

# 第二单元 PLC 的基本应用

## 三相异步电动机的单向点动与自锁混合控制

### 【项目目标】

用 S7-200 PLC 实现三相异步电动机的单向点动、自锁混合控制，运用 STEP7-Micro/WIN32 软件对单向运行控制系统进行联机调试。

### 【学习目标】

- (1) 掌握触点指令与线圈指令的用法。
- (2) 掌握 PLC 的接线方法。
- (3) 掌握继电器电路转换法建立控制系统的方法。
- (4) 熟练运用 S7-200 PLC 的编程软件 STEP7-Micro/WIN32 进行程序的下载与调试。
- (5) 学会利用 PLC 仿真软件进行 PLC 程序仿真调试。

### 【相关知识】

## 一、标准触点与输出指令

触点指令与线圈指令属于位逻辑指令。位逻辑指令在语句表语言中是指对位存储单元的简单逻辑运算，在梯形图中是指对触点的简单连接和对标准线圈的输出。

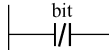
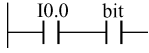
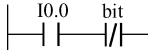
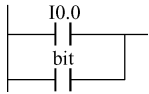
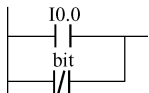
PLC 的程序可写成梯形图的形式，从继电器控制电路的角度进行理解；可写成语句表的形式，从计算机语言的角度进行理解；当然还有其他形式，本书主要使用以上两种形式。

一般来说，语句表语言更适合于熟悉可编程序控制器和逻辑编程方面有经验的编程人员。用这种语言可以编写出用梯形图或功能框图无法实现的程序。

### 1. 标准触点指令（如表 6-1 所示）

表 6-1 S7-200 系列的标准触点指令

指令名称	语句表形式	梯形图形式	功能	操作数
取	LD bit		读入逻辑行或电路块的第一个常开接点	Bit: 为 I, Q, M, SM, T, C, V, S 型

指令名称	语句表形式	梯形图形式	功能	操作数
取反	LDN bit		读入逻辑行或电路块的第一个常闭接点	Bit: 为 I, Q, M, SM, T, C, V, S 型
与	A bit		串联一个常开接点	
与非	AN bit		串联一个常闭接点	
或	O bit		并联一个常开接点	
或非	ON bit		并联一个常闭接点	

对于表 6-1 中的 bit, 它是 PLC 内部的编程元件。如以梯形图的形式编程, 把它理解为是 PLC 内部的继电器。当地址是 bit 的继电器的线圈有电时, 继电器常开触点闭合、常闭触点断开。如以语句表的形式编程, 把它理解为 PLC 内部的存储位。当地址是 bit 的存储位的值为 1 时, 相当于相应的继电器线圈有电, 为 0 相当于无电。PLC 在本质上是按语句表的逻辑来运行程序的, 执行过程中要用堆栈。可编程序控制器中的堆栈与计算机中的堆栈结构相同, 堆栈是一组能够存储和取出数据的暂时存储单元。堆栈的存取特点是“后进先出”, S7-200 PLC 的主机逻辑堆栈结构如表 6-2 所示。

表 6-2 主机逻辑堆栈结构

S0	STACK0	第一个堆栈 (即栈顶)
S1	STACK1	第二个堆栈
S2	STACK2	第三个堆栈
S3	STACK3	第四个堆栈
S4	STACK4	第五个堆栈
S5	STACK5	第六个堆栈
S6	STACK6	第七个堆栈
S7	STACK7	第八个堆栈
S8	STACK8	第九个堆栈

例如, 程序 LD I0.1, A I0.2, 先执行 LD I0.1, 再执行 A I0.2。若 I0.1 的值为 1, 执行 LD I0.1, 就是将 I0.1 的值装入栈顶, 指令的执行对逻辑堆栈的影响如表 6-3 所示。

表 6-3 指令 LD I0.1 的执行

名称	执行前	执行后	说明
STACK0	S0	1	新值 I0.1=1 装入堆栈, 原值依次下移一个单元, S8 丢失
STACK1	S1	S0	
STACK2	S2	S1	
STACK3	S3	S2	
STACK4	S4	S3	
STACK5	S5	S4	
STACK6	S6	S5	
STACK7	S7	S6	
STACK8	S8	S7	

表 6-4 指令 A I0.2 的执行

名称	执行前	执行后	说明
STACK0	1	0	将 I0.2 的值 0 与原栈顶值 1 做与运算, 指令的执行结果 0 代替原栈顶的值
STACK1	S1	S1	
STACK2	S2	S2	
STACK3	S3	S3	
STACK4	S4	S4	
STACK5	S5	S5	
STACK6	S6	S6	
STACK7	S7	S7	
STACK8	S8	S8	

## 2. 输出指令（如表 6-5 所示）

表 6-5 S7-200 系列的输出指令

指令名称	语句表形式	梯形图形式	功能	操作数
输出	= bit	$\text{---} \left( \begin{array}{c} \text{bit} \\ \text{---} \end{array} \right)$	输出逻辑行的运算结果	Bit: Q, M, SM, T, C, V, S (立即指令时只能为 Q)
置位	S bit, N	$\text{---} \left( \begin{array}{c} \text{bit} \\ \text{S} \\ \text{---} \\ \text{N} \end{array} \right)$	置继电器状态为接通	Bit: Q, M, SM, V, S (立即指令时只能为 Q)
复位	R bit, N	$\text{---} \left( \begin{array}{c} \text{bit} \\ \text{R} \\ \text{---} \\ \text{N} \end{array} \right)$	使继电器复位为断开	

### 1) 输出

写成 ( ) 或 = ，例如，程序 LD I0.0, = Q0.1 当 I0.0 为 1 时，Q0.1 线圈有电，也就是输出映像寄存器 Q0.1=1。

### 2) 置位和复位

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。

#### (1) S——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。

#### (2) R——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。

程序及其结果如图 6-1 所示。PLC 循环扫描执行程序，若本次执行这段程序时，I0.1 常开触点闭合，则指令 S Q0.3,1 得以执行，Q0.3 的值置位为 1；若下次执行程序时，I0.1 常开触点断开，则指令 S Q0.3,1 不执行，那么 I0.1 对 Q0.3 无影响，Q0.3 的值仍为 1，所以置位指令有自锁（保持）功能，复位指令也一样。

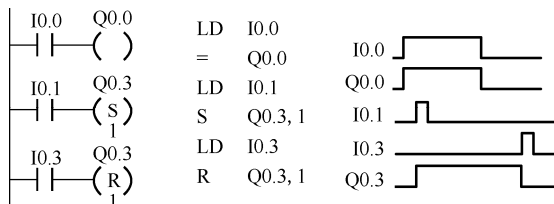


图 6-1 程序及其结果

## 3. 立即指令（如表 6-6 所示）

表 6-6 S7-200 PLC 的立即指令

指令名称	语句表形式	梯形图形式	功能	操作数
取	LDI bit	$\text{---} \left  \begin{array}{c} \text{bit} \\ \text{I} \end{array} \right  \text{---}$	读入逻辑行或电路块的第一个常开接点	Bit: 只能为 I 型，如 LDI I0.0
取反	LDNI bit	$\text{---} \left  \begin{array}{c} \text{bit} \\ \text{/I} \end{array} \right  \text{---}$	读入逻辑行或电路块的第一个常闭接点	
与	AI bit	$\text{---} \left  \begin{array}{c} \text{I0.0} \\ \text{---} \end{array} \right  \left  \begin{array}{c} \text{bit} \\ \text{I} \end{array} \right  \text{---}$	串联一个常开接点	

指令名称	语句表形式	梯形图形式	功能	操作数
与非	ANI bit		串联一个常闭接点	Bit: 只能为 I 型, 如 LDI I0.0
或	OI bit		并联一个常开接点	
或非	ONI bit		并联一个常闭接点	
输出	=I bit	$\text{—}(\text{ I })$	输出逻辑行的运算结果	Bit: 只能为 Q 型, 如 SI Q0.5,1
置位	SI bit, N		置继电器状态为接通	
复位	RI bit, N	$\text{—}(\text{ RI })$	使继电器复位为断开	

### 1) 立即触点指令

立即触点指令只能用于输入触点，执行立即读入物理输入点的值，根据该值决定触点的接通/断开状态，但是并不更新该物理输入点对应的映像寄存器。

### 2) =I——立即输出指令

执行立即输出指令时，则将结果同时立即复制到物理输出点和相应的输出映像寄存器。例如，程序 LD I0.0, =I Q0.1 当 I0.0 为 1 时，Q0.1 线圈有电，也就是输出映像寄存器 Q0.1=1，同时在 Q0.1 的输出端立即产生有效输出。

### 3) SI——立即置位指令

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。

**注意：**bit 只能是 Q 类型。

### 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。

**注意：**bit 只能是 Q 类型，同 SI 指令。

立即 I/O 指令是直接访问物理输入/输出点的，比一般指令访问输入/输出映像寄存器占用 CPU 时间要长，因而不能盲目地使用立即指令，否则，会加长扫描周期时间，反而对系统造成不利影响。

## 4. 取反及跳变指令 (如表 6-7 所示)

### 1) 取反 (NOT)

取反触点将它左边电路的逻辑运算结果取反，运算结果若为 1 则变为 0，为 0 则变为 1。由于运算结果在栈顶，相当于对栈顶值取反。

从继电器 (或能流) 的角度看，能流到达该触点时即停止，若能流未到达该触点，该触

点给右侧供给能流。

表 6-7 S7-200 系列的取反及跳变指令

指令名称	语句表形式	梯形图形式	功能
取反	NOT		能流到达取非触点时，能流就停止；能流未到达取非触点时，能流就通过
正跳变指令	EU		检测到每一次正跳变信号后，让能流通过一个扫描周期的时间
负跳变指令	ED		检测到每一次负跳变信号后，让能流通过一个扫描周期的时间

### 2) 正跳变指令

正跳变触点-|P|-检测到左边的逻辑运算结果的一次正跳变（触点的左边输入信号由 0 变为 1），触点接通一个扫描周期。

### 3) 负跳变指令

负跳变触点-|N|-检测到左边的逻辑运算结果的一次负跳变（触点的左边输入信号由 1 变为 0），触点接通一个扫描周期。

例如，程序及结果如图 6-2 所示。

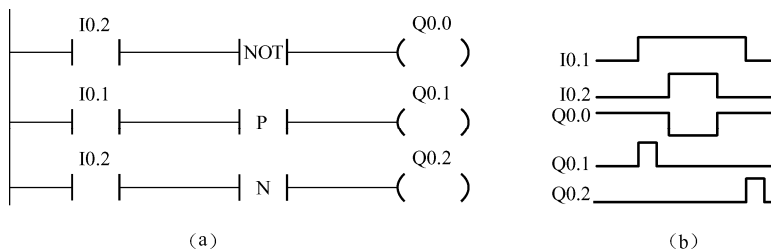


图 6-2 程序及结果

例如，某台设备有两台电动机 M1 和 M2，其交流接触器分别连接 PLC 的输出继电器 Q0.1 和 Q0.2，总启动按钮使用常开按钮，接输入继电器 I0.0 端口，总停止按钮使用常闭按钮，接输入继电器 I0.1 端口。为了减小两台电动机同时启动对供电电路的影响，让 M2 稍微延迟片刻启动。控制要求是：按下启动按钮，M1 立即启动，松开启动按钮时，M2 才启动；按下停止按钮，M1、M2 同时停止。电动机控制程序如图 6-3 所示。电动机控制程序的时序图如图 6-4 所示。

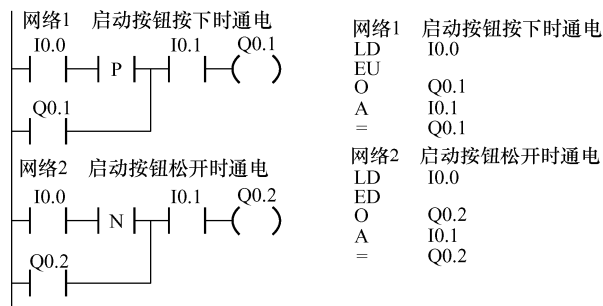


图 6-3 电动机控制程序

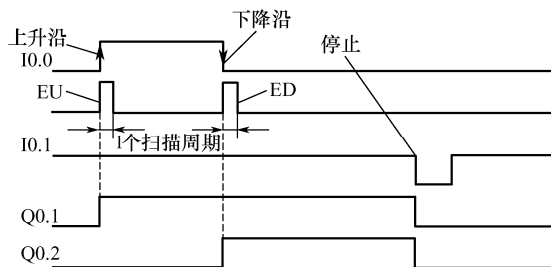


图 6-4 电动机控制程序的时序图

## 二、复杂逻辑指令

复杂逻辑指令没有梯形图形式，只有语句表形式。一般在梯形图程序里边体现不出来，但是把梯形图程序写成语句表形式时就要用到。由于 PLC 本质上是按语句表程序的逻辑来运行的，所以有必要理解复杂逻辑指令的特点，掌握其用法。

### 1. 栈装载与指令

ALD，栈装载与指令（与块）。在梯形图中用于将电路块进行串联连接。

指令 ALD 的执行如表 6-8 所示。

### 2. 栈装载或指令

OLD，栈装载或指令（或块），串联电路块（两个以上触点串联形成的支路称为串联电路块）的并联连接。OLD 指令不需要地址，它相当于要并联的两块电路右端的一段垂直连线。

指令 OLD 的执行如表 6-9 所示。

表 6-8 指令 ALD 的执行

名称	执行前	执行后	说明
STACK0	1	0	假设执行前，S0=1，I=0。本指令对堆栈中的第一层 S0 和第二层 S1 的值进行逻辑与运算，结果放回栈顶。即： $S0=S0 \times S1=1 \times 0=0$ 。执行完本指令后堆栈串行上移 1 格，深度减 1
STACK1	0	S2	
STACK2	S2	S3	
STACK3	S3	S4	
STACK4	S4	S5	
STACK5	S5	S6	
STACK6	S6	S7	
STACK7	S7	S8	
STACK8	S8	X	

表 6-9 指令 OLD 的执行

名称	执行前	执行后	说明
STACK0	1	1	假设执行前，S0=1，S1=0。本指令对堆栈中的第一层 S0 和第二层 S1 的值进行逻辑或运算，结果放回栈顶。即： $S0=S0+S1=1+0=1$ 。执行完本指令后堆栈串行上移 1 格，深度减 1
STACK1	0	S2	
STACK2	S2	S3	
STACK3	S3	S4	
STACK4	S4	S5	
STACK5	S5	S6	
STACK6	S6	S7	
STACK7	S7	S8	
STACK8	S8	X	

### 3. 逻辑推入栈指令

LPS，逻辑推入栈指令（分支或主控指令）。在梯形图中的分支结构中，用于生成一条新的母线，左侧为主控逻辑块时，第一个完整的逻辑行从此处开始。

注意：使用 LPS 指令时，本指令为分支的开始，以后必须有分支结束指令 LPP。即 LPS 与 LPP 指令必须成对出现。

指令 LPS 的执行如表 6-10 所示。

#### 4. 逻辑弹出栈指令

LPP, 逻辑弹出栈指令（分支结束或主控复位指令）。在梯形图中的分支结构中，用于将 LPS 指令生成一条新的母线进行恢复。

注意：使用 LPP 指令时，必须出现在 LPS 的后面，与 LPS 成对出现。

指令 LPP 的执行如表 6-11 所示。

表 6-10 指令 LPS 的执行

名称	执行前	执行后	说明
STACK0	1	1	假设执行前，S0=1。本指令对堆栈中的栈顶 S0 进行复制，并将这个复制值由栈顶压入堆栈。执行完本指令后堆栈串行下移 1 格，深度加 1，原来的栈底 S8 内容将自动丢失
STACK1	S1	1	
STACK2	S2	S1	
STACK3	S3	S2	
STACK4	S4	S3	
STACK5	S5	S4	
STACK6	S6	S5	
STACK7	S7	S6	
STACK8	S8	S7	

表 6-11 指令 LPP 的执行

名称	执行前	执行后	说明
STACK0	0	1	假设执行前，S0=0，S1=1。本指令将堆栈的栈顶 S0 弹出，则第二层 S1 的值上升进入栈顶，即：S0=S1=1。执行完本指令后堆栈串行上移 1 格，深度减 1，栈底 S8 内容将生成一个随机值
STACK1	1	S1	
STACK2	S1	S2	
STACK3	S2	S3	
STACK4	S3	S4	
STACK5	S4	S5	
STACK6	S5	S6	
STACK7	S6	S7	
STACK8	S7	X	

#### 5. 逻辑读栈指令

LRD, 逻辑读栈指令。在梯形图中的分支结构中，当左侧为主控逻辑块时，从第二个逻辑块开始，后边更多的是从逻辑块。

指令 LRD 的执行如表 6-12 所示。

#### 6. 装入堆栈指令

LDS——装入堆栈指令。本指令编程时较少使用。

指令格式：LDSn（n 为 0~8 的整数）

例如，LDS 4。

指令 LDS 的执行如表 6-13 所示。

表 6-12 指令 LRD 的执行

名称	执行前	执行后	说明
STACK0	1	0	假设执行前，S0=1，S1=0。本指令将堆栈中的第二层 S1 的值进行复制，然后将这个复制值放入栈顶 S0，本指令不对堆栈进行压入和弹出操作，即：S0=S1=0。执行完本指令后堆栈不串行上移或下移，除栈顶值之外，其他部分的值不变
STACK1	0	0	
STACK2	S2	S2	
STACK3	S3	S3	
STACK4	S4	S4	
STACK5	S5	S5	
STACK6	S6	S6	
STACK7	S7	S7	
STACK8	S8	S8	

表 6-13 指令 LDS 的执行

名称	执行前	执行后	说明
STACK0	1	0	假设执行前，S0=1，S4=0。本指令将堆栈中的第五层 S4 的值进行复制，并将这个复制值由栈顶压入堆栈。即：S0=S4=0。执行完本指令后堆栈串行下移 1 格，深度加 1，原来的栈底 S8 内容将自动丢失
STACK1	S1	1	
STACK2	S2	S1	
STACK3	S3	S2	
STACK4	0	S3	
STACK5	S5	0	
STACK6	S6	S5	
STACK7	S7	S6	
STACK8	S8	S7	

应用举例：图 6-5 是复杂逻辑指令在实际应用中的一段程序的梯形图。

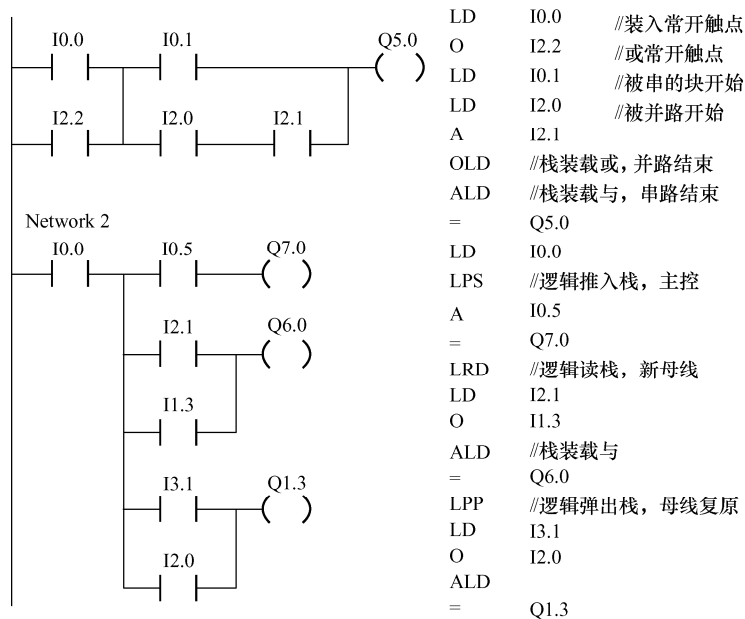


图 6-5 复杂逻辑指令应用

### 三、PLC 程序的继电器电路转换法

PLC 程序设计常用的方法主要有继电器控制电路转换为梯形图法、经验设计法、顺序控制设计法等。下面介绍 PLC 程序的继电器电路转换法。

梯形图与继电器电路图极为相似，如果用 PLC 改造继电器控制系统，根据继电器电路设计梯形图是一条捷径。这是因为原有的继电器控制系统经过长期的使用和考验，已经被证明能完成系统要求的控制功能，而继电器电路图又与梯形图有很多相似之处，因此可以将继电器电路图“翻译”成梯形图，即用 PLC 的外部硬件接线图和梯形图软件来实现继电器系统的功能。这就是 PLC 程序的继电器电路转换法。

这种设计方法一般不需要改动控制面板，保持了系统原有的外部特性，操作人员不用改变长期养成的操作习惯。

在分析 PLC 控制系统的功能时，可以将它想象成一个继电器控制系统中的控制箱，其外部接线图描述了这个控制箱的外部接线，梯形图是这个控制箱的内部“线路图”，梯形图中的输入位（I）和输出位（Q）是这个控制箱与外部世界联系的“输入、输出继电器”，这样就可以用分析继电器电路图的方法来分析 PLC 控制系统。在分析时可以将梯形图中输入位的触点想象成对应的外部输入器件的触点，将输出位的线圈想象成对应的外部负载的线圈。外部负载的线圈除了受梯形图的控制外，还可能受外部触点的控制。

继电器电路图中的交流接触器和电磁阀等执行机构如果用 PLC 的输出位来控制，它们的线圈接在 PLC 的输出端。按钮、控制开关、限位开关、光电开关等用来给 PLC 提供控制命令和反馈信号，它们的触点接在 PLC 的输入端。继电器电路图中的中间继电器和时间继电器的功能用 PLC 内部的存储器位（M）和定时器（T）来完成，它们与 PLC 的输入位、输出位无关。



## 1. 继电器电路图转换为功能相同的 PLC 的外部接线图和梯形图的步骤

(1) 了解和熟悉被控设备的工艺过程和机械的动作情况。

(2) 确定 PLC 的输入信号和输出负载，画出 PLC 外部接线图。

(3) 确定与继电器电路图的中间继电器、时间继电器对应的梯形图中的存储器位和定时器的地址。

(4) 根据上述对应关系，在继电器电路图的基础上改画出梯形图。

(5) 优化梯形图。

## 2. 根据继电器电路图设计 PLC 外部接线图和梯形图时应注意的问题

(1) 正确确定 PLC 的输入信号和输出负载。

热继电器 FR 的触点可以放在输入回路，如果是需要手动复位的热继电器，它的常闭触点也可以放在输出回路，与相应的接触器的线圈串联。

时间继电器 KT 的功能用 PLC 内部定时器实现，它们的线圈不应在输出回路出现。

(2) 输入触点类型的选择。

应尽可能用常开触点提供输入信号，但有的信号使用常闭触点可能更可靠一些，如果使用极限开关的常开触点来防止机械设备冲出限定的区域，常开触点接触不好时起不到保护作用，使用常闭触点则更安全一些。

(3) 硬件互锁电路。

例如，将电动机的正转、反转接触器的常闭触点串接在对方的线圈回路内。

(4) 梯形图结构的选择。

在梯形图中，为了简化电路和分离各线圈的控制电路，可以在梯形图中增加类似“中间继电器”的存储器位。将继电器电路图“翻译”成梯形图后，进一步将梯形图加以优化或简化。

(5) 应考虑 PLC 的工作特点。

继电器电路可以并行工作，而 PLC 的 CPU 是串行工作，即 CPU 同时只能处理 1 条指令，而且 PLC 在处理指令时有先后次序。

(6) 时间继电器瞬动触点的处理。

时间继电器的瞬动触点在时间继电器的线圈通电的瞬间动作，它们的触点符号上无表示延时的圆弧。PLC 的定时器触点虽然与普通触点的符号相同，但它们是延时动作的。

在梯形图中，可以在时间继电器对应的定时器功能块的两端并联存储器位 M 的线圈，用 M 的触点模拟时间继电器的瞬动触点。

(7) 尽量减少 PLC 的输入信号和输出信号。

减少输入信号和输出信号的点数是降低硬件费用的主要措施。

如具有手动复位功能的热继电器的常闭触点可采用与继电器电路相同的方法，将它放在 PLC 输出回路，与相应接触器的线圈串联，而不是将它们作为 PLC 的输入信号，这样可节约 PLC 的一个输入点。

(8) 梯形图的优化设计。

① 在触点的串联电路中，单个触点应放在右边。

② 在触点的并联电路中，单个触点应放在下面。

③ 在线圈的并联电路中，单个线圈应放在线圈与触点串联电路的上面。

(9) 外部负载的额定电压。

PLC 的继电器输出模块和双向晶闸管输出模块只能驱动额定电压 AC 220V 的负载，如原

有的交流接触器线圈电压为 380V，应将线圈换成 220V 的，或设置外部中间继电器。

## 【项目分析】

对于三相异步电动机的单向点动、自锁混合控制，有非常成熟的继电器控制电路，采用继电器电路转换法是一个不错的选择。

电动机主电路如图 6-6 所示，电动机继电器控制电路如图 6-7 所示，可实现电动机单向点动、自锁混合控制。

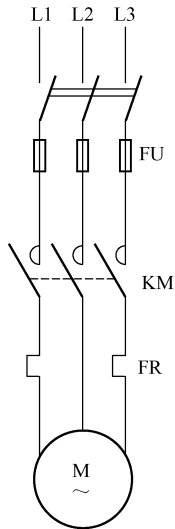


图 6-6 电动机主电路

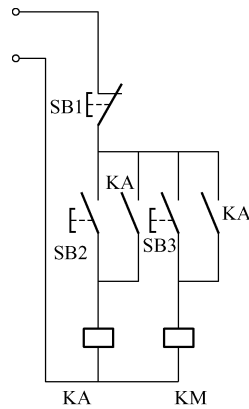


图 6-7 电动机继电器控制电路

继电器单向点动、自锁混合控制电路分析如图 6-8 所示。

按下按钮 SB3，交流接触器 KM 线圈有电、常开触点闭合，三相交流电动机运转；松开按钮 SB3，交流接触器 KM 线圈没电、常开触点断开，三相交流电动机停止。按下按钮 SB2，中间继电器 KA 线圈有电、常开触点闭合并保持自锁，KM 线圈保持有电；松开按钮 SB2，由于 KA 自锁，常开触点一直闭合，KM 线圈保持有电不变，电动机一直保持运转。按下按钮 SB1，KM 线圈没电、常开触点断开，三相交流电动机停止。

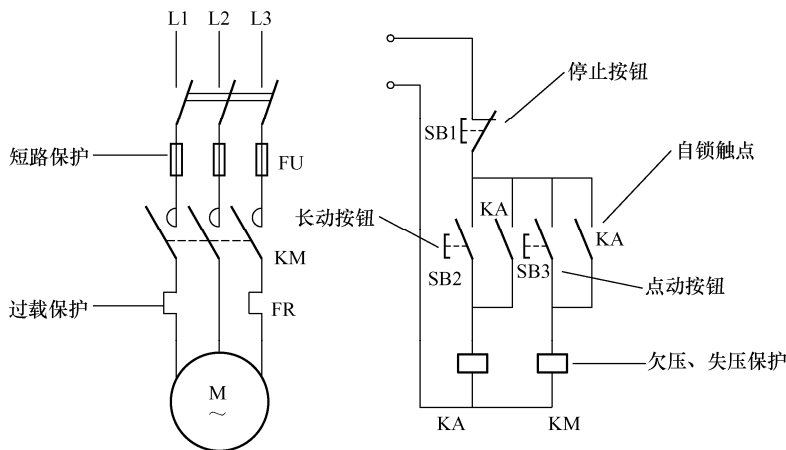


图 6-8 继电器单向点动、自锁混合控制电路分析

采用 PLC 控制后，主电路沿用图 6-6 所示的电路，把图 6-7 所示的继电器控制电路“翻译”成梯形图，即用 PLC 的外部硬件接线图和梯形图软件来实现继电器控制电路的功能。

## 【项目实施】

### 1. 分配 I/O 地址

按钮 SB1、SB2、SB3 给 PLC 提供控制命令，它们的触点接在 PLC 的输入端。用 PLC 的输出位 Q0.0 来控制 KM，KM 的线圈接在 PLC 的输出端。I/O 点分配如表 6-14 所示。

表 6-14 I/O 点分配

元件名称	形式	I/O 点	说明
SB1	常闭按钮	I0.0	停止按钮
SB2	常开按钮	I0.1	长动按钮
SB3	常开按钮	I0.2	点动按钮
KM	交流接触器	Q0.0	触点容量扩大

### 2. 画出 PLC 外部接线图

根据 I/O 类型及点数，CPU221 有 6 个数字量输入点、4 个数字量输出点，由于驱动交流接触器，所以选继电器输出型的 CPU221，画出如图 6-9 所示的 PLC 外部接线图，主电路不变。

### 3. 确定中间继电器、时间继电器的替代者

继电器电路图的中间继电器 KA 用 PLC 中的存储器位 M0.0 来代替。

### 4. 设计梯形图

将继电器电路图“翻译”成梯形图，如图 6-10 所示。

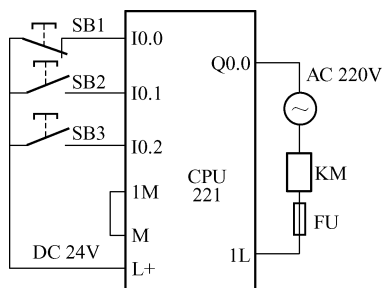


图 6-9 PLC 外部接线图

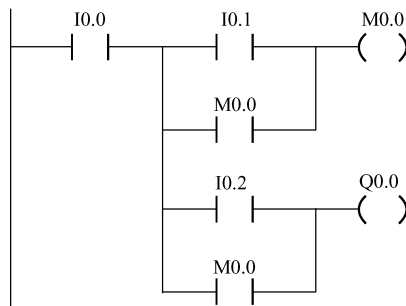


图 6-10 梯形图

### 5. 优化梯形图

图 6-10 的梯形图写成语句表形式如下：

```

LD    I0.0
LPS
LD    I0.1
O     M0.0
ALD
=     M0.0
LPP
LD    I0.2
O     M0.0
ALD
=     Q0.0
    
```

PLC 梯形图的优化规则是：有几个串联电路相并联时，应将串联触点多的回路放在上方；有几个并联电路相串联时，应将并联触点多的回路放在左方。这样所编制的程序简洁明了，

语句较少。将图 6-10 中的 I0.0 放在串联电路的右边，并保证功能相同，优化梯形图如图 6-11 所示。

图 6-11 写成语句表形式如下：

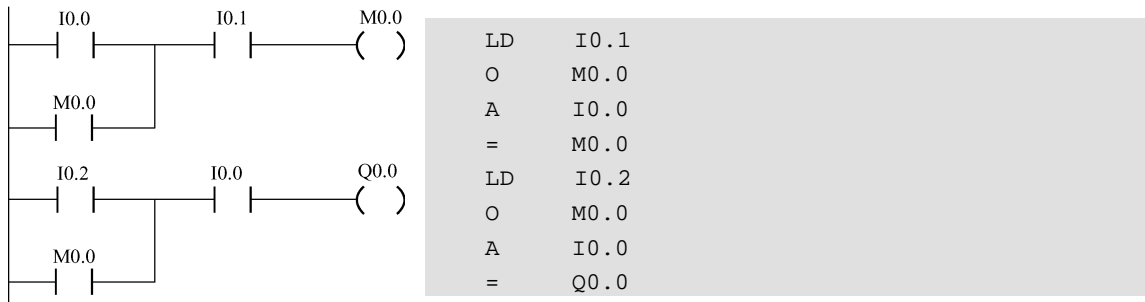


图 6-11 优化梯形图

可以看出第二个语句表程序简单得多，为了减少语句条数，提高程序执行速度，应对梯形图程序进行优化。

图 6-12 是图 6-11 梯形图的一部分，称为启保停电路。

下面结合图 6-9 的任务分析一下启保停电路的特点。

在一般情况下，停止按钮 SB1 一直是闭合的，I0.0 常开触点也是闭合的。按下按钮 SB2，I0.1 常开触点闭合，中间继电器 M0.0 线圈有电、常开触点闭合；松开按钮 SB2，I0.1 常开触点断开，但由于 M0.0 常开触点是闭合的，所以 M0.0 线圈仍然保持有电，电动机一直保持运转。按下 SB1，I0.0 常开触点断开，M0.0 线圈没电并将保持下去。我们把图 6-9 的梯形图程序称为启保停电路，是最简单的启保停电路。该启保停电路的启动电路是 I0.1 常开触点部分，启动条件是启动电路只在启动的瞬间闭合；停止电路是 I0.0 常开触点部分，停止条件是停止电路只在停止的瞬间断开，其他时间是闭合的；保持电路是 M0.0 的常开触点，它与启保停电路中的线圈属于同一个继电器。启保停电路的启动条件和停止条件可以是很复杂的电路，但是必须满足上面所说的特点。

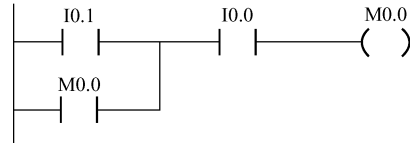


图 6-12 启保停电路

结合启保停电路的特点，同学们思考一下：停止信号使用常开按钮和常闭按钮在软件编程上有何不同？

## 6. 运行调试

(1) 在断电的情况下，按图 6-9 进行 PLC 控制线路接线，主电路接法不变。用编程电缆连接 PLC 和计算机的串行通信接口，接通计算机和 PLC 的电源。

(2) 运行计算机上的 STEP7-Micro/WIN32 编程软件，如果是英文界面，执行菜单命令“Tools”/“Options”，在“General”对话框的“General”选项卡中，选择“Chinese”语言选项，确认后退出，重新运行编程软件，可转换为中文界面。单击工具条上最左边的“新建项目”图标，生成一个新的项目。

(3) 执行菜单命令“PLC”/“类型”，设置 PLC 的型号。设置通信参数，建立起计算机与 PLC 的通信连接。

(4) 执行菜单命令“工具”/“选项”，在“一般”对话框的“一般”选项卡中，选择 SIMATIC 指令集和“国际”助记符集，将“梯形图编辑器”设置为默认的程序编辑器。

(5) 用“查看”菜单选择“梯形图”语言，用“查看”菜单选择“框架”/“指令树”可打开（或关闭）指令树窗口，找到其中的“项目 1”/“程序块”/“主程序（OB1）”，双击“主

程序（OB1）”，在右边“主程序”的编辑窗口中输入图 6-11 所示的梯形图程序。

在“主程序”的编辑窗口中按 F4（也可以单击工具条上的“┌ ┘”或指令树中的“项目 1”/“指令”/“位逻辑”）选中“┌ ┘”后回车，输入“I0.1”后回车，可以故意输入错误的参数，在错误地址的下面将会出现提醒操作者注意的红色波浪线。改正后，波浪线消失。同样的方法完成 I0.1 常开触点的输入，按 F6（也可以单击工具条上的“-( )”或指令树中的“项目 1”/“指令”/“位逻辑”）选中“-( )”后回车，输入“M0.0”后回车，这样第一行程序输入完毕。在第一行程序 I0.1 常开触点的下方单击，然后按上面的方法输入 M0.0 的常开触点，单击 M0.0 常开触点，单击程序输入窗口上方工具条上的“↑”按钮，完成 I0.1 常开触点与 M0.0 常开触点的并联。单击“网络 2”的输入区域中的“→”按钮，按上面的方法完成网络 2 程序的输入。全部程序完成后，如图 6-13 所示。

（6）单击工具条中的“编译”或“全部编译”按钮，编译输入的程序。如果程序没有错误，将显示“0 错误”。否则，改正程序中的错误后才能下载程序。在下载用户程序之前，编程软件将首先自动执行编译操作。

（7）编译成功后，找到指令树中的“项目 1”/“交叉引用”/“交叉引用”，双击“交叉引用”图标，观察出现的交叉引用表。

（8）打开指令树中的“符号表”/“用户定义 1”的符号表，将 I0.1、I0.2 和 Q0.0 设置符号地址“长动”、“点动”和“电动机”，符号表如图 6-14 所示。双击“主程序（OB1）”标签，执行菜单命令“查看”/“符号寻址”，可以切换在主程序中是否显示符号地址。

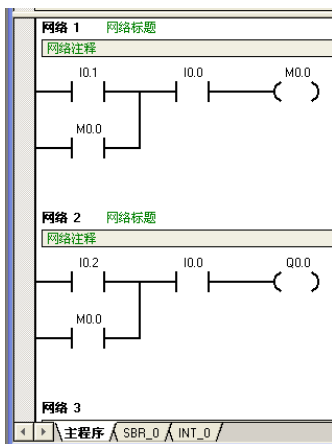


图 6-13 单向点动、自锁混合控制程序

符号表		符号	地址
1		长动	I0.1
2		点动	I0.2
3		电动机	Q0.0
4			
5			


图 6-14 符号表

（9）下载程序与调试程序。下载是从编程计算机将程序装入 PLC；上传则相反，是将 PLC 中存储的程序上传到计算机。计算机与 PLC 建立连接后，将 CPU 模块上的模式开关放在 RUN 位置，单击菜单栏中“文件”/“下载”，或单击工具条中的“↓”按钮，出现图 6-15 所示的下载对话框。在下载对话框中单击“选项”按钮，选择要下载的块。单击“下载”按钮，开始下载。

（10）运行、调试程序。下载成功后，单击“运行”按钮，“RUN”LED 亮，用户程序开始运行。断开数字量输入的全部输入开关，CPU 模块上输入侧的 LED 全部熄灭。用接在端子 I0.2、I0.1 和 I0.0 的开关模拟长动按钮、点动按钮和停止按钮的操作。通过观察各输出点对应的 LED 的状态变化，了解程序的执行情况。



图 6-15 下载程序

(11) 程序运行监控。单击程序状态监控按钮或用菜单命令“调试”→“开始程序状态监控”，在梯形图中显示出各元件的状态。在进入“程序状态”的梯形图中，用彩色块表示位操作数的线圈得电或触点闭合状态。

由于停止按钮为常闭按钮，接在输入端子 I0.0 上，I0.0 线圈有电、常开触点闭合，I0.0 对应的指示灯亮。闭合 I0.2 的开关，Q0.0 指示灯亮，电动机进入点动模式，如图 6-16 所示。断开 I0.2 的开关，Q0.0 指示灯灭，电动机不运行。在 I0.2 开关断开的情况下，接通 I0.1 的开关，Q0.0 指示灯亮，断开 I0.1 的开关，Q0.0 指示灯仍亮，表明长动运行正常，如图 6-17 所示。按下停止按钮，Q0.0 指示灯灭，电动机停止运行。

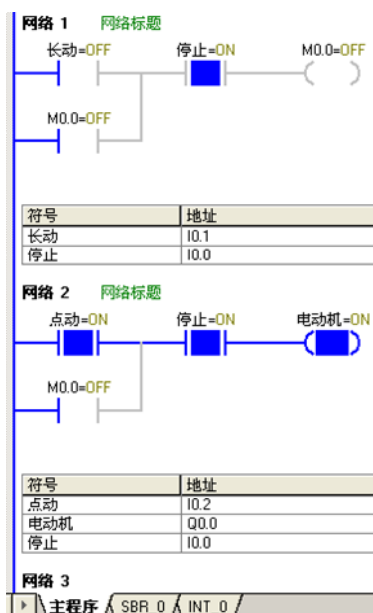


图 6-16 点动运行

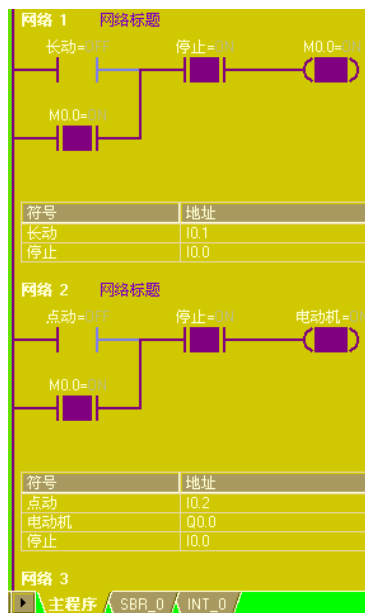


图 6-17 长动运行

## 7. 仿真运行

### 1) 导出程序文本文件

执行菜单命令“文件”/“导出...”，出现如图 6-18 所示的对话框，输入“单向点动、自锁混合控制”，单击“保存”按钮，导出后缀为“awl”的文件“单向点动、自锁混合控制.awl”。因为仿真程序只支持 STEP7-Micro/WIN32 V3.1、V3.2 版导出的程序文件，如果编程软件用的是 STEP7-Micro/WIN32 V4.0 版，导出后用记事本打开文件“单向点动、自锁混合控制.awl”，将第一行中的“主程序”用“MAIN”代替，保存，这样仿真程序就可以使用该程序文件。

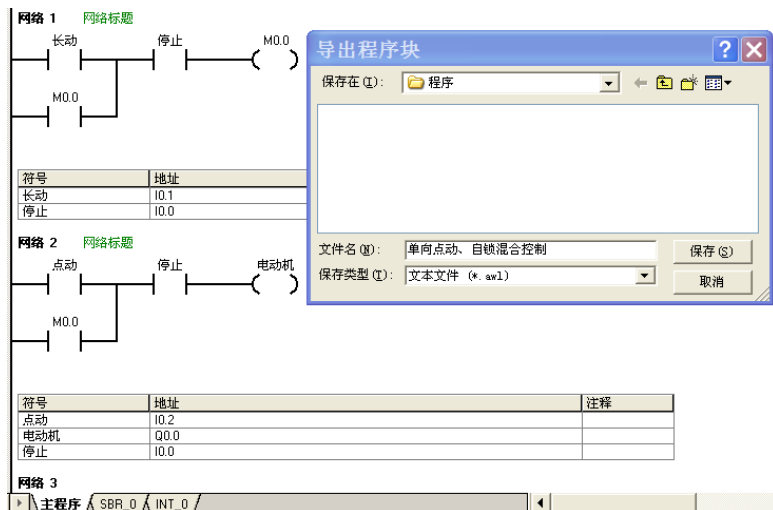


图 6-18 导出文本文件

### 2) 启动仿真程序

双击“S7\_200 汉化版.exe”，输入密码，启动仿真程序，如图 6-19 所示。

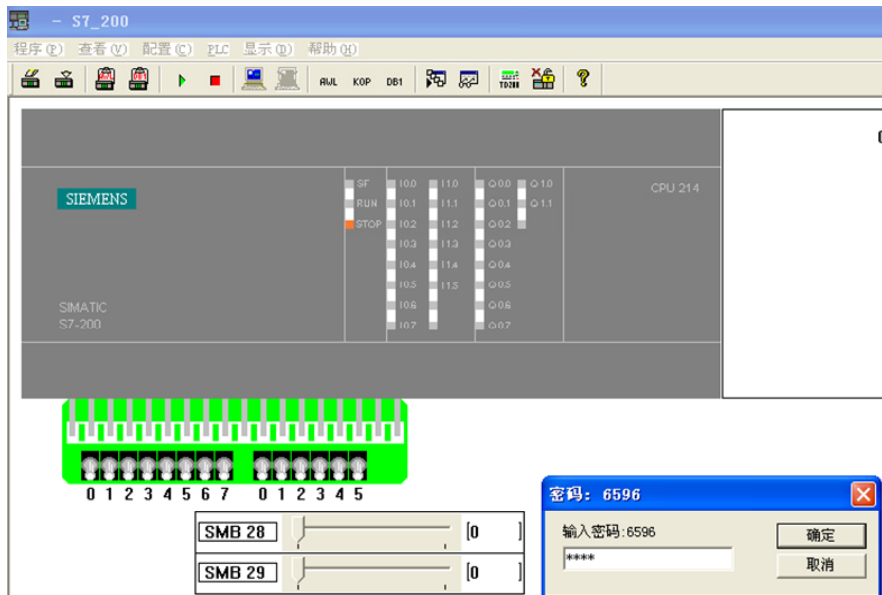


图 6-19 仿真程序的启动

### 3) 选择 CPU

执行菜单命令“配置”/“CPU 型号”，弹出如图 6-20 所示的对话框，选取 CPU221。确定后出现如图 6-21 所示的 CPU221 仿真图形。

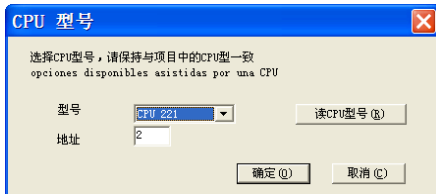


图 6-20 选择 CPU 型号

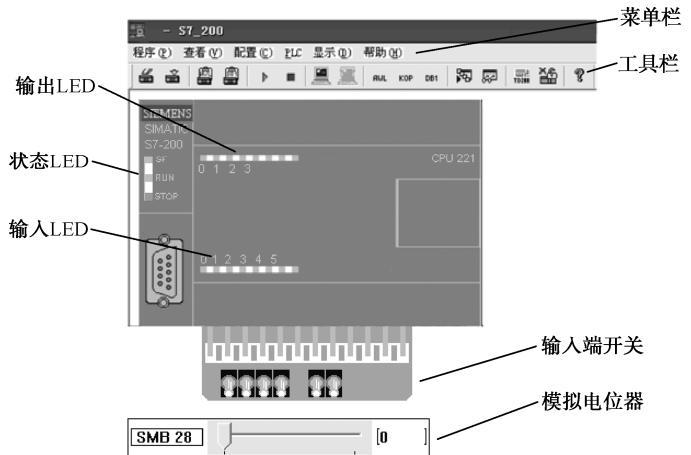


图 6-21 CPU221 仿真图形

### 4) 装入待仿真的程序

执行菜单命令“程序”/“载入程序”，出现图 6-22 所示的对话框，选中逻辑块，确定后出现如图 6-23 所示的对话框，找到上一步保存的文件“单向点动、自锁混合控制.awl”，单击“打开”按钮，进入图 6-24 所示的画面。

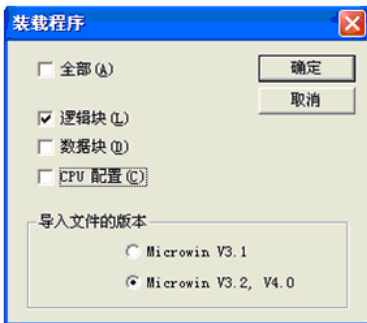


图 6-22 选中逻辑块

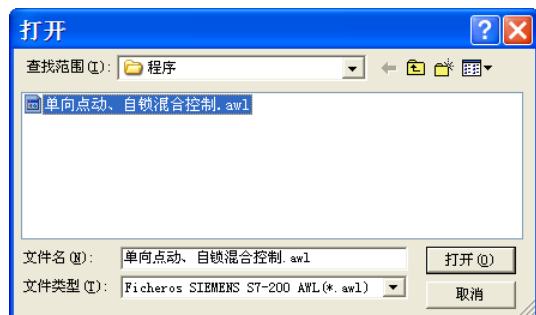




图 6-23 选择程序文件

### 5) 仿真运行

执行菜单命令“PLC”/“运行”或工具条上的“”按钮，出现如图 6-25 所示的对话框，单击“是”按钮，进入运行模式。

### 6) 仿真调试

闭合 I0.0 的开关，用来模拟用作停止功能的常闭按钮，I0.0 对应的指示灯亮。闭合 I0.2 的开关，Q0.0 指示灯亮，电动机进入点动模式。断开 I0.2 的开关，Q0.0 指示灯灭，电动机不运行。点动运行正常。在 I0.2 开关断开的情况下，接通 I0.1 的开关，Q0.0 指示灯亮，断开 I0.1 的开关，Q0.0 指示灯仍亮，长动运行正常。单击“”按钮，输入 I0.0、I0.1、I0.2、Q0.0，进行内存变量监控，如图 6-26 所示。



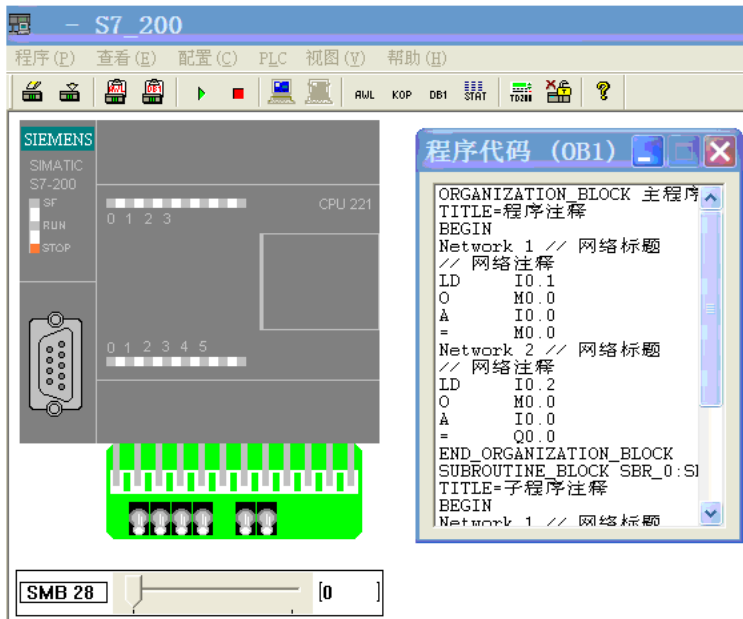


图 6-24 装入程序文件

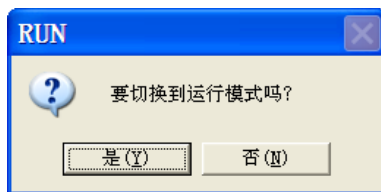


图 6-25 切换运行模式



图 6-26 内存变量监控

### 【评定激励】

按以下标准开展小组自评、互评，成绩填入项目评分细则表，如表 6-15 所示。

表 6-15 项目评分细则表

项目名称			组别		
开始时间			结束时间		
考核内容	考核要求	配分	评分标准	扣分	得分
电路设计	(1) I/O 分配表正确 (2) 输入/输出接线图正确 (3) 主电路正确 (4) 连锁、保护齐全	30 分	(1) 分配表每错一处, 扣 5 分 (2) 输入/输出电路图每错一处, 扣 5 分 (3) 主电路每错一处, 扣 5 分 (4) 连锁、保护每缺一项, 扣 5 分		
安装接线	(1) 元件选择、布局合理, 安装符合要求 (2) 布线合理美观	10 分	(1) 元件选择、布局不合理, 扣 3 分/处; 元件安装不牢固, 扣 3 分/处 (2) 布线不合理、不美观, 扣 3 分/处		
编程调试	(1) 程序编制实现功能 (2) 操作步骤正确 (3) 试车成功	50 分	(1) 输入梯形图错误, 扣 2 分/处 (2) 不会设置及下载, 分别扣 5 分 (3) 一个功能不实现, 扣 10 分 (4) 操作步骤错一步, 扣 5 分 (5) 显示运行不正常, 扣 5 分/处		
安全文明工作	(1) 安全用电, 无人损坏仪器、元件和设备 (2) 保持环境整洁, 秩序井然, 操作习惯良好 (3) 小组成员协作和谐, 态度正确	10 分	(1) 发生安全事故, 扣 10 分 (2) 人为损坏设备、元器件, 扣 10 分 (3) 现场不整洁、工作不文明、团队不协作, 扣 5 分 (4) 不遵守考勤制度, 每次扣 2~5 分		
总成绩					

## 思考与练习

- (1) 简述立即输入指令、立即输出指令的执行过程。
- (2) 分别简述 6 条复杂逻辑指令的执行过程。
- (3) 正跳变与负跳变指令各有什么用途?
- (4) 写出如图 6-27 所示梯形图的语句表程序。
- (5) 根据如下语句表写出梯形图程序。

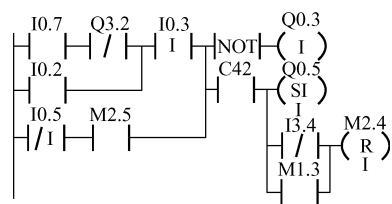


图 6-27 梯形图

LDN	C22	ON	Q0.4
O	M1.3	ALD	
O	M3.5	O	I1.4
LD	M2.1	LPS	
AN	I0.4	EU	
A	T21	S	Q3.3, 1
LD	I0.2	LPP	
AI	I2.7	A	M2.2
OLD		TON	T37, 100