

21 世纪高等院校电气信息类系列教材

S7-200 PLC 编程及应用

第 3 版

廖常初 主编



机械工业出版社

本书全面介绍了 S7-200 的硬件结构、指令系统和编程软件的使用方法；通过大量的实例，介绍了功能指令的使用方法；介绍了设计数字量控制梯形图的顺序控制设计法，这种设计方法易学易用，可以节约大量的设计时间；介绍了 S7-200 的通信网络、通信功能和通信程序的设计方法，以及 PLC 之间、PLC 与变频器之间的通信的编程和实现的方法；还介绍了 PID 控制和 PID 参数的整定方法和提高系统可靠性的措施。各章均配有习题，附录中有 30 多个实验的实验指导书。

读者可以扫描本书封底的二维码，获取下载链接，下载本书配套的 S7-200 编程软件、指令库、触摸屏组态软件、PLC 串口通信调试软件、用户手册和产品样本、30 多个免费视频教程和 40 多个例程。

本书可以作为大专院校的电类和机电一体化专业的教材，也可以供工程技术人员自学。

本书有配套的电子课件，需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载，或联系编辑索取（QQ：1239258369，电话 010-88379739）。

书在版编目（CIP）数据

S7-200 PLC 编程及应用 / 廖常初主编. —3 版. —北京：机械工业出版社，2019.5

21 世纪高等院校电气信息类系列教材

ISBN 978-7-111-62565-0

I. ①S… II. ①廖… III. ①PLC 技术—高等学校—教材 IV. ①TM571.61

中国版本图书馆 CIP 数据核字（2019）第 077024 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：时 静 责任编辑：时 静

责任校对：张艳霞 责任印制：张 博

北京铭成印刷有限公司印刷

2019 年 5 月第 3 版·第 1 次印刷

184mm×260mm·15 印张·351 千字

0001—4000 册

标准书号：ISBN 978-7-111-62565-0

定价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

服务咨询热线：（010）88379833

读者购书热线：（010）68326294

网络服务

机工官网：www.cmpbook.com

机工官博：weibo.com/cmp1952

教育服务网：www.cmpedu.com

金书网：www.golden-book.com

封面防伪标均为盗版

前 言

本书是普通高等教育“十一五”国家级规划教材。S7-200 已于 2017 年退市，S7-200 SMART 是 S7-200 的升级换代产品，二者的指令、程序结构和监控方法相同。熟悉 S7-200 的用户几乎不需要任何培训，就可以使用 S7-200 SMART。此外大量的 S7-200 将会长期运行，因此学习 S7-200 还是很有意义的。

由于 S7-200 SMART 的出现，S7-200 的某些功能和模块已经很少使用了。例如 S7-200 SMART 集成了 60 个 I/O 点的 CPU 的通信功能比 S7-200 的以太网模块强大得多，但是价格比后者便宜。为了减轻读者的负担，这次修订删除或精简了某些 S7-200 用得很少的功能有关的内容，减少了十多页的篇幅。为了帮助用户了解 S7-200 SMART，本书增加了对它的简要介绍。S7-200 SMART 详细的情况见参考文献中作者主编的有关教材。

本书的第 1 章介绍了 S7-200 的硬件、PLC 的工作原理和系统的可靠性措施。第 2 章通过实例详细介绍了编程软件的使用方法。第 3 章介绍了 S7-200 的编程语言和程序结构、位逻辑指令、定时器指令和计数器指令。第 4 章用大量的实例全面介绍了功能指令的使用方法。第 5 章通过编程实例，深入浅出地介绍了设计数字量控制系统梯形图的顺序控制设计法。这种设计简单易用，可以节约大量的设计时间。第 6 章介绍了 S7-200 的通信功能，以及使用各种通信指令、通信协议和通信网络，实现 PLC 之间、PLC 和变频器之间的通信的编程、组态和实验的方法。第 7 章介绍了 PID 闭环控制的编程和参数整定的规则；介绍了用作者编写的模拟被控对象的子程序和例程做 PID 闭环实验，手动和自动整定 PID 参数的方法。附录中有 30 多个实验的指导书。

S7-200 的编程软件为 PLC 的高级应用提供了大量的编程向导，只需输入一些参数，就可以自动生成用户程序。本书详细地介绍了常用的编程向导的使用方法。

本书各章均配有习题，读者可通过扫描本书封底的二维码，获取下载链接，下载本书配套的编程软件、指令库、触摸屏组态软件、PLC 串口通信调试软件、S7-200 和相关产品的用户手册和产品样本、30 多个视频教程和 40 多个例程。本书有配套的授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ: 1239258369，电话 010-88379739）。

本书由廖常初主编，廖亮、文家学、孙明渝参加了编写工作。

因作者水平有限，书中难免有错漏之处，恳请读者批评指正。

重庆大学电气工程学院 廖常初

目 录

前言

第 1 章 PLC 的硬件与工作原理	1
1.1 概述	1
1.2 S7-200 系列 PLC	3
1.2.1 S7-200 的特点	3
1.2.2 CPU 模块	4
1.2.3 数字量输入与数字量输出	6
1.2.4 扩展模块	8
1.3 I/O 地址分配与外部接线	10
1.4 S7-200 SMART 简介	12
1.5 逻辑运算与 PLC 的工作原理	14
1.5.1 用触点和线圈实现逻辑运算	14
1.5.2 PLC 的工作原理	15
1.6 PLC 控制系统的硬件可靠性措施	18
1.7 习题	21
第 2 章 STEP 7-Micro/WIN 编程软件使用指南	23
2.1 编程软件概述	23
2.1.1 编程软件的安装与项目的组成	23
2.1.2 帮助功能的使用与 S7-200 的出错处理	25
2.2 程序的编写与传送	26
2.2.1 生成用户程序	26
2.2.2 下载与调试用户程序	30
2.3 符号表与符号地址的使用	33
2.4 用编程软件监控与调试程序	36
2.4.1 用程序状态监控与调试程序	36
2.4.2 用状态表监控与调试程序	39
2.4.3 写入与强制数据	40
2.4.4 调试用户程序的其他方法	42
2.5 使用系统块设置 PLC 的参数	43
2.5.1 断电数据保持的设置与编程	43
2.5.2 创建与使用密码	45
2.5.3 组态输入输出参数	47
2.6 习题	48

第 3 章 S7-200 编程基础	50
3.1 PLC 的编程语言与程序结构.....	50
3.1.1 PLC 编程语言的国际标准.....	50
3.1.2 S7-200 的程序结构.....	52
3.2 数据类型与寻址方式.....	52
3.2.1 数制.....	52
3.2.2 数据类型.....	54
3.2.3 CPU 的存储区.....	56
3.2.4 直接寻址与间接寻址.....	58
3.3 位逻辑指令.....	60
3.3.1 触点指令与逻辑堆栈指令.....	60
3.3.2 输出类指令与其他指令.....	64
3.4 定时器指令与计数器指令.....	66
3.4.1 定时器指令.....	66
3.4.2 计数器指令.....	69
3.5 习题.....	70
第 4 章 S7-200 的功能指令	73
4.1 功能指令概述.....	73
4.1.1 怎样学习功能指令.....	73
4.1.2 S7-200 的指令规约.....	74
4.2 数据处理指令.....	75
4.2.1 比较指令与数据传送指令.....	75
4.2.2 移位指令与循环移位指令.....	78
4.2.3 数据转换指令.....	79
4.2.4 表格指令.....	81
4.2.5 实时时钟指令.....	83
4.3 数学运算指令.....	85
4.3.1 整数运算指令.....	85
4.3.2 浮点数函数运算指令.....	86
4.3.3 逻辑运算指令.....	87
4.4 程序控制指令.....	89
4.4.1 跳转指令.....	89
4.4.2 循环指令.....	91
4.4.3 其他指令.....	92
4.5 局部变量与子程序.....	93
4.5.1 局部变量.....	93
4.5.2 子程序的编写与调用.....	95
4.6 中断程序与中断指令.....	99
4.6.1 中断的基本概念与中断事件.....	99

4.6.2	中断指令	101
4.6.3	中断程序举例	102
4.7	高速计数器与高速脉冲输出	104
4.7.1	高速计数器的工作模式与外部输入信号	105
4.7.2	高速计数器的程序设计	107
4.7.3	高速脉冲输出与开环位置控制	110
4.8	数据块应用与字符串指令	112
4.8.1	数据块概述	112
4.8.2	字符、字符串与数据的转换指令	113
4.8.3	字符串指令	117
4.9	习题	118
第 5 章	数字量控制系统梯形图程序设计方法	120
5.1	梯形图的经验设计法	120
5.2	顺序控制设计法与顺序功能图	125
5.2.1	步与动作	125
5.2.2	有向连线与转换条件	127
5.2.3	顺序功能图的基本结构	128
5.2.4	顺序功能图中转换实现的基本规则	130
5.3	使用置位复位指令的顺序控制梯形图设计方法	131
5.3.1	单序列的编程方法	131
5.3.2	选择序列与并行序列的编程方法	133
5.3.3	应用举例	134
5.4	使用 SCR 指令的顺序控制梯形图设计方法	137
5.4.1	顺序控制继电器指令的应用	137
5.4.2	选择序列与并行序列的编程方法	139
5.4.3	应用举例	140
5.5	具有多种工作方式的系统的顺序控制梯形图设计方法	142
5.5.1	系统的硬件结构与工作方式	142
5.5.2	公用程序与手动程序	144
5.5.3	自动程序	146
5.6	习题	149
第 6 章	PLC 的通信与自动化通信网络	153
6.1	计算机通信概述	153
6.1.1	串行通信	153
6.1.2	IEEE 802 通信标准	154
6.1.3	现场总线及其国际标准	155
6.2	西门子的工业自动化通信网络	156
6.3	S7-200 的通信功能与串行通信网络	158
6.3.1	S7-200 的网络通信协议	158

6.3.2	S7-200 的通信功能	159
6.3.3	S7-200 的串行通信网络	160
6.4	使用网络读写指令的通信	161
6.5	自由端口模式通信	164
6.5.1	自由端口模式通信的参数设置	164
6.5.2	使用接收完成中断的通信程序设计	167
6.5.3	自由端口模式通信实验	169
6.6	Modbus 协议通信	170
6.6.1	Modbus RTU 通信协议	170
6.6.2	Modbus RTU 从站协议通信的编程	171
6.6.3	Modbus RTU 主站协议通信的编程与调试	173
6.7	S7-200 与变频器的 USS 协议通信	176
6.7.1	硬件接线与 V20 变频器参数设置	176
6.7.2	USS 通信的编程	179
6.7.3	S7-200 与 V20 变频器 USS 通信的实验	180
6.8	习题	182
第 7 章	PLC 在模拟量闭环控制中的应用	184
7.1	闭环控制与 PID 控制器	184
7.1.1	模拟量闭环控制系统	184
7.1.2	PID 控制器的数字化	187
7.1.3	PID 指令向导的应用	189
7.2	PID 参数的整定方法	191
7.2.1	PID 参数的物理意义	191
7.2.2	PID 参数整定的规则	194
7.2.3	PID 参数整定的实验	195
7.3	PID 参数自整定	200
7.3.1	自整定的基本方法与自整定过程	200
7.3.2	PID 参数自整定实验	201
7.4	习题	203
附录		205
附录 A	实验指导书	205
A.1	编程软件使用练习	205
A.2	符号表应用实验	206
A.3	用编程软件调试程序的实验	207
A.4	位逻辑指令的功能与应用实验	208
A.5	定时器应用实验	209
A.6	计数器应用实验	210
A.7	比较指令与传送指令应用实验	210
A.8	移位指令与循环移位指令应用实验	211

A.9	数据转换指令应用实验	212
A.10	实时时钟指令应用实验	213
A.11	数学运算指令应用实验	213
A.12	逻辑运算指令应用实验	214
A.13	跳转指令应用实验	215
A.14	循环指令与看门狗复位指令实验	216
A.15	子程序的编程实验	216
A.16	中断程序的编程实验	217
A.17	高速计数器与高速输出应用实验	217
A.18	数据块与字符串指令应用实验	218
A.19	定时器与计数器应用的编程实验	219
A.20	自动往返的小车控制系统的编程实验	219
A.21	使用置位复位指令的顺序控制程序的编程实验	220
A.22	液体混合控制与剪板机控制系统的编程实验	220
A.23	使用 SCR 指令的顺序控制程序的编程实验	221
A.24	内胎硫化机与专用钻床控制系统的编程实验	222
A.25	具有多种工作方式的系统的控制程序调试实验	222
A.26	使用网络读写指令的通信实验	223
A.27	使用接收完成中断的通信实验	223
A.28	使用 Modbus RTU 协议的通信实验	224
A.29	PLC 与变频器的 USS 协议通信实验	225
A.30	PID 控制器参数手动整定实验	225
A.31	PID 控制器参数自整定实验	225
附录 B	常用特殊存储器位	226
附录 C	S7-200 指令表索引	227
附录 D	配套资源说明	227
参考文献		229

第 1 章 PLC 的硬件与工作原理

1.1 概述

现代社会要求制造业对市场需求做出迅速的反应，生产出小批量、多品种、多规格、低成本和高质量的产品，为了满足这一要求，生产设备和自动生产线的控制系统必须具有极高的可靠性和灵活性，可程序控制器（Programmable Logic Controller, PLC）正是顺应这一要求出现的，它是以微处理器为基础的通用工业控制装置。

PLC 的应用面广、功能强大、使用方便，已经广泛地应用在各种机械设备和生产过程的自动控制系统中。PLC 在其他领域（例如民用和家庭自动化的应用）也得到了迅速的发展。PLC 仍然处于不断的发展之中，其功能不断增强，更为开放，它不但是单机自动化中应用最广的控制设备，在大型工业网络控制系统中也占有不可动摇的地位。PLC 应用面之广、普及程度之高，是其他计算机控制设备不可比拟的。

本书以西门子公司的 S7-200 系列小型 PLC 为主要讲授对象。S7-200 具有极高的可靠性、丰富的指令集和内置的集成功能、强大的通信能力和品种丰富的扩展模块。它可以单机运行，用于代替继电器控制系统，也可以用于复杂的自动化控制系统。由于它有极强的通信功能，在网络控制系统中也能充分发挥其作用。S7-200 以其极高的性能价格比，在国内占有很大的市场份额。

S7-200 主要由 CPU 模块、输入模块和输出模块组成（见图 1-1）。

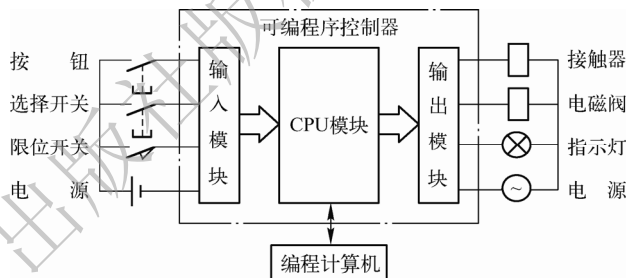


图 1-1 PLC 控制系统示意图

1.1 CPU 模块

CPU 模块简称为 CPU，主要由微处理器、电源和集成的输入电路、输出电路组成。在 PLC 控制系统中，微处理器相当于人的大脑和心脏，它不断地采集输入信号，执行用户程序，刷新系统的输出；存储器用来储存程序和数据。

PLC 的程序分为操作系统和用户程序。操作系统使 PLC 具有基本的智能，能够完成 PLC 设计者规定的各种工作。操作系统由 PLC 生产厂家设计并固化在 ROM（只读存储器）

中，用户不能直接读取。用户程序由用户设计，它使 PLC 能完成用户要求的特定功能。用户程序存储器的容量以字节（Byte，B）为单位。

PLC 使用以下几种物理存储器：

（1）随机存取存储器（RAM）

用户程序和编程软件可以读出 RAM 中的内容，也可以改写 RAM 中的数据。RAM 是易失性的存储器，RAM 芯片电源中断后，储存的信息将会丢失。

RAM 的工作速度快、价格便宜、改写方便。在关断 PLC 的外部电源后，可以用锂电池保存 RAM 中的用户程序和某些数据。锂电池可以用 1~3 年，需要更换锂电池时，由 PLC 发出信号，通知用户。S7-200 不使用锂电池。

（2）只读存储器（ROM）

ROM 的内容只能读出，不能写入。它是非易失性的，它的电源消失后，仍能保存储存的内容。ROM 用来存放 PLC 的操作系统。

（3）可以电擦除可编程的只读存储器（EEPROM）

EEPROM 是非易失性的存储器，掉电后它保存的数据不会丢失。PLC 可以读写它，它兼有 ROM 的非易失性和 RAM 的随机存取的优点，但是写入数据所需的时间比 RAM 长得多，改写的次数有限制。S7-200 用 EEPROM 来存储用户程序和需要长期保存的重要数据。

2. I/O 模块

输入（Input）模块和输出（Output）模块简称为 I/O 模块，I/O 模块和 CPU 的输入电路、输出电路是系统的“眼、耳、手、脚”，是联系外部现场设备和 CPU 模块的桥梁。

输入模块和 CPU 的输入电路用来接收和采集输入信号，开关量输入用来接收从按钮、选择开关、数字拨码开关、限位开关、接近开关、光电开关、压力继电器等提供的开关量输入信号；模拟量输入用来接收电位器、测速发电机和各种变送器提供的连续变化的模拟量信号。开关量输出用来控制接触器、电磁阀、电磁铁、指示灯、数字显示装置和报警装置等输出设备，模拟量输出用来控制电动调节阀、变频器等执行器。

CPU 模块的工作电压一般是 DC 5V，而 PLC 外部的输入/输出电路的电源电压较高，例如 DC 24V 和 AC 220V。从外部引入的尖峰电压和干扰噪声可能损坏 CPU 模块中的元器件，或使 PLC 不能正常工作。在输入/输出电路中，用光耦合器、光敏晶闸管、小型继电器等器件来隔离 PLC 的内部电路和外部电路，输入/输出电路除了传递信号外，还有电平转换与隔离的作用。

3. 编程软件

使用 S7-200 的编程软件 STEP 7-Micro/WIN，可以在计算机屏幕上直接生成和编辑梯形图或指令表程序，程序被编译后下载到 PLC。可以将 PLC 中的程序上传到计算机，还可以用编程软件监控 PLC。一般用 USB/PPI 编程电缆来实现编程计算机与 PLC 的通信。

本书的配套资源提供了包含升级包 SP9 的编程软件 STEP 7-Micro/WIN V4.0。

4. 电源

PLC 使用 AC 220V 电源或 DC 24V 电源。CPU 可以为输入电路和外部的电子传感器（例如接近开关）提供 DC 24V 电源，驱动 PLC 负载的直流电源一般由用户提供。

1.2 S7-200 系列 PLC

西门子公司具有品种非常丰富的 PLC 产品。S7-200、S7-1200 和 S7-200 SMART 是小型 PLC。S7-300、S7-400 和 S7-1500 是模块式大中型 PLC。WinAC 是在 PC（个人计算机）上实现 PLC 功能的“软 PLC”。

1.2.1 S7-200 的特点

1. 功能强

1) S7-200 有 6 种 CPU 模块，最多可以扩展 7 个扩展模块，扩展到 256 点数字量 I/O 或 45 路模拟量 I/O，最多有 24KB 程序存储空间和 10KB 用户数据存储空间。

2) 集成了 6 个有 13 种工作模式的高速计数器，以及两点高速脉冲发生器/脉冲宽度调制器。CPU 224XP 的高速计数器的最高计数频率为 200kHz，高速输出的最高频率为 100kHz。

3) 直接读、写模拟量 I/O 模块，不需要复杂的编程。CPU 224XP 集成有 2 路模拟量输入，1 路模拟量输出。

4) 使用 PID 调节控制面板，可以实现 PID 参数自整定。

5) S7-200 的 CPU 模块集成了很强的位置控制功能，此外还有位置控制模块 EM 253。使用位置控制向导可以方便地实现位置控制的编程。

6) 有配方和数据记录功能，以及相应的编程向导，配方数据和数据记录用存储卡保存。

7) 普通 PLC 的温度适用范围为 0~55℃，宽温型 S7-200 SIPLUS 的温度适用范围为 -25~+70℃。

2. 先进的程序结构

S7-200 的程序结构简单清晰，在编程软件中，主程序、子程序和中断程序分页存放。使用各程序块中的局部变量，易于将程序块移植到别的项目。子程序用输入、输出参数作软件接口，便于实现结构化编程。S7-200 的指令功能强，易于掌握。

3. 灵活方便的存储器结构

S7-200 的输入 (I)、输出 (Q)、位存储器 (M)、顺序控制继电器 (S)、变量存储器 (V) 和局部变量 (L) 均可以按位 (bit)、字节 (B)、字 (W) 和双字 (DW) 读写。

4. 功能强大、使用方便的编程软件

编程软件 STEP 7-Micro/WIN 可以使用包括中文在内的多种语言。有梯形图、语句表和功能块图编程语言，以及 SIMATIC、IEC 61131-3 两种编程模式。

STEP 7-Micro/WIN 的监控功能形象直观、使用方便。可以用 3 种编程语言监控程序的执行情况，用状态表监视、修改和强制变量，用趋势图监视变量的波形。用系统块设置参数方便直观。STEP 7-Micro/WIN 具有强大的中文帮助功能，在线帮助、右键快捷菜单、指令和子程序的拖放功能使编程软件的使用非常方便。

S7-200 有 4 种加密级别，此外还可以对单独的程序块和项目文件加密。

STEP 7-Micro/WIN 提供包含时间标记和事件标志的事件记录，以“后进先出”的原则存储在缓冲区中。

5. 简化复杂编程任务的向导功能

PID 控制、网络通信、高速输入、高速输出、位置控制、数据记录、配方和文本显示器等编程和应用是 PLC 程序设计中的难点，用普通的方法对它们编程既烦琐又容易出错。STEP 7-Micro/WIN 为此提供了大量的编程向导，只需要在向导的对话框中输入一些参数，就可以自动生成包括中断程序在内的用户程序。

6. 强大的通信功能

S7-200 的 CPU 模块有 1 个或 2 个标准的 RS-485 端口，可用于编程或通信，不需增加硬件就可以与别的 S7 PLC、变频器和计算机通信。S7-200 可以使用 PPI、MPI、Modbus RTU 和 USS 等通信协议，以及自由端口通信模式。

通过不同的通信模块，S7-200 可以连接到以太网、互联网和现场总线 PROFIBUS-DP、AS-i。通过 Modem 模块 EM 241，可以用模拟电话线实现与远程设备的通信。STEP 7-Micro/WIN 提供多种与通信有关的向导。

PC Access V1.0 是专门为 S7-200 设计的 OPC 服务器，支持所有的 S7-200 数据形式和所有的 S7-200 协议，支持多 PLC 连接和标准的 OPC 客户机，并具有内置的客户机测试功能。

7. S7-200 配套的人机界面

S7-200 主要与精彩系列触摸屏 Smart 700 IE 和 Smart 1000 IE 配合使用。

8. 完善的网上技术支持

在西门子公司的网站可以下载 S7-200 的软件和手册，S7-200 有专门的子网站。可以在技术论坛与网友切磋技艺、交流经验，在网上向西门子的工程师提交问题或咨询硬件方案。在“找答案”网页可以提出问题，或回答别人的问题。

1.2.2 CPU 模块

S7-200 有 6 种 CPU 模块（见图 1-2），各 CPU 模块的技术指标见表 1-1。

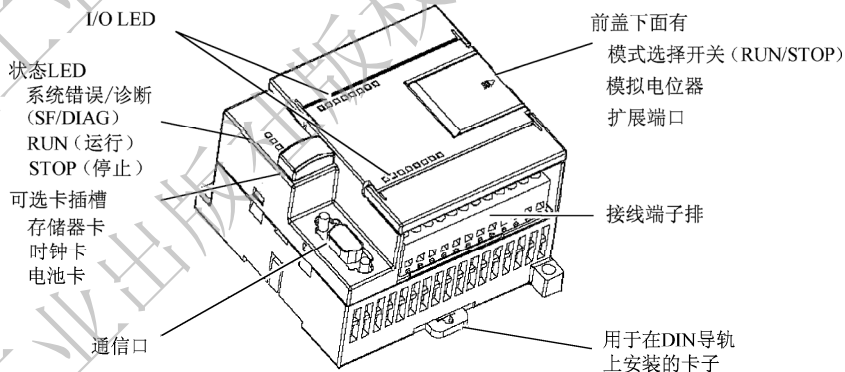


图 1-2 S7-200 CPU 模块的外形图

CPU 的用户存储器使用 EEPROM，后备电池（选件）可使用 200 天，布尔量运算指令执行时间为 $0.22\mu\text{s}/\text{指令}$ ，位存储器（M）、顺序控制继电器各有 256 点，计数器和定时器各有 256 个；有两点定时中断，最大时间间隔为 255ms；有 4 点外部硬件输入中断。

CPU 221 无扩展功能，适合作小点数的微型控制器。CPU 222 有扩展功能，CPU 224 是具有较强控制功能的控制器。

表 1-1 S7-200 CN CPU 技术规范

特 性	CPU 221	CPU 222	CPU 224	CPU 224XP/CPU 224XPsi	CPU 226
本机数字量 I/O	6DI/4DO	8DI/6DO	14DI/10DO	14DI/10DO	24DI/16DO
本机模拟量 I/O	—	—	—	2AI/1AO	—
扩展模块数量	—	2	7	7	7
最大数字量点数	6DI/4DO	48DI/46DO	114DI/110DO	114DI/110DO	128DI/128DO
AI/AO/最大模拟量点数	—	16/8/16	32/28/44	32/29/45, 集成 2AI/1AO	32/28/44
掉电保持时间(电容)/h	50	50	100	100	100
用户程序存储器/KB	4	4	12	16	24
用户数据存储器/KB	2	2	8	10	10
单相高速计数器	4 路 30kHz		6 路 30kHz	4 路 30kHz, 2 路 200kHz	6 路 30kHz
A/B 相高速计数器	其中 2 路 20kHz		其中 4 路 20kHz	其中 3 路 20kHz, 1 路 100kHz	其中 4 路 20kHz
高速脉冲输出	2 路 20kHz		2 路 20kHz	2 路 100kHz	2 路 20kHz
模拟量调节电位器	一个, 8 位分辨率		2 个, 8 位分辨率		
RS-485 通信口/个	1	1	1	2	2
实时时钟	有(时钟卡)	有(时钟卡)	有	有	有
可选卡件	存储卡、电池卡和实时时钟卡		存储卡和电池卡		
脉冲捕捉输入/个	6	8	14	—	24
外形尺寸/mm	90×80×62	90×80×62	120.5×80×62	140×80×62	196×80×62
DC 24V 传感器电流/mA	180	180	280	280	400

CPU 224XP 集成有 2 路模拟量输入(10 位 DC±10V), 1 路模拟量输出(10 位, DC 0~10V 或 0~20mA), 有两个 RS-485 通信端口, 高速脉冲输出频率提高到 100kHz, 高速计数器频率提高到 200kHz。

CPU 226 适合于复杂的中小型控制系统, 可扩展到 256 点数字量和 44 路模拟量, 有两个 RS-485 通信端口。

S7-200 采用主程序、最多 8 级子程序和中断程序的程序结构。监控定时器(看门狗)的定时时间为 500ms, 最多可使用 8 个 PID 控制器。

数字量输入中有 4 点用作硬件中断, 6 点用于高速计数功能。除了 CPU 224XP 外, 32 位高速加/减计数器的最高计数频率为 30kHz, 可以对增量式编码器的两个互差 90° 的脉冲列计数, 计数值等于设定值或计数方向改变时产生中断, 在中断程序中可以及时地对输出进行操作。DC 输出型的两个高速输出可以输出频率和宽度可调的脉冲列。

RS-485 串行通信端口的外部信号与逻辑电路之间没有隔离, 支持 PPI、自由端口模式和点对点 PPI 主站模式, 可以作 MPI 从站。

通信端口可以用于与运行 STEP 7-Micro/WIN 的计算机通信, 与文本显示器和操作员界面的通信, 以及 S7-200 CPU 之间的通信; 通过自由端口模式、Modbus RTU 和 USS 协议, 可以与其他设备进行串行通信。通过 AS-i 通信接口模块, 可以接入 496 个远程数字量输入/输出点。

可选的存储卡可以永久保存用户程序、数据记录、配方和文档记录, 或用来传输程序。可选的电池卡保存数据的时间的典型值为 200 天。用于断电保存数据的超级电容器充电 20min, 可以充 60% 的电量。

1.2.3 数字量输入与数字量输出

各数字量 I/O 点的通/断状态用发光二极管 (LED) 显示, PLC 与外部接线的连接采用接线端子。大多数 CPU 和扩展模块有可拆卸的端子排, 不需断开端子排上的外部连线, 就可以迅速地更换模块。

1. 数字量输入电路

图 1-3 是 S7-200 CPU 的数字量输入点的内部电路和外部接线图, 图中只画出了一路输入电路, 输入电流为数毫安 (见表 1-2)。1M 是同一组输入点各内部输入电路的公共点。S7-200 可以用 CPU 模块内部的 DC 24V 传感器作输入回路的电源 (见图 1-9), 它还可以为接近开关、光电开关之类的传感器提供电流。

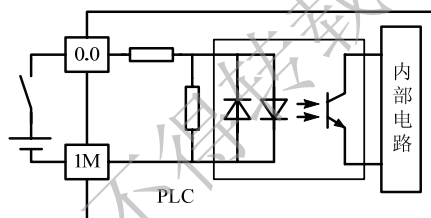


图 1-3 输入电路

当图 1-3 中的外部触点接通时, 光耦合器中两个反并联的发光二极管中的一个亮, 光敏晶体管饱和导通; 外部触点断开时, 光耦合器中的发光二极管熄灭, 光敏晶体管截止, 信号经内部电路传送给 CPU 模块。显然, 可以改变图 1-3 中输入回路的电源极性。

表 1-2 S7-200 CPU 的数字量输入点技术指标

项 目	DC 24V 输入 (CPU 224XP 和 CPU 224XPSi)	DC 24V 输入 (其他 CPU)
输入类型	漏型/源型 (IEC 类型 1, I0.3~I0.5 除外)	漏型/源型 (IEC 类型 1)
输入电压额定值	DC 24V, 典型值 4mA	
输入电压浪涌值	35V/0.5s	
逻辑 1 信号 (最小)	I0.3~I0.5 为 DC 4V, 8mA; 其余为 DC 15V, 2.5mA	DC 15V, 2.5mA
逻辑 0 信号 (最大)	I0.3~I0.5 为 DC 1V, 1mA; 其余为 DC 5V, 1mA	DC 5V, 1mA
输入延迟	0.2~12.8ms 可选	
连接 2 线式接近开关的允许漏电流	最大 1mA	
光电隔离	AC 500V, 1min	
高速计数器输入逻辑 1 电平	DC 15~30V; 单相 20kHz, 两相 10kHz; DC 15~26V; 单相 30kHz, 两相 20kHz	
CPU 224XP 的 HSC4 和 HSC5 的输入	逻辑 1 电平 > DC 4V 时, 单相 200kHz, 两相 100kHz	
电缆长度	非屏蔽电缆 300m, 屏蔽电缆 500m, 高速计数器 50m	

S7-200 的数字量输入滤波器用来过滤输入接线上可能对输入状态造成不良影响的噪声。可以用 STEP 7-Micro/WIN 的系统块来设置输入滤波器的延迟时间。

S7-200 有 AC 120V/230V 数字量输入模块。交流输入方式适合在有油雾、粉尘的恶劣环境下使用。直流输入模块可以直接与接近开关、光电开关等电子输入装置连接。

2. 数字量输出电路

S7-200 的 CPU 模块的数字量输出电路的功率器件有驱动直流负载的场效应晶体管和小型继电器, 后者既可以驱动交流负载, 也可以驱动直流负载, 负载电源由外部提供。

输出电流的额定值与负载的性质有关, 例如 S7-200 的继电器输出电路可以驱动 2A 的电阻性负载, 但是只能驱动 200W 的白炽灯。输出电路一般分为若干组, 对每一组的总电流也有限制。图 1-4 是继电器输出电路, 继电器同时起隔离和功率放大作用, 每一路只给用户提供一对常开触点。

图 1-5 是使用场效应晶体管 (MOSFET) 的输出电路, 其接通和断开的最大延时时间见

表 1-3。输出信号送给内部电路中的输出锁存器，再经光耦合器送给场效应晶体管，后者的饱和导通状态和截止状态相当于触点的接通和断开。图中的稳压管用来抑制关断过电压和外部的浪涌电压，以保护场效应晶体管。

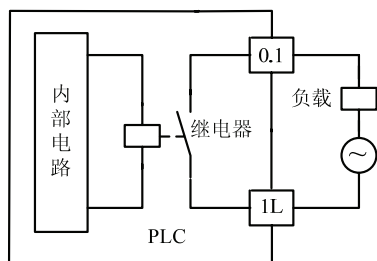


图 1-4 继电器输出电路

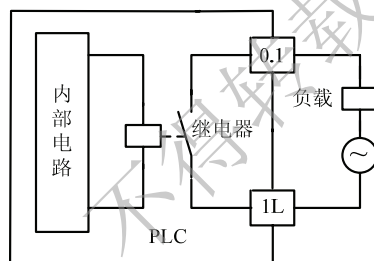


图 1-5 场效应晶体管输出电路

S7-200 的数字量扩展模块中还有一种用双向晶闸管作为输出元件的 AC 230V 的输出模块。每点的额定输出电流为 0.5A，灯负载为 60W。最大漏电流 1.8mA，由接通到断开的最大时间为 0.2ms 与工频半周期之和。

表 1-3 S7-200 CPU 的数字量输出点技术指标

输出类型	DC 24V 输出 (CPU 221、CPU 222、CPU 224 和 CPU 226)	DC 24V 输出 (CPU 224XP/CPU 224XPsi)	继电器型输出
输出电压额定值	DC 24V	DC 24V	DC 24V 或 AC 250V
输出电压范围	DC 20.4~28.8V	DC 5~28.8V (Q0.0~Q0.4) DC 20.4~28.8V (Q0.5~Q1.1)	DC 5~30V, AC 5~250V
浪涌电流	最大 8A, 100ms		5A, 4s, 占空比 0.1
逻辑 1 最小输出电压	DC 20V, 最大电流时	见表后正文中的说明	—
逻辑 0 最大输出电压	DC 0.1V, 10kΩ负载		—
逻辑 1 最大输出电流	0.75A (电阻负载)	0.75A (电阻负载)	2A (电阻负载)
逻辑 0 最大漏电流	10μA	10μA	—
灯负载	5W	5W	DC 30W/AC 200W
接通状态电阻	0.3Ω, 最大 0.6Ω	0.3Ω, 最大 0.6Ω	新的时候最大 0.2Ω
每个公共端的额定电流	6A	3.75A/7.5A	10A
感性箝位电压	L+减 DC 48V, 1W 功耗	L+减 DC 48V, 1W 功耗/ 1M+DC 48V, 1W 功耗	—
从关断到接通最大延时	Q0.0 和 Q0.1 为 2μs, 其他 15μs	Q0.0 和 Q0.1 为 0.5μs, 其他 15μs	—
从接通到关断最大延时	Q0.0 和 Q0.1 为 10μs, 其他 130μs	Q0.0 和 Q0.1 为 1.5μs, 其他 130μs	—
切换最大延时	—	—	10ms
最高脉冲频率	20kHz (Q0.0 和 Q0.1)	100kHz (Q0.0 和 Q0.1)	1Hz

继电器输出电路可用的电压范围广，导通压降小，承受瞬时过电压和过电流的能力较强，但是动作速度较慢，寿命（动作次数）有一定的限制。如果系统输出量的变化不是很频繁，建议优先选用继电器型的输出模块。

场效应晶体管输出电路用于直流负载，它的反应速度快、寿命长，过载能力稍差。

CPU 224XPsi 具有 MOSFET 漏型输出（电流从输出端子流入），可以驱动具有源型输入的设备。S7-200 所有其他场效应晶体管型输出的 CPU 都是 MOSFET 源型输出的（电流从输出端子流出）。

DC 输出的 CPU 224XP 逻辑 1 最小输出电压为 L+减 0.4V（最大电流时），逻辑 0 最大输出电压为 DC 0.1V（10kΩ负载）。

DC 输出的 CPU 224XPsi 逻辑 1 最小输出电压为外部电压减 0.4V（外部接 10kΩ上拉电阻），逻辑 0 最大输出电压为 1M + 0.4V（最大负载时）。

继电器输出的开关延时最大 10ms，无负载时触点的机械寿命 10000000 次，额定负载时触点寿命 100000 次。非屏蔽电缆最大长度 150m，屏蔽电缆 500m。

1.2.4 扩展模块

1. 数字量扩展模块

可以选用 8 点、16 点、32 点和 64 点的数字量输入/输出模块（见表 1-4），来满足不同的控制需要。除了 CPU 221 外，其他 CPU 模块均可以配接多个扩展模块（见表 1-1），连接时 CPU 模块放在最左边，扩展模块用扁平电缆与它左边的模块相连。

表 1-4 数字量扩展模块

型 号	型 号
EM221, 8 数字量输入 DC 24V	EM223, DC 24V 数字量组合 4 输入/4 输出
EM221, 8 数字量输入 AC 120V/230V	EM223, DC 24V 数字量组合 4 输入/4 继电器输出
EM221, 16 数字量输入 DC 24V	EM223, DC 24V 数字量组合 8 输入/8 输出
EM222, 4 数字量输出 DC 24V, 5A	EM223, DC 24V 数字量组合 8 输入/8 继电器输出
EM222, 4 继电器输出, 10A	EM223, DC 24V 数字量组合 16 输入/16 输出
EM222, 8 数字量输出 DC 24V	EM223, DC 24V 数字量组合 16 输入/16 继电器输出
EM222, 8 继电器输出	EM223, DC 24V 数字量组合 32 输入/32 输出
EM222, 8 数字量输出 AC 120V/230V	EM223, DC 24V 数字量组合 32 输入/32 继电器输出

2. PLC 对模拟量的处理

在工业控制中，某些输入量（例如压力、温度、流量、转速等）是模拟量，某些执行机构（例如电动调节阀和变频器等）要求 PLC 输出模拟量信号，而 PLC 的 CPU 只能处理数字量。模拟量首先被传感器和变送器转换为标准量程的电流或电压，例如 4~20mA、1~5V 和 0~10V，模拟量输入模块的 A-D 转换器将它们转换成数字量。带正负号的电流或电压在 A-D 转换后用二进制补码表示。有的模拟量输入模块将温度传感器提供的信号直接转换为温度值。

模拟量输出模块的 D-A 转换器将 PLC 中的数字量转换为模拟量电压或电流，再去控制执行机构。模拟量 I/O 模块的主要任务就是实现 A-D 转换（模拟量输入）和 D-A 转换（模拟量输出）。

A-D 转换器和 D-A 转换器的二进制位数反映了它们的分辨率，位数越多，分辨率越高。模拟量输入/输出模块的另一个重要指标是转换时间。

3. 模拟量输入模块

S7-200 有 9 种模拟量扩展模块（见表 1-5），RTD 是热电阻的简称。可以用模拟量输入模块上的 DIP 开关来设置多种量程。EM 231 模拟量输入模块的输入信号有 5 档量程（DC 0~10V、0~5V、0~20mA、±2.5V 和±5V）。EM 235 模块有 16 档量程。

表 1-5 模拟量扩展模块

型 号
EM231 模拟量输入, 4 输入
EM231 模拟量输入, 8 输入
EM232 模拟量输出, 2 输出
EM232 模拟量输出, 4 输出
EM235 模拟量组合, 4 输入/1 输出
EM231 模拟量输入热电偶, 4 输入
EM231 模拟量输入热电偶, 8 输入
EM231 模拟量输入 RTD, 2 输入
EM231 模拟量输入 RTD, 4 输入

模拟量输入模块的分辨率为 12 位，单极性全量程输入范围对应的数字量输出为 0~32000。双极性全量程输入范围对应的数字量输出为-32000~+32000。电压输入时输入阻抗大于等于 2MΩ，电流输入时输入阻抗为 250Ω。A-D 转换时间小于 250μs，模拟量输入的阶跃响应时间为 1.5ms（达到稳态值的 95%时）。

图 1-6 中的 MSB 和 LSB 分别是最高有效位和最低有效位。最高有效位是符号位，0 表示正值，1 表示负值。模拟量转换为数字量得到的 12 位数被自动地尽可能地往高位移动，称为左对齐。移位后单极性格式的最低位是 3 个连续的 0，相当于 A-D 转换值被乘以 8。双极性格式的最低位是 4 个连续的 0，相当于 A-D 转换值被乘以 16。

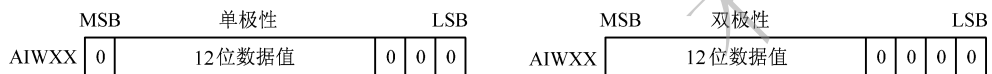


图 1-6 模拟量输入数据字的格式

4. 将模拟量输入模块的输出值转换为实际的物理量

转换时应考虑变送器的输入/输出量程和模拟量输入模块的量程，找出被测物理量与 A-D 转换后的数字值之间的比例关系。

【例 1-1】 某发电机的电压互感器的电压比为 10kV/100V（线电压），电流互感器的电流比为 1000A/5A，功率变送器的额定输入电压和额定输入电流分别为 AC 100V 和 5A，额定输出电压为 DC±5V，模拟量输入模块将 DC±5V 的输入信号转换为数字-32000~+32000。设转换后得到的数字为 N ，试求以 kW 为单位的有功功率值。

解：在设计功率变送器时已考虑了功率因数对功率计算的影响，因此在推导转换公式时，可以按功率因数为 1 来处理。根据互感器额定值计算的一次回路有功功率额定值为

$$\sqrt{3} \times 10000 \times 1000 \text{ W} = 17321000 \text{ W} = 17321 \text{ kW}$$

由以上关系不难推算出互感器一次回路的有功功率与转换后的数字值之间的关系为 17321/32000 kW/字。设转换后的数字为 N ，如果以 kW 为单位显示功率 P ，采用定点数运算时的计算公式为

$$P = N \times 17321 / 32000 \quad (\text{kW})$$

【例 1-2】 量程为 0~10MPa 的压力变送器的输出信号为 DC 4~20mA，模拟量输入模块将 0~20mA 转换为 0~32000 的数字量，设转换后得到的数字为 N ，试求以 kPa 为单位的压力值。

解：4~20mA 的模拟量对应于数字量 6400~32000，即 0~10000kPa 对应于数字量 6400~32000，压力（kPa）的计算公式为

$$P = \frac{(10000 - 0)}{(32000 - 6400)} (N - 6400) = \frac{100}{256} (N - 6400)$$

5. 模拟量输出模块

模拟量输出模块 EM 232 的量程有±10V 和 0~20mA 两种，对应的数字量分别为-32000~+32000 和 0~32000（见图 1-7）。满量程时电压输出和电流输出的分辨率分别为 12 位和 11 位。25℃ 时的精度典型值为±0.5%，电压输出和电流输出的稳定时间分别为 100μs 和 2ms。最大驱动能力如下：电压输出时负载阻抗最小 5kΩ；电流输出时负载阻抗最大 500Ω。

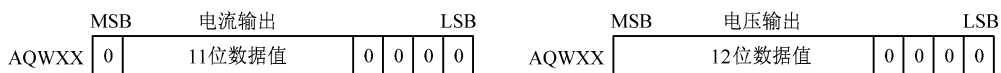


图 1-7 模拟量输出数据字的格式

模拟量输出数据字是左对齐的，最高有效位是符号位，0 表示正值。最低位是 4 个连续的 0，在将数据字装载到 DAC 寄存器之前，低位的 4 个 0 被截断，不会影响输出信号值。

6. 热电偶、热电阻扩展模块

热电偶模块 EM 231 可以与 S、T、R、E、N、K、J 型热电偶配套使用，用模块上的 DIP 开关来选择热电偶的类型。热电偶输出的电压范围为 DC±80mV，模块输出的数字量为 ±27648。

热电阻 (RTD) 的接线方式有 2 线、3 线和 4 线三种，4 线方式的精度最高，因为受接线误差的影响，2 线方式的精度最低。热电阻模块 EM 231 可以通过 DIP 开关来选择热电阻的类型、接线方式、测量单位和开路故障的方向。连接到同一个扩展模块上的热电阻必须是相同类型的。改变 DIP 开关后必须将 PLC 断电后再通电，新的设置才能起作用。

EM 231 热电偶模块和热电阻模块具有冷端补偿电路，如果环境温度迅速变化，则会产生额外的误差，建议将热电偶和热电阻模块安装在环境温度稳定的地方。

2 路热电阻模块和 4 路热电偶模块的采样周期为 405ms (Pt10000 为 700ms)，4 路热电阻和 8 路热电偶模块的采样周期是上述模块的两倍。基本误差和重复性分别为满量程的 0.1%和 0.05%。

7. 称重模块与位置控制模块

称重模块 SIWAREX MS 可以用作电子秤、料斗秤、台秤、吊车秤，或监测输送带张力、测量工业货梯或轧制生产线的负荷。

位置控制模块 EM 253 用于用步进电动机作执行机构的单轴开环位置控制，自带 5 个数字量输入点和 4 个数字量输出点。STEP 7-Micro/WIN 为位置控制模块的组态和编程提供了位置控制向导和 EM 253 控制面板。

1.3 I/O 地址分配与外部接线

1. I/O 地址分配

S7-200 CPU 有一定数量的本机 I/O，本机 I/O 有固定的地址。可以用扩展 I/O 模块来增加 I/O 点数，扩展模块安装在 CPU 模块的右边。I/O 模块分为数字量输入、数字量输出、模拟量输入和模拟量输出 4 类。CPU 分配给数字量 I/O 模块的地址以字节为单位，一个字节由 8 个数字量 I/O 点组成。扩展模块 I/O 点的字节地址由 I/O 的类型和模块在同类 I/O 模块链中的位置来决定。以图 1-8 中的数字量输出为例，分配给 CPU 模块的字节地址为 QB0 和 QB1，分配给 0 号扩展模块的字节地址为 QB2，分配给 3 号扩展模块的字节地址为 QB3。

如果某个模块的数字量 I/O 点数不是 8 的整数倍，最后一个字节中未用的位 (例如图 1-8 中的 I1.6 和 I1.7) 不会分配给 I/O 链中的后续模块。可以像存储器 (M) 那样来使用输出模块最后一个字节中未用的位。输入模块在每次更新输入时都将输入字节中未用的位清零，因此不能用它们来存储数据。模拟量扩展模块以 2 点 (4B) 递增的方式来分配地址，所以

图 1-8 中 2 号扩展模块的模拟量输出的地址应为 AQW4，而不是未使用的 AQW2。

CPU 224XP		模块0	模块1	模块2	模块3	模块4
10.0	Q0.0	4输入 4输出	8输入	4AI 1AO	8输出	4AI 1AO
10.1	Q0.1	12.0 Q2.0	I3.0	AIW4 AQW4	Q3.0	AIW12 AQW8
⋮	⋮	12.1 Q2.1	I3.1	AIW6	Q3.1	AIW14
11.5	Q1.1	12.2 Q2.2	⋮	AIW8	⋮	AIW16
AIW0 AQW0		12.3 Q2.3	I3.7	AIW10	Q3.7	AIW18
AIW2						

图 1-8 CPU 224XP 的 I/O 地址分配举例

2. PLC 的外部接线

S7-200 采用横截面积为 $0.5\sim 1.5\text{mm}^2$ 的导线，使用交流电源的 AC/DC/继电器型的 CPU 222 的外部接线如图 1-9 所示。应提供一个单独的开关，能同时切断 S7-200 CPU、输入电路和输出电路的供电。可以用熔断器或断路器等过流保护装置来限制供电线路中的电流，也可以为输出点分组或分点设置熔断器。S7-200 的交流电源线和 I/O 点之间的隔离电压为 AC 1500V，可以作为交流电源线和低压电路之间的安全隔离。

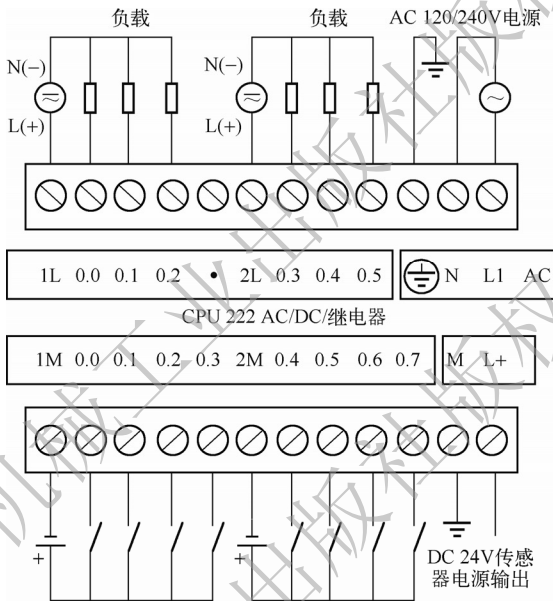


图 1-9 CPU 222 AC/DC/继电器外部接线图

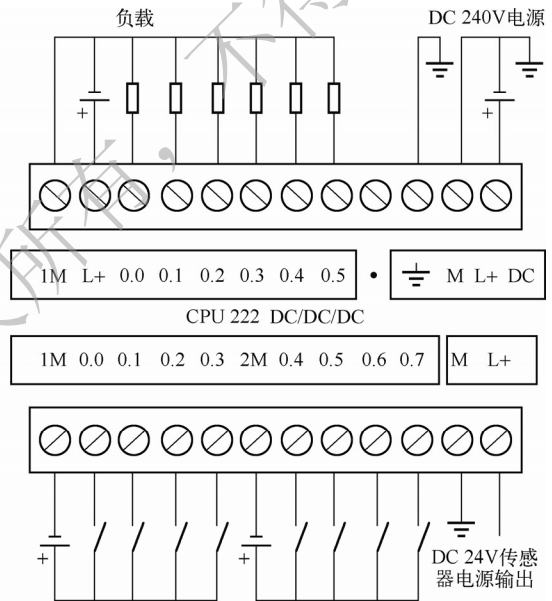


图 1-10 CPU 222 DC/DC/DC 外部接线图

CPU 222 的 8 个输入点 I0.0~I0.7 被分为两组，1M 和 2M 分别是两组输入点内部电路的公共端。6 个输出点 Q0.0~Q0.5 分为两组，1L 和 2L 分别是两组输出点内部电路的公共端。可以用 CPU 提供的 DC 24V 传感器电源作输入回路的电源。

PLC 的交流电源接在 L1 (相线) 和 N (零线) 端子上，此外还有被标示为 \oplus 的保护接地端子。

使用直流电源的 DC/DC/DC 型的 CPU 222 的外部接线如图 1-10 所示，它的电源、输入回路和输出回路的电源电压均为 DC 24V。

3. DC 24V 传感器电源

L+和 M 端子分别是 CPU 模块提供的 DC 24V 传感器电源的正极和负极，该电源可以为输入点、扩展模块上的继电器线圈或者其他设备供电。如果设备用电量超过了传感器电源的额定值，应为系统另外配备一个外部 DC 24V 供电电源。

如果使用了外部的 DC 24V 电源，应确保该电源没有与 S7-200 CPU 上的传感器电源并联使用，因为这种并联可能导致两个电源冲突，甚至损坏电源，并使 PLC 产生不确定的操作。为了加强电子噪声保护，建议将不同电源的公共端（M 端）连在一起。

1.4 S7-200 SMART 简介

2012 年 7 月，西门子推出了为中国客户量身定制的高性价比的小型 PLC S7-200 SMART，2017 年 7 月发布了 V2.3 版。它是 S7-200 的升级换代产品，与精彩系列面板 SMART LINE IE V3、V20 变频器和 V90 伺服系统完美整合，无缝集成，为 OEM（原始设备制造商）客户提供了高性价比的小型自动化解决方案。

S7-200 SMART 继承了 S7-200 的诸多优点，与 S7-200 的指令、程序结构、数据类型、存储区和监控方法相同。熟悉 S7-200 的用户几乎不需要任何培训就可以使用 S7-200 SMART。它具有比 S7-200 更强的通信功能，且价格便宜，具有很高的性价比。编程软件 STEP 7-Micro/WIN SMART 的界面友好，更为人性化。

S7-200 的 CPU、信号模块和配套的 TD 400C 文本显示器已于 2017 年 10 月 1 日起停产，停产后上述产品仅由备件服务部门以备件形式提供。S7-200 在我国企业中具有巨大的保有量，它们还有很长的使用寿命。熟悉了 S7-200，就很容易使用和掌握 S7-200 SMART。因此学习 S7-200 还是很有意义的。

1. S7-200 SMART 的 CPU

S7-200 SMART 的 CPU 分为标准型和继电器输出的紧凑型，后者的型号为 CPU CR_{xs}。标准型分为继电器输出型和 MOSFET（场效应晶体管）输出型。它们的型号分别为 CPU SR_x 和 CPU ST_x。型号中的 x 为 CPU 集成的 I/O 总点数，分别为 20、30、40 和 60 点。

标准型 CPU 有一个以太网端口和一个 RS-485 端口，可以扩展一块信号板和最多 6 块扩展模块。紧凑型 CPU 只有 RS-485 端口，没有扩展功能。

标准型 CPU 最多 6 个高速计数器，最高频率 200kHz。场效应晶体管输出型的标准型 CPU 最多可以输出 3 路 100 kHz 的脉冲。紧凑型 CPU 最多 4 个 100kHz 的高速计数器，没有脉冲输出功能和实时时钟。

S7-200 SMART 可以用手机存储卡传送程序、更新 CPU 的固件和恢复 CPU 的出厂设置。操作完成后存储卡应从 CPU 的存储卡插槽中拔出。

2. 通信功能

S7-200 和 S7-200 SMART 的 RS-485 端口都有自由端口模式、Modbus RTU 协议、USS 协议和 OPC 通信功能，它们都有 PROFIBUS-DP 从站模块。S7-200 SMART 标准型 CPU 集成的以太网接口还有 S7 协议通信、开放式用户通信和 OPC 通信功能。通过以太网，标准型 CPU 之间、标准型 CPU 和 S7-200/300/400/1200 CPU 之间都可以通信。

S7-200 的以太网通信模块 CP 243-1 只有 S7 协议通信功能，但是其价格比 S7-200

SMART 带 60 个 I/O 点的标准型 CPU 还要高。

3. 位置控制功能

S7-200 SMART 有很强的位置控制功能。支持 PWM/PTO 输出方式以及多种运动控制模式，可以用向导设置运动曲线，相当于集成了 3 块 S7-200 昂贵的位置控制模块 EM 253 的功能。

4. I/O 扩展模块

表 1-6 和表 1-7 分别列出了 S7-200 SMART 的数字量扩展模块和模拟量扩展模块。

表 1-6 数字量扩展模块

仅输入/仅输出		输入/输出组合	
8 点数字量输入	8 点继电器型输出	8 点数字量输入/8 点晶体管型输出	16 点数字量输入/16 点晶体管型输出
16 点数字量输入	16 点晶体管型输出	8 点数字量输入/8 点继电器型输出	16 点数字量输入/16 点继电器型输出
8 点晶体管型输出	16 点继电器型输出	—	—

表 1-7 模拟量扩展模块

型 号	描 述	型 号	描 述
EM AE04	4 点模拟量输入	EM AM03	2 点模拟量输入/1 点模拟量输出
EM AE08	8 点模拟量输入	EM AM06	4 点模拟量输入/2 点模拟量输出
EM AQ02	2 点模拟量输出	EM AR02	2 点热电阻输入
EM AQ04	4 点模拟量输出	EM AR04	4 点热电阻输入
—	—	EM AT04	4 点热电偶输入

5. 信号板与通信模块

S7-200 SMART 有 5 种可以安装在 CPU 内的信号板：2 点 DC 24V 数字量直流输入/2 点数字量场效应晶体管直流输出信号板 SB DT04，1 点模拟量输入信号板 SB AE01，1 点模拟量输出信号板 SB AQ01，RS485/RS232 信号板 CM01，以及电池信号板 SB BA01。

EM DP01 是 PROFIBUS-DP 通信模块，可以作 DP 从站和 MPI 从站。

6. S7-200 SMART 与 S7-200 的指令和软件功能的比较

S7-200 SMART 用 GET/PUT 指令取代了 S7-200 的网络读、写指令 NETR/NETW。用获取非致命错误代码指令 GET_ERROR 取代了诊断 LED 指令 DIAG_LED。S7-200 SMART 增加了获取 IP 地址指令 GIP 和设置 IP 地址指令 SIP，以及指令库中的开放式用户通信使用的 8 条指令。S7-200 和 S7-200 SMART 的其他指令完全相同。

与 S7-200 相比，S7-200 SMART 的堆栈由 9 层增加到 32 层，中断程序调用子程序的嵌套层数由 1 层增加到 4 层。

S7-200 SMART 的编程软件功能强大，使用方便。可以同时打开和显示多个窗口。变量表、输出窗口、交叉引用表、数据块、符号表、状态图表均可以浮动、隐藏和停靠在程序编辑器或软件界面的四周。S7-200 SMART 的帮助增加了搜索功能。编程软件提供了高速计数器、运动控制、PID、PWM、GET/PUT、文本显示和数据日志的向导。有 PID 整定控制面板、运动控制面板，以及组态 SMART 驱动器的工具。编程软件自带 Modbus RTU 指令库和 USS 协议指令库，而 S7-200 需要用户安装这些库。

S7-200 SMART 详细的情况见作者主编的本科教材《S7-200 SMART PLC 编程及应用》和高职高专教材《S7-200 SMART PLC 应用教程》。

1.5 逻辑运算与 PLC 的工作原理

1.5.1 用触点和线圈实现逻辑运算

在数字量控制系统中，变量仅有两种相反的工作状态，例如高电平和低电平、继电器线圈的通电和断电、触点的接通和断开，可以用逻辑代数中的 1 和 0 来表示它们，在波形图中，用高电平表示 1 状态，用低电平表示 0 状态。

“与”“或”“非”逻辑运算的输入/输出关系如表 1-8 所示，等式的左边为输出量。用继电器电路或梯形图可以实现“与”“或”“非”逻辑运算（见图 1-11）。用多个触点的串、并联电路可以实现复杂的逻辑运算。

继电器的线圈通电时，其常开触点接通，常闭触点断开；线圈断电时，其常开触点断开，常闭触点闭合。梯形图中的位元件（例如 PLC 的输出点 Q）的触点和线圈也有类似的关系。

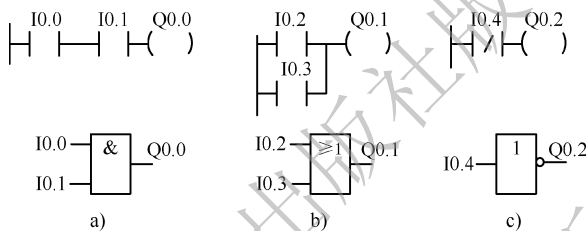


图 1-11 基本逻辑运算

a) 与 b) 或 c) 非

表 1-8 逻辑运算关系表

与		或		非	
$Q0.0 = I0.0 \cdot I0.1$		$Q0.1 = I0.2 + I0.3$		$Q0.2 = \overline{I0.4}$	
I0.0	I0.1	I0.2	I0.3	I0.4	I0.4
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	0	1	
1	1	1	1	1	

图 1-12 是用交流接触器控制异步电动机的主电路、控制电路和有关的波形图。接触器 KM 的结构和工作原理与继电器的基本相同，区别仅在于继电器触点的额定电流较小（例如几十毫安），而接触器是用来控制大电流负载的，例如它可以控制额定电流为几十安甚至数百安的异步电动机。

图中的热继电器 FR 用于过载保护，电动机过载时，经过一段时间后，FR 的常闭触点断开，使 KM 的线圈断电，电动机停止运行。

按下起动按钮 SB1，它的常开触点接通，电流经过 SB1 的常开触点和停止按钮 SB2、FR 的常闭触点，流过交流接触器 KM 的线圈。接触器的衔铁被吸合，使主电路中 KM 的 3 对常开触点闭合。异步电动机 M 的三相电源接通，电动机开始运行，控制电路中接触器 KM 的辅助常开触点同时接通。放开起动按钮后，SB1 的常开触点断开，电流经 KM 的辅

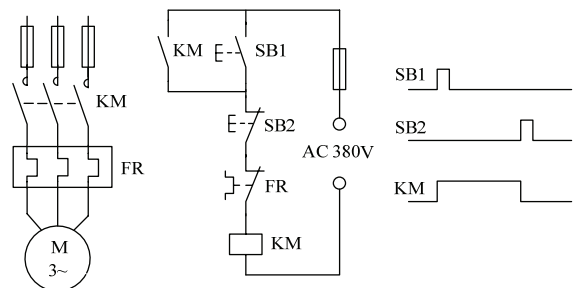


图 1-12 异步电动机的主电路、控制电路与波形图

助常开触点和两个常闭触点流过 KM 的线圈，电动机继续运行。KM 的辅助常开触点实现的这种功能称为“自锁”或“自保持”，它使继电器电路具有类似于 R-S 触发器的记忆功能。

在电动机运行时按下停止按钮 SB2，它的常闭触点断开，使 KM 的线圈失电，KM 的主触点断开，异步电动机的三相电源被切断，电动机停止运行，同时控制电路中 KM 的辅助常开触点断开。当停止按钮 SB2 被放开，其常闭触点闭合后，KM 的线圈仍然失电，电动机继续保持停止运行状态。图 1-12 给出了有关信号的波形图，图中用高电平表示 1 状态（线圈通电、按钮被按下），用低电平表示 0 状态（线圈断电、按钮被放开）。

图 1-12 中的继电器电路称为起动-保持-停止电路，简称为“起保停”电路。它实现的逻辑运算可以用逻辑代数式表示为

$$KM = (SB1 + KM) \cdot \overline{SB2} \cdot \overline{FR}$$

在继电器电路图和梯形图中，线圈的状态是输出量，触点的状态是输入量。上式左边的 KM 与图中的线圈相对应，右边的 KM 与 KM 的常开触点相对应，上画线表示逻辑“非”， $\overline{SB2}$ 与 SB2 的常闭触点相对应。上式中的加号表示逻辑“或”，乘号（小圆点，也可以改用*号）表示逻辑“与”。

与普通算术运算“先乘除后加减”类似，逻辑运算的规则为先“与”后“或”。为了先作“或”运算（触点的并联），用括号将“或”运算式括起来，括号中的运算优先执行。

1.5.2 PLC 的工作原理



1. PLC 的操作模式

PLC 有两种操作模式，即 RUN（运行）模式与 STOP（停止）模式。在 CPU 模块的面板上用 RUN 和 STOP 发光二极管（LED）显示当前的操作模式。

在 RUN 模式，通过执行反映控制要求的用户程序来实现控制功能。在 STOP 模式，CPU 不执行用户程序，可以用编程软件将用户程序和硬件组态信息下载到 PLC。

如果有致命错误，在消除它之前不允许从 STOP 模式进入 RUN 模式。PLC 操作系统储存非致命错误供用户检查，但是不会从 RUN 模式自动进入 STOP 模式。

CPU 模块上的模式开关在 STOP 位置时，将停止用户程序的运行；在 RUN 位置时，将启动用户程序的运行。模式开关在 STOP 或 TERM（terminal，终端）位置时，电源通电后 CPU 自动进入 STOP 模式；在 RUN 位置时，电源通电后自动进入 RUN 模式。

模式开关在 RUN 位置时，STEP 7-Micro/WIN 与 PLC 之间建立起通信连接后，单击工具栏上的运行按钮 ，确认后进入 RUN 模式。单击停止按钮 ，确认后进入 STOP 模式。在程序中插入 STOP 指令，可以使 CPU 由 RUN 模式进入 STOP 模式。

2. PLC 的扫描工作方式

PLC 通电后，需要对硬件和软件作一些初始化工作。为了使 PLC 的输出及时地响应各种输入信号，初始化后反复不停地分阶段处理各种不同的任务（见图 1-13），这种周而复始的循环工作方式称为扫描工作方式，每次循环的时间称为扫描周期。在 RUN 模式，扫描周期由下面的 5 个阶段组成。

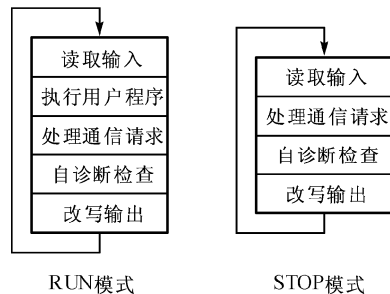


图 1-13 扫描过程

(1) 读取输入

在 PLC 的存储器中，有 128 点过程映像输入寄存器和 128 点过程映像输出寄存器，用来存放输入信号和输出信号的状态。

在读取输入阶段，PLC 把所有外部数字量输入电路的 0、1 状态读入过程映像输入寄存器。外接的输入电路闭合时，对应的过程映像输入寄存器为 1 状态（或称为 ON），梯形图中对应的输入点的常开触点接通，常闭触点断开。外接的输入电路断开时，对应的过程映像输入寄存器为 0 状态（或称为 OFF），梯形图中对应的输入点的常开触点断开，常闭触点接通。

如果没有启用模拟量输入滤波，CPU 在正常扫描周期中不会读取模拟量输入值。当程序访问模拟量输入时，将立即从扩展模块读取模拟量值。

(2) 执行用户程序

PLC 的用户程序由若干条指令组成，指令在存储器中顺序排列。在 STOP 模式不执行用户程序。在 RUN 模式的程序执行阶段，如果没有跳转指令，CPU 从第一条指令开始，逐条顺序地执行用户程序。

在执行指令时，从 I/O 映像寄存器或别的位元件的寄存器读出其 0、1 状态，并根据指令的要求执行相应的逻辑运算，运算的结果写入到相应的映像寄存器中，因此，各寄存器（只读的过程映像输入寄存器除外）的内容随着程序的执行而变化。

在程序执行阶段，即使外部输入信号的状态发生了变化，过程映像输入寄存器的状态也不会随之而变，输入信号变化了的状态只能在下一个扫描周期的读取输入阶段被读入。执行程序时，对输入/输出的读写通常是通过映像寄存器，而不是实际的 I/O 点，这样做有以下好处：

- 1) 在整个程序执行阶段，各输入点的状态是固定不变的，程序执行完后再用过程映像输出寄存器的值更新输出点，使系统运行稳定。

- 2) 用户程序读写 I/O 映像寄存器比读写 I/O 点快得多，这样可以提高程序的执行速度。

- 3) I/O 点是位实体，必须以位或字节为单位来存取，但是可以将映像寄存器作为位、字节、字或双字来存取。

(3) 处理通信请求

在处理通信请求阶段，执行通信所需的所有任务。

(4) 自诊断检查

自诊断测试功能用来保证固件、程序存储器和所有扩展模块正常工作。

(5) 改写输出

CPU 执行完用户程序后，将过程映像输出寄存器的 0、1 状态传送到输出模块并锁存起来。梯形图中某一输出位的线圈“通电”时，对应的过程映像输出寄存器的值为 1。在改写输出阶段，信号经输出模块隔离和功率放大后，继电器型输出模块中对应的硬件继电器的线圈通电，其常开触点闭合，使外部负载通电工作。若梯形图中输出点的线圈“断电”，对应的过程映像输出寄存器的值为 0。在改写输出阶段，将它送到继电器型输出模块，对应的硬件继电器的线圈断电，其常开触点断开，外部负载断电，停止工作。

当程序访问模拟量输出模块时，模拟量输出被立即刷新，而与扫描周期无关。

当 CPU 的操作模式从 RUN 变为 STOP 时，数字量输出被置为系统块中的输出表定义的状态，或保持当时的状态（见 2.5.3 节），默认的设置是将所有的数字量输出清零。

3. 中断程序的处理

如果在程序中使用了中断，中断事件发生时，CPU 停止正常的扫描工作方式，立即执行中断程序，中断功能可以提高 PLC 对中断事件的响应速度。

4. 立即 I/O 处理

在程序执行过程中使用立即 I/O 指令可以直接读、写 I/O 点的值。用立即 I/O 指令读输入点的值时，相应的过程映像输入寄存器的值不会被更新。用立即 I/O 指令来改写输出点时，相应的过程映像输出寄存器的值被更新。

5. PLC 的工作过程举例

下面用一个简单的例子来进一步说明 PLC 的扫描工作过程，图 1-14 中的 PLC 控制系统与图 1-12 中的继电器控制电路的功能相同。起动按钮 SB1 和停止按钮 SB2 的常开触点分别接在编号为 0.1 和 0.2 的输入端，接触器 KM 的线圈接在编号为 0.0 的输出端。如果热继电器 FR 动作（其常闭触点断开）后需要手动复位，可以将 FR 的常闭触点与接触器 KM 的线圈串联，这样可以少用一个 PLC 的输入点。

图 1-14 梯形图中的 I0.1 与 I0.2 是输入变量，Q0.0 是输出变量，它们都是梯形图中的编程元件。I0.1 与接在输入端子 0.1 上的 SB1 的常开触点和过程映像输入寄存器 I0.1 相对应，Q0.0 与接在输出端子 0.0 上的 PLC 内的输出电路和过程映像输出寄存器 Q0.0 相对应。

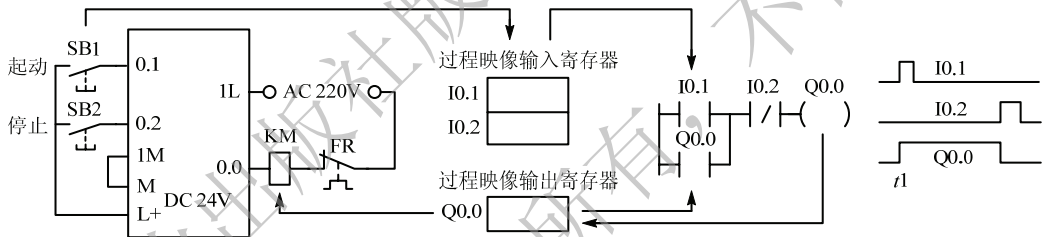


图 1-14 PLC 外部接线图与梯形图

梯形图以指令的形式储存在 PLC 的用户程序存储器中，图 1-14 中的梯形图与下面的 4 条指令相对应，“//”之后是该指令的注释。

LD	I0.1	//接在左侧“电源线”上的 I0.1 的常开触点
O	Q0.0	//与 I0.1 的常开触点并联的 Q0.0 的常开触点
AN	I0.2	//与并联电路串联的 I0.2 的常闭触点
=	Q0.0	//Q0.0 的线圈

梯形图完成的逻辑运算为

$$Q0.0 = (I0.1 + Q0.0) \cdot \overline{I0.2}$$

在读取输入阶段，CPU 将 SB1 和 SB2 的常开触点的接通/断开状态读入相应的过程映像输入寄存器，外部触点接通时将二进制数 1 存入寄存器，反之存入 0。

执行第一条指令时，从过程映像输入寄存器 I0.1 中取出二进制数，并存入堆栈的栈顶，堆栈是存储器中的一片特殊的区域，其功能和结构将在 3.3.1 节中介绍。

执行第二条指令时，从过程映像输出寄存器 Q0.0 中取出二进制数，并与栈顶中的二进制数相“或”（触点的并联对应于“或”运算），运算结果存入栈顶。运算结束后只保留运算结果，不保留参与运算的数据。

执行第三条指令时，因为是常闭触点，取出过程映像输入寄存器 I0.2 中的二进制数后，将它取反（将 0 变为 1，1 变为 0），取反后与前面的运算结果相“与”（电路的串联对应于“与”运算），然后存入栈顶。

执行第四条指令时，将栈顶中的二进制数传送到 Q0.0 的过程映像输出寄存器。

在修改输出阶段，CPU 将各过程映像输出寄存器中的二进制数传送给输出模块并锁存起来，如果 Q0.0 中存放的是二进制数 1，外接的 KM 线圈将通电，反之将断电。

I0.1、I0.2 和 Q0.0 的波形中的高电平表示按下按钮或 KM 线圈通电，当 $t < t_1$ 时，读入过程映像输入寄存器 I0.1 和 I0.2 的值均为二进制数 0，此时过程映像输出寄存器 Q0.0 的值亦为 0，在程序执行阶段，经过上述逻辑运算过程之后，运算结果仍为 $Q0.0 = 0$ ，所以 KM 的线圈处于断电状态。 $t = t_1$ 时，按下起动按钮 SB1，I0.1 变为 ON，经逻辑运算后 Q0.0 也变为 ON，在输出处理阶段，将 Q0.0 对应的过程映像输出寄存器中的数据 1 送到输出模块，输出模块中与 Q0.0 对应的物理继电器的常开触点接通，接触器 KM 的线圈通电。

PLC 在 RUN 模式时，执行一次图 1-13 所示的扫描操作所需的时间称为扫描周期，其典型值为 1~100ms。执行用户程序所需的时间与用户程序的长短、指令的种类和 CPU 执行指令的速度有很大的关系。用户程序较长时，指令执行时间在扫描周期中占相当大的比例。

6. 输入/输出滞后时间

输入/输出滞后时间又称为系统响应时间，是指 PLC 的外部输入信号发生变化的时刻至它控制的有关外部输出信号发生变化的时刻之间的时间间隔，它由输入电路滤波时间、输出电路的滞后时间和因扫描工作方式产生的滞后时间这 3 部分组成。

数字量输入点的滤波器用来滤除由输入端引入的干扰噪声，消除因外接输入触点动作时产生的抖动引起的不良影响，CPU 模块集成的输入点的输入滤波器延迟时间可以用系统块来设置。输出模块的滞后时间与模块的类型有关，继电器型输出电路的滞后时间一般在 10ms 左右；场效应晶体管型输出电路的滞后时间最短为微秒级，最长的在 100 μ s 以上。

由扫描工作方式引起的滞后时间最长可达两三个扫描周期。PLC 总的响应延迟时间一般只有几毫秒至几十毫秒，对于一般的系统来说是无关紧要的。要求输入/输出滞后时间尽量短的系统，可以选用扫描速度快的 PLC，或采用硬件中断、立即输入/立即输出等措施。

1.6 PLC 控制系统的硬件可靠性措施

PLC 是专门为工业环境设计的控制装置，一般不需要采取什么特殊措施，就可以直接在工业环境使用。但是如果环境过于恶劣，电磁干扰特别强烈，或安装使用不当，都不能保证系统的正常安全运行。干扰可能会使 PLC 接收到错误的信号，造成误动作，或使 PLC 内部的数据丢失，严重时甚至会使系统失控。在系统设计时，应采取相应的可靠性措施，以消除或减小干扰的影响，保证系统的正常运行。

1. 电源的抗干扰措施

电源是干扰进入 PLC 的主要途径之一，电源干扰主要是通过供电线路的分布电容和分布电感的耦合产生的，各种大功率用电设备是主要的干扰源。

在干扰较强或对可靠性要求很高的场合，可以在 PLC 的交流电源输入端加接带屏蔽层的隔离变压器和低通滤波器。

隔离变压器可以抑制从电源线窜入的外来干扰，提高抗高频共模干扰的能力。高频干扰信号不是通过变压器绕组的耦合，而是通过一次、二次绕组间的分布电容传递的。在一次、二次绕组之间加绕屏蔽层，并将它和铁心一起接地，可以减少绕组间的分布电容，提高抗高频干扰的能力。可以在互联网上搜索“电源滤波器”“抗干扰电源”和“净化电源”等关键词，选用相应的抗电源干扰的产品。

2. 布线的抗干扰措施

PLC 不能与高压电器安装在同一个开关柜内，在柜内 PLC 应远离动力线，二者之间的距离应大于 200mm。中性线与相线、公共线与信号线应成对布线。I/O 线与电源线应分开走线，数字量、模拟量 I/O 线应分开敷设，交流信号与直流信号应分别使用不同的电缆。不同类型的导线应分别装入不同的电缆管或电缆槽中，并使其有尽可能大的空间距离。距离较长的数字量 I/O 线应采用屏蔽线。应为可能遭雷电冲击的线路安装合适的浪涌抑制设备，干扰较严重时也应设置浪涌抑制设备。

信号线和它的返回线绞合在一起，能减少感性耦合引起的干扰，绞合越靠近端子效果越好。S7-200 的交流电源线和 I/O 点之间的隔离电压为 AC 1500V，可以作为交流线和低压电路之间的安全隔离。

3. 模拟量信号的处理

模拟量信号和高速脉冲信号的传输线应使用双屏蔽的双绞线（每对双绞线和整个电缆都有屏蔽层）。不同的模拟量信号线应独立走线，它们有各自的屏蔽层，以减少线间的耦合。不要把不同的模拟量信号置于同一个公共返回线。

如果模拟量输入/输出信号距离 PLC 较远，应采用 4~20mA 的电流传输方式，而不是易受干扰的电压传输方式。干扰较强的环境应选用有光隔离的模拟量 I/O 模块，使用分布电容小、干扰抑制能力强的配电器为变送器供电，以减少对 PLC 的模拟量输入信号的干扰。应短接未使用的 A-D 通道的输入端，以防止干扰信号进入 PLC，影响系统的正常工作。模拟量输入信号的数字滤波是减轻干扰影响的有效措施。

4. PLC 的接地

(1) 控制设备的两种地

1) 安全保护地（或称电磁兼容性地）。车间里一般有保护接地网络。为了保证操作人员的安全，应将电动机的外壳和控制屏的金属屏体连接到安全保护地。CPU 模块上的 PE（保护接地）端子应连接到大地或者柜体上。

2) 信号地（或称控制地、仪表地）。它是电子设备的电位参考点，例如 CPU 模块的传感器电源输出端子中的 M 端子应接到信号地。PLC 和变频器通信时，应将 PLC 的 RS-485 端口的第 5 脚（5V 电源的负极）与变频器的模拟量输入信号的 0V 端子连接到信号地。

(2) 控制地应一点接地

控制系统中所有的控制设备需要接信号地的端子应保证一点接地。首先以控制屏为单位，将屏内各设备需要接信号地的端子用电缆夹连接到等电位母线上，然后用规定面积的等电位连接导线将各个屏的信号地端子连接到接地网络的某一点。西门子推荐的等电位连接导线的截面积为 16mm^2 。信号地最好采用单独的接地装置。

连接参考电位不同的设备的通信口，可能导致在连接电缆中产生预想不到的电流。这种电流可能导致通信错误或设备损坏。应确保用通信电缆连接的所有设备都共用一个公共电路

参考点或者进行隔离，以防止出现意外电流。S7-200 的 RS-485 通信端口没有绝缘隔离，可以使用有隔离的 RS-485 中继器来连接具有不同地电位的设备。

(3) 屏蔽电缆屏蔽层的接地

一般情况下，屏蔽电缆的屏蔽层应两端接金属机壳，并确保大面积接触金属表面，以便能承受高频干扰。为了减少屏蔽层的电流，两端接地的屏蔽层应与等电位连接导线并联。

在少数情况下，模拟量电缆的屏蔽层可以在控制柜一端接地，另一端通过一个高频小电容接地。如果屏蔽层两端的差模电压不高，并且连接到同一地线上时，也可以将屏蔽层的两端直接接地。

不要使用金属箔屏蔽层电缆，它的屏蔽效果仅有编织物屏蔽层电缆的五分之一。

(4) 信号地不要通过保护接地网络连接

如果将各控制屏或设备的信号地就近连接到当地的安全保护地网络上，强电设备的接地电流可能在两个接地点之间产生较大的电位差，干扰控制系统的工作，严重时可能烧毁设备。

有不少企业因为在车间烧电焊，烧毁了控制设备的通信端口和通信设备。电焊机的二次电压很低，但是焊接电流很大。焊接线的“地线”一般搭在与保护接地网络连接的设备的金属构件上。如果电焊机的接地线的接地点离焊接点较远，焊接电流通过保护接地网络形成回路。如果各设备的信号地不是一点接地，而是就近接到安全保护地网络上，焊接电流有可能窜入通信网络，烧毁设备的通信端口或通信模块。

5. 防止变频器干扰的措施

现在 PLC 越来越多地与变频器一起使用，经常会遇到变频器干扰 PLC 的正常运行的故障，变频器已经成为 PLC 最常见的干扰源。

变频器的主电路为交-直-交变换电路，工频电源被整流为直流电压，输出的是基波频率可变的高频脉冲信号，载波频率可能超过 10kHz。变频器的输入电流为含有丰富的高次谐波的脉冲波，它会通过电力线干扰其他设备。高次谐波电流还通过电缆向空间辐射，干扰邻近的电气设备。

可以在变频器输入侧与输出侧串接电抗器，或安装谐波滤波器（见图 1-15），以吸收谐波，抑制高频谐波电流。

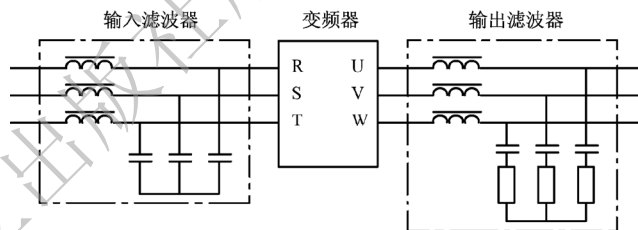


图 1-15 变频器的输入滤波器与输出滤波器

将变频器放在控制柜内，并将其金属外壳接地，对高频谐波有屏蔽作用。变频器的输入、输出电流（特别是输出电流）中含有丰富的谐波，所以主电路也是辐射源。PLC 的信号线和变频器的输出线应分别穿管敷设，变频器的输出线一定要使用屏蔽电缆或穿钢管敷设，以减轻对其他设备的辐射干扰和感应干扰。

变频器应使用专用接地线，并且用粗短线接地，其他邻近的电气设备的接地线必须与变频器的接地线分开。

可以对受干扰的 PLC 采用屏蔽措施，在 PLC 的电源输入端串入滤波电路或安装隔离变压器，以减小谐波电流的影响。

6. 强烈干扰环境中的隔离措施

一般情况下，PLC 的输入/输出信号采用内部的隔离措施就可以保证系统的正常运行。因此一般没有必要在 PLC 外部再设置抗干扰隔离器件。

在发电厂等工业环境，空间极强的电磁场和高电压、大电流断路器的通断将会对 PLC 产生强烈的干扰。由于现场条件的限制，有时很长的强电电缆和 PLC 的低压控制电缆只能敷设在同一个电缆沟内，强电干扰在 PLC 的输入线上产生的感应电压和感应电流相当大，可能使 PLC 输入端的光耦合器中的发光二极管发光，使 PLC 产生误动作。可以用小型继电器来隔离用长线引入 PLC 的数字量信号。S7-200 的数字量输入模块的最小逻辑 1 信号电流为 1mA，而小型继电器的线圈吸合电流为数十毫安，强电干扰信号通过电磁感应产生的能量一般不会使隔离用的继电器误动作。来自开关柜内和距离开关柜不远的输入信号一般没有必要用继电器来隔离。

为了提高抗干扰能力，对长距离的串行通信信号，可以考虑用光纤来传输和隔离，或使用带光耦合器的通信端口。

7. PLC 输出的可靠性措施

如果用 PLC 驱动交流接触器，应将额定电压为 AC 380V 的交流接触器的线圈换成 220V 的。在负载要求的输出功率超过 PLC 的允许值时，应设置外部继电器。PLC 输出电路的小型继电器的触点小，断弧能力差，不能直接用于 DC 220V 的电路，必须通过外部继电器驱动 DC 220V 的负载。

8. 感性负载的处理

感性负载（例如继电器、接触器的线圈）具有储能作用，PLC 内控制它的触点或场效应晶体管断开时，电路中的感性负载会产生高于电源电压数倍甚至数十倍的反电势。触点接通时，会因触点的抖动产生电弧，它们都会对系统产生干扰。对此可以采取下述的措施。

输出端接有直流感性负载时，应在它两端并联一个续流二极管。如果需要更快的断开时间，可以串接一个稳压管（见图 1-16），二极管可以选 IN4001，场效应晶体管输出可以选 8.2V/5W 的稳压管，继电器输出可以选 36V 的稳压管。

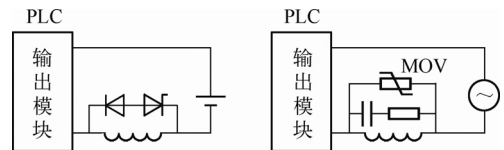


图 1-16 输出电路感性负载的处理

输出端接有 AC 220V 感性负载时，应在它两端并联 RC 串联电路（见图 1-16），可以选 0.1 μ F 的电容和 100~120 Ω 的电阻。电容的额定电压应大于电源峰值电压。要求较高时，还可以在负载两端并联压敏电阻，其压敏电压应大于额定电压有效值的 2.2 倍。

为了减小电动机和电力变压器投切时产生的干扰，可以在 PLC 的电源输入端设置浪涌电流吸收器。

1.7 习题

1. 填空

1) PLC 主要由_____、_____、_____和_____组成。

- 2) 继电器的线圈“断电”时，其常开触点_____，常闭触点_____。
 - 3) 外部的输入电路接通时，对应的过程映像输入寄存器为_____状态，梯形图中后者的常开触点_____，常闭触点_____。
 - 4) 若梯形图中输出 Q 的线圈“断电”，对应的过程映像输出寄存器为_____状态，在修改输出阶段后，继电器型输出模块中对应的硬件继电器的线圈_____，其常开触点_____，外部负载_____。
2. RAM 与 EEPROM 各有什么特点？
 3. 数字量输出模块有哪几种类型？它们各有什么特点？
 4. 简述 PLC 的扫描工作过程。
 5. 频率变送器的量程为 45~55Hz，输出信号为 DC 0~10V，模拟量输入模块输入信号的量程为 DC 0~10V，转换后的数字量为 0~32000，设转换后得到的数字为 N ，试求以 0.01Hz 为单位的频率值。
 6. 布线时应采取哪些抗干扰措施？
 7. 分布很广的系统在接地时应注意哪些问题？
 8. 防止变频器干扰应采取哪些措施？
 9. 在有强烈干扰的环境下，可以采取什么可靠性措施？
 10. 电缆的屏蔽层应怎样接地？
 11. 对 PLC 的感性负载应采取什么抗干扰措施？

第 2 章 STEP 7-Micro/WIN 编程软件使用指南


2.1 编程软件概述

2.1.1 编程软件的安装与项目的组成

1. 编程软件的安装

双击配套资源的文件夹“STEP 7-Micro_WIN V40+SP9”中的 setup.exe，开始安装编程软件，使用默认的安装语言 English，单击“Next”按钮，出现“Preparing Setup”（准备安装）窗口，开始安装。单击欢迎（Welcome）窗口中的“Next”按钮，再单击“License Agreement”（许可协议）窗口的“Yes”按钮。单击 Choose Destination Location 窗口的“Browse”按钮，可以选择软件安装的目标文件夹。单击“Next”按钮，开始安装。

安装结束后，出现“InstallShield Wizard Complete”对话框，表示安装完成。去掉多选框“**Yes, I want to view the Read Me file now**”中的对钩，不阅读软件的自述文件。单击“Finish”按钮退出安装程序。

安装成功后，双击桌面上的 STEP 7-Micro/WIN 图标，打开编程软件，看到的是英文的界面。执行菜单命令“Tools”（工具）→“Options”（选项），单击出现的对话框左边的“General”（常规），在“General”选项卡中，选择 Language（语言）为“Chinese”（中文）。先后单击“确认”按钮和“否”按钮，退出 STEP 7-Micro/WIN 后，再进入该软件，界面和帮助文件均已变成中文的了。

安装好“STEP 7-Micro_WIN V40”后，再安装配套资源中的“STEP 7-Micro WIN V32”指令库。




视频“安装编程软件”可通过扫描二维码 2-1 播放。

二维码 2-1

2. 指令树与浏览条

图 2-1 是 STEP 7-Micro/WIN 的界面，指令树是包含所有项目对象和所有指令的树型视图。双击指令树中的某个对象，将会打开对应的窗口。可以将常用的指令拖放到指令树的“收藏夹”中。

单击指令树中文件夹左边带加、减号的小方框，可以打开或关闭该文件夹。也可以双击某个文件夹打开或关闭它。右键单击指令树中的某个文件夹，可以用快捷菜单中的命令做打开、插入等操作，允许的操作与具体的文件夹有关。右键单击文件夹中的某个对象，可以做打开、剪切、复制、粘贴、插入、删除、重命名、设置属性等操作，允许的操作与具体的对象有关。

将光标放到指令树右侧的垂直分界线上，光标变为水平方向的双向箭头，按住鼠标左键，移动鼠标，可以拖动垂直分界线，调节指令树的宽度。

浏览条的功能与指令树重叠，占的面积也不小。可以用鼠标右键单击浏览条，执行出现的快捷菜单中的“隐藏”命令，关闭浏览条。

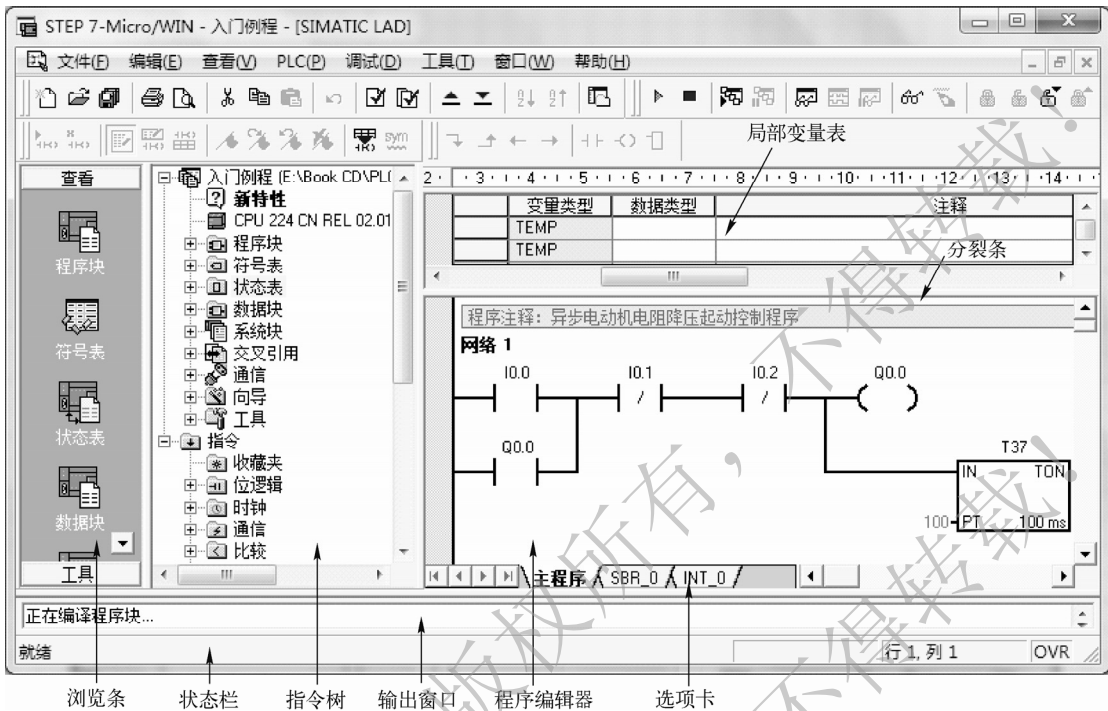


图 2-1 STEP 7-Micro/WIN 的界面

3. 程序编辑器

程序编辑器窗口包含局部变量表和程序视图。局部变量表用来对局部变量赋值，局部变量仅限于在它所在的程序中使用。

将光标放到局部变量表和程序视图之间的分裂条上，光标变为垂直方向的双向箭头 \updownarrow ，按住鼠标左键上下移动鼠标，可以改变分裂条的位置。单击程序编辑器窗口底部的选项卡，可以选择显示哪一个程序。

4. 输出窗口

在编译程序或指令库后，输出窗口提供编译的信息。双击输出窗口中某条程序编译后的错误信息，将会在程序编辑器窗口中显示错误所在的程序块和网络，光标在出错的位置。

5. 状态栏

状态栏位于主窗口底部，提供软件中执行的操作的状态信息。例如光标所在的网络号、网络中的行号和列号，当前是插入（INS）模式还是覆盖（OVR）模式。可以用计算机的〈Insert〉键切换这两种模式。

6. 项目的组成

项目包括下列基本组件：

(1) 程序块

程序块由可执行的代码和注释组成，可执行的代码由主程序（OB1）、可选的子程序和中断程序组成。代码被编译并下载到 PLC，程序注释被忽略。

(2) 数据块

数据块用来对 V 存储器（变量存储器）赋初值，其使用方法详见 4.8.1 节。

(3) 系统块

系统块用来设置系统的参数（见 2.5 节），系统块下载到 PLC 后才起作用。

(4) 符号表

符号表允许程序员用符号来代替存储器的地址，符号地址便于记忆，使程序更容易理解。符号表中定义的符号为全局变量，可以用于所有的程序。程序编译后下载到 PLC 时，所有的符号地址被转换为绝对地址，符号信息不会下载到 PLC。

(5) 状态表

状态表用表格或趋势图来监视、修改和强制程序执行时指定的变量的状态，状态表并不下载到 PLC。

(6) 交叉引用表

交叉引用表用于检查程序中地址的赋值情况，可以防止无意间的重复赋值。

打开指令树中的“交叉引用”文件夹，双击其中的“交叉引用”，可以打开交叉引用表（见图 2-2）。

交叉引用表列举出程序中同一个地址所有的触点、线圈等在哪一个程序块的哪一个网络中出现，以及使用的指令助记符。单击交叉引用表下面的“字节使用”或“位使用”选项卡，还可以查看哪些存储器区域已经被使用，是作为位（b）使用，还是作为字节（B）、字（W）或双字（D）使用。在 RUN 模式下编辑程序时，可以查看程序当前正在使用的正向、负向转换触点的编号（见 2.4.4 节）。交叉引用表并不下载到 PLC，程序编译成功后才能看到交叉引用表的内容。

	元素	块	位置	
1	下限位:IO.1	手动程序 (SBR1)	网络 2	↑↓
2	下限位:IO.1	回原点程序 (SBR)	网络 4	↑↓
3	下限位:IO.1	自动程序 (SBR3)	网络 4	↑↓
4	下限位:IO.1	自动程序 (SBR3)	网络 8	↑↓
5	下限位:IO.1	自动程序 (SBR3)	网络 15	↑↓
6	上限位:IO.2	公用程序 (SBR0)	网络 1	↑↓
7	上限位:IO.2	手动程序 (SBR1)	网络 2	↑↓
8	上限位:IO.2	手动程序 (SBR1)	网络 3	↑↓

图 2-2 交叉引用表

双击交叉引用表中的某一行，可以显示出该行的操作数和指令所在的网络。

视频“编程软件使用入门”可通过扫描二维码 2-2 播放。



二维码 2-2

2.1.2 帮助功能的使用与 S7-200 的出错处理

1. 使用在线帮助

单击选中指令树中的某个文件夹或文件夹中的对象、单击某个菜单项、单击某个窗口、单击指令树或程序编辑器中的某条指令，按〈F1〉键可以得到选中的对象的在线帮助。

2. 从菜单获得帮助

可以用下述的各种方法从菜单获得帮助：

1) 执行菜单命令“帮助”→“目录和索引”，打开帮助窗口，借助目录浏览器可以寻找需要的帮助主题，窗口中的索引部分提供了按字母顺序排列的主题关键词，双击某一关键词，可以获得有关的帮助。

2) 执行菜单命令“帮助”→“这是什么”，出现带问号的光标，用它单击窗口中的用户接口（例如工具栏上的按钮、程序编辑器和指令树中的对象等），将会打开相应的帮助窗口。

3) 执行菜单命令“帮助”→“网上 S7-200”，可以访问为 S7-200 提供技术支持和产品信息的西门子互联网站。单击“中文”可以切换到中文显示模式。



二维码 2-3

视频“帮助功能的使用”可通过扫描二维码 2-3 播放。

3. S7-200 的致命错误

与 PLC 建立起通信连接后，使用菜单命令“PLC”→“信息”，可以查看错误信息，例如错误的代码。

致命错误使 PLC 停止执行程序，取决于错误的致命程度，致命错误使 PLC 无法执行某一功能或全部功能。CPU 检测到致命错误时，自动进入 STOP 模式，点亮 SF/DIAG（系统错误/诊断）和 STOP（停止）LED，并关闭输出。在消除致命错误之前，CPU 一直保持这种状态。

消除了引起致命错误的原因后，必须用下面的方法重新起 CPU。将 PLC 断电后再通电，将模式开关从 TERM 或 RUN 扳至 STOP 位置。如果发现其他致命错误条件，CPU 将会重新点亮系统错误 LED。

有些错误使 PLC 无法进行通信，此时在计算机上看不到 CPU 的错误代码。这表示硬件出错或 CPU 模块需要修理，修改程序或清除 PLC 的存储器不能消除这种错误。

4. 非致命错误

非致命错误会影响 CPU 的某些性能，但是不会使它无法执行用户程序和更新 I/O。有以下几类非致命错误：

(1) 运行时间错误

在 RUN 模式下发现的非致命错误会反映在特殊存储器标识位（SM）上，用户程序可以监控这些位。上电时 CPU 读取 I/O 配置，并将信息存储在 SM 中。如果在运行时 CPU 发现 I/O 配置变化，就会在模块错误字节中设置配置改变位。I/O 模块必须与保存在系统数据存储器中的 I/O 配置符合，CPU 才会对该位复位。它被复位之前，不会更新 I/O 模块。

(2) 程序编译错误

CPU 编译程序成功后才能下载程序，如果编译时检测到程序违反了编译规则，不会下载，并在输出窗口生成错误代码。CPU 的 EEPROM 中原有的程序依然存在，不会丢失。

(3) 程序执行错误

程序运行时，用户程序可能会产生错误。例如因为在程序执行过程中修改了一个编译时正确的间接地址指针，它可能指向超出范围的地址。可以用菜单命令“PLC”→“信息”来判断错误的类型，只有通过修改用户程序才能改正运行时的编程错误。


与某些错误条件相关的信息存储在特殊存储器（SM）中，用户程序可以用它们来监控和处理错误。例如可以用 SM5.0（I/O 错误）的常开触点控制 STOP 指令，在出现 I/O 错误时使 CPU 切换到 STOP 模式。


2.2 程序的编写与传送

2.2.1 生成用户程序

1. 创建项目或打开已有的项目

在为控制系统编程之前，首先应创建一个项目。打开 STEP 7-Micro/WIN，自动生成一个名为“项目 1”的新项目。执行菜单命令“文件”→“另存为”，将项目名称修改为“入门例程”（见配套资源中的同名例程），可以设置项目文件所在的文件夹。打开编程软件后，执

行菜单命令“文件”→“新建”，或者单击工具栏最左边的“新建项目”按钮，也可以生成一个新的项目。

执行菜单命令“文件”→“打开”，或者单击工具栏上对应的按钮，可以打开已有的项目。项目存放在扩展名为 mwp 的文件中。

2. 设置 PLC 的型号

在给 PLC 编程之前，应正确地设置其型号，执行菜单命令“PLC”→“类型”，在出现的对话框中（见图 2-3）设置 PLC 的型号。如果已经成功地建立起与 PLC 的通信连接，单击对话框中的“读取 PLC”按钮，可以通过通信读出 PLC 的型号和 CPU 的版本号。单击“确认”按钮后启用新的型号和版本号。



图 2-3 设置 PLC 的型号

3. 控制要求

下面用一个简单的例子，介绍怎样用 STEP 7-Micro/WIN 来编写、下载和调试梯形图程序。

控制两台异步电动机的 PLC 的外部接线图与梯形图如图 2-4 所示，输入电路使用 CPU 模块提供的 DC 24V 电源。按下起动按钮后，输出点 Q0.0 变为 ON，KM1 的线圈通电，起动 1 号电动机。同时定时器 T37 开始定时，5s 后 T37 的定时时间到，使 Q0.1 变为 ON，KM2 的线圈通电，起动 2 号电动机。按下停止按钮后，Q0.0 变为 OFF，1 号电动机停止运行；T37 被复位，其常开触点断开，Q0.1 变为 OFF，2 号电动机也停止运行。1 号电动机过载时，经过一定的时间后，接在 0.2 输入端的热继电器的常开触点闭合，使梯形图中 I0.2 的常闭触点断开，也会使两台电动机停止运行。

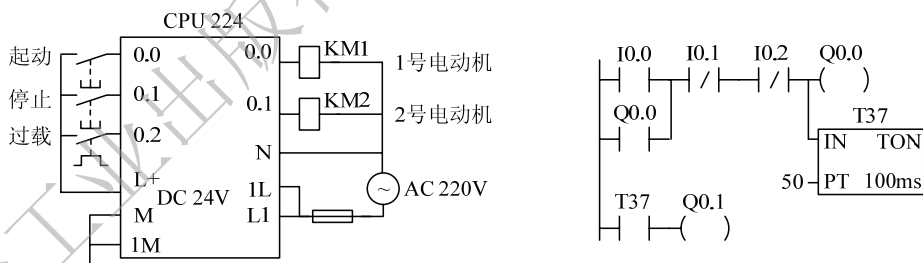


图 2-4 电动机控制的 PLC 外部接线图与梯形图

这是一个很简单的数字量控制系统，程序全部在主程序（OB1）中，没有子程序、中断程序和数据库，没有使用局部变量表。

本例对 CPU 模块和输入输出的参数没有特殊的要求，可以全部采用系统块的默认设置。

4. 编写用户程序

生成项目“入门例程”后，主程序 MAIN (OB1) 被自动打开，网络 1 最左边的箭头处有一个矩形光标（见图 2-5a）。单击工具栏上的触点按钮 **IF**，然后单击出现的对话框中的常开触点，在矩形光标所在的位置出现一个常开触点（见图 2-5b）。

触点上面红色的问号???.?表示地址未赋值，选中它以后输入触点的地址 I0.0，光标移动到触点的右边（见图 2-5c）。

单击工具栏上的触点按钮 **IF**，然后单击出现的对话框中的常闭触点，生成一个常闭触点，输入触点的地址 I0.1。用同样的方法生成 I0.2 的常闭触点（见图 2-5d）。单击工具栏上的线圈按钮 **CO**，然后单击出现的对话框中的 **CO**，生成一个线圈，设置线圈的地址为 Q0.0。

如果双击指令列表的“位逻辑”文件夹中的某个触点或线圈，将在光标处生成一个相同的元件。可以将常用的编程元件拖放到指令列表的“收藏夹”文件夹，在编程时使用它们。

将光标放到 I0.0 的常开触点的下面，生成 Q0.0 的常开触点（见图 2-5e）。将光标放到新生成的触点上，单击工具栏上的“向上连线”按钮 **↑**，将 Q0.0 的触点并联到它上面 I0.0 的触点上（见图 2-5f）。

将光标放到 I0.2 的触点上，单击工具栏上的“向下连线”按钮 **↓**，生成带双箭头的折线（见图 2-5f）。

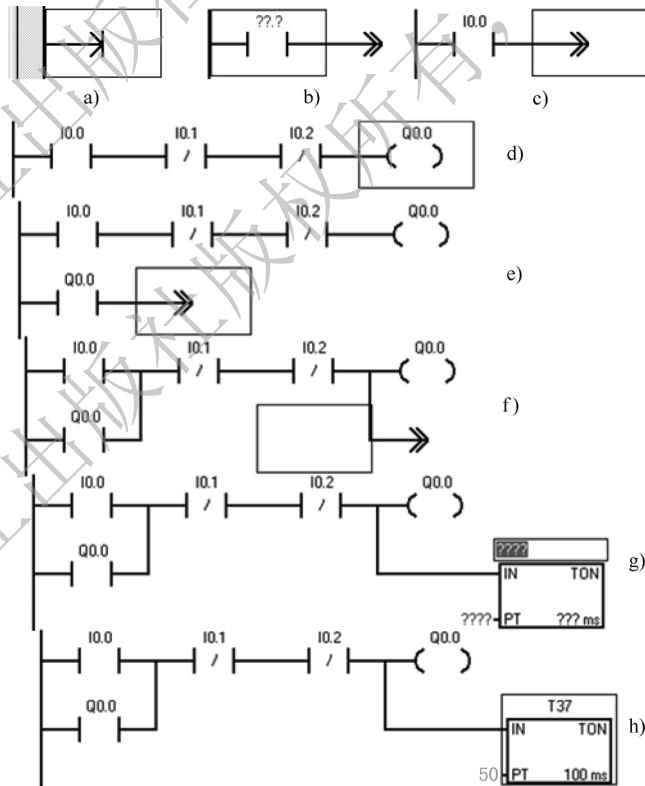



图 2-5 生成梯形图程序

有 3 种方法可以在程序中生成接通延时定时器：

1) 将指令列表的“定时器”文件夹中的“TON”图标拖放到图 2-5f 的双箭头所在的位置（见图 2-5g）。

2) 将光标放到图 2-5f 的双箭头上，然后双击指令列表中的“TON”，在光标处生成接通延时定时器。

3) 将光标放到图 2-5f 的双箭头上，单击工具栏上的“指令盒”按钮 ，向下拖动打开的指令列表中的垂直滚动条，显示出指令 TON 后单击它，在光标处生成接通延时定时器。

出现指令列表后键入 TON，将会显示和选中指令列表中的 TON，单击它将在光标处生成接通延时定时器。

在 TON 方框上面输入定时器的地址 T37。单击 PT 输入端的红色问号????，键入以 100ms 为单位的时间预设值 50 (5s)。图 2-5h 是网络 1 输入结束后的梯形图。



二维码 2-4

在网络 2 生成用 T37 的常开触点控制 Q0.1 线圈的电路（见图 2-15）。

视频“生成用户程序”可通过扫描二维码 2-4 播放。

5. 对网络的操作

梯形图程序被划分为若干个网络，编辑器自动给出网络的编号。一个网络只能有一块不能分开的独立电路，某些网络可能只有一条指令（例如 SCRE）。如果一个网络中有两块独立电路，在编译时将会出现错误，显示“无效网络或网络太复杂无法编译”。

语句表允许将若干个独立电路对应的语句放在一个网络中。没有语法错误的梯形图一定能转换为语句表程序。但是只有将语句表正确地划分为网络，才能将语句表转换为梯形图。不能转换的网络将在网络编号处用红色显示“无效”。



程序编辑器中输入的参数或数字用红色文本表示非法的语法，数值下面的红色波浪线表示数值超出范围或数值对该指令不正确。数值下面的绿色波浪线表示正在使用的变量或符号尚未定义。STEP 7-Micro/WIN 允许先编写程序，后定义变量和符号。

用鼠标左键单击程序区垂直电源线左边的灰色部分（见图 2-5a），对应的网络被选中，整个网络的背景色变为深蓝色。此时按住鼠标左键，在灰色区域内往上或往下拖动，可以选中相邻的若干个网络。可以用〈Delete〉键删除选中的网络，或者通过剪贴板复制、剪切、粘贴选中的网络中的程序。



用矩形光标选中梯形图中单个编程元件后，可以删除它，或者通过剪贴板复制和粘贴它。

选中指令列表或程序中的某条指令后按〈F1〉键，可以得到与该指令有关的在线帮助。

6. 打开和关闭注释

主程序、子程序和中断程序总称为程序组织单元（Program Organizational Unit, POU）。可以在程序编辑器中为 POU 和网络添加注释（见图 2-1）。单击工具栏上的“POU 注释”按钮  或“网络注释”按钮 ，可以打开或关闭对应的注释。


7. 编译程序

单击工具栏上的“编译”按钮 ，编译当前打开的程序块或数据块。单击“全部编译”按钮 ，编译全部项目组件（程序块、数据块和系统块）。如果程序有语法错误，编译后在编辑器下面出现的输出窗口将会显示错误和警告的个数，各条错误的原因和它们在程序中的位置。双击某一条错误，将会打开出错的程序块，用光标指示出错的位置。必须改正程序中

所有的错误才能下载。编译成功后，显示生成的程序和数据块的大小。

如果没有编译程序，在下载之前 STEP 7-Micro/WIN 将会自动地对程序进行编译，并在输出窗口显示编译的结果。

8. 设置程序编辑器的参数

执行菜单命令“工具”→“选项”，或单击工具栏上的“选项”按钮，打开“选项”窗口（见图 2-6）。单击选中左边窗口中的某个对象，再打开右边窗口的某个选项卡，可以进行有关的参数设置。

选中左边窗口的“程序编辑器”，可以用“符号地址”下拉列表框，选择只显示符号，或同时显示符号和地址。还可以设置网格（即矩形光标）的宽度、字符的大小、字体和样式（字符是否加粗或用斜体显示）。可以总体设置（在“类别”列表中选中“所有类别”），也可以分类设置。

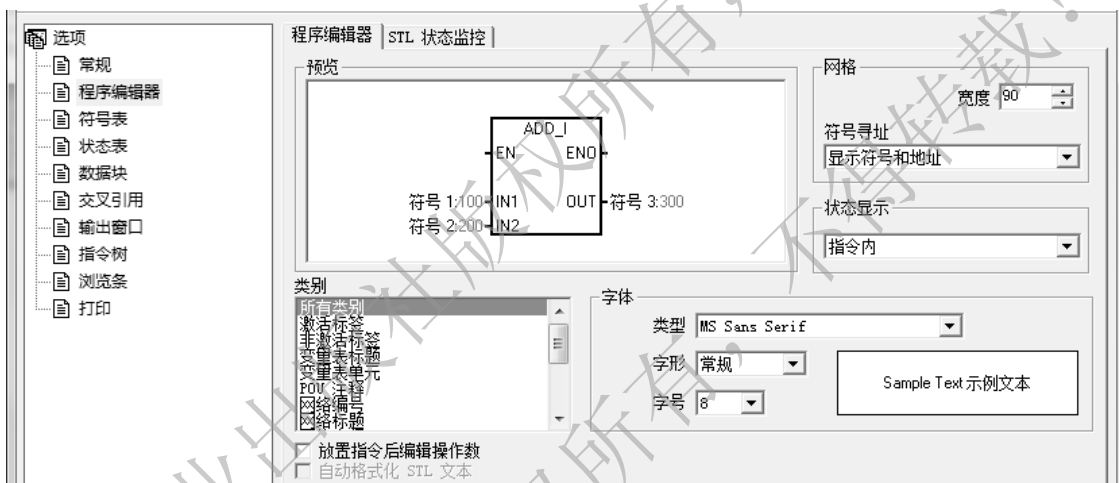


图 2-6 程序编辑器的参数设置

选中左边窗口的“常规”，在右边窗口的“常规”选项卡可以选择使用“IEC 1131-3”或“SIMATIC”编程模式，一般选择 SIMATIC 编程模式。还可以选择使用“国际”或“SIMATIC”助记符集，它们分别使用英语和德语的指令助记符。单击右边窗口的“默认值”选项卡中的“浏览”按钮，可以设置默认的文件位置。

选中图 2-6 左边窗口中的其他选项，可以设置 STEP 7-Micro/WIN 的其他参数，读者可以通过实际操作，来了解和熟悉设置编程软件参数的方法。

视频“程序编辑器的操作”可通过扫描二维码 2-5 播放。



二维码 2-5

2.2.2 下载与调试用户程序

为了实现 PLC 与计算机的通信，必须配备下列设备中的一种：

- 1) RS-232/PPI 多主站电缆或 USB/PPI 多主站电缆。
- 2) 一块插在个人计算机中的通信处理器（CP 卡），其价格较高。

1. RS-232/485 转换的 PC/PPI 多主站编程通信电缆

PC/PPI 多主站编程电缆用于 PLC 与有 RS-232 端口的计算机通信。订货号为 6ES7 901-3CB30-0XA0，带光电隔离，最大波特率为 187.5kbit/s。现在几乎所有的计算机都没有 RS-232

端口，所以这种编程电缆用得很少。使用时需要设置电缆护套的 8 个 DIP 开关（见图 2-7），设置方法见 S7-200 系统手册“技术规范”中的“S7-200 RS-232/PPI 多主站电缆”。

2. USB/RS-485 转换的 PC/PPI 多主站编程通信电缆

现在用得最多的是 USB/RS-485 转换的 PC/PPI 多主站编程通信电缆，订货号为 6ES7 901-3DB30-0XA0，带光电隔离，最大波特率为 187.5 kbit/s。它是一种即插即用设备，无须设置开关和安装驱动程序。

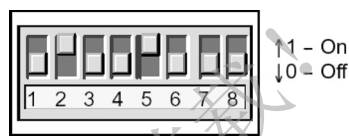


图 2-7 多主站电缆的 DIP 开关

3. 国产的 USB/PPI 编程通信电缆

现在市面上有很多国产的与西门子产品兼容的 USB 电缆，它们大多数是 USB/RS-232 转换器和 PC/PPI 适配器的组合，将 USB 端口映射为一个 RS-232C 端口（俗称为 COM 口）。注意低档的产品不能在 Windows 7 操作系统下使用，不支持 187.5kbit/s 的波特率。

用这种电缆连接好计算机的 USB 端口和 PLC 的 RS-485 端口，安装好 USB 电缆的驱动程序后，打开 Windows 的控制面板的“\系统和安全\系统”下面的“设备管理器”，在“端口（COM 和 LPT）”文件夹中，可以看到 USB 端口映射的 RS-232C 端口，例如“Prolific USB-to-Serial Bridge (COM3)”，或“USB-SERIAL CH340 (COM3)”，表示 USB 端口被映射为 RS-232 端口 COM3。端口的编号与使用计算机的哪个 USB 物理端口有关。

4. 设置 PG/PC 接口

双击指令树的“通信”文件夹中的“设置 PG/PC 接口”，选中打开的对话框中的“PC/PPI cable.PPI.1”，单击“属性”按钮，打开属性对话框的“本地连接”选项卡。如果是 USB/PPI 多主站编程通信电缆，设置连接到 USB。如果是 RS-232/PPI 多主站编程通信电缆，或国产兼容的编程电缆，设置计算机与 PLC 通信使用的 COM 端口，例如 USB 接口映射的 COM3（见图 2-8）。

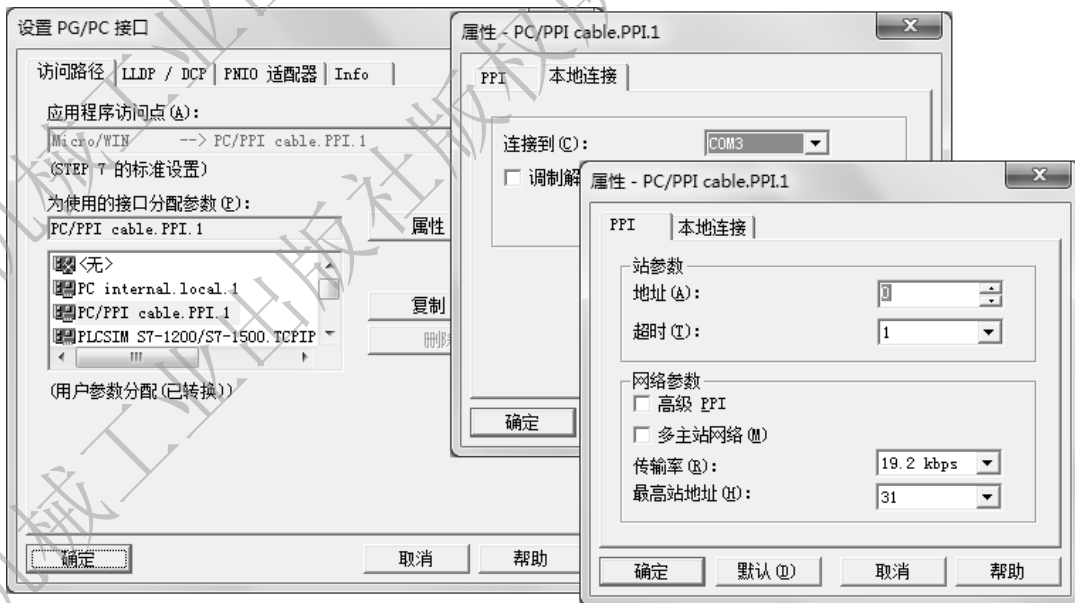


图 2-8 设置 PG/PC 接口

在“PPI”选项卡设置波特率等参数，默认的波特率为 187.5kbit/s。“超时”时间是与通信设备建立联系的最长时间，可采用默认值 1s。最高站地址是 STEP 7-Micro/WIN 停止寻找网络中的其他主站的地址。

对于多主站网络，应设置使用 PPI 协议，并选中“高级 PPI”多选框和“多主站网络”多选框。如果使用 PPI 多主站电缆，可以忽略这两个多选框。在多主站网络中，两台 S7-200 CPU 之间可以用网络读写指令相互读写数据，实现点对点通信。

高级 PPI 功能允许在 PPI 网络中与一个或多个 S7-200 CPU 建立多个连接，S7-200 CPU 的通信口 0 和通信口 1 分别可以建立 4 个连接，EM 277 通信模块可以建立 6 个连接。

也可以使用计算机的通信处理器卡（例如 CP 5511 或 CP 5611）来与 CPU 通信。

5. 用系统块设置 PLC 通信端口的参数

双击指令树文件夹“系统块”中的“通信端口”（见图 2-24），打开系统块的“通信端口”窗口，可以设置 CPU 集成的通信端口的参数，默认的站地址为 2。设置的波特率应与图 2-8 中设置的一致。系统块下载到 PLC 后设置的参数才会起作用。出厂时 CPU 第一个通信端口默认的波特率为 9.6kbit/s，站地址为 2。

6. 建立计算机与 PLC 的在线连接

双击指令树中的“通信”，打开“通信”对话框。在将新的设置下载到 S7-200 之前，应设置远程站（即 S7-200）的地址，CPU 的默认地址为 2。

双击“通信”对话框中的“双击刷新”（见图 2-9），STEP 7-Micro/WIN 将会自动搜索连接在网络上的 S7-200，并用图标显示搜索到的 S7-200。这一步不是建立通信连接必需的操作。不能确定 PLC 端口的波特率时，可以选中“通信”对话框中的多选框“搜索所有波特率”。

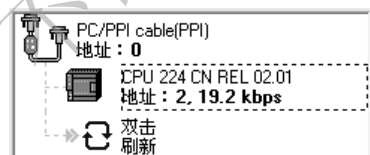


图 2-9 搜索 S7-200 的 CPU

7. 下载程序


计算机与 PLC 建立起通信连接后，单击工具栏上的“下载”按钮，或者执行菜单命令“文件”→“下载”，将会出现下载对话框（见图 2-10）。



图 2-10 “下载”对话框

单击“选项”按钮，可以打开或关闭图 2-10 中该按钮下面的选项区。用户可以用多选

框选择是否下载程序块、数据块、系统块、存储卡中的配方和数据记录配置，打钩表示要下载。单击“下载”按钮，开始下载。

如果设置的 PLC 的型号与 PLC 实际的型号不一致，将会显示提示信息。单击出现的“改动项目”按钮，自动修改 PLC 的型号后，提示信息消失，再进行下载操作。

下载应在 STOP 模式进行，可以设置在下载之前将 CPU 自动切换到 STOP 模式，以及下载结束后自动切换到 RUN 模式是否需要提示。建议最下面的 3 个多选框采用图 2-10 中的设置，这样最为方便快捷。


视频“组态通信与下载程序”可通过扫描二维码 2-6 播放。



二维码 2-6

8. 上载程序

上载前应建立起计算机与 PLC 之间的通信连接，在 STEP 7-Micro/WIN 中新建一个空项目来保存上载的块，项目中原有的内容将被上载的内容覆盖。

单击工具栏上的“上载”按钮，打开上载对话框。上载对话框与下载对话框的结构基本上相同，只是在对话框的右下部分仅有多选框“成功后关闭对话框”。

用户可以用多选框选择是否上载程序块、数据块、系统块、存储卡中的配方和数据记录配置。单击“上载”按钮，开始上载。

9. 运行和调试程序

下载程序后，将 PLC 的工作模式开关拨到 RUN 位置，“RUN”LED 亮，用户程序开始运行。工作模式开关在 RUN 位置时，可以用 STEP 7-Micro/WIN 工具栏上的 RUN 按钮和 STOP 按钮切换 PLC 的操作模式。

在 RUN 模式用接在端子 I0.0~I0.2 上的小开关来模拟按钮提供的起动信号、停止信号和过载信号，将开关接通后马上断开，观察 Q0.0 和 Q0.1 对应的 LED 的状态变化是否正确。

10. PLC 中信息的读取

执行菜单命令“PLC”→“信息...”，将显示出 PLC 的 RUN/STOP 状态、以 ms 为单位的扫描周期、CPU 的型号和固件版本号、错误信息、I/O 模块的配置和状态。“刷新扫描周期”按钮用来读取扫描周期的最新数据。

如果 CPU 配有智能模块，选中要查看的模块，单击“EM 信息...”按钮，将出现一个对话框，显示模块型号、模块版本号、模块错误信息和其他有关的信息。

11. CPU 事件的历史记录

S7-200 保留一份带时间标记的主要 CPU 事件的历史记录，包括上电、进入 RUN 模式、进入 STOP 模式和出现致命错误的时间。应设置实时时钟，这样才能得到事件记录中正确的时间标记。与 PLC 建立通信连接后，执行菜单命令“PLC”→“信息”，在打开的对话框中单击“历史事件”按钮，可以查看 CPU 事件的历史记录。

2.3 符号表与符号地址的使用

1. 打开符号表

为了方便程序的调试和阅读，可以用符号表（见图 2-11）来定义地址或常数的符号。可以为存储器类型 I、Q、M、SM、AI、AQ、V、S、C、T、HC 创建符号名。在符号表中定

义的符号属于全局变量，可以在所有程序组织单元（POU）中使用它们。可以在创建程序之前或之后定义符号。

2. POU 符号表

双击指令树的“符号表”文件夹中的“用户定义 1”，打开符号表。单击符号表窗口下面的“POU 符号”选项卡（见图 2-12），可以看到自动生成的主程序、子程序和中断程序的默认名称，该表格为只读表格，不能用它修改 POU 符号。用右键单击指令树文件夹中的某个 POU，可以用快捷菜单中的“重命名”命令修改它的名称。

	符号	地址	
1	起动按钮	I0.0	常开触点
2	停止按钮	I0.1	常开触点
3	过载	I0.2	常开触点
4	电机1	Q0.0	控制1号电机的接触器
5	电机2	Q0.1	控制2号电机的接触器

图 2-11 “用户定义 1”符号表

	符号	地址	注释
1	SBR_0	SBR0	子程序注释
2	INT_0	INT0	中断程序注释
3	主程序	OB1	程序注释：异步电动机

图 2-12 POU 符号表

3. 使用多个符号表



可以创建多个符号表，但是不同的符号表不能使用同一个符号名和使用同一个地址。右键单击指令树中的“符号表”，执行快捷菜单中的“插入”→“新符号表”命令，将生成新的符号表。成功插入新的符号表后，符号表窗口底部将会出现一个新的选项卡。可以单击这些选项卡来打开不同的符号表。

4. 生成符号


在符号表的“符号”列键入符号名，例如“起动按钮”（见图 2-11），在“地址”列中键入地址或常数。符号名最多可以包含 23 个字符，可以使用英语字母、数字字符、下划线和汉字。可以在“注释”列键入最多 79 个字符的注释。

在为符号指定地址或常数值之前，用绿色波浪下划线表示该符号为未定义符号。在“地址”列键入地址或常数后，绿色波浪下划线消失。

键入时用红色的文本表示下列语法错误：符号以数字开始、使用关键字作符号或使用无效的地址。红色波浪下划线表示用法无效，例如重复的符号名和重复的地址。

符号表用  图标表示地址重叠的符号（例如 VB0 和 VD0）， 图标表示未使用的符号。

5. 表格的通用操作

将鼠标的光标放在表格的列标题分界处，光标出现水平方向的双向箭头  后，按住鼠标的左键，将列分界线拉至所需的位置，可以调节列的宽度。

用鼠标右键单击表格中的某一单元，执行弹出的菜单中的“插入”→“行”命令，可以在所选行的上面插入新的行。将光标置于表格最下面一行的任意单元后，按计算机的 $\langle \downarrow \rangle$ 键，在表格的底部将会增添一个新的行。

按 $\langle \text{Tab} \rangle$ 键光标将移至表格右边的下一个单元格。单击某个单元格，按住 $\langle \text{Shift} \rangle$ 键同时单击另一单元格，将会同时选中两个所选单元格定义的矩形范围内所有的单元格。

单击最左边的行号，可选中整个行。按住鼠标左键在最左边的行号列拖动，可以选中连续的若干行。按删除键可删除选中的行或单元格，可以用剪贴板复制和粘贴选中的对象。

6. 在程序编辑器和状态表中定义、编辑和选择符号

在程序编辑器或状态表中，用鼠标右键单击未连接任何符号的地址，例如 T37。执行出

现的快捷菜单中的“定义符号”命令，可以在打开的对话框中定义符号（见图 2-13）。单击“确认”按钮确认操作并关闭对话框。被定义的符号将同时在程序编辑器或状态表和符号表中出现。

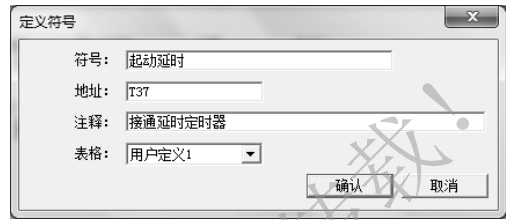




图 2-13 在程序编辑器中创建符号

用鼠标右键单击程序编辑器或状态表中的某个符号，执行快捷菜单中的“编辑符号”命令，可以编辑该符号的地址和注释。用右键单击某个未定义的地址，执行快捷菜单中的“选择符号”命令，出现“选择符号”列表，可以为变量选用表中的符号。

可以在程序中的指令的参数域键入尚未定义的有效符号名。这样生成了一组未分配存储区地址的符号名。单击工具栏上的“建立未赋值符号表”按钮，将这组符号名称传送到新的符号表选项卡，可以在这个新符号表中为符号定义地址。

可以在程序中的指令的参数域键入尚未定义的有效符号名。这样生成了一组未分配存储区地址的符号名。单击工具栏上的“建立未赋值符号表”按钮，将这组符号名称传送到新的符号表选项卡，可以在这个新符号表中为符号定义地址。

7. 符号表的排序

为了方便在符号表中查找符号，可以对符号表中的符号排序。单击“符号”所在的列标题，表中的各行按符号升序排列，即按符号的字母或汉语拼音从 A 到 Z 的顺序排列。再次单击“符号”列标题，表中的各行按符号降序排列。也可以单击地址列的列标题，按地址排序。



二维码 2-7

视频“符号表的操作”可通过扫描二维码 2-7 播放。

8. 切换程序编辑器或状态表中地址的显示方式

执行菜单命令“查看”→“符号寻址”，可以在符号地址和绝对地址显示方式之间切换。图 2-14 是使用符号地址的电机起动控制梯形图中的第一个网络。在符号地址显示方式输入地址时，可以输入符号地址或绝对地址，输入后按设置的显示方式显示地址。

执行菜单命令“工具”→“选项”，选中“选项”对话框左边的“程序编辑器”，在“程序编辑器”选项卡中（见图 2-6），可以用“符号寻址”下拉列表框选择“仅显示符号”或“显示符号和地址”。图 2-15 是同时显示符号地址和绝对地址的例子。

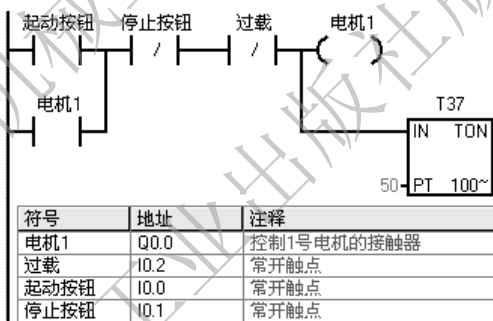


图 2-14 仅显示符号

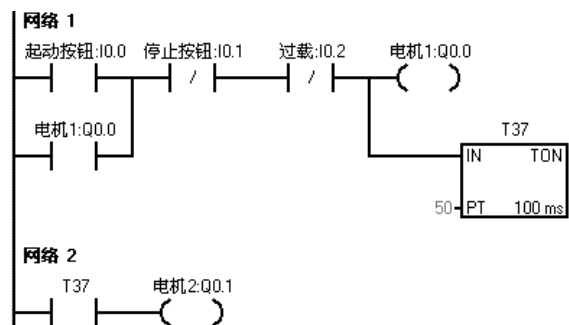



图 2-15 同时显示符号地址和绝对地址


单击工具栏上的“应用项目中的所有符号”按钮，将符号表中定义的所有符号应用到项目，从显示绝对地址切换到显示符号地址。

在程序编辑器或状态表中按〈Ctrl+Y〉键，可以在符号地址和绝对地址显示方式之间切

换。如果符号地址过长，并且选择了显示符号地址或同时显示符号地址和绝对地址，程序编辑器只能显示部分符号名。将鼠标的光标放在这样的符号上，可以在出现的小方框中看到符号地址的全称和绝对地址。

在程序编辑器中引用符号时，可以像绝对地址一样，对符号名使用间接寻址的记号&和*。

9. 符号信息表

单击工具栏上的“切换符号信息表”按钮，可以打开或关闭各网络的符号信息表（见图 2-14）。符号信息表在网络中程序的下面，它列出了网络中使用的符号地址、绝对地址和符号表中的注释。

视频“符号地址的使用”可通过扫描二维码 2-8 播放。




二维码 2-8

2.4 用编程软件监控与调试程序

2.4.1 用程序状态监控与调试程序


1. 启动程序状态监控

在运行 STEP 7-Micro/WIN 的计算机与 PLC 之间建立起通信连接，并将程序下载到 PLC 后，在程序编辑器中打开要监控的 POU，执行菜单命令“调试”→“开始程序状态监控”，或单击工具栏上的“程序状态监控”按钮，就启动了程序状态监控功能。

如果 CPU 中的程序和打开项目的程序可能不同，或者在切换使用的编程语言后启用监控功能，将会出现“时间戳记不匹配”对话框（见图 2-16），单击“比较”按钮，如果经检查确认 PLC 中的程序和打开的项目中的程序相同，对话框中显示“已通过”后，单击“继续”按钮，开始监控。如果 CPU 处于 STOP 模式，将出现对话框询问是否切换到 RUN 模式。如果检查后未通过，应重新下载程序。



图 2-16 “时间戳记不匹配”对话框

单击工具栏上的“暂停程序状态监控”按钮，暂停程序状态监控，当前的数据保留在屏幕上。再次单击该按钮，继续执行程序状态监控。

2. 梯形图程序的程序状态监控

(1) 执行状态的程序状态监控

必须在梯形图程序状态操作开始之前选择程序状态监控的数据采集模式。执行菜单命令“调试”→“使用执行状态”后，进入执行状态，该命令行的前面出现一个“√”。在执行状

态，只是在 PLC 处于 RUN 模式时才更新状态值。不能显示未执行的程序区（例如未调用的子程序、中断程序或被 JMP 指令跳过的区域）的程序状态。

在 RUN 模式启动程序状态功能后，将用颜色显示出梯形图中各元件的状态（见图 2-17），左边的垂直“电源线”和与它相连的水平“导线”变为蓝色。如果位操作数为 1（为 ON），其常开触点和线圈变为蓝色，它们中间出现蓝色方块，有“能流”流过的“导线”也变为蓝色。如果有能流流入方框指令的 EN（使能）输入端，且该指令被成功执行时，方框指令的方框变为蓝色。定时器和计数器的方框为绿色表示它们包含有效数据。红色方框表示执行指令时出现了错误。灰色表示无能流、指令被跳过、未调用，或 PLC 处于 STOP（停止）模式。

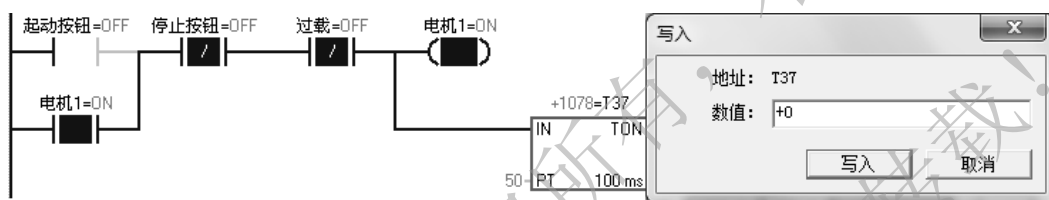


图 2-17 梯形图的程序状态监控

在 RUN 模式启用程序状态监控，将以连续方式采集状态值。“连续”并非意味着实时，而是指编程设备不断地从 PLC 轮询状态信息，并在屏幕上显示，按照通信允许的最快速度更新显示。可能捕获不到某些快速变化的值（例如流过正向、负向检测触点的能流），并在屏幕中显示，或者因为这些值变化太快，无法读取。

开始监控图 2-17 中的梯形图时，各输入点均为 OFF，梯形图中 I0.0 的常开触点断开，I0.1 和 I0.2 的常闭触点接通。用接在端子 I0.0 上的小开关来模拟起动按钮信号，将开关接通后马上断开，梯形图中 Q0.0 的线圈“通电”，T37 开始定时，定时器方框上面 T37 的当前值不断增大。当前值大于等于预置值 50（5s）时，梯形图中 T37 的常开触点接通，Q0.1 的线圈“通电”（见图 2-15）。启用程序状态监控，可以形象直观地看到触点、线圈的状态和定时器当前值的变化情况。

用接在端子 I0.1 上的小开关来模拟停止按钮信号，梯形图中 I0.1 的常闭触点断开后马上接通。Q0.0 和 Q0.1 的线圈断电，T37 被复位。

在 T37 定时的时候，用鼠标右键单击 T37 的当前值，执行出现的快捷菜单中的“写入”命令，可以用出现的对话框（见图 2-17）执行写入操作，来修改 T37 的当前值。

图 2-18 中用定时器 T38 的常闭触点控制它自己的 IN 输入端。进入 RUN 模式时 T38 的常闭触点接通，它开始定时。2s 后定时时间到，T38 的常开触点闭合，使 MB10 加 1；常闭触点断开，使它自己复位，复位后 T38 的当前值变为 0。下一扫描周期因为它的常闭触点接通，使它自己的 IN 输入端重新“得电”，又开始定时。T38 将这样周而复始地工作。从上面的分析可知，图 2-18 上面的网络是一个脉冲信号发生器，脉冲周期等于 T38 的预设值 2s。T38 的当前值按图 2-22 中的锯齿波形不断变化。

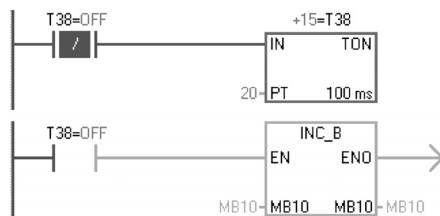



图 2-18 梯形图的程序状态监控

单击工具栏上的“暂停状态开/关”按钮，暂停程序状态的采集，T38 的当前值停止变化。再次单击该按钮，T38 的当前值重新开始变化。

为了节省篇幅，本书一般省略了梯形图中的网络编号，用左侧垂直母线上的断点表示相邻网络的分界点。


(2) 扫描结束状态的状态监控

在上述的执行状态时执行菜单命令“调试”→“使用执行状态”，菜单中该命令行前面的“√”消失，进入扫描结束状态。

“扫描结束”状态显示在程序扫描结束时读取的状态结果。这些结果可能不会反映 PLC 数据地址的所有数值变化，因为随后的程序指令在程序扫描结束之前可能会写入和重新写入数值。由于快速的 PLC 扫描周期和相对慢速的 PLC 状态数据通信之间存在的速度差别，“扫描结束”状态显示的是几个扫描周期结束时采集的数据值。

只在 RUN 模式才会显示触点和线圈中的颜色块，以区别 RUN 和 STOP 模式。

3. 语句表程序的程序状态监控

在语句表显示方式单击“程序状态监控”按钮，启动语句表的程序状态监控功能。程序编辑器窗口分为代码区和用蓝色字符显示数据的状态区。图 2-19 中操作数 3 的右边是逻辑堆栈中的值。最右边的列是方框指令的使能输出位（ENO）的状态。

网络 1

		操作数 1	操作数 2	操作数 3	0123	中
LD	起动手按钮	OFF			0000	0
O	电机1	ON			1000	1
AN	停止按钮	OFF			1000	1
AN	过载	OFF			1000	1
=	电机1	ON			1000	1
TON	T37, 50	+350	50		1000	1

图 2-19 语句表的程序状态监控

在图 2-19 中的“操作数 1”列可以看到 T37 的当前值不断变化。用接在端子 I0.0 和 I0.1 上的小开关来模拟按钮信号，可以看到指令中的位地址的 ON/OFF 状态的变化和定时器的当前值的变化。

用菜单命令“工具”→“选项”打开“选项”对话框，在“程序编辑器”的“STL 状态监控”选项卡（见图 2-20），可以设置语句表程序状态监控的内容，每条指令最多可以监控 17 个操作数、逻辑堆栈中 4 个当前值和 1 个指令状态位。



图 2-20 语句表程序状态监控的设置

状态信息从位于编辑窗口顶端的第一条 STL 语句开始显示。用滚动条向下滚动显示编辑器窗口时，将从 CPU 获取新的信息。

视频“用程序状态监控程序”可通过扫描二维码 2-9 播放。



二维码 2-9

2.4.2 用状态表监控与调试程序

如果需要同时监控的变量不能在程序编辑器中同时显示，可以使用状态表监控功能。

1. 打开和编辑状态表

在程序运行时，可以用状态表来读、写、强制和监控 PLC 中的变量。双击指令树的“状态表”文件夹中的“用户定义 1”，或者执行菜单命令“查看”→“组件”→“状态表”，均可以打开状态表，并对它进行编辑。如果项目中有多个状态表，可以用状态表编辑器底部的选项卡切换它们。

在状态表的“地址”列键入要监控的变量的绝对地址或符号地址（见图 2-21），可以采用默认的显示格式，或者用“格式”列隐藏的下拉式列表更改显示格式。定时器和计数器可以分别按位或按字监控。按位监控显示的是它们的输出位的 ON/OFF 状态，按字监控显示的是它们的当前值。

	地址	格式	当前值	新值
1	I0.0	位	2#0	
2	I0.1	位	2#0	
3	Iw0	二进制	2#0000_0000_0000_0000	
4	Q0.0	位	2#1	
5	T37	位	2#1	
6	T37	有符号	+54	+0
7	T38	有符号	+14	
8	M10.0	位	2#1	

图 2-21 状态表

选中符号表中的符号单元或地址单元，并将其复制到状态表的“地址”列，可以快速创建要监控的变量。

单击状态表某个“地址”列的单元格（例如 VW20）后按〈ENTER〉键，可以在下一行插入或添加一个具有顺序地址（例如 VW22）和相同显示格式的新的行。

执行菜单命令“编辑”→“插入”→“行”，或者用鼠标右键单击状态表中的单元，执行弹出的菜单中的“插入”→“行”命令，可以在状态表中当前光标位置的上面插入新的行。将光标置于状态表最下面一行的任意单元，按计算机向下的箭头键，在状态表的底部将会增添一个新的行。


2. 创建新的状态表


可以根据不同的监控任务，创建几个状态表。用鼠标右键单击指令树中的“状态表”，或右键单击已经打开的状态表，执行弹出的菜单中的“插入”→“状态表”命令，可以创建新的状态表。

3. 通过一段程序代码构建状态表

选中若干个连续的网络，用鼠标右键单击选中的网络，执行出现的菜单中的“创建状态表”命令，选中区域的每个操作数是新的状态表中的一个条目，每次只能添加前 150 个地址。创建后可以编辑表中的条目。


4. 启动和关闭状态表的监控功能

与 PLC 的通信连接成功后，打开状态表，执行菜单命令“调试”→“开始状态表监控”，或单击工具栏上的“状态表监控”按钮（见图 2-23），该按钮被“按下”，将会启动状态表的监控功能。STEP 7-Micro/WIN 从 PLC 收集状态信息，在状态表的“当前值”列出现从 PLC 中读取的动态数据。这时还可以强制修改状态表中的变量。


启动监控后用接在输入端子上的小开关来模拟起动按钮和停止按钮信号，可以看到各个位地址的 ON/OFF 状态和定时器当前值变化的情况。执行菜单命令“调试”→“停止状态表监控”或单击“状态表监控”按钮，可以关闭状态表的监控功能。

用二进制格式监控字节、字或双字（见图 2-21 中对 IWO 的监控），可以在一行中同时监控 8 点、16 点或 32 点位变量。

5. 单次读取状态信息

状态表的监控功能被关闭时，或 PLC 切换到 STOP 模式，执行菜单命令“调试”→“单次读取”或单击工具栏上的“单次读取”按钮，可以从 PLC 收集当前的数据，并在状态表的“当前值”列显示出来。

6. 趋势图

趋势图（见图 2-22）用随时间变化的曲线跟踪 PLC 的状态数据。启动状态表监控功能后，单击工具栏上的趋势图按钮，可以在表格视图与趋势图之间切换。图 2-18 中定时器 T38 的当前值按图 2-22 所示的锯齿波变化。T38 的常开触点每 2s 产生一个脉冲，将字节 MB10 的值加 1。MB10 的最低位 M10.0 的 ON/OFF 状态以 4s 的周期变化。

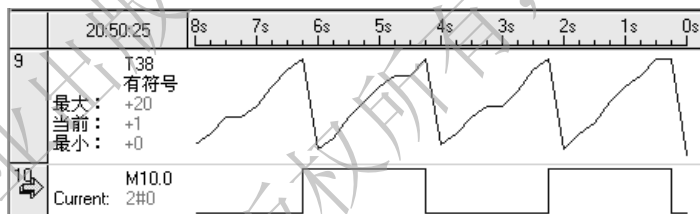




图 2-22 趋势图

用鼠标右键单击趋势图，执行弹出菜单中的命令，可以修改趋势图的时间基准（即时间轴的刻度）。如果更改趋势图的时间基准（0.25s~5min），整个图的数据都会被清除，并用新的时间基准重新显示。执行弹出菜单中的“属性”命令，在弹出的对话框中，可以修改被单击的行变量的地址和显示格式，以及显示的上限和下限。

启动趋势图后单击工具栏上的“暂停趋势图”按钮，或执行菜单命令“调试”→“暂停趋势图”，可以“冻结”趋势图。再次单击该按钮将结束暂停。

实时趋势功能不支持历史趋势，即不会保留超出趋势图窗口的时间范围的趋势数据。

将光标放到两行的分界线上，光标变为垂直方向的双向箭头，按住鼠标左键上、下移动鼠标，拖动水平分界线，可以调整上面一行的高度。

视频“用状态表监控程序”可通过扫描二维码 2-10 播放。




二维码 2-10

2.4.3 写人与强制数据

本节介绍用程序编辑器和状态表将新的值写入或强制给操作数的方法。

1. 写入数据

“写入”功能允许将一个或多个数值写入 PLC 中的变量。将变量新的值键入状态表的“新值”列后（见图 2-21），单击工具栏上的“全部写入”按钮，将“新值”列所有的值传送到 PLC。在 RUN 模式时因为用户程序的执行，修改的数值可能很快被程序改写成新的数值，不能用写入功能改写物理输入点（地址 I 或 AI）的状态。

在程序状态监控时，用鼠标右键单击梯形图中的某个地址或语句表中的某个操作数的值，可以用快捷菜单中的“写入”命令和出现的“写入”对话框来完成写入操作（见图 2-17）。



二维码 2-11

2. 强制的基本概念

强制（Force）功能通过强制 V、M 来模拟逻辑条件，通过强制 I/O 点来模拟物理条件。例如可以通过对输入点的强制代替输入端外接的小开关，来调试程序。

可以强制所有的 I/O 点，此外还可以同时强制最多 16 个 V、M、AI 或 AQ 地址。强制功能可以用于 I、Q、V、M 的字节、字和双字，只能从偶数字节开始以字为单位强制 AI 和 AQ。不能强制 I 和 Q 之外的位地址。强制的数据用 CPU 的 EEPROM 永久性地存储。

在 RUN 模式且对控制过程影响较小的情况下，可以对程序中的某些变量强制性地赋值。

在读取输入阶段，强制值被当作输入读入；在程序执行阶段，强制数据用于立即读和立即写指令指定的 I/O 点。在通信处理阶段，强制值用于通信的读/写请求；在修改输出阶段，强制数据被当作输出写到输出电路。进入 STOP 模式时，输出将变为强制值，而不是系统块中设置的值。虽然在一次扫描过程中，程序可以修改被强制的数据，但是新扫描开始时，会重新应用强制值。

在写入或强制输出时，如果 S7-200 与其他设备相连，可能导致系统出现无法预料的情况，引起人员伤亡或设备损坏，只有合格的人员才能进行强制操作。强制程序值后，务必通知所有有权维修或调试过程的人员。

3. 强制的操作方法



启动状态表的监控功能后，可以用“调试”菜单中的命令或工具栏上与调试有关的按钮（见图 2-23）执行下述的操作。用鼠标右键单击状态表中的某个操作数，从弹出的菜单中可以选择对该操作数强制或取消强制。



	地址	格式	当前值
10	VW0	十六进制	 16#1234
11	VB0	十六进制	 16#12
12	VW1	十六进制	 16#3400

图 2-23 用状态表强制变量

(1) 强制

将要强制的新的值 16#1234 键入状态表中 VW0 的“新值”列，单击工具栏上的“强制”按钮，VW0 被强制为新的值。在“当前值”列的左端出现强制图标（见图 2-23）。

要强制程序状态中的某个地址，可以用鼠标右键单击它，执行快捷菜单中的“强制”命令，然后用出现的“强制”对话框进行强制操作。

一旦使用了强制功能，每次扫描都会将强制的数值用于该操作数，直到取消对它的强制。即使关闭 STEP 7-Micro/WIN，或者断开 S7-200 的电源，都不能取消强制。





黄色的显式强制图标（一把合上的锁）表示该地址被显式强制，对它取消强制之前用其他方法不能改变此地址的值。

图 2-23 中的 VW0 被显式强制，VB0 是 VW0 的一部分，因此它被隐式强制。灰色的隐式强制图标 （合上的灰色的锁）表示该地址被隐式强制。


VW0 被显式强制，因为 VW1 的第一个字节 VB1 是 VW0 的第二个字节，VW1 的一部分也被强制，因此 VW1 被部分隐式强制。灰色的部分隐式强制图标 （半把灰色的锁）表示该地址被部分隐式强制。

不能直接取消对 VB0 的隐式强制和对 VW1 的部分隐式强制，必须取消对 VW0 的显式强制，才能同时取消上述的隐式强制和部分隐式强制。


(2) 取消对单个操作数的强制

选择一个被显式强制的操作数，然后单击工具栏上的“取消强制”按钮 ，被选择的地址的强制图标将会消失。也可以用鼠标右键单击程序状态中被强制的地址，用快捷菜单中的命令取消对它的强制。

(3) 取消全部强制

单击工具栏上的“取消全部强制”按钮 ，可以取消对被强制的全部地址的强制，使用该功能之前不必选中某个地址。

(4) 读取全部强制

单击工具栏上的“读取全部强制”按钮 ，状态表中的当前值将会为已被显式强制，隐式强制和部分隐式强制的所有地址显示出相应的强制图标。



视频“用编程软件强制数据”可通过扫描二维码 2-12 播放。





二维码 2-12

4. 在 STOP 模式下写入和强制输出

如果在写入或强制输出点 Q 时 S7-200 已连接到设备，这些更改将会传送到该设备。这可能导致设备出现异常，从而造成人员伤亡或设备损坏。必须执行菜单命令“调试”→“STOP（停止）模式下写入-强制输出”，才能在 STOP 模式下启用该功能。打开 STEP 7-Micro/WIN 或打开不同的项目时，作为默认状态，没有选中该菜单选项，以防止在 PLC 处于 STOP 模式时写入或强制输出。

2.4.4 调试用户程序的其他方法

1. 使用书签

工具栏上的“切换书签”按钮  用于在当前光标位置指定的网络设置或删除书签。单击  或  按钮，光标将移动到程序中下一个或上一个标有书签的网络。单击  按钮将删除程序中所有的书签。

2. 单次扫描

从 STOP 模式进入 RUN 模式，首次扫描位 (SM0.1) 在第一次扫描时为 ON。由于执行速度太快，在程序运行状态观察不到首次扫描刚结束时某些编程元件的状态。

在 STOP 模式执行菜单命令“调试”→“首次扫描”，PLC 进入 RUN 模式，执行一次扫描后，自动回到 STOP 模式，可以观察到首次扫描后的状态。

3. 多次扫描

PLC 处于 STOP 模式时，执行菜单命令“调试”→“多次扫描”，在出现的对话框中指定执行程序扫描的次数 (1~65535 次)。单击“确认”按钮，执行完指定的扫描次数后，自动返回 STOP 模式。

4. 在 RUN 模式下编辑用户程序

在 RUN（运行）模式下，不必转换到 STOP（停止）模式，便可以对程序做较小的改动，并将改动下载到 PLC。

建立好计算机与 PLC 之间的通信连接后，在 RUN 模式执行菜单命令“调试”→“RUN（运行）模式下程序编辑”，出现“上载”对话框和警告信息。单击“上载”按钮，程序被上载。上载结束后，进入 RUN 模式编辑状态，出现一个跟随鼠标移动的 PLC 图标。

再次执行菜单命令“调试”→“RUN（运行）模式下程序编辑”，将退出 RUN 模式编辑。

编辑前应退出程序状态监控，修改程序后，需要将改动下载到 PLC。下载之前一定要仔细考虑可能对设备或操作人员造成的各种安全后果。

在 RUN 模式编辑状态下修改程序后，CPU 对修改的处理方法可以查阅系统手册。

在 RUN 模式编辑过程中，STEP 7-Micro/WIN 在正向、负向转换触点上面为 EU/ED 指令分配一个临时的编号。同时交叉引用表中出现“边沿使用”选项卡，列出程序中所有的 EU/ED 指令，P 和 N 分别表示指令 EU 指令和 ED 指令。修改程序时可以参考该表，禁止使用编号重复的 EU/ED 指令。

2.5 使用系统块设置 PLC 的参数

执行菜单命令“查看”→“组件”→“系统块”，可以打开系统块。单击指令树的“系统块”文件夹中的某个对象（见图 2-24），可以直接打开系统块中对应的对话框。

2.5.1 断电数据保持的设置与编程

1. S7-200 保存数据的方法

S7-200 CPU 中的数据存储区分为易失性的 RAM 存储区，以及不需要供电就可以永久保存数据的 EEPROM 存储区。前者的电源消失后，存储的数据将会丢失。后者的电源消失后，存储的数据不会丢失。CPU 在工作时，V、M、T、C、Q 等存储区的数据都保存在 RAM 中。

S7-200 用内置的 EEPROM 永久保存程序块、数据块、系统块、强制值、组态为断电保持的存储区，以及在用户程序控制下写入 EEPROM 的指定值。配方和数据记录组态用存储卡保存。

从 CPU 模块上载用户程序时，CPU 将从 EEPROM 上载程序块、数据块和系统块，同时从存储卡上载配方和数据记录组态。通过 S7-200 的资源管理器上载数据记录中的数据。S7-200 提供了多种方法来保存数据。

1) 用 CPU 的超级电容器保存 RAM 中的 V、M、T、C 存储区的数据。超级电容器可以保持几天，保持的时间（50h 或 100h）与 CPU 模块的型号有关。CPU 上电后超级电容开始充电，至少充电 24h 后才能获得表 1-1 中的保持时间。

2) 可选的电池卡可以延长 RAM 保持信息的时间，只是在超级电容器电能耗尽后电池

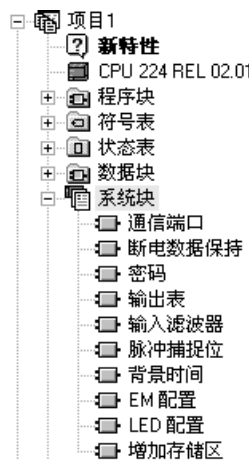


图 2-24 指令树

卡才提供电源。在 PLC 连续断电时，电池的寿命约 200 天。

3) MB0~MB13 如果在系统块中被设置为断电保持，在 CPU 模块断电时被永久保存在 EEPROM 中。

4) 数据块用来给 V 存储区赋初值，数据块下载到 CPU 后，存储在 EEPROM 中，所以可以用数据块来保存程序中用到的不需要改变的数据，例如已调试好的 PID 参数。

5) 用可拆卸的存储卡（EEPROM 卡）来保存程序块、数据块、系统块、配方、数据记录和强制值。通过 S7-200 的资源管理器，可以将文件储存在存储卡中。

静电放电可能损坏存储卡或 CPU 接口，取存储卡时应使用接地垫或戴接地手套，应将存储卡存放在导电的容器中。

2. 设置 PLC 断电后的数据保存方式

双击图 2-24 所示指令树的“系统块”文件夹中的“断电数据保持”，打开“系统块”对话框（见图 2-25）。选择从通电切换到断电时希望保存的存储区。



图 2-25 设置断电数据保持

最多可以定义 6 个在电源掉电时需要保持的存储区范围，图 2-25 中是默认的保持范围。可以设置保存的存储区有 V、M、T 和 C。只能保持 TONR（有记忆接通延时定时器）和计数器的当前值，不能保持定时器位和计数器位，上电时它们被清除。单击“默认值”按钮，将采用 STEP 7-Micro/WIN 推荐的设置值。

单击“确认”按钮确认设置的参数，并自动关闭系统块。将系统块下载到 PLC 后，设置的参数才起作用。

3. 开机后数据的恢复

上电后，CPU 会自动地从 EEPROM 中恢复程序块和系统块，然后检查是否安装了超级电容器和可选的电池卡。如果是，将确认数据是否成功地保存到 RAM。如果保存是成功的，RAM 存储器的保持区将保持不变。EEPROM 中数据块的内容被复制到 V 存储器的非保持区，RAM 中的其他非保持区被清零。

如果 RAM 存储器中的数据没有保持下来（例如长时间断电后），CPU 会清除 RAM 中所有的用户存储区，并在通电后的第一次扫描置“保持数据丢失”标志（SM0.2）为 ON。

此外，EEPROM 中的 V 存储区和 M 存储区的永久区域从 EEPROM 复制到 RAM 中，RAM 的所有其他区域均被清零。

4. 用程序将 V 存储器的数据复制到 EEPROM

可以将 V 存储区任意位置的数据（字节、字和双字）复制到 EEPROM 中。一次写 EEPROM 的操作会使扫描周期增加 10~15ms。新存入的值会覆盖 EEPROM 中原有的数据，写 EEPROM 的操作不会更新存储卡中的数据。

将 V 存储器中的一个数据复制到 EEPROM 中的 V 存储区的步骤如下：

1) 将需要保存的 V 存储器的地址送特殊存储器字 SMW32 中。

2) 数据长度单位写入 SM31.0 和 SM31.1，这两位为二进制数 00 和 01 时表示字节，为 10 时表示字，为 11 时表示双字。

3) 令 SM31.7 = 1，在每次扫描结束时，CPU 自动检查 SM31.7，该位为 1 时将指定的数据存入 EEPROM，CPU 将该位复位为 0 后操作结束。

【例 2-1】在 I0.0 的上升沿将 VW50 的值写入 EEPROM。写入 SMB31 的 16#82 (2#1000 0010) 的最低两位为 2#10，表示要写入字，最高位（写入标志）为 1。

```
LD      I0.0
EU
MOVW   50, SMW32      //指定 V 存储器的地址
MOVB   16#82, SMB31  //令 SM31.7 = 1，将 VW50 的值写入 EEPROM
```

写入 EEPROM 的操作次数是有限制的，最少 10 万次，典型值为 100 万次。应仅在发生特殊事件时才将数据保存到 EEPROM，否则可能会因为写入次数过多使 EEPROM 失效。

例如假设扫描周期为 50ms，在每个扫描周期保存一次某个地址的值，5000s（不到 1.5h）就写入了 10 万次，EEPROM 可能就会失效了。

将写入 EEPROM 的 V 存储区设置为没有数据保持功能，在 CPU 断电又上电后，如果该 V 存储器地址中是写入 EEPROM 的数据，则说明数据已经写入 EEPROM。

2.5.2 创建与使用密码

1. 密码的作用

S7-200 的密码保护功能用于限制对特殊功能的访问。有 4 种限制 CPU 访问功能的等级（见表 2-1）。各等级均有不需要密码的访问功能，默认的 1 级没有限制。如果设置了密码，只有输入正确的密码后，S7-200 才根据授权级别提供相应的操作功能。系统块下载到 CPU 后，密码才起作用。

表 2-1 不同授权级别的访问限制

任 务	1 级	2 级	3 级	4 级
读写用户数据；启动、停止 CPU，上电复位；读写时钟	允许	允许	允许	允许
上载用户程序、数据块和系统块				不允许
下载或删除程序块、数据块和系统块；运行时编辑；将程序块、数据块或系统块复制到存储卡		有限制	有限制	有限制
在状态表中强制数据；执行单次/多次扫描；刷新 PLC 信息中的扫描周期；在 STOP 模式写输出				
执行状态监控，项目比较				不允许

在第 4 级密码的保护下，即使有正确的密码也不能上载程序，不允许下载和删除系统块。允许一个用户使用授权的 CPU 功能就会禁止其他用户使用该功能。在同一时刻，只允许一个用户不受限制地存取。

2. 密码的设置

单击图 2-25 左边窗口中的“密码”，选择限制级别为 2~4 级（见图 2-26），在“密码”和“验证”文本框输入相同的密码，密码最多 8 位，字母不区分大小写。

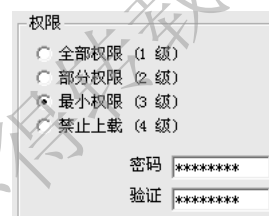


图 2-26 设置密码

3. 忘记密码的处理

如果忘记了密码，必须清除存储器，重新下载程序。清除存储器会使 CPU 进入 STOP 模式，并将网络地址、波特率和时钟之外的其他参数恢复到出厂设置。

计算机与 PLC 建立连接后，执行菜单命令“PLC”→“清除”，显示出“清除”对话框后，选择要清除的块，单击“清除”按钮。如果设置了密码，会显示一个密码授权对话框。在对话框中输入“CLEARPLC”（不区分大小写），确认后执行清除存储器的操作。

清除 CPU 的存储器将关闭所有的数字量输出，模拟量输出将处于某一固定的值。如果 PLC 与其他设备相连，应注意输出的变化是否会影响设备和人身安全。

4. POU 和项目文件的加密

(1) POU 的加密

POU（程序组织单元）包括主程序、子程序和中断服务程序。可以对某个 POU 单独加密。

用鼠标右键单击指令树中要加密的 POU，执行弹出的快捷菜单中的“属性”命令，打开该 POU 的“属性”对话框中的“保护”选项卡。选中多选框“用密码保护本 POU”，在“密码”和“验证”文本框中输入相同的 4 位密码。单击“确认”按钮，退出对话框。

在指令树中被加密的 POU 的图标上和被加密的 POU 中，出现一把锁的图标。程序下载到 CPU 后再上传，也保持加密状态。如果选中多选框“用此密码保护所有 POU”，所有的 POU 都将被设置相同的密码。

(2) 打开被加密的 POU

用鼠标右键单击指令树中被加密的 POU，执行弹出的快捷菜单中的“属性”命令。选中出现的对话框的“保护”选项卡。在“密码”文本框输入正确的密码，然后单击“验证”按钮，就可以打开 POU，查看其中的内容。

即使不知道密码，也可以调用已加密的 POU。虽然看不到程序的内容，在程序编辑器中可以查看其局部变量表中变量的符号名、数据类型和注释等信息。

(3) 项目文件的加密

用鼠标右键单击指令树中的项目，执行“设置密码”指令，用出现的“设置项目密码”对话框对整个项目文件加密。项目文件密码最多 16 个字符，可以是字母或数字的组合，字母区分大小写。需要输入密码才能打开项目文件。

2.5.3 组态输入输出参数

1. 输出表的设置

单击图 2-25 系统块左边窗口中的“输出表”，可以设置从 RUN 模式切换到 STOP 模式后，各输出点的状态。

(1) 数字量输出表的设置

在输出表的“数字量”选项卡中，选中“将输出冻结在最后的最后的状态”多选框，从 RUN 模式变为 STOP 模式时，所有的数字量输出点将保持在 CPU 由 RUN 模式进入 STOP 模式时的状态。

如果未选“冻结”模式，从 RUN 模式变为 STOP 模式时各输出点的状态用输出表来设置。希望进入 STOP 模式之后某一输出位为 ON，则单击该位对应的小方框，使之显示出“√”，输出表的默认值是未选“冻结”模式，并且从 RUN 模式切换到 STOP 模式时，所有输出点的状态被置为 OFF。应按确保系统安全的原则来设置输出表。

(2) 模拟量输出表的设置

输出表的“模拟量”选项卡中的“将输出冻结在最后的最后的状态”选项的意义与数字量输出的相同。如果未选“冻结”模式，可以设置从 RUN 模式变为 STOP 模式后模拟量输出的值（-32768~32767）。

2. 数字量输入滤波器的设置

输入滤波器用来滤除输入线上的干扰噪声，例如机械触点闭合或断开时产生的抖动，以及模拟量输入信号中的脉冲干扰信号。单击图 2-25 左边窗口中的“输入滤波器”，在打开的对话框的“数字量”选项卡中，可以设置 4 点为 1 组的 CPU 的输入点的输入滤波器延迟时间。输入状态发生 ON/OFF 变化时，输入信号必须在设置的延迟时间内保持新的状态，才能被认为有效。延迟时间的设置范围为 0.2~12.8ms，默认值为 6.4ms。为了消除触点抖动的的影响，可选 12.8ms。数字量输入滤波器会影响输入中断和脉冲捕获功能，但是高速计数器不会受它的影响。

3. 模拟量输入滤波器的设置

单击图 2-25 左边窗口中的“输入滤波器”，在打开的对话框的“模拟量”选项卡中，可以设置每个模拟量输入通道是否采用软件滤波。滤波后的值是预选的采样次数的各次模拟量输入的平均值。采样次数多将使滤波后的值稳定，但是响应较慢，采样次数少滤波效果较差，但是响应较快。

滤波器的设定值（采样次数与死区）对所有被选择为有滤波功能的模拟量输入均是一样的。如果信号变化很快，不应使用模拟量滤波。

模拟量输入滤波的默认设置是对所有的模拟量输入滤波（打钩）。取消打钩可以关闭某些模拟输入点的滤波功能。对于没有选择输入滤波的通道，当程序访问模拟量输入时，直接从扩展模块读取模拟值。

CPU 224XP 内置的 AIW0 和 AIW2 模拟量输入在每次扫描都会从 A-D 转换器读取最新的转换结果。该转换器由 A-D 转换器滤波，因此通常无需软件滤波。

滤波器具有快速响应的特点，可以反映信号的快速变化。当输入值与平均值之差超过设置的死区值时，滤波器相对上一次模拟量输入值产生一个阶跃变化。死区值用模拟量输入的数字值来表示。

模拟量滤波功能不能用于以模拟量字传递数字信息或报警指示的模块。不能对 AS-i 主站模块、热电偶模块和热电阻模块使用模拟量输入滤波。

4. 脉冲捕捉功能的设置

因为在每一个扫描周期开始时读取数字量输入，CPU 可能发现不了脉冲宽度小于扫描周期的脉冲（见图 2-27）。脉冲捕捉（Pulse Catch）功能用来捕捉持续时间很短的高电平脉冲或低电平脉冲。

S7-200 的 CPU 模块内置的每个数字量输入点均可以设置为有脉冲捕捉功能。默认的设置是禁止所有的输入点捕捉脉冲。某个输入点启动了脉冲捕捉功能后（多选框打钩），实际输入状态的变化被锁存并保存到下一次读取输入（见图 2-27）。脉冲捕捉功能在输入滤波器之后（见图 2-28），使用脉冲捕捉功能时，必须同时调节输入滤波时间，使窄脉冲不会被输入滤波器过滤掉。

视频“设置输入输出参数”可通过扫描二维码 2-13 播放。



二维码 2-13

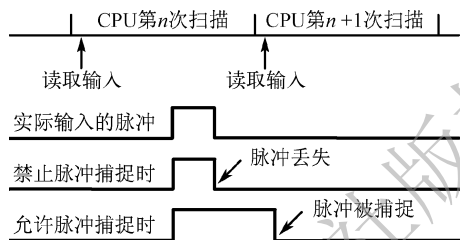


图 2-27 脉冲捕捉

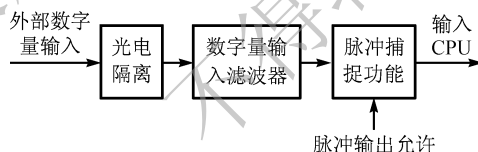


图 2-28 数字量输入电路

一个扫描周期内如果有多个输入脉冲，只能检测出第一个脉冲。如果希望在一个扫描周期内检测出多个脉冲，应使用上升沿/下降沿中断事件（见 4.6 节）。

5. 后台通信时间的设置

单击指令树“系统块”文件夹中的“背景时间”（见图 2-24），可以设置处理运行模式编辑或执行状态监控有关的通信处理所占的时间与扫描周期的百分比，默认值为 10%，最大值为 50%。增大该百分比将增大扫描周期，使控制过程变慢。

2.6 习题

1. 交叉引用表有什么作用？
2. 怎样获得在线帮助？
3. 在梯形图中怎样划分网络？
4. 怎样修改梯形图中网格的宽度？
5. 使用国产的 USB/PPI 来下载程序需要做哪些操作？
6. 怎样修改 CPU 的 RS-485 端口的波特率？
7. 怎样切换 CPU 的工作模式？
8. 怎样在程序编辑器中定义或编辑符号？
9. 怎样更改程序编辑器中地址的显示方式？

10. 程序状态监控有什么优点？什么情况应使用状态表？
11. 写入和强制数据有什么区别？怎样在程序编辑器中写入或强制数据？
12. 怎样长期保存某些 V 存储区中的数据？
13. 希望在 CPU 进入 STOP 模式后保持各数字量输出点的状态不变，应怎样设置？
14. 怎样用系统块设置密码？怎样取消密码？
15. 怎样消除触点抖动的不良影响？
16. 脉冲捕捉功能有什么作用？哪些输入点有脉冲捕捉功能？

机械工业出版社版权所有，不得转载！

机械工业出版社版权所有，不得转载！

第 3 章 S7-200 编程基础

3.1 PLC 的编程语言与程序结构

3.1.1 PLC 编程语言的国际标准

与个人计算机相比，PLC 的硬件、软件的体系结构都是封闭的而不是开放的。各厂家的 PLC 的编程语言、指令的设置和指令的表达方式也不一致，互不兼容。IEC（国际电工委员会）是为电子技术的所有领域制定全球标准的世界性组织。IEC 于 1994 年 5 月公布了 PLC 标准（IEC 61131），其中的第三部分（IEC 61131-3）是 PLC 的编程语言标准。IEC 61131-3 是世界上第一个，也是至今为止唯一的工业控制系统的编程语言标准。

目前已有越来越多的生产 PLC 的厂家提供符合 IEC 61131-3 标准的产品，IEC 61131-3 已经成为各种工控产品事实上的软件标准。

IEC 61131-3 详细地说明了句法、语义和下述 5 种编程语言（见图 3-1）。

- 1) 顺序功能图（Sequential Function Chart, SFC）。
- 2) 梯形图（Ladder Diagram, LD）。
- 3) 功能块图（Function Block Diagram, FBD）。
- 4) 指令表（Instruction List, IL）。
- 5) 结构文本（Structured Text, ST）。

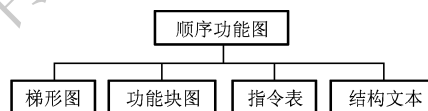


图 3-1 PLC 的编程语言

顺序功能图、梯形图和功能块图是图形编程语言，指令表和结构文本是文字语言。

1. 顺序功能图

这是一种位于其他编程语言之上的图形语言，用来编制顺序控制程序。顺序功能图提供了一种组织程序的图形方法，第 5 章将详细地介绍顺序功能图的使用方法。

2. 梯形图

梯形图是使用得最多的 PLC 图形编程语言。梯形图与继电器控制系统的电路图很相似，具有直观易懂的优点，很容易被工厂熟悉继电器控制的电气人员掌握，特别适用于数字量逻辑控制。有时把梯形图称为电路。

梯形图由触点、线圈和用方框表示的指令组成。触点代表逻辑输入条件，例如外部的开关、按钮和内部条件等。线圈通常代表逻辑输出结果，用来控制外部的指示灯、交流接触器和内部的标志位等。方框用来表示定时器、计数器或者数学运算等指令。

在分析梯形图中的逻辑关系时，为了借用继电器电路图的分析方法，可以想象左右两侧垂直“电源线”之间有一个左正右负的直流电源电压，S7-200 的梯形图（见图 3-2）省略了右侧的垂直电源线。当 I0.0 与 I0.1 的触点接通，或者 Q0.0 与 I0.1 的触点接通时，有一个假

想的“能流”(Power Flow)流过 Q0.0 的线圈。利用能流这一概念,可以帮助我们更好地理解和分析梯形图,能流只能从左向右流动。

梯形图程序被划分为若干个网络(Network),程序中有网络编号,允许以网络为单位,给梯形图加注释。本书为了节省篇幅,一般省略了网络号。一个网络只能有一块不能分开的独立电路。在网络中,逻辑运算按从左到右的方向执行,与能流的方向一致。没有跳转指令时,各网络按从上到下的顺序执行,执行完所有的网络后,下一个扫描周期返回最上面的网络,重新开始执行程序。

3. 语句表

S7 系列 PLC 将指令表称为语句表,简称为 STL。语句表程序由指令组成,PLC 的指令是一种与微机的汇编语言中的指令相似的助记符表达式。图 3-3 是图 3-2 对应的语句表。语句表比较适合熟悉 PLC 和程序设计的经验丰富的程序员使用。

4. 功能块图

这是一种类似于数字逻辑电路的编程语言,有数字电路基础的人很容易掌握。功能块图用类似与门、或门的方框来表示逻辑运算关系,方框的左侧为逻辑运算的输入变量,右侧为输出变量,输入、输出端的小圆圈表示“非”运算,方框被“导线”连接在一起,信号从左向右流动。图 3-4 中的控制逻辑与图 3-2 中的相同。

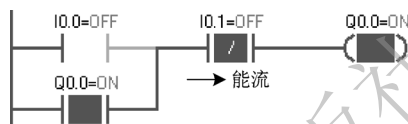


图 3-2 梯形图

LD	I0.0
O	Q0.0
AN	I0.1
=	Q0.0

图 3-3 语句表



图 3-4 功能块图

5. 结构文本

结构文本是为 IEC 61131-3 标准创建的一种专用的高级编程语言。与梯形图相比,它能实现复杂的数学运算,编写的程序非常简洁和紧凑。

6. 编程语言的相互转换和选用

在 STEP 7-Micro/WIN 中,用户可以切换编程语言,选用梯形图、功能块图和语句表来编程。国内很少有人使用功能块图语言。

梯形图与继电器电路图的表达方式极为相似,梯形图中输入信号(触点)与输出信号(线圈)之间的逻辑关系一目了然,易于理解。语句表程序较难阅读,其中的逻辑关系很难一眼看出。在设计复杂的数字量控制程序时建议使用梯形图语言。但是语句表程序输入方便快捷,还可以为每一条指令加上注释,便于复杂程序的阅读。在设计通信、数学运算等高级应用程序时,建议使用语句表。

7. SIMATIC 指令集与 IEC 61131-3 指令集

STEP 7-Micro/WIN 提供两种指令集:SIMATIC 指令集与 IEC 61131-3 指令集,前者由西门子公司提供,它的某些指令不是 IEC 61131-3 中的标准指令。通常 SIMATIC 指令的执行时间短,可以使用梯形图、功能块图和语句表语言,而 IEC 61131-3 指令集只提供前两种语言。

IEC 61131-3 指令集的指令较少,其中的某些指令可以接受多种数据格式。例如 SIMATIC 指令集的正数指令分为 ADD_I(整数相加)、ADD_DI(双字整数相加)与

ADD_R（实数相加）等，IEC 61131-3 的加法指令 ADD 则未作区分，而是通过检验数据格式，由 CPU 自动选择正确的指令。因为 IEC 指令要检查参数中的数据格式，可以减少程序设计中的错误。

在 IEC 61131-3 指令编辑器中，有些指令是 SIMATIC 指令集的指令，它们作为 IEC 61131-3 指令集的非标准扩展，在 STEP 7-Micro/WIN 的指令列表中用红色的“+”号标记。

3.1.2 S7-200 的程序结构

S7-200 CPU 的控制程序由主程序、子程序和中断程序组成。

1. 主程序

主程序（OB1）是程序的主体，每个扫描周期都要执行一次主程序。每一个项目都必须有并且只能有一个主程序。在主程序中可以调用子程序，子程序又可以调用其他子程序。

2. 子程序

子程序是可选的，仅在被其他程序调用时执行。同一个子程序可以在不同的地方被多次调用。使用子程序可以简化程序代码和减少扫描时间。设计得好的子程序容易移植到别的项目中去。

3. 中断程序

中断程序用来及时处理与用户程序的执行时序无关的操作，或者用来处理不能事先预测何时发生的中断事件。中断程序不是由用户程序调用，而是在中断事件发生时由操作系统调用。中断程序是用户编写的。

3.2 数据类型与寻址方式

3.2.1 数制

1. 二进制数

所有数据在 PLC 中都是以二进制形式储存的，在编程软件中可以使用不同的数制。

(1) 用 1 位二进制数表示数字量

二进制数的 1 位（bit）只能取 0 和 1 这两个不同的值，可以用一个二进制位来表示开关量（或称数字量）的两种不同的状态，例如触点的断开和接通，线圈的通电和断电等。如果该位为 1，梯形图中对应的位编程元件（例如 M 和 Q）的线圈“通电”，其常开触点接通，常闭触点断开，以后称该编程元件为 1 状态，或称该编程元件为 ON（接通）。如果该位为 0，对应的编程元件的线圈和触点的状态与上述的相反，称该编程元件为 0 状态，或称该编程元件为 OFF（断开）。位数据的数据类型为 BOOL（布尔）型。

(2) 多位二进制数

多位二进制数用来表示大于 1 的数字，二进制数遵循逢 2 进 1 的运算规则，每一位都有一个固定的权值，从右往左的第 n 位（最低位为第 0 位）的权值为 2^n ，第 3 位至第 0 位的权值分别为 8、4、2、1，所以二进制数又称为 8421 码。

S7-200 用 2#来表示二进制常数。16 位二进制数 2#0000 0100 1000 0110 对应的十进制数为 $2^{10} + 2^7 + 2^2 + 2^1 = 1158$ 。

(3) 有符号数的表示方法

PLC 用二进制补码来表示有符号数，其最高位为符号位，最高位为 0 时为正数，为 1 时为负数。正数的补码是它本身，最大的 16 位二进制正数为 2#0111 1111 1111 1111，对应的十进制数为 32767。

将正数的补码逐位取反（0 变为 1，1 变为 0）后加 1，得到绝对值与它相同的负数的补码。例如将 1158 对应的补码 2#0000 0100 1000 0110 逐位取反后，得到 2#1111 1011 0111 1001，加 1 后得到 -1158 的补码 1111 1011 0111 1010。

将负数的补码的各位取反后加 1，得到它的绝对值对应的正数。例如将 -1158 的补码 2#1111 1011 0111 1010 逐位取反后得到 2#0000 0100 1000 0101，加 1 后得到 1158 的补码 2#0000 0100 1000 0110。表 3-1 给出了不同进制数的表示方法。常数的取值范围见表 3-2。

表 3-1 不同进制数的表示方法

十进制数	十六进制数	二进制数	BCD 码	十进制数	十六进制数	二进制数	BCD 码
0	0	0000	0000 0000	9	9	01001	0000 1001
1	1	00001	0000 0001	10	A	01010	0001 0000
2	2	00010	0000 0010	11	B	01011	0001 0001
3	3	00011	0000 0011	12	C	01100	0001 0010
4	4	00100	0000 0100	13	D	01101	0001 0011
5	5	00101	0000 0101	14	E	01110	0001 0100
6	6	00110	0000 0110	15	F	01111	0001 0101
7	7	00111	0000 0111	16	10	10000	0001 0110
8	8	01000	0000 1000	17	11	10001	0001 0111

表 3-2 常数的取值范围

数据的位数	无符号整数		有符号整数	
	十进制	十六进制	十进制	十六进制
B (字节): 8 位值	0~255	16#0~6#FF	-128~127	16#80~16#7F
W (字): 16 位值	0~65535	16#0~16#FFFF	-32768~32767	16#8000~16#7FFF
D (双字): 32 位值	0~4294967295	16#0~16#FFFF FFFF	-2147483648~2147483647	16#8000 0000~16#7FFF FFFF

2. 十六进制数

多位二进制数读、写很不方便，为了解决这个问题，可以用十六进制数来表示多位二进制数。十六进制数使用 16 个数字符号，即 0~9 和 A~F，A~F 分别对应于十进制数 10~15。可以用数字后面加“H”来表示十六进制常数，例如 AE75H。S7-200 用数字前面的“16#”来表示十六进制常数。4 位二进制数对应于 1 位十六进制数，例如二进制常数 2#1010 1110 0111 0101 可以转换为 16#AE75。

十六进制数采用逢 16 进 1 的运算规则，从右往左第 n 位的权值为 16^n （最低位的 n 为 0），例如 16#2F 对应的十进制数为 $2 \times 16^1 + 15 \times 16^0 = 47$ 。

3. BCD 码

BCD (Binary Coded Decimal) 码是各位按二进制编码的十进制数。每位十进制数用 4

位二进制数来表示，0~9 对应的二进制数为 0000~1001，各位 BCD 码之间的运算规则为逢 10 进 1。以 BCD 码 1001 0110 0111 0101 为例，对应的十进制数为 9675，最高的 4 位二进制数 1001 表示 9000。16 位 BCD 码对应于 4 位十进制数，允许的最大数字为 9999，最小的数字为 0000。

拨码开关（见图 3-5）的圆盘圆周面上有 0~9 这 10 个数字，用它面板上的按钮来增、减各位要输入的数字。它用内部的硬件将一位十进制数转换为 4 位二进制数。PLC 用输入点读取的多位拨码开关的输出值就是 BCD 码，需要用数据转换指令 BCD_I 将它转换为 16 位或 32 位整数。STEP 7-Micro/WIN 用十六进制格式（16#）表示 BCD 码，例如从图 3-5 的拨码开关读取的 BCD 码用 16#829 来表示。

用 PLC 的 4 个输出点给译码驱动芯片 4547 提供输入信号（见图 3-6），可以用 LED 七段显示器显示一位十进制数。需要用数据转换指令 I_BCD 将 PLC 中的 16 位二进制整数转换为 4 位 BCD 码，然后分别送给各位的译码驱动芯片。



图 3-5 拨码开关

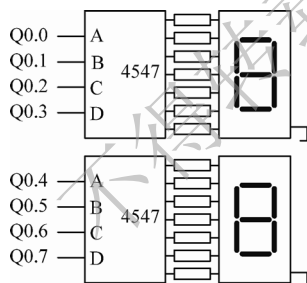


图 3-6 LED 七段显示器电路

3.2.2 数据类型

数据类型定义了数据的长度（位数）和表示方法。S7-200 的指令对操作数的数据类型有严格的要求。

1. 位

位（bit）数据的数据类型为 BOOL（布尔）型，BOOL 变量的值为 2#1 和 2#0。BOOL 变量的地址由字节地址和位地址组成，例如 I3.2 中的区域标示符“**I**”表示输入（Input），字节地址为 3，位地址为 2（见图 3-7）。这种访问方式称为“字节.位”寻址方式。

2. 字节

一个字节（Byte）由 8 个位数据组成，例如输入字节 IB3（B 是 Byte 的缩写）由 I3.0~I3.7 这 8 位组成（见图 3-7）。其中的第 0 位为最低位，第 7 位为最高位。

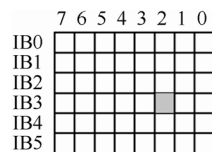


图 3-7 位数据与字节

3. 字和双字

相邻的两个字节组成一个字（Word），相邻的两个字组成一个双字（Double Word）。字和双字都是无符号数，它们用十六进制数来表示。

VW100 是由 VB100 和 VB101 组成的一个字（见图 3-8 和图 3-9），VW100 中的 V 为变量存储器的区域标示符，W 表示字。双字 VD100 由 VB100~VB103（或 VW100 和

VW102) 组成, VD100 中的 D 表示双字。字的取值范围为 16#0000~16#FFFF, 双字的取值范围为 16#0000 0000~16#FFFF FFFF。

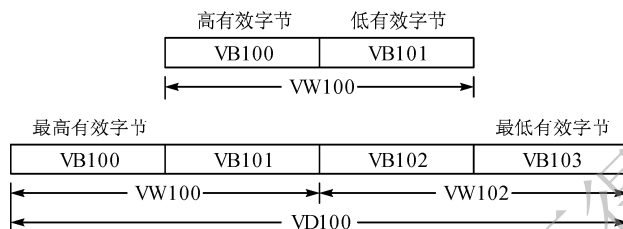


图 3-8 字节、字和双字

	地址	格式	当前值
11	VB100	十六进制	16#12
12	VB101	十六进制	16#34
13	VW100	十六进制	16#1234
14	VW102	十六进制	16#5678
15	VD100	十六进制	16#12345678
16	VD4	浮点数	50.0
17	VD4	二进制	2#0100_0010_0100_1000_0000_0000_0000_0000

图 3-9 状态表

需要注意下列问题:

1) 以组成字 VW100 和双字 VD100 的编号最小的字节 VB100 的编号作为 VW100 和 VD100 的编号。

2) 组成 VW100 和 VD100 的编号最小的字节 VB100 为 VW100 和 VD100 的最高位字节, 编号最大的字节为字和双字的最低位字节。

3) 数据类型字节、字和双字都是无符号数, 它们的数值用十六进制数表示。从图 3-9 可以看出字节、字和双字之间的关系。

4. 16 位整数和 32 位双整数

16 位整数 (Integer, INT) 和 32 位双整数 (Double Integer, DINT) 都是有符号数。整数的取值范围为-32768~32767, 双整数的取值范围为-2147483648~2147483647。

5. 32 位浮点数

实数 (REAL) 又称为浮点数, 可以表示为 $1.m \times 2^E$, 尾数中的 m 和指数 E 均为二进制数, E 可能是正数, 也可能是负数。ANSI/IEEE 754-1985 标准格式的 32 位实数的格式为 $1.m \times 2^e$, 式中指数 $e = E + 127$ ($1 \leq e \leq 254$), 为 8 位正整数。

ANSI/IEEE 标准浮点数的格式如图 3-10 所示, 共占用一个双字 (32 位)。最高位 (第 31 位) 为浮点数的符号位, 最高位为 0 时为正数, 为 1 时为负数; 8 位指数占第 23~30 位; 因为规定尾数的整数部分总是为 1, 只保留了尾数的小数部分 m (第 0~22 位)。第 22 位为 1 对应于 2^{-1} , 第 0 位为 1 对应于 2^{-23} 。浮点数的优点是用很小的存储空间 (4B) 可以表示非常大和非常小的数, 其取值范围为 $\pm 1.175495 \times 10^{-38} \sim \pm 3.402823 \times 10^{38}$ 。

在 STEP 7-Micro/WIN 中, 一般并不使用二进制格式或十六进制格式表示的浮点数, 而是用十进制小数来输入或显示浮点数 (见图 3-9), 例如在 STEP 7-Micro/WIN 中, 50 是 16 位整数, 而 50.0 为 32 位的浮点数。

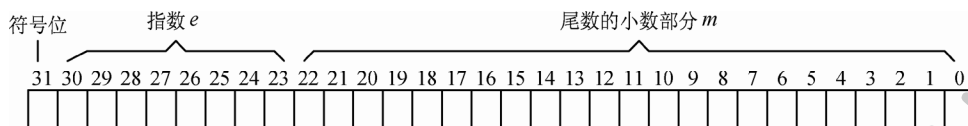


图 3-10 浮点数的结构

PLC 输入和输出的数值（例如模拟量输入值和模拟量输出值）大多是整数，用浮点数来处理这些数据需要进行整数和浮点数之间的相互转换，浮点数的运算速度比整数的运算速度慢一些。

6. ASCII 码字符

ASCII 码（美国信息交换标准代码）由美国国家标准局（ANSI）制定，它已被国际标准化组织（ISO）定为国际标准（ISO 646 标准）。标准 ASCII 码也叫作基础 ASCII 码，用 7 位二进制数来表示所有的英语大写、小写字母，数字 0~9、标点符号，以及在美式英语中使用的特殊控制字符。数字 0~9 的 ASCII 码为十六进制数 30H~39H，英语大写字母 A~Z 的 ASCII 码为 41H~5AH，英语小写字母 a~z 的 ASCII 码为 61H~7AH。

7. 字符串

数据类型为 STRING 的字符串由若干个 ASCII 码字符组成，第一个字节定义字符串的长度（0~254，见图 3-11），后面的每个字符占一个字节。变量字符串最多 255 个字节（长度字节加上 254 个字符）。

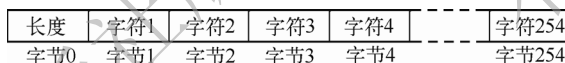


图 3-11 字符串的格式

3.2.3 CPU 的存储区

1. 过程映像输入寄存器（I）

在每个扫描周期的开始，CPU 对物理输入点进行采样，用过程映像输入寄存器来保存采样值。

过程映像输入寄存器是 PLC 接收外部输入的数字量信号的窗口。外部输入电路接通时对应的过程映像输入寄存器为 ON（1 状态），反之为 OFF（0 状态）。输入端可以外接常开触点或常闭触点，也可以接多个触点组成的串并联电路。在梯形图中，可以多次使用输入位的常开触点和常闭触点。

I、Q、V、M、S、SM 和 L 存储器区均可以按位、字节、字和双字来访问，例如 I3.5、IB2、IW4 和 ID6。

2. 过程映像输出寄存器（Q）

在扫描周期的末尾，CPU 将过程映像输出寄存器的数据传送给输出模块，再由后者驱动外部负载。如果梯形图中 Q0.0 的线圈“通电”，继电器型输出模块中对应的硬件继电器的常开触点闭合，使接在 Q0.0 对应的端子的外部负载通电，反之则该外部负载断电。输出模块中的每一个硬件继电器仅有一对常开触点，但是在梯形图中，每一个输出位的常开触点和常闭触点都可以多次使用。

3. 变量存储器区 (V)

变量 (Variable) 存储器用来在程序执行过程中存放中间结果, 或者用来保存与过程或任务有关的其他数据。

4. 位存储区 (M)

位存储器 (M0.0~M31.7) 类似于继电器控制系统的中间继电器, 用来存储中间操作状态或其他控制信息。S7-200 的 M 存储器只有 32B, 如果不够用可以用 V 存储器来代替 M 存储器。

5. 定时器存储区 (T)

定时器相当于继电器系统中的时间继电器。S7-200 有三种时间基准 (1ms、10ms 和 100ms) 的定时器。定时器的当前值为 16 位有符号整数, 用于存储定时器累计的时间基准增量值 (1~32767)。预设值是定时器指令的一部分。

定时器位用来描述定时器的延时动作的触点的状态。定时器位为 ON 时, 梯形图中对应的定时器的常开触点闭合, 常闭触点断开; 为 OFF 时则触点的状态相反。

用定时器地址 (例如 T5) 来访问定时器的当前值和定时器位, 带位操作数的指令用来访问定时器位, 带字操作数的指令用来访问当前值。

6. 计数器存储区 (C)

计数器用来累计其计数输入脉冲电平由低到高 (即上升沿) 的次数, S7-200 有加计数器、减计数器和加减计数器。计数器的当前值为 16 位有符号整数, 用来存放累计的脉冲数 (1~32767)。用计数器地址 (例如 C20) 来访问计数器的当前值和计数器位。带位操作数的指令访问计数器位, 带字操作数的指令访问当前值。

7. 高速计数器 (HC)

高速计数器用来累计比 CPU 的扫描速率更快的事件, 计数过程与扫描周期无关。其当前值和预设值为 32 位有符号整数, 当前值为只读数据。高速计数器的地址由区域标示符 HC 和高速计数器号组成, 例如 HC2。

8. 累加器 (AC)

累加器是可以像存储器那样使用的存储单元, CPU 提供了 4 个 32 位累加器 (AC0~AC3), 可以按字节、字和双字来访问累加器中的数据。按字节、字只能访问累加器的低 8 位或低 16 位, 按双字访问全部的 32 位, 访问的数据长度由所用的指令决定。例如在指令 “MOVW AC2, VW100” 中, AC2 按字 (W) 访问。累加器主要用来临时保存中间的运算结果。

9. 特殊存储器 (SM)

特殊存储器用于 CPU 与用户之间交换信息, 例如 SM0.0 一直为 ON, SM0.1 仅在执行用户程序的第一个扫描周期为 ON。SM0.4 和 SM0.5 分别提供周期为 1min 和 1s 的时钟脉冲。SM1.0、SM1.1 和 SM1.2 分别是零标志、溢出标志和负数标志。附录中的表 B-1 给出了常用特殊存储器位的描述。

10. 局部存储器区域 (L)

S7-200 将主程序、子程序和中断程序统称为 POU (程序组织单元), 各 POU 都有自己的 64B 的局部 (Local) 存储器。使用梯形图和功能块图时 STEP 7-Micro/WIN 保留局部存储器的最后 4B。

局部存储器简称为 L 存储器，仅仅在它被创建的 POU 中有效，各 POU 不能访问别的 POU 的局部存储器。局部存储器作为暂时存储器，或用于子程序的输入、输出参数。变量存储器 (V) 是全局存储器，可以被所有的 POU 访问。

S7-200 给主程序和它调用的 8 个子程序嵌套级别、中断程序和它调用的 1 个子程序嵌套级别各分配 64B 局部存储器。

11. 模拟量输入 (AI)

S7-200 用 A-D 转换器将现实世界连续变化的模拟量 (例如温度、电流、电压等) 转换为一个字长 (16 位) 的数字量，用区域标识符 AI、表示数据长度的 W (字) 和起始字节的地址来表示模拟量输入的地址，例如 AIW2。因为模拟量输入的长度为一个字，应从偶数字节地址开始存放，模拟量输入值为只读数据。

12. 模拟量输出 (AQ)

S7-200 将长度为一个字的数字用 D-A 转换器转换为现实世界的模拟量，用区域标识符 AQ、表示数据长度的 W (字) 和起始字节的地址来表示存储模拟量输出的地址，例如 AQW2。因为模拟量输出的长度为一个字，应从偶数字节地址开始存放，模拟量输出值是只写数据，用户不能读取模拟量输出值。

13. 顺序控制继电器 (S)

32B 的顺序控制继电器 (SCR) 位用于组织设备的顺序操作，与顺序控制继电器指令配合使用，详细的使用方法见 5.4 节。

14. CPU 存储器的范围与特性

各 CPU 具有下列相同的存储器范围：I0.0~I15.7、Q0.0~Q15.7、M0.0~M31.7、S0.0~S31.7、T0~T255、C0~C255、L0.0~L63.7、AC0~AC3、HC0~HC5。S7-200 其他存储器的范围见表 3-3。

表 3-3 S7-200 CPU 的部分存储器范围

描 述	CPU 221	CPU 222	CPU 224	CPU 224XP CPU 224XPsi	CPU 226
模拟量输入	AIW0~AIW30		AIW0~AIW62		
模拟量输出	AQW0~AQW30		AQW0~AQW62		
变量存储器	VB0~VB2047		VB0~VB8191	VB0~VB10239	
特殊存储器	SM0.0~SM179.7	SM0.0~SM299.7	SM0.0~SM549.7		

3.2.4 直接寻址与间接寻址

在 S7-200 中，通过地址访问数据，地址是访问数据的依据，访问数据的过程称为“寻址”。几乎所有的指令和功能都与各种形式的寻址有关。

1. 直接寻址

直接寻址指定了存储器的区域、长度和位置，例如 VW100 是 V 存储区中 16 位的字，其地址为 100。

2. 间接寻址的指针

间接寻址在指令中给出的不是操作数的值或操作数的地址，而是给出一个被称为地址指针的存储单元的地址，地址指针里存放的是真正的操作数的地址。

间接寻址常用于循环程序和查表程序。假设用循环程序来累加一片连续的存储区中的数值，每次循环累加一个数值。应在累加后修改指针中存储单元的地址值，使它指向下一个存储单元，为下一次循环的累加运算做好准备。没有间接寻址，就不能编写循环程序。

地址指针就像收音机调台的指针，改变指针的位置，指针指向不同的电台。改变地址指针中的地址值，地址指针“指向”不同的地址。

旅客入住酒店时，在前台办完入住手续，酒店就会给旅客一张房卡，房卡上面有房间号，旅客根据房间号使用酒店的房间。修改房卡中的房间号，别的旅客用同一张房卡就可以入住不同的房间。这里房卡就是地址指针，房间相当于存储单元，房间号就是存储单元的地址，旅客相当于存储单元中存放的数据。

S7-200 CPU 允许使用指针对下述存储区域进行间接寻址：I、Q、V、M、S、AI、AQ、SM、T（仅当前值）和 C（仅当前值）。间接寻址不能访问单个位（bit）地址、HC、L 存储区和累加器。

使用间接寻址之前，应创建一个指针。指针为双字存储单元，用来存放要访问的存储器的地址，只能用 V、L 或累加器作指针。建立指针时，用双字传送指令 MOVD 将需要间接寻址的存储器地址送到指针中，例如“MOVD &VB200, AC1”（见图 3-12）。&VB200 是 VB200 的地址，而不是 VB200 中的数值。

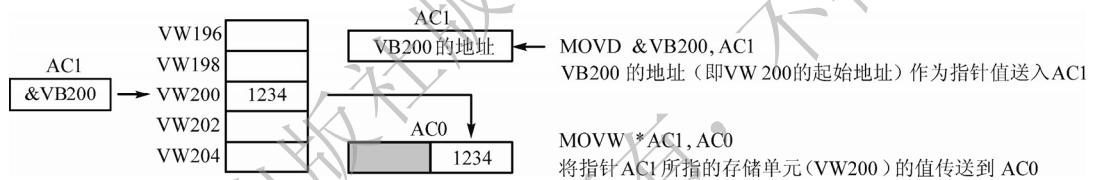


图 3-12 指针与间接寻址

3. 用指针访问数据

用指针访问数据时，操作数前加“*”号，表示该操作数为一个指针。图 3-12 的指令“MOVW *AC1, AC0”中，AC1 是一个指针，*AC1 是 AC1 所指的地址中的数据。图 3-12 存放在 VB200 和 VB201（即 VW200）中的数据被传送到累加器 AC0 的低 16 位。

4. 修改指针

用指针访问相邻的下一个数据时，因为指针是 32 位的数据，应使用双字指令来修改指针值，例如双字加法指令 ADDD 或双字递增指令 INCD。修改时记住需要调整的存储器地址的字节数，访问字节时，指针值加 1，访问字时，指针值加 2，访问双字时，指针值加 4。

【例 3-1】 用于非线性校正的表格存放在 VW100 开始的 10 个字中，表格的偏移量（表格中字的序号，第 1 个字的序号为 0）在 VD20 中。在 I0.0 的上升沿，用间接寻址将表格中相对于偏移量的数据值传送到 VW24 中去。用 AC1 作地址指针。下面是语句表程序。

```
LD      I0.0
EU
MOVD   &VB100, AC1 //表格的起始地址送 AC1
+D     VD20, AC1
+D     VD20, AC1 //起始地址加偏移量
MOVW   *AC1, VW24 //读取表格中的数据
```

一个字由两个字节组成，地址相邻的两个字的地址增量为 2（两个字节），所以用了两条双整数加法指令。图 4-29 给出了在循环程序中使用间接寻址的例子。

3.3 位逻辑指令

3.3.1 触点指令与逻辑堆栈指令

1. 标准触点指令

常开触点对应的位地址为 ON 时，该触点闭合，在语句表中，分别用 LD（Load，装载）、A（And，与）和 O（Or，或）指令来表示开始、串联和并联的常开触点（见图 3-13 和表 3-4）。触点指令中变量的数据类型为 BOOL 型。

表 3-4 标准触点指令

语 句	描 述	语 句	描 述
LD bit	装载，电路开始的常开触点	LDN bit	非（取反后装载），电路开始的常闭触点
A bit	与，串联的常开触点	AN bit	与非，串联的常闭触点
O bit	或，并联的常开触点	ON bit	或非，并联的常闭触点

常闭触点对应的位地址为 OFF 时，该触点闭合，在语句表中，分别用 LDN（Load Not，非，取反后装载）、AN（And Not，与非）和 ON（Or Not，或非）来表示开始、串联和并联的常闭触点。梯形图中触点中间的“/”表示常闭。

2. 输出指令

输出指令（=）对应于梯形图中的线圈。驱动线圈的触点电路接通时，有“能流”流过线圈，输出指令指定的位地址的值为 1，反之则为 0。输出指令将下面要介绍的堆栈的栈顶值复制到对应的位地址。本节的程序见配套资源中的例程“位逻辑指令”。

梯形图中两个并联的线圈（例如图 3-13 中 Q0.0 和 M0.4 的线圈）用两条相邻的输出指令来表示。图 3-13 中 I0.6 的常闭触点和 Q0.2 的线圈组成的串联电路与上面的两个线圈并联，但是该触点应使用 AN 指令，因为它与左边的电路串联。

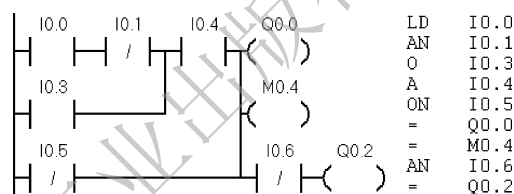


图 3-13 触点与输出指令

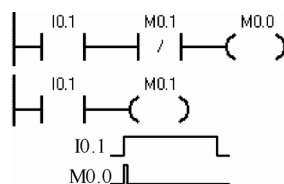


图 3-14 上升沿检测电路

【例 3-2】 已知图 3-14 中 I0.1 的波形，画出 M0.0 的波形。

在 I0.1 上升沿之前，I0.1 的常开触点断开，M0.0 和 M0.1 均为 OFF，其波形用低电平表示。在 I0.1 的上升沿，I0.1 变为 ON，CPU 先执行第一行的电路。因为前一扫描周期 M0.1 为 OFF，M0.1 的常闭触点闭合，所以 M0.0 变为 ON。执行第二行电路后，M0.1 变为 ON。从上升沿之后的第二个扫描周期开始，到 I0.1 变为 OFF 为止，M0.1 均为 ON，其常闭触点

断开，使 M0.0 为 OFF。因此，M0.0 只是在 I0.1 的上升沿 ON 一个扫描周期。

在分析电路的工作原理时，一定要有指令执行的先后顺序和循环扫描的概念。

如果交换图 3-14 中上下两行的位置，在 I0.1 的上升沿，M0.1 的线圈先“通电”，M0.1 的常闭触点断开，因此 M0.0 的线圈不会“通电”。由此可知，如果交换相互有关联的两块独立电路（即两个网络）的相对位置，可能会使有关的线圈“通电”或“断电”的时间提前或延后一个扫描周期，对于绝大多数系统，这是无关紧要的。但是在某些特殊情况下，可能会影响系统的正常运行。

3. 逻辑堆栈的基本概念

S7-200 有一个 9 位的堆栈，最上面的第一层称为栈顶（见图 3-16），用来存储逻辑运算的结果，下面的 8 位用来存储中间运算结果。堆栈中的数据一般按“先进后出”的原则访问，堆栈指令见表 3-5。AENO 指令的应用见 4.1.2 节。

表 3-5 与堆栈有关的指令

语 句	描 述	语 句	描 述
ALD	与装载，电路块串联连接	LPP	逻辑出栈
OLD	或装载，电路块并联连接	LDS N	装载堆栈
LPS	逻辑进栈	AENO	与 ENO
LRD	逻辑读栈		

执行 LD 指令时，将指令指定的位地址中的二进制数装载入栈顶。

执行 A（与）指令时，指令指定的位地址中的二进制数和栈顶中的二进制数作“与”运算，运算结果存入栈顶。栈顶之外其他各层的值不变。每次逻辑运算时只保留运算结果，栈顶原来的数值丢失。

执行 O（或）指令时，指令指定的位地址中的二进制数和栈顶中的二进制数作“或”运算，运算结果存入栈顶。

执行常闭触点对应的 LDN、AN 和 ON 指令时，取出指令指定的位地址中的二进制数后，先将它取反（0 变为 1，1 变为 0），然后再作对应的装载、与、或操作。

4. 或装载指令 OLD

OLD (Or Load) 指令对堆栈第一层和第二层中的两个二进制数进行“或”运算，运算结果存入栈顶。执行 OLD 指令后，堆栈的深度（即堆栈中保存的有效的数据个数）减 1。

触点的串并联指令只能将单个触点与别的触点或电路串并联。要想将图 3-15 中由 I0.3 和 I0.4 的触点组成的串联电路与它上面的电路并联，首先需要完成两个串联电路块内部的“与”逻辑运算（即触点的串联），这两个电路块用 LD 或 LDN 指令来表示电路块的起始触点。前两条指令执行完后，“与”运算的结果 $S0 = \overline{I0.0} \cdot I0.1$ 存放在图 3-16 的堆栈的栈顶，第 3、4 条指令执行完后，“与”运算的结果 $S1 = \overline{I0.3} \cdot I0.4$ 压入栈顶，原来在栈顶的 S0 被推到堆栈的第 2 层，第 2 层的数据被推到第 3 层……堆栈最下面一层的数据丢失。OLD 指令对堆栈第 1 层和第 2 层的数据作“或”运算（将两个串联电路块并联），并将运算结果 $S2 = S0 + S1$ 存入堆栈的栈顶，第 3~9 层中的数据依次向上移动一层。

OLD 指令不需要地址，它相当于需要并联的两块电路右端的一段垂直连线。图 3-16 堆

栈中的 x 表示不确定的值。

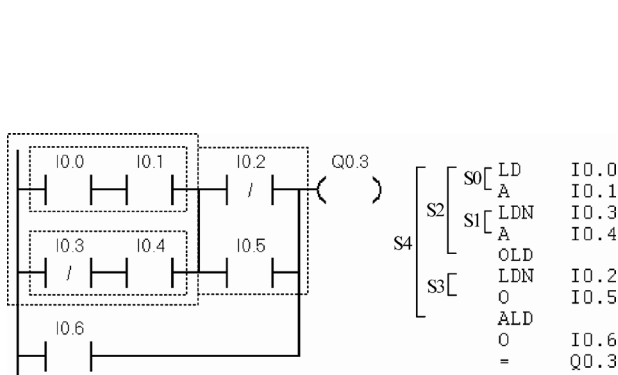


图 3-15 OLD 与 ALD 指令

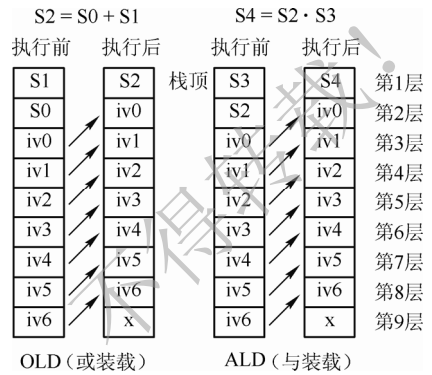


图 3-16 OLD 与 ALD 指令的堆栈操作

5. 与装载指令 ALD

图 3-15 的语句表中 OLD 下面的两条指令将两个触点并联，运算结果 $S3 = \overline{I0.2} + I0.5$ 被压入栈顶，堆栈中原来的数据依次向下一层推移，堆栈最底层的值被推出丢失。ALD (And Load) 指令对堆栈第 1 层和第 2 层的数据作“与”运算（将两个电路块串联），并将运算结果 $S4 = S2 \cdot S3$ 存入堆栈的栈顶，第 3~9 层中的数据依次向上移动一层。

将电路块串并联时，每增加一个用 LD 或 LDN 指令开始的电路块的运算结果，堆栈中将增加一个数据，堆栈深度加 1，每执行一条 ALD 或 OLD 指令，堆栈深度减 1。

梯形图和功能块图编辑器自动地插入处理堆栈操作所需要的指令。用 STEP 7-Micro/WIN 将梯形图转换为语句表程序时，自动生成堆栈指令。写入语句表程序时，必须由编程人员写入这些堆栈处理指令。

【例 3-3】 已知图 3-17 中的语句表程序，画出对应的梯形图。

对于较复杂的程序，特别是含有 ORB 和 ANB 指令时，在画梯形图之前，应分析清楚电路的串并联关系后，再开始画梯形图。首先将电路划分为若干块，各电路块从含有 LD 的指令（例如 LD、LDI 和 LDP 等）开始，在下一条含有 LD 的指令（包括 ALD 和 OLD）之前结束；然后分析各块电路之间的串并联关系。

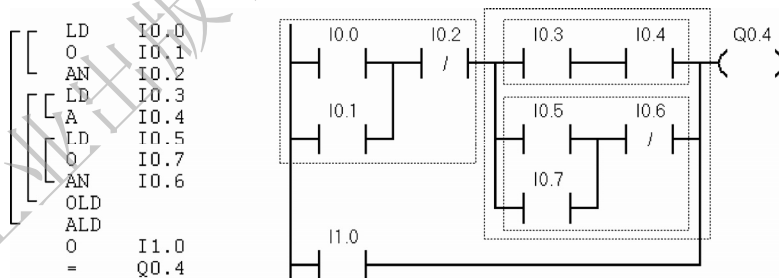


图 3-17 语句表与对应的梯形图

6. 其他逻辑堆栈操作指令

逻辑进栈 (Logic Push, LPS) 指令复制栈顶的值并将其压入堆栈的第 2 层，堆栈中原

来的数据依次向下一层推移，堆栈最底层的值被推出并丢失（见图 3-18）。

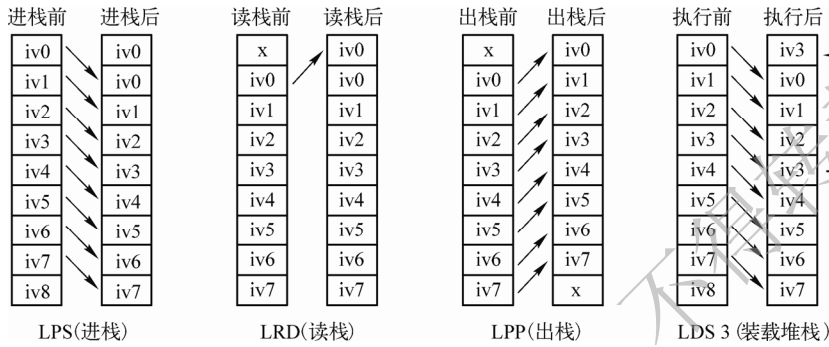


图 3-18 堆栈操作

逻辑读栈（Logic Read, LRD）指令将堆栈第 2 层的数据复制到栈顶，原来的栈顶值被复制值替代。第 2~9 层的数据不变。图中的 x 表示任意的数。

逻辑出栈（Logic Pop, LPP）指令将栈顶值弹出，堆栈各层的数据向上移动 1 层，第 2 层的数据成为新的栈顶值。可以用语句表的程序状态来监控堆栈中的数据（见图 2-20）。

装载堆栈（Load Stack, N = 0~8, LDS N）指令复制堆栈内第 N 层的值到栈顶。堆栈中原来的数据依次向下移动一层，堆栈最底层的值被推出并丢失。一般很少使用这条指令。

图 3-19 和图 3-20 的分支电路分别使用堆栈的第 2 层和第 2、3 层来保存电路分支处的逻辑运算结果。每一条 LPS 指令必须有一条对应的 LPP 指令，中间的支路使用 LRD 指令，处理最后一条支路时必须使用 LPP 指令。在一块独立电路中，用进栈指令同时保存在堆栈中的中间运算结果不能超过 8 个。

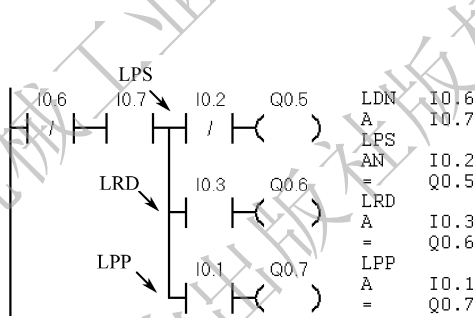


图 3-19 分支电路与堆栈指令

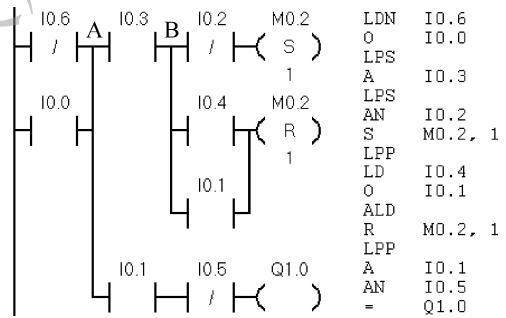


图 3-20 双重分支电路与堆栈指令

图 3-20 中的第 1 条 LPS 指令将栈顶的 A 点逻辑运算结果保存到堆栈的第 2 层，第 2 条 LPS 指令将 B 点的逻辑运算结果保存到堆栈的第 2 层，A 点的逻辑运算结果被“压”到堆栈的第 3 层。第 1 条 LPP 指令将堆栈第 2 层 B 点的逻辑运算结果上移到栈顶，第 3 层中 A 点的逻辑运算结果上移到堆栈的第 2 层。最后一条 LPP 指令将堆栈第 2 层的 A 点的逻辑运算结果上移到栈顶。从这个例子可以看出，堆栈“先入后出”的数据访问方式，刚好可以满足多层分支电路保存和取用分支点逻辑运算结果所要求的顺序。

7. 立即触点

立即 (Immediate) 触点指令只能用于输入位 I，执行立即触点指令时，立即读取物理输入点的值，根据该值决定触点的接通/断开状态，但是并不更新该物理输入点对应的过程映像输入寄存器。在语句表中，分别用 LDI、AI、OI 来表示开始、串联和并联的常开立即触点（见表 3-6）。分别用 LDNI、ANI、ONI 来表示开始、串联和并联的常闭立即触点。触点符号中间的“1”和“/1”分别用来表示常开立即触点和常闭立即触点（见图 3-21）。

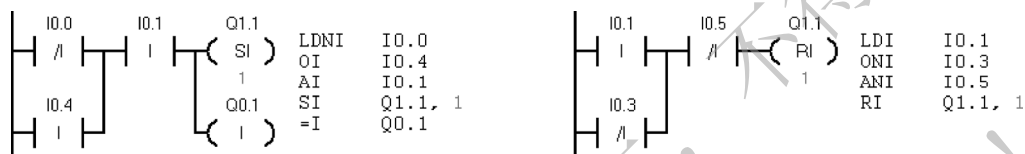


图 3-21 立即触点与立即输出指令

表 3-6 立即触点指令

语 句	描 述	语 句	描 述
LDI bit	立即装载，电路开始的常开触点	LDNI bit	取反后立即装载，电路开始的常闭触点
AI bit	立即与，串联的常开触点	ANI bit	立即与非，串联的常闭触点
OI bit	立即或，并联的常开触点	ONI bit	立即或非，并联的常闭触点

3.3.2 输出类指令与其他指令

输出类指令（见表 3-7）应放在梯形图同一行的最右边，指令中的变量为 BOOL 型（二进制位）。

表 3-7 输出类指令

语 句	描 述	语 句	描 述	语 句	描 述	梯形图符号	描 述
= bit	输出	S bit, N	置位	R bit, N	复位	SR	置位优先双稳态触发器
=I bit	立即输出	SI bit, N	立即置位	RI bit, N	立即复位	RS	复位优先双稳态触发器

1. 立即输出

立即输出指令 (=I) 只能用于输出位 Q，执行该指令时，将栈顶值立即写入指定的物理输出点和对应的过程映像输出寄存器。线圈符号中的“1”用来表示立即输出（见图 3-21）。

2. 置位与复位

置位指令 S (Set) 和复位指令 R (Reset) 用于将指定的位地址开始的 N 个连续的位地址置位 (变为 ON) 或复位 (变为 OFF)，N = 1~255，图 3-22 中 N = 1。

置位指令与复位指令最主要的特点是有记忆和保持功能。如果图 3-22 中 I0.1 的常开触点接通，M0.3 变为 ON。即使 I0.1 的常开触点断开，它也仍然保持为 ON。当 I0.2 的常开触点闭合时，M0.3 变为 OFF。即使 I0.3 的常开触点断开，它也仍然保持为 OFF。图 3-22 中的电路具有和图 1-12 中的起保停电路相同的功能。

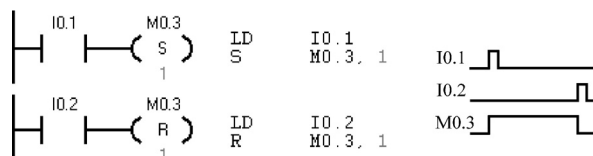


图 3-22 置位指令与复位指令

如果被指定复位的是定时器 (T) 或计数器 (C)，将清除定时器/计数器的当前值，它们的位被复位为 OFF。

使用程序状态监控时，置位指令和复位指令的线圈的状态只能反映线圈的通电和断电。需要用状态表来观察被置位和复位的 M0.3 的 ON/OFF 状态。

3. 立即置位与立即复位

执行立即置位指令 (SI) 或立即复位指令 (RI) 时 (见图 3-21)，从指定位地址开始的 N 个连续的物理输出点将被立即置位或复位， $N = 1 \sim 255$ ，线圈中的 I 表示立即。该指令只能用于输出位 Q，新值被同时写入对应的物理输出点和过程映像输出寄存器。置位指令与复位指令仅将新值写入过程映像输出寄存器。

4. RS、SR 双稳态触发器指令

图 3-23 中标有 SR 的是置位优先双稳态触发器，标有 RS 的是复位优先双稳态触发器。它们相当于置位指令 S 和复位指令 R 的组合，用置位输入和复位输入来控制方框上面的位地址。可选的 OUT 连接反映了方框上面位地址的信号状态。置位输入和复位输入均为 OFF 时，被控位的状态不变。置位输入和复位输入只有一个为 ON 时，为 ON 的起作用。

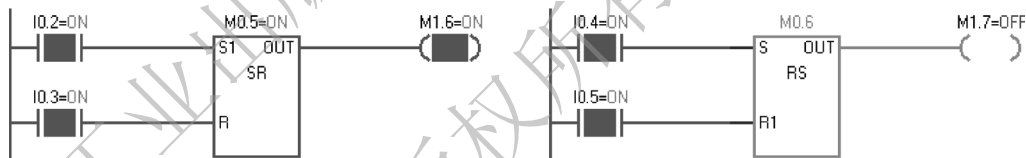


图 3-23 置位优先触发器与复位优先触发器

SR 触发器的置位信号 S1 和复位信号 R 同时为 ON 时，M0.5 被置位为 ON (见图 3-23)。RS 触发器的置位信号 S 和复位信号 R1 同时为 ON 时，M0.6 被复位为 OFF。

5. 其他位逻辑指令

(1) 正向负向转换触点

正向转换 (Positive Transition) 触点 (见图 3-24) 和负向转换 (Negative Transition) 触点没有操作数，触点符号中间的“P”和“N”分别表示表示正向转换和负向转换。

正向转换触点检测到一次正跳变时 (触点的输入信号由 0 变为 1)，或负向转换触点检测到一次负跳变时 (触点的输入信号由 1 变为 0)，触点接通一个扫描周期。语句表中正向、负向转换指令的助记符分别为 EU (Edge Up, 上升沿, 见表 3-8) 和 ED (Edge Down, 下降沿)。EU 或 ED 指令检测到堆栈的栈顶值有跳变时，将栈顶值设置为 1；否则将其设置为 0。

表 3-8 其他位逻辑指令

语 句	描 述
EU	正向转换
ED	负向转换
NOT	取反
NOP N	空操作

表 3-9 定时器地址与分辨率

类型	分辨率/ms	定时范围/s	定时器地址	类型	分辨率/ms	定时范围/s	定时器地址
TON/TOF	1	32.767	T32、T96	TONR	1	32.767	T0、T64
	10	327.67	T33~T36 和 T97~T100		10	327.67	T1~T4 和 T65~T68
	100	3276.7	T37~T63 和 T101~T255		100	3276.7	T5~T31 和 T69~T95

表 3-10 定时器指令与计数器指令

语 句	描 述	语 句	描 述
TON Txxx, PT	接通延时定时器	CITIM IN, OUT	计算间隔时间
TOF Txxx, PT	断开延时定时器	CTU Cxxx, PV	加计数
TONR Txxx, PT	有记忆接通延时定时器	CTD Cxxx, PV	减计数
BITIM OUT	开始间隔时间	CTUD Cxxx, PV	加/减计数

2. 接通延时定时器和有记忆接通延时定时器

定时器和计数器的当前值、定时器的预设时间（Preset Time, PT）的数据类型均为 16 位有符号整数（INT），允许的最大值为 32767。除了常数外，还可以用 VW、IW 等地址作定时器和计数器的预设值。

定时器方框指令左边的 IN 为使能输入端。接通延时定时器 TON 和有记忆接通延时定时器 TONR 的使能输入电路接通后开始定时，当前值不断增大。当前值大于等于 PT 端指定的预设值（1~32767）时，定时器位变为 ON，梯形图中对应的定时器的常开触点闭合，常闭触点断开。达到预设值后，当前值仍继续增加，直到最大值 32767。可以将定时器方框视为定时器的线圈。

定时器的预设时间等于预设值与分辨率的乘积，图 3-26 中的 T37 为 100ms 定时器（见配套资源中的例程“定时器应用”），预设时间为 $100\text{ms} \times 90 = 9\text{s}$ 。

接通延时定时器的使能输入电路断开时，定时器被复位，其当前值被清零，定时器位变为 OFF。还可以用复位（R）指令来复位定时器和计数器。

有记忆接通延时定时器 TONR 的使能输入电路断开时，当前值保持不变。使能输入电路再次接通时，继续定时。可以用 TONR 来累计输入电路接通的若干个时间间隔。图 3-27 中的时间间隔 $t_1 + t_2 = 10\text{s}$ 时，10ms 定时器 T2 的定时器位变为 ON。只能用复位指令来复位 TONR。

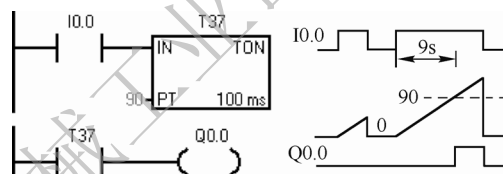


图 3-26 接通延时定时器

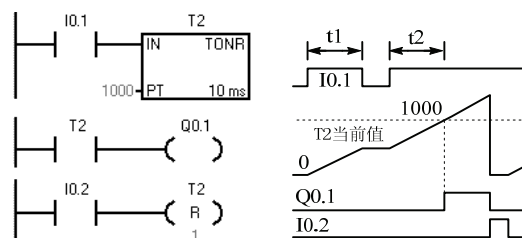


图 3-27 有记忆接通延时定时器

在第一个扫描周期，所有的定时器位被清零。没有记忆的定时器 TON 和 TOF 被自动复位，当前值和定时器位均被清零。可以在系统块中设置有断电保持功能的 TONR 的地址范

围。断电后再上电，有断电保持功能的 TONR 保持断电时的当前值不变。

如果要确保最小时间间隔，应将预设值 PT 增大 1。例如使用 100ms 定时器时，为确保最小时间间隔至少为 2000ms，应将 PV 设置为 21。

图 3-28 是用接通延时定时器编程实现的脉冲定时器程序，在 I0.3 由 OFF 变为 ON 时（波形的上升沿），Q0.2 输出一个宽度为 3s 的脉冲，I0.3 的脉冲宽度可以大于 3s，也可以小于 3s。

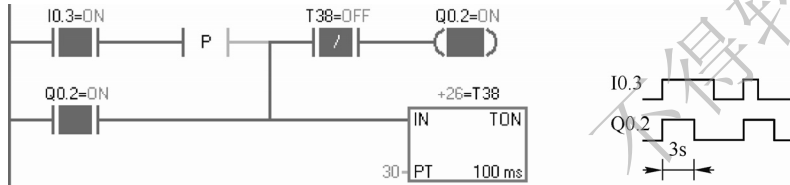


图 3-28 脉冲定时器

3. 断开延时定时器指令

断开延时定时器（TOF，见图 3-29）用来在使能输入（IN）电路断开后延时一段时间，再使定时器位变为 OFF。它用 IN 输入从 ON 到 OFF 的负跳变启动定时。

定时器的使能输入电路接通时，定时器位立即变为 ON，当前值被清零。使能输入电路断开时，开始定时，当前值从 0 开始增大。当前值等于预设值时，输出位变为 OFF，当前值保持不变，直到使能输入电路接通。断开延时定时器可用于设备停机后的延时，例如大型变频电动机的冷却风扇的延时。图 3-29 同时给出了断开延时定时器的语句表程序。



二维码 3-1

视频“定时器应用”可通过扫描二维码 3-1 播放。

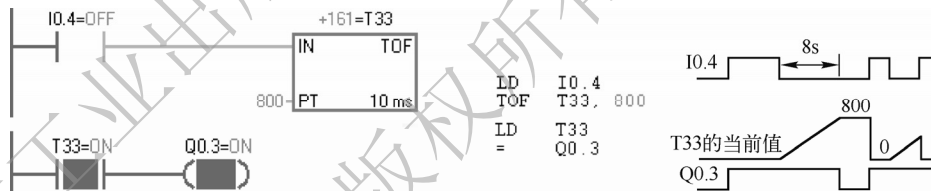


图 3-29 断开延时定时器

TOF 与 TON 不能使用相同的定时器号，例如不能同时对 T37 使用指令 TON 和 TOF。

4. 分辨率对定时器的影响

执行 1ms 分辨率的定时器指令时开始计时，其定时器位和当前值的更新与扫描周期不同步，每 1ms 更新一次。

执行 10ms 分辨率的定时器指令时开始计时，记录自定时器启用以来经过的 10ms 时间间隔的个数。在每个扫描周期开始时，10ms 分辨率定时器的定时器位和当前值被刷新，一个扫描周期累计的 10ms 时间间隔数被加到定时器当前值。定时器位和当前值在整个扫描周期中不变。

100ms 分辨率的定时器记录从定时器上次更新以来经过的 100ms 时间间隔的个数。在执行该定时器指令时，将从前一扫描周期起累积的 100ms 时间间隔个数累加到定时器的当前值。为了使定时器正确地定时，应确保在一个扫描周期中只执行一次 100ms 定时器指令。启用该定时器后如果在某个扫描周期内未执行定时器指令，或者在一个扫描周期多次执行同一

条定时器指令，定时时间都会出错。

5. 间隔时间定时器

在图 3-30 的 Q0.4 的上升沿执行“开始间隔时间”指令 BGN_ITIME，读取内置的 1ms 双字计数器的当前值，并将该值储存在 VD0 中。

“计算间隔时间”指令 CAL_ITIME 计算当前时间与 IN 输入端的 VD0 提供的时间（即图 3-30 中 Q0.4 变为 ON 的时间）之差，并将该时间差储存在 OUT 端指定的 VD4 中。

双字计数器的最大计时间隔为 2^{32} ms 或 49.7 天。CAL_ITIME 指令将自动处理计算时间间隔期间发生的 1ms 定时器的翻转（即定时器的值由最大值变为 0）。

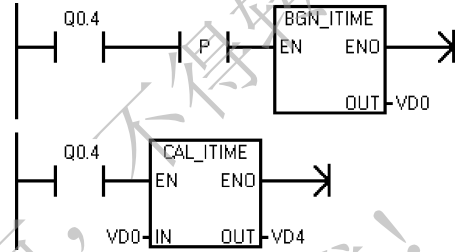


图 3-30 间隔时间定时器

3.4.2 计数器指令

计数器的地址范围为 C0~C255，不同类型的计数器不能共用同一个计数器地址。本节的程序见配套资源中的例程“计数器应用”。

1. 加计数器（CTU）

同时满足下列条件时，加计数器的当前值加 1（见图 3-31），直至计数最大值 32767。

- 1) 接在 R 输入端的复位输入电路断开（未复位）。
- 2) 接在 CU 输入端的加计数脉冲输入电路由断开变为接通（即 CU 信号的上升沿）。
- 3) 当前值小于最大值 32767。

当前值大于等于数据类型为 INT 的预设值 PV 时，计数器位为 ON，反之为 OFF。当复位输入 R 为 ON 或对计数器执行复位指令时，计数器被复位，计数器位变为 OFF，当前值被清零。在首次扫描时，所有的计数器位被复位为 OFF。可以用系统块设置有断电保持功能的计数器的地址范围。断电后又上电，有断电保持功能的计数器保持断电时的当前值不变。

在语句表中，栈顶值是复位输入（R），加计数输入值（CU）放在堆栈的第 2 层。

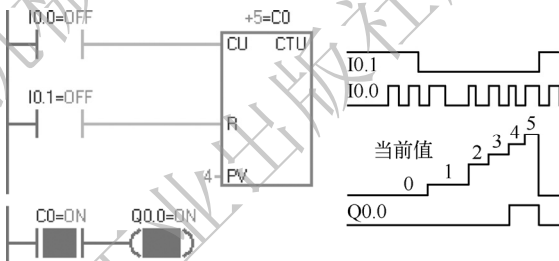


图 3-31 加计数器

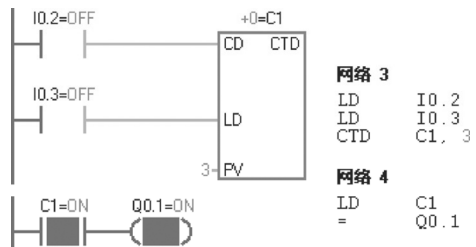


图 3-32 减计数器

2. 减计数器（CTD）

在装载输入 LD 的上升沿，计数器位被复位为 OFF，并把预设值 PV 装入当前值寄存器。在减计数脉冲输入信号 CD（见图 3-32）的上升沿，从预设值开始，减计数器的当前值减 1。减至 0 时，停止计数，计数器位被置位为 ON。

在语句表中，栈顶值是装载输入 LD，减计数输入 CD 放在堆栈的第 2 层。图 3-32 同时

给出了减计数器的语句表程序。

3. 加减计数器 (CTUD)

在加计数脉冲输入 CU (见图 3-33) 的上升沿, 计数器的当前值加 1。在减计数脉冲输入 CD 的上升沿, 计数器的当前值减 1。当前值大于等于预设值 PV 时, 计数器位为 ON, 反之为 OFF。若复位输入 R 为 ON, 或对计数器执行复位 (R) 指令时, 计数器被复位。当前值为最大值 32767 (十六进制数 16#7FFF) 时, 下一个 CU 输入的上升沿使当前值加 1 后变为最小值 -32768 (十六进制数 16#8000)。当前值为 -32768 时, 下一个 CD 输入的上升沿使当前值减 1 后变为最大值 32767。

在语句表中, 栈顶值是复位输入 R, 减计数输入 CD 在堆栈的第 2 层, 加计数输入 CU 在堆栈的第 3 层。

视频“计数器应用”可通过扫描二维码 3-2 播放。

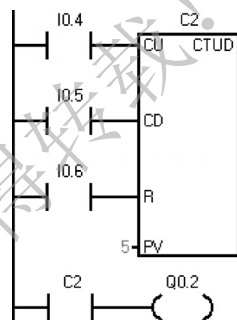


图 3-33 加减计数器



二维码 3-2

3.5 习题

1. 填空

- 1) 输出指令 (对应于梯形图中的线圈) 不能用于过程映像_____寄存器。
- 2) SM_____在首次扫描时为 ON, SM0.0 一直为_____。
- 3) 每一位 BCD 码用_____位二进制数来表示, 其取值范围为二进制数_____~_____。
- 4) 二进制数 2#0100 0001 1000 0101 对应的十六进制数是_____, 对应的十进制数是_____, 绝对值与它相同的负数的补码是_____。
- 5) BCD 码 2#0100 0001 1000 0101 对应的十进制数是_____。
- 6) 接通延时定时器 TON 的使能 (IN) 输入电路_____时开始定时。当前值大于等于预设值时, 其定时器位变为_____, 梯形图中其常开触点_____, 常闭触点_____。
- 7) 接通延时定时器 TON 的使能输入电路_____时被复位, 复位后梯形图中其常开触点_____, 常闭触点_____, 当前值等于_____。
- 8) 有记忆接通延时定时器 TONR 的使能输入电路_____时开始定时, 使能输入电路断开时, 当前值_____。使能输入电路再次接通时_____。必须用_____指令来复位 TONR。
- 9) 断开延时定时器 TOF 的使能输入电路接通时, 定时器位立即变为_____, 当前值被_____. 使能输入电路断开时, 当前值从 0 开始_____. 当前值等于预设值时, 输出位变为_____, 梯形图中其常开触点_____, 常闭触点_____, 当前值_____。
- 10) 若加计数器的计数输入电路 CU_____, 复位输入电路 R_____, 计数器的当前值加 1。当前值大于等于预设值 PV 时, 梯形图中其常开触点_____, 常闭触点_____. 复位输入电路_____时, 计数器被复位, 复位后梯形图中其常开触点_____, 常闭触点_____, 当前值为_____。
2. 2#0010 1010 0011 1001 是 BCD 码吗? 为什么?
3. 求出二进制补码 2#1111 1111 1010 0101 对应的十进制数。
4. 状态表用什么数据格式显示 BCD 码?
5. 字节、字和双字是有符号数还是无符号数?

6. VW20 由哪两个字节组成？谁是高位字节？
7. VD20 由哪两个字组成？由哪 4 个字节组成？谁是低位字？谁是最高位字节？
8. 在 STEP 7-Micro/WIN 中，用什么格式键入和显示浮点数？
9. 字符串的第一个字节用来干什么？
10. 位存储器（M）有多少个字节？
11. T31、T32、T33 和 T38 分别属于什么定时器？它们的分辨率分别是多少毫秒？
12. S7-200 有几个累加器？它们可以用来保存多少位的数据？
13. POU 是什么的缩写？它包括哪些程序？
14. 模拟量输入 AIB2、AIW1 和 AIW2 哪一个表示方式是正确的？
15. &VB100 和*VD120 分别用来表示什么？
16. 地址指针有什么作用？
17. 写出图 3-34 所示梯形图对应的语句表程序。

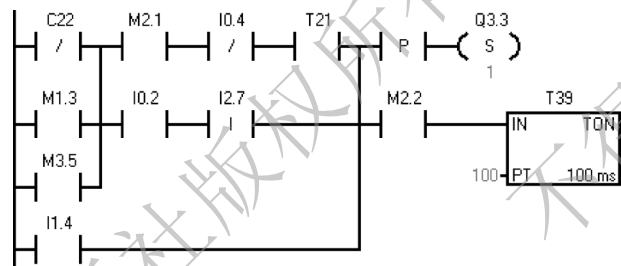


图 3-34 题 17 的图

18. 写出图 3-35 所示梯形图对应的语句表程序。
19. 写出图 3-36 所示梯形图对应的语句表程序。

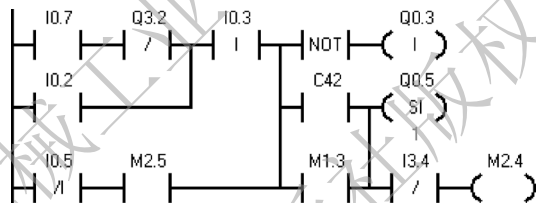


图 3-35 题 18 的图

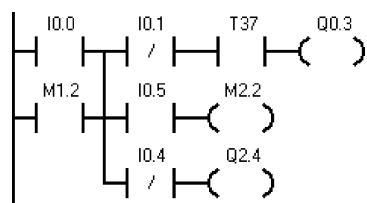


图 3-36 题 19 的图

20. 画出图 3-37 中 M0.0、M0.1 和 Q0.0 的波形图。
21. 指出图 3-38 中的错误。

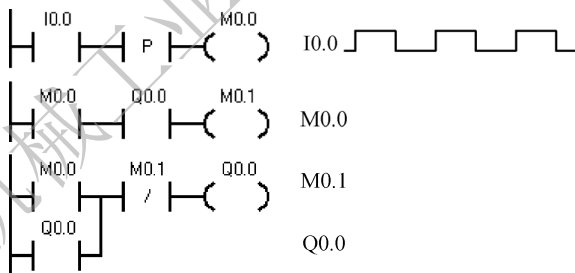


图 3-37 题 20 的图

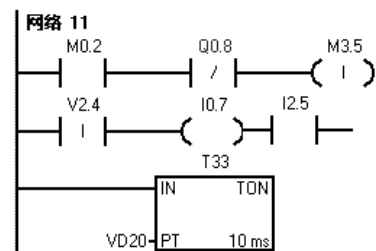


图 3-38 题 21 的图

22. 画出图 3-39a 中的语句表对应的梯形图。
23. 画出图 3-39b 中的语句表对应的梯形图。
24. 画出图 3-39c 中的语句表对应的梯形图。

网络 1		网络 2		网络 3	
LDI	I0.2	LD	I0.1	LD	I0.7
AN	I0.0	AN	I0.0	AN	I2.7
O	Q0.3	LPS		LDI	I0.3
ONI	I0.1	AN	I0.2	ON	I0.1
LD	Q2.1	LPS		A	M0.1
O	M3.7	A	I0.4	OLD	
AN	I1.5	=	Q2.1	LD	I0.5
LDN	I0.5	LPP		A	I0.3
A	I0.4	A	I4.6	O	I0.4
ON	M0.2	R	Q0.3, 1	ALD	
OLD		LPP		ON	M0.2
ALD		LPS		=I	Q0.4
O	I0.4	A	I0.5	网络 4	
EU		=	M3.6	LD	I2.5
=	M3.7	LPP		LD	M3.5
LPP		AN	I0.4	ED	
AN	I0.4	TON	T37, 25	CTU	C41, 30
NOT					
SI	Q0.3, 1				

a)

b)

c)

图 3-39 题 22~题 24 的图

25. 用接在 I0.0 输入端的光电开关检测传送带上通过的产品，有产品通过时 I0.0 为 ON，如果在 10s 内没有产品通过，由 Q0.0 发出报警信号，用 I0.1 输入端外接的开关解除报警信号。画出梯形图，并写出对应的语句表程序。

26. 用 S、R 和转换触点指令设计满足图 3-40 所示波形的梯形图。

27. 在按钮 I0.0 按下时 Q0.0 变为 ON 并自保持（见图 3-41）。用加计数器 C1 计数，I0.1 输入 3 个脉冲后，T37 开始定时。5s 后 Q0.0 变为 OFF，同时 C1 和 T37 被复位。在 PLC 刚开始执行用户程序时，C1 也被复位，设计出梯形图。



图 3-40 题 26 的图

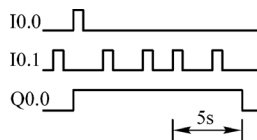


图 3-41 题 27 的图