NVIC中断

NVIC和处理器内核紧密相连，用于总体管理异常，称之为“内嵌向量中断控制器： Nested Vectored Interrupt Controller (NVIC)"。

CM3/4内核支持256个中断，其中包含了16个内核中断和240个核外中断。
具有256级的可编程中断设置。

STM32F4并没有使用CM4内核的全部中断，而是只用了它的一部分。\
-STM32F40xx/STM32F41xx总共有92个中断。\
-STM32F42xx/STM32F43xx则总共有96个中断(包括10个内核中断和82个可屏蔽中断，具有16级可编程的中断优先级，而我们常用的就是这 82个可屏蔽中断。)

NVIC中断优先级分组

分组（*在寄存器SCB->AIRCR中配置*） -> 对每个中断设置抢断优先级和响应优先级

抢断优先级和响应优先级的区别

较高优先级的抢占优先级中断，可以打断正在进行的较低抢占优先级的中断；\
抢占优先级相同的中断，较高响应优先级不可以打断较低响应优先级的中断；\
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；\
抢占优先级和响应优先级都是一样的话，则看哪个中断先发生就先执行。

组	AIRCR[10：8]	IP bit[7：4]分配情况	分配结果	优先级数
0	111	0：4	0位抢占优先级，4位响应优先级	响应优先级0-15
1	110	1：3	1位抢占优先级，3位响应优先级	抢占优先级0-1；响应优先级0-7
2	101	2：2	2位抢占优先级，2位响应优先级	抢占优先级0-3；响应优先级0-3
3	100	3：1	3位抢占优先级，1位响应优先级	抢占优先级0-7；响应优先级0-1
4	011	4：0	4位抢占优先级，0位响应优先级	抢占优先级0-15

中断优先级设置

中断优先级控制的寄存器组：IP[240] 全称：Interrupt Priority Registers；240个8位寄存器，每个中断使用一个寄存器来确定优先级。

STM32F40x系列一共82个可屏蔽中断，使用IP[81]~IP[0]。
每个IP寄存器的高4位[7：4]用来设置抢占优先级和响应优先级（根据分组），低4位[3：0]没有用到。

外部中断

STM32F4的每个IO都可以作为外部中断输入。stm32f4的中断控制器支持23个外部中断请求：\

EXTI线0~15：对应外部IO口的输入（IN）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。\

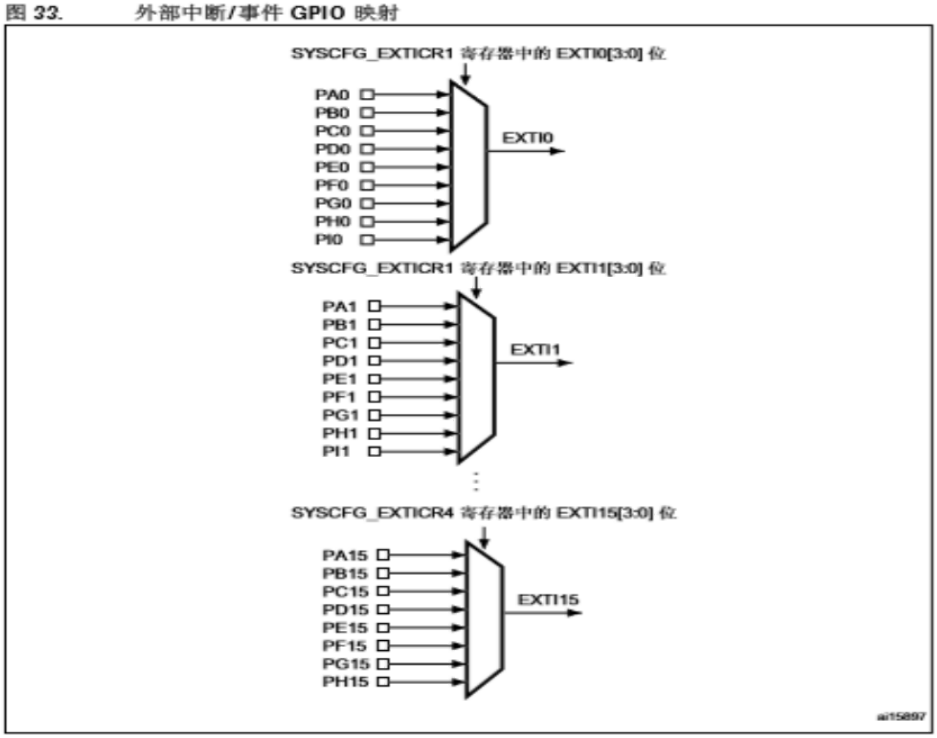
EXTI线22：连接到RTC唤醒事件。\

这些均在stm32内部产生！STM32F4供IO使用的中断线只有16个。

10.2.5 外部中断/事件线映射

多达 140 个 GPIO（STM32F405xx/07xx 和 STM32F415xx/17xx）通过以下方式连接到 16 个外部中断/事件线：

图 33. 外部中断/事件 GPIO 映射




```
GPIOx.0映射到EXTI0
GPIOx.1映射到EXTI1
…
GPIOx.15映射到EXTI15
```

```
void SYSCFG_EXTILineConfig(uint8_t EXTI_PortSourceGPIOx, uint8_t
EXTI_PinSourcex);
      //设置IO口与中断线的映射关系
exp:     SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOE, EXTI_PinSource2);
```

8.2.4 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1)

SYSCFG external interrupt configuration register 1

偏移地址：0x08

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **EXTIx[3:0]**：EXTI x 配置（x = 0 到 3）(EXTI x configuration (x = 0 to 3))
这些位通过软件写入，以选择 EXTIx 外部中断的源输入。
0000：PA[x] 引脚
0001：PB[x] 引脚
0010：PC[x] 引脚
0011：PD[x] 引脚
0100：PE[x] 引脚
0101：PF[C] 引脚
0110：PG[x] 引脚
0111：PH[x] 引脚
1000：PI[x] 引脚

SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

SYSCFG 外部中断配置寄存器 4 (SYSCFG_EXTICR4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

编程格式

```
void EXTIx_IRQHandler(void)
{
    if(EXTI_GetITStatus(EXTI_Linex) != RESET)
    {
        。。。。。。。。。;     //中断服务程序的具体功能代码
        EXTI_ClearITPendingBit(EXTI_Linex);
    }

}
```

作业

总结下列知识点：（独立思考，以书面资料形式总结好）

STM32F407 有多少个中断？向量表？中断名字？服务程序名字？NVIC等优先级管理模式？ 相关寄存器和配置函数？（设置哪些内容？）NVIC优先级设置步骤？ 外部中断（EXTI）共有几个？EXTIx如何配置GPIOx引脚？EXTI还可以有哪些配置？EXTIx中断的编程步骤？

① STM32F407 有多少个中断？

答：STM32F40xx/STM32F41xx的92个中断里面，包括10个内核中断和82个可屏蔽中断，具有16级可编程的中断优先级，而我们常用的就是这82个可屏蔽中断。

② 向量表？中断名字？服务程序名字？（自己列出所有名字和向量表中位置）

提示：中断名字，查看中文参考手册，表45，（具体列出。。。。。）；
服务程序名字：（查看工程中这个文件），内核中断XXX_Handler()，可屏蔽中断XXX_IRQHandler()（具体列出。。。。。）；

③ NVIC等优先级管理模式？

答：首先，对STM32中断进行分组，组0~4。
然后，对每个中断设置： 一个抢占优先级、和一个响应优先级。

④ 相关寄存器和配置函数？（设置哪些内容？）

◆ 对于每个中断怎么设置优先级?
中断优先级控制的寄存器组：IP[240]
全称是：Interrupt Priority Registers

240个8位寄存器，每个中断使用一个寄存器来确定优先级。
STM32F40x系列一共82个可屏蔽中断，使用IP[81]~IP[0]。

每个IP寄存器的高4位[7：4]用来设置抢占优先级和响应优先级（根据分组），低4位[3：0]没有用到。

◆ MDK（编译系统）中NVIC寄存结构体

```
typedef struct
{
  __IO uint32_t ISER[8];               //中断使能寄存器组
     uint32_t RESERVED0[24];
  __IO uint32_t ICER[8];               //中断失能寄存器组
     uint32_t RSERVED1[24];
  __IO uint32_t ISPR[8];               //中断挂起寄存器组
     uint32_t RESERVED2[24];
  __IO uint32_t ICPR[8];               //中断解挂寄存器组
     uint32_t RESERVED3[24];
  __IO uint32_t IABR[8];               //中断激活标志位寄存器组
     uint32_t RESERVED4[56];
  __IO uint8_t  IP[240];                //中断优先级控制的寄存器组
     uint32_t RESERVED5[644];
  __O  uint32_t STIR;                  //软件触发中断寄存器
} NVIC_Type;
```

```
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
设置一次中断分组。
void NVIC_Init(NVIC_InitTypeDef* NVIC_InitStruct);
void  NVIC_SetPendingIRQ(IRQn_Type  IRQn);
uint32_t  NVIC_GetPendingIRQ(IRQn_Type  IRQn);
void  NVIC_ClearPendingIRQ(IRQn_Type  IRQn)
uint32_t NVIC_GetActive(IRQn_Type IRQn)
```

⑤ NVIC优先级设置步骤？

◆ **中断优先级编程步骤：（初始化、操作）**

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);

整个系统执行过程中，只设置一次中断分组。

② 针对每个中断，设置对应的抢占优先级和响应优先级：

void NVIC_Init(NVIC_InitTypeDef NVIC_InitStruct);*

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。

⑥ 外部中断（EXTI）共有几个？

◆ 并不是16个中断线就可以分配16个中断服务函数

◆ IO 口外部中断在中断向量表中只分配了7个中断向量，即只能使用7个中断服务函数。

位置	优先级	优先级类型	名称	说明	地址
6	13	可设置	EXTI0	EXTI线0中断	0x0000 0058H
7	14	可设置	EXTI1	EXTI线1中断	0x0000 005CH
8	15	可设置	EXTI2	EXTI线2中断	0x0000 0060H
9	16	可设置	EXTI3	EXTI线3中断	0x0000 0064H
10	17	可设置	EXTI4	EXTI线4中断	0x0000 0068H
23	30	可设置	EXTI9_5	EXTI线[9:5]中断	0x0000 009CH
40	47	可设置	EXTI15_10	EXTI线[15:10]中断	0x0000 00E0H

◆ 从表中可以看出：

外部中断线 5 ~ 9分配一个中断向量，共用一个中断服务函数

外部中断线10~15分配一个中断向量，共用一个中断服务函数。

⑦ EXTIx如何配置GPIOx引脚？

```
void SYSCFG_EXTILineConfig(uint8_t EXTI_PortSourceGPIOx, uint8_t
EXTI_PinSourcex);
      //设置IO口与中断线的映射关系
exp:     SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOE, EXTI_PinSource2);
```

⑧ EXTI还可以有哪些配置？

```
typedef struct
{
  uint32_t EXTI_Line;                    //指定要配置的中断线
  EXTIMode_TypeDef  EXTI_Mode;           //模式：事件 OR中断
  EXTITrigger_TypeDef  EXTI_Trigger;     //触发方式：上升沿/下降沿/双沿触发
  FunctionalState  EXTI_LineCmd;         //使能 OR失能
} EXTI_InitTypeDef;
```

⑨ EXTIx中断的编程步骤

7 外部中断一般配置过程

① 使能SYSCFG时钟：
RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);

② 初始化IO口为输入。
GPIO_Init();

③ 设置IO口与中断线的映射关系。
SYSCFG_EXTILineConfig();

④ 初始化线上中断，设置触发条件等。
EXTI_Init();

⑤ 配置中断分组（NVIC），并使能中断。
NVIC_Init();

⑥ 编写中断服务函数。
EXTIx_IRQHandler();

⑦ 清除中断标志位
EXTI_ClearITPendingBit();