

# MDK 的中间件的应用 V1.0

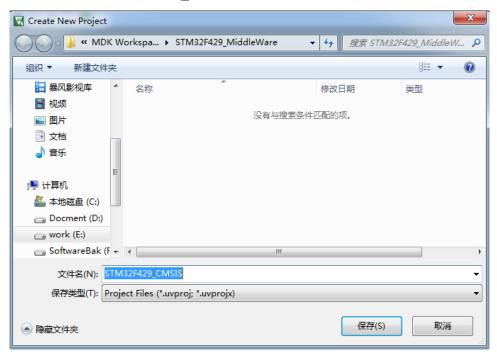
文档类别: 代理产品技术文档

版本信息		
版本号	编写日期	编写人
V 1.0	2016年08月16日	刘立明

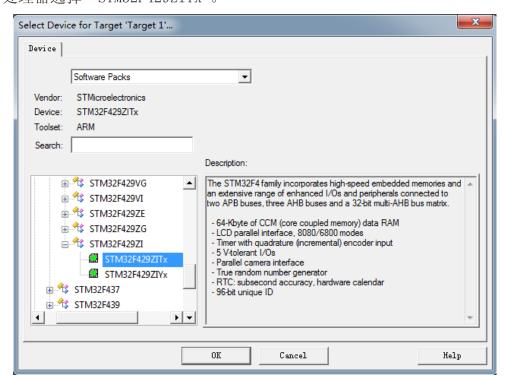


### 一、基础工程序的创建

1、创建一个名为 "STM32F429 CMSIS"的工程。

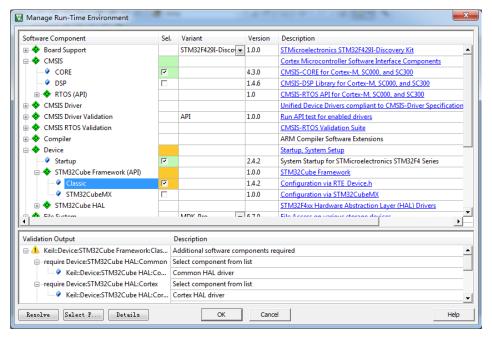


2、处理器选择"STM32F429ZITx"。

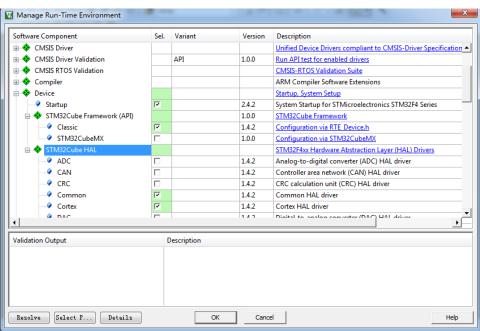




3、选择 "Startup"和 "Classic"组件。

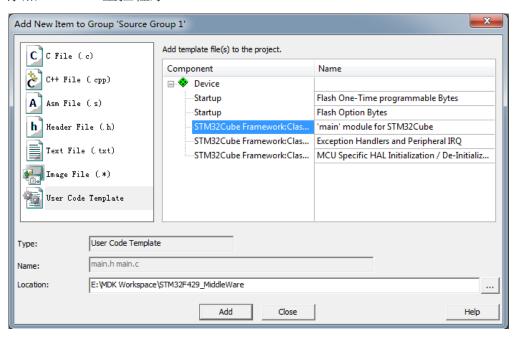


4、选择好相关联的组件。

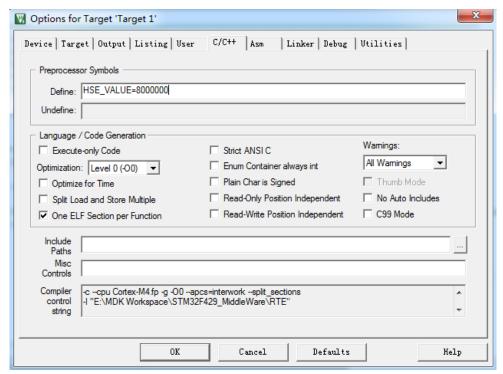




5、添加"main"主控程序。

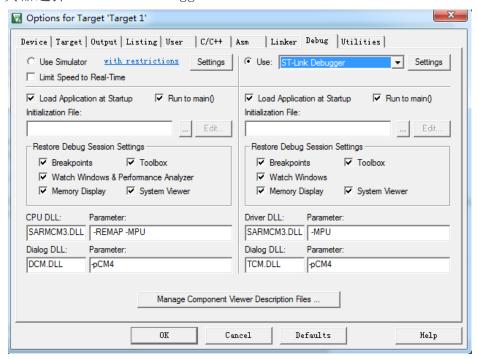


6、添加预处理符号: HSE\_VALUE=8000000。

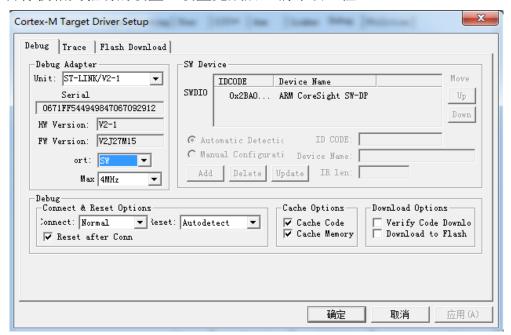




7、仿真器选择 "ST-link Debugger"。

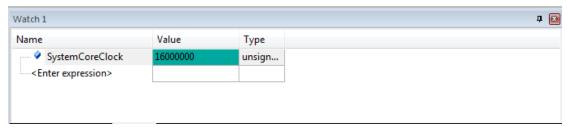


8、目标板相关驱动的设置,设置完成后,编译该工程。

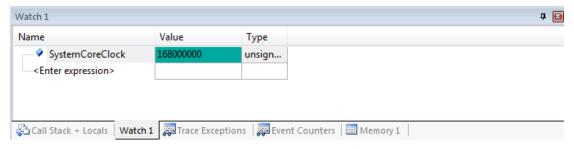




9、在调试窗体中的 Watch 窗体中添加 "SystemCoreClock"变量。



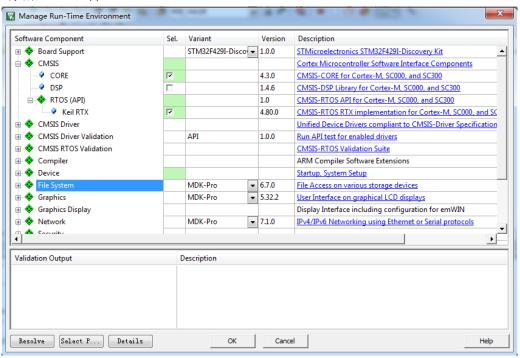
10、单步运行完 "SystemClock\_Config();"



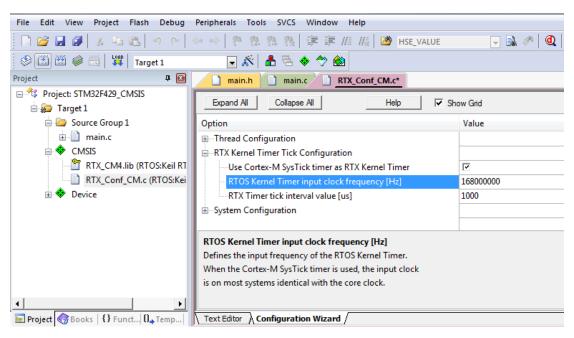


### 二、CMSIS-RTOS (RTX) 的使用

1、添加RTX组件。

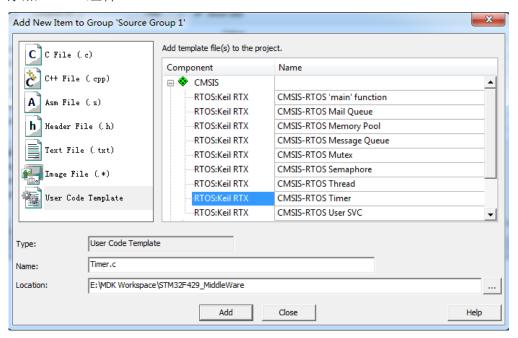


2、 修改 RTOS 的输入时钟值为:168000000.





3、添加 Timer 组件。

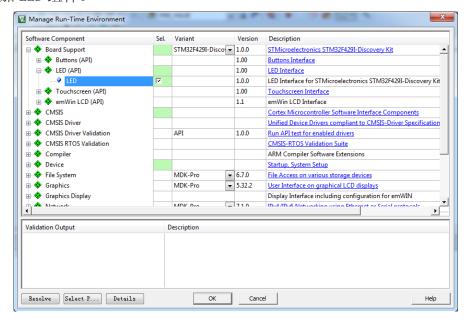


4、在"main.c"中请一下"Init\_Timers"函数,并调用该函数。

```
main.c* RTX_Conf_CM.c* Timer.c
  68 /* Private typedef ----
69 /* Private define -----
  70 /* Private macro ---
  71 /* Private variables ------
72 /* Private function prototypes ----
  73     static void SystemClock_Config(void);
74     static void Error_Handler(void);
  75 extern void Init_Timers (void);
  76
       /* Private functions -
  78 ⊟/**
         * @brief Main program
  79
         * @param None
  80
         * @retval None
  81
  82
  83 int main(void)
  84 🖵 {
main.c* RTX_Conf_CM.c* Timer.c
          /* STM32F4xx HAL library initialization:
   90
               - Configure the Flash prefetch, Flash preread and Buffer caches
   92
                - Systick timer is configured by default as source of time base, but user
                      can eventually implement his proper time base source (a general purpose timer for example or other time source), keeping in mind that {\tt Time} base
   93
   94
                       duration should be kept 1ms since PPP_TIMEOUT_VALUEs are defined and
   96
                       handled in milliseconds basis.
   97
                - Low Level Initialization
   98
  99
          HAL_Init();
 100
 101
          /* Configure the system clock to 168 MHz */
          SystemClock Config();
 102
        Init_Timers();
 104
 105
         /\star Add your application code here
  106
  107
  108 ##ifdef RTE_CMSIS_RTOS
                                                      // when using CMSIS RTOS
        // create 'thread' functions that start executing,
```



5、添加 LED 组件。



6、在"Timer.c"中添加 LED 的头文件,并添加一个时间的控制变量,再在"Timer2 Callback"函数中添加如下代码:

```
timer_cnt ++;
if(timer_cnt & 1) LED_On(0);
else LED_Off(0);
```

```
main.c* RTX_Conf_CM.c* Timer.c* LED_F429Discovery.c
      #include "cmsis_os.h"
#include "Board_LED.h"
                                                                           // CMSIS RTOS header file
                                                  // ::Board Support:LED
       static int timer_cnt = 0;
              Timer: Sample timer functions
  10
     /*---- One-Shoot Timer Example ----*/
  12 static void Timer1_Callback (void const *arg);
                                                                           // prototype for timer cal
  13
  14 static osTimerId id1;
     static uint32_t exec1;
static osTimerDef (Timer1, Timer1_Callback);
  15
                                                                           // argument for the timer
                                                                           // define timers
  16
       // One-Shoot Timer Function
  20 // add user code here
21 }
  19 = static void Timer1_Callback (void const *arg) {
```

```
main.c* RTX_Conf_CM.c* Timer.c* LED_F429Discovery.c
  25  static void Timer2_Callback (void const *arg);
                                                                           // prototype for timer cal
  26
  27
      static osTimerId id2;
                                                                          // timer id
       static uint32_t exec2;
                                                                          // argument for the timer
  29
      static osTimerDef (Timer2, Timer2_Callback);
  30
       // Periodic Timer Example
  32 = static void Timer2_Callback (void const *arg) {
        // add user code here
  33
        timer_cnt ++;
       if(timer_cnt & 1) LED_On(0);
else LED_Off(0);
  35
  36
  38
  39
      // Example: Create and Start timers
  41 \begin{aligned}
void Init_Timers (void) {
  42
         osStatus status;
                                                                          // function return status
```



7、在"main"文件中添加 LED 的头文件,并调用其初始化函数,编译工程。

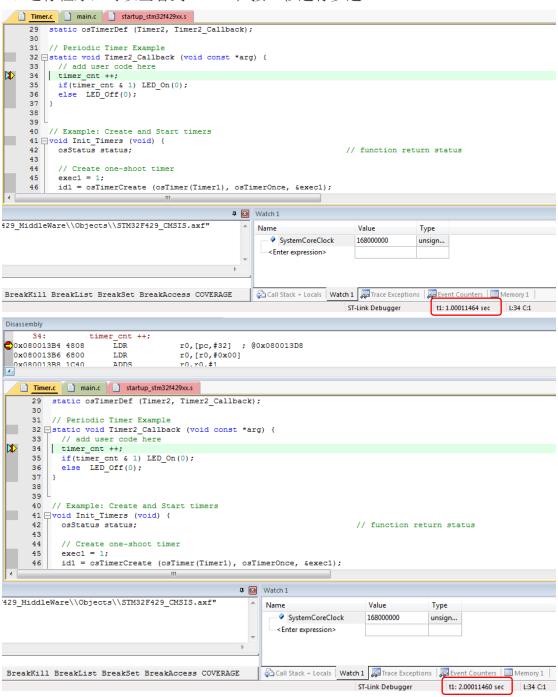
```
Timer.c main.c
    41
    42 /* Includes -----
43 #include "main.h"
    44 #include "Board LED.h"
                                                 // ::Board Support:LED
    45
    46 = #ifdef _RTE_
    47 #include "RTE_Components.h"
48 #endif
                                                 /* Component selection */
    49 ⊟#ifdef RTE CMSIS RTOS
                                                 // when RTE component CMSIS RTOS is used
    50 #include "cmsis os.h"
                                                  // CMSIS RTOS header file
    51 #endif
 Timer.c main.c
    95
                     timer for example or other time source), keeping in mind that Time base
    96
                     duration should be kept 1ms since PPP TIMEOUT VALUEs are defined and
    97
                     handled in milliseconds basis.
    98
                Low Level Initialization
    99
         HAL Init();
   100
   101
          /* Configure the system clock to 168 MHz */
   102
   103
          SystemClock Config();
   104
          Init_Timers();
   105
          LED_Initialize();
   106
   107 🖨
         /* Add your application code here
   108
```

8、调试该工程,在"Timer2 Callback"中设置一个断点。

```
Disassembly
               timer_cnt ++;
0x080013B8 1C40
                    ADDS
  Timer.c startup_stm32f429xx.s
       static osTimerDef (Timer2, Timer2 Callback);
     30
     31
        // Periodic Timer Example
     32 = static void Timer2 Callback (void const *arg) {
     33
          // add user code here
        timer cnt ++;
         if(timer cnt & 1) LED On(0);
    35
    36
         else LED_Off(0);
     37
     38
     39
     40
        // Example: Create and Start timers
     41 □void Init Timers (void) {
         osStatus status;
                                                                 // function re
     42
     43
     44
          // Create one-shoot timer
     45
          exec1 = 1;
     46
          id1 = osTimerCreate (osTimer(Timer1), osTimerOnce, &exec1);
```



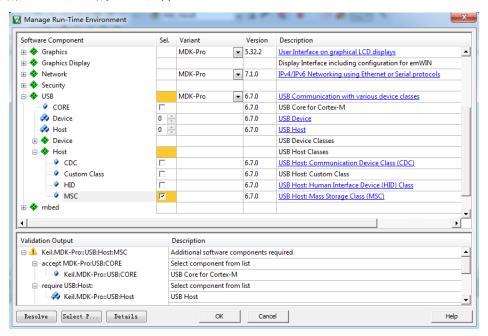
9、运行程序,可以查看到"t1"在按1秒进行步进。



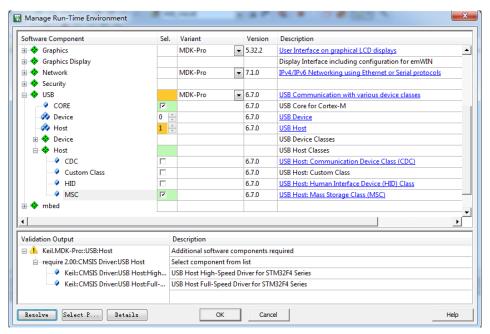


### 三、USB Host 的添加

1、添加 USB Host 的 MSC 组件,

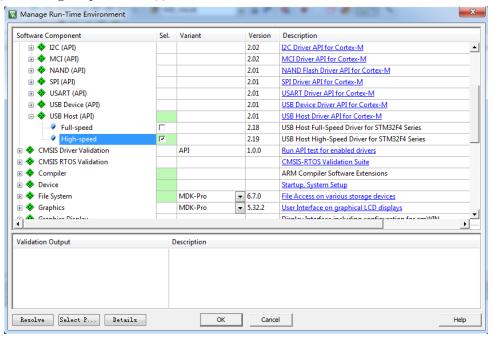


2、点击"Resolve"完成相关组件的添加。

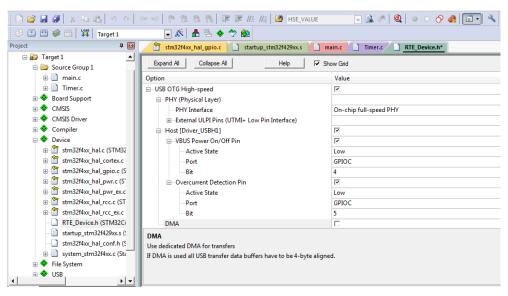




3、添加 "High-speed" 组件。

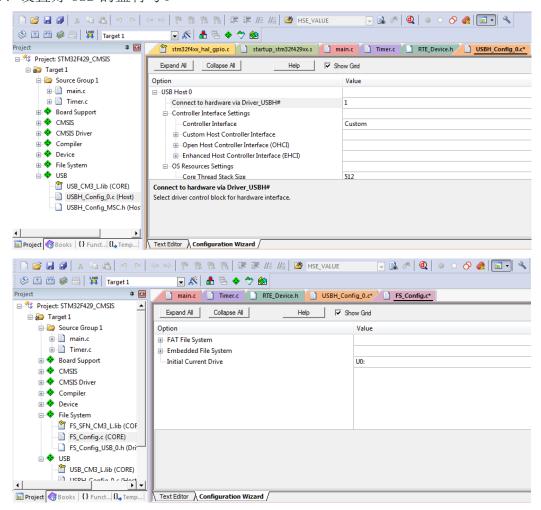


4、设置好 USB 的相关配置。

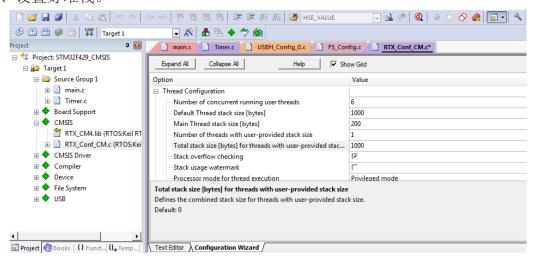




5、设置好 USB 的盘符号。

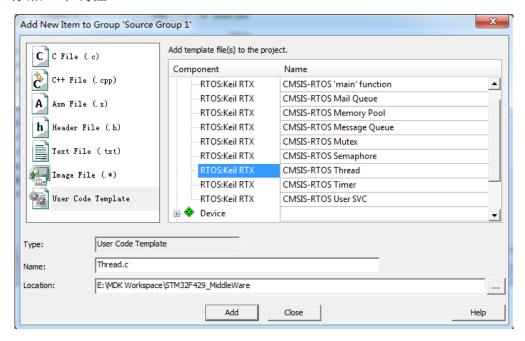


6、设置好堆栈。

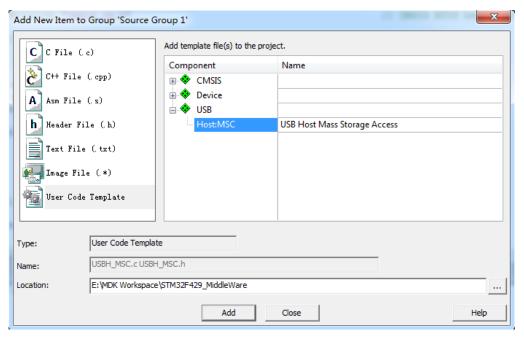




## 7、添加一个线程。



8、添加 USB 存储的处理程序。





9、在"Thread"中添加 USB Host 相关的头文件,设置一个字符数组,并在"Thread"中添加如下代码:

```
static unsigned int result;
           static FILE *f;
           USBH_Initialize (0);
           while (1) {
                result = USBH MSC DriveMount ("U0:");
                if (result == USBH MSC OK) {
                     f = fopen ("Test.txt", "r");
                     if (f) {
                          fread (fbuf, sizeof (fbuf), 1,f);
                          fclose (f);
                osDelay (1000);
main.c Timer.c USBH_Config_0.c Thread.c*
    #include "cmsis os.h"
                                                              // CMSIS RTOS header file
     #include "USBH MSC.h"
                                         // Access storage via USB Host
     char fbuf[2009] = {0};
           Thread 1 'Thread_Name': Sample thread
                                                              // thread function
    void Thread (void const *argument);
                                                              // thread id
 10
    osThreadId tid Thread;
    osThreadDef (Thread, osPriorityNormal, 1, 0);
                                                             // thread object
 13 ⊟int Init Thread (void) {
       tid Thread = osThreadCreate (osThread(Thread), NULL);
      if (!tid_Thread) return(-1);
 17
 18
      return(0);
 19
 21 - void Thread (void const *argument) {
          main.c Timer.c USBH_Config_0.c Thread.c
            19 }
             20
             static unsigned int result;
             24
                  static FILE *f;
             25
             26
                  USBH_Initialize (0);
             27
             28  while (1) {
                    result = USBH_MSC_DriveMount ("U0:");
if (result == USBH_MSC_OK) {
             29
             30 =
                      f = fopen ("Test.txt", "r");
             31
                      if (f) {
             32
                       fread (fbuf, sizeof (fbuf), 1 ,f);
             33
                        fclose (f);
             37
                    osDelay (1000);
             38
                }
             39
             40
```



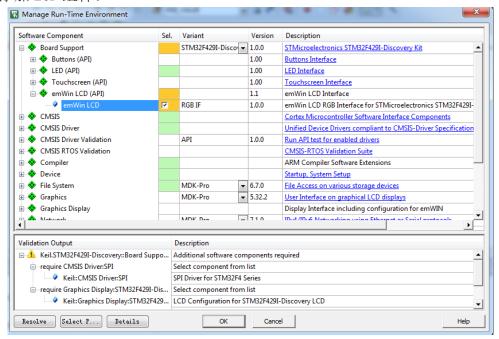
10、在"main"中申明"Init\_Thread"函数,并在"osKernelstart"后调用该函数,编译该工程。

```
main.c* Timer.c USBH_Config_0.c Thread.c
  67
  68
  69 /* Private typedef ------
  70 /* Private define ------*/
  75 static void Error_Handler(void);
  76 extern void Init_Timers (void);
  77 extern void Init_Thread(void);
  78
  79 /* Private functions -----*/
  80 ⊟/**
  * @brief Main program
* @param None
 83 * @retval None
84 */
  85 int main(void)
  86 ⊟ {
main.c* USBH_Config_0.c Thread.c
      Init_Timers();
 105
 106
      LED_Initialize();
 107
108 /* Add your application code here
 109
        */
 110
 111 #ifdef RTE_CMSIS_RTOS
                                  // when using CMSIS RTOS
 112
     // create 'thread' functions that start executing,
     // example: tid name = osThreadCreate (osThread(name), NULL);
 113
 114
 115
      osKernelStart();
                                   // start thread execution
 116 Init Thread();
 117 #endif
 118
 119
     /* Infinite loop */
 120 | while (1)
121 | {
122 -
123 }
 125 🗐 / * *
126 * Ahrief Sustem Clock Configuration
```

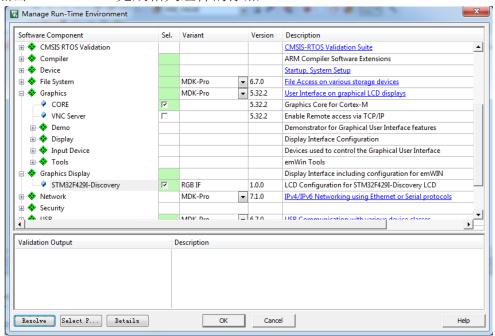


#### 四、图形 GUI 的添加。

1、添加 LCD 组件。

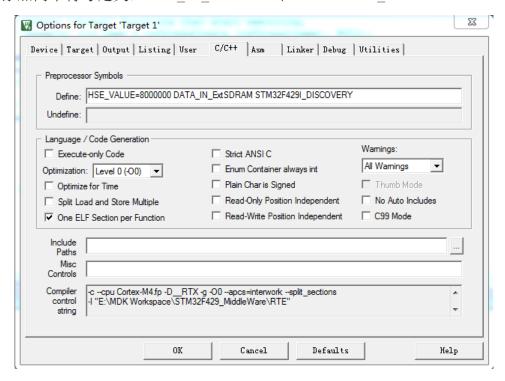


2、点击 "Resolve" 完成相关组件的添加。

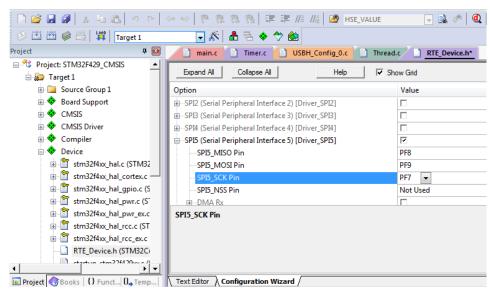




3、添加两个符号定义: DATA\_IN\_ExtSDRAM 和 STM32F429I\_DISCOVERY。

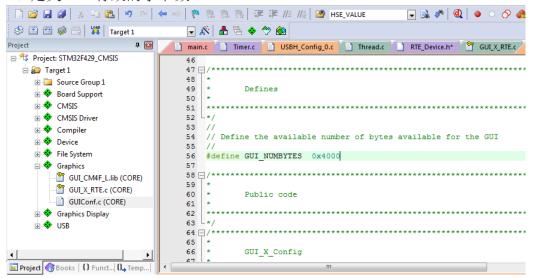


4、根据硬件原理图完成硬件接口的设置。

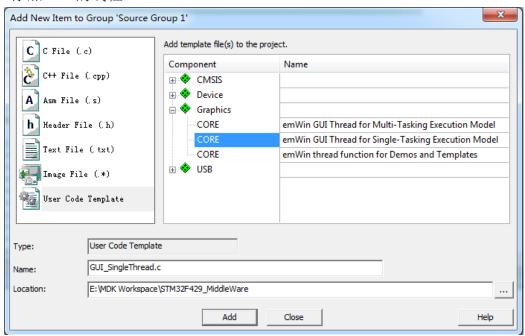




5、定义 GUI 有效的字节数。

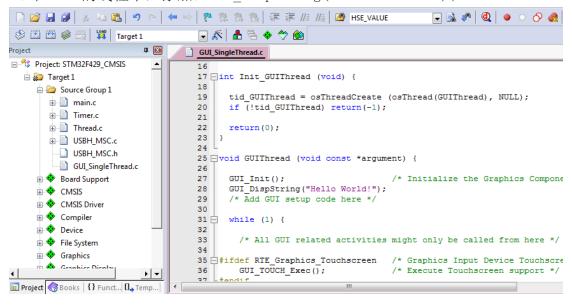


6、添加 GUI 的线程。

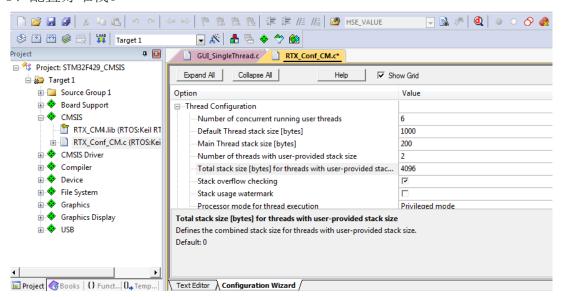




7、在 GUI 的线程中,添加: GUI DispString("Hello World!");

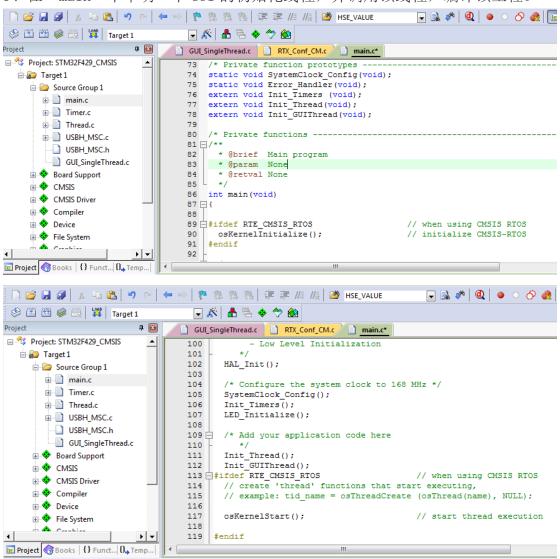


### 8、配置好堆栈。





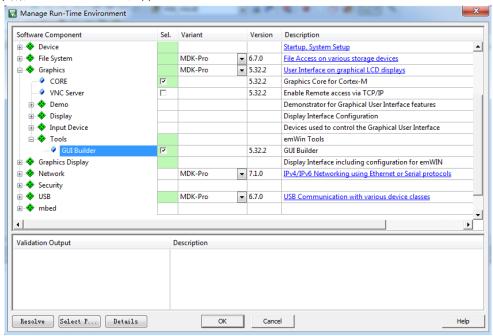
9、在"main"中申明一下GUI的初始化线程,并调用该线程,编译该工程。



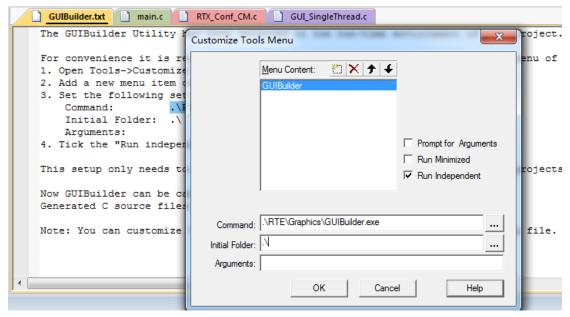


#### 五、创建 GUI。

1、添加 GUI Builder 组件。

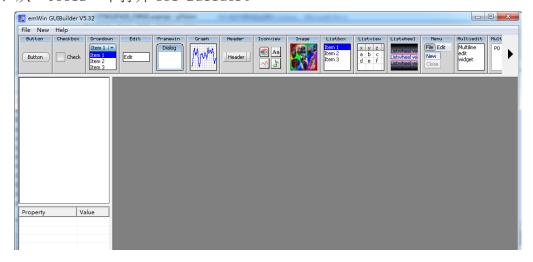


2、在"Tools"中添加 GuiBuilder, 具体配置如下:

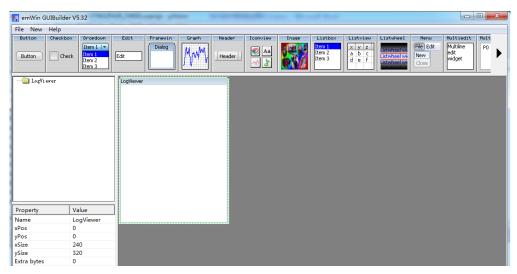




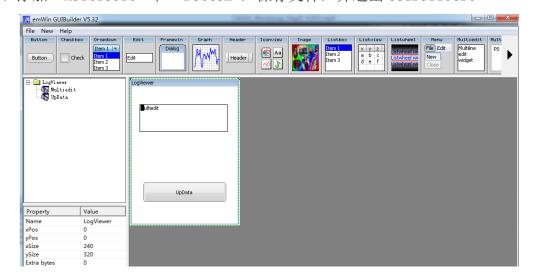
3、从"Tools"中打开GUI Builder。



4、添加 FramWin, 具体配置如下:

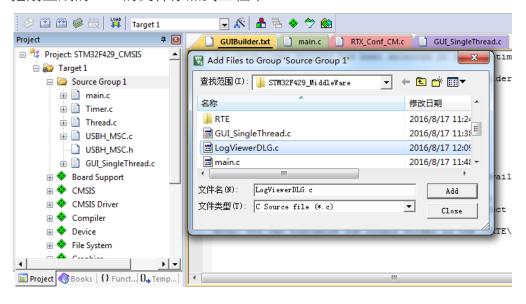


5、添加"Multiedit"和"Bottom",保存文件,并退出GUIBuilder。

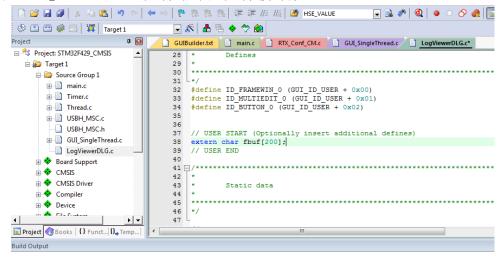




6、把刚生成的 GUI 的文件添加到工程中。



7、在"LogViewerDLG"文件中添加文件缓冲数组。



8、在 GUI 线程文件中添加: #include "dialog.h" 和 extern WM\_HWIN CreateLogViewer(void);

```
GUIBuilder.txt main.c RTX_Conf_CM.c GUI_SingleThread.c* LogViewerDLG.c*
     #include "cmsis_os.h"
                                             // CMSIS RTOS header file
     #include "GUI.h"
     #include "dialog.h"
     extern WM_HWIN CreateLogViewer(void);
  7 = #ifdef _RTE_
     #include "RTE_Components.h"
                                             // Component selection
  8
  9
     #endif
 10
 11 🗐 /*-
 12
      * GUIThread: GUI Thread for Single-Task Execution Model
 15 void GUIThread (void const *argument);
     osThreadId tid GUIThread;
     osThreadDef (GUIThread, osPriorityIdle, 1, 2048); // thread object
 19 int Init_GUIThread (void) {
 20
```



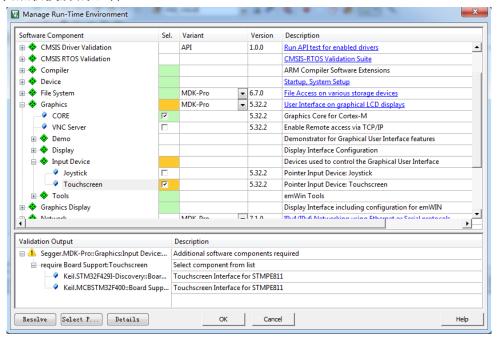
9、在GUI 线程程序中,调用"CreateLogViewer",保存文件并编译工程。

```
GUIBuilder.txt main.c RTX_Conf_CM.c GUI_SingleThread.c* LogViewerDLG.c*
  17 osThreadDef (GUIThread, osPriorityIdle, 1, 2048); // thread object
  18
  19 ☐ int Init GUIThread (void) {
  20
       tid_GUIThread = osThreadCreate (osThread(GUIThread), NULL);
  21
  22
       if (!tid GUIThread) return(-1);
  23
  24
       return(0);
  25 }
  27 - void GUIThread (void const *argument) {
  28
       GUI_Init(); //
GUI_DispString("Hello World!");
                                 /* Initialize the Graphics Component */
  29
  30
  31 CreateLogViewer();
  32
       /* Add GUI setup code here */
  33
  34  while (1) {
  35
        /st All GUI related activities might only be called from here st/
  36
```

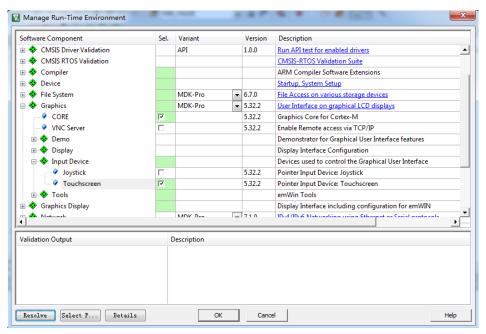


## 六、触摸屏控制的添加。

1、添加触摸屏的组件。

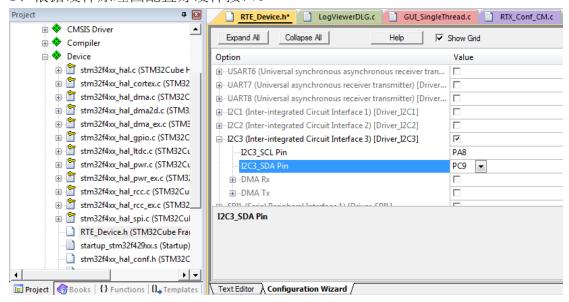


2、添加与触摸屏相关的组件。

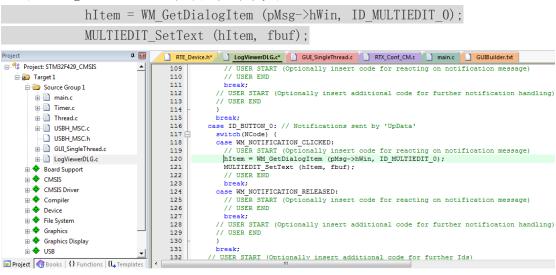




3、根据硬件原理图配置好硬件接口。



4、在"LogViewerDLG"中添加如下代码:



5、保存并编译工程。