

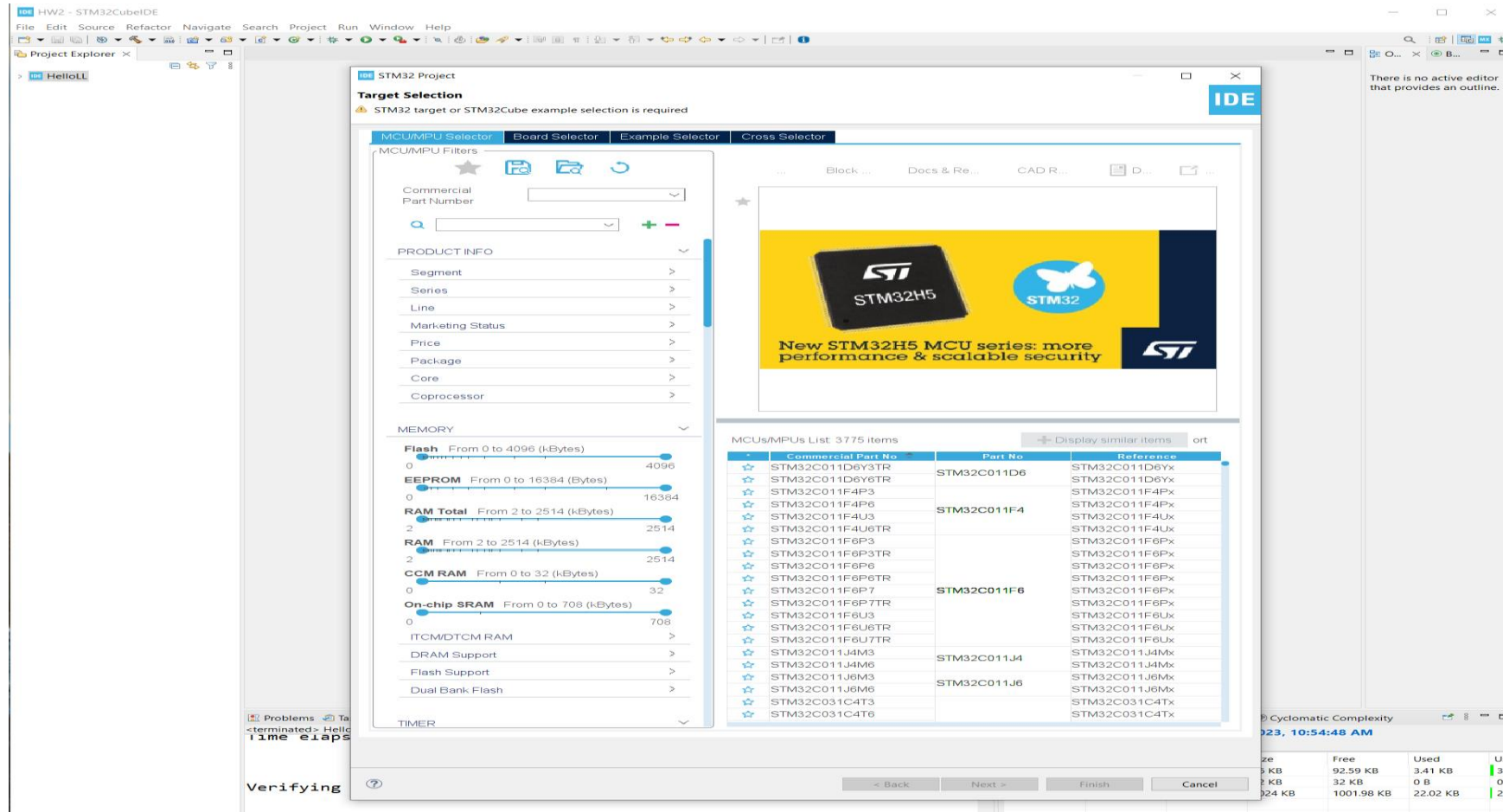
UCSD Embedded C Assignment 7

By

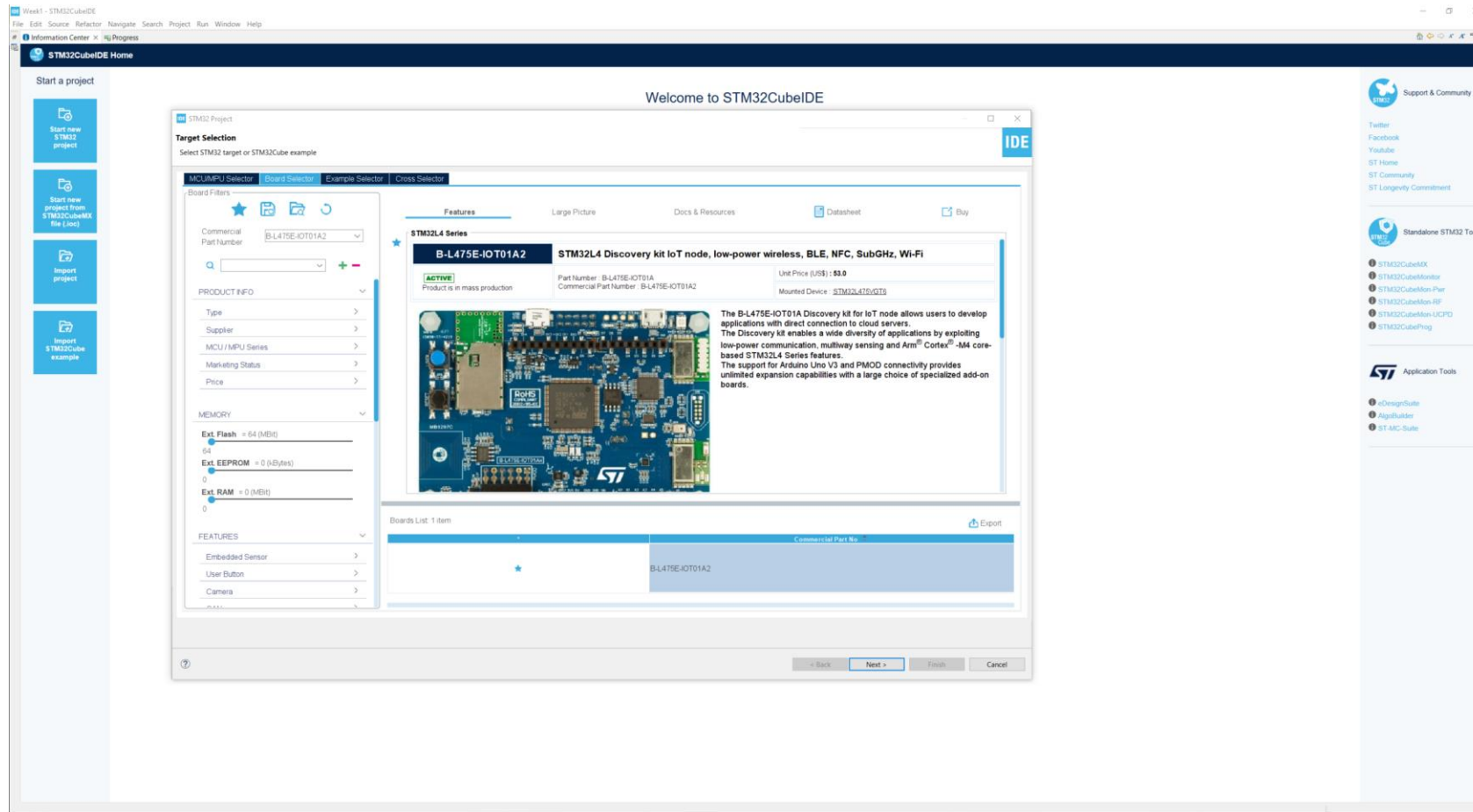
Hsuankai Chang

hsuankac@umich.edu

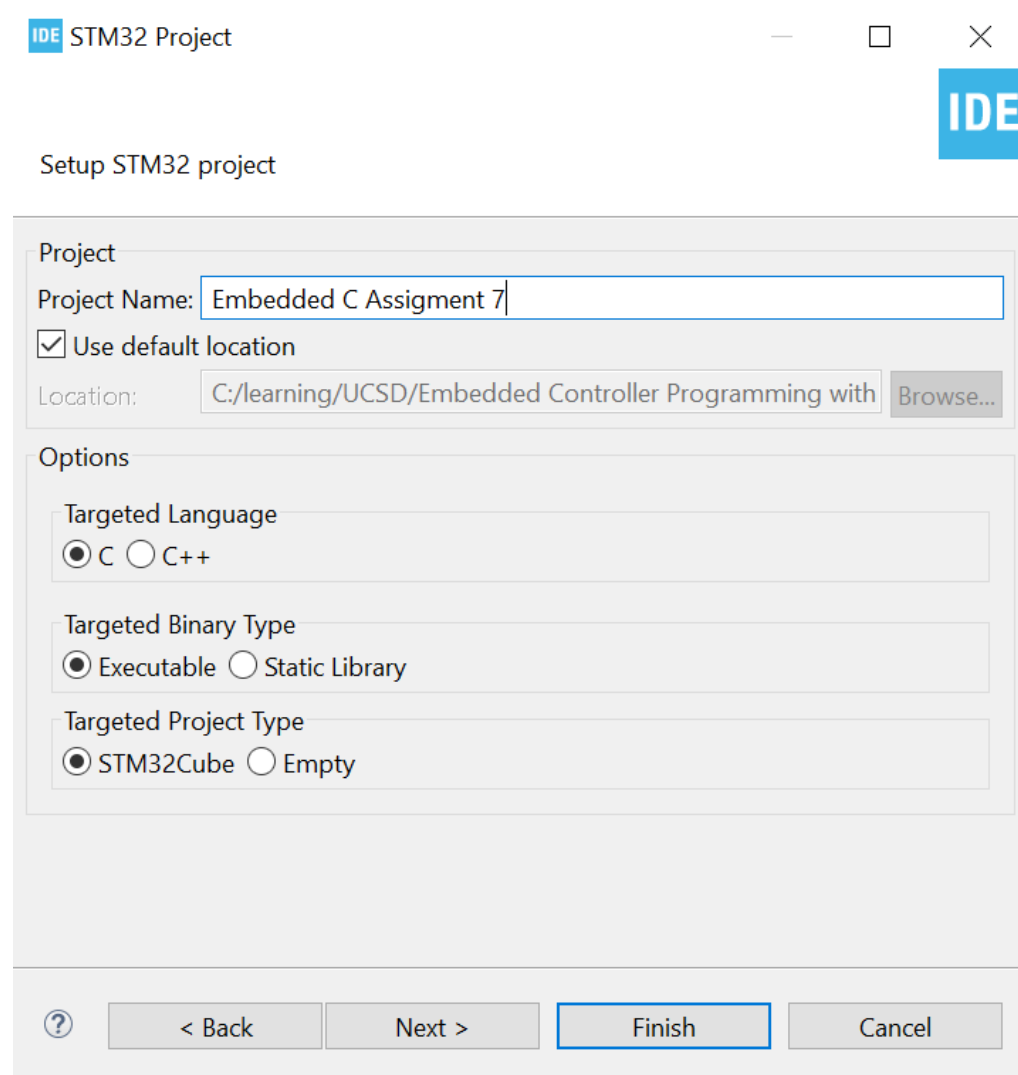
Step 1. Startup STM32CubeIDE and create new STM32 project



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 dialog is titled 'Setup STM32 project'. It contains two main sections: 'Project' and 'Options'. In the 'Project' section, the 'Project Name' field is filled with 'Embedded C Assignment 7'. The 'Use default location' checkbox is checked. The 'Location' field shows the path 'C:/learning/UCSD/Embedded Controller Programming with' followed by 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 C Assignment 7

☒ Use default location

Location: C:/learning/UCSD/Embedded Controller Programming with 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

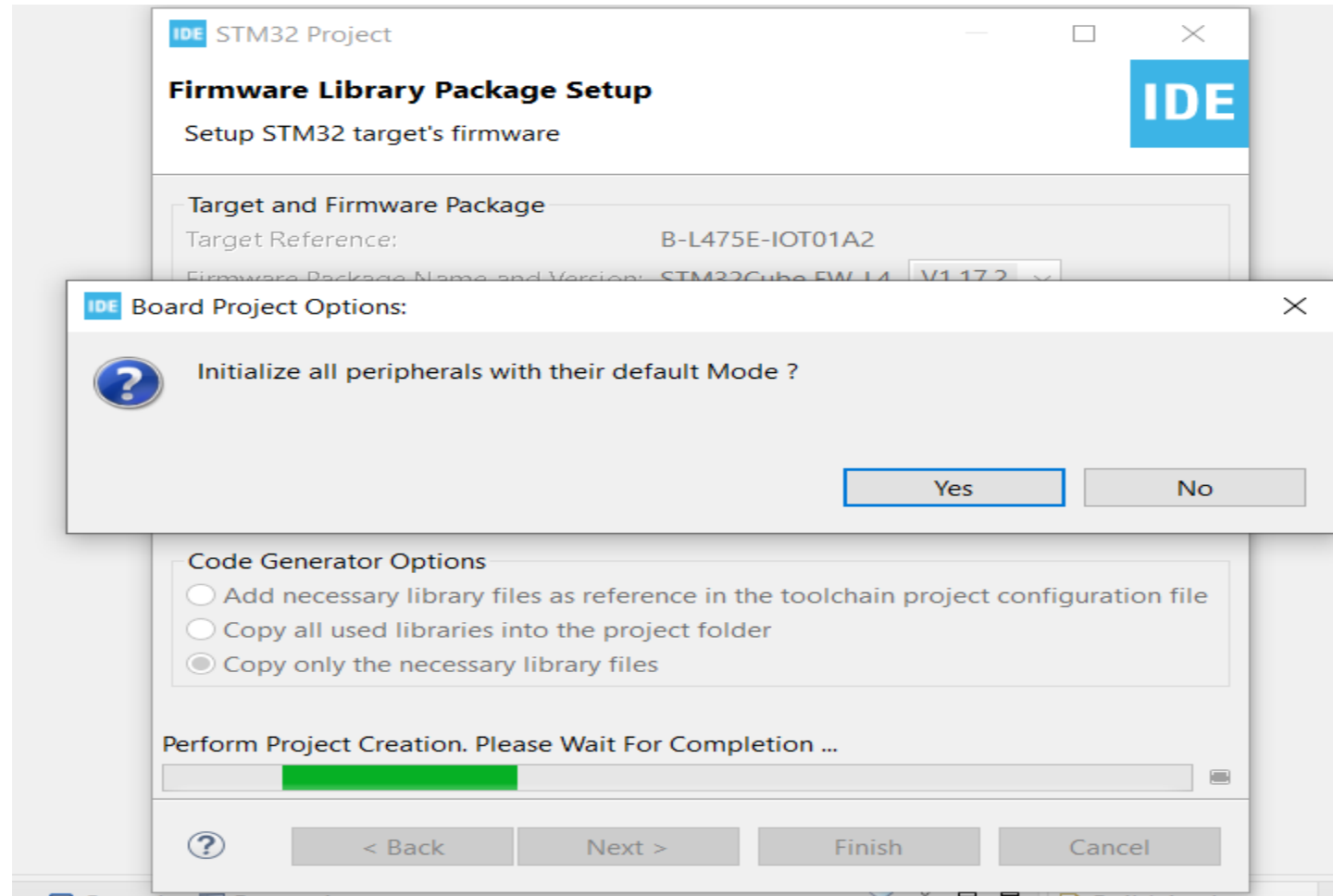
☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

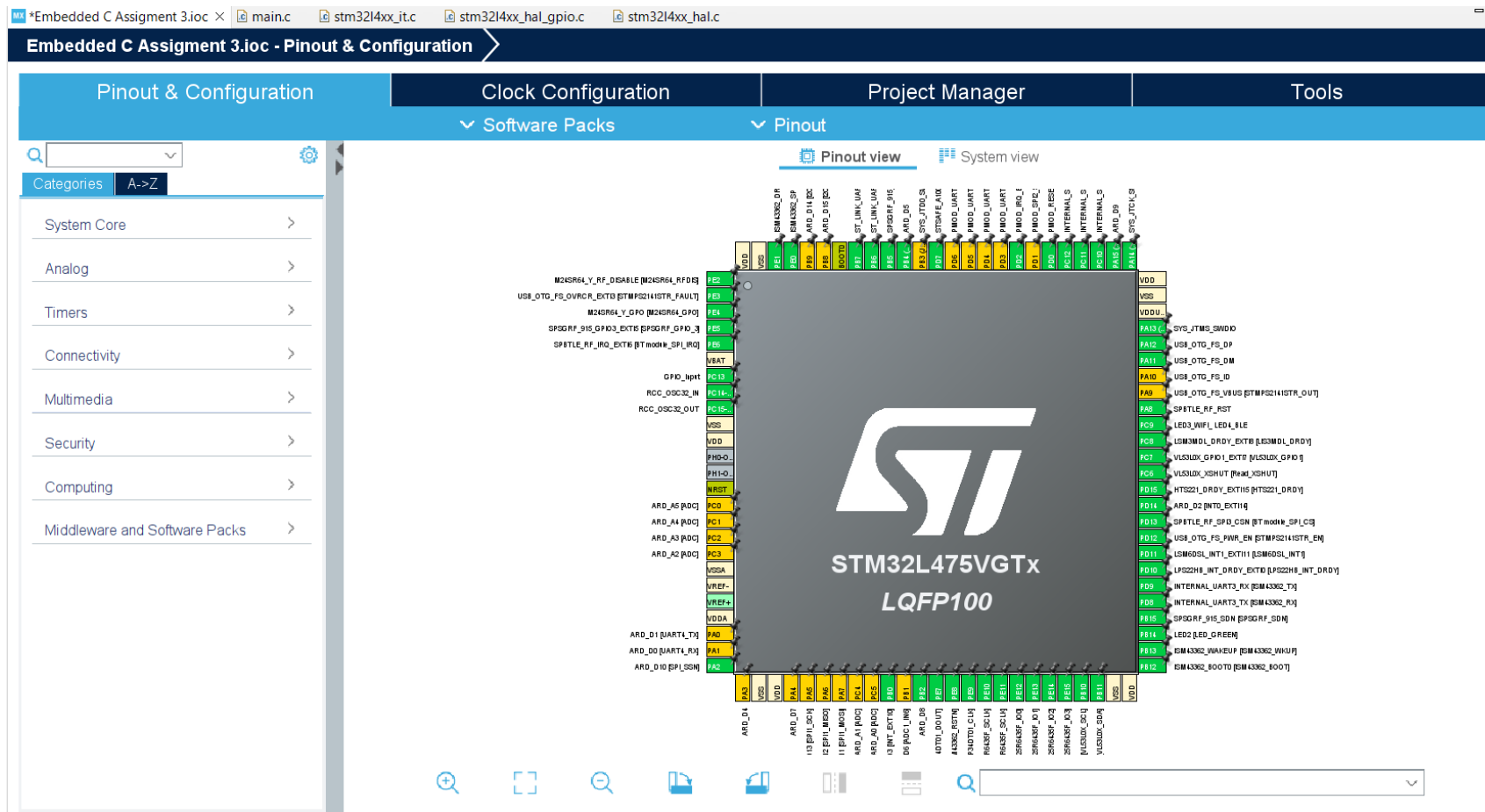
☒ Copy only the necessary library files

? < Back Next > Finish Cancel

Step 5. Click yes to initialize all peripherals to default



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



Step 7. Check the setting for USART1

The screenshot shows the STM32CubeIDE interface with the 'Pinout & Configuration' window open. The left sidebar shows the project structure, and the main area displays the configuration for USART1. The 'Mode' section shows 'Asynchronous' selected. The 'Configuration' section shows 'Parameter Settings' selected. The 'Search Signals' table is visible at the bottom.

USART1 Mode and Configuration

Mode

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

Hardware Flow Control (RS485): ☐

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-u...	Maximum ou...	Fast Mode	User Label	
PB6	USART1_TX	n/a	Alternate Fu...	No pull-up a...	Very High	Disable	ST_LINK_UART1_TX	<input checked="" type="checkbox"/>
PB7	USART1_RX	n/a	Alternate Fu...	No pull-up a...	Very High	Disable	ST_LINK_UART1_RX	<input checked="" type="checkbox"/>

Step 8. Enable UART4 connection

The screenshot displays the STM32CubeMX Pinout & Configuration window for an Embedded C Assignment 7.ioc project. The interface is divided into three main sections: Pinout & Configuration, Clock Configuration, and Project Manager.

Pinout & Configuration:

- Categories:** A-Z, System Core, Analog, Timers, Connectivity.
- Connectivity:** A list of peripheral options including CAN1, FMC, I2C1, I2C2, I2C3, LPUART1, QUADSPI, SDMMC1, SPI1, SPI2, SPI3, SWPMI1, **UART4** (highlighted), UART5, USART1, USART2, USART3, and USB_OTG_FS.
- Multimedia, Security, Computing, and Middleware and Software Packs:** These sections are currently collapsed.

UART4 Mode and Configuration:

- Mode:** Asynchronous.
- Hardware Flow Control (RS232):** Disabled.
- Hardware Flow Control (RS485):** Enabled (indicated by a red square).

Configuration:

- Reset Configuration:** Parameter Settings, User Constants, NVIC Settings, DMA Settings, GPIO Settings.
- Configure the below parameters:**
- Basic Parameters:** Baud Rate (115200 Bits/s), Word Length (8 Bits (including Parity)), Parity (None), Stop Bits (1).
- Advanced Parameters:** Data Direction (Receive and Transmit), Over Sampling (16 Samples), Single Sample (Disable).
- Advanced Features:** Auto Baudrate (Disable), TX Pin Active Level Inversion (Disable), RX Pin Active Level Inversion (Disable), Data Inversion (Disable), TX and RX Pins Swapping (Disable), Overrun (Enable), DMA on RX Error (Enable), MSB First (Disable).

Pinout:

- GPIOs:** A list of pins including PC10, PC11, PC12, PC13, PC14, PC15, VSS, VDD, PH0, PH1, NRST, PC0, PC1, PC2, PC3, VSSA, VREF, VREF+, VDDA, PA0, PA1, PA2, and PA3.
- Functions:** A list of functions including DISABLE (M24SR64_RFOIS), STMPS2141STR_FAULT, M4_Y_GPIO (M24SR64_GPIO), EXTI8 (SPISGRF_GPIO_8), EXTI9 (BT module_SPI_IRQ), VBAT, BUTTON_EXTI13 [B3], RCC_OSC32_IN, RCC_OSC32_OUT, VSS, VDD, PH0, PH1, NRST, ARD_A6 (ADC), ARD_A4 (ADC), ARD_A3 (ADC), ARD_A2 (ADC), VSSA, VREF, VREF+, VDDA, ARD_D1 (UART4_TX), ARD_D0 (UART4_RX), and ARD_D10 (SPI_SSIN).

Step 9. Generate code

Embedded C Assignment 7.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

Categories: A-Z

- System Core
- Analog
- Timers
- Connectivity
 - CAN1
 - FMC
 - I2C1
 - I2C2
 - I2C3
 - IRTIM
 - LP_UART1
 - QUADSPI
 - SDMMC1
 - SPI1
 - SPI2
 - SPI3
 - SWPMI1
 - UART4
 - UART5
 - USART1
 - USART2
 - USART3
 - USB_OTG_FS
- Multimedia
- Security
- Computing
- Middleware and Software Packs

UART4 Mode and Configuration

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

Hardware Flow Control (RS485): Enable

Question: Do you want generate Code?

☐ Remember my decision

Yes No

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Configure the below parameters:

Search (Ctrl+F)

Basic Parameters

- Baud Rate: 115200 Bits/s
- Word Length: 8 Bits (including Parity)
- Parity: None
- Stop Bits: 1

Advanced Parameters

- Data Direction: Receive and Transmit
- Over Sampling: 16 Samples
- Single Sample: Disable

Advanced Features

- Auto Baudrate: Disable
- TX Pin Active Level Inversion: Disable
- RX Pin Active Level Inversion: Disable
- Data Inversion: Disable
- TX and RX Pins Swapping: Disable
- Overrun: Enable
- DMA on RX Error: Enable
- MSB First: Disable

Pinout view | System view

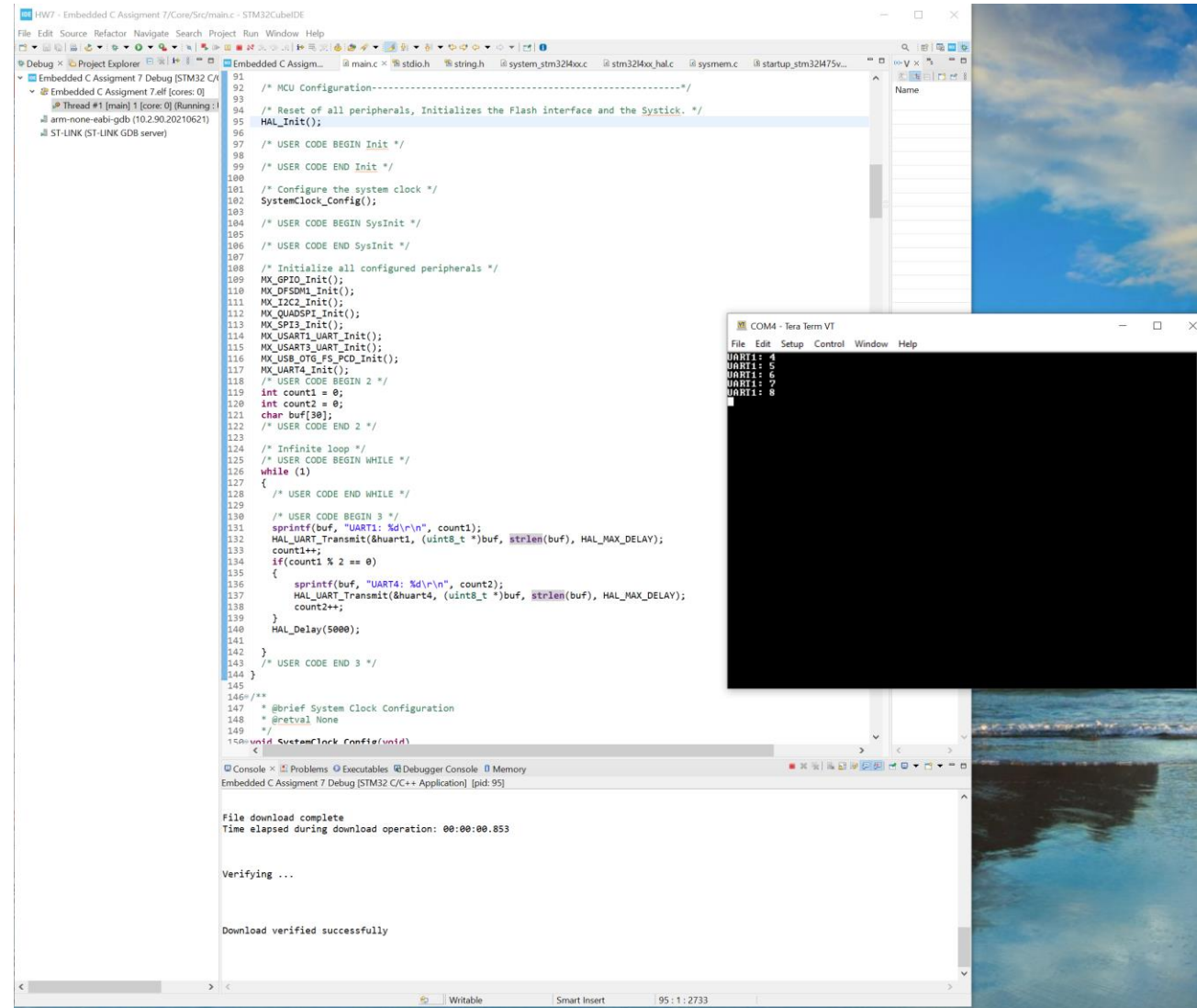
STM32L475VGTx LQFP100

ARD_D0 [UART4_RX]

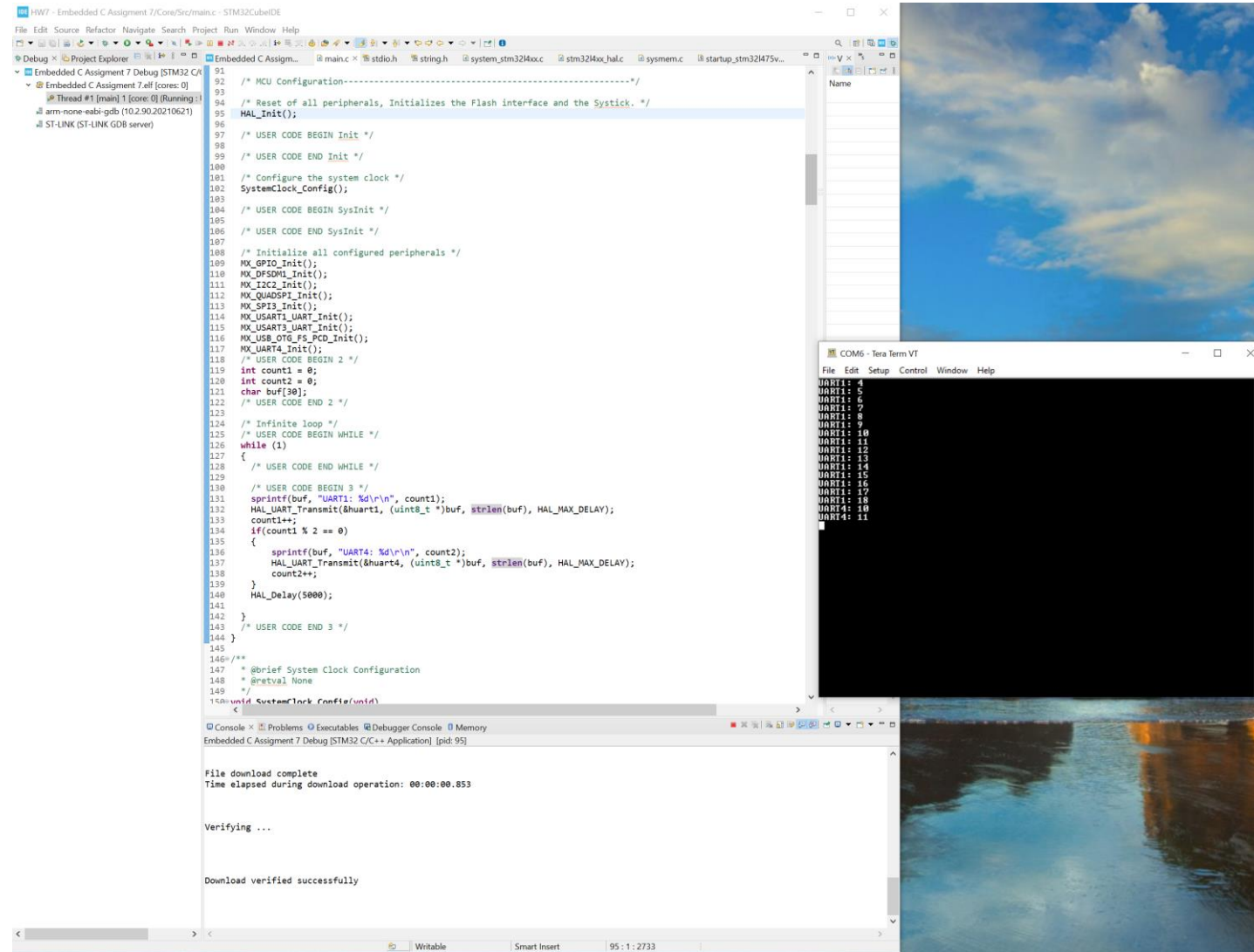
Step 10. In main.c, type the code below to send message to UART1 every 5 second, and to UART4 every 10 seconds

```
Embedded C A... | main.c × | string.h | system_stm32... | stm32l4xx_hal.c | systemem.c | startup_stm3...
91
92 /* MCU Configuration-----*/
93
94 /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
95 HAL_Init();
96
97 /* USER CODE BEGIN Init */
98
99 /* USER CODE END Init */
100
101 /* Configure the system clock */
102 SystemClock_Config();
103
104 /* USER CODE BEGIN SysInit */
105
106 /* USER CODE END SysInit */
107
108 /* Initialize all configured peripherals */
109 MX_GPIO_Init();
110 MX_DFSDM1_Init();
111 MX_I2C2_Init();
112 MX_QUADSPI_Init();
113 MX_SPI3_Init();
114 MX_USART1_UART_Init();
115 MX_USART3_UART_Init();
116 MX_USB_OTG_FS_PCD_Init();
117 MX_UART4_Init();
118 /* USER CODE BEGIN 2 */
119 int count1 = 0;
120 int count2 = 0;
121 char buf[30];
122 /* USER CODE END 2 */
123
124 /* Infinite loop */
125 /* USER CODE BEGIN WHILE */
126 while (1)
127 {
128     /* USER CODE END WHILE */
129
130     /* USER CODE BEGIN 3 */
131     sprintf(buf, "UART1: %d\r\n", count1);
132     HAL_UART_Transmit(&huart1, (uint8_t *)buf, strlen(buf), HAL_MAX_DELAY);
133     count1++;
134     if(count1 % 2 == 0)
135     {
136         sprintf(buf, "UART4: %d\r\n", count2);
137         HAL_UART_Transmit(&huart4, (uint8_t *)buf, strlen(buf), HAL_MAX_DELAY);
138         count2++;
139     }
140     HAL_Delay(5000);
141 }
142 /* USER CODE END 3 */
143 }
144
145 /**
146 * @brief System Clock Configuration
147 * @retval None
148 */
149 void SystemClock_Config(void)
```

Step 11. Build the code, run in debug mode and open Tera Term to see if we can see the UART1 message



Step 12. Switch Teram Term serial port to see if we can UART4 message, test is successful



Appendix. I use USB to UART converter and connect UART4 Rx and Tx to it

