

Introduction to IoT – Spring '22

Lab 3 – Barometric Pressure Sensing

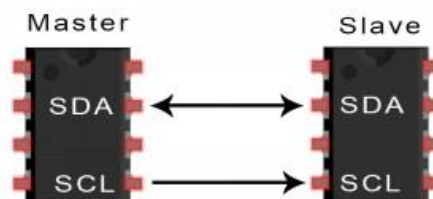
Overview

In this session, you will use a BMP280 sensor to measure atmospheric pressure and temperature. You will interface the sensor and ESP32 using the I2C protocol. This session is divided into two parts –

- Using the default I2C pins in the ESP32 for communication
- Using any other GPIO pins of your choice to establish I2C communication

I2C – Inter Integrated Circuit

- Serial communication protocol used for short distances
- Requires only two wires – Serial Data (SDA) and Serial Clock (SCL)
- Open drain and active low (low voltage = logic 1)
- Half duplex protocol – bi-directional but not simultaneously



BMP280 Sensor

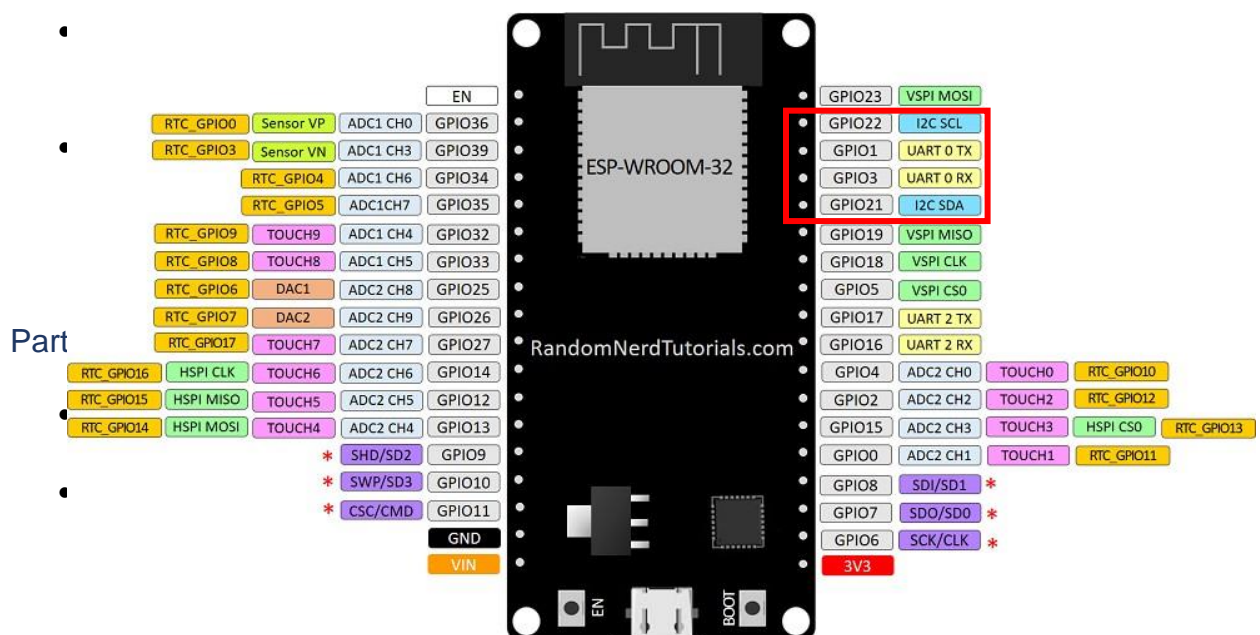


- It is an absolute barometric pressure sensor developed by Bosch Sensortec that can sense both pressure and temperature.

- The sensing ranges are
 - Pressure – 300 to 1100 hPa
 - Temperature – -40 to 85 C
- Since the atmospheric pressure changes with altitude, this sensor can also report approximate altitude.
- It is a MEMS type sensor that detects pressure by measuring how much a diaphragm formed through a capacitive plate is deformed.

Instructions

- The ESP32 supports I2C on GPIO 21 (SDA) and GPIO 22 (SCL) by default.



- The key functions in your code should be [other than `Serial.begin()` for communicating with the serial monitor]
 - `bmp.begin()` – this outputs either a 0 or 1 similar to `Serial.begin()` to indicate that the communication has been established.
 - `bmp.readTemperature()` – reads the temperature value from the sensor in celsius
 - `bmp.readPressure()` – reads the pressure value from the sensor in Pa (pascals)
 - `bmp.readAltitude()` – reads the approximate altitude calculated by the sensor w.r.t sealevel
- None of the above functions need an argument. However, `bmp.begin()` assumes the default chip ID of the BMP sensor as 0x58 and the default I2C address as 0x77. That may not be the case for all of you. Hence, you need to run an [I2CScanner](#) code snippet to find out the I2C address of the sensor given to you.
- If the chip ID is also different, the example code has a section to catch that. If the values you get from the above two procedures are different from the defaults, you need to include them as arguments to the `bmp.begin()` function.

Part 2 – Defining your own I2C pins

- Declare two GPIO pins of your choice as I2C pins – `I2C_SDA` and `I2C_SCL`.
- To initialise them, use a `TwoWire` instance like this (make sure you include the `Wire.h` library)
 - `TwoWire I2CBMP = TwoWire(0);`
- Now, initialise them in the setup block as follows (the third parameter is the clock frequency)
 - `I2CBMP.begin(I2C_SDA, I2C_SCL, 100000);`
- Finally, modify the `bmp` declaration in the initial section of the code as follows
 - `Adafruit_BMP280 bmp → Adafruit_BMP280 bmp(&I2CBMP)`

Expected Output

The serial monitor should display pressure, temperature and altitude values in both parts of the experiment.

Provide the average of all the parameters.