

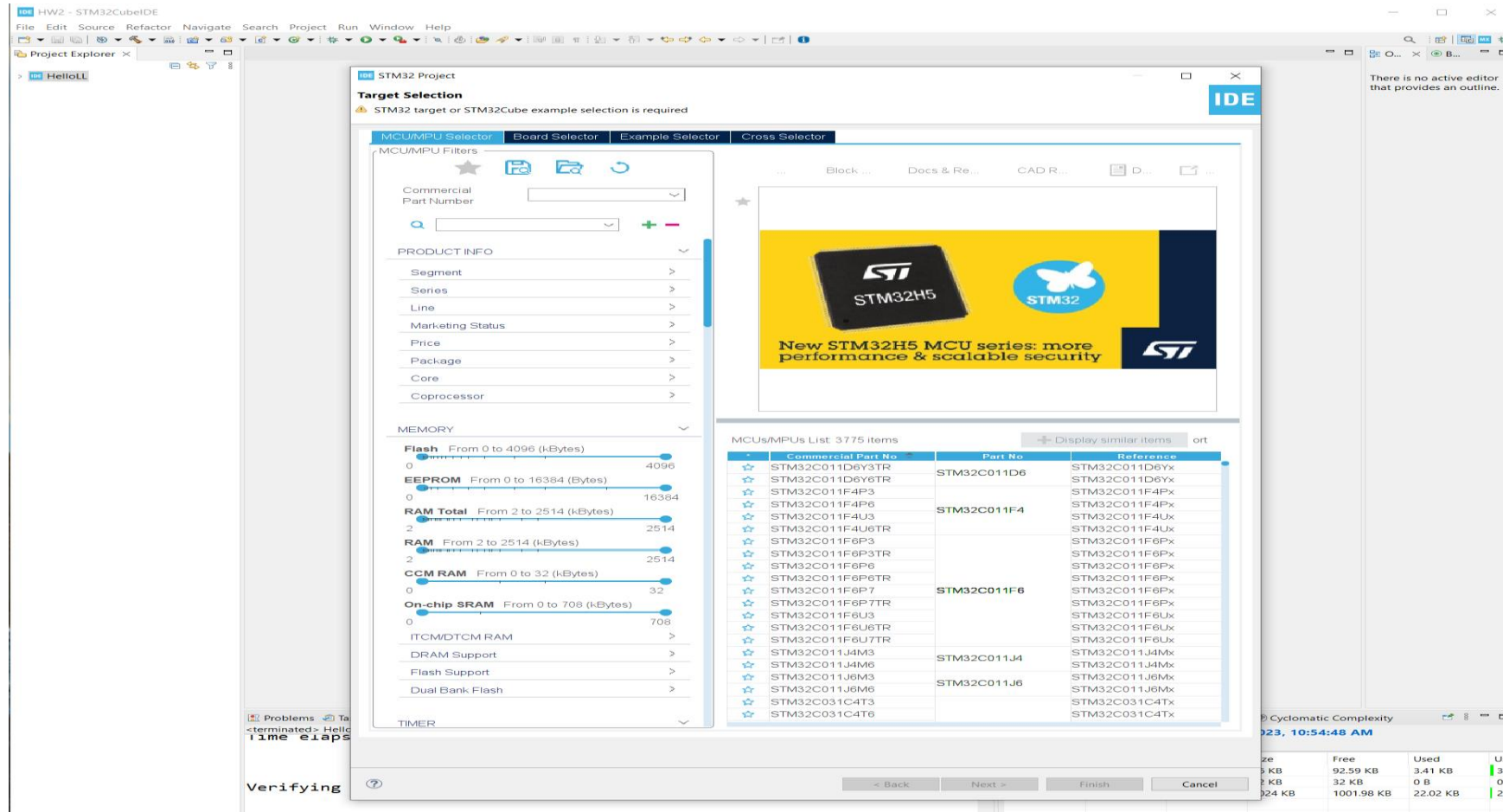
UCSD Embedded RTOS Assignment 7

By

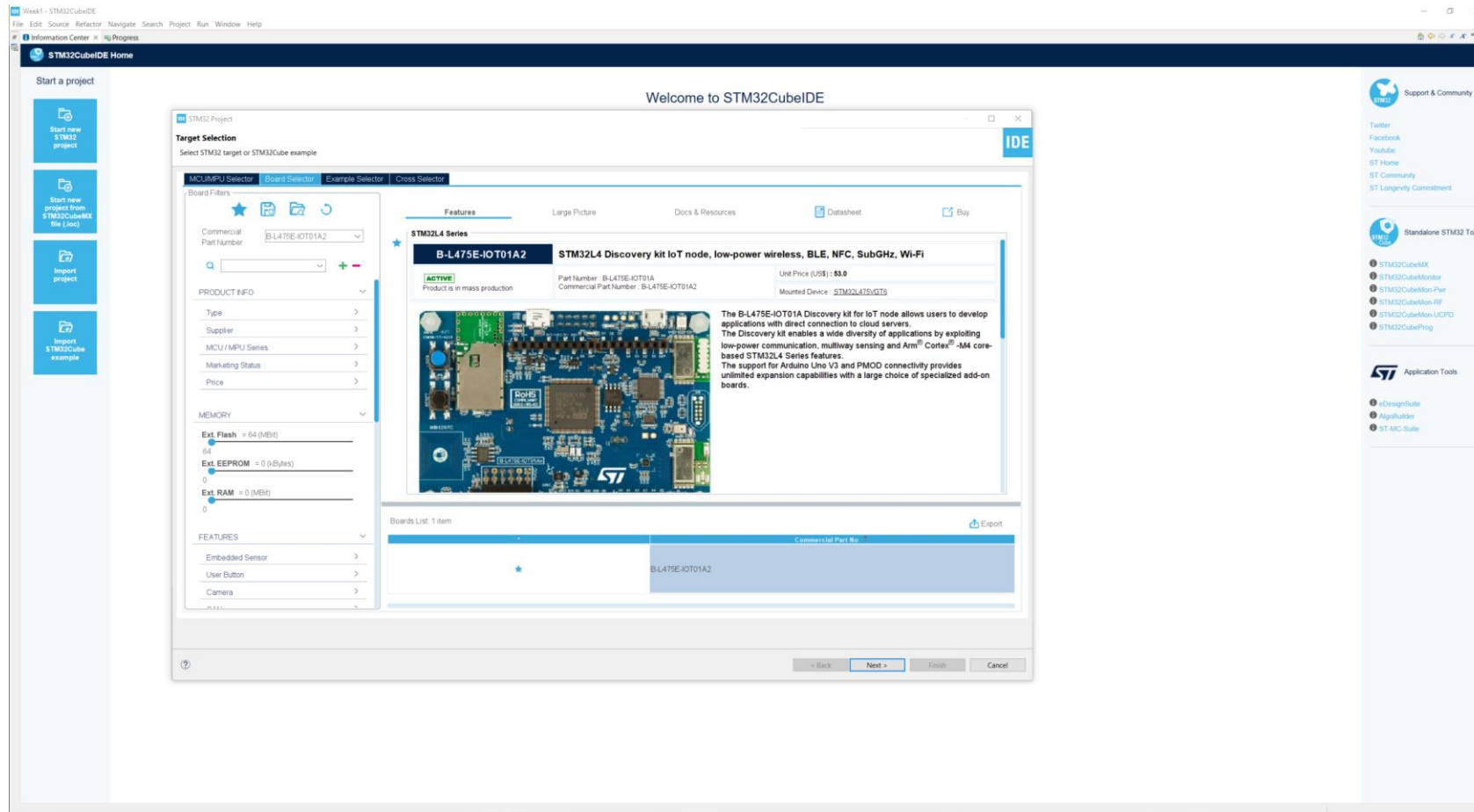
Hsuankai Chang

hsuankac@umich.edu

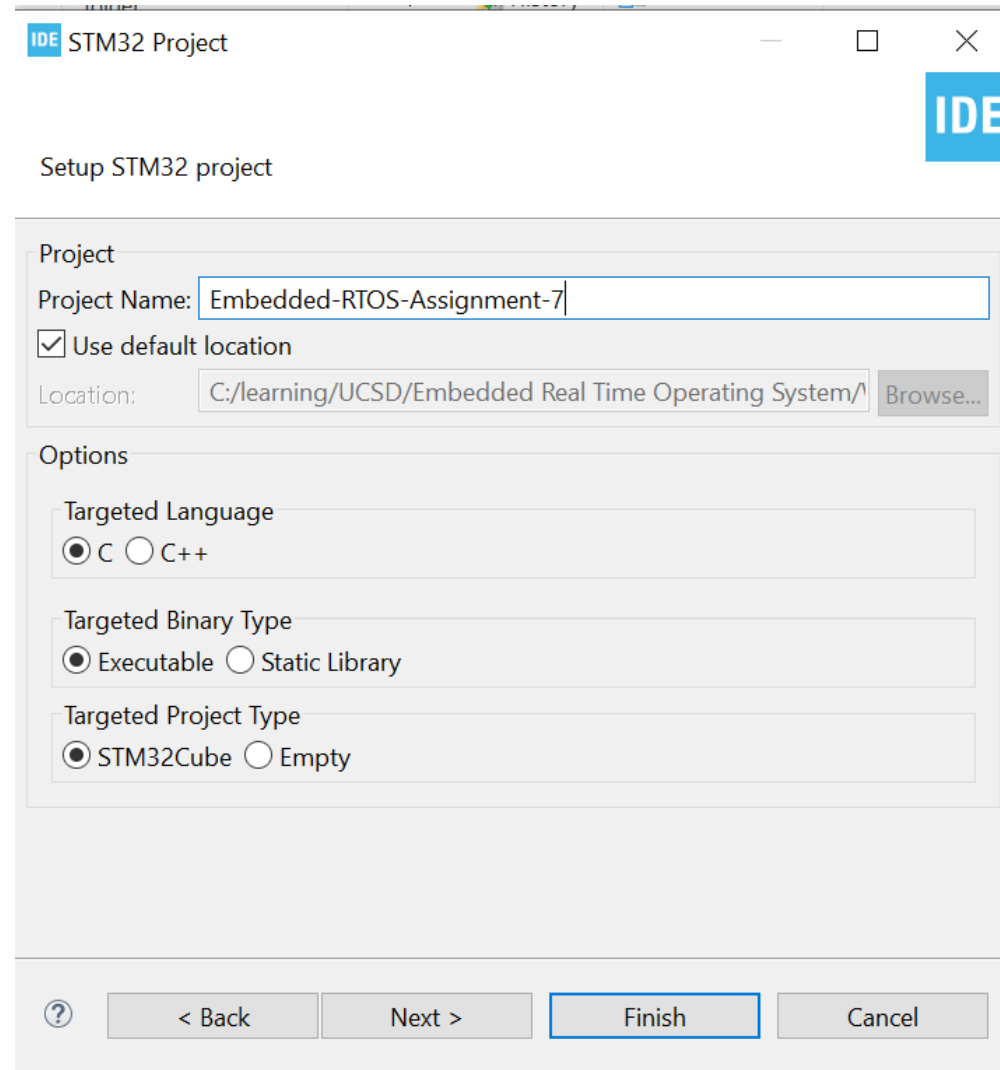
Step 1. Startup STM32CubeIDE and create new STM32 project



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 heading is 'Setup STM32 project'. Under the 'Project' section, the 'Project Name' field contains 'Embedded-RTOS-Assignment-7'. 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: 'Targeted Language' with 'C' selected, 'Targeted Binary Type' with 'Executable' selected, and 'Targeted Project Type' with 'STM32Cube' selected. 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-7

☒ 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 first section, 'Target Reference' is 'B-L475E-IOT01A2' and 'Firmware Package Name and Version' is 'STM32Cube FW_L4 V1.17.2'. In the second section, 'Location' is 'C:\Users\hsuankai.chang\STM32Cube\Repository' and there is a link to 'Firmware Updater'. In the third section, there are three radio button options for code generation, with 'Copy only the necessary library files' 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

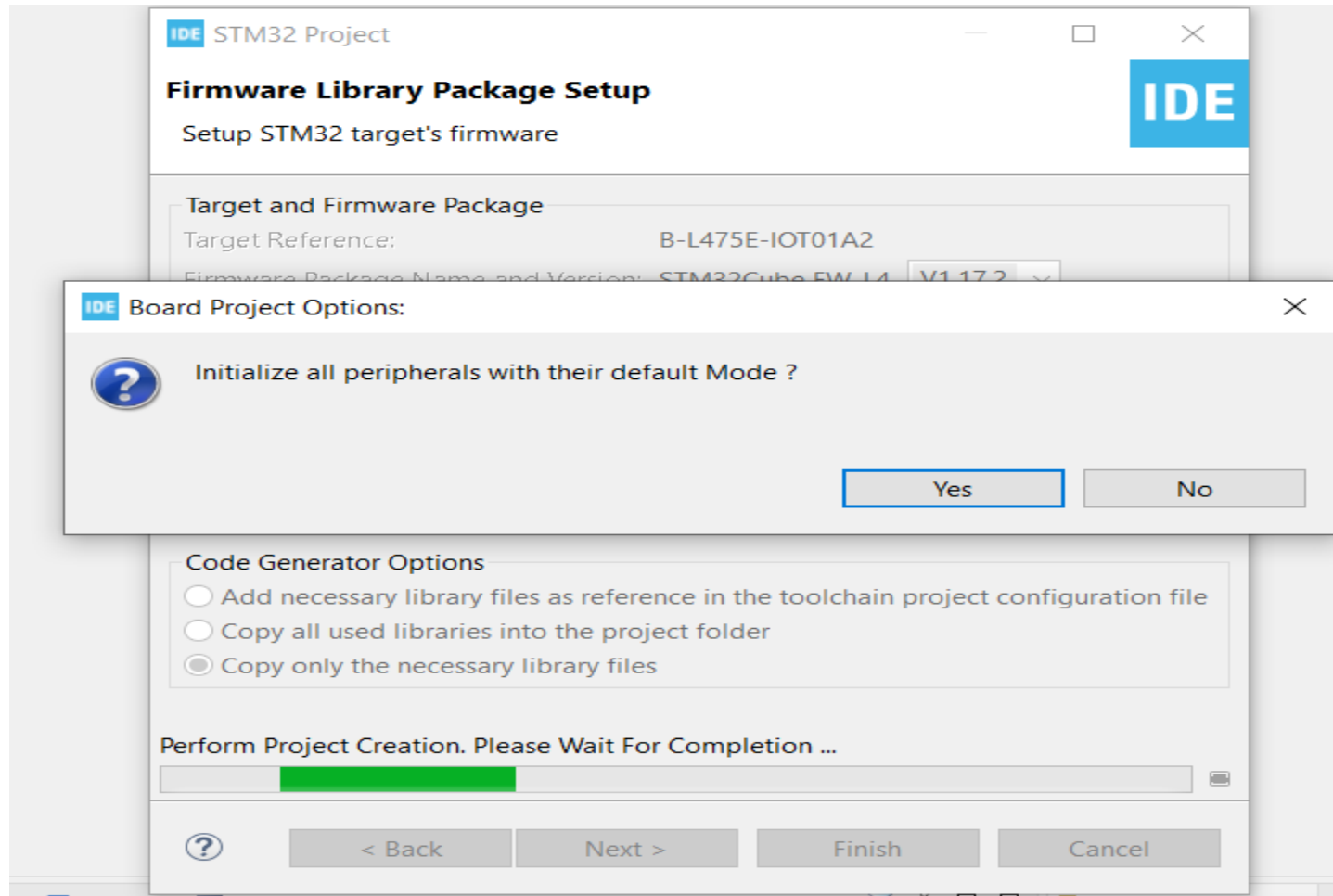
☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

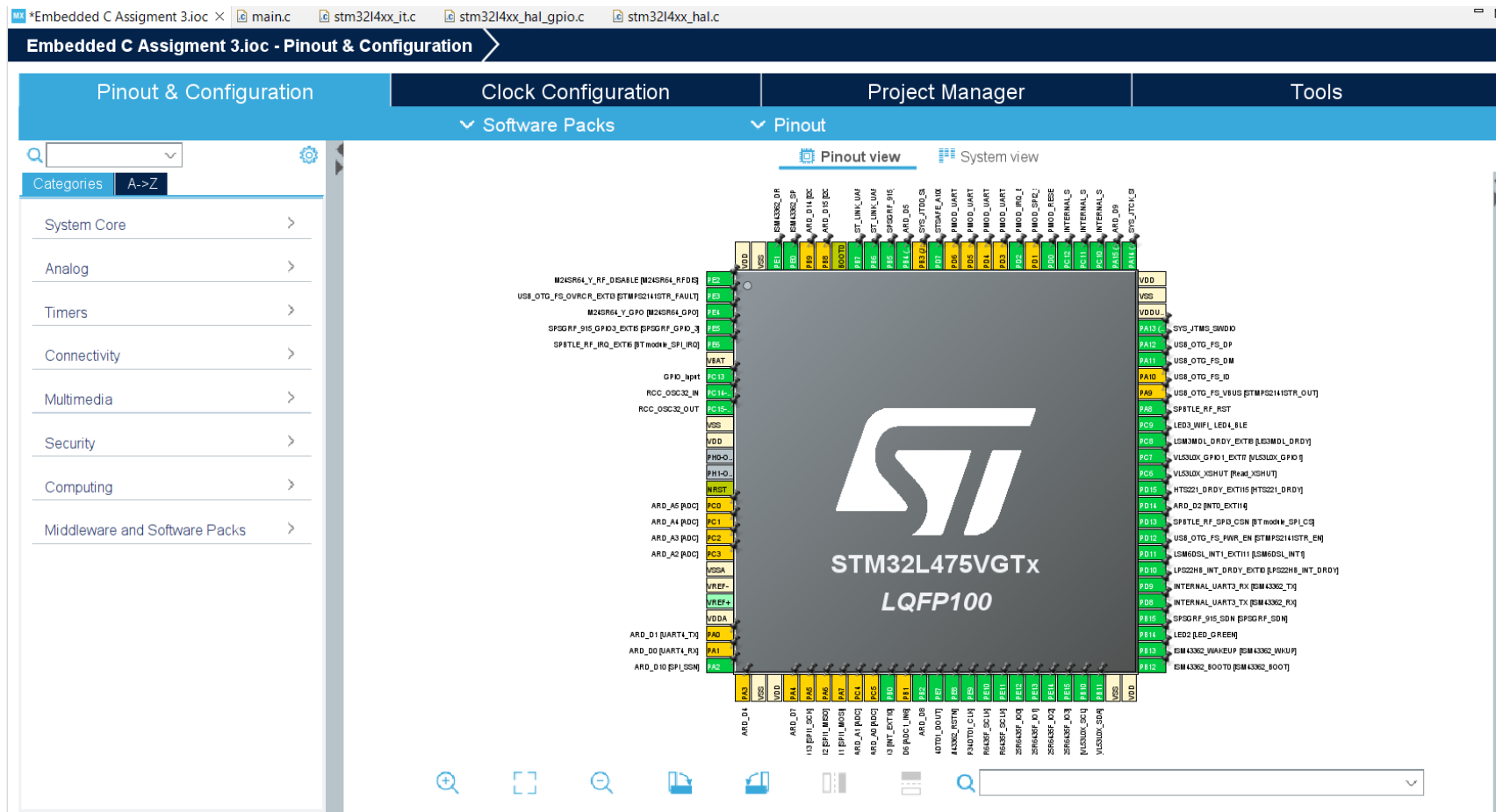
☒ Copy only the necessary library files

? < Back Next > Finish Cancel

Step 5. Click yes to initialize all peripherals to default



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



Step 7. Enable FreeRTOS CMSIS_V1 and add another task

The screenshot displays the STM32CubeIDE interface for configuring a project. The main window is titled "Embedded-RTOS-Assignment-7.ioc - Pinout & Configuration". The left sidebar shows the "Middleware and Software Packs" section, where "FREERTOS" is selected. The right pane shows the "FREERTOS Mode and Configuration" dialog, which is divided into two tabs: "Mode" and "Configuration".

In the "Mode" tab, the "Interface" is set to "CMSIS_V1". The "Configuration" tab is active, showing a "Reset Configuration" button and a list of configuration options. The "Tasks and Queues" section is expanded, showing a table of tasks.

Task N...	Priority	Stack S...	Entry F...	Code G...	Param...	Allocation	Buffer	Control...
default...	osPrior...	128	StartDe...	Default	NULL	Dynamic	NULL	NULL
myTas...	osPrior...	128	StartTa...	Default	NULL	Dynamic	NULL	NULL

Below the table, there are "Add" and "Delete" buttons. The "Queues" section is also visible, showing a table with columns: "Queue Name", "Queue Size", "Item Size", "Allocation", "Buffer Name", and "Control Bloo...".

Step 8. Add mutex

The screenshot shows the 'Embedded-RTOS-Assignment-7.ioc - Pinout & Configuration' window. The left sidebar lists various components, with 'FREERTOS' selected under 'Middleware and Software Packs'. The main area is titled 'FREERTOS Mode and Configuration' and shows the 'Interface' set to 'CMSIS_V1'. Below this, the 'Configuration' section is active, displaying a table of 'Mutexes'.

Configuration

Reset Configuration

Mutaxes

Mutex Name	Allocation	Control Block Name
myMutex01	Dynamic	NULL

Buttons: Add, Delete

Recursive Mutaxes

Mutex Name	Allocation	Control Block Name
------------	------------	--------------------

Buttons: Add, Delete

Step 9. Change time base source to TIM1

Embedded-RTOS-Assignment-7.ioc - Pinout & Configuration

Pinout & Configuration	Clock Configuration	Project Manager																				
Software Packs Pinout																						
Search: <input type="text"/> Categories: A->Z	SYS Mode and Configuration <table border="1"> <thead> <tr> <th colspan="2">Mode</th> </tr> </thead> <tbody> <tr> <td>Debug</td> <td>Serial Wire</td> </tr> <tr> <td><input type="checkbox"/></td> <td>System Wake-Up 1</td> </tr> <tr> <td><input type="checkbox"/></td> <td>System Wake-Up 2</td> </tr> <tr> <td><input type="checkbox"/></td> <td>System Wake-Up 3</td> </tr> <tr> <td><input type="checkbox"/></td> <td>System Wake-Up 4</td> </tr> <tr> <td><input type="checkbox"/></td> <td>System Wake-Up 5</td> </tr> <tr> <td>Power Voltage Detector In</td> <td>Disable</td> </tr> <tr> <td>VREFBUF Mode</td> <td>Disable</td> </tr> <tr> <td>Timebase Source</td> <td>TIM1</td> </tr> </tbody> </table>		Mode		Debug	Serial Wire	<input type="checkbox"/>	System Wake-Up 1	<input type="checkbox"/>	System Wake-Up 2	<input type="checkbox"/>	System Wake-Up 3	<input type="checkbox"/>	System Wake-Up 4	<input type="checkbox"/>	System Wake-Up 5	Power Voltage Detector In	Disable	VREFBUF Mode	Disable	Timebase Source	TIM1
Mode																						
Debug	Serial Wire																					
<input type="checkbox"/>	System Wake-Up 1																					
<input type="checkbox"/>	System Wake-Up 2																					
<input type="checkbox"/>	System Wake-Up 3																					
<input type="checkbox"/>	System Wake-Up 4																					
<input type="checkbox"/>	System Wake-Up 5																					
Power Voltage Detector In	Disable																					
VREFBUF Mode	Disable																					
Timebase Source	TIM1																					
System Core DMA GPIO IWDG NVIC RCC SYS TSC WWDG	Configuration <p>Warning: This peripheral has no parameters to be configured.</p>																					
Analog																						
Timers																						
Connectivity																						

Step 10. Code the interrupt section

```
198 void EXTI15_10_IRQHandler(void)
199 {
200     /* USER CODE BEGIN EXTI15_10_IRQn 0 */
201     int ret = taskENTER_CRITICAL_FROM_ISR();
202     count++;
203     taskEXIT_CRITICAL_FROM_ISR(ret);
204
205     /* USER CODE END EXTI15_10_IRQn 0 */
206     HAL_GPIO_EXTI_IRQHandler(LPS22HB_INT_DRDY_EXTI0_Pin);
207     HAL_GPIO_EXTI_IRQHandler(LSM6DSL_INT1_EXTI11_Pin);
208     HAL_GPIO_EXTI_IRQHandler(BUTTON_EXTI13_Pin);
209     HAL_GPIO_EXTI_IRQHandler(ARD_D2_Pin);
210     HAL_GPIO_EXTI_IRQHandler(HTS221_DRDY_EXTI15_Pin);
211     /* USER CODE BEGIN EXTI15_10_IRQn 1 */
212
213     /* USER CODE END EXTI15_10_IRQn 1 */
214 }
215
```

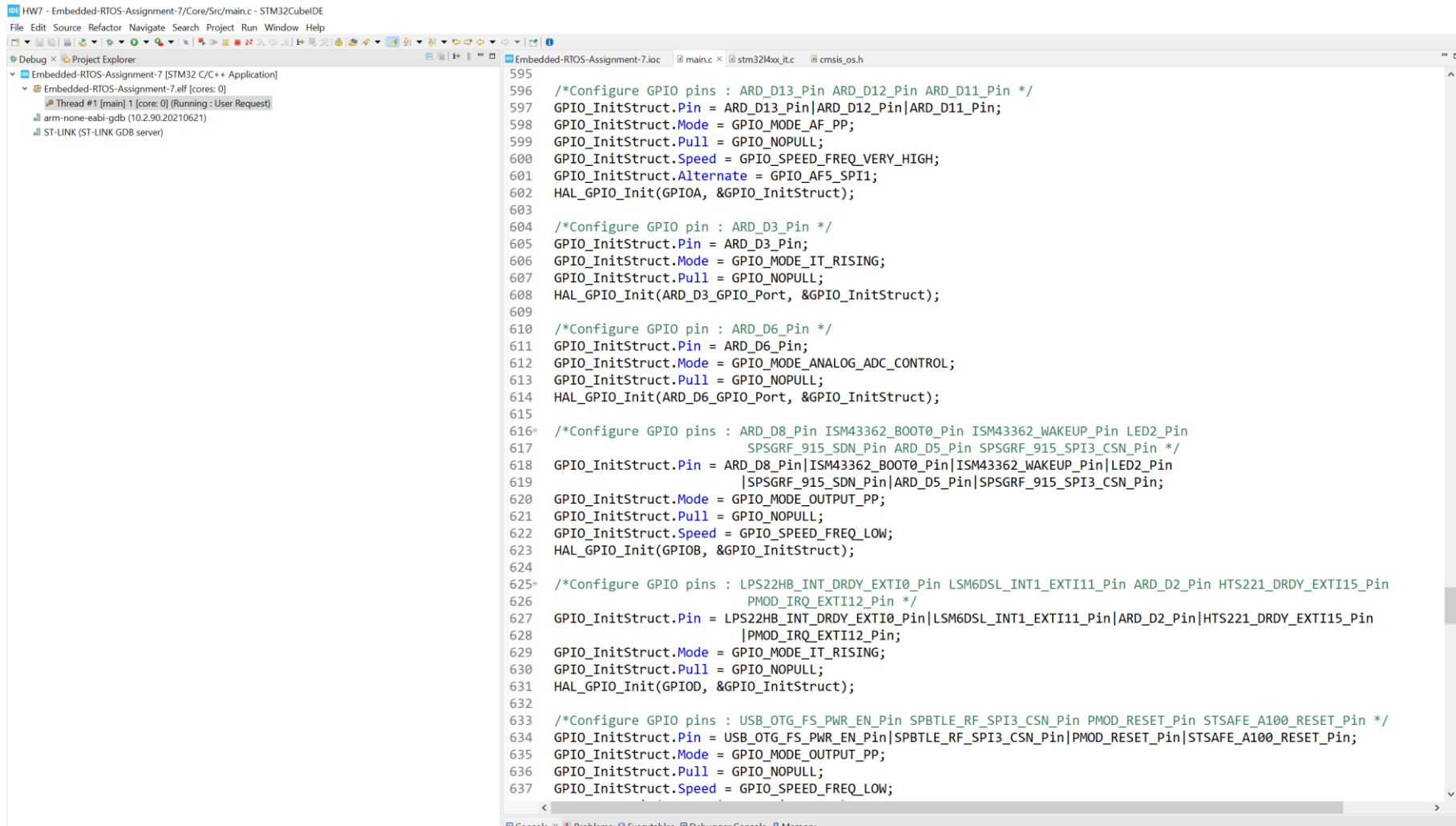
Step 11. Code the first task

```
Embedded-RTOS-Assignment-7.ioc  main.c x stm32l4xx_it.c cmsis_os.h
696  * @retval None
697  */
698  /* USER CODE END Header_StartDefaultTask */
699  void StartDefaultTask(void const * argument)
700  {
701      /* USER CODE BEGIN 5 */
702      /* Infinite loop */
703      for(;;)
704      {
705          osDelay(5000); // Sleep 5 seconds
706          // Read count value
707          int flashCount = 0;
708          taskENTER_CRITICAL();
709          flashCount = count;
710          if(count == 0){
711              taskEXIT_CRITICAL();
712              continue;
713          }
714          // Else clear the count
715          count = 0;
716          taskEXIT_CRITICAL();
717          // Grab the mutex
718          osMutexWait(myMutex01Handle, osWaitForever);
719          // Always end with LED ON
720          HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 1);
721          while(flashCount != 0){
722              flashCount--;
723              osDelay(500);
724              HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
725          }
726          // Always end with LED OFF
727          HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 0);
728          // Release the mutex
729          osMutexRelease(myMutex01Handle);
730      }
731      /* USER CODE END 5 */
732  }
```

Step 12. Code the second task

```
Embedded-RTOS-Assignment-7.ioc  main.c × stm32l4xx_it.c cmsis_os.h
739 */
740 /* USER CODE END Header_StartTask02 */
741 void StartTask02(void const * argument)
742 {
743     /* USER CODE BEGIN StartTask02 */
744     /* Infinite loop */
745     for(;;)
746     {
747         osDelay(5000); // Sleep 5 seconds
748         // Read count value
749         int flashCount = 0;
750         taskENTER_CRITICAL();
751         flashCount = count;
752         if(count == 0){
753             taskEXIT_CRITICAL();
754             continue;
755         }
756         // Else clear the count
757         count = 0;
758         taskEXIT_CRITICAL();
759
760         // Grab the mutex
761         osMutexWait(myMutex01Handle, osWaitForever);
762         // Always end with LED ON
763         HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 1);
764         while(flashCount != 0){
765             flashCount--;
766             osDelay(1000);
767             HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
768         }
769         // Always end with LED OFF
770         HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, 0);
771         // Release the mutex
772         osMutexRelease(myMutex01Handle);
773     }
774     /* USER CODE END StartTask02 */
775 }
```

Step 13. Build and run the code, test is successful



The screenshot displays an IDE window for an STM32CubeIDE project named "Embedded-RTOS-Assignment-7". The interface includes a menu bar (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help), a toolbar, and a Project Explorer on the left. The Project Explorer shows the project structure, including a "Debug" tab and a "Project Explorer" tab. The "Debug" tab is active, showing a "Thread #1 [main] 1 [core: 0] (Running: User Request)" and "arm-none-eabi-gdb (10.2.90.20210621)" as the debugger. The "Project Explorer" tab shows the project files, including "main.c" and "stm324xx_it.c". The main editor displays the source code for "main.c", which is a C program for configuring GPIO pins. The code includes comments and function calls for initializing GPIO pins for various peripherals like SPI1, SPI3, and I2C. The code is as follows:

```
595
596 /*Configure GPIO pins : ARD_D13_Pin ARD_D12_Pin ARD_D11_Pin */
597 GPIO_InitStruct.Pin = ARD_D13_Pin|ARD_D12_Pin|ARD_D11_Pin;
598 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
599 GPIO_InitStruct.Pull = GPIO_NOPULL;
600 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
601 GPIO_InitStruct.Alternate = GPIO_AF5_SPI1;
602 HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
603
604 /*Configure GPIO pin : ARD_D3_Pin */
605 GPIO_InitStruct.Pin = ARD_D3_Pin;
606 GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
607 GPIO_InitStruct.Pull = GPIO_NOPULL;
608 HAL_GPIO_Init(ARD_D3_GPIO_Port, &GPIO_InitStruct);
609
610 /*Configure GPIO pin : ARD_D6_Pin */
611 GPIO_InitStruct.Pin = ARD_D6_Pin;
612 GPIO_InitStruct.Mode = GPIO_MODE_ANALOG_ADC_CONTROL;
613 GPIO_InitStruct.Pull = GPIO_NOPULL;
614 HAL_GPIO_Init(ARD_D6_GPIO_Port, &GPIO_InitStruct);
615
616 /*Configure GPIO pins : ARD_D8_Pin ISM43362_BOOT0_Pin ISM43362_WAKEUP_Pin LED2_Pin
617 SPSGRF_915_SDN_Pin ARD_D5_Pin SPSGRF_915_SPI3_CSN_Pin */
618 GPIO_InitStruct.Pin = ARD_D8_Pin|ISM43362_BOOT0_Pin|ISM43362_WAKEUP_Pin|LED2_Pin
619 |SPSGRF_915_SDN_Pin|ARD_D5_Pin|SPSGRF_915_SPI3_CSN_Pin;
620 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
621 GPIO_InitStruct.Pull = GPIO_NOPULL;
622 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
623 HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
624
625 /*Configure GPIO pins : LPS22HB_INT_DRDY_EXTI0_Pin LSM6DSL_INT1_EXTI11_Pin ARD_D2_Pin HTS221_DRDY_EXTI15_Pin
626 PMOD_IRQ_EXTI12_Pin */
627 GPIO_InitStruct.Pin = LPS22HB_INT_DRDY_EXTI0_Pin|LSM6DSL_INT1_EXTI11_Pin|ARD_D2_Pin|HTS221_DRDY_EXTI15_Pin
628 |PMOD_IRQ_EXTI12_Pin;
629 GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
630 GPIO_InitStruct.Pull = GPIO_NOPULL;
631 HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
632
633 /*Configure GPIO pins : USB_OTG_FS_PWR_EN_Pin SPBTLE_RF_SPI3_CSN_Pin PMOD_RESET_Pin STSAFE_A100_RESET_Pin */
634 GPIO_InitStruct.Pin = USB_OTG_FS_PWR_EN_Pin|SPBTLE_RF_SPI3_CSN_Pin|PMOD_RESET_Pin|STSAFE_A100_RESET_Pin;
635 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
636 GPIO_InitStruct.Pull = GPIO_NOPULL;
637 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
```