第5章

STM32 中断系统原理与项目实践

本章导读

本章以榜样故事——焊接顾问艾爱国的介绍开始，然后介绍中断概述，分析了 NVIC 嵌套向量中断控制器、外部中断/事件控制器（EXTI）、中断相关库函数基本原理，还介绍了外部中断的一般配置步骤，最后通过基本项目实践、拓展项目实践的训练，实现素质、知识、能力目标的融合达成。本章素质、知识、能力结构图如图 5-1 所示。

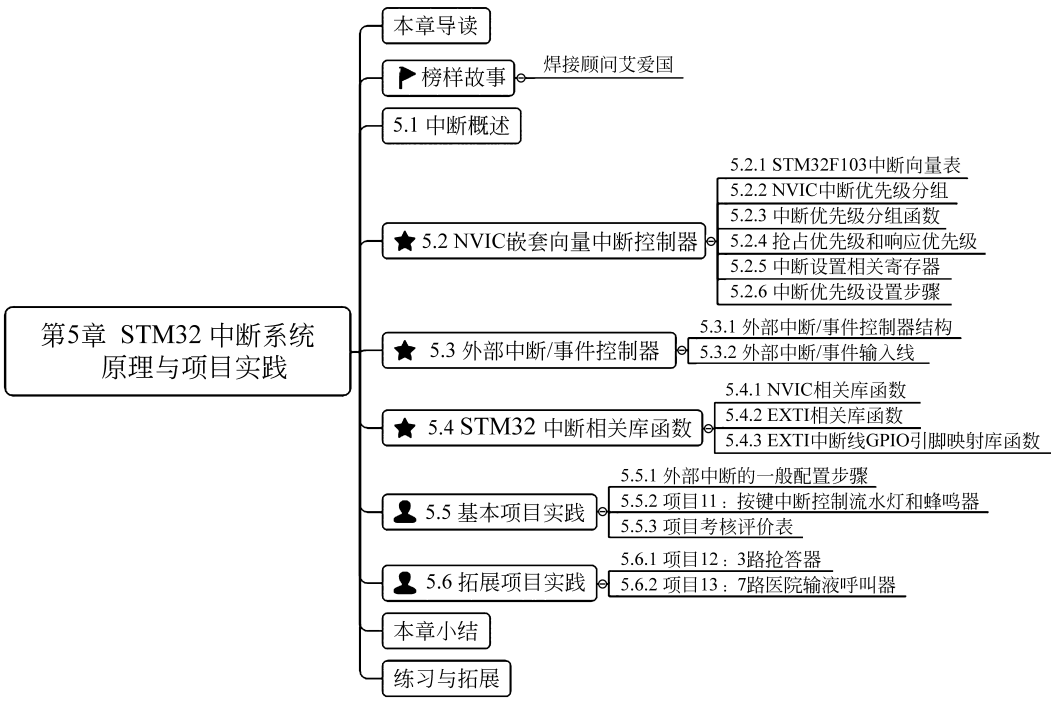

图 5-1　本章素质、知识、能力结构图

本章学习目标

素质目标：艾爱国是爱岗敬业的榜样。学习榜样，培养学生“做事情要做到极致、做人要做到最好”的信念，刻苦学习专业，勤学苦练，掌握扎实的专业理论知识，练就过硬的本领，报效祖国。

知识目标：掌握NVIC嵌套向量中断控制器的基本原理，掌握外部中断/事件控制器(EXTI)和中断相关库函数。

能力目标：具备基本项目开发、创新拓展项目开发能力，培养学生解决综合问题能力和高级思维。

榜样故事

焊接顾问艾爱国(见图5-2)。

出生：1950年3月

籍贯：湖南攸县

职业：湖南华菱湘潭钢铁有限公司焊接顾问

政治面貌：党员

主要荣誉：

图5-2 艾爱国

1987年，被首钢人称为“钢铁缝纫师”。

2021年6月，荣获“七一勋章”。

2021年11月，荣获第八届湖南省道德模范称号(敬业奉献类)。

2021年11月，荣获第八届全国道德模范(全国敬业奉献模范)称号。

2022年3月，被评选为2021年“大国工匠年度人物”。

人物简介：

艾爱国，汉族，1950年3月生，1985年6月入党，是我国焊接领域的领军人物、工匠精神的杰出代表，始终秉持“做事情要做到极致、做人要做到最好”的信念。他从学徒做起，起早贪黑、不怕吃亏、刻苦钻研，攻克焊接技术难关400多个，改进工艺100多项，尤其是在焊接难度最大的紫铜、铝镁合金、铸铁焊接等方面有精深造诣。

1968年9月，艾爱国来到湘钢工作，在焊工岗位上一干就是半个多世纪。艾爱国十分注重技术传承，他主持的湘钢板材焊接实验室被湖南省列为焊接工艺技术重点实验室，并被全国总工会命名为“全国示范性劳模创新工作室”。多年来，他带过的徒弟有几百名。他还无偿地向200多名下岗工人和农村青年传授焊接技术，其中有100余名进入南方电力机车集团、湖南三一重工集团等大型企业。

人物事迹：

艾爱国是我们爱岗敬业的榜样，几十年如一日，以“当工人就要当好工人”作为座右铭，在普通的岗位上勤奋学习、刻苦专研，为党和人民做出了重要贡献。他从进厂那天起，白天认真学艺，晚上刻苦学习专业书籍，长期勤学苦练，系统地阅读了《焊接工艺学》《现代焊接新技术》等(100多本)科技书籍，对专业理论知识掌握较扎实，练就了一手过硬的绝活。1982年，艾爱国在湘潭市锅炉合格焊接考核中，以优异成绩取得气焊、电焊双合格证书，成为全市第一个获得焊接双合格证书者。此后，他更是带头进行生产技术攻关，克服一个又一个难关，创造了一个又一个奇迹。

1983年,艾爱国参加了冶金部为延长高炉风口的使用寿命,组织全国各大钢铁厂研制一种新型风口的攻关。这种新型风口是纯紫铜锥型体,重100多千克,由铸件和锻件组成。紫铜焊件散热快,温度不易掌握,是最难焊的一种金属,加之焊件大,铸件、锻件的材质结构不同,因此,铸件和锻件的焊接成了攻关的最大难题。他大胆提出采取氢弧焊接法进行焊接攻关,并担任主焊手。经过几个月的反复焊接,该新型风口在1984年3月研制成功,安装到高炉上,使用寿命比原风口的使用寿命延长半年;该技术每年节能增效100万元,并获得国家科技进步二等奖。他认真总结这次焊接成功的经验,写成论文《鸽极手工氢弧紫铜风口的焊接工艺》,以后又在深入钻研基础上写出《紫铜氢弧焊接操作法》,比较全面地介绍了各种情况下紫铜焊接的方法。1985年又攻克了氢弧铝合金的难关,撰写了论文《鸽极手工氢弧焊铝及铝合金单面焊双面成型工艺》,还带领17名焊工成功焊接了从德国引进的一台制氧机所有管道的多道焊缝,受到德国专家极力称赞。

如今,艾爱国依然奋战在焊接工艺研究和操作技术开发第一线,为加快我国重点工程建设进度、确保钢结构焊接质量安全做出重要贡献。

5.1 中断概述

5.1.1 中断的概念

中断是一个过程,如在CPU正常执行程序的过程中,遇到外部/内部的紧急事件需要处理,暂时中止当前程序的执行,转而去处理紧急的事件,待处理完后再返回被打断的程序处继续往下执行。中断可以分为“中断响应”“中断处理”“中断返回”3个阶段。中断在计算机多任务处理中能提高CPU的效率,同时能对突发事件做出实时处理。实现程序的并行运行,实现嵌入式系统进程之间的切换。

5.1.2 NVIC介绍

NVIC(nested vectored interrupt controller)是嵌套向量中断控制器,它属于M3内核的一个外设,控制着芯片的中断相关功能。NVIC可支持256个中断,其中包含了16个内核中断和240个外部中断,并且具有256级的可编程中断设置。

内核异常中断指由Cortex-M3内核产生的复位、硬件错误、SysTick定时器中断等中断,而外设中断则是由引脚电平变化、UART或DMA等外设变化引起的中断。

5.2 NVIC嵌套向量中断控制器

5.2.1 STM32F103中断向量表

CM3内核支持256个中断,其中包含了16个内核中断和240个外部中断,并且具有256级的可编程中断设置。STM32并没有使用CM3内核的全部,而是只用了它的一部分。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,其是通过IP bit[7:4]的4位($2^4=16$)进行分配的。而STM32F103系列只有60个可屏蔽中断(STM32F107系列具有68个可屏蔽中断),STM32F103系列中断向量表如表5-1

所示，把优先级从－3～6 的中断向量定义为系统异常，编号为负的（－3、－2、－1）3 个内核异常不能被设置优先级，如 Reset（复位）、NMI（不可屏蔽中断）和硬件失效。优先级从 7 开始，60 个可屏蔽中断为连接到 NVIC 的中断输入信号线，这些中断的优先级可以通过软件进行设置。

表 5-1　STM32F103 中断向量表

位置	优先级	优先级类型	名　　称	说　　明	地址
	—	—	—	保留	0x0000_0000
	－3	固定	Reset	复位	0x0000_0004
	－2	固定	NMI	不可屏蔽中断 RCC 时钟安全系统（CSS）连接到 NMI 向量	0x0000_0008
	－1	固定	硬件失效（HardFault）	所有类型的失效	0x0000_000C
	0	可设置	存储管理（MemManage）	存储器管理	0x0000_0010
	1	可设置	总线错误（BusFault）	预取指失败，存储器访问失败	0x0000_0014
	2	可设置	错误应用（UsageFault）	未定义的指令或非法状态	0x0000_0018
	—	—	—	保留	0x0000_001C～ 0x0000_002B
	3	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000_002C
	4	可设置	调试监控（DebugMonitor）	调试监控器	0x0000_0030
	—	—	—	保留	0x0000_0034
	5	可设置	PendSV	可挂起的系统服务	0x0000_0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000_003C
0	7	可设置	WWDG	窗口定时器中断	0x0000_0040
1	8	可设置	PVD	连到 EXTI 的电源电压检测（PVD）中断	0x0000_0044
2	9	可设置	TAMPER	侵入检测中断	0x0000_0048
3	10	可设置	RTC	实时时钟（RTC）全局中断	0x0000_004C
4	11	可设置	FLASH	闪存全局中断	0x0000_0050
5	12	可设置	RCC	复位和时钟控制（RCC）中断	0x0000_0054
6	13	可设置	EXTIO	EXTI 线 0 中断	0x0000_0058
7	14	可设置	EXTI1	EXTI 线 1 中断	0x0000_005C
8	15	可设置	EXTI2	EXTI 线 2 中断	0x0000_0060
9	16	可设置	EXTI3	EXTI 线 3 中断	0x0000_0064
10	17	可设置	EXTI4	EXTI 线 4 中断	0x0000_0068

续表

位置	优先级	优先级类型	名　　称	说　　明	地址
11	18	可设置	DMA1 通道 1	DMA1 通道 1 全局中断	0x0000_006C
12	19	可设置	DMA1 通道 2	DMA1 通道 2 全局中断	0x0000_0070
13	20	可设置	DMA1 通道 3	DMA1 通道 3 全局中断	0x0000_0074
14	21	可设置	DMA1 通道 4	DMA1 通道 4 全局中断	0x0000_0078
15	22	可设置	DMA1 通道 5	DMA1 通道 5 全局中断	0x0000_007C
16	23	可设置	DMA1 通道 6	DMA1 通道 6 全局中断	0x0000_0080
17	24	可设置	DMA1 通道 7	DMA1 通道 7 全局中断	0x0000_0084
18	25	可设置	ADC1_2	ADC1 和 ADC2 全局中断	0x0000_0088
19	26	可设置	CAN1_TX	CAN1 发送中断	0x0000_008C
20	27	可设置	CAN1_RXO	CAN1 接收 0 中断	0x0000_0090
21	28	可设置	CAN1_RX1	CAN1 接收 1 中断	0x0000_0094
22	29	可设置	CAN_SCE	CAN1SCE 中断	0x0000_0098
23	30	可设置	EXTI9_5	EXTI 线[9:5]中断	0x0000_009C
24	31	可设置	TIM1_BRK	TIM1 刹车中断	0x0000_00A0
25	32	可设置	TIM1_UP	TIM1 更新中断	0x0000_00A4
26	33	可设置	TIM1_TRG_COM	TIM1 触发和通信中断	0x0000_00A8
27	34	可设置	TIM1_Cc	TIM1 捕获比较中断	0x0000_00AC
28	35	可设置	TIM2	TIM2 全局中断	0x0000_00B0
29	36	可设置	TIM3	TIM3 全局中断	0x0000_00B4
30	37	可设置	TIM4	TIM4 全局中断	0x0000_00B8
31	38	可设置	IIC1_EV	IIC1 事件中断	0x0000_00BC
32	39	可设置	IIC1_ER	IIC1 错误中断	0x0000_00C0
33	40	可设置	12C2_EV	IIC2 事件中断	0x0000_00C4
34	41	可设置	IIC2_ER	IIC2 错误中断	0x0000_00C8
35	42	可设置	SPI1	SPI1 全局中断	0x0000_00CC
36	43	可设置	SPI2	SPI2 全局中断	0x0000_00D0
37	44	可设置	USART1	USART1 全局中断	0x0000_00D4
38	45	可设置	USART2	USART2 全局中断	0x0000_00D8
39	46	可设置	USART3	USART3 全局中断	0x0000_00DC
40	47	可设置	EXTI15_10	EXTI 线[15:10]中断	0x0000_00E0
41	48	可设置	RTCAlarm	连到 EXTI 的 RTC 闹钟中断	0x0000_00E4

续表

位置	优先级	优先级类型	名　　称	说　　明	地址
42	49	可设置	USB 唤醒	连到 EXTI 的从 USB 特机唤醒中断	0x0000_00E8
43	50	可设置	TIM8_BRK	TIM8 刹车中断	0x0000_00EC
44	51	可设置	TIM8_UP	TIM8 更新中断	0x0000_00F0
45	52	可设置	TIM8_TRG_COM	TIM8 触发和通信中断	0x0000_00F4
46	53	可设置	TIM8_CC	TIM8 捕获比较中断	0x0000_00F8
47	54	可设置	ADC3	ADC3 全局中断	0x0000_00FC
48	55	可设置	FSMC	FSMC 全局中断	0x0000_0100
49	56	可设置	SDIO	SDIO 全局中新	0x0000_0104
50	57	可设置	TIM5	TIM5 全局中断	0x0000_0108
51	58	可设置	SPI3	SPI3 全局中断	0x0000_010C
52	59	可设置	UART4	UART4 全局中断	0x0000_0110
53	60	可设置	UART5	UART5 全局中断	0x0000_0114
54	61	可设置	TIM6	TIM6 全局中断	0x0000_0118
55	62	可设置	TIM7	TIM7 全局中断	0x0000_011C
56	63	可设置	DMA2 通道 1	DMA2 通道 1 全局中断	0x0000_0120
57	64	可设置	DMA2 通道 2	DMA2 通道 2 全局中断	0×0000_0124
58	65	可设置	DMA2 通道 3	DMA2 通道 3 全局中断	0x0000_0128
59	66	可设置	DMA2 通道 4	DMA2 通道 4 全局中断	0x0000_012C

5.2.2 NVIC 中断优先级分组

NVIC 嵌套向量中断控制器负责 STM32 的中断管理工作，包括中断优先级分组、中断优先级的配置、读中断请求标志、清除中断请求标志、使能中断、清除中断等，控制着 STM32 中断向量表中中断号为 0～59 的 60 个中断。因为我们选择的开发板是 STM32F103 系列，所以就只针对 STM32F103 系列这 60 个可屏蔽中断进行介绍。这 60 个中断是怎么管理的呢？这就涉及 STM32 的中断分组。

对 STM32 中断进行分组，组号为 0～4。同时，对每个中断设置一个抢占优先级和一个响应优先级值。分组配置是在寄存器 SCB－＞AIRCR 中[10:8](3 位)来配置，即 111 为 0 组、110 为 1 组、101 为 2 组、100 为 3 组、011 为 4 组。AIRCR 中断分组设置表如表 5-2 所示。

表 5-2 AIRCR 中断分组设置表

组	AIRCR[10:8]（3 位）	IP bit[7:4]分配情况（4 位）	分配结果	NVIC_PriorityGroupConfig
0	111	0:4	0 位抢占优先级，4 位响应优先级	NVIC_PriorityGroupConfig_0
1	110	1:3	1 位抢占优先级，3 位响应优先级	NVIC_PriorityGroupConfig_1
2	101	2:2	2 位抢占优先级，2 位响应优先级	NVIC_PriorityGroupConfig_2
3	100	3:1	3 位抢占优先级，1 位响应优先级	NVIC_PriorityGroupConfig_3
4	011	4:0	4 位抢占优先级，0 位响应优先级	NVIC_PriorityGroupConfig_4

从表 5-2 可以看出，AIRCR 寄存器的 AIRCR[10:8](3 位)确定是哪种分组，IP 寄存器 IP bit[7:4](4 位)是相对应于那种分组抢占优先级和响应优先级的分配比例。例如，组设置成 2，那么此时所有的 60 个中断优先 IP 寄存器高 4 位中的最高 2 位是抢占优先级，低 2 位为响应优先级。CM3 中定义了 8 个 bit 用于设置中断源的优先级，而 STM32 只选用其中的 4 个 bit。这 4 位优先级分配如下。

0 组：IP bit[7:4]为 0:4。0 位抢占优先级(没有抢占优先级)，4 位响应优先级，即 NVIC 配置的 $2^4=16$ 种中断向量都是响应优先级，没有抢占优先级。

1 组：IP bit[7:4]为 1:3。1 位抢占优先级，3 位响应优先级。这里表示有 $2^1=2$ 种级别的抢占优先级(0 级、1 级)；有 $2^3=8$ 种响应优先级，即 NVIC 配置的 16 种中断向量中有 8 种中断的抢占优先级都是 0 级，而它们的响应优先级分别是 0～7，其余 8 种中断的抢占优先级都为 1 级，响应优先级分别是 0～7。

2 组：IP bit[7:4]为 2:2。2 位抢占优先级，2 位响应优先级。这里表示有 $2^2=4$ 种级别的抢占优先级(0 级、1 级、2 级、3 级)，有 $2^2=4$ 种级别的响应优先级(0 级、1 级、2 级、3 级)。

3 组：IP bit[7:4]为 3:1。3 位抢占优先级，1 位响应优先级。这里表示有 $2^3=8$ 种级别的抢占优先级是 0～7，有 $2^1=2$ 种级别的响应优先级(0 级、1 级)。

4 组：IP bit[7:4]为 4:0。4 位抢占优先级，0 位响应优先级。这里表示有 $2^4=16$ 种中断具有不相同的抢占优先级，没有响应优先级。

5.2.3 中断优先级分组函数

通过调用 STM32 固件库中的函数 void NVIC_PriorityGroupConfig(uint32_t NVIC_PriorityGroup)，选择使用哪种优先级分组。这个函数的参数有如下 5 种。

```
NVIC_PriorityGroupConfig_0;          //选择 0 组:0 位抢占优先级,4 位响应优先级
NVIC_PriorityGroupConfig_1;          //选择 1 组:1 位抢占优先级,3 位响应优先级
NVIC_PriorityGroupConfig_2;          //选择 2 组:2 位抢占优先级,2 位响应优先级
NVIC_PriorityGroupConfig_3;          //选择 3 组:3 位抢占优先级,1 位响应优先级
NVIC_PriorityGroupConfig_4;          //选择 4 组:4 位抢占优先级,0 位响应优先级
```

例如：

```
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);    //分组 2
```

5.2.4 抢占优先级和响应优先级

STM32 的中断源有两种优先级，一种为抢占优先级，另一种为响应优先级，编号数字越小，优先级别越高(0 表示最高、4 表示最低)。

(1) 高优先级的抢占优先级可以打断正在进行的低抢占优先级中断。

例如：分组设置为 2，抢占为 0～3，设置 A 为 0，设置 B 为 1，B 正在中断，A 可以打断 B 中断，获得中断权。

(2) 抢占优先级相同的中断，高响应优先级不可以打断低响应优先级的中断。

(3) 抢占优先级相同的中断，当两个中断同时发生的情况下，哪个响应优先级高，哪个先执行。

(4) 如果两个中断的抢占优先级和响应优先级都一样，则看哪个中断先发生就先执行。

例如：设置中断优先级组为 2，然后设置如下。

中断 3(RTC 中断)的抢占优先级为 2，响应优先级为 1。

中断 6(EXTI0 外部中断 0)的抢占优先级为 3，响应优先级为 0。

中断 7(EXTI1 外部中断 1)的抢占优先级为 2，响应优先级为 0。

那么这 3 个中断的优先级顺序为：中断 7＞中断 3＞中断 6。

【注意】 中断 7 与中断 3 不存在抢占，当两个中断同时发生的情况下，哪个响应优先级高，哪个先执行，因为中断 7 响应优先级 0 高于中断 3 响应优先级 1，所以先执行中断 7，中断 6 再中断，中断 3(中断 7)可以打断。

一般情况下，系统代码执行过程中，只设置一次中断优先级分组(可以通过寄存器 SCB－＞AIRCR 设置，也可以通过库函数设置)，比如分组 2，设置好分组后一般不会再改变分组。随意改变分组会导致中断管理混乱，程序出现意想不到的执行结果。

5.2.5 中断设置相关寄存器

在 CMSIS 库头文件 core_cm3.h 中定义了 NVIC 中断的相关操作，这里介绍开放中断、关闭中断、设置中断请求标志、读中断请求标志、清除中断请求标志、设置中断优先级和获取中断优先级的函数。MDK 为 NVIC 相关的寄存器定义了如下结构体。

```
typedef struct 偏移地址中断设置使能寄存器
{
  __IO uint32_t ISER[8];          //偏移地址:0x000(可读/可写)中断使能寄存器
  uint32_t RESERVED0[24];
  __IO uint32_t ICER[8];          //偏移地址:0x080(可读/可写)中断除能寄存器
  uint32_t RSERVED1[24];
  __IO uint32_t ISPR[8];          //偏移地址:0x100 (可读/可写)中断挂起控制寄存器 * /
  uint32_t RESERVED2[24];
  __IO uint32_t ICPR[8];          //偏移地址:0x180 (可读/可写)中断解挂控制寄存器 * /
  uint32_t RESERVED3[24];
  __IO uint32_t IABR[8];          //偏移地址:0x200(只读)中断激活标志位寄存器
  uint32_t RESERVED4[56];
  __IO uint8_t IP[240];           //偏移地址:0x300(可读/可写)中断优先级寄存器
  uint32_t RESERVED5[644];
  __O  uint32_t STIR;             //偏移地址:0xE00(只写)软件触发中断寄存器
}  NVIC_Type;
```

NVIC嵌套向量中断控制器中断管理主要包括开放中断、关闭中断、设置中断请求标志、读中断请求标志、清除中断请求标志和配置中断优先级等。NVIC嵌套向量中断控制器的寄存器有ISER0、ISER1、ICER0、ICER1、ISPR0、ISPR1、ICPR0、ICPR1、IABR0、IABR1、IPR0～IPR14和STIR，如表5-3所示。

表5-3 NVIC寄存器

循环	地 址	寄存器	名 称	功能描述
1	0xE000E100	ISER0	中断使能寄存器	ISER0[0]～ISER0[31]、ISER1[0]～ISER1[27]依次对应中断号为0～59的中断，各位写0无效，写1开放中断
	0xE000E104	ISER1		
2	0xE000E180	ICER0	中断除能寄存器	ICER0[0]～ICER0[31]、ICER1[0]～ICER1[27]依次对应中断号为0～59的中断，各位写0无效，写1关闭中断
	0xE000E184	ICER1		
3	0xE000E200	ISPR0	中断挂起控制寄存器	ISPR0[0]～ISPR0[31]、ISPR1[0]～ISPR1[27]依次对应中断号为0～59的中断，各位写0无效，写1中断挂起
	0xE000E204	ISPR1		
4	0xE000E280	ICPR0	中断解挂控制寄存器	ICPR0[0]～ICPR0[31]、ICPR1[0]～ICPR1[27]依次对应中断号为0～59的中断，各位写0无效，写1可以将正在挂起的中断解挂
	0xE000E284	ICPR1		
5	0xE000E300	IABR0	中断激活标志位寄存器(只读)	IABR0[0]～IABR0[31]、IABR1[0]～IABR1[27]依次对应中断号为0～59的中断，各位读出1，相应中断激活
	0xE000E304	IABR1		
6	0xE000E400～0xE000E438	IPR0～IPR14	中断优先级寄存器	共有16个优先级，优先级号为0～15，优先级号0表示优先级最高，优先级号15表示优先级最低
7	0xE000EF00	STIR	软件触发中断寄存器	第[8:0]位域有效，写入0～59中的某一中断号，则触发相应的中断

下面重点介绍这几个寄存器。

先介绍几个寄存器组长度为8，这些寄存器是32位寄存器。由于STM32只有60个可屏蔽中断，因此8个32位寄存器中只需要2个就有64位了，每1位控制一个中断。

ISER[8]：ISER(interrupt set-enable registers)中断使能寄存器组。CM3内核支持256个中断，用8个32位(8×32＝256)寄存器来控制，每一个位控制一个中断。而STM32F103系列只有60个可屏蔽中断，只使用到了ISER[0]和ISER[1]，ISER[0]的bit0～bit31分别对应中断0～31，ISER[1]的bit0～bit27对应中断32～59，这样总共60个可屏蔽中断就分别对应上了。要使能某个中断，就必须设置相应的ISER位为1，使该中断被使能(这仅仅是使能，还要配合中断分组、屏蔽、I/O口映射等设置才算完整的中断设置)。

ICER[8]：ICER(interrupt clear-enable registers)中断除能寄存器组。该寄存器的作用与ISER相反。这里专门设置一个ICER来清除中断位，要除能某个中断，就必须设置相应的ICER位为1；使该中断被除能，写0无效。

ISPR[8]：ISPR(interrupt set-pending registers)中断挂起控制寄存器组。通过置1可以将正在进行的中断挂起，执行同级或者更高级别的中断，写0无效。

ICPR[8]：ICPR(interrupt clear-pending registers)中断解挂控制寄存器组。通过置1可以将正在挂起的中断解挂，写0无效。

IABR[8]：IABR(interrupt active-bit registers)中断激活标志位寄存器组。这是一个只读寄存器，通过它可以知道当前在执行的中断是哪一个(为1)，在中断执行完后硬件自动清零。

IP[240]：IP(interrupt priority registers)中断优先级寄存器组。这个用来控制每个中断优先级的。由于STM32只用到了前60个可屏蔽中断，IP[59]～IP[0]分别对应中断59～0。每个可屏蔽中断占用的8bit并没有全部使用，而只用了IP寄存器的高4位，这个IP寄存器的高4位[7:4]用来设置抢占和响应优先级，低4位没有用到，前面已经介绍了这部分内容。

5.2.6 中断优先级设置步骤

(1) 系统运行后先设置中断优先级分组。调用函数：

```
void NVIC_PriorityGroupConfig(uint32_t NVIC_PriorityGroup);
                                                    /*确定抢占、响应各几个*/
```

【注意】 整个系统执行过程中，只设置一次中断分组，通常设置分组2。

(2) 针对每个中断，设置对应的抢占优先级和响应优先级。调用函数：

```
void NVIC_Init(NVIC_InitTypeDef* NVIC_InitStruct);     /*哪个通道、抢占、响应优先级、使能通道*/
```

NVIC_InitTypeDef结构体有以下4个成员变量。

① NVIC_IRQChannel：定义初始化的是哪一个中断。可以在stm32f10x.h文件中查到每个中断对应的名称，如USART1_IRQn。

② NVIC_IRQChannelPreemptionPriority：定义此中断的抢占优先级别。

③ NVIC_IRQChannelSubPriority：定义此中断的响应优先级别。

④ NVIC_IRQChannelCmd：该中断是否使能。

(3) 如果需要挂起/解挂，查看中断当前激活状态，分别调用相关函数即可。

例如，使能串口1中断，抢占优先级为1，响应优先级为2，初始化的方法为：

```
NVIC_InitTypeDef NVIC_InitStructure;
NVIC_InitStructure.NVIC_IRQChannel=USART1_IRQn;          //串口1中断
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority=1;  //抢占优先级为1
NVIC_InitStructure.NVIC_IRQChannelSubPriority=2;         //响应优先级为2
NVIC_InitStructure.NVIC_IRQChannelCmd=ENABLE;            //IRQ通道使能
NVIC_Init(&NVIC_InitStructure);                  //根据上面指定的参数初始化NVIC寄存器
```

5.3 外部中断/事件控制器

外部中断/事件控制器由19个产生事件/中断请求的边沿检测器组成，每个输入线可以独立地配置输入类型(脉冲或挂起)和对应的触发事件(上升沿、下降沿或者双边沿都触发)。每个输入线都可以独立地被屏蔽，挂起寄存器保持着状态线的中断请求。