
AdafruitMCP3xxx Library Documentation

Release 1.0

ladyada

Oct 14, 2019

Contents

1	Dependencies	3
2	Usage Example	5
2.1	MCP3008 Single Ended	5
2.2	MCP3008 Differential	5
2.3	MCP3004 Single-Ended	6
2.4	MCP3004 Differential	6
3	Contributing	9
4	Building locally	11
4.1	Zip release files	11
4.2	Sphinx documentation	11
5	Table of Contents	13
5.1	Simple test	13
5.2	API	15
5.2.1	MCP3xxx	15
5.2.1.1	Implementation Notes	15
5.2.2	MCP3004	15
5.2.3	MCP3008	16
5.2.4	AnalogIn	16
6	Indices and tables	17
	Python Module Index	19
	Index	21

CircuitPython library for the MCP3xxx series of analog-to-digital converters.

Currently supports:

- [MCP3008: 8-Channel 10-Bit ADC With SPI Interface](#)

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

2.1 MCP3008 Single Ended

```
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn

# create the spi bus
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

# create the cs (chip select)
cs = digitalio.DigitalInOut(board.D5)

# create the mcp object
mcp = MCP.MCP3008(spi, cs)

# create an analog input channel on pin 0
chan = AnalogIn(mcp, MCP.P0)

print('Raw ADC Value: ', chan.value)
print('ADC Voltage: ' + str(chan.voltage) + 'V')
```

2.2 MCP3008 Differential

```
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn
```

(continues on next page)

(continued from previous page)

```
# create the spi bus
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

# create the cs (chip select)
cs = digitalio.DigitalInOut(board.D5)

# create the mcp object
mcp = MCP.MCP3008(spi, cs)

# create a differential ADC channel between Pin 0 and Pin 1
chan = AnalogIn(mcp, MCP.P0, MCP.P1)

print('Differential ADC Value: ', chan.value)
print('Differential ADC Voltage: ' + str(chan.voltage) + 'V')
```

2.3 MCP3004 Single-Ended

```
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3004 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn

# create the spi bus
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

# create the cs (chip select)
cs = digitalio.DigitalInOut(board.D5)

# create the mcp object
mcp = MCP.MCP3004(spi, cs)

# create an analog input channel on pin 0
chan = AnalogIn(mcp, MCP.P0)

print('Raw ADC Value: ', chan.value)
print('ADC Voltage: ' + str(chan.voltage) + 'V')
```

2.4 MCP3004 Differential

```
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3004 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn

# create the spi bus
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

# create the cs (chip select)
```

(continues on next page)

(continued from previous page)

```
cs = digitalio.DigitalInOut(board.D5)

# create the mcp object
mcp = MCP.MCP3004(spi, cs)

# create a differential ADC channel between Pin 0 and Pin 1
chan = AnalogIn(mcp, MCP.P0, MCP.P1)

print('Differential ADC Value: ', chan.value)
print('Differential ADC Voltage: ' + str(chan.voltage) + 'V')
```


CHAPTER 3

Contributing

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

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-mcp3xxx --
↪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/mcp3xxx_mcp3008_single_ended_simpletest.py

```
1 import busio
2 import digitalio
3 import board
4 import adafruit_mcp3xxx.mcp3008 as MCP
5 from adafruit_mcp3xxx.analog_in import AnalogIn
6
7 # create the spi bus
8 spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
9
10 # create the cs (chip select)
11 cs = digitalio.DigitalInOut(board.D5)
12
13 # create the mcp object
14 mcp = MCP.MCP3008(spi, cs)
15
16 # create an analog input channel on pin 0
17 chan = AnalogIn(mcp, MCP.P0)
18
19 print('Raw ADC Value: ', chan.value)
20 print('ADC Voltage: ' + str(chan.voltage) + 'V')
```

Listing 2: examples/mcp3xxx_mcp3004_single_ended_simpletest.py

```
1 import busio
2 import digitalio
3 import board
4 import adafruit_mcp3xxx.mcp3004 as MCP
```

(continues on next page)

(continued from previous page)

```

5  from adafruit_mcp3xxx.analog_in import AnalogIn
6
7  # create the spi bus
8  spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
9
10 # create the cs (chip select)
11 cs = digitalio.DigitalInOut(board.D5)
12
13 # create the mcp object
14 mcp = MCP.MCP3004(spi, cs)
15
16 # create an analog input channel on pin 0
17 chan = AnalogIn(mcp, MCP.P0)
18
19 print('Raw ADC Value: ', chan.value)
20 print('ADC Voltage: ' + str(chan.voltage) + 'V')

```

Listing 3: examples/mcp3xxx_mcp3008_differential_simpletest.py

```

1  import busio
2  import digitalio
3  import board
4  import adafruit_mcp3xxx.mcp3008 as MCP
5  from adafruit_mcp3xxx.analog_in import AnalogIn
6
7  # create the spi bus
8  spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
9
10 # create the cs (chip select)
11 cs = digitalio.DigitalInOut(board.D5)
12
13 # create the mcp object
14 mcp = MCP.MCP3008(spi, cs)
15
16 # create a differential ADC channel between Pin 0 and Pin 1
17 chan = AnalogIn(mcp, MCP.P0, MCP.P1)
18
19 print('Differential ADC Value: ', chan.value)
20 print('Differential ADC Voltage: ' + str(chan.voltage) + 'V')

```

Listing 4: examples/mcp3xxx_mcp3004_differential_simpletest.py

```

1  import busio
2  import digitalio
3  import board
4  import adafruit_mcp3xxx.mcp3004 as MCP
5  from adafruit_mcp3xxx.analog_in import AnalogIn
6
7  # create the spi bus
8  spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
9
10 # create the cs (chip select)
11 cs = digitalio.DigitalInOut(board.D5)
12
13 # create the mcp object
14 mcp = MCP.MCP3004(spi, cs)

```

(continues on next page)

(continued from previous page)

```

15
16 # create a differential ADC channel between Pin 0 and Pin 1
17 chan = AnalogIn(mcp, MCP.P0, MCP.P1)
18
19 print('Differential ADC Value: ', chan.value)
20 print('Differential ADC Voltage: ' + str(chan.voltage) + 'V')

```

5.2 API

5.2.1 MCP3xxx

CircuitPython Library for MCP3xxx ADCs with SPI

- Author(s): ladyada, Brent Rubell

5.2.1.1 Implementation Notes

Hardware:

- Adafruit [MCP3008 8-Channel 10-Bit ADC with SPI](#) (Product ID: 856)

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

class `adafruit_mcp3xxx.mcp3xxx.MCP3xxx` (*spi_bus*, *cs*, *ref_voltage=3.3*)
MCP3xxx Interface.

Parameters

- **spi_bus** (*SPIDevice*) – SPI bus the ADC is connected to.
- **cs** (*DigitalInOut*) – Chip Select Pin.
- **ref_voltage** (*float*) – Voltage into (Vin) the ADC.

read (*pin*, *is_differential=False*)
SPI Interface for MCP3xxx-based ADCs reads.

Parameters

- **pin** (*int*) – individual or differential pin.
- **is_differential** (*bool*) – single-ended or differential read.

reference_voltage
Returns the MCP3xxx's reference voltage.

5.2.2 MCP3004

MCP3004 4-channel, 10-bit, analog-to-digital converter instance.

- Author(s): Brent Rubell

class `adafruit_mcp3xxx.mcp3004.MCP3004` (*spi_bus*, *cs*, *ref_voltage=3.3*)
Bases: `adafruit_mcp3xxx.mcp3xxx.MCP3xxx`

MCP3004 Differential channel mapping.

- 0: CH0 = IN+, CH1 = IN-
- 1: CH1 = IN+, CH0 = IN-
- 2: CH2 = IN+, CH3 = IN-
- 3: CH3 = IN+, CH2 = IN-

5.2.3 MCP3008

MCP3008 8-channel, 10-bit, analog-to-digital converter instance.

- Author(s): Brent Rubell

class adafruit_mcp3xxx.mcp3008.**MCP3008** (*spi_bus*, *cs*, *ref_voltage=3.3*)

Bases: *adafruit_mcp3xxx.mcp3xxx.MCP3xxx*

MCP3008 Differential channel mapping.

- 0: CH0 = IN+, CH1 = IN-
- 1: CH1 = IN+, CH0 = IN-
- 2: CH2 = IN+, CH3 = IN-
- 3: CH3 = IN+, CH2 = IN-
- 4: CH4 = IN+, CH5 = IN-
- 5: CH5 = IN+, CH4 = IN-
- 6: CH6 = IN+, CH7 = IN-
- 7: CH7 = IN+, CH6 = IN-

5.2.4 AnalogIn

AnalogIn for single-ended and differential ADC readings.

- Author(s): Brent Rubell

class adafruit_mcp3xxx.analog_in.**AnalogIn** (*mcp*, *positive_pin*, *negative_pin=None*)

AnalogIn Mock Implementation for ADC Reads.

Parameters

- **mcp** (*MCP3004*, *MCP3008*) – The mcp object.
- **positive_pin** (*int*) – Required pin for single-ended.
- **negative_pin** (*int*) – Optional pin for differential reads.

value

Returns the value of an ADC pin as an integer.

voltage

Returns the voltage from the ADC pin as a floating point value.

CHAPTER 6

Indices and tables

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

a

`adafruit_mcp3xxx.analog_in`, [16](#)
`adafruit_mcp3xxx.mcp3004`, [15](#)
`adafruit_mcp3xxx.mcp3008`, [16](#)
`adafruit_mcp3xxx.mcp3xxx`, [15](#)

A

`adafruit_mcp3xxx.analog_in` (*module*), 16
`adafruit_mcp3xxx.mcp3004` (*module*), 15
`adafruit_mcp3xxx.mcp3008` (*module*), 16
`adafruit_mcp3xxx.mcp3xxx` (*module*), 15
`AnalogIn` (*class in adafruit_mcp3xxx.analog_in*), 16

M

`MCP3004` (*class in adafruit_mcp3xxx.mcp3004*), 15
`MCP3008` (*class in adafruit_mcp3xxx.mcp3008*), 16
`MCP3xxx` (*class in adafruit_mcp3xxx.mcp3xxx*), 15

R

`read()` (*adafruit_mcp3xxx.mcp3xxx.MCP3xxx*
method), 15
`reference_voltage`
(*adafruit_mcp3xxx.mcp3xxx.MCP3xxx* *at-*
tribute), 15

V

`value` (*adafruit_mcp3xxx.analog_in.AnalogIn* *at-*
tribute), 16
`voltage` (*adafruit_mcp3xxx.analog_in.AnalogIn*
attribute), 16