

ControlStar 编程软件

用户手册

资料版本 V1.3
归档时间 2007-08-13
BOM 31011119

艾默生网络能源有限公司为客户提供全方位的技术支持，用户可与就近的艾默生网络能源有限公司办事处或客户服务中心联系，也可直接与公司总部联系。

艾默生网络能源有限公司
版权所有，保留一切权利。内容如有改动，恕不另行通知。

艾默生网络能源有限公司
地址：深圳市南山区科技工业园科发路一号
邮编：518057
公司网址：www.emersonnetworkpower.com.cn
客户服务热线：800-820-6510
手机及未开通 800 地区请拨打：021-26037141
客户服务投诉热线：0755-86010800
E-mail：info@emersonnetwork.com.cn

前 言

本手册详细描述了 2.3 版 ControlStar 编程软件的操作方法和界面说明，同时介绍了各功能的使用方法并穿插了应用实例。

如何阅读本书

本书介绍了艾默生网络能源有限公司 EC 系列可编程控制器的 2.3 版 ControlStar 编程环境的操作方法。

本书的章节编排由整体到细节，各章节都具有独立的内容，因此可以从头通读逐步掌握 2.3 版 ControlStar 编程环境的全面内容，也可以随时查阅其中的章节。

详细的 PLC 程序语法、指令使用和示例等内容，请参阅《可编程控制器编程参考手册》。

相关文档和参考书目

阅读本手册的同时也可以参考阅读其他手册：用户操作手册、设备选型手册、编程参考手册等。

目 录

第一章 概述	1
1.1 可编程控制器简介	1
1.2 基本控制原理	1
1.3 编程语言	1
第二章 快速入门	3
2.1 安装和卸载 ControlStar	3
2.2 应用举例	4
2.2.1 进入编程环境	4
2.2.2 建立工程	4
2.2.3 编写控制程序	5
2.2.4 建立通讯连接	6
2.2.5 下载程序	6
2.2.6 启动可编程控制器	6
第三章 编程说明	7
3.1 编程环境	7
3.2 工程使用说明	10
3.3 子程序和调用方式	11
3.3.1 程序块的分类和作用	11
3.3.2 如何建立子程序或中断子程序	12
3.3.3 导出子程序	12
3.3.4 导入子程序	12
3.3.5 编辑子程序	13
3.3.6 调用子程序	13
3.3.7 修改子程序（中断子程序）属性	13
3.4 程序加密解密	15
3.5 数据类型说明	15
3.6 使用联机帮助	15
第四章 程序编辑	17
4.1 通用编辑功能	17
4.1.1 如何进行查找、替换和定位	17
4.1.2 如何在多个程序内进行查找	18
4.1.3 如何进行复制、剪切和粘贴	18

4.1.4 如何进行插入行、删除行或者添加行	19
4.1.5 如何进行撤销恢复	19
4.1.6 如何进行打印预览、打印	19
4.1.7 如何检查程序错误	20
4.2 梯形图编辑器	20
4.2.1 梯形图的工作原理及逻辑组成	20
4.2.2 连接关系	20
4.2.3 如何输入梯形图元素	20
4.2.4 如何修改梯形图元件	23
4.2.5 如何删除梯形图元件	23
4.2.6 如何输入注释	23
4.2.7 如何在梯形图中使用已定义的变量	24
4.2.8 如何优化梯形图程序的显示	25
4.2.9 如何使用梯形图的网络号	25
4.3 顺序功能图编辑器	26
4.3.1 预备知识	26
4.3.2 顺序功能图元件说明	26
4.3.3 编辑顺序功能图程序	27
4.3.4 变量的使用	28
4.4 指令列表编辑器	29
4.4.1 指令编辑	29
4.4.2 变量的使用	30
4.4.3 行号、标签	30
4.5 全局变量表	30
4.5.1 全局变量的定义	30
4.5.2 编辑全局变量表	30
4.5.3 导入导出全局变量表	31
4.5.4 全局变量的使用	31
4.6 局部变量表	31
4.6.1 局部变量的定义	31
4.6.2 编辑局部变量表	32
4.6.3 局部变量的使用	32
4.7 交叉引用表	32
4.7.1 交叉引用表的定义和使用	32
4.7.2 切换编址方式	33
4.7.3 元件浏览表的定义和使用	33
4.7.4 元件浏览表的过滤	33

第五章 与 PLC 通讯	35
5.1 通过串口连接	35
5.2 通过 Modem 连接	36
5.3 断开通讯连接	36
第六章 PLC 设置	37
6.1 系统块	37
6.1.1 配置系统块	37
6.1.2 系统块保存范围	37
6.1.3 系统块输出表	38
6.1.4 系统块设置时间	40
6.1.5 系统块输入过滤器	40
6.1.6 系统块输入点	41
6.1.7 系统块高级设置	42
6.1.8 通讯口设置	42
6.1.9 特殊模块配置	44
6.1.10 中断优先级设置	46
6.2 PID 指令向导	46
6.2.1 PID 指令向导执行步骤	46
6.2.2 PID 参数设置子程序参数含义	49
6.2.3 PID 执行子程序	50
6.3 数据块设置	50
6.3.1 数据块简介	50
6.3.2 数据块编辑	51
6.3.3 数据块编译	51
6.3.4 其他编辑功能	51
第七章 PLC 控制	52
7.1 PLC 控制方法	52
7.2 运行 PLC	53
7.3 停止 PLC	53
7.4 复位 PLC	53
7.5 下载程序	53
7.6 上载程序	54
7.7 比较计算机和 PLC 中用户程序	55
7.8 设置密码	55
7.9 清除 PLC 错误信息	55
7.10 清除 PLC 元件值	55

7.11 清除 PLC 程序.....	56
7.12 清除 PLC 数据块.....	56
7.13 PLC 格式化.....	56
7.14 恢复默认配置.....	56
7.15 获取 PLC 信息.....	56
7.16 显示和设置 PLC 当前时间.....	57
7.17 从 RAM 中生成数据块.....	57
第八章 程序调试	59
8.1 如何进入和退出调试状态.....	59
8.2 注意事项.....	59
8.3 梯形图程序调试.....	59
8.3.1 可调试的元件.....	59
8.3.2 元件值写入和强制.....	60
8.3.3 解除强制.....	61
8.3.4 强制结果及限制规则.....	61
8.3.5 读取所有强制元件值.....	62
8.4 指令列表程序调试.....	62
8.5 顺序功能图程序调试.....	62
8.5.1 可调试的元件.....	62
8.5.2 元件值写入和强制.....	63
8.5.3 解除强制.....	63
8.5.4 读取所有强制元件值.....	64
8.6 元件状态监控表.....	64
8.6.1 添加或删除元件监控表.....	64
8.6.2 编辑模式与监控模式.....	64
8.6.3 切换编址方式.....	65
8.6.4 元件监控表的复制填充与序列填充.....	65
8.6.5 导入与导出元件监控表.....	66
8.7 在线修改.....	66
第九章 变频器子程序库	67
9.1 设置变频器.....	67
9.2 使用变频器指令的限制.....	67
9.3 如何使用变频器指令编程.....	68
9.4 变频器指令的输入.....	68
9.5 变频器指令说明.....	69
9.5.1 FRQ.....	69

9.5.2 FWD.....	70
9.5.3 REV	71
9.5.4 STP.....	72
9.5.5 GET.....	73
9.5.6 RD.....	74
9.5.7 WRT.....	75
第十章 PLC 仿真	76
10.1 仿真软件的启动方式	76
10.1.1 正常启动.....	76
10.1.2 从 ControlStar 启动	76
10.1.3 从资源管理器启动	76
10.2 关于仿真配置	76
10.2.1 新建配置.....	76
10.2.2 打开配置.....	77
10.2.3 保存配置.....	77
10.3 仿真界面说明	78
10.3.1 仿真图介绍.....	78
10.3.2 扩展模块.....	78
10.3.3 仿真运行状态控制.....	78
10.3.4 高速输入设置.....	79
10.4 时序图说明.....	79
10.4.1 时序图介绍.....	79
10.4.2 添加删除采样元件.....	80
10.4.3 时序图相关设置.....	80
10.5 PLC 仿真的通讯功能.....	81
10.5.1 Controlstar 建立与仿真的连接	81
10.5.2 PLC 仿真的监控.....	81
10.6 PLC 仿真和 PLC 区别.....	81
10.7 使用仿真调试程序的实例	82
附录一 软元件说明	84
1. 软元件定义.....	84
2. 元件的间接寻址方式.....	85
附录二 指令一览表	86

第一章 概述

1.1 可编程控制器简介

可编程控制器 PLC (Programmable Logic Controller) 是一种数字运算操作的电子系统, 专为工业环境下的应用而设计。它主要将外部的输入信号如: 按键、感应器、开关及脉冲波等的状态读取后, 依据这些输入信号的状态或数值并根据内部储存预先编写的程序, 以微处理机执行逻辑、顺序、计时、计数及算术运算, 产生相对应的输出信号如: 继电器的开关、控制机械设备的操作。通过电脑或程序书写器可轻易的编辑/修改程序及监控装置状态, 进行现场程序的维护与试机调整。

1.2 基本控制原理

可编程控制器工作原理

可编程控制器采用循环扫描工作方式, 包含输入点扫描、用户程序执行、输出点刷新、内部处理和通讯处理等部分。

在运行可编程控制器前, 可以使用编程软件编写输入点和输出点间的控制逻辑并下载到可编程控制器中, 在可编程控制器运行过程中, 首先会扫描输入点信号, 将之读取到可编程控制器中, 然后根据控制程序完成运算和逻辑处理, 运算和逻辑处理结果将输出点的值改变, 最后将输出点中的值转换为电信号输出并控制各种机械设备的运行。

在可编程控制器运行过程中采用循环扫描的工作方式, 通过反复的执行输入点扫描、用户程序执行、输出点刷新工作达到接收控制并操作设备的目的。

用户程序控制原理

在用户程序中, 可编程控制器的输入点被称为触点, 它的功能和工业设备中的开关触点是一样的, 代表能流的导通或者关断。在可编程控制器中, 输入点存储为一个软元件, 当输入点为高电平时, 对应的软元件为导通状态, 在用户程序中参与逻辑运算并影响输出点的值。输出点称为线圈, 代表输出能流的导通或者关断, 输出点对应软元件的值由输入点和控制逻辑计算结果决定。在输出刷新时, 软元件的值被转换为电信号在输出点的晶体管或者继电器输出, 从而完成对设备的控制。

1.3 编程语言

EC 系列可编程控制器的编程工具支持三种常用的语言: 梯形图 (LD), 指令列表 (IL) 和顺序功能图 (SFC)。主程序可以使用上述三种编程语言中的任意一种来编写, 但是子程序和中断子程序只能使用梯形图或者指令列表编写。另外, 顺序功能图中的内置程序只能使用梯形图编写。

梯形图编程

梯形图是一种和电气原理图非常类似的图形编程语言, 在逻辑上也和电气原理图非常相似。在梯形图中, 各种软元件表示为触点和线圈, 图形上各个触点和线圈之间的电路连接关系就是它们的控制逻辑关系。图 1-1 为梯形图编程示例。

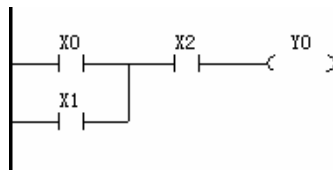


图 1-1 梯形图编程示例

在上面的梯形图中, 首先是装载输入点 X0 的值作为当前值, 然后装载输入点 X1 的值, 将 X1 的值和当前值进行“或”运算后, 运算结果成为当前值, 然后装载 X2 的值并和当前值进行“与”运算, 运算结果将最终控制输出点 Y0 是否有能流导通。

指令列表编程

指令列表程序编辑器是一个文本编辑器，所有的逻辑和运算都使用指令和操作数的方式输入，根据指令所完成的功能和涉及到的操作数中的软元件，完成软元件的值读取、逻辑处理和软元件值写入。表 1-1 为指令列表编程示例。

表1-1 指令列表编程示例

指令	软元件	指令说明
LD	X0	装载输入点 X0 的值作为当前值
OR	X1	装载输入点 X1 的值，和当前值进行“或”运算后结果成为当前值
AND	X2	装载输入点 X2 的值，和当前值进行“与”运算后结果成为当前值
OUT	Y0	根据当前值确定输出点 Y0 的值

顺序功能图编程

顺序功能图是根据机械设备的流程或者工序，将控制分成了多个步和步到步之间转换的一种语言。一个标准的顺序功能图由初始步、一般步、步间的转换条件、跳转和重置组成、每一步就是机械设备的处理工序，一个步中可以有内置梯形图，也就是这一步需要完成的处理工序。转换条件就是一个工序的完成和下一个工序的启动条件，它也需要内置梯形图来表示转换的条件。图 1-2 为顺序功能图编程示例。



图1-2 顺序功能图编程示例

如上面的顺序功能图所示，初始状态是 S0，当 S0 至 S20 的转换条件不满足时，S0 工序会一直被执行，一旦转换条件满足，就停止 S0 的执行，开始 S20 的执行，以此类推，在最后完成 S21 并满足 S21 到跳转 S0 的转移条件后，停止 S21 工序的执行，重新启动初始步 S0 的执行，通过跳转，可以让整个顺序工序循环不停的执行。

程序转换

上述的三种编程语言可以互相转换，根据使用习惯或者实际应用的控制环境，可以选择合适的编程语言。由于顺序功能图语言比较特殊，因此梯形图或者指令列表必须遵守顺序功能图的语法才能正确转换到顺序功能图。

第二章 快速入门

在本章，我们首先简单介绍如何安装和卸载 ControlStar。然后通过一个简单的例子，介绍如何进入编程环境，编写程序以及将程序下载到可编程控制器中并运行。

2.1 安装和卸载 ControlStar

计算机配置要求

ControlStar 编程软件要求在 IBM PC486 以上的微型机或兼容机上运行，支持 Microsoft 的 Windows 98、Me、Windows 2000 及 XP 等操作系统。需要安装 ControlStar 的计算机的最低配置和推荐配置见表 2-1。

表2-1 计算机最低配置和推荐配置

部件	最低配置	推荐配置
CPU	相当于 Intel 公司的 Pentium 233 或以上级别的 CPU	相当于 Intel 公司的 Pentium 1G 或以上级别的 CPU
内存	64M	128M
显卡	可工作于 640*480 分辨率，256 色模式下	可工作于 800*600 分辨率，65535 色模式下

安装过程

ControlStar 发布的安装包为单独的可执行程序，双击即显示图 2-1 所示的安装启动界面。然后按照提示逐步安装即可。



图2-1 安装启动界面

安装完毕后，在开始菜单中会出现 ControlStar 程序组，同时会把 ControlStar 快捷方式放置在桌面上。

卸载

通过选择 ControlStar 程序组中的**卸载 ControlStar** 菜单或者在控制面板中使用**添加/删除程序**功能可以卸载本软件。

需要重新安装或升级 ControlStar 时，必须首先卸载旧版本的软件。为了避免版本升级可能带来的混乱，在安装和卸载的过程中，软件会自动清理注册表中的相关数据，因此，在重新安装或者升级之后，先前定制过的菜单、工具栏、系统选项等信息需要重新设置。

2.2 应用举例

本节以一个简单的跑马灯程序来介绍 ControlStar 的使用方法。用 EC20 控制器的输出端 Y0、Y1、Y2 分别控制红、黄、绿三种颜色的灯，输入端子 X0 作为启动开关。该程序作如下控制：当开关合上时，红黄绿三盏灯交替点亮，每盏灯的点亮时间为 5 秒钟（使用精度为 100ms 的计时器 T0、T1、T2）。当开关打开时，三盏灯全灭。

2.2.1 进入编程环境

ControlStar 安装完成后，在开始菜单中单击 ControlStar 图标即可启动该软件从而进入编程环境。编程环境主界面如图 2-2 所示。

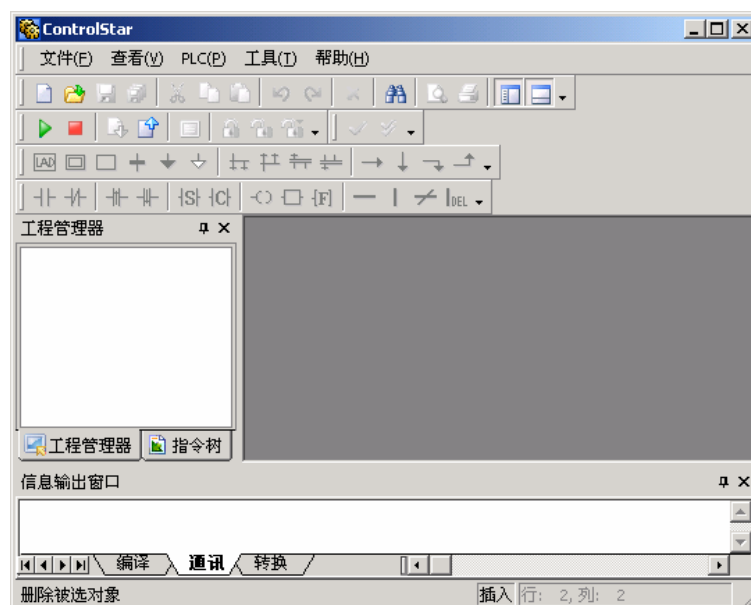


图2-2 ControlStar 编程环境主界面

2.2.2 建立工程

进入编程环境后，首先需要为编写的程序创建一个工程。点击**文件/新建工程**，软件弹出下面的对话框。

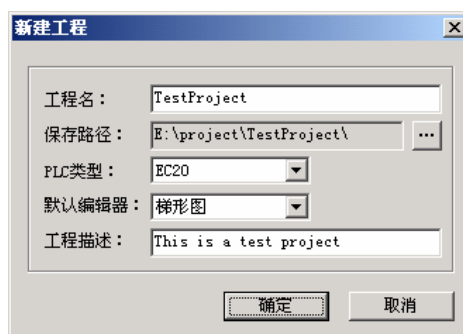


图2-3 新建工程对话框

在本示例中，选择 PLC 类型为 EC20，默认编辑器为梯形图，然后点击对话框中的**确定**按钮，即创建了一个新的工程。工程新建后，系统会自动打开工程中的主程序窗口，并进入程序编辑状态。如图 2-4 所示。

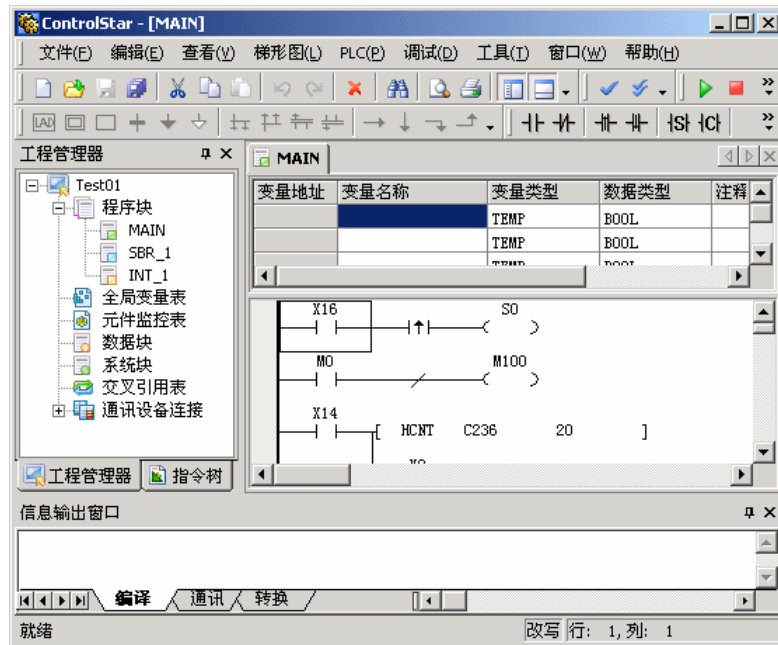


图2-4 主程序界面

2.2.3 编写控制程序

1. 在编程环境中输入如图 2-5 所示的梯形图程序。

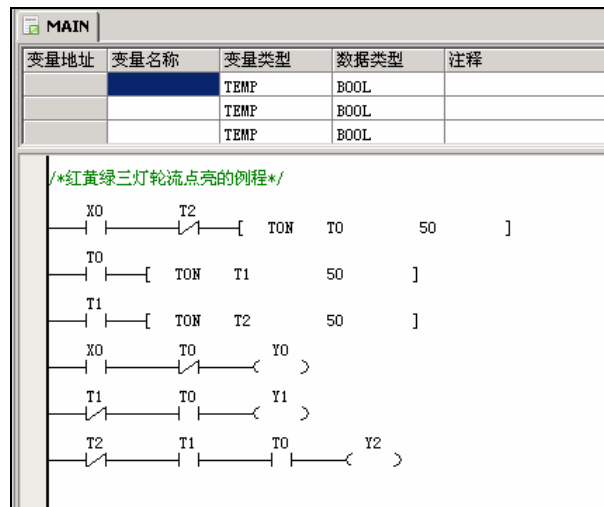


图2-5 输入梯形图程序

2. 程序编写完成后，在菜单条上点击**文件/保存工程**，即可保存当前编写的工程。
3. 保存完成后，需要检查当前编写的程序是否有错误，并且编译为可以下载到可编程控制器中的目标文件。在菜单条上点击 **PLC** 菜单中的**全部编译**菜单项，软件将当前程序进行全部编译，编译结果会显示在输出窗口中，如果没有错误的话，显示如图 2-6。



图2-6 编辑结果输出窗口

2.2.4 建立通讯连接

在下载程序到 PLC 硬件前，需要建立和 PLC 的串口通讯连接，具体设置方法请参考 [通讯口设置](#)。

2.2.5 下载程序

点击 PLC 菜单中的**下载**菜单项，下载程序时，必须要求 PLC 处于停止状态，如果 PLC 正在运行，软件会提示如下的对话框。



图2-7 PLC 运行状态提示信息

点击**确定**按钮停止 PLC 的运行，软件会弹出如下的编译提示对话框。

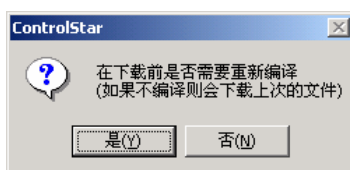


图2-8 编译信息对话框

如果你在 2.2.3 节的步骤 3 中对当前的程序进行了编译，并且编译通过后程序没有再进行任何更改，点击**否**。如果编译后又修改了程序，选择**是**，更改后的程序将重新编译，编译完毕后，会出现下面的下载窗口：



图2-9 下载窗口

根据需要，在下载选项中选择需要下载的内容。本示例只需要下载编写的应用程序，选中后点击**下载**按钮即开始下载。下载过程中会显示进度条，完成后将会提示下载成功。

2.2.6 启动可编程控制器

程序下载完成后，需要启动可编程控制器运行。在菜单条上 PLC 菜单中选择**启动**菜单项，或者将可编程序控制器上用于控制启停的拨码开关拨到 RUN 处来启动可编程控制器。在可编程控制器正常运行后，闭合接在输入端子 X0 上的开关，就可以看到接在输出端子 Y0、Y1、Y2 的三盏灯被轮流点亮。

第三章 编程说明

3.1 编程环境

本程序的主界面基本包括七个部分，见图 3-1 所示。

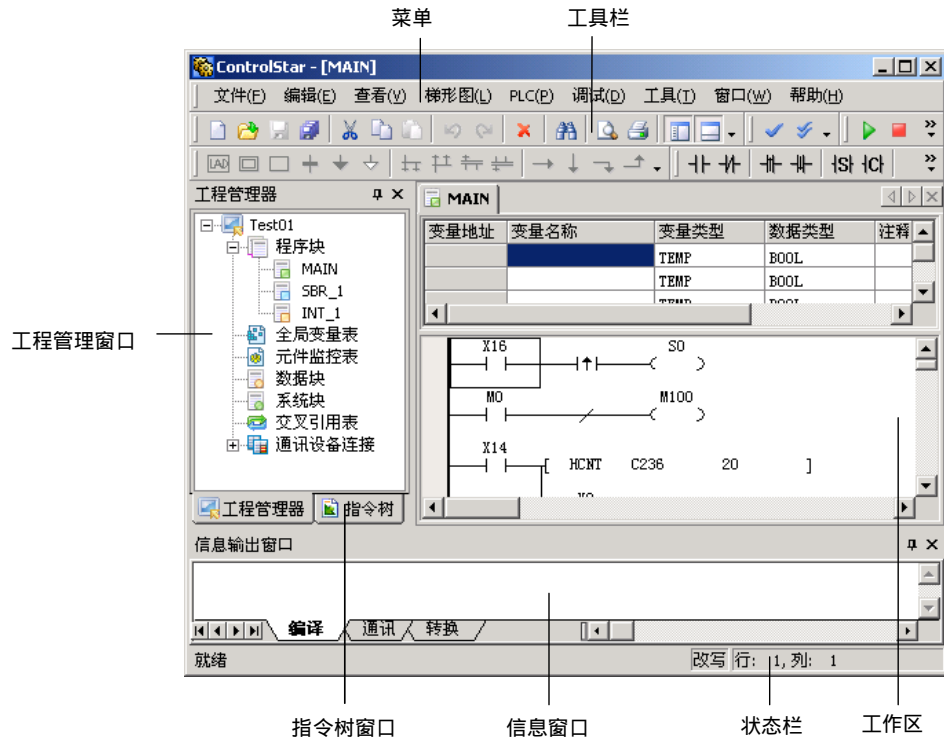


图3-1 程序主界面

菜单

菜单包括了一组含有全部命令的子菜单。当鼠标停留在菜单项上时，菜单项功能的简短描述将显示在状态栏中。其中 PLC 子菜单提供了对 PLC 硬件操作的相关功能。工具子菜单提供了设置 ControlStar 相关属性和复杂指令的向导功能。

ControlStar 会随着当前打开的窗口的不同而显示不同的菜单。各子菜单的菜单项根据正在使用的功能也会有一定差别。

工具栏

本软件提供了几个工具栏，其中包含了用于更快地访问要经常使用的操作的不同命令按钮。这些操作也可通过使用菜单项或预定义的快捷键来完成。

工具栏位于菜单栏的下方。默认情况下，所有的工具栏都是可见的。要想隐藏或显示某一工具栏，请在任意一个工具栏上点击鼠标右键，在弹出的快捷菜单中选择/取消选择某个工具栏。

将鼠标光标放在任一图标上（而不点击它）停留片刻，会出现一个简短的描述文本，被称为工具提示。这些工具提示包括了当前图标的名称。如果工具提示没有出现，请在“选项”对话框中使能它们。

1. 标准工具栏

标准工具栏包含编辑 PLC 程序最常用的基本功能。

2. 编译工具栏

编译工具栏包含编译当前程序和全部编译两个编译最常用的功能：编译当前窗口中的程序/编译所有程序。

3. 梯形图工具栏

梯形图工具栏包含编辑梯形图程序最常用的功能。

4. 顺序功能图工具栏

顺序功能图工具栏包含编辑顺序功能图程序最常用的功能。

5. 标签工具栏

标签工具栏主要应用在指令列表、数据块两个文本编辑器中，功能是在文本窗口中使用可供快速定位的标签。

6. PLC 工具栏

PLC 工具栏提供了操作、访问 PLC 硬件最常用的功能。

每个工具栏的功能都可以在界面菜单中找到对应的菜单项。可以根据需要自定义工具栏，也可以添加新的工具栏，加入常用的图标。

7. 缩放工具栏

在梯形图程序和顺序功能图程序中，可以通过缩放工具栏的调整显示比例。

快捷键

快捷键定义见表 3-1。

表3-1 快捷键定义

类别	快捷键	功能	类别	快捷键	功能	类别	快捷键	功能
通用快捷键	CTRL + N	新建工程	梯形图编辑快捷键	CTRL + 1	常开触点	顺序功能图编辑快捷键	CTRL + 1	梯形图块
	CTRL + O	打开工程		CTRL + 2	常闭触点		CTRL + 2	初始步进行符
	CTRL + A	全选		CTRL + 3	上升沿触点		CTRL + 3	普通步进行符
	CTRL + C	拷贝		CTRL + 4	下降沿触点		CTRL + 4	转移符
	CTRL + X	剪切		CTRL + 5	步进点		CTRL + 5	跳转符
	CTRL + V	粘贴		CTRL + 6	比较触点		CTRL + 6	重置符
	CTRL + F	查找		CTRL + 7	线圈		CTRL + 7	选择分支
	CTRL + H	替换		CTRL + 8	应用指令		CTRL + 8	选择汇合
	CTRL + G	定位		CTRL + 9	子程序调用		CTRL + 9	并行分支
	CTRL + Z	撤销		CTRL + B	插入块注释		CTRL + 0	并行汇合
	CTRL + Y	恢复		CTRL +	竖线		0	打开内置梯形图
	CTRL + S	保存		CTRL+SHIFT+	删除竖线		CTRL +	竖线
	CTRL + P	打印		CTRL +	横线		CTRL +	横线
	Delete	删除所选内容		CTRL + I	插入行		CTRL + I	插入行
	F1	打开联机帮助		CTRL + L	删除行		CTRL + L	删除行
	F5	PLC 运行						
	F6	PLC 停止						
	F7	编译所有文档						
	CTRL + F7	编译当前文档						
F8	下载							
F9	监控							

工程管理器窗口

工程管理器窗口中使用一个工程树显示当前工程的名称及工程中所包含的所有文档，见图 3-2。程序块下的程序节点可以根据实际需求添加或删除（主程序不能删除），而全局变量表、元件监控表、数据块、系统块、交叉引用表等节点是固定的。



图3-2 工程管理器窗口

为了能够明显地区分主程序、子程序和中断子程序，在工程树中每一类程序节点对应不同颜色的图标，主程序显示为绿色，子程序显示为蓝色，中断子程序显示为黄色。

通过工程树可以实现以下功能：

1. 右键单击工程名节点，在弹出菜单中可以选择保存工程、另存工程、关闭工程或者修改工程属性等。
2. 右键单击程序块节点，在弹出菜单中可以选择插入子程序或中断子程序。
3. 右键单击程序块下的某个程序节点，在弹出菜单中可以选择打开该程序、删除该程序（主程序不能删除）或者修改其属性。
4. 右键单击其他节点，在弹出菜单中只能进行打开操作。

指令树窗口

指令树窗口如图 3-3 所示。

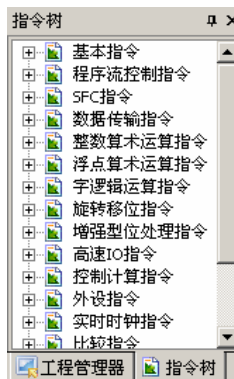


图3-3 指令树窗口

指令树的内容会随着当前工作区窗口的编辑器的改变而变化。如果当前工作区为梯形图编辑器，指令树将显示梯形图语言支持的所有指令。同样，如果当前工作区为指令表编辑器或者顺序功能图编辑器，指令树也显示为指令表语言或者顺序功能图语言支持的所有指令。因此在编写程序过程中，可以通过指令树查找与某个功能匹配的指令。在编程过程中，指令树不仅可以用于查找指令，还可以用于输入指令。

如果要通过指令树输入指令，需要先选中要输入的指令，双击鼠标左键或将其拖动到程序编辑窗口中，即可在程序编辑窗口中输入该指令。在鼠标拖动过程中，其形状可能会发生改变，当鼠标的形状变为箭头时表示在该位置可以输入该指令。如果当前编辑器是梯形图编辑器，将自动启动输入指令的向导，详情见《如何输入梯形图元素》中的向导式输入部分。如果当前编辑器是指令列表或者顺序功能图，编辑器将直接插入该指令或该指令对应的元件。

信息输出窗口

信息输出窗口用于输出软件执行操作之后的结果，见图 3-4。



图3-4 信息输出窗口

状态栏

状态栏用于提供常用的属性信息，见图 3-5。



图3-5 状态栏

工作区

工作区包括程序编辑窗口、全局变量表编辑窗口、元件状态监控表窗口、数据块窗口和交叉引用表窗口。

3.2 工程使用说明

在使用 Controlstar 开发程序的过程中，需要编写或者用到程序块、全局变量表、元件监控表、数据块、系统块、元件交叉引用表等各种类型的文件，ControlStar 使用工程对各种类型的文件进行管理。

工程文件的扩展名为 proj，工程有如下属性，可以在新建工程对话框中对工程属性进行选择或者输入，见图 3-6。

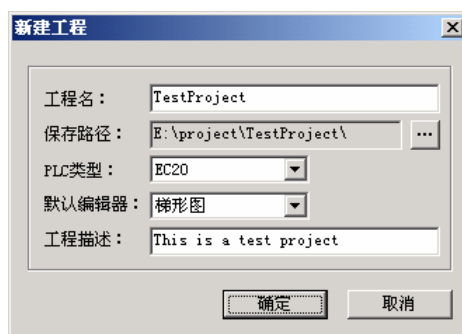


图3-6 新建工程对话框

工程名：工程标识，同时也是工程保存时对应的磁盘文件名。

保存路径：指定工程文件保存的全路径，所指定的路径加上工程名称子目录（自动创建）就是最终的工程文件存放目录。

PLC 类型：表示工程所针对的 PLC 的类型，选择不同的 PLC 类型或版本对 CPU 指令集和系统块配置所支持的程序容量有影响。其中程序容量会影响到编译和下载过程，编译程序时要做容量检查，如果程序步数超过设定值，给出警告信息，在下载时给出错误信息。

默认编辑器：有梯形图、指令列表、顺序功能图三种编辑器可供选择。

工程描述：工程的简单说明，工程描述可为空，最长 128 个字符。

在将程序下载到 PLC 时，软件会对工程定义的 PLC 类型和实际连接的 PLC 类型作比较，如果类型不匹配将不能下载。

如需更改某个工程的 PLC 类型，可以选择菜单文件下的**更改 PLC 类型...**菜单项。或在工程树上选择根节点，然后在右键菜单中选择**属性**，在弹出的工程属性对话框中，除工程名和保存路径不能修改外，其他属性都可修改，包括 **PLC 类型**。更改工程 PLC 类型后，可能导致编译不通过，这可能是由于指令或系统块配置不兼容，需要根据编译提示信息手工更改。

新建工程时的默认 PLC 类型和默认编辑器类型可以在系统选项对话框中指定，如图 3-7。

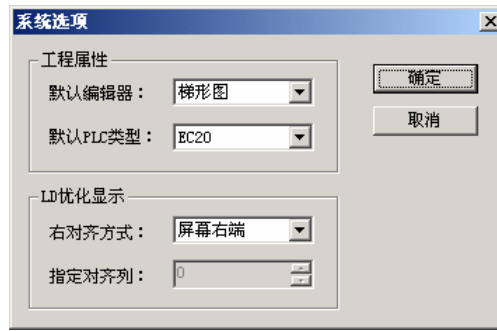


图3-7 系统选项对话框

对应每个工程一般有 CompMonTbl.CMT, CrossRefTbl.CRS, SysCfgInfo.SSC, DatBlock.DB 四个文件, 这四个文件分别保存状态监控表, 交叉引用表, 系统配置表, 以及数据块信息, 他们都保存在 DATA 目录下面。

在 POU 目录下保存着程序单元, 包括主程序、子程序、中断子程序等等, 选择的编辑器类型不同, 生成的程序单元的类型也不同, 指令列表程序、梯形图程序、顺序功能图程序的扩展名分别为*.IL、*.LD、*.SFC。

全局变量表信息一般也保存在该目录下, 全局变量表的文件名为 GlobalVarTbl.GVT。

ControlStar 使用图 3-8 所示的工程管理器管理一个工程。



图3-8 工程管理器

不要在 Windows 中直接删除或者修改上面描述的文件, 否则可能导致工程无法打开或者程序数据被破坏。

3.3 子程序和调用方式

3.3.1 程序块的分类和作用

软件提供主程序、子程序和中断子程序三种不同的程序单元。

主程序：主程序只能有一个, 由软件默认提供, 主程序是 PLC 应用程序启动执行的程序。主程序可以由梯形图、顺序功能图或者指令列表语言中的任何一种编写。

子程序：一个工程内可以有多个子程序, 不能超过 64 个, 子程序由主程序或者其他子程序调用, 主要完成一些常用或者需要重复使用的功能, 子程序只能由梯形图或者指令列表编写, 不能用顺序功能图编写。

中断子程序：可以有多个, 通过和某个中断事件相关联, 在中断事件发生时调用。中断子程序不能超过 40 个。中断子程序只能由梯形图或者指令列表编写, 不能用顺序功能图编写。

使用子程序的目的是为了对程序进行分段和分块, 把一些常用的功能模块写成子程序后可以重复调用, 通过使用较小的程序块, 只在需要时才调用程序块, 可以更有效地使用 PLC, 因为所有的程序块可能无须执行每次扫描, 当主程序调用子程序并执行时, 子程序执行全部指令直至结束。然后, 系统将控制权返回至调用子程序网络中的主程序。

3.3.2 如何建立子程序或中断子程序

右键单击“工程管理器”窗口的程序块，选择插入子程序或者插入中断子程序，如图 3-9 所示。新的子程序名的默认名称是 SBR_*，新的中断子程序名的默认名称是 INT_*（其中的*是由软件自动计算的一个数字），建立完成后可以通过子程序属性对话框将默认程序名改为一个更有意义的名称。

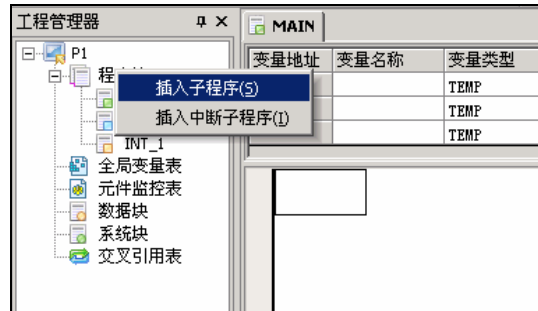


图3-9 建立子程序

在工程树上插入新程序节点的同时，该程序也会被打开，可以马上对它进行编辑。

3.3.3 导出子程序

软件提供了子程序库的导入和导出功能，对于某个通用功能，可以在子程序中编写完成后，通过导入/导出功能，在不同的工程中再次使用，避免相同功能的重复代码编写。ControlStar 使用程序库文件的方式来管理这些程序，通用的子程序首先可以导出到某个目录形成可以重复使用的程序库文件，在需要使用这些子程序的工程中，使用导入功能可以将程序库文件导入到当前工程中作为标准的子程序使用。

选择菜单**文件**下的**程序文件导出...**菜单项，打开程序导出对话框，见图 3-10。

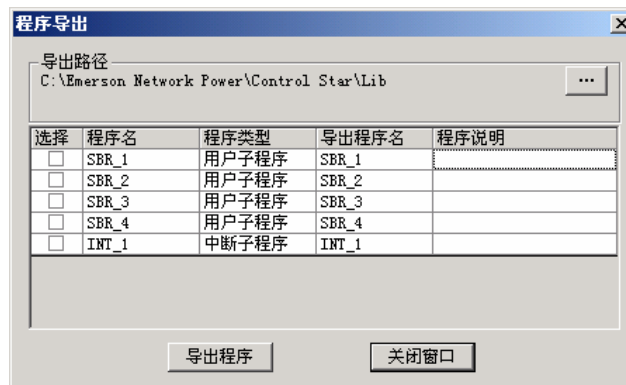


图3-10 程序导出对话框

对话框中自动把该工程中可以导出的程序列出来，程序导出的路径第一次使用时默认为软件安装路径下的\lib 目录下，可以点击...按钮来更改，更改后路径自动成为下次默认的导入/导出路径。

可以编辑程序的导出名以及程序说明。最后在需要导出的程序前打勾，然后点击**导出程序**按钮。程序正确导出后该程序前的勾自动消失，否则仍然处于导出前的状态。

3.3.4 导入子程序

选择菜单**文件**下的**程序文件导入...**菜单项，打开图 3-11 所示的程序导入对话框。

对话框中自动列出了在默认导入导出路径下可以导入的程序信息，可以点击...按钮来更改路径。导入前可以编辑程序导入工程后的名称，当导入程序名不合法时，在列表中用红颜色加下划线来提示，如果所指定的程序名与工程中已有的程序重名时，在导入时会提示是否要覆盖。选择好程序后，点击**导入程序**按钮，程序导入到工程。

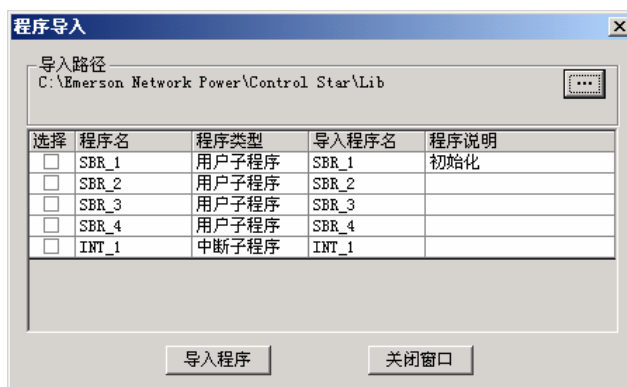


图3-11 程序导入对话框

3.3.5 编辑子程序

从工程树中双击某个子程序节点，可以打开该程序。程序编辑区域由两个部分组成，上面为局部变量表，下面为程序编辑窗口。如果需要为子程序指定参数，可以使用该子程序的局部变量表定义参数，每个子程序调用的最大输入/输出参数限制为 16。

3.3.6 调用子程序

在子程序局部变量表中定义参数（如果有）。参数必须有一个符号名、一个变量类型和一个数据类型。

局部变量表中的变量类型域定义参数的类型：输入参数（IN）、输入输出参数（IN_OUT）或输出参数（OUT）。

IN：如果调用子程序时输入参数是元件，元件中包含的值被传入子程序的局部变量表中对应的变量中。如果参数是数据常量（例如 16#1234），常量数值被输入子程序。

IN_OUT：调用该子程序时，位于指定参数位置的数值被输入子程序中对应的局部变量，来自子程序的结果数值最后通过相同的局部变量返回至调用时作为输入参数的元件中。

OUT：来自子程序的结果数值被返回至指定的参数位置。

TEMP：未用作交接参数的任何本地内存，每个循环周期重复进入子程序时，TEMP 型的变量都会被重新初始化，上次的计算结果不会保留。

需要增加参数数量时，从右键菜单中选择**插入行**，在当前行的上方会插入一个 newRow，新插入行的变量类型和当前行相同。

3.3.7 修改子程序（中断子程序）属性

在工程树上选中要修改的子程序节点，从右键菜单中选择**属性**，可以打开文件属性对话框。在子程序属性对话框中可以修改该子程序的程序名称、作者及程序说明，如图 3-12 所示。

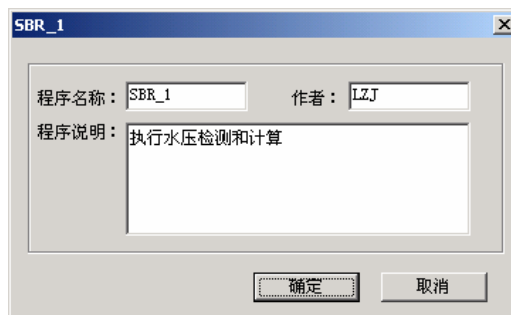


图3-12 修改子程序属性对话框

对于中断子程序，可以为其指定中断号（新建的中断子程序默认中断号为 1，表示未设置），中断子程序属性对话框如图 3-13。

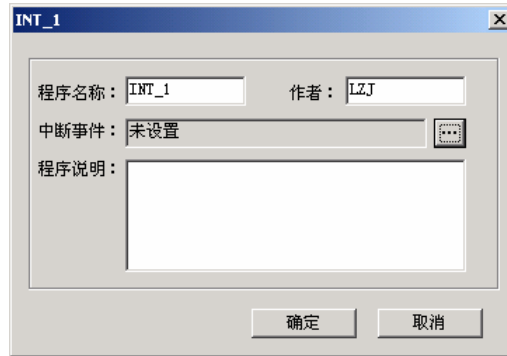


图3-13 修改中断程序对话框

点击...按钮，会弹出中断分配对话框，在这里显示当前可用的空闲中断以及已经被其他中断子程序使用的中断，可以在空闲中断中选择一个中断号指定给当前中断子程序，如图 3-14。

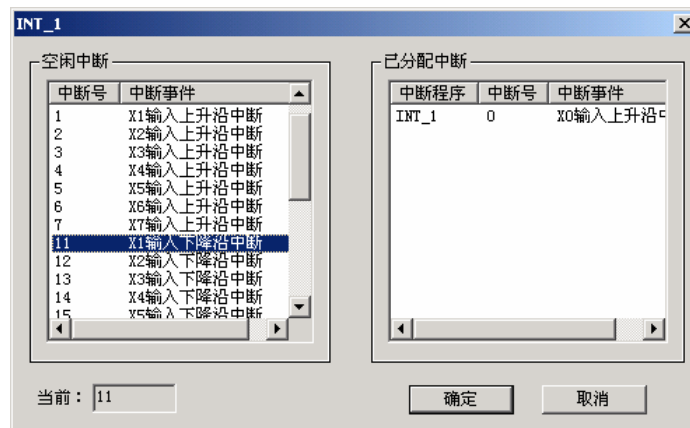


图3-14 指定中断号

由于系统中的某些中断事件是相互冲突的，比如 X0 上升沿和 X0 下降沿是不能同时被系统响应的，因此如果 X0 上升沿中断被分配后，X0 下降沿中断就不能再被使用了，在这种情况下，X0 下降沿中断会在空闲中断中被屏蔽掉。相互冲突的中断事件如下：

X0 输入上升沿中断（中断号：0） \longleftrightarrow X0 输入下降沿中断（中断号：10）
 X1 输入上升沿中断（中断号：1） \longleftrightarrow X1 输入下降沿中断（中断号：11）
 X2 输入上升沿中断（中断号：2） \longleftrightarrow X2 输入下降沿中断（中断号：12）
 X3 输入上升沿中断（中断号：3） \longleftrightarrow X3 输入下降沿中断（中断号：13）
 X4 输入上升沿中断（中断号：4） \longleftrightarrow X4 输入下降沿中断（中断号：14）
 X5 输入上升沿中断（中断号：5） \longleftrightarrow X5 输入下降沿中断（中断号：15）
 X6 输入上升沿中断（中断号：6） \longleftrightarrow X6 输入下降沿中断（中断号：16）
 X7 输入上升沿中断（中断号：7） \longleftrightarrow X7 输入下降沿中断（中断号：17）


使用子程序的几个规则如下：

1. 子程序可以嵌套使用（在某个子程序中调用另外一个子程序），最大嵌套深度为 6。不能从中断程序调用子程序。
2. 子程序之间禁止显式循环调用，如子程序 A 和 B 互相调用对方。
3. 子程序禁止递归调用，例如程序 A 调用程序 B，程序 B 调用程序 C，程序 C 再调用程序 A，形成一个环。此外子程序不允许调用本身。

3.4 程序加密解密

加密

如需保护程序的内容，可以使用加密功能。加密后的程序无法查看和编辑，加密的程序在下载至 PLC 后，再次上载上来仍然是加密的。加密可以对每一个程序单元单独操作，主程序、子程序和中断子程序均可以加密。

加密后的程序，在工程树上会显示一个  标记，在双击打开后，会提示该程序已加密。

欲加密某个子程序，需要在工程树上选中该程序节点，然后从右键菜单中点击**加密/解密**，将弹出图 3-15 所示对话框，密码最长支持 16 位。

解密

如需查看或编辑已加密的程序，则要先进行解密。

欲解密某个子程序，需要在工程树上选中该程序节点，然后从右键菜单中点击**加密/解密**，将弹出图 3-16 所示的密码解密对话框，输入正确的密码即可解密。

加密和解密功能需要 PLC 硬件支持。对于不支持该功能的 PLC 类型，在进行加密或解密时会提示不支持该操作。

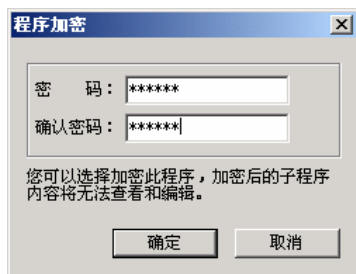


图3-15 密码加密对话框

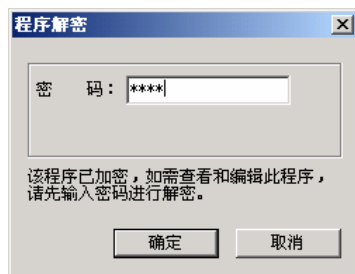


图3-16解密对话框

3.5 数据类型说明

系统支持 BOOL、WORD、DWORD、INT、DINT、REAL 六种数据类型，每种数据类型说明见表 3-2。

表3-2 数据类型说明

数据类型	描述	位	范围
BOOL	布尔	1	ON、OFF
WORD	字	16	0 至 65535
DWORD	双字	32	0 至 4294967295
INT	整数	16	-32768 至+32767
DINT	双整数	32	-2147483648 至+2147483647
REAL	实数	32	±1.175494351 E - 38 ~ ±3.402823466 E + 38

3.6 使用联机帮助

ControlStar 软件提供了丰富的联机帮助功能，在软件使用过程中遇到的问题，应该首先使用联机帮助寻求解答。可以在开始菜单的 ControlStar 程序组或者在软件的任何窗口中选择**帮助/帮助主题**菜单，打开联机帮助。

为了能够快速准确的定位帮助信息，软件还提供了上下文相关的帮助功能：

1. 在工作区中按 F1 键，软件会根据当前窗口的类型（比如梯形图编辑窗口、顺序工程图编辑窗口或者全局变量表编辑窗口）打开与之相对应的文档类型的帮助。
2. 在梯形图中选中某个元件，按 F1 键可以打开该元件所使用的指令的帮助。
3. 在指令列表窗口中选中某行指令，按 F1 键可以打开该指令的帮助。
4. 在指令树上选中某个指令，按 F1 键可以打开该指令的帮助。

5. 在使用某个对话框时，按 F1 键可以打开针对此对话框的帮助。
6. 在编译、转换或者通讯过程中产生的任何错误信息，可以在信息输出窗口点击选中该信息，然后按 F1 键可以打开针对该错误的原因描述和解决方案。

第四章 程序编辑

ControlStar 系统提供了梯形图、指令表、顺序功能图三种程序编辑器。为了定义和使用全局变量和局部变量，ControlStar 系统还提供了全局变量表和局部变量表编辑器。此外系统还提供交叉引用表以用于记录元件在各个程序中的使用情况。

为了使编写程序更加方便和快速，ControlStar 系统提供了一些通用的编辑功能，合理使用这些编辑功能将会更加高效的使用 ConstrolStar 系统编写程序。

4.1 通用编辑功能

4.1.1 如何进行查找、替换和定位

查找

选择**编辑/查找**菜单或按 CTRL+F 键，可以激活查找对话框，如图 4-1 所示。然后可以在当前编辑器中进行查找操作。查找功能可以查找指定的字符串，包括元件地址、变量符号等，前提是目的字符串必须在当前编辑器中有显示。对于梯形图编辑器，块注释中的文字不在搜索范围内。

查找功能支持全字匹配、大小写匹配、查找方向等选项。无论选择向上或向下查找，在查找到一端结束后，都会从另一端重新开始。

对于指令表编辑器，顺序功能图编辑器、以及各个表格将查找到的目标的背景色显示为深色，对于梯形图编辑器则将查找到的目标会用虚线方框标识出来。

替换

选择**编辑/替换**菜单或按 CTRL+H 键，可以激活替换对话框，然后可以在当前编辑器中进行替换操作。

执行替换功能时，要求查找内容和替换内容都不能为空，而且二者不能完全相同。替换后的元件同样要进行正确性检验，如果检验失败将不会执行替换。在替换过程中，可以选择逐个替换或者全部替换，如果选择逐个替换，替换失败时将会提示详细的失败原因，但是在全部替换时则不会。替换对话框如图 4-2。

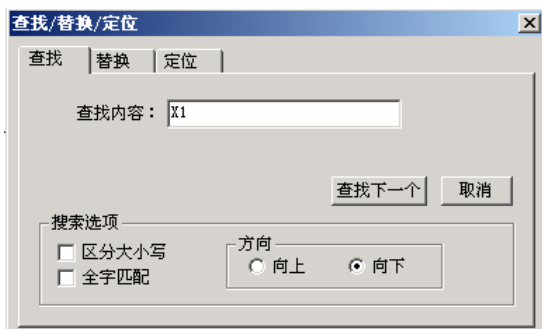


图4-1 查找对话框

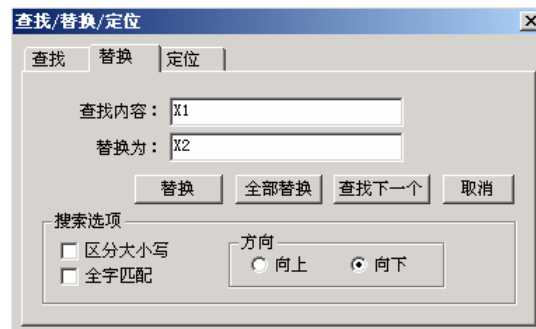


图4-2 替换对话框

定位

选择**编辑/定位**菜单或按 CTRL+G 键，可以激活定位对话框，使用定位功能可以快速定位到当前编辑器中的某一行。定位对话框如图 4-3 所示。

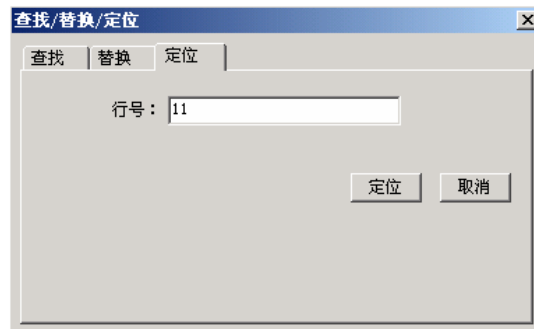



图4-3 定位对话框

4.1.2 如何在多个程序内进行查找

多个程序查找是指查找整个工程中的所有程序，并把查找结果显示在输出窗口，可以选择在所有程序、已打开的程序或当前程序范围内进行查找。变量表、监控表等非程序文档不在查找的范围内，已加密的程序在查找时会略过。

选择**编辑/工程内查找**菜单或按工具条上的按钮，弹出图 4-4 所示对话框。

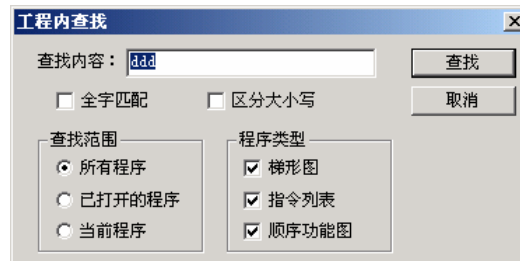


图4-4 在工程中多个程序间查找的对话框

在图 4-4 所示的对话框内可以指定查找的字符串，包括元件地址、变量符号等，并选择查找的范围和程序类型。设置完成后按**查找**按钮即可开始查找。查找的结果将放到输出窗口的查找页面中，如下图所示。

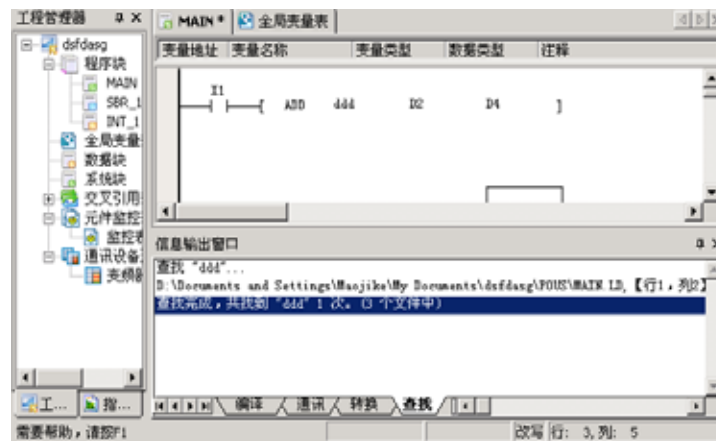


图4-5 查找页面

在上图中双击查找结果即可以定位到相关的程序处。

4.1.3 如何进行复制、剪切和粘贴

复制

选中某个单元格或者拖动鼠标选中某个区域后，使用**编辑/复制**菜单或者按 **CTRL+C** 键，可以将所选的元件拷贝到系统剪贴板。对于梯形图编辑器，只有当某元件所占用的单元格的最左边的一格在选择区域内时，该元件才会被复制。对于梯形图编辑器，当复制的区域内有竖线时，只有当该竖线所跨过的上下单元格都在选择范围内时，该竖线才会被复制到剪贴板中。

剪切

选中某个单元格或者拖动鼠标选中某个区域后，使用**编辑/剪切**菜单或者按 CTRL+X 键，可以将所选的元件拷贝到系统剪贴板，同时清除选中的元件。

粘贴

使用**编辑/粘贴**菜单或者按 CTRL+V 键可以将剪贴板上内容粘贴到当前编辑器的当前位置中。无论当前使用哪种编辑模式，执行粘贴功能时一律使用覆盖模式处理，即先将要粘贴的范围内的元件全部清除，然后再把剪贴板中的元件逐个粘贴上来。不同编辑器之间不能被粘贴。

其它说明

在梯形图和顺序功能图编辑器中，可以通过拖动鼠标选择一个区域，也可以先选中某个单元格，然后按住 Shift 键并用鼠标点击另外一个单元格，这样也可以选择两个单元格之间的区域。

4.1.4 如何进行插入行、删除行或者添加行

插入行


使用行插入功能可以在编辑器中插入一空行，需要使用此功能时，先要在要插入的行上选中任一单元格，然后从右键菜单中选择**行插入**，系统会将当前行及下面所有行的元件向下移动一行，从而在当前行腾出一个空行出来。

请注意，对于梯形图程序如果当前行的元件和上面一行的元件有竖线相连的话，竖线也会被自动拉伸以保持原有连接关系不变。如果当前梯形图中已经达到 4000 行，则不能再插入行。

删除行

使用行删除功能可以删除当前编辑器中的某一行，需要使用此功能时，先要在要删除的行上选中任一单元格，然后从右键菜单中选择**行删除**，系统会将当前行清空，并将下面所有行的元件向上移动一行。同样，也可以选中多行后使用此功能一次性删除多行。

添加行

添加行指在表格的末尾添加一个新行，在全局变量表、局部变量表、元件状态监控表中提供了该功能，可以通过从右键菜单中选择**添加行**，也可以直接按键盘的  键为当前表格添加一个新行。

4.1.5 如何进行撤销恢复



在程序编辑过程中可以单击常用工具栏的**撤销**按钮或者编辑菜单下的**撤销**菜单撤销误操作，最多可撤销最近的 1000 次操作。也可以单击**恢复**按钮或者选择编辑菜单下的**恢复**菜单恢复最近一次撤销的操作，最多可以恢复最近 1000 次撤销操作。

4.1.6 如何进行打印预览、打印

打印预览

打印之前，可以在屏幕上进行打印预览，点击**文件/打印预览**菜单或者工具栏上的  按钮，可以打开打印预览窗口。

打印

需要打印时，请点击工具栏上的  按钮，或在打印预览窗口中点击  按钮，首先弹出标准打印对话框，可以在此设置打印选项，确定后即可将当前程序输出到打印机。

梯形图和顺序功能图打印时，如果某一行因太长而一页打印不下的话，系统会在后面的纸上继续打印该行的剩余部分，直到该行完全被打出来。

梯形图程序打印时，如果某一页打印的最后一行有竖线的话，则其后面一页的顶行会将这些竖线重复打印一遍以示连接关系。

4.1.7 如何检查程序错误


当完成程序的编辑后，点击工具栏上的编译按钮，可以检查当前程序中是否有错误。输出窗口中将显示错误信息和错误位置，双击该错误信息将定位到错误处。输出窗口如下图所示。如果输出窗口显示的错误数为 0，则表示程序无错误，编译通过。








图4-6 编译结果输出信息

4.2 梯形图编辑器

4.2.1 梯形图的工作原理及逻辑组成

梯形图是一种与电气中继图相似的图形语言。一个梯形图程序是由若干个逻辑网络组成的，网络又是由若干个相互连接的图形组件组成的，这些图形组件是构成梯形图程序的最基本元素，大致分为以下几种类型：

1. 触点——代表电气图中的开关。电流仅在触点关闭时通过常开触点，或在触点打开时通过常闭触点。
2. 线圈——代表由功率流充电的中继或输出。
3. 功能块——又称应用指令，代表当功率流到达时执行的一项功能（例如，数据传输、数据运算、计时器、计数器等）。
4. 连接线和——代表电气图中的导线，用于导通彼此相连的其他元件。

网络由以上元素组成并代表一个完整的线路。电源从左边的电源杆流过（在梯形图编辑器中由窗口左边的一条垂直线代表）闭合触点，为线圈或功能指令充电。

4.2.2 连接关系

与电气图类似，在梯形图中，各种元件或由元件组成的块之间有两种关系：串联和并联。分别如图 4-7 所示。

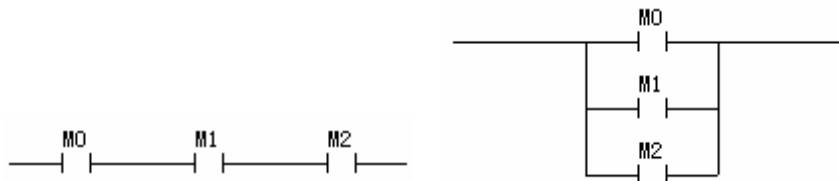


图4-7 串联和并联连接关系

4.2.3 如何输入梯形图元素

在梯形图中，整个输入区域被自动划分为许多个隐式的单元格，可以通过鼠标点击或键盘的方向键来选择单元格，选中的单元格会被黑框包围以提示当前位置。在选定单元格后，才可以在相应位置输入元件，元件根据自身大小不同，可能占用一个或多个单元格。

梯形图中有两种基本的元件输入方法，以下分别介绍。

直接输入

在选中单元格处，直接用键盘输入字母，指令输入对话框即自动弹出，见图 4-8。在此对话框中输入完整的指令和参数后，点击确认按钮或回车，软件会对所输入的指令进行正确性校验，校验无误后即生成该指令所代表的图形元件，见图 4-9。

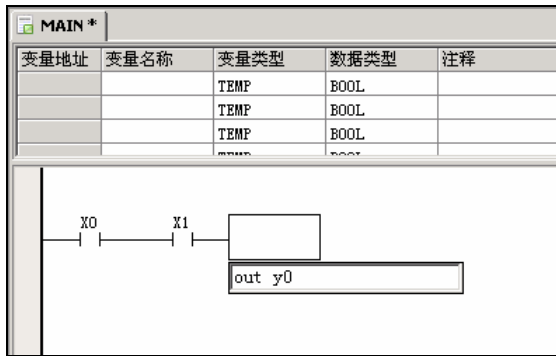


图4-8 指令输入对话框自动弹出

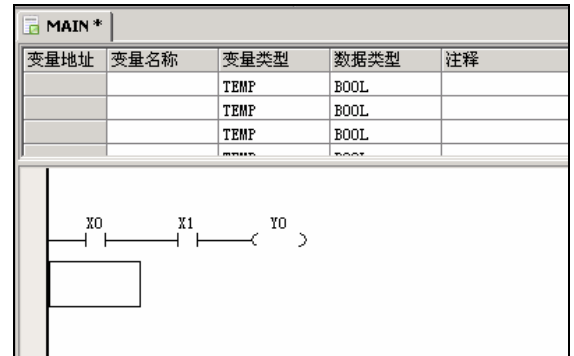


图4-9 生成的图形元件

向导式输入

在选中单元格后，还可以通过选择菜单或工具栏图标来输入相应的元件。所谓向导式，就是在选择了元件类型或指令名称后，软件会自动检测所需的参数数量及格式，然后自动调整界面输入格式，如图 4-10 和图 4-11。

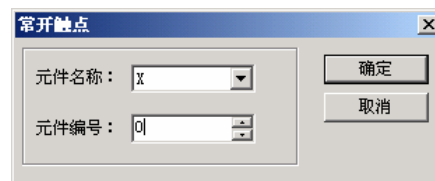


图4-10 常开触点向导式输入对话框

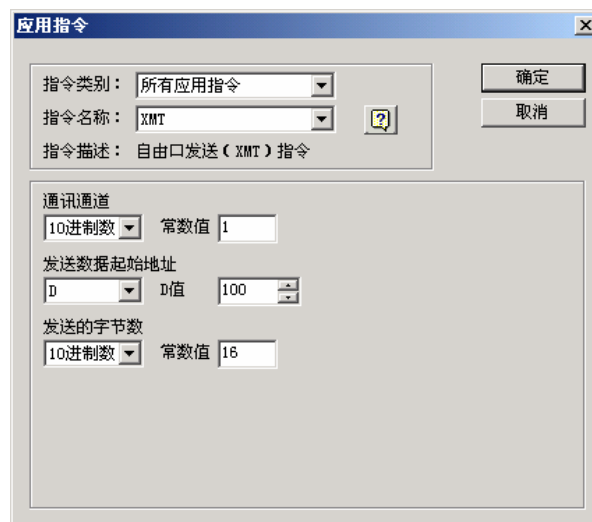


图4-11 应用指令向导式输入对话框

在上图中，选中指令后，点击右面的  按钮或按 F1 键可查看该指令的详细帮助。

由于 CALL 指令比较特殊，它的参数数量是可变的，而且只有选择了调用的子程序名称（CALL 指令的第一个参数）后才能确定其余的参数，因此针对 CALL 指令单独设计了另外一种对话框，如图 4-12 所示。

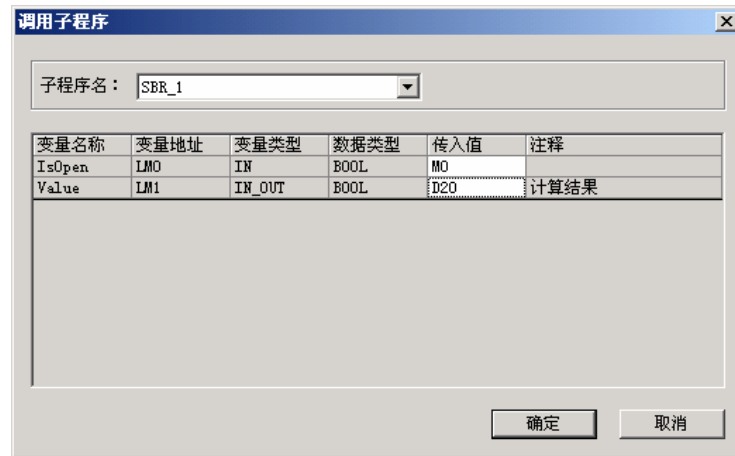
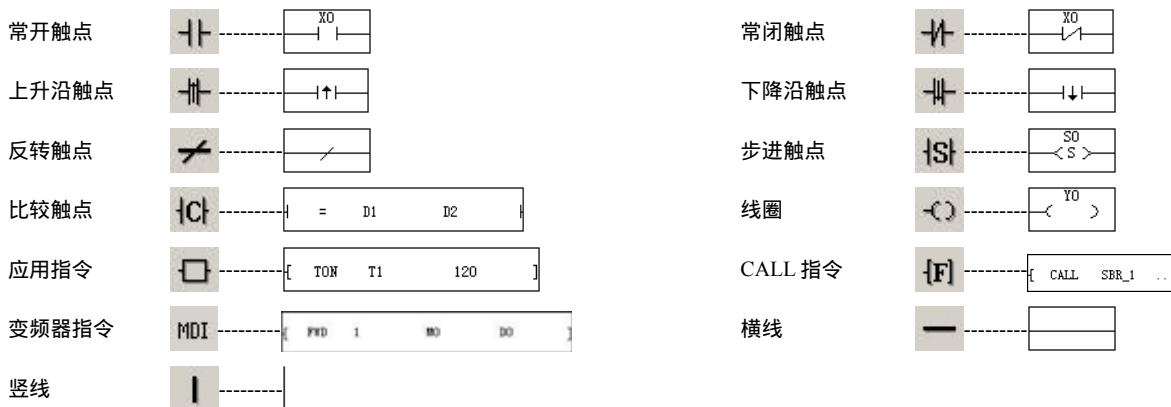


图4-12 CALL 指令对话框

在此对话框中，选择子程序名后，下面的表格中会显示该子程序所有的输入/输入输出/输出型参数的名称、地址、数据类型等信息，可以在“传入值”域输入想要传入的参数。确定后，如果输入无误，就会生成一个 CALL 指令元件。

各工具栏图标所对应的元件类型列举如下：



限制条件

1. 某些指令只能在指令列表编辑器中输入，而不能在梯形图中输入，比如 MPS、MPP 等。当在梯形图中直接输入这些指令时，将不会生成任何元件。
2. 由于线圈和应用指令代表了梯形图一行的结束，所以在这些元件后面，不能够再输入其他任何元件。
3. 由于竖线会占用两行的空间，因此在输入竖线时，如果在下一行的相应位置刚好是其他元件的中间部分，则此竖线不能输入。
4. 梯形图中最大输入列数为 32 列，最大输入行数为 4000 行。

快捷键

在梯形图中，按下快捷键 **Ctrl +** 可直接输入横线。按下快捷键 **Ctrl +** 可直接输入竖线。按下快捷键 **Ctrl + Shift +** 可以删除竖线。

输入模式

元件输入有两种模式：覆盖和插入。在覆盖模式下，新输入的元件会覆盖它所占用的单元格内的所有其他元件。而在插入模式下，原位置的元件及其相连的其他元件会先被向右移动若干个单元格，直至给新输入的元件腾出足够的空间。可以使用键盘上的 **Insert** 键在这两种模式之间切换。

4.2.4 如何修改梯形图元件

如何修改

需要修改梯形图中已经存在的元件时，用鼠标双击要修改的元件或者在该元件处按回车键，就会弹出指令输入对话框，同时所选元件对应的指令显示在输入框中。此时你可以对指令和参数进行修改。

常开触点 NO 与常闭触点 NC ，以及上升沿 Rising Edge 与下降沿 Falling Edge 之间，可以通过菜单中的**反转命令**或 **Ctrl+R** 键实现快速切换。

限制条件

1. 所有的触点元件不能修改成线圈或应用指令，反之亦然。
2. 占一个单元格的触点元件（常开、常闭、上升沿、下降沿、反转、步进）不能修改成比较触点，反之亦然。
3. CALL 指令和变频器连接指令双击时弹出的是向导输入对话框，因此不能修改为其他指令。
4. 修改后的指令，同样必须能够正确通过指令校验，才能够被软件接受。

4.2.5 如何删除梯形图元件

竖线的删除

需要删除梯形图中的某条竖线时，可以用鼠标点选竖线所属单元格，然后点击工具栏上的竖线删除按钮 Del 或使用 **Ctrl + Shift +** 组合键，可删除该竖线。

非竖线元件的删除

对于竖线之外的其他元件，删除时可以用鼠标点选元件所属单元格，然后点击工具栏上的删除按钮 X ，或者按 **Delete** 键，即可删除该元件。如果需要清除某一范围内的多个元件，可以通过拖动鼠标，选中要清除的区域，然后用与删除单个元件相同的操作，清除所选范围内的所有元件。请注意，当所选范围内包含竖线元件时，仅当此竖线所跨过的上下单元格都在选择范围内时，此竖线才会被删除。

4.2.6 如何输入注释

如何操作

为了使梯形图具有更好的可读性，可以在其中输入一些注释，这些注释称为块注释，每个块注释都会占用一整行的空间。

需要输入块注释时，首先选择一个空行，然后从右键菜单中选择**插入块注释**，在弹出的块注释对话框中输入注释文字并确认。软件会自动在所输入的文字两侧分别加上“/*”和“*/”，并以绿色显示。以上操作见图 4-13。

限制条件

由于块注释会占用整行空间，因此如果某行已经存在其他元件，则此行不能再输入块注释。同理，已经被块注释占用的行，也不能再输入其他任何元件。



图4-13 块注释输入过程

4.2.7 如何在梯形图中使用已定义的变量

变量可以在全局变量表（见全局变量的定义）和局部变量表（见局部变量的定义）中定义，正确定义的变量可以在梯形图中使用。需要使用某个地址时，可以用代表此地址的变量名来代替，以增强程序的可读性。图 4-14 为全局变量表中定义的变量示例。

变量地址	变量名称	变量类型	数据类型	注释
		BOOL	BOOL	
		BOOL	BOOL	
		BOOL	BOOL	

变量名	变量地址	注释
1 水压	X0	水压标记
2 计时	T0	计时30分钟
3 启动1	M0	
4 启动2	M1	
5 停止1	M2	
6 告警灯	Y0	水量告警灯
7 阀门	Y1	控制阀门

图4-14 全局变量表示例

符号编址

当使用了定义变量后，可以通过选择**符号编址**菜单在变量名和元件地址之间切换。图 4-15 分别展示了两显示模式下的同一梯形图程序。

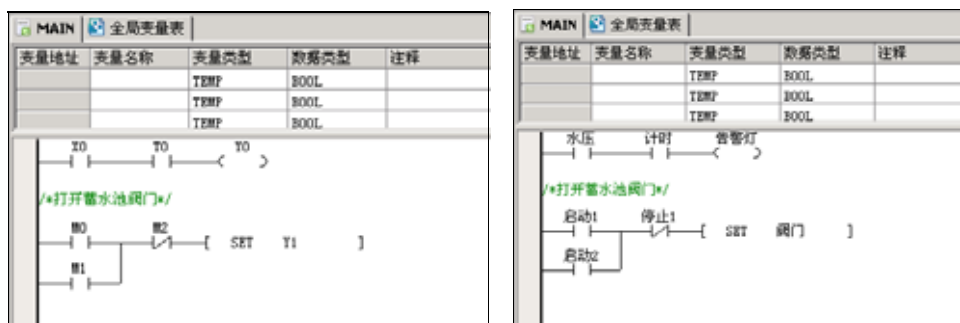


图4-15 两种显示模式下的同一梯形图程序

元件注释

可以通过选择**元件注释**菜单控制是否显示元件注释。图 4-16 是在显示元件注释时的梯形图程序。

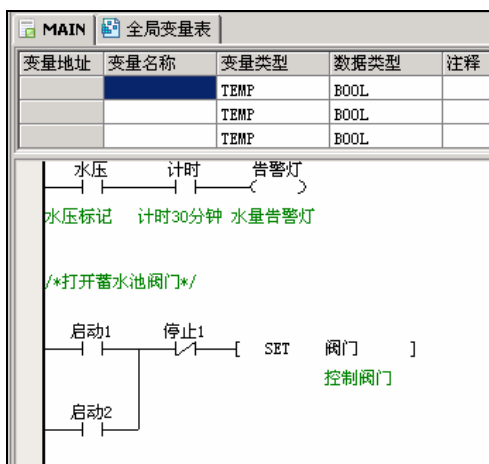


图4-16 显示元件注释时的梯形图程序

4.2.8 如何优化梯形图程序的显示

为使梯形图程序更加整齐和美观，软件提供了梯形图程序优化显示功能。此功能对当前打开的梯形图程序进行整理，即将所有输入元件尽量向左对齐并将输出元件和应用指令对齐到某一指定列。默认情况下，该指定列就是当前窗口的最右端的一列，如果需要对齐到其他列，点击**工具/系统选项**，调出系统选项对话框，在对话框中指定，如图 4-17 所示。需要使用优化显示功能时，请打开某个梯形图程序，然后点击**查看/优化显示**菜单即可。

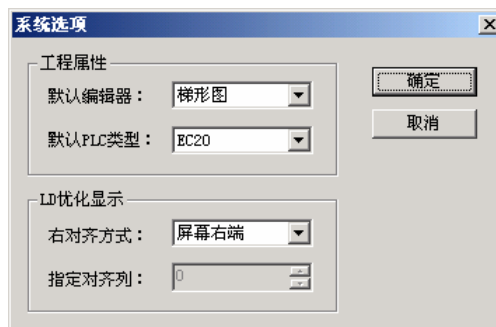


图4-17 系统选项对话框

4.2.9 如何使用梯形图的网络号

网络号可以使梯形图逻辑更加清晰，同时也有助于程序的定位，尤其是打印出来的程序。梯形图在以下情况下会显示网络号（网络号显示为“N”加网络ID）：

1. 编译单个梯形图程序成功后，当前梯形图会显示网络号。
2. 编译所有程序成功后，所有梯形图程序打开后都会显示网络号。

3. 从指令列表或顺序功能图转换为梯形图后会显示网络号。

4. 某个梯形图程序在最近一次保存时如果是显示网络号的，那么下一次打开时也会显示网络号（符号编址模式必须和保存时相同）。

在已显示网络号的梯形图上进行编辑操作后，会暂时隐藏网络号，直到下次被成功编译。显示网络号的梯形图示例如图 4-18。

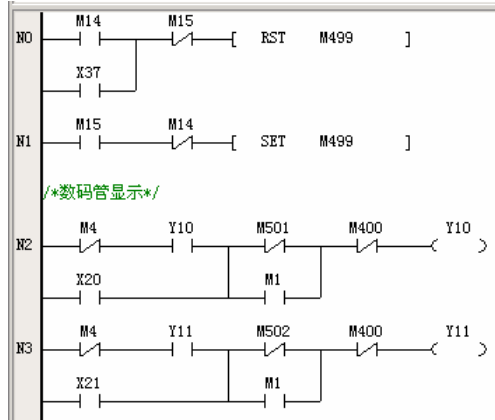


图4-18 显示网络号的梯形图

4.3 顺序功能图编辑器

4.3.1 预备知识

顺序功能图是根据机械设备的流程或者工序，将控制分成了多个步和步到步之间转换的一种语言。一个标准的顺序功能图由初始步、一般步、步间的转换条件、跳转和重置组成、每一步就是机械设备的处理工序，一个步中可以有内置梯形图，也就是这一步需要完成的处理工序。转换条件就是一个工序的完成和下一个工序的启动条件，它也需要内置梯形图来表示转换的条件。如图 4-19 所示。

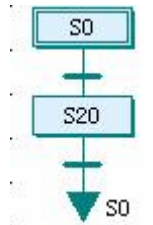


图4-19 顺序功能图示例



如上面的顺序功能图所示，初始状态是 S0。完成 S0 工序中的内置梯形图后，如果满足转换条件，就停止执行 S0，开始执行 S20 工序。如果不满足转换条件，则继续执行 S0 中的内置梯形图。在最后完成 S20 并满足 S20 跳转到 S0 的转移条件后，停止 S20 工序的执行，重新启动初始步 S0 的执行。通过跳转，可以让整个顺序工序循环不停的执行。

4.3.2 顺序功能图元件说明



步进符

步进符分为初始步进符和普通步进符。它们利用内部软元件状态（S）在顺控程序上面进行工序步进控制。



1. 初始步进符

一个顺序功能图网络的执行必须由初始步进符开始。初始步进符对应的图元符号是 ，可以单击  按钮在选中的位置添加初始步进符元件。初始步进符对应的 S 软元件地址范围是：S0 ~ S19。



2. 普通步进符

普通步进符对应的图元符号是 ，可以单击  按钮在选中的位置添加普通步进符元件。普通步进符对应的 S 软元件地址范围是地址大于 19 的所有 S 软元件。



梯形图块

梯形图块对应的图元符号是 ，可以单击  按钮在选中的位置添加梯形图块元件。梯形图块包含一个内置梯形图，它可以用来定义在顺序功能图网络之外执行的代码。



转移符

转移符对应的图元符号是 ，可以单击  按钮在选中的位置添加转移符元件。转移符中包含一个内置梯形图，可以在其中编写代码决定是否将与之连接的 S 软元件状态置位。

重置符

重置符对应的图元符号是 ，可以单击  按钮在选中的位置添加重置符元件。重置符的作用是将指定的 S 软元件复位。



跳转符

跳转符对应的图元符号是 ，可以单击  按钮在选中的位置添加跳转符元件。跳转符的作用是将指定的 S 软元件状态置位，使其包含的内置梯形图代码得以执行。



连接线

连接线将上面所述的顺序功能图元件按自上而下的顺序连接在一起。连接线除起连接作用之外，还包含选择分支、选择汇合、并行分支、并行汇合四种条件连接关系。



1. 选择分支

选择分支对应的图元符号是 ，可以单击  按钮在选中的位置添加选择分支连接线和两个转移符元件。选择分支必须从一个顺序功能图步进符开始，到转移符结束。它代表的意义是：如果顺序功能图步进符执行成功，则根据转移符中规定的条件，选择一条支路继续向下执行。



2. 选择汇合

选择汇合对应的图元符号是 ，可以单击  按钮在选中的位置添加两个转移符和选择汇合连接线。选择汇合必须从多个转移符开始，到一个普通步进符、跳转符、重置符结束。它代表的意义是：只要有一个转移符的条件满足，就运行下面连接的元件。

3. 并行分支


并行分支对应的图元符号是 ，可以单击  按钮在选中的位置添加一个转移符和并行分支连接线。并行分支必须从一个转移符开始，到普通步进符、跳转符、重置符结束。并行分支代表的意义是：如果转移符中的条件被满足，则与并行分支连接的元件将被一同执行。

4. 并行汇合

并行汇合对应的图元符号是 ，可以单击  按钮在选中的位置添加并行汇合连接线和两个转移符。并行汇合必须从普通步进符开始，到一个转移符结束。它代表的意义是：只要有一个普通步进符执行通过，就去执行转移符中的代码。

4.3.3 编辑顺序功能图程序

输入顺序功能图元件

所有的顺序功能图元件都可以在顺序功能图工具栏和顺序功能图菜单中找到对应的图标或菜单项。可以通过单击特定的图标输入所需元件，并在顺序功能图元件输入对话框中输入指定顺序功能图元件的属性。同时，在顺序功能图工具栏和顺序功能图菜单中还提供了添加连接线的快捷方式 ，可以根据程序的需要，在顺序功能图元件中间建立连接关系。

修改顺序功能图元件

针对不同的顺序功能图元件有不同的属性修改范围。初始步进符、不同步进符、跳转符、重置符元件可以修改的属性包括步号和注释。梯形图块元件可以修改的属性有元件名。转移符元件可以修改的属性有元件注释。

具体操作：双击指定的元件，界面显示顺序功能图元件修改对话框，在顺序功能图元件修改对话框中修改选中的元件的属性。对话框如图 4-20 所示。

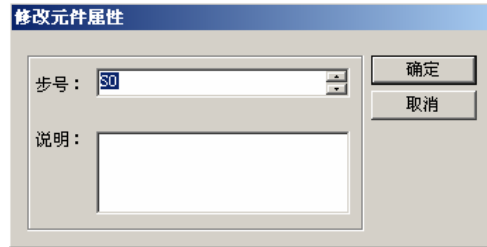


图4-20 顺序功能图元件属性修改对话框

修改顺序功能图元件内置梯形图

在梯形图块、步进符、转移符这三种元件中可以添加自定义的内置梯形图。选中相应元件，点击鼠标右键，点击**内置梯形图**，即可打开内置梯形图编辑器，见图 4-21。



图4-21 点击内置梯形

4.3.4 变量的使用

使用复制/粘贴功能

选中某个单元格或者拖动鼠标选中某个区域后，使用**编辑/复制**菜单或者按 **CTRL+C** 组合键，可以将所选的元件拷贝到系统剪贴板，然后使用**编辑/粘贴**菜单或者按 **CTRL+V** 组合键可以将这些元件粘贴到其他位置或其他顺序功能图中。

添加/删除行

1. 添加行

行插入功能指在顺序功能图中插入一行空白行。

具体操作：先要在要插入的行上选中任一单元格，点击鼠标右键，选择**行插入**。软件会将当前行及下面所有行的元件向下移动一行，从而在当前行腾出一个空行来。

2. 删除行

行删除功能指删除顺序功能图中的特定行。

具体操作：先要在要删除的行上选中任一单元格，点击鼠标右键，选择**行删除**，软件会将当前行清空，并将下面所有行的元件向上移动一行。

在顺序功能图程序中，只能使用全局变量表中定义的 S 软元件对应的变量。

顺序功能图提供符号和元件地址两种编址方式。用鼠标点击**查看**菜单中的**符号编址**菜单项可以在两种编址方式之间切换。**符号编址**菜单项前有选中标记时，表示当前所有定义了变量的元件地址都会使用变量名来表示。**符号编址**菜单项前没有选中标记时，则无论是否定义了变量，都显示为元件地址，输入变量名也会自动转换为元件地址。

4.4 指令列表编辑器

4.4.1 指令编辑

指令输入

指令列表编辑器是一个文本编辑器，用于输入指令、操作数和注释。在编辑器中输入需要的指令后，软件会自动对齐指令、操作数和注释。指令关键字用蓝色显示。操作数中，局部变量、全局变量、变频器别名用紫色显示。常数用灰色显示。元件用黑色显示。注释用绿色显示。错误行用红色显示。如图 4-22 所示。

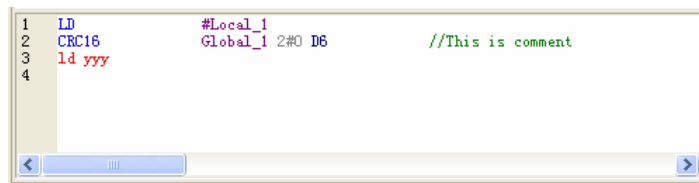


图4-22 指令列表编辑器

在编辑器的任意位置输入双斜线“//”表示当前行后面的内容是注释。

拷贝/剪切/粘贴

使用鼠标在文本编辑器中拖动选择要拷贝的文字内容，当文字显示高亮状态时表示该部分内容已经被选中，使用快捷键 **Ctrl+C** 或者点击**编辑/拷贝**菜单项，可以将选中的内容拷贝到 Windows 剪贴板中。被拷贝的内容可以在任何的标准文本编辑器中粘贴，包括 Windows 写字板、记事本或者 Word 等。

剪切的操作和拷贝非常类似，不同的是被剪切的内容会从编辑器中被删除，使用快捷键 **Ctrl+X** 或者点击**编辑/剪切**菜单项，可以将选中的内容剪切到 Windows 剪贴板中。

拷贝或者剪切完毕后，用鼠标或者键盘将当前光标定位到想要粘贴的位置，使用快捷键 **Ctrl+V** 或者点击**编辑/粘贴**菜单项，即可将被拷贝或者剪切的内容粘贴到光标所在的位置。

正确性校验

输入每一行指令后，如果光标切换到其他行，则会对离开的行进行正确性校验，如果有语法错误，软件会将有错误的文本行显示为红色。当鼠标悬浮在有错误的文本行时，软件会用工具提示的方式提示错误信息。只有校验正确的语句才会格式化显示，否则会用红色突出有语法错误的行。在一次粘贴多行文本内容到编辑器中时，编辑器会对每一行作语法检查。

有语法错误时，请将鼠标放在有错误的位置，软件显示如图 4-23 所示。

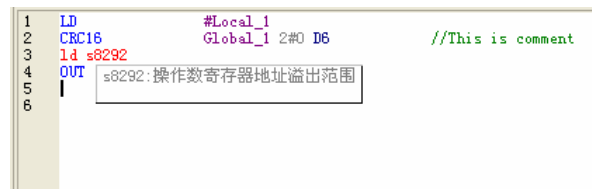


图4-23 正确性校验显示

在图 4-23 中，LD 指令的操作数 S8292 超过了 S 元件的最大范围，因此认为是非法的。

4.4.2 变量的使用

在软件中，可以定义全局变量(见**全局变量的定义**)和局部变量（见**局部变量的定义**），定义好的变量可以在指令列表中使用。在输入元件地址时，如果该元件地址定义了变量，则可以用变量名代替元件地址作为操作数输入，如果使用的是局部变量，软件自动在变量名前加上“#”号和全局变量区别。





指令列表提供符号和元件地址两种编址方式，用鼠标点击**查看/符号编址**菜单项可以在两种编址方式之间切换。**符号编址**菜单项前有选中标记时，表示当前所有定义了变量的元件地址都会使用变量名来表示。**符号编址**菜单项前没有选中标记时，则无论是否定义了变量，都显示为元件地址，输入变量名也会自动转换为元件地址。

4.4.3 行号、标签

行号

行号主要用来对程序的快速定位以及提高程序的可读性。点击**查看/行号**可以进行行号显示和行号隐藏之间的切换。

标签

标签是类似书签功能的标记，它可以在指令列表程序中的任意行加入，使用工具栏上的按钮，或者点击**编辑**菜单中的**添加/删除标签**可以在光标所在行添加标签或者将标签删除。点击或者按钮，可以迅速定位到光标位置的下一个标签或者上一个标签所在的行。点击按钮将清除所有已经存在的标签。

4.5 全局变量表

4.5.1 全局变量的定义

全局变量是为 PLC 的某个地址定义的有意义的符号名称。该符号名称在整个工程范围内都可被访问到，相当于使用该变量对应元件，全局变量在全局变量表中定义。

全局变量声明包含变量名称、变量地址、变量注释三个属性。

全局变量的定义规则为：由 A~Z、a~z、0~9、下划线、汉字混合组合而成。变量名称不能以数字开始，也不能是单独的数字。名称不区分大小写，长度不能超过 16 个英文字符，不能使用元件类型字母+数字作为程序和变量名称。名称中不能包含空格，不能使用和关键字相同的名称，保留的关键字包括：基本数据类型名称、指令名称、指令表语言中的运算符。

全局变量个数不能超过 500 个。

4.5.2 编辑全局变量表

双击工程管理窗口的全局变量表可以打开全局变量表，如下图所示。



图4-24 全局变量表

可以在全局变量表中查看和编辑变量名、元件地址和元件注释。在编辑过程中可以对其进行全局变量表的合法性检查：

- 如果元件地址错误，软件使用不同颜色显示该元件地址。
- 符号表中输入的变量名称必须唯一，对于重复的变量名，软件用不同的颜色显示变量名。
- 元件注释最大长度为 20 个字节。

使用鼠标单击符号表格的列头，可以对变量名、元件地址和元件注释中的任何一列升序或者降序排列。

4.5.3 导入导出全局变量表

全局变量表可以与 EXCEL 表格进行数据交换，可以将用 EXCEL 文件编辑的数据导入到全局变量表中，也可以将全局变量表导出到 EXCEL 文件中。与全局变量表交换的 EXCEL 文件的格式如图 4-25 所示。其中，第 A、B、C 列分别对应变量名、变量地址、注释。

	A	B	C
1	开关1	X1	蓝色开关
2	开关2	X2	红色开关
3			

图4-25 与全局变量表交换的 EXCEL 文件格式

导入全局变量表

打开全局变量表，选择**文件/全局变量表导入**菜单，将弹出 Windows 标准的文件打开对话框，选择要导入的文件，点击**打开**按钮即可将 EXCEL 文件导入全局变量表。

全局变量表导出

打开全局变量表，选择**文件/全局变量表导出**菜单将弹出 Windows 标准的文件另存对话框，输入文件名，点击**保存**按钮，全局变量表将导出为 EXCEL 文件。

4.5.4 全局变量的使用

定义的全局变量可以在各种程序中使用，如果定义了相同名称的局部变量或者全局变量，变量名前加上“#”表示该变量是使用一个局部变量，否则是使用全局变量。

4.6 局部变量表

4.6.1 局部变量的定义

对于每个程序，根据类别不同可以定义该子程序使用的输入参数，输出参数和局部变量，对参数和变量的说明在程序的局部变量表中定义。

局部变量是为 PLC 的某个地址定义的某个有意义的符号名称，该变量具有局部范围，只在某个程序单元内有效，局部变量在局部变量表中定义。

局部变量声明包含变量地址、变量名称、变量类型、数据类型、变量注释五个属性，变量类型有三种类型（IN、IN_OUT、OUT、TEMP）其值是不可以修改的。

局部变量的命名规则与全局变量完全相同。

对于主程序、中断子程序，可以在每个程序的局部变量表中定义局部变量。局部变量只在定义它的程序中有效，程序运行完毕后从内存中释放。局部变量在局部变量表中的类型关键字为 TEMP，一个程序中定义的局部变量个数不能超过 64 个。

对于子程序，可以在局部变量表中除了可以定义局部变量外，还可以定义该子程序的调用参数和返回参数。

输入参数在局部变量表中的类型关键字为 IN。

输出参数在局部变量表中的类型关键字为 OUT。

输入输出参数在局部变量表中的类型关键字为 IN_OUT。

三种参数定义同时不能超过 16 个。

可以在局部变量表中查看变量名称、数据类型和变量注释，还可以修改变量名称和变量注释。数据类型支持 BOOL、WORD、DWORD、INT、DINT 和 REAL。

变量注释最大长度为 20 个字节。

4.6.2 编辑局部变量表

每个程序都有自己的局部变量表，局部变量表是程序的一个组成部分，创建了程序（包括主程序、子程序、中断子程序）后也就创建了相应的局部变量表。编辑窗口的上半部分显示局部变量表，下半部分显示程序，如图 4-26 所示。

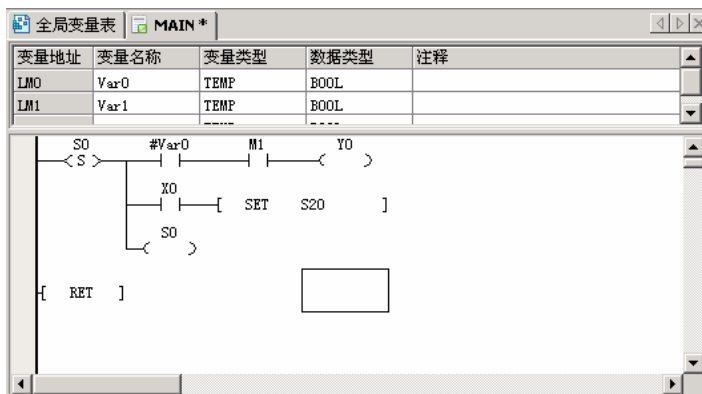


图4-26 编辑局部变量表

可以在局部变量表中查看和编辑变量名称、变量类型、数据类型和注释。变量地址自动分配，只能查看不可编辑。

在编辑局部变量表的过程中可以对其进行合法性检查：

- 对输入不合法的变量名，软件用不同的颜色给出提示。
- 局部变量表中所有输入的变量名称和参数名必须唯一。输入确认后，对于重复的变量名和参数名，软件给出提示并用不同的颜色显示变量名。

变量地址的自动分配：在定义输入/输出参数和局部变量时，一旦确定变量的存储类型，立刻为该变量或者参数在 LM 或者 V 寄存器中分配具体的地址。

对位变量的分配使用 LM 寄存器，对字和双字变量的分配使用 V 寄存器，其中双字变量的寄存器地址显示为第一个字的地址。当修改某个变量的数据类型或者删除某个变量导致占用的字节数改变时，该变量的后续变量地址也会重新分配。

4.6.3 局部变量的使用

定义的局部变量和参数可以在程序中使用。在变量名前加上“#”表示该变量是使用一个局部变量，否则是在使用全局变量。对于一个局部变量且没有相同的全局变量，对指令格式化后将自动在变量名前加上“#”。

如果当前处于元件地址编程模式，那么不显示变量名，并且地址重新分配时不调整代码中的地址。

4.7 交叉引用表

4.7.1 交叉引用表的定义和使用

交叉引用表用于记录元件在各个程序中的使用情况，可以通过双击交叉引用表中的记录定位该交叉引用信息所在的程序和行列。

交叉引用表不可以编辑，只可以进行查找和定位，要使用查找和定位功能，可以在编辑菜单下选择查找和转至子菜单，也可以点击 按钮。

程序在编译或者转换完成后产生交叉引用表，在下次编译成功前保持不变，退出工程后交叉引用表内容保存到文件中。交叉引用表包含的信息如下：

元素：如果元件在全局变量表中定义为变量，且当前显示模式设置为显示变量模式，则显示元件变量名，否则显示元件地址。

块：元件所在的程序。

位置：元件所在程序的行号和列号。

上下文：对元件的使用方式。

4.7.2 切换编址方式

如果在交叉引用表中存在一个或者多个全局变量表或局部变量表中定义的变量，则既可以将该行数据显示为符号也可以显示为地址，你可以通过选择**符号编址**菜单在变量名和元件地址之间切换。地址编址模式下所有元件显示为元件地址，符号编址模式下在全局变量表或者局部变量表中定义的地址显示为符号名称，没有定义的显示为元件地址。

4.7.3 元件浏览表的定义和使用

交叉引用表仅显示已用元件的信息，元件浏览表则以表格方式显示系统支持的所有软元件的使用情况。在元件浏览表中双击某个已用元件时，如果该元件被使用，则定位到使用该元件的程序位置。如果这个元件被多次引用，将定位到第一次使用该元件的位置。元件浏览表如图 4-27 所示。

	0	1	2	3	4	5	6	7	8	9
DO	×					×				×
M10			×							
X0		×							无	无
Y0		×							无	无

图4-27 元件浏览表

“×”表示元件已经被使用，空白表示没有被使用，“无”表示该元件不存在。上图中元件浏览表中已被使用的元件有 D0 (0+0)，D5 (0+5)，D9 (0+9)，M12 (10+2)，X1 (0+1)，Y1 (0+1)，其中 X8 (0+8)，X9 (0+9)，Y8 (0+8)，Y9 (0+9) 元件不存在（因为 X、Y 元件是按照 8 进制编址的）。

双击“×”可以定位到使用该元件的程序位置。

4.7.4 元件浏览表的过滤

元件浏览表除了提供快速定位功能外还提供元件过滤功能。元件过滤，是指用户可以指定哪类元件要显示、哪类元件要隐藏。元件类别按照 X、Y、M、D 等元件名称区分，在要显示的元件类中，还可以指定显示固定范围的元件，比如只显示 1000 ~ 1500 之间的 D 元件。

使用方法：点击鼠标右键，然后点击弹出菜单的**元件过滤**子菜单，也可以点击菜单**交叉引用表/元件过滤**，将弹出如下图的过滤对话框。

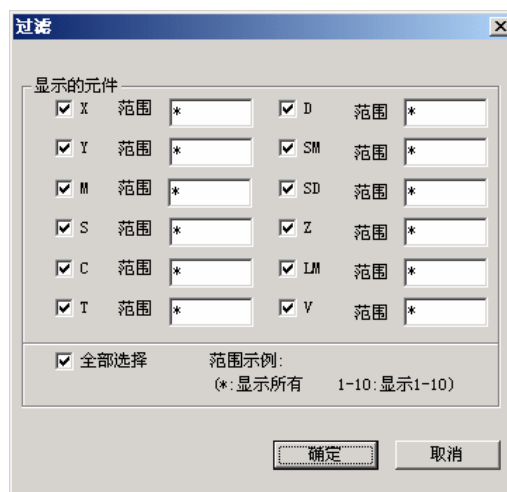


图4-28 过滤对话框

在上面对话框中，范围处输入“*”表示该元件的所有地址都显示，如需指定显示具体的某个范围，可以按照“m-n”格式输入，表示显示从 m 到 n 范围的地址。需要注意的是，交叉引用表和元件浏览表都是在某个程序被保存后才更新相应程序的信息。

第五章 与 PLC 通讯

5.1 通过串口连接

在开始与 PLC 硬件进行通讯前，需要建立和 PLC 串口的通讯连接。首先将串口线的两端连接到计算机和可编程控制器上，然后启动 ControlStar 应用程序。如果此时不能和 PLC 通讯或者需要使用不同的通讯速率，需要重新设置串口通讯参数。选择工具/串口/串口配置菜单，会弹出如图 5-1 所示的通讯设置窗口。

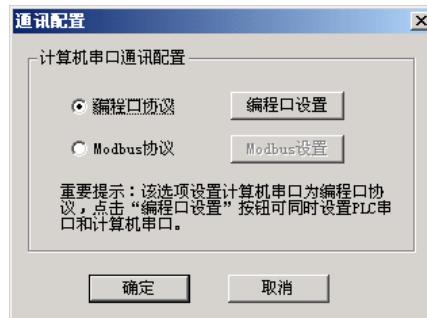


图5-1 通讯配置窗口

首先要确定 PLC 串口的协议，计算机串口的设置必须与 PLC 串口一致。如果 PLC 串口是编程口协议，则选择编程口协议进行配置。如果是 Modbus 协议则选择 Modbus 设置进行配置。配置编程口协议时将弹出如图 5-2 所示对话框。



图5-2 编程口设置对话框

在串口下拉框中选择计算机实际连接的串口，选择通讯波特率后，点击**确定**按钮保存设置结果。对于编程口协议设置，可以同时设置 PLC 串口和计算机串口的波特率，也可以单独设置计算机串口波特率。配置 Modbus 协议时将弹出如图 5-3 所示对话框。



图5-3 ModBus 协议配置对话框

在串口下拉框中选择计算机实际连接的串口、输入波特率、PLC 站号等内容后，点击**确定**按钮保存设置结果。

对于 Modbus 协议设置，只能设置计算机串口而不能设置 PLC 串口。如需将 PLC 串口设置为 Modbus 协议，则应在系统块中设置，然后下载运行才生效。此外，如果 Modbus 协议通讯时经常出现超时错误，可能是主模式超时时间设置太短，可以通过提高波特率或者增加超时时间的方法解决。

串口线连接和通讯设置完成后，可以测试计算机是否可以和 PLC 正常通讯，点击 **PLC/PLC 信息** 菜单项。如果通讯正常，会弹出一个窗口显示当前连接的 PLC 的各种信息，如果多次使用本功能都提示“命令超时”，则说明串口连接或者设置可能不正确，请重新检查硬件连接和通讯设置内容。

5.2 通过 Modem 连接

可以通过 Modem 实现 ControlStar 与 PLC 的远程连接。使用一台 Modem 连接计算机的串口，另一台 Modem 连接 PLC 的串口，当两台 Modem 通过电话线互相连通后，就可以使用 ControlStar 远程控制 PLC 了。

PLC 端连接的 Modem 的串口线需要做一些特殊处理。如果使用 25 针，需要将 4、5 针短接，6、20 针短接。如果使用 9 针，需要将 7、8 短接，4、6 针短接。以上硬件连接无误后，就可以调用 ControlStar 的拨号连接了。

选择**工具/Modem/拨号**菜单，将调用拨号程序，弹出图 5-4 所示对话框。



图5-4 Modem 拨号连接框

输入连接 PLC 端的电话号码后，点击**拨号**按钮，随后将弹出如下信息框显示拨号过程。

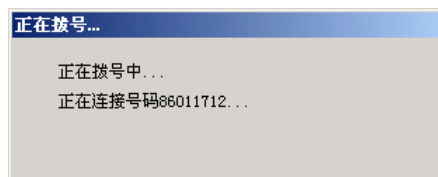


图5-5 拨号信息提示框

拨号成功后，将在状态栏上显示 Modem 已连接的图标。此时，就可以控制远端的 PLC 了。与 PLC 通讯完成后，可以使用**工具/Modem/断开**菜单，断开 Modem 连接。

在此需要注意的是，要使 PLC 端的 Modem 可以被正常拨通，必须保证 PLC 串口波特率与 Modem 设置的波特率一致。可以按照以下步骤确保二者波特率的一致性，首先使用 ControlStar 先读取 PLC 串口波特率（使用 **PLC/PLC 串口波特率** 菜单），然后将计算机串口设为与 PLC 波特率一致，再使用 ControlStar 初始化 Modem（使用 **工具/Modem/初始化** 菜单）。

使用 Modem 连接 PLC 时，如果串口波特率为 19200，误码率会比较高，因此可能经常会出现通讯命令执行失败或超时的情况，因此，建议使用 9600 波特率。

5.3 断开通讯连接

与 PLC 进行通讯时，ControlStar 将占用计算机的串口，当其他应用程序需要使用串口进行通讯时，会存在串口资源的争用。断开通讯连接功能就是放弃当前占用的串口，这样可以在不关闭 ControlStar 的情况下释放串口给其他应用程序使用。

选择**工具/串口/关闭串口**菜单，可断开通讯连接。需要注意的是，不能在监控状态或在线修改状态关闭串口。

第六章 PLC 设置

6.1 系统块

6.1.1 配置系统块

通过系统块可以对掉电保存的元件、时间、通讯、输入点等进行设置。设置完以后就可以将系统块下载至 PLC 中，这些设置将会生效。

在工程管理器窗口中双击选择**系统块**，将会弹出系统块对话框，如图 6-1 所示。不同的 PLC 型号，某些配置页面可能由于硬件不支持而被隐藏。

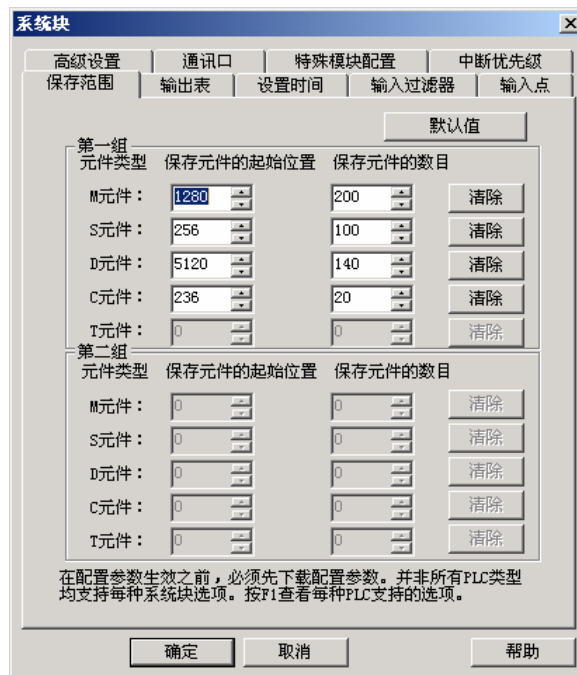


图6-1 系统块对话框

配置好系统块以后，选择 **PLC/全部编译** 菜单，项目的系统块文件被编译。选择 **PLC/下载** 菜单，会弹出消息框提示是否需要重新编译，选择**是**，则系统块信息会被编译并下载。

6.1.2 系统块保存范围

双击打开图 6-1 所示的系统块对话框以后，在对话框的第一页可以看到**保存范围**，用以配置保存元件地址范围。需要注意的是，可保存的元件地址范围和数量会因 PLC 型号的不同而异。最多可以定义两组保存范围（某些型号的 PLC 只支持一组），最终的保存范围是定义的两组保存范围的并集。如图 6-2 所示。

掉电时 PLC 将按照图 6-2 中定义的范围，将要保持的元件保存到掉电备份文件中。

上电时 PLC 检查掉电备份文件是否正确。如果正确则将保持元件的值恢复成上次掉电时的保存值。如果备份文件丢失或错误，系统将所有元件值清零。

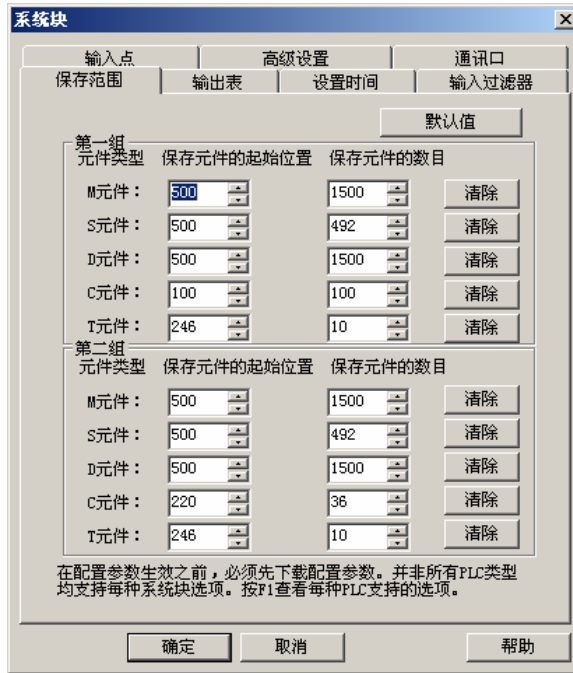


图6-2 保存范围

6.1.3 系统块输出表

单击**输出表**标签，可以设置 PLC 停机时的 I/O 点输出状态。如图 6-3 所示。

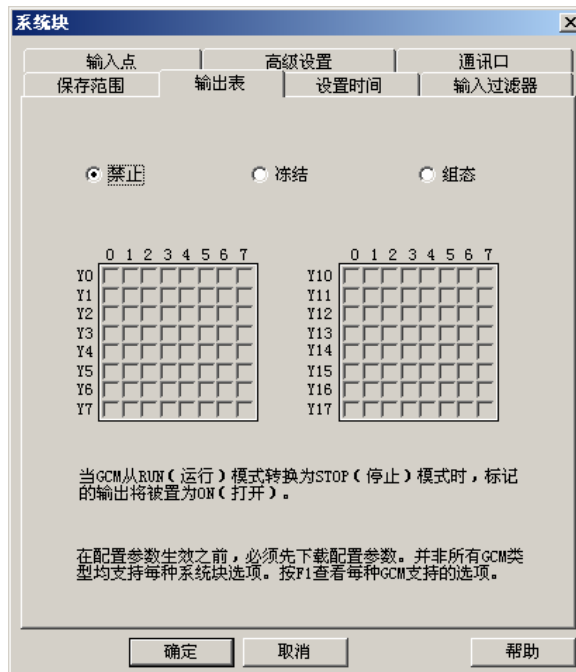


图6-3 输出表

功能说明

设置停止状态时的输出点组态，当 CPU 处于停止状态时，输出点有三种选择：

1. 保持停止前的输出点状态
2. 禁止所有的输出点
3. 按输出点组态表（Y00 ~ Y177 可组态）设置输出状态

输出点组态表

输出点组态表见表 6-1。其中 表示输出 ON。

表6-1 输出点组态表

	0	1	2	3	4	5	6	7
Y000								
Y010								
Y020								
Y030								
Y040								
Y050								
Y060								
Y070								
Y100								
Y110								
Y120								
Y130								
Y140								
Y150								
Y160								
Y170								

禁止

选择**禁止**选项，PLC 将在停机时禁止所有的输出点，在运行到停止的时候生效。

冻结（保持原状态输出）

选择**冻结**选项，PLC 将在停机时将所有数字输出冻结在最后的狀態中。

组态（按输出组态输出）

选择**组态**选项，PLC 将在停机时设置输出点为已知状态。所有输出点默认状态为“关闭（0）”的状态。如图 6-4 所示。

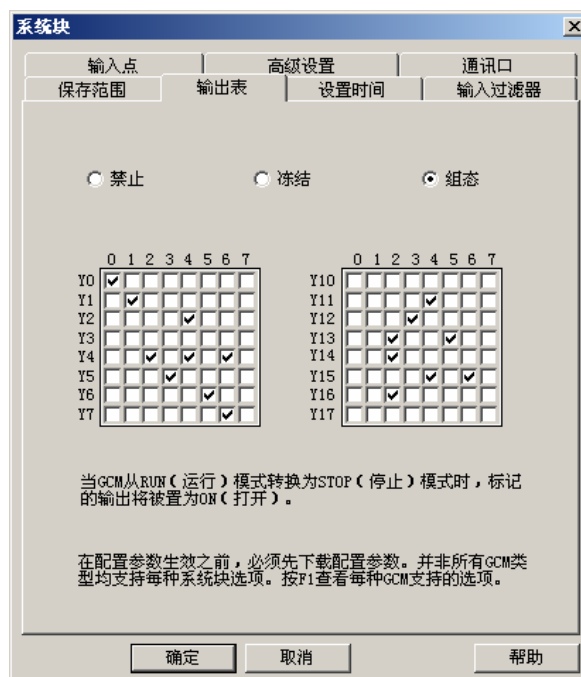


图6-4 组态选项

6.1.4 系统块设置时间

系统块设置时间界面见图 6-5。

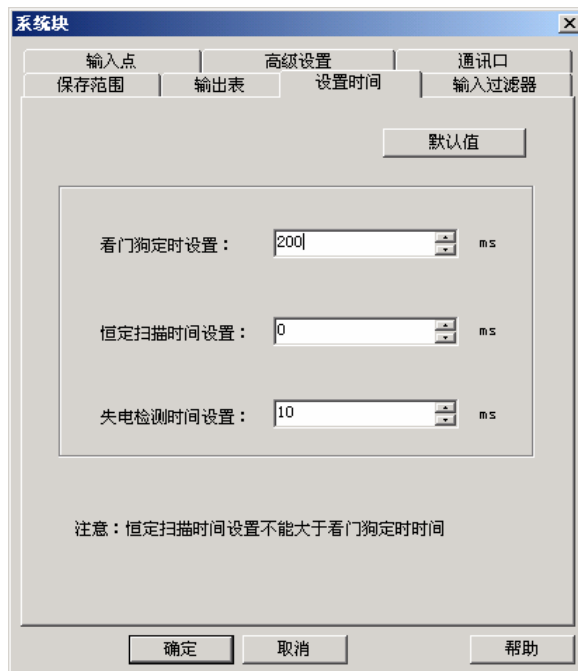


图6-5 系统块设置时间

看门狗定时器

看门狗定时器用于设置用户程序运行看门狗时间。看门狗时间是指允许用户程序运行的最大时间，当用户程序的实际执行时间超过看门狗时间时，PLC 会停止用户程序，点亮程序警告灯（红色），输出按系统配置进行输出。看门狗定时器时间可设定范围为 100ms ~ 1000ms，默认值为 200ms。

恒定扫描时间

恒定扫描时间是指系统在恒定时间扫描寄存器的时间。读取系统的恒定时间扫描设定寄存器，在恒定时间内对用户程序只作一次扫描。恒定时间可设定范围为 0ms ~ 1000ms。默认值为 0ms。恒定扫描时间设置不能大于看门狗定时时间。

失电检测时间

当瞬时失电时间大于设定的失电检测时间时，PLC 会进入停止状态。系统按照配置的停电保持元件范围进行元件值的保存。失电检测时间可设定范围为 10ms ~ 100ms。默认值为 10ms。对不支持失电检测时间设置的 PLC 型号，该设置项不可见。

6.1.5 系统块输入过滤器

单击**输入过滤器**标签，可以为 PLC 输入点设置一个延迟时间，过滤掉外部的干扰。通过设置输入延迟时间，可以过滤数字输入信号。可以利用编程软件设置 X0 ~ X17 输入点数字滤波常数。数字滤波常数的范围和默认值根据不同的 PLC 型号而异。见图 6-6。

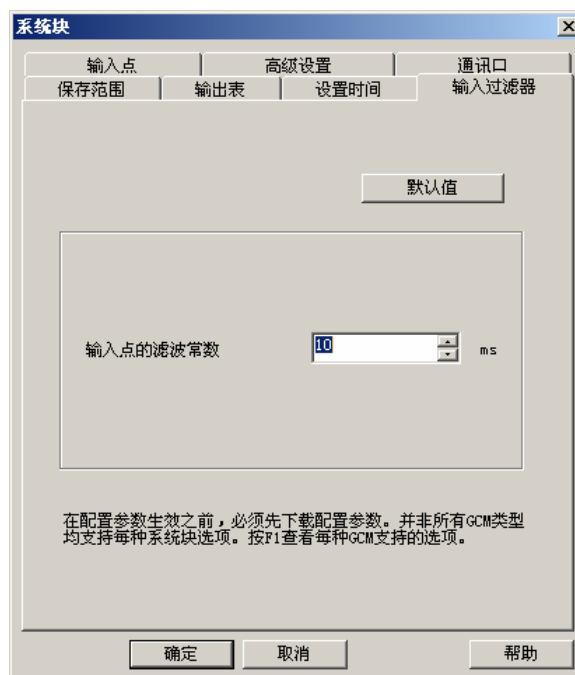


图6-6 输入过滤器界面

6.1.6 系统块输入点

输入点启动

指定在 X0 ~ X17 中某个输入点为强制 RUN 输入点，当系统处于 STOP 状态下，检测该点为 ON 时将系统状态由停止切换到运行状态，见图 6-7。

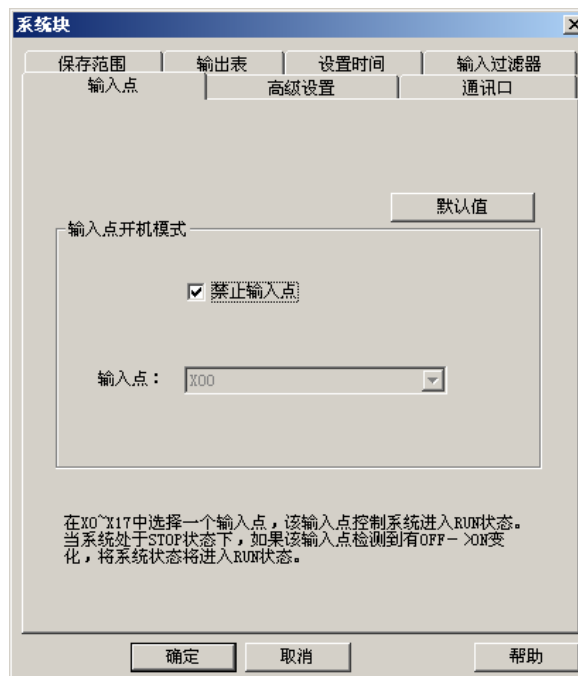


图6-7 输入点界面

禁止输入点

选择单选框**禁止输入点**，禁止输入点启动。

6.1.7 系统块高级设置

系统块高级设置用于配置一些诸如数据块有效、元件值保持、无电池模式等设置，见图 6-8。

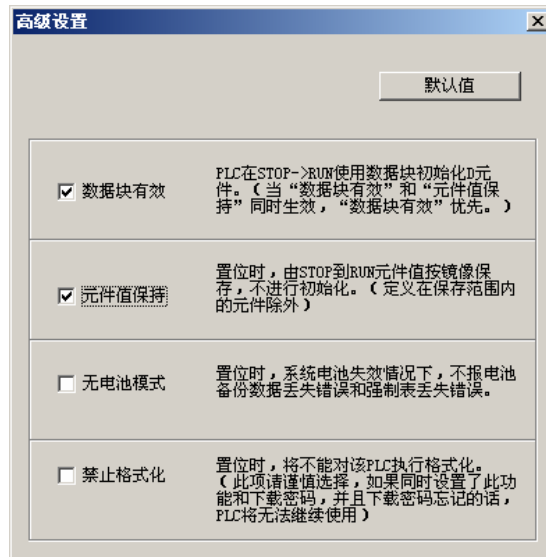


图6-8 系统块高级设置界面

数据块有效：如果选择了该项，PLC 在 STOP 到 RUN 状态时使用数据块初始化 D 元件。

元件值保持：如果选择了该项，由 STOP 到 RUN 状态时，元件值按镜像保存，不进行初始化。如果同时选择了**数据块有效**和**元件值保持**，**数据块有效**优先。

无电池模式：如果选择了该项，系统电池失效情况下，不报电池备份数据丢失错误和强制表丢失错误。对于没有电池的 PLC，此项设置将无法选择。

禁止格式化：如果选择了该项，将不能对该 PLC 执行格式化。此项请谨慎选择，如果同时设置了此功能和下载密码，并且下载密码忘记的话，PLC 将无法继续使用。对于不支持该功能的 PLC，此项设置将无法选择。

6.1.8 通讯口设置

通过此页面可以对软件的通讯口进行设置，包括使用什么样的通讯协议、如何支持 Modbus 协议、ECbus 协议、自由口协议、编程口协议等。见图 6-9。

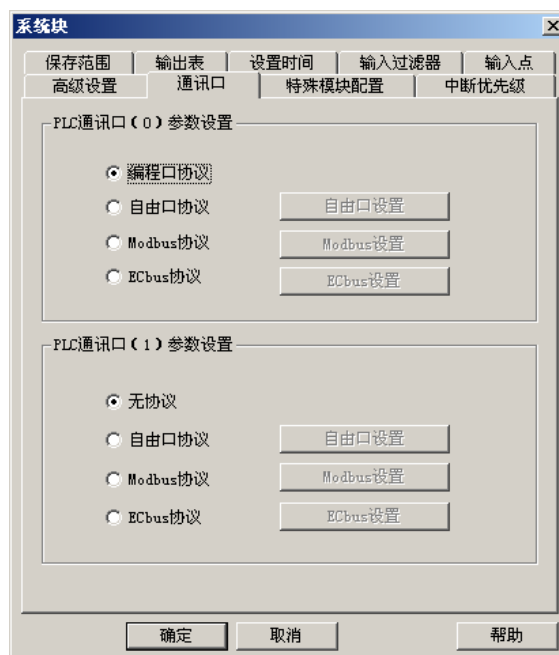


图6-9 系统块通信口设置

系统对话框中的通讯口 0 默认是编程口协议。通讯口 1 的默认是无协议的。可以分别对通讯口 0 和通讯口 1 进行设置。

自由口协议

自由口协议是自定义数据文件格式的通讯方式。自由端口通讯模式支持 ASCII 和二进制两种数据格式。只有在 PLC 位于运行 (RUN) 模式时才能使用自由端口通讯。在自由端口模式下,无法与编程设备通讯。STOP 状态下,串口 0 只能用于编程口通讯。

点击图 6-9 中的**自由口设置**按钮将弹出自由口协议设置界面,见图 6-10。在此界面中,可以设定传输的信息中信息字符的起始位,字符间超时时间、接收字符的最大值。接收指令允许接收一个或多个字符的缓冲器,最多可达 255 个字符。串口按照接收开始条件,开始自由口接收。其中的**有效字节**设置对某些型号的 PLC 不可见。



图6-10 自由口协议界面

Modbus 协议

通过 MODBUS 协议 MODBUS 的主站可以与从站（包括变频器）进行通讯，通过 MODBUS 协议提供的 MODBUS 功能码，发送对从站的控制帧。

1. 通讯口【0】

可以设置为 MODBUS 从站。通过 MODBUS 协议提供的功能码进行通讯，对主站的请求进行回应，判断是本地的报文，进行处理，同时统计接收的报文信息，供 MODBUS 主站查询。点击图 6-9 中的上面一个 **MODBUS 设置**按钮将弹出图 6-11 所示的界面来设置 Modbus 协议参数。

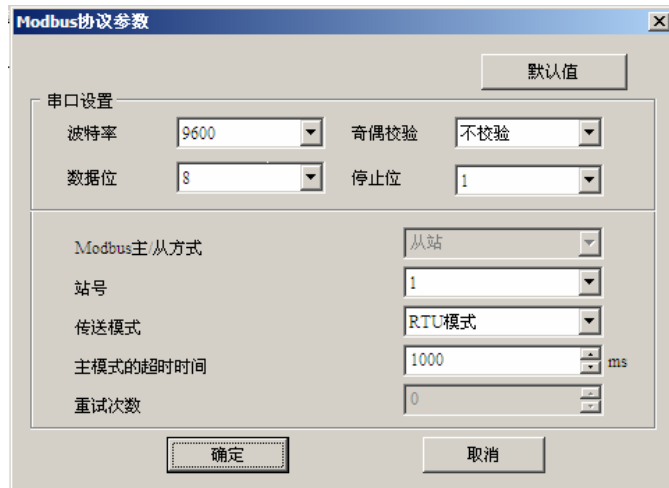


图6-11 Modbus 协议参数设置界面

2. 通讯口【1】

串口 1 可以设为从站和主站，最多支持 31 个从站，1~31 可设。主站与从站站号不能相同。等待接收，在超时时间（可以通过系统块和 SD 设置）内接收到数据，进行地址、CRC（RTU 模式）或 LRC（ASCII 模式）校验、字符间超时（RTU 模式）的判断，从而判断通讯的正确与否。

点击图 6-9 中的下面一个 MODBUS 设置按钮，弹出的界面与图 6-11 完全相同。设置内容见图 6-11。

ECbus 协议

ECbus 网络是建立在多个 PLC 之间的对等网络。网络中的每一个 PLC 都可以通过共享数据区自动与其他 PLC 进行数据交换。对于不支持 ECbus 协议的 PLC，此选项会被禁止。

ECbus 协议可设置内容见图 6-12。模式可设置为单层或多层，刷新模式的可选范围会根据模式设置而变化。

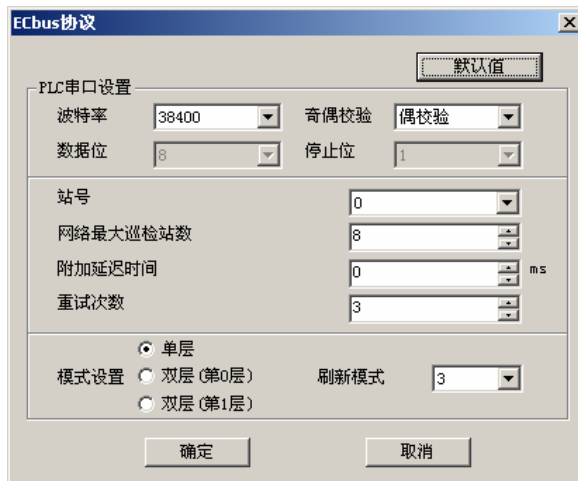


图6-12 Ecbus 协议设置界面

6.1.9 特殊模块配置

在系统块界面中，点击**特殊模块配置**项，即出现图 6-13 所示页面。通过此页面可以对 PLC 主模块所带的特殊模块类型以及每个特殊模块的数据区进行设置。对不支持此项设置的 PLC 主模块，此页面不可见。

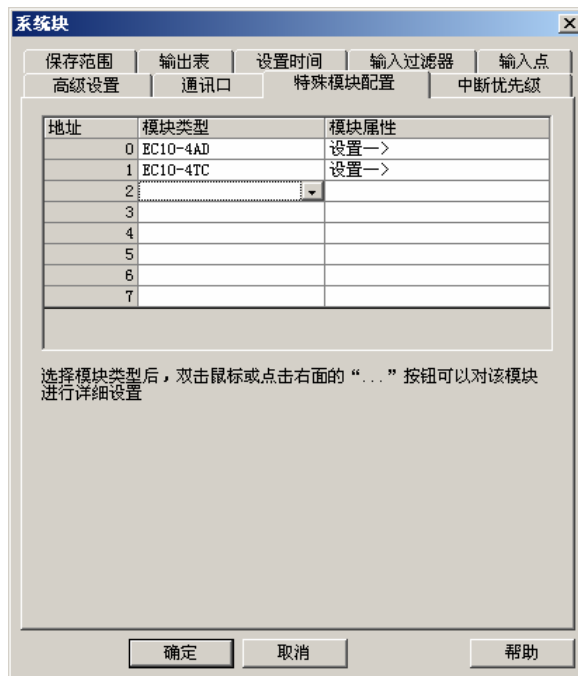


图6-13 特殊模块配置页面

最多可设置 8 个特殊模块，具体可设置的模块类型会因主模块类型不同而不同。在设置时应保证模块地址编号与实际连接的模块顺序编号一致，否则主模块上电时会报错误。

在某个地址上选定模块类型后，双击该行或点击后面的**设置**按钮可以对该模块进行详细设置。

下面分别介绍各个扩展模块的设置界面。在各个设置界面中，如果需要使用模块内置的默认值，请将对应项设为空或“自动”。前面有“D”标记的项表示该项对应主模块 D 寄存器地址。

模拟量输入输出模块

模拟量输入、输出、混合模块的配置如图 6-14 所示。

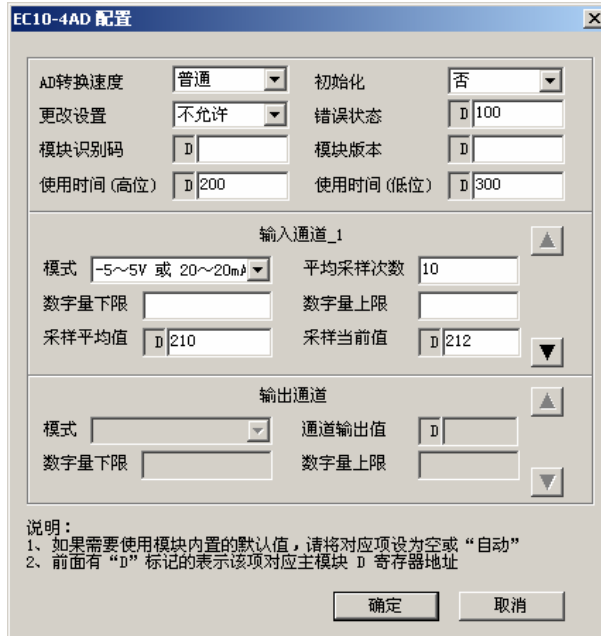


图6-14 模拟量输入、输出、混合模块配置界面

此界面整体分为三部分：

1. 上面区域是模块的整体配置，这些配置项对整个模块或所有通道都有效，例如 AD 转化速度、错误状态等。
2. 中间区域对应该模块的输入通道的配置项。可在此配置输入通道的模式、采样次数、数字量上下限等，通过右侧的上下箭头按钮可以在各通道间切换。对于没有输入通道的模块，此区域不可配置。
3. 下面区域对应该模块的输出通道的配置项，可在此配置输出通道的模式、输出值、数字量上下限等。同样，通过右侧的上下箭头按钮可以在各通道间切换。对于没有输出通道的模块，此区域不可配置。

热电偶、热电阻模块

热电偶、热电阻模块的配置如图 6-15 所示。

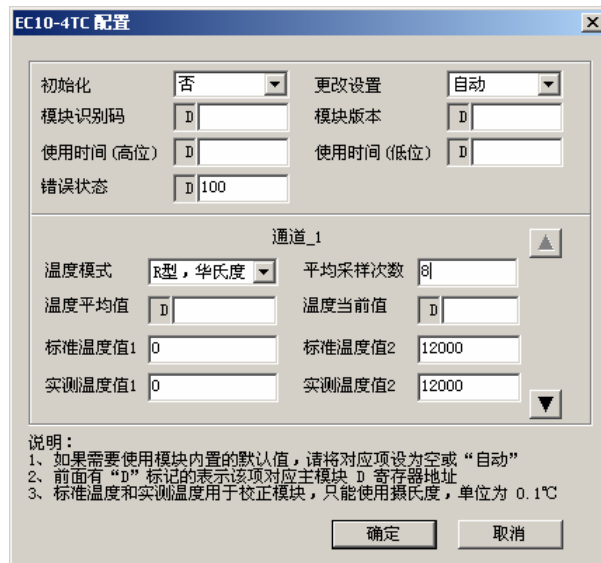


图6-15 热电偶、热电阻模块配置界面

此界面整体分为两部分：上面区域是模块的整体配置，这些配置项对整个模块或所有通道都有效，例如是否初始化、错误状态等。下面区域对应各通道的配置项，可在此配置通道模式、采样次数、标准温度值、实测温度值等。通过右侧的上下箭头按钮可以在各通道间切换。

标准温度和实测温度用于校正模块，只能使用摄氏度，单位为 0.1

6.1.10 中断优先级设置

在系统块界面中，单击**中断优先级**标签，可以设置 PLC 各种中断事件的优先级。对不支持此项设置的 PLC，此页面不可见。

中断优先级分为高、低两级，系统默认每个中断都有固定的优先级，当默认设置不能满足实际需要时，可在此可以手动设置。高优先级的中断可以打断低优先级的中断的执行。

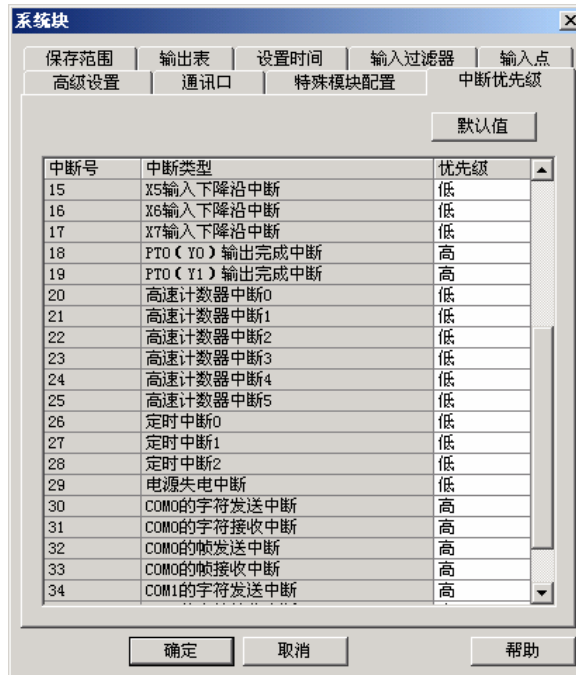


图6-16 系统块中断优先级页面

6.2 PID 指令向导

6.2.1 PID 指令向导执行步骤

对于许多参数复杂的指令，生成这些指令不是很方便，所以软件提供指令向导功能以方便生成这些指令。

PID 指令向导通过对话框列出 PID 计算的各项参数的含义和输入框。输入完成后，PID 指令向导自动生成 PID 参数设置和 PID 执行两个子程序。PID 参数设置子程序是一次设置 PID 指令的参数，当能流有效时，每个运行周期每次执行 PID 指令的子程序。执行步骤如下：

1. 点击**工具/指令向导**菜单项，调出指令向导对话框，见图 6-17。

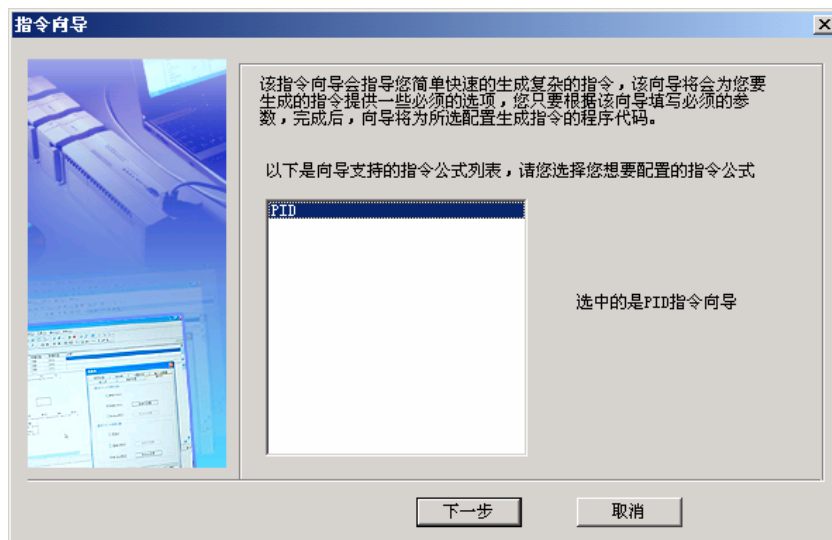


图6-17 指令向导对话框

2. 选择指令向导的类型, 单击下一步, 出现图 6-18 所示对话框。输入相关参数。

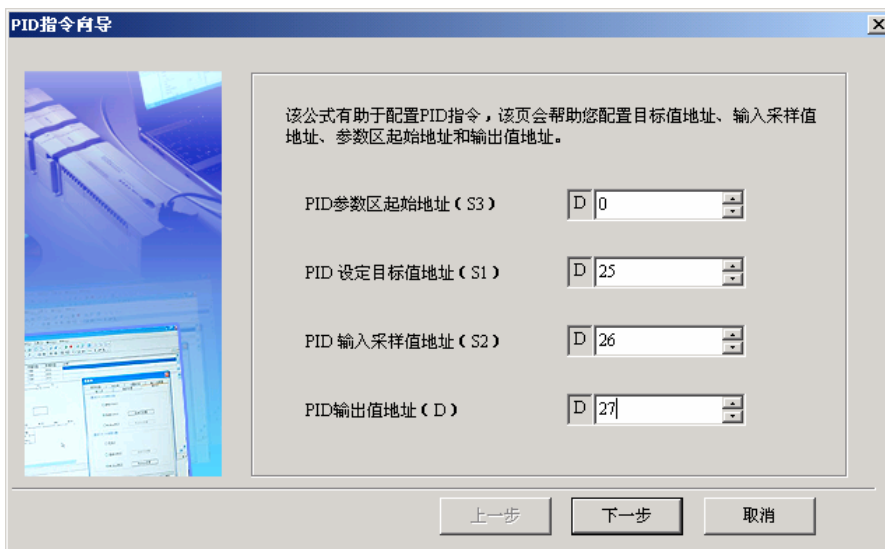


图6-18 PID 指令参数设置对话框

3. 单击下一步, 出现图 6-19 所示对话框。输入目标值和采样时间。

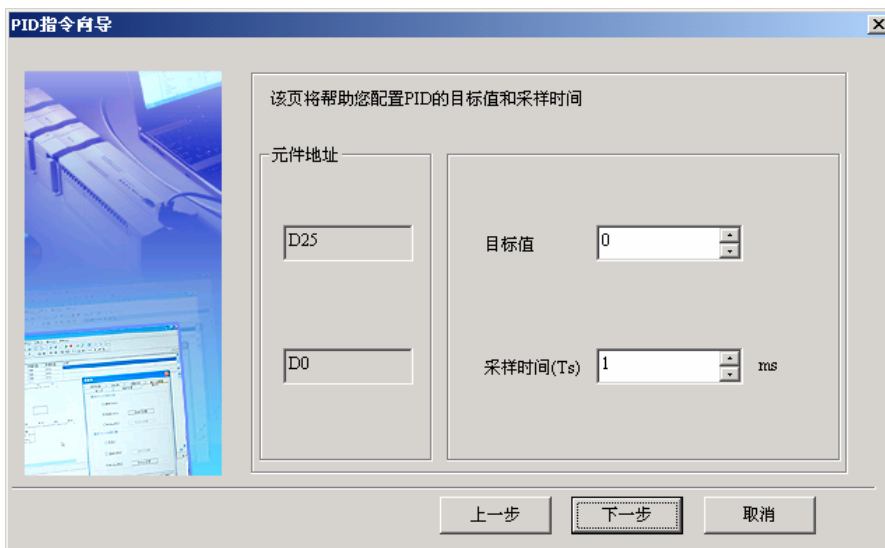


图6-19 目标值和采样时间设置对话框

4. 单击下一步，出现图 6-20 所示对话框。配置 S3+1 单元——动作方向和报警是否有效。

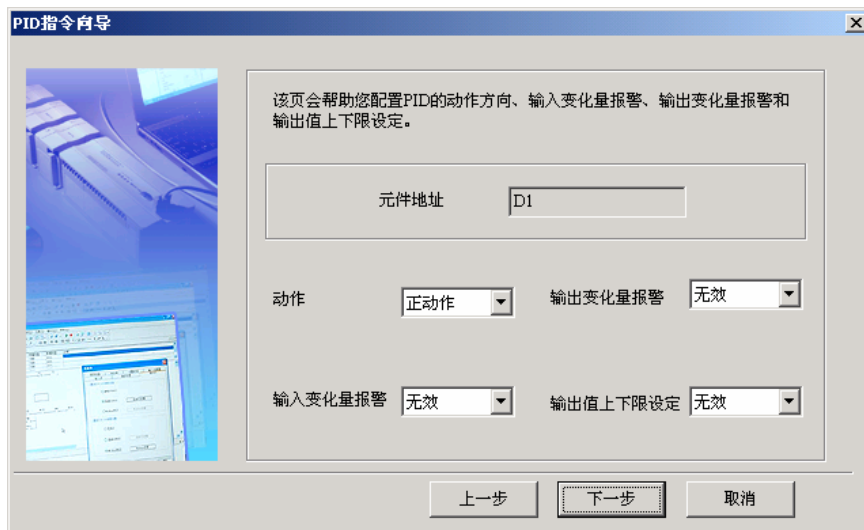


图6-20 动作方向及报警参数配置

5. 单击下一步，出现图 6-21 所示对话框。配置输入变化量报警值和输出变化量报警值。



图6-21 输入和输出变化量报警值配置界面

6. 单击下一步，出现图 6-22 所示对话框。配置输出输入滤波常数、比例增益、积分时间、微分增益、微分时间。



图6-22 滤波常数、比例增益、积分时间、微分增益、微分时间配置界面

7. 单击下一步，出现图 6-23 所示对话框。输入初始化子程序和执行子程序的名字

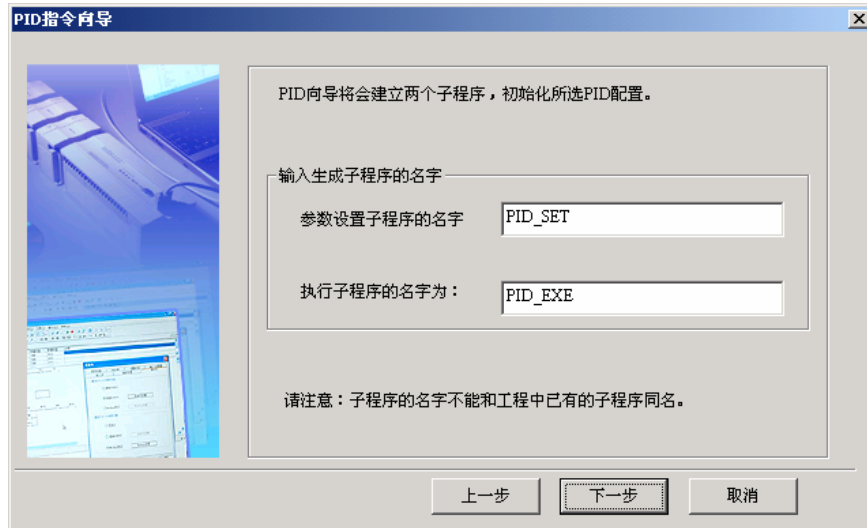


图6-23 输入子程序名字界面

8. 点击下一步，出现图 6-24 所示对话框，显示最后要生成的两个子程序的名字。

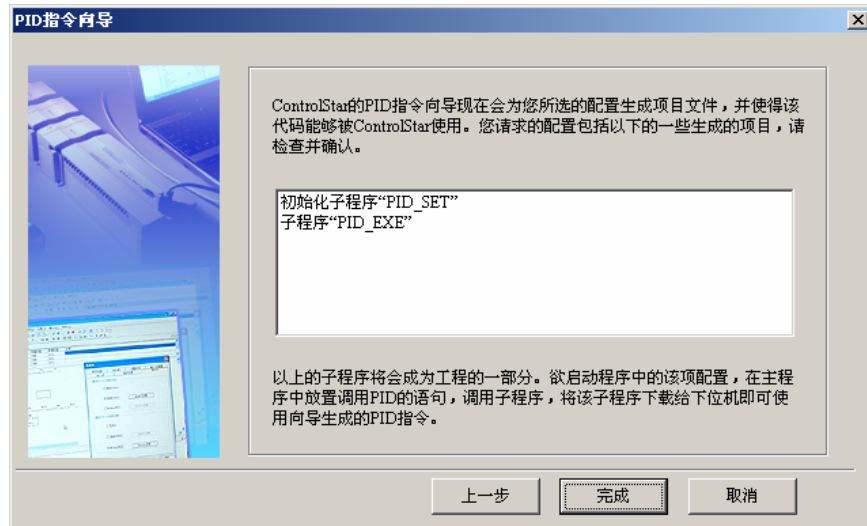


图6-24 子程序显示对话框

9. 点击完成按钮，指令向导结束。你可以在工程管理器中找到新生成的两个子程序节点。

6.2.2 PID 参数设置子程序参数含义

指令向导生成的 PID 参数设置子程序（IL 类型）中 PID 指令各个具体参数说明如表 6-2 所示。

表6-2 PID 指令各个具体参数说明

参数	含义	说明
D	执行程序时，输出运算结果（MV）	
S1	设定目标值（SV）	
S2	设定当前测量值（PV）	
S3 单元	采样时间（Ts）	范围为 1 ~ 32767（ms）但比运算周期短的时间数值无法执行
S3+1 单元	动作方向（ACT）	BIT0 为 0 正动作，为 1 逆动作 BIT1 为 0 输入变化量报警无效，为 1 输入变化量报警有效 BIT2 为 0 输出变化量报警无效，为 1 输出变化量报警有效 BIT3、BIT4 保留 BIT5 为 0 输出值上下限设定无效，为 1 输出值上下限设定有效 BIT6 ~ BIT15 保留

参数	含义	说明
S3+2 单元	输入滤波常数 ()	范围 0 ~ 99[%], 为 0 没有输入滤波
S3+3 单元	比例增益 (Kp)	范围 1 ~ 32767[%]
S3+4 单元	积分时间 (TI)	范围 0 ~ 32767 (×100ms), 为 0 时作为∞处理 (无积分)
S3+5 单元	微分增益 (KD)	范围 0 ~ 100[%], 为 0 时无微分增益
S3+6 单元	微分时间 (TD)	范围 0 ~ 32767 (×10ms), 为 0 时无微分处理
S3+7 ~ S3+14 单元	PID 运算内部数据存储寄存器	
S3+15 单元	输入变化量 (增侧) 报警设定值	0 ~ 32767 (S3+1 的 BIT1=1 时)
S3+16 单元	输入变化量 (减侧) 报警设定值	0 ~ 32767 (S3+1 的 BIT1=1 时)
S3+17 单元	输出变化量 (增侧) 报警设定值	0 ~ 32767 (S3+1 的 BIT2=1 和 BIT5=0 时)。 输出上限设定值-32768 ~ 32767 (S3+1 的 BIT2=0 和 BIT5=1 时)
S3+18 单元	输出变化量 (减侧) 报警设定值	0 ~ 32767 (S3+1 的 BIT2=1 和 BIT5=0 时)。 输出下限设定值-32768 ~ 32767 (S3+1 的 BIT2=0 和 BIT5=1 时)
S3+19 单元	报警输出	BIT0 输入变化量 (增侧) 溢出 BIT1 输入变化量 (减侧) 溢出 BIT2 输出变化量 (增侧) 溢出 BIT3 输出变化量 (减侧) 溢出

6.2.3 PID 执行子程序

指令向导还会生成一个包含 PID 指令调用的子程序, 该子程序会针对上面的参数设置, 将合适的元件地址传入到 PID 指令的各个参数中。需要调用 PID 指令时, 可以很方便的直接调用这个 PID 执行子程序, 而无需自己去写各个参数。

指令格式:

PID	S1	S2	S3	D
-----	----	----	----	---

S1 为目标值用 SV 表示。

S2 为实际的测量值用 PV 表示。执行程序时, 运算结果 MV 被存入 D 中。

S3 ~ S3+6 为设定控制参数。

6.3 数据块设置

6.3.1 数据块简介

数据块用于设置 D 寄存器中的缺省值, 设置完成并通过编译后可以下载到可编程控制器中, 在可编程控制器中的程序运行使用到 D 寄存器时, 将使用所设置的值。

在数据块编辑器中, 可以对 D 寄存器 (数据内存) 进行初始数据赋值, 可以对 D 内存的字或双字赋值, 但是不能对字节赋值。也可以在数据块编辑器中编写注释, 在字符串前增加双斜线可以将其后的内容设为注释。

数据块赋值可以通过 “D 地址 值 1[,值 2][,值 N...]” 的方式为一个或者多个地址赋值。每个数值顺序占用下一个存储区域, 系统自动根据值需要占用的字节数分配存储空间和下一个值的起始地址。在行的上方已经指定过起始地址的情况下, 也可以通过直接输入数字 “值 1[,值 2][,值 N...]” 来完成对存储空间的赋值。

对于值只需要占用一个字但是需要按照双字使用时, 可以用 DD 寄存器, 例如 “DD100 10” 表示将 10 作为一个双字存放到 D100 和 D101 寄存器表示的一个双字中。

数据块的第一行如果没有一个明确地址赋值, 则默认从 D0 开始赋值, 其后的行可包含明确或隐含地址赋值, 已经显示或者隐含赋值的存储区域不能重复赋值。在单地址赋值后键入多个数据值或键入仅包含数据值的行时, 由编辑器指定隐含地址赋值。编辑器根据先前的地址分配及数据值大小 (字或双字) 指定适当的 D 内存数量。

数据块编辑器是一种自由格式文本编辑器, 支持文本编辑器中的常用编辑操作包括拷贝、剪切、粘贴和拖动。键入一行按 ENTER 键、粘贴完毕或者拖动后, 数据块编辑器格式化行 (对齐地址列、数据、注解。捕获 D 内存地址) 并重新显示行。数据块编辑器接受大小写字母并允许使用逗号、制表符或空格, 作为地址和数据值之间的分隔符。

数据块经过编译后, 可以下载到 PLC 中使用。

6.3.2 数据块编辑

寄存器值设置

数据块编辑器是一个文本编辑器，用于输入寄存器地址、寄存器值和注释，它和指令列表编辑器是非常类似的，支持文本编辑器中的常用编辑操作，包括拷贝、剪切、粘贴和拖动。双击工程管理器中的**数据块**节点，将弹出数据块编辑器。在编辑器中输入需要的指令后，软件会自动对齐指令、操作数和注释并将指令关键字用蓝色显示，操作数用黑色显示，注释用绿色显示，如下图所示。

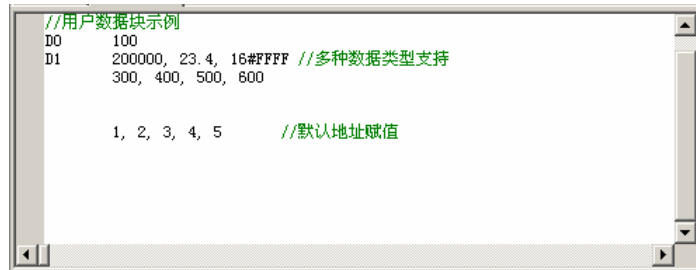


图6-25 数据块编辑器举例

在编辑器的任意位置输入双斜线“//”表示当前行后面的内容是注释。

正确性校验

输入每一行指令后，如果光标切换到其他行，则会对离开的行进行正确性校验，如果有语法错误，软件会将有错误的文本行显示为红色。当鼠标悬浮在有错误的文本行时，软件会用工具提示的方式提示错误信息。只有校验正确的语句才会格式化显示，否则会在行头显示一个“!”并且用红色突出有语法错误的行。在一次粘贴多行文本内容到编辑器中时，编辑器会对每一行作语法检查。

有语法错误时，请将鼠标放在有错误的位置，软件显示如下图所示。

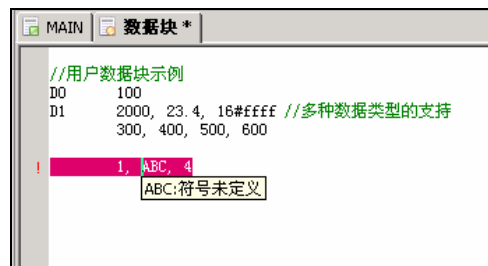



图6-26 语法错误显示举例

在图中，由于不能使用数值外的字符赋值，因此软件不能识别“ABC”。

6.3.3 数据块编译

编辑完成数据块后，点击工具栏上的  按钮，可以检查当前数据块中是否有错误。如果无错通过编译，则输出窗口中显示如下内容。

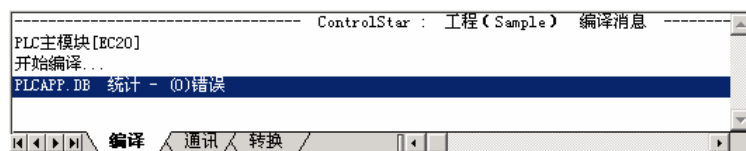


图6-27 编译结果输出窗口

6.3.4 其他编辑功能

数据块编辑器和指令列表编辑器一样提供拷贝、剪切、粘贴、查找、替换、定位、标签、预览和打印功能，详细信息参见通用编辑功能。

第七章 PLC 控制

PLC 控制功能是 ControlStar 编程软件通过串口对 PLC 完成各种控制操作，包括控制 PLC 的运行状态、改变 PLC 中存储的数据内容、从 PLC 中读取各种信息等等。

7.1 PLC 控制方法

在使用 PLC 控制功能前，需要将计算机和 PLC 硬件通过串口正确连接，在 ControlStar 编程软件中设置连接信息，设置方法参见与 PLC 通讯。

在 ControlStar 编程软件中，和 PLC 控制相关的功能在主菜单的 **PLC (P)** 菜单中，常用的控制按钮也可以在 PLC 工具栏中找到。

控制 PLC 时，使用 **PLC (P)** 菜单中的某个子菜单，控制结果会在输出窗口的**通讯**页面中显示。图 7-1 为控制 PLC 运行成功后的提示。

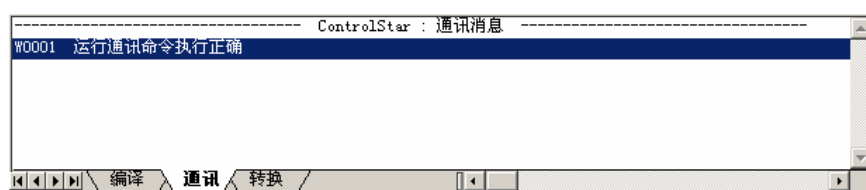


图7-1 控制 PLC 运行成功后的提示信息

图 7-2 为控制 PLC 运行失败后的提示。

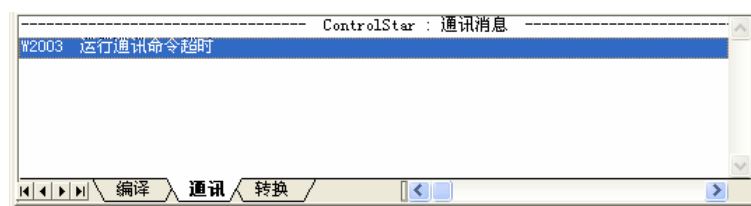


图7-2 控制 PLC 运行失败后的提示信息

某些控制操作需要 PLC 处于停止状态，如果使用时 PLC 正处于运行状态，系统会提示是否自动停止 PLC 的运行，如下图所示。



图7-3 是否自动停止 PLC 运行的提示信息

点击**确定**按钮停止 PLC 运行继续操作，点击**取消**按钮退出当前操作。

某些控制操作可能会使 PLC 运行数据丢失或者无法继续正常工作，包括 PLC 元件值清除、PLC 程序清除、PLC 数据块清除、PLC 格式化和恢复默认配置。在控制生效前，软件会显示确认对话框，请输入和对话框中显示的数字相同的内容后继续，确保这些控制操作不是误操作引起的。

注意

在使用上述控制前，请确保操作结果不会影响 PLC 的正常工作甚至引发其他意外情况。

7.2 运行 PLC

点击 **PLC/运行** 菜单项，如果 PLC 开关位于“ON”或“TM”位置，PLC 将开始运行。如果 PLC 已经运行，控制不会引起任何的变化。

7.3 停止 PLC

点击 **PLC/停止** 菜单项，无论 PLC 的开关处于哪个位置，都会使 PLC 停止运行。如果 PLC 已经停止，控制不会引起任何的变化。

7.4 复位 PLC

点击 **PLC/复位** 菜单项，PLC 复位，停止执行用户程序。PLC 内部状态恢复到初始状态，然后重新启动运行。

7.5 下载程序

下载功能用于把 ControlStar 软件生成的系统块、数据块、用户程序通过串口下载到 PLC 中，下载时要求 PLC 处于停止运行状态。

点击 **PLC/下载** 菜单，如果上次编译的结果文件已存在，会提示是否需要重新编译程序，如下图所示。

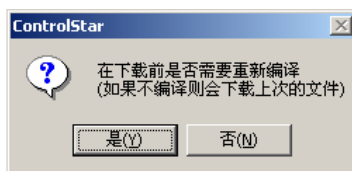


图7-4 是否编译信息提示

选择**是**按钮重新编译程序，如果程序有错误，软件退出下载过程，需要先修改程序错误。选择**否**按钮不编译当前程序继续下载操作。需要注意的是，选择**否**不重新编译，软件会使用上次编译的结果，下载到 PLC 中运行的程序和软件界面显示的程序可能会不一样。

编译正确后，软件弹出下载窗口，选择要下载的内容。如图 7-5 所示。

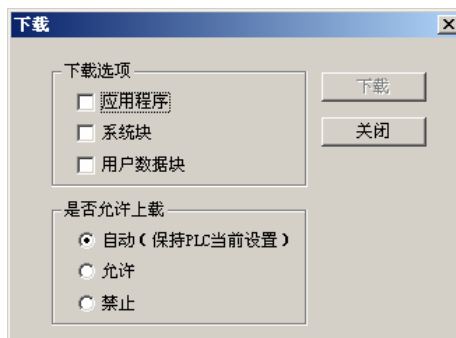


图7-5 下载内容选择窗口

选择好需要下载的数据项后，**下载**按钮变为可用，点击**下载**按钮下载数据到 PLC，也可以点击**关闭**终止下载操作。

可以通过**是否允许上载**选项，控制本次下载的程序是否允许以后上载。**允许**表示以后可以上载，**禁止**表示以后不能上载，**自动**表示不改变 PLC 中的当前设置，默认为**自动**。对于不支持**禁止上载**功能的 PLC 型号，该选项将无法设置。

必须同时选择**应用程序**、**系统块**、**用户数据块**三项后，才能选择**允许**上载选项。

选择**下载**按钮后，如果设有下载密码，并且在启动软件后没有输入过下载密码，软件会弹出密码窗口要求输入写密码，如下图所示。



图7-6 密码校验窗口

密码输入校验正确后开始下载，密码错误会提示重新输入密码。点击**取消**按钮退出下载。开始下载后软件显示下载进度，下载完毕后进度框自动消失。

7.6 上传程序

上传功能用于把 PLC 中的系统块、数据块、用户程序等内容通过串口上传到计算机中，并且为其新建工程保存。

点击 **PLC/上传** 菜单项，这时软件会保存关闭已经打开的工程，自动新建工程保存上传内容。此时需要为上传内容指定工程名，然后点击**确定**继续上传过程，点击**取消**终止上传过程。如图 7-7 所示。



图7-7 上传工程对话框

在图 7-8 所示对话框中指定要上传的内容，点击**上传**按钮继续上传。点击**关闭**按钮可以退出上传。

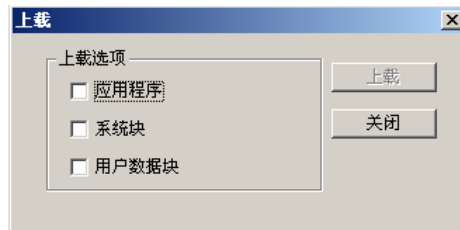


图7-8 上传内容选择对话框

点击**上传**按钮后，如果设有上传密码，并且在启动软件后没有输入过上传密码，软件会弹出密码窗口要求输入写密码，如图 7-9 所示。否则直接开始上传。



图7-9 密码校验对话框

如果密码输入正确则开始上传，密码错误则软件提示并返回上传对话框界面。开始上传后，软件弹出上传进度框，显示上传进度，上传完毕后进度框自动消失。

上传成功后，软件会自动在工程中打开上传的主程序。

7.7 比较计算机和 PLC 中用户程序

点击 **PLC/程序比较** 菜单项，可以比较计算机和 PLC 中用户程序是否相同。如果程序相同，信息输出窗口中显示执行正确的信息，如图 7-10 所示。



图7-10 程序比较信息输出窗口

如果程序不相同，则显示如下信息。



图7-11 程序不同的提示信息框

需要注意的是，在计算机和 PLC 中的用户程序一致时，如果使用转换功能，将当前的程序使用不同的编程语言显示，程序执行逻辑不会发生改变，但是可能会导致生成的用户程序和 PLC 中保存的用户程序不完全一致。此时使用比较功能也会提示两个程序不相同。

7.8 设置密码

点击 **PLC/密码设置** 菜单项下的各子菜单，可以设置 PLC 上载、下载、监控密码。设置好的密码将保存在 PLC 中。密码设置对话框如图 7-12 所示。

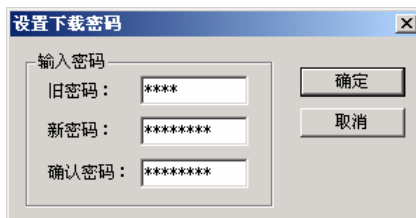


图7-12 密码设置对话框

对话框标题显示了当前正在设置的是上载、下载还是监控密码。如果 PLC 中已经设置过密码，需要在**旧密码**中输入原有的密码，只有旧密码正确，才允许设置新密码。

7.9 清除 PLC 错误信息

PLC 发生错误后，其错误灯将被点亮，错误修复后，错误状态不会自动恢复（错误灯仍然被点亮）。点击 **PLC/PLC 错误信息清除** 菜单项可以手动将 PLC 中的错误状态清除。

7.10 清除 PLC 元件值

点击 **PLC/PLC 元件值清除** 菜单项清除 PLC 中所有元件值。清除元件值需要 PLC 处于停止运行状态。在系统块中设置为掉电保持的元件，清除元件值时也全部被清除。

注意

该功能可能导致 PLC 运行不正确或者中间工作数据丢失，请谨慎使用！

操作时软件会显示下图所示确认对话框，继续操作选择**确定**按钮，退出操作点击**取消**按钮。

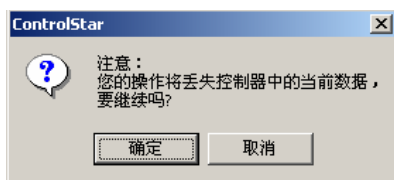


图7-13 清除 PLC 元件值确认对话框

7.11 清除 PLC 程序

点击 **PLC/PLC 程序清除** 菜单项，可以清除 PLC 中的用户程序，清除用户程序需要 PLC 处于停止运行状态。

注意

清除 PLC 中的用户程序会导致 PLC 运行后不执行任何用户程序，仅完成输入输出、通讯、内部处理功能，请谨慎使用该功能！

操作时软件会显示图 7-13 所示确认对话框，继续操作选择**确定**按钮，退出操作选择**取消**按钮。

7.12 清除 PLC 数据块

点击 **PLC/PLC 数据块清除** 菜单项，将清除 PLC 中所有数据块设置。清除数据块需要 PLC 处于停止运行状态。

注意

清除 PLC 数据块会导致 PLC 运行后不再使用数据块的预设值初始化 D 元件，请谨慎使用！

操作时软件会显示图 7-13 所示确认对话框。继续操作选择**确定**按钮，退出操作选择**取消**按钮。

7.13 PLC 格式化

点击 **PLC/PLC 格式化** 将格式化 PLC 中所有数据，包括清除用户程序、恢复默认配置、清除数据块，清除数据块需要 PLC 处于停止运行状态。

注意

本操作会丢失 PLC 中已经下载和设置的全部数据，请谨慎使用该功能！

操作时软件会显示图 7-13 所示确认对话框，继续操作选择**确定**按钮，退出操作选择**取消**按钮。

7.14 恢复默认配置

点击 **PLC/恢复默认配置**，将使 PLC 中所有配置恢复为出厂配置。该操作需要 PLC 处于停止状态。

注意

本操作会将 PLC 中的设置恢复为出厂的默认配置，已经下载的系统块内容不再起作用。出厂默认配置和实际的使用情况可能不相符合，有可能导致 PLC 无法正常工作，请谨慎使用此功能！

操作时软件会显示图 7-13 所示确认对话框。继续操作选择**确定**按钮，退出操作选择**取消**按钮。

7.15 获取 PLC 信息

点击 **PLC/PLC 信息** 可以从 PLC 中获取并显示各种运行数据和 PLC 重要信息。

在信息显示窗口上可以看到 PLC 当前运行的重要信息，如下图所示。

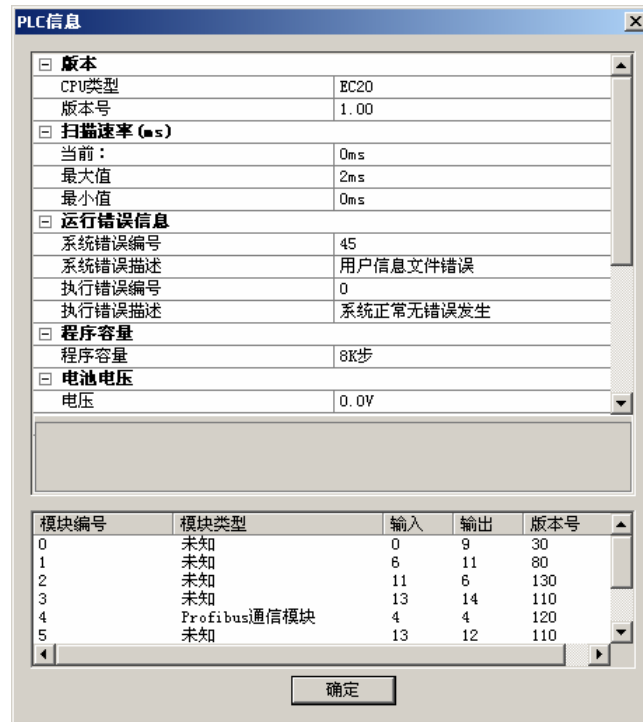


图7-14 PLC 信息显示窗口

点击**确定**按钮退出。

7.16 显示和设置 PLC 当前时间

点击 **PLC/PLC 时钟** 可以显示和设置 PLC 当前时间，时钟窗口如下图所示。

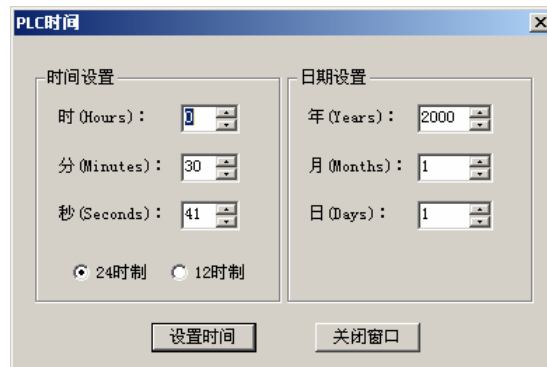


图7-15 PLC 时钟窗口

窗口中显示当前从 PLC 中读取的日期和时间，可以输入新的日期和时间，点击**设置时间**按钮，新的时间将设置到 PLC 中。点击**关闭窗口**按钮退出 PLC 时钟窗口。

7.17 从 RAM 中生成数据块

该功能可以从 PLC 中连续读取最多 500 个 D 寄存器的数据值并显示出来，结果可以合并到数据块中或者覆盖原来的数据块。

点击 **PLC/从 RAM 生成数据块**，弹出图 7-16 所示窗口。

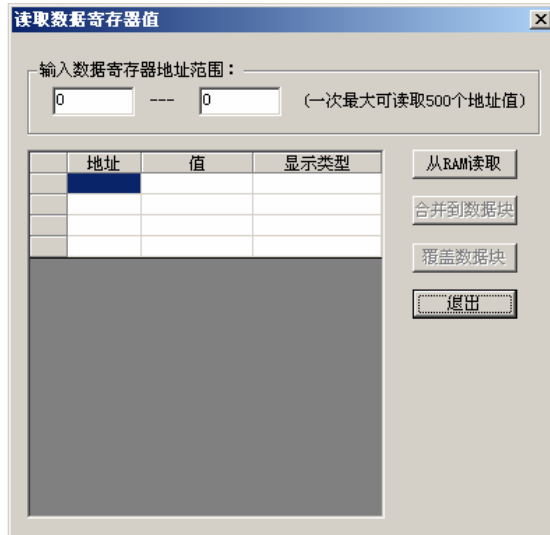


图7-16 读取数据寄存器值窗口

输入要读取的数据块的范围，然后点击**从RAM读取**按钮，执行正确后数据被读取到列表中。如图7-17所示。

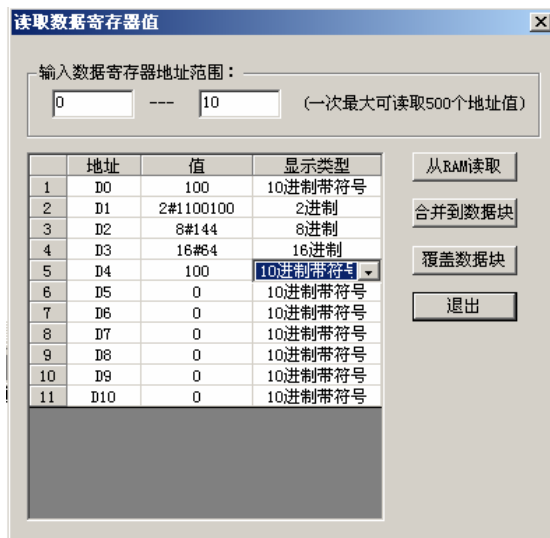


图7-17 读取数据寄存器值后的显示界面

可以在图7-17的**显示类型**中选择以16、10、8、2进制来显示数据。读取成功后**合并到数据块**和**覆盖数据块**按钮变为可用。点击**合并到数据块**将生成结果追加到当前数据块的内容后面。点击**覆盖数据块**将生成结果替代数据块中已经存在的内容。退出寄存器值读取窗口后，将提示数据块已经发生变化并自动打开数据块窗口。如图7-18所示。

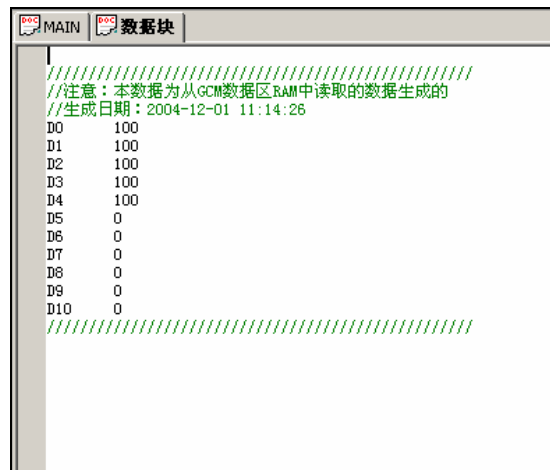


图7-18 数据块改变提示信息

第八章 程序调试


程序下载到硬件设备中后，为了检测程序逻辑的正确性，可以对程序进行调试。需要进行程序调试，请确认编程设备与 PLC 硬件之间通讯正常，且 PLC 中的程序和当前打开的程序一致。

注意


程序下载或者上载后，如果将当前的程序使用不同的编程语言显示，程序执行逻辑不会发生改变，但是可能会导致生成的程序和 PLC 中保存的程序不完全一致。

8.1 如何进入和退出调试状态

进入调试状态

选择**调试/监控**菜单或点击工具栏上的监控按钮，软件会首先校验计算机和 PLC 中程序的一致性，不一致则会给出提示。然后会检查硬件设备中的程序是否需要监控密码。如果需要，必须正确输入监控密码后才能开始监控。如果上述检查都能通过的话，即可进入监控状态。

退出调试状态

需要停止调试时，同样选择**调试/监控**菜单或点击工具栏上的监控按钮，程序会退出调试状态，恢复为编辑状态。





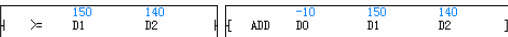
8.2 注意事项

1. 在使用调试功能时，软件并不要求 PLC 硬件必须处于运行状态，但是如果需要调试硬件中的程序的逻辑正确性的话，建议将硬件设置为运行状态，否则调试结果可能会不准确。
2. 在调试状态下，软件定时到 PLC 硬件中读取软元件值，并显示在程序窗口中，即调试结果并不是实时的。同时，PLC 硬件的 CPU 循环扫描周期很短，远远小于软件定时读取的周期，因此，在调试时观察到的元件值和实际值比起来，必然有一定的滞后。在 9600 的串口通讯速率并且没有干扰的情况下，界面的元件值更新比 PLC 中有 0.2 秒的滞后，通讯不畅通的情况下，可能会有数秒的滞后。了解这一点，对正确把握程序逻辑及判断执行结果，都是非常重要的。
3. 为了保证调试的效率，软件只在进入调试状态时检测计算机和 PLC 中程序的一致性，所以如果在调试过程中切换所连接的 PLC 的话，软件并不能识别这个操作，这样就会导致调试结果与实际运行的程序不一致，因此，建议不要进行这样的操作。

8.3 梯形图程序调试

8.3.1 可调试的元件

在梯形图中，以下元件可以调试（在调试时会根据值的变化有不同的显示）：

1. 常开触点——当软元件值为 1 时，显示为连通状态 
2. 常闭触点——当软元件值为 0 时，显示为连通状态 
3. 步进触点——当状态位为 1 时，显示为连通状态 
4. 线圈——当软元件值为 1 时，显示为连通状态 
5. 比较触点及应用指令——此类元件引用多个地址，且每个地址不能简单的以逻辑值 1 和 0 表示，必须显示具体的数值 

一个正在调试中的完整的梯形图程序如图 8-1 所示。

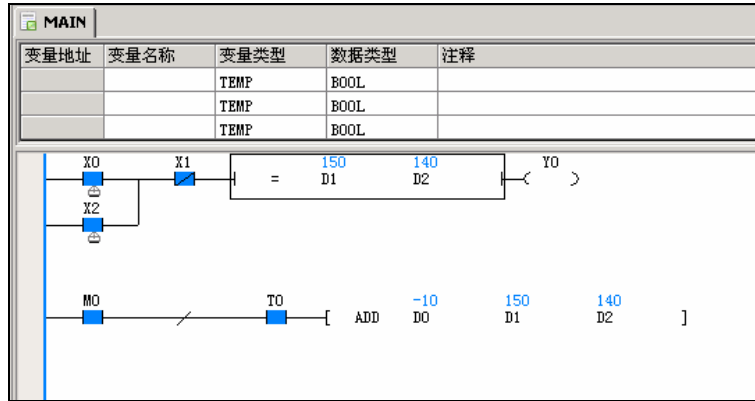


图8-1 正在调试中的完整的梯形图程序

8.3.2 元件值写入和强制

在调试过程中，可能为了达成某些条件，而需要手工改变某些软元件的值。元件值写入和强制提供了这个功能。写入与强制的不同之处在于：写入元件值仅一次有效，写入后的值可能会随着程序的运行被改变，但是强制的元件值会一直记录在 PLC 硬件中，直到取消强制为止。

需要执行写入或强制功能时，首先选中要写入或强制的元件，从右键菜单中选择**写入**或者**强制**。此时会弹出相应对话框，对话框中将列出所选元件引用到的所有软元件地址。更改需要写入或强制的地址中的值，点击**确定**，这些值就会下发到 PLC 硬件。如图 8-2，点击地址 M0 对应的值所在的单元格，在下拉框中选择“ON”或者“OFF”，点击**确定**，这些值就会下发到 PLC 硬件。当这些值在硬件中生效后，在后续的调试过程中就可以看到变化结果。元件值写入对话框如图 8-2 所示。

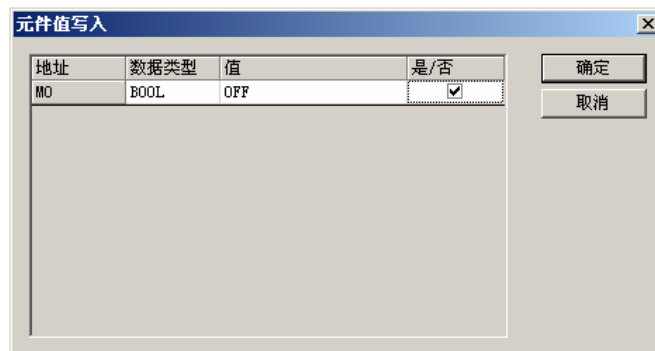


图8-2 元件值写入对话框

元件值强制的操作过程同写入，强制对话框如图 8-3 所示。

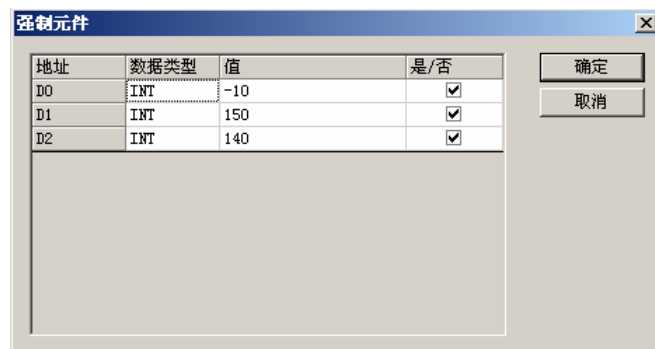


图8-3 元件值强制对话框

强制的软元件，在梯形图中会有一个锁标记，如图 8-4 所示。

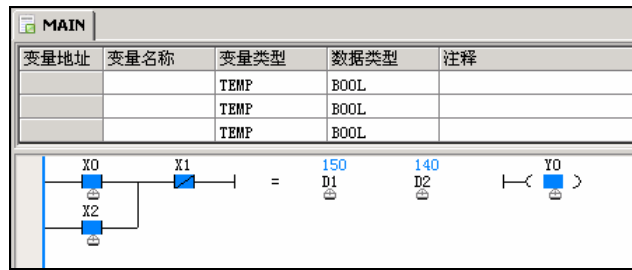


图8-4 梯形图中强制元件标志

8.3.3 解除强制

对于不再需要强制的元件，可以将它解除强制。需要使用解除强制功能时，先选中要解除的元件，从右键菜单选择**取消强制**。此时会弹出相应对话框，如图 8-5 所示。对话框中将列出所选元件中已经被强制过的软元件。可以有选择的解除某些软元件的强制，点击**确定**后，这些强制值会从 PLC 硬件中删除，该软元件对应的锁标记也会消失。

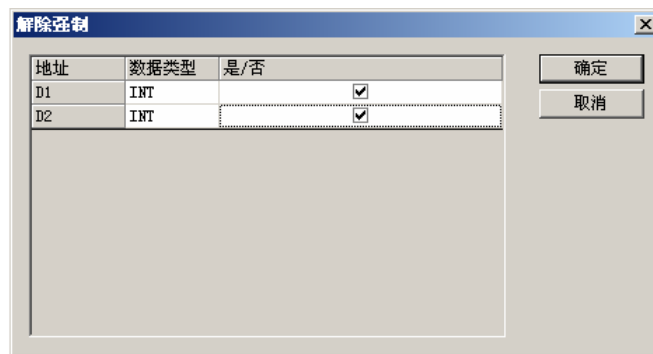


图8-5 解除强制对话框

8.3.4 强制结果及限制规则

强制结果的显示及其含义对照见表 8-1。

表8-1 强制结果含义及显示

强制结果	说明	界面显示
没有强制	该地址没有任何强制信息	无
部分强制（左）	某个双字地址的左半个单字被强制，而右半个单字没有被强制	
部分强制（右）	某个双字地址的右半个单字被强制，而左半个单字没有被强制	
间接强制	某个双字地址被强制，则它所包含的两个单字地址都被间接强制	
完全强制	对于位地址或单字地址，当它被直接强制时。 对于双字地址，当它包含的两个单字地址都被强制时	

写入和强制的限制

1. 只读的元件和常数不能够被写入或强制。
2. LM 和 V 元件不能被写入或强制。
3. 带变址的元件不能被写入或强制。
4. 所有 Kn 寻址的元件都不能被强制，Kn 址的 SM 元件不能被写入。
5. SPD 指令在执行时，对它的输入端口（操作数 1）使用的软元件强制无效。PLSY、PWM 指令在执行时，对它的输出端口（操作数 3）使用的软元件强制无效。PLSR 指令在执行时，对它的输出端口（操作数 4）使用的软元件强制无效。
6. 对于单字元件，如果它所属的双字元件被强制，则它不能再被强制。
7. 对于双字元件，如果所包含的两个单字软元件任一个被其他元件强制过，则它不能再被强制。

解除强制的限制

有些元件既可以作单字使用，也可以作双字使用。当某个元件被作为单字强制时，不能作为双字来解除强制，反之亦然。

8.3.5 读取所有强制元件值

由于被强制的所有软元件都是保存在 PLC 硬件中的，因此在调试时可以将这些被强制的软元件读取上来查看。从右键菜单选择**读取所有强制**，将弹出对话框。对话框显示当前已被强制的所有软元件及强制值，如图 8-6 所示。

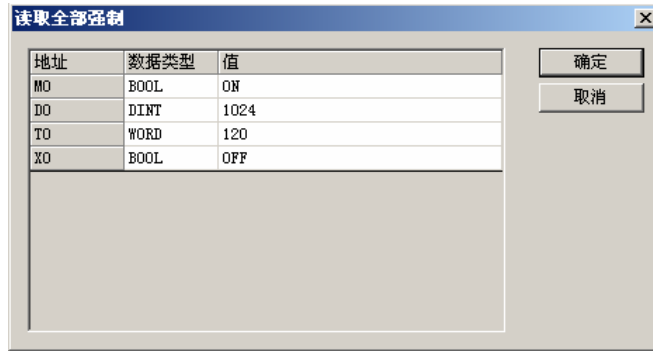


图8-6 读取全部强制对话框

8.4 指令列表程序调试

指令列表程序的调试功能比较简单，在监控状态下，在有可监控的软元件的代码行点击鼠标右键，会看到**添加到监控表**菜单项，见图 8-7。点击该菜单项就可完成将当前行的可监控软元件添加到监控表的操作。可以在监控表中查看被添加的软元件的监控值。

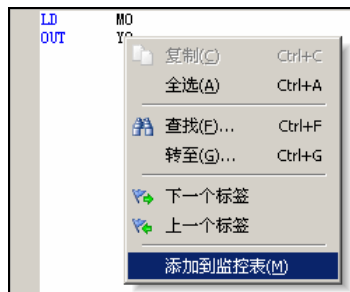


图8-7 指令列表程序调试显示菜单

8.5 顺序功能图程序调试

8.5.1 可调试的元件

只有初始步进符和普通步进符元件可以被调试。当他们对应的 S 软元件状态为“ON”的时候，他们显示为通路，如图 8-8 所示。

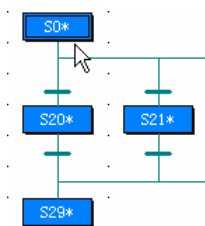


图8-8 显示为通路的元件

8.5.2 元件值写入和强制

在调试过程中，可能为了达成某些条件，而需要手工改变某些软元件的值，元件值写入和强制提供了这个功能。写入与强制的不同之处在于：写入元件值仅一次有效，写入后的值可能会随着程序的运行被改变，但是强制的元件值会一直记录在 PLC 硬件中，直到取消强制为止。

需要执行写入或强制功能时，首先选中要写入或强制的元件，从右键菜单中选择**写入**或者**强制**，此时会弹出相应对话框，列出所选元件引用到的所有软元件地址。更改需要写入或强制的地址中的值，点击**确定**，这些值就会下发到 PLC 硬件。如图 8-9，点击地址 S0 对应的值所在的单元格，在下拉框中选择“ON”或者“OFF”，点击**确定**，这些值就会下发到 PLC 硬件。当这些值在硬件中生效后，在后续的调试过程中就可以看到变化结果。

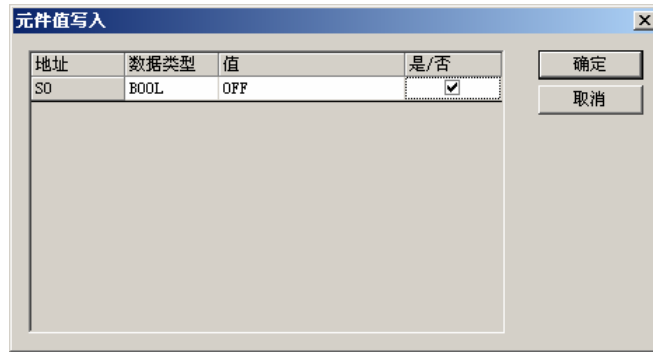


图8-9 元件值写入对话框

元件值强制的操作方法同写入，强制对话框如图 8-10 所示。

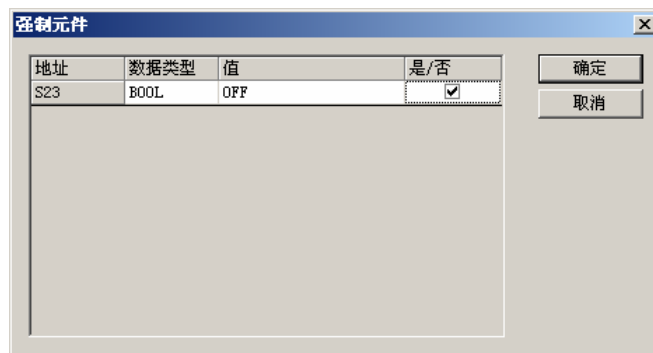



图8-10 元件值强制对话框

强制的软元件地址，在顺序功能图中会有一个锁标记，如 .

8.5.3 解除强制

对于不再需要强制的元件，可以将它解除强制。需要使用解除强制功能时，先选中要解除的元件，从右键菜单选择**取消强制**，此时会弹出相应对话框，见图 8-11。对话框中列出了所选元件中已经被强制过的软元件地址。可以有选择的解除对某些地址的强制，点击**确定**后，这些强制值会从 PLC 硬件中删除，该地址对应的锁标记也会消失。

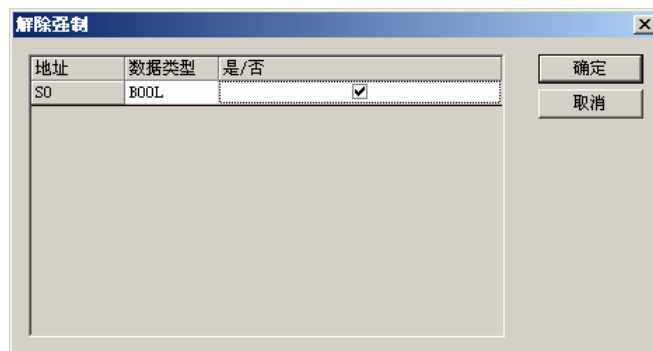


图8-11 解除强制对话框

8.5.4 读取所有强制元件值

由于被强制的所有元件地址都是保存在 PLC 硬件中的，因此在调试时可以将这些被强制的地址读取上来查看。从右键菜单选择**读取所有强制**，将弹出对话框。对话框中将显示当前已被强制的所有软元件地址及强制值，如图 8-12 所示。

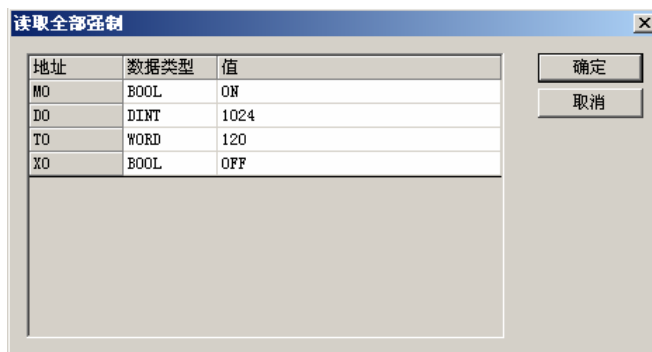


图8-12 读取全部强制对话框

8.6 元件状态监控表

元件状态监控表提供了在调试过程中监控元件值的功能。调试程序时允许将程序输入元件、输出元件置入元件监控表中，以便在将程序下载至 PLC 之后追踪其状态。

元件监控表有两种模式：编辑模式和监控模式。编辑模式下可以执行所有的编辑功能，不能执行监控功能。在监控模式下，可以同时执行监控功能和编辑功能。

在线梯形图功能中，修改元件值和强制元件后，元件自动被加入到元件状态监控表。元件状态监控表在监控模式下自动刷新元件值。

元件状态监控表可提供编辑、排序、查找、自动刷新显示指定元件的当前值、写入元件值、强行指定元件/变量的值、解除强制等功能。

元件状态监控表的字段包括：元件地址、数据类型、显示格式、当前值、新数值（写入值）。

下图是元件状态监控表的示意图。

	元件名称	数据类型	显示格式	当前值	新数值
1	X1	BOOL	二进制		
2	Y1	BOOL	二进制		
3	D1	WORD	十进制		
4	D2	WORD	十进制		
5	开关6	BOOL	二进制		
6		WORD	十进制		
7		WORD	十进制		
8		WORD	十进制		

图8-13 元件状态监控表

8.6.1 添加或删除元件监控表

新建工程有一个默认的元件监控表，可以根据实际需要，在工程内添加多个元件监控表（不多于 8 个）。

右键单击**工程管理**窗口的**元件监控表**节点，在弹出菜单中选择**插入监控表**，即可新增一个元件监控表。新增的元件监控表的默认名称为“监控表_1”、“监控表_2”……“监控表_8”，可以使用**重命名**将其改名。

右键单击某个监控表，在弹出菜单中选择**删除**，可以删除所选监控表。

8.6.2 编辑模式与监控模式

元件监控表有两种模式：监控模式和编辑模式。在监控模式下，ControlStar 除了具有编辑功能外，还会自动读取 PLC 数据，刷新元件值，并提供元件值写入、元件强制、元件解除强制和元件显示功能。在编辑模式下仅仅只有编辑功能。

在编辑模式下单击**调试/监控**，或者点击图标，将进入监控模式。在监控模式下执行相同的操作将进入编辑模式。

元件值写入功能

选定元件后，点击右键，在弹出菜单中选择**写入元件值**，输入要写入的元件值。系统将对输入内容作合法性检查（写入数据类型，写入值等），合法的数据在执行写入功能后被写入 PLC，并立即刷新该元件当前值。不合法的输入不能执行写入功能，并提示输入错误。需要注意的是，有些元件不可以被写入，具体的元件参见**梯形图程序调试**。

元件强制功能

选定元件后，点击右键，在弹出菜单中选择**强制**。将选定的元件状态设置为强制状态，如果这时在值写入区有合法值，则元件进入强制状态，同时值写入区的值被写入 PLC。强制后能在元件状态监控表中明显看到元件被强制，同时在线梯形图中也要能被识别出来。需要注意的是，有些元件不可以被强制，具体元件参见**梯形图程序调试**。

元件解除强制功能

选定元件后，点击右键，在弹出菜单中选择**解除强制**。将选择某个被强制的元件解除强制，也可以选择解除全部被强制元件强制功能。强制解除后，在元件状态监控表和在线梯形图上可以明显看到元件强制被解除。

元件显示功能

元件状态监控表中的元件可以显示为不同的显示格式和数据类型，显示格式包括二进制、十进制、十六进制三种格式，因为不同的元件支持不同的数据类型，在输入元件地址编辑数据类型时，数据类型列以下拉列表的方式显示该元件支持的所有数据类型。元件值可能被显示为半个锁、灰锁等不同的形式，具体参见**梯形图程序调试**。

8.6.3 切换编址方式

在元件监控表中输入了元件地址后，如果该元件地址在全局变量表中有定义，可以通过选择**符号编址**菜单在变量名和元件地址之间切换。地址编址模式下所有元件显示为元件地址。

符号编址模式下在全局变量表中定义的地址显示为全局变量名称，没有定义的显示为元件地址。需要注意的是，局部变量不能切换为符号编址。

8.6.4 元件监控表的复制填充与序列填充

元件监控表除了提供复制粘贴等常用的编辑功能外，还提供复制填充和序列填充功能，当需要批量输入相连的元件地址、元件类型以及显示格式时可使用该功能。

序列填充

在需要填充范围内的最上面一行输入数据，然后选择其后面的若干行，在右键菜单中选择**序列填充**，可自动将最上单元格的值，以序列的方式向下填充，填充后形成一系列连续的数据。序列填充功能仅可用于**元件名称**列。序列填充前后界面如图 8-14 和图 8-15 所示。



图8-14 序列填充前界面

元件名称	数据类型	显示格式
1	D100	WORD 十进制
2	D101	WORD 十进制
3	D102	WORD 十进制
4	D103	WORD 十进制
5	D104	WORD 十进制
6	D105	WORD 十进制
7	D106	WORD 十进制
8	D107	WORD 十进制
9	D108	WORD 十进制
10		WORD 十进制
11		WORD 十进制
12		WORD 十进制

图8-15 序列填充后显示界面

复制填充

在需要填充范围内的最上面一行输入数据，然后选择其后面的若干行，在右键菜单中选择**复制填充**，可将最上面单元格的值，复制到后面选择的所有单元格中。

复制填充功能可用于**元件名称**、**数据类型**和**显示格式**列。在**数据类型**和**显示格式**填充过程中，如果某个元件地址不能使用第一格的数据，则其数据保持不变。

复制填充前后显示界面如图 8-16 和图 8-17 所示。



图8-16 复制填充前显示界面



图8-17 复制填充后显示界面

8.6.5 导入与导出元件监控表

元件监控表可以与 EXCEL 表格进行数据交换，可以将用 EXCEL 文件编辑的数据导入到元件监控表中，也可以将元件监控表导出到 EXCEL 文件中。与元件监控表交换的 EXCEL 文件格式如图 8-18 所示。其中 A、B、C、D、E 列分别对应元件名称、数据类型、显示格式、当前值、新数值。

	A	B	C	D	E
1	D0	WORD	十进制		0
2	D1	WORD	十进制		1
3	D2	WORD	十进制		2
4	D3	WORD	十进制		3

图8-18 与元件监控表交换的 EXCEL 文件格式

导入与导出功能的操作可参见**导入导出全局变量表**。

8.7 在线修改

需要在 PLC 运行状态下修改其中的程序时，可以使用在线修改功能。

如何操作

点击**调试/在线修改**菜单可以切换在线修改状态。使用此功能之前，请确保软件已经与 PLC 硬件成功建立了通讯关系，且 PLC 处于运行状态。

在在线修改状态下，可以与正常编辑时一样修改主程序、子程序及中断子程序的内容。修改完之后，点击**PLC/下载**菜单，软件将会编译当前工程的所有程序并自动下载至 PLC 硬件中。下载完成后，PLC 就会按照新下载的程序运行。

在退出在线修改状态时，针对某些 PLC 类型，软件会询问是否需要固化已下载的程序，只有选择**是**才能将之前在线修改的内容保存在 PLC 中，选择**否则** PLC 内会恢复到在线修改之前的程序。

限制条件

1. 在线修改状态下，不能修改全局变量表及任何程序的局部变量表，也不能添加或删除任何子程序/中断子程序。
2. 当程序处于在线修改状态下时，如果 PLC 被停止，软件会自动退出在线修改状态。

第九章 变频器子程序库

为了方便使用 PLC 控制变频器，ControlStar2.0 及以后版本中内置了控制变频器的子程序库。这些子程序库可以以指令的形式进行调用（以下简称变频器指令）。通过这些变频器指令，可以方便实现变频器的正转、反转、停车和设置频率等功能。

注意

PLC 本身自带的变频器指令和本章所描述子程序库实现的变频器指令有着本质的区别。前者是内置于 PLC 中的指令，而后者是 ControlStar 的一个可选软件工具包。该软件包是根据 PLC 的通信指令（MODBUS、XMT、RCV）以及变频器的通信协议开发而成的。

9.1 设置变频器

在使用变频器指令和变频器通信之前，必须通过变频器操作面板正确设置变频器的以下参数：

1. 设定变频器通讯口的波特率、奇偶校验、数据位、停止位等，该设置要与 ControlStar 系统块中选择的与变频器通讯的串口设置保持一致。
2. 设定变频器的本机地址，变频器的地址对应 ControlStar 中变频器连接表中设定的站号，当使用 MODBUS 协议时，PLC 的站号不能和变频器的地址重复（PLC 的站号在系统块中设置）。
3. 设定变频器运行命令选择方式为通讯控制。
4. 设定变频器的通讯设定方式为通讯给定方式。
5. 使用自由口协议时，需要在设置变频器的应答延时，延时的长短应根据具体的网络情况确定。

9.2 使用变频器指令的限制

使用变频器指令前必须确保 PLC 拥有以下资源：

1. 子程序资源。变频器指令要占用子程序资源，如果一个 PLC 程序的子程序数量已达到 64 个，则不能使用变频器指令。使用一个不同的变频器指令将占用一个子程序资源。如果反复调用同一型号变频器的同一控制指令将只占用一个子程序资源。因此使用变频器指令前必须确保系统还有子程序资源可供使用。
2. 需要 60 个 D 寄存器资源。变频器指令固定使用最后的 60 个 D 寄存器。对于 EC20 变频器指令固定使用 D7940 ~ D7999，其中 D7940 ~ D7969 作为发送缓冲区段，D7970 ~ D7999 作为接收缓冲区段（具体各指令占用的每个区段的长度不同，请参见各指令说明），可以通过监控这些缓冲区来调试程序。当使用变频器指令时，请不要再使用这些 D 寄存器，以免发生冲突。
3. 如果使用的是自由口协议，调用变频器指令前应确保 SM122 和 SM123 为 OFF。指令执行时会使用到 XMT 和 RCV 指令。执行完成后 SM122 和 SM123 的状态由 XMT 和 RCV 指令决定。如果使用的是 MODBUS 协议，调用变频器指令前应确保 SM135 和 SM136 为 OFF。指令执行时会使用到 MODBUS 指令，执行完成后 SM135 和 SM136 的状态由 MODBUS 指令决定。
4. 变频器指令中使用到了变址寄存器 Z0，虽然在指令退出时会将 Z0 值复原，但是在中断子程序中不应该再使用 Z0 以避免冲突。
5. 变频器指令执行时，如果执行错误，请根据所选择的协议查看 MODBUS 错误标志位或者自由口的错误标志位。变频器指令的执行结果（成功、失败）只有在完成标记被置位时才有效。
6. 使用 MODBUS 协议的变频器指令时，执行一次会需要几次 CPU 扫描才能执行完毕，调用该指令时必须使该指令前面一直有能流存在直到执行完毕，否则会导致以后的调用不能成功。在使用中请注意这一点，比如不要使用上升沿触发调用任何变频器指令，如果已经发生这种情况，请先停机再重新运行。

9.3 如何使用变频器指令编程

编程顺序

1. 首先保证 PLC 系统块设置的通讯口参数包括波特率、奇偶校验、数据位、停止位等与变频器中参数设置的一致。
2. PLC 系统块中设置的协议信息（编程口协议、自由口协议、MODBUS 协议）与变频器连接表中选择的协议要保持一致。
3. 配置变频器连接表。变频器连接表是根据变频器型号、组网协议以及设定的变频器本机地址配置的，比如在 PLC 做主站的 MODBUS 网络中有一台控制发动机的变频器 PV2000（变频器的配置需符合本章 9.1 节的要求），如果该变频器设定的本机地址为 2，则变频器连接表中应配置一条如图 9-1 的数据。其中站号就是变频器的本机地址号，别名可以任意给定，最长可为 8 个汉字，厂商、产品型号以及协议根据实际情况通过下拉列表选择。

	别名	站号	厂商	产品型号	协议
1	PV2000发动机1	2	Emerson	PV2000	MODBUS
2					

图9-1 变频器配置表


注意事项

由于不同型号的变频器其协议可能不同，因此要慎重选择变频器的型号，否则可能导致严重后果，因为 PLC 会根据指令和协议发送不同的数据给变频器。当然如果产品型号的下拉列表中没有所需的变频器型号，而又能确认该变频器的通讯协议与下拉列表中的某个变频器的协议一致，也可以用其代替。

9.4 变频器指令的输入

输入变频器指令的方法与输入应用指令基本类似，但也有细微的差别，下面分向导式输入和文本输入两部分进行介绍。

向导式输入

将光标放在要输入的单元格内，点击  或者点击 **梯形图** 菜单下的 **变频器指令** 菜单项。也可以拖动指令树中的某个变频器指令到目标单元格或者将光标放到目标单元格后双击变频器指令，将弹出如下对话框。

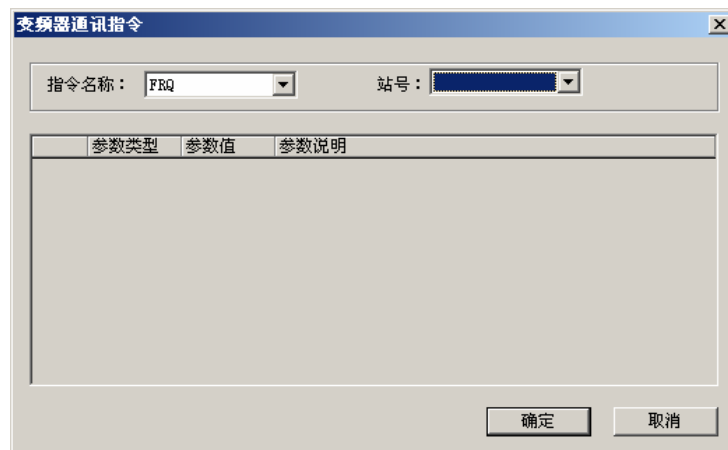


图9-2 变频器通讯指令对话框

首先选择站号，如果所需的站号在下拉列表中不存在，则应检查变频器连接表。输入站号后，将显示该指令的参数，如下所示。

	参数类型	参数值	参数说明
参数1	INT		频率
参数2	BOOL		命令执行成功与否

图9-3 指令参数显示部分

指令参数的含义将在下面的变频器指令中介绍。

需要注意的是对于同一个变频器指令，如果选择的站号不同，其参数也可能会不同，甚至完全不支持某指令，这是因为不同种类的变频器的协议不同。任何一条变频器指令至少有两个参数（标识该指令执行完成与否及其成功与否），如果没有显示输入参数则可能是该变频器不支持某协议。

文本输入

因为不同的协议其变频器指令的参数可能不同，因此除非对变频器的协议和变频器指令非常熟悉，否则建议使用向导输入。比如输入文本 FWD 2 M0 D0，如果 2 号站是 MODBUS 协议，就是正确的，如果是自由口协议则需要输入频率信息。使用文本方式输入变频器指令时回车后不进行校验，如果有错误将在编译时提示。

9.5 变频器指令说明

9.5.1 FRQ

设置变频器频率		适用机型		EC20											
		影响标志位													
指令列表：FRQ (S1) (S2) (D3) (D4)															
操作数	类型	适用软件件													变址
S1	INT	常数	别名												
S2	WORD	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√
D3	BOOL			Y	M	S	LM	SM							
D4	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√

操作数说明

S1 源数据，变频器的站号，在变频器连接表中正确定义的别名或站号。

S2 源数据，要设置的频率值。

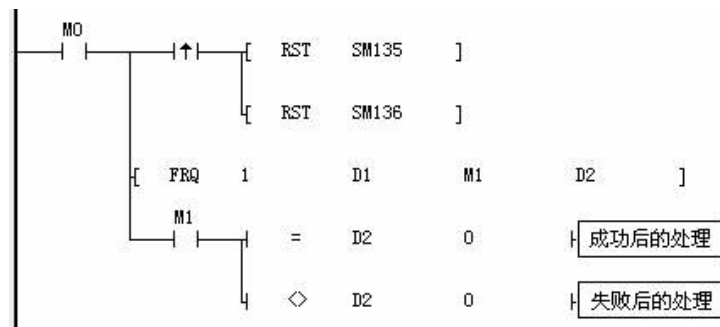
D3 目的数据，表明该指令执行完成与否，执行完成为 ON，未执行完成时为 OFF。

D4 目的数据，指令执行结果，0 表示成功，其他表示失败。

功能说明

设置变频器 S1 的频率为 S2。

使用示例



当 M0 被置位后，设置站号为 1 的变频器的频率为 D1，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

注意事项

该指令仅在 Modbus 协议中使用，发送长度为 6 个字节（D7940 ~ D7945），接收长度为 8 个字节（D7970 ~ D7977）。

9.5.2 FWD

控制变频器正转		适用机型		EC20											
		影响标志位													
使用 MODBUS 协议时：FWD (S1) (D3) (D4)															
使用自由口协议时：FWD (S1) (S2) (D3) (D4)															
操作数	类型	适用软件													变址
S1	INT	常数	别名												
S2	WORD	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√
D3	BOOL			Y	M	S	LM	SM							
D4	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√

操作数说明

S1 源数据，变频器的站号，在变频器连接表中正确定义的别名或站号。

S2 源数据，要设置的频率值。

D3 目的数据，表明该指令执行完成与否，执行完成为 ON，未执行完成时为 OFF。

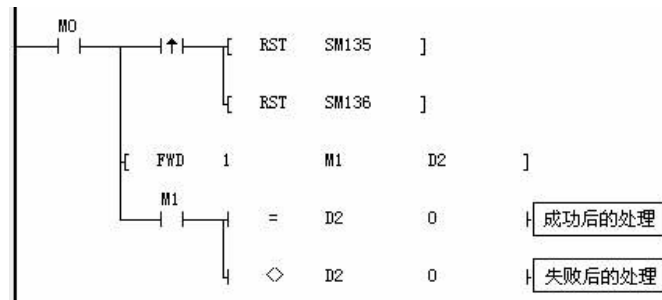
D4 目的数据，指令执行结果，0 表示成功，其他表示失败。

功能说明

使用 MODBUS 协议时，控制变频器 S1 正转。使用自由口协议时，控制变频器 S1 按照 S2 频率正转。

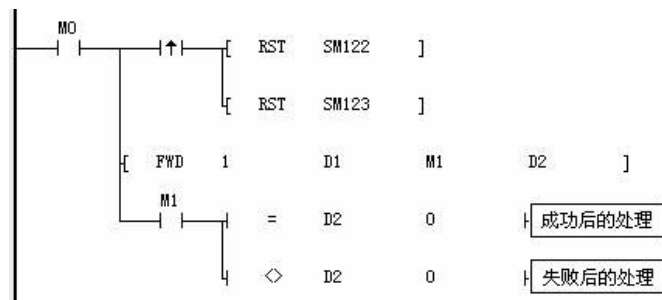
使用示例

1. MODBUS 协议



当 M0 被置位后，控制站号为 1 的变频器正转，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

2. 自由口协议



当 M0 被置位后，控制站号为 1 的变频器按照频率 D1 正转，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

注意事项

Modbus 协议：发送长度为 6 个字节 (D7940 ~ D7945)，接收长度为 8 个字节 (D7970 ~ D7977)。

自由口协议：发送长度为 18 个字节 (D7940 ~ D7957)，接收长度为 18 个字节 (D7970 ~ D7987)。

9.5.3 REV

控制变频器反转		适用机型	EC20													
		影响标志位														
使用 MODBUS 协议时：REV (S1) (D3) (D4)																
使用自由口协议时：REV (S1) (S2) (D3) (D4)																
操作数	类型	适用软元件													变址	
S1	INT	常数	别名													
S2	WORD	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√	
D3	BOOL			Y	M	S	LM	SM								
D4	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√	

操作数说明

S1 源数据，变频器的站号，在变频器连接表中正确定义的别名或站号。

S2 源数据，要设置的频率值。

D3 目的数据，表明该指令执行完成与否，执行完成为 ON，未执行完成时为 OFF。

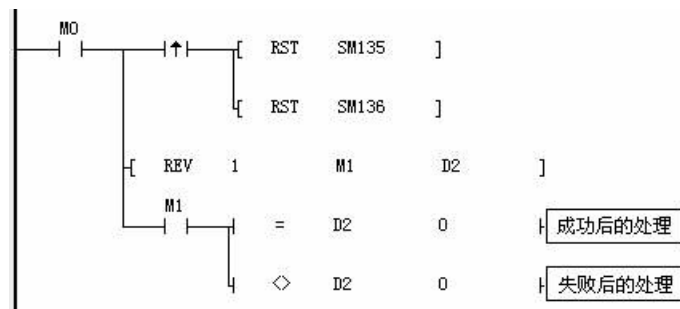
D4 目的数据，指令执行结果，0 表示成功，其他表示失败。

功能说明

使用 MODBUS 协议时，控制变频器 S1 反转。使用自由口协议时，控制变频器 S1 按照 S2 频率反转。

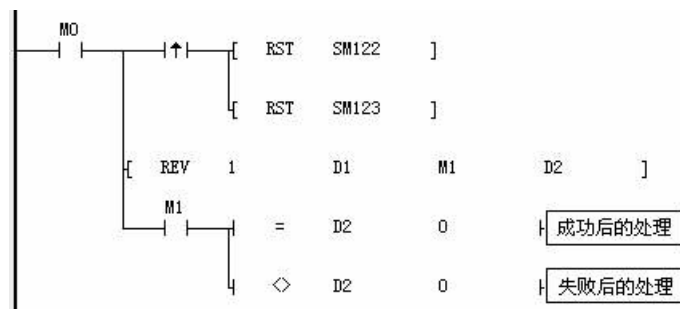
使用示例

1. MODBUS 协议



当 M0 被置位后，控制站号为 1 的变频器反转，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

2. 自由口协议



当 M0 被置位后，控制站号为 1 的变频器按照频率 D1 反转，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

注意事项

Modbus 协议：发送长度为 6 个字节 (D7940 ~ D7945)，接收长度为 8 个字节 (D7970 ~ D7977)。

自由口协议：发送长度为 18 个字节 (D7940 ~ D7957)，接收长度为 18 个字节 (D7970 ~ D7987)。

9.5.4 STP

控制变频器停车		适用机型		EC20											
		影响标志位													
指令列表：STP (S1) (S2) (D3) (D4)															
操作数	类型	适用软元件													变址
S1	INT	常数	别名												
S2	WORD	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√
D3	BOOL			Y	M	S	LM	SM							
D4	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√

操作数说明

S1 源数据，变频器的站号，在变频器连接表中正确定义的别名或站号。

S2 源数据，停车方式：0 表示减速停车，1 表示自由停车，2 表示紧急停车。

D3 目的数据，表明该指令执行完成与否，执行完成为 ON，未执行完成为 OFF。

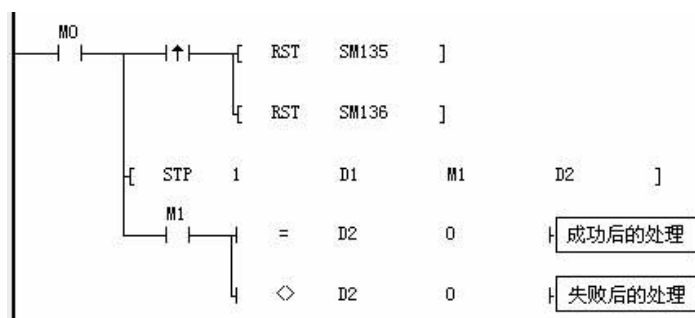
D4 目的数据，指令执行结果，0 表示成功，其他表示失败。

功能说明

控制变频器 S1 按照指定方式停车。

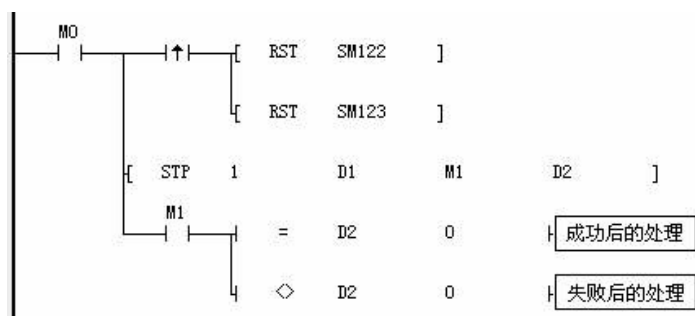
使用示例

1. MODBUS 协议



当 M0 被置位后，控制站号为 1 的变频器按照 D1 方式停车，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

2. 自由口协议



当 M0 被置位后，控制站号为 1 的变频器按照 D1 方式停车，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

注意事项

Modbus 协议：发送长度为 6 个字节 (D7940 ~ D7945)，接收长度为 8 个字节 (D7970 ~ D7977)。

自由口协议：发送长度为 14 个字节 (D7940 ~ D7953)，接收长度为 14 个字节 (D7970 ~ D7983)。

9.5.5 GET

读取变频器参数		适用机型		EC20											
		影响标志位													
指令列表：GET (S1) (S2) (D3) (D4) (D5)															
操作数	类型	适用软件件													变址
S1	INT	常数	别名												
S2	INT	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√
D3	BOOL			Y	M	S	LM	SM							
D4	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√
D5	WORD			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√

操作数说明

S1 源数据，变频器的站号，在变频器连接表中正确定义的别名或站号。

S2 源数据，读取的参数类型：1 表示运行频率，2 表示输出电流，3 表示运行转速，4 表示输出功率。

D3 目的数据，表明该指令执行完成与否，执行完成为 ON，未执行完成为 OFF。

D4 目的数据，指令执行结果，0 表示成功，其他表示失败。

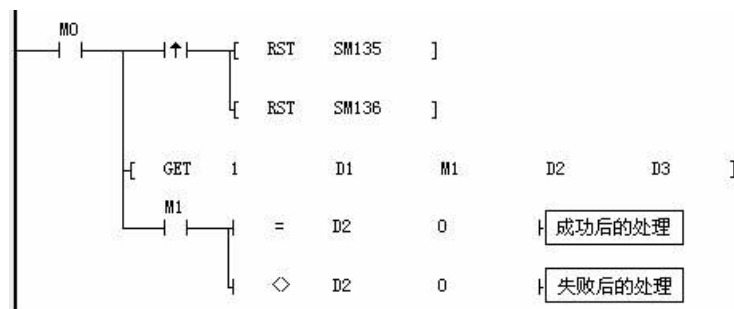
D5 目的数据，读取结果。

功能说明

读取变频器 S1 的 S2 型参数值，结果放到 D5 中。不能通过自由口协议读取 EV1000 和 EV2000 变频器的输出功率。

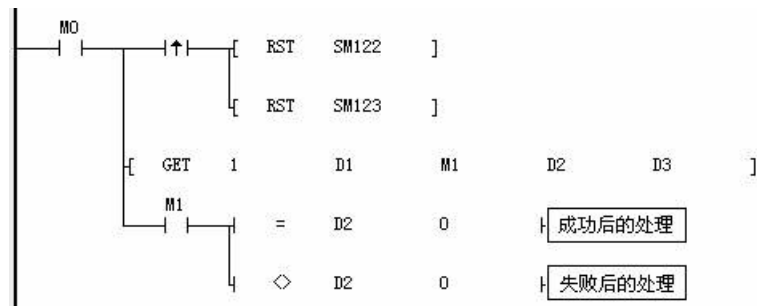
使用示例

1. MODBUS 协议



当 M0 被置位后，读取站号为 1 的变频的类型为 D1 的参数值，将读取的结果放到 D3 中，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

2. 自由口协议



当 M0 被置位后，读取站号为 1 的变频的类型为 D1 的参数值，将读取的结果放到 D3 中，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

注意事项

Modbus 协议：发送长度为 6 字节（D7940～D7945），接收长度为 8 个字节（D7970～D7977）。

自由口协议：发送长度为 14 个字节（D7940～D7953），接收长度为 18 个字节（D7970～D7987）。

9.5.6 RD

读取变频器寄存器的值		适用机型		EC20											
		影响标志位													
指令列表: RD (S1) (S2) (D3) (D4) (D5)															
操作数	类型	适用软元件													变址
S1	INT	常数	别名												
S2	WORD	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√
D3	BOOL			Y	M	S	LM	SM							
D4	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√
D5	WORD			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√

操作数说明

S1 源数据，变频器的站号，在变频器连接表中正确定义的别名或站号。

S2 源数据，读取的寄存器地址。

D3 目的数据，表明该指令执行完成与否，执行完成为 ON，未执行完成为 OFF。

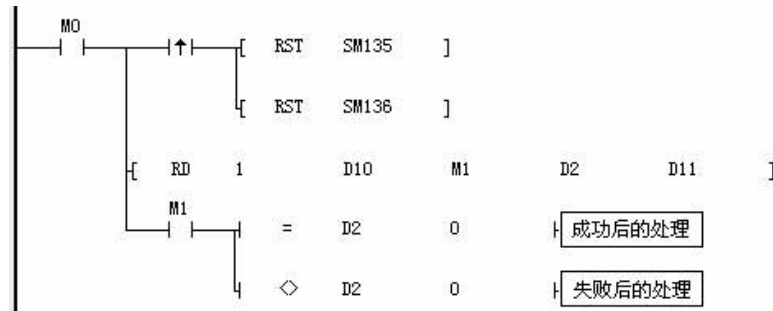
D4 目的数据，指令执行结果，0 表示成功，其他表示失败。

D5 目的数据，读取结果。

功能说明

读取变频器 S1 的 S2 寄存器的值，结果放到 D5 中。

使用示例



当 M0 被置位后，读取站号为 1 的变频的 D10 寄存器的值，将读取的结果放到 D11 中，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

注意事项

该指令仅在 Modbus 协议中使用，发送长度为 6 个字节（D7940 ~ D7945），接收长度为 7 个字节（D7970 ~ D7976）。

9.5.7 WRT

写入变频器的寄存器		适用机型		EC20											
		影响标志位													
指令列表：RD (S1) (S2) (D3) (D4) (D5)															
操作数	类型	适用软元件													变址
S1	INT	常数	别名												
S2	WORD	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√
S3	WORD	常数	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	√
D4	BOOL			Y	M	S	LM	SM							
D5	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	√

操作数说明

S1 源数据，变频器的站号，在变频器连接表中正确定义的别名或站号。

S2 源数据，写入的寄存器地址。

S3 源数据，写入的值。

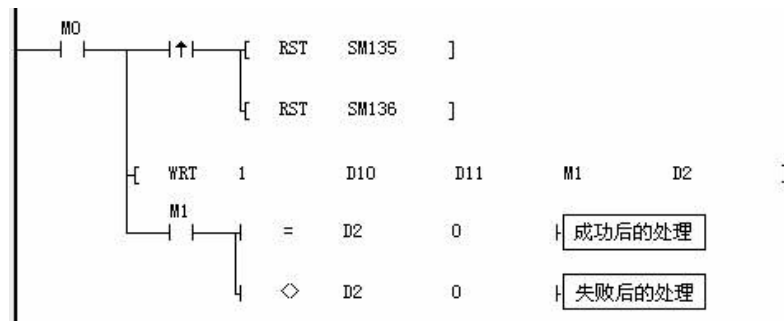
D4 目的数据，表明该指令执行完成与否，执行完成为 ON，未执行完成为 OFF。

D5 目的数据，指令执行结果，0 表示成功，其他表示失败。

功能说明

将变频器 S1 的 S2 寄存器的值写为 S3。

使用示例



当 M0 被置位后，将站号为 1 的变频的 D10 寄存器的值写为 D11，并将完成标志和执行结果分别输出到 M1 和 D2。当执行完成后，根据 D2 的值分别进行成功后的处理和失败后的处理。

注意事项

该指令仅在 Modbus 协议中使用，发送长度为 9 个字节（D7940 ~ D7948），接收长度为 8 个字节（D7970 ~ D7977）。

第十章 PLC 仿真

10.1 仿真软件的启动方式

10.1.1 正常启动

仿真软件是与编程软件 ControlStar 一起安装的，安装完毕后，点击任务栏上的**开始**菜单，选择**开始->所有程序->Emerson Network Power->Controlstar->PLC 仿真**，可以启动仿真软件。

用这种方式启动后，仿真软件处于空白状态，接下来可以新建一套全新的仿真配置，或者打开以前保存在硬盘上的仿真配置。

10.1.2 从 ControlStar 启动

仿真也可以从 ControlStar 中直接启动，当 ControlStar 处于有工程的状态时，选择**工具**下的**工作在仿真状态**菜单，仿真软件就会自动启动起来，并按照当前工程的 PLC 系列类型（例如 EC10、EC20、EC10A 等），自动创建一套与该类型对应的仿真配置，然后用户可以直接使用这套仿真配置进行下载和调试程序。

注意

自动创建仿真配置时，主模块使用的是某系列中默认的主模块，如果点数不能符合要求，用户可以再手动切换主模块型号。

10.1.3 从资源管理器启动

如果以前曾将某个仿真配置保存到了硬盘文件中，可以从资源管理器中直接双击相应得仿真配置文件（*.smp 文件）打开该文件，此时仿真软件将读取最后一次保存过的配置内容，并自动载入最后一次下载到这个仿真配置中的用户程序。

10.2 关于仿真配置

10.2.1 新建配置

仿真启动后，选择**文件**菜单下的**新建**菜单，会弹出**新建仿真**对话框，如下图所示：



图10-1 新建仿真对话框

在此对话框中用户可以指定该配置所要仿真的 PLC 系列（如 EC10、EC20 等），并选择一个合适 I/O 点数的主模块，点击**确定**键后，将出现仿真图形界面，如下图所示：

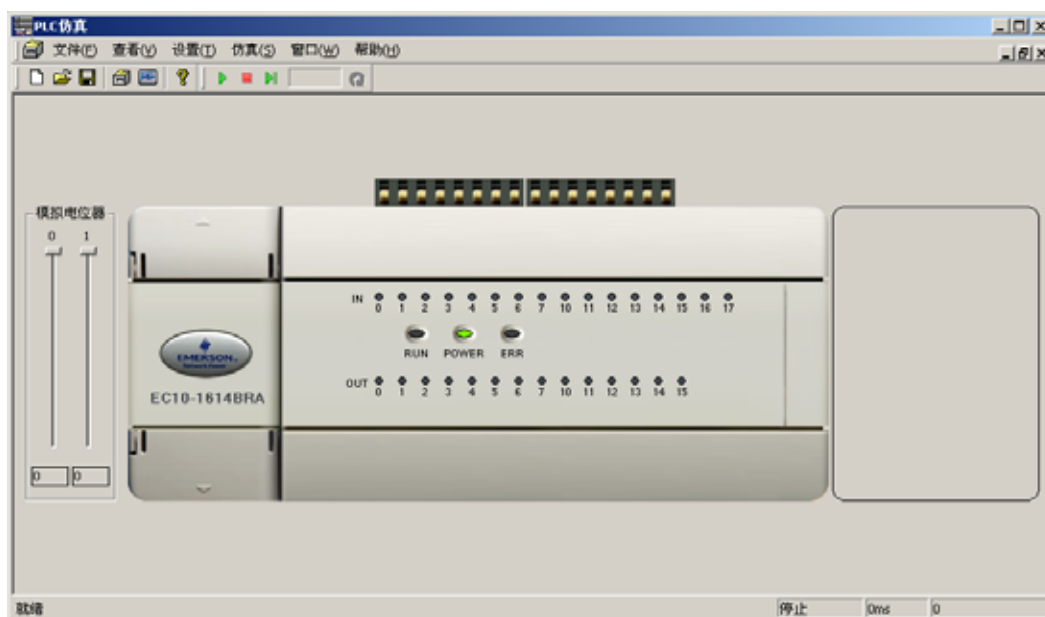


图10-2 PLC 仿真图形界面

10.2.2 打开配置

如果以前曾经将某个仿真配置保存到硬盘文件（*.smp 文件），则可以通过“文件”菜单下的“打开”功能，将该仿真配置重新读取出来。

10.2.3 保存配置

用户可以选择将仿真配置保存到硬盘文件，以备日后继续使用。当需要保存时，选择**文件->保存**。此时如果该仿真配置已经保存过，则直接保存到上次指定的文件中，否则将弹出对话框提示用户选择要保存的位置。

保存到磁盘的仿真配置文件包含以下内容：

- 当前配置的 PLC 系列、主模块、扩展模块
- 用户下载的程序
- 用户下载的系统块配置
- 用户下载的数据块配置
- 掉电保存元件的值
- 时序图中的所有元件地址

10.3 仿真界面说明

10.3.1 仿真图介绍

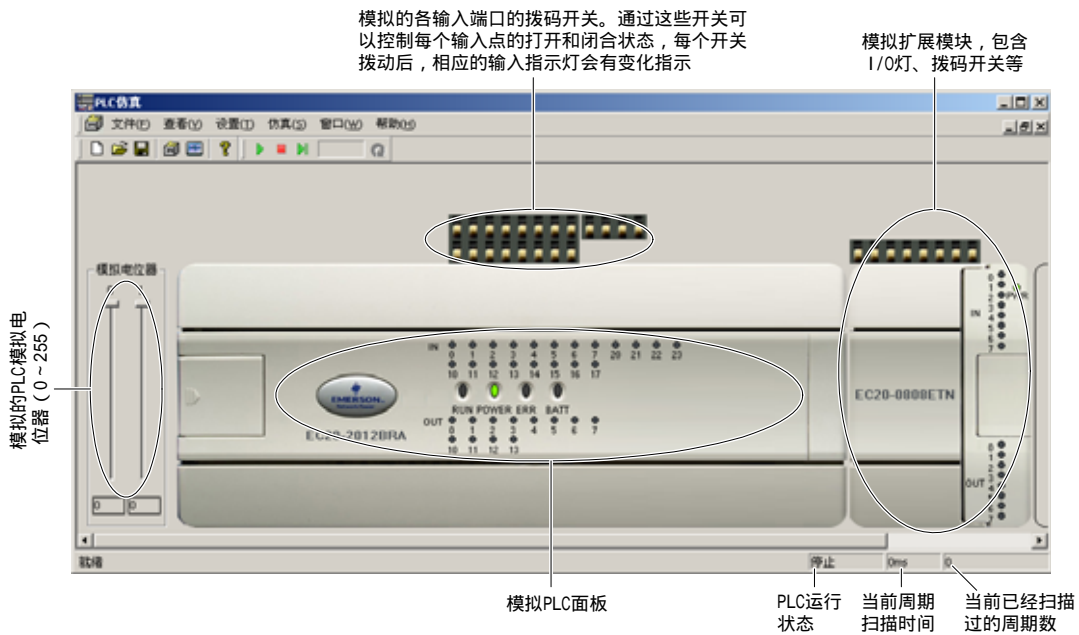



图10-3 PLC 仿真图形界面

某个仿真配置新建或打开后，默认都会打开仿真图窗口。此窗口可以随时关闭，关闭后再打开可以通过选择**查看->仿真图**，或点击工具栏中的按钮。

10.3.2 扩展模块

仿真配置创建出来后，默认只有主模块，没有扩展模块，如果需要扩充 I/O 点数，可以手动添加扩展模块。初始情况下，主模块后面紧跟着一个扩展模块空位，在此空位上双击鼠标左键，会弹出**扩展模块**对话框，如下图所示：




图10-4 扩展模块对话框


用户可以在此对话框中指定要在当前位置放置某种扩展模块，如果用户在空位上新增了某个扩展模块，则系统会自动在后面新增一个空位以备下次扩展（在扩展模块数量允许的情况下）。

在主模块或者已存在的某个扩展模块上双击鼠标左键，可以切换模块类型，选择**无**可以删除该模块（主模块不能删除）。

10.3.3 仿真运行状态控制

可以通过**仿真**菜单下的各菜单项或控制工具栏中的各个按钮控制仿真 PLC 的运行和停止。

除正常的运行和停止状态外，仿真软件还提供了**调试运行**的方式，可以更方便的调试用户程序。使用此方式前，应先确保仿真 PLC 处于停止状态，然后选择**仿真->调试**或点击按钮，此后仿真 PLC 将进入调试暂停状态（从状态栏可以看到，此时 PLC 不执行用户程序，但是各元件值都会保持），此时在控制工具栏中的输入框中输入希望运行的周期

数，然后选择**仿真->继续**或点击按钮，则仿真 PLC 将会在连续运行指定周期数后再次暂停。使用这种方式，就可以观察到各元件在每一个或每几个周期运行结束后的值，从而判断程序运行是否正确。

10.3.4 高速输入设置

当需要模拟高速脉冲输入时，请先确保仿真 PLC 处于停止状态，然后在仿真图界面下选择**设置->高速输入**，此时将弹出**高速输入设置**对话框，如下图所示：

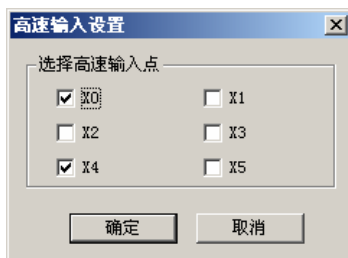


图10-5 高速输入设置对话框

用户可以在此设置 X0~X5 这 6 个输入点中，哪些需要模拟高速输入。如上图中选择了 X0 和 X4，点击确定后，在原仿真图的输入按钮区域上方，会出现输入频率设置滑动条，相应的输入点外观也将变换为脉冲输入模拟图。见下图：




图10-6 输入点外观

此时原来控制输入点打开和闭合的开关则变为控制高速输入脉冲的启动和停止，在脉冲输入启动后，脉冲输入图标将会以定时闪烁以提示当前正在输入。通过个滑动条可以设置某路输入的脉冲频率（以 K 为单位调节），将鼠标移至某个滑动条上方可以看到脉冲输入的确切频率，也可以使用键盘的上下箭头键及 PageDown 或 PageUp 键进行精确调节。

10.4 时序图说明

10.4.1 时序图介绍

使用时序图，可以观察元件值随扫描周期变换的曲线，从而判断用户程序的运行过程。时序图配合调试运行模式，对调试程序逻辑的正确性可以起到很大作用。

仿真软件启动后，默认时序图窗口是不打开的。需要打开时，可通过选择**查看->时序图**，或点击工具栏中的按钮。

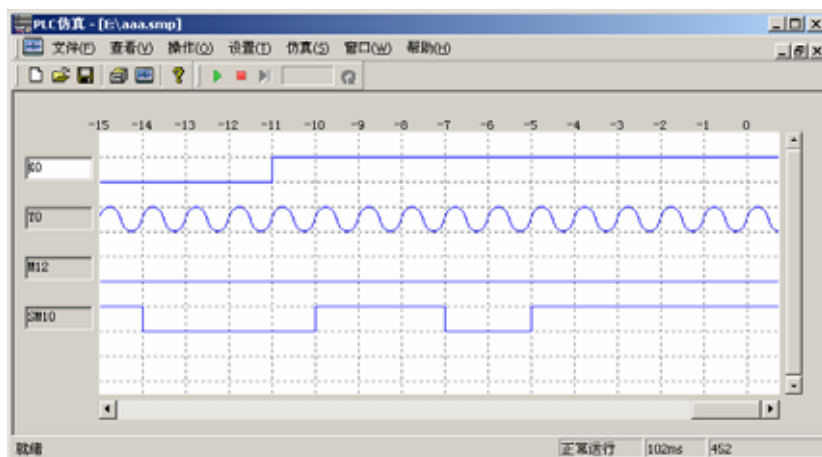


图10-7 时序图窗口

典型时序图窗口如图 10-7：

- 左边列出了所有采样的元件，网格图中则显示了每个元件随每个扫描周期的变换过程
- 网格图中的每条竖线表示了一个扫描周期，从上方的数字中可以看出每条线对应的是第几个周期。随着程序的运行，网格图向左逐渐推进，最右面固定为第“0”周期，即当前周期，其左边的第一个周期为“-1”周期，代码刚刚过去的那个周期，依次论推，越左边的周期表明时间越靠前。
- 每个元件在网格图中都对应相邻的上下两条横线，分别对应该元件的 0 和 1 值，将每个周期的值连接起来，即可看出该元件在运行期间整个变换过程。
- 对于用作高速输出的 Y 端口，有特殊的表示方式，当该端口有脉冲输出时，其图形为正弦波曲线，没有输出则为直线。

通过选择**操作->采样**，可以启动或停止采样过程。只有当采样启动时，图中的元件数据才会更新，并且在采样过程中，图中只显示最近的若干扫描周期（具体周期数与窗口大小有关）的变化情况。当停止采样后，可以拖动横向滚动条查看之前的扫描周期的变化情况。

时序图中可以存储最近 4000 个扫描周期的数据，4000 个周期之前的数据会被抛弃。并且，在每次开始采样之前，会先清除图中的现有数据。

10.4.2 添加删除采样元件

在时序图窗口中，通过**操作**菜单下的各菜单项，可以添加或删除需要采样的元件，包括**添加元件**、**插入元件**、**删除元件**和**删除所有元件**。其中添加元件和插入元件的区别在于：采用添加元件时，新元件将放在元件列表的最后；而采用插入元件时，新元件将放在当前选择的元件前面。新增元件的对话框如下图所示：



图10-8 新增监控元件对话框

采样元件仅限于位元件，包括：X、Y、M、S、T、C、SM，且同时采样的元件数量不能超过 64 个。

10.4.3 时序图相关设置

1. 显示比例

在时序图窗口中，通过选择**设置->显示比例**，可以设置图形比例。此处的图形比例是指网格图中相邻两条竖线之前代表的周期数目，显示比例最大为 40，最小为 1。显示比例越小，所显示的数据变化过程越详细。

改变了显示比例后，网格图上方的数字会显示新的比例数字。下图就是显示比例设置为 20 时的图形：

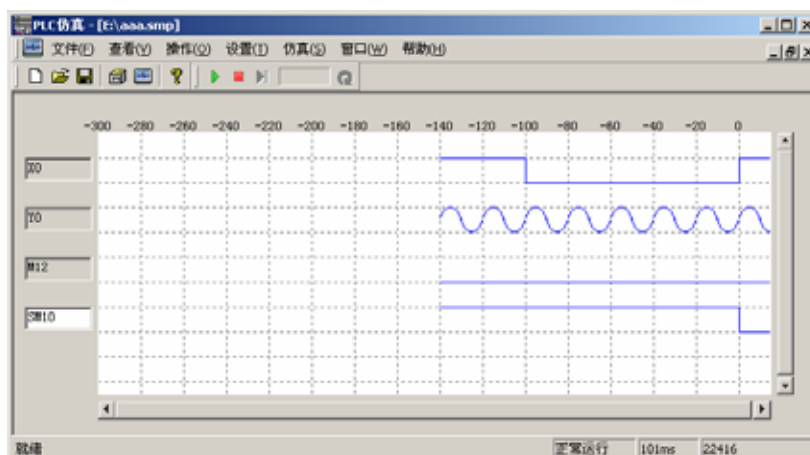


图10-9 显示比例设置为 20 时的时序图

2. 扫描时间

仿真软件运行时，会受到计算机硬件资源和同时运行的其他软件的影响，扫描周期常会有较大变化，并且当扫描周期很短时，时序图会由于更新过快而无法看清楚数据的变化。因此，如果需要比较精确的查看变化曲线，建议在系统块中设置一个比较大的恒定扫描时间，这样可以使得图形更加稳定也更加准确。

10.5 PLC 仿真的通讯功能

PLC 仿真软件单独运行时,可以模拟 PLC 运行用户程序、模拟 PLC 输入输出及其面板指示灯等，除了单独运行外，PLC 仿真还可以与 Controlstar 进行通讯，由 Controlstar 监控仿真的执行情况。

10.5.1 Controlstar 建立与仿真的连接

在 Controlstar 的运行环境内点击**工具->工作在仿真状态**，可以连接当前已经启动的仿真软件。如果仿真软件还没有运行，则启动默认的仿真器。默认的仿真器的型号与当前工程的 PLC 型号相同，I/O 点数均为默认值。如果 Controlstar 与仿真连接成功，则在**信息输出窗口**显示**仿真连接成功**（如下图所示），同时子菜单**工作在仿真状态**会被选中打钩（ **工作在仿真状态(W)**）。

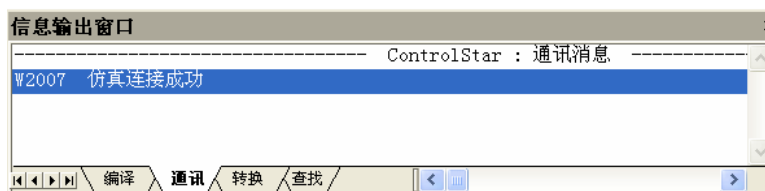


图10-10 仿真连接成功时的信息输出

连接成功后，即可象使用真实的 PLC 一样操作。

10.5.2 PLC 仿真的监控

当 ControlStar 与 PLC 仿真软件连接成功后，ControlStar 即可与 PLC 仿真软件进行通信并对其进行监控。

PLC 仿真的监控的实质是 Controlstar 与 PLC 仿真软件之间数据交换的过程。在监控过程中，Controlstar 读取 PLC 仿真软件的数据；而 PLC 监控则是 Controlstar 通过串口通信协议读取 PLC 数据的过程；当用户在 Controlstar 内点击监控命令时，Controlstar 首先检测当前的配置如果是工作在仿真状态，则与 PLC 仿真软件进行数据交互，否则，发送通信命名到串口。

10.6 PLC 仿真和 PLC 区别

PLC 仿真程序主要用于调试用户程序逻辑的正确性，以及方便用户在没有 PLC 的情况下学习和了解 PLC 的使用，实现了模拟 PLC 运行用户程序的功能，包括子程序和中断子程序；模拟 PLC 输入、输出端子及面板指示灯、支持 PLC 所有

的逻辑运算、数据运算指令和部分高速、外设等指令；支持运行状态控制、扫描周期控制、实时时钟、IO 控制等系统功能；支持系统块、数据块配置；仿真程序可以与编程软件（ControlStar）进行通讯，使编程软件可以像操作真正的 PLC 一样，进行下载、上载、控制、监控等功能。

由于 PLC 仿真侧重于调试用户程序的逻辑性，在具体指令方面，PLC 的某些指令 PLC 仿真不支持，具体来说，PLC 仿真不支持以下指令：

1. 通讯指令（MODBUS、XMT、RCV）
2. 定位指令（ABS、ZRN、PLSV、DRVI、DRVA）
3. 变频器指令（EVFWD、EVREV、EVDREV、EVDREV、EVSTOP、EVFRQ、EVWRT、EVRDST、EVRD）
4. 外设指令（FROM、DFROM、TO、DTO、REF、REF、EROMWR）
5. 程序流控制指令（WDT）

仿真程序支持的中断也与 PLC 有一定的差别，主要支持 X0-X7 的上升沿与下降沿中断，高速计数器中断、定时中断等，但不支持以下中断类型：

1. 失电中断
2. 串口中断

10.7 使用仿真调试程序的实例

以下示例模拟一个水池的进水阀和排水阀。其开排水阀即可以手动控制，也可以自动运行。其中：

1. 开进水阀的每秒流量为 10 升
2. 开排水阀的每秒流量为 20 升
3. 自动运行，开进水阀关排水阀，当水量达到 2500 升时，关进水阀，开排水阀
4. 自动运行，开排水阀时，当水量低于 500 升时，关排水阀开进水阀

如图 10-11 所示：

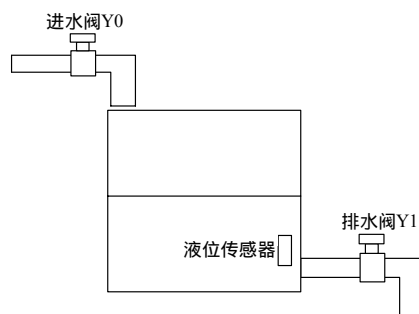


图10-11 水池进出水阀模拟图

编写梯形图程序如下所示。其中：X0 表示手动开进水阀门，X1 表示手动开排水阀门，X2 为自动手动切换开关，Y0 为进水阀控制，Y1 为排水阀控制，程序中 N 4 和 N 5 模拟的是一个测试水量的模拟量输入模块，相当于将水池的水量（液位传感器）读入到 D0 元件中，程序编写完成后，即可以使用仿真进行调试了。

1. 首先启动仿真，选择**工具->工作在仿真状态**。
2. 点击**下载**，将下面的梯形图程序下载到仿真软件中。

下载完成后就可以像监控 PLC 一样监控仿真了，同时也可以仿真界面上输入 X0、X1、X2 的值。

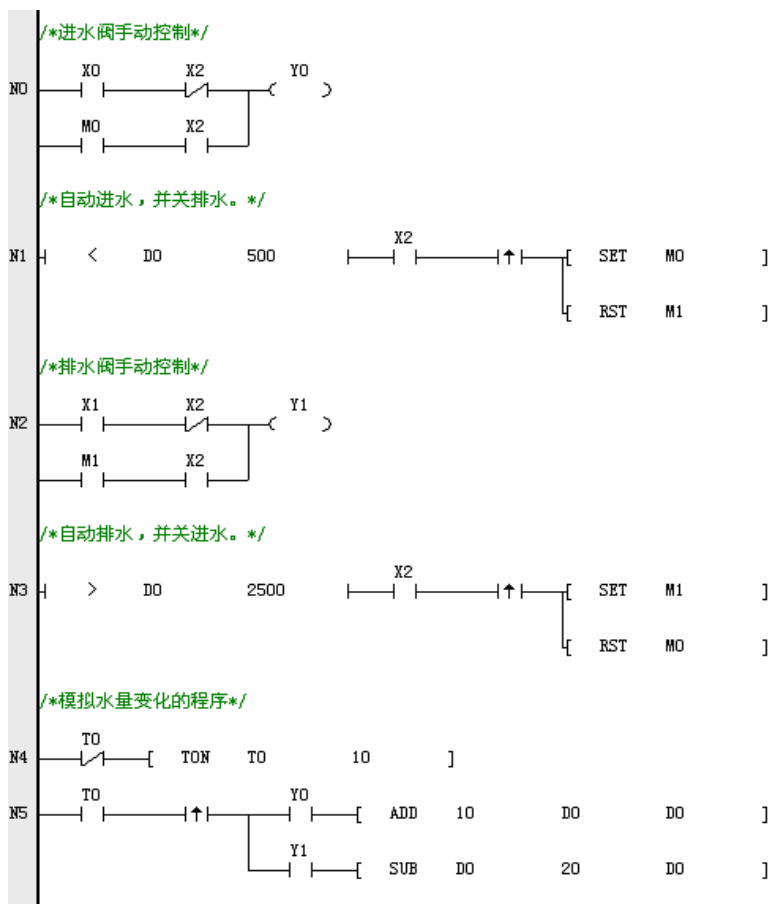


图10-12 梯形图例程

附录一 软元件说明

1. 软元件定义

软元件类型及属性说明见表 1。不同类型 PLC 的软元件地址范围见表 2。

表1 软元件类型及属性

元件标号	元件名	位元件	字元件	功能说明
X	输入继电器			反映输入端口的输入状态
Y	输出继电器			控制输出端口的输出状态
M	辅助继电器			存储位逻辑运算的中间运算结果
LM	局部辅助继电器			存储局部有效的位变量 子程序调用时用于传递位型参数
S	步进状态继电器			用于顺序控制，代表步进状态
SM	特殊辅助继电器			系统标志位和控制位
T	计时器			用于计时功能： 作为字元件记录 16 位的计时值 作为位元件反映计时器线圈状态
C	计数器			用于计数功能： 作为字或双元件记录 16 位或 32 位的计数值 作为位元件反映计数器线圈状态
D	数据寄存器			存储 16 位或 32 位数据
V	局部数据寄存器			存储局部有效的字变量 子程序调用时用于传递字型参数
Z	变址寻址寄存器			用于变址寻址，存储地址偏移量
SD	特殊数据寄存器			存储系统状态字和控制字

表10-1表 2 不同类型 PLC 的软元件地址范围

元件	子项	EC10	EC20
X		X0 ~ X177 (8 进制编码)	X0 ~ X377 (8 进制编码)
Y		X0 ~ X177 (8 进制编码)	X0 ~ X377 (8 进制编码)
M		M0 ~ M2047	M0 ~ M1999
LM		LM0 ~ LM63	LM0 ~ LM63
S		S0 ~ S1023	S0 ~ S991
SM		SM0 ~ SM255	SM0 ~ SM255
T	100ms 精度	T0 ~ T209	T0 ~ T209
	10ms 精度	T210 ~ T251	T210 ~ T251
	1ms 精度	T252 ~ T255	T252 ~ T255
C	16 位增计数器	C0 ~ C199	C0 ~ C199
	32 位增减计数器	C200 ~ C235	C200 ~ C235
	32 位高速计数器	C236 ~ C255	C236 ~ C255
D		D0 ~ D7999	D0 ~ D7999
V		V0 ~ V63	V0 ~ V63
Z		Z0 ~ Z15	Z0 ~ Z15
SD		SD0 ~ SD255	SD0 ~ SD255

2. 元件的间接寻址方式

提供变址寻址方式，局部数据寄存器 Z 为变址寄存器使用。

位元件的变址寻址例举

```
LD    M1
MOV   6 Z1
SFTR  X0Z1 M0 8 2
```

以上命令实际等价于：

```
LD    M1
SFTR  X6 M0 8 2
```

其寻址过程如下：

Z1=6

$X0Z1 = X(0 + Z1) = X6$

位串的变址寻址例举

```
LD    M1
MOV   3 Z10
MOV   K1X0Z10 D0
```

以上命令等价于：

```
LD    M1
MOV   K1X3 D0
```

其寻址过程如下：

Z10=3

$K1X0Z10 = K1X(0 + Z10) = K1X3$

字元件的变址寻址例举

```
LD    M1
MOV   30 Z20
MOV   D100Z20 D0
```

以上命令等价于：

```
LD    M1
MOV   D130 D0
```

其寻址过程如下：

Z20=30

$D100Z20 = D(100 + Z20) = D130$

附录二 指令一览表

本处列出指令一览表，指令详细描述请参考《EC20系列可编程控制器编程参考手册》。

类别	指令名称	功能说明	备注	
基本指令	LD	常开触点能流装载		
	LDI	常闭触点能流装载		
	AND	常开触点能流与		
	ANI	常闭触点能流与		
	OR	常开触点能流或		
	ORI	常闭触点能流或		
	OUT	能流输出		
	SET	置位		
	RST	清零		
	ANB	能流块与		
	ORB	能流块或		
	INV	能流取反		
	NOP	空操作		
	MPS	输出能流入栈		
	MRD	读输出能流栈顶值		
	MPP	输出能流栈出栈		
	EU	能流上升沿检测		
	ED	能流下降沿检测		
	TON	接通延时计时指令		
	TOF	断开延时计时指令		
	TMON	单稳计时指令		
	TONR	接通延时记忆计时指令		
	CTU	16位计数器增计数指令		
	CTR	16位计数器循环计数指令		
	DCNT	32位计数指令		
	程序流控制指令	LBL	跳转标号定义	
		CJ	条件跳转	
		CALL	用户子程序调用	
CSRET		用户子程序条件返回		
CFEND		用户主程序条件结束		
CIRET		用户中断子程序条件返回		
FOR		循环指令		
NEXT		循环返回		
MC		主控		
MCR		主控清除		
WDT		用户程序看门狗清零		
STOP		用户程序停止		
EI		中断使能指令		
DI	中断禁止指令			
顺序功能图指令	STL	顺序功能图状态装载指令		
	SET Sxx	顺序功能图状态转移		
	OUT Sxx	顺序功能图状态跳转		
	RST Sxx	顺序功能图状态清除		
	RET	顺序功能图程序结束		

类别	指令名称	功能说明	备注
数据传输指令	MOV	字数据传输	
	DMOV	长字数据传输指令	
	RMOV	浮点数数据传输指令	
	BMOV	块数据传输指令	
	SWAP	上下字节交换指令	
	XCH	字交换指令	
	DXCH	长字换指令	
	FMOV	块数据填充	
	DFMOV	数据块双字填充指令	
	WSFR	字串右移指令	
	WSFL	字串左移指令	
	PUSH	加入队列	
	FIFO	先入先出指令	
	LIFO	后入先出指令	
	整数/长整数算术运算指令	ADD	整数加法指令
DADD		长整数加法指令	
SUB		整数减法指令	
DSUB		长整数减法指令	
INC		整数增一指令	
DINC		长整数增一指令	
DEC		整数减一指令	
DDEC		长整数减一指令	
MUL		整数乘法指令	
DMUL		长整数乘法指令	
DIV		整数除法指令	
DDIV		长整数除法指令	
VABS		整数绝对值指令	
DVABS		长整数绝对值指令	
NEG		整数取负指令	
DNEG		长整数取负指令	
SQT		整数平方根指令	
DSQT	长整数平方根指令		
浮点算术运算指令	RADD	浮点数加法指令	
	RSUB	浮点数减法指令	
	RMUL	浮点数乘法指令	
	RDIV	浮点数除法指令	
	RVABS	浮点数绝对值指令	
	RNEG	浮点数取负指令	
	RSQT	浮点数平方根指令	
	SIN	浮点数 SIN 指令	
	COS	浮点数 COS 指令	
	TAN	浮点数 TAN 指令	
	LN	浮点数自然对数指令 LN	
	EXP	浮点数自然数幂指令 EXP	
POWER	浮点数求幂指令		

类别	指令名称	功能说明	备注
累加器指令	SUM	16位累加器	
	DSUM	32位累加器	
	RSUM	浮点累加器	
字/长字逻辑运算	WAND	字与指令	
	DWAND	长字与指令	
	WOR	字或指令	
	DWOR	长字或指令	
	WXOR	字异或指令	
	DWXOR	长字异或指令	
	WINV	字取非指令	
移位转指令	DWINV	长字取非指令	
	ROR	字位循环右移指令	
	DROR	长字位循环右移指令	
	ROL	字位循环左移指令	
	DROL	长字位循环左移指令	
	RCR	字位带进位循环右移指令	
	DRCR	长字位带进位循环右移指令	
	RCL	字位带进位循环左移指令	
	DRCL	长字位带进位循环左移指令	
	SHR	字右移指令	
	DSHR	长字右移指令	
	SHL	字左移指令	
	DSHL	长字左移指令	
	SFTL	位串左移指令	
SFTR	位串右移指令		
增强型位处理指令	DECO	解码指令	
	ENCO	编码指令	
	BITS	字中ON位统计指令	
	DBITS	长字中ON位统计指令	
	ZRST	批量位清零指令	
ZSET	批量位置位指令		
高速IO	HCNT	高速计数器驱动指令	
	DHSCS	高速计数比较置位指令	
	DHSCR	高速计数比较复位指令	
	DHSCI	高速计数比较中断触发指令	
	DHSZ	高速计数区间比较指令	
	DHST	高速计数表格比较指令	
	DHSP	高速计数表格比较脉冲输出指令	
	SPD	测频指令	
	PLS	包络线脉冲输出指令	仅 EC10 支持
	PLSY	计数脉冲输出指令	
PLSR	带加减速的计数脉冲输出指令		
PWM	PWM 脉冲输出指令		
控制计算指令	PID	PID 功能指令	
	RAMP	斜坡信号输出指令	
	HACKLE	锯齿波发生器指令	
	TRIANLE	三角波函数发生器指令	
外设指令	FROM	字读特殊模块缓冲寄存器指令	仅 EC20 支持
	DFROM	长字读特殊模块缓冲寄存器指令	
	TO	字写特殊模块缓冲寄存器指令	
	DTO	长字写特殊模块缓冲寄存器指令	
	VRRD	读模拟电位器值指令	
外设指令	REFF	设置输入滤波常数指令	
	REF	刷新 I/O 镜像指令	
	EROMWR	EEPROM 写指令	仅 EC10 支持
实时时钟指令	TRD	实时时钟读指令	
	TWR	实时时钟写指令	
	TADD	时间加指令	
	TSUB	时间减指令	
	HOURL	小时计指令	
	DCMP=	日期等于比较	
日期时间比较指令	DCMP<	日期小于比较	
	DCMP>	日期大于比较	
	DCMP<=	日期小于等于比较	
	DCMP>=	日期大于等于比较	
	DCMP<>	日期不等于比较	
	TCMP=	时间等于比较	
	TCMP<	时间小于比较	
	TCMP>	时间大于比较	
	TCMP<=	时间小于等于比较	
	TCMP>=	时间大于等于比较	
TCMP<>	时间不等于比较		
比较触点指令	LD=	整数比较 LD=指令	
	LDD=	长整数比较 LD=指令	
	LDR=	浮点数比较 LD=指令	
	LD>	整数比较 LD>指令	
	LDD>	长整数比较 LD>指令	
	LDR>	浮点数比较 LD>指令	
	LD>=	整数比较 LD>=指令	
	LDD>=	长整数比较 LD>=指令	
	LDR>=	浮点数比较 LD>=指令	
	LD<	整数比较 LD<指令	
	LDD<	长整数比较 LD<指令	
	LDR<	浮点数比较 LD<指令	
	LD<=	整数比较 LD<=指令	
	LDD<=	长整数比较 LD<=指令	
	LDR<=	浮点数比较 LD<=指令	
	LD<>	整数比较 LD<>指令	
	LDD<>	长整数比较 LD<>指令	
	LDR<>	浮点数比较 LD<>指令	
	AND=	整数比较 AND=指令	
	ANDD=	长整数比较 AND=指令	
	ANDR=	浮点数比较 AND=指令	
	AND>	整数比较 AND>指令	
	ANDD>	长整数比较 AND>指令	
	ANDR>	浮点数比较 AND>指令	
AND>=	整数比较 AND>=指令		
ANDD>=	长整数比较 AND>=指令		
ANDR>=	浮点数比较 AND>=指令		
AND<	整数比较 AND<指令		
ANDD<	长整数比较 AND<指令		
ANDR<	浮点数比较 AND<指令		
AND<=	整数比较 AND<=指令		
ANDD<=	长整数比较 AND<=指令		

类别	指令名称	功能说明	备注
比较 触点 指令	ANDR<=	浮点数比较 AND<=指令	
	AND<>	整数比较 AND<>指令	
	ANDD<>	长整数比较 AND<>指令	
	ANDR<>	浮点数比较 AND<>指令	
	OR=	整数比较 OR=指令	
	ORD=	长整数比较 OR=指令	
	ORR=	浮点数比较 OR=指令	
	OR>	整数比较 OR>指令	
	ORD>	长整数比较 OR>指令	
	ORR>	浮点数比较 OR>指令	
	OR>=	整数比较 OR>=指令	
	ORD>=	长整数比较 OR>=指令	
	ORR>=	浮点数比较 OR>=指令	
	OR<	整数比较 OR<指令	
	ORD<	长整数比较 OR<指令	
	ORR<	浮点数比较 OR<指令	
	OR<=	整数比较 OR<=指令	
	ORD<=	长整数比较 OR<=指令	
	ORR<=	浮点数比较 OR<=指令	
	OR<>	整数比较 OR<>指令	
ORD<>	长整数比较 OR<>指令		
ORR<>	浮点数比较 OR<>指令		
数值 转换 指令	ITD	整数转换长整数指令	
	DTI	长整数转换整数指令	
	FLT	整数转换浮点数指令	
	DFLT	长整数转换浮点数指令	
	INT	浮点数转换整数指令	
	DINT	浮点数转换长整数指令	
	BCD	整数转换为 BCD 整数指令	
	DBCD	长整数转换为 BCD 整数指令 DBCD	
	BIN	BCD 整数转换整数指令	
	DBIN	BCD 整数转换整数指令	
	GRY	整数转换为格雷码指令	
	DGRY	长整数转换为格雷码指令	
	GBIN	格雷码转换为整数指令	
	DGBIN	32 位格雷码转换为长整数指令	
	SEG	BCD 码转换位 7 段码	
	ASC	ASCII 码转换指令	
	ATI	ASCII 码数转换 16 位 16 进制指令	仅 EC10 支持
ITA	16 位 16 进制数转换 ASCII 码指令		

类别	指令名称	功能说明	备注	
字触 点指 令	BLD	字位触点 LD 指令		
	BLDI	字位触点 LDI 指令		
	BAND	字位触点 AND 指令		
	BANI	字位触点 ANI 指令		
	BOR	字位触点 OR 指令		
	BORI	字位触点 ORI 指令		
	BSET	字位触点置位指令		
	BRST	字位触点清除指令		
	BOUT	字位触点输出指令		
	XMT	自由口发送 (XMT) 指令		
	通信 指令	RCV	自由口发送 (RCV) 指令	
		MODBUS	modbus 协议指令	
CCITT		CCITT 校验指令		
数据 校验 指令	CRC16	CRC16 校验指令		
	LRC	LRC 校验指令		
定位 指令	ABS	ABS 当前值读取指令	仅 EC10 支持	
	ZRN	原点回归指令		
	PLSV	可变速脉冲输出指令		
	DRVI	相对位置控制指令		
	DRVA	绝对位置控制指令		
变频 器指 令	EVFWD	变频器正转指令	仅 EC10 支持	
	EVREV	变频器反转指令		
	EVDFWD	变频器点动正转指令		
	EVDREV	变频器点动反转指令		
	EVSTOP	变频器停止指令		
	EVFRQ	设置变频器频率指令		
	EVWRT	写变频器单个寄存器值指令		
	EVRDST	读取变频器状态指令		
EVRD	读变频器单个寄存器值指令			
变频 器子 程序 库	FWD	变频器正转指令		
	REV	变频器反转指令		
	STP	变频器停车指令		
	FRQ	设置变频器频率指令		
	GET	读取变频器参数指令		
	RD	读取变频器寄存器指令		
	WRT	写入变频器寄存器指令		