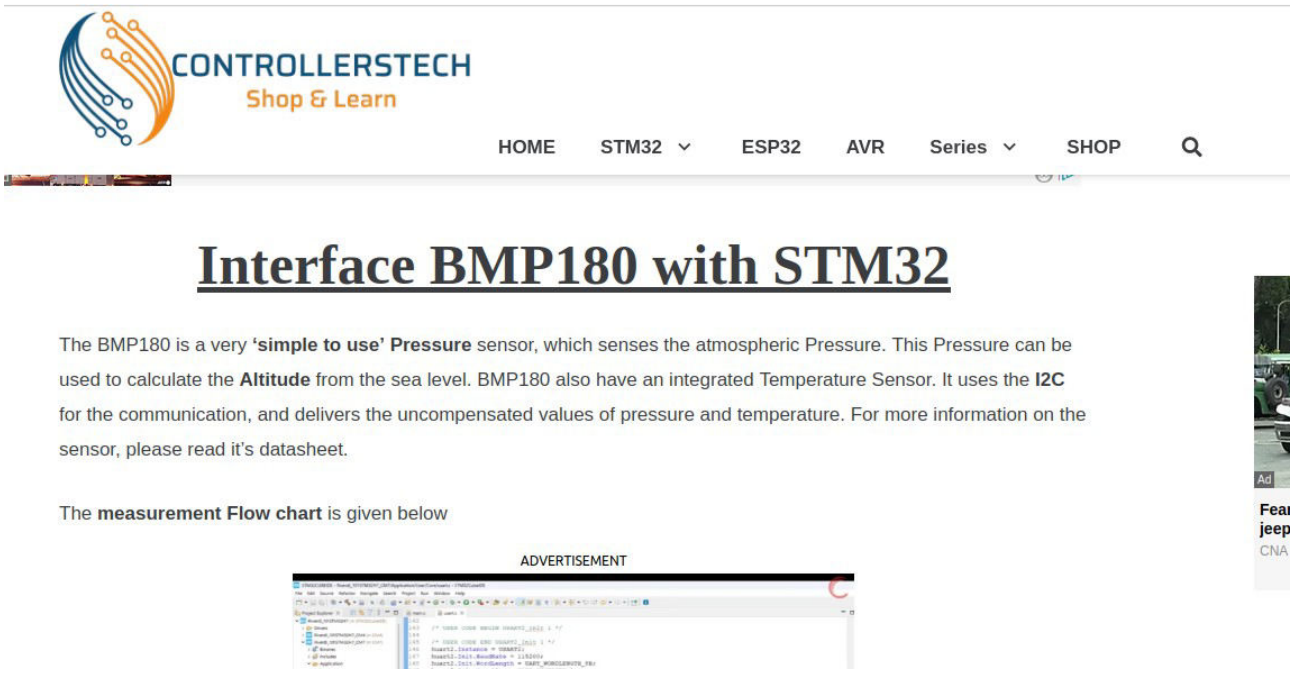
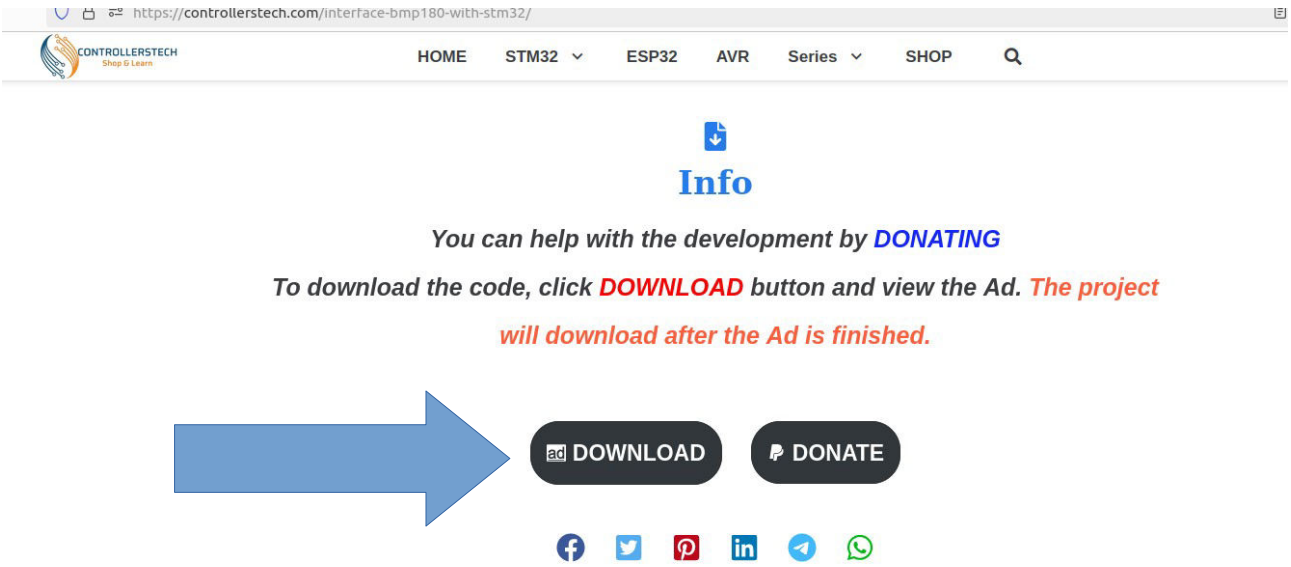
Interfacing with bmp180 sensor with stm32.Essential step has been given below

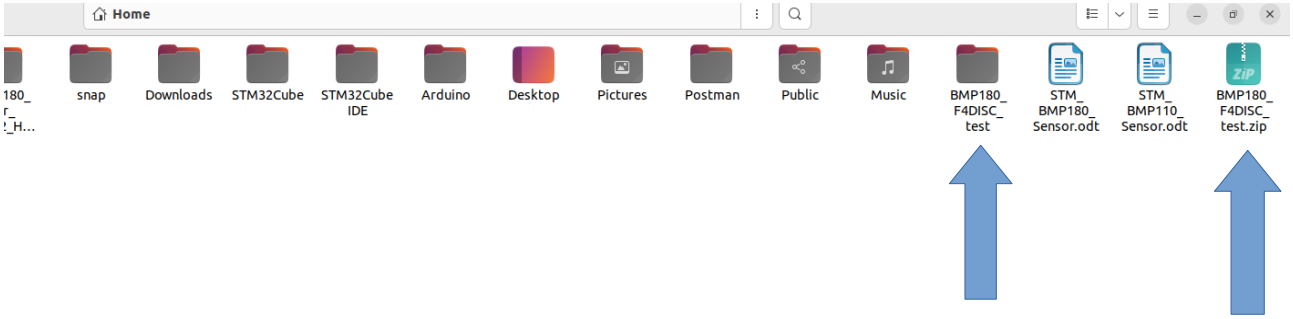
Step 1: We have to download the BMP180 header file as well as bmp.c so first visit this site https://controllerstech.com/interface-bmp180-with-stm32



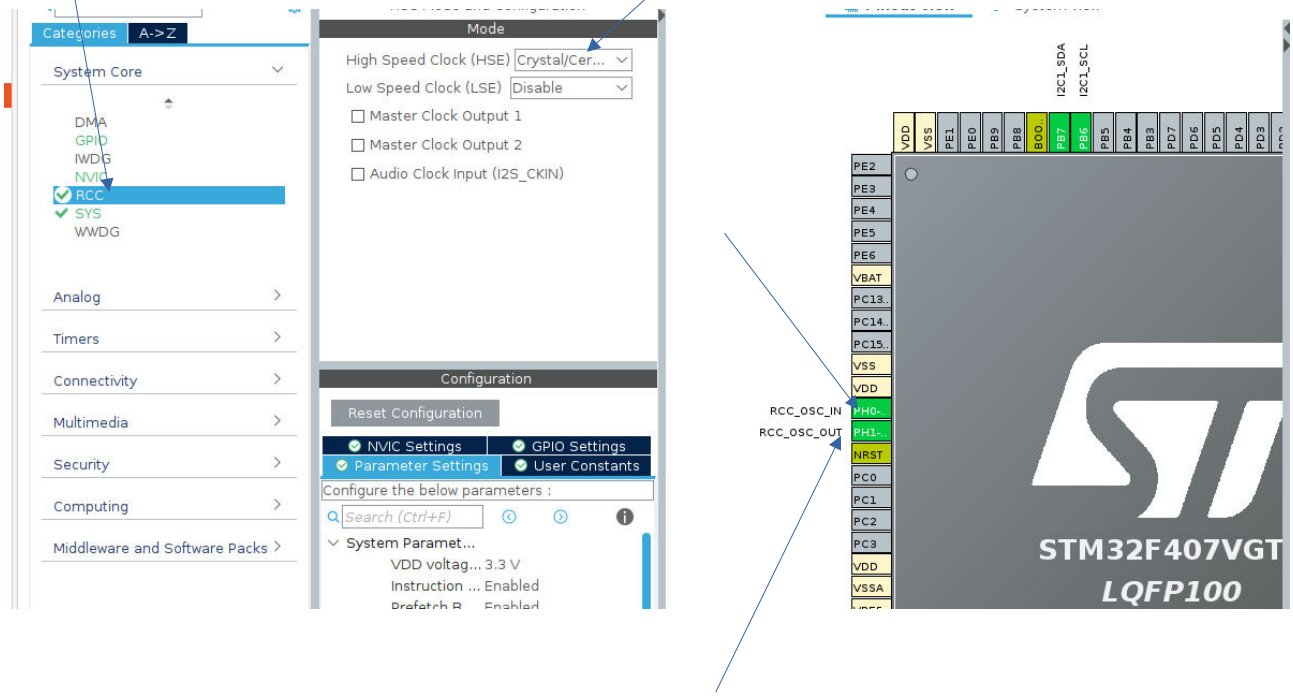
Step 2: Download zip file for BMP180 package and library



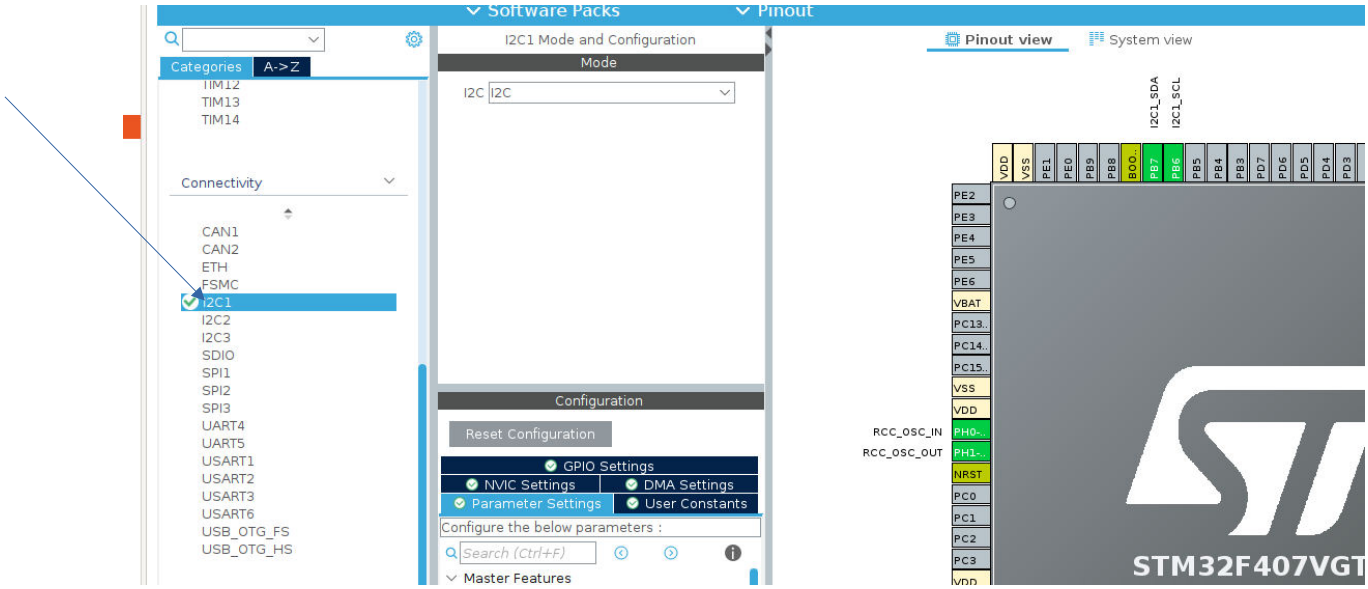
Step 3: Unzip the file for using this



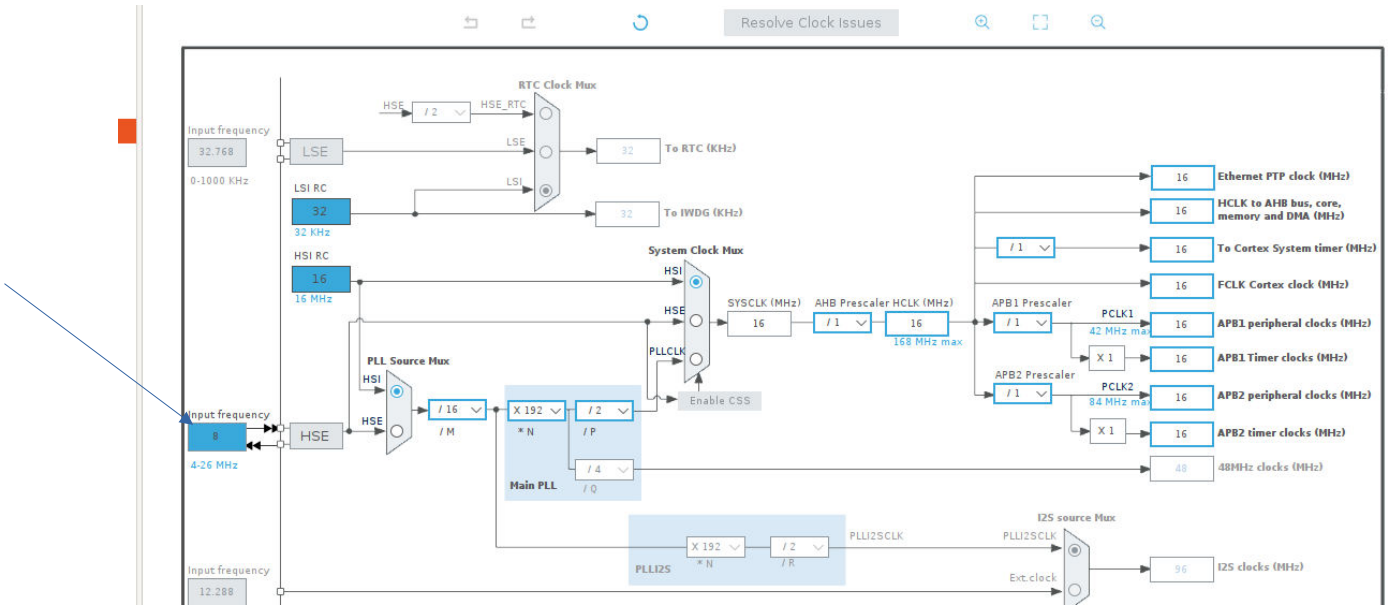
Step 4: Make a project on the stm2 cube ide and then make changes in ioc file something like this RCC as Crystal/ceramic



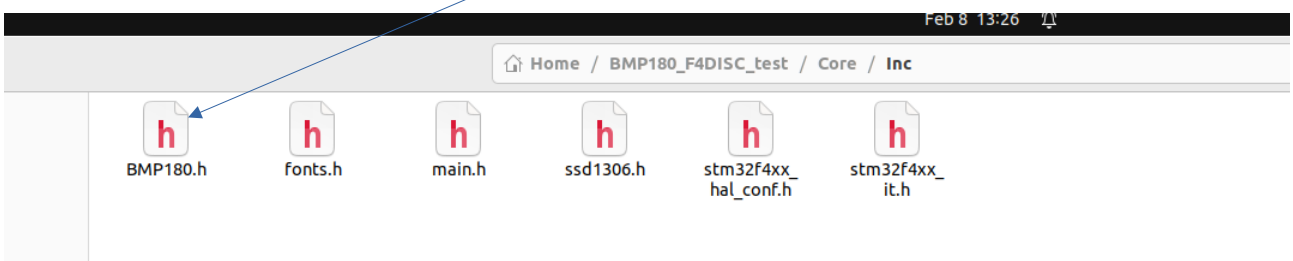
Step 5: Enable i2c feature



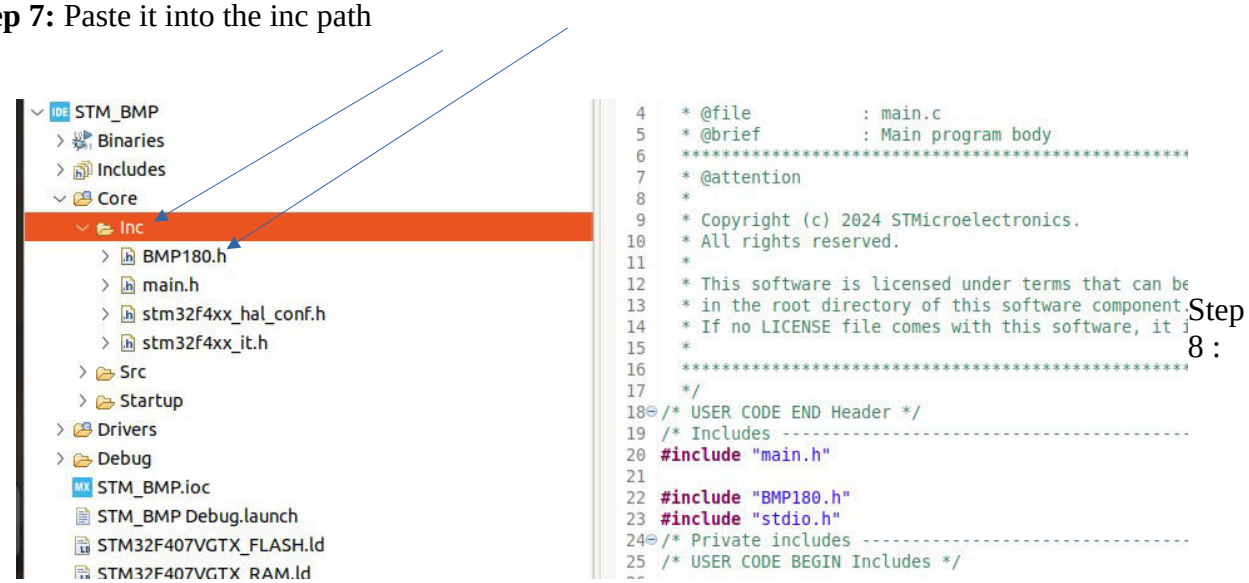
Step 6: Clock configuration has been given below.There is only one changes



Step 7 : After above step copy BMP180.h

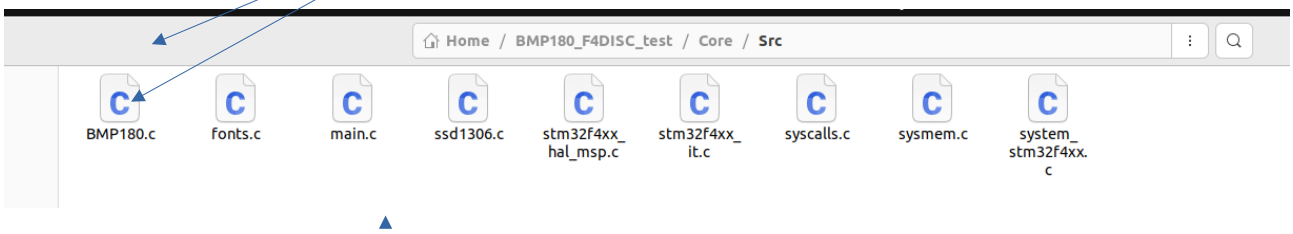


Step 7: Paste it into the inc path

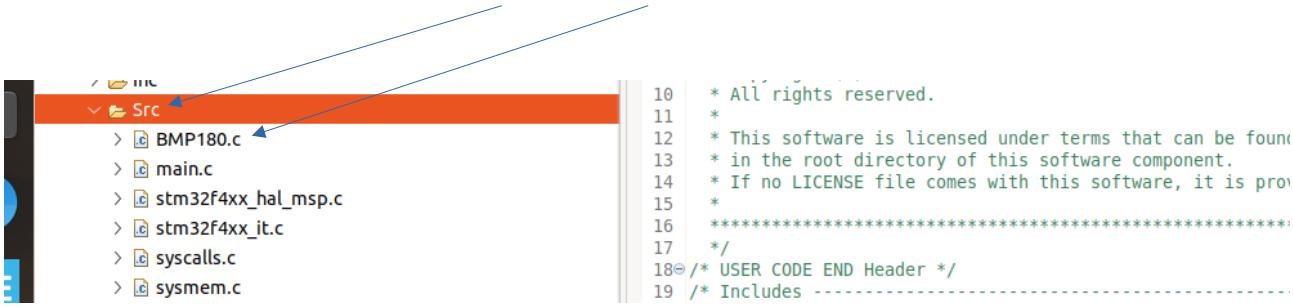


Step 8 :

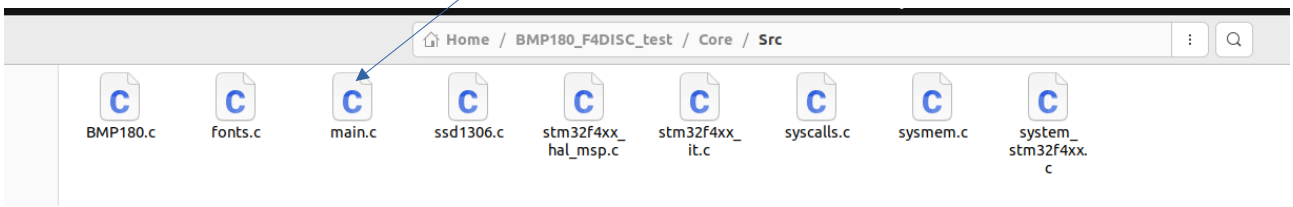
Then copy the BMP180.c from downloaded folder



Step 8: Paste into the src path



Step 9: make chaneges in the main.c file according to downloaded file as per main.c



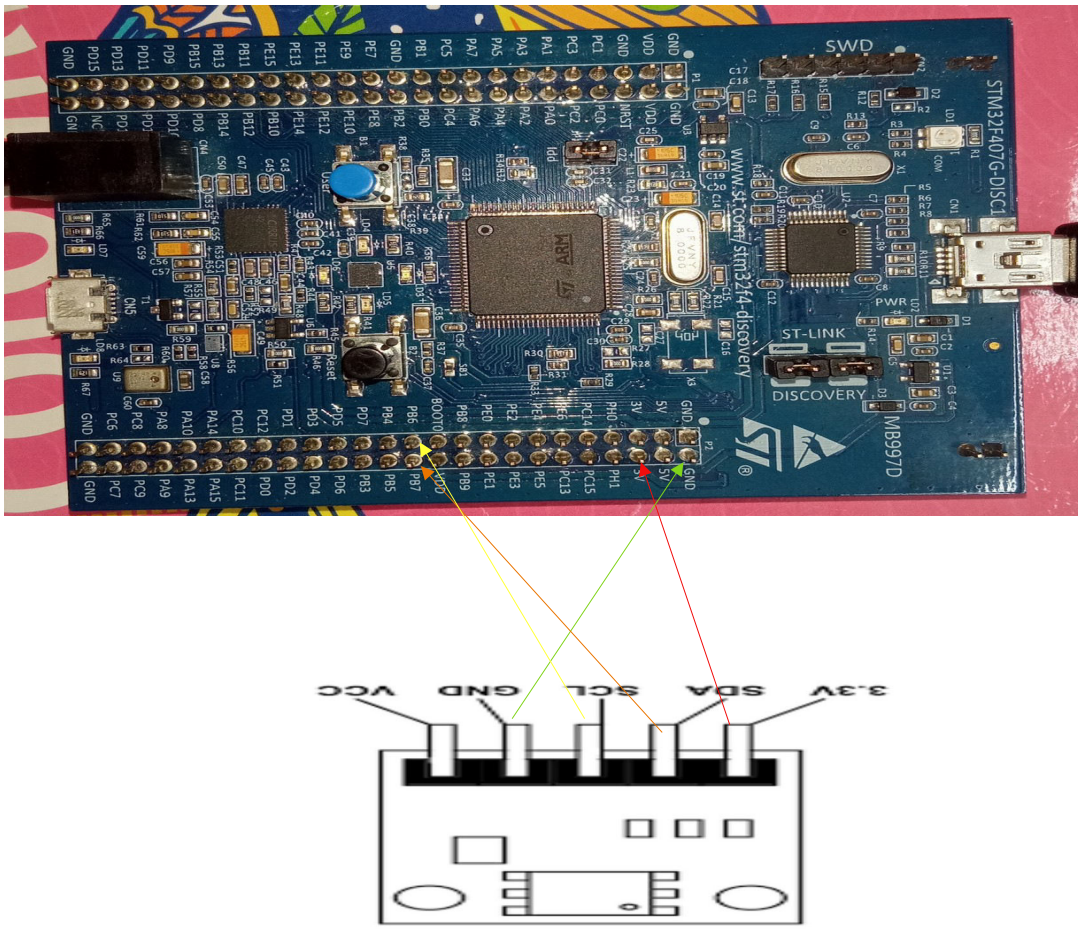
Step 9 : Changes highlighted with red color

```
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2024 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"

#include "BMP180.h"
#include "stdio.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */

/* 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 ---------------------------------------------------------*/
I2C_HandleTypeDef hi2c1;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
float Temperature = 0;
```

```c
float Pressure = 0;
float Altitude = 0;

char Temperature1[10];
char Pressure1[10];
char Altitude1[10];
/* 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_I2C1_Init();
  /* USER CODE BEGIN 2 */
  BMP180_Start();
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
        Temperature = BMP180_GetTemp();

              Pressure = BMP180_GetPress (0);

              Altitude = BMP180_GetAlt(0);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```

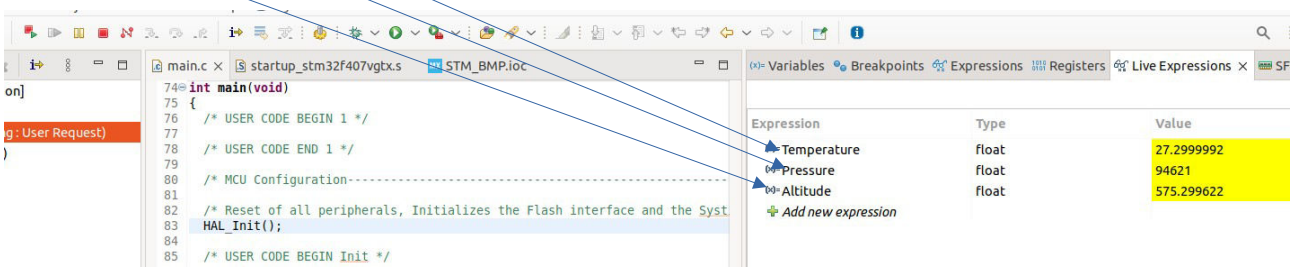
Note: No furthere changes in this code

Step 10: Do not connect vcc and 3.3 together always use one pin and according to i2c configuration your pin may change configuration



Step 11 : During the program execution you need to declare 3 variable in the Live expression such as

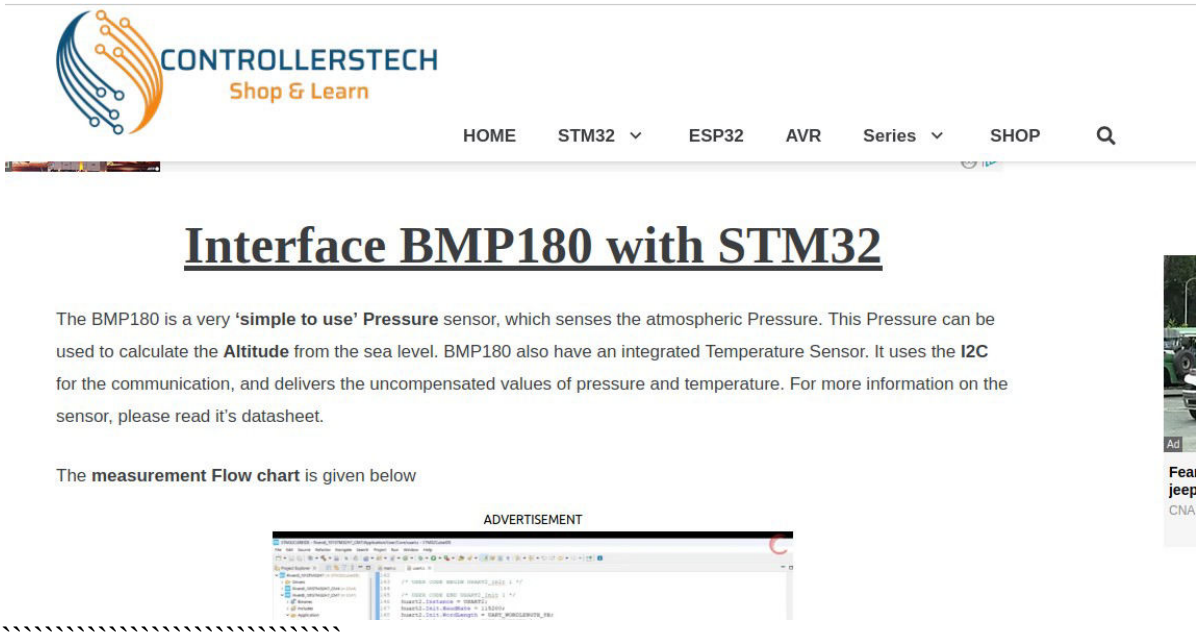
1.Temprature
2.Presssure
3.Altitude



INTERFACING BMP180 WITH OLED DISPLAY MODULE(0.96 inch I2C-WHITE)

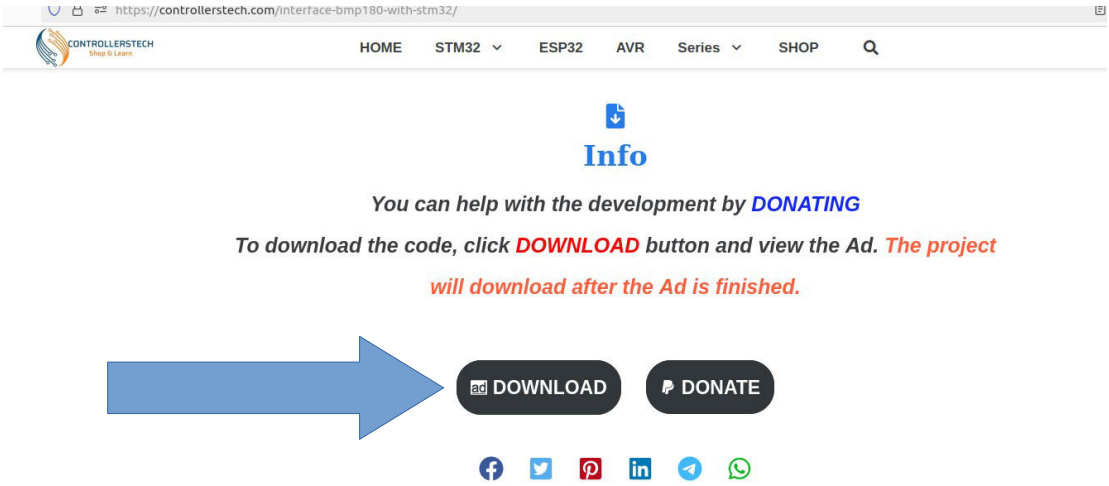
Objective : In this project we will interface with oled display for showing the temprature ,pressure ,and altitude data with STM32.Essential step has been given below

Step 1: We have to download the BMP180 library zip file so first visit this site
https://controllerstech.com/interface-bmp180-with-stm32

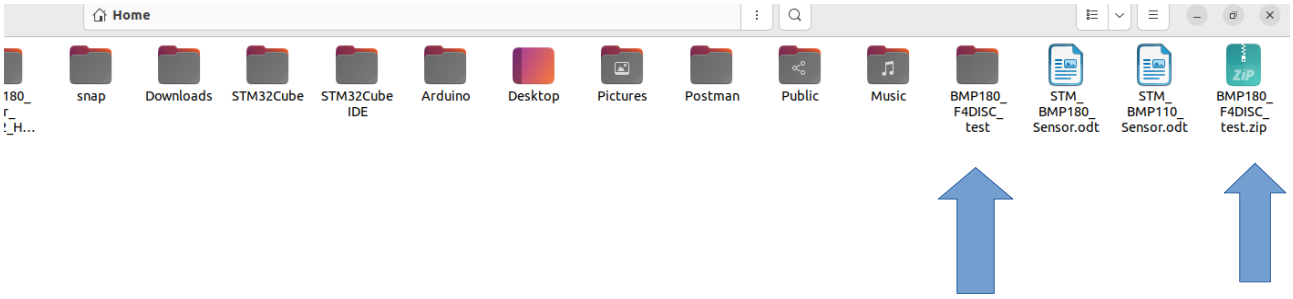



```
```

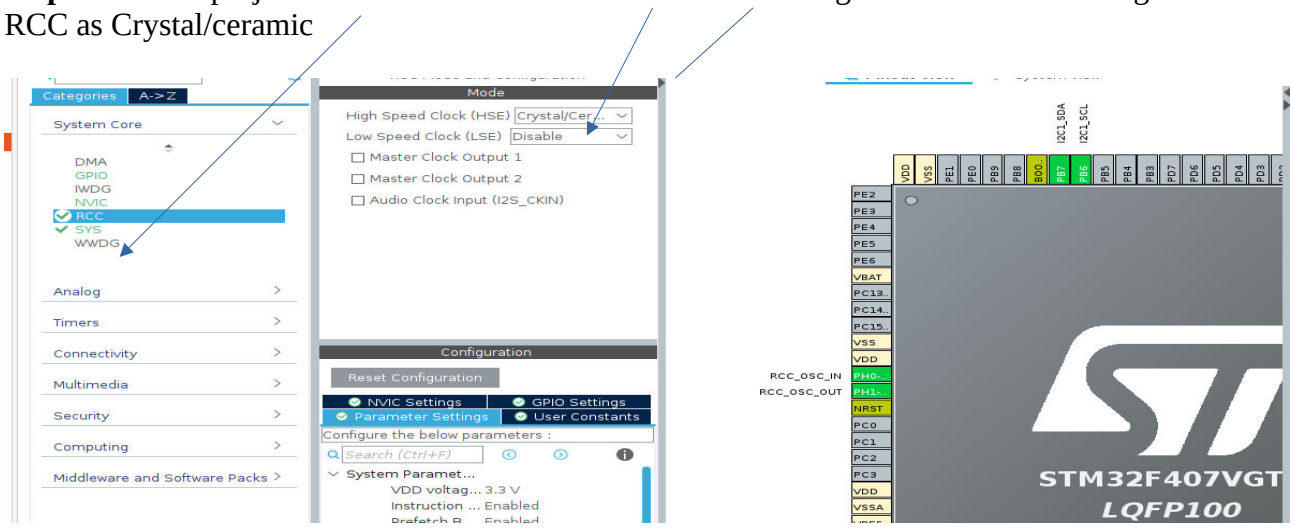
Step 2: Download zip file for BMP180 package and library



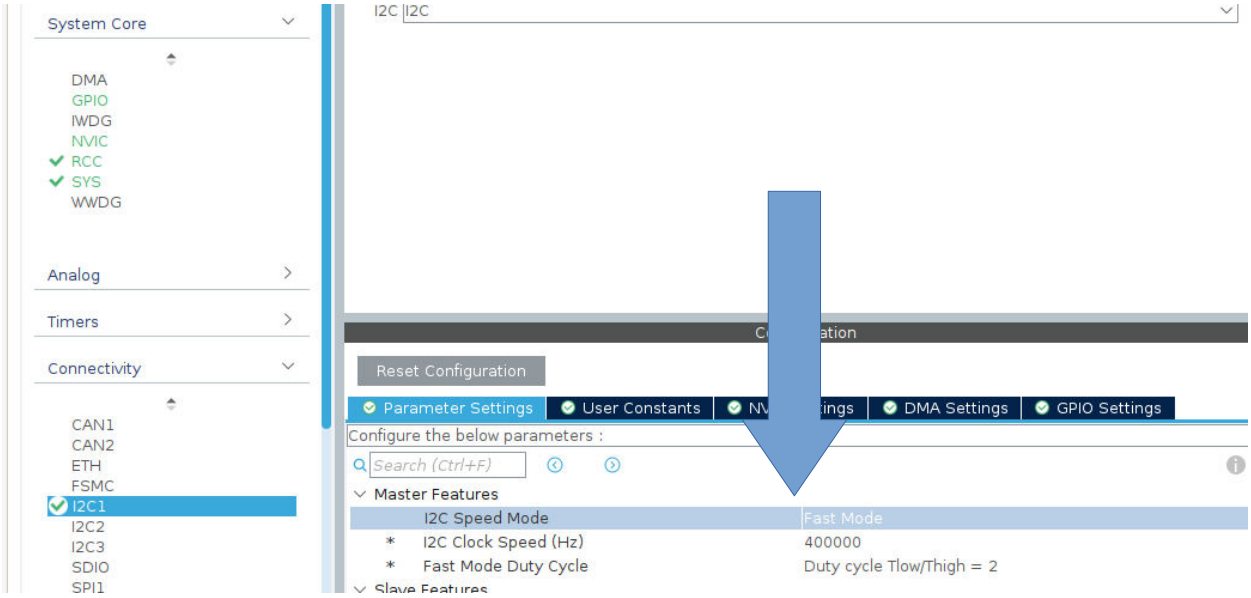
Step 3: Unzip the file



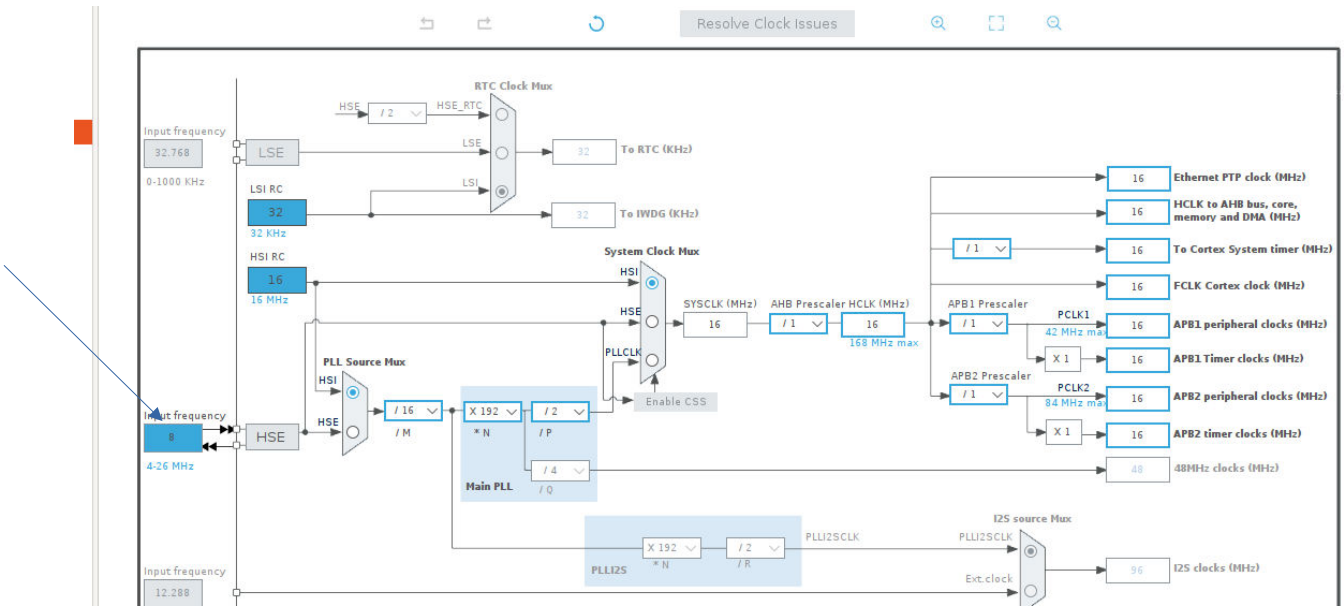
Step 4: Make a project on the stm2 cube ide and then make changes in ioc file something like this RCC as Crystal/ceramic



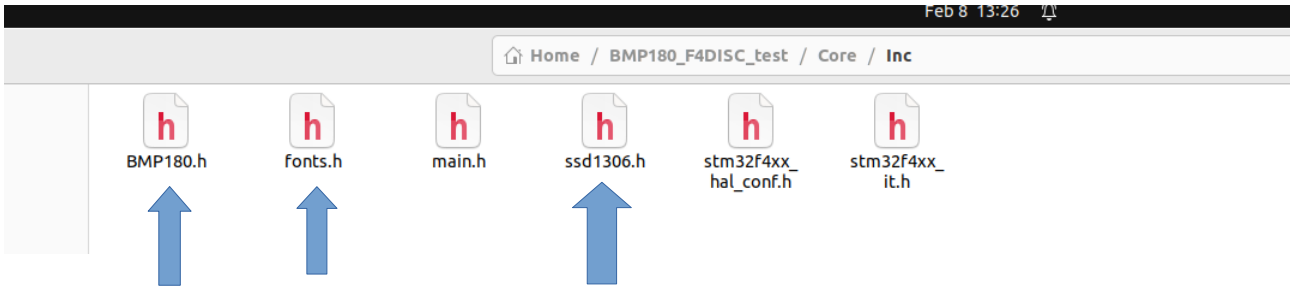
Step 5: Enable i2c feature in **Fast mode**



Step 5 : Clock Configuration only one change here you can use any frequency as per use

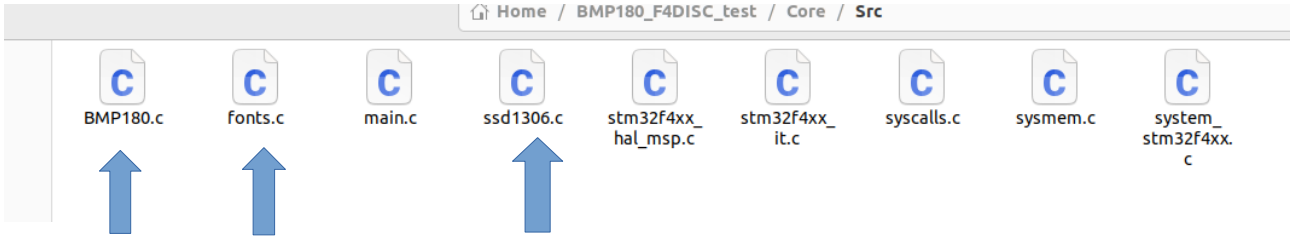


Step 7 : After above step copy BMP180.h fonts.h nad ssd1306.h and paste in the inc path.Take this file from downloaded file

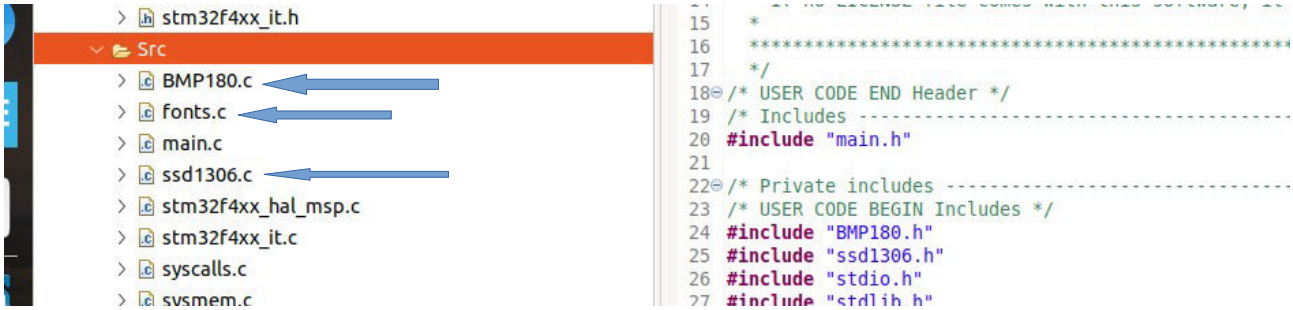


Step 8 : Paste in inc path

Step 9: Copy three file from downloaded zip file



Step 9: Paste in src path



Step 10 :

```
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2024 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 "BMP180.h"
#include "ssd1306.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.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 ---------------------------------------------------------*/
I2C_HandleTypeDef hi2c1;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
float Temperature = 0;
float Pressure = 0;
float Altitude = 0;

char Temperature1[10];
char Pressure1[10];
char Altitude1[10];
/* 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 */
```

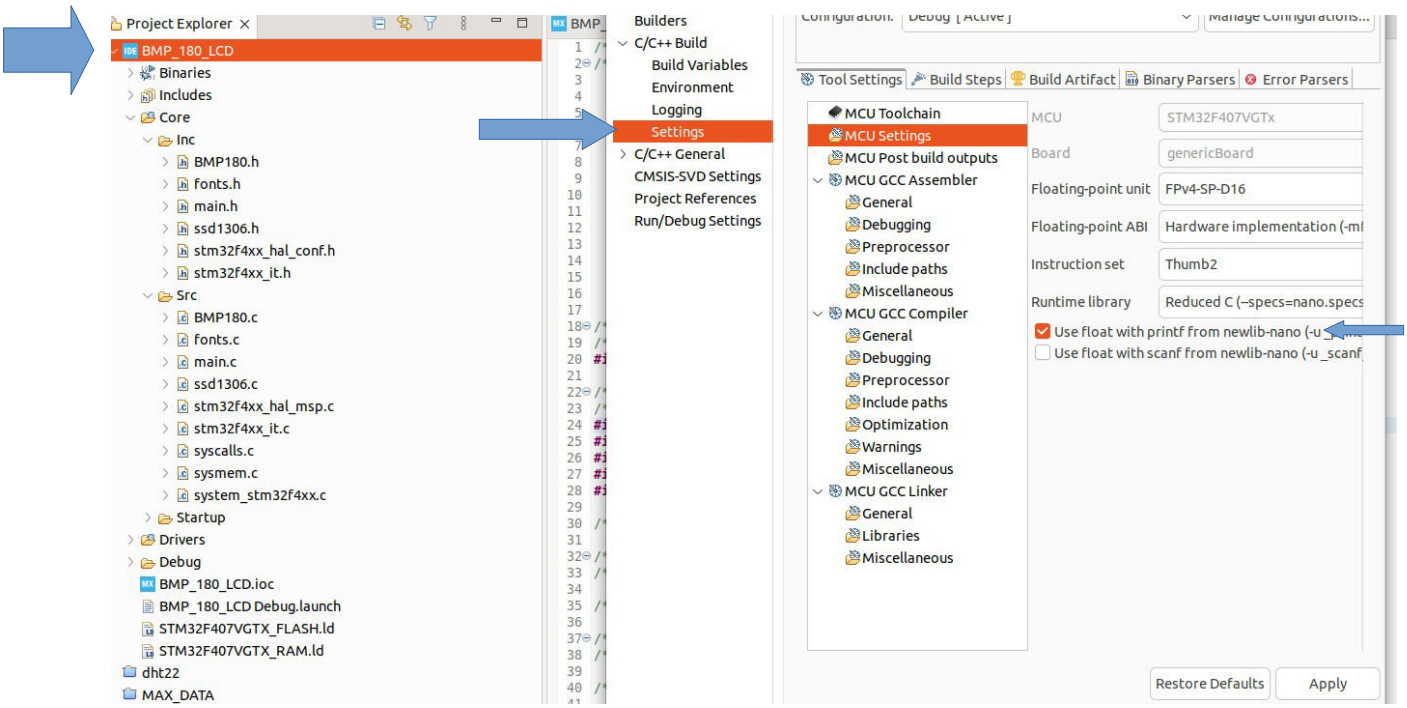
```
  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_I2C1_Init();
  /* USER CODE BEGIN 2 */

  BMP180_Start();

  SSD1306_Init();

  SSD1306_GotoXY (35,0);
  SSD1306_Puts ("BMP180", &Font_11x18, 1);
  SSD1306_GotoXY (10,20);
  SSD1306_Puts ("Barometric", &Font_11x18, 1);
  SSD1306_GotoXY (30,40);
  SSD1306_Puts ("Sensor", &Font_11x18, 1);
  SSD1306_GotoXY (20,40);
  SSD1306_UpdateScreen(); //display
  HAL_Delay(2000);
  SSD1306_Clear();
  /* USER CODE END 2 */

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

        Temperature = BMP180_GetTemp();

        Pressure = BMP180_GetPress (0);

        Altitude = BMP180_GetAlt(0);

/***********To display on OLED**************/

        SSD1306_GotoXY (35,0);
        SSD1306_Puts ("BMP180", &Font_11x18, 1);
        SSD1306_GotoXY (0,0);
        SSD1306_Puts ("Temperature", &Font_11x18, 1);
        SSD1306_GotoXY (20,40);
        sprintf(Temperature1, "%.2f", Temperature);
        SSD1306_Puts(Temperature1, &Font_11x18, 1);
        SSD1306_DrawCircle(80, 40, 2, 1);  //To print degree only
        SSD1306_GotoXY (85,40);  //To print celcius
        SSD1306_Puts ("C", &Font_11x18, 1);
        SSD1306_UpdateScreen(); //display
        HAL_Delay(2000);
        SSD1306_Clear();


        SSD1306_GotoXY (20,0);
        SSD1306_Puts ("Pressure", &Font_11x18, 1);
        SSD1306_GotoXY (10,40);
        sprintf(Pressure1, "%.2f", Pressure);
        SSD1306_Puts(Pressure1, &Font_11x18, 1);
        SSD1306_GotoXY (100,40);
        SSD1306_Puts ("pa", &Font_11x18, 1);
        SSD1306_UpdateScreen(); //display
        HAL_Delay(2000);
        SSD1306_Clear();

        SSD1306_GotoXY (20,0);
        SSD1306_Puts ("Altitude", &Font_11x18, 1);
```

```
        SSD1306_GotoXY (15,40);
        sprintf(Altitude1, "%.2f", Altitude);
        SSD1306_Puts(Altitude1, &Font_11x18, 1);
        SSD1306_GotoXY (90,40);
        SSD1306_Puts ("m", &Font_11x18, 1);
        SSD1306_UpdateScreen(); //display
        HAL_Delay(2000);
        SSD1306_Clear();

        HAL_Delay (2000);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
============================No further changes =========================
```

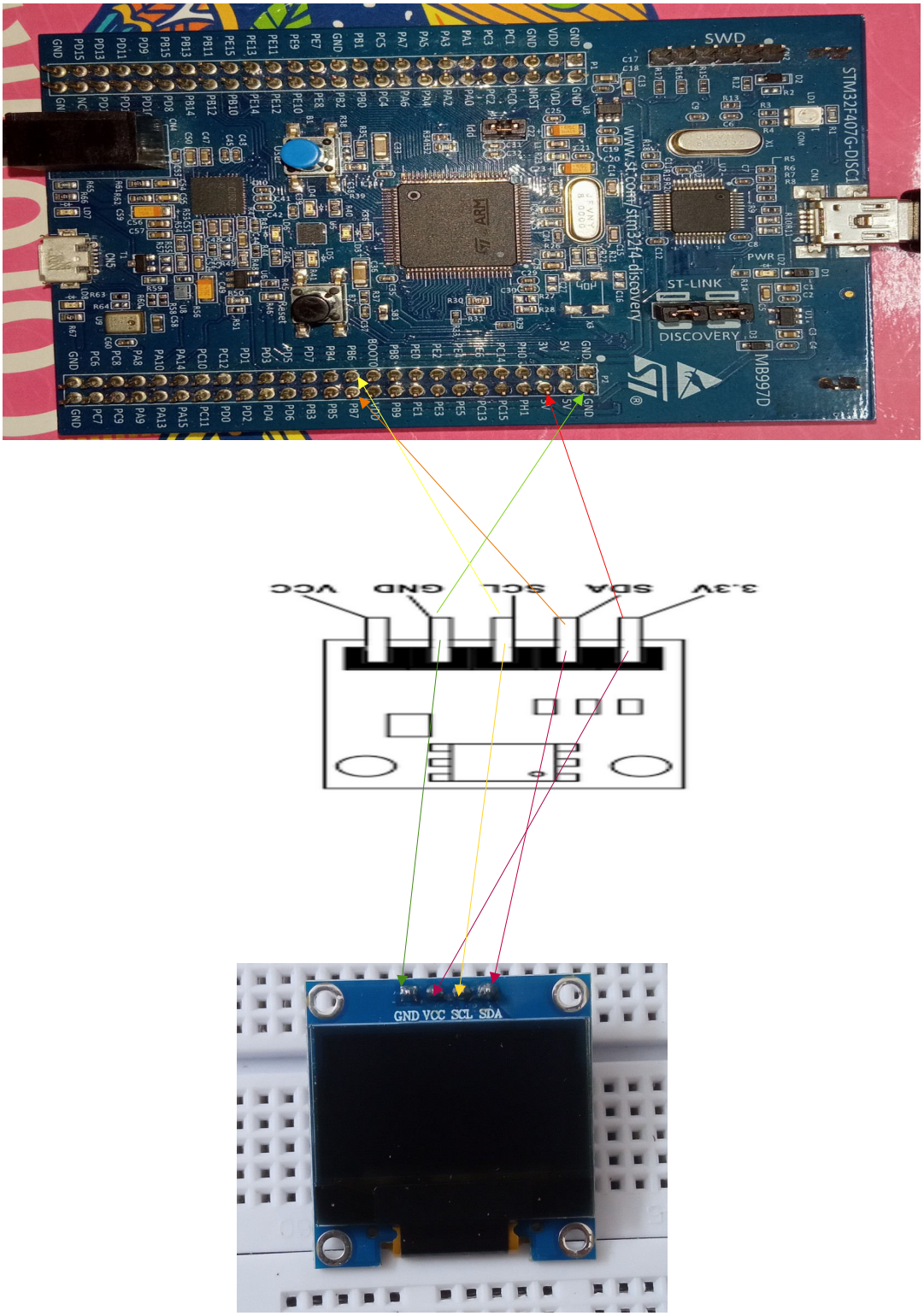
Step 11 : you may get one problem regarding below

```
  sprintf(Temperature1, "%.2f", Temperature);
  sprintf(Pressure1, "%.2f", Pressure);
  sprintf(Altitude1, "%.2f", Altitude);
```

for solving this make changes in setting (BMP180_LCD then Properties then settings then MCU setting then use float with printf option click and apply



Step 10: Do not connect vcc and 3.3 together always use one pin and according to i2c configuration your pin may change configuration



Step 11 : During the program execution you need to declare 3 variable in the Live expression such as

1.Temprature
2.Pressure
3.Altitude