

第 10 课，4 个按键 4 级变速的跑马灯试验，多任务的工作方式

这一课，我们要用 4 个按键，控制跑马灯的 4 种不同的跑动速度。

按键的控制我们也做过了，结合跑马灯，很容易程序就出来了。只是每按一个键，就赋给一个不同的定时器 2 溢出次数而已。

我们设置为 1 秒，1/2 秒，1/5 秒，1/10 秒四个档次，分别时 K1 - K4 控制。

这个程序的主程序执行了 2 个任务。一个是跑马灯，一个是检测按键。程序的结构非常清晰。

程序如下：

```
#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;
sbit K2= P3^5;
sbit K3= P2^4;
sbit K4= P2^5;

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

char code dx516[3]_at_ 0x003b;//这是为了仿真设置的
//自动变速的跑马灯试验
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

    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) speed=30; //检查到按键，设置对应的跑马速度
    if(!K2) speed=15;
    if(!K3) speed=6;
    if(!K4) speed=3;
}
}
//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
    {
        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 lesson10 目录的工程，编译，运行。

可以看到，启动后，以默认的速度跑马，按 K1，速度是 1 秒一个灯，按 K2，是 1/2 秒一个灯，按 K3 是 1/5 秒一个灯，按 K4，则最快，是 1/10 秒。

作业：

修改为其他的速度档次