



“十三五”普通高等教育本科规划教材

PLC YUANLI JI GONGCHENG YINGYONG

# PLC原理及工程应用

## (第二版)

刘星平 主 编

赖指南 杨德良 副主编



中国电力出版社  
CHINA ELECTRIC POWER PRESS

## “十三五”普通高等教育本科规划教材

微机原理与接口技术	王亭岭
自动控制理论（第四版）	陈青
自动控制原理	宋晓燕
电力电子技术及应用	李练兵
电机与电力拖动基础	蒋惠忠
电器与可编程控制器	王大虎
可编程控制器与现场总线	梁涛

### PLC原理及工程应用（第二版） 刘星平

电气控制与PLC技术应用实训教程	李胜多
GE可编程自动控制器原理与应用	陈志武
检测技术及仪表（第二版）	马宏忠
信号与系统分析	杜保强
工业控制网络技术（第二版）	赵新秋
智能控制系统及应用（第二版）	张国忠
过程控制系统	黄伟
过程控制系统设计实践指导	黄伟
过程参数检测技术	苏杰
运动控制系统	张崇巍
LabVIEW程序设计	崔凤英
电气与自动化类专业毕业设计指导	李阳
自动化专业英语（第二版）	王建国
计算机控制技术（第二版）	苏小林



刮开涂层  
查询真伪



中国电力出版社官方微信



中国电力教材服务官方微信

中国电力出版社教材中心

教材网址 <http://jc.cepp.sgcc.com.cn>

服务热线 010-63412548 63412523

ISBN 978-7-5123-8118-6



9 787512 381186 >

定价：29.00 元



“十三五”普通高等教育本科规划教材

# PLC原理及工程应用

## (第二版)

主 编 刘星平  
副主编 赖指南 杨德良  
编 写 刘团林 易灵芝 陈铁军 沈细群  
主 审 郁汉琪



中国电力出版社  
CHINA ELECTRIC POWER PRESS

## 内 容 提 要

本书为“十三五”普通高等教育本科规划教材。全书共分9章，主要内容包括继电器控制系统基础、可编程控制器概述、S7-200系列PLC的硬件和编程软件、S7-200 PLC的编程指令、S7-200 PLC的编程方法、S7-200 PLC在模拟量闭环控制中的应用、S7-200 PLC的通信及网络控制、S7-200 PLC控制系统的设计与应用、HMI的组态与应用。

本书可作为本科高等院校自动化、电气工程、电子信息、机电一体化、测控技术等相关专业的教材，也可供工程技术人员自学或作为培训教材使用。

## 图书在版编目(CIP)数据

PLC原理及工程应用/刘星平主编. —2版. —北京: 中国电力出版社, 2015. 8

“十三五”普通高等教育本科规划教材

ISBN 978-7-5123-8118-6

I. ①P… II. ①刘… III. ①plc技术-高等学校-教材 IV. ①TM571.6

中国版本图书馆CIP数据核字(2015)第176128号

中国电力出版社出版、发行

(北京市东城区北京站西街19号 100005 <http://www.cepp.sgcc.com.cn>)

北京市同江印刷厂印刷

各地新华书店经售

\*

2009年8月第一版

2015年8月第二版 2015年8月北京第四次印刷

787毫米×1092毫米 16开本 14.25印张 346千字

定价 29.00元

## 敬告读者

本书封底贴有防伪标签，刮开涂层可查询真伪  
本书如有印装质量问题，我社发行部负责退换

版权专有 翻印必究

# 前 言

本书是在第一版的基础上，为适应 PLC 的最新发展与应用，对原教材中的部分内容作了调整和删除，增加了触摸屏控制、S7-200 SMART 系列 PLC 等一些新内容，章节的编排上更适合深入浅出的学习顺序和规律。

随着技术的发展，可编程控制器的功能也越来越强大，原来很多由继电器控制系统（俗称电气控制技术）实现的功能可以很容易地由可编程控制器来实现，正是基于这一考虑，所以把电气控制技术与可编程控制器这两门课程合并，书中只保留了继电器控制系统的经典知识及基本应用。

可编程控制器简称 PLC，是以微处理器为核心的工业自动控制通用装置。它具有控制功能强、可靠性高、使用灵活方便、易于扩展、通用性强等优点，不仅可以取代继电器控制系统，还可以进行复杂的生产过程控制和应用于工厂自动化网络，被誉为现代工业生产自动化的三大支柱之一。因此，学习和掌握 PLC 应用技术已成为工程技术人员的紧迫任务。

本书编写时力求由浅入深、通俗易懂、理论联系实际、注重应用，充分体现教材内容的适用性和先进性。结合作者多年的工程应用实践和教学经验，在书中的相关章节列举了大量的工程实例，所列举的例子都可以作为教学案例来讲课和学习。

本书从应用的角度出发，系统地介绍了 PLC 硬件组成、工作原理和性能指标，以西门子公司 S7-200 系列 PLC 为样机，详细介绍了其指令系统及应用、PLC 程序设计的方法与技巧、PLC 控制系统设计应注意的问题等。为了适应新的发展需要，本书还介绍了 PLC 在模拟量过程控制系统中的应用、触摸屏或计算机对 PLC 控制系统的监控、PLC 的网络通信与控制、高速计数及脉冲输出的编程应用等。

全书共分 9 章。第 1 章继电器控制系统基础、第 2 章可编程控制器概述、第 3 章 S7-200 系列 PLC 的硬件和编程软件、第 4 章 S7-200 PLC 的编程指令、第 5 章 S7-200 PLC 的编程方法、第 6 章 S7-200 PLC 在模拟量闭环控制中的应用，第 7 章 S7-200 PLC 的通信及网络控制、第 8 章 S7-200 PLC 控制系统的设计与应用、第 9 章 HMI 的组态与应用。每章后附有习题，供读者练习与上机实践。书中标有 \* 号的章节可根据教学的要求和学时选用或参考。

本书由湖南工程学院刘星平主编。湖南工程学院赖指南（第 1、4 章的部分内容）和湘潭职业技术学院杨德良（第 8 章）任副主编。昆山市佰奥自动化设备科技有限公司的刘团林工程师，湘潭大学的易灵芝教授，湘潭技师学院的陈铁军老师，湖南工程学院的沈细群老师也参加了本书的编写工作。全书由刘星平统稿。南京工程学院郁汉琪教授担任本书主审，提出了许多宝贵的意见，在此表示诚挚的谢意。

由于编者水平有限，书中难免有不足和错误之处，恳请读者批评指正。

编 者

2015 年 7 月

# 目 录

前言

<b>第 1 章 继电器控制系统基础</b> .....	1
1.1 继电器控制系统的基本单元 .....	1
1.2 继电器控制系统常用的一些概念及基本控制方式 .....	4
* 1.3 电液控制 .....	8
习题 .....	10
<b>第 2 章 可编程控制器概述</b> .....	11
2.1 可编程控制器的产生及定义 .....	11
2.2 PLC 的组成 .....	11
2.3 PLC 的工作原理 .....	14
2.4 PLC 的特点及应用领域 .....	17
习题 .....	19
<b>第 3 章 S7-200 系列 PLC 的硬件和编程软件</b> .....	21
3.1 S7-200 可编程控制器的组成 .....	21
3.2 S7-200(CN)PLC 的编程元件 .....	25
3.3 S7-200 PLC 的数据类型及表示方法 .....	29
3.4 S7-200 PLC 的寻址方式 .....	30
3.5 S7-200 编程语言及程序结构 .....	31
3.6 STEP7-Micro/WIN V4.0 编程软件的使用与安装 .....	33
3.7 梯形图程序的执行原理及编程规则 .....	38
习题 .....	39
<b>第 4 章 S7-200 PLC 的编程指令</b> .....	40
4.1 基本逻辑指令 .....	40
4.2 运算指令 .....	51
4.3 数据处理指令 .....	57
4.4 程序控制类指令 .....	60
* 4.5 特殊指令 .....	67
4.6 S7-200 系列 PLC 仿真软件及其应用 .....	80
习题 .....	82
<b>第 5 章 S7-200 PLC 的编程方法</b> .....	83
5.1 梯形图的基本电路 .....	83
5.2 梯形图的经验设计法 .....	88
5.3 顺序控制设计法与顺序功能图 .....	90
5.4 使用启、保、停电路的顺序控制梯形图编程方法 .....	97

5.5	使用置位复位指令的顺序控制梯形图编程方法 .....	104
5.6	具有多种工作方式的系统的编程方法 .....	108
	习题 .....	114
<b>第 6 章</b>	<b>S7 - 200 PLC 在模拟量闭环控制中的应用 .....</b>	<b>117</b>
6.1	模拟量闭环控制的基本概念 .....	117
6.2	S7 - 200 PLC 的模拟量扩展模块 .....	121
6.3	数字 PID 控制器 .....	123
6.4	S7 - 200 PLC 的 PID 控制及其应用 .....	125
	习题 .....	136
<b>第 7 章</b>	<b>S7 - 200 PLC 的通信及网络控制 .....</b>	<b>137</b>
7.1	计算机通信方式与串行接口 .....	137
7.2	S7 - 200 PLC 的通信控制网络 .....	139
7.3	网络读写 .....	147
7.4	自由口通信技术 .....	153
7.5	S7 - 200 PLC 的 PROFIBUS - DP 通信 .....	160
	习题 .....	165
<b>第 8 章</b>	<b>S7 - 200 PLC 控制系统的设计与应用 .....</b>	<b>166</b>
8.1	系统设计 .....	166
8.2	S7 - 200 PLC 应用系统的可靠性措施 .....	169
8.3	节省 PLC 输入输出点数的方法 .....	172
8.4	设计实例一:3 工位旋转工作台的 PLC 控制 .....	174
8.5	设计实例二:移动式卫生间 PLC 控制系统的设计与应用 .....	178
8.6	设计实例三:水箱加热系统的 PLC 位式温度控制 .....	181
8.7	S7 - 200 SMART 系列 PLC 及其应用 .....	185
	习题 .....	190
<b>第 9 章</b>	<b>HMI 的组态与应用 .....</b>	<b>192</b>
9.1	人机操作界面 .....	192
9.2	文本显示器的组态与应用 .....	193
9.3	触摸屏的组态与应用 .....	197
9.4	基于 PC 的组态与应用 .....	202
9.5	组态软件对 PLC 的监控应用举例一 .....	206
9.6	组态软件对 PLC 的监控应用举例二 .....	211
	习题 .....	212
<b>附录 A</b>	<b>S7 - 200 PLC 的精简指令集 .....</b>	<b>214</b>
<b>附录 B</b>	<b>S7 - 200 PLC 的 CPU 模块的技术性能指标 .....</b>	<b>219</b>
<b>附录 C</b>	<b>S7 - 200 SMART PLC 的 CPU 模块的技术性能指标 .....</b>	<b>220</b>
	参考文献 .....	221

## 第1章 继电器控制系统基础

PLC是从继电器控制系统发展而来的。它的梯形图程序与继电器系统电路图相似，所以梯形图中的某些编程元件也沿用了继电器这一名称。

这种用计算机程序实现的软继电器，与继电器系统中的物理继电器在功能上有某些相似之处。下面先介绍一下继电接触器控制系统的基本知识。

### 1.1 继电器控制系统的基本单元

#### 1.1.1 继电器

继电器用于控制电路，电流小，没有灭弧装置，可在电量或非电量的作用下动作。它由电磁线圈、铁芯、触点和复位弹簧组成。

继电器是一种电子控制器件，它具有控制系统（又称输入回路）和被控制系统（又称输出回路），通常应用于自动控制电路中，它实际上是用较小的电流去控制较大电流的一种“自动开关”。电磁式继电器一般由铁芯、线圈、衔铁、触点簧片等组成（如图1-1所示）。只要在线圈两端加上一定的电压，线圈中就会流过一定的电流，从而产生电磁效应，衔铁就会在电磁力吸引的作用下克服返回弹簧的拉力吸向铁芯，从而带动衔铁的动触点与静触点（动合触点）吸合。当线圈断电后，电磁的吸力也随之消失，衔铁就会在弹簧的反作用力下返回到原来的位置，使动触点与原来的静触点（动断触点）吸合。这样通过吸合、释放，从而达到了在电路中的导通、切断的目的。从继电器的工作原理可以看出，它是一种机电元件，通过机械动作来实现触点的通断，是有触点元件。

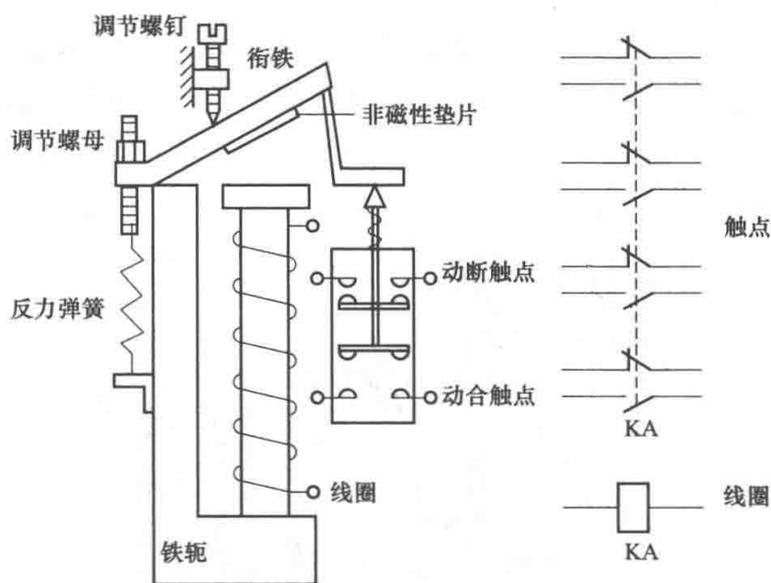


图 1-1 电磁式继电器组成示意图及图形符号

衔铁就会在弹簧的反作用力下返回到原来的位置，使动触点与原来的静触点（动断触点）吸合。这样通过吸合、释放，从而达到了在电路中的导通、切断的目的。从继电器的工作原理可以看出，它是一种机电元件，通过机械动作来实现触点的通断，是有触点元件。

#### 1.1.2 接触器

接触器的结构和工作原理与继电器的基本相同，接触器也是利用电磁吸力的原理工作的，主要由触头系统、电磁系统、灭弧装置、支架底座、外壳组成。它用于主电路，电流大，有灭弧装置，一般只能在电压作用下动作。电磁机构通常包括吸引线圈、铁芯和衔铁三部分。图1-2为接触器的原理结构示意图及图形符号。

#### 1.1.3 热继电器

电动机长期过载、频繁启动、欠电压、断相运行均会引起过电流。热继电器是具有过载

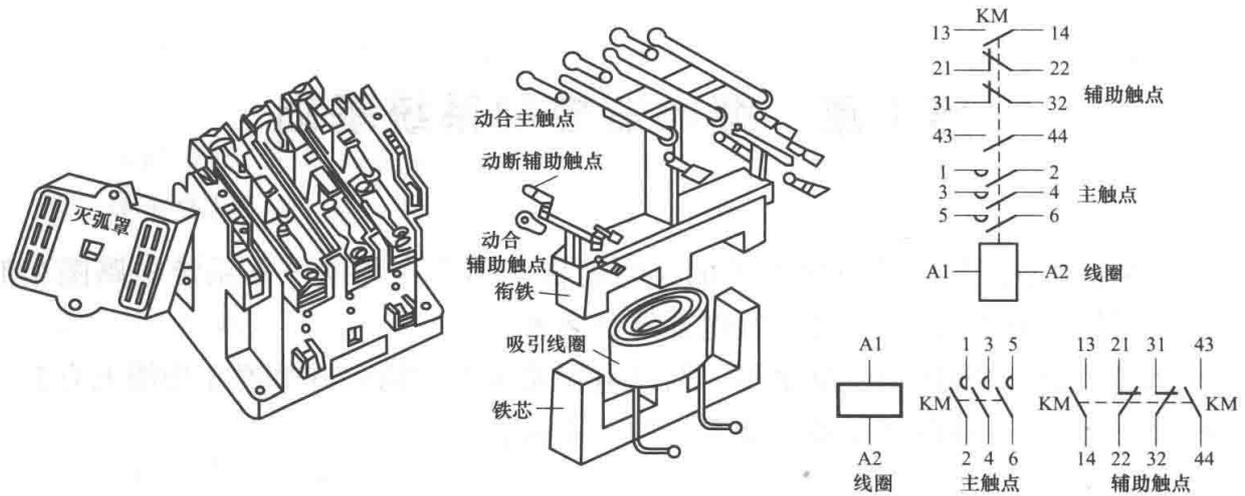


图 1-2 接触器的原理结构示意图及图形符号

保护特性的过电流继电器，它是利用电流的热效应来切断电路的保护电器。它在控制电路中，用作电动机的过载保护和断相保护，既能保证电动机不超过容许的过载，又可以最大限度地保证电动机的过载能力。当然，首先要保证电动机的正常启动。

图 1-3 为热继电器结构原理图，作电动机过载保护时，将热元件 3 串接在电动机定子绕组中。当电动机正常运行时，热元件产生的热量虽然能使双金属片 2 发生弯曲，但不足以使热继电器动作。当电动机过载时，热元件产生的热量增大，使双金属片弯曲位移增大，从而推动导板 4，并通过补偿金属片 5 与推杆 14 带动触点系统动作。通常用热继电器的动断触点断开控制电路，以实现过载保护。

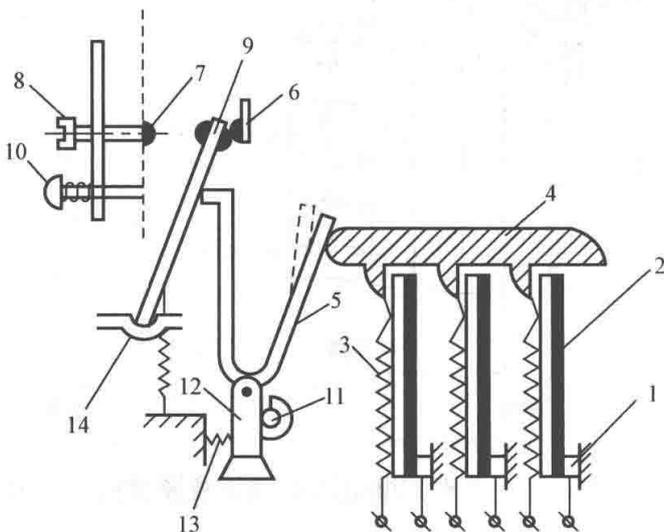


图 1-3 热继电器结构原理图

1—双金属片固定支点；2—双金属片；3—热元件；4—导板；5—补偿双金属片；6—动断触点；7—动合触点；8—复位螺钉；9—动触点；10—复位按钮；11—调节旋钮；12—支撑件；13—弹簧；14—推杆

图 1-4 为热继电器的图形符号。

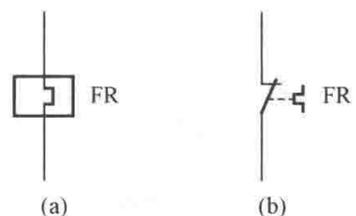


图 1-4 热继电器的图形符号

(a) 热元件；(b) 动断触点

#### \* 1.1.4 时间继电器

时间继电器是一种实现触点延时接通或断开的自动控制电器，其种类很多，常用的有电磁式、电子式、钟表式、电动机式时间继电器。时间继电器的延时动作，区别于一般的固有动作时间。

#### 1.1.5 速度继电器

速度继电器用来感受转速。它的感受部分主要包括转子和定子两大部分，执行机构是触

头系统。当被控电机转动时，带动继电器转子以同样速度旋转而产生电磁转矩，使定子克服外界反作用力转动一定角度，转速越高，角度越大。当转速高于设定值时，速度继电器的触点发生动作，当速度小于这一设定值时，触点又复原。速度继电器常用于电机的降压启动和反接制动，其图形符号如图 1-5 所示。

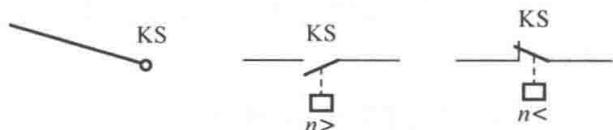


图 1-5 速度继电器图形符号

### 1.1.6 按钮

按钮是手动控制电器的一种，用来发出信号和接通或断开控制电路。图 1-6 是按钮的结构示意图和图形符号。

### 1.1.7 万能转换开关

万能转换开关用来选择工作状态，转换测信号回路，控制小容量电机。不同型号的万能转换开关，其手柄有不同的挡位（操作位置），其各触点的分合状态与手柄所处的挡位有关。万能转换开关的图形符号如图 1-7 所示。图中的万能转换开关具有 3 个挡位、5 对触点，在电路图中除了要

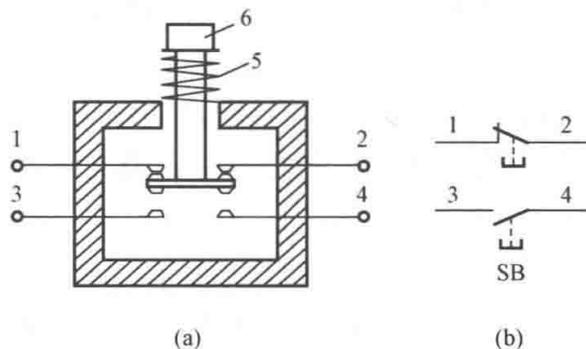


图 1-6 按钮的结构示意图及图形符号

(a) 结构示意图；(b) 图形符号

1、2—动断触点；3、4—动合触点；  
5—复位弹簧；6—按钮帽

画出相应触点外，还要标记出手柄位置（挡位）与触点分合状态的对应关系，有两种标记表示方法。一种是图形方法，用虚线表示挡位，而用有无实心点（·）表示触点在该挡位的分合状态，如图 1-7 (a) 所示；另一种方法是以表格的形式（称为接通表）描述手柄处于不同挡位时各触点的分合状态，如图 1-7 (b) 所示，表中符号“×”表示触点处于闭合状态。

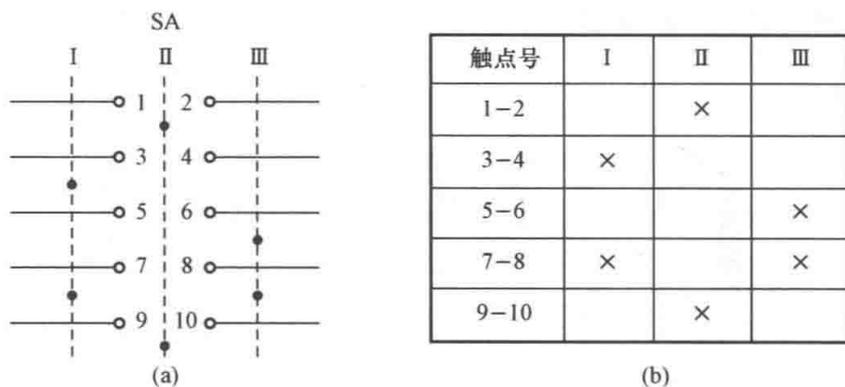


图 1-7 万能转换开关的图形符号及标记表示方法

(a) 图形方法；(b) 表格方法

### 1.1.8 接近开关

接近式位置开关是一种非接触式位置开关，简称接近开关。它由感应头、高频振荡器、放大器和外壳组成。当运动部件与接近开关的感应头接近时，就使其输出一个电信号。

接近开关分为电感式和电容式两种。

电感式接近开关的感应头是一个具有铁氧体磁芯的电感线圈，只能用于检测金属体。振荡器在感应头表面产生一个交变磁场，当金属块接近感应头时，金属中产生的涡流吸收了振荡的能量，使振荡减弱以至停振，因而产生振荡和停振两种信号，经整形放大器转换成二进制的开关信号，从而起到“开”、“关”的控制作用。

电容式接近开关的感应头是一个圆形平板电极，与振荡电路的地线形成一个分布电容，当有导体或其他介质接近感应头时，电容量增大而使振荡器停振，经整形放大器输出电信号。电容式接近开关既能检测金属，又能检测非金属及液体。

常用的电感式接近开关型号有 L11、LJ2 等系列, 电容式接近开关型号有 LXJ15、TC 等系列。

### 1.1.9 红外线光电开关

红外线光电开关分为反射式和对射式两种。

#### 1. 反射式光电开关

反射式光电开关是利用物体对光电开关的红外线反射回去, 由光电开关接收, 从而判断是否有物体存在。如果有物体存在, 光电开关接收到红外线, 其触点动作, 否则其触点复位。

它有三根连接线, 分别连接直流电源的正极、负极、OUT (输出信号), 如图 1-8 所示, 当与挡块接近时输出电平为低电平, 否则为高电平。需要注意检测距离不要离光电开关太近, 否则光电开关不能动作。

#### 2. 对射式光电开关

对射式光电开关由分离的发射器和接收器组成。当无遮挡物时, 接收到发射器发出的红外线, 其触点动作; 当有物体挡住时, 接收器便接收不到红外线, 其触点复位。

对射式光电开关的输出状态一般为 NPN 输出, 输出晶体管的动作状态可分为入光时 ON 和遮光时 ON 两种。入光时为 ON 的对射式光电开关的结构如图 1-9 所示, 当 24V 电压加到发光二极管 LED1 时, 它将光发射给发光二极管 LED2, LED2 接收到光导通, 三极管导通, 输出为 ON; 当发光二极管 LED1 发射出的光被物体挡住使发光二极管 LED2 接收不到时, LED2 不导通, 三极管也不导通, 输出为 OFF。

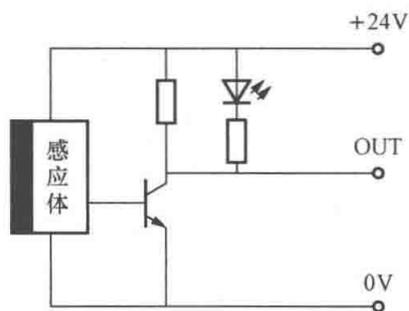


图 1-8 反射式光电开关原理图

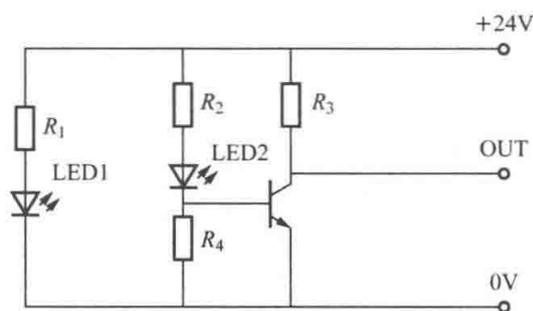


图 1-9 对射式光电开关原理图

光电开关和接近开关的用途已远超出一般行程控制和限位保护, 可用于高速计数、测速、液面控制、检测物体的存在、检测零件尺寸等许多场合。

此外, 继电器接触器控制系统还有行程开关、机械式凸轮开关、微动开关、干簧管开关、压力开关、液位开关、物位开关等, 这些电器元件的图形符号可参见有关的低压电器设备手册。

## 1.2 继电器控制系统常用的一些概念及基本控制方式

### 1.2.1 点动控制

#### 1. 线路设计思想

点动, 顾名思义, 点一下, 动一下, 不点则不动。即要求按下启动按钮后, 电动机启动运转, 松开按钮时, 电动机就停止转动, 点动控制也叫短车控制或点车控制。

主电路由刀开关 Q、熔断器 FU1、接触器主触点 KM、热继电器 FR 和三相电动机组成。控制回路包括熔断器 FU2、按钮、接触器线圈和热继电器动断触点，如图 1-10 所示。

### 2. 三相异步电动机点动控制电路的动作流程

电动机启动时，按下按钮 (SB) → 线圈 (KM) 通电 → 触头 (KM) 闭合 → 电动机转动。

电动机停车时，按钮松开 → 线圈 (KM) 断电 → 触头 (KM) 打开 → 电动机停止。

### 1.2.2 长动控制

#### 1. 线路设计思想

长动，又称连动，即控制对象能够持续运转，即使松开启动按钮后，吸引线圈通过其辅助触点仍保持继续通电，维持吸合状态。这个辅助触点常称为自锁触点。

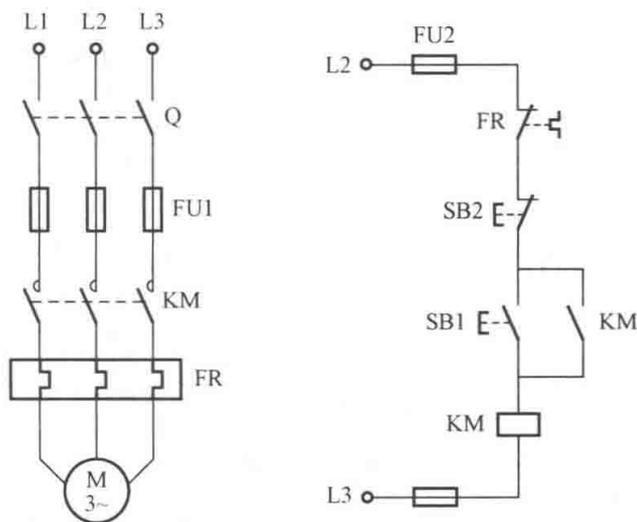


图 1-11 三相异步电动机的长动控制电路

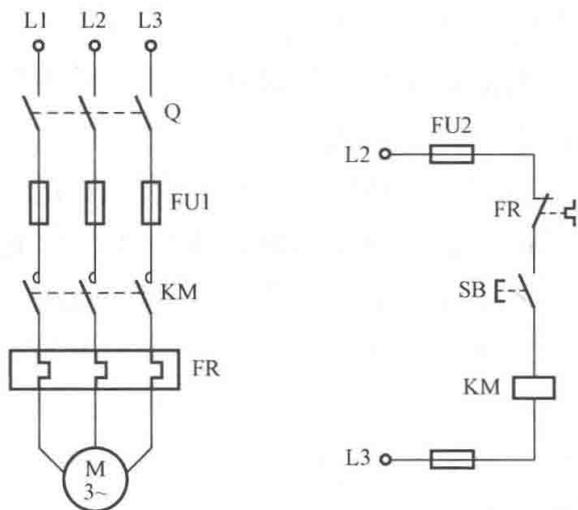


图 1-10 三相异步电动机的点动控制电路

#### 2. 典型电路设计

如图 1-11 所示，合上开关 Q，按下启动按钮 SB1，控制电路中接触器 KM 的线圈得电，在主电路中，接触器的主触点 KM 闭合，电动机得电启动，在控制电路中的接触器的辅助动合触点 KM 闭合，虽然 SB1 可能已经松开了，但接触器线圈通过辅助触点得以继续供电，从而维持其吸合状态。也就是说，由于自锁触点的存在，使得电动机启动后，松开启动按钮，电动机仍可继续运行。当电动机正常运行时，按下停止按钮 SB2，接触器线圈 KM 失电，接触器主触点断开，电动机停止运行；同时接触器的辅助触点也断开，自锁功能丧失，电路恢复至初始状态。

上述控制电路也称为电动机的启、保、停控制电路。

上述控制电路也称为电动机的启、保、停控制电路。

#### 3. 点动控制与长动控制的区别

点动控制与长动控制的区别主要在于自锁触点的设置。点动控制电路没有自锁触点，同时点动按钮兼起停止按钮的作用；而长动控制电路，必须设有自锁触点，并另设停止按钮。

### 1.2.3 点动+长动复合控制

#### 1. 线路设计思想

在工程应用中，单一点动控制电路或长动控制电路使用场合十分受限，实际的控制电路往往要求既能实现点动控制，又能实现连续运行的复合电路。有鉴于此，在控制电路设计时，要想实现点动+长动复合控制，必须根据点动控制与长动控制线路的区别，着重强调对自锁触点的处理。

如图 1-12 所示，在 KM 的动合触点回路上串接点动按钮 SB3 的动断触点。

## 2. 三相电动机的点动+长动复合控制电路工作原理

当需要电动机连续运行时,按下长动按钮 SB1,电动机通电启动运转。欲使电动机停转,按下停止按钮 SB2 即可。

当需要电动机点动控制时,按下点动按钮 SB3,电动机通电启动运转。由于按钮 SB3 断开了接触器的自锁回路,故松开 SB3 时电动机断电停止运转。

值得注意的是,图 1-12 所示的控制电路存在一定的隐患。如果接触器 KM 的释放时间大于按钮 SB3 的恢复时间,则松开按钮 SB3 后,SB3 的动断触点先闭合,而 KM 的辅助动合触点尚未断开,将会使 KM 的自锁回路起作用,点动控制无法实现。这种现象称为触点间的“竞争”。存在竞争的电路工作是不可靠的,所以在设计控制电路时应尽可能避免竞争现象的发生。

解决竞争现象常用的方法是引入中间继电器,控制电路如图 1-13 (a) 所示。长期工作时中间继电器 KA 线圈得电并自锁,同时使接触器 KM 吸合;点动控制时按下 SB3,由于 KM 无法自锁,因此电路可以可靠地实现点动控制。

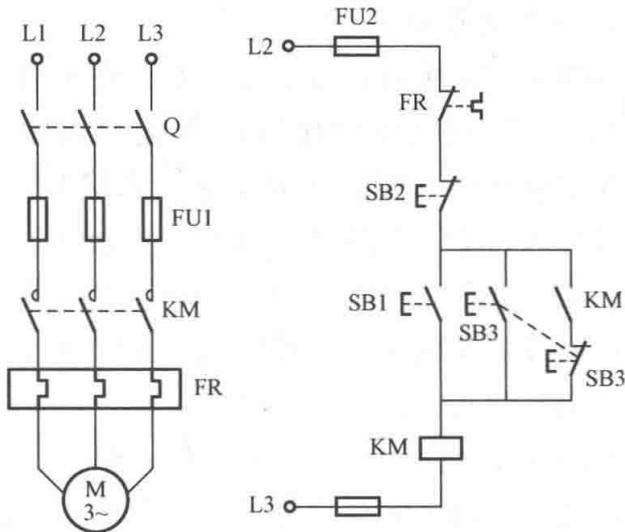


图 1-12 三相电动机的点动+长动复合控制电路

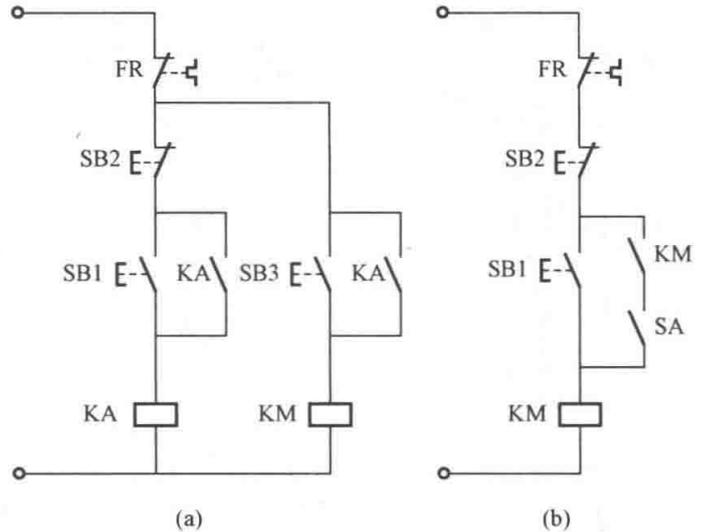


图 1-13 点动与长动结合的控制电路

图 1-13 (b) 所示的电路也能可靠地实现点动与连续运行。手动开关 SA 断开时,按动按钮 SB1 即可实现点动控制;当 SA 闭合时,KM 的自锁触点被接入,电路可方便地实现连续运行。

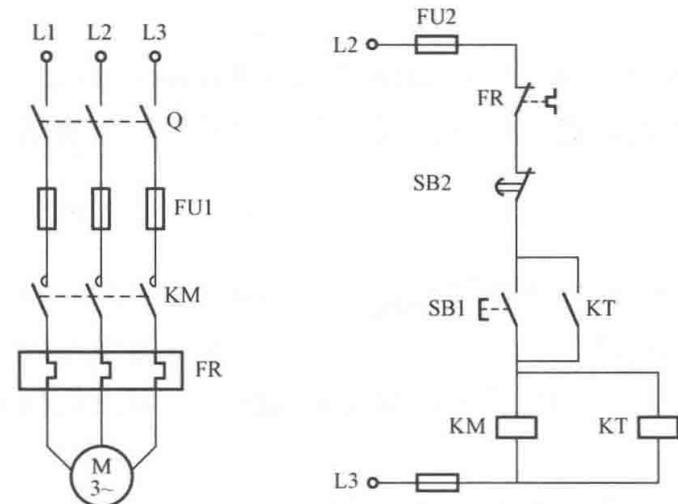


图 1-14 三相异步电动机的按钮延时控制电路

### 1.2.4 点动+延时复合控制

在工程应用中,常有一些用按钮起到延时一段时间又自动关闭的控制应用,这时可以用一个按钮和一个延时时间继电器来实现。在控制电路设计时,要想实现点动+延时复合控制,就要求时间继电器有一个瞬动的动合触点和一个延时断开的动断触点,着重强调对自锁动合触点的处理,如图 1-14 所示。

### 1.2.5 多地控制

有些生产机械和设备常要求可以在两

个或两个以上的地点进行启、停控制，称为多地控制或多点控制。多地控制要求在每个地点都装有启动按钮和停止按钮。若要求在任一地点按下启动按钮电动机均能启动，则应将所有的启动按钮（动合触点）并联起来（逻辑或）；若要求在任一地点按下停止按钮电动机均能停止，则应将所有的停止按钮（动断触点）串联起来（逻辑与）。

### 1.2.6 正反向接触器的联锁控制

许多生产机械在工作过程中常要求具有上下、左右、前后、往返等相反方向的运动，这就要求能对电动机进行正反转控制。可以通过控制接触器改变定子绕组相序来实现。应用在要求两个控制对象不能同时处于动作状态的场合。

典型的控制电路如图 1-15 所示。该电路由两个基本的启、保、停电路构成，此外在控制电路中附加了一定的联锁条件，将 KM1（或者 KM2）的辅助动断触点串入接触器 KM2（或者 KM1）的控制电路中，这就能够保证在任何时刻只能有一个接触器工作。

可以避免因正反两个接触器同时工作而造成的短路事故，这种方法称为互锁或联锁控制。

为了能够实现直接正反转控制，可以采用带按钮联锁的正反转控制电路。如图 1-15 所示，电动机正转时，无需按下停止按钮 SB1、而直接按下反转启动按钮 SB3，即可实现电动机由正转向反转的切换。

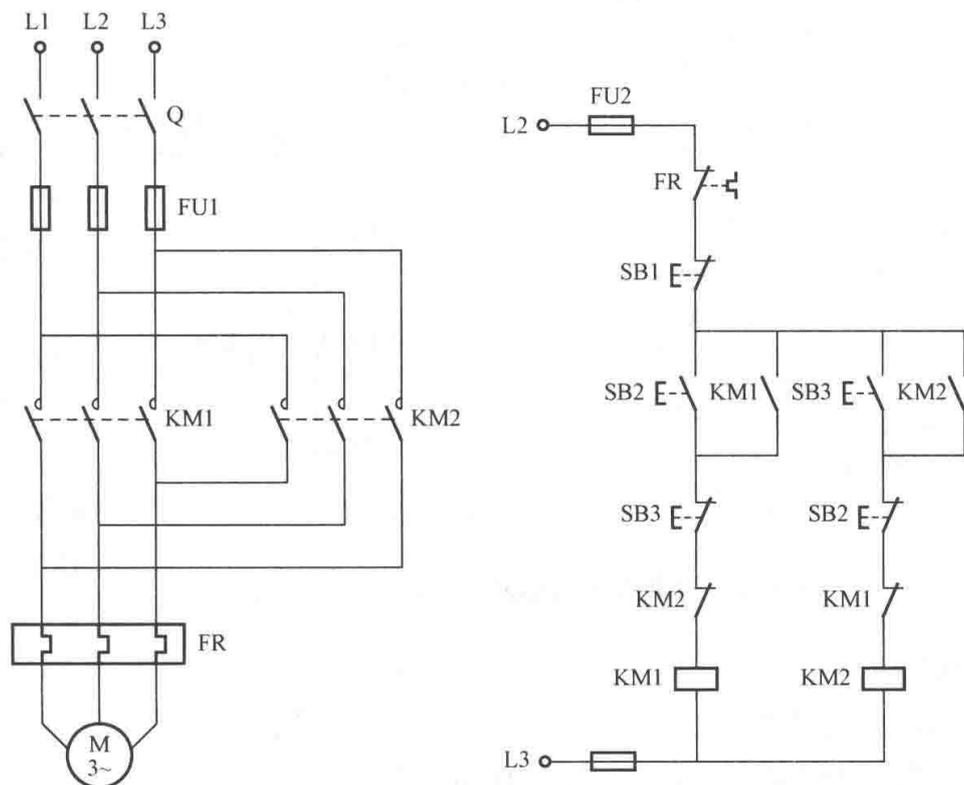


图 1-15 正反联锁控制原理图

为了能够实现直接正反转控制，可以采用带按钮联锁的正反转控制电路。如图 1-15 所示，电动机正转时，无需按下停止按钮 SB1、而直接按下反转启动按钮 SB3，即可实现电动机由正转向反转的切换。

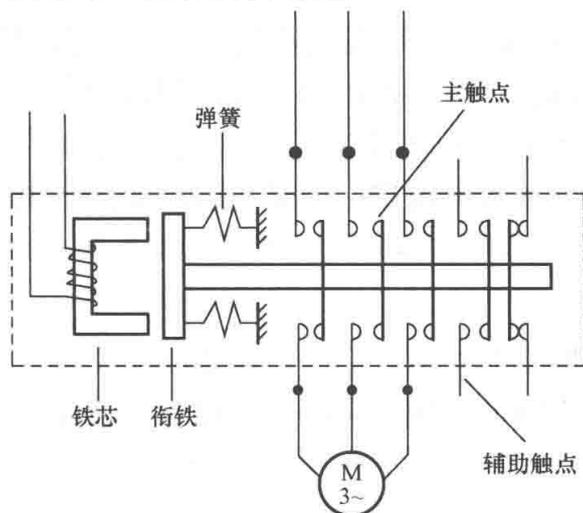


图 1-16 失压保护控制的动作示意图

### 1.2.7 三相异步电动机启动控制电路的相关保护

#### 1. 失压保护

采用接触器实现保护控制，如图 1-16 所示。

保护原理：在线圈上施加交流电压后，铁芯中产生磁通，对衔铁产生克服弹簧拉力的电磁吸力，使衔铁带动触点动作。当线圈中电压的值降到电源电压值的 85% 时，铁芯中的磁通下降，吸力减小到不足以克服弹簧的反力时，衔铁就在弹簧的作用下复位。

失压保护控制的动作流程：

线圈通电→衔铁被吸合→触点闭合→电动机接通电源。

线圈失电→衔铁被断开→触点复位→电动机脱离电源。

当电源电压消失而又重新恢复时,要求所有的电动机或负载均不能自行启动,以确保操作人员和设备的安全。由于控制电路中采用的接触器的自锁触点能够保证这一点,在电源电压消失而又重新恢复时,接触器的线圈也不能得电,电动机也就不能再启动,除非再次按下启动按钮。

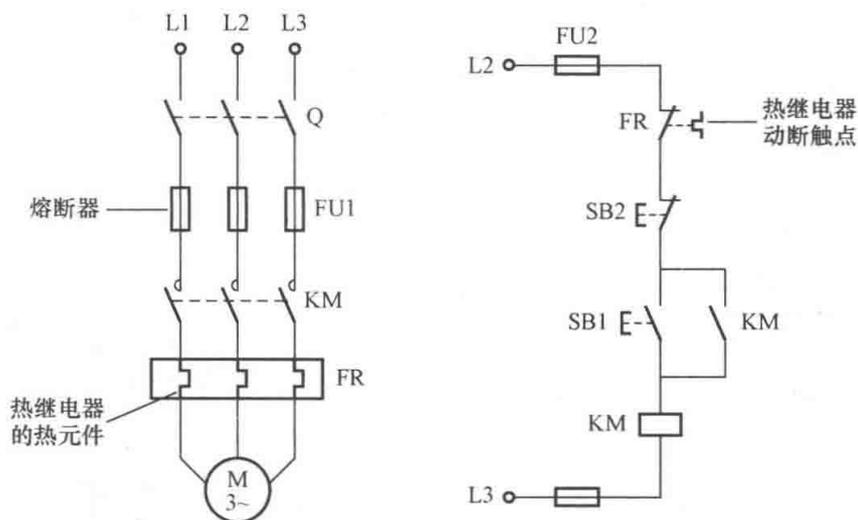


图 1-17 过载保护及熔断器保护控制的电路

## 2. 短路保护

采用熔断器实现保护控制。

保护原理:操作时熔断器串接于被保护电路中,当电路发生严重过载或短路时,利用电流的热效应原理,熔体熔断而切断电路,实现短路保护(见图 1-17)。

短路保护控制的注意事项:异步电动机的启动电流  $I_{st}$  约为额定电流  $I_N$  的 5~7 倍。选择熔体额定电流  $I_{RT}$  时,必须躲开启动电流,但对短路电流仍能起保护作用。

保护作用。

## 3. 过载保护

采用加装热继电器实现保护控制。

保护原理:发热元件接入电机主电路,若长时间过载,双金属片被烤热。因双金属片的下层膨胀系数大,使其向上弯曲,扣板被弹簧拉回,动断触头断开,从而切断电路,实现过载保护。

## \* 1.3 电液控制

自动化设备或系统中,除了大量使用电动机来实现控制对象的各种动作外,还广泛使用液压缸或气缸来实现各种动作控制,电液(或气压)控制系统是现代机电设备中,特别是机、电、液(或气体)一体化设备中一个重要的组成部分,电液控制也是控制领域中的一个重要的分支。电磁换向阀是用电磁的效应进行控制的,主要的控制方式是继电器控制时,本节主要讲解电液控制中常用的电磁换向阀的工作原理及其基本应用,气动控制的原理与电液控制类似。

### 1.3.1 电磁换向阀

电磁换向阀是用来控制流体的自动化基础元件,属于执行器,它并不限于液压控制,也广泛应用于气动控制,用于控制液流(或气体)流动方向,实现运动换向,以及接通或关断油路。

电磁阀中有个密闭的腔,在不同位置开有通孔,每个孔连接不同的油管,腔中间是活塞,一边或者两边有电磁铁,哪边的电磁铁线圈通电阀体就会被吸引到那边,通过控制阀体

的移动来开启或关闭不同的排油孔，而进油孔是常开的，液压油就会进入不同的排油管，然后通过油的压力来推动油缸的活塞，活塞又带动连接有机械装置的活塞杆。

图 1-18 为电磁换向阀的结构原理及通道工作情况。符号中方格表示滑阀的位，图 1-18 中 (a) ~ (c) 三个阀为二位；(d)、(e) 两个阀为三位，箭头表示阀内液流方向，符号“⊥”表示阀内通道堵塞。电磁阀有交流电磁阀和直流电磁阀两种，以电磁铁所用电源而定。

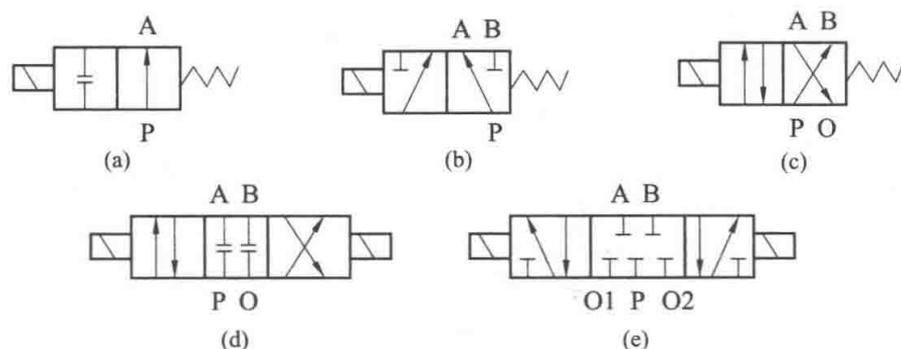


图 1-18 电磁换向阀的结构原理及通道工作情况  
(a) ~ (c) 二位；(d)、(e) 三位

以图 1-18 中的 (c) 为例说明其工作原理，图中阀口 P 为压力油口（进油口），阀口 O 为回油口。A、B 为工作油口，接液压缸右、左两个腔。靠近弹簧符号的方框为电磁换向阀的常态，即线圈断电时的状态；靠近线圈符号的方框是线圈得电时的状态。图中所示电磁换向阀，当电磁铁断电时阀口 P 与 B 通，A 与 O 通；当电磁铁得电时，P 与 A 通，B 与 O 通，即改变了压力油进入液压缸的方向，实现了油路的换向，也就改变了油缸的动作方向。

### 1.3.2 电液控制的应用举例

液压动力头是既能完成进给运动，又能同时完成刀具切削运动的动力部件。液压动力头的自动工作循环是由控制线路控制液压系统来实现的。图 1-19 是动力头工作进给的液压系统和电气控制线路图，实现动力头快进→工作进给→快速退回（或延时再快速退回）到原位。图中 1U、2U 为油过滤器，L 为溢流阀。

下面分析自动工作循环的控制原理，在原位时 SQ1 动合触点闭合，动断触点断开。当转换开关处于“1”时，按启动按钮 SB1，中间继电器 KA1 得电→电磁铁 YA1、YA3 通电→电磁阀 YV1 推向左端、动力头向前运动，又由于 YA3 得电，电磁阀 YV2 将油压缸右腔中的回油排入左腔，加大了油的流量，因此动力头可以快速向前移动；当动力头移动到限位开关 SQ2 时，KA2 得电动作并且自保持，KA2 的动断触点断开了 YA3，动力头就自动转换为工进状态；当工作进给达到终点位置 SQ3 时，KA3 得电动作并且自保持，KA3 的动断触点断开，YA1 断电，停止进给，KA3 的动断触点闭合，YA2 得电，电磁阀 YV1 右移，动力头将自动转换为快退状态；当动力头退回到原位 SQ1 时，KA3 断电，YA2 也断电，动力头停止在原位处。

当转换开关处于“3”时，可以实现对液压动力头的点动调整控制（或手动控制），另外，当动力头不在原位需要快退调回原位时，可按动按钮 SB2 调整动力头到原位。

在上述控制线路的基础上，增加一个时间继电器 KT，适当改变控制线路，就能得到延时停留再返回的自动工作循环方式：快进→工进→延时停留→快退。限于篇幅，控制线路本书从略。

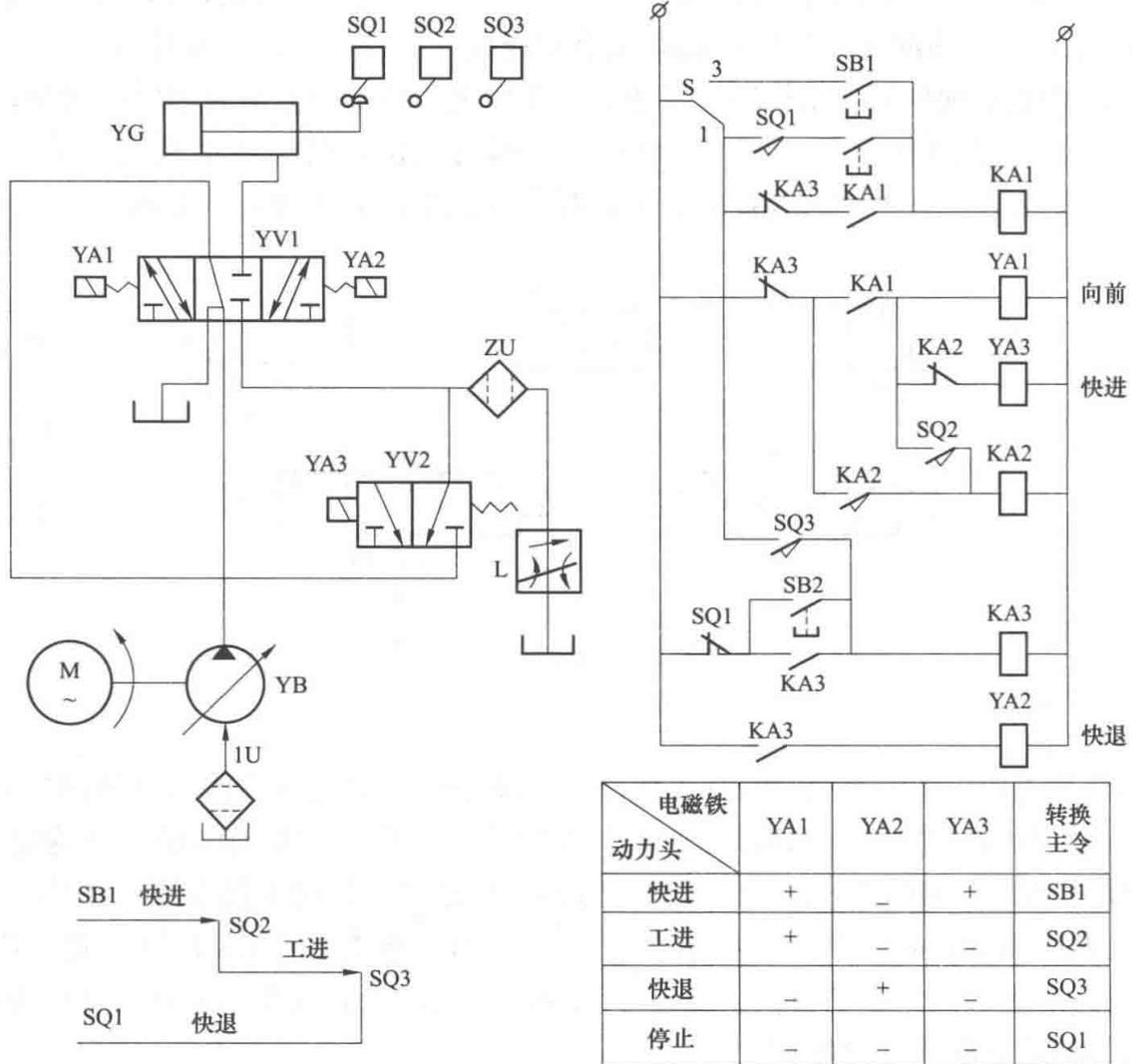


图 1-19 动力头工作进给的液压系统和电气控制线路图

### 习 题



1. 自动控制线路中常设置哪几种保护? 过载保护与短路保护有什么区别? 各用什么电器实现?

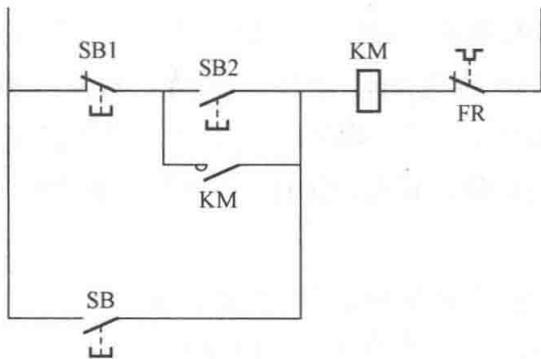


图 1-20 题 4 的图

2. 什么叫互锁? 它有何作用?
3. 什么叫自保? 它有何作用?
4. 图 1-20 所示控制电路能否实现既能点动又能长动连续运行? 试分析原因。
5. 继电器和接触器有什么区别?
6. 在电动机的正反向接触器的联锁控制中, 有了按钮的机械互锁后, 电气互锁是否能省去? 为什么?

## 第2章 可编程控制器概述

### 2.1 可编程控制器的产生及定义

#### 2.1.1 可编程控制器的产生

1968年由美国通用汽车公司（GE）提出了研制可编程序控制器（即可编程控制器）的基本设想，希望尽量减少重新设计和更换继电器控制系统的硬件和接线，减少系统维护和升级时间，降低成本。希望将计算机的优点与继电器控制系统简单易懂、操作方便、价格便宜等优点相结合，设计一种通用的控制装置来满足生产需求。

1969年由美国数字设备公司（DEC）研制成功世界上第一台可编程控制器，有逻辑运算、定时、计算功能，称为PLC（Programmable Logic Controller）。

1980年后，由于计算机技术的发展，PLC采用通用微处理器为核心，功能扩展到各种算术运算，PLC运算过程控制并可与上位机通信，实现远程控制。

#### 2.1.2 可编程控制器的定义

国际电工委员会（IEC）1985年颁布的可编程逻辑控制器的定义如下：“可编程逻辑控制器是专为在工业环境下应用而设计的一种数字运算操作的电子装置，是带有存储器、可以编制程序的控制器。它能够存储和执行命令，进行逻辑运算、顺序控制、定时、计数和算术运算等操作，并通过数字式和模拟式的输入输出，控制各种类型的机械或生产过程。可编程控制器及其有关的外围设备，都应按易于工业控制系统形成一个整体、易于扩展其功能的原则设计”。

可编程控制器是一种用程序来改变控制功能的工业控制计算机，它是以微处理器为基础的通用工业控制装置。

可编程控制器最初是用于替代继电器控制系统的新型控制器，现在的PLC功能更加完善，除了开关逻辑控制的场合能够大显身手外，在要求有模拟量闭环控制的场合，也不会比单片机逊色。单片机能够完成的工作PLC都能完成，而且PLC更适用工业生产现场环境，具有更高的可靠性及较好的电磁兼容性。

### 2.2 PLC的组成

尽管PLC种类繁多，有着不同的结构和分类，但其基本组成是相同的。都是由中央处理单元（CPU）、存储器、输入输出单元（I/O单元）、电源单元、编程器等组成，如图2-1所示。

#### 2.2.1 中央处理单元

与普通计算机一样，CPU是系统的核心部件，是由大规模或超大规模的集成电路微处理器芯片构成的，主要完成运算和控制任务，可以接收并存储从编程器输入的用户程序和数

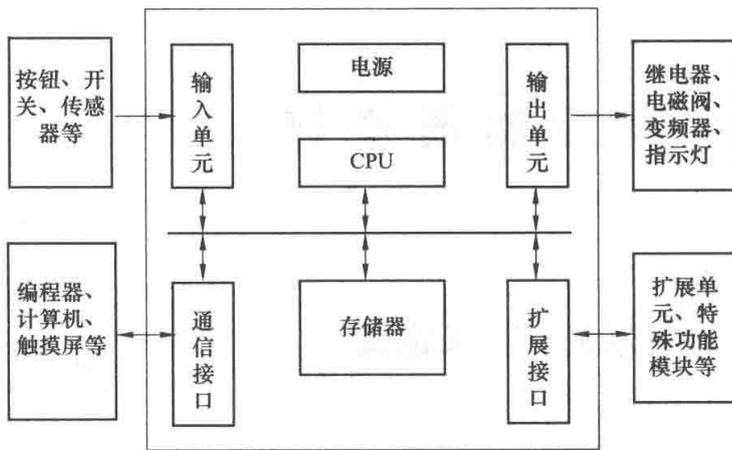


图 2-1 PLC 组成原理图

据。进入运行状态后，用扫描的方式接收输入装置的状态或数据，从内存逐条读取用户程序，通过解释后按指令的规定产生控制信号。分时、分渠道地执行数据的存取、传送、比较和变换等处理过程，完成用户程序设计的逻辑或算术运算任务，并根据运算结果控制输出设备。PLC 中的中央处理单元多用 8~32 位字长的单片机。

### 2.2.2 存储器

存储器包括系统存储器和用户存储器。系统存储器存放系统管理程序。用户存储器存放用户编制的控制程序。

按照物理性能，存储器可以分为两类，即随机存储器 (RAM) 和只读存储器 (ROM)。

随机存储器由一系列寄存器阵组成，每位寄存器可以代表一个二进制数，在刚开始工作时，它的状态是随机的，只有经过置“1”或清“0”的操作后，它的状态才确定。若关断电源，状态丢失。这种存储器可以进行读、写操作，主要用来存储输入输出状态和计数器、定时器以及系统组态的参数。只读存储器有两种。一种是不可擦除 ROM，这种 ROM 只能写入一次，不能改写；另一种是可擦 ROM，这种 ROM 经过擦除以后还可以重写。其中 EPROM 只能用紫外线擦除内部信息，EEPROM 可以用电擦除内部信息，这两种存储器的信息可保留 10 年左右。

对于不同的 PLC，其存储器的容量随 PLC 的规模不同而有较大的差别，大型 PLC 的用户程序存储器容量一般大于 40kB，而小型 PLC 的容量多小于 8kB，用户程序存储器容量的大小，关系到用户程序容量的大小和内部软元件的多小，是反映 PLC 性能的重要指标之一。

### 2.2.3 输入输出单元 (I/O)

输入输出单元通常也叫 I/O 单元或 I/O 模块，是 PLC 与被控对象间传递输入输出信号的接口部件。输入部件是开关、按钮、传感器等，PLC 通过输入接口可以检测被控对象的各种数据，以这些数据作为 PLC 对被控对象进行控制的依据。输出部件是指示灯、电磁阀、接触器、继电器、变频器等，PLC 通过输出接口将处理结果送给被控对象，以实现控制目的。

#### 1. 输入接口电路

通常 PLC 的输入类型可以是直流、交流和交直流。输入电路的电源可由外部供给，有的也可由 PLC 内部提供。图 2-2 为 PLC 的直流输入接口电路图，图 2-3 为 PLC 的交流输入接口电路图，采用的是外接电源。

图 2-4 描述了一个输入点的接口电路。其输入电路的一次电路与二次电路用光耦合器相连，当行程开关闭合时，输入电路和一次电路接通，上

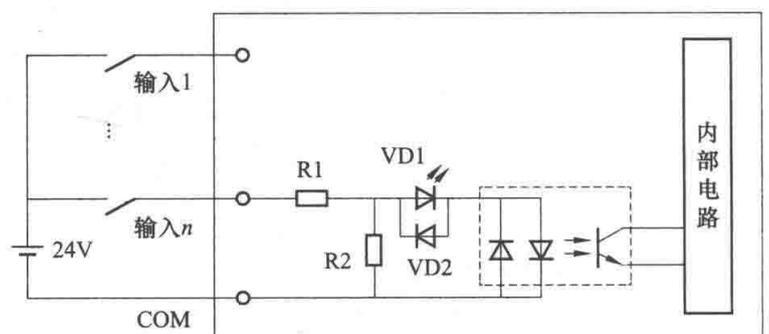


图 2-2 PLC 的直流输入接口电路图

面的发光管用于对外显示，同时光耦合器中的发光管使三极管导通，信号进入内部电路，此输入点对应的位由0变为1，即输入映像寄存器的对应位由0变为1。

2. 输出接口电路

PLC 输出电路用来驱动被控负载（电磁铁、继电器、接触器线圈等）。PLC 输出电路结构形式分为继电器型

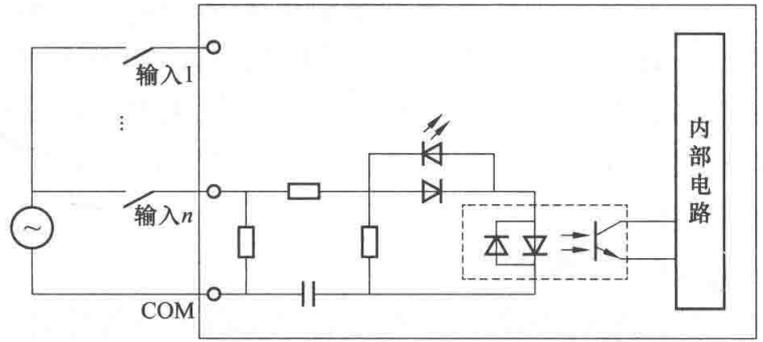


图 2-3 PLC 的交流输入接口电路图

式、晶体管型式、晶闸管型式三种。

继电器型输出电路如图 2-5 (a) 所示。内部电路使继电器的线圈通电，它的动合触点闭合，使外部负载得电工作。继电器同时起隔离和功率放大的作用，每一路只给用户仅提供一对动合触点。与触点并联的 RC 电路和压敏电阻用来消除触点断开时产生的电弧，以减轻它对 CPU 的干扰。继电器型输出电路的滞后时间一般在 10ms 左右。

晶体管集电极输出电路如图 2-5 (b) 所示。各组的公共点接外部直流电源的负极。输出信号送给内部电路中的输出锁存器，再经光耦合器送给输出晶体管，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，以保护晶体管，晶体管输出电路的延迟时间小于 1ms。场效应晶体管输出电路的结构与晶体管输出电路基本相同。

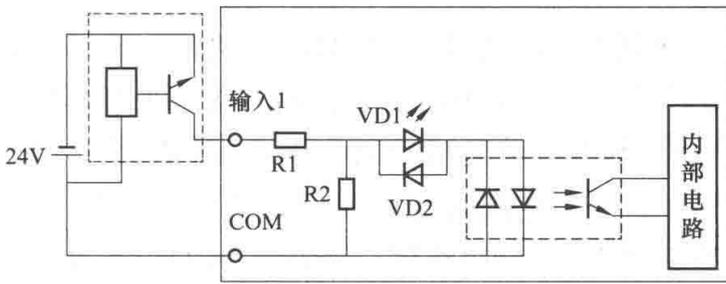


图 2-4 输入端为无触点开关 (NPN) 型的电路图

继电器型输出电路如图 2-5 (a) 所示。内部电路使继电器的线圈通电，它的动合触点闭合，使外部负载得电工作。继电器同时起隔离和功率放大的作用，每一路只给用户仅提供一对动合触点。与触点并联的 RC 电路和压敏电阻用来消除触点断开时产生的电弧，以减轻它对 CPU 的干扰。继电器型输出电路的滞后时间一般在 10ms 左右。

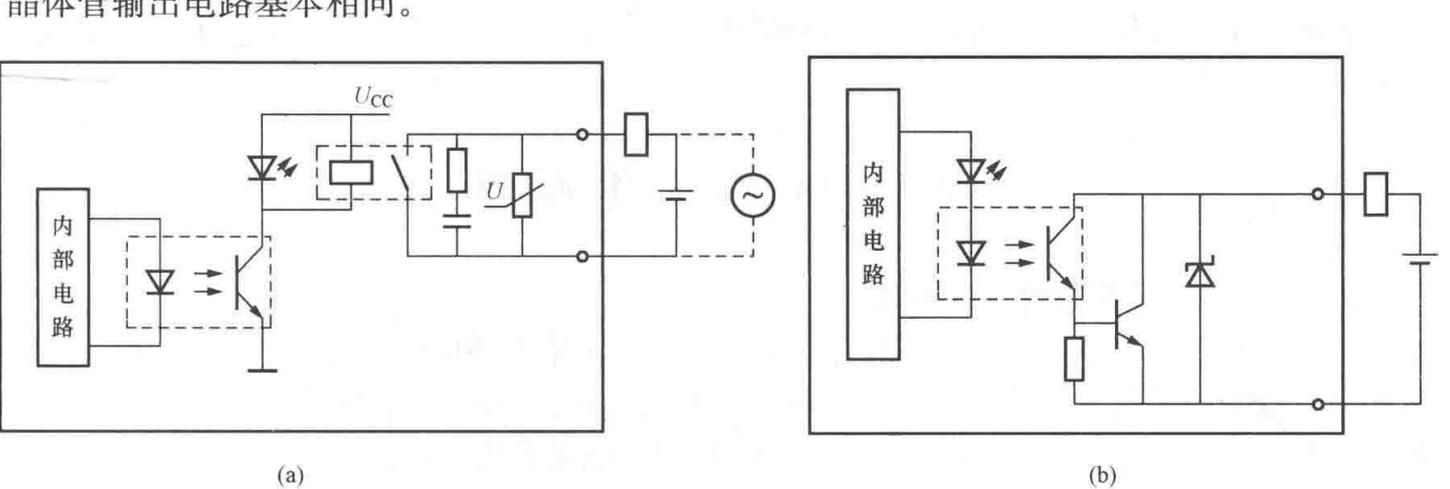


图 2-5 PLC 的开关量输出电路原理图

(a) 继电器型输出电路；(b) 晶体管集电极输出电路

除了上述两种输出电路外，还有双向晶闸管输出电路，它用光敏晶闸管实现隔离。双向晶闸管由关断变为导通的延迟时间小于 1ms，由导通变为关断的最大延迟时间小于 10ms。

2.2.4 电源

PLC 的交流输入一般为单相交流 (AC85~260V, 50/60Hz)，有的也采用直流 24V 电

源, PLC 对外部工作电源的稳定度要求不高, 一般可允许 $\pm 15\%$ 的波动范围, 抗干扰能力比较强。有些 PLC 还配有大容量电容作为数据后备电源, 停电时可以保持 50h。使用单相交流电源的 PLC, 其内部配有开关式稳压电源, 该电源可以向 CUP、存储器、I/O 模块提供 DC5V 工作电源, 在容量许可的条件下, 还可同时向外部提供 DC24V 电源, 供直流输入或输出使用。

### 2.2.5 编程器

编程器是外围设备, 利用编程器将用户程序送入 PLC 的存储器, 检查程序。编程器一般由 PLC 生产厂家提供, 且只能用于某个品牌、某个系列的 PLC。

编程器主要分为专用编程器和专用编程软件两类。

#### 1. 专用编程器

用于编制特定 PLC 软件的编程装置, 分为简易编程器和图形编程器两种。简易编程器只能编辑语句表指令程序, 不能直接编辑梯形图程序, 使用简易编程器时必须把梯形图程序先转化为语句表指令程序。因此, 简易编程器一般用于小型 PLC 的编程, 或者用于 PLC 控制系统的现场调试和维修。图形编程器本质上是一台便携式专用计算机系统, 具有 LCD 或 CRT 图形显示功能, 用户可以在线或离线地编制 PLC 应用程序, 所能编辑的也不再局限于语句表指令, 可直接使用梯形图编程。

#### 2. 专用编程软件

除了专用编程器以外, 各 PLC 厂家都提供了能在 PC 上运行的专用编程软件, 借助于相应的通信接口, 用户可以在 PC 上通过专用编程软件来编辑和调试用户程序, 而且专用编程软件一般可适应于同一厂商的多种型号 PLC。专用编程软件具有功能强大、通用性强、升级方便、价格低廉等特点, 在个人计算机和便携式电脑日益普及的情况下, 是用户首先的编程装置。

### 2.2.6 其他外围设备

其他外围设备包括人机接口装置 (HMI)、存储器卡、打印机、盒式磁带机、EPROM 写入器等。

## 2.3 PLC 的工作原理

### 2.3.1 PLC 控制系统等效电路

PLC 是从继电器控制系统发展而来的。以图 1-11 中电动机的控制电路为例, 该电路是以触点、线圈的组合实现启、保、停控制及热继电器保护控制, 用 PLC 实现等效的电气控制逻辑非常方便, 如图 2-6 所示, 可以将 PLC 等效电路分成 3 部分, 即输入部分、内部程序执行部分和输出部分。它的梯形图程序与继电器系统电路图相似, 所以梯形图中的某些编程元件也沿用了继电器这一名称。

#### 1. 输入部分

输入部分由外部输入电路、输入端子和输入继电器 (内部软元件) 组成, 每个外部输入信号经由输入端子驱动相同编号的输入继电器。当外部信号状态为“1”时 (外部触点处于闭合状态), 对应的输入继电器状态为“1”, 程序中与该元件对应的动合触点闭合 (状态为“1”)、动断触点断开 (状态为“0”)。同理, 当外部输入信号断开时, 相应的输入继电器

状态为“0”，程序中与之对应的动合触点状态为“0”、动断触点状态为“1”。

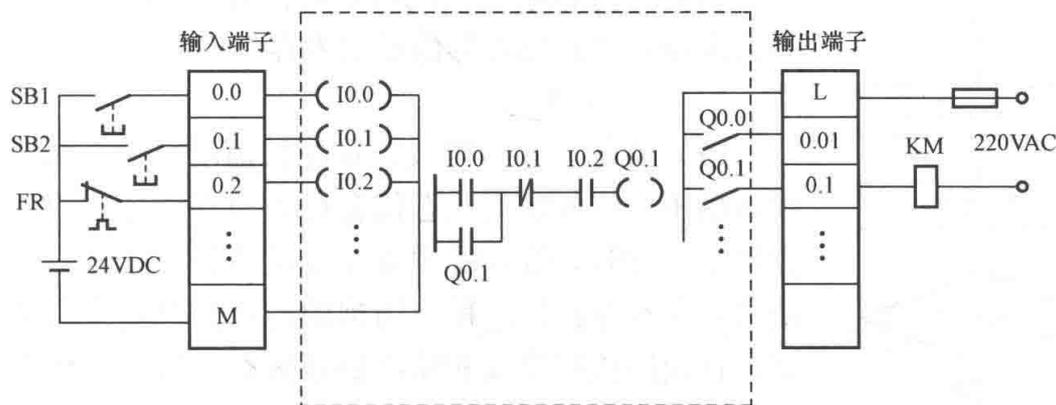


图 2-6 用 PLC 实现等效电气控制的电路

输入部分的主要功能就是检测各外部输入信号的状态，并将结果存放到输入映像寄存器中。

### 2. 内部程序执行过程

PLC 的程序是电气控制线路的软件实现，程序的执行过程相当于电气控制的逻辑运算过程。PLC 按照用户程序规定的逻辑关系，对输入信号和输出信号的状态进行检测、判断、运算和处理，得到相应的输出，这些输出同样对应于 PLC 的某些内部软元件。

### 3. 输出部分

以继电器输出形式为例，输出部分由 PLC 内部输出继电器动合触点、输出端子和外部输出电路组成，输出端子与同编号的输出继电器相对应，通过输出继电器动合触点的分合来驱动外部负载。

输出继电器的状态由程序执行结果决定。当内部软元件的逻辑运算结果为“1”时，相应的输出继电器得电，其动合触点闭合，接通外部负载电路，以驱动相应的控制电路。当逻辑运算结果为“0”时，相应的输出继电器的动合触点断开，切断外部驱动电路。

总之，PLC 是根据检测到的输入元件的状态及内部其他元件的顺序执行用户程序，然后将得到的结果输出至外部负载电路，以驱动相应的控制电器（继电器、接触器、电磁阀等）。

#### 2.3.2 扫描工作方式

PLC 有两种基本的工作模式，即运行和停止。可以由外设的钮子开关设置，也可以由编程软件设置，在程序编辑、修改、上载和下载时 PLC 应处在停止模式，在 PLC 执行控制时必须处在运行模式。运行和停止模式都有相应的状态指示灯指示。

PLC 采用循环扫描的工作方式，包括内部处理、通信操作、输入处理、程序执行、输出处理几个阶段，如图 2-7 所示。全过程扫描一次所需的时间称为扫描周期。

当处于 RUN 状态时，上述扫描周期不断循环。当处于 STOP 状态时，PLC 只完成内部处理和通信服务。

#### 1. 内部处理阶段

主要完成自检、自诊断及完成一些其他工作等，如检查 CPU 模块内部的硬件是否正常、将监控定时器复位等。

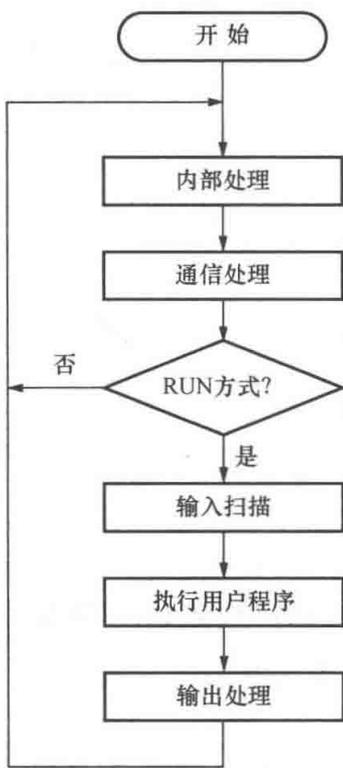


图 2-7 PLC 的循环扫描工作方式

### 2. 通信服务阶段

PLC 与其他的带微处理器的智能装置通信，响应编程器键入的命令，更新编程器的显示内容。

### 3. 输入处理

PLC 在输入采样阶段，首先扫描所有输入端子，包括未接线的端子，将各输入状态信息存入内存中各对应的输入映像寄存器中。此时，输入映像寄存器被刷新。接着，进入程序执行阶段，在程序执行过程中用到的输入信息均来自输入映像寄存器，在程序执行阶段和输出刷新阶段，输入映像寄存器与外界隔离，无论输入信号如何变化，其内容保持不变，直到下一个扫描周期的输入采样阶段，才重新写入输入端的新内容。

输入映像寄存器中的变量称为输入继电器，一般用 I 或 X 表示，如 I0.0、I0.1 和 X000、X001 等，其状态分为有输入（“ON”或“1”）和没有输入（“OFF”或“0”）两种，而且完全由外界的输入端决定，不能由程序改变其状态。

### 4. 程序执行

在程序执行阶段，根据 PLC 梯形图程序扫描原则，PLC 按先左后右、先上后下的步序语句逐句扫描。但遇到程序跳转指令，则根据跳转条件是否满足来决定程序的跳转地址。当指令中涉及输入、输出状态时，PLC 就从输入映像寄存器“读入”上一阶段采入的对应输入端子状态，从元件映像寄存器“读入”对应元件（软继电器）的当前状态。然后，进行相应的运算，运算结果再存入元件映像寄存器中。对元件映像寄存器来说，每一个元件（软继电器）的状态会随着程序执行过程而变化。

### 5. 输出处理

在输出刷新阶段，在所有指令执行完毕后，元件映像寄存器中所有输出继电器的状态（接通/断开）在输出刷新阶段转存到输出锁存器中，通过一定方式输出，驱动外部负载。

可以通过图 2-8 所示的例子来进一步分析 PLC 的工作原理。图中的程序是用梯形图来表示的，梯形图是一种软件，是 PLC 图形化的程序，但是实际上梯形图是以指令的形式储存在 PLC 的用户程序存储器中，图 2-8 中的梯形图程序与下面的 5 条指令相对应。

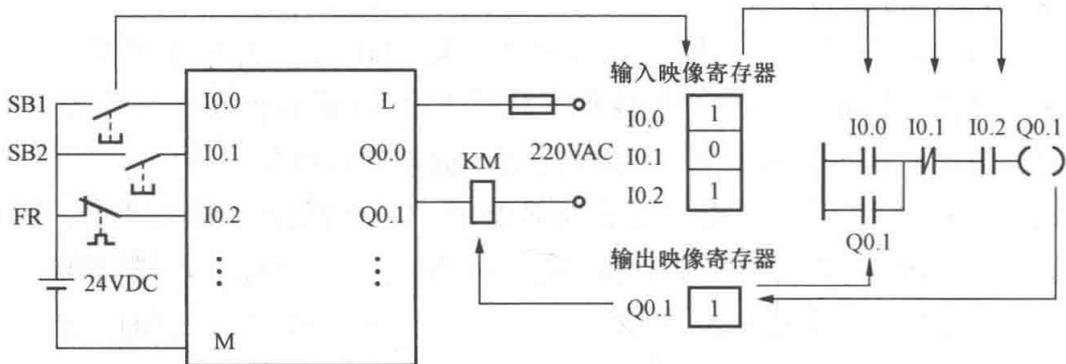


图 2-8 PLC 的外部接线及梯形图

```

LD I0.0 // 逻辑开始
O Q0.1 // 并联条件
AN I0.1 // 串联条件
  
```

```
A  I0.2      // 串联条件  
=  Q0.1      // 启动输出 1
```

在输入处理阶段，CPU 将 SB1、SB2 的动合触点、FR 的动断触点的状态读入相应的输入映像寄存器，外部触点接通时存入寄存器的是二进制，反之存入 0。

执行第 1 条指令时，从 I0.0 对应的输入映像寄存器中取出二进制数并保存起来。执行第 2 条指令时，取出 Q0.1 对应的输出寄存器中的二进制数，与 I0.0 对应的二进制数相“或”。运算结束后只保留运算结果，不保留取出来的参与运算的数据。

执行第 3 条指令时，取出 I0.1 对应的输入映像寄存器中的二进制数，因为是动断触点，先将取出的数取反，再与前面的运算结果相与，然后存入运算结果寄存器。第 4 条指令的执行与第 3 条指令的执行类似，只是取出的数不要取反而已。

执行第 5 条指令时，将运算结果寄存器的二进制数送入 Q0.1 对应的输出映像寄存器。

在输出处理阶段，CPU 将各输出映像寄存器的二进制数传送给输出模块并锁存起来，如果 Q0.1 对应的输出映像寄存器存放的是二进制数 1，外接的接触器的线圈将通电，反之将断电。

### 2.3.3 扫描周期

在 RUN 工作模式时，PLC 的工作方式是一个不断循环的顺序扫描工作方式。全过程扫描一次所需的时间称为扫描周期或工作周期。CPU 从第一条指令开始，按顺序逐条地执行用户程序直到用户程序结束，然后返回第一条指令开始新一轮扫描。PLC 就是这样周而复始地重复上述循环扫描的。

PLC 运行正常时，扫描周期的长短与 CPU 的运算速度、与 I/O 点的情况、与用户应用程序的长短及编程情况等均有关。通常用 PLC 执行 1K 指令所需时间来说明其扫描速度（一般 1~10ms/K）。值得注意的是，不同指令其执行时间是不同的，从零点几微秒到上百微秒不等，故选用不同指令所用的扫描时间将会不同。若用于高速系统则要缩短扫描周期时，可从软硬件上考虑。

### 2.3.4 输入输出滞后时间

输入输出滞后时间又称系统相应时间，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三部分组成。

PLC 是一种工业控制计算机，故它的工作原理是建立在计算机工作原理基础上的，即是通过执行反映控制要求的用户程序来实现的。但是 CPU 是以分时操作方式来处理各项任务的，在每一瞬间只能做一件事，所以程序的执行是按程序顺序依次完成相应各继电器的动作，便成为时间上的串行，实际输入输出的响应是有滞后的。

但 PLC 总的响应时间一般只有几十毫秒，对于一般的系统是无紧要的。要求输入输出信号之间的滞后时间尽量短的系统，可以选用扫描速度快的 PLC 或采取其他措施。

## 2.4 PLC 的特点及应用领域

### 2.4.1 PLC 的特点

#### 1. 编程简单，使用方便

目前，大多数 PLC 编程系统均采用了类似于继电器—接触器控制线路的梯形图编程语

言,与常用的计算机语言相比更容易为一般工程技术人员所理解和掌握。同时,PLC 编程软件或简易编程器的操作与使用也比较简单,现场可修改程序。

## 2. 维修工作量少,维护方便

可编程控制器的故障率很低,并且有完善的自诊断和显示功能。可编程控制器或外部的输入装置和执行机构发生故障时,可以根据可编程控制器上的发光二极管或编程器提供的信息迅速地查明故障的原因,用更换模块的方法可以迅速地排除故障。

## 3. 可靠性高于继电器控制系统

传统的继电器控制系统使用了大量的中间继电器、时间继电器。由于触点接触不良,容易出现故障。PLC 用软件代替了中间继电器和时间继电器,只剩下与输入和输出有关的少量硬件元件,接线可减少到继电器控制系统的  $1/10$  以下,大大减少了因触点接触不良造成的故障。

PLC 使用了一系列硬件和软件抗干扰措施,具有很强的抗干扰能力,平均无故障时间达到数万小时以上,可以直接用于有强烈干扰的工业生产现场,PLC 被广大用户公认为是最可靠的工业设备之一。

## 4. 系统的设计、安装、调试工作量少

PLC 用软件功能取代了继电器控制系统中大量的中间继电器、时间继电器、计数器等器件,使控制柜的设计、安装、接线工作量大大减少。

PLC 的梯形图程序可以用顺序控制设计法来设计。这种设计方法很有规律,很容易掌握。对于复杂的控制系统,用这种方法设计程序的时间比设计继电器系统电路图的时间要少得多。

大多数的 PLC 可以用仿真软件来模拟 CPU 模块的功能,用它来调试用户程序。在现场调试过程中,一般通过修改程序就可以解决发现的问题,系统的调试时间比继电器系统少得多。

## 5. 功能强,性能价格比高

PLC 内部提供了许多可供用户使用的编程元件,有很大的存储容量,能实现非常复杂的控制功能。另外,PLC 还可以提供联网通信功能,实现自动化控制系统的集成。与相同的硬件继电器控制系统相比,PLC 具有很高的性能价格比。

## 6. 体积小,能耗低

复杂的控制系统使用 PLC 后,可以减少大量的中间继电器和时间继电器,小型 PLC 的体积相当于几个继电器的大小,因此可以将开关柜的体积缩小到原来的  $1/2 \sim 1/10$ 。可编程序控制器的体积小、质量轻,以三菱公司的 FX0S-14 型超小型可编程序控制器(14 个输入/输出点)为例,其底部尺寸为  $90\text{mm} \times 60\text{mm}$ ,只有卡片大小,由于体积小,很容易装在机械设备内部,是实现机电一体化的理想控制设备。

## 7. 硬件配套齐全,通用性强,适用性强

PLC 产品已经标准化、系列化、模块化,配备有品种齐全的各种硬件装置供用户选用,用户能灵活、方便地进行系统配置,组成不同功能、不同规模的系统。PLC 的安装接线也很方便,一般用接线端子连接外部接线。PLC 有较强的带负载能力,可以直接驱动大多数电磁阀和中小型交流接触器。

硬件配置确定后,通过修改用户程序,就可以方便、快速地适应工艺条件的变化。

### 2.4.2 PLC 的应用领域

随着 PLC 功能的不断完善,性能价格比的不断提高,PLC 的应用面也越来越广。目前,PLC 在国内外已广泛应用于钢铁、采矿、水泥、石油、化工、电子、机械制造、汽车、船舶、装卸、造纸、纺织、环保、娱乐等各行各业。

PLC 的应用范围通常可分为以下五种类型:

#### 1. 顺序控制 (开关量逻辑控制)

这是 PLC 最广泛应用的领域,它取代传统的继电器顺序控制。PLC 应用于单机控制、多机群控制、生产自动线控制,例如,注塑机、印刷机械、订书机械、切纸机械、组合机床、磨床、装配生产线、包装生产线、电镀流水线及电梯控制等。

#### 2. 运动控制

PLC 制造商目前已提供了拖动步进电机或伺服电机的单轴或多轴位置控制模块。在多数情况下,PLC 把描述目标位置的数据送给模块,模块移动一轴或数轴到目标位置,当每个轴移动时,位置控制模块保持适当的速度和加速度,确保运动平滑。

运动的编程可用 PLC 的编程语言完成,通过编程器输入。操作员用手动方式把轴移动到某个目标位置,模块就得知了位置和运动参数,之后可用编辑程序来改变速度和加速度等运动参数,使运动平滑。

相对来说,位置控制模块比 CNC 装置体积更小,价格更低,速度更快,操作更方便。

#### 3. (闭环) 过程控制

PLC 能控制大量的物理参数,如温度、压力、速度和流量等。PID (Proportional - Integral - Derivative) 模块的提供使 PLC 具有闭环控制功能,即一个具有 PID 控制能力的 PLC 可用于过程控制。当控制过程中某个变量出现偏差时,PID 控制算法会计算出正确的输出,把变量保持在设定值上。PID 算法一旦适应了工艺,就不管工艺混乱而保持设定值。

#### 4. 数据处理

现代的 PLC 具有数学运算和数据传送、转换、排序和查表、位操作等功能,可以完成数据的采集、分析和处理,实现数据比较、传送或打印。

#### 5. 通信和联网

为了适应国外近几年来兴起的工厂自动化 (FA) 系统、柔性制造系统 (FMS) 及集散系统等发展的需要,首先,必须发展 PLC 之间、PLC 和上级计算机之间、PLC 与其他智能控制设备之间的通信功能。PLC 与其他智能控制设备一起,可以组成集中管理、分散控制的分布式控制系统。

PLC 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,例如在民用和家用自动化设备控制系统中也得到了迅速的发展。



1. 在复杂的电气控制中,采用 PLC 控制比传统的继电器控制有哪些优越性?

2. 什么是可编程控制器? 它的特点是什么?
3. PLC 由哪几部分组成? 各有什么作用?
4. PLC 控制系统与传统的继电器控制系统有何区别?
5. PLC 开关量输出接口按输出开关器件的种类不同, 有几种形式?
6. 简述 PLC 的扫描工作过程。
7. 为什么 PLC 中软继电器的触点可无数次使用?
8. PLC 按 I/O 点数和结构形式可分为几类?
9. PLC 执行程序是以循环扫描方式进行的, 请问每一扫描过程分为哪几个阶段?

## 第3章 S7-200 系列 PLC 的硬件和编程软件

德国的西门子（SIEMENS）公司是欧洲最大的电子和电气设备制造商，生产的 SIMATIC 可编程控制器在欧洲处于领先地位。其第一代可编程控制器是 1975 年投放市场的 SIMATIC S3 系列的控制系统。1979 年，微处理器技术被应用到可编程控制器中，产生了 SIMATIC S5 系列，取代了 S3 系列，SIMATIC S7 系列 PLC 是西门子公司于 20 世纪末推出的，根据控制规模的不同分为 S7-200、S7-300、S7-400 三个子系列，分别对应小型、中型、大型 PLC。随着 PLC 功能的逐步完善，小型 PLC 对市场的占有率也逐年提高，从本章开始，我们以 S7-200 系列 PLC 为例，讲述系统的基本构成及内部元件的编址方法，程序设计及其应用等。

### 3.1 S7-200 可编程控制器的组成

S7-200 系列 PLC 为单体式结构，配有 RS-485 通信端口、内置电源系统和部分 I/O 接口。它体积小、运算速度快、可靠性高，具有丰富的指令，系统操作简便，可方便地实现系统的 I/O 扩展，性能价格比高，是中小型控制系统的理想控制设备。S7-200 CN 是我国专用产品，具有更高的性价比。

#### 3.1.1 S7-200 PLC 的基本组成

SIMATIC S7-200 PLC 为单体式结构，即可用单独的 CPU 构成简单的开关量控制系统，也可用 I/O 扩展或通信联网功能构成中等规模的控制系统。SIMATIC S7-200 系统由硬件和工业软件两大部分构成，如图 3-1 所示。

硬件部分由基本单元、扩展单元、通信及网络设备、人机界面设备及其他相关设备组成。

工业软件是为更好地管理和使用这些设备而开发的与之相配套的程序、文档及其规则的总和，它主要由标准工具、工程工具、运行软件和人机接口等构成。

#### 3.1.2 CPU 模块结构

S7-200 系列 PLC 的 CPU 模块有两个子系列，即 CPU21X 和 CPU22X。CPU22X 系列已经成为市场的主流产品。

##### 1. CPU 模块的外形结构

如图 3-2 所示，PLC 主机单元上有工作方式开关、模拟电位器、I/O 扩展接口、工作状态指示和用户程序存储卡、I/O 接线端子排及发光指示等。

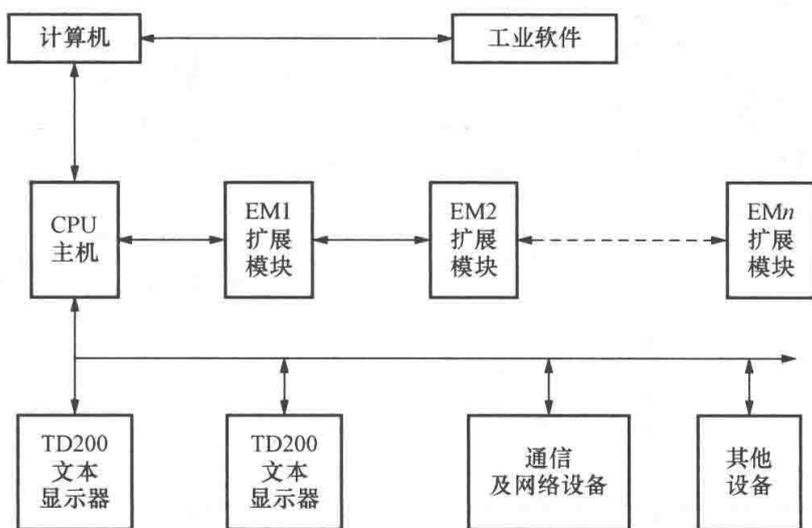


图 3-1 S7-200 PLC 的系统组成

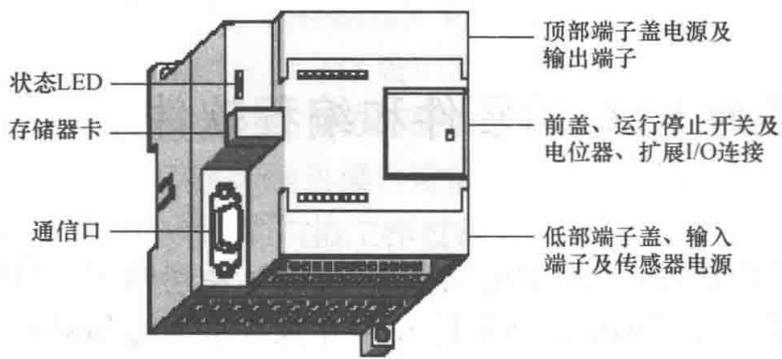


图 3-2 S7-200 PLC 的外形图

一般 PLC 主机及其扩展模块的宽度和厚度是固定的，长度随点数的不同而不同。

主机箱体外部的 RS-485 通信接口，用以连接编程器（手持式或 PC 机）、文本/图形显示器、PLC 网络等外部设备。

## 2. CPU 模块的型号

CPU 模块的型号有 5 种，见表 3-1。S7-200 系列 PLC 的电源供电形式有两种：一种为直流输入（24V DC），一种为交流输入（120~240V AC），分别用 DC 和 AC 描述；输入类型是指输入端子的输入形式，一般为直流，用 DC 描述；输出类型是指输出端子的输出形式，有两种形式的输出，即晶体管输出和继电器输出，分别用 DC 和 Relay 描述。如 CPU224 AC/DC/Relay 表示 PLC 型号为 224，交流电源供电，继电器输出；CPU226 DC/DC/DC 表示 PLC 型号为 226，直流电源供电，晶体管输出。通常，晶体管输出时，CPU 模块供电电源为直流；继电器输出时，供电电源为交流。CPU224XP 是在 CPU224 的基础上添加了 1 个通信口和 1 个集成的模拟量输入/输出通道。

表 3-1

S7-200 PLC 的 CPU 型号

CPU 模块	CPU 供电 (标称)	数字量输入	数字量输出	通信口	模拟量输入	模拟量输出	可拆卸连接
CPU221	24V DC	6×24V DC	4×24V DC	1	否	否	否
CPU221	120~240V AC	6×24V DC	4×继电器	1	否	否	否
CPU222	24V DC	5×24V DC	6×24V DC	1	否	否	否
CPU222	120~240V AC	5×24V DC	6×继电器	1	否	否	否
CPU224	24V DC	14×24V DC	10×24V DC	1	否	否	是
CPU224	120~240V AC	14×24V DC	10×继电器	1	否	否	是
CPU224XP	24V DC	14×24V DC	10×24V DC	2	2	1	是
CPU224XP	120~240V AC	14×24V DC	10×继电器	2	2	1	是
CPU226	24V DC	24×24V DC	16×24V DC	2	否	否	是
CPU226	120~240V AC	24×24V DC	16×继电器	2	否	否	是

## 3. S7-200 PLC 的 CPU 模块的技术性能指标

S7-200 22X 各主机的主要技术性能指标如表 3-2 所示。

表 3-2

S7-200 PLC 的主要技术性能指标

性能指标	CPU221	CPU222	CPU224	CPU224XP	CPU226
外形尺寸 (mm)	90×80×62	90×80×62	120.5×80×62	140×80×62	196×80×62
用户程序	4096	4096	8192	12288	16384
用户数据	2048	2048	8192	10240	10240
掉电保持时间 (h)	50	50	100	100	100

续表

性能指标	CPU221	CPU222	CPU224	CPU224XP	CPU226
本机数字量 I/O	6/4	8/6	14/10	14/10	24/16
本机模拟量 I/O	无	无	无	2/1	无
扩展模块数量	0	2	7	7	7
数字量 I/O 映像区	128 入/128 出	128 入/128 出	128 入/128 出	128 入/128 出	128 入/128 出
模拟量 I/O 映像区	无	16 入/16 出	32 入/32 出	32 入/32 出	32 入/32 出
脉冲捕捉输入	6	8	14	14	24
脉冲输出	2	2	2	2	2
辅助继电器 (M)	256	256	256	256	256
定时器/计数器	256/256	256/256	256/256	256/256	256/256
状态寄存器 (S)	256	256	256	256	256
高速计数器	4	4	6	6	6
定时中断	2 (1~255ms)	2 (1~255ms)	2 (1~255ms)	2 (1~255ms)	2 (1~255ms)
边沿中断	4 个上升沿或 4 个下降沿				
模拟电位器	1 (8bit)	1 (8bit)	2 (8bit)	2 (8bit)	2 (8bit)
布尔指令执行速度 ( $\mu\text{s}/\text{指令}$ )	0.22	0.22	0.22	0.22	0.22
口令保护	有	有	有	有	有
通信口	1	1	1	2	2
通信协议	PPI、DP、自由口				
最多主站数	32	32	32	32	32

#### 4. CPU 模块的端子接线

以 CPU224 例, CPU224DC/DC/DC 端子接线如图 3-3 所示, CPU224AC/DC/Relay 端子接线如图 3-4 所示。输入和输出都是采用分组式结构, 继电器输出形式的 PLC 既可以带直流负载, 也可以带交流负载, 而晶体管输出形式的 PLC 只能带直流负载。

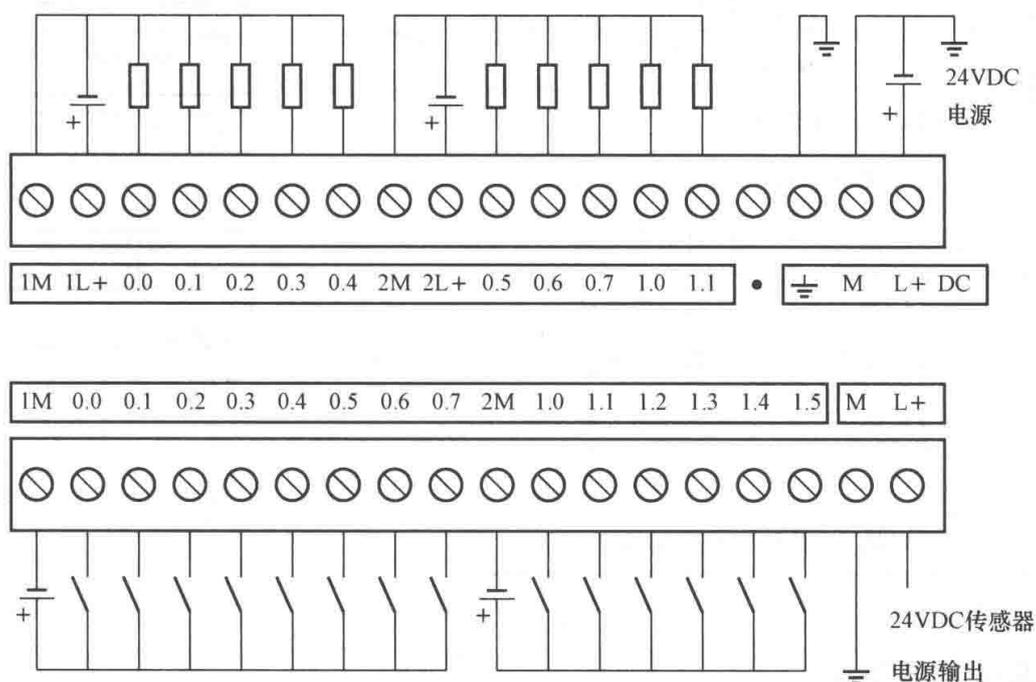


图 3-3 CPU224DC/DC/DC 端子接线

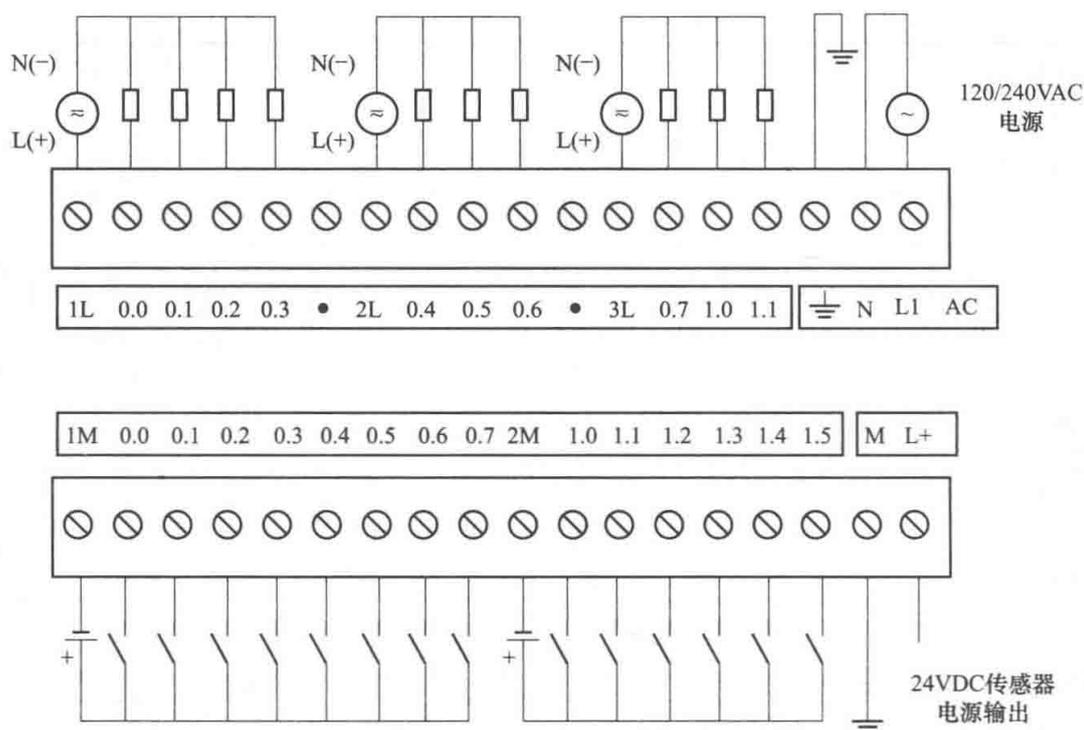


图 3-4 CPU224AC/DC/Relay 端子接线

### 3.1.3 数字量扩展模块

S7-200 PLC 有 3 大类共 10 种数字量输入/输出扩展模块, 见表 3-3。可根据应用系统的实际要求灵活配置 CPU 模块及数字量扩展模块。

数字量扩展模块的接线与 CPU 类似, 图 3-5 所示为 EM223 (4 输入/4 继电器输出) 模块的端子接线。其中 M、L+ 端子通常与 CPU 主机单元的 M、L+ 相连。

表 3-3 数字量输入/输出扩展模块

型号	扩展模块
EM221	8 点 24V DC 输入, 光电隔离
	16 点 24V DC 输入, 光电隔离
EM222	8 点 24V DC 输出型
	8 点继电器输出型
EM223	24V DC 输入 4 点/晶体管输出 4 点
	24V DC 输入 4 点/继电器输出 4 点
	24V DC 输入 8 点/晶体管输出 4 点
	24V DC 输入 8 点/继电器输出 8 点
	24V DC 输入 16 点/晶体管输出 16 点
	24V DC 输入 16 点/继电器输出 16 点

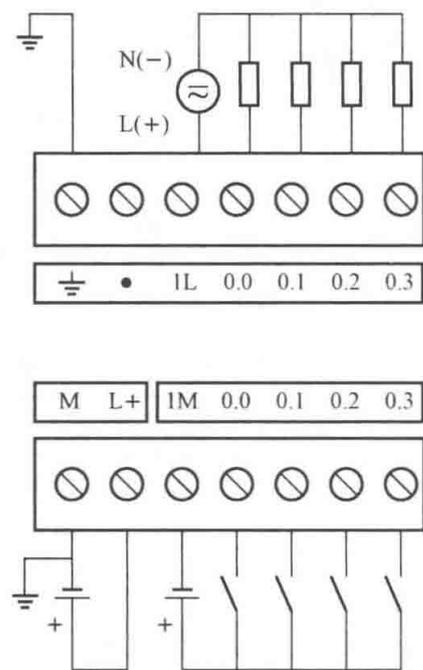


图 3-5 EM223 模块端子接线

### 3.1.4 其他扩展模块

除基本单元及数字量控制模块外, S7-200 (CN) PLC 还提供了许多扩展模块, 以满

足 PLC 的扩展应用, 扩展模块可以根据实际控制系统的要求来加以选择。

有用于模拟量信号检测与控制的模拟量扩展模块 EM231、EM232、EM235。有专用于温度信号检测的热电偶扩展模块 EM231CT、热电阻扩展模块 EM231RTD。有用于网络通信的模块, 如 PROFIBUS-DP 模块 EM277、工业以太网通信处理器 CP243-1、调制解调器模块 EM241 等。这些模块的性能及应用分别在相关的章节里详细介绍。关于特定模块的相关技术信息, 可参看西门子 S7-200 系列 PLC 产品手册。

### 3.2 S7-200(CN)PLC 的编程元件

程序设计中需要用到 PLC 的内部元件, 如输入输出继电器、辅助继电器、定时器、计数器等。这些元件具有与低压电器相似的功能, 但它们在 PLC 内部是以寄存器的形式出现的, 每个元件对应于一个或多个内部单元, 而非实际的硬件元件, 所以称为内部软元件或编程元件

S7-200 将编程元件统一归为存储器单元, 存储单元按字节进行编址, 无论所寻址的是何种数据类型, 通常应指出它在所在存储区域和在区域内的字节地址。每个单元都有唯一的地址, 地址用名称和编号两部分组成。

#### 3.2.1 编程元件的数据表示

(1) 用位来表示, 如 I0.0 (I0 表示 0 通道, I0.0 表示 0 通道的 0 位), 为 1 位。

(2) 用字节来表示, 如 IB0、VB1500 等, 为 8 位。

(3) 用字来表示, 相邻的两个字节组成一个字, 如 IW0、VW300 等, 为 16 位。IW0 是由 IB0 和 IB1 组成的, 其中 I 是区域标识符, 表示输入映像寄存器区域, B 表示字节 (byte), W 表示字 (word)。

(4) 用双字来表示, 如 ID0、VD100 等, 为 32 位, 其中 D 表示 (double word)。如图 3-6 所示的 ID0 为 32 位双字数据, 表示映像寄存器中编号为 0 的双字, 由 IB0、IB1、IB2、IB3 这 4 个字节组成。MSB 表示最高位, LSB 表示最低位。

用字节表示的序号是连续的, 但用字表示时就要用 IW0、IW2、IW4 等, 用双字表示时就要用 ID0、ID4、ID8 等, 这样就不会造成地址重叠。

MSB								LSB							
I0.7	...	I0.1	I0.0	I1.7	...	I1.1	I1.0	I2.7	...	I2.0	I3.7	...	I3.0		
IB0				IB1				IB2				IB3			
IW0								IW2							
ID0															

图 3-6 双字类型的数据表示

#### 3.2.2 S7-200 (CN) 的编程元件

##### 1. 数字量输入继电器 (I)

输入映像寄存器中的每一位对应于一个输入端子, 从而对应于一个数字量输入点。这里只是沿用了继电器—接触器控制系统的传统叫法, 称输入映像寄存器为输入继电器, 用字母 I 表示。可用位表示, 也可以用字节、字或双字表示, 如用 I0.0~I15.7 表示 128 个输入继电器。

## 2. 数字量输出继电器 (Q)

数字量输出继电器对应于 PLC 存储器中的输出映像寄存器, 用字母 Q 表示。与输入继电器的表示方法相同, 用 Q0.0~Q15.7 表示 128 个输出继电器。数字量输出继电器是用来将 PLC 的输出信号传递给负载, 线圈用程序指令驱动。

## 3. 模拟量输入寄存器 (AIW) 和模拟量输出寄存器 (AQW)

模拟量输入信号经 A/D 转换后的数字量信息存储在模拟量输出寄存器中, 将要经 D/A 转换成为模拟量的数字量信息存储在模拟量输出寄存器中。因为 CPU 处理的数字量为 16 位数据, 为字类型数据, 所以模拟量输入寄存器和模拟量输出寄存器常用 AIW 和 AQW 表示。AI 编址范围 AIW0、AIW2、…、AIW62, 起始地址定义为偶数字节地址, 共有 32 个模拟量输入点。AQ 编址范围 AQW0、AQW2、…、AQW62, 起始地址也采用偶数字节地址, 共有 32 个模拟量输出点。

模拟量输入寄存器只能读取, 而模拟量输出寄存器只能读入。

## 4. 变量寄存器 (V)

S7-200 (CN) PLC 还提供了大量的变量寄存器, 可用于数据运算、变量传送以及保存程序执行过程中的中间结果, 变量寄存器用字母 V 表示, 可用位表示, 也可以用字节、字或双字表示, 如 V200.0、V200.1、VB100、VW300、VD302 等。CPU 224 有 VB0.0~VB5119.7 的 5K 存储字节。

## 5. 辅助继电器 (M)

辅助继电器也称为辅助寄存器, 用字母 M 表示, 其功能相当于传统电气控制系统的辅助继电器或中间继电器, 只能起到中间状态的暂存作用, 不能直接驱动外部负载。可用位表示, 也可以用字节、字或双字表示, 但多以位的形式出现。

## 6. 特殊功能寄存器 (SM)

特殊功能寄存器也称为特殊标志寄存器或特殊继电器, 用字母 SM 表示。特殊功能寄存器为用户提供了一系列特殊的控制功能和系统信息, 有助于用户程序的编制和系统的各类状态信息的获取。用户也可将控制过程中的某些特殊要求通过特殊功能寄存器传递给 PLC, 可用位表示, 也可以用字节、字或双字编址。

CPU 224 编址范围 SM0.0~SM549.7, 共 550 个字节。其中 SM0.0~SM29.7 的 30 个字节为只读型区域, 用户不能更改。

(1) SMB0 为状态位字节。在每次扫描循环结尾由 S7-200 CPU 更新, 定义如下:

- 1) SM0.0: RUN 状态监控, PLC 在运行 RUN 状态, SMB0 该位始终为 1。
- 2) SM0.1: 首次扫描时为 1, PLC 由 STOP 转为 RUN 状态时, ON (1 态) 一个扫描周期, 用于程序的初始化。
- 3) SM0.2: 当 RAM 中数据丢失时, ON 一个扫描周期, 用于出错处理。
- 4) SM0.3: PLC 上电进入 RUN 方式, ON 一个扫描周期。
- 5) SM0.4: 分脉冲, 该位输出一个占空比为 50% 的分时钟脉冲, 用作时间基准或简易延时。
- 6) SM0.5: 秒脉冲, 该位输出一个占空比为 50% 的秒时钟脉冲, 可用作时间基准。
- 7) SM0.6: 扫描时钟, 一个扫描周期为 ON (高电平), 另一为 OFF (低电平), 循环交替。

8) SM0.7: 工作方式开关位置指示, 0 为 TERM 位置, 1 为 RUN 位置。为 1 时, 使自由端口通信方式有效。

(2) SMB1 为指令状态位字节。用于指示潜在错误的 8 个状态位, 这些位可由指令在执行时进行置位或复位, SMB1 部分位定义如下:

- 1) SM1.0: 零标志, 运算结果为 0 时, 该位置 1。
- 2) SM1.1: 溢出标志, 运算结果溢出或查出非法数值时, 该位置 1。
- 3) SM1.2: 负数标志, 数学运算结果为负时, 该位为 1。

常用的特殊功能寄存器如下:

(3) SMB2: 自由口通信接收字符缓冲区, 在自由口通信方式下, 接收到的每个字符都放在这里, 便于用户程序存取。

(4) SMB3: 用于自由口通信的奇偶校验, 当出现奇偶校验错误时, 将 SM3.0 置 1。

(5) SMB4: 用于表示中断是否允许和发送口是否空闲。

(6) SMB5: 用于表示 I/O 系统发生的错误状态。

(7) SMB6: 用于识别 CPU 的类型。

(8) SMB8~SMB21: 用于 I/O 扩展模板的类型识别及错误状态存储。

(9) SMW22~SMW26: 用于提供扫描时间信息, 如毫秒计的上次扫描时间、最短扫描时间及最长扫描时间。

(10) SMB28 和 SMB29: 分别对应模拟电位器 0 和 1 的当前值, 数值范围为 0~255。

(11) SMB30 和 SMB130: 分别为自由口 0 和 1 的通信控制寄存器。

(12) SMB34 和 SMB35: 用于存储定时中断间隔时间。

(13) SMB36~SMB65: 用于监视和控制高速计数器 HSC0、HSC1 和 HSC2 的操作。

(14) SMB66~SMB85: 用于监视和控制脉冲输出 (PTO) 和脉冲宽度调制 (PWM) 功能。

(15) SMB86~SMB94 和 SMB186~SMB194: 用于控制和读出接收信息指令的状态。

(16) SMB98 和 SMB99: 用于表示有关扩展模板总线的错误。

(17) SMB131~SMB165: 用于监视和控制高速计数器 HSC3、HSC4、HSC5 的操作。

(18) SMB166~SMB194: 用于显示包络表的数量、地址和变量寄存器在表中的首地址。

## 7. 定时器 (T)

定时器是 PLC 程序设计中的重要元件, 其作用相当于继电器—接触器控制系统中的时间继电器。S7-200 CPU22X 系列 PLC 共有 256 个定时器, 编号为 T0~T255, 有三种类型的时间基 (定时精度), 即 1、10、100ms。定时器的延时时间由指令的预设值和时间基确定, 即

$$\text{延时时间} = \text{定时器预设值} \times \text{时间基}$$

每个定时器有两种操作数: 一种是字类型, 用于存储定时器的当前值, 为 16 位有符号整数; 另一种是位类型, 称为定时器位, 用于反映定时器的延时状态, 相当于时间继电器的延时触点。这两种数据类型的字符表达与定时器编号完全相同, 在指令执行中具体访问哪种类型取决于指令的形式, 字类型操作指令取定时器当前值, 位类型操作指令取定时器位的值。

定时器有三种指令格式,即通电延时定时器(TON)、断电延时定时器(TOF)和带保持的通电延时定时器(RTON)。TON和TOF指令的动作特性与通电延时时间继电器和断电延时时间继电器相同。不同的指令格式、定时器编号,其时间基不同,定时器的刷新方法也不同。

#### 8. 计数器(C)

计数器也是 PLC 应用中的重要编程元件,计数器主要用来累计输入脉冲个数。S7-200 CPU22X 系列 PLC 共有 256 个计数器,编号为 C0~C255。计数器的预设值由程序设定。

每个计数器有两种操作数:一种是字类型,用于存储计数器的当前值,当前值寄存器用以累计脉冲个数;另一种是位类型,称为计数器位,用于反映计数状态,计数器当前值大于或等于预置值时,状态位置 1。这两种数据类型的字符表示与计数器编号相同,在指令执行中具体访问哪种类型的数据取决于指令的形式,字类型操作指令取计数器的当前值,位类型操作指令取计数器位的值。

计数器指令有加计数(CTU)、减计数(CTD)和加减计数(CTUD)三种形式。

一般计数器的计数频率受扫描周期的影响,频率不能太高。对于高频输入的计数应使用高速计数器。

#### 9. 高速计数器(HSC)

对高频输入信号计数时,可使用高速计数器。高速计数器只有一种数据类型,即有符号的 32 位的双字类型整数,用于存储高速计数器的当前值。CPU 22X 提供了 6 个高速计数器,即 HC0、HC1、…、HC5(每个计数器最高频率为 30kHz),用来累计比 CPU 扫描速率更快的事件。高速计数器的当前值为双字长的符号整数。

#### 10. 累加器(AC)

累加器是 S7-200 系列 PLC 内部使用较为灵活的存储器,可用于向子程序传递参数,或从子程序返回参数,也可以用来存放数据、运算结果等。S7-200 CPU22X 系列 PLC 提供了 4 个 32 位的累加器,编号为 AC0~AC3。累加器可以支持字节类型、字类型和双字类型的指令,数据存取时的长度取决于指令形式。若为字节类型指令,则只有低 8 位参与运算;若为字类型指令,则只有低 16 位参与运算;若为双字类型指令,则 32 位数据全部参与运算。

#### 11. 状态寄存器(S)

状态寄存器也称为状态元件或顺序控制继电器,是使用步进控制指令编程时的重要元件,以实现顺序控制和步进控制。S7-200 CPU22X 系列 PLC 共有 256 个状态寄存器(32 个字节),S7-200 PLC 编址范围为 S0.0~S31.7,它们常以“字节.位”的形式出现,与步进控制指令 LSCR、SCRT、SCRE 结合使用,实现顺序控制功能图的编程。

#### 12. 局部变量寄存器(L)

局部变量寄存器与变量寄存器很相似,其主要区别在于变量寄存器是全局有效的,而局部变量寄存器是局部有效的。“全局”指的是同一个寄存器可以被任何一个程序读取,如主程序、子程序、中断程序;而“局部”是指该寄存器只与特定的程序相关。S7-200 系列 PLC 给每个程序(主程序、各子程序和各中断程序)都分配有最多 64 字节的局部变量存储器,其中 60 个字节可以用作暂时存储器或者给予子程序传递参数,最后 4 个字节为系统保留字节,可以按位、字节、字和双字访问局部变量寄存器。

表 3-4 列出了 S7-200 CPU22X 系列 PLC 的存储器范围,可供编程时参考。

表 3-4 S7-200 CPU22X 系列 PLC 存储器范围

CPU 类型 编程元件	CPU221	CPU222	CPU224	CPU224XP	CPU226
输入映像存储器	I0.0~I15.7	I0.0~I15.7	I0.0~I15.7	I0.0~I15.7	I0.0~I15.7
输出映像存储器	Q0.0~Q15.7	Q0.0~Q15.7	Q0.0~Q15.7	Q0.0~Q15.7	Q0.0~Q15.7
模拟量输入	AIW0~AIW30	AIW0~AIW30	AIW0~AIW30	AIW0~AIW30	AIW0~AIW30
模拟量输出	AQW0~AQW30	AQW0~AQW30	AQW0~AQW30	AQW0~AQW30	AQW0~AQW30
变量寄存器	VB0~VB2047	VB0~VB2047	VB0~VB8191	VB0~VB10239	VB0~VB10239
局部变量寄存器	LB0~LB63	LB0~LB63	LB0~LB63	LB0~LB63	LB0~LB63
辅助继电器	M0.0~M31.7	M0.0~M31.7	M0.0~M31.7	M0.0~M31.7	M0.0~M31.7
特殊功能寄存器	SM0.0~SM179.7 SM0.0~SM29.7	SM0.0~SM279.7 SM0.0~SM29.7	SM0.0~SM549.7 SM0.0~SM29.7	SM0.0~SM549.7 SM0.0~SM29.7	SM0.0~SM549.7 SM0.0~SM29.7
定时器	T0~T255 带保持的通电延时, 时基为 1ms: T0, T64; 10ms: T1~T4, T65~T68; 100ms: T5~T31, T65~T95。 通电/断电延时, 时基为 1ms: T32, T96; 10ms: T33~T36, T97~T001; 100ms: T37~T63, T101~T255				
计数器	C0~C255	C0~C255	C0~C255	C0~C255	C0~C255
高速计数器	HC0~HC5	HC0~HC5	HC0~HC5	HC0~HC5	HC0~HC5
状态寄存器	S0.0~S31.7	S0.0~S31.7	S0.0~S31.7	S0.0~S31.7	S0.0~S31.7
累加器	AC0~AC3	AC0~AC3	AC0~AC3	AC0~AC3	AC0~AC3
调用/子程序	0~63	0~63	0~63	0~63	0~63
中断程序	0~127	0~127	0~127	0~127	0~127
跳转/标号	0~255	0~255	0~255	0~255	0~255

### 3.3 S7-200 PLC 的数据类型及表示方法

#### 3.3.1 数据类型、范围及数据表示

SIMATIC S7-200 系列 PLC 数据类型可以是布尔型、整型和实型（浮点数）。实数采用 32 位数来表示，其数值有较大的表示范围：正数为  $+1.175495E-38 \sim +3.402823E+38$ ；负数为  $-1.175495E+38 \sim -3.402823E+38$ 。不同长度的整数所表示的数值范围如表 3-5 所示。

表 3-5 数据类型及范围

整数长度	无符号整数表示范围		有符号整数表示范围	
	十进制	十六进制	十进制	十六进制
字节 B (8 位)	0~255	0~FF	-128~+127	80~7F
字 W (16 位)	0~65535	0~FFFF	-32768~32767	8000~7FFF
双字 D (32 位)	0~42949	0~FFFFFFFF	-2147483648~2147483647	80000000~7FFFFFFF

### 3.3.2 常数

在编程中经常会使用常数。常数数据长度可为字节、字和双字，在机器内部的数据都以二进制存储，但常数的书写可以用二进制、十进制、十六进制、ASCII 码或浮点数（实数）等多种形式。几种常数表示如表 3-6 所示。

表 3-6 常 数 表 示

进 制	书写格式	举 例
十进制	十进制数值	1180
十六进制	16#十进制数值	16#4AC5
二进制	2#二进制值	2#0011_0101_1101_0010
ASCII 码	'ASCII 码文本'	'ABCD'
浮点数	IEEE75-1985 标准	+1.175495E-38~+3.402823E+38

## 3.4 S7-200 PLC 的寻址方式

寻址方式是指程序执行时 CPU 如何找到指令操作数存放地址的方式。S7-200 系列 PLC 将数据信息存放于不同的存储器单元，每个单元都有确定的地址。根据对存储器数据信息的访问方式的不同，寻址方式可以分为直接寻址和间接寻址。

### 3.4.1 直接寻址方式

所谓直接寻址就是明确指出存储单元的地址，程序中指令的参数直接指明存储区域的名称（内部软元件符号）、地址编号和长度。

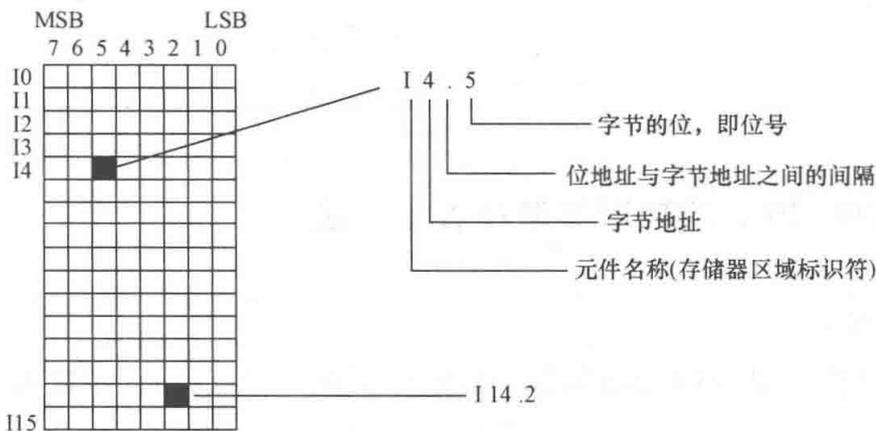


图 3-7 位寻址举例

常用的直接寻址方式有位寻址、字节寻址、字寻址和双字寻址。直接寻址方式也是 PLC 用户程序使用最多、最普遍的方式。可以按位、字节、字、双字方式对 I、Q、S、V、SM、M、L 等存储区域进行存取操作。

若要存取存储区的某一位，则必须指定地址，包括

存储器标识符、字节地址和位号。图 3-7 是一个位寻址的例子（也称为“字节·位”寻址）。在这个例子中，存储器区、字节地址（I 代表输入，4 代表字节 3）和位地址（第 5 位）之间用点号（“.”）相隔开。图 3-7 中 MSB 表示最高位，LSB 表示最低位。

若要存取 CPU 中的一个字节、字或双字数据，则必须以类似位寻址的方式给出地址，包括存储器标识符、数据大小以及该字节、字或双字的起始字节地址，如 VW100、MW2 等。VW100 表示变量储存区 V 中的字类型数据，地址为 100，执行指令时表示以字的方式存取数据。MW2 表示对由字节 MB2、MB3 组成的字进行存取操作。

### \* 3.4.2 间接寻址方式

间接寻址使用指针访问存储器中的数据。指针是包含另一个存储单元地址的双字存储单元。只能将 V 存储单元、L 存储单元或累加器 (AC1、AC2、AC3) 用作指针。不能使用间接寻址访问单个位。这种间接寻址方式与计算机的间接寻址方式相同。间接寻址在处理内存连续地址中的数据时非常方便,而且可以缩短程序所生成的代码的长度,使编程更加灵活。

用间接寻址方式存取数据需要做的工作有 3 种:建立指针、间接存取和修改指针。

#### 1. 建立指针

建立指针必须用双字传送指令 (MOVD),相应的操作数用符号 “&” 引导,用以表示存储器的地址。装入目的操作数的是源操作数的地址而不是源操作数数据本身,将存储器所要访问的单元的地址装入用来作为指针的存储器单元或寄存器。

#### 2. 间接存取

指令中在操作数的前面加 “\*” 表示该操作数为一个指针。

下面是建立指针和间接存取的应用例子。

若存储区的地址及单元中所存的数据如图 3-8 所示。执行过程如下:

```
LD    SM0.0
MOVB  16# 12, VB200
MOVB  16# 34, VB201
MOVB  16# 56, VB202
MOVB  16# 78, VB203
MOVD  &VB200, AC1
MOVB  * AC1, MB2
MOVW  * AC1, MW4
```

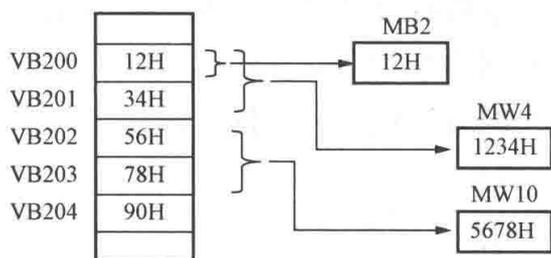


图 3-8 间接寻址方式的应用例子

#### 3. 修改指针

修改指针可以改动指针数值。由于指针是 32 位数值,使用双字指令修改指针数值。可使用简单算术操作 (例如加或递增) 修改指针数值。例如:

```
LD    SM0.0
+ D   2, AC1
MOVW  * AC1, MW10
```

## 3.5 S7-200 编程语言及程序结构

PLC 的编程语言有梯形图、语句表 (指令表)、逻辑符号图及其他高级语言等,但在西门子 S7-200 (CN) 系列 PLC 的工程应用项目中,主要应用梯形图 (LAD)、语句表 (STL)、逻辑符号图 (FBD) 三种。

### 3.5.1 S7-200 编程语言

#### 1. 梯形图

梯形图是一种以图形符号及图形符号在图中的相互关系表示控制关系的编程语言,它是从继电器控制电路图演变过来的。梯形图将继电器控制电路图进行简化,同时加进了许多功能强大、使用灵活的指令,将微机的特点结合进去,使编程更加容易,而实现的功能却大大超过传统继电器控制电路图,是目前最普通的一种可编程控制器编程语言。如图 3-9 所示。

#### 2. 语句表 (STL)

语句表 (STL) 语言类似于计算机的汇编语言,特别适合于来自计算机领域的工程人

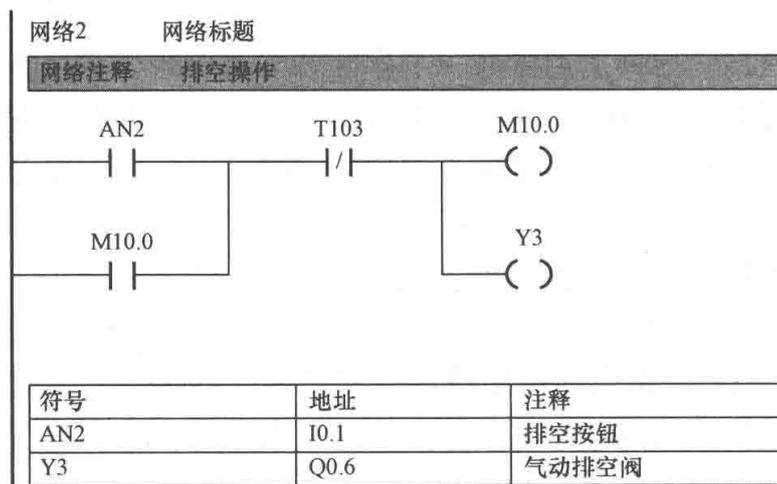


图 3-9 梯形图程序

员。用指令助记符创建用户程序,属于面向机器硬件的语言,梯形图编程语言的优点是直观、简便,但要求用带 CRT 屏幕显示的图形编程器才能输入图形符号。小型的编程器一般无法满足,而是采用经济便携的编程器(指令编程器)将程序输入到可编程控制器中,这种编程方法使用指令语句(助记符语言),它类似于微机中的汇编语言。

语句是语句表编程语言的基本单元,每个控制功能有一个或多个语句

组成的程序来执行。每条语句规定可编程控制器中 CPU 如何动作的指令,它是由操作码和操作数组成的。

编程软件 STEP7 - Micro/Win V4.0 SP3 的语句表举例如下:

网络 2    网络标题

网络注释      排空操作

```
LD    AN2                    //逻辑开始
O     M10.0                 //并联条件
AN    T103                  //串联条件
=     M10.0                 //启动输出 1
=     Y3                    //启动排空阀
```

### 3. 功能块图

功能图编程语言实际上是用逻辑功能符号组成的功能块来表达命令的图形语言,与数字电路中逻辑图类似,它极易表现条件与结果之间的逻辑功能。图 3-10 所示为先“或”后“与”再输出操作的功能块图程序。

由图 3-10 可见,这种编程方法是根据信息流将各种功能块加以组合,是一种逐步发展起来的新式的编程语言,在 SIEMENS 的小型控制器“LOGO!”中使用了 FBD 编程语言。

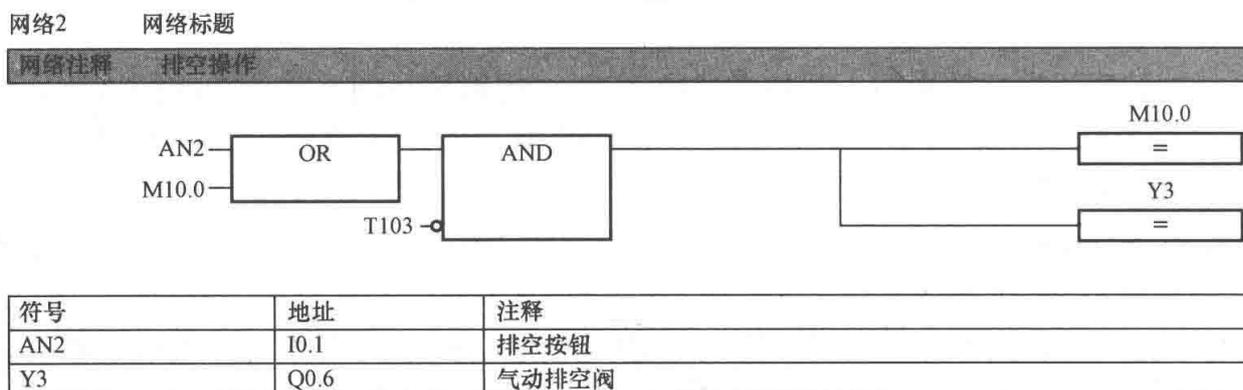


图 3-10 功能块图程序

总之, LAD 程序被分为程序段。一个程序段是按照顺序安排的以一个完整电路的形式连接在一起的触点、线圈,不能短路或者开路,也不能有能流倒流的现象存在。可以利用编

程软件为每一个程序段加注释。STL 程序不用分段，但可以用关键词 NETWORK 将程序分段。FBD 编程使用程序段的概念对程序进行分段和注释。

### 3.5.2 程序结构

一个程序块由可执行代码和注释组成。可执行代码由主程序和若干子程序或者中断服务程序组成。可执行代码被编译并下载到 S7-200 中，而程序注释不会被下载。可以使用组织组件（主程序、子程序和中断服务程序）来结构化控制程序。如果编程使用的是手编器，主程序应安排到程序的最前面。其他部分的位置安排没有严格的顺序，但习惯上把子程序安排在中断程序的前面，如图 3-11 所示。

#### 1. 主程序

主程序中包括控制应用的指令。S7-200 在每一个扫描周期中顺序执行这些指令。主程序也被表示为 OB1。

#### 2. 子程序

子程序是应用程序中的可选组件。只有被主程序、中断服务程序或者其他子程序调用时，子程序才会执行。

当希望重复执行某项功能时，子程序是非常有用的。与其在主程序中的不同位置多次使用相同的程序代码，不如将这段程序逻辑写在子程序中，然后在主程序中需要的地方调用。调用子程序有以下优点：

(1) 用子程序可以减小程序的长度。

(2) 由于将代码从主程序中移出，因而用子程序可以缩短程序扫描周期。S7-200 在每个扫描周期中处理主程序中的代码，而子程序只有在被调用时，S7-200 才会处理其代码，在不调用子程序时，S7-200 不会处理其代码。

(3) 用子程序创建的程序代码是可传递的。可以在一个子程序中完成一个独立的功能，然后将它复制到另一个应用程序中而无需做重复工作。

#### 3. 中断处理程序

中断服务程序是应用程序中的可选组件。当特定的中断事件发生时，中断服务程序执行。可以为一个预先定义好的中断事件设计一个中断服务程序。当特定的事件发生时，S7-200 PLC 会执行中断服务程序。

中断服务程序不会被主程序调用。只有当中断服务程序与一个中断事件相关联，且在该中断事件发生时，S7-200 才会执行中断服务程序。



图 3-11 程序结构

## 3.6 STEP7 - Micro/WIN V4.0 编程软件的使用与安装

SIMATIC S7-200 编程软件是指西门子公司为 S7-200 系列可编程控制器编制的工业编程软件的集合，其中 STEP7 - Micro/WIN SP3 软件是基于 Windows 的应用软件。本书以 STEP7 - Micro/WIN V4.0 编程软件为例，介绍编程软件的功能、安装和使用方法，并结合应用实例讲解用户程序的输入、编辑、调试及监控运行的方法。

### 3.6.1 SIMATIC S7-200 编程软件的基本功能

STEP7-Micro/WIN V4.0 编程软件在离线条件下,可以实现程序的输入、编辑、编译等功能。

编程软件在联机工作方式可实现程序的上/下载、通信测试及实时监控等功能。编程软件安装完毕后,双击 STEP 7-Micro/WIN 软件图标,即可进入编程软件主界面,如图 3-12 所示。主界面的各组成部分及其功能简述如下:

#### 1. 菜单栏

菜单栏包括文件 (File)、编辑 (Edit)、视图 (View)、可编程控制器 (PLC)、调试 (Debug)、工具 (Tools)、窗口 (Windows) 和帮助 (Help) 等 8 个主菜单项,提供了文件操作、程序编辑、PLC 参数设置、系统组态、编程环境设置、窗口切换以及帮助等多种功能,包括了用户在 STEP 7-Micro/WIN 环境下可以进行的绝大部分操作。

#### 2. 工具条

工具条是以按钮形式提供给用户在程序编辑、调试和运行时最常用的操作。用户可通过菜单命令 View/Toolbars 自定义工具条。

#### 3. 浏览条

浏览条可实现编程过程中各功能窗口的快速切换。单击浏览条中的任一控制按钮,即可将编程主窗口切换为该按钮对应的窗口。浏览条中的各控制按钮及其功能如下:

(1) 程序块 (Program Block)。单击程序块按钮,可将编程主窗口切换至程序编辑器,如图 3-12 中的 SIMATIC LAD 窗口,用户可在该窗口下编制梯形图程序。如果使用的是语句表或功能块图,则将分别打开 SIMATIC STL 窗口和 SIMATIC FBD 窗口,这 3 种窗口可通过“查看”菜单下的 LAD、STL 和 FBD 命令切换。

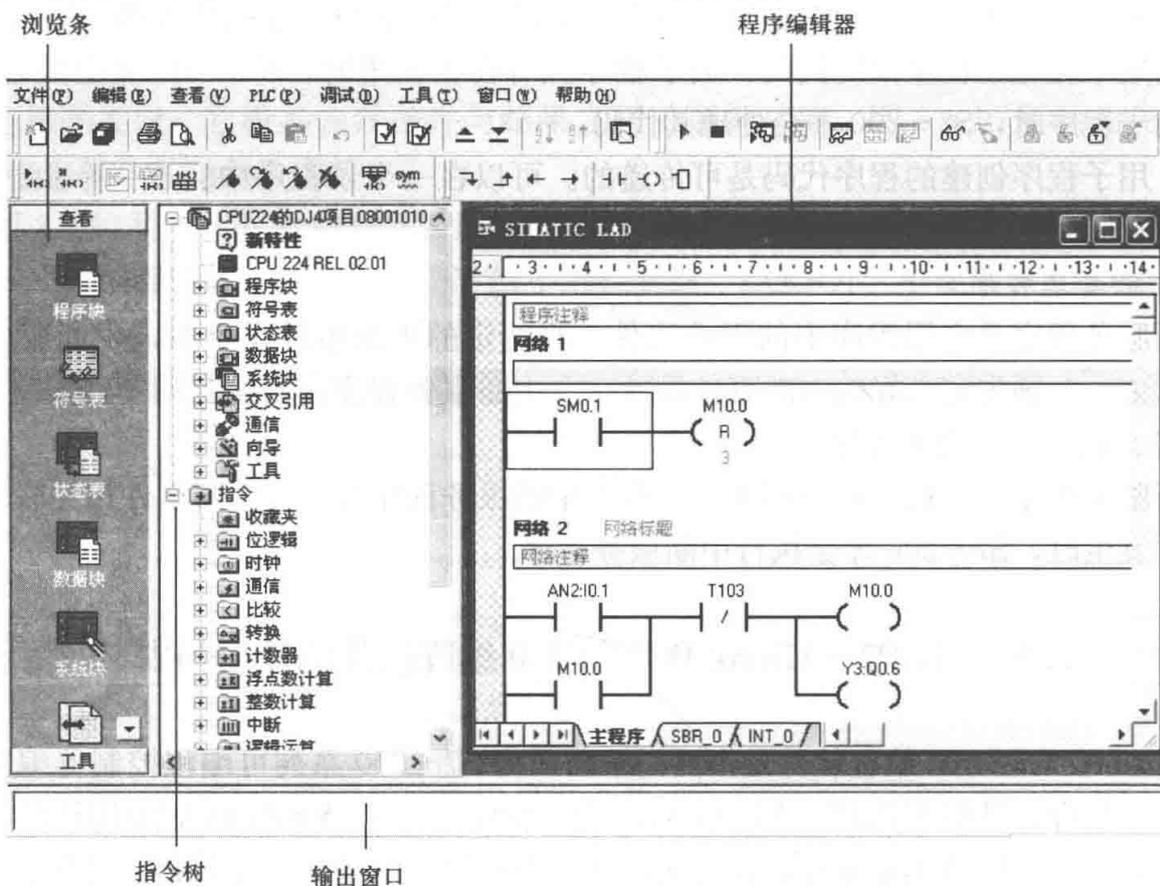


图 3-12 STEP7-Micro/WIN V4.0 编程软件主界面

单击 SIMATIC LAD 窗口下部的 MAIN、SBR\_0、INT\_0 等标签可分别进入主程序、子程序 0、中断程序 0 的编辑界面。

程序编辑器的上部为程序的局部变量表，可用于定义局部变量。在带参数的子程序调用中，参数的传递是通过局部变量表进行的。

(2) 符号表 (Symbol Table)。符号表可将 PLC 内部元件的地址定义为特定符号，即用具有某种实际意义的文字符号名称替换内部元件的实际地址。如系统的启动按钮地址为 I0.0，如果在符号表中将 I0.0 定义为 START，则在梯形图中所有的 I0.0 地址均由 START 代替。为增强程序的可读性，还可以在符号表窗口内附加一些注释信息。

(3) 状态图表 (Status Chart)。状态图表窗口用于联机调试时监视所选择变量的状态及当前值。只需要在地址栏中 (Address) 写入欲监视的变量地址，在数据格式 (Format) 栏中注明所选择变量的数据类型，选择 Debug 菜单下的 Single Read 命令就可以在运行时监视这些变量的状态及当前值。

(4) 数据块 (Data Block)。在数据块窗口中，可以设置和修改变量寄存器中的一个或多个变量值。使用时应注意变量地址和变量类型以及数据的范围要匹配。

(5) 系统块 (System Block)。系统块窗口主要用于系统组态，允许为 S7-200 PLC 配置不同的硬件参数，如数字量输入滤波设置、模拟量输入滤波设置、脉冲捕捉设置、配置数字量输出表、存储器保持范围设定、CPU 密码设置、模拟电位器设置、高速计数器设置和高速脉冲输出设置等。

(6) 交叉索引 (Cross Reference)。当用户程序编译完成后，交叉索引窗口将提供相关索引信息，如交叉索引信息、字节使用情况信息和位使用情况信息等。

(7) 通信 (Communications)。通信窗口的功能是建立计算机与 PLC 之间的通信连接及设置通信参数。

#### 4. 指令树 (Instruction Tree)

指令树包括两部分：项目文件管理系统和指令系统。项目文件管理系统主要包括 CPU 型号确定、程序重命名、子程序与中断程序的添加和删除等。指令系统以树的形式按类列举了所有的快捷操作指令和 PLC 指令。

#### 5. 输出窗口

输出窗口用于显示程序的编译结果。如果程序有错误，则通过输出窗口可查询错误类型、错误所属的程序以及其具体位置等信息。

#### 6. 状态条

状态条在编程软件的最低端，也称为任务栏，用于指示编程主窗口的状态。

#### 7. 程序编辑器

程序编辑器是程序编辑的窗口，当创建一个新的项目后，在浏览条中单击程序块按钮即可打开程序编辑器，图 3-12 所示为 LAD 程序编辑器窗口界面，图中的方框（也称为光标）为当前正在编辑的指令，每个指令由指令图标和操作数（或操作数地址）组成。

单击程序编辑器底部的标签，可以在主程序、子程序和中断程序之间移动。

### 3.6.2 编程软件的安装 (安装方法)

#### 1. 安装准备

关闭所有的应用程序，在光盘驱动器中插入驱动光盘，如果没有禁止光盘插入自动运

行, 安装程序会自动进行, 或者在 Windows 资源管理器中打开光盘上的 “Setup.exe”。

### 2. 按照安装程序的提示完成安装

首先选择安装程序界面语言, 再选择安装目的文件夹, 对于自动弹出的 “Set PG/PC Interface” 窗口, 单击 OK 即可, 安装完成后重启计算机。

### 3. 中英文语言选择

如要选择编程软件界面为中文格式, 可以双击桌面上的 STEP7 - Micro/WIN V4.0 图标, 在 Tools/Options/General 的语言栏中选择 “Chinese”, 然后重启编程软件。

### 3.6.3 建立与 S7-200 CPU 的通信

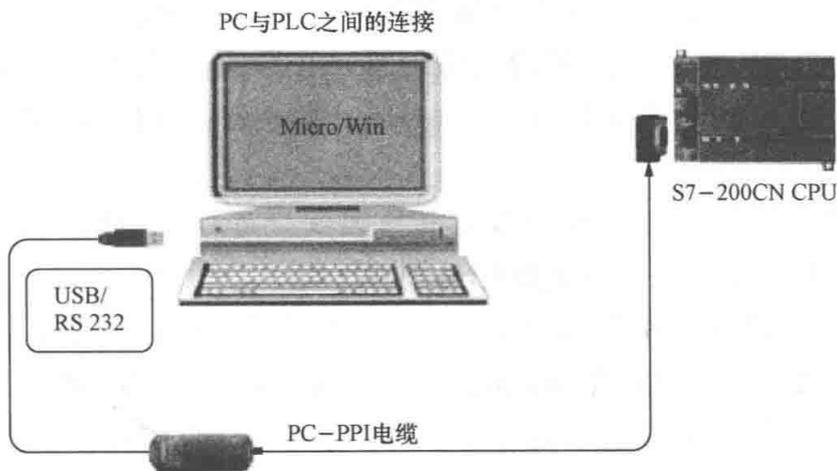


图 3-13 通信线的连接

用 STEP 7 - Micro/WIN 软件或其他编程器对用户程序进行下载、调试、监控时, 应首先建立与 PLC 的通信连接。图 3-13 所示为连接 S7-200 与编程设备的 RS-232/PPI 多主站电缆。

#### 1. 通信电缆连接

连接 RS-232/PPI 多主站电缆的 RS-232 端 (标识为 “PC”) 到编程设备的通信口上 (COM2)。

连接 RS-232/PPI 多主站电缆

的 RS 485 端 (标识为 “PPI”) 到 S7-200 的端口 0 或端口 1。

#### 2. 通信参数设置

单击浏览条中的通信按钮 (Communications), 或选择菜单命令 View/Communications, 将弹出通信对话框, 双击 “PC/PPI 电缆” 图标或单击左下方的 Set PG/PC Interface 按钮, 弹出设置 PG/PC 接口对话框, 选择 PC/PPI Cable, 单击 Properties 按钮, 弹出 PC/PPI 电缆属性对话框, 用于设置通信参数, 如图 3-14 所示。

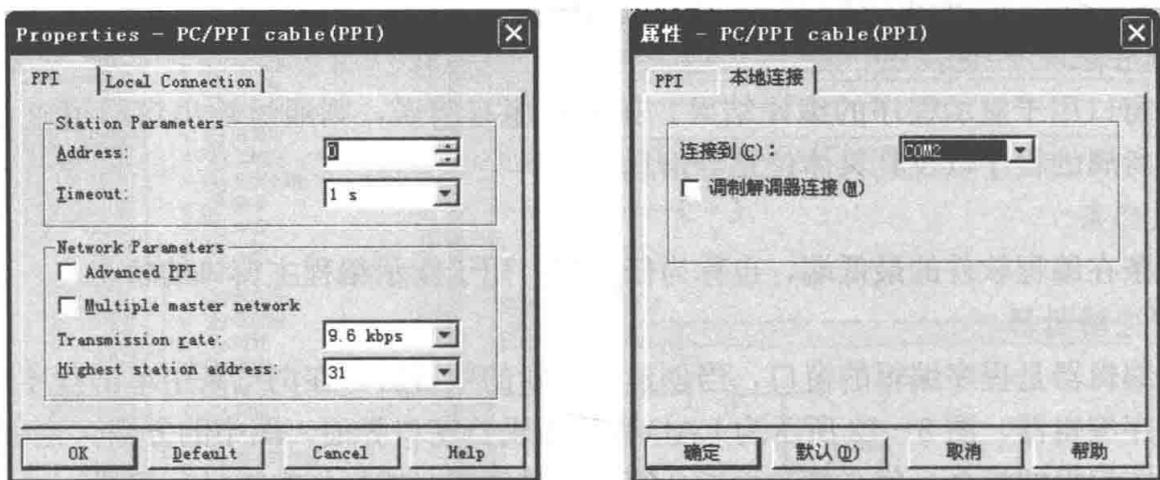


图 3-14 PC/PPI 通信参数设置

在属性对话框中, 可在 PPI 中设置主站地址 (默认为 0)、波特率 (默认 9.6kbps)、最高站地址等, 在 Local Connection 中设置通信连接接口等。

### 3. 建立通信连接

设置好通信参数后，返回至通信对话框，双击刷新图标，STEP 7 - Micro/WIN 将检查所有已连接的 S7 - 200 系列 PLC 的 CPU 站，并为每个 CPU 站建立一个 PLC 图标，如图 3 - 15 所示，图中显示已检测到 1 个 PLC 站，站号为 2，PLC 的型号为 CPU224。

若网络中有多个 PLC 站，则单击相应 PLC 站的图标，在对话框左边会显示出各站的地址信息和通信参数。

如果双击 PLC 站图标，则会弹出 PLC Information 信息框，显示 PLC 的型号、内部错误显示、I/O 错误显示及硬件组态等信息。

建立了通信连接后，即可应用 STEP 7 - Micro/WIN 软件的在线监控功能，对用户程序进行调试。

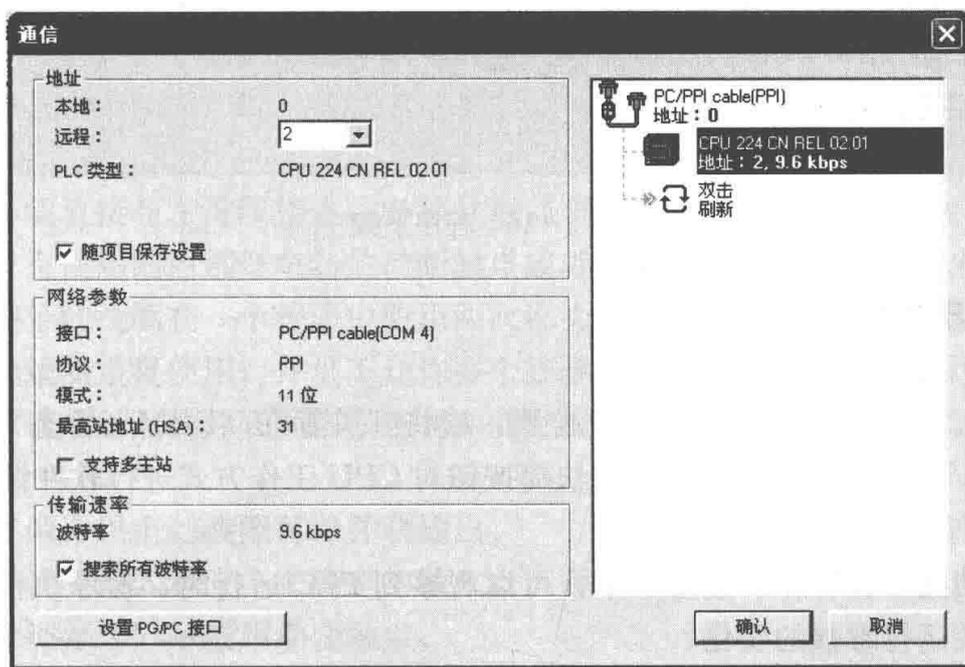


图 3 - 15 PC/PPI 通信对话框

### 3.6.4 程序编制

#### 1. 建立项目（用户程序）

可以用新建菜单建立一个项目，也可以用打开菜单打开已有的项目，也可以用另存为菜单保存项目并修改项目名称，还可以在文件菜单中为项目设置密码。

#### 2. 程序的编辑、修改

通常利用梯形图编辑器进行梯形图程序的编辑、修改，程序的编辑包括程序的剪切、拷贝、粘贴、插入和删除，字符串替换、查找等。

#### 3. 程序的编译及上/下载

(1) 编译。程序的编译，能明确指出错误的网络段，编程者可以根据错误提示对程序进行修改，然后再次编译，直至编译无误。

(2) 下载。用户程序编译成功后，将下载块中选中下载内容下载到 PLC 的存储器中，如图 3 - 16 所示。

(3) 载入（上载）。载入可以将 PLC 中未加密的程序或数据向上送入编程器（PC 机）。将选择的程序块、数据块、系统块等内容上载后，可以在程序窗口显示上载的 PLC 内

部程序和数据信息。

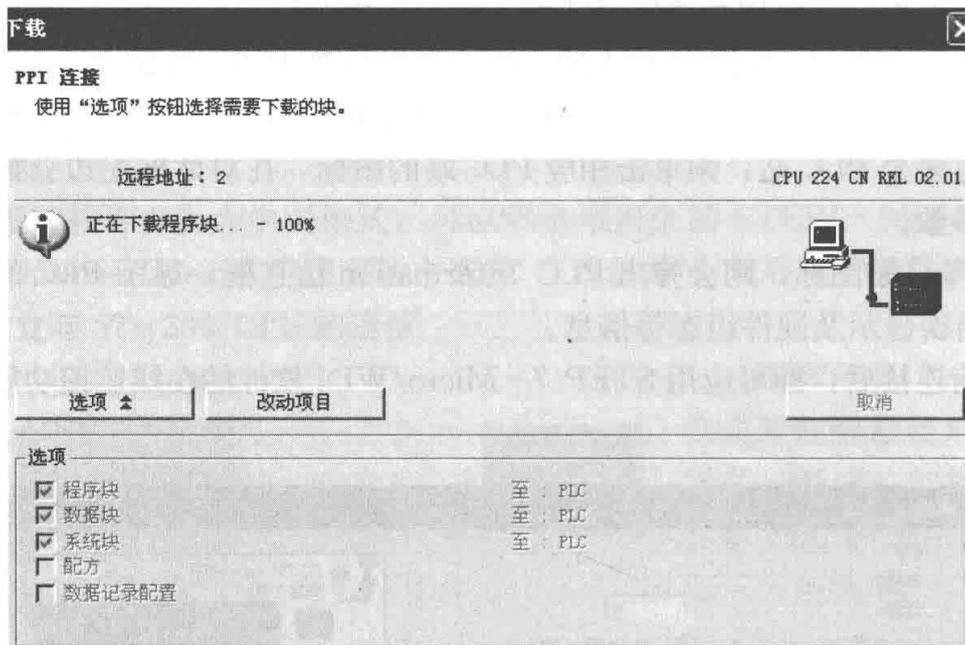


图 3-16 程序下载

### 3.6.5 程序的监视、运行、调试

#### 1. 程序运行方式的设置

将 CPU 的工作方式开关置在 RUN 位置，或将开关置在 TERM（暂态）位置时，操作 STEP7 - Micro/WIN V4.0 菜单命令或快捷按钮对 CPU 工作方式进行软件设置。

#### 2. 程序运行状态的监视

运用监视功能，在程序状态打开下，可以观察到 PLC 运行时，程序执行的过程中各元件的工作状态及运行参数的变化。

#### 3. 变量的强制操作

在实验室调试程序时，往往会因为没有现场实际设备使得用户程序无法正常调试。STEP 7 - Micro/WIN 提供了软件“强制”功能，允许用命令的形式来改变程序中的各变量的值，使调试过程简单快捷。

总之，STEP 7 - Micro/WIN V4.0 软件提供了丰富的资源供用户使用，这里由于篇幅有限就不一一介绍了，使用者可在实际使用中逐一掌握。

## 3.7 梯形图程序的执行原理及编程规则

### 3.7.1 梯形图程序的执行原理

梯形图语言是一种图形化语言，在梯形图程序分析中，可以认为有一个假想的能流从左向右流过，触点代表电流（能量流）的控制开关，线圈代表由电流充电的中继或输出；框盒（指令盒）代表能流到达此框时执行指令盒的功能。若将左侧母线视为能流源，当能流流经某个元件时，如果该元件是导通的，则能流继续流到下一个元件。如果该元件是断开的，则能流停止流动。对于指令盒（或称为功能指令）来说，其 EN 和 ENO 端可以认为是能流的输入和输出端。

“→”称为可选的能流指示器，该符号可以结束能流，能流也可以在该符合后继续流到下一个元件。在 STEP 7 - Micro/WIN 编程软件的梯形图窗口中，任何一个空的程序段网络都是以该符合开始的。

值得注意的是，引入能流的概念仅仅是为了便于理解梯形图程序的执行过程，在实际上并不存在这种能量的流到过程。

### 3.7.2 梯形图程序编程规则

(1) 梯形图程序由若干个网路段组成。梯形图网络段的结构不增加程序长度，软件编译结果可以明确指出错误语句所在的网络段，清晰的网络结构有利于程序的调试，正确的使用网络段，有利于程序的结构化设计，使程序简明易懂。

(2) 梯形图程序必须符合顺序执行的原则，即从左到右、从上到下执行。

(3) 梯形图每一行都是从左母线开始，线圈接在右边。触点不能放在线圈的右边，在继电器控制的原理图中，热继电器的接点可以加在线圈的右边，而 PLC 的梯形图是不允许的。

(4) 外部输入/输出继电器、内部继电器、定时器、计数器等器件的触点可多次重复使用。

(5) 线圈不能直接与左母线相连，必须从触点开始，以线圈或指令盒结束。如果需要，可以通过一个没有使用的内部继电器的动断触点或者特殊内部继电器的动合触点来连接。

(6) 同一编号的线圈在一个程序中使用两次称为双线圈输出。双线圈输出容易引起误操作，应尽量避免线圈重复使用，并且不允许多个线圈串联使用。

(7) 梯形图程序触点的并联网络多连在左侧母线，设计串联逻辑关系时，应将单个触点放在右边。

(8) 两个或两个以上的线圈可以并联输出。

(9) 每一个开关输入对应一个确定的输入点，每一个负载对应一个确定的输出点。外部按钮（包括启动和停车）一般用动合触点。

(10) 输出继电器的使用方法。输出端不带负载时，控制线圈应使用内部继电器 M 或其他线圈，不要使用输出继电器 Q 的线圈。

## 习 题



1. 简述 S7-200 PLC 系统的基本组成。
2. S7-200 PLC 有哪些编程工具？各有什么特点？
3. S7-200 PLC 有哪些常用的编程元件？
4. S7-200 PLC 的编程语言有哪些？各有什么特点？
5. CPU224 CN AC/DC/Relay 是基本单元还是扩展单元？有多少个输入/输出点？属于什么输出类型？
6. 可编程控制器梯形图编程规则的主要内容是什么？

## 第4章 S7-200 PLC的编程指令

S7-200 PLC指令系统功能强大，内容丰富。随着控制系统的要求越来越高，其指令系统也在不断地完善。最初为取代继电器控制系统而开发的指令称为基本指令，为扩展PLC的功能发展出来的指令称为功能指令或应用指令（指令盒）。

### 4.1 基本逻辑指令

#### 4.1.1 位操作指令

##### 1. 指令格式

LD	BIT	LDN	BIT	//用于网络段起始的动合/动断触点
A	BIT	AN	BIT	//动合/动断触点串联,逻辑与/与非指令
O	BIT	ON	BIT	//动合/动断触点并联,逻辑或/或非指令
=	BIT			//线圈输出,逻辑置位指令

##### 2. 基本位操作指令操作数寻址范围

基本位操作指令操作数寻址范围：I、Q、M、SM、T、C、V、S、L等。

##### 3. 指令助记符

指令助记符：LD (Load)、LDN (Load Not)、A (And)、AN (And Not)、O (Or)、ON (Or Not)、= (Out)。

基本位操作指令如图4-1所示。

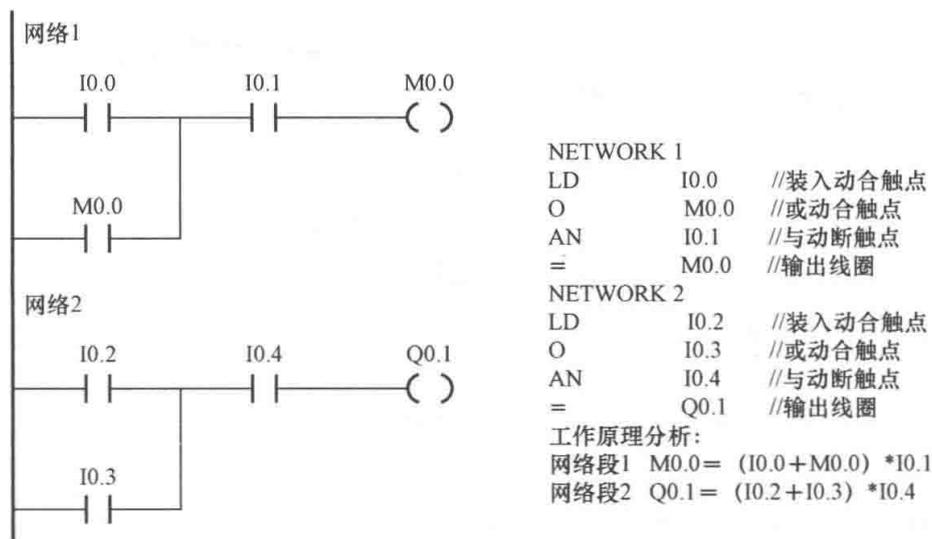


图4-1 基本位操作指令

#### 4.1.2 取非和空操作指令

取非和空操作指令格式 (LAD、STL、功能)：

NOT		取非	
NOP	N	空操作指令	次数 N=0~255

**【例 4-1】** 图 4-2 为取非指令和空操作指令应用举例。

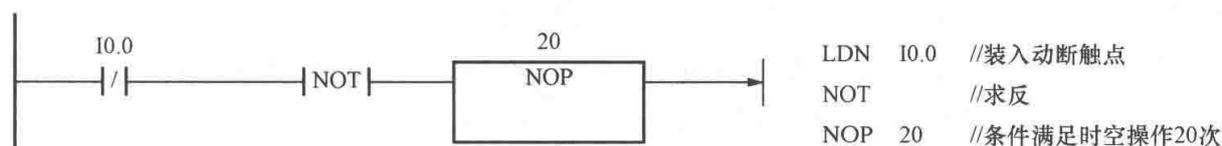


图 4-2 取非指令和空操作指令

### 4.1.3 边沿触发指令（脉冲生成）

边沿触发指令（或称正负跳变指令）检测到脉冲的每一次负跳变后，产生一个微分脉冲。

#### 1. 指令格式

指令格式：EU（Edge Up）正跳变，无操作元件；

ED（Edge Down）负跳变，无操作元件。

**【例 4-2】** 图 4-3 是跳变指令程序片断。图 4-4 是图 4-3 指令执行的时序图。

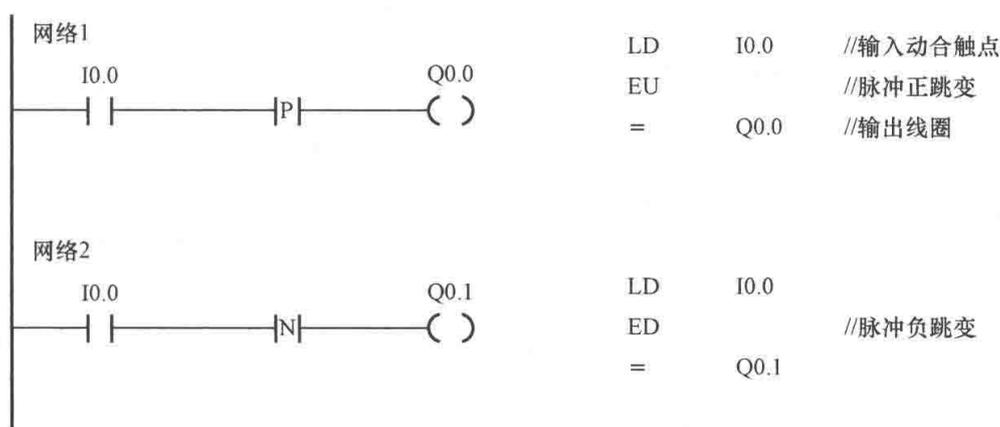


图 4-3 跳变指令程序片断

#### 2. 用途

边沿触发是指用边沿触发信号产生一个机器周期的扫描脉冲，通常用作脉冲整形。

#### 3. 分类

边沿触发指令分为正跳变触发（上升沿）和负跳变触发（下降沿）两大类。

正跳变触发指输入脉冲的上升沿，使触点 ON 一个扫描周期。负跳变触发指输入脉冲的下降沿，使触点 ON 一个扫描周期。

### 4.1.4 置位和复位指令

#### 1. 置位指令

将位存储区的指定位（位 bit）开始的 N 个同类存储器位置位。

用法：S bit, N 例：S Q0.0, 1

#### 2. 复位指令

将位存储区的指定位（位 bit）开始的 N 个同类存储器位复位。当用复位指令时，如果是对定时器 T 位或计数器 C 位进行复位，则定时器位或计数器位被复位，同时，定时器或计数器的当前值被清零。

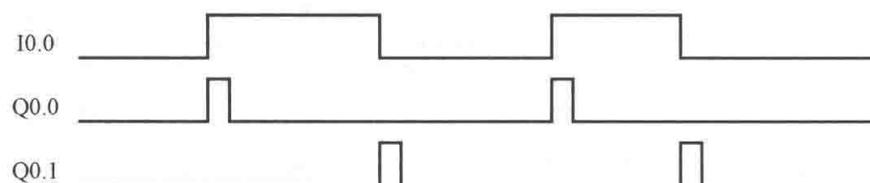


图 4-4 时序图

用法: R bit, N 例: R Q0.2, 3

置位即置 1, 复位即置 0。置位和复位指令可以将位存储区的某一位开始的一个或多个 (最多可达 255 个) 同类存储器位置 1 或置 0。这两条指令在使用时需指明三点: 操作性质、开始位和位的数量。各操作数类型及范围如表 4-1 所示。

表 4-1 操作数类型及范围

操作数	范 围	类 型
位 bit	I, Q, M, SM, T, C, V, S, L	BOOL 型
数量 N	VB, I, QB, SMB, LB, SB, AC, *VD, *AC, *LD, 常数	BYTE 型

【例 4-3】图 4-5 为置位和复位指令应用程序片断及对应的时序图。

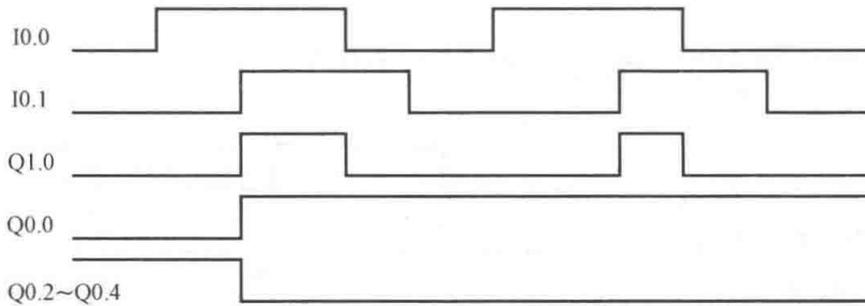
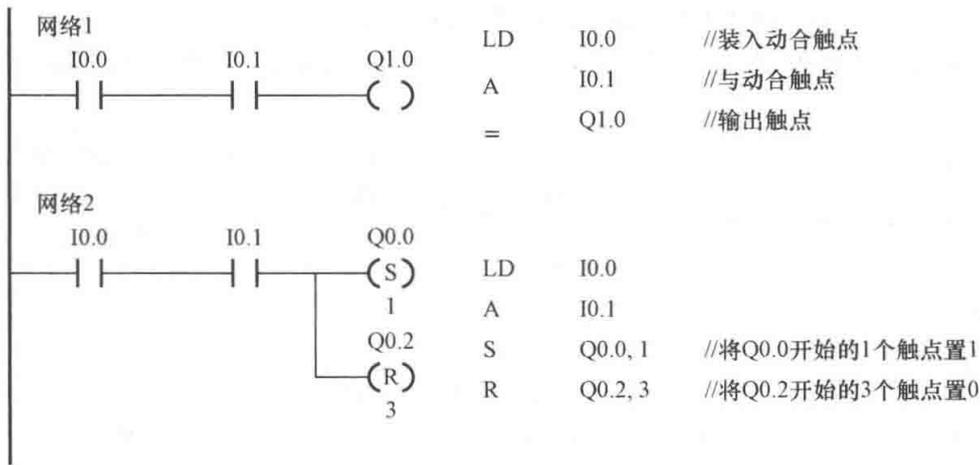


图 4-5 置位和复位指令应用程序片断及对应的时序图

### 3. RS 触发器和 SR 触发器

RS 触发器和 SR 触发器在梯形图中的指令盒形式如图 4-6 所示。

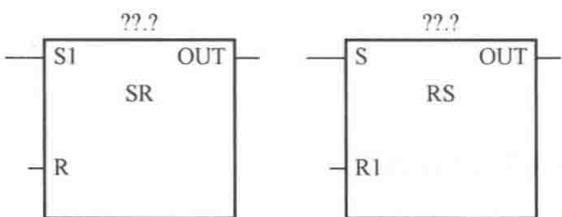


图 4-6 RS 触发器和 SR 触发器在梯形图中的指令盒形式

置位优先触发器 (SR): 当置位端 (S1) 和复位端 (R) 均为 “1” 时, 输出位 bit 为 “1”。

复位优先触发器 (RS): 当置位端 (S) 和复位端 (R1) 均为 “1” 时, 输出位 bit 为 “0”。

对 RS 触发器和 SR 触发器, 当置位端为 “1”、复位端为 “0” 时, 输出位 bit 为 “1”; 当置位端为 “0”、复位端为 “1” 时, 输出位 bit 为 “0”; 当置位端、复位端均为 “0” 时, 输出位保持原状态不变。

但在实际的程序设计中，RS 或 SR 触发器通常由置位、复位指令实现。

#### \* 4. 立即指令

(1) 立即触点指令。在每个标准触点指令的后面加“I”。指令执行时，立即读取物理输入点的值，但是不刷新对应映像寄存器的值。

这类指令包括 LDI、LDNI、AI、ANI、OI 和 ONI。

用法：LDI bit

bit 只能是 I 类型。

(2) 立即输出指令。执行指令时，新值被写入实际输出和对应的过程映像寄存器位置。这与非立即输出不同，非立即输出指令仅将新值写入过程映像寄存器。用立即输出指令访问输出点时，相应的输出映像寄存器的内容也被刷新。

用法：=Ibit

(3) SI，立即置位指令。bit 只能是 Q 类型。

用立即置位指令访问输出点时，从指令所指出的位 (bit) 开始的 N 个 (最多为 128 个) 物理输出点被立即置位，同时，相应的输出映像寄存器的内容也被刷新。

用法：SI bit, N

例：SI Q0.0, 2

(4) RI，立即复位指令。用立即复位指令访问输出点时，从指令所指出的位 (bit) 开始的 N 个 (最多为 128 个) 物理输出点被立即复位，同时，相应的输出映像寄存器的内容也被刷新。

用法：RI bit, N

例：RI Q0.0, 1

注意：bit 只能是 Q 类型。

**【例 4-4】**图 4-7 为立即指令应用程序片断。

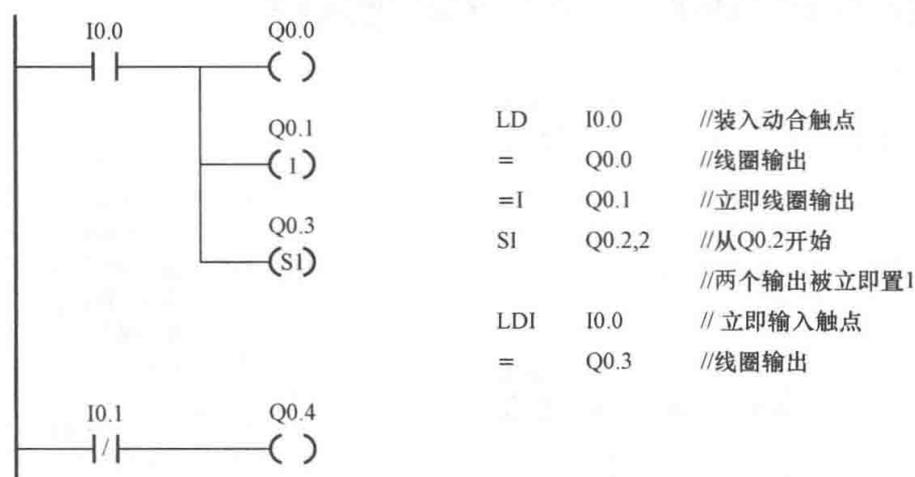


图 4-7 立即指令应用程序片断

### 4.1.5 复杂逻辑指令

#### 1. 块“或”操作指令格式

块“或”操作指令格式：OLD (无操作元件)。

块“或”操作是将梯形图中相邻的两个以 LD 起始的电路块并联起来。

## 2. 块“与”操作指令格式

块“与”操作指令格式：ALD (无操作元件)。

块“与”操作是将梯形图中相邻的两个以 LD 起始的电路块串联起来。

**【例 4-5】**块“或”、块“与”指令的应用举例，图 4-8 为梯形图程序。

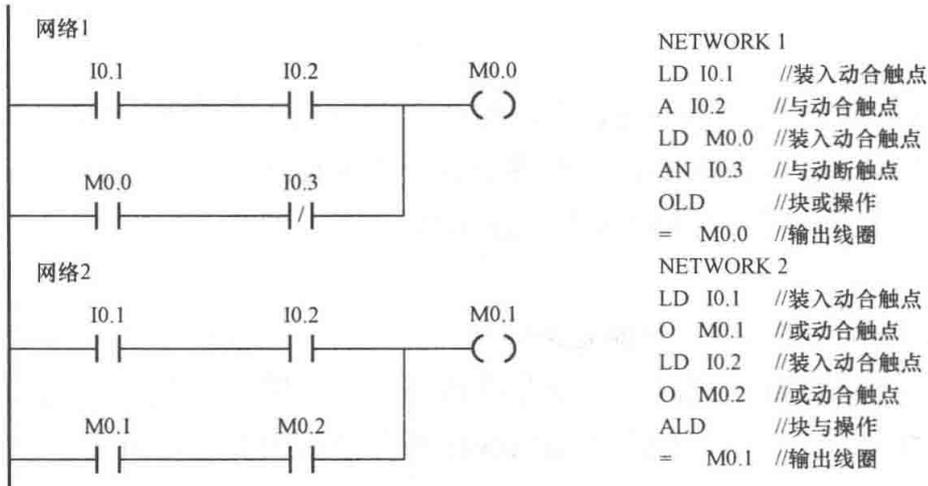


图 4-8 块或块非指令的应用

## 3. 栈操作指令

LD 装载指令是从梯形图最左侧母线画起的，如果要生成一条分支的母线，则需要利用语句表的栈操作指令来描述。

栈操作语句表指令格式：

LPS (无操作元件)：(Logic Push) 逻辑堆栈操作指令。

LRD (无操作元件)：(Logic Read) 逻辑读栈指令。

LPP (无操作元件)：(Logic Pop) 逻辑弹栈指令。

**【例 4-6】**栈操作指令应用程序，图 4-9 为梯形图程序。

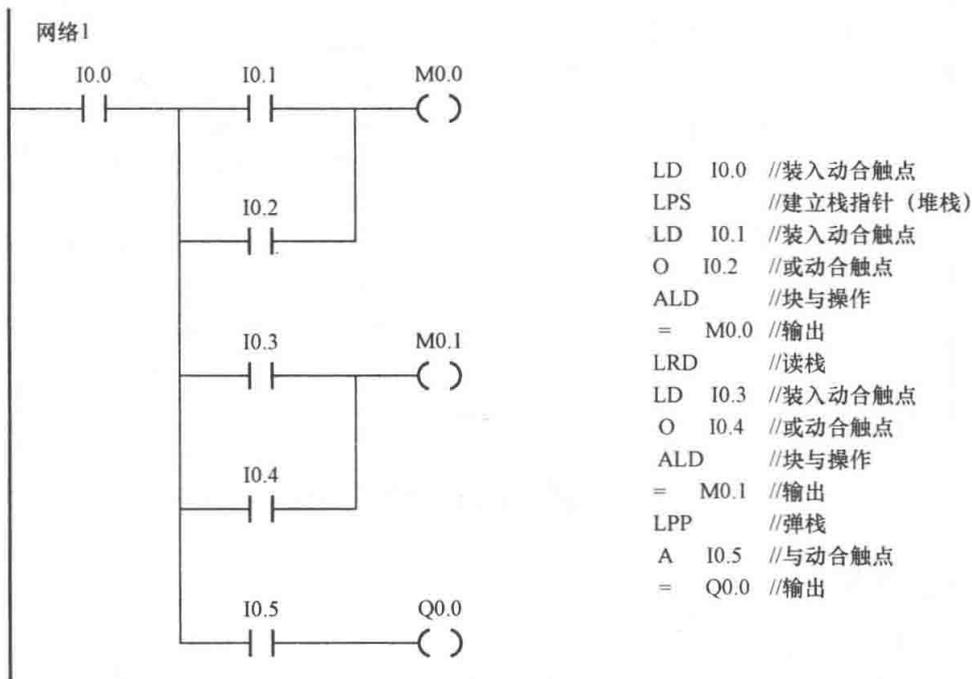


图 4-9 栈操作指令的应用

### 4.1.6 定时器指令

系统提供3种定时指令：TON、TONR和TOF。S7-200定时器的精度（时间增量/时间单位/分辨率）有3个等级：1ms、10ms和100ms。图4-10为三种定时器的梯形图指令盒形式。

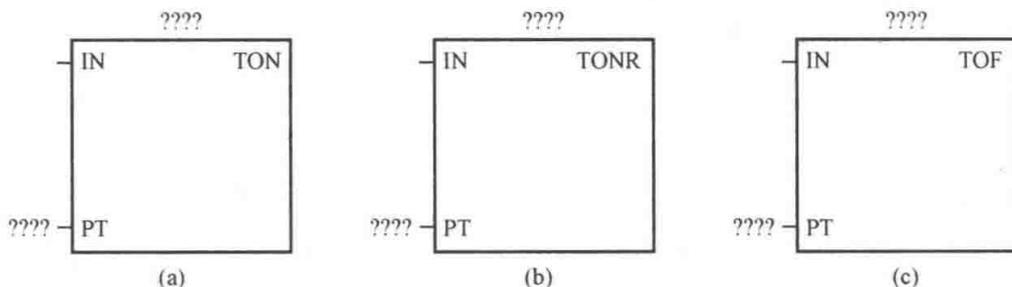


图4-10 3种定时器的梯形图指令盒形式

(a) 接通延时定时器；(b) 有记忆接通延时定时器；(c) 断开延时定时器

#### 1. 接通延时定时器

TON，接通延时定时器指令，用于单一间隔的定时。上电周期或首次扫描，定时器位OFF，当前值为0。使能输入接通时，定时器位为OFF，当前值从0开始计数时间，当前值达到预设值时，定时器位ON，当前值连续计数到32767。使能输入断开，定时器自动复位，即定时器位OFF，当前值为0。

指令格式：TON T<sub>xxx</sub>, PT

#### 2. 有记忆接通延时定时器

TONR，有记忆接通延时定时器指令，用于对许多间隔的累计定时。上电周期或首次扫描，定时器位OFF，当前值保持。使能输入接通时，定时器位为OFF，当前值从0开始计数时间。使能输入断开，定时器位和当前值保持最后状态。使能输入再次接通时，当前值从上次的保持值继续计数，当累计当前值达到预设值时，定时器位ON，当前值连续计数到32767。

TONR定时器只能用复位指令进行复位操作。

指令格式：TONR T<sub>xxx</sub>, PT

#### 3. 断开延时定时器

TOF，断开延时定时器指令，用于断开后的单一间隔定时。上电周期或首次扫描，定时器位OFF，当前值为0。使能输入接通时，定时器位为ON，当前值为0。当使能输入由接通到断开时，定时器开始计数，当前值达到预设值时，定时器位OFF，当前值等于预设值，停止计数。

TOF复位后，如果使能输入再有从ON到OFF的负跳变，则可实现再次启动。

指令格式：TOF T<sub>xxx</sub>, PT

**【例4-7】**图4-11是介绍3种定时器的工作特性的程序片断，其中T35为通电延时定时器，T2为有记忆通电延时定时器，T36为断电延时定时器。

本梯形图程序中输入输出执行时序关系如图4-12所示。

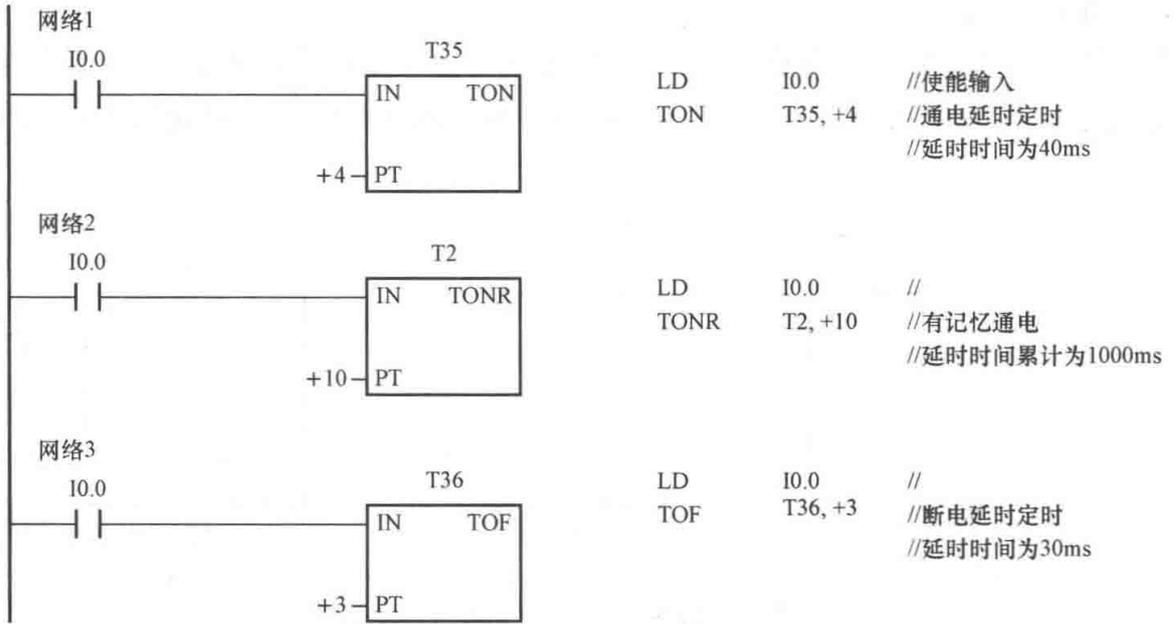


图 4-11 定时器指令的应用

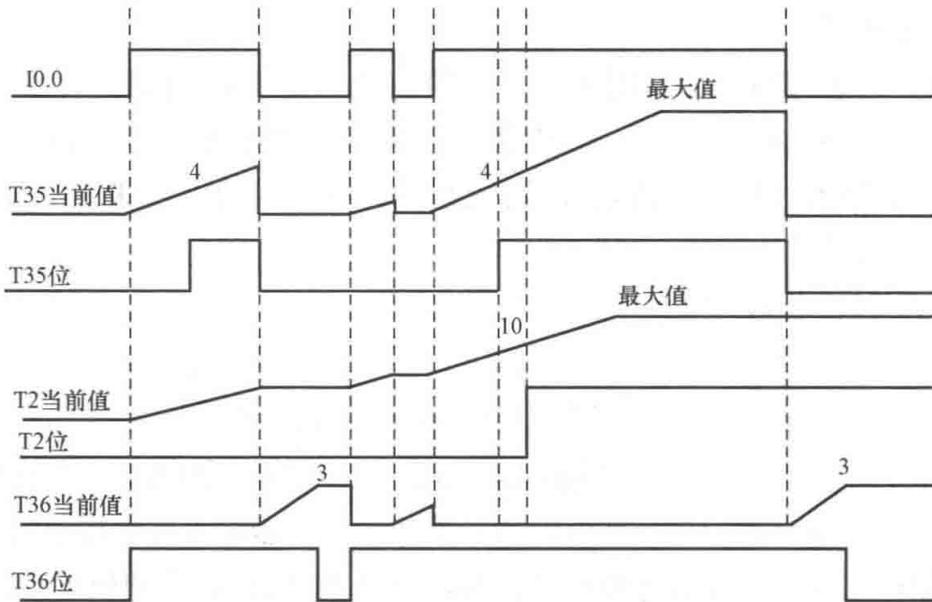


图 4-12 时序图

### 4.1.7 计数器指令

#### 1. 概述

计数器用来累计输入脉冲的次数。计数器也是由集成电路构成的，是应用非常广泛的编程元件，经常用来对产品进行计数。

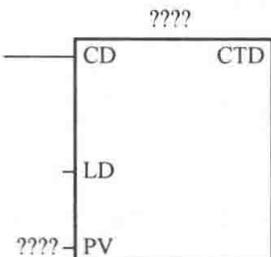


图 4-13 增计数器指令梯形图的指令盒形式

计数器指令有 3 种：增计数 CTU、增减计数 CTUD 和减计数 CTD。

指令操作数有 4 方面：编号、预设值、脉冲输入和复位输入。

#### 2. 增计数器

CTU，增计数器指令，梯形图的指令盒形式如图 4-13 所示。首次扫描，定时器位 OFF，当前值为 0。脉冲输入的每个上升沿，计数器计数 1 次，当前值增加 1 个单位，当前值达到预设值 PV 时，

计数器位 ON，当前值继续计数到 32767 停止计数。复位输入有效或执行复位指令，计数器自动复位，即计数器位 OFF，当前值为 0。

指令格式：CTU Cxxx, PV

例：CTU C20, 3

【例 4-8】图 4-14 所示为增计数器的程序片断和时序图。

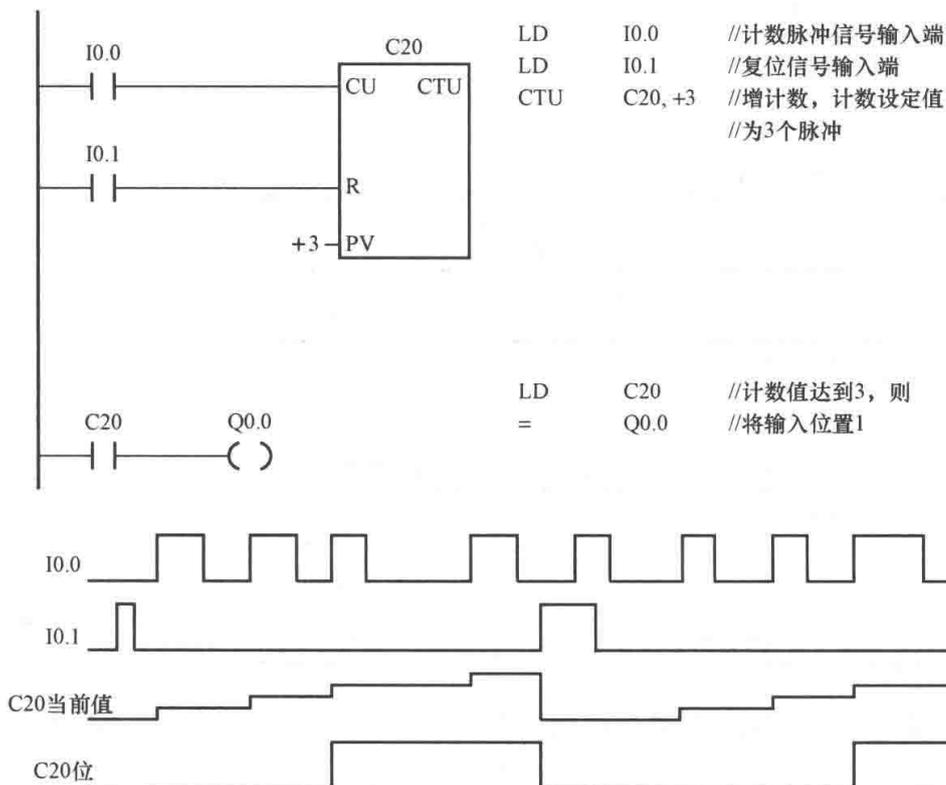


图 4-14 增计数器程序和时序图

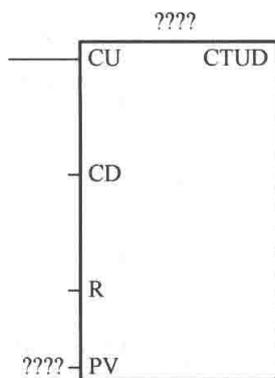
### 3. 增减计数器

CTUD，增减计数器指令，梯形图的指令盒形式如图 4-15 所示。有两个脉冲输入端：CU 输入端用于递增计数，CD 输入端用于递减计数。

指令格式：CTUD Cxxx, PV

例：CTUD C30, 5

【例 4-9】如图 4-16 所示为增减计数器的程序片断和时序图。



### 4. 减计数器

CTD，减计数器指令，梯形图的指令盒形式如图 4-17 所示。图 4-15 增减计数器指令脉冲输入端 CD 用于递减计数。首次扫描，定时器位 OFF，当前值为等于预设值 PV。计数器检测到 CD 输入的每个上升沿时，计数器当前值减小 1 个单位，当前值减到 0 时，计数器位 ON。

复位输入有效或执行复位指令，计数器自动复位，即计数器位 OFF，当前值复位为预设值，而不是 0。

指令格式：CTD Cxxx, PV

例：CTD C40, 4

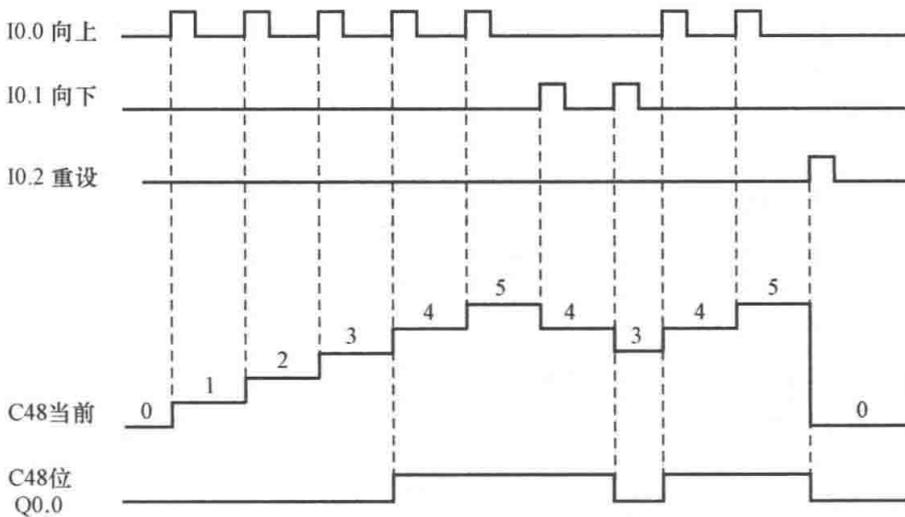
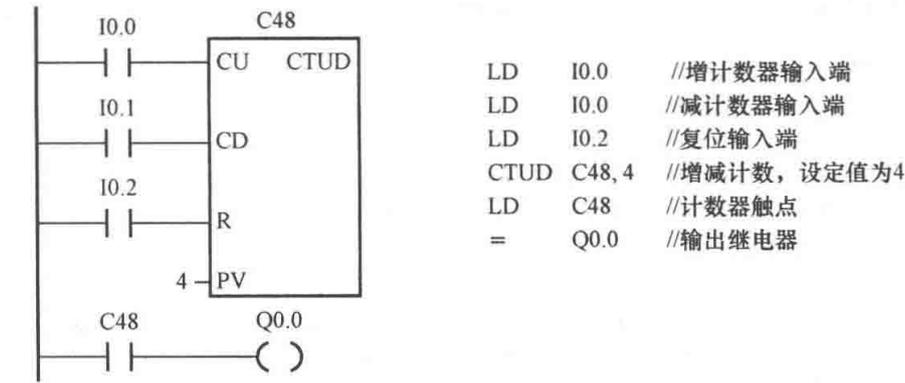


图 4-16 增减计数器程序和时序图

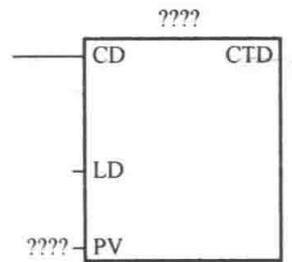


图 4-17 减计数器指令梯形图的指令盒形式

【例 4-10】图 4-18 为减计数器的程序片断和时序图。

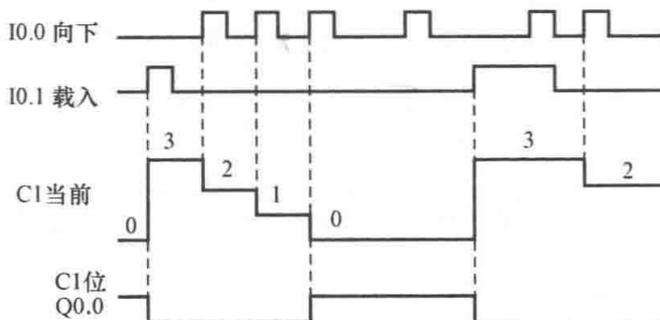
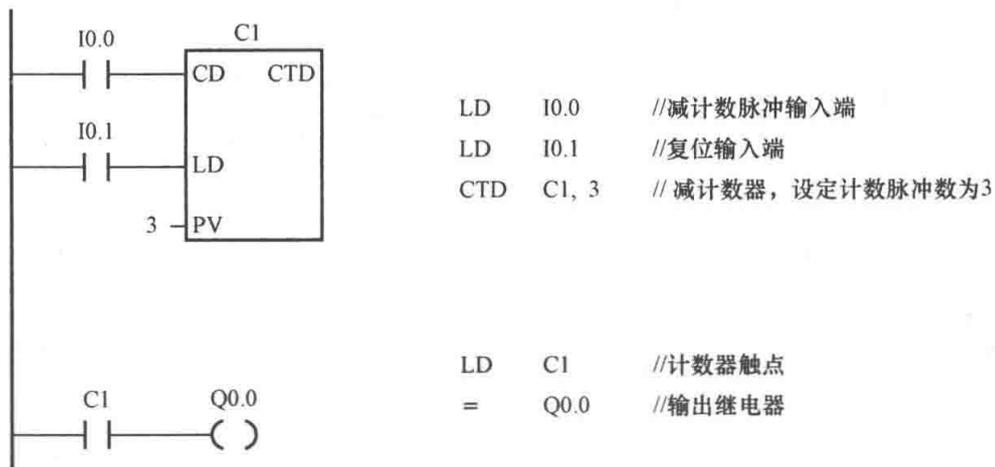


图 4-18 减计数器程序和时序图

5. 应用举例

(1) 循环计数。以上三种类型的计数器如果在使用时，将计数器位的动合触点作为复位输入信号，则可以实现循环计数。

(2) 长延时实现。用计数器和定时器配合增加延时时间，如图 4-19 所示。试分析以下程序中实际延时为多长时间。

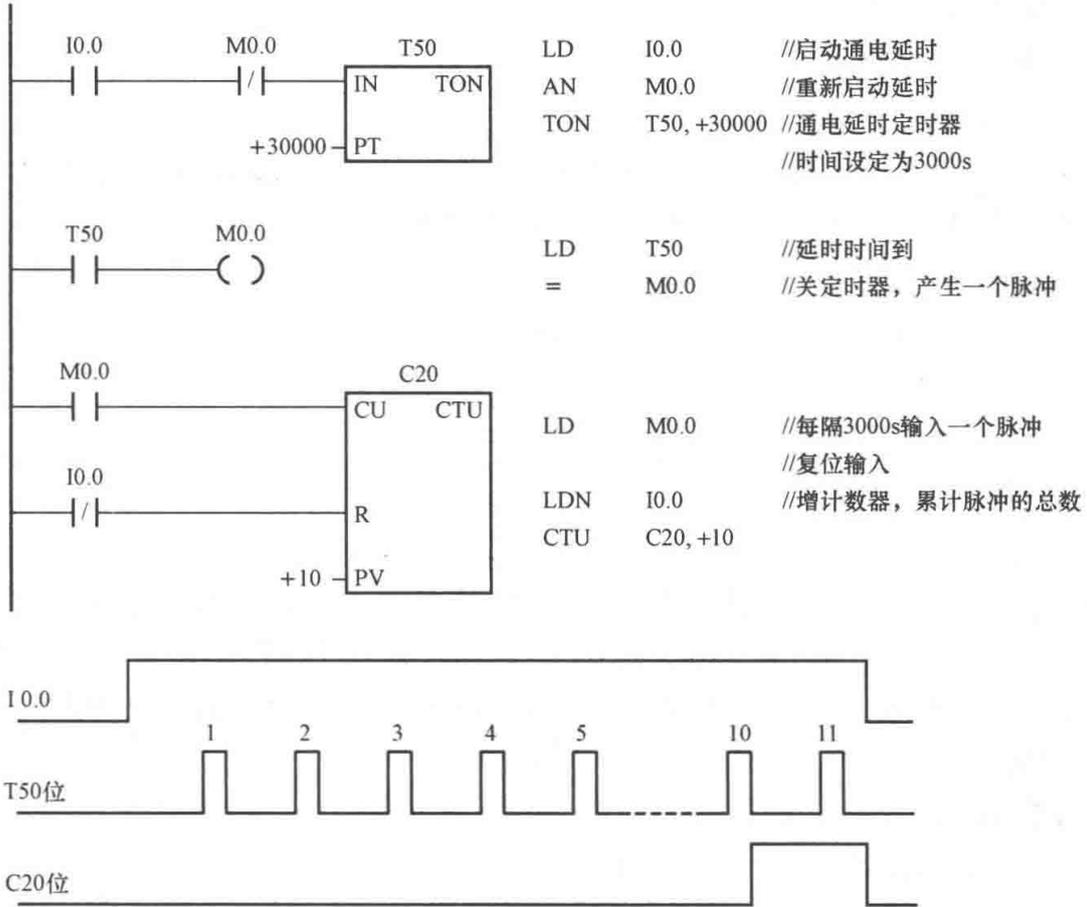


图 4-19 长延时梯形图程序和时序图

4.1.8 比较指令

1. 字节比较

字节比较用于比较两个字节型整数值 IN1 和 IN2 的大小，字节比较是无符号的。比较式可以是 LDB、AB 或 OB 后直接加比较运算符构成。

如：LDB=、AB<>、OB>= 等。

整数 IN1 和 IN2 的寻址范围：VB、IB、QB、MB、SB、SMB、LB、\* VD、\* AC、\* LD和常数。

指令格式例：

LDB= VB10, VB12

AB<> MB0, MB1

OB<= AC1, 116

2. 整数比较

整数比较用于比较两个一字长整数值 IN1 和 IN2 的大小，整数比较是有符号的（整数范围为 16#8000 至 16#7FFF）。比较式可以是 LDW、AW 或 OW 后直接加比较运算符构成。

如: LDW=、AW<>、OW>= 等。

整数 IN1 和 IN2 的寻址范围: VW、IW、QW、MW、SW、SMW、LW、AIW、T、C、AC、\* VD、\* AC、\* LD 和常数。

指令格式例:

LDW=VW10, VW12

AW<>MW0, MW4

OW<=AC2, 1160

### 3. 双字整数比较

双字整数比较用于比较两个双字长整数值 IN1 和 IN2 的大小, 双字整数比较是有符号的 (双字整数范围为 16#80000000 至 16#7FFFFFFF)。

指令格式例:

LDD=VD10, VD14

AD<>MD0, MD8

OD<=AC0, 1160000

LDD>=HC0, \* AC0

### 4. 实数比较

实数比较用于比较两个双字长实数值 IN1 和 IN2 的大小, 实数比较是有符号的, 负实数范围为  $-1.175495E-38$  至  $-3.402823E+38$ , 正实数范围为  $+1.175495E-38$  至  $+3.402823E+38$ 。比较式可以是 LDR、AR 或 OR 后直接加比较运算符构成。

指令格式例:

LDR=VD10, VD18

AR<>MD0, MD12

OR<=AC1, 1160.478

AR>\* AC1, VD100

### 5. 应用举例

控制要求: 一自动仓库存放某种货物, 最多 6000 箱, 需对所存的货物进出计数。货物多于 1000 箱, 灯 L1 亮; 货物多于 5000 箱, 灯 L2 亮。其中, L1 和 L2 分别受 Q0.0 和 Q0.1 控制, 数值 1000 和 5000 分别存储在 VW20 和 VW30 字存储单元中。

本控制系统的程序如图 4-20 所示。程序执行时序如图 4-21 所示。

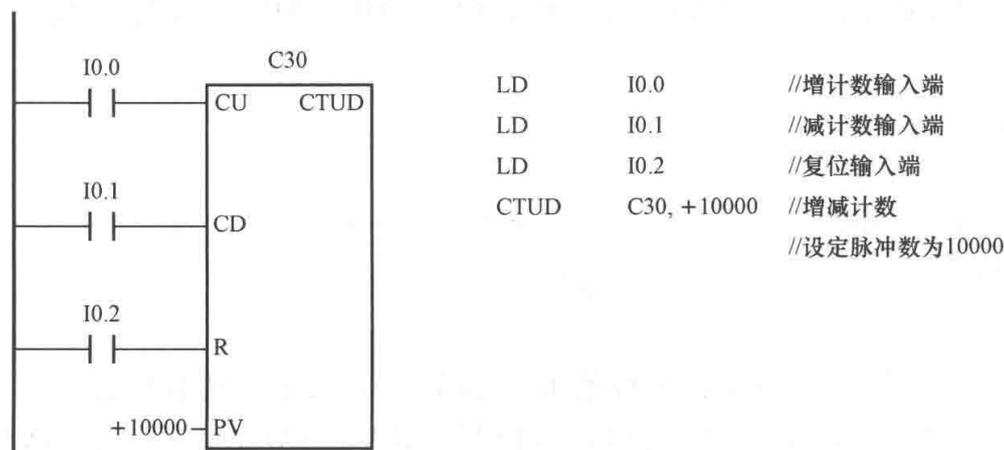


图 4-20 控制程序 (一)

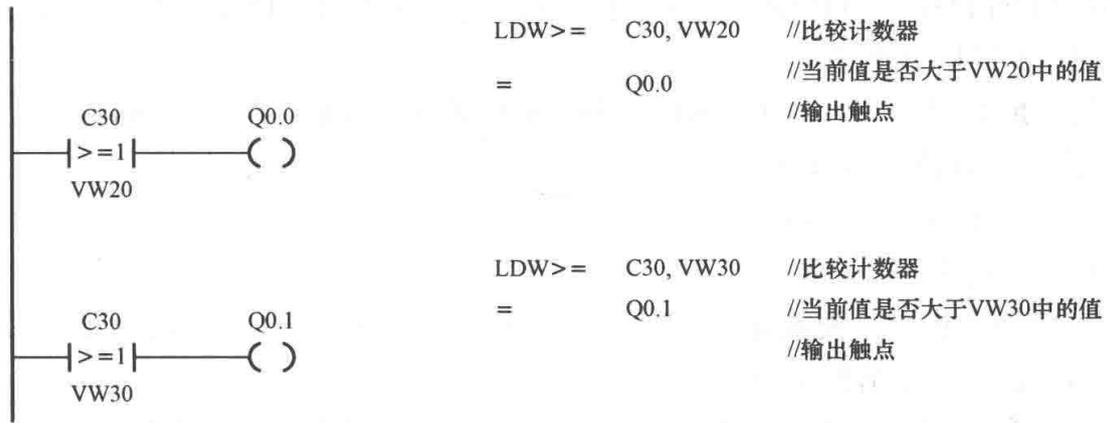


图 4-20 控制程序 (二)

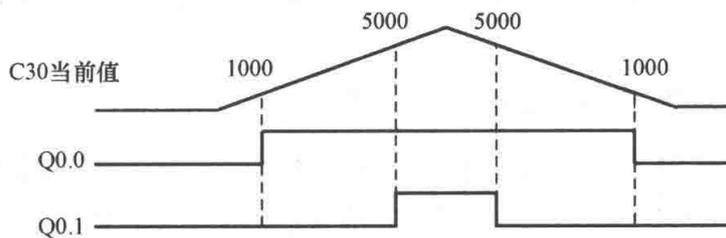


图 4-21 时序图

## 4.2 运算指令

S7-200 PLC 的运算指令包括算术运算和逻辑运算两大类指令。算术运算指令有加法、减法、乘法、除法和数学函数、增 1 减 1 指令。逻辑运算指令有逻辑与、或、非、异或。数据类型为字节、字、双字。

### 4.2.1 加法

#### 1. 整数加法

+I, 整数加法指令, 梯形图的指令盒形式如图 4-22 所示。使能输入有效时, 将两个单字长 (16 位) 的符号整数 IN1 和 IN2 相加, 产生一个 16 位整数结果 OUT。

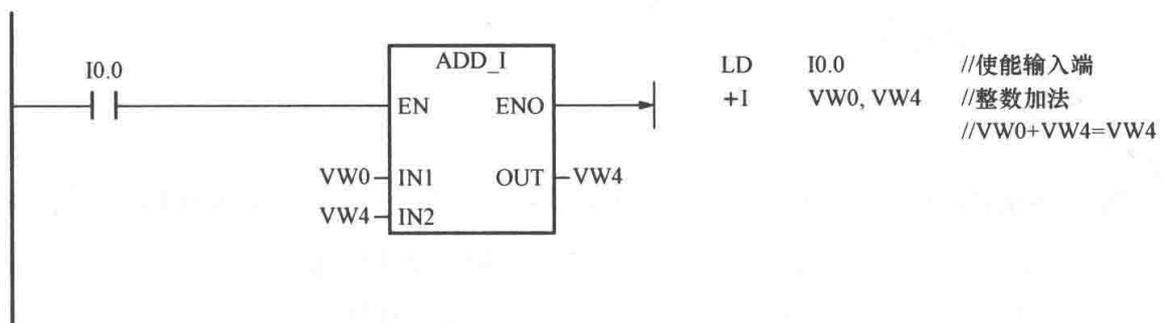


图 4-22 加法指令程序图

#### 2. 双整数加法

+D, 双整数加法指令。使能输入有效时, 将两个双字长 (32 位) 的符号双整数 IN1 和 IN2 相加, 产生一个 32 位双整数结果 OUT。

在 LAD 和 FBD 中, 以指令盒形式编程, 执行结果:  $IN1 + IN2 = OUT$ 。在 STL 中, 执行结果:  $IN1 + OUT = OUT$ 。

OUT 的寻址范围: VD、ID、QD、MD、SD、SMD、LD、AC、\*VD、\*AC、\*LD。

指令格式: +D IN1, OUT

例: +D VD0, VD4

### 3. 实数加法指令

+R, 实数加法指令。使能输入有效时, 将两个双字长 (32 位) 的实数 IN1 和 IN2 相加, 产生一个 32 位实数结果 OUT。

在 LAD 和 FBD 中, 以指令盒形式编程, 执行结果:  $IN1 + IN2 = OUT$ 。

OUT 的寻址范围: VD、ID、QD、MD、SD、SMD、LD、AC、\*VD、\*AC、\*LD。

本指令影响的特殊存储器位: SM1.0 (零); SM1.1 (溢出); SM1.2 (负)。

## 4.2.2 减法

减法指令的梯形图指令盒形式如图 4-23 所示。

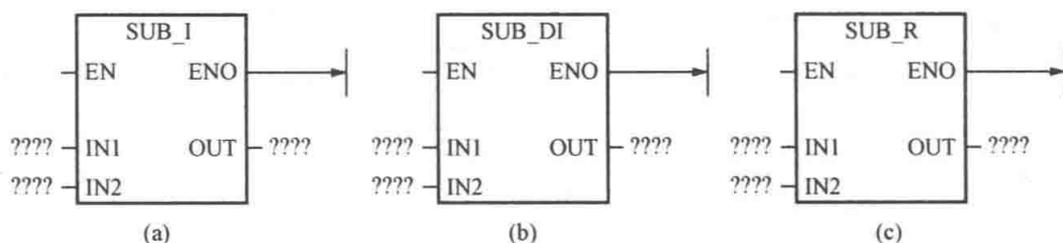


图 4-23 减法指令的梯形图指令盒形式

(a) 整数减法; (b) 双整数减法; (c) 实数减法

减法指令是对有符号数进行相减操作, 包括整数减法、双整数减法和实数减法。这三种减法指令与所对应的加法指令除运算法则不同之外, 其他方面基本相同。

在 LAD 和 FBD 中, 以指令盒形式编程, 执行结果:  $IN1 - IN2 = OUT$ 。

在 STL 中, 执行结果:  $OUT - IN2 = OUT$ 。

指令格式: -I IN2, OUT (整数减法)

-DI IN2, OUT (双整数减法)

-RI IN2, OUT (实数减法)

例: -I AC0, VW4

## 4.2.3 乘法

### 1. 整数乘法

\* I, 整数乘法指令。使能输入有效时, 将两个单字长 (16 位) 的符号整数 IN1 和 IN2 相乘, 产生一个 16 位整数结果 OUT。

指令格式: \* I IN1, OUT

例: \* I VW0, AC0

整数乘法的梯形图指令盒形式如图 4-24 所示。

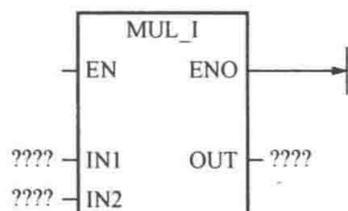


图 4-24 整数乘法的梯形图指令盒形式

### 2. 完全整数乘法

MUL, 完全整数乘法指令。使能输入有效时, 将两个单字长 (16 位) 的符号整数 IN1 和 IN2 相乘, 产生一个 32 位双整数结

果 OUT。

在 LAD 和 FBD 中，以指令盒形式编程，执行结果： $IN1 * IN2 = OUT$ 。

OUT 的寻址范围：VD、ID、QD、MD、SD、SMD、LD、AC、\* VD、\* AC、\* LD。

本指令影响的特殊存储器位：SM1.0（零）；SM1.1（溢出）；SM1.2（负）；SM1.3（被 0 除）。

指令格式：MUL IN1, OUT

例：MUL AC0, VD10

完全整数乘法的梯形图指令盒形式如图 4-25 所示。

### 3. 双整数乘法

\* D，双整数乘法指令。使能输入有效时，将两个双字长（32 位）的符号整数 IN1 和 IN2 相乘，产生一个 32 位双整数结果 OUT。

在 STL 中，执行结果： $IN1 * OUT = OUT$ 。

IN1 和 IN2 的寻址范围：VD、ID、QD、MD、SD、SMD、LD、HC、AC、\* VD、\* AC、\* LD 和常数。

OUT 的寻址范围：VD、ID、QD、MD、SD、SMD、LD、AC、\* VD、\* AC、\* LD。本指令影响的特殊存储器位：SM1.0（零）；SM1.1（溢出）；SM1.2（负）；SM1.3（被 0 除）。

指令格式：\* D IN1, OUT

例：\* D VD0, AC0

双整数乘法的梯形图指令盒形式如图 4-26 所示。

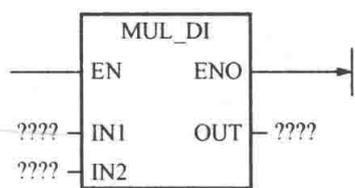


图 4-26 双整数乘法的梯形图指令盒形式

### 4. 实数乘法

\* R，实数乘法指令。使能输入有效时，将两个双字长（32 位）的实数 IN1 和 IN2 相乘，产生一个 32 位实数结果 OUT。

在 LAD 和 FBD 中，以指令盒形式编程，执行结果： $IN1 * IN2 = OUT$ 。

在 STL 中，执行结果： $IN1 * OUT = OUT$ 。

IN1 和 IN2 的寻址范围：VD、ID、QD、MD、SD、SMD、LD、AC、\* VD、\* AC、\* LD 和常数。

OUT 的寻址范围：VD、ID、QD、MD、SD、SMD、LD、AC、\* VD、\* AC、\* LD。

本指令影响的特殊存储器位：SM1.0（零）；SM1.1（溢出）；SM1.2（负）；SM1.3（被 0 除）。

指令格式：\* R IN1, OUT

例：\* R VD0, AC0

实数乘法的梯形图指令盒形式如图 4-27 所示。

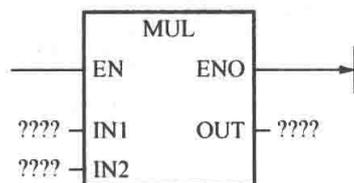


图 4-25 完全整数乘法的梯形图指令盒形式

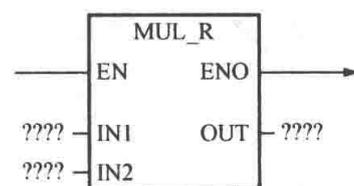


图 4-27 实数乘法的梯形图指令盒形式

#### 4.2.4 除法

除法指令的梯形图指令盒形式如图 4-28 所示。

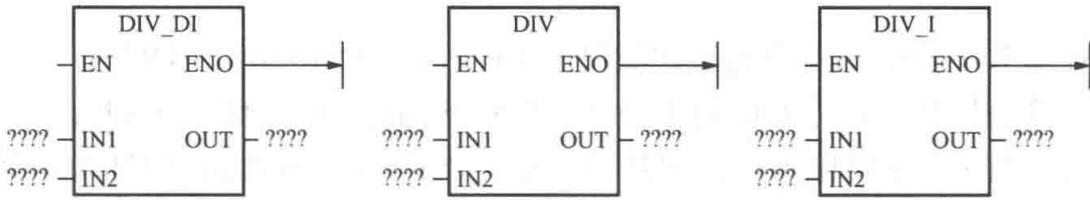


图 4-28 除法指令的梯形图指令盒形式

在 LAD 和 FBD 中, 以指令盒形式编程, 执行结果:  $IN1/IN2=OUT$ 。

在 STL 中, 执行结果:  $OUT/IN2=OUT$ 。

指令格式: `/I IN2, OUT` (整数除法)

`DIV IN2, OUT` (整数完全除法)

`/D IN2, OUT` (双整数除法)

`/R IN2, OUT` (实数除法)

例: `DIV VW10, VD100`

`/I VW20, VW200`

两条指令的编程及执行情况比较如图 4-29 所示。

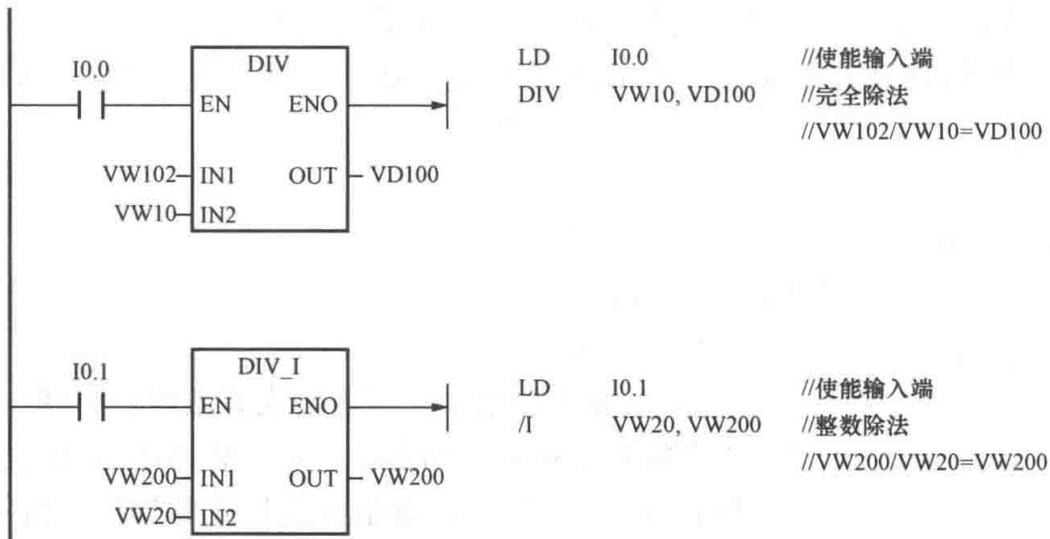


图 4-29 除法指令

#### 4.2.5 数学函数指令

##### 1. 平方根、自然对数、指数

平方根指令是把一个双字长 (32 位) 的实数 IN 开平方, 得到 32 位的实数结果。通过 (OUT) 指定的存储器单元输出。

自然对数指令是将一个双字长 (32 位) 的实数 IN 取自然对数, 得到 32 位的实数结果。

指数指令是将一个双字长 (32 位) 的实数 IN 取以 e 为底的指数, 得到 32 位的实数结果 OUT。IN、OUT 操作数的数据类型为 REAL。

平方根指令的指令格式: `SQRT IN, OUT` //求平方根指令  $SQRT(IN) = OUT$

自然对数指令的指令格式: `LN IN, OUT`

指数指令的指令格式: `EXP IN, OUT`

平方根/自然对数/指数指令的梯形图指令盒形式如图 4-30 所示。

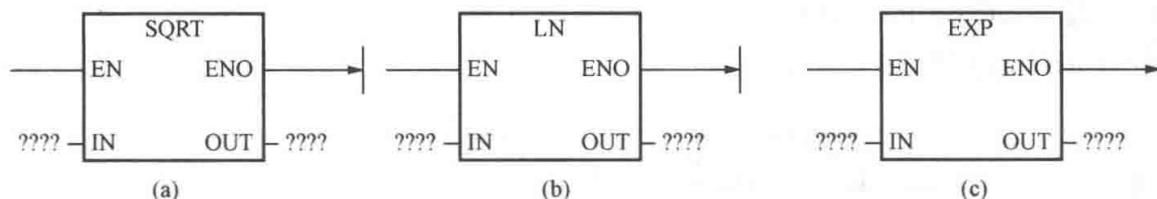


图 4-30 平方根、自然对数、指数指令的梯形图指令盒形式

(a) 平方根; (b) 自然对数; (c) 指数

**【例 4-11】** 求以 10 为底, 150 的常用对数, 150 存于 VD100, 结果放到 AC1 (应用对数的换底公式求解)。

当求解以 10 为底的常用对数时, 用实数除法指令将自然对数除以 2.302585 即可。(LN10 $\approx$ 2.302585)

程序如图 4-31 所示。

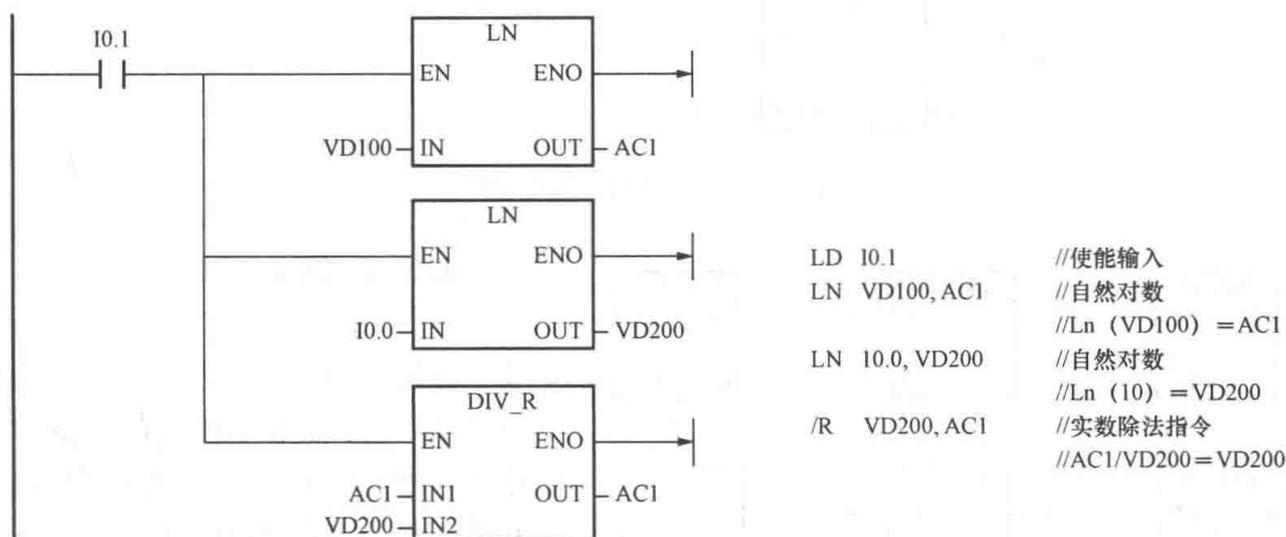


图 4-31 自然对数指令应用

## 2. 正弦、余弦、正切

SIN、COS、TAN, 即正弦、余弦、正切指令。将一个双字长 (32 位) 的实数弧度值 IN 分别取正弦、余弦、正切, 各得到 32 位的实数结果。

如果已知输入值为角度, 要先将角度值转化为弧度值, 方法: 使用 (\*R) MUL\_R 指令用角度值乘以  $\pi/180^\circ$  即可。IN、OUT 操作数的数据类型为 REAL。

指令格式如下:

```

SIN IN,OUT //SIN(IN)=OUT
COS IN,OUT // COS(IN)=OUT
TAN IN,OUT // TAN(IN)=OUT
    
```

正弦、余弦、正切指令的梯形图指令盒形式如图 4-32 所示。

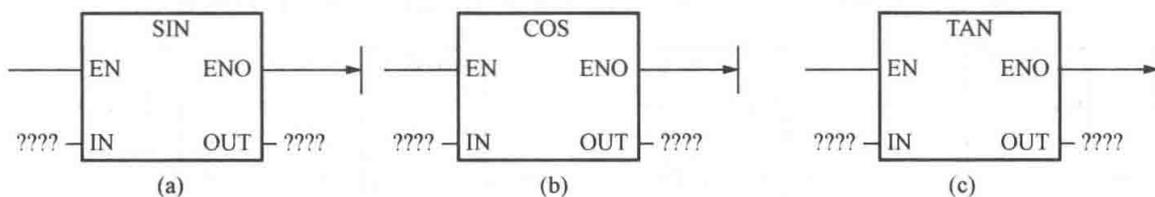


图 4-32 正弦、余弦、正切指令的梯形图指令盒形式

(a) 正弦; (b) 余弦; (c) 正切

**【例 4-12】** 求  $65^\circ$  的正切值。

设计的梯形图程序如图 4-33 所示。

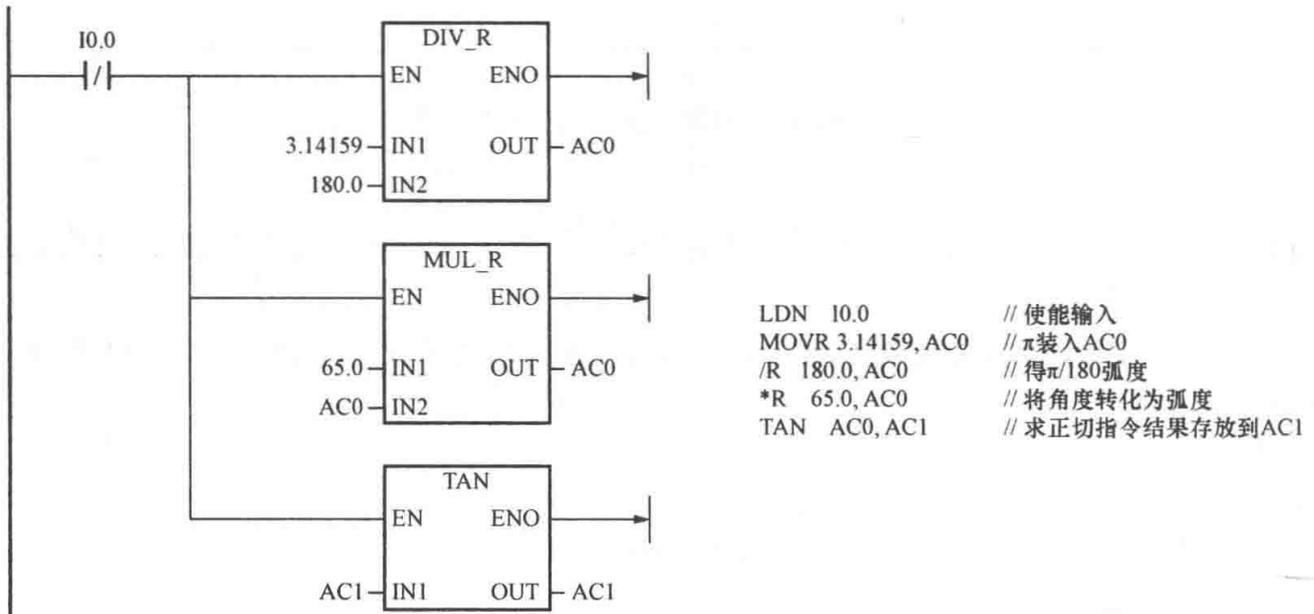


图 4-33 三角函数指令应用

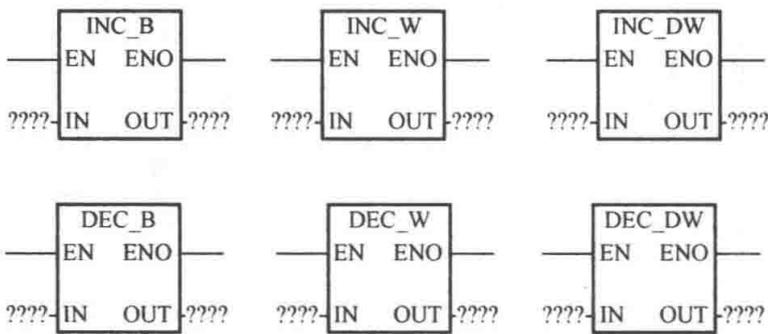


图 4-34 增 1/减 1 指令的梯形图指令盒形式

#### 4.2.6 增 1/减 1 指令

增 1/减 1 指令 (或增 1/减 1 计数器) 在输入字节 (IN) 上加 1 或减 1, 并将结果置入 OUT 指定的变量中。它用于自增、自减操作, 以实现累加计数和循环控制等程序的编制。它分为字节、字、双字增 1 和字节、字、双字减 1。增 1/减 1 指令的梯形图指令盒形式如图 4-34 所示。

#### 4.2.7 逻辑运算

逻辑运算是对无符号数进行的逻辑处理, 主要包括逻辑与、逻辑或、逻辑异或和取反等运算指令。逻辑运算梯形图指令盒形式如图 4-35 所示。

**与 (或、异或) 指令功能:** 使能输入有效时, 把两个字节 (字、双字) 长的输入逻辑数按位相与 (或、异或), 得到的一个字节 (字、双字) 逻辑运算结果, 送到 OUT 指定的存储器单元输出。

**取反指令功能:** 使能输入有效时, 将一个字节 (字、双字) 长的逻辑数按位取反, 得到的一个字节 (字、双字) 逻辑运算结果, 送到 OUT 指定的存储器单元输出。

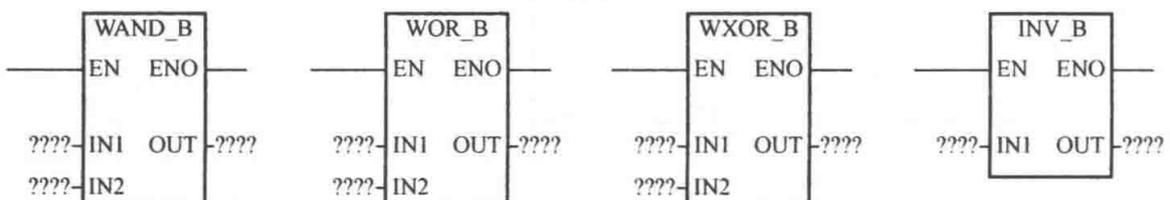


图 4-35 逻辑运算的梯形图指令盒形式

**【例 4-13】** 字或/双字异或/字求反/字节与操作编程举例。

编程应用如图 4-36 所示。

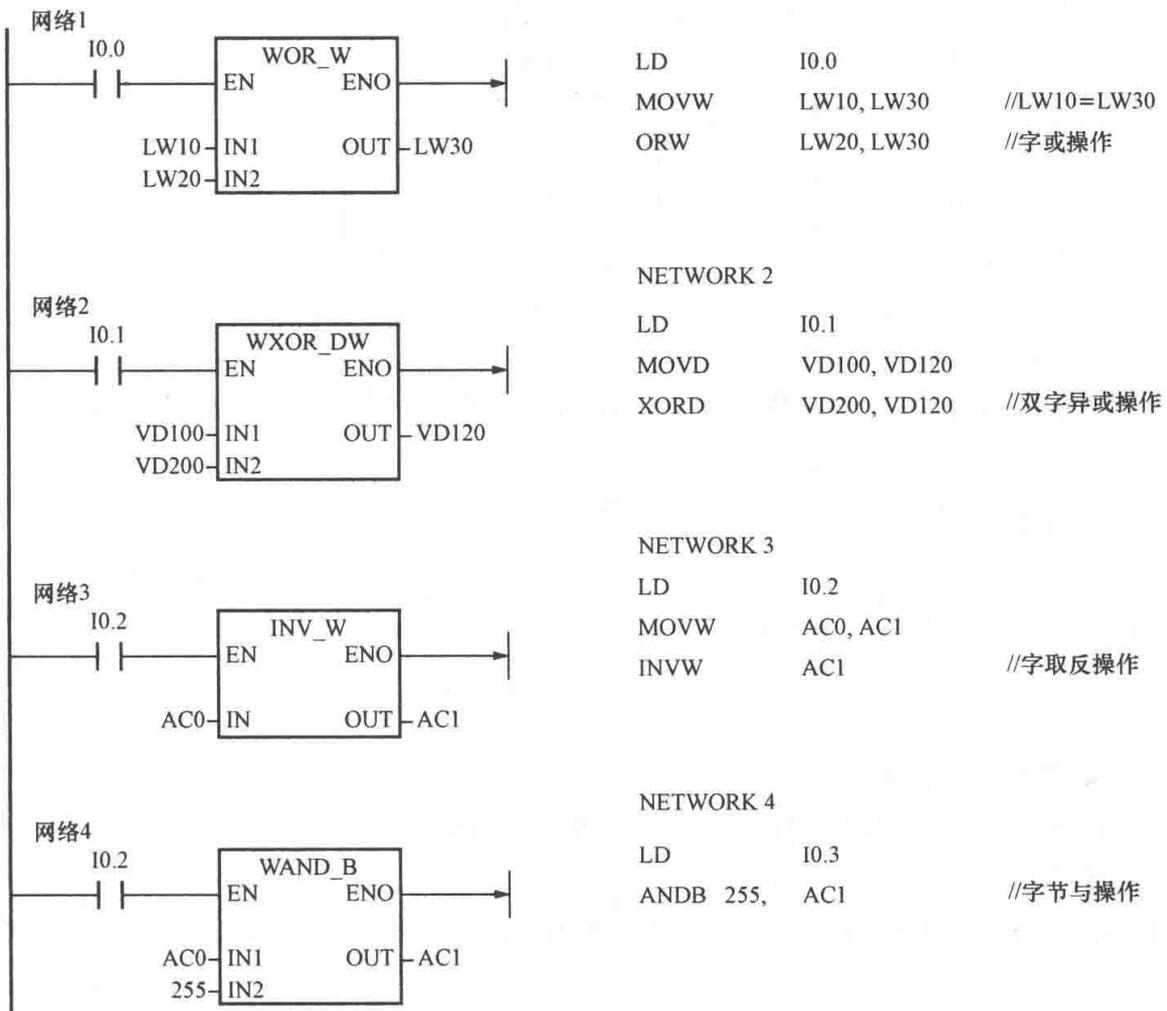


图 4-36 逻辑运算指令应用

### 4.3 数据处理指令

数据处理指令包括数据传送指令、字节交换/填充指令、移位指令等。

#### 4.3.1 数据传送指令

##### 1. 单个数据传送

单个数据传送指令包括字节、字和双字传送指令。梯形图中的指令盒形式如图 4-37 所示。

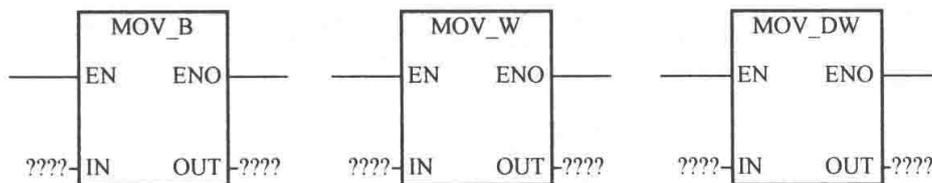


图 4-37 单个数据传送指令的梯形图指令盒形式

指令格式为：MOV IN, OUT

##### 2. 数据块传送

指令类型：字节、字或双字的 N 个数据成组传送。梯形图中的指令盒形式如图 4-38

所示。

功能：使能输入 (EN) 有效时，把从输入 (IN) 字节开始的 N 个字节数据传送到以输出字节 (OUT) 开始的 N 个字节、字或双字中。

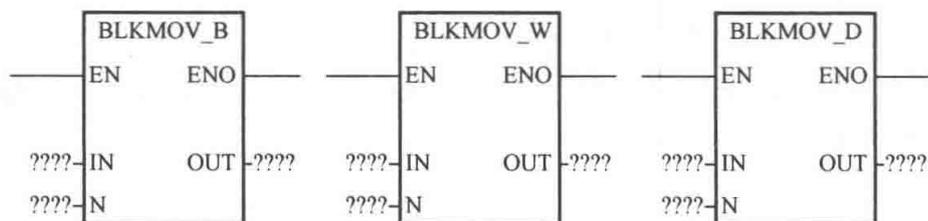


图 4-38 数据块传送指令的梯形图指令盒形式

**【例 4-14】** 将变量存储器 VW100 中内容送到 VW200 中。梯形图程序如图 4-39 所示。

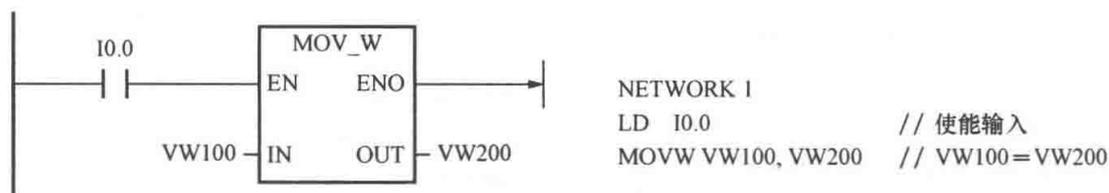


图 4-39 数据传送指令应用

### 4.3.2 字节交换/填充指令

字节交换/填充指令的梯形图指令盒形式如图 4-40 所示。

#### 1. 字节交换指令 (SWAP)

使能输入 (EN) 有效时，将输入字 (IN) 的高、低字节交换的结果输出到 (IN) 存储器单元。

#### 2. 字填充 (FILL IN, OUT, N)

使能输入 (EN) 有效时，字型输入数据 (IN) 填充从输出 (OUT) 指定单元开始的 N 个字存储单元。N (BYTE) 的数据范围为 0~255，如图 4-41 所示。

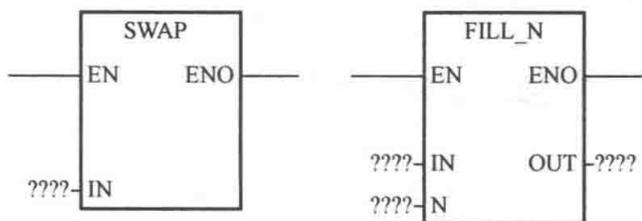


图 4-40 字节交换/填充指令的梯形图指令盒形式

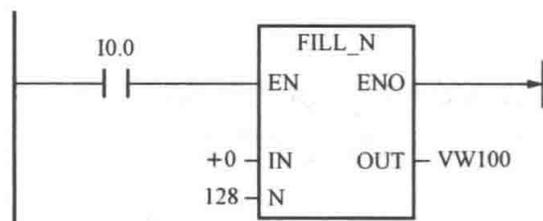


图 4-41 字填充指令应用

**【例 4-15】** 将从 VW100 开始的 256 个字节 (128 个字) 存储单元清零。

NETWORK 1

LD I0.0 //使能输入

FILL +0, VW100, 128 //128 个字填充 0

本条指令执行结果：从 VW100 开始的 256 个字节 (VW100~VW354) 的存储单元清零。

### 4.3.3 移位指令

移位指令分为左、右移位和循环左、右移位及寄存器移位指令 3 大类。左、右移位指令

和循环左、右移位指令的梯形图指令盒形式如图 4-42 所示。

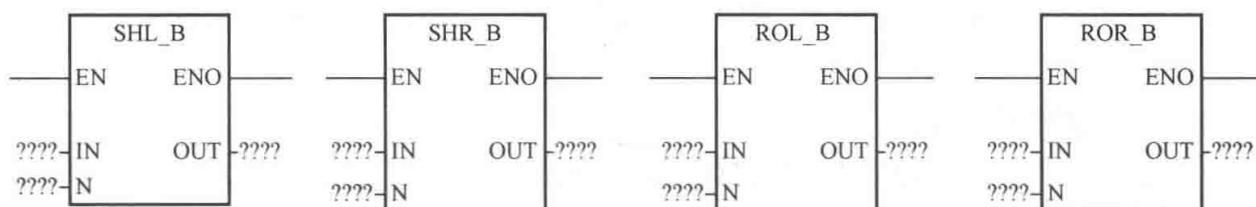


图 4-42 左右移位指令和循环左、右移位指令的梯形图指令盒形式

左、右移位和循环左、右移位指令按移位数据的长度又分为字节型、字型、双字型 3 种。

#### 1. 左、右移位指令

左、右移位指令 (SHL、SHR) 的功能: 使能输入有效时, 将输入的字节、字或双字 (IN) 左、右移  $N$  位后 (右、左端补 0), 将结果输出到 OUT 所指定的存储单元中, 最后一次移出位保存在 SM1.1。

#### 2. 循环左、右移位指令

循环左、右移位指令 (ROL、ROR) 的功能: 使能输入有效时, 字节、字或双字 (IN) 数据循环左移  $N$  位后, 将结果输出到 OUT 所指定的存储单元中, 并将最后一次移出位送 SM1.1。

#### 【例 4-16】 右移指令的应用示例。

最大移位位数  $N \leq$  数据类型 (B、W、D) 对应的位数, 移位位数 (次数)  $N$  为字节型数据。梯形图程序如图 4-43 所示。

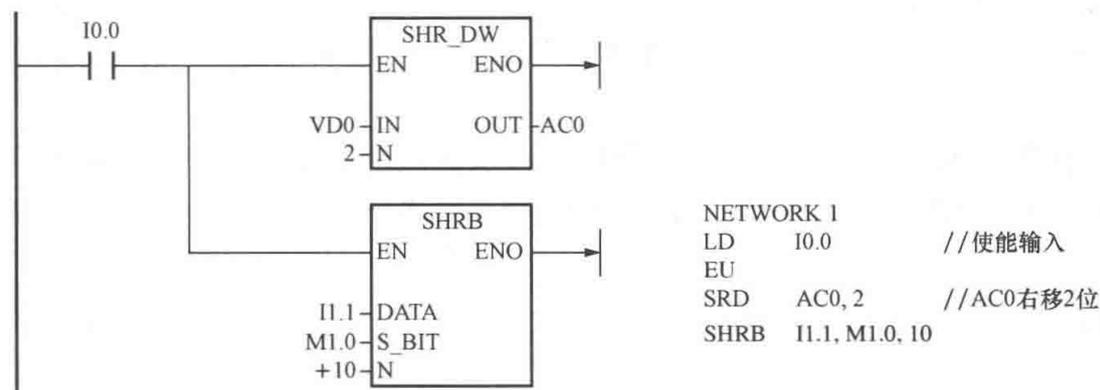


图 4-43 移位指令应用

#### \* 4.3.4 寄存器移位指令

指令将 DATA 数值移入移位寄存器。S\_BIT 指定移位寄存器的最低位。N 指定移位寄存器的长度和移位方向 (移位加 =  $N$ , 移位减 =  $-N$ )。

S\_BIT 为寄存器的最低位,  $N$  为移位寄存器的长度 (1~64)。每次使能有效时, 整个移位寄存器移动 1 位。

$N$  为正值时, 左移位 (由低位到高位), DATA 值从 S\_BIT 位移入, 移出位进入 SM1.1;  $N$  为负值时, 右移位 (由高位到低位), S\_BIT 移出到 SM1.1, 高端补充 DATA 移入位的值。

图 4-43 第 2 个梯形图指令盒对应的指令如下:

SHRB I1.1, M1.0, +10

该指令执行时,若 I0.1 为 ON,把 I1.1 的数值移入移位寄存器(由 M1.0 至 M2.1 所组成)中。

#### \* 4.3.5 数据转换指令

数据转换指令可分为数据类型的转换,编码与译码,段译码,ASCII 码与十六进制数的转换,整数、双整数、实数转换为 ASCII 码,整数、双整数、实数转换为字符串,子字符串转换为整数、双整数、实数这 7 类指令。经常使用的指令为数据类型的转换,这里不详细阐述,结合后面章节的应用例子进行讲解。

## 4.4 程序控制类指令

程序控制类指令包括系统控制、跳转、循环、子程序调用、中断服务程序等指令。

### 4.4.1 系统控制类指令

#### 1. 暂停指令 (STOP)

使能输入有效时,立即终止程序的执行。

#### 2. 结束指令 (END/MEND)

结束指令直接连在左侧电源母线时,为无条件结束指令 (MEND),通常在程序编辑时可以不写,因为编程软件在编译过程中会在主程序的末尾自动加入该指令。

不连在左侧母线时,条件结束指令 (END)。在主程序中也可以使用条件结束指令来提前结束主程序。条件结束指令 (END) 不能出现在子程序或中断服务程序中。

#### 3. AENO 与 ENO 指令

AENO 指令 (And ENO) 的作用是和前面的指令盒输出端 ENO 相与,只能在语句表中使用。

ENO 是梯形图和功能框图编程时指令盒的布尔能流输出端。如果指令盒的能流输入有效,同时执行没有错误,ENO 就置位,将能流向上传递。当用梯形图编程时,指令盒后串联一个指令盒或线圈。

指令格式: AENO (无操作数)

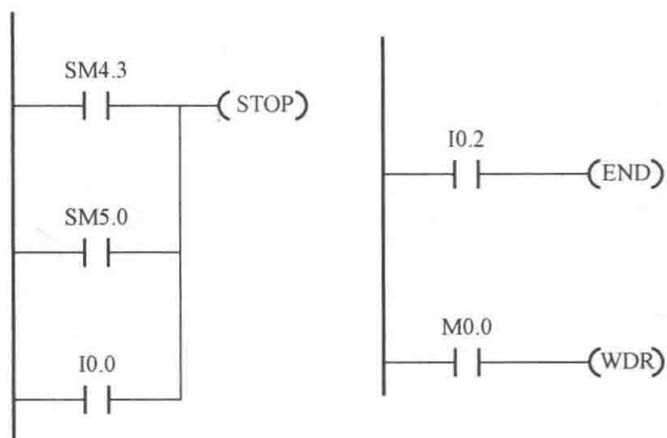


图 4-44 系统控制指令的应用

#### \* 4. 看门狗复位指令 (WDR)

看门狗又称为监控定时器 (Watchdog),它的定时时间为 500ms,PLC 正常工作时扫描周期少于 500ms,它不起作用。如果扫描周期超过 500ms,CPU 会自动切换到 STOP 模式。为此,可以在程序中使用 WDR 指令,当使能输入有效时,将看门狗定时器复位。在没有看门狗错误的情况下,可以扩展允许使用的扫描周期。

【例 4-17】暂停 (STOP)、条件结束 (END)、看门狗指令应用举例。如图 4-44 所示。

### 4.4.2 跳转、循环指令

#### 1. 程序跳转指令 (JMP)

JMP	n	跳转指令
LBL	n	跳转标号

跳转指令 (JMP) 和跳转地址标号指令 (LBL) 配合实现程序的跳转。使能输入有效时, 使程序跳转到指定标号 n 处执行 (在同一程序内), 跳转标号 n=0~255。使能输入无效时, 程序顺序执行。

#### 2. 循环控制指令 (FOR)

循环控制指令, 用于描述一段程序的重复循环执行。由 FOR 和 NEXT 指令构成程序的循环体。FOR 标记循环开始, NEXT 为循环体结束。

FOR 指令为指令盒格式, 主要参数有使能输入 EN、当前值计数器 INDX、循环次数初始值 INIT、循环计数终值 FINAL。

工作原理: 使能输入 (EN) 有效, 循环体开始执行, 执行到 NEXT 指令时返回, 每执行一次循环体, 当前计数器 (INDX) 增 1, 达到终值 (FINAL) 时, 循环结束。

图 4-45 给出了一个跳转、循环指令应用的例子。对应的 STL 指令如下:

```

LD    I0.4
JMP   10           //跳转
LD    I0.5
EU
FOR   VW100,1,20   //循环开始
LDW> = VW100,15
=     Q1.0
NEXT
//循环返回
LBL   10           //标号

```

### 4.4.3 子程序调用指令 (SBR)

子程序将程序分成容易管理的小块, 使程序结构简单清晰, 易于查错和维护。可以多次调用同一个子程序, 使用子程序可以减少扫描时间。

#### 1. 建立子程序

可用编程软件编辑菜单中的插入选项, 选择子程序, 以建立或插入一个新的子程序, 同时在指令树窗口可以看到新建的子程序图标, 默认的程序名是 SBR\_n, 编号 n 从 0 开始按递增顺序生成, 可以在图标上直接更改子程序的程序名。在指令树窗口双击子程序的图标就可对子程序进行编辑。

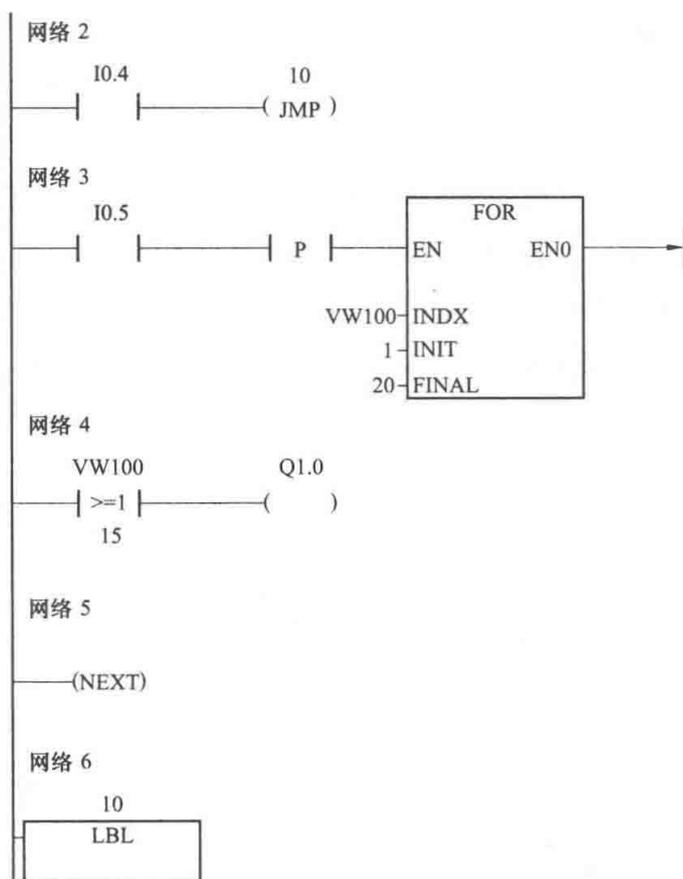


图 4-45 跳转、循环指令的应用

## 2. 子程序调用

将指令树中的子程序“拖”到程序编辑器中需要的位置。子程序可以多次被调用，也可以嵌套（最多 8 层），还可以递归调用（自己调用自己）。一般来讲，子程序在执行到末尾时自动返回，不必加返回指令。

```
CALL    SBR0           //子程序调用
CRET    //条件返回
RET     //无条件返回(自动)
```

**【例 4-18】** 用一个按钮控制 I0.4，用输出点 Q0.0 控制接触器 KM，接触器 KM 再控制电动机，要求按一下按钮，电动机启动，再按按钮，可以实现电动机停止。

**解** 采用子程序调用的方法设计满足控制要求的梯形图程序，如图 4-46 所示。对应的语句表指令如下：

主程序：

```
LD     I0.4
EU
CALL  SBR_0:SBR0
```

子程序：

```
LDN   Q0.0
=     Q0.0
```

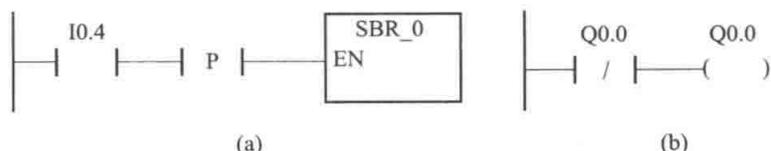


图 4-46 子程序调用指令的应用

(a) 主程序；(b) 子程序 SBR\_0

### \* 3. 带参数的子程序调用指令

在 [例 4-18] 中，子程序和调用它的主程序之间不需要进行任何参数的传送，所以在子程序中不需要定义有关变量，子程序调用指令中也不包含任何参数。

如果子程序可能有要传递的参数（变量和数据），这些参数可以在子程序与调用子程序的程序之间传送，这时参数就必须在子程序的局部变量表中定义，定义参数时必须指定参数的符号名称、变量类型和数据类型。一个子程序最多可以传送 16 个参数。在子程序的局部变量表中所定义的变量称为局部变量，局部变量只在它被创建的 POU（程序组织单元）中有效，而与此相对应的是全局变量（全局变量是在整个程序项目中都可使用的变量）。

局部变量表中的变量有 IN、OUT、IN/OUT 和 TEMP 等 4 种类型。

IN 类型：将指定位置的参数传入子程序。

OUT 类型：从子程序的结果值（数据）传入到指定参数位置。

IN/OUT 类型：将指定位置的参数传到子程序，从子程序来的结果值被返回到同样的地址。

TEMP 类型：局部存储器只用作子程序内部的暂时存储器，不能用来传递参数。

图 4-47 给出了带参数的子程序及调用的应用举例，该例中利用带参数的子程序来完成较复杂的数学计算功能，而在主程序中只需要输入要计算的角度参数即可得到相应的输出值。

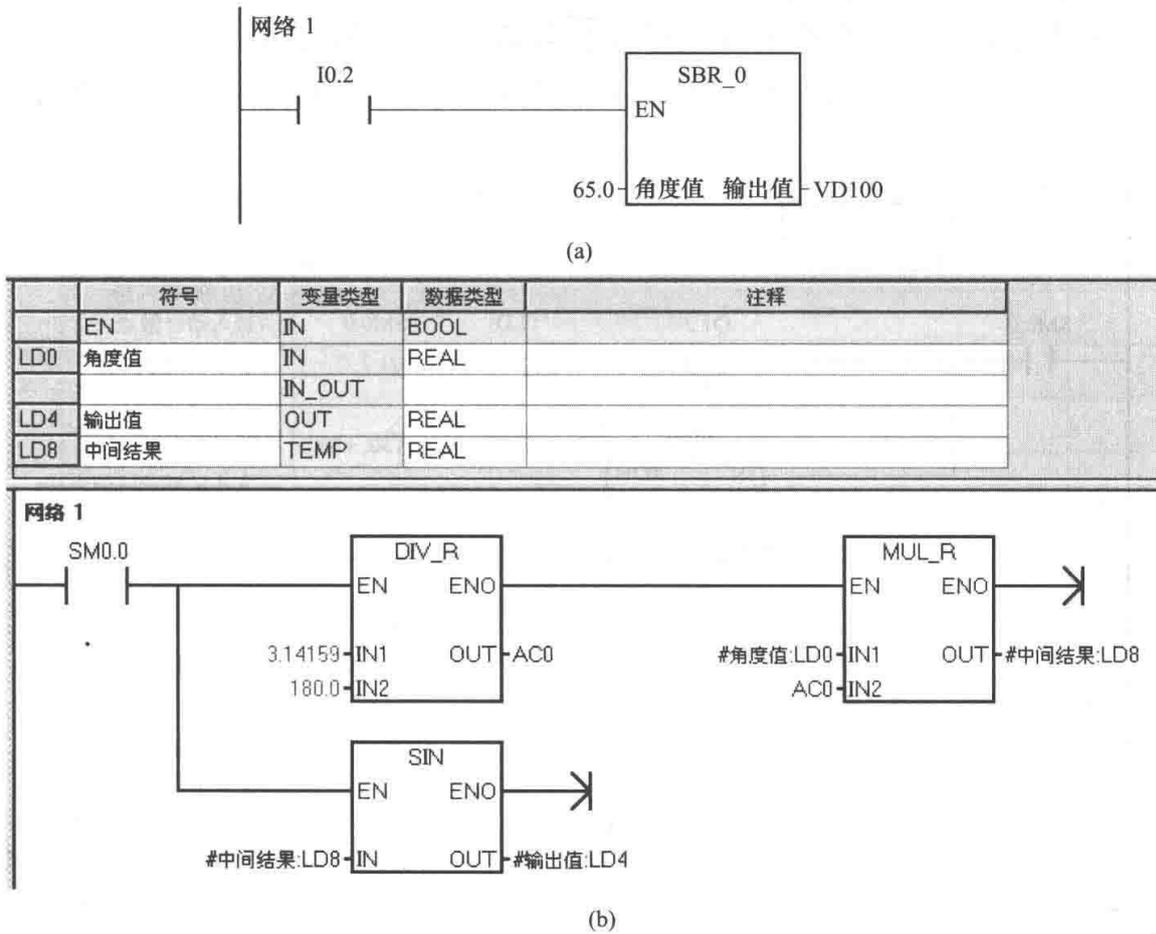


图 4-47 带参数的子程序及调用的应用举例  
(a) 主程序；(b) 子程序 SBR\_0

在局部变量表中定义的变量表和带参数的子程序如图 4-47 所示。局部变量的地址由编程软件自动分配。子程序中变量名称前面的“#”表示局部变量，也是软件自动添加的。图中定义的 3 个局部变量都是实数类型。EN 的输入为布尔型能流输入。子程序调用时，输入参数被复制到局部存储器。子程序完成时，从局部存储器复制输出参数到指定的输出参数地址。

#### 4.4.4 顺序控制指令

在 S7-200 PLC 中规定只能用状态寄存器来表示顺序控制步，每个步由一个状态寄存器位表示。步进控制指令包括步的开始、步的结束和步的转移指令。顺序控制指令的梯形图指令盒形式如图 4-48 所示。

顺序控制指令的 STL 指令如下：

LSCR Sn. x 步开始，其操作数 Sn. x 为状态寄存器 Sn 中的一个位，称为该 SCR 步标志位，当其为“1”时，允许该 SCR 步工作。

SCRT Sm. y 步转移，其操作数 Sm. y 为状态寄存器 Sm 中的一个位，当其为

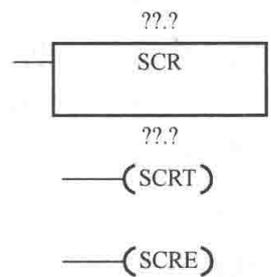


图 4-48 顺序控制指令的梯形图指令盒形式

“1”时,程序转移至由 Sm. y 表示的 SCR 段,同时自动停止当前 SCR 步的工作。

SCRE 步结束,顺序控制步无条件结束。

每个顺序控制步必须有 LSCR 和 SCRE 指令。

顺序控制指令可以将程序功能流程图转换成梯形图程序。

**【例 4-19】**用顺序控制指令实现顺序控制中的一个步的程序段,这一步实现的功能是使两个电机 M1 和 M2 启动运行 20s 后停止,切换到下一步。程序如图 4-49 所示。

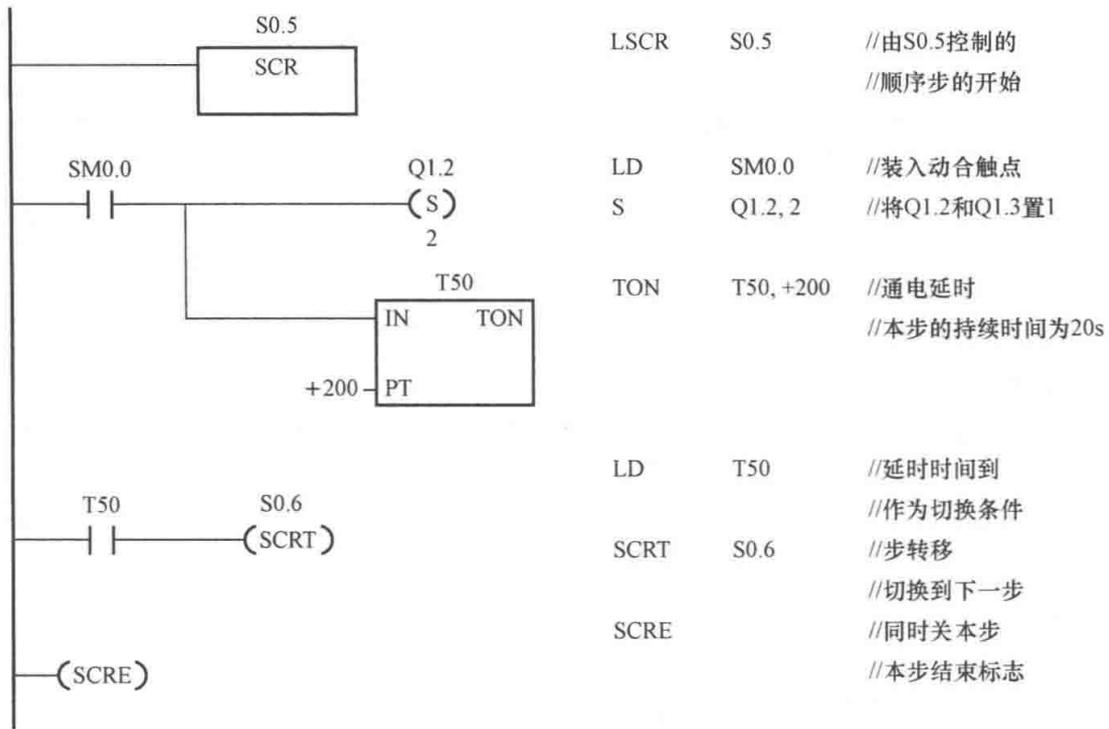


图 4-49 顺序控制指令的应用

步进条件为时间步进型。在 S0.5 所代表的步为活动步时,启动两个输出的同时启动定时器 T50,步进条件满足时(时间到)进入下一步(S0.6),关断上一步。

#### 4.4.5 中断

中断服务程序也是应用程序中的可选组件。当特定的中断事件发生时,执行相应的中断服务程序。中断服务程序用 INT\_n 来表示,编号 n 从 0 开始按递增顺序生成,可以在图标上直接更改中断程序的程序名。

##### 1. 中断源

(1) 中断源及种类。中断源,即中断事件发出中断请求的来源。S7-200 可编程控制器具有最多可达 34 个中断源,每个中断源都分配一个编号用以识别,称为中断事件号。这些中断源大致分为 3 大类:通信中断、输入输出中断和时基中断。

(2) 中断优先级。中断优先级由高到低依次是:通信中断、输入输出中断、时间中断。每种中断中的不同中断事件又有不同的优先权。主机中的所有中断事件及优先级见表 4-2。

表 4-2

中断事件及优先级

优先级组	组内优先级	中断事件号	中断事件描述	组内的优先级	
通信中断 (最高级)	通信口 0	8	接收字符	0	
		9	发送完成	0	
		23	接收信息完成	0	
	通信口 1	24	接收信息完成	1	
		25	接收字符	1	
		26	发送完成	1	
I/O 中断 (次高级)	脉冲串输出	19	PT0 完成中断	0	
		20	PT1 完成中断	1	
	外部输入	0	I0.0 上升沿中断	2	
		2	I0.1 上升沿中断	3	
		4	I0.2 上升沿中断	4	
		6	I0.3 上升沿中断	5	
		1	I0.0 下降沿中断	6	
		3	I0.1 下降沿中断	7	
		5	I0.2 下降沿中断	8	
		7	I0.3 下降沿中断	9	
	高速计数器	12	HSC0 CV=PV (当前值等于预设值)	10	
		27	HSC0 输入方向改变	11	
		28	HSC0 外部复位	12	
		13	HSC1 CV=PV (当前值等于预设值)	13	
		14	HSC1 输入方向改变	14	
		15	HSC1 外部复位	15	
		16	HSC2 CV=PV (当前值等于预设值)	16	
		17	HSC2 输入方向改变	17	
		18	HSC2 外部复位	18	
		32	HSC3 CV=PV (当前值等于预设值)	19	
		29	HSC4 CV=PV (当前值等于预设值)	20	
		30	HSC4 输入方向改变	21	
	时间中断 (最低级)	定时	10	定时中断 0 SMB34	0
			11	定时中断 1 SMB35	1
		定时器	21	定时器 T32 (当前值等于预设值)	2
	22		定时器 T96 (当前值等于预设值)	3	

## 2. 中断指令与中断设置

S7-200 中断指令见表 4-3, 要使用中断, 首先必须启动中断, 并且将中断事件和中断服务子程序相关联。S7-200 PLC 程序设计时, 可以将多个中断和一个中断服务子程序相关联。

表 4-3

S7-200 中断指令

指令	名称	功能
ENI	启动中断	全局性启用所有附加中断事件进程
DISI	禁用中断	全局性禁用所有中断事件进程
RETI	中断返回	可根据先前逻辑事件用于从中断返回
ATCH	附加中断	将中断事件 (EVNT) 与中断例行程序号码 (INT) 相联系, 并启用中断事件
DTCH	分离中断	取消中断事件 (EVNT) 与所有中断例行程序之间的联系, 并禁用中断事件

## 3. 中断程序构成

中断程序必须由 3 部分构成: 中断程序标号、中断程序指令和无条件返回指令。在 STEP7-Micro/WIN 中没有无条件返回指令, 应用程序在编译过程中会在各中断程序的末尾自动加入无条件返回指令。中断控制指令的应用如图 4-50 所示。

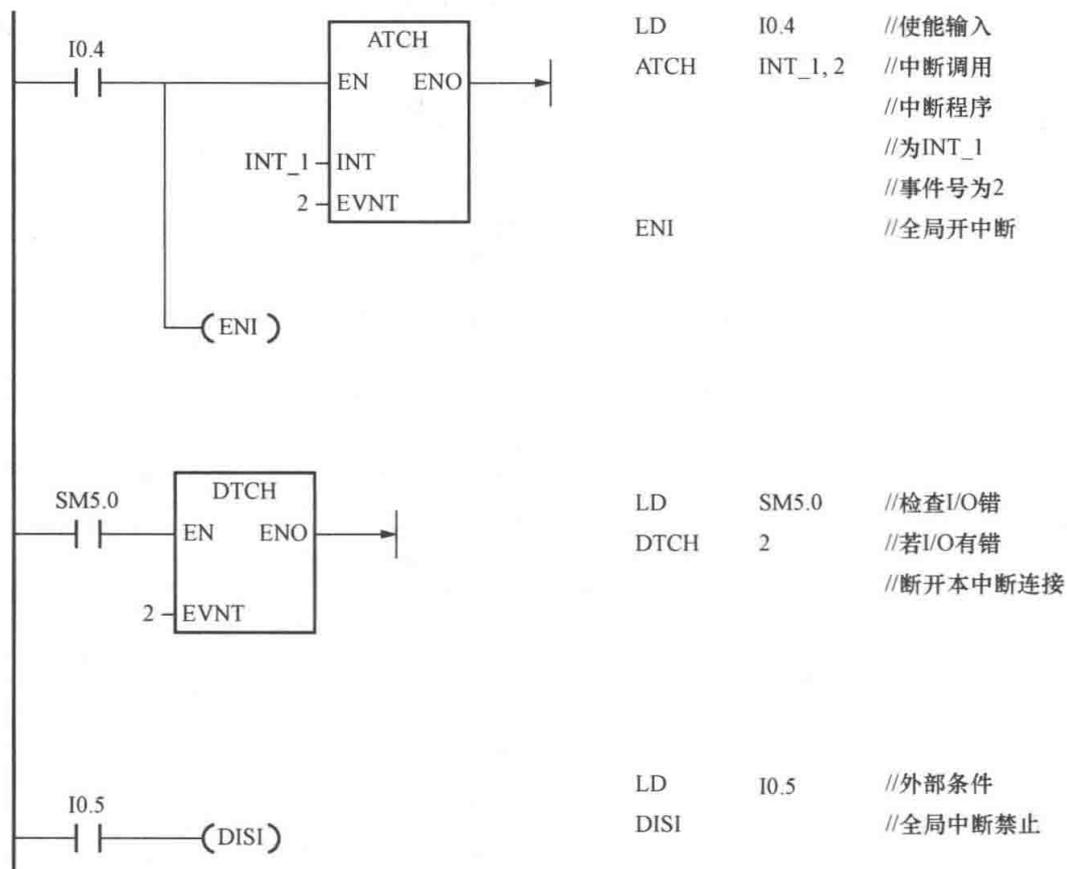


图 4-50 中断控制指令的应用

## 4. 中断调用

中断调用即调用中断程序, 使系统对特殊的内部或外部事件作出响应。系统响应中断时自动保存逻辑堆栈、累加器和某些特殊标志存储器位, 即保护现场。中断处理完成时, 又自动恢复这些单元原来的状态, 即恢复现场。但要注意, 用户不能在程序中通过子程序调用指

令调用中断子程序。

**【例4-20】**控制要求：程序实现的功能是调用 I0.1 输入点的上升沿中断，若发现 I/O 错误，则禁止本中断，用外部条件可以禁止全局中断。

程序实现：本程序如图 4-50 所示。

## \* 4.5 特殊指令

PLC 的有些特殊功能是通过特殊应用指令来实现的，这样可以使某些复杂控制任务的程序设计过程变得简单和容易，S7-200 PLC 的特殊应用指令有实时时钟的设定和读取指令、高速计数指令、脉冲输出指令、通信指令和 PID 控制指令，

本章主要介绍实时时钟的设定和读取指令、高速计数指令、脉冲输出指令的应用，通信指令和 PID 控制指令在后面的章节中介绍。

### 4.5.1 时钟指令

#### 1. 读实时时钟

TODR，读实时时钟指令。当使能输入有效时，系统读当前时间和日期，并把它装入一个 8 字节的缓冲区。

#### 2. 写实时时钟

TODW，写实时时钟指令，用来设定实时时钟。当使能输入有效时，系统将包含当前时间和日期，一个 8 字节的缓冲区将装入时钟。

### 4.5.2 高速计数器指令

前面介绍的一般计数器指令是按照扫描工作方式工作的，计数的频率受扫描周期的限制，而在实际应用中，某些输入信号的频率远比 PLC 的扫描频率高，这时就不能采用一般的计数器指令了，必须采用高速计数器完成输入脉冲的计数任务。

#### 1. 高速计数器

(1) 数量及编号。高速计数器在程序中使用时的地址编号用 HCn 来表示（在非程序中有时用 HSCn），HC 表示编程元件名称为高速计数器，n 为编号。

HCn 除了表示高速计数器的编号之外，还代表两方面的含义：高速计数器位和高速计数器当前值。编程时，从所用的指令可以看出是位还是当前值。

不同型号的 PLC 主机，高速计数器的数量对应见表 4-4。

表 4-4 S7-200 高速计数器的数量及编号

主机型号	CPU221	CPU222	CPU224	CPU226
可用 HSC 数量	4	4	6	6
HSC 编号范围	HC0, HC3, HC4, HC5	HC0, HC3, HC4, HC5	HC0~HC5	HC0~HC5

(2) 中断事件类型。高速计数器的计数和动作可采用中断方式进行控制，与 CPU 的扫描周期关系不大，各种型号的 PLC 可用的高速计数器的中断事件大致分为 3 类：当前值等于预设值中断、输入方向改变中断和外部复位中断。所有高速计数器都支持当前值等于预设值中断。

每个高速计数器的 3 种中断的优先级由高到低，不同高速计数器之间的优先级又按编号顺序由高到低。具体对应关系见表 4-5。

表 4-5

S7-200 高速计数器中断事件类型

高速计数器	当前值等于预设值中断		计数方向改变中断		外部信号复位中断	
	事件号	优先级	事件号	优先级	事件号	优先级
HSC0	12	10	27	11	28	12
HSC1	13	13	14	14	15	15
HSC2	16	16	17	17	18	18
HSC3	32	19	无	无	无	无
HSC4	29	20	30	21	无	无
HSC5	33	23	无	无	无	无

(3) 高速计数器的输入端连接。输入端连接, 选用某个高速计数器在某种工作模式下工作, 高速计数器的输入端不是可以任意选择的, 必须按系统指定的输入点, 见表 4-6。

表 4-6

高速计数器的输入端连接

高速计数器编号	I0. y y 的取值							I1. y y 的取值						
	0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC0	✓	✓	✓											
HSC1							✓	✓	✓	✓				
HSC2											✓	✓	✓	✓
HSC3		✓												
HSC4				✓	✓	✓								
HSC5					✓									

(4) 高速计数器的状态字节。S7-200 PLC 为每个高速计数器提供状态字节, 以监控高速计数器的工作状态, 状态字的各位表示当前计数方向以及当前值是否大于或等于预设值。

只有在执行高速计数器指令及对应的中断程序时, 状态字节的各状态位才有效。监控高速计数器状态的目的在于启用对正在执行的操作有重大影响的事件的中断程序。状态位的定义见表 4-7。

表 4-7

高速计数器的状态字节的格式

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	功能描述
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	不用
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	不用
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	不用
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	不用
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	不用
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	计数方向: 0—减计数, 1—加计数
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	当前值等于设定值: 1—等于, 0—不等于
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	当前值大于设定值: 1—大于, 0—少于等于

(5) 高速计数器的控制字节。每个高速计数器指定了一个特殊功能寄存器 (SMB37、SMB47、SMB57、SMB137、SMB147、SMB157) 作为控制字, 通过对该字节中相应位的设

定, 来确定高速计数器的工作方式, 在控制字节中可设置启动输入信号和复位输入信号的有效电平, 正交计数器的计数倍率, 计数方向采用内部控制时的有效电平, 是否允许改变计数方向, 是否允许更新设定值, 是否允许更新当期值, 以及是否允许执行高速计数指令等, 各位的具体含义见表 4-8。

表 4-8 控制位含义

控制位	功能描述	适用的计数器 HCn
SM××7.0	复位输入控制: 0—高电平有效; 1—低电平有效	0, 1, 2, 4
SM××7.1	启动输入控制: 0—高电平有效; 1—低电平有效	1, 2
SM××7.2	倍率选择控制: 0—4 倍频; 1—1 倍频	0, 1, 2, 4
SM××7.3	计数方向控制: 0—减计数; 1—加计数	0, 1, 2, 3, 4, 5
SM××7.4	改变计数方向控制: 0—不允许改变; 1—允许改变	0, 1, 2, 3, 4, 5
SM××7.5	改变设定值控制: 0—不允许改变; 1—允许改变	0, 1, 2, 3, 4, 5
SM××7.6	改变当前值控制: 0—不允许改变; 1—允许改变	0, 1, 2, 3, 4, 5
SM××7.7	高速计数控制: 0—禁止计数; 1—允许计数	0, 1, 2, 3, 4, 5

(6) 高速计数器的当前值寄存器和设定值寄存器。每个高速计数器指定了一个 32 位的当前值寄存器 (SMD) 和一个 32 位的设定值寄存器 (SMD), 当前值和设定值均为有符合整数。S7-200 PLC 高速计数器当前值和设定值对应的特殊寄存器见表 4-9。

表 4-9 高速计数器当前值和设定值对应的特殊寄存器

计数器	HC0	HC1	HC2	HC3	HC4	HC5
当前值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
设定值	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

## 2. 高速计数器的工作模式

S7-200 PLC 中每个高速计数器都有多种工作模式, 但每个高速计数器的工作模式不完全相同, 最多的有 12 种工作模式, 可使用高速计数器定义指令来选定工作模式。

(1) HC0 是一个通用型增减计数器, 可通过程序来设定 8 种不同的工作模式, HC0 的工作模式见表 4-10。

表 4-10 HC0 的工作模式

模式	描述	控制位	I0.0	I0.1	I0.2
0	具有内部方向控制的单相增减计数器	SM37.3=0, 减	脉冲	—	—
1		SM37.3=1, 增			复位
3	具有外部方向控制的单相增减计数器	I0.1=0, 减	脉冲	方向	—
4		I0.1=1, 增			复位
6	具有增减计数脉冲输入端的双相计数器	外部输入控制	脉冲增	脉冲减	—
7					复位
9	A/B 相正交计数器	A 超前 B, 顺时针	外部输入控制	A 相脉冲	B 相脉冲
10		B 超前 A, 逆时针			

(2) HC1 共有 12 种工作模式, 见表 4-11。

表 4-11 HC1 的工作模式

模式	描 述	控制位	I0.6	I0.7	I1.0	I1.1
0	具有内部方向控制的单相增减计数器	SM47.3=0, 减; SM47.3=1, 增	脉冲	—	—	—
1					复位	启动
2						
3	具有外部方向控制的单相增减计数器	I0.7=0, 减; I0.7=1, 增	脉冲	方向	—	—
4					复位	启动
5						
6	具有增减计数脉冲输入端的双相计数器	外部输入控制	脉冲增	脉冲减	—	—
7					复位	启动
8						
9	A/B 相正交计数器。	外部输入控制	A 相脉冲	B 相脉冲	—	—
10	A 相超前 B 相 90°, 顺时针旋转;				复位	启动
11	B 相超前 A 相 90°, 逆时针旋转					

(3) HC2 也有 12 种工作模式, 见表 4-12。

表 4-12 HC2 的工作模式

模式	描 述	控制位	I1.2	I1.3	I1.4	I1.5
0	具有内部方向控制的单相增减计数器	SM57.3=0, 减; SM57.3=1, 增	脉冲	—	—	—
1					复位	启动
2						
3	具有外部方向控制的单相增减计数器	I1.3=0, 减; I1.3=1, 增	脉冲	方向	—	—
4					复位	启动
5						
6	具有增减计数脉冲输入端的双相计数器	外部输入控制	脉冲增	脉冲减	—	—
7					复位	启动
8						
9	A/B 相正交计数器。	外部输入控制	A 相脉冲	B 相脉冲	—	—
10	A 相超前 B 相 90°, 顺时针旋转;				复位	启动
11	B 相超前 A 相 90°, 逆时针旋转					

(4) HC3 只有一种工作模式, 见表 4-13。

表 4-13 HC3 的工作模式

模式	描 述	控 制 位	I0.1
0	具有内部方向控制的单相增减计数器	SM137.3=0, 减; SM137.3=1, 增	脉冲

(5) HC4 共有 8 种工作模式, 见表 4-14。

表 4-14 HC4 的工作模式

模式	描述		控制位	I0.3	I0.4	I0.5
0	具有内部方向控制的单相增减计数器		SM147.3=0, 减	脉冲	—	—
1			SM147.3=1, 增			复位
3	具有外部方向控制的单相增减计数器		I0.4=0, 减	脉冲	方向	—
4			I0.4=1, 增			复位
6	具有增减计数脉冲输入端的双相计数器		外部输入控制	脉冲增	脉冲减	—
7						复位
9	A/B相正交计数器	A超前B, 顺时针	外部输入控制	A相脉冲	B相脉冲	—
10		B超前A, 逆时针				复位

(6) HC5 只有一种操作模式, 见表 4-15。

表 4-15 HC5 的工作模式

模式	描述	控制位	I0.4
0	具有内部方向控制的单相增减计数器	SM157.3=0, 减; SM157.3=1, 增	脉冲

(7) 高速计数器的工作模式说明。从以上各高速计数器的工作模式可以看出: 6 个高速计数器所具有的功能不完全相同, 最多的有 12 种工作模式, 可分为 4 种类型。下面以 HC1 的工作模式为例, 说明高速计数器的工作模式。

1) 具有内部方向控制的单相增减计数器。在模式 0、模式 1 和模式 2 中, HC1 可作为具有内部方向控制的单相增减计数器, 它根据 PLC 内部的特殊标志存储器位 SM47.3 的状态来确定计数方向 (1 表示增, 0 表示减), I0.6 作为计数脉冲的输入端。在模式 1 和模式 2 中, I1.0 作为复位输入端。在模式 2 中, I1.1 作为启动输入端。

2) 具有外部方向控制的单相增减计数器。在模式 3、模式 4 和模式 5 中, HC1 可作为具有外部方向控制的单相增减计数器, 它根据 PLC 外部输入点 I0.7 的状态 (1 表示增计数, 0 表示减计数) 来确定计数方向, 外部输入点 I0.6 作为计数脉冲的输入端。在模式 4 和模式 5 中, I1.0 作为复位输入端。在模式 5 中, I1.1 作为启动输入端。

3) 具有增减计数脉冲输入端的双向计数器。在模式 6、模式 7 和模式 8 中, HC1 可作为具有增减计数脉冲输入的双向计数器, PLC 外部输入点 I0.6 和 I0.7 分别作为外部增、减计数脉冲的输入端。在模式 7 和模式 8 中, I1.0 作为复位输入端。在模式 8 中, I1.1 作为启动输入端。

如果增计数脉冲的上升沿与减计数脉冲的上升沿出现的时间间隔在 0.3ms 之内, 则 CPU 会认为这两个计数脉冲是同时到来的, 此时, 计数器的当前值保持不变, 也不会发出计数方向改变的信号。当增计数脉冲的上升沿与减计数脉冲的上升沿出现的时间间隔大于 0.3ms 时, 高速计数器就可以捕获这一独立的事件。

4) A/B 相正交计数器。在模式 9、模式 10 和模式 11 中, HC1 可作为 A/B 相正交计数器 (所谓正交, 是指 A、B 两相输入脉冲相差 90°)。外部输入点 I0.6 为 A 相脉冲输入, I0.7 为 B 相脉冲输入。在模式 10 和模式 11 中, I1.0 作为复位输入信号。在模式 11 中, I1.1 作为启动输入信号。

当 A 相脉冲超前 B 相脉冲 90° 时, 计数方向为递增计数; 当 B 相脉冲超前 A 相脉冲 90°

时,计数方向为递减计数。

正交计数器有两种工作状态,其中一种是每输入 1 个计数脉冲,当前值计 1 个数,即计数倍率为 1。

而在许多位移测量系统中,常常采用光电编码器,将光电编码器的 A、B 两相输出信号作为高速计数器的输入信号,为提高测量精确度,可以对光电编码器的 A、B 相脉冲信号作 4 倍频计数。当 A 相脉冲信号超前 B 相脉冲信号  $90^\circ$  时,为正转(顺时针转动);当 B 相脉冲信号超前 A 相脉冲信号  $90^\circ$  时,为反转(逆时针转动)。为满足这种需要,正交计数器的另一种工作状态是每输入 1 个计数脉冲,当前值计 4 个数,即计数倍率为 4。

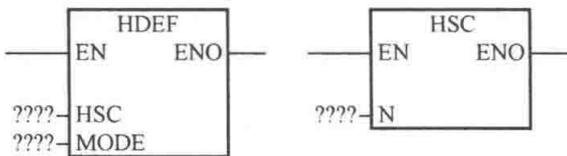


图 4-51 高速计数指令的梯形图指令盒形式

### 3. 高速计数指令

高速计数器指令有两条: HDEF 和 HSC。梯形图中的指令盒形式如图 4-51 所示。

(1) HDEF 指令。HDEF, 定义高速计数器指令。使能输入有效时,为指定的高速计数器分配一种工作模式,即用来建立高速计数器与工作模式之间的联系。梯形图指令盒中有两个数据输入端: HSC, 高速计数器编号,为 0~5 的常数,字节型; MODE, 工作模式,为 0~11 的常数,字节型。

(2) HSC 指令。HSC, 高速计数器指令。使能输入有效时,根据高速计数器特殊存储器位的状态,并按照 HDEF 指令指定的工作模式,设置高速计数器并控制其工作。梯形图指令盒数据输入端 N: 高速计数器编号,为 0~5 的常数,字节型。

### 4. 高速计数器的使用方法

使用高速计数器时,要按以下步骤进行:

- (1) 选择计数器及工作模式。
- (2) 设置控制字节。
- (3) 执行 HDEF 指令。
- (4) 设定当前值和预设值。
- (5) 设置中断事件并全局开中断。
- (6) 执行 HSC 指令,激活高速计数器。

**【例 4-21】**要对一高速事件精确控制,通过对脉冲信号进行增计数,计数当前值达到 24 产生中断,重新从 0 计数,对中断次数进行累计。计数方向用一个外部信号控制,并能实现外部复位。所用的主机型号为 CPU221。

设计步骤: ①选择高速计数器 HSC0,并确定工作方式 4。②设置控制字,令 SMB37=16#F8。③执行 HDEF 指令,输入端 HSC 为 0,MODE 为 4。④装入当前值,令 SMD38=0。⑤装入设定值,令 SMD42=24。⑥执行中断连接 ATCH 指令,输入端 INT 为 INT0, EVNT 为 12。高速脉冲输入端连 PLC 的输入端的 I0.0,增方向控制连接 PLC 的输入端的 I0.1,复位控制连接 PLC 的输入端的 I0.2。

主程序、中断程序和初始化子程序分别如图 4-52~图 4-54 所示。

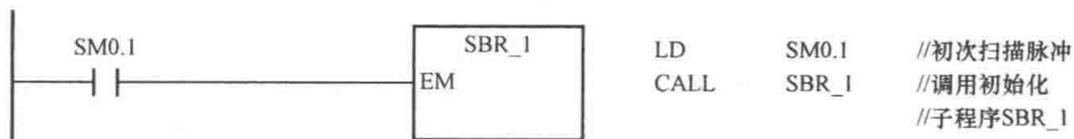


图 4-52 主程序

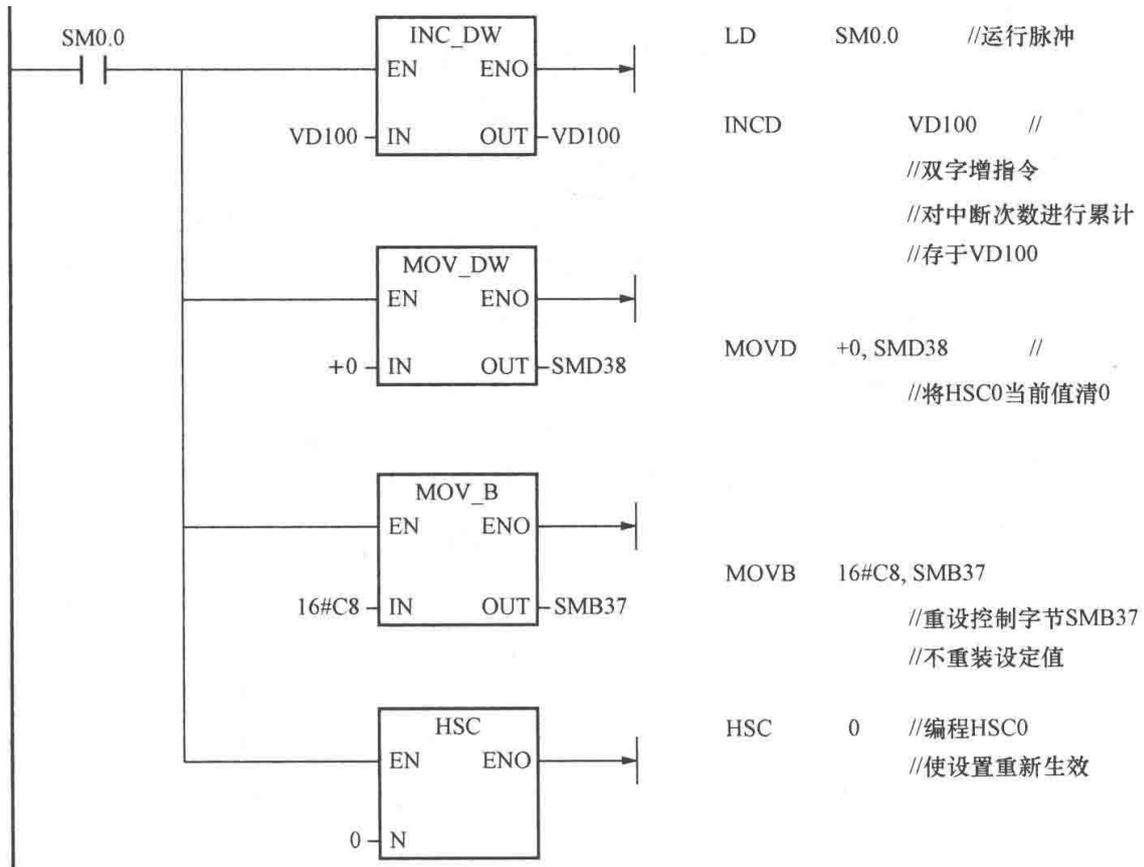


图 4-53 中断程序

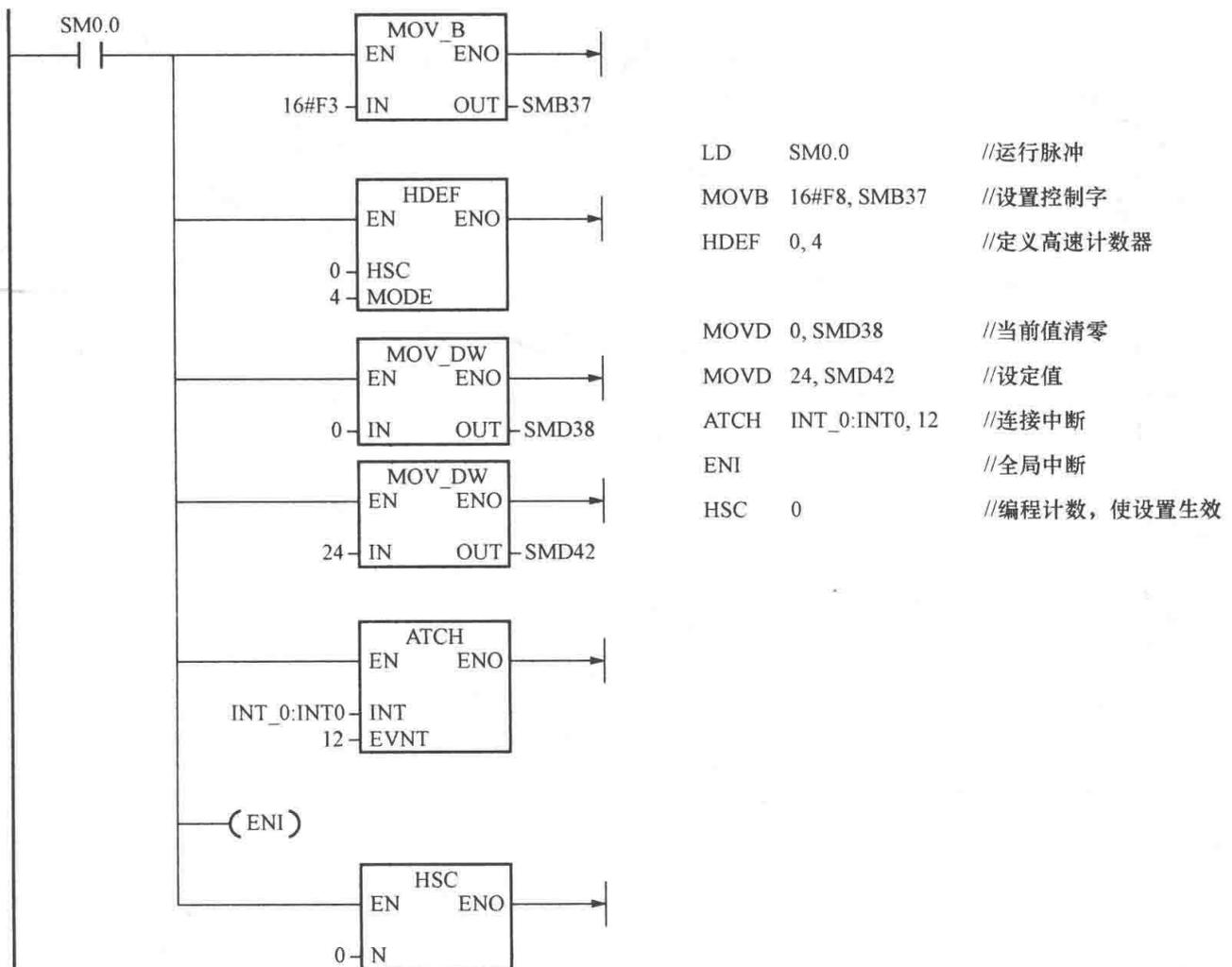


图 4-54 初始化子程序

### 4.5.3 高速脉冲输出指令

在 S7-200 系列 PLC 中。只有晶体管输出类型的 CPU 的两个输出点 Q0.0 和 Q0.1 可作为 PTO/PWM 输出使用。PTO 可以输出一串脉冲 (占空比 50%)，用户可以控制脉冲的周期和个数。PWM 可以输出连续的、占空比可调的脉冲串，用户可以控制脉冲的周期和脉宽。

PTO/PWM 发生器与过程映像寄存器共用 Q0.0 和 Q0.1。当在 Q0.0 或 Q0.1 上激活 PTO 或 PWM 功能时，PTO/PWM 发生器对输出拥有控制权，同时普通输出点功能被禁止。输出波形不受过程映像区状态、输出点强制值或者立即输出指令执行的影响。当不使用 PTO/PWM 发生器功能时，对输出点的控制权交回到过程映像寄存器。

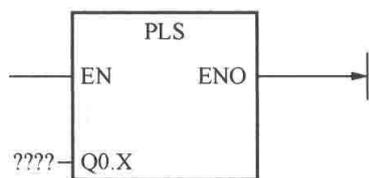


图 4-55 高速脉冲输出指令的梯形图指令盒形式

#### 1. 高速脉冲输出指令形式

高速脉冲输出指令的梯形图指令盒形式如图 4-55 所示，指令格式为：PLS Q0.X

当 EN 端有效时，指令检测各相关特殊功能寄存器的状态，根据定义的控制字节执行高速脉冲输出操作。

脉冲输出指令 (PLS) 用于在高速输出 (Q0.0 和 Q0.1) 上控制脉冲串输出 (PTO) 和脉宽调制 (PWM) 功能。

S7-200 PLC 有两个 PTO/PWM 发生器，它们可以产生一个高速脉冲串或者一个脉宽调制波形。

#### 2. 与高速脉冲输出指令有关的特殊功能寄存器

在 S7-200 系列 PLC 中，每路 PTO/PWM 发生器都对应一定数量的特殊功能寄存器，这些寄存器包括控制字节寄存器、状态字节寄存器和参数数值寄存器，用以控制高速脉冲的输出形式及反映输出状态和参数，各寄存器分配见表 4-16。

表 4-16

相关寄存器表

Q0.0 的寄存器	Q0.1 的寄存器	名称及功能描述
SMB66	SMB76	状态字节，在 PTO 方式下，跟踪脉冲串的输出状态
SMB67	SMB77	控制字节，控制 PTO/PWM 脉冲输出的基本功能
SMW68	SMW78	周期值，字型，PTO/PWM 的周期值，范围：2~65535
SMW70	SMW80	脉宽值，字型，PWM 的脉宽值，范围：0~65535
SMD72	SMD82	脉冲数，双字型，PTO 的脉冲数，范围：1~4294967295
SMB166	SMB176	段数，多段管线 PTO 进行中的段数
SMW168	SMW178	偏移地址，多段管线 PTO 包络表的起始字节的偏移地址

(1) 状态字节。每个高速脉冲输出都有一个状态字节，程序运行时根据运行状况自动使某些位置位，可以通过程序来读相关位的状态，用以作为判断条件实现相应的操作。状态字节中各状态位的功能见表 4-17。

表 4-17

状态字节表

状态位	SM×6.0~SM×6.3	SM×6.4	SM×6.5	SM×6.6	SM×6.7
功能描述	不用	PTO 包络因计算错误终止： 0—无错； 1—终止	PTO 包络因用户命令终止： 0—无错； 1—终止	PTO 管线溢出： 0—无溢； 1—溢出	PTO 空闲： 0—执行中； 1—空闲

(2) 控制字节。每个高速脉冲输出都对应一个控制字节，通过对控制字节中指定位的编程，可以根据操作要求设置字节中各控制位，如脉冲输出允许、PTO/PWM 模式选择、单段/多段选择、更新方式、时间基准、允许更新等。控制字节中各控制位的功能见表 4-18。

### 3. 高速脉冲串输出 PTO

#### (1) 周期和脉冲数。

1) 周期：单位可以是微秒 ( $\mu\text{s}$ ) 或毫秒 (ms)；为 16 位无符号数据，周期变化范围是  $50\sim 65535\mu\text{s}$  或  $2\sim 65535\text{ms}$ ，通常应设定周期值为偶数，若设置为奇数，则会引起输出波形占空比的轻微失真。如果编程时设定周期单位小于 2，系统默认按 2 进行设置。

2) 脉冲数：用双字长无符号数表示，脉冲数取值范围在  $1\sim 4294967295$  之间。如果编程时指定脉冲数为 0，则系统默认脉冲数为 1 个。

表 4-18 控制字节中各控制位的功能

控制位	控制位	功能描述
SM67.0	SM77.0	PTO/PWM 更新周期值允许：0—不更新；1—允许更新
SM67.1	SM77.1	PWM 更新脉冲宽度值允许：0—不更新；1—允许更新
SM67.2	SM77.2	PTO 更新输出脉冲数允许：0—不更新；1—允许更新
SM67.3	SM77.3	PTO/PWM 时间基准选择：0— $\mu\text{s}$ 单位时基；1— $\mu\text{s}$ 单位时基
SM67.4	SM77.4	PWM 更新方式：0—异步更新；1—同步更新
SM67.5	SM77.5	PTO 单/多段方式：0—单段管线；1—多段管线
SM67.6	SM77.6	PTO/PWM 模式选择：0—选用 PTO 模式；1—选用 PWM 模式
SM67.7	SM77.7	PTO/PWM 脉冲输出允许：0—禁止；1—允许

(2) PTO 的种类。PTO 方式中，如果要输出多个脉冲串，允许脉冲串进行排队，形成管线，当前输出的脉冲串完成之后，立即输出新脉冲串，这保证了脉冲串顺序输出的连续性。

1) 单段管线：在单段管线模式，需要为下一个脉冲串更新特殊寄存器。一旦启动了起始 PTO 段，就必须按照第二个波形的要求改变特殊寄存器，并再次执行 PLS 指令。第二个脉冲串的属性在管线中一直保持到第一个脉冲串发送完成。在管线中一次只能存储一段脉冲串的属性。当第一个脉冲串发送完成时，接着输出第二个波形，此时管线可以用于下一个新的脉冲串。

单段管线的各段脉冲串可以采用不同的时间基准，但有可能造成脉冲串之间的不平稳过渡。输出多个高速脉冲串时，编程复杂，所以一般选用多段管线模式。

2) 多段管线：在多段管线模式，CPU 自动从 V 存储器区的包络表中读出每个脉冲串的特性。在该模式下，仅使用特殊存储器区的控制字节和状态字节。选择多段操作，必须装入包络表在 V 存储器中的起始地址偏移量 (SMW168 或 SMW178)。时间基准可以选择微秒或者毫秒，但是，在包络表中的所有周期值必须使用同一个时间基准，而且在包络正在运行时不能改变。执行 PLS 指令来启动多段操作。

包络表由包络段数和各段构成。每段记录的长度为 8 个字节，由 16 位周期值、16 位周期增量值和 32 位脉冲计数值组成。以包络 3 段的包络表为例，包络表的结构见表 4-19。

表 4-19 包络表格式

字节偏移地址	名称	描述
VBn	段标号	段数, 为 1~255, 数 0 将产生非致命性错误, 不产生 PTO 输出
VWn+1	段 1	初始周期, 取值范围为 2~65535
VWn+3		每个脉冲的周期增量, 符号整数, 取值范围为 -32768~+32767
VDn+5		输出脉冲数, 为 1~4294967295 之间的列符号整数
VWn+9	段 2	初始周期, 取值范围为 2~65535
VWn+11		每个脉冲的周期增量, 符号整数, 取值范围为 -32768~+32767
VDn+13		输出脉冲数, 为 1~4294967295 之间的列符号整数
VWn+17	段 3	初始周期, 取值范围为 2~65535
VWn+19		每个脉冲的周期增量, 符号整数, 取值范围为 -32768~+32767
VDn+21		输出脉冲数, 为 1~4294967295 之间的列符号整数

可以通过编程的方式使脉冲的周期自动增减。在周期增量处输入一个正值将增加周期; 输入一个负值将减少周期; 输入 0 将不改变周期。

当 PTO 包络执行时, 当前启动的段的编号保存在 SMB166 (或 SMB176)。

(3) 中断事件类型。高速脉冲串输出可以采用中断方式进行控制, 各种型号的 PLC 可用的高速脉冲串输出的中断事件有两个, 见表 4-20。

表 4-20 中 断 操 作

中断事件号	事件描述	优先级 (在 I/O 中断中的次序)
19	PTO 0 高速脉冲串输出完成中断	0
20	PTO 1 高速脉冲串输出完成中断	1

(4) PTO 的使用。使用高速脉冲串输出时, 要按以下步骤进行: 确定脉冲发生器及工作模式, 设置控制字节, 写入周期值、周期增量值和脉冲数, 装入包络的首地址, 设置中断事件并全局开中断, 执行 PLS 指令。

#### 4. 宽度可调脉冲输出 PWM

PWM 产生一个占空比变化周期固定的脉冲输出, 可以以微秒或者毫秒为单位指定其周期和脉冲宽度。

##### (1) 周期和脉冲宽度。

- 1) 周期:  $10\sim 65535\mu\text{s}$  或者  $2\sim 65535\text{ms}$ 。
- 2) 脉宽:  $0\sim 65535\mu\text{s}$  或者  $0\sim 65535\text{ms}$ 。

##### (2) 更新方式。

1) 同步更新: 如果不需要改变时间基准, 就可以进行同步更新。利用同步更新, 波形特性的变化发生在周期边沿, 提供平滑转换。

2) 异步更新: PWM 的典型操作是当周期时间保持常数时变化脉冲宽度。所以, 不需要改变时间基准。但是, 如果需要改变 PTO/PWM 发生器的时间基准, 就要使用异步更新。异步更新会造成 PTO/PWM 功能被瞬时禁止, 与 PWM 波形不同步。这会引入被控设备的振动。由于这个原因, 建议采用 PWM 同步更新。选择一个适合于所有周期时间的时

基准。

控制字节中的 PWM 更新方式位 (SM67.4 或 SM77.4) 用于指定更新方式。当 PLS 指令执行时变化生效。

如果改变了时间基准,会产生一个异步更新,而与 PWM 更新方式位的状态无关。

PTO/PWM 发生器的多段管线功能在许多应用中非常有用,尤其在步进电动机控制中。步进电动机在启动和停止时有一个加速及减速过程,且加速度越小则冲击越小,动作越平稳。所以,步进电动机工作时一般要经历这样一个变化过程:加速→恒速(高速)→减速→恒速(低速)→停止。步进电动机转速与脉冲频率成正比,所以输入步进电动机的脉冲频率也要经过一个类似变化过程。结合下面的例子来进一步分析和说明。

**【例 4-22】**某步进电动机转动过程中,要从 A 点加速到 B 点后恒速运行,又从 C 点开始减速到 D 点,完成这一过程时用指示灯显示。电动机的转动受脉冲控制, A 点和 D 点的脉冲频率为 2kHz, B 点和 C 点的频率为 10kHz,加速过程的脉冲数为 400 个,恒速转动的脉冲数为 4000 个,减速过程脉冲数为 200 个。

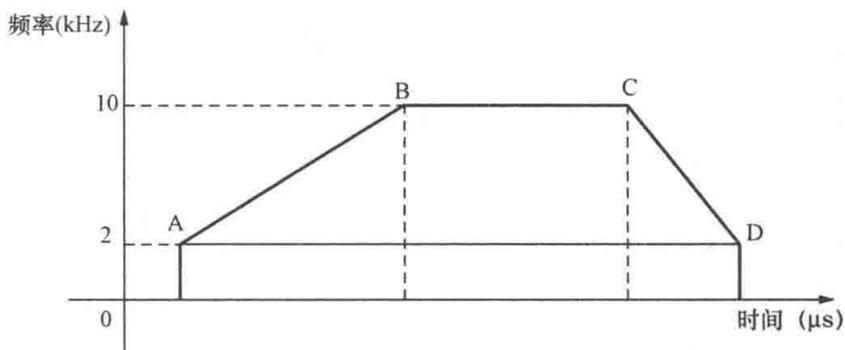


图 4-56 步进电动机工作过程

工作过程如图 4-56 所示。

(1) 分析。

1) 确定脉冲发生器及工作模式,选择 Q0.0 输出,选择 3 段管式 PTO 的输出形式。设置控制字节,选择的时基单位为  $\mu\text{s}$ ,向控制字节写入控制字。

2) 确认并写入周期值、周期增量值和脉冲数,由每段的初始频率可得初始周期,某一段每个脉冲周期增量值  $\Delta$  用下式确定:周期增量值  $\Delta = (\text{该段结束时的周期时间} - \text{该段初始的周期时间}) / \text{该段的脉冲数}$ 。则计算出 1 段的周期增量值  $\Delta = -1\mu\text{s}$ , 2 段的周期增量值  $\Delta = 0$ , 3 段的周期增量值  $\Delta = 2\mu\text{s}$ 。

3) 装入包络表首地址,假设包络表位于从 VB400 开始的 V 存储区中,这里选择的首地址为 VB400。

4) 中断调用,设置中断事件并全局开中断。如果想在 PTO 完成后,立即执行相关功能,则需设置中断,将脉冲串完成事件(中断事件号 19)连接一中断程序。

5) 执行 PLS 指令,使 S7-200 为 PTO 发生器编程,高速脉冲串由 Q0.0 输出。

(2) 程序实现。本控制系统主程序如图 4-57 所示。初始化子程序 SBR\_1 如图 4-58 所示。包络表子程序 SBR\_0 如图 4-59 所示。中断程序 INT\_0 如图 4-60 所示。

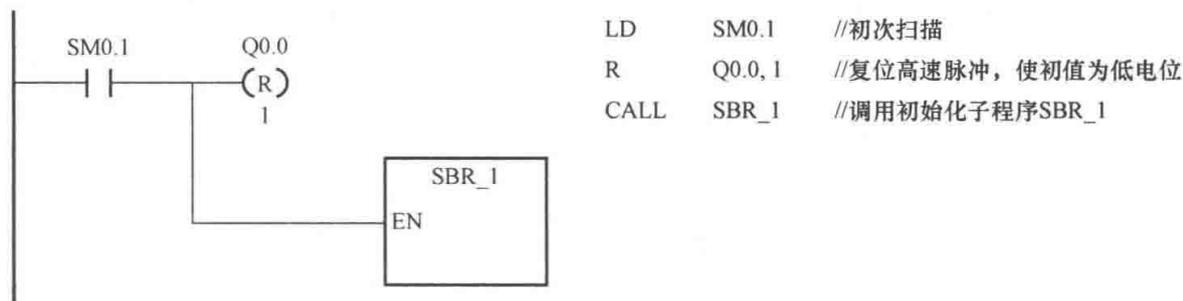


图 4-57 主程序

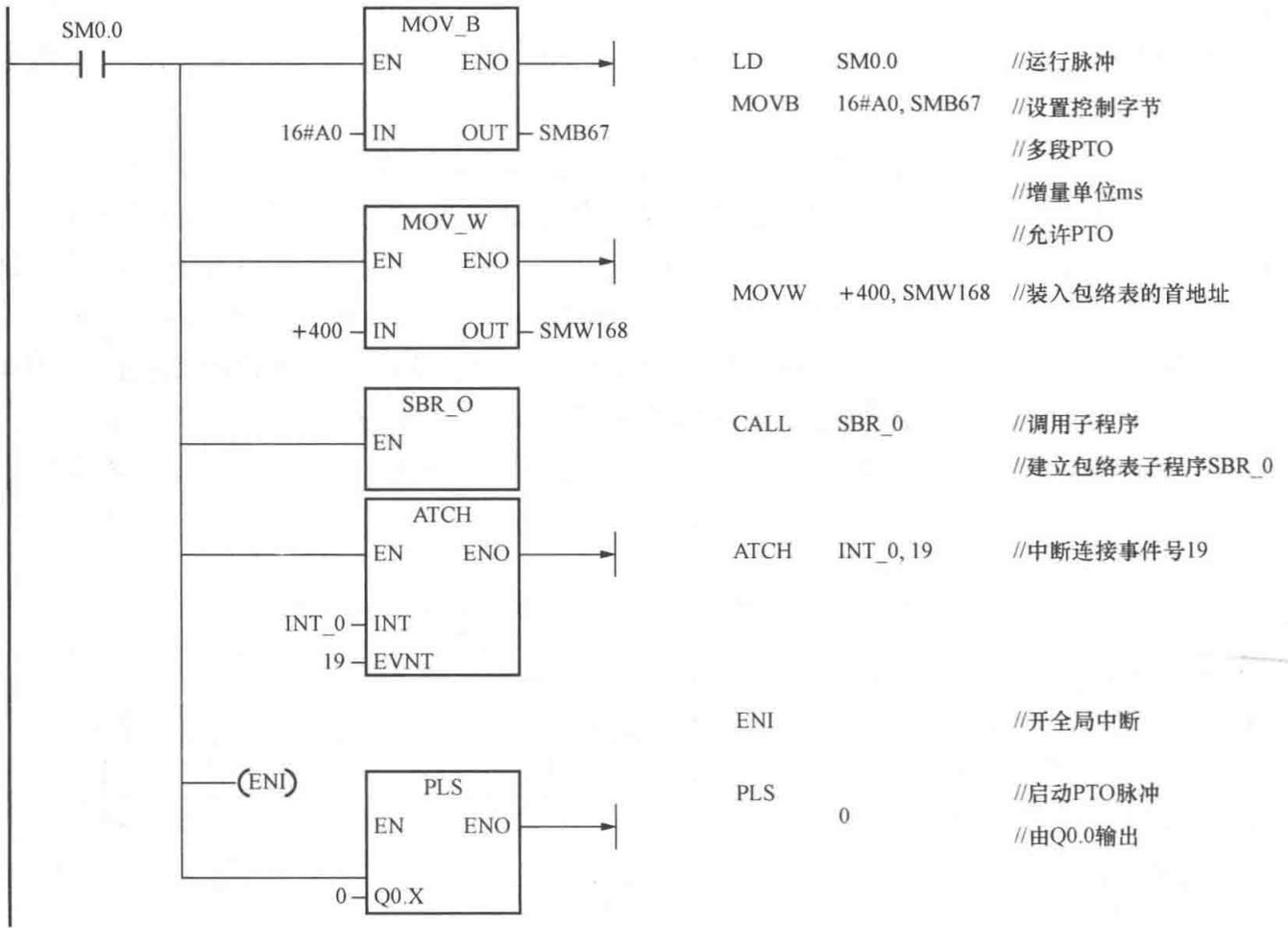


图 4-58 初始化子程序 SBR\_1

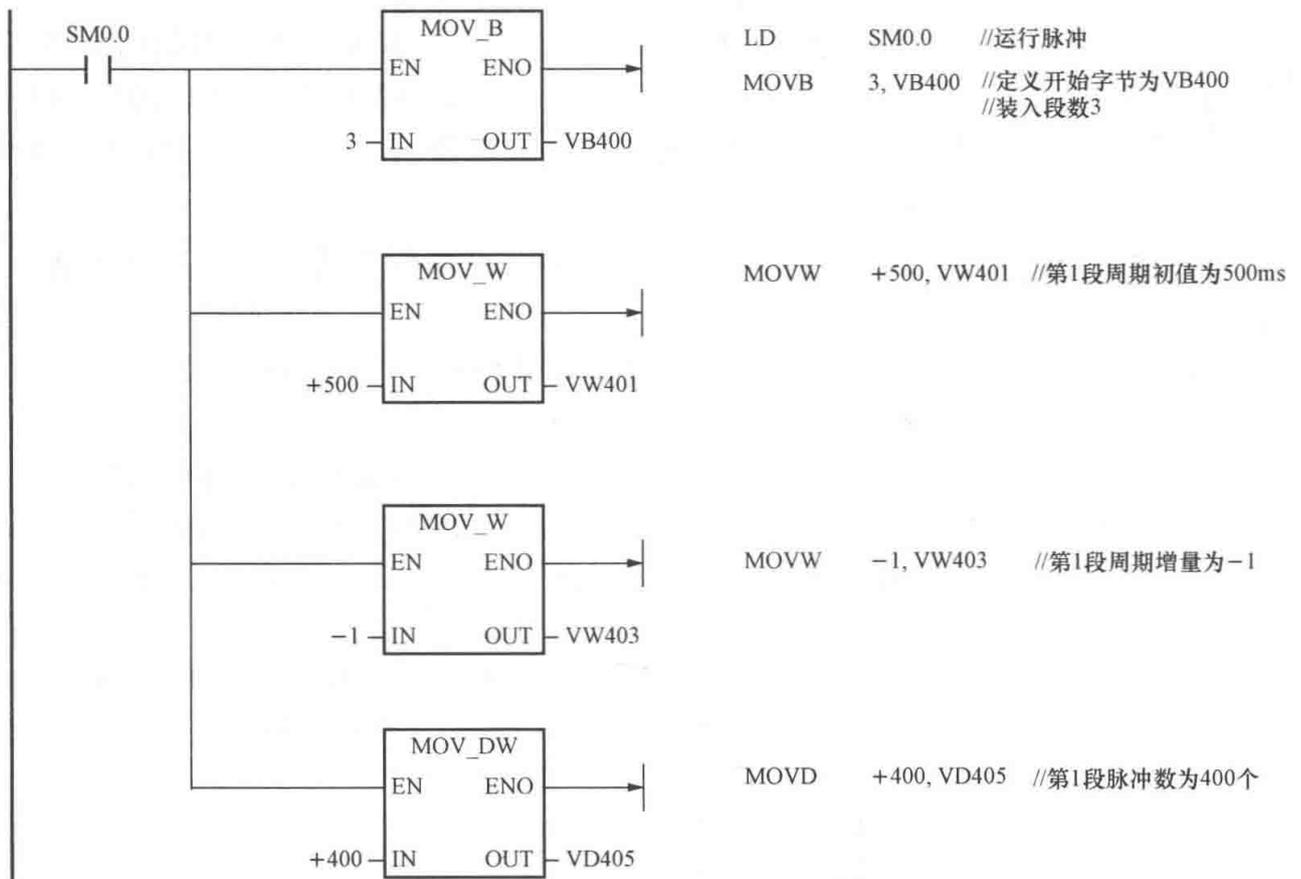


图 4-59 包络表子程序 SBR\_0 (一)

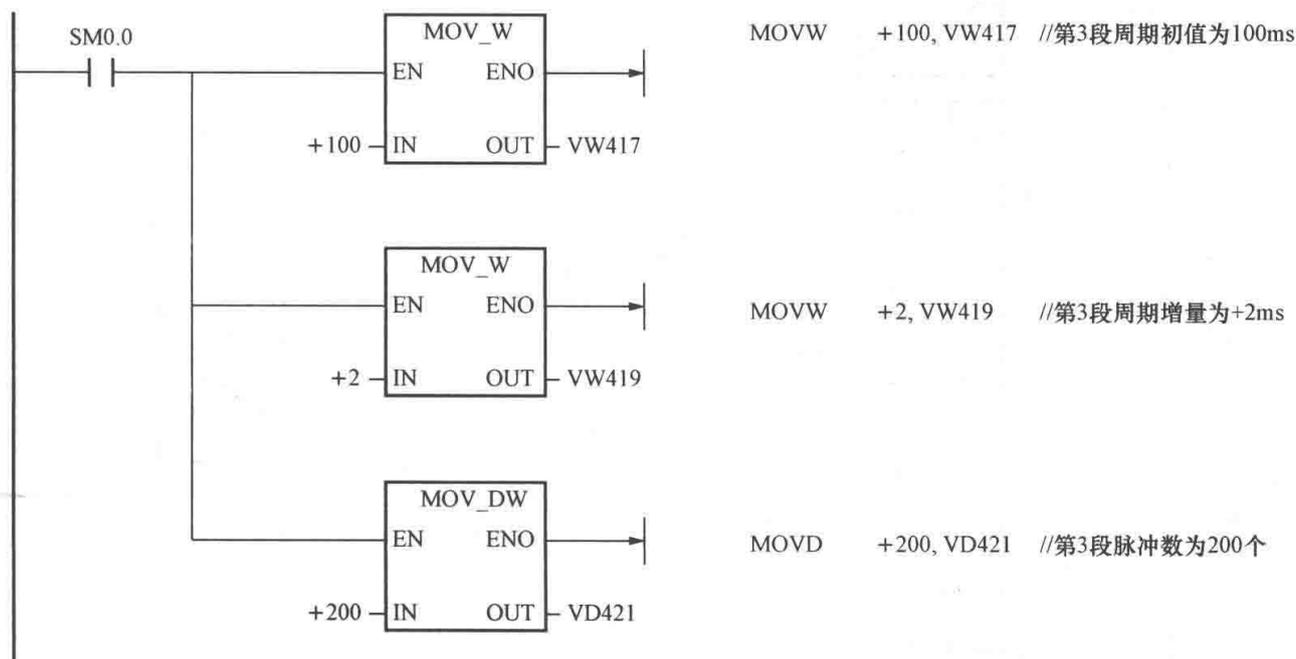
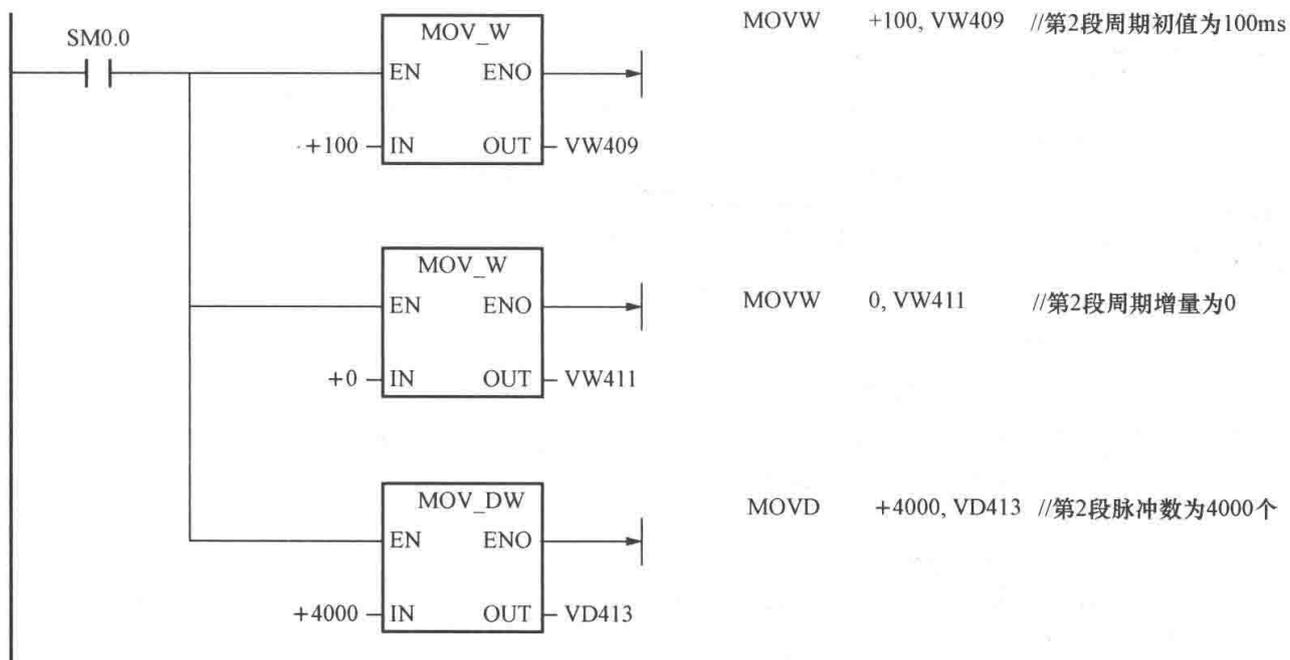


图 4 - 59 包络表子程序 SBR\_0 (二)

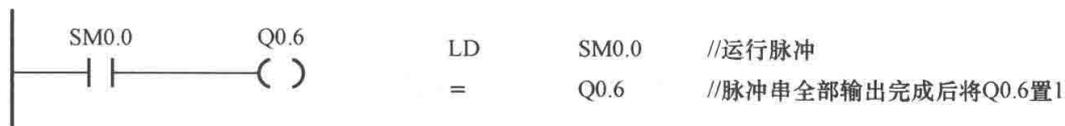


图 4 - 60 中断程序 INT\_0

**【例 4 - 23】** 要求采用同步更新方式在 Q0.1 上输出脉冲宽度调制波，允许脉宽由 10%到 50%变化。

设计的主程序如图 4 - 61 (a) 所示，初始化子程序如图 4 - 61 (b) 所示，脉宽设定值改变的子程序如图 4 - 61 (c) 所示。

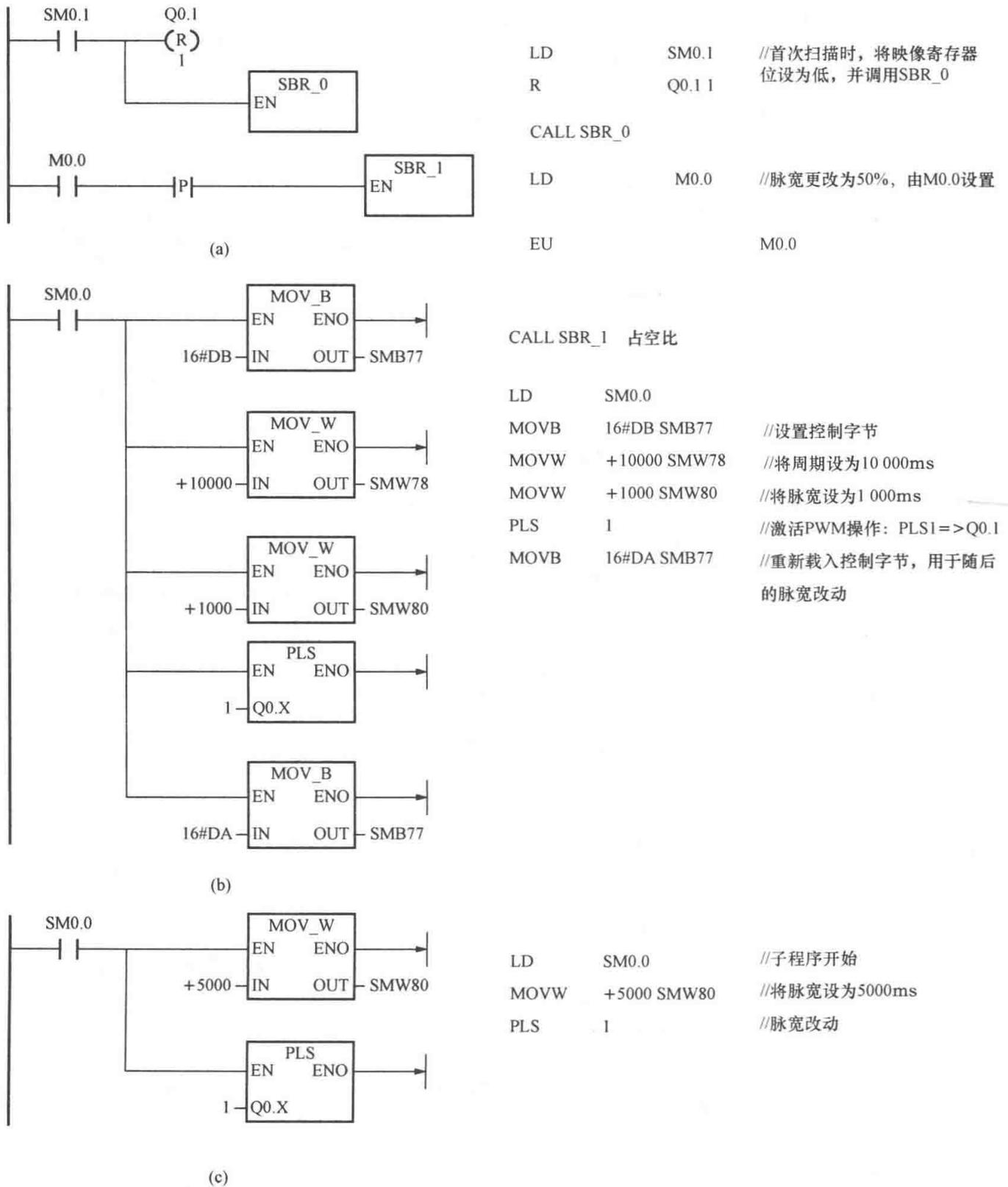


图 4-61 PWM 应用

(a) 主程序; (b) 初始化子程序; (c) 脉宽设定值改变的子程序

## 4.6 S7-200 系列 PLC 仿真软件及其应用

S7-200 仿真软件不需要安装, 可用于简单程序的仿真调试, 能模拟 PLC 的基本指令功能, 但不能模拟 S7-200 的全部指令和全部功能。下面介绍该仿真软件的应用方法。

### 1. 将要仿真调试的程序项目导出生成 ASCII 文本文件

在 STEP7-Micro/WIN V4.0 或 STEP 7-Micro/WIN SMART 2.0 编程软件中打开编译成功的程序，执行菜单命令“文件”→“导出”。生成扩展名为“awl”的 ASCII 文本文件。

### 2. 打开仿真软件

双击 S7\_200 汉化仿真 V3.0 文件夹，再双击其中的 S7\_200 汉化版.exe 软件，单击屏幕中间出现的画面，输入密码，进入仿真软件。

### 3. 硬件设置

执行菜单命令“配置”→“CPU 型号”，将 CPU 的型号改为 CPU 22X。双击紧靠已配置的模块右侧的空白方框，添加 I/O 扩展模块。

### 4. 下载程序

单击仿真软件工具栏的下载按钮，下载“\*.awl”文件。如果用户程序中有仿真软件不支持的指令或功能，单击“运行”按钮后，“RUN”LED 的状态不变，就不能运行程序。

### 5. 模拟调试程序

用鼠标单击模块下面的小开关，产生输入信号。单击工具栏上的“监视梯形图”按钮，启用梯形图程序状态功能。

### 6. 监控变量

单击工具栏上的“状态表”按钮，用出现的视图可以监视 V、M、T、C 等内部变量的值。用二进制格式监视字节、字和双字，可以在一行中同时监视多个位变量。如图 4-62 所示。

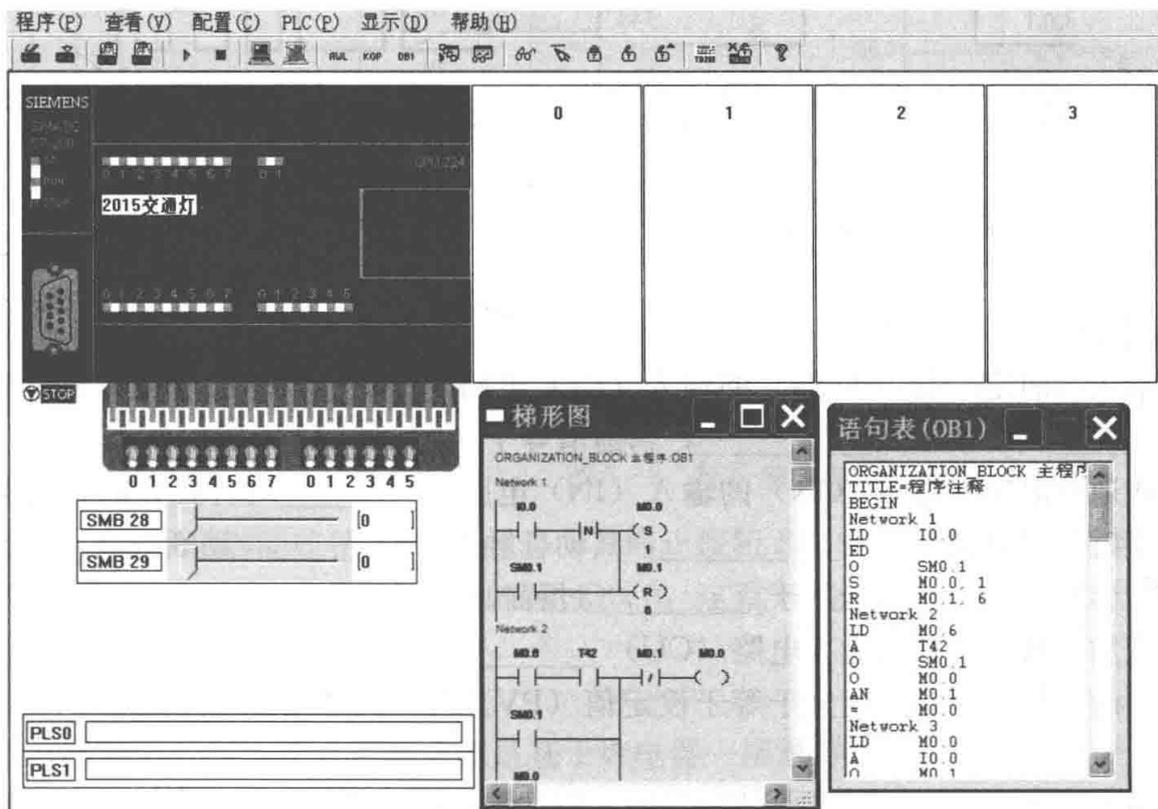


图 4-62 S7-200 仿真软件监控视图

## 习 题



1. 简述定时器指令 TON、TOF、TONR 的工作特性。
2. 编写一段程序，求  $55^\circ$  的正切值。
3. 用一个按钮控制 I0.0，用输出点 Q0.1 控制接触器 KM，接触器 KM 再控制电动机，要求按一下按钮，电动机启动，再按按钮，可以实现电动机停止。请编写程序来实现控制要求。
4. 用一个按钮控制 I0.0，用输出点 Q0.1 控制接触器 KM，接触器 KM 再控制电动机，要求按一下按钮，电动机启动，延时 6s 的时间后，电动机停止。请编写程序来实现控制要求。
5. 简述步进控制指令的使用方法及其特点，并设计如图 4-63 所示波形要求的梯形图程序。
6. 按下按钮 I0.0 后 Q0.0 变为 ON 并自保持（见图 4-64），T40 定时 8s 后，用 C0 对 I0.1 输入的脉冲计数，计满 4 个脉冲后，Q0.0 变为 OFF，同时 C0 和 T40 被复位，在 PLC 刚开始执行用户程序时，C0 也被复位，设计出梯形图。

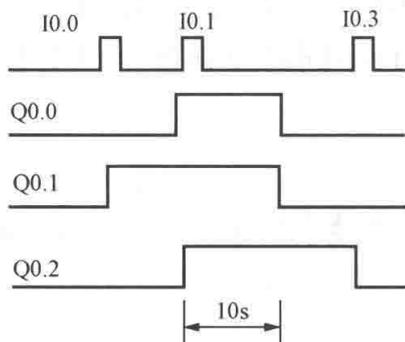


图 4-63 题 5 的图

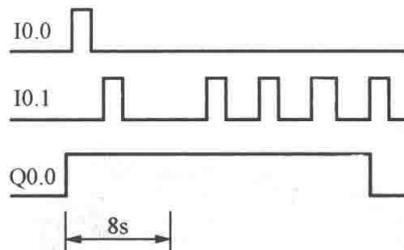


图 4-64 题 6 的图

## 7. 填空题

- (1) 接通延时定时器 (TON) 的输入 (IN) 电路\_\_\_\_\_时被复位，复位后其动合触点\_\_\_\_\_，动断触点\_\_\_\_\_，当前值等于\_\_\_\_\_。
- (2) 接通延时定时器 (TON) 的输入 (IN) 电路\_\_\_\_\_时开始定时，当前值大于等于设定值时其定时器位变为\_\_\_\_\_，其动合触点\_\_\_\_\_，动断触点\_\_\_\_\_。
- (3) 输出指令 (=) 不能用于\_\_\_\_\_过程映像寄存器。
- (4) 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。当前值大于等于设定值 (PV) 时，其动合触点\_\_\_\_\_，动断触点\_\_\_\_\_。  
复位输入电路\_\_\_\_\_时，计数器被复位，复位后其动合触点\_\_\_\_\_，动断触点\_\_\_\_\_，当前值为\_\_\_\_\_。

## 第5章 S7-200 PLC的编程方法

### 5.1 梯形图的基本电路

#### 5.1.1 应用可编程控制器实现对三相异步电动机的点动及连续运转控制

##### 1. 可编程控制器的硬件连接

实现电动机的点动及连续运行所需的器件有：启动按钮 SB1、停止按钮 SB2、交流接触器 KM、热继电器 FR 及刀开关 Q 等。主电路如图 5-1 所示。PLC 控制的外部接线如图 5-2 所示。

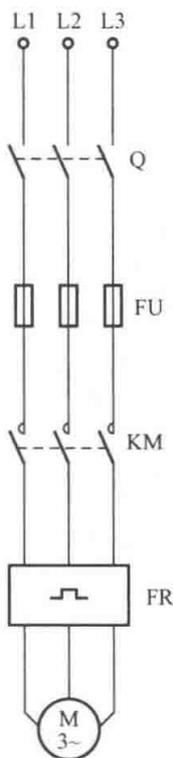


图 5-1 主电路

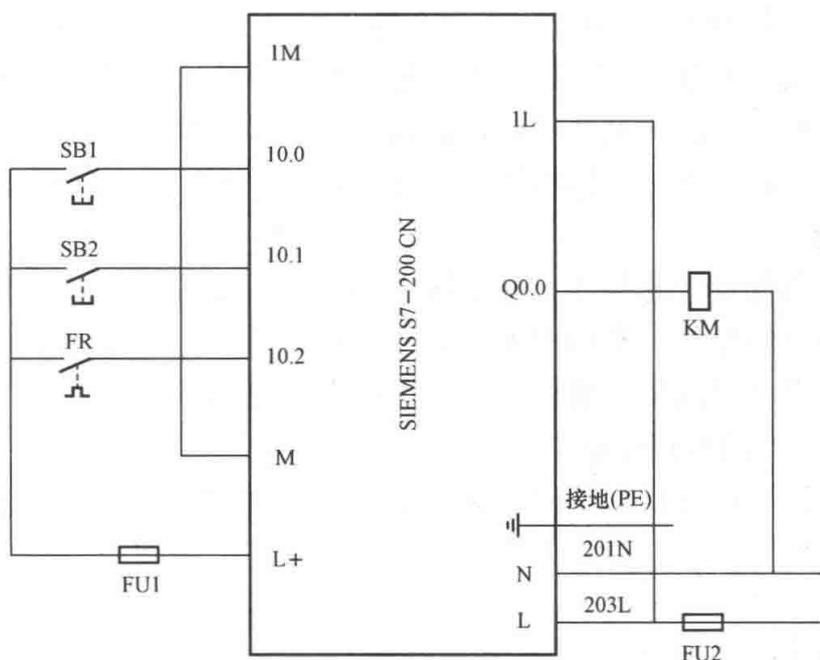


图 5-2 PLC 控制的外部接线

由图 5-2 可知，启动按钮 SB1 接于 I0.0，停止按钮 SB2 接于 I0.1，热继电器 FR 动合触点接于 I0.2，交流接触器 KM 的线圈接于 Q0.0，这就是端子分配，其实质是为程序安排控制系统中的机内元件。

##### 2. 梯形图程序的设计

可编程控制器的基本逻辑控制功能是基于继电器—接触器控制系统而设计的，而控制功能的实现是由应用程序来完成的，而用户程序是由使用者根据可编程控制器生产厂家所提供的编程语言并结合所要实现的控制任务而设计的。梯形图便是诸多编程语言中较常用的一种类型，它是以图形符号及图形符号在图中的相互关系表示控制关系的编程语言。

根据输入输出接线图可设计出异步电动机点动运行的梯形图程序，如图 5-3 (a) 所示。工作过程分析如下：当按下 SB1 时，输入继电器 I0.0 得电，其动合触点闭合，因为异

步电动机未过热,热继电器动合触点不闭合,输入继电器 I0.2 不接通,其动断触点保持闭合,则此时输出继电器 Q0.0 接通,进而接触器 KM 得电,其主触点接通电动机的电源,则电动机启动运行。当松开按钮 SB1 时,I0.0 失电,其触点断开,Q0.0 失电,接触点 KM 断电,电动机停止转动,即本梯形图可实现点动控制功能。大家可能发现,在 PLC 接线图中使用的热继电器的触点为动合触点,如果要使用动断触点,梯形图应如何设计?

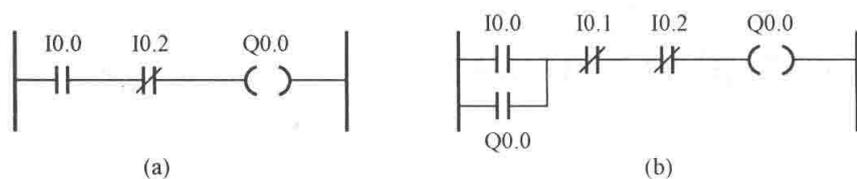


图 5-3 电动机控制梯形图

(a) 点动运行; (b) 连续运行

可编程控制器的优点之一是不改变硬件接线的情况下,通过变更软件设计,可完成不同的控制任务。图 5-3 (b) 为电动机连续运行的梯形图,其工作过程分析如下:

当按钮 SB1 被按下时, I0.0 接通, Q0.0 置 1, 这时电动机连续运行。需要停车时,按下停车按钮 SB2, 串联于 Q0.0 线圈回路中的 I0.1 的动断触点断开, Q0.0 置 0, 电动机失电停车。

图 5-3 (b) 称为启、保、停电路。这个名称主要来源于图中的自保持触点 Q0.0。并联在 I0.0 动合触点上的 Q0.0 动合触点的作用是按按钮 SB1 松开, 输入继电器 I0.0 断开时, 线圈 Q0.0 仍然能保持接通状态。工程中把这个触点叫做“自保持触点”。启、保、停电路是梯形图中最典型的单元, 它包含了梯形图程序的全部要素。它们是:

(1) 事件。每一个梯形图支路都针对一个事件。事件用输出线圈(或功能框)表示, 本例中为 Q0.0。

(2) 事件发生的条件。梯形图支路中除了线圈外还有触点的组合, 使线圈置 1 的条件即是事件发生的条件, 本例中为启动按钮 I0.0 置 1。

(3) 事件得以延续的条件。触点组合中使线圈置 1 得以持久的条件。本例中为与 I0.0 并联的 Q0.0 的自保持触点。

(4) 使事件终止的条件。触点组合中使线圈置 0 中断的条件。本例中为 I0.1 的动断触点断开。

### 3. 语句表

点动运行控制即图 5-3 (a) 所使用到的基本指令有: 从母线取用动合触点指令 LD、动断触点的串联指令 AN; 输出继电器的线圈驱动指令 =。而每条指令占用一个程序步, 语句表如下:

```
LD    I0.0
AN    I0.2
=     Q0.0
```

连续运行控制即图 5-3 (b) 所使用到的基本指令有: 从母线取用动合触点指令 LD、动合触点的并联指令 O、动断触点的串联指令 AN、输出继电器的线圈驱动指令 =。语句表如下:

```
LD    I0.0
O     Q0.0
AN    I0.2
AN    I0.2
=     Q0.0
```

### 5.1.2 应用可编程控制器实现异步电动机的 Y/ $\Delta$ 启动控制

#### 1. 继电器—接触器实现的 Y/ $\Delta$ 降压控制电路

由电动机及拖动基础可知，三相交流异步电动机启动时电流较大，一般是额定电流的5~7倍。故对于功率较大的电动机，应采用降压启动方式，Y/ $\Delta$ 降压启动是常用的方法之一。

启动时，定子绕组首先接成星形，待转速上升到接近额定转速时，再将定子绕组的接线换成三角形，电动机便进入全电压正常运行状态。图5-4为继电器—接触器实现的 Y/ $\Delta$ 降压控制电路。

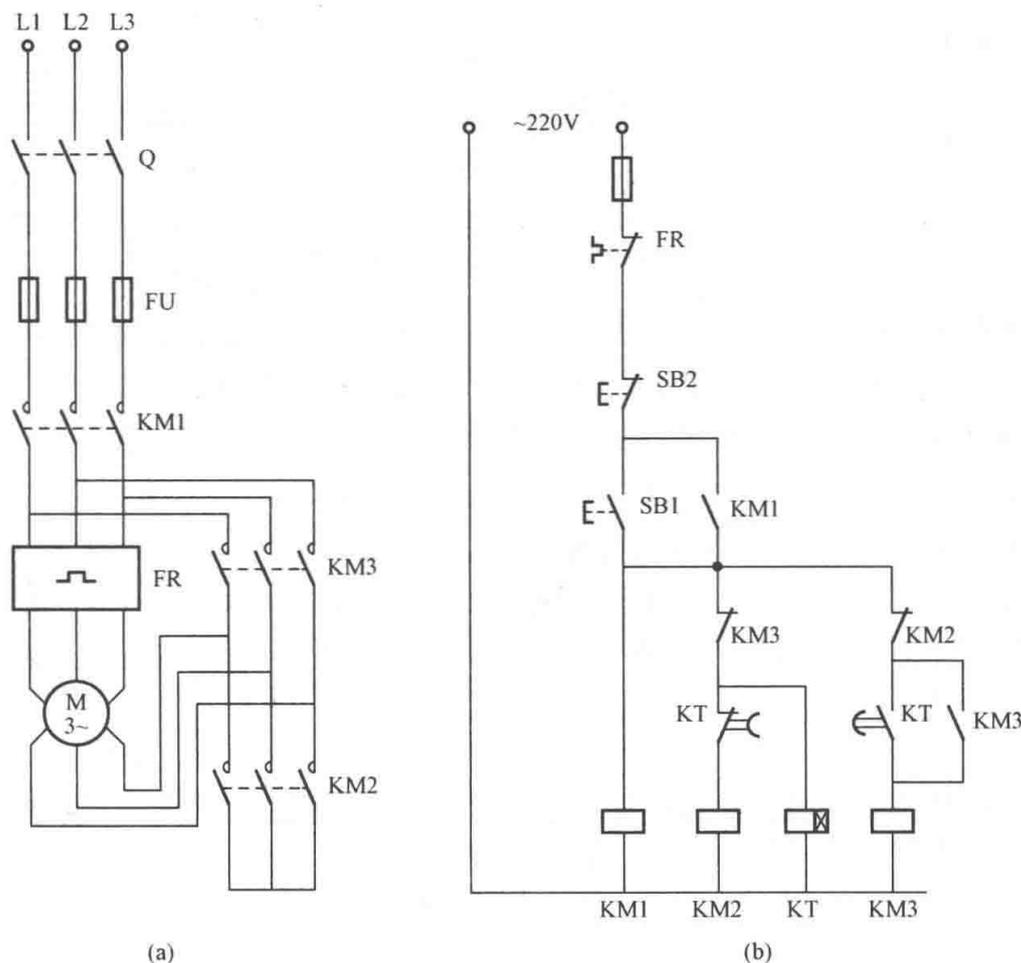


图 5-4 异步电动机 Y/ $\Delta$ 降压启动控制电路图

(a) 主电路；(b) 控制电路

它是根据启动过程中的时间变化，利用时间继电器来控制 Y/ $\Delta$ 的换接的。由图 5-4 (a) 知，工作时，首先合上刀开关 Q，当接触器 KM1 及 KM2 接通时，电动机 Y 形启动。当接触器 KM1 及 KM3 接通时，电动机  $\Delta$  形运行。

线路中 KM2 和 KM3 的动断触点构成电气互锁，保证电动机绕组只能接成一种形式，即 Y 形或  $\Delta$  形，以防止同时连接成 Y 形及  $\Delta$  形而造成电源短路。

#### 2. 可编程控制器的硬件配置

本模块所需的硬件及输入/输出端口分配如图 5-5 所示。由图可见，本模块除可编程控制器（要增加输入输出点数）、SB1、SB2、FR 之外还增添了两个接触器 KM2、KM3，其中，SB1 为启动按钮，SB2 为停止按钮，FR 为热继电器的动合触点，KM1 为主电源接触器，KM3 为  $\Delta$  形运行接触器，KM2 为 Y 形启动接触器。

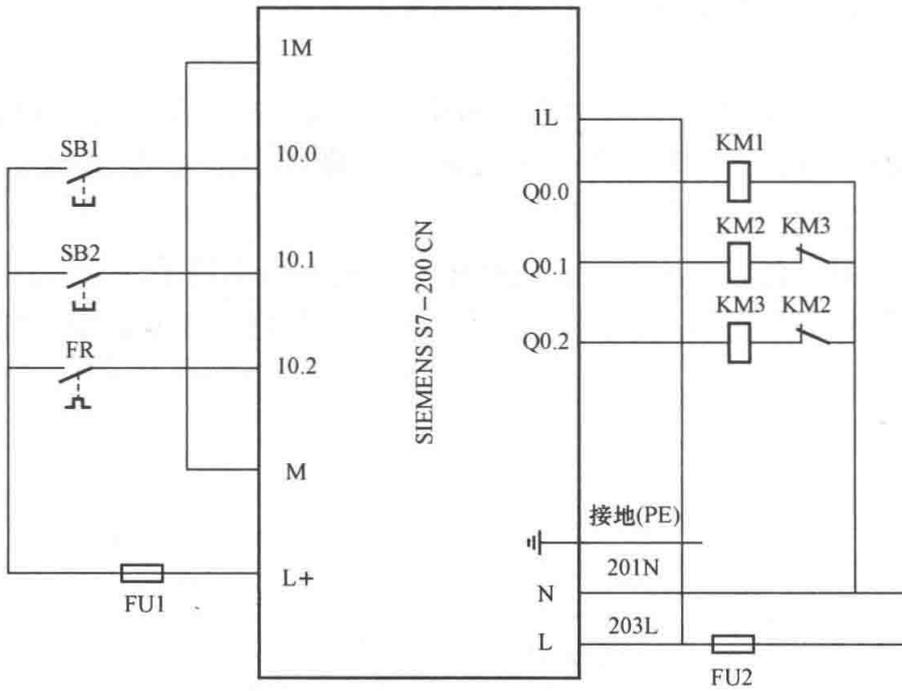


图 5-5 异步电动机 Y/Δ 降压启动的 PLC 控制电路图

### 3. 软件设计

本模块的软件设计除应用前述的部分基本指令及软元件之外，还新增软元件定时器 T40。可编程控制的梯形图程序如图 5-6 所示。

工作过程分析如下：按下启动按钮 SB1 时，输入继电器 I0.0 的动合触点闭合，如果没有故障，输出继电器 Q0.0 接通，并通过 Q0.0 的动断触点自锁，接触器 KM1 得电吸合，接着 Q0.1 接通，接触器 KM2 得电吸合，电动机在 Y 形接线方式下启动；

同时定时器 T40 开始计时，延时 8s 后 T40 动作，使 Q0.1 断开；Q0.1 断开后，KM2 失电，使输出继电器 Q0.2 接通，接触器 KM3 得电，电动机在 Δ 形接线方式下运行。

若要使电动机停止，按下 SB2 按钮或过载保护 (FR) 动作，不论电动机是启动或运行情况下都可使输出继电器断开，电动机停止运行。

在梯形图中，将 Q0.1 和 Q0.2 的动断触点分别与对方的线圈串联，可以保证它们不会同时为 ON，因此 KM2 和 KM3 的线圈不会同时通电，这种安全措施是通过程序控制来实现的，叫做“程序互锁”。

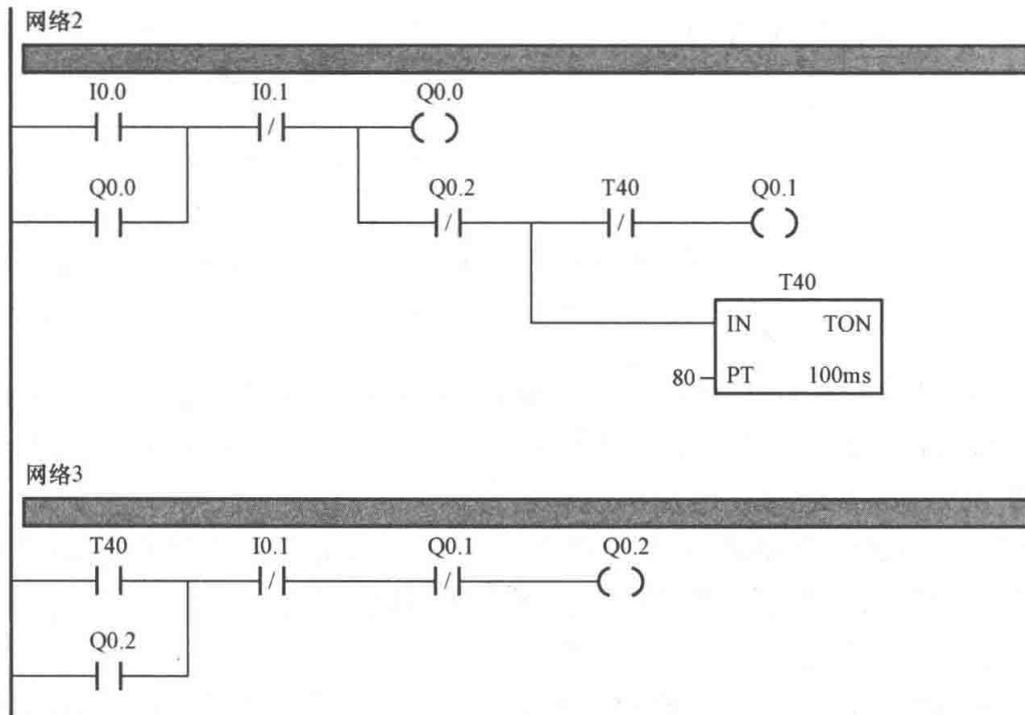


图 5-6 异步电动机 Y/Δ 降压启动的 PLC 控制梯形图程序

但是程序互锁还只能保证 PLC 输出模块中与 Q0.1 和 Q0.2 相对应的硬件继电器的动合

触点不会同时接通，由于控制电动机主回路的接触器在切换过程中电感的延时作用，可能会出现一个接触器的主触点还未断弧，另一个的主触点已经合上的现象，从而造成瞬间短路事故。这时可以在 PLC 的外部输出回路中，将 KM2 和 KM3 的辅助动断触点串联在对方接触器线圈的回路中（如图 5-5 所示）。这种安全措施叫做“硬件互锁”。

### 5.1.3 定时器、计数器的应用程序

#### 1. 定时器的延时扩展

定时器的计时时间都有一个最大值，如 100ms 的定时器最大定时时间为 32767.7s。如工程中所需的延时时间大于这个数值时，一个最简单的方法是采用定时器接力方式，即先启动一个定时器计时，定时时间到时，用第一只定时器的动合触点启动第二只定时器，再使用第二只定时器启动第三只，如此等等。记住使用最后一个定时器的触点去控制最终的控制对象就可以了。图 5-7 中的梯形图即是一个这样的例子。

所以利用多定时器的计时时间相加可以获得长延时。此外还可以利用定时器配合计数器获得长延时，如图 5-8 所示。图中动合触点 I0.1 是这个电路的工作条件，当 I0.1 保持接通时电路工作。在定时器 T60 的线圈回路中接有定时器 T60 的动断触点，它使得定时器 T60 的动合触点每隔 30s 接通一次，接通时间为一个扫描周期。定时器 T60 动合触点的每一次接通都使计数器 C100 计一个数。当计数器的当前值等于计数器的设定值时，计数器的动合触点闭合，使输出继电器 Q0.0 接通，从 I0.1 接通为始点的延时时间为定时器的设定值乘上计数器的设定值。I0.2 为计数器 C100 的复位条件。

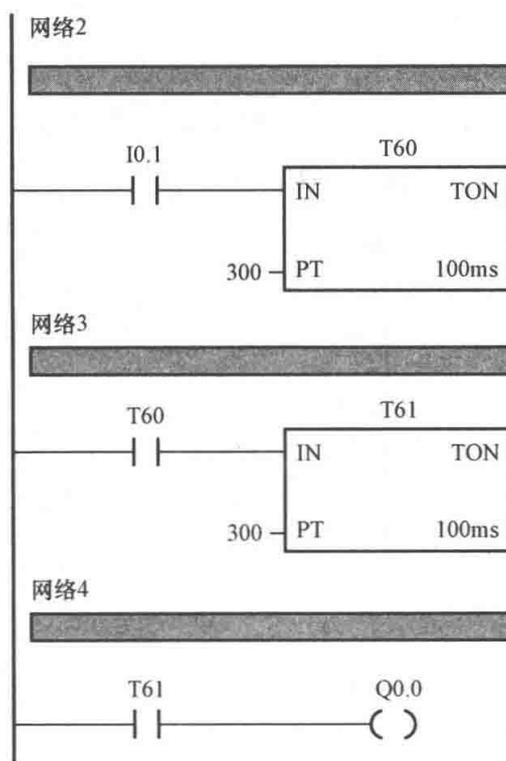


图 5-7 时间延长的方法 1

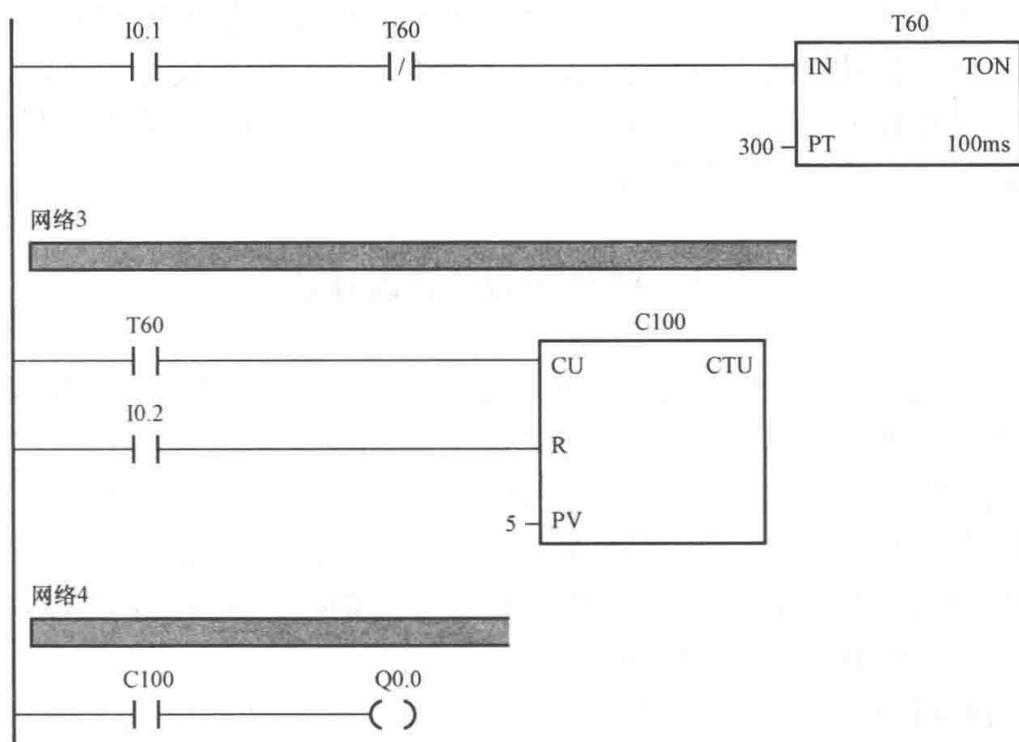


图 5-8 时间延长的方法 2

## 2. 闪烁电路

采用基本逻辑的编程实现信号灯的控制。灯亮采用编程软件定时器实现，灯闪采用由定时器组成的脉冲发生器实现。现在我们来分析一下由 T60 及 T61 组成脉冲发生器的梯形图。

由图 5-9 可知，当 I0.1 闭合时，T60 得电，延时 0.5s 后，T60 动合触点闭合，定时器 T61 得电，延时 0.5s 后，其动断触点 T61 断开，T60 线圈失电，T60 动合触点断开，而定时器 T60 再次得电，0.5s 后，T60 的动合触点再次闭合……，如此周而复始，即可得到 T60 触发的输出波形，如图 5-9 所示。

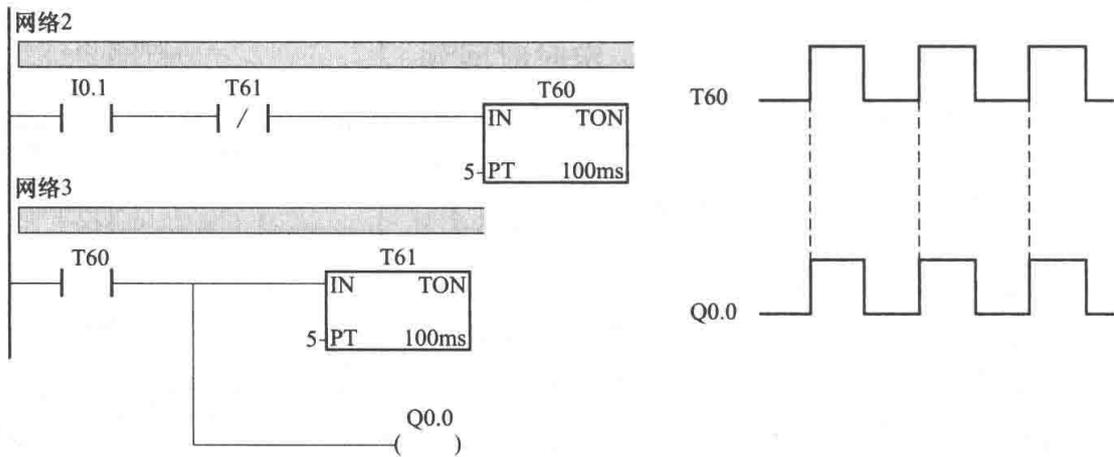


图 5-9 闪烁控制及 Q0.0 的输出波形

### 5.1.4 动断触点输入信号的处理

如果要将图 5-2 中 FR 的动合触点换成动断触点。没有过载时 FR 的动断触点闭合，I0.2 为 1 状态，其动合触点闭合，动断触点断开。为了保证没有过载时电动机的正常运行，很显然应在 Q0.0 的线圈回路中串联 I0.2 的动合触点。过载时 FR 的动断触点断开，I0.2 为 0 状态，其动合触点断开，使 Q0.0 的“线圈”断电，起到了保护作用。

实际上，有了 PLC 之后，输入的开关量信号均可由外部动合触点提供（推荐使用）。但有些特殊情况，某些信号只能用动断触点输入，此时，PLC 程序设计的处理方法是：可以按输入全部为动合触点来设计，然后将梯形图中相应的输入位的触点改为相反的触点，即动合触点改为动断触点，动断触点改为动合触点。

## 5.2 梯形图的经验设计法

经验设计法类似于通常设计继电器电路图的方法，在一些典型电路程序设计的基础上，根据被控对象对控制系统的具体要求，不断地修改和完善梯形图。

经验设计法的特点：无规律可循，有较大的随意性和试探性，结果不是唯一的，与设计者的经验有很大的关系。它一般用于较简单的梯形图（如手动程序）的设计。

自动往返控制的梯形图设计：图 5-10 中的小车开始时停在左边，左限位开关 SQ1 的动合触点闭合。要求按下列顺序控制小车：

- (1) 按下右行启动按钮 SB1，小车右行。
- (2) 走到右限位开关 SQ2 处停止运动，延时 5s 后开始左行。

(3) 回到左限位开关 SQ1 处时停止运行。

在异步电动机正反转控制电路的基础上设计的满足上述要求的梯形图如图 5-10 所示。在控制右行的 Q0.0 的线圈回路中串联了 I0.4 的动断触点，小车走到右限位开关 SQ2 处时，I0.4 的动断触点断开，使 Q0.0 的线圈断电，小车停止右行。同时 I0.4 的动合触点闭合，T100 的线圈通电，开始定时。8s 后定时时间到，T100 的动合触点闭合，使 Q0.1 的线圈通电并自保持，小车开始左行。离开限位开关 SQ2 后，I0.4 的动合触点断开，T100 的动合触点因为其线圈断电而断开。小车运行到左边的起始点时，左限位开关 SQ1 的动合触点闭合，I0.3 的动合触点断开，使 Q0.1 的线圈断电，小车停止运动。

设计的梯形图程序如图 5-11 所示。在梯形图中，保留了左行启动按钮和停止按钮分别对应的输入寄存器的触点，使系统具有手动操作的功能。

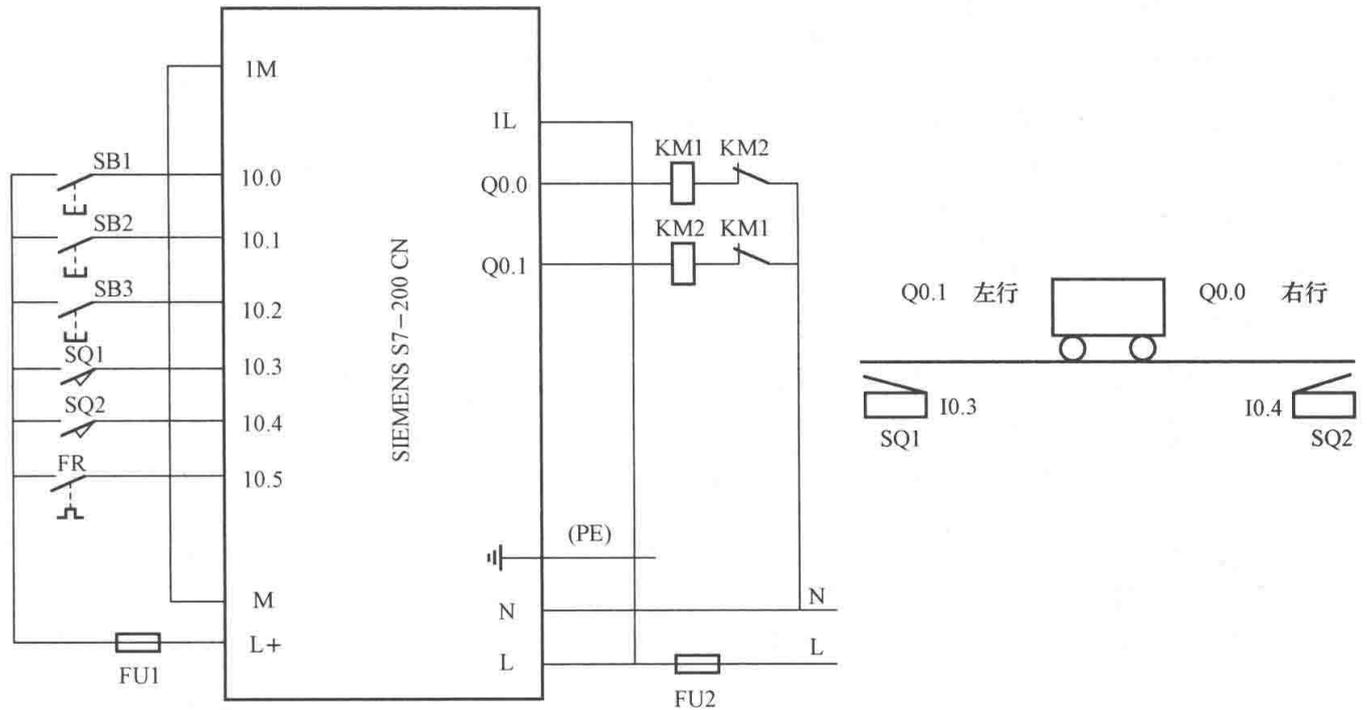


图 5-10 PLC 的原理电路图及小车运动示意图

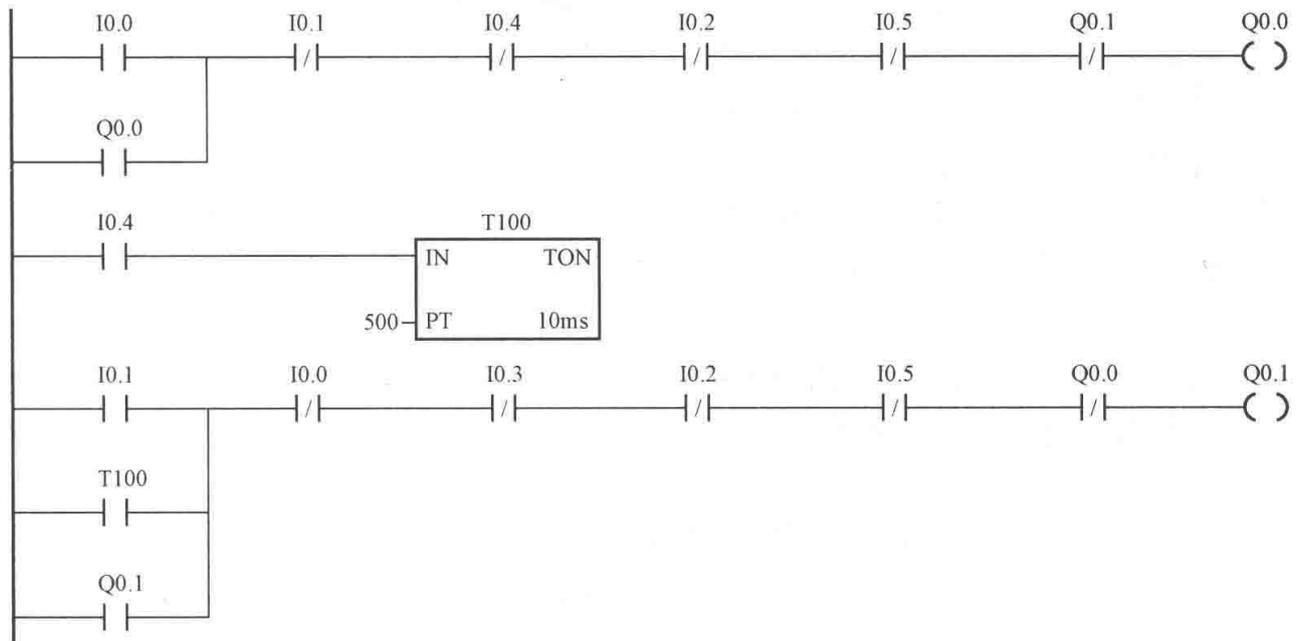


图 5-11 小车控制的梯形图程序

程序中有多处软件互锁的功能,除了用 Q0.1 和 Q0.2 的动断触点分别与对方的线圈串联来保证 KM1 和 KM2 的线圈不会同时通电外,程序中还采用了由输入映像继电器 I0.0、I0.1 的动断触点来实现正反操作的互锁,采用了由输入映像继电器 I0.3、I0.4 的动断触点实现左右限位的互锁。

较复杂的自动往返小车控制程序,在图 5-11 的基础上所示系统的基础上,增加左端延时功能,即小车碰到限位开关 X3 后停止左行,延时 3s 后自动右行。程序设计由学习者思考。

## 5.3 顺序控制设计法与顺序功能图

### 5.3.1 顺序控制设计法

用经验设计法设计梯形图时,没有一套固定的方法和步骤可以遵循,具有很大的试探性和随意性,对于不同的控制系统,没有一种通用的容易掌握的设计方法。在设计复杂系统的梯形图时,用大量的中间单元来完成记忆、联锁和互锁等功能,由于需要考虑的因素很多,它们往往又交织在一起,分析起来非常困难,一般不可能把所有问题都考虑得很周到,程序设计出来后,需要模拟调试或在现场调试,发现问题后再针对问题对程序进行修改。即使是非常有经验的工程师,也很难做到设计出的程序能一次成功。修改某一局部电路时,很可能会引发出别的问题,对系统的其他部分产生意想不到的影响,因此梯形图的修改也很麻烦,往往花了很长的时间还得不到一个满意的结果。用经验法设计出的梯形图很难阅读,给系统的维修和改进带来了很大的困难。

所谓顺序控制,就是按照生产工艺预先规定的顺序,在各个输入信号的作用下,根据内部状态和时间的顺序,在生产过程中各个执行机构自动地有秩序地进行操作。使用顺序控制设计法时首先根据系统的工艺过程,画出顺序功能图(Sequential Function Chart),然后根据顺序功能图画出梯形图。

顺序控制设计法是一种先进的设计方法,很容易被初学者接受,对于有经验的工程师,也会提高设计的效率,节约大量的设计时间。程序的调试、修改和阅读也很方便。只要正确地画出了描述系统工作过程的顺序功能图,一般都可以做到调试程序时一次成功。

顺序控制设计法最基本的思想是将系统的一个工作周期划分为若干个程序相连的阶段,这些阶段称为步(Step),然后用编程元件(如存储器位 M)来代表各步,步是根据输出量的 ON/OFF 状态的变化来划分的,在任何一步之内,各输出量的状态不变,但是相邻两步输出量总的状态是不同的,步的这种划分方法使代表各步的编程元件的状态与各输出量的状态之间有着极为简单的逻辑关系。

使系统由当前步进入下一步的信号称为转换条件,转换条件可以是外部的输入信号,如按钮、指令开关、限位开关的接通/断开等,也可以是 PLC 内部产生的信号,如定时器、计数器的触点提供的信号,还可能是若干个信号的与、或、非逻辑组合。

顺序控制设计法用转换条件控制代表各步的编程元件,让它们的状态按一定的顺序变化,然后用代表各步的编程元件去控制 PLC 的各输出位。

顺序功能图并不涉及所描述的控制功能的具体技术,它是一种通用的直观的技术语言,可以供进一步设计和不同专业的人员之间进行技术交流。对于熟悉设备和生产流程的现场情

况的电气工程师来说，顺序功能图是很容易画出的。

在 IEC 的 PLC 标准 (IEC 61131) 中，顺序功能图是 PLC 位居首位的编程语言。顺序功能图主要由步、有向连线、转换、转换条件和动作（或命令）组成。

### 5.3.2 步与动作

#### 1. 步

图 5-12 是液压动力滑台的进给运动示意图和输入输出信号的时序图，为了节省篇幅，将几个脉冲输入信号的波形画在一个波形图中。设动力滑台在初始位置时停在左边，限位开关 I0.3 为 1 状态，Q0.0~Q0.2 是控制动力滑台运动的 3 个电磁阀。按下启动按钮后，动力滑台的一个工作周期由快进、工进、暂停和快退组成，返回初始位置后停止运动。根据 Q0.0~Q0.2 的 ON/OFF 状态的变化，一个工作周期可以分为快进、工进、暂停和快退 4 步，另外还应设置等待启动的初始步，图中分别用 M0.0~M0.4 来代表这 5 步。图 5-12 是描述该系统的顺序功能图，图中用矩形方框表示步，方框中可以用数字表示各步的编号，也可以用代表各步的存储器位的地址作为步的编号，例如 M0.0 等，这样在根据顺序功能图设计梯形图时就较为方便。

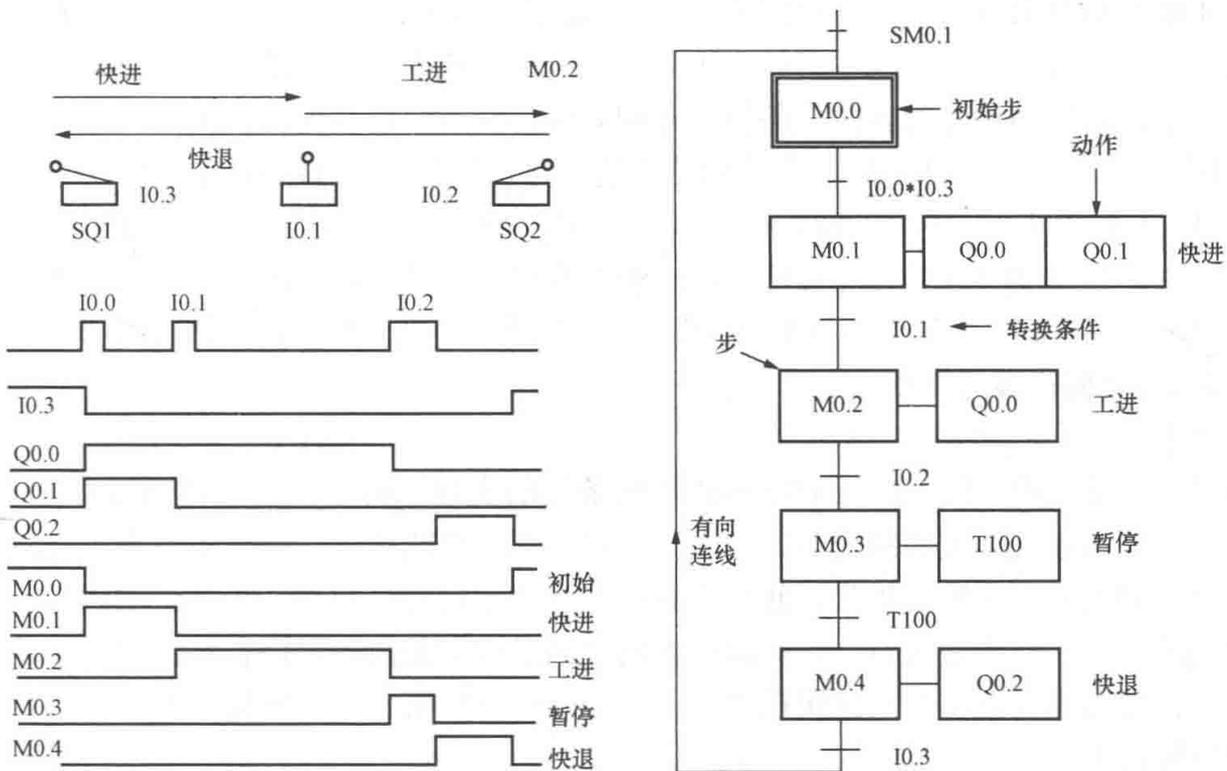


图 5-12 液压动力滑台的进给运动示意图和输入输出信号的时序图及顺序功能图

#### 2. 初始步

初始状态一般是系统等待启动命令的相对静止的状态。系统在开始进行自动控制之前，首先应进入规定的初始状态。与系统的初始状态相对应的步称为初始步，初始步用双线方框来表示，每一个顺序功能图至少应该有一个初始步。

#### 3. 与步对应的动作或命令

可以将一个控制系统划分为被控制系统和施控系统，例如在数控车床系统中，数控装置是施控系统，而车床是被控系统。对于被控系统，在某一步中要完成某些“动作” (action)；对于施控系统，在某一步中则要向被控系统发出某些“命令” (command)。为了

叙述方便,下面将命令或动作统称为动作,并用矩形框中的文字或符号来表示动作,该矩形框与相应的步的方框用水平短线相连。

如果某一步有几个动作,可以用图 5-13 中的两种画法来表示,但是并不隐含这些动作之间的任何顺序。

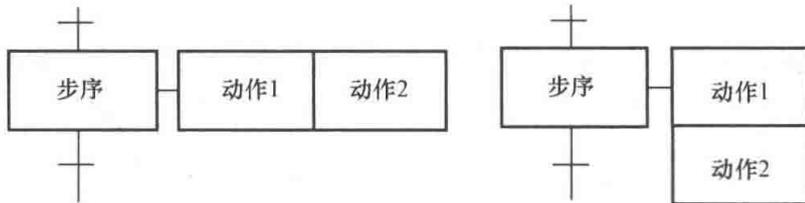


图 5-13 动作的表示方法

当系统正处于某一步所在的阶段时,该步处于活动状态,称该步为“活动步”,步处于活动状态时,相应的动作被执行;处于不活动状态时,相应的非存储型动作被停止执行。

说明命令的语句应清楚地表明该命令是存储型的还是非存储型的。非存储型动作“打开 1 号阀”,是指该步为活动步时打开 1 号阀,为不活动时关闭 1 号阀。非存储型动作与它所在的步是“同生共死”的,例如图 5-12 中的 M0.4 与 Q0.2 的波形完全相同,它们同时由 0 状态变为 1 状态,又同时由 1 状态变为 0 状态。

某步的存储型命令“打开 1 号阀并保持”,是指该步为活动步时 1 号阀被打开,该步变为不活动步时继续打开,直到在某一步 1 号阀被复位。在表示动作的方框中,可以用 S 和 R 来分别表示对存储型动作的置位(例如打开阀并保持)和复位(例如关闭阀门)。

在图 5-12 的暂停步中,PLC 所有的输出量均为 0 状态。接通延时定时器 T100 用来给暂停步定时,在暂停步,T100 的线圈应一直通电,转换到下一步后,T100 的线圈断电。从这个意义上来说,T100 的线圈相当于暂停步的一个非存储型的动作,因此可以将这种为某一步定时的接通延时定时器放在与该步相连的动作框内,它表示定时器的线圈在该步内“通电”。

### 5.3.3 有向连线与转换

#### 1. 有向连线

在顺序功能图中,随着时间的推移和转换条件的实现,将会发生步的活动状态的进展,这种进展按有向连线规定的路线和方向进行。在画顺序功能图时,将代表各步的方框按它们成为活动步的先后次序顺序排列,并且用有向连线将它们连接起来。步的活动状态习惯的进展方向是从上到下或从左至右,在这两个方向有向连线上的箭头可以省略。如果不是上述的方向,应在有向连线上用箭头注明进展方向。在可以省略箭头的有向连线上,为了更易于理解也可以加箭头。

如果在画图时有向连线必须中断,例如在复杂的图中,或用几个图来表示一个顺序功能图时,应在有向连线中断之处标明下一步的标号和所在的页数。

#### 2. 转换

转换用有向连线上与有向连线垂直的短画线来表示,转换将相邻两步分隔开。步的活动状态的进展是由转换的实现来完成的,并与控制过程的发展相对应。

#### 3. 转换条件

转换条件是与转换相关的逻辑命题,转换条件可以用文字语言来描述,例如“触点 A 与触点 B 同时闭合”,可以用表示转换的短线旁边的布尔代数表达式来表示,例如  $I0.1 + I2.0$ ,一般用布尔代数表达式来表示转换条件。

图 5-14 中用高电平表示步 M10.0 为活动步,反之则用低电平来表示。转换条件 I0.1

表示 I0.1 为 1 状态时转换实现，转换条件表示 I2.0 为 0 状态时转换实现。转换条件 I0.1 + I2.0 表示 I0.1 的动合触点闭合或 I2.0 的动断触点闭合时转换实现，在梯形图中则用两个触点的并联来表示这样的“或”逻辑关系。

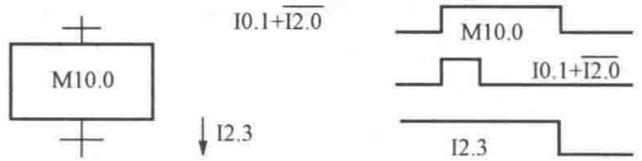


图 5-14 转换与转换条件

符号  $\uparrow I2.3$  和  $\downarrow I2.3$  分别表示当 I2.3 从 0 状态变为 1 状态和从 1 状态变为 0 状态时转换实现。实际上转换条件  $\uparrow I2.3$  和 I2.3 是等效的，因为一旦 I2.3 由 0 状态变为 1 状态（即在 I2.3 的上升沿），转换条件 I2.3 也会马上起作用。

在图 5-12 中，转换条件 T100 相当于接通延时定时器 T100 的动合触点，即在 T100 的定时时间到时转换条件满足。

### 5.3.4 顺序功能图的基本结构

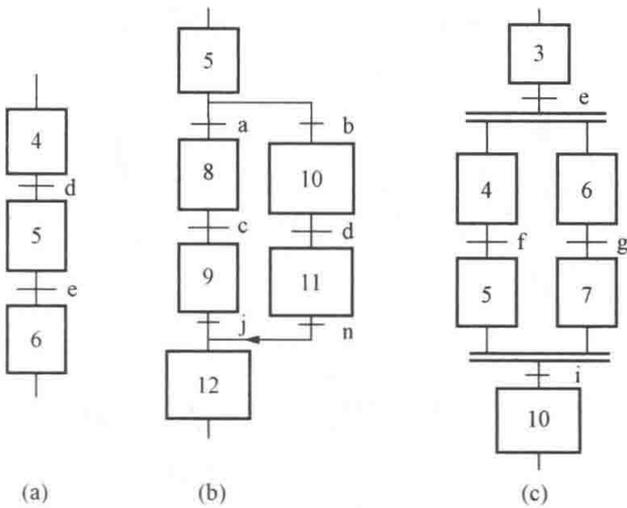


图 5-15 单序列、选择序列与并行序列

(a) 单序列；(b) 选择序列；(c) 并行序列

#### 1. 单序列

单序列由一系列相机激活的步组成，每一步的后面仅有一个转换，每一个转换的后面只有一个步 [见图 5-15 (a)]，单序列的特点是没有分支与合并。

#### 2. 选择序列

选择序列的开始称为分支 [见图 5-15 (b)]，转换符号只能标在水平连线之下。如果步 5 是活动步，并且转换条件  $a=1$ ，则发生由步 5→步 8 的进展。如果步 5 是活动步，并且  $b=1$ ，则发生由步 5→步 10 的进展。

在步 5 之后选择序列的分支处，每次只允许选择一个序列，如果将选择条件 b 改为  $a * \bar{b}$ ，则当 a 和 b 同时为 ON 时，将优先选择 a 对应的序列。

选择序列的结束称为合并 [见图 5-15 (b)]，几个选择序列合并到一个公共序列时，用需要重新组合的序列相同数量的转换符号和水平连线来表示，转换符号只允许标在水平连线之上。

如果步 9 是活动步，并且转换条件  $j=1$ ，则发生由步 9→步 12 的进展。如果步 10 是活动步，并且  $n=1$ ，则发生由步 10→步 12 的进展。

允许选择序列的某一条分支上没有步，但是必须有一个转换，这种结构称为“跳步”。跳步是选择序列的一种特殊情况。

#### 3. 并行序列

并行序列的开始称为分支 [见图 5-15 (c)]，当转换的实现导致几个序列同时激活时，这些序列称为并行序列。当步 3 是活动的，并且转换条件  $e=1$ ，4 和 6 这两步同时变为活动步，同时步 3 变为不活动步。为了强调转换的同步实现，水平连线用双线表示。步 4、6 被同时激活后，每个序列中活动步的进展将是独立的。在表示同步的水平双线之上，只允许有一个转换符号。并行序列用来表示系统的几个同时工作的独立部分的工作情况。

并行序列的结束称为合并 [见图 5-15 (c)]，在表示同步的水平双线之下，只允许有一个

转换符号。当直接连在双线上的所有前级步 (步 5、7) 都处于活动状态, 并且转换条件  $i=1$  时, 才会发生步 5、7 到步 10 的进展, 即步 5、7 同时变为不活动步, 而步 10 变为活动步。

4. 复杂的顺序功能图举例

如图 5-16 所示, 某专用钻床用来加工圆盘状零件上均匀分布着 6 个孔, 上面是侧视图, 下面是工件的俯视图。在进入自动运行之前, 两个钻头应在最上面, 上限位开关 I0.3 和 I0.5 为 ON, 系统处于初始步, 复位计数器 C0 当前值清 0。在图 5-17 中用存储器位 M 来代表各步, 顺序功能图中包含了选择序列和并行序列。操作人员放好工件后, 按下启动按钮 I0.0, 转换条件满足, 由初始步转换到步 M0.1, Q0.0 变为 ON, 工件被夹紧。夹紧后压力继电器 I0.1 为 ON, 由步 M0.1 转换到步 M0.2 和 M0.5, Q0.1 和 Q0.3 使两只钻头同时开始向下钻孔。大钻头钻到由限位开关 I0.2 设定的深度时, 进入步 M0.3, Q0.2 使大钻头上 升, 升到由限位开关 I0.3 设定的起始位置时停止上升, 进入等待步 M0.4。小钻头钻到由限位开关 I0.4 设定的深度时, 进入步 M0.6, Q0.4 使小钻头上 升, 计数器 C0 的当前值加 1, 升到由限位开关 I0.5 设定的起始位置时停止上升, 进入等待步 M0.7, 若未达到设定值 3, C0 的动断触点闭合, 转换条件  $\overline{C0}$  满足, 将转换到步 M1.0。Q0.5 使工件旋转 120°, 旋转到位时 I0.6 为 ON, 又返回步 M0.2 个 M0.5, 开始钻第二对孔。3 对孔都钻完后, 计数器的当前值变为 3, 其动合触点闭合, 转换条件 C0 满足, 进入步 M1.1, Q0.6 使工件松开。松开到位时, 限位开关 I0.7 为 ON, 系统返回初始步 M0.0。

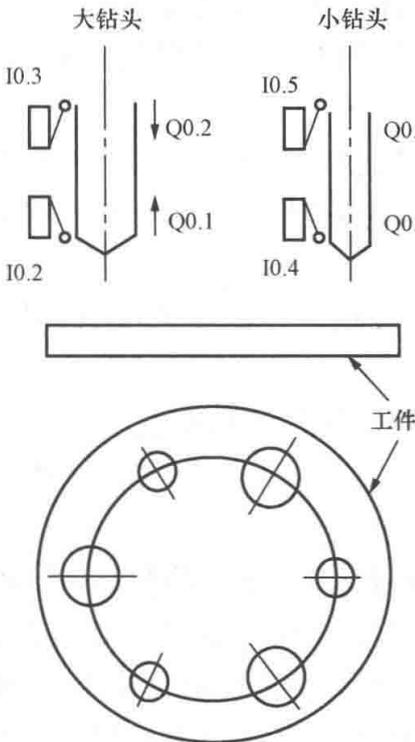


图 5-16 组合钻床示意图

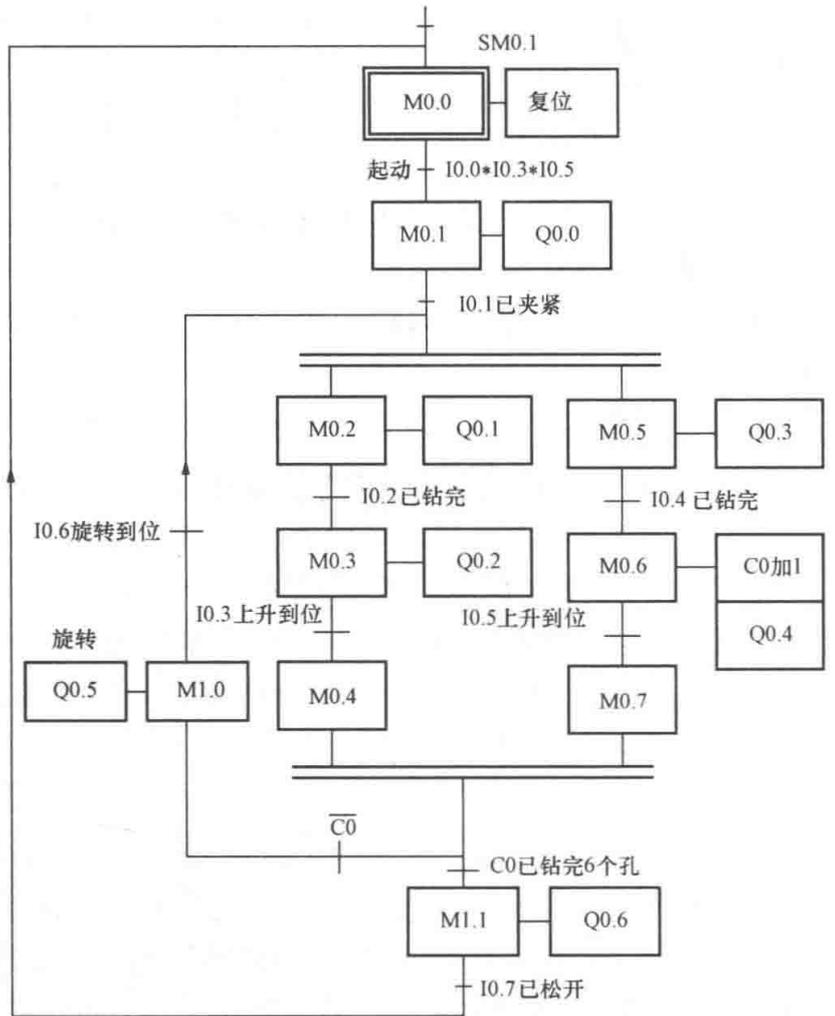


图 5-17 组合钻床的顺序功能图

因为要求两个钻头向下钻孔和钻头提升的过程同时进行,采用并行序列来描述上述的过程。由 M0.2~M0.4 和 M0.5~M0.7 组成的两个单序列分别用来描述大钻头和小钻头的工件过程。在步 M0.1 之后,有一个并行序列的分支。当 M0.1 为活动步,且转换条件 I0.1 得到满足 (I0.1 为 1 状态),并行序列中两个单序列中的第 1 步 (步 M0.2 和 M0.5) 同时变为活动步。此后两个单序列内部各步的活动状态的转换是相互独立的,例如大孔和小孔钻完时的转换一般不是同步的。

两个单序列中的最后 1 步 (步 M0.4 和 M0.7) 应同时变为不活动步。但是两个钻头一般不会同时上升到位,不可能同时结束运动,所以设置了等待步 M0.4 和 M0.7,它们用来同时结束两个并行序列。当两个钻头均上升到位,限位开关 I0.3 和 I0.5 分别为 1 状态,大、小钻头两个子系统分别进入两个等待步,并行序列将会立即结束。

在步 M0.4 和 M0.7 之后,有一个选择序列的分支。没有钻完 3 对孔时 C0 的动断触点闭合,转换条件满足  $\overline{C0}$ ,如果两个钻头都上升到位,将从步 M0.4 和 M0.7 转换到步 M1.0。如果已钻完 3 对孔, C0 的动合触点闭合,转换条件 C0 满足,将从步 M0.4 和 M0.7 转换到步 M1.1。

在步 M0.1 之后,有一个选择序列的合并。当步 M0.1 为活动步,而且转换条件 I0.1 得到满足 (I0.1 为 ON),将转换到步 M0.2 和 M0.5。当步 M1.0 为活动步,而且转换条件 I0.6 得到满足,也会转换到步 M0.2 和 M0.5。

### 5.3.5 顺序功能图中转换实现的基本规则

#### 1. 转换实现的条件

在顺序功能图中,步的活动状态的进展是由转换的实现来完成的。转换实现必须同时满足以下两个条件:

- (1) 该转换所有的前级步都是活动步。
- (2) 相应的转换条件得到满足。

如图 5-18 所示,如果转换的前级步或后续步不止一个,转换的实现称为同步实现。为了强调同步实现,有向连线的水平部分用双线表示。

#### 2. 转换实现应完成的操作

转换实现时应完成以下两个操作:

(1) 使所有由有向连线与相应转换符号相连的后续步都变为活动步。

(2) 使所有由有向连线与相应转换符号相连的前级步都变为不活动步。

以上规则可以用于任意结构中的转换,其区别如下:在单序列中,一个转换仅有一个前级步和一个后续步。在选择序列的分支与合并处,一个转换也只有一个前级步和一个后续步,但是一个步可能有多个前级步或多个后续步 (见图 5-15)。在并行序列的分支处,转换有几个后续步 (见图 5-18),在转换实现时应同时将它们对应的编程元件位置。在并行序列的合并处,转换有几个前级步,它们均为活动步时才有可能实现转换,在转换实现时,应将它们对应的编程元件全部复位。

转换实现的基本规则是根据顺序功能图设计梯形图的基础,它适用于顺序功能图中的各种基本结构,也是下面要介绍的各种设计顺序控制梯形图的方法的基础。

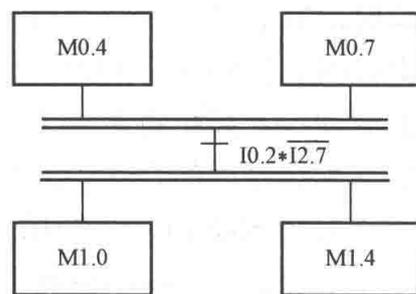


图 5-18 转换的同步实现

在梯形图中,用编程元件(如存储器位 M)来代表步,当某步为活动步时,该步对应的编程元件为 1 状态。当该步之后的转换条件满足时,转换条件对应的触点或电路接通,因此可以将该触点或电路与代表所有前级步的编程元件的动合触点串联,作为与转换实现的两个条件同时满足对应的电路。例如,图 5-18 中转换条件的布尔代数表达式  $I_{0.2} \times \overline{I_{2.7}}$ ,它的两个前级步用 M0.4 和 M0.7 来表示,所以应将 I2.7 的动断触点和 I0.2、M0.4、M0.7 的动合触点串联,作为转换实现的两个条件同时满足对应的电路。在梯形图中,该电路接通时,应使所有代表前级步的编程元件(M0.4 和 M0.7)复位,同时使所有代表后续步的编程元件(M1.0 和 M1.4)置位(变为 1 状态并保持)。

### 5.3.6 绘制顺序功能图的注意事项

下面是针对绘制顺序功能图时常见的错误提出的注意事项:

- (1) 两个步绝对不能直接相连,必须用一个转换将它们隔开。
- (2) 两个转换也不能直接相连,必须用一个步将它们隔开。
- (3) 顺序功能图中的初始步一般应对于系统等待启动的初始状态,这一步可能没有什么输出处于 ON 状态,因此在画顺序功能图时很容易遗漏这一步。初始步是必不可少的,一方面因为该步与它的相邻步相比,从整体上说输出变量的状态各不相同;另一方面如果没有该步,无法表示初始状态,系统也无法返回停止状态。

(4) 自动控制系统应能多次重复执行同一工艺过程,因此在顺序功能图中一般应有由步和有向连线组成的闭环,即在完成一次工艺过程的全部操作之后,应从最后一步返回初始步,系统停留在初始状态(单周期操作,见图 5-12),在连续循环工作方式时,将从最后一步返回下一工作周期开始运行的第一步(见图 5-17)。

(5) 如果选择有断电保持功能的存储器位(M)来代表顺序图中的各位,在交流电源断电瞬时的状态开始继续运行。如果用没有断电保持功能的存储器位代表各步,进入 RUN 工作方式时,它们处于 OFF 状态,必须 SM0.1 将初始步预置为活动步,否则因顺序功能图中没有活动步,系统将无法工作。如果系统有自动、手动两种工作方式,顺序功能图是用来描述自动工作过程的,这时还应在系统由手动工作方式进入自动工作方式时,用一个适当的信号(一般用 SM0.1)将初始步置为不活动步。

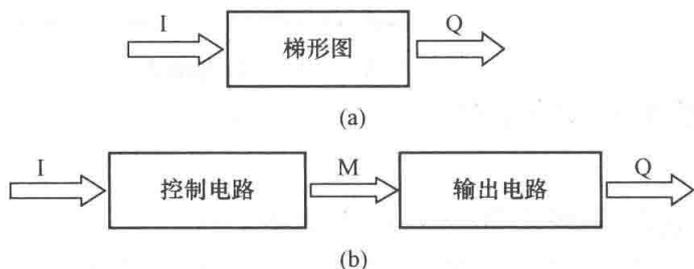


图 5-19 信号关系图

(a) 经验设计法; (b) 顺序控制设计法

### 5.3.7 顺序控制设计法的本质

经验设计法实际上是试图用输入信号 I 直接控制输出信号 Q [见图 5-19 (a)], 如果无法直接控制, 或为了实现记忆、联锁、互锁等功能, 只好被动地增加一些辅助元素和辅助触点。由于不同的系统的输出量 Q 与输入量 I 之间的关系各不相同, 以及它们对联锁、互锁的要求千变万化,

不可能找出一种简单用的设计方法。

顺序控制设计法则是用输入量 I 控制代表各步的编程元件(例如存储器位 M), 再用它们控制输出量 Q [见图 5-19 (b)]。步是根据输出量 Q 的状态划分的, M 与 Q 之间具有很简单的“与”逻辑关系, 输出电路的设计极为简单。任何复杂系统的代表步的 M 存储器位的控制电路, 其设计方法都是相同的, 并且很容易掌握, 所以顺序控制设计法具有简单、规

范、通用的优点。由于 M 是依次顺序变为 I 状态的，实际上已经基本上解决了经验设计法中的记忆、联锁等问题。

## 5.4 使用启、保、停电路的顺序控制梯形图编程方法

### 5.4.1 设计顺序控制梯形图的一些基本问题

在可编程控制器的实际应用中，有两种通用的编程方法，即使用启、保、停电路的编程方法和以转换为中心的编程方法，本章介绍的两种通用的编程方法很容易掌握，用它们可以迅速地、得心应手地设计出任意复杂的数字量控制系统的梯形图，它们的适用范围广，可以用于所有生产厂家的各种型号的 PLC。这两种方法都是根据顺序功能图来设计顺序控制梯形图的编程方法。

#### 1. 程序的基本结构

绝大多数自动控制系统除了自动工作模式外，还需要设置手动工作模式。在下列两种情况下需要工作在手动模式：

(1) 启动自动控制程序之前，系统必须处于要求的初始状态。如果系统的状态步满足启动自动程序的要求，需进入手动工作模式，用手动操作使系统进入规定的初始状态，然后再回到自动工作模式。一般在调试阶段使用手动工作模式。

(2) 顺序自动控制对硬件的要求很高，如果有硬件故障，例如某个限位开关有故障，不可能正确地完成整个自动控制过程。在这种情况下，为了使设备不至于停机，可以进入手动工作模式，对设备进行手动控制。

有自动、手动工作方式的控制系统的两种典型的程序结构如图 5-20 所示，公用程序用于处理自动模式和手动模式都需要执行的任务，以及处理两种模式的相互转换。

#### 2. 执行自动程序的初始状态

开始执行自动程序之前，要求系统处于规定的初始状态，如果开机时系统没有处于初始状态，则要进入手动工作方式，用手动操作使系统进入规定的初始状态后，再切换到自动工作方式。

系统满足规定的初始状态后，应将顺序功能图的初始步对应的存储器位置 1，使初始步变为活动步，为启动自动运行做好准备。同时还应将其余步对应的存储器位置 0。在 S7-200 PLC 中，常用特殊寄存器位 SM0.1 将初始步对应的存储器位置 1，也常用 SM0.1 将其余存储器位的状态置 0。

#### 3. 双线圈问题

在图 5-20 所示的自动程序和手动程序中，都需要控制 PLC 的输出 Q，因此同一个输出位的线圈可能会出现两次或多次，称为双线圈现象。

在跳步条件相反的两个程序段（如图 5-20 中的自动程序和手动程序）中，允许出现双线圈，即同一元件的线圈可以在自动程序和手动程序中分别出现一次。实际上 CPU 在每一

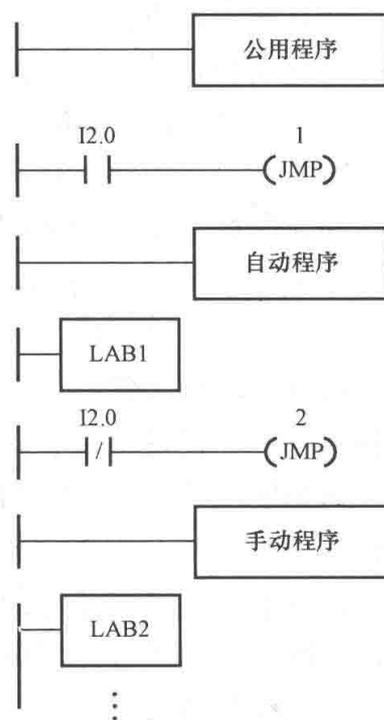


图 5-20 自动、手动程序

次循环中,只执行自动程序或只执行手动程序,不可能同时执行这两个程序。对于分别位于这两个程序中的两个相同的线圈,每次循环只处理其中一个,因此在本质上并没有违反不允许出现双线圈的规定。

#### 4. 设计顺序控制的基本方法

根据顺序功能图设计梯形图时,可以用存储器位 M 来代表步。为了便于将顺序功能图转换为梯形图,用代表各步的存储器位的地址作为步的代号,并用编程元件地址的逻辑代数表达式来标注转换条件,用编程元件的地址来标注各步的动作。

由图 5-19 可知,顺序控制程序分为控制电路和输出电路两部分。输出电路的输入量是代表步的编程元件 M,输出量是 PLC 的输出位 Q。它们之间的逻辑关系是极为简单的相等或相“或”的逻辑关系,输出电路是很容易设计的。

控制电路用 PLC 的输入量来控制代表步的编程元件,5.3.5 节中介绍的转换实现的基本规则是设计控制电路的基础。

某一步为活动步时,对应的存储器位 M 为 1 状态,某一转换实现时,该转换的后续步应变为活动步,前级步应变为不活动步。可以用一个串联电路来表示转换实现的这两个条件,该电路接通时,应将该转换所有的后续步对应的存储器位 M 置为 1 状态,将所有前级步对应的 M 复位为 0 状态。由 5.4.2 的分析可知,转换实现的两个条件对应的串联电路接通的时间只有一个扫描周期,因此应使用有记忆功能的电路或指令来控制代表步的存储器位。启、保、停电路和置位、复位电路都有记忆功能,本节和下一节将分别介绍使用启、保、停电路和置位复位电路的编程方法。

#### 5.4.2 单序列的编程方法

启、保、停电路只使用与触点和线圈有关的指令,任何一种 PLC 的指令系统都有这一类指令,因此这是一种通用的编程方法,可以用于任意型号的 PLC。

##### 1. 控制电路的编程方法

图 5-21 给出了图 5-12 中的液压动力滑台的进给运动示意图、顺序功能图和梯形图。在初始状态时动力滑台停在左边,限位开关 I0.3 为 1 状态。按下启动按钮 I0.0,动力滑台在各步中分别实现快进、工进、暂停和快退,最后返回初始位置和初始步后停止运动。

如果使用的 M 区被设置为没有断电保持功能,在开机时 CPU 调用 SM0.1 将初始步对应的 M0.0 置为 1 状态,开机时其余各步对应的存储器位被 CPU 自动复位为 0 状态。

设计启、保、停电路的关键是确定它的启动条件和停止条件。根据转换实现的基本规则,转换实现的条件是它的前级步为活动步,并且相应的转换条件满足。以控制 M0.2 的启、保、停电路为例,步 M0.2 的前级步为活动步时,M0.1 的动合触点闭合,它前面的转换条件满足时,I0.1 的动合触点闭合。两个条件同时满足时,M0.1 和 I0.1 的动合触点组成的串联电路接通。因此在启、保、停电路中,应将代表前级步的 M0.1 的动合触点和代表条件的 I0.1 的动合触点串联,作为控制 M0.2 的启动电路。

在快进步,M0.1 一直为 1 状态,其动合触点闭合。滑台碰到中限位开关时,I0.1 的动合触点闭合,由 M0.1 和 I0.1 的动合触点串联而成的 M0.2 的启动电路接通,使 M0.2 的线圈通电。在下一个扫描周期,M0.2 的动断触点断开,使 M0.1 的线圈断电,其动合触点断开,使 M0.2 的启动电路断开。由以上的分析可知,启、保、停电路的启动电路只能接通一个扫描周期,因此必须用有记忆功能的电路来控制代表步的存储器位。

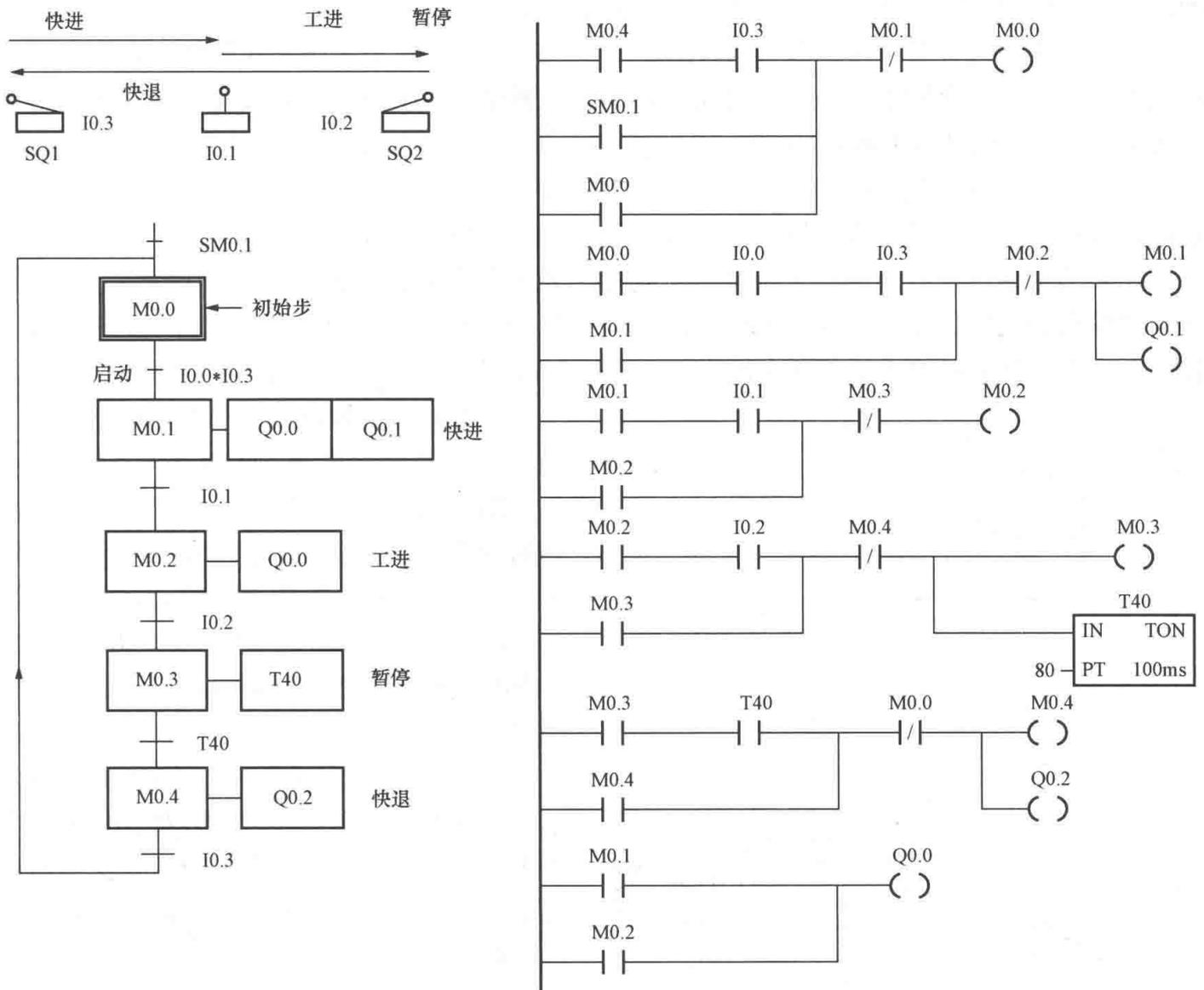


图 5-21 液压动力滑台的进给运动示意图、顺序功能图和梯形图

当 M0.2 和 I0.2 的动合触点均闭合时，步 M0.3 变为活动步，这时步 M0.2 应变为不活动步，因此可以将 M0.3=1 作为使存储器位 M0.2 变为 0 状态的条件，即将 M0.3 的动断触点与 M0.2 的线圈串联。上述的逻辑关系可以用逻辑代数式表示为

$$M0.2 = (M0.1 \cdot I0.1 + M0.2) \cdot \overline{M0.3}$$

在这个例子中，可以用 I0.2 的动断触点代替 M0.3 的动断触点。但是当转换条件由多个信号“与、或、非”逻辑运算组合而成时，需要将它的逻辑表达式求反，经过逻辑代数运算后再将对应的触点串并联电路作为启、保、停电路的停止电路，不如使用后续步对应的动断触点这样简单方便。

根据上述的编程方法和顺序功能图，很容易画出梯形图。以步 M0.1 为例，由顺序功能图可知，M0.0 是它的前级步，两者之间的转换条件为 I0.0 · I0.3，所以应将 M0.0、I0.0 和 I0.3 的动合触点串联，作为 0.1 的启动电路。启动电路并联了 M0.0 的自保持触点。后续步 M0.2 的动断触点与 M0.1 的线圈串联，M0.2 为 1 时 M0.1 的线圈“断电”，步 M0.1 变为不活动步。

### 2. 输出电路的编程方法

下面介绍设计梯形图的输出电路部分的方法。因为步是根据输出变量的状态变化来划分

的, 它们之间的关系极为简单, 可以分为两种情况来处理:

某一输出量仅在某一步中为 ON, 例如图 5-21 中的 Q0.1 就属于这种情况, 可以将它的线圈与对应步的存储器位 M0.1 的线圈并联。从顺序功能图还可以看出可以将定时器 T40 的线圈与 M0.3 的线圈并联, 将 Q0.2 的线圈和 M0.4 的线圈并联。有人也许觉得既然如此, 不如用这些输出位来代表该步, 例如用 Q0.1 代替 M0.1。这样可以节省一些编程元件, 但是存储器位来代替步具有概念清楚、编程规范、梯形图易于阅读和差错的优点。

如果某一输出在几步中都为 1 状态, 应将代表各有关步的存储器位的动合触点并联后, 驱动该输出的线圈。图 5-21 中 Q0.0 在 M0.1 和 M0.2 这两步中均应工作, 所以用 M0.1 和 M0.2 的动合触点组成的并联电路来驱动 Q0.0 的线圈。

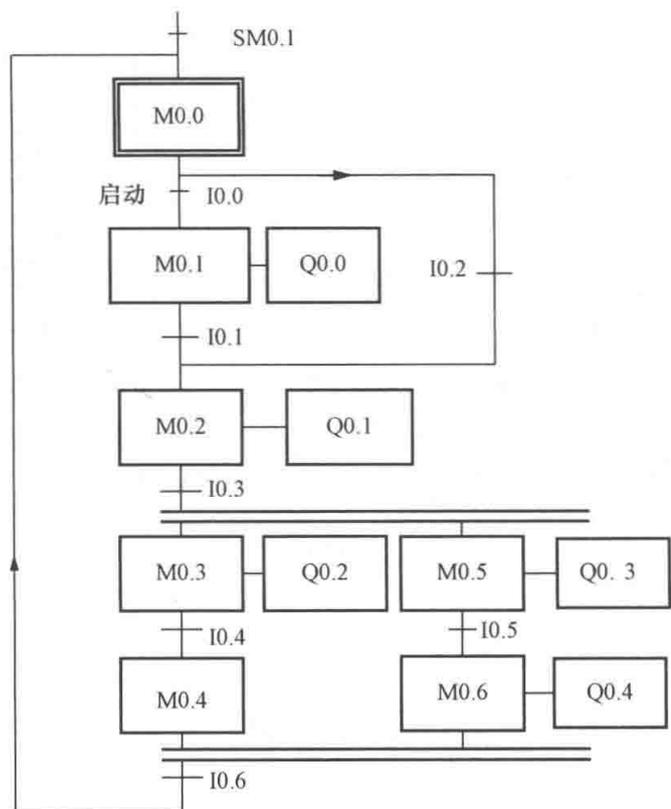


图 5-22 选择序列与并行序列

### 5.4.3 选择序列的编程方法

#### 1. 选择序列的分支的编程方法

图 5-22 中步 M0.0 之后有一个选择序列的分支, 设 M0.0 为活动步, 当它的后续步 M0.1 或 M0.2 变为活动步时, 它都应变为不活动步 (M0.0 变为 0 状态), 所以应将 M0.1 和 M0.2 的动断触点与 M0.0 的线圈串联。

如果某一步的后面有一个由 N 条分支组成的选择序列, 该步可能转换到不同的 N 步去, 则应将这 N 个后续步对应存储器位的动断触点与该步的线圈串联, 作为结束该步的条件。

#### 2. 选择序列的合并的编程方法

在图 5-22 中, 步 M0.2 之前有一个选择序列的合并, 当步 M0.1 为活动步 (M0.1 为 1), 并且转换条件 I0.1 满足, 或步 M0.0 为活动步并且转换条件 I0.2 满足, 步 M0.2 都应变为活动步, 即代表该步的存储器位 M0.2 的启动条件应为  $M0.1 \cdot I0.1 + M0.0 \cdot I0.2$ , 对应的启动电路由两条并联支路组成, 每条支路分别由 M0.1、I0.1 或 M0.0、I0.2 的动合触点串联而成。

一般来说, 对于选择序列的合并, 如果某一步之前有 N 个转换, 即有 N 条分支进入该步, 则代表该步的存储器位的启动电路由 N 条支路并联而成, 各支路由某一前级对应的存储器位的动合触点与相应转换条件对应的触点或电路串联而成。

### 5.4.4 并行序列的编程方法

#### 1. 并行序列的分支的编程方法

图 5-22 中的步 M0.2 之后有一个并行序列的分支, 当步 M0.2 是活动步并且转换条件 I0.3 满足时, 步 M0.3 与步 M0.5 应同时变为活动步, 这是用 M0.2 和 I0.3 的动合触点组成的串联电路分别作为 M0.3 和 M0.5 的启动电路来实现的; 与此同时, 步 M0.2 应变为不活动步。步 M0.3 和 M0.5 是同时变为活动步的, 只需将 M0.3 或 M0.5 的动断触点与 M0.2

的线圈串联即可。

### 2. 并行序列的合并的编程方法

步 M0.0 之前有一个并行序列的合并, 该转换实现的条件是所有的前级步 (即步 M0.4 和 M0.6) 都是活动步和转换条件 I0.6 满足。由此可知, 应将 M0.4、M0.6 和 I0.6 的动合触点串联, 作为控制 M0.0 的启、保、停电路的启动电路。M0.4 和 M0.6 的线圈都串联了 M0.0 的动合触点, 使步 M0.4 和步 M0.6 在转换实现时同时变为不活动步。

任何复杂的顺序功能图都是由单序列、选择序列和并行序列组成的, 掌握了单序列的编程方法和选择序列、并行序列的分支、合并的编程方法, 就不难迅速地设计出任意复杂的顺序功能图描述的数字化控制系统的梯形图。如图 5-23 所示。

### 5.4.5 仅有两步的闭环的处理

如果在顺序功能图中有仅由两步组成的小闭环 [见图 5-24 (a)], 用启、保、停电路设计的梯形图不能正常工作。例如 M0.2 和 I0.0 均为 1 时, M0.3 的启动电路接通, 但是这时与 M0.3 的线圈串联的 M0.2 的动断触点却是断开的, 所以 M0.3 的线圈不能“通电”。出现上述问题的根本原因在于步 M0.2 既是步 M0.3 的前级步, 又是它的后续步。将图 5-24 (b) 中的 M0.2 的动断触点改为转换条件 I0.2 的动断触点, 就可以解决这个问题。

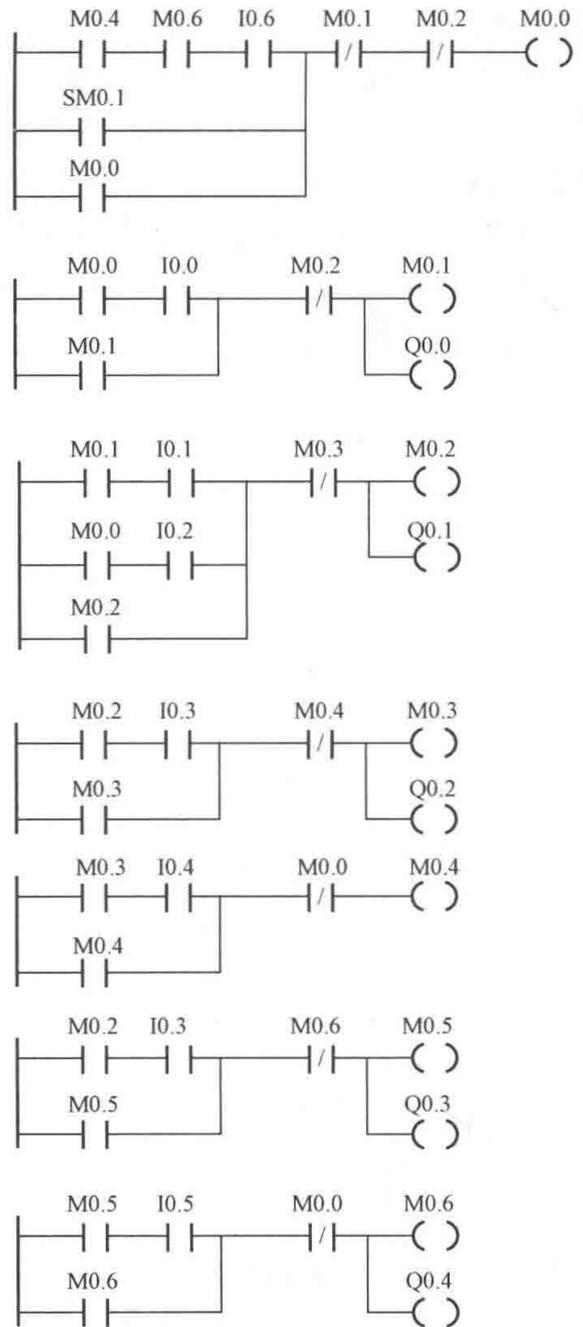
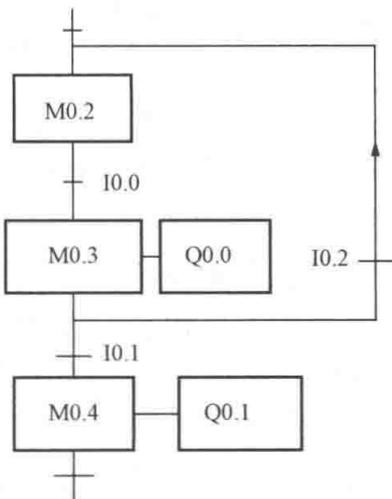
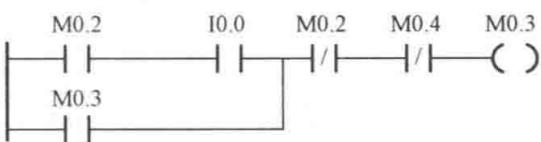


图 5-23 梯形图程序



(a)



(b)

图 5-24 仅有两步的闭环的处理

### 5.4.6 应用举例

如图 5-25 所示的物料混合装置用来将粉末状的固体物料 (粉料) 和液体物料 (液料) 按一定的比例混合在一起, 经过一定时间的搅拌后便得到成品。粉料和液料都用电子秤来计量。

初始状态时粉料秤秤斗、液料秤秤斗和搅拌器都是空的, 它们底部的排料阀关闭; 液料仓的放料阀关闭, 粉料仓下部的螺旋输送机的电动机和搅拌机的电动机停转; 用 Q0.0 ~ Q0.4 分别与 M1、YV1、YV2、M2、YV3 对应, 初始时均为 0 状态。

PLC 开机后用 SM0.1 将初始步对应的 M0.0 置为 1 状态, 将其余各步对应的存储器复位为 0 状态,

并将 MW10 和 MW12 中的计数预置值分别送给减计数器 C0 和 C1。

按下启动按钮 I0.0, Q0.0 变为 1 状态, 螺旋输送机的电动机旋转, 粉料进入粉料秤的秤斗; 同时 Q0.1 变为 1 状态, 液料仓的放料阀打开, 液料进入液料秤的秤斗。电子秤的光电码盘输出与秤斗内物料重量成正比的脉冲信号。减计数器 C0 和 C1 分别对粉料和液料秤产生的脉冲计数。粉料脉冲计数值减至 0 时, 其动合触点闭合, 粉料秤的秤斗内的物料等于预置值。Q0.0 变为 0 状态, 螺旋输送机的电动机停机。液料脉冲计数值减至 0 时, 其动合触点闭合, 液料秤的秤斗内的物料等于预置值。Q0.1 变为 0 状态, 关闭液料仓的放料阀。

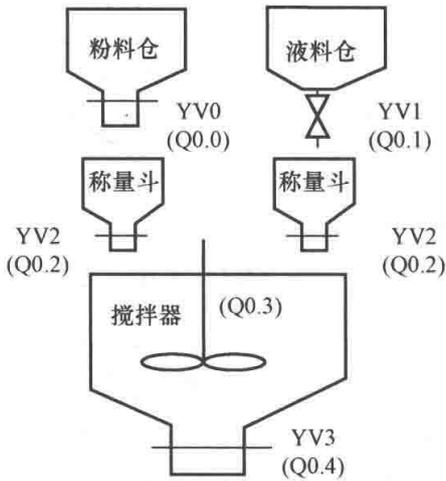


图 5-25 物料混合控制系统示意图

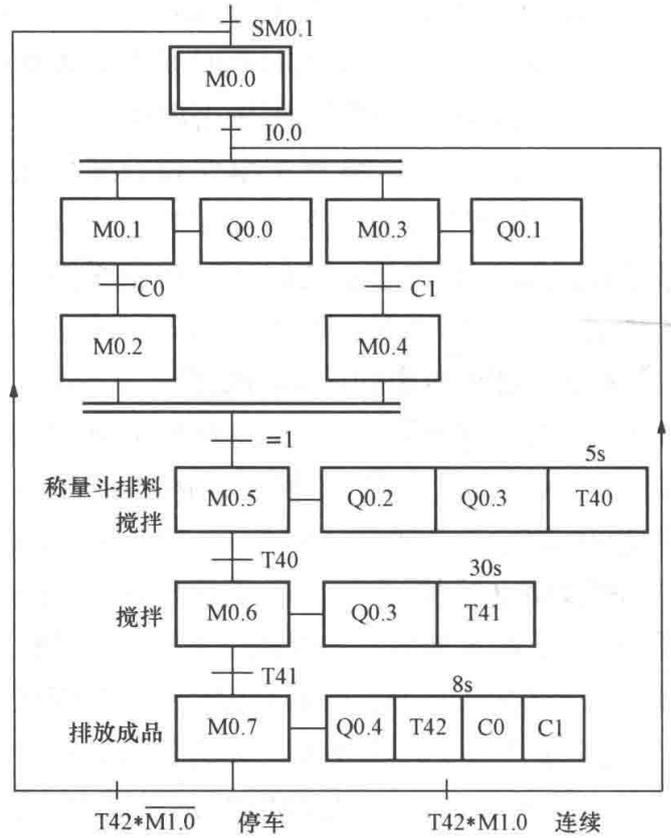


图 5-26 物料混合控制系统的功能顺序图

计数器的当前值非 0 时, 计数器的输出位为 1, 反之为 0。粉料称量结束后, C0 的动合触点闭合, 转换条件 C0 满足, 粉料秤从步 M0.1 转换到等待步 M0.2。同样地, 液料称量结束后, 液料秤从步 M0.3 转换到等待步 M0.4。步 M0.2 和 M0.4 后面的转换条件“=1”表示转换条件为二进制常数 1, 即转换条件总是满足的。因此在两个秤的称量都结束后, M0.2 和 M0.4 同时为活动步, 系统将“无条件地”转换到步 M0.5, Q0.2 变为 1 状态, 打开电子秤下部的排料门, 两个电子秤开始排料, 排料过程用定时器 T40 定时。同时 Q0.3 变为 1 状态, 搅拌机开始搅拌。T40 的定时时间到时排料结束, 转换到步 M0.6, 搅拌机继续搅拌。T41 的定时时间到时停止搅拌, 转换到步 M0.7, 将预设值送给 C0、C1, 为下一次称量做好准备, 同时 Q0.4 变为 1 状态, 搅拌器底部的排料门打开, 经过 T2 的定时时间后, 关闭排料门, 一个工作循环结束。

本系统要求在按启动按钮 I0.0 后, 能连续不停地工作下去。按了停止按钮 I0.1 后, 并不立即停止运行, 要等到当前工艺周期的全部工作完成, 成品排放结束后, 再从步 M0.7 返回到初始步 M0.0。

图 5-27 中的第一个启、保、停电路用来实现上述要求, 按下启动按钮 I0.0, M1.0 变

为 1 状态，系统处于连续工作模式。在顺序功能图最下面一步执行完后，T42 的动合触点闭合，转换条件  $T42 \cdot M1.0$  满足，将从步 M0.7 转换到步 M0.1 和 M0.3，开始下一个周期的工作。在工作循环中的任意一步（步 M0.1~M0.7）为活动步时按下停止按钮 I0.1，“连续”标志位 M1.0 变为 0 状态，但是它不会马上起作用，要等到最后一步 M0.7 的工作结束，T42 的动合触点闭合，转换条件  $T42 \cdot \overline{M1.0}$  满足，才会从步 M0.7 转换到初始步 M0.0，系统停止运行。

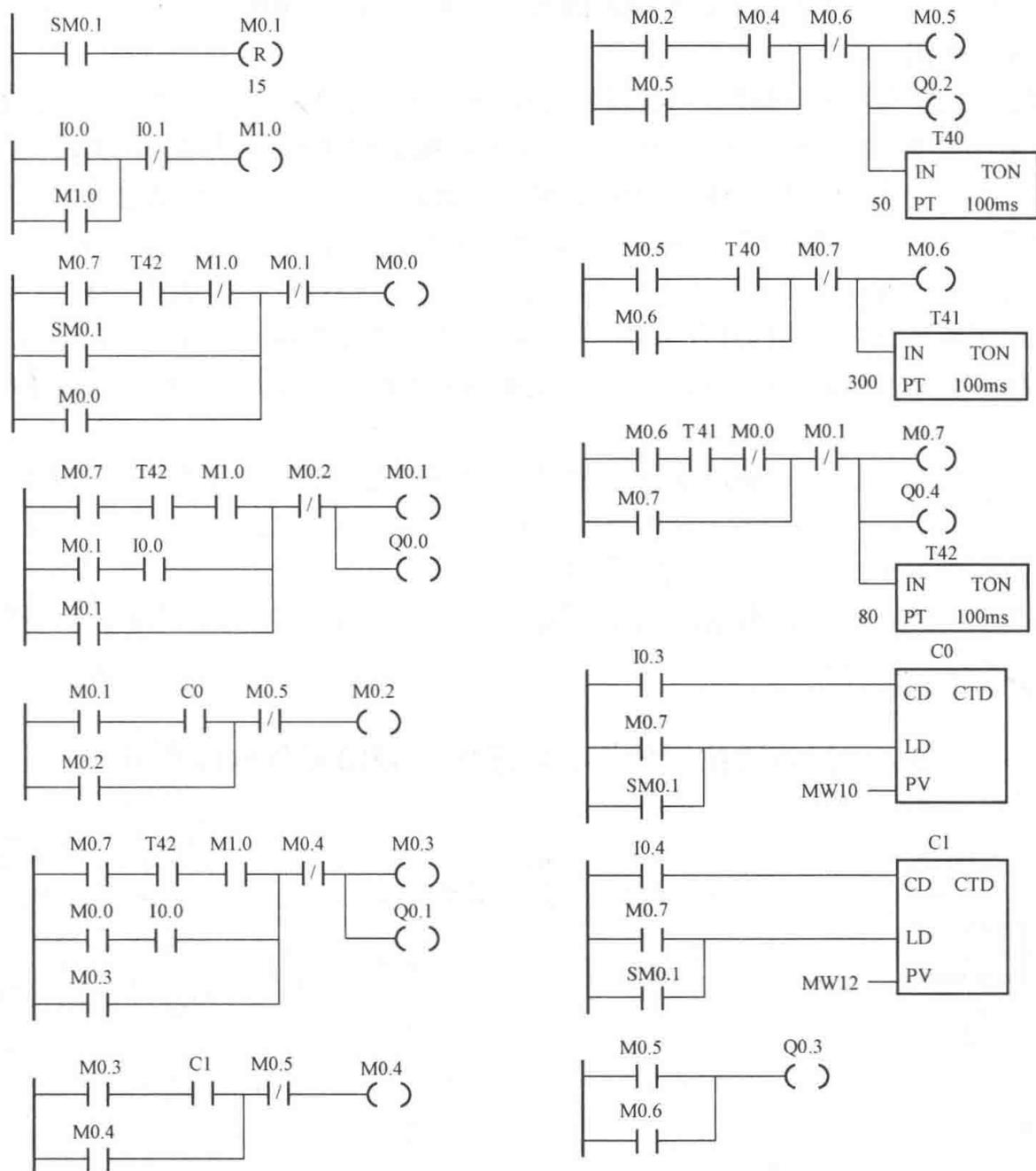


图 5-27 物料混合控制系统的梯形图

步 M0.7 之后有一个选择序列的分支，当它的后续步 M0.0、M0.1 和 M0.3 变为活动步时，它都应变为不活动步。但是 M0.1 和 M0.3 是同时变为 1 状态的，所以只需要将 M0.0 和 M0.1 的动断触点或 M0.0 和 M0.3 的动断触点与 M0.7 的线圈串联。

步 M0.1 和步 M0.3 之前有一个选择序列的合并，当步 M0.0 为活动步并且转换条件 I0.0 满足，或步 M0.7 为活动步并且转换条件满足，步 M0.1 和步 M0.3 都应变为活动步，即代表这两步的存储器位 M0.1 和步 M0.3 的启动条件应为  $M0.0 \cdot I0.0 + M0.7 \cdot T42 \cdot$

M1.0, 对应的启动电路由两条并联支路组成, 每条支路分别由 M0.0、I0.0 或 M0.7、T42、M1.0 的动合触点串联而成 (见图 5-27)。

图 5-26 中步 M0.0 之后有一个并行序列的分支, 当 M0.0 是活动步, 并且转换条件 I0.0 满足; 或者 M0.7 是活动步, 并且转换条件  $T4 \cdot M0.1$  满足, 步 M0.1 与步 M0.3 都应同时变为活动步。M0.1 和 M0.3 的启动电路完全相同, 保证了这两步同时变为活动步。

步 M0.1 与步 M0.3 是同时变为活动步的, 它们的动断触点同时断开, 因此 M0.0 的线圈只需要串联 M0.1 或 M0.3 的动断触点即可。当然也可以同时串联 M0.1 与 M0.3 的动断触点, 但是要多用一条指令。

步 M0.5 之前有一个并行序列的合并, 由步 M0.2 和步 M0.4 转换到 M0.5 的条件是所有的前级步 (即步 M0.2 和 M0.4) 都是活动步和转换条件 (=1) 满足。因为转换条件总是满足的, 所以只需将 M0.2 和 M0.4 的动合触点串联, 作为 M0.5 的启动电路就可以了。可以将转换条件 “=1” 理解为启动电路中一条看不见的短接线。

为了进一步提高生产效率, 两个电子秤的称量过程与搅拌过程可以同时进行, 它们的工作过程可以用有 2 条单序列的并行系列来描述, 在称量和搅拌都完成后排放成品, 然后开始搅拌和将秤斗中的原料放入搅拌机中。放料结束后关闭秤斗底部的卸料门, 两个秤的秤斗又开始进料和称量的过程。

实际的物料混合系统 (例如混凝土搅拌系统和橡胶工业中的密炼机配料控制系统) 要复杂得多, 输入/输出量要多得多。这里为了突出重点, 为使读者尽快地掌握顺序控制梯形图的编程方法, 对实际的系统作了大量的简化。

本例中使用的是 PLC 的普通计数器, 其计数频率较低, 在实际系统中一般用高速计数器来对编码器发出的脉冲计数。

## 5.5 使用置位复位指令的顺序控制梯形图编程方法

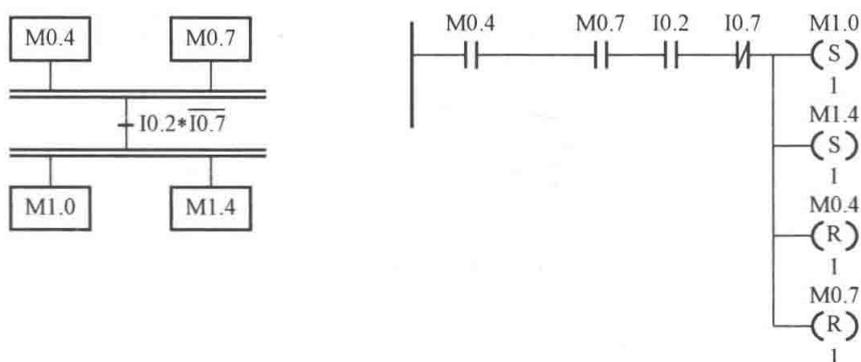


图 5-28 以转换为中心的编程方法

使用置位复位指令的顺序梯形图编程方法又称为以转换为中心的编程方法。图 5-28 给出了顺序功能图与梯形图的对应关系。实现图中的转换需要同时满足以下两个条件:

(1) 该转换所有的前级步都是活动步, 即 M0.4 和 M0.7 均为 1 状态, M0.4 和

M0.7 的动合触点同时闭合。

(2) 转换条件  $I0.2 \cdot I0.7$  满足, 即 I0.2 的动合触点和 I0.7 的动断触点组成的电路接通。

在梯形图中, 可用 M0.4、M0.7 和 I0.2 的动合触点与 I0.7 的动断触点组成的串联电路来表示上述两个条件同时满足。这种串联电路实际上就是使用启、保、停电路的编程方法中的启动电路。根据上一节的分析, 该电路接通的时间只有一个扫描周期。因此需要用有记忆功能的电路来保持它引起的变化, 本节用置位、复位指令来实现记忆功能。

该电路接通时，应执行以下两个操作：

(1) 应将高转换所有的后续步变为活动步，即将代表后续步的存储器位变为 1 状态，并使它保持 1 状态。这一要求刚好可以用有保护功能的置位指令 (S 指令) 来完成。

(2) 应将该转换所有的前级步变为活动步，即将代表前级步的存储器位变为 0 状态，并使它们保持 0 状态。这一要求刚好可以用复位指令 (R 指令) 来完成。

这种编程方法与转换实现的基本规则之间有着严格的对应关系，在任何情况下，代表步的存储器位的控制电路都可以用这个统一的规则来设计，每一个转换对应一个图 5-28 所示的控制置位和复位的电路块，有多少个转换就有多少个这样的电路块。这种编程方法特别有规律，在设计复杂的顺序功能图的梯形图时既容易掌握，又不容易出错。用它编制复杂的顺序功能图的梯形图时，更能显示出它的优越性。

相对而言，使用启、保、停电路的编程方法的规则较为复杂，选择序列的分支与合并、并行序列的分支与合并都有单独的规则需要记忆。

### 5.5.1 单序列的编程方法

某工作台旋转运动的示意图如图 5-29 所示。工作台在初始状态时停在限位开关 I0.1 处，I0.1 为 1 状态。按下启动按钮 I0.0，工作台正转，旋转到限位开关 I0.2 处改为反转，返回限位开关 I0.1 处时又改为正转，旋转到限位开关 I0.3 处又改为反转，回到起始点时停止运动。图 5-29 同时给出了系统的顺序功能图和用以转换为中心的编程方法设计的梯形图。

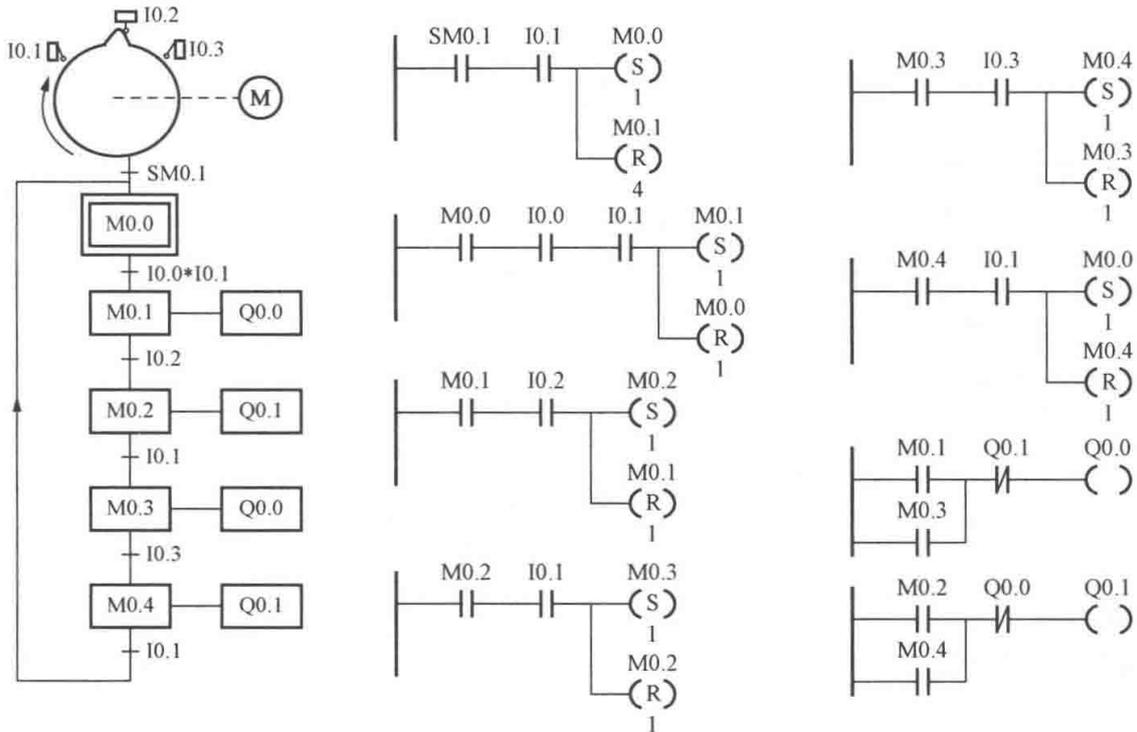


图 5-29 工作台旋转运动的顺序功能图和梯形图

以转换条件 I0.2 对应的电路为例，该转换的前级步为 M0.1，后续步为 M0.2，所以用 M0.1 和 I0.2 的动合触点组成的串联电路来控制对后续步 M0.2 的置位和对前级步 M0.1 的复位。每一个转换对应一个这样的“标准”电路，有多少个转换就有多少个这样的电路。设计时应注意不要遗漏掉某一个转换对应的电路。

使用这种编程方法时，不能将输出位 Q 的线圈与置位指令和复位指令并联，这是因为



### 5.5.3 并行序列的编程方法

图 5-30 中步 M0.2 之后有一个并行序列的分支，当 M0.2 是活动步，并且转换条件 I0.3 满足时，步 M0.3 与步 M0.5 应同时变为活动步，这是用 M0.2 和 I0.3 的动合触点组成的串联电路使 M0.3 和 M0.5 同时置位来实现的；与此同时，步 M0.2 应变为不活动步，这是用复位指令来实现的。

I0.6 对应的转换之前有一个并行序列的合并，该转换实现的条件是所有的前级步（即步 M0.4 和 M0.6）都是活动步和转换条件 I0.6 满足。由此可知，应将 M0.4、M0.6 和 I0.6 的动合触点串联，作为使后续步 M0.0 置位和使前级步 M0.4、M0.6 复位的条件。

### 5.5.4 应用举例

以专用钻床控的控制系统为例，图 5-17 给出了专用钻床控制系统的顺序功能图，图 5-31 是用以转换为中心的方法编制的梯形图。

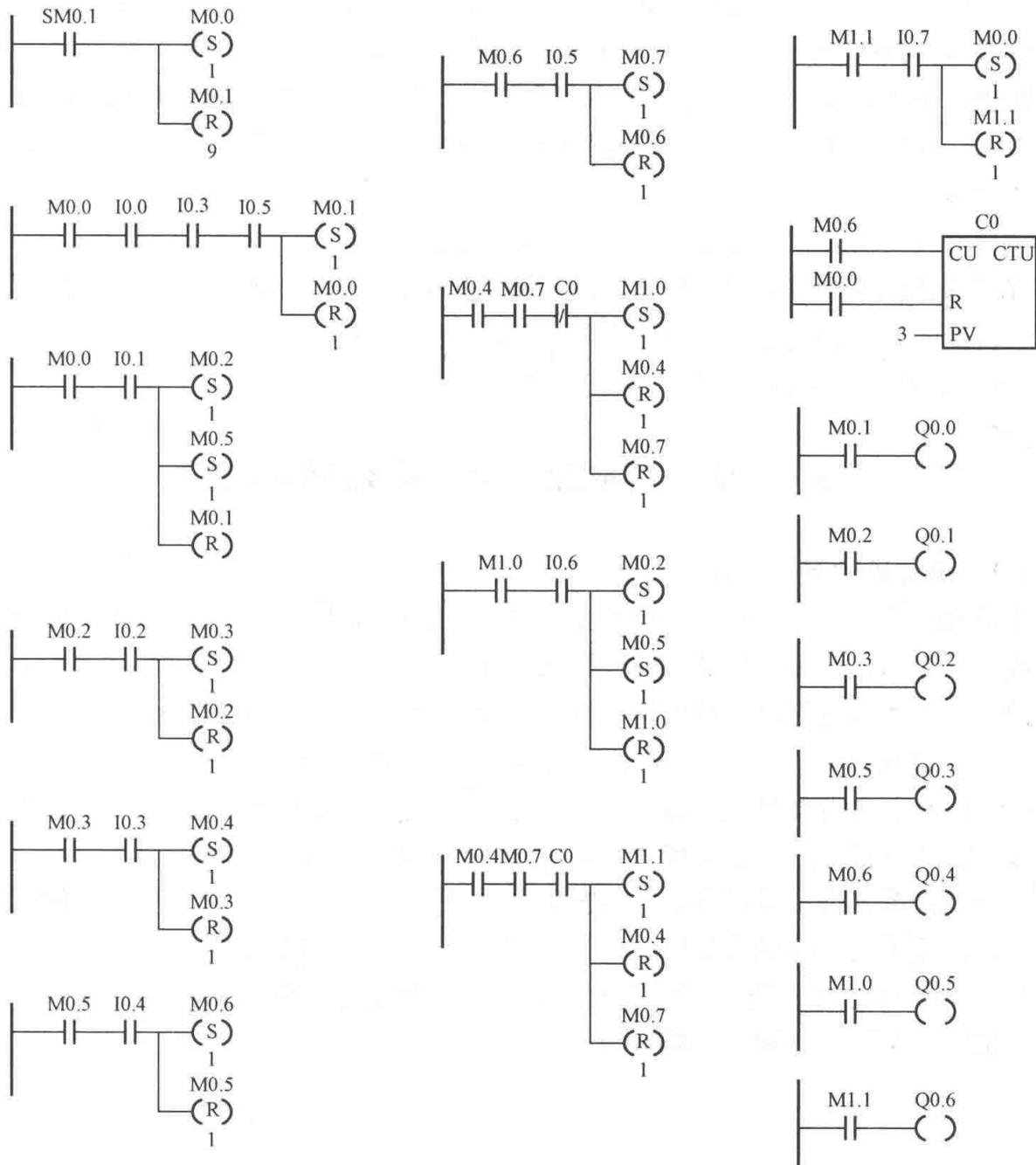


图 5-31 组合钻床控制系统的梯形图

图 5-17 中分别由 M0.2~M0.4 和 M0.5~M0.7 组成的两个单序列是并行工作的,设计梯形图时应保证这两个序列同时开始工作和同时结束,即两个序列的第一步 M0.2 和 M0.5 应同时变为活动步,两个序列的最后一步 M0.4 和 M0.7 应同时变为不活动步。

并行序列的分支的处理是很简单的,在图 5-17 中,当步 M0.1 是活动步,并且转换条件 I0.1 为 ON 时,步 M0.2 和 M0.5 同时变为活动步,两个序列开始同时工作。在梯形图中,用 M0.1 和 I0.1 的动合触点组成的串联电路来控制对 M0.2 和 M0.5 的同时置位和对前级步 M0.1 的复位。

另一种情况是当步 M1.0 为活动步,并且转换条件 I0.6 为 ON 时,步 M0.2 和 M0.5 也应同时变为活动步,两个序列开始同时工作。在梯形图中,用 M1.0 和 I0.6 的动合触点组成的串联电路来控制对 M0.2 和 M0.5 的同时置位和对前级步 M1.0 的复位。

图 5-17 中并行序列合并处的转换有两个前级步 M0.4 和 M0.7,根据转换实现的基本规则,当它们均为活动步并且转换条件 C0 满足时,将实现并行序列的合并。未钻完 3 对孔时,加计数器 C0 的当前值非 3,其动断触点闭合,转换条件  $\overline{C0}$  满足时,将转换到步 M1.0。在梯形图中,用 M0.4、M0.7 的动合触点和 C0 的动断触点组成的串联电路使 M1.0 置位,后续步 M1.0 变为活动步;同时用 R 指令将 M0.4 和 M0.7 复位,使前级步 M0.4 和 M0.7 变为不活动步。

钻完 3 个孔时,C0 的当前值为 3,其动合触点闭合,转换条件 C0 满足将转换到步 M1.1。在梯形图中,用 M0.4、M0.7 的动合触点和 C0 的动合触点组成的串联电路使 M1.1 置位,后续步 M1.1 变为不活动步;同时用 R 指令将 M0.4、M0.7 复位,前级步 M0.4、M0.7 变为不活动步(见图 5-31)。

## 5.6 具有多种工作方式的系统的编程方法

### 5.6.1 机械手控系统简介

为了满足生产的需要,很多设备要求设置多种工作方式,如手动方式和自动方式,后者包括连续、单周期、单步、自动返回初始状态几种工作方式。手动程序比较简单,一般用经验法设计,复杂的自动程序一般根据系统的顺序功能图用顺序控制法设计。

如图 5-32 所示,某机械手用来将工件从 A 点到 B 点,控制面板如图 5-33 所示,图 5-34 是 PLC 的外部接线图。输出 Q0.1 为“1”时工件被夹紧,为“0”时工件被松开。

工作方式选择开关的 4 个位置分别对应于 4 种工作方式,操作面板左上部的方式选择开关为手动调整开关。为了保证在紧急情况下(包括 PLC 发生故障时)能可靠地切断 PLC 的负载电源,设置了交流接触器 KM(见图 5-34)。在 PLC 开始运行时按下“负载电源”按钮,使 KM 线圈得电并自锁,KM 的主触点接通,给外部负载提供交流电源,出现紧急情况时用“紧急停车”按钮断开负载电源。

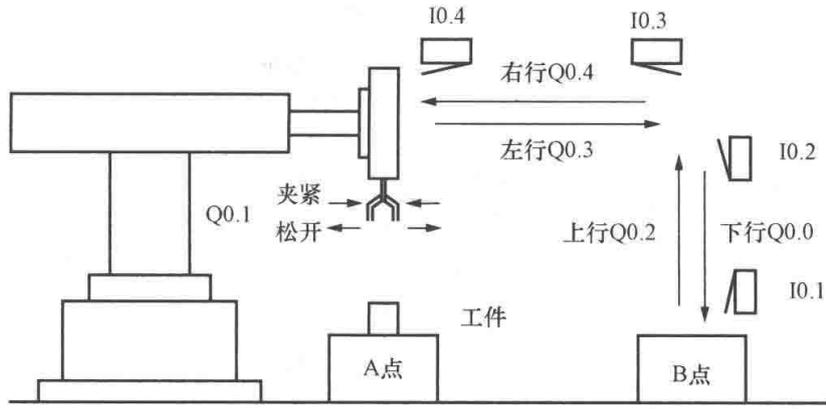


图 5-32 机械手示意图

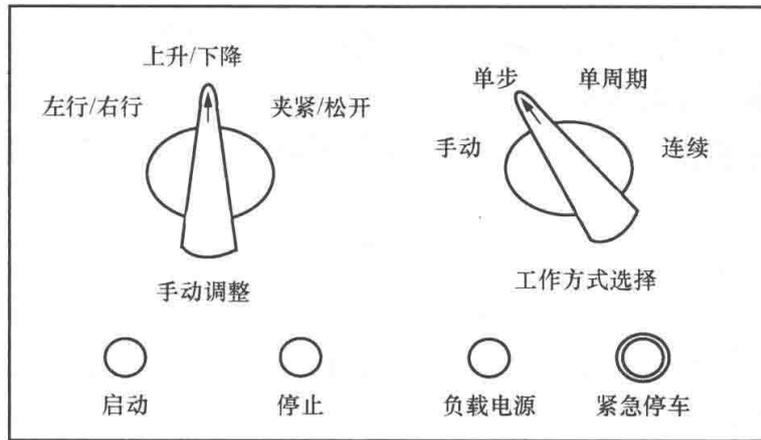


图 5-33 控制面板

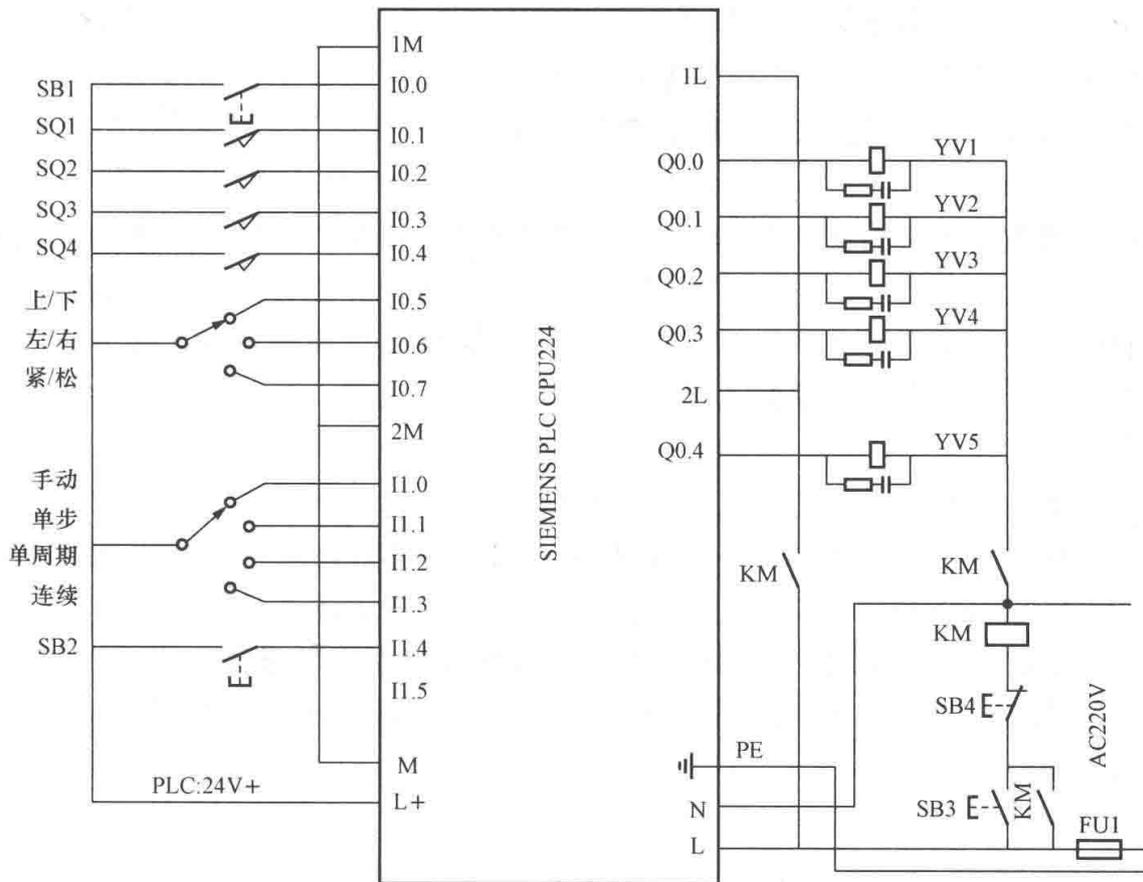


图 5-34 PLC 的外部接线图

表 5-1

符号表

符号	地址	符号	地址	符号	地址
启动按钮 SB1	I0.0	松紧调整	I0.7	转换允许	M0.6
下限位	I0.1	手动方式	I1.0	连续标志	M0.7
上限位	I0.2	单步方式	I1.1	下降阀	Q0.0
右限位	I0.3	单周期方式	I1.2	夹紧阀	Q0.1
左限位	I0.4	连续方式	I1.3	上升阀	Q0.2
上下调整	I0.5	停止按钮 SB2	I1.4	左行阀	Q0.3
左右调整	I0.6			右行阀	Q0.4

系统设有手动、单周期、单步、连续 4 种工作方式,机械手在最上面和最左边且松开时称为系统处于原点状态(或称为初始状态)。在公用程序中,左限位开关 I0.4、上限位开关 I0.2 的动合触点和表示机械手松开的 Q0.1 的动断触点的串联电路接通时,“原点条件”存储器位 M0.5 变为 ON。

如果选择的是单周期工作方式,按下启动按钮 I0.0 后,从初始步 M0.0 开始,机械手按顺序功能图(见图 5-38)的规定完成一个周期的工作后,返回并停留在初始步。如果选择连续工作方式,在初始状态按下启动按钮,并不马上停止工作,完成最后一个周期的工作后,系统才返回并停留在初始步。在单步工作方式,从初始步开始,按一下启动按钮,系统转换到下一步,完成该步的任务后,自动停止工作并停在该步,再按一下启动按钮,又往前走一步。单步工作方式常用于系统的调试。

进入单周期、连续和单步工作方式之前,系统应处于原点状态,在原点状态,顺序功能图中的初始步 M0.0 为 ON,为进入单周期、连续和单步工作方式做好准备。

### 5.6.2 使用启、保、停电路的编程方法

#### 1. 程序的总体结构

控制系统的主程序结构如图 5-35 所示,用跳转指令 JMP 来实现各种工作方式的切换。由外部接线图 5-34 可知,工作方式选择开关是单刀 4 掷开关,同时只能选择一种工作方式。例如选择开关位于手动方式时,I1.0 为“1”,执行手动控制程序。在手动方式时由切换开关来实现上下、左右、松紧的手动调整和控制。

#### 2. 公用程序

机械手处于最上面和最左边的位置、夹紧装置松开时,系统处于规定的初始条件,称为“原点条件”,此时左限位开关 I0.4、上限位开关 I0.2 的动合触点和表示夹紧装置松开的 Q0.1 的动断触点组成的串联电路接通,存储器位 M0.5 为 1 状态。

图 5-36 中的公用程序用于自动程序和手动程序相互切换的处理。当系统处于手动工作方式时,I1.0 为 1 状态,如果此时满足原点条件,顺序功能图中的初始步对应的 M0.0 被置位,反之则被复位。

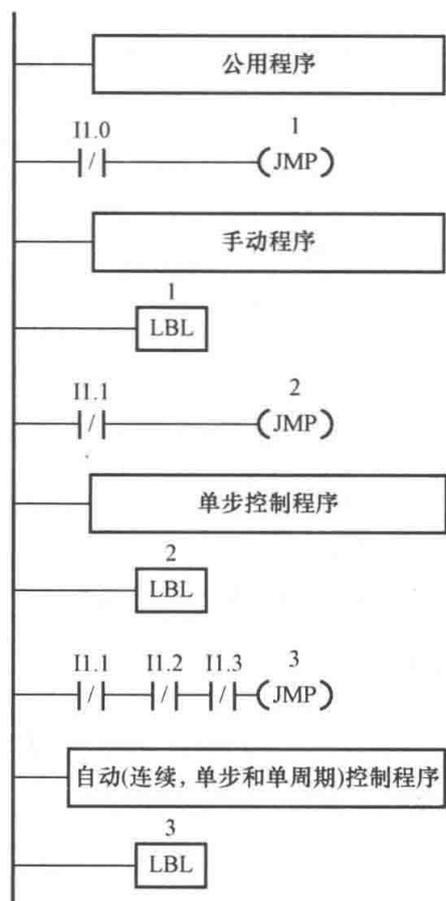


图 5-35 程序结构

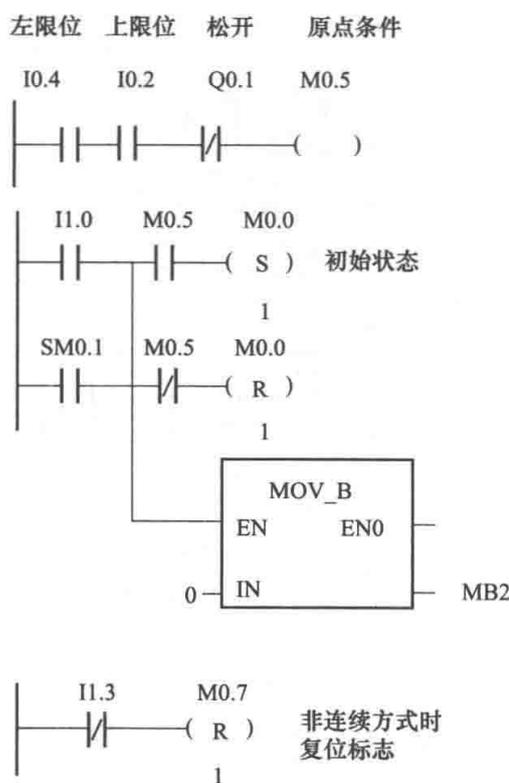


图 5-36 公用程序

在 CPU 刚进入 RUN 模式的第一个扫描周期执行图 5-36 中的程序时，如果原点条件满足，M0.5 为 1 状态，顺序功能图中的初始步对应的 M0.0 由初始化脉冲信号 SM0.1 置位，为进入单步、单周期和连续工作方式做好准备。

当系统处于手动工作方式时，I1.0 的动合触点闭合，用 MOVE 指令将顺序功能图中除初始步以外的各步对应的存储器位 (M2.0~M2.7) 复位，否则当系统从自动工作方式切换到手动工作方式，然后又返回自动工作方式时，可能会出现同时有两个活动步的异常情况，引起错误的动作。在非连续方式，将表示连续工作状态的标志 M0.7 复位。

### 3. 手动程序

图 5-37 是手动程序，手动操作时用 I0.5~I0.7 对应的 3 个开关控制机械手的升、降、左行、右行和夹紧、松开。为了保证系统的安全运行，在手动程序中设置了一些必要的联锁，例如上升与下降之间、左行与右行之间的互锁用来防止功能相反的两个输出同时为 ON。上限位开关 I0.2 的动合触点与控制左、右行的 Q0.4 和 Q0.3 的线圈串联，机械手升到最高位置才能左右移动，以防止机械手在较低的位置运行时与别的物体碰撞。

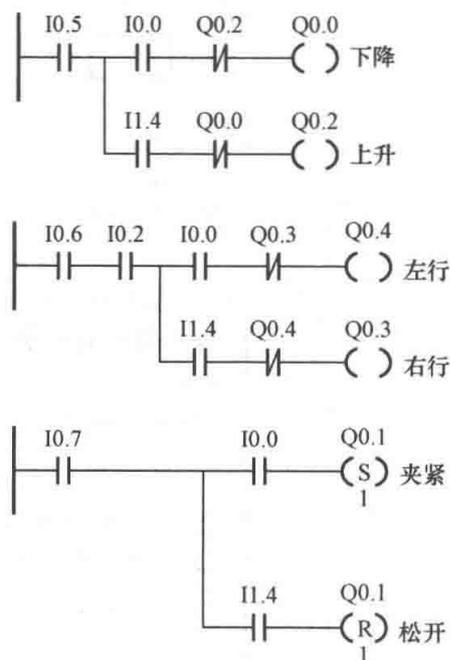


图 5-37 手动程序

4. 单周期、连续和单步程序

图 5-38 是处理单周期、连续和单步工作方式的顺序功能图 and 梯形图。

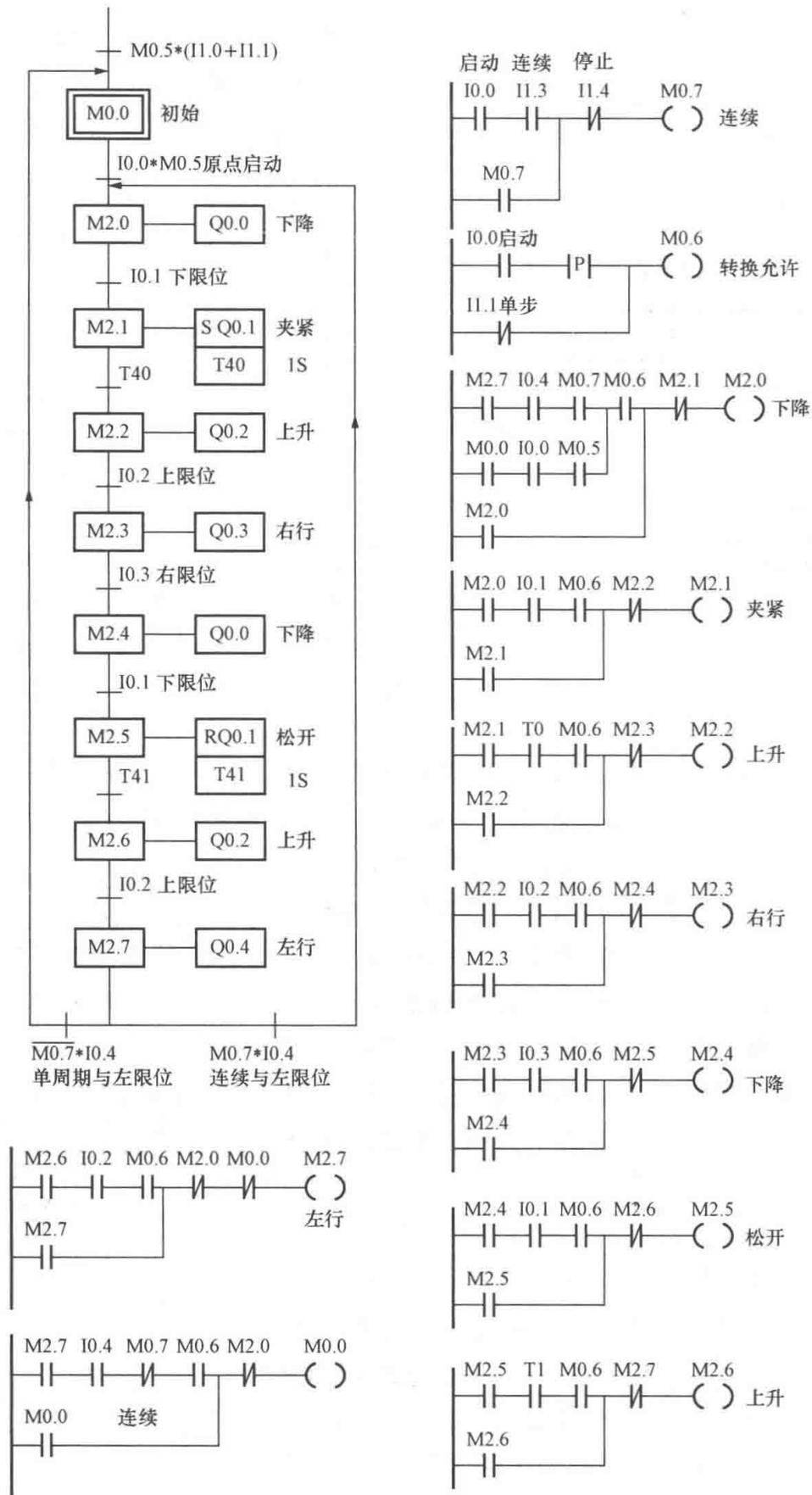


图 5-38 顺序功能图与梯形图

M0.0 和 M2.0~M2.7 用典型的启、保、停电路来控制。

单周期、连续和单步这3种工作方式主要是用“连续”标志 M0.7 和“转换允许”标志 M0.6 来区分的。

(1) 单步与非单步的区分。M0.6 的动合触点接在每一个控制代表步的存储器位的启动电路中，它们断开时禁止步的活动状态的转换。如果系统处于单步工作方式，I1.1 为 1 状态，它的动断触点断开，“转换允许”存储器位 M0.6 在一般情况下为 0 状态，不允许步与步之间的转换。当某一步的工作结束后，转换条件满足，如果没有按启动按钮 I0.0，M0.6 处于 0 状态，启、保、停电路的启动电路处于断开状态，不会转换到下一步。一直要等到按下启动按钮 I0.0，M0.6 在 I0.0 的上升沿 ON 一个扫描周期，M0.6 的动合触点接通，系统才会转换到下一步。

系统工作在连续、单周期（非单步）工作方式时，I1.1 的动断触点接通，使 M0.6 为“1”状态，串联在各启、保、停电路的启动电路中的 M0.6 的动合触点接通，允许步与步之间的正常转换。

(2) 单周期与连续的区分。在连续工作方式，I1.3 为 1 状态。在初始状态按下启动按钮 I0.0，M2.0 变为 1 状态，机械手下降。与此同时，控制连续工作的 M0.7 的线圈“通电”并自保持。

当机械手在步 M2.7 返回最左边时，I0.4 为 1 状态，因为“连续”标志位 M0.7 为 1 状态，转换条件  $M0.7 * I0.4$  满足，系统将返回步 M2.0，反复连续地工作下去。

按下停止按钮 I1.4 后，M0.7 变为 0 状态，但是系统不会立即停止工作，在完成当前工作周期的全部操作后，在步 M2.7 返回最左边，左限位开关 I0.4 为 1 状态，转换条件  $M0.7 * I0.4$  满足，系统才返回并停留在初始步。

在单周期工作方式，M0.7 一直处于 0 状态。当机械手在最后一步 M2.7 返回最左边时，左限位开关 I0.4 为 1 状态，转换条件  $M0.7 * I0.4$  满足，系统返回并停留在初始步。按一次启动按钮，系统只工作一个周期。

(3) 单周期工作过程。在单周期工作方式，I1.1（单步）的动断触点闭合，M0.6 的线圈“通电”，允许转换。在初始步时按下启动按钮 I0.0 在 M2.0 的启动电路中，M0.0、I0.0、M0.5（原点条件）和 M0.6 的动合触点均接通，使 M2.0 的线圈“通电”，系统进入下降步，Q0.0 的线圈“通电”，机械手下降；碰到下限位开关 I0.1 时，转换到夹紧步 M2.1，Q0.1 被置位，夹紧电磁阀线圈通电并保持。同时接通延时定时器 T40 开始定时，定时时间到时，工件被夹紧，1s 后转换条件 T40 满足，转换到步 M2.2。以后系统将这样一步一步地工作下去，直到步 M2.7，机械手左行返回原点位置，左限位开关 I0.4 变为 1 状态，因为连续工作标志 M0.7 为 0 状态，将返回初始步 M0.0，机械手停止运动。

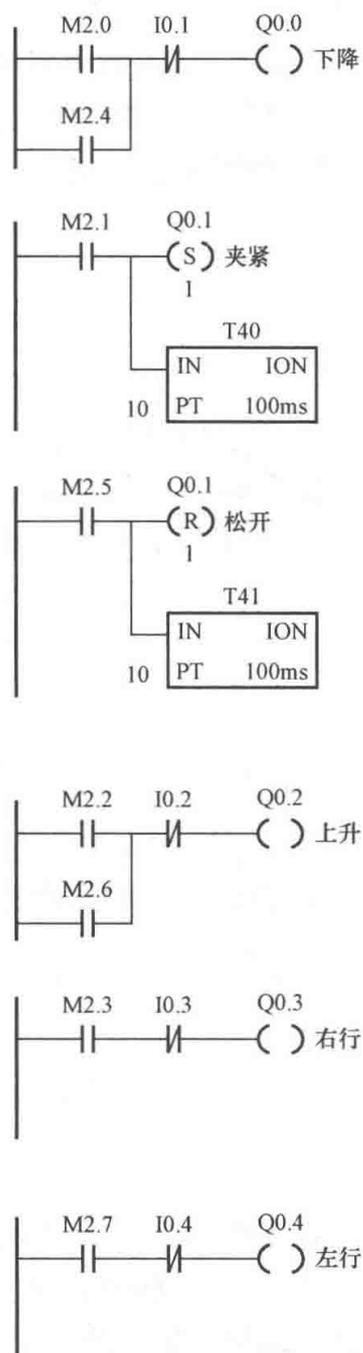


图 5-39 输出电路

图 5-38 转换控制 M0.0 的启、保、停电路如果放在控制 M2.0 的启、保、停电路之前,在单步工作方式步 M2.7 为活动步时按启动按钮 I0.0,返回步 M0.0 后, M2.0 的启动条件满足,将马上进入步 M2.0。在单步工作方式,这样连续跳两步是不允许的。将控制 M2.0 的启、保、停电路放在控制 M0.0 的启、保、停电路之前和 M0.6 的线圈之后可以解决这一问题。在图 5-39 中,控制 M0.6 (转换允许)的是启动按钮 I0.0 的上升沿检测信号,在步 M2.7 按启动按钮, M0.6 仅 ON 一个扫描周期,它使 M0.0 的线圈通电后,下一扫描周期处理控制 M2.0 的启、保、停电路时, M0.6 已变为 0 状态,所以不会使 M2.0 变为 1 状态,要等到下一次按启动按钮时, M2.0 才会变为 1 状态。

(4) 输出电路。输出电路(见图 5-40)是自动程序的一部分,输出电路中 I0.1~I0.4 的动断触点是单步工作方式设置的。以下降为例,当小车碰到限位开关 I0.1 后,与下降步对应的存储器位 M2.0 或 M2.4 不会马上变为 OFF,如果 Q0.0 的线圈不与 I0.1 的动断触点串联,机械手不能停在下限位开关 I0.1 处,还会继续下降,对于某些设备,可能造成事故。

### 5.6.3 使用置位复位指令的编程方法

与使用启、保、停电路的编程方法相比,顺序功能图、公用程序、手动程序和自动程序中的输出电路完全相同。请学习者应用置位复位指令的编程方法自己编写程序。

## 习 题

1. 设计电动机正反转控制系统梯形图程序。

控制要求:按下启动按钮 I0.0,电动机正转 3s,停 2s,反转 3s,停 2s,循环 3 次。

2. 设计通电和断电延时的梯形图程序,见图 5-40。

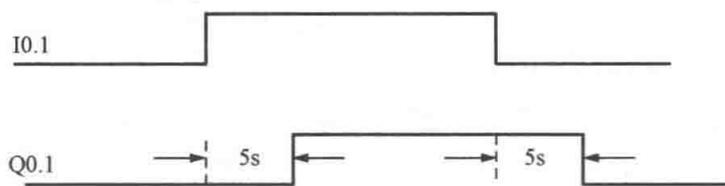


图 5-40 题 2 的图

3. 设计振荡电路的梯形图程序,见图 5-42。

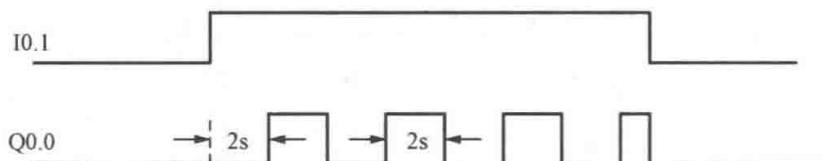


图 5-41 题 3 的图

4. 设计电机正反转控制的梯形图及 PLC 外部接线图。

要求:正反转启动信号 I0.1、I0.2,停止信号 I0.3,输出信号 Q0.2、Q0.3。具有软件互锁和机械互锁功能。

5. 有一 3 台皮带运输机传输系统,分别用电动机 M1、M2、M3 带动,控制要求如下:

按下启动按钮，先启动最末一台皮带机 M3，经 5s 后再依次启动其他皮带机。正常运行时，M3、M2、M1 均工作。按下停止按钮时，先停止最前一台皮带机 M1，待料送完毕后，再依次停止其他皮带机。

写出 I/O 分配表；画出梯形图。

6. 使用传送机，将大、小球分类后分别传送，控制系统的工艺动作要求如下：

左上为原点，按启动按钮 SB1 后，其动作顺序为：下降 → 吸收（延时 1s）上升 → 右行 → 下降 → 释放（延时 1s） → 上升 → 左行。

其中：LS1：左限位；LS3：上限位；LS4：小球右限位；LS5：大球右限位；LS2：大球下限位 LS0：小球下限位。

注意：机械壁下降时，吸住大球，则下限位 LS2 接通，然后将大球放到大球容器中。

若吸住小球，则下限位 LS0 接通，然后将小球放到小球容器中。

(1) 设计 I/O。

(2) 画梯形图。

(3) 写出指令系统。

7. 某系统有两种工作方式，手动和自动。现场的输入设备有：6 个行程开关（SQ1～SQ6）和 2 个按钮（SB1～SB2）仅供自动程序使用，6 个按钮（SB3～SB8）仅供手动程序使用，4 个行程开关（SQ7～SQ10）为手动、自动两程序共用。现有 S7-200 CPU224 型 PLC，其输入点 14 个（I0.0～I1.5），是否可以使用？若可以，试画出相应的外部输入硬件接线图。

8. 设计喷泉电路。

要求：喷泉有 A、B、C 三组喷头。启动后，A 组先喷 5s，后 B、C 同时喷，5s 后 B 停，再 5s C 停，而 A、B 又喷，再 2s，C 也喷，持续 5s 后全部停，再 3s 重复上述过程。

说明：A（Q0.0），B（Q0.1），C（Q0.2），启动信号 I0.0。

9. 设计一工作台自动往复控制程序。

要求：正反转启动信号 I0.0、I0.1，停车信号 I0.2，左右限位开关 I0.3、I0.4，输出信号 Q0.0、Q0.1。具有电气互锁和机械互锁功能。

10. 设计钻床主轴多次进给运动的 PLC 控制程序。

要求：该机床进给由液压驱动。电磁阀 DT1 得电主轴前进，失电后退。同时，还用电磁阀 DT2 控制前进及后退速度，得电快速，失电慢速。其工作过程如图 5-42 所示。

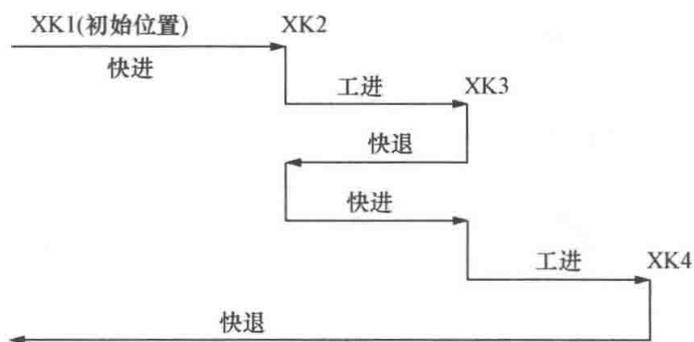


图 5-42 题 10 的图

11. 设计十字路口交通灯控制系统的梯形图程序。

要求：按下启动按钮，东西方向红灯亮，同时，南北方向绿灯亮 7s，随后南北方向绿灯闪烁 3s，之后南北方向黄灯亮 2s；紧接着南北方向红灯亮，东西方向绿灯亮 7s，随后东西方向绿灯闪烁 3s，之后东西方向黄灯亮 2s，实现一个循环。如此循环，实现交通灯的控制。按下停止按钮，交通灯立即停止工作。

12. 设计物料传送系统控制的梯形图程序。

要求：图 5-43 所示为两组带机组成的原料运输自动化系统，该自动化系统启动顺序为：盛料斗 D 中无料，先启动带机 C，5s 后，再启动带机 B，经过 7s 后再打开电磁阀 YV，该自动化系统停机的顺序恰好与启动顺序相反，即先关闭电磁阀 TV，7s 后带机 B 停止，5s 后带机 C 停止。试完成梯形图设计。

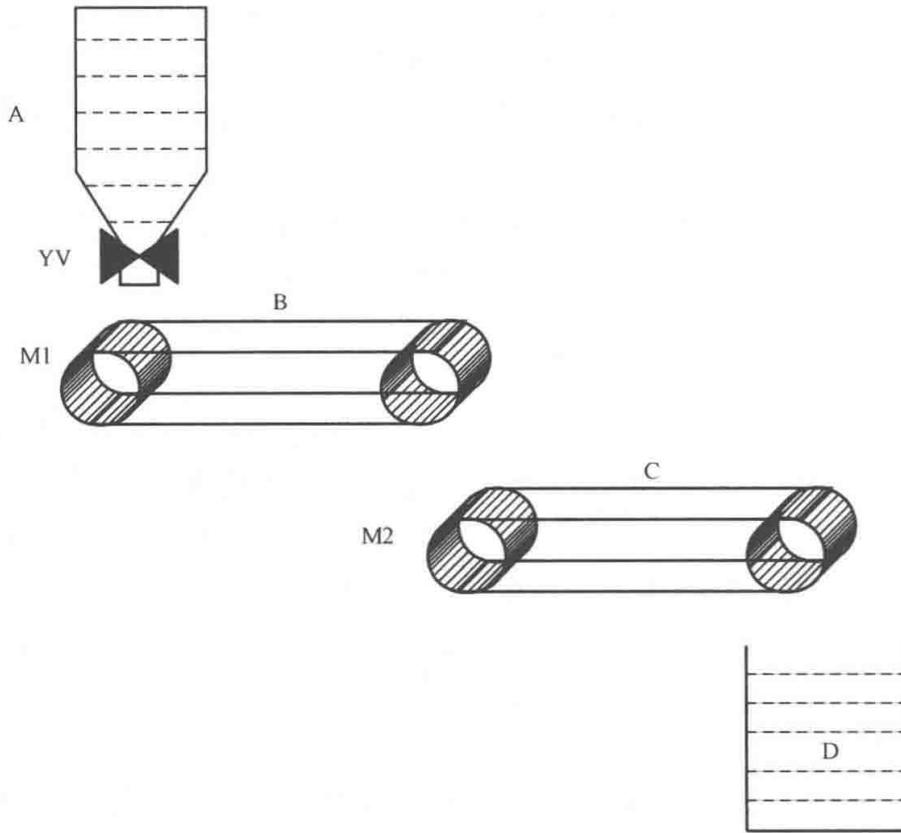


图 5-43 题 12 的图

## 第 6 章 S7-200 PLC 在模拟量闭环控制中的应用

### 6.1 模拟量闭环控制的基本概念

#### 6.1.1 模拟量闭环控制系统的组成

##### 1. 模拟量单闭环控制系统的组成

典型的 PLC 模拟量单闭环控制系统框图如图 6-1 所示，图中虚线中的部分是用 PLC 实现的。

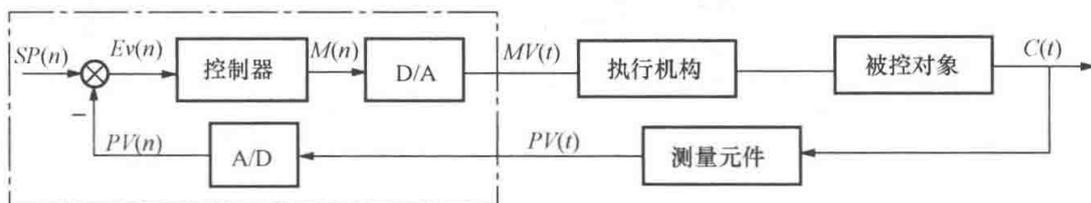


图 6-1 PLC 模拟量闭环控制系统框图

在模拟量闭环控制系统中，被控量  $C(t)$ （如压力、温度、流量、转速等）是连续变化的模拟量，大多数执行机构（如晶闸管调速装置、电动调节阀和变频器等）要求 PLC 输出模拟信号  $MV(t)$ ，而 PLC 的 CPU 只能处理数字量。 $C(t)$  首先被测量元件（传感器）和变送器转换为标准的直流电流信号或直流电压信号  $PV(t)$ ，例如  $4\sim 20\text{mA}$ ， $1\sim 5\text{V}$ ， $0\sim 10\text{V}$ ，PLC 用 A/D 转换器将它们转换为数字量  $PV(n)$ 。

模拟量与数字量之间的相互转换和 PID 程序的执行都是周期性的操作，其间隔时间称为采样周期  $T_s$ 。各数字量括号中的  $n$  表示该变量是第  $n$  次采样计算时的数字量。

图 6-1 中的  $SP(n)$  是给定值， $PV(n)$  为 A/D 转换后的反馈量，误差  $E_v(n) = SP(n) - PV(n)$ 。

D/A 转换器将 PID 控制器输出的数字量  $M(n)$  转换为模拟量（直流电压或直流电流） $MV(t)$ ，再去控制执行机构。

例如在加热炉温度闭环控制系统中，用热电偶检测炉温，温度变送器将热电偶输出的微弱的电压信号转换为标准量程的电流或电压，然后送给模拟量输入模块，经 A/D 转换后得到与温度成比例的数字量，CPU 将它与温度设定值比较，并按某种控制规律（例如 PID 控制算法）对误差值进行运算，将运算结果（数字量）送给模拟量输出模块，经 D/A 转换后变为电流信号或电压信号，用来控制电动调节阀的开度，通过它控制加热用的天然气的流量，实现对温度的闭环控制。 $C(t)$  为系统的输出量，即被控量，例如加热炉中的温度。

模拟量控制系统分为恒值控制系统和随动系统。恒值控制系统的给定值由操作人员提供，一般很少变化，例如温度控制系统、转速控制系统等。随动系统的输入量是不断变化的随机变量，例如高射炮的瞄准控制系统和电动调节阀的开度控制系统就是典型的随动系统。闭环负反馈控制可以使控制系统的反馈量  $PV(n)$  等于或跟随给定值  $SP(n)$ 。以炉温控制系统为例，假设输出的温度值  $C(t)$  低于给定的温度值，反馈量  $PV(n)$  小于给定值

$SP(n)$ , 误差  $E_V(n)$  为正, 控制器的输出量  $MV(t)$  将增大, 使执行机构 (电动调节阀) 的开度增大, 进入加热炉的天然气流量增加, 加热炉的温度升高, 最终使实际温度接近或等于给定值。

天然气压力的波动、工件进入加热炉, 这些因素称为扰动量, 它们会破坏炉温的稳定。闭环控制可以有效地抑制闭环中各种扰动的影响, 使被控量趋近于给定值。

闭环控制系统的结构简单, 容易实现自动控制, 因此在各个领域得到了广泛的应用。

2. 复杂的控制系统的结构

(1) 级联控制。级联控制又称串级控制, 两个控制器串行连接, 第一个控制器 (主控制器) 的输出  $SP_2$  是伺服控制器的设定值 (见图 6-2)。

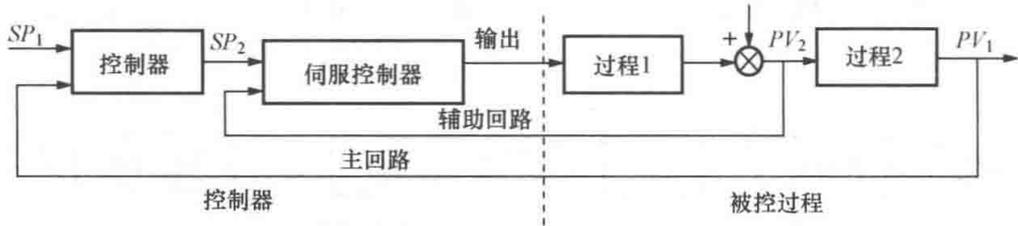


图 6-2 级联控制器

级联控制的控制性能可以用附加的过程变量来改进。为此在系统中适当的地方引出辅助过程变量  $PV_2$ , 将它与主控制器的输出  $SP_2$  比较, 得到的误差值作为伺服控制器的输入量。

主控制器的给定值为  $SP_1$ , 反馈值为  $PV_1$ , 它调整  $SP_2$ , 以便尽可能快地使过程变量  $PV_1$  在没有超调的情况下达到设定值。

以电动调节阀为执行机构的控制系统就是一个级联控制系统。电动调节阀有一个位置随动系统, 图 6-2 中的  $PV_2$  是阀门的开度, 即阀芯的位置。通过辅助回路的调节作用, 阀门的开度与主控制器的输出量  $SP_2$  成正比。

(2) 混合控制器。混合控制器总的设定值  $SP$  按一定的比例分配给各控制组件, 各混合系数之和应为 1, 例如图 6-3 中的  $FAC1 + \dots + FAC4 = 1$ 。

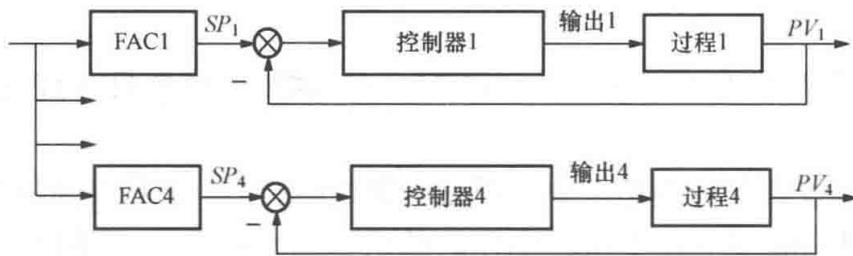


图 6-3 混合控制器

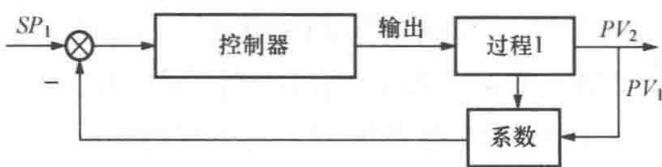


图 6-4 单闭环比例控制器

(3) 单闭环比例控制器。对于某些控制系统来说, 控制两个过程变量之间的比例比控制它们的绝对数值更重要, 例如控制两台需要同步的设备的速度, 可以使用图 6-4 中的单闭环比例控制器。

(4) 多闭环比例控制器。多闭环比例控制使两个过程变量  $PV_1$  和  $PV_2$  之比保持为常数。为此用第一个控制闭环的过程量  $PV_1$  来计算第 2 个控制闭环的设定值。在过程变量  $PV_1$  动

态变化的过程中也能保证  $PV_1$  与  $PV_2$  之间保持设定的比例。如图 6-5 所示。

(5) 二级控制器。二级控制器 (Two-Step Controller) 只能提供开关量的两个相反的输出状态, 例如开关的接通和断开。典型控制为通过继电器输出的脉冲宽度调制信号对加热系统的控制。

(6) 三级控制器。三级控制器只能提供开关量的 3 个输出状态。应区分脉冲宽度调制 (例如加热和冷却, 加热—关机—冷却)

和使用积分式执行机构的步进控制 (例如右—停止—左) 之间的区别。

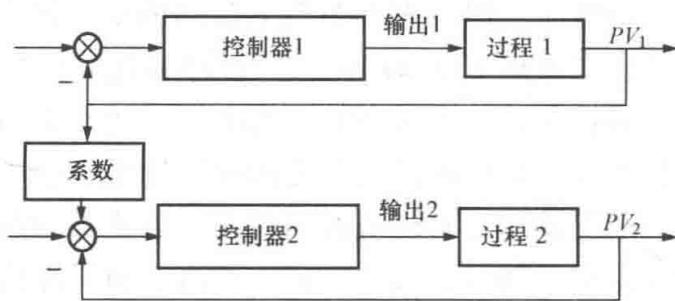


图 6-5 多闭环比例控制器

### 6.1.2 闭环控制的主要性能指标

由于给定输入信号或扰动输入信号的变化, 系统的输入量达到稳态值之前的过程称为过渡过程或动态过程。系统的动态性能常用阶跃响应 (阶跃输入时输出量的变化) 的参数来描述。阶跃输入信号在  $t=0$  之前为 0,  $t>0$  时为某一恒定值。

输出量第一次达到稳态值的时间  $t_r$  称为上升时间, 上升时间反映了系统在响应初期的快速性。

系统进入并停留在稳态值  $c(\infty)$  上下  $\pm 5\%$  (或  $2\%$ ) 的误差带内的时间  $t_s$  称为调节时间, 到达调节时间表示过渡过程已基本结束。

设动态过程中输出量的最大值为  $c_{\max}(t)$ , 如果它大于输出量的稳态值  $c(\infty)$ , 则超调量为

$$\sigma\% = \frac{c_{\max}(t) - c(\infty)}{c(\infty)} \times 100\%$$

超调量反映了系统的相对稳定性, 它越小动态稳定性越好, 一般希望超调量小于  $10\%$ 。

系统的稳态误差是进入稳态后的期望值与实际值之差, 它反映了系统的稳态精度。

### 6.1.3 闭环控制反馈极性的确定

闭环控制必须保证系统是负反馈 (误差 = 给定值 - 反馈值), 而不是正反馈 (误差 = 给定值 + 反馈值)。如果系统接成了正反馈, 将会失控, 被控量会往单一方向增大或减小, 给系统的安全带来极大的威胁。

闭环控制系统的反馈极性与很多因素有关, 例如因为接线改变了变送器输出电流或输出电压的极性, 在 PID 控制程序中改变了误差的计算公式, 改变了某些直线位移传感器或转角位移传感器的安装方向, 都会改变反馈的极性。

可以用下述的方法来判断反馈的极性: 在调试时断开 D/A 转换器与执行机构之间的连线, 在开环状态下运行 PID 控制程序。如果控制器中有积分环节, 因为反馈被断开了, 不能消除误差, 这时 D/A 转换器的输出电压会向一个方向变化。这时如果假设接上执行机构, 能减小误差, 则为负反馈, 反之为正反馈。

以温度控制系统为例, 假设开环运行时给定值大于反馈值, 若 D/A 转换器的输出值不断增大, 如果形成闭环, 将使电动调节阀的开度增大, 闭环后温度反馈值将会增大, 使误差减小, 由此可以判定系统是负反馈。

### 6.1.4 变送器的选择

变送器用于将传感器提供的电量或非电量转换为标准的直流电流或直流电压信号, 例如

DC0~10V 和 4~20mA。变送器分为电流输出型和电压输出型，电压输出型变送器具有恒压源的性质，PLC 模拟量输入模块的电压输入端的输入阻抗很高，例如  $100k\Omega\sim 10M\Omega$ 。如果变送器距离 PLC 较远，通过线路间的分布电容和分布电感产生的干扰信号电流在模块的输入阻抗上将产生较高的干扰电压。例如  $1\mu A$  干扰电流在  $10M\Omega$  输入阻抗上将产生 10V 的干扰电压信号，所以远程传送模拟量电压信号时抗干扰能力很差。

电流输出具有恒流源的性质，恒流源的内阻很大。PLC 的模拟量输入模块输入电流时，输入阻抗较低（例如  $250\Omega$ ）。线路上的干扰信号在模块的输入阻抗上产生的干扰电压很低，所以模拟量电流信号适于远程传送。

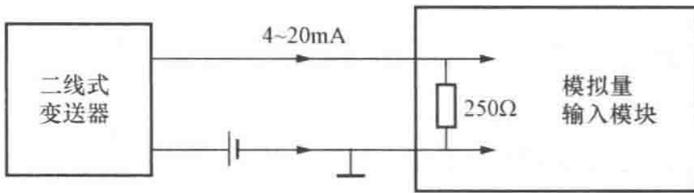


图 6-6 二线式变送器

电流传送比电压传送的传送距离远得多，S7-200 的模拟量输入模块使用屏蔽电缆信号线时允许的最大距离为 200m。

变送器分为二线式和三线式，三线式变送器有 3 根线：电源线、信号线和公共线。二线式变送器只有两根外部接线，见图 6-6，它们既是电源线，也是信号线，输出 4~20mA 的信号电流，DC24V 电源串接在回路中。通过调试，在被检测信号满量程时输出电流为 20mA。二线式变送器的接线少，信号可以远传，在工业中得到了广泛的应用。

### 6.1.5 PLC 的模拟量输入接口

它的作用是把现场连续变化的模拟量标准信号转换成适合可编程序控制器内部处理的由若干位二进制数字表示的信号。模拟量输入接口接受标准模拟信号，无论是电压信号还是电流信号均可。这里标准信号是指符合国际标准的通用交互用电压电流信号值，如 4~20mA 的直流电流信号、1~10V 的直流电压信号等。工业现场中模拟量信号的变化范围一般是不标准的，在送入模拟量接口时一般都需经变送处理才能使用。图 6-7 是模拟量输入接口的内部电路框图。

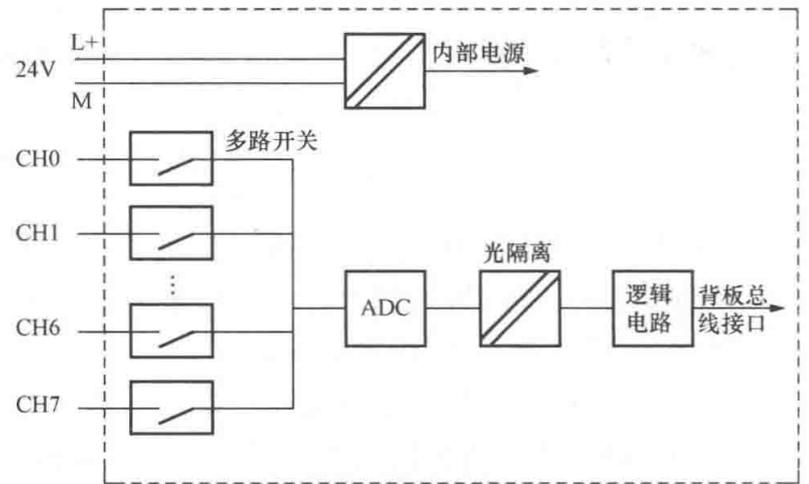


图 6-7 模拟量输入接口的内部电路框图

模拟量信号输入后一般经运算放大器放大后进行 A/D 转换，再经光电耦合后为可编程控制器提供一定位数的数字量信号。

### 6.1.6 PLC 的模拟量输出接口

它的作用是将可编程控制器运算处理后的若干位数字量信号转换为响应的模拟量信号输出，以满足生产过程现场连续控制信号的需求。模拟量输出接口一般由光电隔离、D/A 转换和信号驱动等环节组成。其原理框图如图 6-8 所示。

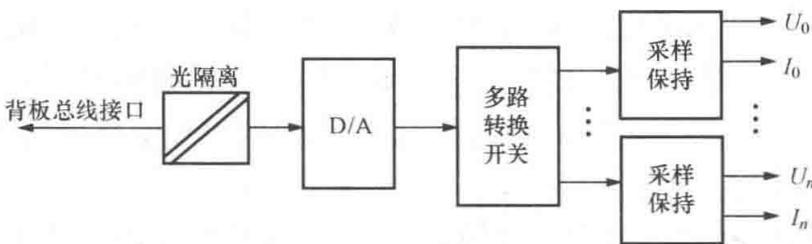


图 6-8 模拟量输出电路原理框图

## 6.2 S7-200 PLC的模拟量扩展模块

### 6.2.1 模拟量扩展模块的功能

模拟量扩展模块提供了模拟量输入输出的功能。具有以下特点：

- (1) 最佳适应性，可适用于复杂的控制场合。
- (2) 直接与传感器和执行器相连，12位的分辨率和多种输入输出范围能够不用外加放大器而与传感器和执行器直接相连，例如EM231RTD模块可直接与PT100热电阻相连。
- (3) 灵活性，当实际应用变化时，PLC可以相应地进行扩展，并可非常容易的调整用户程序。

(4) 扩展模块具有与基本单元相同的设计特点，S7-200 (CN) PLC的扩展模块种类很多，固定方式与CPU相同。如果需要扩展模块较多时，模块连接起来会过长，这时可以使用扩展转接电缆重叠排布。

(5) 安装方便，可以在标准导轨上安装，模块卡装在紧挨CPU右侧的导轨上，通过总线连接电缆与CPU互相连接。也可以直接安装，模块上有固定螺孔，可以用螺钉将模块固定在柜板上，这种安装方式建议在剧烈振动的情况下使用。

### 6.2.2 S7-200 PLC模拟量扩展模块的分类

#### 1. 常规模拟量控制模块

它有EM231模拟量输入模块、EM232模拟量输出模块、EM235模拟量输入输出模块。它适用于CPU222、CPU224、CPU224XP和CPU226系列的PLC。图6-9为EM235模块端子接线图。

#### 2. 热电阻和热电偶模块

RTD(热电阻)和热电偶模块用于CPU222、CPU224、CPU224XP和CPU226。

RTD和热电偶模块安装在一个稳定的温度环境内时，具有最佳的性能。例如，EM231CT热电偶模块有专门的冷端补偿电路。该电路在模块连接器处测量温度，并对测量值作出必要的修正，以补偿基准温度和模块处温度之间的温度差。如果EM231热电偶模块安装环境的温度变化很剧烈，则会引起附加的误差。为了达到最大的精度和重复性，热电阻扩展模块EM231RTD和热电偶模块EM231CT要安装在环境温度稳定的地方。

使用屏蔽线是最好的噪声抑制方法。如果热电偶的输入未使用，短接未使用的通道，或将它们并行连接到其他通道上。

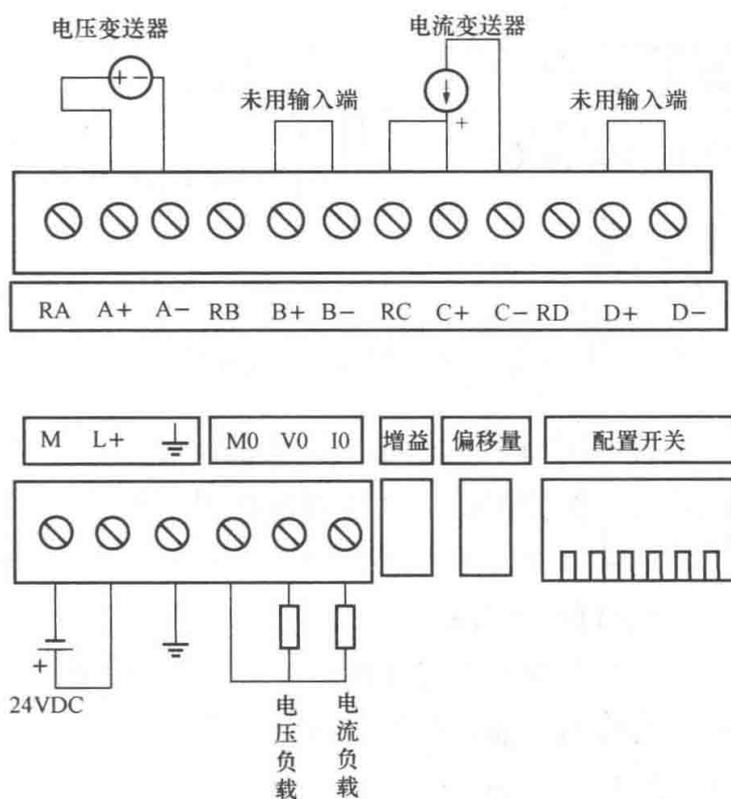


图6-9 EM235模块端子接线图

(1) EM231RTD 热电阻模块。EM231RTD 热电阻模块为 S7-200 连接各种型号的热电阻提供了方便的接口。它也允 S7-200 测量 3 个不同的电阻范围。连接到模块的热电阻必须是相同的类型。端子接线图见图 6-10。

组态 EM231RTD (热电阻) 模块使用 DIP 开关可以选择热电阻的类型、接线方式、温度测量单位和传感器熔断方向。要使 DIP 开关设置起作用, 需要重新给 PLC 和用户的 24V 电源上电。

(2) EM231 热电偶模块。它为 S7-200 系列产品提供了连接 7 种类型热电偶的使用方便、带隔离的接口: J、K、E、N、S、T 和 R。它可以使 S7-200 能连接低电平模拟信号, 测量范围为  $\pm 80\text{mV}$ 。所有连接到该模块的热电偶都必须是同一类型的。端子接线图见图 6-11。

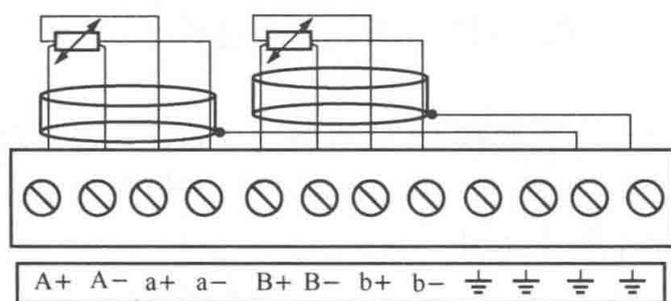


图 6-10 EM231 热电阻模块端子接线图

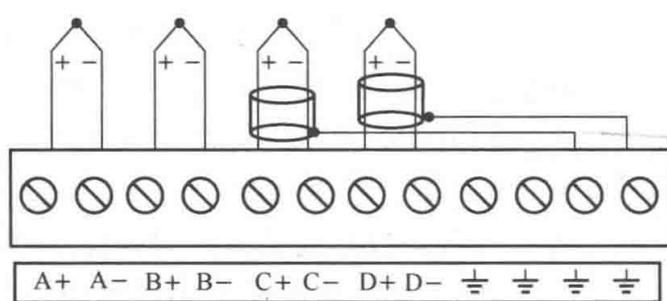


图 6-11 EM231 热电偶模块端子接线图

配置 DIP 开关位于模块的底部, 可以选择热电偶模块的类型、断线检测、温度范围和冷端补偿。要使 DIP 开关设置起作用, 需要给 PLC 和用户的 24V 电源重新上电。

### 3. 模拟量扩展模块接线安装的要求

首先确保 24VDC 传感器电源无噪声、稳定。其次传感器线尽可能短, 传感器线使用屏蔽的双绞线, 仅在传感器侧将屏蔽接终端, 避免将导线弯成锐角, 使用电缆槽进行敷线, 避免将信号线与高能量线平行布置, 若两条线必须交叉, 应以直角度相交。再次, 未用通道的输入端应短接, 通过隔离输入信号或输入信号参考于模拟量模块外部 24V 电源的公共端, 从而确保输入信号范围在技术规范所规定的共模电压之内。

## 6.2.3 根据模拟量输入模块的输出值计算对应的物理量

### 1. 模拟输入量的模拟值数据格式

模拟量输入输出模块中模拟量对应的数字称为模拟值, 模拟值用 16 位二进制补码来表示。最高位为符号位, 正数的符号位为 0, 负数的符号位为 1。对于单极性输入模拟量满量程对应的模拟值为  $0\sim 32767$ , 双极性输入模拟量满量程对应  $-32768\sim +32767$ 。

## 2. 应用举例

**【例 6-1】** 压力变送器的量程为 0~10MPa，输出信号为 4~20mA，模拟量输入模块的量程为 4~20mA，转换后的数字量为 0~32000，设转换后得到的数字为  $N$ ，试求以 kPa 为单位的压力值。

**解** 0~10MPa (0~10000kPa) 对应于转换后的数字 0~32000，转换公式为

$$P = 10000 \times N / 32000 \text{ (kPa)}$$

**【例 6-2】** 某温度变送器的量程为 -200~500℃，输出信号为 4~20mA，某模拟量输入模块将 0~20mA 的电流信号转换为数字 0~32000，对应关系见图 6-12，设转换后得到的数字为  $N$ ，求以 0.1℃ 为单位的温度值。

**解** 单位为 0.1℃ 的温度值 -2000~5000 对应于数字量 6400~32000，根据比例关系（见图 2-34），得出温度  $T$  的计算公式为

$$\frac{T - (-2000)}{N - 6400} = \frac{5000 - (-2000)}{32000 - 6400}$$

$$T = \frac{7000 \times (N - 6400)}{25600} - 2000 \text{ (0.1℃)}$$

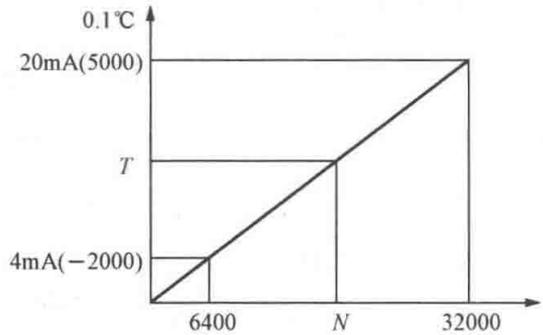


图 6-12 模拟量和转换数值的对应关系图

## 6.3 数字 PID 控制器

PID 是比例、微分、积分的缩写，PID 控制器是应用最广的闭环控制器，估计现在有 90% 以上的闭环控制采用 PID 控制器。这是因为 PID 控制具有以下优点：

(1) 不需要被控对象的数学模型。自动控制理论中的分析和设计方法基本上是在被控对象的线性定常数学模型的基础上的。该模型忽略了实际系统中的非线性和时变的因素，与实际系统有较大的差距。对于许多工业控制对象，根本就无法建立较为准确的数学模型，因此自动控制理论中的设计方法对大多数实际系统是无能为力的。对于这一类系统，使用 PID 控制可以得到比较满意的效果。

(2) 结构简单，容易实现。PID 控制器的结构典型，程序设计简单，计算工作量较小，各参数相互独立，有明确的物理意义，参数调整方便，容易实现多回路控制、串级控制等复杂的控制。

(3) 有较强的灵活性和适应性。根据被控对象的具体情况，可以采用 PID 控制器的多种变种和改进的控制方式，例如 PI、PD、带死区的 PID、积分分离式 PID 和变速积分 PID 等，但比例控制一般是必不可少的。随着智能控制技术的发展，PID 控制与神经网络控制等现代控制方法相结合，可以实现 PID 控制器的参数自整定，使 PID 控制器具有经久不衰的生命力。

(4) 使用方便。由于用途广泛、使用灵活，已有多种控制产品具有 PID 控制功能，使用中只需设定一些比较容易整定的参数即可，有的产品还具有参数自整定功能。

### 1. PID 算法

PID 控制器调节输出，保证偏差 ( $e$ ) 为零，使系统达到稳定状态，偏差 ( $e$ ) 是给定值 ( $SP$ ) 和过程变量 ( $PV$ ) 的差。PID 控制的原理基于下面的算式；输出  $M(t)$  是比例项、积分项和微分项的函数。

输出 = 比例项 + 积分项 + 微分项

$$M(t) = K_C e + K_C \int_0^t e dt + M_{\text{initial}} + K_C de/dt \quad (6-1)$$

式中  $M(t)$  ——PID 回路的输出, 是时间的函数;

$K_C$  ——PID 回路的增益;

$e$  ——PID 回路的偏差 (给定值与过程变量之差);

$M_{\text{initial}}$  ——PID 回路输出的初始值。

为了能让数字计算机处理这个控制算式, 连续算式必须离散化为周期采样偏差算式, 才能用来计算输出值。数字计算机处理的算式如下

$$M_n = K_C e_n + K_I \sum_1^n e_X + K_D (e_n - e_{n-1}) + M_{\text{initial}} \quad (6-2)$$

式中  $M_n$  ——PID 回路的输出, 是时间的函数;

$K_C$  ——PID 回路的增益;

$e_n$  ——PID 回路的偏差 (给定值与过程变量之差);

$e_{n-1}$  ——回路偏差的前一个值 (在采样时刻  $n-1$ );

$M_{\text{initial}}$  ——PID 回路输出的初始值;

$e_X$  ——采样时刻  $X$  的回路偏差值;

$K_I$  ——积分项的比例常数;

$K_D$  ——微分项的比例常数。

从式 (6-2) 可以看出, 积分项是从第 1 个采样周期到当前采样周期所有误差项的函数, 微分项是当前采样和前一次采样的函数, 比例项仅是当前采样的函数。在数字计算机中, 不保存所有的误差项, 实际上也不必要。

由于计算机从第一次采样开始, 每有一个偏差采样值必须计算一次输出值, 只需要保存偏差前值和积分项前值。作为数字计算机解决的重复性的结果, 可以得到在任何采样时刻必须计算的方程的一个简化算式。即

$$M_n = K_C e_n + K_I e_n + MX + K_D (e_n - e_{n-1}) \quad (6-3)$$

式中  $MX$  ——积分项前值。

CPU 实际使用以上简化算式的改进形式计算 PID 输出。这个改进型算式是

$$M_n = MP_n + MI_n + MD_n \quad (6-4)$$

式中  $M_n$  ——第  $n$  采样时刻的计算值;

$MP_n$  ——第  $n$  采样时刻的比例项值;

$MI_n$  ——第  $n$  采样时刻的积分项值;

$MD_n$  ——第  $n$  采样时刻的微分项值。

## 2. PID 方程的比例项

比例项  $MP$  是增益 ( $K_C$ ) 和偏差 ( $e$ ) 的乘积。其中  $K_C$  决定输出对偏差的灵敏度, 偏差 ( $e$ ) 是给定值 ( $SP$ ) 与过程变量值 ( $PV$ ) 之差。S7-200 解决的求比例项的算式是

$$MP_n = K_C (SP_n - PV_n) \quad (6-5)$$

式中  $MP_n$  ——第  $n$  采样时刻比例项的值;

$K_C$  ——增益;

$SP_n$  ——第  $n$  采样时刻的给定值；

$PV_n$  ——第  $n$  采样时刻的过程变量值。

### 3. PID 方程的积分项

积分项值  $MI$  与偏差成正比。S7-200 解决的求积分项的算式是

$$MI_n = K_C T_S / T_I (SP_n - PV_n) + MX \quad (6-6)$$

式中  $MI_n$  ——第  $n$  采用时刻的积分项值；

$K_C$  ——回路增益；

$T_S$  ——回路采样时间；

$T_I$  ——积分时间；

$SP_n$  ——第  $n$  采用时刻的给定值；

$PV_n$  ——第  $n$  采用时刻的过程变量值；

$MX$  ——积分项前值（第  $n-1$  采用时刻的积分值）。

积分和 ( $MX$ ) 是所有积分项前值之和。在每次计算出  $MI_n$  之后，都要用  $MI_n$  去更新  $MX$ 。其中  $MI_n$  可以被调整或限定。 $MX$  的初值通常在第一次计算输出以前被设置为  $M_{\text{initial}}$  (初值)。积分项还包括其他几个常数：增益 ( $K_C$ )、采样时间间隔 ( $T_S$ ) 和积分时间 ( $T_I$ )。其中采样时间是重新计算输出的时间间隔，而积分时间控制积分项在整个输出结果中影响的大小。

### 4. PID 方程的微分项

微分项值  $MD$  与偏差的变化成正比。S7-200 使用下列算式来求解微分项

$$MD_n = K_C T_D / T_S [(SP_n - PV_n) - (SP_{n-1} - PV_{n-1})] \quad (6-7)$$

为了避免给定值变化的微分作用而引起的跳变，假定给定值不变 ( $SP_n = SP_{n-1}$ )。这样，可以用过程变量的变化替代偏差的变化，计算式可改进为

$$MD_n = K_C T_D / T_S (SP_n - PV_n - SP_{n-1} + PV_{n-1}) \quad (6-8)$$

或

$$MD_n = K_C T_D / T_S (PV_{n-1} - PV_n) \quad (6-9)$$

式中  $MD_n$  ——第  $n$  采样时刻的微分项值；

$K_C$  ——回路增益；

$T_S$  ——回路采样时间；

$T_D$  ——微分时间；

$SP_n$  ——第  $n$  采样时刻的给定值；

$SP_{n-1}$  ——第  $n-1$  采样时刻的给定值；

$PV_n$  ——第  $n$  采样时刻的过程变量值；

$PV_{n-1}$  ——第  $n-1$  采样时刻的过程变量值。

为了下一次计算微分项值，必须保存过程变量，而不是偏差。在第一采样时刻，初始化为  $PV_{n-1} = PV_n$ 。

## 6.4 S7-200 PLC 的 PID 控制及其应用

### 6.4.1 PID 功能

S7-200 CPU 提供了 8 个回路的 PID 功能，用以实现需要按照 PID 控制规律进行自动

调节的控制任务, 比如温度、压力和流量控制等。PID 功能一般需要模拟量输入, 以反映被控制的物理量的实际数值, 称为反馈; 而用户设定的调节目标值, 即为给定。PID 运算的任务就是根据反馈与给定的相对差值, 按照 PID 运算规律计算出结果, 输出到固态开关元件 (控制加热棒), 或者变频器 (驱动水泵) 等执行机构进行调节, 以达到自动维持被控制的量跟随给定变化的目的。

### 1. PID 指令

S7-200 中 PID 功能的核心是 PID 指令。PID 指令需要为其指定一个以 V 变量存储区地址开始的 PID 回路表 (TBL), 以及 PID 回路号 (LOOP)。PID 回路表提供了给定和反馈, 以及 PID 参数等数据入口, PID 运算的结果也在回路表输出。

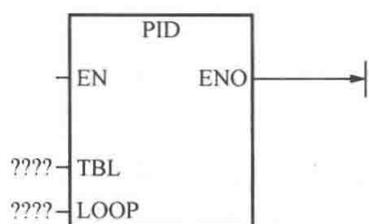


图 6-13 PID 指令的梯形图指令盒形式

PID 指令 (又称为 PID 回路指令) 的梯形图指令盒形式如图 6-13 所示。使能输入有效时, 该指令利用回路表中的输入信息和组态信息, 进行 PID 运算。梯形图的指令盒中有 2 个数据输入端: TBL, 回路表的起始地址, 是由 VB 指定的字节型数据; LOOP, 回路号, 是 0~7 的常数。

STL 指令格式: PID TBL, LOOP

在程序中最多可以用 8 条 PID 指令。如果两个或两个以上的 PID 指令用了同一个回路号, 那么即使这些指令的回路表不同, 这些 PID 运算之间也会相互干涉, 产生不可预料的结果。

### 2. PID 回路表

回路表有 80 字节长, 它的格式如表 6-1 所示。

回路表包含 9 个参数, 用来控制和监视 PID 运算。这些参数分别是过程变量当前值 ( $PV_n$ ), 过程变量前值 ( $PV_{n-1}$ ), 给定值 ( $SP_n$ ), 输出值 ( $M_n$ ), 增益 ( $K_C$ ), 采样时间 ( $T_S$ ), 积分时间 ( $T_I$ ), 微分时间 ( $T_D$ ) 和积分项前值 ( $MX$ )。

为了让 PID 运算以预想的采样频率工作, PID 指令必须用在定时发生的中断程序中, 或者用在主程序中被定时器所控制以一定频率执行。采样时间必须通过回路表输入到 PID 运算中。

表 6-1 PID 回路表

偏移地址	域	格式	类型	描述
T+0	过程变量 (反馈) ( $PV_n$ )	双字—实数	输入	过程变量, 必须在 0.0~1.0 之间
T+4	设定值 (给定) ( $SP_n$ )	双字—实数	输入	给定值, 必须在 0.0~1.0 之间
T+8	输出值 ( $M_n$ )	双字—实数	输入/输出	输出值, 必须在 0.0~1.0 之间
T+12	增益 (P 参数) ( $K_C$ )	双字—实数	输入	增益是比例常数, 可正可负
T+16	采样时间 ( $T_S$ )	双字—实数	输入	单位为 s, 必须是正数
T+20	积分时间 (I 参数) ( $T_I$ )	双字—实数	输入	单位为 min, 必须是正数
T+24	微分时间 (D 参数) ( $T_D$ )	双字—实数	输入	单位为 min, 必须是正数
T+28	积分项前项 ( $MX$ )	双字—实数	输入/输出	积分项前项, 必须在 0.0~1.0 之间
T+32	过程变量前值 ( $PV_{n-1}$ )	双字—实数	输入/输出	最近一次 PID 运算的过程变量值
T+36~ T+79	保留给 PID 自整定的变量设置用			

过程变量和给定值是PID运算的输入值,因此回路表中的这些变量只能被PID指令读而不能被改写。

输出变量是由PID运算产生的,所以在每一次PID运算完成之后,需更新回路表中的输出值,输出值被限定在0.0~1.0之间。当输出由手动转变为PID(自动)控制时,回路表中的输出值可以用来初始化输出值。

回路表中的给定值与过程变量的差值( $e$ )是用于PID运算中的差分运算,用户最好不要去修改此值。

### 3. 回路控制类型的选择

在许多控制系统中,只需要一种或两种回路控制类型。例如只需要比例回路或者比例积分回路。通过设置常量参数,可以选择需要的回路控制类型。

如果不想要积分动作(PID计算中没有“I”),可以把积分时间(复位)置为无穷大“INF”。即使没有积分作用,积分项还是不为零,因为有初值MX。

如果不想要微分回路,可以把微分时间置为零。

如果不想要比例回路,但需要积分或积分微分回路,可以把增益设为0.0,系统会在计算积分项和微分项时,把增益当作1.0看待。

### 4. 回路输入的转换和标准化

每个PID回路有两个输入量,即给定值(SP)和过程变量(PV)。给定值通常是一个固定的值,比如设定的汽车速度。过程变量是与PID回路输出有关,可以衡量输出对控制系统作用的大小。在汽车速度控制系统的实例中,过程变量应该是测量轮胎转速的测速计输入。

给定值和过程变量都可能是现实世界的值,它们的大小、范围和工程单位都可能不一样。在PID指令对这些现实世界的值进行运算之前,必须把它们转换成标准的浮点型表达形式。

转换的第一步是把16位整数值转成浮点型实数值。下面的指令序列提供了实现这种转换的方法:

```
ITD AIW0,AC0      //将输入值转换为双整数。
DTR AC0, AC0      //将32位双整数转换为实数。
```

下一步是将现实世界的值的实数值表达形式转换成0.0~1.0之间的标准化值。下面的算式可以用于标准化给定值或过程变量值

$$R_{\text{Norm}} = (R_{\text{Raw}}/S_{\text{Pan}}) + \text{Offset} \quad (6-10)$$

式中  $R_{\text{Norm}}$ ——标准化的实数值;

$R_{\text{Raw}}$ ——没有标准化的实数值或原值;

$\text{Offset}$ ——单极性为0.0,双极性为0.5;

$S_{\text{Pan}}$ ——值域大小,可能的最大值减去可能的最小值,单极性为32000(典型值),双极性为64000(典型值)。

下面的例子是把双极性实数标准化为0.0~1.0之间的实数,通常用在第一步转换之后:

```
/R 64000.0, AC0      //累加器中的标准化值
+ R 0.5, AC0         //加上偏置,使其在0.0~1.0之间
MOVR AC0, VD100     //标准化的值存入回路表
```

### 5. 回路输出值转换成刻度整数值

回路输出值一般是控制变量，比如在汽车速度控制中，可以是油阀开度的设置。回路输出是 0.0~1.0 之间的一个标准化了的实数值。在回路输出可以用于驱动模拟输出之前，回路输出必须转换成一个 16 位的标定整数值。这一过程，是给定值或过程变量的标准化转换的逆过程。第一步是使用下面给出的公式，将回路输出转换成一个标定的实数值

$$R_{\text{Scal}} = (M_n - \text{Offset}) S_{\text{Pan}} \quad (6-11)$$

式中  $R_{\text{Scal}}$ ——回路输出的刻度实数值；

$M_n$ ——回路输出的标准化实数值；

$\text{Offset}$ ——单极性为 0.0，双极性为 0.5；

$S_{\text{Pan}}$ ——值域大小，可能的最大值减去可能的最小值，单极性为 32000（典型值），双极性为 64000（典型值）。

这一过程可以用下面的指令序列完成：

```
MOVW VD108, AC0      //把回路输出值移入累加器
-R 0.5, AC0          //仅双极性有此句
*R 64000.0, AC0      //在累加器中得到刻度值
```

下一步是把回路输出的刻度转换成 16 位整数，可通过下面的指令序列来完成：

```
ROUND AC0, AC0      //把实数转换为 32 位整数
DTI AC0, LW0        //把 32 位整数转换为 16 位整数
MOVW LW0, AQW0      //把 16 位整数写入模拟输出寄存器
```

### 6. 正作用或反作用回路

如果增益为正，那么该回路为正作用回路；如果增益为负，那么是反作用回路（对于增益值为 0.0 的 I 或 ID 控制，如果指定积分时间、微分时间为正，就是正作用回路；如果指定为负值，就是反作用回路）。

### 7. 控制方式

S7-200 的 PID 回路没有设置控制方式，只有当 PID 指令盒接通时，才执行 PID 运算。在这种意义上说，PID 运算存在一种“自动”运行方式。当 PID 运算不被执行时，称为“手动”模式。

同计数器指令相似，PID 指令有一个使能位。当该使能位检测到一个信号的正跳变（从 0 到 1），PID 指令执行一系列的动作，使 PID 指令从手动方式无扰动地切换到自动方式。为了达到无扰动切换，在转变到自动控制前，必须把手动方式下的输出值填入回路表中的  $M_n$  栏。PID 指令对回路表中的值进行下列动作，以保证当使能位正跳变出现时，从手动方式无扰动切换到自动方式：

- (1) 置给定值 ( $SP_n$ ) = 过程变量 ( $PV_n$ )。
- (2) 置过程变量前值 ( $PV_{n-1}$ ) = 过程变量现值 ( $PV_n$ )。
- (3) 置积分项前值 ( $MX$ ) = 输出值 ( $M_n$ )。

PID 使能位的默认值是 1，在 CPU 启动或从 STOP 方式转到 RUN 方式时建立。CPU 进入 RUN 方式后首次使 PID 块有效，没有检测到使能位正跳变，那么就没有无扰动切换的动作。

### 8. 报警与特殊操作

PID指令是执行PID运算的简单而功能强大的指令。如果需要其他处理,如报警检查或回路变量的特殊计算等,则这些处理必须使用S7-200支持的基本指令来实现。

### 9. 出错条件

如果指令指定的回路表起始地址或PID回路号操作数超出范围,那么在编译期间,CPU将产生编译错误(范围错误),从而编译失败。

PID指令不检查回路表中的一些输入值,必须保证过程变量和设定值(以及作为输入的偏置和前一次过程变量)在0.0~1.0之间。

如果PID计算的算术运算发生错误,那么特殊存储器标志位SM1.1(溢出或非法值)会被置1,并且中止PID指令的执行(要想消除这种错误,单靠改变回路表中的输出值是不够的,正确的方法是在下一次执行PID运算之前,改变引起算术运算错误的输入值,而不是更新输出值)。

### 10. PID指令使用中的其他问题

系统冷启动时,测量值PV巨大的变化将导致微分部分产生过大的校正作用,这时最好去掉微分部分。可以用PV(或MV)的变化值来决定从PI到PID的切换点。

PID数据块中有四个报警值,它们用来设置MV和PV的上限和下限,用于警告系统脱离控制。通过反映PV和MV变化的报警标志,可以监视系统的状态并且调节PID的参数。当系统接近设定值SV时,PV和MV的变化很小,此时应使用完整的PID控制,使系统的输出跟随SV。

#### 【例6-3】水箱加热系统的PLC闭环自动控制系统。

本实例中利用PLC的模拟量输入通道实现对温度的自动测量。利用PLC的模拟量输出通道实现对晶闸管调功器的控制。

系统的组成如图6-14所示,本系统的被控对象是1KW电加热管,被控制量是水箱的水温,PLC的模拟量输出信号控制调功器的电功率输出,由调功器控制电加热管的通断,被控对象为水箱中的单相电热管,被控制量为水箱水温。它由铂电阻Pt100测定,输入到温度变送器上,量程为0~100℃。温度变送器变换为0~10V电压信号,传送给PLC的模拟量输入通道。PLC的模拟量输入通道的量程范围为-10~10V,对应的数据为-32000~32000。PLC的模拟量输出通道为电压形式,电压范围为0~10V。对应的数据范围为0~32000。采用PLC实现水箱温度的自动调节控制,由PLC模拟量输入通道采集水箱温度数值,由PLC模拟量输出通道向晶闸管调功器发出控制信号,从而达到控制水箱温度的目的。

设计满足控制要求的主程序如图6-15所示,中断程序如图6-16所示。假设定值为35℃,采用下列控制参数值: $K_c$ 为2.0, $T_s$ 为0.1s, $T_i$ 为1min, $T_d$ 为0.01min。

#### 6.4.2 PID向导

STEP 7 - Micro/WIN提供了PID Wizard(PID指令向导),可以帮助用户方便地生成一个闭环控制过程的PID算法。用户只要在向导的指导下填写相应的参数,就可以方便快捷地完成PID运算的自动编程。用户只需在应用程序中调用PID向导生成的子程序,就可以完成PID控制任务。向导最多允许配置8个PID回路。

PID向导既可以生成模拟量输出的PID控制算法,也支持开关量输出(如控制加热棒);

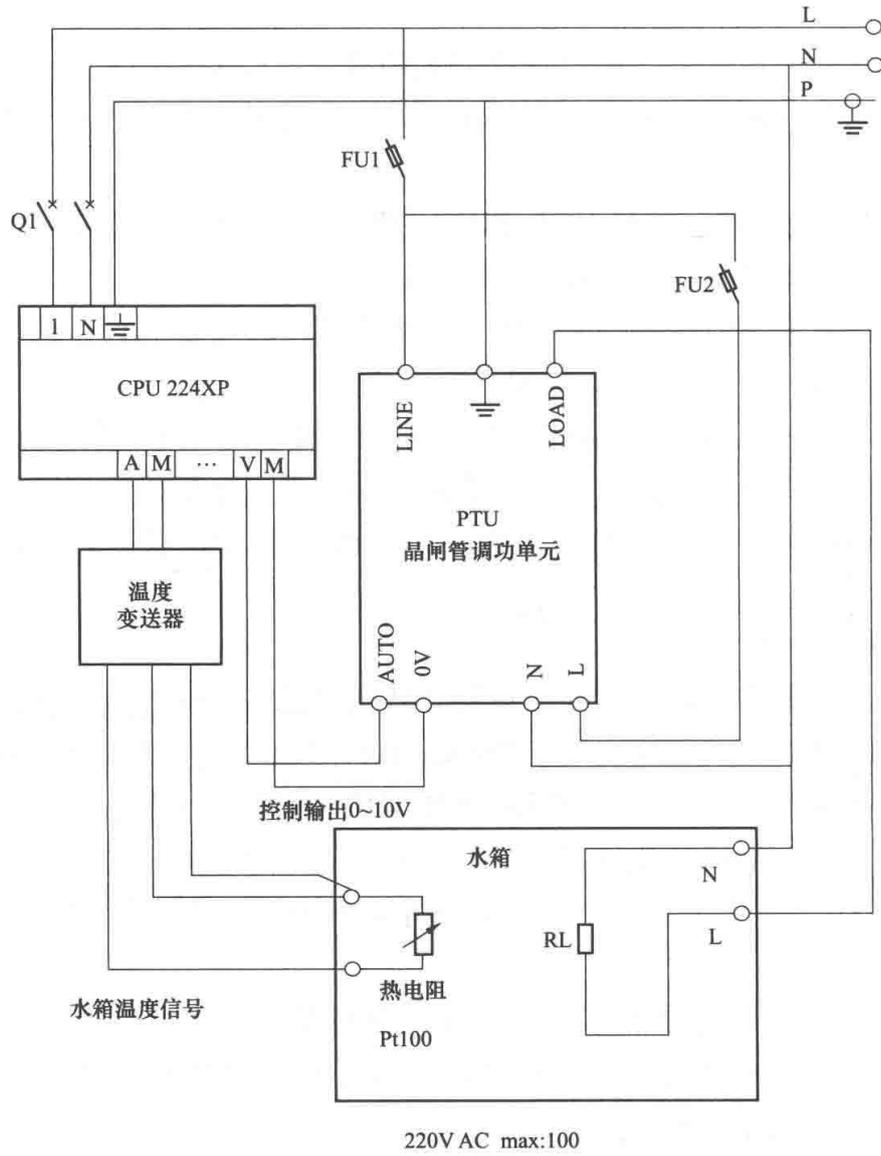


图 6-14 温度闭环自动调节控制原理图

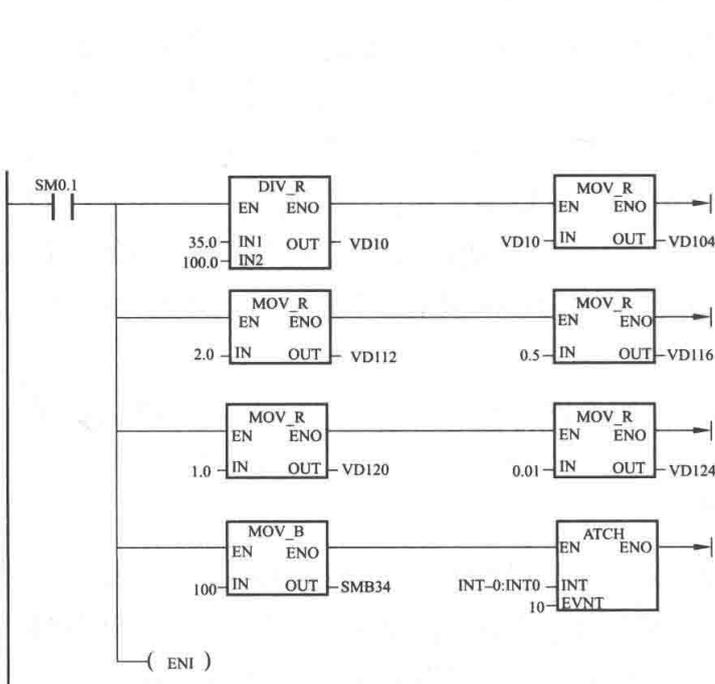


图 6-15 主程序

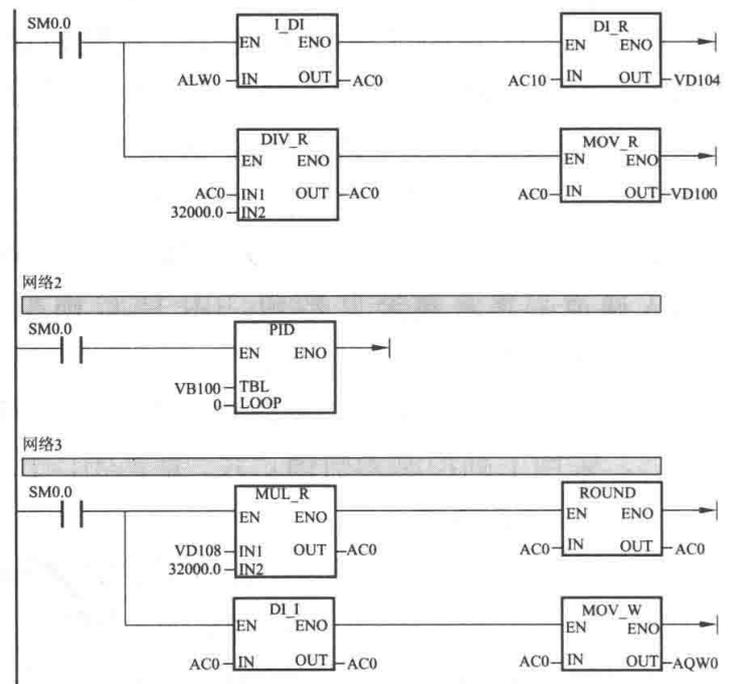


图 6-16 中断程序

即支持连续自动调节，也支持手动参与控制，并能实现手动到自动的无扰切换。除此之外，它还支持PID反作用调节。

PID功能块只接受0.0~1.0之间的实数（实际上就是百分比）作为反馈、给定与控制输出的有效数值，如果是直接使用PID功能块编程，必须保证数据在这个范围之内，否则会出错。其他如增益、采样时间、积分时间和微分时间都是实数。但PID向导已经把外围实际的物理量与PID功能块需要的输入输出数据之间进行了转换，不再需要用户自己编程进行输入/输出的转换与标准化处理。

建议使用PID向导对PID编程，以简化编程及避免不必要的错误。

### 1. PID向导的使用

PID向导可以指导用户在几分钟内迅速地生成一个PID控制程序，方法是单击Micro/WIN导航栏“Tools”中的“指令向导”图标或在命令菜单中选择“Tools”>“Instruction Wizard”，然后在指令向导窗口中选择PID向导进入配置。PID向导的使用步骤如下：

第一步：定义需要配置的PID回路号。

第二步：设定PID回路参数，如图6-17所示。

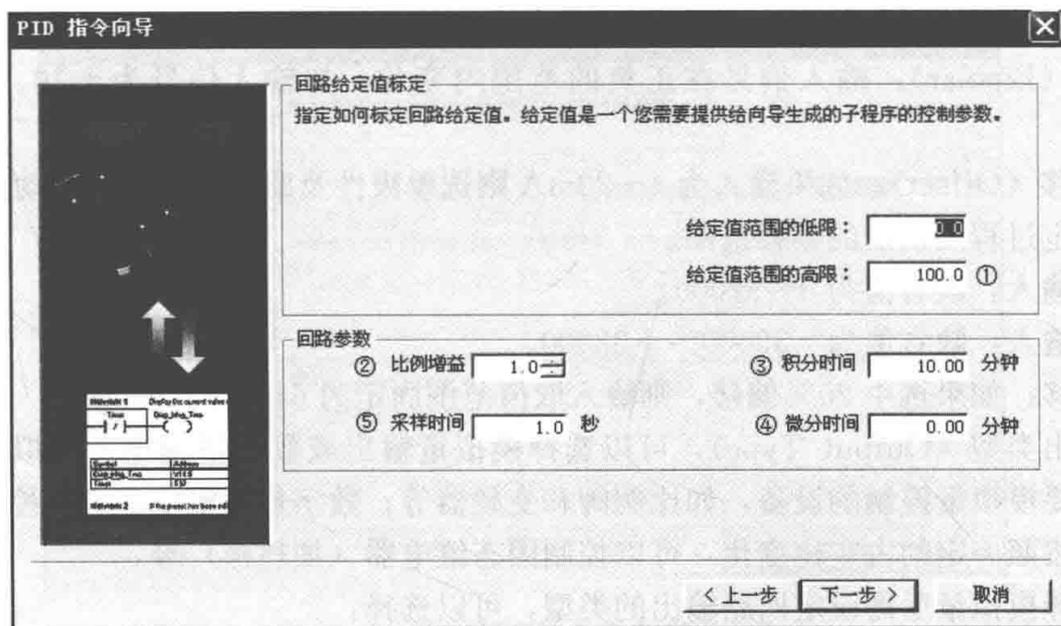


图 6-17 设定 PID 回路参数

(1) 定义回路设定值（SP，即给定）的高低限的范围与过程变量范围相对应。在低限（Low Range）和高限（High Range）输入域中输入实数，默认值为0.0和100.0，表示给定值的大小占过程反馈量程的百分比，也可以用实际的工程单位量程表示。

(2) 增益（Gain）：即比例常数。

(3) 积分时间（Integral Time）：如果不想要积分作用，可以把积分时间设为无穷大。

(4) 微分时间（Derivative Time）：如果不想要微分回路，可把微分时间设为0。

(5) 采样时间（Sample Time）：是PID控制回路对反馈采样和重新计算输出值的时间间隔。

在一般的控制系统中，经常只用到PI调节，这时需要把微分参数设为零。

第三步：设定PID回路输入输出参数，如图6-18所示。

(1) 指定输入类型。

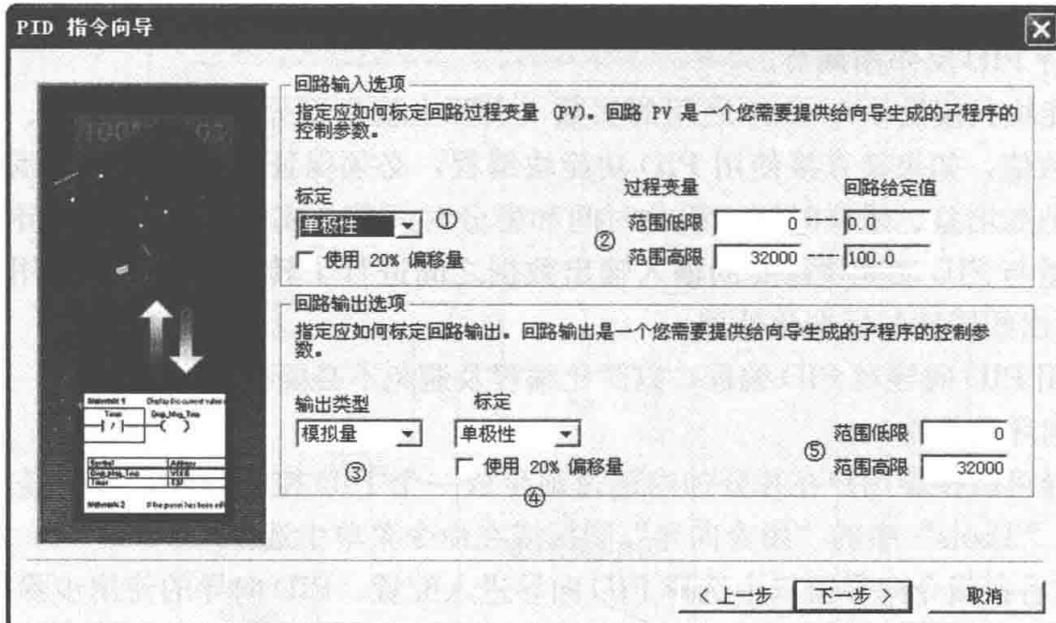


图 6-18 设定 PID 输入输出参数

单极性 (Unipolar): 即输入的信号为正, 如 0~10V 或 0~20mA 等。

双极性 (Bipolar): 输入信号在正负的范围內变化, 如输入信号为 ±10、±5V 等时选用。

20%偏移 (Offset): 如果输入为 4~20mA 则选单极性及此项, 向导会自动进行转换。

(2) 设定过程反馈值的量程范围。

单极性输入: 缺省值为 0~32000。

双极性输入: 缺省值为 -32000~+32000。

20%偏移: 如果选中 20%偏移, 则输入取值范围固定为 6400~32000。

(3) 输出类型 (Output Type), 可以选择模拟量输出或数字量输出。模拟量输出用来控制一些需要模拟量控制的设备, 如比例阀和变频器等; 数字量输出实际上是控制输出点的通、断状态按照一定的占空比变化, 可以控制固态继电器 (加热棒) 等。

(4) 选择模拟量后需设定回路输出的类型, 可以选择:

单极性输出: 可为 0~10V 或 0~20mA 等。

双极性输出: 可为 ±10V 或 ±5V 等。

20%偏移: 如果选中 20%偏移, 使输出为 4~20mA。

(5) 选择模拟量后需设定回路输出变量值的范围, 可以选择:

单极性输出: 缺省值为 0~32000。

双极性输出: 缺省值为 -32000~+32000。

20%偏移: 如果选中 20%偏移, 则输出取值范围为 6400~32000。

如果选择了开关量输出, 需要设定输出占空比控制的周期。

第四步: 设定回路报警选项 (也可不选)。

第五步: 指定 PID 运算数据存储区。

PID 向导需要一个 120 字节的数据存储区 (V 区), 如图 6-19 所示, 要注意在程序的其他地方不要重复使用这些地址。

第六步: 指定向导所生成的 PID 子程序名和中断程序名 (可默认的或者自己定义) 及

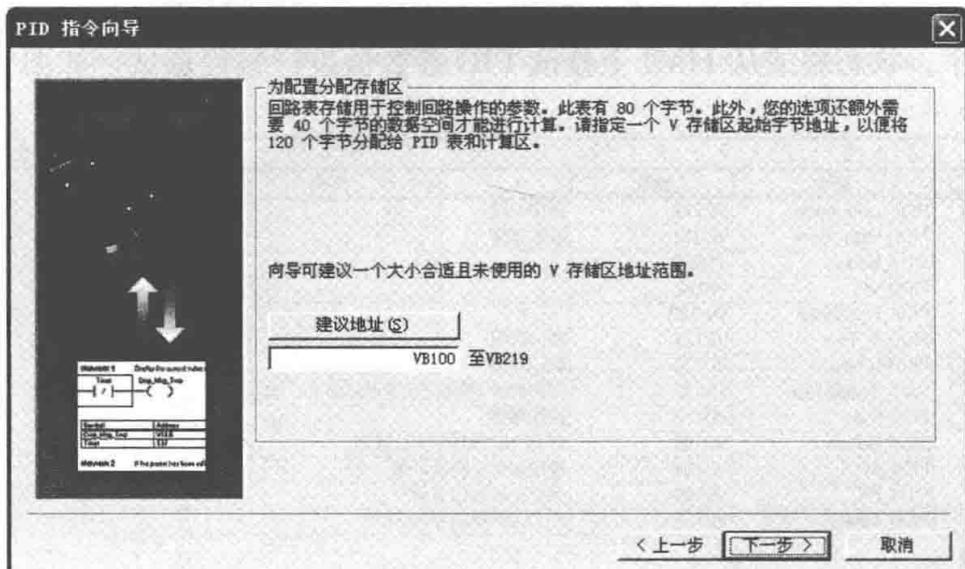


图 6-19 设定 PID 运算数据存储区参数

添加手动模式，如图 6-20 所示。

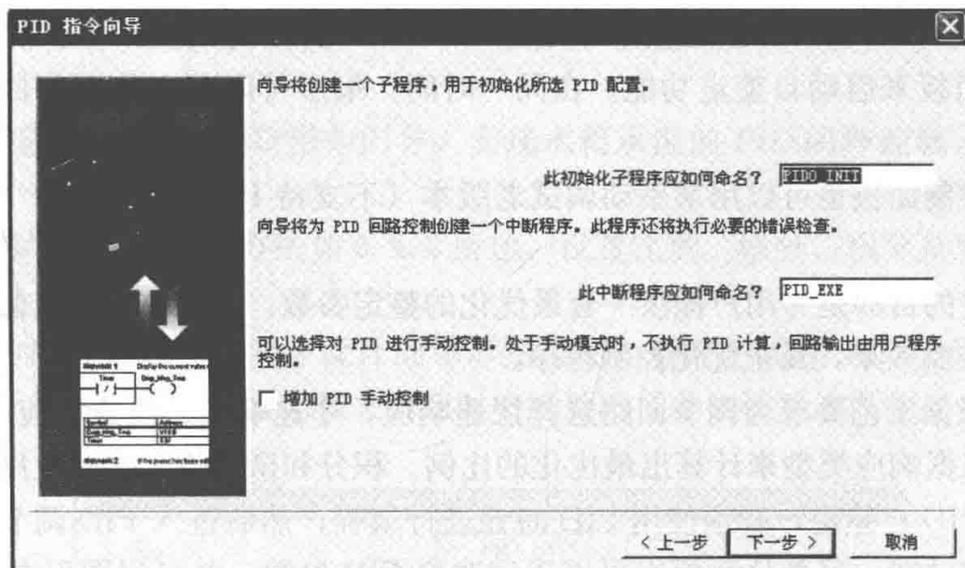


图 6-20 设定子程序及中断程序名称

第七步：生成 PID 子程序、中断程序及符号表等。

PID 向导使用了 SMB34 定时中断，在实现其他的编程任务时，不能再使用此中断，否则会引起 PID 运行错误。

在完成向导配置后，只要在程序中调用向导所生成的 PIDx\_INIT 即可，如图 6-21 所示。它包括：  
①反馈过程变量值地址；②设定值（可以是实数也可以是设定值变量的地址）；③手/自动控制转换；④手动输出值；⑤控制输出地址。

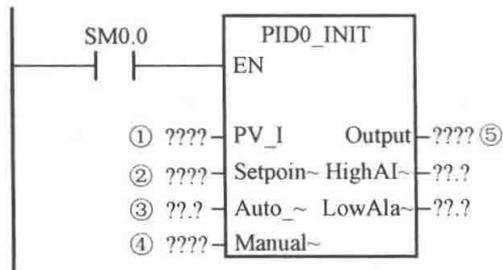


图 6-21 调用 PID 向导生成的子程序

只能使用 SM0.0 作唯一的条件调用 PIDx\_INIT，否则会造成 PID 控制功能不运行。

### 2. PID 向导符号表

完成 PID 向导配置后，会自动生成一个 PID 向导符号表，在这个符号表中可以找到 P

(比例)、I (积分)、D (微分) 等参数的地址。利用这些参数地址用户可以方便地在 Micro/WIN 中使用程序、状态表或从 HMI 上修改 PID 参数值进行编程调试, 如图 6-22 所示。

符号	地址	注释
PID0_Low_Alarm	VD216	报警下限
PID0_High_Alarm	VD212	报警上限
PID0_Mode	V182.0	
PID0_wS	VB182	
PID0_D_Counter	Vw180	
PID0_D_Time	VD124	微分时间
PID0_I_Time	VD120	积分时间
PID0_SampleTime	VD116	采样时间 (要修改请重新运行 PID 向导)
PID0_Gain	VD112	回路增益
PID0_Output	VD108	标准化的回路输出计算值
PID0_SP	VD104	标准化的过程给定值
PID0_PV	VD100	标准化的过程变量
PID0_Table	VB100	PID 0 的回路表起始地址

图 6-22 PID 符号表

### 6.4.3 PID 自整定

在硬件上, S7-200 CPU V23 以上版本已经支持 PID 自整定功能。在软件上, 在 STEP 7-Micro/WIN V4.0 以上版本中增加了 PID 调节控制面板。可以使用用户程序或 PID 调节控制面板来启动自整定功能。在同一时间, 最多可以有 8 个 PID 回路同时进行自整定。

PID 调节控制面板也可以用来手动调试老版本 (不支持 PID 自整定的) CPU 的 PID 控制回路。

PID 自整定的目的是为用户提供一套最优化的整定参数, 使用这些整定值可以使控制系统达到最佳的控制效果, 真正优化控制程序。

用户可以根据工艺要求为调节回路选择快速响应、中速响应、慢速响应或极慢速响应。PID 自整定会根据响应类型来计算出最优化的比例、积分和微分值, 并可应用到控制中。

要想使用 PID 自整定, 必须使用 PID 向导进行编程, 然后进入 PID 调节控制面板, 启动、停止自整定功能。另外从面板中可以手动改变 PID 参数, 并可用图形方式监视 PID 回路的运行。

在 Micro/WIN V4.0 在线的情况下, 单击导航栏 “Tools” 中的 PID 控制面板或从主菜单 “Tools” > “PID Tune Control Panel” 进入 PID 调节控制面板中, 如图 6-23 所示。如果面板没有被激活 (所有地方都是灰色), 可单击 “Configure” (配置) 按钮运行 CPU。

为了保证 PID 自整定的成功, 在启动 PID 自整定前, 需要调节 PID 参数, 使 PID 调节器基本稳定, 输出、反馈变化平缓, 并且使反馈比较接近给定。另外设置合适的给定值, 使 PID 调节器的输出远离趋势图的上、下坐标轴, 以免 PID 自整定开始后输出值的变化范围受限制。

在程序中使 PID 调节器工作在自动模式下, 然后单击 “开始自动调节” 按钮启动 PID 自整定功能, 这时按钮变为 “停止自动调节”。自整定控制器会在回路的输出中加入一些小的阶跃变化, 使得控制过程产生小的振荡, 自动计算出优化的 PID 参数并将计算出的 PID 参数显示在 PID 参数区。当按钮再次变为 “停止自动调节” 时, 表示系统已经完成了 PID 自整定。此时 PID 参数区所显示的为整定后的参数, 如果希望系统更新为自整定后的 PID

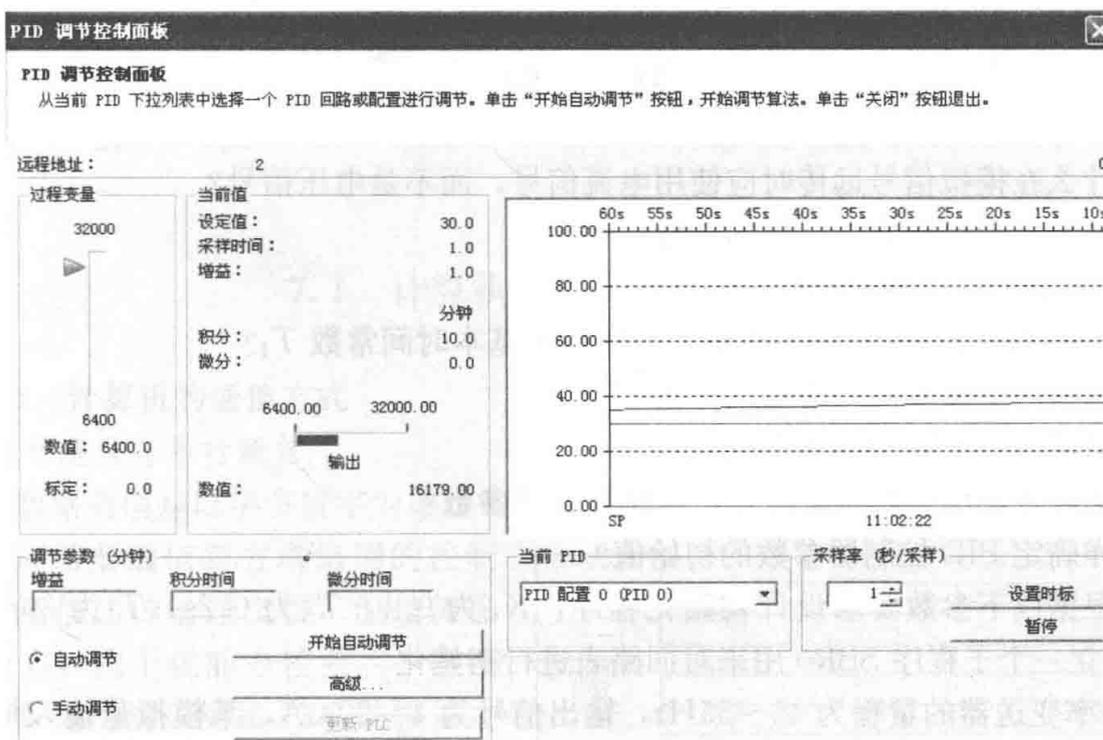


图 6-23 PID 的调节控制板

参数，单击“更新 PLC”按钮即可。

**【例 6-4】**应用 PLC 的 PID 指令向导，实现水箱水温的 PID 闭环控制，系统的控制要求同 [例 6-3] 类似。

PID 指令向导参数的设置方法如 6.4.2 所述，设置比例、微分、积分及采用时间，变量表的起始地址设为 VB2100。

图 6-24 为根据 PID 指令向导设计的梯形图程序。程序中调用 PID 子程序时，不用考虑中断程序，子程序会自动初始化相关的定时中断处理任务，然后中断程序会自动执行。

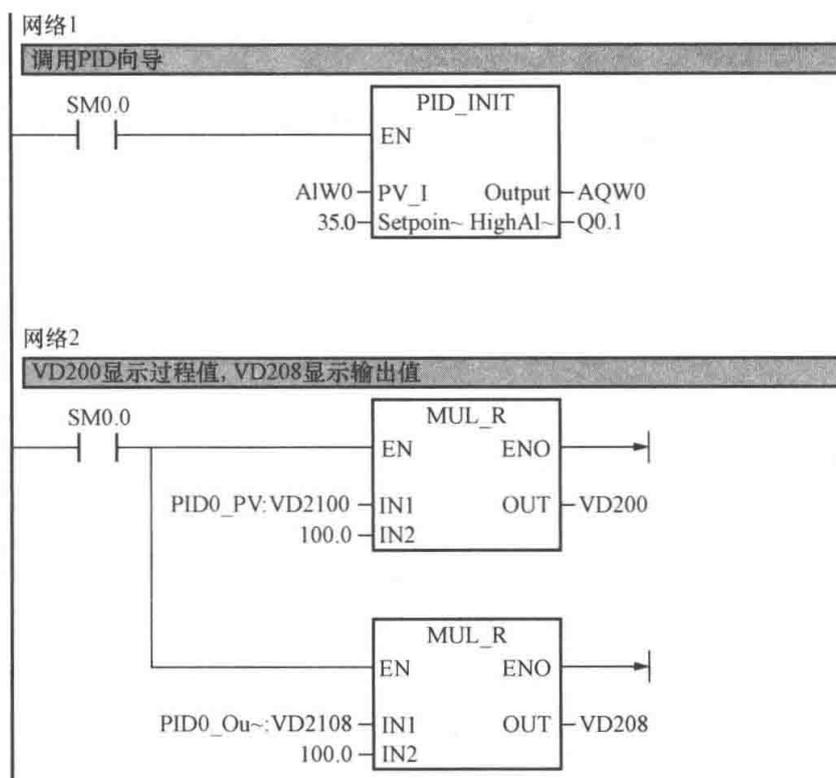


图 6-24 根据 PID 指令向导设计的梯形图程序

## 习 题



1. 为什么在模拟信号远传时应使用电流信号, 而不是电压信号?
2. 怎样判别闭环控制中反馈的极性?
3. PID 控制为什么会得到广泛的使用?
4. PID 中的积分部分有什么作用, 怎样调节基本时间常数  $T_I$ ?
5. 什么情况下需要使用增量式算法 PID?
6. 反馈量微分 PID 算法有什么优点?
7. 如果闭环响应的超调量过大, 应调节哪些参数?
8. 怎样确定 PID 控制器参数的初始值?
9. 试根据以下参数要求设计初始化程序:  $K_C$  为 0.4,  $T_S$  为 0.2s,  $T_I$  为 30min,  $T_D$  为 15min, 建立一个子程序 SBR0 用来对回路表进行初始化。
10. 频率变送器的量程为 45~55Hz, 输出信号为 4~20mA, 某模拟量输入模块输入信号的量程为 4~20mA, 转换后的数字量为 0~32000, 设转换后得到的数字为  $N$ , 试求以 Hz 为单位的频率值, 并设计出程序。
11. 某温度变送器的量程为 -100~500°C, 输出信号为 4~20mA, 某模拟量输入模块将 0~20mA 的电流信号转换为数字 0~27648, 设转换后得到的数字为  $N$ , 求以 0.1°C 为单位的温度值, 并设计出程序。

## 第 7 章 S7-200 PLC 的通信及网络控制

### 7.1 计算机通信方式与串行接口

#### 7.1.1 计算机的通信方式

##### 1. 并行通信与串行通信

并行数据通信是以字节或字为单位的数据传输方式，除了 8 根或 16 根数据线、1 根公共线外，还需要通信双方联络用的控制线。并行通信的传输速度快，但是传输线的根数多，抗干扰能力较差，一般用于近距离数据传输，例如 PLC 的模块之间的数据传输，如图 7-1 所示。

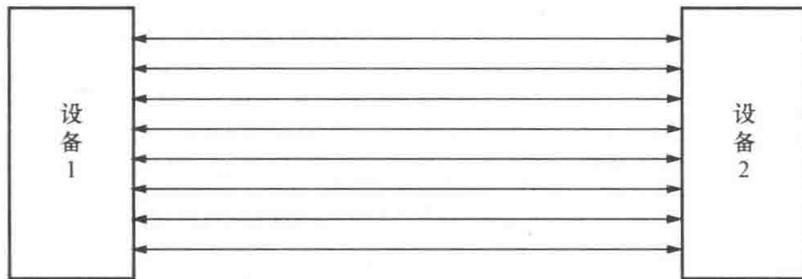


图 7-1 并行通信

串行通信多用于可编程序控制器与计算机之间，及多台可编程序控制器之间的数据传送。传送时，数据的各个不同位分时使用同一条传输线，从低位开始一位接一位按顺序传送，数据有多少位就需要传送多少次，如图 7-2 所示。

串行通信多用于可编程序控制器与计算机之间，及多台可编程序控制器之间的数据传送。传送时，数据的各个不同位分时使用同一条传输线，从低位开始一位接一位按顺序传送，数据有多少位就需要传送多少次，如图 7-2 所示。



图 7-2 串行通信

串行通信按信息在设备间的传送方向又分为单工、半双工和全双工三种方式。

串行通信按时钟可分为同步传送和异步传送两种方式。

##### 2. 异步通信与同步通信

在串行通信中，接收方和发送方应使用相同的传输速率。接收方和发送方的标称传输速率虽然相同，但他们之间总是有些微小的差别。如果不采取措施，在连续传送大量的信息时，将会因积累误差造成发送和接收的数据错位，使接收方收到错误的信息。为了解决这一问题，需要使发送过程和接收过程同步。按同步方式的不同，串行通信分为异步通信和同步通信。

(1) 异步通信：允许传输线上的各个部件有各自的时钟，在各部件之间进行通信时没有

统一的时间标准, 相邻两个字符传送数据之间的停顿时间长短是不一样的, 它是靠发送信息时同时发出字符的开始和结束标志信号来实现的, 如图 7-3 所示。

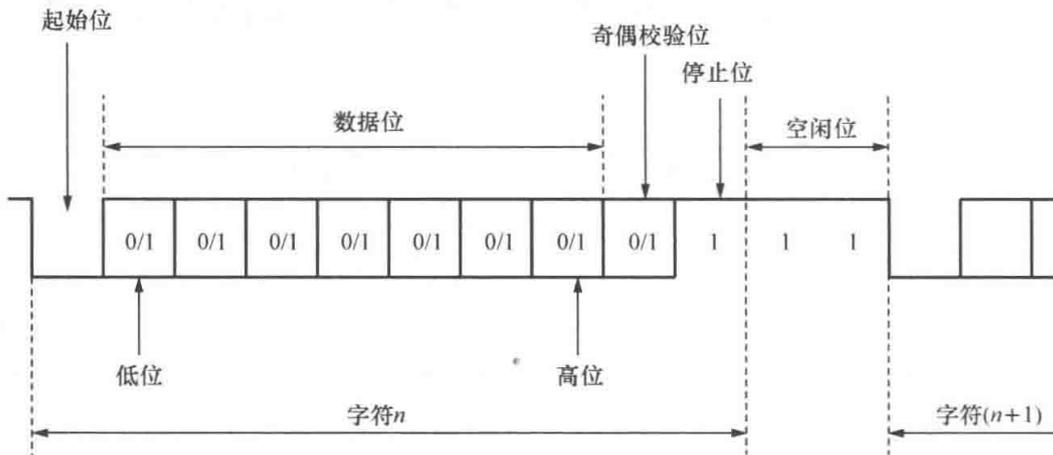


图 7-3 异步通信的数据格式

(2) 同步通信: 以字节为单位, 每次传送 1~2 个同步字符、若干个数据字节和效验字符。同步字符起联络作用, 它用来通知接收方开始接收数据。在同步通信中, 发送方和接收方要保持完全的同步, 也就是说发送方和接收方应使用同一时钟频率。它可以通过调制解调方式在数据流中提取出同步信号, 使接收方得到与发送方同步的接收时钟信号。

由于同步通信方式不需要在每个数据字符中增加起始位、停止位和奇偶效验位, 只需要在发送的数据块之前一两个同步字符, 所以传输效率高, 但是对硬件的要求较高。

### 7.1.2 网络结构概述

#### 1. 简单网络

多台设备通过传输线相连, 可以实现多设备间的通信, 就形成网络结构。如图 7-4 所示就是一种最简单的网络结构, 它由单主设备和多个从设备构成。

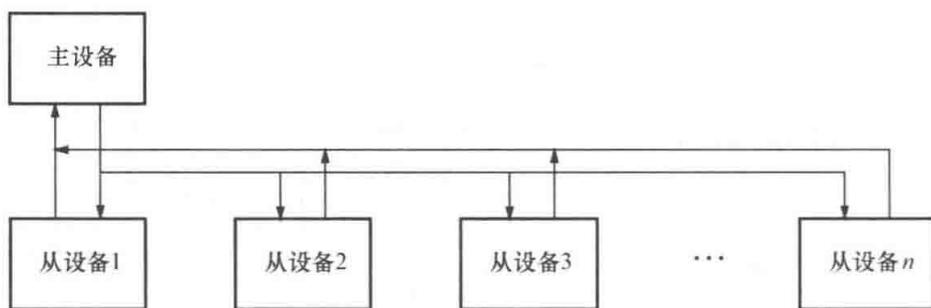


图 7-4 简单网络

#### 2. 多级网络

现代大型工业企业中, 一般采用多级网络的形式, 可编程序控制器制造商经常用生产金字塔结构来描述其产品可实现的功能。这种金字塔结构的特点是: 上层负责生产管理, 底层负责现场检测与控制, 中间层负责生产过程的监控与优化。

国际标准化组织 (ISO) 对企业自动化系统确立了初步的模型, 如图 7-5 所示。

#### 3. 通信协议

(1) 通用协议。国际标准化组织 ISO (International Standard Organization) 于 1978 年

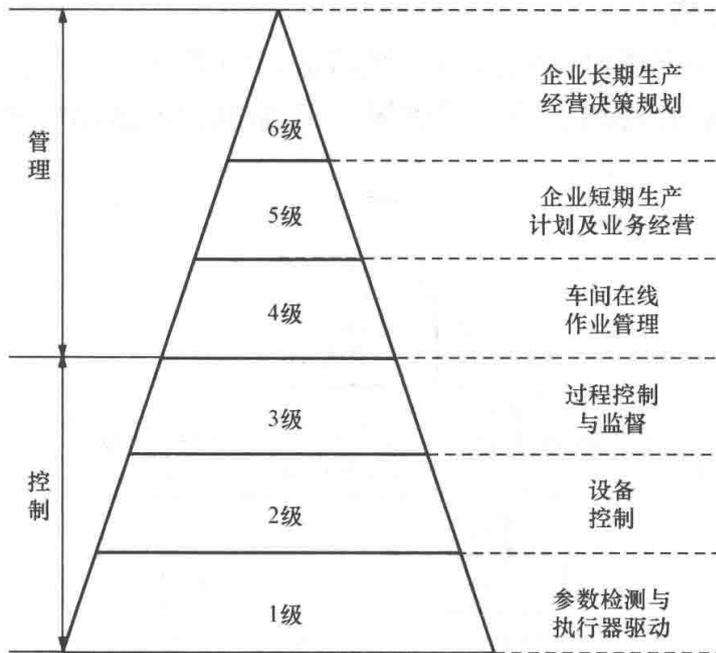


图 7-5 多级网络模型

提出通信协议，它所用的通信协议一般为 7 层，如图 7-6 所示。

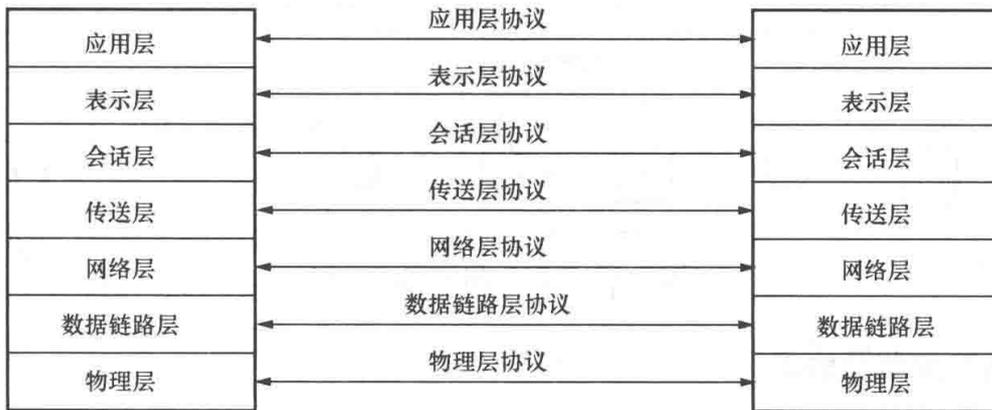


图 7-6 开放系统互联 OSI (Open Systems Interconnection) 的模型

(2) 公司专用协议。低层子网和中层子网一般采用公司专用协议，尤其是最底层子网，由于传送的是过程数据及控制命令，这种信息较短，但实时性要求高。公司专用协议的层次一般只有物理层、链路层及应用层，而省略了通用协议所必需的其他层，信息传送速率快。

## 7.2 S7 - 200 PLC 的通信控制网络

### 7.2.1 字符数据格式

#### 1. 10 位字符数据

传送数据由 1 个起始位、8 个数据位、无校验位、1 个停止位组成。传送速率一般为 9600bit/s。

#### 2. 11 位字符数据

传送数据由 1 个起始位、8 个数据位、1 个偶校验位、1 个停止位组成。传送速率一般为 9600bit/s 或 19200bit/s。

### 7.2.2 网络层次结构

西门子公司生产金字塔网络结构由 4 级组成，由下到上依次是：过程测量与控制级、过程监控级、工厂与过程管理级、公司管理级。S7 系列的网络结构如图 7-7 所示。

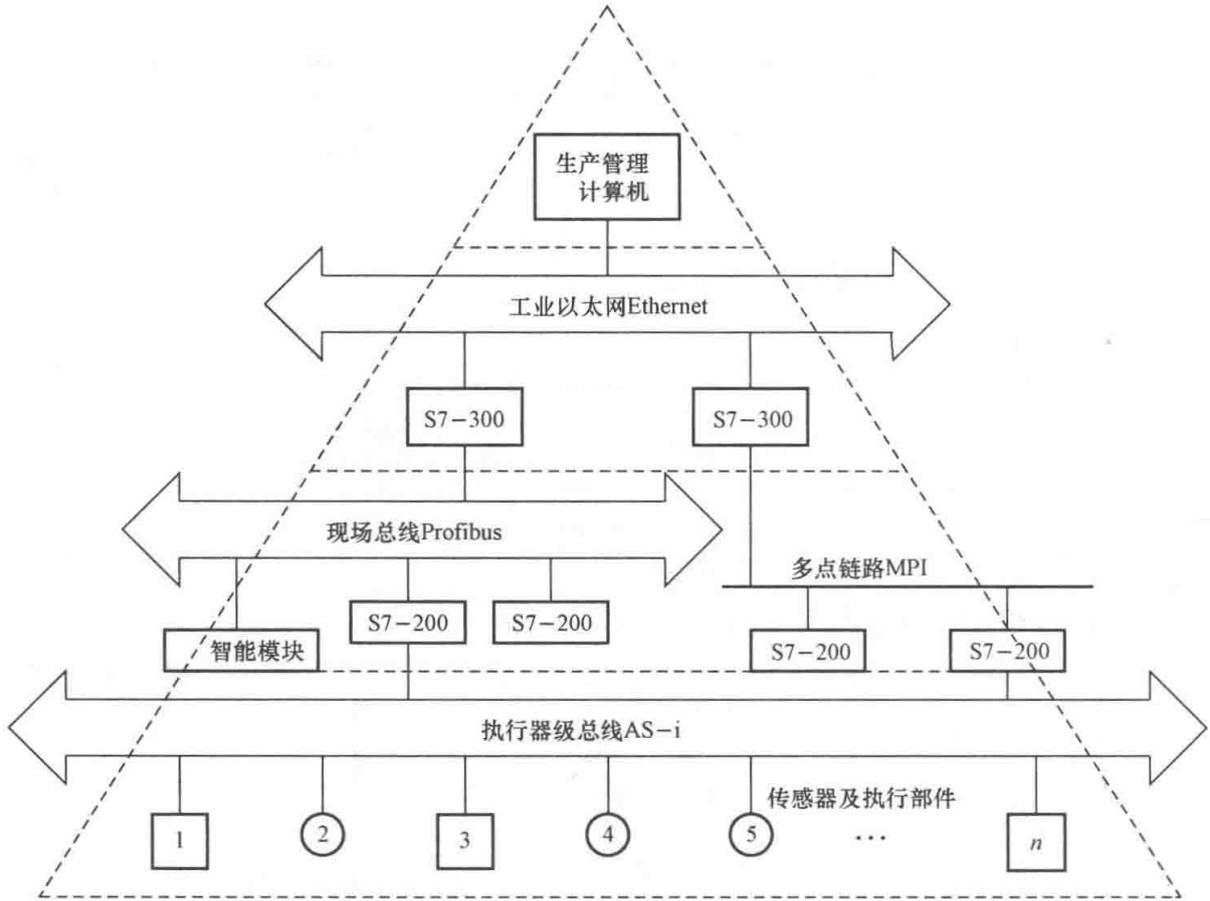


图 7-7 S7 系列的网络结构

### 7.2.3 通信类型及协议

#### 1. 通信类型

S7-200 PLC 的控制网络常见的有以下类型：把计算机或编程器作为主站、把操作员界面作为主站和把 PLC 作为主站等类型，这几种类型又各有两种连接：单主站和多主站。单主站的网络连接见图 7-8，多主站的网络连接见图 7-9。

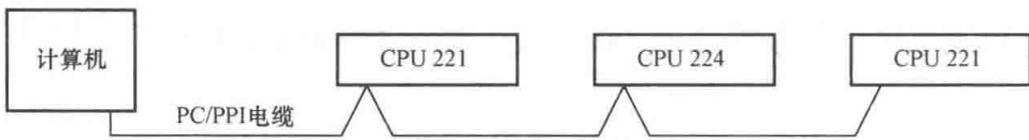


图 7-8 单主站的网络连接

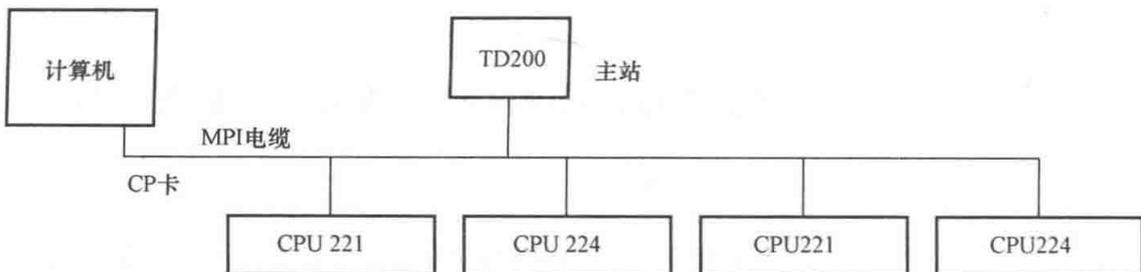


图 7-9 多主站的网络连接

## 2. 通信协议选择

S7-200 CPU 所支持的协议有点对点接口 (PPI)、多点接口 (MPI)、PROFIBUS。

在开放系统互联 (OSI) 七层模式通信结构的基础上, 这些通信协议在一个令牌环网络上实现。令牌环网络符合欧洲标准 EN 50170 中定义的 PROFIBUS 标准。这些协议是非同步的字符协议, 有 1 位起始位、8 位数据位、偶校验位和 1 位停止位。通信结构依赖于特定的起始字符和停止字符、源和目的地地址、报文长度和数据校验和。在波特率一致的情况下, 这些协议可以同时在一个网络上运行, 并且互不干扰。

如果带有扩展模块 CP243-1 和 CP243-1 IT, 那么 S7-200 也能运行在以太网上。

(1) PPI 协议。PPI 是一种主从协议。主站器件发送要求到从站器件, 从站器件响应。从站器件不发信息, 只是等待主站的要求并对要求作出响应。如图 7-10 所示。

主站靠一个 PPI 协议管理的共享连接来与从站通信。

PPI 并不限制与任意一个从站通信的主站数量, 但是在一个网络中, 主站的个数不能超过 32。

如果在用户程序中使能 PPI 主站模式, S7-200 CPU 在运行模式下可以作主站。在使能 PPI 主站模式之后, 可以使用网络读写指令来读写另外一个 S7-200。当 S7-200 作 PPI 主站时, 它仍然可以作为从站响应其他主站的请求。

PPI 高级允许网络设备建立一个设备与设备之间的逻辑连接。对于 PPI 高级, 每个设备的连接个数是有限制的。S7-200 支持的连接个数如表 7-1 所示。

所有的 S7-200 CPU 都支持 PPI 和 PPI 高级协议, 而 EM277 模块仅仅支持 PPI 高级协议。

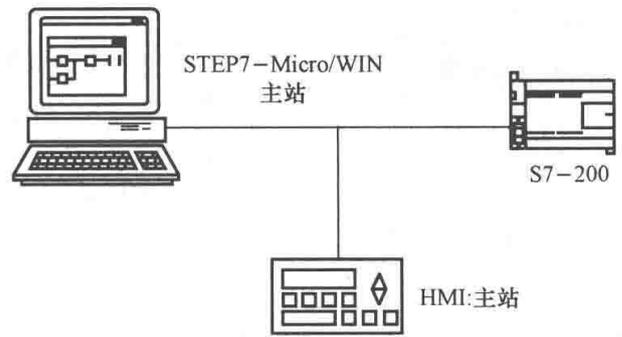


图 7-10 PPI 网络

表 7-1 S7-200 CPU 和 EM277 模块的连接个数

模 块	波特率 (bit/s)	连接数
S7-200 CPU 通信口 0 通信口 1	9.6k、19.2k 或 187.5k	4
	9.6k、19.2k 或 187.5k	4
EM277	9.6k 到 12M	6 (每个模块)

(2) MPI 协议。MPI 允许主主通信和主从通信, 见图 7-11。与一个 S7-200 CPU 通

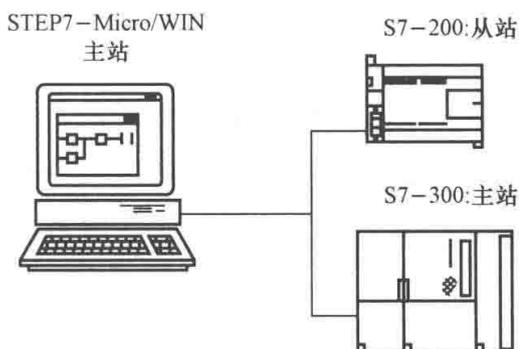


图 7-11 MPI 网络

信, STEP 7-Micro/WIN 建立主从连接。MPI 协议不能与作为主站的 S7-200 CPU 通信。

网络设备通过任意两个设备之间的连接通信 (由 MPI 协议管理)。设备之间通信连接的个数受 S7-200 CPU 或者 EM277 模块所支持的连接个数的限制。

S7-200 支持的连接个数如表 7-1 所示。

对于 MPI 协议, S7-300 和 S7-400 PLC 可以用 XGET 和 XPUT 指令来读写 S7-200 的数据。

(3) PROFIBUS 协议。PROFIBUS 协议通常用于实现与分布式 I/O (远程 I/O) 的高速通信。可以使用不同厂家的 PROFIBUS 设备。这些设备包括简单的输入或输出模块、电机控制器和 PLC。

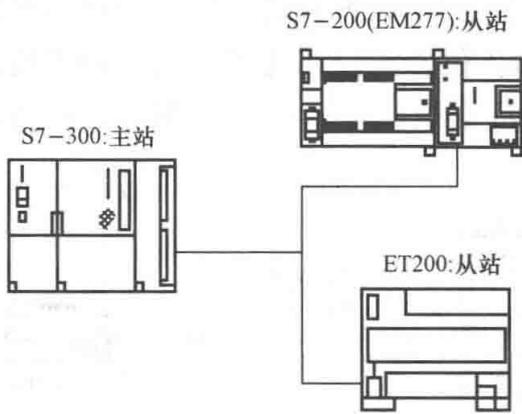


图 7-12 PROFIBUS 网络

PROFIBUS 网络通常有一个主站和若干个 I/O 从站, 如图 7-12 所示。主站器件通过配置可以知道 I/O 从站的类型和站号。主站初始化网络使网络上的从站器件与配置相匹配。主站不断地读写从站的数据。

当一个 DP 主站成功配置了一个 DP 从站之后, 它就拥有了这个从站器件。如果在网上有第二个主站器件, 那么它对第一个主站的从站的访问将会受到限制。

(4) TCP/IP 协议。通过以太网扩展模块 (CP243-1) 或互联网扩展模块 (CP243-1 IT), S7-200 将能支持 TCP/IP 以太网通信。表 7-2 列出了这些模块所支持的波特率和连接数。

表 7-2 以太网模块 (CP243-1) 和互联网模块 (CP243-1 IT) 的连接数

模块	波特率 (bit/s)	连接数
以太网 (CP243-1) 模块 互联网 (CP243-1 IT) 模块	10~100M	8 个普通连接 1 个 STEP 7 - MicroMIN 连接

更多信息可参考 SIMATIC NET CP243-1 工业以太网通信处理器手册或 SIMATIC NETCP243-1 IT 工业以太网及信息技术通信处理器手册。

(5) AS-Interface 协议。利用扩展模块 CP243-2, S7-200 PLC 可以作为传感器—执行器接口网络的主站, 读写从站的数据。

(6) 自由口协议。S7-200 CPU 的通信口还提供了建立在自由串行通信基础上的自由通信能力, 数据传输协议完全由用户程序决定。通过自由口方式, S7-200 PLC 可以与串行打印机、条码阅读器等通信。

#### 7.2.4 通信设备

##### 1. 通信口

S7-200 CPU 主机上的通信口是符合欧洲标准 EN 50170 中 Profibus 标准的 RS-485 兼容 9 针 D 型接口。接口引脚如图 7-13 所示, 端口 0 或端口 1 的引脚与 Profibus 的名称对应关系如表 7-3 所示。

##### 2. 网络连接器

网络连接器可以用来把多个设备很容易地连接到网络中。网络连接器有两种类型: 一种是仅提供连接到主机的接口, 另一种增加了一个编程接口。带有编程口的连接器可以把编程器或操作员面板直接增加到网络中, 编程口传递主机信号的同时, 为这些设备提供电源, 而不用另加电源。

##### 3. 网络中继器

网络中继器在 Profibus 可以用来延长网络的距离、允许给网络加入设备、隔离不同网

络段，每个中继器为网络段提供偏置和终端匹配。

表 7-3 对应关系表

针号	端口 0/端口 1	Profibus 名称
1	逻辑地	屏蔽
2	逻辑地	24V 返回
3	RS-485 信号 B	RS-485 信号 B
4	RTSTTL	发送申请
5	逻辑地	5V 返回
6	+5V, 100Ω, 串联电阻	+5V
7	+24V	+24V
8	RS-485 信号 A	RS-485 信号 A
9	10 位协议选择	不用
端口外壳	机壳接地	屏蔽

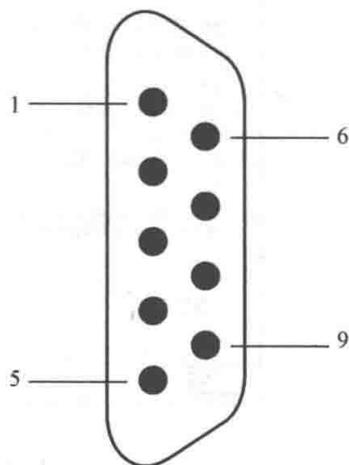


图 7-13 接口引脚图

每个网络中最多可以有 9 个中继器，每个中继器最多可再增加 32 个设备。

#### 4. 调制解调器

用调制解调器可以实现计算机或编程器与 PLC 主机之间的远距离通信。以 11 位调制解调器为例，通信连接示意图如图 7-14 所示。

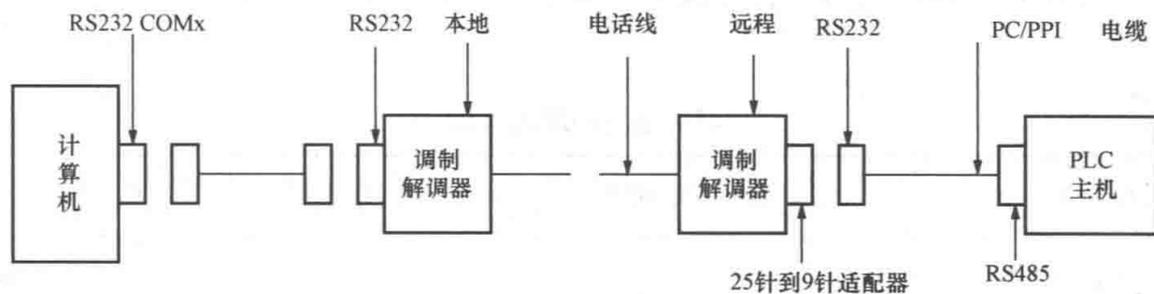


图 7-14 调制解调器的通信连接示意图

#### 5. 通信接口的安装和删除

在 Set PG/PC Interface 对话框中，可以使用安装/删除接口对话框来安装或者删除计算机上的通信接口，如图 7-15 所示。

(1) 在 Set PG/PC Interface 对话框中，点击 Select，弹出安装/删除接口对话框。选择框中列出了可以使用的接口，安装框中显示计算机上已经安装了的接口。

(2) 要添加一个接口：选择计算机上已经安装了的通信硬件并点击安装。当关闭安装/删除接口对话框后，Set PG/PC Interface 对话框中会在 Interface Parameter Assignment Used 框中显示接口。

(3) 要删除一个接口：选择要删除的接口并点击删除。当关闭安装/删除接口对话框后，Set PG/PC Interface 对话框中会在 Interface Parameter Assignment Used 框中删除该接口。

### 7.2.5 网络的建立

#### 1. 基本原则

(1) 导线必须安装合适的浪涌抑制器，这样可以避免雷击浪涌。

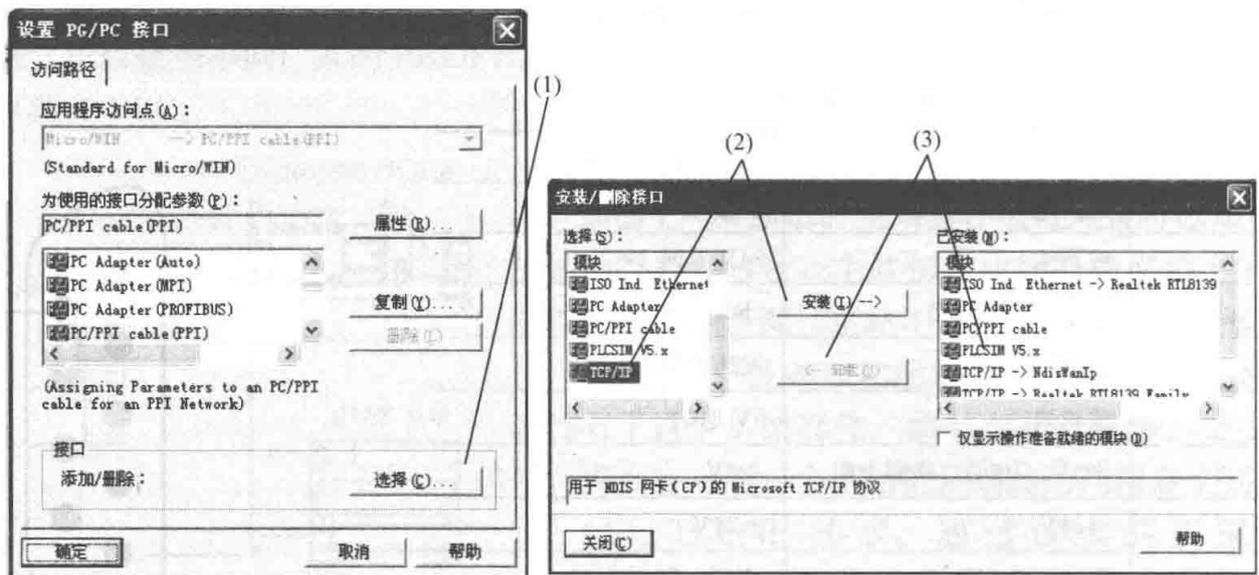


图 7-15 Set PG/PC Interface 和安装/删除接口对话框

(2) 应避免将低压信号线和通信电缆与交流导线和高能量、快速开关的直流导线布置在同一线槽中。要成对使用导线，用中性线或公共线与能量线或信号线配对。

(3) S7-200 CPU 的端口是不隔离的。如果想使网络隔离，应考虑使用 RS-485 中继器或者 EM277。

(4) 为网络确定通信距离、通信速率和电缆类型。

如表 7-4 所示，网段的最大长度取决于两个因素：隔离（使用 RS-485 中继器）和波特率。

表 7-4 网络电缆的最大长度

波特率 (bit/s)	非隔离 CPU 端口 1	有中继器的 CPU 端口或者 EM277
9.6k~187.5k	50m	1000m
500k	不支持	400m
1M~1.5M	不支持	200m
3M~12M	不支持	100m

当连接具有不同地电位的设备时需要隔离。当接地点之间的距离很远时，有可能具有不同的地电位。即使距离较近，大型机械的负载电流也能导致地电位不同。

## 2. 在网络中使用中继器

RS-485 中继器为网段提供偏压电阻和终端电阻。中继器有以下用途：

(1) 增加网络的长度。在网络中使用一个中继器可以使网络的通信距离扩展 50m。如图 7-16 所示，如果在已连接的两个中继器之间没有其他节点，那么网络的长度将能达到波特率允许的最大值。在一个串联网络中，最多可以使用 9 个中继器，但是网络的总长度不能超过 9600m。

(2) 网络增加设备。在 9600 的波特率下，50m 距离之内，一个网段最多可以连接 32 个

设备。使用一个中继器允许在网络上再增加 32 个设备。

(3) 实现不同网段的电气隔离。如果不同的网段具有不同的地电位, 将它们隔离会提高网络的通信质量。

一个中继器在网络中被算作网段的一个节点, 尽管如此, 它没有被指定站地址。

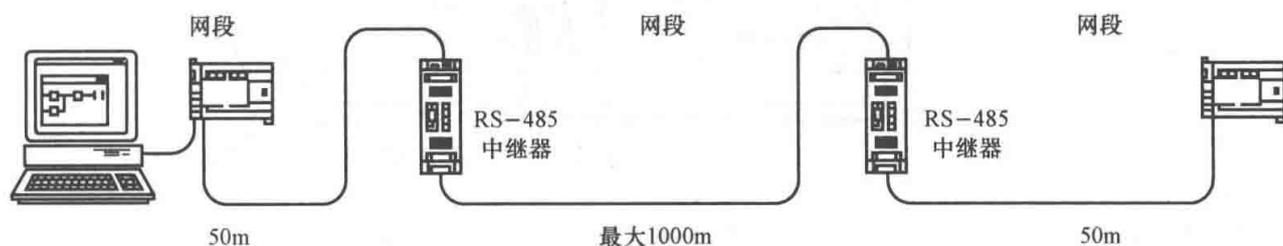


图 7-16 带中继器的网络举例

### 3. 选择网络电缆

S7-200 网络使用 RS-485 标准, 使用双绞线电缆。表 7-5 中列出了网络电缆的技术指标。每个网段中最多只能连接 32 个设备。

表 7-5 网络电缆的通信技术标准

技术指标	描述	技术指标	描述
电缆类型	屏蔽双绞线	衰减	0.9dB/100m (频率为 200MHz)
回路阻抗	$\leq 115\Omega/\text{km}$	导线截面积	0.3~0.5mm <sup>2</sup>
有效电容	30pF/m	电缆直径	8mm±0.5mm
标称阻抗	大约 135~160Ω (频率为 3~20MHz)		

### 4. 网络电缆的偏压电阻和终端电阻

为了能够把多个设备很容易地连接到网络中, 西门子公司提供两种网络连接器: 一种标准网络连接器和一种带编程接口的连接器, 后者允许在不影响现有网络连接的情况下, 再连接一个编程站或者一个 HMI 设备到网络中。带编程接口的连接器将 S7-200 的所有信号 (包括电源引脚) 传到编程接口。这种连接器对于那些从 S7-200 取电源的设备 (如 TD200) 尤为有用。

两种连接器都有两组螺钉连接端子, 可以用来连接输入连接电缆和输出连接电缆。两种连接器也都有网络偏置和终端匹配的选择开关。典型的网络连接器偏置和终端如图 7-17 所示。电缆的两个末端必须有终端和偏置。

### 5. 为网络选择 PPI 多主站电缆或 CP 卡

STEP 7-Micro/WIN 支持多种 CP 卡以及 RS-232/PPI 多主站电缆和 USB/PPI 多主站电缆, 并允许编程站 (计算机或 SIMATIC 编程器) 作为网络的主站。

当波特率小于等于 187.5k 时, PPI 多主站电缆能以最简单和经济的方式将 STEP 7-Micro/WIN 连接到 S7-200 CPU 或 S7-200 网络。PPI 多主站电缆有两种类型, 它们都能将 STEP 7-Micro/WIN 连接到 S7-200 网络。

USB/PPI 多主站电缆是一种即插即用设备, 适用于支持 USB1.1 版的 PC。当在 187.5k 的波特率下进行 PPI 通信时, 它能将 PC 和 S7-200 网络隔离。此时, 无需设置任何开关, 只需连上电缆, 将 PPI 电缆设为接口并选用 PPI 协议, 然后在 PC 接连标签下设置好 USB

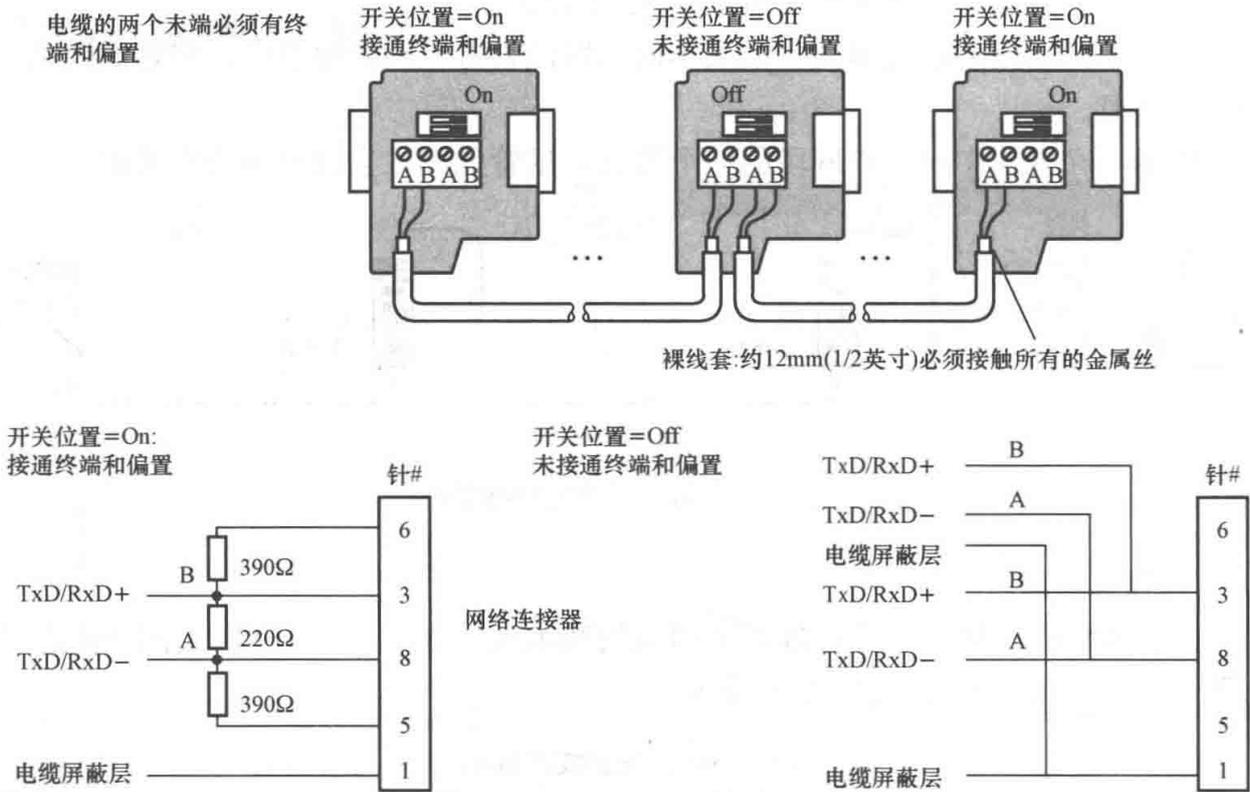


图 7-17 典型的网络连接器偏置和终端

端口即可。但在使用 STEP 7 - Micro/WIN 时，不能同时将多根 USB/PPI 多主站电缆连接到 PC 上。

RS-232/PPI 多主站电缆带有 8 个 DIP 开关，其中两个是用来配置电缆，使之可以用于 STEP7 - Micro/WIN 的。

如果需要将电缆连到 PC 上，则需选择 PPI 模式（开关 5=1）和本地操作（开关 6=0）。

如果需要将电缆连在调制解调器上，则需选用 PPI 模式（开关 5=1）和远程操作（开关 6=1）。

该电缆能将 PC 和 S7-200 网络隔离。要实现此功能，需将 PPI 电缆设为接口，并在 PC 接连标签下设置好 RS-232 端口。然后在 PPI 标签下，选定站地址和网络波特率。这时，协议将根据 RS-232/PPI 多主站电缆自动调整，因此无需再做更多的设置。

USB/PPI 多主站电缆和 RS-232/PPI 多主站电缆都带有 LED，用来指示 PC 或网络是否在进行通信。Tx LED 用来指示电缆是否正在将信息传送给 PC。Rx LED 用来指示电缆是否在接收 PC 传来的信息。而 PPI LED 则用来指示电缆是否在网络上传输信息。由于多主站电缆是令牌持有方，因此，当 STEP 7 - Micro/WIN 发起通信时，PPI LED 会保持点亮。而当与 STEP 7 - Micro/WIN 的连接断开时，PPI LED 会关闭。在等待加入网络时，PPI LED 也会闪烁，其频率为 1Hz。

CP 卡为编程站管理多主网络提供了硬件，并且支持多种波特率下的不同协议，如表 7-6 所示。每一块 CP 卡为网络连接提供了一个单独的 RS-485 接口。CP 5511 PCMCIA 卡有一个提供 9 针 D 型接口的适配器。可以将通信电缆的一端接到 CP 卡的 RS-485 接口上，另一端接入网络。

如果通过 CP 卡建立 PPI 通信，那么，STEP 7 - Micro/WIN 将无法支持在同一块 CP 卡上同时运行两个应用。在通过 CP 卡将 STEP 7 - Micro/WIN 连接到网络之前，必须关掉

另外一种应用。如果使用的是 MPI 或 PROFIBUS 通信,那么将允许多个 STEP7 - Micro/WIN 应用在网络上同时进行通信。

表 7-6 STEP - Micro/WIN 支持的 CP 卡和协议

配置	波特率 (bit/s)	协议
RS-232/PPI 多主站或 USB/PPI 多主站电缆	9.6k~187.5k	PPI
CP5511 PCMCIA 卡 (适用于笔记本电脑)	9.6k~12M	PPI、MPI 和 PROFIBUS
CP5512 PCMCIA 卡 (适用于笔记本电脑)	9.6k~12M	PPI、MPI 和 PROFIBUS
CP5611 (版本 3 以上) PCI 卡	9.6k~12M	PPI、MPI 和 PROFIBUS
CP1613 S7613 PCI 卡	10M 或 100M	TCP/IP
CP1612 SoftNet7 PCI 卡	10M 或 100M	TCP/IP
CP1512 SoftNet7 PCMCIA 卡 (适用于笔记本电脑)	10M 或 100M	TCP/IP

### 6. 在网络中使用 HMI 设备

S7-200 CPU 支持西门子公司的多种 HMI 设备,同时也支持其他厂家的产品。有些 HMI 设备(如 TD 200 或 TP070)不允许选择此设备所使用的通信协议,而另一些(如 OP7 和 TP170)则允许用户选择它们所用的通信协议。

如果 HMI 设备允许选择通信协议,应考虑以下原则:

- (1) 对于直接连接在 S7-200 CPU 通信端口上的 HMI 设备,如果网络上没有其他设备,既可以选择 PPI 协议,又可以选择 MPI 协议。
- (2) 对于连接在 EM277 模块上的 HMI 设备,可以选择 MPI 或 PROFIBUS。
- (3) 如果网络中有 S7-300 或 S7-400 PLC,为 HMI 设备选择 MPI 协议。
- (4) 如果 HMI 设备连接在一个 PROFIBUS 网络中,为 HMI 设备选择 PROFIBUS 协议与其他主站相兼容。
- (5) 如果 HMI 设备所连接的 S7-200 CPU 已经被配置为主站,为 HMI 设备选择 PPI,最好使用 PPI 高级。MPI 和 PROFIBUS 协议不支持 S7-200 CPU 作主站。

## 7.3 网络读写

S7-200 CPU 提供网络读写指令,用于 S7-200 CPU 之间的联网通信。网络读写指令只能在网络中充当主站的 CPU 执行;从站 CPU 不必作通信编程,只需准备通信数据。主站可以对 PPI 网络中的其他任何 CPU (包括主站)进行网络读写。

### 7.3.1 网络指令

网络指令有两条:NETR 和 NETW。它们的梯形图指令盒如图 7-18 所示。

#### 1. NETR 指令

NETR,网络读指令。使能输入有效时,指令初始化通信操作,通过通信端口 PORT 从远程设备上接收数据并形成数据表 TBL。

NETR 指令最多可从远程站点上读 16 个字节的

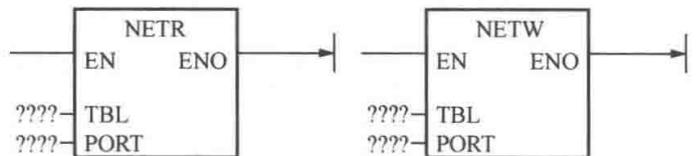


图 7-18 网络读写指令的指令盒形式

指令格式: NETR TBL, PORT

## 2. NETW 指令

NETW 网络读指令。使能输入有效时, 指令初始化通信操作, 通过端口 PORT 将数据表 TBL 中的数据发送到从远程设备。

NETW 指令最多可向远程站点上写 16 个字节的信。

指令格式: NETW TBL, PORT

### 7.3.2 控制寄存器和传送数据表

#### 1. 控制寄存器

SMB30 控制和设置通信端口 0, 如果 PLC 主机上有通信端口 1, 则用 SMB130 来进行控制和设置。SMB30 和 SMB130 的各位及其的含义如下:

P	P	D	B	B	B	M	M
---	---	---	---	---	---	---	---

(1) PP 位: 奇偶选择。

(2) D 位: 有效位数。

(3) BBB 位: 自由口波特率。

(4) MM 位: 协议选择, 即 MM 用来决定相应通信口的工作模式, 其中:

MM=00: PPI 从站模式 (默认设置为 PPI/从站模式)。

MM=01: 自由口模式。

MM=10: PPI 主站模式。

所以, 只要向 SMB30 或 SMB130 中写入数值 2 (即二进制的 10), 就可以将通信口设置为 PPI 主站模式。PPI 通信速率在“系统块”中设置, 在 PPI 模式下忽略 2~7 位。

#### 2. 传送数据表

(1) 数据表格式。执行网络读写指令时, PPI 主站与从站之间的数据以数据表的格式传送。传送数据表的格式如表 7-7 所示。

表 7-7 传送数据表格式

字节偏移地址	名称	描述
0	状态字节	反映网络指令的执行结果状态及错误码
1	远程站地址	被访问的从 PLC 站的地址
2	远程站的数据 的指针	被访问数据的间接指针; 指针可以指向 I、Q、M 和 V 数据区
3		
4		
5		
6	数据长度	过程站上被访问的数据的长度
7	数据字节 0	对 NETR 指令, 执行后, 从远程站读的数据放到这个数据区; NETW 指令, 执行前, 要发送到远程站的数据放到这个数据区
8	数据字节 1	
...	...	
22	数据字节 15	

(2) 状态字节。传送数据表中的第一个字节为状态字节, 各位及其的含义如下:

D	A	E	0	E1	E2	E3	E4
---	---	---	---	----	----	----	----

其中, D=1 表示操作已完成, D=0 表示操作未完成; A=1 表示操作有效, A=0 表示操作无效。E1、E2、E3、E4 为错误编码, 如果执行指令后 E 位为 1, 则由这 4 位返回一个错误码。这 4 位组成的错误编码及含义如表 7-8 所示。

表 7-8 错误编码及含义

E1 E2 E3 E4	错误码	说 明
0000	0	无错误
0001	1	时间溢出错误, 远程站点不响应
0010	2	接收错误: 奇偶校验错, 响应时帧或检查时出错
0011	3	离线错误: 相同的站地址或无效的硬件引发冲突
0100	4	队列溢出错误: 激活了超过 8 个 NETR 和 NETW 指令
0101	5	违反通信协议: 没有在 SMB30 中允许 PPI 协议而执行网络指令
0110	6	非法参数: NETR 和 NETW 指令中包含非法或无效的值
0111	7	没有资源: 远程站点正在忙中, 如上装或下装顺序正在处理中
1000	8	第 7 层错误, 违反应用协议
1001	9	信息错误: 错误的地址或不正确的数据长度
1010 - 1111	A - F	未用, 为将来的使用保留

### 【例 7-1】网络通信应用。

有一简单网络, 结构如图 7-19 所示。其中 PC 为编程电脑, 在 RUN 模式下, CPU 226 在用户程序中允许 PPI 主站模式, 可以利用 NETR 和 NETW 指令来不断读写 CPU 224 模块中的数据。

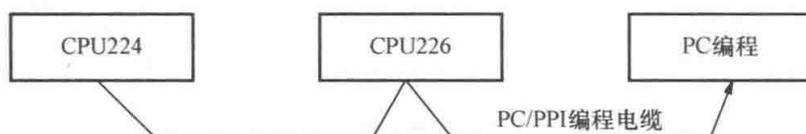


图 7-19 网络结构图

操作要求: 要求将主站的 I0.0~I0.7 的状态映射到从站的 Q0.0~Q0.7, 将从站的 I0.0~I0.7 的状态映射到主站的 Q0.0~Q0.7。

CPU226 通信端口号为 1, 接收缓冲区从 VB300 开始, 发送缓冲区从 VB200 开始, 内容如表 7-9 所示。CPU224 通信端口号为 0, 从 VB200 开始设置接收和发送缓冲区。

表 7-9 数 据 表

站号	发送缓冲区		接收缓冲区	
CPU226	VB200	状态字节	VB300	状态字节
	VB201	CPU224 的地址 (4)	VB301	CPU224 的地址 (4)
	VB202	&VB300, CPU224 的接收缓冲区地址	VB302	&VB200, CPU224 的发送缓冲区地址
	VB206	1, 数据长度字节数	VB306	1, 数据长度字节数
	VB207	数据字节 1	VB307	数据字节 1
CPU224	VB200	发送缓冲区字节 1	VB300	接收缓冲区字节 1

主站网络通信的程序如图 7-20 所示。

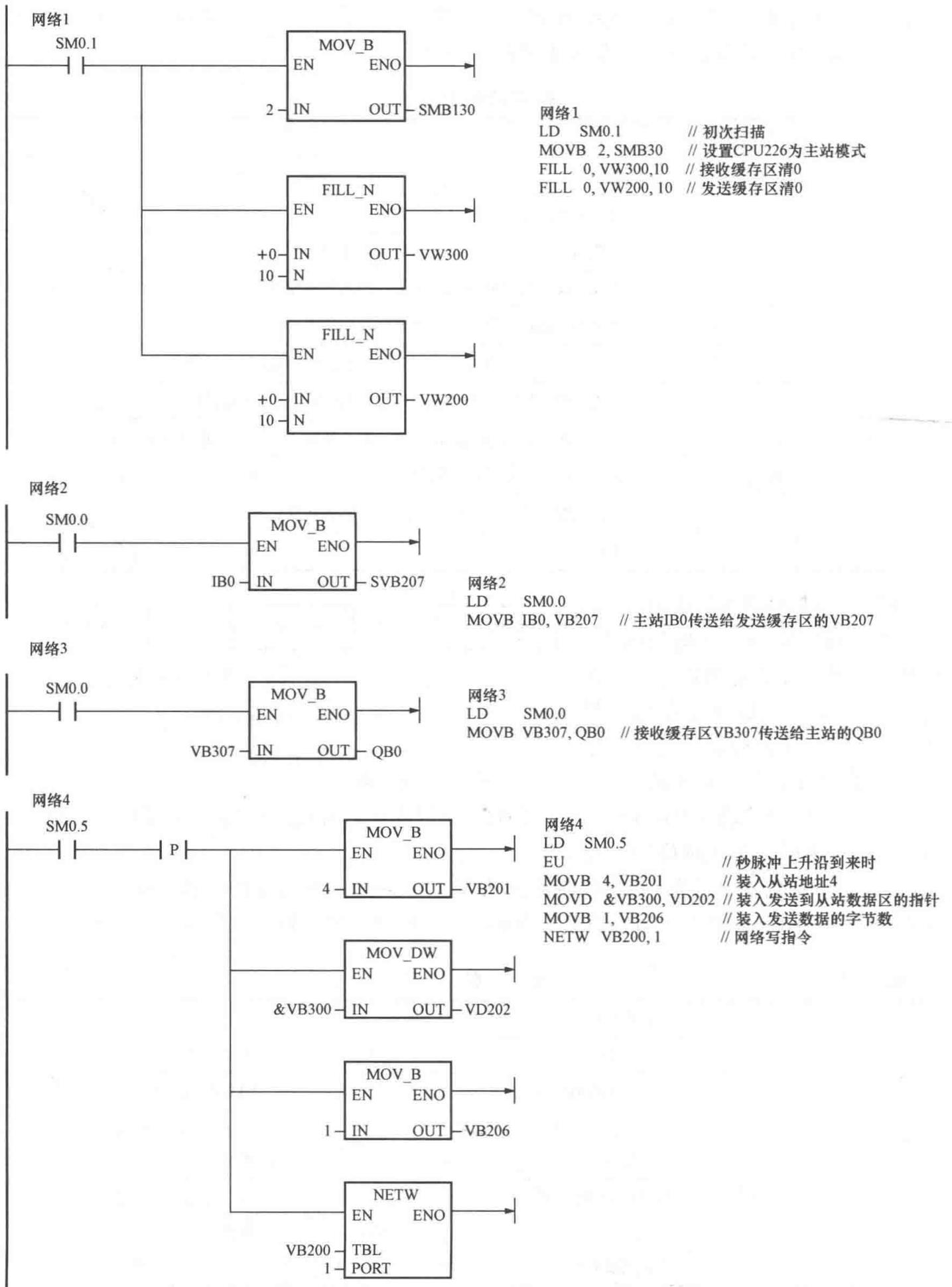


图 7-20 主站网络通信程序 (一)

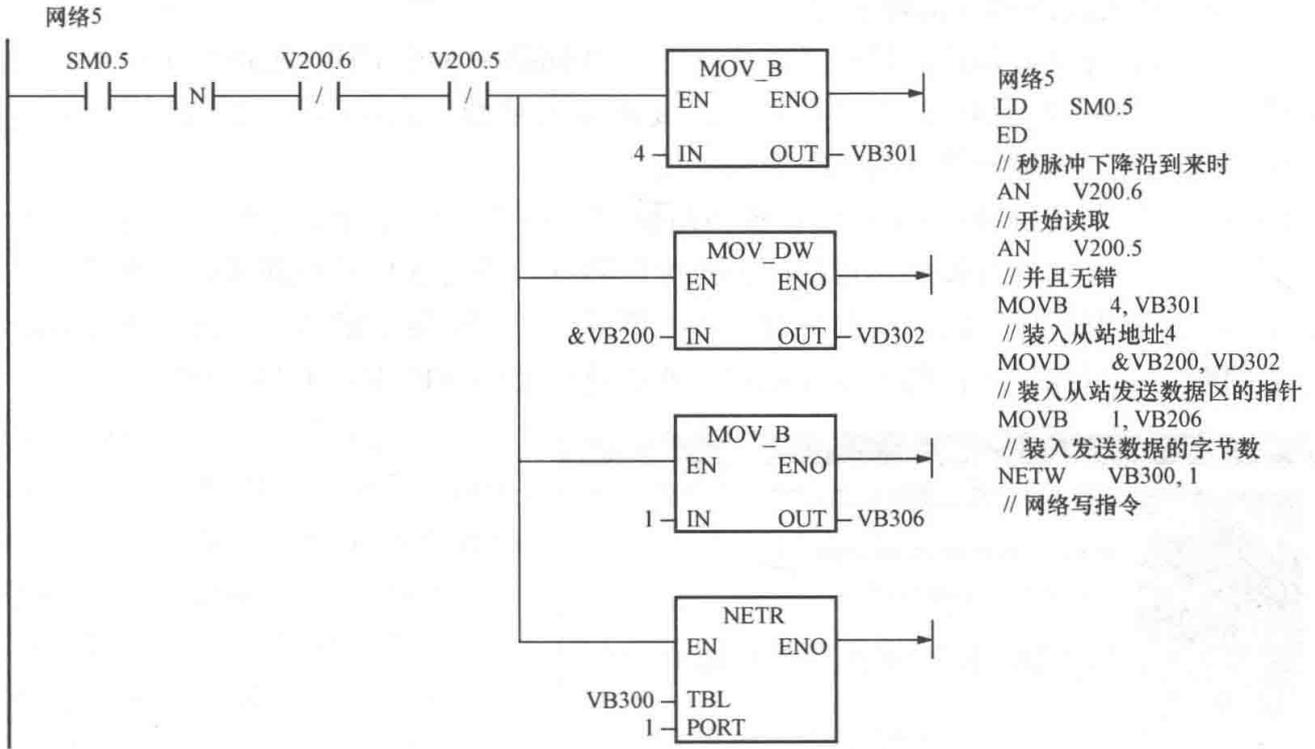


图 7-20 主站网络通信程序 (二)

从站通信程序如图 7-21 所示。

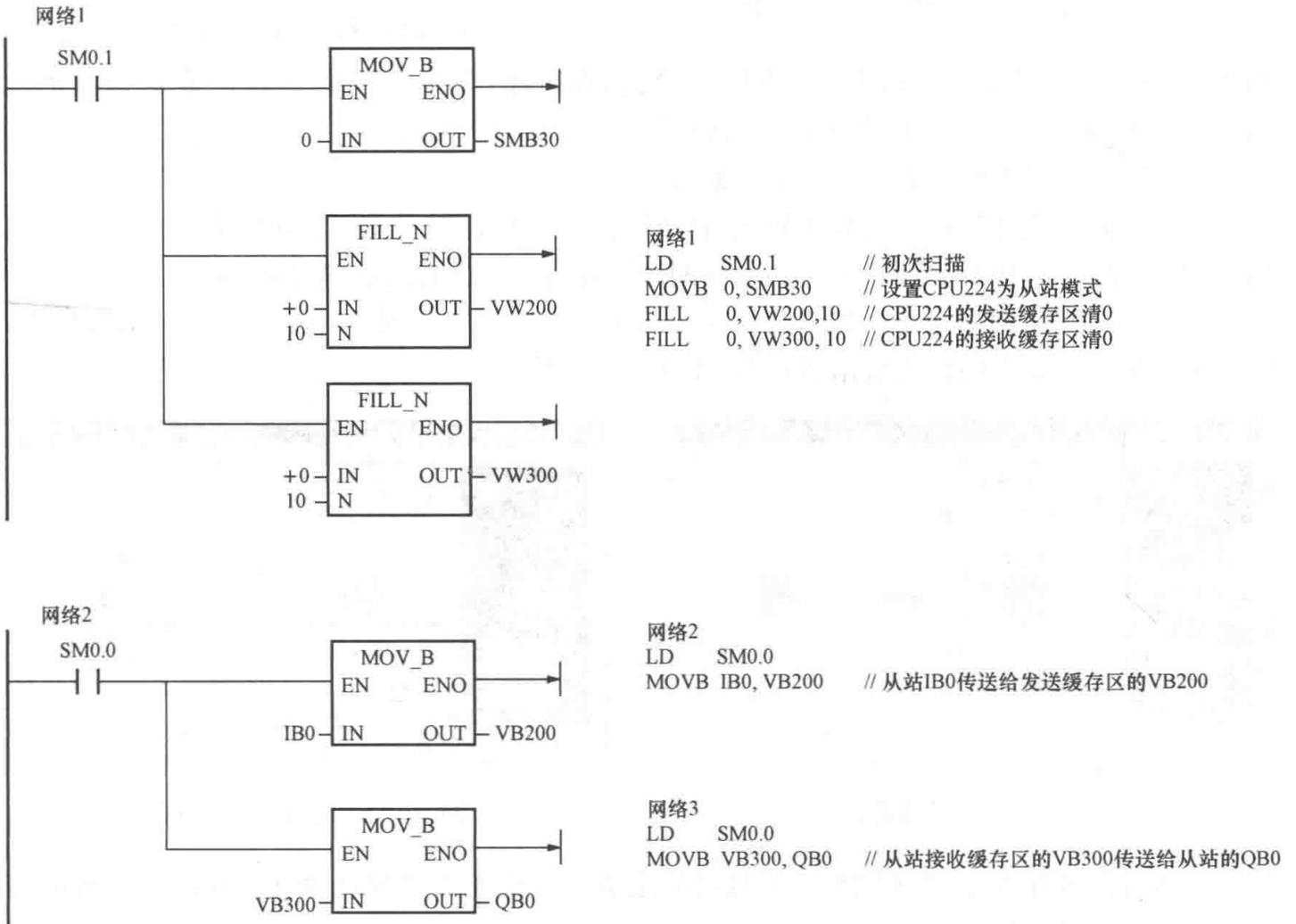


图 7-21 从站通信程序

### 7.3.3 使用网络读写向导编程

网络读写向导可以帮助用户自动生成一个 PPI 网络中多个 CPU 之间的通信指令，简化网络读写的编程步骤。用户只要按照向导的要求输入初始信息，CPU 之间的读写通信数据区，向导就会自动生成网络读写指令及数据块。

网络读写向导会自动将 CPU 设置成主站模式，用户不必另行编程设置。用户只需为主站编写通信程序，从站直接使用通信缓冲区中的数据，或将数据整理到通信区即可。向导的通信使用顺序控制程序，同一时刻只有一条 NETR/NETW 指令激活，并且对读写通信状态进行了判断，可以保证通信的可靠及稳定。所以建议使用 NETR/NETW 编程。

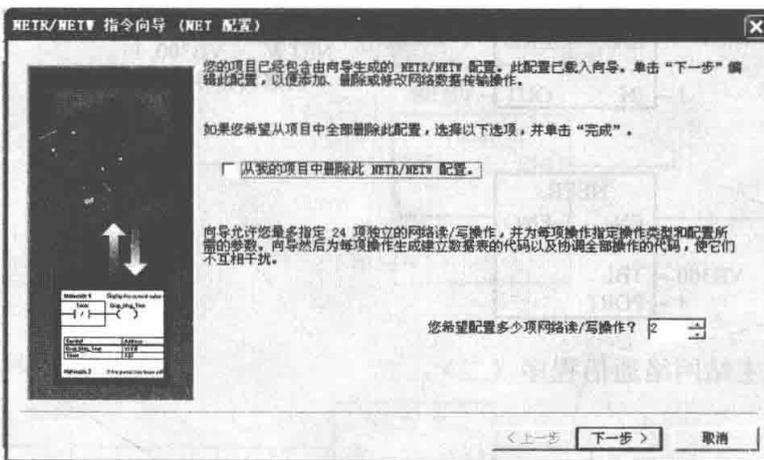


图 7-22 网络读写指令向导

在 STEP 7 - Micro/WIN 导航栏中的“Tools”中单击“指令向导”图标或者在命令菜单中选择“Tools”>“Instruction Wizard”，然后在指令向导窗口中选择“NETR/NETW”指令进入 NETR/NETW 向导。如图 7-22 所示。

下面结合具体的例子来说明网络通信向导的编程应用。

**【例 7-2】**网络通信向导指令的应用，设远程 PLC 的站地址为 4，本地 PLC 的站地址为 2，要求将远程站上 IB0 的内容通过 PPI 网络通信传输给本地 PLC 的输出 QB0。将主站的 I0.0~I0.7 的状态映射到从站的 Q0.0~Q0.7。

将主站的 I0.0~I0.7 的状态映射到从站的 Q0.0~Q0.7。

第一步：定义通信所需网络操作的数目。

第二步：定义通信口和向导生成的子程序名（可使用默认名）。对于有两个通信口的 CPU 可以选择 Port0 口或 Port1 口，所有网络操作将由定义的通信口完成。

第三步：定义网络操作。如图 7-23 所示为网络读操作。图 7-24 所示为网络写操作，每一个网络操作指令通信的数据最多为 16 字节。

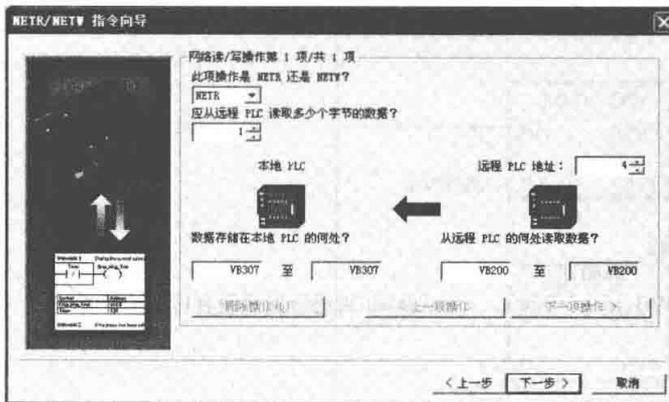


图 7-23 网络读操作设置

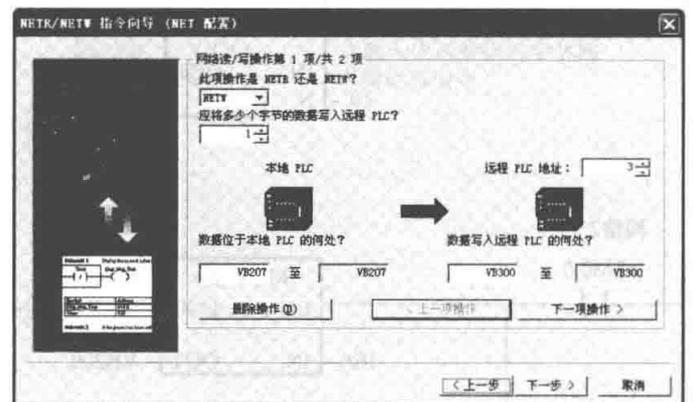


图 7-24 网络写操作设置

第四步：分配 V 存储区地址。可自己指定也可直接单击“建议地址”让向导分配程序中未用过的地址空间。

第五步：自动生成网络读写指令及符号表。在完成向导配置后，只需在 CPU 程序中调

用向导所生成的网络读写指令即可，利用网络通信向导指令编写的主站程序如图 7-25 所示。只能用 SM0.0 来调用 NETR/NETW 子程序，以保证它的正常运行。

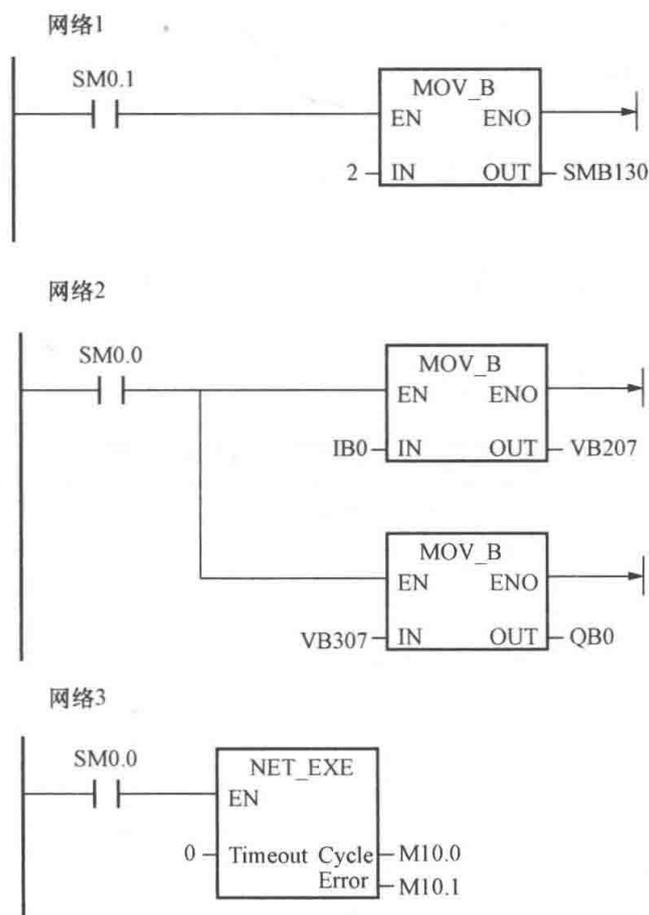


图 7-25 主站调用 NET\_EXE 子程序

PPI 从站的程序与非向导编程一样，如图 7-21 所示。

## 7.4 自由口通信技术

S7-200 CPU 拥有自由口通信能力。自由口通信是建立在半双工 RS-485 硬件基础上的一种通信方式，它允许用户自己定义一些简单、基本的字符通信格式，如数据长度和奇偶校验等。灵活运用自由口功能，可以编程适应比较复杂的帧格式，以实现各种通信协议。

处于自由口通信模式时，通信功能完全由用户程序控制，所有的通信任务必须由用户编程完成。

如果 S7-200 CPU 的某个通信端口工作在自由口模式下，它就不能用于其他模式的通信。例如，STEP 7-Micro/WIN 软件就不能通过一个定义为自由口模式的通信口与 CPU 通信。当 CPU 处于 STOP（停止）模式时，自由口便不能工作，从而可以建立正常的编程通信。

在 CPU 运行状态下，可以通过程序对通信口进行重新定义。例如，可以使用特殊寄存器位 SM0.7 来控制自由口模式，这样可以在 CPU 处于运行模式时，使用 STEP 7-Micro/WIN 软件监控。SM0.7 的状态取决于 CPU 上的模式开关的位置。

调试 S7-200 CPU 的自由口通信时，可以用 PC/PPI 电缆将 CPU 和 PC 机连接起来，在 PC 机上运行串口调试软件，如 Windows 操作系统集成的 Hyper Terminal（超级终端）应用程序（如果 Windows 中没有超级终端程序，可能需要添加安装 Windows 组件），向 CPU 发送数据，或从 CPU 接收数据。

### 7.4.1 相关寄存器及标志

#### 1. 控制寄存器

SMB30 控制自由端口 0 的通信方式, 如果 PLC 主机上有通信端口 1, 则用 SMB130 来控制自由端口 1 的通信方式。其含义如表 7-10 所示。

#### 2. 特殊标志位及中断

(1) 特殊标志位。SM4.5 和 SM4.6: 分别用来表示口 0 和口 1 发送空闲状态。

(2) 中断。

接收中断: 中断事件号为 8 (口 0) 和 25 (口 1)。

发送完成中断: 中断事件号为 9 (口 0) 和 26 (口 1)。

接收完成中断: 中断事件号为 23 (口 0) 和 24 (口 1)。

表 7-10 自由口模式控制字节各位的含义

MSB	自由端口控制字节							LSB									
			P	P	D	B	B	B	M	M							
7	6	5	4	3	2	1	0										
校验选择		每个字符的数据位	自由口波特率 (kbit/s)				协议选择										
00=不选择	01=偶校验	10=不校验	11=奇校验	0 = 8 位/字符	1 = 7 位/字符	000=38.4	001=19.2	010=9.6	011=4.8	100=2.4	101=1.2	110=115.2	111=57.6	00=PPI/从站模式	01=自由口协议	10=PPI/主站模式	11=保留

#### 3. 特殊存储器字节

接收信息时用到一系列特殊功能存储器。对端口 0 用 SMB86~SMB94; 对端口 1 用 SMB186~SMB194。各字节及内容描述如表 7-11 所示。

表 7-11 特殊存储器

端口 0	端口 1	说明
SMB86	SMB186	接收信息状态字节
SMB87	SMB187	接收信息控制字节
SMB88	SMB188	信息字符的开始
SMB89	SMB189	信息字符的结束
SMD90	SMD190	空闲时间段毫秒设定, 空闲后收到的第一个字符是新信息的首字符
SMD92	SMD192	中间字符定时器溢出值按毫秒设定, 超过这一时间则终止接收信息
SMB94	SMB194	要接收的最大字符数

### 7.4.2 自由口指令

#### 1. XMT 指令

XMT, 发送指令。使能输入有效时, 指令初始化通信操作, 通过通信端口 PORT 将数据表 TBL 中的数据发送到远程设备。发送指令允许 S7-200 的通信口上发送最多 255 个字节。发送中断通知程序发送完成。

发送指令执行完成后, 会产生一个中断事件 (Port0 为中断事件 9, 对 Port1 为中断事

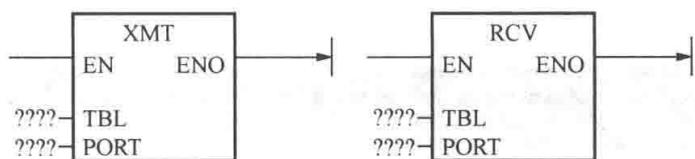


图 7-26 发送和接收指令的梯形图指令盒形式

形式如图 7-26 所示。

2. RCV 指令

RCV, 接收指令。使能输入有效时, 指令初始化通信操作, 通过通信端口 PORT 从远程设备上接收数据并放到缓冲区 (数据表) TBL。接收缓冲区 (数据表) TBL 的格式如表 7-12 所示。

接收指令从通信口接收整条信息, 当接收完成后会产生一个中断事件 (对 Port0 为中断事件 23, 对 Port1 为中断事件 24)。特殊寄存器 SMB86 (对 Port0) 和 SMB186 (对 Port1) 也提供了接收信息状态字节, 以使用户程序使用。

接收指令的梯形图指令盒形式如图 7-26 所示。

表 7-12 发送接收缓存区 TBL 的格式

发送缓冲区	接收缓冲区
发送字节个数	接收字节个数
字节 1	字节 1
字节 2	字节 2
...	...
字节 n	字节 m

7.4.3 自由口通信发送指令 (XMT) 应用

**【例 7-3】**把 CPU 224 的 Port0 定义为自由口通信模式。在一个定时中断程序中对定时中断次数计数, 并将计数值转换为 ASCII 字符串, 再从 Port0 发送出去。计算机接收并利用超级终端 (Hyper Terminal) 显示与 S7-200 CPU 的通信的内容。

自由口通信模式以字节为单位发送数据, 而不考虑其内容, 在这个例子中使用 ASCII 字符只是为了便于在 PC 机上显示。

1. 参数设置

自由口通信模式。通信协议为: 波特率 9600, 无奇偶校验, 每字符 8 位。规定发送缓冲区从 VB100 开始, 如表 7-13 所示。在本例中设置 16#0D0A 为结束字符, 是因为在 Hyper Terminal (超级终端) 中 16#0D0A 正好是字符“回车”, 可用来换行显示。

表 7-13 发送缓冲区数据定义

VB100	14	发送数据字节数
VB101~112	数据字节	ASCII 字符表 (共 12 字节长)
VB113	16#0D	消息结束字符
VB114	16#0A	即“回车”符

件 26)。在 SMB4 中也有相应的位对应于发送指令状态 (SM4.5 置位对应 Port0 空闲, SM4.6 对应 Port1 空闲)。

发送缓冲区 (数据表) TBL 的格式如表 7-12 所示。发送指令的梯形图指令盒

2. 程序

使用 Data Block (数据块) 定义发送缓冲区, 如图 7-27 所示。

本例 PLC 程序包括主程序、子程序 SBR\_1、SBR\_0 和中断子程序, 分别如图 7-28~图 7-31 所示。

主程序: 初始化自由口通信设置, 并根据“模式选择开关”的状态重新设置通信端口 0。SBR\_0:

定义通信端口 0 为自由口, 初始化定时中断。SBR\_1: 定义通信端口 0 为普通 PPI 从站通信口。INT\_0: 对定时中断计数并从 Port0 (端口 0) 发送计数值。

```

STEP 7-Micro/WIN - 第7章例7-3: 自由口通信的应用1 - [数据块]
文件(F) 编辑(E) 查看(V) PLC(P) 调试(D) 工具(T) 窗口(W) 帮助(H)
2 3 4 5 6 7 8 9 10 11 12 13 14 15
//数据块注释
//按F1键获取帮助和示范数据页
VB100 14 //发送字节数
VD200 0 //累加器清零
VB113 16#0D //发送数据的结束字符
VB114 16#0A //“回车”的16进制码
    
```

图 7-27 数据块

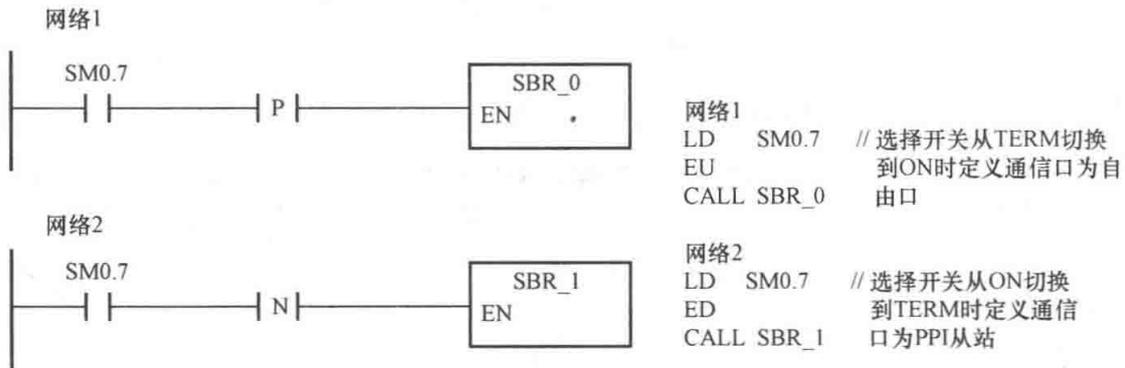


图 7-28 主程序

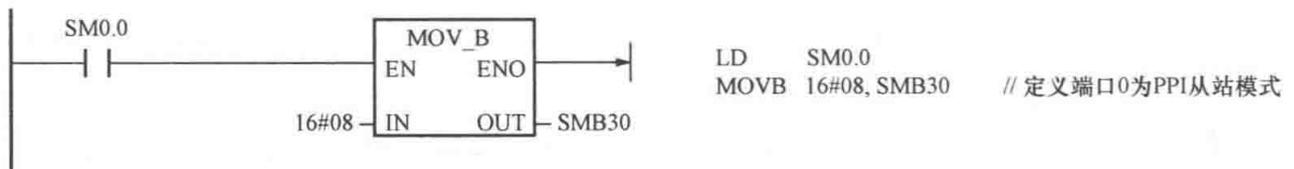


图 7-29 SBR\_1 编程

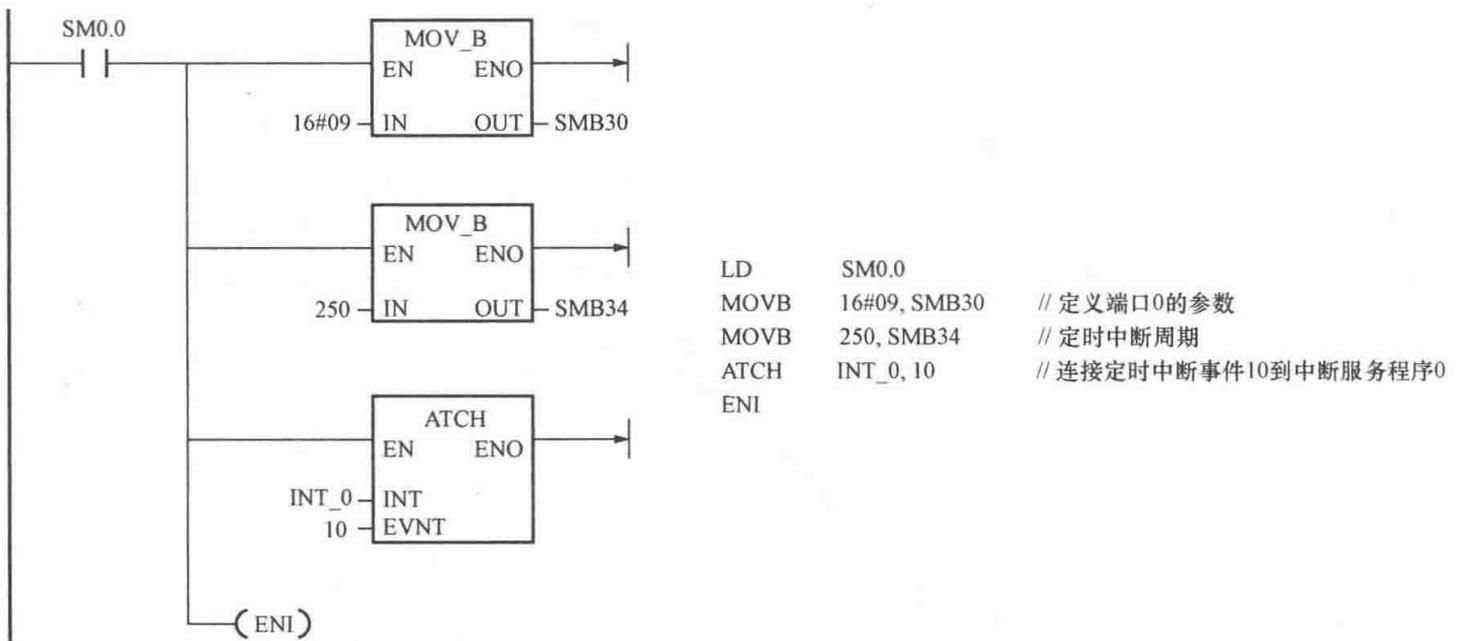


图 7-30 SBR\_0 编程

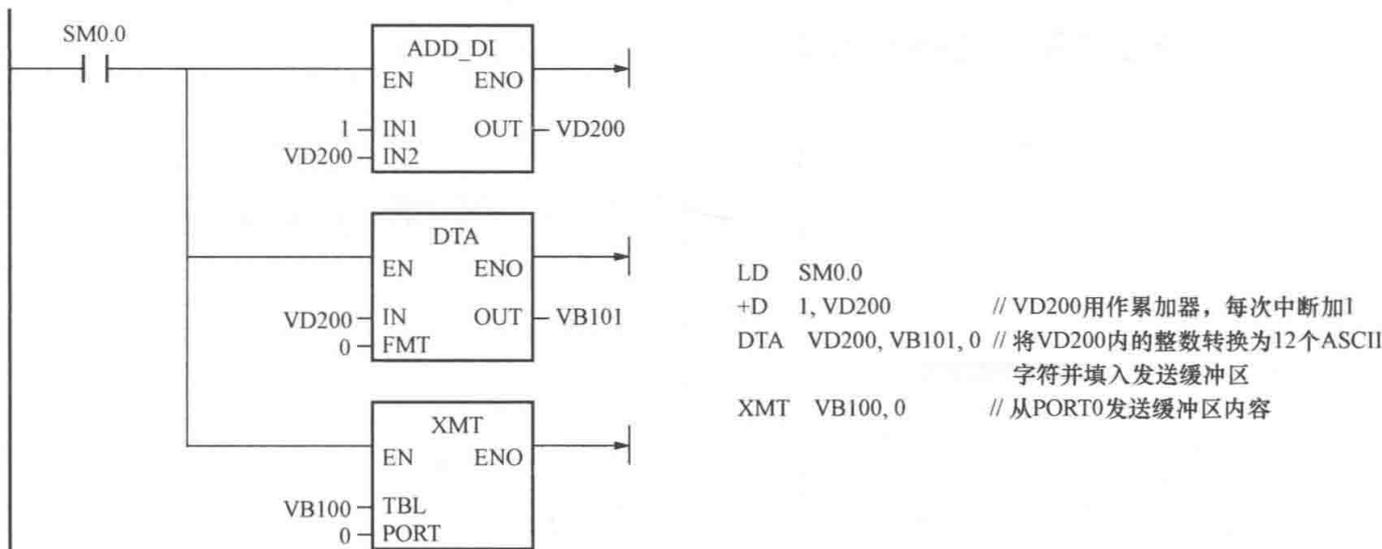


图 7-31 INT\_0 编程

### 3. 使用 Hyper Terminal 监视串口通信

如同编程时那样连接 PC/PPI 电缆。使用 Hyper Terminal 时需要注意不要让多个应用程序争夺串行通信口的控制权。

如果使用了多主站 RS-232/PPI 电缆，需将 DIP 开关 5 拨为“0”，并设置适当的通信速率。

打开 Windows 系统的 Hyper Terminal 程序，选择图标，指定一个连接名称，如图 7-32 所示。

选择 PC 机连接 PC/PPI 电缆的串行通信端口（这里是 COM1），如图 7-33 所示。



图 7-32 指定连接名称

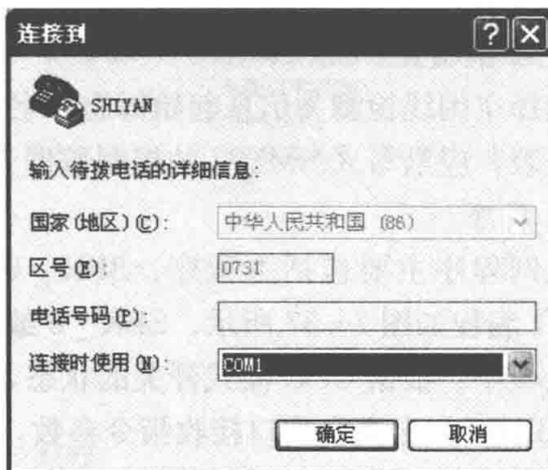


图 7-33 选择串行通信端口

选择通信口参数，如图 7-34 所示。在 Hyper Terminal（超级终端）窗口中应当显示由 S7-200 CPU 发送来的字符串，如图 7-35 所示。

#### 7.4.4 自由口通信接收指令 (RCV) 的应用

RCV 指令的所有控制均通过程序设置接收指令控制特殊寄存器完成。接收指令的启动、结束（或终止），以及消息起始和结束条件，都通过接收指令控制字节（SMB87 对 Port0，SMB187 对 PORT1）和其他一些控制特殊存储器设置。

##### 1. 接收指令的启动和结束（或中止）

启动接收指令后，S7-200 CPU 的通信控制器就处于接收状态。接收状态可以由用户

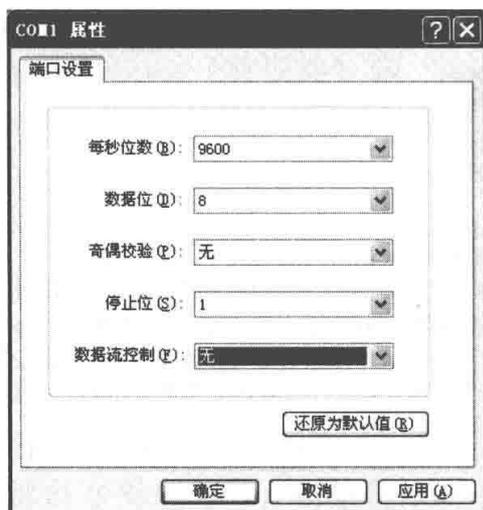


图 7-34 选择通信口参数

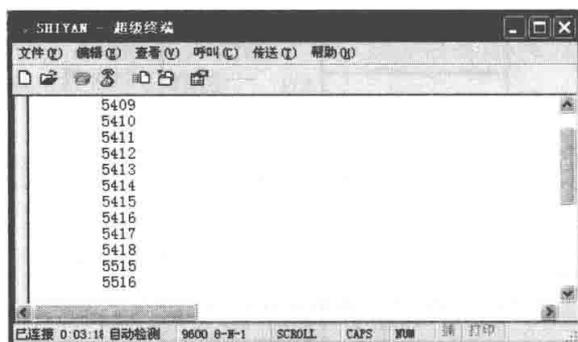


图 7-35 CPU224 发送的字符串

程序中止，接收指令结束。

## 2. 消息串起始/结束的判断

使用接收指令时需要设置消息起始和结束的判断条件。接收指令启动后，通信控制器用这些条件来判断消息的开始和结束。判断消息结束时，接收状态终止；否则，通信口会一直处在接收状态。

**【例 7-4】** S7-200 CPU 的通信口 0 接收字符串，并在信息接收中断服务程序中把接收到的第一个字节传送到 CPU 输出字节 QB0 上显示。

使用 PC/PPI 电缆连接 S7-200 CPU 和编程 PC 的串口。应用 Hyper Terminal 向 CPU 发送字符串。

### 1. 参数设置

选择空闲线检测为信息起始标志，字符 16#0A 为消息结束字符，根据接收字节控制字节定义表，应当写入 SMB87 的控制数据为 16#B0。

### 2. 程序

此例程序主要包括主程序、SBR\_0、SBR\_1、INT\_0。主程序如图 7-36 所示。SBR\_1 编程如图 7-37 所示。SBR\_0 编程如图 7-38 所示。INT\_0 编程如图 7-39 所示。

主程序：根据 CPU 模式开关的状态，定义通信口。

SBR\_0：定义自由口接收指令参数，连接接收结束中断，开始接收。

SBR\_1：重定义 PPI 通信口。

INT\_0：传送消息首字节到 QB0 输出，开始下一个接收过程。

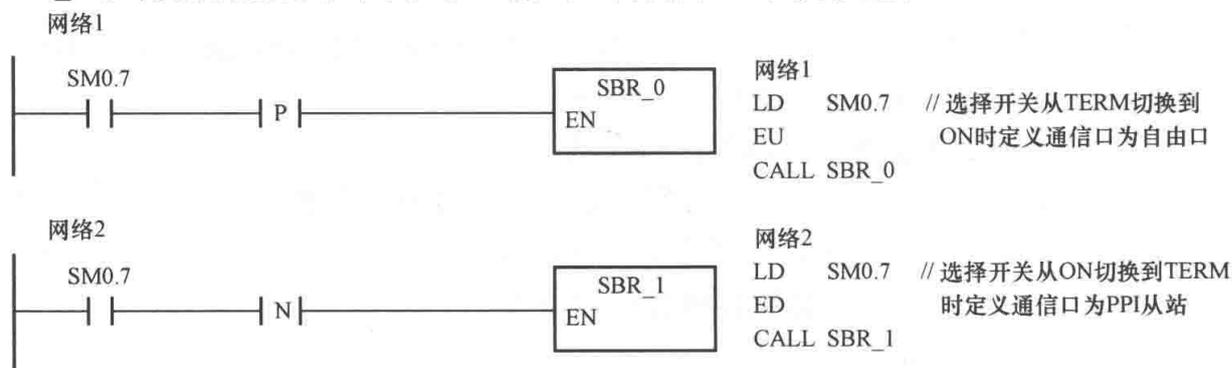


图 7-36 主程序

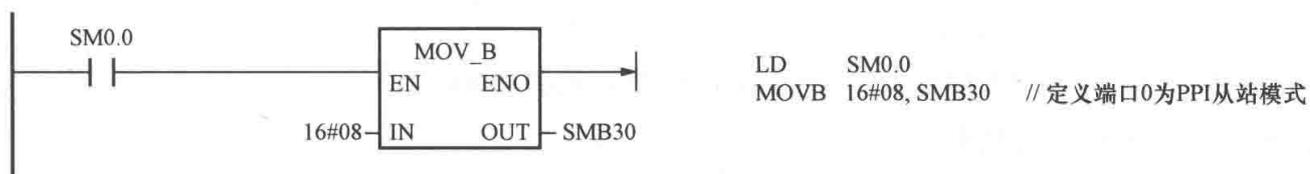


图 7-37 SBR\_1 编程

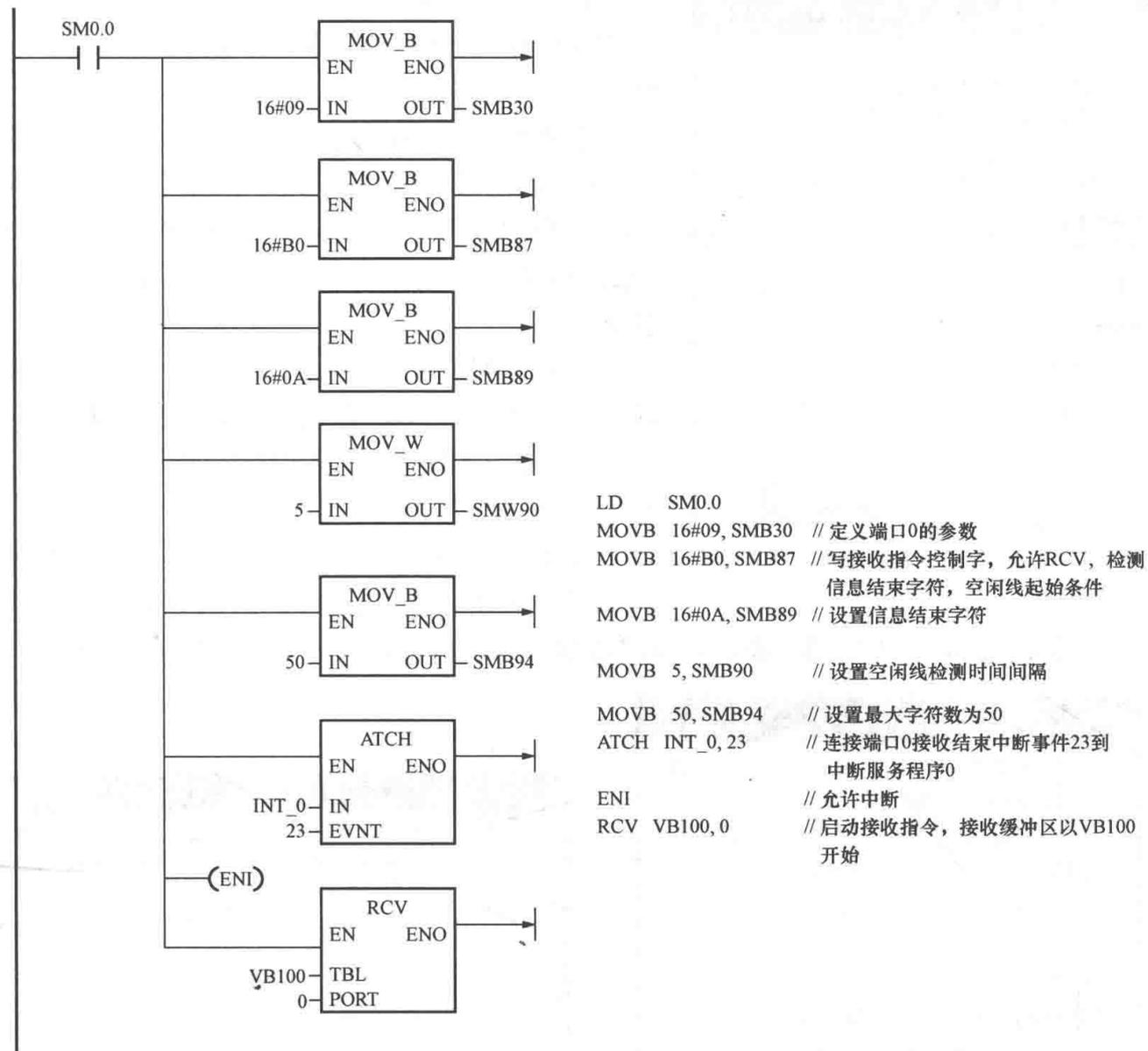


图 7-38 SBR\_0 编程

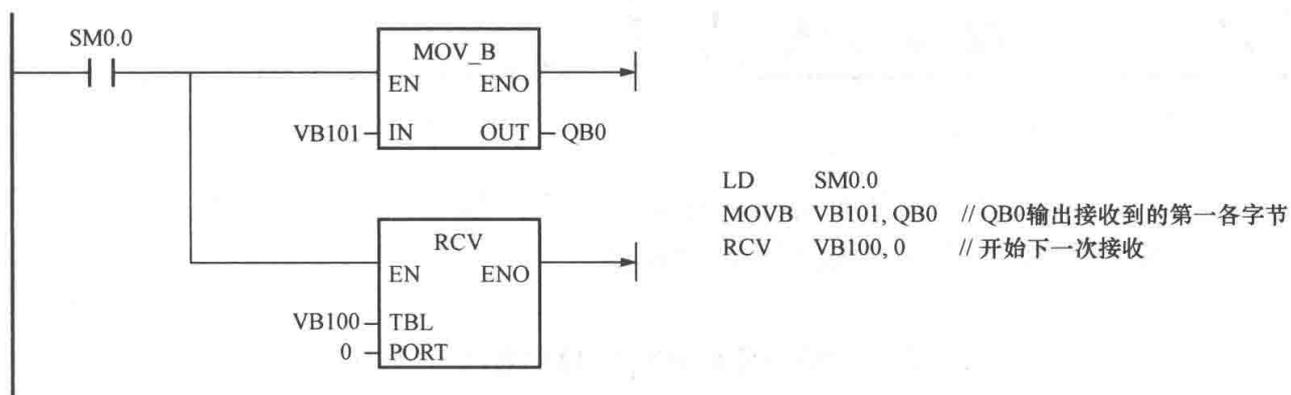


图 7-39 INT\_0 编程

### 3. 使用 Hyper Terminal 调试

打开 Windows 系统的 Hyper Terminal 程序, 选择图标, 指定一个连接名称, 本例利用了 [例 7-3] 所定义的连接。

选择监控通信口, 设置波特率后进入主界面。在“File” (文件) 菜单中选择“Properties” (属性) 选项, 如图 7-40 所示。



图 7-40 属性窗口

在“属性”菜单中, 单击“Settings” (设置) > “ASCII Setup” (ASCII 设置), 如图 7-41 所示。

下载 S7-200 项目后断开 STEP 7-Micro/WIN 与 CPU 的连接。将 S7-200 CPU 上的模式选择开关拨动到 RUN (运行) 位置。在 Hyper Terminal (超级终端) 中输入字符串, 观察 CPU 上 QB0 的状态。

如果在 Hyper Terminal 工具栏上按挂断按钮, 或在 Call (呼叫) 菜单中选择 Disconnect (断开连接) 命令, 可以释放 Hyper Terminal 对 PC 机串行口的占用。

将 S7-200 CPU 上模式开关从 ON 拨到 TERM, 重新定义自由口为 PPI 从站模式。在 STEP 7-Micro/WIN 中使用状态表, 在线观察缓冲区内容, 如图 7-42 所示。

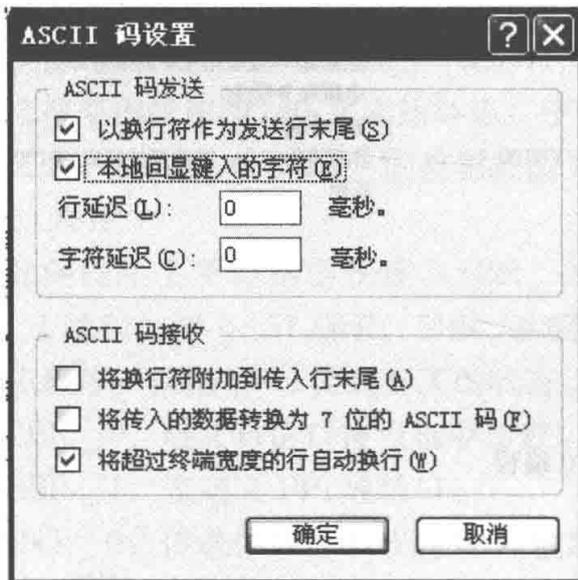


图 7-41 ASCII 码设置窗口



图 7-42 状态表监控缓冲区内容

这时 VB100 (即接收字节数) 为零。这是因为在未接收到任何字符前重新设置了 S7-200 CPU 的通信模式, 恢复为普通 PPI 从站模式。

## 7.5 S7-200 PLC 的 PROFIBUS-DP 通信

PROFIBUS-DP 是通用的国际标准, 符合该标准的第三方设备可以作 DP 网络的主站

或从站。第三方设备作主站时，组态软件由第三方提供。第三方设备作从站时，可以用STEP 7组态。需要在HW Config中安装GSD文件后，才能在硬件目录窗口中看到第三方设备和对它组态。

S7-200 CPU可以通过EM277 PROFIBUS-DP从站模块连入PROFIBUS-DP网，可以作为PROFIBUS-DP从站与主站通信，主站可以通过EM277对S7-200 CPU进行读写数据。

### 7.5.1 PROFIBUS从站模块EM 277

PROFIBUS从站模块EM 277用于将S7-200 CPU连接到PROFIBUS-DP网络，波特率为9.6kbit/s~12Mbit/s。在S7-300/400或其他系统通信时，尽量使用这种通信方式。EM 277是智能模块，能自适应通信速率，其RS-485接口是隔离型的。作为DP从站，EM 277接收来自主站的I/O组态，向主站发送和接收数据，主站可以读写S7-200的V存储区，每次可以与EM 277交换1~128B的信息。

EM 277只能作从站，必须设定与主站组态中的地址相匹配的DP端口地址。从站地址是使用EM 277模块上的旋转开关设定的。不需要在S7-200一侧对PROFIBUS-DP通信组态和编程。

EM 277在网络中除了作DP主站的从站外，还能作为MPI从站，与同一网络中的编程计算机或S7-300/400 CPU等其他主站进行通信。S7-200的编程软件可以通过EM 277对S7-200编程。模块共有6个连接，其中的两个分别保留给编程器(PG)和操作员面板(OP)。EM 277实际上是通信端口的扩展，可以用于连接人机界面(HMI)等。

### 7.5.2 EM 277的PROFIBUS通信的数据交换

EM 277模块不仅能传输I/O数据，还能读写S7-200 CPU中定义的变量数据块。这样，能使从站与主站交换任何类型的数据。首先将数据移到S7-200 CPU中的变量存储器，就可将输入、计数值、定时器值或其他计算值送到主站。其次，从主站来的数据存储在S7-200 CPU中的变量存储器内，也可移到其他数据区。

主站通过将其输出区来的信息发送给从站的输出缓冲区(称为“接收信箱”)，如图7-43所示，与其每个从站交换数据。从站将其输入缓冲区(称为“发送信箱”)的数据返回给主站的输入区，以响应从主站来的信息。

EM 277可用DP主站组态，以接收从主站来的输出数据，并将输入数据返回给主站。输出和输入数据缓冲区驻留在S7-200 CPU的变量存储器(V存储器)内。当组态DP主站时，

应定义V存储器内的字节位置，从这个位置开始为输出数据缓冲区，它应作为EM 277的参数赋值信息的一个部分。还要定义I/O组态，它是写入到S7-200 CPU的输出数据总量和从S7-200 CPU返回的输入数据总量。EM 277从I/O组态确定输入和输出缓冲区的大小。DP主站将参数赋值和I/O组态信息写入到EM 277模块，然后，EM 277将V存

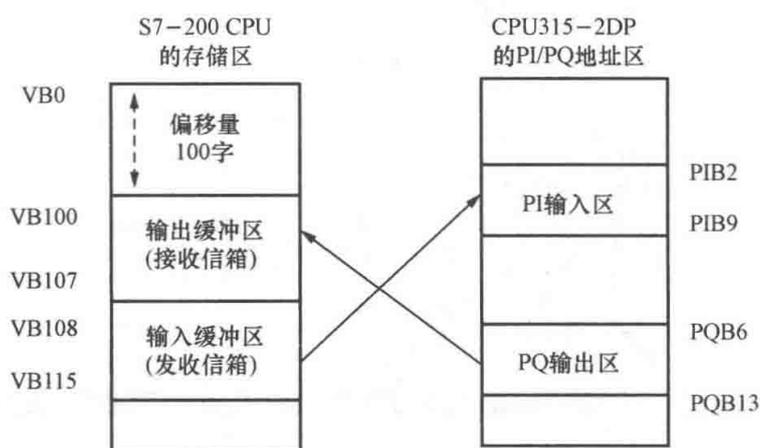


图7-43 V存储区和I/O地址区域

存储器地址和输入及输出数据长度传送给 S7 - 200 CPU。

图 7-43 表示 CPU 224 中的 V 存储器的一个存储器模型, 以及一个 DP 主站 CPU 的 I/O 地址区。DP 主站已定义了 8 输出字节和 8 输入字节的一种 I/O 组态, 以及 V 存储器偏移为 100。S7 - 200 CPU 中的输出缓冲区和输入缓冲区长度 (由 I/O 组态确定) 都是 8 字节。输出数据缓冲区从 VB100 开始; 输入数据缓冲区紧紧跟随输出缓冲区, 并在 VB108 处开始。输出数据 (从主站来) 放置在 V 存储器中的 VB100; 输入数据 (传送到主站) 取自 V 存储器的 VB108。

### 7.5.3 从站模块 EM 277 的组态应用举例

本例中, S7 - 300 与 S7 - 200 通过 EM 277 进行 PROFIBUS DP 通信, CPU315 - 2DP 是主站, S7 - 200 PLC 通过 EM277 模块设为从站, 要求将 S7 - 300 的输入 IB0 写到 S7 - 200 的输出 QB0 中, S7 - 200 的输入 IB2 写到 S7 - 300 的输出 QB4 中。

需要在 STEP7 中对 S7 - 300 和 EM 277 组态。在 S7 - 200 中, 只需将待发送的数据传送到组态时指定的 V 存储区, 或者在组态时指定的 V 存储区中读取接收的数据就可以了。其实现的步骤如下:

#### 1. 组态 S7 - 300 站

在 STEP 7 的 SIMATIC 管理器中, 生成一个项目, CPU 模块的型号为 CPU 315 - 2DP。选中该站后, 点击右边窗口的“硬件”图标, 打开硬件组态工具, 按硬件安装顺序一次插入机架、电源、CPU 模块及输入输出模块进行 S7 - 300 PLC 的硬件组态, 如图 7-44 所示。

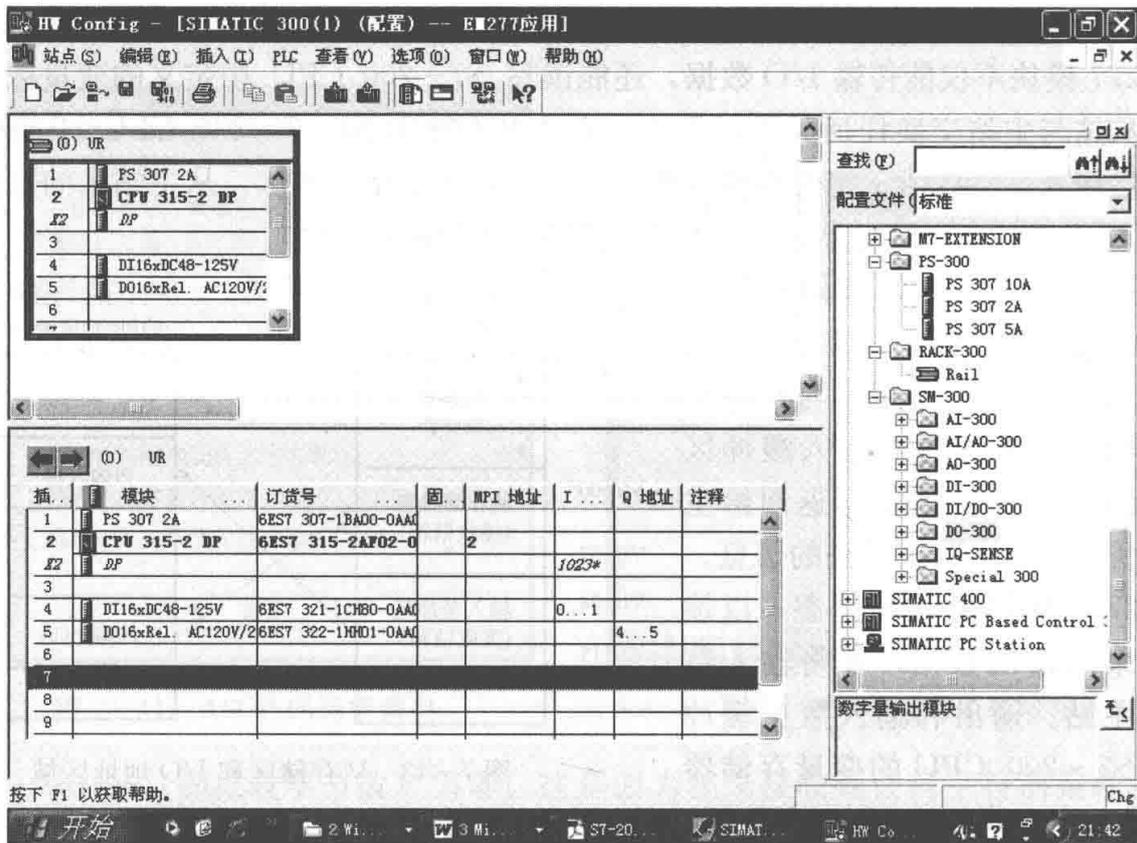


图 7-44 硬件组态

双击“DP”所在的行, 点击打开的对话框的“常规”选项卡中的“属性”按钮, 在出现的对话框的“参数”选项卡中设置站地址, 默认的站地址为 2。点击“新建”按钮, 生成

一条 PROFIBUS-DP 网络，如图 7-45 所示。点击“确定”按钮返回 HW Config，形成图 7-46 的硬件组态画面。

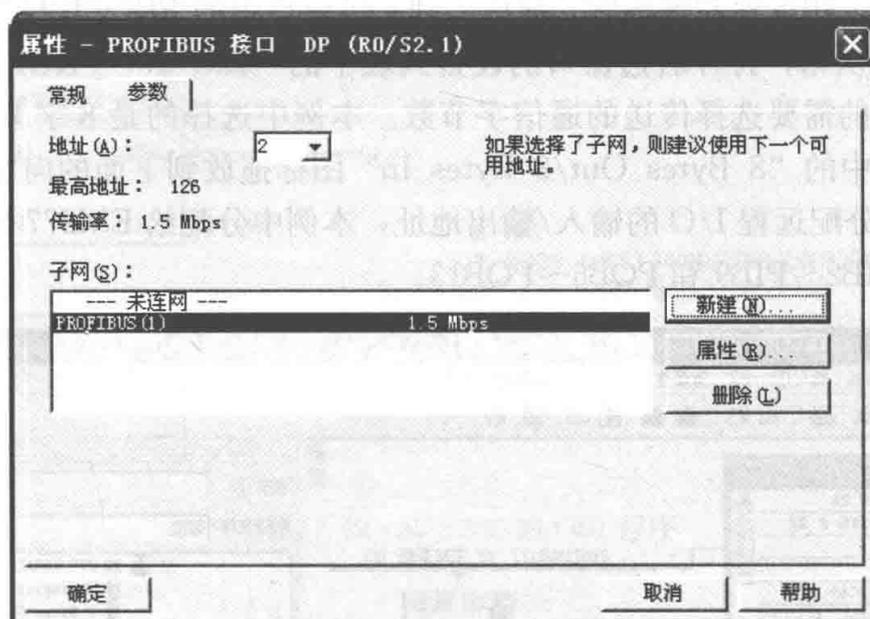


图 7-45 PROFIBUS 的建立

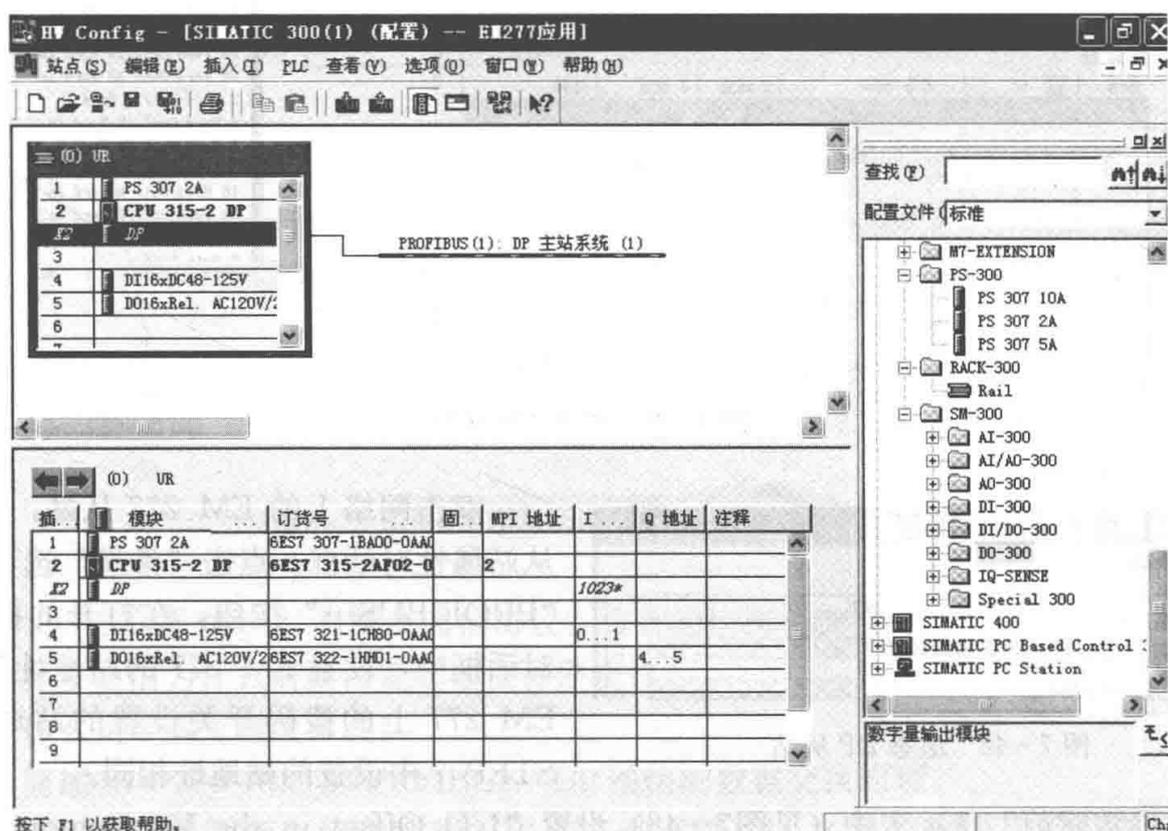


图 7-46 带有 PROFIBUS 网络的组态画面

## 2. 安装 EM 277 的 GSD 文件

EM 277 作为一个特殊的 PROFIBUS-DP 从站模块，其相关参数是以 GSD 文件的形式保存的。在 STEP 7 组态之前，需要安装它的 GSD 文件。

执行图 7-46 中的菜单命令“选项”→“安装 GSD 文件”（可以下载），安装 EM 277 从站配置文件。

### 3. 组态 EM 277 从站

导入 GSD 文件后, 在右侧窗口的设备列表中找到 EM 277 从站“EM 277 PROFIBUS - DP”(见图 7-46), 用鼠标左键将它“拖放”到左边窗口中的 PROFIBUS - DP 网络上。用鼠标选中 EM 277 从站, 打开右边窗口的设备列表中的“EM 277 PROFIBUS - DP”文件夹, 根据实际系统的需要选择传送的通信字节数。本例中选择的是 8 字节输入/8 字节输出方式, 将图 7-47 中的“8 Bytes Out/8 Bytes In”图标拖放到下面的向窗口中表格的第一行。STEP 7 自动分配远程 I/O 的输入/输出地址, 本例中分配给 EM277 模块的输入、输出字节地址分别为 PIB2~PIB9 和 PQB6~PQB13。

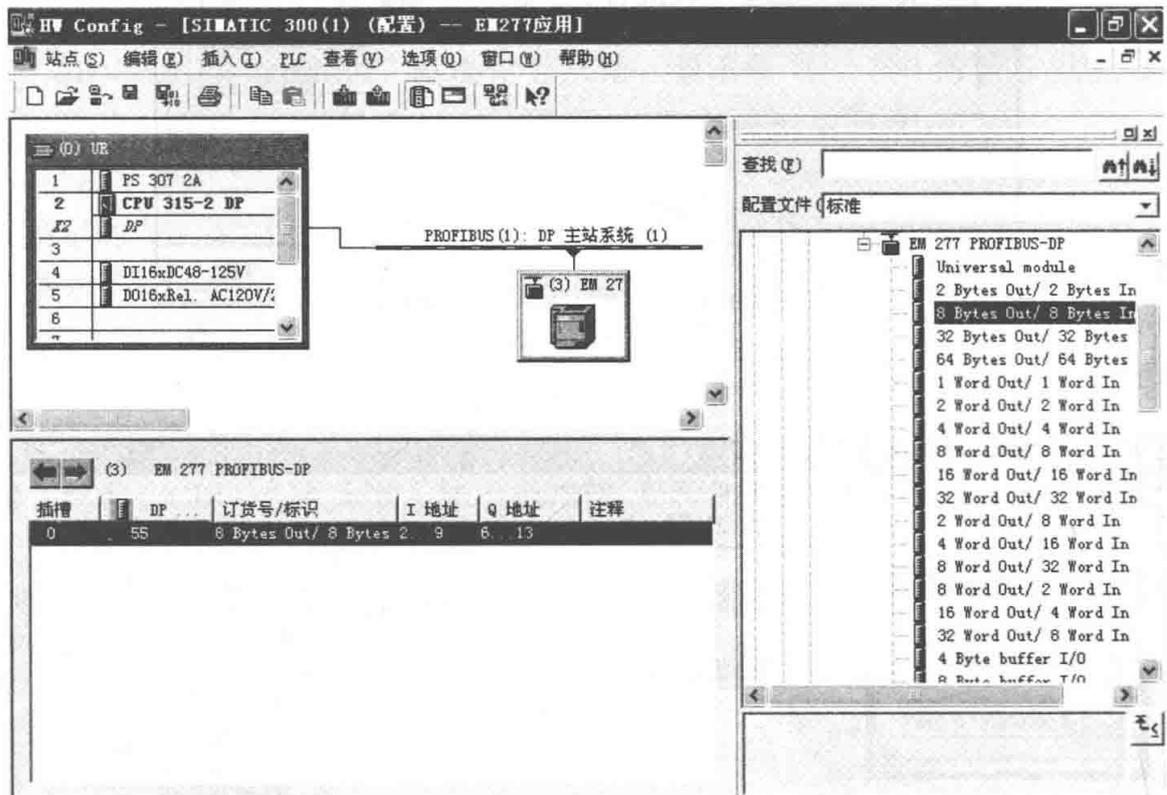


图 7-47 组态 DP 从站

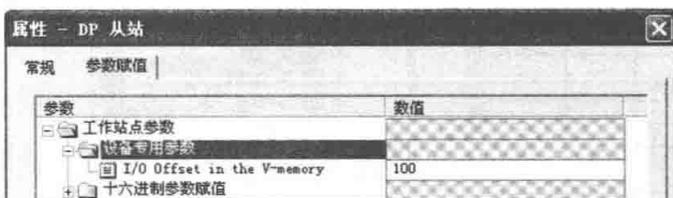


图 7-48 组态 DP 从站

双击网络上的 EM 277 从站, 打开 DP 从站属性对话框。点击“常规”选项卡中的“PROFIBUS...”按钮, 在打开的接口属性对话框中, 设置 EM 277 的站地址为 3。用 EM 277 上的拨码开关设置的站地址应与 STEP 7 中设置的站地址相同。

在“参数赋值”选项卡中(见图7-48), 设置“I/O Offset in the V-memory”(V 存储区中的 I/O 偏移量)为 100, 即用 S7-200 的 VB100~VB115 与 S7-300 的 PIB2~PIB9 和 PQB6~PQB13 交换数据。组态完系统的硬件配置后, 应将硬件信息下载到 S7-300 的 CPU 模块。

### 4. S7-200 的编程

本例中 S7-200 通过 VB100~VB115 与主站交换数据。VB100~VB107 是 S7-300 写到 S7-200 的数据, 对应于 S7-300 的 PQB6~PQB13; VB108~VB115 是 S7-300 从 S7-200 读取的数据, 对应于 S7-300 的 PIB2~PIB9。用 STEP 5.4 中文版编写的 S7-300 PLC

的 OB1 程序如图 7-49 所示。S7-200 PLC 的主程序如图 7-50 所示。

在运行时可以用 STEP 7 的变量表和 STEP 7 - Micro/Win 的状态表来监控通信的数据。

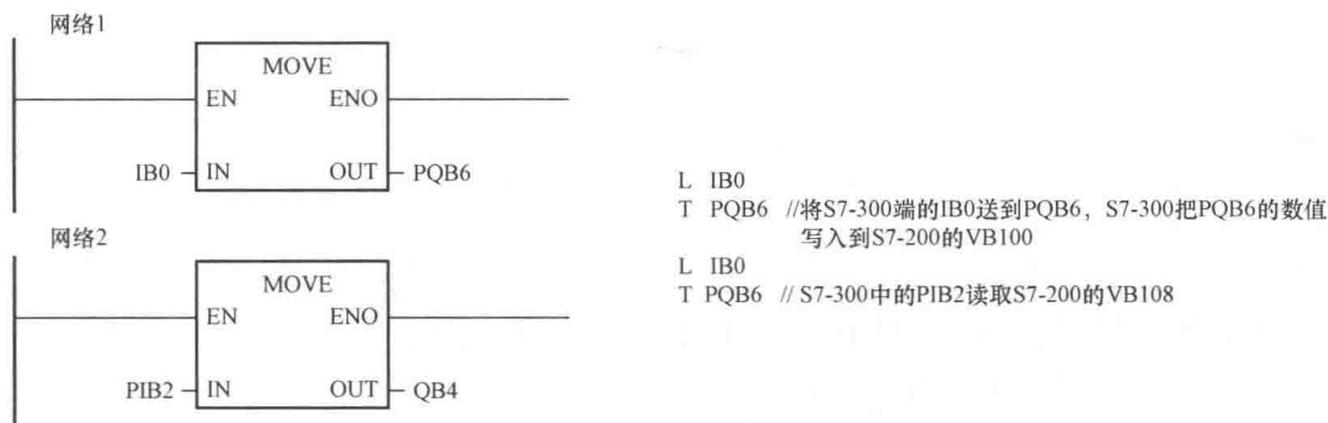


图 7-49 S7-300 的 OB1 程序

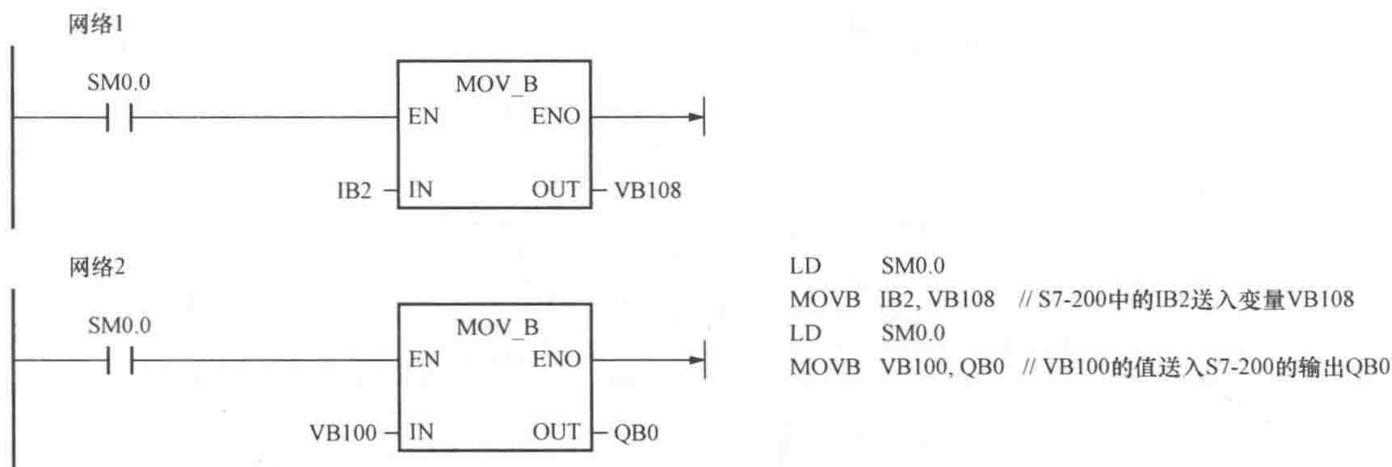


图 7-50 S7-200 的主程序

## 习 题



1. S7-200 系列 PLC 有哪些通信协议? 每种通信协议各有什么特点?
2. 异步通信为什么需要设置起始位和停止位?
3. 什么是半双工通信方式?
4. 简述 S7-200 PLC 的 PROFIBUS-DP 通信的数据交换原理。
5. 利用网络读写指令编程实现 CPU226 和 CPU224 之间的数据交换, 把 CPU226 中 MB100 传送到 CPU224 的 QB0。

## 第8章 S7-200 PLC 控制系统的设计与应用

### 8.1 系统设计

#### 8.1.1 系统设计的原则

在可编程控制器控制系统的设计中，应该最大限度地满足生产机械或生产流程对电气控制的要求，在满足控制要求的前提下，力求 PLC 控制系统简单、经济、安全、可靠、操作和维修方便，而且应使系统能尽量降低使用者长期运行的成本，图 8-1 为设计调试过程示意图。

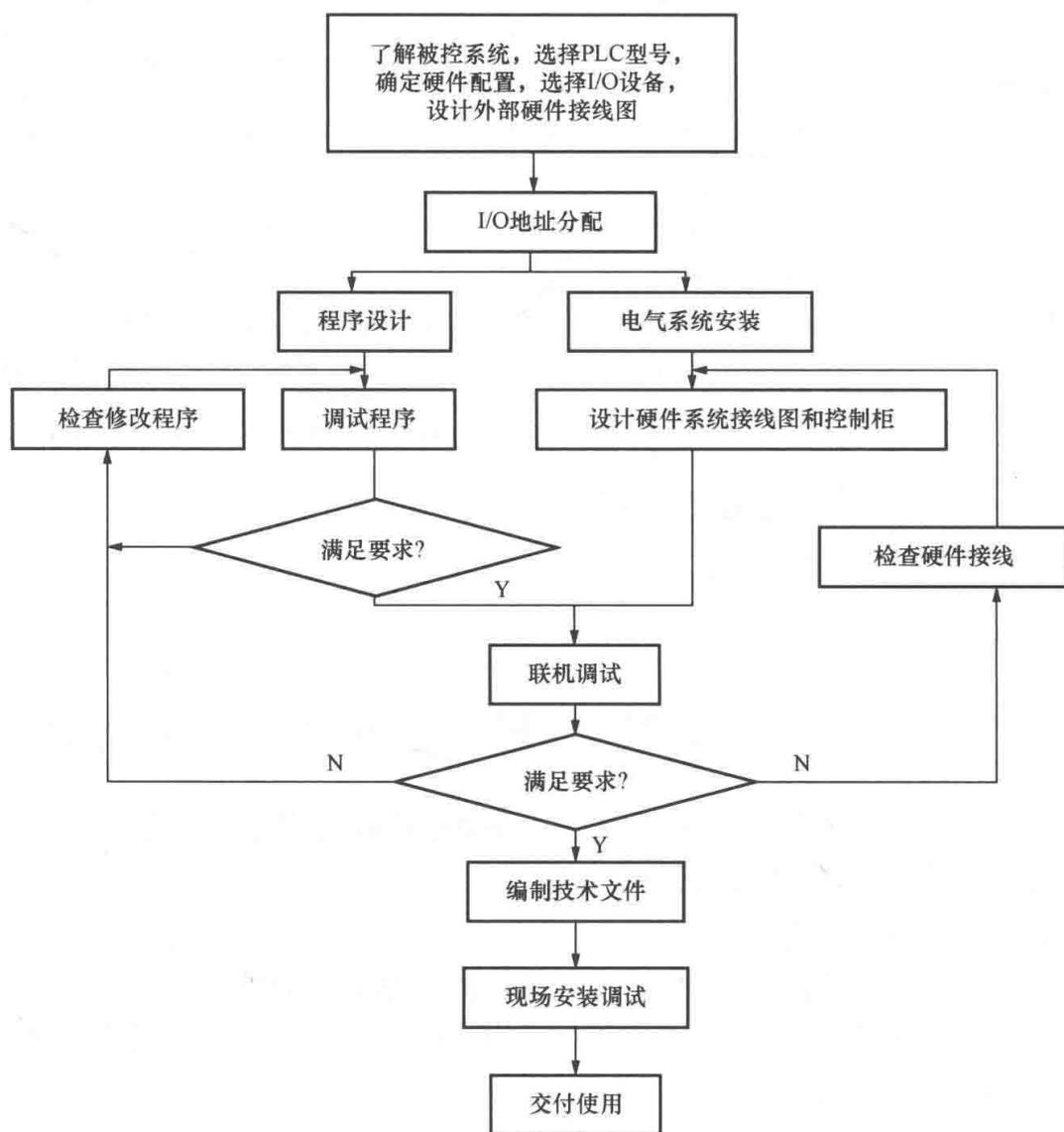


图 8-1 设计调试过程示意图

#### 8.1.2 PLC 控制系统的设计和调试步骤

##### 1. 熟悉被控对象

这一步是系统设计的基础。设计前应熟悉图纸资料，深入调查研究，与工艺、机械方面

的技术人员和现场操作人员密切配合，共同讨论，解决设计中出现的问题。应详细了解被控对象的全部功能，如是否需要模拟量控制，是否需要通信联网控制，哪些信号需要输入给 PLC，哪些负载需要由可编程控制器控制等。

## 2. 选择 PLC 型号、I/O 设备选择并设计外部硬件接线图

(1) 选择 PLC 型号。选择 PLC 的型号应考虑到 PLC 的硬件功能、PLC 指令系统的功能、PLC 的物理结构及输入输出的点数等方面的情况。

(2) I/O 设备选择。根据 I/O 表和可供选择的 I/O 模块的类型，确定 I/O 模块的型号和块数。选择 I/O 模块时，I/O 点数一般应留有一定的余量。

开关量输入模块的电压一般为 DC24V 和 AC220V。直流输入电路的延时时间较短，可以直接与接近开关、光电开关等电子输入装置连接。交流输入方式适合于有油雾、粉尘的恶劣环境下。

开关量输出模块要考虑动作速度、触点容量、工作电压范围等因素。如果系统的输出信号变化不是很频繁，一般优先选用继电器输出型的模块。

(3) 设计外部硬件接线图。PLC 的交流系统接线图如图 8-2 所示，交流接线安装时用一个单刀切断开关将电源与 CPU、所有的输入电路和输出（负载）电路隔离开。用一台过电流保护设备以保护 CPU 的电源、输出点以及输入点。根据情况也可以为每个输出点加上熔丝进行范围更广的保护。主机单元的直流传感器电源可用来为主机单元的输入。

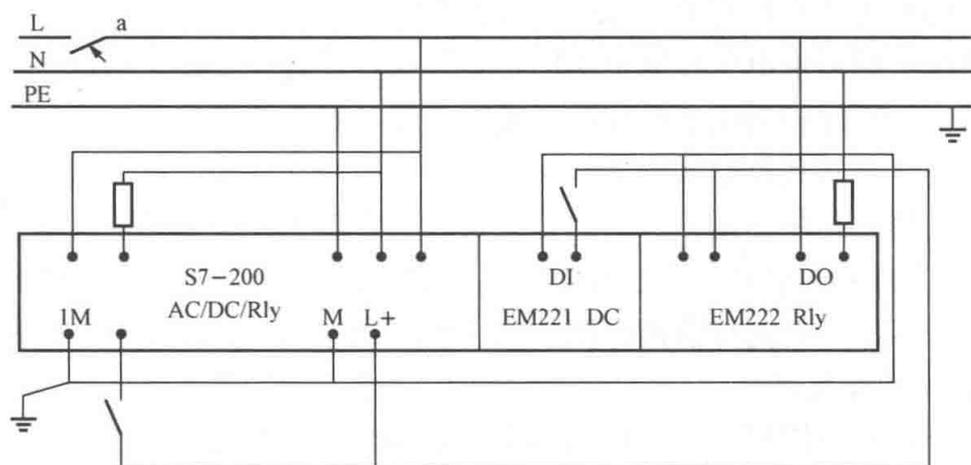


图 8-2 PLC 的交流系统接线图

PLC 的直流系统接线图如图 8-3 所示，直直流接线安装时用一个单刀开关 a 将电源同 CPU、所有的输入电路和输出（负载）电路隔离开。用过电流保护设备以保护 CPU 电源，c 输出点，以及 d 输入点。也可以在每个输出点加上熔丝进行过电流防护。

在给 CPU 进行供电接线时，一定要特别小心分清是哪一种供电方式，如果把 220V AC 接到 24V DC 供电的 CPU 上，或者不小心接到 24V DC 传感器输出电源上，都会造成 CPU 的损坏。

当使用 PLC24V DC 传感器电源时，可以取消输入点的外部过电流保护，因为该传感器电源具有短路保护功能。

## 3. 程序设计

程序设计的内容包括：编写程序、编译程序、模拟运行及调试程序等。

程序设计的方法是指用什么方法和编程语言来编写用户程序。

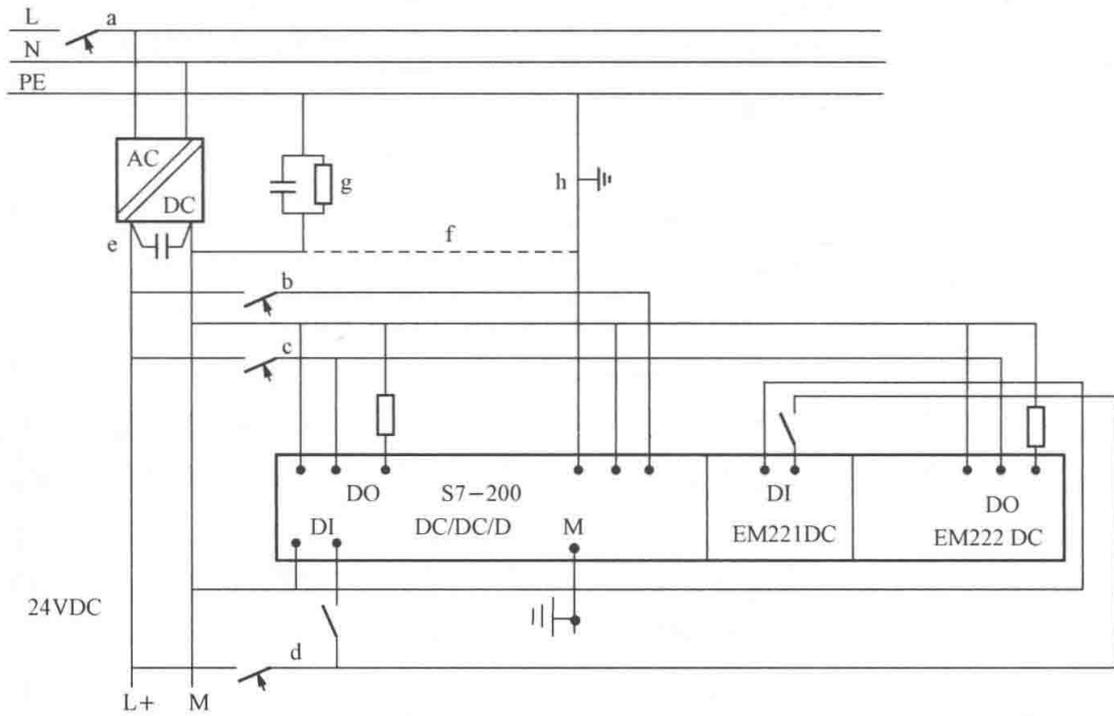


图 8-3 PLC 的直流系统接线图

程序设计有多种方法：顺序控制设计法是最通用的方法，启、保、停电路的编程方式和以转换为中心的编程方式是所有 PLC 都适用的编程方法。如果控制系统是改造原有成熟的继电接触控制系统，则可由电气控制电路图很容易地转化为梯形图，生成控制程序。如果是时序控制，则可以用时序控制或者顺序控制设计法设计梯形图程序。

首先应根据总体要求和控制系统的具体情况，确定用户程序的基本结构，画出程序流程图或开关量控制系统的顺序功能图。它们是编程的主要依据，应尽可能准确和详细。

较简单的系统的梯形图可以用经验法设计，比较复杂的系统一般采用顺序控制设计法。画出系统的顺序功能图后，选择前面介绍的某一种编程方式，设计出梯形图程序。

#### 4. 梯形图程序的模拟调试

对用户程序一般先做模拟调试，根据顺序功能图，用小开关和按钮来模拟可编程控制器实际的输入信号，例如用它们发出操作指令，或在适当的时候用它们来模拟实际的反馈信号，如限位开关触点的接通和断开。通过输出模块上各输出继电器对应的发光二极管，观察各输出信号的变化是否满足设计的要求。

调试顺序控制程序的主要任务是检查程序的运行是否符合顺序功能图的规定，即在某一转换条件实现时，是否发生步的活动状态的正确变化，该转换所有的前级步是否变为不活动步，所有的后续步是否变为活动步，以及各步被驱动的负载是否发生相应的变化。

在调试时应充分考虑各种可能的情况，对系统各种不同的工作方式、顺序功能图中的每一条支路、各种可能的进展路线，都应逐一检查，不能遗漏。发现问题后及时修改程序，直到在各种可能的情况下输入信号与输出信号之间的关系完全符合要求。

如果程序中某些定时器或计数器的设定值过大，为了缩短调试时间，可以在调试时将它们减小，模拟调试结束后再写入它们的实际设定值。

在设计和模拟调试程序的同时，可以设计、制作控制台或控制柜，可编程控制器之外的其他硬件的安装、接线工作也可以同时进行。

### 5. 现场调试

完成上述工作后,将可编程控制器安装在控制现场,接入实际的输入信号和负载。在联机总调试过程中将暴露出系统中可能存在的传感器、执行器和接线等硬件方面的问题,以及可编程控制器的外部接线图和梯形图设计中的问题,发现问题后在现场加以解决,直到完全符合要求。

### 6. 编写技术文件

系统交付使用后,应根据调试的最终结果整理出完整的技术文件,并提供给用户,以利于系统的维修和改进。技术文件应包括:

- (1) 可编程控制器的外部接线图和其他电器图纸。
- (2) 可编程控制器的编程元件表,包括程序中使用的输入/输出继电器、辅助继电器、定时器、计数器等元件号、名称、功能,以及定时器、计数器的设定值等。
- (3) 顺序功能图、带注释的梯形图和必要的总体文字说明。

## 8.2 S7-200 PLC 应用系统的可靠性措施

### 8.2.1 S7-200 PLC 使用中应注意的问题

市场上经常出现继电器问题的用户现场有一个共同的特点就是:出现故障的输出点动作频率比较快,驱动的负载都是继电器、电磁阀或接触器等感性负载而且没有吸收保护电路。因此建议在 PLC 输出类型选择和使用时应注意以下几点:

#### 1. 负载容量

输出端口须遵守允许最大电流限制,以保证输出端口的发热限制在允许范围。继电器的使用寿命与负载容量有关,当负载容量增加时,触点寿命将大大降低,因此要特别关注。

#### 2. 负载性质

感性负载在开合瞬间会产生瞬间高压,因此表面上看负载容量可能并不大,但是实际上负载容量很大,继电器的寿命将大大缩短,因此当驱动感性负载时应在负载两端接入吸收保护电路。尤其在工作频率比较高时务必增加保护电路。从客户的使用情况来看,增加吸收保护电路后的改善效果十分明显。

根据电容的特性,如果直接驱动电容负载,在导通瞬间将产生冲击浪涌电流,因此原则上输出端口不宜接入容性负载,若有必要,需保证其冲击浪涌电流小于规格说明中的最大电流。

#### 3. 动作频率

当动作频率较高时,建议选择晶体管输出类型,如果同时还要驱动大电流,则可以使用晶体管输出驱动中间继电器的模式。当控制步进电机/伺服系统,或者用到高速输出/PWM波,或者用于动作频率高的节点等场合,只能选用晶体管型。PLC 对扩展模块与主模块的输出类型并不要求一致,因此当系统点数较多而功能各异时,可以考虑继电器输出的主模块扩展晶体管输出或晶体管输出主模块扩展继电器输出以达到最佳配合。

事实证明,根据负载性质和容量以及工作频率进行正确选型和系统设计,输出口的故障率明显下降,客户十分满意。

#### 4. S7-200 的电源需求与计算

S7-200 CPU 模块提供 5VDC 和 24VDC 电源。当有扩展模块时 CPU 通过 I/O 总线为其提供 5V 电源, 所有扩展模块的 5V 电源消耗之和不能超过该 CPU 提供的电源额定。若不够用不能外接 5V 电源。每个 CPU 都有一个 24VDC 传感器电源, 它为本机输入点和扩展模块输入点及扩展模块继电器线圈提供 24VDC。如果电源要求超出了 CPU 模块的电源定额, 你可以增加一个外部 24VDC 电源来提供给扩展模块。所谓电源计算, 就是用 CPU 所能提供的电源容量, 减去各模块所需要的电源消耗量。24VDC 电源需求取决于通信端口上的负载大小。CPU 上的通信口, 可以连接 PC/PPI 电缆和 TD 200 并为它们供电, 此电源消耗已经不必再纳入计算。

#### 8.2.2 安装和布线的注意事项

开关量信号一般对信号电缆没有严格的要求, 可以选用普通电缆, 信号传输距离较远时, 可以选用屏蔽电缆。模拟量信号和高速信号 (例如光电编码器等提供的信号) 应选择屏蔽电缆。有的通信电缆的信号频率很高, 一般应选用专用电缆或光纤电缆, 在要求不高或信号频率较低时, 也可以选用带屏蔽的多芯电缆或双绞线电缆。

PLC 应避免强干扰源, 例如大功率晶闸管装置、变频器、高频焊机和大型动力设备等。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, 两者之间的距离应大于 200mm。与 PLC 装在同一个开关柜内的电感性元件, 例如继电器、接触器的线圈, 应并联 RC 消弧电路。

信号线与功率线应分开走线, 电力电缆应单独走线, 不同类型的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, 信号线应尽量靠近地线或接地的金属导体。

当开关量 I/O 线不能与动力线分开布线时, 可以用继电器来隔离输入/输出线上的干扰。当信号线距离超过 300m 时, 应采用中间继电器来转接信号, 或使用 PLC 的远程 I/O 模块。

I/O 线与电源线应分开走线, 并保持一定的距离。如果不得已要在同一线槽中布线, 应使用屏蔽电缆。交流线与直流线应分别使用不同的电缆; 开关量、模拟量 I/O 线应分开敷设, 后者应采用屏蔽线。如果模拟量输入/输出信号距离 PLC 较远, 应采用 DC4~20mA 的电流传输方式, 而不是易受干扰的电压传输方式。

传送模拟信号的屏蔽线, 其屏蔽层应一端接地, 为了泄放高频干扰, 数字信号线的屏蔽层应并联电位均衡线, 其电阻应小于屏蔽层电阻的 1/10, 并将屏蔽层两端接地。如果无法设置电位均衡线, 或只考虑抑制低频干扰时, 也可以一端接地。

不同的信号线最好不用同一个接插转接, 如果必须用同一个接插件, 要用备用端子或地线端子将它们分隔开, 以减少相互干扰。

#### 8.2.3 控制系统的接地

良好的接地是 PLC 安全可靠运行的重要条件, PLC 与强电设备最好分别使用接地装置, 接地点与 PLC 的距离应小于 50m。将 S7-200 的所有地线端子同最近接地点相连接, 以获得最好的抗干扰能力。一般所有的接地端子都使用  $1.5\text{mm}^2$  的电线连接到独立导电点上 (亦称一点接地), 如图 8-3 所示的 h 点。

在大部分的安装接线中, 如果把 PLC 上传感器电源的 M 端子接到地上可以获得最佳的

噪声抑制。

在大部分的应用中，把所有的 DC 电源接到地可以得到最佳的噪声抑制。在未接地 DC 电源的公共端与保护地 PE 之间并联电阻与电容，如图 8-3 所示的 b 点。电阻提供了静电释放通路，电容提供高频噪声通路，它们的典型值是  $1M\Omega$  和  $4700pF$ 。

在发电厂或变电站中，有接地网络可供使用。各控制屏和自动化元件可能相距甚远，若分别将它们在就近的接地点接地，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。为防止不同信号回路接地线上的电流引起交叉干扰，必须分系统（例如以控制屏为单位）将弱电信号的内部地线接通，然后各自用规定截面积的导线统一引到接地网络的同一点，从而实现控制系统一点接地的要求。

#### 8.2.4 抑制电路的使用

继电器控制接触器等感性负载的开合瞬间，由于电感具有电流不可突变的特点，因此根据  $U=L(dI/dt)$ ，将产生一个瞬间的尖峰电压在继电器的两个触点之间，该电压幅值超过继电器的触点耐压的降额；继电器采用的电磁式继电器，触点间的耐受电压是  $1000V$  ( $1min$ )，若触点间的电压长期的工作在  $1000V$  左右的话，容易造成触点金属迁移和氧化，出现接触电阻变大、接触不良和触点粘接的现象，而且动作频率越快现象越严重。瞬间高压如持续的时间在  $1ms$  以内，幅值为  $1kV$  以上。晶体管输出为感性负载时也同样存在这个问题，该瞬时高压可能导致晶体管的损坏。

因此当驱动感性负载时应在负载两端接入吸收保护电路。当驱动直流回路的感性负载（如继电器线圈）时，用户电路需并联续流二极管（需注意二极管极性）；若驱动交流回路的感性负载时，用户电路需并联 RC 浪涌吸收电路，以保护 PLC 的输出触点。PLC 输出触点的保护电路如图 8-4 所示。

#### 8.2.5 强烈干扰环境中的隔离措施

PLC 内部用光耦合器、输出模块中的小型继电器和光敏晶闸管等器件来实现对外部开关量信号的隔离，PLC 的模拟量 I/O 模块一般也用光耦合器来实现隔离。这些器件除了能减少或消除外部干扰对系统的影响外，还可以保护 CPU 模块，使之免受从外部窜入 PLC 的高电压的危害，因此一般没有必要在 PLC 外部再设置干扰隔离器件。

在某些工业环境，PLC 受到强烈的干扰。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电缆沟内，强电干扰在输入线上产生的感应电压和感应电流相当大，足以使 PLC 输入端的光耦合器中的发光二极管发光，光耦合器的隔离作用失效，使 PLC 产生误动作。在这种情况下，对于用长线引入 PLC 的开关量信号，可以用小型继电器来隔离。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。

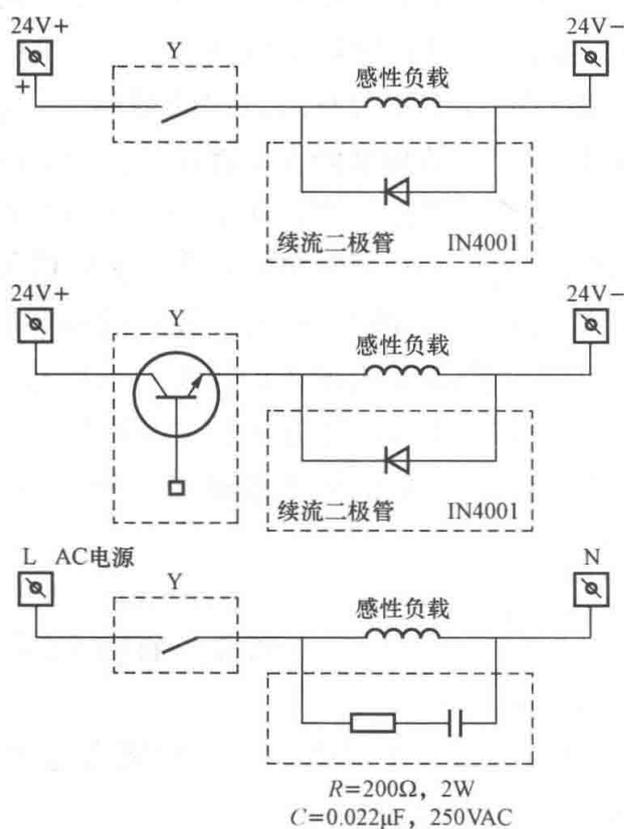


图 8-4 PLC 输出触点的保护电路

为了提高抗干扰能力和防雷击, PLC 和计算机之间的串行通信线路可以考虑使用光纤, 或采用带光耦合器的通信接口。

PLC 的 24 VDC 电源回路与设备之间, 以及 120/230 VAC 电源与危险环境之间, 必须提供安全电气隔离。

### 8.2.6 故障的检测与诊断

PLC 的可靠性很高, 本身有很完善的自诊断功能, 如果出现故障, 借助自诊断程序可以方便地找到出现故障的部件, 更换它后就可以恢复正常工作。

大量的工程应用实践表明, PLC 外部的输入、输出元件, 例如限位开关、电磁阀、接触器等的故障率远远高于 PLC 本身的故障率, 这些元件出现故障后, PLC 一般不能觉察出来, 不会自动停机, 可能使故障扩大, 直至强电保护装置动作后停机, 有时甚至会造成设备和人身事故。停机后, 查找故障也要花费很多时间。为了及时发现故障, 在没有酿成事故之前自动停机和报警, 也为了方便查找故障, 提高维修效率, 可用梯形图程序实现故障的自诊断和自动处理。

(1) 超时检测。机械设备在各工步的动作所需的时间一般是不变的, 即使变化也不会太大, 因此可以以这些时间为参考, 在 PLC 发出输出信号, 相应的外部执行机构开始动作时启动一个定时器定时, 定时器的设定值比正常情况下该动作的持续时间长一些。例如设某执行机构在正常情况下运行 10s 后, 它驱动的部件使限位开关动作, 发出动作结束信号。在该执行机构开始动作时启动设定值为 12s 的定时器定时, 若 12s 后还没有接收到动作结束信号, 由定时器的动合触点发出故障信号, 该信号停止正常的程序, 启动报警和故障显示程序, 使操作人员和维修人员能迅速判别故障的种类, 及时排除故障。

(2) 逻辑错误检测。在系统正常运行时, PLC 的输入、输出信号和内部的信号 (例如存储器位的状态) 相互之间存在着确定的关系, 如果出现异常的逻辑信号, 则说明出现了故障。因此, 可以编制一些常见故障的异常逻辑关系, 一旦异常逻辑关系为 ON 状态, 就应按故障处理。例如某机械运动过程中先后有两个限位开关动作, 这两个信号不会同时为 ON。若它们同时为 ON, 说明至少有一个限位开关被卡死, 应停机进行处理。在梯形图中, 用这两个限位开关对应的输入继电器的动合触点串联, 来驱动一个表示限位开关故障的辅助继电器。

## 8.3 节省 PLC 输入输出点数的方法

PLC 的每一 I/O 点的平均价格高达数十元, 减少所需 I/O 点数是降低系统硬件费用的主要措施。

### 8.3.1 减少所需输入点数的方法

#### 1. 分时分组输入

自动程序和手动程序不会同时执行, 自动和手动这两种工作方式分别使用的输入量可以分成两组输入 [见图 8-5 (a)]。I0.0 用来输入自动/手动命令信号, 供自动程序和手动程序切换之用。

图 8-5 (a) 中的二极管用来切断寄生电路。如果没有二极管, 系统处于自动状态,

K1、K2、K3 闭合，K4 断开，这时电流从 I0.2 端子流入，经 K2、K1、K3 形成的寄生回路流入 1M 端子，使输入继电器 I0.2 错误地变为 ON。各开关串联了二极管后，切断了寄生回路，避免了错误输入的产生。

### 2. 输入触点的合并

如果某些外部输入信号总是以某种“与或非”组合的整体形式出现在梯形图中，可以将它们对应的触点在 PLC 外部串、并联后作为一个整体输入 PLC，只占 PLC 的一个输入点。

例如某负载可在多处启动和停止，可以将多个启动用的动合触点并联，将多个停止用的动断触点串联，分别送给 PLC 的两个输入点。与每一个启动信号或停止信号分别占用一个输入点的方法相比，不仅节约了输入点，还简化了梯形图电路。

### 3. 将信号设置在 PLC 之外

系统的某些输入信号，例如手动操作按钮、过载保护动作后需手动复位的电动机热继电器 FR 的动断触点提供的信号，可以设置在 PLC 外部的硬件电路中 [见图 8-5 (b)]。某些手动按钮需要串接一些安全联锁触点，如果外部硬件联锁电路过于复杂，则应考虑仍将有关信号送入 PLC，用梯形图实现联锁。

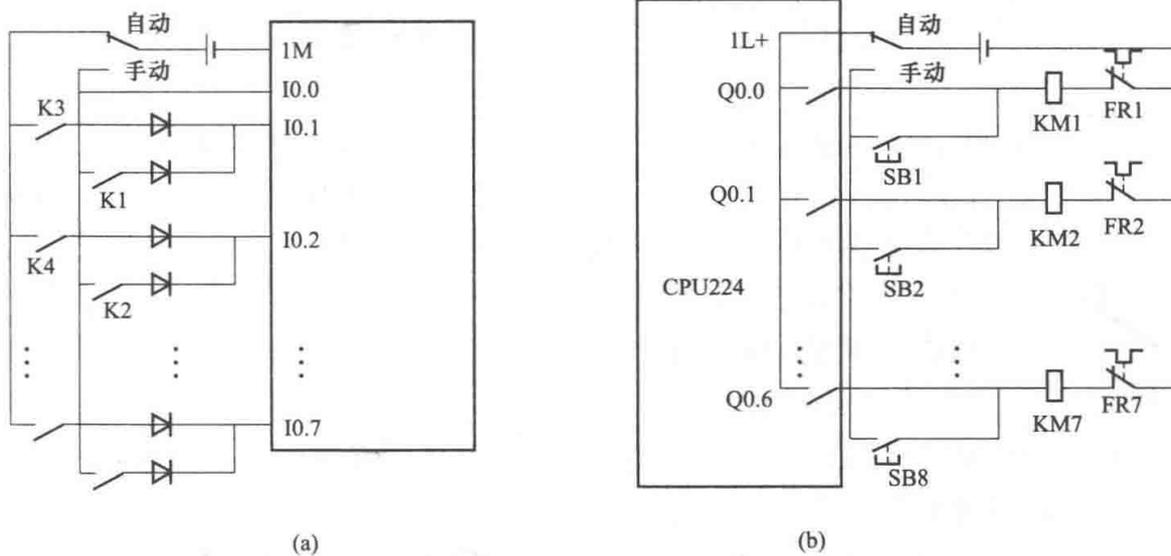


图 8-5 减少输入输出点数的方法示意图

(a) 分时分组输入；(b) 将信号放置在 PLC 之外

### 8.3.2 减少所需输出点数的方法

在 PLC 的输出功率允许的条件下，通/断状态完全相同的多个负载并联后，可以共用一个输出点，通过外部的或 PLC 控制的转换开关的切换，一个输出点可以控制两个或多个不同时工作的负载。与外部元件的触点配合，可以用一个输出点控制两个或多个有不同要求的负载。用一个输出点控制指示灯常亮或闪烁，可以显示两种不同的信息。

在需要用指示灯显示 PLC 驱动的负载（例如接触器线圈）状态时，可以将指示灯与负载并联，并联时指示灯与负载的额定电压应相同，总电流不应超过允许的值。可以选用电流小、工作可靠的 LED（发光二极管）指示灯。

可以用接触器的辅助触点来实现 PLC 外部的硬件联锁。

系统中某些相对独立或比较简单的部分，可以不进 PLC，直接用继电器电路来控制，这样同时减少了所需的 PLC 的输入点和输出点。

## 8.4 设计实例一：3 工位旋转工作台的 PLC 控制

### 8.4.1 系统描述

设计一个 3 工位旋转工作台 PLC 控制系统，其工作示意如图 8-6 所示。三个工位分别完成上料、钻孔和卸件。

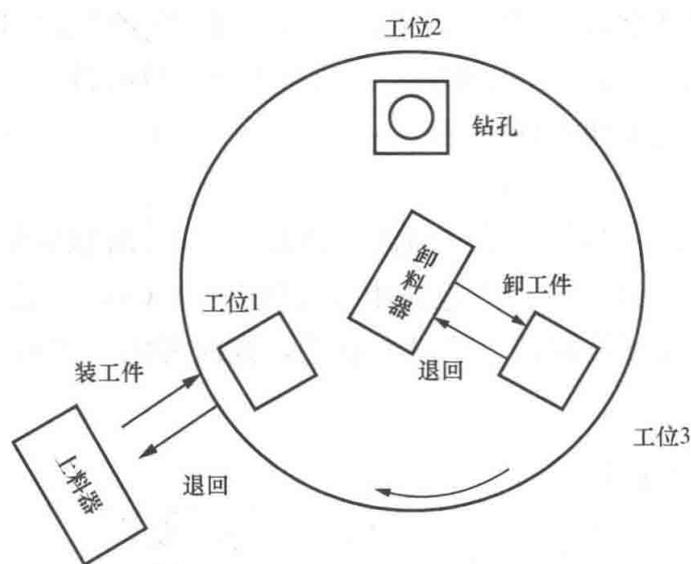


图 8-6 旋转工作台工作示意图

#### 1. 动作特性

工位 1：上料器推进，料到位后退回等待。

工位 2：将料夹紧后，钻头向下进给钻孔，下钻到位后退回，退回到位后，工件松开，放松完成后等待。

工位 3：卸料器向前将加工完成的工件推出，推出到位后退回，退回到位后等待。

#### 2. 控制要求

通过选择开关可实现自动运行、半自动运行和手动操作。

### 8.4.2 制定控制方案

(1) 用选择开关来决定控制系统的全

自动、半自动运行和手动调整方式。

(2) 手动调整采用按钮点动的控制方式。

(3) 系统处于半自动工作方式时，每执行完成一个工作循环，用一个启动按钮来控制进入下一次循环。

(4) 系统处于全自动运行方式时，可实现自动往复地循环执行。

(5) 系统运动不很复杂，采用 4 台电机。

(6) 对于部分与顺序控制和工作循环过程无关的主令部件和控制部件，采用不进入 PLC 的方法以节省 I/O 点数。

(7) 由于点数不多，所以用中小型 PLC 可以实现。可用 CPU 224 与扩展模块，或用一台 CPU 226。

### 8.4.3 系统配置及输入输出对照表

输入信号对照表见表 8-1，输出信号对照表见表 8-2。

表 8-1

输入信号对照表

信号名称	外部元件	内部地址	信号名称	外部元件	内部地址
总停按钮	SB1	不进 PLC	手动运行选择	SA4-1	I0.0
主轴电机启动停止	SA1	不进 PLC	半自动运行选择	SA4-2	I0.1
液压电机启动停止	SA2	不进 PLC	全自动运行选择	SA4-3	I0.2
冷却电机启动停止	SA3	不进 PLC	半自动运行按钮	SB11	I0.3

续表

信号名称	外部元件	内部地址	信号名称	外部元件	内部地址
上料器推进按钮	SB2	I0.4	送料推进到位行程开关	SQ1	I1.5
上料器退回按钮	SB3	I0.5	送料器退回到位行程开关	SQ2	I1.6
工件夹紧按钮	SB4	I0.6	钻头下钻到位行程开关	SQ3	I1.7
放松按钮	SB5	I0.7	钻头上升到位行程开关	SQ4	I2.0
钻头下钻控制按钮	SB6	I1.0	卸料器推出到位行程开关	SQ5	I2.1
钻头上升按钮	SB7	I1.1	卸料器退回到位行程开关	SQ6	I2.2
卸料器推出按钮	SB8	I1.2	工作台旋转到位行程开关	SQ7	I2.3
卸料器退回按钮	SB9	I1.3	工件夹紧完成压力继电器	SP1	I2.4
工作台旋转按钮	SB10	I1.4	工件放松完成压力继电器	SP2	I2.5

表 8-2

输出信号对照表

信号名称	外部元件	内部地址	信号名称	外部元件	内部地址
主轴电机接触器	KM1	不进 PLC	工件夹紧电磁阀	YV3	Q0.2
液压电机接触器	KM2	不进 PLC	工件放松电磁阀	YV4	Q0.3
冷却电机接触器	KM3	不进 PLC	钻头下钻电磁阀	YV5	Q0.4
旋转电机接触器	KM4	Q1.0	钻头退回电磁阀	YV6	Q0.5
送料推进电磁阀	YV1	Q0.0	卸料推出电磁阀	YV7	Q0.6
送料退回电磁阀	YV2	Q0.1	卸料退回电磁阀	YV8	Q0.7

#### 8.4.4 设计 PLC 外部接线图

图 8-7 为 PLC 外部接线图, 实际接线时, 还应考虑到以下几个方面:

(1) 应有电源输入线, 通常为 220V、50Hz 交流电源, 允许电源电压有一定的浮动范围, 并且必须有保护装置, 如熔断器等。

(2) 1M、2M 为输入端每个分组的公共端, 把 1M、2M 连接起来再连到 PLC24V DC 电源的 M 端。1L、2L 为输出端每个分组的公共端, 把 1L、2L 连接起来再连到交流 220V 电源的 L 端。

(3) 输出端的线圈和电磁阀必须加保护电路, 如并接阻容吸收回路或续流二极管。

#### 8.4.5 设计功能流程图

图 8-8 为手动控制部分的功能顺序图, 图 8-9 为自动半自动控制的顺序功能图, 引入初始步的条件可以是初始化脉冲 SM0.1 或者是由手动调整之后的原位信号。

#### 8.4.6 建立步与继电器对照表

辅助继电器对照表见表 8-3。

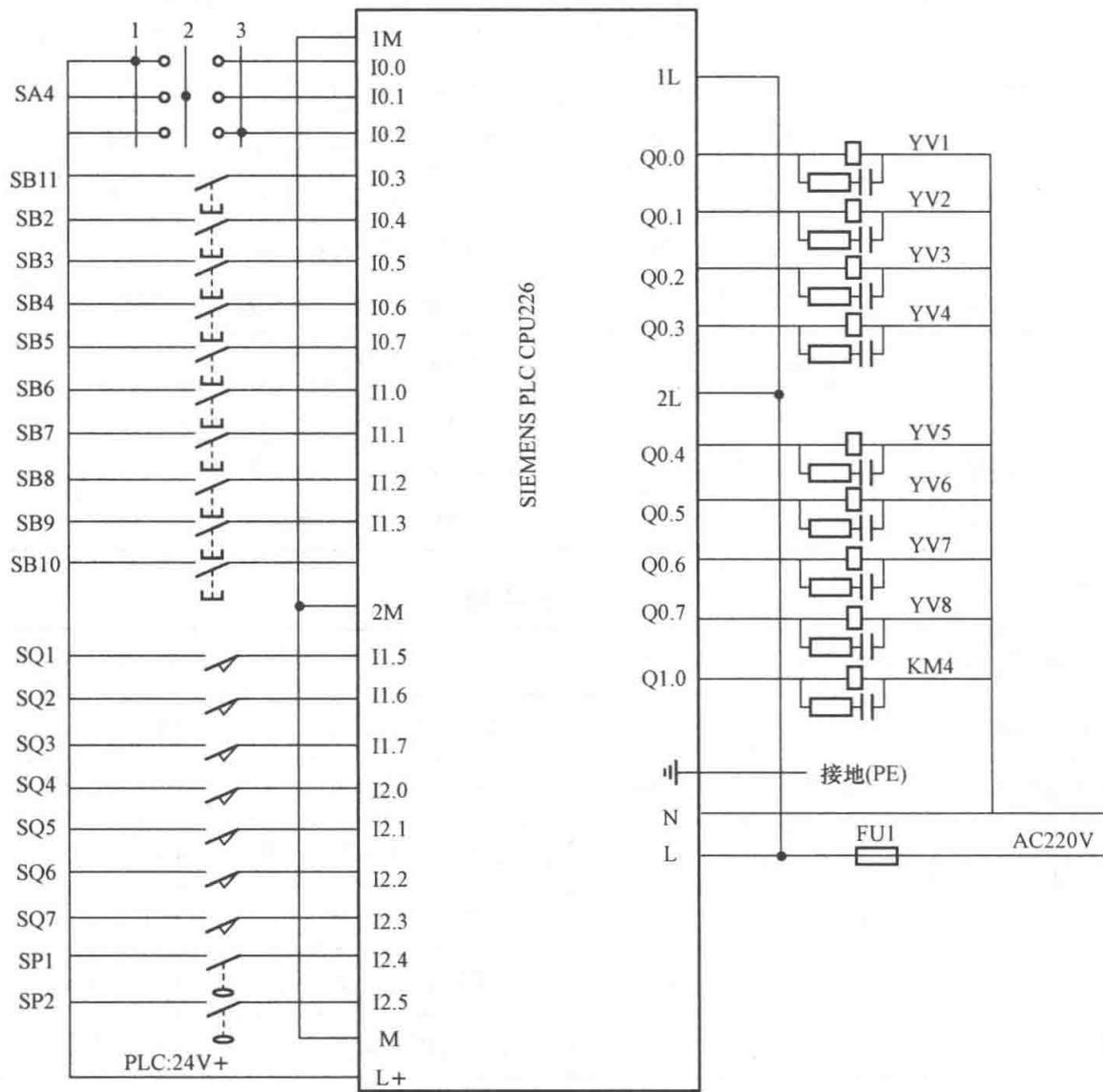


图 8-7 PLC 外部接线图

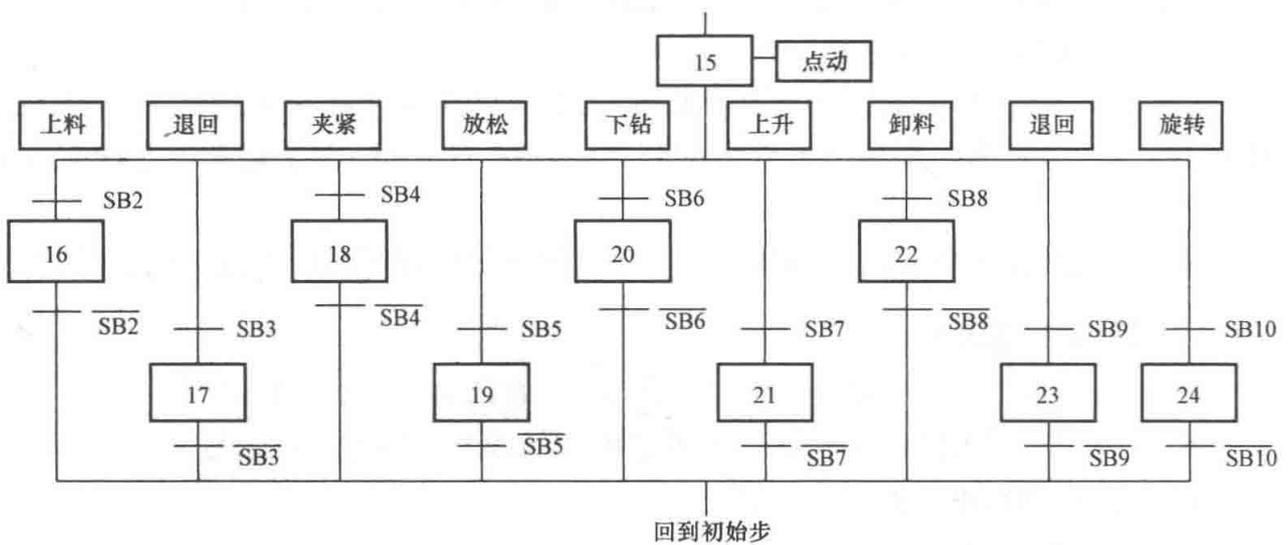


图 8-8 手动控制部分的功能顺序图

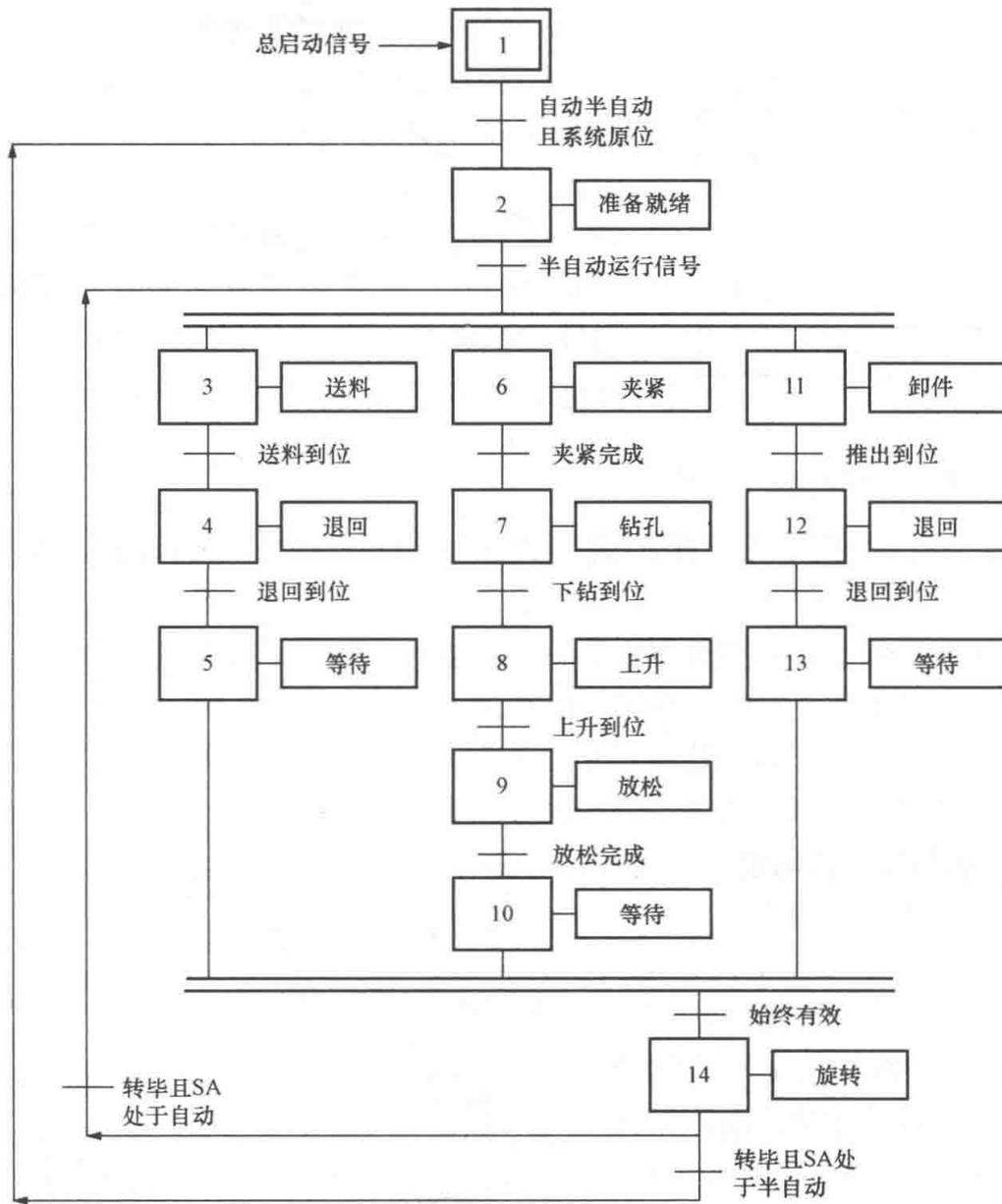


图 8-9 自动半自动控制的顺序功能图

表 8-3 辅助继电器对照表

名称	编号	内部地址	名称	编号	内部地址
初始步	1	M0.0	等待	13	M1.4
自动半自动	2	M0.1	工作台旋转	14	M1.5
送料	3	M0.2	点动调整	15	M1.6
送料器退回	4	M0.3	送料点动	16	M1.7
等待	5	M0.4	退回点动	17	M2.0
工件夹紧	6	M0.5	夹紧点动	18	M2.1
向下钻孔	7	M0.6	放松点动	19	M2.2
钻头上升	8	M0.7	下钻点动	20	M2.3
工件放松	9	M1.0	升钻头点动	21	M2.4
等待	10	M1.1	卸件点动	22	M2.5
卸工件	11	M1.2	退回点动	23	M2.6
卸料器退回	12	M1.3	旋转点动	24	M2.7

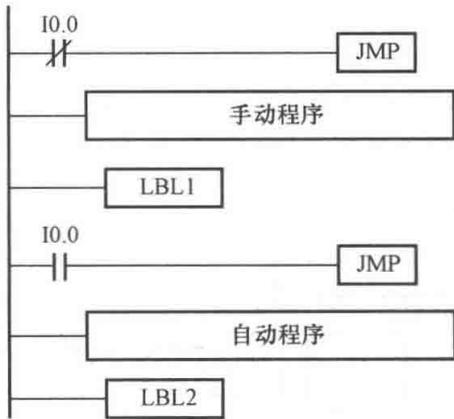


图 8-10 控制系统的梯形图总体结构

### 8.4.7 设计梯形图程序

控制系统的梯形图总体结构如图 8-10 所示，选择手动工作方式时 I0.0 为 ON，将跳过自动程序；选择自动或半自动工作方式时 I0.0 为 OFF，将跳过手动程序。根据功能顺序图自动或半自动的梯形图程序，梯形图完成后便可以将可编程控制器与计算机连接，把程序及组态数据下载到 PLC 内并进行调试，程序无误后即可结合施工设计将系统用于实际。

## 8.5 设计实例二：移动式卫生间 PLC 控制系统的设计与应用

列车用集污系统用于对铁路旅客列车厕所污物的收集，应用密闭式方法，采用真空集便技术，即厕所下水道不是直接通向线路，而是进入车厢底部悬挂的污物回收箱，到达指定车站后，再通过专用设备进行卸污处理。这就保证了机车在整个运行过程中污物不会暴露在空气中，避免沿路外排对铁路沿线的污染，符合节能及环保的要求。

### 8.5.1 系统组成及其功能

系统主要由坐便器、清水箱、污物箱、真空粉碎泵、灰水（洗漱用水等）箱收集系统、电控系统组成，如图 8-11 所示。其中冲洗系统采用气动辅助帮助冲洗。冲洗的控制、故障报警及显示控制均由可编程控制器来实现。真空泵能产生真空吸入污物，用高速旋转的刀片粉碎污物，采用压力将污物送入污物箱。污物箱和清水箱装有液位开关传感器。

系统能实现预冲、小便冲洗、大便冲洗、冲洗用水存储箱自动补水、污物箱液位显示、箱满报警、清水箱液位显示、缺水报警、洗漱用水收集、排空等功能。

### 8.5.2 工艺要求及动作分析

#### 1. 便盆湿润

感应开关触发→电磁阀 V1 打开 1s→电磁阀 V1 关闭→水增压器自动加水。

#### 2. 大便冲洗

按下冲洗按钮 SB2 大于 5s→进水电磁阀 V1 打开 2s→排污增压电磁阀 V5 打开、同时真空粉碎泵 M 启停 3 次→排污电磁阀 V4 启动、真空粉碎泵 M 启动一定时间→排污增压电磁阀 V5 打开、进水电磁阀 V1 打开约 2s→排污增压电磁阀 V5 打开、同时真空粉碎泵 2 启停 3

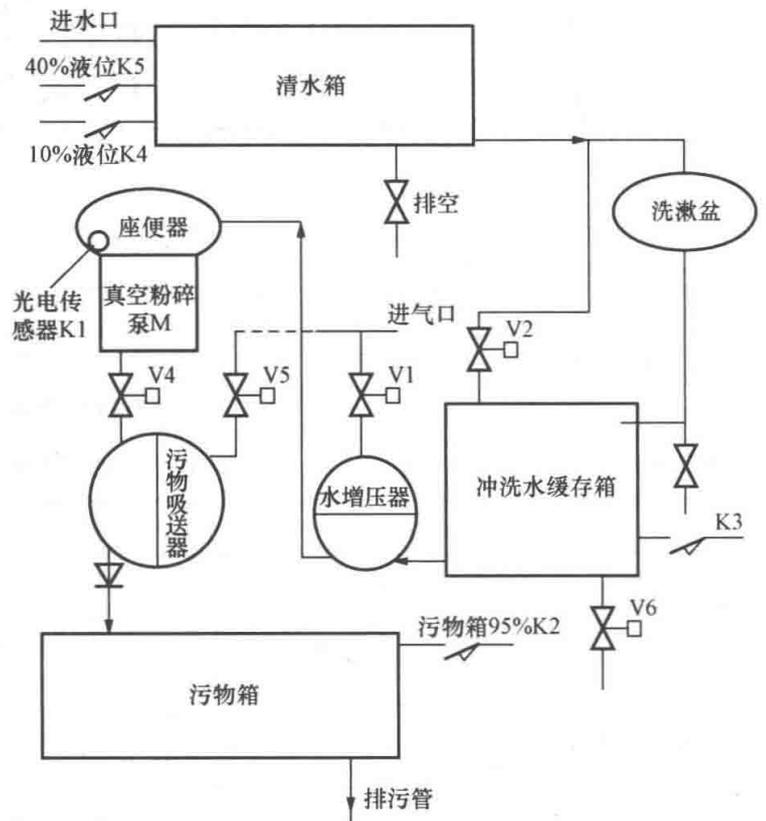


图 8-11 系统组成原理示意图

次→排污电磁阀 V4 启动、真空粉碎泵 M 启动一定时间→排污增压电磁阀 V5 打开、进水电磁阀 V1 打开约 2s→排污增压电磁阀 V5 打开→进水电磁阀 V1、排污增压电磁阀 V5 打开 1s→排污增压电磁阀 V5 打开 1s。

### 3. 小便冲洗

按下冲洗按钮 SB2 小于 5s→进水电磁阀 V1 打开 2s→排污增压电磁阀 V5 打开→真空粉碎泵 M、排污电磁阀 V4 启动 3s→排污增压电磁阀 V5 打开→进水电磁阀 V1 打开 1s。

### 4. 冲洗用水存储箱自动补水

冲洗用水存储箱自动补水液位传感器 K3 触发→补水电磁阀 V2 打开→自动从清水箱补水→补水电磁阀 V2 打开 10s 后关闭。

### 5. 污物箱液位显示、箱满报警

液位传感器 K2 触发→95%指示灯亮并报警，再使用 5 次系统停止。

### 6. 清水箱液位显示、缺水报警

液位传感器 K5 触发→40%指示灯亮；液位传感器 K4 触发→10%指示灯亮并报警，再使用 5 次系统停止工作。

### 7. 清水存储箱加水

进水阀打开→车站自来水对清水箱加水→清水箱水满后对冲洗水存储箱加水→水满后从溢流管流出→关闭进水阀。

### 8. 洗漱用水收集

洗漱用水→通过过滤网从管道自动进入冲洗水存储箱。

### 9. 排空

按下排空按钮 SB1→电磁阀 V2、V6 开启 10min 后关闭→电磁阀 V1、排污阀 V4 开停二次后关闭。两个水箱内的水从溢流口排出，水增压器和管道内水从污物箱排出。

## 8.5.3 控制系统的硬件设计

根据对工艺动作的分析，确定控制系统所需要的输入输出点数为 6/9 点。选用西门子 S7-200 系列的 CPU-224CN，输入输出点数的分配如表 8-4 所示，PLC 的外部接线图如图 8-12 所示。真空粉碎泵及信号报警指示的电源由 24V 开关电源 PS 提供，真空粉碎泵的交流电源由固态继电器控制。

表 8-4 输入输出点数分配表

信号名称	外部元件	内部地址	信号名称	外部元件	内部地址
10%液位	K4	I0.0	水箱 10%指示	L1	Q0.0
排空按钮	SB1	I0.1	冲洗指示灯	L2	Q0.1
感应开关	K1	I0.2	固态继电器	SSR3	Q0.2
冲洗按钮	SB2	I0.3	污物箱 95%指示	L3	Q0.3
灰水箱液位	K3	I0.4	进水电磁阀	V1	Q0.4
污水箱 95%液位	K2	I0.5	补水电磁阀	V2	Q0.5
			排污电磁阀	V4	Q0.7
			增压电磁阀	V5	Q1.0
			排空电磁阀	V6	Q1.1

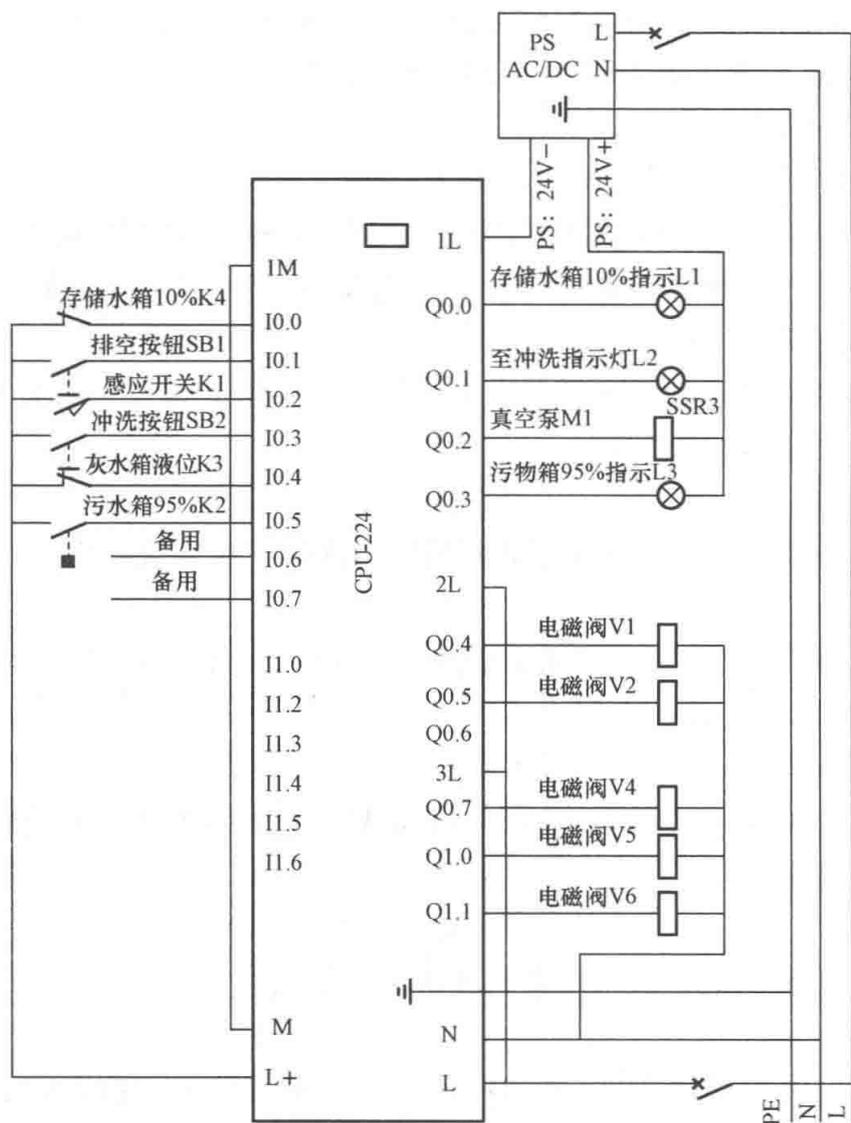


图 8-12 PLC 的外部接线图

### 8.5.4 控制程序的设计

#### 1. 输入输出变量

程序设计中所用的变量如表 8-4 所示, 其中, 小便冲洗和大便冲洗控制元件为同一个按钮, 由按钮的时间来区分。

#### 2. 程序设计

(1) 程序结构。主要的控制过程分为两大部分, 即排空操作和冲洗操作进行排空操作时绝对不允许进行补水和冲洗操作; 进行冲洗操作进行时, 也不允许进行排空操作。这可以通过软件程序来实现互锁。

#### (2) 主程序流程框图。

冲洗操作为该控制系统的主操作部分。冲洗操作包括大小便冲洗操作、水箱水位故障报警的保护和显示。污箱水位达 95% 时, 系统允许再使用 3 次冲洗操作。因此冲洗操作的程序控制最为复杂。限于篇幅, 这里只给出了总体程序结构图, 如图 8-13 所示。冲洗主程序流程框图如图 8-14 所示。

在本系统中, 因为电磁阀和固态继电器的驱动功率较少, 所以在输出回路中不用加保护元件。指示灯的耗电量更少, 所以也不需加保护元件。

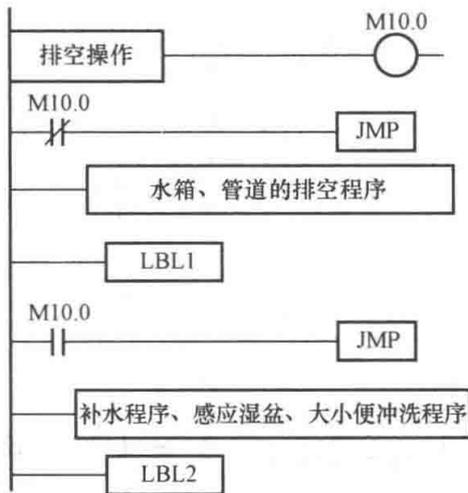


图 8-13 总体程序结构图

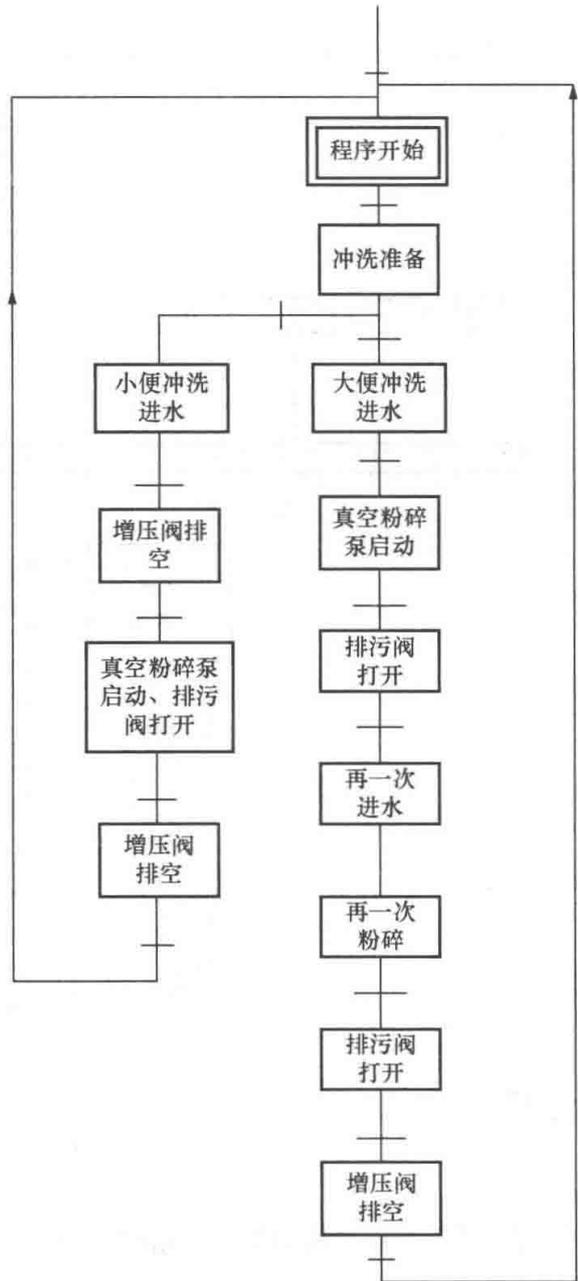


图 8-14 冲洗主程序流程框图

### 8.6 设计实例三：水箱加热系统的 PLC 位式温度控制

二位控制是位式控制规律中最简单的一种。本实例中利用 PLC 的模拟量输入输出模块实现二位式温度控制。

#### 8.6.1 系统组成

系统的组成如图 8-15 所示，本系统的被控对象是 1kW 电加热管，被控制量是水箱的水温  $T$ ，PLC 的模拟量输出控制调功器的输出，由调功器控制电加热管的通断，被控对象为水箱中的单相电热管，被控制量为水箱水温。它由铂电阻 Pt100 测定，输入到温度变送器上，量程为  $0\sim 100^{\circ}\text{C}$ 。温度变送器变换为  $4\sim 20\text{mA}$ ，传送给 PLC 的模拟量输入通道。根据给定值加上  $dF$  与测量的温度值相比较的结果，PLC 模拟量输出通道向晶闸管调功器发出控制信号，从而达到控制水箱温度的目的。

该系统的工作原理是当被控制的水温  $T$  小于给定值时，即给定值  $>$  测量值，且当  $e =$

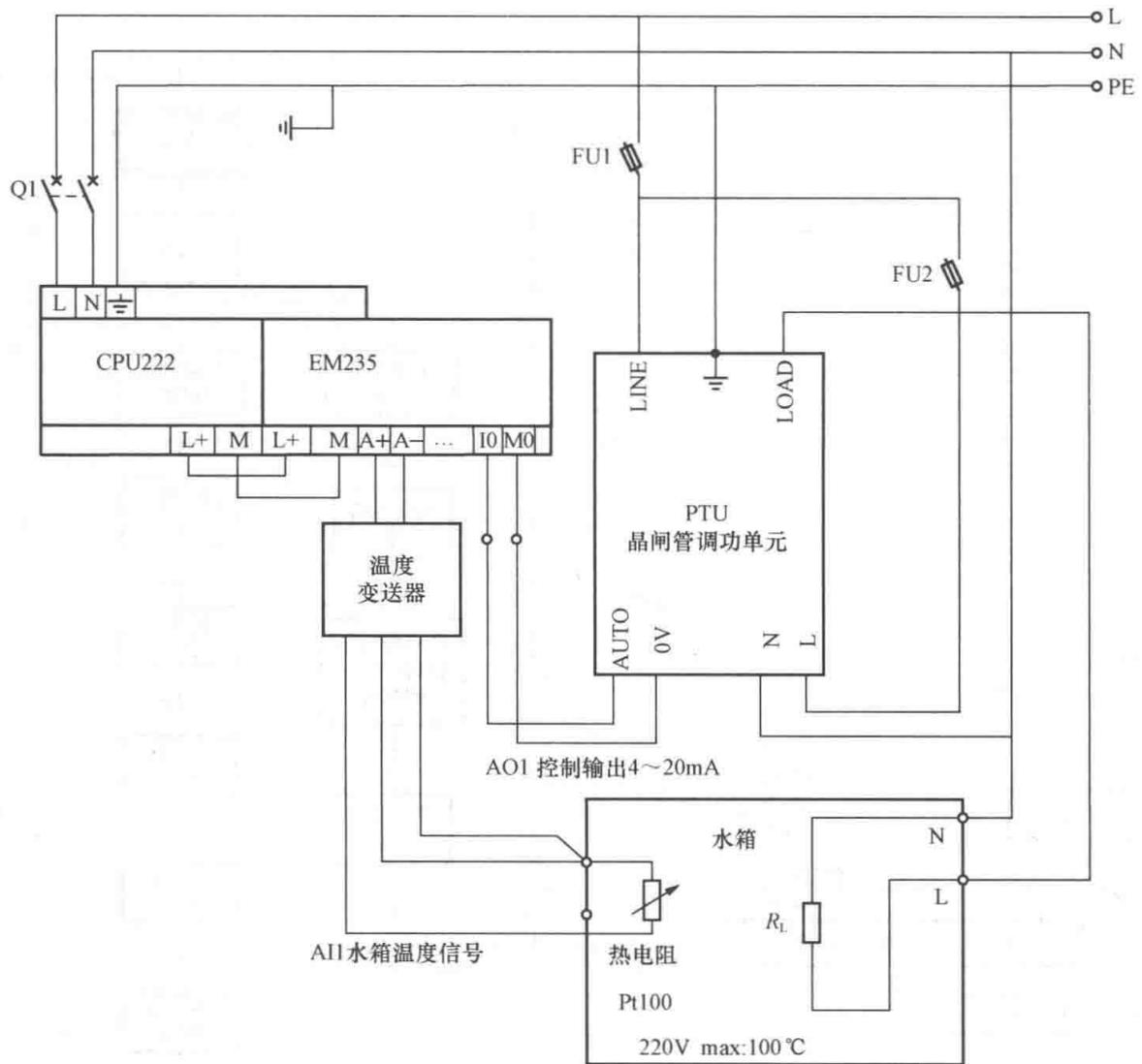


图 8-15 温度位式调节器的原理接线图

$SP - PV \geq dF$  时, 调功器的输出为最大, 电加热管接通 220V 电源而加热。随着水温  $T$  的升高,  $PV$  也不断增大,  $e$  相应变小。若水温  $T$  高于给定值, 即  $PS < PV$ ,  $e = PS - PV =$  负值; 若  $e \leq -dF$  时, 调功器的输出为最小, 切断电加热管的供电。

该系统实质上是一个典型的非线性控制系统。执行器只有“开”或“关”两种极限工作状态, 故称这种控制器为两位控制器, 取代传统的继电—接触器的两位控制, 对控制质量要求不高的系统可以使用这种方法。通常也可以利用 PLC 的开关量输出来控制固态继电器, 再由固态继电器来控制电源的通断。

### 8.6.2 PLC 及模块选择

根据对控制系统及其要求的分析, 可以选择基本的 PLC 控制单元 CPU222, 一个模拟量输入输出模块 EM235。程序控制所需要的变量如表 8-5 所示。

表 8-5 水箱温度位式控制变量表

输入通道	输出通道	设定值	手动设定值	测量值
AIW0	AQW0	VD304 温度设定值	VD 308 (回差) $dF$	VD300 水箱温度测量值
			VD 356 ( $SP + dF$ ) 设定值加回差	
			VD352 ( $SP - dF$ ) 设定值减回差	

### 8.6.3 程序设计

控制程序分为主程序与中断程序，如图 8-16、图 8-17 所示。在主程序中主要完成中断调用及数据的四则运算、控制模拟量输出通道的输出值，在中断程序中主要实现模拟量输入信号的转换。

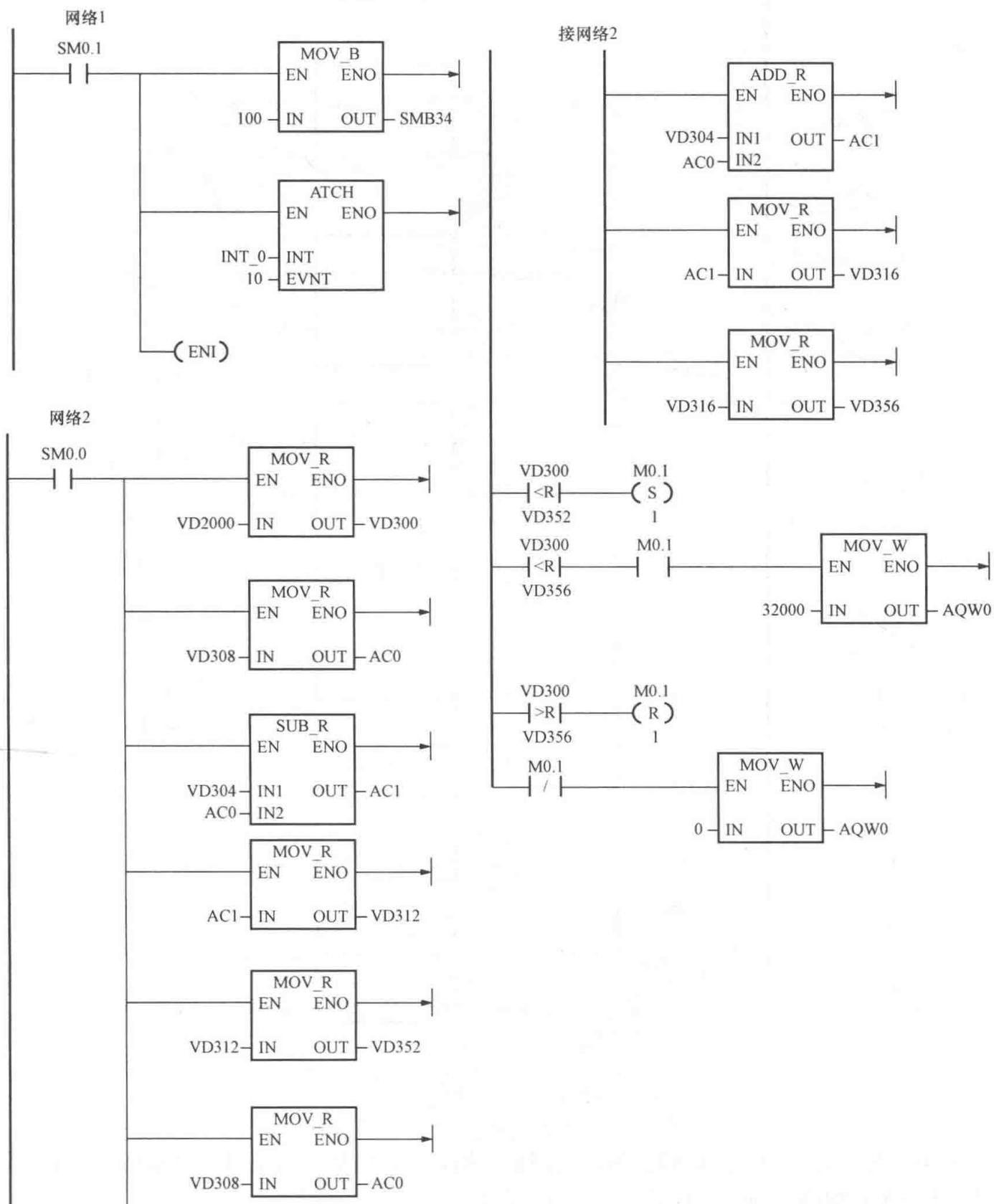


图 8-16 主程序

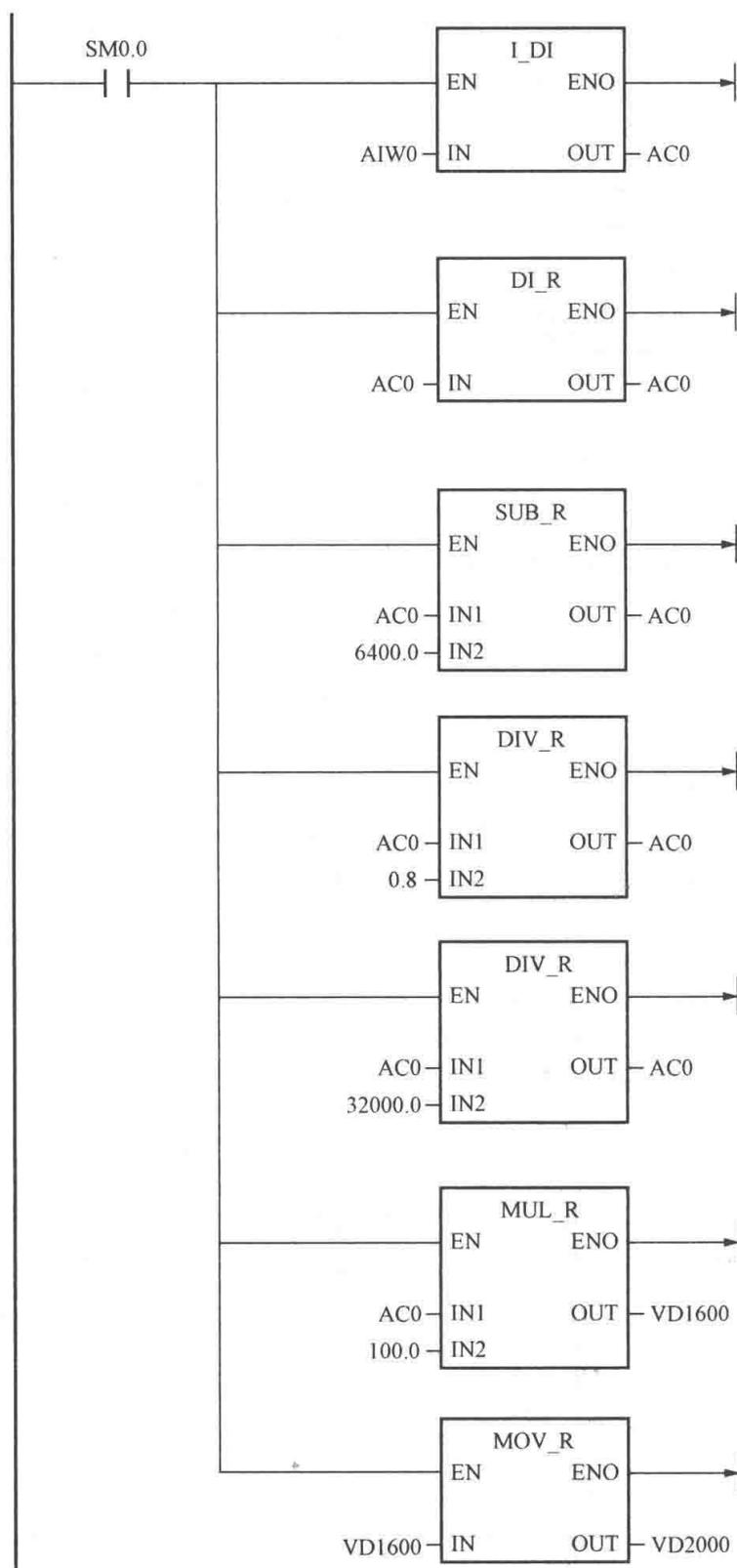


图 8-17 中断程序

在此例中，如果选择的是热电偶、热电阻测量模块，则热电偶、热电阻测量得到的数字量除以 10 便是实际的温度数值。计算过程将更为简单。

## 8.7 S7-200 SMART 系列 PLC 及其应用

### 8.7.1 S7-200 SMART 系列 PLC 的组成

S7-200 SMART 系列 PLC 硬件部分由基本单元、扩展单元、通信及网络设备、人机界面设备及其他相关设备组成，如图 8-18 所示。

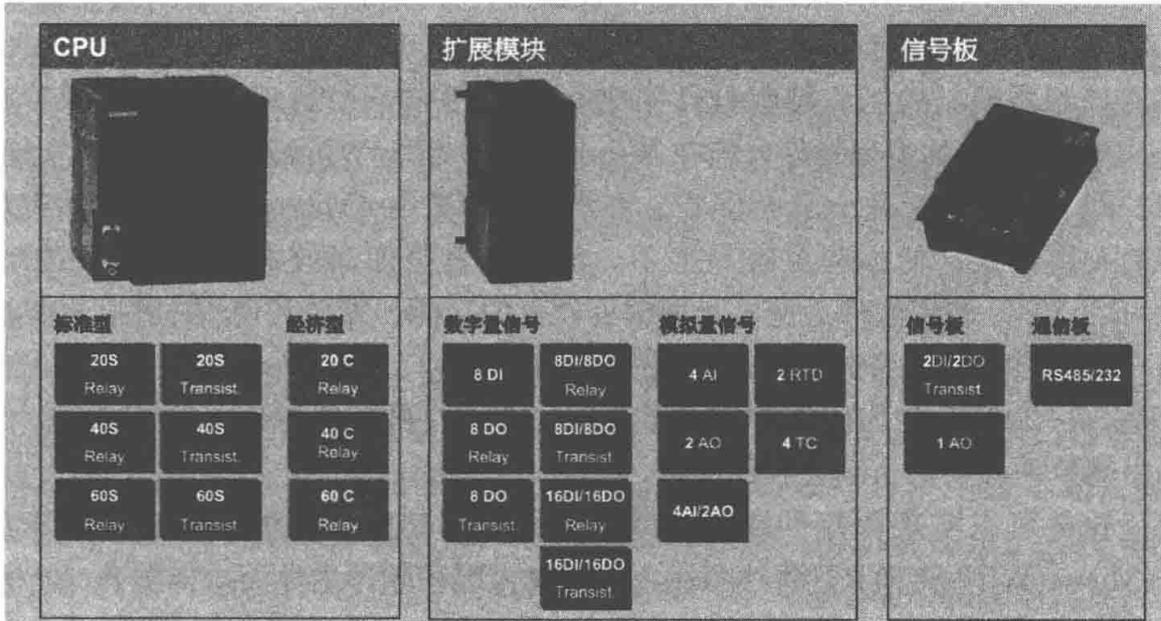


图 8-18 S7-200 SMART 系列 PLC 的硬件组成

为更好地满足应用要求，S7-200 SMART 系列 PLC 包括各种扩展模块和信号板。一般 PLC 主机及其扩展模块的宽度和厚度是固定的，长度随点数的不同而不同。

S7-200 SMART 系列 PLC 的基本单元是将微处理器 CPU、集成电源、输入电路和输出电路组合到一个结构紧凑的外壳中形成功能强大的 Micro PLC。下载用户程序后，CPU 将包含监控应用中的输入和输出设备所需的逻辑，如图 8-19 所示。

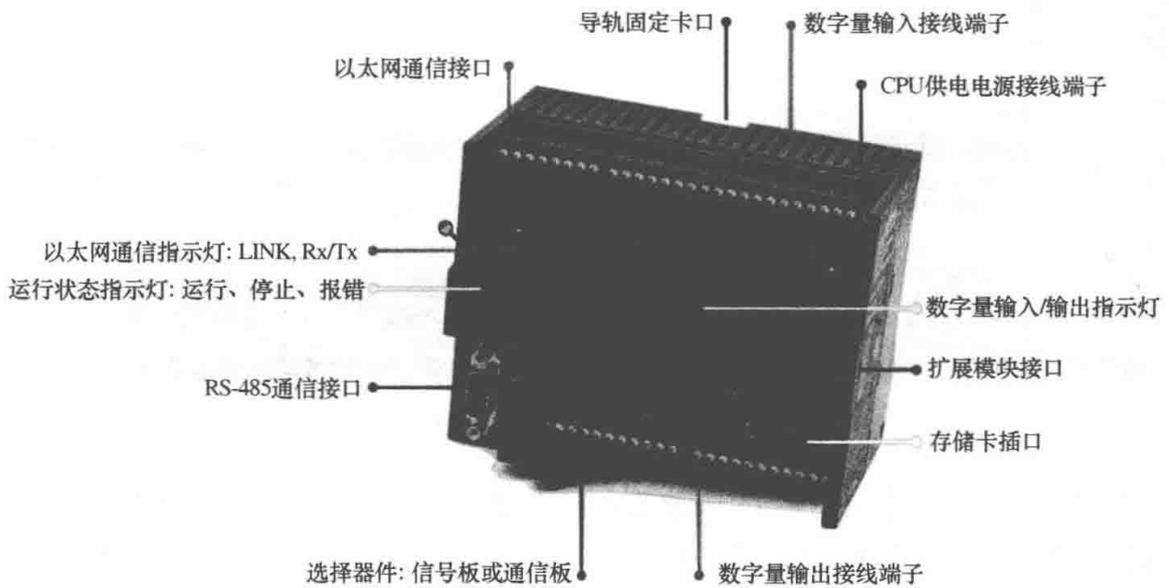


图 8-19 S7-200 SMART PLC 的基本单元

PLC 主机上的 RS-485 通信接口,用以连接编程器(手持式或 PC 机)、文本/图形显示器、PLC 网络等外部设备。

PLC 主机上的以太网接口用于直接通过以太网网络连接 PC 机或以太网交换机。以太网口连接电脑编程,只需要网线连接即可,无需其他编程电缆。

适用于 S7-200 SMART 的人机界面设备有文本显示器,还支持 Comfort HMI、SMART LINE HMI、Basic HMI、Micro HMI 等。

### 8.7.2 S7-200 SMART 系列 PLC 的编程软件

SIMATIC S7-200 SMART PLC 编程软件是指西门子公司为 S7-200 SMART 系列 PLC 可编程控制器编制的工业编程软件的集合,STEP 7-Micro/WIN SMART 是专门为 S7-200 SMART 开发的编程软件,能在 Windows XP SP3/Windows 7 上运行,支持 LAD、FBD、STL 语言。安装文件小于 100MB。在沿用 STEP 7-Micro/WIN 优秀编程理念的同时,更多的人性化设计使编程更容易上手,项目开发更加高效。本书以 STEP7-Micro/WIN SMART V2.0 编程软件为例,介绍编程软件的功能、安装和使用方法,并结合应用实例讲解用户程序的输入、编辑、调试及监控运行的方法。

只要是 S7-200CN 的代码没有 SMART PLC 不支持的特殊指令,都能将 S7-200CN 中的项目程序简单平移到这个编程软件中来。

该编程软件可以实现程序的输入、编辑、编译等功能。编程软件安装完毕后,双击 STEP 7-Micro/WIN SMART 软件图标,即可进入编程软件主界面,如图 8-20 所示。

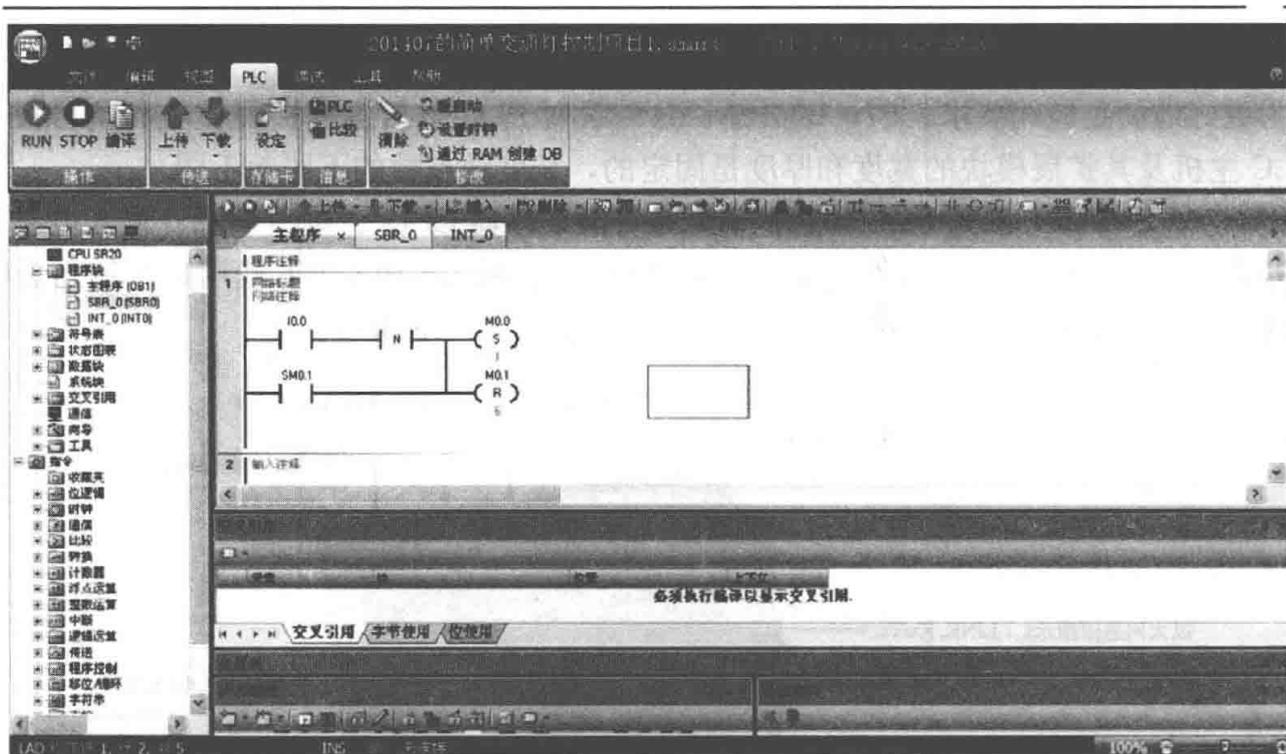


图 8-20 编程软件界面

图 8-20 中,顶部是常见任务的快速访问工具栏,其后是所有公用功能的菜单。左边是用于对组件和指令进行便捷访问的项目树和导航栏。打开的程序编辑器和其他组件占据用户界面的剩余部分。STEP7-Micro/WIN SMART 提供三种程序编辑器(LAD、FBD 和 STL),用于方便高效地开发适合用户应用的控制程序。

对比 STEP 7 - Micro/WIN 4.0 编程软件, STEP 7 - Micro/WIN SMART 有以下特点:

(1) 全新菜单设计。摒弃了传统的下拉式菜单, 采用了新颖的带状式菜单设计, 所有菜单选项一览无余, 形象的图标显示, 操作更加方便快捷。双击菜单即可隐藏, 给编程窗口提供更多的可视空间。

(2) 全移动式窗口设计。软件界面中的所有窗口均可随意移动, 并提供 8 种拖拽放置方式。主窗口、程序编辑窗口、输出窗口、变量表、状态图等均可按照用户的习惯进行组合, 最大限度地提高编程效率。

(3) 变量定义与程序注释。用户可根据工艺需求自定义变量名, 并且直接通过变量名进行调用, 完全享受高级编程语言的便利。根据实现的功能, 特殊功能寄存器调用后自动命名, 更加便捷。STEP 7 - Micro/WIN SMART 提供了完善的注释功能, 能为程序块、编程网络、变量添加注释, 大幅提高程序的可读性。当鼠标移动到指令块时, 自动显示各管脚支持的数据类型。

(4) 强大的密码保护。STEP 7 - Micro/WIN SMART 不仅对计算机中的程序源提供密码保护, 同时对 CPU 模块中的程序也提供密码保护。STEP 7 - Micro/WIN SMART 对程序源实现三重保护, 包括为工程、POU (程序组织单元)、数据页设置密码, 只有授权的用户才能查看并修改相应的内容。编程软件对 CPU 模块里的程序提供 4 级不同权限密码保护: ①全部权限 (1 级); ②部分权限 (2 级); ③最小权限 (3 级); ④禁止上载 (4 级)。

(5) 新颖的设置向导。STEP 7 - Micro/WIN SMART 集成了简易快捷的向导设置功能, 只需按照向导提示设置每一步的参数即可完成复杂功能的设定。新的向导功能允许用户直接对其中某一步的功能进行设置, 修改已设置的向导便无需重新设置每一步。向导设置支持以下功能: ①HSC (高速计数); ②运动控制; ③PID; ④PWM (脉宽调制); ⑤文本显示。

(6) 状态监控。在 STEP 7 - Micro/WIN SMART 状态图中, 可监测 PLC 每一路输入/输出通道的当前值, 同时可对每路通道进行强制输入操作来检验程序逻辑的正确性。状态监测值既能通过表格形式来显示, 也能通过比较直观的波形图来显示, 二者可相互切换。

(7) 便利的指令库。在 PLC 编程中, 一般将多次反复执行的相同任务编写成一个子程序, 将来可以直接调用。使用子程序可以更好地组织程序结构, 便于调试和阅读。STEP 7 - Micro/WIN SMART 提供便利的指令库功能, 将子程序转化成指令块, 与普通指令块一样, 直接拖拽到编程界面就能完成调用。指令库功能提供了密码保护功能, 防止库文件被随意查看或修改。另外, 西门子公司提供了大量完成各种功能的指令库, 均可轻松添加到软件中。

### 8.7.3 S7-200 SMART 系列 PLC 的应用

#### 1. 编程软件的安装

关闭所有的应用程序, 在光盘驱动器中插入驱动光盘, 如果没有禁止光盘插入自动运行, 安装程序会自动进行, 或者在 Windows 资源管理器中打开光盘上的“Setup.exe”, 也可以把光盘资料复制到硬盘中, 单击“Setup.exe”进行安装。使用默认的安装语言 (简体中文), 然后按照安装程序的提示完成安装。安装完成后最好重启计算机。

#### 2. 建立项目或打开已有的项目

可以用新建菜单建立一个项目, 可以用打开菜单打开已有的项目, 也可以用另存为菜单保存项目并修改项目名称, 还可以在文件菜单中为项目设置密码。

项目的基本组件包括：①程序块由主程序、可选的子程序和中断程序组成；②用于给 V 存储区地址分配数据初始值的数据块；③用于给 S7-200 SMART CPU、信号板、扩展模块组态与设置各种参数的系统块；④用于给变量添加注释的符号表，符号表中所定义的符号为全局变量，可以用于所有的程序；⑤状态图表可以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。

### 3. 硬件的组态

该步的目的就是要生成一个与实际硬件系统相对应的系统，包括各模块和信号板的参数设置，选择的 CPU 基本单元及扩展模板或信号板要与实际的硬件安装位置和型号相对应。

### 4. 建立 PC 机与 S7-200 SMART CPU 的通信

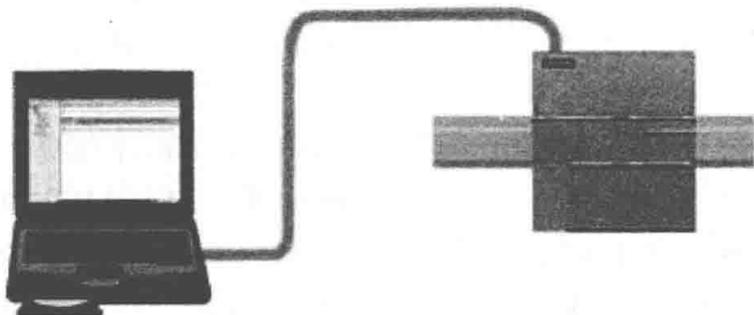


图 8-21 PC 机与 S7-200 SMART CPU 的连接

用以太网线连接电脑和 CPU 的以太网接口，如图 8-21 所示。首先组态 CPU 的以太网地址，有三种方法。

(1) 用系统块设置 CPU 的 IP 地址，打开“系统块”对话框，出现如图 8-22 所示的视图，在右边窗口中设置以太网地址及 RS-485 通信端口的参数，如果勾选了图中的“IP 地址数据固定为下面的值，不能通过其他方式更改”对话框，则 IP 地址只能在系统块对话框中更改并将其下载到 CPU。

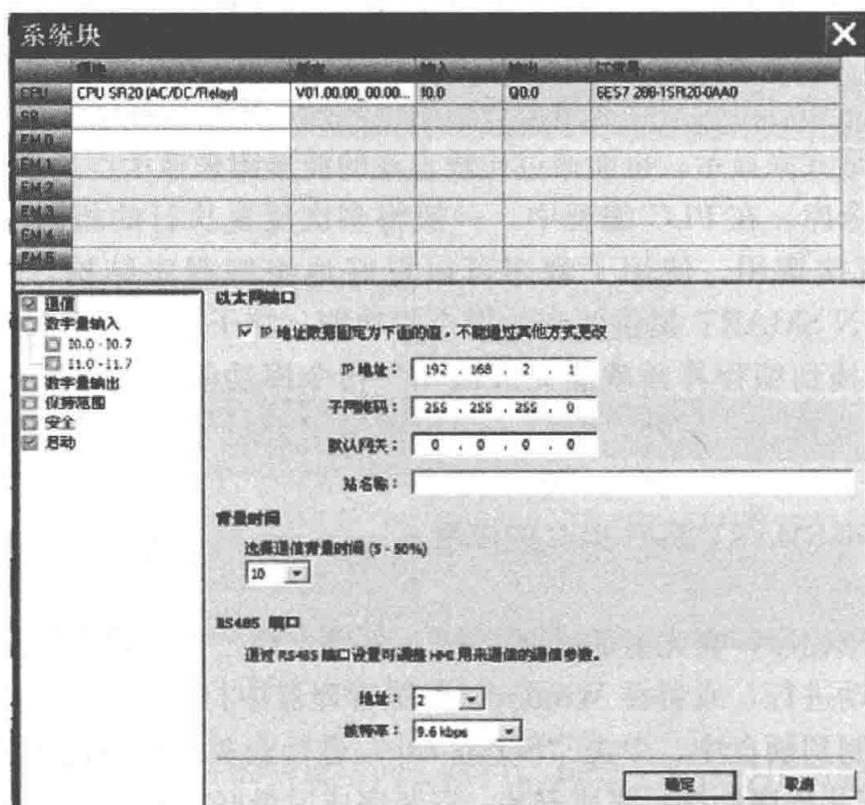


图 8-22 CPU 以太网参数的设置

(2) 用通信对话框设置 CPU 的 IP 地址，双击项目树中的“通信”图标，在“网络接口

卡”下拉式菜单中选择 PC 机所使用的以太网卡，双击“更新可访问的设备”，左边窗口将会自动显示出网络上可访问的设备的 IP 地址，如图 8-23 所示。如果网络上有多个 CPU，选中与编程计算机通信的 CPU，就可建立和相应的 CPU 的连接，电脑就可以下载程序到该 CPU 并可监控该 CPU，或者上载 CPU 中的程序到空白项目中。

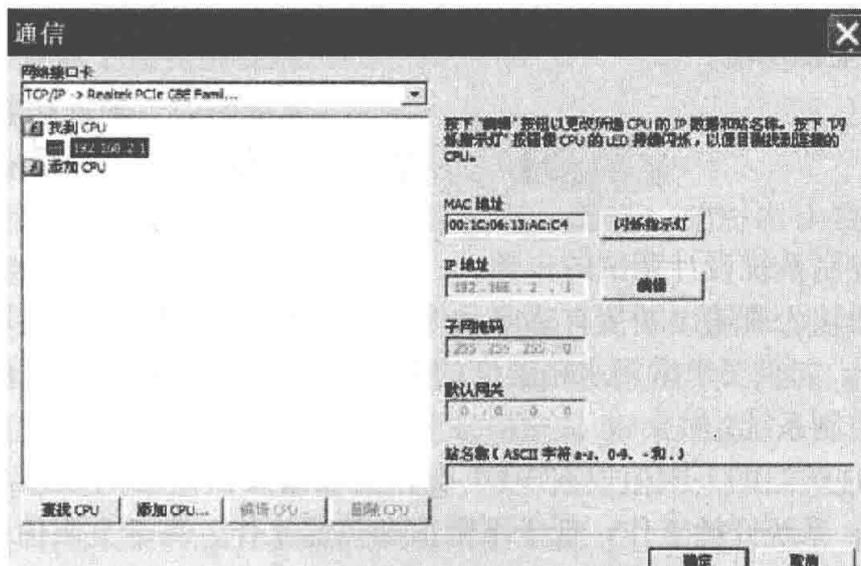


图 8-23 CPU 的通信对话框

(3) 在程序中设置 IP 地址，利用 SIP - ADDR 指令用参数 ADDR、MASK、GATE 分别设置 CPU 的 IP 地址、子网掩码及网关。

然后再设置计算机的 IP 地址，这与我们通常设置计算机网卡的方法、步骤基本相同。注意 IP 地址和子网掩码地址应设置在与 CPU 相对应的范围内，另外，一对一通信不需要以太网交换机，网络中有两个以上的设备时需要以太网交换机。

#### 5. 根据控制要求编写程序

通常利用梯形图编辑器进行梯形图程序的编辑、修改，程序的编辑包括程序的剪切、复制、粘贴、插入和删除，字符串替换、查找等。

#### 6. 程序的编译及上、下载

(1) 编译。程序的编译，能明确指出错误的网络段，编程者可以根据错误提示对程序进行修改，然后再次编译，直至编译无误。

(2) 下载。用户程序编译成功后，将下载块中选中的内容下载到 PLC 的存储器中。

(3) 载入（上载）。载入可以将 PLC 中未加密的程序或数据向上送入编程器（PC 机）。将选择的程序块、数据块、系统块等内容上载后，可以在程序窗口显示上载的 PLC 内部程序和数据信息。

#### 7. 程序的监视、运行、调试

(1) 程序运行方式的设置。将 CPU 的工作方式开关置在 RUN 位置，或将开关置在 TERM（暂态）位置时，操作 STEP7-Micro/WIN SMART 2.0 菜单命令或快捷按钮对 CPU 工作方式进行软件设置。

(2) 程序运行状态的监视。运用监视功能，在程序状态打开下，可以到观察 PLC 运行时，程序执行的过程中各元件的工作状态及运行参数的变化。

(3) 变量的强制操作。在实验室调试程序时，往往会因为没有现场实际设备而使得用户

程序无法正常调试。STEP 7 - Micro/WIN SMART 提供了软件“强制”功能,允许用命令的形式来改变程序中的各变量的值,使调试过程简单快捷。

#### 8. 仿真调试

S7 - 200 仿真软件既适用于 S7 - 200 系列 PLC,也适用于 S7 - 200 SMART 系列 PLC,利用仿真软件能模拟 S7 - 200 SMART PLC 的基本指令功能,但不能模拟 S7 - 200 SMART PLC 的全部指令和全部功能。

### 习 题

1. 简述 PLC 控制系统设计调试的步骤。
2. 如果 PLC 的输入或输出端接有感应元件,应采取什么措施来保证 PLC 的正常运行?
3. 在实例三中,如果要求采用水箱温度的 PID 控制,该如何设计程序?
4. 炉温 PLC 控制系统。

要求:假定允许炉温的下限值放在 VD1 中,上限值放在 VD2 中,实测炉温放在 VD10 中,按下启动按钮,系统开始工作,低于下限值加热器工作,高于上限值停止加热,上、下限之间维持。按下停止按钮,系统停止。试设计该系统。

#### 5. 两种液体混合装置的 PLC 控制系统的设计。

要求:有两种液体 A、B 需要在容器中混合成液体 C 待用,初始时容器是空的,所有输出均失效。按下启动信号,阀门 X1 打开,注入液体 A;到达 I 时, X1 关闭,阀门 X2 打开,注入液体 B;到达 H 时, X2 关闭,打开加热器 R;当温度传感器达到 60℃ 时,关闭 R,打开阀门 X3,释放液体 C;当最低位液位传感器 L=0 时,关闭 X3 进入下一个循环。按下停车按钮,要求停在初始状态。如图 8 - 24 所示。

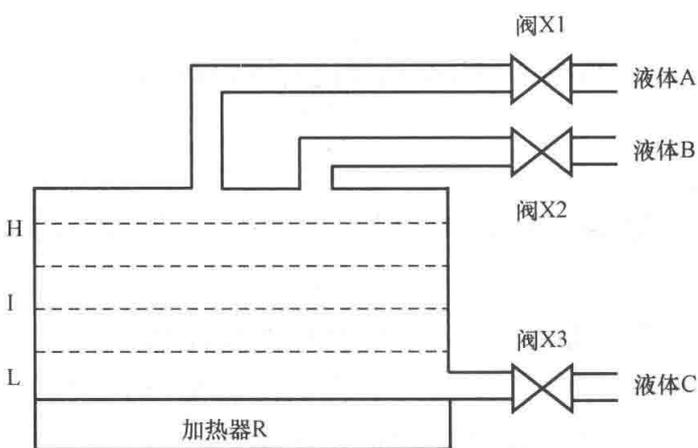


图 8 - 24 题 5 的图

启动信号 I0.0, 停止信号 I0.1, 高液位 H (I0.2), I (I0.3), L (I0.4), 温度传感器为 K 分度热电偶, 阀门 X1 (Q0.1), 阀门 X2 (Q0.2), 阀门 X3 (Q0.3), 加热器 R (Q0.4)。

#### 6. 三自由度机械手控制系统的设计。控制要求:

- (1) 在初始位置 (上、左、松限位开关确定) 处, 按下启动按钮, 系统开始工作。
- (2) 机械手首先向下运动, 运动到最低位置停止。
- (3) 机械手开始夹紧工件, 一直到把工件夹紧为止 (由定时器控制)。
- (4) 机械手开始向上运动, 一直运动到最上端 (由上限位开关确定)。
- (5) 上限位开关闭合后, 机械手开始向右运动。
- (6) 运行到右端后, 机械手开始向下运动。
- (7) 向下到位后, 机械手把工件松开, 一直到松限位开关有效 (由松限位开关控制)。
- (8) 工件松开后, 机械手开始向上运动, 直至触动上限位开关 (上限位开关控制)。

(9) 到达最上端后, 机械手开始向左运动, 直到触动左限位开关, 此时机械手已回到初始位置。

(10) 要求实现连续循环工作。

(11) 正常停车时, 要求机械手回到初始位置时才能停车。

(12) 按下急停按钮时, 系统立即停止。

要求设计程序完成上述控制要求。

7. 自动钻床控制系统。控制要求:

(1) 按下启动按钮, 系统进入启动状态。

(2) 当光电传感器检测到有工件时, 工作台开始旋转, 此时由计数器控制其旋转角度(计数器计满 2 个数)。

(3) 工作台旋转到位后, 夹紧装置开始夹工件, 一直到夹紧限位开关闭合为止。

(4) 工件夹紧后, 主轴电机开始向下运动, 一直运动到工作位置(由下限位开关控制)。

(5) 主轴电机到位后, 开始进行加工, 此时用定时 5s 来描述。

(6) 5s 后, 主轴电机回退, 夹紧电机后退(分别由后限位开关和上限位开关来控制)。

(7) 接着工作台继续旋转由计数器控制其旋转角度(计数器计满 2 个)。

(8) 旋转电机到位后, 开始卸工件, 由计数器控制(计数器计满 5 个)。

(9) 卸工件装置回到初始位置。

(10) 如再有工件到来, 实现上述过程。

(11) 按下停车按钮, 系统立即停车。

要求设计程序完成上述控制要求。

## 第9章 HMI的组态与应用

PLC控制系统的监控有很多种方法，有用简单的文本显示器实现监控的，有用触摸屏实现监控的，也有用PC机实现监控的。文本显示器实现监控是一种低成本的方式，由相关的编程软件或向导对文本显示器进行设置，就可实现对PLC控制系统的简单监控。使用触摸屏的监控方式需要在PC上进行项目的组态开发设计，设计好的项目要下载到触摸屏中，一般不能由PC机直接监控，需要由触摸屏实现对PLC控制系统的监控。要在PC机上实现对PLC控制系统的监控，需要有相应的软件平台也就是组态软件。组态软件是数据采集及监视控制SCADA的软件平台，具有系统集成功能，还能够使生产过程可视化，因此基于PC的组态监控具有十分广泛的应用。

### 9.1 人机操作界面

HMI是人机界面（Human Machine Interface）的简称，是控制系统与操作人员交换信息的设备。人机界面能在环境较恶劣的场合使用，已成为现代工业实现自动化控制必不可少的设备之一。

传统的人机控制操作界面包括指示灯、主令按钮、开关和电位器等。操作人员通过这些设备把操作指令传输到自动控制器中，控制器也通过它们显示当前的控制数据和状态。这是一个综合的人机交互界面。

随着技术的进步，新的模块化的、集成的人机操作界面产品被开发出来。这些HMI产品一般具有灵活的可由用户（开发人员）自定义的信号显示功能，用图形和文本的方式显示当前的控制状态；现代HMI产品还提供了固定或可定义的按键，或者触摸屏输入功能。

#### 9.1.1 HMI设备的组成及工作原理

人机界面设备由硬件和软件两部分组成，硬件部分包括处理器、显示单元、输入单元、通信接口、数据存储单元等，其中处理器的性能决定了HMI产品的性能高低，是HMI的核心单元。根据HMI的产品等级不同，处理器可分别选用8、16、32位的处理器。HMI软件一般分为两部分，即运行于HMI硬件中的系统软件和运行于PC机Windows操作系统下的画面组态软件（如JB-HMI画面组态软件）。使用者都必须先使用HMI的画面组态软件制作“工程文件”，再通过PC机和HMI产品的串行通信口，把编制好的“工程文件”下载到HMI的处理器中运行。

HMI设备作为一个网络通信主站与PLC或其他智能设备相连，因此也有通信协议、站地址及通信速率等属性。通过串行通信在两者之间建立的数据对应关系，也就是CPU内部存储区与HMI输入/输出元素间的对应关系。比如HMI上的按键对应于CPU内部Mx.x的数字量“位”，按下按键时Mx.x置位（为“1”），释放按键时Mx.x复位（为“0”）；或者HMI上某个一个字（Word）长的数值输入（或者输出）域，对应于CPU内部V存储区VWx，如图9-1所示。

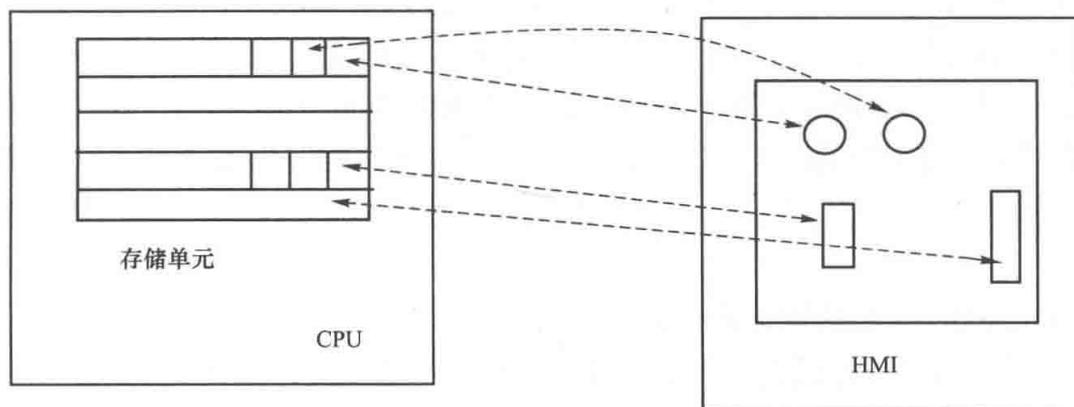


图 9-1 CPU 存储器和 HMI 元素的对应关系

只有建立了这种对应关系，操作人员才可以与 PLC 的内部用户程序建立交互关系。这种联系，以及 HMI 上究竟如何安排、定义各种元素，怎样进行软件的设置，一般统称为“组态”。各种不同的 HMI 各自有组态的软件和方法。

### 9.1.2 HMI 设备的功能

HMI 设备的作用是提供自动化设备操作人员与自控系统（PLC 系统）之间的交互界面接口。使用 HMI 设备，可以实现以下功能：

- (1) 在 HMI 上显示当前的控制状态、过程变量，包括数字量（开关量）和数值等数据。
- (2) 显示报警信息。
- (3) 通过硬件或可视化图形按键输入数字量、数值等控制参数。
- (4) 使用 HMI 的内置功能对 PLC 内部进行简单的监控、设置等。

### 9.1.3 HMI 设备的分类

HMI 设备主要分为三类：触摸屏、文本终端（文本显示器）、平板电脑（PC 机），其中以触摸屏为代表产品。有许多公司生产触摸屏和文本显示器，本书中主要介绍西门子系列的触摸屏和文本显示器。

薄膜键输入的 HMI，显示尺寸小于 5.7in，画面组态软件免费，属初级产品。如文本显示器 TD 400C。

触摸屏输入的 HMI，显示屏尺寸为 5.7~12.1in，画面组态软件免费，属中级产品。如 SMART LINE 系列触摸屏、KTP 系列单色或彩色触摸显示屏等，带有 RS-485 接口或以太网接口，可通过 MPI 电缆、PROFIBUS 电缆或以太网线到连接 PLC 或智能设备的通信口。

基于 PC 计算机的、多有种通信口的高性能 HMI，显示尺寸大于 10.4in，画面组态软件收费，属高端产品。如基于组态王、WINCC、MCGS、力控等组态软件开发设计的监控项目。

## 9.2 文本显示器的组态与应用

### 9.2.1 概述

用文本显示器实现的 PLC 控制系统监控是一种低成本监控方式。可编程逻辑控制器与文本显示的配套连接使用在工业控制领域获得了广泛的应用，它外观简洁美观、小巧精致，

通常用在环保、制冷、空调、工业控制及自动控制的其他领域。它具有价格低廉、操作简便、界面友好和兼容性好的特点,在控制系统应用中大大减少了控制柜操作面板的按钮,也避免了使用价格较贵的触摸屏,在适应各种控制要求时极其灵活,因此深受广大中小企业的喜爱。TD 系列文本显示器是专门用于 S7-200 系列 PLC 或 S7-200 SMART 系列 PLC 的文本显示和操作员界面。其中 TD 400C 支持中文操作和文本显示,是应用较多的一种文本显示器。

### 9.2.2 TD 400C 监控的设计及应用

TD 400C 属于一种智能设备,自身带 F1~F8 8 个按键和 SHIFT+F1~SHIFT+F8 8 个组合按键,以及 Esc、Enter 等按键。TD 400C 还具有以下功能和特点:

- (1) 读取并显示 PLC CPU 的信息。
- (2) 提供为具有实时时钟的 CPU 设置时间和日期的功能。

(3) 可以改变程序变量,允许强制和取消强制 I/O 点。

(4) 可以单独供电,也可以由 PLC CPU 通过 TD/CPU 电缆供电。如果采用外部电源,必须使用 DC 24V 的电源。

(5) TD 400C 通过 TD/CPU 电缆与 CPU 通信,可以进行一对一配置和多个 CPU 配置。

由于 TD 400C 具有上述独特之处,对于输入/输出点数不是特别多(一般不超过 16 个)的系统来说,TD 400C 的优点就十分明显,可以在配置、用户菜单和报警等 3 个栏目根据控制要求进行开发设计,可以实现控制系统的开停机、定时、数字量输入/输出、模拟量显示、当前时间显示、设备运行状态显示故障报警和显示及手动复位故障消除等功能。

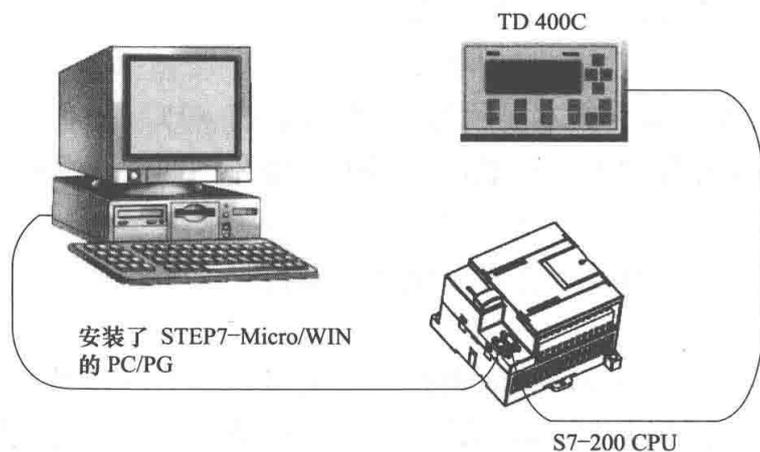


图 9-2 文本显示器与 PLC 的连接

图 9-2 给出了连接编程设备、S7-200 CPU 和 TD 400C 的实例。图中 TD 400C 的组态是在编程设备上使用组态软件来创建的。在组态后,TD 400C 即可与 S7-200 CPU 进行通信。

#### 1. TD 400C 配置

可以使用 Keypad Designer 和“文本显示向导”对 TD 400C 进行文本显示消息和其他数据的组态。但使用“文本显示向导”进行配置更加方便快捷,STEP 7-Micro/WIN 软件文本显示向导可以指导用户快速地完成 TD 400 的配置或组态。只有当 STEP 7-Micro/WIN 的语言设置为中文时才可以用来组态 TD 400C。

打开 STEP 7-Micro/WIN 软件,单击工具栏的“工具”,出现下拉菜单,选中“文本显示向导”,单击“下一步”,出现选择“TD 型号和版本”界面,只要在其列表中选 TD 400C 版本 2.0,再单击“下一步”,出现“标准菜单和更新频率”页面,此页面不必理会,这一步可以采用其默认方式,直接单击“下一步”,出现“本地化显示”或“语言设置”页面,在语言栏选择所需要的语言,然后单击“下一步”,进入“配置键盘按钮”页面,在列表中列出了按键名称,就是前面说到的 8 个单按键和 8 个组合按键,“按键符号”表示按键和组合按键的代号,以便在 PLC 程序中可以方便的查看到;“按键动作”栏下可以设置按键

是“点动或者长动”，在对应行的置位后单击，出现下拉菜单，在列表下有两个选项，置位就是长动，瞬动触点就是点动，如图 9-3 所示。根据实际控制要求来选择点动还是长动，在所有配置完成及程序编译正确后按下该按钮时，在 TD 400C 屏幕上出现一个提示信号，并伴有声音提示，表示按下的键得到 CPU 的响应。感兴趣的同学可以自己试试，到此单击“下一步”就完成了 TD 的配置，接下来就可以进行用户菜单和报警的设置。

## 2. 用户菜单的介绍和应用举例

TD 200 V3.0 及以上版本支持菜单组态方式，最多可配置 8 个菜单，每个菜单下最多可以组态 8 个文本显示屏，最多可以配置 64 个文本显示屏。用户可以使用面板上的箭头按键在各菜单及显示屏之间自由切换，菜单屏可以嵌入 S7-200 数据变量。

下面以模拟量输入信号的测量显示及温度数据的设定为例，介绍如何进行参数的显示和设定。

第一步：在 TD 配置页面中，单击页面左边的用户菜单，进入用户菜单设置。在最上一行的空白栏输入“模拟量测量菜单画面 1”，在第二行的空白栏输入“设定值”，如图 9-4 所示。

第二步：添加屏幕并编辑屏幕信息。在对应的菜单栏单击右侧的添加屏幕，出现一个对话框“为此菜单项添加一个屏幕吗？”，单击“是”进入菜单项“模拟量测量菜单画面 1”的屏幕，在屏幕 0 这个画面中，可以显示“数据采集 1”和“数据采集 2”等模拟量测量值，在屏幕 1 这个画面中，可以显示“数据采集 2”，如图 9-5 所示。



图 9-3 文本显示器键盘按钮设置

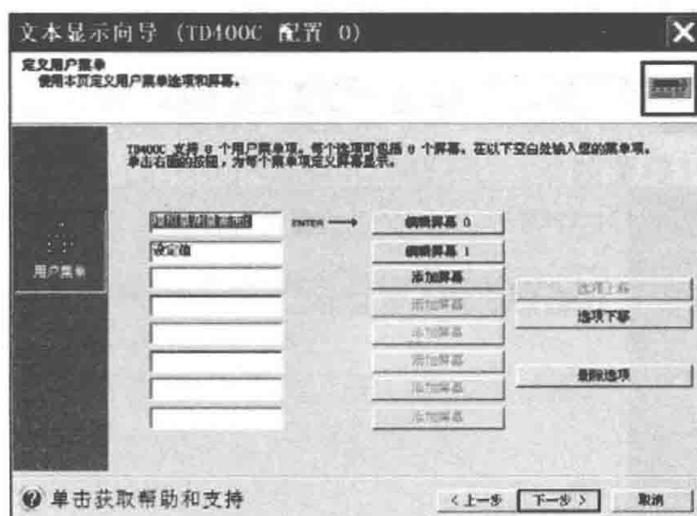


图 9-4 选择菜单击定义用户菜单



图 9-5 菜单屏幕编辑

第三步：在显示画面中嵌入 PLC 的数据，在如图 9-5 所示的屏幕中的“数据采集 1”之后的光标处，单击“插入 PLC 数据”，出现“插入 PLC 数据”页面，在“数据地址”栏输入在程序中用于显示模拟量测量值的变量，例如“VW100”或者 VD100，前者是字变量，后者是双字变量；在“数据格式”栏中可以选取有符号或者无符号，在“小数点右侧位数”栏可以选取用于表示数据的小数点的位数，显示数据当然只能显示，不能设置更改，如图 9-6 所示。

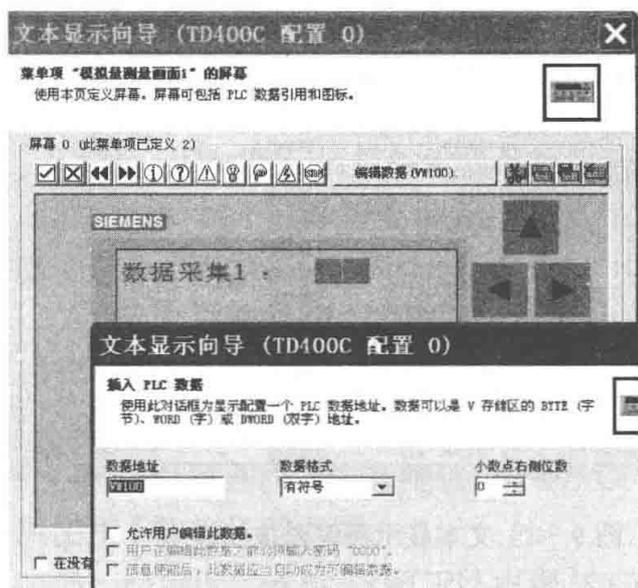


图 9-6 PLC 数据定义

第四步：编辑“设定值”菜单的屏幕 0，在“设定值”菜单的屏幕 0 中输入“温度设定”，在“插入 PLC 数据”的数据地址栏中输入用于温度设定值的变量，与上述不同的是该变量需要由文本显示器进行设置，因此该变量若需要编辑，只需选择“允许用户编辑此数据”，其他操作与模拟量测量值显示的操作一样。

一切设置好后，当程序下载到 CPU 后，通过按键 Esc 和 Enter 进入用户菜单，可以读取 PLC 输出的实际模拟量测量值，也可以进行温度设定。在文本显示器监控 PLC 时，只要通过按 Esc 键，进入用户菜单中的“模拟量测量菜单画面 1”，按 Enter 键就可以显示“数据采集 1”的值，用“▲”或“▼”键移动上下行或翻页，当光标落在“温度设定”的数据上时，再按一下 Enter 键，“温度设定”后面的数字开始闪烁，表示用户可以编辑此数据了，通过“▲”或“▼”键可以增加或者减小设定值，设定完毕再按 Enter 确认，这样设定的新值就赋给了 CPU 内对应的变量了。

### 3. 报警设置及应用举例

TD 400C 还可以显示多达 80 条报警消息，报警消息的显示与否由 TD 400C 的组态及 CPU 中报警消息的使能位的状态决定。报警画面中也可以嵌入 S7-200 PLC 的数据变量。

报警是 TD 400C 一个很重要的功能，当在程序里设定报警条件和报警信息后，把程序下载到 CPU 后，如果报警条件被使能，屏幕上会显示报警提示，操作员可以根据报警提示查看报警内容及消除报警。

第一步：单击报警弹出定义报警页面，按“下一步”进入报警选项页面，如图 9-7 所示。

第二步：选择文本显示的行数及显示模式，再按“下一步”，出现对话框“您希望在此 TD 400 配置中增加一条报警吗”，单击“是”，出现报警画面。如图 9-8 所示，在画面中输入报警信息，比如“急停按钮被按下”，如果勾选了“此报警要求

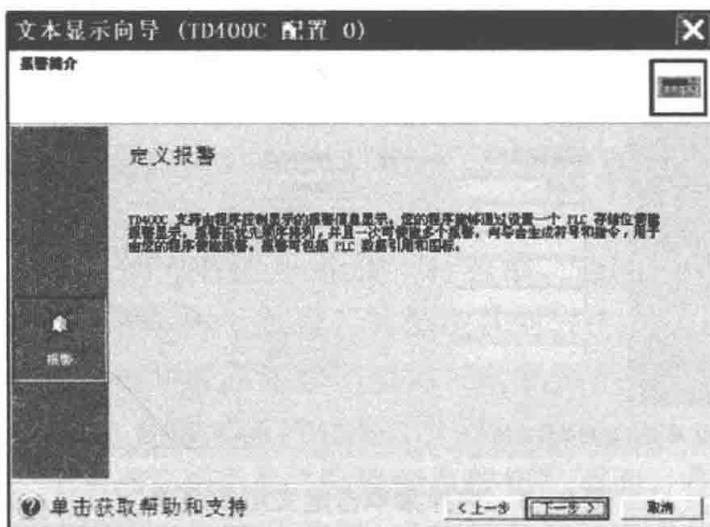


图 9-7 定义报警页面

操作员确认”，在当报警出现后，操作员除了要排除故障复位后，还需要按 Enter 键进行确认，方可继续操作。在此页面的下部，有“此报警的符号名”和“报警确认位”两报警项，前者是报警信息使能标志，后者则是报警确认条件，在程序执行的过程中，如果报警位被使能，报警信息就显示出来，在屏幕上以闪动提醒，此时，可以通过功能键 Esc 和 Enter 进行翻页、退出等操作。在 PLC 的程序中设计一段如图 9-9 所示的程序，如果急停按钮（对应 I0.3）被按下，即 PLC 的输入点 I0.3 接通，就激发了报警，报警的使能位是 0，地址为 V46.7，符号是 Alarm0\_0，这时屏幕上就出现报警闪烁，操作员通过功能键可以查看到报警内容为“急停按钮被按下”，当需要操作员确认时，必须按下 Enter 键，方能消除报警。还可以按“新报警”添加下一条报警，报警编辑的方法与上述相同。



图 9-8 编辑报警页面

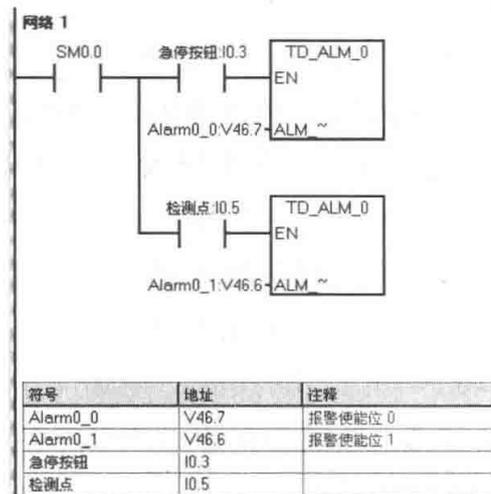


图 9-9 与报警有关的程序

#### 4. TD 400C 的系统菜单操作

按 Esc 键，进入文本显示器的菜单方式，可用的菜单项目有：

- (1) DISPLAY ALARMS（显示报警消息）。
- (2) OPERATOR MENU（操作员菜单）。在此菜单中用户可以查看 CPU 状态，设置 CPU 的实时时钟，及完成 TD 200 的语言切换。
- (3) DIAGNOSTIC MENU（诊断菜单）。在这一菜单中可以查看 TD 200 的文本信息、报警消息，强制 I/O 点及对 TD 200 进行设置。其中 TD 200 的正确设置是保证 TD 200 与 S7-200 正确通信的关键，用户可以在这个菜单中完成 CPU 地址、TD 200 地址、通信波特率、参数块地址的设定。
- (4) PASSWORD PROTECT（密码保护）。可以通过此功能进行密码保护设置。

### 9.3 触摸屏的组态与应用

西门子的人机界面软件包括 WinCC 和 WinCC flexible。WinCC 是过程可视化系统监控组态平台，用来进行单用户或多用户的组态设计和运行监控，可以组态基于 PC 的可视化工

作站,是实现大规模自动化系统集成的组态软件。WinCC flexible 是早期 ProTool 与 ProTool/Pro 软件的升级产品,可以组态所有 SIMATIC 操作面板(触摸屏),用于工厂和机械工程中机器级的操作员控制和自动化过程监测。

### 9.3.1 触摸屏组态软件 WinCC flexible 的特点

本书以 WinCC flexible 2008 SP4 为例来讲解触摸屏组态软件的特点及应用举例。WinCC flexible 2008 SP4 具有以下特点:

(1) 基于最新软件技术的创新性组态界面,集成了 ProTool 的简易性、耐用性和 WinCC 的开放性、扩展性。

(2) 功能块库:可自定义及重复使用各种功能块,并可对其进行集中更改。

(3) 高效率的组态:动态面板、智能工具。

(4) 使用用户 ID 或密码进行访问保护。

(5) 配方管理。

(6) 报表系统。

(7) 提供广泛的语言支持,在一个项目中可组态 32 种语言。

(8) 支持多语言文本和自动翻译的文本库。

(9) 提供简单的文本导入/导出功能。

(10) 开放简易的扩展功能。

### 9.3.2 组态软件的应用方法

#### 1. Wincc flexible 的启动

启动 WinCC flexible,单击“开始”→SIMATIC→WinCC flexible 2008→WinCC flexible 选项,或者直接单击桌面上对应的图标。

#### 2. 直接创建项目

使用 WinCC flexible 可创建不同类型的项目。WinCC flexible 的项目可以使用项目向导来创建,也可以直接创建项目。本书以直接创建项目为例。打开 WinCC flexible 软件后,执行“项目”菜单中的“新建”命令或单击“创建一个空项目”,来新建一个项目,单击“新建”后,出现设备选择对话框,根据现场设备来选择 HMI 设备与控制器,其型号必须与实物相符合。单击“确定”按钮,完成项目的建立,如图 9-10 所示。所选的设备为 SMART 700。

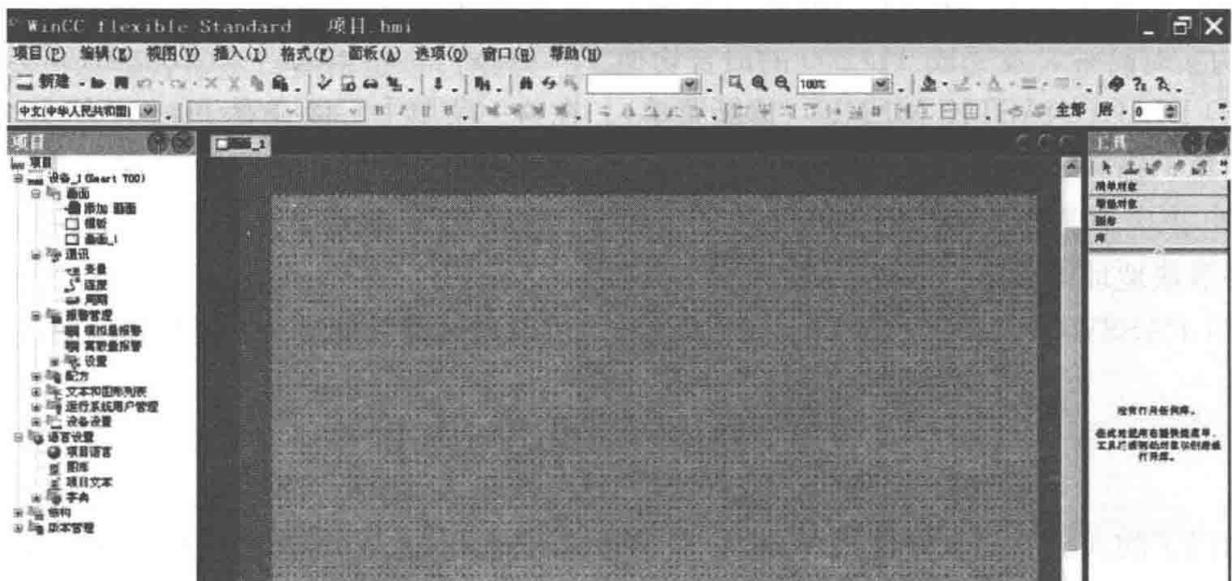


图 9-10 直接创建项目

接下来需要建立 HMI 设备与控制器之间的连接。双击项目视图中“通讯”文件夹下的“连接”，打开连接编辑器建立连接。在“通讯驱动程序”的下拉菜单中选择与 HMI 设备相连接的控制器，设置 HMI 设备与控制器之间的连接方式及相关的参数，如图 9-11 所示。

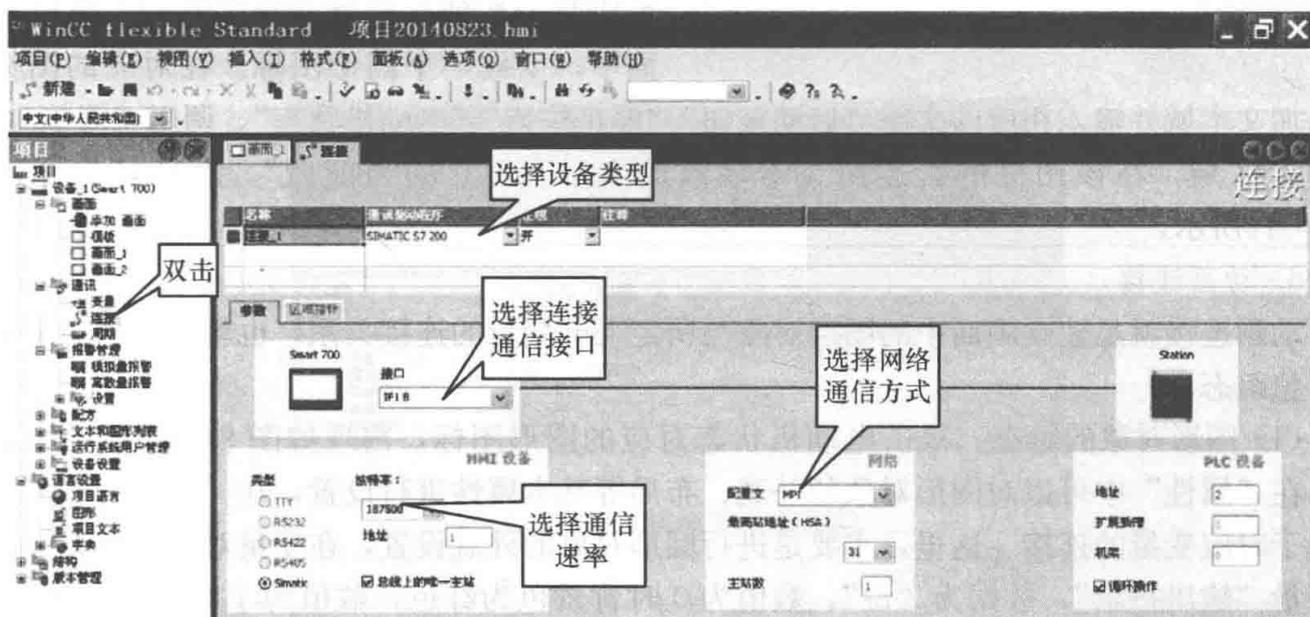


图 9-11 设备的连接

### 3. 设计 PLC 控制器中的程序

在此设计一个简单的电动机启停控制加延时控制的程序，如图 9-12 所示。M1.0 和 M1.1 是触摸屏画面中启动和停止按钮对应的变量，VW10 是接通延时定时单元 T40 的当前值，可以用触摸屏上的启动和停止按钮来控制电动机（由输出点 Q0.3 控制）的启停，用画面中的指示灯显示 Q0.3 的状态。

### 4. 定义变量

变量是数据库的主要组成部分，而数据库数据库是组态软件的核心部分，工业现场的生产状况要以动画的形式反映在屏幕上，操作者在计算机前发布的指令也要迅速送达生产现场，所有这一切都是以实时数据库为中介环节，所以说数据库是联系上位机和下位机的桥梁。变量定义时要指定变量名和变量类型，某些类型的变量还需要一些附加信息。选择工程浏览器左侧大纲项的“变量”，在工程浏览器右侧用鼠标右键单击“添加变量”图标，添加一行，在“变量名称”一栏把变量名称修改为“启动按钮”，在“连接”一栏选择所连接的设备，在“数据类型”一栏选择

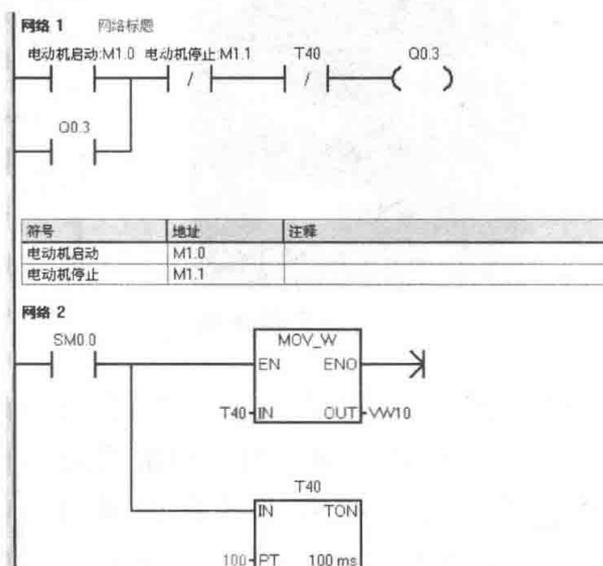


图 9-12 程序段落

“BOOL”，在“地址”一栏选择“M1.0”，在“采集周期”一栏可以设置采集时间，一般选择 500ms。用同样的方法添加“停止按钮”变量、“输出指示”变量、“时间显示”变量，但“时间当前值显示”变量是字类型变量，选择“VW10”。所定义的变量如图 9-13 所示。

名称	连接	数据类型	地址	数组计数	采集周期
停止按钮	连接_1	Bool	M 1.1	1	500 ms
输出指示	连接_1	Bool	Q 0.3	1	500 ms
时间当前值指示	连接_1	Int	VW 10	1	500 ms
启动按钮	连接_1	Bool	M 1.0	1	500 ms
变量_9	连接_1	Bool	M 11.1	1	1 s
变量_8	连接_1	Bool	Q 0.0	1	100 ms

图 9-13 变量的定义

面添加文本域并输入相应的文字“启动按钮”“停止按钮”“电动机状态”。调用“简单对象”中的 I/O 域，在该图标的右边用文本域添加文字“T40 的当前值”。设计好的画面如图 9-14 所示。

### 6. 动画连接

动画连接就是建立画面中的图形对象与所定义的变量的连接关系，也就是为画面中的图形对象组态。

(1) 图形对象的组态。双击电动机状态对应的图形图标，出现如图 9-15 所示的对话框，在“属性”中可以对图形对象的外观、布局等基本属性进行设置，在“动画”中可以建立起于对应变量的连接。这里，主要是进行圆形对象的外观设置，在变量对话框中选择对应的变量“输出指示”，数据为“位”，数值为 0 时背景色为红色，数值为 1 时背景色为绿色。这样组态好后，进入监控时，如果变量 Q0.3 为 1，则该图形显示为绿色；如果变量为 0，则该图形显示为红色。

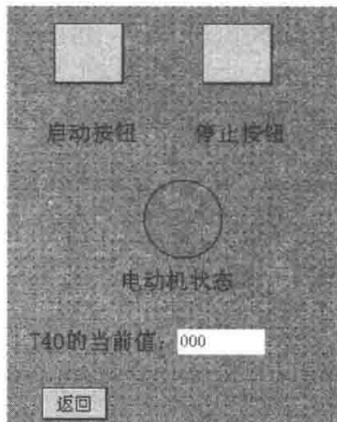


图 9-14 设计监控画面

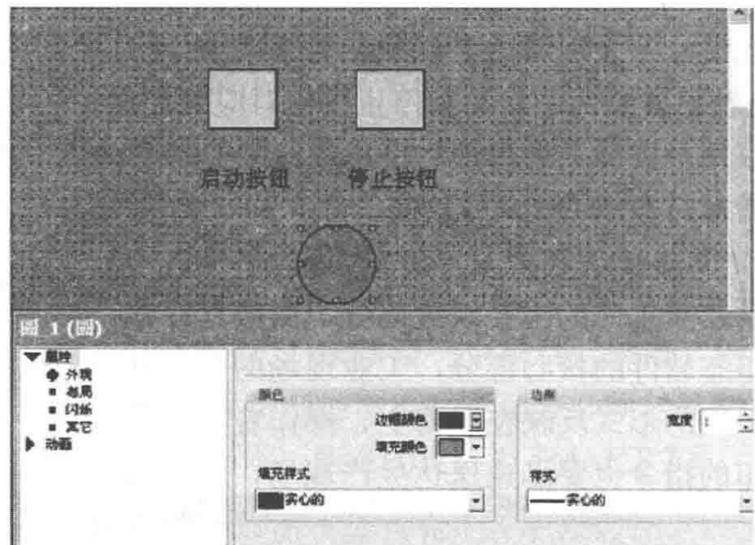


图 9-15 图形对象的组态

(2) 按钮图形的组态。按钮是 HMI 设备上的虚拟键，可以用来控制生产过程。按钮的模式共有以下 3 种：①文本按钮：根据按钮上显示的文本可以确定按钮的状态。②图形按钮：根据按钮上显示的图形可以确定按钮的状态。③不可见按钮：该按钮在运行时不可见。本例中常规属性里选择图形按钮，如图 9-16 所示。

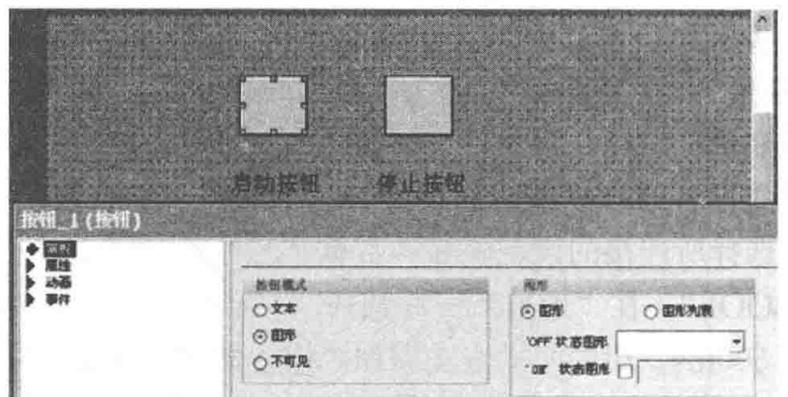


图 9-16 按钮对象的组态

### 5. 监控画面的设计

按添加画面，会出现一个新画面，默认画面名称“画面 4”，在右边工具箱的“简单对象”里选择“按钮”图标，拖放到画面中，生成两个按钮，选择“圆”图标，拖放到画面中，生成一个圆形图标。在对应的图标下面

在“事件”选项下的“按下”对话框中，单击右边窗口最上面一行“无函数”对应的下拉菜单，选择系统函数列表的“编辑位”子目录下的函数“SetBit”，直接单击表中第二行右边的所隐藏的下拉菜单，选择先前所定义过的变量“启动按钮”，如图9-17所示。在“事件”选项下的“释放”对话框中，采用与上述类似的方法调用系统函数“ResetBit”，将变量“启动按钮”复位为OFF。

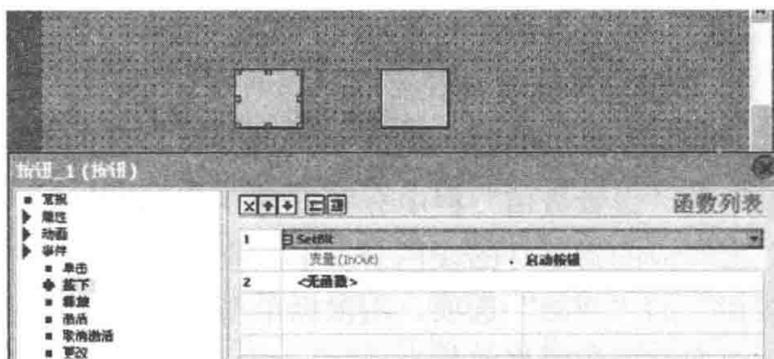


图9-17 按钮的动画连接

对“停止按钮”所对应的按钮的动画组态与对“启动按钮”所对应的按钮的动画组态的方法相同。

(3) I/O域的组态。I/O域有3种模式：①输出域（用于显示变量的数值）；②输入域（用于将输入的数值保存到控制器所对应的变量中）；③输入/输出域（同时具有输入和输出功能，可以修改PLC中变量的数值并将修改后的数据显示出来）。

双击I/O域所对应的图标，在属性视图的“常规”组，模式选择为“输出域”，过程变量选择为“时间当前值显示”。

#### 7. 项目的下载和监控

将组态好的项目保存，先将PLC的程序下载至PLC，然后将PC机上的触摸屏组态项目下载至对应型号的触摸屏中。

单击组态软件WinCC flexible工具栏上的按钮，打开“选择设备进行传送”对话框，设置通信模式为“USB/PPI多主站电缆”，选择触摸屏上的“传送”，完成组态项目的下载。下载成功后，触摸屏自动返回运行状态，显示下载的项目的初始画面。

用两端都为9针接头的DP或MPI电缆连接SMART 700触摸屏与S7-200 PLC，接通电源，PLC处于RUN模式，在触摸屏上按启动按钮对应的图标，输出点Q0.3启动，对应的电动机状态显示为绿色，按停止按钮对应的图标，输出点Q0.3复位，时间当前值显示与时间单元T40相对应，数值在不断变化。电动机启动后也可由定时时间来实现停止。

#### 8. 使用仿真器启动运行系统

如果没有实物触摸屏，可以用运行模拟器来模拟触摸屏的部分功能，用于调试已组态的触摸屏的基本功能，可以模拟画面的切换、变量的数值显示或位变量的状态等。

单击工具栏上的按钮，启动仿真器。系统会自动启动编译，只有在编译成功后才能模拟运行。图9-18为画面监控，图9-19为运行模拟器。



图 9-18 画面监控



图 9-19 运行模拟器

在变量“输出指示”的“设置数值”栏中分别设置 0 或 1, 按“开始”选项, 可以看到画面上“电动机状态”所对应的指示灯的颜色会发生相应的变化

选择变量“启动按钮”的“开始”选项, 用鼠标单击启动按钮对应的图标, 可以看到按下按钮时运行模拟器中对应的变量的当前值为-1 (表示为 ON 状态), 松开鼠标时当前值为 0 (表示为 OFF 状态)。

在变量“输出指示”的“数值设置”列输入常数 123, 按键盘确认键, 可以看到画面上 T40 的当前值显示出相同的数值。

#### 9. 用控制面板设置触摸屏的参数

以型号为 SMART 700 的触摸屏设备为例, 接通电源启动触摸屏后, Smart 700 的屏幕点亮, 几秒后显示进度条。启动后出现“装载程序”对话框。如果触摸屏已经装载了项目, 在出现装载对话框后经过设置的延时时间, 将自动打开项目。

也可以装载新的项目文件来覆盖触摸屏中原有的项目文件, 接通电源后, 在屏幕出现“装载程序”对话框时, 可以单击“Transfer”进行项目文件的下载。

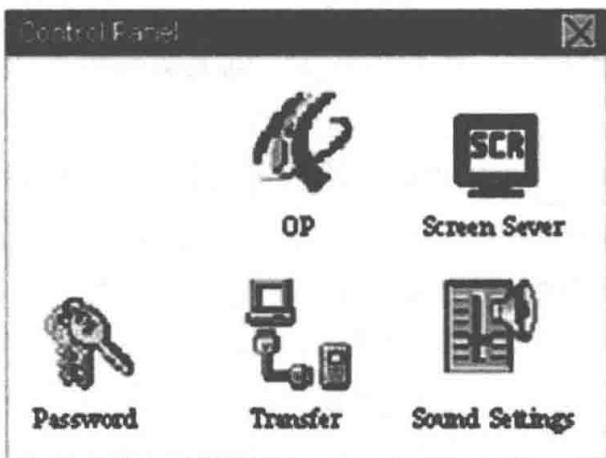


图 9-20 触摸屏的控制面板

可以单击图屏幕上的“Control Panel”按钮, 打开触摸屏的控制面板 (见图 9-20), 用控制面板设置触摸屏的各种参数, 如设置启动延时时间、密码、声音, 还可查看设备信息和许可信息等, 具体设置内容和方法请参见有关手册。

双击控制面板中的“Transfer”图标, 打开“传输设置”对话框。选中“Channel 1” (通道 1) 域中串行端口 (Serial) 的“Enable Channel” (激活通道) 复选框, Smart 700 使用 RS-485/422 端口与 PLC 通信。

## 9.4 基于 PC 的组态与应用

### 9.4.1 组态王简介

“组态王”是在流行的 PC 机上建立工业控制对象人机接口的一种智能软件包, 它以 Windows XP/Windows 7 中文操作系统作为其操作平台, 充分利用了 Windows 图形功能完

备、界面一致性好、易学易用的特点。它使采用 PC 机开发的系统工程比以往使用专用机开发的工业控制系统更有通用性，大大减少了工控软件开发者的重复性工作，并可运用 PC 机丰富的软件资源进行二次开发。目前，组态王 6.53 以下的版本适合于 Windows XP 系统，组态王 6.55 以上的版本适合于 Windows XP/Windows 7 系统。

“组态王”软件包由工程管理器 (ProjManager)、工程浏览器 (TouchExplorer)、画面运行系统 (TouchView) 3 部分组成。

(1) 工程管理器用于新建工程、工程管理等。

(2) 工程浏览器是“组态王”软件的核心部分和管理开发系统，它将画面制作系统中已设计的图形画面、命令语言、设备驱动程序管理、配方管理、数据库访问配置等工程资源进行集中管理，并在一个窗口中以树形结构排列，这种功能与 Windows 操作系统中的资源管理器的功能相似。工程浏览器内嵌画面开发系统，即组态王开发系统。工程浏览器内嵌画面开发系统，进入画面开发系统的操作方法有以下两种：①在工程浏览器的上方图标快捷菜单中用左键单击“MAKE”图标。②在工程浏览器左边窗口用左键选中“文件”下的“画面”，则在工程浏览器右边窗口显示“新建”图标和已有的画面文件图标，左键双击“新建”图标或画面文件图标，则进入组态王开发系统。

画面开发系统是应用程序的集成开发环境。工程人员在这个环境中完成界面的设计、动画连接的定义等。画面开发系统具有先进完善的图形生成功能；数据库中有多种数据类型，能合理地抽象控制对象的特性，对数据的报警、趋势曲线、过程记录、安全防范等重要功能有简单的操作办法。利用组态王丰富的图库，用户可以大大减少设计界面的时间，从整体上提高工控软件的质量。

(3) 组态王运行软件 TouchView 是“组态王”软件的实时运行环境，用于显示画面开发系统中建立的动画图形画面，并负责数据库与 I/O 服务程序（数据采集组件）的数据交换。它通过实时数据库管理从一组工业控制对象采集到各种数据，并把数据的变化用动画的方式形象地表示出来，同时完成报警、历史记录、趋势曲线等监视功能，并可生成历史数据文件。

工程浏览器 (TouchExplorer) 和画面运行系统 (TouchView) 是各自独立的 Windows 应用程序，均可单独使用；两者又相互依存，在工程浏览器的画面开发系统中设计开发的画面应用程序必须在画面运行系统 (TouchView) 运行环境中才能运行。

#### 9.4.2 WinCC 监控软件

WinCC 是结合西门子在过程自动化领域中的先进技术和 Microsoft 的强大功能的产物。作为一个先进的人机界面 (HMI) 软件和 SCADA 系统，WinCC 提供了适用于工业的图形显示、消息、归档以及报表的功能模板，并具有高性能的过程耦合、快速的画面更新以及可靠的数据功能。WinCC 还为用户解决方案提供了开放的界面，使得将 WinCC 集成为复杂、广泛的自动化项目成为可能。WINCC 是完善的 HMI/SCADA 软件系统，是高性能的实时信息监控软件平台及企业级的管理信息系统平台。

WinCC 包含编辑和运行两个系统。

WinCC 编辑器包含以下编辑工具：

(1) WinCC 浏览器。WinCC 浏览器管理属于一个项目的全部数据，编辑数据所需要的工具由 WinCC 浏览器自行启动。

(2) 图形编辑器。图形编辑器是一种用于创建过程画面的面向矢量的作图程序。可以用包含在对象和样式选项板中众多的图形对象来创建复杂的过程画面, 可以通过动作编程将动态添加到单个图形对象上, 也可以在库中存储自己的图形对象。

(3) 报警记录。报警记录提供了显示和操作选项来获取和归档结果。可以任意地选择消息块、消息级别、消息类型、消息显示以及报表。

(4) 变量记录。变量记录被用来从运行过程中采集数据并准备将它们显示和归档。可以自由地选择归档、采集和归档定时器的数据格式。可以通过 WinCC 在线趋势和表格控件显示过程值, 并分别在趋势和表格形式下显示。

(5) 报表编辑器。报表编辑器是为消息、操作、归档内容和当前或已归档的数据的定时器或事件控制文档的集成的报表系统, 可以自由选择用户报表或项目文档的形式; 提供舒适的带工具和图形选项板的用户界面, 同时支持各种报表类型; 具有多种标准的系统布局和打印作业。

(6) 全局脚本。全局脚本是 C 语言函数和动作的通称, 根据其不同的类型, 可用于一个给定的项目或众多项目中。脚本被用于给对象组态动作并通过系统内部 C 语言编译器来处理。全局脚本动作用于过程执行的运行中。一个触发可以开始动作的执行。

(7) 用户管理器。用户管理器用于分配和控制用户的单个组态和运行系统编辑器的访问权限。每建立一个用户, 就设置 WinCC 功能的访问权力并独立地分配给此用户, 至多可分配 999 个不同的授权。

### 9.4.3 MCGS 组态软件

MCGS (Monitor and Control Generated System, 通用监控系统) 是一套用于快速构造和生成计算机监控系统的组态软件, 它能够在基于 Microsoft 的各种 32 位 Windows 平台上运行, 通过对现场数据的采集处理, 以动画显示、报警处理、流程控制和报表输出等多种方式向用户提供解决实际工程问题的方案, 在工业控制领域有着广泛的应用。MCGS 组态软件功能强大、操作简单、易学易用, 普通工程人员经过短时间的培训就能迅速掌握多数工程项目的设计和运行操作。同时使用 MCGS 组态软件能够避开复杂的计算机软、硬件问题, 集中精力去解决工程问题本身, 根据工程作业的需要和特点, 组态配置出高性能、高可靠性和高度专业化的工业控制监控系统。

#### 1. MCGS 的主要特性和功能

(1) 实时性与并行处理。MCGS 充分利用了 Windows 操作平台的多任务、按优先级分时操作的功能, 使 PC 机广泛应用于工程测控领域成为可能。工程作业中, 大量的数据和信息需要及时收集、即时处理, 在计算机测控技术领域称其为实时性关键任务, 如数据采集、设备驱动和异常处理等。另外许多工作则是非实时性的, 称为非时间关键任务, 如画面显示, 可在主机运行周期时间内插空进行。

(2) 建立实时数据库, 便于用户分步组态, 保证系统安全可靠运行。MCGS 组态软件由主控窗口、设备窗口、用户窗口、实时数据库和运行策略 5 部分构成。其中的“实时数据库”是整个系统的核心。实时数据库是一个数据处理中心, 是系统各个部分及其各种功能性构件的公用数据区。

(3) 设立“设备工具箱”, 针对外部设备的特征, 用户从中选择某种“构件”, 设置于设备窗口内, 赋予相关的属性, 建立系统与外部设备的连接关系, 即可实现对该种设备

的驱动和控制。不同的设备对应于不同的构件，所有的设备构件均通过实时数据库建立联系，而建立时又是相互独立的，即对某一构件的操作或改动，不影响其他构件和整个系统的结构。

(4) 面向窗口”的设计方法，增加了可视性和可操作性。以窗口为单位，构造用户运行系统的图形界面，使得 MCGS 的组态工作既简单直观，又灵活多变。用户可以使用系统的缺省构架，也可以根据需要自己组态配置，生成各种类型和风格的图形界面，包括 DOS 风格的图形界面、标准 Windows 风格的图形界面以及带有动画效果的工具条和状态条。

(5) 利用丰富的“动画组态”功能，快速构造各种复杂生动的动态画面。以图像、图符、数据、曲线等多种形式，为操作员及时提供系统运行中的状态、品质及异常报警等有关信息。用数据变化大小、颜色改变、明暗闪烁、移动翻转等多种手段，增强画面的动态显示效果。图元、图符对象定义相应的状态属性，即可实现动画效果。同时，MCGS 为用户提供了丰富的动画构件，模拟工程控制与实时监测作业中常用的物理器件的动作和功能。每个动画构件都对应一个特定的动画功能，如实时曲线构件、历史曲线构件、报警显示构件、自由表格构件等。

(6) 引入“运行策略”的概念。复杂的工程作业，运行流程都是多分支的。用传统的编程方法实现，既繁琐又容易出错。MCGS 开辟了“策略窗口”，用户可以选用系统提供的各种条件和功能的“策略构件”，用图形化的方法构造多分支的应用程序，实现自由、精确地控制运行流程，按照设定的条件和顺序，操作外部设备，控制窗口的打开或关闭，与实时数据库进行数据交换。同时，也可以由用户创建新的策略构件，扩展系统的功能。

(7) MCGS 系统由五大功能部件组成，主要的功能部件以构件的形式来构造。不同的构件有着不同的功能，且各自独立。三种基本类型的构件（设备构件、动画构件、策略构件）完成了 MCGS 系统三大部分（设备驱动、动画显示和流程控制）的所有工作。用户也可以根据需要，定制特定类型构件，使 MCGS 系统的功能得到扩充。这种充分利用“面向对象”的技术，大大提高了系统的可维护性和可扩充性。

(8) 支持 OLE Automation 技术。MCGS 允许用户在 Visual Basic 中操作 MCGS 中的对象，提供了一套开放的可扩充接口，用户可根据自己的需要用 VB 编制特定的功能构件来扩充系统的功能。

(9) MCGS 中数据的存储不再使用普通的文件，而是用数据库来管理一切。组态时，系统生成的组态结果是一个数据库；运行时，数据对象、报警信息的存储也是一个数据库。利用数据库来保存数据和处理数据，提高了系统的可靠性和运行效率，同时，也使其他应用软件系统能直接处理数据库中的存盘数据。

(10) 设立“对象元件库”，解决了组态结果的积累和重新利用问题。所谓对象元件库，实际上是分类存储各种组态对象的图库。组态时，可把制作完好的对象（包括图形对象、窗口对象、策略对象及位图文件等）以元件的形式存入图库中，也可把元件库中的各种对象取出，直接为当前的工程所用。随着工作的积累，对象元件库将日益扩大和丰富，组态工作将会变得越来越简单方便。

(11) 提供对网络的支持。考虑到工控系统今后的发展趋势，MCGS 充分运用现今发展

的 DCCW (Distributed Computer Cooperator Work) 技术, 即分布式计算机协同工作方式, 来使分散在不同现场之间的采集系统和 workstation 之间协同工作。通过 MCGS, 不同的 workstation 之间可以实时交换数据, 实现对工控系统的分布式控制和管理。

## 2. MCGS 系统的组态环境和运行环境

用户的所有组态配置过程都在组态环境中进行, 组态环境相当于一套完整的软件工具, 它帮助用户设计和构造自己的应用系统。用户组态生成的结果是一个数据库文件, 称为组态结果数据库。

运行环境是一个独立的运行系统, 它按照组态结果数据库中用户指定的方式进行各种处理, 完成用户组态设计的目标和功能。运行环境本身没有任何意义, 必须与组态结果数据库一起作为一个整体, 才能构成用户应用系统。一旦组态工作完成, 运行环境和组态结果数据库就可以离开组态环境而独立运行在监控计算机上。

## 9.5 组态软件对 PLC 的监控应用举例一

本应用是利用组态王软件在计算机上实现对 S7-200 PLC 的监控。在 PC 机上组态画面, 通过点击画面上的启动和停止按钮实现控制, 把 PLC 的输出信号显示在画面上。

以延时控制为例, 组态王软件监控 PLC 的基本方法如下:

### 9.5.1 编写 PLC 的控制程序

利用编程软件 STEP7 - Micro/WIN V4.0 编写基本的延时控制程序, 下载并让 PLC 运行。

### 9.5.2 建立组态王应用工程

建立组态王应用工程的一般步骤: ①设计图形界面 (定义画面); ②定义设备; ③构造数据库 (定义变量); ④建立动画连接; ⑤运行和调试。

双击桌面图标组态王 6.53, 进入组态王工程管理器, 创建工程路径, 要建立新的组态王工程, 首先为工程指定工作目录 (或称工程路径)。“组态王”用工作目录标识工程, 不同的工程应置于不同的目录。工作目录下的文件由“组态王”自动管理。

启动“组态王”工程管理器 (ProjManager), 选择菜单“文件 \ 新建工程”或单击“新建”按钮。

单击“新建工程”进入新建工程向导, 按向导提示建立延时控制工程项目, 并将该工程项目设为当前工程, 定义的工程信息会出现在工程管理器的信息表格中。

#### 1. 创建组态画面

“组态王”采用面向对象的编程技术, 使用户可以方便地建立画面的图形界面。用户构图时可以像搭积木那样利用系统提供的图形对象完成画面的生成。同时支持画面之间的图形对象复制, 可重复使用以前的开发结果。

双击延时控制工程项目进入工程浏览器, 定义新画面, 选择工程浏览器左侧大纲项“文件 \ 画面”, 在工程浏览器右侧双击“新建”图标。在“画面名称”处输入新的画面名称, 如“监控”等。如图 9-21 所示。

在组态王开发系统中从“工具箱”中分别选择各种图标, 绘制相应的图形对象或文本对象, 如图 9-22 所示。



图 9-21 画面创建

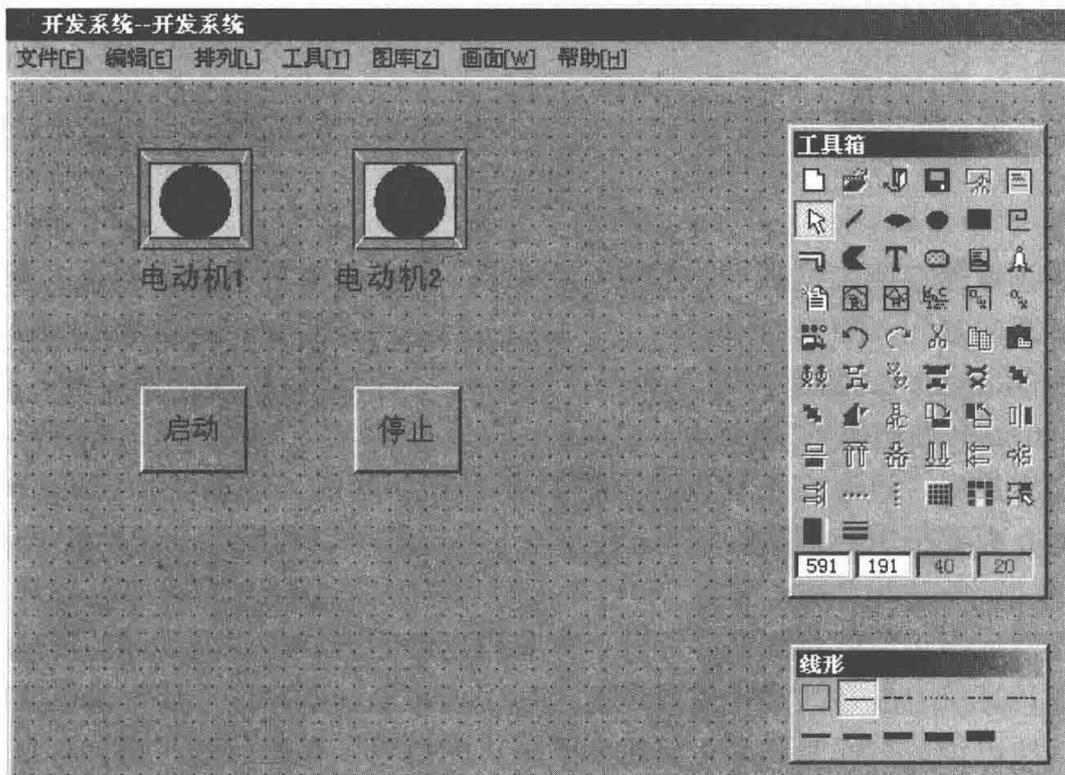


图 9-22 画面组态

## 2. 定义 I/O 设备

组态王把那些需要与之交换数据的设备或程序都作为外部设备。外部设备包括下位机 (PLC、仪表、模块、板卡、变频器等)，它们一般通过串行口和上位机交换数据；其他 Windows 应用程序，它们之间一般通过 DDE 交换数据。外部设备还包括网络上的其他计算

机。只有在定义了外部设备之后，组态王才能通过 I/O 变量和它们交换数据。为方便定义外部设备，组态王设计了“设备配置向导”引导用户一步步完成设备的连接。

选择“S7-200PLC 系列”的“PPI”项，单击“下一步”，弹出“设备配置向导”，为外部设备取一个名称，如“CPU224”，为设备选择连接串口，假设为 COM1，填写设备地址，假设为 2。如图 9-23 所示。



图 9-23 定义 I/O 设备

选择工程浏览器左侧大纲项“设备 \ COM1”，在工程浏览器右侧双击“新建”图标，运行“设备配置向导”。

串口通信的参数：波特率：9600bit/s；数据位：8；奇偶校验：偶校验；停止位：1；通信超时：200ms（最小值）；通信方式：RS-232。

设备定义完成后，可以在工程浏览器的右侧看到新建的外部设备“CPU224”。在定义数据库变量时，只要把 IO 变量连接到这台设备上，它就可以和组态王交换数据了。

### 3. 构造数据库

数据库是“组态王”软件的核心部分，工业现场的生产状况要以动画的形式反映在屏幕上，操作者在计算机上发布的指令也要迅速送达生产现场，所有这一切都是以实时数据库为中介环节的，所以说数据库是联系上位机和下位机的桥梁。在运行时，它含有全部数据变量的当前值。变量在画面制作系统系统中定义，定义时要指定变量名和变量类型，某些类型的变量还需要一些附加信息。数据库中变量的集合形象地称为“数据词典”，数据词典记录了所有用户可使用的数据变量的详细信息。

选择工程浏览器左侧大纲项“数据库/数据词典”，在工程浏览器右侧双击“新建”图标，弹出“变量属性”对话框，此对话框可以对数据变量完成定义、修改等操作，以及数据库的管理工作，例如，在“变量名”处输入变量名，如“M1”；在“变量类型”处选择变量

类型，如 I/O 离散；在“连接设备”中选择先前定义好的 I/O 设备 CPU224；在“寄存器”中定义为 M10.0；在“数据类型”中定义为 BIT 类型。在“读写属性中”中定义为只写。其他属性不用更改，单击“确定”即可。如图 9-24 所示。



图 9-24 定义变量

用相似方法定义其他变量，注意变量类型、寄存器、数据类型的不同。

由于在组态王中，S7-200 系列 PLC 的输入寄存器 I 的读写属性只有只读属性是有效的，无法通过组态王令输入寄存器位变量产生变化来控制 PLC。因此，我们需要对原程序进行修改，在不影响原来现场操作的情况下，能通过组态王软件来实现对 PLC 的远程控制。在原程序的动合触点 I0.0 上并联一个位存储器 M10.0 的动合触点，在动断触点 I0.1 后串联一个位存储器 M10.1 的动断触点，这些位存储器在组态王上用按钮的形式模拟控制信号的输入。

#### 4. 建立动画连接

定义动画连接是指在画面的图形对象与数据库的数据变量之间建立一种关系，当变量的值改变时，在画面上以图形对象的动画效果表示出来；或者由软件使用者通过图形对象改变数据变量的值。“组态王”提供了 21 种动画连接方式：填充属性变化、文本色变化、位置与大小变化、填充、缩放、旋转、水平移动、垂直移动、值输出、模拟值输出、离散值输出、字符串输出等。

双击启动图形对象，可弹出“动画连接”对话框，选命令语言连接，按“按下时”按钮，弹出命令语言对话框，从变量域中选择变量“M1”并设“M1=1”，按确定键返回“动画连接”对话框。再按“弹起时”按钮，弹出命令语言对话框，从变量域中选择变量“M1”并设“M1=0”，按确定键返回。如图 9-25 所示。

用相似方法对其他按钮进行动画连接。对于指示灯，选择了相应的变量之后，只要求确定正常色和报警色即可。



图 9-25 设置命令语言

### 5. 运行和调试

组态王工程初步建立起来后,即进入到运行和调试阶段。在组态王开发系统中选择“文件\切换到 View”菜单命令,进入组态王运行系统。在运行系统中选择“画面\打开”命令,从“打开画面”窗口选择“监控 1”画面。显示出组态王运行系统画面,按下启动按钮, Q0.0 有输出,延时一定的时间后, Q0.1 有输出,再按下停止按钮,输出全部复位,指示灯的信号也复位。如图 9-26 所示。

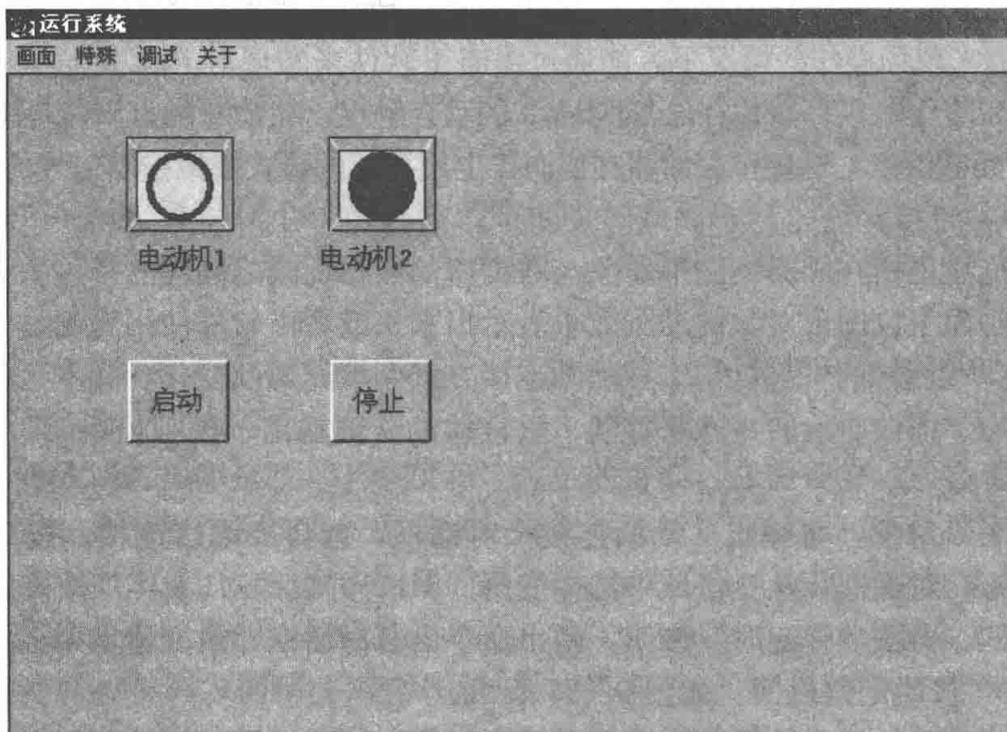


图 9-26 运行系统

## 9.6 组态软件对 PLC 的监控应用举例二

以第8章的水箱温度自动控制为例，PLC选择CPU224和EM235模块，通过EM235模块上的DIP开关确定EM235模块的输入/输出量程范围，本例中选择输入信号的量程为1~5V，通过一个250Ω的电阻将4~20mA电流转换为1~5V的电压信号再送到模拟量输入端，对应的温度为0~100℃。输出端为0~20mA，作为调功器的输入控制信号。

在PC机组态画面上有温度实际（过程）值、输出值的显示。设定值可以在组态画面中设定并显示，定义好组态画面上图形元素与梯形图程序中对应的变量。设备连接的方法、通信端口的设置、组态设计的步骤和应用举例一的方法相同，组态开发的画面如图9-27所示，监控运行的画面如图9-28所示。

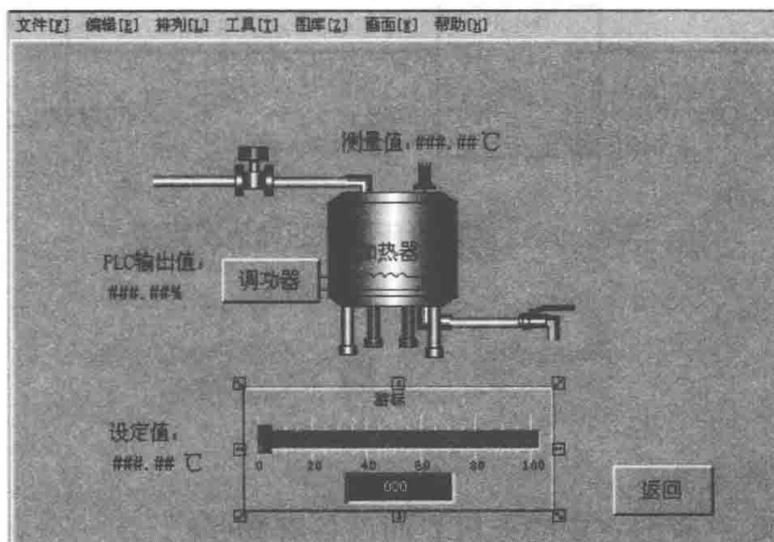


图9-27 组态开发的画面

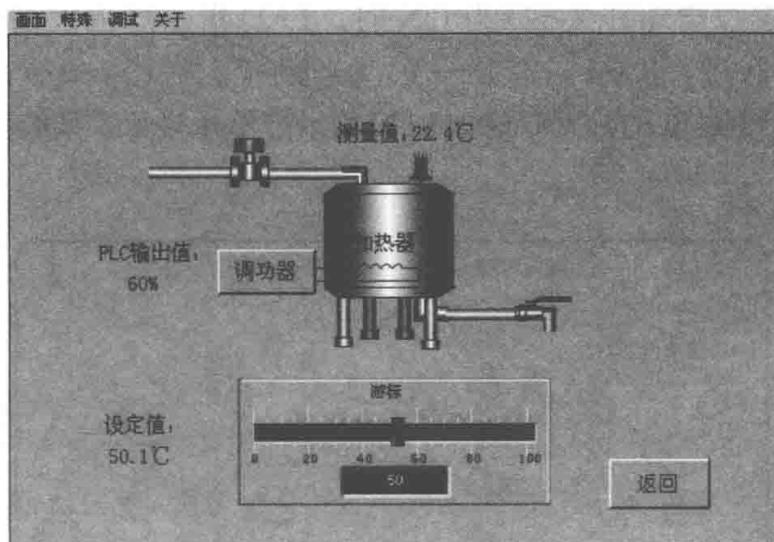


图9-28 监控运行的画面

PLC利用PID向导编程，所编写的温度闭环控制的梯形图程序如图9-29所示。

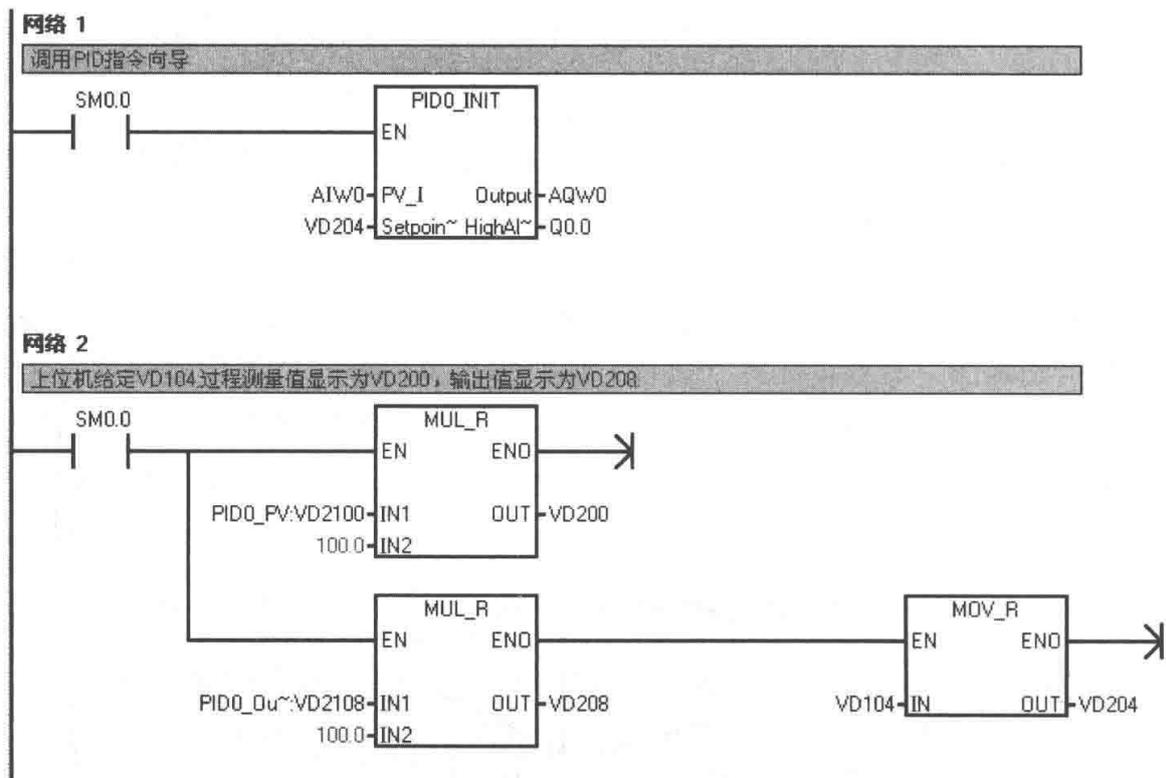


图 9-29 温度闭环控制的梯形图程序

## 习 题

1. 人机界面有哪些功能?
2. 与 S7-200 PLC 对应的主要的人机界面设备有哪些?
3. 人机界面有哪些监控组态软件?
4. 采用 PLC 和触摸屏 (或组态软件) 控制 8 个彩灯。

(1) 系统描述。置位启动开关 SD 为 ON 时, LED 指示灯依次循环显示 1→2→3…→8→1、2→3、4→5、6→7、8→1、2、3→4、5、6→7、8→1→2…，模拟当前喷泉“水流”状态。置位启动开关 SD 为 OFF 时, LED 指示灯停止显示, 系统停止工作。如图 9-30 所示。

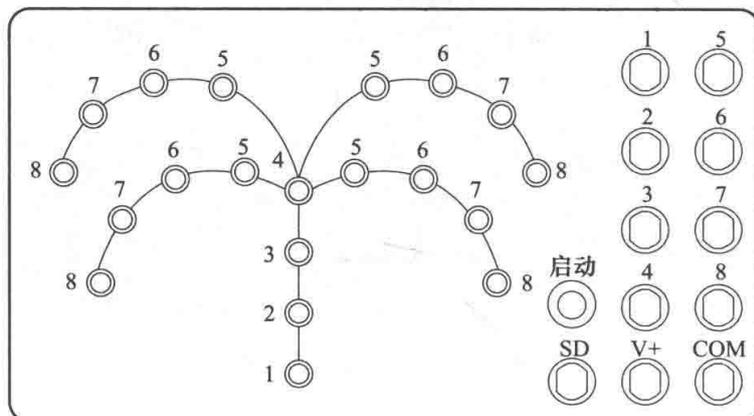


图 9-30 题 4 的图

(2) 控制任务。

- 1) 制作触摸屏画面，在画面中能完成指令的控制，能监控彩灯的工作状况。
- 2) 通过 PLC 外部输入端钮能控制 8 个彩灯。

(3) 任务要求。

- 1) 编写触摸屏（或组态）程序。
- 2) 编写 PLC 程序。
- 3) 正确接线并调试。

## 附录 A S7-200 PLC 的精简指令集

## 布尔指令

LD	N	装载 (开始的动合触点)
LDI	N	立即装载
LDN	N	取反后装载 (开始的动断触点)
LDNI	N	取反后立即装载
A	N	与 (串联的动合触点)
AI	N	立即与
AN	N	取反后与 (串联的动断触点)
ANI	N	取反后立即与
O	N	或 (并联的动合触点)
OI	N	立即或
ON	N	取反后或 (并联的动断触点)
ONI	N	取反后立即与
LDBx	N1, N2	装载字节比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
ABx	N1, N2	与字节比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
OBx	N1, N2	或字节比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
LDWx	N1, N2	装载字比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
AWx	N1, N2	与字比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
OWx	N1, N2	或字比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
LDDx	N1, N2	装载双字比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
ADx	N1, N2	与双字比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
ODx	N1, N2	或双字比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
LDRx	N1, N2	装载实数比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
ARx	N1, N2	与实数比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
ORx	N1, N2	或实数比较结果 N1 (x: <, <=, =, >=, >, <>=) N2
NOT		栈顶值取反
EU		上升沿检测
ED		下降沿检测
=	N	赋值 (线圈)
=I	N	立即赋值
S	S_BIT, N	置位一个区域
R	S_BIT, N	复位一个区域
SI	S_BIT, N	立即置位一个区域
RI	S_BIT, N	立即复位一个区域

续表

传送、移位、循环和填充指令		
MOVB	IN, OUT	字节传送
MOVW	IN, OUT	字传送
MOVD	IN, OUT	双字传送
MOVR	IN, OUT	实数传送
BIR	IN, OUT	立即读取物理输入字节
BIW	IN, OUT	立即写物理输出字节
BMB	IN, OUT, N	字节块传送
BMW	IN, OUT, N	字块传送
BMD	IN, OUT, N	双字块传送
SWAP	IN	交换字节
SHRB	DATA, S_BIT, N	移位寄存器
SRB	OUT, N	字节右移 N 位
SRW	OUT, N	字右移 N 位
SRD	OUT, N	双字右移 N 位
SLB	OUT, N	字节左移 N 位
SLW	OUT, N	字左移 N 位
SLD	OUT, N	双字左移 N 位
RRB	OUT, N	字节右移 N 位
RRW	OUT, N	字右移 N 位
RRD	OUT, N	双字右移 N 位
RLB	OUT, N	字节左移 N 位
RLW	OUT, N	字左移 N 位
RLD	OUT, N	双字左移 N 位
FILL	IN, OUT, N	用指定的元素填充存储器空间
逻辑操作		
ALD		电路块串联
OLD		电路块并联
LPS		入栈
LRD		读栈
LPP		出栈
LDS		装载堆栈
AENO		对 ENO 进行与操作
ANDB	IN1, OUT	字节逻辑与
ANDW	IN1, OUT	字逻辑与
ANDD	IN1, OUT	双字逻辑与
ORB	IN1, OUT	字节逻辑或
ORW	IN1, OUT	字逻辑或
ORD	IN1, OUT	双字逻辑或
XORB	IN1, OUT	字节逻辑异或
XORW	IN1, OUT	字逻辑异或
XORD	IN1, OUT	双字逻辑异或
INVB	OUT	字节取反 (1 的补码)
INWV	OUT	字取反
INVD	OUT	双字取反

续表

## 表、查找和转换指令

ATT	TABLE, DATA	把数据加到表中
LIFO	TABLE, DATA	从表中取数据, 后人先出
FIFO	TABLE, DATA	从表中取数据, 先入先出
FND=	TBL, PATRN, INDX	在表中查找符合比较条件的数据
FND<>	TBL, PATRN, INDX	
FND<	TBL, PATRN, INDX	
FND>	TBL, PATRN, INDX	
BCDI	OUT	BCD 码转换成整数
IBCD	OUT	
BTI	IN, OUT	字节转换成整数
IBT	IN, OUT	整数转换成字节
ITD	IN, OUT	整数转换成双整数
TDI	IN, OUT	双整数转换成整数
DTR	IN, OUT	双整数转换成实数
TRUNC	IN, OUT	实数四舍五入为双整数
ROUND	IN, OUT	实数截位取整为双整数
ATH	IN, OUT, LEN	ASCII 码→16 进制数
HTA	IN, OUT, LEN	16 进制数→ASCII 码
ITA	IN, OUT, FMT	整数→ASCII 码
DTA	IN, OUT, FMT	双整数→ASCII 码
RTA	IN, OUT, FMT	实数→ASCII 码
DECO	IN, OUT	译码
ENCO	IN, OUT	编码
SEG	IN, OUT	7 段译码
中断指令		
CRETI		从中断程序有条件返回
ENI		允许中断
DISI		禁止中断
ATCH	INT, EVENT	给事件分配中断程序
DTCH	EVENT	解除中断事件
通信指令		
XMT	TABLE, PORT	自由端口发送
RCV	TABLE, PORT	自由端口接收
NETR	TABLE, PORT	网络读
NETW	TABLE, PORT	网络写
GPA	ADDR, PORT	获取端口地址
SPA	ADDR, PORT	设置端口地址
高速计数器指令		
HDEF	HSC, MODE	定义高速计数器模式
HSC	N	激活高速计数器
PLS	X	脉冲输出

续表

数学、加 1 减 1 指令		
+I +D +R	IN1, OUT IN1, OUT IN1, OUT	整数, 双整数或实数法 $IN1 + OUT = OUT$
-I -D -R	IN1, OUT IN1, OUT IN1, OUT	整数, 双整数或实数法 $OUT - IN1 = OUT$
MUL *R *I *D	IN1, OUT IN1, OUT IN1, OUT IN1, OUT	整数乘整数得双整数 实数、整数或双整数乘法 $IN1 \times OUT = OUT$
MUL /R /I /D	IN1, OUT IN1, OUT IN1, OUT IN1, OUT	整数除整数得双整数 实数、整数或双整数除法 $OUT / IN1 = OUT$
SQRT	IN, OUT	平方根
LN	IN, OUT	自然对数
LXP	IN, OUT	自然指数
SIN	IN, OUT	正弦
COS	IN, OUT	余弦
TAN	IN, OUT	正切
INCB INCW INCD	OUT OUT OUT	字节加 1 字加 1 双字加 1
DECB DECW DECD	OUT OUT OUT	字节减 1 字减 1 双字减 1
PID	Table, Loop	PID 回路
定时器和计数器指令		
TON TOF TONR	Txxx, PT Txxx, PT Txxx, PT	通电延时定时器 断电延时定时器 保持型通延时定时器
CTU CTD CTUD	Txxx, PV Txxx, PV Txxx, PV	加计数器 减计数器 加/减计数器
实时时钟指令		
TODR TODW	T T	读实时时钟 写实时时钟
程序控制指令		
END		程序的条件结束
STOP		切换到 STOP 模式
WDR		“看门狗”复位 (300ms)
JMP LBL	N N	跳到指定的标号 定义一个跳转的标号

续表

程序控制指令	
CALL    N (N1, ...) CRET	调用子程序, 可以有 16 个可选参数 从子程序条件返回
FOR    INDX, INIT, FINAL NEXT	For/Next 循环
LSCR    N SCRT    N SCRE	顺控继电器段的启动 顺控继电器段的转换 顺控断电器段的结束

## 附录 B S7-200 PLC 的 CPU 模块的技术性能指标

性能指标 \ CPU 类型	CPU221	CPU222	CPU224	CPU224XP	CPU226
外形尺寸 (mm×mm×mm)	90×80×62	90×80×62	120.5×80×62	140×80×62	196×80×62
用户程序	4096	4096	8192	12 288	16 384
用户数据	2048	2048	8192	10 240	10 240
掉电保持时间 (h)	50	50	100	100	100
本机数字量 I/O	6/4	8/6	14/10	14/10	24/16
本机模拟量 I/O	无	无	无	2/1	无
扩展模块数量	0	2	7	7	7
数字量 I/O 映像区	128 入/128 出	128 入/128 出	128 入/128 出	128 入/128 出	128 入/128 出
模拟量 I/O 映像区	无	16 入/16 出	32 入/32 出	32 入/32 出	32 入/32 出
脉冲捕捉输入	6	8	14	14	24
脉冲输出	2	2	2	2	2
辅助继电器 (M)	256	256	256	256	256
定时器/计数器	256/256	256/256	256/256	256/256	256/256
状态寄存器 (S)	256	256	256	256	256
高速计数器	4	4	6	6	6
定时中断	2 (1~255ms)	2 (1~255ms)	2 (1~255ms)	2 (1~255ms)	2 (1~255ms)
边沿中断	4 个上升沿或 4 个下降沿				
模拟电位器	1 (8bit)	1 (8bit)	2 (8bit)	2 (8bit)	2 (8bit)
布尔指令执行 速度 (μs/指令)	0.22	0.22	0.22	0.22	0.22
口令保护	有	有	有	有	有
通信口	1	1	1	2	2
通信协议	PPI、DP、自由口				
最多主站数	32	32	32	32	32

## 附录 C S7-200 SMART PLC 的 CPU 模块的技术性能指标

CPU 类型 性能指标	经济型			标准型		
	20C	40C	60C	20S	40S	60S
尺寸 (mm×mm×mm)	90×100×81	125×100×81	175×100×81	90×100×81	125×100×81	175×100×81
指令速度						
布尔指令计算 (ns)	150	150	150	150	150	150
整数指令计算 (μs)	1.2	1.2	1.2	1.2	1.2	1.2
浮点数指令计算 (μs)	3.6	3.6	3.6	3.6	3.6	3.6
存储空间						
程序存储空间 (Byte)		12K		12K	24K	30K
数据存储空间 (Byte)		8K		8K	16K	20K
保持存储空间 (Byte)	—			10K, V 区、M 区、T 和 C 保持存储空间		
M 存储区 (Byte)	32					
断电保持时间	—			7 天		
储存卡	可选 (程序传输、固件升级、恢复出厂设置)					
本地 I/O						
本地集成 I/O 数量	12/8	24/16	36/24	12/8	24/16	36/24
I/O 过程映像区	256/256	256/256	256/256	256/256	256/256	256/256
信号板数量	—			1		
扩展模版数量	—			4		
数字量最大点数	20	40	60	76/72	88/80	100/88
模拟量最大点数	无			16/8+1		
基本特性						
HSC	4 个 30kHz, 单相			4 个 60kHz, 单相		
	2 个 20kHz, A/B 相			2 个 40kHz, A/B 相		
RTO	无			2 个 100kHz	3 个 100kHz	
定时中断	1ms					
沿中断	4 上升沿/4 下降沿					
脉冲捕捉数量		14		12+2	14+2	14+2
定时器/计数器	256 (1ms×4/10ms×16/100ms×236)/256					
上升沿个数	1024					
PID 回路	无			8		

## 参 考 文 献

- [1] 廖常初. FX 系列 PLC 编程及应用. 北京: 机械工业出版社, 2008.
- [2] 李仁. 电器控制. 北京: 机械工业出版社, 2006.
- [3] 刘华波. S7 - 200 PLC 编程及应用案例精选. 北京: 机械工业出版社, 2009.
- [4] 蔡行健, 黄文钰. 深入浅出西门子 S7 - 200 PLC. 北京: 北京航空航天大学出版社, 2003.
- [5] SIEMENS AG. S7 - 200CN 可编程控制器产品样本, 2013.
- [6] SIEMENS AG. S7 - 200SMART 系统手册, 2013.
- [7] 廖常初. S7 - 200 PLC 编程及应用. 2 版. 北京: 机械工业出版社, 2014.
- [8] 任振辉, 邵利敏. 现代电气控制技术. 北京: 机械工业出版社, 2012.

Images have been losslessly embedded. Information about the original file can be found in PDF attachments. Some stats (more in the PDF attachments):

```
{
  "filename": "MTM4MzlyNDcuemlw",
  "filename_decoded": "13832247.zip",
  "filesize": 41113393,
  "md5": "6ec04f8cdeab6707baa29ffeca1cc078",
  "header_md5": "e2d5397368a276c5e69af16241a6aafe",
  "sha1": "35b8a195552d2a07a3eee5b707e6ceae3a39819f",
  "sha256": "442df110f43a79efa9fe6584176f47e33bf87da0f81e9903068f801bb249a8ac",
  "crc32": 2055164713,
  "zip_password": "28zrs",
  "uncompressed_size": 50527420,
  "pdg_dir_name": "PLC\u539f\u7406\u53ca\u5de5\u7a0b\u5e94\u7528 \u7b2c2\u7248_13832247",
  "pdg_main_pages_found": 221,
  "pdg_main_pages_max": 221,
  "total_pages": 228,
  "total_pixels": 1340938192,
  "pdf_generation_missing_pages": false
}
```