

---

# AdafruitMCP3xxx Library Documentation

*Release 1.0*

**ladyada**

**Jan 14, 2020**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Example</b>	<b>7</b>
3.1	MCP3008 Single Ended . . . . .	7
3.2	MCP3008 Differential . . . . .	7
3.3	MCP3004 Single-Ended . . . . .	8
3.4	MCP3004 Differential . . . . .	8
<b>4</b>	<b>Contributing</b>	<b>11</b>
<b>5</b>	<b>Documentation</b>	<b>13</b>
<b>6</b>	<b>Table of Contents</b>	<b>15</b>
6.1	Simple test . . . . .	15
6.2	API . . . . .	18
6.2.1	MCP3xxx . . . . .	18
6.2.1.1	Implementation Notes . . . . .	18
6.2.2	AnalogIn . . . . .	19
6.2.3	MCP3008 . . . . .	19
6.2.4	MCP3004 . . . . .	20
6.2.5	MCP3002 . . . . .	20
<b>7</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



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

---

### 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-mcp3xxx
```

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

```
sudo pip3 install adafruit-circuitpython-mcp3xxx
```

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-mcp3xxx
```



### 3.1 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')
```

### 3.2 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')
```

### 3.3 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')
```

### 3.4 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 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/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\_mcp3002\_single\_ended\_simpletest.py

```

1  import busio
2  import digitalio
3  import board
4  import adafruit_mcp3xxx.mcp3002 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.MCP3002(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 4: 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)

```

(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')

```

Listing 5: 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)
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 6: examples/mcp3xxx\_mcp3002\_differential\_simpletest.py

```

1 import busio
2 import digitalio
3 import board
4 import adafruit_mcp3xxx.mcp3002 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.MCP3002(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')

```

## 6.2 API

### 6.2.1 MCP3xxx

CircuitPython Library for MCP3xxx ADCs with SPI

- Author(s): ladyada, Brent Rubell

#### 6.2.1.1 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](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)

---

**Note:** The ADC chips’ input pins (AKA “channels”) are aliased in this library as integer variables whose names start with “P” (eg `MCP3008.P0` is channel 0 on the MCP3008 chip). Each module that contains a driver class for a particular ADC chip has these aliases predefined accordingly. This is done for code readability and prevention of erroneous SPI commands.

---



---

**Important:** The differential reads (comparisons done by the ADC chip) are limited to certain pairs of channels. These predefined pairs are referenced in this documentation as differential channel mappings. Please refer to the driver class of your ADC chip (*MCP3008*, *MCP3004*, *MCP3002*) for a list of available differential channel mappings.

---

**class** `adafruit_mcp3xxx.mcp3xxx.MCP3xxx` (*spi\_bus*, *cs*, *ref\_voltage=3.3*)

This abstract base class is meant to be inherited by *MCP3008*, *MCP3004*, or *MCP3002* child classes.

##### 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. Due to 10-bit accuracy, the returned value ranges [0, 1023].

##### Parameters

- **pin** (*int*) – individual or differential pin.
- **is\_differential** (*bool*) – single-ended or differential read.

---

**Note:** This library offers a helper class called *AnalogIn* for both single-ended and differential reads. If you opt to not implement *AnalogIn* during differential reads, then the `pin` parameter should be the first of the two pins associated with the desired differential channel mapping.

---

**reference\_voltage**

Returns the MCP3xxx's reference voltage. (read-only)

## 6.2.2 AnalogIn

AnalogIn for single-ended and differential ADC readings.

- Author(s): Brent Rubell

**Warning:** The ADC chips supported by this library do not use negative numbers. If the resulting differential read is less than 0, then the returned integer value (and voltage value) is 0. If for some reason the voltage on a channel is greater than the reference voltage or less than 0, then the returned integer value is 65472 or 0 respectively.

**class** adafruit\_mcp3xxx.analog\_in.**AnalogIn** (*mcp, positive\_pin, negative\_pin=None*)  
AnalogIn Mock Implementation for ADC Reads.

**Parameters**

- **mcp** (*MCP3002, 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. Due to 10-bit accuracy of the chip, the returned values range [0, 65472].

**voltage**

Returns the voltage from the ADC pin as a floating point value. Due to the 10-bit accuracy of the chip, returned values range from 0 to (`reference_voltage * 65472 / 65535`)

## 6.2.3 MCP3008

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

- Author(s): Brent Rubell

For proper wiring, please refer to the [Package Types diagram](#) and [Pin Description section](#) of the MCP3004/MCP3008 datasheet.

**class** adafruit\_mcp3xxx.mcp3008.**MCP3008** (*spi\_bus, cs, ref\_voltage=3.3*)  
Bases: `adafruit_mcp3xxx.mcp3xxx.MCP3xxx`

MCP3008 Differential channel mapping. The following list of available differential readings takes the form (`positive_pin, negative_pin`) = (channel A) - (channel B).

- (P0, P1) = CH0 - CH1
- (P1, P0) = CH1 - CH0
- (P2, P3) = CH2 - CH3
- (P3, P2) = CH3 - CH2
- (P4, P5) = CH4 - CH5
- (P5, P4) = CH5 - CH4
- (P6, P7) = CH6 - CH7

- (P7, P6) = CH7 - CH6

See also the warning in the *AnalogIn* class API.

## 6.2.4 MCP3004

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

- Author(s): Brent Rubell

For proper wiring, please refer to [Package Types diagram](#) and [Pin Description section](#) of the MCP3004/MCP3008 datasheet.

**class** adafruit\_mcp3xxx.mcp3004.**MCP3004** (*spi\_bus, cs, ref\_voltage=3.3*)

Bases: *adafruit\_mcp3xxx.mcp3xxx.MCP3xxx*

MCP3004 Differential channel mapping. The following list of available differential readings takes the form (positive\_pin, negative\_pin) = (channel A) - (channel B).

- (P0, P1) = CH0 - CH1
- (P1, P0) = CH1 - CH0
- (P2, P3) = CH2 - CH3
- (P3, P2) = CH3 - CH2

See also the warning in the *AnalogIn* class API.

## 6.2.5 MCP3002

MCP3002 2-channel, 10-bit, analog-to-digital converter instance.

- Author(s): Brent Rubell, Brendan Doherty

For proper wiring, please refer to [Package Type diagram](#) and [Pin Description section](#) of the MCP3002 datasheet.

**class** adafruit\_mcp3xxx.mcp3002.**MCP3002** (*spi\_bus, cs, ref\_voltage=3.3*)

Bases: *adafruit\_mcp3xxx.mcp3xxx.MCP3xxx*

MCP3002 Differential channel mapping. The following list of available differential readings takes the form (positive\_pin, negative\_pin) = (channel A) - (channel B).

- (P0, P1) = CH0 - CH1
- (P1, P0) = CH1 - CH0

See also the warning in the *AnalogIn* class API.



# CHAPTER 7

---

## Indices and tables

---

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



**a**

adafruit\_mcp3xxx.analog\_in, 19  
adafruit\_mcp3xxx.mcp3002, 20  
adafruit\_mcp3xxx.mcp3004, 20  
adafruit\_mcp3xxx.mcp3008, 19  
adafruit\_mcp3xxx.mcp3xxx, 18



## A

`adafruit_mcp3xxx.analog_in` (*module*), 19  
`adafruit_mcp3xxx.mcp3002` (*module*), 20  
`adafruit_mcp3xxx.mcp3004` (*module*), 20  
`adafruit_mcp3xxx.mcp3008` (*module*), 19  
`adafruit_mcp3xxx.mcp3xxx` (*module*), 18  
`AnalogIn` (*class in adafruit\_mcp3xxx.analog\_in*), 19

## M

`MCP3002` (*class in adafruit\_mcp3xxx.mcp3002*), 20  
`MCP3004` (*class in adafruit\_mcp3xxx.mcp3004*), 20  
`MCP3008` (*class in adafruit\_mcp3xxx.mcp3008*), 19  
`MCP3xxx` (*class in adafruit\_mcp3xxx.mcp3xxx*), 18

## R

`read()` (*adafruit\_mcp3xxx.mcp3xxx.MCP3xxx*  
*method*), 18  
`reference_voltage`  
(*adafruit\_mcp3xxx.mcp3xxx.MCP3xxx* *at-*  
*tribute*), 18

## V

`value` (*adafruit\_mcp3xxx.analog\_in.AnalogIn* *at-*  
*tribute*), 19  
`voltage` (*adafruit\_mcp3xxx.analog\_in.AnalogIn*  
*attribute*), 19