



第五章 中断系统



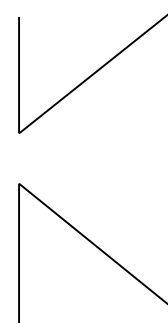
isszym 2020.10.14

概述

□ 什么是中断？

- ❖ 所谓中断(**Interrupt**)，就是处理事件的一个过程，例如，有按键发生就可能引起中断，此时会向**CPU**发出**中断请求**。**ARM**把中断称为**异常(Exception)**。引起中断的事件为**中断源**。
- ❖ 此时，**CPU**会中止当前程序去响应中断，即转去执行此中断的处理程序，执行完毕后再返回到中断点继续执行原来的程序。为异常事件提供服务的程序称为**异常(中断)处理程序**。
- ❖ 如果没有中断功能，只能通过查询方式发现事件有没有发生。

当前程序 中断服务程序



□ 中断优先级

- ❖ 如果正在执行一个异常的处理程序时另一个异常到来，是否可以转去执行这个异常的处理程序呢？这就要看它们的**优先级**。
- ❖ 高优先级的异常可以中断低优先级异常的处理程序，形成**中断嵌套**。
- ❖ 低优先级或同级的异常不能中断当前异常的处理程序。

- ❖ 不同的中断(异常)具有不同的中断优先级，**ARM**处理器中有**7**种类型的异常，它们的优先级从高到低排列：
 - 复位异常(reset)
 - 数据异常(data abort)
 - 快速中断异常(fiq)
 - 外部中断异常(irq)
 - 预取异常(prefetch abort)
 - 软件中断(swi)
 - 未定义指令异常(undef)

□ 中断分类

- ❖ 中断可以分为可屏蔽中断和不可屏蔽中断。由外设产生的中断，如：键盘中断、定时器中断等，可以通过设置中断控制器进行屏蔽，也就是不让它们发生。硬件出错，如突然掉电、除数为零、单步中断等引起的中断是不可屏蔽中断。
- ❖ 除了硬件产生的硬中断，还有软中断，软中断主要用于调用的一段程序。

程序状态寄存器

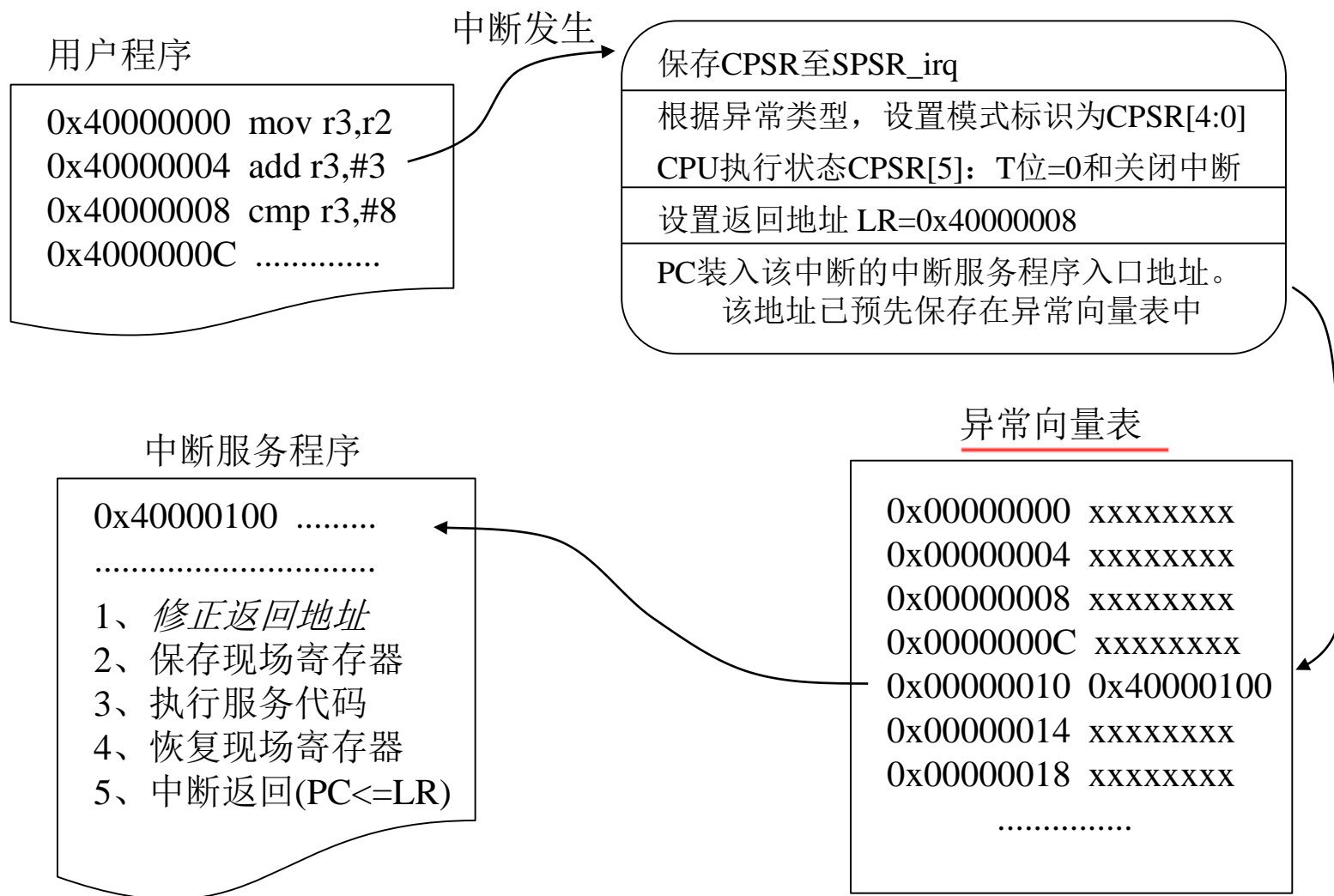
- CPSR: 当前程序状态寄存器 (current program status register) , 在任何处理器模式下被访问。它包含了条件标志位、中断禁止位、当前处理器模式标志以及其他的一些控制和状态位。CPSR在用户级编程时用于存储条件码。
- SPSR: 程序状态保存寄存器 (saved program status register) , 每一种处理器模式下都有一个状态寄存器SPSR, SPSR用于保存CPSR的状态, 以便异常返回后恢复异常发生时的工作状态。当特定的异常中断发生时, 这个寄存器用于存放当前程序状态寄存器的内容。在异常中断退出时, 可以用SPSR来恢复CPSR。由于用户模式和系统模式不是异常中断模式, 所以他没有SPSR。当用户在用户模式或系统模式访问SPSR, 将产生不可预知的后果。

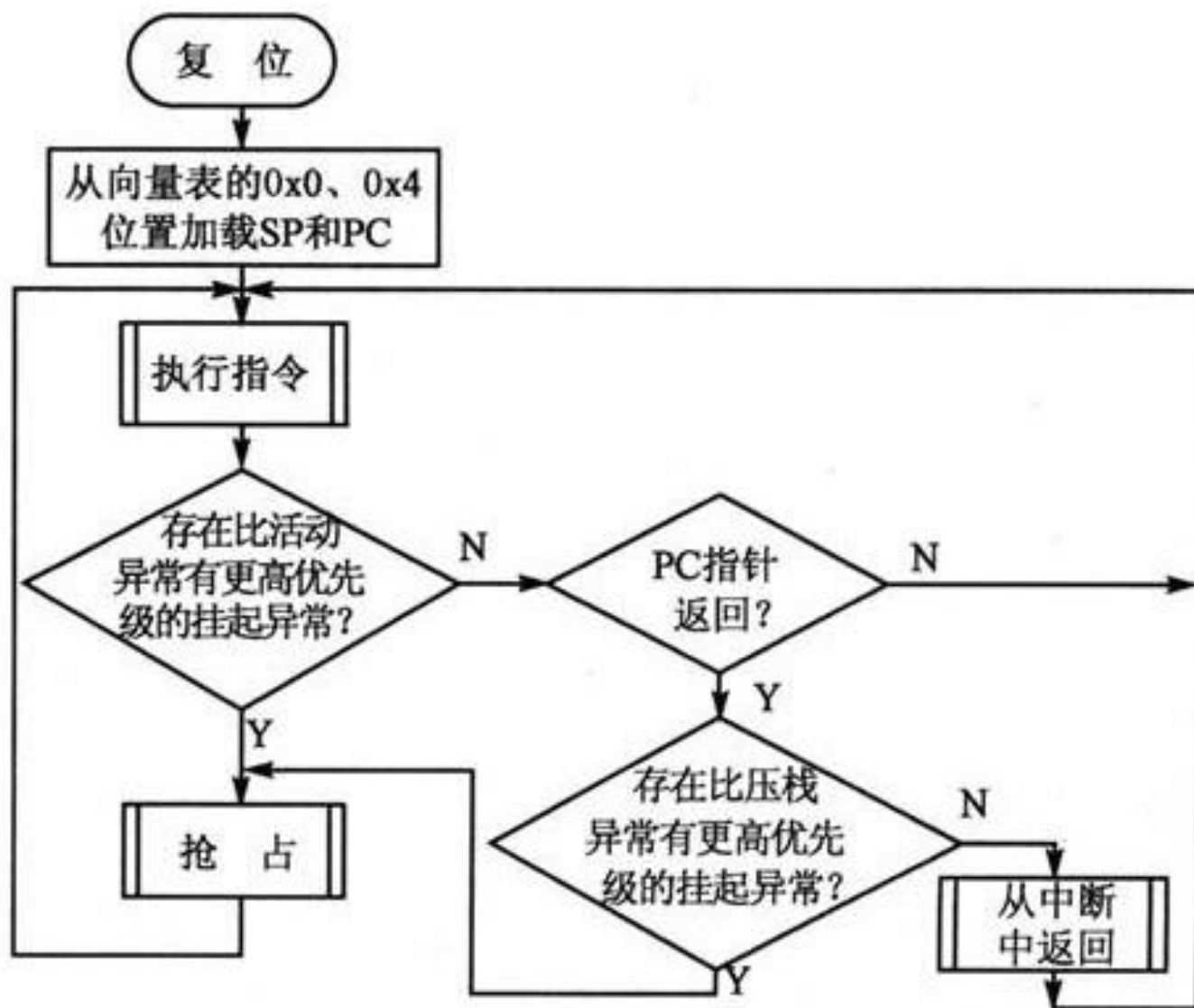
CPSR格式如下所示。SPSR和CPSR格式相同。

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----------|--|---|---|---|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| N | Z | C | V | Q | DNM(RAZ) | | I | F | T | M4 | M3 | M2 | M1 | M0 |

中断过程

硬件自动完成





STM32F103的中断

❑ 可屏蔽中断

- ❖ STM32芯片内部有16个中断源，外部支持60个可屏蔽中断源。由于STM32F103的EXTI控制器只能同时监听19路中断线，外部中断被分成了19组。它们都是通过边沿来触发的。每个中断设有状态位，有独立的触发和屏蔽设置。

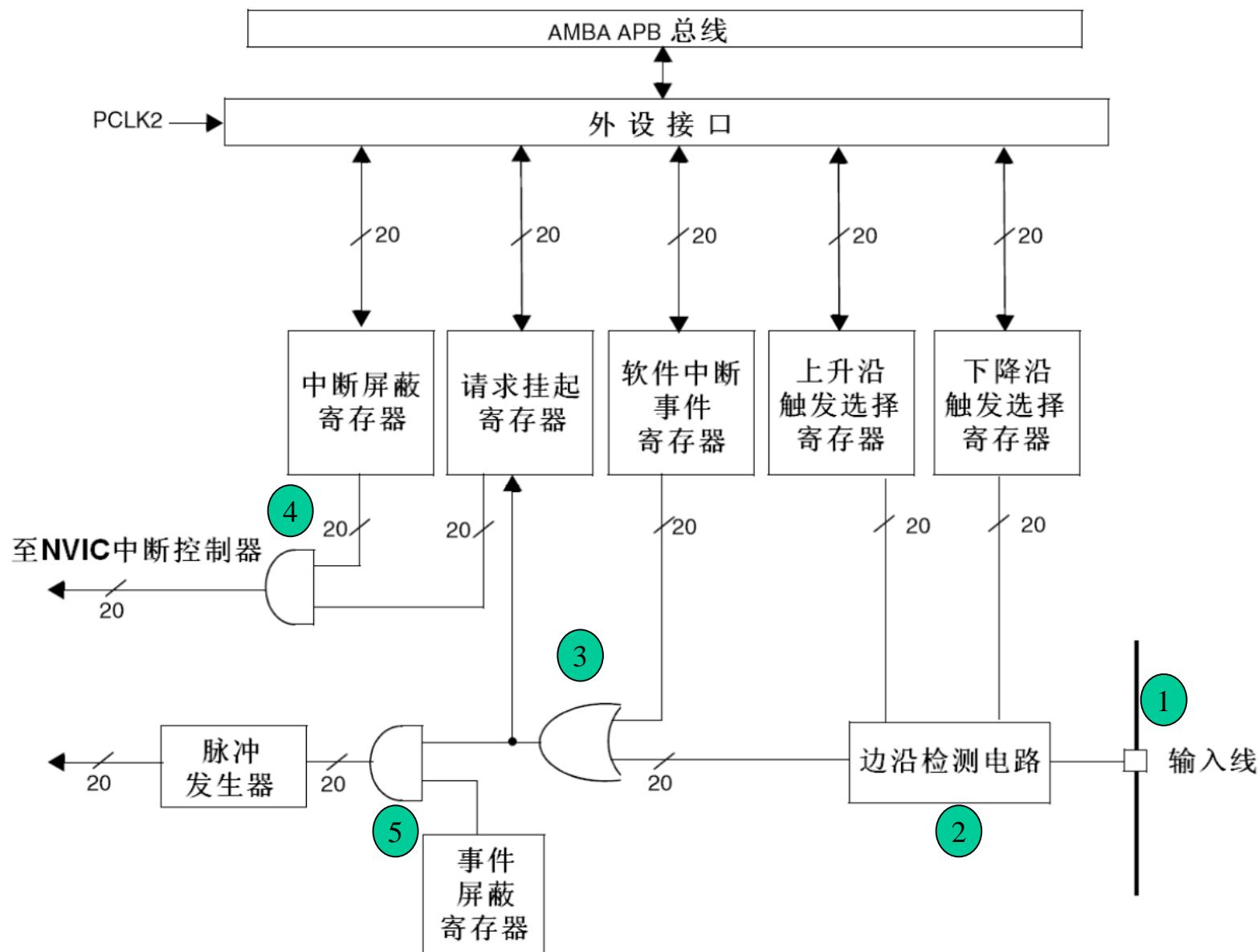
❑ 中断线

- ❖ EXTI_Line0~15: 对应外部 IO口的输入中断。
- ❖ EXTI_Line16: 连接到 PVD(Programmable Voltage Detector)输出
- ❖ EXTI_Line17: 连接到 RTC 闹钟事件。
- ❖ EXTI_Line18: 连接到 USB 唤醒事件。

❑ STM32外部中断分组

- ❖ 每一个GPIO引脚的都能配置成一个外部中断源。
- ❖ STM32 通过根据引脚序号将众多中断触发源分成16组，比如：PA0、PB0、...、PG0一组。每一组中同时只能有一个中断源工作，也就是说，最多有16个外部中断同时工作。

工作原理



□ 中断条件

- ❖ 要产生中断，必须先配置好并使能中断线。根据需要的边沿检测设置2个触发寄存器，同时在中断屏蔽寄存器的相应位写'1'允许中断请求。
- ❖ 当外部中断线上发生了期待的边沿时，将产生一个中断请求，对应的挂起位也随之被置'1'。在挂起寄存器的对应位写'1'，将清除该中断请求。如果需要产生事件，必须先配置好并使能事件线。
- ❖ 根据需要的边沿检测通过设置2个触发寄存器，同时在事件屏蔽寄存器的相应位写'1'允许事件请求。当事件线上发生了需要的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置'1'。
- ❖ 通过在软件中断/事件寄存器写'1'，也可以通过软件产生中断/事件请求。

□ 硬件事件选择

通过下面的过程，可以配置20个线路为事件源：

- ❖ 配置20个事件线的屏蔽位(EXTI_EMR)
- ❖ 配置事件线的触发选择位(EXTI_RTSR和EXTI_FTSR)

❑ 软件中断/事件的选择

20个线路可以被配置成软件中断/事件线。下面是产生软件中断的过程：

- ❖ 配置**20**个中断/事件线屏蔽位(**EXTI_IMR**, **EXTI_EMR**)
- ❖ 设置软件中断寄存器的请求位(**EXTI_SWIER**)

❑ 硬件中断选择

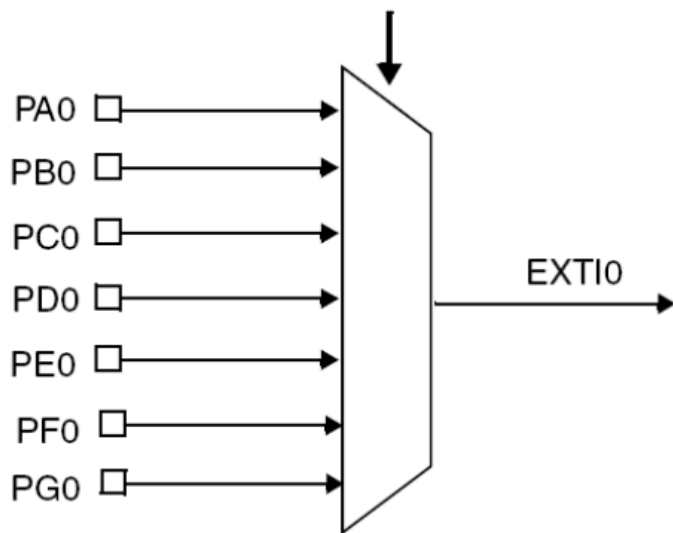
通过下面的过程来配置**20**个线路做为中断源：

- ❖ 配置**20**个中断线的屏蔽位(**EXTI_IMR**)
- ❖ 配置所选中断线的触发选择位(**EXTI_RTSR**和**EXTI_FTSR**)；
- ❖ 配置对应到外部中断控制器(**EXTI**)的**NVIC**中断通道的使能和屏蔽位，使得**20**个中断线中的请求可以被正确地响应。

外部中断/事件线路映像

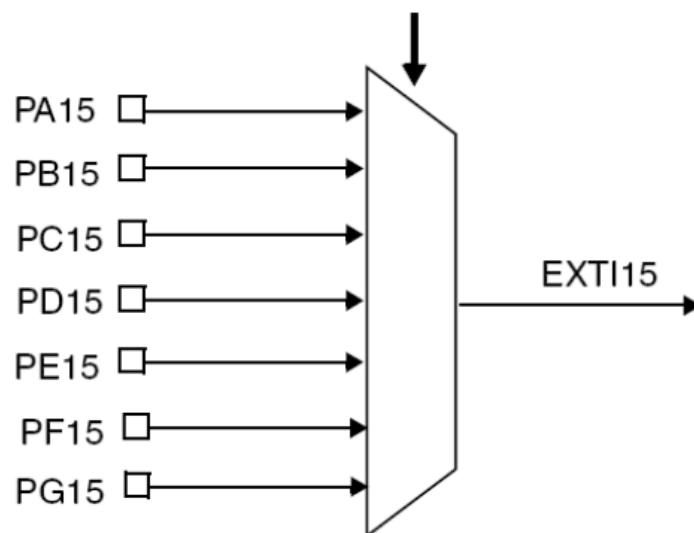
□ 112通用I/O端口连接到16个外部中断/事件线上：

在AFIO_EXTICR1寄存器的EXTI0[3:0]位

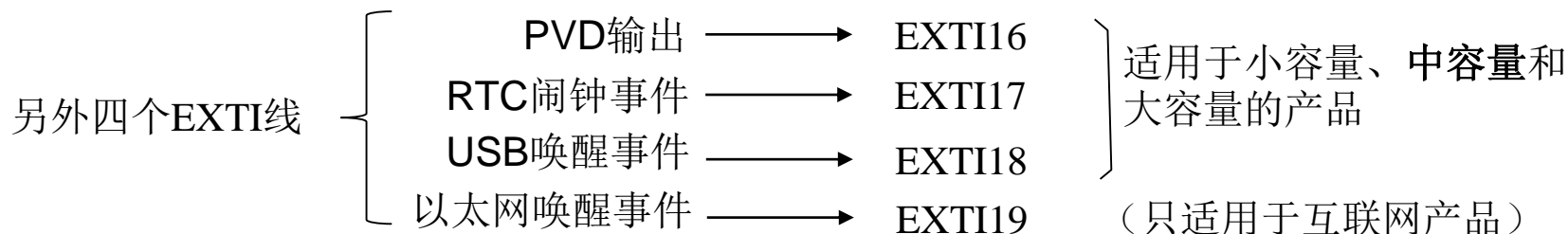


...

在AFIO_EXTICR4寄存器的EXTI15[3:0]位



通过AFIO_EXTICRx配置GPIO线上的外部中断/事件，必须先使能AFIO时钟。



外部中断配置寄存器1(AFIO_EXTICR1)

地址偏移: 0x08 复位值: 0x0000

基地址: 0x40010000



位15:0 **EXTIx[3:0]**: EXTIx配置(x = 0 ... 3) (EXTI x configuration) 这些位可由软件读写, 用于选择EXTIx外部中断的输入源。

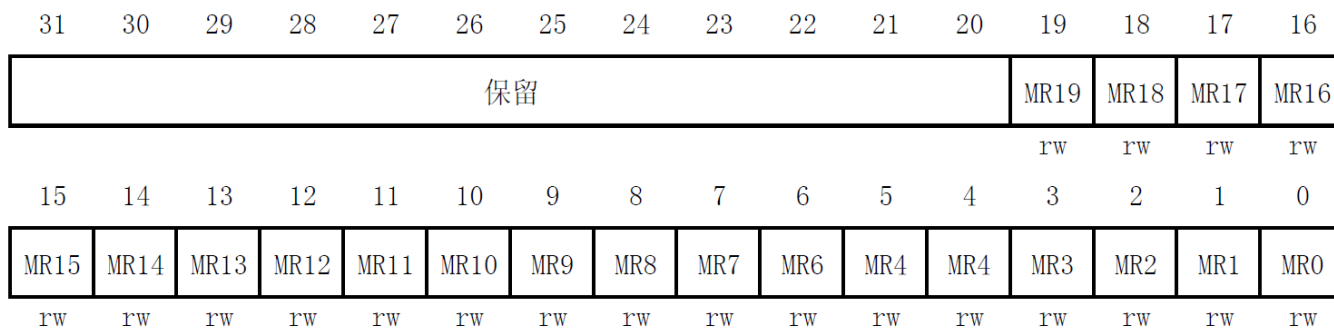
| | | | |
|---------------|---------------|---------------|---------------|
| 0000: PA[x]引脚 | 0001: PB[x]引脚 | 0010: PC[x]引脚 | 0011: PD[x]引脚 |
| 0100: PE[x]引脚 | 0101: PF[x]引脚 | 0110: PG[x]引脚 | |

AFIO_EXTICR2~4用于配置另外12个EXTI

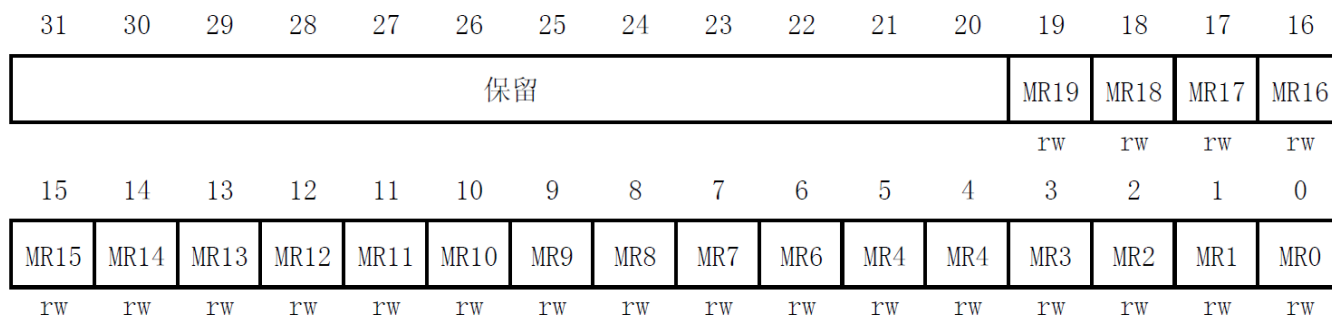
EXTI 寄存器描述

基地址: 0x4001 0400

❑ 中断屏蔽寄存器(EXTI_IMR) 偏移地址: 0x00 复位值: 0x0000 0000



❑ 事件屏蔽寄存器(EXTI_EMR) 偏移地址: 0x04 复位值: 0x0000 0000



- * 0: 屏蔽来自线x上的中断请求; 1: 开放来自线x上的中断请求。
- * 20~31位保留, 必须始终保持为复位状态(0)。
- * 必须以字(32位)的方式操作所有EXTI寄存器。

□ 上升沿触发选择寄存器(**EXTI_RTSR**)

偏移地址：0x08 复位值：0x0000 0000

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | | | | | | TR19 | TR18 | TR17 | TR16 |
| | | | | | | | | | | | | rW | rW | rW | rW |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TR15 | TR14 | TR13 | TR12 | TR11 | TR10 | TR9 | TR8 | TR7 | TR6 | TR5 | TR4 | TR3 | TR2 | TR1 | TR0 |
| rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW |

□ 下降沿触发选择寄存器(**EXTI_FTSR**)

偏移地址：0x0C 复位值：0x0000 0000

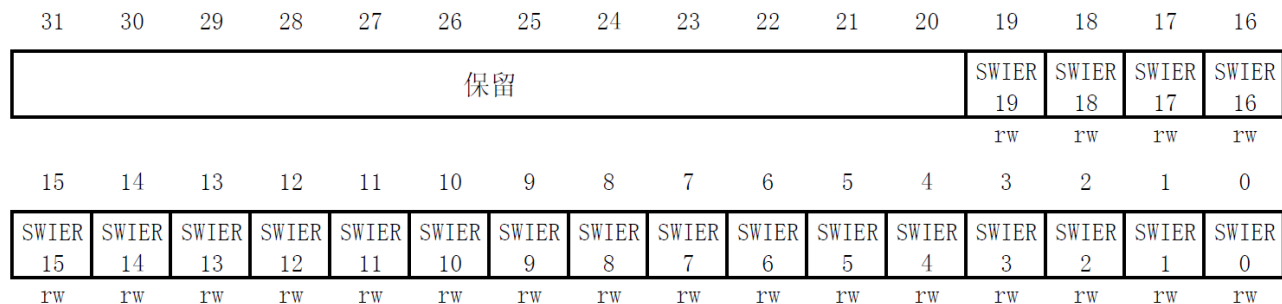
| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | | | | | | TR19 | TR18 | TR17 | TR16 |
| | | | | | | | | | | | | rW | rW | rW | rW |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TR15 | TR14 | TR13 | TR12 | TR11 | TR10 | TR9 | TR8 | TR7 | TR6 | TR5 | TR4 | TR3 | TR2 | TR1 | TR0 |
| rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW |

- * 0-禁止触发 1-允许触发
- * 外部唤醒线是边沿触发的，这些线上不能出现毛刺信号。
- * 在写**EXTI_RTSR**寄存器时，在外部中断线上的上升沿信号不能被识别，挂起位也不会被置位。在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

❑ 软件中断事件寄存器(EXTI_SWIER)

偏移地址: 0x10

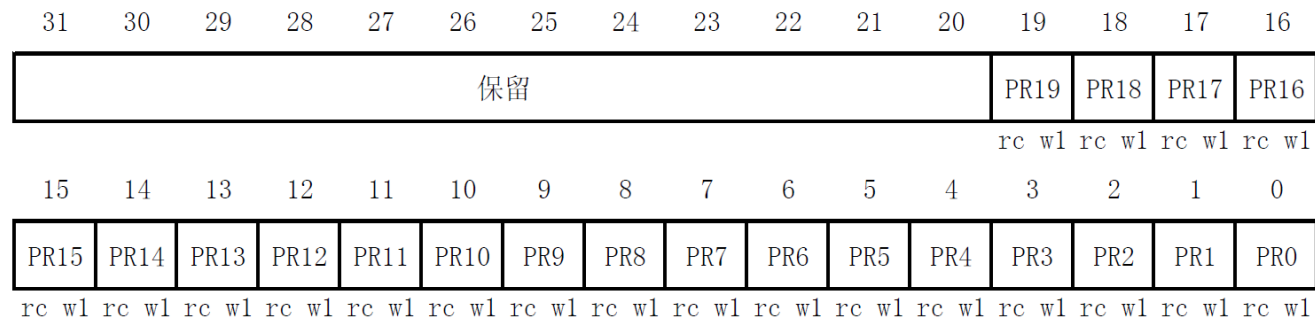
复位值: 0x0000 0000



- * 当该位为'0'时，写'1'将设置EXTI_PR中相应的挂起位。如果在EXTI_IMR和EXTI_EMR中允许产生该中断，则此时将产生一个中断。
- * 通过清除EXTI_PR的对应位(写入'1')，可以清除该位为'0'。

❑ 挂起寄存器(EXTI_PR)

偏移地址: 0x14 复位值: 0xFFFF XXXX



- * 0: 没有发生触发请求 1: 发生了选择的触发请求
- * 当在外部中断线上发生了选择的边沿事件，该位被置'1'。在该位中写入'1'可以清除它，也可以通过改变边沿检测的极性清除。

中断向量表(Interrupt Vector Table)

NVIC
register
map

| 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)联接到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 | EXTI0 | 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 |
| 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的全局中断 | 0x0000_0088 |
| 19 | 26 | 可设置 | USB_HP_CAN_TX | USB高优先级或CAN发送中断 | 0x0000_008C |
| 20 | 27 | 可设置 | USB_LP_CAN_RX0 | USB低优先级或CAN接收0中断 | 0x0000_0090 |
| 21 | 28 | 可设置 | CAN_RX1 | CAN接收1中断 | 0x0000_0094 |
| 22 | 29 | 可设置 | CAN_SCE | CAN SCE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 | I2C1_EV | I ² C1事件中断 | 0x0000_00BC |
| 32 | 39 | 可设置 | I2C1_ER | I ² C1错误中断 | 0x0000_00C0 |
| 33 | 40 | 可设置 | I2C2_EV | I ² C2事件中断 | 0x0000_00C4 |
| 34 | 41 | 可设置 | I2C2_ER | I ² C2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 |
| 42 | 49 | 可设置 | USB唤醒 | 连到EXTI的从USB待机唤醒中断 | 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全局中断 | 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 | 0x0000_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_5 | DMA2通道4和DMA2通道5全局中断 | 0x0000_012C |

位置即NVIC(Nested vectored interrupt controller) number for register map

□ NVIC register map

The table provides shows the NVIC register map and reset values. The base address of the main NVIC register block is 0xE000E100. The NVIC_STIR register is located in a separate block at 0xE000EF00.

NVIC register map and reset values

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x000 | NVIC_ISER0 | SETENA[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x004 | NVIC_ISER1 | SETENA[63:32] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x008 | NVIC_ISER2 | Reserved | | | | | | | | | | | | | | | | SETENA [80:64] | | | | | | | | | | | | | | | |
| | Reset Value | | | | | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x080 | NVIC_ICER0 | CLRENA[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x084 | NVIC_ICER1 | CLRENA[63:32] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x088 | NVIC_ICER2 | Reserved | | | | | | | | | | | | | | | | CLRENA [80:64] | | | | | | | | | | | | | | | |
| | Reset Value | | | | | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x100 | NVIC_ISPR0 | SETPEND[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x104 | NVIC_ISPR1 | SETPEND[63:32] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x108 | NVIC_ISPR2 | Reserved | | | | | | | | | | | | | | | | SETPEND [80:64] | | | | | | | | | | | | | | | |
| | Reset Value | | | | | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x180 | NVIC_ICPR0 | CLRPEND[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x184 | NVIC_ICPR1 | CLRPEND[63:32] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x188 | NVIC_ICPR2 | Reserved | | | | | | | | | | | | | | | | CLRPEND [80:64] | | | | | | | | | | | | | | | |
| | Reset Value | | | | | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x200 | NVIC_IABR0 | ACTIVE[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------------|-------------|---------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----------------|----|----|----|----|----|------------|---|-------|---|---|---|---|---|---|---|
| 0x204 | NVIC_IABR1 | ACTIVE[63:32] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x208 | NVIC_IABR2 | Reserved | | | | | | | | | | | | | | | | ACTIVE [80:64] | | | | | | | | | | | | | | | |
| | Reset Value | | | | | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x300 | NVIC_IPR0 | IP[3] | | | | | | | | IP[2] | | | | | | | | IP[1] | | | | | | | | IP[0] | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| : | : | : | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x320 | NVIC_IPR20 | Reserved | | | | | | | | | | | | | | | | | | | | | | IP[80] | | | | | | | | | |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| SCB registers | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0xE00 | NVIC_STIR | Reserved | | | | | | | | | | | | | | | | | | | | | | INTID[8:0] | | | | | | | | | |
| | Reset Value | Reserved | | | | | | | | | | | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

□ Interrupt set-enable registers (NVIC_ISERx)

Address offset: 0x00 - 0x0B Reset value: 0x00000000

Required privilege: Privileged

| | | | | | | | | | | | | | | | |
|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SETENA[31:16] | | | | | | | | | | | | | | | |
| rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SETENA[15:0] | | | | | | | | | | | | | | | |
| rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs | rs |

Bits 31:0 SETENA[31:0]: Interrupt set-enable bits.

Write: 0: No effect 1: Enable interrupt

Read: 0: Interrupt disabled 1: Interrupt enabled.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

❑ Interrupt clear-enable registers (NVIC_ICERx)

Address offset: 0x00 - 0x0B Reset value: 0x0000 0000 Required privilege: Privileged

The ICER0-ICER2 registers disable interrupts, and show which interrupts are enabled.

| | | | | | | | | | | | | | | | |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CLRENA[31:16] | | | | | | | | | | | | | | | |
| rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLRENA[15:0] | | | | | | | | | | | | | | | |
| rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

Bits 31:0 CLRENA[31:0]: Interrupt clear-enable bits.

Write: 0: No effect 1: Disable interrupt

Read: 0: Interrupt disabled 1: Interrupt enabled.

- ❑ Interrupt set-pending registers (NVIC_ISPRx)

Address offset: 0x00 - 0x0B Reset value: 0x0000 0000 Required privilege: Privileged

The ISPR0-ISPR2 registers force interrupts into the pending state, and show which interrupts are pending.

[illegible]

Bits 31:0 SETPEND[31:0]: Interrupt set-pending bits

Write: 0: No effect 1: Changes interrupt state to pending

Read: 0: Interrupt is not pending 1: Interrupt is pending

Writing 1 to the ISPR bit corresponding to an interrupt that is pending:

- has no effect.

Writing 1 to the ISPR bit corresponding to a disabled interrupt:

- sets the state of that interrupt to pending.

- ❑ Interrupt clear-pending registers (NVIC_ICPRx)

Address offset: 0x00 - 0x0B Reset value: 0x0000 0000 Required privilege: Privileged

The ICPR0-ICPR2 registers remove the pending state from interrupts, and show which interrupts are pending.

[illegible]

Bits 31:0 CLRPEND[31:0]: Interrupt clear-pending bits

[illegible]

Read: 0: Interrupt is not pending 1: Interrupt is pending

Writing 1 to an ICPR bit does not affect the active state of the corresponding interrupt.

通过中断实现led灯闪烁

本项目通过中断实现led灯闪烁(每秒一次)

; 本项目通过中断实现led灯闪烁, 接线方法:

; 把一个led二极管的正极引脚(更长的那根)

; 通过一个电阻接到实验板的A2引脚,

; 再把led二极管的正极引脚(更短的那根)接到实验板的地线(G)。

LED2ON EQU 0x00000004 ; GPIO_BSRR: bit2=1-PA2 on

LED2OFF EQU 0x00040000 ; GPIO_BSRR: bit18=1-PA2 off

GPIOCFG_A EQU 0x0300 ; PA.2--推挽输出(50MHz)

GPIOA_BASE EQU 0x40010800 ; GPIOA基地址

GPIOA_CRL EQU GPIOA_BASE+0x00 ; 低配置寄存器

GPIOA_CRH EQU GPIOA_BASE+0x04 ; 高配置寄存器

GPIOA_IDR EQU GPIOA_BASE+0x08 ; 输入数据寄存器

GPIOA_ODR EQU GPIOA_BASE+0x0c ; 输出数据寄存器

GPIOA_BSRR EQU GPIOA_BASE+0x10 ; 位端口置位/清零寄存器

GPIOA_BRR EQU GPIOA_BASE+0x14 ; 位端口清零寄存器

TIM2 EQU 0x40000000 ; TIM2基地址

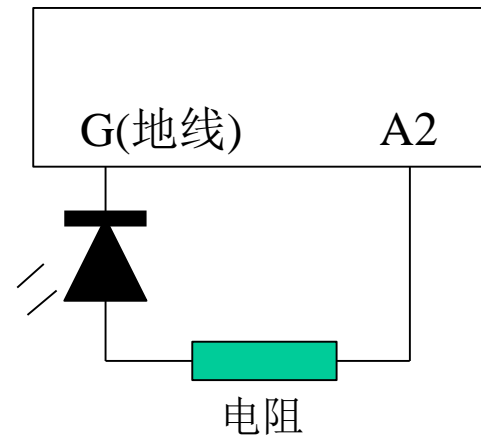
TIM2_ARR EQU TIM2+0x2c ; 自动装载寄存器

TIM2_PSC EQU TIM2+0x28 ; 预分频器

TIM2_DIER EQU TIM2+0x0c ; DMA/中断使能寄存器

TIM2_CR1 EQU TIM2+0x00 ; 控制寄存器1

TIM2_SR EQU TIM2+0x10 ; 状态寄存器



```

RCC_BASE EQU 0x40021000
RCC_CR EQU RCC_BASE+0x00
RCC_CFGR EQU RCC_BASE+0x04
RCC_CIR EQU RCC_BASE+0x08
RCC_CR_HSERDY EQU 0x00020000

RCC_APB1ENR EQU 0x4002101c
TIM2EN EQU 0x00000001 ;TIM2使能位

RCC_APB2ENR EQU 0x40021018
GPIOAEN EQU 0x00000004 ;GPIOA使能位

ISER_TIM2 EQU 0xe000e100 ; NVIC_ISER0
TIM2_ITEN EQU 0x10000000 ; bit28(TIM2)

STACK_TOP EQU 0X20002000
AREA RESET, CODE, READONLY ; AREA不能顶格写
DCD STACK_TOP ; MSP主堆栈指针
DCD START ; 复位, PC初始值
SPACE 0xB0-8;
DCD TIM2_IRQHandler
ENTRY ; 指示开始执行
START ; 所有的标号必须顶格写
BL RCC_CONFIG_72MHZ

; 配置RCC的APB1使能寄存器, 启动TIM2时钟
LDR R1, =RCC_APB1ENR
LDR R2, =TIM2EN
STR R2, [R1]

; 配置RCC的APB2使能寄存器, 启动GPIOA
LDR R1, =RCC_APB2ENR
LDR R2, =GPIOAEN
STR R2, [R1]

```

```

; 配置寄存器: 设置GPIOA.2 (LED2)--PA.2 推挽输出, 50MHz
MOV R0, #GPIOCFG_A
LDR R1, =GPIOA_CRL
STR R0, [R1]

; 设置定时器Tim2的重装载寄存器
MOV R0, #(10000-1)
LDR R1, =TIM2_ARR
STR R0, [R1]

; 设置定时器Tim2的分频器
MOV R0, #(7200-1)
LDR R1, =TIM2_PSC
STR R0, [R1]

; 设置定时器Tim2的DMA/中断允许寄存器
MOV R0, #1
LDR R1, =TIM2_DIER
STR R0, [R1]

; 设置NVIC的中断设置允许寄存器(ISER)
MOV R0, #TIM2_ITEN ; 第28位
LDR R1, =ISER_TIM2 ; 0xE000E100 (NVIC_ISER0)
STR R0, [R1]

; 设置定时器Tim2的配置寄存器, 启动计数
MOV R0, #1
LDR R1, =TIM2_CR1
STR R0, [R1]
MOV R3, #0

LOOP
B LOOP

```

```

;*****
;;RCC 时钟配置 HCLK=72MHz=HSE*9
;;PCLK2=HCLK PCLK1=HCLK/2
;*****
RCC_CONFIG_72MHZ
LDR R1,=0X40021000 ;RCC_CR
LDR R0,[R1]
LDR R2,=0X00010000 ;HSEON
ORR R0,R2
STR R0,[R1]
WAIT_HSE_RDY
LDR R2,=0X00020000 ;HSERDY
LDR R0,[R1]
ANDS R0,R2
CMP R0,#0
BEQ WAIT_HSE_RDY
LDR R1,=0X40022000 ;FLASH_ACR
MOV R0,#0X12
STR R0,[R1]
LDR R1,=0X40021004 ;RCC_CFGR时钟配置寄存器
LDR R0,[R1]
; PLL倍频系数,PCLK2,PCLK1分频设置
; HSE 9倍频PCLK2=HCLK,PCLK1=HCLK/2
; HCLK=72MHz 0x001D0400
; 64MHz 0x00190400 48MHz 0x00110400
; 32MHz 0x00090400 24MHz 0x00050400
; 16MHz 0x00010400
LDR R2,=0x001D0400
ORR R0,R2
STR R0,[R1]
LDR R1,=0X40021000 ;RCC_CR
LDR R0,[R1]
LDR R2,=0X01000000 ;PLLON
ORR R0,R2
STR R0,[R1]
WAIT_PLL_RDY
LDR R2,=0X02000000 ;PLLRDY

```

```

LDR R0,[R1]
ANDS R0,R2
CMP R0,#0
BEQ WAIT_PLL_RDY
LDR R1,=0X40021004 ;RCC_CFGR
LDR R0,[R1]
MOV R2,#0X02
ORR R0,R2
STR R0,[R1]
WAIT_HCLK_USEPLL
LDR R0,[R1]
ANDS R0,#0X08
CMP R0,#0X08
BNE WAIT_HCLK_USEPLL
BX LR
TIM2_IRQHandler PROC
PUSH {R0,R1,R2,LR}
LDR R0,=GPIOA_BSRR
LDR R1,=GPIOA_ODR
LDR R2,[R1] ; 读入PA.2的输出
TST R2,#0x4
BEQ TURNON ; 如果为0, 则灯亮
MOV R2,#LED2OFF ; 否则, 灯灭
STR R2,[R0] ; 将PA.2输出低电平
B EX
TURNON
LDR R2,=LED2ON ; 将PA.2输出高电平
STR R2,[R0]
EX
LDR R1,=TIM2_SR
MOV R2,#0 ; 清除更新事件状态位
STR R2,[R1]
POP {R0,R1,R2,PC}
ENDP
END

```

外部中断例子

本工程实现按一次按钮（接通）就改变led灯的亮灭，即亮变灭或灭变亮。

; 本工程实现按一次按钮（接通）就改变led灯的亮灭：

; 按钮：引脚B0通过一个电阻和按钮接电源引脚(3.3或v3)，

; led： 引脚B1通过电阻连接led再接地，接地的一端为二极管短脚（负极）

PULL0DOWN EQU 0x00010000

; ODR0=0----PX0输入下拉 (默认) 这里用于PB0

PULL0UP EQU 0x00000001

; ODR0=1----PX0输入上拉

LED2ON EQU 0x00000002

; GPIOX_BSRR: bit1=1---PX1 on

LED2OFF EQU 0x00020000

; GPIOX_BSRR: bit17=1---PX1 off

CFGB EQU 0x0038

; GPIOB配置：PB0--下拉输入；PB1--推挽输出(50MHz)；

GPIOB_BASE EQU 0x40010c00

; GPIOB基地址

GPIOB_CRL EQU GPIOB_BASE+0x00

; GPIOB低配置寄存器

GPIOB_CRH EQU GPIOB_BASE+0x04

; GPIOB高配置寄存器

GPIOB_IDR EQU GPIOB_BASE+0x08

; GPIOB输入数据寄存器

GPIOB_ODR EQU GPIOB_BASE+0x0c

; GPIOB输出数据寄存器

GPIOB_BSRR EQU GPIOB_BASE+0x10

; GPIOB位端口置位/清零寄存器

RCC_APB2ENR EQU 0x40021018

; RCC时钟APB2使能寄存器

GIOPBEN EQU 0x00000008

; RCC时钟GPIOB使能位

AFIOEN EQU 0x00000001

; AFIO时钟使能位

APB2ENALL EQU GIOPBEN :OR: AFIOEN

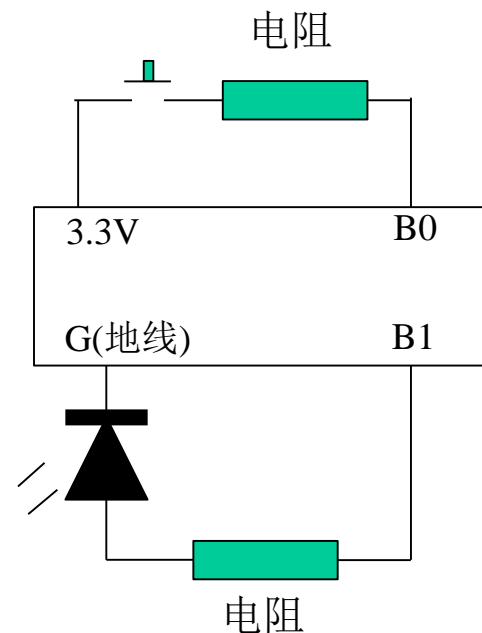
AFIO_BASE EQU 0x40010000

AFIO_EXTICR1 EQU AFIO_BASE+0X08

; 外部中断(EXTI)配置寄存器1

AFIO_EXTIO_PB EQU 0x1

; EXTI0选择PB0作为输入



```

EXTI_BASE EQU 0x40010400
EXTI_IMR EQU EXTI_BASE+0x00 ; EXTI中断屏蔽寄存器
EXTI_EMR EQU EXTI_BASE+0x04 ; EXTI事件屏蔽寄存器
EXTI_PR EQU EXTI_BASE+0x14 ; EXTI挂起寄存器
EXTI_RTSR EQU EXTI_BASE+0x08 ; EXTI上升沿触发选择寄存器
RTSR_EXTI0 EQU 1 ; EXTI0选择上升沿触发

```

```

NVIC_ISER0 EQU 0xe000e100 ; NVIC中断设置允许寄存器
EXTI0_ITEN EQU 0x40 ; 允许EXTI0中断

```

```

STACK_TOP EQU 0x20002000
AREA RESET, CODE, READONLY ; AREA不能顶格写
DCD STACK_TOP ; MSP主堆栈指针
DCD START ; 复位, PC初始值
SPACE 0x58-8;
DCD EXTI0_IRQHandler
ENTRY ; 指示开始执行
START ; 所有的标号必须顶格写

```

; 设置RCCAPB2使能寄存器: 打开GPIOA、GPIOB和AFIO部件
; 启动它们进入工作状态

```

LDR R1, =RCC_APB2ENR
LDR R2, =APB2ENALL
STR R2, [R1]

```

; 设置AFIO的EXTI配置寄存器: 把PB0连至EXTI0

```

MOV R0, #AFIO_EXTI0_PB
LDR R1, =AFIO_EXTICR1
STR R0, [R1]

```

; 设置EXTI中断屏蔽寄存器: 允许EXTI0中断

```

MOV R0, #1
LDR R1, =EXTI_IMR
STR R0, [R1]

```

; 设置事件屏蔽寄存器: 允许EXTI0事件中断

```

MOV R0, #1
LDR R1, =EXTI_EMR
STR R0, [R1]

```

; 设置EXTI上升沿触发设置寄存器: EXTI0采用上升沿触发

```

MOV R0, #RTSR_EXTI0
LDR R1, =EXTI_RTSR
STR R0, [R1]

```

; 设置NVIC的中断设置允许寄存器(ISER): 允许EXTI0

```

MOV R0, #EXTI0_ITEN ;第6位
LDR R1, =NVIC_ISER0
STR R0, [R1]

```

; 设置GPIOB低配置寄存器: PB.0 下拉输入

```

MOV R0, #CFGB
LDR R1, =GPIOB_CRL
STR R0, [R1]
LDR R1, =GPIOB_BSRR
LDR R2, =PULL0DOWN
STR R2, [R1]

```

; 将PB.1输出高电平

```

LDR R1, =GPIOB_BSRR
LDR R2, =LED2ON
STR R2, [R1]

```

```

LOOP
B LOOP

```

```

EXTI0_IRQHandler PROC
    PUSH {R0,R1,R2,LR}

    LDR R1,=GPIOB_ODR
    LDR R2,[R1]          ; 读入PB.1的输出
    TST R2,#0x2
    BEQ TURNON           ; 如果为0, 则灯亮
    LDR R1,=GPIOB_BSRR
    MOV R2,#LED2OFF      ; 否则, 灯灭
    STR R2,[R1]          ; 将PB.1输出低电平
    B EX
TURNON
    LDR R1,=GPIOB_BSRR
    LDR R2,=LED2ON       ; 将PB.1输出高电平
    STR R2,[R1]
EX
    LDR R1,=EXTI_PR
    MOV R2,#1            ; 清除EXTI0的触发请求
    STR R2,[R1]

    POP {R0,R1,R2,PC}
ENDP
NOP
END

```

附录1、 NVIC number for register map

```

NonMaskableInt_IRQn    = -14, /*!< 2 Non Maskable Interrupt */
MemoryManagement_IRQn  = -12, /*!< 4 Cortex-M3 Memory Management Interrupt */
BusFault_IRQn          = -11, /*!< 5 Cortex-M3 Bus Fault Interrupt */
UsageFault_IRQn        = -10, /*!< 6 Cortex-M3 Usage Fault Interrupt */
SVCall_IRQn            = -5,  /*!< 11 Cortex-M3 SV Call Interrupt */
DebugMonitor_IRQn      = -4,  /*!< 12 Cortex-M3 Debug Monitor Interrupt */
PendSV_IRQn            = -2,  /*!< 14 Cortex-M3 Pend SV Interrupt */
SysTick_IRQn           = -1,  /*!< 15 Cortex-M3 System Tick Interrupt */
/***** Cortex-M3 Processor Exceptions Numbers ( NVIC number for register map ) *****/
NonMaskableInt_IRQn    = -14, /*!< 2 Non Maskable Interrupt */
MemoryManagement_IRQn  = -12, /*!< 4 Cortex-M3 Memory Management Interrupt */
BusFault_IRQn          = -11, /*!< 5 Cortex-M3 Bus Fault Interrupt */
UsageFault_IRQn        = -10, /*!< 6 Cortex-M3 Usage Fault Interrupt */
SVCall_IRQn            = -5,  /*!< 11 Cortex-M3 SV Call Interrupt */
DebugMonitor_IRQn      = -4,  /*!< 12 Cortex-M3 Debug Monitor Interrupt */
PendSV_IRQn            = -2,  /*!< 14 Cortex-M3 Pend SV Interrupt */
SysTick_IRQn           = -1,  /*!< 15 Cortex-M3 System Tick Interrupt */

/***** STM32 specific Interrupt Numbers *****/
WWDG_IRQn              = 0,    /*!< Window WatchDog Interrupt */
PVD_IRQn               = 1,    /*!< PVD through EXTI Line detection Interrupt */
TAMPER_IRQn            = 2,    /*!< Tamper Interrupt */
RTC_IRQn               = 3,    /*!< RTC global Interrupt */
FLASH_IRQn             = 4,    /*!< FLASH global Interrupt */
RCC_IRQn               = 5,    /*!< RCC global Interrupt */
EXTI0_IRQn             = 6,    /*!< EXTI Line0 Interrupt */
EXTI1_IRQn             = 7,    /*!< EXTI Line1 Interrupt */
EXTI2_IRQn             = 8,    /*!< EXTI Line2 Interrupt */
EXTI3_IRQn             = 9,    /*!< EXTI Line3 Interrupt */
EXTI4_IRQn             = 10,   /*!< EXTI Line4 Interrupt */
DMA1_Channel1_IRQn     = 11,   /*!< DMA1 Channel 1 global Interrupt */
DMA1_Channel2_IRQn     = 12,   /*!< DMA1 Channel 2 global Interrupt */
DMA1_Channel3_IRQn     = 13,   /*!< DMA1 Channel 3 global Interrupt */
DMA1_Channel4_IRQn     = 14,   /*!< DMA1 Channel 4 global Interrupt */
DMA1_Channel5_IRQn     = 15,   /*!< DMA1 Channel 5 global Interrupt */
DMA1_Channel6_IRQn     = 16,   /*!< DMA1 Channel 6 global Interrupt */
DMA1_Channel7_IRQn     = 17,   /*!< DMA1 Channel 7 global Interrupt */
ADC1_2_IRQn            = 18,   /*!< ADC1 et ADC2 global Interrupt */
USB_HP_CAN1_TX_IRQn    = 19,   /*!< USB High Priority or CAN1 TX Interrupts */
USB_LP_CAN1_RX0_IRQn   = 20,   /*!< USB Low Priority or CAN1 RX0 Interrupts */
CAN1_RX1_IRQn          = 21,   /*!< CAN1 RX1 Interrupt */
CAN1_SCE_IRQn          = 22,   /*!< CAN1 SCE Interrupt */
EXTI9_5_IRQn           = 23,   /*!< External Line[9:5] Interrupts

```



```

TIM1_BRK_IRQn      = 24, /*!< TIM1 Break Interrupt          */
TIM1_UP_IRQn       = 25, /*!< TIM1 Update Interrupt          */
TIM1_TRG_COM_IRQn  = 26, /*!< TIM1 Trigger and Commutation Interrupt */
TIM1_CC_IRQn       = 27, /*!< TIM1 Capture Compare Interrupt    */
TIM2_IRQn          = 28, /*!< TIM2 global Interrupt          */
TIM3_IRQn          = 29, /*!< TIM3 global Interrupt          */
#ifdef STM32F10X_LD
TIM4_IRQn          = 30, /*!< TIM4 global Interrupt          */
#endif
I2C1_EV_IRQn       = 31, /*!< I2C1 Event Interrupt          */
I2C1_ER_IRQn       = 32, /*!< I2C1 Error Interrupt          */
#ifdef STM32F10X_LD
I2C2_EV_IRQn       = 33, /*!< I2C2 Event Interrupt          */
I2C2_ER_IRQn       = 34, /*!< I2C2 Error Interrupt          */
#endif
SPI1_IRQn          = 35, /*!< SPI1 global Interrupt          */
SPI2_IRQn          = 36, /*!< SPI2 global Interrupt          */
USART1_IRQn        = 37, /*!< USART1 global Interrupt        */
USART2_IRQn        = 38, /*!< USART2 global Interrupt        */
#ifdef STM32F10X_LD
USART3_IRQn        = 39, /*!< USART3 global Interrupt        */
#endif
EXTI15_10_IRQn     = 40, /*!< EXTI Line[15:10] Interrupts      */
RTCArm_IRQn        = 41, /*!< RTC Alarm through EXTI Line Interrupt */
USBWakeUp_IRQn     = 42, /*!< USB WakeUp from suspend through EXTI Line Interrupt */
#ifdef STM32F10X_HD
TIM8_BRK_IRQn      = 43, /*!< TIM8 Break Interrupt          */
TIM8_UP_IRQn       = 44, /*!< TIM8 Update Interrupt          */
TIM8_TRG_COM_IRQn  = 45, /*!< TIM8 Trigger and Commutation Interrupt */
TIM8_CC_IRQn       = 46, /*!< TIM8 Capture Compare Interrupt    */
ADC3_IRQn          = 47, /*!< ADC3 global Interrupt          */
FSMC_IRQn          = 48, /*!< FSMC global Interrupt          */
SDIO_IRQn          = 49, /*!< SDIO global Interrupt          */
TIM5_IRQn          = 50, /*!< TIM5 global Interrupt          */
SPI3_IRQn          = 51, /*!< SPI3 global Interrupt          */
UART4_IRQn         = 52, /*!< UART4 global Interrupt          */
UART5_IRQn         = 53, /*!< UART5 global Interrupt          */
TIM6_IRQn          = 54, /*!< TIM6 global Interrupt          */
TIM7_IRQn          = 55, /*!< TIM7 global Interrupt          */
DMA2_Channel1_IRQn = 56, /*!< DMA2 Channel 1 global Interrupt    */
DMA2_Channel2_IRQn = 57, /*!< DMA2 Channel 2 global Interrupt    */
DMA2_Channel3_IRQn = 58, /*!< DMA2 Channel 3 global Interrupt    */
DMA2_Channel4_5_IRQn = 59 /*!< DMA2 Channel 4 and Channel 5 global Interrupt */
#endif

```

附录2、APB使能寄存器

□ APB1外设时钟使能寄存器(RCC_APB1ENR)

偏移地址：0x1C 复位值：0x0000 0000

RCC基地址：0X40021000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------|---------|--------|---------|----|--------|----|--------|---------|---------|----------|----------|-----------|-----------|---------|---------|
| 保留 | DACEN | PWR EN | BKP EN | 保留 | CAN EN | 保留 | USB EN | I2C2 EN | I2C1 EN | UART5 EN | UART4 EN | USART3 EN | USART2 EN | 保留 | |
| | rw | rw | rw | | rw | | rw | rw | rw | rw | rw | rw | rw | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPI3 EN | SPI2 EN | 保留 | WWDG EN | 保留 | | | | | | TIM7 EN | TIM6 EN | TIM5 EN | TIM4 EN | TIM3 EN | TIM2 EN |
| rw | rw | | rw | | | | | | | rw | rw | rw | rw | rw | rw |

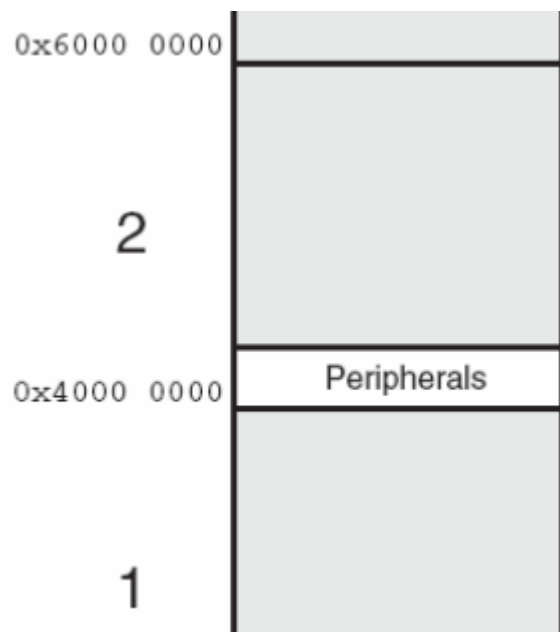
□ APB2外设时钟使能寄存器(RCC_APB2ENR)

偏移地址：0x18 复位值：0x0000 0000

RCC基地址：0X40021000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------|-----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----|---------|
| 保留 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC3 EN | USART1 EN | TIM8 EN | SPI1 EN | TIM1 EN | ADC2 EN | ADC1 EN | IOPG EN | IOPF EN | IOPE EN | IOPD EN | IOPC EN | IOPB EN | IOPA EN | 保留 | AFIO EN |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw |

附录3、基地址



| | |
|-------------|---------------------------------|
| 0x4001 0400 | AFIO |
| 0x4001 0000 | reserved |
| 0x4000 7400 | PWR |
| 0x4000 7000 | BKP |
| 0x4000 6C00 | reserved |
| 0x4000 6800 | bxCAN |
| 0x4000 6400 | shared 512 byte USB/CAN SRAM |
| 0x4000 6000 | USB Registers |
| 0x4000 5C00 | I2C2 |
| 0x4000 5800 | I2C1 |
| 0x4000 5400 | reserved |
| 0x4000 4C00 | USART3 |
| 0x4000 4800 | USART2 |
| 0x4000 4400 | reserved |
| 0x4000 3C00 | SPI2 |
| 0x4000 3800 | reserved |
| 0x4000 3400 | IWDG |
| 0x4000 3000 | WWDG |
| 0x4000 2C00 | RTC |
| 0x4000 2800 | reserved |
| 0x4000 0C00 | TIM4 |
| 0x4000 0800 | TIM3 |
| 0x4000 0400 | TIM2 |
| 0x4000 0000 | |

| | |
|-------------|-----------------|
| 0x4002 3400 | CRC |
| 0x4002 3000 | reserved |
| 0x4002 2400 | Flash Interface |
| 0x4002 2000 | reserved |
| 0x4002 1400 | RCC |
| 0x4002 1000 | reserved |
| 0x4002 0400 | DMA |
| 0x4002 0000 | reserved |
| 0x4001 3C00 | USART1 |
| 0x4001 3800 | reserved |
| 0x4001 3400 | SPI1 |
| 0x4001 3000 | TIM1 |
| 0x4001 2C00 | ADC2 |
| 0x4001 2800 | ADC1 |
| 0x4001 2400 | reserved |
| 0x4001 1C00 | Port E |
| 0x4001 1800 | Port D |
| 0x4001 1400 | Port C |
| 0x4001 1000 | Port B |
| 0x4001 0C00 | Port A |
| 0x4001 0800 | EXTI |
| 0x4001 0400 | |

| 起始地址 | 外设 | 总线 |
|--|---------------------|------|
| 0x5000 0000 – 0x5003 FFFF | USB OTG 全速 | AHB |
| 0x4003 0000 – 0x4FFF FFFF | 保留 | |
| 0x4002 8000 – 0x4002 9FFF | 以太网 | |
| 0x4002 3400 - 0x4002 3FFF | 保留 | |
| 0x4002 3000 - 0x4002 33FF | CRC | AHB |
| 0x4002 2000 - 0x4002 23FF | 闪存存储器接口 | |
| 0x4002 1400 - 0x4002 1FFF | 保留 | |
| 0x4002 1000 - 0x4002 13FF | 复位和时钟控制(RCC) | |
| 0x4002 0800 - 0x4002 0FFF | 保留 | |
| 0x4002 0400 - 0x4002 07FF | DMA2 | |
| 0x4002 0000 - 0x4002 03FF | DMA1 | |
| 0x4001 8400 - 0x4001 7FFF | 保留 | |
| 0x4001 8000 - 0x4001 83FF | SDIO | |
| 0x4001 4000 - 0x4001 7FFF | 保留 | |
| 0x4001 3C00 - 0x4001 3FFF | ADC3 | |
| 0x4001 3800 - 0x4001 3BFF | USART1 | |
| 0x4001 3400 - 0x4001 37FF | TIM8定时器 | |
| 0x4001 3000 - 0x4001 33FF | SPI1 | |
| 0x4001 2C00 - 0x4001 2FFF | TIM1定时器 | APB2 |
| 0x4001 2800 - 0x4001 2BFF | ADC2 | |
| 0x4001 2400 - 0x4001 27FF | ADC1 | |
| 0x4001 2000 - 0x4001 23FF | GPIO端口G | |
| 0x4001 2000 - 0x4001 23FF | GPIO端口F | |
| 0x4001 1800 - 0x4001 1BFF | GPIO端口E | |
| 0x4001 1400 - 0x4001 17FF | GPIO端口D | |
| 0x4001 1000 - 0x4001 13FF | GPIO端口C | |
| 0x4001 0C00 - 0x4001 0FFF | GPIO端口B | |
| 0x4001 0800 - 0x4001 0BFF | GPIO端口A | |
| 0x4001 0400 - 0x4001 07FF | EXTI | |
| 0x4001 0000 - 0x4001 03FF | AFIO | |
| 0x4000 7800 - 0x4000FFFF | 保留 | APB1 |
| 0x4000 7400 - 0x4000 77FF | DAC | |
| 0x4000 7000 - 0x4000 73FF | 电源控制(PWR) | |
| 0x4000 6C00 - 0x4000 6FFF | 后备寄存器(BKP) | |
| 0x4000 6800 - 0x4000 6BFF | bxCAN2 | |
| 0x4000 6400 - 0x4000 67FF | bxCAN1 | |
| 0x4000 6000 ⁽¹⁾ - 0x4000 63FF | USB/CAN共享的512字节SRAM | |

| | | |
|---------------------------|-------------|------|
| 0x4000 5C00 - 0x4000 5FFF | USB全速设备寄存器 | APB1 |
| 0x4000 5800 - 0x4000 5BFF | I2C2 | |
| 0x4000 5400 - 0x4000 57FF | I2C1 | |
| 0x4000 5000 - 0x4000 53FF | UART5 | |
| 0x4000 4C00 - 0x4000 4FFF | UART4 | |
| 0x4000 4800 - 0x4000 4BFF | USART3 | |
| 0x4000 4400 - 0x4000 47FF | USART2 | |
| 0x4000 4000 - 0x4000 3FFF | 保留 | |
| 0x4000 3C00 - 0x4000 3FFF | SPI3/I2S3 | |
| 0x4000 3800 - 0x4000 3BFF | SPI2/I2S3 | |
| 0x4000 3400 - 0x4000 37FF | 保留 | |
| 0x4000 3000 - 0x4000 33FF | 独立看门狗(IWDG) | |
| 0x4000 2C00 - 0x4000 2FFF | 窗口看门狗(WWDG) | |
| 0x4000 2800 - 0x4000 2BFF | RTC | |
| 0x4000 1800 - 0x4000 27FF | 保留 | |
| 0x4000 1400 - 0x4000 17FF | TIM7定时器 | |
| 0x4000 1000 - 0x4000 13FF | TIM6定时器 | |
| 0x4000 0C00 - 0x4000 0FFF | TIM5定时器 | |
| 0x4000 0800 - 0x4000 0BFF | TIM4定时器 | |
| 0x4000 0400 - 0x4000 07FF | TIM3定时器 | |
| 0x4000 0000 - 0x4000 03FF | TIM2定时器 | |