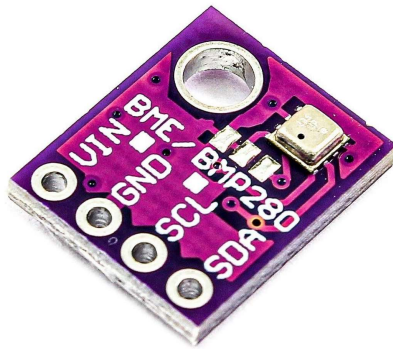


The BME280 I2C is a high precision combined digital pressure, humidity and temperature sensor module with I2C interface and 5V or 3.3V operation.



PACKAGE INCLUDES:

- BME280 I2C Pressure Humidity Temperature Sensor Module
- 4-pin male header

KEY FEATURES OF BME I2C PRESSURE HUMIDITY TEMPERATURE SENSOR MODULE:

- Temperature: -40 to 85°C
- Humidity: 0 to 100%
- Pressure: 300 to 1100 hPa
- Altitude: 0 to 30,000 ft
- I2C Interface
- 3.3V and 5V logic compatible
- 3.3V or 5V operation

The BME280 is a precision Bosch sensor that can be used in many applications from weather monitoring to gaming controls to altitude measurements with enough precision to know if an object has been lifted a couple of feet.

The BME280 is a higher performance version of the BMP280 with much better noise performance and also includes humidity sensing that the BMP280 does not have. Since a common PCB is used for both sensors, if the board is not marked on the silkscreen you can tell which one you have by looking at the first letter on the 2nd line stamped on the sensor. The BME280 will have a 'U' while the BMP280 will have a 'K'.

Compared with the [GY-BME280 module](#), this one exposes the I2C interface only (no SPI), but it includes a voltage regulator and MOSFET logic level shifters and will operate at either 5V or 3.3V. It is also packaged in a smaller module size with 4-pin header instead of 6-pin.

The temperature measurement is used internally to compensate the pressure and humidity sensors and is also available to estimate the ambient temperature as well.

The device has numerous sampling and filtering options that can be employed to optimize the device for specific applications.

Measuring Temperature

The BME280 can measure temperature over the range of -40 to 80°C.

Full accuracy of $\pm 1.0^{\circ}\text{C}$ is obtained over the range of 0 to 65°C. Outside that range, the accuracy can decrease to $\pm 1.5^{\circ}\text{C}$.

Take note of the fact that the temperature measurement is mainly used for internal temperature compensation of the other sensors. The sensor has some amount of self-heating and so the temperature measurement will usually read several degrees higher than actual temperature. If this is important for your application, you may want to compare the BME280 measured temperature against another known measurement device and apply an offset to the reading.

Measuring Humidity

The BME280 can measure humidity over the range of 0 to 100% with an accuracy of $\pm 3\%$.

The sensor can measure up to 100% humidity over the temperature range of 0 to 60°C. At very high or low temperatures, the maximum measurable humidity decreases per the graph on page 10 of the datasheet.

Measuring Pressure

The BME280 can measure pressure over the range of 300 to 1100 hPa with excellent accuracy of ± 1.0 hPa.

Full accuracy is obtained over the temperature range of 0 to 65°C. This gives an elevation measurement accuracy of approximately ± 1 meter. Outside that range the accuracy can decrease to ± 1.7 hPa

Calculating Altitude / Elevation

The BME280 does not measure altitude directly, but it can be calculated using the pressure reading. Most libraries for this device include altitude calculation routines.

Since the device does a very good job of measuring pressure, it can do a very good job of calculating relative altitude. If you have an altitude reading with the device sitting on a table and then move it to the floor, it will show a 2 foot decrease in altitude.

If on the other hand you are trying to measure absolute altitude, such as the altitude of your table relative to sea level, things get more complicated. Since altitude is relative to sea level the device needs to know the current air pressure corrected to sea level so that it has a reference by which to calculate the altitude given the air pressure that it is currently reading.

You can get somewhat close by finding the reported air pressure from a local airport or weather service on-line which are corrected for sea level. In our example program down below, you would enter this value in the SEA_LEVEL_PRESSURE constant. In our example, we have 1013.25 loaded which is a typical reading.

Since the air pressure is constantly changing based on time, location and weather conditions, unless you have an accurate barometer corrected to sea level with you to reference, it will be difficult to get closer than 20-30 feet.

If you know the altitude that the sensor is at, you can also back into a reading by modifying the `SEA_LEVEL_PRESSURE` constant to give you the correct altitude reading. This might be useful if you want to make absolute measurements based on your current elevation, such as sending a rocket up into the air.

Using the I2C Interface

The I2C interface provides MOSFET logic level shifters for use with either 3.3V or 5V MCUs.

The module supports two different I2C addresses, either 0x76 or 0x77 which allows up to 2 sensors to be used on the same bus. As shipped, a trace between pads on the board next to the BME280 chip grounds the SDO pin on the BME280 chip which pulls SDO low so 0x76 is the default address. If it is desired to change the I2C address, cut the trace connecting the two pads and solder short the other two pads which pulls the SDO pin high for address 0x77.

The **SCL** and **SDA** pins connect to the SCL and SDA pins on the MCU.

Module Connections

The module brings out the following connections.

1 x 4 Header

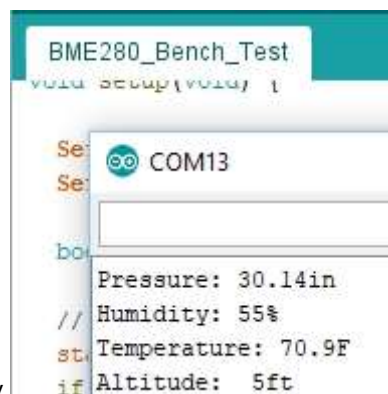
- **VIN**= 3.3 or 5V to match MCU
- **GND** = Ground
- **SCL** = I2C Clock
- **SDA** = I2C Data

Module Assembly

The module ships with the male header strip loose. The header can be soldered to the top or bottom of the module depending on the planned use or wires can be used to make the connections.

For breadboard use, we put the headers on the bottom. Soldering is easiest if the header is inserted into a solderless breadboard to hold it in position during the soldering process.

OUR EVALUATION RESULTS:



```
BME280_Bench_Test
void setup(void) {
  Serial.begin(9600);
  Serial.println("COM13");
  // Pressure: 30.14in
  // Humidity: 55%
  // Temperature: 70.9F
  // Altitude: 5ft
}
```

These are nice little assemblies with good build quality.

The program below is a simple test program which prints the data from the sensor to the Serial Monitor Window.

A similar program that displays the same information on a 4 x 20 LCD to create a mini weather station can be found on the [GY-BME280 page](#).

There are many libraries for the BME280 sensor. In the example here, we are using the Adafruit libraries. Note that the Adafruit_sensor library will need to be manually downloaded from this link: https://github.com/adafruit/Adafruit_Sensor

The other libraries can be downloaded via the IDE library manager.

Just connect VIN to 5V or 3.3V to match the MCU and connect GND to ground. Also connect SCL to SCL on MCU and SDA to SDA on MCU.

Download the program and open the Serial Monitor Window to see the results. Ensure the baud rate is set to 9600.

BME280 Test Program

```
/*  
  BME280 Test Program  
  
  Connect I2C interface to BME280  
    SCL connects to A5 or dedicated SCL pin  
    SDA connects to A4 or dedicated SDA pin  
  Connect BME280 Vcc to 3.3 or 5V and GND to ground  
  Need to install library Adafruit_BME280  
  Need to manually install library Adafruit_Sensor  
*/  
#include <Adafruit_Sensor.h>  
#include <Adafruit_BME280.h>  
#include <Wire.h>  
  
float temperature;  
float humidity;  
float pressure;  
float altitude;  
  
float const ALTITUDE = 81.0;           // Altitude at my location in meters  
float const SEA_LEVEL_PRESSURE = 1013.25; // Pressure at sea level
```

```
Adafruit_BME280 bme; // I2C

//=====
// Initialization
//=====

void setup(void) {

  Serial.begin(9600);
  Serial.println("Reading sensor");

  bool status;

  // default settings
  status = bme.begin(0x76); // The I2C address of the sensor is 0x76
  if (!status) {           // Loop if sensor not found
    Serial.print("Error. Check Connections");
    while (1);
  }
}

//=====
// Main
//=====

void loop() {

  getPressure(); // Get sensor data and print to serial monitor window
  getHumidity();
  getTemperature();
  getAltitude();

  delay(2000); // Update readings every 2 seconds
}

//=====
// getTemperature - Subroutine to get and print temperature
//=====

void getTemperature()
{
  temperature = bme.readTemperature();
}
```

```

temperature = temperature * 9 / 5 + 32; // Convert C to F
String temperatureString = String(temperature, 1); // One decimal position
Serial.print("Temperature: ");
Serial.print(temperatureString);
Serial.println("F");
}
//=====
//  getHumidity - Subroutine to get and print humidity
//=====
void getHumidity()
{
  humidity = bme.readHumidity();
  String humidityString = String(humidity, 0);
  Serial.print("Humidity: ");
  Serial.print(humidityString);
  Serial.println("%");
}
//=====
//  getPressure - Subroutine to get and print pressure
//=====
void getPressure()
{
  pressure = bme.readPressure();
  pressure = bme.seaLevelForAltitude(ALTITUDE, pressure);
  pressure = pressure / 3386.39; // Convert hPa to in/Hg
  String pressureString = String(pressure, 2);
  Serial.print("Pressure: ");
  Serial.print(pressureString);
  Serial.println("in");
}
//=====
//  getAltitude - Subroutine to get and print temperature
//=====
void getAltitude()
{
  altitude = bme.readAltitude(SEA_LEVEL_PRESSURE);
  altitude = altitude * 3.28084; // Convert meters to feet

```

```
String altitudeString = String(altitude, 0);
Serial.print("Altitude: ");
Serial.print(altitudeString);
Serial.println("ft");
}
```

BEFORE THEY ARE SHIPPED, THESE MODULES ARE:

- Sample inspected and tested for basic operation per incoming shipment

TECHNICAL SPECIFICATIONS

Operating Ratings

Vcc Range		3.3V or 5V
Humidity	Range	0 – 100%
	Accuracy	±3%
Pressure	Range	300 – 1100hPa
	Accuracy	±1hPa
Temperature	Range	-40 – 80°C
	Accuracy	±1.0C
Dimensions	L x W (PCB)	13 x 10.5mm (0.52 x 0.41")
Country of Origin		China
Datasheet	Bosch	BME280

FURTHER READING

Wikipedia: https://en.wikipedia.org/wiki/Atmospheric_pressure

Air Pressure Calculator: <https://www.mide.com/pages/air-pressure-at-altitude-calculator>