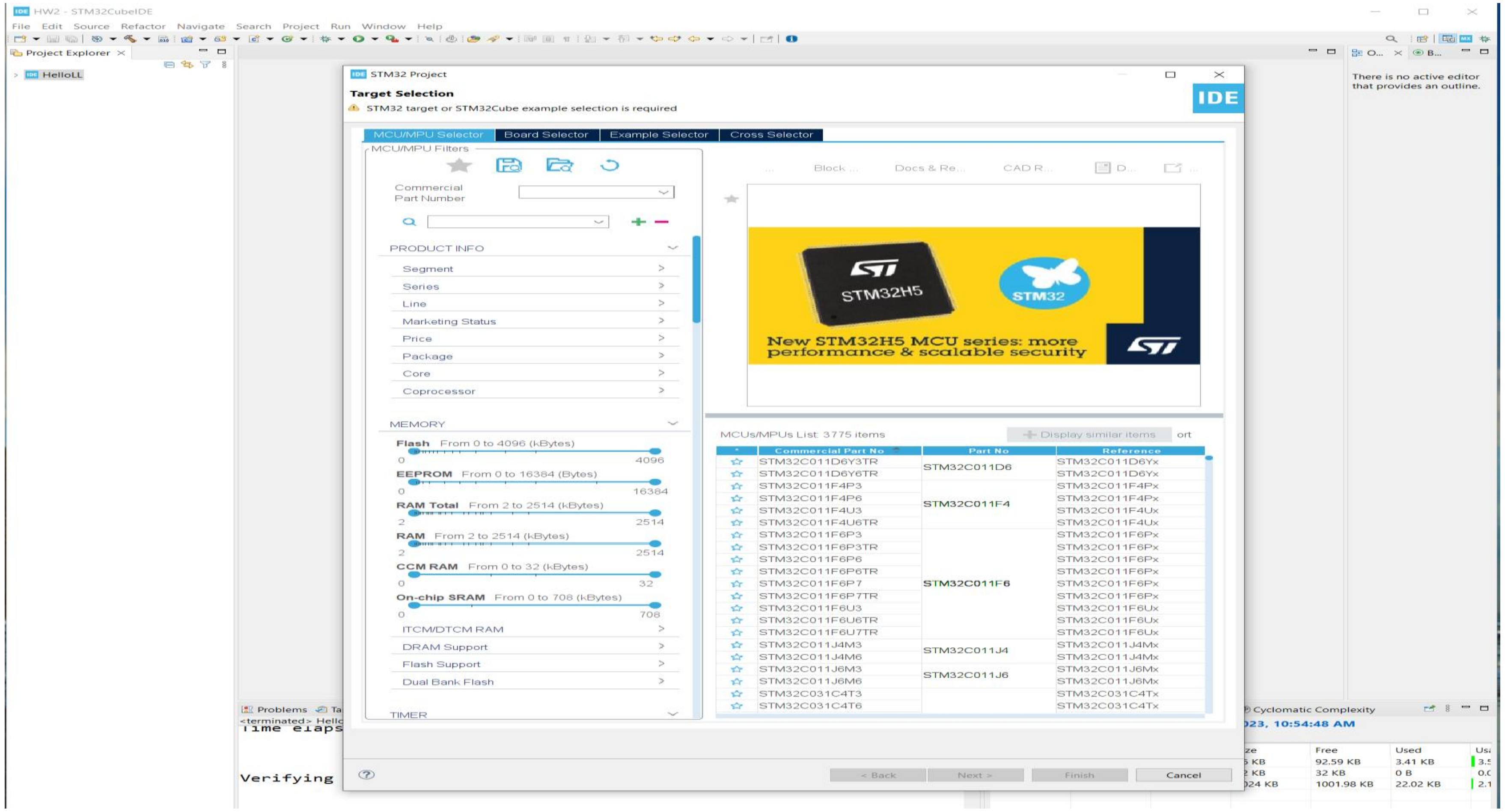
UCSD Embedded RTOS Assignment 2

By

Hsuankai Chang

hsuankac@umich.edu

Step 1. Startup STM32CubeIDE and create new STM32 project



Target Selection

STM32 target or STM32Cube example selection is required

Flash From 0 to 4096 (kBytes) 0 4096

EEPROM From 0 to 16384 (Bytes) 0 16384

RAM Total From 2 to 2514 (kBytes) 2 2514

RAM From 2 to 2514 (kBytes) 2 2514

CCM RAM From 0 to 32 (kBytes) 0 32

On-chip SRAM From 0 to 708 (kBytes) 0 708

There is no active editor that provides an outline.

MCUs/MPUs List: 3775 items

*	Commercial Part No	Part No	Reference
☆	STM32C011D6Y3TR	STM32C011D6	STM32C011D6Yx
☆	STM32C011D6Y6TR		STM32C011D6Yx
☆	STM32C011F4P3	STM32C011F4	STM32C011F4Px
☆	STM32C011F4P6		STM32C011F4Px
☆	STM32C011F4U3		STM32C011F4Ux
☆	STM32C011F4U6TR		STM32C011F4Ux
☆	STM32C011F6P3	STM32C011F6	STM32C011F6Px
☆	STM32C011F6P3TR		STM32C011F6Px
☆	STM32C011F6P6		STM32C011F6Px
☆	STM32C011F6P6TR		STM32C011F6Px
☆	STM32C011F6P7		STM32C011F6Px
☆	STM32C011F6P7TR		STM32C011F6Px
☆	STM32C011F6U3		STM32C011F6Ux
☆	STM32C011F6U6TR		STM32C011F6Ux
☆	STM32C011F6U7TR		STM32C011F6Ux
☆	STM32C011J4M3	STM32C011J4	STM32C011J4Mx
☆	STM32C011J4M6		STM32C011J4Mx
☆	STM32C011J6M3	STM32C011J6	STM32C011J6Mx
☆	STM32C011J6M6		STM32C011J6Mx
☆	STM32C031C4T3		STM32C031C4Tx
☆	STM32C031C4T6		STM32C031C4Tx

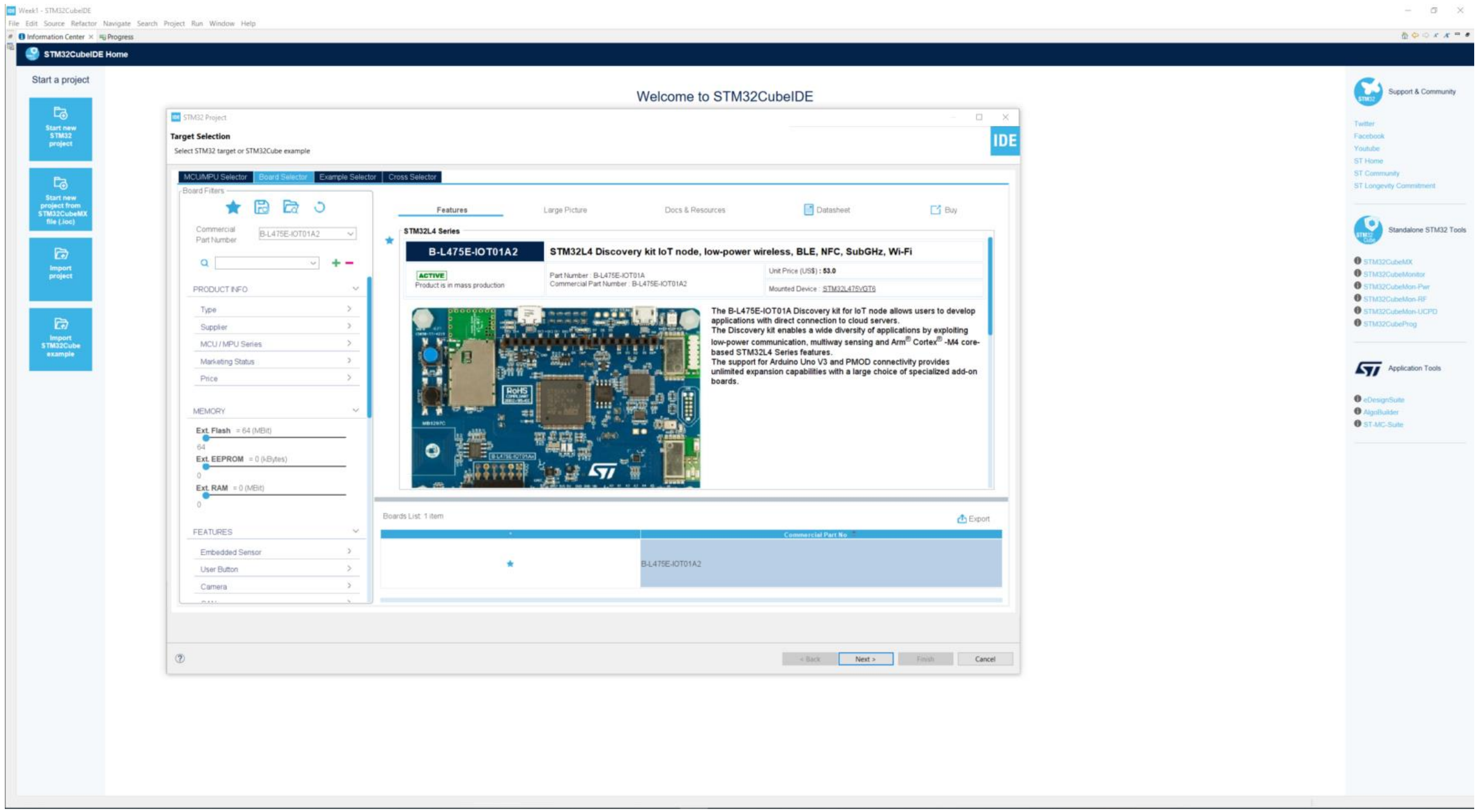
```
<terminated> Hello
Time elaps

Verifying
```

023, 10:54:48 AM

ze	Free	Used	Us
5 KB	92.59 KB	3.41 KB	3.5
2 KB	32 KB	0 B	0.0
024 KB	1001.98 KB	22.02 KB	2.1

Step 2. Access board selector and type in the board you use, click Next



Target Selection

Select STM32 target or STM32Cube example

Commercial Part Number: B-L475E-IOT01A2

STM32L4 Series

B-L475E-IOT01A2 **STM32L4 Discovery kit IoT node, low-power wireless, BLE, NFC, SubGHz, Wi-Fi**

ACTIVE
Product is in mass production

Part Number : B-L475E-IOT01A
Commercial Part Number : B-L475E-IOT01A2

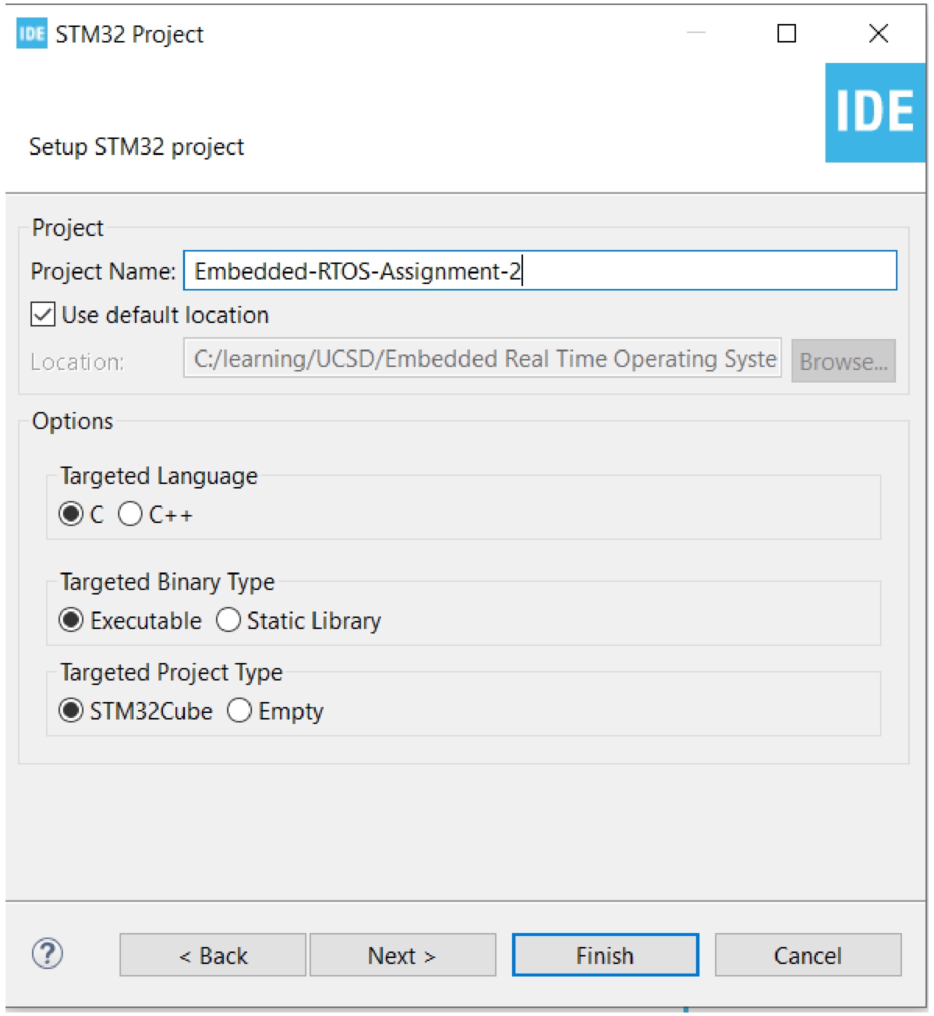
Unit Price (US$) : **53.0**
Mounted Device : STM32L475VGT6

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.

Boards List: 1 item

*	Commercial Part No
★	B-L475E-IOT01A2

Step 3. Enter the project name then click Next


STM32 Project

Setup STM32 project

Project

Project Name: Embedded-RTOS-Assignment-2

[✓] Use default location

Location: C:/learning/UCSD/Embedded Real Time Operating Syste

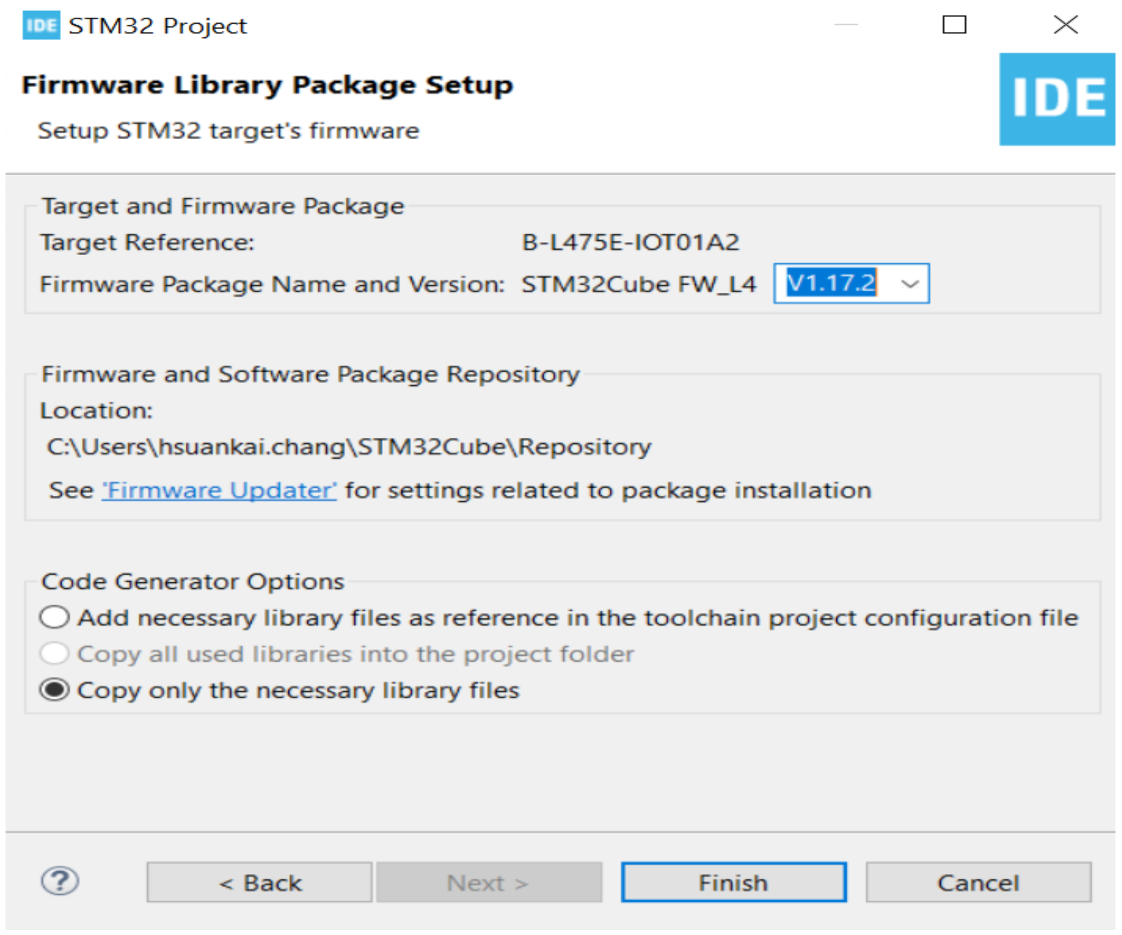
Options

Targeted Language: [✓ C] C++

Targeted Binary Type: [✓ Executable] Static Library

Targeted Project Type: [✓ STM32Cube] Empty

Step 4. See the firmware package name and version


STM32 Project

Firmware Library Package Setup

Setup STM32 target's firmware

Target and Firmware Package

Target Reference: B-L475E-IOT01A2

Firmware Package Name and Version: STM32Cube FW_L4 V1.17.2

Firmware and Software Package Repository

Location:

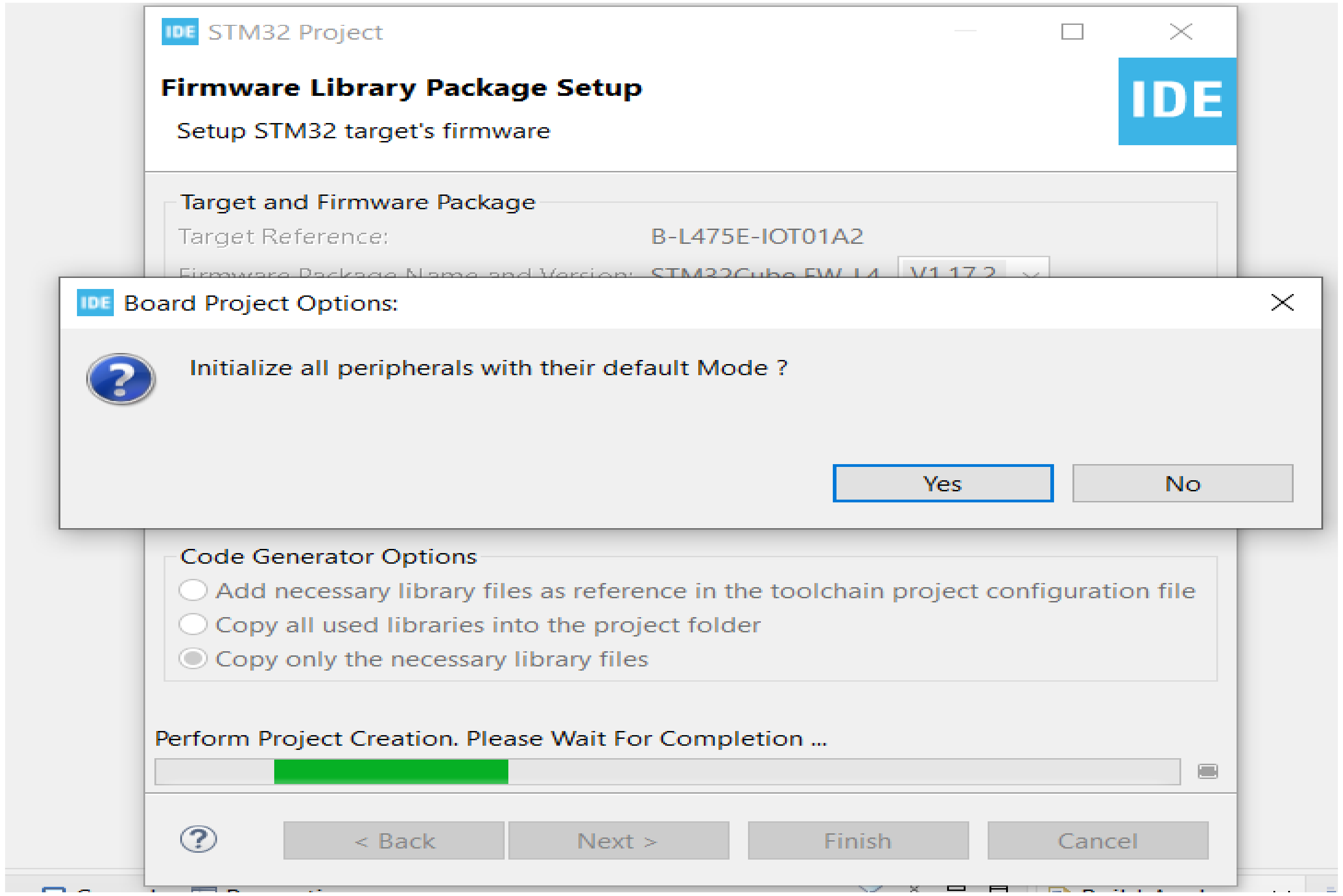
C:\Users\hsuankai.chang\STM32Cube\Repository

See 'Firmware Updater' for settings related to package installation

Code Generator Options

- ○ Add necessary library files as reference in the toolchain project configuration file
- ○ Copy all used libraries into the project folder
- ● Copy only the necessary library files

Step 5. Click yes to initialize all peripherals to default



STM32 Project

Firmware Library Package Setup

Setup STM32 target's firmware

Target and Firmware Package

Target Reference: B-L475E-IOT01A2

Board Project Options:

Initialize all peripherals with their default Mode ?

Yes No

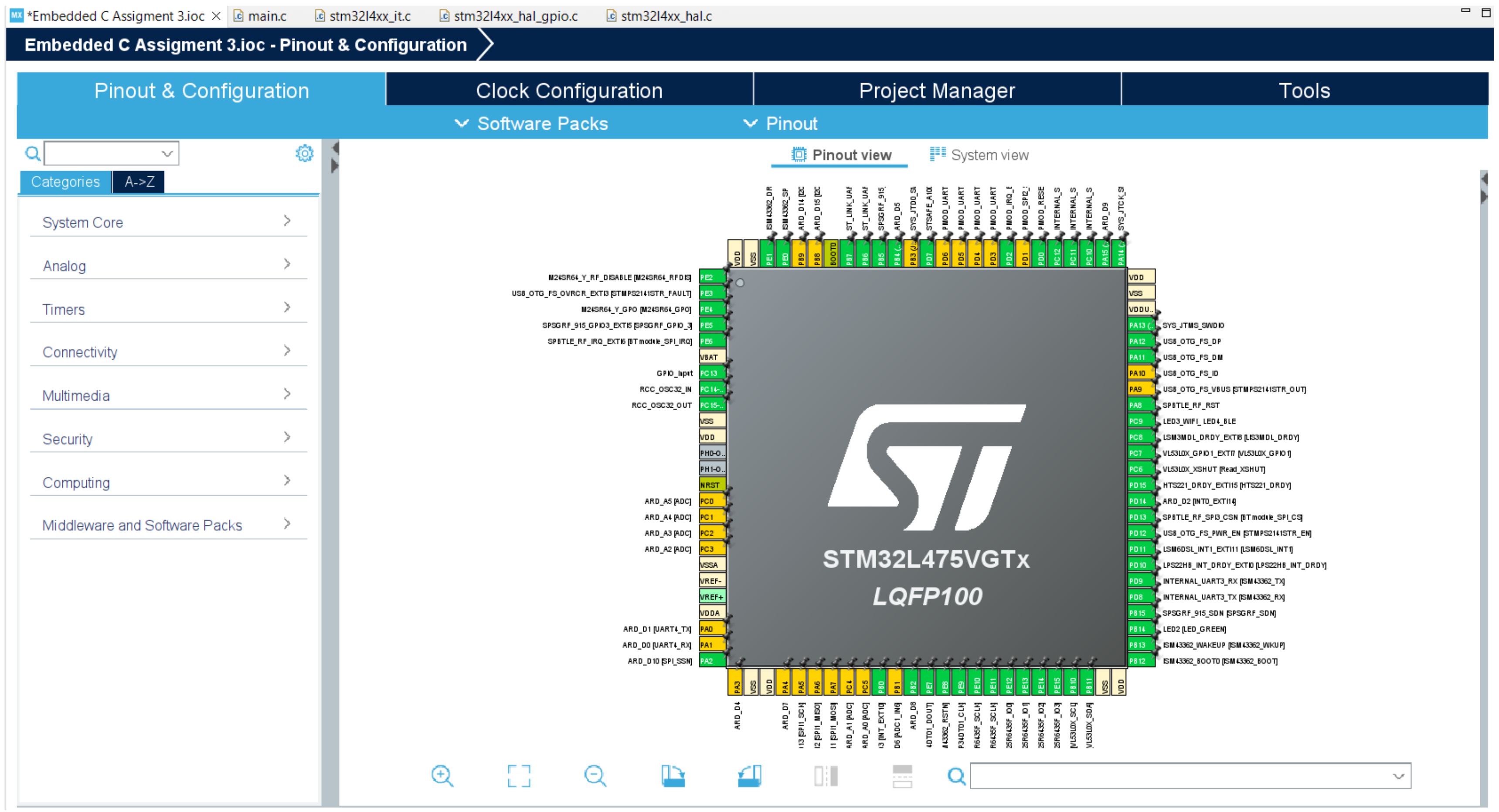
Code Generator Options

- ○ Add necessary library files as reference in the toolchain project configuration file
- ○ Copy all used libraries into the project folder
- ● Copy only the necessary library files

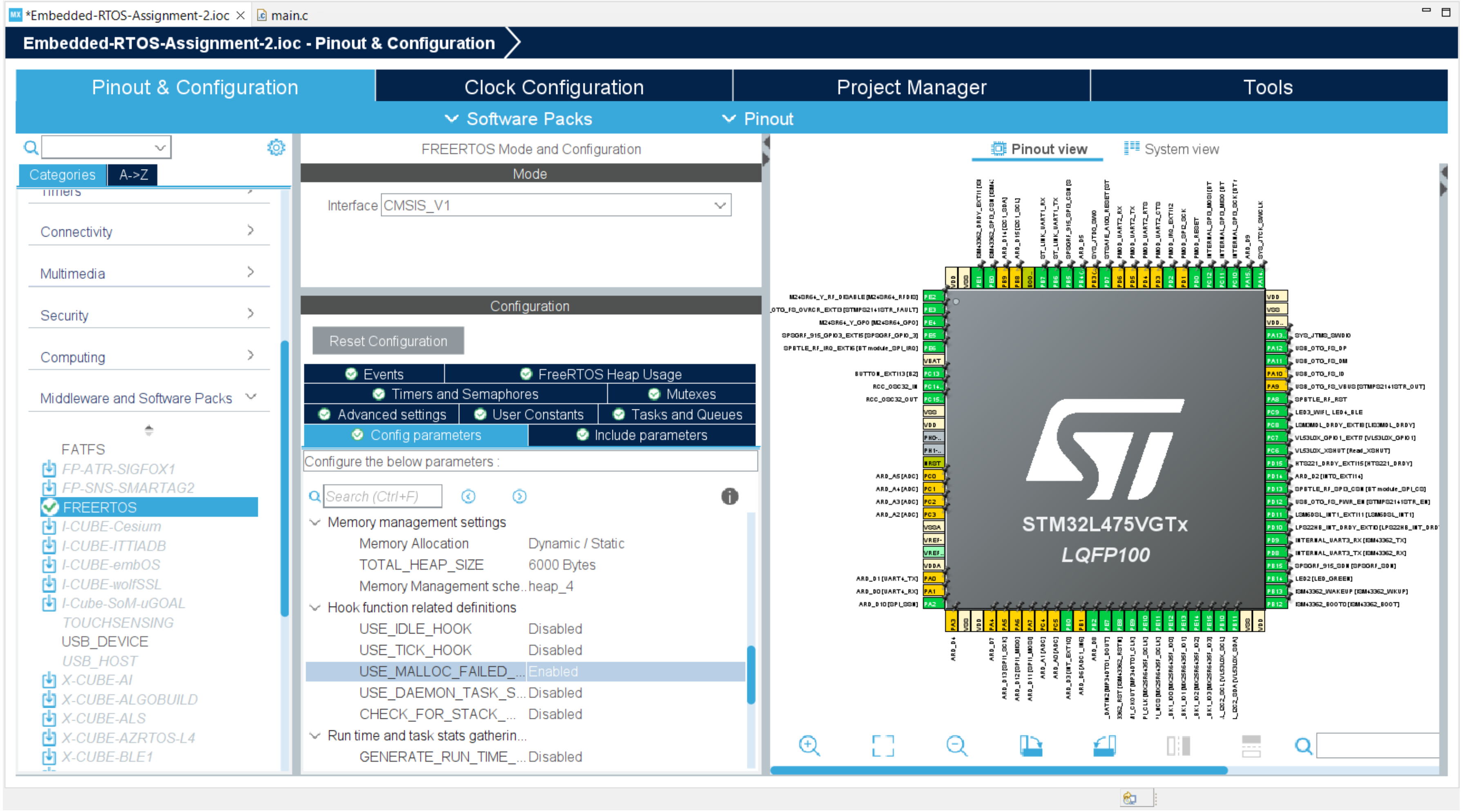
Perform Project Creation. Please Wait For Completion ...

< Back Next > Finish Cancel

Step 6. When in .ioc file, click Pinout & Configurations


*Embedded C Assigment 3.ioc
main.c
stm32l4xx_it.c
stm32l4xx_hal_gpio.c
stm32l4xx_hal.c
Embedded C Assigment 3.ioc - Pinout & Configuration
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
Categories
A->Z
System Core
Analog
Timers
Connectivity
Multimedia
Security
Computing
Middleware and Software Packs
Pinout view
System view
M24SR64_Y_RF_DISABLE [M24SR64_RFDIS]
USB_OTG_FS_OVRCR_EXTI3 [STMPS2141STR_FAULT]
M24SR64_Y_GPO [M24SR64_GPO]
SPSGRF_915_GPIO3_EXTI5 [SPSGRF_GPIO_3]
SPBTLE_RF_IRQ_EXTI6 [BT module_SPI_IRQ]
GPIO_Input
RCC_OSC32_IN
RCC_OSC32_OUT
ARD_A5 [ADC]
ARD_A4 [ADC]
ARD_A3 [ADC]
ARD_A2 [ADC]
ARD_D1 [UART4_TX]
ARD_D0 [UART4_RX]
ARD_D10 [SPI_SSN]
SYS_JTMS_SWDIO
USB_OTG_FS_DP
USB_OTG_FS_DM
USB_OTG_FS_ID
USB_OTG_FS_VBUS [STMPS2141STR_OUT]
SPBTLE_RF_RST
LED3_WIFI_ LED4_BLE
LSM3MDL_DRDY_EXTI8 [LIS3MDL_DRDY]
VL53L0X_GPIO1_EXTI7 [VL53L0X_GPIO1]
VL53L0X_XSHUT [Read_XSHUT]
HTS221_DRDY_EXTI15 [HTS221_DRDY]
ARD_D2 [INT0_EXTI14]
SPBTLE_RF_SPI3_CSN [BT module_SPI_CS]
USB_OTG_FS_PWR_EN [STMPS2141STR_EN]
LSM6DSL_INT1_EXTI11 [LSM6DSL_INT1]
LPS22HB_INT_DRDY_EXTI0 [LPS22HB_INT_DRDY]
INTERNAL_UART3_RX [ISM43362_TX]
INTERNAL_UART3_TX [ISM43362_RX]
SPSGRF_915_SDN [SPSGRF_SDN]
LED2 [LED_GREEN]
ISM43362_WAKEUP [ISM43362_WKUP]
ISM43362_BOOT0 [ISM43362_BOOT]
STM32L475VGTx
LQFP100

Step 7. Enable the FreeRTOS in middleware. Configure FreeRTOS for a HEAP memory size of 6000. Keep the default dynamic/static allocation along with the default memory 4 allocation option. Enable the malloc failed hook macro.



*Embedded-RTOS-Assignment-2.ioc | main.c

Embedded-RTOS-Assignment-2.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

Categories | A->Z

Connectivity
Multimedia
Security
Computing
Middleware and Software Packs

FATFS
FP-ATR-SIGFOX1
FP-SNS-SMARTAG2
FREERTOS
I-CUBE-Cesium
I-CUBE-ITTIADB
I-CUBE-embOS
I-CUBE-wolfSSL
I-Cube-SoM-uGOAL
TOUCHSENSING
USB_DEVICE
USB_HOST
X-CUBE-AI
X-CUBE-ALGOBUILD
X-CUBE-ALS
X-CUBE-AZRTOS-L4
X-CUBE-BLE1

FREERTOS Mode and Configuration

Mode

Interface CMSIS_V1

Configuration

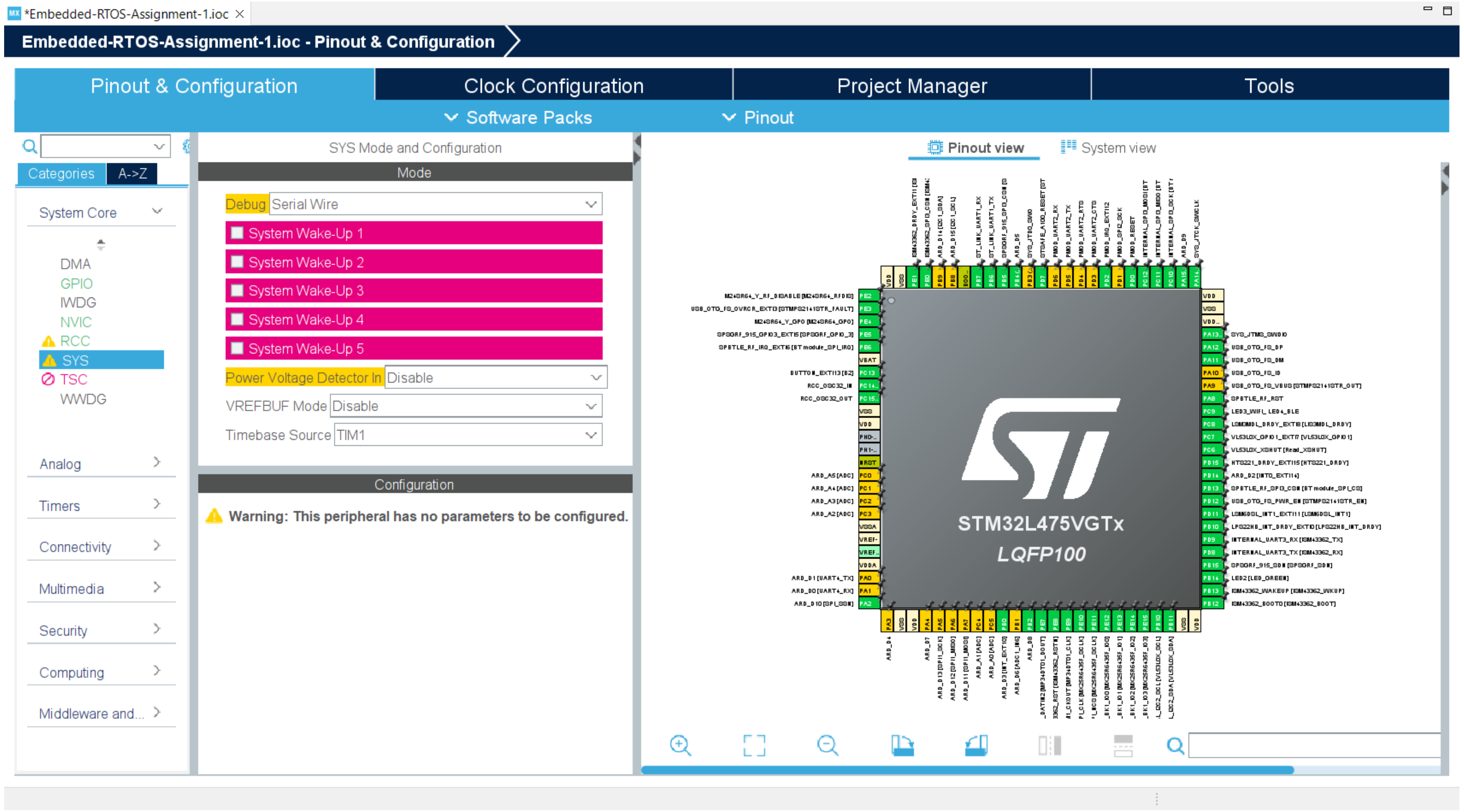
Reset Configuration

Events | FreeRTOS Heap Usage
Timers and Semaphores | Mutexes
Advanced settings | User Constants | Tasks and Queues
Config parameters | Include parameters

Configure the below parameters :

Memory management settings	
Memory Allocation	Dynamic / Static
TOTAL_HEAP_SIZE	6000 Bytes
Memory Management sche...	heap_4
Hook function related definitions	
USE_IDLE_HOOK	Disabled
USE_TICK_HOOK	Disabled
USE_MALLOC_FAILED_...	Enabled
USE_DAEMON_TASK_S...	Disabled
CHECK_FOR_STACK_...	Disabled
Run time and task stats gatherin...	
GENERATE_RUN_TIME_...	Disabled

Step 8. Change Timebase from systick to TIM1



*Embedded-RTOS-Assignment-1.ioc

Embedded-RTOS-Assignment-1.ioc - Pinout & Configuration

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

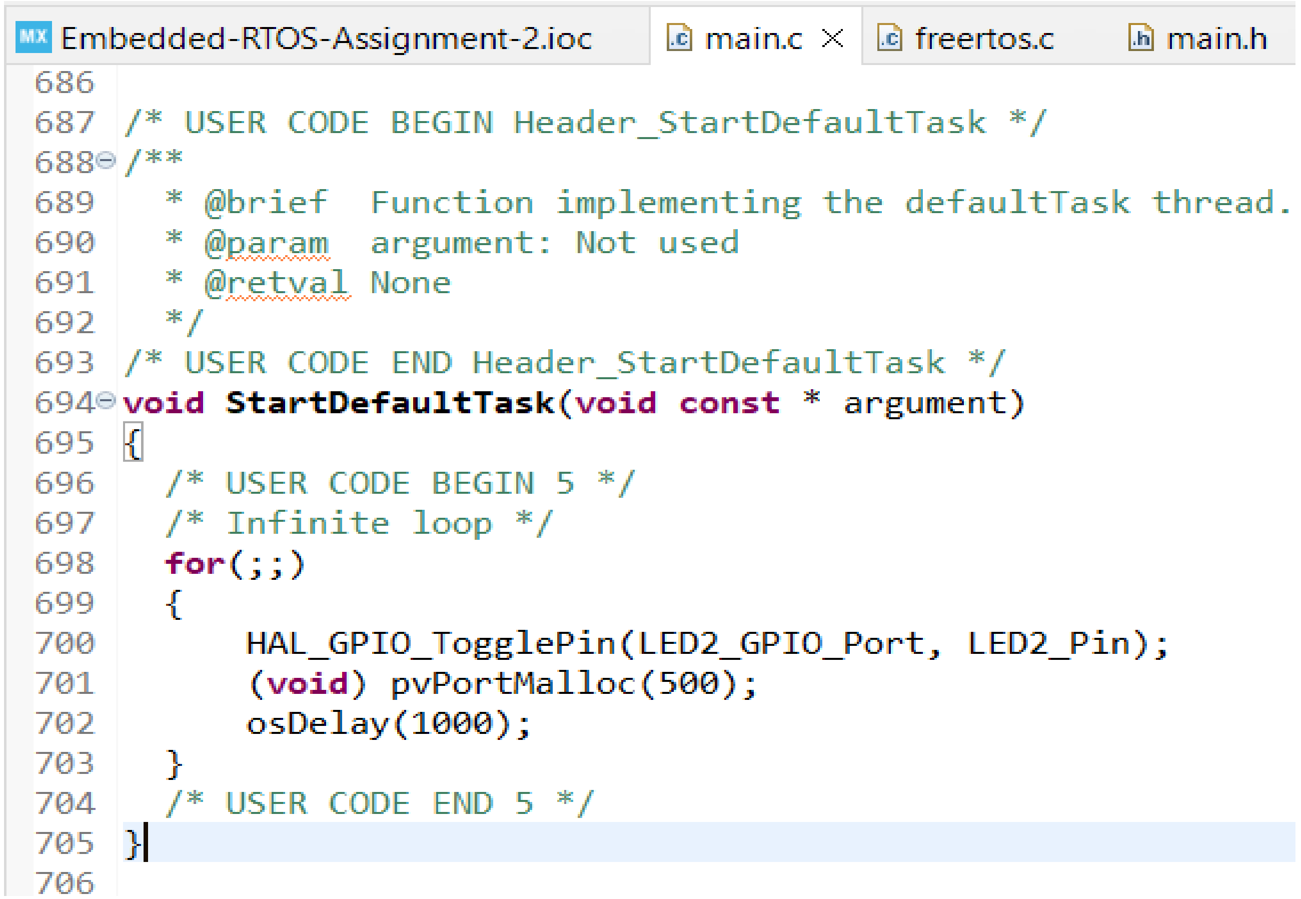
VREFBUF Mode Disable

Timebase Source TIM1

Configuration

Warning: This peripheral has no parameters to be configured.

Step 9. In your default task, blink the LED2 every second and also use pvPortMalloc() to allocate 500 bytes every time you blink the LED2.

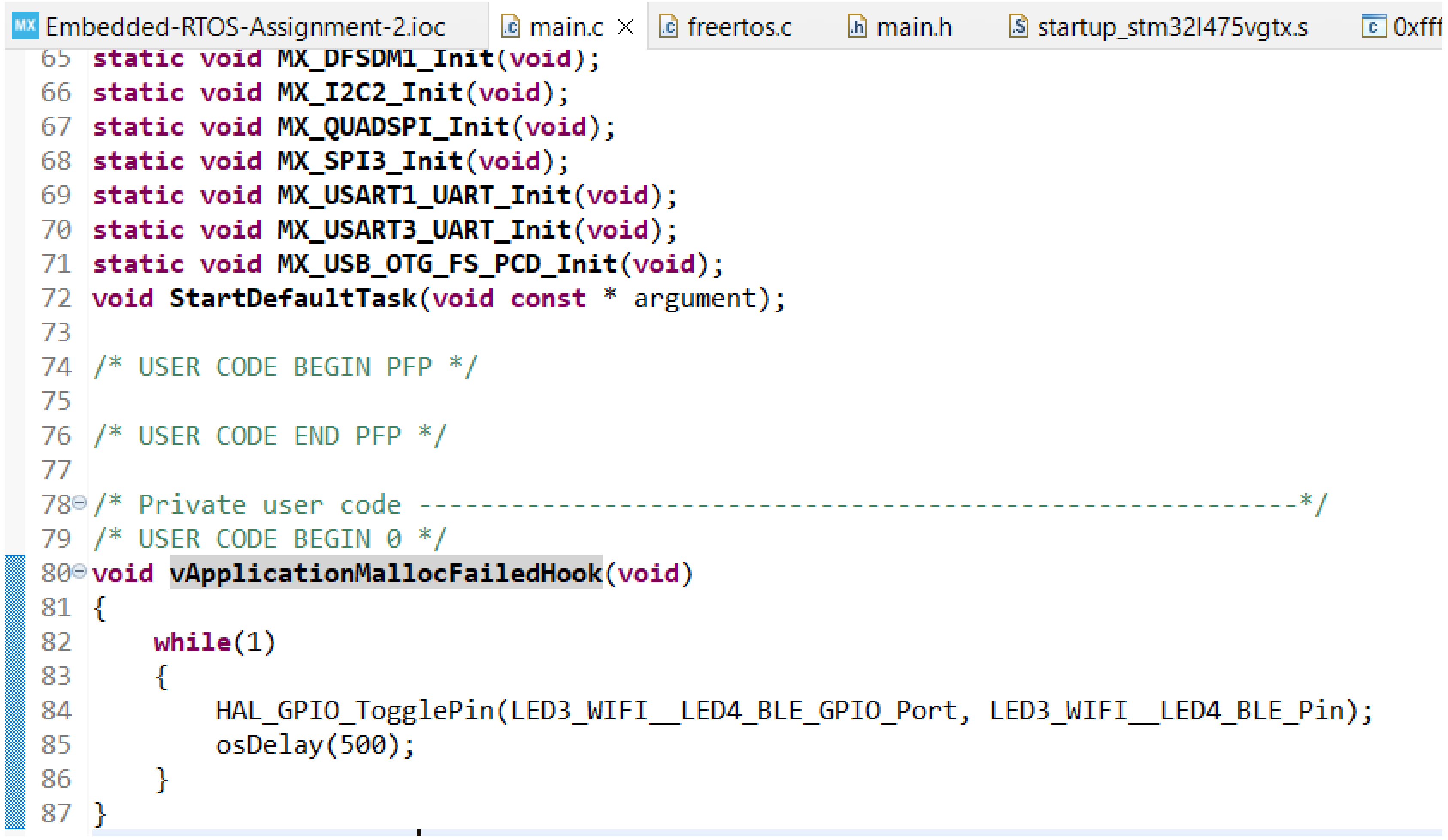


```

/* USER CODE BEGIN Header_StartDefaultTask */
/**
  * @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(;;)
  {
	  HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
	  (void) pvPortMalloc(500);
	  osDelay(1000);
  }
  /* USER CODE END 5 */
}

```

Step 10. Add the vApplicationMallocFailedHook() to your code so that when you run out of memory (remember you have a memory leak from in the default task as described in the previous bullet). The code in the vApplication MallocFailedHook() should be a "while (1) { }" that blinks the Wifi/BlE LEDs at a 0.5 second (1/2 second) rate.



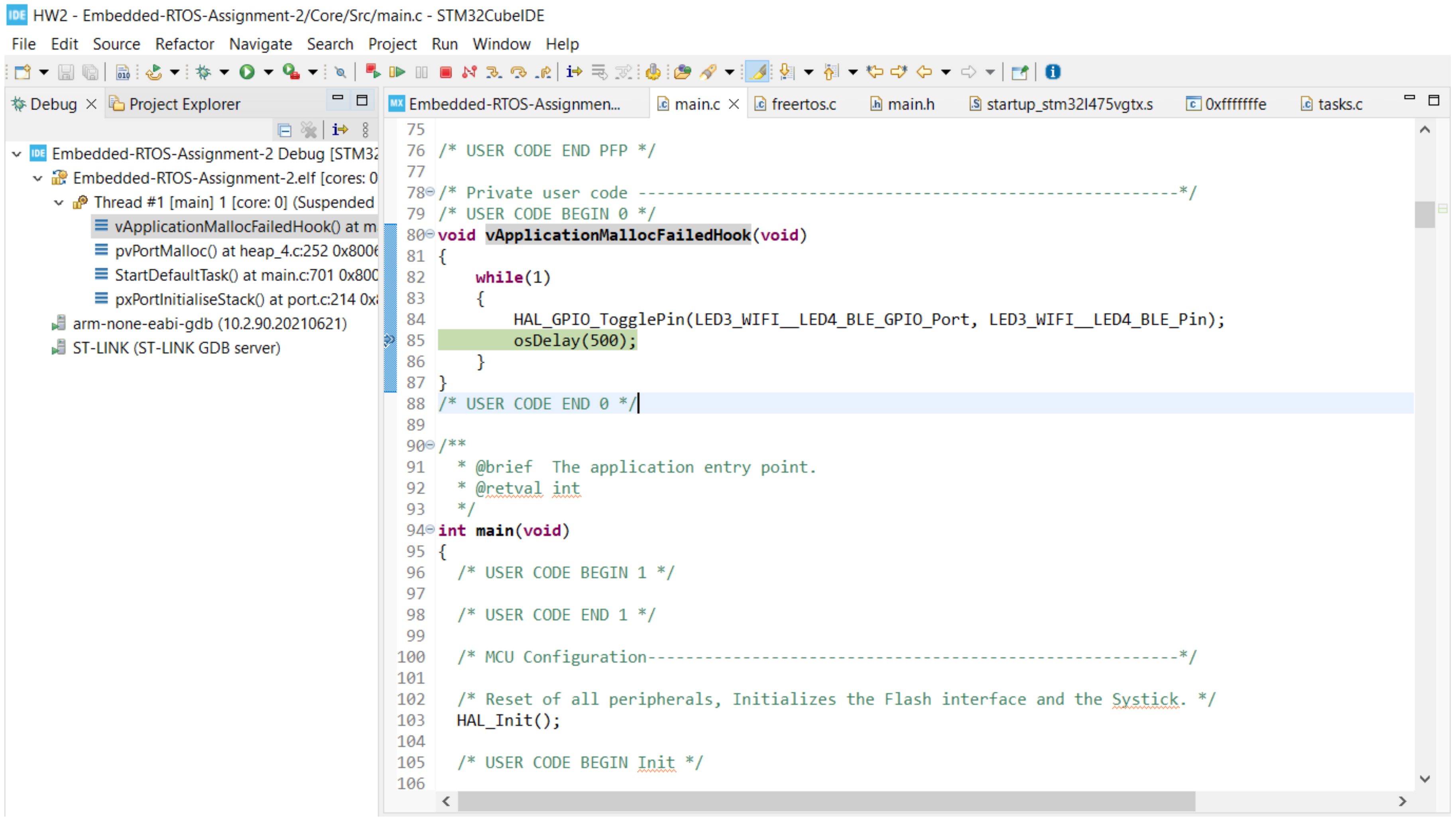
```
static void MX_DFSDM1_Init(void);
static void MX_I2C2_Init(void);
static void MX_QUADSPI_Init(void);
static void MX_SPI3_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_USART3_UART_Init(void);
static void MX_USB_OTG_FS_PCD_Init(void);
void StartDefaultTask(void const * argument);

/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
void vApplicationMallocFailedHook(void)
{
    while(1)
    {
        HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
        osDelay(500);
    }
}
```

Step 11. Build and run the code, since after several loops, memory leaks occur, so the process will stuck in vApplicationMallocFailedHook and keep blinking WIFI/BLE LED rather than blinking LED2.


HW2 - Embedded-RTOS-Assignment-2/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Project Explorer
Embedded-RTOS-Assignmen...
main.c
freertos.c
main.h
startup_stm32l475vgtx.s
0xfffffffe
tasks.c

```
Embedded-RTOS-Assignment-2 Debug [STM32
  Embedded-RTOS-Assignment-2.elf [cores: 0
    Thread #1 [main] 1 [core: 0] (Suspended
      vApplicationMallocFailedHook() at m
      pvPortMalloc() at heap_4.c:252 0x800
      StartDefaultTask() at main.c:701 0x800
      pxPortInitialiseStack() at port.c:214 0x
  arm-none-eabi-gdb (10.2.90.20210621)
  ST-LINK (ST-LINK GDB server)
```

```c

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
void vApplicationMallocFailedHook(void)
{
    while(1)
    {
        HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
        osDelay(500);
    }
}
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

```