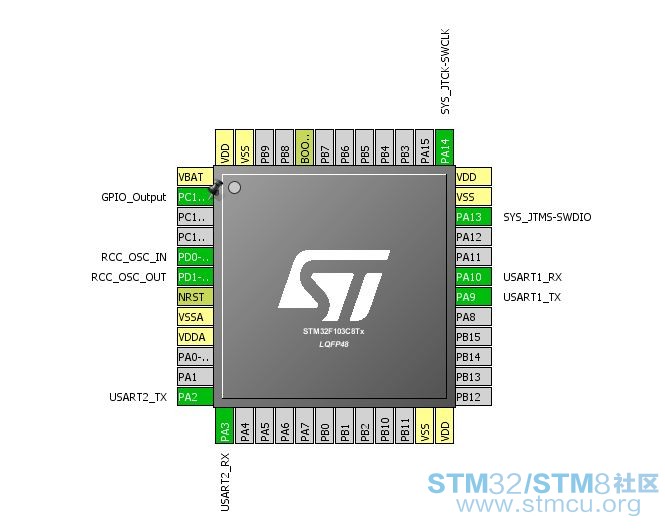
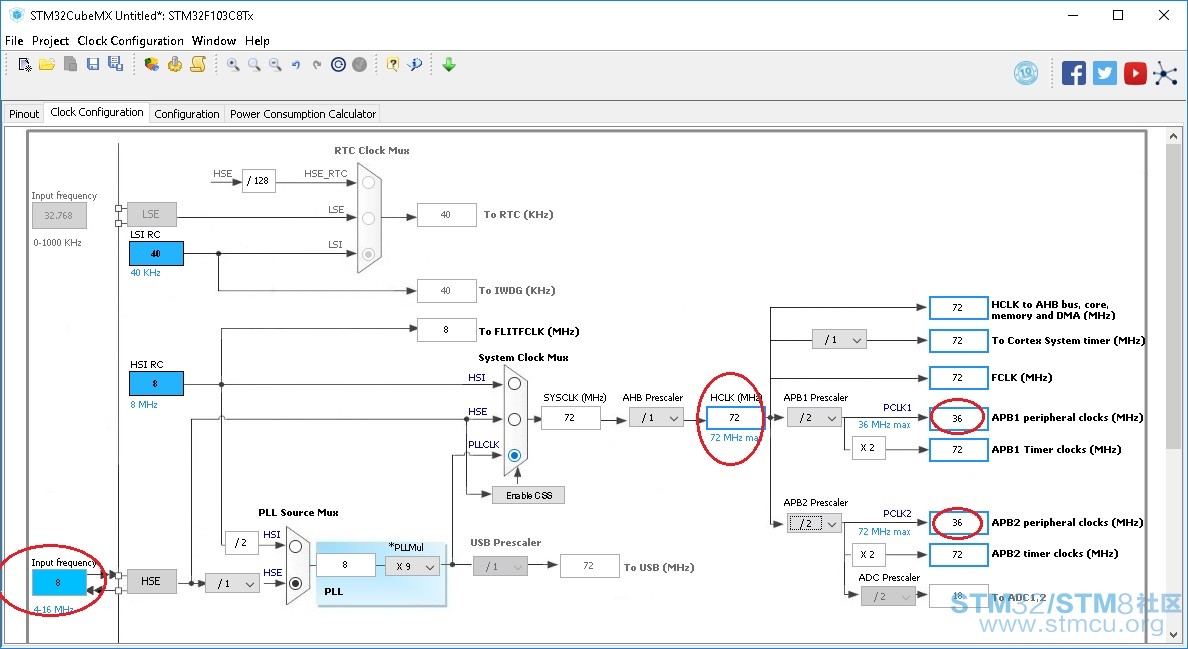
STM32基于STM32CubeMX的高速串口收发程序（中断模式）

看到大家经常写一些关于串口的问题，我也尝试开一个使用串口的工程。  
简单的要求是，USART1和USART2之间互相转发。  
发现使用CubeMX平台，建立测试工程没有难度，情况如下：  
  
1. 配置管脚，使用BluePill最小板，PC13接有LED：  
MCU是STM32F103C8T6，这个不重要. 不要忘记配置SWD调试接口。  


**0.JPG** (47.23 KB, 下载次数: 0)

[下载附件](http://www.stmcu.org/module/forum/forum.php?mod=attachment&aid=NDE2MTIyfDA4MjIzOTg5fDE1MzIwNzIyODB8MHw2MTY2MTM%3D&nothumb=yes)

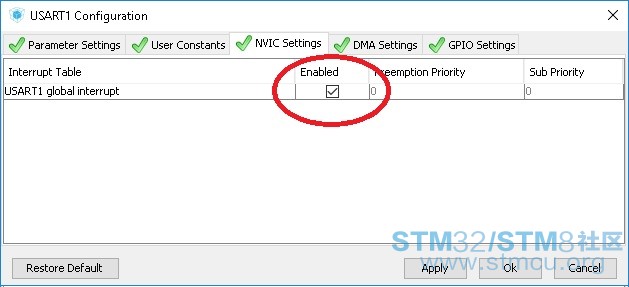
6 天前 上传

2. 配置时钟，重点已经标出：  


**1.jpg** (132.99 KB, 下载次数: 0)

[下载附件](http://www.stmcu.org/module/forum/forum.php?mod=attachment&aid=NDE2MTIzfDk0YzQyOWZifDE1MzIwNzIyODB8MHw2MTY2MTM%3D&nothumb=yes)

6 天前 上传

3. 配置中断选项，由于不使用DMA，可以不理DMA的选项：  
（USART2也要同样配置）  


**2.jpg** (36.73 KB, 下载次数: 0)

[下载附件](http://www.stmcu.org/module/forum/forum.php?mod=attachment&aid=NDE2MTI0fGZhNjVkOTFkfDE1MzIwNzIyODB8MHw2MTY2MTM%3D&nothumb=yes)

6 天前 上传

（然后，生成工程代码）  
  
4. 添加代码  
4.1 在main的初始化部分，添加接收中断使能：

1. \_\_HAL\_UART\_ENABLE\_IT(&huart1, UART\_IT\_RXNE); //enable Rx INT
2. \_\_HAL\_UART\_ENABLE\_IT(&huart2, UART\_IT\_RXNE); //enable Rx INT

4.2 打开stm32f1xx\_it.c，在前面添加接收缓存结构体：

1. /\* USER CODE BEGIN 0 \*/
2. #define UART\_BUFFER\_SIZE    64 //here must be 2^n
3. struct bufer\_st {
4. unsigned int in;                // Next In Index
5. unsigned int out;               // Next Out Index
6. char buffer [UART\_BUFFER\_SIZE]; // Buffer
7. };
8. static struct bufer\_st uart1buffer = { 0, 0, };
9. static struct bufer\_st uart2buffer = { 0, 0, };
10. /\* USER CODE END 0 \*/

4.3 在stm32f1xx\_it.c的底下，修改USART1和USART2的中断服务程序如下：

1. /\*\*
2. \* @brief This function handles USART1 global interrupt.
3. \*/
4. void USART1\_IRQHandler(void)
5. {
6. /\* USER CODE BEGIN USART1\_IRQn 0 \*/
7. struct bufer\_st \*p;
8. /\* USER CODE END USART1\_IRQn 0 \*/
9. HAL\_UART\_IRQHandler(&huart1);
10. /\* USER CODE BEGIN USART1\_IRQn 1 \*/
11. p = &uart1buffer; //use UART1 buffer
13. //raed UART1
14. if (USART1->SR & UART\_IT\_RXNE) // read interrupt
15. {
16. USART1->SR &= ~UART\_IT\_RXNE; // clear interrupt
17. if (((p->in - p->out) & ~(UART\_BUFFER\_SIZE-1)) == 0) {
18. p->buffer[p->in & (UART\_BUFFER\_SIZE-1)] = (USART1->DR & 0xFF); //read DATA
19. p->in++;
20. }
21. }
22. //send to UART2
23. if (p->in != p->out)
24. {
25. USART2->DR = (p->buffer[p->out & (UART\_BUFFER\_SIZE-1)] & 0xFF); //send out
26. p->out++;
27. }
28. /\* USER CODE END USART1\_IRQn 1 \*/
29. }
30. /\*\*
31. \* @brief This function handles USART2 global interrupt.
32. \*/
33. void USART2\_IRQHandler(void)
34. {
35. /\* USER CODE BEGIN USART2\_IRQn 0 \*/
36. struct bufer\_st \*p;
37. /\* USER CODE END USART2\_IRQn 0 \*/
38. HAL\_UART\_IRQHandler(&huart2);
39. /\* USER CODE BEGIN USART2\_IRQn 1 \*/
40. p = &uart2buffer; //use UART2 buffer
42. //raed UART2
43. if (USART2->SR & UART\_IT\_RXNE) // read interrupt
44. {
45. USART2->SR &= ~UART\_IT\_RXNE; // clear interrupt
46. if (((p->in - p->out) & ~(UART\_BUFFER\_SIZE-1)) == 0)
47. {
48. p->buffer[p->in & (UART\_BUFFER\_SIZE-1)] = (USART2->DR & 0xFF); //read DATA
49. p->in++;
50. }
51. }
52. //send to UART1
53. if (p->in != p->out)
54. {
55. USART1->DR = (p->buffer[p->out & (UART\_BUFFER\_SIZE-1)] & 0xFF); //send out
56. p->out++;
57. }
58. /\* USER CODE END USART2\_IRQn 1 \*/
59. }

（看得出，中断里面是收到一个字符发送一个字符。 发送是直接发送，不处理发送中断。）

大功告成了！！ 编译后烧录。

测试条件，使用两个UART转USB板子（FTDI）和sscom5上位机程序（开两个），发送区都摆放700多个字符，以10ms的间隔连续发送。

测试结果，速度115200bps和1Mbps，双向同时收发100万字符无差错。 2Mbps，单方向100万字符无差错。

（高速测试时，需要修改main.c里面的串口速度BaudRate配置。 我没有测试“自动波特率”的模式是否正常。）

完整代码请见附件。

原文作者是STM32中文社区网友：radio2radio

原文网址：<http://www.stmcu.org/module/forum/thread-616613-1-1.html>

更多STM32经验分享：<https://www.stmcu.com.cn/share>