**FreeRTOS学习之六：互斥量**

前提：默认已经装好MDK V5和STM32CubeMX，并安装了STM32F1xx系列的支持包。

硬件平台：STM32F1xx系列。

目的：学习互斥量的使用。

多任务系统中存在一种潜在的风险。当一个任务在使用某个资源的过程中，即还没有完全结束对资源的访问时，便被切出运行态，使得资源处于非一致，不完整的状态。如果这个时候有另一个任务或者中断来访问这个资源，则会导致数据损坏或是其它相似的错误。

考虑如下情形，有两个任务都试图往一个 LCD 中写数据：

• 任务 A 运行，并往串口写字符串“Hello world”。

• 任务 A 被任务 B 抢占，但此时字符串才输出到“Hello w”。

• 任务 B 往串口写“Abort, Retry, Fail?”，然后进入阻塞态。

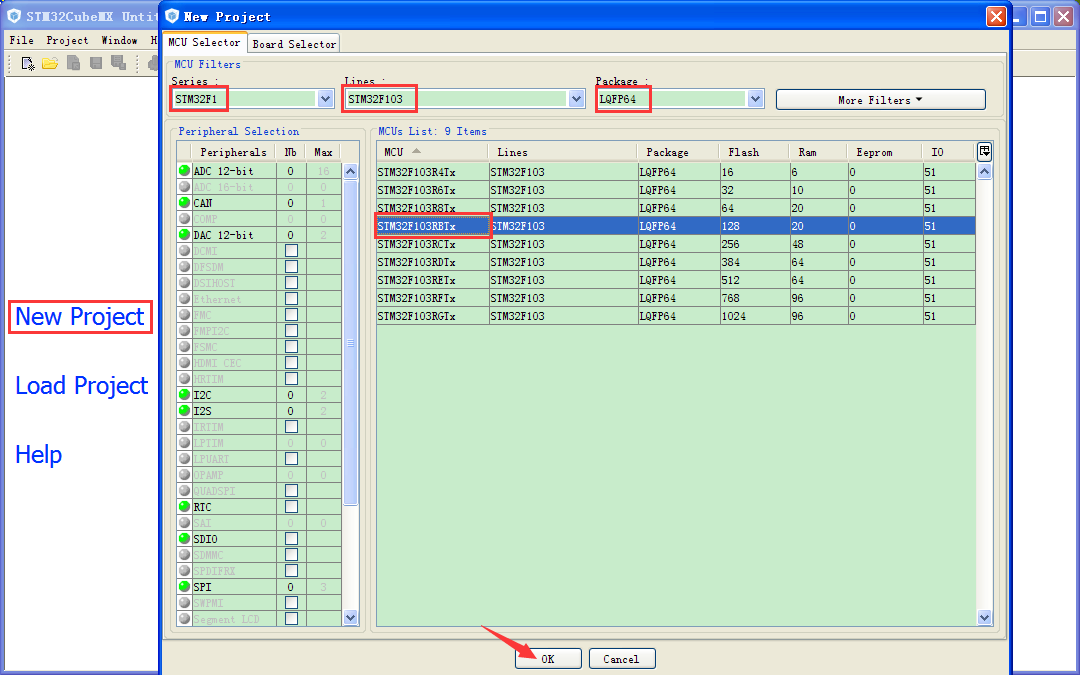
• 任务 A 从被抢占处继续执行，完成剩余的字符输出——“orld”。

现在串口接收端收到的是被破坏了的字符串“Hello wAbort, Retry, Fail?orld”。

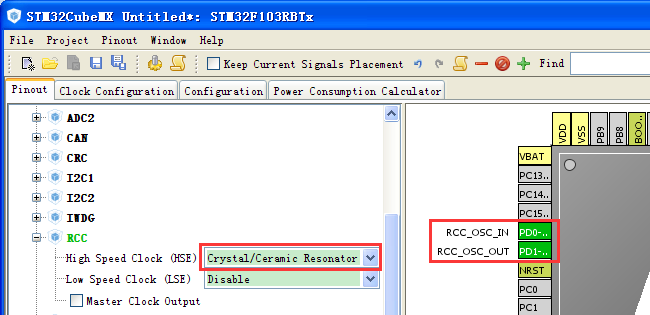
本文例子将再现上述问题，并用互斥量解决该问题。

本文例子使用STM32CubeMX配置创建两个任务，一个任务连续发送字符串“Hello world!”，另一个连续发送字符串“Abort, Retry, Fail?”。

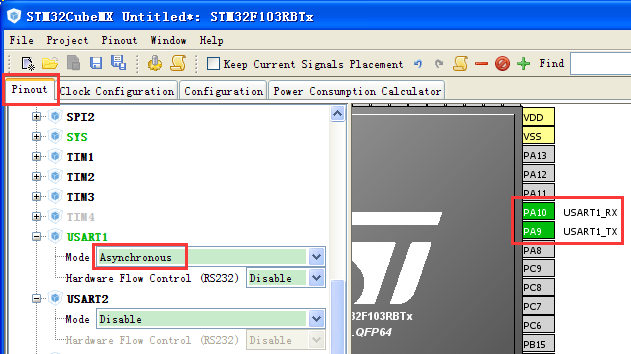
Step1.打开STM32CubeMX，点击“New Project”，选择芯片型号，STM32F103RBTx。



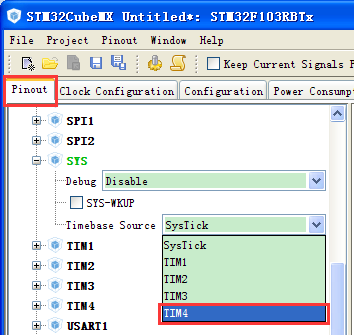
Step2.配置时钟引脚。



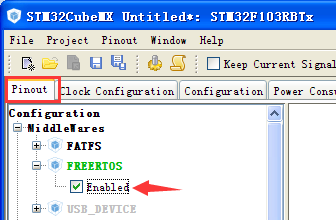
Step3.配置USART1为异步串口。



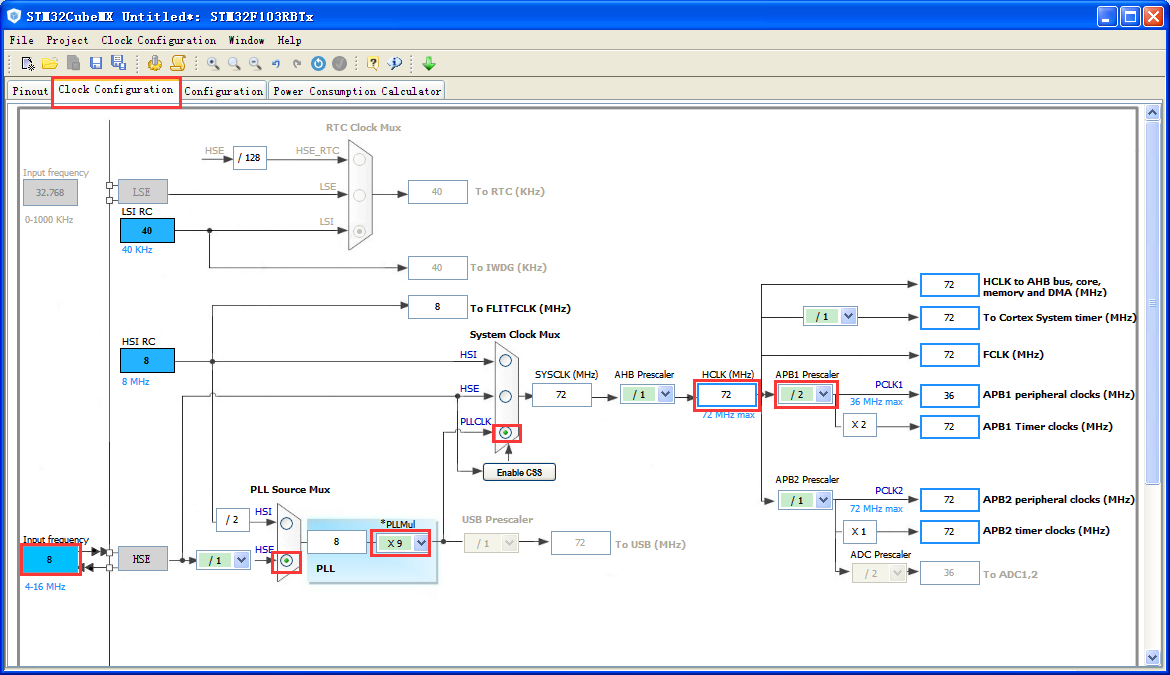
Step4.将系统时基源改为TIM4。



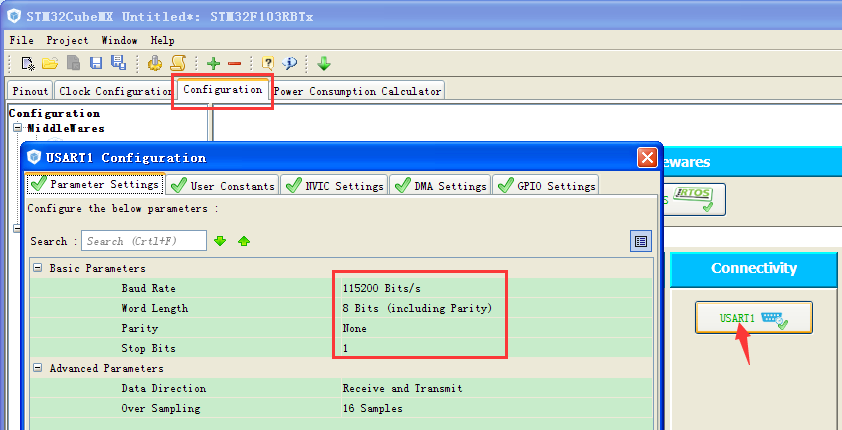
Step5.使能FreeRTOS。



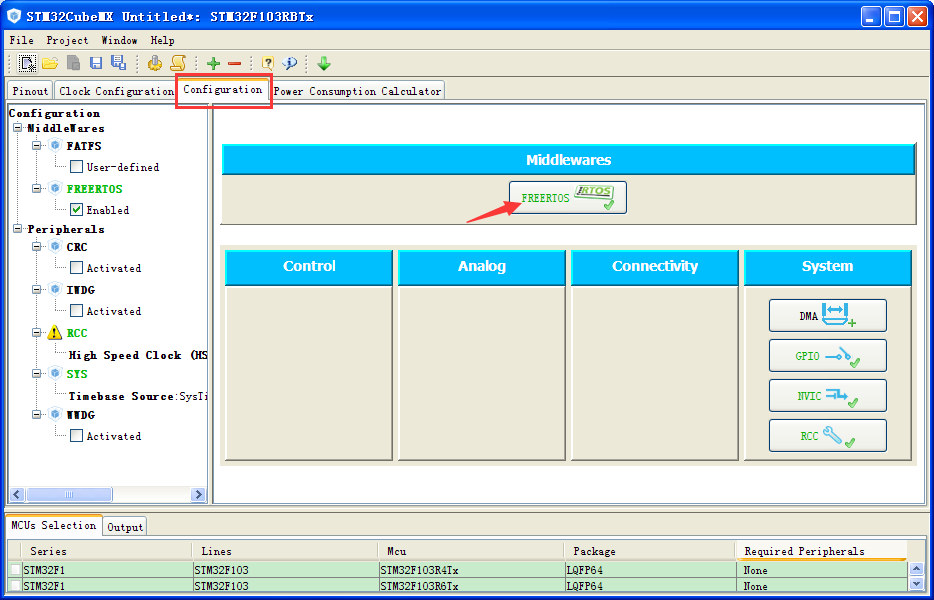
Step6.配置时钟树。8M输入时，通过PLL得到72M内部时钟。



Step7.配置串口参数，使用默认值即可。

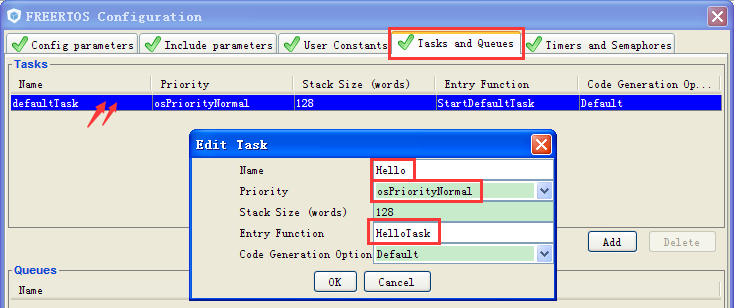


Step8.配置FreeRTOS。

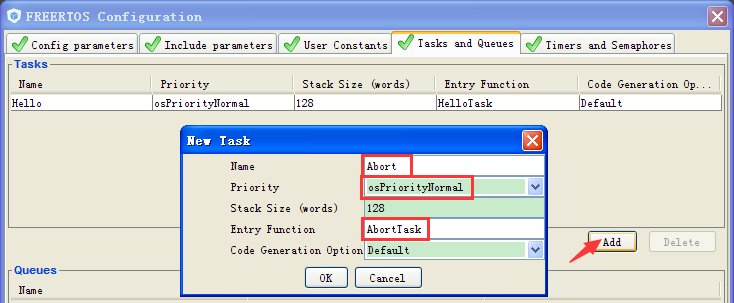


在Tasks and Queues选项卡，默认配置了一个名为defaultTask的任务，其优先级为普通，任务堆栈大小为128字，任务函数名为StartDefaultTask。

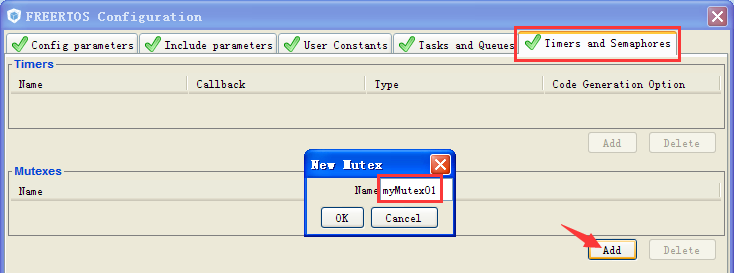
双击蓝色的地方，弹出对话框，将任务名修改为Hello，将任务函数名修改为HelloTask。



点击Add按钮，增加一个任务Abort，优先级设置为Normal，函数名为AbortTask。

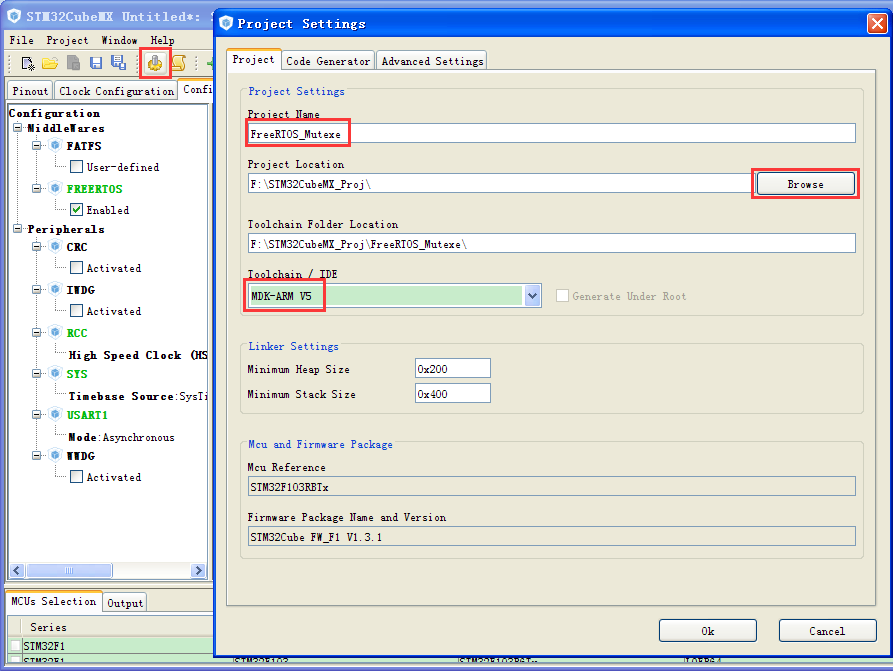


在Timers and Semaphores选项卡，添加互斥量myMutex01。

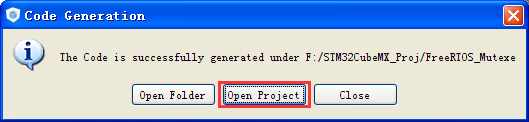


注：其他的都使用默认参数。

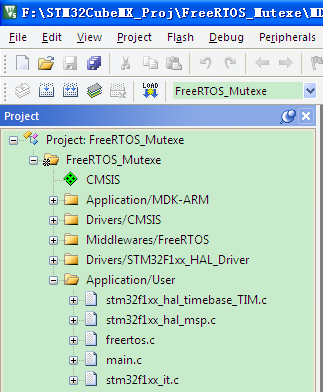
Step9.生成代码。



等完成后直接打开工程。

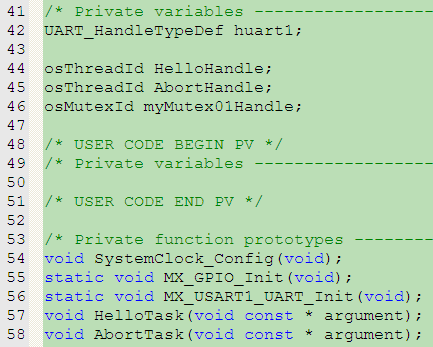


工程基本组织结构如下图，其中Application/User组中的文件是用户可以修改的，而其他组中的文件一般不进行修改。

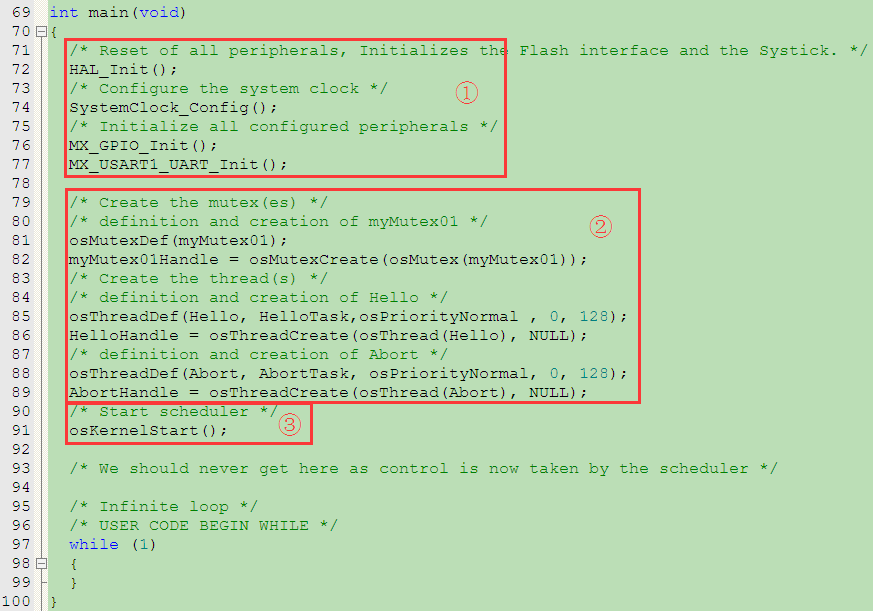


Step10.分析程序结构。

在进入main函数之前，先定义了几个变量，声明了几个函数。



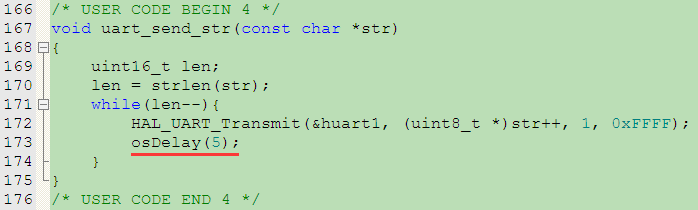
再看main函数。将main函数整理，删除很多注释之后，得到下图所示内容。



其中第①部分，是硬件配置；第②部分，创建了一个互斥量和两个任务；第③部分，启动调度器。

Step11.添加代码。

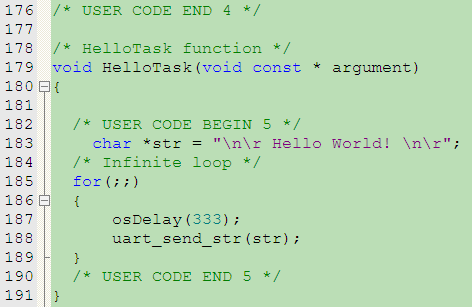
先实现一个串口打印字符串的函数：

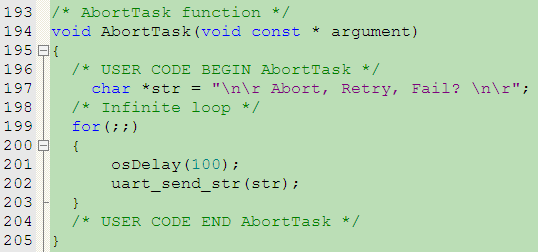


要特别说明的是，因为HAL\_UART\_Transmit()函数本身使用了硬件锁来避免串口共享冲突，所以在此使用了osDelay()函数，目的是人为制造UART共享冲突，以模拟多任务系统中资源共享冲突的情况。

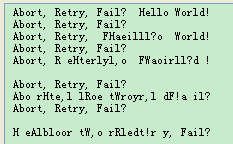
在此因为用到了strlen()函数，所以要在main文件中包含头文件string.h。

在两个任务函数中分别添加功能代码。





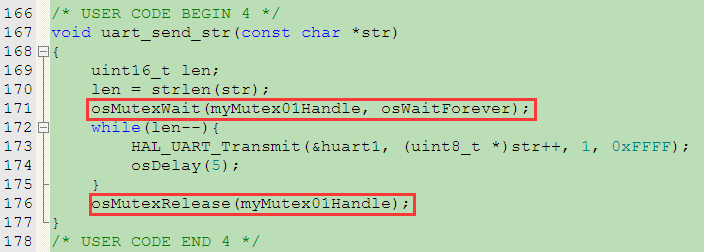
Step12.编译下载运行。通过串口调试助手接收MCU发送的数据，下图是一种可能的结果。



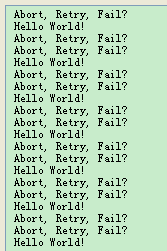
Step13.用互斥量解决共享冲突。

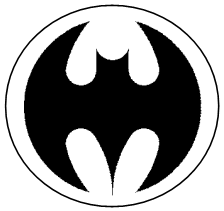
在实际应用中，比如使用LCD输出显示，就可能会遇到上述的共享冲突情况。

把串口打印字符串的函数函数改为：



在使用串口输出前，先申请信号量，输出完字符串后，释放信号量。这样就解决了上面的冲突。下图是输出结果实例。





S.D.Lu 于 深圳

2016年8月