

## 第 14 课，模拟 PWM 输出控制灯的 10 个亮度级别

LED 一般是恒流操作的，如何改变 LED 的亮度呢？答案就是 PWM 控制。在一定的频率的方波中，调整高电平和低电平的占空比，即可实现。比如我们用低电平点亮一个 LED 灯，我们假设把一个频率周期分为 10 个时间等份，如果方波中的高低电平占空比是 9：1，这就是一个比较暗的亮度，如果方波中高低电平占空比是 10：0，这时，全部是高电平，灯是灭的。如果占空比是 5：5，就是一个中间亮度，如果高低比是 1：9，是一个比较亮的亮度，如果高低是 0：10，这时全部是低电平，就是最亮的。

实际上应用中，电视屏幕墙中的几十百万 LED 像素都是这样控制的，而且每一个像素都有红绿蓝 3 个 LED，每个 LED 可以变化的亮度是几百到几万或者更多的级别，以实现真彩色的显示。还有在您的手机中，背光灯的亮度如果是可以变化的，也应该是这种工作方式。目前的城市彩灯也有很多都使用了 LED，需要控制亮度也是 PWM 控制。

下面来分析我们的例程，在这个例程中，我们将定时器 2 溢出定为 1/1200 秒。每 10 次脉冲输出一个 120HZ 频率。这每 10 次脉冲再用来控制高低电平的 10 个比值。这样，在每个 1/120 秒的方波周期中，我们都可以改变方波的输出占空比，从而控制 LED 灯的 10 个级别的亮度。

为什么输出方波的频率要 120HZ 这么高？因为如果频率太低，人眼就会看到闪烁感觉。一般起码要在 60HZ 以上才感觉好点，120HZ 就基本上看不到闪烁，只能看到亮度的变化了。

下面请看程序，程序中有比较多的注释：

```
#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; //要控制的 LED 灯
sbit K1 = P3^2; //按键 K1

uchar scale; //用于保存占空比的输出 0 的时间份额,总共 10 份

char code dx516[3] _at_ 0x003b; //这是为了仿真设置的
//模拟 PWM 输出控制灯的 10 个亮度级别
void main(void) // 主程序
{
    uint n;

    RCAP2H = 0xF3; //赋 T2 的预置值，溢出 1 次是 1/1200 秒钟
    RCAP2L = 0x98;
    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) //程序循环
    {
        //主程序在这里就不断自循环，实际应用中，这里是做主要工作
```

```

        for(n=0;n<50000;n++); //每过一会儿就自动加一个档次的亮度
        scale++;
        if(scale==10)scale=0;
    }
}
//1/1200 秒定时器 2 中断
timer2() interrupt 5
{
    static uchar tt; //tt 用来保存当前时间在一秒中的比例位置
    TF2=0;

    tt++;
    if(tt==10) //每 1/120 秒整开始输出低电平
    {
        tt=0;

        if(scale!=0) //这里加这一句是为了消除灭灯状态产生的鬼影
            P10=0;
    }

    if(scale==tt) //按照当前占空比切换输出高电平
        P10=1;
}

```

在主程序中，每延时一段时间，就自动换一个占空比，以使亮度自动变化，方便观察。  
编译，运行，看结果。

可以看到，LED 的亮度以每种亮度 1 秒左右不断变化，共有 10 个级别。

作业：

1. 改用两个定时器的方法驱动 PWM 输出。一个输出占空比，一个输出频率。这样可以产生更少的中断，避免过多的干扰主程序的运行。