第八十七节： 一个定时中断产生 N 个软件定时器。

【87.1 信手拈来的软件定时器。】

初学者会疑惑，51 单片机只有 2 个定时器 T0 和 T1，是不是太少了一点？2 个定时器怎能满足实际项目的需要，很多项目涉及到的定时器往往十几个，怎么办？这个问题的奥秘就在本节的内容。

51 单片机内置的 2 个定时器 T0 和 T1，是属于硬件定时器，硬件定时器是一个母体，它可以孕育出 N 个软件定时器，实际项目中，我们需要多少个定时器只需要从同一个硬件定时器中断里构造出对应数量的软件定时器即可。构造 N 个软件定时器的框架如下：

```
//“软件定时器 1”的相关变量
volatile unsigned char vGu8TimeFlag_1=0;
volatile unsigned int vGu16TimeCnt_1=0;

//“软件定时器 2”的相关变量
volatile unsigned char vGu8TimeFlag_2=0;
volatile unsigned int vGu16TimeCnt_2=0;

//“软件定时器 3”的相关变量
volatile unsigned char vGu8TimeFlag_3=0;
volatile unsigned int vGu16TimeCnt_3=0;

void main()
{
    vGu8TimeFlag_1=0;
    vGu16TimeCnt_1=1000;  //“软件定时器 1”的定时时间是 1000ms
    vGu8TimeFlag_1=1;

    vGu8TimeFlag_2=0;
    vGu16TimeCnt_2=500;  //“软件定时器 2”的定时时间是 500ms
    vGu8TimeFlag_2=1;

    vGu8TimeFlag_3=0;
    vGu16TimeCnt_3=250;  //“软件定时器 3”的定时时间是 250ms
    vGu8TimeFlag_3=1;

    while(1)  //主循环
    {
        if(0==vGu16TimeCnt_1)  //“软件定时器 1”的时间到了
        {
            ...在这里执行具体的功能代码
         }

        if(0==vGu16TimeCnt_2)  //“软件定时器 2”的时间到了
```

```
        {
            ...在这里执行具体的功能代码
        }

        if(0==vGu16TimeCnt_3  //“软件定时器 3”的时间到了
        {
            ...在这里执行具体的功能代码
        }

    }
}

void T0_time() interrupt 1    //每 1ms 中断一次的定时中断函数
{
    if(1==vGu8TimeFlag_1&&vGu16TimeCnt_1>0) //在定时中断里衍生出“软件定时器 1”
    {
            vGu16TimeCnt_1--;
    }

    if(1==vGu8TimeFlag_2&&vGu16TimeCnt_2>0) //在定时中断里衍生出“软件定时器 2”
    {
            vGu16TimeCnt_2--;
    }

    if(1==vGu8TimeFlag_3&&vGu16TimeCnt_3>0) //在定时中断里衍生出“软件定时器 3”
    {
            vGu16TimeCnt_3--;
    }

    //按上面的套路继续写，可以衍生出 N 个“软件定时器”，只要不超过单片机的 RAM 和 ROM。
}
```

【87.2 练习例程。】

现在根据上述程序框架，编写 3 个 LED 灯闪烁的程序。第 1 个 LED 灯的一闪一灭的周期是 2 秒，第 2 个 LED 灯的一闪一灭的周期是 1 秒，第 3 个 LED 灯一闪一灭的周期是 0.5 秒。这 3 个灯的闪烁频率是不一样的，因此需要 3 个软件定时器。该例子其实也是一个多任务并行处理的典型例子，这 3 个 LED 灯就代表 3 个不同的任务，它们之间是通过 switch 这个关键语句进行多任务并行处理的。switch 的精髓在于根据某个特定条件切换到对应的步骤（或称“跳转到对应的步骤”）。

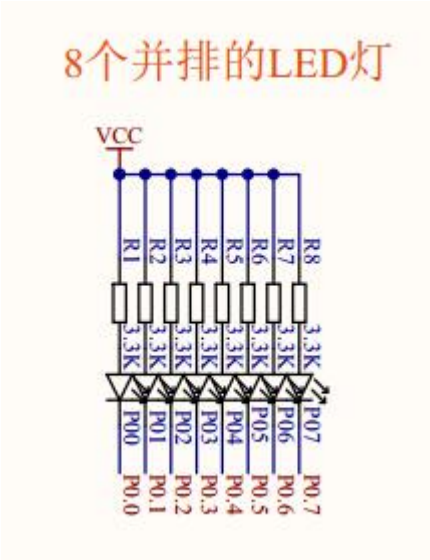

图 87.2.1　灌入式驱动 8 个 LED

```
#include "REG52.H"

#define  BLINK_TIME_1    1000    //时间是 1000ms
#define  BLINK_TIME_2    500     //时间是 500ms
#define  BLINK_TIME_3    250     //时间是 250ms

sbit P0_0=P0^0;
sbit P0_1=P0^1;
sbit P0_2=P0^2;

//“软件定时器 1”的相关变量
volatile unsigned char vGu8TimeFlag_1=0;
volatile unsigned int vGu16TimeCnt_1=0;

//“软件定时器 2”的相关变量
volatile unsigned char vGu8TimeFlag_2=0;
volatile unsigned int vGu16TimeCnt_2=0;

//“软件定时器 3”的相关变量
volatile unsigned char vGu8TimeFlag_3=0;
volatile unsigned int vGu16TimeCnt_3=0;

unsigned char Gu8Step_1=0;  //软件定时器 1 的 switch 切换步骤
unsigned char Gu8Step_2=0;  //软件定时器 2 的 switch 切换步骤
unsigned char Gu8Step_3=0;  //软件定时器 3 的 switch 切换步骤

void main()
```

```
{
    TMOD=0x01;  //设置定时器 0 为工作方式 1
    TH0=0xfc;   //产生 1ms 中断的 TH0 初始值
    TL0=0x66;   //产生 1ms 中断的 TL0 初始值
    EA=1;       //开总中断
    ET0=1;      //允许定时 0 的中断
    TR0=1;      //启动定时 0 的中断


    while(1)  //主循环
    {

      //软件定时器 1 控制的 LED 灯闪烁
       switch(Gu8Step_1)
       {
           case 0:
               if(0==vGu16TimeCnt_1)
               {
                    P0_0=0;
                    vGu8TimeFlag_1=0;
                    vGu16TimeCnt_1=BLINK_TIME_1;
                    vGu8TimeFlag_1=1;

                    Gu8Step_1=1;
                }
               break;

           case 1:

              if(0==vGu16TimeCnt_1)
               {
                    P0_0=1;
                    vGu8TimeFlag_1=0;
                    vGu16TimeCnt_1=BLINK_TIME_1;
                    vGu8TimeFlag_1=1;

                    Gu8Step_1=0;
                }
               break;
        }

      //软件定时器 2 控制的 LED 灯闪烁
       switch(Gu8Step_2)
       {
```

```
        case 0:
            if(0==vGu16TimeCnt_2)
            {
                P0_1=0;
                vGu8TimeFlag_2=0;
                vGu16TimeCnt_2=BLINK_TIME_2;
                vGu8TimeFlag_2=1;

                Gu8Step_2=1;
             }
            break;

        case 1:

           if(0==vGu16TimeCnt_2)
            {
                P0_1=1;
                vGu8TimeFlag_2=0;
                vGu16TimeCnt_2=BLINK_TIME_2;
                vGu8TimeFlag_2=1;

                Gu8Step_2=0;
             }
            break;
     }

    //软件定时器 3 控制的 LED 灯闪烁
     switch(Gu8Step_3)
     {
        case 0:
            if(0==vGu16TimeCnt_3)
            {
                P0_2=0;
                vGu8TimeFlag_3=0;
                vGu16TimeCnt_3=BLINK_TIME_3;
                vGu8TimeFlag_3=1;

                Gu8Step_3=1;
             }
            break;

        case 1:

           if(0==vGu16TimeCnt_3)
```

```
                {
                    P0_2=1;
                    vGu8TimeFlag_3=0;
                    vGu16TimeCnt_3=BLINK_TIME_3;
                    vGu8TimeFlag_3=1;

                    Gu8Step_3=0;
                 }
                break;
        }

   }
}


void T0_time() interrupt 1    //定时器 0 的中断函数，每 1ms 单片机自动执行一次此函数
{
    if(1==vGu8TimeFlag_1&&vGu16TimeCnt_1>0) //在定时中断里衍生出“软件定时器 1”
    {
            vGu16TimeCnt_1--;
    }

    if(1==vGu8TimeFlag_2&&vGu16TimeCnt_2>0) //在定时中断里衍生出“软件定时器 2”
    {
            vGu16TimeCnt_2--;
    }

    if(1==vGu8TimeFlag_3&&vGu16TimeCnt_3>0) //在定时中断里衍生出“软件定时器 3”
    {
            vGu16TimeCnt_3--;
    }


    TH0=0xfc;   //重装初值，不能忘
    TL0=0x66;   //重装初值，不能忘
}
```