

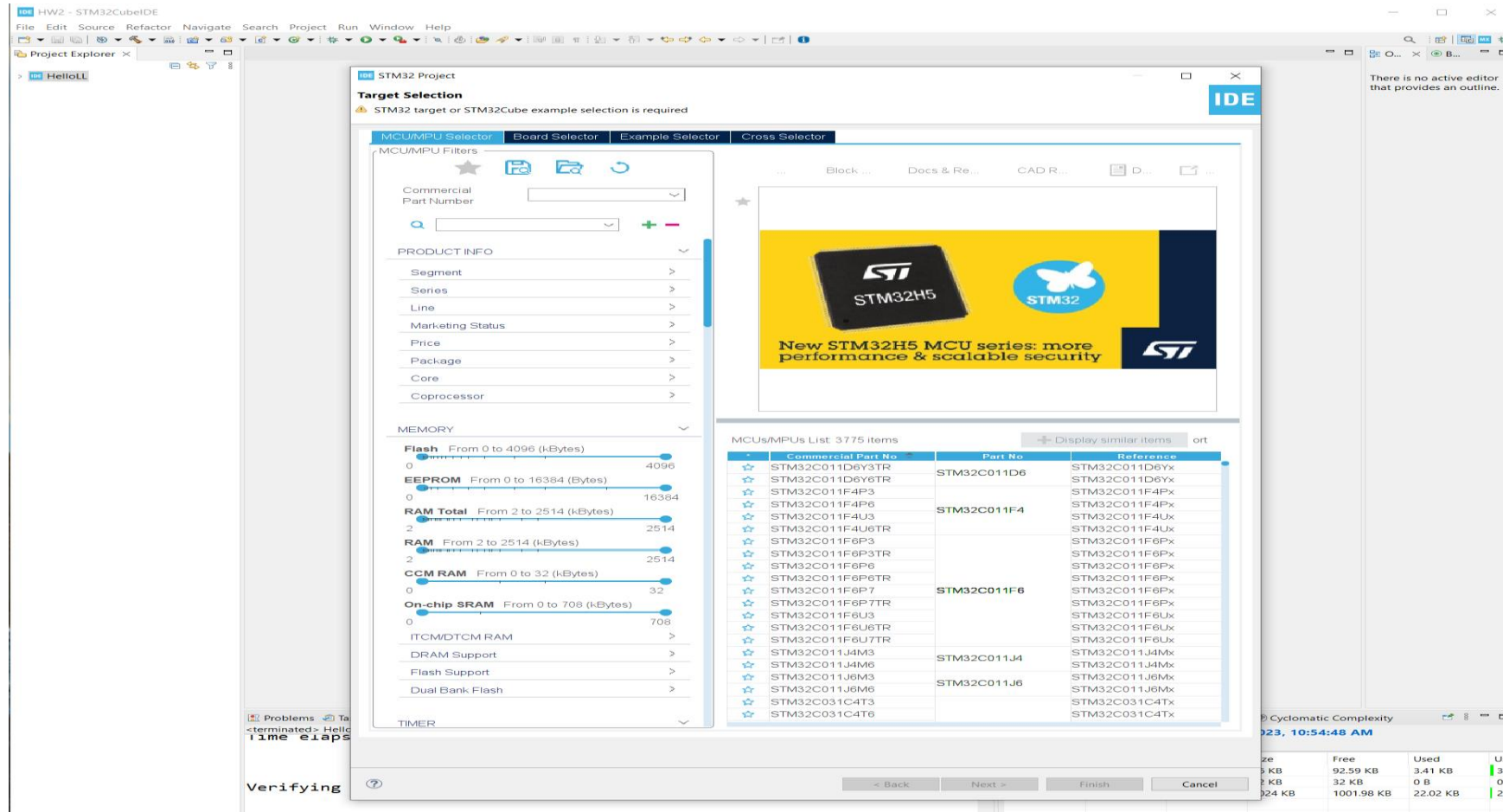
UCSD Embedded C Assignment 3

By

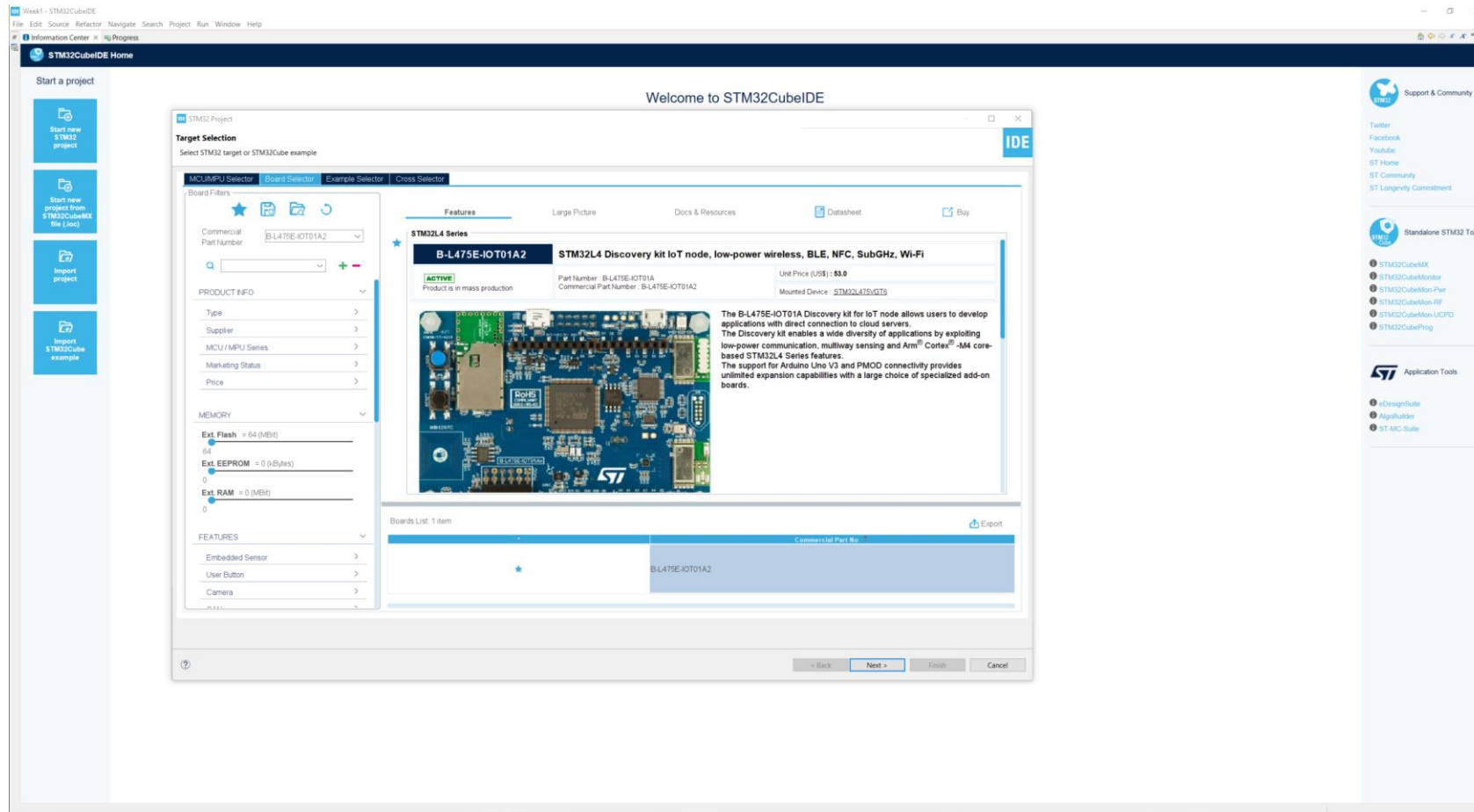
Hsuankai Chang

hsuankac@umich.edu

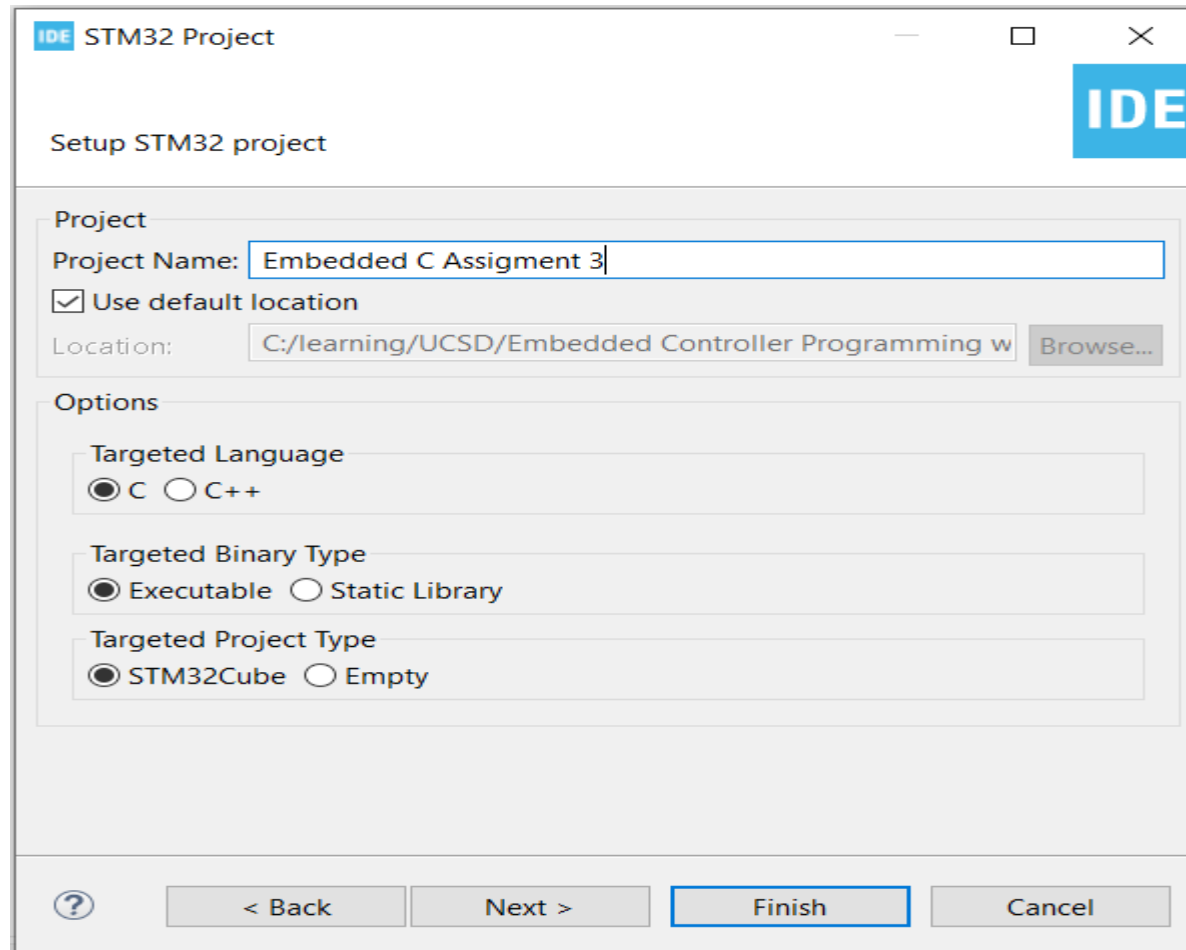
Step 1. Startup STM32CubeIDE and create new STM32 project



Step 2. Access board selector and type in the board you use, click Next



Step 3. Enter the project name then click Finish



The screenshot shows a dialog box titled "STM32 Project" with a subtitle "Setup STM32 project". The dialog is divided into two main sections: "Project" and "Options".

Project Section:

- Project Name:** A text field containing "Embedded C Assignment 3".
- Use default location:** A checked checkbox.
- Location:** A text field containing "C:/learning/UCSD/Embedded Controller Programming w" and a "Browse..." button.

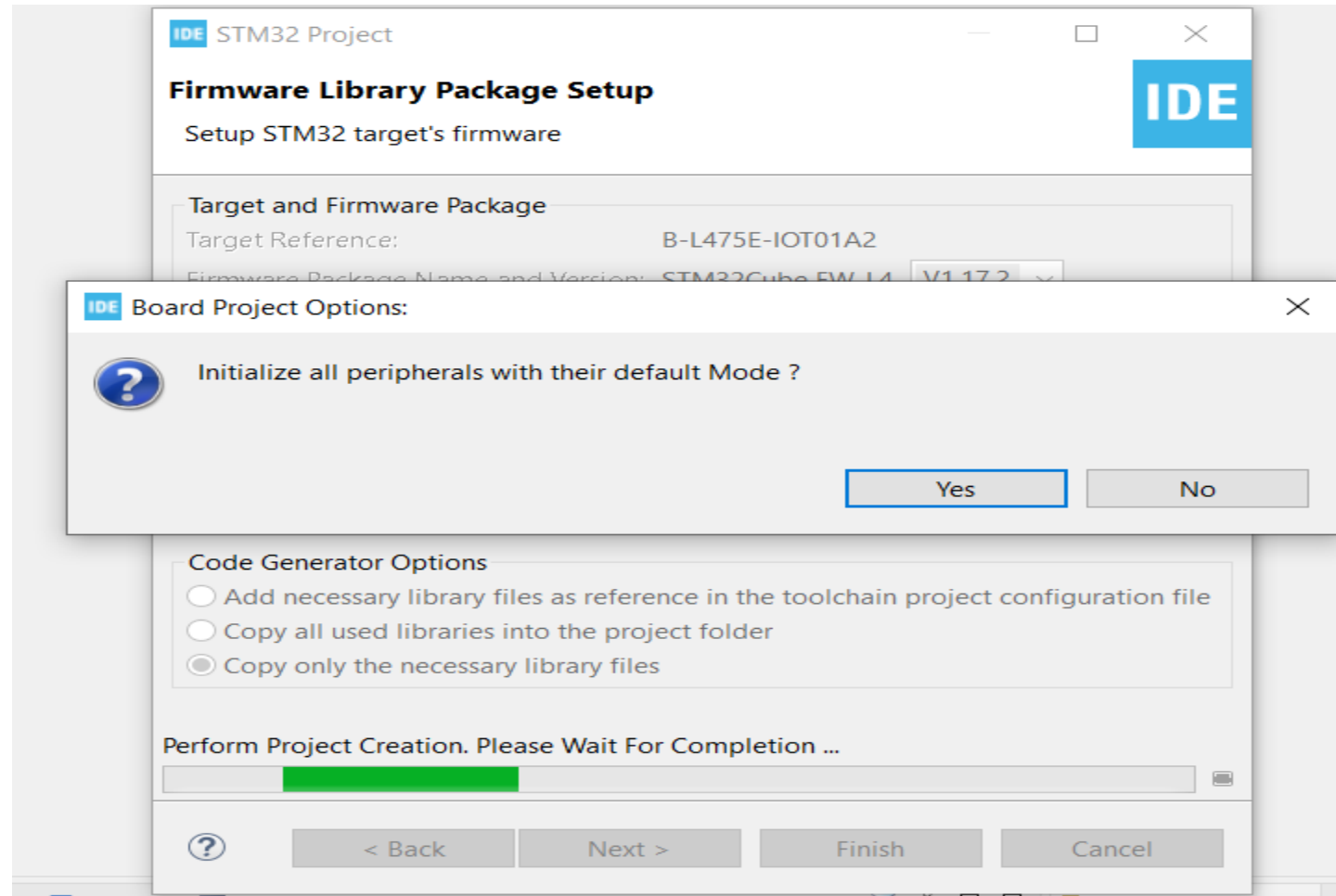
Options Section:

- Targeted Language:** Radio buttons for "C" (selected) and "C++".
- Targeted Binary Type:** Radio buttons for "Executable" (selected) and "Static Library".
- Targeted Project Type:** Radio buttons for "STM32Cube" (selected) and "Empty".

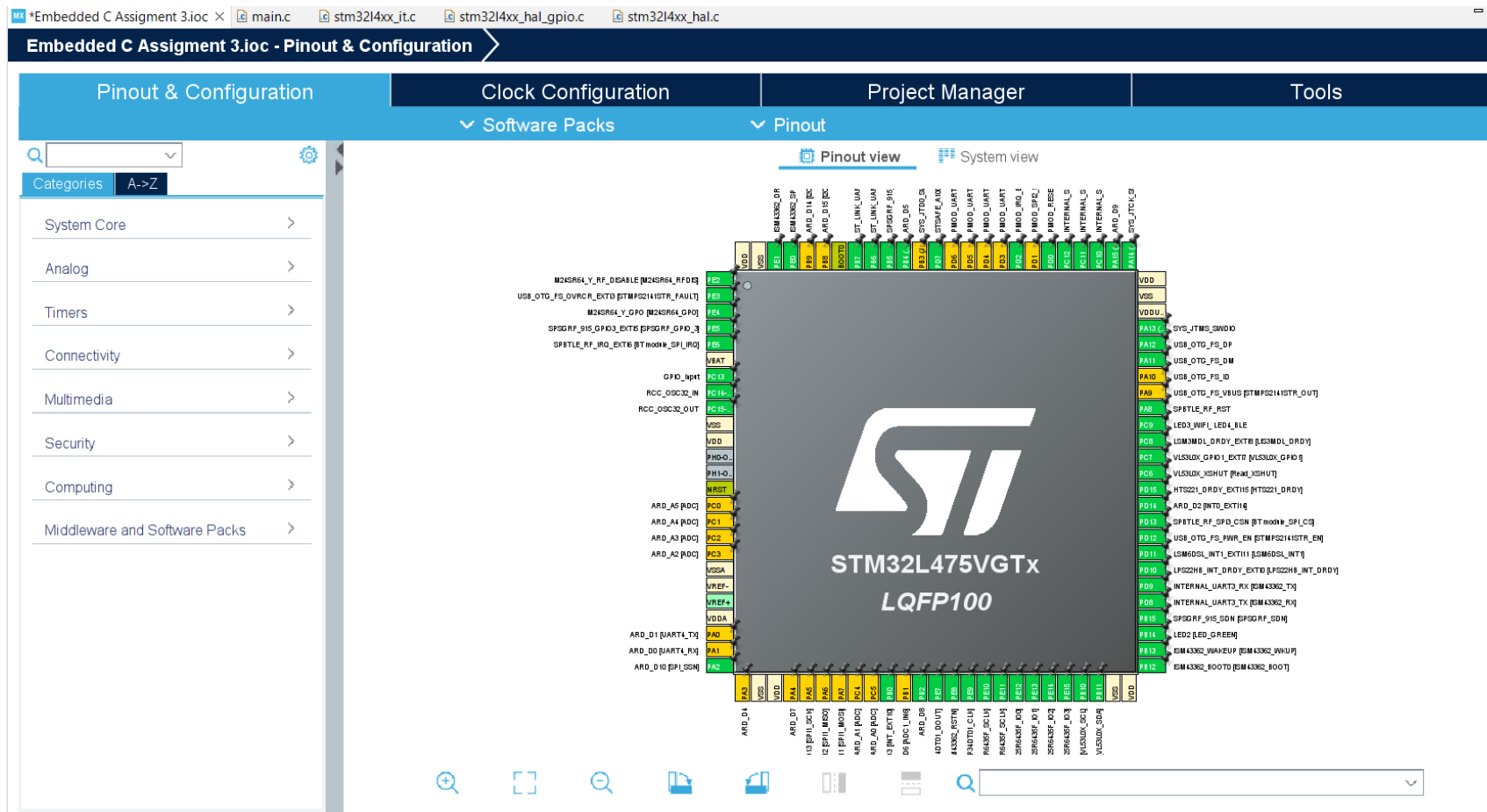
Bottom Bar:

- A help icon (?) on the left.
- Buttons for "< Back", "Next >", "Finish" (highlighted with a blue border), and "Cancel".

Step 4. Click yes to initialize all peripherals to default



Step 5. When in .ioc file, click Pinout & Configurations, make sure BUTTON and LED2 are configured correctly



Step 6. Click Project Manager -> Advanced Settings -> Driver Selector, make sure the required driver use HAL

Embedded C Assignment 3.ioc - Project Manager

Pinout & Configuration | Clock Configuration | **Project Manager** | Tools

Project

Code Generator

Advanced Settings

Driver Selector

Search (Ctrl+F)

RCC	HAL
GPIO	HAL
> DFSDM	HAL
> I2C	HAL
> QUADSPI	HAL
> SPI	HAL
> USART	HAL
USB_OTG_FS	HAL

Register CallBack

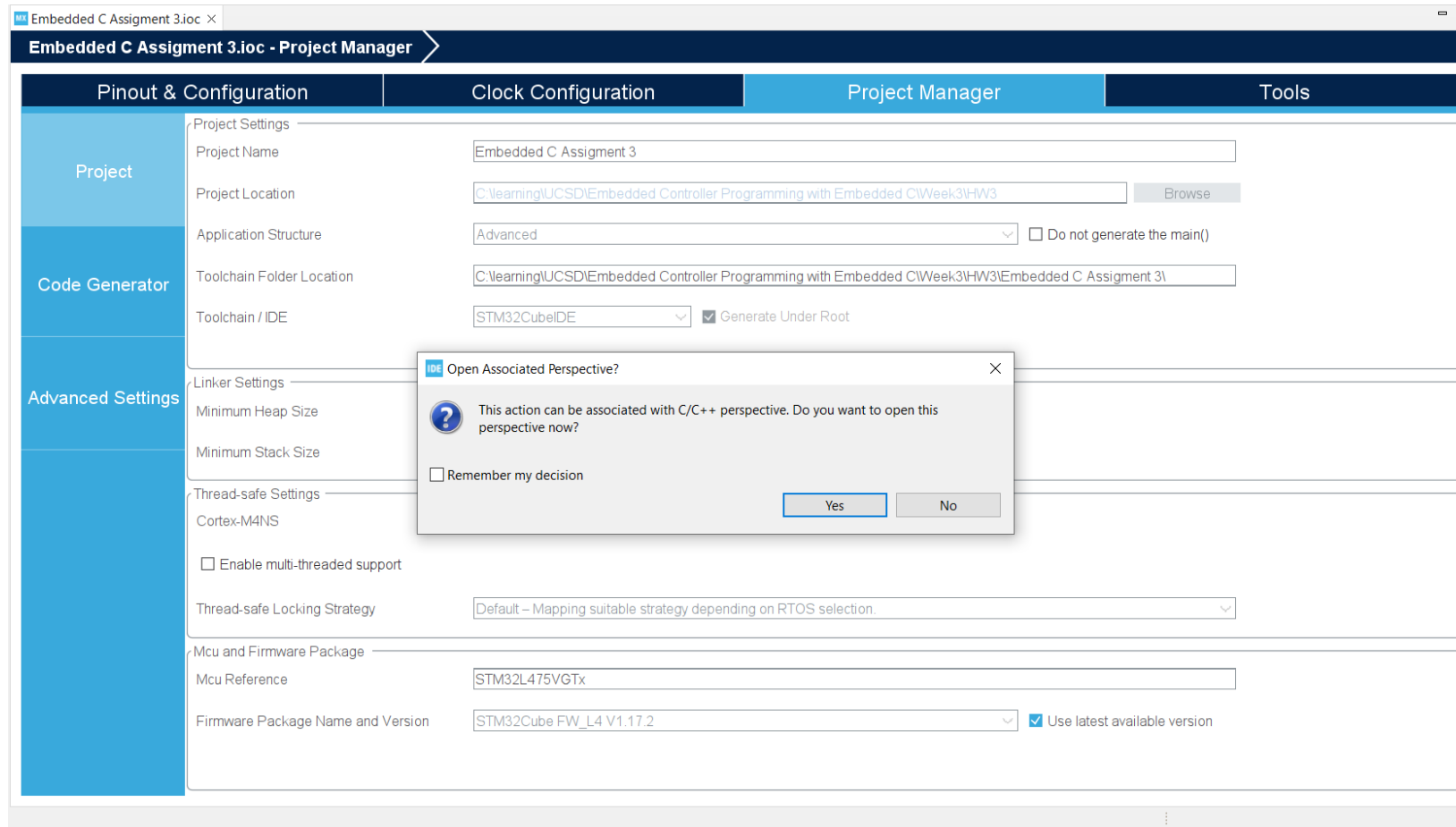
Search (Ctrl+F)

ADC	DISABLE
CAN	DISABLE
COMP	DISABLE
CRYP	DISABLE
DAC	DISABLE
DCMI	DISABLE
DFSDM	DISABLE
DMA2D	DISABLE
DSI	DISABLE
GFXMMU	DISABLE
HASH	DISABLE
HCD	DISABLE
I2C	DISABLE
IRDA	DISABLE
LPTIM	DISABLE
LTDC	DISABLE
MMC	DISABLE
OPAMP	DISABLE
OSPI	DISABLE
PCD	DISABLE
QSPI	DISABLE
RNG	DISABLE
RTC	DISABLE
SAI	DISABLE
SD	DISABLE
SMARTCARD	DISABLE
SMBUS	DISABLE
SPI	DISABLE
SWPMI	DISABLE

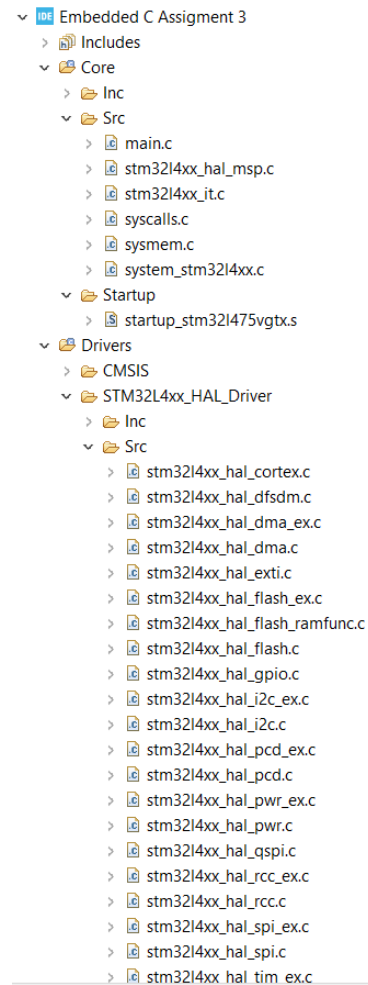
Generated Function Calls

Generate Code	Rank	Function Name	Peripheral Instance Na...	Do Not Generate Function Call	Visibility (Static)
<input checked="" type="checkbox"/>	1	SystemClock_Config	RCC	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	3	MX_DFSDM1_Init	DFSDM1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	4	MX_I2C2_Init	I2C2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	5	MX_QUADSPI_Init	QUADSPI	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	6	MX_SPI3_Init	SPI3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	7	MX_USART1_UART_I...	USART1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	8	MX_USART3_UART_I...	USART3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	9	MX_USB_OTG_FS_P...	USB_OTG_FS	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Step 7. Click yes to generate code and open in c/c++ perspective



Step 8. Observe the result project structure



```

Embedded C Assignment 3
├── Includes
├── Core
│   ├── Inc
│   └── Src
│       ├── main.c
│       ├── stm32l4xx_hal_msp.c
│       ├── stm32l4xx_it.c
│       ├── syscalls.c
│       ├── sysmem.c
│       └── system_stm32l4xx.c
├── Startup
│   └── startup_stm32l475vgtx.s
├── Drivers
│   ├── CMSIS
│   └── STM32L4xx_HAL_Driver
│       ├── Inc
│       └── Src
│           ├── stm32l4xx_hal_cortex.c
│           ├── stm32l4xx_hal_dfsdm.c
│           ├── stm32l4xx_hal_dma_ex.c
│           ├── stm32l4xx_hal_dma.c
│           ├── stm32l4xx_hal_exti.c
│           ├── stm32l4xx_hal_flash_ex.c
│           ├── stm32l4xx_hal_flash_ramfunc.c
│           ├── stm32l4xx_hal_flash.c
│           ├── stm32l4xx_hal_gpio.c
│           ├── stm32l4xx_hal_i2c_ex.c
│           ├── stm32l4xx_hal_i2c.c
│           ├── stm32l4xx_hal_pcd_ex.c
│           ├── stm32l4xx_hal_pcd.c
│           ├── stm32l4xx_hal_pwr_ex.c
│           ├── stm32l4xx_hal_pwr.c
│           ├── stm32l4xx_hal_qspi.c
│           ├── stm32l4xx_hal_rcc_ex.c
│           ├── stm32l4xx_hal_rcc.c
│           ├── stm32l4xx_hal_spi_ex.c
│           ├── stm32l4xx_hal_spi.c
│           └── stm32l4xx_hal_tim_ex.c

```

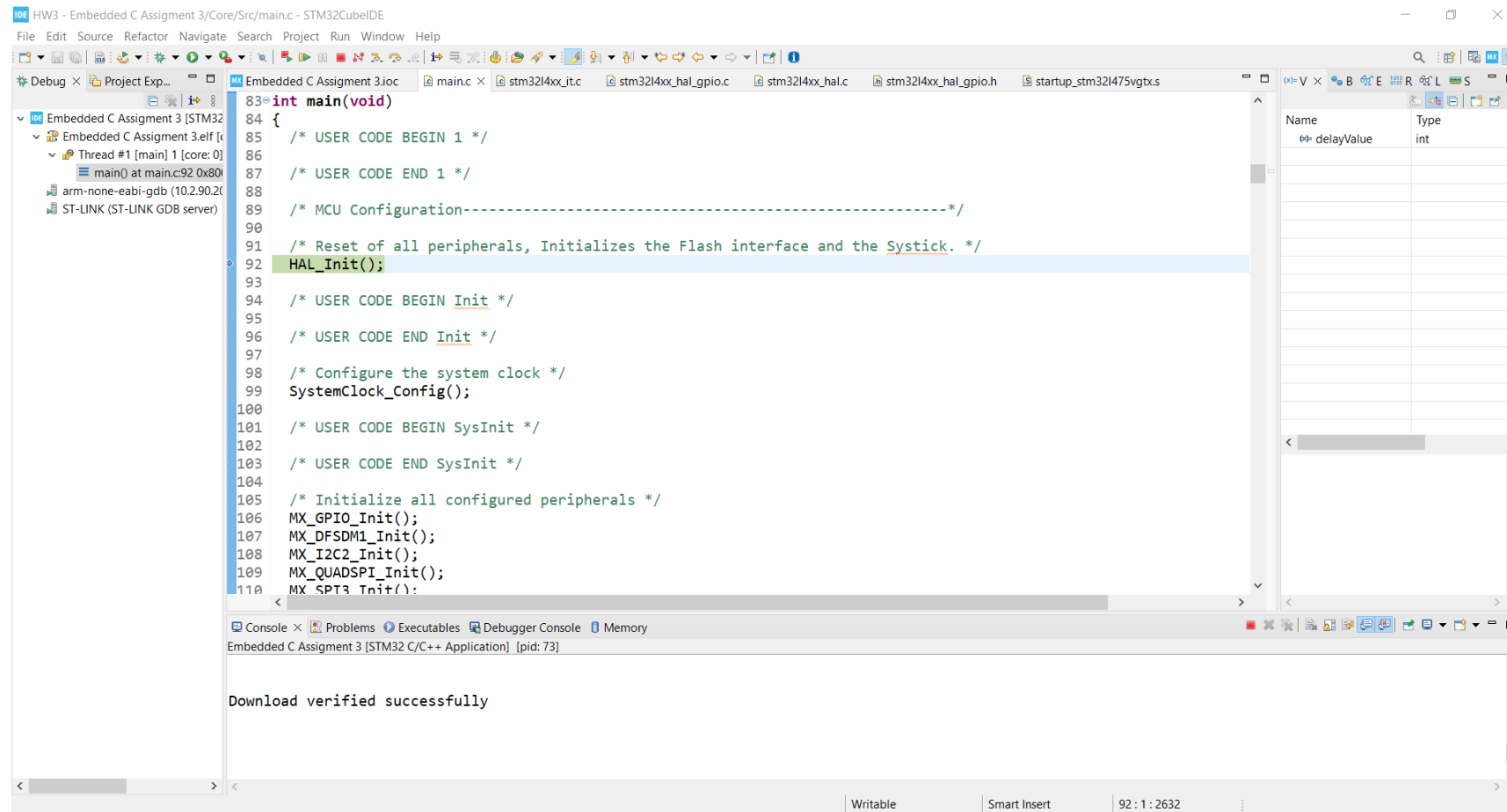
The image shows a project explorer in an IDE for a project named "Embedded C Assignment 3". The structure is as follows:

- Embedded C Assignment 3
 - Includes
 - Core
 - Inc
 - Src
 - main.c
 - stm32l4xx_hal_msp.c
 - stm32l4xx_it.c
 - syscalls.c
 - sysmem.c
 - system_stm32l4xx.c
 - Startup
 - startup_stm32l475vgtx.s
 - Drivers
 - CMSIS
 - STM32L4xx_HAL_Driver
 - Inc
 - Src
 - stm32l4xx_hal_cortex.c
 - stm32l4xx_hal_dfsdm.c
 - stm32l4xx_hal_dma_ex.c
 - stm32l4xx_hal_dma.c
 - stm32l4xx_hal_exti.c
 - stm32l4xx_hal_flash_ex.c
 - stm32l4xx_hal_flash_ramfunc.c
 - stm32l4xx_hal_flash.c
 - stm32l4xx_hal_gpio.c
 - stm32l4xx_hal_i2c_ex.c
 - stm32l4xx_hal_i2c.c
 - stm32l4xx_hal_pcd_ex.c
 - stm32l4xx_hal_pcd.c
 - stm32l4xx_hal_pwr_ex.c
 - stm32l4xx_hal_pwr.c
 - stm32l4xx_hal_qspi.c
 - stm32l4xx_hal_rcc_ex.c
 - stm32l4xx_hal_rcc.c
 - stm32l4xx_hal_spi_ex.c
 - stm32l4xx_hal_spi.c
 - stm32l4xx_hal_tim_ex.c

Step 9. Edit the code the main.c file, if BUTTON is in RESET state, toggle the LED2 pin in 1000ms, else, toggle in 250 ms

```
112  MX_USART3_UART_Init();
113  MX_USB_OTG_FS_PCD_Init();
114  /* USER CODE BEGIN 2 */
115  int delayValue;
116  /* USER CODE END 2 */
117
118  /* Infinite loop */
119  /* USER CODE BEGIN WHILE */
120  while (1)
121  {
122      /* USER CODE END WHILE */
123
124      /* USER CODE BEGIN 3 */
125      if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_SET)
126      {
127          delayValue = 250;
128      }
129      else
130      {
131          delayValue = 1000;
132      }
133      HAL_GPIO_TogglePin(GPIOB, LED2_Pin);
134      HAL_Delay(delayValue);
135  }
136  /* USER CODE END 3 */
137 }
138
```

Step 10. Build the project, debug and run the code. When you push the user button, you should see the LED toggle rate change



The screenshot shows the STM32CubeIDE interface. The main window displays the `main.c` file with the following code:

```
83 int main(void)
84 {
85     /* USER CODE BEGIN 1 */
86     /* USER CODE END 1 */
87
88     /* MCU Configuration-----*/
89
90     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
91     HAL_Init();
92
93     /* USER CODE BEGIN Init */
94     /* USER CODE END Init */
95
96     /* Configure the system clock */
97     SystemClock_Config();
98
99     /* USER CODE BEGIN SysInit */
100    /* USER CODE END SysInit */
101
102    /* Initialize all configured peripherals */
103    MX_GPIO_Init();
104    MX_DFSDM1_Init();
105    MX_I2C2_Init();
106    MX_QUADSPI_Init();
107    MX_SPT3_Init();
108}
```

The `HAL_Init();` line is highlighted. The left sidebar shows the project structure with `main.c` selected. The bottom console window displays the message: "Download verified successfully".