基于 FRDM-KE02Z40M 的 ucos-ii 移植

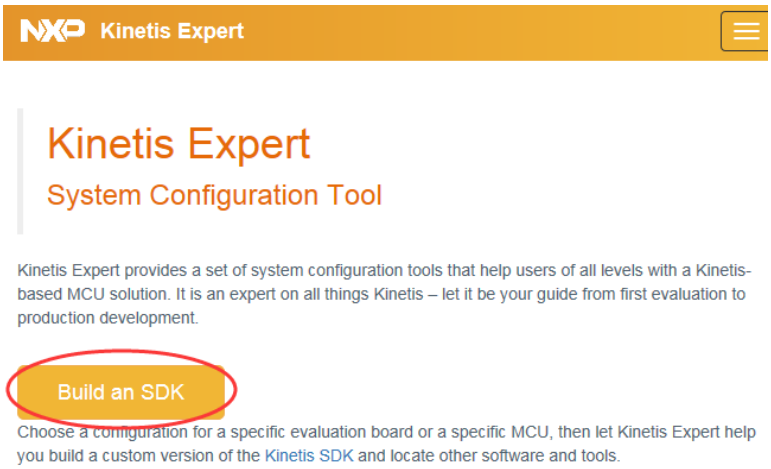
作者：Jianhui Tong
邮箱：Jianhui.tong@nxp.com

第一部分 准备工作

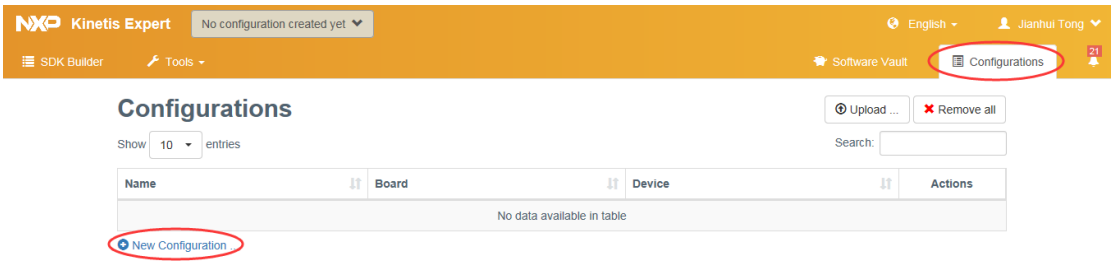
1. 下载 FRDM-KEXX Driver Library Package，即 KEXX_DRIVERS_V1.2.1_DEVD.zip 文件。

   下载链接为：
   http://www.nxp.com/webapp/sps/download/license.jsp?colCode=KEXX_DRIVERS_V1.2.1_DEVD&location=null&fromsite=zh-Hans&fsrch=1&sr=1&pageNum=1&Parent_nodeId=&Parent_pageType

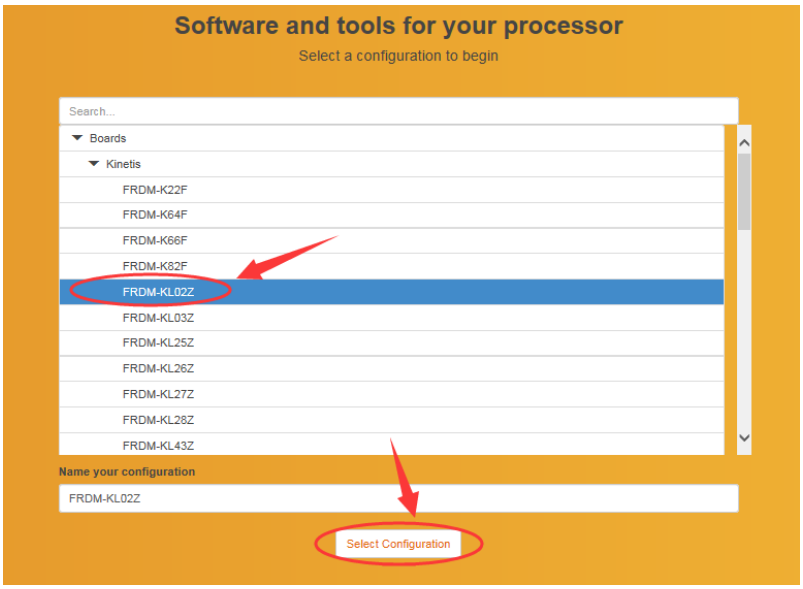
2. 下载 KL02 SDK2.0，下载时需要勾选 ucos-ii。
   访问 Kinetis Expet 网站：http://kex.nxp.com/en/welcome，点击“Build an SDK”。



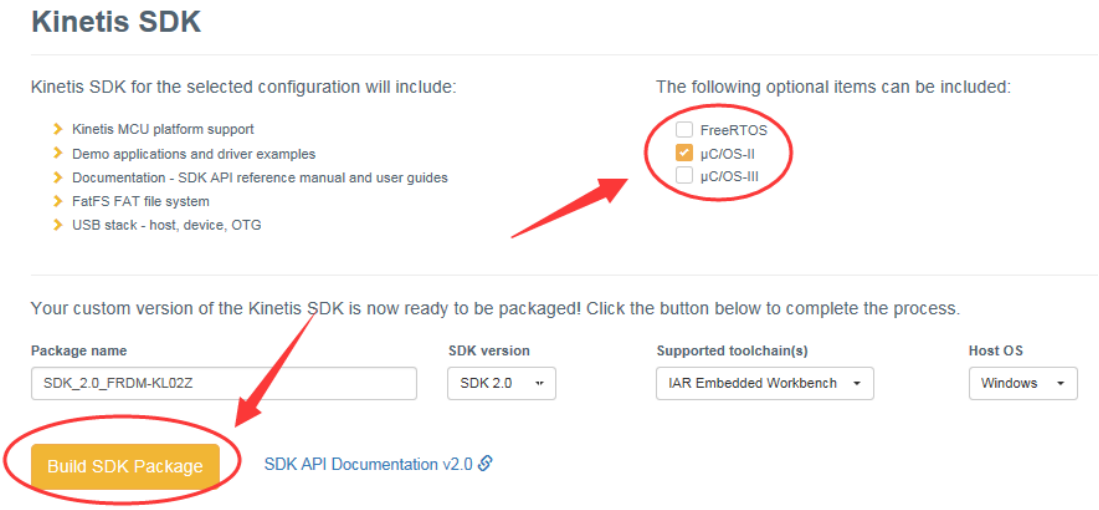
   选择”Configuration”标签页，并点击”New Configuration”新建配置。



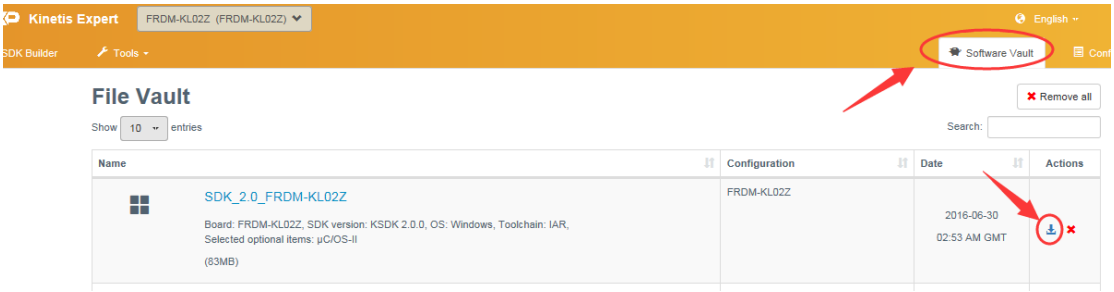
   选择 FRDM-KL02Z 开发板，并点击”Select Configuration”。



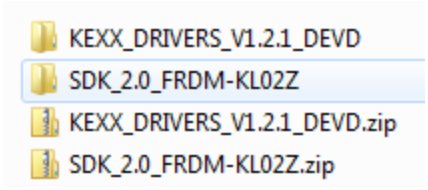
勾选 ucos-ii，并点击”Build SDK Package”。



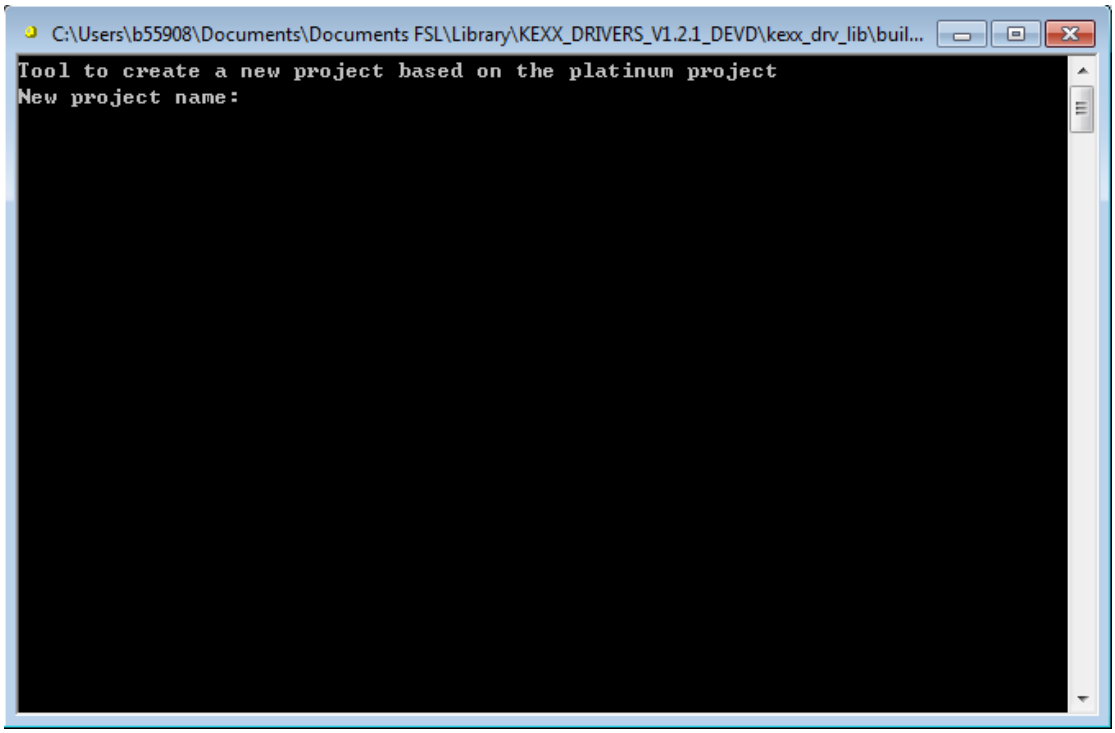
切换到”Software Vault”标签页，并下载刚刚 Build 完成的 SDK。



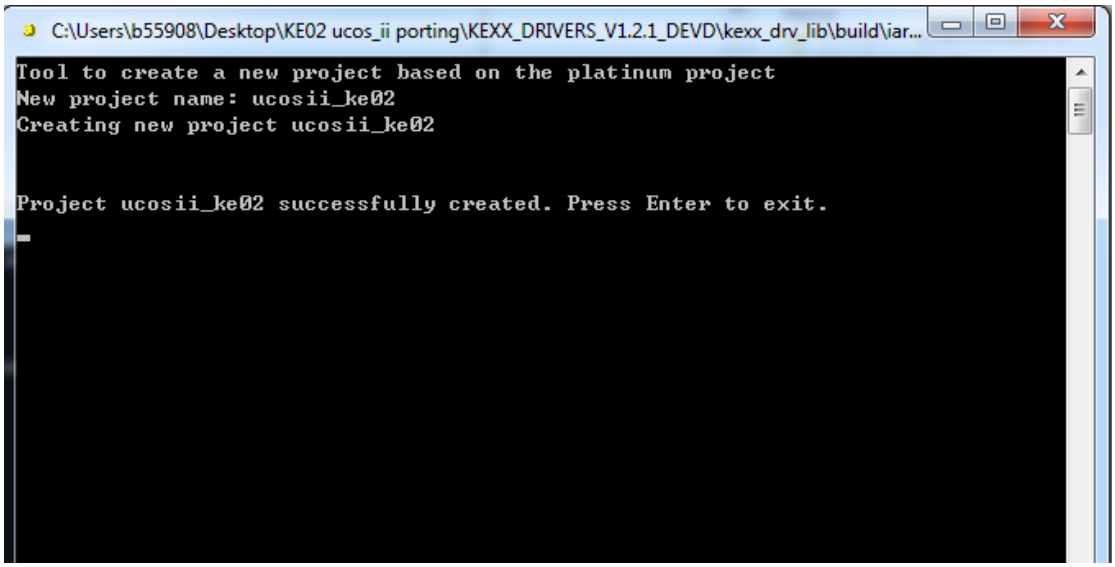
3. 将下载好的 KE 驱动库和 SDK 库解压



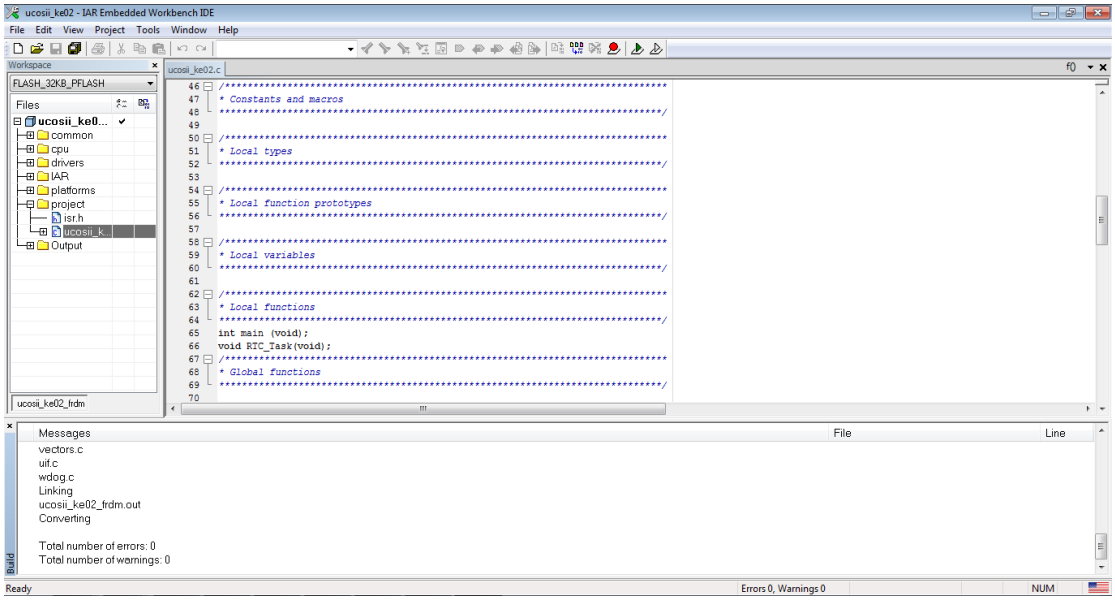
4. 运行..\KEXX_DRIVERS_V1.2.1_DEVD\kexx_drv_lib\build\iar\ke02 目录下的 make_new_project_ke02.exe 文件。



5. 输入新建工程名字，如 ucosii_ke02，回车确认。



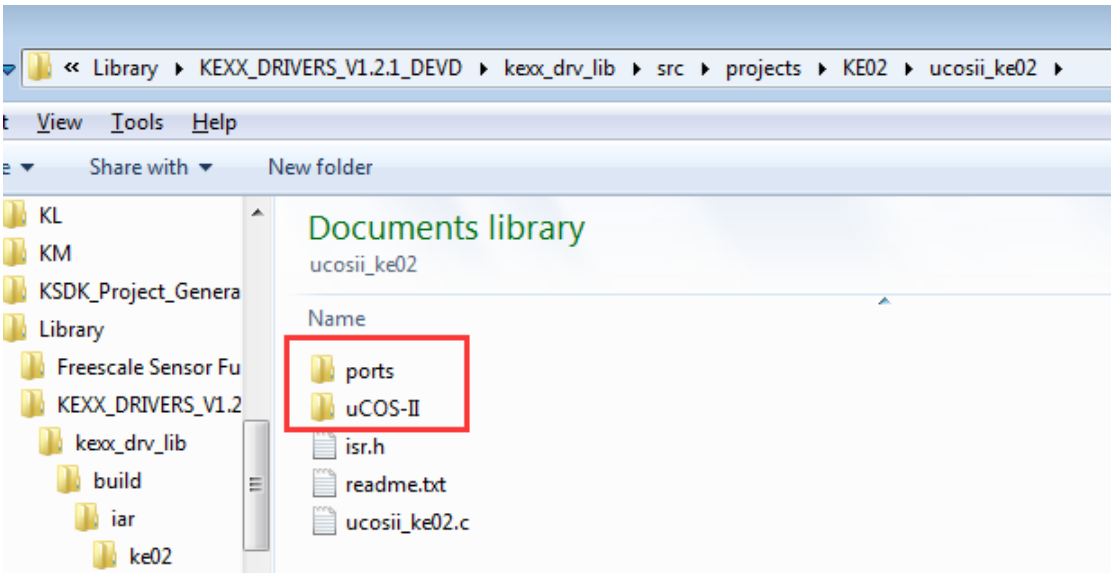
6. 新建完成后，再次按下回车键退出。
7. 打开..\KEXX_DRIVERS_V1.2.1_DEVD\kexx_drv_lib\build\iar\ke02\ucosii_ke02 目录下的 ucosii_ke02.eww 文件，并编译工程。



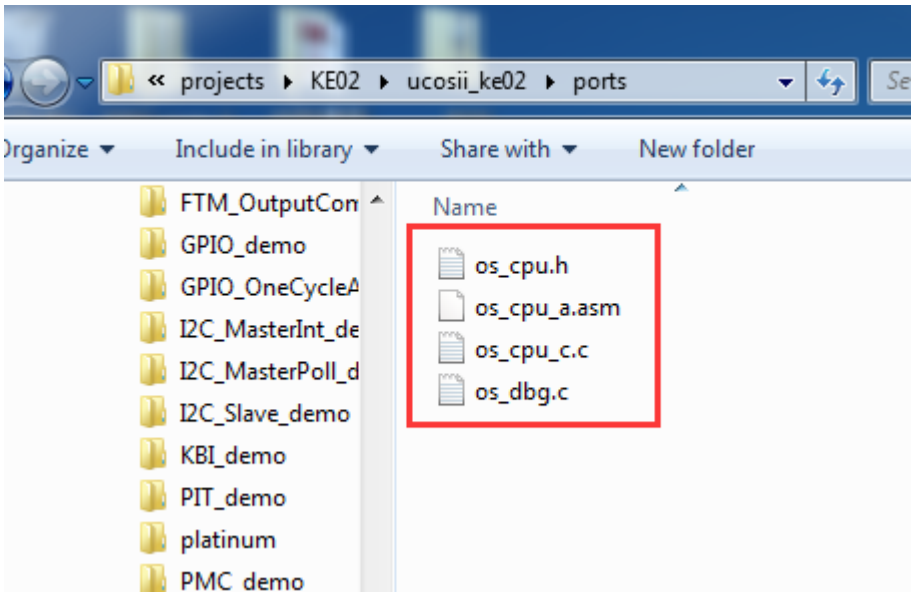
8. 此时，即得到了使用 KE driver 库的 Demo 工程，后面移植工作将基于此工程进行修改。

第二部分 移植 ucos-ii

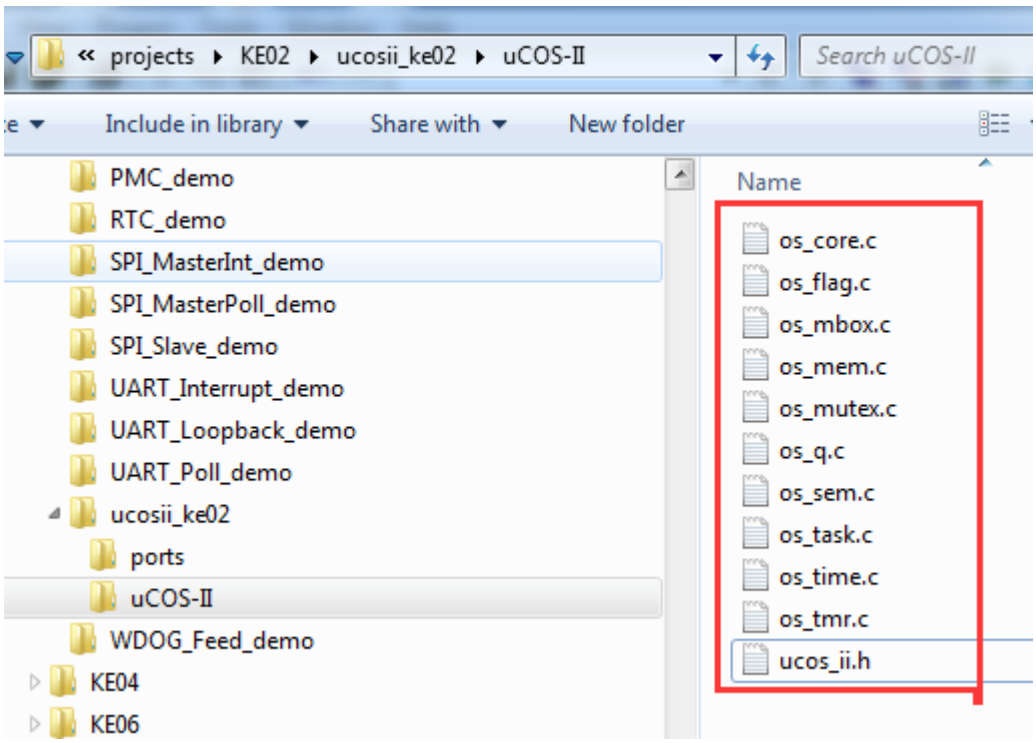
1. ..\KEXX_DRIVERS_V1.2.1_DEVD\kexx_drv_lib\src\projects\KE02\ucosii_ke02 目录下新建两个文件夹，并分别命名为“ports”，“uCOS-II”。



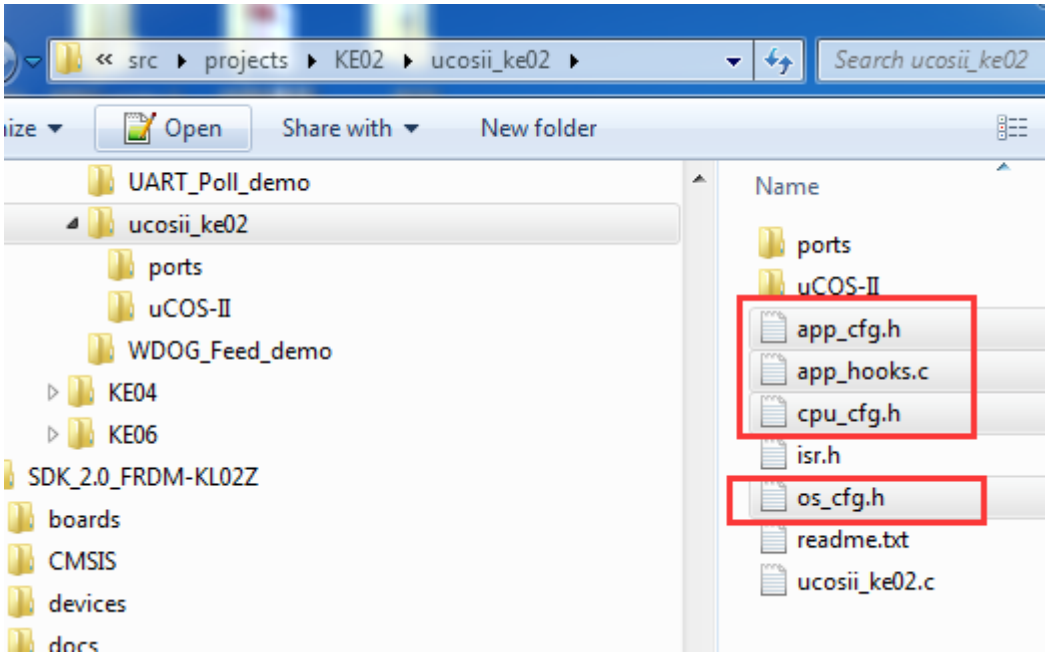
2. 将 ..\SDK_2.0_FRDM-KL02Z\rtos\ucosii_2.92.11\uCOS-II\Ports\ARM-Cortex-M0\Generic\IAR 目录下的 4 个文件 os_cpu.h, os_cpu_a.asm, os_cpu_c.c, os_dbg.c 复制到 ..\KEXX_DRIVERS_V1.2.1_DEVD\kexx_drv_lib\src\projects\KE02\ucosii_ke02\ports



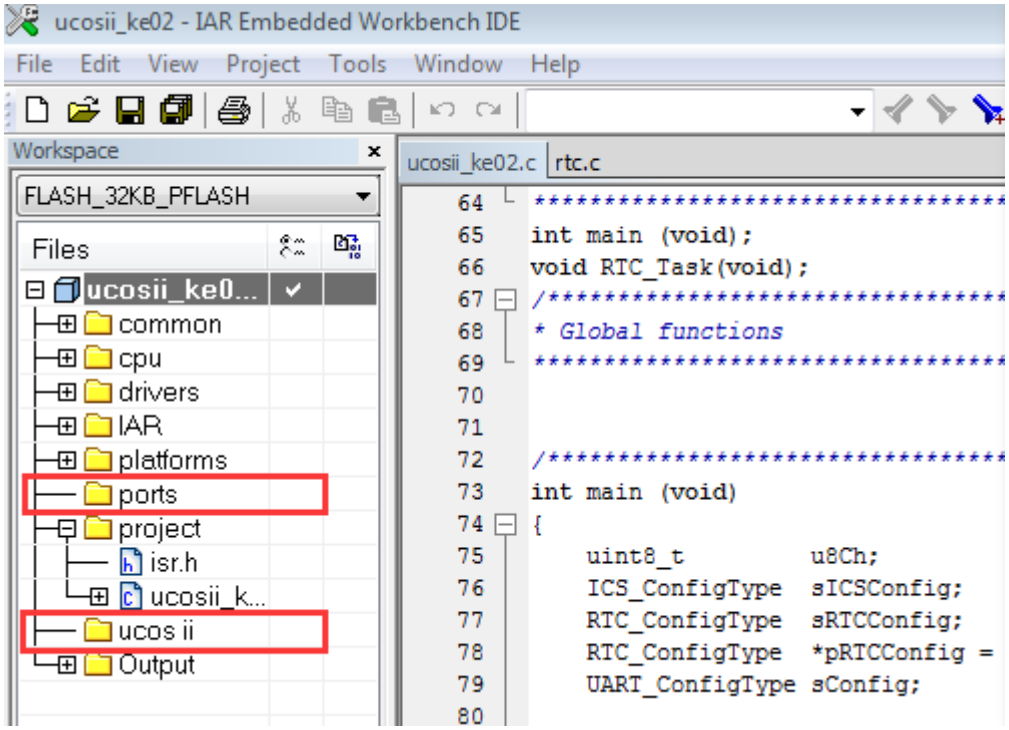
3. 将 ..\SDK_2.0_FRDM-KL02Z\rtos\ucosii_2.92.11\uCOS-II\Source 目录下的 11 个文件 os_core.c, os_flag.c, os_mbox.c, os_mem.c, os_mutex.c, os_q.c, os_sem.c, os_task.c, os_time.c, os_tmr.c, ucos_ii.h 复制到 ..\KEXX_DRIVERS_V1.2.1_DEVD\kexx_drv_lib\src\projects\KE02\ucosii_ke02\uCOS-II



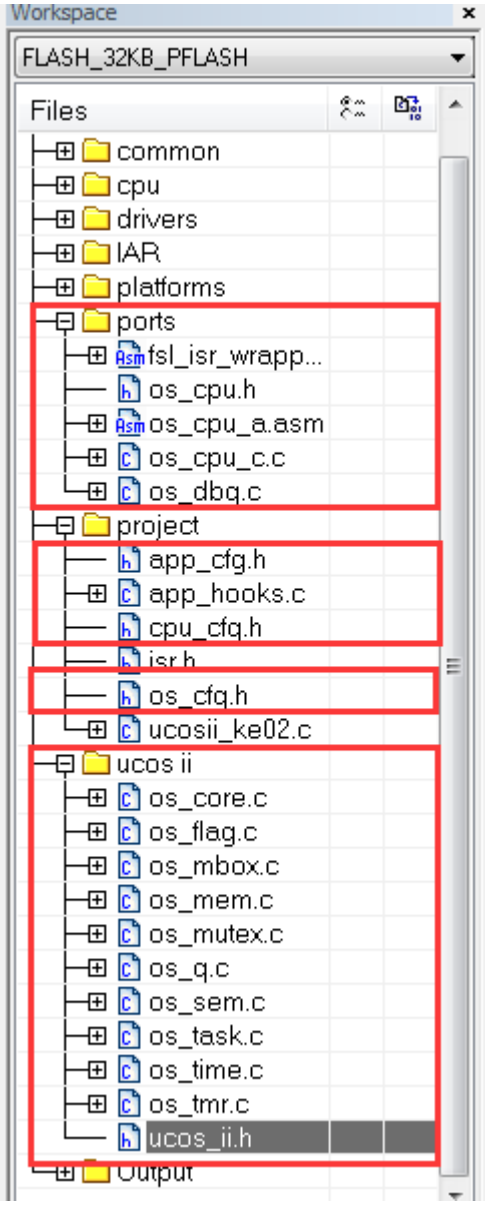
4. 将..\SDK_2.0_FRDM-KL02Z\rtos\ucosii_2.92.11\template_application 目录下的 4 个文件 app_cfg.h, app_hooks.c, cpu_cfg.h, os_cfg.h 复制到 ..\KEXX_DRIVERS_V1.2.1_DEVD\kexx_drv_lib\src\projects\KE02\ucosii_ke02



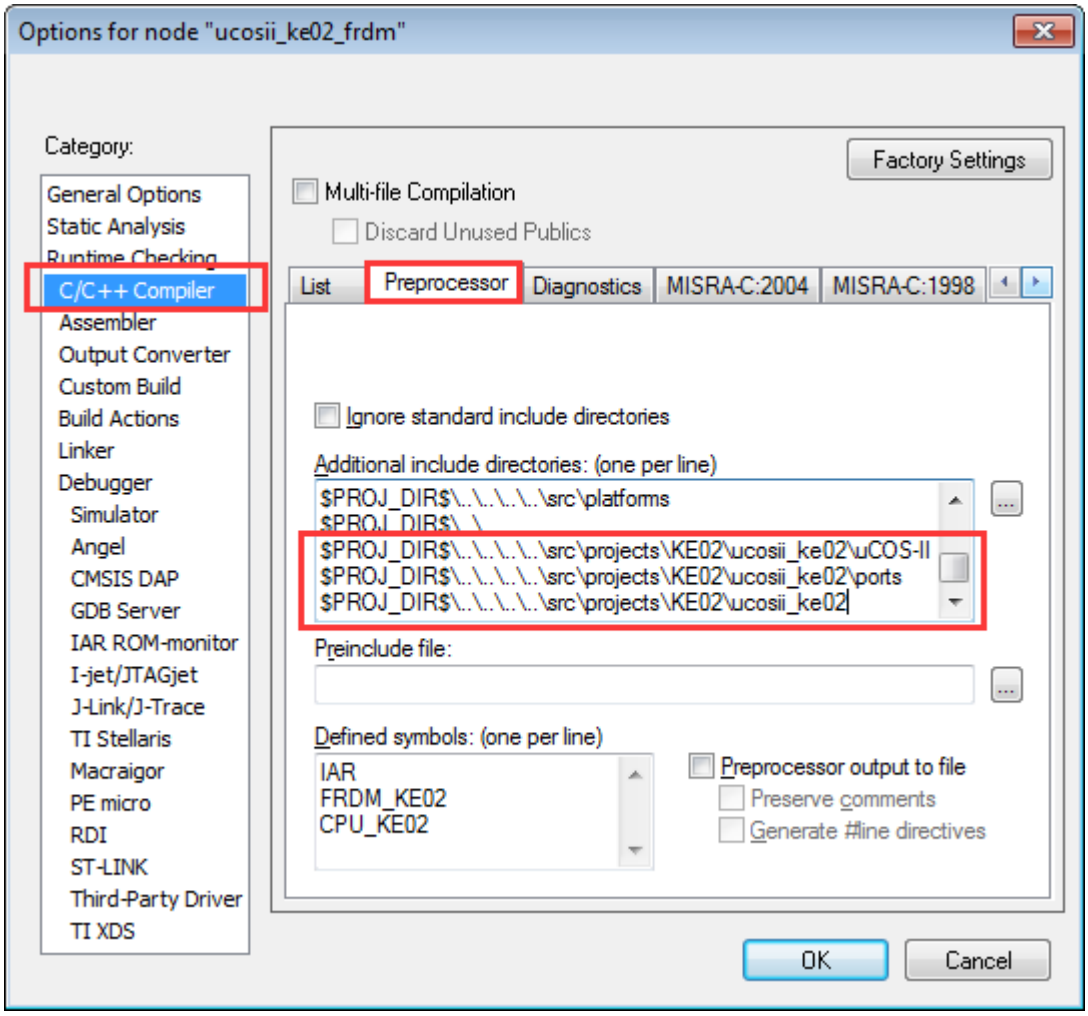
在 IAR 工程目录中新建两个文件夹“ports”和“ucos ii”。



5. 将上述步骤中复制的文件添加到 IAR 工程中，如下图所示。



6. 修改编译器搜索路径。在 IAR 中打开 Project > Options > C/C++ Compiler >Preprocessor，在“Additional include directories”一栏中添加上文复制文件所在的路径。



7. 修改 app_cfg.h 文件，添加下图红框中圈出的代码。

```
#ifndef _APP_CFG_H_
#define _APP_CFG_H_

#define APP_CFG_TASK_START_PRIO          2u
#define APP_CFG_TASK_LED_R_PRIO          3u
#define APP_CFG_TASK_LED_G_PRIO          4u
#define APP_CFG_TASK_LED_B_PRIO          5u

#define APP_CFG_TASK_START_STK_SIZE      32u
#define APP_CFG_TASK_LED_R_STK_SIZE      32u
#define APP_CFG_TASK_LED_G_STK_SIZE      32u
#define APP_CFG_TASK_LED_B_STK_SIZE      32u

/* Timer task priority */
#define OS_TASK_TMR_PRIO (OS_LOWEST_PRIO - 2u)

#endif /* _APP_CFG_H_ */
```

8. 修改 ucosii_ke02.c 文件，用下面的代码替换。

```
/*****************************************************************************
*
* Freescale Semiconductor Inc.
* (c) Copyright 2013 Freescale Semiconductor, Inc.
* ALL RIGHTS RESERVED.
*
*****************************************************************************
*
* THIS SOFTWARE IS PROVIDED BY FREESCALE "AS IS" AND ANY EXPRESSED OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL FREESCALE OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
* INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
* IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*
*****************************************************************************//*!
*
* @file uCOS-II_KE02.c
*
* @author Freescale
*
* @version 0.0.1
*
* @date Jun. 25, 2013
*
* @brief providing framework of demo cases for MCU.
*
*****************************************************************************/

#include "common.h"
#include "ics.h"
#include "uart.h"
#include "sysinit.h"
#include "ucos_ii.h"
#include "system_MKE02Z2.h"

/*****************************************************************************
* Global variables
```

```
*********************************************************************************/

/*********************************************************************************
* Constants and macros
*********************************************************************************/

/*********************************************************************************
* Local types
*********************************************************************************/

/*********************************************************************************
* Local function prototypes
*********************************************************************************/

/*********************************************************************************
* Local variables
*********************************************************************************/
static OS_STK      AppTaskStartStk[APP_CFG_TASK_START_STK_SIZE];
static OS_STK      AppTaskLedRStk[APP_CFG_TASK_LED_R_STK_SIZE];
static OS_STK      AppTaskLedGStk[APP_CFG_TASK_LED_G_STK_SIZE];
static OS_STK      AppTaskLedBStk[APP_CFG_TASK_LED_B_STK_SIZE];

static OS_EVENT* RSem;
static OS_EVENT* GSem;
static OS_EVENT* BSem;
static OS_EVENT* OffSem;
/*********************************************************************************
* Local functions
*********************************************************************************/
int main (void);
static void AppTaskStart (void *p_arg);
static void AppTaskLedR(void *p_arg);
static void AppTaskLedG(void *p_arg);
static void AppTaskLedB(void *p_arg);
/*********************************************************************************
* Global functions
*********************************************************************************/


/*********************************************************************/
int main (void)
{

    OSInit();
```

```
        OSTaskCreate(AppTaskStart,  (void*)0,  &AppTaskStartStk[APP_CFG_TASK_START_STK_SIZE-1],
(INT8U)APP_CFG_TASK_START_PRIO );

        OSStart();

        return (1);

    }

    static void AppTaskStart (void *p_arg)
    {
      INT8U err;

      (void)p_arg;

      sysinit();
      OS_CPU_SysTickInit(20000000/OS_TICKS_PER_SEC);

      PORT->HDRVE |=0x02;

      LED0_Init();
      LED1_Init();
      LED2_Init();
      LED0_Off();
      LED1_Off();
      LED2_Off();

      RSem = OSSemCreate(1);
      GSem = OSSemCreate(1);
      BSem = OSSemCreate(1);
      OffSem = OSSemCreate(1);

      OSTaskCreate(AppTaskLedR,  (void*)0,  &AppTaskLedRStk[APP_CFG_TASK_LED_R_STK_SIZE-1],
(INT8U)APP_CFG_TASK_LED_R_PRIO );
      OSTaskCreate(AppTaskLedG,  (void*)0,  &AppTaskLedGStk[APP_CFG_TASK_LED_G_STK_SIZE-1],
(INT8U)APP_CFG_TASK_LED_G_PRIO );
      OSTaskCreate(AppTaskLedB,  (void*)0,  &AppTaskLedBStk[APP_CFG_TASK_LED_B_STK_SIZE-1],
(INT8U)APP_CFG_TASK_LED_B_PRIO );

      while (1)
      {
        OSTimeDlyHMSM(0,0,3,0);
        OSSemPost(RSem);
```

```
        OSSemPend(OffSem, 0, &err);

    }
}

static void AppTaskLedR(void *p_arg)
{
    INT8U err;

    while(1)
    {
        OSSemPend(RSem,0,&err);
        LED0_On();LED2_Off();
        OSTimeDlyHMSM(0,0,1,0);
        OSSemPost(GSem);
    }
}

static void AppTaskLedG(void *p_arg)
{
    INT8U err;

    while(1)
    {
        OSSemPend(GSem,0,&err);
        LED1_On();LED0_Off();
        OSTimeDlyHMSM(0,0,1,0);
        OSSemPost(BSem);
    }
}

static void AppTaskLedB(void *p_arg)
{
    INT8U err;

    while(1)
    {
        OSSemPend(BSem,0,&err);
        LED2_On();LED1_Off();
        OSTimeDlyHMSM(0,0,1,0);
        OSSemPost(OffSem);
    }
}
```

9. 修改 os_cfg.h 文件，用下面的代码替换。（由于 KE02 的 RAM 较小，因此关闭了部分未使用的功能）

```
    /*
    *********************************************************************************************************
    *                                                uC/OS-II
    *                                          The Real-Time Kernel
    *                                  uC/OS-II Configuration File for V2.9x
    *
    *                                (c) Copyright 2005-2013, Micrium, Weston, FL
    *                                           All Rights Reserved
    *
    *
    * File        : OS_CFG.H
    * By          : Jean J. Labrosse
    * Version : V2.92.11
    *
    * LICENSING TERMS:
    * ---------------
    *     uC/OS-II is provided in source form for FREE evaluation, for educational use or for peaceful research.
    * If you plan on using  uC/OS-II  in a commercial product you need to contact Micrium to properly license
    * its use in your product. We provide ALL the source code for your convenience and to help you experience
    * uC/OS-II.     The fact that the  source is provided does  NOT  mean that you can use it without  paying a
    * licensing fee.
    *********************************************************************************************************
    */

    #ifndef OS_CFG_H
    #define OS_CFG_H

    /* ---------------------- MISCELLANEOUS ----------------------- */
    #define OS_APP_HOOKS_EN 0u /* Application-defined hooks are called from the uC/OS-II hooks */
    #define OS_ARG_CHK_EN 0u      /* Enable (1) or Disable (0) argument checking */
    #define OS_CPU_HOOKS_EN 1u /* uC/OS-II hooks are found in the processor port files */
```

```
    #define OS_DEBUG_EN 0u /* Enable(1) debug variables */

    #define OS_EVENT_MULTI_EN 0u /* Include code for OSEventPendMulti() */
    #define OS_EVENT_NAME_EN 0u  /* Enable names for Sem, Mutex, Mbox and Q */

    #define OS_LOWEST_PRIO 63u /* Defines the lowest priority that can be assigned ... */
                                /* ... MUST NEVER be higher than 254! */

    #define OS_MAX_EVENTS 10u  /* Max. number of event control blocks in your application */
    #define OS_MAX_FLAGS 5u    /* Max. number of Event Flag Groups in your application */
    #define OS_MAX_MEM_PART 5u /* Max. number of memory partitions */
    #define OS_MAX_QS 4u       /* Max. number of queue control blocks in your application */
    #define OS_MAX_TASKS 10u   /* Max. number of tasks in your application, MUST be >= 2 */

    #define OS_SCHED_LOCK_EN 1u /* Include code for OSSchedLock() and OSSchedUnlock() */

    #define OS_TICK_STEP_EN 1u    /* Enable tick stepping feature for uC/OS-View */
    #define OS_TICKS_PER_SEC 1000u /* Set the number of ticks in one second */

    #define OS_TLS_TBL_SIZE 0u /* Size of Thread-Local Storage Table */

    /* ---------------------- TASK STACK SIZE ---------------------- */
    #define OS_TASK_TMR_STK_SIZE 64u  /* Timer task stack size (# of OS_STK wide entries) */
    #define OS_TASK_STAT_STK_SIZE 64u /* Statistics task stack size (# of OS_STK wide entries) */
    #define OS_TASK_IDLE_STK_SIZE 64u /* Idle task stack size (# of OS_STK wide entries) */

    /* ---------------------- TASK MANAGEMENT ---------------------- */
```

```
    #define OS_TASK_CHANGE_PRIO_EN  1u      /*              Include code for OSTaskChangePrio()                         */
    #define OS_TASK_CREATE_EN 1u            /*          Include code for OSTaskCreate()                                 */
    #define OS_TASK_CREATE_EXT_EN 1u    /*          Include code for OSTaskCreateExt()                              */
    #define OS_TASK_DEL_EN 0u               /*          Include code for OSTaskDel()                                    */
    #define OS_TASK_NAME_EN  0u             /*              Enable task names                                           */
    #define OS_TASK_PROFILE_EN 0u           /*          Include variables in OS_TCB for profiling                       */
    #define OS_TASK_QUERY_EN 0u             /*          Include code for OSTaskQuery()                                  */
    #define OS_TASK_REG_TBL_SIZE 1u         /*          Size of task variables array (#of INT32U entries)               */
    #define OS_TASK_STAT_EN 0u              /*          Enable (1) or Disable(0) the statistics task                    */
    #define OS_TASK_STAT_STK_CHK_EN 0u /*          Check task stacks from statistic task                           */
    #define OS_TASK_SUSPEND_EN 0u           /*          Include code for OSTaskSuspend() and OSTaskResume()             */
    #define OS_TASK_SW_HOOK_EN 1u           /*          Include code for OSTaskSwHook()                                 */

    /* ---------------------- EVENT FLAGS ------------------------ */
    #define OS_FLAG_EN 0u               /* Enable (1) or Disable (0) code generation for EVENT FLAGS       */
    #define OS_FLAG_ACCEPT_EN 1u        /*          Include code for OSFlagAccept()                                 */
    #define OS_FLAG_DEL_EN 1u           /*          Include code for OSFlagDel()                                    */
    #define OS_FLAG_NAME_EN 1u          /*          Enable names for event flag group                               */
    #define OS_FLAG_QUERY_EN 1u         /*          Include code for OSFlagQuery()                                  */
    #define OS_FLAG_WAIT_CLR_EN 1u /* Include code for Wait on Clear EVENT FLAGS                                    */
    #define OS_FLAGS_NBITS 16u          /* Size in #bits of OS_FLAGS data type (8, 16 or 32)               */

    /* --------------------- MESSAGE MAILBOXES ---------------------- */
    #define OS_MBOX_EN 0u               /* Enable (1) or Disable (0) code generation for MAILBOXES         */
```

```
    #define OS_MBOX_ACCEPT_EN 1u        /*       Include code for OSMboxAccept()
*/
    #define OS_MBOX_DEL_EN 1u           /*       Include code for OSMboxDel()
*/
    #define OS_MBOX_PEND_ABORT_EN 1u /*              Include code for
OSMboxPendAbort()                                 */
    #define OS_MBOX_POST_EN 1u          /*       Include code for OSMboxPost()
*/
    #define OS_MBOX_POST_OPT_EN 1u    /*       Include code for OSMboxPostOpt()
*/
    #define OS_MBOX_QUERY_EN 1u         /*       Include code for OSMboxQuery()
*/

    /* ---------------------- MEMORY MANAGEMENT ---------------------- */
    #define OS_MEM_EN 0u            /* Enable (1) or Disable (0) code generation for
MEMORY MANAGER */
    #define OS_MEM_NAME_EN 1u    /*          Enable memory partition names
*/
    #define OS_MEM_QUERY_EN 1u /*           Include code for OSMemQuery()
*/

    /* ---------------- MUTUAL EXCLUSION SEMAPHORES ---------------- */
    #define OS_MUTEX_EN 0u              /* Enable (1) or Disable (0) code generation for
MUTEX               */
    #define OS_MUTEX_ACCEPT_EN 1u /*         Include code for OSMutexAccept()
*/
    #define OS_MUTEX_DEL_EN 1u        /*         Include code for OSMutexDel()
*/
    #define OS_MUTEX_QUERY_EN 1u    /*       Include code for OSMutexQuery()
*/

    /* ---------------------- MESSAGE QUEUES ---------------------- */
    #define OS_Q_EN 0u                  /* Enable (1) or Disable (0) code generation for
QUEUES              */
    #define OS_Q_ACCEPT_EN 1u           /*         Include code for OSQAccept()
*/
    #define OS_Q_DEL_EN 1u              /*         Include code for OSQDel()
*/
    #define OS_Q_FLUSH_EN 1u            /*         Include code for OSQFlush()
*/
    #define OS_Q_PEND_ABORT_EN 1u /*         Include code for OSQPendAbort()
*/
    #define OS_Q_POST_EN 1u             /*         Include code for OSQPost()
*/
```

```
    #define  OS_Q_POST_FRONT_EN  1u  /*          Include code for OSQPostFront()
*/
    #define  OS_Q_POST_OPT_EN  1u       /*          Include code for OSQPostOpt()
*/
    #define  OS_Q_QUERY_EN  1u          /*          Include code for OSQQuery()
*/

    /* ------------------------ SEMAPHORES ------------------------ */
    #define OS_SEM_EN 1u                /* Enable (1) or Disable (0) code generation for
SEMAPHORES        */
    #define  OS_SEM_ACCEPT_EN  1u       /*        Include code for OSSemAccept()
*/
    #define  OS_SEM_DEL_EN  1u          /*        Include code for OSSemDel()
*/
    #define OS_SEM_PEND_ABORT_EN 1u /*       Include code for OSSemPendAbort()
*/
    #define  OS_SEM_QUERY_EN  1u        /*        Include code for OSSemQuery()
*/
    #define  OS_SEM_SET_EN  1u          /*        Include code for OSSemSet()
*/

    /* --------------------- TIME MANAGEMENT ---------------------- */
    #define OS_TIME_DLY_HMSM_EN 1u     /*        Include code for OSTimeDlyHMSM()
*/
    #define OS_TIME_DLY_RESUME_EN 1u /*        Include code for OSTimeDlyResume()
*/
    #define OS_TIME_GET_SET_EN 1u      /*        Include code for OSTimeGet() and
OSTimeSet()                */
    #define OS_TIME_TICK_HOOK_EN 1u    /*        Include code for OSTimeTickHook()
*/

    /* ---------------------- TIMER MANAGEMENT ---------------------- */
    #define OS_TMR_EN 0u                /* Enable (1) or Disable (0) code
generation for TIMERS              */
    #define OS_TMR_CFG_MAX 16u          /*        Maximum number of timers
*/
    #define OS_TMR_CFG_NAME_EN 1u       /*        Determine timer names
*/
    #define OS_TMR_CFG_WHEEL_SIZE 7u    /*        Size of timer wheel (#Spokes)
*/
    #define OS_TMR_CFG_TICKS_PER_SEC 10u /*        Rate at which timer management
task runs (Hz)             */

#endif
```

10. 修改 isr.h 文件。

```
#ifndef __ISR_H
#define __ISR_H


/* Example */
/*
#undef  VECTOR_036
#define VECTOR_036 RTC_Isr

// ISR(s) are defined in your project directory.
extern void RTC_Isr(void);
*/

/*!
 * @brief define interrupt service routine for different vectors.
 *
 */
#undef  VECTOR_014
#define VECTOR_014      OS_CPU_PendSVHandler
#undef  VECTOR_015
#define VECTOR_015      OS_CPU_SysTickHandler

extern void OS_CPU_PendSVHandler(void);
extern void OS_CPU_SysTickHandler(void);

#endif  //__ISR_H
```

11. 完成后保存，并重新编译工程。将下载程序到 FRDM-KE02Z40M 中，复位运行程序就可以看到 LED 灯的颜色随时间变化啦！