第三讲 键盘扫描的学习

提要：主要学习 Atmega8 通用数字 I/O 接口扫描键盘的应用。

前面我们学习了 ATmega8 的 I/O 口作为通用数字输入/输出口来用时对 LED 数码管控制的应用，其实主要是作为输出口的应用。下面我们就来学习一下用作输入口的应用－－扫描键盘。

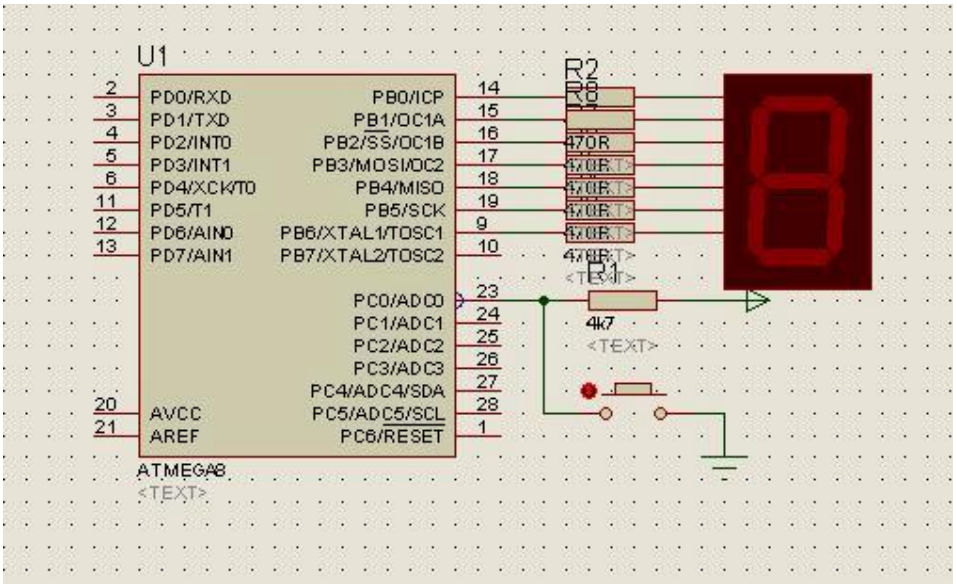
3.1 按键的使用特点：

按键的应用主要就是在按键闭合时改变电路的电平，但是一般情况下按键的开关都是机械弹性触点开关，它利用触点的接触和分离来实现电路的通断，在按键按下和释放时往往会产生抖动干扰，所以要想办法消除抖动干扰。

消除抖动干扰即可用硬件设计也可用软件设计的方法。硬件消抖就要在硬件设计上增加消抖电路，如用 R-S 触发器等，这样就会增加系统成本。软件消抖就是要在软件中对按键进行二次测试确认，既当第一次检测到按键被按下后，间隔 10 毫秒左右再次检测该按键是否被按下，只有两次都册到按键按下时才确认该按键被按下了，从而消除抖动干扰。另外我们还要判别按键的释放，为了设计方便，我们现不考虑长时间按下按键的情况，只简单考虑检测到按下和释放才计作一次按键操作。

3.2 单键盘扫描的应用

下面我们就用软件消抖的方法来做个简单的应用，我们用PB 口接一个 LED 数码管，用来显示数据，用 PC0 端口接一个按键电路，我先画出电路：



我们要实现的功能是每按一次按键，LED 数码管显示的数据加 1，到 9 回 0。怎么样不是很复杂吧，是不是很快就可以些出来了呢？下面是我写的，你也参考一下吧：

```
/*****************************************/
//文件名：OneKey.c
//功能：  按键扫描的简单应用
//作者：  young
//时间：  2006.11.6
//目标MCU:ATmega8
//晶振：  8MHZ
/*****************************************/
#include <iom8v.h>
#include "Delay.h"

unsigned char CountNum;   //全局变量，用来计数

//按键扫描函数
void ScanKey(void)
{
    unsigned char key;
    key=PINC;
    if(1==key&0x01)
        return;
    delay_ms(10);

    key=PINC;
    if(1==key&0x01)
        return;
    CountNum++;
    while(0==key&0x01)
        key=PINC;
}
```

```
//主函数，扫描按键显示数据
void main()
{
    unsigned char num[10]={0x3F, 0x06, 0x5B, 0x4F, 0x66,
                           0x6D, 0x7D, 0x07, 0x7F, 0x6F};
    //初始化端口
    DDRB=0xFF;    //设置B口为输出模式
    PORTB=0xFF;   //置高电平
    DDRC=0x00;
    //PORTC=0xFF;

    CountNum=0;   //初始化全局变量
    while (1)
    {
        ScanKey();//扫描按键
        if(CountNum>=10)
            CountNum-=10;
        PORTB=num[CountNum];
     }
}
```

上面的程序没考虑按键长按的情况，如果象我们使用的键盘一样，长时间按下一个按键，在屏幕上就不断的打印该字符，在这个例子里就是长时间按下按键

的话就对 CountNum 加一，而不是每次按下按键一次就加一一次，该怎么考虑呢，你可以试试能不能实现。

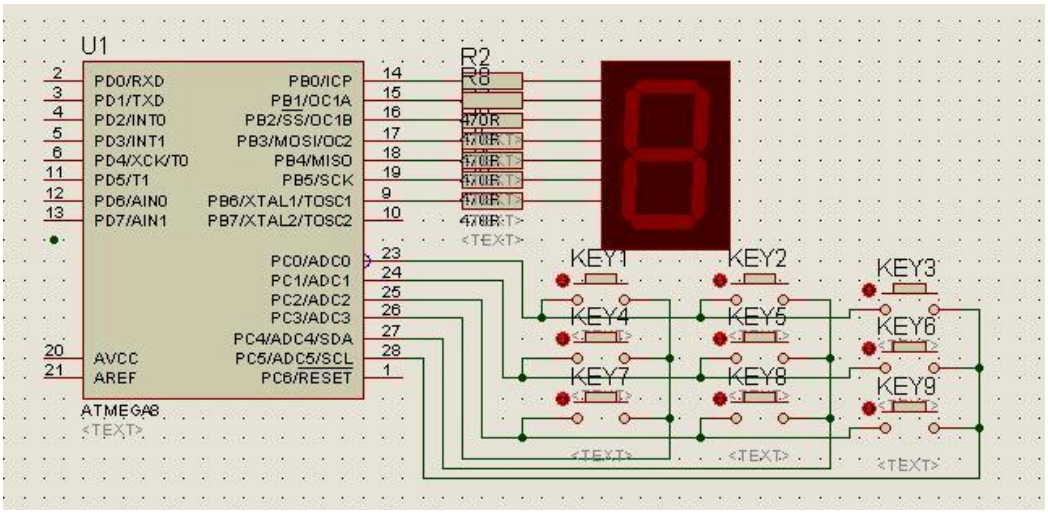
3.3 矩阵按键（键盘）扫描的应用

如果按键多的话，我们一般就要考虑节省 I/O 资源，通常会考虑采用矩阵式的接口。矩阵键盘由行和列组成，每个按键都有它的行值和列值，行值和列值的组合就是识别每个键盘的编码。

确定哪个按键的流程是：先在行和列的一个口中输出高电平，在另一个行列口读取一个扫描码；然后在后一个行列口输出高电平，在前一行列口读取第二个扫描码，然后查表就可确定哪个按键被按下了。

有了确定哪个按键的方法，就简单多了。键盘的处理程序也就基本出来了，因为它个单按键的扫描程序是很类似的：先确定有无按键按下，然后确定是哪个按键，返回该键值或处理对应的任务就可以了。当然这里也要考虑抖动的消除，等待按键的断开。

下面我们就设计一个 3X3 的键盘来学习多按键的应用。我先画出电路图，然后再描述要实现的功能：



我们要实现的目标是每按下一个按键就要在 LED 数码管中显示出该按键对应的值，按键断开后显示”一”，这个符号不在前面介绍的内容中，你可以试着计算一下该值应该是多少。好了，我给出一个参考：

```
/*****************************************/
//文件名：3X3Key.c
//功能：  键盘扫描的应用
//作者：  young
//时间：  2006.11.6
//目标MCU:ATmega8
//晶振：  8MHZ
/*****************************************/
#include <iom8v.h>
#include "Delay.h"
```

```
//按键扫描函数，返回按键的值
unsigned char ScanKey(void)
{
    unsigned char temp,temp1,key;
    temp=PINC;
    temp&=0x07;
    switch(temp) //判断行中哪条线有低电平
    {
    case 0x06:
        DDRC=0X07;  PORTC=0X38;
        delay_us(1);
        temp1=PINC; temp1&=0x38;
        switch(temp1) //判断列中哪条线有低电平
        {
            case 0x30: key=0x01;  //得到键值
                break;
            case 0x28: key=0x02;
                break;
            case 0x18: key=0x03;
                break;
            default:   key=0;
                break;
        }
        DDRC=0X38;  PORTC=0X07;
        break;

    case 0x05:
        DDRC=0X07;  PORTC=0X38;
        delay_us(1);
        temp1=PINC; temp1&=0x38;
        switch(temp1)
        {
        case 0x30: key=0x04;
            break;
        case 0x28: key=0x05;
            break;
        case 0x18: key=0x06;
            break;
        default:   key=0;
            break;
        }
        DDRC=0X38; PORTC=0X07;
        break;
```

```
    case 0x03:
        DDRC=0X07;  PORTC=0X38;
        delay_us(1);
        temp1=PINC; temp1&=0x38;
        switch(temp1)
        {
        case 0x30: key=0x07;
            break;
        case 0x28: key=0x08;
            break;
        case 0x18: key=0x09;
            break;
        default:   key=0;
            break;
        }
        DDRC=0X38; PORTC=0X07;
        break;
    default:
        key=0;
        break;
    }
    return (key);
}
```

```
//主函数，扫描按键显示数据
void main()
{
    unsigned char temp, keynum;
    unsigned char num[10]={0x3F, 0x06, 0x5B, 0x4F, 0x66,
                           0x6D, 0x7D, 0x07, 0x7F, 0x6F};
    //初始化端口
    DDRB=0xFF;     //设置B口为输出模式
    PORTB=0xFF;    //置高电平
    DDRC=0X38;
    PORTC=0X07;

    while(1)
    {
       PORTB=0x40;
       temp=PINC;
       temp&=0x7;
       if(temp==0x7)//检测是否有按键按下
           continue;
       delay_ms(10);
       temp=PINC;
       temp&=0x7;
       if(temp==0x7)//检测是否有按键按下
           continue;
       keynum=ScanKey();
       PORTB=num[keynum];
       while(temp!=0x7)
       {
           temp=PINC;
           temp&=0x7;
       }
    }
}
```

好了，我是在主函数中判断是否有按键按下，然后消除抖动干扰的，然后用 ScanKey 函数的到按键值，显示在 LED 数码管中，最后等待按键释放；ScanKey 的功能主要就是的到扫描码确定是哪个按键，返回该按键的值。

除了像上面的对按键的接口不停的扫描，还可以使用定时扫描，例如用一个定时器，每隔 10MS 对按键接口进行扫描，看是否有按键按下；也可以使用中断的方式去扫描，当按键按下时由硬件电路产生一个中断，MCU 响应该中断，确定哪个按键被按下，处理相应函数。这些内容在后面讲到时钟和中断的时候会再作介绍，先在这里提一下，您好有个数。

好了，关于按键的应用我们先学习到这里，你可以休息一下然后进行新的内容了。