

浙江大学实验报告

专业：__自动化__

姓名：__边玥心__

学号：__3170103500__

实验日期：__19.10.11__

地点：__

课程名称：__嵌入式系统__ 指导老师：__陆玲霞 于淼__ 成绩：__

实验名称：__外部中断__

一、 实验目的和要求（必填）

- 1、 掌握 STM32F1 的 IO 口最基本的操作，掌握 STM32F1 的 IO 口作为外部中断输入，以中断的方式实现外部中断设计。
- 2、 掌握硬件设计，软件设计和中断服务子程序添加
- 3、 熟练学习初始化图形设置，初始化和编写中断服务函数

二、 实验内容和原理（必填）

实验内容：

使用图形界面工具 CUBEMX 设置 STM32 的外部中断设计

在配置每个中断的时候一般有 3 个编程要点：

- 1、 使能外设某个中断，这个具体由外设的相关中断使能位控制，比如串口有发送完成中断，接受完成中断。这两个中断都由串口控制寄存器的相关中断使能位控制。
- 2、 在 HAL 库，没有了 NVIC 以及 EXTI 的初始化结构体函数，而是将 EXTI 的初始化设置归纳在 GPIO_InitTypeDef 结构体中，在该结构体的 Mode 中，可设置某 GPIO 为中断或者事件，中断（IT）与事件（EVT）都各有三种可选的触发模式。HAL_GPIO_Init 函数起到了引脚参数设置以及中断/事件的基层配置，用户使用后需要进行 NVIC 的对应中断优先级设置以及使能。调用设置 NVIC 优先级函数 HAL_NVIC_SetPriority 可以对对应 IRQn 中断的主优先级 PreemptPriority 以及子优先级 SubPriority 进行配置，主优先级与子优先级可设置范围由优先级组决定，HAL_Init 初始化函数后，则已经设置好优先级组 Group4；调用 NVIC 使能函数 HAL_NVIC_EnableIRQ 可以 IRQn 进行使能。
- 3、 编写中断服务函数
在启动文件 startup_stm32f10x.s 中预先为每个中断都写了一个中断服务函数，只是这些中断函数都是为空，为的只是初始化中断向量表。实际的中断服务函数都需要我们重新编写，关于中断服务函数的函数名必须跟启动文件里面预先设置的一样，如果写错，系统就在中断向量表找不到中断服务函数的入口，直接跳转到启动文件里面预先写好的空函数，并且在里面无限循环，实现不了中断。

实验原理：

STM32F103 在内核水平上搭载了一个中断响应系统，支持为数众多的系统内部中断和外部中断。

STM32 有 84 个中断，包括 16 个内核中断和 68 个可屏蔽中断，具有 16 级可编程的中断优先级。而 STM32F103 系列上面，16 个内核中断异常不变，而可屏蔽中断只有 60。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。

在固件库中，NVIC 的结构定义为每个寄存器都预留了很多位，在配置中断的时候一般值用

ISER,ICER,IP 这三个寄存器，ISER 用来使能中断，ICER 用来失能中断，IP 用来设置中断优先级。

在 NVIC 有一个专门的寄存器：中断优先级寄存器 NVIC_IPRx，用来配置外部中断的优先级，IPR 宽度为 8bit，原则上每个外部中断可配置的优先级为 0-255，数值越小，优先级越高。用来表达优先级的常使用 4bit，在本次实验中由于只用一个外部中断，所以不需要用到优先级的设置。

老师重点介绍了 EXTI 功能框图，EXTI 的功能框图包含了 EXTI 最核心内容，掌握了功能框图，对 EXTI 就有一个整体的把握，在编程时思路就非常清晰。

很多信号线上有斜杠并标注 20 字样，这个表示在控制器内部类似的信号线路有 20 个，这与 EXTI 总共有 20 个中断线是吻合的，只要明白其中一个的原理，那其他 19 个线路原理也就知道。

EXTI 分为两个部分功能，一个是产生中断，另一个是产生事件，这两个功能从硬件上就有所不同。

产生中断的线路，最终信号流入到 NVIC 控制器内，输入线可以通过寄存器设置为任意一个 GPIO 或外设事件，输入线一般是存在电平变化的信号。

在有边沿检测电路，根据上升沿触发选择寄存器和下降沿触发选择寄存器对应位来控制信号触发，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。

编号 3 电路实际是或门电路，一个来自编号 2 电路，另外一个输入来自软件中断事件寄存器

EXTI_SWIER，允许我们通程序控制就可以启动中断/事件线。编号 4 是与门电路，用中断屏蔽寄存器可以简单控制来实现是否产生中断的目的。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。

而产生事件的线路，最终输出一个脉冲信号，产生事件线路是在编号 3 电路之后有所不同，编号 6 是与门，另一个来自事件屏蔽寄存器，通过控制这个寄存器是 0 还是 1，可以实现是否产生事件的目的。编号 7 是一个脉冲发生器电路，当它的输入端是一个有效信号 1 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，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，这样的脉冲信号一般用来触发 TIM 或者 ADC 开始转换。

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。

STM32CubeMX 是意法半导体全新设计的一个自动初始化代码生成软件，基于 ST 最新推出的 HAL 库，原来的 STD 库已不再更新；对于一般的项目应用，可以直接使用该软件快速地生成相应的工程包，进行快速应用开发。对于初学者，该软件可帮助快速搭建 STM32 工程，以图形化直观的方法配置外设，减小开发者负担，功能相当强大，目前已支持 STM32 全系列，并且新的 F7,L4 系列只支持 HAL 库，这是 ST 今后推行的开发方式。

三、 主要仪器设备（必填）

1. 计算机
2. 计算机的应用软件：KEIL5 CUBEMX

四、 操作方法和实验步骤

1. 按照老师给的指导书查看任务详情和 STM32 外部中断知识
2. 了解实验原理和实验过程，了解借助图形界面工具进行设置
3. 编写代码，进行错误调试，完成 STM32 外部中断设计。

五、 实验结果与分析（必填）

5.1 操作

- 1、打开 STM32CubeMX 图形化工具，新建项目
- 2、选择 MCU 匹配型号
- 3、进行初始设置，如管脚 PE3 选择 GPIO_OUTPUT 形式；PA0 设置为 GPIO_EXTI0 功能等
- 3、在结构模块中分别设置 GPIO 和 NVIC 模块，在这些模块中配置一定的模式保证设计能够成功实现
注意：需要在 NVIC 模块中手动在 EXIT LINE0 INTERRUPT 对应的 Enabled 位置打上勾，方便中断使能
- 4、设置完毕后，可以自动生成代码，Generat Code
- 5、进入 MDK 环境后，找到相应文件，并找到中断子程序 void EXTI0_IRQHandler(void)，在该函数添加相应代码，实现实验效果。

5.2 代码与注释

这里引入中断子程序相应代码：

```
void EXTI0_IRQHandler(void)
{
    /* USER CODE BEGIN EXTI0_IRQn 0 */

    HAL_GPIO_TogglePin(GPIOE,GPIO_PIN_3);    //使该管脚 1 置 0，0 置 1 实现反置；赶下终端触发按钮即可灯亮/灭设定
    HAL_GPIO_TogglePin(GPIOB,GPIO_PIN_8);    //使该管脚 1 置 0，0 置 1 实现反置；赶下终端触发按钮即可使蜂鸣器响/闭
    //HAL_Delay(500);                          //本来想作延时的，但发现在中断程序使用延时会导致无法产生相应终端效果，从而也有了经验
    /* USER CODE END EXTI0_IRQn 0 */
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
    /* USER CODE BEGIN EXTI0_IRQn 1 */

    /* USER CODE END EXTI0_IRQn 1 */
}
```

最终，顺利实现实验任务要求，按下起动外部中断的按键，蜂鸣器响起来，LED 灯亮起来；再按下按键，进入中断，蜂鸣器不响，LED 灯灭。
外部中断设计实现。

六、思考与实验反思

- 1、实验要求比较简单，在进行一定的预习基础上能够了解外部中断的设置和初始化。因为本次实验是借助图形化编程工具，在进行初始化设置后可以 generate code，这相比之前的实验更简单。
- 2、在进入中断子函数编写，不需要写延时函数，因为 CPU 处理延迟的问题，最终不能产生需要的实验效果。所以子函数编写代码的 HAL_Delay 函数最终被除掉了。
- 3、本次实验让我们熟悉了解图形界面工具编程的流程，相比于第一次实验，更加简单快捷，但也要了解中间的硬件设计，才能在设置上设置好让之后的实验顺利进行。