

---

# **Adafruit BNO055 Library Documentation**

***Release 1.0***

**Radomir Dopieralski**

**Jan 27, 2020**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Notes</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple test . . . . .	13
6.2	adafruit_bno055 - Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 . . .	14
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</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

---

### 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-bno055
```

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

```
sudo pip3 install adafruit-circuitpython-bno055
```

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-bno055
```



## CHAPTER 3

---

### 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 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](#).





## 6.1 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.acceleration))
12     print('Magnetometer (microteslas): {}'.format(sensor.magnetic))
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)
```

## 6.2 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. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

### **calibrated**

Boolean indicating calibration status.

### **calibration\_status**

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

### **euler**

Gives the calculated orientation angles, in degrees. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

### **external\_crystal**

Switches the use of external crystal on or off.

### **gravity**

Returns the gravity vector, without acceleration in m/s. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

### **gyro**

Gives the raw gyroscope reading in radians per second. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

### **linear\_acceleration**

Returns the linear acceleration, without gravity, in m/s. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

### **magnetic**

Gives the raw magnetometer readings in microteslas. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

### **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. If a sensor is disabled, it will return an empty tuple.

legend: x=on, -=off +-----+-----+-----+-----+ | Mode | Accel | Compass  
 | Gyro | Absolute | +=====+=====+=====+=====+  
 | CONFIG\_MODE | - | - | - | - | +-----+-----+-----+-----+ | AC-  
 ONLY\_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.

#### **offsets\_accelerometer**

Calibration offsets for the accelerometer

#### **offsets\_gyroscope**

Calibration offsets for the gyroscope

#### **offsets\_magnetometer**

Calibration offsets for the magnetometer

#### **quaternion**

Gives the calculated orientation as a quaternion. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

#### **radius\_accelerometer**

Radius for accelerometer (cm?)

#### **radius\_magnetometer**

Radius for magnetometer (cm?)

#### **temperature**

Measures the temperature of the chip in degrees Celsius.

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

Switches the use of external crystal on or off.



## CHAPTER 7

---

### Indices and tables

---

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



### **a**

adafruit\_bno055, [13](#)





## A

acceleration (*adafruit\_bno055.BNO055 attribute*), [14](#)

adafruit\_bno055 (*module*), [13](#)

## B

BNO055 (*class in adafruit\_bno055*), [14](#)

## C

calibrated (*adafruit\_bno055.BNO055 attribute*), [14](#)

calibration\_status (*adafruit\_bno055.BNO055 attribute*), [14](#)

## E

euler (*adafruit\_bno055.BNO055 attribute*), [14](#)

external\_crystal (*adafruit\_bno055.BNO055 attribute*), [14](#)

## G

gravity (*adafruit\_bno055.BNO055 attribute*), [14](#)

gyro (*adafruit\_bno055.BNO055 attribute*), [14](#)

## L

linear\_acceleration (*adafruit\_bno055.BNO055 attribute*), [14](#)

## M

magnetic (*adafruit\_bno055.BNO055 attribute*), [14](#)

mode (*adafruit\_bno055.BNO055 attribute*), [14](#)

## O

offsets\_accelerometer (*adafruit\_bno055.BNO055 attribute*), [15](#)

offsets\_gyroscope (*adafruit\_bno055.BNO055 attribute*), [15](#)

offsets\_magnetometer (*adafruit\_bno055.BNO055 attribute*), [15](#)

## Q

quaternion (*adafruit\_bno055.BNO055 attribute*), [15](#)

## R

radius\_accelerometer (*adafruit\_bno055.BNO055 attribute*), [15](#)

radius\_magnetometer (*adafruit\_bno055.BNO055 attribute*), [15](#)

## T

temperature (*adafruit\_bno055.BNO055 attribute*), [15](#)

## U

use\_external\_crystal (*adafruit\_bno055.BNO055 attribute*), [15](#)