



STM32 嵌入式操作系统动手实验 FreeRTOS

Nov. 2017

目标

- 本动手实验的目的是学习通过STM32CubeMX在STM32F401RE-Nucleo板上生成FreeRTOS工程，通过实验了解任务创建、消息
- 实验分成三个部分：
 - **任务1:** 使用STM32CubeMX 建立一个基于NUCLEO-F40RE板子的FreeRTOS工程，设定好串口用于跟PC通讯
 - **任务2:** 实习FreeRTOS软件定时器
 - **任务3:** 了解消息机制
 - **任务4:** 了解信号量机制

硬件准备

- NUCLEO-F401RE
- Mini USB 线

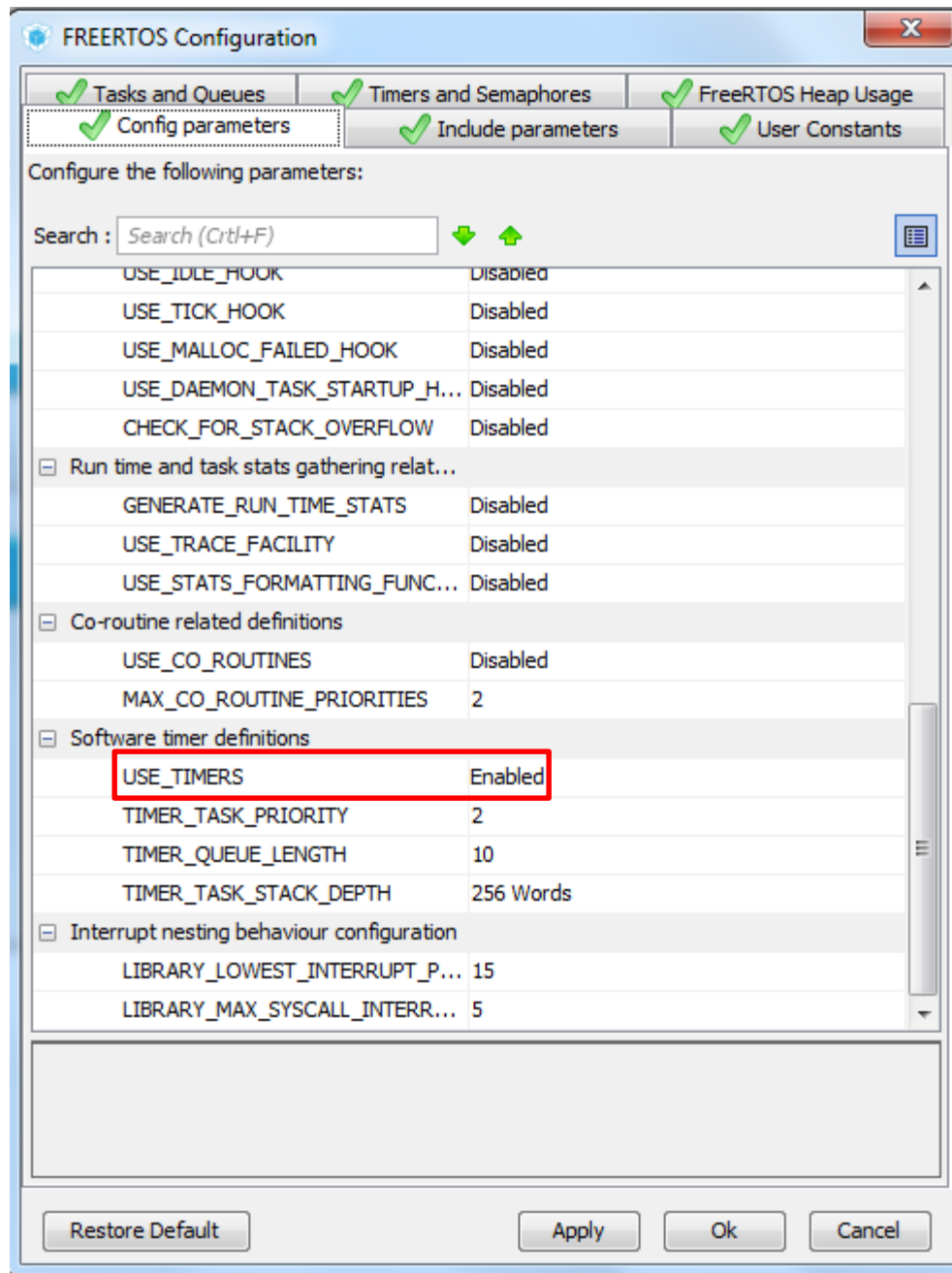



软件准备

- **STM32CubeMX** （U盘培训包中）
- **IAR EWARM 7.8 及以上** （U盘培训包中有**IAR 8.2**, **License**如下
 - **7900-030-271-6938**
 - **7900-030-325-1489**
 - **7900-030-400-7521**
 - 有限期到**2018.1.2**
- **ST-Link Utility** （U盘培训包中）

动手实验步骤（1）

- 1. 使用**STM32CubeMX**创建工程
- 2. 使用**STM32CubeMX**设置使用到的外设，包括**LED**，**UART**
- 3. 使用**STM32CubeMX**创建任务(串口接收，串口发送，**LED**任务)
- 4. 使用**STM32CubeMX**创建软件定时器
- 5. 使用**STM32CubeMX**创建信号量，消息队列




FREERTOS Configuration
✕

✔ Config parameters
✔ Include parameters
✔ User Constants
✔ **Tasks and Queues**
✔ Timers and Semaphores
✔ FreeRTOS Heap Usage

Tasks

Task Name	Priority	Stack Size (Words)	Entry Function	Code Generation...	Allocation	Buffer Name	Control Block Name
RedLED	osPriorityLow	128	RedLED_Task	Default	Dynamic	NULL	NULL
UART_Rcv	osPriorityHigh	128	UART_Rcv_Task	Default	Dynamic	NULL	NULL
UART_Send	osPriorityNormal	128	UART_Send_Task	Default	Dynamic	NULL	NULL

Add
Delete

Queues

Queue Name	Queue Size	Item Size	Allocation	Buffer Name	Control Block Name
uart_rcv_msg	16	uint16_t	Dynamic	NULL	NULL
led_msg	16	uint16_t	Dynamic	NULL	NULL

Add
Delete

Restore Default
Apply
Ok
Cancel

FREERTOS Configuration

X

✔ Config parameters

✔ Include parameters

✔ User Constants

✔ Tasks and Queues

✔ Timers and Semaphores

✔ FreeRTOS Heap Usage

Timers

Timer Name	Callback	Type	Code Generation Option	Allocation	Control Block Name
myTimer01	Callback01	osTimerPeriodic	Default	Dynamic	NULL

AddDelete

Mutexes

Mutex Name	Allocation	Control Block Name
------------	------------	--------------------

AddDelete

Recursive Mutexes

Mutex Name	Allocation	Control Block Name
------------	------------	--------------------

AddDelete

Binary Semaphores

Semaphore Name	Allocation	Control Block Name
uart_Sem	Dynamic	NULL

AddDelete

Counting Semaphores

Semaphore Name	Count	Allocation	Control Block Name
----------------	-------	------------	--------------------

AddDelete

Restore Default

Apply

Ok

Cancel

动手实验步骤（2）

➤6. 生成工程，增加应用代码

1) 软件定时器回调函数

- 检测DMA cnt是否有变化，有-接收到了uart数据

2) 如接收到了uart数据，发送消息给UART串口接收任务

动手实验步骤（3）

➤7. 串口接收任务收到消息后进行解析

- 1) LED-1S - 发消息给LED任务，LED任务控制LED 1S闪烁
- 2) LED-500MS -发消息给LED任务， LED任务控制LED 500MS闪烁
- 3) LED-50MS - 发消息给LED任务， LED任务控制LED 50MS闪烁
- 4) LED-RELEASE - 释放信号量，UART发送任务发送"hello,world!"
- 5) LED-LOCK - 获取信号量，UART发送任务发送"I'm locked!"

动手实验步骤（4）

➤ 8. 串口发送任务获取信号量

成功获取 – 发送“hello,world!”

获取不到 – 发送“I’m locked!”