

International Institute of Information Technology

Introduction to IoT

Lab 3

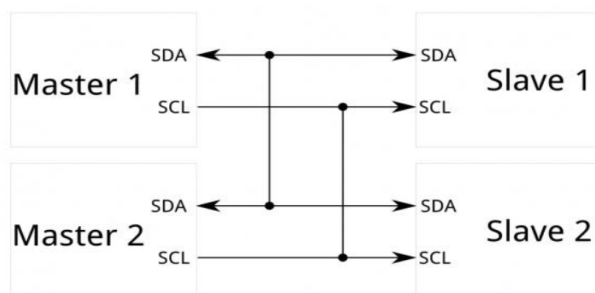
Overview:

In this lab session, you will use the bmp180 sensor to read the values of pressure, and temperature using I2C Module and display them in the Serial Monitor.

I2C Communication:

I2C stands for **Inter-Integrated Circuit**. It is a bus interface connection protocol incorporated into devices for serial communication.

It allows multiple slaves to communicate with one or more masters.



I2C Bus (Interface wires) consists of just two wires and are named as Serial Clock Line (SCL) and Serial Data Line (SDA). The data to be transferred is sent through the SDA wire and is synchronized with the clock signal from SCL.

- The clock signal is always generated by the current bus master.
- I2C bus drivers are "open drain", meaning that they can pull the corresponding signal line low, but cannot drive it high.
- Thus, there can be no bus contention where one device is trying to drive the line high while another tries to pull it low.

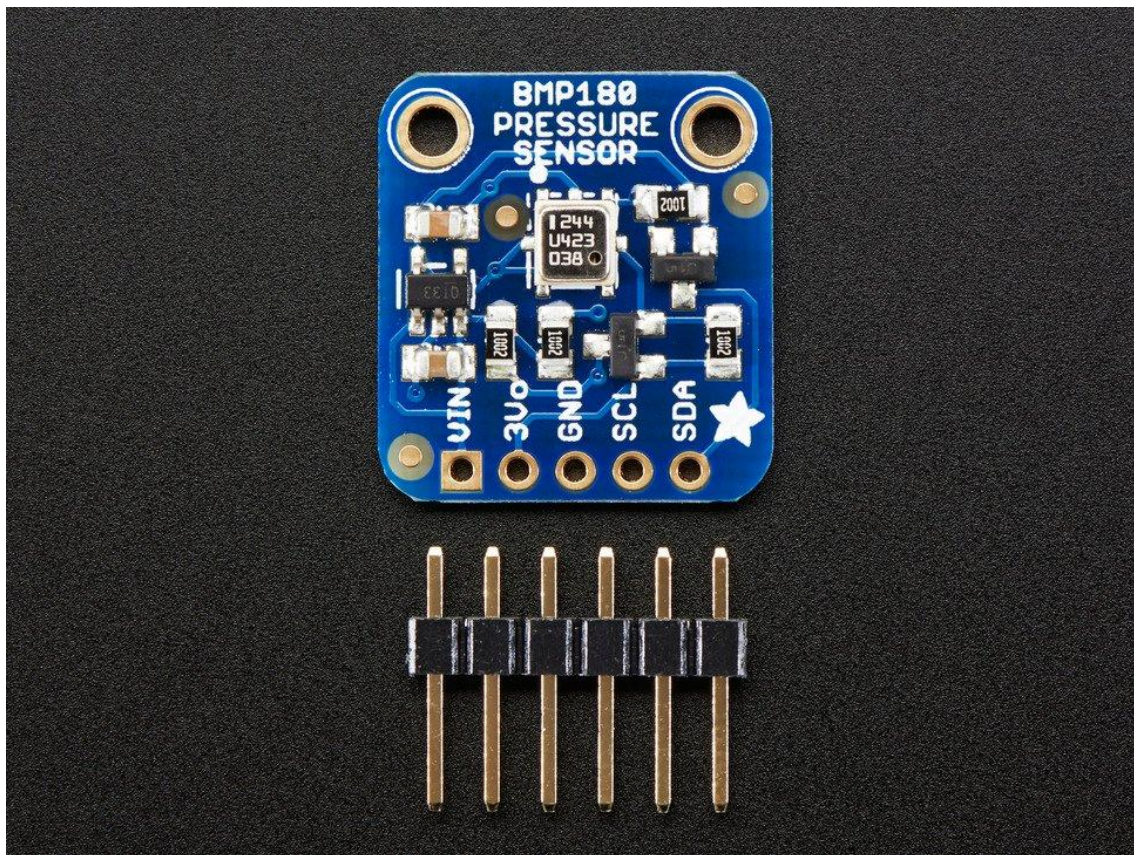
- Each signal line has a pull-up resistor on it, to restore the signal to high when no device is asserting it low.
- Because the devices on the bus don't actually drive the signals high, I2C allows for some flexibility in connecting devices with different I/O voltages.

The ESP32 supports I2C communication through its two I2C bus interfaces that can serve as I2C master or slave, depending on the user's configuration.

The default I2C pins in Esp32 are GPIO 22 (SCL) and GPIO 21 (SDA) but we can configure our code to use any other pins.

Bmp Sensor:

BMP180 is an atmospheric pressure sensor. It is mainly used to measure atmospheric pressure or barometric pressure



- The working principle of the bmp pressure sensor is very simple, it works based on the weight of air. Because the air around us has a certain weight, and this weight has a specific pressure.
- There are only two pins available to communicate with the module. And that communication is the I2C interface. The data is sent to the module or received from the I2C interface module

BMP 180 Specifications:

- Vin: 3 to 5VDC
- Logic: 3 to 5V compliant
- Pressure sensing range: 300-1100 hPa (9000m to -500m above sea level)
- Up to 0.03hPa / 0.25m resolution
- -40 to +85°C operational range, +-2°C temperature accuracy
- This board/chip uses I2C 7-bit address 0x77.

Hardware Required:

- Breadboard
- ESP32
- BMP 180 Sensor
- Jumper wires

Part 1: Read the values of temperature and pressure using the bmp sensor using the default I2C pins in the ESP32.

Circuit Connections:

1. Connect vcc and gnd pins from esp32 to the pins in the bmp sensor respectively.
2. Connect D21 in Esp32 to SDA in the bmp sensor.
3. Connect D22 in Esp32 to SCL in the bmp sensor.

Libraries Required:

[BMP_085 library by Adafruit](#)

You can download it from the github and use it or install it in arduino by this way, Tools -> Manage Libraries -> Search for this library and install it.

PseudoCode:

1. Include the library <Adafruit_BMP085.h>
2. create an Adafruit_BMP085 object called bmp.
3. In the Setup function:
 - a. Use Serial.begin() to begin the serial communication
 - b. Start the bmp object using bmp.begin() function and wait till it is started. bmp.begin() should return true, wait till that becomes true.
4. In the Loop block:
 - a. To read the temperature you just need to use the readTemperature() method on the bmp object.
 - b. Reading the pressure is also straightforward. You use the readPressure() method.
 - c. You can also read altitude like this using a function readAltitude() on the bmp object.
 - d. Print these values in the Serial Monitor using the function Serial.println();

Part 2:

Extend the Code for using the Custom pins in the ESP32 as SDA and SCL for I2C Communication.

Circuit Connections:

Use two different pins for SDA and SCL instead of the default pins and connect those from esp32 to the pins in the bmp sensor respectively.

PseudoCode:

1. Include the library <Wire.h>
2. Declare two pins I2C_SCL and I2C_SDA
3. Create a new TwoWire instance like this .
TwoWire I2CBMP = TwoWire(0);
4. In the Setup block:
 - a. initialize the I2C communication with the pins you've defined earlier. The third parameter is the clock frequency.
I2CBMP.begin(I2C_SDA, I2C_SCL, 100000);
 - b. Modify the bmp function to this:
bmp.begin(0x76,&I2CBMP)
Here 0x76 is the I2C address of the bmp sensor.

Expected Output:

The output values should be the Pressure and Temperature values measured by the BMP180 sensor at each instant it is sensing corresponding to your delay.

***** THE END *****