

外部中断实验

中断的概念大家应该都了解的差不多了，所以我就略过介绍中断这一环节。**外部中断（EXTI）是STM32众多中断中的一种。STM32一共有16组外部中断，分别对应GPIOx.1~GPIOx.15(x=A,B,C,...,H,I),也就是说序号相同的IO口触发的是同一个外部中断，但是外部中断无法判断是由哪个GPIO外设所触发的，所以STM32最多可以监听16个外部中断，且触发这些外部中断的IO口序号不能相同。**

接下来我们来看看使用 HAL 库配置外部中断的一般步骤。

1.是能IO口时钟。

首先，我们要使用 IO 口作为中断输入，所以我们要使能相应的 IO 口时钟。

2.设置 IO 口模式，触发条件。

当我们使用 HAL 库的时候，则都是在函数 HAL_GPIO_Init 中一次性完成的。例如我们要设置 PA0 链接中断线 0，并且为上升沿触发，代码为：

```
1  GPIO_InitTypeDef GPIO_InitStructure;  
2  GPIO_InitStructure.Pin=GPIO_PIN_0;           //PA0  
3  GPIO_InitStructure.Mode=GPIO_MODE_IT_RISING; //外部中断，上升沿触发  
4  GPIO_InitStructure.Pull=GPIO_PULLDOWN;      //默认下拉  
5  HAL_GPIO_Init(GPIOA,&GPIO_InitStructure);
```

当我们调用 HAL_GPIO_Init 设置 IO 的 Mode 值为 GPIO_MODE_IT_RISING（外部中断上升沿触发），GPIO_MODE_IT_FALLING（外部中断下降沿触发）或者GPIO_MODE_IT_RISING_FALLING（外部中断双边沿触发）的时候，该函数内部会通过判断 Mode 的值来开启 SYSCFG 时钟，并且设置 IO 口和中断线的映射关系。

因为我们这里初始化的是 PA0，调用该函数后中断线 0 会自动连接到 PA0。如果某个时间，我们又同样的方式初始化了 PB0，那么 PA0 与中断线的链接将被清除，而直接链接 PB0 到中断线 0。

3.配置中断优先级，并使能中断。

我们设置好中断线和 GPIO 映射关系，然后又设置好了中断的触发模式等初始化参数。既然是外部中断，**涉及到中断我们当然还要设置 NVIC 中断优先级**。这个在前面已经讲解过，这里我们就接着上面的范例，设置中断线 0 的中断优先级并使能外部中断 0 的方法为：

```
1  HAL_NVIC_SetPriority(EXTIO_IRQn,2,1); //抢占优先级为 2，子优先级为 1  
2  HAL_NVIC_EnableIRQ(EXTIO_IRQn);     //使能中断线 2
```

4.编写中断服务函数。

中断服务函数的名字是在 HAL 库中事先有定义的。这里需要说明一下，STM32F4 的 IO 口外部中断函数只有 7 个，分别为：

```
1 void EXTI0_IRQHandler();
2 void EXTI1_IRQHandler();
3 void EXTI2_IRQHandler();
4 void EXTI3_IRQHandler();
5 void EXTI4_IRQHandler();
6 void EXTI9_5_IRQHandler();
7 void EXTI15_10_IRQHandler();
```

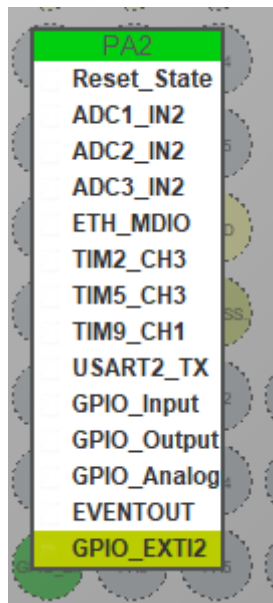
5) 编写中断处理回调函数 HAL_GPIO_EXTI_Callback

HAL 库为了用户使用方便，它提供了一个中断通用入口函数 HAL_GPIO_EXTI_IRQHandler，在该函数内部直接调用回调函数 HAL_GPIO_EXTI_Callback.用户在使用时可以重新定义HAL_GPIO_EXTI_Callback函数来实现用户需要的外部中断逻辑。

下面用Cube来实现外部中断使得PF14每两秒闪烁一次。

Step1.创建工程，配置时钟略。

Step2.设置PA2为外部中断IO口，如图所示设置。并配置PA2为上升沿触发模式，并默认下拉。记得使能外部中断。



PA2	n/a	n/a	External I...	Pull-down	n/a		✓
PA3	n/a	Low	Output Pu...	Pull-down	Low		✓
PF14	n/a	Low	Output Pu...	Pull-up	Low		✓

PA2 Configuration :

GPIO mode

External Interrupt Mode with Rising edge trigger detection

GPIO Pull-up/Pull...

Pull-down

User Label

EXTI line2 interrupt	✓	0	0
----------------------	---	---	---

再开一个PA3，连接PA2、PA3，通过翻转PA3的电平，来实现外部中断。

Step3.生成keil文件。

Step4.在主函数中添加两行语句。

```

/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_3);
    HAL_Delay(1000);
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */

```

Step5.在main.c中重新定义外部中断回调函数，在中断逻辑里实现PF14的翻转。

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    //test++;
    HAL_GPIO_TogglePin(GPIOF,GPIO_PIN_14);
}

```

教程结束。