

单片机控制与应用实验报告

实验三 步进电机原理及应用

实验原理

（该实验涉及的基本原理及其在实验中的使用方法）

1、本实验采用定时器中断实现，使用定时器时，首先应由外部条件得到要定时的时间长度 t ，如本实验中，就是根据要求的速度计算出的每一步之间的间隔。然后选择适当的定时器工作方式，去计算想要设定的计数器初值 s ，使用如下方程。

$$(2 \text{ 定时器最大位数} - s) \times \text{定时周期} = t$$

定时周期 = $12/\text{CPU 晶振频率}$

$$(2 \text{ 定时器最大位数} - s) \times \text{定时周期} = t$$

得到的 s 需要分成高 8 位和低 8 位，分别放入计数器 THx 和 TLx 中（x 为 0 或 1）。如果 s 为负数，说明需要的定时时间太长，即使定时器的最大时间也无法满足要求。这种情况下，需要加入软件循环才能实现。我们可以将需要的定时时间分成 n 份，利用定时器达到 t/n 的时间长度，然后在定时器处理程序中，累计某一变量，如果到达 n ，说明总的时间 t 已经达到。

要想使用定时器中断，除了上面的定时器初值设定外，还需要将其他相关的特殊功能寄存器也都设置好。如果使用方式 0 和方式 1，不要忘记在计数结束后重新恢复计数器初值。

2、我们使用的单片机系统的频率是 12M；步进电机转动一周需要 24 步。

本步进电机实验板，使用 FAN8200 作为驱动芯片。CPU 通过如下 4 个引脚与 FAN8200 相连，即：

CPU	FAN8200
P1.1	CE1
P1.4	CE2
P3.2	IN1
P1.0	IN2

3、本实验使用简单的双四拍工作方式即可，这也是 FAN8200 比较方便的工作方式。只要将 CE1 和 CE2 分别置为高，然后 IN1 和 IN2 按照预定的脉冲输出，即 01→11→10→00→01 这个循环构成一个方向旋转的输出脉冲，将此序列翻转，就是相反方向的输出脉冲。

4、数码管显示：

本开发平台有 3 个数码管，使用串行方式连接在一起，具体电路参见实验原理。要想输出一个字形码，就需要从高位到低位依次向移位寄存器输出 8 个比特。移位寄存器的数据线

和时钟线分别接到单片机的 P4.5 和 P4.4 管脚，可以使用 MCS-51 里面的位操作指令进行输出。连续输出 3 个字形，24 个 bit 之后，欲显示的字形将稳定地显示在数码管上，程序可以转而执行其他工作。

七段字形的编码方式需要通过实验获得。这些编码作为程序中的常数，使用 DB 命令存放。在程序中，需要将数值转换为相应的字形编码，可以使用 MOVC 指令来完成。

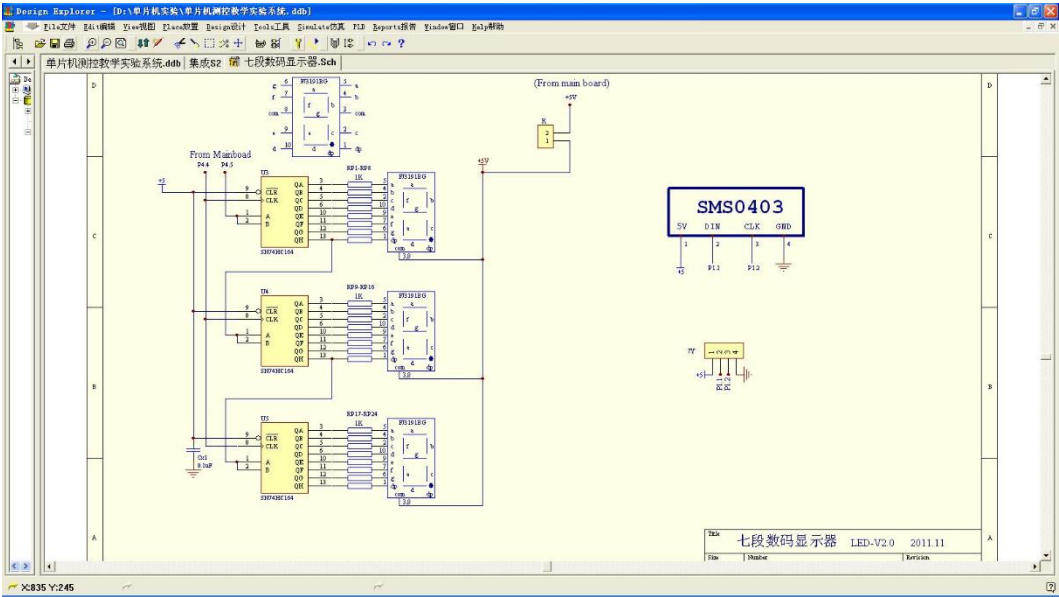
5、74HC164 是高速 CMOS 器件。74HC164 是 8 位边沿触发式移位寄存器，串行输入数据，然后并行输出。数据通过两个输入端（A 或 B）之一串行输入；任一输入端可以用作高电平使能端，控制另一输入端的数据输入。两个输入端或者连接在一起，或者把不用的输入端接高电平，一定不要悬空。

6、时钟（CLK）每次由低变高时，数据右移一位，输入到 Q0，Q0 是两个数据输入端（A 和 B）的逻辑与，它将上升时钟沿之前保持一个建立时间的长度。

7、主复位（CLR）输入端上的一个低电平将使其它所有输入端都无效，同时非同步地清除寄存器，强制所有的输出为低电平。

8、采用 3 个 74HC164 级联控制三个数码管的显示，具体实验原理如下图所示。其中使用单片机 P4.5 作为模拟串口数据，使用 P4.4 模拟串口时钟，CLR 端接高电平。使用上一个 74HC164 的 Q7 作为下一个 74HC164 的输入端。

实验涉及到原理图：



实验程序

```
ORG 0000H
    LJMP START
ORG 000BH
```

```

    LJMP T_INT
ORG 0040H
START:
    P4 EQU 0C0H
    P4SW EQU 0BBH
    MOV P4SW,#70H
    DAT EQU P4.5
    CLK EQU P4.4

    SW1 EQU P3.6
    SW2 EQU P3.7

    CE1 EQU P1.1
    CE2 EQU P1.4
    IN1 EQU P3.2
    IN2 EQU P1.0

    SETB CE1
    SETB CE2
    CLR IN1
    CLR IN2

    MOV R6,#0
    MOV R5,#0
    MOV R4,#0

    MOV TMOD,#01H
;MOV IE,#10000010B
    MOV TL0,#3DH
    MOV TH0,#5DH
    SETB TR0
    SETB ET0
    SETB EA

MAIN:
    MOV A,R6
    MOV DPTR,#TAB
    MOVC A,@A+DPTR
    LCALL DISPLAY
    MOV A,R5
    MOV DPTR,#TAB
    MOVC A,@A+DPTR
    LCALL DISPLAY
    MOV A,R4

```

```
MOV DPTR,#TAB
MOVC A,@A+DPTR
LCALL DISPLAY
```

```
JB SW2,FAN
```

```
ZHENG:
```

```
CLR CE1
CLR CE2
JNB IN1,Z1
JNB IN2,Z2
CPL IN1
SETB CE1
SETB CE2
JMP BUJIA
```

```
Z1:
```

```
JNB IN2,Z3
CPL IN2
SETB CE1
SETB CE2
JMP BUJIA
```

```
Z2:
```

```
CPL IN2
SETB CE1
SETB CE2
JMP BUJIA
```

```
Z3:
```

```
CPL IN1
SETB CE1
SETB CE2
JMP BUJIA
```

```
FAN:
```

```
CLR CE1
CLR CE2
JNB IN1,F1
JNB IN2,F2
SETB IN1
CLR IN2
SETB CE1
SETB CE2
JMP BUJIA
```

```
F1:
```

```
JNB IN2,F3
SETB IN1
```

SETB IN2
SETB CE1
SETB CE2
JMP BUJIA

F2:

CLR IN1
CLR IN2
SETB CE1
SETB CE2
JMP BUJIA

F3:

CLR IN1
SETB IN2
SETB CE1
SETB CE2

BUJIA:

INC R6
CJNE R6,#10,ZHONGDUAN
INC R5
MOV R6,#0
CJNE R5,#10,ZHONGDUAN
INC R4
MOV R5,#0
CJNE R4,#10,ZHONGDUAN
MOV R4,#0

ZHONGDUAN:

MOV R3,#6H

TT1:

MOV R0,#0

DELAY:

CJNE R0,#1,DELAY
JNB SW1,OUT
DJNZ R3,TT1

OUT:

JMP MAIN

DISPLAY:

MOV R0,#8

D1:

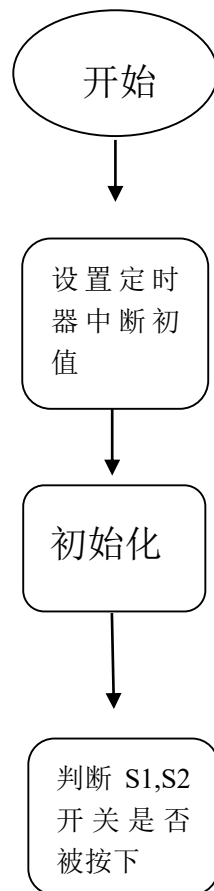
CLR CLK

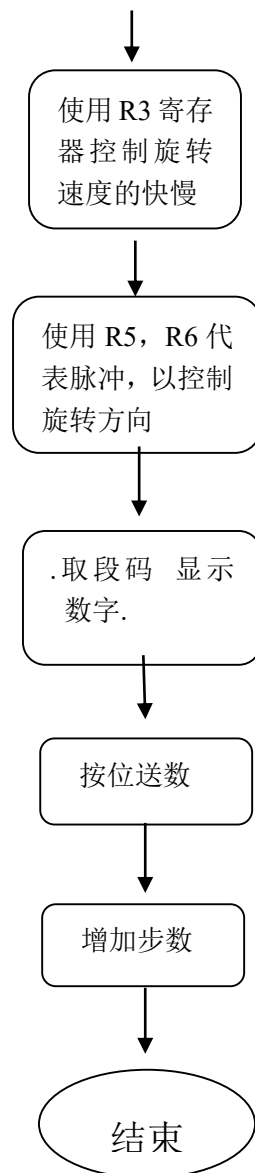
```
RLC A
MOV DAT,C
SETB CLK
DJNZ R0,D1
RET
```

```
T_INT:
MOV R0,#1
MOV TL0,#3DH
MOV TH0,#5DH
RETI
```

```
TAB: DB 0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H
END
```

流程图





思考题

(1) 如采用单四拍工作模式, 每次步进角度是 15° , 改变相位值: 01, 10, 00, 00

(2) 如采用单双八拍工作模式, 每次步进角度是 7.5°

八拍: A-AB-B-BC-C-CD-D-DA-A 11 10 10 00 00 00 01

修改: 定时器定时周期变了, 故需修改定时初始值; 相位值需改变, 还有循环次数

(3) 脉冲频率。由于物理因素 (比如摩擦、机械惯性、响应时间等), 步进电机的最高转速有限制。上限根据电机不同而不同, 下限为 0

(4) 改变脉冲顺序

(5) 额定功率, 额定电流, 转速, 马力, 额定转矩

(6) MCS-51 存储器片内 RAM、片外 RAM ROM 三空间

片内 RAM: 00H-7FH (52 系列延伸 FFH)

其 00H-1FH, 共 32 字节, 分成四个工作寄存器区, 每区 8 个寄存器 (R0~R7)
20H-2FH, 共 16 字节, 为位寻址区, 共 128 位, 位址: 00~7FH
80H-FFH, 共 128 个地址号码, 其离散布着 21 特殊功能寄存器, 必须直接寻址才能读写

对上述空间读写必须使用 MOV 指令

片外 RAM: 0000H-FFFFH, 容量 64KB

片外 RAM 读写必须使用 MOVX 指令

ROM: 0000H-FFFFH, 容量 64KB, 其 0000~0FFFH 即 4K 片内其片外

ROM 读必须使用 MOVC 指令

MCS-51 有五个独立的寻址空间。

64K 字节程序存储器空间 (0—0FFFFH)

64K 外部数据存储器空间 (0—0FFFFH)

256 字节内部 RAM 空间 (0—0FFH)

256 位寻址空间 (0—0FFH)

工作寄存器区这些寻址空间中, 工作寄存器区重合在内部 RAM 的前 128 字节空间中, 后 128 字节是内部特殊功能寄存器 (SFR) 空间, 位寻址区的前 128 个地址重合在内部 RAM 中, 后 128 个地址重合在 SFR 中的一部分寄存器中。MCS-51 系列中 不同型号的单片机, 特殊功能寄存器的定义和使用不完全一致。

(7) 查表指令 MOVC A, @A+PC 这条指令以 PC 作为基址寄存器, A 的内容作为无符号数和 PC 内容 (下一条指令的起始地址) 相加得到的 16 位地址, 由该地址指出的程序存储器单元内容 送到累加器。此指令常用于查表, 要求表格整个存放在查表指令以下的 256 字节之内。

MOVC A, @A+DPTR 此指令也用于查表, 以 DPTR 作为基址寄存器, A 的内容和 DPTR 相加作为 程序存储器的地址, 此地址内容送到 A。这种查表指令比较方便使用, 表格可以存放

到任意地址, 可在多处使用。但表格的大小仍不能超过 256 字节。

(8) 大多数 MCS-51 的指令执行时间为一个机器周期, 小部分指令为 2 个机器周期, 只有乘除法需要 4 个机器周期。

实验中遇到的问题及解决

- 1、每次进入中断, 应当首先设置计数初值。
- 2、通过三个寄存器作为步数与显示器的连接。
- 3、通过脉冲变化控制步进电机旋转方向。

实验四 LED 点阵显示屏

原理总结

（该实验涉及的基本原理及其在实验中的使用方法）

1、高亮度 LED 发光管构成点阵，通过编程控制可以显示中英文字符、图形及视频动态图形。所显示字符的点阵数据可以自行编写（即直接点阵画图），也可从标准字库（如 ASC16、HZ16）中提取。后者需要正确掌握字库的编码方法和字符定位的计算。

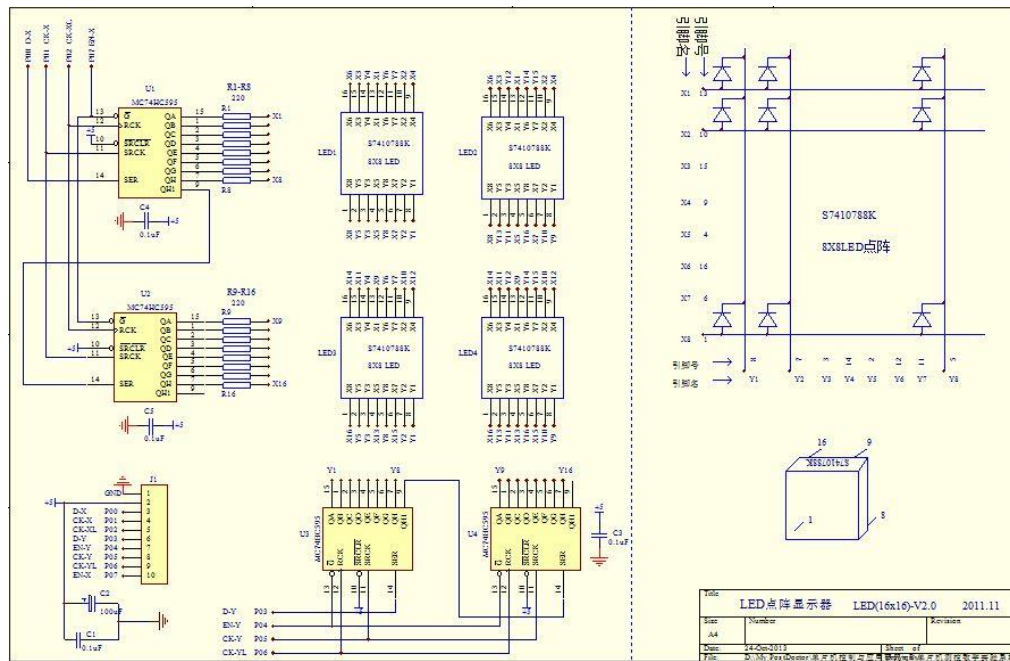
2、实验用的 LED 点阵显示屏为 16*16 点阵。行和列分别使用两个移位寄存器作为输出。当移位寄存器输出的第 i 行为 0，第 j 列为 1 时点亮点(i,j)。为了能够显示出一个点阵字型，需要进行循环扫描，也就是每一次只点亮一行，然后在列上输出该列对应的 16 个点阵值。输出一行后暂停一段时间，输出下一行。为了达到较好的显示效果，整屏总的扫描时间不高于 40ms。上述过程中行列可以互换。

3、实验中使用的移位寄存器是 74HC595，它是一个同时具有串行移位和输出锁存驱动功能的器件。74HC595 是具有 8 位移位寄存器和一个存储器，三态输出功能。移位寄存器和存储器是分别的时钟。

4、数据在 SRCK（移位寄存器时钟输入）的上升沿输入到移位寄存器中，在 RCK（存储器时钟输入）的上升沿输入到存储寄存器中去。移位寄存器有一个串行移位输入（行 Dx (P00)、列 Dy (P03)），和一个串行输出（QH），和一个异步的低电平复位，存储寄存器有一个并行 8 位的，具备三态的总线输出，当使能（P02 和 P07 为低电平）时，存储寄存器的数据输出到总线。

5、在控制 74HC595 时，首先将数据放到串行输入的 SI 端，然后在串行时钟 SRCK 上产生一个脉冲，即可输出一个 bit，重复以上步骤 16 次，输出所有列值。然后给存储器时钟 RCK 一个脉冲，将串行数据锁存起来。将使能端输出低电平，驱动到 LED 点阵上。行的输出每次只移位一次，并重新锁存即可。

本实验涉及到的电路原理图：



程序分析

```

D_YEQU    P0.0;列
CK_Y      EQU    P0.1
CK_YL     EQU    P0.2
D_XEQU    P0.3
EN_X      EQU    P0.4
CK_X      EQU    P0.5
CK_XL     EQU    P0.6
EN_Y      EQU    P0.7

```

```

ORG      00H
LJMP     START
ORG      40H
START:
    CLRCK_X
    CLRCK_XL
    CLRCK_Y
    CLRCK_YL
    SETB  EN_X
    SETB  EN_Y
    MOV   DPTR,#TA
    MOV   R7,#0
LOOP:
    MOV   A,#0

```

```

MOV    R0,#0
MOV    R1,#0
MOV    R5,#255

LOOP_0:
MOV    A,R0
ADD    A,R7
CJNE   A,#224,L0
L0:
JC     L1
SUBB   A,#224
L1:
MOVC   A,@A+DPTR
MOV    R2,A
INC    R0
MOV    A,R0
ADD    A,R7
CJNE   A,#224,L2
L2:
JC     L3
SUBB   A,#224
L3:
MOVC   A,@A+DPTR
MOV    R3,A
INC    R0

MOV    A,R3
MOV    R4,#8
Y1:
RRC    A
MOV    D_Y,C
SETB   CK_Y
NOP
CLRCK_Y
DJNZ   R4,Y1

MOV    A,R2
MOV    R4,#8
Y2:
RRC    A
MOV    D_Y,C
SETB   CK_Y
NOP
CLRCK_Y

```

```

    DJNZ    R4,Y2

    SETB    CK_YL

    CJNE    R1,#0,LOOP1
    ACALL   OUTDX
    MOV     R1,#1
    LJMP    LOOP2
LOOP1:
    SETB    D_X
    SETB    CK_X
    NOP
    CLRCK_X

    SETB    CK_XL

LOOP2:
    CLRCK_XL
    CLRCK_YL
    CLREN_X
    CLREN_Y
    ACALL   DELAY
    SETB    EN_X
    SETB    EN_Y

    MOV     A,#0
    MOV     R4,#8
C1:
    RRC     A
    MOV     D_Y,C
    SETB    CK_Y
    NOP
    CLRCK_Y
    DJNZ    R4,C1

    MOV     A,#0
    MOV     R4,#8
C2:
    RRC     A
    MOV     D_Y,C
    SETB    CK_Y
    NOP
    CLRCK_Y
    DJNZ    R4,C2

```

```

    SETB    CK_YL
    NOP
    CLRCK_YL
    CLRCK_XL
    CLREN_X
    CLREN_Y
    ACALL   DELAY
    SETB    EN_X
    SETB    EN_Y

    CJNE    R0,#32,LOOP3
    MOV     R0,#0
    MOV     R1,#0
LOOP3:
    DJNZ    R5,LOOP5
    INC R7
    INC R7
    CJNE    R7,#224,LOOP4
    MOV     R7,#0
LOOP4:
    LJMP    LOOP
LOOP5:
    LJMP    LOOP_0
OUTDX:
X0:
    MOV     A,#255
    MOV     R4,#8
X1:
    RLCA
    MOV     D_X,C
    SETB    CK_X
    NOP
    CLRCK_X
    DJNZ    R4,X1

    MOV     A,#254
    MOV     R4,#8
X2:
    RLCA
    MOV     D_X,C
    SETB    CK_X
    NOP
    CLRCK_X

```

```

        DJNZ    R4,X2

        SETB    CK_XL
RET

DELAY:
        MOV     R6,#255
DE1:    INC R6
        DEC     R6
        DJNZ    R6,DE1
RET
TA:

DB 00H,00H,07H,F0H,08H,08H,10H,04H,10H,04H,08H,08H,07H,F0H,00H,00H

DB 00H,00H,0EH,38H,11H,44H,10H,84H,10H,84H,11H,44H,0EH,38H,00H,00H

DB 00H,00H,0EH,0CH,10H,14H,10H,24H,10H,44H,10H,84H,0FH,0CH,00H,00H

DB 00H,00H,07H,F0H,08H,08H,10H,04H,10H,04H,08H,08H,07H,F0H,00H,00H

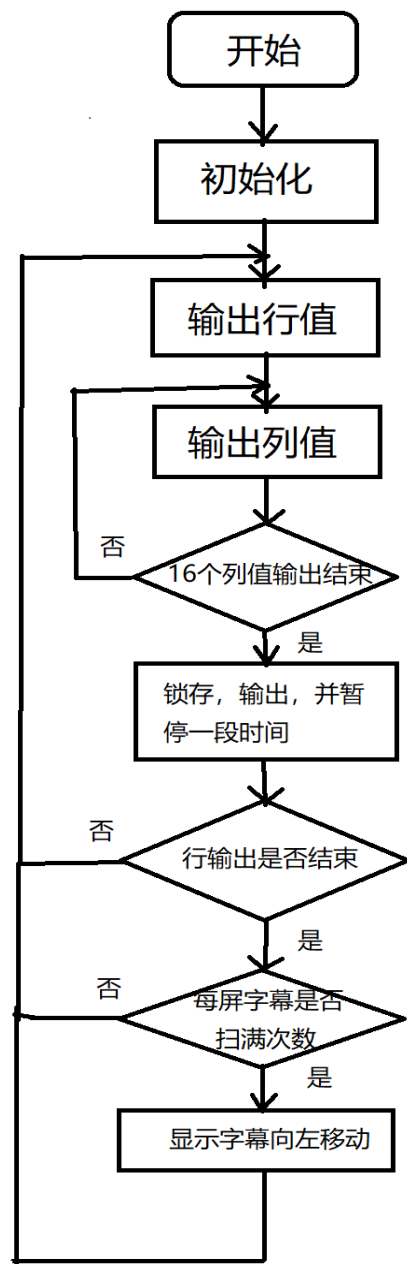
DB 20H,20H,20H,41H,21H,82H,2EH,44H,24H,28H,F4H,30H,27H,C0H,20H,00H
DB 27H,FCH,F4H,02H,24H,42H,24H,22H,27H,C2H,20H,02H,20H,1EH,00H,00H

DB 20H,01H,20H,41H,3FH,49H,A2H,49H,62H,49H,24H,49H,24H,49H,20H,FFH
DB 01H,49H,7EH,49H,54H,49H,55H,49H,54H,C9H,7FH,41H,00H,01H,00H,00H

END

```

流程图



思考题

- (1) 提高刷新频率
- (2) 提高刷新频率
- (3) 改变刷新频率，位数，增加硬件锁存设备

实验中遇到的问题及解决

- 1、为消除在切换行显示数据的时候产生拖尾现象，要先关闭显示屏，等显示数据打入输出锁存器并锁存，然后再输出新的行号，重新打开显示。
- 2、通过调整延时程序调整字码的飘过速度。