

# 实验三 步进电机原理及应用

姓名：廖歆恺 班级：八班 学号：21160832 教学号：53160832

## 一、实验原理

步进电机工作在双四拍工作模式。三个数码管的显示由 3 个 74HC164 级联控制。单片机的 P4.5 引脚作为模拟串口数据，P4.4 作为模拟串口时钟，CLR 端接高电平。将前一个 74HC164 的 Q7 作为后一个 74HC164 的输入端。

## 二、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-ISP 程序下载工具

## 三、实验内容

编写 MCS-51 程序，使用定时器中断实现，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。

其中，当按下 S1 开关时，旋转速度为 60 转/分；当松开时，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。

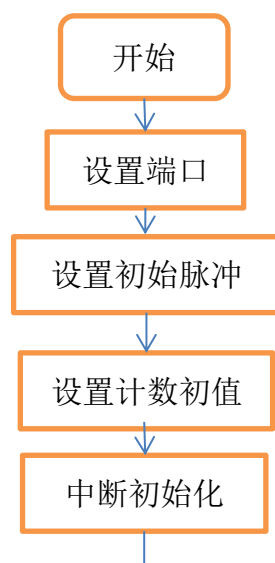
## 四、实验过程

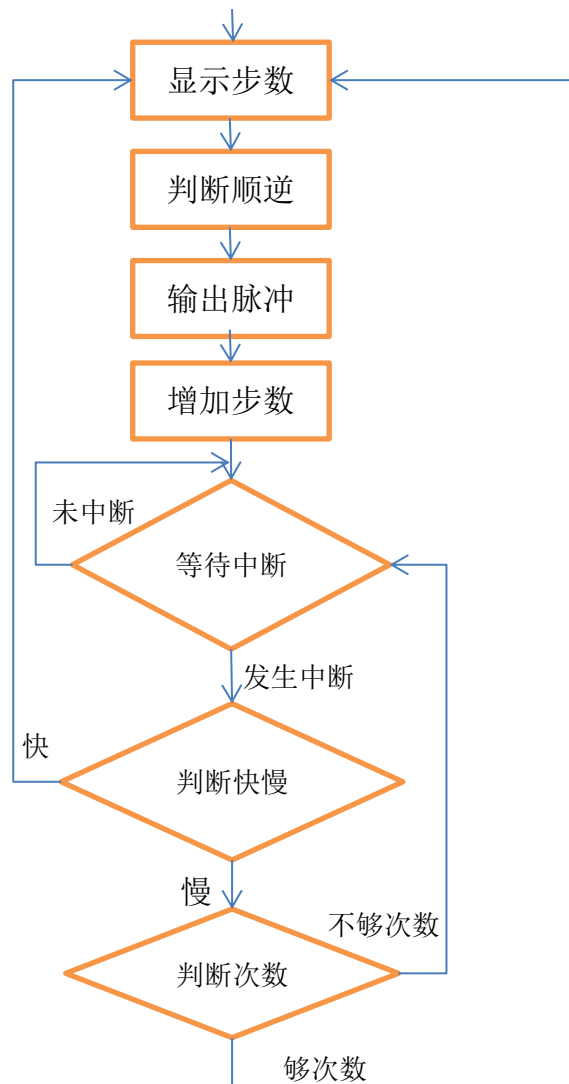
1. 阅读附录资料。2. 在 Windows 下启动软件。3. 创建工程，编写简单代码并调试。4. 编写完成功能的代码，运行调试。5. 完成思考题。

## 五、实验结果

所有开关处于断开状态时，步进电机按逆时针，以 10 转每分钟的速度旋转；按下 S1 后，步进电机的旋转速度变为 60 转每分钟；按下 S2 后，步进电机的旋转方向变为顺时针方向。同时三个数码管按百十个位的顺序显示已转步数。超过 999 步后从 0 重新计算。

## 六、流程图





## 七、实验代码

```
ORG 0000H
```

```
LJMP START
```

```
ORG 000BH
```

```
LJMP T0IN;中断向量表
```

```
ORG 0040H
```

```
START:;初始化
```

```
    P4 EQU 0C0H;P4 端口的地址
```

```
    MOV P4, #0FFH
```

```
    P4SW EQU 0BBH
```

```
    MOV P4SW, #70H
```

```
DAT EQU P4.5
```

```
CLK EQU P4.4;时钟
```

SW1 EQU P3.6;开关 S1

SW2 EQU P3.7;开关 S2

CE1 EQU P1.1 ;引脚相连

CE2 EQU P1.4

IN1 EQU P3.2 ;01-→11-→10-→00-→01, 输出脉冲

IN2 EQU P1.0 ;01-→11-→10-→00-→01

SETB CE1;设置工作模式

SETB CE2

CLR IN1;设置初始脉冲

CLR IN2

MOV R6, #0;个位

MOV R5, #0;十位

MOV R4, #0;百位

MOV TMOD, #01H;和定时器/计数器相关的特殊功能寄存器, 通过对  
;它的设置和读写就可以控制使用定时器/计数器, 设置定时器工作方  
;式

MOV IE, #10000010B;中断允许寄存器

MOV TMOD, #01H

MOV TL0, #LOW(65536-41667) ;计数器初值, 快的情况 23869

MOV TH0, #HIGH(65536-41667)

SETB TR0;运行控制位, 控制一个定时器是否允许计数

SETB ET0;中断允许位, 为 1 时允许中断

SETB EA ;CPU 中断允许标志

MAIN:

MOV A, R6

MOV DPTR, #TAB

MOVC A, @A+DPTR

LCALL DISPLAY;显示步数

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: ; 逆时针
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: ; 顺时针
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:;十进制
    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:;等待中断

MOV R3, #6H;慢的情况下，需要计数 6 次才能满足时间要求

TT1:

MOV R0, #0

DELAY:

CJNE R0, #1, DELAY

JNB SW1, OUT

DJNZ R3, TT1

OUT:

JMP MAIN

DISPLAY:;显示函数

MOV R0, #8

D1:

CLR CLK

RLC A

MOV DAT, C

SETB CLK

DJNZ R0, D1

RET

T0IN:;中断

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;0~9  
END

## 八、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？

每次步进角度不变，依然是 15 度；设 A 为 IN1=1，B 为 IN2=1，! 号表示取反，修改输出脉冲 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？

每次步进角度变为 7.5 度。设 A 为 IN1=1，B 为 IN2=1，! 号表示取反，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？

其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；转速有上限无下限。

4. 如何改变步进电机的转向？

将序列翻转，就得到相反方向的输出脉冲。

5. 步进电机有那些规格参数，如何根据需要进行选择型号？

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、空载启动频率和工作频率等。可以根据定位精度和最大速度进行选择。

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) 64K 字节程序存储器空间 (0—0FFFFH)

(2) 64K 外部数据存储器空间 (0—0FFFFH)

(3) 256 字节内部 RAM 空间 (0—0FFH)

(4) 256 位寻址空间 (0—0FFH)

(5) 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. 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。

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？

能控制，方法是通过修改刷新频率来实现。

## 九、感悟

首先一点是，要确定计数器初值，根据所给公式和晶振条件，得出初值约为 23869。本次实验中接触到了大量的 MCS51 的指令，同时也体验了实验板操作。在此次编写代码过程中，有一个很重要的问题，那就是在中断调用后，刷新计数器初值，那么显示和调整转速转向由谁来控制？无论交给谁，执行时都有被打断的可能性。不过鉴于中断程序应该简短的规定，所以中断程序只负责给计数器赋值。如果要完善功能的话，可以考虑中断时设置一个累加器，在显示步数前查询累加器的值以便统一步数。

中断控制字的设置，指令 MOV IE, #10000010B; EA 置位，允许中断。ET0 置位，允许定时器 T0 中断。SETB ET0 与 SETB EA 完成同样的功能。

# 实验四 LED 点阵显示屏

姓名：廖歆恺 班级：八班 学号：21160832 教学号：53160832

## 一、实验原理

通过编程控制由高亮度 LED 发光管构成的点阵，可以显示字符以及图形。可以通过自行编写获得点阵数据，也可以从标准字库中提取要显示的字符点阵数据。从字库提取时需掌握字库的编码方法和字符定位的计算。

## 二、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

## 三、实验内容

学习 16\*16 点阵电路的原理。编写显示一行至少三个汉字字符的汇编语言程序。其显示方法为从左到右（或从右到左）循环显示，并且所显示字的宽高与屏幕尺寸相适应。

## 四、实验过程

1. 了解 LED 点阵显示器的控制方法；
2. 使用 MCS-51 汇编语言编程，使得 LED 点阵显示器显示出正确

的汉字字符及动态效果；

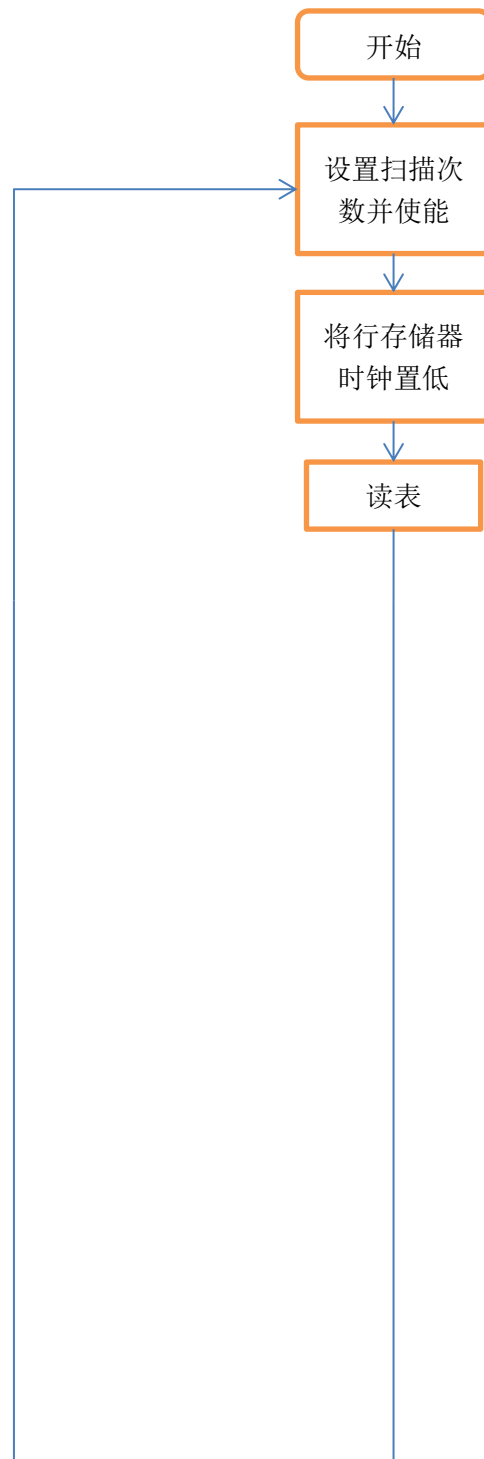
3. 将编译后的程序下载到 51 单片机。

4. 观察 LED 显示屏的显示结果。

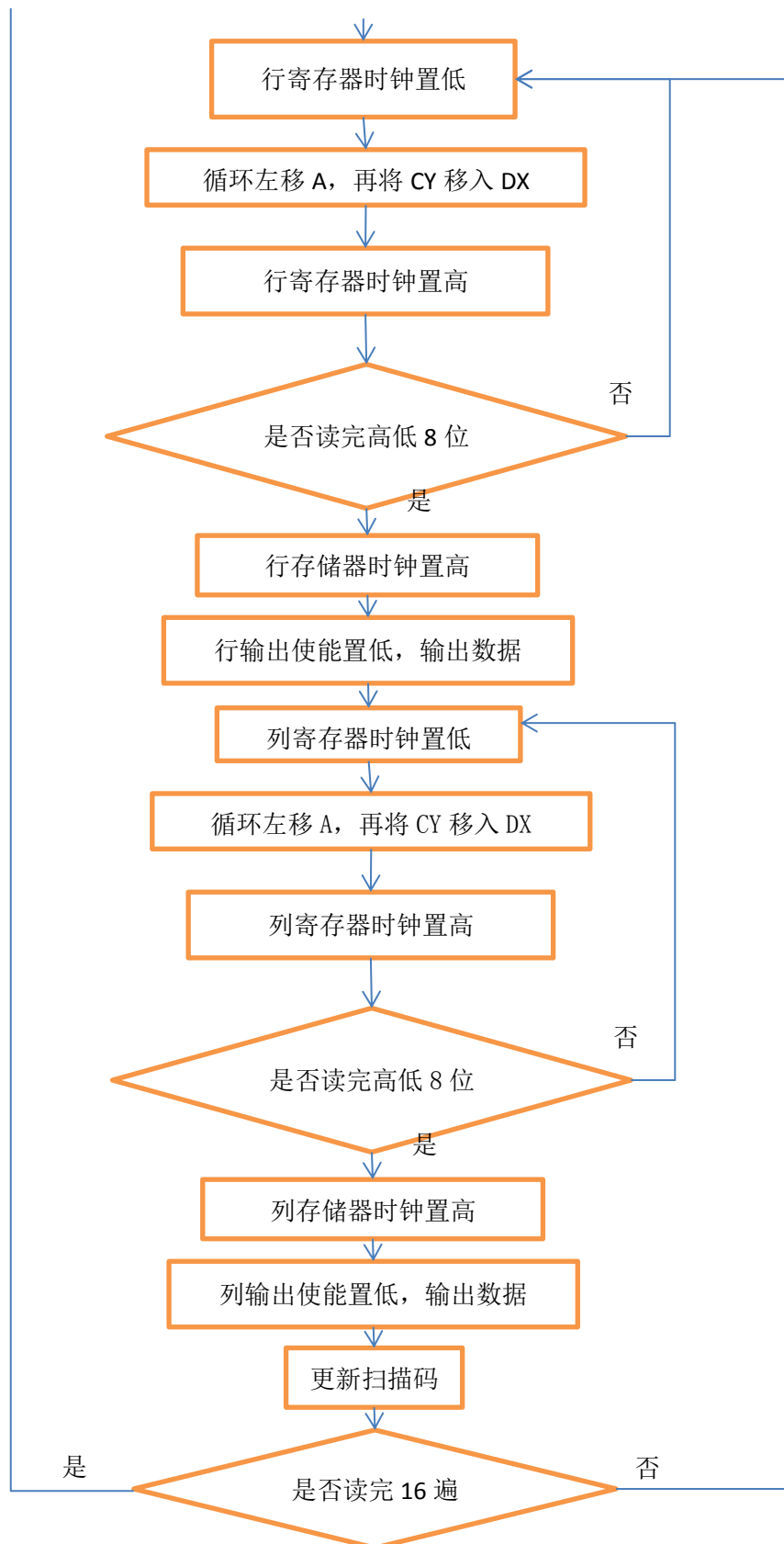
## 五、实验结果

LED 点阵显示器上从左至右循环显示“廖歆恺”三个字。

## 七、流程图







## 七、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

MOV R3, #0FEH

MOV R7, #0

.....

TB:

INC R3

INC R3

CJNE R3, #0, TTT

TTT:

MOV B, R3; 运算器，通用寄存器 B 寄存器

PUSH B

MOV R6, #0FH

SM: ;扫描

POP B

PUSH B

MOV R3, B

MOV B, R6

PUSH B

MOV B, R3

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 ;行存储器时钟置低电平

MOV DPTR, #TABLE1;数据指针 DPTR

MOV A, R0;R0 初值为 0
MOVC A, @A+DPTR
MOV R6, #8
YW1:
    CLR CLKYWX;行移位寄存器时钟置低电平
    RLC A;将 A 和进位标志一起向左循环移位 1 位。位 7 移入 CY,
        ;CY 移入位 0。不影响标志。
    MOV DX, C;C 是布尔累加处理器, CY
    SETB CLKYWX ;将高 8 位行选码按位送入到移位寄存器中
    DJNZ R6, YW1;这组指令把源操作数减 1, 结果回送到源操作数中;
        ;如果结果不为 0 则转移。
    MOV A, R1;R1 初值为 1
    MOVC A, @A+DPTR;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。

    MOV R6, #8
YW0: CLR CLKYWX
    RLC A
    MOV DX, C
    SETB CLKYWX ;将低 8 位行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

LCALL DELAY1

;.....送列扫描码

CLR CLKCCY ;列存储器时钟置低电平

MOV DPTR, #TABLE

MOV A, R4

ADD A, B

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

ADD A, B

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

        JMP START
        POP B
        MOV R6, B
        DJNZ R6, SM
        POP B
        MOV R3, B

        INC R7
        CJNE R7, #32, TEST

TEST:
        JMP TB
;..... 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 0FEH, 0FFH, 80H, 03H, 0FFH, 0FBH, 0C1H, 0BH, 0DBH, 6BH, 0D7H, 5BH  
DB 0DAH, 6BH, 0F9H, 7BH;  
DB 0E7H, 9BH, 9EH, 0E3H, 0FBH, 3BH, 0FDH, 0CBH, 0EEH, 7DH, 0F3H, 8DH  
DB 0FCH, 7EH, 0FFH, 8FH; "廖", 0

DB 0FBH, 0F7H, 0FBH, 0EFH, 0FBH, 00H, 81H, 0FFH, 0BDH, 0BDH, 0DEH  
DB 0DBH, 0F7H, 00H, 0F7H, 0FFH;  
DB 0F7H, 81H, 0F7H, 0BDH, 0EBH, 0BDH, 0EBH, 81H, 0EDH, 0BDH, 0DDH  
DB 0BDH, 0DEH, 81H, 0BFH, 3DH; "歆", 1

DB 0FBH, 0F7H, 0DBH, 77H, 0DBH, 77H, 0DBH, 77H, 0C0H, 67H, 0FFH, 0D5H  
DB 0C0H, 75H, 0DFH, 0F5H;  
DB 0DFH, 0F6H, 0C0H, 77H, 0FFH, 77H, 0FFH, 77H, 0BFH, 77H, 0BFH, 77H  
DB 80H, 0F7H, 0FFH, 0F7H; "恺", 2

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

## 八、思考题

1. 如何使用软件调整和控制 LED 点阵的亮度减少延时时间。
2. 如何尽量避免显示过程中的闪烁加大扫描频率。
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)将本实验中的行列以及扫描码作为参数, 当要考虑具体情况时, 根据实际需求设置参数即可。

## 九、感悟

本实验的难度较大, 一是在于理解教程中对芯片引脚的操作, 明白在什么时机下, 数据将被送到总线上。根据操作手册上的指令引导去设计整个程序的流程; 二是在于理解 PCtoLCD2002 软件所生成字模的含义。首先要依据所设计好的程序流程去设计字模。本次实验中, 选择阳刻更为合适。其次取模方式需明确是逐行或是逐列等。最后要理解软件所生成的字模到底是什么含义。我在这上面迷惑了一段时间。最终明白了, 在逐行方式下所生成的字模是行码, 依照教程说法, 当要点亮  $(i, j)$  时, 行  $i=0$ , 列  $j=1$ 。所以事实上, 只需要行码即可, 列码全部设为 1。当看到自己的名字在 LED 上滑过时, 我的心情十分激动。