

第 12 课，可编程自动控制跑马灯

这一课，我们学习如何让跑马灯自动按照我们预定的顺序进行。这种控制在工控场合经常用到。

这个程序里，我们预先定义了一个变化的顺序 speedcode，每跑一圈灯就根据预定设置的表格数据来决定下一圈的跑马速度。这样我们就实现了按照预定的顺序自动变化运行。

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;
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
uchar code speedcode[10]={3,1,5,12,3,20,2,10,1,4}; //10 个预定义的速度

char code dx516[3]_at_ 0x003b;//这是为了仿真设置的
//可编程自动控制跑马灯
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
    uchar i;

    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

            //每跑一圈灯就根据预定设置的表格来决定下一圈的跑马速度
            speed=speedcode[i];
            i++;if(i==10)i=0;
        }
    }
}
//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，并看运行结果。

作业：改为每跑 2 圈才改变一次跑马速度。