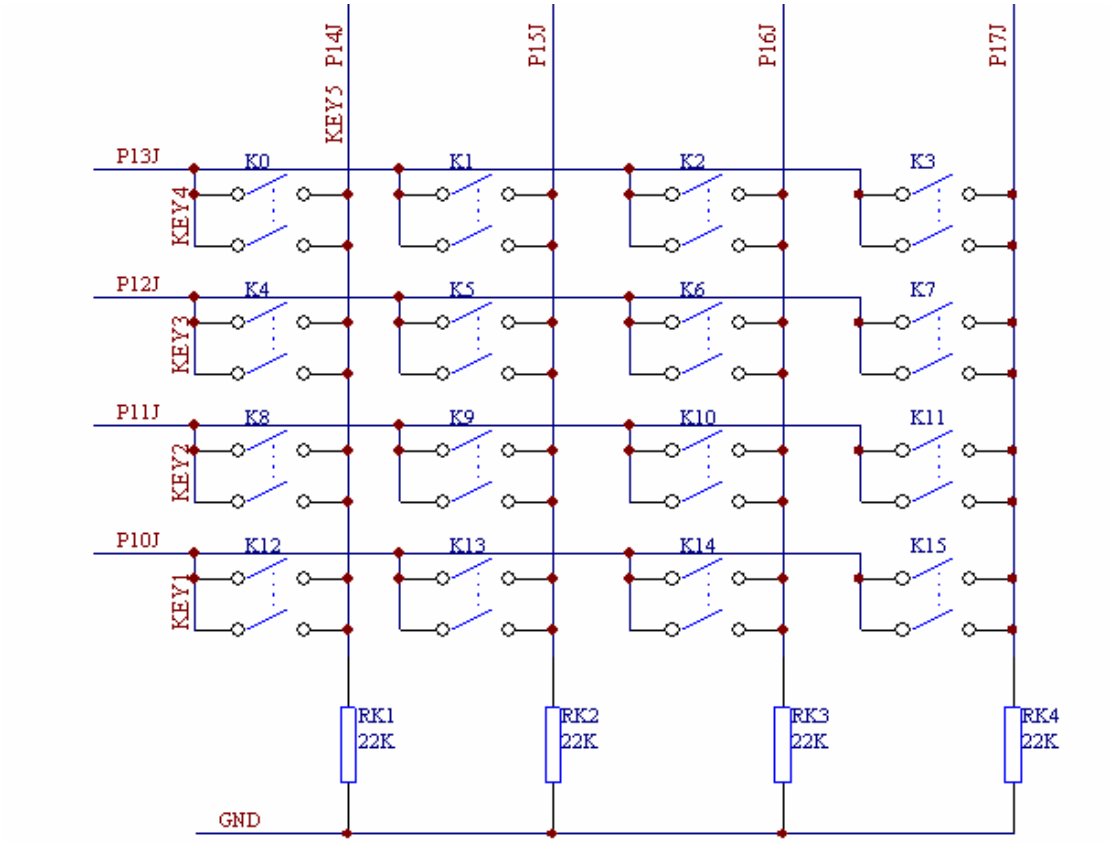
键盘显示应用

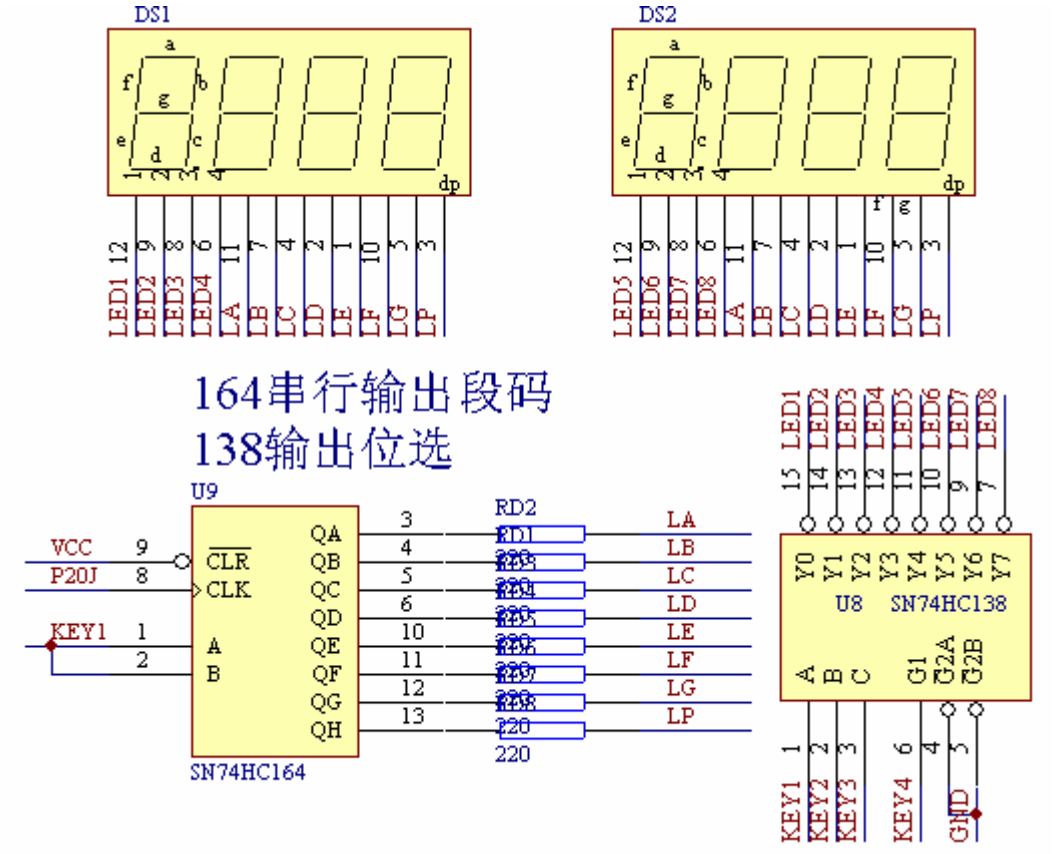
南京航空航天大学　　魏小龙

硬件连接：

1、 4×4 扫描键盘连接在 P1 端口，P10---P13 为行线，P14---P17 为列线，列线下拉到地。

2、 8 只数码管的显示电路，通过 164 串行移位输出 8 位段码，138 位 8 只数码管的位选信号。也都连接在 P1 端口上。与键盘复用引脚。





软件分析：

1、 扫描显示。

扫描显示的原理在于利用人眼睛的视觉暂停，让每个数码管只显示一点时间，所有的数码管轮流显示，而人眼睛看起来就象所有的都在显示一样。所以硬件上所有的数码管的段码端都连接在一起，而每一个数码管的公共端（地）不连接在一起，而由 138 选中每个是数码管。

所以软件上，很明显，分几个步骤。

第一、将要显示的数码转换为段码。可使用查表的方式。比如要显示“1、2、3、4、5、6、7、8”分别在 8 个数码管上，首先安排段码表，设置一个数组 seg[]：

unsigned char seg[]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71}

则 5 的段码为 seg[5]。

第二、将要显示的段码输出，这里使用 164 移位输出。

第三、每输出一个要显示的段码，则使用 138 选中应该显示的数码管。

第四、延时一小段时间。这个时间不能长，也不能短。太长则 8 只数码管看起来很抖动，太短则 8 只数码管一片模糊。

第五、循环第一到第四。

2、 扫描键盘。

本程序的目的在于知道那个键被按下了。二实现此目的需要以下步骤。

第一、 判断有没有键被按下。

第二、 如果有，消除键盘的机械抖动。这是必须的，抖动是机械键盘必然存在的。

第三、 判断是哪个键被按下。

第四、 等待按键松开。这也是必须的，不然将会输出很多重复的按键值。

3、 将按键的键值显示的数码管上。

调用键盘程序，调用显示程序即可。

具体程序编写。

1、 显示。

延时程序是必须的。

```
void delay(int v)
      {
            while(v!=0)v--;
      }
```

要显示的数据放在数组 disbuffer[ ]中。数组的第一个数据对应第一个数码管。

```
void disp(void)
 {
 unsigned char i=0;
 unsigned char temp_wei=0x04,temp_duan=0 ;
 P1DIR   =   0x1f ;
 for(i=0;i<8;i++)
    {
      P1OUT   &=   ~BIT3 ;
      temp_duan=seg[disbuffer[i]] ;
      for(j=0;j<8;j++)
            {
             if(temp_duan&0x80)       P1OUT   |=     BIT0 ;
             else         P1OUT   &=   ~BIT0;
             temp_duan=temp_duan<<1;
             P2OUT   &=     ~BIT0;     P2OUT   |=   BIT0;
            }
       P1OUT   =   (P3IN&0xf8) | temp_wei;
       P1OUT   |=     BIT3;
       temp_wei++;
       delay(80);
    }
  for(i=0;i<8;i++)
  { P1OUT   &=   ~BIT0;     P2OUT   &=     ~BIT0;     P2OUT   |=   BIT0;
  }
 P1DIR=0XF;
 P1OUT=0XF;
 P1IFG=0;
 P1IE=0xf0;
 }
```

此程序有个小小的细节，就是 8 位数码管全显示完毕后全不显示（最后一个循环语句）。其原理为，如果继续显示最后一位的数据，则效果为最后一位比其他的亮一些。而这样处理后，所有的 8 位数码管的亮度均匀。

2、 键盘程序

1） 判断是否有按键程序。

```
uchar key_just(void)       //00:down    0xff:no down
    {
    P1DIR=0xf;
    P1OUT=0xf;
    if((P1IN&0xf0)==0)
            return(0xff);
    return(0);
    }
```

2） 扫描得到键值。下面的程序非常直观。使用扫描的办法，由第一根行线开始一次扫描，如果扫描到则返回对应的值。4 根行线全扫描一遍。

```
uchar key_code(void)
      {uchar in;
       P1DIR=0XF;
       P1OUT=0x8;
       in=P1IN&0x0f0;
       if(in!=0)
            {
            if(in==0X10)
               return(0); //C-F
            if(in==0X20)
               return(1);
            if(in==0X40)
               return(2);
            if(in==0X80)
               return(3);
             }
       P1OUT=0X4;
       in=P1IN&0x0f0;
       if(in!=0)
            {
            if(in==0X10)
               return(4);      //8-B
            if(in==0X20)
               return(5);
            if(in==0X40)
               return(6);
            if(in==0X80)
               return(7);
            }
       P1OUT=0X2;
       in=P1IN&0x0f0;
       if(in!=0)
            {
            if(in==0X10)
```

```
                return(8);      //4-7
            if(in==0X20)
                return(9);
            if(in==0X40)
                return(10);
            if(in==0X80)
                return(11);
            }
        P1OUT=0X1;
        in=P1IN&0x0f0;
        if(in!=0)
            {
            if(in==0X10)
                return(12);     //0-3
            if(in==0X20)
                return(13);
            if(in==0X40)
                return(14);
            if(in==0X80)
                return(15);
             }
          return(0xff);
       }
```

3） 键盘抖动消除与键盘松开的等待。这些事情将在键盘主程序中完成。

```
uchar key(void)
    {
    uchar temp= key_buffer      ;//=key_buffer;
    P1IFG=0;
    if(key_just()==0)
        delay(500);
    if(key_just()!=0)
        return(temp);
    temp=key_code();
    key_buffer=temp;
key_loop:
    if(key_just()==0)
            goto   key_loop;
    else
        {
        P1IFG=0;
        return(temp);
        }
    }
```

3、将按键值显示在数码管上。

```
Void        main (void)
    {
    char i=0,j=0;
    for(;;)
        {
        j=key();
        for(i=1;i<8;i++)
        disbuffer[i]=j;
        }
    }
```

主程序的思考

本程序的思路还是很清楚的：主程序就是将按键的键值送数码管显示。这里还是有值得思考的地方，整个程序的结构，是使用死循环呢，还是使用中断呢。两部分都可以使用比如下面方案：

1、 可以中断作显示（定时器定时调用显示）；

2、 也可以在主循环中使用不断调用显示；

3、 可以在主程序中只使用键盘程序，而在键盘的消除抖动的时候使用显示显示程序作延时，而在主程序中不调用任何显示程序；

4、 可以使用主程序只显示，而口线中断的方式实现键盘；

5、 也可以定时查询键盘。

由此可见，在程序设计上，没有固定方法，要看具体的需要。下面具体分析这些方案的利弊。

方案 1、定时器为硬件运行，不受程序的干扰，显示很均匀。不会出现没有调用到显示时，不显示的情况。而且主程序可以在没有事情的时候进入睡眠。

方案 2、是初学单片机最常使用的方法。

方案 3、也是初学单片机最常使用的方法。因为主程序就是判断有什么键按下？该做什么事情？该做的事情做完后再看是什么键按下，又该做什么事情，如果没有键按下，则不断察看。在这里需要有软件的键盘消除抖动，就直接调用显示程序。

方案 4、方案 5 大同小异，只是键盘的中断来源不一样。