中断：串口接收中断实验

1. **Stm32中断机制简述**

STM32系列总共有84个中断，因此STM32中有一个中断的管理机制来管理这么多中断，使能中断并且配置相关中断的优先级，这就是NVIC的工作。

NVIC（Nested Vectored Interrupt Controller）内嵌向量中断控制器。对于每个中断需要设置抢占优先级（Preemption Priority）和响应优先级（Sub Priority），多个中断会先比较抢占优先级，抢占优先级相同则比较响应优先级。这里需要注意的是，高抢占优先级能够打断低抢占优先级的中断，但是相同的抢占优先级的情况下，高响应优先级不能够打断低响应优先级。

NVIC利用四个位来保存抢占优先级和响应优先级。我们可以自由设置用哪几个位来保存抢占优先级，哪几位用来保存响应优先级。如表1.所示。

表1. 中断分组表

|  |  |  |  |
| --- | --- | --- | --- |
| 中断优先级分组 | 分配结果 | 抢占优先级取值 | 响应优先级取值 |
| 0 | 0位抢占优先级，4位响应优先级 | 0~0 | 0~15 |
| 1 | 1位抢占优先级，3位响应优先级 | 0~1 | 0~7 |
| 2 | 2位抢占优先级，2位响应优先级 | 0~3 | 0~3 |
| 3 | 3位抢占优先级，1位响应优先级 | 0~7 | 0~1 |
| 4 | 4位抢占优先级，0位响应优先级 | 0~15 | 0~0 |

需要注意的是，无论是抢占优先级还是响应优先级，值越小则表明优先级越高。

1. **实验说明**

由上位机的串口调试助手发送字符到飞控板，应用串口中断的方式，将接收到串口数据再发回到上位机去。硬件连接方式与实验7相同。

1. **工程配置**

依照图1. 所示，配置完成stm32F427芯片的串口3中断配置。

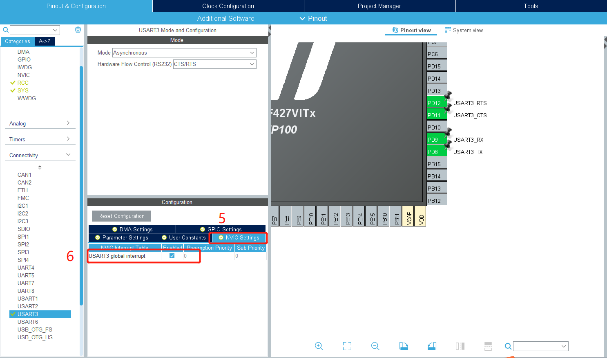
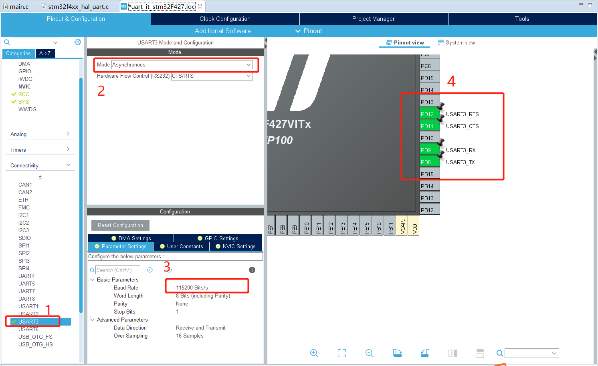


图1. 串口3中断配置

1）~4）步骤在之前的实验中已经讲解过。

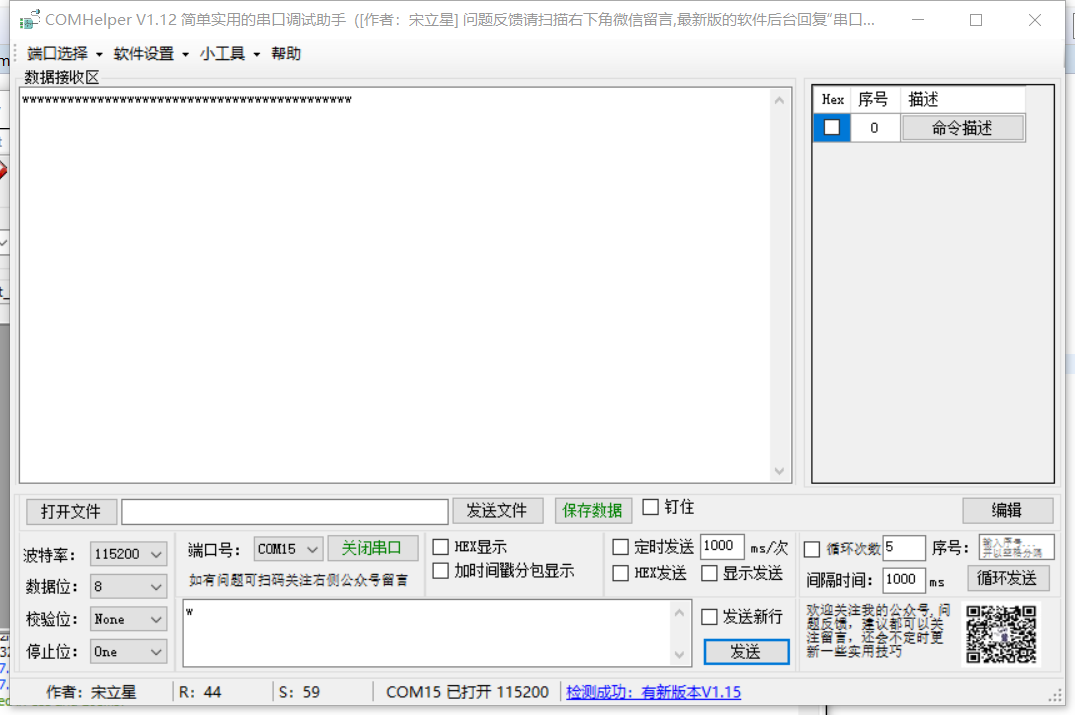
5）~6）NVIC：使能NVIC，抢占优先级为0，响应优先级为0（只有一个中断，只能这样配置）

在主函数中，按如下方式加入代码：

|  |
| --- |
| 1. 声明全局变量   //设置全局变量，在主函数之前  uint8\_t uart1\_rxbuf[10];   1. 在主函数之后，添加中断处理函数   /\*  \* 这个函数是一个weak函数，HAL库接收到数据的时候会调用它，你可以重写它，HAL库里  \* 原本的它什么也不做  \*/  **void** **HAL\_UART\_RxCpltCallback**(UART\_HandleTypeDef \*huart)  {  HAL\_UART\_Transmit(&huart3, (uint8\_t \*)uart1\_rxbuf, 1, 100); // 把收到的字节原样发送出去  HAL\_UART\_Receive\_IT(&huart3, uart1\_rxbuf, 1); // 重新注册一次，要不然下次收不到了  } |

1. **实验效果**

通过串口助手向下位机发送‘w’字符，串口助手接收到了‘w’。如下图所示：



1. **延伸**

修改代码，将串口中断与LED实验结合到一起，发送字符串“ON”，则LED灯亮；发送“OFF”，则LED灯灭的功能。