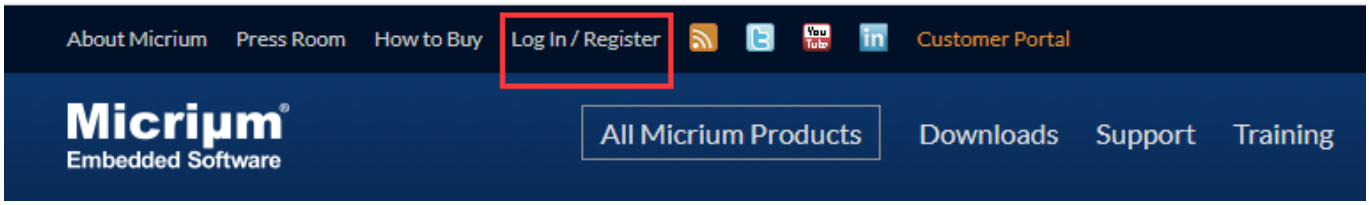
UCOS II 移植到 STM32F407

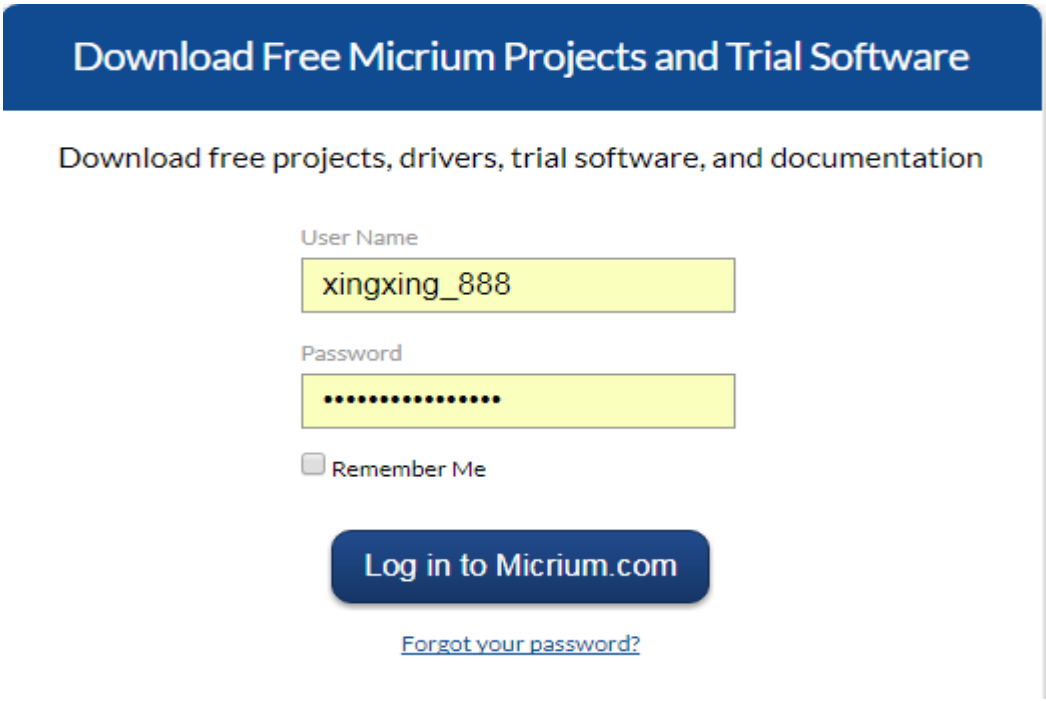
一。UCOS II 源码的获取

https://www.silabs.com/developers/micrium

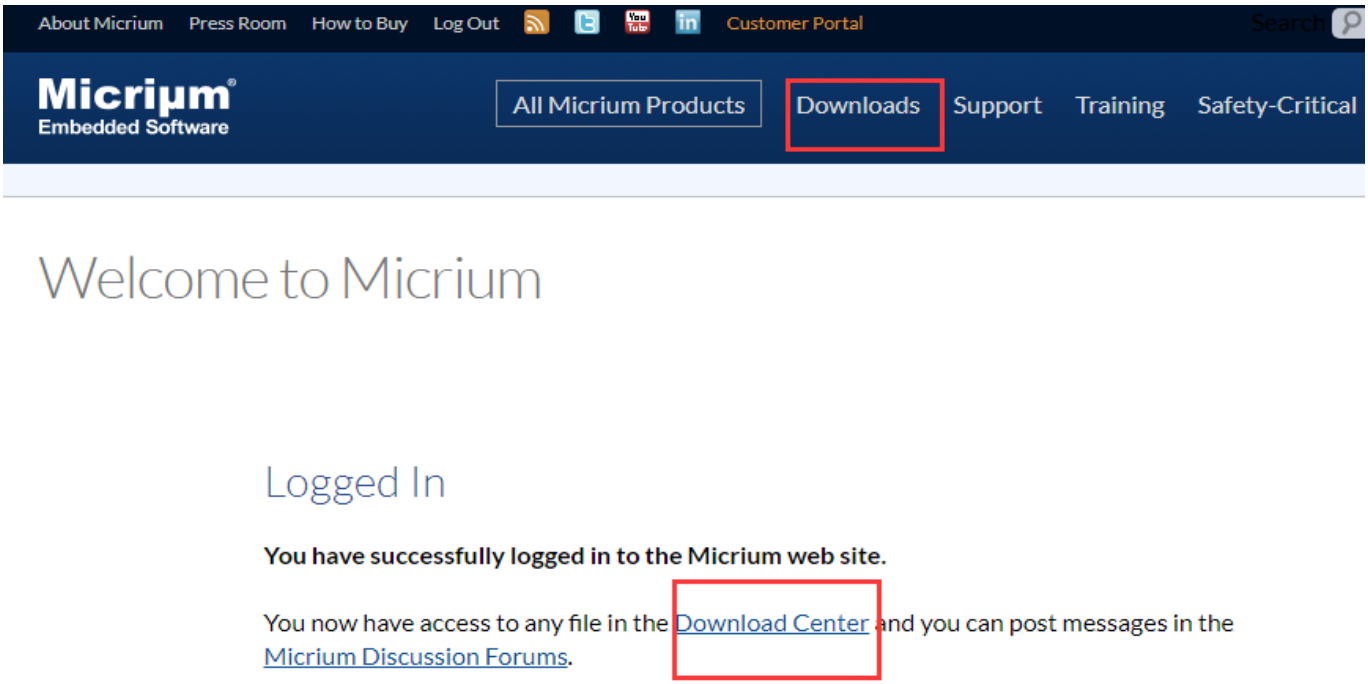
1. 登陆 µC/OS-官方网站 http://micrium.com/，如下所示：



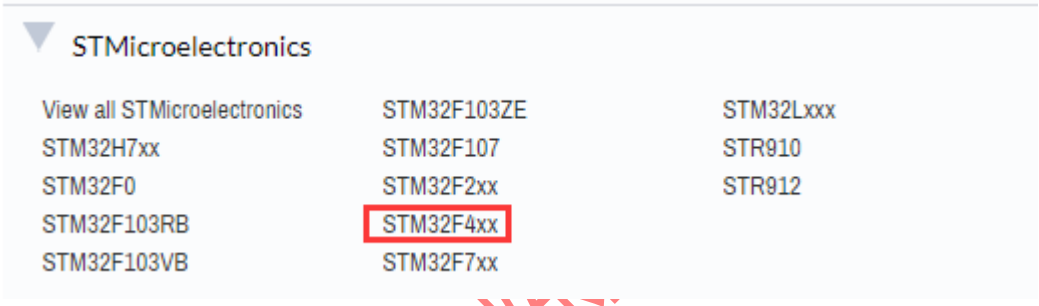
2. 点击网站上方的 Login/Registration ，进行登陆或注册一个新帐号。如果没有帐号要先注册一个，注册过程不再描述。我已经有帐号，直接登陆。
3. 点击后转入登陆界面，网址是 http://micrium.com/wp-login.php ， 直接输入这个进行登陆也可以，界面如下：



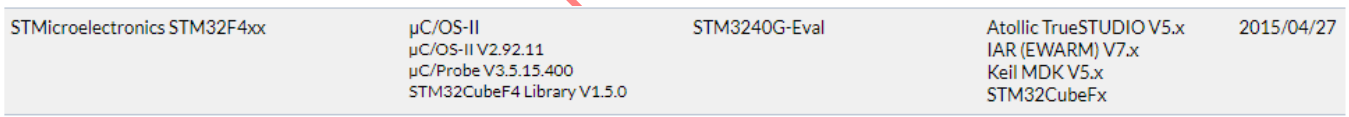
4. 在这个界面中输入用户名和密码，点击 Log In,即可以进入，登陆后可以下载网站上的免费资源。



5. 点击上面的“Downloads Center ”，进入下载页面。

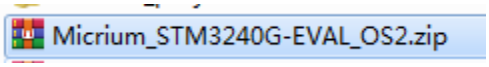


6. 点击下载，然后选择 ucos ii 版本下载

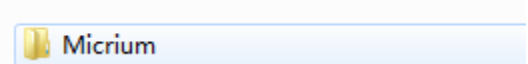


7. 下载得到一个压缩包，并解压。

解压前：

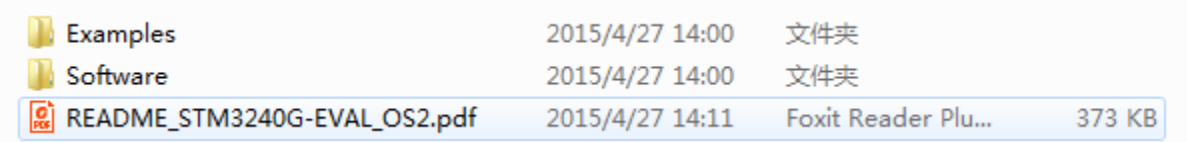


解压后：



二。UCOS II 源码分析

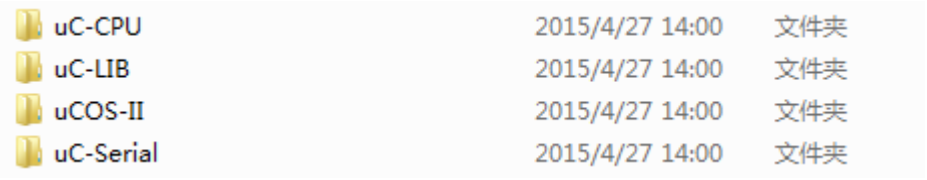
1. 打开 Micrium 文件



Examples：官方提供的例程

Software：官方提供的源码

2. 打开 Software



µC-CPU：这是和 CPU 紧密相关的文件，我们不需要使用。

µC-LIB：Micrium 公司提供的官方库文件，如字符串操作、内存操作等函数接口，可用可不用。

µCOS-II：这是关键目录文件，我们接下来要详细分析的文件，跟移植、使用密切相关的。

µC-Serial：µC/OS-II 接口文件文件。

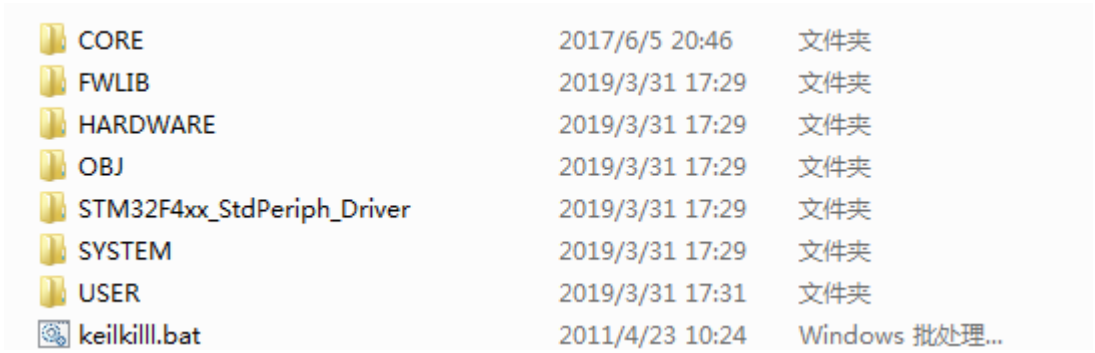
3. 打开 uCOS-LIB

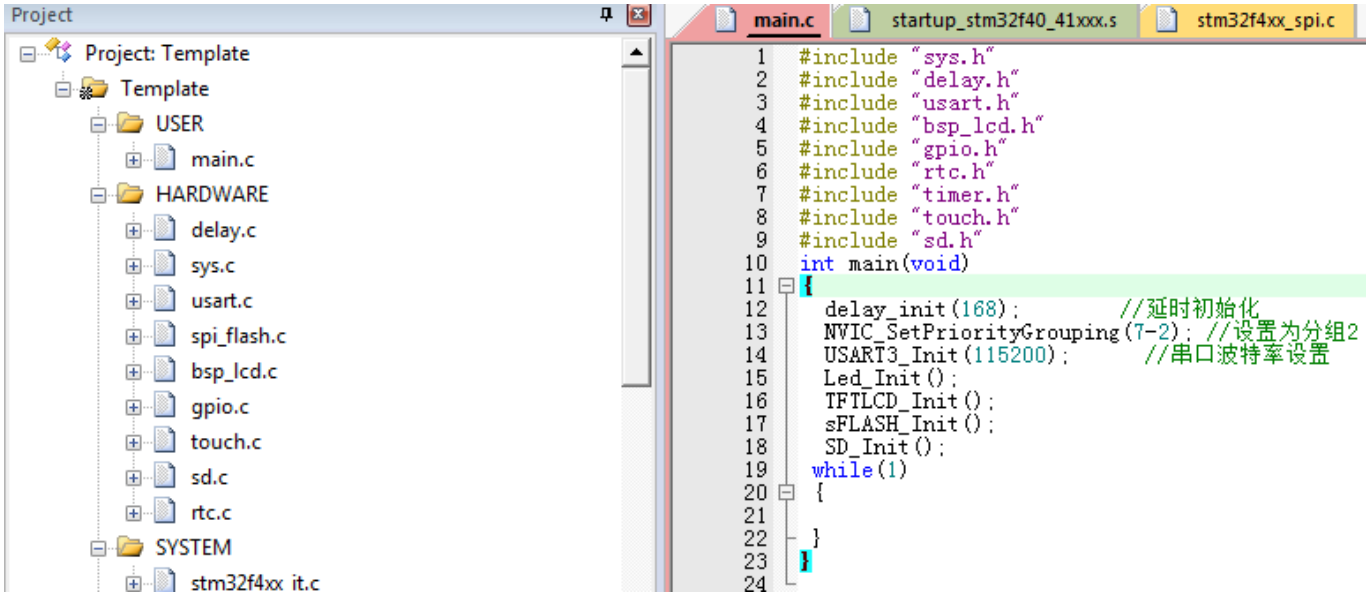
4. 打开 uCOS-II/ Source

三。UCOS II 移植

3.1 裸机工程的准备

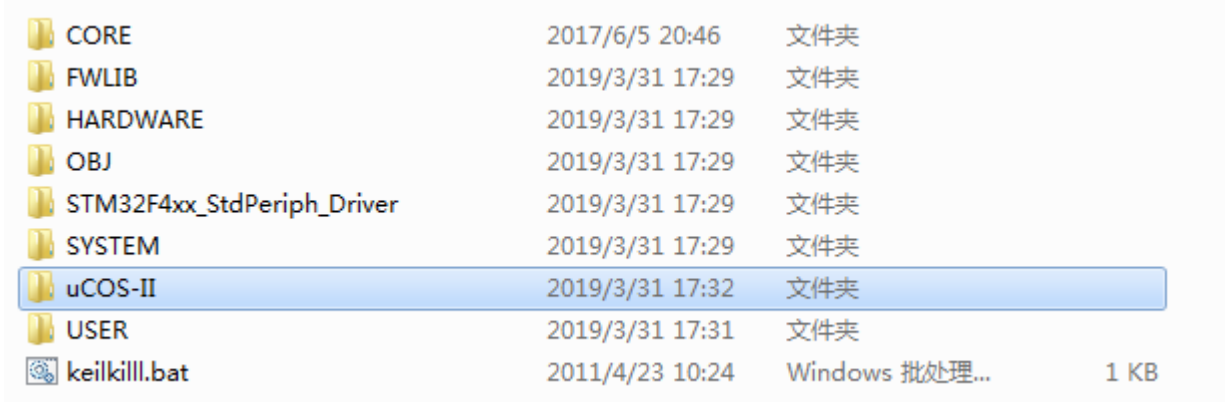
找一个之前的裸机工程，最好带串口功能，方便调试。





3.2 复制相关 µC/OS- II 的移植文件

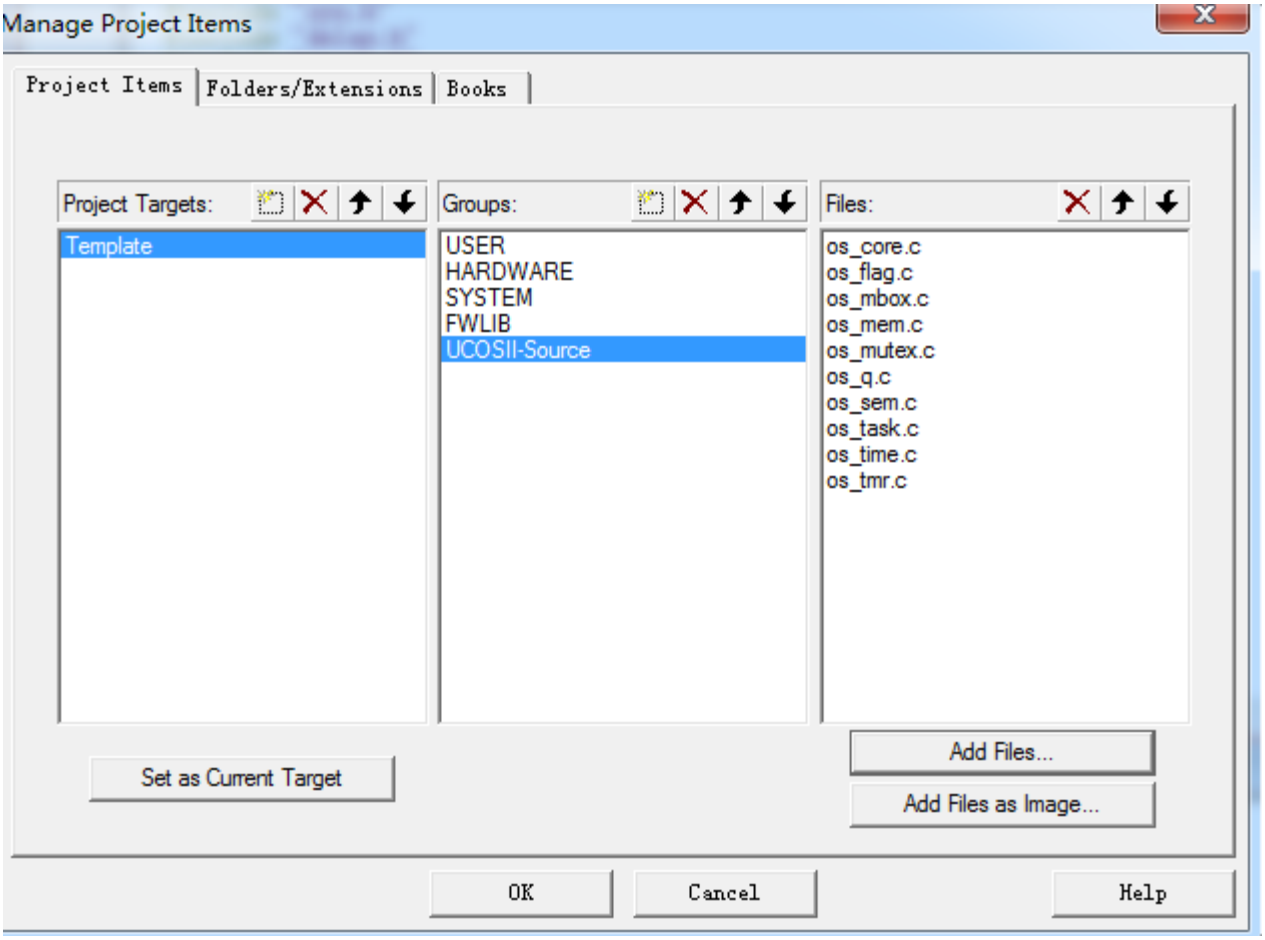
1. 复制 ucos –II 到工程中



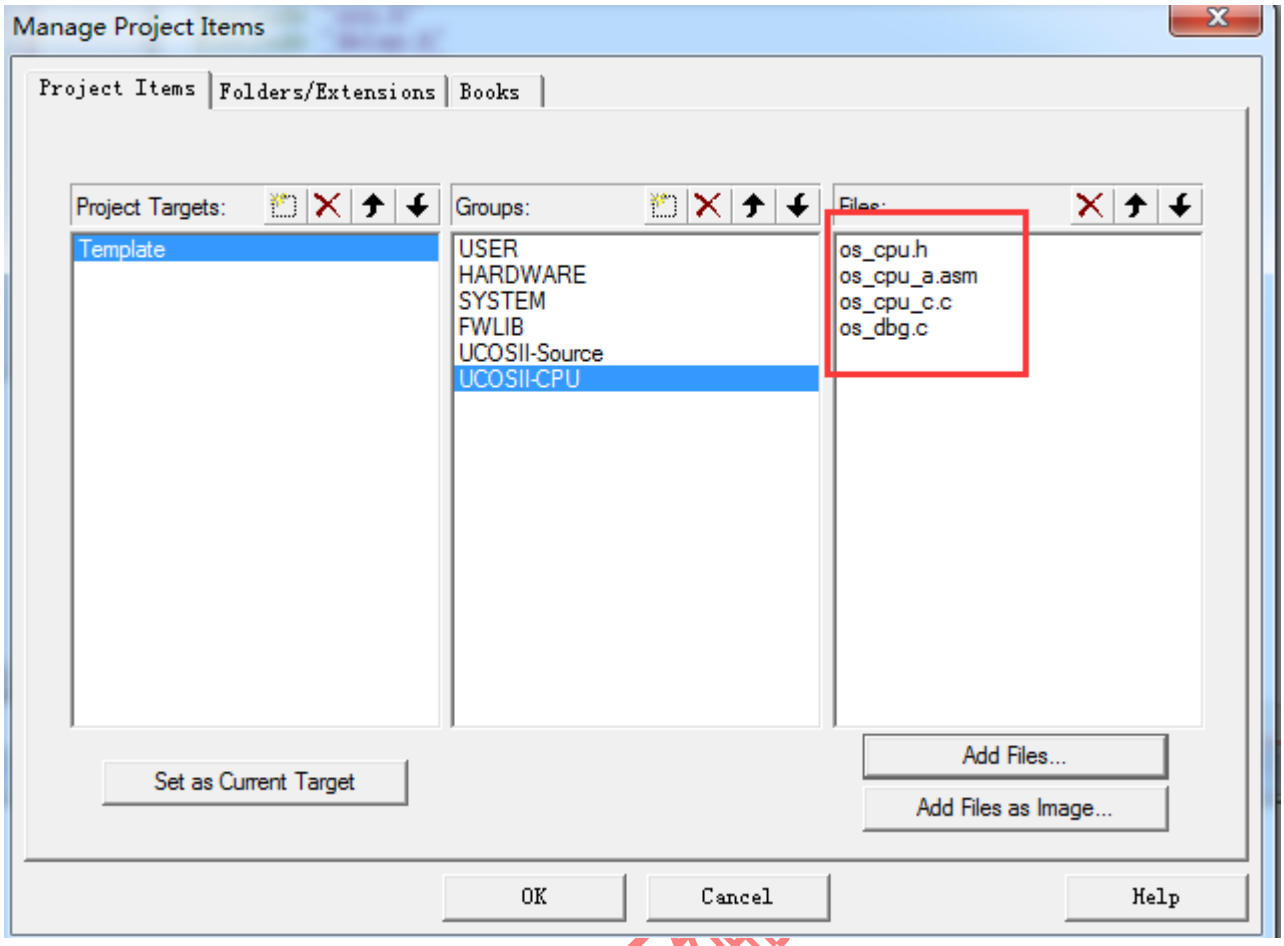
2. 复制\Micrium\Examples\ST\STM3240G-EVAL\OS2\ os_cfg.h 到 uCOS-II\Source 中

os.h	2014/4/17 16:25	C++ Header file	2 KB
os_cfg.h	2015/4/21 15:05	C++ Header file	11 KB
os_core.c	2014/12/11 16:59	C Source file	88 KB
os_flag.c	2014/4/17 16:25	C Source file	56 KB
os_mbox.c	2014/4/17 16:25	C Source file	31 KB
os_mem.c	2014/4/17 16:25	C Source file	20 KB
os_mutex.c	2014/4/17 16:25	C Source file	38 KB
os_q.c	2014/4/17 16:25	C Source file	42 KB
os_sem.c	2014/4/17 16:25	C Source file	29 KB
os_task.c	2014/4/17 16:25	C Source file	60 KB
os_time.c	2014/4/17 16:25	C Source file	11 KB
os_tmr.c	2014/4/17 16:25	C Source file	44 KB
ucos_ii.h	2019/3/31 17:58	C++ Header file	81 KB

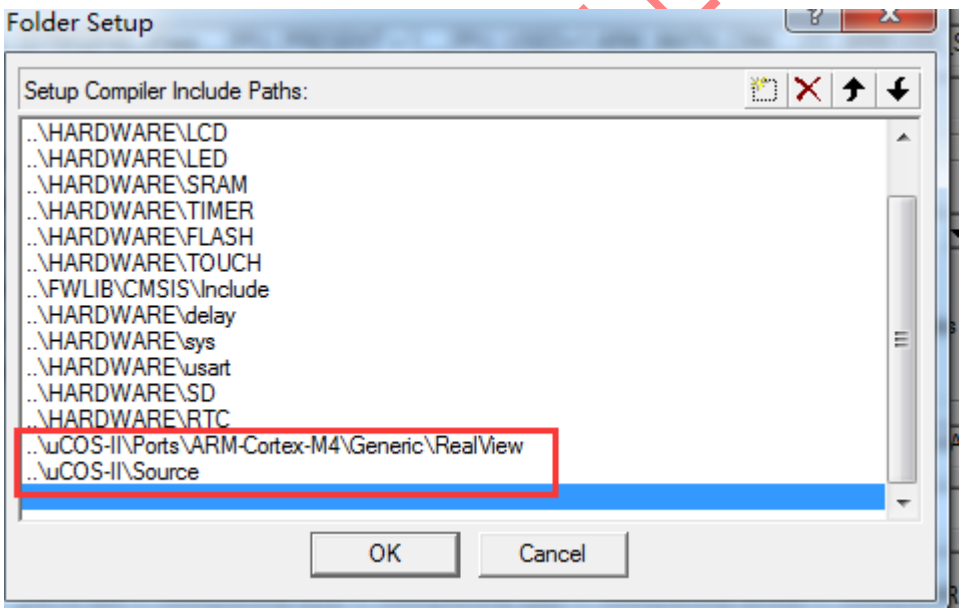
3. 添加 ucos-ii 源码到工程中



4. 添加 uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView 中的 CPU 相关文件到工程中



5. 添加头文件路径



3.3 编译纠错

1. 编译

```
..\uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView\os_cpu_c.c: 0 warnings, 1 error
compiling os_dbg.c...
..\uCOS-II\Source\ucos_ii.h(44): error:  #5: cannot open source input file "app_cfg.h": No such file or directory
  #include <app_cfg.h>
..\uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView\os_dbg.c: 0 warnings, 1 error
"..\OBJ\Template.axf" - 12 Error(s), 0 Warning(s).
Target not created.
Build Time Elapsed:  00:01:10
```

找不到此头文件，此头文件在\Micrium\Examples\ST\STM3240G-EVAL\OS2\目录下，我们并没有使用到此头文件的功能，所以把包含此头文件屏蔽即可。

main.c* ucos_ii.h*

```
43
44  //#include <app_cfg.h>
45  #include <os_cfg.h>
46  #include <os_cpu.h>
47
```

2. 编译

```
compiling os_cpu_c.c...
..\uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView\os_cpu_c.c(53): error:  #5: cannot open source input file "lib_def.h": No such file or directory
  #include  <lib_def.h>
..\uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView\os_cpu_c.c: 0 warnings, 1 error
compiling os_dbg.c...
"..\OBJ\Template.axf" - 1 Error(s), 0 Warning(s).
Target not created.
Build Time Elapsed:  00:00:10
```

此关文件在\Micrium\Software\uC-LIB 目录下，不需要它，再次屏蔽。

main.c ucos_ii.h os_cpu_c.c*

```
50  */
51
52  #include  <ucos_ii.h>
53  //#include  <lib_def.h>
54
```

3. 编译

Build Output

```
..\uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView\os_cpu_c.c(347): error:  #65: expected a ";"
        *--p_stk = (CPU_STK)0x40A00000u;                        /* S5                                       */
..\uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView\os_cpu_c.c(348): error:  #65: expected a ";"
        *--p_stk = (CPU_STK)0x40800000u;                        /* S4                                       */
..\uCOS-II\Ports\ARM-Cortex-M4\Generic\RealView\os_cpu_c.c: 0 warnings, 30 errors
"..\OBJ\Template.axf" - 30 Error(s), 0 Warning(s).
Target not created.
Build Time Elapsed:  00:00:02
```

main.c ucos_ii.h os_cpu_c.c

```
317 #if (OS_CPU_ARM_FP_EN == DEF_ENABLED)
318     if ((opt & OS_TASK_OPT_SAVE_FP) != (INT16U)0) {
319         *--p_stk = (CPU_STK)0x02000000u;                    /* FPSCR
320                                                             /* Initiali
321         *--p_stk = (CPU_STK)0x41F80000u;                    /* S31
322         *--p_stk = (CPU_STK)0x41F00000u;                    /* S30
323         *--p_stk = (CPU_STK)0x41E80000u;                    /* S29
324         *--p_stk = (CPU_STK)0x41E00000u;                    /* S28
```

把#if 部分内容屏蔽：

main.c | ucos_ii.h | os_cpu_c.c*

```
514     *(--p_stk) = (OS_STK)0x06060606uL;                  /* R6                                                   */
515     *(--p_stk) = (OS_STK)0x04040404uL;                  /* R4                                                   */
516
517     //#if (OS_CPU_ARM_FP_EN == DEF_ENABLED)
518     //    if ((opt & OS_TASK_OPT_SAVE_FP) != (INT16U)0) {
519     //        *--p_stk = (CPU_STK)0x02000000u;                /* FPSCR                                        */
520     //                                                        /* Initialize S0-S31 floating point registers   */
521     //        *--p_stk = (CPU_STK)0x41F80000u;                /* S31                                          */
522     //        *--p_stk = (CPU_STK)0x41F00000u;                /* S30                                          */
523     //        *--p_stk = (CPU_STK)0x41E80000u;                /* S29                                          */
524     //        *--p_stk = (CPU_STK)0x41E00000u;                /* S28                                          */
525     //        *--p_stk = (CPU_STK)0x41D80000u;                /* S27                                          */
526     //        *--p_stk = (CPU_STK)0x41D00000u;                /* S26                                          */
527     //        *--p_stk = (CPU_STK)0x41C80000u;                /* S25                                          */
528     //        *--p_stk = (CPU_STK)0x41C00000u;                /* S24                                          */
529     //        *--p_stk = (CPU_STK)0x41B80000u;                /* S23                                          */
530     //        *--p_stk = (CPU_STK)0x41B00000u;                /* S22                                          */
531     //        *--p_stk = (CPU_STK)0x41A80000u;                /* S21                                          */
532     //        *--p_stk = (CPU_STK)0x41A00000u;                /* S20                                          */
533     //        *--p_stk = (CPU_STK)0x41980000u;                /* S19                                          */
534     //        *--p_stk = (CPU_STK)0x41900000u;                /* S18                                          */
535     //        *--p_stk = (CPU_STK)0x41880000u;                /* S17                                          */
536     //        *--p_stk = (CPU_STK)0x41800000u;                /* S16                                          */
537     //        *--p_stk = (CPU_STK)0x41700000u;                /* S15                                          */
538     //        *--p_stk = (CPU_STK)0x41600000u;                /* S14                                          */
539     //        *--p_stk = (CPU_STK)0x41500000u;                /* S13                                          */
540     //        *--p_stk = (CPU_STK)0x41400000u;                /* S12                                          */
541     //        *--p_stk = (CPU_STK)0x41300000u;                /* S11                                          */
542     //        *--p_stk = (CPU_STK)0x41200000u;                /* S10                                          */
543     //        *--p_stk = (CPU_STK)0x41100000u;                /* S9                                           */
544     //        *--p_stk = (CPU_STK)0x41000000u;                /* S8                                           */
545     //        *--p_stk = (CPU_STK)0x40E00000u;                /* S7                                           */
546     //        *--p_stk = (CPU_STK)0x40C00000u;                /* S6                                           */
547     //        *--p_stk = (CPU_STK)0x40A00000u;                /* S5                                           */
548     //        *--p_stk = (CPU_STK)0x40800000u;                /* S4                                           */
549     //        *--p_stk = (CPU_STK)0x40400000u;                /* S3                                           */
550     //        *--p_stk = (CPU_STK)0x40000000u;                /* S2                                           */
551     //        *--p_stk = (CPU_STK)0x3F800000u;                /* S1                                           */
552     //        *--p_stk = (CPU_STK)0x00000000u;                /* S0                                           */
553     //    }
554     //#endif
```

4. 编译

```
linking...
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TCBInitHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TaskCreateHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TaskDelHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TaskIdleHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TaskReturnHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TaskStatHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TaskSwHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol App_TimeTickHook (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol DEF_BIT_FIELD (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol DEF_BIT_MASK (referred from os_cpu_c.o).
Not enough information to list image symbols.
Finished: 1 information, 0 warning and 10 error messages.
"..\OBJ\Template.axf" - 10 Error(s), 2 Warning(s).
Target not created.
Build Time Elapsed:  00:00:04
```

函数没定义.

```
#if OS_APP_HOOKS_EN > 0u
    App_TaskSwHook();
#endif
}
```

把宏定义改为 0，让条件不成立。

main.c | ucos_ii.h | os_cpu_c.c | os_cfg.h*

```
30  #define OS_APP_HOOKS_EN          0u    /* A
```

5. 编译

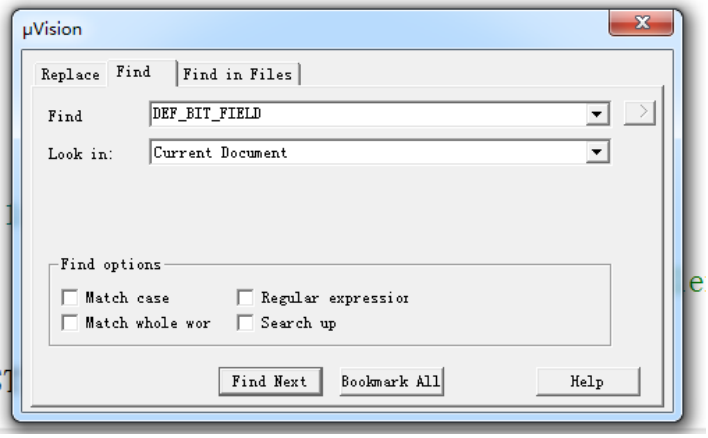
```
..\OBJ\Template.axf: Error: L6218E: Undefined symbol DEF_BIT_FIELD (referred from os_cpu_c.o).
..\OBJ\Template.axf: Error: L6218E: Undefined symbol DEF_BIT_MASK (referred from os_cpu_c.o).
Not enough information to list image symbols.
Finished: 1 information, 0 warning and 2 error messages.
"..\OBJ\Template.axf" - 2 Error(s), 2 Warning(s).
```

没定义，搜索在哪。

```
void  OS_CPU_SysTickInit (INT32U  cnts)
{
    INT32U  prio;

    OS_CPU_CM4_NVIC_ST_RELOAD = cnts -

    prio  = OS_CPU_CM4_NVIC_SHPRI3;
    prio &= DEF_BIT_FIELD(24, 0);
    prio |= DEF_BIT_MASK(OS_CPU_CFG_SYST
```



此函数为系统滴答初始化，把内容屏蔽，换成我们之前写的。

core_cm4.h | main.c | ucos_ii.h | os_cpu_c.c | os_cfg.h

```
1676  */
1677  __STATIC_INLINE uint32_t SysTick_Config(uint32_t ticks)
1678  {
1679    if ((ticks - 1) > SysTick_LOAD_RELOAD_Msk)  return (1);      /* Reload value impossible */
1680
1681    SysTick->LOAD  = ticks - 1;                                  /* set reload register */
1682    NVIC_SetPriority (SysTick_IRQn, (1<<__NVIC_PRIO_BITS) - 1);  /* set Priority for Systick Interrupt
1683    SysTick->VAL   = 0;                                          /* Load the SysTick Counter Value */
1684    SysTick->CTRL  = SysTick_CTRL_CLKSOURCE_Msk |
1685                     SysTick_CTRL_TICKINT_Msk   |
1686                     SysTick_CTRL_ENABLE_Msk;                    /* Enable SysTick IRQ and SysTick Time
1687    return (0);                                                  /* Function successful */
1688  }
```

```
void  OS_CPU_SysTickInit (INT32U  cnts)
{
//    INT32U  prio;
//    OS_CPU_CM4_NVIC_ST_RELOAD = cnts - 1u;
//    prio  = OS_CPU_CM4_NVIC_SHPRI3;
//    prio &= DEF_BIT_FIELD(24, 0);
//    prio |= DEF_BIT_MASK(OS_CPU_CFG_SYSTICK_PRIO, 24);
//    OS_CPU_CM4_NVIC_SHPRI3 = prio;
//    OS_CPU_CM4_NVIC_ST_CTRL |= OS_CPU_CM4_NVIC_ST_CTRL_CLK_SRC |
//                               OS_CPU_CM4_NVIC_ST_CTRL_ENABLE;
//    OS_CPU_CM4_NVIC_ST_CTRL |= OS_CPU_CM4_NVIC_ST_CTRL_INTEN;
}
```

改为：

```
#include "stm32f4xx.h"
void  OS_CPU_SysTickInit (INT32U  cnts)
{
  if ((cnts - 1) > SysTick_LOAD_RELOAD_Msk)  return ;       /
  SysTick->LOAD  = cnts - 1;
  NVIC_SetPriority (SysTick_IRQn, (1<<__NVIC_PRIO_BITS) - 1)
  SysTick->VAL   = 0;
  SysTick->CTRL  = SysTick_CTRL_CLKSOURCE_Msk
                   SysTick_CTRL_TICKINT_Msk
                   SysTick_CTRL_ENABLE_Msk;
  return ;
}
```

6. 编译

```
compiling os_cpu_c.c...
linking...
Program Size: Code=32668 RO-data=6532 RW-data=64 ZI-data=12600
FromELF: creating hex file...
"..\OBJ\Template.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:04
```

3.4 修改文件

1. 修改中断服务函数

core_cm4.h | main.c | ucos_ii.h | os_cpu_c.c* | os_cfg.h

```
460
461  void  OS_CPU_SysTickHandler (void)
462  {
463      OS_CPU_SR  cpu_sr;
464      OS_ENTER_CRITICAL();
465      OSIntNesting++;
466      OS_EXIT_CRITICAL();
467      OSTimeTick();
468      OSIntExit();
469  }
```

main.c | os_cpu_a.asm

```
276  ;*********************************************************
277
278  OS_CPU_PendSVHandler
279      CPSID   I
280      MRS     R0, PSP
281      CBZ     R0, OS_CPU_PendSVHandler_nosave
282
```

可以看出此中断服务函数跟之前学习的不一样，所以需要修改。

修改启动文件的中断服务函数名。

```
startup_stm32f40_41xxx.s*  main.c  os_cpu_c.c
82  DCD     0                           ; Reserved
83  ;DCD    PendSV_Handler              ; PendSV Handler
84  ;DCD    SysTick_Handler             ; SysTick Handler
85  ;修改为
86  DCD     OS_CPU_PendSVHandler            ; PendSV Handler modifi by chenxingxing
87  DCD     OS_CPU_SysTickHandler           ; SysTick Handler modifi by chenxingxing
88
```

加上声明

```
startup_stm32f40_41xxx.s*  main.c  os_cpu_a.asm
65  EXPORT  __Vectors
66  EXPORT  __Vectors_End
67  EXPORT  __Vectors_Size
68
69  IMPORT OS_CPU_SysTickHandler        ;by chenxingxing
70  IMPORT OS_CPU_PendSVHandler         ;by chenxingxing
71
```

2. 修改 main.c

1）添加 ucos 头文件

#include "ucos_ii.h"

2） 添加系统滴答初始化

OS_CPU_SysTickInit(168000000 / OS_TICKS_PER_SEC);

3）添加 ucos 初始化

OSInit();

4）创建 ucos 任务


```
#include "stm32f4xx.h"
#include "led.h"
#include "key.h"
#include "usart.h"
#include "string.h"
#include "ucos_ii.h"

#define OS_TASK_START_PRIO 10
#define OS_TASK_START_STK_SIZE 128
OS_STK OSTaskStartStk[OS_TASK_START_STK_SIZE];
void    OS_TaskStart (void *p_arg);
int main ()
{

    LED_Init(); //LED 初始化
    KEY_Init(); //按键初始化
    usart1_init(9600);//串口初始化，波特率为 9600
    OS_CPU_SysTickInit(168000000 / OS_TICKS_PER_SEC);//系统滴答初始化
    OSInit();     //OS 初始化
     //创建任务
```

```
 OSTaskCreate(OS_TaskStart,
                                      (void *)0,
                                       &OSTaskStartStk[OS_TASK_START_STK_SIZE - 1u],
                                       OS_TASK_START_PRIO);
    OSStart();//开启 os,开始调度,跳到已经就绪优先级最高的任务中
}
void   OS_TaskStart (void *p_arg)
{
    p_arg = p_arg; /* Prevent compiler warning for not using 'p_arg'         */
    for (;;) {
            printf("start task\r\n");
        //  OSTimeDly(100);//释放 CPU 使用权，释放了多长时间？ 10* 心跳节拍时间。
            OSTimeDlyHMSM(0,0,1,0);//释放 CPU1 秒时间
    }
}
```