

《单片机控制与应用实验》

实 验 报 告

学 号: 21160817

教 学 号: 53160817

姓 名: 商健文

学 院: 计算机科学与技术学院

专 业: 计算机科学与技术

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

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、实验原理

单片机系统的频率是 12M；步进电机转动一周需要 24 步。

本实验使用简单的双四拍工作模式，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。

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

计算想要设定的计数器初值 s ，使用如下方程：

$$(2^{\text{定时器最大位数}} - s) \times \text{定时周期} = t$$

$$\text{定时周期} = 12/\text{CPU 晶振频率}$$

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）。

在本题中，快速旋转时转速为 60 转/分，即为 1 转/秒。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，所以步进电机转动一步需要 1/24 秒，这样的话通过上面的公式我们就可以算出在快速旋转状态下需要被设定的计数器初值。即定时周期 = $12/12 \times 10^{-6} = 10^{-6}$ 。定时器最大位数为 16，所以公式可写为 $(2^{16} - s) \times 10^{-6} = 1/24$ ，可得 $s = 23870$ （十进制数），转换为十六进制即为 5D3EH，所以需要将该定时器的计数初值设为 5D3EH 来控制步进电机快速旋转。

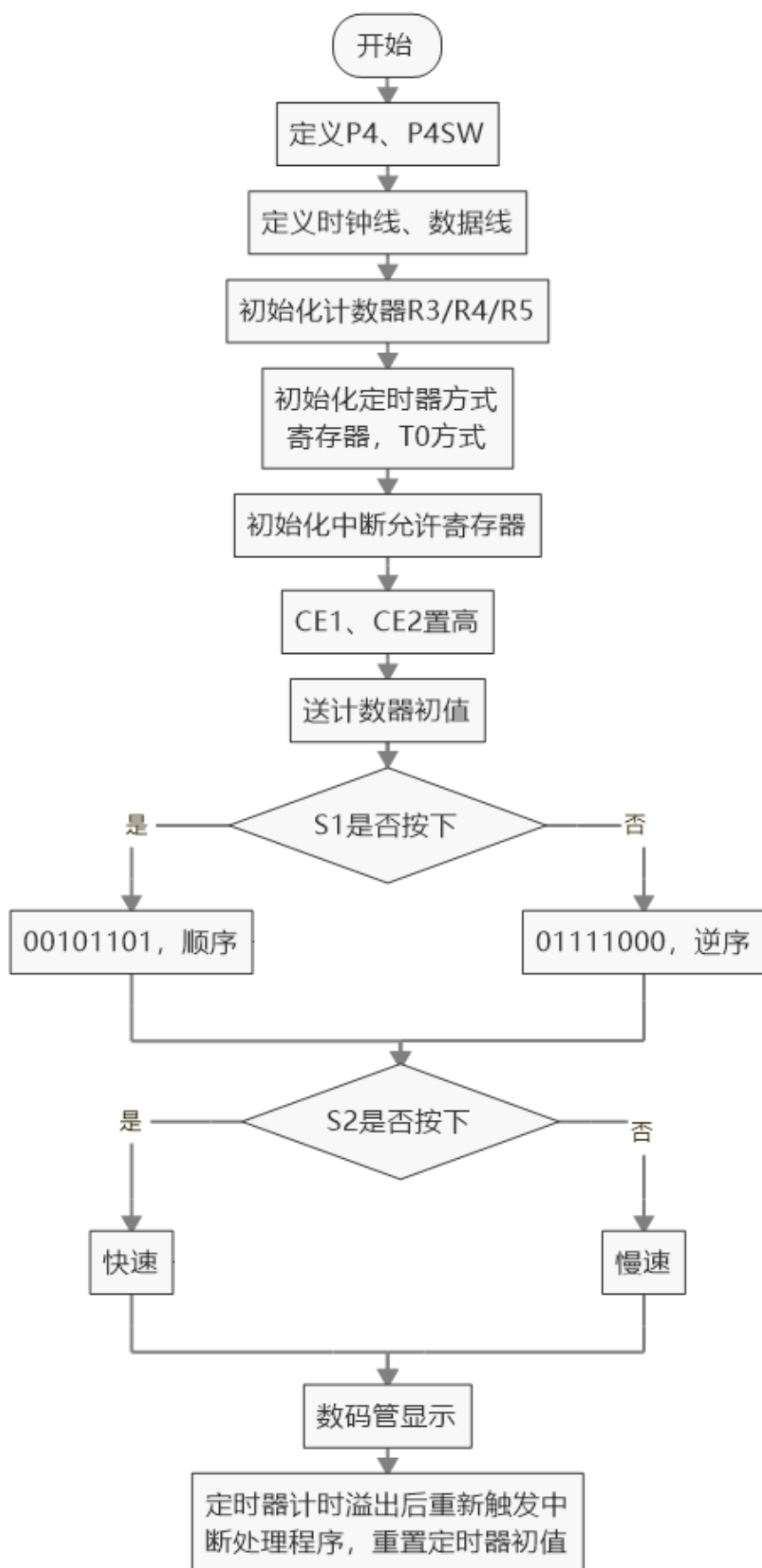
本开发平台有 3 个数码管，使用串行方式连接在一起，具体电路参见实验原理。要想输出一个字形码，就需要从高位到低位依次向移位寄存器输出 8 个比特。移位寄存器的数据线和时钟线分别接到单片机的 P4.5 和 P4.4 管脚，可以使用 MCS-51 里面的位操作指令进行输出。连续输出 3 个字形，24 个 bit 之后，欲显示的字形将稳定地显示在数码管上，程序可以转而执行其他工作。

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。

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。

五、实验代码与流程图

程序流程图：



实验代码:

```
ORG 0000H                ;复位起始地址
    LJMP START
ORG 000BH                ;中间地址保留给中断向量表
    LJMP EINT0            ;定时器 0 中断程序入口地址
ORG 0040H                ;程序实际起始地址
START:
    P4    EQU 0C0H
    P4SW EQU 0BBH
    CLK EQU P4.4  ;时钟线
    DAT EQU P4.5  ;数据线
    SW EQU  P3.6

    MOV P4SW,#70H
    MOV DPTR,#TAB        ;s = 23869  --5D3E

LP:
    MOV R3,#0            ;计数
    MOV R4,#0
    MOV R5,#0
I1:  MOV TMOD,#01H        ;选择工作方式, 即对 TMOD 赋初值。T0 工作在方式 1,
    ;16 位的计数器    GATE(GATE 一门控位, 控制定时器的两种启动方式,)等于 0, 不受外部
    ;控制

    MOV IE,#82H          ;全局中断, T0 中断允许 中断控制字;直接对中断允许寄存
    ;器 IE 和 优先级寄存器 IP 设置
    ;ORL IP,#2H          ;逻辑或, T0 中断优先级高

    SETB P1.1            ;CE1 置高
    SETB P1.4            ;CE2 置高

NEXT:
    JB  P3.7,OPP          ;如果 P3.7 等于 1 则转移
    MOV R0,#00101101B    ;按下, 顺时针
    MOV 20H,R0

    LJMP SS1;不会再执行 OPP 的内容
OPP:  MOV R0,#01111000B    ;松开, 逆时针
    MOV 20H,R0
SS1:
    JB  P3.6,SPD          ;如果 P3.6 等于 1 则转移
    MOV R2,#0H            ;按下, 快速        ;23870 5D3E
    LJMP L0                ;不会再执行 SPD 的内容
SPD:  MOV R2,#1H          ;松开, 慢速
```

```

L0:    MOV R1,#4
        MOV R0,20H
L1:    MOV A,R0
        RLC A                ;循环左移操作
        MOV P3.2,C           ;IN1
        RLC A
        MOV P1.0,C           ;IN2
        MOV R0,A
        LCALL NUM
        LCALL TIME
        DJNZ R1,L1

        LJMP NEXT

TIME:
        CJNE R2,#1,QUICK      ;如果 R2 等于 1，则跳转到快速
        MOV R6,#6             ;慢速                ;分 6 次来计时，通过把一
分钟分成 6 次来计时，来实现 10 转每分
TIM2:   MOV TH0,#5DH
        MOV TL0,#3EH
        SETB TR0
        MOV R7,#0H
TIM3:   CJNE R7,#1H,TIM3
        DJNZ R6,TIM2
        LJMP OUT

QUICK:  MOV TH0,#5DH          ;定时器 0 启动;;快速,60 转/分
        MOV TL0,#3EH
        SETB TR0
        MOV R7,#0H
TIM1:   CJNE R7,#1H,TIM1
OUT:
RET
EINT0:
        MOV R7,#1
RETI
NUM:
        ;显示已转动的步数，每转动一次显示一个数
S0:     MOV A,R3              ;R3、R4、R5 分别表示 3 个数码管
        CALL EXP
        MOV A,R4
        CALL EXP
        ;mov r5,0ffh
        MOV A,R5
        CALL EXP

```

CJNE R3,#9,S1 ;若 R3 不等于 9, 则跳转到 S1 实现自增, 若等于 9, 则 R3 置 0, 重新开始循环。R4、R5 类似

```
MOV R3,#0
CJNE R4,#9,S2
MOV R4,#0
CJNE R5,#9,S3
MOV R5,#0
```

```
S1: INC R3
    LJMP STOP
```

```
S2: INC R4
    LJMP STOP
```

```
S3: INC R5
    LJMP STOP
```

```
STOP: ;停止返回程序
RET
```

```
EXP: ;显示程序
```

```
MOV 21H,R0
MOVC A,@A+DPTR ;寄存器间接寻址, DPTR
MOV R0,#8
```

```
CLY: CLR CLK ;P4.4 ;时钟线低电平
      RLC A ;累加器 A 的逻辑操作指令
      MOV DAT,C
      SETB CLK ;P4.4 ;时钟线高电平
```

```
DJNZ R0,CLY
MOV R0,21H
```

```
RET
```

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

六、实验思考题

1. 如采用单四拍工作模式, 每次步进角度是 15° , 程序要如何修改?

单四拍: 修改输出脉冲为 A-B~A~B-A 即可。

2. 如采用单双八拍工作模式, 每次步进角度是 7.5° , 程序要如何修改?

八拍： A-AB-B-BC-C-CD-D-DA-A

01 11 10 10 00 00 00 01

修改：定时器定时周期改变，故需修改定时初始值；以及相位值、循环次数也需改变。

3. 步进电机的转速取决于那些因素？有没有上、下限？

取决于脉冲频率。当然，由于各种物理因素（包括摩擦、机械惯性、响应时间等），步进电机的最高转速有限制。上限根据电机的差别而不同，下限则为 0。

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）工作寄存器组（00H——1FH）

内部 RAM 的 0-1FH 为四组工作寄存器区，每个区有 8 个工作寄存器（R0—R7）。在同一时刻，只能使用一组工作寄存器，这是通过程序状态字 PSW 的地 3，4 位来控制的。例如当此两位为 00 时，使用第 0 组工作寄存器，对应于 00H 到 07H 的内部 RAM 空间。也就是说，这时指令中使用 R0 与直接使用 00 单元是等价的，不过使用工作寄存器的指令简单，且执行快。

（2）可位寻址 RAM 区（20H——2FH）

内部 RAM 的 20H—2FH 为位寻址区域，这 16 个单元的每一位都对应一个位地址，占据位地址空间的 0—7FH，每一位都可以独立置位、清除、取反等操作。

（3）通用的 RAM 区（30H——7FH）

在中断和子程序调用中都需要堆栈。MCS—51 的堆栈理论上可以设置在内部 RAM 的任意区域，但由于 0—1FH 和 20—2FH 区域有上面说的特殊功能，因此一般设置在 30H 以后。

在内部 RAM 中，所有的单元都可以作为通用的数据存储器使用，存放输入的数据或计算的中间结果等，也可以作为条件转移的条件使用。

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，通过修改刷新频率即可实现。

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

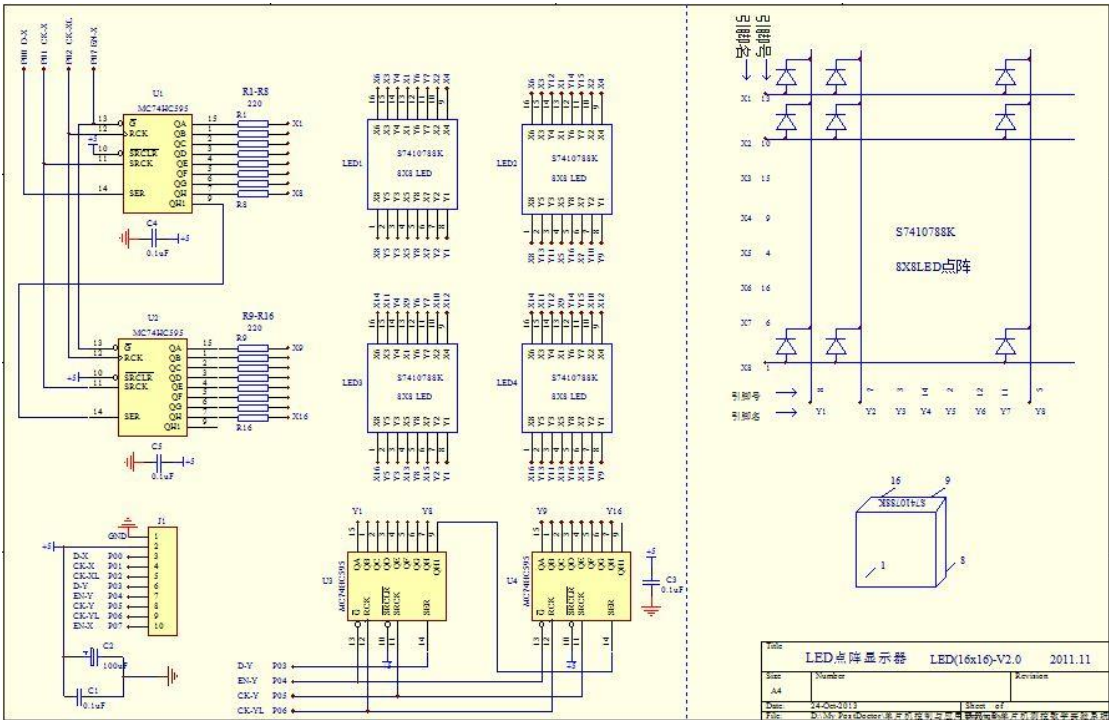
了解 16*16 点阵电路的原理，编写汇编语言程序。

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）。

四、实验原理

实验原理图如下：



实验的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。

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。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。

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。

先送出第 1 行控制发光管亮灭的数据并锁存，然后选通第 1 行使其点亮一段时间；然后送出第 2 行控制发光管亮灭的数据并锁存，然后选通第 2 行使其点亮相同时间.....第 16 行之后又重头点亮第 1 行，反复轮回，速度够快，由于人眼的暂留效果，就能看到屏上稳定的字。

每次循环显示完 16 行即一屏之后，再从第 1 行开始显示时，从上次开始的列数的下一列开始取值，比如上次从第 1 列开始显示，则循环完 16 行后，再从第 2 列开始显示，即可形成字符滚动显示的现象。

五、实验流程图与代码

实验代码：

```
ORG 0000H
```

```
LJMP START
```

```
ORG 0040H
```

```
START:
```

```
D_X EQU P0.0
```

```
D_Y EQU P0.3
```

```
CKX EQU P0.1
```

```
CKY EQU P0.5
```

```
CK_XL EQU P0.2
```

```
CK_YL EQU P0.6
```

```
EN_X EQU P0.7
```

```
EN_Y EQU P0.4
```

```
FORMOVE:
```

```
MOV R7,#0
```

SM:

MOV R0,#0

MOV R1,#1

MOV R4, #0

MOV R5, #1

MOV R3,#16

SM16:

SETB EN_X

CLR CK_XL

MOV DPTR,#TABLE1

MOV A,R0

MOVC A,@A+DPTR

MOV R6,#8

YW1:

CLR CKX

RLC A

MOV D_X,C

SETB CKX

DJNZ R6,YW1

MOV A,R1

MOVC A,@A+DPTR

MOV R6,#8

YW0:

CLR CKX

RLC A

MOV D_X,C

SETB CKX

```
DJNZ R6,YW0  
SETB CK_XL  
CLR EN_X  
SETB EN_Y  
CLR CK_YL  
MOV DPTR,#TABLE  
MOV A,R4  
ADD A,R7  
MOVC A,@A+DPTR  
MOV R6,#8
```

```
YW3:  
CLR CKY  
RLC A  
CPL C  
MOV D_Y,C  
SETB CKY  
DJNZ R6,YW3  
MOV A,R5  
ADD A,R7  
MOVC A,@A+DPTR  
MOV R6,#8
```

```
YW2:  
CLR CKY  
RLC A  
CPL C  
MOV D_Y,C  
SETB CKY  
DJNZ R6,YW2
```

```
SETB CK_YL
CLR EN_Y
LCALL DELAY
INC R0
INC R0
INC R1
INC R1
INC R4
INC R4
INC R5
INC R5
DJNZ R3,SM16
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
INC R7
INC R7
MOV A,R7
SUBB A,#96
JZ FORMOVE
LJMP SM
```

```
DELAY:
MOV R6,#0FFH
LOOPD1:
MOV R2,#0FH
LOOPD2:
```

DJNZ R2,LOOPD2

DJNZ R6,LOOPD1

RET

TABLE:

DB 00H,00H,04H,01H,04H,01H,04H,02H,04H,04H,04H,08H,04H,30H,04H,0C0H ;大

DB 0FFH,00H,04H,0C0H,04H,30H,04H,08H,04H,04H,04H,02H,04H,01H,04H,01H

DB 20H,00H,20H,04H,20H,04H,20H,04H,20H,04H,20H,04H,20H,04H,3FH,0FCH ;工

DB 20H,04H,20H,04H,20H,04H,20H,04H,20H,04H,20H,04H,20H,00H,00H,00H

DB 08H,24H,06H,24H,01H,0A4H,0FFH,0FEH,01H,23H,06H,22H,40H,00H,49H,3EH ;程

DB 49H,22H,49H,22H,7FH,22H,49H,22H,49H,22H,49H,3EH,41H,00H,00H,00H

;DB 00H,00H,00H,00H,0FH,0F0H,08H,08H,08H,10H,0FFH,0FFH,08H,00H,08H,00H

;DB 0FH,0F8H,00H,00H,00H,00H,0FFH,0F0H,00H,0CH,00H,02H,3FH,0E1H,00H,00H

;DB 00H,00H,00H,1EH,10H,02H,08H,02H,04H,02H,02H,02H,01H,02H,0FFH,0FCH

;DB 00H,00H,02H,08H,02H,08H,02H,04H,02H,04H,7FH,0FEH,00H,00H,00H,00H

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

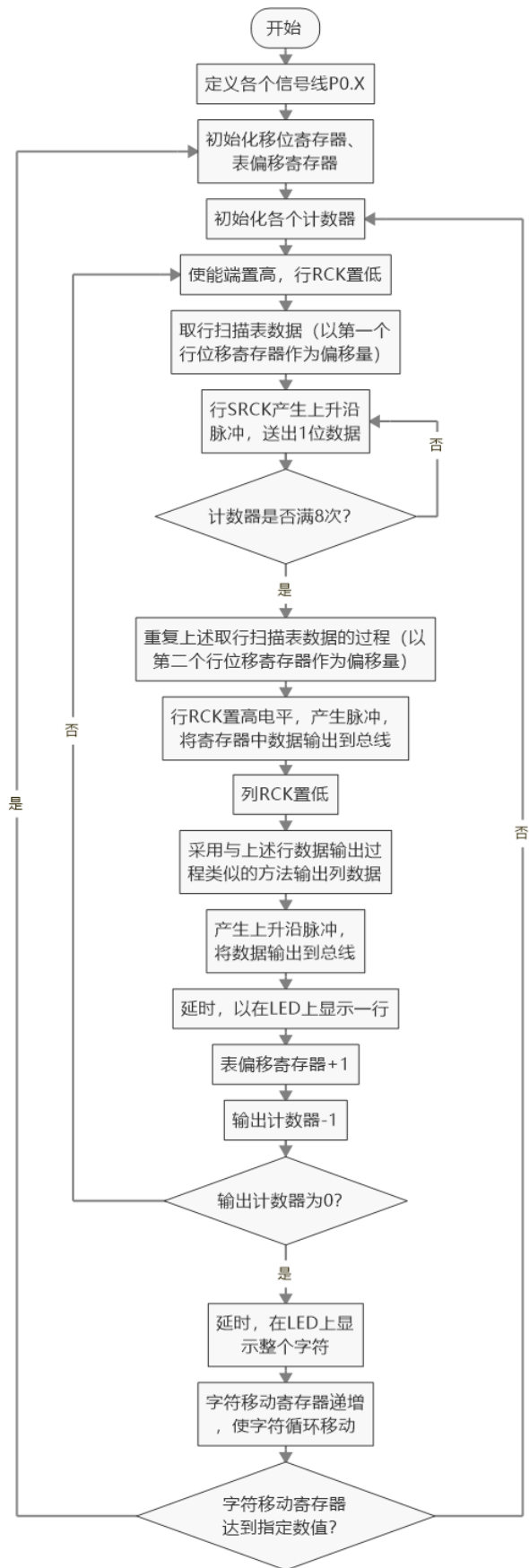
DB 00H,02H

DB 00H,01H

END

实验流程图：

（版面原因见下页）



(版面原因见下页)

六、实验思考题

1. 如何使用软件调整和控制 LED 点阵的亮度?

对于 LED 显示屏，点的亮度值对应于有效的显示时间，这是由列的显示时间来实现的，时间越长，亮度越高。

2. 如何尽量避免显示过程中的闪烁?

对于用户来说，显示频率越高，屏的视觉稳定性越强；显示频率越低，屏的视觉稳定性越差，即显示屏看起来会有闪烁和抖动的现象。因此为避免显示过程中的闪烁，应提高显示频率。

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）?

可以多增加几个控制行列的移位寄存器。