

临界区保护及调度器的挂起和恢复

1, 临界段代码保护简介（熟悉）

什么是临界段：临界段代码也叫做临界区，是指那些必须完整运行，不能被打断的代码段

适用场合

1, 外设：需严格按照时序初始化的外设：IIC、SPI等等

2, 系统：系统自身需求

3, 用户：用户需求

2, 临界段代码保护函数介绍（掌握）

FreeRTOS 在进入临界段代码的时候需要关闭中断，当处理完临界段代码以后再打开中断

有关于临界区的API函数

任务级进入临界段：taskENTER_CRITICAL()

任务级退出临界段：taskEXIT_CRITICAL()

中断级进入临界段：taskENTER_CRITICAL_FROM_ISR()

中断级退出临界段：taskEXIT_CRITICAL_FROM_ISR()

特点

1、成对使用

2、支持嵌套（与单纯的开关中断最大的区别）

3、尽量保持临界段耗时短

强悍！临界区是直接屏蔽了中断，系统任务调度靠中断，ISR也靠中断

3, 任务调度器的挂起和恢复（熟悉）

挂起任务调度器， 只单纯挂起调度器使任务无法调度，但中断正常

调度器挂起和恢复相关API函数

挂起任务调度器：vTaskSuspendAll()

恢复任务调度器：xTaskResumeAll()

特点

1、与临界区不一样的是，挂起任务调度器，未关闭中断；

2、它仅仅是防止了任务之间的资源争夺，中断照样可以直接响应；

3、挂起调度器的方式，适用于临界区位于任务与任务之间；既不用去延时中断，又可以做到临界区的安全

调度器挂起和恢复API函数

挂起任务调度器：vTaskSuspendAll() 调用一次挂起调度器，该变量uxSchedulerSuspended就加一 变量uxSchedulerSuspended的值，将会影响Systick触发PendSV中断，即影响任务调度

恢复任务调度器：xTaskResumeAll() 调用一次恢复调度器，该变量uxSchedulerSuspended就减一 如果等于0，则允许调度

1、将所有在xPendingReadyList中的任务移到对应的就绪链表中

2、移除等待就绪列表中的列表项，恢复至就绪列表，直到xPendingReadyList列表为空