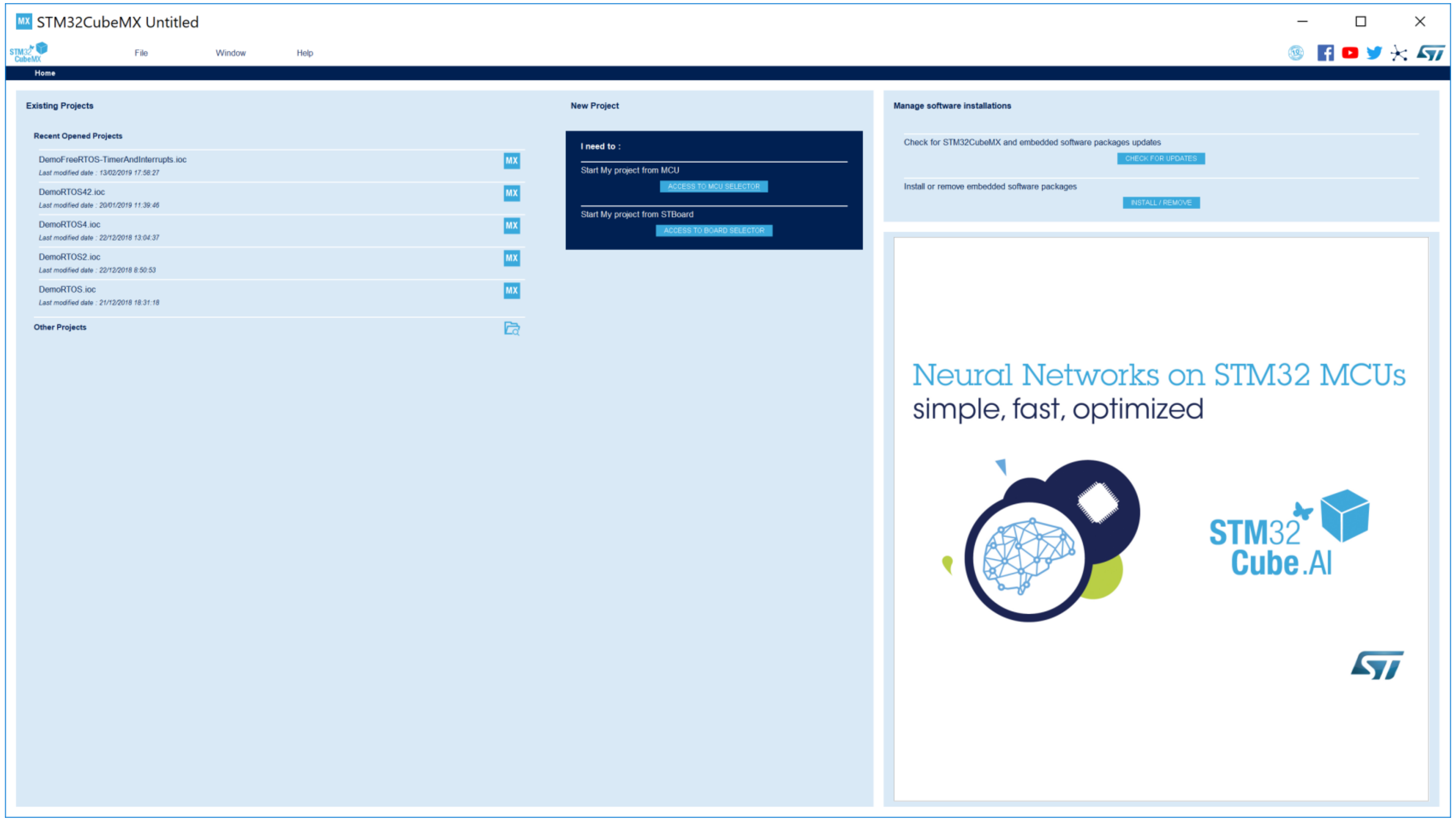
Embedded RTOS Assignment 8 Resource Groups

By
Norman McEntire

Norman.mcentire@gmail.com

Step 1. Startup STM32CubeMX



Existing Projects

Recent Opened Projects

DemoFreeRTOS-TimerAndInterrupts.ioc
Last modified date : 13/02/2019 17:58:27

DemoRTOS42.ioc
Last modified date : 20/01/2019 11:39:46

DemoRTOS4.ioc
Last modified date : 22/12/2018 13:04:37

DemoRTOS2.ioc
Last modified date : 22/12/2018 8:50:53

DemoRTOS.ioc
Last modified date : 21/12/2018 18:31:18

Other Projects

New Project

I need to :

Start My project from MCU
ACCESS TO MCU SELECTOR

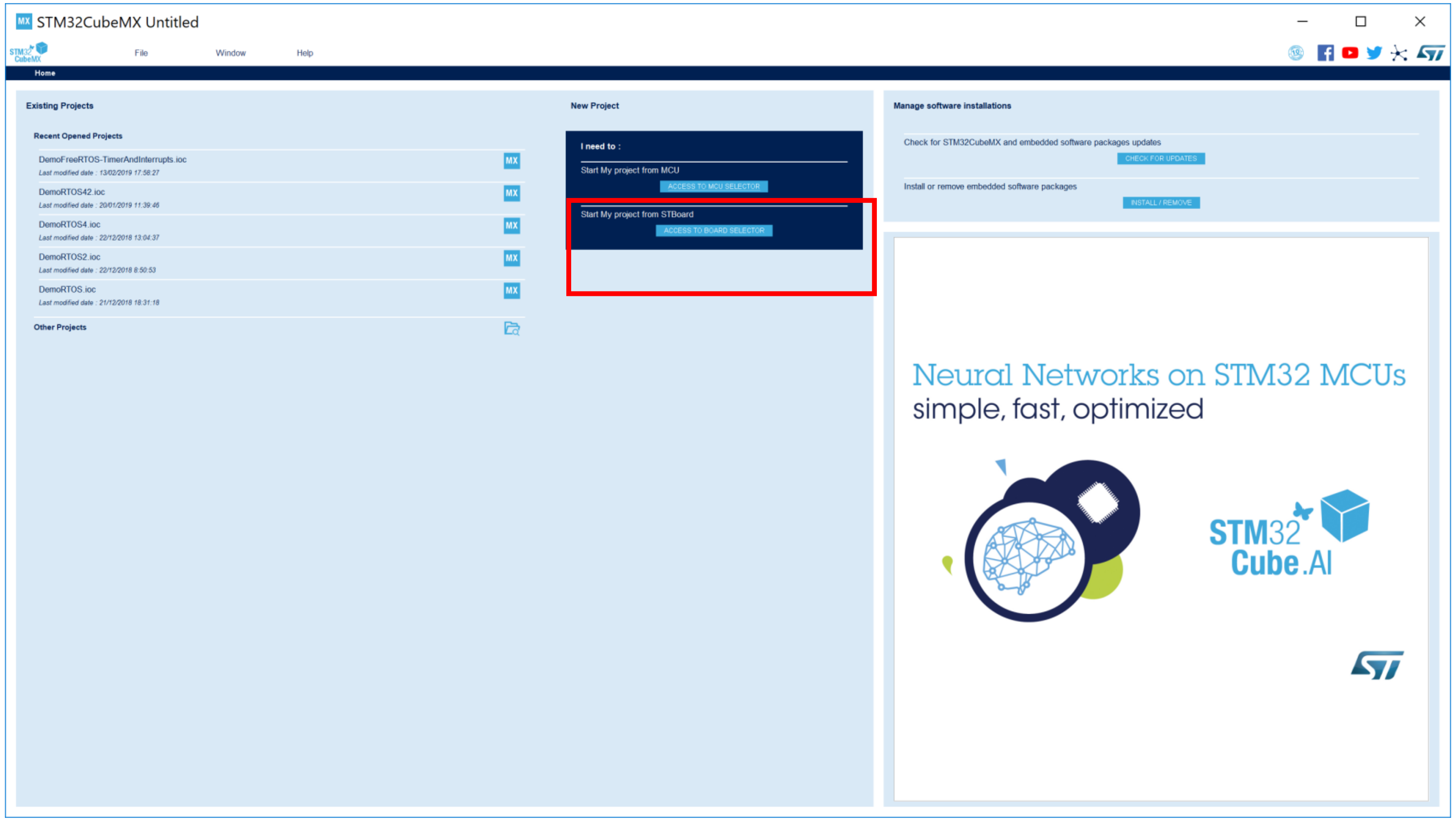
Start My project from STBoard
ACCESS TO BOARD SELECTOR

Manage software installations

Check for STM32CubeMX and embedded software packages updates
CHECK FOR UPDATES

Install or remove embedded software packages
INSTALL / REMOVE

Step 2. Access Board Selector



STM32CubeMX Untitled

Existing Projects

Recent Opened Projects

DemoFreeRTOS-TimerAndInterrupts.ioc
Last modified date : 13/02/2019 17:58:27

DemoRTOS42.ioc
Last modified date : 20/01/2019 11:39:46

DemoRTOS4.ioc
Last modified date : 22/12/2018 13:04:37

DemoRTOS2.ioc
Last modified date : 22/12/2018 8:50:53

DemoRTOS.ioc
Last modified date : 21/12/2018 18:31:18

Other Projects

New Project

I need to :

Start My project from MCU

Start My project from STBoard

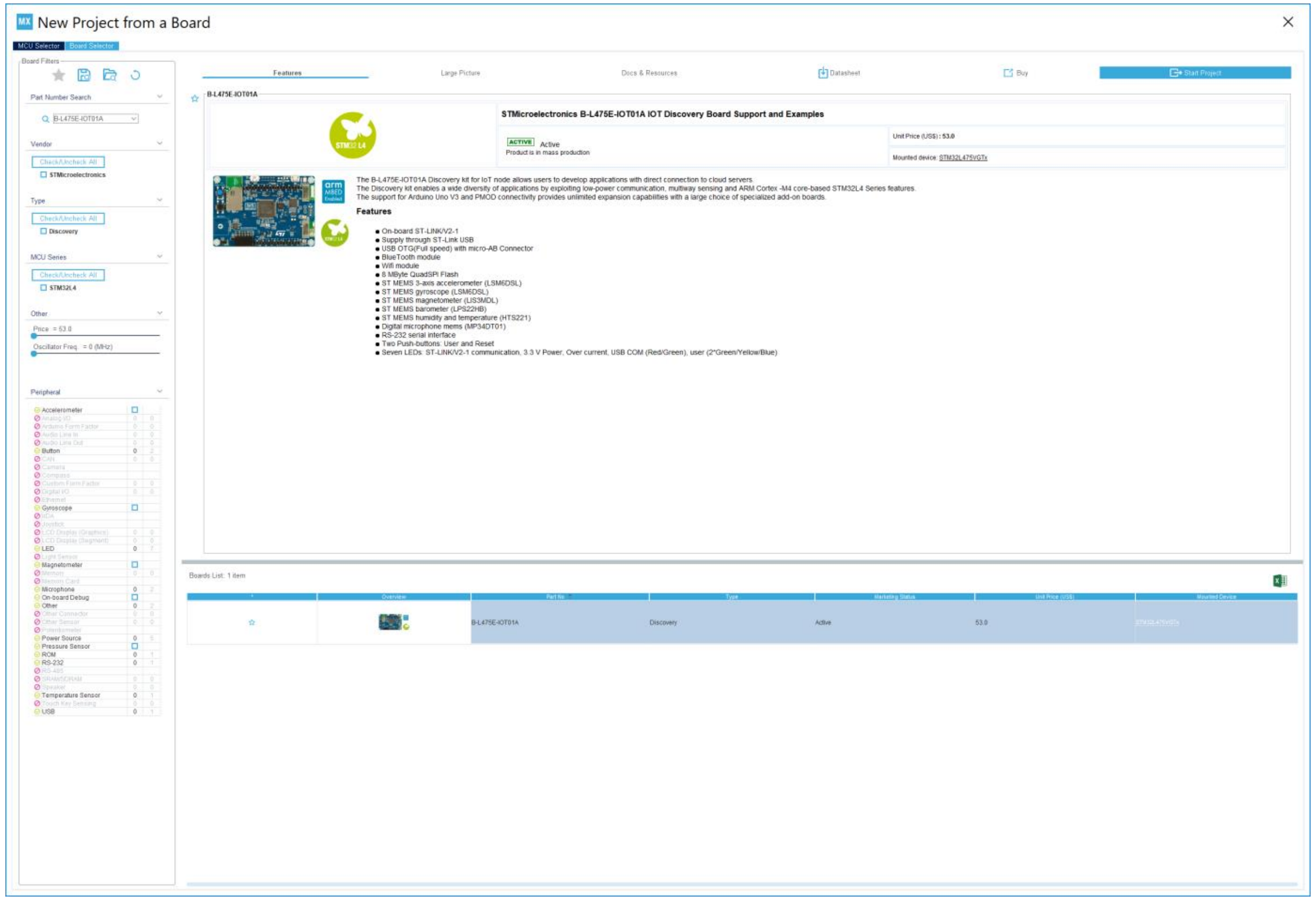
Manage software installations

Check for STM32CubeMX and embedded software packages updates

Install or remove embedded software packages

Neural Networks on STM32 MCUs
simple, fast, optimized

Step 3. Select “B-L475E-IOT01A” Board



Part Number Search: B-L475E-IOT01A

Vendor: STMicroelectronics

Type: Discovery

MCU Series: STM32L4

Other: Price = 53.0; Oscillator Freq. = 0 (MHz)

Peripheral: Accelerometer, Button, Gyroscope, LED, Magnetometer, Microphone, On-board Debug, Other, Power Source, Pressure Sensor, ROM, RS-232, Temperature Sensor, USB

B-L475E-IOT01A

STMicroelectronics B-L475E-IOT01A IOT Discovery Board Support and Examples

ACTIVE Active
Product is in mass production

Unit Price (US$) : 53.0

Mounted device: STM32L475VGTx

The B-L475E-IOT01A Discovery kit for IoT node allows users to develop applications with direct connection to cloud servers.
The Discovery kit enables a wide diversity of applications by exploiting low-power communication, multiway sensing and ARM Cortex -M4 core-based STM32L4 Series features.
The support for Arduino Uno V3 and PMOD connectivity provides unlimited expansion capabilities with a large choice of specialized add-on boards.

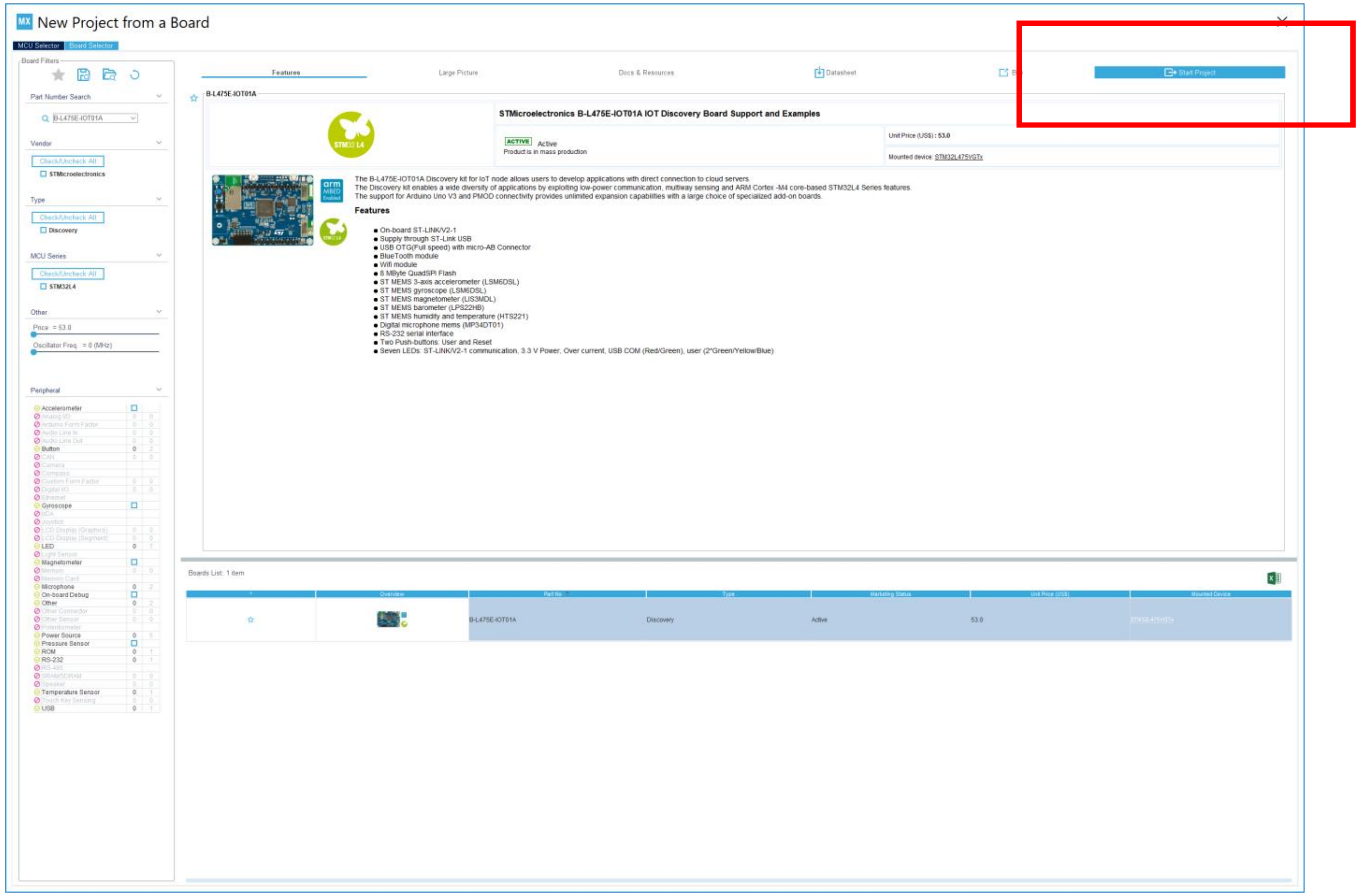
Features

- On-board ST-LINK/V2-1
- Supply through ST-Link USB
- USB OTG(Full speed) with micro-AB Connector
- BlueTooth module
- Wifi module
- 8 MByte QuadSPI Flash
- ST MEMS 3-axis accelerometer (LSM6DSL)
- ST MEMS gyroscope (LSM6DSL)
- ST MEMS magnetometer (LIS3MDL)
- ST MEMS barometer (LPS22HB)
- ST MEMS humidity and temperature (HTS221)
- Digital microphone mems (MP34DT01)
- RS-232 serial interface
- Two Push-buttons: User and Reset
- Seven LEDs: ST-LINK/V2-1 communication, 3.3 V Power, Over current, USB COM (Red/Green), user (2*Green/Yellow/Blue)

Boards List: 1 item

*	Overview	Part No	Type	Marketing Status	Unit Price (US$)	Mounted Device
☆		B-L475E-IOT01A	Discovery	Active	53.0	STM32L475VGTx

Step 4. Select “Start Project”



STMicroelectronics B-L475E-IOT01A IOT Discovery Board Support and Examples

Active
Product is in mass production

Unit Price (US$) : 53.0

Mounted device: STM32L475VGTx

The B-L475E-IOT01A Discovery kit for IoT node allows users to develop applications with direct connection to cloud servers.
The Discovery kit enables a wide diversity of applications by exploiting low-power communication, multiway sensing and ARM Cortex -M4 core-based STM32L4 Series features.
The support for Arduino Uno V3 and PMOD connectivity provides unlimited expansion capabilities with a large choice of specialized add-on boards.

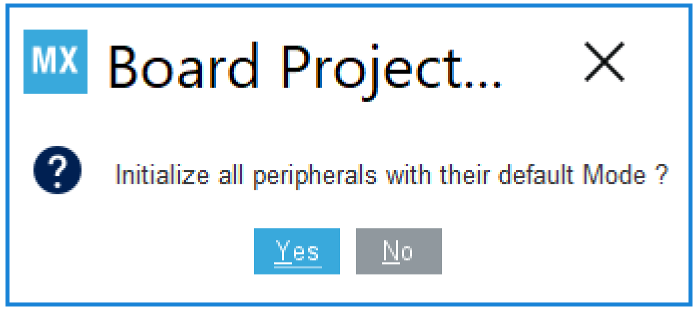
Features

- On-board ST-LINK/V2-1
- Supply through ST-Link USB
- USB OTG(Full speed) with micro-AB Connector
- BlueTooth module
- Wifi module
- 8 MByte QuadSPI Flash
- ST MEMS 3-axis accelerometer (LSM6DSL)
- ST MEMS gyroscope (LSM6DSL)
- ST MEMS magnetometer (LIS3MDL)
- ST MEMS barometer (LPS22HB)
- ST MEMS humidity and temperature (HTS221)
- Digital microphone mems (MP34DT01)
- RS-232 serial interface
- Two Push-buttons: User and Reset
- Seven LEDs: ST-LINK/V2-1 communication, 3.3 V Power, Over current, USB COM (Red/Green), user (2*Green/Yellow/Blue)

Boards List: 1 item

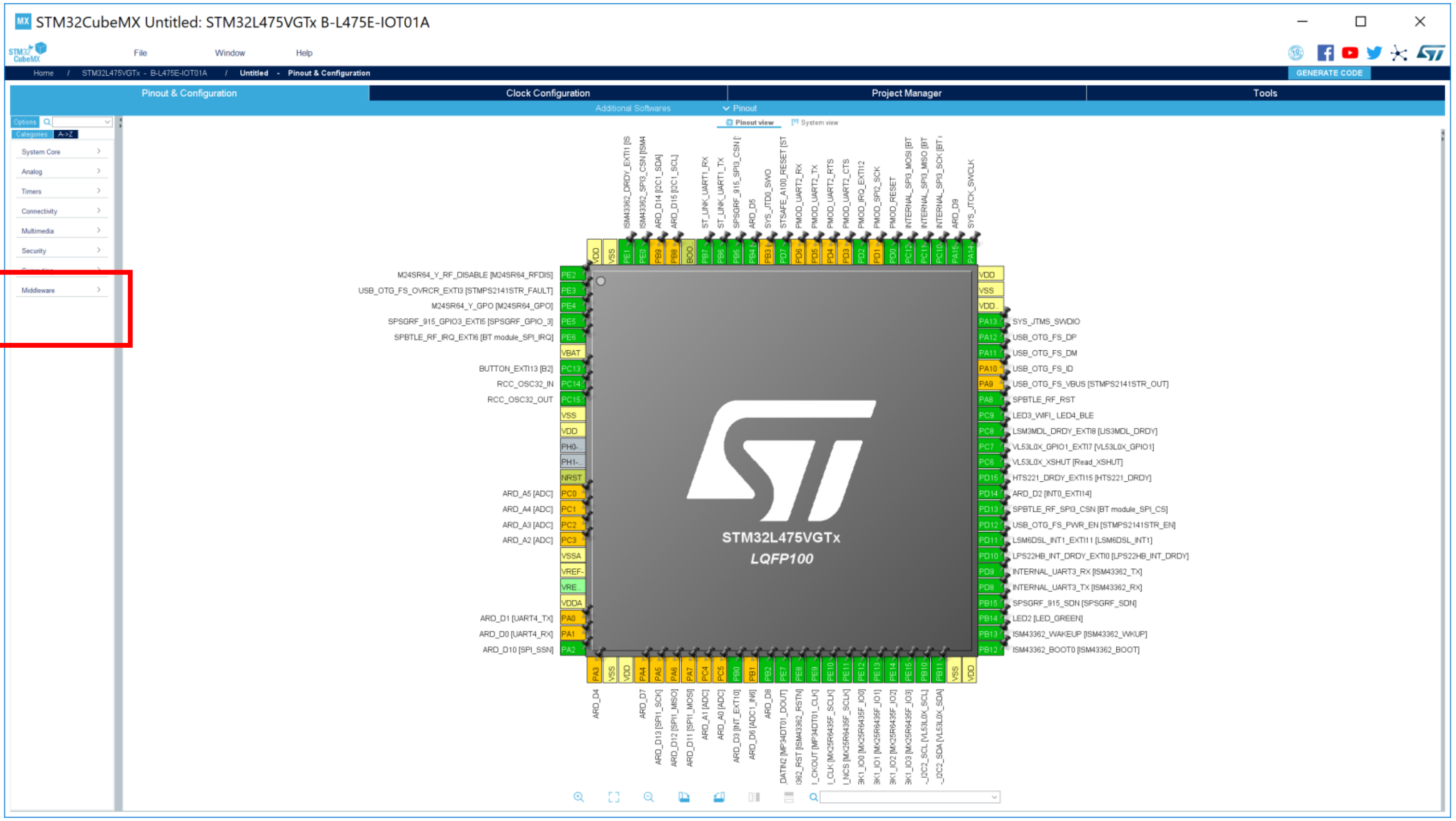
*	Overview	Part No	Type	Marketing Status	Unit Price (US$)	Mounted Device
☆		B-L475E-IOT01A	Discovery	Active	53.0	STM32L475VGTx

Step 5. Select YES: “Initialize all peripherals with their default Mode”

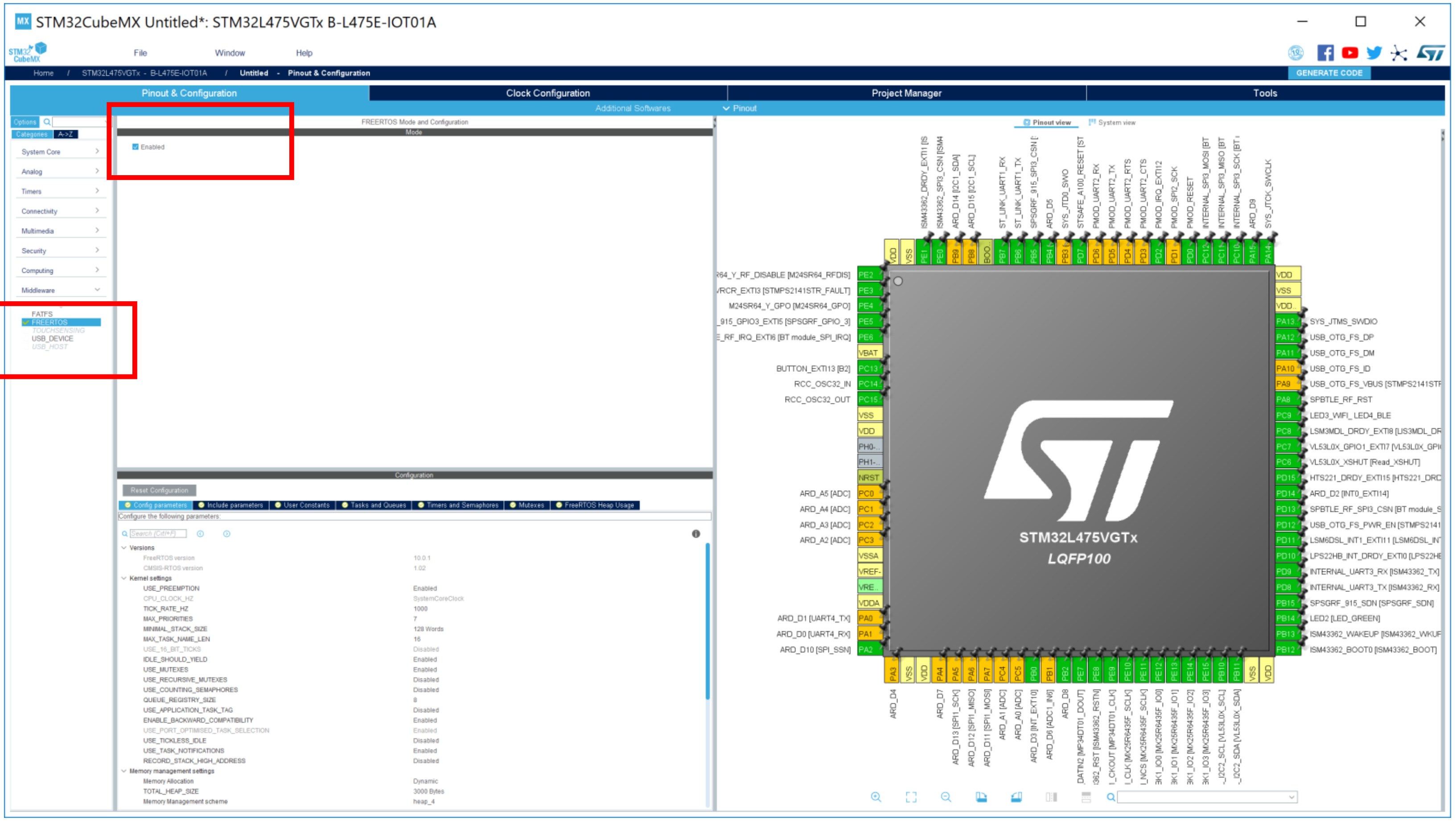


Initialize all peripherals with their default Mode ?

Step 6. Select “Middleware”



Step 7. Select “FreeRTOS”, then select “Enable”

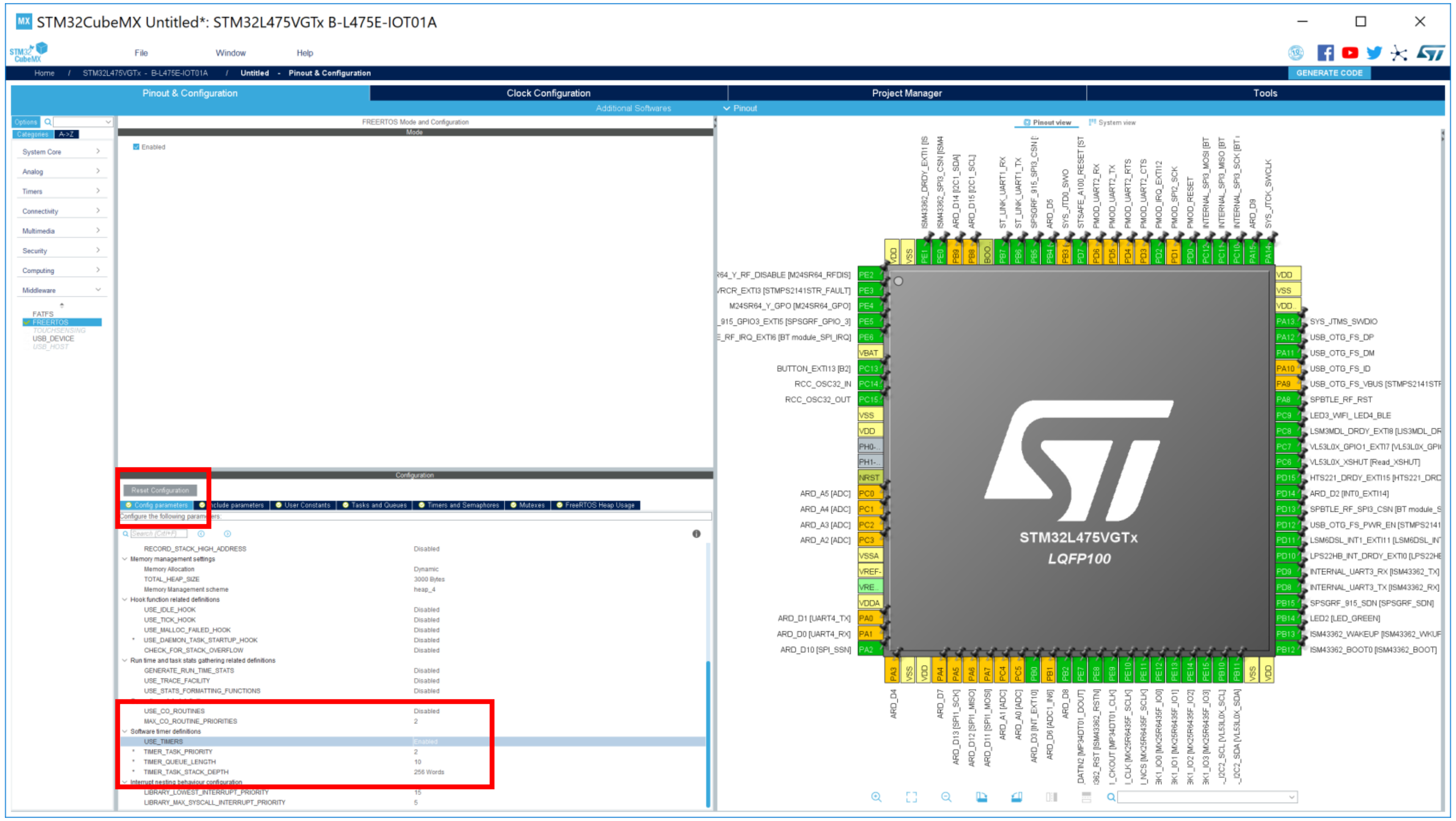


[✓ Enabled]

Configure the following parameters:

Versions	
FreeRTOS version	10.0.1
CMSIS-RTOS version	1.02
Kernel settings	
USE_PREEMPTION	Enabled
CPU_CLOCK_HZ	SystemCoreClock
TICK_RATE_HZ	1000
MAX_PRIORITIES	7
MINIMAL_STACK_SIZE	128 Words
MAX_TASK_NAME_LEN	16
USE_16_BIT_TICKS	Disabled
IDLE_SHOULD_YIELD	Enabled
USE_MUTEXES	Enabled
USE_RECURSIVE_MUTEXES	Disabled
USE_COUNTING_SEMAPHORES	Disabled
QUEUE_REGISTRY_SIZE	8
USE_APPLICATION_TASK_TAG	Disabled
ENABLE_BACKWARD_COMPATIBILITY	Enabled
USE_PORT_OPTIMISED_TASK_SELECTION	Enabled
USE_TICKLESS_IDLE	Disabled
USE_TASK_NOTIFICATIONS	Enabled
RECORD_STACK_HIGH_ADDRESS	Disabled
Memory management settings	
Memory Allocation	Dynamic
TOTAL_HEAP_SIZE	3000 Bytes
Memory Management scheme	heap_4

Step 8. Enable Timers (they are disabled by default)

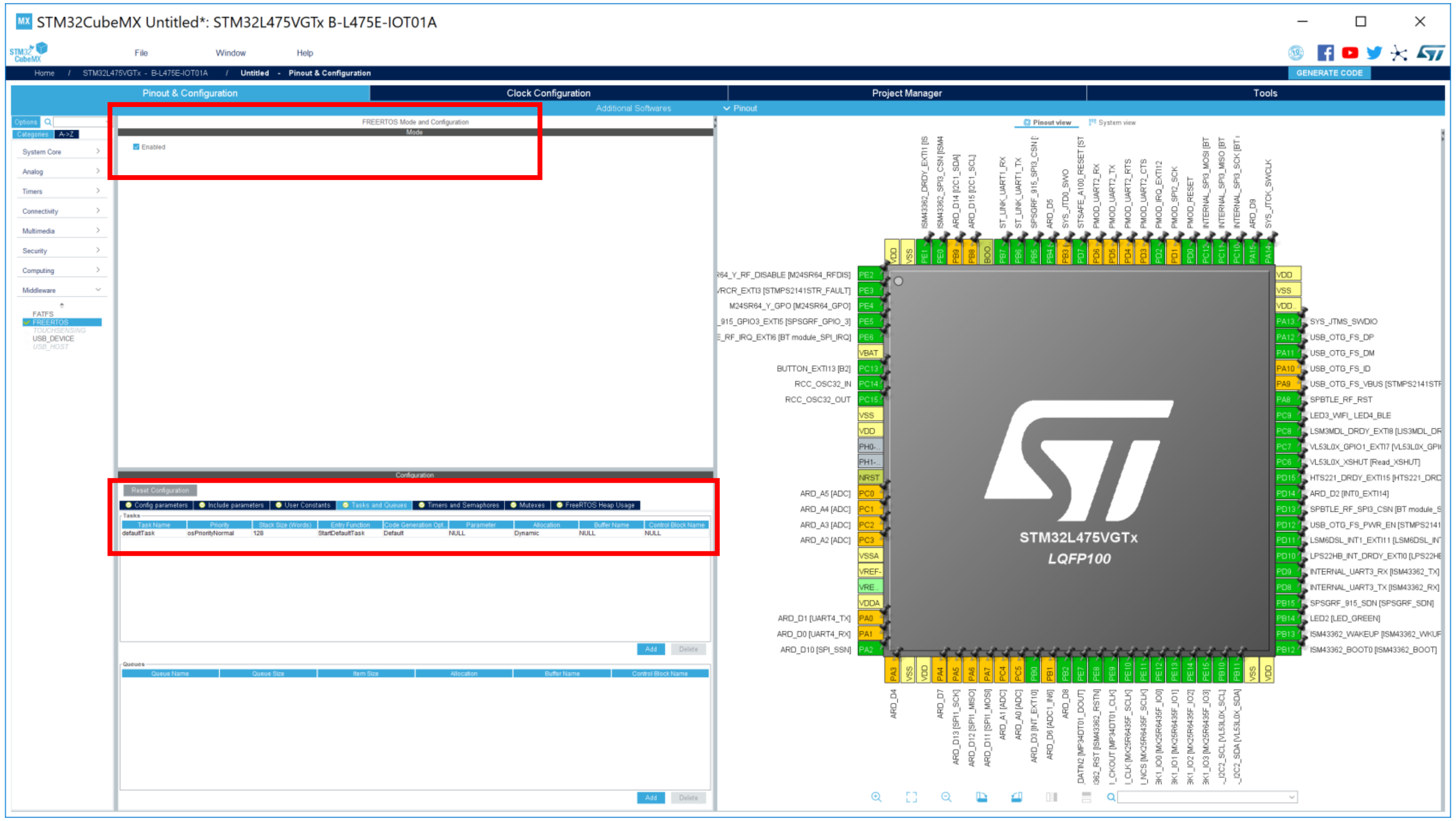


STM32CubeMX Untitled*: STM32L475VGTx B-L475E-IOT01A

Configure the following parameters:

```
    RECORD_STACK_HIGH_ADDRESS                Disabled
Memory management settings
    Memory Allocation                        Dynamic
    TOTAL_HEAP_SIZE                          3000 Bytes
    Memory Management scheme                 heap_4
Hook function related definitions
    USE_IDLE_HOOK                            Disabled
    USE_TICK_HOOK                            Disabled
    USE_MALLOC_FAILED_HOOK                   Disabled
  * USE_DAEMON_TASK_STARTUP_HOOK             Disabled
    CHECK_FOR_STACK_OVERFLOW                 Disabled
Run time and task stats gathering related definitions
    GENERATE_RUN_TIME_STATS                  Disabled
    USE_TRACE_FACILITY                       Disabled
    USE_STATS_FORMATTING_FUNCTIONS           Disabled
    USE_CO_ROUTINES                          Disabled
    MAX_CO_ROUTINE_PRIORITIES                2
Software timer definitions
    USE_TIMERS                               Enabled
  * TIMER_TASK_PRIORITY                      2
  * TIMER_QUEUE_LENGTH                       10
  * TIMER_TASK_STACK_DEPTH                   256 Words
Interrupt nesting behaviour configuration
    LIBRARY_LOWEST_INTERRUPT_PRIORITY        15
    LIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY   5
```

Step 9. Select "Task and Queues", and observe 1 default task created



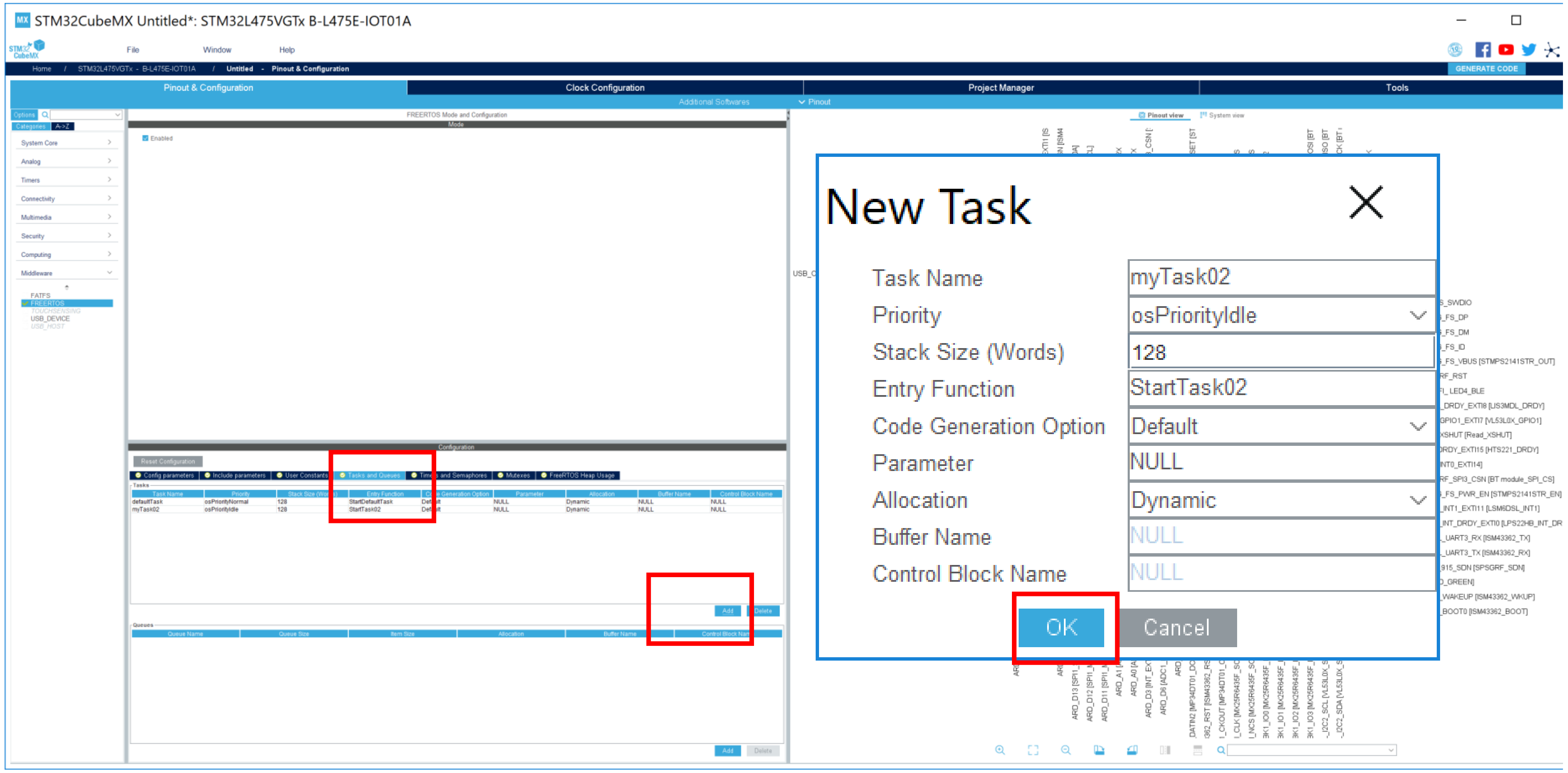
Tasks

Task Name	Priority	Stack Size (Words)	Entry Function	Code Generation Opt...	Parameter	Allocation	Buffer Name	Control Block Name
defaultTask	osPriorityNormal	128	StartDefaultTask	Default	NULL	Dynamic	NULL	NULL

Queues

Queue Name	Queue Size	Item Size	Allocation	Buffer Name	Control Block Name

Step 10. Click Add to add 2nd Task



Tasks

Task Name	Priority	Stack Size (Words)	Entry Function	Code Generation Option	Parameter	Allocation	Buffer Name	Control Block Name
defaultTask	osPriorityNormal	128	StartDefaultTask	Default	NULL	Dynamic	NULL	NULL
myTask02	osPriorityIdle	128	StartTask02	Default	NULL	Dynamic	NULL	NULL

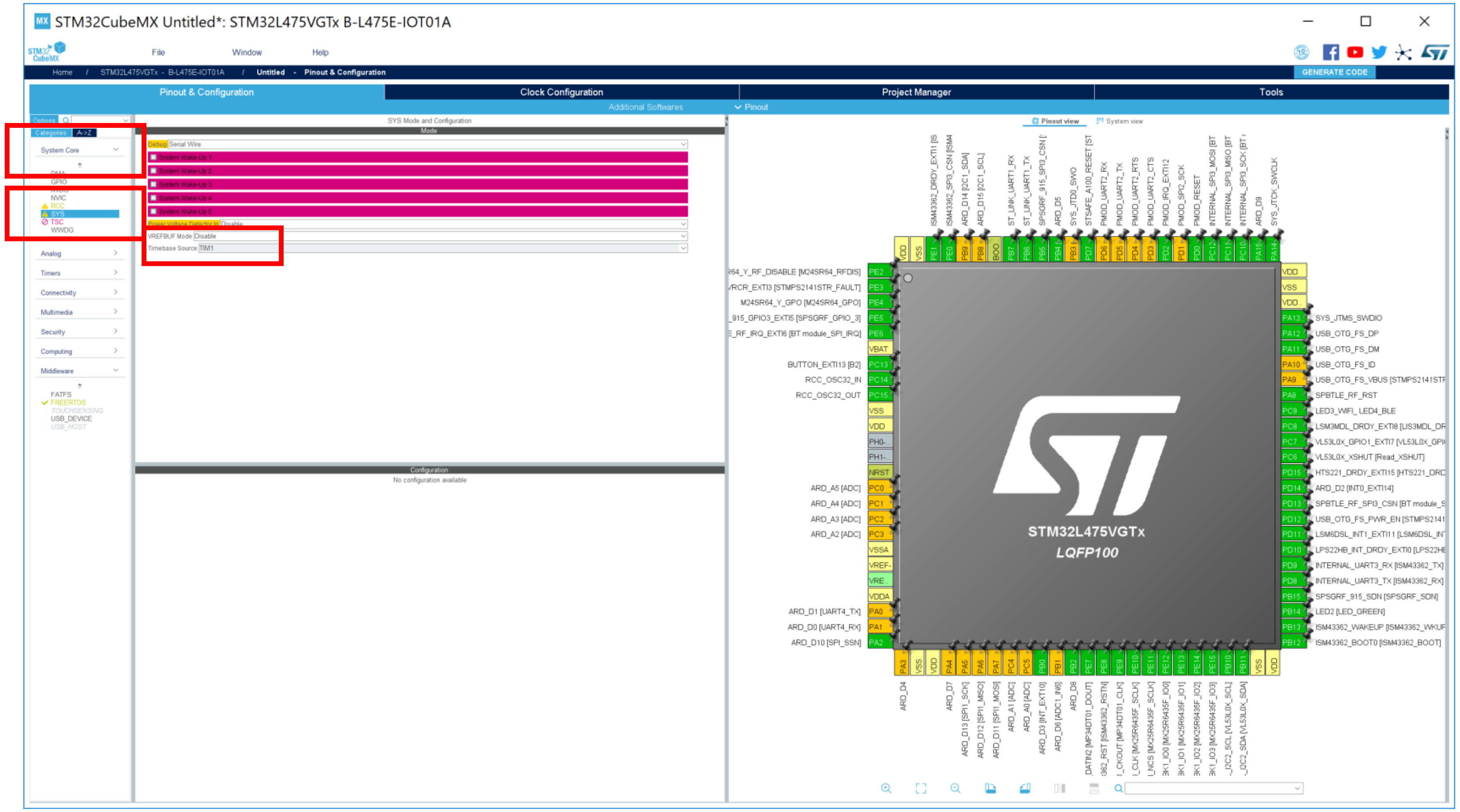
Queues

Queue Name	Queue Size	Item Size	Allocation	Buffer Name	Control Block Name

New Task

Task Name	myTask02
Priority	osPriorityIdle
Stack Size (Words)	128
Entry Function	StartTask02
Code Generation Option	Default
Parameter	NULL
Allocation	Dynamic
Buffer Name	NULL
Control Block Name	NULL

Step 11. Select System Core, Sys, Timebase Source, TIM1



SYS Mode and Configuration

Mode

Debug Serial Wire

System Wake-Up 1

System Wake-Up 2

System Wake-Up 3

System Wake-Up 4

System Wake-Up 5

Power Voltage Detector In Disable

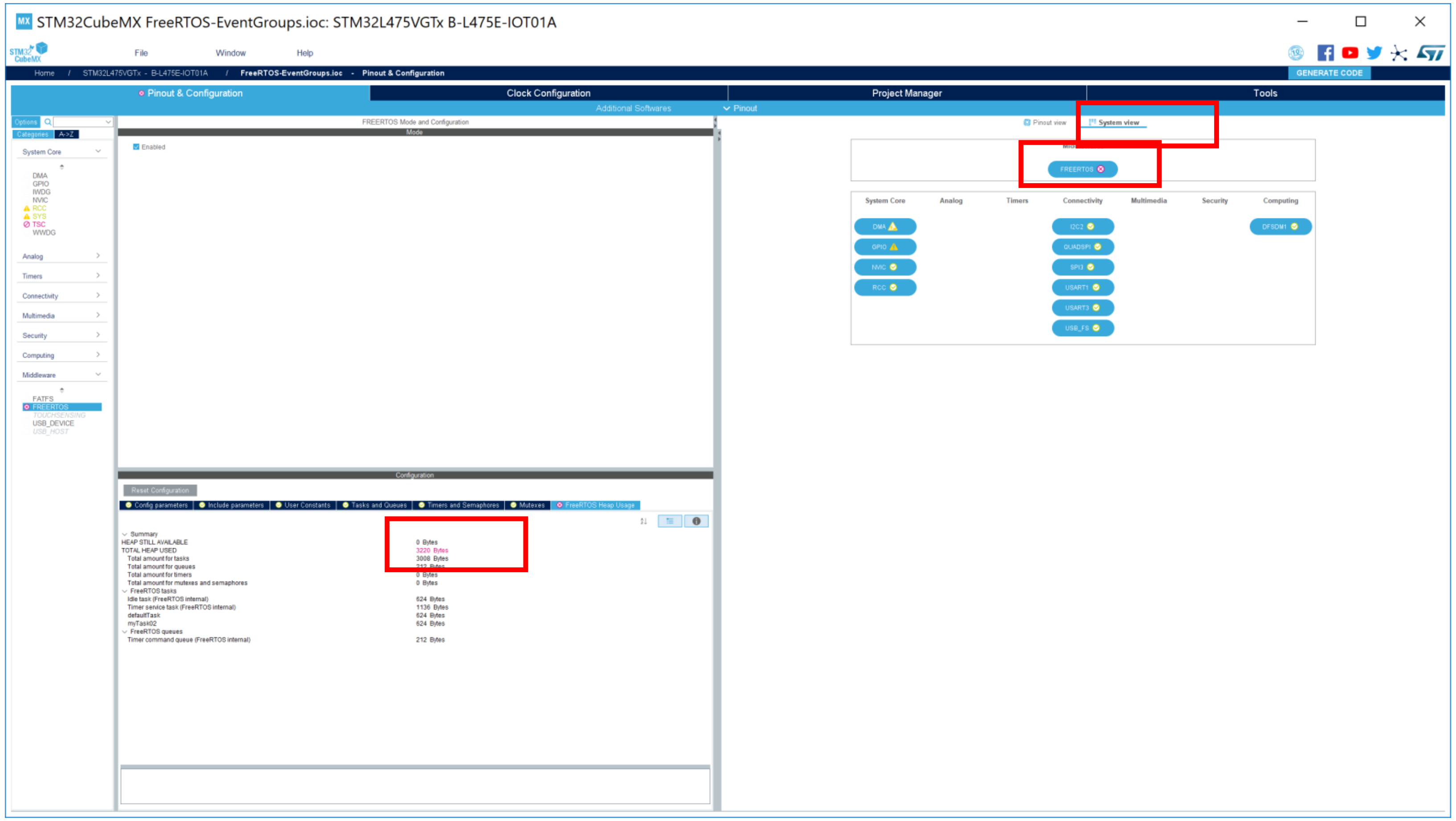
VREFBUF Mode Disable

Timebase Source TIM1

Configuration

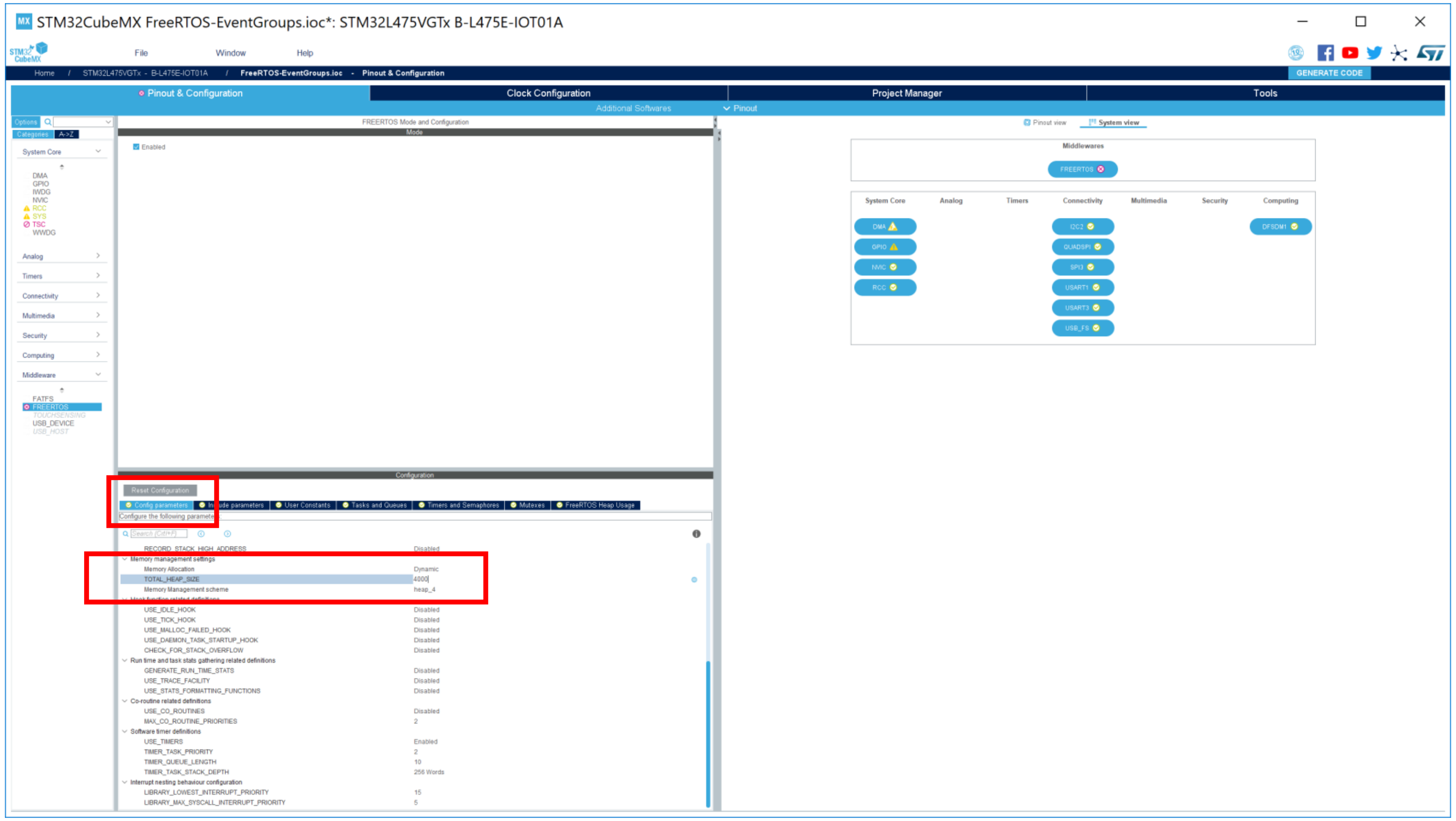
No configuration available

Step 12. Heap usage will be too big. Observe error message.



```
Summary
HEAP STILL AVAILABLE                              0  Bytes
TOTAL HEAP USED                                   3220  Bytes
  Total amount for tasks                          3008  Bytes
  Total amount for queues                         212  Bytes
  Total amount for timers                         0  Bytes
  Total amount for mutexes and semaphores         0  Bytes
FreeRTOS tasks
  Idle task (FreeRTOS internal)                   624  Bytes
  Timer service task (FreeRTOS internal)          1136  Bytes
  defaultTask                                     624  Bytes
  myTask02                                        624  Bytes
FreeRTOS queues
  Timer command queue (FreeRTOS internal)         212  Bytes
```

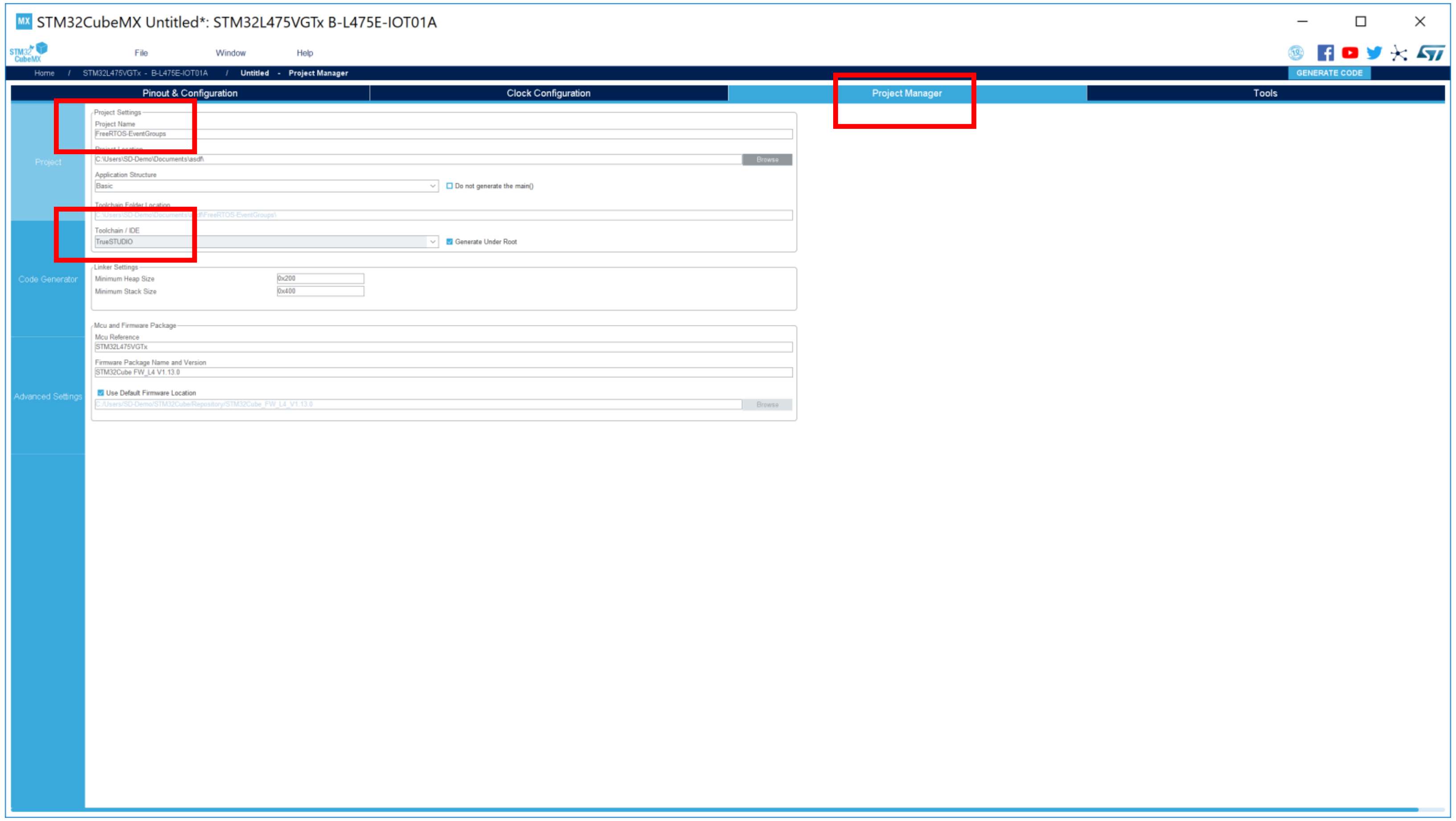
Step 13. Change Heap Size from 3000 to 4000


STM32CubeMX FreeRTOS-EventGroups.ioc*: STM32L475VGTx B-L475E-IOT01A

Configure the following parameters:

```
   RECORD_STACK_HIGH_ADDRESS                 Disabled
v Memory management settings
   Memory Allocation                         Dynamic
   TOTAL_HEAP_SIZE                           4000
   Memory Management scheme                  heap_4
v Hook function related definitions
   USE_IDLE_HOOK                             Disabled
   USE_TICK_HOOK                             Disabled
   USE_MALLOC_FAILED_HOOK                    Disabled
   USE_DAEMON_TASK_STARTUP_HOOK              Disabled
   CHECK_FOR_STACK_OVERFLOW                  Disabled
v Run time and task stats gathering related definitions
   GENERATE_RUN_TIME_STATS                   Disabled
   USE_TRACE_FACILITY                        Disabled
   USE_STATS_FORMATTING_FUNCTIONS            Disabled
v Co-routine related definitions
   USE_CO_ROUTINES                           Disabled
   MAX_CO_ROUTINE_PRIORITIES                 2
v Software timer definitions
   USE_TIMERS                                Enabled
   TIMER_TASK_PRIORITY                       2
   TIMER_QUEUE_LENGTH                        10
   TIMER_TASK_STACK_DEPTH                    256 Words
v Interrupt nesting behaviour configuration
   LIBRARY_LOWEST_INTERRUPT_PRIORITY         15
   LIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY    5
```

Step 14. Enter Project Name: “FreeRTOS-EventGroups” and Toolchain/IDE: TrueStudio



Project Settings

Project Name: FreeRTOS-EventGroups

Project Location: C:\Users\SD-Demo\Documents\asdf\

Application Structure: Basic ☐ Do not generate the main()

Toolchain Folder Location: C:\Users\SD-Demo\Documents\asdf\FreeRTOS-EventGroups\

Toolchain / IDE: TrueSTUDIO ✓ Generate Under Root

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

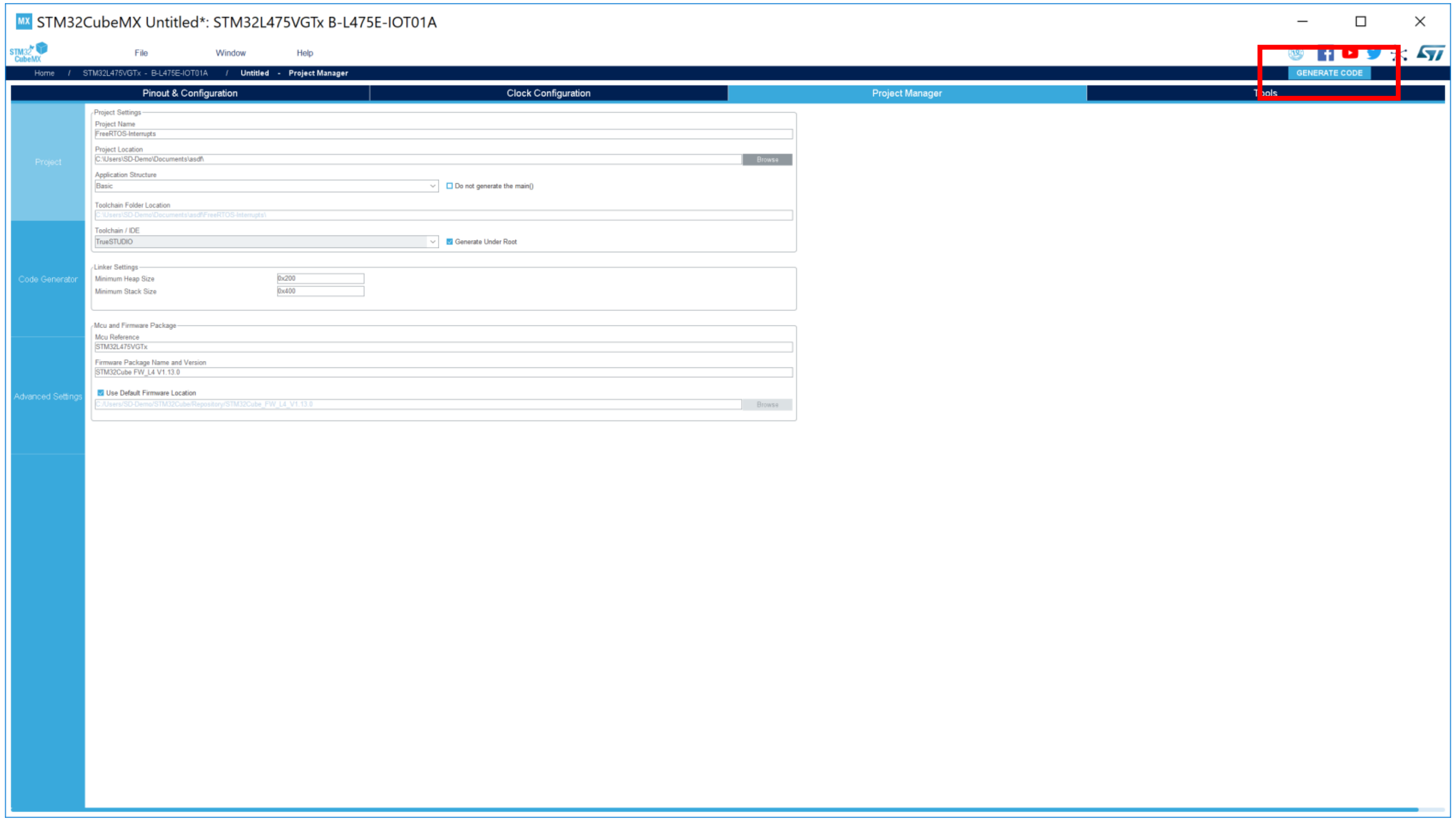
Mcu Reference: STM32L475VGTx

Firmware Package Name and Version: STM32Cube FW_L4 V1.13.0

✓ Use Default Firmware Location

C:/Users/SD-Demo/STM32Cube/Repository/STM32Cube_FW_L4_V1.13.0

Step 15. Select “Generate Code”

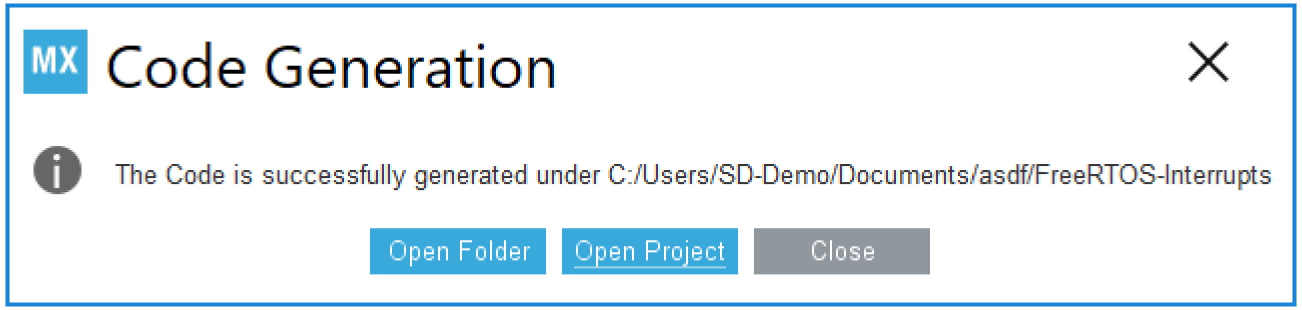


Project Settings
Project Name: FreeRTOS-Interrupts
Project Location: C:\Users\SD-Demo\Documents\asdf\
Application Structure: Basic
☐ Do not generate the main()
Toolchain Folder Location: C:\Users\SD-Demo\Documents\asdf\FreeRTOS-Interrupts\
Toolchain / IDE: TrueSTUDIO
☑ Generate Under Root

Linker Settings
Minimum Heap Size: 0x200
Minimum Stack Size: 0x400

Mcu and Firmware Package
Mcu Reference: STM32L475VGTx
Firmware Package Name and Version: STM32Cube FW_L4 V1.13.0
☑ Use Default Firmware Location
C:/Users/SD-Demo/STM32Cube/Repository/STM32Cube_FW_L4_V1.13.0

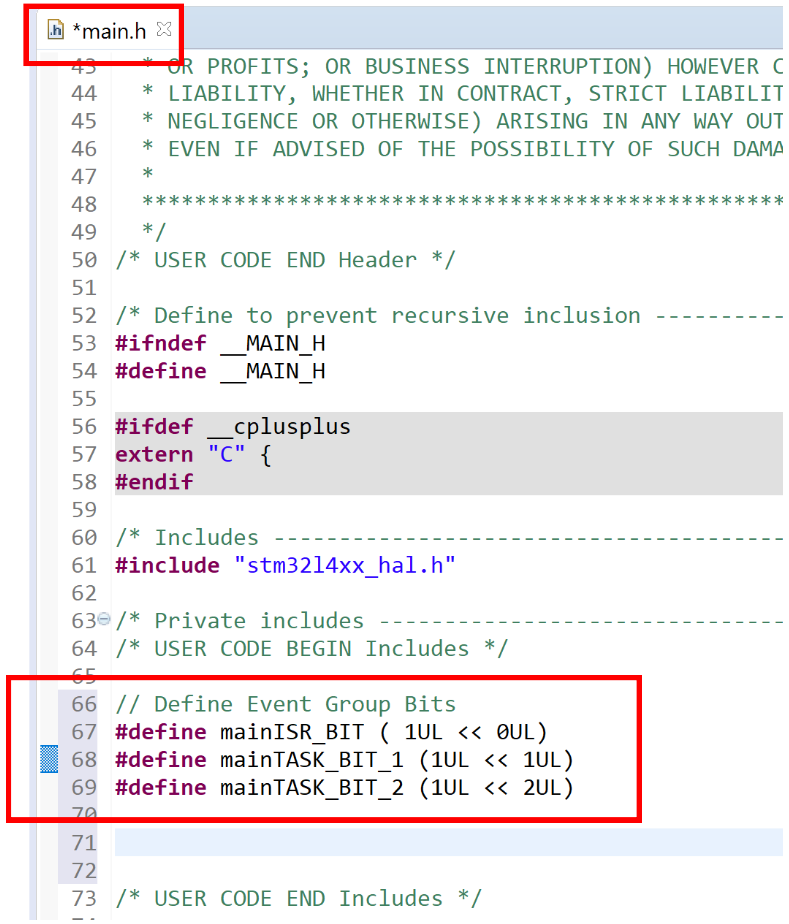
Step 16. Select “Open Project”



Code Generation

The Code is successfully generated under C:/Users/SD-Demo/Documents/asdf/FreeRTOS-Interrupts

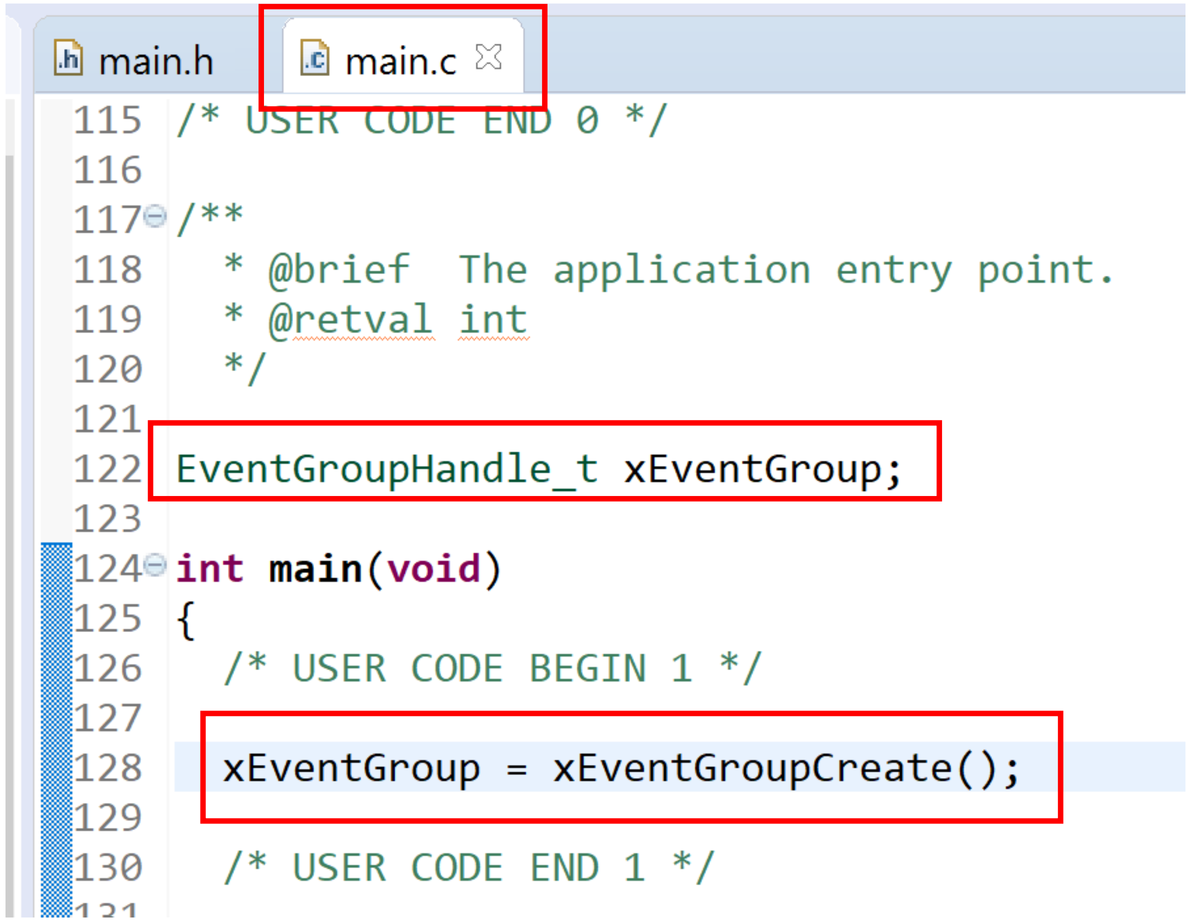
Step 17. Edit main.h to add bit Event Group bit definitions



*main.h

```
  * OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER C
  * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILIT
  * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
  * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMA
  *
  **************************************************
  */
/* USER CODE END Header */

/* Define to prevent recursive inclusion ----------
#ifndef __MAIN_H
#define __MAIN_H

#ifdef __cplusplus
extern "C" {
#endif

/* Includes ----------------------------------------
#include "stm32l4xx_hal.h"

/* Private includes --------------------------------
/* USER CODE BEGIN Includes */

// Define Event Group Bits
#define mainISR_BIT ( 1UL << 0UL)
#define mainTASK_BIT_1 (1UL << 1UL)
#define mainTASK_BIT_2 (1UL << 2UL)



/* USER CODE END Includes */
```

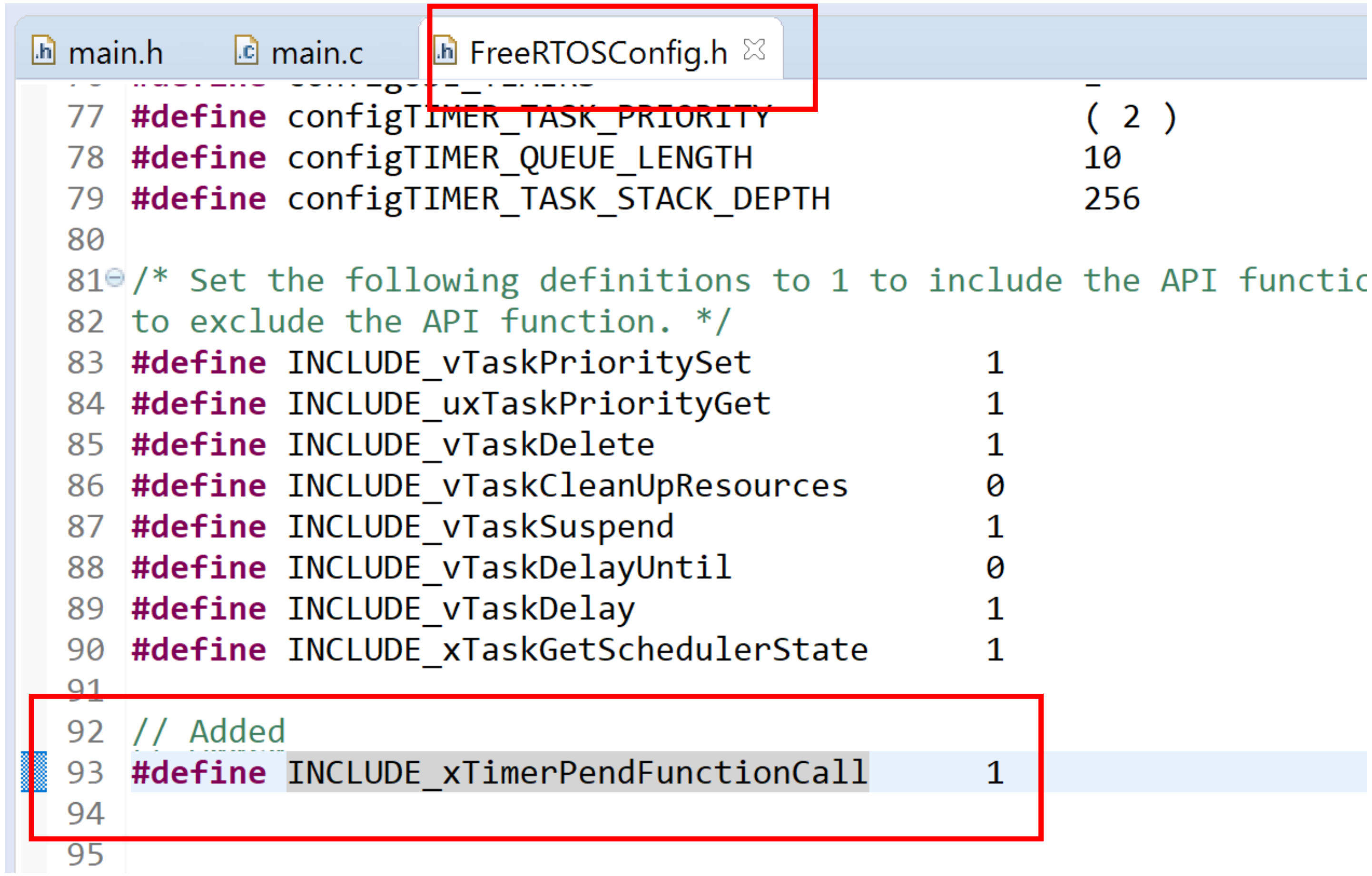
Step 18. Create myEventGroup in main.c



main.h
main.c

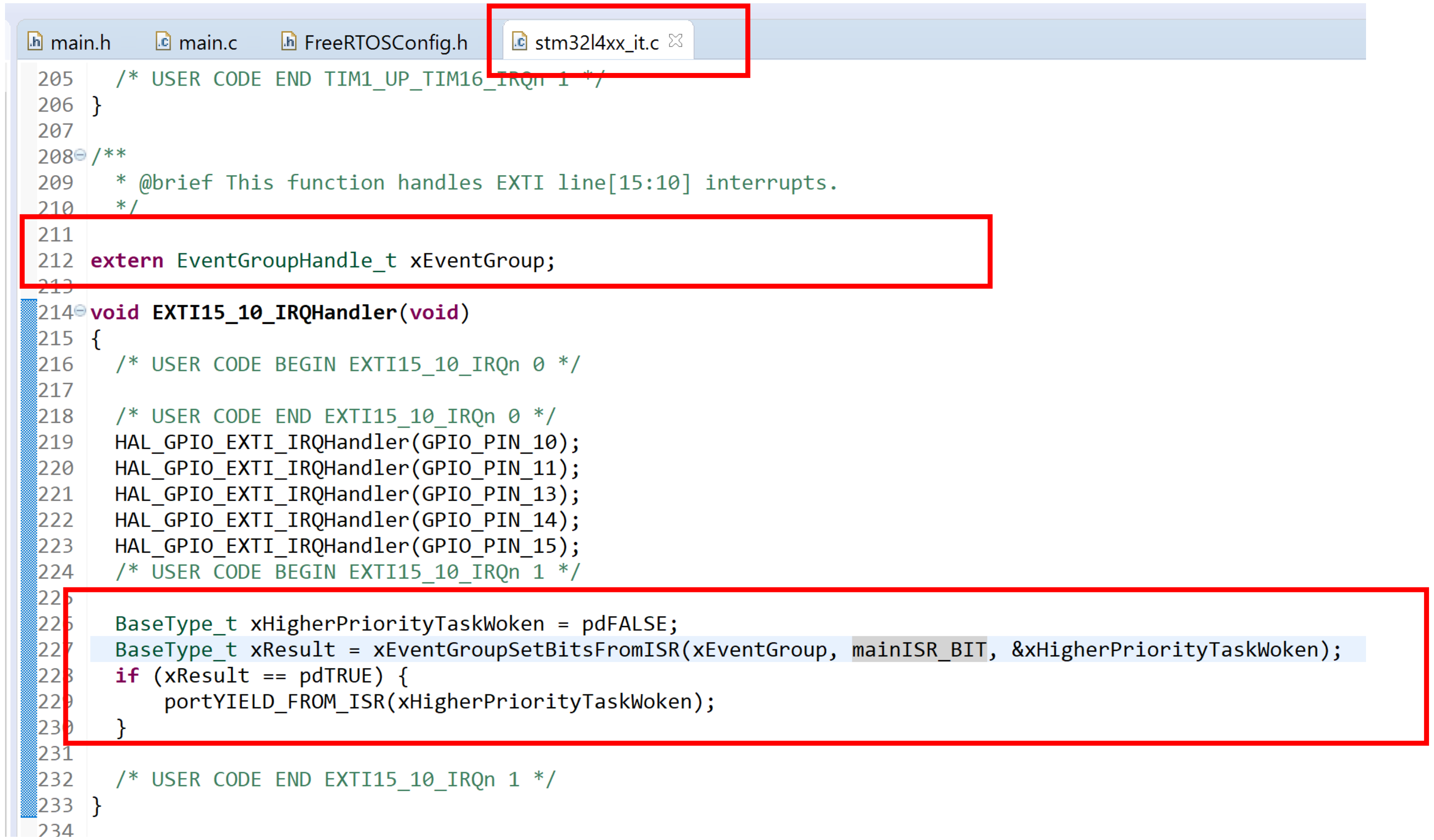
```
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */

EventGroupHandle_t xEventGroup;

int main(void)
{
  /* USER CODE BEGIN 1 */

  xEventGroup = xEventGroupCreate();

  /* USER CODE END 1 */
```

Step 19. Add configuration option to FreeRTOSConfig.h



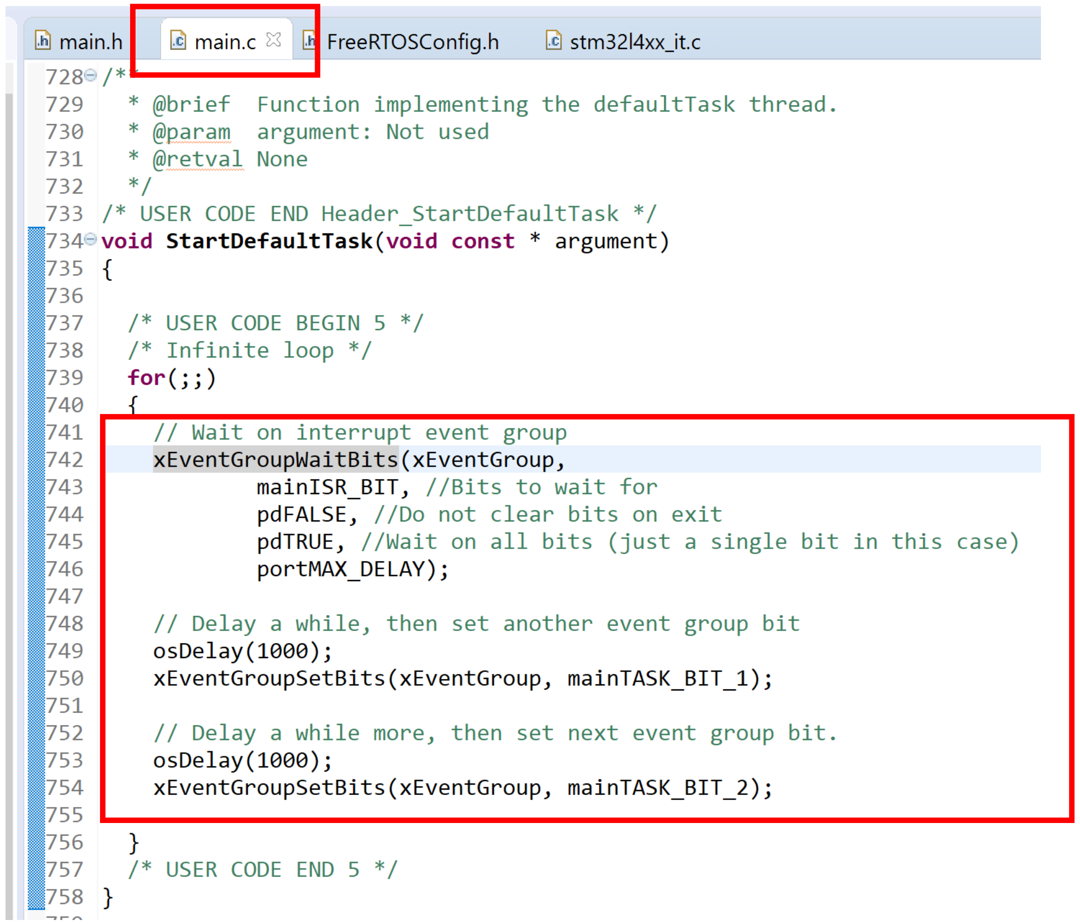
```
#define configTIMER_TASK_PRIORITY              ( 2 )
#define configTIMER_QUEUE_LENGTH               10
#define configTIMER_TASK_STACK_DEPTH           256

/* Set the following definitions to 1 to include the API functic
to exclude the API function. */
#define INCLUDE_vTaskPrioritySet           1
#define INCLUDE_uxTaskPriorityGet          1
#define INCLUDE_vTaskDelete                1
#define INCLUDE_vTaskCleanUpResources      0
#define INCLUDE_vTaskSuspend               1
#define INCLUDE_vTaskDelayUntil            0
#define INCLUDE_vTaskDelay                 1
#define INCLUDE_xTaskGetSchedulerState     1

// Added
#define INCLUDE_xTimerPendFunctionCall     1

```

Step 20. Add interrupt handling code.


main.h main.c FreeRTOSConfig.h stm32l4xx_it.c

```
  /* USER CODE END TIM1_UP_TIM16_IRQn 1 */
}

/**
  * @brief This function handles EXTI line[15:10] interrupts.
  */

extern EventGroupHandle_t xEventGroup;

void EXTI15_10_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI15_10_IRQn 0 */

  /* USER CODE END EXTI15_10_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_10);
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_11);
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_13);
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_14);
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_15);
  /* USER CODE BEGIN EXTI15_10_IRQn 1 */

  BaseType_t xHigherPriorityTaskWoken = pdFALSE;
  BaseType_t xResult = xEventGroupSetBitsFromISR(xEventGroup, mainISR_BIT, &xHigherPriorityTaskWoken);
  if (xResult == pdTRUE) {
      portYIELD_FROM_ISR(xHigherPriorityTaskWoken);
  }

  /* USER CODE END EXTI15_10_IRQn 1 */
}
```

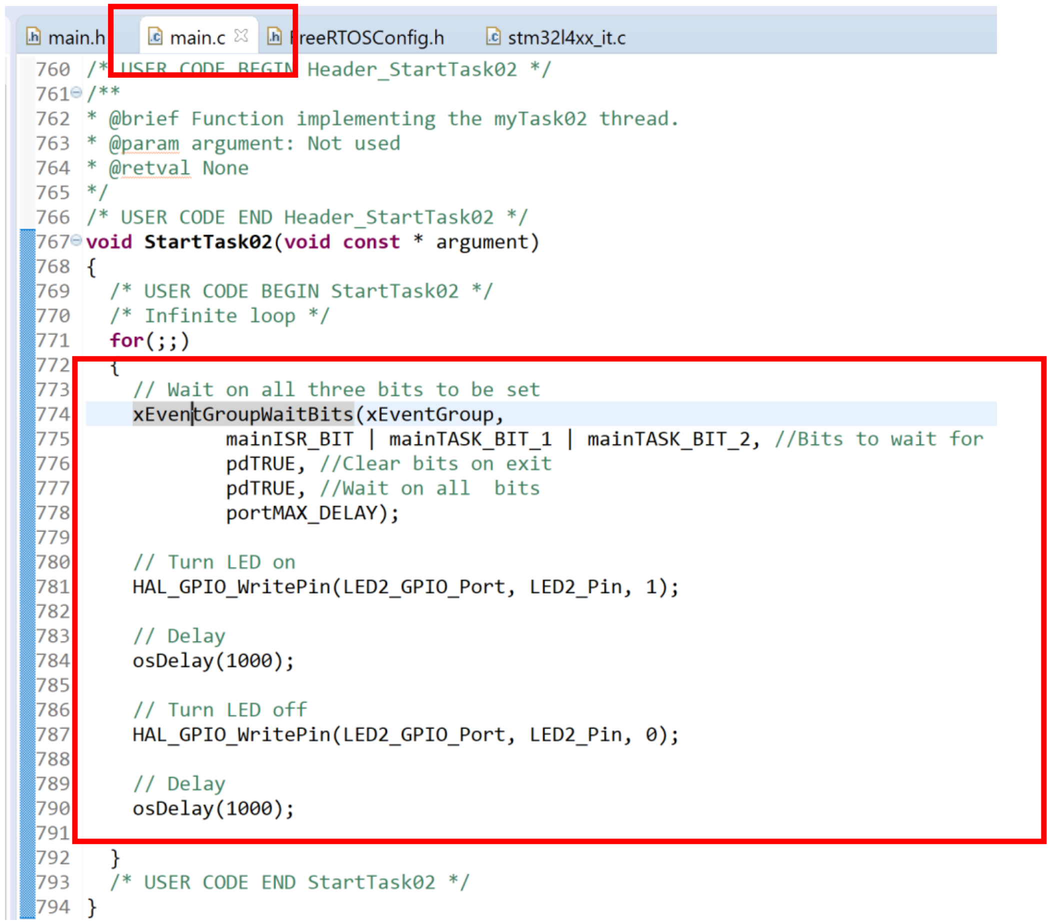
Step 21. Add Task 1 Code.



main.h main.c FreeRTOSConfig.h stm32l4xx_it.c

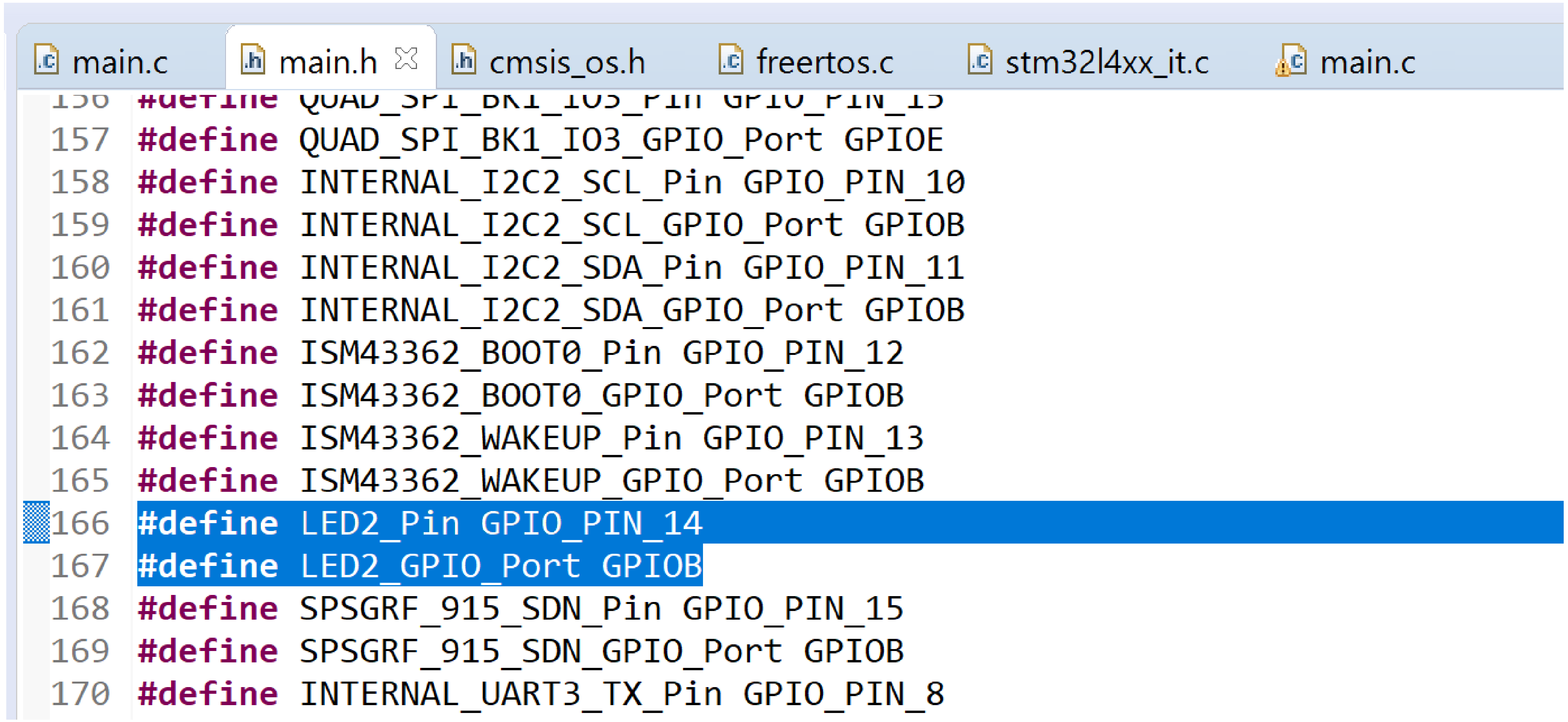
```
/**
  * @brief  Function implementing the defaultTask thread.
  * @param  argument: Not used
  * @retval None
  */
/* USER CODE END Header_StartDefaultTask */
void StartDefaultTask(void const * argument)
{

  /* USER CODE BEGIN 5 */
  /* Infinite loop */
  for(;;)
  {
    // Wait on interrupt event group
    xEventGroupWaitBits(xEventGroup,
            mainISR_BIT, //Bits to wait for
            pdFALSE, //Do not clear bits on exit
            pdTRUE, //Wait on all bits (just a single bit in this case)
            portMAX_DELAY);

    // Delay a while, then set another event group bit
    osDelay(1000);
    xEventGroupSetBits(xEventGroup, mainTASK_BIT_1);

    // Delay a while more, then set next event group bit.
    osDelay(1000);
    xEventGroupSetBits(xEventGroup, mainTASK_BIT_2);

  }
  /* USER CODE END 5 */
}
```

Step 22. Add Task 2 Code.



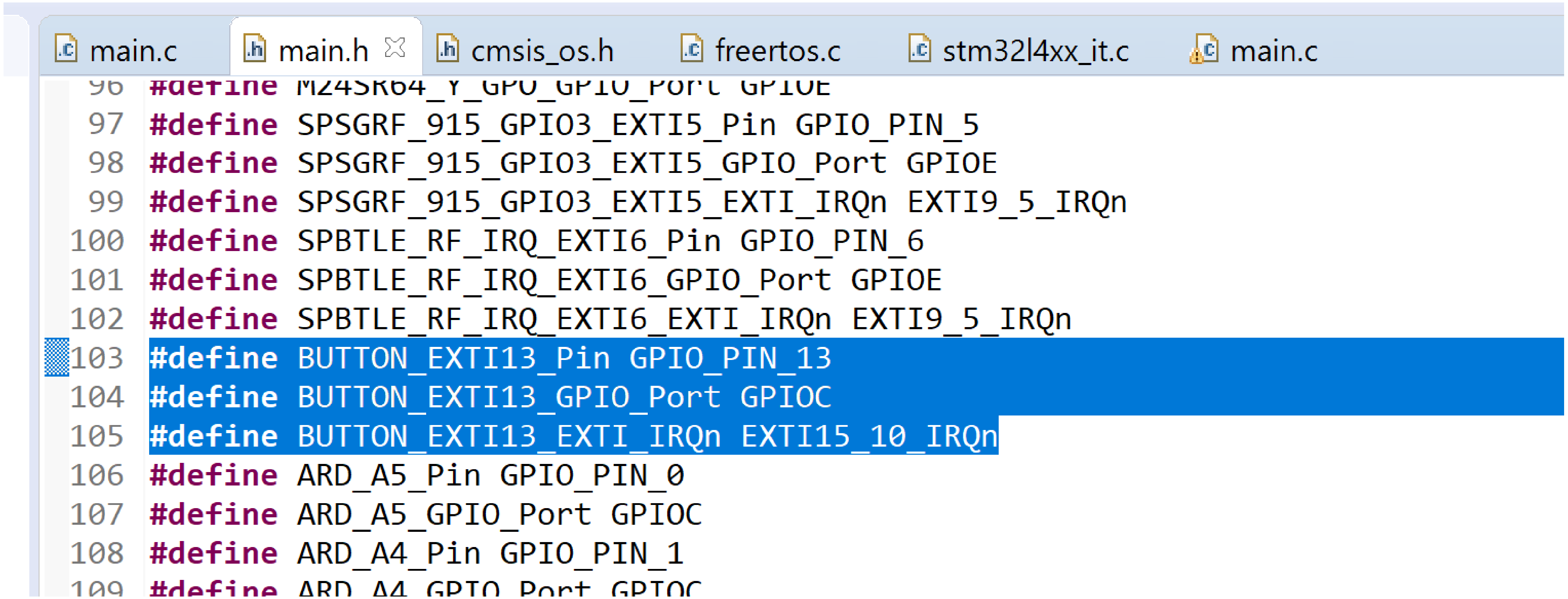
```
/* USER CODE BEGIN Header_StartTask02 */
/**
* @brief Function implementing the myTask02 thread.
* @param argument: Not used
* @retval None
*/
/* USER CODE END Header_StartTask02 */
void StartTask02(void const * argument)
{
  /* USER CODE BEGIN StartTask02 */
  /* Infinite loop */
  for(;;)
  {
    // Wait on all three bits to be set
    xEventGroupWaitBits(xEventGroup,
            mainISR_BIT | mainTASK_BIT_1 | mainTASK_BIT_2, //Bits to wait for
            pdTRUE, //Clear bits on exit
            pdTRUE, //Wait on all  bits
            portMAX_DELAY);

    // Turn LED on
    HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 1);

    // Delay
    osDelay(1000);

    // Turn LED off
    HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 0);

    // Delay
    osDelay(1000);

  }
  /* USER CODE END StartTask02 */
}
```

Screenshot of LED2 #defines



```
#define QUAD_SPI_BK1_IO3_Pin GPIO_PIN_15
#define QUAD_SPI_BK1_IO3_GPIO_Port GPIOE
#define INTERNAL_I2C2_SCL_Pin GPIO_PIN_10
#define INTERNAL_I2C2_SCL_GPIO_Port GPIOB
#define INTERNAL_I2C2_SDA_Pin GPIO_PIN_11
#define INTERNAL_I2C2_SDA_GPIO_Port GPIOB
#define ISM43362_BOOT0_Pin GPIO_PIN_12
#define ISM43362_BOOT0_GPIO_Port GPIOB
#define ISM43362_WAKEUP_Pin GPIO_PIN_13
#define ISM43362_WAKEUP_GPIO_Port GPIOB
#define LED2_Pin GPIO_PIN_14
#define LED2_GPIO_Port GPIOB
#define SPSGRF_915_SDN_Pin GPIO_PIN_15
#define SPSGRF_915_SDN_GPIO_Port GPIOB
#define INTERNAL_UART3_TX_Pin GPIO_PIN_8
```

Screenshot of BLUE Button Interrupt Defines



```
#define M24SR64_Y_GPO_GPIO_Port GPIOE
#define SPSGRF_915_GPIO3_EXTI5_Pin GPIO_PIN_5
#define SPSGRF_915_GPIO3_EXTI5_GPIO_Port GPIOE
#define SPSGRF_915_GPIO3_EXTI5_EXTI_IRQn EXTI9_5_IRQn
#define SPBTLE_RF_IRQ_EXTI6_Pin GPIO_PIN_6
#define SPBTLE_RF_IRQ_EXTI6_GPIO_Port GPIOE
#define SPBTLE_RF_IRQ_EXTI6_EXTI_IRQn EXTI9_5_IRQn
#define BUTTON_EXTI13_Pin GPIO_PIN_13
#define BUTTON_EXTI13_GPIO_Port GPIOC
#define BUTTON_EXTI13_EXTI_IRQn EXTI15_10_IRQn
#define ARD_A5_Pin GPIO_PIN_0
#define ARD_A5_GPIO_Port GPIOC
#define ARD_A4_Pin GPIO_PIN_1
#define ARD_A4_GPIO_Port GPIOC
```