Adafruit MPR121 Library Documentation

Release 1.0

Tony DiCola

Jan 16, 2019

Contents

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.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/mpr121_simpletest.py

```
# Simple test of the MPR121 capacitive touch sensor library.
# Will print out a message when any of the 12 capacitive touch inputs of the
# board are touched.  Open the serial REPL after running to see the output.
# Author: Tony DiCola
import time
import board
import busio
# Import MPR121 module.
import adafruit_mpr121

# Create I2C bus.
i2c = busio.I2C(board.SCL, board.SDA)

# Create MPR121 object.
mpr121 = adafruit_mpr121.MPR121(i2c)

# Note you can optionally change the address of the device:
#mpr121 = adafruit_mpr121.MPR121(i2c, address=0x91)

# Loop forever testing each input and printing when they're touched.
while True:
    # Loop through all 12 inputs (0-11).
    for i in range(12):
        # Call is_touched and pass it then number of the input.  If it's touched
        # it will return True, otherwise it will return False.
        if mpr121[i].value:
            print('Input {} touched!'.format(i))
```

(continues on next page)

(continued from previous page)

```
    time.sleep(0.25)  # Small delay to keep from spamming output messages.
```

Listing 2: examples/piano.py

```
# MPR121 piano demo.
# Listens to the first 7 inputs of the MPR121 and plays a middle scale note
# when an input is touched.  Note only one note is played at a time!
# For use with microcontrollers or computers with PWM support only!
# Author: Tony DiCola
# Modified by: Carter Nelson

import board
import busio
import pulseio

# Import MPR121 module.
import adafruit_mpr121


# Configure PWM buzzer and other state:
BUZZER_PIN = board.D9
TONE_ON_DUTY = 2**15   # Duty cycle of tone when turned on, a square wave.
TONE_OFF_DUTY = 0      # Duty cycle of tone when turned off, 0 or no signal.
NOTE_FREQS = [261,  # Input 0 = 261 hz = middle C
              294,  # Input 1 = middle D
              329,  # Input 2 = middle E
              349,  # Input 3 = middle F
              392,  # Input 4 = middle G
              440,  # Input 5 = middle A
              493,  # Input 6 = middle B
              0,    # Input 7 = nothing (set to a frequency in hertz!)
              0,    # Input 8
              0,    # Input 9
              0,    # Input 10
              0]    # Input 11


# Create I2C bus.
i2c = busio.I2C(board.SCL, board.SDA)

# Create MPR121 class.
mpr121 = adafruit_mpr121.MPR121(i2c)
# Note you can optionally change the address of the device:
#mpr121 = adafruit_mpr121.MPR121(i2c, address=0x91)

# Setup buzzer PWM output.
buzzer = pulseio.PWMOut(BUZZER_PIN, duty_cycle=TONE_OFF_DUTY, frequency=440,
                        variable_frequency=True)

last_note = None
while True:
    # Get touched state for all pins
    touched = mpr121.touched_pins
    # If no pins are touched, be quiet
    if True not in touched:
        last_note = None
```

(continues on next page)

(continued from previous page)

```
        buzzer.duty_cycle = TONE_OFF_DUTY
        continue
    # Get index of touched pin
    note = touched.index(True)
    # Play note if pin is different and has a defined note
    if note != last_note and NOTE_FREQS[note] != 0:
        last_note = note
        buzzer.frequency = NOTE_FREQS[note]
        buzzer.duty_cycle = TONE_ON_DUTY
```

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.

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 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index