

FreeRTOS学习笔记（二）

FreeRTOSConfig.h文件

1. “INCLUDE_”开始的宏：用来使能和除能系统中相关的API函数，起到裁剪和配置的作用。
2. “config”开始的宏：配置和裁剪FreeRTOS（需要仔细阅读文档，不同的宏对应不同的功能）

FreeRTOS中断配置和临界段

1. cortex-M内核的MCU提供了一个用于中断管理的嵌套向量中断（NVIC），最多支持240个中断请求，1个不可屏蔽中断，1个滴答中断和多个系统中断。（CM3的中断架构还需要继续了解）
2. FreeRTOS中断配置的宏（在FreeRTOSConfig.h文件中），用来配置芯片的中断优先级，在portmacro.h文件中对PendSV和SysTick中断的优先级进行设置。
3. `configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY` 这是一个FreeRTOS中断管理的一个优先级的阈值，也就是高于这个值的优先级不归FreeRTOS管理。
4. `configMAX_SYSCALL_INTERRUPT_PRIORITY` 这个宏就是设置中断是否可以安全调用FreeRTOS的API，低于这个值就是则可以安全使用系统的API，高于这个值FreeRTOS是不能禁止的，中断服务函数也不能调用FreeRTOS的API

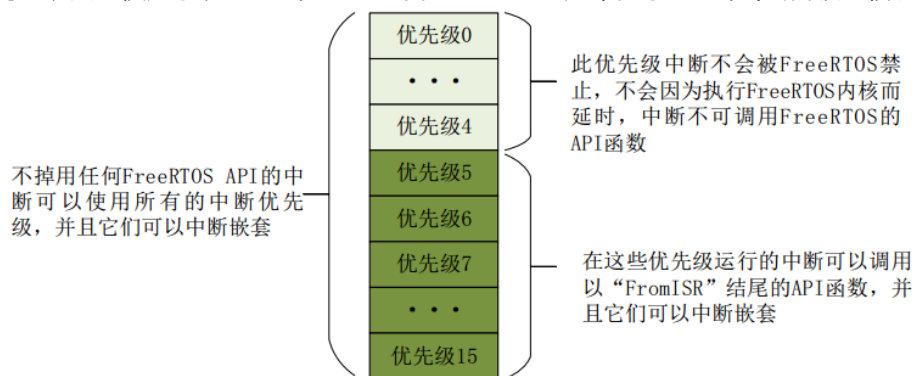


图 4.2.5.1 中断配置结果图

函数。

5. `portDISABLE_INTERRUPTS ()` ; `portENABLE_INTERRUPTS ()` ; 这两个函数用作开关中断。根据上述那个阈值来进行开关。（这里关闭中断之后，注意不要使用滴答定时器中断来进行延时）
6. 临界段代码 `taskENTER_CRITICAL ()` , `taskEXIT_CRITICAL ()` , `taskEXIT_CRITICAL_FROM_ISR ()` , `taskENTER_CRITICAL_FROM_ISR ()` 用这四个函数来进入临界段，前两个是任务级的临界代码保护，后两个是中断级的临界段代码保护。