

FreeRTOS中断管理

1, 什么是中断? (了解)

简介: 让CPU打断正常程序的运行, 转而紧急处理的事件(程序), 就叫中断

中断执行机制, 可简单概括为三步:

1, 中断请求: 外设产生中断请求 (GPIO外部中断、定时器中断等)

2, 响应中断: CPU停止执行当前程序, 转而去执行中断处理程序 (ISR)

3, 退出中断: 执行完毕, 返回被打断的程序处, 继续往下执行

2, 中断优先级分组设置 (熟悉)

ARM Cortex-M 使用了 8 位宽的寄存器来配置中断的优先等级, 这个寄存器就是中断优先级配置寄存器, 所以中断优先级配置范围在0~255

STM32, 只用了中断优先级配置寄存器的高4位 [7 : 4], 所以STM32提供了最大16级的中断优先等级

STM32 的中断优先级可以分为抢占优先级和子优先级

抢占优先级: 抢占优先级高的中断可以打断正在执行但抢占优先级低的中断

子优先级: 当同时发生具有相同抢占优先级的两个中断时, 子优先级数值小的优先执行

共有5种优先级分组分配方式

NVIC_PriorityGroup_00bit 用于抢占优先级, 4bit 用于子优先级

NVIC_PriorityGroup_11bit 用于抢占优先级 3bit 用于子优先级

NVIC_PriorityGroup_22bit 用于抢占优先级 2bit 用于子优先级

NVIC_PriorityGroup_33bit 用于抢占优先级 1bit 用于子优先级

NVIC_PriorityGroup_44bit 用于抢占优先级 0bit 用于子优先级

相关说明可查看官网: <https://www.freertos.org/RTOS-Cortex-M3-M4.html>

特点

1、低于configMAX_SYSCALL_INTERRUPT_PRIORITY优先级的中断里才允许调用FreeRTOS 的API函数

2、建议将所有优先级位指定为抢占优先级位, 方便FreeRTOS管理

3、中断优先级数值越小越优先, 任务优先级数值越大越优先

3, 中断相关寄存器 (熟悉)

三个系统中断优先级配置寄存器, 分别为分别为 SHPR1、 SHPR2、 SHPR3

通过SHPR3将PendSV和SysTick的中断优先级设置为最低优先级, 保证系统任务切换不会阻塞系统其他中断的响应

三个中断屏蔽寄存器, 分别为 PRIMASK、 FAULTMASK 和BASEPRI

FreeRTOS所使用的中断管理就是利用的BASEPRI这个寄存器

BASEPRI: 屏蔽优先级低于某一个阈值的中断

比如: BASEPRI设置为0x50, 代表中断优先级在5~15内的均被屏蔽, 0~4的中断优先级正常执行

学习资料参考《Cortex M3权威指南(中文)》手册

4, FreeRTOS中断管理实验 (掌握)

实验目的: 学会使用FreeRTOS的中断管理

本实验会使用两个定时器, 一个优先级为4, 一个优先级为6, 注意: 系统所管理的优先级范围: 5~15, 现象: 两个定时器每1s, 打印一段字符串, 当关中断时, 停止打印, 开中断时持续打印。

2、实验设计: 将设计2个任务: start_task、task1

start_task用来创建task1任务

task1中断测试任务, 任务中将调用关中断和开中断函数来体现对中断的管理作用!