

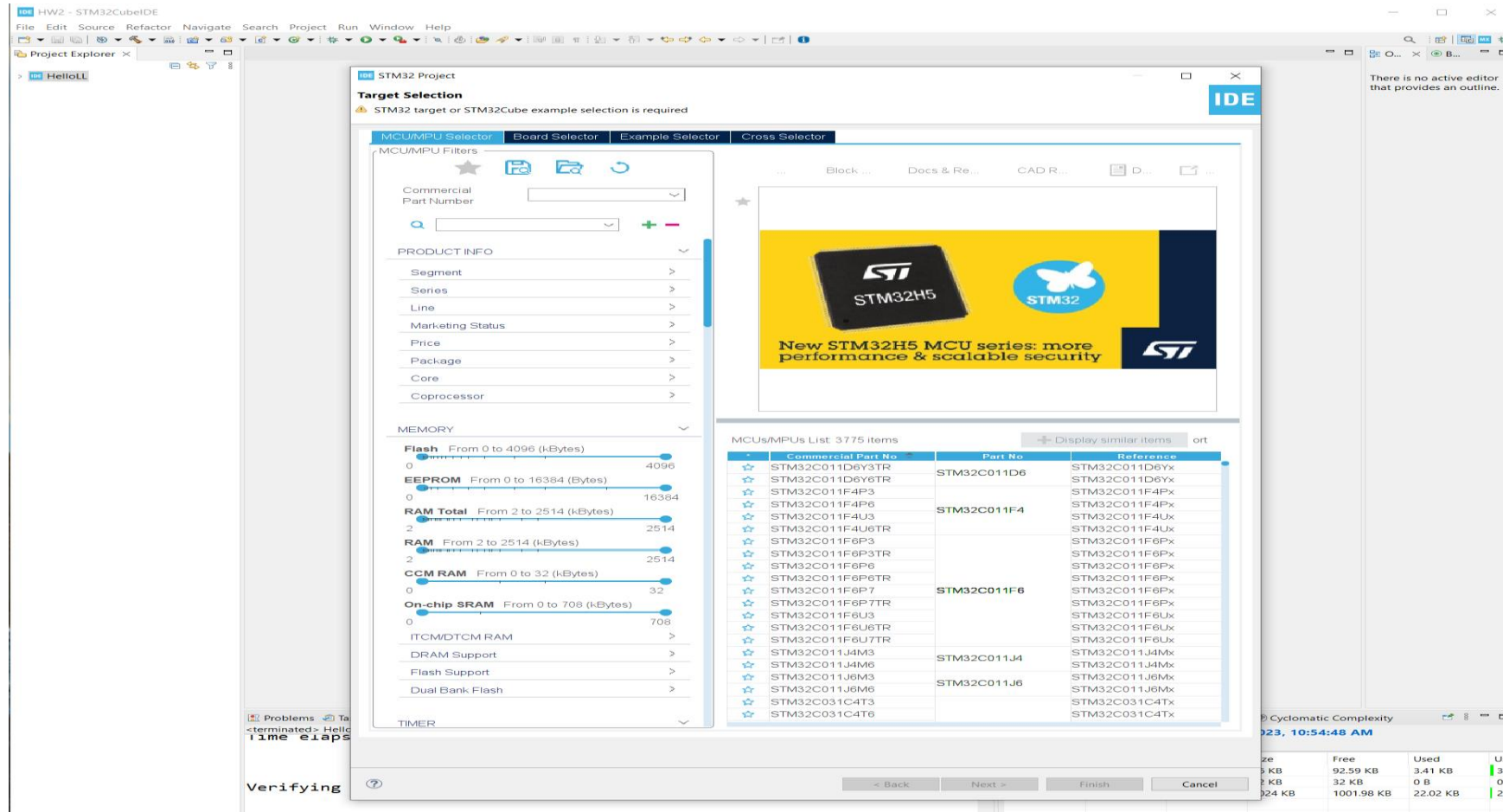
USCD Embedded C Assignment 2

By

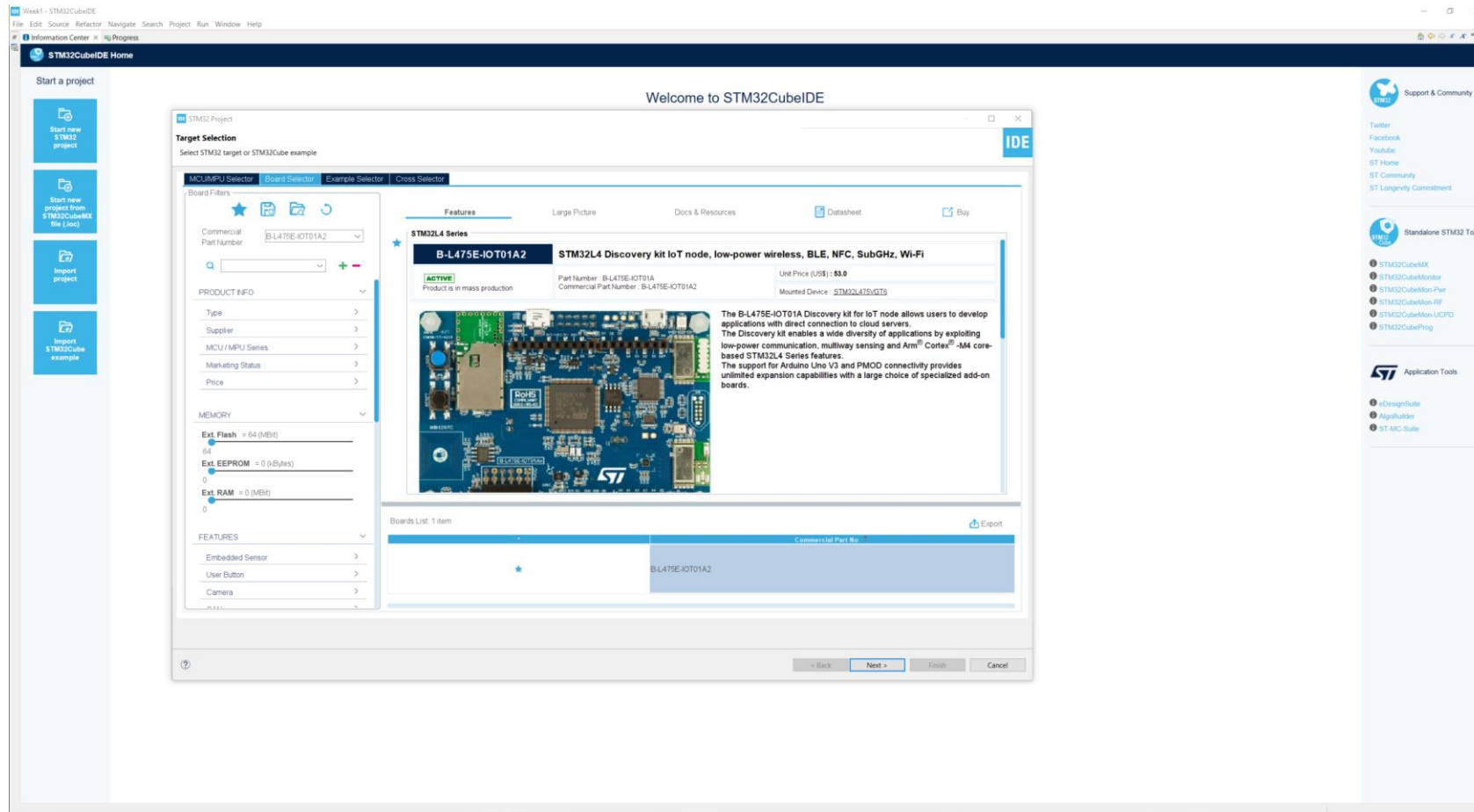
Hsuankai Chang

hsuankac@umich.edu

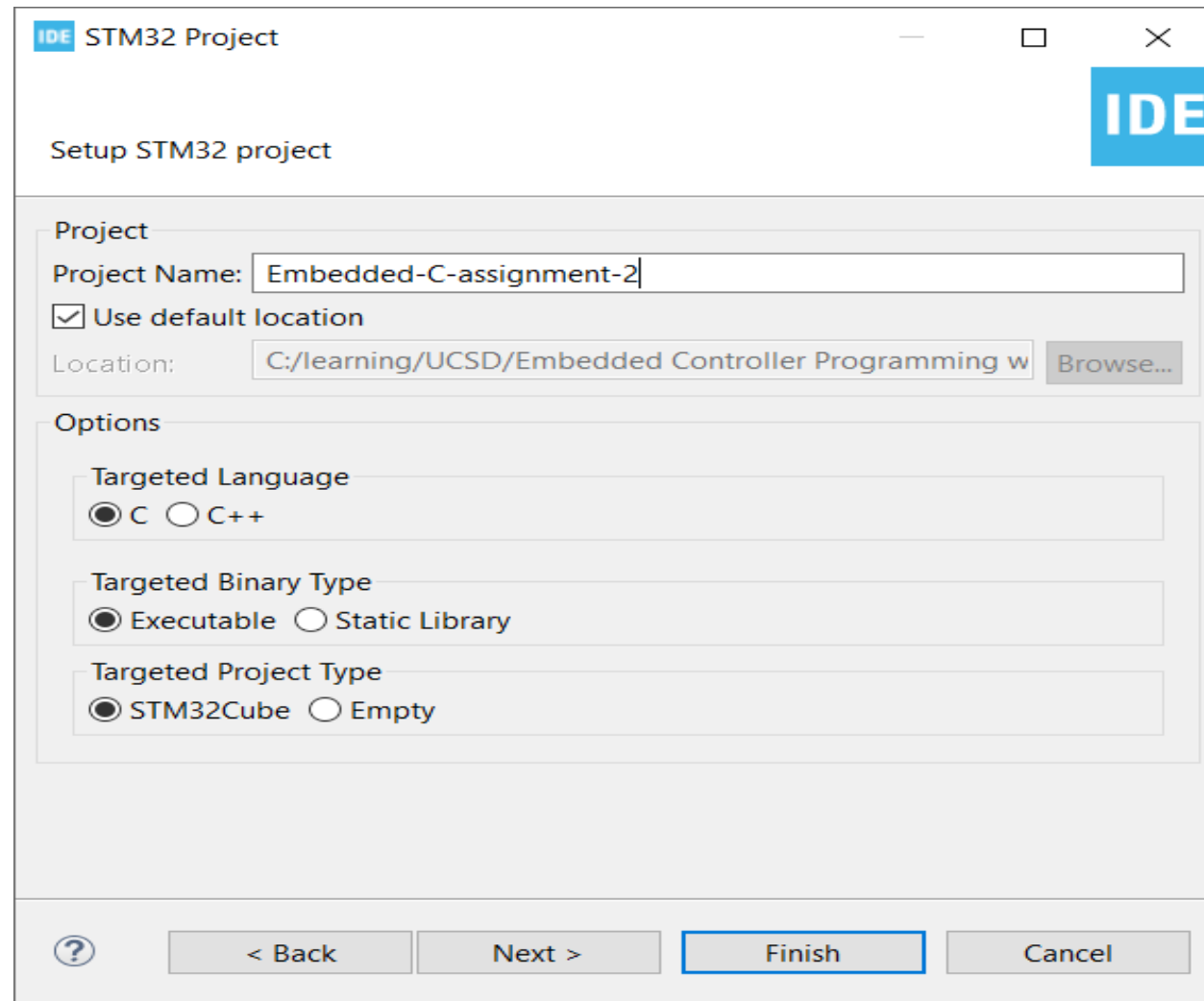
Step 1. Startup STM32CubeIDE and create new STM32 project



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



Step 4. Enter the project name then click Finish



The image shows a 'Setup STM32 project' dialog box from the IDE. The window title is 'IDE STM32 Project'. The main title is 'Setup STM32 project'. The 'Project' section contains a 'Project Name' field with the text 'Embedded-C-assignment-2', a checked 'Use default location' checkbox, and a 'Location' field with the path 'C:/learning/UCSD/Embedded Controller Programming w' and a 'Browse...' button. 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 buttons for '?', '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

IDE STM32 Project

Setup STM32 project

Project

Project Name: Embedded-C-assignment-2

☒ Use default location

Location: C:/learning/UCSD/Embedded Controller Programming w Browse...

Options

Targeted Language

☒ C ☐ C++

Targeted Binary Type

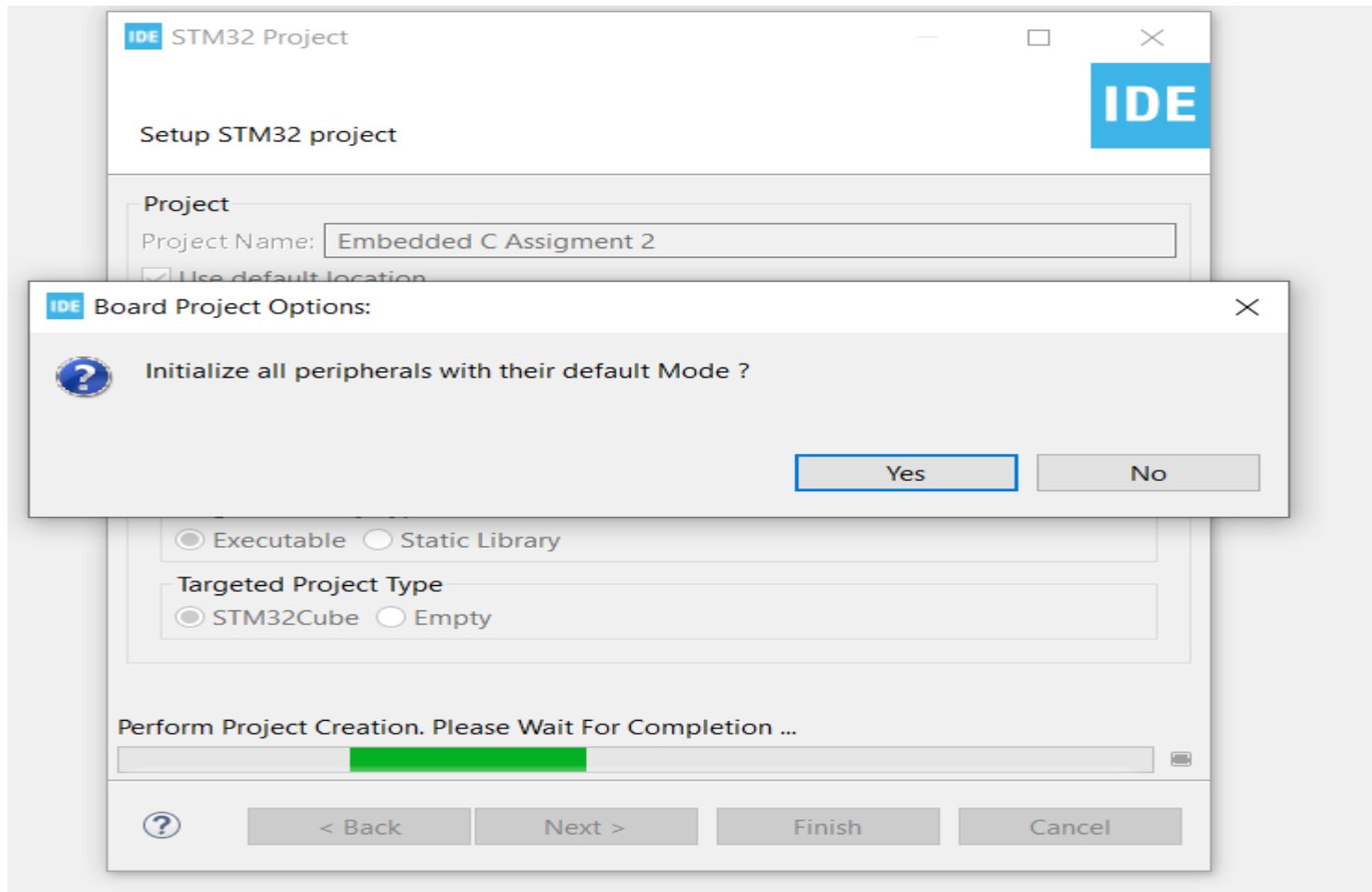
☒ Executable ☐ Static Library

Targeted Project Type

☒ STM32Cube ☐ Empty

? < Back Next > Finish Cancel

Step 5. Click yes to initialize all peripherals to default



Step 6. When in .ioc file, click Project Manager -> advanced settings -> Driver Selector and change the GPIO driver to use LL

The screenshot shows the STM32CubeIDE Project Manager window for a project named "Embedded C Assignment 2.ioc". The "Project Manager" tab is active, and the "Advanced Settings" section is expanded. The "Driver Selector" section is visible, showing a list of drivers and their status. The "Register Callback" section is also visible, showing a list of callbacks and their status.

Driver Selector

Driver	Status
RCC	HAL
GPIO	LL
DFSDM	HAL
I2C	HAL
QUADSPI	HAL
SPI	HAL
USART	HAL
USB_OTG_FS	HAL

Register Callback

Callback	Status
ADC	DISABLE
CAN	DISABLE
COMP	DISABLE
CRYP	DISABLE
DAC	DISABLE
DCMI	DISABLE
DFSDM	DISABLE
DMA2D	DISABLE
DSI	DISABLE
QFMMIO	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
TIM	DISABLE
TSC	DISABLE
UART	DISABLE
USART	DISABLE
WWDG	DISABLE

Generated Function Calls

Generate Code	Rank	Function Name	Peripheral Instance N.	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	USART1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	8	MX_USART3_UART	USART3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	9	MX_USB_OTG_FS...	USB_OTG_FS	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Step 7. Save and click yes to generate code

Code Generator

Advanced Settings

Question

Do you want generate Code?

☐ Remember my decision

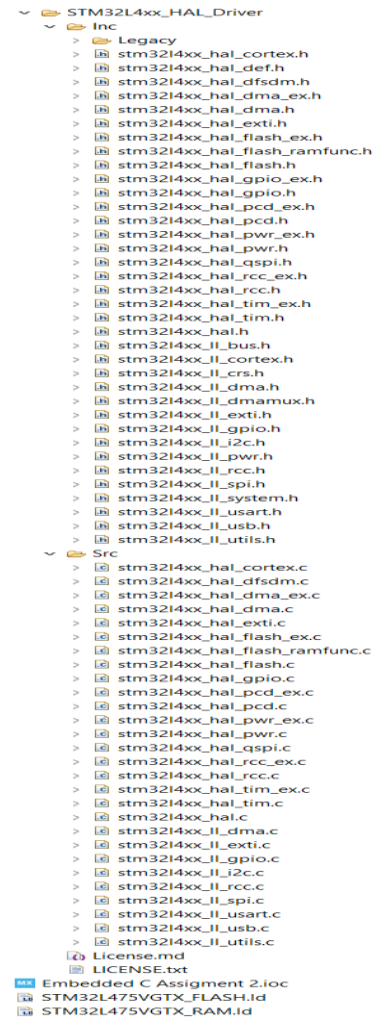
YesNo

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
TIM	DISABLE
TSC	DISABLE
UART	DISABLE
USART	DISABLE
WWDG	DISABLE

Generated Function Calls

Generate Code	Rank	Function Name	Peripheral Instance N.	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_Init	USART1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	8	MX_USART3_UART_Init	USART3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	9	MX_USB_OTG_FS_Init	USB_OTG_FS	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Step 8. See if we have all the needed LL file included in the file folder



Step 9. Find GPIO Init code that initializes the LED2 using LL (highlighted in blue below)

```
530 }
531
532 /**
533  * @brief GPIO Initialization Function
534  * @param None
535  * @retval None
536  */
537 static void MX_GPIO_Init(void)
538 {
539     LL_EXTI_InitTypeDef EXTI_InitStructure = {0};
540     LL_GPIO_InitTypeDef GPIO_InitStructure = {0};
541     /* USER CODE BEGIN MX_GPIO_Init_1 */
542     /* USER CODE END MX_GPIO_Init_1 */
543
544     /* GPIO Ports Clock Enable */
545     LL_AHB2_GRP1_EnableClock(LL_AHB2_GRP1_PERIPH_GPIOE);
546     LL_AHB2_GRP1_EnableClock(LL_AHB2_GRP1_PERIPH_GPIOC);
547     LL_AHB2_GRP1_EnableClock(LL_AHB2_GRP1_PERIPH_GPIOA);
548     LL_AHB2_GRP1_EnableClock(LL_AHB2_GRP1_PERIPH_GPIOB);
549     LL_AHB2_GRP1_EnableClock(LL_AHB2_GRP1_PERIPH_GPIOD);
550
551     /**/
552     LL_GPIO_ResetOutputPin(GPIOE, M24SR64_Y_RF_DISABLE_Pin|M24SR64_Y_GPO_Pin|ISM43362_RST_Pin);
553
554     /**/
555     LL_GPIO_ResetOutputPin(GPIOA, ARD_D10_Pin|SPBTLE_RF_RST_Pin|ARD_D9_Pin);
556
557     /**/
558     LL_GPIO_ResetOutputPin(GPIOB, ARD_D8_Pin|ISM43362_BOOT0_Pin|ISM43362_WAKEUP_Pin|LED2_Pin
559                             |SPSGRF_915_SDN_Pin|ARD_D5_Pin);
560
561     /**/
562     LL_GPIO_ResetOutputPin(GPIOD, USB_OTG_FS_PWR_EN_Pin|PMOD_RESET_Pin|STSAFE_A100_RESET_Pin);
563
564     /**/
565     LL_GPIO_ResetOutputPin(GPIOC, VL53L0X_XSHUT_Pin|LED3_WIFI__LED4_BLE_Pin);
566
567     /**/
568     LL_GPIO_SetOutputPin(SPBTLE_RF_SPI3_CSN_GPIO_Port, SPBTLE_RF_SPI3_CSN_Pin);
569
570     /**/
571     LL_GPIO_SetOutputPin(SPSGRF_915_SPI3_CSN_GPIO_Port, SPSGRF_915_SPI3_CSN_Pin);
572
573     /**/
574     LL_GPIO_SetOutputPin(ISM43362_SPI3_CSN_GPIO_Port, ISM43362_SPI3_CSN_Pin);
575
576     /**/
577     GPIO_InitStructure.Pin = M24SR64_Y_RF_DISABLE_Pin|M24SR64_Y_GPO_Pin|ISM43362_RST_Pin|ISM43362_SP
578     GPIO_InitStructure.Mode = LL_GPIO_MODE_OUTPUT;
579     GPIO_InitStructure.Speed = LL_GPIO_SPEED_FREQ_LOW;
```

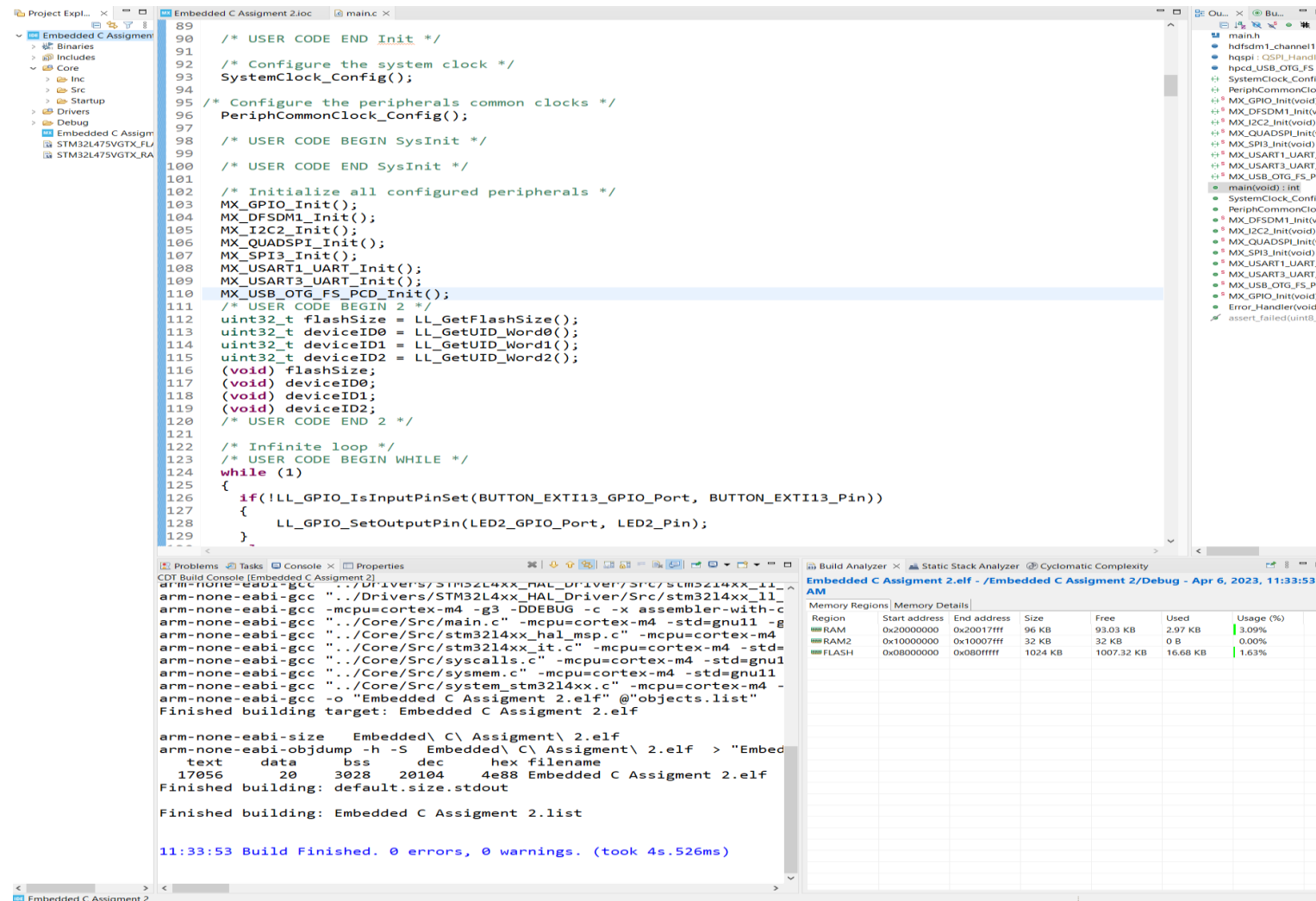
Step 10. Enter the code in CODE BEGIN section to get the flash size and device ID per requirement

```
Embedded C Assignment 2.ioc  main.c
89
90  /* USER CODE END Init */
91
92  /* Configure the system clock */
93  SystemClock_Config();
94
95  /* Configure the peripherals common clocks */
96  PeriphCommonClock_Config();
97
98  /* USER CODE BEGIN SysInit */
99
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_SPI3_Init();
108 MX_USART1_UART_Init();
109 MX_USART3_UART_Init();
110 MX_USB_OTG_FS_PCD_Init();
111 /* USER CODE BEGIN 2 */
112 uint32_t flashSize = LL_GetFlashSize();
113 uint32_t deviceID0 = LL_GetUID_Word0();
114 uint32_t deviceID1 = LL_GetUID_Word1();
115 uint32_t deviceID2 = LL_GetUID_Word2();
116 (void) flashSize;
117 (void) deviceID0;
118 (void) deviceID1;
119 (void) deviceID2;
120 /* USER CODE END 2 */
121
122 /* Infinite loop */
123 /* USER CODE BEGIN WHILE */
124 while (1)
125 {
126     if(!LL_GPIO_IsInputPinSet(BUTTON_EXTI13_GPIO_Port, BUTTON_EXTI13_Pin))
127     {
128         LL_GPIO_SetOutputPin(LED2_GPIO_Port, LED2_Pin);
129     }
130     else
131     {
132         LL_GPIO_ResetOutputPin(LED2_GPIO_Port, LED2_Pin);
133     }
134     /* USER CODE END WHILE */
135
136     /* USER CODE BEGIN 3 */
137 }
138 /* USER CODE END 3 */
```

Step 11. If the blue button is pushed, turn on the LED2, else, turn it off

```
Embedded C Assignment 2.ioc  main.c x
89
90 /* USER CODE END Init */
91
92 /* Configure the system clock */
93 SystemClock_Config();
94
95 /* Configure the peripherals common clocks */
96 PeriphCommonClock_Config();
97
98 /* USER CODE BEGIN SysInit */
99
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_SPI3_Init();
108 MX_USART1_UART_Init();
109 MX_USART3_UART_Init();
110 MX_USB_OTG_FS_PCD_Init();
111 /* USER CODE BEGIN 2 */
112 uint32_t flashSize = LL_GetFlashSize();
113 uint32_t deviceID0 = LL_GetUID_Word0();
114 uint32_t deviceID1 = LL_GetUID_Word1();
115 uint32_t deviceID2 = LL_GetUID_Word2();
116 (void) flashSize;
117 (void) deviceID0;
118 (void) deviceID1;
119 (void) deviceID2;
120 /* USER CODE END 2 */
121
122 /* Infinite loop */
123 /* USER CODE BEGIN WHILE */
124 while (1)
125 {
126     if(!LL_GPIO_IsInputPinSet(BUTTON_EXTI13_GPIO_Port, BUTTON_EXTI13_Pin))
127     {
128         LL_GPIO_SetOutputPin(LED2_GPIO_Port, LED2_Pin);
129     }
130     else
131     {
132         LL_GPIO_ResetOutputPin(LED2_GPIO_Port, LED2_Pin);
133     }
134     /* USER CODE END WHILE */
135
136     /* USER CODE BEGIN 3 */
137 }
138 /* USER CODE END 3 */
```

Step 12. Right click on the project name and build it, test the .elf file, also examine the RAM and FLASH size



Step 13. Setup for debug configuration, test on the board. If you press the blue button, the LED will be on

The screenshot displays an IDE with three main panels:

- Left Panel (Project Explorer):** Shows the project structure for 'Embedded-C-Assignment-1'. The 'Debug' configuration is selected.
- Center Panel (Code Editor):** Displays the C code for 'main.c'. The code includes initialization for peripherals, GPIO, and a while loop that toggles an LED pin.
- Right Panel (Debug Configurations):** Shows the 'Debug Configurations' dialog for 'Embedded-C-Assignment-1 Debug'. The 'Main' tab is active, showing settings for the GDB connection (ST-Link), interface (SWD), and device settings (Cortex-M4).

Below the code editor, the 'Problems' and 'Console' tabs are visible. The 'Console' tab shows the build output, indicating that the target 'Embedded-C-Assignment-1.elf' was successfully built.

```
arm-none-eabi-size Embedded-C-Assignment-1.elf
arm-none-eabi-objdump -h -S Embedded-C-Assignment-1.elf > "Embedded-C-Assignment-1.list"
18896 20 3476 22392 5778 Embedded-C-Assignment-1.elf
Finished building: default.size.stdout

Finished building: Embedded-C-Assignment-1.list

13:49:02 Build Finished. 0 errors, 0 warnings. (took 4s.931ms)
```

At the bottom right, the 'Build Targets' panel shows the memory regions for the target:

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM	0x20000000	0x20017FFF	96 KB	92.59 KB	3.41 KB	3.55%
RAM2	0x10000000	0x10007FFF	32 KB	32 KB	0 B	0.00%
FLASH	0x08000000	0x0800FFFF	1024 KB	1005.53 KB	18.47 KB	1.80%