**FreeRTOS CLI移植记录**

# 一、目的

为什么非要用CLI，最主要的目的就是为了向操作Linux系统一样，就比如说我要配置一个DDS，那么如果我用协议的话，还需要自己写协议，还需要自己编写上位机，就比较麻烦，总之CLI具有非常大的优势。

# 二、FreeRTOS CLI移植过程

## 1，操作步骤

包含所需头文件，然后发现configCOMMAND\_INT\_MAX\_OUTPUT\_SIZE没有定义。查了查资料发现可能是缺少文件，我是按照文章[STM32环境下的FreeRTOS+CLI V1.0.4的移植\_stm32 使用 freertos\_ cli例子\_￥风笛￥的博客-CSDN博客](https://blog.csdn.net/lastliudexi/article/details/105759334)来看缺少哪些文件。只不过serial.c调用的是stm32f103的，不知道是否需要后续修改。然后手动宏定义最大输出大小为1024，定义在了FreeRTOS\_CLI.h上

2，发现serial.c中有报错

原来是用的f103的标准库，那么用h7的话，就用LL库就行了。然后发现更改完之后其它的错误也没了，很不错。其实主要是编译器问题。之前的错误都在FreeRTOS\_CLI中，然后起始跟踪错误的地方是有定义的，只不过因为定义的头文件包含层数过深，编译器无法检测到，所以报错。但是编译过程是没问题的，如果嫌烦，可以再手动包含一下，这里我就手动包含了一下”protmacro.h”，然后错误就全没了

3，初始注册函数宏定义

参考[【FreeRtos任务运行时间统计】\_portget\_run\_time\_counter\_value()\_小殷学长的博客-CSDN博客](https://blog.csdn.net/boybs/article/details/124781763)

在vRegisterSampleCLICommands命令里，有几个宏没有定义，导致一些重要的命令如tast-stats没有注册上，因此需要把宏定义一下。定义之后发现缺了很多函数，需要自己重新定义，于是我重新配置了一下CubeMX，主要是配置TIM3为FreeRTOS提供时基

4，TIM3时基定义

定时器3挂在APB1总线上。就是200Mhz.200分频就是1Mhz，计数到10000，就是每10ms中断一次,优先级设为1,0

5，vtracestop等函数没有定义

最后还是把trace宏定义给去掉了，然后执行其它的。

6，出现rror: A1586E: Bad operand types的问题，在port.c中

这是因为进行操作时，把优先级设置成4U导致的，把U删掉就好了。请参照这个文章[FreeRTOS STM32CubeMX port.c(483): error: A1586E: Bad operand types (UnDefOT, Constant) ...\_EmbededCoder的博客-CSDN博客](https://blog.csdn.net/u012308586/article/details/94977190)

最终成功了，但是任务无限制的执行，具体可以运行看一看。重新看手册，然后后续再试试吧，盲目试下去没有好结果。最终命令行无限制输出解决了，是因为serial.c的串口中断函数出问题了。修改过来就好了。

但是输入一次命令会输出两次结果。

***以上内容均由Firmware1实现，下面重新看手册构建Firmware2***

# 三、Firmware\_V2

1，我首先把源文件添加到工程，但是有两个文件，Sample-CLI-commands.c和UARTCommandConsole.c没有对应的头文件，因此我把这两个文件拆分了一下，新建了两个头文件，以防这两个c文件没有头文件而无法被别的文件调用。

但是Firmware\_V2仍然有输入一次命令输出两次结果的情况，看样子FreeRTOS的CLI还是比较鸡肋，换Github开源项目nr\_micro\_shell试试

# 四、nr\_micro\_shell移植测试

步骤参照：[nr\_micro\_shell在STM32+FreeRTOS平台的移植和使用\_freertos有shell吗-CSDN博客](https://blog.csdn.net/luobeihai/article/details/126627158)

1，没有结果输出

按照步骤进行，没有错误，只是没有结果输出，挨个测试了一下发现是printf有问题，修改了一下就有输出了。但是发送命令没反应，应当用信号量对串口接收数据做一下改造，也就是接收数据后，给shell函数一个信号量，执行shell函数，然后这样是比较合理的。

2，串口接收无法中断

只是使能串口中断了，但是没有使能串口接收中断，重新加一条语句就好了。

3，信号量释放错误

可能是因为用的二值信号量，串口中断太快导致释放的信号量还没有被接收又被再次释放，导致信号量拥堵释放失败，应当采用freertos的direct to task notification机制来进行处理。

4，把信号量去掉了，然后把串口重构了一下，重构成了循环队列。然后使用的时候发现stm32一直进入hardfault，仔细查看原来是定义的指针没有进行malloc，之前没有这种情况，可能是因为引入freertos的原因，以后定义指针都要进行malloc，不过这个指针是什么类型的。

5，USART1\_Data\_Array结构体初始化之后，从其中的一个数组读取一个数据后，结构体地址自己变化，从而结构体数据也全部变了。

解决办法：把编译器优化等级改为-O0就可以了。这应该涉及到编译器优化过程以及单片机内存分配、程序编译的过程，后续应当对此写一个博客。不必写博客了，因为有人写过这个博客了。这里只需记录一下指定优化等级的操作。

-O3：优化出错。

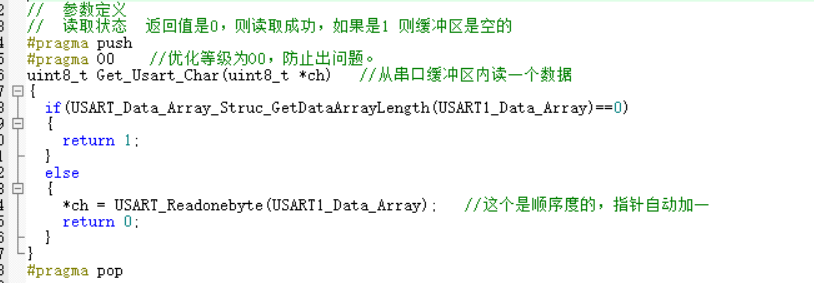
-O2：优化出错。

-O1：优化出错。

-O0：优化正确。

为什么会有这种结果呢？肯定是malloc有问题，一般是没问题的，唯一有联系的肯定是freeRTOS，地址偏移很可能是freeRTOS自己的heap\_4.c有关系。

因为freeRTOS会自己整理内存碎片，而malloc不在freerRTOS中有定义，因此或许采用freeRTOS提供的pvPortMalloc函数应该会比较好。发现仍然不行，这可能和SRAM的读写策略有点关系。最后还是选择了改优化等级，只不过是将出问题的这部分函数的优化等级改了，可以这样写。全局优化等级为O2，仅仅这部分函数优化为O0。



经过验证，这样是正确的。

# 五、终端软件选择 mobaXterm

Nr\_micro\_shell项目里用的终端是hyper terminal 14.0，但是在win10系统闪退了，因此选用mobaXterm作为终端软件。

但是为了和linux终端一直，需要进行一些配置。

1. 在终端内右键 change terminal setting。然后local echo和local line editing都选用force on。
2. 发送的时候不是enter，而是ctrl+enter。
3. 配置串口的时候把流控关掉。效果如下。



如果要使用xshell的话，应当在vt模式里把echo mode打开。

# 六、命令函数编写与注册

终端和单片机shell都移植完毕，下一步应当编写自己的命令，目的是为了输入led on的时候led亮，输入led off的时候led灭。

首先应当定义自己的功能函数，编写完毕后再注册一下就好了。

成功！！！！！！！！