1. 中断请求和事件请求有何区别

中断请求是外部设备要求处理器服务的信号，事件请求是外部设备通知处理器发生了什么的信号。

中断请求可以随时发生，事件请求只能在处理器允许的时候发生。中断请求必须被处理器响应，事件请求可以被处理器忽略或者延迟处理。

中断请求会打乱处理器的任务，事件请求不会改变处理器的任务。

1. 解释什么是NVIC和EXTI

NVIC是中断向量控制器（Nested Vectored Interrupt Controller）的缩写，是一种硬件组件，用于处理中断请求。NVIC可以管理和处理多个中断，每个中断具有自己的优先级和中断处理程序。

EXTI是外部中断控制器（External Interrupt Controller）的缩写，也是一种硬件组件，用于处理外部中断请求。EXTI可以监测并处理这些外部中断请求，并触发相应的中断处理程序。

1. 外中断请求线和中断请求通道的区别及映射关系（见PPT32页图）

NVIC是中断向量控制器（Nested Vectored Interrupt Controller）的缩写，是一种硬件组件，用于处理中断请求。NVIC可以管理和处理多个中断，每个中断具有自己的优先级和中断处理程序。

EXTI是外部中断控制器（External Interrupt Controller）的缩写，也是一种硬件组件，用于处理外部中断请求。EXTI可以监测并处理这些外部中断请求，并触发相应的中断处理程序。

1. 中断嵌套的规则是怎样的

中断嵌套是指在一个中断服务程序（ISR）执行期间，另一个中断被触发并请求执行。中断嵌套通常是在多任务嵌入式系统中使用，以处理多个同时发生的事件或任务。

中断优先级：通常，中断控制器会支持中断优先级。中断被分为不同的优先级，较高优先级的中断可以中断正在执行的较低优先级中断。这允许开发者确保关键的中断能够及时处理。

中断嵌套的规则和行为是高度依赖硬件和操作系统的。在设计嵌套中断处理时，需要详细了解所使用的硬件架构和中断控制器的规范，以确保正确的中断优先级和嵌套行为。

1. 中断服务子程(ISR)最后一条语句为什么要清中断标志

中断服务子程的最后一条语句清除中断标志是为了确保中断的正确处理。在中断发生时，CPU会将中断标志置位，以禁用其他中断，同时保证当前中断能够得到及时响应。但是，在中断服务子程执行完毕后，为了回到正常的程序执行流程，需要重新允许其他中断。因此，清除中断标志可以重置CPU的中断状态，使得其他中断能够继续被处理。

1. 描述一下通用定时器的时基单元构成及各部分功能

通用定时器的时基单元由以下几个部分组成：

- 时钟源选择：可以选择内部时钟源或外部时钟源作为定时器的时钟输入。

- 预分频器：用于将时钟源的频率进行预分频，以得到更低的计数频率。

- 自动重载寄存器（ARR）：用于设置定时器的计数上限，当定时器计数达到该值时，定时器将重新开始计数。

- 计数器：用于记录定时器的计数值，计数器的值从0开始，每次计数增加1。

- 捕获/比较寄存器（CCR）：用于设置定时器的比较值，可以用于产生输出或触发中断。

- 中断控制器：用于控制定时器中断的使能和响应。

每个部分的功能如下：

- 时钟源选择：选择定时器的时钟输入源。

- 预分频器：将时钟源的频率进行预分频，得到更低的计数频率。

- ARR：设置定时器的计数上限，决定定时器的计数周期。

- 计数器：记录定时器的计数值。

- CCR：设置定时器的比较值，可用于产生输出或触发中断。

- 中断控制器：控制定时器中断的使能和响应。

7. 若APB1总线频率20MHz，TIM3的预分频为4000，计数上限为5000，列式计算定时长 - 计算定时器的频率: 20MHz / 4000 = 5kHz

- 计算每个计数周期的时间: 1 / 5kHz = 0.2ms

- 计算定时长: 5000 \* 0.2ms = 1000ms = 1s

8. 定时器中断编程分哪三步走

定时器中断编程的主要步骤如下：

1. 初始化定时器：配置定时器的时钟源、预分频和计数上限等参数。

2. 配置中断：使能定时器的中断，并设置中断触发条件（比较值或计数溢出等）。

3. 编写中断处理函数：当定时器中断触发时，执行相应的中断处理代码。在中断处理函数中可以执行需要定时执行的任务，例如更新显示、处理数据等。

这些步骤可以根据具体的编程平台和语言进行实现。