

# 单片机控制与应用实验报告



班级：8班

姓名：杨钊

学号：21160834

实验一 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、实验步骤

- 1 预习
- 2 简单程序录入和调试
- 3 程序调试
- 4 编写程序，完成功能

## 五、实验原理

### 1、旋转方向

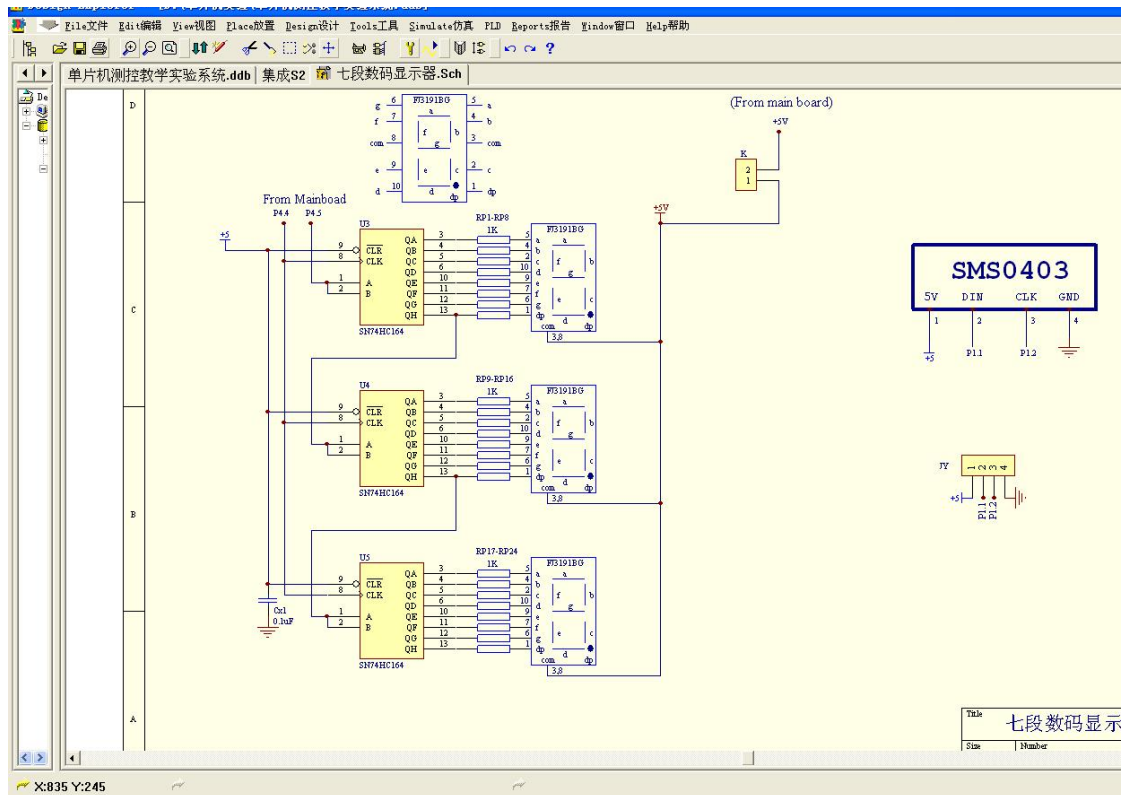
本实验使用简单的双四拍工作模式即可，这也是 FAN8200 比较方便的工作方式。只要将 CE1 和 CE2 分别置为高，然后 IN1 和 IN2 按照预定的脉冲输出，即 01->11->10->00->01 这个循环构成一个方向旋转的输出脉冲，将此序列翻转，就是相反方向的输出脉冲。

### 2、旋转速度

只需要给计时器设置不同的计数初值即可。

### 3、数码管显示

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。

## 何时输入？

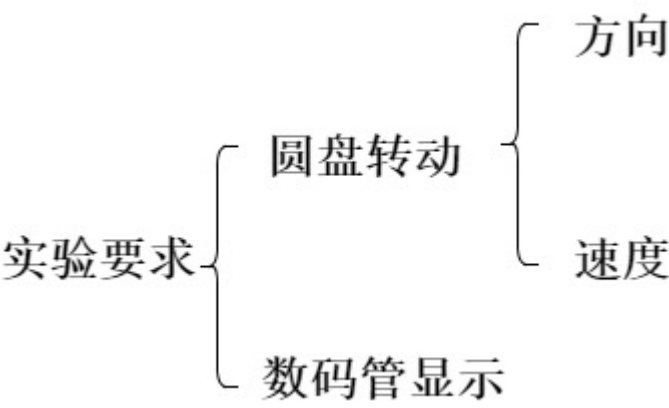
(本次实验中, 由于移位寄存器的数据线和时钟线分别接到单片机的 P4.5 和 P4.4 管脚, 因此, 当 P4.4 从 0 变 1 时, 即上升沿时, 数据输入至 Q0。)

## 输入什么？

(由于输入信号是 A 和 B 的逻辑与, 且 A 和 B 均连接的是上一个 74HC164 的 Q7 (第一个连接的是 p4.5), 因此, 若上一个 74HC164 的 Q7 是 0, 则输入 0; 若是 1, 则输入 1)

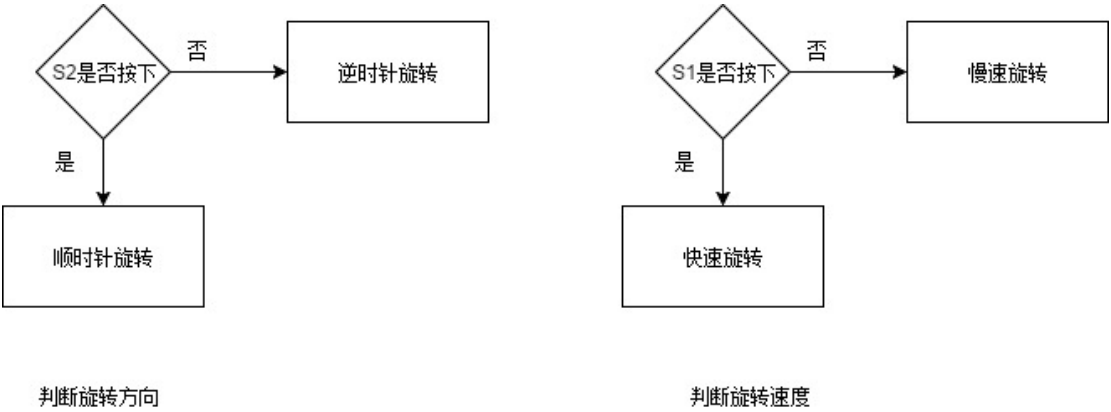
六、代码实现过程

实验要求分析如下：

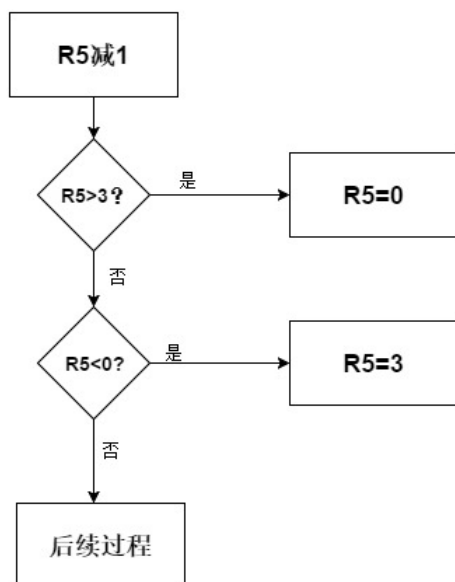


【功能 1、圆盘转动】

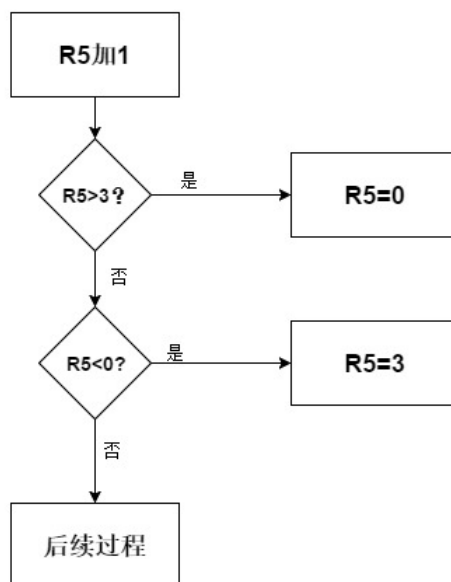
本次实验用计时器 T0 控制圆盘的转动。中断发生时需要扫描两个按键，分别用来控制旋转方向和旋转速度，流程图如下所示：



顺时针旋转和逆时针旋转实现过程流程图如下：

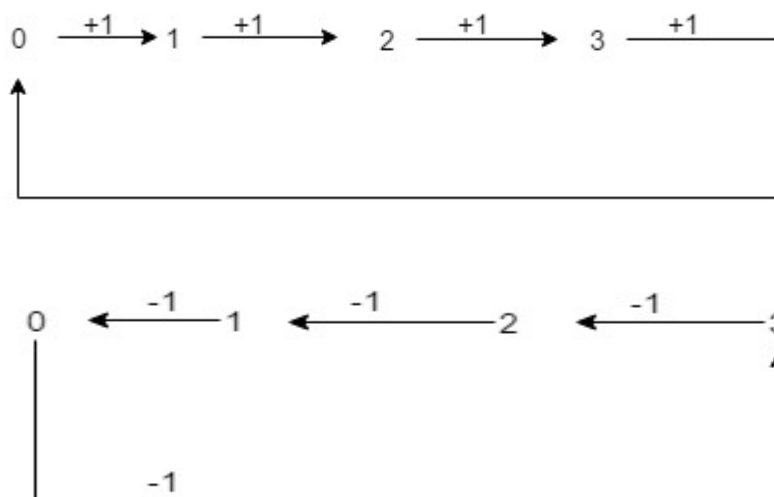


顺时针旋转



逆时针旋转

(注：顺时针和逆时针旋转只有第一步对 R5 的操作不同，如果是从前往后的顺序 00H→ 01H→ 03H→ 02H，则是逆时针，顺时针则正好相反。可见，R5 在这里是起到一个下标的作用，注意下标的合法范围是 0--3，因此如果下标范围不合法要进行相应的调整。具体调整方法为：



至此，我们可以给出顺时针旋转和逆时针旋转的全过程。

1、将 00H,01H,03H,02H（注意顺序）依次存放在 0040H 为起始地址的存储空间。（注意这些数据的低两位对应了控制旋转方向的脉冲序列）

ORG 0040H

DB 00H, 01H, 03H, 02H

2、以 R5 为下标，取出存储空间不同的数据。(如 R5 为 0 时取出第一个数据 00H，R5 为 3 时取出第四个数据 02H，可见 R5 增大时是从前往后取数据，对应逆时针)。然后按照 S2 是否按下，对 R5 进行相应的加或减操作。(如果 S2 没有按下，R5 加 1，取数的顺序是从前往后 00H→01H→03H→02H，对应逆时针，否则 R5 减 1，取数顺序是 00H←01H←03H←02H，对应顺时针。可见，旋转方向就是通过对 R5 执行加减操作控制的)

MOV A, R5  
MOV DPTR, #0040H  
MOVC A, @A+DPTR

3、将 CE1 和 CE2 分别置为高，然后 IN1 和 IN2 按照预定的脉冲输出，即将第 2 步取出的数据**低两位**送给 IN1 和 IN2。每次中断服务程序中都不断循环以上三步，即可完成指定方向的旋转。

```
CLR P1.1          ;CE1, CE2 至置高，才能按照指定脉冲显示
CLR P1.4
RRC A             ;右移
MOV P1.0, C       ;第八位给 P1.0 :IN2
RRC A
MOV P3.2, C       ;第七位给 P3.2 :IN1
SETB P1.1         ;CE1, CE2 置高，才能按照指定脉冲显示
SETB P1.4
```

以上是对旋转方向的说明。

旋转速度通过给计时器设置不同的计数初值即可实现，不再赘述。

## 【功能 2、数码管显示】

1、存储段码表。

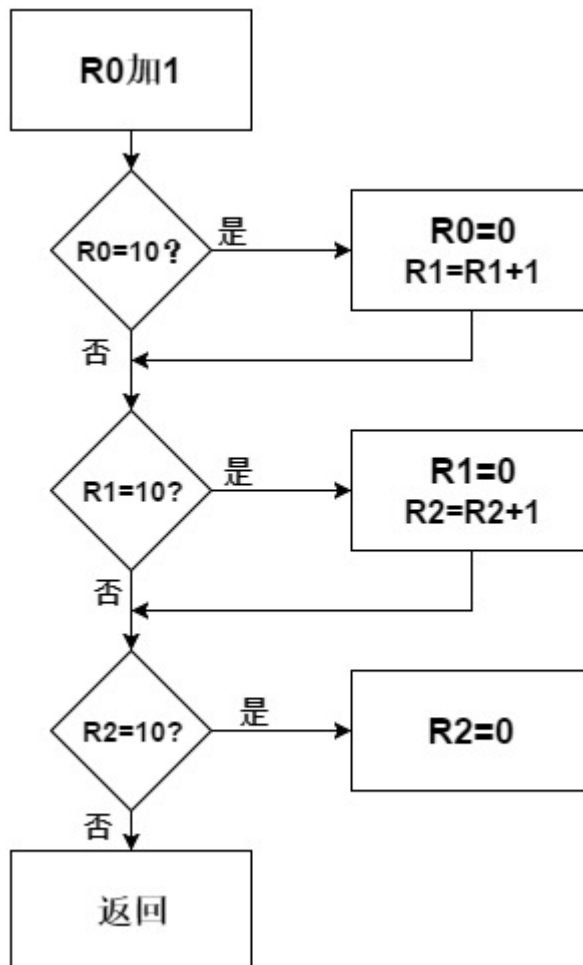
TABLE:

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

2、每次步进电机前进一步（不论顺时针还是逆时针），计数值均加 1。

（R0、R1、R2 分别对应了个位、十位、百位）

（注意进位问题，例如 R0 原来值为 9，加一之后变成 10，应该 R0 置 0，R1 加 1）



3、显示数字

因为要想输出一个字形码，就需要从高位到低位（因此是循环左移）依次向移位寄存器输出 8 个比特。因此需要对 R0、R1、R2 分别向移位寄存器输出，在 24 个 bit 之后，欲显示的字形将稳定地显示在数码管上。

DISPLAY:

MOV A,R0

MOVC A,@A+DPTR ;DPTR 是 table

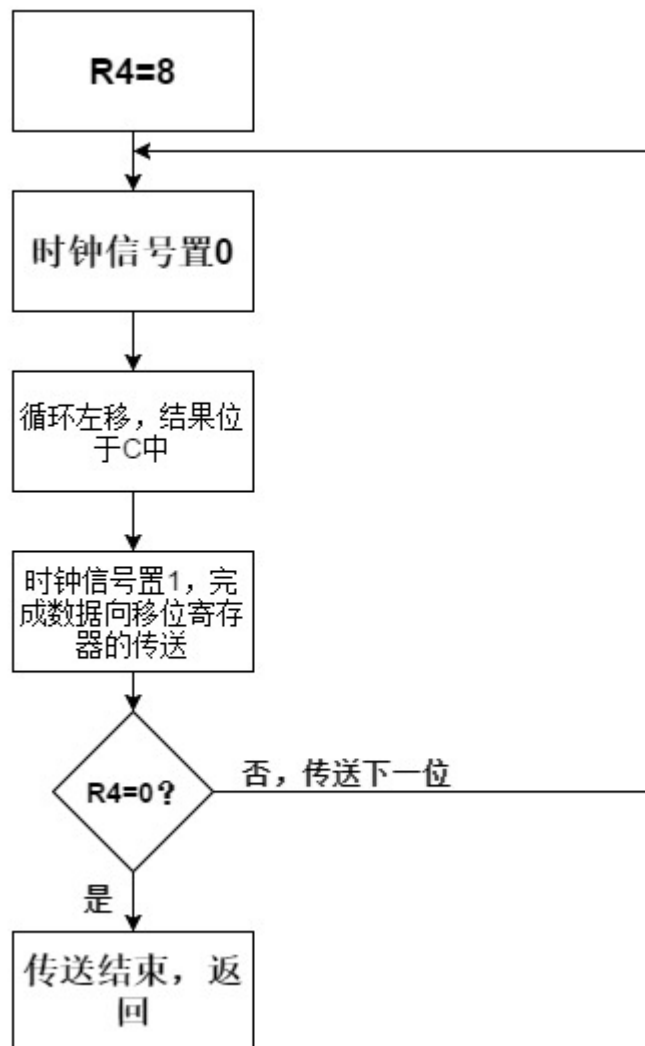
LCALL SENDNUM ; 对 R0 调用 SENDNUM

MOV A,R1  
MOVC A,@A+DPTR  
LCALL SENDNUM ; 对 R0 调用 SENDNUM

MOV A,R2  
MOVC A,@A+DPTR  
LCALL SENDNUM ; 对 R0 调用 SENDNUM

RET

SENDNUM 流程图如下：



当对 R0、R1、R2 均完成数据传送后（ $8 \times 3 = 24$  位），计数值将稳定地显示在数码管上，程序可以转而执行其他工作。



## 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  $01 \rightarrow 11 \rightarrow 10 \rightarrow 00 \rightarrow 01$  改为： $00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$

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。

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？

答：(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。

答: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?

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。

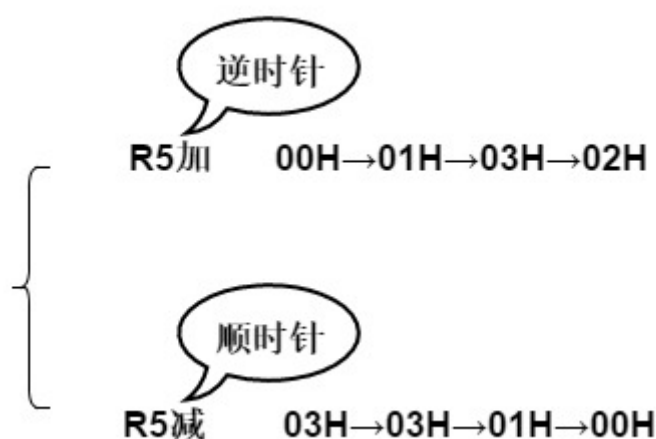
### 9. 在本实验环境下,能否控制显示数码的亮度?如何实现?

答:能,通过修改刷新频率

## 八、老师布置思考题

比较已有的两份代码之间差异,解释为什么代码 1 能使圆盘旋转一周而代码 2 只能使圆盘旋转一部分?

答:代码 1 的实现过程在上面【六、代码实现过程】中已有详细解释。代码 1 利用 R5 完成旋转操作。



代码一实现如下:

CLR P1.1 ;CE1, CE2 至置高, 才能按照指定脉

冲显示

CLR P1.4

RRC A ;右移

MOV P1.0, C ;第八位给 P1.0 :IN2

RRC A

MOV P3.2, C ;第七位给 P3.2 :IN1

SETB P1.1 ;CE1, CE2 至置高, 才能按照指定脉

冲显示

SETB P1.4

需要注意的是, 程序并不是将 CE1、CE2 置高就能顺利完成对 IN1、IN2 的数据传送, 而是每次都先置 0、再置 1, 在上升沿完成对数据的传送! 从这个角度考虑, 之所以代码二不能旋转完整的一周, 应该是时序有问题。经检查代码二发现: 代码二只在实验起始位置对 CE1、CE2 置 1, 接下来的代码里没有涉及对 CE1、CE2 的操作, 由于不是处于 CE1、CE2 上升沿, 因此代码二在对 IN1、IN2 进行数据传送时出错。

修改:

每次数据传送都加上如下四行代码即可:

CLR CE1

CLR CE2 ; CE1、CE2 置 0

SETB IN1

SETB IN2

MOV R5,#1

MOV R6,#1

SETB CE1

SETB CE2 ; CE1、CE2置1

## 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

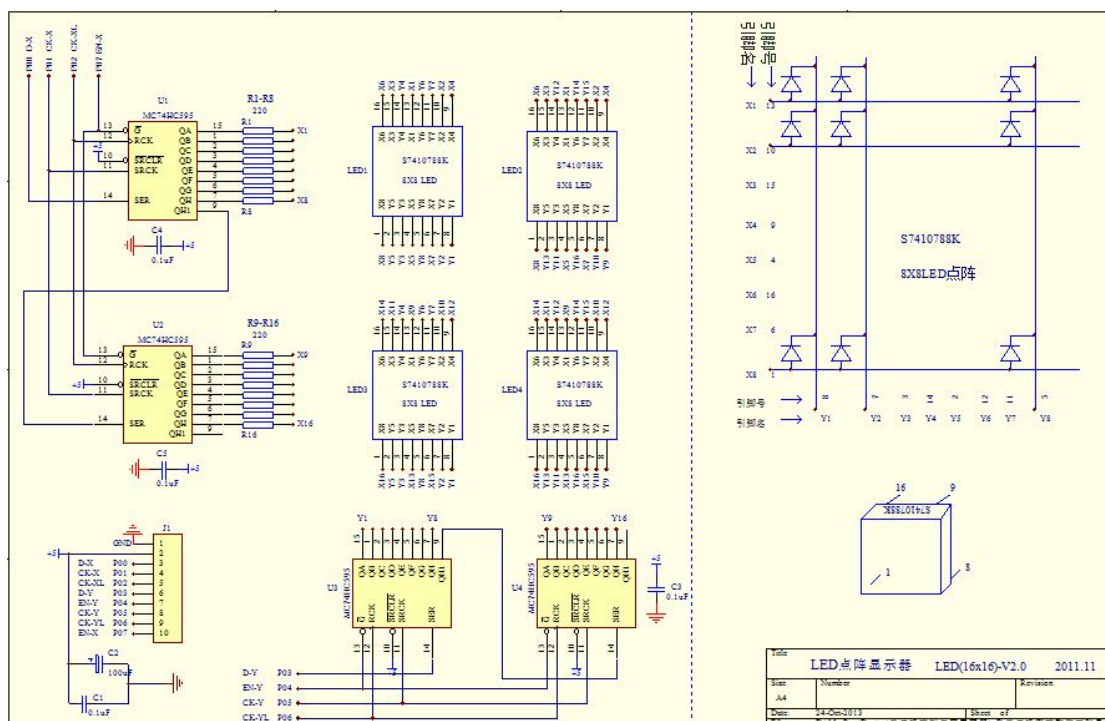
了解 16\*16 点阵电路的原理，编写汇编语言程序。

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）。

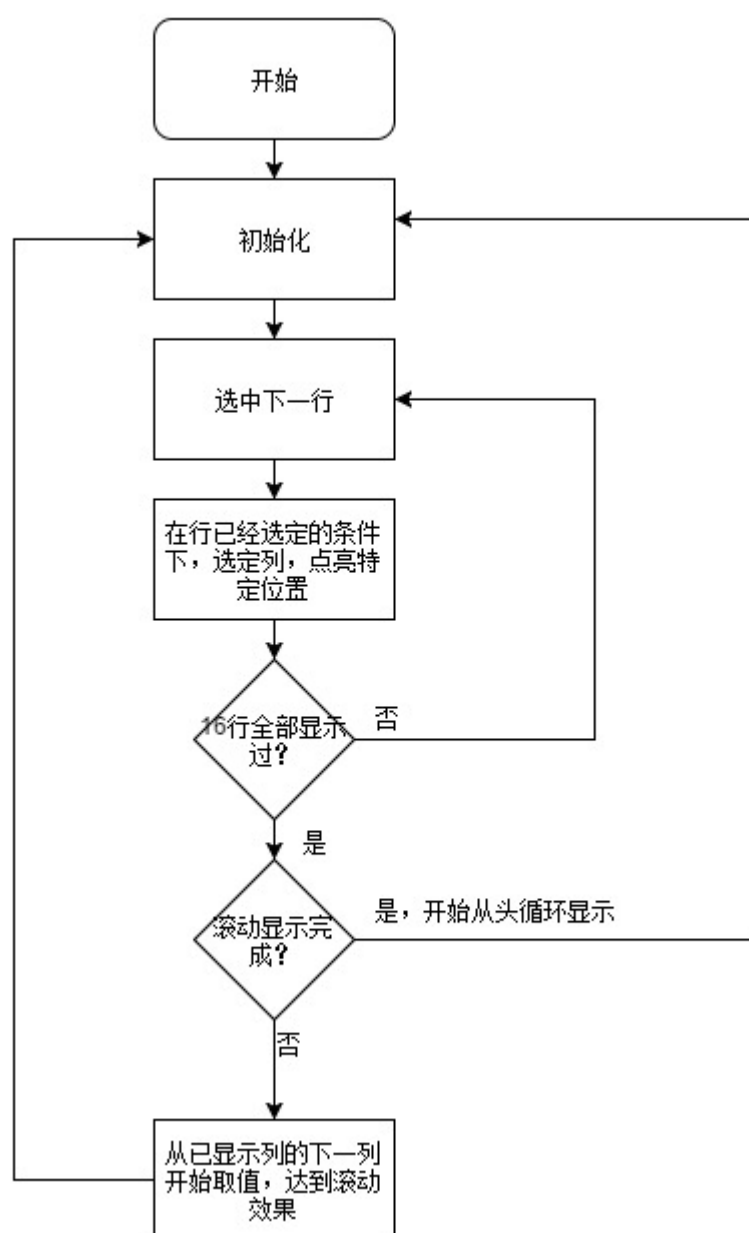
### 四、实验原理

本实验原理图如下：

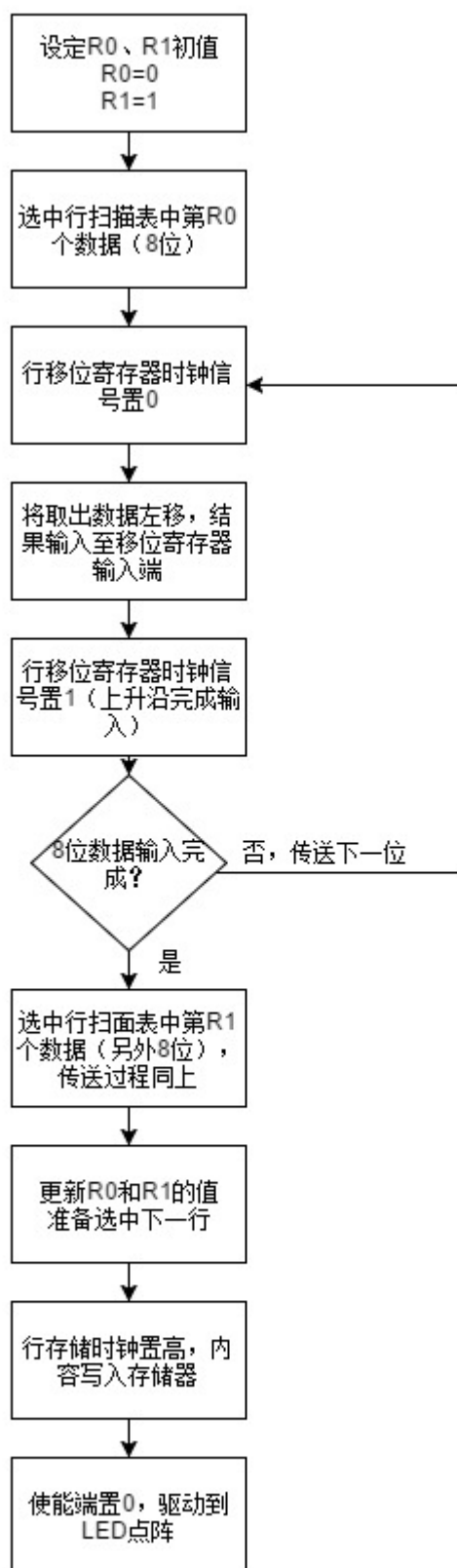


从上次开始的列数的下一列开始取值，比如上次从第 1 列开始显示，则循环完 16 行后，再从第 2 列开始显示……这样就能形成字符滚动显示的现象。

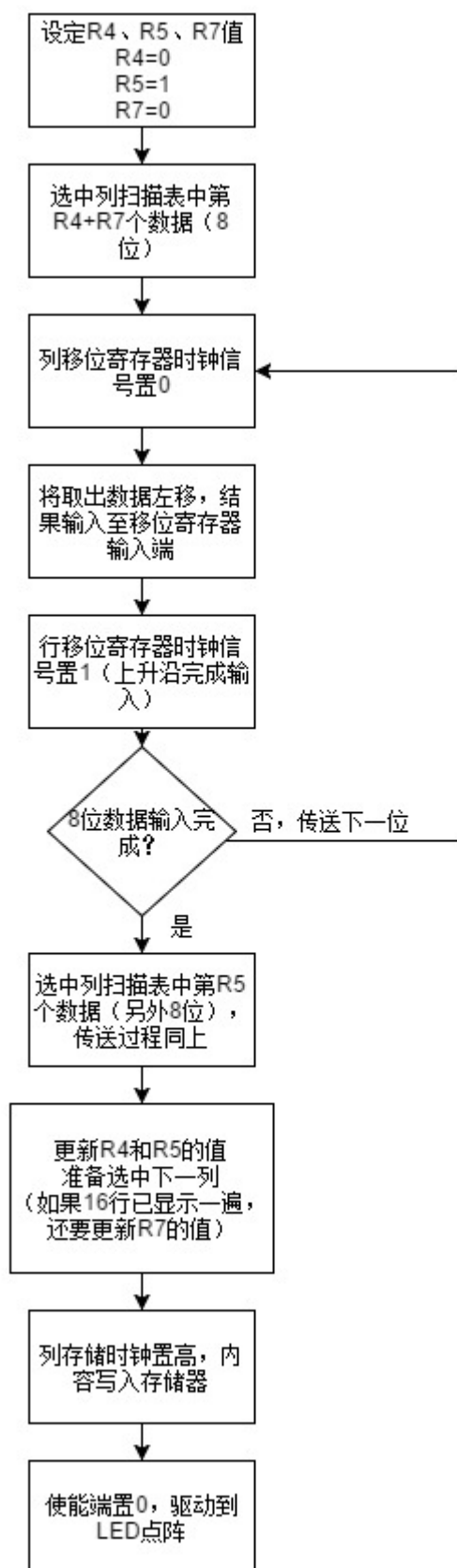
## 五、实验流程图



其中，选中某行的流程图如下：



列选中流程图如下：



## 六、思考题



1. 如何使用软件调整和控制 LED 点阵的亮度

答：可以通过控制行显示延时调整亮度。延时越短，扫描频率越快，LED 点阵越亮。

2. 如何尽量避免显示过程中的闪烁

答：增加每一屏显示次数，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）

答：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 0ffffH 改为 64 位的 0ffffffffffffeH 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。