
Adafruit MPR121 Library Documentation

Release 1.0

Tony DiCola

Oct 16, 2018

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_mpr121	13
5.2.1	Implementation Notes	13
6	Indices and tables	15
	Python Module Index	17

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

Usage Example

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

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-mpr121 --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/mpr121_simpletest.py

```
1  # Simple test of the MPR121 capacitive touch sensor library.
2  # Will print out a message when any of the 12 capacitive touch inputs of the
3  # board are touched. Open the serial REPL after running to see the output.
4  # Author: Tony DiCola
5  import time
6  import board
7  import busio
8  # Import MPR121 module.
9  import adafruit_mpr121
10
11 # Create I2C bus.
12 i2c = busio.I2C(board.SCL, board.SDA)
13
14 # Create MPR121 class.
15 mpr121 = adafruit_mpr121.MPR121(i2c)
16 # Note you can optionally change the address of the device:
17 #mpr121 = adafruit_mpr121.MPR121(i2c, address=0x91)
18
19 # Loop forever testing each input and printing when they're touched.
20 while True:
21     # Loop through all 12 inputs (0-11).
22     for i in range(12):
23         # Call is_touched and pass it then number of the input. If it's touched
24         # it will return True, otherwise it will return False.
25         if mpr121[i].value:
26             print('Input {} touched!'.format(i))
27     time.sleep(0.25) # Small delay to keep from spamming output messages.
```

Listing 2: examples/piano.py

```

1  # MPR121 piano demo.
2  # Listens to the first 7 inputs of the MPR121 and plays a middle scale note
3  # when an input is touched. Note only one note is played at a time!
4  # For use with microcontrollers or computers with PWM support only!
5  # Author: Tony DiCola
6  import time
7
8  import board
9  import busio
10 import pulseio
11
12 # Import MPR121 module.
13 import adafruit_mpr121
14
15
16 # Configure PWM buzzer and other state:
17 BUZZER_PIN = board.D9
18 TONE_ON_DUTY = 2**15 # Duty cycle of tone when turned on, a square wave.
19 TONE_OFF_DUTY = 0    # Duty cycle of tone when turned off, 0 or no signal.
20 NOTE_FREQS = [261,   # Input 0 = 261 hz = middle C
21               294,   # Input 1 = middle D
22               329,   # Input 2 = middle E
23               349,   # Input 3 = middle F
24               392,   # Input 4 = middle G
25               440,   # Input 5 = middle A
26               493,   # Input 6 = middle B
27               0,     # Input 7 = nothing (set to a frequency in hertz!)
28               0,     # Input 8
29               0,     # Input 9
30               0,     # Input 10
31               0]     # Input 11
32
33
34 # Create I2C bus.
35 i2c = busio.I2C(board.SCL, board.SDA)
36
37 # Create MPR121 class.
38 mpr121 = adafruit_mpr121.MPR121(i2c)
39 # Note you can optionally change the address of the device:
40 #mpr121 = adafruit_mpr121.MPR121(i2c, address=0x91)
41
42 # Setup buzzer PWM output.
43 buzzer = pulseio.PWMOut(BUZZER_PIN, duty_cycle=TONE_OFF_DUTY, frequency=440,
44                          variable_frequency=True)
45
46 # Main loop.
47 # First grab an initial touch state for all of the inputs. The touched()
48 # function can quickly get the state of all input pins and returns them as a
49 # 12-bit value with a bit set to 1 for each appropriate input (i.e. bit 0 is
50 # input 0, bit 1 is input 1, etc.)
51 last = mpr121.touched()
52 while True:
53     # Every loop iteration get an updated touch state and look to see if it
54     # changed since the last iteration.
55     current = mpr121.touched()

```

(continues on next page)

(continued from previous page)

```

56     if last != current:
57         # Some pin changed, turn off playback and look for any touched pins.
58         buzzer.duty_cycle = TONE_OFF_DUTY
59         # Loop through all 12 inputs (0-11) and look at their bits in the
60         # current touch state. A bit that's set is touched!
61         for i in range(12):
62             if (1 << i) & current > 0:
63                 print('Input {} touched!'.format(i))
64                 # Grab the frequency for the associated pin and check that it's
65                 # not zero (unused).
66                 freq = NOTE_FREQS[i]
67                 if freq != 0:
68                     # Pin with a specified frequency was touched, play the tone!
69                     buzzer.frequency = NOTE_FREQS[i]
70                     buzzer.duty_cycle = TONE_ON_DUTY
71         # Update touch state and delay a bit before next loop iteration.
72         last = current
73         time.sleep(0.01)

```

5.2 adafruit_mpr121

CircuitPython driver for the MPR121 capacitive touch breakout board.

See usage in the examples/simpletest.py file.

- Author(s): Tony DiCola

5.2.1 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.

set_thresholds (*touch, release*)

Set the touch and release threshold for all inputs to the provided values. Both touch and release should be a value between 0 to 255 (inclusive).

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.

value

Whether the touch pad is being touched or not.

CHAPTER 6

Indices and tables

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

a

adafruit_mpr121, [13](#)

A

`adafruit_mpr121` (module), [13](#)

B

`baseline_data()` (`adafruit_mpr121.MPR121` method), [13](#)

F

`filtered_data()` (`adafruit_mpr121.MPR121` method), [13](#)

I

`is_touched()` (`adafruit_mpr121.MPR121` method), [13](#)

M

`MPR121` (class in `adafruit_mpr121`), [13](#)

`MPR121_Channel` (class in `adafruit_mpr121`), [14](#)

R

`raw_value` (`adafruit_mpr121.MPR121_Channel` attribute), [14](#)

`reset()` (`adafruit_mpr121.MPR121` method), [13](#)

S

`set_thresholds()` (`adafruit_mpr121.MPR121` method), [13](#)

T

`touched()` (`adafruit_mpr121.MPR121` method), [14](#)

`touched_pins` (`adafruit_mpr121.MPR121` attribute), [14](#)

V

`value` (`adafruit_mpr121.MPR121_Channel` attribute), [14](#)