

上海大学自学考试

数控技术及应用  
实验指导

上海大学

2009年10月10日

# 实验一 步进电动机基本原理实验

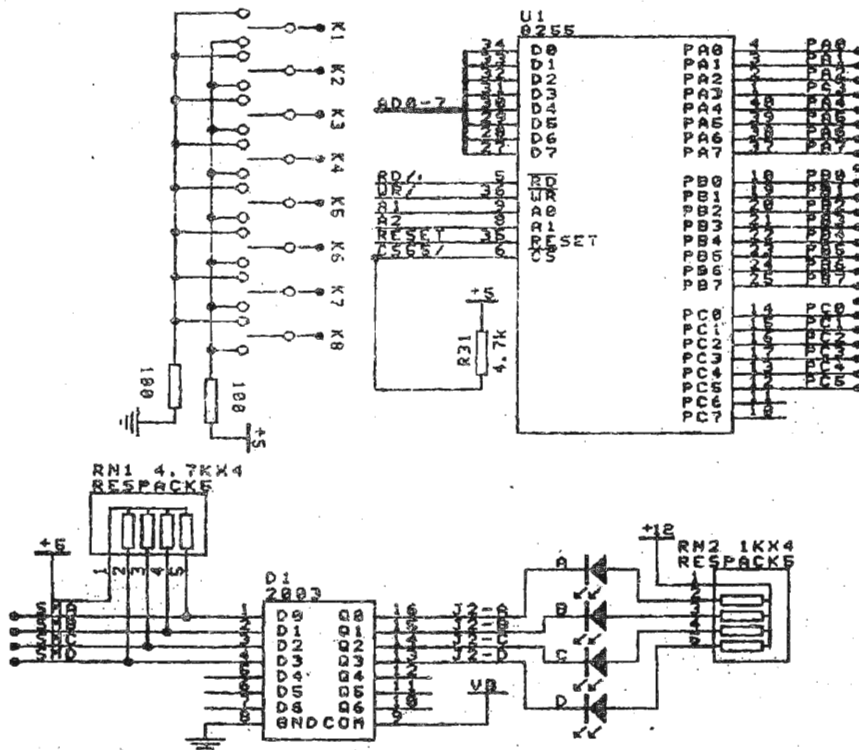
## 一、实验要求

- 1、编制程序，利用单片机 P1 口产生驱动步进电动机运转的脉冲信号（四相单四拍、四相双四拍、四相八拍）。
- 2、编制程序，实现步进电动机自锁功能。
- 3、编制程序，演示步进电动机的步距角。

## 二、实验目的

- 1、了解步进电动机的基本结构和工作原理。
- 2、掌握步进电动机驱动程序的设计方法。

## 三、实验电路及连线



机电平台的选择开关打在步进电机位置。实验机并行输出模块 273 的 PO4 接+5V。

8255的PC0-PC3接K1-K4, CS55/接8000H, 8031/32的P1.0-P1.3接SMA-SMD.

#### 四、实验说明

步进电动机又称为脉冲电机,是工业过程控制和仪表中一种能够快速启动、反转和制动的执行元件。其功用是将电脉冲转换为相应的角位移或直线位移。步进电动机的运转是由电脉冲信号控制的,步进电动机的角位移量或线位移量与脉冲数成正比,每给一个脉冲,步进电机就转动一个角度(步距角)或前进/倒退一步。步进电机旋转的角度由输入的电脉冲数确定,所以,也有人称步进电动机为一个数字/角度转换器。

当某一相绕组通电时,对应的磁极产生磁场,并与转子形成磁路,这时,如果定子和转子的小齿没有对齐,在磁场的作用下,由于磁通具有力图走磁阻最小路径的特点,转子将转动一定的角度,使转子与定子的齿相互对齐,由此可见,错齿是促使电机旋转的原因。

四相步进电动机以四相单四拍、四相双四拍、四相八拍方式工作时的脉冲分配表如表1,表2和表3

表1 四相单四拍脉冲分配表

	A	B	C	D
N	1	0	0	0
N+1	0	1	0	0
N+2	0	0	1	0
N+3	0	0	0	1

表2 四相双四拍脉冲分配表

	A	B	C	D
N	1	1	0	0
N+1	0	1	1	0
N+2	0	0	1	1
N+3	1	0	0	1

表3 四相八拍脉冲分配表

	A	B	C	D
N	1	0	0	0
N+1	1	1	0	0
N+2	0	1	0	0
N+3	0	1	1	0
N+4	0	0	1	0
N+5	0	0	1	1
N+6	0	0	0	1
N+7	1	0	0	1

如步进电动机每一相均停止通电,则电机处于自由状态;若某一相一直通直流电时,则电机可以保持在固定的位置上,即停在最后一个脉冲控制的角位移的终点位置上,这样,步进电动机可以实现停车时转子定位。这就是步进电动机的自锁功能。当步进电机处于自锁时,若用手旋转它,感觉很难转动。

本实验用拨动开关K1-K4来确定步进电动机的工作方式(表4)。

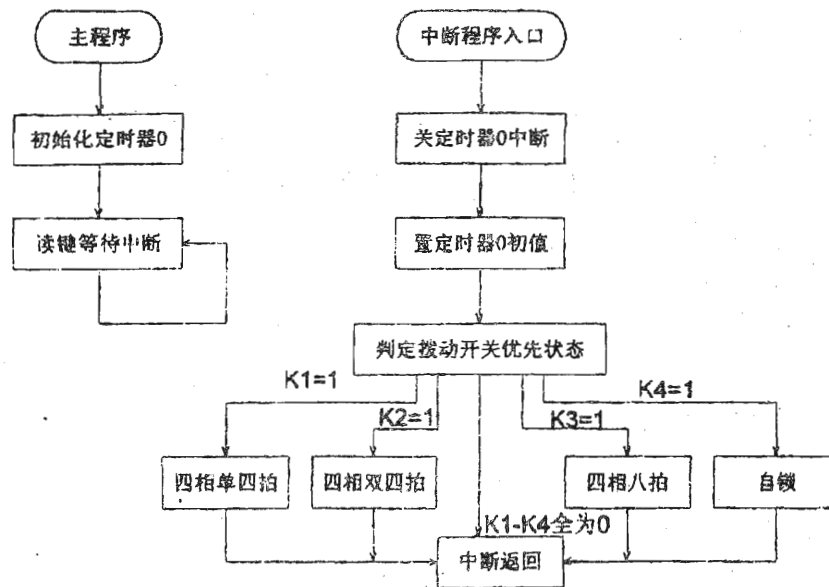
表4 步进电动机工作方式(X代表任意逻辑)

	K4	K3	K2	K1
四相单四拍	X	X	X	1
四相双四拍	X	X	1	0
四相八拍	X	1	0	0
锁定功能	1	0	0	0

## 五、实验参考程序框图

实验参考程序见盘片中的 Expl.asm

参考程序框图



见附录二

## 实验二 步进电动机速度调节和方向控制实验

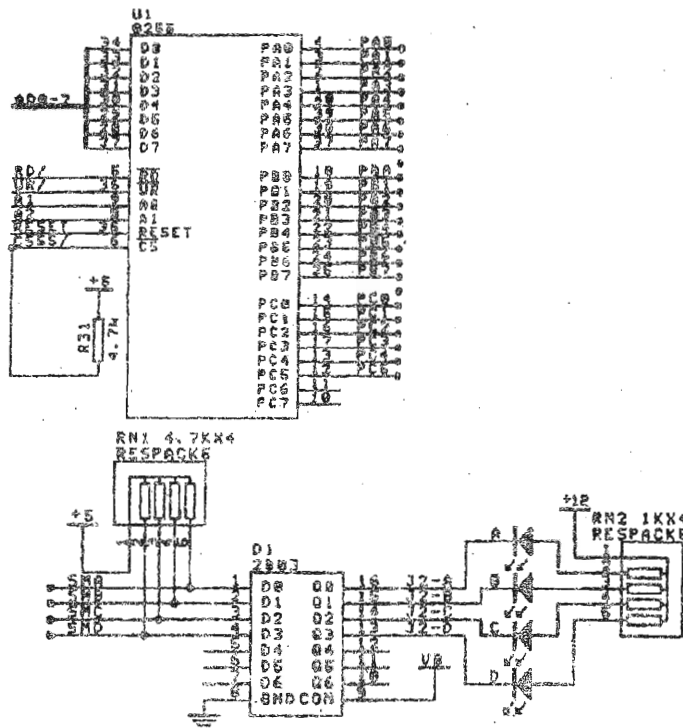
### 一、实验要求

- 1、编制程序，控制步进电动机的运转速度。
- 2、编制程序，控制步进电动机的旋转方向。

### 二、实验目的

- 1、掌握步进电动机控制系统的硬件设计方法。
- 2、掌握步进电动机速度调节、方向控制技术。
- 3、进一步学习编制步进电动机驱动程序的软件设计方法。

### 三、实验电路及连线



机电平台的选择开关打在步进电机位置。实验机并行输出模块 273 的 PC4 接+5V。8255 的 PA0-PA7 接至 K1-K8，PC0-PC3 接 SMA-SMD，CS55/接 8400H。

### 四、实验说明

步进电动机是用电脉冲进行控制的电机，改变脉冲输入频率，就可以改变电机的速度；改变通电顺序，即改变定子磁场旋转的方向，就可以达到控制步进电动机正反转目的。

#### 1、运行速度的控制

从图 1 可以看出，当改变电脉冲的周期，ABCD 四相绕组高低电平的宽度将发生变化，这就导致通电和断电的变化率发生变化，使电机转速改变，所以调节电脉冲的周期就可以控制步进电机的运转速度。

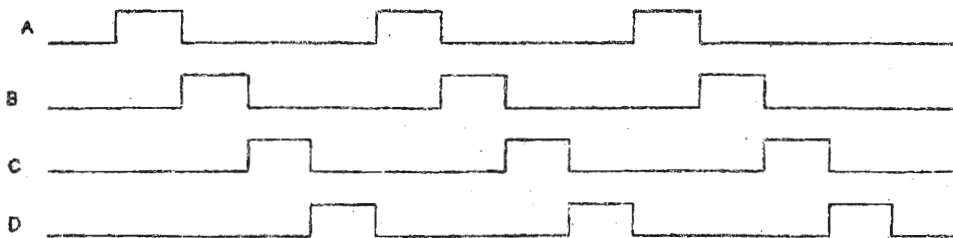


图 1

## 2、运转方向的控制

从图 1 仍可看出，步进电机以四相单四拍方式工作时，按 A→B→C→D→A 次序通电时，为正转；如按 A→D→C→B→A 次序通电时，为反转。

## 3、拨动开关 K1-K8 状态

本实验通过拨动开关 K1-K4 确定步进电动机的转速和 K8 控制步进电动机的运转方向（表 1）

表 1 拨动开关状态表

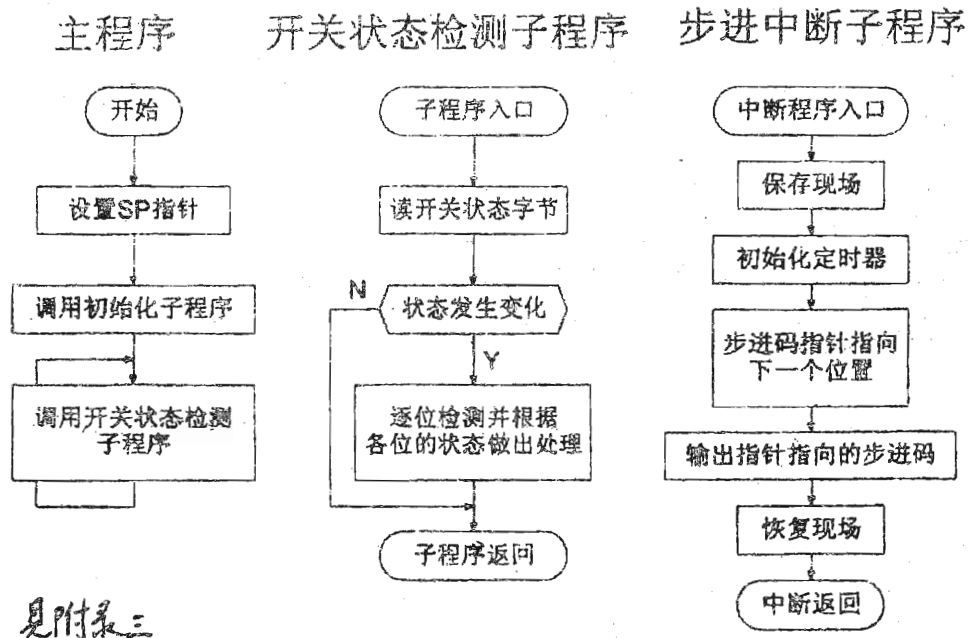
速度等级	K1	K2	K3	K4	正转	反转
					K8	K8
0	0	0	0	0	0	1
1	1	X	X	X		
2	0	1	X	X		
3	0	0	1	X		
4	0	0	0	1		

注：X 代表任意逻辑电平

## 五、实验参考程序框图

实验参考程序见盘片中的 Exp2.asm

参考程序框图



## 实验三 逐点比较法的直线插补软件设计

### 一、实验目的

1. 掌握逐点比较法的直线插补原理。
2. 了解直线插补的软件编程方法。

### 二、实验原理

在直线或圆弧廓形加工中，需要 $x$ 向和 $y$ 向驱动电机同时动作，合成所需的运动轨迹。由于一个进给脉冲只能沿坐标轴（ $x$ 轴或 $y$ 轴）进给一步，这个距离（步距）称为脉冲当量。因此，进给机构的运动轨迹是由一段段沿坐标轴的直线段所组成，不能与所要求的理想直线或其它曲线完全重合。根据加工直线或曲线确定出进给机构运动轨迹的过程叫做插补。

逐点比较插补法的基本原理是：进给机构每进给一步，计算新的坐标点位置和它与理想直线（或曲线）的偏差，根据偏差的正、负号来确定下一步 $x$ 或 $y$ 向电机的走向，使进给机构向减少偏差的方向进给。

设要插补图4.10中的直线 $\bar{OA}$ ，坐标系为 $\bar{o}-xy$ ，坐标原点位于被插补直线的起点。直线终点为 $A(x_0, y_0)$ 。

#### 1. 偏差计算

设有一点 $M(x, y)$ 若点 $M$ 在直线上则有

$$y/x = y_0/x_0 \quad \text{或} \quad yx_0 - xy_0 = 0$$

若点 $M$ 不在直线上，则 $yx_0 - xy_0 \neq 0$ ，令偏差

$$F(x, y) = yx_0 - xy_0 \quad (4.1)$$

若点 $M(x, y)$ 在直线的上方，则有

$$\frac{y}{x} > \frac{y_0}{x_0} \quad \text{即} \quad yx_0 > xy_0$$

则  $F(x, y) = yx_0 - xy_0 > 0$

若点 $M(x, y)$ 在直线的下方，则有

$$\frac{y}{x} < \frac{y_0}{x_0} \quad \text{即} \quad yx_0 < xy_0$$

则  $F(x, y) = yx_0 - xy_0 < 0$

当点 $M(x, y)$ 在直线上时，则

$$F(x, y) = 0$$

如果在汇编语言程序中直接用式(4.1)，则计算偏差时要用到乘法，给编程带来困难，所以常采用递推算法。

图4.10所示的第I象限直线插补进给方向是 $+x$ 和 $+y$ 。当向 $+x$ 方向进给一步后，新偏差与原偏差的关系为

$$\begin{aligned} F(x+1, y) &= yx_0 - (x+1)y_0 = yx_0 - xy_0 - y_0 \\ &= F(x, y) - y_0 \end{aligned} \quad (4.2)$$

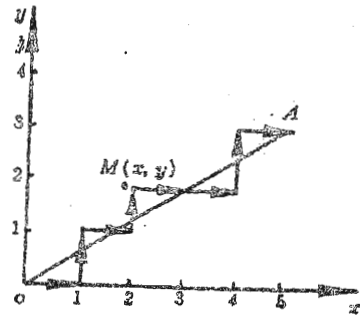


图4.10 被插补直线

向+y方向发一个进给脉冲后

$$F(x, y+1) = (y+1)x_0 - xy_0 = yx_0 - xy_0 + x_0 \\ = F(x, y) + x_0 \quad (4.3)$$

## 2. 进给脉冲的分配

脉冲分配从坐标原点开始, 这时 $F(0, 0) = 0$ , 规定先向+x方向发一个进给脉冲, 以后就用偏差判别。

若 $F(x, y) \geq 0$ , 说明当前的插补点在直线的上方或在直线上, 应向+x方向发出进给脉冲; 若 $F(x, y) < 0$ , 说明当前插补点在直线的下方, 应向+y方向发一个进给脉冲。

## 3. 终点判别

插补过程中, 每走一步达到新的一插补点都要与插补的理想直线进行比较, 看是否到达终点。若未到终点, 就应继续插补, 否则停止插补运算。

终点判别方法可用一个方向的坐标值来判断。若 $x_0 > y_0$ , 就用x方向的走步总数 $x_0$ 作为终点判断的依据, 看x是否等于 $x_0$ , 否则用y方向的走步总数 $y_0$ 作为终点判据。

## 4. 插补程序设计

第I象限直线插补程序算法如下:

(1) 判别 $F \geq 0$ 还是 $F < 0$ 。

(2) 若 $F \geq 0$ , 则向+x方向发一个脉冲, 计算新的偏差 $F(x+1, y) = F(x, y) - y_0$ , 并修改流动坐标 $x \leftarrow x+1$ ; 若 $F < 0$ , 则向+y方向发一个脉冲, 计算新偏差 $F(x, y+1) = F(x, y) + x_0$ , 并修改流动坐标 $y \leftarrow y+1$ 。

(3) 终点判别: 若 $x = x_0$  (或 $y = y_0$ ), 则停止插补运算, 否则转入判别 $F \geq 0$ 还是 $F < 0$ , 重复上述步骤。

第I象限直线插补程序框图见图4.11。

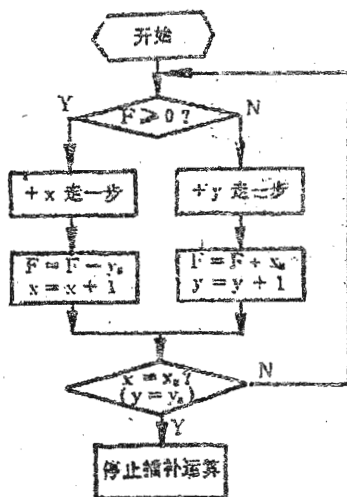


图4.11 直线插补程序框图

在插补程序中, 对上述经常要用到的数据, 例如偏差 $F$ , 流动坐标值 $x$ 和 $y$ 及终点坐标值 $x_0$ 和 $y_0$ , 都要设定相应的寄存器或内存单元, 这些工作单元的字节数与直线最大尺寸有关。考虑要有符号位, 工作单元所能存储的最大数是 $2^{n-1}-1$ , 其中 $n$ 是工作单元的位数。

## 三、实验器材

MCS51单片机开发装置或者TP801单板机1台, 步进电机功率放大器(NCL-1型)1台, x-y步进电机驱动绘图仪(NCL-1型)1台, 直流稳压电源(NCL-1型)1台, 磁带记录仪1台, 导线若干。



# 附录一 使用MCS51单片机开发装置时的参考程序①

## 一、环行分配器软件的汇编语言程序

### 1. 程序框图(图4.19)

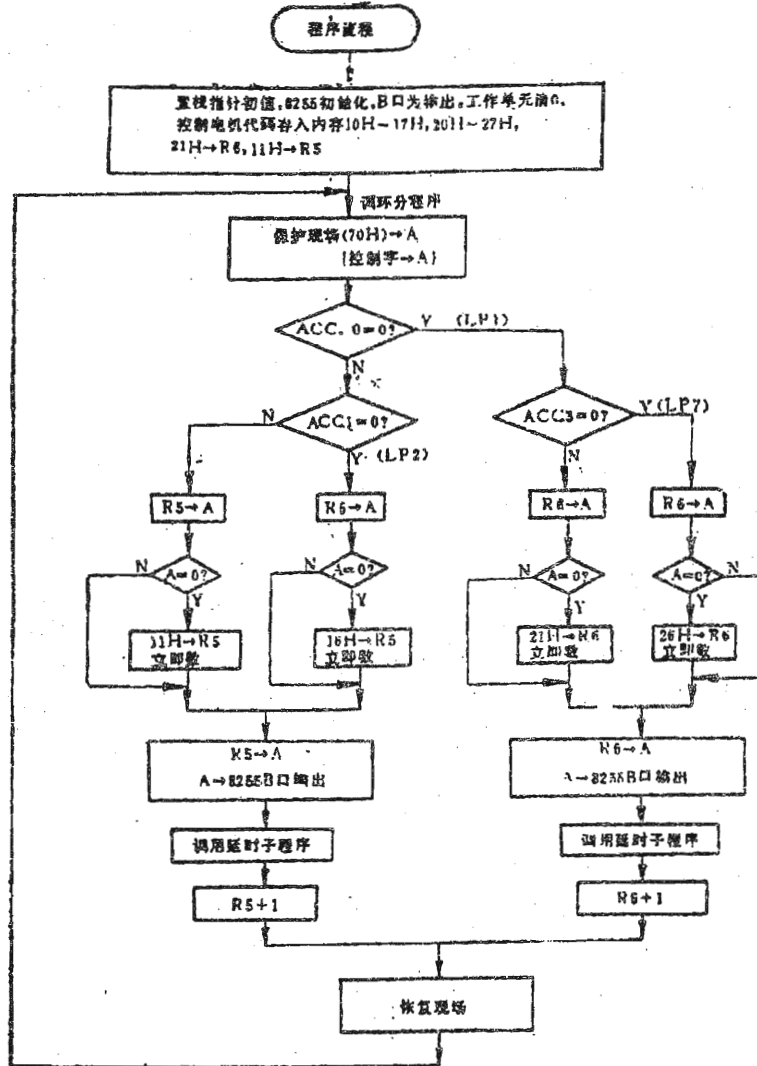


图4.19 环行分配器程序框图

## 2. 实验参考程序

		ORG	2000H
2000	758160	MAIN, MOV	SP, *60H ; 主程序
2003	78EB	MOV	R0, *0EBH, 置8255B口输出
2005	7480	MOV	A, *80H
2007	F2	MOVX	@R0, A
2008	122050	LCALL	DSUP ; 调装状态码子程序
200B	7D11	MOV	R5, *11H
200D	7E21	MOV	R6, *21H
200F	122100	LPO, LCALL	HXFPQ ; 调环行分配器子程序
2012	010F	AJMP	LPO
		ORG	2050H
2050	7920	DSUP, MOV	R1, *20H, 装代码子程序 DSUP
2052	7810	MOV	R0, *10H
2054	E4	CLR	A
2055	F6	TOPI, MOV	@R0, A
2056	08	INC	R0
2057	D9FC	DJNZ	R1, TOPI
2059	7810	MOV	R0, *10H
205B	7600	MOV	@R0, *00H
205D	08	INC	R0
205E	7601	MOV	@R0, *01H
2060	08	INC	R0
2061	7603	MOV	@R0, *03H
2063	08	INC	R0
2064	7602	MOV	@R0, *02H
2066	08	INC	R0
2067	7606	MOV	@R0, *06H
2069	08	INC	R0
206A	7604	MOV	@R0, *04H
206C	08	INC	R0
206D	7605	MOV	@R0, *05H
206F	08	INC	R0
2070	7600	MOV	@R0, *00H
2072	7920	MOV	R1, *20H
2074	7700	MOV	@R1, *00H
2076	09	INC	R1
2077	7708	MOV	@R1, *08H

2079	09		INC R1	
207A	7718		MOV @R1, *18H	
207C	09		INC R1	
207D	7710		MOV @R1, *10H	
207F	09		INC R1	
2080	7730		MOV @R1, *30H	
2082	09		INC R1	
2083	7720		MOV @R1, *20H	
2085	09		INC R1	
2086	7728		MOV @R1, *28H	
2088	09		INC R1	
2089	7700		MOV @R1, *00H	
208B	22		RET	; 装代码子程序结束
			ORG 2100H	; 环行分配器子程序开始
2100	C0D0	HXFPO:	PUSH PSW	
2102	C0E0		PUSH ACC	
2104	E570		MOV A,70H	
2106	30E035		JNB ACC.0,LP1	
2109	30E119		JNB ACC.1,LP2	
210C	ED		MOV A,R5	
210D	F8		MOV R0,A	
210E	E6		MOV A,@R0	
210F	B40005		CJNE A, *00H,LP3	
2112	7D11		MOV R5, *11H	; +x
2114	ED		MOV A,R5	
2115	F8		MOV R0,A	
2116	E6		MOV A,@R0	
2117	79E9	LP3:	MOV R1, *0E9H	
2119	F3		MOVX @R1,A	
211A	7A20		MOV R2,*20H	
211C	122550	LP4:	LCALL YANS	
211F	0D		INC R5	
2120	D0E0		POP ACC	
2122	D0D0		POP PSW	
2124	22		RET	
2125	ED	LP2:	MOV A,R5	
2126	F8		MOV R0,A	
2127	E6		MOV A,@R0	

```

CJNE    A,#00H,LP5
MOV     R5,#16H           ; -x
MOV     A,R5
MOV     R0,A
MOV     A,@R0
LP5:    MOV     R1,#0E9H
        MOVX    @R1,A
        MOV     R2,#20H
LP6:    LCALL   YANS
        DEC     R5
        POP     ACC
        POP     PSW
        RET
LP1:    JNB     ACC.3,LP7
        MOV     A,R6
        MOV     R0,A
        MOV     A,@R0
        CJNE   A,#00H,LP8
        MOV     R6,#21H           ; +y
        MOV     A,R6
        MOV     R0,A
        MOV     A,@R0
LP8:    MOV     R1,#0E8H
        MOVX    @R1,A
        MOV     R2,#20H
LP9:    LCALL   YANS           ; 调延时子程序
        INC     R6
        POP     ACC
        POP     PSW
        RET
LP7:    MOV     A,R6
        MOV     R0,A
        MOV     A,@R0
        CJNE   A,#00H,LP10
        MOV     R6,#26H           ; -y
        MOV     A,R6
        MOV     R0,A
        MOV     A,@R0

```

2165	79E8	LP10:	MOV	R1, *0E8H	
2167	F3		MOVX	@R1, A	
2168	7A20		MOV	R2, *20H	
216A	122550	LP11:	LCALL	YANS	
216D	1E		DEC	R6	
216E	D0E0		POP	ACC	
2170	D0D0		POP	PSW	
2513	22		RET		
			ORG	2550H	延时子程序
2550	C002	YANS:	PUSH	02H	
2552	C002	LOS3:	PUSH	02H	
2554	C002	LOS2:	PUSH	02H	
2556	DAFE	LOS1:	DJNZ	R2, LOS1	
2558	D002		POP	02H	
255A	DAF8		DJNZ	R2, LOS2	
255C	D002		POP	02H	
255E	DAF2		DJNZ	R2, LOS3	
2560	D002		POP	02H	
2562	DAEC		DJNZ	R2, YANS	
2564	22		RET		

## 二、逐点比较法的直线插补程序

### 1. 程序框图 (见图4.20)

### 2. 实验参考程序 (直线插补)

			ORG	2300H	
2300	758160	MAIN:	MOV	SP, *60H	; 主程序开始
2303	78EB		MOV	R0, *0EBH	
2305	7480		MOV	A, *80H	
2307	F2		MOVX	@R0, A	
2308	122050		LCALL	DSUP	; 装状态代码
230B	7D11		MOV	R5, *11H	
230D	7E21		MOV	R6, *21H	
230F	7B14		MOV	R3, *14H	
2311	7528C8	LOP4:	MOV	28H, *0C8H	; xe
2314	7529C8		MOV	29H, *0C8H	; ye
2317	752A00		MOV	2AH, *00H	; x
231A	752B00		MOV	2BH, *00H	; y
231D	752E00		MOV	2EH, *00H	; F
2320	75700A		MOV	70H, *0AH	

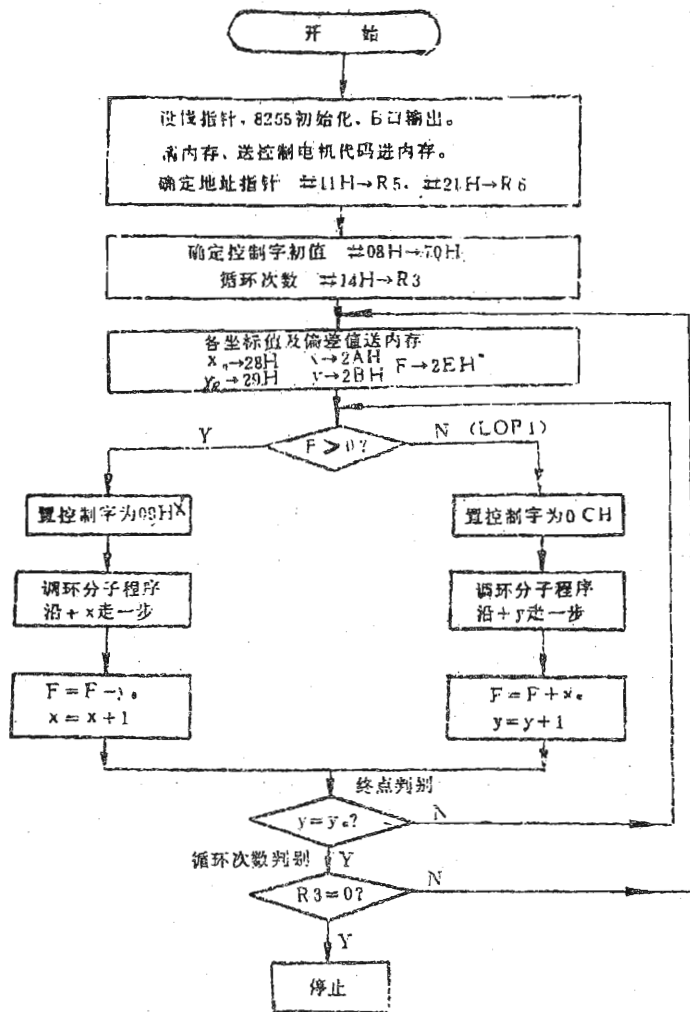


图4.20 直线插补程序框图

2323	E52E	LOP3: MOV	A, 2EH
2325	20E713	JB	ACC.7, LOP1
2328	E570	MOV	A, 70H
232A	D2E0	SETB	ACC.0
232C	C2E2	CLR	ACC.2
232E	F570	MOV	70H, A ; 0B + x
2330	122100	LCALL	HXFPO; 调环行分配器子程序, x方向走一步
2333	E52E	MOV	A, 2EH
2335	9529	SUBB	A, 29H ; F - ye

```

                INC      2AH          ; x+1
                AJMP    LOP2
LOP1:          MOV      A,70H
                SETB   ACC.2
                CLR    ACC.0
                MOV    70H,A          ; 0E +y
                LCALL  HXFPQ          ; y方向走一步
                MOV    A,2EH
                ADD    A,28H          ; F+xe
                INC    2BH            ; y+1
LOP2:          MOV    2EH,A
                MOV    A,28H
                CJNE   A,2AH,LOP3     ; xe=x?
                DJNZ   R3,LOP4
                MOV    R0,#0EBH
                MOV    A,#00H
                MOVX   @R0,A
                LJMP   0000H

```

环行分配器子程序 (参阅实验二)

装状态代码子程序 (参阅实验二)

## 附录二

## 步进电机基本原理实验

```

D8155 EQU 0FF00H ;8155 命令口地址
D8155C EQU 0FF03H ;8155PC 口地址
ORG 0000H
SJMP START
ORG 000BH
LJMP TO_INT
ORG 40H
START: MOV SP, #60H
MOV R4, #0GH
MOV R5, #00H
MOV R6, #00H
MOV TMOD, #00 ;初始化定时器 0
MOV TH0, #4CH
MOV TLO, #00H
MOV DPTR, #D8155 ;初始化 8155, 使 PC 口为输入口
MOV A, #0GH
MOVX @DPTR, A
SETB ET0 ;允许定时器 T0 中断
SETB TR0 ;启动定时器 0
SETB EA ;开总中断 EA
WAIT: LCALL READKEY ;读拨动开关状态
SJMP WAIT ;等待中断
READKEY: ;读拨动开关状态子程序
MOV DPTR, #D8155C
MOVX A, @DPTR
MOV 30H, A ;开关状态存入 30H 单元
RET
TO_INT: ;定时器 0 中断子程序
CLR TR0 ;关定时器 0 溢出中断
MOV TH0, #4CH ;置定时器 0 初始值
MOV TLO, #00H
SETB TR0
MOV A, 30H
CLR C ;清进位标志
MOV 31H, #0H
MOV RO, #4H
NEXT: INC 31H
RRC A
JC ENDSHIFT
DJNZ RO, NEXT
MOV P1, #00H ;电机自由状态
LJMP ENDINT

```



```

ENDSHIPT:
    MOV     A, 31H
    CJNE   A, #01H, NEXT1      ;K1 为 1, 产生四相四拍脉冲
    LJMP   D4S4
NEXT1:    CJNE   A, #02H, NEXT2      ;K2 为 1, 产生四相双四拍脉冲
    LJMP   D4D4
NEXT2:    CJNE   A, #03H, NEXT3      ;K3 为 1, 产生四相八拍脉冲
    LJMP   D4S8
NEXT3:    CJNE   A, #4H, NEXT4      ;K4 为 1, 电机处于自锁状态
    LJMP   LOCK
NEXT4:    MOV     P1, #00H          ;电机为自由状态
    RETI
D4S4:     MOV     DPTR, #PD4S4      ;四相单四拍程序
    INC     R4
    MOV     A, R4
    MOVC   A, @A+DPTR
    MOV     P1, A
    CJNE   R4, #04H, ENDINT
    MOV     R4, #0
    SJMP   ENDINT
D4D4:     MOV     DPTR, #PD4D4      ;四相双四拍程序
    INC     R5
    MOV     A, R5
    MOVC   A, @A+DPTR
    MOV     P1, A
    CJNE   R5, #04H, ENDINT
    MOV     R5, #0
    SJMP   ENDINT
D4S8:     MOV     DPTR, #PD4S8      ;四相八拍程序
    INC     R6
    MOV     A, R6
    MOVC   A, @A+DPTR
    MOV     P1, A
    CJNE   R6, #08H, ENDINT
    MOV     R6, #0
    SJMP   ENDINT
LOCK:     MOV     P1, #01H          ;电机自锁程序
ENDINT:   RETI                    ;中断返回
PD4S4:    DB     00H, 01H, 02H, 04H, 08H
PD4D4:    DB     00H, 03H, 06H, 0CH, 09H
PD4S8:    DB     00H, 01H, 03H, 02H, 06H, 04H, 0CH, 08H, 09H
    END                                ;程序结束

```

## 附录三

## 步进电机速度调节和方向控制实验

```

D8155 EQU OFF00H ;8155 控制字口地址
D8155C EQU OFF03H ;8155 PC 口地址
D8155A EQU OFF01H ;8155 PA 口地址
SPDTR EQU 20H ;速度值(延时时间)暂存器
STEPPT EQU 21H ;步进码表指针
DIRL EQU 22H ;方向(步进码表首址)值暂存器低位
DIRH EQU 23H ;方向(步进码表首址)值暂存器高位
SWITCH EQU 24H ;开关状态码暂存器

ORG 0000H
LJMP START
ORG 000BH ;步进中断程序
LJMP RUNINT
ORG 0100H

START: MOV SP, #40H
LCALL INIT ;调用初始化子程序
LCALL CHKSWITCH ;调用开关状态检测子程序
SJMP $-3

INIT: CLR EA ;初始化子程序, 关中断
MOV DPTR, #D8155
MOV A, #0CH
MOVX @DPTR, A ;A 口输入, C 口输出
MOV DPTR, #MRSTEP ;初始化方向暂存器
MOV DIRH, DPH
MOV DIRL, DPL
MOV SPDTR, #00H ;初始化速度暂存器
MOV SWITCH, #00H ;初始化开关状态暂存器
MOV TMOD, 00H ;T0 设为定时器方式 0
SETB ET0 ;T0 中断允许
SETB EA ;开中断
RET ;初始化子程序返回

CHKSWITCH: ;开关状态检测子程序
MOV DPTR, #D8155A
MOVX A, @DPTR ;读入开关状态
ANL A, #8FH ;屏蔽无效位
CJNE A, SWITCH, $+6 ;开关状态
LJMP SWITCHRET ;未改变则子程序返回
MOV SWITCH, A ;保存新的开关状态值
CLR TR0 ;关定时器 T0
RLC A ;检测 K8 状态
JC $+7 ;为 0 则保存左移码表首址
MOV DPTR, #MLSTEP ;为 1 则保存右移码表首址
SJMP $+5
MOV DPTR, #MRSTEP ;为 1 则保存右移码表首址
MOV DIRL, DPL ;保存码表首址于方向暂存器中
MOV DIRH, DPH
RLC A
RLC A
RLC A

```

```

        RLC     A
        JC     SPD4           ;K4 为 1 则跳转
        RLC     A
        JC     SPD3           ;K3 为 1 则跳转
        RLC     A
        JC     SPD2           ;K2 为 1 则跳转
        RLC     A
        JC     SPD1           ;K1 为 1 则跳转
        SJMP    SPDO
SPD4:   MOV     A, #04H
        SJMP    CHGSPD
SPD3:   MOV     A, #03H
        SJMP    CHGSPD
SPD2:   MOV     A, #02H
        SJMP    CHGSPD
SPD1:   MOV     A, #01H
        SJMP    CHGSPD
SPDO:   MOV     A, #00H
CHGSPD: MOV     DPTR, #SPD
        MOVC   A, @A+DPTR
        MOV    SPDTR, A           ;查表将速度值放入暂存器
        JZ     $+4               ;速度值为 0 则不开中断
        SETB   TRO
SWITCHRET: RET                   ;开关状态检测程序返回
RUNINT: PUSH   PSW               ;步进中断程序。保存现场
        PUSH   ACC
        PUSH   DPH
        PUSH   DPL
        CLR    TRO
        MOV    TH0, SPDTR
        MOV    TLO, #00H
        SETB   TRO
        MOV    DPH, DIRH           ;取步进码表首址
        MOV    DPL, DIRL
        MOV    A, STEPPT
        INC    A                   ;修改指针指向下一位置
        ANL   A, #07H           ;指针指到表尾时自动返回表头
        MOV    STEPPT, A
        MOVC   A, @A+DPTR
        MOV    DPTR, #D8155C
        MOVX   @DPTR, A           ;输出步进码
        POP    DPL
        POP    DPH
        POP    ACC
        POP    PSW               ;恢复现场
        RETI   ;步进中断返回

SPD:    DB 00H, 17H, 77H, 0A7H, 0C7H
MRSTEP: DB 01H, 03H, 02H, 06H, 04H, 0CH, 08H, 09H
MLSTEP: DB 08H, 0CH, 04H, 06H, 02H, 03H, 01H, 09H
END

```