

## 第 13 课，用外中断方式读按键，控制灯的亮灭

这一课，我们学习外中断的用法。也就是外部 IO 的中断 INT0，和 INT1。对应的引脚是 P32 和 P33。在我们的电路图中，P32 也就是接在 K1 的引脚。所以当我们按下 P32 接到地的时候，可以触发一个 INT0 中断，当然，必须预先初始化才会启动。

这种中断方式的按键，可以实现按键的立即响应。对于需要快速响应的场合是很有用的。外部 IO 中断还常用在用 IO 模拟通讯的场合，可以对数据的到来立即响应。

下面请看代码：

```
#define uchar unsigned char //定义一下方便使用
#define uint   unsigned int
#define ulong  unsigned long
#include <reg52.h>          //包括一个 52 标准内核的头文件

sbit P10 = P1^0; //头文件中没有定义的 IO 就要自己来定义了
sbit P11 = P1^1;
sbit P12 = P1^2;
sbit P13 = P1^3;
sbit K1 = P3^2;

bit ldelay=0; //长定时溢出标记,预置是 0
uchar speed=10; //设置一个变量保存默认的跑马灯的移动速度
uchar speedlever=0; //保存当前的速度档次

char code dx516[3] _at_ 0x003b; //这是为了仿真设置的
//用外中断方式读按键 K1，点亮一个 LED
void main(void) // 主程序
{
    IT0=1; //外中断跳变产生中断
    EX0=1;
    EA=1; //打开总中断

    while(1) //主程序循环
    {
    }
}
//外中断 0
int0() interrupt 0
{
    P10=0; //在中断里点亮 LED
}
```

这个程序里，按一下 K1(P32)之后，就会触发 INT0 中断，在该中断里点亮 LED 灯。  
请编译运行，并看结果。可以看到，在按下 K1 之后，LED1 变处于亮着的状态。

作业：

用中断方式的按键 K1，控制一个 LED 灯的亮和灭两种状态

提示：需要去抖操作，否则按键抖动会引起多种中断。