
Adafruit L3GD20 Library Documentation

Release 1.0

Michael McWethy

Apr 29, 2021

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_l3gd20	14
6.2.1	Implementation Notes	14
7	Indices and tables	17
	Python Module Index	19
	Index	21

Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - L3GD20 Driver

CHAPTER 1

Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)
- [Register](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the [Adafruit 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-l3gd20
```

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

```
sudo pip3 install adafruit-circuitpython-l3gd20
```

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-l3gd20
```


CHAPTER 3

Usage Example

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

```
import adafruit_l3gd20
```

This driver takes an instantiated and active I2C object as an argument to its constructor.

```
import board  
  
i2c = board.I2C()
```

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

```
sensor = adafruit_l3gd20.L3GD20_I2C(i2c)
```

And then you can start reading the measurements:

```
print(sensor.gyro)
```


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

For I2C or SPI communications, ensure your device works with this simple test.

Listing 1: examples/l3gd20_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_l3gd20
7
8  # Hardware I2C setup:
9  I2C = board.I2C() # uses board.SCL and board.SDA
10 # Initializes L3GD20 object using default range, 250dps
11 SENSOR = adafruit_l3gd20.L3GD20_I2C(I2C)
12 # Initialize L3GD20 object using a custom range and output data rate (ODR).
13 # SENSOR = adafruit_l3gd20.L3GD20_I2C(
14 #     I2C, rng=adafruit_l3gd20.L3DS20_RANGE_500DPS, rate=adafruit_l3gd20.L3DS20_RATE_
15 #     ↪ 200HZ
16 # )
17
18 # Possible values for rng are:
19 # adafruit_l3gd20.L3DS20_Range_250DPS, 250 degrees per second. Default range
20 # adafruit_l3gd20.L3DS20_Range_500DPS, 500 degrees per second
21 # adafruit_l3gd20.L3DS20_Range_2000DPS, 2000 degrees per second
22
23 # Possible values for rate are:
24 # adafruit_l3gd20.L3DS20_RATE_100HZ, 100Hz data rate. Default data rate
25 # adafruit_l3gd20.L3DS20_RATE_200HZ, 200Hz data rate
26 # adafruit_l3gd20.L3DS20_RATE_400HZ, 400Hz data rate
27 # adafruit_l3gd20.L3DS20_RATE_800HZ, 800Hz data rate

```

(continues on next page)

(continued from previous page)

```

27
28 # Hardware SPI setup:
29 # import digitalio
30 # CS = digitalio.DigitalInOut(board.D5)
31 # SPIB = board.SPI()
32 # SENSOR = adafruit_l3gd20.L3GD20_SPI(SPIB, CS)
33 # SENSOR = adafruit_l3gd20.L3GD20_I2C(
34 #     SPIB,
35 #     CS,
36 #     rng=adafruit_l3gd20.L3DS20_RANGE_500DPS,
37 #     rate=adafruit_l3gd20.L3DS20_RATE_200HZ,
38 # )
39
40 while True:
41     print("Angular Momentum (rad/s): {}".format(SENSOR.gyro))
42     print()
43     time.sleep(1)

```

6.2 adafruit_l3gd20

Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - L3GD20

This is a CircuitPython driver for the Bosch L3GD20 nine degree of freedom inertial measurement unit module with sensor fusion.

- Author(s): Michael McWethy

6.2.1 Implementation Notes

Hardware:

- Adafruit L3GD20H Triple-Axis Gyro Breakout Board

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://circuitpython.org/downloads>
- Adafruit's Register library: https://github.com/adafruit/Adafruit_CircuitPython_Register

class `adafruit_l3gd20.L3GD20` (*rng=0, rate=0*)

Driver for the L3GD20 3-axis Gyroscope sensor.

Parameters

- **rng** (*int*) – a range value one of:
 - `L3DS20_RANGE_250DPS`
 - `L3DS20_RANGE_500DPS`
 - `L3DS20_RANGE_2000DPS`
 Defaults to `L3DS20_RANGE_250DPS`
- **rate** (*int*) – a rate value one of
 - `L3DS20_RATE_100HZ`
 - `L3DS20_RATE_200HZ`

- L3DS20_RATE_400HZ
 - L3DS20_RATE_800HZ
- Defaults to L3DS20_RATE_100HZ

gyro

x, y, z angular momentum tuple floats, rescaled appropriately for range selected

class adafruit_l3gd20.L3GD20_I2C(*i2c*, *rng*=0, *address*=107, *rate*=0)

Driver for L3GD20 Gyroscope using I2C communications

Parameters

- **i2c** (*I2C*) – The I2C bus the device is connected to
- **rng** (*int*) – range value. Defaults to 0x68
- **rate** (*int*) – rate value. Defaults to L3DS20_RATE_100HZ

Quickstart: Importing and using the device

Here is an example of using the `L3GD20_I2C` class. First you will need to import the libraries to use the sensor

```
import board
import adafruit_l3gd20
```

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
sensor = adafruit_l3gd20.L3GD20_I2C(i2c)
```

Now you have access to the `gyro` attribute

```
gyro_data = sensor.gyro
```

gyro_raw

Gives the raw gyro readings, in units of rad/s.

read_register (*register*)

Returns a byte value from a register

Parameters **register** – the register to read a byte

write_register (*register*, *value*)

Update a register with a byte value

Parameters

- **register** (*int*) – which device register to write
- **value** – a byte to write

class adafruit_l3gd20.L3GD20_SPI(*spi_busio*, *cs*, *rng*=0, *baudrate*=100000, *rate*=0)

Driver for L3GD20 Gyroscope using SPI communications

Parameters

- **spi_busio** (*SPI*) – The SPI bus the device is connected to
- **cs** (*DigitalInOut*) – digital in/out to use as chip select signal
- **rng** (*int*) – range value. Defaults to L3DS20_RANGE_250DPS.
- **baudrate** – SPI baud rate. Defaults to 100000

- **rate** (*int*) – rate value. Defaults to L3DS20_RATE_100HZ

Quickstart: Importing and using the device

Here is an example of using the `L3GD20_SPI` class. First you will need to import the libraries to use the sensor

```
import board
import adafruit_l3gd20
```

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

```
spi = board.SPI()
sensor = adafruit_l3gd20.L3GD20_SPI(spi)
```

Now you have access to the `gyro` attribute

```
gyro_data = sensor.gyro
```

gyro_raw

Gives the raw gyro readings, in units of rad/s.

read_bytes (*register, buffer*)

Low level register stream reading over SPI, returns a list of values

Parameters

- **register** – the register to read bytes
- **buffer** (*bytearray*) – buffer to fill with data from stream

read_register (*register*)

Low level register reading over SPI, returns a list of values

Parameters **register** – the register to read a byte

write_register (*register, value*)

Low level register writing over SPI, writes one 8-bit value

Parameters

- **register** (*int*) – which device register to write
- **value** – a byte to write

CHAPTER 7

Indices and tables

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

a

adafruit_l3gd20, [14](#)

A

`adafruit_l3gd20` (*module*), 14

G

`gyro` (*adafruit_l3gd20.L3GD20 attribute*), 15

`gyro_raw` (*adafruit_l3gd20.L3GD20_I2C attribute*), 15

`gyro_raw` (*adafruit_l3gd20.L3GD20_SPI attribute*), 16

L

`L3GD20` (*class in adafruit_l3gd20*), 14

`L3GD20_I2C` (*class in adafruit_l3gd20*), 15

`L3GD20_SPI` (*class in adafruit_l3gd20*), 15

R

`read_bytes()` (*adafruit_l3gd20.L3GD20_SPI method*), 16

`read_register()` (*adafruit_l3gd20.L3GD20_I2C method*), 15

`read_register()` (*adafruit_l3gd20.L3GD20_SPI method*), 16

W

`write_register()` (*adafruit_l3gd20.L3GD20_I2C method*), 15

`write_register()` (*adafruit_l3gd20.L3GD20_SPI method*), 16