

---

# **Adafruit BNO055 Library Documentation**

***Release 1.0***

**Radomir Dopieralski**

**Oct 18, 2019**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Notes</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Building locally</b>	<b>9</b>
4.1	Sphinx documentation . . . . .	9
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_bno055 - Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 . . .	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>







# CHAPTER 1

---

## Dependencies

---

This driver depends on the [Register](#) and [Bus Device](#) libraries. Please ensure they are also available on the CircuitPython filesystem. This is easily achieved by downloading [a library and driver bundle](#).





## CHAPTER 2

---

### Usage Notes

---

Of course, you must import the library to use it:

```
import adafruit_bno055
```

This driver takes an instantiated and active I2C object (from the `busio` or the `bitbangio` library) as an argument to its constructor. The way to create an I2C object depends on the board you are using. For boards with labeled SCL and SDA pins, you can:

```
from busio import I2C
from board import SDA, SCL

i2c = I2C(SCL, SDA)
```

Once you have the I2C object, you can create the sensor object:

```
sensor = adafruit_bno055.BNO055(i2c)
```

And then you can start reading the measurements:

```
print(sensor.temperature)
print(sensor.euler)
print(sensor.gravity)
```



## 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-bno055 --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.



## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/bno055\_simpletest.py

```
1 import time
2 import board
3 import busio
4 import adafruit_bno055
5
6 i2c = busio.I2C(board.SCL, board.SDA)
7 sensor = adafruit_bno055.BNO055(i2c)
8
9 while True:
10     print('Temperature: {} degrees C'.format(sensor.temperature))
11     print('Accelerometer (m/s^2): {}'.format(sensor.accelerometer))
12     print('Magnetometer (microteslas): {}'.format(sensor.magnetometer))
13     print('Gyroscope (rad/sec): {}'.format(sensor.gyro))
14     print('Euler angle: {}'.format(sensor.euler))
15     print('Quaternion: {}'.format(sensor.quaternion))
16     print('Linear acceleration (m/s^2): {}'.format(sensor.linear_acceleration))
17     print('Gravity (m/s^2): {}'.format(sensor.gravity))
18     print()
19
20     time.sleep(1)
```

## 5.2 adafruit\_bno055 - Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055

This is a CircuitPython driver for the Bosch BNO055 nine degree of freedom inertial measurement unit module with sensor fusion.

- Author(s): Radomir Dopieralski

**class** `adafruit_bno055.BNO055` (*i2c*, *address=40*)  
Driver for the BNO055 9DOF IMU sensor.

**acceleration**

Gives the raw accelerometer readings, in m/s.

**accelerometer**

Gives the raw accelerometer readings, in m/s.

**Warning:** This is deprecated. Use `acceleration` instead. It'll work with other drivers too.

**calibrated**

Boolean indicating calibration status.

**calibration\_status**

Tuple containing sys, gyro, accel, and mag calibration data.

**euler**

Gives the calculated orientation angles, in degrees.

**external\_crystal**

Switches the use of external crystal on or off.

**gravity**

Returns the gravity vector, without acceleration in m/s.

**gyro**

Gives the raw gyroscope reading in radians per second.

**gyroscope**

Gives the raw gyroscope reading in degrees per second.

**Warning:** This is deprecated. Use `gyro` instead. It'll work with other drivers too.

**linear\_acceleration**

Returns the linear acceleration, without gravity, in m/s.

**magnetic**

Gives the raw magnetometer readings in microteslas.

**magnetometer**

Gives the raw magnetometer readings in microteslas.

**Warning:** This is deprecated. Use `magnetic` instead. It'll work with other drivers too.

**mode**

Switch the mode of operation and return the previous mode.



Mode of operation defines which sensors are enabled and whether the measurements are absolute or relative:

Mode	Accel	Compass	Gyro	Absolute
CONFIG_MODE	•	•	•	•
ACCONLY_MODE	X	•	•	•
MAGONLY_MODE	•	X	•	•
GYRONLY_MODE	•	•	X	•
ACCMAG_MODE	X	X	•	•
ACCGYRO_MODE	X	•	X	•
MAGGYRO_MODE	•	X	X	•
AMG_MODE	X	X	X	•
IMUPLUS_MODE	X	•	X	•
COMPASS_MODE	X	X	•	X
M4G_MODE	X	X	•	•
NDOF_FMC_OFF_MODE	X	X	X	X
NDOF_MODE	X	X	X	X

The default mode is NDOF\_MODE.

#### **quaternion**

Gives the calculated orientation as a quaternion.

#### **temperature**

Measures the temperature of the chip in degrees Celsius.

#### **use\_external\_crystal**

Switches the use of external crystal on or off.



## CHAPTER 6

---

### Indices and tables

---

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



### a

adafruit\_bno055, [11](#)



## A

`acceleration` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`accelerometer` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`adafruit_bno055` (*module*), [11](#)

## B

`BNO055` (*class in adafruit\_bno055*), [12](#)

## C

`calibrated` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`calibration_status` (*adafruit\_bno055.BNO055 attribute*), [12](#)

## E

`euler` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`external_crystal` (*adafruit\_bno055.BNO055 attribute*), [12](#)

## G

`gravity` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`gyro` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`gyroscope` (*adafruit\_bno055.BNO055 attribute*), [12](#)

## L

`linear_acceleration` (*adafruit\_bno055.BNO055 attribute*), [12](#)

## M

`magnetic` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`magnetometer` (*adafruit\_bno055.BNO055 attribute*), [12](#)  
`mode` (*adafruit\_bno055.BNO055 attribute*), [12](#)

## Q

`quaternion` (*adafruit\_bno055.BNO055 attribute*), [13](#)

## T

`temperature` (*adafruit\_bno055.BNO055 attribute*), [13](#)

## U

`use_external_crystal`  
(*adafruit\_bno055.BNO055 attribute*), [13](#)