

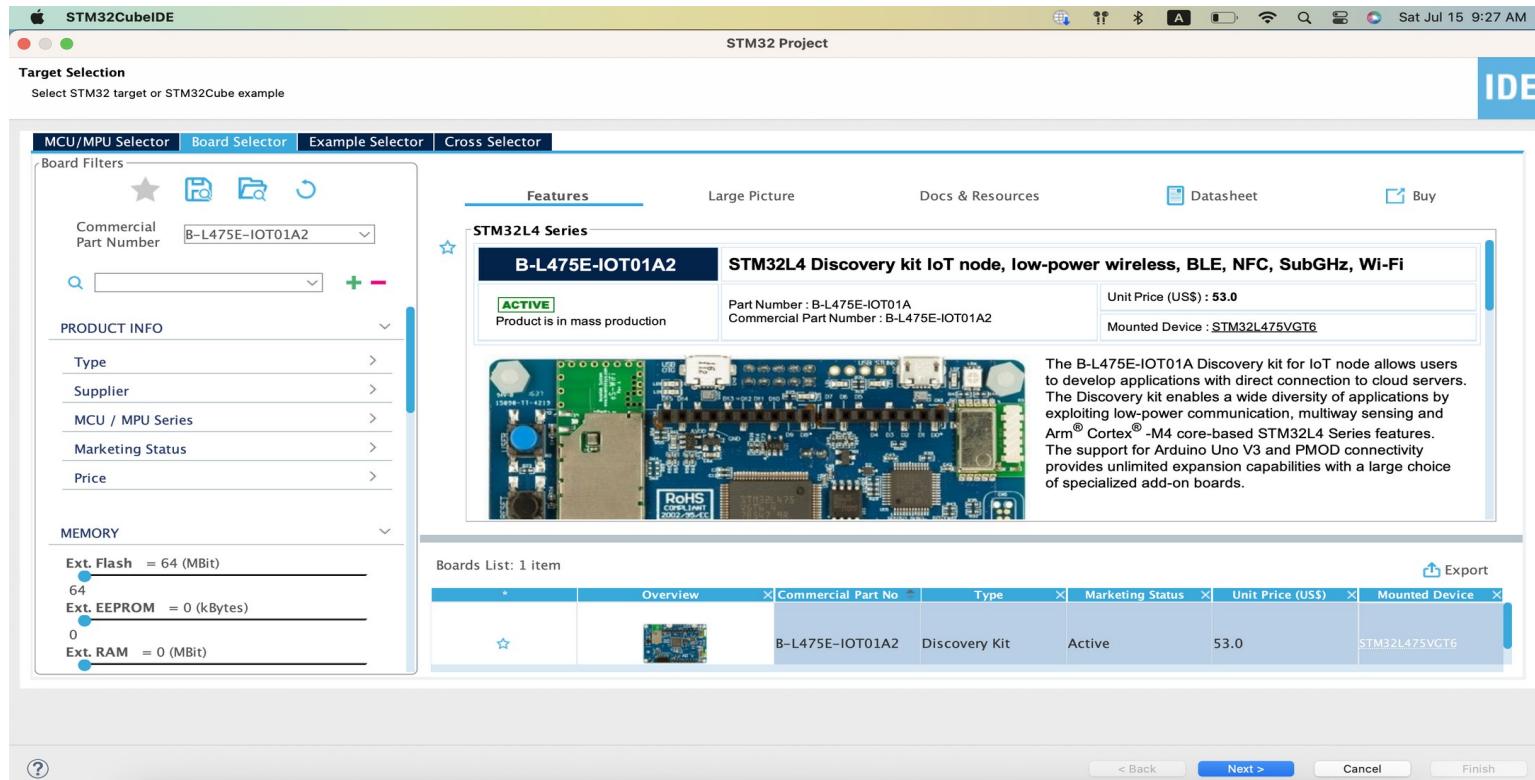
UCSD Embedded C Assignment 1

By

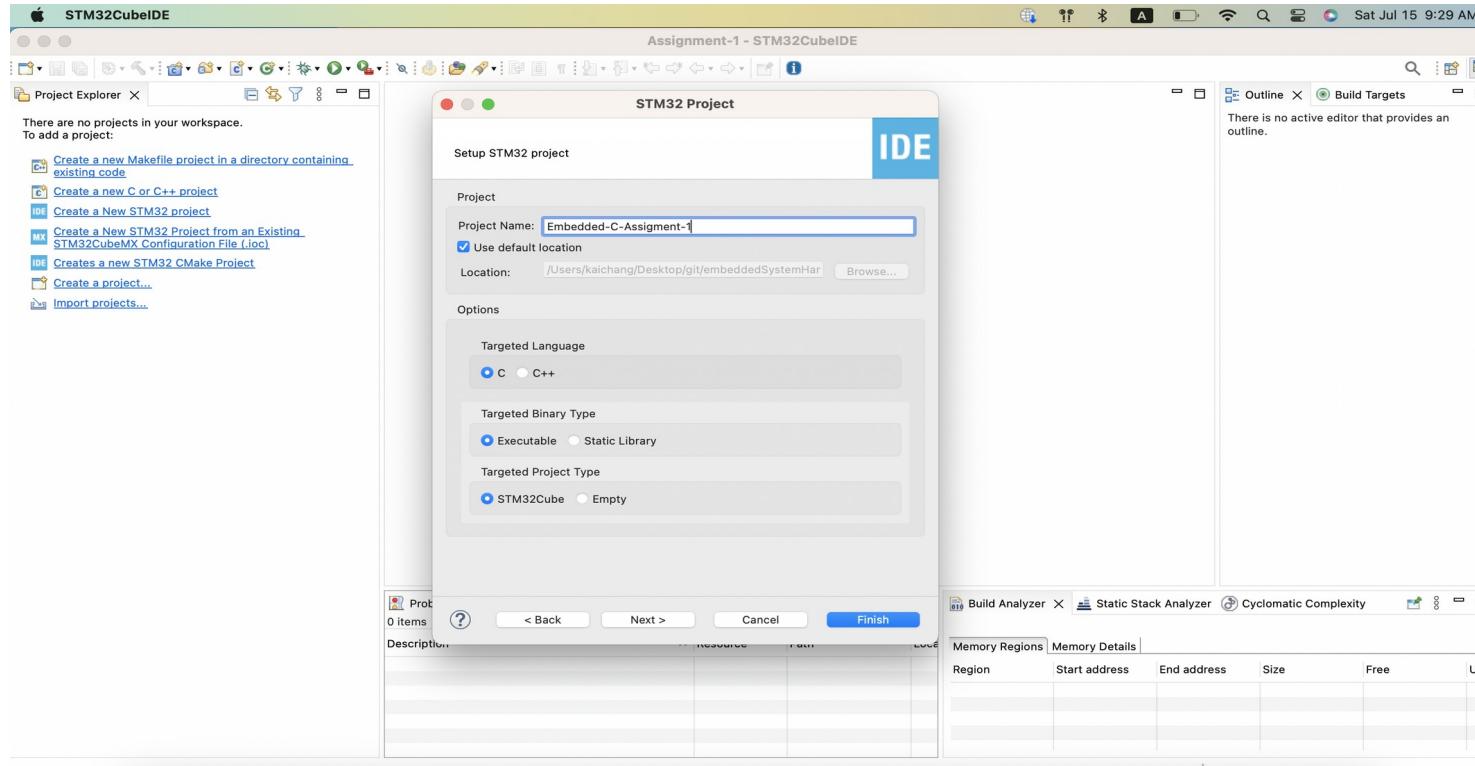
Hsuankai Chang

hsuankac@umich.edu

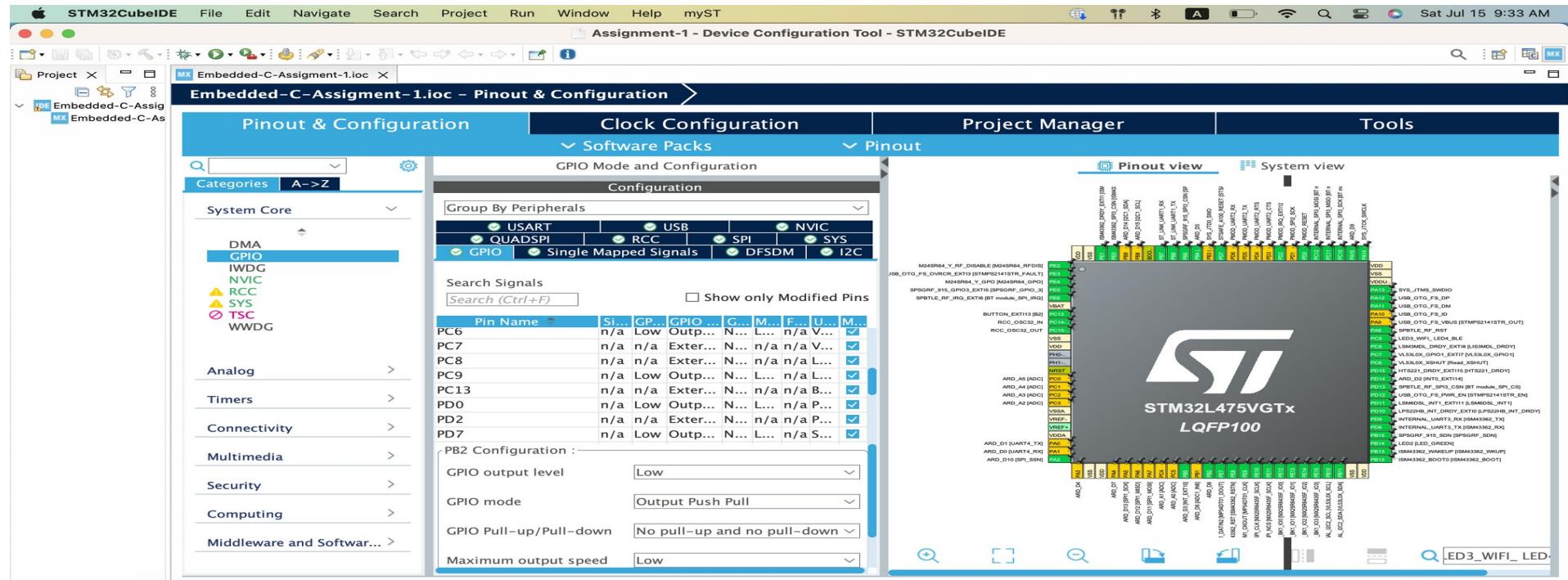
1. Startup STM32Cube IDE and select the correct board



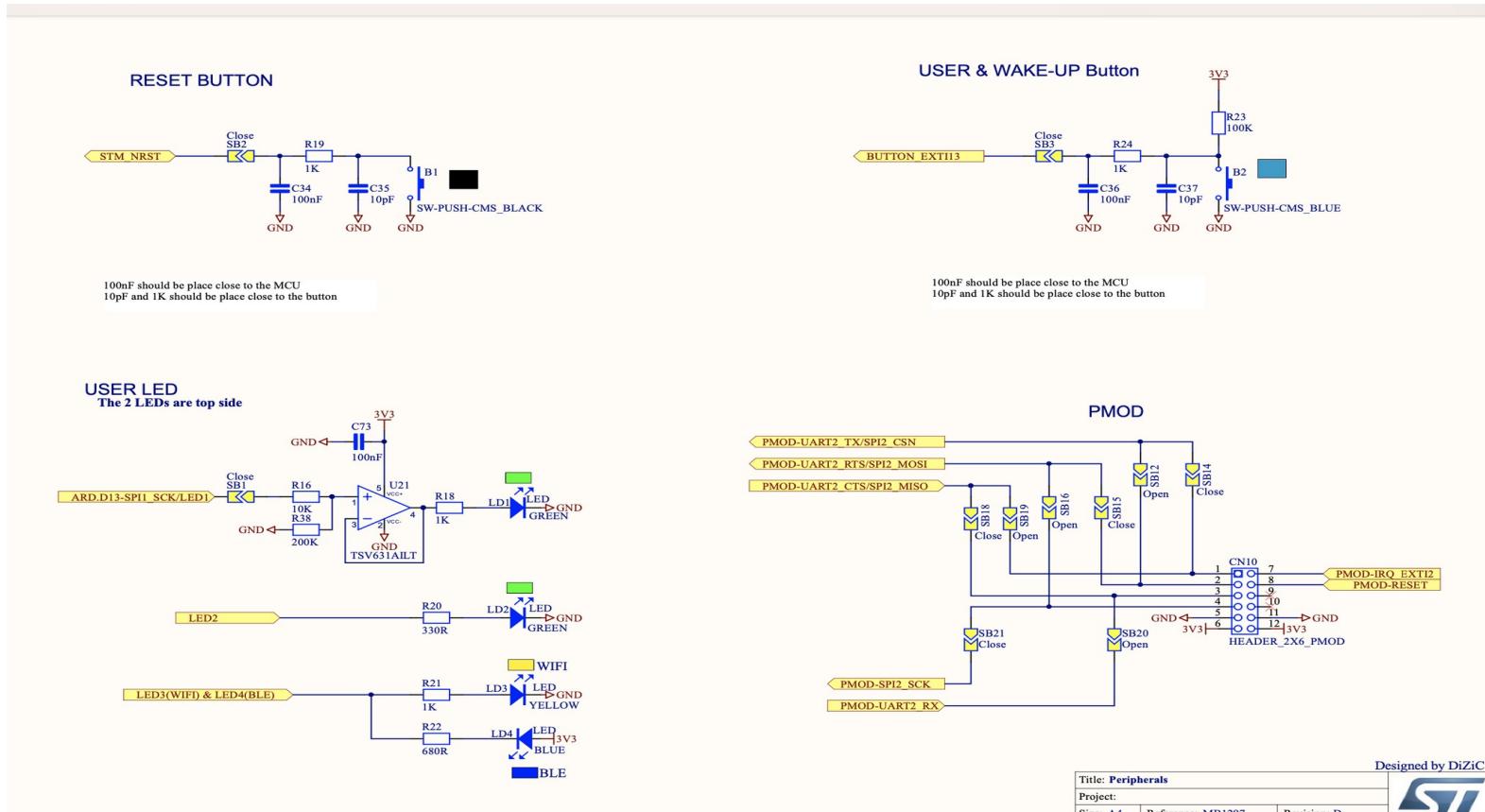
2. Enter project name and click finish to generate ioc file



3. In the ioc file, double check the pin configuration for button interrupt, LED2 and LED3, then generate code



4. Double check the pin number and configuration for push button and LED in the schematic



5. Reviewing the file structure when code is generated

The screenshot shows the STM32CubeIDE interface with the following details:

- Project Explorer:** Displays the project structure for "Assignment-1 - Embedded-C-Assigment-1". It includes the Core, Drivers, CMSIS, and STM32L4xx_HAL_Driver components, along with their respective sub-directories like Inc, Src, and Startup.
- Code Editor (main.c):** Shows the main.c file content. The code includes the standard header guard, copyright notice, and license information. It also includes #include "main.h" and various #include statements for HAL components. The code is annotated with comments indicating sections for User Code Header, Includes, Private Includes, and Private typedef.
- Outline View:** Lists the symbols defined in the current file, such as main.h, DFSDM_ChannelHandle, I2C_HandleTypeDef, SPI_HandleTypeDef, USART_HandleTypeDef, and GPIO_InitTypeDef.
- Build Analyzer:** Shows the build targets and their associated functions, including MX_GPIO_Init, MX_DFSDM1_Init, MX_I2C2_Init, MX_QUADSPI_Init, MX_SPI3_Init, MX_USART1_UART_Init, MX_USART3_UART_Init, MX_USB_OTG_FS_PCD_Init, main, SystemClock_Config, MX_DFSDM1_Init, MX_I2C2_Init, MX_QUADSPI_Init, MX_SPI3_Init, MX_USART1_UART_Init, MX_USART3_UART_Init, MX_USB_OTG_FS_PCD_Init, MX_GPIO_Init, and Error_Handler.
- Problems:** Shows 0 errors, 1 warning, and 0 others.
- Build Analyzer:** Shows the memory regions and their details.

6. Code to blink LED2 three times at startup at a 1 second rate

The screenshot shows the STM32CubeIDE interface with the project 'Assignment-1' open. The main.c file in the center editor window contains the following code:

```
97  /* Configure the system clock */
98  SystemClock_Config();
99
100 /* USER CODE BEGIN SysInit */
101
102 /* USER CODE END SysInit */
103
104 /* Initialize all configured peripherals */
105 MX_GPIO_Init();
106 MX_DFSDM1_Init();
107 MX_I2C2_Init();
108 MX_QUADSPI_Init();
109 MX_SPI3_Init();
110 MX_USART1_UART_Init();
111 MX_USART2_UART_Init();
112 MX_USART3_UART_Init();
113 MX_USB_OTG_FS_PCD_Init();
114
115 /* USER CODE BEGIN 2 */
116 HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
117 HAL_Delay(1000);
118 HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
119 HAL_Delay(1000);
120 HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
121 HAL_Delay(1000);
122
123 /* Infinite loop */
124 /* USER CODE BEGIN WHILE */
125 while (1)
126 {
127     /* USER CODE END WHILE */
128
129     /* USER CODE BEGIN 3 */
130 }
131 /* USER CODE END 3 */
132 }
```

The code includes initializations for various peripherals and a main loop that toggles LED2 pin every second. The STM32CubeIDE interface also shows the Project Explorer, Outline, Build Targets, Problems, Tasks, Console, Properties, and Build Analyzer panes.

7. After blinking LED2 from the previous step, blinks the Wifi and BLE LED on/off at a 1 second on/off rate "forever".

The screenshot shows the STM32CubeIDE interface with the following details:

- Project Explorer:** Shows the project structure for "Embedded-C-Assigment-1". The "Src" folder contains "main.c", which is currently selected.
- Code Editor:** Displays the C code for "main.c". The code includes logic to toggle LED2 (GPIO Port, Pin), followed by an infinite loop where it toggles two other pins (LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin) every 1000ms.
- Outline View:** Shows a list of symbols and functions defined in the code, such as `main.h`, `SystemClock_Config(void)`, and various handle definitions for I2C, SPI, USART, and DFSDM.
- Build Targets:** Shows the build targets for the project.

```
115     HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
116     HAL_Delay(1000);
117     HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
118     HAL_Delay(1000);
119     HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
120     HAL_Delay(1000);
121     /* USER CODE END 2 */
122
123     /* Infinite loop */
124     /* USER CODE BEGIN WHILE */
125     while (1)
126     {
127         /* USER CODE END WHILE */
128
129         /* USER CODE BEGIN 3 */
130         HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
131         HAL_Delay(1000);
132     }
133     /* USER CODE END 3 */
134 }
135
136 /**
137 * @brief System Clock Configuration
138 * @retval None
139 */
140 void SystemClock_Config(void)
141 {
142     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
```

8. Pressing the Blue button should generate an interrupt that toggles the LED on/off

The screenshot shows the STM32CubeIDE interface with the following details:

- Project Explorer:** Shows the project structure for "Embedded-C-Assigment-1" with files like main.c, HAL.h, and various STM32L475VGTx components.
- Code Editor:** Displays the main.c file containing C code for GPIO initialization and an EXTI callback function. The code includes comments for EXTI15_10_IRQn and Error_Handler().
- Outline View:** Lists all the functions defined in the project, including the generated HAL library functions.
- Build Analyzer:** Shows the build status for "Embedded-C-Assigment-1.elf" at "Jul 15, 2023, 10:01:30 AM".
- Memory Regions:** A table showing memory usage across RAM, RAM2, and FLASH regions.

```
main.c
...
637     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
638     GPIO_InitStruct.Alternate = GPIO_AF4_I2C1;
639     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
640
641     /* EXTI interrupt init*/
642     HAL_NVIC_SetPriority(EXTI9_5_IRQn, 0, 0);
643     HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
644
645     HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
646     HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
647
648     /* USER CODE BEGIN MX_GPIO_Init_2 */
649     /* USER CODE END MX_GPIO_Init_2 */
650
651     /* USER CODE BEGIN 4 */
652     void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
653     {
654         if(GPIO_Pin == GPIO_PIN_13)
655         {
656             HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
657         }
658     }
659     /* USER CODE END 4 */
660
661     /**
662      * @brief This function is executed in case of error occurrence.
663      * @retval None
664      */
665     void Error_Handler(void)
666     {
667         /* USER CODE BEGIN Error_Handler_Debug */
668         /* User can add his own implementation to report the HAL error return state */
669         __disable_irq();
670         while (1)
671     }
```

CDT Build Console [Embedded-C-Assigment-1]
Finished building: default.size.stdout
Finished building: Embedded-C-Assigment-1.list
10:01:30 Build Finished. 0 errors, 0 warnings. (took 45.722ms)

Region	Start address	End address	Size	Free	Used
RAM	0x20000000	0x2001ffff	96 KB	92.59 KB	3.41 KB
RAM2	0x10000000	0x10007fff	32 KB	32 KB	0 B
FLASH	0x08000000	0x080fffff	1024 KB	1005.26 KB	18.74 KB

9. Build the project and flash on the board, result is successful. First shows that bluetooth button is on



10. Second, shows that WIFI button is on



11. Finally, press the user button, and we can toggle LED2

