Design a Datalogger code to send HTU21DF sensor data from STM32 to NanoEdge AI Studio on PC

Objective:

The Objective of this experiment is to create a datalogger code for HTU21DF sensor. The datalogger code will create a buffer where all the temperature & humidity sensor data sample will be stored, using which we will be able to create datasets of temperature & humidity samples to build a machine learning model in the NanoEdge AI Studio.
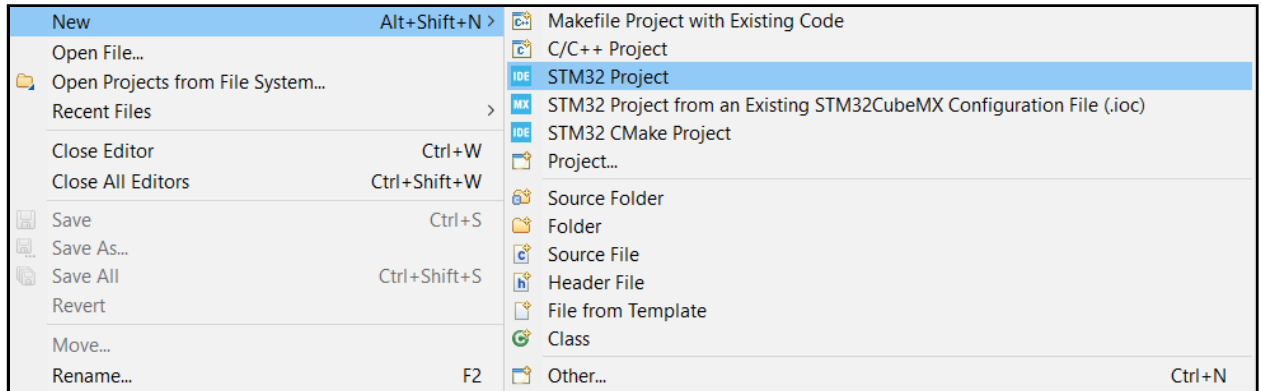
Requirements:

1. STM32 Cube IDE software.
2. HTU21DF Sensor (I2C).
3. STM32 Microcontroller.
4. USB Cable for the microcontroller.
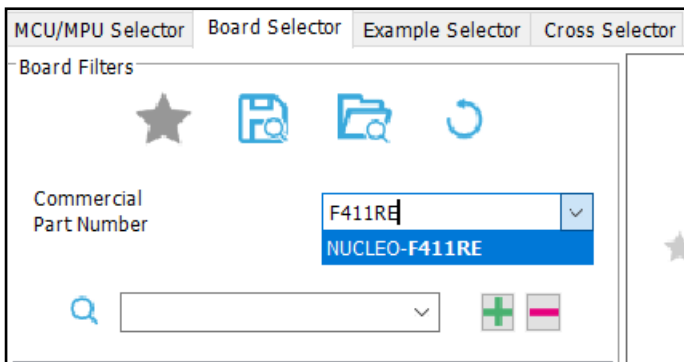5. Jumper Wires.

Connection Diagram:
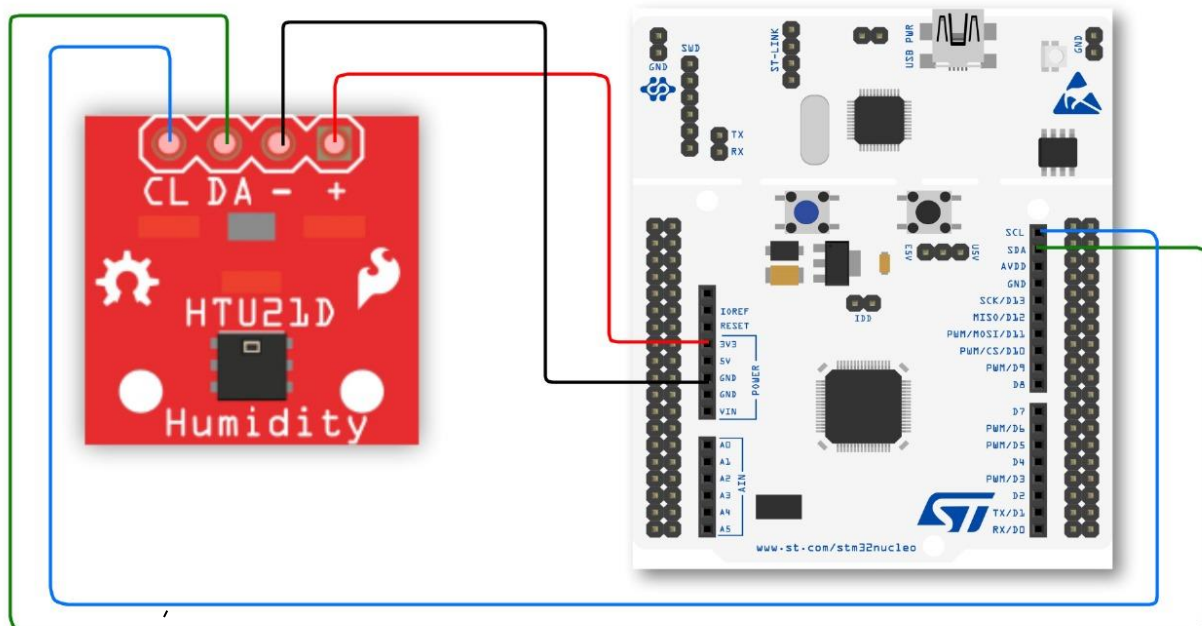
Procedure:

1. Click on **File→New→STM32 Project** to start your project on Cube IDE.
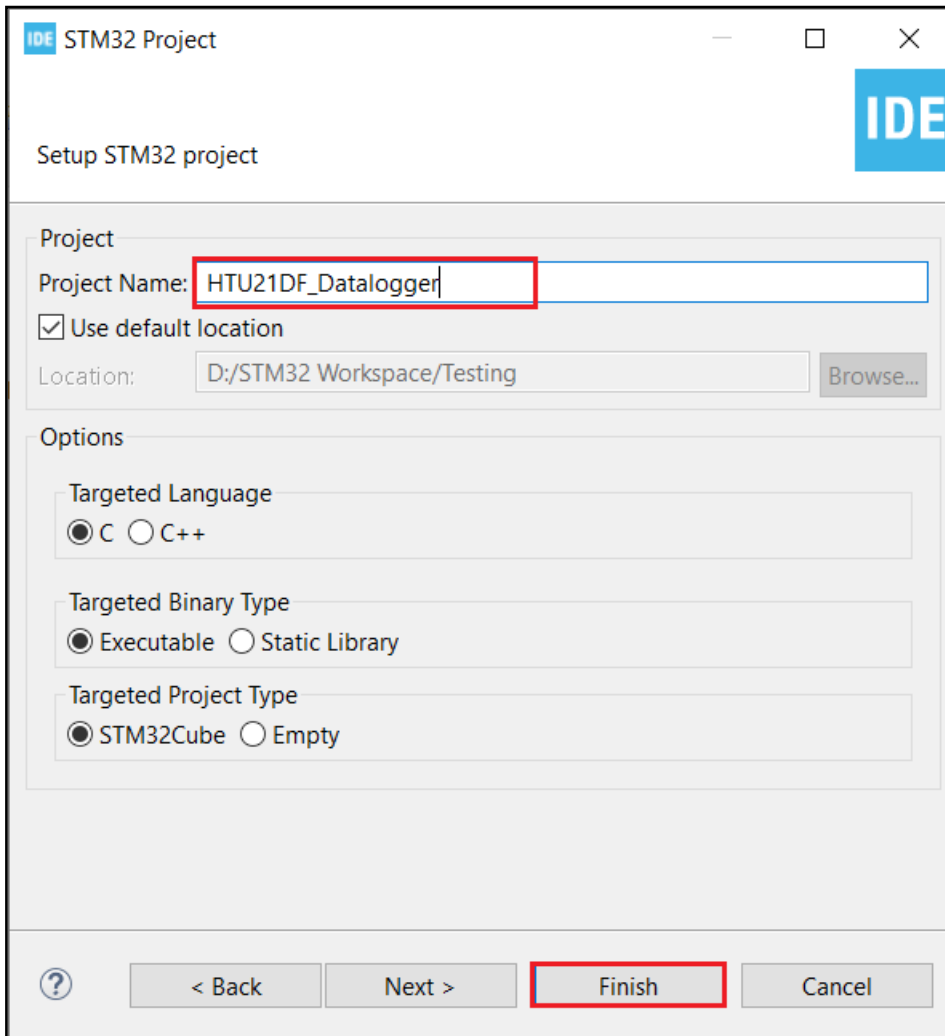
2. A **Target Selection** window will open. Click on **Board Selector**, where you need to select the microcontroller board you are working with.
   (**NB:** If you are having Nucleo-F401RE, you have to select the said Commercial Part Number)

3. After this on the right-hand side of the window, under **Board List** you will see the board you have selected. Click on the board and then click on **Next.**

4. Give your project a name, rest of the things will remain by default as it is for now. Click on **Finish.**

5. Cube IDE will ask if you want to initialize all peripherals with their default mode, click on **Yes.**
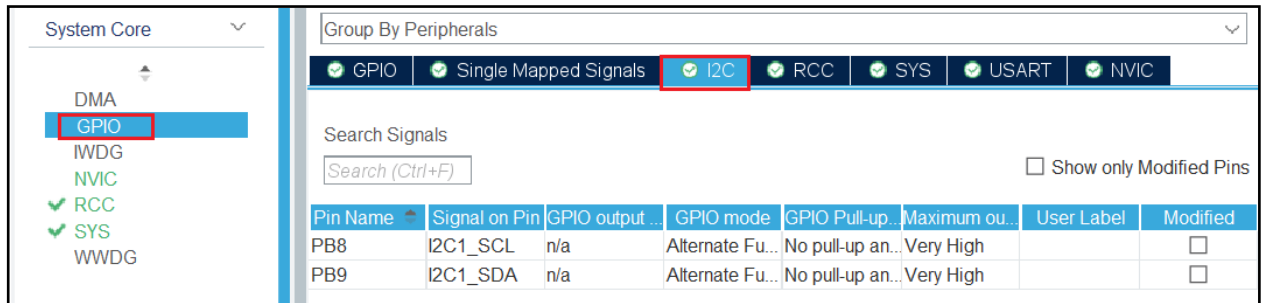
6. In the **Pinout & Configuration** tab, click on **PB8** pin and select it as an **I2C1_SCL** and **PB9** pin as an **I2C1_SDA**.
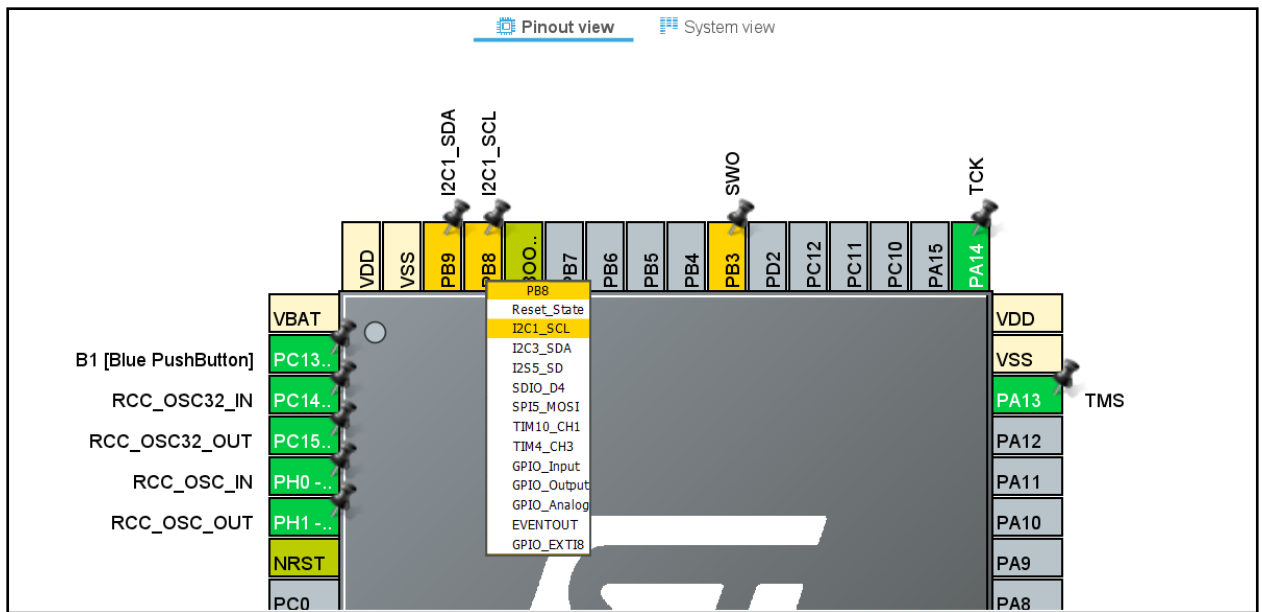
7. Next on the left-hand side under **Categories** →**Connectivity**, select I2C1 and enable it.
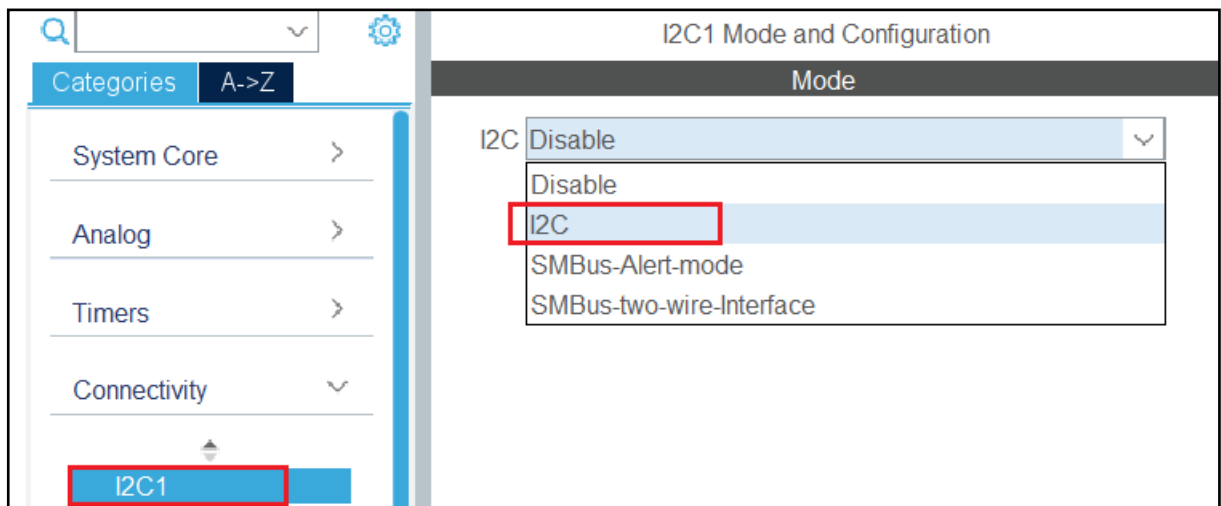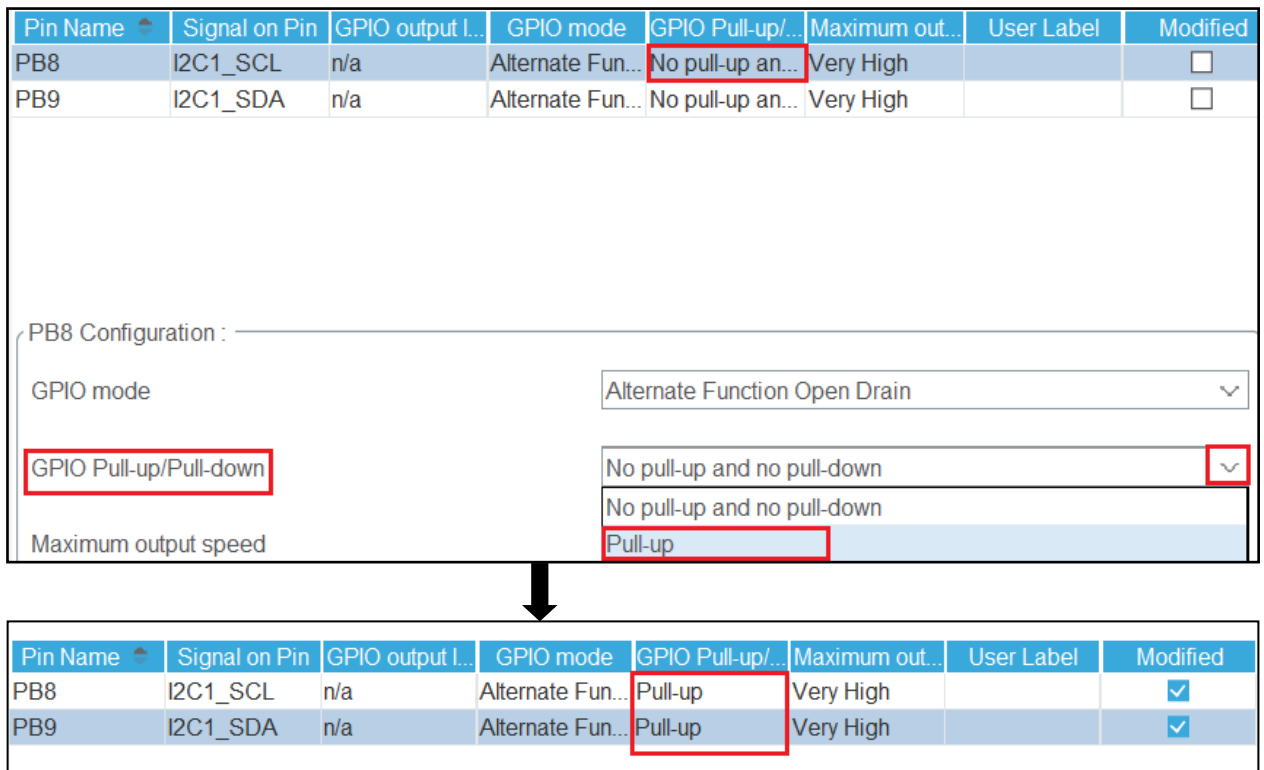
8. Now go to **System Core** → **GPIO** and select **I2C** option.

9. After selecting **I2C** option, change the GPIO pin **PB8 and PB9** as **Pull-up.**

10. Press **Ctrl+S** to generate your code. On the left-hand side of the Cube IDE, under **Project Explorer** go to the project you have created (For example, I have named my project as (HTU21DF_Datalogger) **HTU21DF_Datalogger→Core →Src→main.c (**double click to load the code).
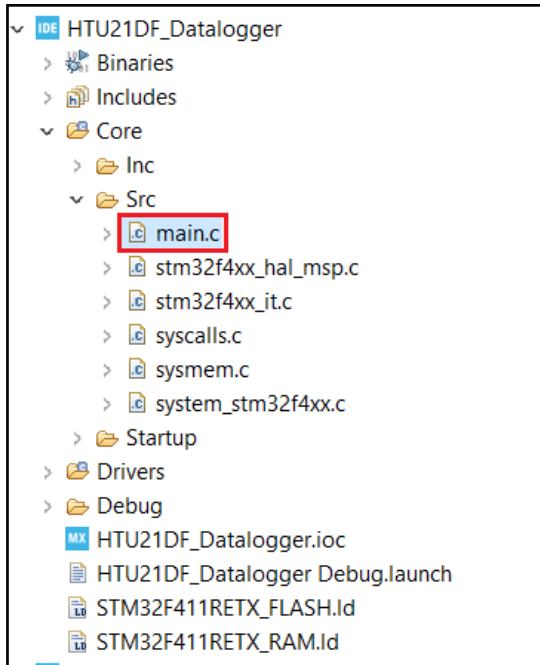
11. Open your project tree **HTU21DF_Datalogger.** Right click on your **Drivers** folder and create a new **Folder** name as BSP.

Folder name: BSP

Advanced >>

(?) Finish   Cancel
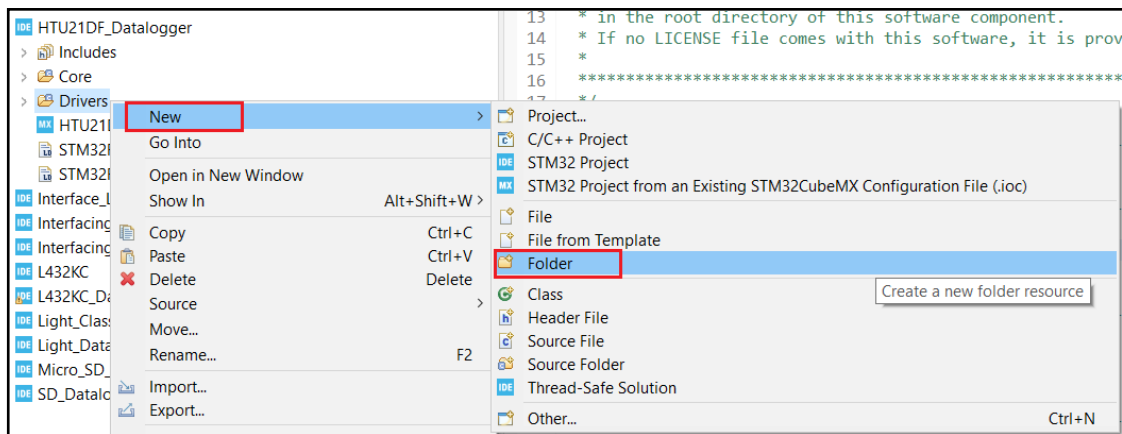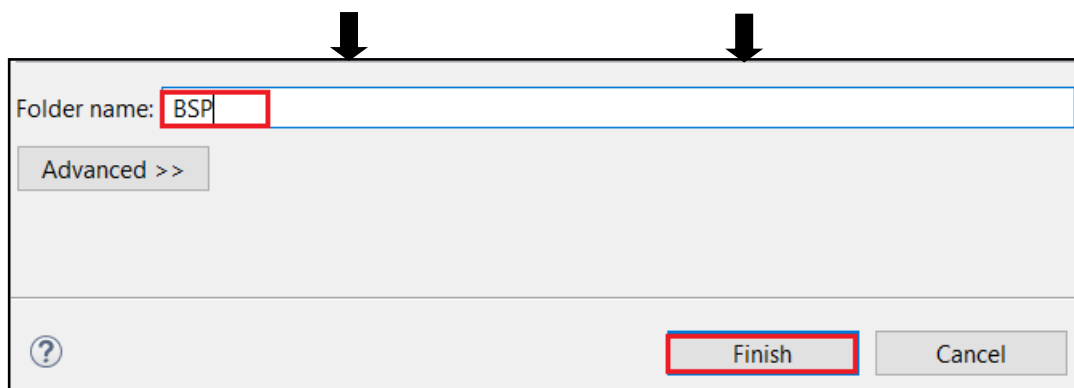
12. Now right click on the **BSP** folder and create a new folder name as **HTU21.**

IDE HTU21DF_Datalogger
> ⚙ Binaries
> 📖 Includes
> 📂 Core
∨ 📂 Drivers
  > 📂 BSP
  > 📂 CM   New   >   📄 Project...
  > 📂 STM   Go Into   ⓒ C/C++ Project
> 📂 Debug   IDE STM32 Project
  MX HTU21 Open in New Window   MX STM32 Project from an Existing STM32CubeMX Configuration File (.ioc)
  📄 HTU21   Show In   Alt+Shift+W >
  📄 STM32   📄 File
  📄 STM32   Copy   Ctrl+C   📄 File from Template
           Paste   Ctrl+V   📁 Folder
IDE Interface_ Delete   Delete
IDE Interfacin Source   > ⓖ Class
IDE Interfacin Move...   h Header File
IDE L432KC   Rename...   F2   ⓒ Source File
IDE L432KC_D   📁 Source Folder
IDE Light_Clas Import...   IDE Thread-Safe Solution
IDE Light_Dat Export...   📄 Other...   Ctrl+N

45   3   110g   2048 LSB/g 7
46
47   Ax = accel_data[0]/2048.0;
48   Ay = accel_data[1]/2048.0;
49   Az = accel_data[2]/2048.0;
50 }
51

Folder name: HTU21

Advanced >>

(?) Finish   Cancel

13. 13. Right click on the HTU21 folder and create a new Source file and the name the file as **HTU21.h** and select on **Finish.**
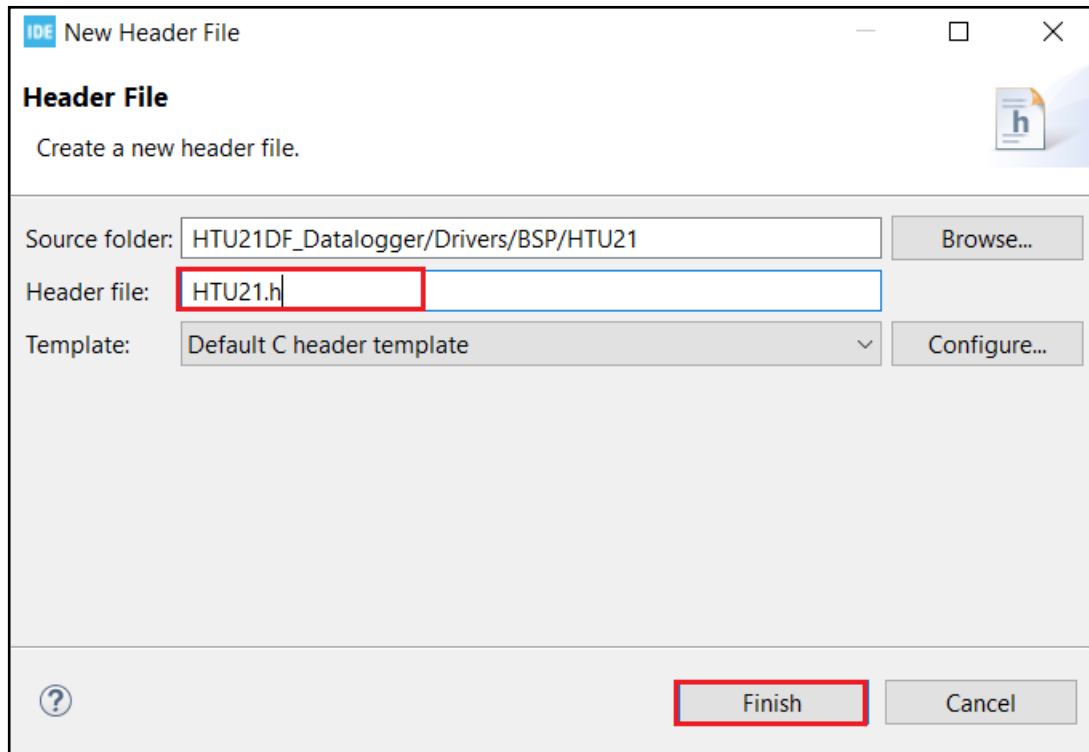
14. Below is the code snippets, please put your code in the appropriate places in the **HTU21.h** file.

```c
2
3  #ifndef SRC_HTU21_H_
4  #define SRC_HTU21_H_
5
6  #include "stm32f4xx_hal.h"
7  extern I2C_HandleTypeDef hi2c1;
8
9  /** Default I2C address for the HTU21D. */
10 #define HTU21_I2CADDR          (0x80)
11
12 /** Read temperature register. */
13 #define HTU21_READTEMP         (0xE3)
14
15 /** Read humidity register. */
16 #define HTU21_READHUM          (0xE5)
17
18 /** Write register command. */
19 #define HTU21_WRITEREG         (0xE6)
20
21 /** Read register command. */
22 #define HTU21_READREG          (0xE7)
23
24 /** Reset command. */
25 #define HTU21_RESET            (0xFE)
26
27 #ifdef __cplusplus
28 extern "C" {
29 #endif
30 char    HTU21_Init(void);
31 float   HTU21_GetTemp(void);
32 float   HTU21_GetHum(void);
33 void    HTU21_Reset(void);
34 #ifdef __cplusplus
35 }
36 #endif
37
38 #endif /* SRC_HTU21_H_ */
39
```

15. Right click on the HTU21 folder again and create a new source file and the name the file as **HTU21.c** and select on **Finish.**
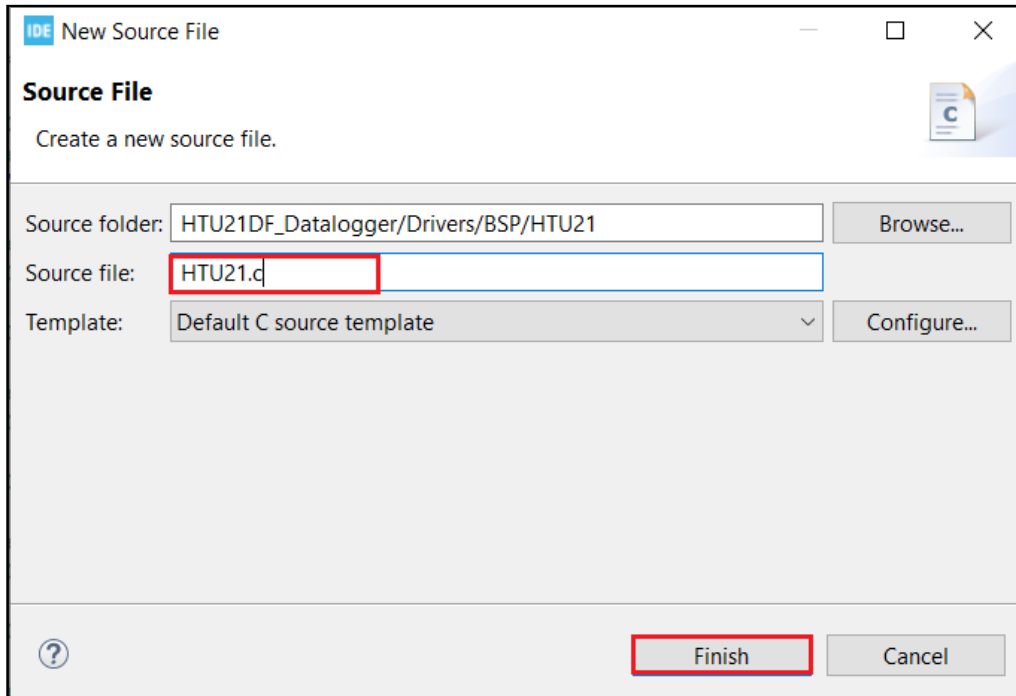
16. Below is the code snippets, please put your code in the appropriate places in the **HTU21.c** file.

```
2  #include "HTU21.h"
3
4  void HTU21_Reset()
5  {
6      uint8_t Data=HTU21_RESET;
7
8      HAL_Delay(15);
9      HAL_I2C_Master_Transmit(&hi2c1, HTU21_I2CADDR, &Data, 1, 1000);
10     HAL_Delay(15);
11
12     }
13
14 char HTU21_Init()
15 {
16     uint8_t check;
17     uint8_t Data=HTU21_READREG;
18
19     HAL_I2C_Master_Transmit(&hi2c1, HTU21_I2CADDR, &Data, 1, 1000);
20     HAL_I2C_Master_Receive(&hi2c1, HTU21_I2CADDR, &check, 1, 1000);
21
22     return check;
23     }
24
```

```c
float HTU21_GetTemp()
{
    uint8_t Temp_Data[2];
    uint8_t Data=HTU21_READTEMP;

    HAL_I2C_Master_Transmit(&hi2c1, HTU21_I2CADDR, &Data, 1, 1000);
    HAL_I2C_Master_Receive(&hi2c1, HTU21_I2CADDR, Temp_Data, 2, 1000);

    uint16_t Raw_Temp = (uint16_t)(Temp_Data[0] << 8 | (Temp_Data[1]& 0b11111100));

    float temp = Raw_Temp;
    temp *= 175.72f;
    temp /= 65536.0f;
    temp -= 46.85f;

    return temp;
    }

float    HTU21_GetHum()
{
    uint8_t Hum_Data[2];
    uint8_t Data=HTU21_READHUM;

    HAL_I2C_Master_Transmit(&hi2c1, HTU21_I2CADDR, &Data, 1, 1000);
    HAL_I2C_Master_Receive(&hi2c1, HTU21_I2CADDR, Hum_Data, 2, 1000);

    uint16_t Raw_Hum = (uint16_t)(Hum_Data[0] << 8 | (Hum_Data[1]& 0b11110000));

    float temp = Raw_Hum;
    temp *= 175.72f;
    temp /= 65536.0f;
    temp -= 46.85f;

    return temp;
    }
```

17. Cube IDE automatically generates a code format based on the configurations you have done. Cube IDE uses HAL libraries. Below is the code snippets, please put your code in the appropriate places in the **main.c** file.

```
23  /* USER CODE BEGIN Includes */
24  #include "stdio.h"
25  #include "string.h"
26  #include "HTU21.h"
27  /* USER CODE END Includes */
28
29  /* Private typedef -------------------------------------------------------*/
30  /* USER CODE BEGIN PTD */
31
32  /* USER CODE END PTD */
33
34  /* Private define --------------------------------------------------------*/
35  /* USER CODE BEGIN PD */
36
37  /* USER CODE END PD */
38
39  /* Private macro ---------------------------------------------------------*/
40  /* USER CODE BEGIN PM */
41  float my_temperature, my_humidity;
42  #define DATA_INPUT_USER 256
43  #define AXIS_NUMBER 2
44  /* USER CODE END PM */
45
46  /* Private variables -----------------------------------------------------*/
47  I2C_HandleTypeDef hi2c1;
48
49  UART_HandleTypeDef huart2;
50
51  /* USER CODE BEGIN PV */
52  float htu21_buffer[DATA_INPUT_USER * AXIS_NUMBER] =  {0};
53  /* USER CODE END PV */
61  /* USER CODE BEGIN PFP */
62  void fill_htu21_buffer();
63  /* USER CODE END PFP */
101      /* USER CODE BEGIN 2 */
102        HAL_Delay(1000);
103        HTU21_Reset();
104        HTU21_Init();
105      /* USER CODE END 2 */
106
107      /* Infinite loop */
108      /* USER CODE BEGIN WHILE */
109      while (1)
110      {
111          fill_htu21_buffer();
112        /* USER CODE END WHILE */
```

```
273  /* USER CODE BEGIN 4 */
274
275  void fill_htu21_buffer() {
276      for (int i = 0; i < DATA_INPUT_USER; i++) {
277          my_temperature = HTU21_GetTemp();
278          my_humidity = HTU21_GetHum();
279          htu21_buffer[AXIS_NUMBER * i] = my_temperature;
280          htu21_buffer[AXIS_NUMBER * i + 1] = my_humidity;
281          HAL_Delay(3);
282          printf("%.2f", htu21_buffer[AXIS_NUMBER * i]);
283          printf(" ");
284          printf("%.2f", htu21_buffer[AXIS_NUMBER * i + 1]);
285          printf(" ");
286      }
287      printf("\r\n");
288  }
289
290  int __io_putchar(int ch) {
291      HAL_UART_Transmit(&huart2, (uint8_t*) &ch, 1, HAL_MAX_DELAY);
292      return ch;
293  }
294
295  /* USER CODE END 4 */
```

18. Now right click on your project tree, go to **Properties**.
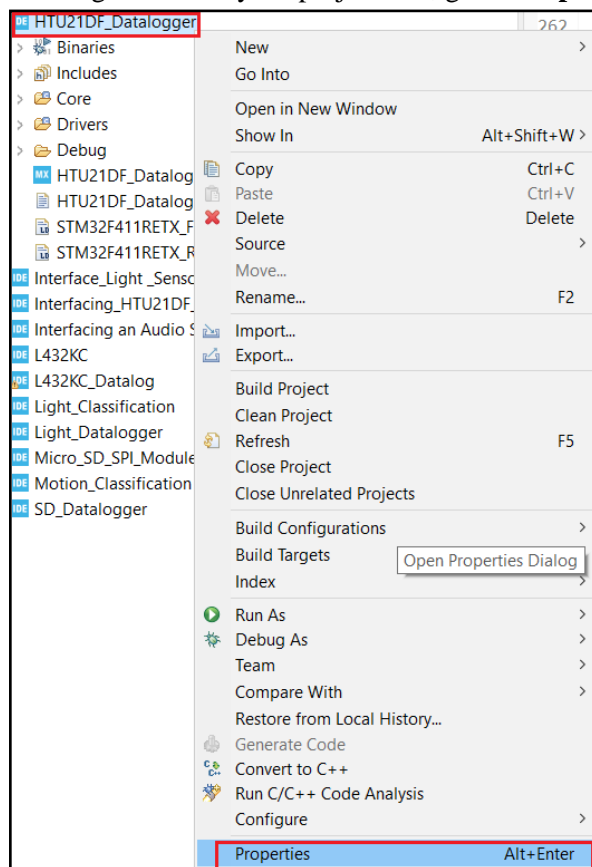
19. After that select **Settings→MCU Settings** and enable the **float printf.**
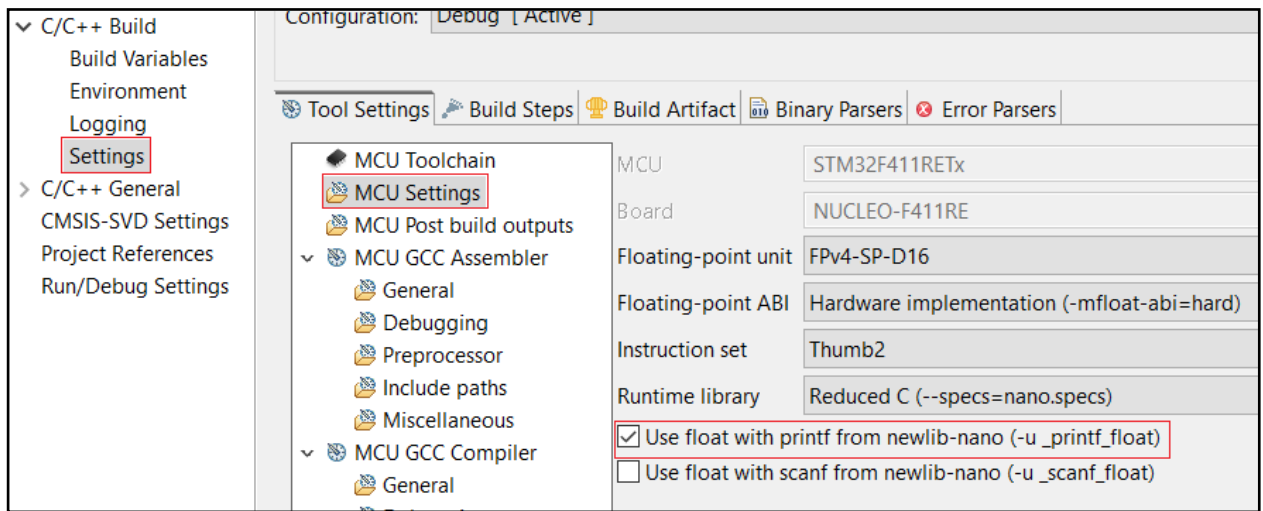
20. Now click on the build symbol on the top left corner on your Cube IDE. If you have done everything correctly your code should be built without any errors.

```
make -j4 all
arm-none-eabi-size   Interfacing_HTU21DF_Sensor.elf
   text    data    bss    dec    hex filename
  27612     476   2060  30148   75c4 Interfacing_HTU21DF_Sensor.elf
Finished building: default.size.stdout


14:33:44 Build Finished. 0 errors, 0 warnings. (took 736ms)
```

21. Next connect your STM32 board with your audio sensor connect to it to your PC and click on the
    Debug icon to start the Debugging process. An **Edit Configuration** window will open, click on **OK,** without making any changes.

22. In the debug mode, go to the bottom right hand side corner, click on open console. Select the **Connection Type** as **Serial Port**, then click on **New.** In the new window, in **Connection name** give some name to your new connection, and select the **Serial port** correctly. Then click on **Finish** and then **Ok.** A console with the given name will be opened at the bottom of your screen.

23. Click on the **Resume icon** to run your code. You should be able to see the value of light sensor in the **Console**.

```
25.78 48.82 25.77 48.82 25.78 48.77 25.78 48.77 25.78 48.77 25.79 48.73 25.79 48.73 25.80 48.73 25.79 48.73 25.79 48.69 25.79 48.69 25.79 48
.69 25.80 48.65 25.79 48.65 25.80 48.60 25.80 48.60 25.79 48.60 25.80 48.60 25.80 48.56 25.80 48.56 25.80 48.56 25.81 48.56 25.80 48.56 25.8
2 48.52 25.81 48.52 25.81 48.52 25.81 48.52 25.81 48.47 25.81 48.47 25.81 48.47 25.81 48.47 25.81 48.43 25.83 48.43 25.81 48.43 25.81 48.43
25.82 48.43 25.83 48.39 25.82 48.39 25.81 48.39 25.82 48.39 25.81 48.35 25.81 48.35 25.82 48.35 25.82 48.35 25.83 48.30 25.83 48.35 25.83 48
.30 25.82 48.30 25.83 48.30 25.83 48.30 25.83 48.30 25.82 48.30 25.82 48.26 25.82 48.26 25.82 48.30 25.84 48.30 25.83 48.30 25.83 48.26 25.8
2 48.26 25.82 48.26 25.82 48.26 25.83 48.26 25.84 48.26 25.83 48.26 10.80 48.26 25.83 48.26 25.82 48.26 25.83 48.26 25.82 48.26 25.83 48.22
25.82 48.22 25.82 48.22 25.82 48.22 25.82 48.22 25.81 48.22 25.84 48.17 25.83 48.17 25.82 48.17 25.83 48.17 25.83 48.17 25.84 48.13 25.83 48
.13 25.83 48.13 25.83 48.13 25.83 48.13 25.83 48.09 25.83 48.09 25.83 48.09 25.83 48.09 25.83 25.82 48.09 48.05 25.83 48.05 25.84 48.05 25.8
3 48.05 25.84 48.05 25.86 48.00 25.83 48.00 25.84 48.05 25.84 48.05 25.84 48.05 25.84 48.05 25.84 48.05 25.84 48.05 25.84 48.05 25.83 48.05
25.84 48.09 25.86 48.09 25.84 48.09 25.84 48.09 25.84 48.09 25.84 48.09 25.84 48.09 25.86 48.09 25.86 48.09 25.86 48.09 25.87 48.09 25.86 48
.09 25.86 48.09 25.86 48.09 25.84 48.09 25.86 48.05 25.86 48.05 25.86 48.05 25.87 48.05 25.87 48.05 25.87 48.00 25.87 48.00 25.87 48.00 25.8
5 48.00 25.87 47.96 25.86 47.96 25.87 47.96 25.86 47.96 25.87 47.92 25.87 47.96 25.88 47.92 25.87 47.92 25.87 47.92 25.84 47.92 25.87 47.87
25.86 47.87 25.86 47.87 25.87 47.87 25.87 47.87 25.86 47.83 25.87 47.83 25.87 47.83 25.87 47.83 25.87 47.83 25.87 47.83 25.87 47.83 25.87 47
.83 25.87 47.83 25.87 47.79 25.88 47.79 25.84 47.79 25.86 47.79 25.87 47.79 25.86 47.79 25.86 47.79 25.86 47.79 25.87 47.79 25.87 47.79 25.8
```

24. Before moving out of the debugging mode, click on **Disconnect** and close the console then click on the **Terminate** icon. You will be moved out of the debugging mode.

Note: All important steps and parts are highlighted with a red colour box for the proper understanding of the user. This document is for the use of education purpose only.