

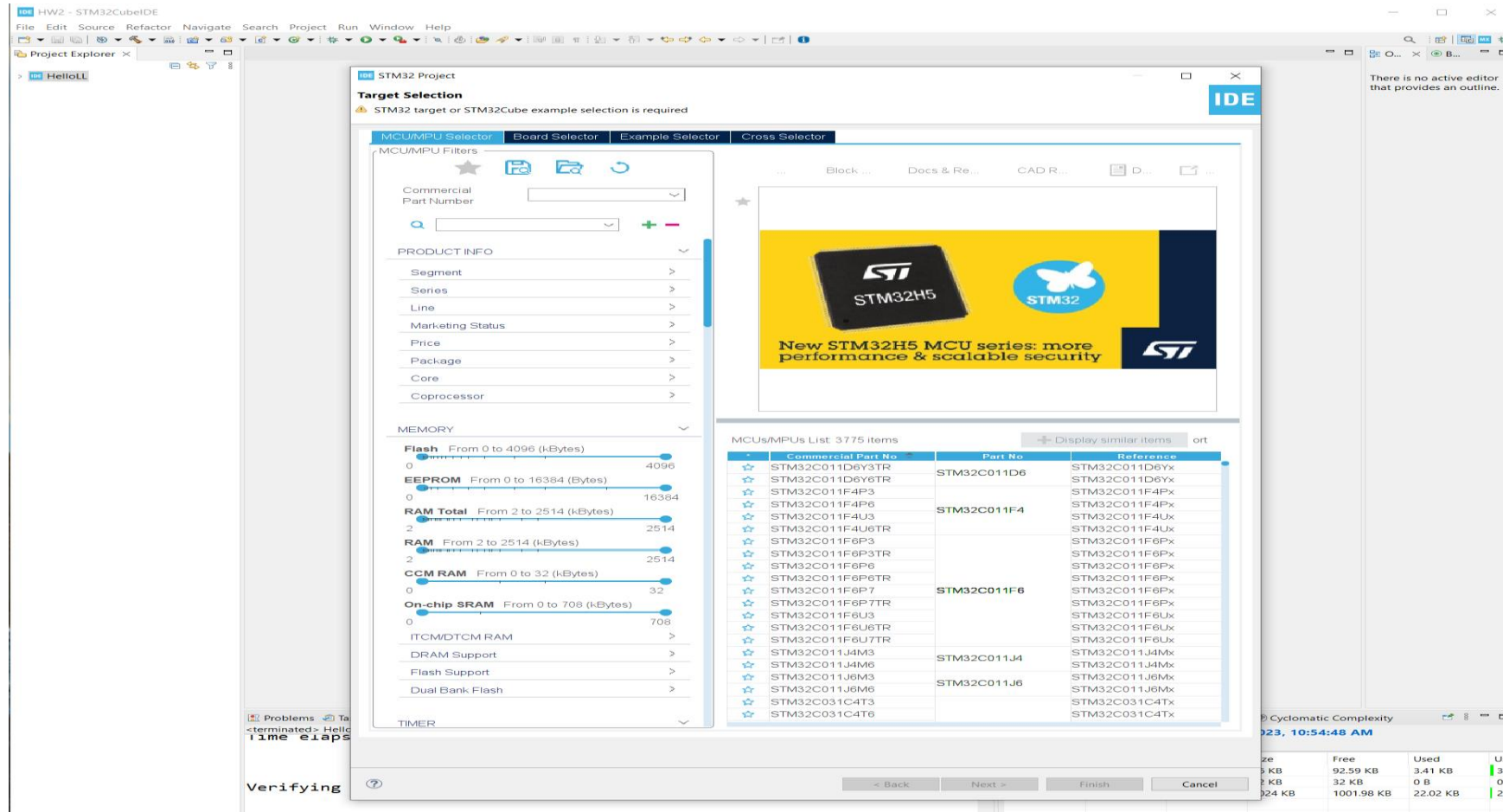
UCSD Embedded RTOS Assignment 6

By

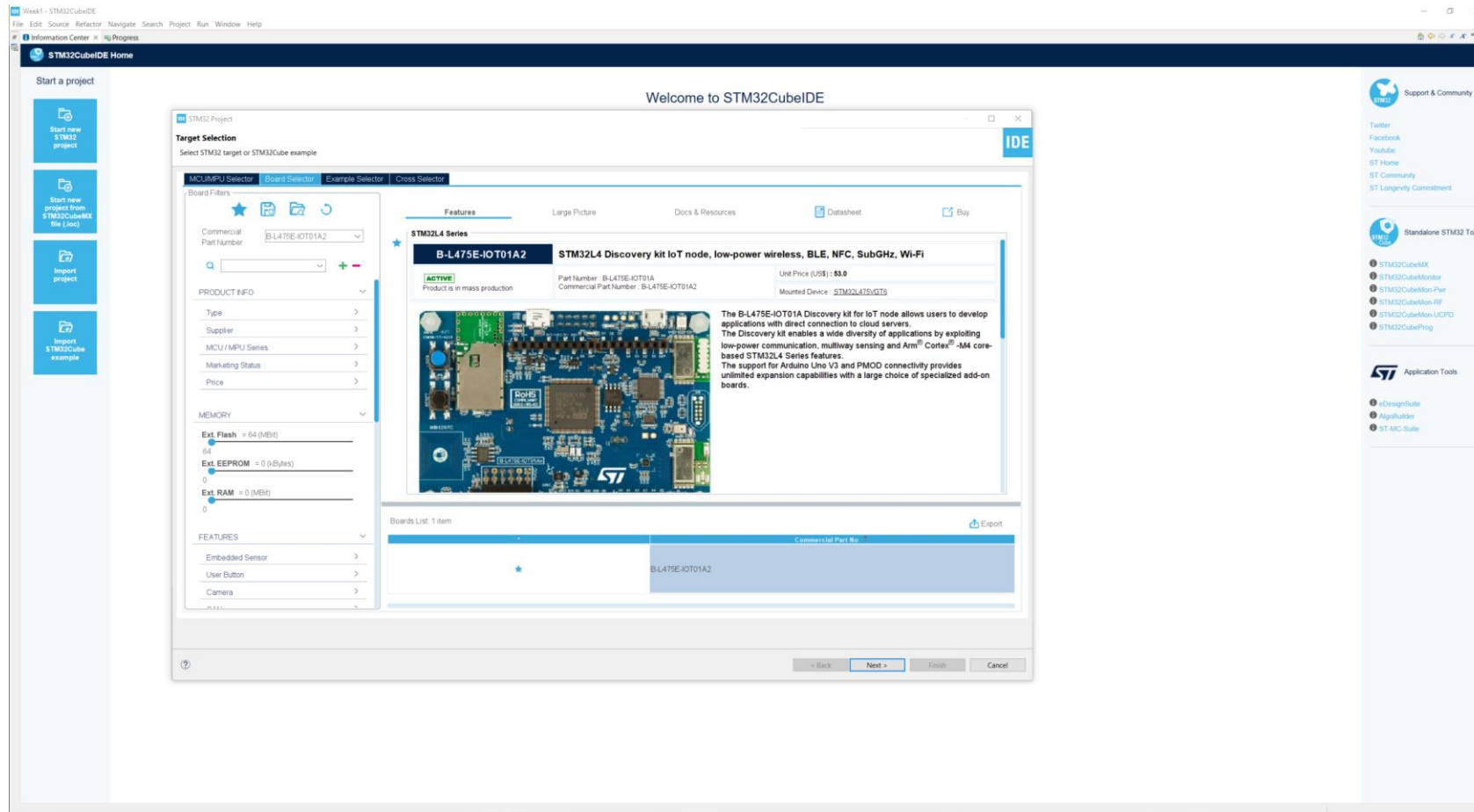
Hsuankai Chang

hsuankac@umich.edu

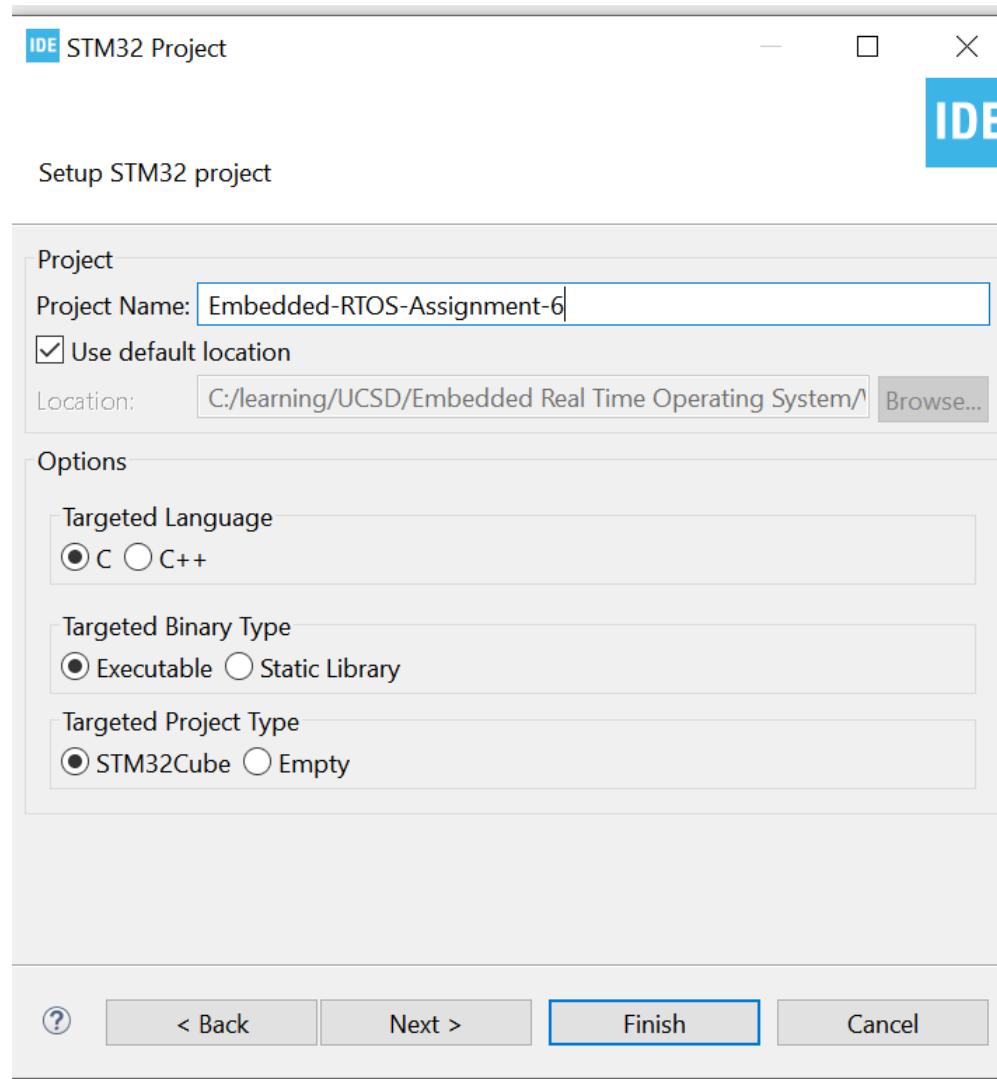
Step 1. Startup STM32CubeIDE and create new STM32 project



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 dialog is titled 'Setup STM32 project'. It has two main sections: 'Project' and 'Options'. In the 'Project' section, the 'Project Name' field contains 'Embedded-RTOS-Assignment-6'. The 'Use default location' checkbox is checked. The 'Location' field shows 'C:/learning/UCSD/Embedded Real Time Operating System/' with a 'Browse...' button. The 'Options' section has three groups of radio buttons: 'Targeted Language' with 'C' selected, 'Targeted Binary Type' with 'Executable' selected, and 'Targeted Project Type' with 'STM32Cube' selected. At the bottom, there are buttons for '?', '< Back', 'Next >', 'Finish' (which is highlighted with a blue border), and 'Cancel'.

IDE STM32 Project

Setup STM32 project

Project

Project Name: Embedded-RTOS-Assignment-6

☒ 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

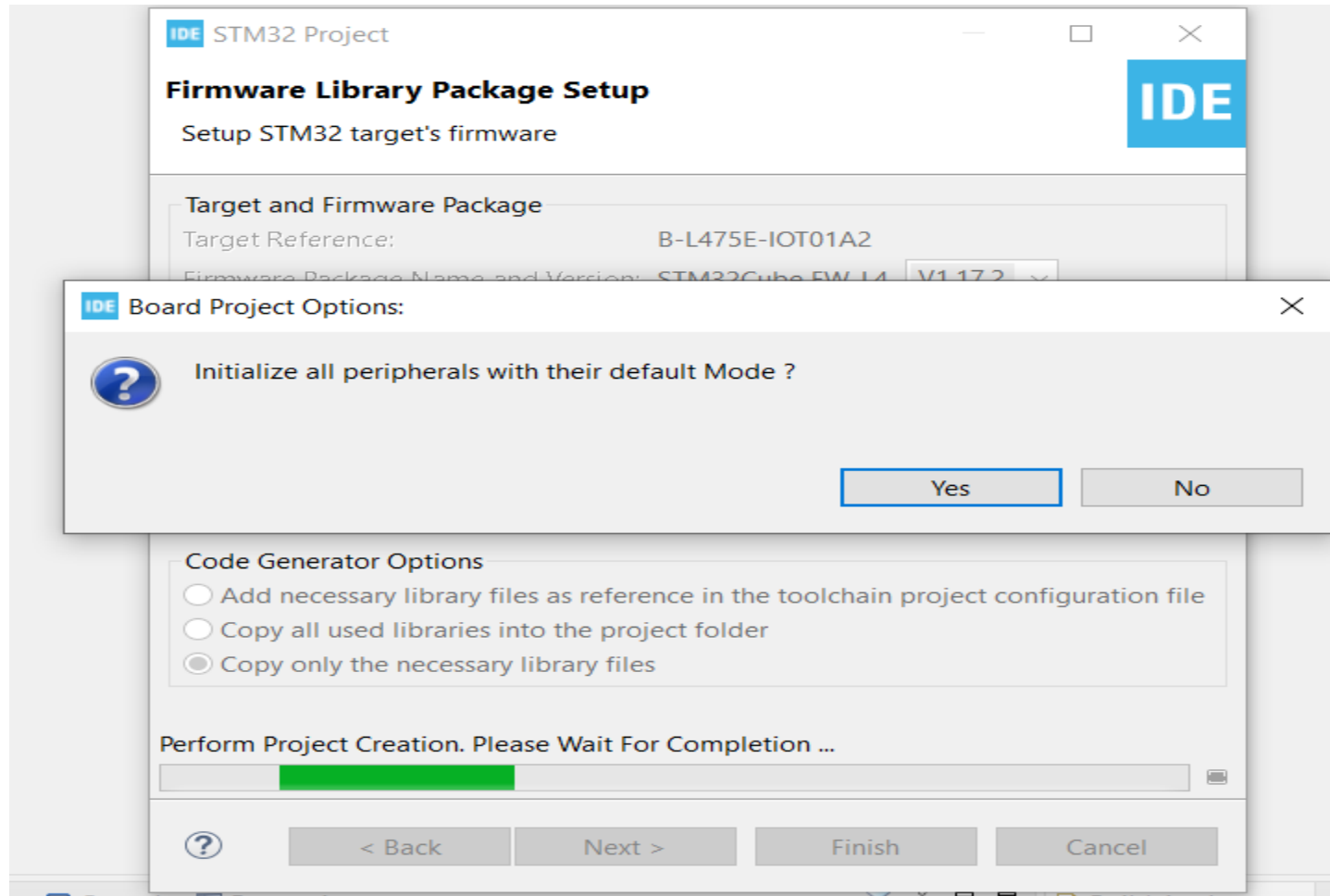
☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

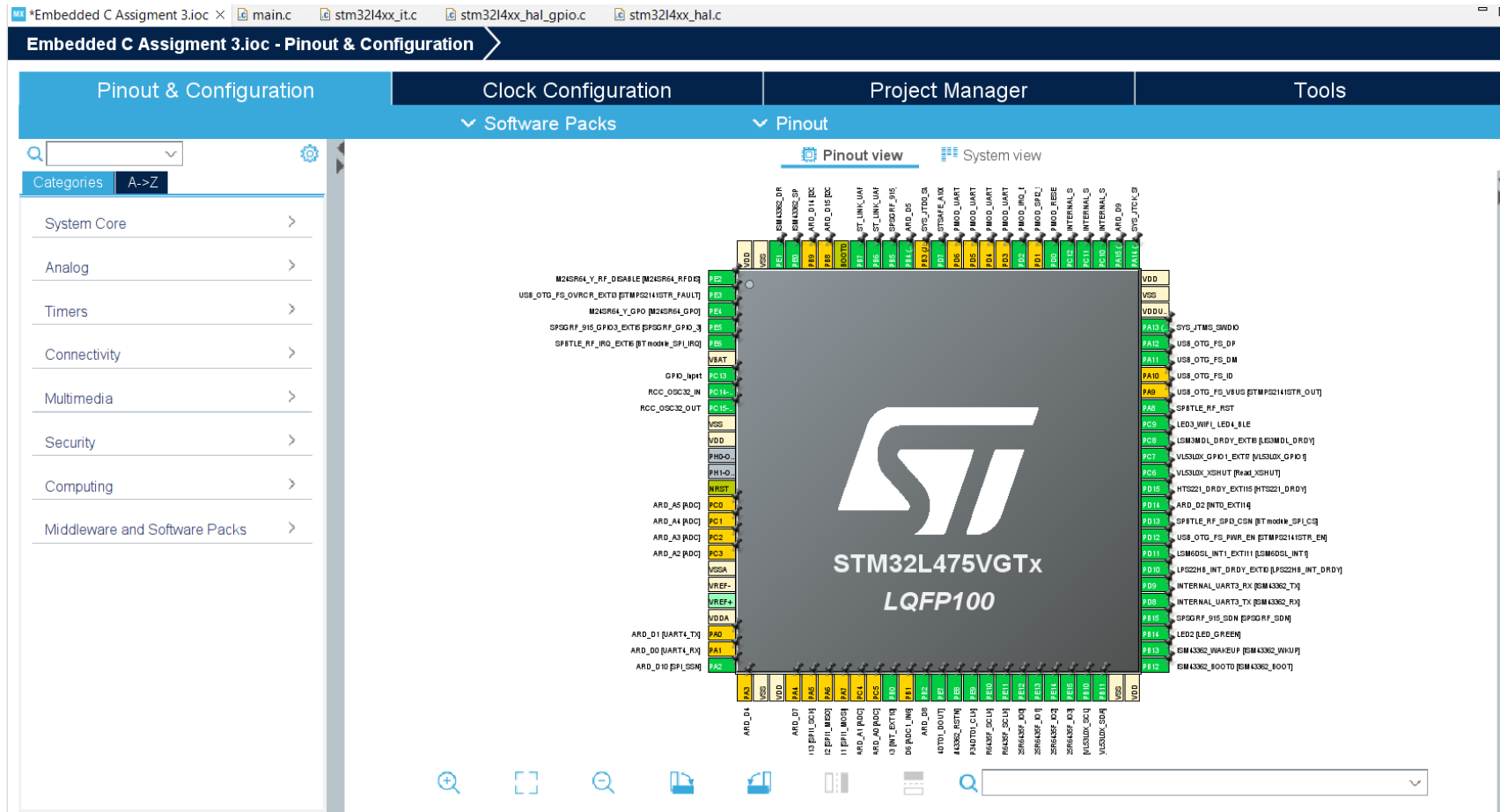
☒ Copy only the necessary library files

? < Back Next > Finish Cancel

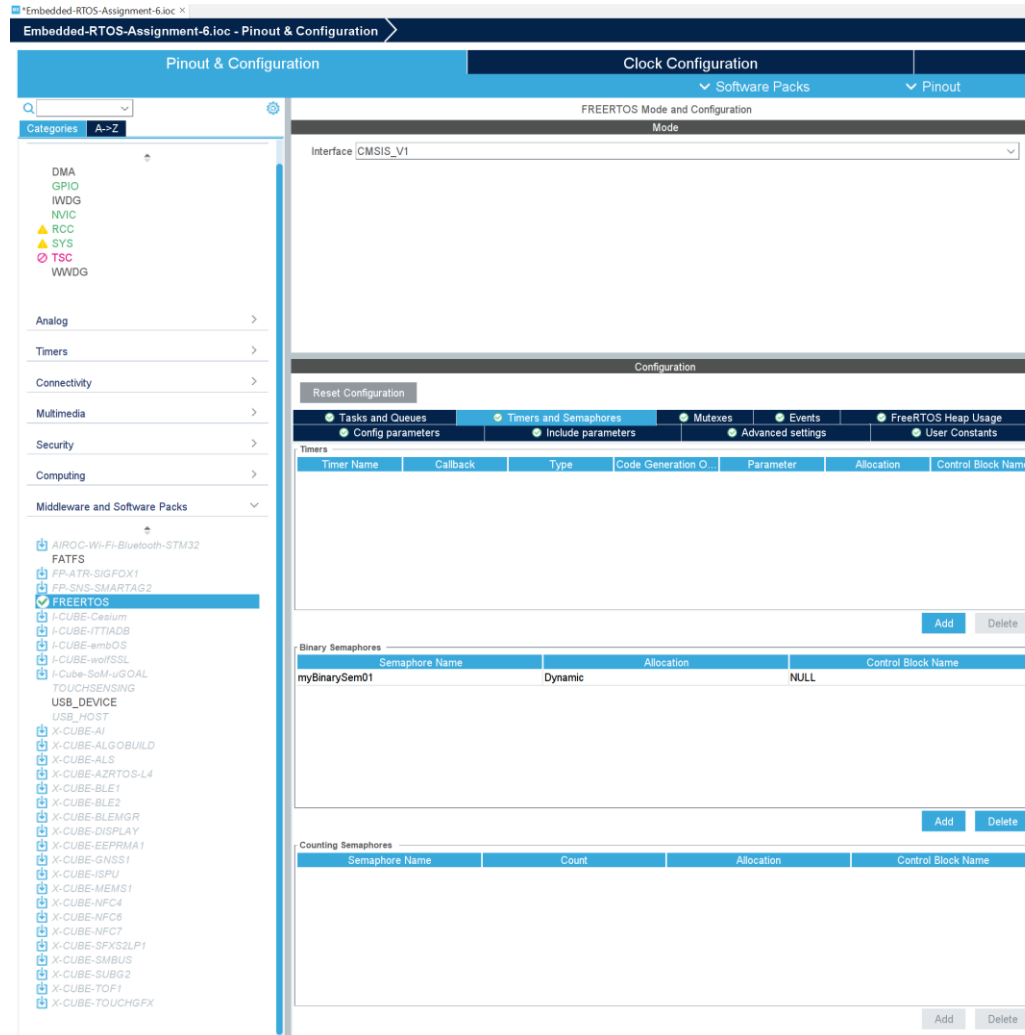
Step 5. Click yes to initialize all peripherals to default



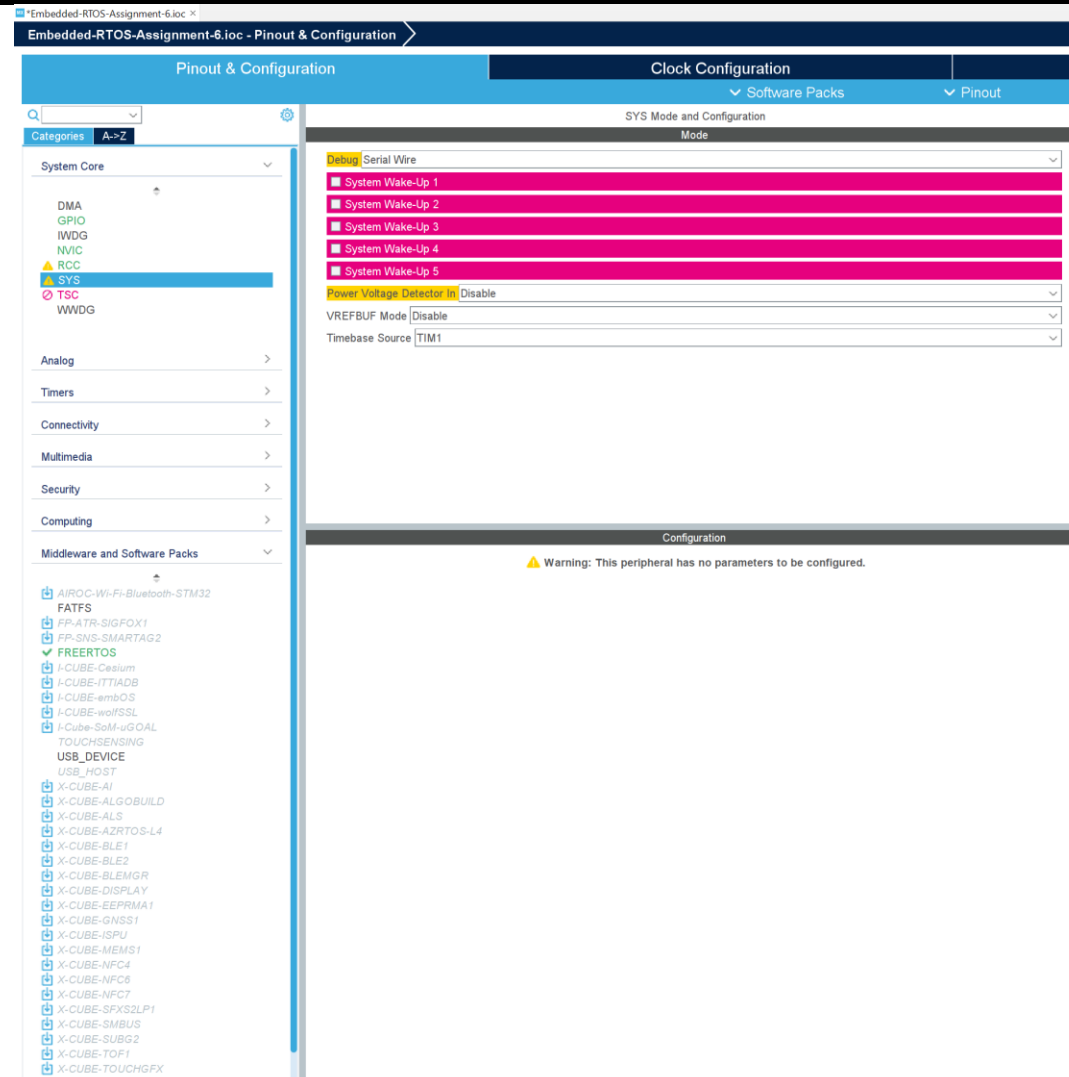
Step 6. When in .ioc file, click Pinout & Configurations



Step 7. Enable FreeRTOS CMSIS_V1 and add binary semaphore.



Step 8. Change Timebase from systick to TIM1



Step 9. Add release semaphore code in interrupt source file, don't forget to include necessary header file for osSemaphoreId

```
Embedded-RTOS-Assignment-6.ioc  stm32l4xx_it.c ×  main.c
28 /* USER CODE BEGIN TD */
29
30 /* USER CODE END TD */
31
32 /* Private define -----
33 /* USER CODE BEGIN PD */
34
35 /* USER CODE END PD */
36
37 /* Private macro -----
38 /* USER CODE BEGIN PM */
39
40 /* USER CODE END PM */
41
42 /* Private variables -----
43 /* USER CODE BEGIN PV */
44
45 /* USER CODE END PV */
46
47 /* Private function prototypes -----
48 /* USER CODE BEGIN PFP */
49
50 /* USER CODE END PFP */
51
52 /* Private user code -----
53 /* USER CODE BEGIN 0 */
54 extern osSemaphoreId myBinarySem01Handle;
55 /* USER CODE END 0 */
56
57 /* External variables -----
58 extern TIM_HandleTypeDef htim1;
```

```
Embedded-RTOS-Assignment-6.ioc  stm32l4xx_it.c ×  main.c
150 /* USER CODE BEGIN DebugMonitor_IRQn 1 */
151
152 /* USER CODE END DebugMonitor_IRQn 1 */
153 }
154
155 /*-----
156 /* STM32L4xx Peripheral Interrupt Handlers
157 /* Add here the Interrupt Handlers for the used peripherals.
158 /* For the available peripheral interrupt handler names,
159 /* please refer to the startup file (startup_stm32l4xx.s).
160 /*-----
161
162 /**
163  * @brief This function handles EXTI line[9:5] interrupts.
164  */
165 void EXTI9_5_IRQHandler(void)
166 {
167     /* USER CODE BEGIN EXTI9_5_IRQn 0 */
168
169     /* USER CODE END EXTI9_5_IRQn 0 */
170     HAL_GPIO_EXTI_IRQHandler(SPSGRF_915_GPIO3_EXTI5_Pin);
171     HAL_GPIO_EXTI_IRQHandler(SPTBLE_RF_IRQ_EXTI6_Pin);
172     HAL_GPIO_EXTI_IRQHandler(VL53L0X_GPIO1_EXTI7_Pin);
173     HAL_GPIO_EXTI_IRQHandler(LSM3MDL_DRDY_EXTI8_Pin);
174     /* USER CODE BEGIN EXTI9_5_IRQn 1 */
175
176     /* USER CODE END EXTI9_5_IRQn 1 */
177 }
178
179 /**
180  * @brief This function handles TIM1 update interrupt and TIM16 global interrupt.
181  */
182 void TIM1_UP_TIM16_IRQHandler(void)
183 {
184     /* USER CODE BEGIN TIM1_UP_TIM16_IRQn 0 */
185
186     /* USER CODE END TIM1_UP_TIM16_IRQn 0 */
187     HAL_TIM_IRQHandler(&htim1);
188     /* USER CODE BEGIN TIM1_UP_TIM16_IRQn 1 */
189
190     /* USER CODE END TIM1_UP_TIM16_IRQn 1 */
191 }
192
193 /**
194  * @brief This function handles EXTI line[15:10] interrupts.
195  */
196 void EXTI15_10_IRQHandler(void)
197 {
198     /* USER CODE BEGIN EXTI15_10_IRQn 0 */
199     osSemaphoreRelease(myBinarySem01Handle);
200     /* USER CODE END EXTI15_10_IRQn 0 */
201     HAL_GPIO_EXTI_IRQHandler(LPS22HB_INT_DRDY_EXTI0_Pin);
202     HAL_GPIO_EXTI_IRQHandler(LSM6DSL_INT1_EXTI11_Pin);
203     HAL_GPIO_EXTI_IRQHandler(BUTTON_EXTI13_Pin);
204     HAL_GPIO_EXTI_IRQHandler(ARD_D2_Pin);
205     HAL_GPIO_EXTI_IRQHandler(HTS221_DRDY_EXTI15_Pin);
206     /* USER CODE BEGIN EXTI15_10_IRQn 1 */
207
208     /* USER CODE END EXTI15_10_IRQn 1 */
209 }
210
211 /* USER CODE BEGIN 1 */
212
```

Step 10. In task function, get the semaphore and toggle LED2

```
Embedded-RTOS-Assignment-6.ioc  stm32l4xx_it.c  main.c x
672 HAL_NVIC_SetPriority(EXTI9_5_IRQn, 5, 0);
673 HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
674
675 HAL_NVIC_SetPriority(EXTI15_10_IRQn, 5, 0);
676 HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
677
678 /* USER CODE BEGIN MX_GPIO_Init_2 */
679 /* USER CODE END MX_GPIO_Init_2 */
680 }
681
682 /* USER CODE BEGIN 4 */
683
684 /* USER CODE END 4 */
685
686 /* USER CODE BEGIN Header_StartDefaultTask */
687 /**
688  * @brief Function implementing the defaultTask thread.
689  * @param argument: Not used
690  * @retval None
691  */
692 /* USER CODE END Header_StartDefaultTask */
693 void StartDefaultTask(void const * argument)
694 {
695     /* USER CODE BEGIN 5 */
696     /* Infinite loop */
697     for(;;)
698     {
699         osSemaphoreWait(myBinarySem01Handle, osWaitForever);
700         HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
701     }
702     /* USER CODE END 5 */
703 }
704
705 /**
706  * @brief Period elapsed callback in non blocking mode
707  * @param hrtimer_t * hrtimer: pointer to the current timer handle
```

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

