单片机控制与应用实验报告

计科八班　姚金喆　53160812

实验三 步进电机原理及应用

1. 实验目的和要求：

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。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。了解数码管输出的原理及编程方式。

2. 实验设备：

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-ISP 程序下载工具

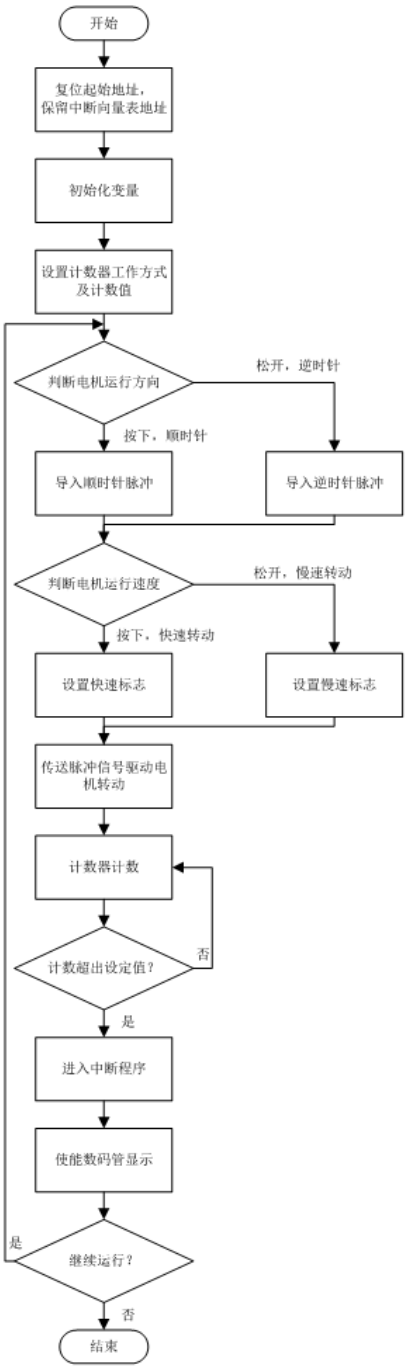
3. 实验要求：

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，并将已转动的步数显示在数码管上。

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。

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。

4. 程序流程图：



5. 程序代码

```
ORG 0000H
LJMP    START  ；长转移
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    ; 数码管时钟线
        DAT EQU P4.5    ; 数码管数据线
        MOV P4SW,#30H

        SWH1    EQU P3.6    ;S1
        SWH2    EQU P3.7    ;S2
        IN1 EQU P3.2    ; 步进电机
        IN2 EQU P1.0    ; 步进电机
        CE1 EQU P1.3    ; 步进电机
        CE2 EQU P1.4    ; 步进电机

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

        MOV R0,#0
        MOV R1,#0
        MOV R2,#0
        MOV R3,#50
        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 ，寄存器位置 1
        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

        ; TH0，TL0 为 T0 的 16 位计数器的高 8 位和低 8 位，TMOD 是方式寄存器，TCON 是状态和控制寄存器
        MOV TMOD,#01H;T0 计数器，01 方式 1，16 位的定时器，00000001，高四位控制 T1，低四位控制 T0，C/T=0 定时方式，gate=0 定时器不受外部控制
        MOV TL0,#3EH
        MOV TH0,#5DH;计数初值，每一步之间间隔 T=1/24=0.041666，s=23870,
        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，GATE 为 0 时，TR 为 1 时允许计数

    LL1:    LJMP    LL1

    ;.............中断服务程序.....................
    T0_INT:
```

```
        PUSH    ACC ； 累加器 acc
        ;PUSH   PSW
        ;PUSH   DPL
        ;PUSH   DPH
        CLR TR0 ； 禁止计数
        MOV TL0,#3EH
        MOV TH0,#5DH;计数初值
        SETB    TR0 ； 允许计数
        DJNZ    R3,IEND     ； 减一不是零就跳转

        JNB SWH1,V1;为 0 跳转（SWH1 按下）
        MOV R3,#6;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   ； movc 基址变址寻址，dptr 基址寄存器，a 变址寄存器，相加作为地址访问程序存储器
        LCALL   SENDNUM     ； 长调用 lcall 调用子程序

        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:    CLR CLK ; 清零
    RLC A   ; 带进位循环左移
    MOV DAT,C   ; 布尔处理器，psw 中进位标志 c
    SETB    CLK ; 置 1
    DJNZ    R4,SE1 ; 减一不是零就跳转

    RET
; 数据每次由低变高时，数据右移一位


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)
    CLR IN1;(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)
    CLR IN1;(R5==0,R6==1)
    CLR IN2;送 00
    MOV R5,#0
    MOV R6,#0
    LJMP    ST0

N2: SETB    IN1;(R5==0,R6==0)
    CLR IN2;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)
    CLR IN2;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:    CLR IN1;(R5==0,R6==0)
    SETB    IN2;送 01
    MOV R5,#0
    MOV R6,#1
    LJMP    ST0

SH3:    CLR IN1;(R5==1,R6==0)
    CLR IN2;送 00
    MOV R5,#0
    MOV R6,#0
    LJMP    ST0

;.....................增加步数.............................
ST0:    INC R0
    CJNE    R0,#10,ST1 ； 比较不相等转移
    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 ；共阳极 0-9
END

6. 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。

计数器初值计算：（$2^{16}-s$）*（12/12M）= 1/24 s = 23870

实验四 LED 点阵显示屏

1. 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。

2. 实验设备：

   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

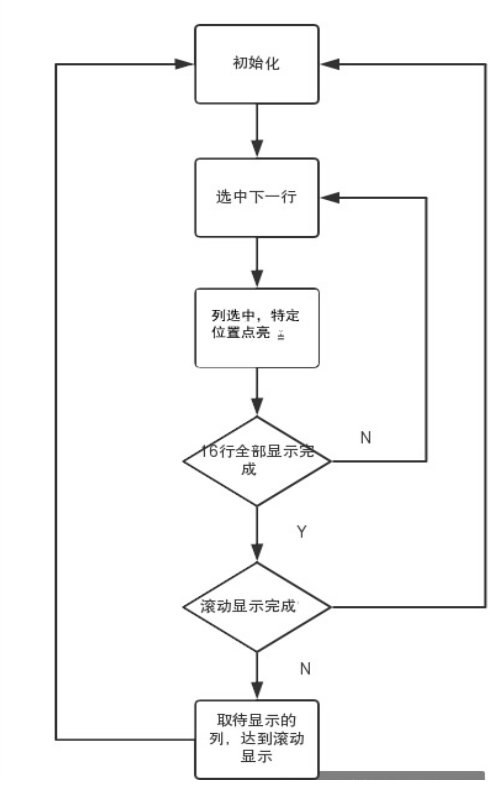
3. 实验要求：

   了解 16*16 点阵电路的原理，编写汇编语言程序。

   编写一行汉字字符（至少三个字）的显示程序。

   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）。

4. 程序流程图：



5. 程序代码：

```
D_Y EQU P0.0;列
CK_Y    EQU P0.1
CK_YL   EQU P0.2
D_X EQU 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:
    CLR CK_X
    CLR CK_XL
    CLR CK_Y
    CLR CK_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
    CLR CK_Y
    DJNZ    R4,Y1

    MOV A,R2
    MOV R4,#8
Y2:
    RRC A
    MOV D_Y,C
    SETB    CK_Y
    NOP
    CLR CK_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
    CLR CK_X

    SETB    CK_XL

LOOP2:
    CLR CK_XL
    CLR CK_YL
    CLR EN_X
    CLR EN_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
    CLR CK_Y
    DJNZ    R4,C1

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

    SETB    CK_YL
    NOP
    CLR CK_YL
    CLR CK_XL
    CLR EN_X
    CLR EN_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:
```

```
        RLC A
        MOV D_X,C
        SETB    CK_X
        NOP
        CLR CK_X
        DJNZ    R4,X1

        MOV A,#254
        MOV R4,#8
    X2:
        RLC A
        MOV D_X,C
        SETB    CK_X
        NOP
        CLR CK_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;"0
",0

    DB
00H,00H,0EH,38H,11H,44H,10H,84H,10H,84H,11H,44H,0EH,38H,00H,00H;"8
",1

    DB
00H,00H,00H,00H,08H,04H,08H,04H,1FH,FCH,00H,04H,00H,04H,00H,00H;"1
",2

    DB
00H,00H,0EH,0CH,10H,14H,10H,24H,10H,44H,10H,84H,0FH,0CH,00H,00H;"2
",3
    DB
08H,82H,0FH,44H,F8H,28H,08H,30H,0FH,CCH,00H,41H,10H,82H,09H,0CH;
```

```
    DB
FFH,F0H,00H,00H,00H,00H,FFH,FEH,05H,01H,08H,81H,10H,4FH,00H,00H;"
姚",0

    DB
01H,02H,01H,02H,02H,42H,04H,52H,0AH,4EH,12H,42H,22H,42H,C3H,FEH;
    DB
22H,42H,12H,42H,0AH,46H,04H,5AH,02H,42H,01H,02H,01H,02H,00H,00H;"
金",1

    DB
10H,00H,12H,7FH,12H,42H,FEH,42H,12H,42H,12H,7FH,10H,00H,00H,00H;
    DB
10H,00H,12H,7FH,12H,42H,FEH,42H,12H,42H,12H,7FH,10H,00H,00H,00H;"
喆",2
END
```

6. 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