
Adafruit MCP9808 Library Documentation

Release 1.0

Phillip Moyer

Apr 26, 2021

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Notes	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_mcp9808	14
6.2.1	Implementation Notes	14
7	Indices and tables	15
	Python Module Index	17
	Index	19

The MCP9808 is an awesome, high accuracy temperature sensor that communicates over I2C. Its available on [Adafruit](#) as a [breakout](#).

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-mcp9808
```

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

```
sudo pip3 install adafruit-circuitpython-mcp9808
```

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-mcp9808
```


CHAPTER 3

Usage Notes

Getting the temperature in Celsius is easy! First, import all of the pins from the board, `board.I2C()` for native I2C communication and the thermometer library itself.

```
from board import *  
import adafruit_mcp9808
```

Next, initialize the I2C bus in a `with` statement so it always gets shut down ok. Then, construct the thermometer class:

```
# Do one reading  
with i2c = board.I2C() as i2c:  
    t = adafruit_mcp9808.MCP9808(i2c)  
  
    # Finally, read the temperature property and print it out  
    print(t.temperature)
```


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/mcp9808_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6  import adafruit_mcp9808
7
8  i2c = board.I2C()  # uses board.SCL and board.SDA
9
10 # To initialise using the default address:
11 mcp = adafruit_mcp9808.MCP9808(i2c)
12
13 # To initialise using a specified address:
14 # Necessary when, for example, connecting A0 to VDD to make address=0x19
15 # mcp = adafruit_mcp9808.MCP9808(i2c_bus, address=0x19)
16
17
18 while True:
19     tempC = mcp.temperature
20     tempF = tempC * 9 / 5 + 32
21     print("Temperature: {} C {} F ".format(tempC, tempF))
22     time.sleep(2)
```

6.2 adafruit_mcp9808

CircuitPython library to support MCP9808 high accuracy temperature sensor.

- Author(s): Scott Shawcroft

6.2.1 Implementation Notes

Hardware:

- Adafruit [MCP9808 High Accuracy I2C Temperature Sensor Breakout](#) (Product ID: 1782)

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

Notes:

1. Datasheet: <http://www.adafruit.com/datasheets/MCP9808.pdf>

class `adafruit_mcp9808.MCP9808` (*i2c_bus*, *address=24*)
Interface to the MCP9808 temperature sensor.

Parameters

- **i2c_bus** (*I2C*) – The I2C bus the MCP9808 is connected to.
- **address** (*int*) – The I2C address of the device. Defaults to 0x18

Quickstart: Importing and using the MCP9808

Here is an example of using the `MCP9808` class. First you will need to import the libraries to use the sensor

```
import board
import adafruit_mcp9808
```

Once this is done you can define your `board.I2C` object and define your sensor object

```
i2c = board.I2C() # uses board.SCL and board.SDA
mcp = adafruit_mcp9808.MCP9808(i2c_bus)
```

Now you have access to the change in temperature using the `temperature` attribute. This temperature is in Celsius.

```
temperature = mcp.temperature
```

temperature

Temperature in Celsius. Read-only.

CHAPTER 7

Indices and tables

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

Python Module Index

a

adafruit_mcp9808, [13](#)

A

`adafruit_mcp9808` (*module*), [13](#)

M

`MCP9808` (*class in adafruit_mcp9808*), [14](#)

T

`temperature` (*adafruit_mcp9808.MCP9808* *attribute*), [14](#)