

---

# **AdafruitBMP3XX Library Documentation**

***Release 1.0***

**Carter Nelson**

**Apr 26, 2021**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
1.1	Installing from PyPI . . . . .	3
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Documentation</b>	<b>9</b>
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_bmp3xx . . . . .	12
5.2.1	Implementation Notes . . . . .	12
<b>6</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



CircuitPython driver from BMP3XX Temperature and Barometric Pressure sensor.



# 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](#).

### 1.1 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-bmp3xx
```

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

```
sudo pip3 install adafruit-circuitpython-bmp3xx
```

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-bmp3xx
```





## CHAPTER 2

---

### Usage Example

---

See usage examples in the examples folder.



## CHAPTER 3

---

### Contributing

---

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.



## CHAPTER 4

---

### Documentation

---

For information on building library documentation, please check out [this guide](#).



## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/bmp3xx\_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6  import adafruit_bmp3xx
7
8  # I2C setup
9  i2c = board.I2C() # uses board.SCL and board.SDA
10 bmp = adafruit_bmp3xx.BMP3XX_I2C(i2c)
11
12 # SPI setup
13 # from digitalio import DigitalInOut, Direction
14 # spi = board.SPI()
15 # cs = DigitalInOut(board.D5)
16 # bmp = adafruit_bmp3xx.BMP3XX_SPI(spi, cs)
17
18 bmp.pressure_oversampling = 8
19 bmp.temperature_oversampling = 2
20
21 while True:
22     print(
23         "Pressure: {:6.4f} Temperature: {:5.2f}".format(bmp.pressure, bmp.
24         ↪temperature)
25     )
26     time.sleep(1)
```

## 5.2 adafruit\_bmp3xx

CircuitPython driver from BMP388 Temperature and Barometric Pressure sensor.

- Author(s): Carter Nelson

### 5.2.1 Implementation Notes

#### Hardware:

- [Adafruit BMP388](#)

#### 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](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)

**class** adafruit\_bmp3xx.BMP3XX

Base class for BMP3XX sensor.

**altitude**

The altitude in meters based on the currently set sea level pressure.

**filter\_coefficient**

The IIR filter coefficient.

**pressure**

The pressure in hPa.

**pressure\_oversampling**

The pressure oversampling setting.

**reset ()**

Perform a power on reset. All user configuration settings are overwritten with their default state.

**sea\_level\_pressure = None**

Sea level pressure in hPa.

**temperature**

The temperature in degrees Celsius

**temperature\_oversampling**

The temperature oversampling setting.

**class** adafruit\_bmp3xx.BMP3XX\_I2C (i2c, address=119)

Driver for I2C connected BMP3XX.

#### Parameters

- **i2c** (*I2C*) – The I2C bus the BMP388 is connected to.
- **address** (*int*) – I2C device address. Defaults to 0x77. but another address can be passed in as an argument

#### Quickstart: Importing and using the BMP388

Here is an example of using the `BMP3XX_I2C` class. First you will need to import the libraries to use the sensor

```
import board
import adafruit_bmp3xx
```



Once this is done you can define your `board.I2C` object and define your sensor object

```
i2c = board.I2C()    # uses board.SCL and board.SDA
bmp = adafruit_bmp3xx.BMP3XX_I2C(i2c)
```

Now you have access to the temperature and pressure attributes

```
temperature = bmp.temperature
pressure = bmp.pressure
```

**class** `adafruit_bmp3xx.BMP3XX_SPI` (*spi*, *cs*)  
Driver for SPI connected BMP3XX.

#### Parameters

- **spi** (*SPI*) – SPI device
- **cs** (*DigitalInOut*) – Chip Select

#### Quickstart: Importing and using the BMP388

Here is an example of using the `BMP3XX_SPI` class. First you will need to import the libraries to use the sensor

```
import board
import adafruit_bmp3xx
from digitalio import DigitalInOut, Direction
```

Once this is done you can define your `board.SPI` object and define your sensor object

```
spi = board.SPI()
cs = DigitalInOut(board.D5)
bmp = adafruit_bmp3xx.BMP3XX_SPI(spi, cs)
```

Now you have access to the temperature and pressure attributes

```
temperature = bmp.temperature
pressure = bmp.pressure
```



## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### a

adafruit\_bmp3xx, [11](#)



### A

`adafruit_bmp3xx` (*module*), [11](#)  
`altitude` (*adafruit\_bmp3xx.BMP3XX attribute*), [12](#)

### B

`BMP3XX` (*class in adafruit\_bmp3xx*), [12](#)  
`BMP3XX_I2C` (*class in adafruit\_bmp3xx*), [12](#)  
`BMP3XX_SPI` (*class in adafruit\_bmp3xx*), [13](#)

### F

`filter_coefficient` (*adafruit\_bmp3xx.BMP3XX attribute*), [12](#)

### P

`pressure` (*adafruit\_bmp3xx.BMP3XX attribute*), [12](#)  
`pressure_oversampling`  
(*adafruit\_bmp3xx.BMP3XX attribute*), [12](#)

### R

`reset()` (*adafruit\_bmp3xx.BMP3XX method*), [12](#)

### S

`sea_level_pressure` (*adafruit\_bmp3xx.BMP3XX attribute*), [12](#)

### T

`temperature` (*adafruit\_bmp3xx.BMP3XX attribute*),  
[12](#)  
`temperature_oversampling`  
(*adafruit\_bmp3xx.BMP3XX attribute*), [12](#)