

```
//=====
//      STM32 中断与嵌套NVIC 快速入门。
//      netjob 2008-8-1
//=====
```

我也是靠看这本书才弄懂的：

Cortex-M3 权威指南  
Joseph Yiu 著  
宋岩 译

其实很简单。

//CM3 有 最多240个中断（通常外部中断写作IRQs），就是 软件上说的 IRQ  
CHANAELx(中断通道号x)  
每个中断有自己的可编程的中断优先级【有唯一对应的 中断优先级寄存器】。

由于CM3支持 硬件中断嵌套，所以可以有 256 级的可编程优先级  
和 256级中断嵌套【书上称：抢占（preempt） 优先级】  
所以大家可以设：

IRQ CHANAEL 0	通道 = 2	中断优先级	WWDG 窗口定时器中断
IRQ CHANAEL 1	通道 = 0	中断优先级	PVD 联到EXTI的电源电压检测 (PVD) 中断
IRQ CHANAEL 3	通道 = 255	中断优先级	RTC 实时时钟 (RTC) 全局中断
IRQ CHANAEL 6	通道 = 10	中断优先级	EXTIO EXTI线0中断
.....			
IRQ CHANAEL 239	通道 = (0<x<255)	中断优先级	..

这个实在是太恐怖了！是的，其实CM3 并没有这样做。  
实在的芯片例如STM32等就只有设计来可用才64级可编程优先级和8级中断嵌套。

对 64级中断就是说：（INT0 到 INT63）这个大家比较好理解，其它的64...239就不用了。  
IRQ CHANAEL 0  
。。。  
IRQ CHANAEL 63

而8级中断嵌套这又是何解呢？

是这样的，上面说 一个【中断】对应 一个【中断优先级寄存器】，而这个寄存器是 8 位的。  
当然就是 256级了。而现在就用了 它其中的 BIT7, IT6, BIT5 三位来表示，而且是MSB对齐的。

用了3 个位来表达优先级 (MSB 对齐的我们能够使用的8  
个优先级为：0x00（最高），0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0 以及0xE0。）  
这样我们在【中断优先级寄存器】就不能按理论的填 0到255之间的数了，  
而只能填0x00（最高），0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0 以及0xE0。）

所以大家可以设：

IRQ CHANAEL 0	通道 = 0x20	中断优先级	WWDG 窗口定时器中断
IRQ CHANAEL 1	通道 = 0x40	中断优先级	PVD 联到EXTI的电源电压检测 (PVD) 中断
IRQ CHANAEL 3	通道 = 0x20	中断优先级	RTC 实时时钟 (RTC) 全局中断
IRQ CHANAEL 6	通道 = 0xA0	中断优先级	EXTIO EXTI线0中断
.....			
IRQ CHANAEL 63	通道 =	【0x00（最高），0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0 以及0xE0。）】	

大家注意到了，上面通道0和通道3 的优先级都是0x20，这怎么办？

// 如果优先级完全相同的多个异常同时悬起，则先响应异常编号最小的那一个。如IRQ  
#0会比IRQ #3 先得到响应

而且文中还讲了 **【优先级分组】**，这又是怎么回事？

其实我回头看来，这个 **【优先级分组】** 和 **【抢占优先级】** **【亚优先级】** 都毫无意义的。  
如果当时用 256级即是  
把 **【中断优先级寄存器】** 的8位都全用上，就没这个必要了。还什么优先级分组呢！  
就是因为厂家现在 **【偷工减料】**，才搞出这个明堂来的。

是这样的，在 应用程序中断及复位控制寄存器(AIRCR) 中的 10：8  
位 **【3位】** 是表示 **【优先级分组】**  
它作用主要是  
用于对 **【中断优先级寄存器】** 『我们现在中用了BIT7,BIT6,BIT5三位』的功能的说明。  
有一个表，在《Cortex-M3 权威指南》的110页， 例如我们把AIRCR的10：8 位设为 **【5】**，  
查表可得 **【抢占优先级】** = **【7：6】**， **【亚优先级】** = **【5：0】**，

对于 **【中断优先级寄存器】** 只用了BIT7,6,5, 因此我们可以看作是 **【7：6】**， **【5】**。那4—0  
可以不管。

现在我们的 IRQ0=0X20, IRQ3=0X20, 也就是 **【0 0 1 0 】** 『  
bit7=0,bit6=0,bit5=1,bit4=0』  
因为大家（IRQ0/IRQ3）的 **【抢占优先级】** = **【7：6】** 都是0，  
说明它们的中断相应级别是一样的。  
再继续判断它们哪个更优先的责任就要看 **【5】**，结果连 **【5】** 都是一样的！  
那就按默认：

// 如果优先级完全相同的多个异常同时悬起，则先响应异常编号最小的那一个。如IRQ  
#0会比IRQ #3 先得到响应

由于CM3没有 进中断 **【关全局中断相应】** 这事，只要是中断通道打开了，就会存在 通道间的  
嵌套，即是会发生  
**【抢占】** 的情况了。

上面就简短的说明，如果要详细理解，可以看《Cortex-M3 权威指南》。  
有任何理解不当，请各位多多指教！