## **实验四 LED点阵显示屏**

21200903

1. **原理总结**

1、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。

2、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。

3、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。

4、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 Dx（P00）、列 Dy(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。

5、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。

1. **程序分析**
2. 代码

org 0000h

jmp Start

org 0040h

Start:

P4 EQU 0C0H

P4SW EQU 0BBH

CE1 EQU P1.1

CE2 EQU P1.4

IN1 EQU P3.2

IN2 EQU P1.0

S1 EQU P3.6

S2 EQU P3.7

DATA\_X EQU P0.0

CLK\_X EQU P0.1

WR\_X EQU P0.2

EN\_X EQU P0.7

DATA\_Y EQU P0.3

CLK\_Y EQU P0.5

WR\_Y EQU P0.6

EN\_Y EQU P0.4 ; initial

CLR EN\_X ;set 0

CLR EN\_Y

MOV R7,#0

NEXT: ;0-1 2-3 4-5 ....

MOV DPTR,#TAB\_X

MOV A,R7

MOVC A,@A+DPTR

MOV R0,A

INC R7

MOV A,R7

MOVC A,@A+DPTR

MOV R1,A

DEC R7

MOV DPTR,#TAB\_Y

MOV A,R7

MOVC A,@A+DPTR

MOV R2,A

INC R7

MOV A,R7

MOVC A,@A+DPTR

MOV R3,A

call display

call delay

INC R7

CJNE R7,#20H,NEXT

JB S1,START ;false, go back to show again

MOV R4,#00H ; zuo shang jiao qi dian

MOV R5,#00H

;CALL display

INC R4

STEP2:

CALL display\_shape

JB S1,KN0

MOV R1,#02H

DLL0: MOV R0,#0FFH

DL0: CALL delay

DJNZ R0,DL0

DJNZ R1,DLL0

INC R5

KN0:

JB S2,KN1

MOV R1,#02H

DLL1: MOV R0,#0FFH

DL1:CALL delay

DJNZ R0,DL1

DJNZ R1,DLL1

INC R4

KN1:

JMP STEP2

JMP START

display: ;show name

PUSH ACC

SETB EN\_X

SETB EN\_Y

CLR WR\_X

CLR WR\_Y

MOV R6,#8

MOV A,R0

D0:

RLC A

CLR CLK\_X

MOV DATA\_X,C

SETB CLK\_X

DJNZ R6,D0

MOV R6,#8

MOV A,R1

D1:

RLC A ;16\*16

CLR CLK\_X

MOV DATA\_X,C

SETB CLK\_X

DJNZ R6,D1

MOV R6,#8

MOV A,R2

D2:

RLC A

CLR CLK\_Y

MOV DATA\_Y,C

SETB CLK\_Y

DJNZ R6,D2

MOV R6,#8

MOV A,R3

D3:

RLC A

CLR CLK\_Y

MOV DATA\_Y,C

SETB CLK\_Y

DJNZ R6,D3

SETB WR\_X ;start

CLR EN\_X

SETB WR\_Y

CLR EN\_Y

POP ACC

RET

display\_shape:

MOV R0,#00H

MOV R1,#01H

MOV R2,#80H

MOV R3,#0FFH

MOV A,R4

JZ DS1

MOV R7,A

DD0:

MOV A,R1

RLC A

MOV R1,A

MOV A,R0

RLC A

MOV R0,A

DJNZ R7,DD0

DS1: MOV A,R5

JZ DS2

MOV R7,A

DD1:

MOV A,R3

RRC A

MOV R3,A

MOV A,R2

RRC A

MOV R2,A

DJNZ R7,DD1

DS2:

call display

call delay

CLR CY

MOV A,R1

RLC A

MOV R1,A

MOV A,R0

RLC A

MOV R0,A

call display

call delay

CLR CY

MOV A,R1

RLC A

MOV R1,A

MOV A,R0

RLC A

MOV R0,A

call display

call delay

CLR CY

MOV A,R1

RLC A

MOV R1,A

MOV A,R0

RLC A

MOV R0,A

call display

call delay

CLR CY

MOV A,R1

RLC A

MOV R1,A

MOV A,R0

RLC A

MOV R0,A

call display

call delay

CLR CY

MOV A,R1

RLC A

MOV R1,A

MOV A,R0

RLC A

MOV R0,A

call display

call delay

RET

delay:

MOV R6,#0FFH

LOOP0:DJNZ R6,LOOP0

RET

TAB\_X:

DB 00H,01H,00H,02H,00H,04H,00H,08H,00H,10H,00H,20H,00H,40H,00H,80H;

DB 01H,00H,02H,00H,04H,00H,08H,00H,10H,00H,20H,00H,40H,00H,80H,00H;

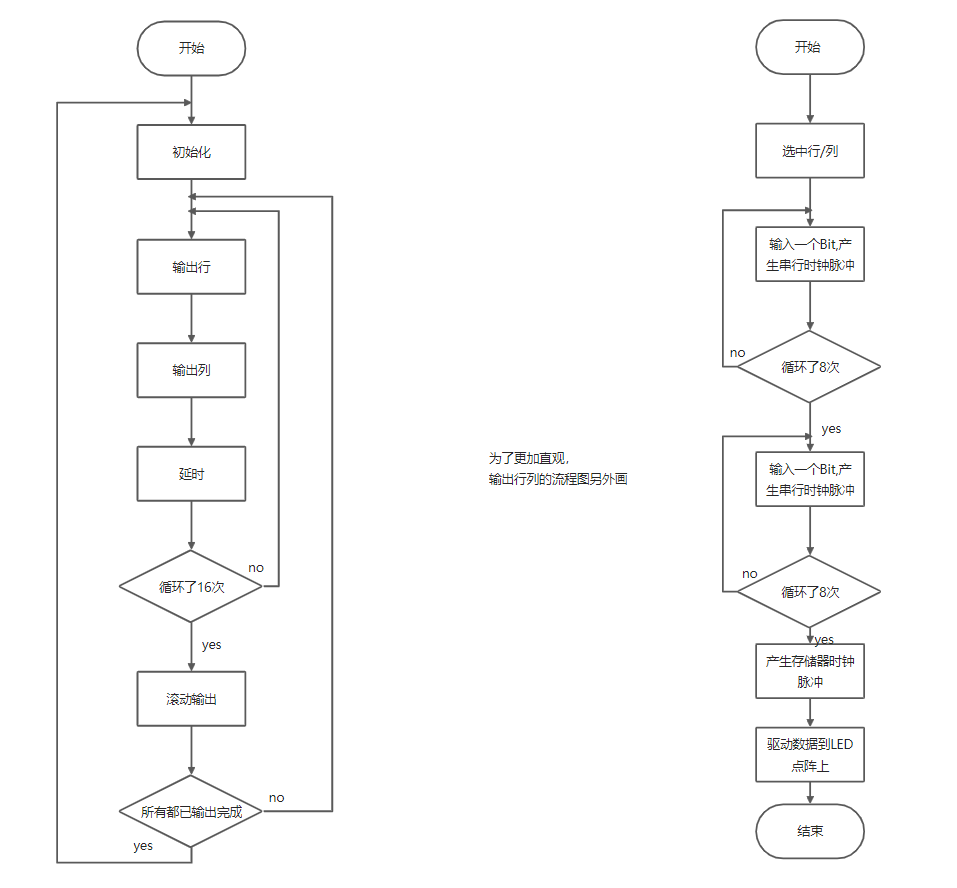
TAB\_Y:

DB 0FFH,0FFH,0FFH,0DFH,0FFH,0DDH,0E0H,1DH,0EEH,0DBH,0EEH,0D3H,80H,1FH,0EEH,0DFH;

DB 0EEH,0DFH,80H,1FH,0EEH,0D7H,0EEH,0D3H,0E0H,1BH,0FFH,0D9H,0FFH,0DFH,0FFH,0FFH;"?",0

;====================================================================

END



1. **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. 如何提高显示图案的对比度，避免模糊？

LED点阵是动态扫描，都是有一个扫描频率，若低于一定的扫描频率，显示会闪烁，模糊，对比度低，不稳定。故可以提高扫描频率。当然，频率也不能过高，这样亮度较差且占用cpu资源。

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。

1. **实验过程中遇到的问题**

实验过程中，对于输出字符的字符串中出现编译错误，当有语法错误时，需在前面添加零，来使得能够正常编译运行。在控制方块移动的过程中，需设定正方形边长，通过计算与尝试，能够完成方块的设定。