

Raspberry Pi BME Sensors Tutorial

Contents

Raspberry Pi BME Sensors Tutorial	1
BME280 and BME680 Sensor Documentation	1
BME280 and BME680	1
RealVNC Viewer	2
Tutorial 0: Install BME280 and BME680 Libraries.....	2
Tutorial 1: Hello World BME280 Read.....	2
Tutorial 1: Hello World BME680 Read.....	5
Tutorial 2: BME280 Read 2	7
Tutorial 2: BME680 Read 2	9
Tutorial 3: BME280 Read 3	11
Tutorial 3: BME680 Read 3	12
Air Quality	15
bme68x_test.py	15

BME280 and BME680 Sensor Documentation

Pimoroni BME280 sensor documentation: <https://github.com/pimoroni/bme280-python>

Pimoroni BME680 sensor documentation: <https://github.com/pimoroni/bme680-python>

BME280 and BME680

This tutorial is for the Grove - Temperature, Humidity and Pressure BME280 sensor (BME280) and the Grove - Temperature, Humidity, Pressure and Gas Sensor (BME680).

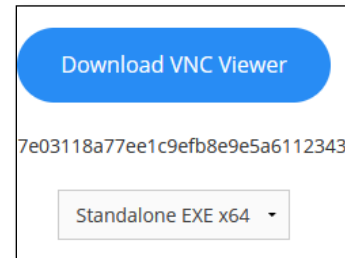
<https://www.seeedstudio.com/Grove-BME280-Environmental-Sensor-Temperature-Humidity-Barometer.html>

<https://www.seeedstudio.com/Grove-Temperature-Humidity-Pressure-and-Gas-Sensor-BME68-p-3109.html>

RealVNC Viewer

RealVNC viewer allows us to remotely control the Raspberry Pi in headless mode.

1. Go to
<https://www.realvnc.com/en/connect/download/viewer/>
2. Download the VNC Viewer Standalone EXE anywhere you want to run the program from. You don't have to install it.
3. Double Click **VNC Viewer**.
4. Type in the IP address of your robot → Click **Connect**.



Tutorial 0: Install BME280 and BME680 Libraries

1. Shutdown the Pi. (Do not connect sensors when the Pi has power.)
2. Plug the BME280 or BME680 sensor into an I2C port.
3. Mount the sensor on a sensor mount.
4. Powerup the Pi.
5. Open a terminal.
6. Install BME280 drivers – Docs: <https://pypi.org/project/pimoroni-bme280>
 - a. **sudo pip3 install pimoroni-bme280**
7. Install BME680 drivers - Docs: <https://pypi.org/project/bme680>
 - a. **sudo pip3 install bme680**

Tutorial 1: Hello World BME280 Read

1. Shutdown the Pi. (Do not connect sensors when the Pi has power.)
2. Plug the BME280 sensor into an I2C port.
3. Mount the sensor on a sensor mount.
4. Powerup the Pi.
5. Open a terminal.

6. Type: **geany bme280_read_1.py**
7. This will open the geany code editor and create a Python file.
8. Copy and paste the following code. This code will not change. It imports the sensor library and initializes the sensor.

```
#!/usr/bin/env python3
"""
    Name: bme280_read_1.py
    Purpose: Use Pimoroni library to read
    temperature, pressure, and humidity from Bosch bme280 sensor
    !Connect to I2C bus
    Press Ctrl+C to exit
"""
from time import sleep
try:
    from smbus2 import SMBus
except ImportError:
    from smbus import SMBus

# sudo pip3 install pimoroni-bme280
from bme280 import BME280

# Initialize the BME280 sensor
bus = SMBus(1)
sensor = BME280(i2c_dev=bus)
```

9. Enter the following code after the copied and pasted section.

```

1  #!/usr/bin/env python3
2  """
3      Name: bme280_read_1.py
4      Purpose: Use Pimoroni library to read
5      temperature, pressure, and humidity from Bosch bme280 sensor
6      !Connect to I2C bus
7      Press Ctrl+C to exit
8  """
9  from time import sleep
10 try:
11     from smbus2 import SMBus
12 except ImportError:
13     from smbus import SMBus
14
15 # sudo pip3 install pimoroni-bme280
16 from bme280 import BME280
17
18 # Initialize the BME280 sensor
19 bus = SMBus(1)
20 sensor = BME280(i2c_dev=bus)
21
22 print("BME280 Read temperature, pressure, and humidity")
23 print("Ctrl+C to exit!")
24
25 while True:
26     # Temperature in celsius
27     print(sensor.get_temperature())
28
29     # Barometric ressure in hPa (hectopascal)
30     print(sensor.get_pressure())
31
32     # Relative humidity in %
33     print(sensor.get_humidity())
34
35     sleep(1)

```

1. Open a terminal
2. **python3 bme280_read_1.py**

Example run:

```
pi@raspberrypi3:~/Code $ python3 bme280_simple_1.py
BME280 Read temperature, pressure, and humidity
Ctrl+C to exit!
23.537171707028754
618.5009584899727
65.11997946527883
64.7010000730000104
```

Tutorial 1: Hello World BME680 Read

1. Shutdown the Pi. (Do not connect sensors when the Pi has power.)
2. Plug the BME680 sensor into an I2C port.
3. Mount the sensor on a sensor mount.
4. Powerup the Pi.
5. Open a terminal.
6. Type: **geany bme680_simple_1.py**
7. This will open the geany code editor and create a Python file.
8. Copy and paste the following code. This code will not change. It imports the sensor library and initializes the sensor.

```
#!/usr/bin/env python3
"""
    Filename: bme680_read_1.py
    Purpose: Use Pimoroni library to read
    temperature, pressure, and humidity from Bosch bme280 sensor
    !Connect to I2C bus
    Press Ctrl+C to exit
"""
from time import sleep

# sudo pip3 install bme680
import bme680

# Initialize sensor object, make connection to sensor over I2C
sensor = bme680.BME680(bme680.I2C_ADDR_PRIMARY)
```

9. Enter the following code starting after the copied and pasted section.

```

1  #!/usr/bin/env python3
2  """
3      Filename: bme680_read_1.py
4      Purpose: Use Pimoroni library to read
5      temperature, pressure, and humidity from Bosch bme280 senso
6      !Connect to I2C bus
7      Press Ctrl+C to exit
8  """
9  from time import sleep
10
11  # sudo pip3 install bme680
12  import bme680
13
14  # Initialize sensor object, make connection to sensor over I2C
15  sensor = bme680.BME680(bme680.I2C_ADDR_PRIMARY)
16
17  print(" BME680 Read temperature, pressure, and humidity")
18  print(" Ctrl+C to exit!")
19
20  while True:
21      # Read the sensor
22      sensor.get_sensor_data()
23
24      # Temperature in celsius
25      print(sensor.data.temperature)
26
27      # Barometric ressure in hPa (hectopascal)
28      print(sensor.data.pressure)
29
30      # Relative humidity in %
31      print(sensor.data.humidity)
32
33      sleep(1)

```

1. Open a terminal
2. **python3 bme280_read_1.py**

Example run:

```
pi@raspberrypi3:~/Code $ python3 bme680_simple_1.py
BME680 Read temperature, pressure, and humidity
Ctrl+C to exit!
22.28
890.84
42.284
22.28
890.84
42.284
22.28
```

Tutorial 2: BME280 Read 2

1. Shutdown the Pi. (Do not connect sensors when the Pi has power.)
2. Plug the BME280 sensor into an I2C port.
3. Mount the sensor on a sensor mount.
4. Powerup the Pi.
5. Open a terminal.
6. Type: **geany bme280_read_2.py**
7. This will open the geany code editor and create a Python file.
8. Enter the following code.

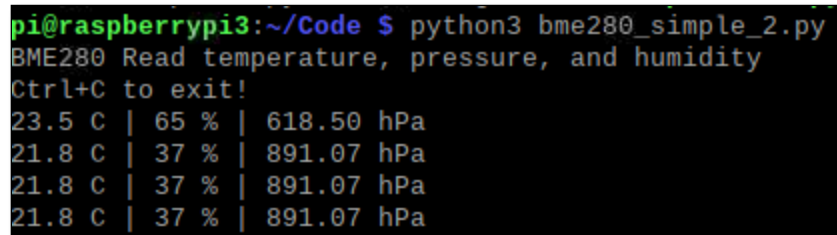
```

1  #!/usr/bin/env python3
2  """
3      Name: bme280_read_2.py
4      Purpose: Use Pimoroni library to read
5      temperature, pressure, and humidity from Bosch bme280 sensor
6      !Connect to I2C bus
7      Press Ctrl+C to exit
8  """
9  from time import sleep
10 try:
11     from smbus2 import SMBus
12 except ImportError:
13     from smbus import SMBus
14
15 # sudo pip3 install pimoroni-bme280
16 from bme280 import BME280
17
18 # Initialize the BME280 sensor
19 bus = SMBus(1)
20 sensor = BME280(i2c_dev=bus)
21
22 print("BME280 Read temperature, pressure, and humidity")
23 print("Ctrl+C to exit!")
24
25
26 def main():
27     try:
28         while True:
29             # Temperature in celsius
30             temp_c = sensor.get_temperature()
31
32             # Relative humidity in %
33             humidity = sensor.get_humidity()
34
35             # Barometric pressure in hPa (hectopascal)
36             pressure_hpa = sensor.get_pressure()
37
38             print(f"{temp_c:.1f} C | {humidity:.0f} % | {pressure_hpa:.2f} hPa")
39
40             sleep(1)
41
42     except KeyboardInterrupt:
43         print("Bye!")
44         exit(0)
45
46
47 # If a standalone program, call the main function
48 # Else, use as a module
49 if __name__ == '__main__':
50     main()
51

```


1. Open a terminal
2. **python3 bme280_read_2.py**

Example run:

A terminal window on a Raspberry Pi showing the execution of a Python script. The prompt is 'pi@raspberrypi3:~/Code \$'. The command 'python3 bme280_simple_2.py' has been entered. The output shows the program title 'BME280 Read temperature, pressure, and humidity' and a prompt 'Ctrl+C to exit!'. Below this, four lines of sensor data are displayed in a table-like format: temperature in Celsius, humidity in percent, and pressure in hPa.

```
pi@raspberrypi3:~/Code $ python3 bme280_simple_2.py
BME280 Read temperature, pressure, and humidity
Ctrl+C to exit!
23.5 C | 65 % | 618.50 hPa
21.8 C | 37 % | 891.07 hPa
21.8 C | 37 % | 891.07 hPa
21.8 C | 37 % | 891.07 hPa
```

Tutorial 2: BME680 Read 2

1. Shutdown the Pi. (Do not connect sensors when the Pi has power.)
2. Plug the BME280 sensor into an I2C port.
3. Mount the sensor on a sensor mount.
4. Powerup the Pi.
5. Open a terminal.
6. This command will make a copy of the last version of this program
 - a. **cp bme280_read_1.py bme280_read_2.py**
7. This will open the file in the geany code editor.
 - a. **geany bme680_read_2.py**
8. Modify the following code.

```

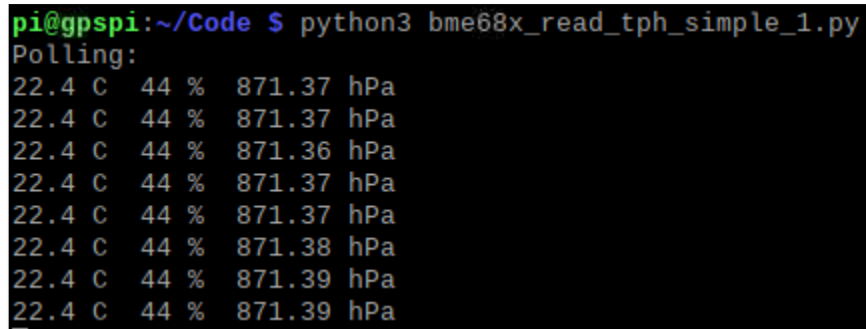
1  #!/usr/bin/env python3
2  """
3      Filename: bme680_read_2.py
4      Purpose: Use Pimoroni library to read
5              temperature, pressure, and humidity from Bosch bme680 sensor
6              !Connect to I2C bus
7              Press Ctrl+C to exit
8  """
9  from sys import exit
10 from time import sleep
11
12 # sudo pip3 install bme680
13 import bme680
14
15 # Initialize sensor object, make connection to sensor over I2C
16 sensor = bme680.BME680(bme680.I2C_ADDR_PRIMARY)
17
18 print(" BME680 Read temperature, pressure, and humidity")
19 print(" Press CTRL+C to Exit")
20
21 try:
22     while True:
23         # Can the sensor data can be retrieved successfully?
24         if sensor.get_sensor_data():
25             # If sensor data retrieval is successful,
26             # retrieve and display the data
27
28             # Sensor output in celsius
29             temp_c = sensor.data.temperature
30
31             # Relative humidity in %
32             humidity = sensor.data.humidity
33
34             # Sensor output in hectoPascals (hPa), also called millibars
35             pressure_hpa = sensor.data.pressure
36
37             print(f" {temp_c:.1f} C | {humidity:.0f} % | {pressure_hpa:.2f} hPa")
38
39             sleep(1)
40
41 except KeyboardInterrupt:
42     print("Bye!")
43     exit(0)

```

1. Open a terminal

2. `python3 bme680_read_2.py`

Example run:



```
pi@gpspi:~/Code $ python3 bme68x_read_tph_simple_1.py
Polling:
22.4 C 44 % 871.37 hPa
22.4 C 44 % 871.37 hPa
22.4 C 44 % 871.36 hPa
22.4 C 44 % 871.37 hPa
22.4 C 44 % 871.37 hPa
22.4 C 44 % 871.38 hPa
22.4 C 44 % 871.39 hPa
22.4 C 44 % 871.39 hPa
```

Tutorial 3: BME280 Read 3

Barometric pressure compensation for altitude:

https://www.engineeringtoolbox.com/barometers-elevation-compensation-d_1812.html

1. Shutdown the Pi. (Do not connect sensors when the Pi has power.)
2. Plug the BME280 sensor into an I2C port.
3. Mount the sensor on a sensor mount.
4. Powerup the Pi.
5. Open a terminal.
6. This command will make a copy of the last version of this program
 - a. **`cp bme280_read_2.py bme280_read_3.py`**
7. This will open the file in the geany code editor.
 - a. **`geany bme280_read_3.py`**
8. Modify the following code.

```

21  try:
22      while True:
23          # Can the sensor data can be retrieved successfully?
24          if sensor.get_sensor_data():
25              # If sensor data retrieval is successful,
26              # retrieve and display the data
27
28              # Sensor output in celsius
29              temp_c = sensor.data.temperature
30              # Convert celsius to fahrenheit
31              temp_f = ((temp_c * 9.0) / 5.0) + 32
32
33              # Relative humidity in %
34              humidity = sensor.data.humidity
35
36              # Sensor output in hectoPascals (hPa), also called millibars
37              pressure_pascals = sensor.data.pressure
38              # Convert hPa hectopascals to inHg Inches of Mercury
39              pressure_inhg = pressure_pascals / 33.863886666667
40              # Compensate for 3960' altitude 4.04
41              # Scottsbluff, NE, Heilig Field, 4.04
42              pressure_inhg = pressure_inhg + 4.04
43
44              print(f" {temp_f:.1f} °F | {humidity:.1f}% | {pressure_inhg:.2f} inHg")
45
46              sleep(5)
47
48
49  except KeyboardInterrupt:
50      print("Bye!")
51      exit(0)

```

1. Open a terminal
2. **python3 bme280_read_3.py**

Tutorial 3: BME680 Read 3

Barometric pressure compensation for altitude:

https://www.engineeringtoolbox.com/barometers-elevation-compensation-d_1812.html

1. Shutdown the Pi. (Do not connect sensors when the Pi has power.)
2. Plug the BME680 sensor into an I2C port.

3. Mount the sensor on a sensor mount.
4. Powerup the Pi.
5. Open a terminal.
6. This command will make a copy of the last version of this program
 - a. **cp bme680_read_2.py bme680_read_3.py**
7. This will open the file in the geany code editor.
 - a. **geany bme680_read_3.py**
8. Modify the following code.

```

21     try:
22         while True:
23             # Can the sensor data can be retrieved successfully?
24             if sensor.get_sensor_data():
25                 # If sensor data retrieval is successful,
26                 # retrieve and display the data
27
28                 # Sensor output in celsius
29                 temp_c = sensor.data.temperature
30                 # Convert celsius to fahrenheit
31                 temp_f = ((temp_c * 9.0) / 5.0) + 32
32
33                 # Relative humidity in %
34                 humidity = sensor.data.humidity
35
36                 # Sensor output in hectoPascals (hPa), also called millibars
37                 pressure_pascals = sensor.data.pressure
38                 # Convert hPa hectopascals to inHg Inches of Mercury
39                 pressure_inhg = pressure_pascals / 33.863886666667
40                 # Compensate for 3960' altitude 4.04
41                 # Scottsbluff, NE, Heilig Field, 4.04
42                 pressure_inhg = pressure_inhg + 4.04
43
44                 print(f" {temp_f:.1f} °F | {humidity:.1f}% | {pressure_inhg:.2f} inHg")
45
46                 sleep(5)
47
48
49     except KeyboardInterrupt:
50         print("Bye!")
51         exit(0)
52

```

1. Open a terminal
2. **python3 bme680_read_3.py**

Example run:

```

Press CTRL+C to Exit
Polling:
69.2 °F - 42.4% - 29.83 inHg - 873.37 KPa
69.2 °F - 42.4% - 29.83 inHg - 873.38 KPa
69.2 °F - 42.5% - 29.83 inHg - 873.38 KPa
69.2 °F - 42.5% - 29.83 inHg - 873.39 KPa

```

Air Quality

The sensor needs to be calibrated for 5 days when it is first used. The sensor value is not accurate before/during calibration. Wait about 2 mins after module powered on for gas heater enter a steady state. The gas value at this time is correct.

The IAQ value corresponding air quality:

0-50: Good

51-100: Moderate

101-150: Unhealthy for Sensitive Groups

151-200: Unhealthy

201-300: Very Unhealthy

301-: Hazardous

bme68x_test.py

This program will read the BME680 every 5 seconds and display to the console.