

第八十八节： 两大核心框架理论（四区一线，switch 外加定时中断）。

【88.1 四区一线。】

提出“四区一线”理论，主要方便初学者理解单片机程序大概的“空间分区”。

“四区”代表四大主流函数，分别是：系统初始化函数，外设初始化函数，主程序的任务函数，定时中断函数。

“一线”是指“系统初始化函数”与“外设初始化函数”的“分割线”，这个“分割线”是一个 delay 的延时函数。

“四区一线”的布局如下：

```
void main()
{
    SystemInitial();           // “四区一线”的“第一区”
    Delay(10000);              // “四区一线”的“一线”
    PeripheralInitial();        // “四区一线”的“第二区”
    while(1) //主循环
    {
        LedService();          // “四区一线”的“第三区”
        KeyService();          // “四区一线”的“第三区”
        UsartService();        // “四区一线”的“第三区”
        ...                    //凡是在主循环里的函数都是属于“第三区”
    }
}

void TO_time() interrupt 1     // “四区一线”的“第四区”
{
}
```

“第一区”的函数 SystemInitial(), 是一个系统的初始化函数，专门用来初始化单片机自己的寄存器以及个别外围要求响应速度快的输出设备，防止刚上电之后，由于输出 I/O 口电平状态不确定而导致外围设备误动作，比如驱动继电器的误动作等等。

“一线”的函数 Delay(10000)，是一个延时函数，为什么这里要插入一个延时函数？主要目的是为接下来的 PeripheralInitial() 做准备的。上电后先延时一段时间，再执行 PeripheralInitial() 函数，因为 PeripheralInitial() 函数专门用来初始化不要求上电立即处理的外设芯片和模块。比如液晶模块，AT24C02 存储芯片，DS1302 时钟芯片，等等。这些芯片在上电的瞬间，内部自身的复位需要一点时间，以及外部电压稳定也需要一点时间，只有过了这一点时间，这些芯片才处于正常的工作状态，这个时候单片机才能跟它正常通信，所以“一线”函数 Delay(10000) 的意义就在这里。

“第二区”的函数 PeripheralInitial(), 是一个外设的初始化函数。专门用来初始化不要求上电立即处理的外设芯片和模块。

“第三区”的函数 LedService(), KeyService(), UsartService(), 等等，是一些在主循环里不断扫描的任务函数。

“第四区”的函数 void TO_time() interrupt 1, 是一个定时中断函数，一个系统必须标配一个定时中

断函数才算完美齐全，这个中断函数提供系统的节拍时间，以及处理扫描一些跟 I/O 口消抖动相关的函数，以及跟蜂鸣器驱动相关的函数。

【88.2 switch 外加定时中断。】

提出“switch 外加定时中断”理论，主要方便初学者理解单片机程序大概的“逻辑框架”。

switch 是一个万能语句，它外加 while 与 for 循环就可以做任何复杂的算法，比如，搜索算法，运动算法，提取关键词算法，等等。它外加定时中断，就可以搭建一个系统的基本框架。比如，做通信的程序框架，人机界面的程序框架，按键服务的程序框架，等等。switch 的精髓在于“根据条件进行步骤的灵活切换”。具体内容请看本节的练习程序。

【88.3 练习例程。】

根据上述的两大核心框架理论，编写 1 个 LED 灯闪烁的程序。

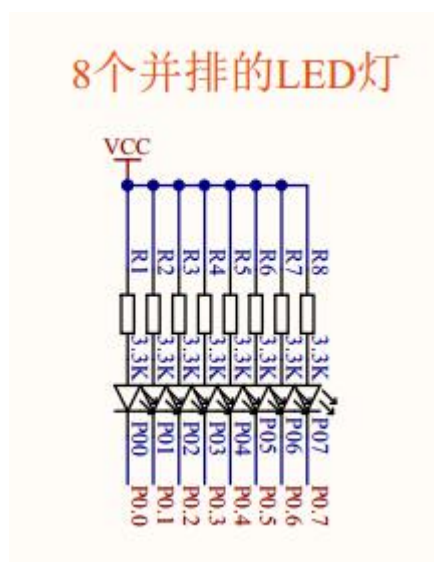


图 88.3.1 灌入式驱动 8 个 LED

```
#include "REG52.H"

void TO_time();
void SystemInitial(void) ;
void Delay(unsigned long u32DelayTime) ;
void PeripheralInitial(void) ;
void LedService(void);

#define BLINK_TIME_1 1000

sbit P0_0=P0^0;

volatile unsigned char vGu8TimeFlag_1=0;
```

```

volatile unsigned int vGul6TimeCnt_1=0;

void main()
{
    SystemInitial();          // “四区一线”的“第一区”
    Delay(10000);             // “四区一线”的“一线”
    PeripheralInitial();       // “四区一线”的“第二区”
    while(1) //主循环
    {
        LedService();         // “四区一线”的“第三区”
    }
}

void T0_time() interrupt 1    // “四区一线”的“第四区”
{
    if(1==vGu8TimeFlag_1&&vGul6TimeCnt_1>0)
    {
        vGul6TimeCnt_1--;
    }

    TH0=0xfc;
    TL0=0x66;
}

void SystemInitial(void)
{
    TMOD=0x01;
    TH0=0xfc;
    TL0=0x66;
    EA=1;
    ET0=1;
    TR0=1;
}

void Delay(unsigned long u32DelayTime)
{
    for(;u32DelayTime>0;u32DelayTime--);
}

void PeripheralInitial(void)
{
}

```

```

void LedService(void)
{
    static unsigned char Su8Step=0; //加 static 修饰的局部变量，每次进来都会保留上一次值。

    switch(Su8Step)
    {
        case 0:
            if(0==vGu16TimeCnt_1) //时间到
            {
                P0_0=0;
                vGu8TimeFlag_1=0;
                vGu16TimeCnt_1=BLINK_TIME_1; //重装定时的时间
                vGu8TimeFlag_1=1;

                Su8Step=1; //切换到下一个步骤，精髓语句！
            }
            break;

        case 1:

            if(0==vGu16TimeCnt_1) //时间到
            {
                P0_0=1;
                vGu8TimeFlag_1=0;
                vGu16TimeCnt_1=BLINK_TIME_1; //重装定时的时间
                vGu8TimeFlag_1=1;

                Su8Step=0; //返回到上一个步骤，精髓语句！
            }
            break;
    }
}

```