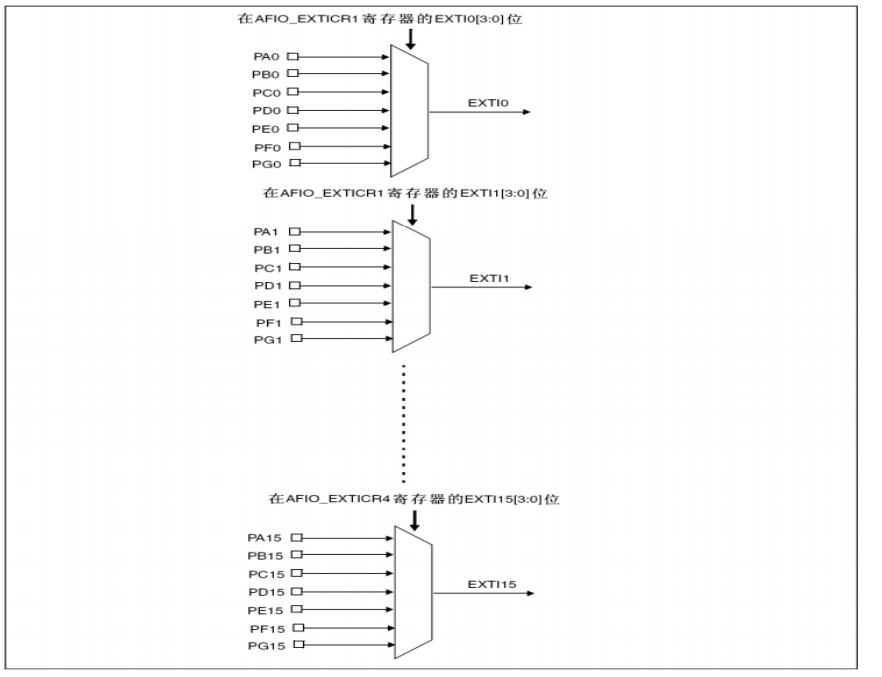
**STM32共提供了16根中断线，但外部引脚数量远远大于中断线的数量，ST公司的设计如下图：**



**可以看到PA0，PB0，……PG0都对应第一根中断线，由图所示，所有引脚都有自己对应的中断线。但是每个中断线只能选择他的候选项中的一条，这样不固定引脚，大大提升了使用的自由。**

**GPIO\_EXTILineConfig(GPIO\_PortSourceGPIOA,GPIO\_PinSource3); //将中断3和GPIOA连接。**

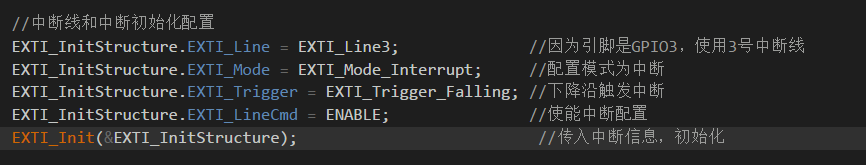
**GPIO\_EXTILineConfig函数负责将你选用的引脚和他的中断线绑定起来。上例即使用3号中断，GPIOA，即PA3。**

**在使用外部中断之前，记得开启引脚复用时钟：**

**RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_AFIO,ENABLE); //使能管脚复用时钟**

**同样要初始化中断使用的GPIO，初始化为上拉输入状态。**

**中断的配置和GPIO的配置一样，需要设定自己的配置信息结构体。（该图引脚号与原理图不同，请对照原理图使用）**



**从定义可以看出，有 4 个参数需要设置 。 第一个参数是中断线的标号，范围为EXTI\_Line0……EXTI\_Line15。第二个参数是中断模式， 可选值为中断 EXTI\_Mode\_Interrupt 和事件 EXTI\_Mode\_Event。第三个参数是触发方式，可以是下降沿触发 EXTI\_Trigger\_Falling，上升沿触发 EXTI\_Trigger\_Rising，或者任意电平（上升沿和下降沿） 触发EXTI\_Trigger\_Rising\_Falling，相信学过 51 的对这个不难理解。 最后一个参数就是使能中断线了。**

**既然涉及到中断，就必须要考虑中断的优先级问题。**

**中断优先级的分配表如下**

**在例程中我们采用2组分配方式，即2：2。**

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);

**二进制共四位，对半分配则抢占优先级的可选范围为0到3，相应优先级为0到1）如果两个中断的抢占优先级和响应优先级都一样，则哪个中断先发生执行哪个。**

**2）高抢占优先级可以打断正在执行的低抢占优先级的中断，抢占优先级相同，高响应优先级的中断无法打断低响应优先级的中断。接下来我们配置中断的优先级。**

//中断优先级配置

    NVIC\_InitStructure.NVIC\_IRQChannel=EXTI3\_IRQn; //使能按键外部中断通道

    NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority=2;//抢占优先级为2

    NVIC\_InitStructure.NVIC\_IRQChannelSubPriority=2; //子优先级为2

    NVIC\_InitStructure.NVIC\_IRQChannelCmd=ENABLE; //使能配置

    NVIC\_Init(&NVIC\_InitStructure);  //传入配置信息，初始化

**这里的外部中断通道其实就是使能该中断，中断号都在stm32f10x.h中定义。外部中断号如下：**

**EXTI0\_IRQn**

**EXTI1\_IRQn**

**EXTI2\_IRQn**

**EXTI3\_IRQn**

**EXTI4\_IRQn**

**EXTI9\_5\_IRQn**

**EXTI15\_10\_IRQn**

**可以看到5号到9号中断共用一个，10号到15号共用一个。因为中断号和中断函数是一一对应的。中断函数名可以在启动文件\*.s的汇编文件中找到。**

**中断函数即在中断发生后执行的函数。因为15到10号中断函数比较特殊，拿此举例：**

**void EXTI15\_10\_IRQHandler()**

**{**

**if(EXTI\_GetITStatus(EXTI\_Line12) != RESET)**

**{**

**//添加你自己的代码**

**EXTI\_ClearITPendingBit(EXTI\_Line12);**

**}**

**}**

**该句判断12号中断是否发生**

**if(EXTI\_GetITStatus(EXTI\_Line12) != RESET);**

**当你在10号到15号设定了多个中断时，你就需要好几个判断语句来判定到底是哪个中断发生了。当执行完自己的代码以后，通过清除中断占用标志位来恢复中断的使能EXTI\_ClearITPendingBit(EXTI\_Line12);该句清除12号中断的中断标志位。**