



















uC/OS-II移植过程记录

准备资源包




uC-OS2 > Cfg > Template

名称
 app_cfg.h
 app_hooks.c
 os_cfg.h

uC-OS2 > Source

名称
 os_core.c
 os_dbg_r.c
 os_flag.c
 os_mbox.c
 os_mem.c
 os_mutex.c
 os_q.c
 os_sem.c
 os_task.c
 os_time.c
 os_tmr.c
 ucos_ii.c
 os.h
 os_trace.h
 ucos_ii.h

uC-OS2 > Ports > ARM-Cortex-M > ARMv6-M > IAR

名称	修改日期
 os_cpu.h	2023/1/6 14:10
 os_cpu_a.asm	2023/1/6 14:10
 os_dbg.c	2023/1/6 14:10

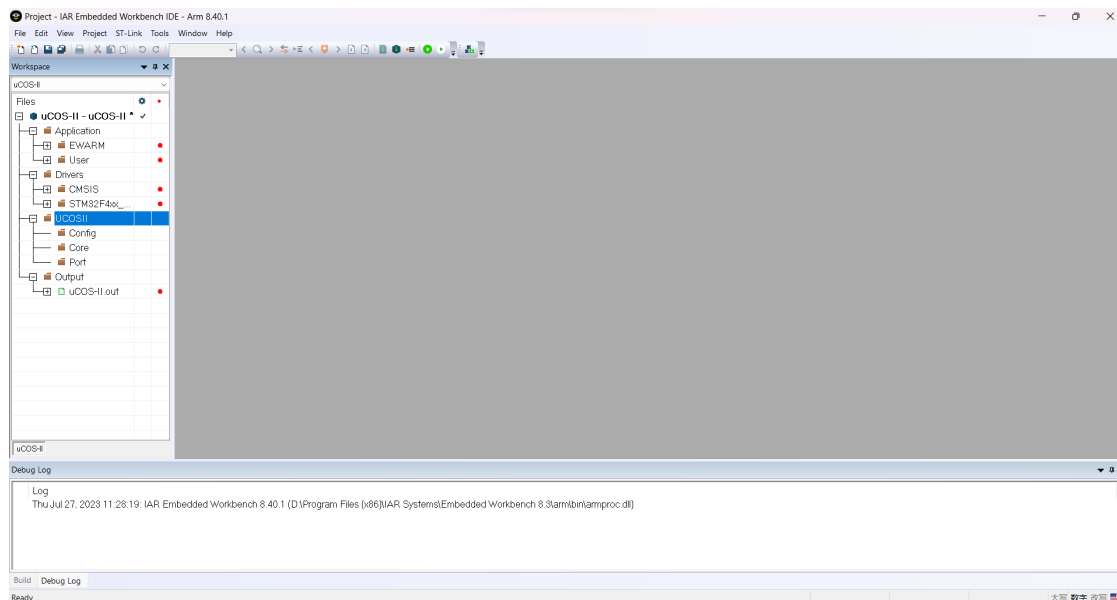
移植流程

1. 建立IAR工程

可以使用标准库搭建或者CubeMX生成

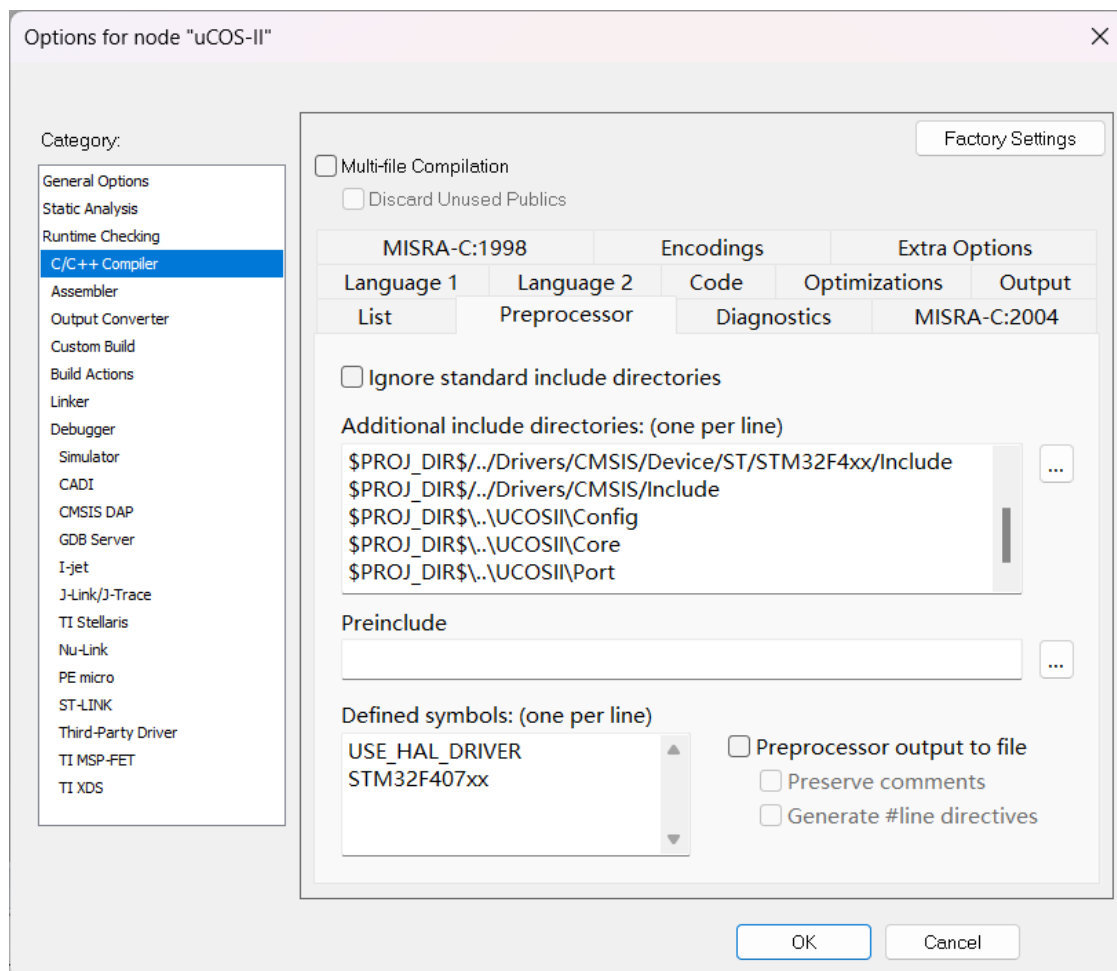
2. 在工程文件夹下建立文件夹UCOSII，并建立子文件夹Core、Config、Port分别存放资源包中Source、Template、IAR文件夹下的内容

3. 增加分组UCOSII，并增加子分组Core、Config、Port

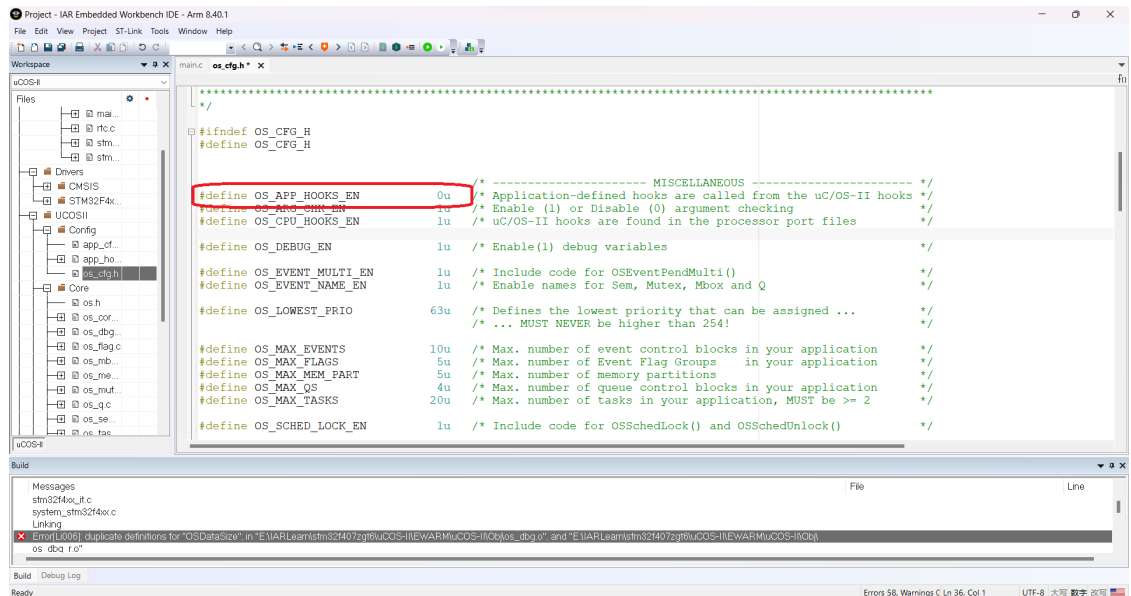


在子分组中添加对应文件夹下的文件，这些是与处理器无关的操作系统内核代码。需要注意ucos_ii.c和os_dbg_r.c不添加，否则会报重复定义错误。

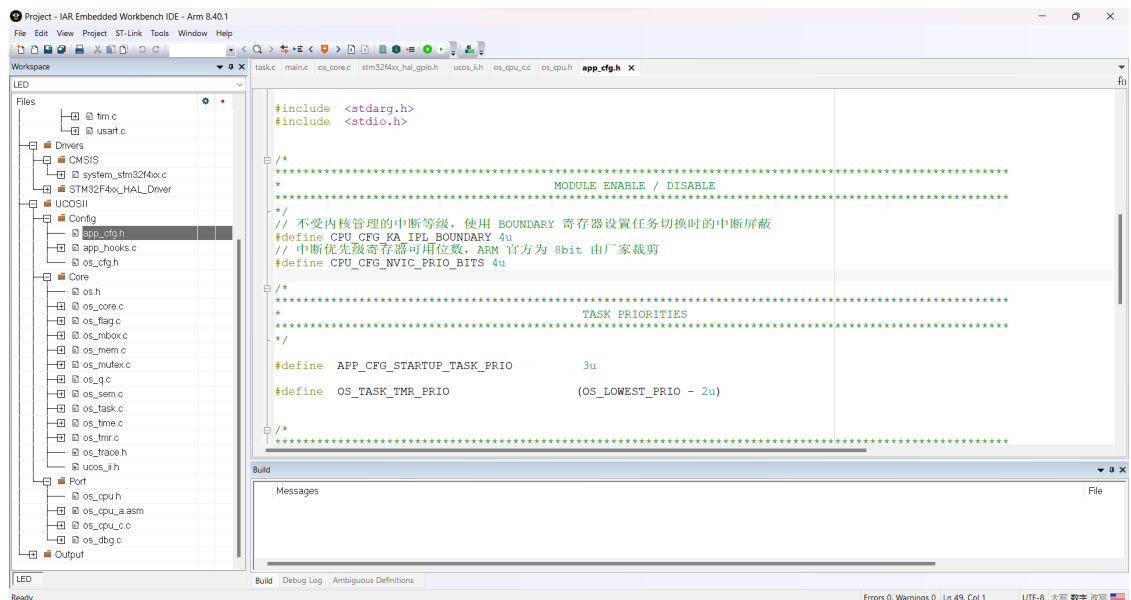
4. 添加路径



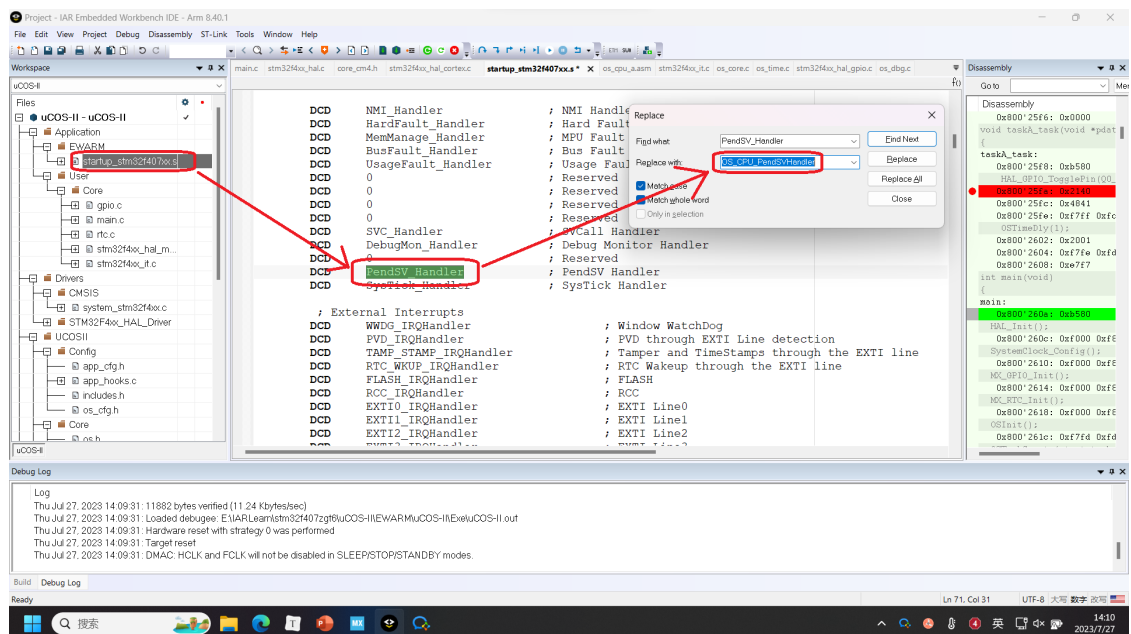
5. 取消APP钩子函数的使用



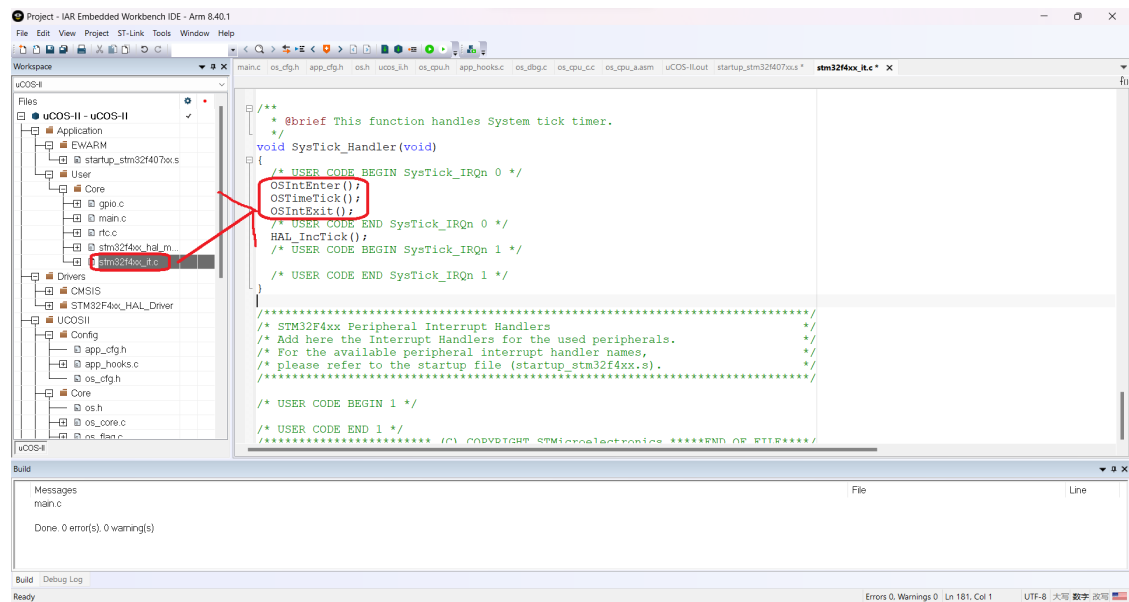
6. 增加两个宏定义



7. 处理系统中断函数



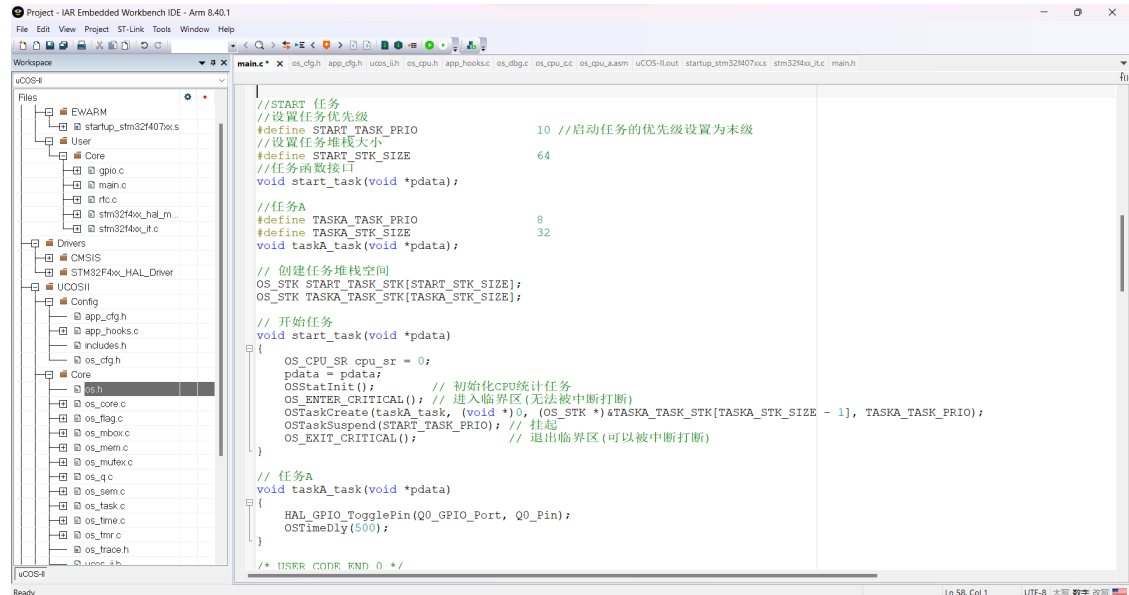
8. 处理系统滴答，增加系统滴答处理函数



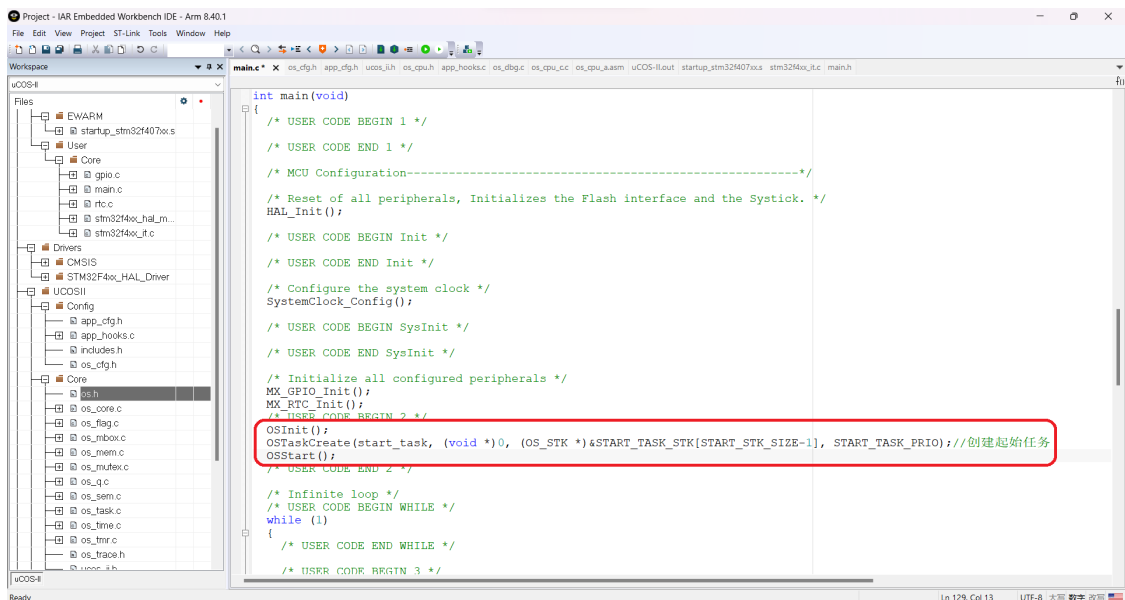
9. 在main.h中包含os.h文件即可包含系统相关的所有头文件

10. 编译通过后就可以开始配置任务了

11. 首先定义任务优先级、堆栈、接口函数等任务基础配置信息，创建任务堆栈空间。之后就可以建立启动任务，在启动任务函数中建立任务A，在任务A函数中书写一个LED翻转功能



12. 在main函数中，在各初始化函数运行完后，创建启动任务并开启任务调度



13、注释PendSV Handler

```
os_cfg.h app_cfg.h startup_stm32f407xx.s stm32f4xx_it.c main.c stm32f4xx_hal_tim.c os_time.c os_core.c os_cpu_a.asm os_cpu_c.c stm32f4xx_hal_rcc.c stm32f4xx_hal.c stm32f4xx_hal_cortex.c core_cm4.h gpio.c os_task.c f()

156  */
157  void DebugMon_Handler(void)
158  {
159      /* USER CODE BEGIN DebugMonitor_IRQn 0 */
160
161      /* USER CODE END DebugMonitor_IRQn 0 */
162      /* USER CODE BEGIN DebugMonitor_IRQn 1 */
163
164      /* USER CODE END DebugMonitor_IRQn 1 */
165  }
166
167  /**
168   * @brief This function handles Pendable request for system service.
169   */
170  void PendSV_Handler(void)
171  {
172      /* USER CODE BEGIN PendSV_IRQn 0 */
173
174      /* USER CODE END PendSV_IRQn 0 */
175      /* USER CODE BEGIN PendSV_IRQn 1 */
176
177      /* USER CODE END PendSV_IRQn 1 */
178  }
179
180  /**
181   * @brief This function handles System tick timer.
182   */
183  void SysTick_Handler(void)
184  {
185      /* USER CODE BEGIN SysTick_IRQn 0 */
186
187      /* USER CODE END SysTick_IRQn 0 */
188  }
```