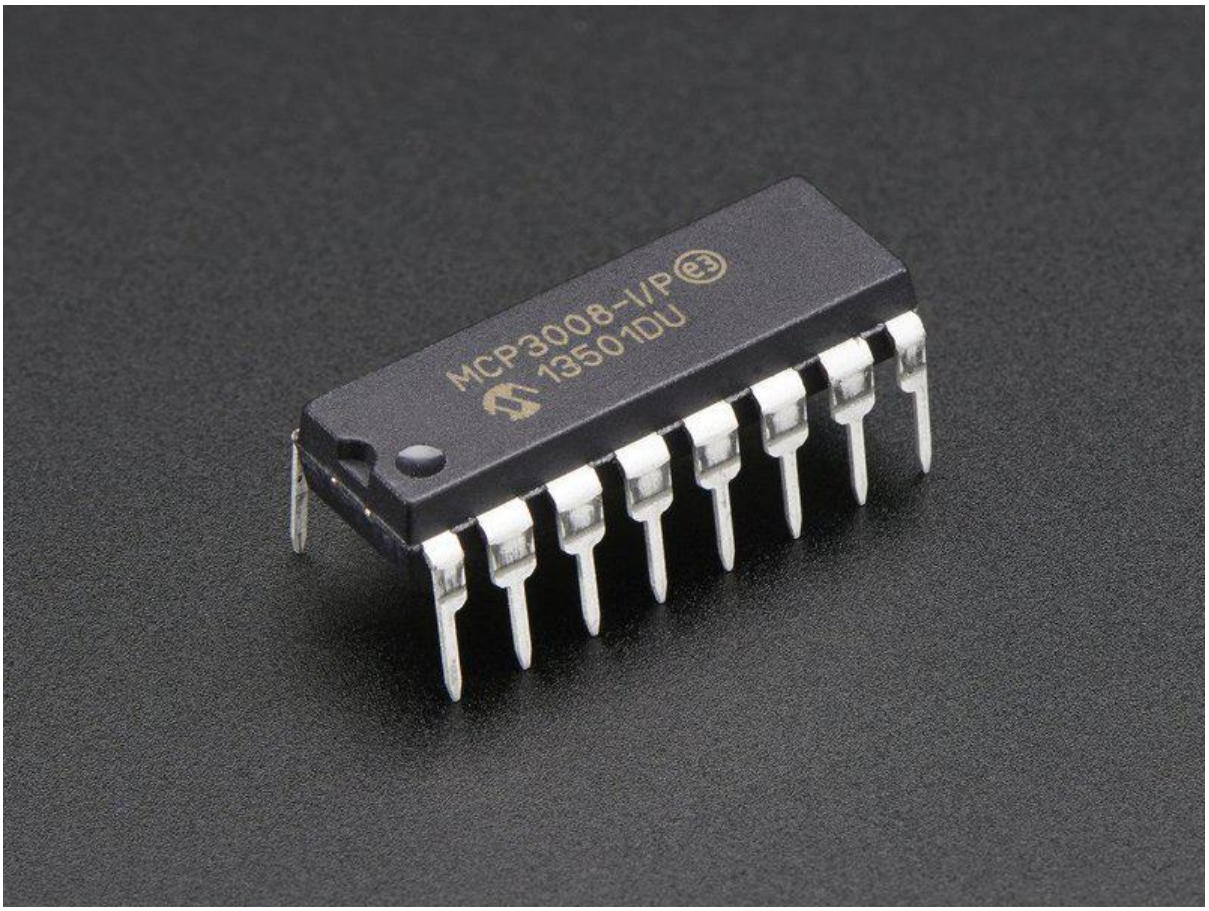
MCP3008 - 8-Channel 10-Bit ADC With SPI Interface

Created by Kattni Rembor

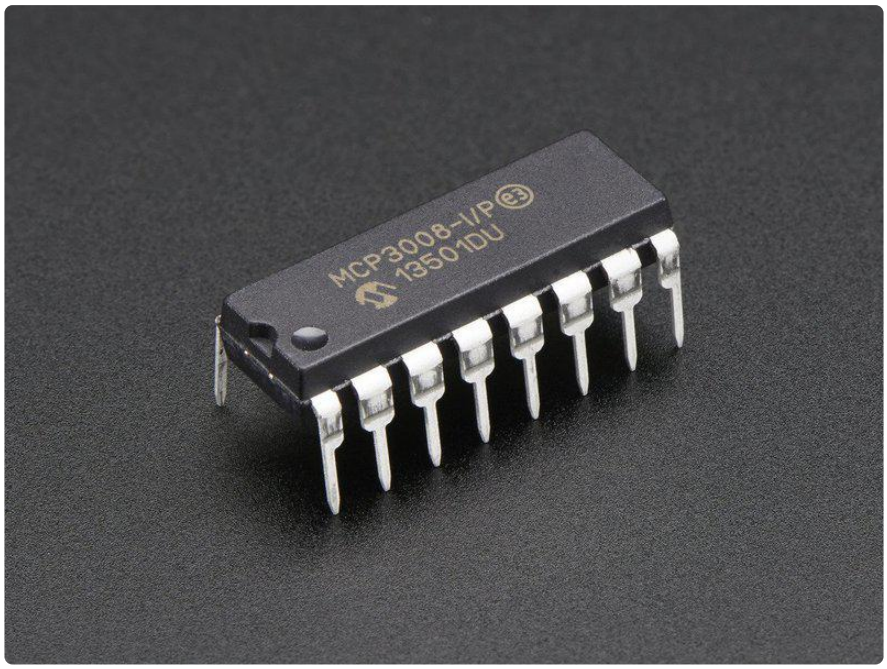


https://learn.adafruit.com/mcp3008-spi-adc

Last updated on 2022-12-01 03:22:43 PM EST

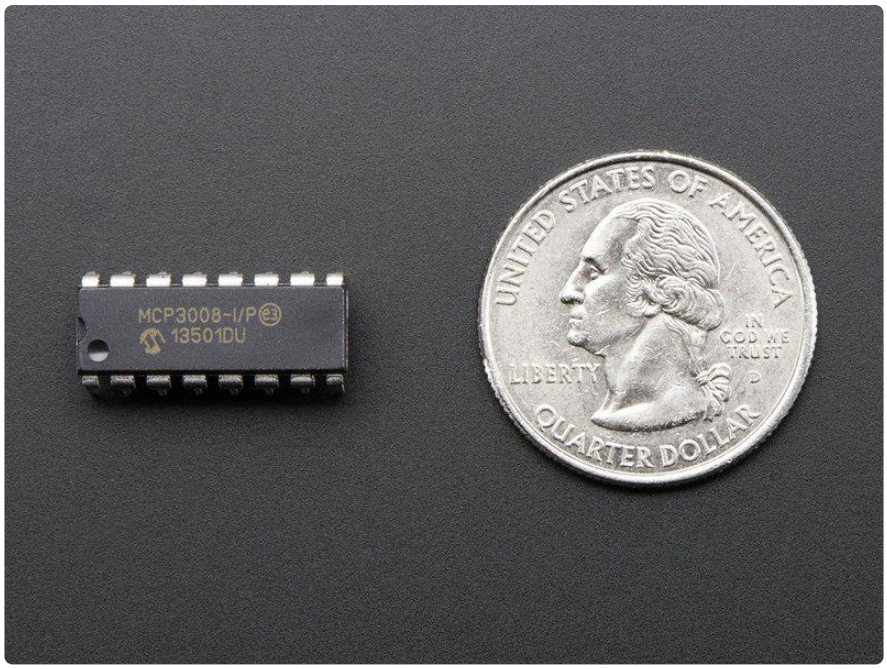
Table of Contents

Overview



Need to add analog inputs? This chip will add 8 channels of 10-bit analog input to your microcontroller or microcomputer project. It's super easy to use, and uses SPI so only 4 pins are required.

This chip is useful for situations needing 8 analog inputs that don't need to be very fast, especially for Linux computers that don't have analog inputs. This is the cheapest way to add 8 analog inputs and because it's SPI, you only need 4 wires. And, you get good, quality readings. And it's through-hole which is super great for quick prototyping!



Python & CircuitPython

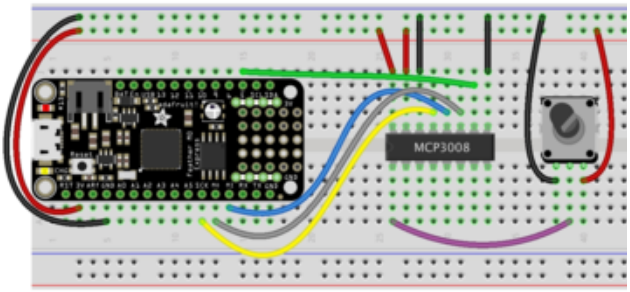
It's easy to use an MCP3008 8-channel ADC with Python or CircuitPython and the Adafruit CircuitPython MCP3xxx () module. This module allows you to easily write Python code to add extra digital inputs and outputs.

You can use this ADC with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library ().

CircuitPython Microcontroller Wiring

Connect your MCP3008 to your CircuitPython board using a standard SPI connection.

Here's an example of wiring a MCP3008 to a Feather M0 board:

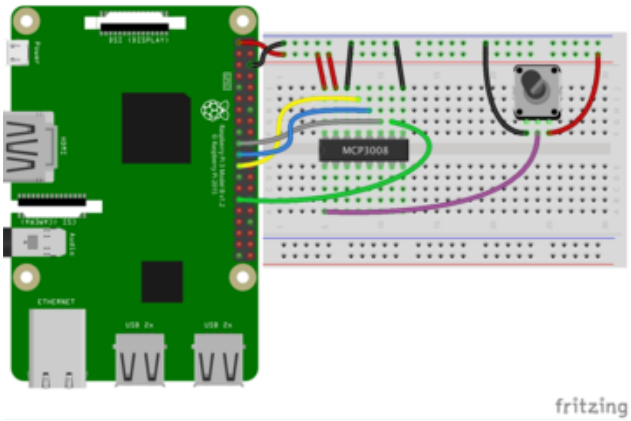


MCP3008 CLK to Feather M0 SCK
MCP3008 DOUT to Feather M0 MISO
MCP3008 DIN to Feather M0 MOSI
MCP3008 CS to Feather M0 D5
MCP3008 VDD to Feather M0 3V
MCP3008 VREF to Feather M0 3V
MCP3008 AGND to Feather M0 GND
MCP3008 DGND to Feather M0 GND
MCP3008 CH0 to Potentiometer middle pin
Potentiometer left pin to Feather M0 GND
Potentiometer right pin to Feather M0 3V

Python Computer Wiring

Since there's dozens of Linux computers/boards you can use we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported ().

Here's the Raspberry Pi wired to the MCP3008:



MCP3008 CLK to Pi SCLK
MCP3008 DOUT to Pi MISO
MCP3008 DIN to Pi MOSI
MCP3008 CS to Pi D5
MCP3008 VDD to Pi 3.3V
MCP3008 VREF to Pi 3.3V
MCP3008 AGND to Pi GND
MCP3008 DGND to Pi GND
MCP3008 CH0 to Potentiometer middle pin
Potentiometer left pin to Pi GND
Potentiometer right pin to Pi 3.3V

CircuitPython Installation of MCP3xxx Library

You'll need to install the Adafruit CircuitPython MCP3xxx () library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython () for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (). Our CircuitPython starter guide has a great page on how to install the library bundle ().

For non-express boards like the Trinket M0 or Gemma M0, you'll need to manually install the necessary libraries from the bundle:

- adafruit_mcp3xxx
- adafruit_bus_device

Before continuing make sure your board's lib folder or root filesystem has the adafruit_mcp3xxx and adafruit_bus_device files and folders copied over.

Next connect to the board's serial REPL () so you are at the CircuitPython >>> prompt.

Python Installation of MCP3xxx Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready ()!

Once that's done, from your command line run the following command:

- `sudo pip3 install adafruit-circuitpython-mcp3xxx`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

To demonstrate the usage of the device we'll initialize it and read the analog inputs from the Python REPL.

Run the following code to import the necessary modules, initialize the SPI connection, assign a chip select pin, and create the MCP3008 object:

```
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
cs = digitalio.DigitalInOut(board.D5)
mcp = MCP.MCP3008(spi, cs)
```

Next, we'll create an analog input channel on the MCP3008 pin 0:

```
channel = AnalogIn(mcp, MCP.P0)
```

Now you're ready to read the raw ADC value and the channel voltage with the following properties:

- value - Returns the value of an ADC pin as an integer.
- voltage - Returns the voltage from the ADC pin as a floating point value, scaled 16 bits to remain consistent with other ADCs.

For example, to print the raw ADC value and the channel voltage, run the following:

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

```
>>> print('Raw ADC Value: ', channel.value)
Raw ADC Value:  3392
>>> print('ADC Voltage: ' + str(channel.voltage) + 'V')
ADC Voltage: 0.170803V
```

You can run the same code in a loop. Then try rotating the potentiometer to see the values change.

```
import time
while True:
    print('Raw ADC Value: ', channel.value)
    print('ADC Voltage: ' + str(channel.voltage) + 'V')
    time.sleep(0.5)
```

```
>>> while True:
...     print('Raw ADC Value: ', channel.value)
...     print('ADC Voltage: ' + str(channel.voltage) + 'V')
...     time.sleep(0.5)
...
Raw ADC Value:  64
ADC Voltage: 0.0032227V
Raw ADC Value:  768
ADC Voltage: 0.0483406V
Raw ADC Value:  14208
ADC Voltage: 0.725108V
Raw ADC Value:  35392
ADC Voltage: 1.78538V
Raw ADC Value:  52352
ADC Voltage: 2.62973V
Raw ADC Value:  62592
ADC Voltage: 3.1518V
Raw ADC Value:  65344
ADC Voltage: 3.28716V
Raw ADC Value:  65088
ADC Voltage: 3.27104V
```

Even though the MCP3008 is a 10-bit ADC, the value returned is a 16-bit number to provide a consistent interface across ADCs in CircuitPython

That's all there is to getting started with the MCP3008 and CircuitPython!

Full Example Code

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

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")
```

Python Docs

Python Docs ()