
Adafruit MPR121 Library Documentation

Release 1.0

Tony DiCola

Jan 22, 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_mpr121	15
6.2.1	Implementation Notes	15
7	Indices and tables	17
	Python Module Index	19
	Index	21

Adafruit CircuitPython module for the MPR121 capacitive touch breakout board.

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

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-mpu121
```

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

```
sudo pip3 install adafruit-circuitpython-mpu121
```

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-mpu121
```


CHAPTER 3

Usage Example

See usage in the `examples/mpr121_simpletest.py` file.

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

Ensure your device works with this simple test.

Listing 1: examples/mpr121_simpletest.py

```
1  # SPDX-FileCopyrightText: 2017 Tony DiCola for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  # Simple test of the MPR121 capacitive touch sensor library.
5  # Will print out a message when any of the 12 capacitive touch inputs of the
6  # board are touched. Open the serial REPL after running to see the output.
7  # Author: Tony DiCola
8  import time
9  import board
10 import busio
11
12 # Import MPR121 module.
13 import adafruit_mpr121
14
15 # Create I2C bus.
16 i2c = busio.I2C(board.SCL, board.SDA)
17
18 # Create MPR121 object.
19 mpr121 = adafruit_mpr121.MPR121(i2c)
20
21 # Note you can optionally change the address of the device:
22 # mpr121 = adafruit_mpr121.MPR121(i2c, address=0x91)
23
24 # Loop forever testing each input and printing when they're touched.
25 while True:
26     # Loop through all 12 inputs (0-11).
27     for i in range(12):
```

(continues on next page)

(continued from previous page)

```

28     # Call is_touched and pass it then number of the input.  If it's touched
29     # it will return True, otherwise it will return False.
30     if mpr121[i].value:
31         print("Input {} touched!".format(i))
32     time.sleep(0.25)  # Small delay to keep from spamming output messages.

```

Listing 2: examples/mpr121_piano.py

```

1  # SPDX-FileCopyrightText: 2017 Tony DiCola, Carter Nelson for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  # MPR121 piano demo.
5  # Listens to the first 7 inputs of the MPR121 and plays a middle scale note
6  # when an input is touched.  Note only one note is played at a time!
7  # For use with microcontrollers or computers with PWM support only!
8
9
10 import board
11 import busio
12 import pulseio
13
14 # Import MPR121 module.
15 import adafruit_mpr121
16
17
18 # Configure PWM buzzer and other state:
19 BUZZER_PIN = board.D9
20 TONE_ON_DUTY = 2 ** 15  # Duty cycle of tone when turned on, a square wave.
21 TONE_OFF_DUTY = 0  # Duty cycle of tone when turned off, 0 or no signal.
22 NOTE_FREQS = [
23     261,  # Input 0 = 261 hz = middle C
24     294,  # Input 1 = middle D
25     329,  # Input 2 = middle E
26     349,  # Input 3 = middle F
27     392,  # Input 4 = middle G
28     440,  # Input 5 = middle A
29     493,  # Input 6 = middle B
30     0,    # Input 7 = nothing (set to a frequency in hertz!)
31     0,    # Input 8
32     0,    # Input 9
33     0,    # Input 10
34     0,
35 ]  # Input 11
36
37
38 # Create I2C bus.
39 i2c = busio.I2C(board.SCL, board.SDA)
40
41 # Create MPR121 class.
42 mpr121 = adafruit_mpr121.MPR121(i2c)
43 # Note you can optionally change the address of the device:
44 # mpr121 = adafruit_mpr121.MPR121(i2c, address=0x91)
45
46 # pylint: disable-msg=no-member
47 # Setup buzzer PWM output.
48 buzzer = pulseio.PWMOut(

```

(continues on next page)

(continued from previous page)

```

49     BUZZER_PIN, duty_cycle=TONE_OFF_DUTY, frequency=440, variable_frequency=True
50 )
51 # pylint: disable-msg=no-member
52
53 last_note = None
54 while True:
55     # Get touched state for all pins
56     touched = mpr121.touched_pins
57     # If no pins are touched, be quiet
58     if True not in touched:
59         last_note = None
60         buzzer.duty_cycle = TONE_OFF_DUTY
61         continue
62     # Get index of touched pin
63     note = touched.index(True)
64     # Play note if pin is different and has a defined note
65     if note != last_note and NOTE_FREQS[note] != 0:
66         last_note = note
67         buzzer.frequency = NOTE_FREQS[note]
68         buzzer.duty_cycle = TONE_ON_DUTY

```

6.2 adafruit_mpr121

CircuitPython driver for the MPR121 capacitive touch breakout board.

See usage in the examples/simpletest.py file.

- Author(s): Tony DiCola

6.2.1 Implementation Notes

Hardware:

- Adafruit 12-Key Capacitive Touch Sensor Breakout - MPR121 (Product ID: 1982)
- Adafruit 12 x Capacitive Touch Shield for Arduino - MPR121 (Product ID: 2024)

Software and Dependencies:

- Adafruit CircuitPython firmware 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_mpr121.MPR121` (*i2c*, *address=90*)
 Driver for the MPR121 capacitive touch breakout board.

baseline_data (*pin*)

Return baseline data register value for the provided pin (0-11). Useful for debugging.

filtered_data (*pin*)

Return filtered data register value for the provided pin (0-11). Useful for debugging.

is_touched (*pin*)

Return True if the specified pin is being touched, otherwise returns False.

reset()

Reset the MPR121 into a default state ready to detect touch inputs.

touched()

Return touch state of all pins as a 12-bit value where each bit represents a pin, with a value of 1 being touched and 0 not being touched.

touched_pins

A tuple of touched state for all pins.

class adafruit_mpr121.**MPR121_Channel** (*mpr121, channel*)

Helper class to represent a touch channel on the MPR121. Not meant to be used directly.

raw_value

The raw touch measurement.

release_threshold

The release threshold.

threshold

The touch threshold.

value

Whether the touch pad is being touched or not.

CHAPTER 7

Indices and tables

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

a

adafruit_mpr121, [15](#)

A

`adafruit_mpr121` (*module*), 15

B

`baseline_data()` (*adafruit_mpr121.MPR121 method*), 15

F

`filtered_data()` (*adafruit_mpr121.MPR121 method*), 15

I

`is_touched()` (*adafruit_mpr121.MPR121 method*), 15

M

`MPR121` (*class in adafruit_mpr121*), 15

`MPR121_Channel` (*class in adafruit_mpr121*), 16

R

`raw_value` (*adafruit_mpr121.MPR121_Channel attribute*), 16

`release_threshold` (*adafruit_mpr121.MPR121_Channel attribute*), 16

`reset()` (*adafruit_mpr121.MPR121 method*), 15

T

`threshold` (*adafruit_mpr121.MPR121_Channel attribute*), 16

`touched()` (*adafruit_mpr121.MPR121 method*), 16

`touched_pins` (*adafruit_mpr121.MPR121 attribute*), 16

V

`value` (*adafruit_mpr121.MPR121_Channel attribute*), 16