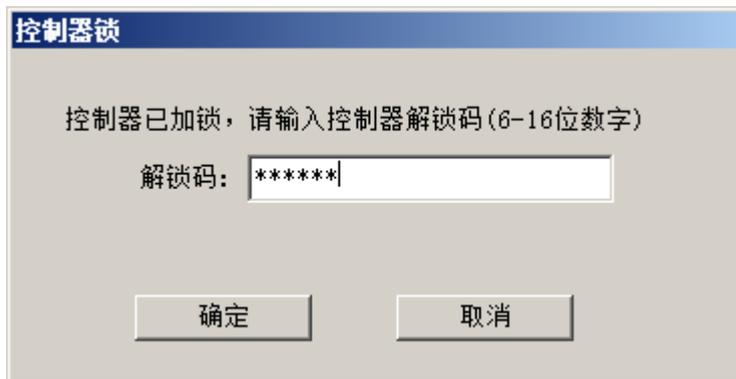


图 5.8-58 控制器解锁

4. 网络配置

此标签页中可以读取或修改 LK CPU 模块的两个以太网口的 IP 地址以及路由信息。

【网络配置】标签页如图 5.8-59 所示。

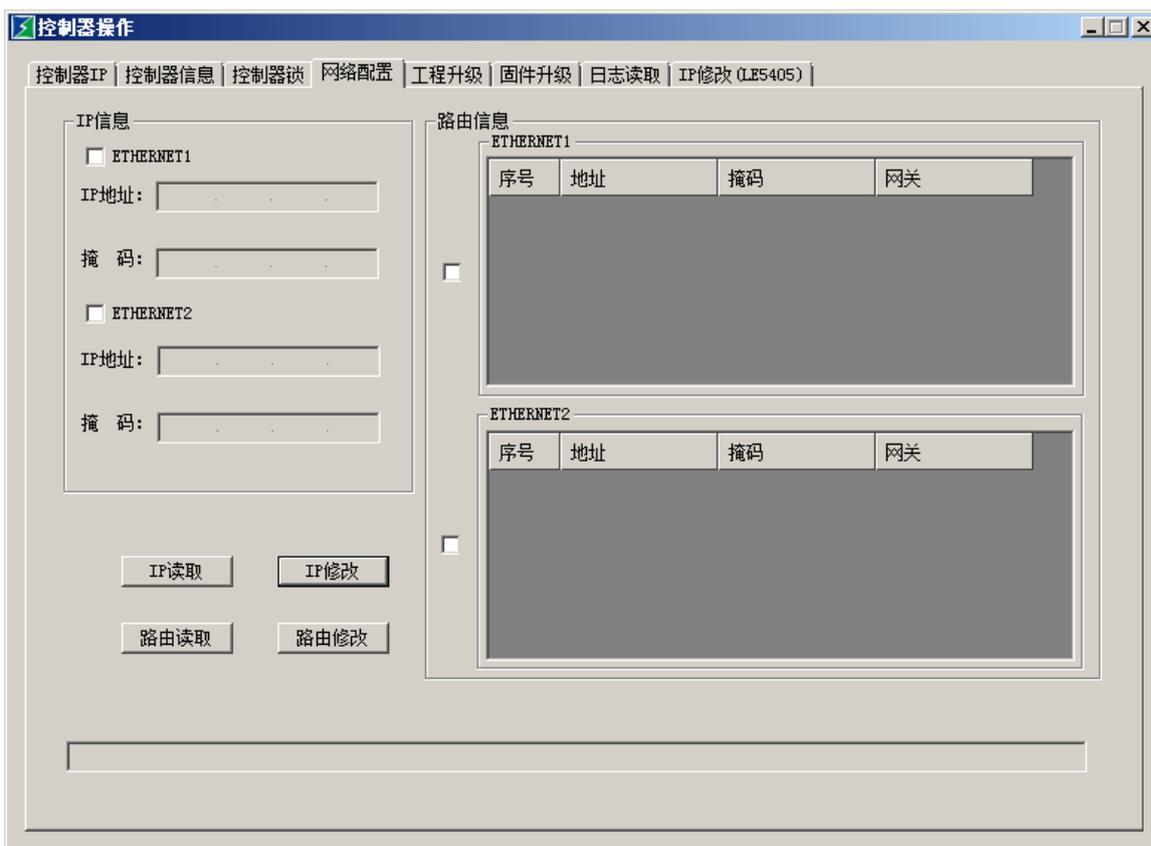


图 5.8-59 网络配置对话框

(1) IP 读取与修改

单击 **IP 读取** 按钮，可以读取两个以太网口的 IP 和掩码，并在【IP 信息】中显示。修改 IP 地址时，可勾选需要修改的以太网口，输入新的 IP 地址和掩码，单击 **IP 修改** 按钮后，即可将新的 IP 地址写入到控制器。

(2) 配置路由表

勾选需要配置的以太网口，通过路由表区域右键菜单中的**添加**按钮，添加路由信息，如图 5.8-60 所示。



图 5.8-60 显示了一个名为“添加路由”的对话框。对话框顶部有一个标题栏，包含“添加路由”和关闭按钮。对话框主体包含三个输入框，分别用于输入“地址”、“掩码”和“网关”，每个输入框内都有预置的占位符点。在输入框下方有两个按钮，分别是“确定”和“取消”。

图 5.8-60 添加路由

输入**地址**、**掩码**、**网关**后，单击**确定**按钮，在路由表区域生成路由信息，如图 5.8-61 所示。单击**路由修改**按钮，将修改后的路由信息写入控制器。

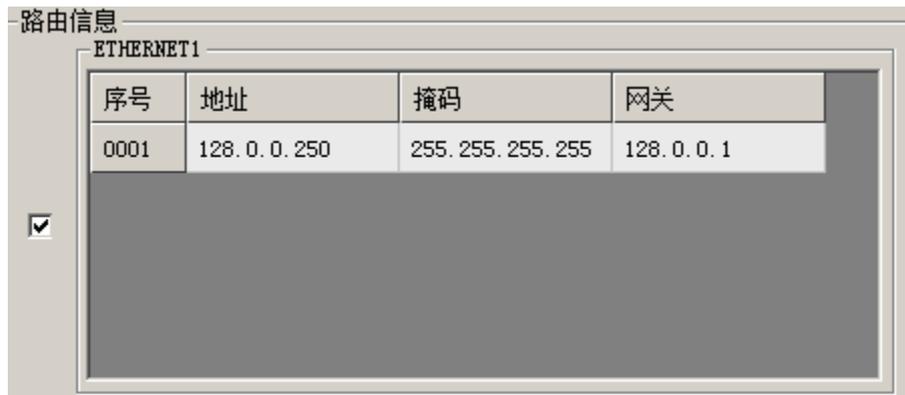


图 5.8-61 显示了一个名为“路由信息”的窗口。窗口顶部有一个复选框，下方是一个表格。表格标题为“ETHERNET 1”。表格包含以下数据：

序号	地址	掩码	网关
0001	128.0.0.250	255.255.255.255	128.0.0.1

窗口底部有一个复选框，处于勾选状态。

图 5.8-61 添加的路由信息

(3) 路由读取与修改

通过**路由读取**按钮，可以获取两个以太网口的路由信息，包含 IP 地址、掩码和网关。修改路由信息时，勾选需要修改的以太网口，输入新的路由信息，单击**路由修改**按钮，完成路由修改设置。



- 仅 LK 支持此工具。
- 路由设置规则同 PC 机路由设置一致。
- 新设置的 IP 信息或路由信息必须重启控制器才能生效。

5. 工程升级

此标签页实现传输单独的.at 或.key 工程文件到控制器，对下装到控制器的工程进行备份。



- LK 控制器仅支持.at 文件，MC 控制器对.at 与.key 文件都支持。

(1) 升级工程

单击**路径**栏的按钮，选择需要上传的.at 文件（AutoThink 编译生成的逻辑文件）或.key 文件（工程文件加密工具生成的加密文件），单击**升级**，弹出升级确认提示框，单击**是**，将该.at 或.key 文件上传至控制器当中。如图 5.8-62 所示。

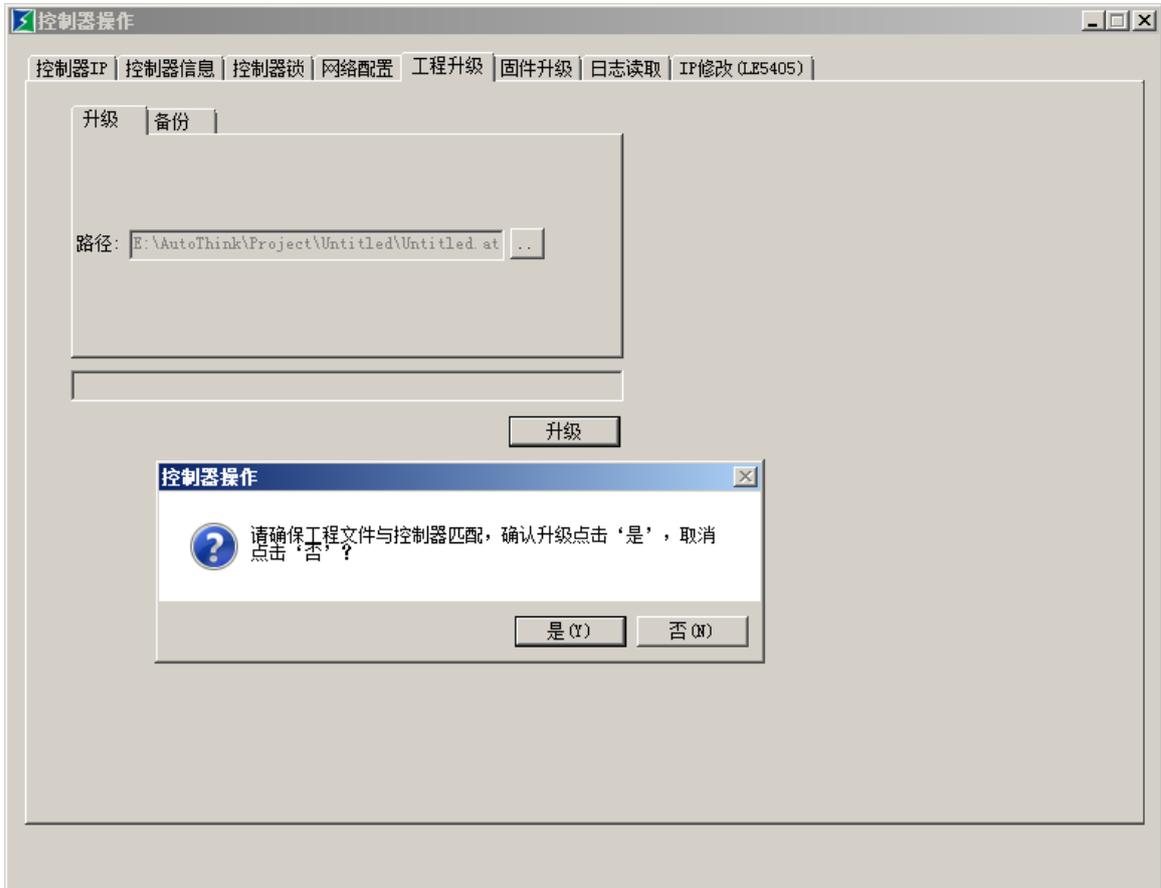


图 5.8-62 上传工程文件

传输完成时会弹出“文件传输成功”的提示框，如图 5.8-63 所示。单击**确定**，关闭提示框。



图 5.8-63 文件传输成功

(2) 备份工程

在【备份】标签页中，可将下装到控制器的工程备份到本地路径下。单击**路径**栏的按钮 **...**，选择路径，单击**备份**按钮进行备份。备份完成后，弹出“文件备份成功”的提示框。如图 5.8-64 所示。

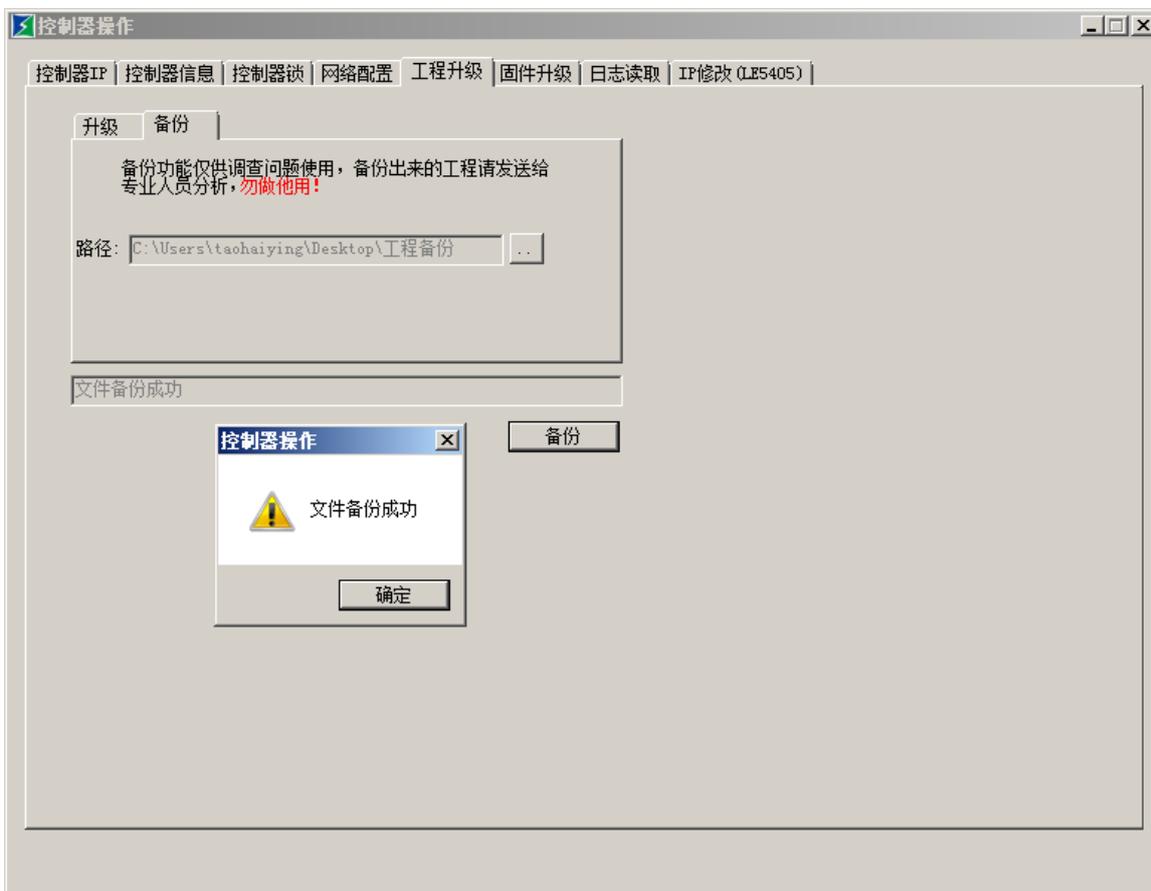


图 5.8-64 工程文件备份

6. 固件升级

此标签页实现控制器固件升级和固件文件的备份。

(1) 控制器固件升级

在【升级】标签页中，单击**路径**栏的按钮`...`，选择控制器固件.bin 文件后，单击**升级**按钮，弹出升级确认提示框，单击**是**，固件文件被上传到控制器中。

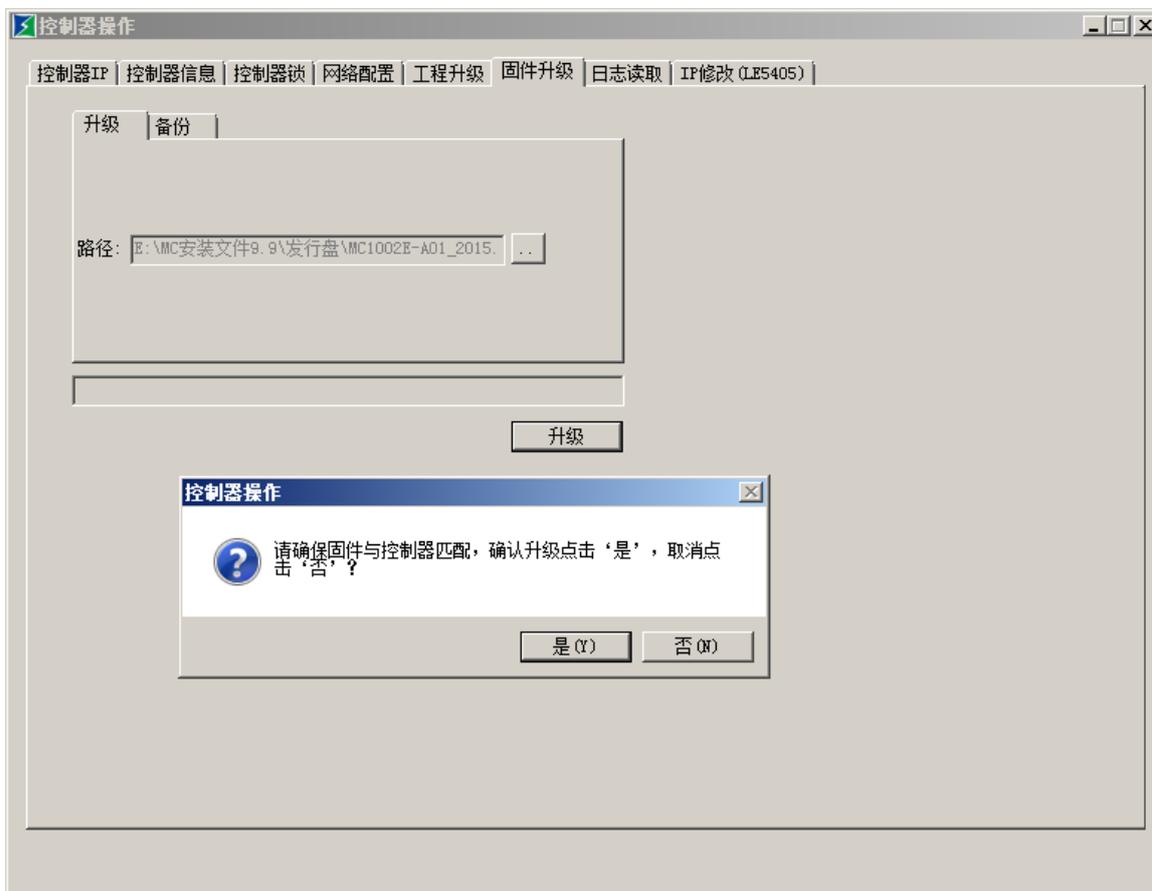


图 5.8-65 固件升级对话框

控制器接收到固件文件后开始升级，此时会弹出如图 5.8-66 所示的提示框。单击**确定**，等待控制器完成升级即可。



图 5.8-66 固件文件传输提示框

(2) 备份固件文件

在【备份】标签页中，可将控制器中的固件文件备份到本地路径下。用于专业人员分析问题使用。单击**路径**栏的按钮 ，选择备份路径，单击**备份**按钮进行备份。



- MC 不支持固件备份功能。

7. 日志读取

通过此标签页，可以读取控制器中的日志文件，保存到本地路径下，供专业人员分析问题用。

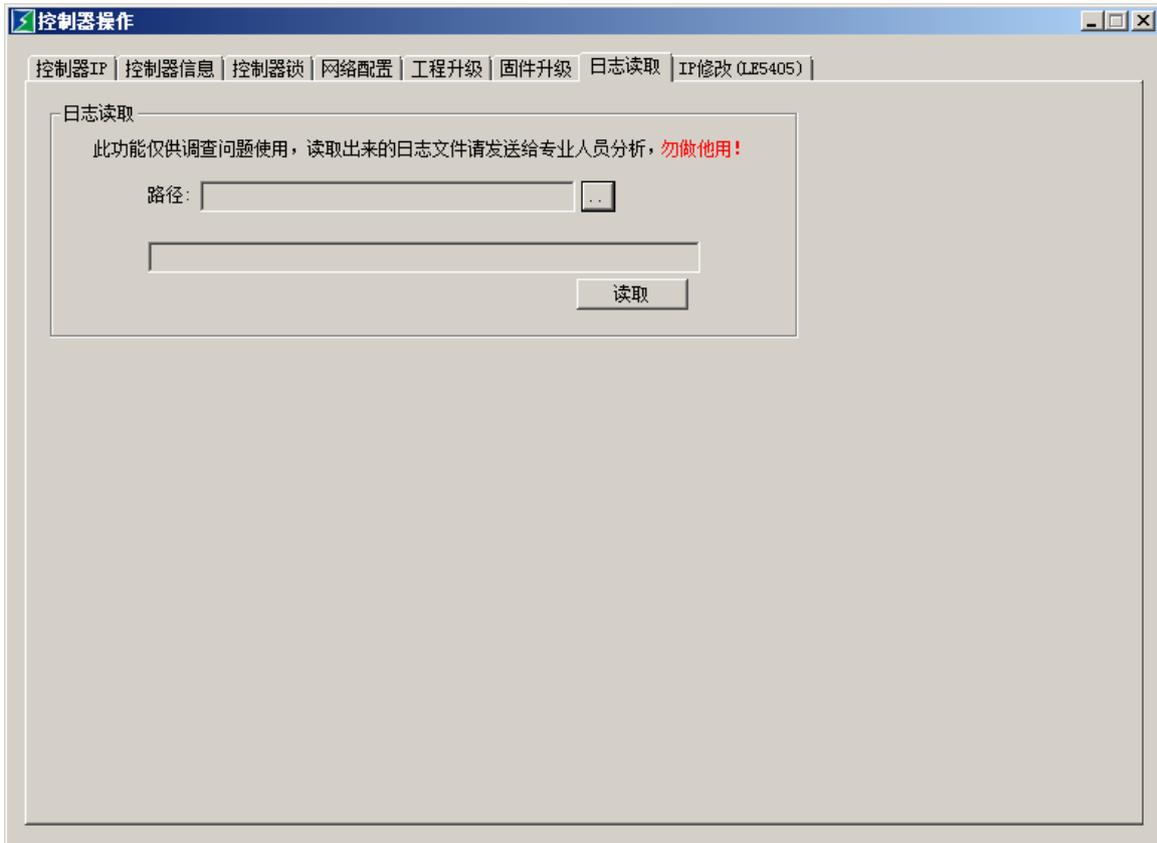


图 5.8-67 日志读取对话框

单击**路径**栏的按钮，选择备份路径，单击**读取**按钮，控制器日志文件被下载到本地备份路径下，读取成功时会弹出提示框。

8. IP 修改 (LE5405)

通过该标签页可以修改 LE5405 的 IP 地址。首先在【控制器 IP】标签页中，设置 LE5405 的 IP 地址，并连接成功。在【IP 修改 (LE5405)】标签页中单击**IP 读取**，LE5405 模块的 IP 地址显示在**IP 地址**框中。IP 地址第四字段的百位值由下拉框中选择的数字决定，可选值 0、1、2；十位和个位分别由 LE5405 模块上两个拨码开关位置 0~9 组成，读取后显示在末尾的地址框中，不可进行修改。第四字段 IP=设置的下拉框数字×100+末尾框中的拨码开关设置值。

如 LE5405 模块的 IP 地址为 192.168.0.98，读取 IP 信息如图 5.8-68 所示。

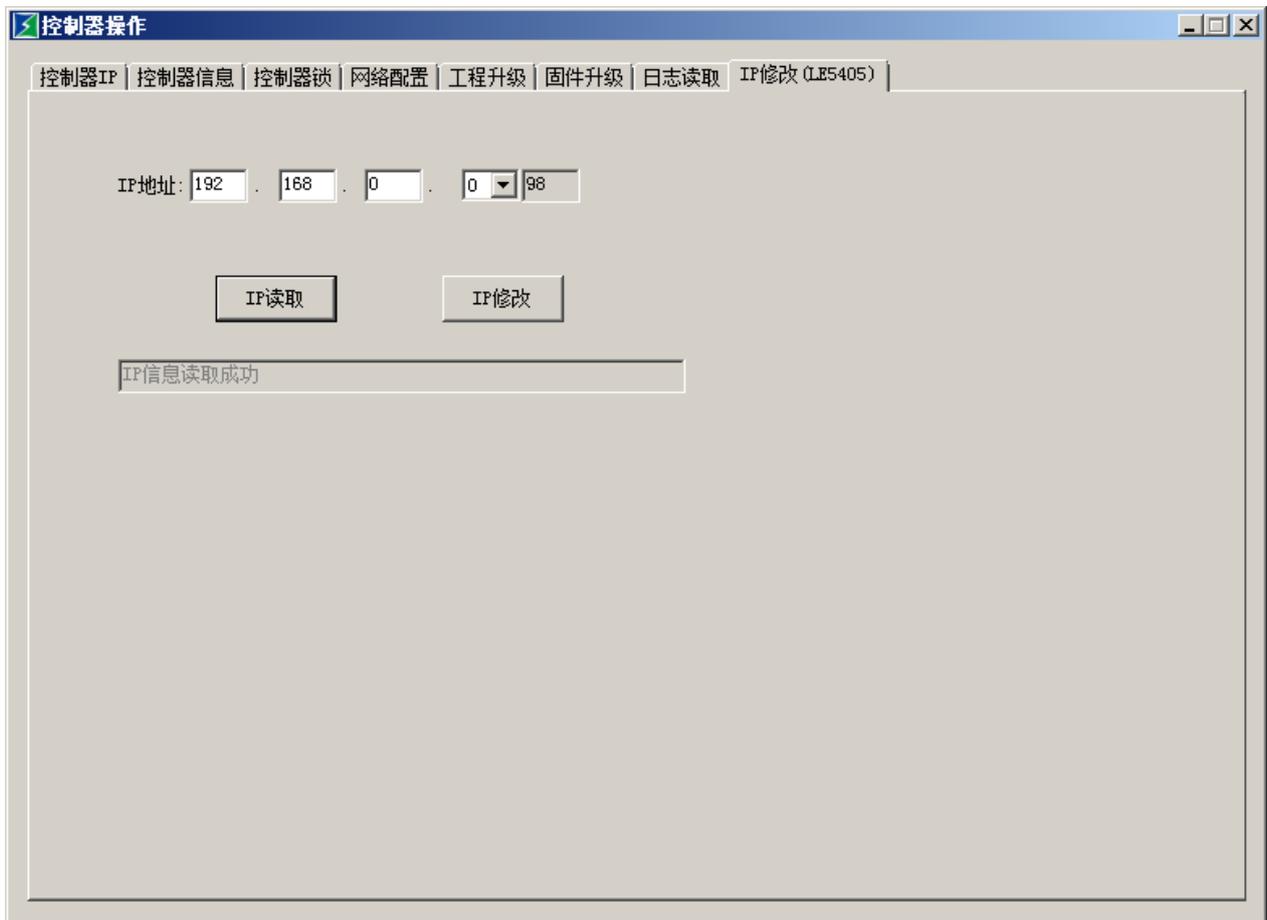


图 5.8-68 读取 LE5405 IP 成功

用户可以修改 IP 地址的前三个字段和第四个字段的百位，当末尾地址框中的值（拨码开关值）> 54 时，下拉框中的值只能设置为 1 或 0。

通过拨码开关设置第四字段地址时，不能设置为表 5.8-10 所示的无效地址值。

表 5.8-10 无效地址值

IP 地址	第四字段非法地址	含义	备注
x.x.x.0	0	保留地址	通过拨码开关修改 IP 地址的第四字段后，导致第四字段地址为 0、1 或者大于等于 255，则视为无效 IP 设置，IP 修改失败，此时 IP 值的最后一个字段变为 99，前三字段地址不变
x.x.x.1	1	网关地址	
x.x.x.255	255	广播地址	
x.x.x.x99	x99	IP 复位	拨码开关设为 99，则复位 LE5405 的 IP 为 192.168.0.99

IP 地址	第四字段 非法地址	含义	备注
			IP 地址 x.x.x.199 为非法地址，不可设置

输入要修改的 IP 地址后，单击 **IP 修改**，IP 地址修改成功，如图 5.8-69 所示。

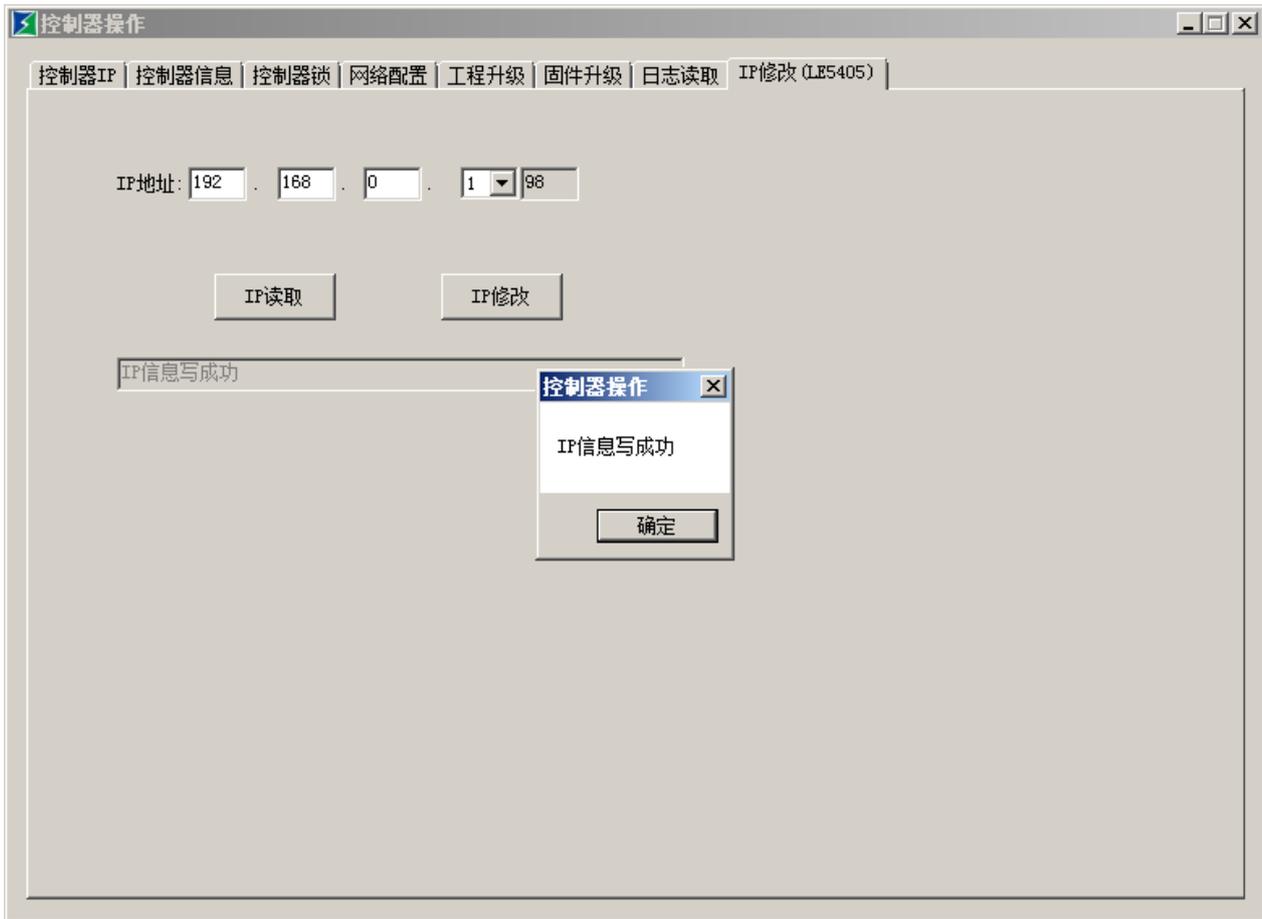


图 5.8-69 修改 LE5405 IP 成功

第6章 库管理

库管理器窗口关联 POU 编辑窗口,以库管理器窗口和 POU 编辑窗口阐述它们之间的关联关系。

1. 关联加载

当首次打开 POU 编辑窗口时,系统自动加载库管理窗口,如图 6-1 所示。

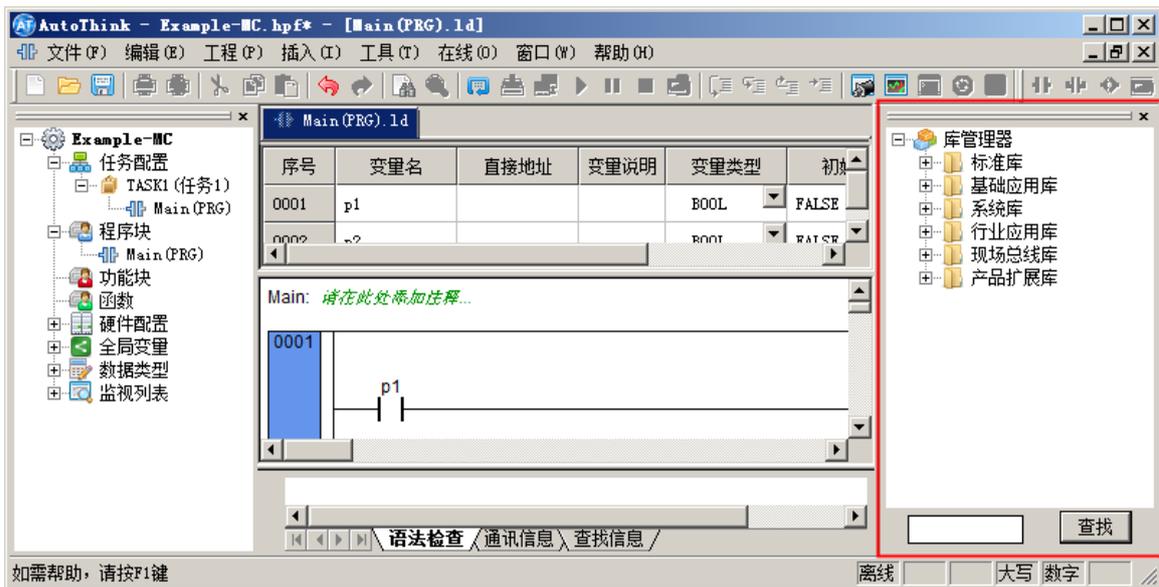


图 6-1 POU 编辑窗口关联库管理器窗口

如果打开工程管理树上其他节点的编辑窗口,例如全局变量编辑窗口,则 POU 编辑窗口和库管理器窗口自动切换为非活动状态,隐藏到全局变量编辑窗口后。如图 6-2 所示。



图 6-2 全局变量编辑窗口

2. 记忆关联关系

上述操作后，单击 POU 编辑窗口标签页，可以切换回 POU 编辑窗口显示，如图 6-1 所示，同时显示 POU 编辑窗口和库管理器窗口。

但是，如果在切换之前，库管理器窗口已经被手动关闭（单击库管理器窗口右上角的关闭按钮），或者通过【窗口】菜单—【库管理窗口】命令关闭，那么，再次切换回 POU 编辑窗口时将不再自动加载库管理器窗口。即库管理器窗口有记忆状态功能，在窗口切换操作前后，对前一次 POU 编辑窗口是否关联库管理器窗口的关系进行记忆，如果之前无关联关系，那么即使切换回 POU 编辑窗口，也不会自动加载库管理器窗口。

此时，可以通过菜单命令如图 6-3 所示，手动加载库管理器窗口。

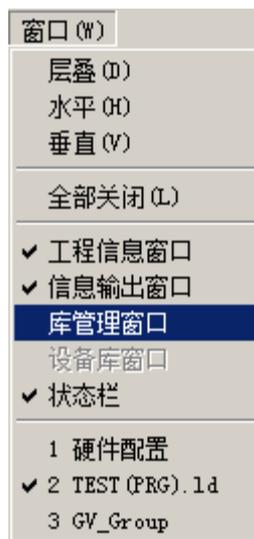


图 6-3 手动加载库管理器窗口

设备库窗口与硬件配置窗口之间的关系也是如此，不再赘述。

6.2 库管理器

库管理器窗口如图 6.2-1 所示。该窗口采用树状结构管理方式，以【库管理器】为根节点，下面包含相关的库文件目录，文件目录下包含相关的库文件。

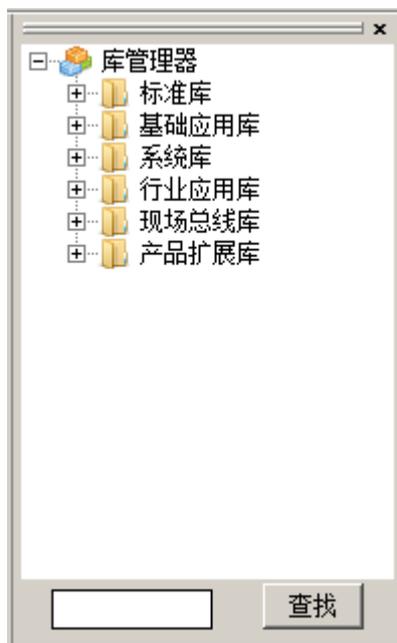


图 6.2-1 库管理器窗口



- 不同的目标平台（配置文件）的工程，库管理器下包含的库文件目录会不相同，以及加载的可用指令库也不相同。
- 现场总线库只有 MC1002E 和 MC1002R 控制器支持。

6.2.1 查找

库管理器提供查找功能，在窗口底端的查找输入框中输入查找的条件，搜索条件不区分字母大小写，然后单击**查找**按钮进行检索，搜索完成后，在库列表下会显示符合搜索条件的指令类型，双击搜索到的指令将定位到指令列表。如图 6.2-2 所示。



图 6.2-2 “指令库”窗口的搜索功能

6.2.2 库配置

所有在库管理器中显示的库文件，需要在“库配置”窗口中使能后，才可以正常使用。否则，使用时会编译报错。



- 库管理器：右击空白区域，单击【库配置】。

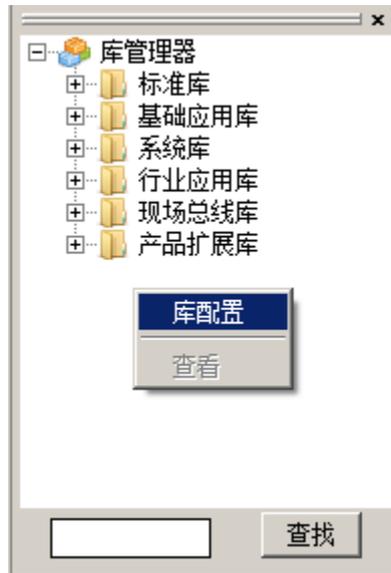


图 6.2-3 库配置菜单项

弹出“库配置”对话框，如图 6.2-4 所示，勾选需要使能的库文件，单击**确定**按钮，该库文件生效。用户可以调用已生效库文件下的指令。



- 新建的工程中，库管理器下缺省显示的库文件已使能，可以正常使用。

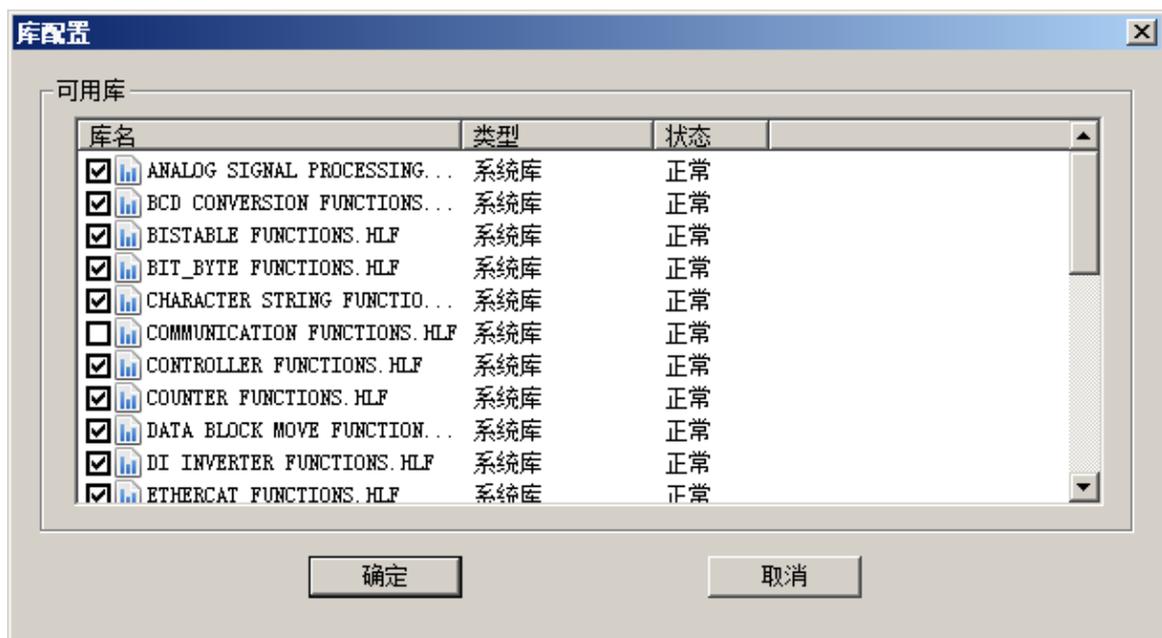


图 6.2-4 “库配置”对话框

6.2.3 删除库

删除库只是让某个库文件不使能，并不是从库管理器中将该库文件删除。在“库配置”对话框中去掉某个库文件前的勾选，单击**确定**按钮，则该库文件不可用。

当删除相互关联的库时，需要遵循一定的规则，若删除的库会影响其他库的编译时，则该库不能被删除，否则可以删除。

举例：如图 6.2-5 所示，倒金字塔库关系图，B、C 库均引用 A 库，D、E 库均引用 B 库，F、G 库均引用 C 库。

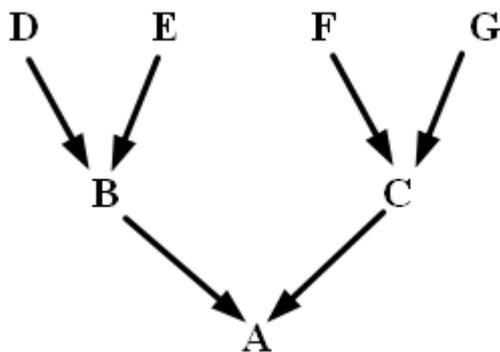


图 6.2-5 关联的库关系图

如果要删除 B 库，则因为 D 库依赖于 B 库，则删除 B 库时会弹出系统提示框“此库文件被‘D.HLF’引用，删除失败！”。删除 D 之后，可以正常删除 B。示例：添加的定时器指令“TIMER FUNCTIONS.HLF”库关联函数操纵器指令“ANALOG SIGNAL PROCESSING FUNCTIONS.HLF”库文件，如果删除定时器指令“TIMER FUNCTIONS.HLF”，则提示如图 6.2-6 所示。

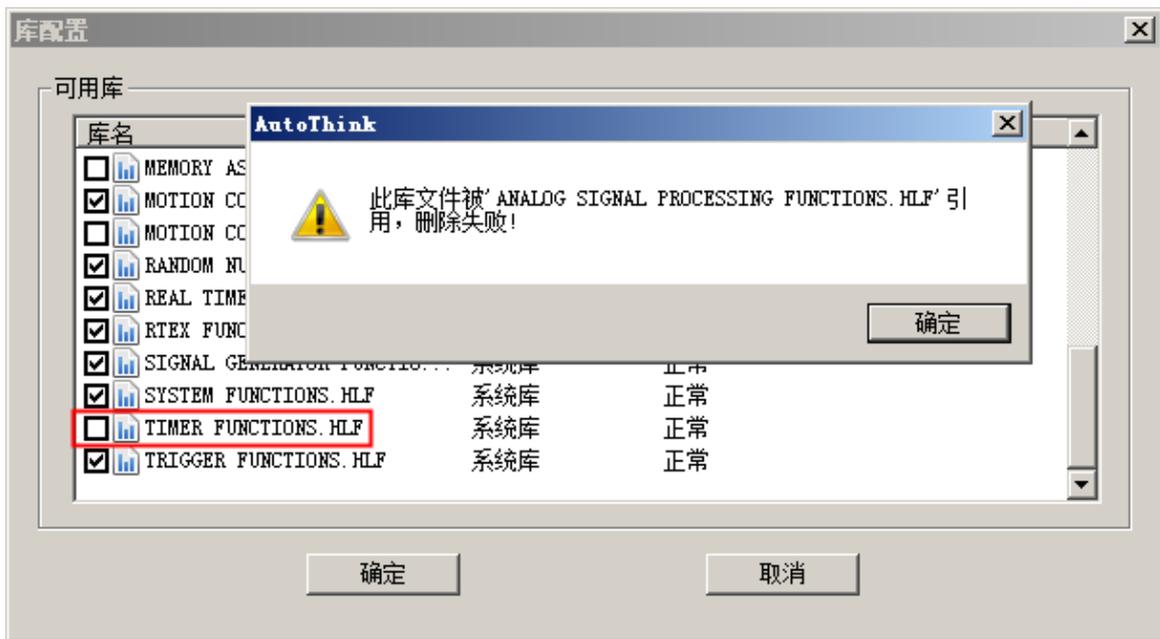


图 6.2-6 系统提示框-不能删除具有被引用关系的库



- 删除库是将已经添加的库的链接关系删除，并不是将库文件从 Library 文件夹中删除。
- 所有非库管理器中默认的库文件，在“库配置”对话框中勾选，则添加到库管理器中。去掉勾选则从库管理器下删除。

6.2.4 查看

通过此命令可以查看库指令的信息。



- 库管理器：右击指令名称，单击【查看】。

在左侧的编辑区域显示该库指令的信息窗口：名称（函数类型）、注释、输入变量、输出变量、输入输出变量、局部变量、图例等信息。如图 6.2-7 所示。

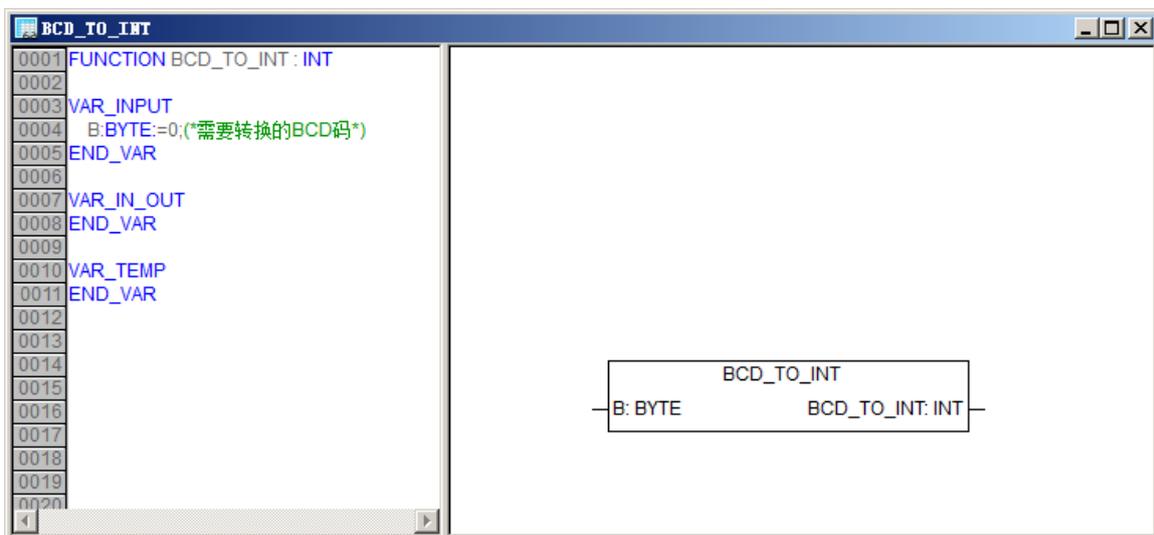


图 6.2-7 查看功能块或函数信息

6.2.5 库的更新

6.2.5.1 库检测

打开工程时，AutoThink 软件会自动检测工程中已经引用的库的版本信息是否与缺省路径下存在的库文件版本信息一致，当检测版本信息一致时，不提示更新库；当库的版本信息发生变动时，提示按顺序更新库，如图 6.2-8 所示。



图 6.2-8 系统提示框-库的更新

单击**确定**按钮，关闭提示框。

此时在“库配置”对话框中显示异常的库文件，如图 6.2-9 所示。编译时在信息窗口提示更新库文件。

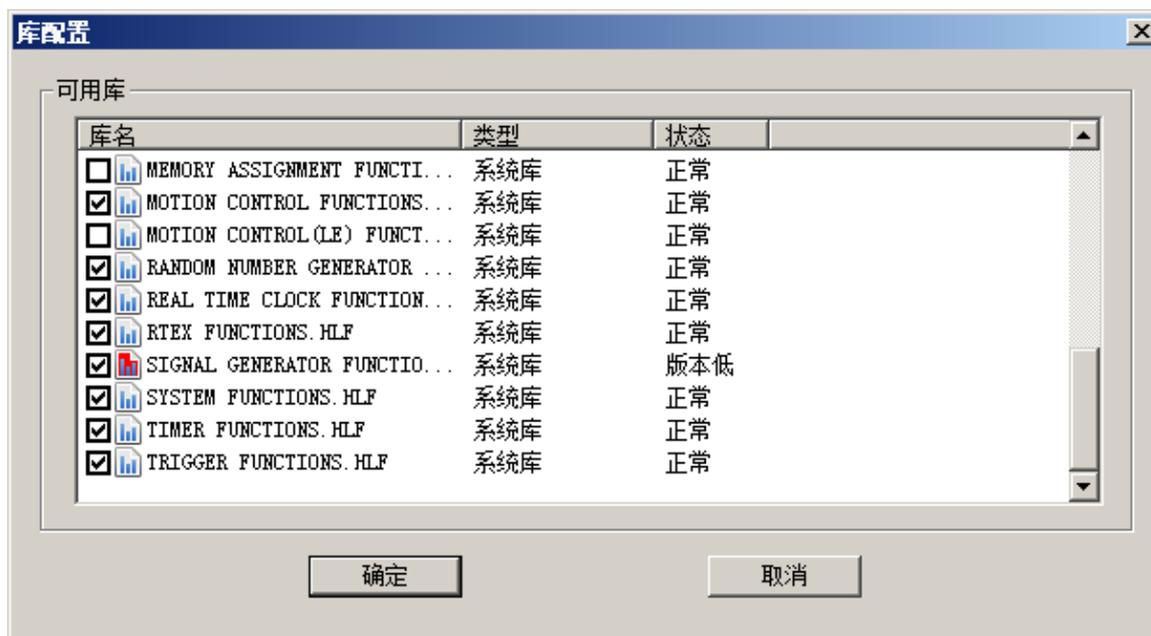


图 6.2-9 库配置对话框异常状态



- 更新库之后，请检查“库配置”窗口中的库状态是否正常，若发现异常状态时请联系专业人员提供帮助。

6.2.5.2 库自动更新

库自动更新功能是指当 AutoThink 软件的.trg 文件被修改，或者软件版本进行了升级，打开旧版本工程时，如果不更新 Library 目录下的库文件，那么这些库文件将不能正常添加到当前工程中，导致无法引用库中的各种指令。此版本软件能够实现对于 Library 目录下库文件的自动更新，无需手动替换、配置。

需要进行库自动更新的三种情况：

当添加新建的库文件时，此时库文件的.trg 文件 ID 号与已编译保存时记录的 ID 号不一致，该库文件将不能正常添加。

当软件进行了版本升级，使用新版本打开旧版工程时，由于两个序列化版本号不一致时，库文件将不能正常添加，无法在工程中使用库。

库文件间有引用时，即出现一个库应用了另外一个库，如果被引用的库发生变化时，引用库也需做自动更新。

下面以库之间存在引用关系的情形为例，介绍使用库自动更新的方法。

当库之间存在引用关系，被引用库发生变化时，引用它的库将无法正常使用，这时软件会弹出提示框，建议使用自动更新。

第1步 新建库 test1，并在库中添加一个功能块 POU: Id_fb，如图 6.2-10 和图 6.2-11 所示。将该库编译保存。



图 6.2-10 新建库，库名 test1

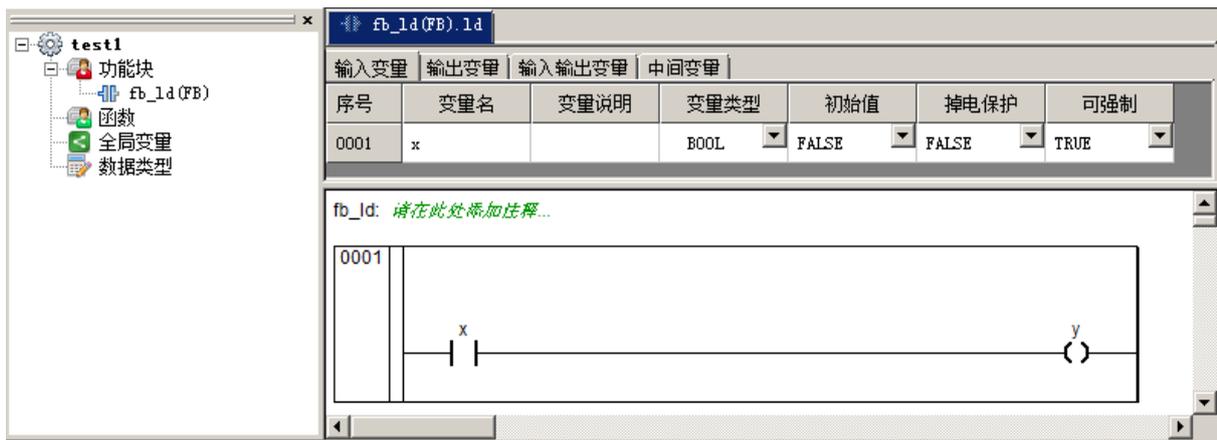


图 6.2-11 在新建的库中添加功能块

第2步 再新建一个库 test2，并添加对库 test1 的引用，如图 6.2-12 所示。

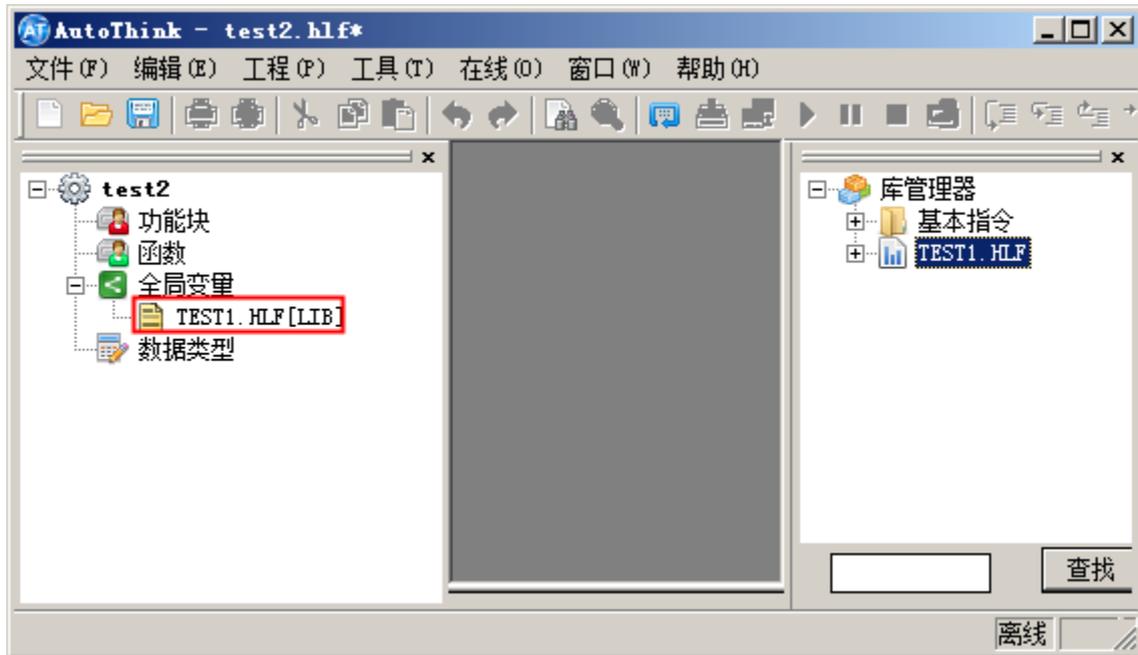


图 6.2-12 添加库 TEST1 完成

第3步 在工程中添加这两个库文件，如图 6.2-13 所示。

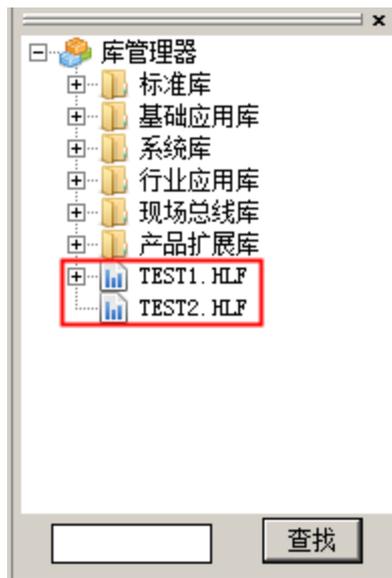


图 6.2-13 添加两个自定义的库

第4步 现在删除库 test1 中 POU 的线圈，则引用它的库 test2 将受到影响，再打开工程时，出现提示更新的对话框，如图 6.2-14 所示。



图 6.2-14 需要库更新提示对话框

打开工程后，库管理器下的库 TEST2 显示为红色，异常状态。编译后，提示“请以以下顺序更新库：TEST2.HLF”。如图 6.2-15 所示。

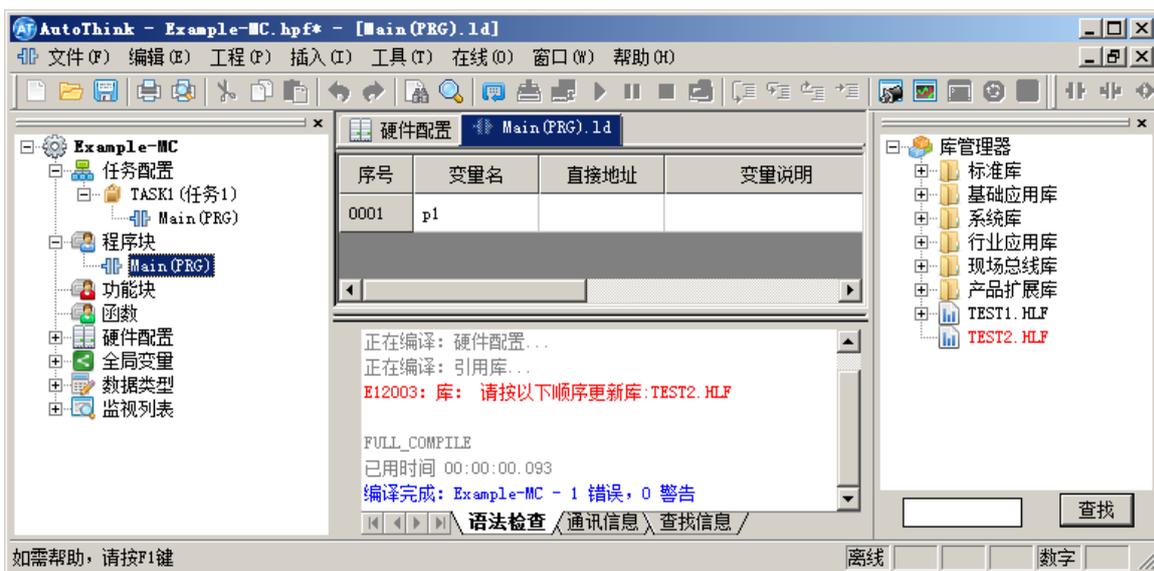


图 6.2-15 库变化导致的编译错误

第5步 如图 6.2-16 所示，选择菜单项【工程】—【库自动更新】命令，在弹出的提示框单击**确定**，系统自动进行更新。



图 6.2-16 选择库自动更新

更新时会弹出提示框，如图 6.2-17 所示。



图 6.2-17 库自动更新提示框

单击**确定**，库管理器中的该库文件显示正常，编译通过。如图 6.2-18 所示。

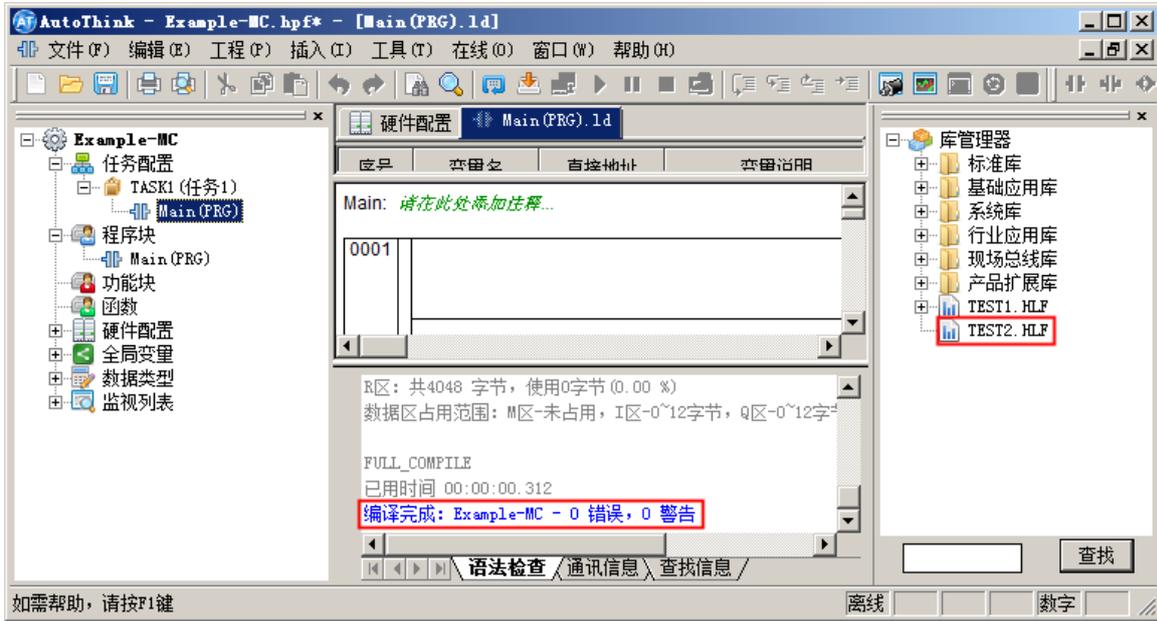


图 6.2-18 更新完成, 显示、编译均正常

第7章 常见问题

7.1 工程升级

当工程调用了自定义库，用户对该工程进行升级之前，需要预先对自定义库进行如下操作。将低版本软件安装路径中的 **Project** 文件夹和相关的目标配置文件夹进行备份。如 **Example-LE** 工程的目标配置是 **V3.1.3B1**，那么对该工程升级时，需要将 **V3.1.3B1** 软件安装路径下的文件夹 **Project** 和 **V3.1.3_2014.03.30** 进行备份。

7.1.1 不切换目标配置

第1步 拷贝自定义库

将引用的自定义库拷入高版本软件安装路径下的相应文件夹中。

按照上面要求，用户需要将自定义库 **Quality.hlf** 从已备份的 **V3.1.3_2014.03.30** 文件夹中复制到高版本软件安装路径 **\AutoThink\V3.1.3_2014.03.30\Library** 下的同名文件夹中。

第2步 打开工程文件

打开 **\AutoThink\Project\Example-LE** 路径下的 **Example-LE.hpf** 后，自定义库（如 **Quality.HLF**）会显示红色，标识版本异常。

第3步 库版本升级

需要通过【工程】—【库自动更新】进行升级。

7.1.2 切换目标配置

例如，切换到 **V3.1.4B1** 目标配置，具体步骤如下：

第1步 拷贝自定义库

将引用的自定义库拷入高版本软件安装路径下的相应文件夹中。

按照上面要求，用户需要将自定义库 **Quality.hlf** 从已备份的 **V3.1.3_2014.03.30** 文件夹中复制到高版本软件安装路径 **\AutoThink\V3.1.4_2015.02.09\Library** 下的同名文件夹中。

第2步 打开库文件

在打开库文件 **\AutoThink\V3.1.4_2015.02.09\Library\Quality.hlf**，更改库的目标配置为 **V3.1.4B1**，同时对库进行编译和保存。

第3步 修改工程的目标配置

打开\AutoThink\Project\Example-LE 路径下的 Example-LE.hpf 后，更改工程的目标配置为 V3.1.4B1，依次进行编译和保存。



- 控制器型号为 MC1008-A01 的工程不支持上述内容。
- 升级 V3.1.3_2014.03.30 工程后需手动加载 IO 操作指令库，加载方法可参见章节 [6.2.2 库配置](#)中关于添加库的内容。

切换到其它目标配置时，选择安装路径下的相应文件夹进行以上操作。目标配置与安装路径下文件夹的对应关系如表 7.1-1 所示。

表 7.1-1 目标配置与安装路径下文件夹对应关系

目标配置	安装盘下对应文件夹
V3.1.1B1	V3.1.1_2012.5.15
V3.1.1B2	V3.1.1_2012.11.15
V3.1.2B1	V3.1.2_2013.04.30
V3.1.2B2	V3.1.2_2013.09.30
V3.1.3B1	V3.1.3_2014.03.30
V3.1.3B1PATCH1	V3.1.3_2014.08.25
V3.1.4B1	V3.1.4_2015.02.09
V3.1.4B2	V3.1.4_2015.05.18
V3.1.4B3	V3.1.4_2015.09.15
V3.1.4B5	V3.1.4_2016.06.30

索引

- AutoThink, 9
- CFC 编辑器, 178
- I/O 变量, 321
- LD 编辑器, 129
- PID 指令向导, 420
- POU, 87, 88
- POU 的调用, 97
- POU 的操作, 93
- SFC 编辑器, 197
- ST 编辑器, 155
- 工作区域, 61
- 工具栏, 57
- 工程升级, 485
- 工程管理, 57
- 下传用户文件, 453
- 下装, 50, 397
- 中断配置, 102
- 文件夹重命名, 119
- 示波器, 446
- 功能块, 88
- 平台切换, 387
- 写入, 405
- 写入存储卡, 402
- 执行顺序, 193
- 在线, 404
- 在线调试, 410
- 存储格式, 265
- 网络配置, 83
- 任务, 119
- 仿真, 405
- 仿真调试, 413
- 自定义数据类型, 276
- 全下装, 397
- 全部释放, 408
- 全编译, 391
- 创建 LE 工程, 26
- 安装步骤, 14
- 安装环境, 13
- 设置控制器 IP, 404
- 访问变量, 299
- 导入变量, 301
- 导出变量, 303
- 运行, 400
- 批量加密, 455
- 块元件, 131
- 串口通讯指令, 428
- 串口通讯指令向导, 428
- 返回, 149

- 删除文件夹, 119
- 状态栏, 60
- 库自动更新, 479
- 库配置, 474
- 冷复位, 401
- 软件界面, 55
- 变量, 278
- 变量名, 281
- 单独加密, 455
- 函数, 88
- 线圈, 131
- 标准数据类型, 268
- 标题栏, 56
- 查看轴参数, 445
- 轴参数显示设置, 443
- 卸载, 22
- 选项, 76
- 复位, 401
- 信息窗口, 60
- 结构体, 276
- 热复位, 401
- 配置 Modbus 从站, 333
- 配置 RTEX 主从站, 337
- 监视, 404
- 监视列表, 295
- 特殊寄存器区 (S) , 250
- 调用树, 104
- 调试, 50
- 通讯设置, 48, 395
- 通讯参数信息, 323
- 通道参数信息, 321
- 控制器校时, 403
- 菜单栏, 56
- 辅助工具, 454
- 停止, 400
- 清空存储卡, 403
- 清空控制器, 402
- 添加 CPU 模块, 317
- 添加 POU, 89
- 硬件配置, 29, 314
- 硬件配置文件, 314
- 硬件配置窗口, 316
- 程序, 88
- 程序组织单元, 87
- 强制, 407
- 强制变量表, 408
- 编译, 47, 391
- 编译结果, 393
- 跳转, 148
- 置反, 191
- 触点, 130
- 解除强制, 407
- 数据, 248
- 数据区, 248
- 数据类型, 266, 268
- 模块参数信息, 320
- 增量下装, 399
- 增量编译, 392
- 操作日志, 420
- 操作符, 159



地址：北京经济技术开发区地盛中路2号院

邮编：100176

电话：010-58981000

传真：010-58981100

<http://www.hollysys.com>



关注和利时