# Adafruit MCP9808 Library Documentation

*Release 1.0*

**Phiilip Moyer**

**Jan 22, 2020**

# Contents

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, busio for native I2C communication and the thermometer library itself.

```
from board import *
import busio
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 busio.I2C(SCL, SDA) 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.

CHAPTER 6

# Table of Contents

## 6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/mcp9808_simpletest.py

```
import time
import board
import busio
import adafruit_mcp9808

i2c_bus = busio.I2C(board.SCL, board.SDA)

# To initialise using the default address:
mcp = adafruit_mcp9808.MCP9808(i2c_bus)

# To initialise using a specified address:
# Necessary when, for example, connecting A0 to VDD to make address=0x19
# mcp = adafruit_mcp9808.MCP9808(i2c_bus, address=0x19)



while True:
    tempC = mcp.temperature
    tempF = tempC * 9 / 5 + 32
    print('Temperature: {} C {} F '.format(tempC, tempF))
    time.sleep(2)
```

## 6.2 `adafruit_mcp9808` - MCP9808 I2C Temperature Sensor

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 (0.8.0+) 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

**Notes:**

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

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

**temperature**
Temperature in celsius. Read-only.

CHAPTER 7

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## a

# Index