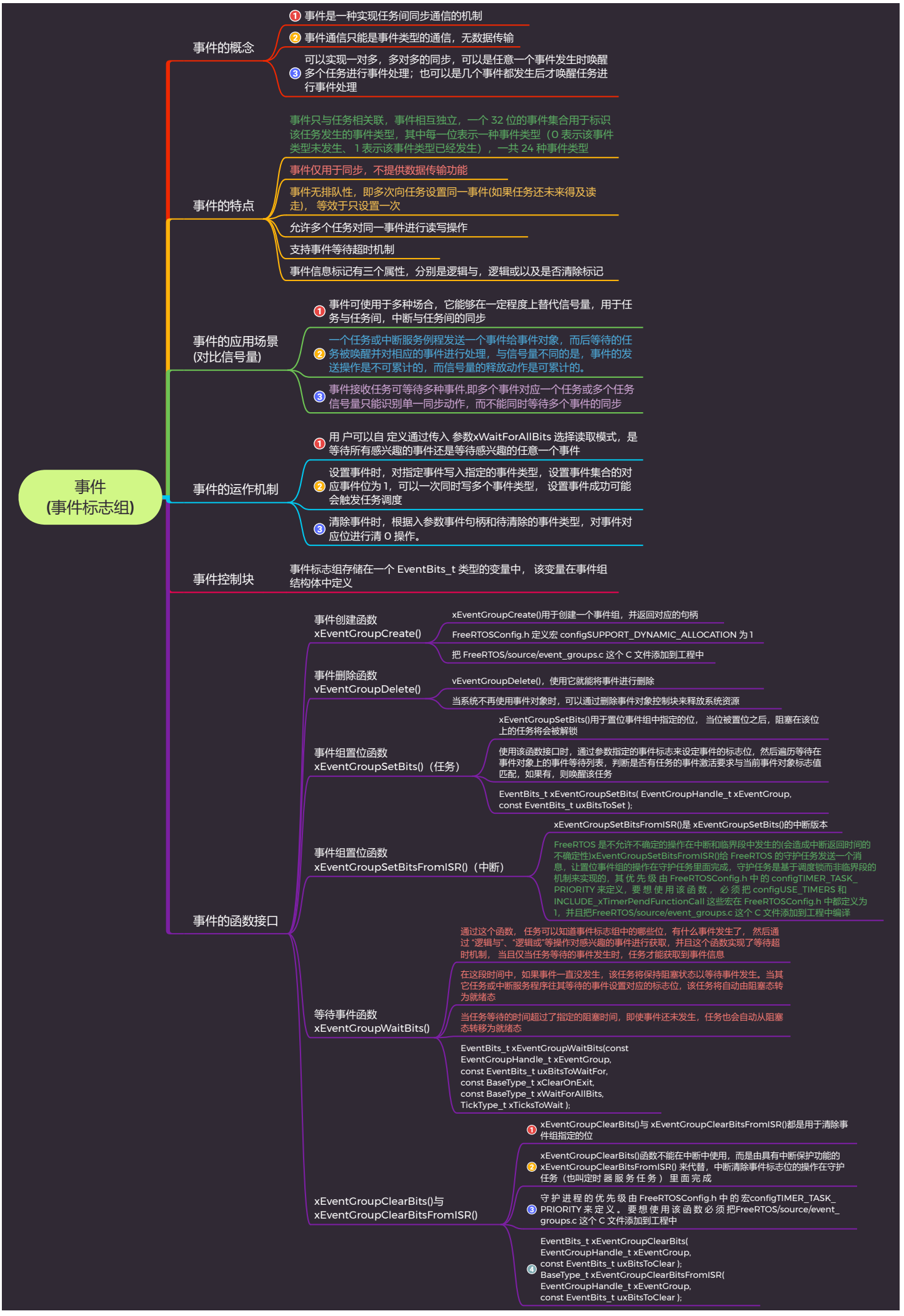
事件
(事件标志组)
事件的概念
① 事件是一种实现任务间同步通信的机制
② 事件通信只能是事件类型的通信，无数据传输
③ 可以实现一对多，多对多的同步，可以是任意一个事件发生时唤醒多个任务进行事件处理；也可以是几个事件都发生后才唤醒任务进行事件处理
事件的特点
事件只与任务相关联，事件相互独立，一个 32 位的事件集合用于标识该任务发生的事件类型，其中每一位表示一种事件类型（0 表示该事件类型未发生、 1 表示该事件类型已经发生），一共 24 种事件类型
事件仅用于同步，不提供数据传输功能
事件无排队性，即多次向任务设置同一事件(如果任务还未来得及读走)， 等效于只设置一次
允许多个任务对同一事件进行读写操作
支持事件等待超时机制
事件信息标记有三个属性，分别是逻辑与，逻辑或以及是否清除标记
事件的应用场景
(对比信号量)
① 事件可使用于多种场合，它能够在一定程度上替代信号量，用于任务与任务间，中断与任务间的同步
② 一个任务或中断服务例程发送一个事件给事件对象，而后等待的任务被唤醒并对相应的事件进行处理，与信号量不同的是，事件的发送操作是不可累计的，而信号量的释放动作是可累计的。
③ 事件接收任务可等待多种事件,即多个事件对应一个任务或多个任务信号量只能识别单一同步动作，而不能同时等待多个事件的同步
事件的运作机制
① 用 户可以自 定义通过传入 参数xWaitForAllBits 选择读取模式，是等待所有感兴趣的事件还是等待感兴趣的任意一个事件
② 设置事件时，对指定事件写入指定的事件类型，设置事件集合的对应事件位为 1，可以一次同时写多个事件类型， 设置事件成功可能会触发任务调度
③ 清除事件时，根据入参数事件句柄和待清除的事件类型，对事件对应位进行清 0 操作。
事件控制块
事件标志组存储在一个 EventBits_t 类型的变量中， 该变量在事件组结构体中定义
事件的函数接口
事件创建函数
xEventGroupCreate()
xEventGroupCreate()用于创建一个事件组，并返回对应的句柄
FreeRTOSConfig.h 定义宏 configSUPPORT_DYNAMIC_ALLOCATION 为 1
把 FreeRTOS/source/event_groups.c 这个 C 文件添加到工程中
事件删除函数
vEventGroupDelete()
vEventGroupDelete()，使用它就能将事件进行删除
当系统不再使用事件对象时，可以通过删除事件对象控制块来释放系统资源
事件组置位函数
xEventGroupSetBits()（任务）
xEventGroupSetBits()用于置位事件组中指定的位， 当位被置位之后，阻塞在该位上的任务将会被解锁
使用该函数接口时，通过参数指定的事件标志来设定事件的标志位，然后遍历等待在事件对象上的事件等待列表，判断是否有任务的事件激活要求与当前事件对象标志值匹配，如果有，则唤醒该任务
EventBits_t xEventGroupSetBits(EventGroupHandle_t xEventGroup,
const EventBits_t uxBitsToSet);
事件组置位函数
xEventGroupSetBitsFromISR()（中断）
xEventGroupSetBitsFromISR()是 xEventGroupSetBits()的中断版本
FreeRTOS 是不允许不确定的操作在中断和临界段中发生的(会造成中断返回时间的不确定性)xEventGroupSetBitsFromISR()给 FreeRTOS 的守护任务发送一个消息，让置位事件组的操作在守护任务里面完成，守护任务是基于调度锁而非临界段的机制来实现的，其 优 先 级 由 FreeRTOSConfig.h 中 的 configTIMER_TASK_PRIORITY 来定义，要 想 使 用 该 函 数， 必 须 把 configUSE_TIMERS 和 INCLUDE_xTimerPendFunctionCall 这些宏在 FreeRTOSConfig.h 中都定义为 1，并且把FreeRTOS/source/event_groups.c 这个 C 文件添加到工程中编译
等待事件函数
xEventGroupWaitBits()
通过这个函数，任务可以知道事件标志组中的哪些位，有什么事件发生了， 然后通过 “逻辑与”、“逻辑或”等操作对感兴趣的事件进行获取，并且这个函数实现了等待超时机制， 当且仅当任务等待的事件发生时，任务才能获取到事件信息
在这段时间中，如果事件一直没发生，该任务将保持阻塞状态以等待事件发生。当其它任务或中断服务程序往其等待的事件设置对应的标志位，该任务将自动由阻塞态转为就绪态
当任务等待的时间超过了指定的阻塞时间，即使事件还未发生，任务也会自动从阻塞态转移为就绪态
EventBits_t xEventGroupWaitBits(const
EventGroupHandle_t xEventGroup,
const EventBits_t uxBitsToWaitFor,
const BaseType_t xClearOnExit,
const BaseType_t xWaitForAllBits,
TickType_t xTicksToWait);
xEventGroupClearBits()与
xEventGroupClearBitsFromISR()
① xEventGroupClearBits()与 xEventGroupClearBitsFromISR()都是用于清除事件组指定的位
② xEventGroupClearBits()函数不能在中断中使用，而是由具有中断保护功能的 xEventGroupClearBitsFromISR() 来代替，中断清除事件标志位的操作在守护任务（也叫定时 器 服 务 任 务 ） 里 面 完 成
③ 守 护 进 程 的 优 先 级 由 FreeRTOSConfig.h 中 的 宏configTIMER_TASK_PRIORITY 来 定 义 。 要 想 使 用 该 函 数 必 须 把FreeRTOS/source/event_groups.c 这个 C 文件添加到工程中
④ EventBits_t xEventGroupClearBits(
EventGroupHandle_t xEventGroup,
const EventBits_t uxBitsToClear);
BaseType_t xEventGroupClearBitsFromISR(
EventGroupHandle_t xEventGroup,
const EventBits_t uxBitsToClear);