

STM32(Cortex-M3)中的优先级概念

STM32(Cortex-M3)中有两个优先级的概念——抢占式优先级和响应优先级，有人把响应优先级称作'亚优先级'或'副优先级'，每个中断源都需要被指定这两种优先级。

具有高抢占式优先级的中断可以在具有低抢占式优先级的中断处理过程中被响应，即中断嵌套，或者说高抢占式优先级的中断可以嵌套低抢占式优先级的中断。

当两个中断源的抢占式优先级相同时，这两个中断将没有嵌套关系，当一个中断到来后，如果正在处理另一个中断，这个后到来的中断就要等到前一个中断处理完之后才能被处理。如果这两个中断同时到达，则中断控制器根据他们的响应优先级高低来决定先处理哪一个；如果他们的抢占式优先级和响应优先级都相等，则根据他们在中断表中的排位顺序决定先处理哪一个。

既然每个中断源都需要被指定这两种优先级，就需要有相应的寄存器位记录每个中断的优先级；在 Cortex-M3 中定义了 8 个比特位用于设置中断源的优先级，这 8 个比特位可以有 8 种分配方式，如下：

所有 8 位用于指定响应优先级

最高 1 位用于指定抢占式优先级，最低 7 位用于指定响应优先级

最高 2 位用于指定抢占式优先级，最低 6 位用于指定响应优先级

最高 3 位用于指定抢占式优先级，最低 5 位用于指定响应优先级

最高 4 位用于指定抢占式优先级，最低 4 位用于指定响应优先级

最高 5 位用于指定抢占式优先级，最低 3 位用于指定响应优先级

最高 6 位用于指定抢占式优先级，最低 2 位用于指定响应优先级

最高 7 位用于指定抢占式优先级，最低 1 位用于指定响应优先级

这就是优先级分组的概念。

Cortex-M3 允许具有较少中断源时使用较少的寄存器位指定中断源的优先级，因此 STM32 把指定中断优先级的寄存器位减少到 4 位，这 4 个寄存器位的分组方式如下：

第 0 组：所有 4 位用于指定响应优先级

第 1 组：最高 1 位用于指定抢占式优先级，最低 3 位用于指定响应优先级

第 2 组：最高 2 位用于指定抢占式优先级，最低 2 位用于指定响应优先级

第 3 组：最高 3 位用于指定抢占式优先级，最低 1 位用于指定响应优先级

第 4 组：所有 4 位用于指定抢占式优先级

可以通过调用 STM32 的固件库中的函数 `NVIC_PriorityGroupConfig()` 选择使用哪种优先级分组方式，这个函数的参数有下列 5 种：

`NVIC_PriorityGroup_0` => 选择第 0 组

`NVIC_PriorityGroup_1` => 选择第 1 组

`NVIC_PriorityGroup_2` => 选择第 2 组

`NVIC_PriorityGroup_3` => 选择第 3 组

`NVIC_PriorityGroup_4` => 选择第 4 组

接下来就是指定中断源的优先级，下面以一个简单的例子说明如何指定中断源的抢占式优先级和响应优先级：

// 选择使用优先级分组第 1 组

```
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
```

// 使能 EXTI0 中断

```
NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQn; // 这里原来  
为 EXTI0_5_IRQChannel,但要改为 EXTI0_5_IRQn
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1; // 指 定  
抢占式优先级别 1
```

```
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0; // 指定响应优  
先级别 0
```

```
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
```

```
NVIC_Init(&NVIC_InitStructure);
```

// 使能 EXTI9_5 中断

```
NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQn; // 这里原来  
为 EXTI9_5_IRQChannel,但要改为 EXTI9_5_IRQn
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; // 指 定  
抢占式优先级别 0
```

```
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1; // 指定响应优先级别 1
```

```
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
```

```
NVIC_Init(&NVIC_InitStructure);
```

要注意的几点是：

- 1) 如果指定的抢占式优先级别或响应优先级别超出了选定的优先级分组所限定的范围，将可能得到意想不到的结果；
- 2) 抢占式优先级别相同的中断源之间没有嵌套关系；
- 3) 如果某个中断源被指定为某个抢占式优先级别，又没有其它中断源处于同一个抢占式优先级别，则可以为这个中断源指定任意有效的响应优先级别