

## 第9课，自动变速的跑马灯试验

这一课，我们仍然使用上一个定时器跑马灯工作方式，但是我们让跑动的速度自动变化，从慢到快。

相对于上一颗的跑马灯试验，我们新设置了一个变量 speed，用来保存跑马灯的移动速度，其实也就是定时器的累计时间溢出次数。

我们在程序中修改 speed 的数值，溢出的时间就会改变，跑马灯的移动速度也就改变了。我们是在每循环跑完一圈，就改变一次速度的。

请仔细研究代码，做到充分理解。

源代码如下：请打开对应目录里的例程，和下面的代码是一样的。

```
#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;
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
            speed--;
            if(speed==0)speed=10;//每循环显示一次，就调快一次溢出速度
        }
    }
}
//定时器 2 中断
timer2() interrupt 5
{
    static uchar t;
    TF2=0;
    t++;
    if(t==speed) //比较一个变化的数值，以实现变化的时间溢出
    {
        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
    }
}

```

请编译，运行，并查看结果。

作业：修改为，跑马灯从慢到快后，再从快到慢，而不是一下子变成最慢。