

## uCOS 移植手册

- (1) 硬件平台:宝马 LPC1768 开发板  
(2) 软件资料 : LED 灯例程+uCOSII 源码
- 到 Mirciurm 官网下载 LPC1768 系列的工程模板,我们会使用到这个工程中的文件。  
官网地址: <https://www.micrium.com/>。(使用前需要注册,才可以下载文件)



### Projects

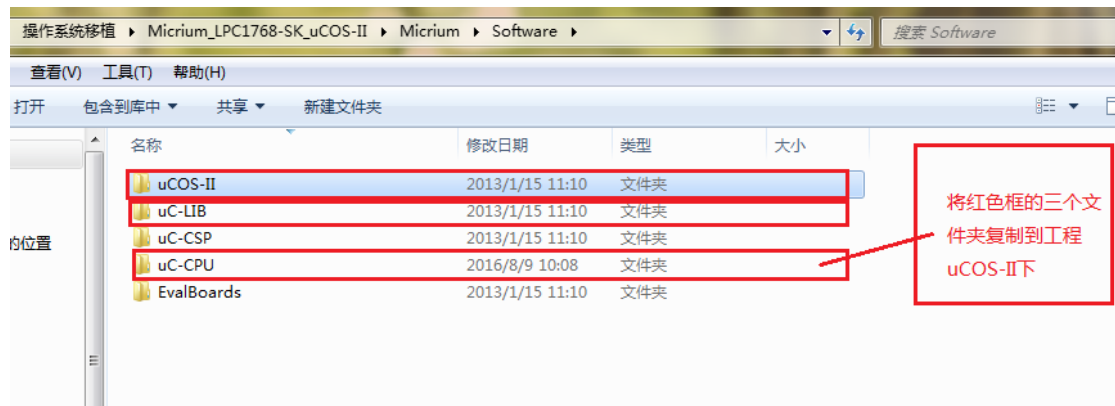
MCU	Micrium Product	Evaluation Board	Toolchain	Date
NXP LPC1768	$\mu$ C/OS-III $\mu$ C/OS-III V3.03.01	IAR LPC1768-SK	Atollic TrueSTUDIO V3.x IAR (EWARM) V6.x Keil MDK V4.x	2012/11/30
NXP LPC1768	$\mu$ C/OS-II $\mu$ C/OS-II V2.92.07	IAR LPC1768-SK	Atollic TrueSTUDIO V3.x IAR (EWARM) V6.x Keil MDK V4.x	2012/11/30

下载uC/OS-II工程文件

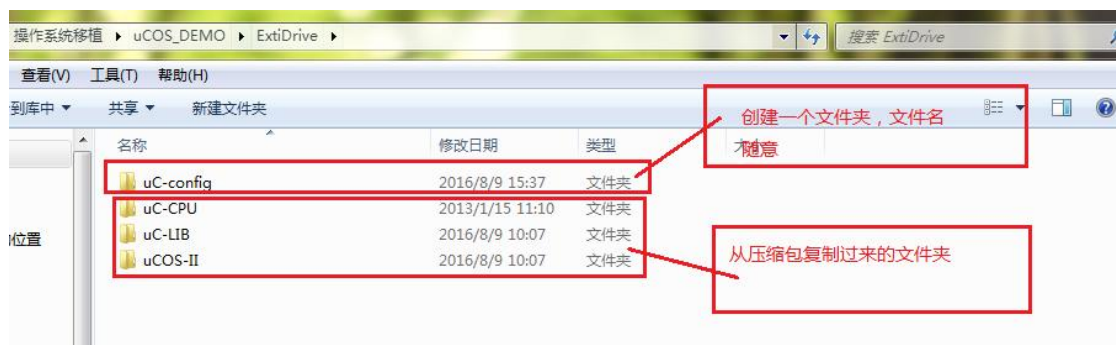
### Micrium Book Projects

MCU	Micrium Product	Evaluation Board	Toolchain	Date
NXP LPC1758 LPC1768	$\mu$ C/OS-III $\mu$ C/OS-III V3.02.00 Projects and examples for the book $\mu$ C/OS-III: The Real-Time Kernel for the Keil MCB1700	Keil MCB1700	Keil MDK V4.x	2012/12/05

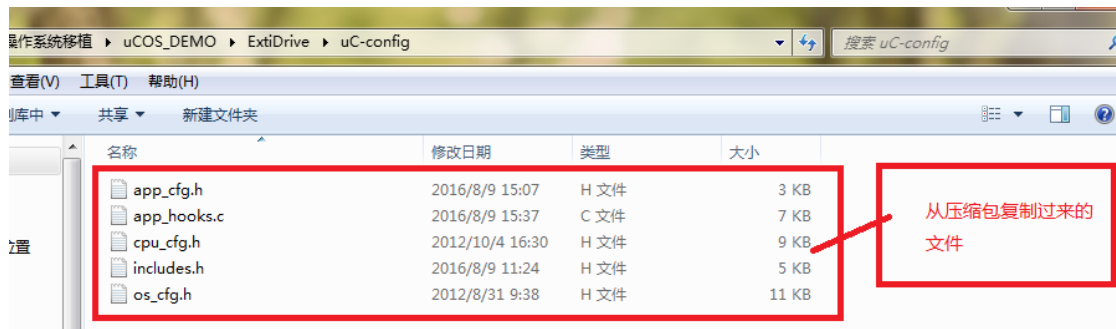
- 解压下载的压缩包,将压缩包里面的的 uC-CPU、uC-LIB 和 UCOS-II 这三个文件夹复制到 uCOS-II 文件夹中。如下图所示



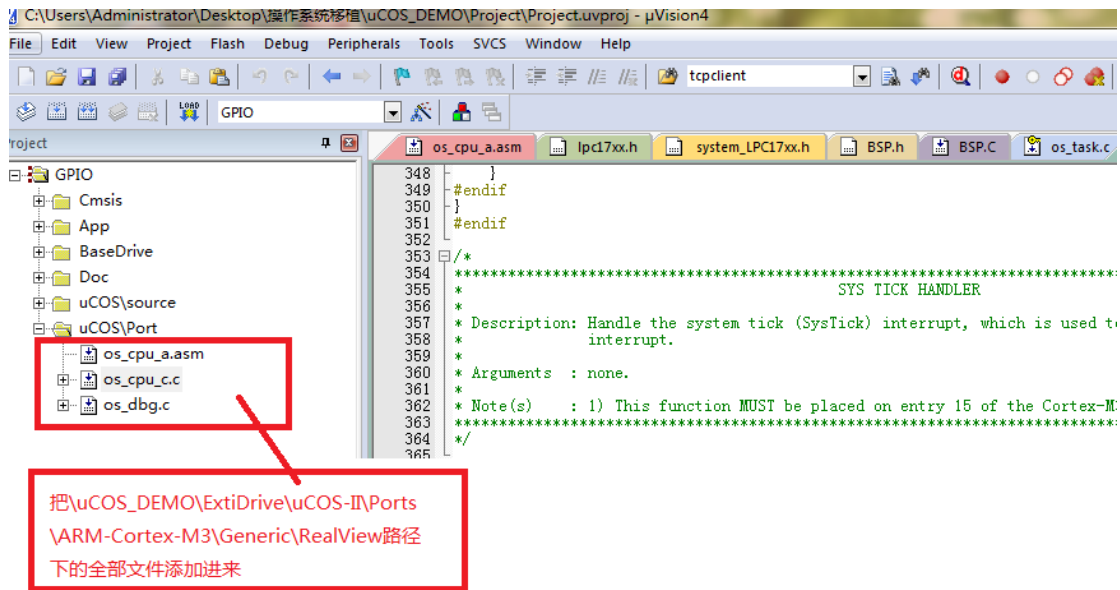
4. 在 uCOS\_DEMO 下的 ExtiDrive 下创建一个文件名为 uC-config 的文件。如下图所示



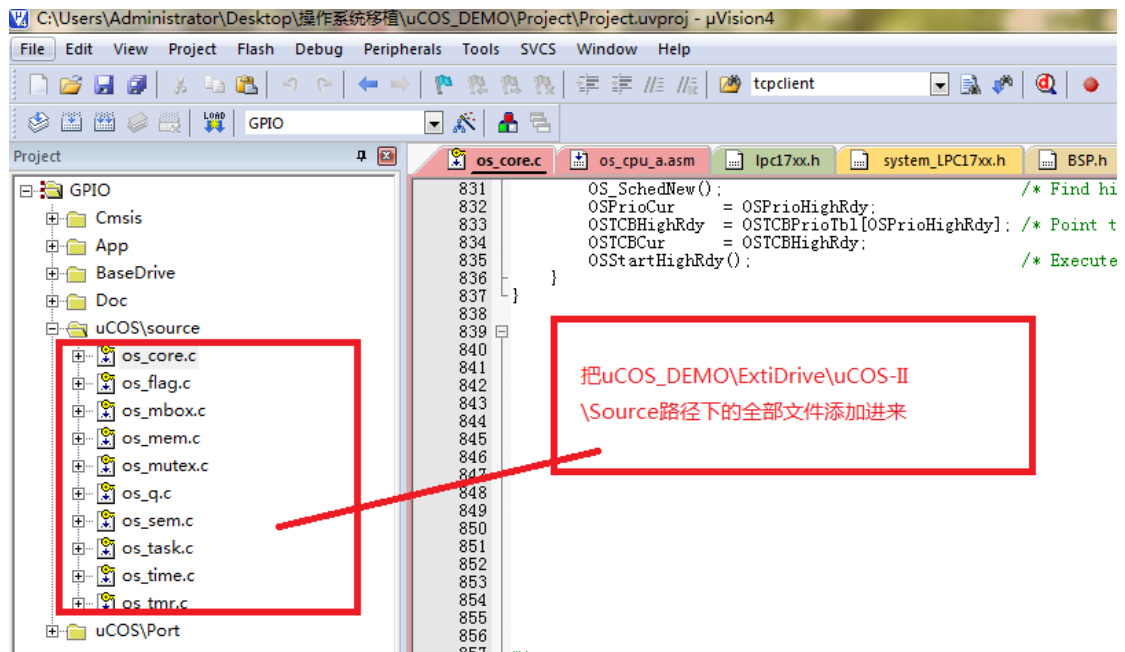
5. 在 uC-config 文件夹添加以下文件 app\_cfg.h , app\_hooks.c , cpu\_cfg.h , includes.h , os\_cfg.h , 如下图所示



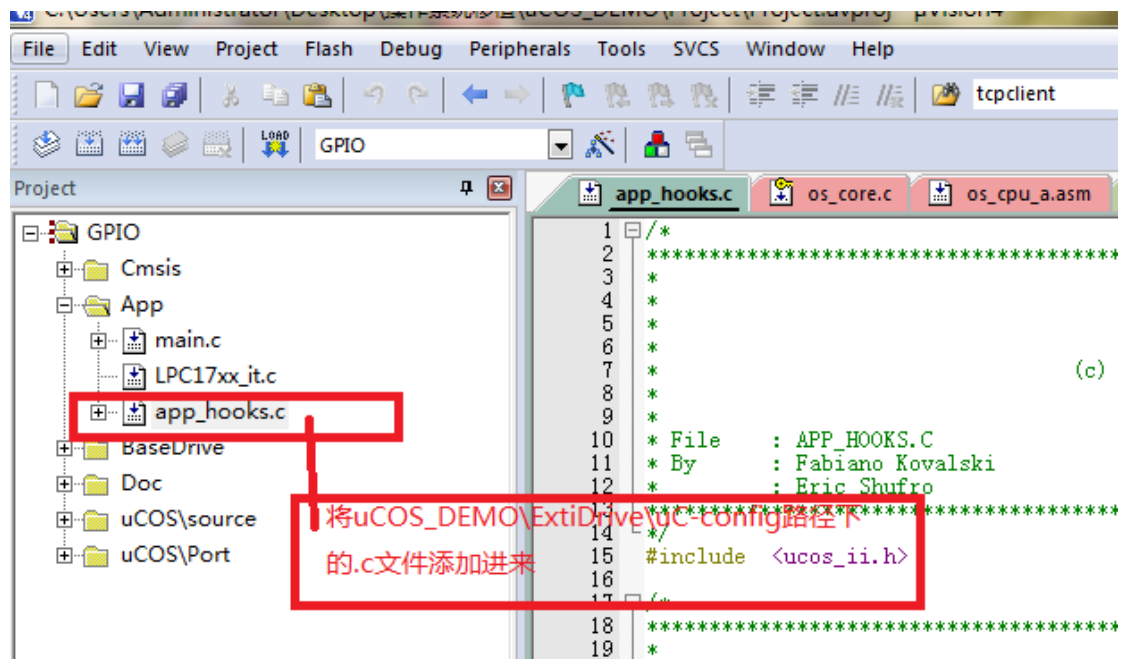
6. 在工程目录 uCOS\Port 中添加以下文件, 如下图所示



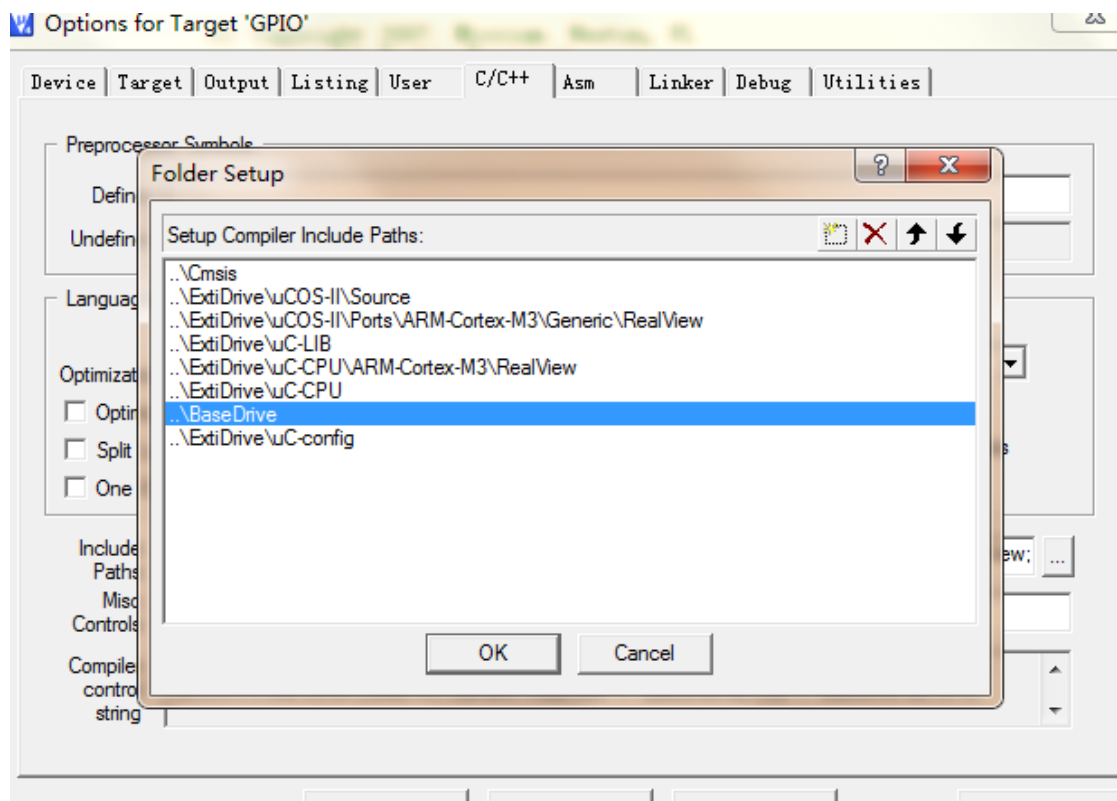
7. 在工程目录 uCOS\source 中添加以下文件，如下图所示



8. 在工程目录 uCOS\app 中添加以下文件，如下图所示

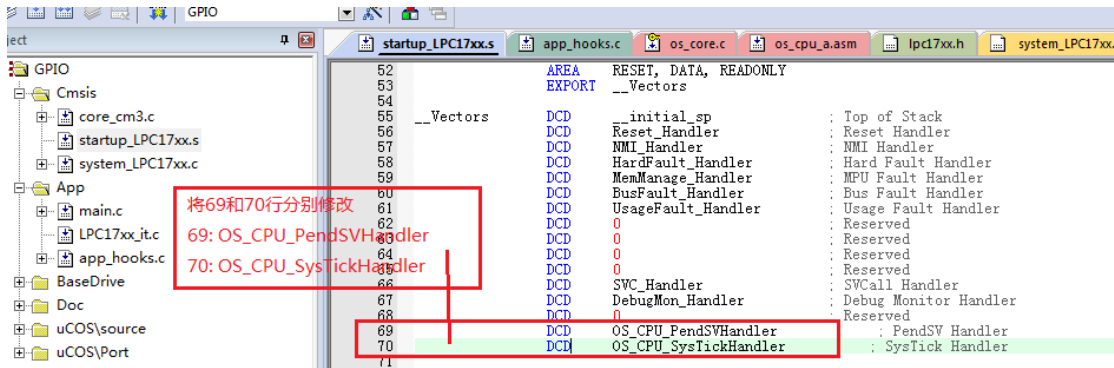


9. 将.h的文件路径添加进来，如下图所示



10. 将启动文件 startup\_LPC17XX.s 中的 69 行和 70 行修改为：如下图所示

69: OS\_CPU\_PendSVHandler  
70: OS\_CPU\_SysTickHandler



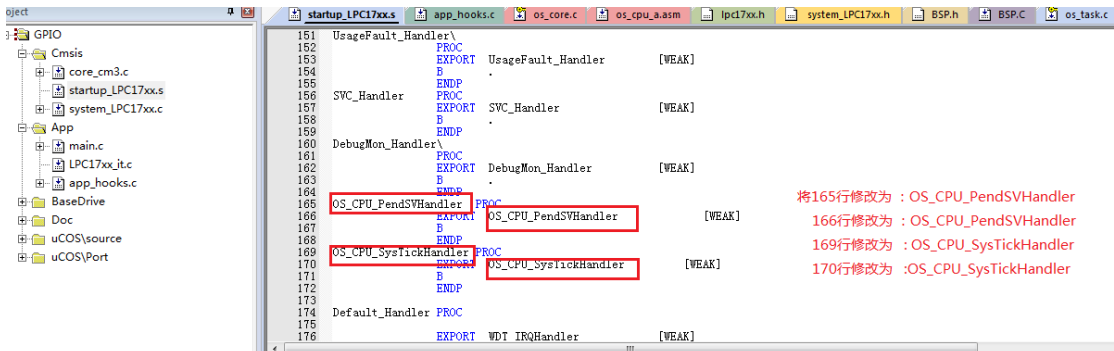
11. 将启动文件 startup\_LPC17XX.s 文件修改，如下图所示

165 行修改为: OS\_CPU\_PendSVHandler

166 行修改为 : OS\_CPU\_PendSVHandler

169 行修改为: OS\_CPU\_SysTickHandler

170 行修改为 : OS\_CPU\_SysTickHandler



12. 实现两个任务的功能

```

52
53 /*****
54 ** 函数名称: static void AppTask1 (void *p_arg)
55 ** 功能描述: 任务1 led灯闪烁
56 ** 入口参数: p_arg
57 ** 出口参数: 无
58 *****/
59 void AppTask1 (void *p_arg)
60 {
61     while(1)
62     {
63         LPC_GPIO2->FIOPIN0 |= 0xff;
64         OSTimeDlyHMSM(0, 0, 1, 0);
65     }
66 }
67
68
69

```

```

69
70 /*****
71 ** 函数名称: static void AppTask2 (void *p_arg)
72 ** 功能描述: 任务2 LED灯闪烁三次后，就不闪烁，任务被删除
73 ** 入口参数: p_arg
74 ** 出口参数: 无
75 *****/
76 void AppTask2 (void *p_arg)
77 {
78     while(1)
79     {
80         INT8U i;
81         for(i=0;i<6;i++)
82         {
83             LPC_GPIO2->FIOPIN0 = 0x00;
84             OSTimeDlyHMSM(0, 0, 0, 500);
85         }
86     }
87     OSTaskDel (APP_TASK2_PRIO); //删除任务2
88 }
89

```

### 13. 创建两个任务

```
81
82 static void App_TaskStart (void)
83 {
84     INT8U err;
85
86     BSP_Start();
87     OSTaskCreateExt(AppTask1,
88                     (void *)0,
89                     (OS_STK *)&AppTask1Stk[APP_TASK1_STK_SIZE-1],
90                     APP_TASK1_PRIO,
91                     APP_TASK1_PRIO,
92                     (OS_STK *)&AppTask1Stk[0],
93                     APP_TASK1_STK_SIZE,
94                     (void *)0,
95                     OS_TASK_OPT_STK_CHK|OS_TASK_OPT_STK_CLR): //创建任务1
96
97     OSTaskNameSet(APP_TASK1_PRIO, "AppTask1", &err);
98
99     OSTaskCreateExt(AppTask2,
100                    (void *)0,
101                    (OS_STK *)&AppTask2Stk[APP_TASK2_STK_SIZE-1],
102                    APP_TASK2_PRIO,
103                    APP_TASK2_PRIO,
104                    (OS_STK *)&AppTask2Stk[0],
105                    APP_TASK2_STK_SIZE,
106                    (void *)0,
107                    OS_TASK_OPT_STK_CHK|OS_TASK_OPT_STK_CLR): //创建任务2
108
109     OSTaskNameSet(APP_TASK2_PRIO, "AppTask2", &err);
110 }
111
112
```

### 14. 主函数在创建一个任务

```
46
47 /******
48 ** 函数名称: int main(void)
49 ** 功能描述: 主函数入口
50 ** 入口参数: 无
51 ** 出口参数: 无
52 ** 功能描述: 初始化外设, 启动任务调度。
53 *****/
54 int main (void)
55 {
56     BSP_Init();
57
58     OSInit();
59     OSTaskCreateExt((void (*)(void *))App_TaskStart,
60                    (void *)0,
61                    (OS_STK *)&App_TaskStartStk[APP_CFG_TASK_START_STK_SIZE - 1],
62                    (INT8U)APP_CFG_TASK_START_PRIO,
63                    (INT16U)APP_CFG_TASK_START_PRIO,
64                    (OS_STK *)&App_TaskStartStk[0],
65                    (INT32U)APP_CFG_TASK_START_STK_SIZE,
66                    (void *)0,
67                    (INT16U) (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
68
69     OSStart();
70     return(1);
71 }
```

15. 编译文件, 如果没有错误, 开发板的实验现象是 LED 等闪烁三次, 后就一直亮。