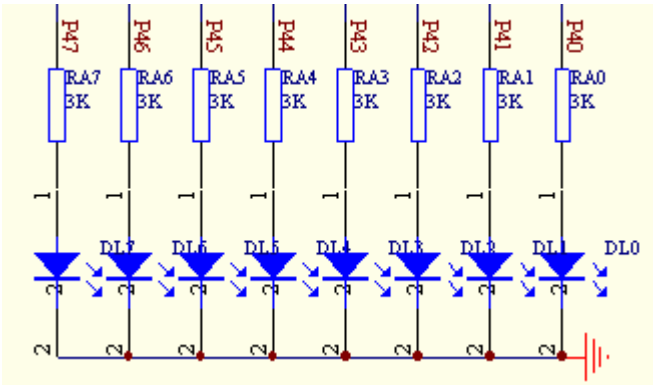
简单的端口、显示、中断综合应用

南京航空航天大学　　魏小龙

本讲将结合定时器、端口、中断等 430 资源进行综合应用。具体要求如下：

1、 硬件连接很简单，在 P4 端口连接了 8 只发光二极管 LED0-7，在 P1 端口连接了 4×4 键盘（16 键）。P1 同时通过 138、164 扩展了 8 位数码显示器。



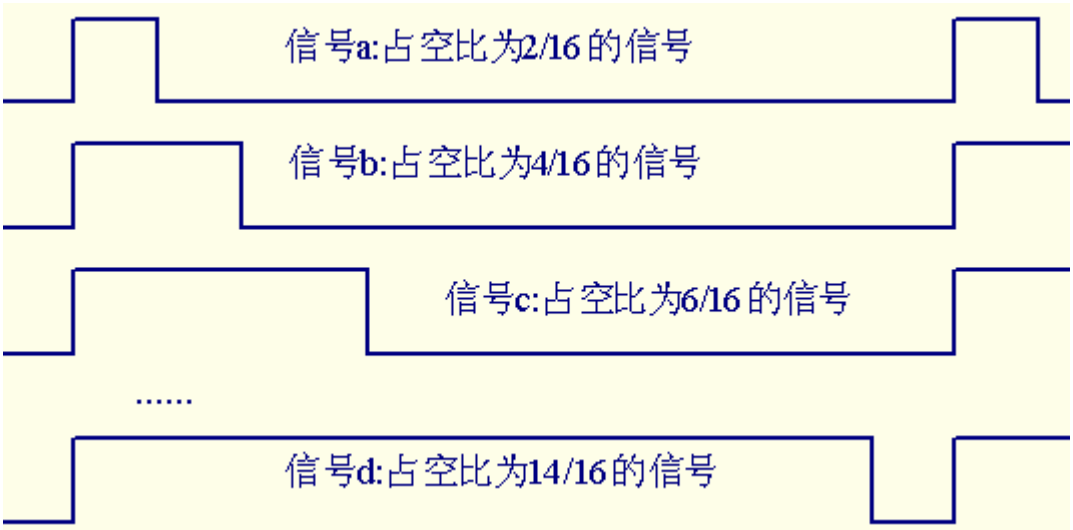
2、 软件要求:

a) 编写扫描键盘软件；
b) 编写数码管显示程序；
c) 键盘使用中断编写；
d) P4 连接的发光二极管的发光强度通过键盘控制，共分 16 亮度等级；
e) 分别控制单个发光二极管（可对 LED0 到 LED7 进行分别控制其亮度）；
f) 在数码管上显示每只发光二极管的亮度，对应关系为：第一只数码管显示的数值对应第一只发光二极管的亮度，后面依次类推，第八只数码管显示的数值对应第八只发光二极管的亮度。

分析：

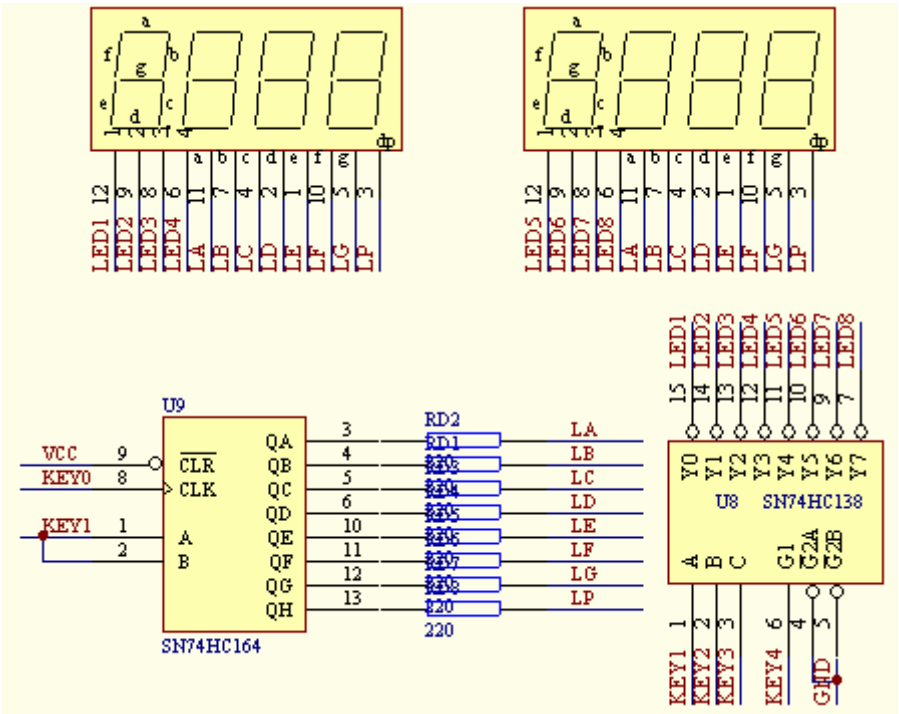
首先编写基本硬件程序，键盘与数码管显示已经在上一讲说清楚了，这里只管调用就可以了。发光二极管显示也很简单，高电平亮，低电平熄。

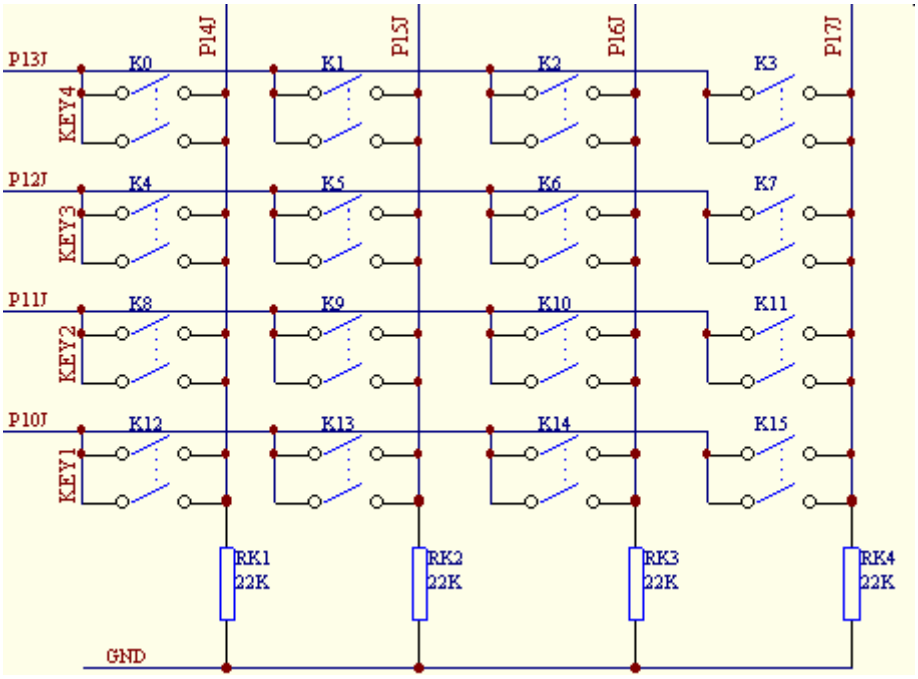
那么如何实现发光二极管 LED1-8 显示的亮暗调节呢。先看看下面的 4 个不同信号 a、b、c、d。假设他们的周期都是 100Hz，则这些信号送发光二极管后，我们看不到发光二极管的亮与熄，我们看到的都是亮，只不过亮度不一样而已（因为人眼睛的视觉暂停缘故）。很显然，信号 a 驱动发光二极管最暗，信号 d 驱动发光二极管最亮。本讲所使用的硬件有 16 只扫描键盘，我们定义按 0 号按键显示最暗，按 F 号按键显示最亮。



如何实现分别对单个发光二极管亮度的控制呢，根据软件要求 e,我们可以将亮度数据存放在显示缓存里，则 disbuffer[0]的数据表示 LED0 的亮度，则 disbuffer[1]的数据表示 LED0 的亮度.....这样只要在显示缓存中取对应的亮度数值去控制发光二极管的显示就可以了。

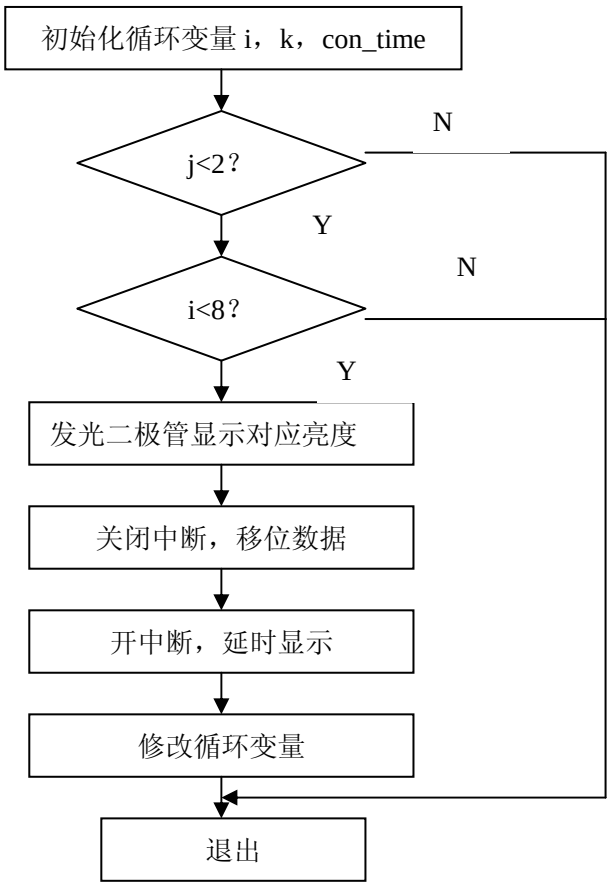
数码管的显示方法为：在显示缓存中取数据移位到驱动对应数码管的 74HC164，并延时显示一会儿。这一讲的硬件共有 8 只数码管，需要 8 次移位与延时等待，而亮度等级有 16 级，所以可以使用两次移位，也就是显示程序再来一遍，构成 16 次。这样可以将显示程序作为整个系统的主程序，发光二极管的亮度控制内嵌显示程序中，则要将上一讲的显示程序进行改造。同时需要注意：键盘与显示程序的兼容性能，这一讲要求键盘使用中断，而显示与键盘合用口线，见下图：





因为是合用口线，所以在显示程序中如果改变了键盘中断线的状态而满足了中断条件，则必然会进入中断，因为 CPU 认为是有按键了，从而进入了键盘中断，所以要避免这种情况。具体方法：在显示程序中进行端口操作之前关闭中断，在端口操作完成之后，清除端口的中断标志，然后打开中断。只有在显示的延时显示时间段，键盘才可能中断。这样就避免了显示程序对键盘的干扰，同时也实现了口线的合用。

很显然，显示程序可以直接作为整个程序的主循环！显示程序的框图：



发光二极管显示对应亮度的程序思路：

在显示程序中每显示一个数码管，变量 con_time 增一，则在显示程序的开始判断与控制发光二极管的显示亮度：如果第一个发光二极管的亮度值为 1，则在显示程序的 16 次循环中只有第一次亮，如果第一个发光二极管的亮度值为 5，则在显示程序的 16 次循环中只有前 5 次亮，其他都熄。

显示程序如下：

```
void disp(void)
    {
    unsigned char i=0,j=0,k=0,con_time=0;
    unsigned char temp_wei=0x04,temp_duan=0;
    _DINT();
    for(k=0;k<2;k++)          //两次循环，构成 16 个时间段
    {
    P1DIR   =   0x1f;
    for(i=0;i<8;i++)          //显示 8 个数码管
        {
        P4OUT=0;                          //亮度显示程序
        if(disbuffer[0]>con_time)         //下面将分别显示 8 只的不同亮度
            P4OUT   |=    BIT0;
        if(disbuffer[1]>con_time)
            P4OUT   |=    BIT1;
        if(disbuffer[2]>con_time)
            P4OUT   |=    BIT2;
        if(disbuffer[3]>con_time)
            P4OUT   |=    BIT3;
        if(disbuffer[4]>con_time)
            P4OUT   |=    BIT4;
        if(disbuffer[5]>con_time)
            P4OUT   |=    BIT5;
        if(disbuffer[6]>con_time)
            P4OUT   |=    BIT6;
        if(disbuffer[7]>con_time)
            P4OUT   |=    BIT7;
        P1OUT   &=   ~BIT3;                   //关闭所有数码管显示
        temp_duan=seg[disbuffer[i]] ;         //移位输出
        for(j=0;j<8;j++)                      //移出一个段码
            {
             if(temp_duan&0x80)
                 P1OUT   |=    BIT0;
             else
                 P1OUT   &=   ~BIT0;
             temp_duan=temp_duan<<1;
```

```
                P1OUT  &=   ~BIT4;   P1OUT  |=  BIT4;
               }
            P1OUT  =  (P3IN&0xf8) | temp_wei;
            P1OUT  |=   BIT3;
            temp_wei++;
            P1IFG=0;
            _EINT();
            delay(200);                          //延时显示一会儿，同时等待键盘中断
            con_time++;
            _DINT();
           }
       for(i=0;i<8;i++)                          //清除数码管显示，使得显示均匀
           {
           P1OUT  &=  ~BIT0;   P1OUT  &=   ~BIT4;   P1OUT  |=  BIT4;
           }
       }
     P1DIR=0XF;
     P1OUT=0XF;
     P1IFG=0;
     P1IE=0xf0;
     }
```

在中断中输入对应的发光二极管的亮度值程序：

```
interrupt[PORT1_VECTOR] void p1_iint(void)
     {
     uchar j,temp;
main_loop:
            P1IFG=0;
     disp();
     if(key_just()!=0)              //如果没有按键就等待按键
          goto main_loop;
     temp=key_code();               //有，判断键值
     key_buffer=temp;               //第一个键值是将修改的数码管的位数
key_loop_m1:
     if(key_just()==0)
        goto   key_loop_m1;      //等松开按键
     j=temp;
     if(j>=8)
          goto   main_loop;
main_loop2:
     disp();
     if(key_just()!=0)
          goto main_loop2;
```

```
    temp=key_code();              //得到第二键值
    key_buffer=temp;
key_loop_m2:
    if(key_just()==0)
        goto   key_loop_m2;
    disbuffer[j]=temp;            //送对应数码管显示
      P1IFG=0;                    //清除因按键引起的可能中断标志
    }
```

系统主程序：

首先初始化；

然后主循环：调用显示程序。

显示程序同时用数码管显示各个发光二极管的亮度等级值与发光二极管的不同亮度显示。在显示的延时中等待键盘中断，输入哥参数。