

可编程控制器 (PLC) 原理 及应用 实验指导书

雷旭 杨霞 编

长安大学电控学院

自动化与交通控制工程实验中心

2009.6

目录

第一章 系统简介	3
一、可编程序控制器（PC）主机	3
二、编程装置	3
三、输入输出部分	3
四、输入/输出接口的使用方法	5
第二章 Micro/Win V3.2 的安装与使用	7
一、软件的安装	7
二、软件的使用	8
第三章 S7-200 内部资源介绍	15
第四章 PLC 控制实验	19
实验一 基本指令实验	19
实验二 置位/复位指令实验	24
实验三 跳转指令实验	26
实验四 定时器及计数器指令实验	28
实验五 移位寄存器指令实验	36
实验六 常用功能指令实验	40
实验七 舞台灯的 PLC 控制	45
实验八 LED 数码管显示控制	47
实验九 交通信号灯的自动控制	49
实验十 驱动步进电机的 PLC 控制	51
实验十一 电机的星/三角启动控制	53
实验十二 机械手的 PLC 自动控制	55
实验十三 四层电梯的 PLC 控制	57
实验十四 刀库捷径方向选择控制	60
实验十五 物料混合控制实验	62
实验十六 水塔水位控制	64
实验十七 邮件分拣控制	66
实验十八 四级传送带的控制	68

第一章 系统简介

西门子（SIMATIC）S7-200 系列小型 PLC（Micro PLC）可应用于各种自动化系统。紧凑的结构、低廉的成本以及功能强大的指令使得 S7-200 PLC 成为各种小型控制任务的理想的解决方案。S7-200 产品的多样化以及基于 Windows 的编程工具，使您能够更加灵活地完成自动化任务。

S7-200 功能强，体积小，使用交流电源可在 85~265V 范围内变动，且机内还设有供输入用的 DC-24V 电源。可编程序控制器（简称 PC）在进行生产控制或实验时，都要求将用户程序的编码表送入 PC 的程序存贮器，运行时 PC 根据检测到的输入信号和程序进行运算判断，然后通过输出电路去控制对象。所以典型的 PC 系统由以下三部分组成：输入/输出接口、PC 主机、通讯口。

S7-200 CPU 将微处理器、集成电源、输入电路和输出电路集成在一个紧凑的外壳中，从而形成了一个功能强大的 Micro PLC。参见图 1--1。

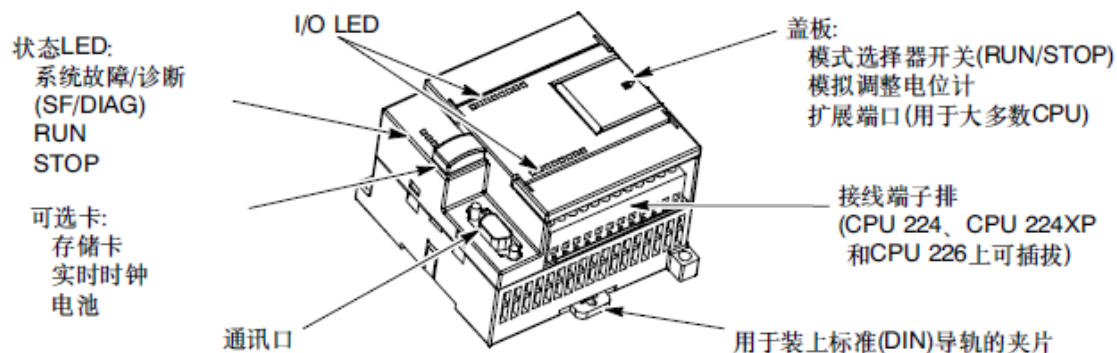


图 1-1 S7-200 Micro PLC

一、可编程序控制器（PC）主机

在我们的实验箱中，选用的 PC 主机是 SIMATIC S7-200 CPU224，有 14 个输入点，10 个输出点，可采用助记符和梯形图两种编程方式。

二、编程装置

通常采用微型计算机作为编程装置。安装 SIEMENS 公司的 PLC 编译调试软件 STEP 7 MicroWIN V4.0，用专用的编程电缆将电脑的 USB 口和 PLC 主机的编程通讯接口连接起来，运行 MicroWIN V4.0，即可将 PLC 程序的编码表下载至 PLC 的存储器中，运行程序，即可进行各种控制实验。关于上位软件的安装与使用请参见第二章。

三、输入输出部分

实验箱下面板布置示意图如图 1-2 所示

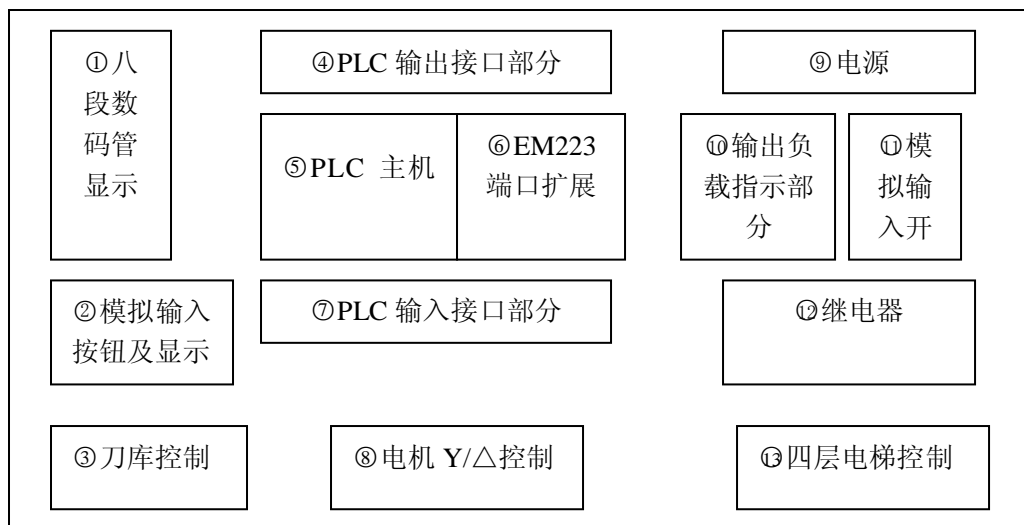


图 1-2 实验箱下面板结构示意图

在图 1-2 中：

①为八段数码管显示接口部分。可显示数字和部分字母。

②模拟输入按钮。共 4 个按钮，S1 ~ S6，COM 为公共端。按钮属点动型开关。

③刀库控制部分

④PLC 输出接口部分。共 14 点，分为四组

表 1-1 是输出控制端 1L、2L、3L、21L 与输出端的对应关系，第四组为扩展的输出端。

表 1-1

	输出端	控制端
一组	Q0.0、Q0.1、Q0.2、Q0.3	1L
二组	Q0.4、Q0.5、Q0.6	2L
三组	Q0.7、Q1.0、Q1.1	3L
四组	Q2.0、Q2.1、Q2.2、Q2.3	21L

其中 1L、2L、3L、21L 分别为各组的公共端。用于控制输出电平的有效电平，当 L 端接高电平，则当输出端有效时，输出端为高电平。当 L 端接低电平，则当输出端有效时，输出端为低电平。输出方式为继电器触点输出。

⑤PLC 主机部分。

⑥EM223 端口扩展部分，一共扩展输入端口 4 个，输出端口 4 个。

⑦PLC 输入接口部分。共 18 点，分为三组

表 1-2 是输出控制端 1M、2M、21M 与输出端的对应关系。

表 1-2

	输入端	控制端
一组	I0.0 ~ I0.7	1M
二组	I1.1 ~ I1.5	2M
三组	I2.0 ~ I2.3	21M

其中 1M、2M、21M 分别为各组的公共端。用于控制输入电平的有效电平，当 M 端

接高电平，则输入端有效电平为高电平。当 M 端接低电平，则输入端有效电平为低电平。

⑧电机控制

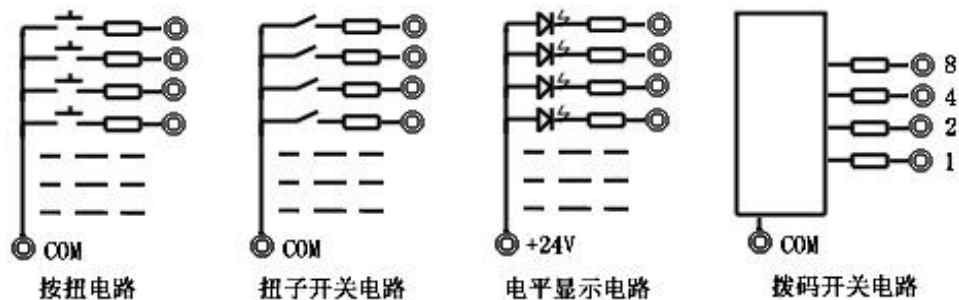
⑨直流稳压电源部分。输出电压为 24V，最大负载电流为 3A，带输出指示灯，电源开关。

⑩输出负载指示部分。含有 8 个发光二极管，COM+为二极管公共端。发光二极管为共阳接法。

⑪模拟输入开关。共 8 个钮子开关，K1 ~ K8，COM 为公共端。钮子开关属长动型开关。

⑫继电器负载，上触电常开，下触电常闭。

⑬电梯控制

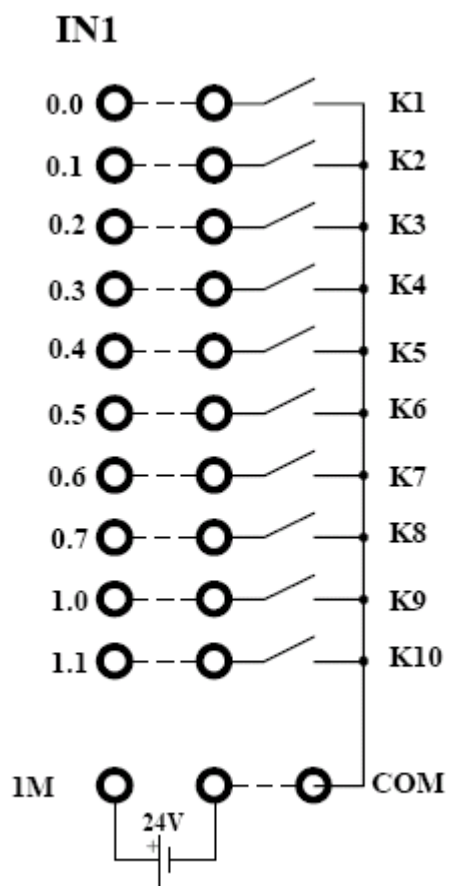


我们已经将 PLC 主机上的输入/输出端和实验板上输入输出端对应接好。实验时只须用双头线按实验电路图搭接即可。实验箱上的输入模拟开关、输出负载指示、拨码开关、数码管、及继电器主要用于基本指令的学习。综合实验可通过实验演示屏进行，这样比较直观，易于理解。

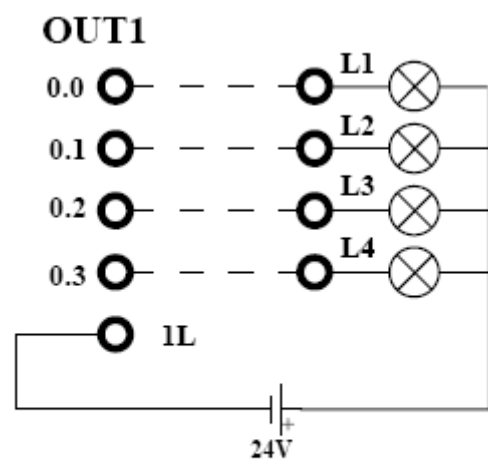
四、输入/输出接口的使用方法

1、输入接口：将输入接口的相应端口，根据需要与钮子开关或按钮用双头线相连。输入接口的控制端 1M 或 2M 接 24V，钮子开关或按钮的公共端接 0V。这样，当开关闭合或按下按钮时，相应端口的输入指示灯就会点亮，表示有输入到 PLC。

2、输出接口：将输出接口的相应端口，根据需要接发光二极管，输出接口的控制端 1L 或 2L 或 3L 接 0V，发光二极管的公共端接 24V。这样当 PLC 的相应的输出端口有输出时，所接的发光二极管点亮。



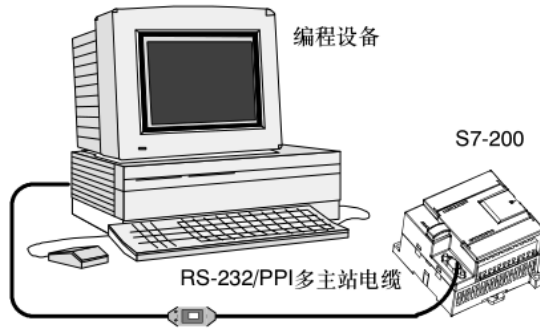
输入开关接法



输出负载接法

第二章 Micro/Win V3.2 的安装与使用

SIEMENS S7-200 系列可编程序控制器的编译调试软件为 **Micro/Win V3.2**。下面介绍该软件的安装与使用。

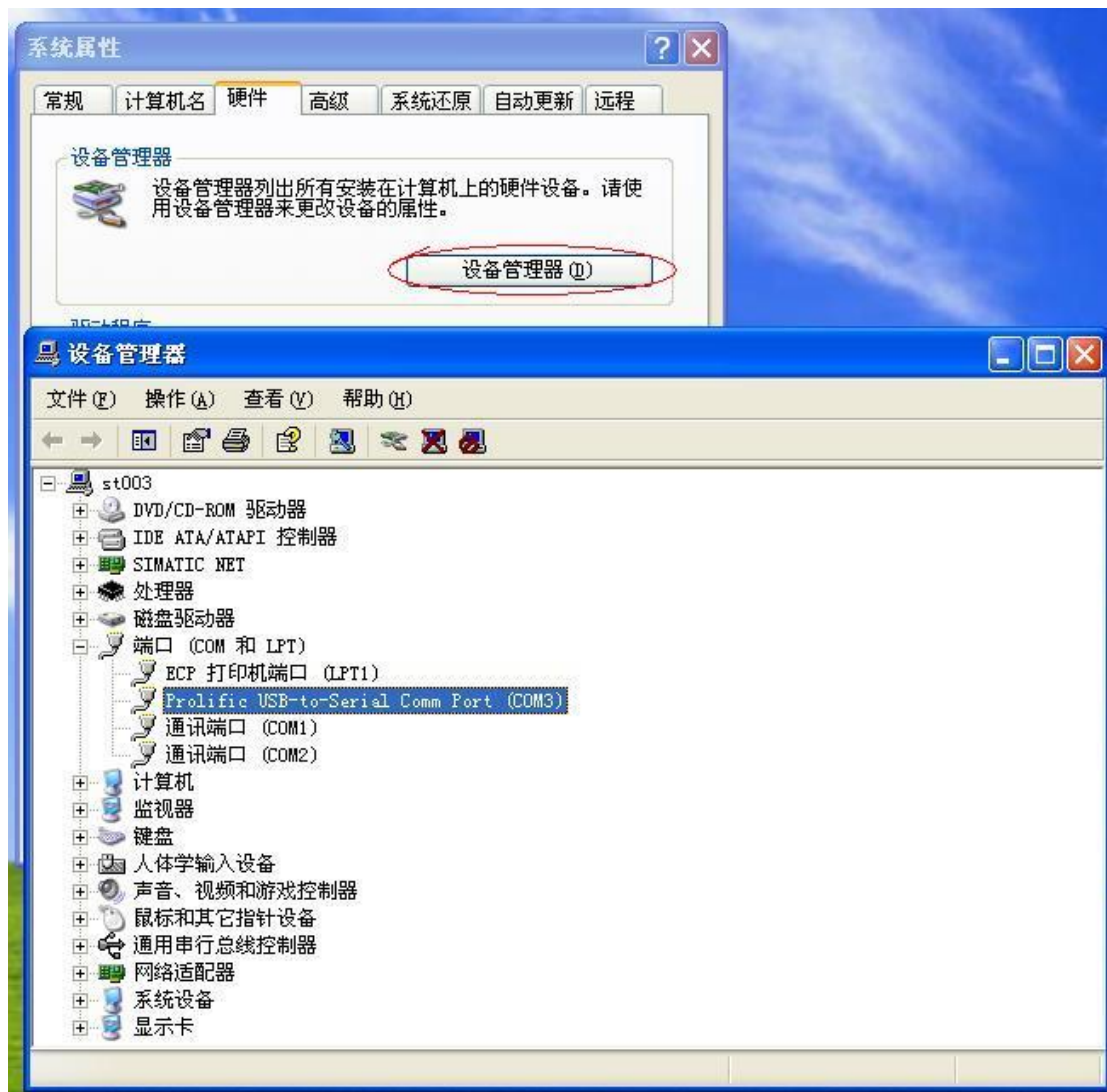


一、软件的安装

- 1、控制器和计算机之间通过 RS-232/PPI 多主站电缆连接，所以首先要安装 PC/PPI cable 通信连接软件。安装好后可在电脑中查看为控制器配备的通讯端口，方法如下：



- 2、查看端口设置是否成功，并记住端口号 COM?。（在编译软件 step7 中通信设置中使用。）

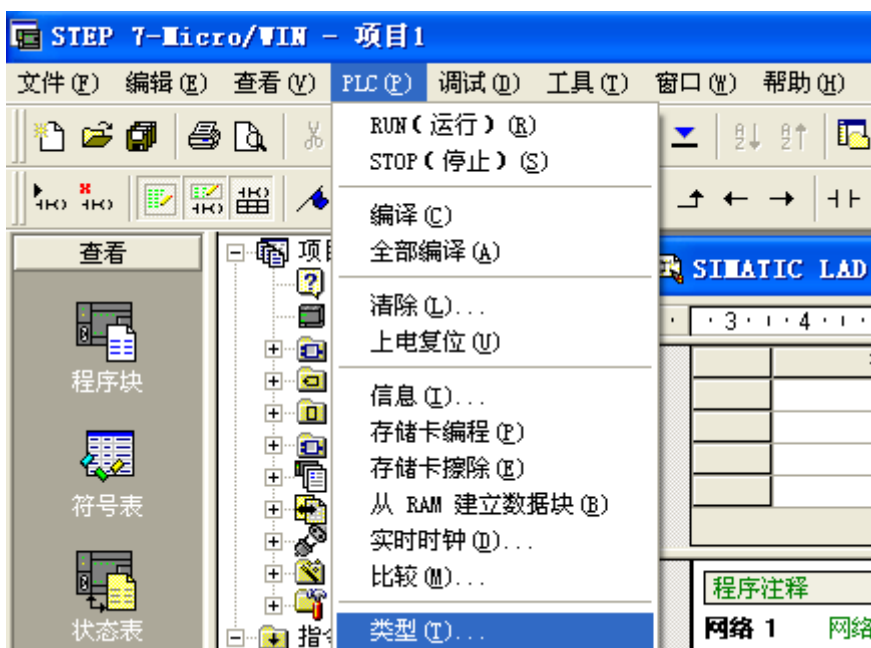


3、在我们提供的光盘中的 MicroWINV32 目录下找到 Setup.exe 文件，运行 Setup.exe，开始安装。安装向导将引导你完成整个安装过程。

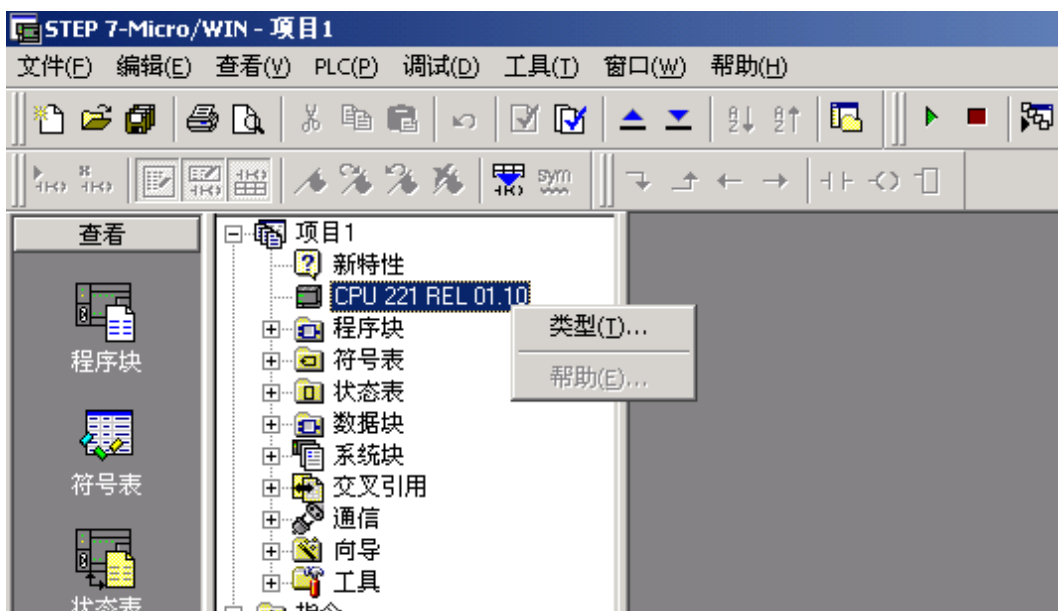
4、安装完成后，在桌面上会创建一个快捷方式。

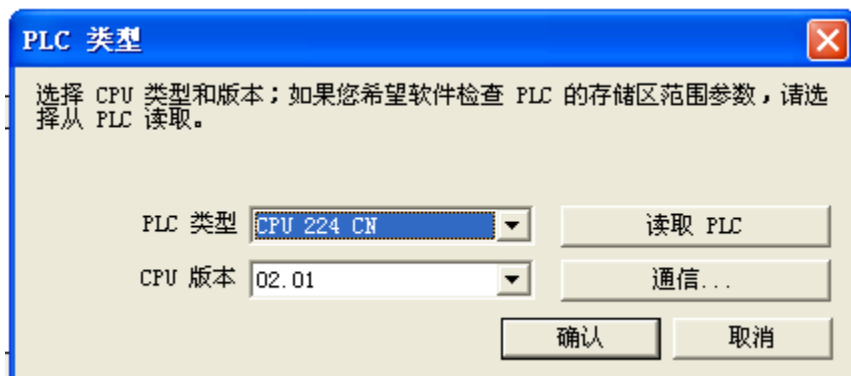
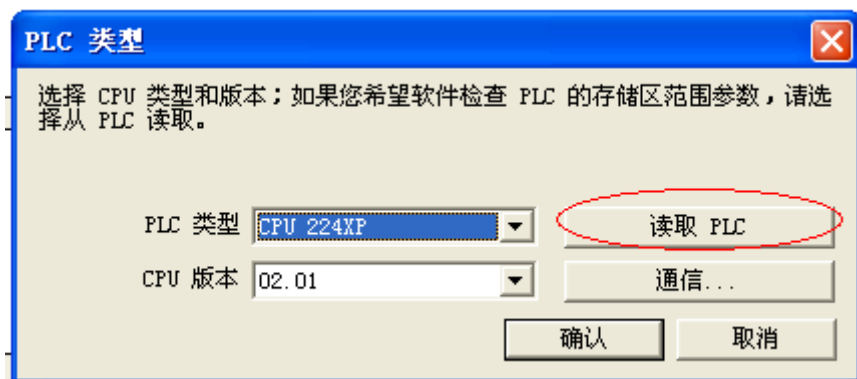
二、软件的使用

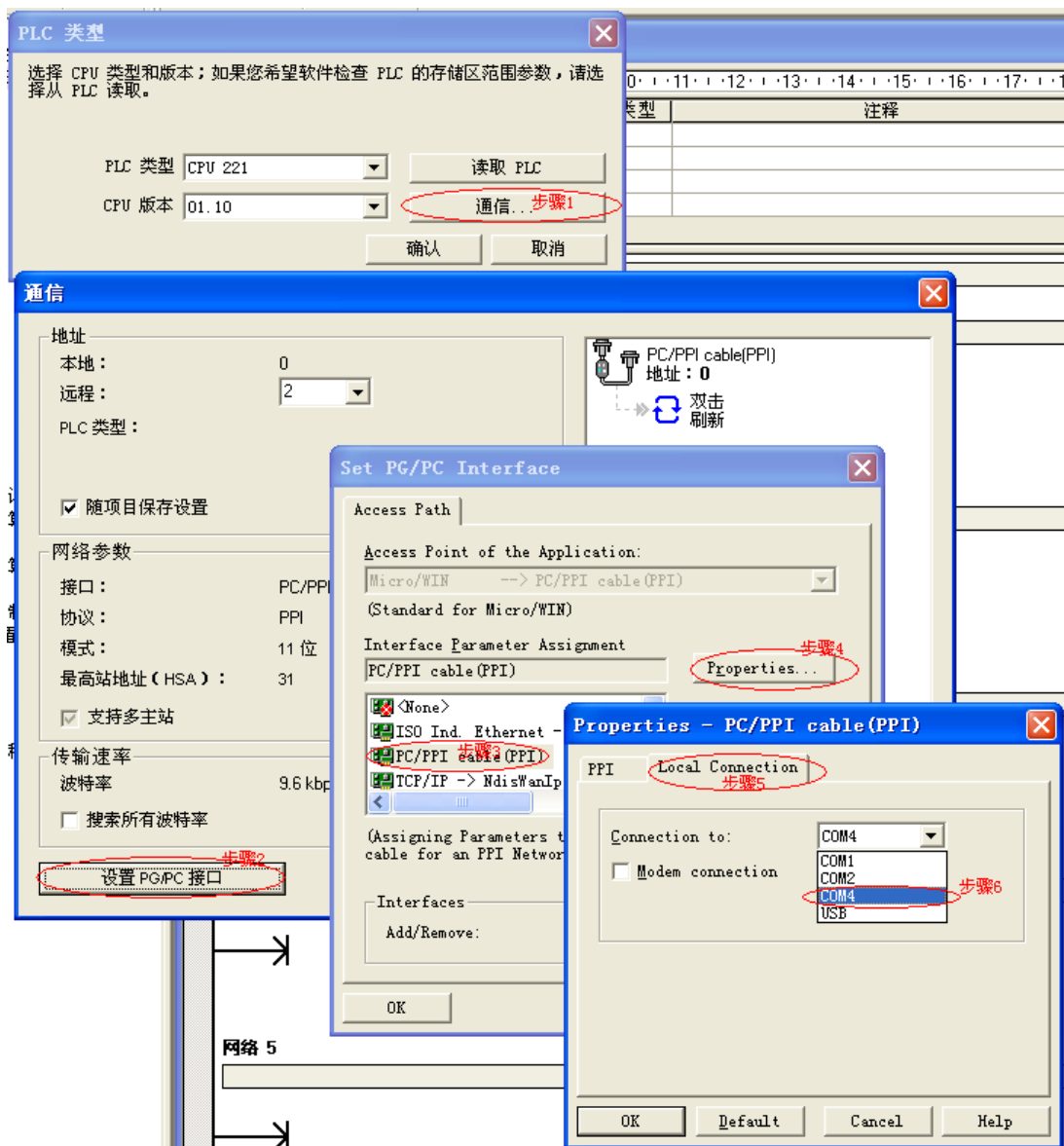
- 1、双击桌面上的快捷方式，启动程序。
- 2、先用编程电缆将电脑的 USB 口和 PLC 主机的通讯口 1 连接起来，并打开实验箱电源。
注意跟换 USB 口后，要更改连接通讯口。
- 3、软件通讯连接。



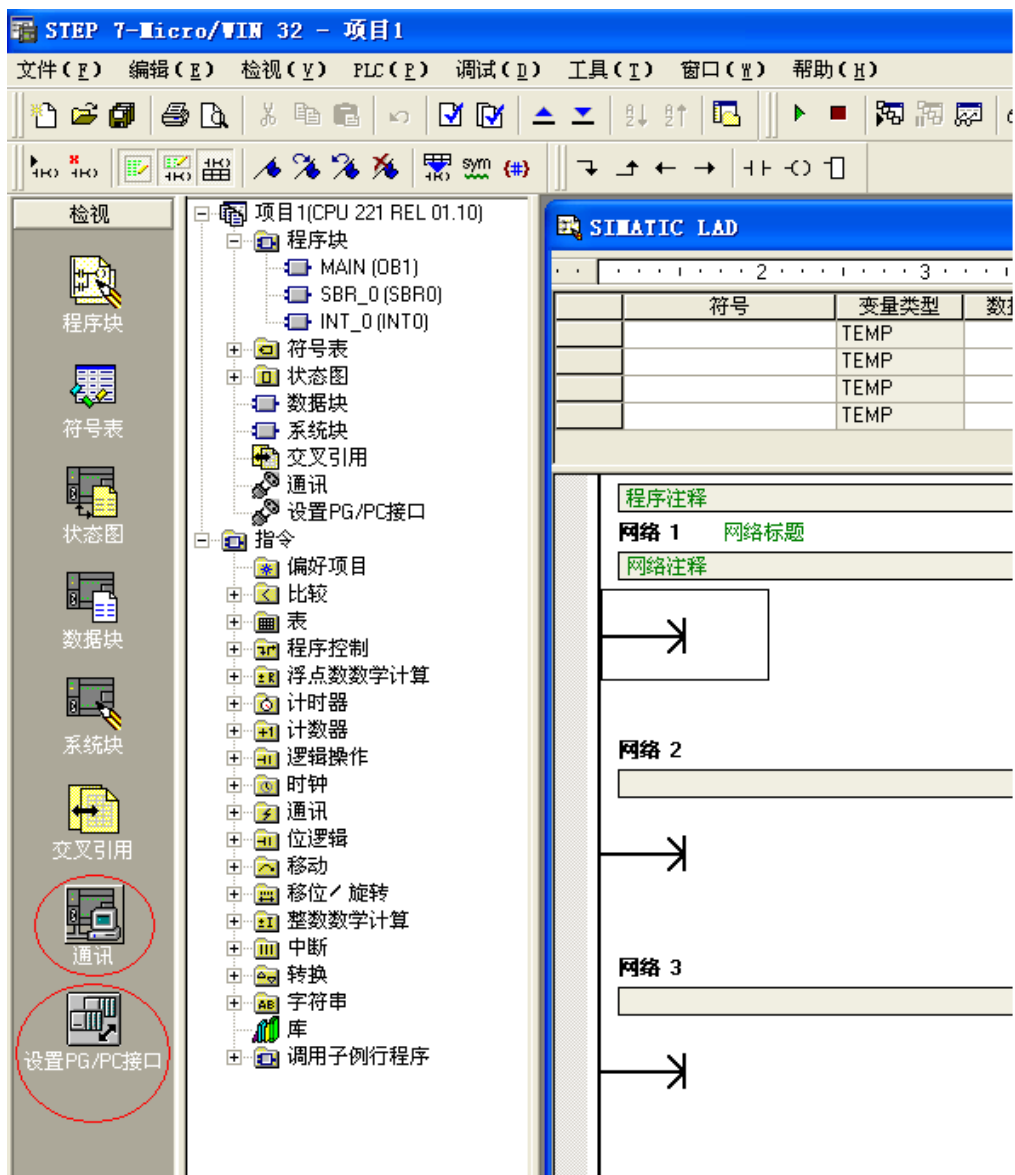
或是



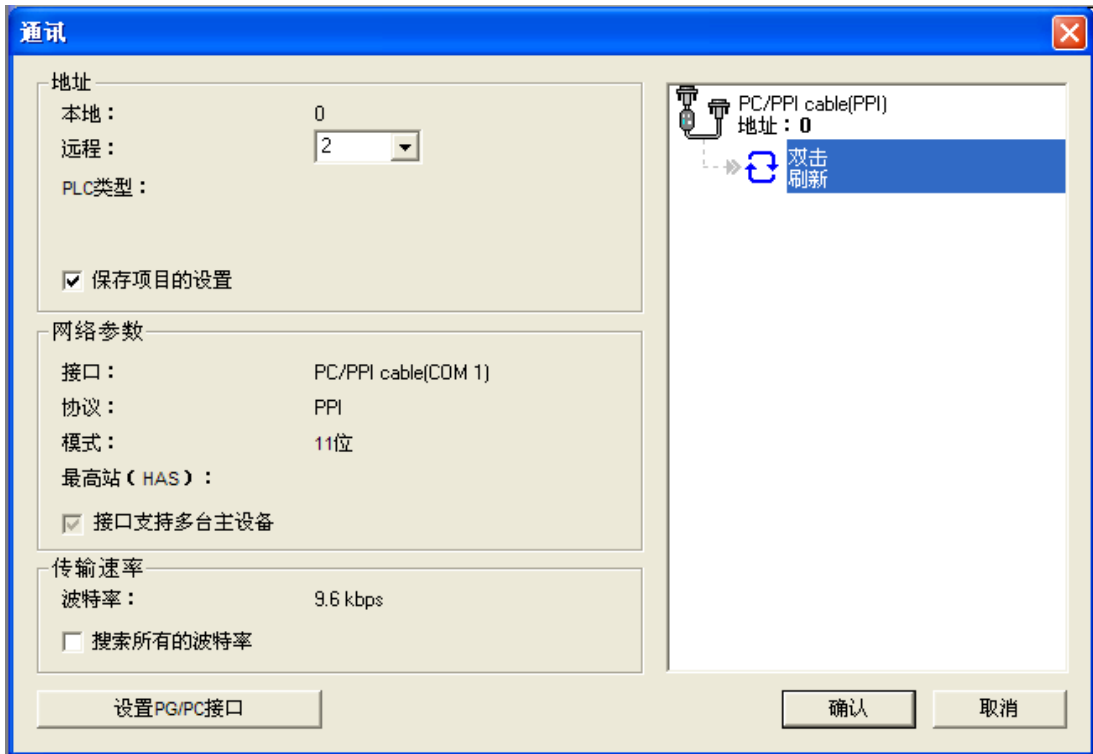




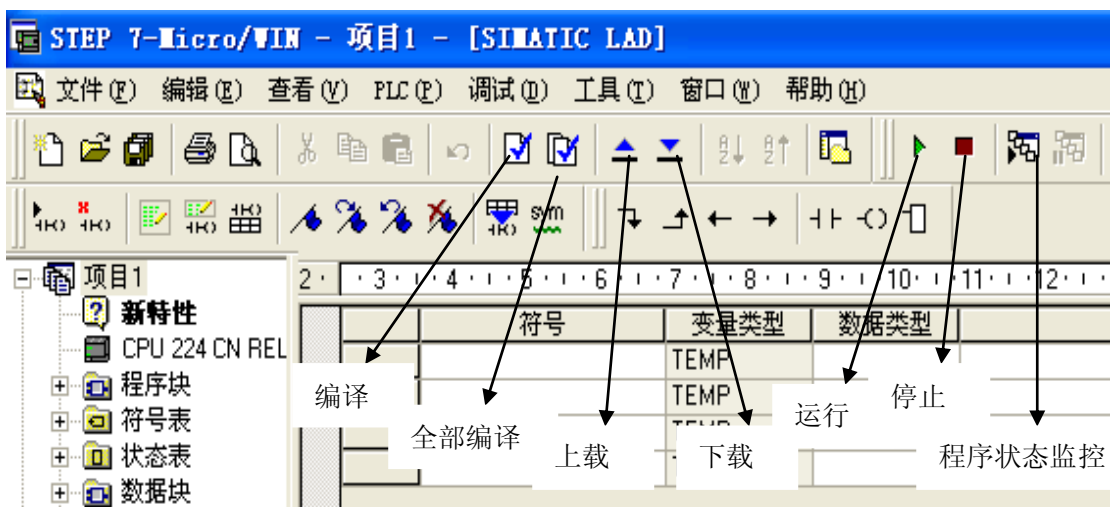
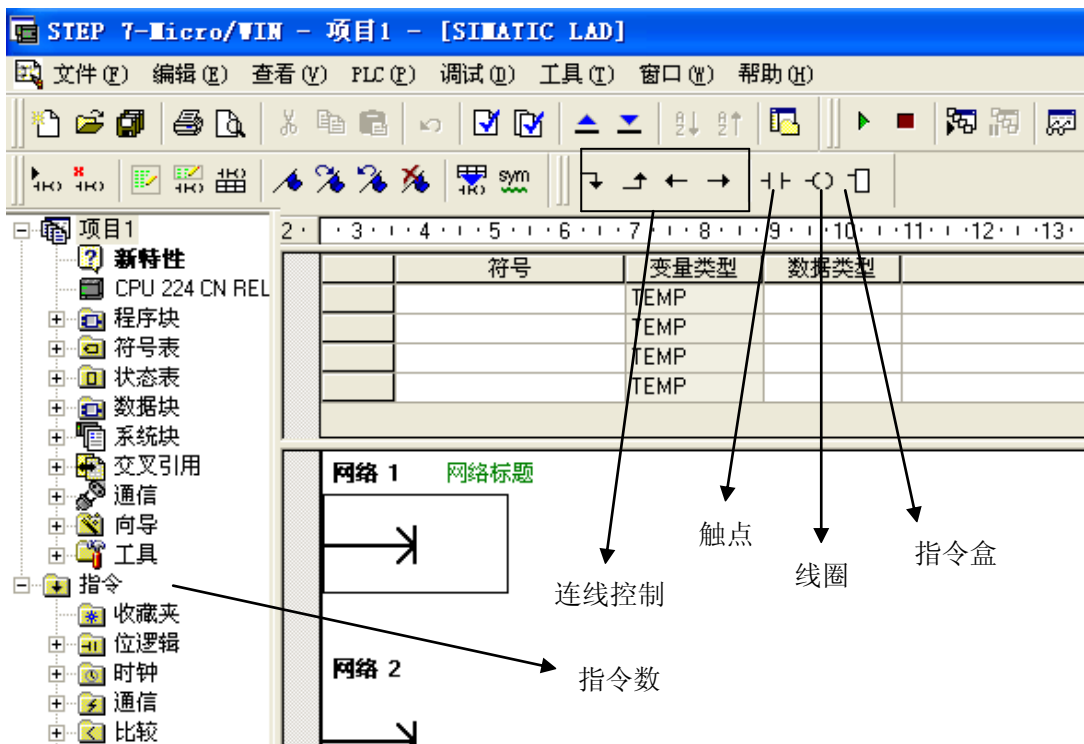
或是



通讯属性设置完成以后，开始连接计算机和 PLC，如下图所示。不断的双击刷新，直到显示正确的 CPU 及其类型，确认连接。



- 4、从文件菜单中点击新建命令，输入程序，并保存为“*.MWP”的项目文件。
- 5、在< PLC > 菜单下选择 < 编译 >，检查程序语法正确与否，直至没有错误。
- 6、点击< 文件 >菜单下 < 下载 >。
- 7、再选择< PLC >菜单下< 运行 >命令，就可以运行你自己的程序了。
- 8、在运行模式中，你可以选择< 调试 >菜单下< 程序状态 >，来查看各输入输出端口、内部触点的运行状态，以确定程序设计是否正确，提高你的调试效率。
- 9、软件使用图示：



以上是该软件的快速使用指南。如想深入了解该软件的其他功能，请参阅帮助菜单下的帮助文档，那里有更详细的说明。

第三章 S7-200 内部资源介绍

S7-200存储器范围及特性

表6-1 S7-200CPU存储器范围及特性

描述	CPU221	CPU222	CPU224	CPU 224XP CPU 224XPsi	CPU226
用户程序大小 在运行模式下编辑 不在运行模式下编辑	4096字节 4096字节	4096字节 4096字节	8192字节 12288字节	12288字节 16384字节	16384字节 24576字节
用户数据大小	2048字节	2048字节	8192字节	10240字节	10240字节
输入映像寄存器	I0.0 - I15.7	I0.0 - I15.7	I0.0 - I15.7	I0.0 - I15.7	I0.0 - I15.7
输出映像寄存器	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(只读)	AIW0 - AIW30	AIW0 - AIW30	AIW0 - AIW62	AIW0 - AIW62	AIW0 - AIW62
模拟量输出(只写)	AQW0 - AQW30	AQW0 - AQW30	AQW0 - AQW62	AQW0 - AQW62	AQW0 - AQW62
变量存储器(V)	VB0 - VB2047	VB0 - VB2047	VB0 - VB8191	VB0 - VB10239	VB0 - VB10239
局部存储器(L) ¹	LB0 - LB63	LB0 - LB63	LB0 - LB63	LB0 - LB63	LB0 - LB63
位存储器(M)	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7
特殊存储器(SM) 只读	SM0.0 - SM179.7 SM0.0 - SM29.7	SM0.0 - SM299.7 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	256 (T0 - T255)	256 (T0 - T255)	256 (T0 - T255)	256 (T0 - T255)	256 (T0 - T255)
保持接通延时	1 ms 10 ms 100 ms	T0, T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95	T0, T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95	T0, T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95	T0, T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95
开/关延时	1 ms 10 ms 100 ms	T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255	T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255	T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255	T32, T96 T33 - T36, T97 - T100 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
顺序控制继电器(S)	S0.0 - S31.7	S0.0 - S31.7	S0.0 - S31.7	S0.0 - S31.7	S0.0 - S31.7
累加器寄存器	AC0 - AC3	AC0 - AC3	AC0 - AC3	AC0 - AC3	AC0 - AC3
跳转/标号	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
调用/子程序	0 - 63	0 - 63	0 - 63	0 - 63	0 - 127
中断程序	0 - 127	0 - 127	0 - 127	0 - 127	0 - 127
正/负跳变	256	256	256	256	256
PID回路	0 - 7	0 - 7	0 - 7	0 - 7	0 - 7
端口	端口0	端口0	端口0	端口0、端口1	端口0、端口1

¹ LB60-LB63为STEP 7-Micro/WIN32的3.0版本或以后的版本软件保留。

表6-2 S7-200 CPU的操作数范围

存取方式		CPU221	CPU222	CPU224	CPU 224XP CPU 224XPsi	CPU226	
位存取(字节.位)		0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	
	Q	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	
	V	0.0 - 2047.7	0.0 - 2047.7	0.0 - 8191.7	0.0 - 10239.7	0.0 - 10239.7	
	M	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	
	SM	0.0 - 165.7	0.0 - 299.7	0.0 - 549.7	0.0 - 549.7	0.0 - 549.7	
	S	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	
	T	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255	
	C	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255	
	L	0.0 - 63.7	0.0 - 63.7	0.0 - 63.7	0.0 - 63.7	0.0 - 63.7	
字节存取	IB	0 - 15	0 - 15	0 - 15	0 - 15	0 - 15	
	QB	0 - 15	0 - 15	0 - 15	0 - 15	0 - 15	
	VB	0 - 2047	0 - 2047	0 - 8191	0 - 10239	0 - 10239	
	MB	0 - 31	0 - 31	0 - 31	0 - 31	0 - 31	
	SMB	0 - 165	0 - 299	0 - 549	0 - 549	0 - 549	
	SB	0 - 31	0 - 31	0 - 31	0 - 31	0 - 31	
	LB	0 - 63	0 - 63	0 - 63	0 - 63	0 - 63	
	AC	0 - 3	0 - 3	0 - 3	0 - 255	0 - 255	
	KB(常数)	KB(常数)	KB(常数)	KB(常数)	KB(常数)	KB(常数)	
字存取	IW	0 - 14	0 - 14	0 - 14	0 - 14	0 - 14	
	QW	0 - 14	0 - 14	0 - 14	0 - 14	0 - 14	
	VW	0 - 2046	0 - 2046	0 - 8190	0 - 10238	0 - 10238	
	MW	0 - 30	0 - 30	0 - 30	0 - 30	0 - 30	
	SMW	0 - 164	0 - 298	0 - 548	0 - 548	0 - 548	
	SW	0 - 30	0 - 30	0 - 30	0 - 30	0 - 30	
	T	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255	
	C	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255	
	LW	0 - 62	0 - 62	0 - 62	0 - 62	0 - 62	
	AC	0 - 3	0 - 3	0 - 3	0 - 3	0 - 3	
	AIW	0 - 30	0 - 30	0 - 62	0 - 62	0 - 62	
	AQW	0 - 30	0 - 30	0 - 62	0 - 62	0 - 62	
	KB(常数)	KB(常数)	KB(常数)	KB(常数)	KB(常数)	KB(常数)	
	双字存取	ID	0 - 12	0 - 12	0 - 12	0 - 12	0 - 12
		QD	0 - 12	0 - 12	0 - 12	0 - 12	0 - 12
VD		0 - 2044	0 - 2044	0 - 8188	0 - 10236	0 - 10236	
MD		0 - 28	0 - 28	0 - 28	0 - 28	0 - 28	
SMD		0 - 162	0 - 296	0 - 546	0 - 546	0 - 546	
SD		0 - 28	0 - 28	0 - 28	0 - 28	0 - 28	
LD		0 - 60	0 - 60	0 - 60	0 - 60	0 - 60	
AC		0 - 3	0 - 3	0 - 3	0 - 3	0 - 3	
HC		0 - 5	0 - 5	0 - 5	0 - 5	0 - 5	
KD(常数)		KD(常数)	KD(常数)	KD(常数)	KD(常数)	KD(常数)	

存储器标志符

最后一个字节不能作为字地址

最后三个字节不能作为字地址

**注：1.定时器 T 和计数器 C 只能位存取和字存取。
2.AC0~AC3 四个累加器的数据大小视情况而定。**

S7-200 数据的存取

表4-1 不同长度的数据表示的十进制和十六进制数范围

数制	字节(B)	字(W)	双字(D)
无符号整数	0到255 0到FF	0到65,535 0到FFFF	0到4,294,967,295 0到FFFF FFFF
符号整数	-128到 + 127 80到7F	-32,768到+32,767 8000到7FFF	-2,147,483,648到+2,147,483,647 8000 0000到7FFF FFFF
实数IEEE 32 位浮点数	不适用	不适用	+1.175495E-38到+3.402823E+38 (正数) -1.175495E-38到-3.402823E+38 (负数)

在 S7-200 的许多指令中，都可以使用常数值。常数可以是字节、字或者双字。S7-200 以二进制数的形式存储常数，可以分别表示十进制数、十六进制数、ASCII 码或者实数(浮点数)。见表 4-2。

表4-2 常数表示法

数制	格式	举例
十进制	[十进制值]	20047
十六进制	16#[十六进制值]	16#4E4F
二进制	2#[二进制数]	2#1010_0101_1010_0101
ASCII码	'[ASCII码文本]'	'ABCD'
实数	ANSI/IEEE 754-1985	+1.175495E-38 (正数)-1.175495E-38 (负数)
字符串	"[字符串文本]"	"ABCDE"

S7-200 将信息存于不同的存储器单元，每个单元都有唯一的地址。可以明确指出要访问的存储器地址。这就允许用户程序直接访问这个信息。

若要访问存储区的某一位，则必须指定地址，包括**存储器标识符**、**字节地址和位号**。图 4-4 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 = 输入，3 = 字节 3)之后用点号(“.”)来分隔位地址(第 4 位)。

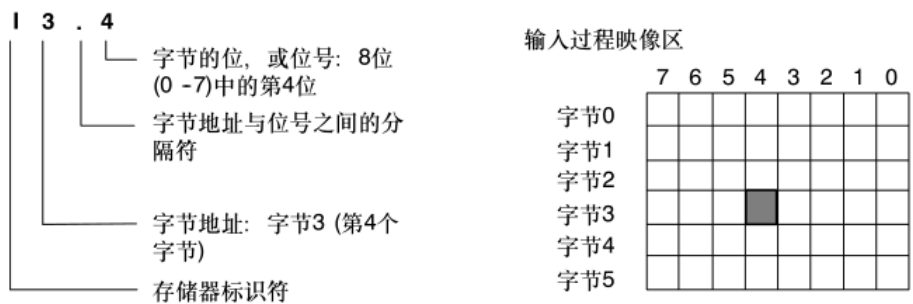


图4-4 字节.位寻址

使用这种字节寻址方式，可以按照字节、字或双字来访问许多存储区(V、I、Q、M、S、L 及 SM)中的数据。若要访问 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**，

如图 4-5 所示。

使用包括区域标识符和设备号的地址格式来访问其他 CPU 存储区(如 T、C、HC 和累加器)中的数据。

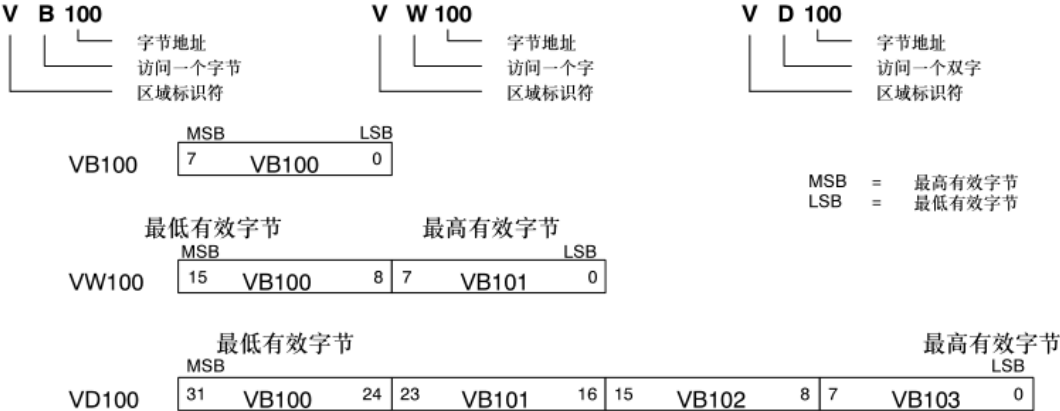


图4-5 对同一地址进行字节、字和双字存取操作的比较。

第四章 PLC 控制实验

实验一 基本指令实验

一、实验目的

掌握常用基本指令的使用方法。

学会用基本逻辑与、或、非等指令实现基本逻辑组合电路的编程。


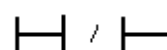





熟悉编译调试软件的使用。


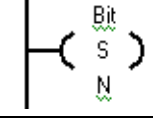
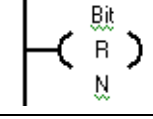
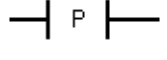
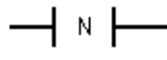
二、实验器材

PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容

SIEMENS S7-200 系列可编程序控制器的常用基本指令有 10 条。本次实验进行常用基本指令 LD、LDN、A、AN、NOT、O、ON、ALD、OLD、= 指令的编程操作训练。先简要介绍如下：

	指令名称	梯形图符	数据	功能
1	取指令 LD		接点号	读入逻辑行（又称为支路）的第一个常开接点。
2	取反指令 LDN		接点号	读入逻辑行的第一个常闭接点。
3	与指令		接点号	逻辑与操作，即串联一个常开接点。
4	与非指令		接点号	逻辑与非操作，即串联一个常闭接点。
5	或指令		接点号	逻辑或操作，即并联一个常开接点。
6	或非指令		接点号	逻辑或非操作，即并联一个常闭接点。
7	非指令 NOT		接点号	逻辑或非操作，即并联一个常闭接点。

8	输出指令= 线圈号		继电器 线圈号	
9	置位指令 S			将从 Bit 指定的地址开始的 N 个点置位。 相当于锁存器，当其置位 (ON) 后，将一直保持，直至复位为止。
10	复位指令 R			将从 Bit 指定的地址开始的 N 个点复位。 相当于锁存器，当其复位 (OFF) 后，将一直保持，直至置位为止。
11	正跳变指令 EU			在输入脉冲的前沿 (由 0 到 1)，使指定的继电器接通一个扫描周期后又释放，即把输入状态的前沿微分输出到指定的继电器。
12	负跳变指令 ED			在输入脉冲的后沿 (由 1 到 0)，使指定的继电器接通一个扫描周期后又释放，即把输入状态的后沿微分输出到指定的继电器。

备注：

在梯形图中，每一逻辑行必须以接点开始，所以必须使用取或取反指令。此外，这条指令还用于电路块中每一支路的开始，或分支点后分支电路的起始，并与其它一此指令配合使用。

置位 (S) 或复位 (R) 指令可以一次置位或复位 1-255 个点。如果复位指令指定的是定时器或者计数器，指令不但复位定时器位或计数器位，而且清除定时器或者计数器的当前值。

正/负跳变这两条微分指令都是在输入状态发生变化时才起作用。在程序运行中，一直接通的输入条件，不会引起 EU 的执行；一直处于断开的输入不会引起 ED 的执行。

四、实验步骤

实验前，先用下载电缆将 PC 机串口与 S7-200-CPU224 主机的 PORT1 端口连好，然后对实验箱通电，并打开 24V 电源开关。主机和 24V 电源的指示灯亮，表示工作正常，可进入下一步实验。

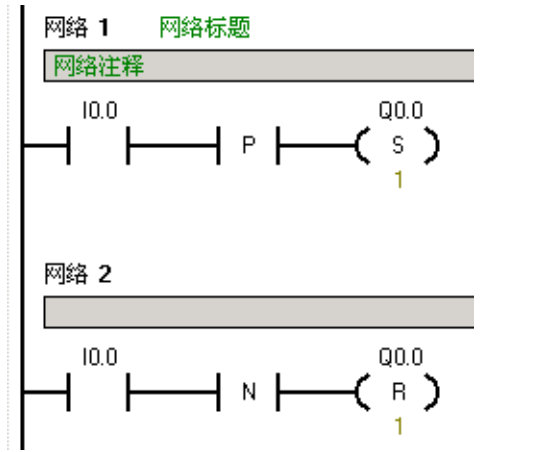
进入编译调试环境，用指令符或梯形图输入下列练习程序。

根据程序，进行相应的连线。(接线可参见第一章中“输入/输出端口的使用方法”)

下载程序并运行，观察运行结果。

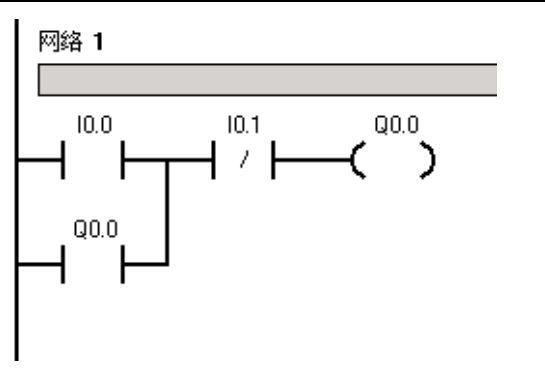
常犯的的错误：双线圈或多线圈输出

练习： 正负跳变指令的编程

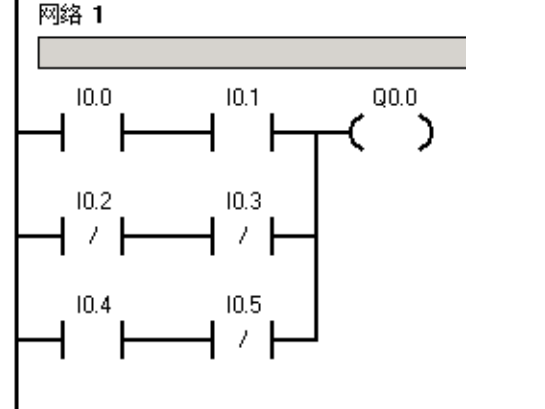
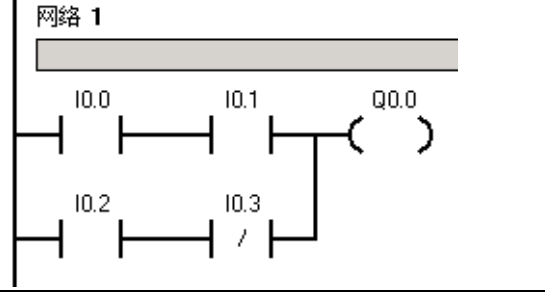


I0.0 接到按钮，
当按钮按下时，相当于在 I0.0 上产生一个由 0 到 1 的跳变，因此输入接通一个扫描周期，Q0.0 置位，即点亮。
当按钮松开时，相当于在 I0.0 上产生一个由 1 到 0 的跳变，因此输入接通一个扫描周期，Q0.0 复位，即熄灭。

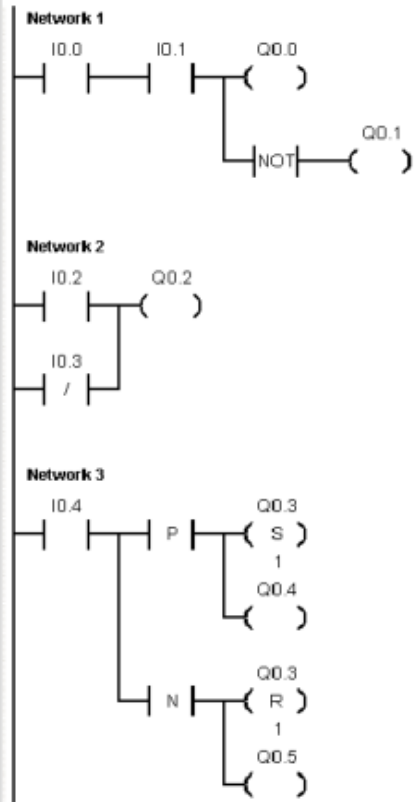
练习： 输入自锁



I0.0 接”启动”按钮， I0.1 接”停止”按钮，
启动按钮按下时， Q0.0 输出为 1， 当按钮松开时， Q0.0 的输出保持。
停止按钮按下时， Q0.0 输出为 0， 当按钮松开时， Q0.0 输出仍为 0。



实例: 触点指令



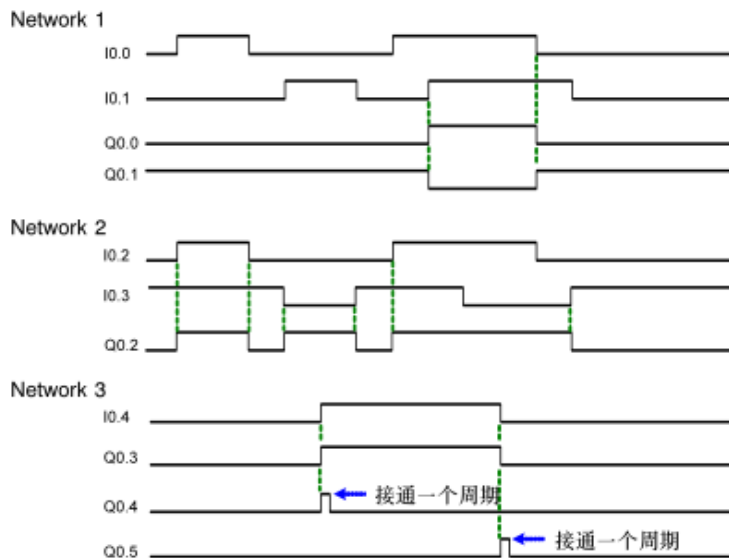
```

Network 1 //要想激活Q0.0, 常开触点I0.0
           //和I0.1必须为接通(闭合)。NOT
           //指令作为一个反向器使用。在RUN
           //模式下, Q0.0和Q0.1具有相反的逻辑状态。
LD      I0.0
A       I0.1
=       Q0.0
NOT
=       Q0.1

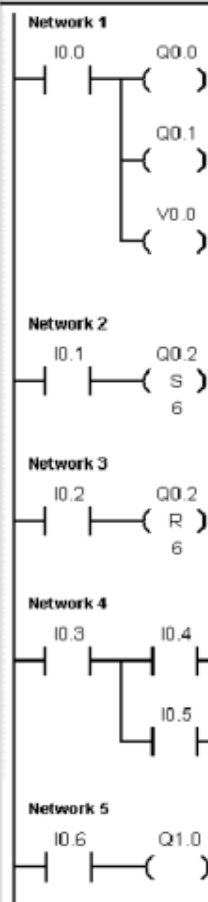
Network 2 //常开接点I0.2必须为接通或者常闭接点
           //I0.3必须为断开以激活Q0.2。
           //个或多个平行LAD分支
           //(“或”逻辑输入)必须为真,
           //以使输出激活。
LD      I0.2
ON      I0.3
=       Q0.2

Network 3 //在P触点的一个上升沿或者在N触点的一个下降
           //沿出现时, 一个扫描周期内输出一个脉冲。
           //在RUN模式,
           //Q0.4和Q0.5的脉冲状态变化太快
           //以至于在程序中无法用状态图监视。
           //置位和复位指令将
           //Q0.3的状态变化锁存,
           //使程序可以监视。
LD      I0.4
LPS
EU
S       Q0.3, 1
=       Q0.4
LPP
ED
R       Q0.3, 1
=       Q0.5
    
```

时序图



实例: 线圈指令



```

Network1 //输出指令为外部I/O(I、Q)和内部存储器
// (M、SM、T、C、V、S、L)指定位值。
LD I0.0
= Q0.0
= Q0.1
= V0.0

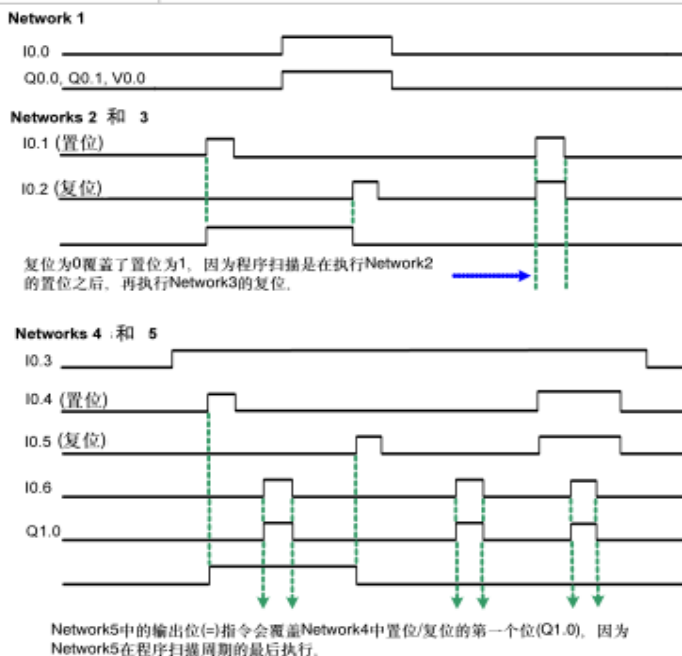
Network 2 //将一个6位顺序组设为数值1。
//指定起始位地址, 以及要设置的位数, 当第一位
// (Q0.2)的值为1时, 置位指令
//的程序状态指示器为ON。
LD I0.1
S Q0.2, 6

Network 3 //将一个6位顺序组复位到数值0。
//指定起始位地址和要复位的位数。
//当第一位(Q0.2)的值为0时, 复位指
//令的程序状态指示器为ON。
LD I0.2
R Q0.2, 6

Network4 //置位和复位一组8个输出位(Q1.0~Q1.7)。
LD I0.3
LPS
A I0.4
S Q1.0, 8
LPP
A I0.5
R Q1.0, 8

Network5 //置位和复位指令实现锁存器功能。
//完成置位/复位功能, 必须确保这些
//位没有在其他指令中被改写。在本例中,
//Network4置位和复位一组
//8个输出位(Q1.0~Q1.7)。在RUN模式
//下Network5会覆盖Q1.0的值, 从而
//控制Network4中的程序状态显示器。
LD I0.6
= Q1.0
    
```

时序图



实验二 置位/复位指令实验

一、实验目的

- 1、掌握 RS 触发器指令的使用方法。
- 2、熟悉编译调试软件的使用。

二、实验器材

PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容及步骤

RS 触发器指令

置位优先触发器是一个置位优先的锁存器。当置位信号(S1)和复位信号(R)都为真时，输出为真。

复位优先触发器是一个复位优先的锁存器。当置位信号(S)和复位信号(R1)都为真时，输出为假。

Bit 参数用于指定被置位或者复位的布尔参数。可选的输出反映 Bit 参数的信号状态。

表 6-7 中给出了范例程序的真值表

表6-6 RS触发器指令的有效操作数

输入/输出	数据类型	操作数
S1、R	BOOL	I、Q、V、M、SM、S、T、C、功率流
S、R1、OUT	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
位	BOOL	I、Q、V、M、S

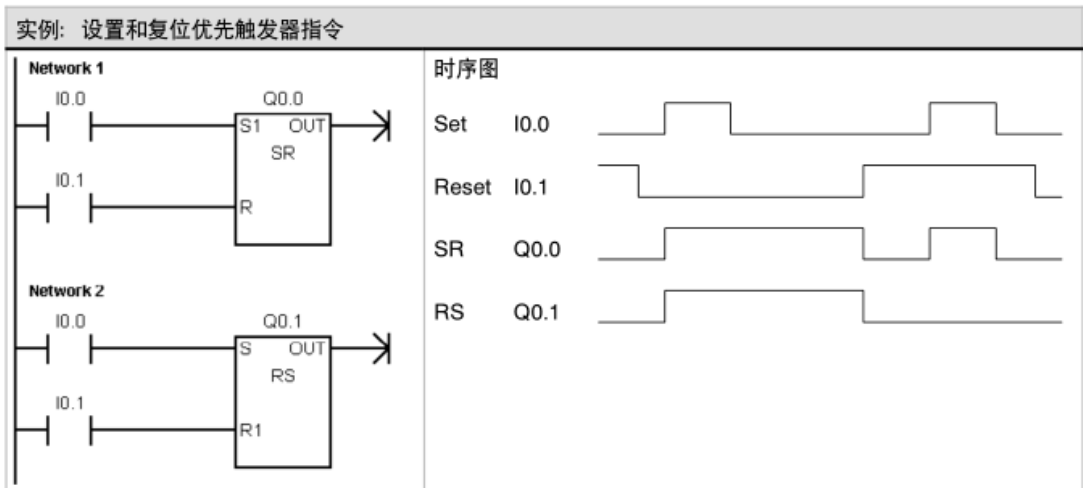


表6-7 RS触发器指令真值表

指令	S1	R	输出(Bit)
置位优先触发器指令(SR)	0	0	保持前一状态
	0	1	0
	1	0	1
	1	1	1
指令	S	R1	输出(Bit)
复位优先触发器指令(RS)	0	0	保持前一状态
	0	1	0
	1	0	1
	1	1	0

四、实验步骤

1、实验前，先用下载电缆将 PC 机串口与 S7-200-CPU224 主机的 PORT1 端口连好，然后对实验箱通电，并打开 24V 电源开关。主机和 24V 电源的指示灯亮，表示工作正常，可进入下一步实验。

2、进入编译调试环境，用指令符或梯形图输入下列练习程序。

3、根据程序，进行相应的连线。（接线可参见第一章中“输入/输出端口的使用方法”）

4、下载程序并运行，观察运行结果。

实验三 跳转指令实验

一、实验目的

- 1、掌握跳转指令的使用方法。
- 2、熟悉编译调试软件的使用。

二、实验器材

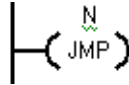
PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容

1、跳转指令

指令符：JMP (N)

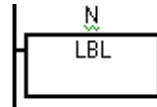
梯形图符：



2、标号指令

指令符：LBL (N)

梯形图符：



跳转指令可以使程序流程跳转到指定的标号 N 处的程序分支。可用于工作方式的选择。
标号指令标记跳转目的地的位置 N。

四、实验步骤

1、实验前，先用下载电缆将 PC 机串口与 S7-200-CPU224 主机的 PORT1 端口连好，然后对实验箱通电，并打开 24V 电源开关。主机和 24V 电源的指示灯亮，表示工作正常，可进入下一步实验。

进入编译调试环境，用指令符或梯形图输入下列练习程序。

根据程序，进行相应的连线。（接线可参见第一章中“输入/输出端口的使用方法”）

下载程序并运行，观察运行结果。

练习 1:

Network 1

LD SM0.0

AN T38

TON T37, +5

Network 2

LD T37

TON T38, +5

= M0.0

Network 3

LD I0.0

JMP 1

Network 4

LD M0.0

= Q0.0

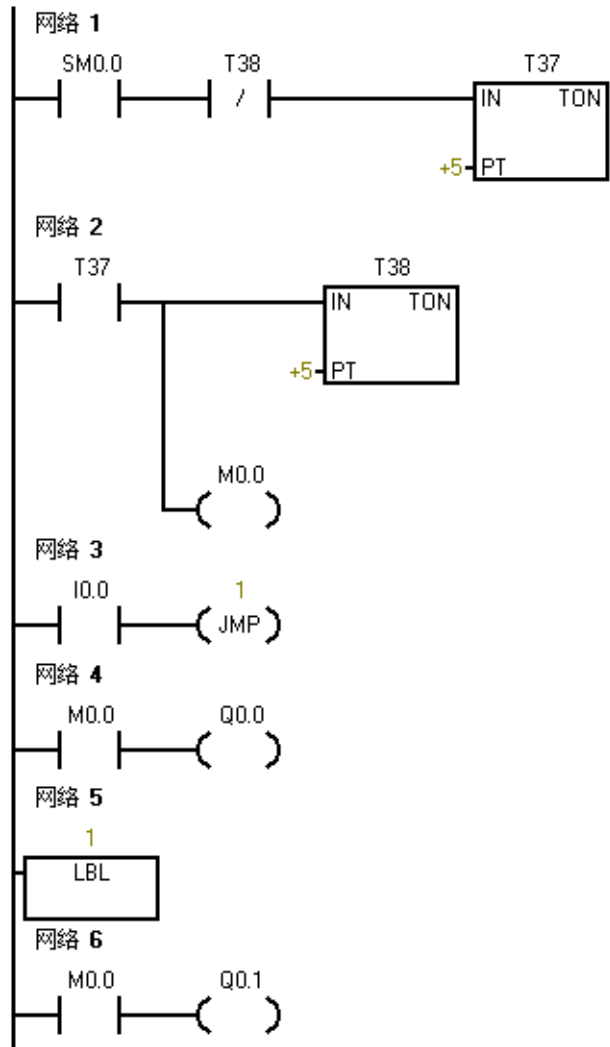
Network 5

LBL 1

Network 6

LD M0.0

= Q0.1



练习 2: 自行设计一个使用跳转指令的程序，并上机验证。

实验四 定时器及计数器指令实验

一、实验目的

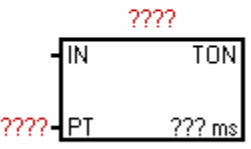
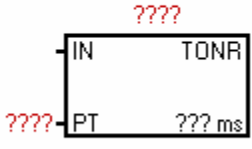
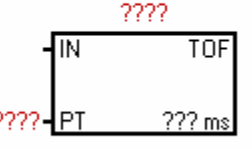
- 1、掌握常用定时指令的使用方法。
- 2、掌握计数器指令的使用
- 3、掌握定时器/计数器内部时基脉冲参的设置。
- 4、熟悉编译调试软件的使用。

二、实验器材

PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容及步骤

定时器的指令及使用

格式	名 称		
	接通延时定时器	有记忆接通延时定时器	断开延时定时器
LAD			
STL	TON T***,PT	TONR T***,PT	TOF T***,PT

指令符：TONxx

梯形图符：

数据：xx (37)：为选定的定时器号；

PT (+10)：是定时器的设定值，范围1~32768。

功能：定时时间到接通定时器接点。

接通延时定时器 TON (On-Delay Timer)

1. 接通延时定时器用于单一时间间隔的定时。
2. 上电周期或首次扫描时，定时器位为 OFF，当前值为 0。
3. 输入端接通时，定时器位为 OFF，当前值从 0 开始计时，当前值达到设定值时，定时器位为 ON，当前值仍连续计数到 32 767。
4. 输入端断开，定时器自动复位，即定时器位为 OFF，当前值为 0。

记忆接通延时定时器 TONR (Retentive On-Delay Timer)

1. 记忆接通延时定时器具有记忆功能,它用于对许多间隔的累计定时。
2. 上电周期或首次扫描时,定时器位为掉电前的状态,当前值保持在掉电前的值。
3. 当输入端接通时,当前值从上次的保持值继续计时,当累计当前值达到设定值时,定时器位 ON,当前值可继续计数到 32 767。
4. TONR 定时器只能用复位指令 R 对其进行复位操作。TONR 复位后,定时器位为 OFF,当前值为 0。掌握好对 TONR 的复位及启动是使用好 TONR 指令的关键。

断开延时定时器 TOF (Off-Delay Timer)

1. 断开延时定时器用于断电或故障事件后的单一间隔时间计时。
2. 上电周期或首次扫描,定时器位为 OFF,当前值为 0。
3. 输入端接通时,定时器位为 ON,当前值为 0。当输入端由接通到断开时,定时器开始计时。当达到设定值时定时器位为 OFF,当前值等于设定值,停止计时。
4. 输入端再次由 OFF→ON 时,TOF 复位,这时 TOF 的位为 ON,当前值为 0。如果输入端再从 ON→OFF,则 TOF 可实现再次启动。

表6-73 定时器指令的操作数

类型	当前值>=预设值	使能输入(IN)的状态	上电周期/首次扫描
TON	定时器位ON, 当前连续计数到32767	ON: 当前值计数时间 OFF: 定时器位关闭, 当前值 = 0	定时器位OFF, 当前值=0
TONR	定时器位ON, 当前连续计数到32767	ON: 当前值计数时间 OFF: 定时器位和当前值保持最后 状态	定时器位OFF 当前值可以保持 ¹
TOF	定时器位OFF 当前值=预设值, 停止计数	ON: 定时器位接通, 当前值 = 0 OFF: 在接通至断开转换后定时器开始 计数	定时器位OFF 当前值=0

¹ 有记忆定时器的当前值可以设为掉电保持, 有关S7-200 CPU有记忆存储器的详细内容参阅第4章。

表6-74 定时器号和分辨率

定时器类型	分辨率	用秒(s)表示的最大值	定时器号
TONR	1ms	32.767 s (0.546分钟)	T0,T64
	10 ms	327.67s	T1 - T4, T65 - T68
	100 ms	3276.7s	T5 - T31, T69-T95
TON、TOF	1ms	32.767s	T32,T96
	10 ms	327.67s	T33 - T36, T97 - T100
	100 ms	3276.7s	T37 - T63, T101 - T255

注: TON 和 TOF 使用相同范围的定时器编号, 所以在**同一个 PLC 程序中决不能把同一个定时器号同时用作 TON 和 TOF**。例如在程序中, 不能既有接通延时 (TON) 定时器 T32, 又有断开延时 (TOF) 定时器 T32

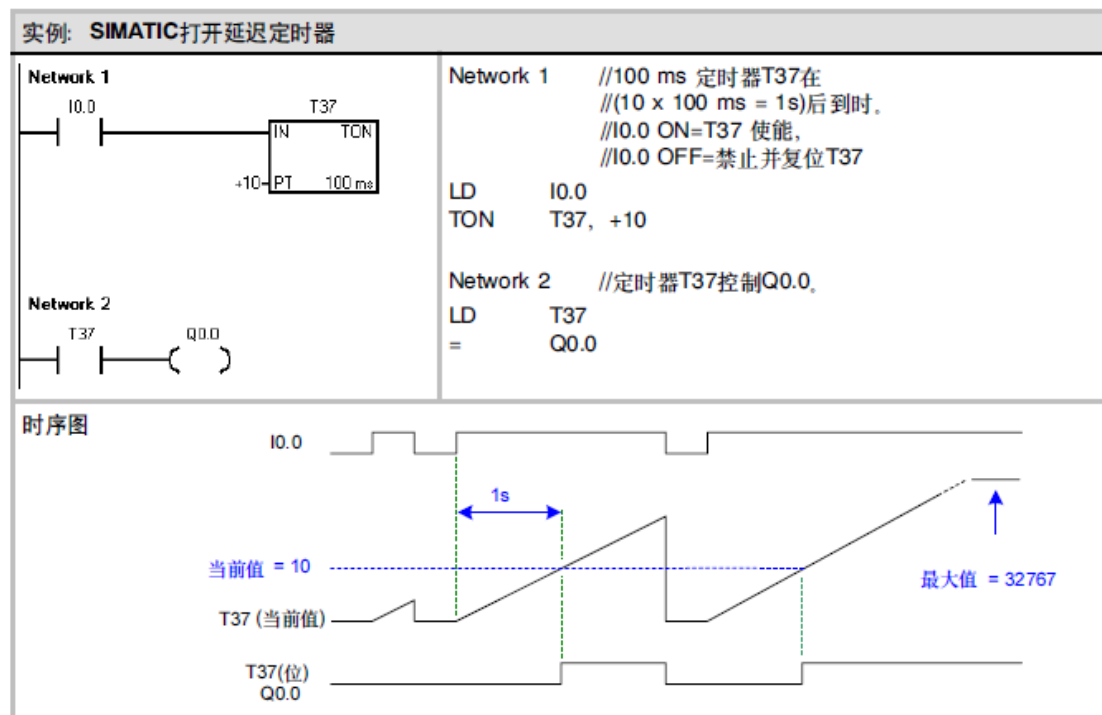
定时器的刷新方式和正确使用

1 ms、10 ms、100 ms 定时器的刷新方式是不同的，从而在使用方法上也有很大的不同。这和其他 PLC 是有很大区别的。

1 ms 定时器 1 ms 定时器由系统每隔 1 ms 刷新一次，与扫描周期及程序处理无关。它采用的是中断刷新方式。因此，当扫描周期大于 1 ms 时，在一个周期中可能被多次刷新。其当前值在一个扫描周期内不一定保持一致。

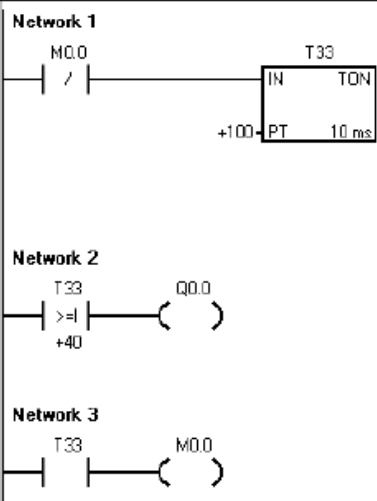
10 ms 定时器 10 ms 定时器由系统在每个扫描周期开始时自动刷新，由于是每个扫描周期只刷新一次，故在一个扫描周期内定时器位和定时器的当前值保持不变。

100 ms 定时器 100 ms 定时器在定时器指令执行时被刷新，因此，如果 100 ms 定时器被激活后，如果不是每个扫描周期都执行定时器指令或在一个扫描周期内多次执行定时器指令，都会造成计时失准，所以在后面讲到的跳转指令和循环指令段中使用定时器时，要格外小心。100 ms 定时器仅用在定时器指令在每个扫描周期执行一次的程序中。



为了确保在每一次定时器达到预设值时，自复位定时器的输出都能接通一个程序扫描周期，用一个常闭触点来代替定时器位作为定时器的使能输入。

实例: SIMATIC自复位打开延迟定时器



Network 1 //10 ms 定时器T33在
 //(100 x 10 ms = 1s)后到时。
 //M0.0脉冲过快，以致在状态视图中
 //无法监视

LDN M0.0
 TON T33, +100

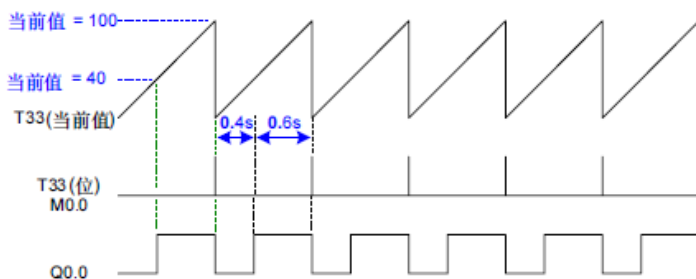
Network 2 //比较指令为真的时间较长，
 //可以在状态表中监视，
 //Q0.0的占空比为40%。

LDW>= T33, +40
 = Q0.0

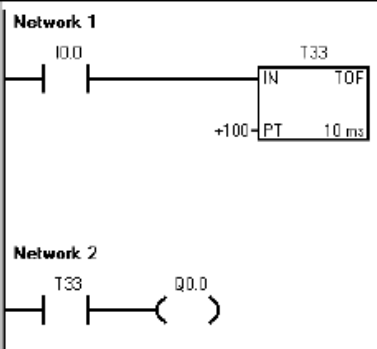
Network 3 //T33 (位)的脉冲过窄，
 //在状态表中无法监视。
 //在1秒后复位M0.0。

LD T33
 = M0.0

时序图



实例: SIMATIC关断延迟定时器



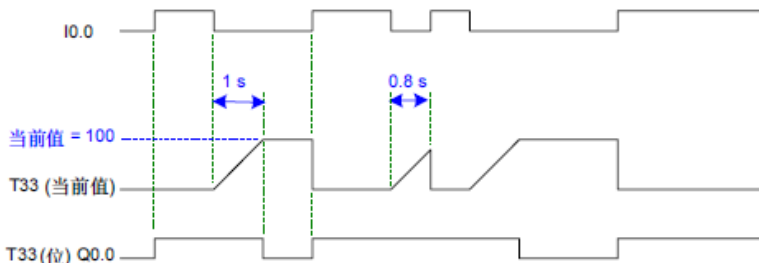
Network 1 //10 ms定时器T33在(100 x 10 ms = 1s)后到时，
 //I0.0关断使能T33，
 //I0.0接通T33复位。

LD I0.0
 TOF T33, +100

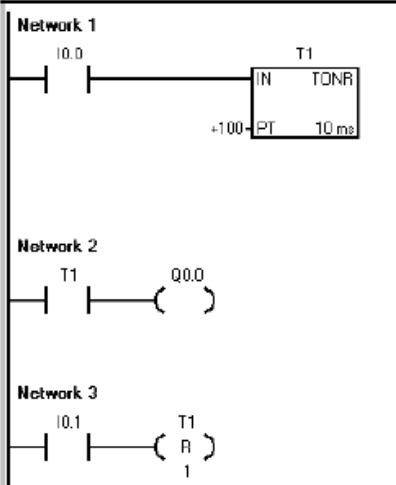
Network 2 //定时器T33用其输出位控制Q0.0。

LD T33
 = Q0.0

时序图



实例: SIMATIC保持性打开延迟定时器

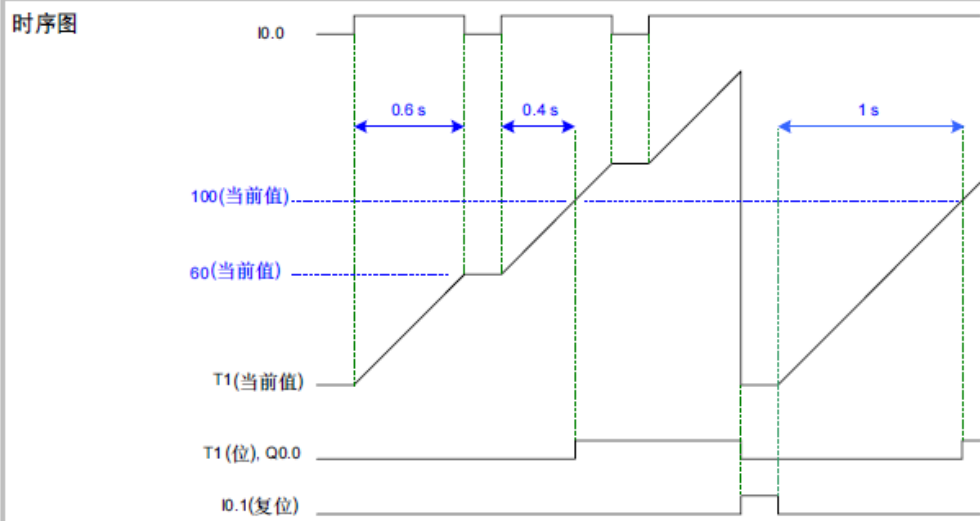


```

Network 1 //10 ms TONR定时器T1在
           //PT=(100 x 10 ms=1s)后到时。
LD I0.0
TONR T1, +100

Network 2 //T1位控制Q0.0。
           //1秒后T1使Q0.0接通。
LD T1
= Q0.0

Network 3 //TONR定时器必须用复位
           //指令才能复位。
           //当I0.1接通时，复位T1。
LD I0.1
R T1, 1
    
```



3、 计数器指令

名称 格式	增计数器	增减计数器	减计数器
LAD			
STL	CTU C***, PV	CTUD C***, PV	CTD C***, PV

指令符：CTU、CTD 梯形图符：

其中：

Cxx: xx 是计数器号 00~255;

PV: 是计数设定值, 计数范围是 1~32767。

R、LD: 为复位端。

CTU 为增计数器; CTD 为减计数器。

功能：

增计数器 CTU:

1. 首次扫描时, 计数器位为 OFF, 当前值为 0。
2. 在计数脉冲输入端 CU 的每个上升沿, 计数器计数 1 次, 当前值增加一个单位。
当前值达到设定值时, 计数器位为 ON, 当前值可继续计数到 32 767 后停止计数。
3. 复位输入端有效或对计数器执行复位指令, 计数器自动复位, 即计数器位为 OFF, 当前值为 0。
4. 注意: 在语句表中, CU、R 的编程顺序不能错误

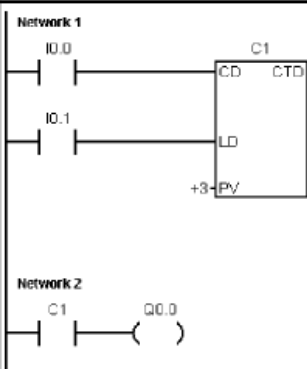
减计数器 CTD:

1. 首次扫描时, 计数器位为 ON, 当前值为预设设定值 PV。
2. 对 CD 输入端的每个上升沿计数器计数 1 次, 当前值减少一个单位, **当前值减小到 0 时, 计数器位置位为 ON**。
3. 复位输入端有效或对计数器执行复位指令, 计数器自动复位, 即计数器位 OFF, 当前值复位为设定值。
4. 注意: 减计数器的复位端是 LD, 而不是 R。在语句表中, CD、LD 的顺序不能错误

增减计数器 CTUD

1. 增减计数器有两个计数脉冲输入端: CU 输入端用于递增计数, CD 输入端用于递减计数。
2. 首次扫描时, 计数器位为 OFF, 当前值为 0。CU 输入的每个上升沿, 计数器当前值增加 1 个单位; CD 输入的每个上升沿, 都使计数器当前值减小 1 个单位, **当前值达到设定值时, 计数器位置位为 ON**。
3. 增减计数器当前值计数到 32 767 (最大值) 后, 下一个 CU 输入的上升沿将使当前值跳变为最小值 (-32 768); 当前值达到最小值-32 768 后, 下一个 CD 输入的上升沿将使当前值跳变为最大值 32767。
4. 复位输入端有效或使用复位指令对计数器执行复位操作后, 计数器自动复位, 即计数器位 OFF, 当前值为 0。
5. 注意: 在语句表中, CU、CD、R 的顺序不能错误。

实例: SIMATIC减计数器指令

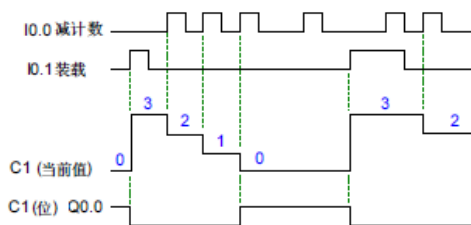


```

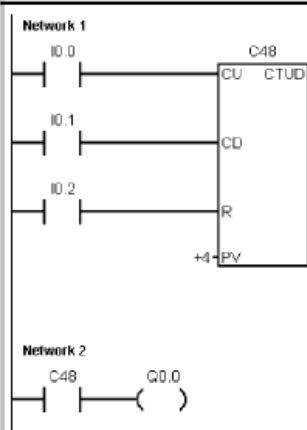
Network1 //当I0.1断开时,
//减计数器C1的当前
//值从3变到0, I0.0的上升沿使C1的
//当前值减少, I0.1接通时装载预设值3.
LD I0.0
LD I0.1
CTD C1, +3

Network 2 //当计数器C1的当前值=0时, C1接通.
LD C1
= Q0.0
    
```

时序图



实例: SIMATIC增/减计数器指令

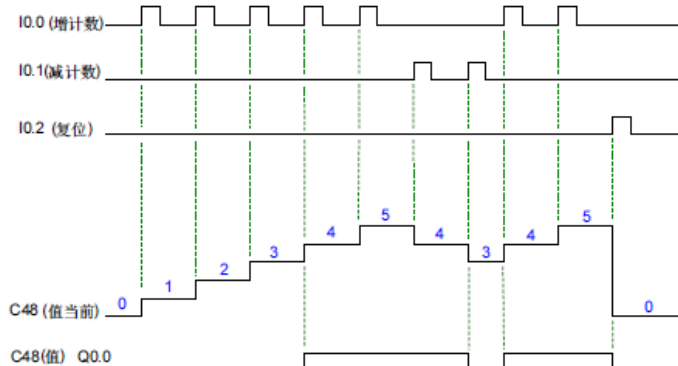


```

Network 1 //I0.0增计数
//I0.1减计数
//I0.2将当前值复位为0
LD I0.0
LD I0.1
LD I0.2
CTUD C48, +4

Network2 //当当前值>=4时,
//将增/减计数器C48接通.
LD C48
= Q0.0
    
```

时序图



四、实验步骤

1、实验前，先用下载电缆将 PC 机串口与 S7-200-CPU224 主机的 PORT1 端口连好，然后对实验箱通电，并打开 24V 电源开关。主机和 24V 电源的指示灯亮，表示工作正常，可进入下一步实验。

进入编译调试环境，用指令符或梯形图输入下列练习程序。

根据程序，进行相应的连线。（接线可参见第一章中“输入/输出端口的使用方法”）

下载程序并运行，观察运行结果。

实验五 移位寄存器指令实验

一、实验目的

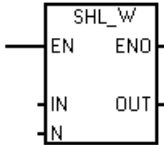
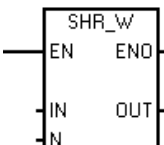
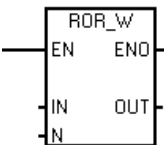
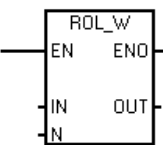
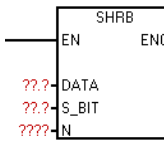
- 1、掌握移位指令的使用方法。
- 2、熟悉编译调试软件的使用。

二、实验器材

PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容

S7-200 提供丰富的移位指令，有左移、右移指令，循环左移、循环右移指令，以及字节移位、字移位、双字移位指令。这里介绍几个常用的指令，其它的请参见有关手册。

左移指令	右移指令	循环左移指令	循环右移指令	移位寄存器指令
				

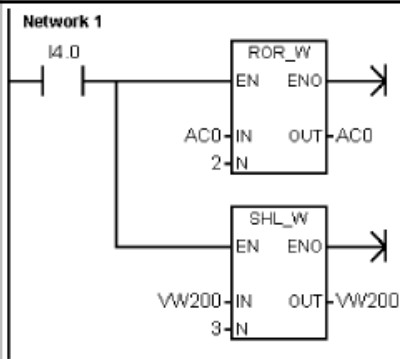
移位指令将输入值 IN 右移或者左移 N 位，并将输出结果装载到 OUT 中。

移位指令对移出的位自动补零。如果位数 N 大于或等于最大允许值（对于字节操作为 8，对于字操作为 16，对于双字操作为 32），那么移位操作的次数为最大允许值。如果移位次数大于 0，溢出标志位（SM1.1）上就是最近移出的位值。如果移位操作的结果为 0，零存储器位（SM1.0）置位。

循环移位指令将输入值 IN 循环右移或者循环左移 N 位，并将输出结果装载到 OUT 中。如果位数 N 大于或等于最大允许值（对于字节操作为 8，对于字操作为 16，对于双字操作为 32），S7-200 在执行循环移位之前，会执行取模操作，得到一个有效的移位次数。取模操作的结果对于字节操作为 0 到 7，对于字操作为 0 到 15，对于双字操作为 0 到 31。如果移位次数为 0，循环移位指令不执行。如果循环移位指令执行，最后一位的值会复制到溢出标志位（SM1.1）。

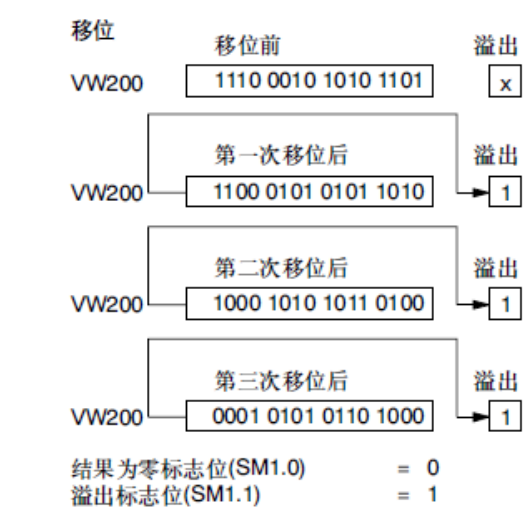
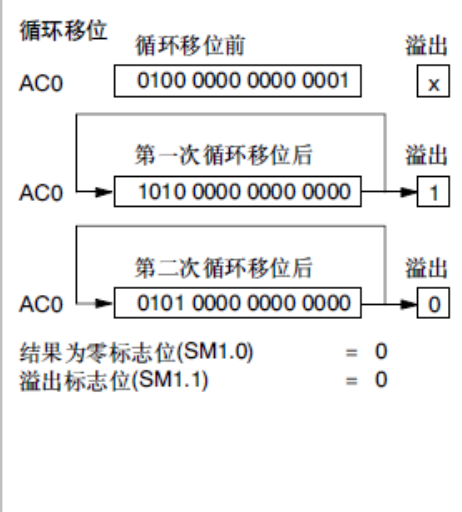
字节操作是无符号的，对于使用字或双字操作，当使用符号数据类型时，符号位也被移动。

实例：移位和循环移位指令



```

Network 1
LD    I4.0
RRW   AC0, 2
SLW   VW200, 3
    
```



移位寄存器指令将一个数值移入移位寄存器中。移位寄存器指令提供了一种排列和控制产品流或者数据的简单方法。使用该指令，每个扫描周期，整个移位寄存器移动一位。移位寄存器指令把输入的DATA数值移入移位寄存器。其中，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(正向移位=N，反向移位=-N)。SHRB指令移出的每一位都被放入溢出标志位(SM1.1)。这条指令的执行取决于最低有效位(S_BIT)和由长度(N)指定的位数。

位移位寄存器的最高位(MSB.b)可通过下面公式计算求得：

$$MSB.b = [(S_BIT的字节号) + ([N] - 1 + (S_BIT的位号)) / 8]. [除8的余数]$$

例如：如果S_BIT是V33.4、N是14，下列计算显示

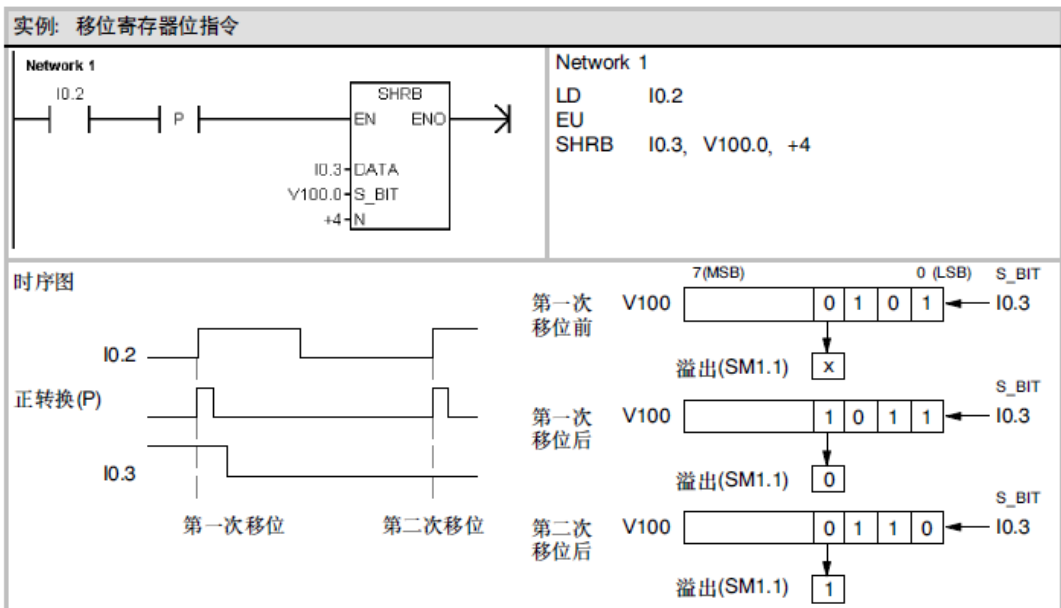
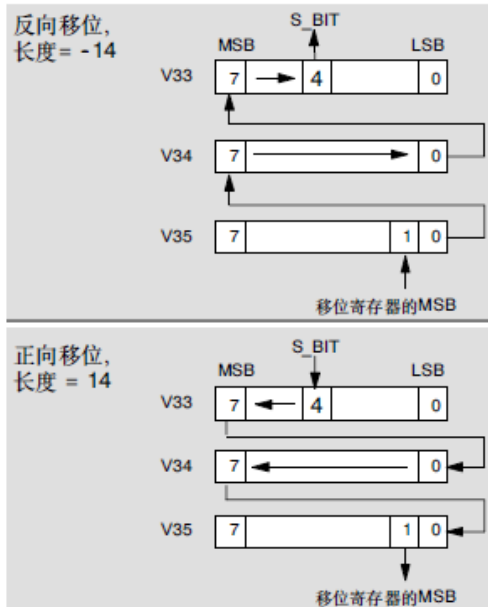
MSB.b是V35.1。

$$\begin{aligned}
 MSB.b &= V33 + ([14] - 1 + 4) / 8 \\
 &= V33 + 17 / 8 \\
 &= V33 + 2 \text{ (余数为1)} \\
 &= V35.1
 \end{aligned}$$

当反向移动时，N为负值，输入数据从最高位移入，最低位(S_BIT)移出。移出的数据放在溢出标志位(SM1.1)中。

当正向移动时，N为正值，输入数据从最低位(S_BIT)移入，最高位移出。移出的数据放在溢出标志位(SM1.1)中。

移位寄存器的最大长度为64位，可正可负。图6--35中给出了N为正和负两种情况下的移位过程。



字节交换指令用来交换输入字 IN 的高字节和低字节，输入数据类型为字型。

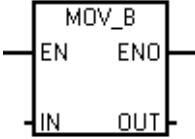
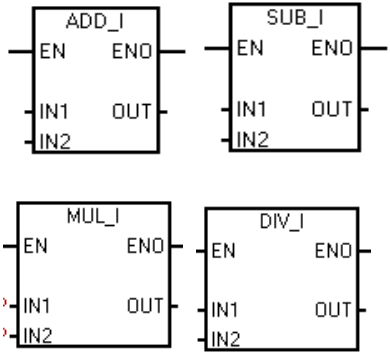
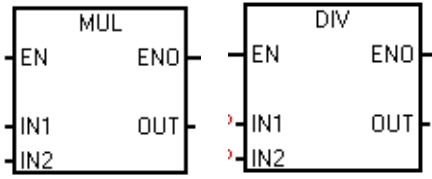
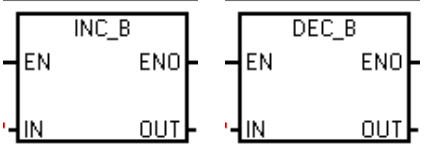
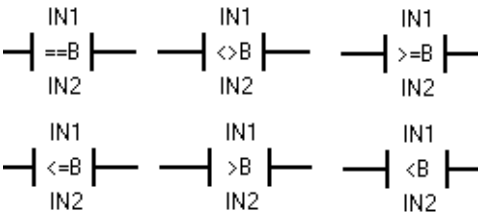
实验六 常用功能指令实验

一、实验目的

- 1、掌握数据比较指令、数据传送指令、加减乘除法指令、逻辑操作指令的使用方法。
- 2、熟悉编译调试软件的使用。

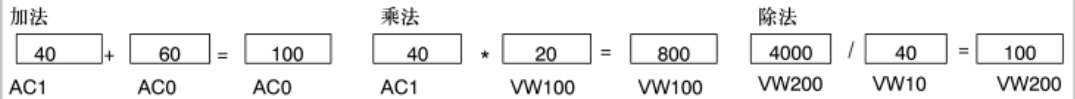
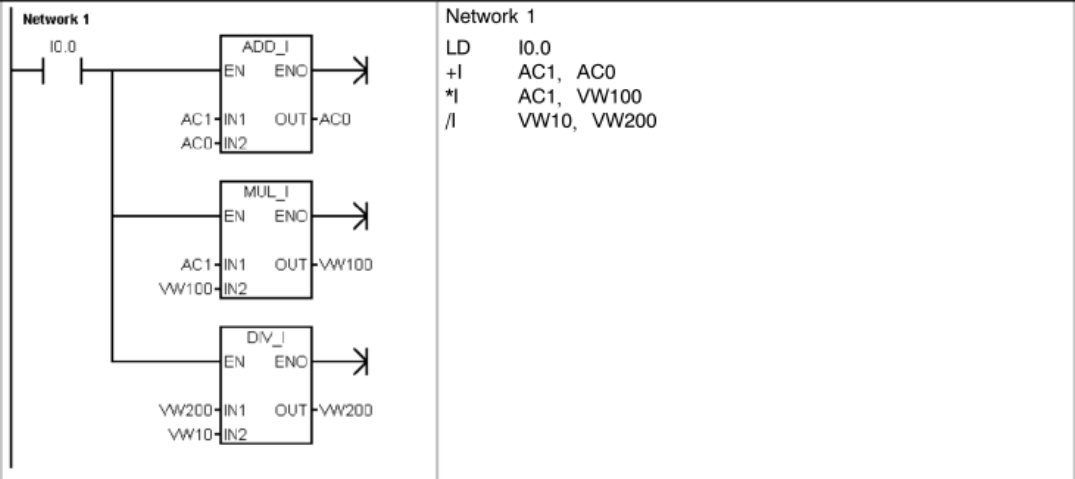
二、实验器材

PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

	<p>传送指令 (MOV_□) 在不改变原值的情况下将 IN 中的值传送到 OUT 中。</p>
	<p>整数运算 (□_I) 指令：将两个 16 位整数加/减/乘/除运算，产生一个 16 位整数结果。</p> <p>双整数运算 (□_DI) 指令：将两个 32 位整数加/减/乘/除运算，产生一个 32 位双整数结果。</p> <p>实数运算 (□_R) 将两个 32 位实数加/减/乘/除运算，产生一个 32 位实数结果。</p>
	<p>整数乘法产生双整数指令 (MUL)，将两个 16 位整数相乘 $IN1 \times IN2$，得到 32 位结果 OUT。</p> <p>带余数的整数除法指令 (DIV)，将两个 16 位整数相除，得到 32 位结果。其中 16 位为余数 (高 16 位字中)，另外 16 位为商 (低 16 位字中)。</p>
	<p>增加 (INC_□) 或者减少指令 (DEC_□) 将输入 IN 加 1 或者减 1，并将结果存放在 OUT 中。</p> <p>字节操作是无符号的。字和双字操作是有符号的。</p>
	<p>比较指令用于比较两个数值： $IN1 = IN2$, $IN1 \geq IN2$, $IN1 \leq IN2$ $IN1 > IN2$, $IN1 < IN2$, $IN1 \neq IN2$</p> <p>字节 (B) 比较操作是无符号的。 整数 (I) 和双字 (D) 比较操作是有符号的。 当比较结果为真时，比较指令使能点闭合或者输出接通。</p>

三、实验内容

实例：整数数学运算指令



实例：实数数学运算指令

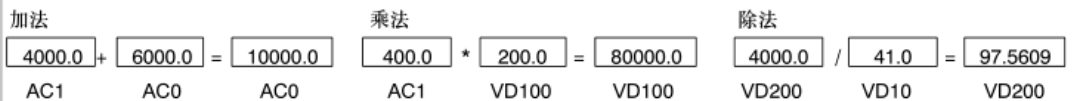
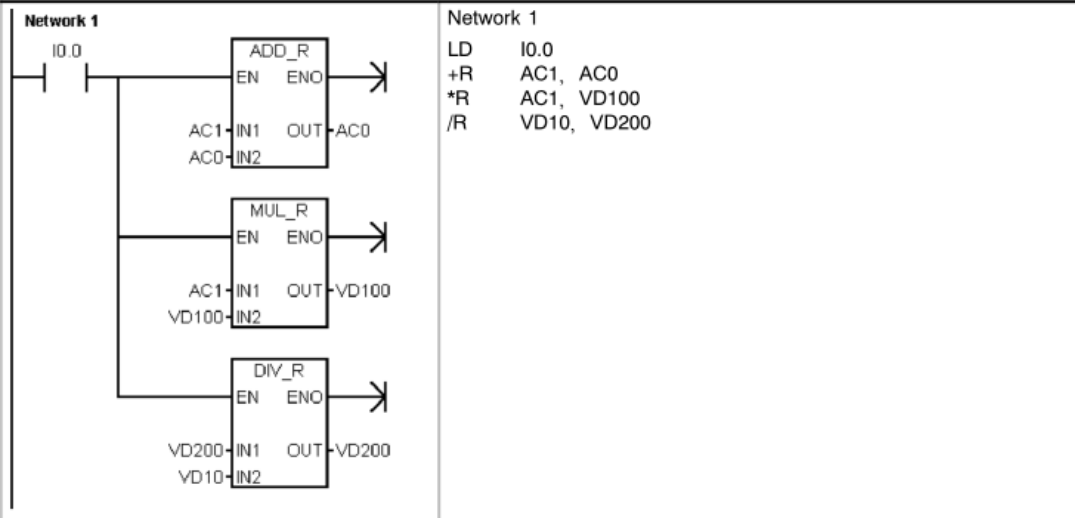


表6-40 整数乘法产生双整数和带余数的整数除法指令的有效操作数

输入/输出	数据类型	操作数
IN1、IN2	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
OUT	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

实例：乘以整数到长整数指令和除以整数带余数指令

Network 1

```
LD I0.0
MUL AC1, VD100
DIV VW10, VD200
```

整数乘法产生双整数

400	*	200	=	80000
AC1		VW102		VD100

带余数的整数除法

4000	/	41	=	23	97
VW202		VW10		余数 VW200	商 VW202 VD200

注意：VD100包括：VW100和VW102，VD200包括：VW200和VW202。

四、实验步骤

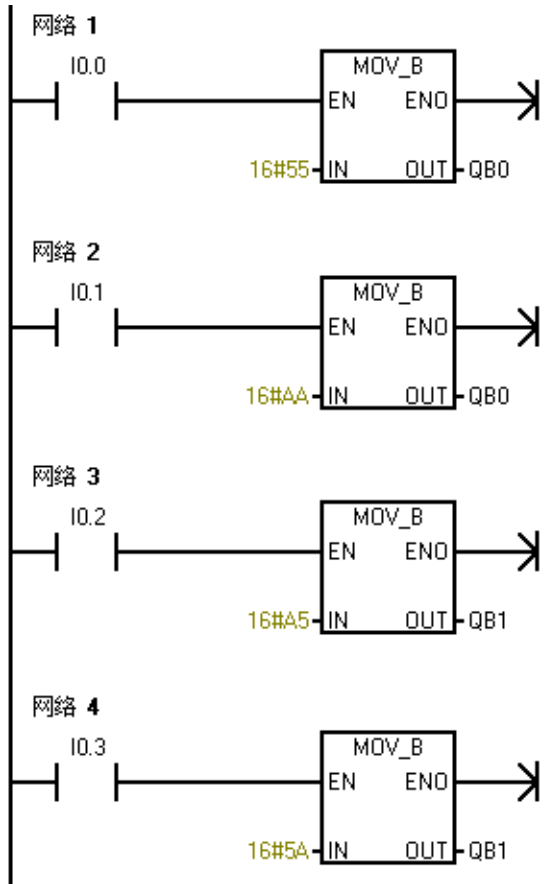
1、实验前，先用下载电缆将 PC 机串口与 S7-200-CPU224 主机的 PORT1 端口连好，然后对实验箱通电，并打开 24V 电源开关。主机和 24V 电源的指示灯亮，表示工作正常，可进入下一步实验。

进入编译调试环境，用指令符或梯形图输入下列练习程序。

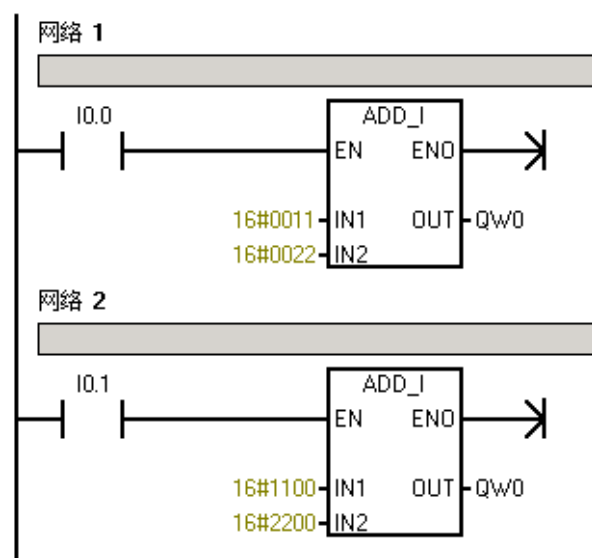
根据程序，进行相应的连线。（接线可参见第一章中“输入/输出端口的使用方法”）

下载程序并运行，观察运行结果。

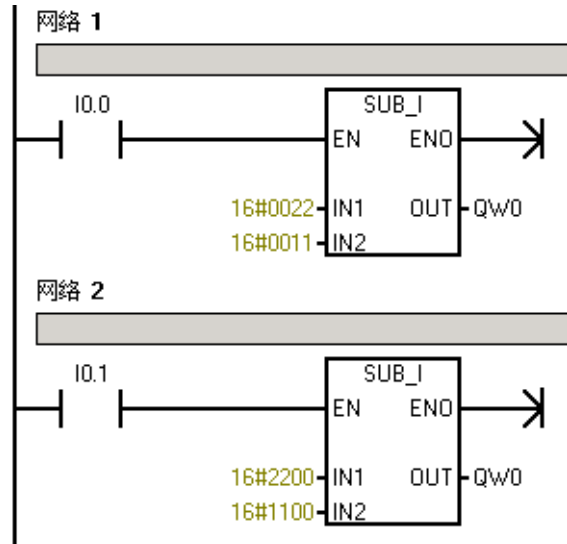
练习 1、MOV 指令



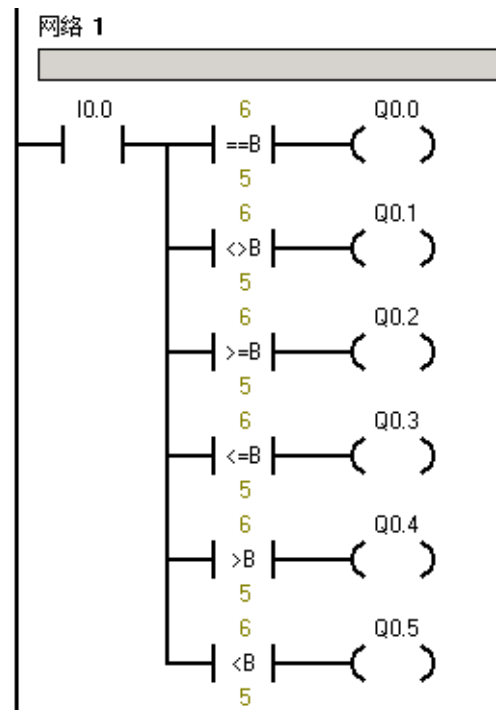
练习 2、加法指令



练习 3、减法指令



练习 4、比较指令



练习 5、按自己的想法修改 练习 1——练习 4 的程序中的指令，如把字节指令换成字指令或双字指令，或者自己编写另外的程序，观察运行结果。熟悉指令的应用。

实验七 舞台灯的 PLC 控制

一、实验目的

- 1、掌握移位寄存器指令的应用
- 2、掌握 PLC 与外围电路的接口连线

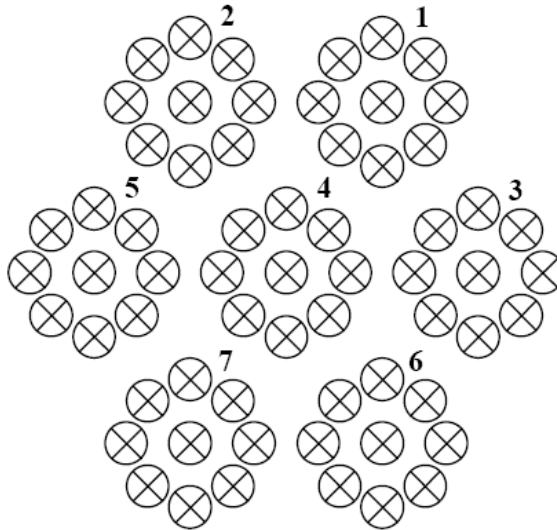
二、实验器材

PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容及步骤

1、设计要求

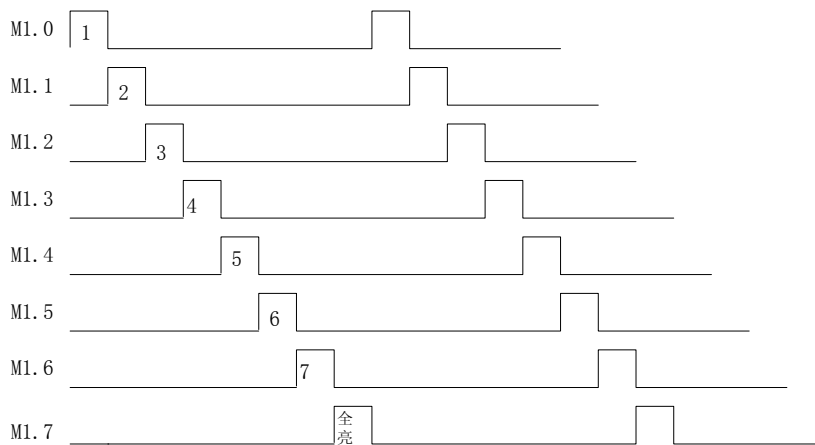
舞台灯光控制可以采用 PLC 来控制，如灯光的闪耀、移位及各种时序的变化。舞台灯控制模块共有 7 组指示灯组成，如下图所示：



舞台灯控制示意图

现要求 1~7 组灯闪亮的时序如下：

- (1) 1~7 号灯依次点亮，再全亮。
 - (2) 重复 (1)，循环往复。
- 2、确定输入、输出端口、并编写程序
 - 3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。
 - 4、调试程序，直至符合设计要求。



时序图

表 7-1 输入输出接线图

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	K1	启动	Q0.0	1	灯 1 区
I0.1	K2	停止	Q0.1	2	灯 2 区
I0.3	K4	复位	Q0.2	3	灯 3 区
1M	24V	开关公共端	Q0.3	4	灯 4 区
	0V←→COM		Q0.4	5	灯 5 区
			Q0.5	6	灯 6 区
			Q0.6	7	灯 7 区
			1L、2L		24V

实验八 LED 数码管显示控制

一、实验目的

- 1、掌握移位寄存器指令的应用
- 2、掌握用 PLC 控制数码管显示

二、实验器材

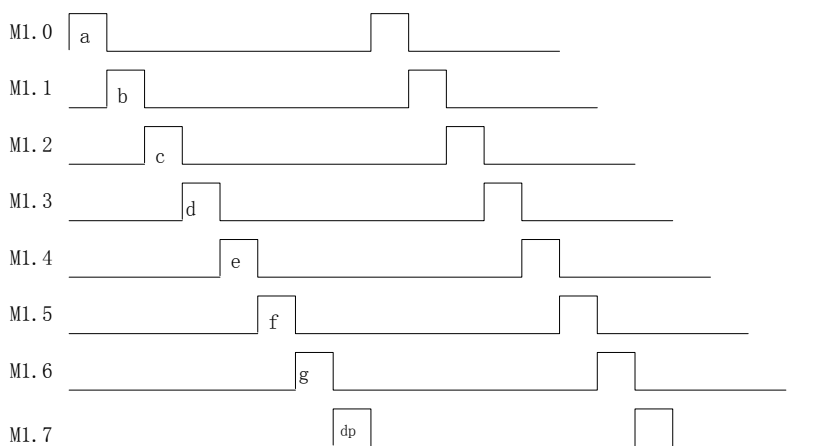
PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容及步骤

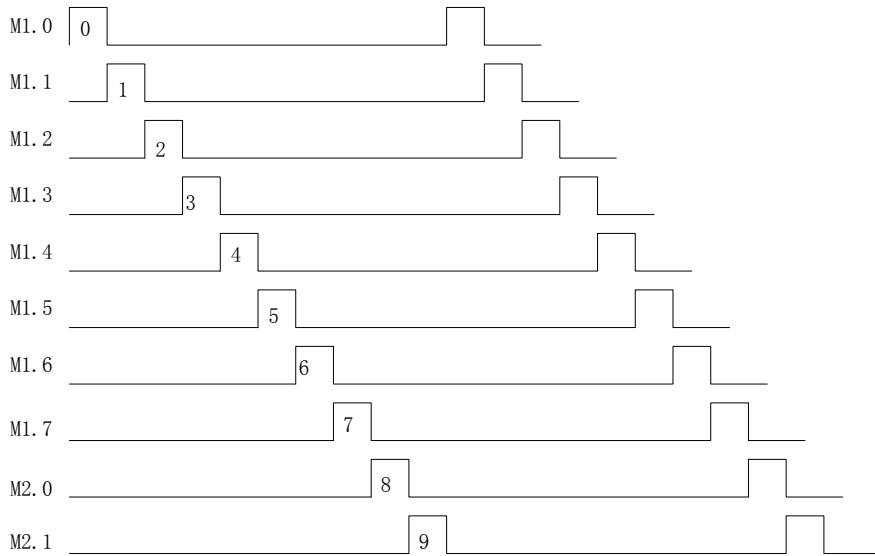
1、设计要求

设计一个数码管循环显示程序。显示值数字 0~9。数码管为共阴极型。A、B、C、D、E、F、G、Dp 为数码管段码，COM 为数码管公共端（位码），当段码输入高电平，位码输入低电平时，相应的段点亮。

- 2、确定输入、输出端口、并编写程序
- 3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。
- 4、调试程序，直至符合设计要求。



段码循环时序



0~9 数字循环时序

表 8-1 输入输出接线图

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	启动	启动	Q0.0	A	段码 A
I0.1	停止	停止	Q0.1	B	段码 B
I0.2	K1	0-9 循环	Q0.2	C	段码 C
I0.3	K2	段码循环	Q0.3	D	段码 D
			Q0.4	E	段码 E
			Q0.5	F	段码 F
			Q0.6	G	段码 G
			Q0.7	DP	小数点
	COM←→0V	开关公共端		COM←→0V	LED 公共端
1M	24V		1L	24V	
			2L	24V	

实验九 交通信号灯的自动控制

一、实验目的

- 1、掌握 PLC 功能指令的用法
- 2、掌握用 PLC 控制交通灯的方法

二、实验器材

PC 机一台、PLC 实验箱一台、编程电缆一根、导线若干

三、实验内容及步骤

1、设计要求

设计一个十字路口交通信号灯的 control 程序。要求为：南北向红灯亮 10 秒，东西向绿灯亮 4 秒闪 3 秒，东西向黄灯亮 3 秒，然后东西向红灯亮 10 秒，南北向绿灯亮 4 秒闪 3 秒，南北向黄灯亮 3 秒，并不断循环反复。

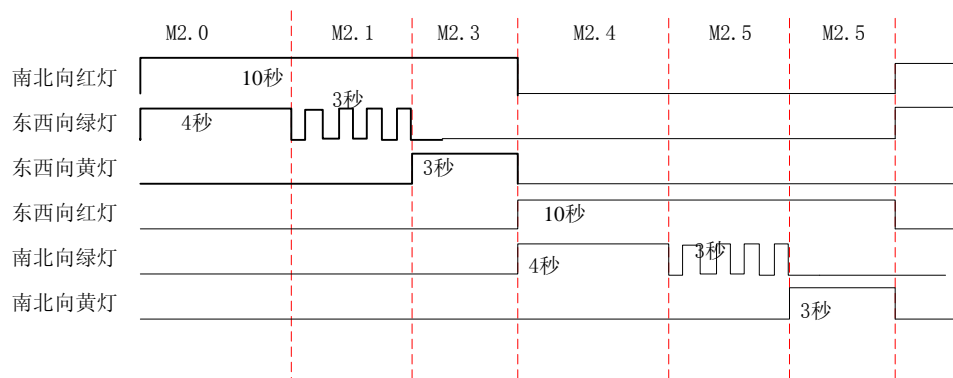
2、确定输入、输出端口、并编写程序。

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

表 9-1 输入输出接线图

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	K1	启动	Q0.0	A1	红灯（上下）
I0.1	K2	停止	Q0.1	A2	绿灯（上下）
1M	24V		Q0.2	A3	黄灯（上下）
	0V←→COM	开关公共端	Q0.3	B1	红灯（左右）
			Q0.4	B2	绿灯（左右）
			Q0.5	B3	黄灯（左右）
				CA1←→B1	人行道红灯（上下）
				CA2←→B2	人行道绿灯（上下）
				CB1←→A1	人行道红灯（左右）
				CB2←→A2	人行道绿灯（左右）
			1L	24V	



红绿灯时序图

实验十 驱动步进电机的 PLC 控制

一、实验目的

- 1、掌握 PLC 功能指令的用法
- 2、掌握用 PLC 控制步进电机的方法

二、实验器材

- 1、PC 机一台 2、PLC 实验箱一台 3、步进电机控制模块 4、编程电缆一根 5、导线若干

三、实验内容及步骤

1、设计要求

控制模块中的步进电机工作方式为四相八拍，电机的四相线圈分别用 A、B、C、D 表示，公共端已接地。

当电机正转时，其工作方式如下：A→AB→B→BC→C→CD→D→DA→A。

当电机反转时，其工作方式如下：A→-AD→D→DC→C→CB→B→BA→A。

设计程序，要求能控制步进电机正反转，并能控制它的转速。

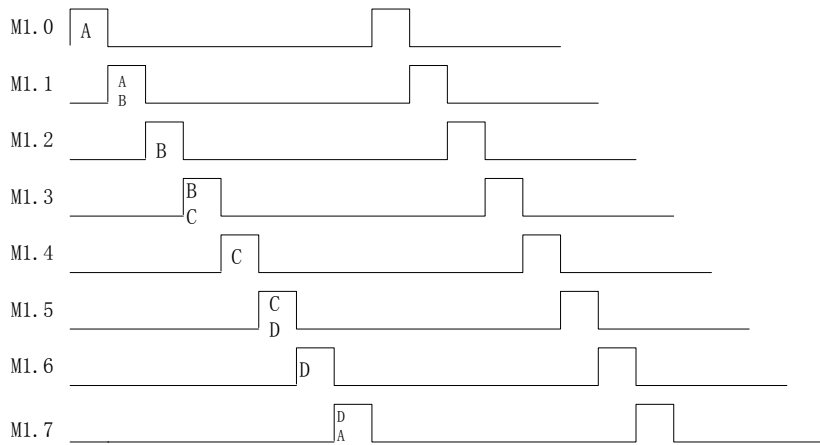
2、确定输入、输出端口、并编写程序

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

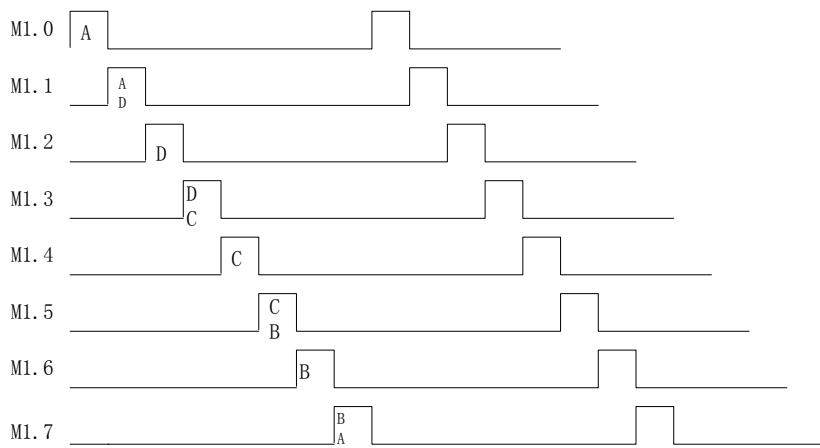
4、调试程序，直至符合设计要求。

表 10-1 输入输出接线图

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	启动	启动	Q0.0	A	步进电机 A 相
I0.1	停止	停止	Q0.1	B	步进电机 B 相
I0.2	正转	正转	Q0.2	C	步进电机 C 相
I0.3	反转	反转	Q0.3	D	步进电机 D 相
I0.4	快速	快速			
I0.5	慢速	慢速	1L	24V	
1M	24V				
	0V←→COM	开关公共端			



正转时序



反转时序

实验十一 电机的星/三角启动控制

一、实验目的

- 1、掌握 PLC 功能指令的用法
- 2、掌握用 PLC 控制交流电机的可逆起动控制电路及星/三角起动的电路

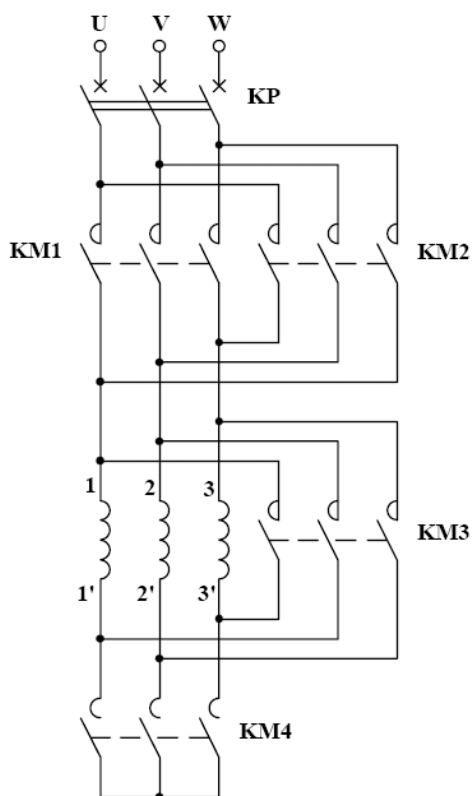
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、电机控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

设计通过 PLC 控制电机的 Y/ Δ 起动电路的程序。



电机星三角控制示意图

当按下正转启动按钮时，电机正转（继电器 KM1 控制），并运行在 Y 形接法（低速运行，继电器 KM4 控制）。过 3 秒后 KM4 断开，电机运行在△接法（全速运行，继电器 KM3 控制）。

当按下停止按钮时，电机停转。

当按下反转启动按钮时，电机反转（继电器 KM2 控制），并运行在 Y 形接法（低速运行，继电器 KM4 控制）。过 3 秒后 KM4 断开，电机运行在△接法（全速运行，继电器 KM3 控制）。

2、确定输入、输出端口、并编写程序

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

表 11-1 输入输出接线图

输入			输出				
主机	实验模块	注释	主机	实验模块	注释		
I0.0	正转	正转	Q0.0	J1 (KM1)	电机正转		
I0.1	反转	反转	Q0.1	J2 (KM2)	电机反转		
I0.2	停止	停止	Q0.2	J3 (KM3)	三角形接法		
			Q0.3	J4 (KM4)	星形接法		
1M		24V	1L		0V		
				实验区	24V	电源区	24V
					0V		0V

实验十二 机械手的 PLC 自动控制

一、实验目的

- 1、掌握 PLC 功能指令的用法
- 2、掌握用 PLC 对机械手步进控制程序的设计

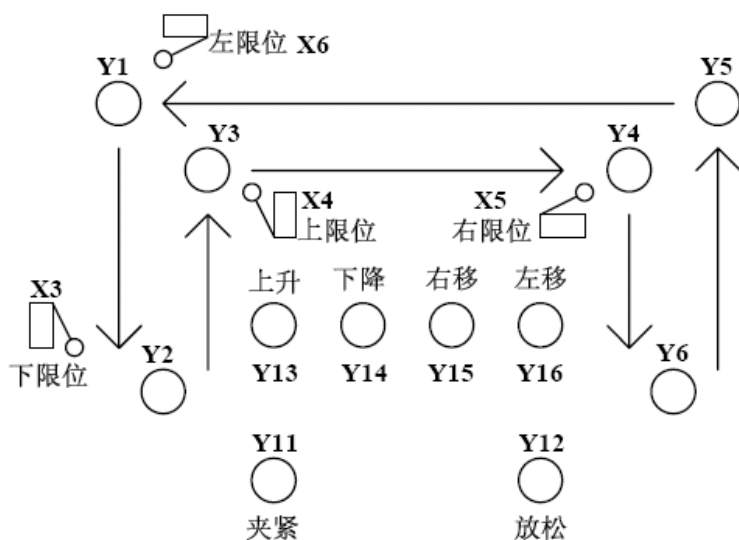
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、机械手控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

机械手工作示意图如下：



机械手控制示意图

控制过程如下：

开始时，机械手处于原始位置，且为放松状态，则 Y1 亮，Y12 亮。按下启动按钮后，机械手开始顺序动作：

- (1)、机械手下降：延时 1 秒后，Y1 灭，Y14 亮，到下限位时，按 X3，则 Y14 灭，Y2 亮。
- (2)、机械手夹紧：延时 3 秒后，Y12 灭，Y11 亮，
- (3)、机械手上升：Y2 灭，Y13 亮，到上限位后，按 X4，则 Y13 灭，Y3 亮，
- (4)、机械手右行：Y3 灭，Y15 亮，到右限位后，按 X5，则 Y15 灭，Y4 亮，
- (5)、机械手下降：Y4 灭，Y14 亮，到下限位后，按 X3，则 Y14 灭，Y6 亮，
- (6)、机械手放松：延时 3 秒后，Y11 灭，Y12 亮，
- (7)、机械手上升：Y6 灭，Y13 亮，到上限位后，按 X4，则 Y13 灭，Y5 亮，

(8)、机械手左移：Y5 灭，Y16 亮，到左限位后，按 X6，则 Y16 灭，Y1 亮，回到原始位置。

2、确定输入、输出端口、并编写程序。

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

表 12-1 输入输出接线图

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	启动	启动	Q0.0	Y1	原点
I0.1	停止	停止	Q0.1	Y2	放松下降到位
			Q0.2	Y3	夹紧上升到位
I0.4	上限位	上限开关	Q0.3	Y4	夹紧右移到位
I0.5	下限位	下限开关	Q0.4	Y5	放松上升到位
I0.6	左限位	左限开关	Q0.5	Y6	夹紧下降到位
I0.7	右限位	右限开关	Q0.6	Y11	夹紧指示
			Q0.7	Y12	放松指示
			Q1.0	Y13	上升中
			Q1.1	Y14	下降中
			Q1.2	Y15	右移中
			Q1.3	Y16	左移中
	COM	24V	1L		0V
1M		0V	2L		0V
			3L		0V

实验十三 四层电梯的 PLC 控制

一、实验目的

- 1、掌握 PLC 基本指令、功能指令的综合应用
- 2、掌握用 PLC 对电梯的基本控制程序的设计

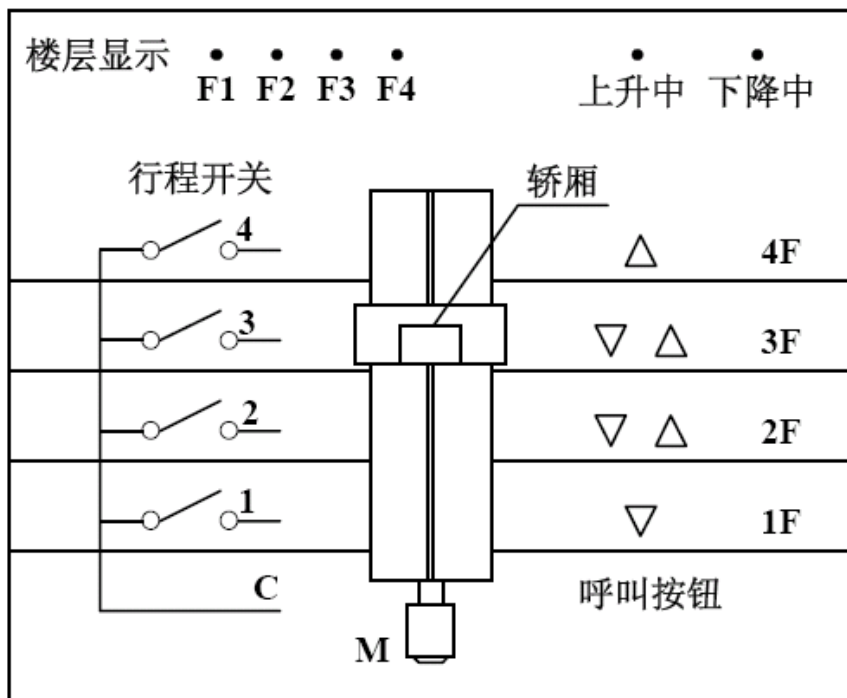
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、电梯控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

电梯模拟实验台结构示意图如下：



其动作要求如下：

(1) 电梯上行：

- ①当电梯停于 1 楼 1F 或 2F、3F 时，4F 呼叫，则上行到 4 楼，碰行程开关后停止。

- ②当电梯停于 1F 或 2F 时，3F 呼叫，则上行到 3F，碰行程开关后停止。
- ③当电梯停于 1F 时，2F 呼叫，则上行到 2F，碰行程开关后停止。
- ④当电梯停于 1F 时，2F、3F 同时呼叫，则上行到 2F 后，停 5 秒种，继续上行到 3F 后停止。
- ⑤当电梯停于 1F 时，3F、4F 同时呼叫，则上行到 3F 后，停 5 秒后，继续上行到 4F 后停止。
- ⑥当电梯停于 1F 时，2F、4F 同时呼叫，则上行到 2F 后，停 5 秒后，继续上行到 4F 后停止。
- ⑦当电梯停于 1F 时，2F、3F、4F 同时呼叫，则上行到 2F 后，停 5 秒后，继续上行到 3F 后，停 5 秒后，继续上行到 4F 后停止。
- ⑧当电梯停于 2F 时，3F、4F 同时呼叫，则上行到 3F 后，停 5 秒后，继续上行到 4F 后停止。

(2) 电梯下行

- ①当电梯停于 4F 或 3F、2F 时，1F 呼叫，则下行到 1F，碰行程开关后停止。
- ②当电梯停于 4F 或 3F 时，2F 呼叫，则下行到 2F，碰行程开关后停止。
- ③当电梯停于 4F 时，3F 呼叫，则下行到 3F，碰行程开关后停止。
- ④当电梯停于 4F 时，3F、2F 同时呼叫，则下行到 3F 后，停 5 秒种，继续下行到 2F 后停止。
- ⑤当电梯停于 4F 时，3F、1F 同时呼叫，则下行到 3F 后，停 5 秒后，继续下行到 1F 后停止。
- ⑥当电梯停于 4F 时，2F、1F 同时呼叫，则下行到 2F 后，停 5 秒后，继续下行到 1F 后停止。
- ⑦当电梯停于 4F 时，3F、2F、1F 同时呼叫，则下行到 3F 后，停 5 秒后，继续下行到 2F 后，停 5 秒后，继续下行到 1F 后停止。
- ⑧当电梯停于 3F 时，2F、1F 同时呼叫，则下行到 2F 后，停 5 秒后，继续下行到 1F 后停止。

(3) 各楼层运行时间应在 15 秒以内，否则认为有故障。

(4) 电梯停于某一层时，数码管应显示该层的楼层数。

(5) 电梯上、下行时，相应的标志灯应亮。

(6) 启动时，当电梯没有停在任何楼层，可自动向上或向下停靠。

(7) 设定关闭电梯的功能，当电梯完成最后一个呼叫请求后，延时 3 秒后，自动返回 1F。

确定输入、输出端口、并编写程序

输入地址：

一层楼行程开关	I0.0	一层上呼开关	I0.7	二层下呼开关	I0.4
二层楼行程开关	I0.1	二层上呼开关	I1.0	三层下呼开关	I0.5
三层楼行程开关	I0.2	三层上呼开关	I1.1	四层下呼开关	I0.6

四层楼行程开关 I0.3

关闭电梯开关 I1.4 向上停靠开关 I1.5 向下停靠开关 I1.6

输出地址:

电梯上行 Q0.0

电梯下行 Q0.1

一层指示 Q0.2

二层指示 Q0.3

三层指示 Q0.4

四层指示 Q0.5

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

表 13-1 输入输出接线图

输入			输出				
主机	实验模块	注释	主机	实验模块		注释	
I0.0	K1	一层行程开关	Q0.0	DJA		电机正转	
I0.1	K2	二层行程开关	Q0.1	DJB		电机反转	
I0.2	K3	三层行程开关	Q0.2	F1		一层指示	
I0.3	K4	四层行程开关	Q0.3	F2		二层指示	
I0.4	2 ↓	二层下呼	Q0.4	F3		三层指示	
I0.5	3 ↓	三层下呼	Q0.5	F4		四层指示	
I0.6	4 ↓	四层下呼					
I0.7	1 ↑	一层上呼	1L、2L			0V	
I1.0	2 ↑	二层上呼		实 验 区	24V	电 源 区	24V
I1.1	3 ↑	三层上呼			0V		0V
I1.4	复位	复位					
I1.5	上调	向上微调					
I1.6	下调	向下微调					
1M、2M		24V					
	COM	0V					

实验十四 刀库捷径方向选择控制

一、实验目的

- 1、掌握功能指令中数据处理指令的用法
- 2、掌握用加工中心刀库捷径方向选择控制的程序设计

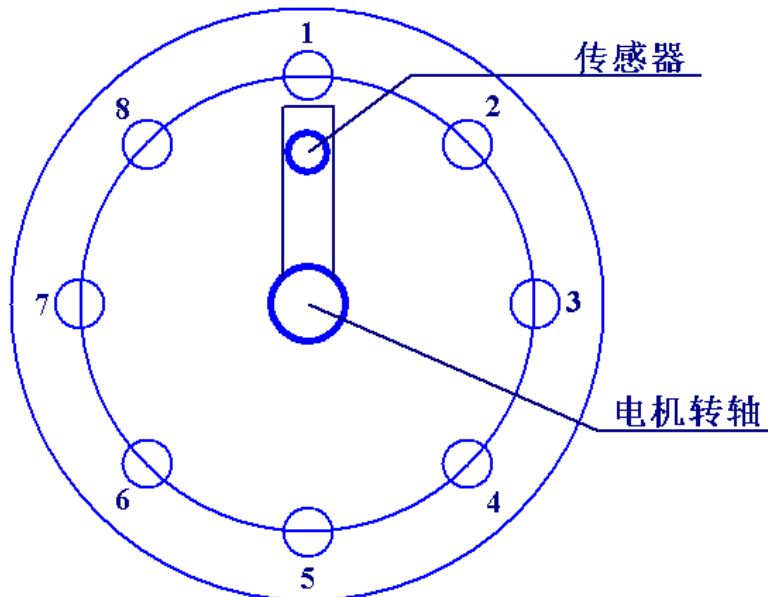
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、刀库控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

如图所示，为加工中心刀库回台式模拟装置，上面设有 8 把刀，每把刀有相应的刀号地址，分别为 1、2、……、8。刀库转动时，将由传感器测试其刀号位置，并由一只数码管显示当前刀号位置。



当按下启动开关后，如果传感器没有找到任何刀，电机将按顺时针方向自动转动，直到指传感器检测到某把刀，电机停转。这时数码管将显示该刀号地址。拨码开关用来选择刀号。选好刀号后，再按一下“送刀号”按钮，即可将所选择的刀号送入程序。程序将对所选择的刀号与当前刀号进行比较、运算，然后指挥电机按照离当前刀号最近的方向旋转（即最捷径方向，正转或反转），转到所选刀号位置后，电机停转。

- 2、确定输入、输出端口、并编写程序
- 3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。
- 4、调试程序，直至符合设计要求。

表 14-1 输入输出接线图

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	启动	启动按钮	Q0.0	DJA	电机正转
I0.1	送刀号	送刀号按钮	Q0.1	DJB	电机反转
I0.2	1	BCD 码 1			
I0.3	2	BCD 码 2			
I0.4	4	BCD 码 4	Q0.7	A	段码 A
I0.5	8	BCD 码 8	Q1.0	B	段码 B
			Q1.1	C	段码 C
I1.0	1	一号刀位置	Q2.0	D	段码 D
I1.1	2	二号刀位置	Q2.1	E	段码 E
I1.2	3	三号刀位置	Q2.2	F	段码 F
I1.3	4	四号刀位置	Q2.3	G	段码 G
I1.4	5	五号刀位置			
I1.5	6	六号刀位置	1L		24V
I2.0	7	七号刀位置	2L		24V
I2.1	8	八号刀位置	3L		24V
1M		24V		24V	24V
2M		24V		0V	0V
	COM	0V			

实验十五 物料混合控制实验

一、实验目的

- 1、掌握功能指令的用法
- 2、掌握物料混合控制程序的设计

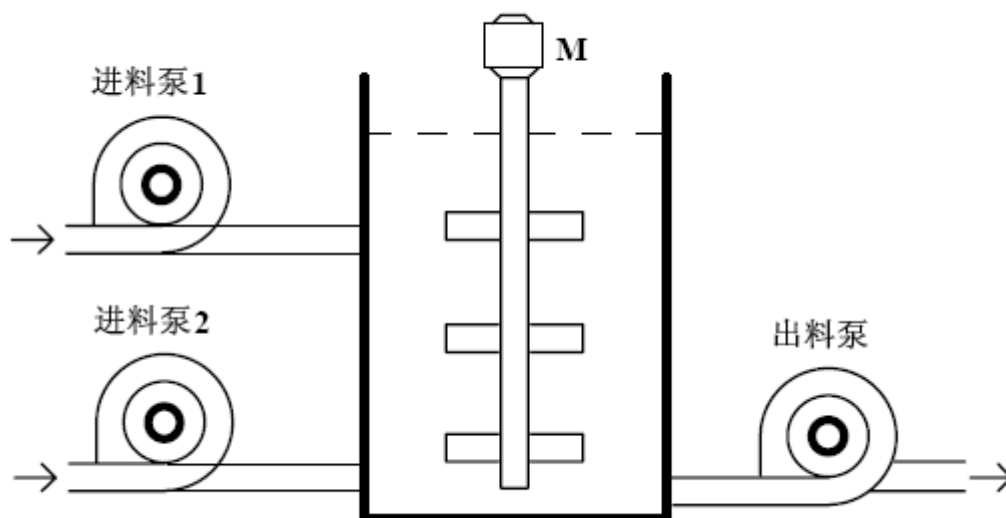
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、物料混合控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

设计一个物料混合控制程序，模块示意图如下：



物料混合系统示意图

要求先启动进料泵 1，进料完毕后，进料泵 1 关闭，再启动进料泵 2，进料完毕后，进料泵 2 关闭，搅拌机开始工作，先正转 5 分钟，然后反转 5 分钟，搅拌机停止工作，打开出料阀出料。待料出完后，重复上述过程。当按下停止按钮后，如果正在一个循环中，那么

等待当前循环结束，即出料完毕后，程序停止运行。

2、确定输入、输出端口、并编写程序

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

表 11-1 输入输出接线图

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	启动	启动	Q0.0	进料 1	Q0.0
I0.1	停止	停止	Q0.1	进料 2	Q0.1
			Q0.2	出料	Q0.2
1M	24V		1L	24V	
	0V ↔ COM				
			Q0.4	正转	继电器接法参照电 机左边的接线图
			Q0.5	反转	
			2L	0V	

实验十六 水塔水位控制

一、实验目的

- 1、掌握功能指令的用法
- 2、掌握水塔水位控制程序的设计

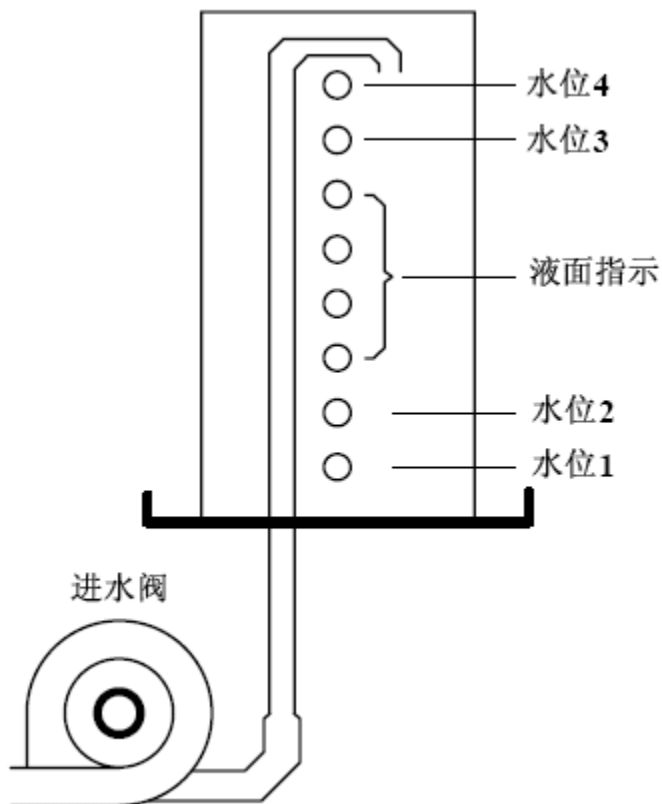
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、水塔水位控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

设计一个水塔水位控制程序。模块示意图如下：



水塔水位控制模块示意图

控制面板上，水箱位置中的四只绿色指示灯分别指示高水位液面传感器，及低水位传感器的工作状态，按钮水位 1、水位 2、水位 3、水位 4 分别用于模拟高、低水位传感器的工作状态。水位指示灯亮，表示当前水位低于指示水位。

中间的四个绿灯（3、4、5、6）用于指示液面状态。从下往上依次循环点亮，表示水面在上升。

低水位指示灯亮（高水位指示灯也亮），表示水箱缺水。这时进水阀开（指示灯亮），水箱开始进水；按下水位 1 按钮，水位 1 指示灯灭（高水位灯仍亮）；按下水位 2 按钮，水位 2 指示灯灭（高水位灯仍亮）；按下水位 3 按钮，水位 3 指示灯灭（高水位灯仍亮），这时中间四个绿灯点亮速度变慢，表示水面在缓慢上升；按下水位 4 按钮，水位 4 指示灯灭，中间四个绿灯熄灭，表示水箱满。进水阀关闭（指示灯灭）。

2、确定输入、输出端口、并编写程序

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	水位 1	水位 1 按钮	Q0.0	SP	蜂鸣器
I0.1	水位 2	水位 2 按钮	Q0.1	DJ	电机工作指示灯
I0.2	水位 3	水位 3 按钮	Q0.2	指示灯 1	水位 1
I0.3	水位 4	水位 4 按钮	Q0.3	指示灯 2	水位 2
I0.4	启动	启动按钮	1L	24V	
I0.5	停止	停止按钮			
1M	24V		Q0.4	指示灯 3	3、4、5、6 组合指示 液面变化
	0V←→COM		Q0.5	指示灯 4	
			Q0.6	指示灯 5	
			Q0.7	指示灯 6	
			2L	24V	
			Q1.0	指示灯 7	水位 3
			Q1.1	指示灯 8	水位 4
			3L	24V	

实验十七 邮件分拣控制

一、实验目的

- 1、掌握功能指令的用法
- 2、掌握邮件分拣控制程序的设计

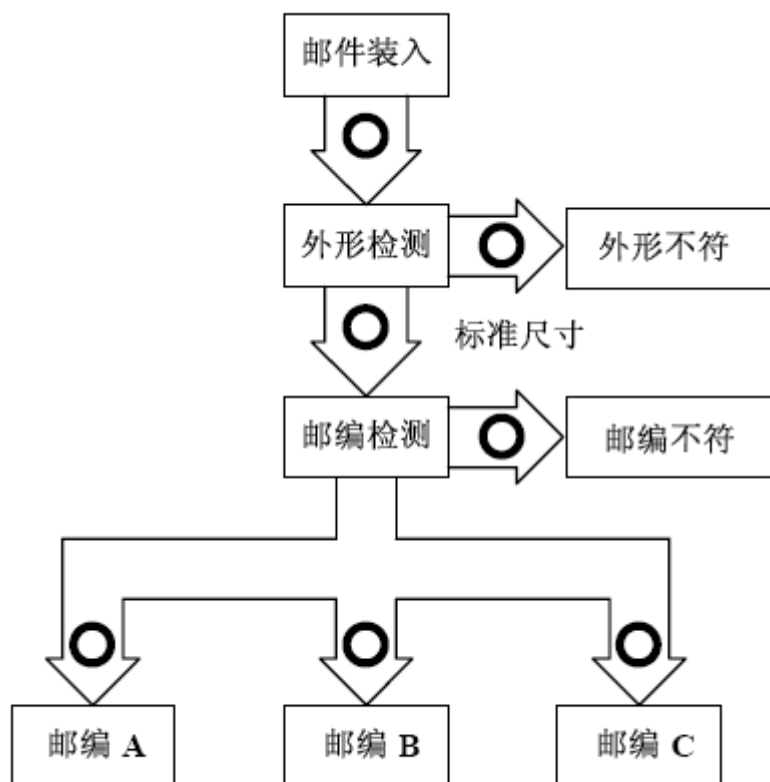
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、邮件分拣控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

设计一个邮件分拣控制程序。模块示意图如下：



邮件分拣系统示意图

按下启动按钮后，邮件开始进入流水线，指示灯 1 亮，延时 2 秒后，进入外形检测区，指示灯 2 亮，延时 2 秒后进入邮编检测区，指示灯 3 亮，延时 2 秒后进入邮编分档区，该区指示灯 4、5、6 闪烁点亮。

外形不符、邮编不符按钮：用来模拟非标准尺寸信号及无法识别信号。当按住按钮时，表示该信号有效，流水线暂停，等待处理，当放开按钮时，表示信号无效，流水线继续运行。邮编 A、邮编 B、邮编 C 三个按钮指定处理相应邮编的信号，当按住某个按钮时，邮编分档区中只闪烁点亮邮编与该按钮相对应的邮件。

2、确定输入、输出端口、并编写程序

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	启动	启动	Q0.0	1	邮件输入
I0.1	停止	停止	Q0.1	2	外形检测
I0.2	邮编 A	邮编 A	Q0.2	3	邮编检测
I0.3	邮编 B	邮编 B	Q0.3	4	邮编 A
I0.4	邮编 C	邮编 C	Q0.4	5	邮编 B
I0.5	外形不符	非标准尺寸	Q0.5	6	邮编 C
I0.6	邮编不符	无法识别	Q0.6	7	邮编不符
			Q0.7	8	外形不符
1M		24V	1L		24V
			2L		24V

实验十八 四级传送带的控制

一、实验目的

- 1、掌握功能指令的用法
- 2、掌握四级传送带控制程序的设计

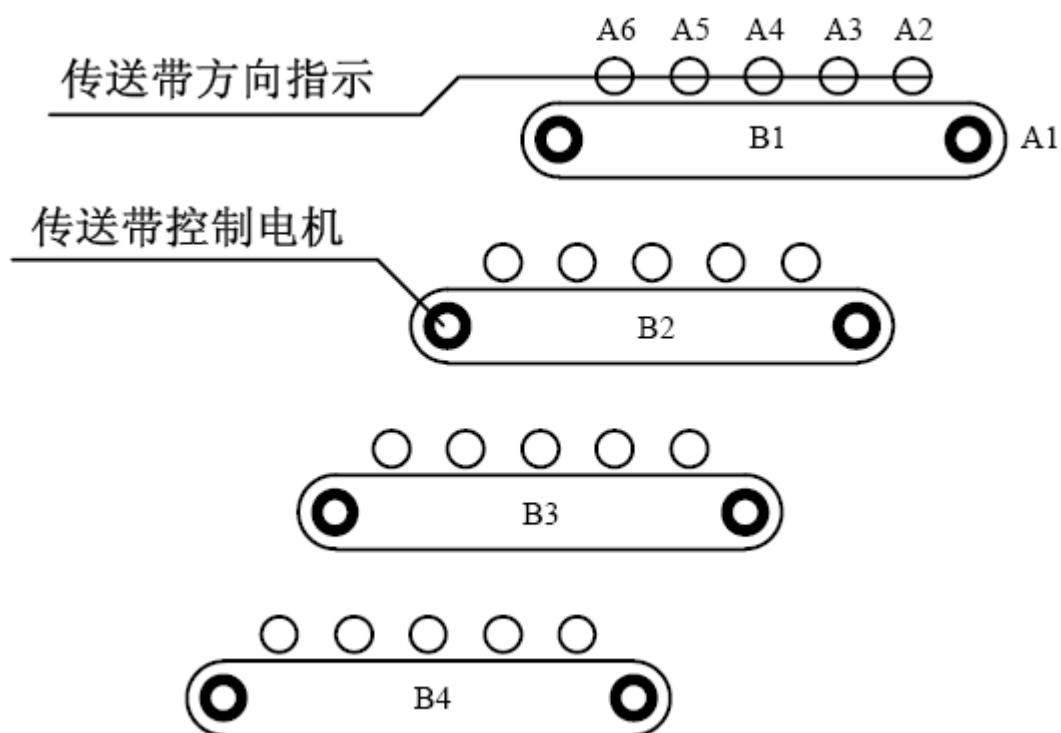
二、实验器材

- 1、PC 机一台
- 2、PLC 实验箱一台
- 3、四级传送带控制模块
- 4、编程电缆一根
- 5、导线若干

三、实验内容及步骤

1、设计要求

设计一个四级传送带控制程序。 模块示意图如下：



四级传送带示意图

实验程序的工作流程：

控制模块中从上往下依次为一、二、三、四级传送带。

当按下启动按钮后，进料指示灯亮，四级传送带开始工作，绿灯点亮，红灯开始从右往左循环点亮；延时 5 秒后，三级传送带开始工作；延时 5 秒后，二级传送带开始工作；延时 5 秒后，一级传送带开始工作。

开关 K1、K2、K3、K4，分别用来模拟一、二、三、四级传送带的故障信号。

当 K1 闭合，一级传送带出现故障，一级传送带立即停止，延时 5 秒后，二级传送带停止，延时 5 秒后，三级传送带停止，延时 5 秒后，四级传送带停止。

当 K2 闭合，二级传送带出现故障，一级、二级传送带立即停止，延时 5 秒后，三级传送带停止，延时 5 秒后，四级传送带停止。

当 K3 闭合，三级传送带出现故障，一级、二级、三级传送带立即停止，延时 5 秒后，四级传送带停止。

当 K4 闭合，四级传送带出现故障，一级、二级、三级、四级传送带立即停止。

当故障排除后，即开关断开，四级传送带即同时开始工作。

当按下停止按钮时，一级传送带立即停止，延时 5 秒后，二级传送带停止，延时 5 秒后，三级传送带停止，延时 5 秒后，四级传送带停止。

2、确定输入、输出端口、并编写程序

3、编译程序，无误后下载至 PLC 主机的存储器中，并运行程序。

4、调试程序，直至符合设计要求。

输入			输出		
主机	实验模块	注释	主机	实验模块	注释
I0.0	启动	启动	Q0.0	A1	
I001	停止	停止			
			Q0.1	A2	皮带传送 方向指示
			Q0.2	A3	
I0.3	K1	一级传送带故障模拟	Q0.3	A4	
I0.4	K2	二级传送带故障模拟	Q0.4	A5	
I0.5	K3	三级传送带故障模拟	Q0.5	A6	
I0.6	K4	四级传送带故障模拟			
			Q1.1	B1	一级传送带电机
			Q1.2	B2	二级传送带电机
			Q1.3	B3	三级传送带电机
			Q1.4	B4	四级传送带电机
	COM	24V	1L		24V
			2L		24V
1M		0	3L		0V