

实验三步进电机原理及应用

一：实验目的和要求

初步学习和掌握 MCS51 的体系结构和汇编语言, 了解 Kei 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。

二、实验环境和开发环境

- (1) 单片机测控实验系统
- (2) 步进电机控制实验模块
- (3) Keil 开发环境
- (4) STC-ISP 程序下载工具

三、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。

四、实验步骤

参考附录二、附录三和 expr/资料/原理的辅助材料, 学习 MCS-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管显示方式。

五：实验代码：

```
ORG 0000H
    LJMP START
ORG 000BH
    LJMP T_INT
ORG 0040H
START:
    P4 EQU 0C0H
    MOV P4, #0FFH
    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 ;01->11->10->00->01
IN2 EQU P1.0 ;01->11->10->00->01
```

```
SETB CE1
SETB CE2
CLR IN1
CLR IN2
```

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

```
MOV TMOD, #01H
MOV IE, #10000010B
MOV TMOD, #01H
MOV TLO, #LOW(65536-41667)
MOV TH0, #HIGH(65536-41667)
SETB TRO
SETB ETO
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:

```
JNB IN1, Z1
JNB IN2, Z2
CPL IN1
JMP BUJIA
```

Z1:

```
JNB IN2, Z3
```

```
CPL IN2
JMP BUJIA
Z2:
CPL IN2
JMP BUJIA
Z3:
CPL IN1
JMP BUJIA

FAN:
JNB IN1, F1
JNB IN2, F2
SETB IN1
CLR IN2
JMP BUJIA
F1:
JNB IN2, F3
SETB IN1
SETB IN2
JMP BUJIA
F2:
CLR IN1
CLR IN2
JMP BUJIA
F3:
CLR IN1
SETB IN2

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

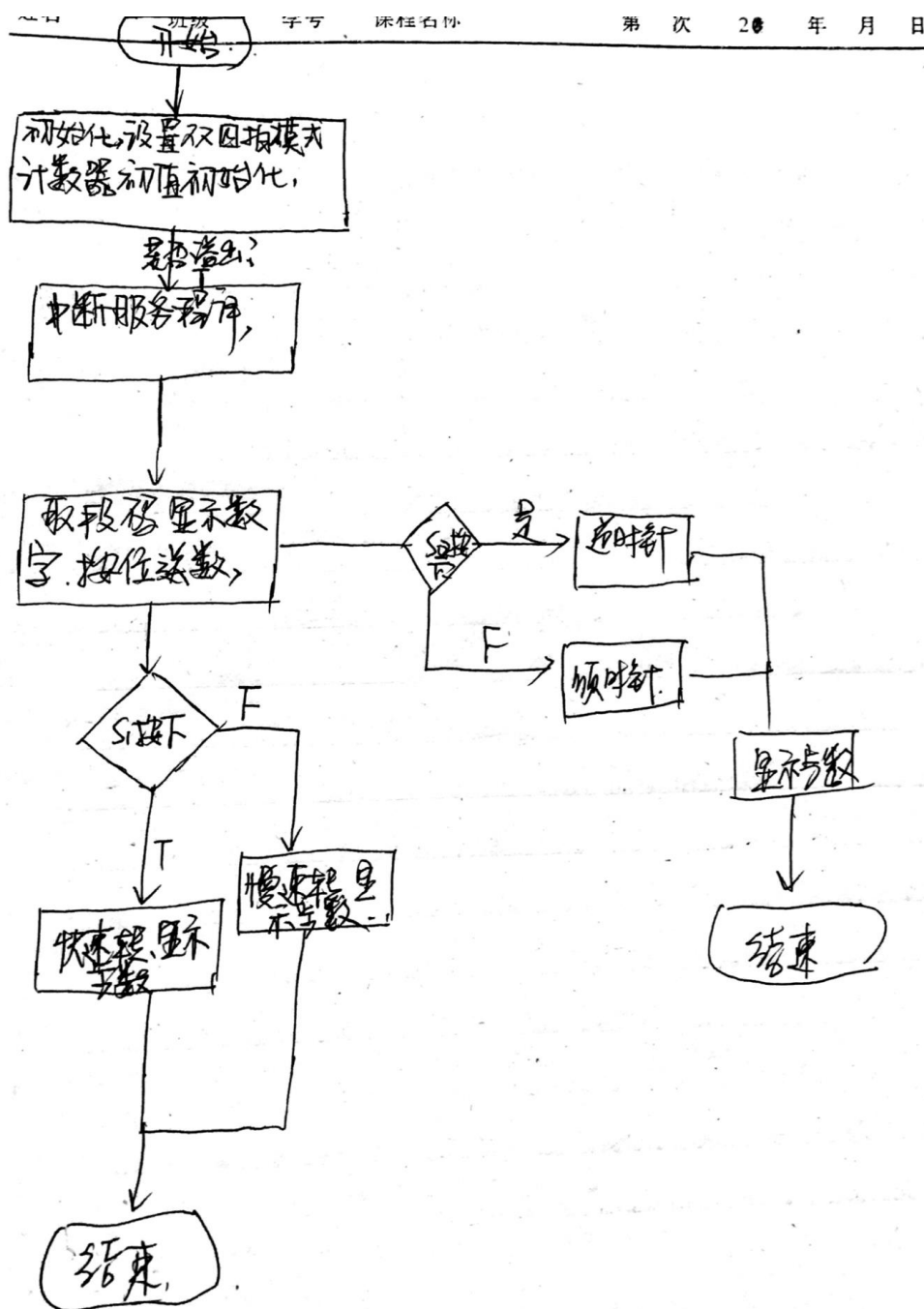
DISPLAY:
    MOV R0, #8
D1:
    CLR CLK
    RLC A
    MOV DAT, C
    SETB CLK
    DJNZ R0, D1
    RET

T_INT:
    MOV R0, #1
    MOV TLO, #LOW(65536-41667)
    MOV TH0, #HIGH(65536-41667)
    RETI

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

```

六：实验流程图：



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?

答: 每次步进角度是 15 度。

设 $A = in1, B = in2$, $(!A)$ 表示 $in1=0$, $(!B)$ 表示 $in2=0$

输出脉冲修改为: $A \rightarrow B \rightarrow (!A) \rightarrow (!B) \rightarrow A$

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?

答: 每次步进角度是 7.5 度。

输出脉冲修改为: $A \rightarrow AB \rightarrow B \rightarrow B(!A) \rightarrow (!A) \rightarrow !A!B \rightarrow !B \rightarrow (!B)A$

3. 步进电机的转速取决于那些因素? 有没有上、下限?

答: 步进电机的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?

答: 通过反向 IN1 和 IN2 的输入即可, 如将 $01-11-10 \rightarrow 00 \rightarrow 01$ 改为:

$00 \rightarrow 10-1 \rightarrow 01 \rightarrow 00$

5. 步进电机有那些规格参数, 如何根据需要进行选择型号?

答: 步进电机的主要参数有最大工作电压、最小启动电压、最大允许功耗和工作频率等。

5. 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?

答: (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)

6. 说明 MOV C 指令的使用方法。

答: MOV C 用来读取程序存储器; 以 16 位的程序计数器 PC 或数据指针 DPR 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:

MOV C A, GA+PC

MOV C A, @ADPTR

7. MCS51 的指令时序是什么样的, 哪类指令的执行时间较长?

答: 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

8. 在本实验环境下, 能否控制显示数码管的亮度? 如何实现?

答: 能, 通过修改刷新频率

八. 体会与收获

经过这次试验后, 我对中断有了更深刻的理解, 理解到了本实验中断服务程序作用原理, 当计时器计数到时, 则会自动发生中断, 而不是人为的去控制中断。在这个过程中, 我也遇到过很多挫折, 比如 g 值的计算, 计数器初值的设定等等通过这次试验, 我对步进电机的原理和应用有了进一步的理解, 同时也复习了上学期数码管的相关知识。

实验四 LED 点阵实验、

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

- 三，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)。

- 四、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。

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
OFFH, OFFH, 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

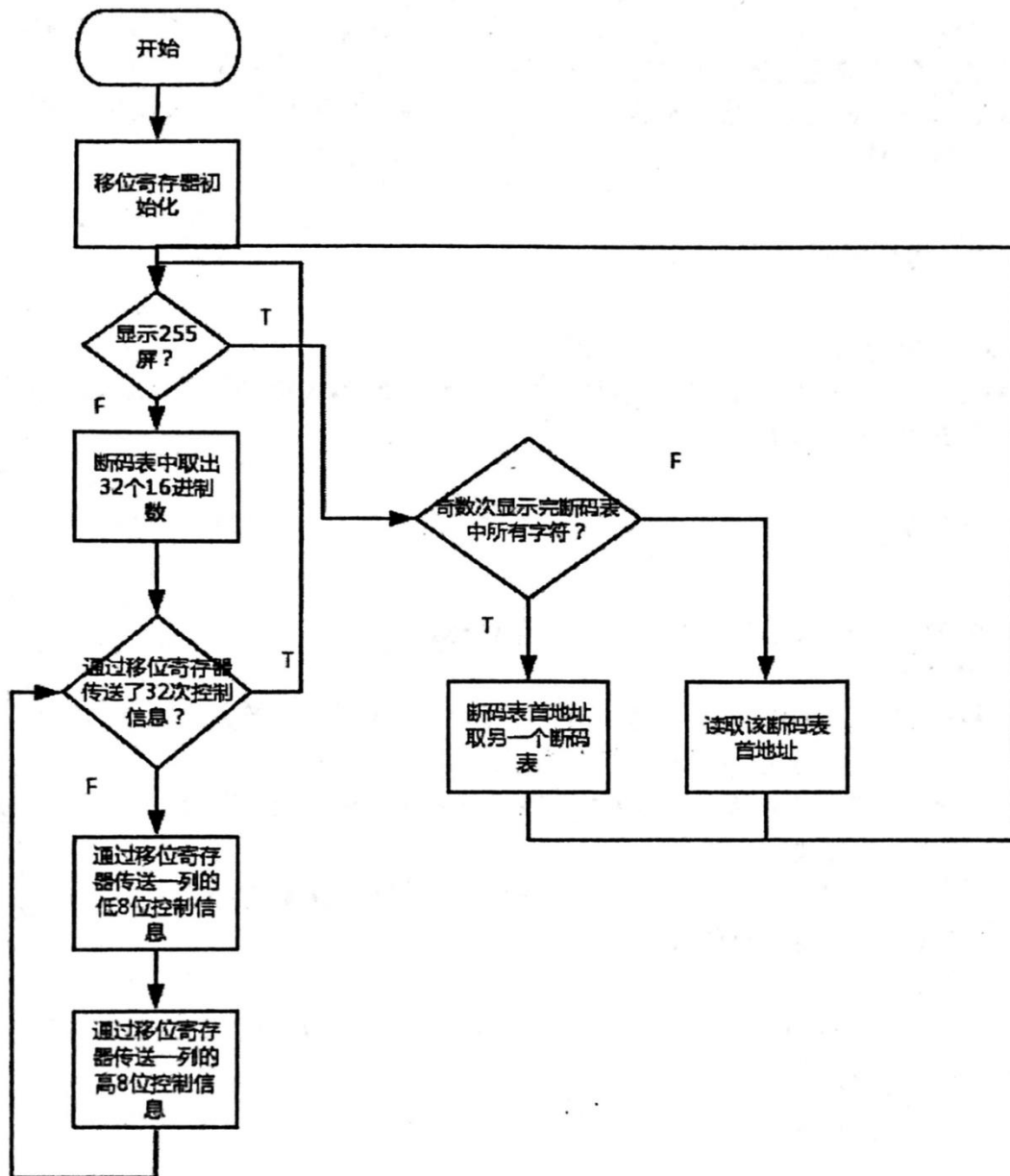
DB 00H, 04H

DB 00H, 02H

DB 00H, 01H

END

六：实验流程图



七：思考题

1. 如何使用软件调整和控制 LED 点阵的亮度
可以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, 一般刷新频率提高到 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)
硬件方面可以通过添加新的 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 0fffffffffffffe1 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

八：实验收获与心得：

循环显示的主要思想是一个虚拟的视窗，可以把所有要显示的字符看作一个 $16 \times N$ 的数组，而这个虚拟的视窗大小为 16×16 （即点阵的大小，视窗从数组的左端开始向右滑动，每次把视窗内的数位输出到点阵中，若视窗内的数位超出了数组的右端，则超出的数位从左端开始计算，本次实验使我掌握了点阵 LED 显示屏的控制方法。让我对汇编程序有了更深一步的应用与理解。