

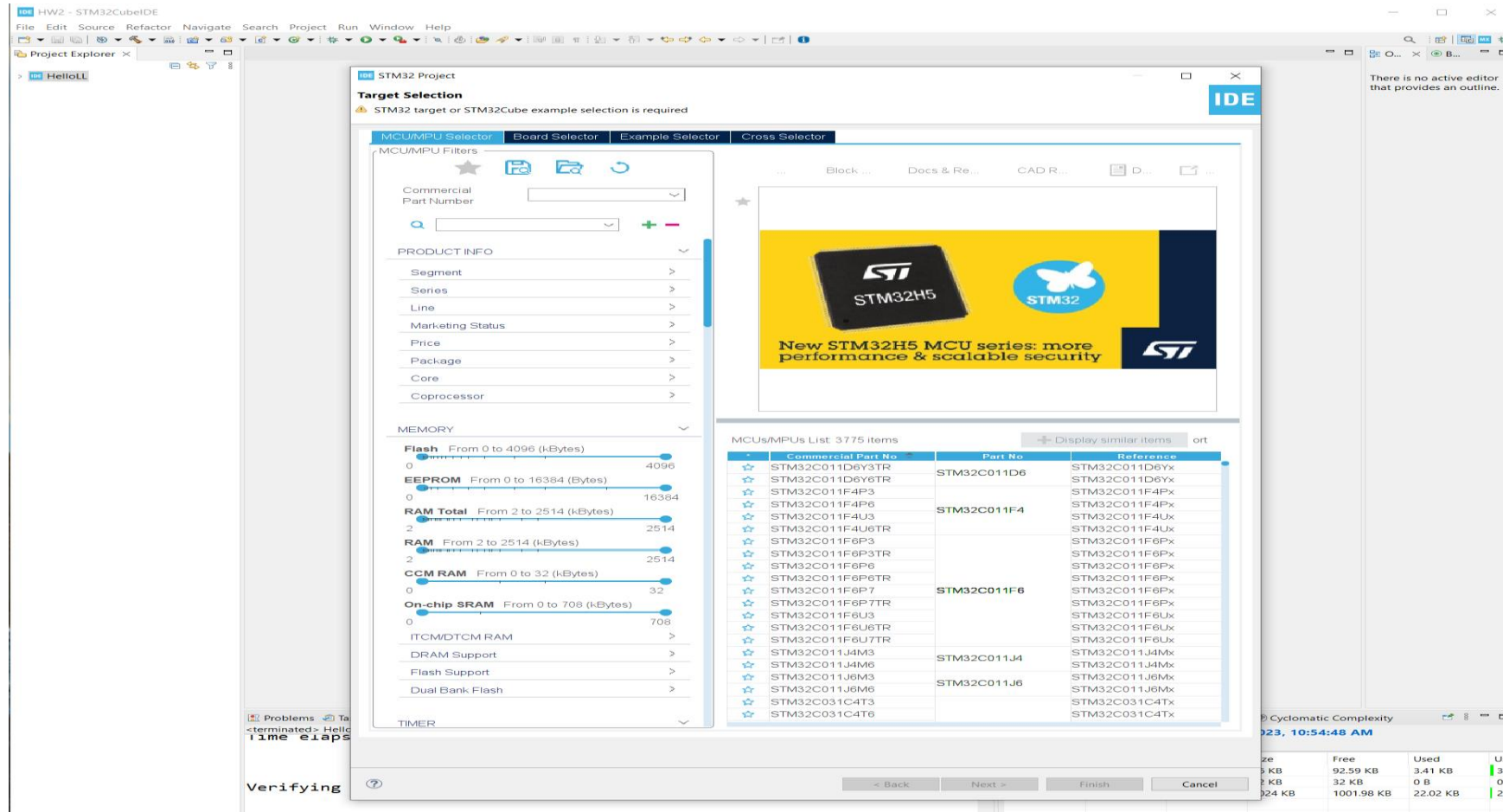
UCSD Embedded RTOS Assignment 5

By

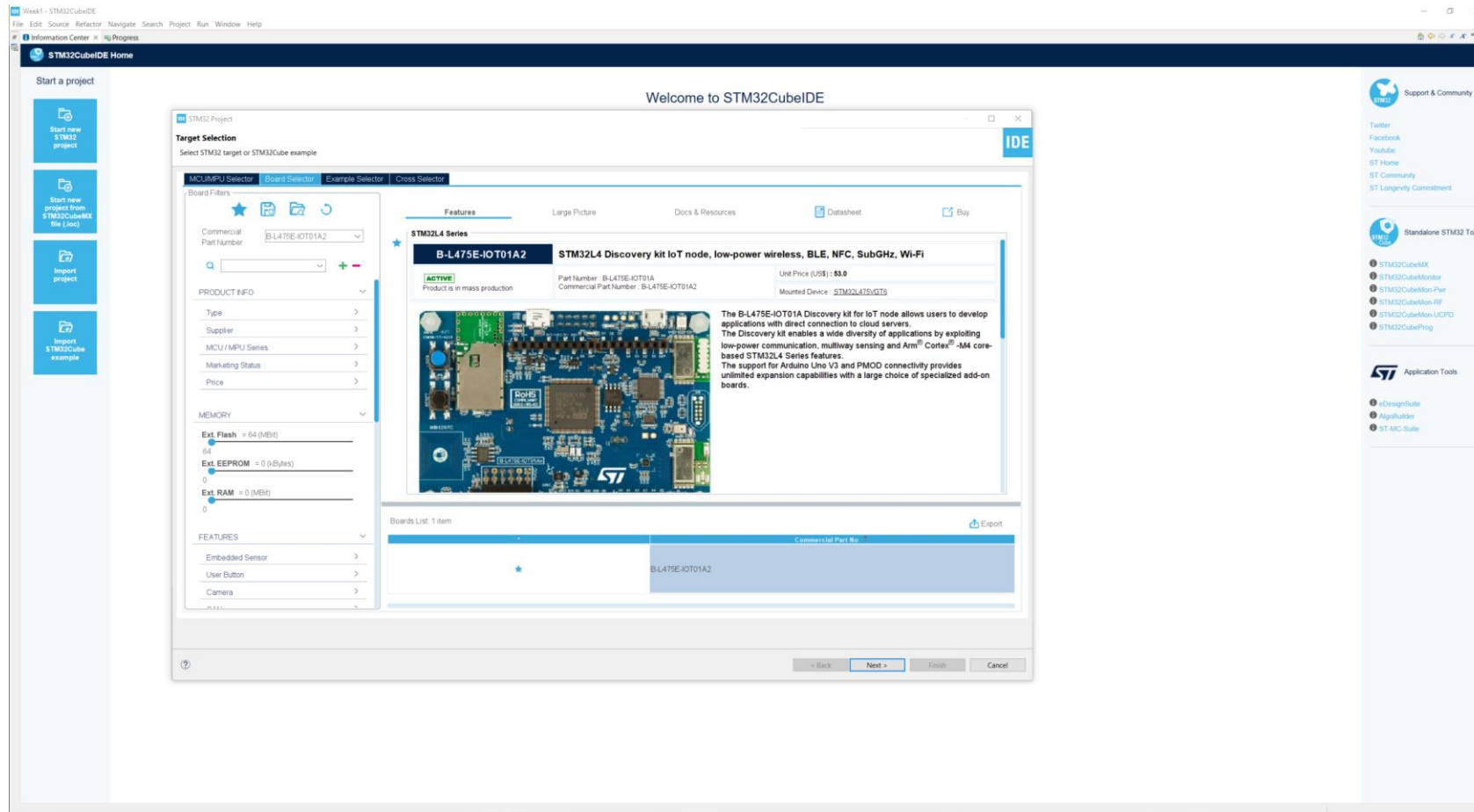
Hsuankai Chang

hsuankac@umich.edu

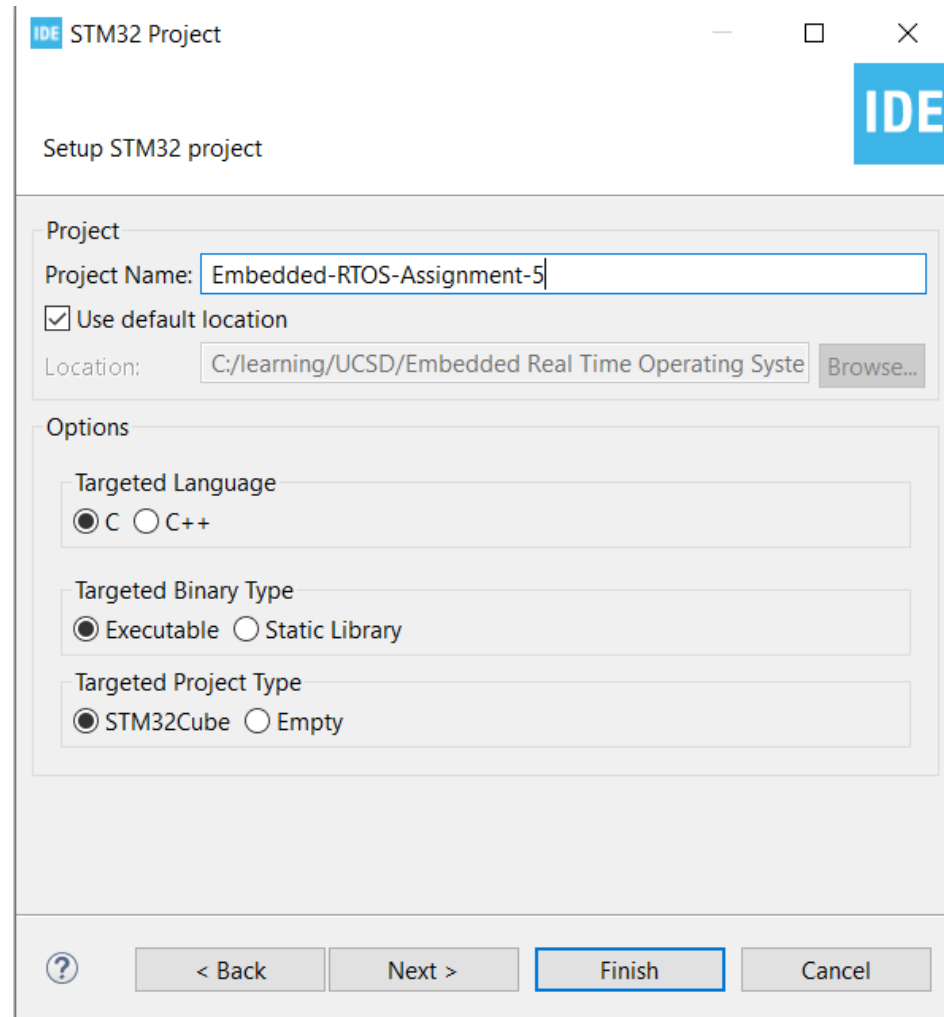
Step 1. Startup STM32CubeIDE and create new STM32 project



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 an IDE. The window title is 'IDE STM32 Project'. The main title is 'Setup STM32 project'. The 'Project' section contains a 'Project Name' field with the text 'Embedded-RTOS-Assignment-5' and a checked 'Use default location' checkbox. The 'Location' field shows 'C:/learning/UCSD/Embedded Real Time Operating System' with a 'Browse...' button. The 'Options' section has three groups: 'Targeted Language' with radio buttons for 'C' (selected) and 'C++'; 'Targeted Binary Type' with radio buttons for 'Executable' (selected) and 'Static Library'; and 'Targeted Project Type' with radio buttons for 'STM32Cube' (selected) and 'Empty'. At the bottom 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-5

☒ 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 screenshot of the 'Firmware Library Package Setup' dialog box in the STM32 Project IDE. The dialog has a title bar with the IDE logo and the text 'STM32 Project'. The main title is 'Firmware Library Package Setup' and the subtitle is '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 'Target and Firmware Package' section, the 'Target Reference' is 'B-L475E-IOT01A2' and the 'Firmware Package Name and Version' is 'STM32Cube FW_L4 V1.17.2'. In the 'Firmware and Software Package Repository' section, the 'Location' is 'C:\Users\hsuankai.chang\STM32Cube\Repository' and there is a link to 'Firmware Updater'. In the 'Code Generator Options' section, there are three radio buttons: 'Add necessary library files as reference in the toolchain project configuration file', 'Copy all used libraries into the project folder', and 'Copy only the necessary library files'. The 'Finish' button is highlighted with a blue border.

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

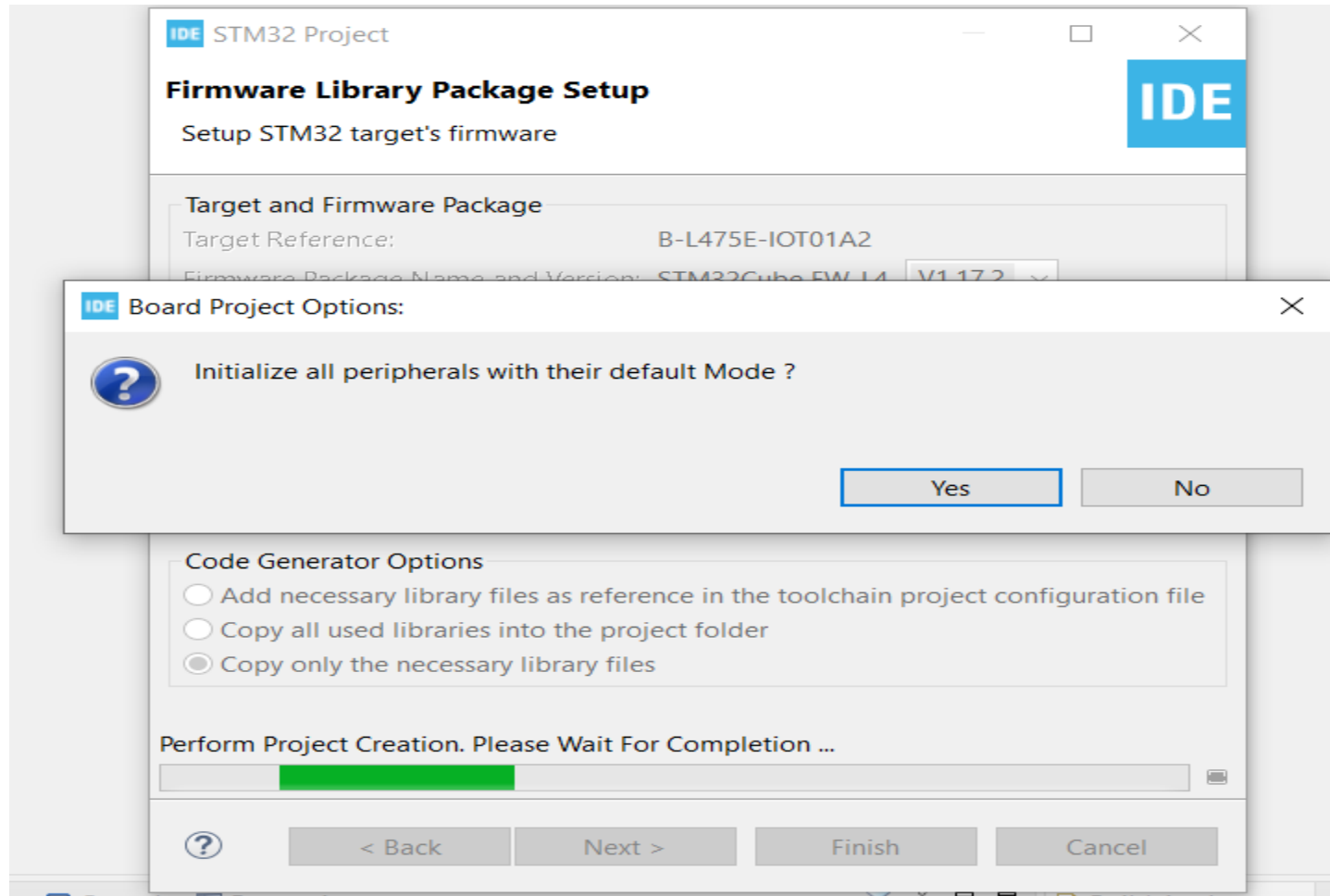
☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

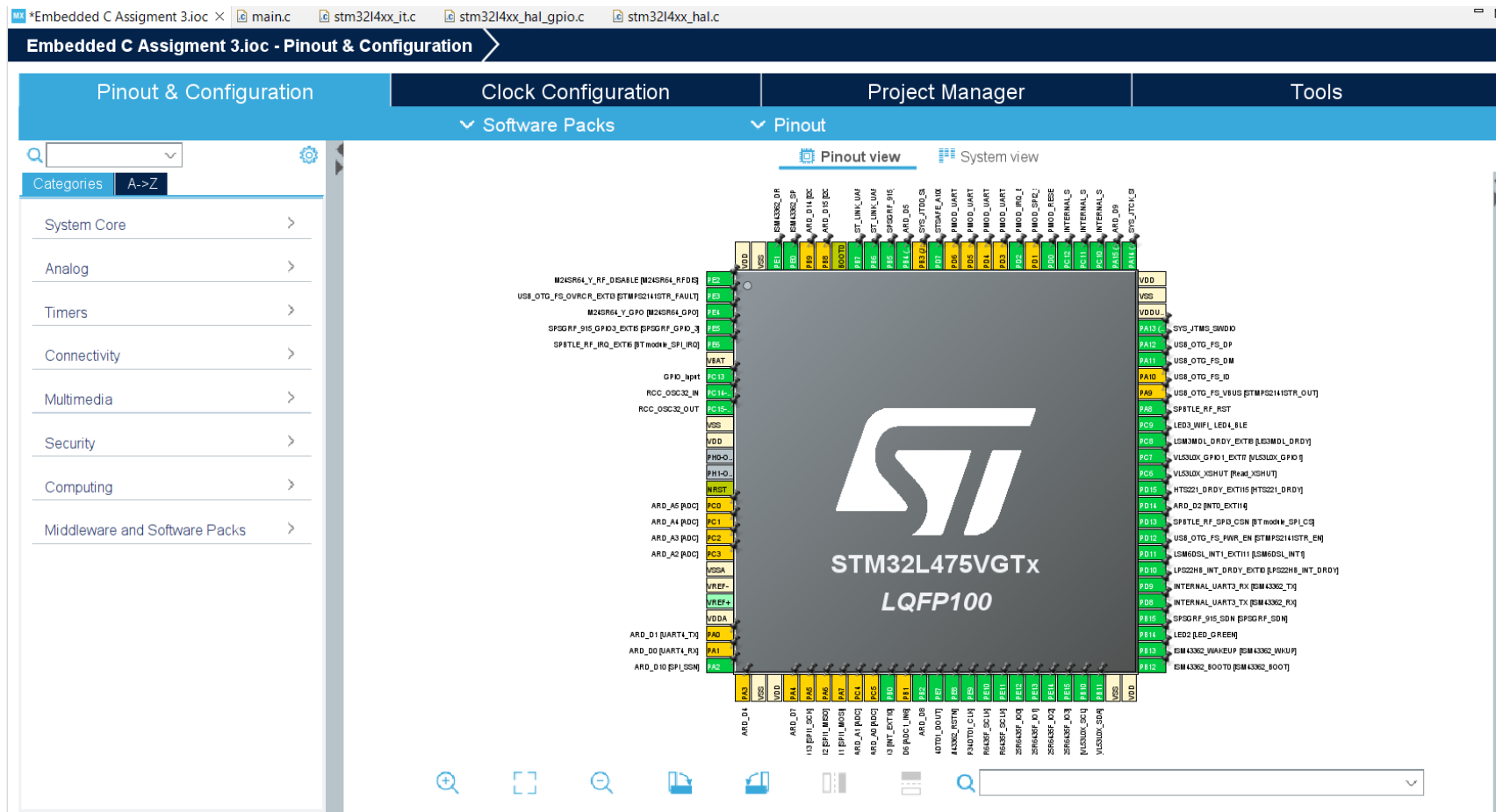
☒ Copy only the necessary library files

? < Back Next > Finish Cancel

Step 5. Click yes to initialize all peripherals to default



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



Step 7. Enable the CMSIS_V1 RTOS, and create sender, receiver tasks. Also create a queue with 5 uint32_t elements.

The screenshot shows the STM32CubeIDE Pinout & Configuration window. The left sidebar lists various categories, with 'Middleware and Software Packs' expanded to show a list of packs. 'FREERTOS' is selected and highlighted. The main area is titled 'FREERTOS Mode and Configuration' and shows the 'Interface' set to 'CMSIS_V1'. Below this, the 'Configuration' section has a 'Reset Configuration' button and several checked options: 'Tasks and Queues', 'Timers and Semaphores', 'Mutexes', 'Events', 'FreeRTOS Heap Usage', 'Config parameters', 'Include parameters', 'Advanced settings', and 'User Constants'. The 'Tasks' table lists two tasks: 'Task1' and 'Task2'. The 'Queues' table lists one queue: 'Queue01'.

Task Name	Priority	Stack Size (Words)	Entry Function	Code Generation	Parameter	Allocation	Buffer Name	Control Block Name
Task1	osPriorityNormal	128	SenderTask	Default	NULL	Dynamic	NULL	NULL
Task2	osPriorityNormal	128	ReceiverTask	Default	NULL	Dynamic	NULL	NULL

Queue Name	Queue Size	Item Size	Allocation	Buffer Name	Control Block Name
Queue01	5	uint32_t	Dynamic	NULL	NULL

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... >

Software Packs

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

Pinout view System view

STM32L475VGTX LQFP100

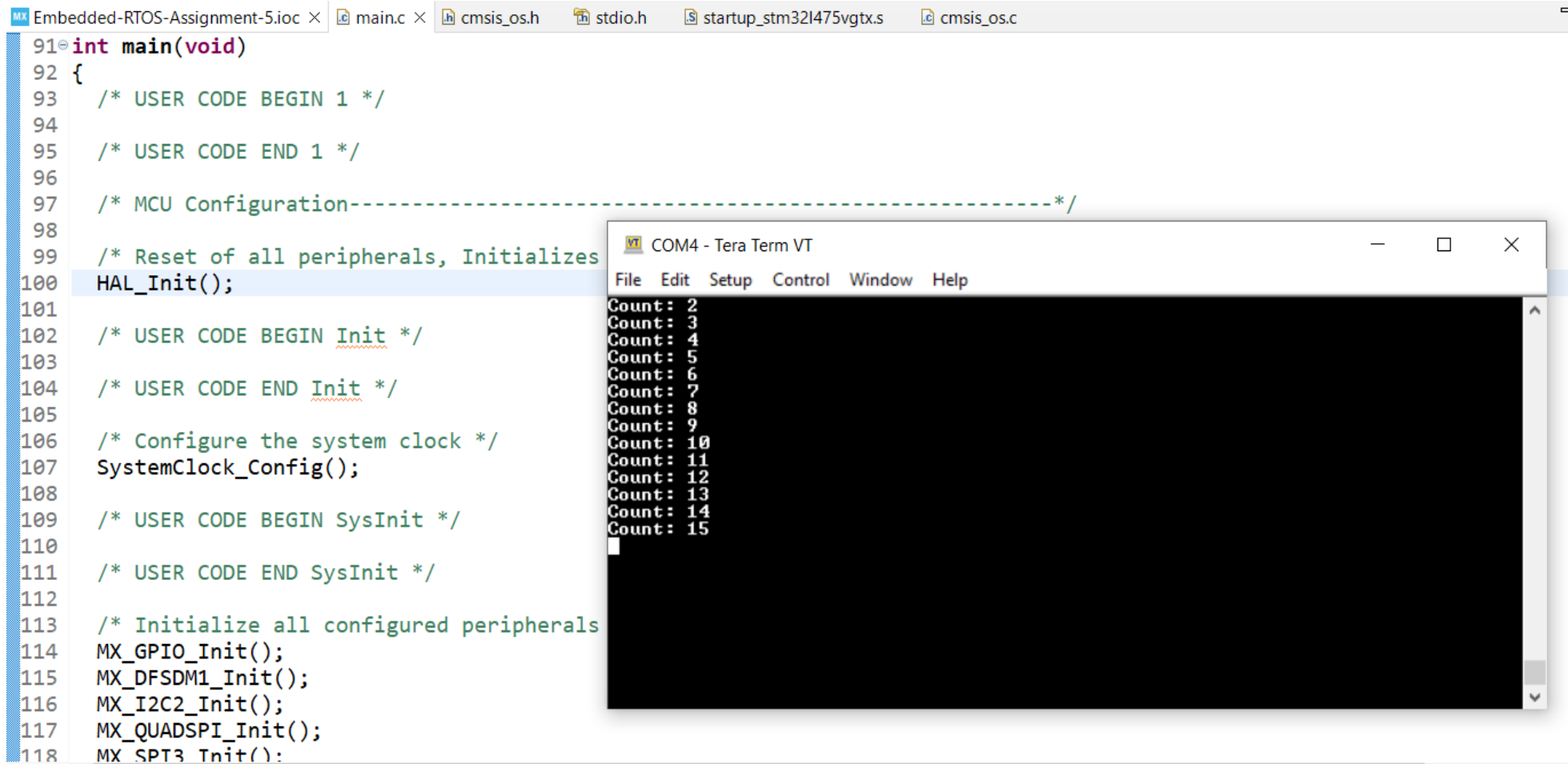
Step 9. Code the sender task, I use CMSIS-RTOS API

```
MX Embedded-RTOS-Assignment-5.ioc  main.c × cmsis_os.h stdio.h startup_stm32l4
694 /**
695  * @brief Function implementing the Task1 thread.
696  * @param argument: Not used
697  * @retval None
698  */
699 /* USER CODE END Header_SenderTask */
700 void SenderTask(void const * argument)
701 {
702     /* USER CODE BEGIN 5 */
703     static int count = 0;
704     /* Infinite loop */
705     for(;;)
706     {
707         osMessagePut(Queue01Handle, count, osWaitForever);
708         count++;
709         osDelay(2000);
710     }
711     /* USER CODE END 5 */
712 }
```

Step 10. Code the receiver task

```
MX Embedded-RTOS-Assignment-5.ioc  main.c × cmsis_os.h stdio.h startup_stm32l475vgtx.s cmsis_os.c
712 }
713
714 /* USER CODE BEGIN Header_ReceiverTask */
715 /**
716  * @brief Function implementing the Task2 thread.
717  * @param argument: Not used
718  * @retval None
719  */
720 /* USER CODE END Header_ReceiverTask */
721 void ReceiverTask(void const * argument)
722 {
723     /* USER CODE BEGIN ReceiverTask */
724     osEvent evt;
725     char buf[100];
726     /* Infinite loop */
727     for (;;) {
728         evt = osMessageGet(Queue01Handle, osWaitForever);
729         if (evt.status == osEventMessage) {
730             snprintf(buf, sizeof(buf), "Count: %d\n\r", (int)(evt.value.p));
731             HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), 100);
732         }
733     }
734     /* USER CODE END ReceiverTask */
735 }
```

Step 11. Compile and run the code, test is successful



The image shows a screenshot of an IDE with several open files: Embedded-RTOS-Assignment-5.ioc, main.c, cmsis_os.h, stdio.h, startup_stm32l475vgtx.s, and cmsis_os.c. The main.c file is active, showing C code for an embedded system. The code includes comments for user code and MCU configuration, and calls to HAL_Init(), SystemClock_Config(), and various peripheral initialization functions (MX_GPIO_Init, MX_DFSDM1_Init, MX_I2C2_Init, MX_QUADSPI_Init, MX_SPT3_Init). A terminal window titled 'COM4 - Tera Term VT' is overlaid on the code, displaying a series of 'Count' values from 2 to 15, indicating a successful test.

```
91 int main(void)
92 {
93     /* USER CODE BEGIN 1 */
94
95     /* USER CODE END 1 */
96
97     /* MCU Configuration-----*/
98
99     /* Reset of all peripherals, Initializes
100    HAL_Init();
101
102     /* USER CODE BEGIN Init */
103
104     /* USER CODE END Init */
105
106     /* Configure the system clock */
107    SystemClock_Config();
108
109     /* USER CODE BEGIN SysInit */
110
111     /* USER CODE END SysInit */
112
113     /* Initialize all configured peripherals
114    MX_GPIO_Init();
115    MX_DFSDM1_Init();
116    MX_I2C2_Init();
117    MX_QUADSPI_Init();
118    MX_SPT3_Init();
```

COM4 - Tera Term VT

File Edit Setup Control Window Help

Count: 2
Count: 3
Count: 4
Count: 5
Count: 6
Count: 7
Count: 8
Count: 9
Count: 10
Count: 11
Count: 12
Count: 13
Count: 14
Count: 15