

第4章 FX系列PLC的应用指令

- 4.1 应用指令概述
- 4.2 数据处理指令(FNC40~FNC49)
- 4.3 四则运算及逻辑运算(FNC20~FNC29)
- 4.5 程序流程控制 (FNC00~FNC09)
- 4.6 高速处理(FNC50~FNC59)
- 4.7 方便指令 (FNC60~FNC69)
- 4.8 外部设备指令

PLC应用指令的应用相关设计任务

任务一 送料小车往返于工作台控制系统设计

任务二 数字钟显示系统设计

任务三 自动售货机控制系统设计

任务四 流水灯控制系统设计



任务五 数字钟显示控制系统设计

任务六 声光报警控制系统设计

任务七 钢板裁剪控制系统设计

4.1 概述

定义

可编程控制器的基本指令——基于继电器、定时器、计数器类软元件,主要用于逻辑处理的指令。

功能指令(应用程序)——用于数据的传送、运算、变换及程序控制等功能。

功能指令的特点

- > 功能强大
- > 指令处理的数据多
- > 数据在存储单元中流转的过程复杂



一. 功能指令的结构

X8002

指令名称

操作数

▶ 1、指令名称:

作用:

表示指令实现的功能。

表示:

助记符: 指令功能的英文缩写

编号: 每条指令对应一个编号。

FNC00-FNC246

▶ 2、操作数

作用:

指令执行时使用的或产生的数据

分类:

源操作数:

指令执行时使用的数据,用[S.]表示

目的操作数:

指令执行时产生的数据,用[D.]表示

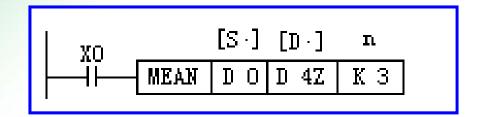
其它操作数:

补充说明的常数,用[m.]、[n.]表示

^



例如下图中的功能指令是一个取平均值的指令



其功能如下式表达:

$[(D0)+(D1)+(D2)] \div 3 \rightarrow (D4Z)$

图中标注 [S·] 指取值首元件。n指定取值个数。[D·]指定 计算结果存放地址。



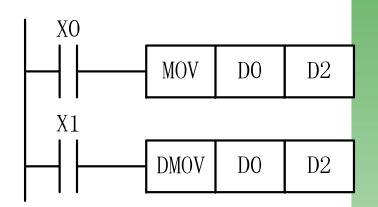
二、功能指令的格式及说明

- 1、功能指令使用的软元件
- 功能指令使用的软元件有字元件和位元件两种类型。
- 能表达数值的元件叫做字元件,字元件有三种类型:
- (1)、常数: "K"表示十进制常数, "H"表示十六进制常数, 如K1369, H06C8,
- (2)、位元件组成的字元件: KnX、KnY、KnM、KnS,如 K1X0,K4M10,K3S3,
- (3)、数据寄存器: D、V、Z、T、C,如D100,T0。

三、指令处理的数据长度

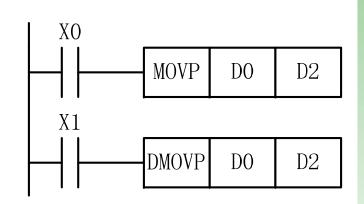
功能指令可以处理16位和32位数据

在助记符前加"D"就变成32位指令



四、指令执行形式

应用指令有连续执行和脉冲执行型两种执行形式。

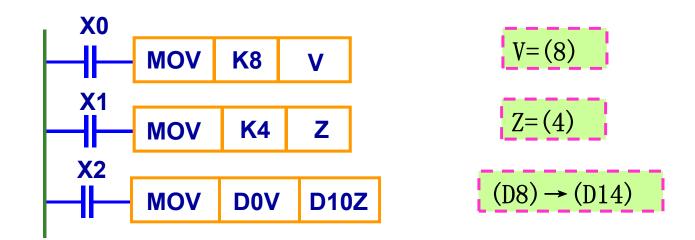


在应用指令助记符中标有"P"的表示该指令为脉冲 执行型。



五、变址寄存器V、Z

- ❖ 变址: 改变操作数的地址
- * 变址寄存器的作用: 存放改变地址的数据
- * 实际地址=当前地址+变址数据
- ❖ 32位运算时V和Z组合使用, V为高16位, Z为低16位。





五、常用特殊辅助继电器

* 功能指令执行结果的标志

▶ M8020: 零标志

▶M8021: 借位标志

▶M8022: 进位标志

▶ M8029: 执行完毕标志

▶ M8064: 参数出错标志

▶ M8065: 语法出错标志

▶ M8066: 电路出错标志

▶M8067: 运算出错标志

4.2 数据处理指令之传送和比较指令

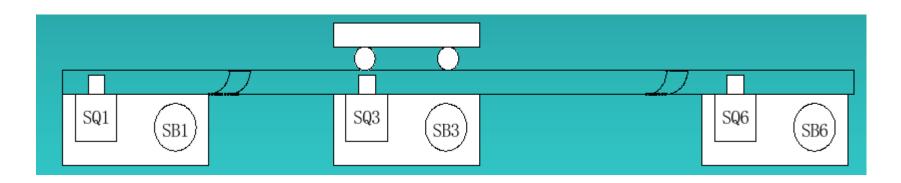
- 一、比较指令 FNC10 CMP
- 二、区间比较指令 FNC 11 ZCP
- 三、传送指令 FNC 12 MOV
- 四、移位传送指令 FNC 13 SMOV
- 五、取反传送指令 FNC 14 CML
- 六、块传送指令 FNC 15 BMOV
- 七、多点传送指令 FNC 16 FMOV
- 八、数据交换指令 FNC 17 XCH
- 九、变换指令 FNC 18 BCD FNC 19 BIN

任务一 送料小车往返于工作台控制系统设计

一、任务导入

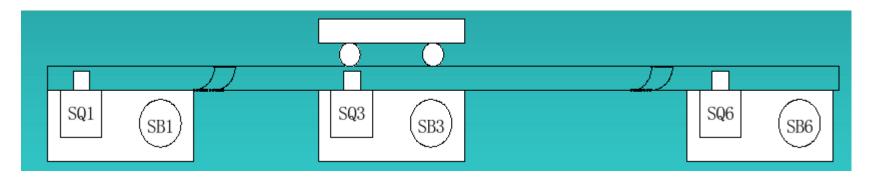
某车间有 6 个工作台,送料车往返于工作台之间送料,如下图所示。每个工作台设有一个到位开关(SQ)和一个呼吸按扭(SB)。具体控制要求如下:

- (1)送料车开始应能停留在6个工作台中任意一个到位开关的位置上。
- (2) 设送料车现暂停于 m 号工作台(SQ m 为 ON) 处,这时 n 号工作台呼叫(SB n 为 ON),若:



任务一 送料小车往返于工作台控制系统设计

一、任务导入

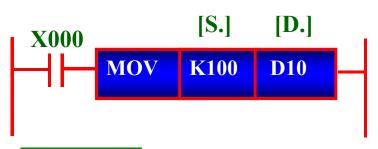


- (a) m>n,送料车左行,直至 SQ n 动作,到位停车。即送料车所停位置 SQ 的编号大于呼叫按扭 SB 的编号时,送料车往左行运行至呼叫位置后停止。
- (b) m<n,送料车右行,直至 SQ n 动作,到位停车。即送料车所停位置 SQ 的编号小于呼叫按扭 SB 的编号时,送料车往右运行至呼叫位置后停止。
- (c) m=n,送料车原位不动。即送料车所停位置 SQ 的编号与呼叫按扭 SB 的编号相同时,送料车不动。

二、关键知识

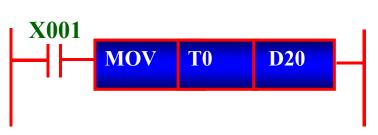
传送指令(MOV): MOV传送指令是将源操作数内的数据传送到指定的目的操作数去,即S→D。

MOV指令使用说明

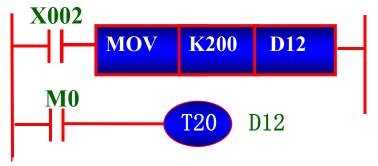


当X0为0N时,源操作数 [S.] 中的常数K100传送到目标操作元件D10中。当指令执行时,常数K100自动转换成二进制数。

举例



计数器、计时器当前值读出



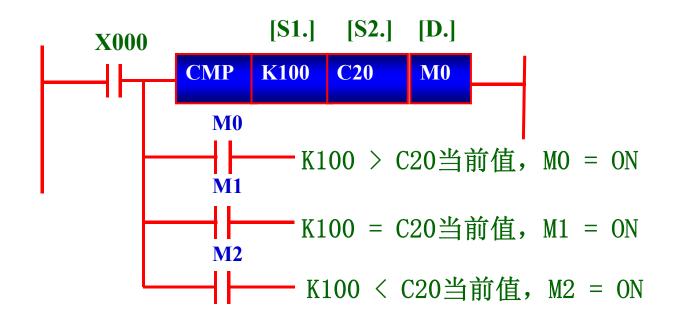
计数器、计时器设定值被指定

传送比较指令说明

2. 比较指令(CMP): 比较指令是将源操作数S1、S2的数据,接照代数规则进行大小比较,并将比较结果送到目的操作数D中,驱动目的操作数及相邻的位元件。

- 1) 比较指令中的所有的源操作数据都按二进制数值处理。
- 2) 当条件断开不执行CMP指令时,目标元件保持条件断开前的状态,要清除结果,要采用RST或ZRST复位指令。

CMP指令使用说明



比较指令用于建立控制点

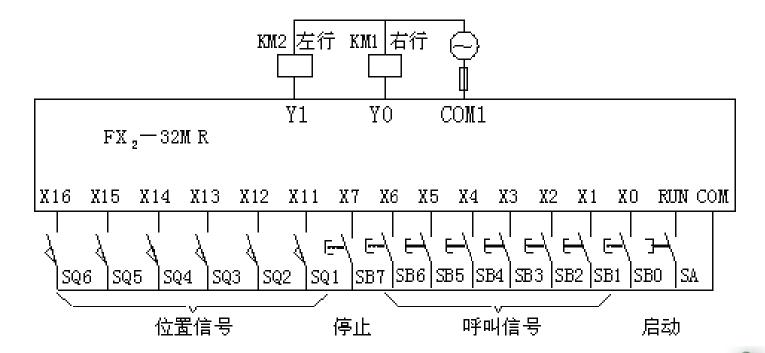
控制现场常有将某个物理量的量值或变化区间作为控制点的情况。如温度低于多少度就打开电热器,速度高于或低于一个区间就报警等。作为一个控制"阀门",比较指令常出现在工业控制程序中。

二、 PLC 硬件的实现

1 、 1/0 的分配

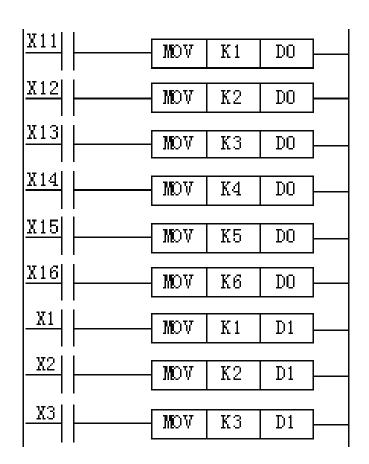
	输入	功能说明		输出	功能说明
SB0	ХO	启动	KM1	YO	右行
SB1	X 1	呼叫 1	KM2	Y1	左行
SB2	X2	呼叫 2			
SB3	ХЗ	呼叫 3			
SB4	X4	呼叫 4			
SB5	X5	呼叫 5			
SB6	Х6	呼叫 6			
SB7	Х7	停止			
SQ1	X 1 1	限位 1			
SQ2	X12	限位 2			
SQ3	X13	限位 3			
SQ4	X14	限位 4			
SQ5	X15	限位 5			
SQ6	X16	限位 6			

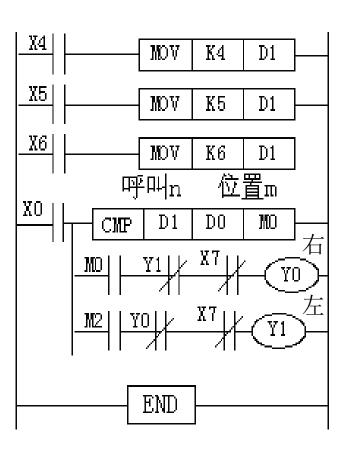
2 、 I/O 的外部接线





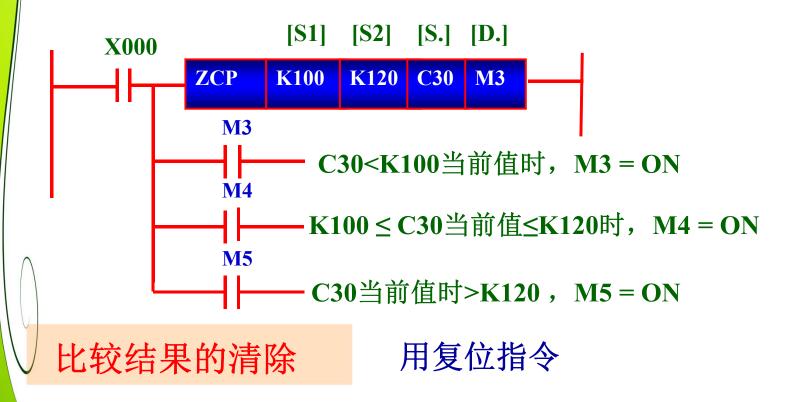
(三)、 PLC 软件的实现





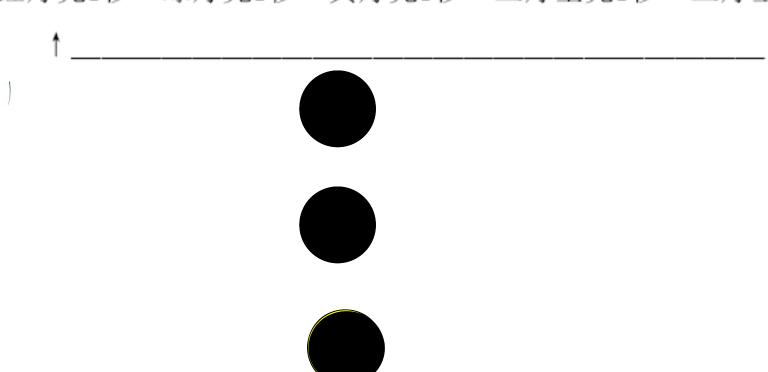
四、知识链接

区间比较指令(ZCP): 比较指令是将源操作数S1、S2的数据,按照代数规则进行大小比较,并将比较结果送到目的操作数D中,驱动目的操作数及相邻的位元件。



例子4.1

红灯亮1秒→绿灯亮1秒→黄灯亮1秒→三灯全亮1秒→三灯全灭1秒



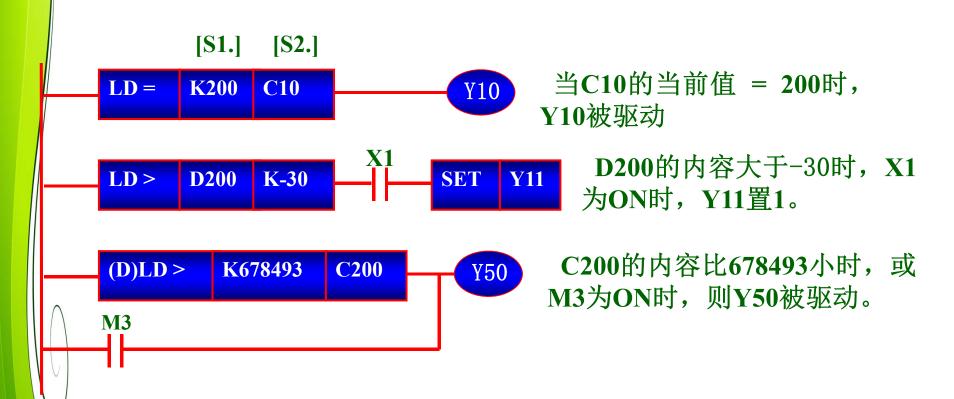
- ■一、问题的提出
- 1、可不可以用1个定时器实现以上功能

红灯亮1秒→绿灯亮1秒→黄灯亮1秒→三灯全亮1秒→三灯全灭1秒

```
M8000
                                                                   K50
           ΤO
                                                               -(TO
                       -[ZCP
                                 K10
                                           K20
                                                    ΤO
                                                               M10
                       -[ZCP
                                 K30
                                           K40
                                                    ΤO
                                                               M13
 M10
                                                               -(Y000
 M14
 M11
                                                               -(Y001
 M14
 M12
          M15
                                                               -(Y002
 M14
```

四、知识链接

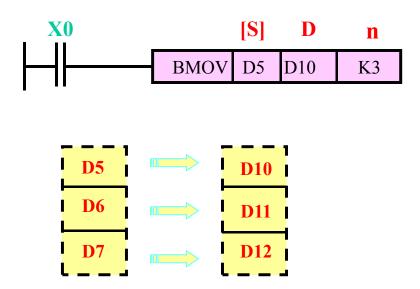
触点形比较指令: 是使用触点符号进行数据比较的指令, 根据比较结果确定触点是否允许能流通过。

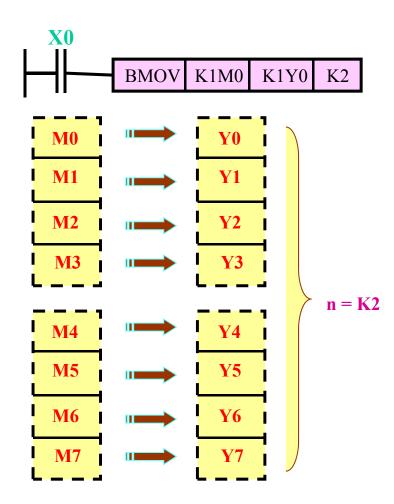




		_	
)	助记符	比较条件	逻辑功能
取比较	LD=	[S1]=[S2]	[S1]与[S2]相等
指令	LD>	[S1]>[S2]	[S1]大于[S2]
	LD<	[S1]<[S2]	[S1]小于[S2]
	TD<>	$[S1] \neq [S2]$	[S1]与[S2]不相等
	TD<=	[S1]≤[S2]	[S1]小于等于[S2]
	$\Gamma D \rangle =$	[S1]≥[S2]	[S1]大于等于[S2]
串联比	AND=	[S1]=[S2]	[S1]与[S2]相等
较指令	AND>	[S1]>[S2]	[S1]大于[S2]
	AND<	[S1]<[S2]	[S1]小于[S2]
	AND<>	$[S1] \neq [S2]$	[S1]与[S2]不相等
	AND<=	[S1]≤[S2]	[S1]小于等于[S2]
	AND>=	[S1]≥[S2]	[S1]大于等于[S2]
并联比	OR=	[S1]=[S2]	[S1]与[S2]相等
较指令	OR>	[S1]>[S2]	[S1]大于[S2]
	OR<	[S1]<[S2]	[S1]小于[S2]
	OR<>	$[S1] \neq [S2]$	[S1]与[S2]不相等
	OR < =	[S1]≤[S2]	[S1]小于等于[S2]
	OR>=	[S1]≥[S2]	[S1]大于等于[S2]

块传送指令 FNC15 BMOV





多点传送指令 FNC16 FMOV



数据交换指令 FNC17 XCH

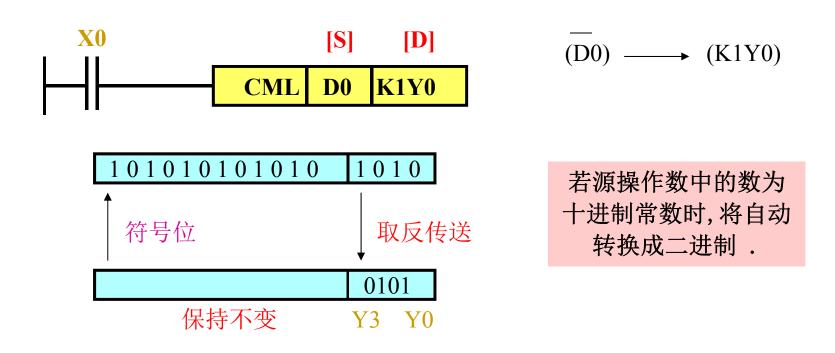


❖ 注意

交换指令一般要在脉冲方式执行,否则不能正常工作。

取反传送指令 FNC14 CML

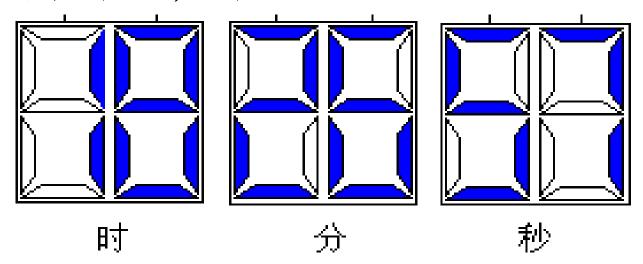
❖ 梯形图



任务二 数字钟显示控制系统设计

一、任务引入

设计一个24h时钟,分别用七段数码管显示时、分、秒,并 能通过外部调节按钮,调节时间显示值。



根据任务要求,可以利用PLC的计数器分别计时间的时、分、秒,然后用编码指令将计数器中的二进制数转换成BCD码,最后用7段码译码指令将相应的时间数据显示出来,这就需要用到BCD码指令和SEGD指令。

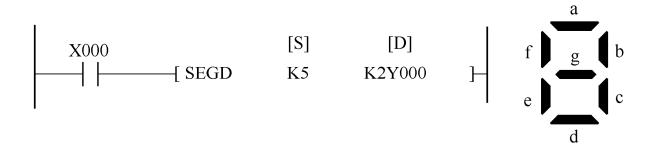
二、相关知识

(一) 七段译码指令SEGD

7段译码指令SEGD的助记符、操作数等指令属性见表所示。

比人分秒	ロ ル ンコ <i>ケ</i> ケ	元·公比 口.	操作数				
指令名称 	助记符	功能号	[S]	[D]			
7段译码	SEGD (P)		K、H、KnX、 KnY、KnM、KnS、 T、C、D、V、Z				

7段译码指令SEGD (P) 如图所示,将源操作数[S]中指定元件的低4位所确定的十六进制数 (0~F) 经译码后存于[D] 指定的元件中,以驱动7段数码管,[D]的高8位保持不变。



如图所示,当X0闭合时,对数字5执行7段译码指令SEGD,并将译码H6D 存入输出位组件K2Y0,即输出继电器Y7~Y0的位状态为0110 1101。

2. 使用SEGD指令时应注意:

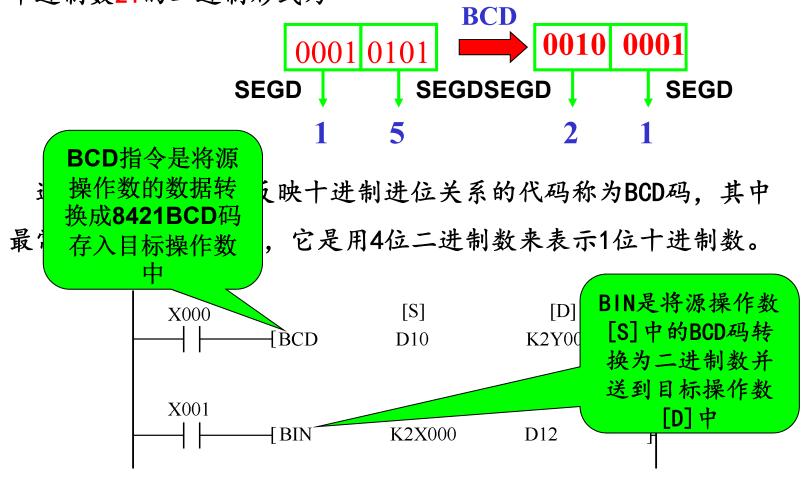
- ①源操作数[S]可取K、H、KnX、KnY、KnM、KnS、T、C、D、V和Z; 目标操作数[D]可取KnY、KnM、KnS、T、C、D、V和Z。
- ②SEGD指令是对4位二进制数编码,若源操作数大于4位,只对最低4位编码。
- ③SEGD指令的译码范围为一位十六进制数字0~9、A~F。

七段码译码表

源操作数		一年北州悠	目标输出						
16进制数	低4位数	- 七段数码管	g	f	е	d	С	b	a
0	0000		1	1	1	1	1	1	1
1	0001		0	0	0	0	1	1	0
2	0010	a	1	0	1	1	0	1	1
3	0011		1	0	0	1	1	1	1
4	0100	f g b	1	1	0	0	1	1	0
5	0101		1	1	0	1	1	0	1
6	0110		1	1	1	1	1	0	1
7	0111	e c	0	1	0	0	1	1	1
8	1000		1	1	1	1	1	1	1
9	1001		1	1	0	1	1	1	1
A	1010	d	1	1	1	0	1	1	1
В	1011		1	1	1	1	1	0	0
С	1100		0	1	1	1	0	0	1
D	1101		1	0	1	1	1	1	0
Е	1110		1	1	1	1	0	0	1
F	1111		1	1	1	0	0	0	1

(二) 数据变换指令BCD和BIN

十进制数21的二进制形式为

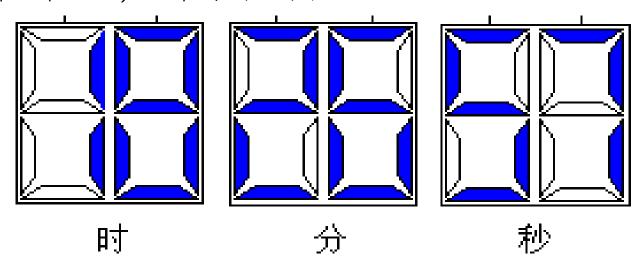


BCD指令注意事项

- (1) BCD指令是将源操作数的数据转换成8421BCD码存入目标操作数中。在目标操作数中每4位表示1位十进制数,从低到高分别表示个位、十位、百位、千位......16位数表示的范围为0~9999,32位数表示的范围为0~9999999;
- (2) BCD指令若转换成32位数字时,前面要加D,采用脉冲执行方式时,指令后面要加P。

(一) 任务要求

设计一个24h时钟,分别用七段数码管显示时、分、秒,并能通过外部调节按钮,调节时间显示值。



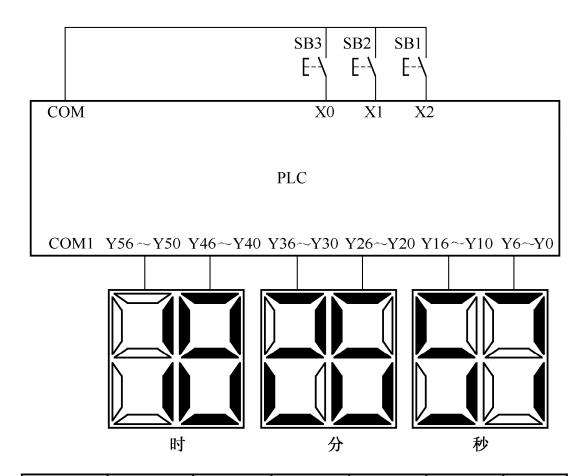
根据任务要求,可以利用PLC的计数器分别计时间的时、分、秒,然后用编码指令将计数器中的二进制数转换成BCD码,最后用7段码译码指令将相应的时间数据显示出来,这就需要用到BCD码指令和SEGD指令。

(二) 硬件1/0分配及接线

1. I/0分配: 通过分析任务,各元件的I/0分配和作用如表所示。

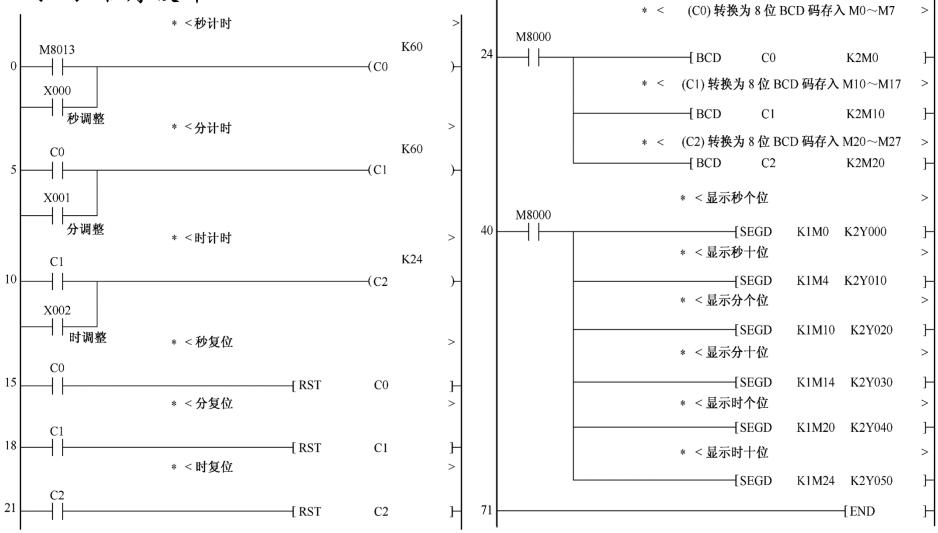
输入			输出		其他软元件		
输入继电器	输入元件	作用	输出继电器	作用	СО	秒计数	
X000	按钮SB3	秒调整	Y006~Y000	秒个位	C1	分计数	
X001	按钮SB2	分调整	Y016~Y010	秒十位	C2	时计数	
X002	按钮SB1	时调整	Y026~Y020	分个位	M3∼M0	存秒个位	
			Y036~Y030	分十位	M7∼M4	存秒十位	
			Y046~Y040	时个位	M13~M10	存分个位	
			Y056~Y050	时十位	M17~M14	存分十位	
					M23~M20	存时个位	
					$M27\sim$ $M24$	存时十位	

2. PLC硬件接线



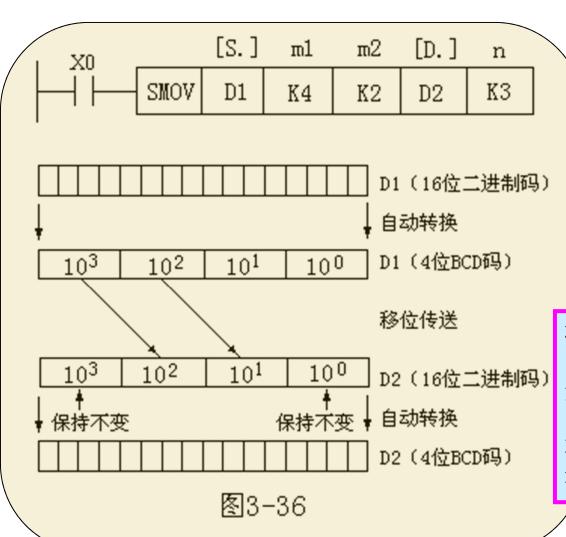
输出	Y56~Y50	Y46~Y40	Y36~Y30	Y26~Y20	Y16~Y10	Y6~Y0
BCD 码	1	8	2	6	5	7

(三) 程序设计



四、知识的链接

移位传送指令SMOV SMOV(P)



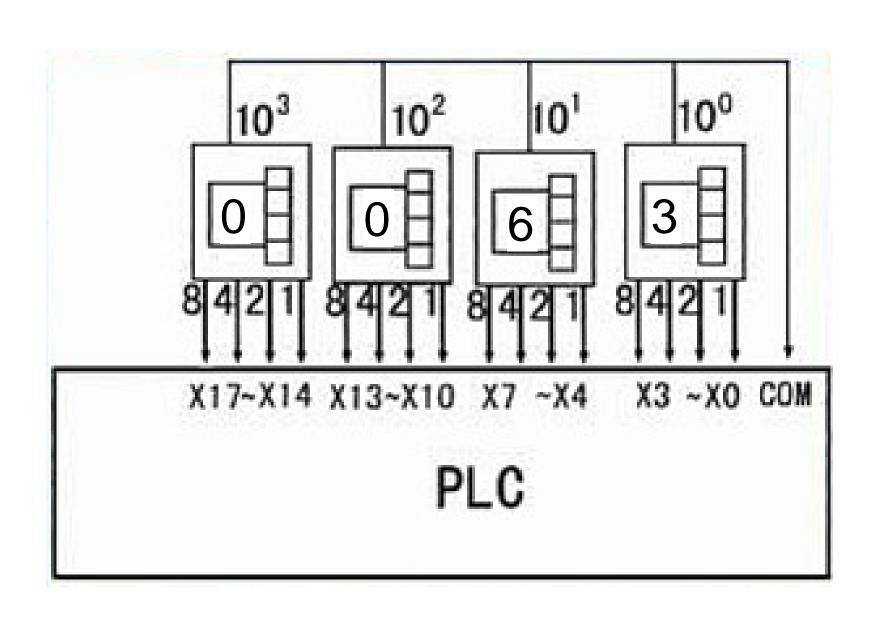
m1:从哪一位开始移动

m2:移动多少位

梯形图表达:

源数据BCD码右起第4位(m1=4) 开始的2位(m2=2)移到目标的第3位 (n=3)和第2位。然后将BCD码 转换为二进制,其中第1、4位不受 移位指令的影响。

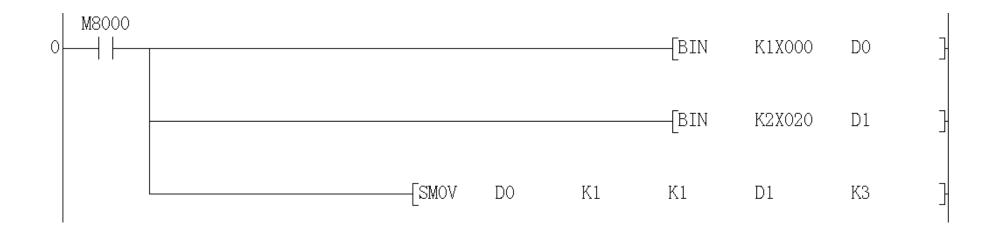
例子4.2: PLC外接拨码开关



BCD码=63,如果直接输入是二进制01100011(十进制的99)会出错。

BIN先将63看成10进制转化成二进制00111111,不会出错。





LD M8000

OUT M8168 //BCD码方式

MOV K1X0 D0

MOV K2X20 D1

SMOV D0 K1 K1 D1 K3

四、知识的链接

数据交换指令 FNC17 XCH

*操作数

[D1], [D2]: KnY, KnM, KnS, T, C, D, V, Z

❖ 梯形图



❖ 注意

交换指令一般要在脉冲方式执行,否则不能正常工作。



ROR: 右循环移位

■ ROL: 左循环移位

■ SFTR: 位右移指令

■ SFTL: 位左移指令

■ WSFR: 字右移指令

■ WSFL: 字左移指令

■ SFWR: 移位寄存器写入指令

SFRD:移位寄存器读出指令



任务三 流水灯控制系统设计

一、任务导入

某灯光招牌有L1~L8,8个灯接于K2Y000,要求当X000为0N时, 灯先以正序每隔1S轮流点亮,当Y007亮后,停2S;然后以反序每隔1S轮流点亮,当Y000再亮后,停2S,重复上述过程。当X001为 ON时,停止工作。

根据任务要求,可以用PLC的循环移位指令实现上述任务。

二、相关知识

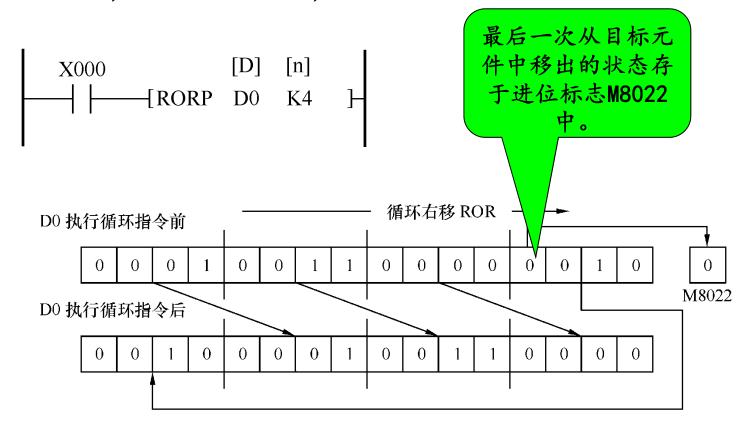
(一)循环移位指令ROR、ROL、RCR和RCL

循环移位指令包括ROR、ROL、RCR和RCL指令。这些指令的名称、助记符功能号、操作数见表所示。

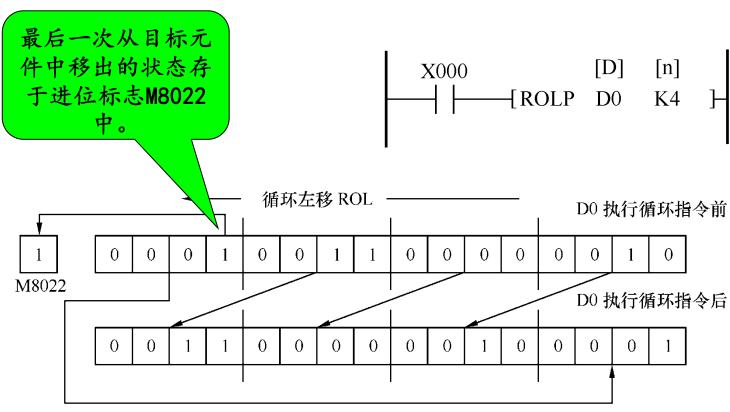
指令名称	助记符	功能号	操作数		
相令石孙	奶儿行		[D·]	n	
循环右移	ROR	FNC30			
循环左移	ROL	FNC31	KnY、KnM、 KnS、T、C、 D、V、Z	K、H 16に操作、n < 16	
带进位右移	RCR	FNC32		16位操作: n≤16 32位操作: n≤32	
带进位左移	RCL	FNC33			

1.右、左循环移位指令ROR、ROL

右移指令ROR: 设(D0)循环前为H1302,则执行"RORP D0 K4"指令后,(D0)为H2130,进位标志位(M8022)为0。



指令ROL:设(DO)循环前为H1302,则执行"ROLP DO 指令后,(DO)为H3021,进位标志位(M8022)为1。

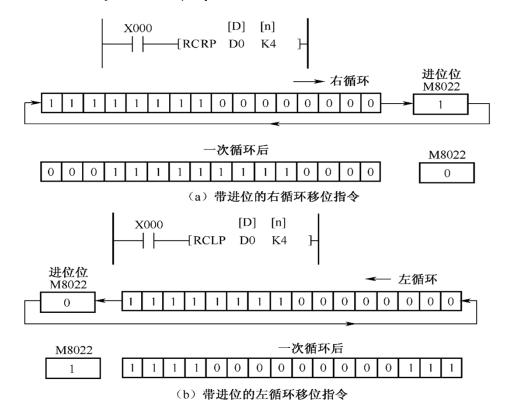


左循环移位指令ROR、ROL的说明

- (1)如上图所示,在X0由OFF变为ON时,循环移位指令ROR或ROL执行,将目标操作数D0中的各位二进制数向右或向左循环移动4位,最后一次从目标元件中移出的状态存于进位标志M8022中。
- (2)循环移位是周而复始的移位,D为要移位的目标操作数,n为移动的位数。ROR和ROL指令的功能是将D中的二进制数向右或向左移动n位。移出的最后一位状态存在进位标志位M8022中。
 - (3) 若在目标元件中指定位元件组的组数时,只能用K4(16位指令或K8(32位指令)表示,如K4M0或K8M0。字操作指令
- (4)在指令的连续执行方式中,每一个扫描周期都会移位一次。在 实际控制中,常采用脉冲执行方式。



2. 带进位的循环移位指令



注意该指令最好采用脉冲指令形式,只有在X0由0FF变为0N时,目标操作数D0中的各位二进制数才移位一次,否则在每一个扫描周期都会移位。

三、任务实施

(一) 任务要求

某灯光招牌有L1~L8个灯接于K2Y000,要求当X000为0N时,灯先以正序每隔1S轮流点亮,当Y007亮后,停2S;然后以反序每隔1S轮流点亮,当Y000再亮后,停2S,重复上述过程。当X001为0N时,停止工作。

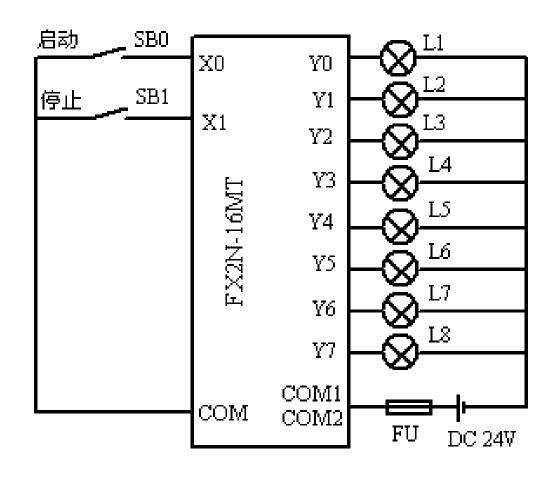
硬件1/0分配及接线

1. 1/0分配

通过分析任务要求知,该控制系统有2个输入按钮,8个输出灯,因此, 具体1/0分配如表所示。

	输入		输 出		
输入继 电器	输入元件	作用	输出继 电器	输出 元件	作用
X000	按钮SB0	起动按钮	Y000∼ Y007	L1~ L8	流水灯
X001	按钮SB1	停止按钮			

2. 产 使件接线



程序设计

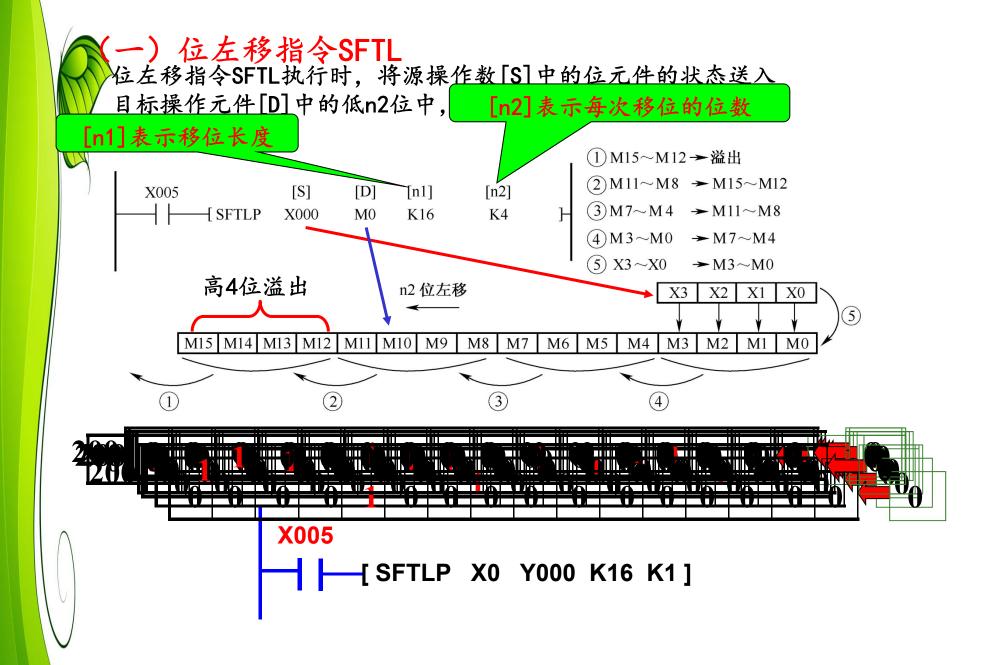
下起动按钮X0, Y000=1, 因X0是瞬动信号,因此X0有效 时,置位MO,将起动信号保存 下来,在MO有效的情况下,每 隔 IS,从Y000开始,循环向左 移位,轮流点亮流水灯; 当L8 灯点亮时,即Y007=1,置位M1, 延时2S后,从Y007开始,循环 <mark>向</mark>右移位,逆序点亮流水灯, <mark>使</mark>向右循环移位停止,延时5S, 时间到,复位M1,置位M0,程 序重复运行。

```
X000
                                             -[MOVP
                                                       K1
                                                                   K4Y000
     X000
                 M1
                          X001
                                                                  (MO
      MO
      T1
      MO
               M8013
13
                                             -FROLP
                                                       K4Y000
                                                                   K1
     Y007
                                                        [SET
20
                                                                   M1
      M1
                                                                      K20
22
                          X001
      T0
               M8013
                                     M2
26
                                             -[RORP
                                                       K4Y000
                                                                   K1
                                                                      K50
                Y000
                                                                  \{M2\}
      Т1
                                                        -[RST
                                                                   M1
     X001
     X001
                                             -[MOVP
44
                                                       ΚO
                                                                   K4Y000
50
                                                                  FEND
```

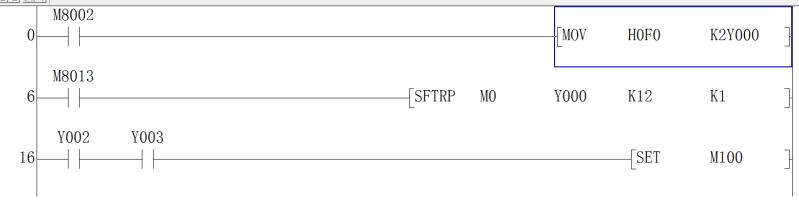
四、知识链接

(一)移位指令包括SFTR、SFTL、WSFR和WSFL。这些指令的名称、助记符、功能号、操作数等如表所示。

指令名称 助记》	助记符	功能号	操作数			
相令石孙	四十一十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十		[S·]	[D ·]	n1 n2	
位右移	SFTR (P)	FNC34	X, Y, M, S	Y, M, S	K、H n2≤n1≤1024	
位左移	SFTL (P)	FNC35				
字右移	WSFR (P)	FNC36	KnX、KnY、 KnM、KnS 、T、C、D	KnY, KnM, KnS, T, C, D	K, H	
字左移	WSFL (P)	FNC37			n2≤n1≤512	





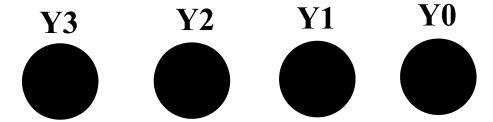




- (1) S为移位的源操作数的最低位, D为被移位的目标操作数的最低位。n1为目标操作数个数, n2 为源操作数个数。
- (2) 位左移就是源操作数从目标操作数的低位移入n2位,目标操作数各位向高位方向移n2位,目标操作数中的高n2位溢出。源操作数各位状态不变。
- (3) 在指令的连续执行方式中,每一个扫描 周期都会移位一次。在实际控制中,常采用脉冲执 行方式。

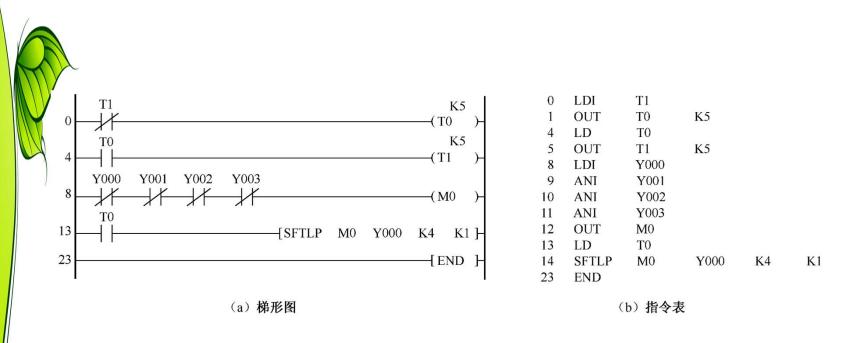
位左移指令SFTL举例

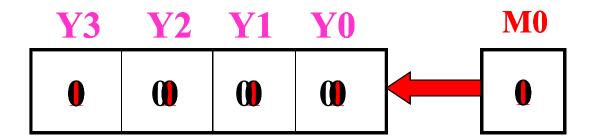
根据控制要求,写出4盏流水灯的真值表,如表所示。



4盏流水灯循环左移真值表

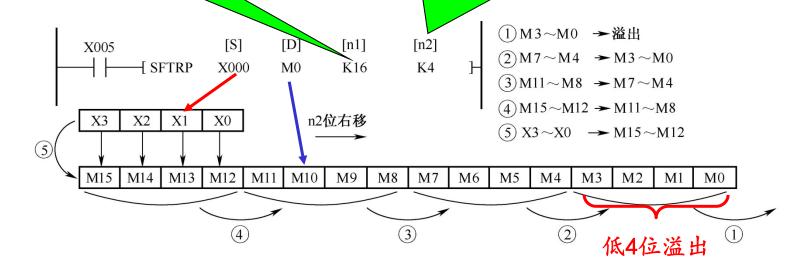
	脉冲	Y3	Y2	Y1	Y0
	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	0
	3	0	1	0	0
	4	1	0	0	0





二)位右移指令SFTR

位右移指令SFTR执行时,将源操作数[S]中的位元件的 [n1]表示移位长度 作元件[D]中的低n2位中,并依次将目 你探作致。 [n2]表示每次移位的位数

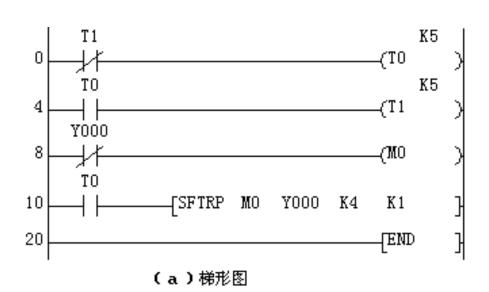


位右移指令SFTR的说明

- (1) S为移位的源操作数的最低位, D为被移位的目标操作数的最低位。n1为目标操作数个数, n2为源操作数个数。
- (2) 位右移就是源操作数从目标操作数的高位移入n2位,目标操作数各位向低位方向移n2位,目标操作数中的低n2位溢出。源操作数各位状态不变。
- (3) 在指令的连续执行方式中,每一个扫描 周期都会移位一次。在实际控制中,常采用脉冲 执行方式。

2. 位右移指令SFTL举例 4盏流水灯循环右移真值表

脉冲	Y3	Y2	Y 1	Y0
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1



四、知识链接

(二) 先入先出(FIFO)写入SFWR、读出SFRD

指令名称 助记符	助泊绘	功能号		操作	数
	別心行		[S·]	[D·]	n
FIFO写入	SFWR	FNC38	K, H, KnX, KnY、KnM、 KnS、T、C、D	KnY、KnM 、KnS、T、	K、H 2≤n≤512
FIFO读出	SFRD	FNC39	KnY、KnM、 KnS、T、C、D	C, D, V, Z	<u>2</u> ≤11 ≥312

n指定数据的长度。

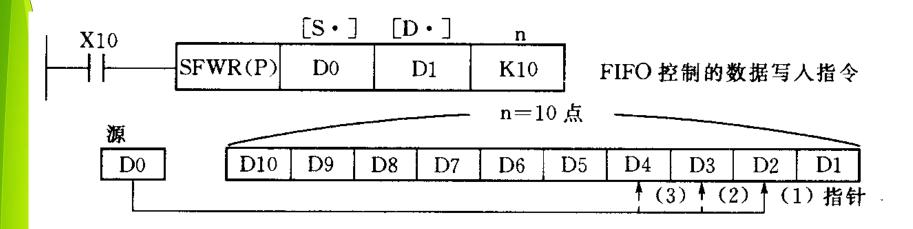


图 2-62 SFWR 指令的使用说明

- •当X10首次由OFF变为ON,SFWR将源元件DO中的数据 写入D2,而D1作为指针变为1(D1要先清O);当
- •X10再次由OFF变为ON时,DO中的数据写入D3,D1中的数据加1变为2。依次类推,将DO中的数据依次写入寄存器。
- •SFWR数据是从最右边的寄存器开始依次写入,写入的次数放在D1中,D1称为指针。
- *当D1的内容达到n-1后,操作不再执行,进位标M8022 置1。



移位写入(假设D10初始值为0, D20--D24的初始值为0)

LDP M0 ADD D10 K2 D10 SFWR D10 D20 K5



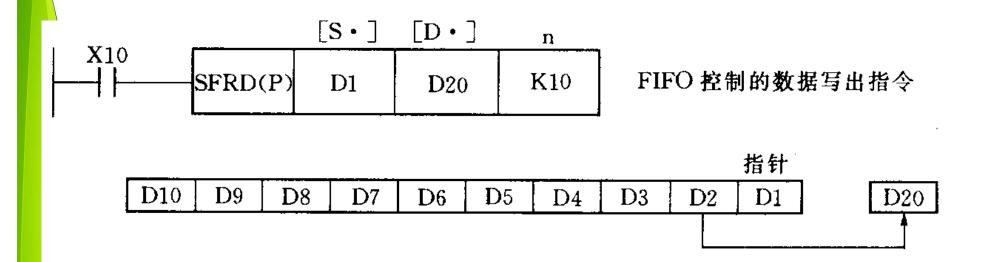


图 2-63 SFRD 指令的使用说明

- •当X10首次由OFF变为ON,SFRD将源元件D2中的数据读出到D20,而D1作为指针减1,D10到D3的数据右移一字。
- •若连续指令SFRD,则每个扫描周期数据右移一字,而数据总是从D2读出。
- •当指针D1为O时,操作不执行,零标志M8O2O置1。
- 先入先出(FIFO)控制常用于按产品入库并顺序从 库中取出产品。



例子:移位读出(假设D30=4, D31=2,

D32=4, D33=6, D34=8)

LD M0

SFRDP D30 D20 K5





产品入库出库(FIFO)控制

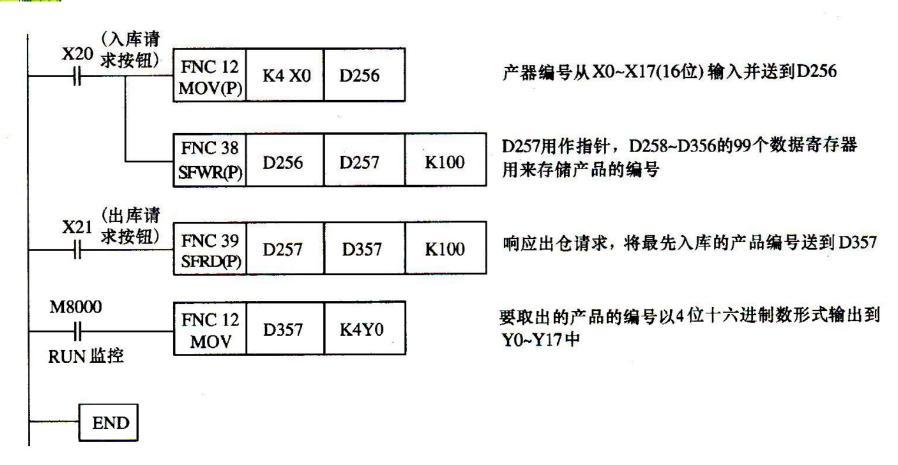


图 6-86 产品入库出库梯形图

4.2数据处理指令之数据处理指令

- 区间复位指令(ZRST)
- 解码指令(DECO)与编码指令(ENCO)
- <u>求置ON位总数(SUM)与ON位判别指令(BON)</u>
- 报警器置位(ANS)和复位(ANR)指令
- 其他指令 (MEAN、SQR、FLT、SWAP)

任务四用解码指令实现单按钮控制五台电动机的启停

一、任务导入

单按钮控制五台电动机的启停如图4-2-1所示。按钮按数次,最后一次保持 1s 以上后,则号码与按钮次数相同的电机运行,再按按钮,该电机停止。五台电动机接于 Y0-Y4。

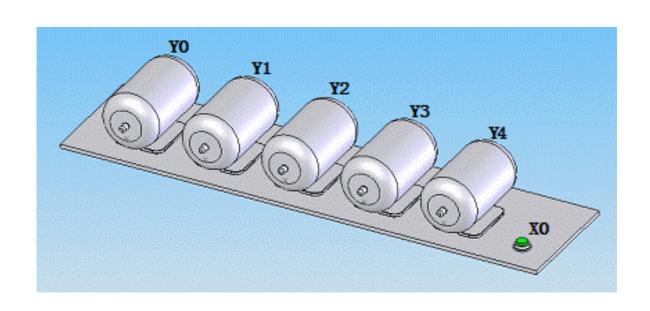


图4-2-1 单按钮控制五台电动机启停

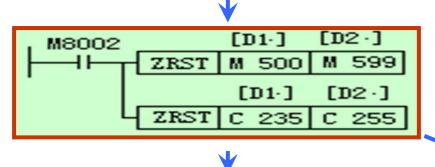
任务二

二、关键知识

指令: ZRST FNC40(P) (16)区间复位

程序步数: ZRST, ZRST(P)...7步

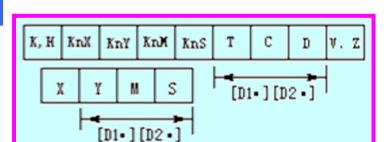




梯形图表达:

PLC送电运行时M500~M599, C235~C255成批复位

操作元件(其中2≤n≤ 512)



注: [D1·][D2·]指定同一元件, [D1·]号≤[D2·]号

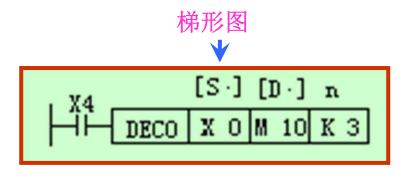
功能:

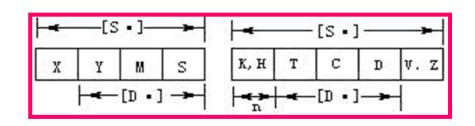
用于一批元件的同时复位。如控制程序的初始化处理等。

指令: DECO FNC41(P)(16/32)解码

程序步数: DECO,DECO(P)...7步

操作元件(其中n=1~8)



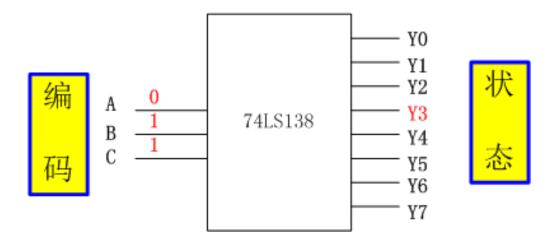


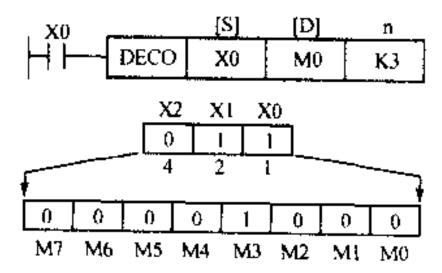
梯形图表达:

当X4=ON时, X2、X1、X0组成的二进制数等于几,就将M10~M17对应的M× 置1。若[D·]指定目标是T、C、或D,应使 n≤4。

功能:

[S·]指定的源元件和n指定的个数,组成二进制数等于几,就将[D·]指定的目标开始的8个寄存器中的对应位置的寄存器置1。





指令: ENCO FN42(P) (16)编码

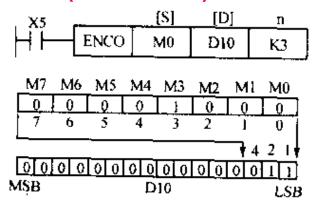
程序步数: ENCO, ENCO(P)...7步



梯形图表达:

当X5=ON时,M10~M17对应的M×为1。就将该位对应的二进制数送到D12、D11、D10组成的寄存器组内.若[S·]指定目标是T、C、D,V或Z,应使n≤4。若指定的源中为1的不止一处时,则只有最高位的1有效。若指定源中所有bit均为0,则出错。

操作元件(其中n=1~8)



功能:

[S·]指定的源元件开始的8个寄存器中为1的位,以二进制数输出到[D·]指定的目标和n指定的个数组成的寄存器组内。

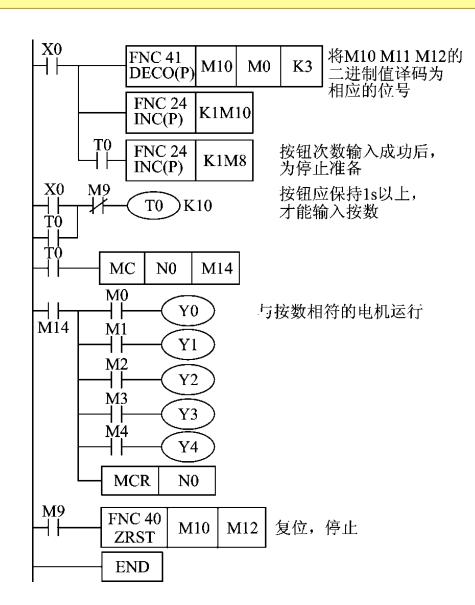
四、任务实施

1、控制系统程序分析:

- (1) 系统要求使用X0被按下的次数来表示需要起动电机的编号,即按下X0一下并保持1S,电机Y0启动;按下X0两下并保持1S,电机Y1启动;按下X0三下并保持1S,电机Y2起动;按下X0四下并保持1S,电机Y3启动;按下X0五下并保持1S,电机Y4启动。
 - (2) 使用K1M10来计按钮X0的次数,再通过解码指令将按钮的次数(即启动电机的序号)用M0,M1,M2,M3,M4分别区分出来。
- (3) 按下X0超过1S时间后, T0动作, 通过主控指令使M0-M4的状态 传送给输出,同时K1M8加1,即此时M9为1,使M0-M12所有数据清零,T0复位。

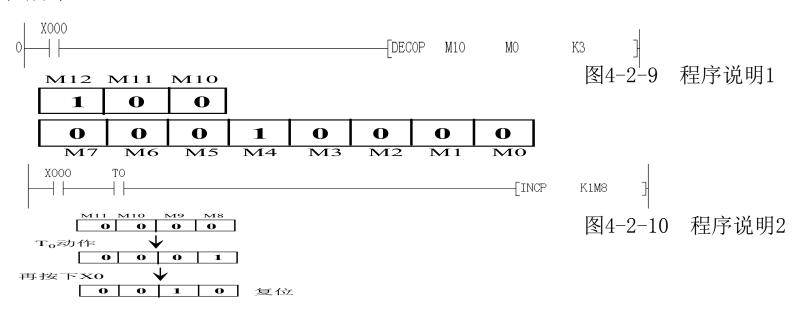
任务四用解码指令实现单按钮控制五台电动机的启停

三、任务实现



【程序说明】

1、按下X0,程序对M10解码,解码的结果放到M0中,同时 K1M10加1,若X0按下5次,则K1M10=0100,M4=1,如图4-2-9程序说 明1所示。



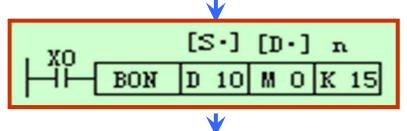
- 2、当最后一次按下按钮,并停留1S,T0动作,激活主控内程序,将解码的结果传送到输出端,若X0按下5次,M4=1,则Y4=1。
- 3、T0动作后的一个扫描周期, M8=1, 下一次重新选择电机启动, 按下X0后, M9=1定时器复位, M0-M12所有数据清零, 如图4-2-10程序说明2所示。

指令: BON FNC44(P) (16/32) ON位判别

程序步数: BON、BON(P)...3步

(D)BON、(D)BON(P)...13步

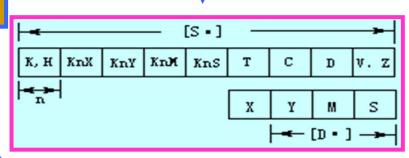




梯形图表达:

当X0为ON时,若D10中的第15bit为ON时,即是XO变为OFF,M0亦保持为1不变。

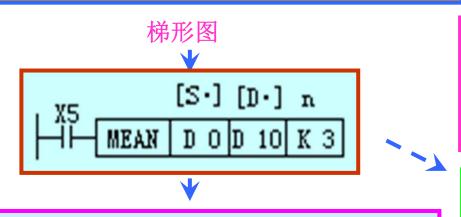
操作元件[其中n=0~15(16bit 操作)n=0~31(32bit操作)]



功能:

当条件为ON时,若[S·]指定的源数据中的第n指定的位为ON时,将[D·]指定的目标寄存器置1。即使条件变为OFF目标寄存器也保持不变。

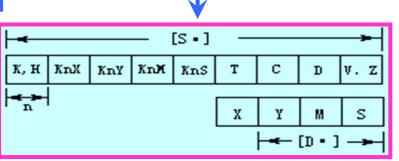
指令: MEAN FNC45(P) (16)平均值 程序步数: MEAN、MEAN(P)...3步



梯形图表达:

当**X5为ON时,将D0、D1、 D2**的平均值送入**D10**。

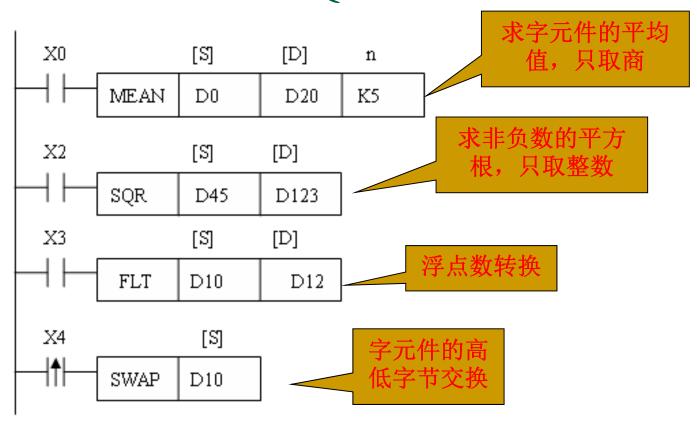
操作元件[其中n=1~64]



功能:

将[S·]指定的源开始的n个数据的平均值送入[D·]指定的目标。

其他指令MEAN、SQR、FLT、SWAP



4.3四则运算指令

分类	FNC NO.	指令助记符	功能说明	
算	20	ADD	二进制加法运算	
术与	21	SUB	二进制减法运算	
逻	22	MUL	二进制乘法运算	
辑 23 运		DIV	二进制除法运算	
算	24	INC	二进制加1运算	
	25	DEC	二进制减1运算	
	26	WAND	字逻辑与	
	27	WOR	字逻辑或	
	28	WXOR	字逻辑异或	
	29	NEG	求二进制补码	

任务五 自动售货机控制系统设计

一、任务引入

用PLC对自动售汽水机进行控制,工作要求如下:

- 1. 此售货机可投入0.5元、1元硬币,投币口为TB1,TB2。
- 2. 当投入的硬币总值大于等于6元时,汽水指示灯L1亮,此时按下汽水按钮SB1,则汽水从口CK1出汽水,12秒后自动停止。
- 3. 不找钱,不结余,下一位投币又重新开始。

请用PLC进行设计, A. 给出I/O分配表, 画出PLC I/O口的硬件接线图; B. 设计出梯形图并调试。

根据任务要求,如果想买到汽水,需要对投币数值进行计算,因此,要用到PLC的算术运算功能指令。

PLC算术运算指令包括ADD、SUB、MUL、DIV(二进制加、减、乘、除)以及INC和DEC指令,这些指令的名称、助记符、功能号、操作数见表

			操作数		
指令名称	助记符	功能号	[S1 •] [S2 •]	[D •]	
加法	ADD (P)	FNC20	K、H、KnX、KnY、 KnM、KnS、T、C、	KnY, KnM, KnS, T, C, D, V, Z	
减法	SUB (P)	FNC21	D, V, Z		
乘法	MUL (P)	FNC22	K、H、KnX、KnY、 KnM、KnS、T、C、	KnY、KnM、KnS、T、C、D	
除法	DIV (P)	FNC23	D, V, Z		

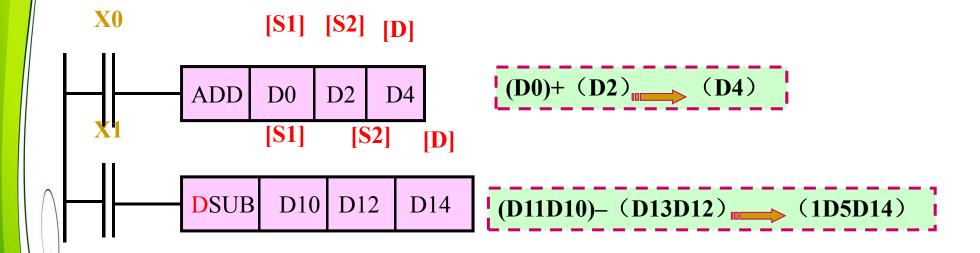
4.3四则运算指令

二进制加减运算指令

加法 FNC20 ADD

减法 FNC21 SUB

❖ 梯形图



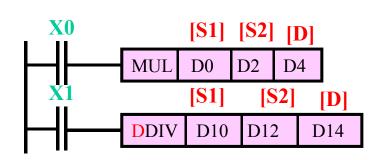
4.3四则运算指令

二、二进制乘除运算指令

乘法 FNC22 MUL

减法 FNC23 DIV

❖ 梯形图



```
被乘数 乘数 积 (D0)×(D2) → (D5D4)

被除数 除数 商 余数 (D11D10)÷ (D13D12) → (1D5D14) ··· (D17D16)
```

三、任务实施

一)任务要求

用PLC实现自动售汽水机的控制,工作要求如下:

- 1. 此售货机可投入0.5元、1元硬币,投币口为TB1,TB2。
- 2. 当投入的硬币总值大于等于 6 元时,汽水指示灯L1亮,此时按下汽水按钮 SB1,则汽水从口CK1出汽水,12秒后自动停止。
- 3. 不找钱,不结余,下一位投币又重新开始。

请用PLC进行设计, A. 给出I/0分配表, 画出PLC I/0口的硬件接线图;

B. 设计出梯形图并调试。

根据任务要求可知,该控制系统有4个输入,2个输出,因此,选用FX2N-16MT型号PLC即可满足控制要求。

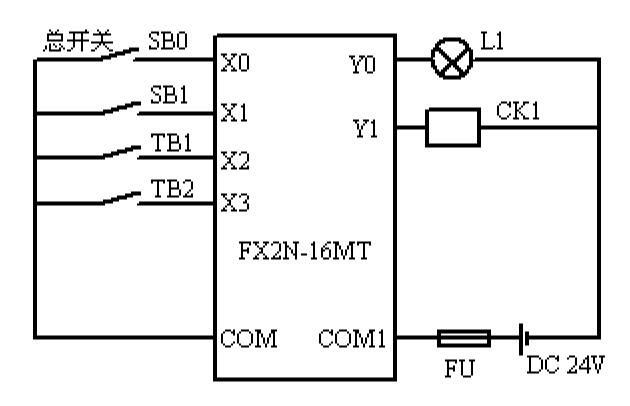
硬件I/0分配及接线

/0分配

通过分析任务要求知,该控制系统有4个输入,2个输出,因此,具体 I/0分配如表所示。

	输入		输 出		
输入继电器	输入元件	作用	输出继电器	输出元件	作用
X000	按钮SB0	总控制开关	Y000	L1	汽水指 示灯
X001	TB1	投币0.5元口	Y001	CK1	出汽水口
X002	TB2	投币1元口			
X003	按钮SB1	出汽水按钮			

2. 红硬件接线





程序设计

```
X000
             X001
                                                                       K12
                                                                                0.5元计数
              X002
                                                                        K6
    X000
              \dashv \uparrow \vdash
                                                                    -(C1
                                                                                1元计数
    X000
                                        -[MUL
K5
                                                          C0
                                                                    D0
    X000
                                        -[MUL
18
                                                 K1
                                                          C1
                                                                    D1
    X000
                                        -[DIV
26
                                                 DO
                                                          K10
                                                                    D2
                                                                              ∄换算投0.5元钱数
    X000
                                                                                投币总值
                                        -[ADD
34
                                                 D1
                                                          D2
                                                                    D3
    X000
                                                                    000Y)-
                    D3
                             K6
    X003
                                                          SET
                                                                    MO
    X000
             Y000
                        ΜO
                                                                    Y001
54
                                                                       K120
                                                                    -(T0
    X000
              T0
                                                  -[ZRST
                                                          C0
                                                                    C1
                                                          -[RST
                                                                    MO
                                                                    -END
69
```



二、二进制加1减1指令

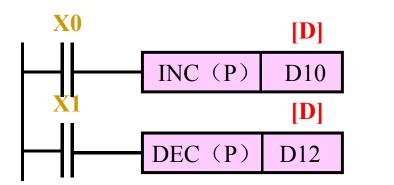
加1 FNC24 INC

减1 FNC25 DEC

❖ 操作数

[D]: KnY, KnM, KnS, T, C, D, V, Z

❖ 梯形图



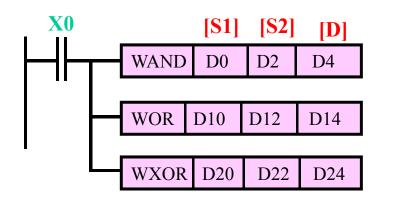
$$(D10)+1 \longrightarrow (D10)$$

$$(D12)-1 \longrightarrow (D12)$$

逻辑与 FNC26 WAND

逻辑或 FNC27 WOR

逻辑异或 FNC28 WXOR

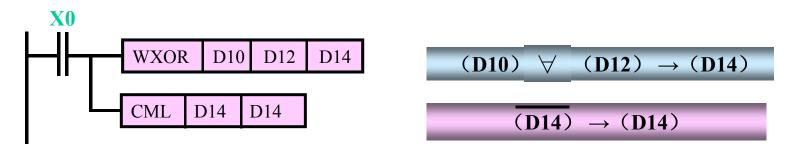


$$(\mathbf{D0}) \ \land \ (\mathbf{D2}) \ \rightarrow (\mathbf{D4})$$

$$(D10) \lor (D12) \rightarrow (D14)$$

$$(D20) \quad \forall \quad (D22) \quad \rightarrow \quad (D24)$$

❖ 异或非运算的梯形图



求补指令 FNC19 NEG



❖ 说明

- ➤ 求补指令是把二进制数各位取反再加1后,送入目标操作数[D]中。实际是绝对值不变的变号操作。
- ➤ PLC的负数以二进制的补码形式表示,其绝对值可以通过求补指令求得。

4.5 程序流控制

一、条件跳转指令 FNC00 CJ

二、子程序指令 FNC 01 CALL

FNC 02 SRET

三、中断指令 FNC 03 IRET

FNC 04 EI

FNC 05 DI

四、主程序结束指令 FNC 06 FEND

五、警戒时钟定时器指令 FNC 07 WDT

六、循环指令 FNC 08 FOR

FNC 09 NEXT

任务六 声光报警控制系统设计

一、任务引入

有一个声光报警控制系统,报警系统起动之后,灯闪,亮0.5s,灭0.5s,蜂鸣器响。灯闪烁30次之后,灯灭,蜂鸣器停,间歇5s。如此进行3次,自动熄灭。

此任务里有重复的动作,即灯闪、蜂鸣器响是反复进行的,如果将重复的动作编写成子程序,在主程序里通过调用子程序的方法实现控制系统设计,程序的结构将会很清楚。因此,编程中将用到PLC的程序流转控制类指令。

二、相关知识

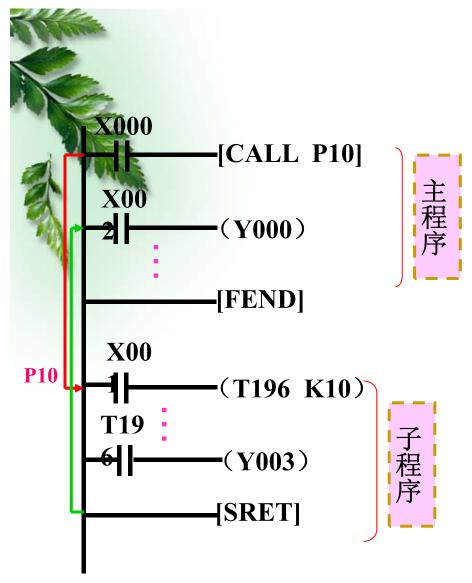
一) 子程序调用指令CALL和子程序返回指令SRET

调用指令的助记符、指令代码、操作数、程序步见表所示

指令名称	助记符	功能号	操作数	程序步	
1日 〈 石 47	197 1611	列配す	[D •]	1主/ 1/20	
子程序调用	CALL (P)	FNC01	指针P0~P127嵌套5级	3步;指令标号1步;	
子程序返回	SRET	FNC02	无	1步	

子程序是为一些特定的控制要求编制的相对独立的程序。为了区别于主程序,规定在程序编排时,将主程序排在前边,子程序排在后边,并以主程序结束指令FEND(FNC06)将这两部分分隔开。

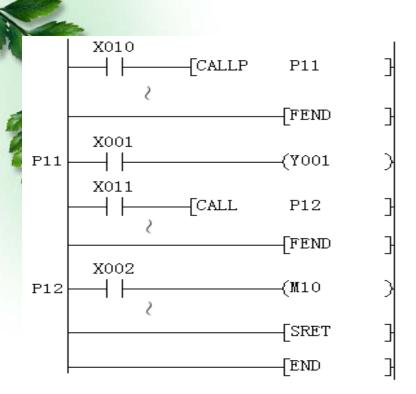
子程序调用指令在梯形图中使用的情况如下图所示。图中,如果X0接通,则转到标号P10处去执行子程序。当执行SRET指令时,返回到CALL指令的下一步执行。



子程序指令

子程序调用指令 FNC01 CALL 操作数: 指针P0~P127

- 子程序返回指令 FNCO2 SRET 无操 作数
- ❖ 说明
 - > 子程序应该在主程序结束之后编程。
 - ▶CJ指令的指针与CALL的指针不能重复。
 - ▶子程序允许嵌套,嵌套级别最多为 5级。
 - ▶子程序只能用T192~T199和T246~ T249作定时器。



上图是一级嵌套的例子。子程序P11 是脉冲执行方式,即X010置1一次, 子程序P11只执行一次。当子程序 P11开始执行并X011置1时,程序转 去执行子程序P12,当P12执行完毕 后又回到P11原断点处执行P11。直 到P11执行完成后返回主程序。 使用子程序调用与返回指令时应注意 如下3点。

- 1. 转移标号不能重复,也不可与跳转 指令CJ的标号重复;
- 2. 主程序在前,子程序在后,即子程序一定要放在FEND指令之后。不同位置的"CALL"指令可以调用相同标号的子程序,但同一标号的指针只能使用一次;
- 3. 子程序可以调用下一级子程序,成为子程序嵌套,最多可5级嵌套。当有多个子程序排列在一起时,标号和最近的一个子程序返回指令构成一个子程序。如左图所示。
- 4. 子程序只能用T192~T199和T246~ T249作定时器。

(二) 主程序结束指令FEND

主程序结束指令FEND无操作数。

FEND表示主程序结束,当执行到FEND时,PLC进行输入/输出处理,监视定时器刷新,完成后返回起始步。

END是指整个程序(包括主程序和子程序)结束。

一个完整的程序可以没有子程序,但一定要有主程序。

使用FEND指令时应注意:

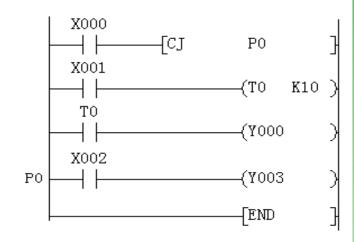
- (1) 子程序和中断服务程序应放在FEND之后。
- (2) 子程序和中断服务程序必须写在FEND和END之间,否则出错。

三)条件跳转指令CJ

条件跳转指令的助记符、指令代码、操作数、程序步见表所示。

指令名称	助记符	功能号	操作数 [D•]	程序步
条件跳转	CJ (P)	FNC00	指针P0~ P127P63即END	CJ (P) 3步; 指 令标号1步;

条件跳转指令用来选择执行指定的程序段,跳过 智时不需要执行的程序段。如右图所示,当X0接通 时,是由"CJ PO"指令跳到标号为PO的指令处开始 执行,跳过了程序的一部分,减少了扫描周期。如 果X0断开,跳转不会执行,则程序按原顺序执行。



使用跳转指令应注意如下几点:

- 1. 条件跳转指令CJ(P)的操作数为指针标号P0~P127, P用于分支和跳转程序;
- 2. 标号P放置在左母线的左边,在一个程序中一个标号只能出现一次,可以有多条跳转指令使用同一标号,但不允许一个跳转指令对应两个标号的情况,即在同一程序中不允许存在二个相同的标号;
- 3. 若跳转条件满足,则执行跳转指令,程序跳到以标号P为入口的程序段中执行,否则不执行跳转指令,按顺序执行下一条指令;
 - 4. 不在同一个指针标号的程序段中出现的同一个线圈不被看作双线圈;

- 5. 使用CJ(P)指令时,跳转只执行一个扫描周期,但若用辅助继电器M8000 作为跳转指令的工作条件,跳转就成为无条件跳转;
- 6. 在跳转执行期间,即使被跳过程序段的驱动条件改变,但其线圈(或结果) 仍保持跳转前的状态;
- 7. 如果在跳转开始时定时器和计数器已在工作,则在跳转执行期间它们将停止工作,到跳转条件不满足后又继续工作。对于正在工作的定时器T192~T199和高速计数器C235~C255,不管有无跳转仍连续工作;
- 8. 定时器、计数器的复位指令具有优先权,即使复位指令位于被跳过的程序段中,当执行条件满足时,复位工作也将被执行;
- 9. P63是END所在的步序,在程序中不需要设置P63。指针标号允许用变址寄存器修改,在编写跳转程序的指令表时,标号需占用一行。

四)条件跳转指令应用实例

1. 控制要求

某台设备具有手动/自动两种工作方式,SB3是工作方式选择开关,当SB3处于断开状态时,选择手动工作方式;当SB3处于接通方式时,选择自动工作方式,不同工作方式进程如下。

- (1) 手动方式:按下起动按钮SB2,电动机旋转;按停止按钮SB1,电动机停止。
- (2) 自动方式:按下起动按钮SB2,电动机连续运转1min后,自动停机,按停止按钮SB1,电动机立即停机。

2. 确定输入/输出并分配I/0地址

通过分析任务,各元件的I/0分配和作用如表所示。

	输入		输出			
输入继电器	输入元件	作用	输出继电器	输出元件	作用	
X000	FR	过载保护	Y000	KM	控制电机	
X001	按钮SB1	停止按钮				
X002	按钮SB2	起动按钮				
X003	按钮SB3	方式选择				

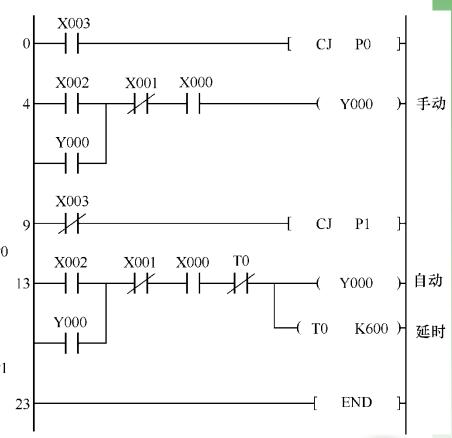
3. 程序设计

根据控制要求,设计的程序如右图所示。

程序说明如下:

- (1) 手动工作方式。当SB3处于断开状态时,X3的常开触点断开,不执行"CJ P0"指令,而顺序执行程序步4~步8的手动程序段。此时,因X3的常闭触点闭合,执行"CJ P1"指令,跳过自动工作方式程序段到结束指令语句;
- (2) 自动工作方式。当SB3处于接通状态时,X3的常开触点闭合,执行"CJ P0"指令,跳过程序步4~步12手户动程序段,执行步13~步23的自动程序段,然后顺序执行结束指令。

由于手动程序和自动程序不能同时 执行,所以程序中的线圈Y0不能视为双 线圈。



三、任务实施

(一) 任务要求

有一个声光报警控制系统,报警系统起动之后,灯闪,亮0.5s,灭0.5s,蜂鸣器响。灯闪烁30次之后,灯灭,蜂鸣器停,间歇5s。如此进行3次,自动熄灭。

编写程序时,可以将重复的动作,灯闪、蜂鸣器响作为子程序,放在FEND之后。而调用子程序CALL放在主程序之中。

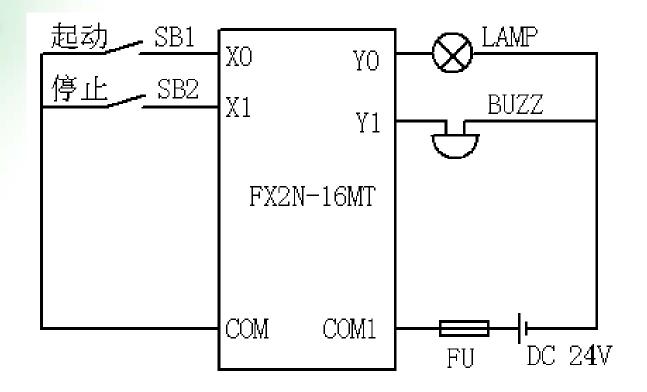
硬件I/0分配及接线

1.1/0分配

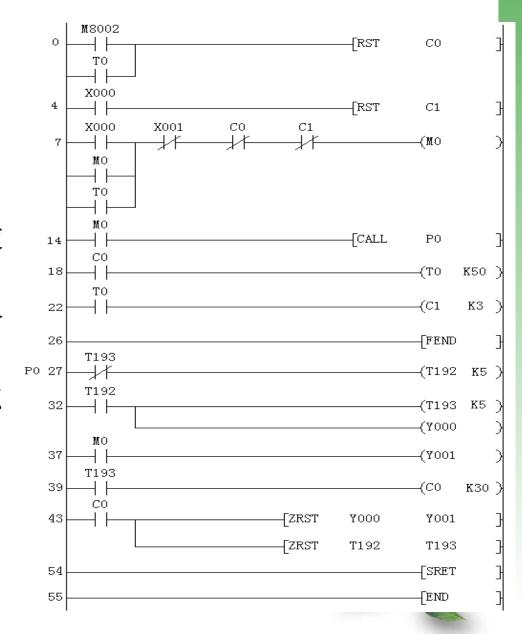
通过分析任务,各元件的I/0分配和作用如表所示。

	输入		输出		
输入继电器	输入元件	作用	输出继电器	输出元件	作用
X000	按钮SB1	起动按钮	Y000	LAMP	灯
X001	按钮SB2	停止按钮	Y001	BUZZ	蜂鸣 器





程序设计 图中,步0~步26为主程序, 少步54为子程序。主程序中, MO是调用子程序的控制触点,每次 接通MO,调用子程序一次,本任务 共调用3次。注意: 在子程序中,使 用了T192、T193定时器,这种定时 器在执行线圈指令时或执行END指令 时计时。如果计时达到设定值,则 执行线圈指令或END指令,输出触点 动作。因此当子程序执行到SRET返 回到步18执行之后,Y0、Y1仍为ON, 不停止。为此,设置了成批复位指 令ZRST指令,使YO、Y1失电之后再 返回步18执行。



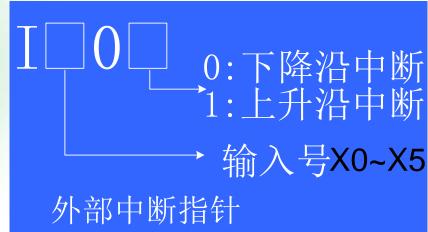
三、中断指令

说明

- > 在执行某个中断子程序时,禁止其它中断请求。
- >中断程序允许嵌套,嵌套级别为2级。
- ➤ 中断指针共有15个:输入中断6个,定时器中断3个, 计数器中断6个。
- ▶中断程序用T192~T199和T246~T249作定时器。
- > 中断的优先级别
 - ✓多个中断信号不同时产生时,按先后顺序中断。
 - ✓多个中断信号同时产生时,按指针大小中断。

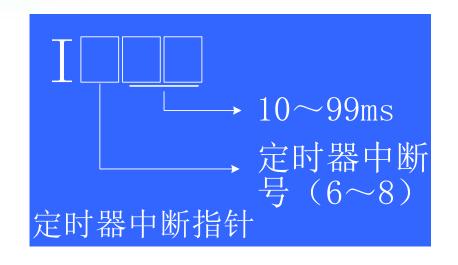


(2) 中断指针



例: I000

I101





■中断指针应放在FEND指令之后。



(2) 中断的指针

- 1) 当M8050~M8058为ON时,禁止执行相应I0□□~ I8□□的中断,M8059为ON时则禁止所有计数器中断:
- 3) 无需中断禁止时,可只用EI指令,不必用DI指令;
- 4) 执行一个中断服务程序时,如果在中断服务程序中有 EI和DI,可实现二级中断嵌套,否则禁止其它中断。

	输入编号	指针编号		禁止中断指令
THE PARTY OF		上升中断	下降中断	
	X000	1001	1000	M8050
	X001	I101	I100	M8051
	X002	I201	1200	M8052
	X003	I301	1300	M8053
	X004	I401	1400	M8054
	X005	I501	1500	M8055

	禁止中断指令
定时中断 16**	M8056
定时中断 I7**	M8057
定时中断 18**	M8058
所以计数器中断	M8059

例6-1 在X0的上升沿通过中断使Y0立即变为ON,在X1的下降沿通过中断使Y0立即OFF,编写中断程序。

//主程序	
EI	//允许中断
FEND	//主程序结束
//中断子程序	
I001	//X0上升沿中断程序
LD M8000	//M8000一直为ON
SET Y0	//Y0被置位
REF Y0 K8	//Y0~Y7被立即on
IRET	//中断结束标志
I100	//X1下降沿中断程序
LD M8000	
RST Y0	//Y0被复位
REF Y0 K8	//Y0~Y7被立即off
IRET	//中断程序结束
END	

例6-2 用定时器中断,每1s将Y0~Y7组成的8位二进制数加1。

LD M8002	//首次扫描
RST D0	//复位中断次数计数器
EI	//允许中断
FEND	
//中断子程序	
1650	//50ms定时中断
LD M8000	//M8000一直为ON
INC D0	//中断次数计数器加1
LD= K20 D0	//如果如果中断了20次
INC K2Y0	//K2Y0加1
RST D0	//复位D0
IRET	//中断返回
END	

例子6-3,从X0的上升沿开始,用定时器中断使D0的值每隔10ms加1,在100s内由0线性增大到10000,大于100s时D0值保持不变。

EI //允许中断

■ LDP X0 // X0的上升沿

ANI M8002

■ //禁止X0在第一个扫描周期自动出现的上升沿起作用,不是一定要

RST M8056

FEND

■ I610 //每隔10ms中断1次

LD M8000

■ INC D0

■ LD= K10000 D0//100S到

SET M8056

SET Y0

IRET

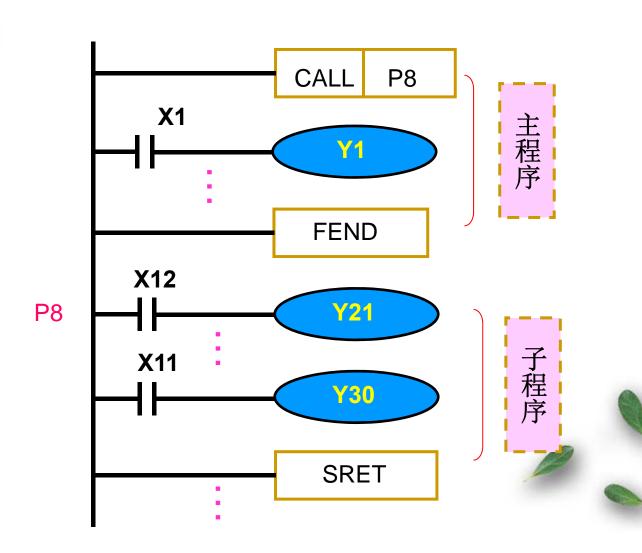
END



主程序结束指令 FNC06 FEND

无操作数

❖ 梯形图



四、主程序结束指令 FNC06 FEND

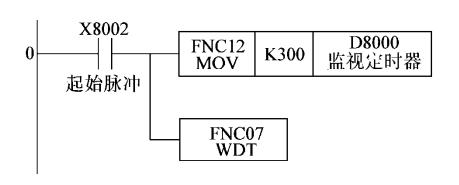
说明

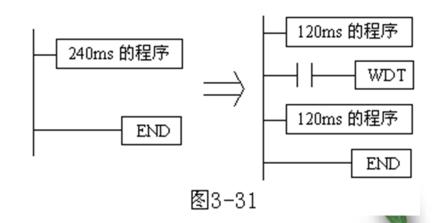
- >与END指令的功能一样,执行到该指令时程序返回到 0步。
- ▶中断服务子程序和子程序应该写在FEND之后,并且 用IRET和SRET返回。
- ➤ 如果多次使用FEND指令,在最后的FEND和END之间编写子程序或中断子程序

五、监视定时器指令

■监视定时器指令WDT(P)

监视定时器缺省值为200ms(可用D8000来设定)





六、循环指令

循环开始 FNC08 FOR

❖ 操作数

[S]: K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z

- ❖ 循环结束 FNC09 NEXT 无操作数
- ❖ 说明
 - ▶ n为循环次数,其范围为1~32767有效。如果指定为-32768~0,则作n=1处理。
 - ▶ 循环指令最多可以嵌套5级。
 - ▶程序中FOR-NEXT是成对出现的,FOR在前,NEXT在后不可倒置,否则出错。
 - > 编程时NEXT应该在FEND或END之前,否则出错。

六、循环指令

例子在X1上升沿,将50、55、60...90分别送D10~D18

LDP X1

RST Z0

MOV K50 D10

FOR K8

ADD D10Z0 K5 D11Z0

INC Z0

NEXT

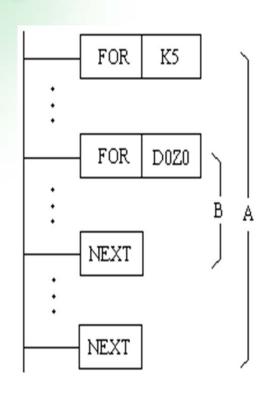


图3-32



■ **例6** 用两个开关X1, X0控制一个信号灯Y0, 当X1, X0=00时灯灭, X1, X0=01时灯以1秒脉冲闪, X1, X0=10时灯以2秒脉冲闪, X1, X0=11时灯常亮。

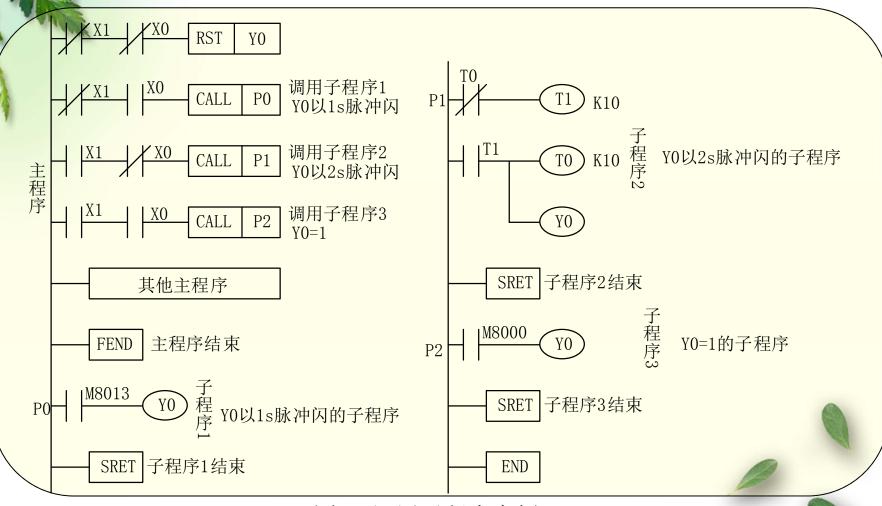
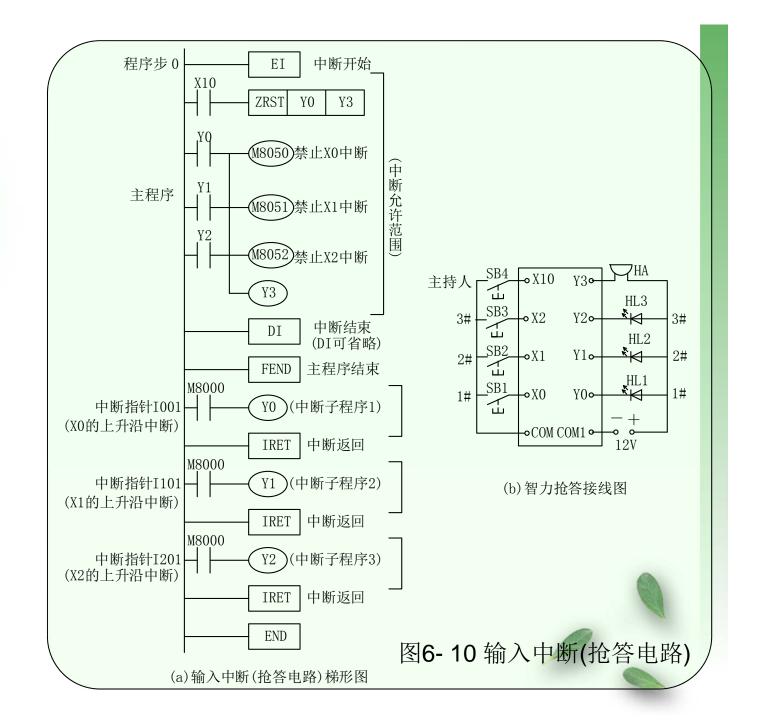


图6- 调用子程序实例





列6-4 定时器中断用于斜波信号

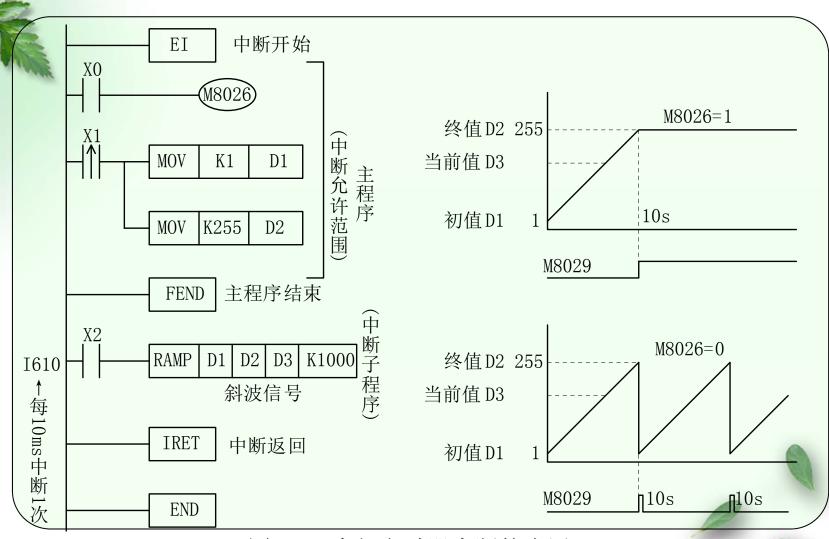


图6-11 内部定时器中断的应用

4.6高速处理指令

- (1)刷新REF
- (2) 刷新和滤波时间调整REFF
- (3)矩阵输入MTR
- (4) 高速计数器置位HSCS
- (5) 高速计数器复位HSCR
- (6) 高速计数器比较HSZ
- (7) 速度检测SPD
- (8) 脉冲输出PLSY
- (9) 脉宽调制PWM