Adafruit BNO055 Library Documentation

Release 1.0

Radomir Dopieralski

Contents

In	dex	21
Ру	thon Module Index	19
7	Indices and tables	17
6	1	13 13 14
5	Documentation	11
4	Contributing	9
3	Usage Notes	7
2	Installing from PyPI	5
1	Dependencies	3

Contents 1

2 Contents

				- 4
CH	A	PT	FF	₹ I

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

CHAPTER	5
---------	---

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/bno055_simpletest.py

```
import time
   import board
   import busio
   import adafruit_bno055
   # Use these lines for I2C
   i2c = busio.I2C(board.SCL, board.SDA)
   sensor = adafruit_bno055.BNO055_I2C(i2c)
   # User these lines for UART
   # uart = busio.UART(board.TX, board.RX)
11
   # sensor = adafruit_bno055.BNO055_UART(uart)
12
13
   while True:
14
       print("Temperature: {} degrees C".format(sensor.temperature))
15
       print("Accelerometer (m/s^2): {}".format(sensor.acceleration))
16
17
       print("Magnetometer (microteslas): {}".format(sensor.magnetic))
       print("Gyroscope (rad/sec): {}".format(sensor.gyro))
18
       print("Euler angle: {}".format(sensor.euler))
19
       print("Quaternion: {}".format(sensor.quaternion))
20
       print("Linear acceleration (m/s^2): {}".format(sensor.linear_acceleration))
21
       print("Gravity (m/s^2): {}".format(sensor.gravity))
22
       print()
24
       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.

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.

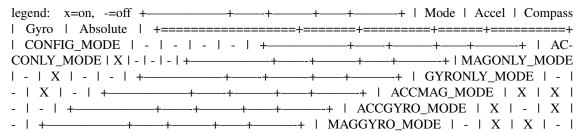
magnetic

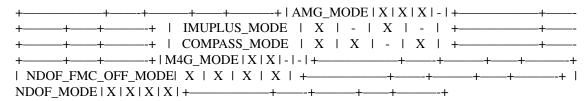
Gives the raw magnetometer readings in microteslas. Returns an empty tuple of length 3 when this property has been disabled by the current mode.

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.





The default mode is NDOF_MODE.

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?)

$\mathsf{CHAPTER}\ 7$

Indices and tables

- genindex
- modindex
- search

Python Module Index

а

adafruit_bno055,13

20 Python Module Index

Index

A	offsets_gyroscope
acceleration (adafruit_bno055.BNO055 attribute),	(adafruit_bno055.BNO055_I2C attribute), 15
adafruit_bno055 (module), 13	offsets_gyroscope (adafruit_bno055.BNO055_UART attribute),
В	15
BNO055 (class in adafruit_bno055), 14 BNO055_I2C (class in adafruit_bno055), 15 BNO055_UART (class in adafruit_bno055), 15	<pre>offsets_magnetometer (adafruit_bno055.BNO055_I2C attribute),</pre>
C	(adafruit_bno055.BNO055_UART attribute),
<pre>calibrated (adafruit_bno055.BNO055 attribute), 14 calibration_status (adafruit_bno055.BNO055</pre>	15 Q
E	quaternion (adafruit_bno055.BNO055 attribute), 15
euler (adafruit_bno055.BNO055 attribute), 14	R
external_crystal (adafruit_bno055.BNO055 attribute), 14	radius_accelerometer (adafruit_bno055.BNO055_I2C attribute), 15
G	radius_accelerometer
gravity (adafruit_bno055.BNO055 attribute), 14 gyro (adafruit_bno055.BNO055 attribute), 14	(adafruit_bno055.BNO055_UART attribute), 15
L	radius_magnetometer (adafruit_bno055.BNO055_I2C attribute), 15
linear_acceleration (adafruit_bno055.BNO055 attribute), 14	radius_magnetometer (adafruit_bno055.BNO055_UART attribute),
M	15
magnetic (adafruit_bno055.BNO055 attribute), 14 mode (adafruit_bno055.BNO055 attribute), 14	T temperature (adafruit_bno055.BNO055 attribute),
0	15
offsets_accelerometer (adafruit_bno055.BNO055_I2C attribute), 15 offsets_accelerometer (adafruit_bno055.BNO055_UART attribute), 15	U use_external_crystal (adafruit_bno055.BNO055 attribute), 15