u/COS\_II操作系统移植资料整理

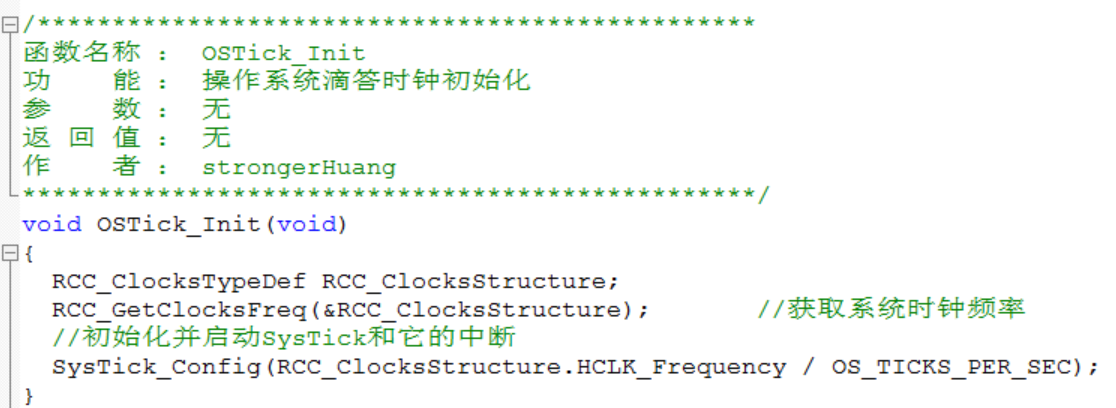
1. 硬件平台：基于STM32F103系列开发板；
2. 下载u/COS\_II源码，提取工程中有用的文件，下载链接：<https://www.micrium.com/download/micrium_stm32f107_ucos-ii>
3. 部分文件介绍：

* app\_cfg.c:应用的配置文件（例如任务优先级，堆栈大小等）；
* app\_task.c:执行具体的各个子任务，设置启动任务，自定义各个任务；
* os\_cfg.h:系统配置的文件（使能相应功能函数，就是所谓的系统使能裁剪）；
* stm32f10x\_it.c:主要中断函数SysTick\_Handler(void)中断函数；

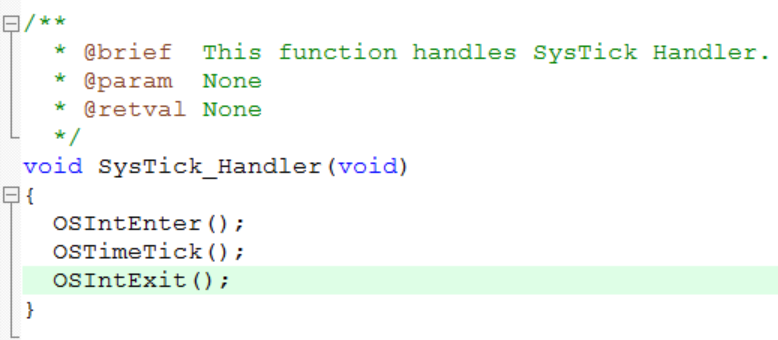
1. u/COS\_II文件夹内容介绍：

* Ports：是与系统息息相关的端口配置文件；
* Source: 操作系统的全部文件；
* BSP：所有底层配置的初始化，例如管脚输入输出性能，时钟初始化，中断向量表初始化等等；
* APP:应用层文件，主要实现各个子任务的操作；

1. 配置系统滴答时钟；



中断函数需要调用系统相关的函数，也就是系统进行滴答相关的函数，这里是规定这么调用的。

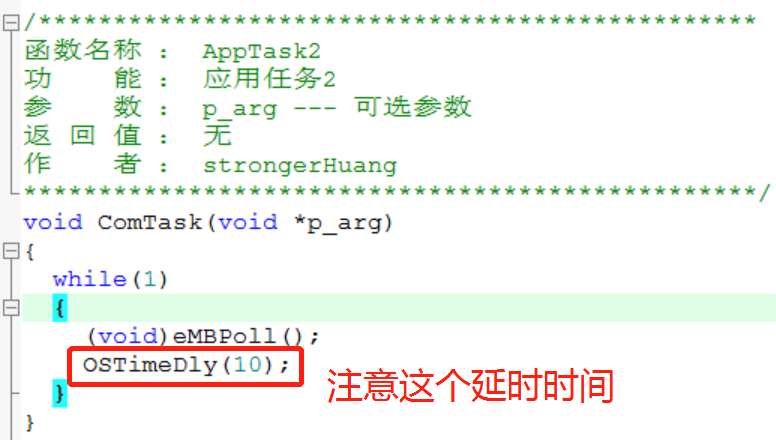


1. 系统的可裁剪，其实就是对默写功能的打开和关闭，集中在os\_cfg.h文件中。关注几个重要参数：

* 最低优先级OS\_LOWEST\_PRIO:配置任务优先级，最低优先级任务配置最大值。优先级越小，数值越大，**相反优先级越大，数值越小**；
* 系统每秒滴答数目OS\_TICKS\_PER\_SEC：系统每秒钟滴答的次数，它与系统延时（OSTimeDly）关系密切，OSTimeDly(100)相当于延时100个滴答，即延时1s的意思；
* 任务堆栈大小OS\_TASK\_XXX\_STK\_SIZE,使能相应功能时候有用，如果RAM资源有限，任务堆栈的大小需要根据任务来评估下；

1. Free Modbus任务实现

基于Free Modbus文件，在之前的基础上，移植到u/COS\_II操作系统上，能够进行正常的通信，进行正常的数据读写；实际在应用过程中，当有多个任务时候，需要将Modbus通信设为优先级最高，同时主要该任务执行完之后的延时函数或者说是等待时间，该时间的长度会影响到串口通信质量，延时时间越长，通信误码率越高；



1. 总结

该例程主要是实现u/COS\_II操作系统的代码在STM32开发板上的移植，重点实现了两个任务，一个是Modbus通信，另一个则是LED灯闪烁；能够实现基本的操作功能；