

---

# **Adafruit MLX90393 Library Documentation**

***Release 1.0***

**Kevin Townsend**

**May 22, 2020**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
1.1	Installing from PyPI . . . . .	3
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Documentation</b>	<b>9</b>
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_mlx90393 . . . . .	11
5.2.1	Implementation Notes . . . . .	12
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



Adafruit CircuitPython driver for the MLX90393 3-axis magnetometer.



# 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](#).

### 1.1 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-mlx90939
```

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

```
sudo pip3 install adafruit-circuitpython-mlx90939
```

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-mlx90939
```





## CHAPTER 2

---

### Usage Example

---

```
import time
import busio
import board

import adafruit_mlx90393

I2C_BUS = busio.I2C(board.SCL, board.SDA)
SENSOR = adafruit_mlx90393.MLX90393(I2C_BUS, gain=adafruit_mlx90393.GAIN_1X)

while True:
    MX, MY, MZ = SENSOR.magnetic
    print("[{}]" .format(time.monotonic()))
    print("X: {} uT".format(MX))
    print("Y: {} uT".format(MY))
    print("Z: {} uT".format(MZ))
    # Display the status field if an error occurred, etc.
    if SENSOR.last_status > adafruit_mlx90393.STATUS_OK:
        SENSOR.display_status()
    time.sleep(1.0)
```



## CHAPTER 3

---

### Contributing

---

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



## CHAPTER 4

---

### Documentation

---

For information on building library documentation, please check out [this guide](#).



## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/mlx90393\_simpletest.py

```
1 import time
2 import busio
3 import board
4
5 import adafruit_mlx90393
6
7 I2C_BUS = busio.I2C(board.SCL, board.SDA)
8 SENSOR = adafruit_mlx90393.MLX90393(I2C_BUS, gain=adafruit_mlx90393.GAIN_1X)
9
10 while True:
11     MX, MY, MZ = SENSOR.magnetic
12     print("{} {}".format(time.monotonic(), MX))
13     print("X: {} uT".format(MX))
14     print("Y: {} uT".format(MY))
15     print("Z: {} uT".format(MZ))
16     # Display the status field if an error occurred, etc.
17     if SENSOR.last_status > adafruit_mlx90393.STATUS_OK:
18         SENSOR.display_status()
19     time.sleep(1.0)
```

## 5.2 adafruit\_mlx90393

This is a breakout for the Adafruit MLX90393 magnetometer sensor breakout.

- Author(s): ktown

## 5.2.1 Implementation Notes

### Hardware:

- Adafruit [MLX90393 Magnetometer Sensor Breakout Board](#) (Product ID: 4022)

### Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: [https://github.com/adafruit/Adafruit\\_CircuitPython\\_BusDevice](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)

**class** `adafruit_mlx90393.MLX90393` (*i2c\_bus*, *address=12*, *gain=7*, *debug=False*)

Driver for the MLX90393 magnetometer. :param *i2c\_bus*: The `busio.I2C` object to use. This is the only required parameter. :param *int address*: (optional) The I2C address of the device. :param *int gain*: (optional) The gain level to apply. :param *bool debug*: (optional) Enable debug output.

**display\_status** ()

Prints out the content of the last status byte in a human-readable format.

**gain**

Gets the current gain setting for the device.

**last\_status**

Returns the last status byte received from the sensor.

**magnetic**

The processed magnetometer sensor values. A 3-tuple of X, Y, Z axis values in microteslas that are signed floats.

**read\_data**

Reads a single X/Y/Z sample from the magnetometer.

**read\_reg** (*reg*)

Gets the current value of the specified register.

**reset** ()

Performs a software reset of the sensor.



## CHAPTER 6

---

### Indices and tables

---

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



### a

adafruit\_mlx90393, [11](#)



### A

`adafruit_mlx90393` (*module*), [11](#)

### D

`display_status()` (*adafruit\_mlx90393.MLX90393*  
*method*), [12](#)

### G

`gain` (*adafruit\_mlx90393.MLX90393* *attribute*), [12](#)

### L

`last_status` (*adafruit\_mlx90393.MLX90393* *at-*  
*tribute*), [12](#)

### M

`magnetic` (*adafruit\_mlx90393.MLX90393* *attribute*),  
[12](#)

`MLX90393` (*class in adafruit\_mlx90393*), [12](#)

### R

`read_data` (*adafruit\_mlx90393.MLX90393* *attribute*),  
[12](#)

`read_reg()` (*adafruit\_mlx90393.MLX90393* *method*),  
[12](#)

`reset()` (*adafruit\_mlx90393.MLX90393* *method*), [12](#)