

实验三 步进电机原理及应用

一. 实验目的和要求

- (1) 初步学习和掌握 MCS-51 的体系结构和汇编语言，了解 Keil 编程环境和程序下载工具的使用方法。
- (2) 了解步进电机的工作原理，学习用单片机的步进电机控制系统的硬件设计方法，掌握定时器和中断系统的应用，熟悉单片机应用系统的设计与调试方法。
- (3) 了解数码管输出的原理及编程方式。

二. 实验原理

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模式即可，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。

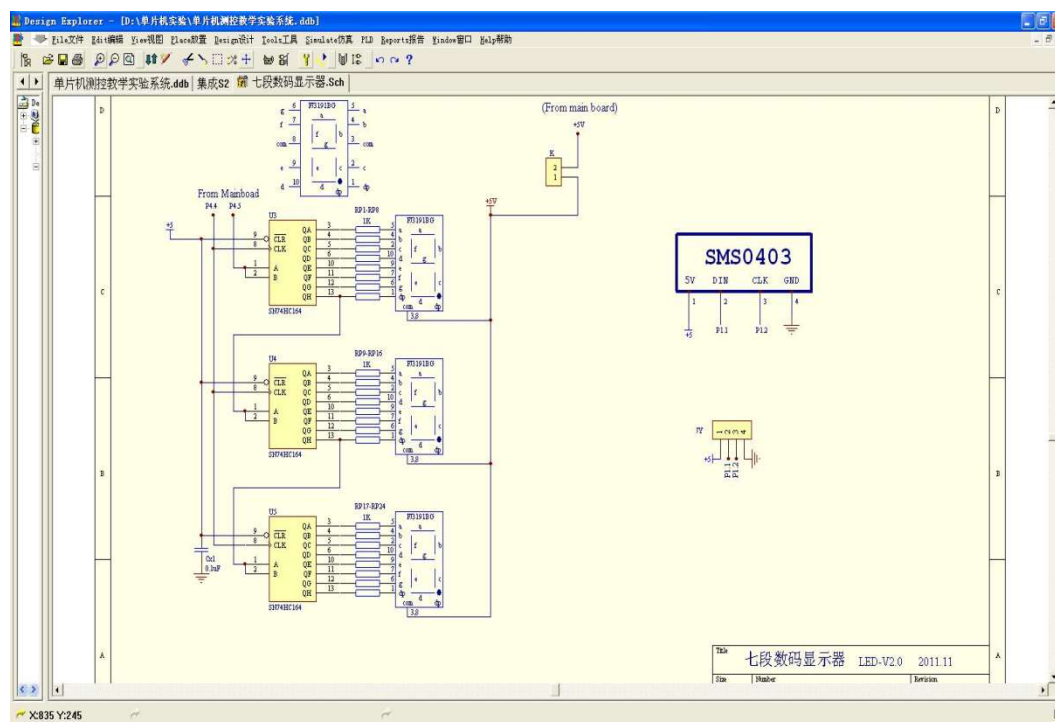
5、74HC164 是高速 CMOS 器件。74HC164 是 8 位边沿触发式移位寄存器，串行输入数据，然后并行输出。数据通过两个输入端（A 或 B）之一串行输入；任一输入端可以用作高电平使能端，控制另一输入端的数据输入。两个输入端或者连接在一起，或者把不用的输入端接高电平，一定不要悬空。

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：



三、实验器材

单片机测控实验系统，步进电机控制实验模块，Keil 开发环境，STC-1S 程序下载工具

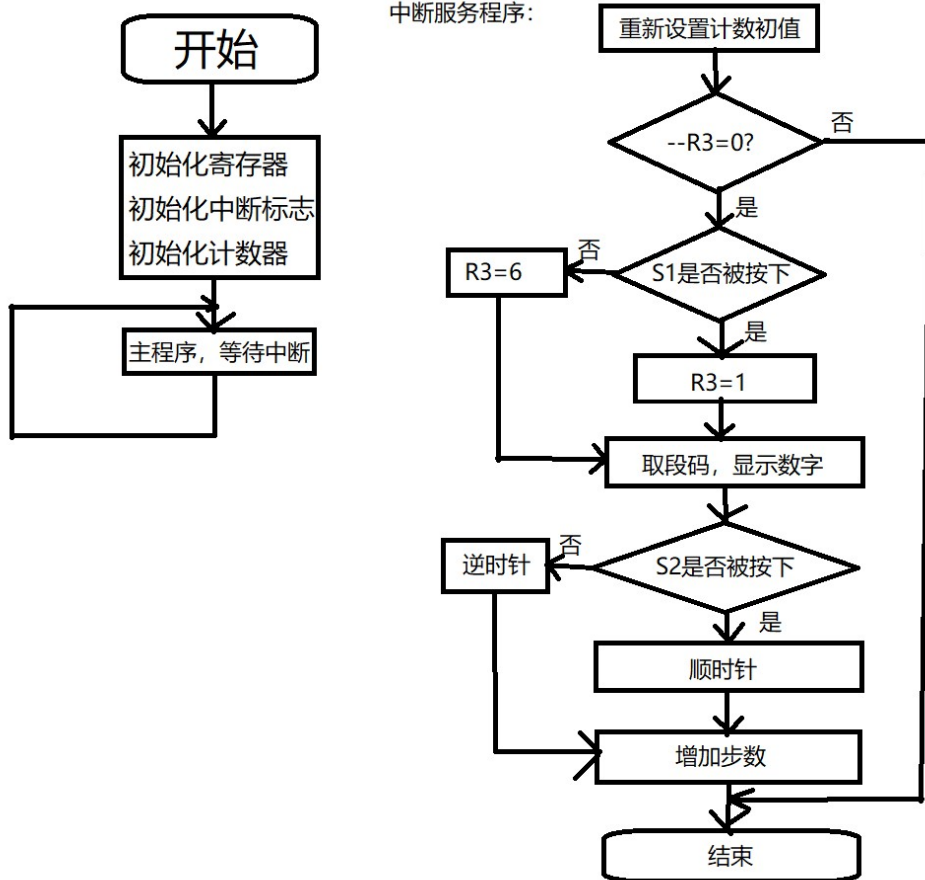
四、实验内容

1. 编制 MCS-51 程序使步进电机按照规定的转速和方向进行旋转，并将已转动的步显示在数码管上。
2. 步进电机的转速分为两档，当按下 S1 开关时，进行快速旋转，速度为 60 转/分。当松开开关时，进行慢速旋转，速度为 10 转/分。当按下 S2 开关时，按照顺时针旋转；当松开时，按照逆时针旋转。
3. 本程序要求使用定时器中断来实现，不准使用程序延时的方式。

五、实验步骤

1. 预习
2. 单程序录入和调试
3. 程序调试
4. 编写程序，完成功能

六、流程图



七、程序分析

（程序设计的思路、程序代码+注释）

```

ORG    0000H
LJMP   START
ORG    000BH   ;T0 中断服务程序
LJMP   T0_INT
ORG    0040H

START:
    P4 EQU    0C0H   ;P4 地址
    P4SW EQU    0BBH   ;P4 方式控制字地址
    ;MOV    P4,#0FFH
    CLK EQU    P4.4
    DATEQU    P4.5
    MOV    P4SW,#30H

    SWH1 EQU    P3.6;S1
    SWH2 EQU    P3.7;S2

```

```
IN1 EQU    P3.2
IN2 EQU    P1.0
CE1 EQU    P1.3
CE2 EQU    P1.4
```

```
;MOV    SP,#60H
MOV     DPTR,#TABLE
```

```
MOV     R0,#0
MOV     R1,#0
MOV     R2,#0
MOV     R3,#6
MOV     R5,#1
MOV     R6,#1;从 11 开始
```

```
SETB    CE1 ;双四拍工作模式,只要将 CE1 和 CE2 分别置为高
SETB    CE2
SETB    EA ;EA 是整个 CPU 的中断允许标志。当 EA=1 时, CPU 可以响应中断;
SETB    ET0 ;ET1 和 ET0 是 T1 和 T0 的中断允许位
```

```
MOV     TMOD,#01H;T0 计数器, 方式 1
MOV     TL0,#3EH
MOV     TH0,#5DH;计数初值
SETB    TR0 ;运行控制位 TR0 和 TR1 分别控制两个定时器是否允许计数
```

```
LL1:LJMP    LL1
```

```
;.....中断服务程序.....
```

```
T0_INT:
```

```
    PUSH    ACC
;PUSH    PSW
;PUSH    DPL
;PUSH    DPH
    CLRTR0
    MOV     TL0,#3EH
    MOV     TH0,#5DH;计数初值
    SETB    TR0
    DJNZ    R3,IEND
```

```
JNB SWH1,V1;为 0 跳转 (SWH1 按下)
```

```
MOV     R3,#5;慢速
JMP V2
```

```
V1: MOV     R3,#1;快速
```

V2: LCALL DISPLAY;显示步数
LCALL STEP;电机转动

IEND:

;POP DPH
;POP DLH
;POP PSW
POP ACC
RETI

;.....取段码 显示数字.....

DISPLAY:

MOV A,R0
MOVC A,@A+DPTR
LCALL SENDNUM

MOV A,R1
MOVC A,@A+DPTR
LCALL SENDNUM

MOV A,R2
MOVC A,@A+DPTR
LCALL SENDNUM

RET

;.....按位送数.....

SENDNUM:

MOV R4,#8

SE1:CLRCLK

RLCA
MOV DAT,C
SETB CLK
DJNZ R4,SE1

RET

STEP:

JB SWH2,SHUN;按下，跳转，顺时针

;.....逆时针.....

```
CJNE    R5,#1,N1;R5 不为 1 转移(R5==0)
CJNE    R6,#1,N3;R6 不为 1 转移(R6==0)
CLRIN1;(R5==1,R6==1)
SETB    IN2;送 01
MOV     R5,#0
MOV     R6,#1
LJMP    ST0
```

```
N1: CJNE    R6,#1,N2;R6 不为 1 转移(R6==0)
      CLRIN1;(R5==0,R6==1)
      CLRIN2;送 00
      MOV     R5,#0
      MOV     R6,#0
      LJMP    ST0
```

```
N2: SETB    IN1;(R5==0,R6==0)
      CLRIN2;送 10
      MOV     R5,#1
      MOV     R6,#0
      LJMP    ST0
```

```
N3: SETB    IN1;(R5==1,R6==0)
      SETB    IN2;送 11
      MOV     R5,#1
      MOV     R6,#1
      LJMP    ST0
```

;.....顺时针.....

SHUN:

```
CJNE    R5,#1,SH1;R5 不为 1 转移(R5==0)
CJNE    R6,#1,SH3;R6 不为 1 转移(R6==0)
SETB    IN1;(R5==1,R6==1)
CLRIN2;送 10
MOV     R5,#1
MOV     R6,#0
LJMP    ST0
```

```
SH1:   CJNE    R6,#1,SH2;R6 不为 1 转移(R6==0)
      SETB    IN1;(R5==0,R6==1)
      SETB    IN2;送 11
      MOV     R5,#1
      MOV     R6,#1
```

```

        LJMP    ST0

SH2:    CLRIN1;(R5==0,R6==0)
        SETB   IN2;送 01
        MOV    R5,#0
        MOV    R6,#1
        LJMP    ST0

SH3:    CLRIN1;(R5==1,R6==0)
        CLRIN2;送 00
        MOV    R5,#0
        MOV    R6,#0
        LJMP    ST0

;.....增加步数.....
ST0:INC R0
        CJNE   R0,#10,ST1
        MOV    R0,#0
        INC R1
ST1:CJNE   R1,#10,ST2
        MOV    R1,#0
        INC R2
ST2:CJNE   R2,#10,ST3
        MOV    R2,#0

ST3:RET

;.....段码表.....
TABLE:
        DB    0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H

END

```


八. 实验中遇到的问题及解决

①开始时程序无法正常运行：发现 P4 口没有正常定义为接口。加上对 P4SW 接口的正确定义。

②根据脉冲确定顺逆时针的方向，使用两个寄存器控制。

③每次进入中断，应重新设置初值。

实验四 LED 点阵显示屏

一、 实验目的和要求

了解 LED 点阵显示的基本原理和实现方法。掌握点阵汉字库的编码和从标准字库中提取汉字编码的方法。

二、 实验原理

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。

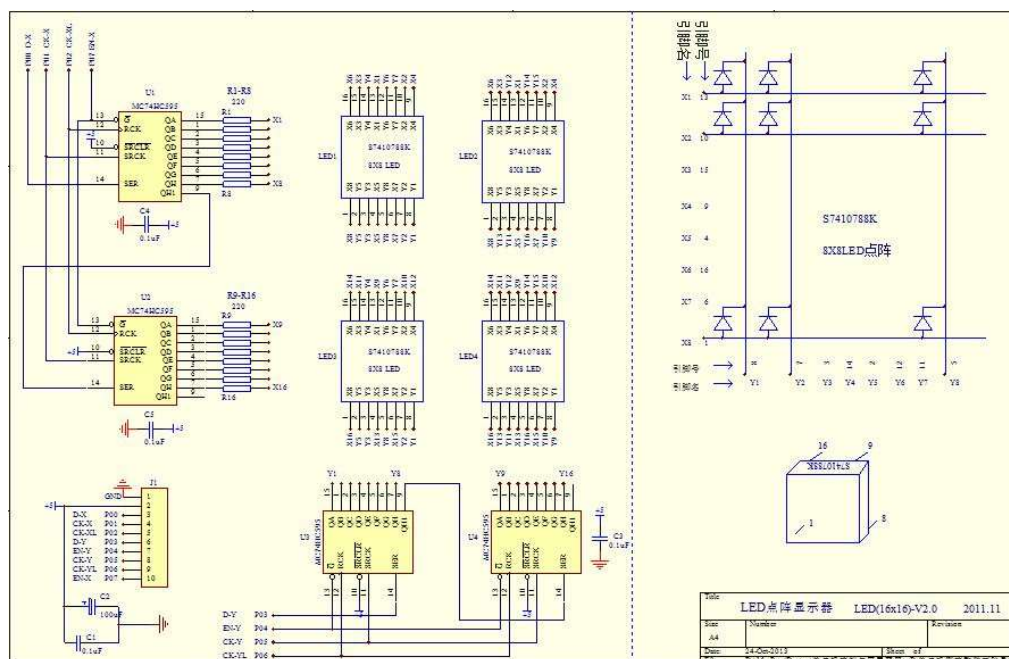
移位寄存器有一个串行移位输入（行 D_x （P00）、列 D_y （P03）），和一个串行输出（QH），和一个异步的低电平复位，存储寄存器有一个并行 8 位的，具备三态的总线输出，当使能（P02 和 P07 为低电平）时，存储寄存器的数据输出到总线。

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：



三、 实验器材

单片机测控实验系统，LED 点阵显示器实验模块，Keil 开发环境，STC-ISP 程序下载工具

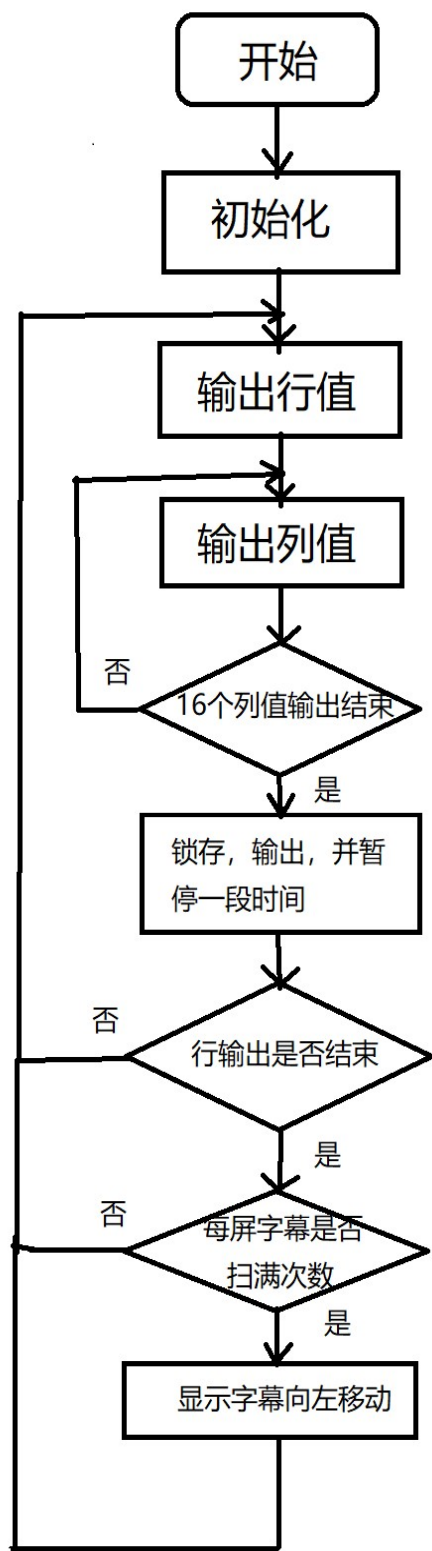
四、 实验内容

- ① 了解 16*16 点阵电路的原理，编写汇编语言程序。
- ② 编写一行汉字字符（至少三个字）的显示程序。
- ③ 能够从左到右（或从右到左）循环显示（要求显示过程中字的大小与屏幕尺寸相适应）。

五、 实验步骤

1. 掌握点阵式 LED 显示屏的控制方法；
2. 使用 MCS-51 汇编语言，使用 LED 点阵显示器显示出正确的汉字字符及动态效果；
3. 将编译后的程序下载到 51 单片机，观察 LED 显示屏的显示结果。

六、 流程图



七、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
000H,000H,007H,0F0H,008H,008H,010H,004H,010H,004H,008H,008H,007H,0F0H,000H,000H
;"0",0

DB
000H,000H,00CH,018H,010H,004H,011H,004H,011H,004H,012H,088H,00CH,070H,000H,000
H;"3",1

```



```

DB
000H,000H,00CH,018H,010H,004H,011H,004H,011H,004H,012H,088H,00CH,070H,000H,000
H;"3",2
DB
000H,000H,00CH,018H,010H,004H,011H,004H,011H,004H,012H,088H,00CH,070H,000H,000
H;"3",3

DB 0x00,0x80,0x00,0x80,0x00,0x80,0x3F,0xFC,0x20,0x84,0x20,0x88,0x20,0x80,0x2F,0xF0
DB 0x24,0x10,0x24,0x10,0x22,0x20,0x21,0x40,0x20,0x80,0x41,0x40,0x46,0x30,0x98,0x0E

DB 0x02,0x00,0x02,0x00,0x03,0xF8,0x02,0x00,0x02,0x00,0x02,0x00,0xFF,0xFE,0x02,0x00
DB 0x02,0x00,0x02,0x40,0x02,0x20,0x02,0x10,0x02,0x08,0x02,0x00,0x02,0x00,0x02,0x00

DB 0x00,0x10,0x00,0x78,0x1F,0x80,0x10,0x00,0x10,0x00,0x10,0x00,0x10,0x00,0x1F,0xFC
DB 0x10,0x40,0x10,0x40,0x10,0x40,0x10,0x40,0x10,0x40,0x10,0x40,0xFF,0xFE,0x00,0x00

END

```

八、实验中遇到的问题及解决

- 1、在所有的字显示结束后，后面出现一串乱码（一条斜杠或者是）。
调整“在所有字显示完毕后”控制归零的那一条语句，计算显示三个字码需要的正确长度。
- 2、字码飘过速度太快。
解决方法：调整延时程序。
- 3、字码显示不清楚
解决办法：修改试验程序使其能清晰显示。