



AWaDH

Agriculture and Water
Technology Development Hub

INTERFACING OF bme680 Sensor

What will you learn from this module:

We will be able to find the value of temperature, humidity, pressure and gas using bme680 Sensor and Development Board.

Requirements:

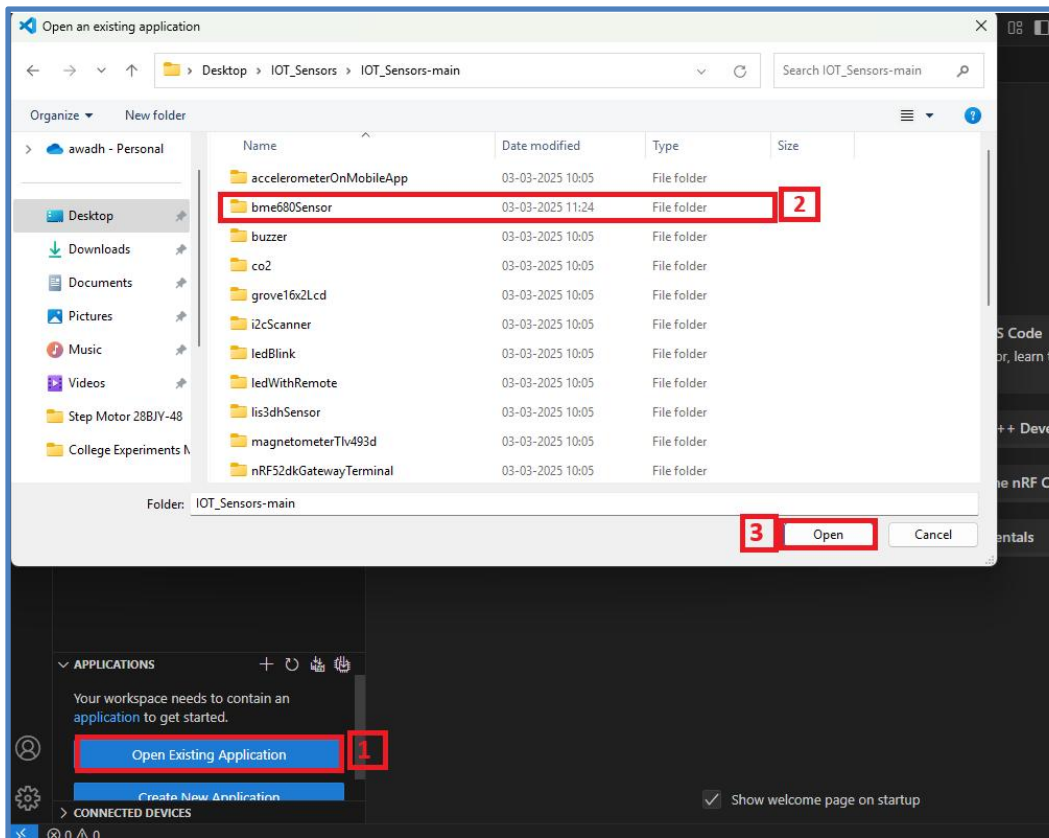
- nRF connect desktop software.
- nRF Command line tools.
- Visual studio code.
- USB cable.
- nRF52832 Development Board.
- Bme680 Sensor.

Prerequisites:

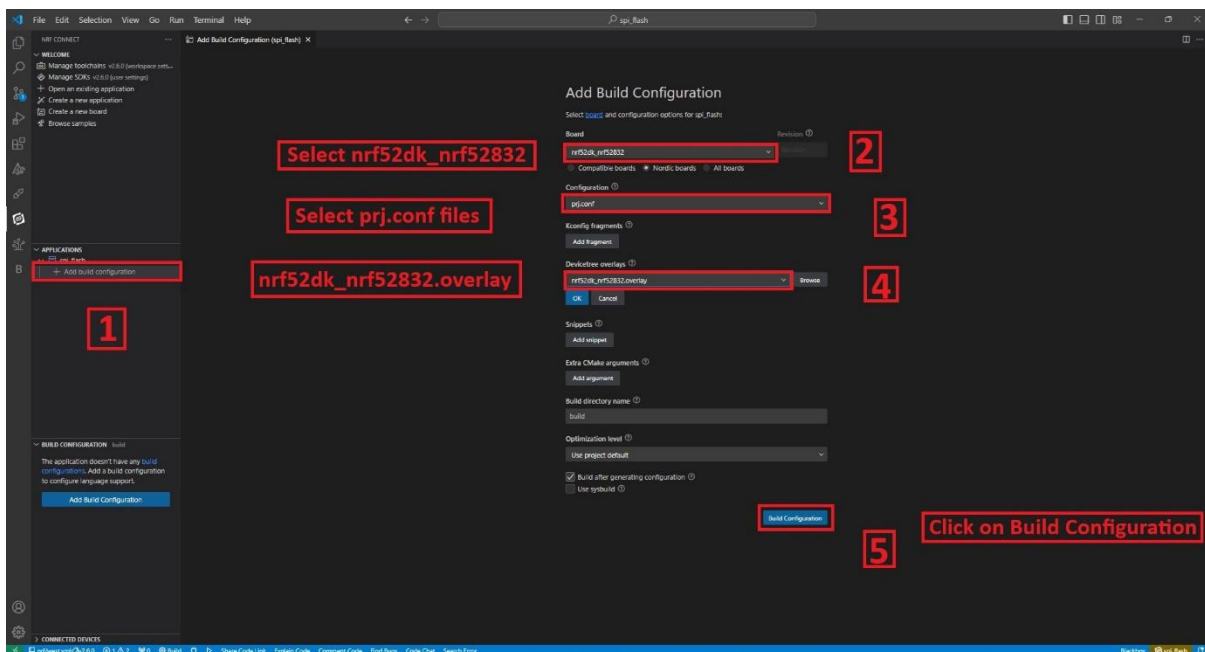
- Basic knowledge of C/C++
- Basic knowledge of communication protocol.
- Basic project setup.

Setup and Configuration

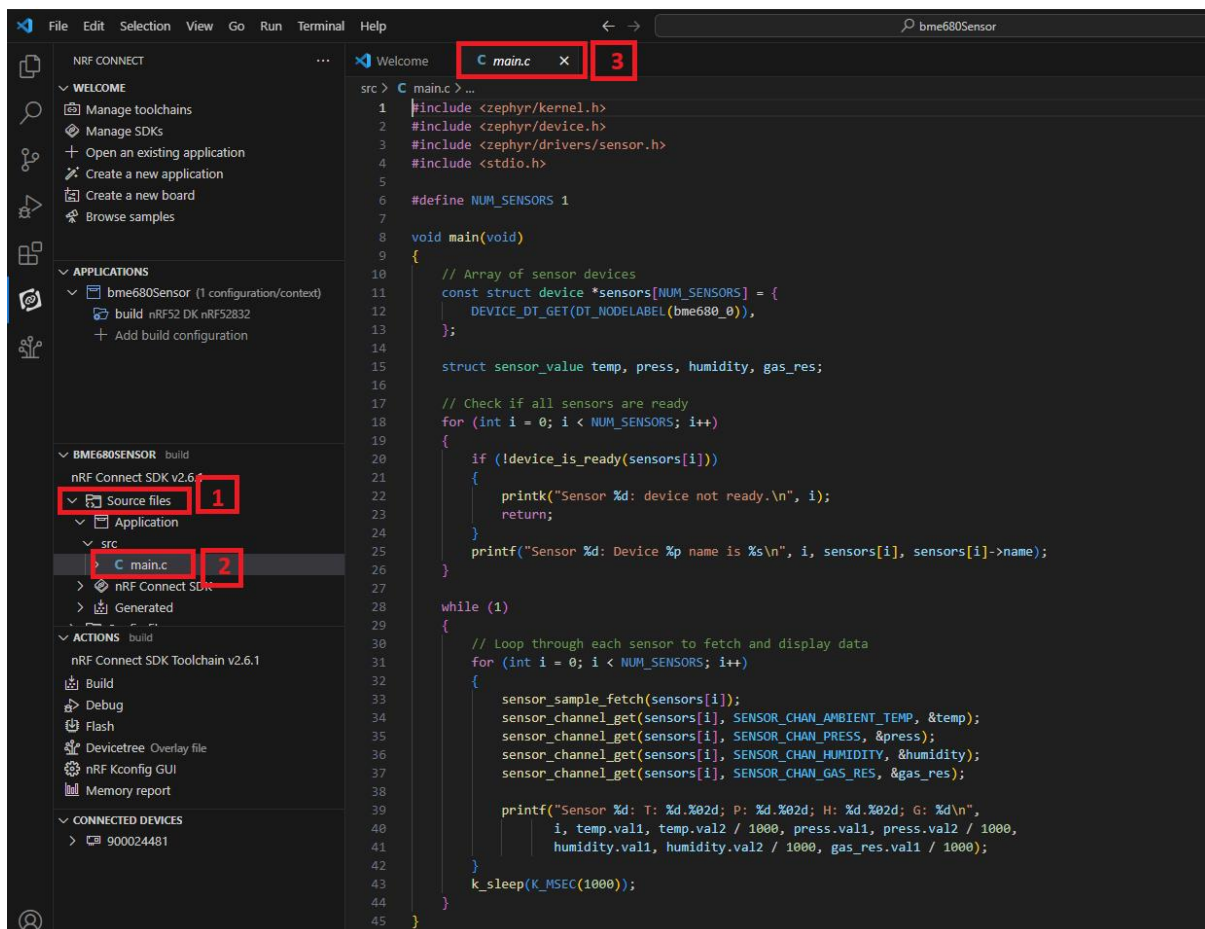
- Open VS Code and click on **Open Existing Application [1]** > click on **bme680 Sensor [2]** > **Open [3]** as shown in the picture below.



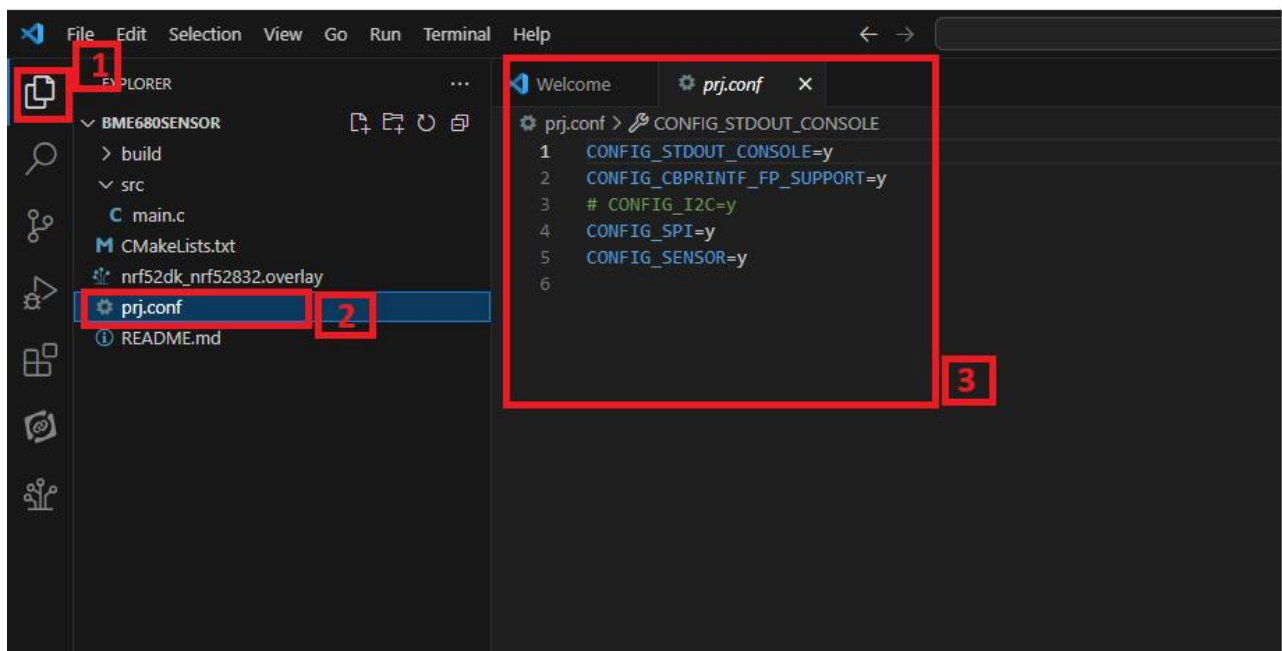
- Click on **Create new build configuration** [1]. Here you can change the board version, if you are using nRF52832, then select **nrf52dk_nrf52832** [2] or you can change from dropdown menu for another version like nRF52833 etc.
- Click on the Configuration and select **prj.conf** [3] from dropdown menu and then click on the devicetree overlay & select **nrf52dk_nrf52832.overlay** [4].
- Then click on the **Build Configuration** [5] as shown below in the picture.



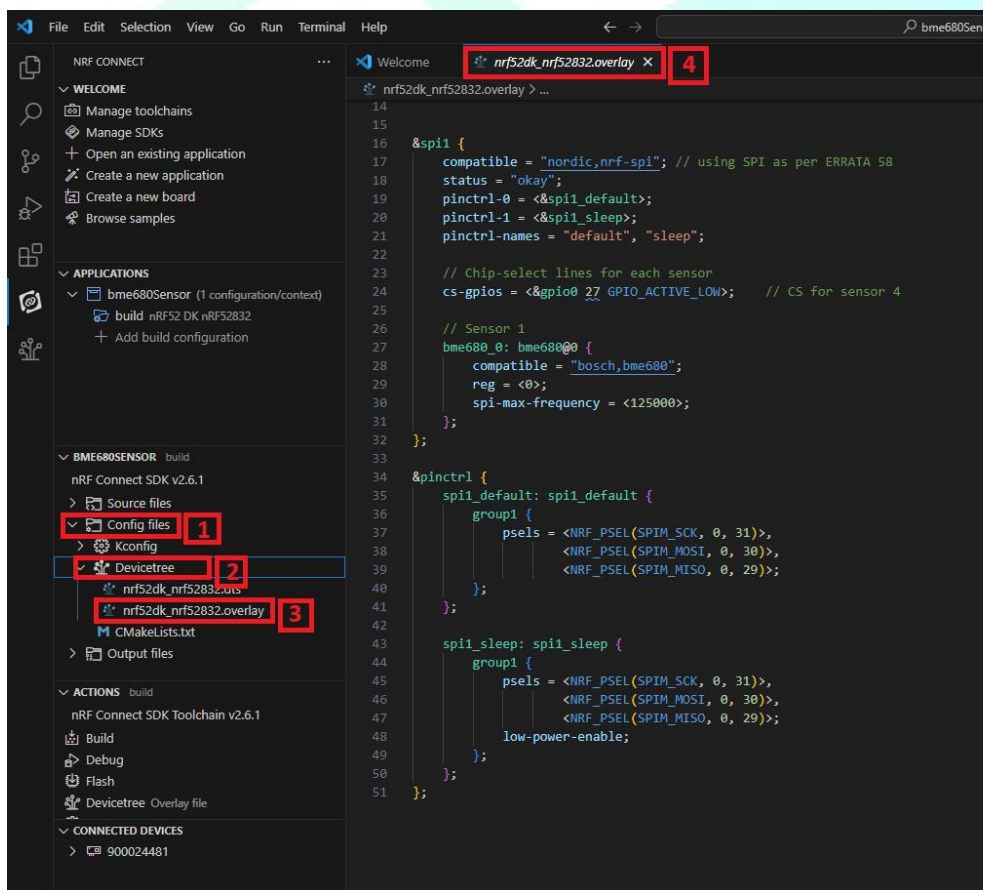
- Go to source file, click **source file [1]** > click on **Application** > click on **src** > click on **main.c [2]**.
- By clicking on **main.c** file and you will see the code on your screen [3].



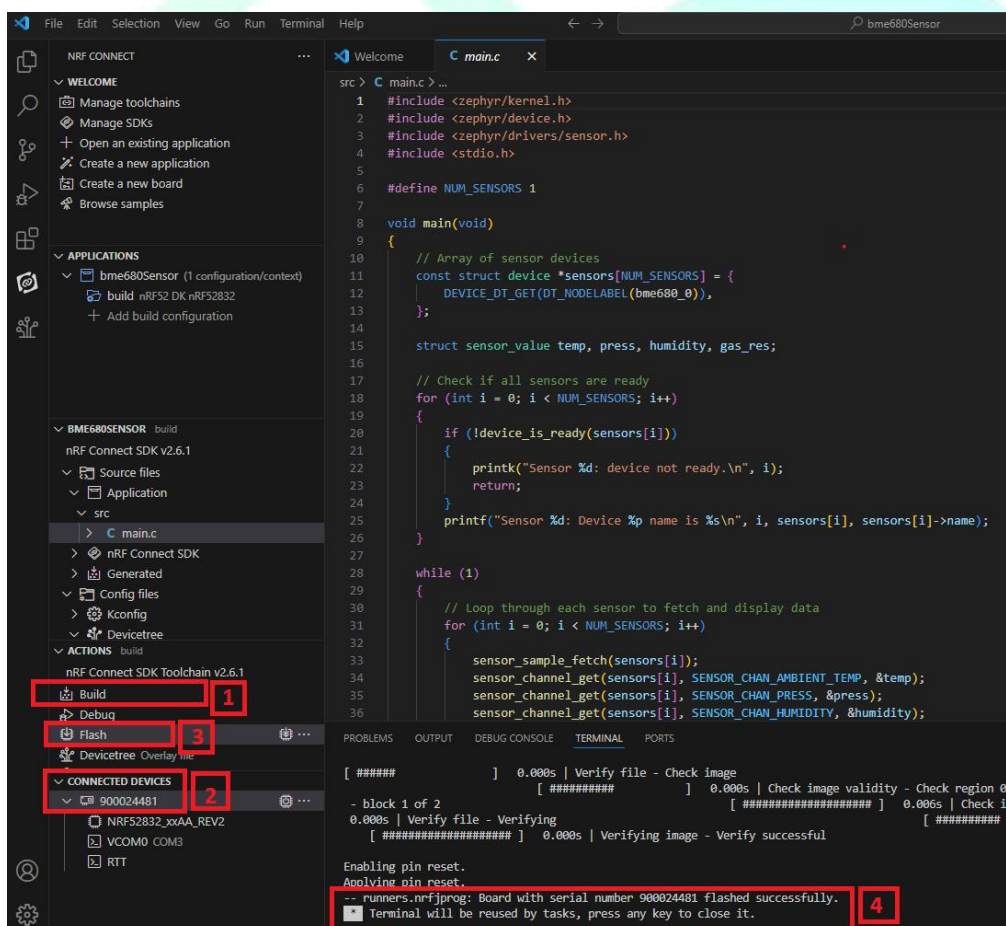
- To configure the prj configuration, click on **Config files [1]** > click on **Kconfig** > click on **prj.conf [2]**.
- The prj configuration will appear on your screen **[3]** as shown in the picture below.



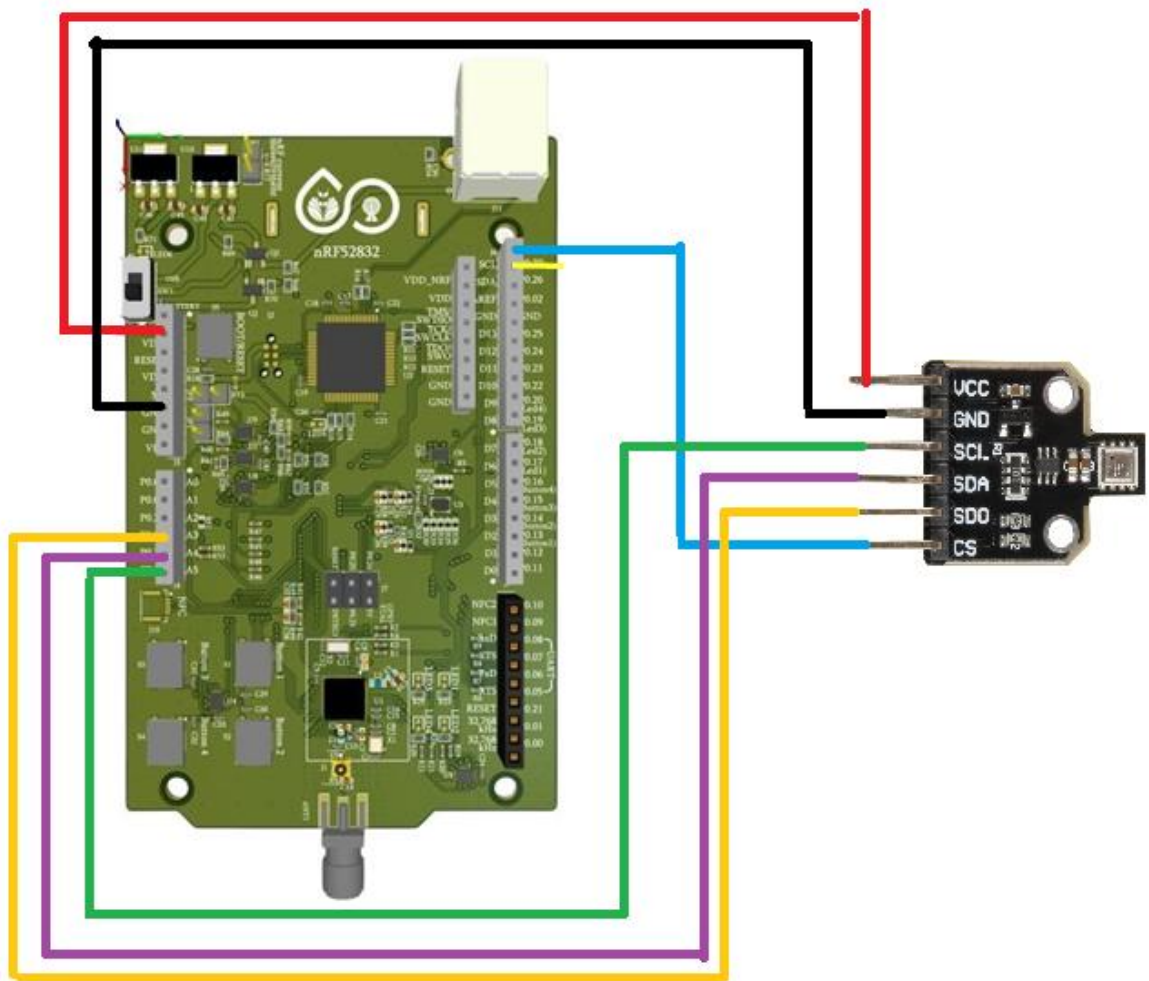
- To configure the i2c protocol, you have to enable it in the **overlay file**.
- Click on the **Config files [1]** > click on **Kconfig** > click on **Devicetree [2]** > click on **nrf52dk_nrf52832.overlay [3]**.
- The .overlay file will appear on your screen and add the given code to the .overlay file as shown in the picture given below [4].



- Click on **Build [1]** configuration again and check the **CONNECTED DEVICES [2]**.
- If device id is visible, then **Flash [3]** the code in Dev Kit.
- If **flashed successfully [4]** message is displayed on serial terminal, then flash process is complete.



❖ Pin Configuration



Board Pins -> Sensor Pins

VDD -> VCC

GND -> GND

SCL -> P0.31

SDA -> P0.30 (MOSI)

SDO -> P0.29 (MISO)

CS -> P0.27



OUTPUT

The screenshot displays the Zephyr development environment. The left sidebar shows the project structure for 'bme680Sensor'. The main editor shows the source code for 'main.c'. The terminal window at the bottom shows the output of the program, which is a series of sensor readings for temperature, pressure, humidity, and gas.

```
src > C main.c > ...
1  #include <zephyr/kernel.h>
2  #include <zephyr/device.h>
3  #include <zephyr/drivers/sensor.h>
4  #include <stdio.h>
5
6  #define NUM_SENSORS 1
7
8  void main(void)
9  {
10     // Array of sensor devices
11     const struct device *sensors[NUM_SENSORS] = {
12         DEVICE_DT_GET(DT_NODELABEL(bme680_0)),
13     };
14
15     struct sensor_value temp, press, humidity, gas_res;
16
17     // Check if all sensors are ready
18     for (int i = 0; i < NUM_SENSORS; i++)
19     {
20         if (!device_is_ready(sensors[i]))
21         {
22             printk("Sensor %d: device not ready.\n", i);
23             return;
24         }
25         printf("Sensor %d: Device %p name is %s\n", i, sensors[i], sensors[i]->name);
26     }
27
28     while (1)
29     {
30         // Loop through each sensor to fetch and display data
31         for (int i = 0; i < NUM_SENSORS; i++)
32         {
33             sensor_sample_fetch(sensors[i]);
34             sensor_channel_get(sensors[i], SENSOR_CHAN_AMBIENT_TEMP, &temp);
35             sensor_channel_get(sensors[i], SENSOR_CHAN_PRESS, &press);
36             sensor_channel_get(sensors[i], SENSOR_CHAN_HUMIDITY, &humidity);
```

Terminal Output:

```
Sensor 0: T: 25.270; P: 98.117; H: 48.258; G: 58
Sensor 0: T: 25.290; P: 98.115; H: 48.194; G: 60
Sensor 0: T: 25.300; P: 98.113; H: 48.150; G: 61
Sensor 0: T: 25.320; P: 98.113; H: 48.129; G: 62
Sensor 0: T: 25.330; P: 98.113; H: 48.67; G: 63
Sensor 0: T: 25.340; P: 98.117; H: 48.27; G: 64
Sensor 0: T: 25.350; P: 98.115; H: 47.997; G: 66
Sensor 0: T: 25.350; P: 98.113; H: 47.961; G: 67
Sensor 0: T: 25.360; P: 98.113; H: 47.915; G: 68
```

- The value of temperature, pressure, humidity and gas is received at the output as shown in above figure.