Adafruit BME680 Library Documentation

Release 1.0

ladyada

Apr 01, 2018

Contents

1 Dependencies **3**

2 Usage Example **5**

3 Contributing **7**

4 Building locally **9**
4.1 Sphinx documentation 9

5 Table of Contents **11**
5.1 Simple test 11
5.2 `adafruit_bme680` - Adafruit BME680 - Temperature, Humidity, Pressure & Gas Sensor 11

6 Indices and tables **13**

Python Module Index **15**

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

Usage Example

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

gc.collect()
print("Free mem:",gc.mem_free())

# 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 3

Contributing

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

CHAPTER 4

Building locally

To build this library locally you'll need to install the circuitpython-build-tools package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-bme680 --library_
↪location .
```

4.1 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the index.html in your browser to view them. It will also (due to -W) error out on any warning like Travis will. This is a good way to locally verify it will pass.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 5.1: examples/bme680_simpletest.py

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

# 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

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(1)
```

5.2 `adafruit_bme680` - Adafruit BME680 - Temperature, Humidity, Pressure & Gas Sensor

CircuitPython driver from BME680 air quality sensor

- Author(s): ladyada

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.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_bme680, 11

Index

A

Adafruit_BME680 (class in adafruit_bme680), 12
adafruit_bme680 (module), 11
Adafruit_BME680_I2C (class in adafruit_bme680), 12
altitude (adafruit_bme680.Adafruit_BME680 attribute), 12

F

filter_size (adafruit_bme680.Adafruit_BME680 attribute), 12

G

gas (adafruit_bme680.Adafruit_BME680 attribute), 12

H

humidity (adafruit_bme680.Adafruit_BME680 attribute), 12
humidity_oversample (adafruit_bme680.Adafruit_BME680 attribute), 12

P

pressure (adafruit_bme680.Adafruit_BME680 attribute), 12
pressure_oversample (adafruit_bme680.Adafruit_BME680 attribute), 12

S

sea_level_pressure (adafruit_bme680.Adafruit_BME680 attribute), 12

T

temperature (adafruit_bme680.Adafruit_BME680 attribute), 12
temperature_oversample (adafruit_bme680.Adafruit_BME680 attribute), 12