

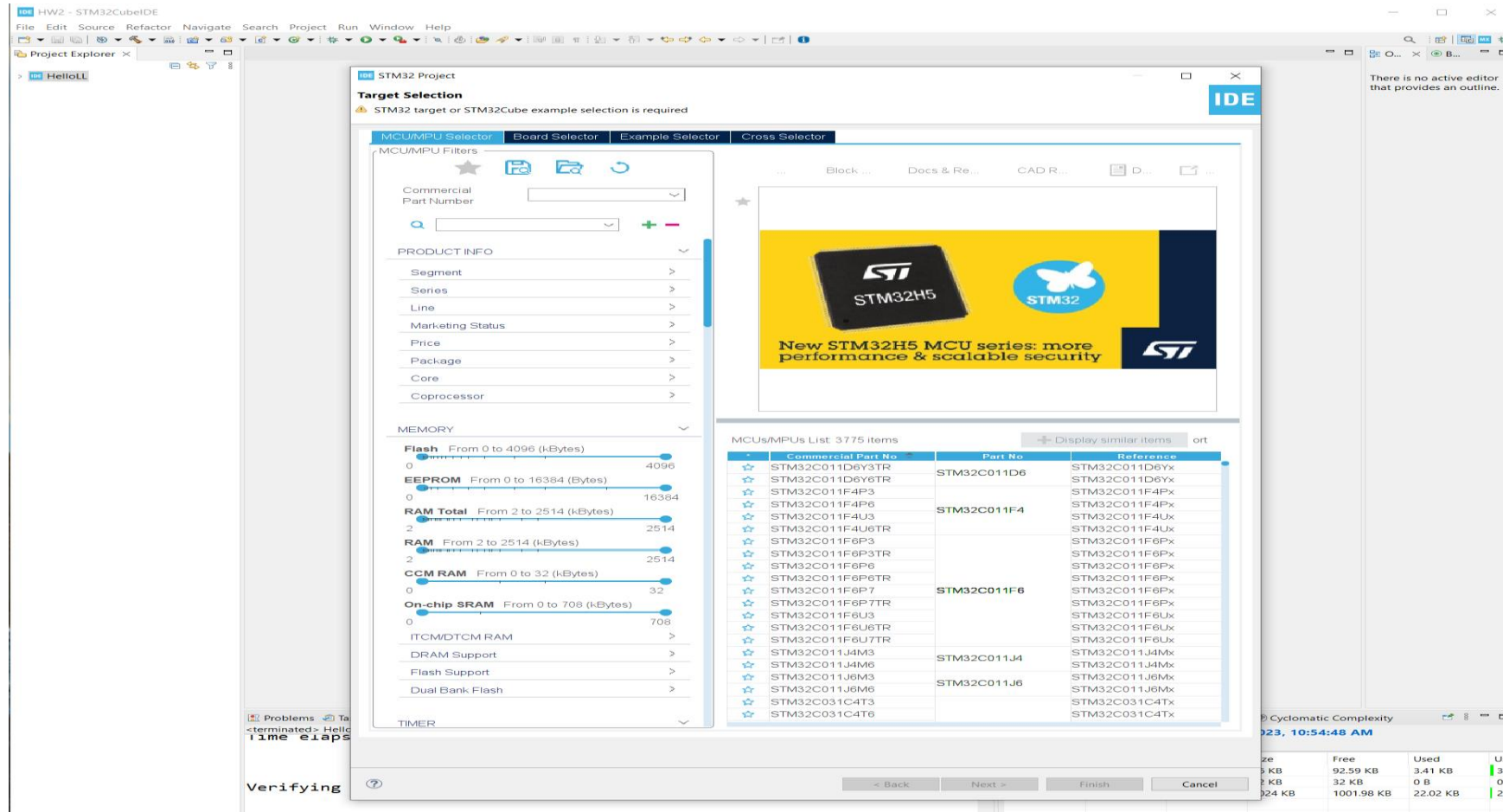
UCSD Embedded RTOS Assignment 1

By

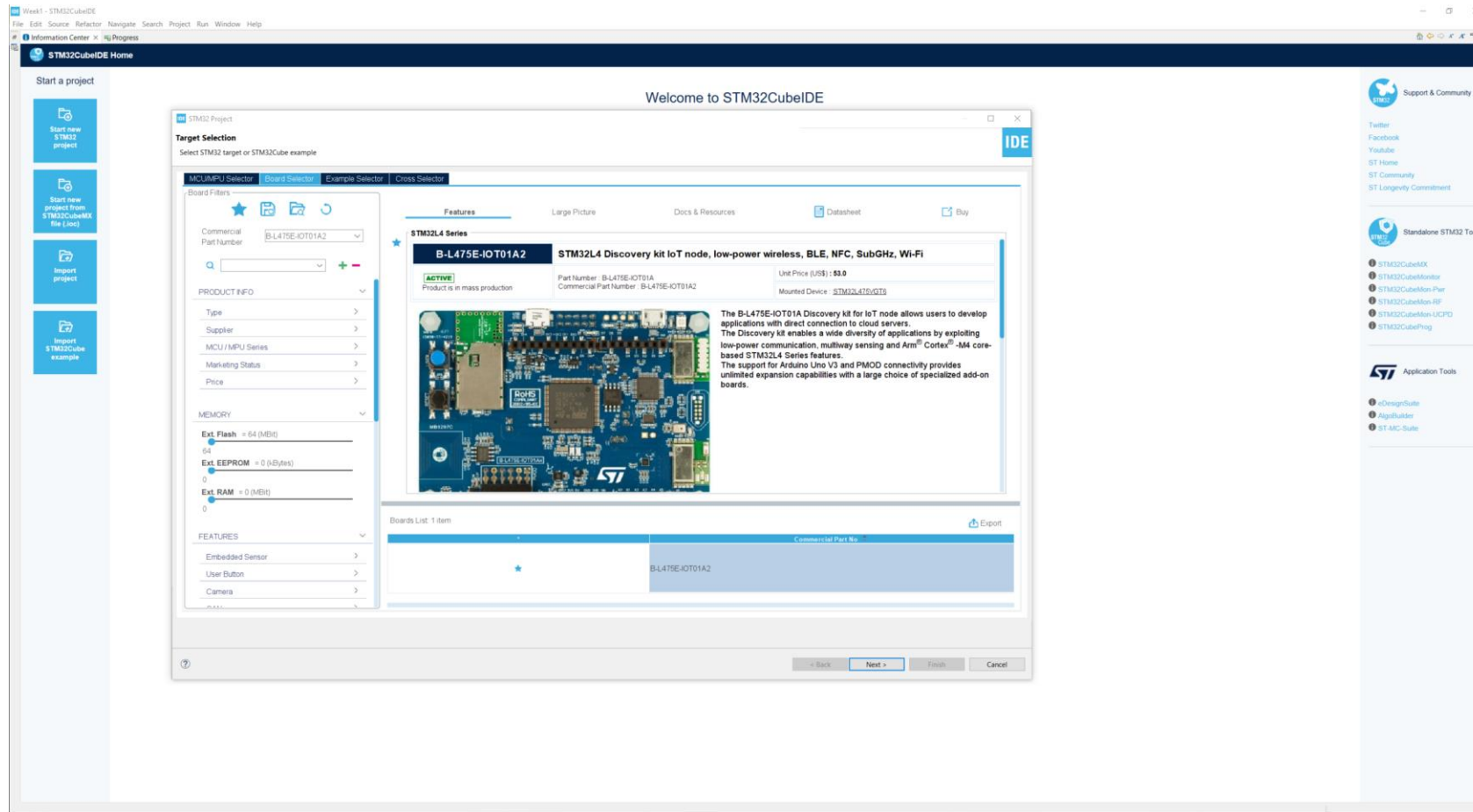
Hsuankai Chang

hsuankac@umich.edu

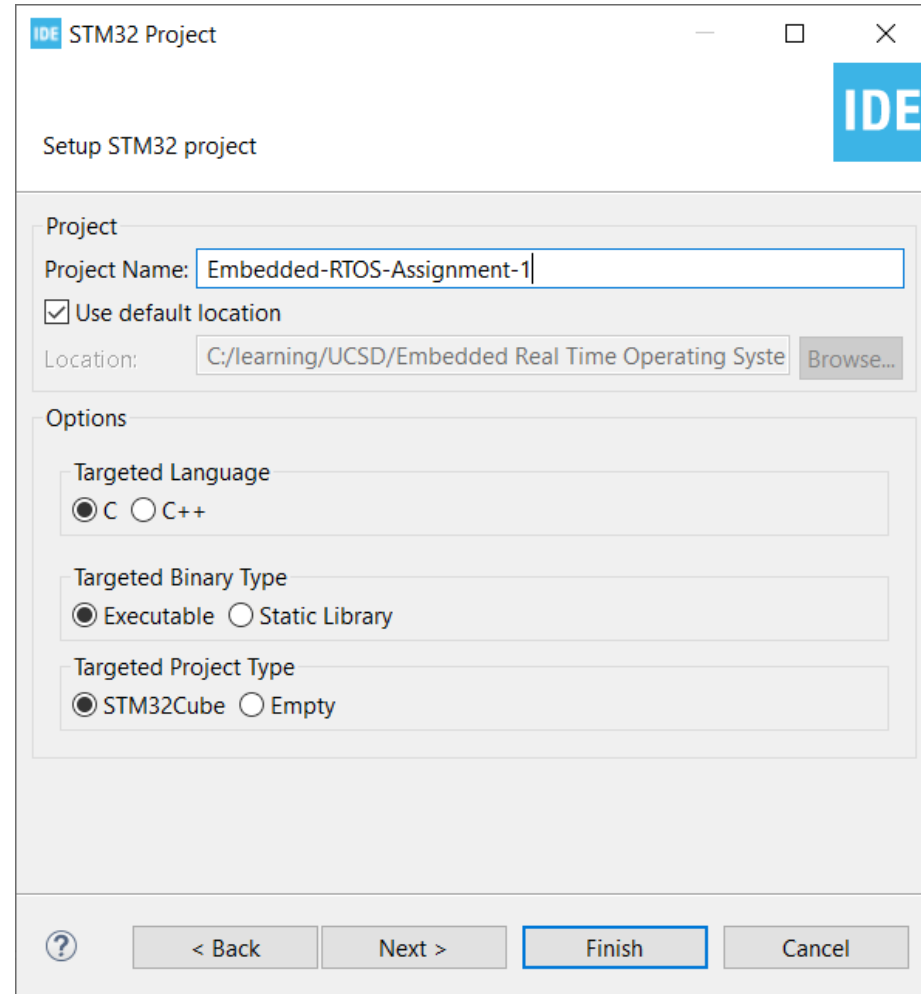
Step 1. Startup STM32CubeIDE and create new STM32 project



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



Step 3. Enter the project name then click Next



The image shows a 'Setup STM32 project' dialog box from the IDE. The window title is 'IDE STM32 Project'. The main heading is 'Setup STM32 project'. Under the 'Project' section, the 'Project Name' field contains 'Embedded-RTOS-Assignment-1'. The 'Use default location' checkbox is checked. The 'Location' field shows 'C:/learning/UCSD/Embedded Real Time Operating System' with a 'Browse...' button. Under the 'Options' section, 'Targeted Language' has 'C' selected, 'Targeted Binary Type' has 'Executable' selected, and 'Targeted Project Type' has 'STM32Cube' selected. At the bottom, there are buttons for '?', '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

IDE STM32 Project

Setup STM32 project

Project

Project Name: Embedded-RTOS-Assignment-1

☒ Use default location

Location: C:/learning/UCSD/Embedded Real Time Operating System Browse...

Options

Targeted Language

☒ C ☐ C++

Targeted Binary Type

☒ Executable ☐ Static Library

Targeted Project Type

☒ STM32Cube ☐ Empty

? < Back Next > Finish Cancel

Step 4. See the firmware package name and version



The image shows a Windows-style dialog box titled "STM32 Project" with a subtitle "Firmware Library Package Setup". The subtitle is followed by the instruction "Setup STM32 target's firmware". The dialog is divided into three sections: "Target and Firmware Package", "Firmware and Software Package Repository", and "Code Generator Options". In the first section, "Target Reference" is set to "B-L475E-IOT01A2" and "Firmware Package Name and Version" is set to "STM32Cube FW_L4" with a dropdown menu showing "V1.17.2". The second section shows the "Location" as "C:\Users\hsuankai.chang\STM32Cube\Repository" and includes a link to the "Firmware Updater". The third section contains three radio button options for code generation, with the third option, "Copy only the necessary library files", being selected. At the bottom, there are buttons for "?", "< Back", "Next >", "Finish", and "Cancel".

IDE 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

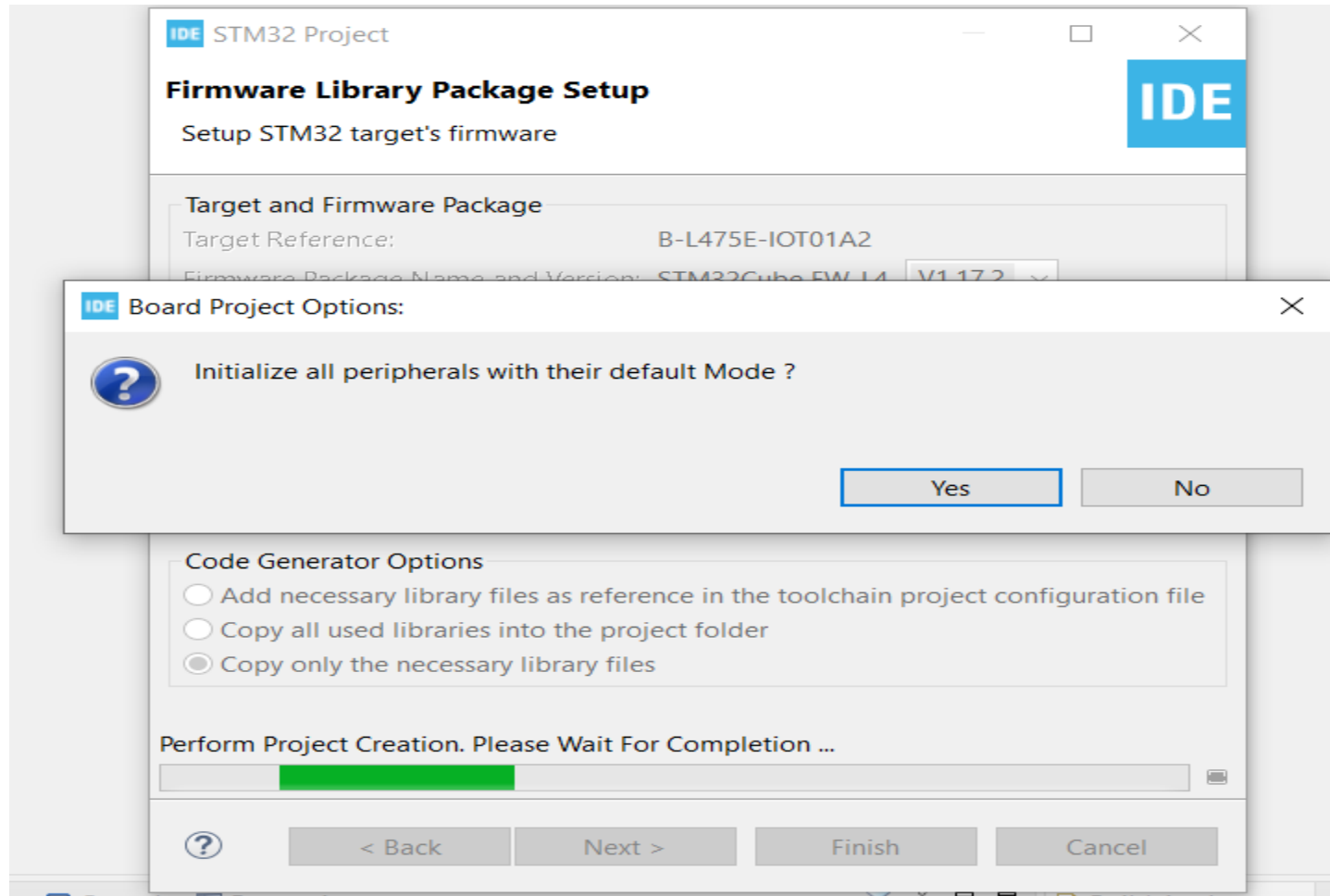
☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

☒ Copy only the necessary library files

? < Back Next > Finish Cancel

Step 5. Click yes to initialize all peripherals to default



Step 7. Enable the FreeRTOS in middleware, leave all the settings as default

MX *Embedded-RTOS-Assignment-1.ioc X

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

Pinout & Configuration Clock Configuration Project Manager Tools

Software Packs Pinout

Categories A-Z

Security >

Computing >

Middleware an... >

FATFS

FP-ATR-SIGFC

FP-SNS-SMAR

I-CUBE-Cesium

I-CUBE-embOS

I-CUBE-wolfSSI

I-Cube-SoM-uG

TOUCHSENSII

USB_DEVICE

USB_HOST

X-CUBE-AI

X-CUBE-ALGO

X-CUBE-ALS

X-CUBE-AZRT

X-CUBE-BLE1

X-CUBE-BLE2

X-CUBE-BLEM

X-CUBE-DISPL

X-CUBE-EEPR

X-CUBE-GNSS

X-CUBE-ISPU

X-CUBE-MEM

FREERTOS

Mode

Interface CMSIS_V1

Configuration

Reset Configuration

Mutexes Events FreeRTOS Heap Usage

Tasks and Queues Timers and Semaphores

Advanced settings User Constants

Config parameters Include parameters

Configure the below parameters:

Search (Ctrl+F)

API

FreeRTOS API CMSIS v1

Versions

FreeRTOS version 10.3.1

CMSIS-RTOS version 1.02

MPU/FPU

ENABLE_MPU Disabled

ENABLE_FPU Disabled

Pinout view System view

STM32L475VGTx LQFP100

Pinout view showing the STM32L475VGTx LQFP100 package with various pins labeled, including USB, UART, SPI, I2C, and other peripherals.

Step 8. Change Timebase from systick to TIM1

*Embedded-RTOS-Assignment-1.ioc X

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

Pinout & Configuration

Categories A-Z

System Core

- DMA
- GPIO
- IWDG
- NVIC
- RCC
- SYS**
- TSC
- WWDG

Analog >

Timers >

Connectivity >

Multimedia >

Security >

Computing >

Middleware and... >

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

VREBUF Mode Disable

Timebase Source TIM1

Configuration

Warning: This peripheral has no parameters to be configured.

Pinout view

STM32L475VGTx LQFP100

Step 9. Generate the code and add the LED2 toggle in default task

```
MX Embedded-RTOS-Assignment-1.ioc  main.c x
677
678 /* USER CODE END 4 */
679
680 /* USER CODE BEGIN Header_StartDefaultTask */
681 /**
682  * @brief Function implementing the defaultTask thread.
683  * @param argument: Not used
684  * @retval None
685  */
686 /* USER CODE END Header_StartDefaultTask */
687 void StartDefaultTask(void const * argument)
688 {
689     /* USER CODE BEGIN 5 */
690     /* Infinite loop */
691     for(;;)
692     {
693         HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
694         osDelay(1000);
695     }
696     /* USER CODE END 5 */
697 }
698
699 /**
700  * @brief Period elapsed callback in non blocking mode
701  * @note This function is called when TIM1 interrupt took place, inside
702  * HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment
703  * a global variable "uwTick" used as application time base.
704  * @param htim : TIM handle
705  * @retval None
706  */
707 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
708 {
```

Step 10. Build and debug the code, you should see LED2 toggling

The screenshot displays an IDE interface for an Embedded-RTOS assignment. The main window shows the source code for `main.c`, which includes a function `StartDefaultTask` that toggles LED2. The bottom panel shows the CDT Build Console with the build output, and the right panel shows the Build Targets and Memory Regions.

Source Code (main.c):

```
677 /* USER CODE END 4 */
678
679
680 /* USER CODE BEGIN Header_StartDefaultTask */
681 /**
682  * @brief Function implementing the defaultTask thread.
683  * @param argument: Not used
684  * @retval None
685  */
686 /* USER CODE END Header_StartDefaultTask */
687 void StartDefaultTask(void const * argument)
688 {
689     /* USER CODE BEGIN 5 */
690     /* Infinite loop */
691     for(;;)
692     {
693         HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
694         osDelay(1000);
695     }
696     /* USER CODE END 5 */
697 }
698
699 /**
700  * @brief Period elapsed callback in non blocking mode
701  * @note This function is called when TIM1 interrupt took place, inside
702  * HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment
703  * a global variable "uTick" used as application time base.
```

Build Console Output:

```
arm-none-eabi-size Embedded-RTOS-Assignment-1.elf
arm-none-eabi-objdump -h -S Embedded-RTOS-Assignment-1.elf > "Embedded-RTOS-Assignment-1.elf"
text data bss dec hex filename
26476 124 7596 34196 8594 Embedded-RTOS-Assignment-1.elf
Finished building: default.size.stdout
Finished building: Embedded-RTOS-Assignment-1.list
21:40:17 Build Finished. 0 errors, 0 warnings. (took 7s.394ms)
```

Build Targets:

- hspi3 : SPI_HandleTypeDef
- huart1 : UART_HandleTypeDef
- huart3 : UART_HandleTypeDef
- hpcd_USB_OTG_FS : PCD_HandleTypeDef
- defaultTaskHandle : osThreadId
- SystemClock_Config(void) : void
- MX_GPIO_Init(void) : void
- MX_DFSDM1_Init(void) : void
- MX_I2C2_Init(void) : void
- MX_QUADSPI_Init(void) : void
- MX_SPI3_Init(void) : void
- MX_USART1_UART_Init(void) : void
- MX_USART3_UART_Init(void) : void
- MX_USB_OTG_FS_PCD_Init(void) : void
- StartDefaultTask(const void*) : void
- main(void) : int
- SystemClock_Config(void) : void
- MX_DFSDM1_Init(void) : void
- MX_I2C2_Init(void) : void
- MX_QUADSPI_Init(void) : void
- MX_SPI3_Init(void) : void
- MX_USART1_UART_Init(void) : void

Memory Regions:

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM	0x20000000	0x20017fff	96 KB	88.47 KB	7.53 KB	7.85%
RAM2	0x10000000	0x10007fff	32 KB	32 KB	0 B	0.00%
FLASH	0x08000000	0x080fffff	1024 KB	998.02 KB	25.98 KB	2.54%