

实验三 步进电机原理及应用

一、实验目的

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、实验内容

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。

三、实验步骤

学习 MC-51 汇编语言使用和步进电机原理，阅读数码显示器的电路图，重点理解步进电机的工作方式和数码显示方式。

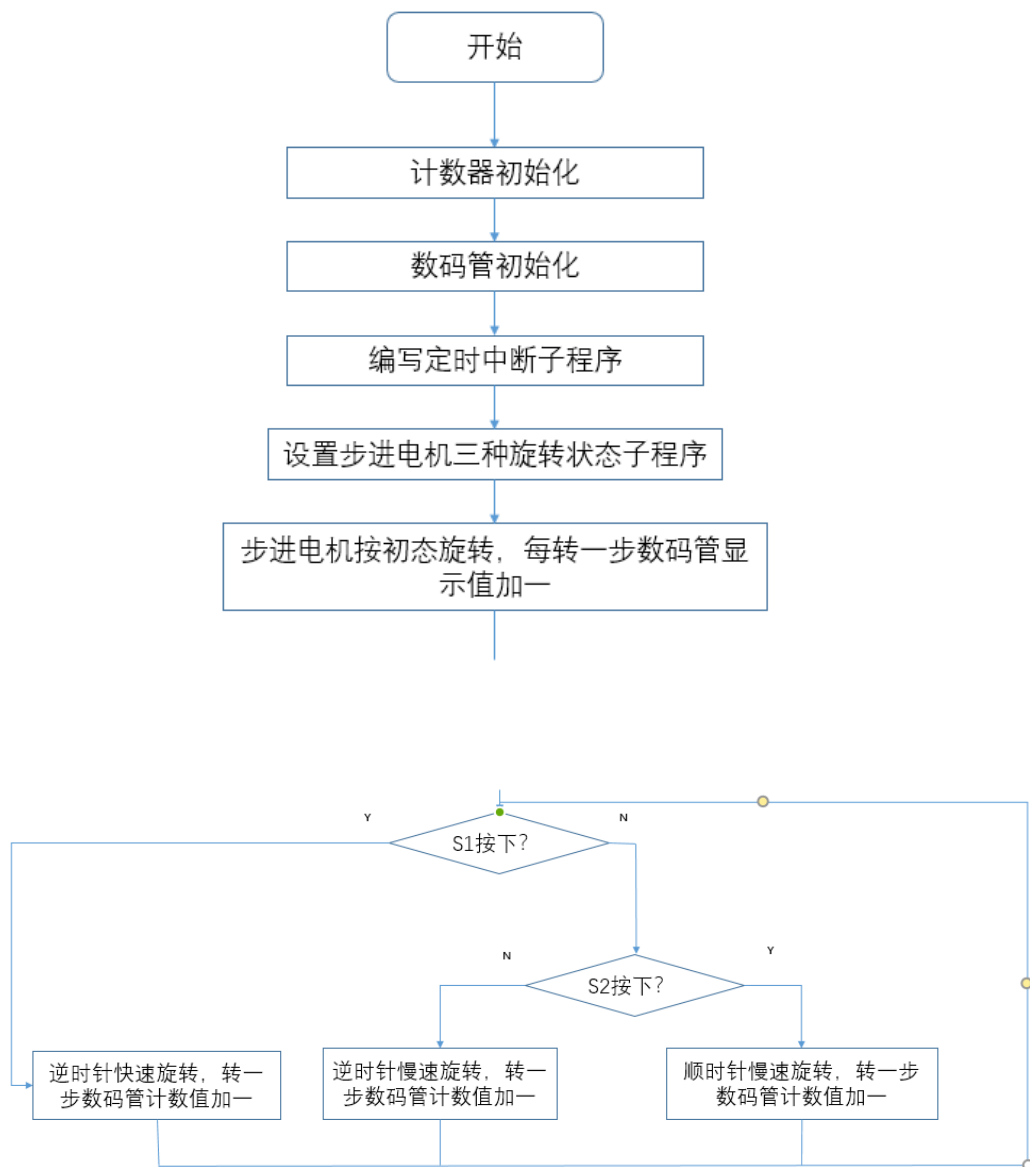
四、中断执行过程分析

当 CPU 检测到有中断触发信号后，首先执行完成当前执行的指令，将下一条指令的地址压入堆栈，

CPU 进入中断入口地址（CPU 自动从中断向量取得入口地址），开始执行中断服务程序（当然在中断服务程序中不能破坏正常运行程序的各种寄存器的状态和数据），

中断服务程序执行完成，最后执行一条 RETI 的中断返回指令，堆栈弹出原先保存的地址，完成中断服务程序，返回中断前的下一条指令地址继续执行正常程序。

五、实验流程图



六、实验代码

```

ORG 0000H
LJMP START
ORG 000BH
LJMP TO_INT
ORG 0040H
START:
    P4 EQU 0C0H
    P4SW EQU 0BBH
    ; MOV P4, 0FFH
    CLK EQU P4.4

```

```

    DAT EQU P4.5
    MOV P4SW, #30H
    SWH1 EQU P3.6
    SWH2 EQU P3.7
    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, #50
    MOV R5, #0
    MOV R6, #1
    SETB CE1
    SETB CE2
    SETB EA
    SETB ETO
    MOV TMOD, #01H
    MOV TLO, #3EH
    MOV TH0, #5DH
    SETB TR0
LL1:    LJMP LL1
TO_INT:
    PUSH ACC
    ; PUSH PSW
    ; PUSH DPL
    ; PUSH DPH
    CLR TR0
    MOV TLO, #3EH
    MOV TH0, #5DH
    SETB TR0
    DJNZ R3, IEND
    JNB SWH1, V1
    MOV R3, #5
    JMP V2
V1:    MOV R3, #1
V2:    LCALL DISPLAY
    LCALL STEP
IEND:
    ; POP DPH
    ; POP DLH

```

```
; POP PSW
POP ACC
```

```
RETI
```

```
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
```

```
SENDNUM:MOV R4, #8
```

```
SE1:   CLR CLK
      RLC A
      MOV DAT, C
      SETB CLK
      DJNZ R4, SE1
      RET
```

```
STEP:  JNB SWH2, SHUN
      CJNE R5, #1, N1
      CJNE R6, #1, N3
      CLR IN1
      SETB IN2
      MOV R5, #0
      MOV R6, #1
      LJMP ST0
```

```
N1:    CJNE R6, #1, N2
      CLR IN1
      CLR IN2
      MOV R5, #0
      MOV R6, #0
      LJMP ST0
```

```
N2:    SETB IN1
      CLR IN2
      MOV R5, #1
      MOV R6, #0
      LJMP ST0
```

```
N3:    SETB IN1
      SETB IN2
      MOV R5, #1
```

```

MOV R6, #1
LJMP ST0
SHUN: CJNE R5, #1, SH1
      CJNE R6, #1, SH3
      SETB IN1
      CLR IN2
      MOV R5, #1
      MOV R6, #0
      LJMP ST0
SH1:  CJNE R6, #1, SH2
      SETB IN1
      SETB IN2
      MOV R5, #1
      MOV R6, #1
      LJMP ST0
SH2:  CLR IN1
      SETB IN2
      MOV R5, #0
      MOV R6, #1
      LJMP ST0
SH3:  CLR IN1
      CLR IN2
      MOV R5, #0
      MOV R6, #0
      LJMP ST0
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
TABLE:
      DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H,
90H
      END

```

七、思考题

1. 如采用单四拍工作模式，每次步进角度是多少，程序要如何修改？
每次步进角度为 15 度。
A=IN1 , B=IN2 , (!A)表示 in1=0 , (!B)表示 in2=0
输出脉冲修改为：A→ B→ (!A)→ (!B)→ A
2. 如采用单双八拍工作模式，每次步进角度是多少，程序要如何修改？
每次步进角度为 7.5 度。
输出脉冲修改为：A→ AB→ B→ B(!A)→ (!A)→ !A!B→ !B→ (!B)A
3. 步进电机的转速取决于那些因素？有没有上、下限？
转速主要由时钟的周期控制，通过改变输入脉冲的个数决定转过的角度；
转速有上限，通过加大控制电压和降低线圈的时间常数可以提高上限；转速无下限。
4. 如何改变步进电机的转向？
通过反向 IN1 和 IN2 的输入即可，如将 01→ 11→ 10→ 00→ 01 改为 00→ 10→ 11→ 01→ 00
5. 步进电机有那些规格参数，如何根据需要选择型号？
规格参数：最大工作电压、最小启动电压、最大允许功耗和工作频率等。
根据实际需要选择型号
6. MCS51 中有哪些可存取的单元，存取方式如何？它们之间的区别和联系有哪些？
 - (1) 工作寄存器组 (00H-1FH)
 - (2) 可位寻址 RAM 区 (20H-2FH)
 - (3) 通用的 RAM 区 (30H-7FH)
7. 说明 MOVC 指令的使用方法。
MOVC 用来读取程序存储器；以 16 位的程序计数器 PC 或数据指针 DPTR 作为基寄存器，以 8 位的累加器 A 作为变址寄存器，基址寄存器和变址寄存器的内容相加作为 16 位的地址访问程序存储器。如：
MOVC A, @A+PC
MOVC A, @A+DPTR
8. MCS51 的指令时序是什么样的，哪类指令的执行时间较长？
一个机器周期包含 6 个状态 (S1-S4)，每个状态分为两个节拍 P1 和 P2，通常，一个机器周期会出现两次高电平 S1P2 和 S4P2，每次持续一个状态 S。乘法及除法指令占 4 个周期，三字节指令均为双周期指令。
9. 在本实验环境下，能否控制显示数码的亮度？如何实现？
能，通过修改刷新频率

实验四 LED 点阵显示屏

一、实验目的

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。

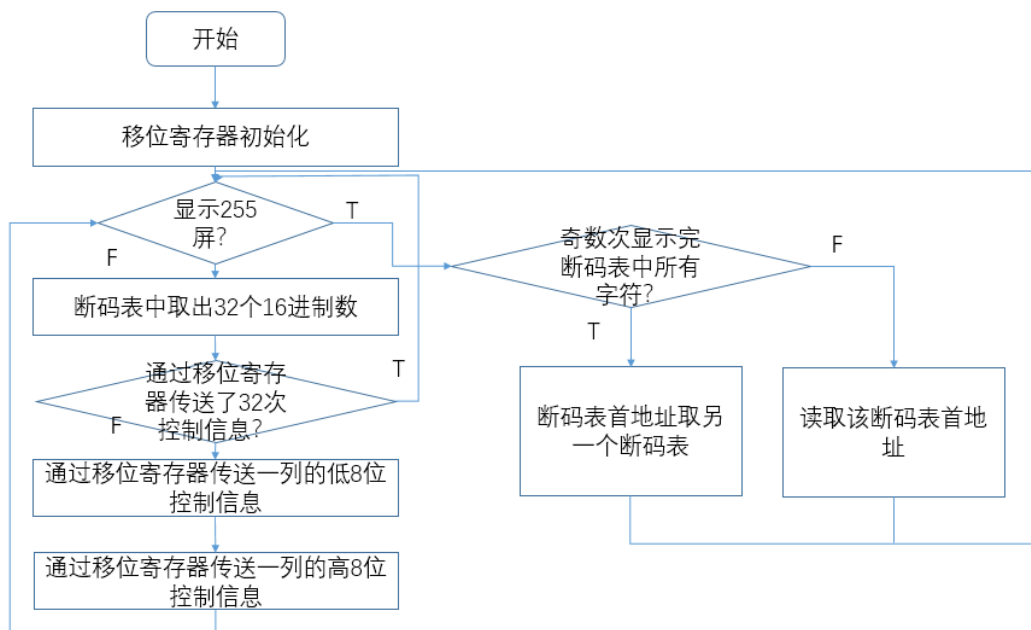
二、实验内容

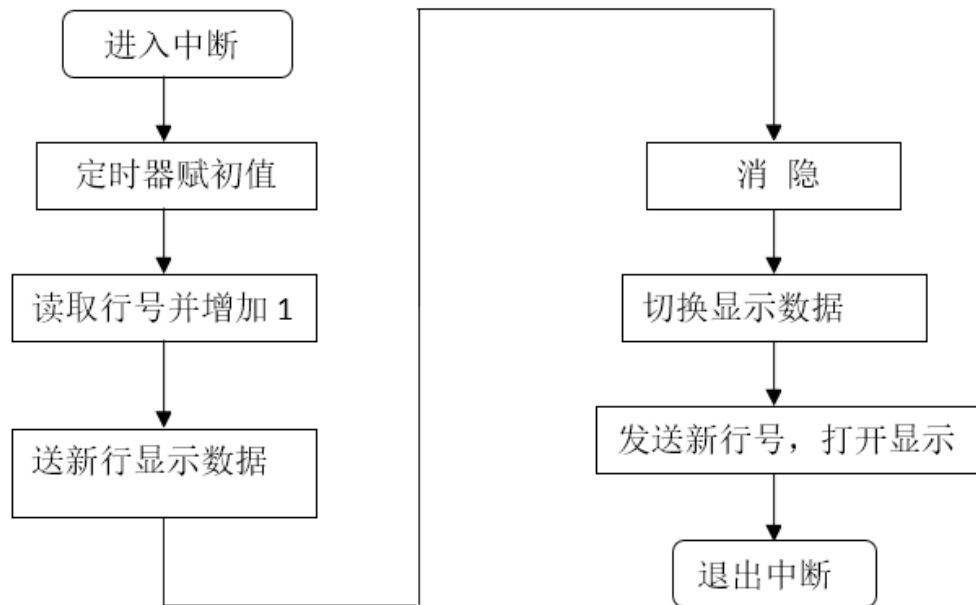
1. 了解 16*16 点阵电路的原理，编写汇编语言程序。
2. 编写一行汉字字符(至少三个字)的显示程序。
3. 能够从左到右(或从右到左)循环显示(要求显示过程中字的大小与屏幕尺寸相适应)。

三、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

四、实验流程图





五、实验代码

```

ORG 000H
LJMP START
ORG 0040H

```

START:

```

DX EQU P0.0 ; 行数据口
DY EQU P0.3 ; 列数据口
CLKYWX EQU P0.1 ; 行移位寄存器时钟
CLKYWY EQU P0.5 ; 列移位寄存器时钟
CLKCCX EQU P0.2 ; 行存储器时钟
CLKCCY EQU P0.6 ; 列存储器时钟

```

```

OUTX EQU P0.7 ; 行输出使能
OUTY EQU P0.4 ; 列输出使能

```

;.....

SM: ; 无限循环

```

MOV R0, #0
MOV R1, #1

```

```

MOV R4, #1 ; table 高 8 位指针
MOV R5, #0 ; table 低 8 位指针

```

;.....逐行扫描.....


```

MOV R3, #16 ; 扫描 16 次
SM16: SETB OUTX ; 行输出使能置高电平
      SETB OUTY ; 列输出使能置高电平

; ..... 送行扫描码.....

      CLR CLKCCX ; 列存储器时钟置低电平

      MOV DPTR, #TABLE1

      MOV A, R0
      MOVC A, @A+DPTR
      MOV R6, #8
YW1:   CLR CLKYWX
      RLC A
      MOV DX, C
      SETB CLKYWX ; 将高 8 位列选码按位送入到移位寄存器中
      DJNZ R6, YW1

      MOV A, R1
      MOVC A, @A+DPTR
      MOV R6, #8
YW0:   CLR CLKYWX
      RLC A
      MOV DX, C
      SETB CLKYWX ; 将低 8 位列选码按位送入到移位寄存器中
      DJNZ R6, YW0

      SETB CLKCCX ; 将移位寄存器中的数据送到存储器中

      CLR OUTX ; 将行输出使能置低电平

; ..... 送列扫描码.....

      CLR CLKCCY ; 列存储器时钟置低电平

      MOV DPTR, #TABLE
      MOV A, R4
      MOVC A, @A+DPTR
      MOV R6, #8
YW3:   CLR CLKYWY
      RRC A
      MOV DY, C

```

SETB CLKYWY ; 将高 8 位列选码按位送入到移位寄存器中
DJNZ R6, YW3

MOV A, R5
MOVC A, @A+DPTR
MOV R6, #8

YW2: CLR CLKYWY

RRC A
MOV DY, C
SETB CLKYWY ; 将低 8 位列选码按位送入到移位寄存器中
DJNZ R6, YW2

SETB CLKCCY ; 将移位寄存器中的数据送到存储器中

CLR OUTY ; 将列输出使能置低电平

LCALL DELAY1

;.....更新扫描码.....

INC R0
INC R0
INC R1
INC R1

INC R4 ; TABLE 指针 R4, R5 分别加 2
INC R4
INC R5
INC R5

DJNZ R3, SM16 ; 进行下一次扫描

LJMP SM ; 重新扫描

;.....延时函数.....

DELAY1:

MOV R6, #20
DEL1: MOV R2, #20
DEL2: DJNZ R2, DEL2
DJNZ R6, DEL1

RET

;.....扫描码表.....

TABLE:

DB 0FFH, 0FFH, 0F7H, 0BFH, 0CFH, 0BFH, 0DBH, 0BFH, 0DBH, 0BFH, 0DBH,
0BDH, 5BH, 0BEH, 9BH, 01H;

DB 0DAH, 0BFH, 0D9H, 0BFH, 0DBH, 0BFH, 0DFH, 0BFH, 0D7H, 0BFH, 0CFH,
0BFH, 0FFH, 0BFH, 0FFH, 0FFH; "字", 0

TABLE1:

DB 80H, 00H
DB 40H, 00H
DB 20H, 00H
DB 10H, 00H
DB 08H, 00H
DB 04H, 00H
DB 02H, 00H
DB 01H, 00H

DB 00H, 80H
DB 00H, 40H
DB 00H, 20H
DB 00H, 10H
DB 00H, 08H
DB 00H, 04H
DB 00H, 02H
DB 00H, 01H

END

六、思考题

1. 如何使用软件调整和控制 LED 点阵的亮度
通过软件调整扫描频率来控制
2. 如何尽量避免显示过程中的闪烁
提高扫描频率(24hz 以上)
3. 如何将本实验的软硬件推广到多行多列的 LED 显示屏(如 64*1280)
硬件方面: 添加新的 LED 以及 74HC59 来实现
软件方面: 将控制行扫描的 16 位数字 0fffeH 改为 64 位的
0fffffffffffffffffeH, 将读入列值的 2 字节改为 160 字节, 及重复输出

1280bit，结束后令行的输出移位一次。