FreeRTOS移植笔记

0：本文是FreeRTOS的移植笔记

目标是将FreeRTOS系统移植到中微MCU的BAT32A237 MCU上。

1：首先要有一个简单的基础工程，作为移植的基础，可以是GPIO的，可以是UART的。

2：将FreeRTOS中需要的文件加入到工程中。

主要有：

FreeRTOSversion\FreeRTOS\Source\文件夹下的所有文件。

每一个应用Demo里面都有一个FreeRTOSConfig.h文件，选择和自己目标平台相同或者相近的文件拷贝过来待修改使用。

3：将需要文件加入到工程中。

主要有：

FreeRTOSversion\FreeRTOS\Source\文件夹下的所有.c文件。

FreeRTOSversion\FreeRTOS\Source\portable\文件夹下的port.c和portmacro.h,找各自的编译器和处理器对应的子文件夹。

FreeRTOSversion\FreeRTOS\Source\portable\MemMang\文件夹下的heap\_4.c文件。

也要根据情况增加对应的头文件路径。

4：初步修改

4.1：在FreeRTOSConfig.h中要使用stdint.h中的内容，还要使用SystemCoreClock；

所以要根据自己的情况，添加声明文件。类似如下：

/\* Ensure stdint is only used by the compiler, and not the assembler. \*/

#ifdef \_\_ICCARM\_\_

#include <stdint.h>

extern uint32\_t SystemCoreClock;

#endif

#ifdef \_\_CC\_ARM

#include <stdint.h>

//#include "Nano100Series.h"

#include "BAT32A237.h"

#endif

4.2：在FreeRTOS中已经定义了SysTick\_Handler函数，所以要将自己的对应函数禁止掉。注意RTOS将这个函数的名字重新定义为了xPortSysTickHandler，所以名字变了。

同样的情况还有SVC\_Handler和PendSV\_Handler;都要注意将自己的函数禁止掉。

/\* Definitions that map the FreeRTOS port interrupt handlers to their CMSIS

standard names - or at least those used in the unmodified vector table. \*/

#define vPortSVCHandler SVC\_Handler

#define xPortPendSVHandler PendSV\_Handler

#define xPortSysTickHandler SysTick\_Handler

4.3：编译器可能报告没有定义vApplicationIdleHook函数，处理方式可以禁止configUSE\_IDLE\_HOOK宏定义，也可以自己定义一个函数void vApplicationIdleHook( void );

vApplicationTickHook函数和configUSE\_TICK\_HOOK宏，有类似情况。

4.4：到此很可能已经可以编译通过了。如果依然有问题，根据编译报告的情况做对应的修改，总之要将FreeRTOS编译进工程中。

5：FreeRTOSConfig.h修改

这个文件是配置FreeRTOS的各方面的性质的，最好看一下它的一些宏，根据自己的具体情况，做一些简单的开关动作。

6：初始化系统，建立任务，应该就可以运行了。

注意，一般初始化的时候，会设置并且启动SysTick定时器，如果先启动了定时器再做别的工作，如果在定时中断到来的时候，还没有建立好任何，则系统不会正常运行，会在中断处理任务的时候异常。所以要延迟设置并启动SysTick定时器。

正常情况下，应该可以运行任务了。