

## 实验三 外部中断

### 一、实验目的：

- 1、了解 ARM 处理器中的的外部输入中断；
- 2、熟悉外部中断的工作原理；
- 3、掌握外部中断的程序设计。

### 二、实验环境：

- 1、proteus 仿真软件
- 2、VScode 开发环境
- 3、ARM gun 交叉编译工具链

### 三、实验内容

- 1、采用中断的方式获取 key1 的状态，如果按下，控制 LED1 和 LED2 亮，获取 key2 的状态，如果按下，控制 LED3 和 LED4 亮，获取 key3 的状态，如果按下，控制 LED1~LED4 熄灭。
- 2、创建 key\_int.h 和 key\_int.c，将按键中断操作封装成函数。

### 四、实验步骤

- 1、查看电路原理图，分析控制 KEY 所对应的 GPIO 引脚；
- 2、确定需要使用引脚对应的相关寄存器位；
- 3、中断配置步骤；
  - 1) 初始化 IO 口为输入(可直接使用实验二中的 KEY 的初始化)。
  - 2) 开启 SYSCFG 时钟 (APB2 中 bit14)，设置 IO 口与中断线的映射关系。(8.2.4 节)
  - 3) 开启与该 IO 口相对的线上中断 (EXTI->IMR)，设置触发条件 EXTI->RTSR 或 EXTI->FTSR。
  - 4) 配置中断控制器 (NVIC)，使能中断、分组并设置优先级。
  - 5) 编写中断服务函数，处理完成后清中断 (EXTI->PR)。
- 4、完成实验报告

### 五、进阶

自行设计按键中断处理函数的功能。

## 附录

ISER[8]：ISER 全称是：Interrupt Set-Enable Registers，这是一个中断使能寄存器组。M4 内核支持 256 个中断，用 8 个 32 位寄存器来控制，每个位控制一个中断。但是 STM32F4 的可屏蔽中断最多只有 82 个，所以有用的就是三个 (ISER[0~2])，总共可以表示 96 个中断。而 STM32F4 只用了其中的前 82 个。ISER[0]的 bit0~31 分别对应中断 0~31；ISER[1]的 bit0~32 对应中断 32~63；ISER[2]的 bit0~17 对应中断 64~81；这样总共 82 个中断就分别对应上了。要使能某个中断，必须设置相应的 ISER 位为 1，使该中断被使能(这里仅仅是使能，还要配合中断分组、屏蔽、IO 口映射等设置才算是一个完整的中断设置)。

IP[240]：全称是：Interrupt Priority Registers，是一个中断优先级控制的寄存器组。这个寄存器组相当重要！STM32F4 的中断分组与这个寄存器组密切相关。IP 寄存器组由 240 个 8bit 的寄存器组成，每个可屏蔽中断占用 8bit，这样总共可以表示 240 个可屏蔽中断。而 STM32F4 只用到了其中的 82 个。IP[81]~IP[0]分别对应中断 81~0。而每个可屏蔽中断占用的 8bit 并没有全部使用，而是只用了高 4 位。这 4 位，又分为抢占优先级和子优先级。抢占优先级在前，子优先级在后。而这两个优先级各占几个位又要根据 SCB->AIRCRR 中的中

断分组设置来决定。这里简单介绍一下 STM32F4 的中断分组：STM32F4 将中断分为 5 个组，组 0~4。该分组的设置是由 SCB->AICR 寄存器的 bit10~8 来定义的。具体的分配关系如下表所示。

组	AICR[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

通过这个表，我们就可以清楚的看到组 0~4 对应的配置关系，例如组设置为 3，那么此时所有的 82 个中断，每个中断的中断优先寄存器的高四位中的最高 3 位是抢占优先级，低 1 位是响应优先级。每个中断，你可以设置抢占优先级为 0~7，响应优先级为 1 或 0。抢占优先级的级别高于响应优先级。而数值越小所代表的优先级就越高。