# Adafruit BME680 Library Documentation

*Release 1.0*

**ladyada**

**Jul 31, 2020**

# Contents

CircuitPython driver for BME680 sensor over I2C

# CHAPTER 1

# Dependencies

This driver depends on:

- Adafruit CircuitPython
- Bus Device

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

# CHAPTER 2

# Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI. To install for current user:

```
pip3 install adafruit-circuitpython-bme680
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-bme680
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-bme680
```

# CHAPTER 3

## Usage Example

```
from busio import I2C
import adafruit_bme680
import time
import board

# Create library object using our Bus I2C port
i2c = I2C(board.SCL, board.SDA)
bme680 = adafruit_bme680.Adafruit_BME680_I2C(i2c)

# change this to match the location's pressure (hPa) at sea level
bme680.sea_level_pressure = 1013.25

while True:
    print("\nTemperature: %0.1f C" % bme680.temperature)
    print("Gas: %d ohm" % bme680.gas)
    print("Humidity: %0.1f %%" % bme680.humidity)
    print("Pressure: %0.3f hPa" % bme680.pressure)
    print("Altitude = %0.2f meters" % bme680.altitude)

    time.sleep(2)
```

# CHAPTER 4

# Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

# CHAPTER 5

# Documentation

For information on building library documentation, please check out this guide.

CHAPTER 6

# Table of Contents

## 6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/bme680_simpletest.py

```
import time
import board
from busio import I2C
import adafruit_bme680

# Create library object using our Bus I2C port
i2c = I2C(board.SCL, board.SDA)
bme680 = adafruit_bme680.Adafruit_BME680_I2C(i2c, debug=False)

# change this to match the location's pressure (hPa) at sea level
bme680.sea_level_pressure = 1013.25

# You will usually have to add an offset to account for the temperature of
# the sensor. This is usually around 5 degrees but varies by use. Use a
# separate temperature sensor to calibrate this one.
temperature_offset = -5

while True:
    print("\nTemperature: %0.1f C" % bme680.temperature + temperature_offset)
    print("Gas: %d ohm" % bme680.gas)
    print("Humidity: %0.1f %%" % bme680.humidity)
    print("Pressure: %0.3f hPa" % bme680.pressure)
    print("Altitude = %0.2f meters" % bme680.altitude)

    time.sleep(1)
```

## 6.2 adafruit_bme680

CircuitPython library for BME680 temperature, pressure and humidity sensor.

- Author(s): Limor Fried

### 6.2.1 Implementation Notes

**Hardware:**

- Adafruit BME680 Temp, Humidity, Pressure and Gas Sensor

**Software and Dependencies:**

- Adafruit CircuitPython firmware for the supported boards: https://github.com/adafruit/circuitpython/releases
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

**class** adafruit_bme680.**Adafruit_BME680**(*, *refresh_rate=10*)

Driver from BME680 air quality sensor

**Parameters** **refresh_rate** (*int*) – Maximum number of readings per second. Faster property reads will be from the previous reading.

**altitude**

The altitude based on current `pressure` vs the sea level pressure (`sea_level_pressure`) - which you must enter ahead of time)

**filter_size**

The filter size for the built in IIR filter

**gas**

The gas resistance in ohms

**humidity**

The relative humidity in RH %

**humidity_oversample**

The oversampling for humidity sensor

**pressure**

The barometric pressure in hectoPascals

**pressure_oversample**

The oversampling for pressure sensor

**sea_level_pressure = None**

Pressure in hectoPascals at sea level. Used to calibrate `altitude`.

**temperature**

The compensated temperature in degrees celsius.

**temperature_oversample**

The oversampling for temperature sensor

**class** adafruit_bme680.**Adafruit_BME680_I2C**(*i2c*, *address=119*, *debug=False*, *, *refresh_rate=10*)

Driver for I2C connected BME680.

**Parameters**

- **address** (*int*) – I2C device address

- **debug** (*bool*) – Print debug statements when True.
- **refresh_rate** (*int*) – Maximum number of readings per second. Faster property reads will be from the previous reading.

**class** adafruit_bme680.**Adafruit_BME680_SPI**(*spi*, *cs*, *baudrate=100000*, *debug=False*, ***, *refresh_rate=10*)

Driver for SPI connected BME680.

**Parameters**

- **spi** (*busio.SPI*) – SPI device
- **cs** (*digitalio.DigitalInOut*) – Chip Select
- **debug** (*bool*) – Print debug statements when True.
- **baudrate** (*int*) – Clock rate, default is 100000
- **refresh_rate** (*int*) – Maximum number of readings per second. Faster property reads will be from the previous reading.

CHAPTER 7

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## a

# Index