

---

# AdafruitMPRLS Library Documentation

*Release 1.0*

**ladyada**

**Jul 20, 2019**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Building locally</b>	<b>9</b>
4.1	Zip release files . . . . .	9
4.2	Sphinx documentation . . . . .	9
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_mprls . . . . .	12
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>



CircuitPython library to support Honeywell MPRLS digital pressure sensors.



# 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

---

```
import time
import board
import busio
import adafruit_mprls

i2c = busio.I2C(board.SCL, board.SDA)

# Simplest use, connect to default over I2C
mpr = adafruit_mprls.MPRLS(i2c, psi_min=0, psi_max=25)

while True:
    print((mpr.pressure,))
    time.sleep(1)
```



## CHAPTER 3

---

### Contributing

---

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



## CHAPTER 4

---

### Building locally

---

#### 4.1 Zip release files

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-mprls --library_
↳ location .
```

#### 4.2 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/mprls\_simpletest.py

```
1 import time
2 import board
3 import busio
4 import adafruit_mprls
5
6 i2c = busio.I2C(board.SCL, board.SDA)
7
8 # Simplest use, connect to default over I2C
9 mpr = adafruit_mprls.MPRLS(i2c, psi_min=0, psi_max=25)
10
11 # You can also specify both reset and eoc pins
12 """
13 import digitalio
14 reset = digitalio.DigitalInOut(board.D5)
15 eoc = digitalio.DigitalInOut(board.D6)
16 mpr = adafruit_mprls.MPRLS(i2c, eoc_pin=eoc, reset_pin=reset,
17                            psi_min=0, psi_max=25)
18 """
19
20 while True:
21     print((mpr.pressure,))
22     time.sleep(1)
```

## 5.2 adafruit\_mprls

CircuitPython library to support Honeywell MPRLS digital pressure sensors

- Author(s): ladyada

### 5.2.1 Implementation Notes

**Hardware:**

**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_mprls.MPRLS` (*i2c\_bus*, \*, *addr*=24, *reset\_pin*=None, *eoc\_pin*=None, *psi\_min*=0, *psi\_max*=25)

Driver base for the MPRLS pressure sensor :param *i2c\_bus*: The `busio.I2C` object to use. This is the only required parameter. :param *int addr*: The optional I2C address, defaults to 0x18 :param *microcontroller.Pin reset\_pin*: Optional digitalio pin for hardware resetting :param *microcontroller.Pin eoc\_pin*: Optional digitalio pin for getting End Of Conversion signal :param *float psi\_min*: The minimum pressure in PSI, defaults to 0 :param *float psi\_max*: The maximum pressure in PSI, defaults to 25

**pressure**

The measured pressure, in hPa



## CHAPTER 6

---

### Indices and tables

---

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



### **a**

adafruit\_mprls, [11](#)



## A

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

## M

`MPRLS` (*class in `adafruit_mprls`*), [12](#)

## P

`pressure` (*`adafruit_mprls.MPRLS` attribute*), [12](#)