
Adafruit BNO055 Library Documentation

Release 1.0

Radomir Dopieralski

Apr 14, 2021

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Notes	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_bno055 - Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 . . .	14
7	Indices and tables	19
	Python Module Index	21
	Index	23

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(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  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6  import busio
7  import adafruit_bno055
8
9  # Use these lines for I2C
10 i2c = busio.I2C(board.SCL, board.SDA)
11 sensor = adafruit_bno055.BNO055_I2C(i2c)
12
13 # User these lines for UART
14 # uart = busio.UART(board.TX, board.RX)
15 # sensor = adafruit_bno055.BNO055_UART(uart)
16
17 last_val = 0xFFFF
18
19
20 def temperature():
21     global last_val # pylint: disable=global-statement
22     result = sensor.temperature
23     if abs(result - last_val) == 128:
24         result = sensor.temperature
25         if abs(result - last_val) == 128:
26             return 0b00111111 & result
27     last_val = result
```

(continues on next page)

(continued from previous page)

```

28     return result
29
30
31 while True:
32     print("Temperature: {} degrees C".format(sensor.temperature))
33     """
34     print(
35         "Temperature: {} degrees C".format(temperature())
36     ) # Uncomment if using a Raspberry Pi
37     """
38     print("Accelerometer (m/s^2): {}".format(sensor.acceleration))
39     print("Magnetometer (microteslas): {}".format(sensor.magnetic))
40     print("Gyroscope (rad/sec): {}".format(sensor.gyro))
41     print("Euler angle: {}".format(sensor.euler))
42     print("Quaternion: {}".format(sensor.quaternion))
43     print("Linear acceleration (m/s^2): {}".format(sensor.linear_acceleration))
44     print("Gravity (m/s^2): {}".format(sensor.gravity))
45     print()
46
47     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

Base class for the BNO055 9DOF IMU sensor.

accel_bandwidth

Switch the accelerometer bandwidth and return the new bandwidth. Default value: 62.5 Hz See table 3-8 in the datasheet.

accel_mode

Switch the accelerometer mode and return the new mode. Default value: Normal See table 3-8 in the datasheet.

accel_range

Switch the accelerometer range and return the new range. Default value: +/- 4g See table 3-8 in the datasheet.

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.

axis_remap

Return a tuple with the axis remap register values.

This will return 6 values with the following meaning:

- **X axis remap** (a value of **AXIS_REMAP_X**, **AXIS_REMAP_Y**, or **AXIS_REMAP_Z**. which indicates that the physical X axis of the chip is remapped to a different axis)
- **Y axis remap** (see above)

- Z axis remap (see above)
- **X axis sign** (a value of `AXIS_REMAP_POSITIVE` or `AXIS_REMAP_NEGATIVE` which indicates if the X axis values should be positive/ normal or negative/inverted. The default is positive.)
- Y axis sign (see above)
- Z axis sign (see above)

Note that the default value, per the datasheet, is NOT P0, but rather P1 ()

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.

gyro_bandwidth

Switch the gyroscope bandwidth and return the new bandwidth. Default value: 32 Hz See table 3-9 in the datasheet.

gyro_mode

Switch the gyroscope mode and return the new mode. Default value: Normal See table 3-9 in the datasheet.

gyro_range

Switch the gyroscope range and return the new range. Default value: 2000 dps See table 3-9 in the datasheet.

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.

magnet_mode

Switch the magnetometer power mode and return the new mode. Default value: Forced See table 3-10 in the datasheet.

magnet_operation_mode

Switch the magnetometer operation mode and return the new mode. Default value: Regular See table 3-10 in the datasheet.

magnet_rate

Switch the magnetometer data output rate and return the new rate. Default value: 20Hz See table 3-10 in the datasheet.

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

legend: x=on, -=off (see Table 3-3 in datasheet)

Mode	Accel	Compass (Mag)	Gyro	Fusion solute	Ab- solute	Fusion Rela- tive
CONFIG_MODE	.	.	.	.	.	.
ACCONLY_MODE	.	.	.	.	.	.
MAGONLY_MODE	.	X	.	.	.	.
GYRONLY_MODE	.	.	X	.	.	.
ACCMAG_MODE	.	X	.	.	.	.
ACCGYRO_MODE	.	.	X	.	.	.
MAGGYRO_MODE	.	X	X	.	.	.
AMG_MODE	X	X	X	.	.	.
IMUPLUS_MODE	.	.	X	.	.	X
COMPASS_MODE	.	X	.	X	.	.
M4G_MODE	X	X	.	.	.	X
NDOF_FMC_OFF_MODE	.	X	X	X	.	.
NDOF_MODE	X	X	X	X	.	.

The default mode is NDOF_MODE.

You can set the mode using the line below:

```
sensor.mode = adafruit_bno055.ACCONLY_MODE
```

replacing ACCONLY_MODE with the mode you want to use

CONFIG_MODE

This mode is used to configure BNO, wherein all output data is reset to zero and sensor fusion is halted.

ACCONLY_MODE

In this mode, the BNO055 behaves like a stand-alone acceleration sensor. In this mode the other sensors (magnetometer, gyro) are suspended to lower the power consumption.

MAGONLY_MODE

In MAGONLY mode, the BNO055 behaves like a stand-alone magnetometer, with acceleration sensor and gyroscope being suspended.

GYRONLY_MODE

In GYROONLY mode, the BNO055 behaves like a stand-alone gyroscope, with acceleration sensor and magnetometer being suspended.

ACCMAG_MODE

Both accelerometer and magnetometer are switched on, the user can read the data from these two sensors.

ACCGYRO_MODE

Both accelerometer and gyroscope are switched on; the user can read the data from these two sensors.

MAGGYRO_MODE

Both magnetometer and gyroscope are switched on, the user can read the data from these two sensors.

AMG_MODE

All three sensors accelerometer, magnetometer and gyroscope are switched on.

IMUPLUS_MODE

In the IMU mode the relative orientation of the BNO055 in space is calculated from the accelerometer and gyroscope data. The calculation is fast (i.e. high output data rate).

COMPASS_MODE

The COMPASS mode is intended to measure the magnetic earth field and calculate the geographic direction.

M4G_MODE

The M4G mode is similar to the IMU mode, but instead of using the gyroscope signal to detect rotation, the changing orientation of the magnetometer in the magnetic field is used.

NDOF_FMC_OFF_MODE

This fusion mode is same as NDOF mode, but with the Fast Magnetometer Calibration turned 'OFF'.

NDOF_MODE

This is a fusion mode with 9 degrees of freedom where the fused absolute orientation data is calculated from accelerometer, gyroscope and 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.

temperature

Measures the temperature of the chip in degrees Celsius.

use_external_crystal

Switches the use of external crystal on or off.

class adafruit_bno055.**BNO055_I2C** (*i2c*, *address=40*)

Driver for the BNO055 9DOF IMU sensor via I2C.

offsets_accelerometer

Calibration offsets for the accelerometer

offsets_gyroscope

Calibration offsets for the gyroscope

offsets_magnetometer

Calibration offsets for the magnetometer

radius_accelerometer

Radius for accelerometer (cm?)

radius_magnetometer

Radius for magnetometer (cm?)

class adafruit_bno055.**BNO055_UART**(*uart*)

Driver for the BNO055 9DOF IMU sensor via UART.

offsets_accelerometer

Calibration offsets for the accelerometer

offsets_gyroscope

Calibration offsets for the gyroscope

offsets_magnetometer

Calibration offsets for the magnetometer

radius_accelerometer

Radius for accelerometer (cm?)

radius_magnetometer

Radius for magnetometer (cm?)

CHAPTER 7

Indices and tables

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

a

adafruit_bno055, [14](#)

A

[accel_bandwidth](#) (*adafruit_bno055.BNO055 attribute*), [14](#)
[accel_mode](#) (*adafruit_bno055.BNO055 attribute*), [14](#)
[accel_range](#) (*adafruit_bno055.BNO055 attribute*), [14](#)
[acceleration](#) (*adafruit_bno055.BNO055 attribute*), [14](#)
[adafruit_bno055](#) (*module*), [14](#)
[axis_remap](#) (*adafruit_bno055.BNO055 attribute*), [14](#)

B

[BNO055](#) (*class in adafruit_bno055*), [14](#)
[BNO055.ACCGYRO_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.ACCMAG_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.ACCONLY_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.AMG_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.COMPASS_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.CONFIG_MODE](#) (*in module adafruit_bno055*), [16](#)
[BNO055.GYRONLY_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.IMUPLUS_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.M4G_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.MAGGYRO_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.MAGONLY_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.NDOF_FMC_OFF_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055.NDOF_MODE](#) (*in module adafruit_bno055*), [17](#)
[BNO055_I2C](#) (*class in adafruit_bno055*), [17](#)
[BNO055_UART](#) (*class in adafruit_bno055*), [18](#)

C

[calibrated](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[calibration_status](#) (*adafruit_bno055.BNO055 attribute*), [15](#)

E

[euler](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[external_crystal](#) (*adafruit_bno055.BNO055 attribute*), [15](#)

G

[gravity](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[gyro](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[gyro_bandwidth](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[gyro_mode](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[gyro_range](#) (*adafruit_bno055.BNO055 attribute*), [15](#)

L

[linear_acceleration](#) (*adafruit_bno055.BNO055 attribute*), [15](#)

M

[magnet_mode](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[magnet_operation_mode](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[magnet_rate](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[magnetic](#) (*adafruit_bno055.BNO055 attribute*), [15](#)
[mode](#) (*adafruit_bno055.BNO055 attribute*), [15](#)

O

[offsets_accelerometer](#) (*adafruit_bno055.BNO055_I2C attribute*), [17](#)
[offsets_accelerometer](#) (*adafruit_bno055.BNO055_UART attribute*), [18](#)

offsets_gyroscope
 (*adafruit_bno055.BNO055_I2C* *attribute*),
 17
offsets_gyroscope
 (*adafruit_bno055.BNO055_UART* *attribute*),
 18
offsets_magnetometer
 (*adafruit_bno055.BNO055_I2C* *attribute*),
 18
offsets_magnetometer
 (*adafruit_bno055.BNO055_UART* *attribute*),
 18

Q

quaternion (*adafruit_bno055.BNO055 attribute*), 17

R

radius_accelerometer
 (*adafruit_bno055.BNO055_I2C* *attribute*),
 18
radius_accelerometer
 (*adafruit_bno055.BNO055_UART* *attribute*),
 18
radius_magnetometer
 (*adafruit_bno055.BNO055_I2C* *attribute*),
 18
radius_magnetometer
 (*adafruit_bno055.BNO055_UART* *attribute*),
 18

T

temperature (*adafruit_bno055.BNO055 attribute*),
 17

U

use_external_crystal
 (*adafruit_bno055.BNO055 attribute*), 17