

## 第3章 外部中断实验

### 3.1 实验目的

1. 掌握 NVIC 中断优先级配置。
2. 学会外部中断配置。

### 3.2 实验要求

学习和掌握中断的概念，设计并实现外部中断功能。参考 3.5 编程提示中的示例 1

### 3.3 实验任务

#### 1、基础部分

用 C 语言编程实现以下功能，系统上电时执行主程序，程序可采用第一次实验的 LED 流水灯程序；使用按键做外部中断源输入，当按键有按下时触发中断，中断程序可以控制交通灯的亮灭，亮灭方式自定；

#### 2、提高部分

用 C 语言编程实现系统上电时执行主程序，程序实现用 1 位数码管显示 0-9 数字；使用 2 个按键做外部中断源输入，当按键有按下时触发中断，考虑中断的优先级问题，自拟中断服务程序，实现并演示；

### 3.4 实验原理

电路结构如图 3.1 所示

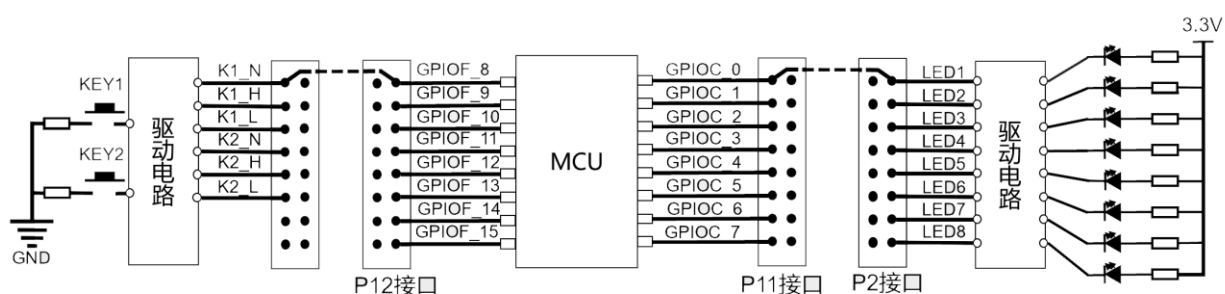


图 3.1 电路结构图

#### 1. NVIC 中断优先级

NVIC 是嵌套向量中断控制器，控制着整个芯片中断相关的功能，它跟内核

紧密耦合，是内核里面的一个外设。但是各个芯片厂商在设计芯片的时候会对 Cortex-M4 内核里面的 NVIC 进行裁剪，把不需要的部分去掉，所以说 STM32 的 NVIC 是 Cortex-M4 的 NVIC 的一个子集。

CM4 内核可以支持 256 个中断，包括 16 个内核中断和 240 个外部中断，256 级的可编程中断设置。对于 STM32F4 没有用到 CM4 内核的所有东西，只是用到了一部分，对于 STM32F40 和 41 系列共有 92 个中断，其中有 10 个内核中断和 82 个可屏蔽中断，常用的为 82 个可屏蔽中断。

ISER[8]—中断使能寄存器组，用来使能中断，每一位控制一个中断，由于上面已经说明了控制 82 个可屏蔽的中断，因此利用 ISER[0~2]这三个 32 位寄存器就够了。一下的几个寄存器同理。

ICER[8]—中断除能寄存器组，用来消除中断。

ISPR[8]—中断挂起控制寄存器组，用来挂起中断。

ICPR[8]—中断解挂控制寄存器组，用来解除挂起。

IABR[8]—中断激活标志寄存器组，对应位如果为 1 则表示中断正在被执行。

IP[240]—中断优先级控制寄存器组，它是用来设置中断优先级的。我们只用到 IP[0]~IP[81]，每个寄存器只用到了高 4 位，这 4 位又用来设置抢占优先级和响应优先级（有关抢占优先级和响应优先级后面会介绍到），而对于抢占优先级和响应优先级各占多少位则由 AIRCR 寄存器控制，相关设置如表 3.1 所示。

表 3.1 中断分组

组	AIRCR[10:8]	bit[7:4]分配情况	分配结果
0	111	0: 4	0 位抢占优先级，4 位响应优先级
1	110	1: 3	1 位抢占优先级，3 位响应优先级
2	101	2: 2	2 位抢占优先级，2 位响应优先级
3	100	3: 1	3 位抢占优先级，1 位响应优先级
4	011	4: 0	4 位抢占优先级，0 位响应优先级

关于抢占优先级和响应优先级的理解，可以将它们简单的理解为两个级别，抢占优先级的级别要比响应优先级的级别高，简单的理解为一个为长辈的一个为晚辈的，晚辈要让着长辈，因此抢占优先级的中断可以打断响应优先级的中断，而同级别的中断就得有个先来后到的了，先来的先执行。抢占优先级

与相应优先级示例，如表 3.2 所示。

表 3.2 抢占优先级与响应优先级示例

中断向量	抢占优先级	响应优先级	
A	0	1	抢占优先级相同，响应优先级数值小的优先级高
B	0	2	
A	1	2	响应优先级相同，抢占优先级数值小的优先级高
B	0	2	
A	1	0	抢占优先级比响应优先级高
B	0	2	
A	1	1	抢占优先级和响应优先级均相同，则中断向量编号小的先执行
B	1	1	

2. 外部中断

外部中断/事件控制器(EXTI)管理了控制器的 23 个中断/事件线。每个中断/事件线都对应有一个边沿检测器，可以实现输入信号的上升沿检测和下降沿的检测。EXTI 可以实现对每个中断/事件线进行单独配置，可以单独配置为中断或者事件，以及触发事件的属性。

EXTI 功能框图如图 3.2 所示。

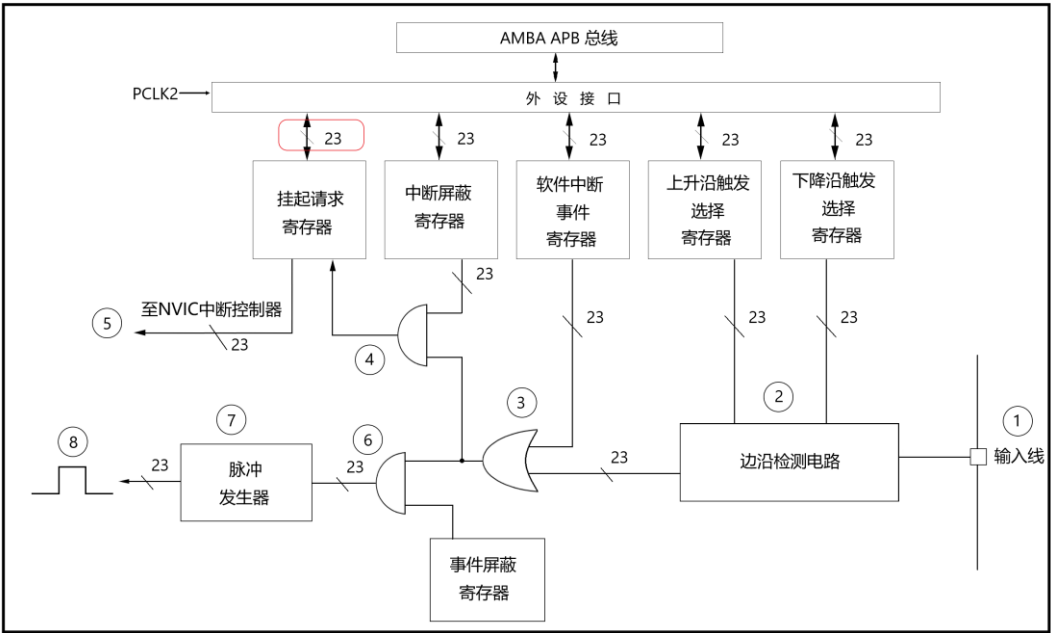


图 3.2 EXTI 功能框图

在图 3.2 中在信号线上打一个斜杠并标注“23”字样表示在控制器内部类

---

似的信号线路有 23 个，这与 EXTI 总共有 23 个中断/事件线是吻合的。

EXTI 可分为两大部分功能，一个是产生中断，另一个是产生事件。图 3.2 中线路①②③④⑤指示的电路流程是一个产生中断的线路，最终信号流入到 NVIC 控制器内。

编号 1 是输入线，EXTI 控制器有 23 个中断/事件输入线，这些输入线可以通过寄存器设置为任意一个 GPIO，也可以是一些外设的事件，输入线一般是存在电平变化的信号。

编号 2 是一个边沿检测电路，它会根据上升沿触发选择寄存器 (EXTI\_RTSR) 和下降沿触发选择寄存器 (EXTI\_FTSR) 对应位的设置来控制信号触发。边沿检测电路以输入线作为信号输入端，如果检测到有边沿跳变就输出有效信号 1 给编号 3 电路，否则输出无效信号 0。而 EXTI\_RTSR 和 EXTI\_FTSR 两个寄存器可以控制需要检测哪些类型的电平跳变过程，可以是只有上升沿触发、只有下降沿触发或者上升沿和下降沿都触发。

编号 3 电路实际就是一个或门电路，它一个输入来自编号 2 电路，另外一个输入来自软件中断事件寄存器 (EXTI\_SWIER)。EXTI\_SWIER 允许我们通程序控制就可以启动中断/事件线，这在某些地方非常有用。

编号 4 电路是一个与门电路，它一个输入编号 3 电路，另外一个输入来自中断屏蔽寄存器 (EXTI\_IMR)。与门电路要求输入都为 1 才输出 1，导致的结果如果 EXTI\_IMR 设置为 0 时，那不管编号 3 电路的输出信号是 1 还是 0，最终编号 4 电路输出的信号都为 0；如果 EXTI\_IMR 设置为 1 时，最终编号 4 电路输出的信号才由编号 3 电路的输出信号决定，这样我们可以简单的控制 EXTI\_IMR 来实现是否产生中断的目的。编号 4 电路输出的信号会被保存到挂起寄存器 (EXTI\_PR) 内，如果确定编号 4 电路输出为 1 就会把 EXTI\_PR 对应位置 1。

编号 5 是将 EXTI\_PR 寄存器内容输出到 NVIC 内，从而实现系统中断事件控制。它是一个产生事件的线路，最终输出一个脉冲信号。

编号 6 电路是一个与门，它一个输入编号 3 电路，另外一个输入来自事件屏蔽寄存器 (EXTI\_EMR)。如果 EXTI\_EMR 设置为 0 时，那不管编号 3 电路的输出信号是 1 还是 0，最终编号 6 电路输出的信号都为 0；如果 EXTI\_EMR 设置为 1 时，最终编号 6 电路输出的信号才由编号 3 电路的输出信号决定，这样我们可以简单的控制、EXTI\_EMR 来实现是否产生事件的目的。

---

编号 7 是一个脉冲发生器电路，当它的输入端，即编号 6 电路的输出端，是一个有效信号 1 时就会产生一个脉冲；如果输入端是无效信号就不会输出脉冲。

编号 8 是一个脉冲信号，就是产生事件的线路最终的产物，这个脉冲信号可以给其他外设电路使用，比如定时器 TIM、模拟数字转换器 ADC 等等。

产生中断线路目的是把输入信号输入到 NVIC，进一步会运行中断服务函数，实现功能，这样是软件级的。而产生事件线路目的就是传输一个脉冲信号给其他外设使用，并且是电路级别的信号传输，属于硬件级的。

EXTI 有 23 个中断/事件线，每个 GPIO 都可以被设置为外部中断的中断输入口，这点也是 STM32F4 的强大之处。STM32F407 的中断控制器支持 23 个外部中断/事件请求。每个中断设有状态位，每个中断/事件都有独立的触发和屏蔽设置。STM32F407 的 23 个外部中断为：

EXTI 线 0~15：对应外部 IO 口的输入中断。

EXTI 线 16：连接到 PVD 输出。

EXTI 线 17：连接到 RTC 闹钟事件。

EXTI 线 18：连接到 USB OTG FS 唤醒事件。

EXTI 线 19：连接到以太网唤醒事件。

EXTI 线 20：连接到 USB OTG HS (在 FS 中配置)唤醒事件。

EXTI 线 21：连接到 RTC 入侵和时间戳事件。

EXTI 线 22：连接到 RTC 唤醒事件。

STM32F4 供 IO 口使用的中断线只有 16 个，但是 STM32F4 的 IO 口却远远不止 16 个，那么 STM32F4 是怎么把 16 个中断线和 IO 口一一对应起来的呢？于是 STM32 就这样设计，GPIO 的管脚 GPIOx.0~GPIOx.15 (x=A, B, C, D, E, F, G, H, I) 分别对应中断线 0~15。这样每个中断线对应了最多 9 个 IO 口，以线 0 为例：它对应了 GPIOA.0、GPIOB.0、GPIOC.0、GPIOD.0、GPIOE.0、GPIOF.0、GPIOG.0、GPIOH.0、GPIOI.0。而中断线每次只能连接到 1 个 IO 口上，这样就需要通过配置来决定对应的中断线配置到哪个 GPIO 上了。GPIO 跟中断线的映射关系如图 3.3 所示。

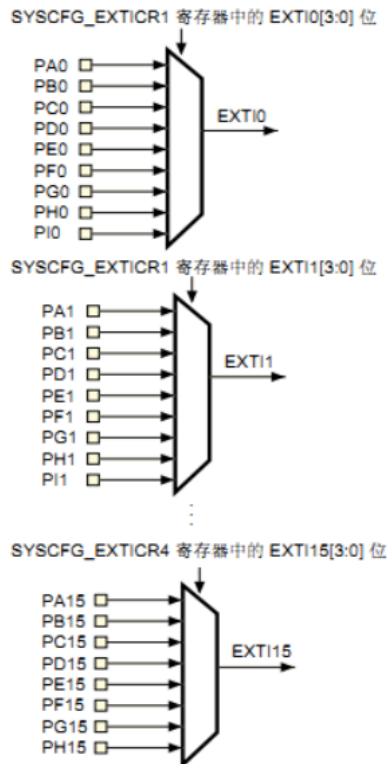


图 3.3 GPIO 中断线映射

## 3.5 编程提示

### 3.5.1 示例 1（注：源程序可在 FTP 服务器下载）

1、**程序功能：**使用按键做外部中断输入，中断控制 LED1 亮灭状态转换。

系统上电时，LED1 状态为点亮，当按键有按下时，LED1 的亮灭状态切换；

2、**参考流程图，**如图 3.4 所示：

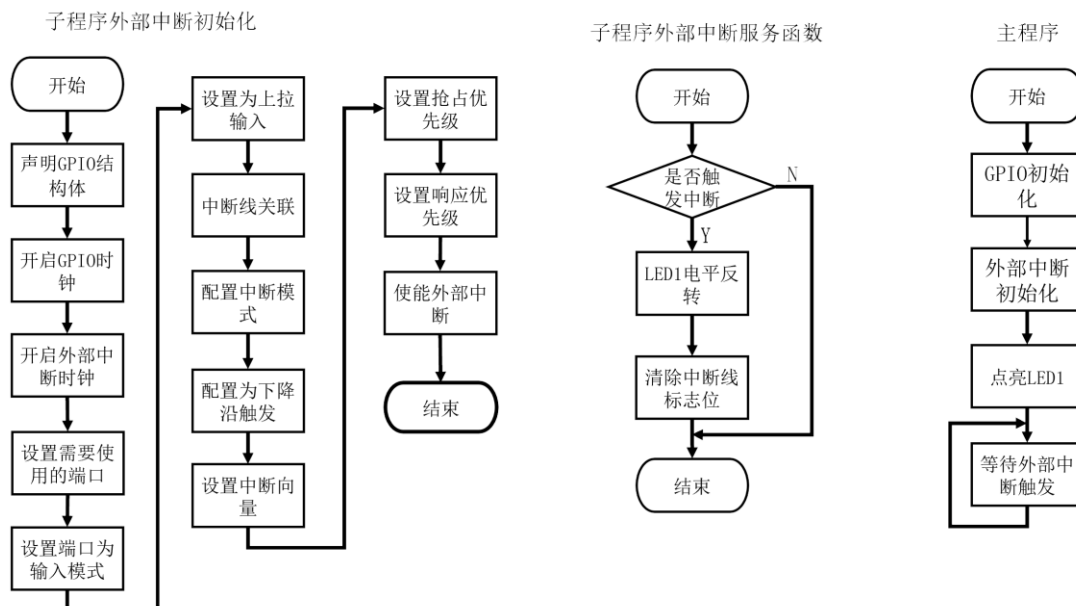


图 3.4 程序流程图

### 3、参考代码如下：

#### C 语言参考代码

```
/******  
功 能：外部中断服务函数  
参 数：无  
返回值：无  
*****/  
void EXTI9_5_IRQHandler(void)  
{  
    if(EXTI_GetITStatus(EXTI_Line8))  
    {  
        if(GPIO_ReadInputDataBit(GPIOF, GPIO_Pin_8) == 0)  
        {  
            LED1(!LED1_DATA());  
            EXTI_ClearITPendingBit (EXTI_Line8);  
        }  
    }  
}  
  
/******  
功 能：外部中断配置  
参 数：无  
返回值：无  
*****/  
void EXTI_Configure(void)  
{  
    GPIO_InitTypeDef GPIO_TypeDefStructure;  
    EXTI_InitTypeDef EXTI_TypeDefStructure;  
    NVIC_InitTypeDef NVIC_TypeDefStructure;  
  
    //开启中断输入端口时钟  
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOF, ENABLE);  
  
    //开启外部中断时钟  
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);  
  
    //KEY_UP for EXTI in Pin  
    GPIO_TypeDefStructure.GPIO_Pin = GPIO_Pin_8;  
    GPIO_TypeDefStructure.GPIO_Mode = GPIO_Mode_IN;           //通用输入模式  
    GPIO_TypeDefStructure.GPIO_PuPd = GPIO_PuPd_UP;          //上拉  
    GPIO_Init(GPIOF, &GPIO_TypeDefStructure);  
  
    //中断线关联  
    SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOF, EXTI_PinSource8);  
  
    EXTI_TypeDefStructure.EXTI_Line = EXTI_Line8;  
    EXTI_TypeDefStructure.EXTI_Mode = EXTI_Mode_Interrupt;  
    EXTI_TypeDefStructure.EXTI_Trigger = EXTI_Trigger_Falling; //下降沿触发  
    EXTI_TypeDefStructure.EXTI_LineCmd = ENABLE;  
    EXTI_Init(&EXTI_TypeDefStructure);  
  
    //EXTI2_IRQn中断向量优先级设置  
    NVIC_TypeDefStructure.NVIC_IRQChannel = EXTI9_5_IRQn;  
    NVIC_TypeDefStructure.NVIC_IRQChannelPreemptionPriority = 0;  
    NVIC_TypeDefStructure.NVIC_IRQChannelSubPriority = 7;  
    NVIC_TypeDefStructure.NVIC_IRQChannelCmd = ENABLE;  
    NVIC_Init(&NVIC_TypeDefStructure);  
}
```

### 4、源程序（可在 FTP 服务器下载）

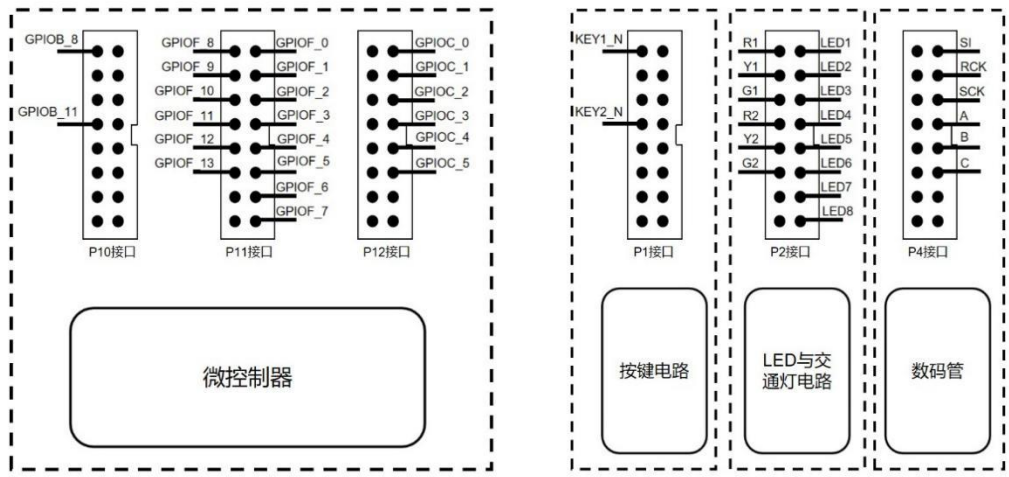
3.5.2 示例 2（注：源程序可在 FTP 服务器下载）

1、程序功能：主程序数码管循环显示数字 0~7。

按键查询：当数码管显示为 5 时，同时按下按键 2，则数码管停止计时同时 LED 呈现流水灯现象，大约 5 秒后流水灯停止，数码管恢复计时；当数码管当前显示不是 5 时，按下按键 2 则没有任何反应。

中断查询：不管数码管当前显示的任何数字，按下按键 1 则触发外部中断，数码管停止计时同时交通灯闪烁，大约 5 秒后交通灯闪烁停止，数码管恢复计时。

2、电路框图



接线：GPIOF\_0~GPIOF\_7 (P12 接口)      接      LED1~LED8 (P2 接口)

P11 接口-----P4 接口

PC0	接	SI
PC1	接	RCK
PC2	接	SCK
PC3	接	A
PC4	接	B
PC5	接	C

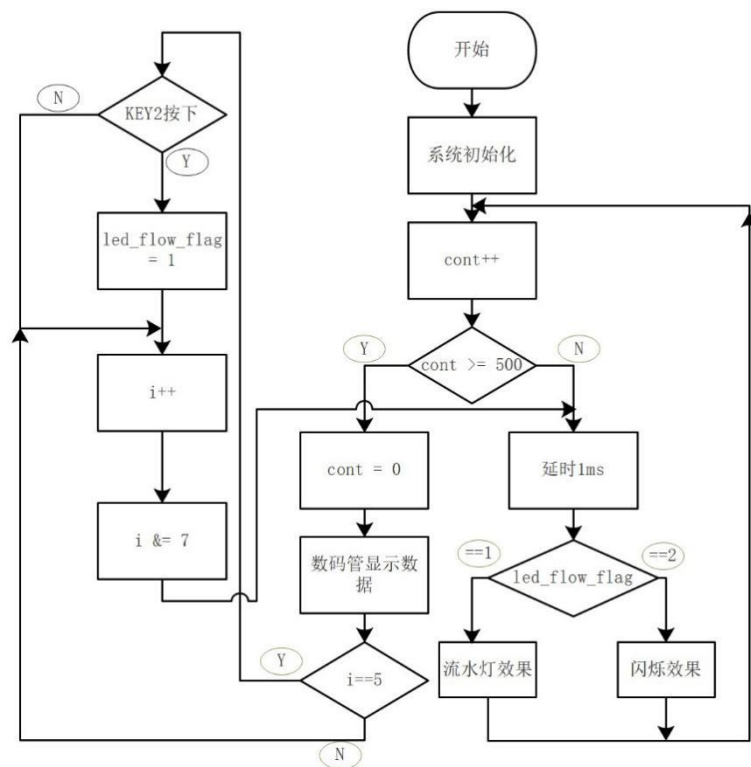
P10 接口-----P1 接口

PB8	接	KEY1_N
PB11	接	KEY2_N

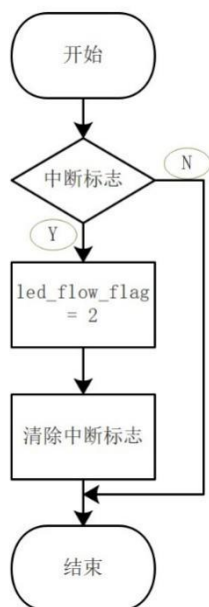


### 3、程序流程图

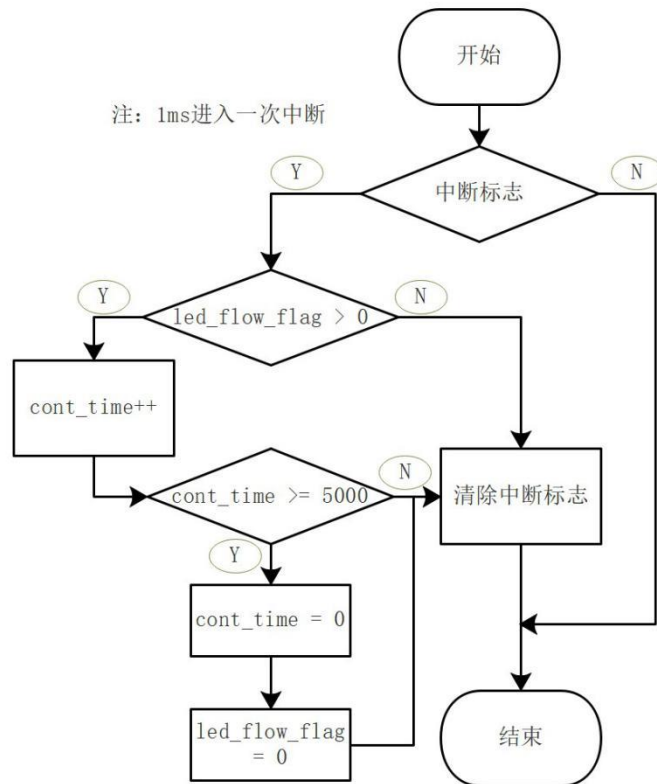
主函数



外部中断服务函数



### 定时器中断服务函数



4、源程序（可在 FTP 服务器下载）

### 3.6 实验报告

1. 画出程序流程图，打印程序清单，并加注释。
2. 实验结果讨论。
3. 实验心得与收获建议。

### 3.7 思考题

考虑如何使用电平触发方式触发外部中断。