
学号：53160831 姓名：张志帆

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 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

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、实验原理

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。

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,

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

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。

七段字形的编码方式需要通过实验获得。这些编码作为程序中的常数, 使用 DB 命令存放。在程序中, 需要将数值转换为相应的字形编码, 可以使用 MOVC 指令来完成

三、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

四、实验内容

1 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

2 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

3 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

五、实验步骤

1 预习

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

2 简单程序录入和调试

MCS51 单片机汇编语言的基本格式比较简单，程序中可以使用通用寄存器或者内存单元进行计算。另外，单片机的程序没有退出到操作系统的概念，一般都是死循环程序。

MCS51 单片机汇编语言的基本格式比较简单，程序中可以使用通用寄存器或者内存单元进行计算。另外，单片机的程序没有退出到操作系统的概念，一般都是死循环程序。

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

$$\text{定时周期} = 12/\text{CPU 晶振频率}$$

得到的 s 需要分成高 8 位和低 8 位，分别放入计数器 TH $_x$ 和 TL $_x$ 中 (x 为 0 或 1)。如果 s 为负数，说明需要的定时时间太长，即使定时器的最大时间也无法满足要求。这种情况下，需要加入软件循环才能实现。我们可以将需要的定时时间分成 n 份，利用定时器达到 t/n 的时间长度，然后在定时器处理程序中，累计某一变量，如果到达 n ，说明总的时间 t 已经达到。

要想使用定时器中断，除了上面的定时器初值设定外，还需要将其他相关的特殊功能寄存器也都设置好。如果使用方式 0 和方式 1，不要忘记在计数结束后重新恢复计数器初值。

3 程序调试

用单步、断点、连续方式调试程序，观察状态指示灯及电机状态，检查运行结果。如果需要，可以将四个输出信号的状态同时输出到 P0 口的某些位上，便于观察

4 编写程序，完成功能

六、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 $A = \text{in1}$ $B = \text{in2}$, (!A)表示 $\text{in1}=0$, (!B)表示 $\text{in2}=0$

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?

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)

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?

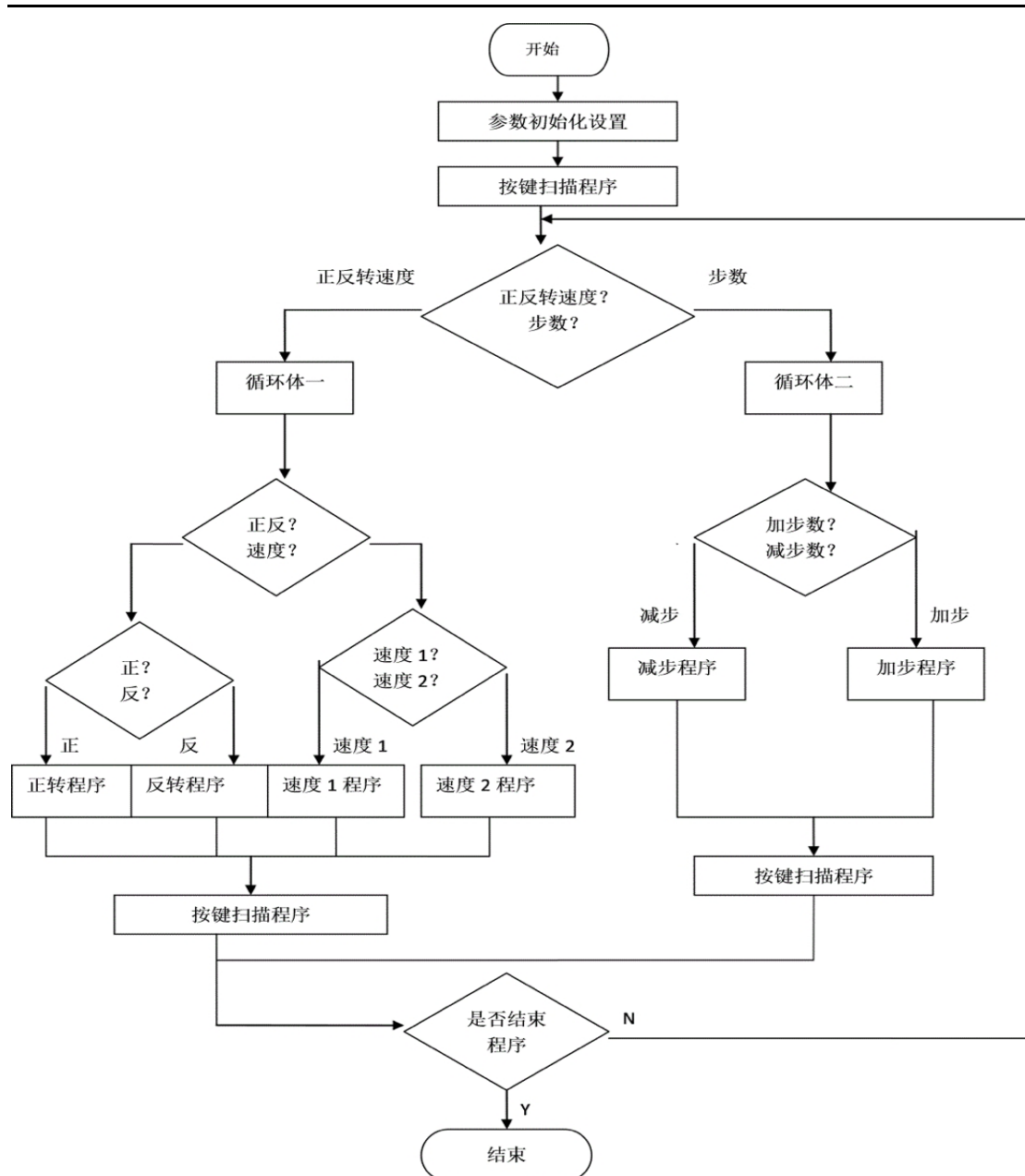
答: 一个机器周期包含 6 个状态 (S1-S4), 每个状态分为两个节拍 P1 和 P2, 通常, 一个机器周期会出现两次高电平 S1P2 和 S4P2, 每次持续一个状态 S。乘法及除法指令占 4 个周期, 三字节指令均为双周期指令。

9, 在本实验环境下, 能否控制显示数码的亮度? 如何实现?

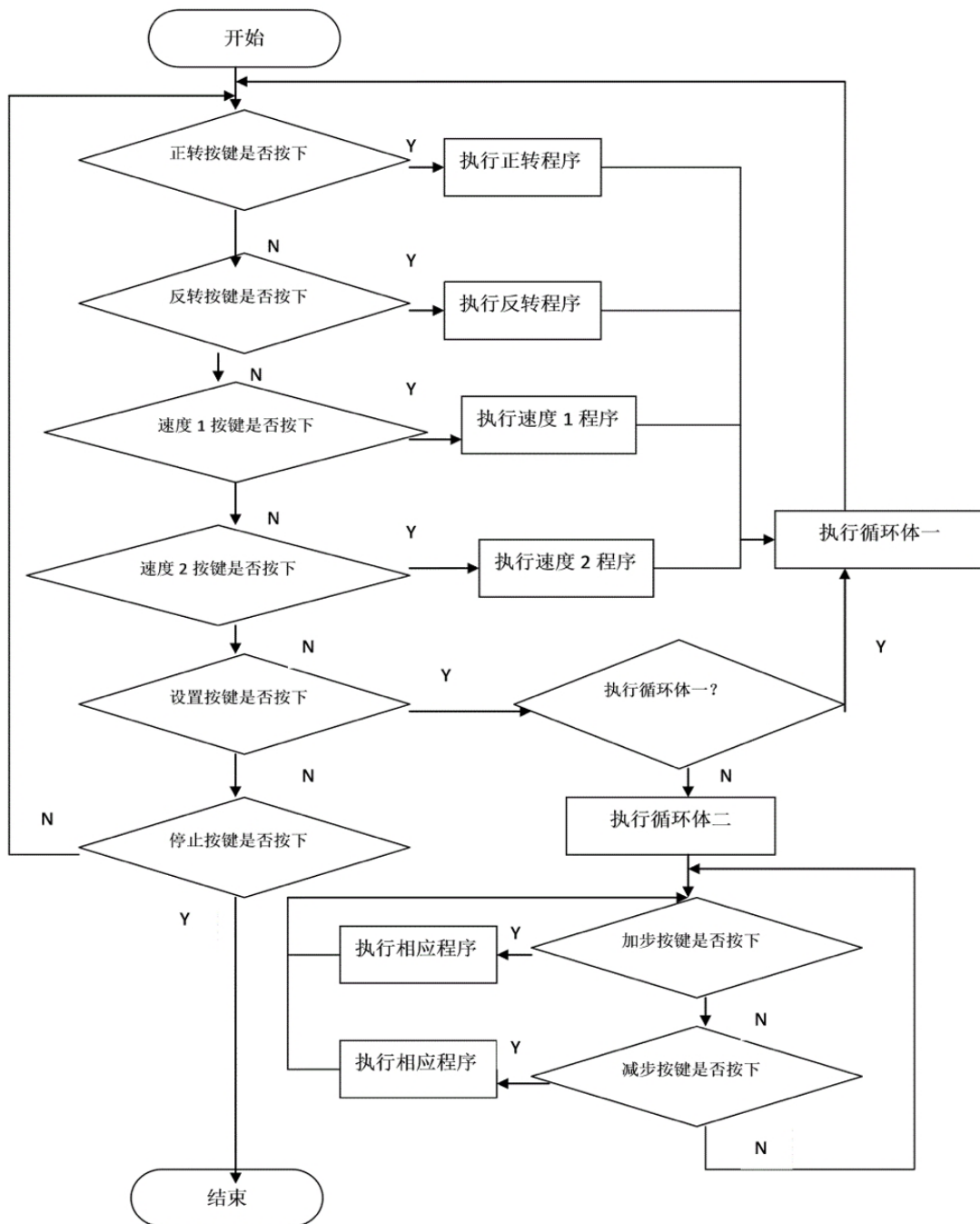
答: 能, 通过修改刷新频率

七流程图

主流程图



按键扫描程序流程图



八、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 TL0,#LOW(65536-41667)

MOV TH0,#HIGH(65536-41667)

SETB TR0

SETB ET0

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 TL0,#LOW(65536-41667)
    MOV TH0,#HIGH(65536-41667)
    RETI

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

```

实验四 LED 点阵显示屏

一、实验目的和要求

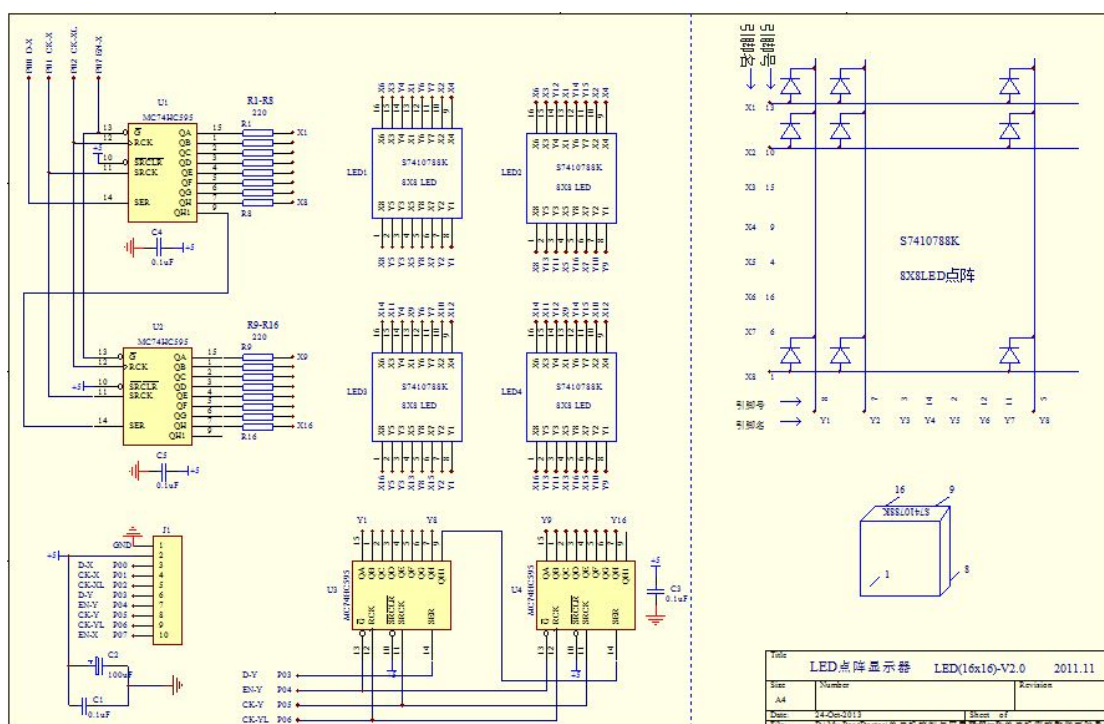
了解 LED 点阵显示的基本原理和实现方法。掌握点阵汉字库的编码和从标准字库中提取汉字编码的方法

二、实验原理

高亮度 LED 发光管构成点阵，通过编程控制可以显示中英文字符、图形及视频动态图

形。所显示字符的点阵数据可以自行编写（即直接点阵画图），也可从标准字库（如 ASC16、HZ16）中提取。后者需要正确掌握字库的编码方法和字符定位的计算

本实验示意电路原理图



实验用的 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。

其他信息见给定的参考资料。

三、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

四、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）

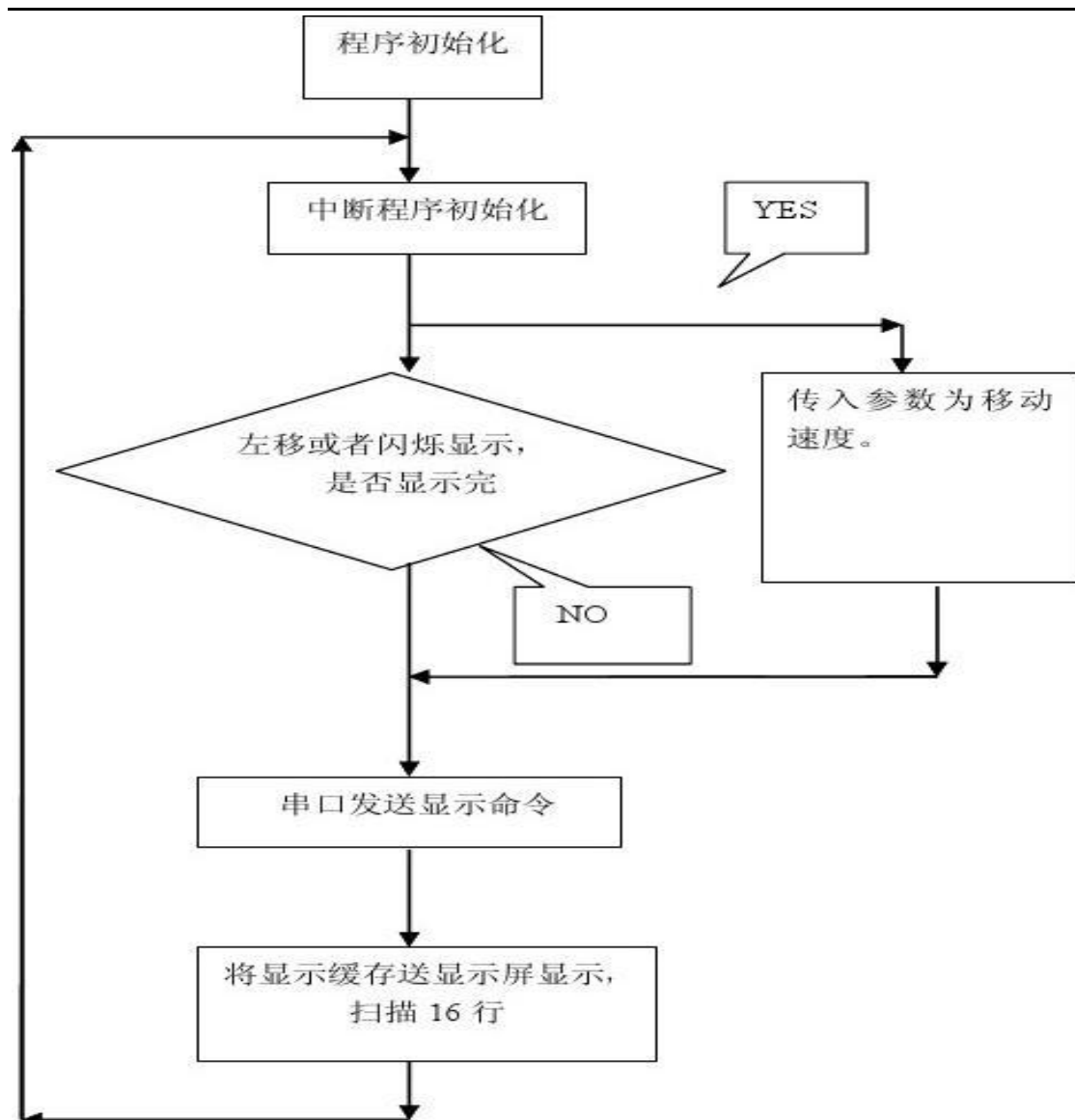
五、实验步骤

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。

六、思考题

- 1、如何使用软件调整和控制 LED 点阵的亮度
使用控制软件里的亮度调节功能
2. 如何尽量避免显示过程中的闪烁
加大扫描频率，减小闪烁周期
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）

七、流程图



八、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
```

两次实验的体会与收获

通过这两次实验，使我对 STC-ISP 、Keil uvision2 等软件的认识有了进一步的加深，初步学习和掌握 MCS-51 的体系结构和汇编语言，对它们的使用能力也得到了进一步的加强。同时也加深对单片机课程的全面掌握和认识，对单片机课程的应用得到了更深的了解。了解了步进电机的工作原理，学习了用单片机的步进电机控制系统的硬件设计方法，掌握了定时器和中断系统的应用，熟悉了单片机应用系统的设计与调试方法

虽然设计中只使用了一块 16×16LED 点阵，电路简单，但是已经包涵了 LED 汉字滚动显示屏的电路基本原理、基本程序和 Proteus 软件仿真，只要扩展单片机的 10 接口，并增加一些 LED 点阵和相关芯片，就能设计出更大面积、更多花样的 LED 显示屏。

在一开始做课程设计的时候由于缺乏经验和资料，没有找准目标，思路和设计都缺乏针对性，在一些小细节上浪费了很多时间。通过和同学的交流，找准了此次课程设计的重点，在网上有目的的找到了相关的资料，在同学的帮助和自己的努力下完成了最后的课程设计。虽然还有很多地方需要完善，但通过本次课程设计使我对自己的动手能力更加的有自信，同时也了解到人多力量大的道理，和同学的交流以及上网搜索资料能更快更好的完成任务