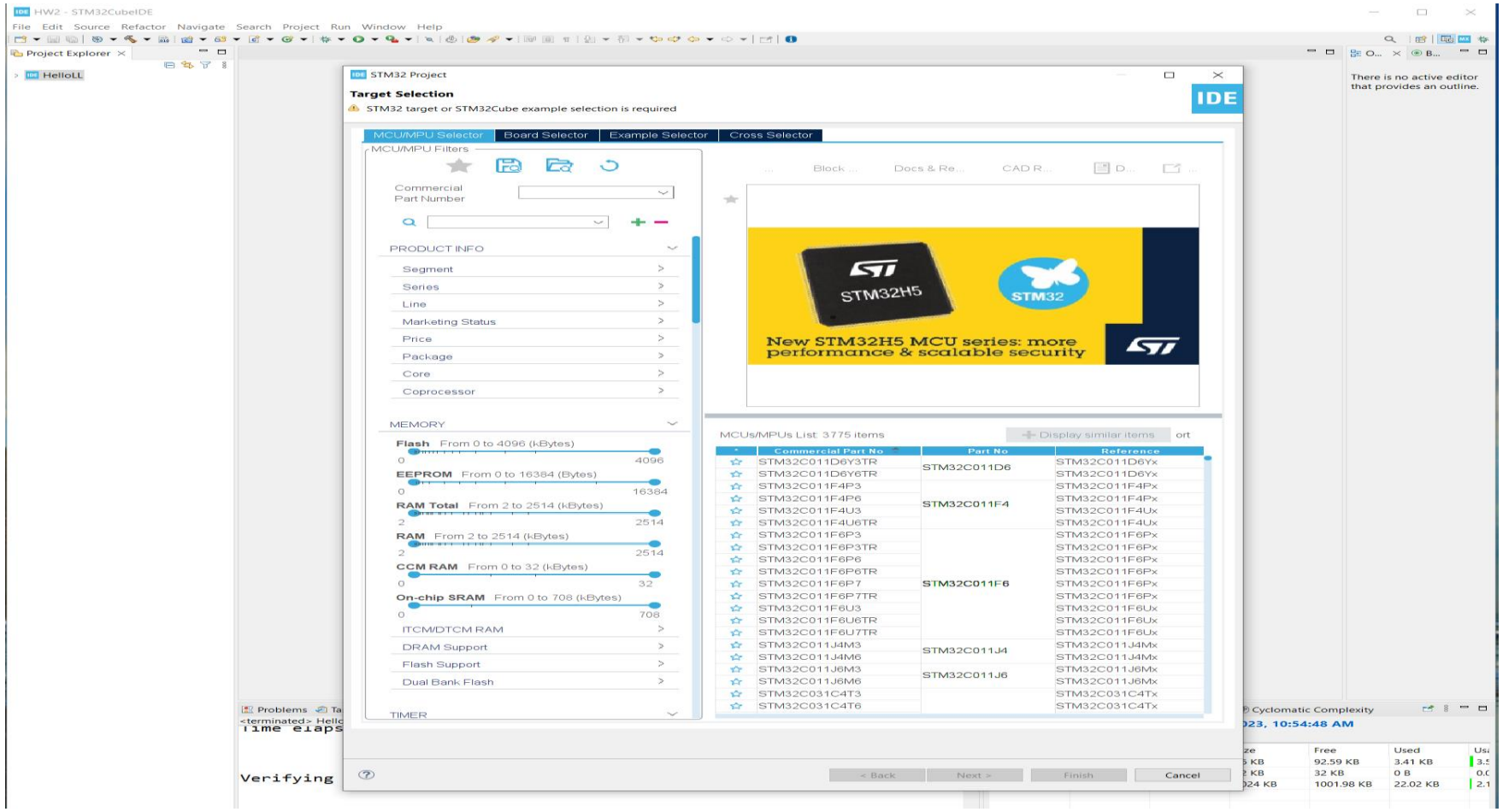
UCSD Embedded C Final Assignment

By

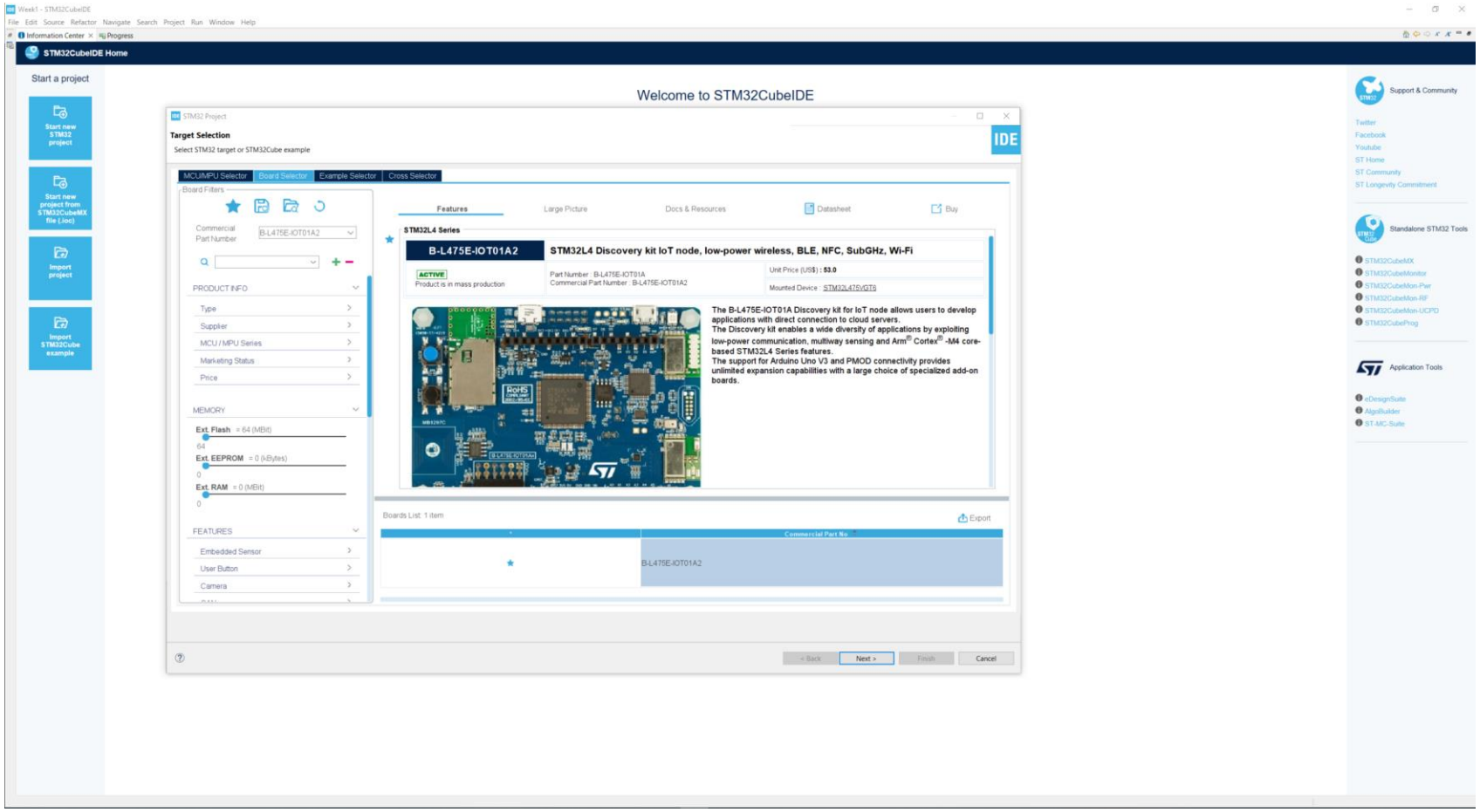
Hsuankai Chang

hsuankac@umich.edu

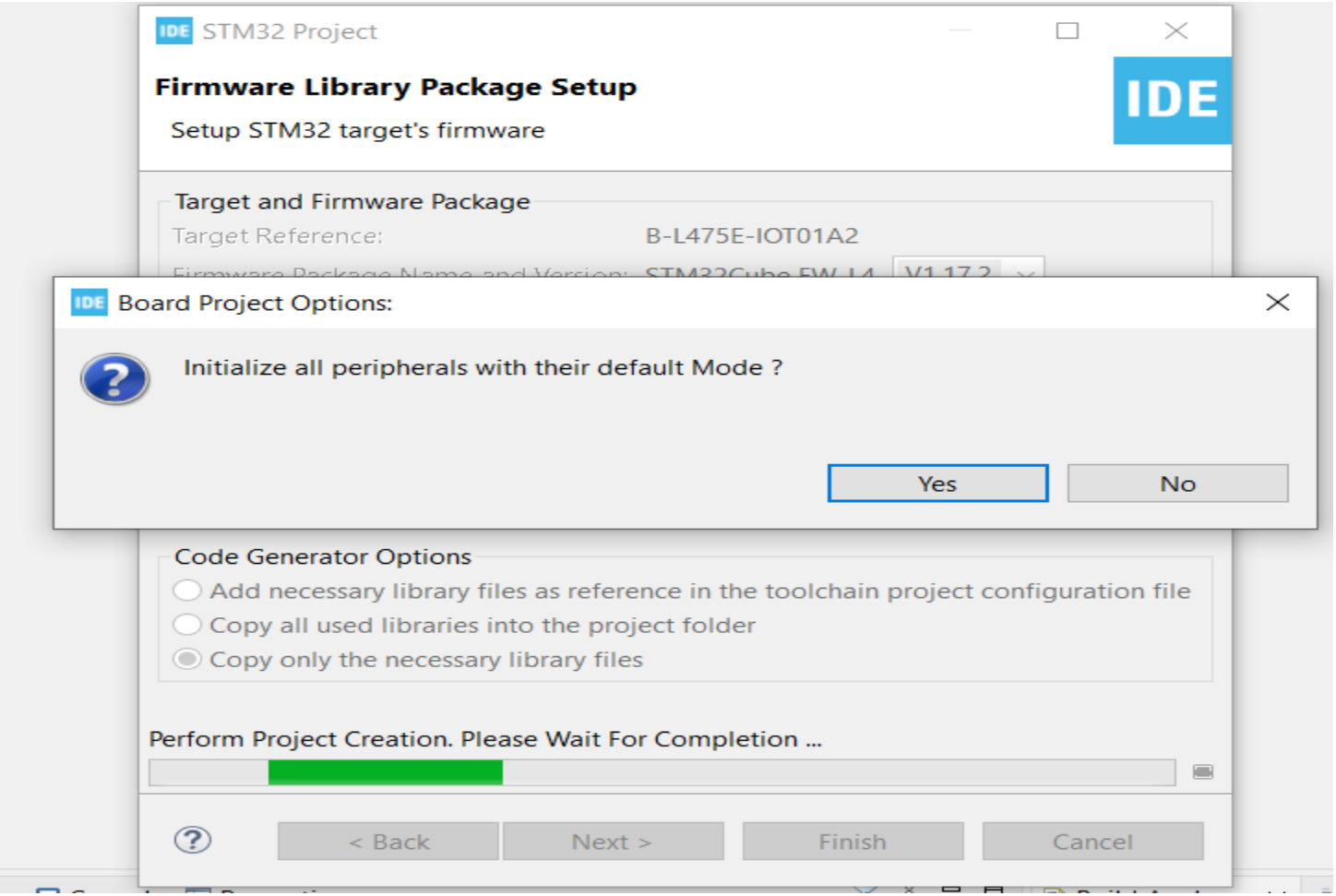
Step 1. Startup STM32CubeIDE and create new STM32 project



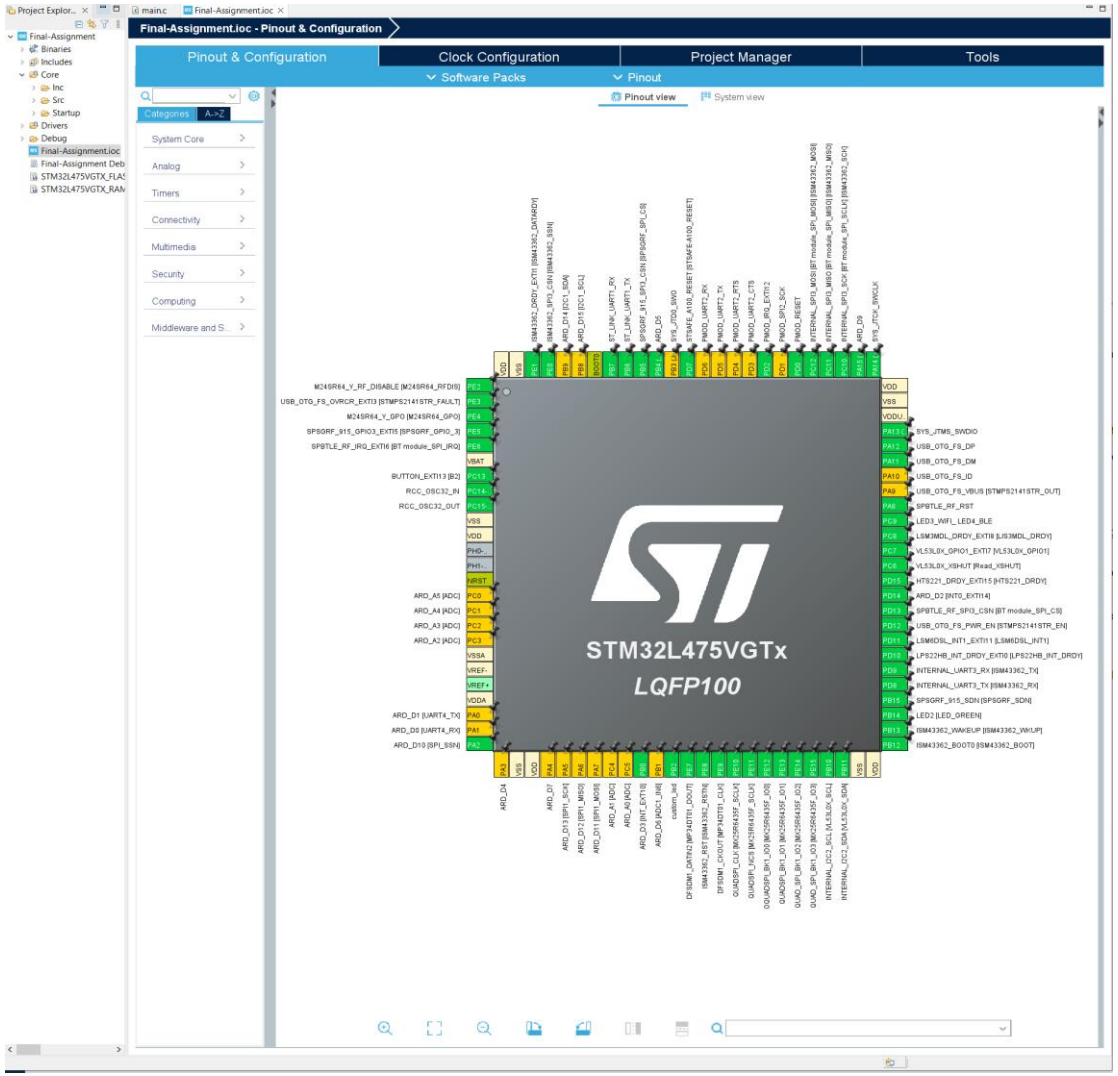
Step 2. Access board selector and type in the board you use, enter project name and click Next



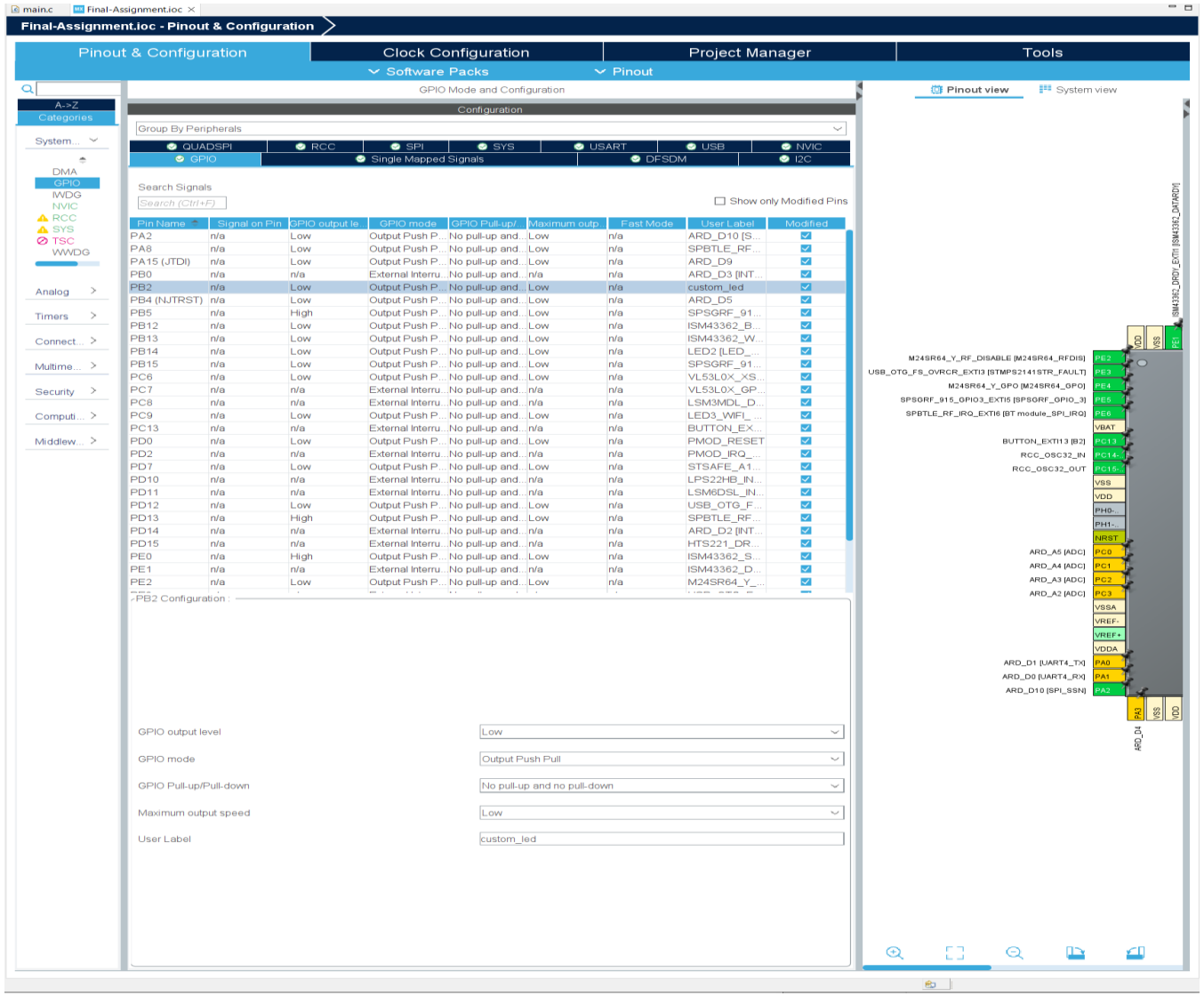
Step 3. Click yes to initialize all peripherals to default



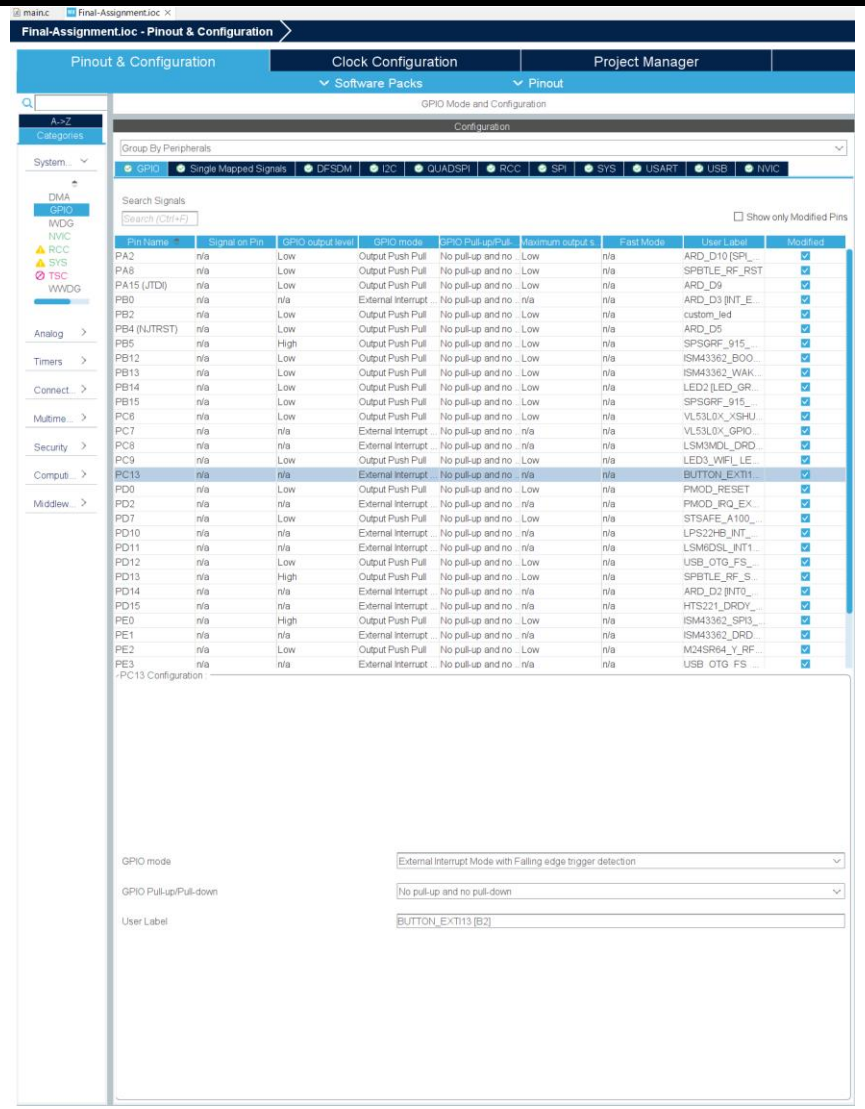
Step 4. When in .ioc file, view the overall default settings

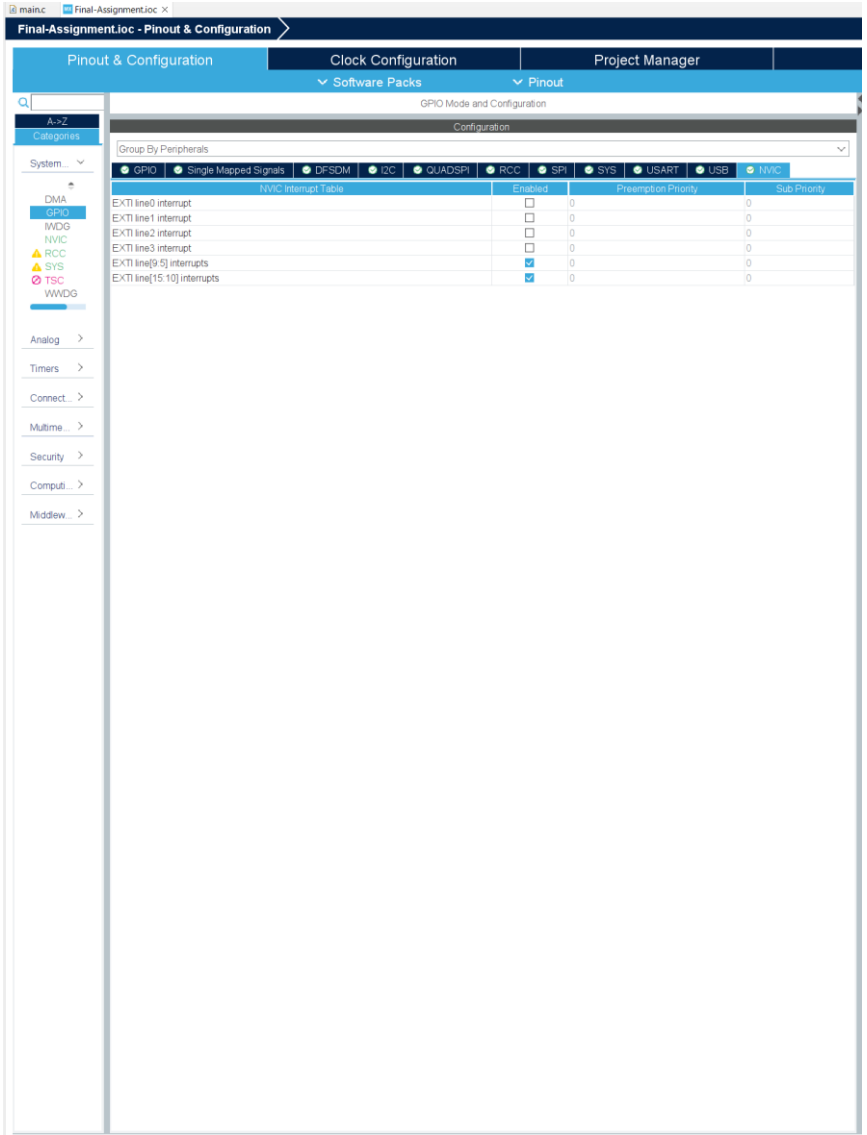


Step 5. Click GPIO and set PB2 pin as custom_led output push pull pin, we will use this pin to toggle external LED for demo 4



Step 6. Confirm that blue button has set up as external interrupt trigger



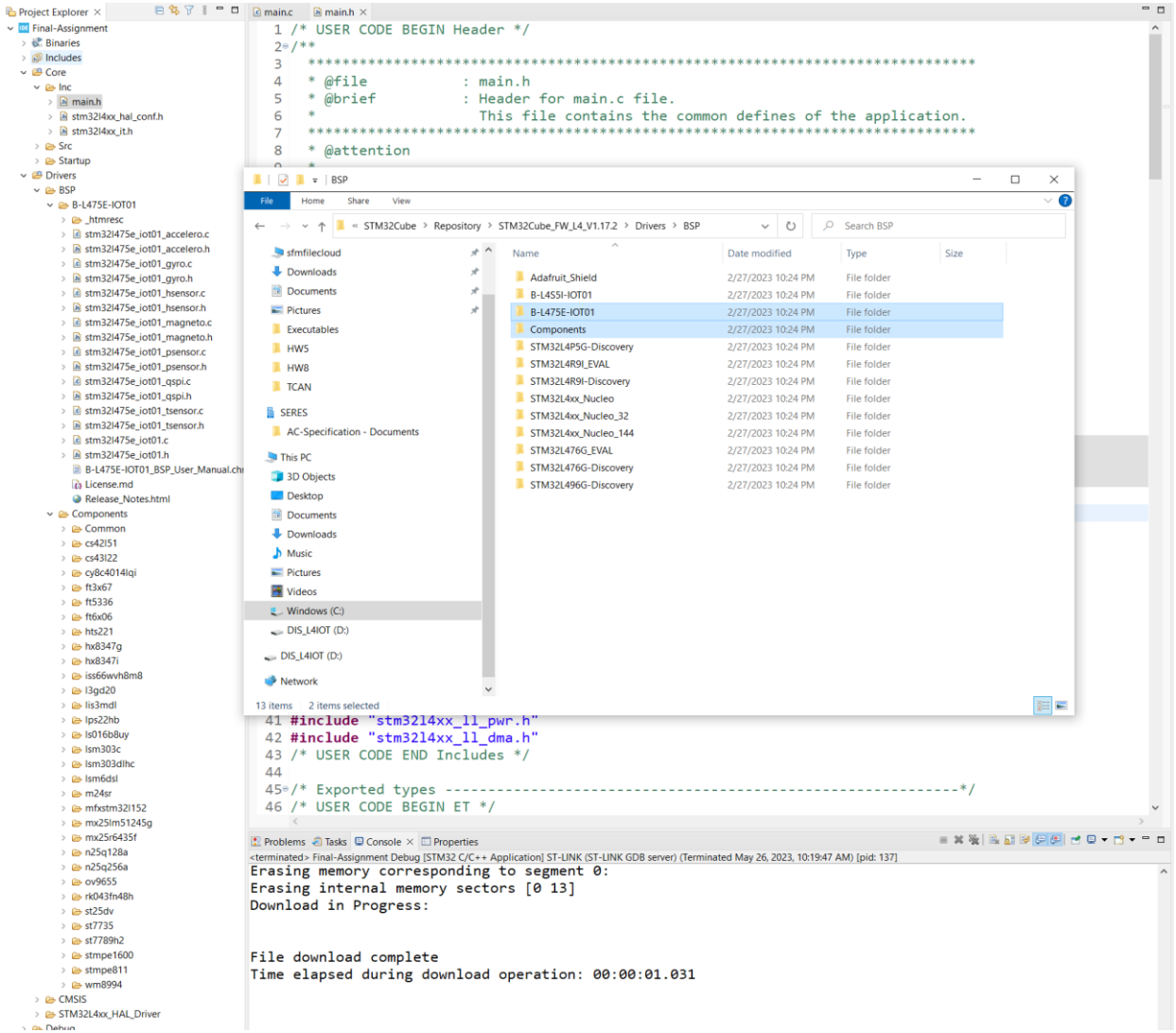


Step 7. Generate the code, and in main.h file, and the required low layer header file for use in demo 1

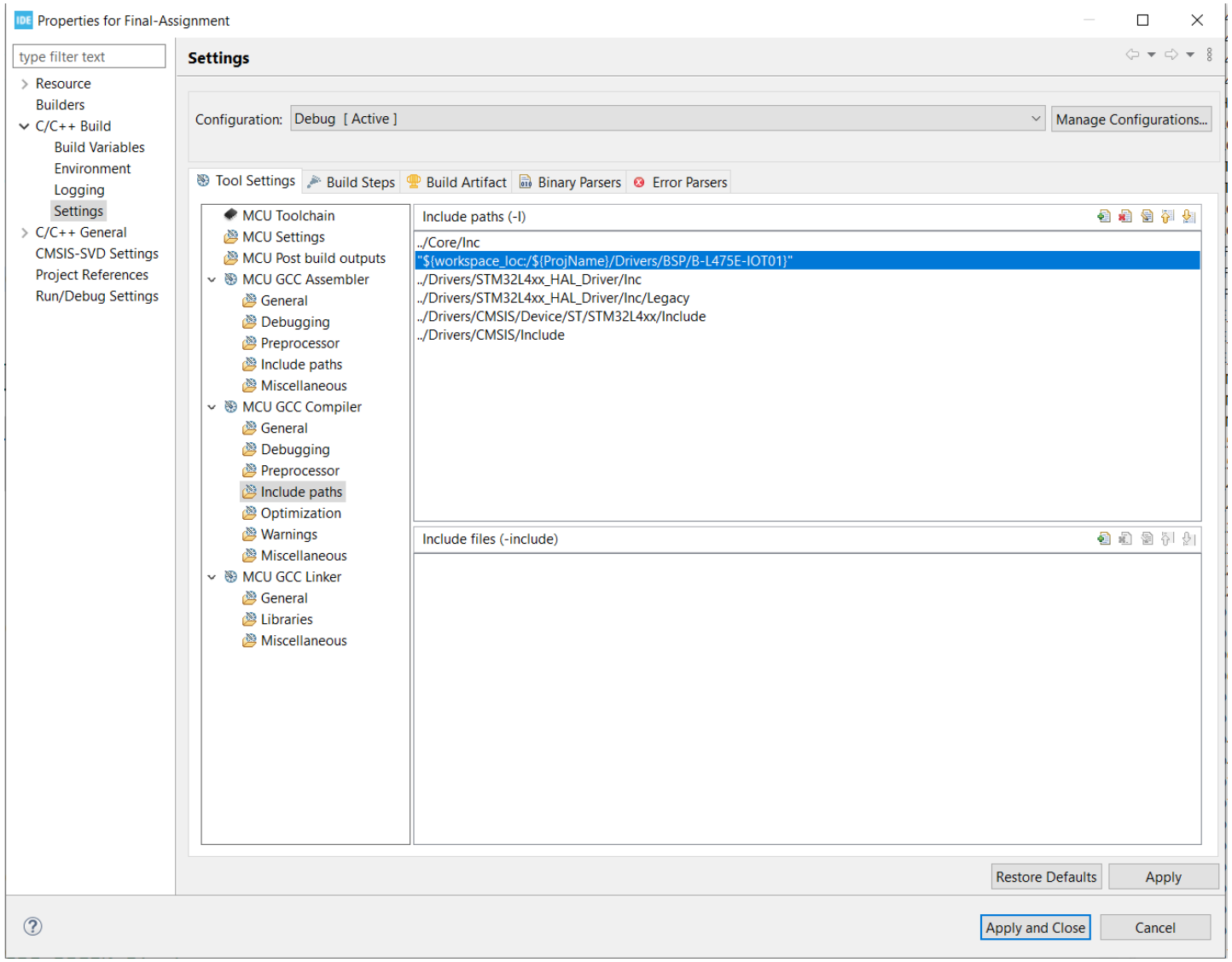
main.c main.h

```c
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.h
  * @brief          : Header for main.c file.
  *                   This file contains the common defines of the application.
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2023 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
  ******************************************************************************
  */
/* USER CODE END Header */

/* Define to prevent recursive inclusion -------------------------------------*/
#ifndef __MAIN_H
#define __MAIN_H

#ifdef __cplusplus
extern "C" {
#endif

/* Includes ------------------------------------------------------------------*/
#include "stm32l4xx_hal.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "stm32l4xx_ll_system.h"
#include "stm32l4xx_ll_gpio.h"
#include "stm32l4xx_ll_exti.h"
#include "stm32l4xx_ll_bus.h"
#include "stm32l4xx_ll_cortex.h"
#include "stm32l4xx_ll_rcc.h"
#include "stm32l4xx_ll_utils.h"
#include "stm32l4xx_ll_pwr.h"
#include "stm32l4xx_ll_dma.h"
/* USER CODE END Includes */

/* Exported types ------------------------------------------------------------*/
/* USER CODE BEGIN ET */
```

Step 8. Create BSP folder under Drivers folder, and copy and paste the local Components and B-L475E-IOT01 folder to it. This is for using BSP package for temp sensor in demo 3



Step 9. Add the BSP folder to the include path



Step 10. In main.c file, add the required header files

main.c × main.h

```c
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2023 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
  ******************************************************************************
  */
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include <string.h>
#include <stdio.h>
#include <math.h>
#include "stm32l475e_iot01.h"
#include "stm32l475e_iot01_tsensor.h"
/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
DFSDM_Channel_HandleTypeDef hdfsdm1_channel1;

I2C_HandleTypeDef hi2c2;

QSPI_HandleTypeDef hqspi;

SPI_HandleTypeDef hspi3;

UART_HandleTypeDef huart1;
```

Step 11. In main.c, add the needed variables

main.c × main.h

```c
/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
DFSDM_Channel_HandleTypeDef hdfsdm1_channel1;

I2C_HandleTypeDef hi2c2;

QSPI_HandleTypeDef hqspi;

SPI_HandleTypeDef hspi3;

UART_HandleTypeDef huart1;
UART_HandleTypeDef huart3;

PCD_HandleTypeDef hpcd_USB_OTG_FS;

/* USER CODE BEGIN PV */
uint8_t demo0_count = 0;
uint8_t demo1_count = 0;
uint8_t demo_count = 0;
char buf[100];
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DFSDM1_Init(void);
static void MX_I2C2_Init(void);
static void MX_QUADSPI_Init(void);
static void MX_SPI3_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_USART3_UART_Init(void);
static void MX_USB_OTG_FS_PCD_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
```

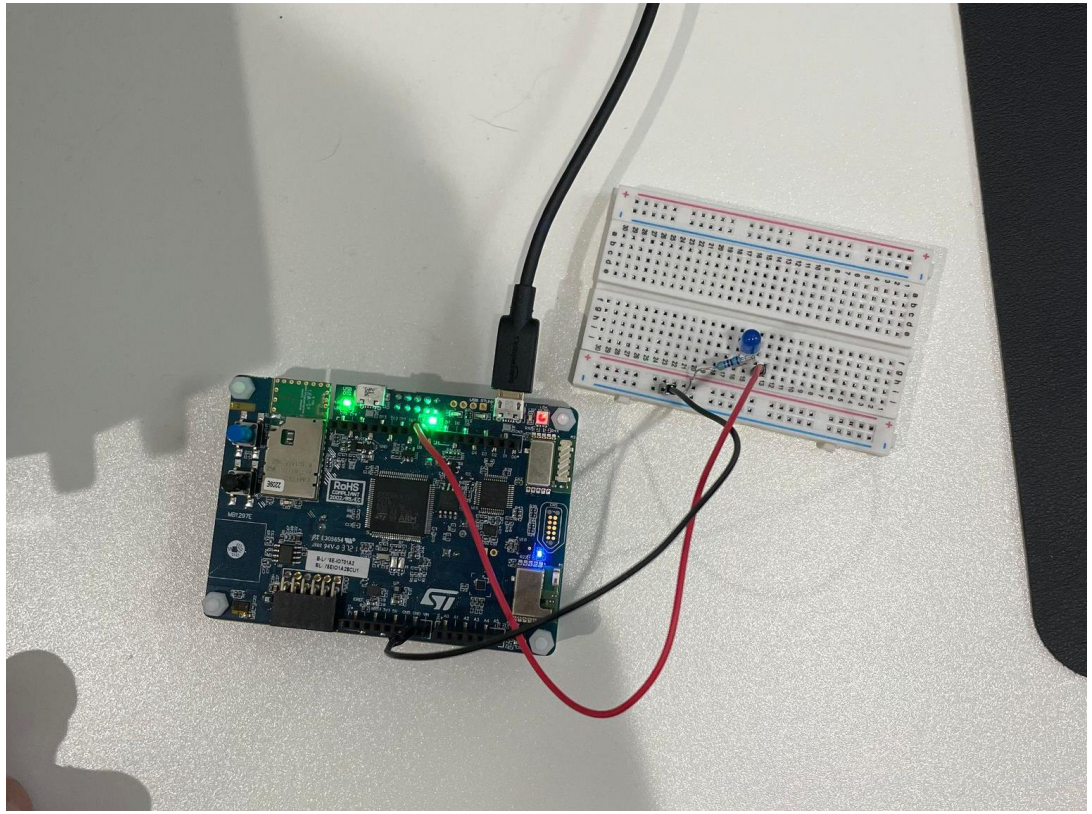
Step 12. Add the function definition for EXTI call back function

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == BUTTON_EXTI13_Pin)
    {
        demo_count = (demo_count + 1) % 4;
        if(demo_count == 0){
            demo0_count = 0;
        }else if (demo_count == 1){
            demo1_count = 0;
        }
        sprintf(buf, "UART1: current demo is: %d\r\n", demo_count + 1);
        HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
    }
}
/* USER CODE END 4 */

/**
```

Step 13. Add the code for demo 1 to demo 4

```c
while (1)
{
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
  switch(demo_count)
  {
      case 0:
      {
          if(demo0_count == 0)
          {
              uint32_t flashSize = LL_GetFlashSize();
              uint32_t deviceID0 = LL_GetUID_Word0();
              uint32_t deviceID1 = LL_GetUID_Word1();
              uint32_t deviceID2 = LL_GetUID_Word2();
              sprintf(buf, "LL flash size is; %lu\r\n", flashSize);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              sprintf(buf, "LL device ID 0 is; %lu\r\n", deviceID0);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              sprintf(buf, "LL device ID 1 is; %lu\r\n", deviceID1);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              sprintf(buf, "LL device ID 2 is; %lu\r\n", deviceID2);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              demo0_count++;
          }
          LL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
          HAL_Delay(1000);
          break;
      }
      case 1:
      {
          if(demo1_count == 0)
          {
              uint32_t DeviceID = HAL_GetDEVID();
              uint32_t deviceUID0 = HAL_GetUIDw0();
              uint32_t deviceUID1 = HAL_GetUIDw1();
              uint32_t deviceUID2 = HAL_GetUIDw2();

              sprintf(buf, "HAL device ID is; %lu\r\n", DeviceID);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              sprintf(buf, "HAL unique device ID 0 is; %lu\r\n", deviceUID0);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              sprintf(buf, "HAL unique device ID 1 is; %lu\r\n", deviceUID1);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              sprintf(buf, "HAL unique device ID 2 is; %lu\r\n", deviceUID2);
              HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
              demo1_count++;
          }
          HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
          HAL_Delay(2000);
          break;
      }
      case 2:
      {
          float temp = BSP_TSENSOR_ReadTemp();
          int tempInt1 = temp;
          float tempFrac = temp - tempInt1;
          int tempInt2 = trunc(tempFrac * 100);
          sprintf(buf, "Temperature is : %d.%02d\r\n", tempInt1, tempInt2);
          HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
          BSP_LED_Toggle(LED2);
          HAL_Delay(3000);
          break;
      }
      case 3:
      {
          HAL_GPIO_TogglePin(custom_led_GPIO_Port, custom_led_Pin);
          HAL_Delay(4000);
          break;
      }
      default:
          break;
  }
}
/* USER CODE END 3 */
}
```

Step 14. Setup the wire connection for demo 4 LED blinking



Step 15. Compile and run the code in debug mode, open tera term and test from demo 1. test is successful, print the flash size and device ID once when entering demo 1 code, and continue blinking LED2 at 1 second interval

main.h | main.c | system_stm32l4xx.c | startup_stm32l475vgtx.s

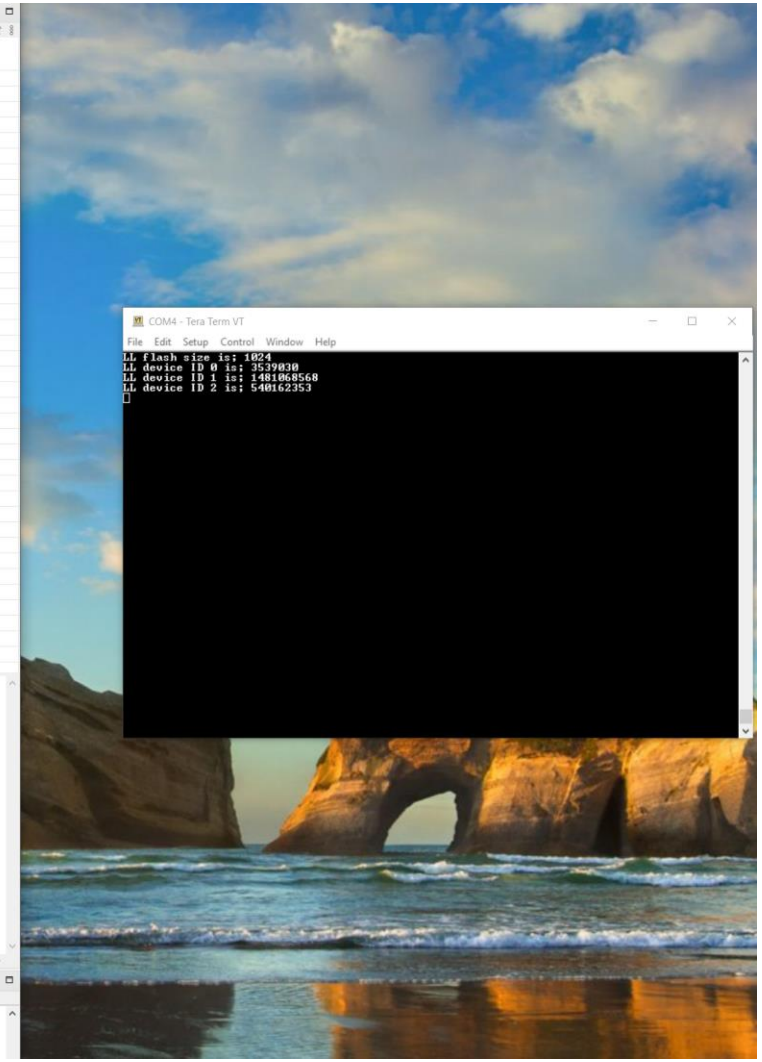
```c
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  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();
  /* USER CODE BEGIN 2 */
  BSP_TSENSOR_Init();
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    switch(demo_count)
    {
```

Expression	Type	Value
demo_count	uint8_t	0 '\0'
Add new expres		

Console | Problems | Executables | Debugger Console | Memory

Final-Assignment Debug [STM32 C/C++ Application] [pid: 154]

```
File download complete
Time elapsed during download operation: 00:00:01.034
```



Step 16. Press the user button, we will see HAL device ID and unique ID only print once, and continue blinking LED for 2 seconds interval. Demo 2 success

main.h | main.c | system_stm32l4xx.c | startup_stm32l475vgtx.s

```c
        }
        case 1:
        {
            if(demo1_count == 0)
            {
                uint32_t DeviceID = HAL_GetDEVID();
                uint32_t deviceUID0 = HAL_GetUIDw0();
                uint32_t deviceUID1 = HAL_GetUIDw1();
                uint32_t deviceUID2 = HAL_GetUIDw2();

                sprintf(buf, "HAL device ID is; %lu\r\n", DeviceID);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 0 is; %lu\r\n", deviceUID0);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 1 is; %lu\r\n", deviceUID1);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 2 is; %lu\r\n", deviceUID2);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                demo1_count++;
            }
            HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
            HAL_Delay(2000);
            break;
        }
        case 2:
        {
            float temp = BSP_TSENSOR_ReadTemp();
            int tempInt1 = temp;
            float tempFrac = temp - tempInt1;
            int tempInt2 = trunc(tempFrac * 100);
            sprintf(buf, "Temperature is : %d.%02d\r\n", tempInt1, tempInt2);
            HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
            BSP_LED_Toggle(LED2);
            HAL_Delay(3000);
            break;
        }
        case 3:
        {
            HAL_GPIO_TogglePin(custom_led_GPIO_Port, custom_led_Pin);
            HAL_Delay(4000);
            break;
        }
        default:
            break;
    }
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
```

Expression	Type	Value
demo_count	uint8_t	0 '\0'
Add new expres		

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
LL flash size is; 1024
LL device ID 0 is; 3539030
LL device ID 1 is; 1481068568
LL device ID 2 is; 540162353
UART1: current demo is: 2
HAL device ID is; 1045
HAL unique device ID 0 is; 3539030
HAL unique device ID 1 is; 1481068568
HAL unique device ID 2 is; 540162353
```

Step 17. Press the user button again, now we can read the temp value and send it through UART1. Demo 3 success

main.h | main.c | system_stm32l4xx.c | startup_stm32l475vgtx.s

```c
        }
        case 1:
        {
            if(demo1_count == 0)
            {
                uint32_t DeviceID = HAL_GetDEVID();
                uint32_t deviceUID0 = HAL_GetUIDw0();
                uint32_t deviceUID1 = HAL_GetUIDw1();
                uint32_t deviceUID2 = HAL_GetUIDw2();

                sprintf(buf, "HAL device ID is; %lu\r\n", DeviceID);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 0 is; %lu\r\n", deviceUID0);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 1 is; %lu\r\n", deviceUID1);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 2 is; %lu\r\n", deviceUID2);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                demo1_count++;
            }
            HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
            HAL_Delay(2000);
            break;
        }
        case 2:
        {
            float temp = BSP_TSENSOR_ReadTemp();
            int tempInt1 = temp;
            float tempFrac = temp - tempInt1;
            int tempInt2 = trunc(tempFrac * 100);
            sprintf(buf, "Temperature is : %d.%02d\r\n", tempInt1, tempInt2);
            HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
            BSP_LED_Toggle(LED2);
            HAL_Delay(3000);
            break;
        }
        case 3:
        {
            HAL_GPIO_TogglePin(custom_led_GPIO_Port, custom_led_Pin);
            HAL_Delay(4000);
            break;
        }
        default:
            break;
    }
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
```

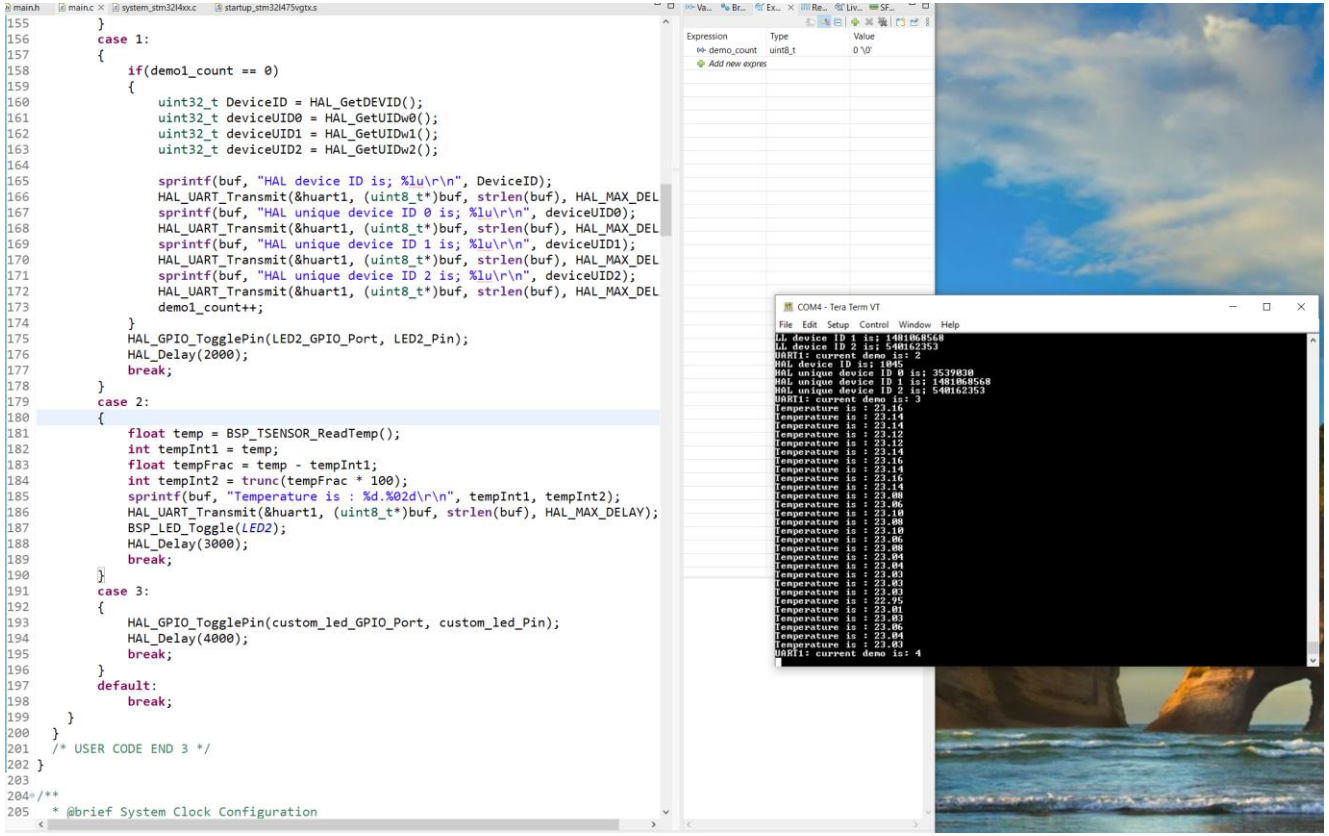
Expression	Type	Value
demo_count	uint8_t	0 '\0'
Add new expres		

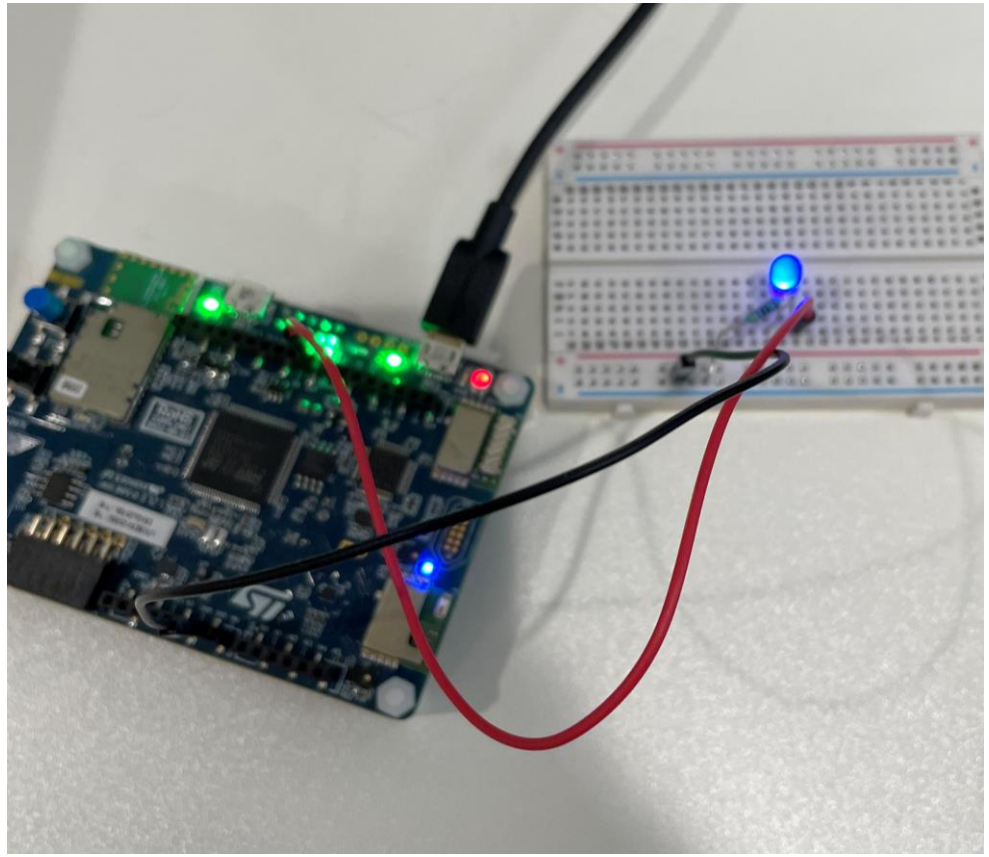
COM4 - Tera Term VT

File Edit Setup Control Window Help

```
LL flash size is; 1024
LL device ID 0 is; 3539030
LL device ID 1 is; 1481068568
LL device ID 2 is; 540162353
UART1: current demo is: 2
HAL device ID is; 1045
HAL unique device ID 0 is; 3539030
HAL unique device ID 1 is; 1481068568
HAL unique device ID 2 is; 540162353
UART1: current demo is: 3
Temperature is : 23.16
Temperature is : 23.14
```

Step 18. Press the user button again, we can see the external LED blinking at 4 seconds interval. Demo 4 success





Step 19. Press the user button again, it will loop back to demo 1

main.h | main.c | system_stm32l4xx.c | startup_stm32l475vgtx.s

```c
        }
        case 1:
        {
            if(demo1_count == 0)
            {
                uint32_t DeviceID = HAL_GetDEVID();
                uint32_t deviceUID0 = HAL_GetUIDw0();
                uint32_t deviceUID1 = HAL_GetUIDw1();
                uint32_t deviceUID2 = HAL_GetUIDw2();

                sprintf(buf, "HAL device ID is; %lu\r\n", DeviceID);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 0 is; %lu\r\n", deviceUID0);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 1 is; %lu\r\n", deviceUID1);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                sprintf(buf, "HAL unique device ID 2 is; %lu\r\n", deviceUID2);
                HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DEL
                demo1_count++;
            }
            HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
            HAL_Delay(2000);
            break;
        }
        case 2:
        {
            float temp = BSP_TSENSOR_ReadTemp();
            int tempInt1 = temp;
            float tempFrac = temp - tempInt1;
            int tempInt2 = trunc(tempFrac * 100);
            sprintf(buf, "Temperature is : %d.%02d\r\n", tempInt1, tempInt2);
            HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);
            BSP_LED_Toggle(LED2);
            HAL_Delay(3000);
            break;
        }
        case 3:
        {
            HAL_GPIO_TogglePin(custom_led_GPIO_Port, custom_led_Pin);
            HAL_Delay(4000);
            break;
        }
        default:
            break;
    }
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
```

Expression	Type	Value
demo_count	uint8_t	0 '\0'
Add new expres		

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
HAL unique device ID 1 is; 1481068568
HAL unique device ID 2 is; 540162353
UART1: current demo is: 3
Temperature is : 23.16
Temperature is : 23.14
Temperature is : 23.14
Temperature is : 23.12
Temperature is : 23.12
Temperature is : 23.14
Temperature is : 23.16
Temperature is : 23.14
Temperature is : 23.16
Temperature is : 23.14
Temperature is : 23.08
Temperature is : 23.06
Temperature is : 23.10
Temperature is : 23.08
Temperature is : 23.10
Temperature is : 23.06
Temperature is : 23.08
Temperature is : 23.04
Temperature is : 23.04
Temperature is : 23.03
Temperature is : 23.03
Temperature is : 23.03
Temperature is : 22.95
Temperature is : 23.01
Temperature is : 23.03
Temperature is : 23.06
Temperature is : 23.04
Temperature is : 23.03
UART1: current demo is: 4
UART1: current demo is: 1
LL flash size is; 1024
LL device ID 0 is; 3539030
LL device ID 1 is; 1481068568
LL device ID 2 is; 540162353
```