
Adafruit INA219 Library Documentation

Release 1.0

Dean Miller

Jul 12, 2019

Contents

1	Dependencies	3
2	Usage Example	5
3	Contributing	7
4	Building locally	9
4.1	Sphinx documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_ina219	12
5.2.1	Implementation Notes	12
6	Indices and tables	15
	Python Module Index	17
	Index	19

CircuitPython driver for the [INA219](#) current sensor.

CHAPTER 1

Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the [Adafruit library and driver bundle](#).

CHAPTER 2

Usage Example

see [example](#)

CHAPTER 3

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

CHAPTER 4

Building locally

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-ina219 --library_
↪location .
```

4.1 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/ina219_simpletest.py

```

1  """Sample code and test for adafruit_ina219"""
2
3  import time
4  import board
5  from adafruit_ina219 import ADCResolution, BusVoltageRange, INA219
6
7
8  i2c_bus = board.I2C()
9
10 ina219 = INA219(i2c_bus)
11
12 print("ina219 test")
13
14 # display some of the advanced field (just to test)
15 print("Config register:")
16 print("  bus_voltage_range:    0x%1X" % ina219.bus_voltage_range)
17 print("  gain:                   0x%1X" % ina219.gain)
18 print("  bus_adc_resolution:      0x%1X" % ina219.bus_adc_resolution)
19 print("  shunt_adc_resolution:    0x%1X" % ina219.shunt_adc_resolution)
20 print("  mode:                    0x%1X" % ina219.mode)
21 print("")
22
23 # optional : change configuration to use 32 samples averaging for both bus voltage_
24 ↪and shunt voltage
25 ina219.bus_adc_resolution = ADCResolution.ADCRES_12BIT_32S
26 ina219.shunt_adc_resolution = ADCResolution.ADCRES_12BIT_32S
27 # optional : change voltage range to 16V

```

(continues on next page)

(continued from previous page)

```

27 ina219.bus_voltage_range = BusVoltageRange.RANGE_16V
28
29 # measure and display loop
30 while True:
31     bus_voltage = ina219.bus_voltage      # voltage on V- (load side)
32     shunt_voltage = ina219.shunt_voltage  # voltage between V+ and V- across the_
↪shunt
33     current = ina219.current              # current in mA
34
35     # INA219 measure bus voltage on the load side. So PSU voltage = bus_voltage +_
↪shunt_voltage
36     print("PSU Voltage:    {:.3f} V".format(bus_voltage + shunt_voltage))
37     print("Shunt Voltage:  {:.3f} V".format(shunt_voltage))
38     print("Load Voltage:   {:.3f} V".format(bus_voltage))
39     print("Current:        {:.3f} A".format(current/1000))
40     print("")
41
42     time.sleep(2)

```

5.2 adafruit_ina219

CircuitPython driver for the INA219 current sensor.

- Author(s): Dean Miller

5.2.1 Implementation Notes

Hardware:

- [Adafruit INA219 High Side DC Current Sensor Breakout](#)
- [Adafruit INA219 FeatherWing](#)

Software and Dependencies:

- Adafruit CircuitPython firmware (2.2.0+) for the ESP8622 and M0-based boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class adafruit_ina219.**ADCResolution**
 Constants for bus_adc_resolution or shunt_adc_resolution

class adafruit_ina219.**BusVoltageRange**
 Constants for bus_voltage_range

class adafruit_ina219.**Gain**
 Constants for gain

class adafruit_ina219.**INA219** (i2c_bus, addr=64)
 Driver for the INA219 current sensor

bus_voltage
 The bus voltage (between V- and GND) in Volts

calibration
 Calibration register (cached value)

current

The current through the shunt resistor in milliamps.

power

The power through the load in Watt.

set_calibration_16V_400mA()

Configures to INA219 to be able to measure up to 16V and 400mA of current. Counter overflow occurs at 1.6A.

Note: These calculations assume a 0.1 ohm shunt resistor is present

set_calibration_32V_1A()

Configures to INA219 to be able to measure up to 32V and 1A of current. Counter overflow occurs at 1.3A.

Note: These calculations assume a 0.1 ohm shunt resistor is present

set_calibration_32V_2A()

Configures to INA219 to be able to measure up to 32V and 2A of current. Counter overflow occurs at 3.2A.

..note :: These calculations assume a 0.1 shunt ohm resistor is present

shunt_voltage

The shunt voltage (between V+ and V-) in Volts (so +/-0.327V)

class adafruit_ina219.**Mode**

Constants for mode

CHAPTER 6

Indices and tables

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

a

adafruit_ina219, [12](#)

A

`adafruit_ina219` (*module*), [12](#)
`ADCResolution` (*class in adafruit_ina219*), [12](#)

B

`bus_voltage` (*adafruit_ina219.INA219 attribute*), [12](#)
`BusVoltageRange` (*class in adafruit_ina219*), [12](#)

C

`calibration` (*adafruit_ina219.INA219 attribute*), [12](#)
`current` (*adafruit_ina219.INA219 attribute*), [12](#)

G

`Gain` (*class in adafruit_ina219*), [12](#)

I

`INA219` (*class in adafruit_ina219*), [12](#)

M

`Mode` (*class in adafruit_ina219*), [13](#)

P

`power` (*adafruit_ina219.INA219 attribute*), [13](#)

S

`set_calibration_16V_400mA()`
 (*adafruit_ina219.INA219 method*), [13](#)
`set_calibration_32V_1A()`
 (*adafruit_ina219.INA219 method*), [13](#)
`set_calibration_32V_2A()`
 (*adafruit_ina219.INA219 method*), [13](#)
`shunt_voltage` (*adafruit_ina219.INA219 attribute*),
 [13](#)