

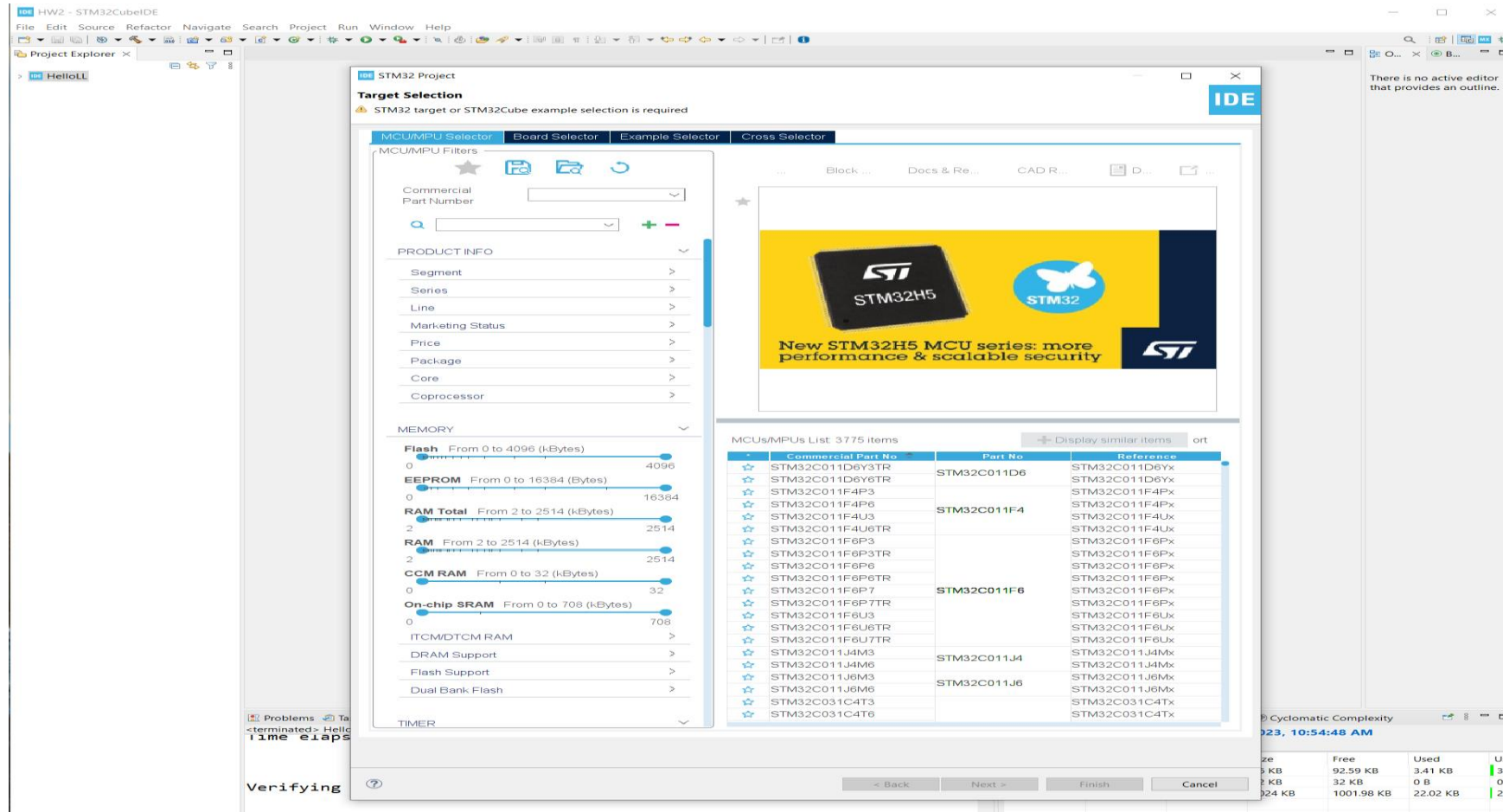
UCSD Embedded C Assignment 4

By

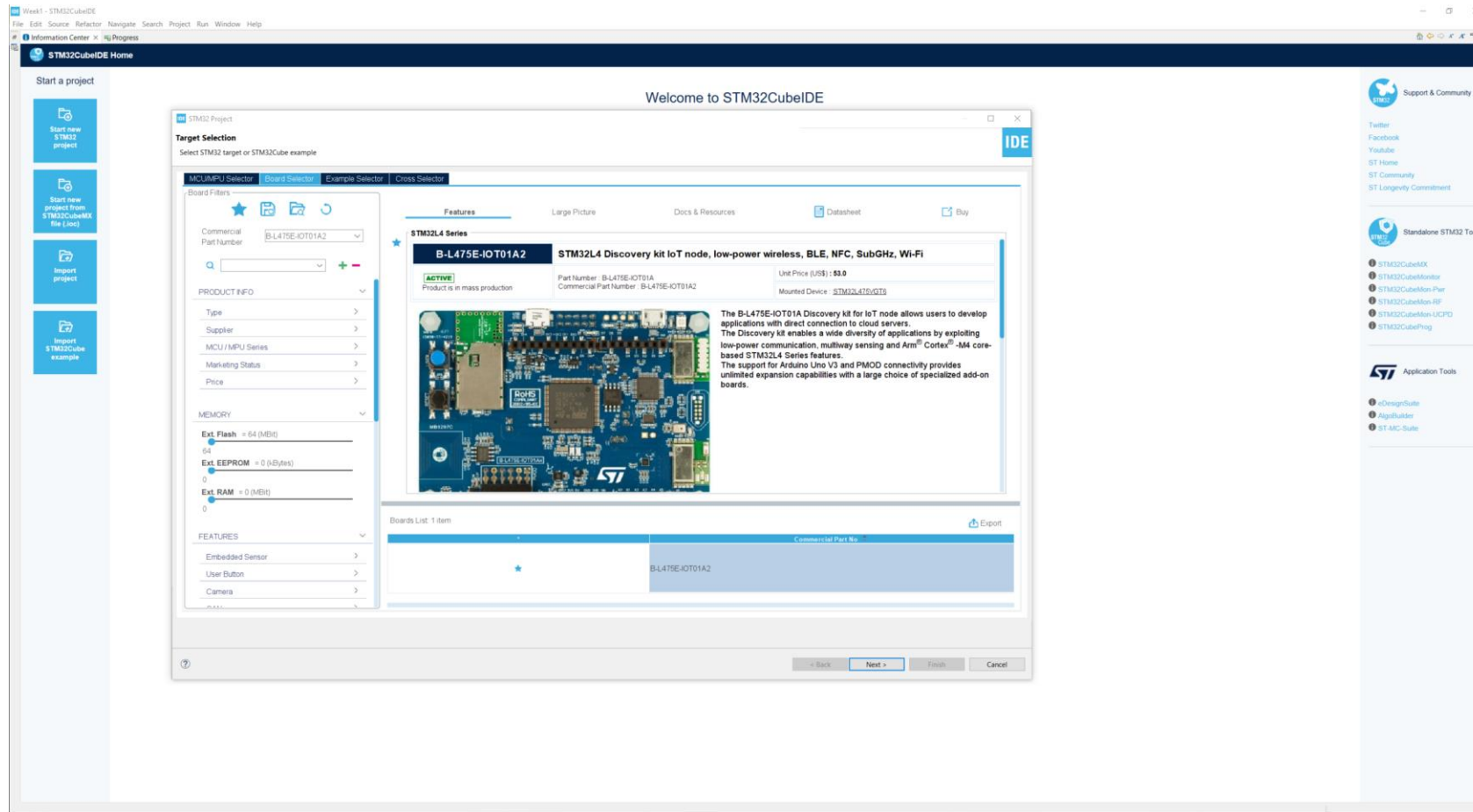
Hsuankai Chang

hsuankac@umich.edu

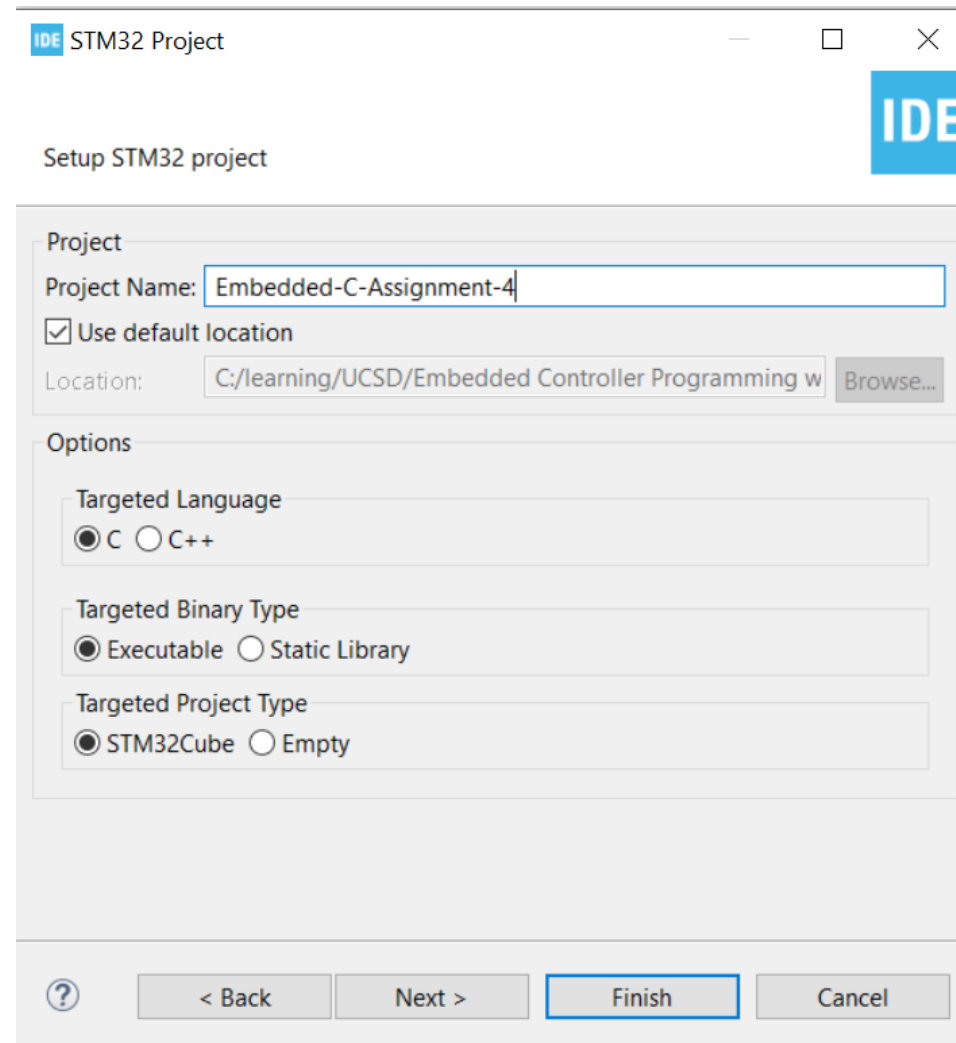
Step 1. Startup STM32CubeIDE and create new STM32 project



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 dialog is titled 'Setup STM32 project'. It has two main sections: 'Project' and 'Options'. In the 'Project' section, the 'Project Name' field contains 'Embedded-C-Assignment-4'. Below it, the 'Use default location' checkbox is checked. The 'Location' field shows 'C:/learning/UCSD/Embedded Controller Programming w' with a 'Browse...' button to its right. The 'Options' section contains 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 four buttons: a help button (question mark icon), '< Back', 'Next >', and 'Finish' (which is highlighted with a blue border). A 'Cancel' button is also present.

IDE STM32 Project

Setup STM32 project

Project

Project Name: Embedded-C-Assignment-4

☒ Use default location

Location: C:/learning/UCSD/Embedded Controller Programming w 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 "B-L475E-IOT01A2" and "Firmware Package Name and Version" is "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 a link to the "Firmware Updater". The third section has three radio button options for code generation, with "Copy only the necessary library files" selected. At the bottom 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

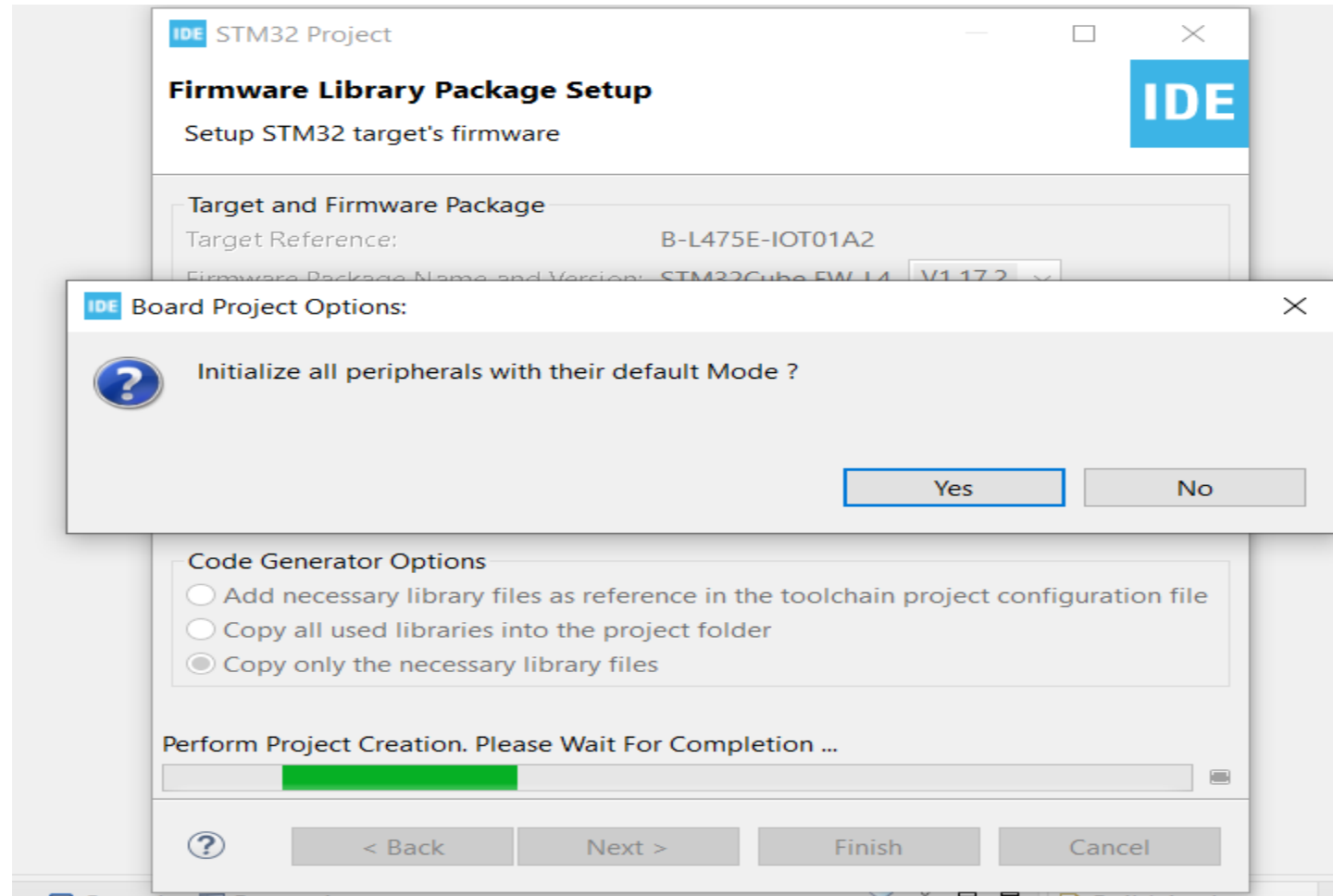
☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

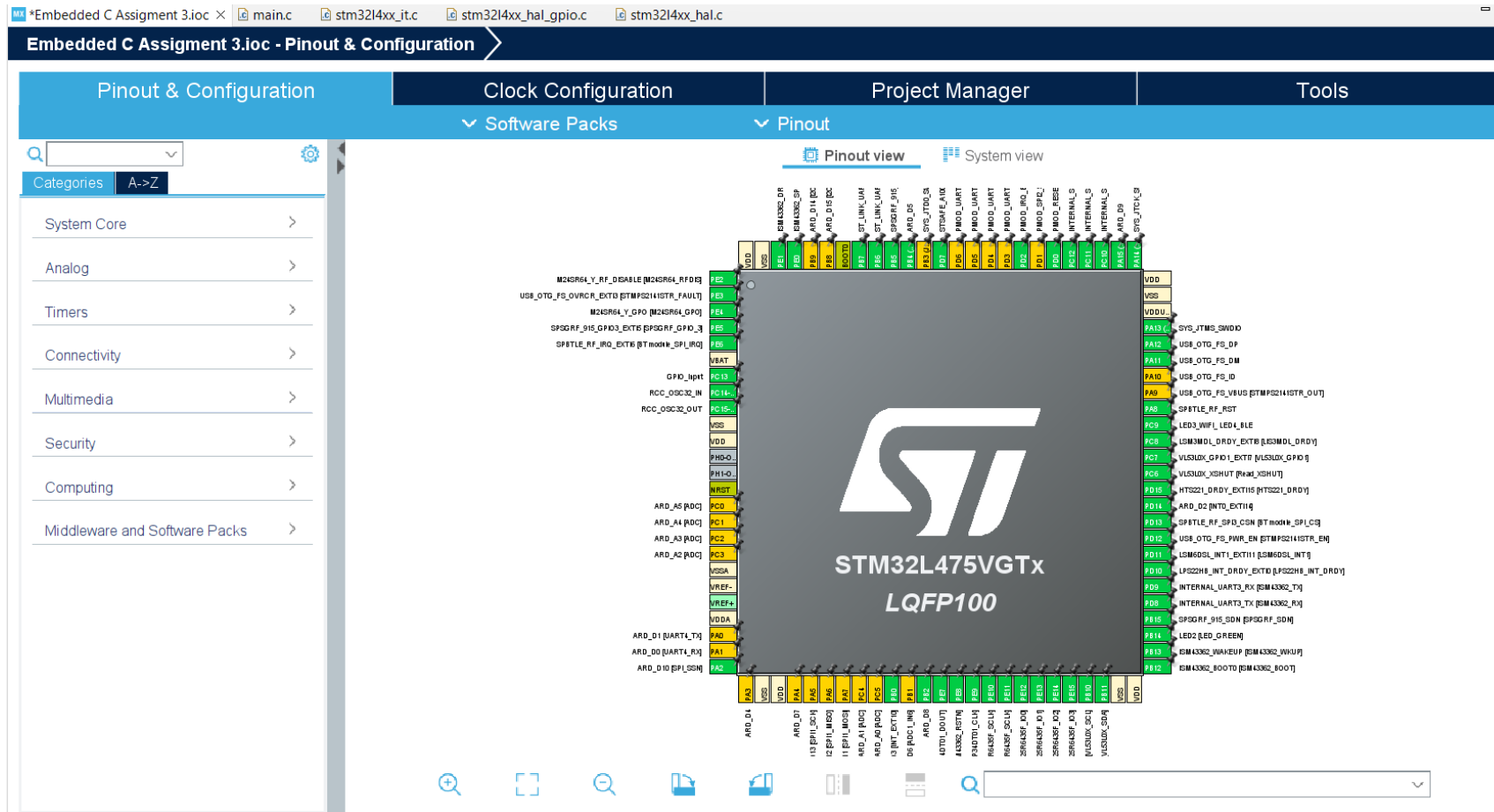
☒ Copy only the necessary library files

? < Back Next > Finish Cancel

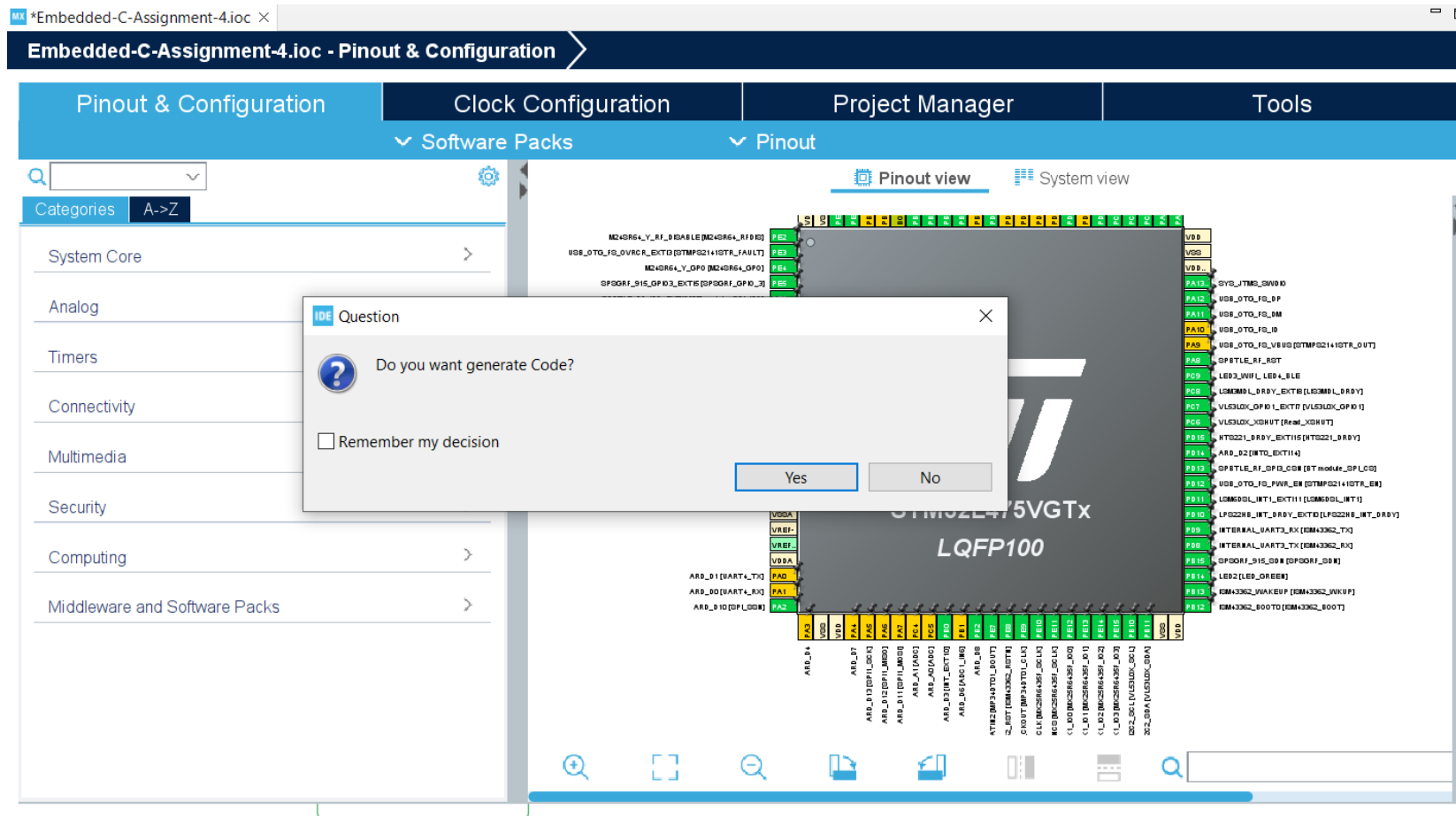
Step 5. Click yes to initialize all peripherals to default



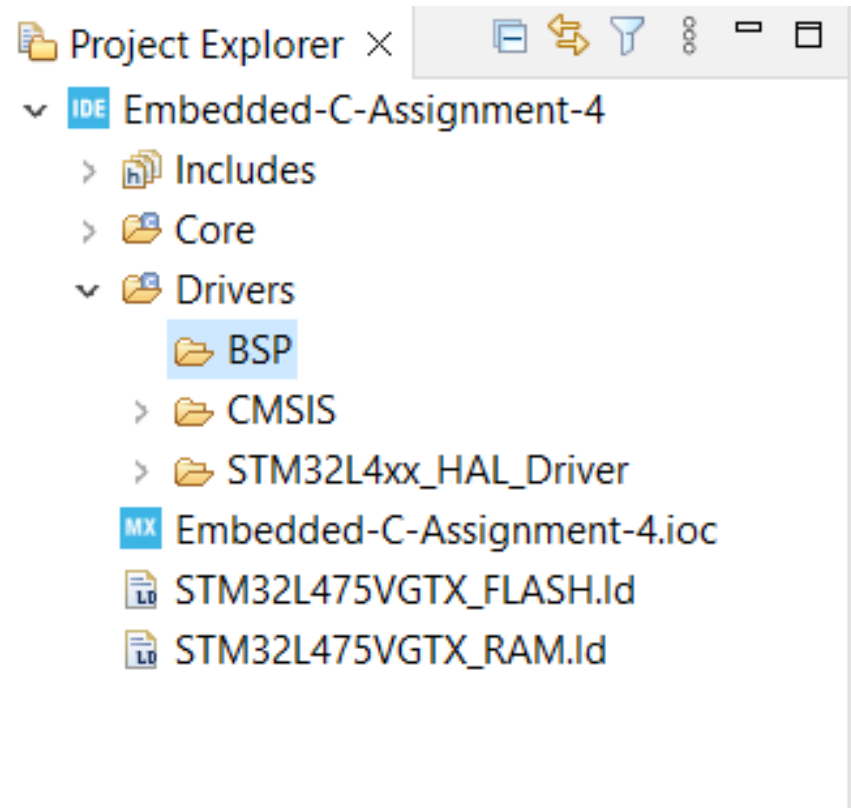
Step 6. When in .ioc file, click Pinout & Configurations, make sure BUTTON and LED2 are configured correctly



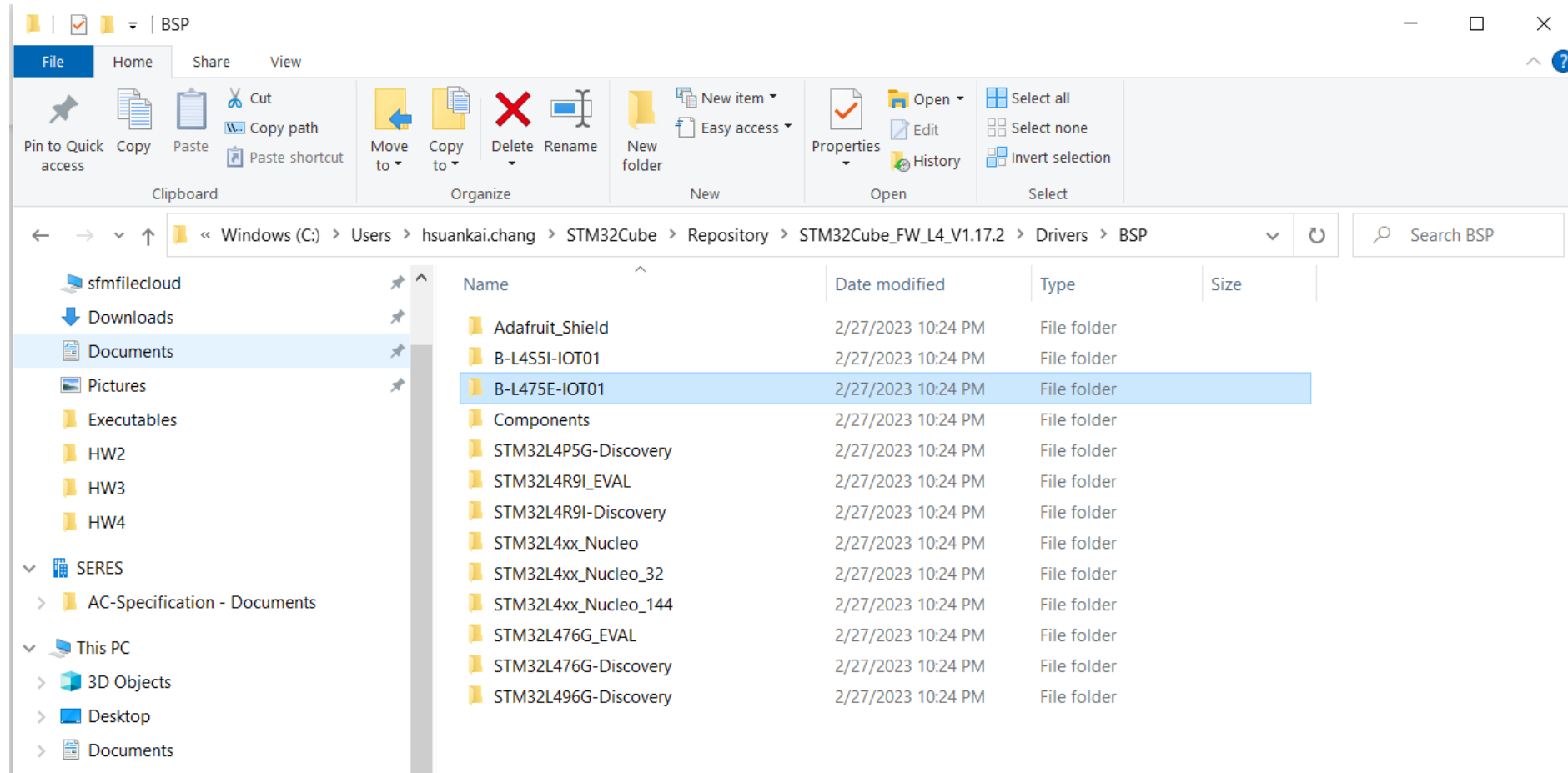
Step 7. Generate Code



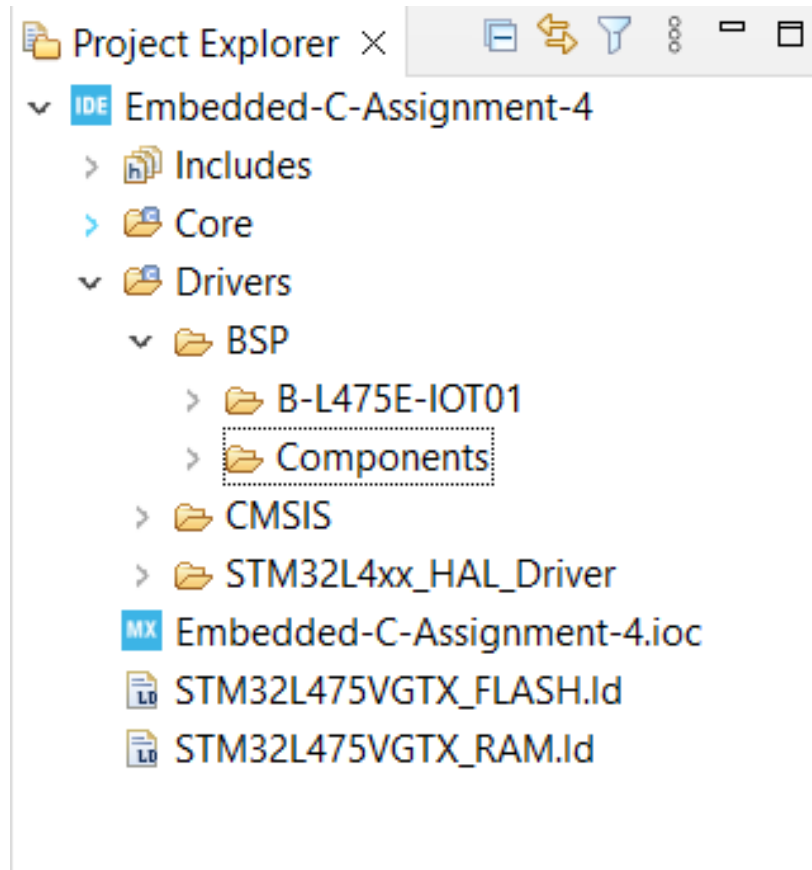
Step 8. Create new BSP folder under Drivers



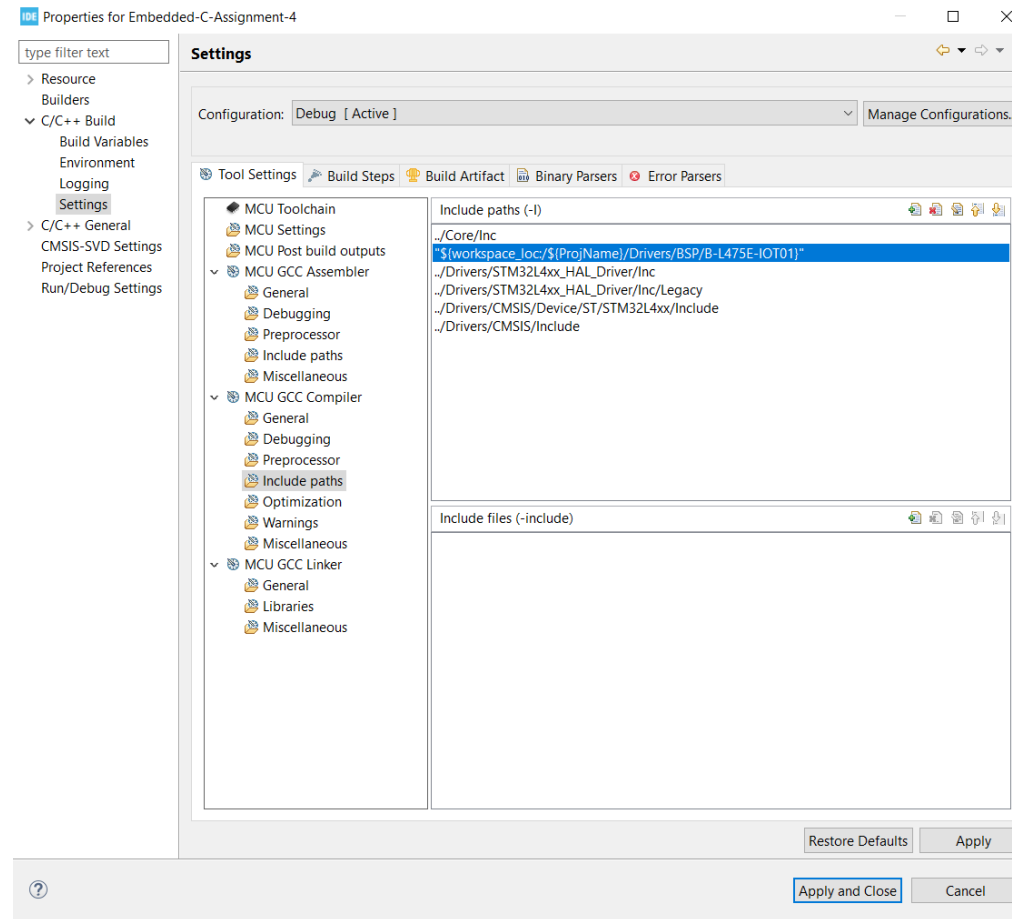
Step 9. Find the BSP code folder, B-L475E-IOT01 and Components folder that is included in Repository/../../Drivers/BSP



Step 10. Copy and paste it under the BSP folder



Step 11. Add BSP/B-L475E-IOT01 to include path



Step 12. Add BSP related code header files

```
17  */
18  /* USER CODE END Header */
19  /* Includes -----*/
20  #include "main.h"
21
22  /* Private includes -----*/
23  /* USER CODE BEGIN Includes */
24  #include "stm32l475e_iot01.h"
25  #include "stm32l475e_iot01_tsensor.h"
26  #include <stdio.h>
27  /* USER CODE END Includes */
28
29  /* Private typedef -----*/
30  /* USER CODE BEGIN PTD */
31
```

Step 13. Added the BSP code in main.c as below

```
main.c ×
118  /* USER CODE END 2 */
119
120  /* Infinite loop */
121  /* USER CODE BEGIN WHILE */
122
123  BSP_TSENSOR_Init();
124
125  while (1)
126  {
127      /* USER CODE END WHILE */
128
129      /* USER CODE BEGIN 3 */
130      float temp = BSP_TSENSOR_ReadTemp();
131      printf("temp : %f", temp);
132
133      uint32_t button = BSP_PB_GetState(BUTTON_USER);
134      if(button)
135      {
136          BSP_LED_Off(LED_GREEN);
137      }
138      else
139      {
140          BSP_LED_On(LED_GREEN);
141      }
142      HAL_Delay(1000);
143  }
144  /* USER CODE END 3 */
145 }
146
147 /**
148  * @brief System Clock Configuration
```

Step 14. See the result when finish building the project

The screenshot shows the CDT Build Console window. The top toolbar includes icons for Problems, Tasks, Console, and Properties. The console output displays the following commands and results:

```
arm-none-eabi-gcc -mcpu=cortex-m4 -std=gnu11 -g3 -DDEBUG -DUSE_HAL_DRIVER -I../Core/Src/stm32l4xx_hal_msp.c -I../Core/Src/stm32l4xx_it.c -mcpu=cortex-m4 -std=gnu11 -g3 -DDEBUG -DUSE_HAL_DRIVER -I../Core/Src/syscalls.c -mcpu=cortex-m4 -std=gnu11 -g3 -DDEBUG -DUSE_HAL_DRIVER -I../Core/Src/system_stm32l4xx.c -mcpu=cortex-m4 -std=gnu11 -g3 -DDEBUG -DUSE_HAL_DRIVER -o "Embedded-C-Assignment-4.elf" @objects.list -T"C:\Program Files\ARM\GNU Tools for ARM-based Processors\bin\ld.exe"
Finished building target: Embedded-C-Assignment-4.elf
```

Below this, the linker command and its output are shown:

```
arm-none-eabi-size Embedded-C-Assignment-4.elf
arm-none-eabi-objdump -h -S Embedded-C-Assignment-4.elf > "Embedded-C-Assignment-4.list"
```

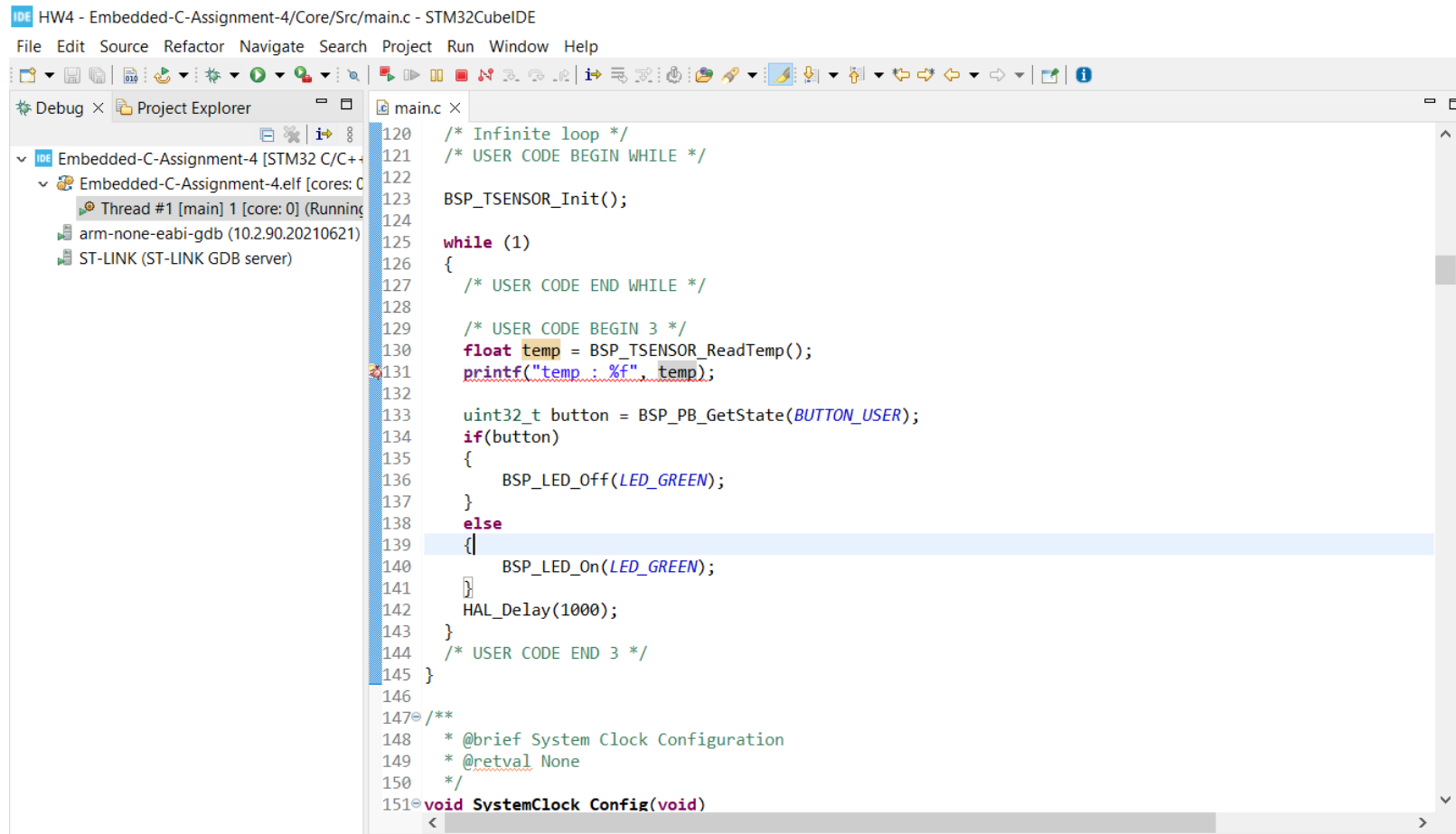
	text	data	bss	dec	hex	filename
	29816	144	3584	33544	8308	Embedded-C-Assignment-4.elf

Finished building: default.size.stdout

Finished building: Embedded-C-Assignment-4.list

21:49:43 Build Finished. 0 errors, 0 warnings. (took 7s.86ms)

Step 15. Run in debug mode, confirm that LED toggles on/off when you press the button



The screenshot shows the STM32CubeIDE interface with the following components:

- Project Explorer:** Shows the project structure with 'Embedded-C-Assignment-4' selected. The 'Thread #1 [main] 1 [core: 0] (Running)' is visible.
- Code Editor:** Displays the file 'main.c' with the following code:

```
120  /* Infinite loop */
121  /* USER CODE BEGIN WHILE */
122
123  BSP_TSSENSOR_Init();
124
125  while (1)
126  {
127      /* USER CODE END WHILE */
128
129      /* USER CODE BEGIN 3 */
130      float temp = BSP_TSSENSOR_ReadTemp();
131      printf("temp : %f", temp);
132
133      uint32_t button = BSP_PB_GetState(BUTTON_USER);
134      if(button)
135      {
136          BSP_LED_Off(LED_GREEN);
137      }
138      else
139      {
140          BSP_LED_On(LED_GREEN);
141      }
142      HAL_Delay(1000);
143  }
144  /* USER CODE END 3 */
145  }
146
147  /**
148   * @brief System Clock Configuration
149   * @retval None
150   */
151  void SystemClock_Config(void)
```