# NucleoF429 移植 FreeRTOS

这一个帖子只讲怎么把 FreeRTOS 移植到 NucleoF429 上并运行三个 LED 闪烁的任务，至于 FreeRTOS 的相关知识，网上也有许多文章和书籍，这里就不讲了（实际原因是我对 FreeRTOS 也一知半解，希望能和诸位一起讨论学习^_^）
事实上 STM32CubeMx 的自带了 FreeRTOS 的模块，只要勾选后进行一些简单的配置，就可以生成可用的工程，所以这一个帖子除了“重新发明轮子”之外，有一点可取之处就是可以理清移植 FreeRTOS 需要做那些事，不使用 CubeMx 时或者在其他 MCU 上移植 FreeRTOS 该怎么做。

**移植分两步：**
Step 1、使用 F4 的固件库建立一个工程模板。
Step 2、在新建的工程上移植 FreeRTOS。

在移植之前需要下载 F4 的固件库，我使用的固件库版本是：
【STM32F4xx_DSP_StdPeriph_Lib_V1.7.1】。可以到 ST 的官网上下载，最新版地址：
http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32-standard-peripheral-libraries/stsw-stm32065.html
还需要 FreeRTOS 的 Source Code，我使用的版本是 FreeRTOSV8.2.3，FeeRTOS 官网：http://www.freertos.org ，可自行下载。

## 第一步：建立工程模板

具体的建立过程和文件夹结构如下：
1、新建文件夹 NucleoF429_FreeRTOS。
2、在 NucleoF429_FreeRTOS 下新建三个文件夹 Project 和 SysLib 和 User。
3、在 SysLib 新建四个文件夹：Startup、CMSIS、inc、src。
接下来在各文件中添加对应的文件
Startup：复制 startup_stm32f429_439xx.s 文件到 Startup 中，该文件位于 ...\Libraries\CMSIS\Device\ST\STM32F4xx\Source\Templates\arm 中。

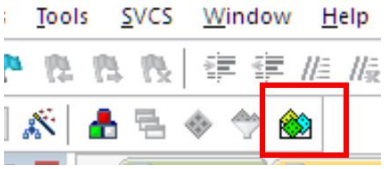
CMSIS：复制 stm32f4xx.h、system_stm32f4xx.h、system_stm32f4xx.c 到 CMSIS 中。上述文件位于...\Libraries\CMSIS\Device\ST\STM32F4xx\Include 和...\Project\STM32F4xx_StdPeriph_Templates 中。

Inc: 复制...\Libraries\STM32F4xx_StdPeriph_Driver\inc 中的.h 文件到 Inc 文件夹中，也可以按实际需要的外设选择相应的文件。

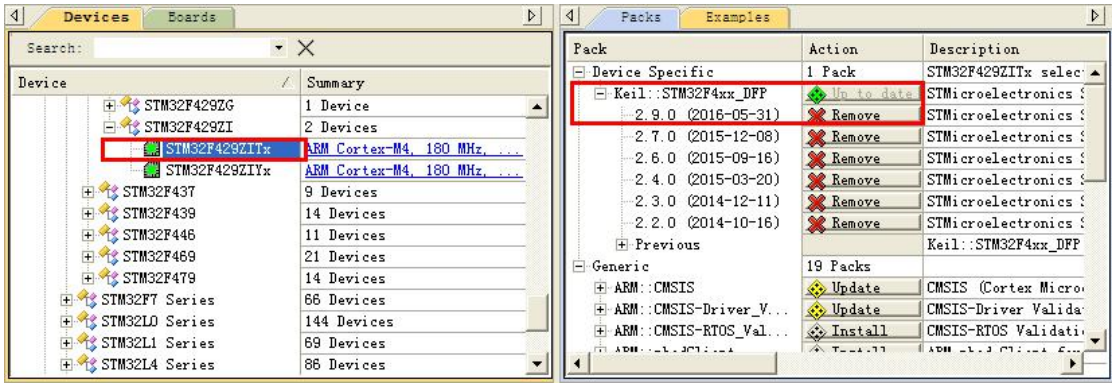
Src：复制...\Libraries\STM32F4xx_StdPeriph_Driver\src 中的.c 文件到 src 文件夹中，与 inc 文件对应，按需选择。

User：复制 stm32f4xx_conf.h、stm32f4xx_it.c、stm32f4xx_it.h 三个文件到 User 中，该文件位于...\Project\STM32F4xx_StdPeriph_Templates
在 User 中建立一个 main.c 文件，包含 stm32f4xx.h 即可。

工程需要的文件已添加完成，现在开始建立 Keil 工程，在建立 Keil 工程之前，先查看你的 Keil 是否有 STM32F429 所需要的库。如图：



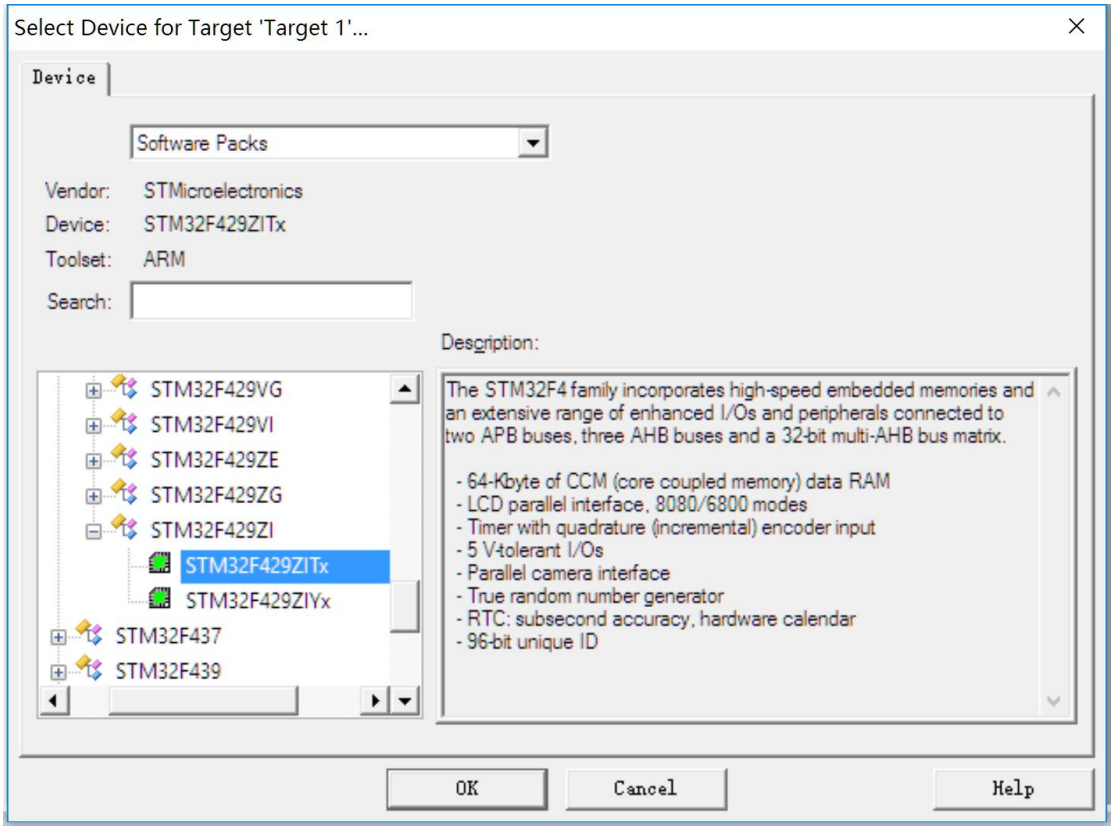

如果没有 STM32F4 的库就需要添加，第一种方式是在这个 Pack Installer 中添加




第二种是到 Keil 官网先下载 F4 的.Pack 文件，再用 Pack Installer 导入。第一种方式下载速度慢且不支持断点续传，往往下载到一半就提示下载失败，所以推荐第二种方式。Keil 的 Pack 文件下载地址是：http://www.keil.com/dd2/Pack 找到需要的文件下载后再用 Pack Installer 导入即可。

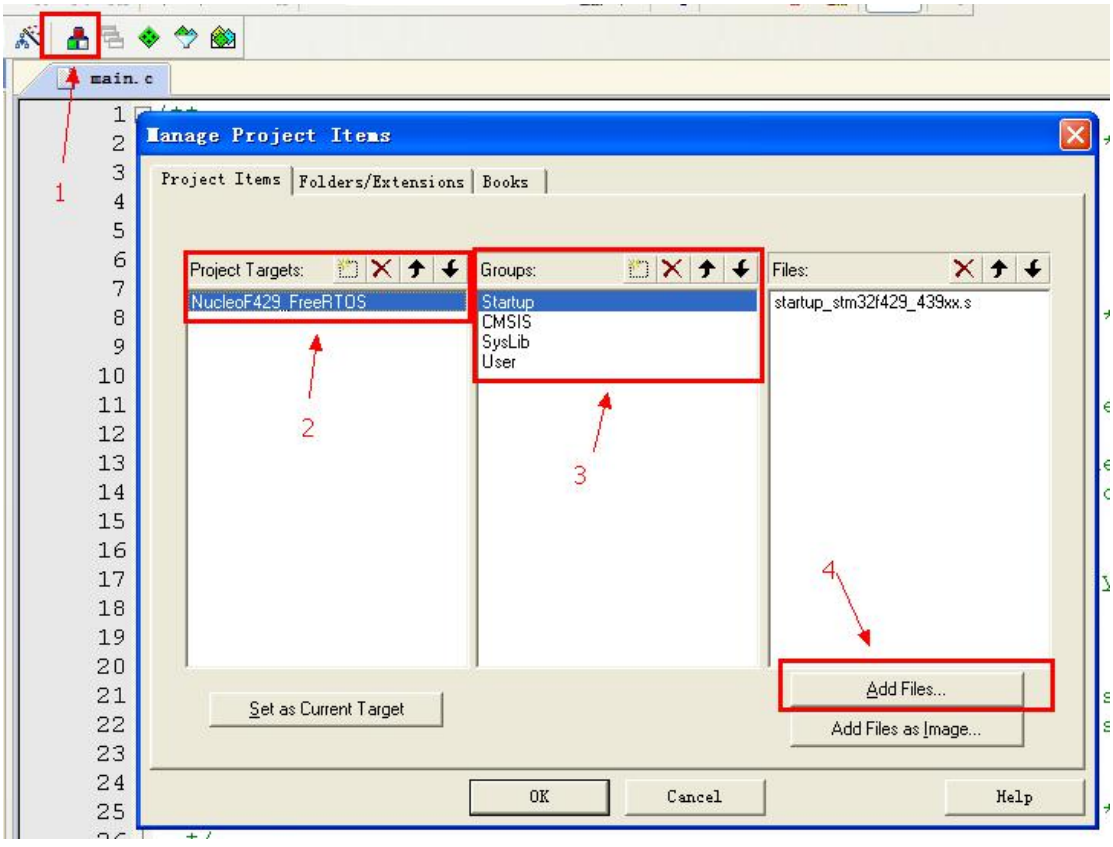
接下来建立工程：

Project-->New uVision Project 选中 NucleoF429_FreeRTOS 下的 Project 文件夹，工程命名为 NucleoF429_FreeRTOS 点击保存，在弹出的 Select Device 窗口选中 ST 的 STM32F429ZITx 型号后点击 OK




这时会有另一个窗口弹出，不用理会直接 OK 即可。

按顺序编辑管理结构树：




四个文件中添加的文件为：

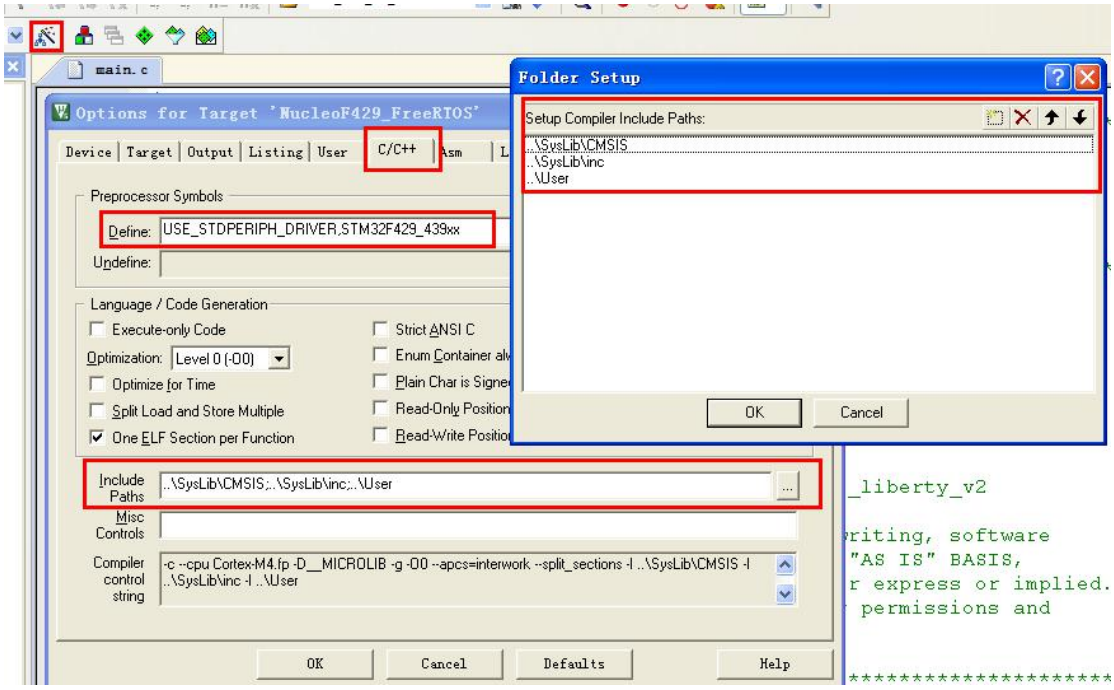
Startup：添加文件夹 Startup 中的.s 文件，注意.s 默认不会显示出来，在文件类型里选择.s 格式

CMSIS：添加文件夹 CMSIS 中的.c 文件。

SysLib：添加文件夹 src 中的.c 文件。

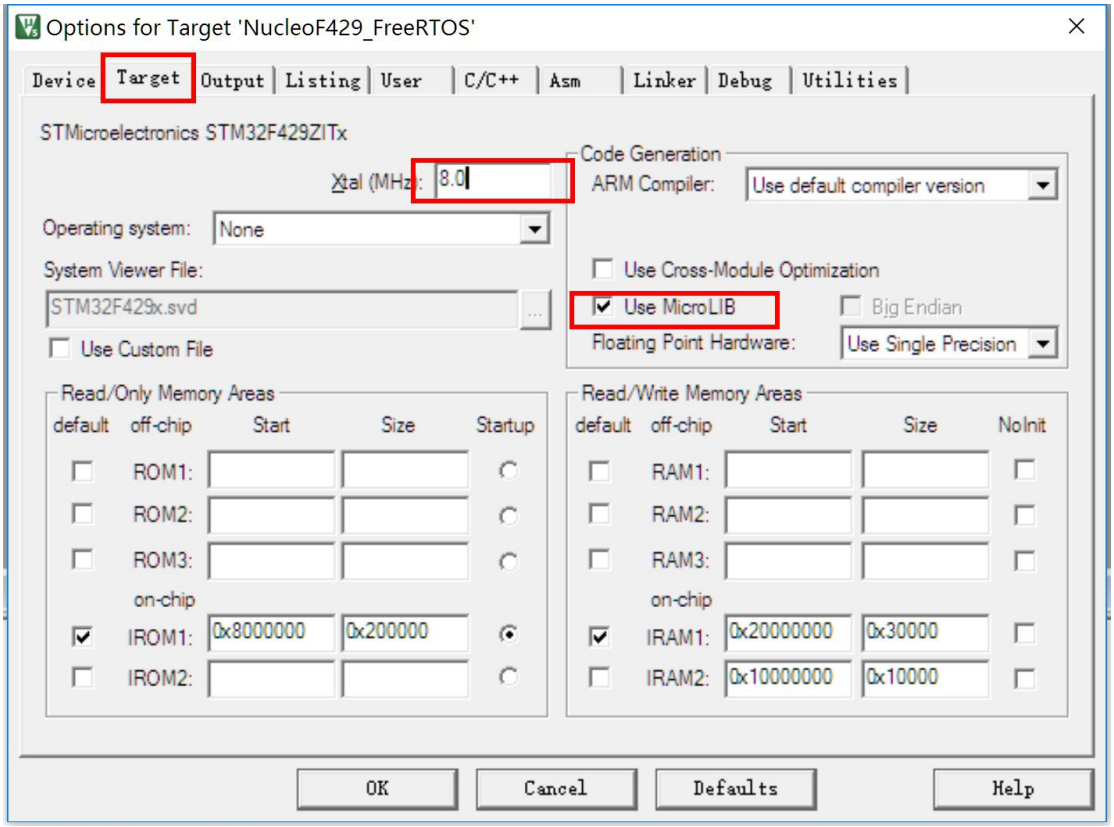
User： 添加文件夹 User 中的.c 文件。

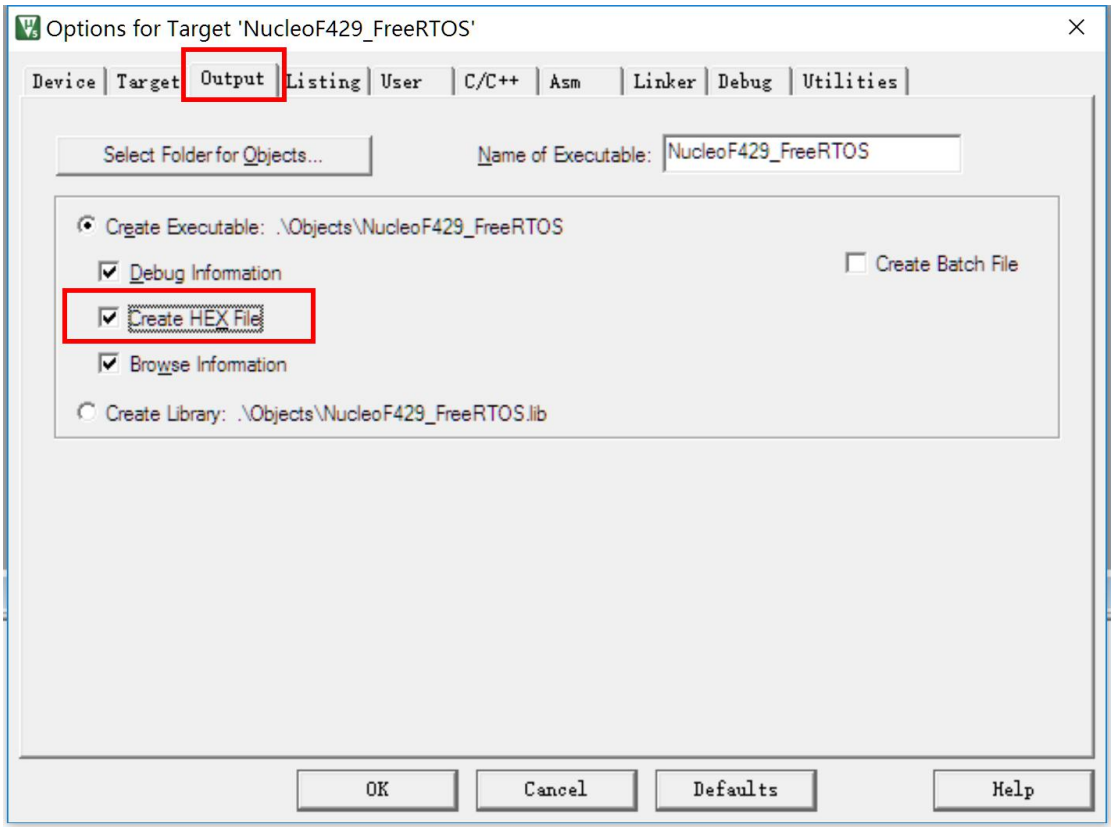
完成后再进行头文件搜索路径设置

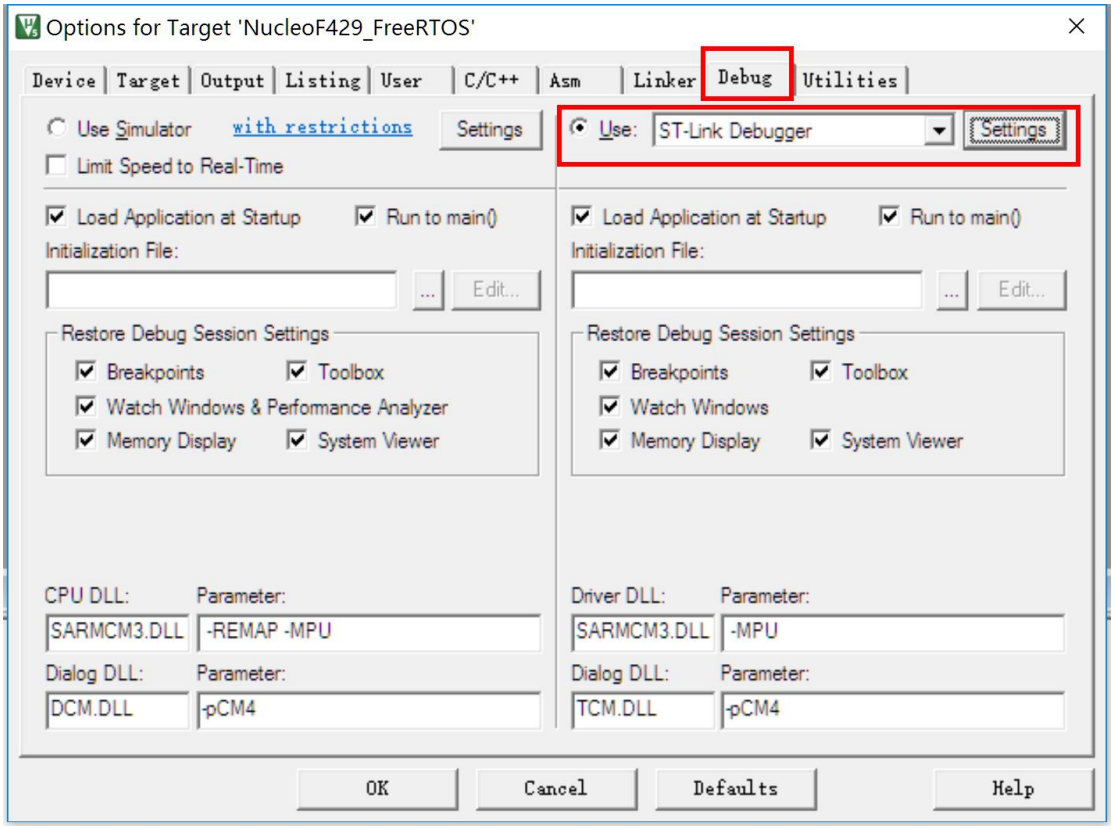



注意 Define 这一栏填写：【USE_STDPERIPH_DRIVER,STM32F429_439xx】

之后再进行一些其他设置：


Options for Target 'NucleoF429_FreeRTOS'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
STMicroelectronics STM32F429ZITx
Xtal (MHz): 8.0
Code Generation
ARM Compiler: Use default compiler version
Operating system: None
System Viewer File:
STM32F429x.svd
Use Custom File
Use Cross-Module Optimization
Use MicroLIB
Big Endian
Floating Point Hardware: Use Single Precision
Read/Only Memory Areas
default off-chip Start Size Startup
ROM1:
ROM2:
ROM3:
on-chip
IROM1: 0x8000000 0x200000
IROM2:
Read/Write Memory Areas
default off-chip Start Size NoInit
RAM1:
RAM2:
RAM3:
on-chip
IRAM1: 0x20000000 0x30000
IRAM2: 0x10000000 0x10000
OK
Cancel
Defaults
Help


Options for Target 'NucleoF429_FreeRTOS'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
Select Folder for Objects...
Name of Executable: NucleoF429_FreeRTOS
Create Executable: .\Objects\NucleoF429_FreeRTOS
Debug Information
Create HEX File
Browse Information
Create Batch File
Create Library: .\Objects\NucleoF429_FreeRTOS.lib
OK
Cancel
Defaults
Help

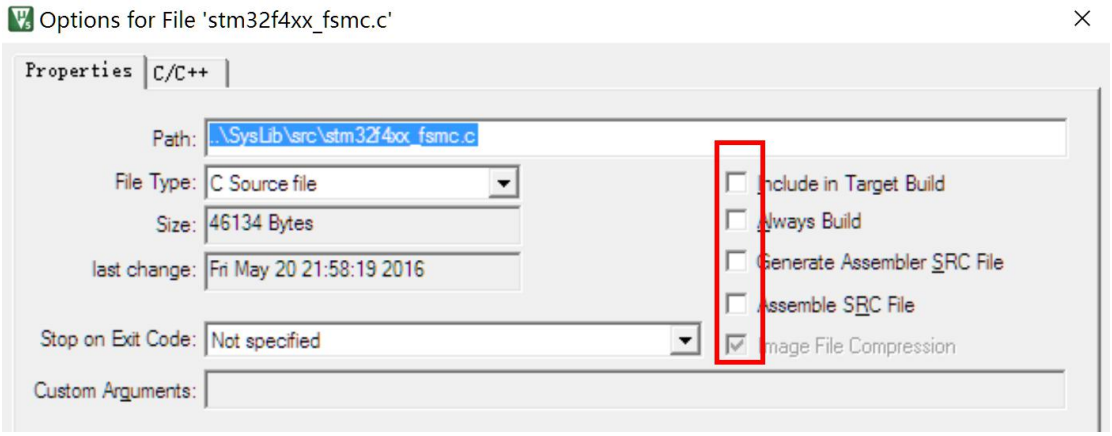



调试方式选择 SWD 模式。

到此，基本的设置和建立都已完成，可以进行编译。

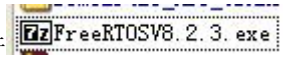
但这时候会报 identifier "FSMC_Bank1" is undefined 等 30 个错误，这是由于 STM32F4xx.h 中没有定义 FSMC_Bank1 等，只有 FMC_Bank1，不知道这是 Bug 还是我哪里设置不对，因为现在暂时不用 FSMC，所以将 stm32f4xx_fsmc.c 排除，不进行编译。

具体操作为，在 SysLib 下找到 stm32f4xx_fsmc.c，右键 options for file 'stm32f4xx_fsmc.c' 去掉下图中的四个勾，再进行编译就没错了



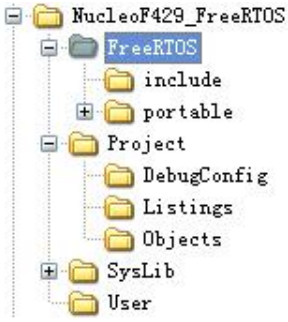

到此为止，已经建立了一个空白的 F429 工程模板。

## 第二步：移植 FreeRTOS

1、FreeRTOS 下载下来是一个执行文件 

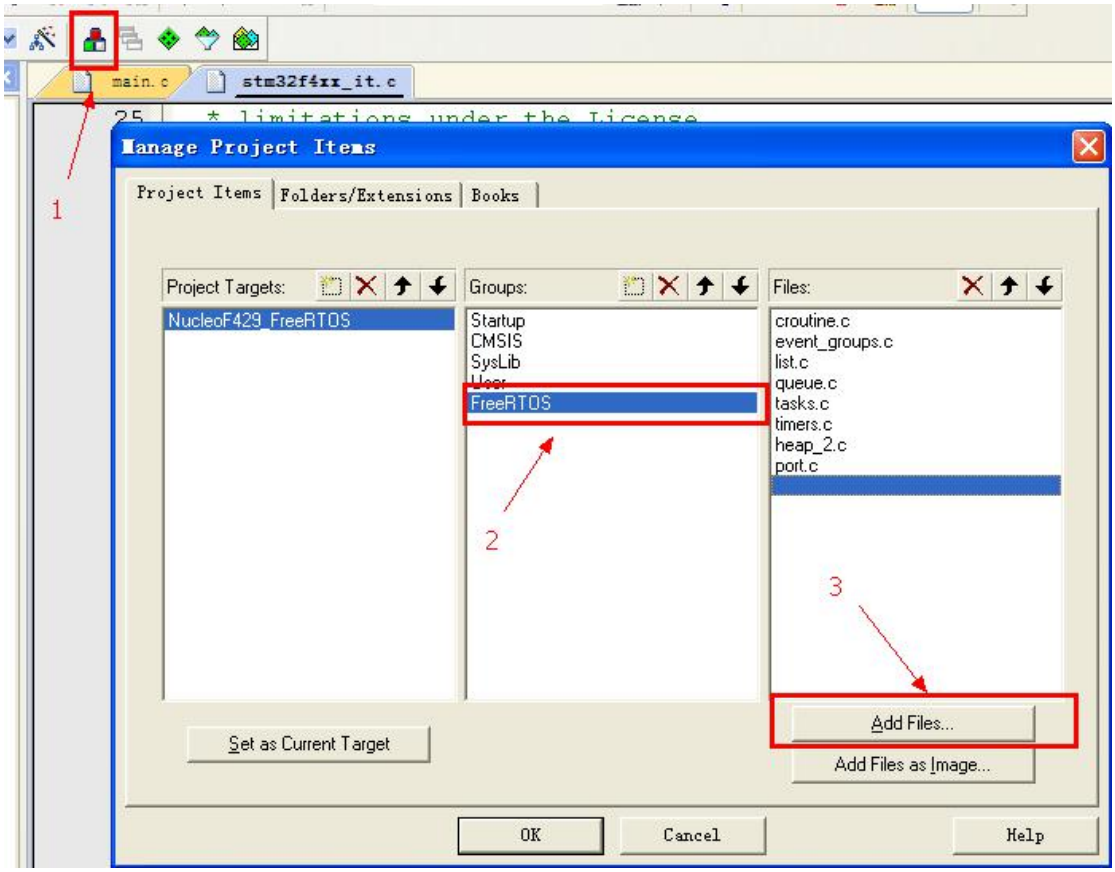
，点击运行后会解压出 Source Code。我们只是用 Source 文件夹里的文件。


2、在工程模板文件夹下建立 FreeRTOS 文件夹，将解压出的 Source 中的文件全部复制到新建的 FreeRTOS 中，

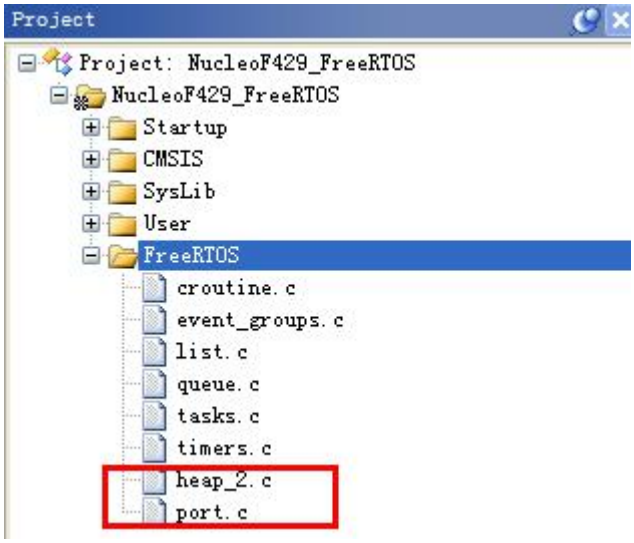




3、在 Keil 工程里添加 FreeRTOS 需要的文件：

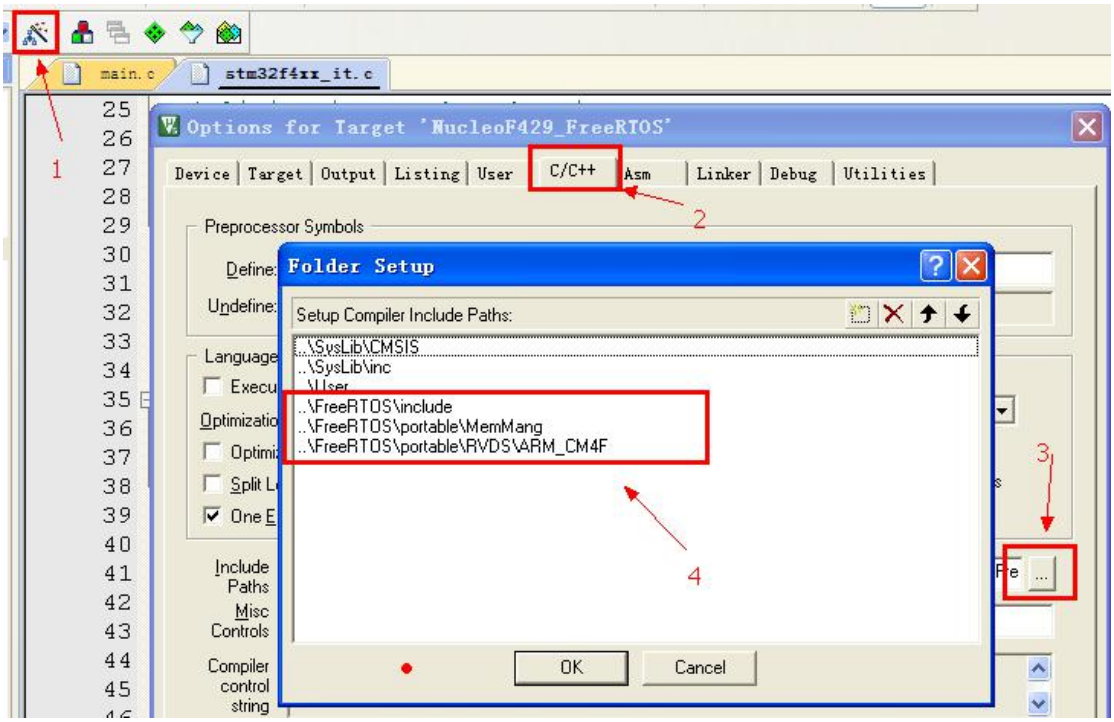



除了 FreeRTOS 文件夹中的所有.c 文件以外，还需要 Heap_2.c 和 Port.c 文件，这两个文件分别位于：...FreeRTOS\portable\MemMang 和...\FreeRTOS\portable\RVDS\ARM_CM4F 中




除了以上的 C 文件，还需要一个 FreeRTOS 的配置头文件，FreeRTOS 的自带的 Demo 中有 STM32F407 的例程，复制...\FreeRTOSV8.2.3\FreeRTOS\Demo\CORTEX_M4F_STM32F407ZG-SK 下的 FreeRTOSConfig.h 到工程的 Include 中。

4、添加 FreeRTOS 的头文件搜索路径。




以上基本完成了移植工作，但还有一些编译错误需要解决，可能是由于版本的原因和移植方式的原因，之前在 STM32F103 上移植是并没有多少错误，但这次 F429 上多了一些。

FreeRTOS 的一些中断服务程序需要代替 F429 自带的程序，比如在 FreeRTOSConfig.h 定义了这三个中断服务程序，就需要把 STM32F4xx_it.c 中的 SVC_Handler()/PendSV_Handler()/SysTick_Handler()这三个程序屏蔽掉，否则会报重复定义错误。（另一中解决办法是修改.s 文件中的中断向量表，网上大多数采用这种方法。）

```
167 #define vPortSVCHandler SVC_Handler
168 #define xPortPendSVHandler PendSV_Handler
169 #define xPortSysTickHandler SysTick_Handler
```

另一类错误就是 FreeRTOSConfig.h 定义了一些开关量，但实际的程序实现并没有给出，需要用户自己编写，所以我将这些都定义为 0，默认不编译。

在 FreeRTOS 里添加任务之前，需要改一下默认的系统时钟，否则时间不准。在进入 main 之前，程序默认初始化了系统时钟，都是按照外接 25MHz 设置到 180Mhz 的，NucleoF429 默认无外接高速晶振，所以要改变分频，在 system_stm32F4xx.c 中 PLL_M 由原来的 25 为 8 即可.

编译无错后，可以添加任务了。

我使用了板子上自带的三个 LED，实现不同频率的闪烁。

步骤分别是 1、硬件初始化，2、创建任务，3、启动任务序列。

```
    prvSetupHardware(); //Hardware Init
    xTaskCreate( vTask1, "Task_1", 1000, NULL, 1, NULL );
    xTaskCreate( vTask2, "Task_2", 1000, NULL, 2, NULL );
    xTaskCreate( vTask3, "Task_3", 1000, NULL, 2, NULL );
    vTaskStartScheduler(); //Start Task
```

其中，任务的实现为：

```
//-------------------------------------------------------------------//
void vTask1( void *pvParameters )
{
    while (1)
  {
    GPIO_ToggleBits(GPIOB,GPIO_Pin_0);
    GPIO_ToggleBits(GPIOC,GPIO_Pin_10);
    vTaskDelay(200);
  }
}

//-------------------------------------------------------------------//
void vTask2( void *pvParameters )
{
    while (1)
  {
    GPIO_ToggleBits(GPIOB,GPIO_Pin_7);
    GPIO_ToggleBits(GPIOC,GPIO_Pin_11);
    vTaskDelay(400);
  }
}

//-------------------------------------------------------------------//
void vTask3( void *pvParameters )
{
    while (1)
  {
    GPIO_ToggleBits(GPIOB,GPIO_Pin_14);
    GPIO_ToggleBits(GPIOC,GPIO_Pin_12);
    vTaskDelay(800);
  }
}
```

FreeRTOS 在遇到延时时会自动让出 CPU，所以三个任务会轮流执行。GPIOC 是为了测量时间。以上是建立工程和移植 FreeRTOS 的全部过程。如果里面有错误的地方，欢迎各位留言讨论。工程源码在帖子附件。

----本帖发自 ARM 中文社区