

第5课，标记的用法，用一个按键控制1个LED灯的亮灭，按键去抖

这一课，我们学习怎么用一个按键 K1 控制 1 个 LED 灯的亮和灭两种状态。按一次 K1 灯亮，再按一次 K1 灯灭。再按一次又亮，再按一次又灭。

我们学习一下用一个 bit 变量来做一个标记，然后在按键的控制下，这个标记会变化，再根据这个标记的值，LED 也输出不同的状态。

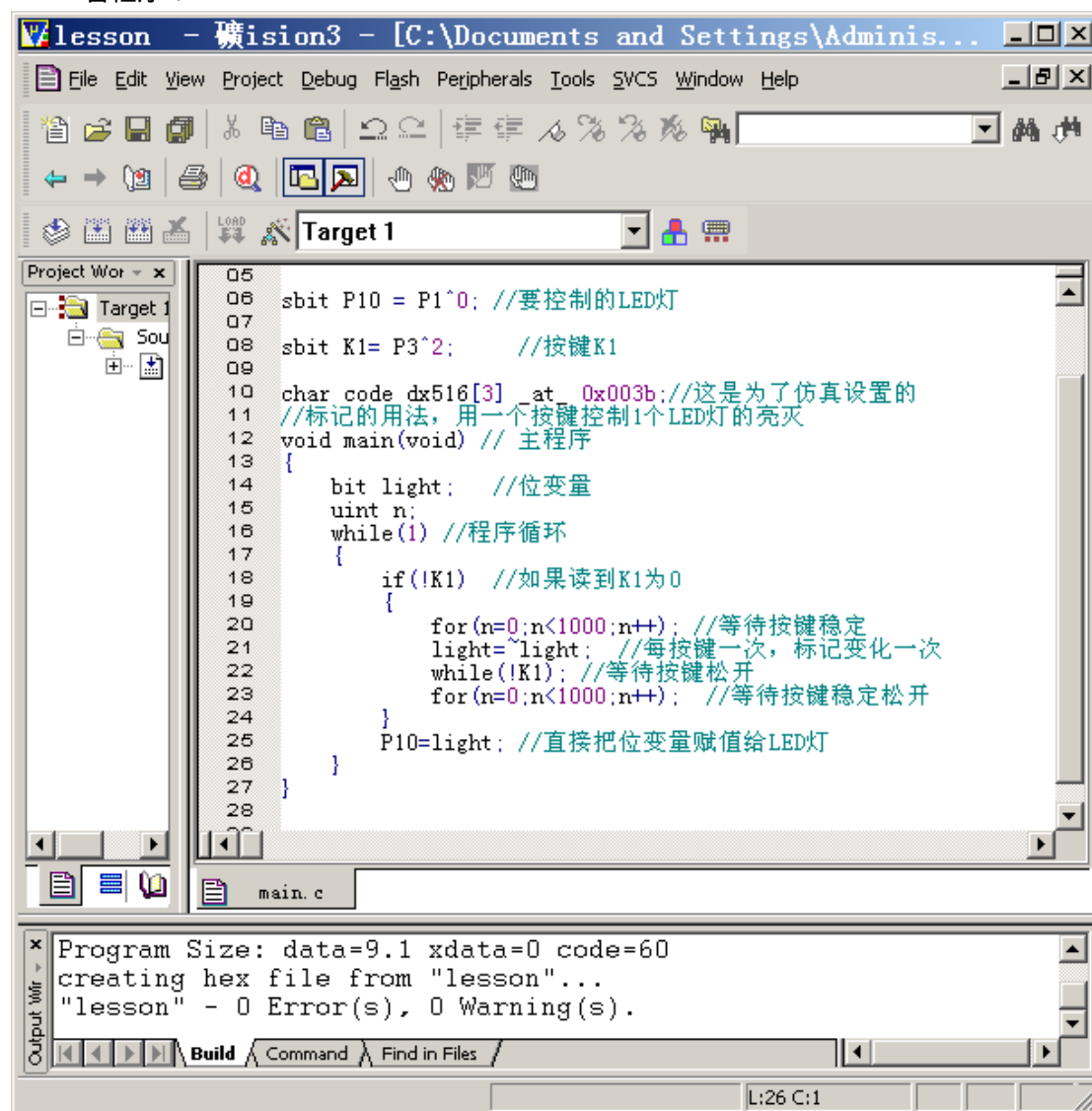
因为按键按下时可能会有抖动情况，每次按下时，可能会发生了人难以觉察到的多次抖动，相当于一下子按下了很多次。这会导致程序无法识别出您真正的按键意图。

但是抖动一般都是发生在刚按下键和松开键的时候，所以，我们只要避开这一段时间，等键稳定按下或者松开时，再去读它的值，一般就可以正确读取了。

所以，当读到第一次按键的值时，要延时等待一会，再处理。在松开后，也延时一会，免得检测到松开的抖动以为又有按键。（注，更复杂的应用，需要在按下延时之后重新验证按键，为了简化和方便理解，这个例程里没有这样做。）

另外，因为程序是循环运行的，当一次按键处理后，又会再循环回来继续检测，如果您的按键这时还没有松开，又会被读到一次新的按键，并做处理。所以我们还要做一个特殊的处理，识别到一个按键并处理完成之后，还要等待这个按键松开后，再继续循环运行。

看程序：



```
05
06 sbit P10 = P1^0; //要控制的LED灯
07
08 sbit K1= P3^2;    //按键K1
09
10 char code dx516[3] _at_ 0x003b; //这是为了仿真设置的
11 //标记的用法，用一个按键控制1个LED灯的亮灭
12 void main(void) // 主程序
13 {
14     bit light; //位变量
15     uint n;
16     while(1) //程序循环
17     {
18         if(!K1) //如果读到K1为0
19         {
20             for(n=0;n<1000;n++); //等待按键稳定
21             light=~light; //每按键一次，标记变化一次
22             while(!K1); //等待按键松开
23             for(n=0;n<1000;n++); //等待按键稳定松开
24         }
25         P10=light; //直接把位变量赋值给LED灯
26     }
27 }
28
```

Output Win - x
Program Size: data=9.1 xdata=0 code=60
creating hex file from "lesson"...\n"lesson" - 0 Error(s), 0 Warning(s).

Build Command Find in Files

L:26 C:1

请根据例程里的注释理解程序。

请编译，进入仿真，全速运行，看结果。

全速后，由于 light 变量初始化时默认为 0，所以灯是亮的。按下 K1，松开后，灯灭了；再按一次 K1,松开后，灯灭了。

这个例子里，我们只用一个按键就控制了灯的亮灭，这种方法可以节省了硬件资源，也就是节省了硬件成本。在实际项目设计中，有成本优势，产品就更具竞争力。所以我们应该多学习类似的可以节省资源的方法。

作业：

改为 4 个按键，分别控制 4 个 LED 的亮和灭。相当应用到多个房间的单键开关灯共用一个 cpu 处理。