

第 11 课，一个按键控制的 10 级变速跑马灯试验

在本课中，我们要用一个按键来实现跑马灯的 10 级调速。这又会涉及到键的去抖的问题。

本课的试验结果是，每按一次按键，跑马速度就降低一级，共 10 级。

这里我们又增加了一个变量 speedlever，来保存当前的速度档次。

在按键里的处理中，多了当前档次的延时值的设置。

请看程序：

```
#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
//一个按键控制的 10 级变速跑马灯试验
void main(void) // 主程序
{
    uchar code ledp[4]={0xfe,0xfd,0xfb,0xf7}; //预定的写入 P1 的值
    uchar ledi; //用来指示显示顺序
    uint n;

    RCAP2H =0x10; //赋 T2 的预置值 0x1000，溢出 30 次就是 1 秒钟
    RCAP2L =0x00;
    TR2=1; //启动定时器
    ET2=1; //打开定时器 2 中断
    EA=1; //打开总中断

    while(1) //主程序循环
    {
        if(ldelay) //发现有时间溢出标记，进入处理
        {
            ldelay=0; //清除标记
            P1=ledp[ledi]; //读出一个值送到 P1 口
```

```

        ledi++;      //指向下一个
        if(ledi==4)
        {
            ledi=0; //到了最后一个灯就换到第一个
        }
    }
    if(!K1) //如果读到 K1 为 0
    {
        for(n=0;n<1000;n++); //等待按键稳定
        while(!K1); //等待按键松开
        for(n=0;n<1000;n++); //等待按键稳定松开

        speedlever++;
        if(speedlever==10)speedlever=0;
        speed=speedlever*3; //档次和延时之间的预算法则，也可以用查表方法，做出
不规律的法则
    }
}
//定时器 2 中断
timer2() interrupt 5
{
    static uchar t;
    TF2=0;
    t++;

    if((t==speed)||(t>30)) //比较一个变化的数值，以实现变化的时间溢出,同时限制了最慢速度为 1 秒
    {
        t=0;
        ldelay=1; //每次长时间的溢出，就置一个标记，以便主程序处理
    }
}

```

请打开 lesson11 目录的工程，编译，运行，看结果：

按 K1，速度则降低一次，总共 10 个档次。

作业：

用查表的方法改变每个档次的跑动速度，而不是例程里的档次和延时的 3 倍关系。