
Adafruit L3GD20 Library Documentation

Release 1.0

Michael McWethy

Jan 20, 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 (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_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 busio
7  import adafruit_l3gd20
8
9  # Hardware I2C setup:
10 I2C = busio.I2C(board.SCL, board.SDA)
11 # Initializes L3GD20 object using default range, 250dps
12 SENSOR = adafruit_l3gd20.L3GD20_I2C(I2C)
13 # Initialize L3GD20 object using a custom range and output data rate (ODR).
14 # SENSOR = adafruit_l3gd20.L3GD20_I2C(
15 #     I2C, rng=adafruit_l3gd20.L3DS20_RANGE_500DPS, rate=adafruit_l3gd20.L3DS20_RATE_
16 #     ↪ 200HZ
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

```

(continues on next page)

(continued from previous page)

```
27 # adafruit_l3gd20.L3DS20_RATE_800HZ, 800Hz data rate
28
29 # Hardware SPI setup:
30 # import digitalio
31 # CS = digitalio.DigitalInOut(board.D5)
32 # SPIB = busio.SPI(board.SCK, board.MOSI, board.MISO)
33 # SENSOR = adafruit_l3gd20.L3GD20_SPI(SPIB, CS)
34 # SENSOR = adafruit_l3gd20.L3GD20_I2C(
35 #     SPIB,
36 #     CS,
37 #     rng=adafruit_l3gd20.L3DS20_RANGE_500DPS,
38 #     rate=adafruit_l3gd20.L3DS20_RATE_200HZ,
39 # )
40
41 while True:
42     print("Angular Momentum (rad/s): {}".format(SENSOR.gyro))
43     print()
44     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:

- [L3GD20H Triple-Axis Gyro Breakout Board](#)

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- 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` (default), `L3DS20_RANGE_500DPS`, or `L3DS20_RANGE_2000DPS`
- **rate** (*int*) – a rate value one of `L3DS20_RATE_100HZ` (default), `L3DS20_RATE_200HZ`, `L3DS20_RATE_400HZ`, or `L3DS20_RATE_800HZ`

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*) – initialized busio I2C class
- **rng** (*int*) – the optional range value: L3DS20_RANGE_250DPS(default), L3DS20_RANGE_500DPS, or L3DS20_RANGE_2000DPS
- **address** – the optional device address, 0x68 is the default address

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*) – initialized busio SPI class
- **cs** (*DigitalInOut*) – digital in/out to use as chip select signal
- **rng** (*int*) – the optional range value: L3DS20_RANGE_250DPS(default), L3DS20_RANGE_500DPS, or L3DS20_RANGE_2000DPS
- **baudrate** – spi baud rate default is 100000

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*), [14](#)

`gyro_raw` (*adafruit_l3gd20.L3GD20_I2C attribute*), [15](#)

`gyro_raw` (*adafruit_l3gd20.L3GD20_SPI attribute*), [15](#)

L

`L3GD20` (*class in adafruit_l3gd20*), [14](#)

`L3GD20_I2C` (*class in adafruit_l3gd20*), [14](#)

`L3GD20_SPI` (*class in adafruit_l3gd20*), [15](#)

R

`read_bytes()` (*adafruit_l3gd20.L3GD20_SPI method*), [15](#)

`read_register()` (*adafruit_l3gd20.L3GD20_I2C method*), [15](#)

`read_register()` (*adafruit_l3gd20.L3GD20_SPI method*), [15](#)

W

`write_register()` (*adafruit_l3gd20.L3GD20_I2C method*), [15](#)

`write_register()` (*adafruit_l3gd20.L3GD20_SPI method*), [15](#)