

---

# **AdafruitMCP9600 Library Documentation**

*Release 1.0*

**Dan Cogliano**

**Jun 07, 2021**



---

## 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>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple test . . . . .	13
6.2	adafruit_mcp9600 . . . . .	13
6.2.1	Implementation Notes . . . . .	14
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



This is a CircuitPython driver for the MCP9600 thermocouple I2C amplifier. In addition to the MCP9600 breakout, you will also need a thermocouple, which can be found in the Adafruit store. The MCP9600 supports several thermocouple types for different temperature ranges. The “K” type is the default, with a range of -200C to +1372C.



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

- [Adafruit CircuitPython](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the [Adafruit library and driver bundle](#).



---

### 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-mcp9600
```

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

```
sudo pip3 install adafruit-circuitpython-mcp9600
```

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-mcp9600
```



## CHAPTER 3

---

### Usage Example

---

This is a simple example showing the hot junction temperature (the temperature at the tip of the thermocouple). You may need to adjust the I2C frequency if you receive input/output errors.

```
import board
import busio
from adafruit_bus_device.i2c_device import I2CDevice
from adafruit_mcp9600 import MCP9600

i2c = busio.I2C(board.SCL, board.SDA, frequency=200000)
try:
    # using default I2C register and "K" thermocouple
    device = MCP9600(i2c)
    print("temperature(C):", device.temperature)
except ValueError:
    print("MCP9600 sensor not detected")
```

This example displays the ambient/room and hot junction temperatures at 1 second intervals. Turn on the Mu editor's plotter option to view the temperatures in a real-time graph.

```
import board
import busio
import time
from adafruit_bus_device.i2c_device import I2CDevice
from adafruit_mcp9600 import MCP9600

i2c = busio.I2C(board.SCL, board.SDA, frequency=200000)

try:
    device = MCP9600(i2c)
    print("version:", device.version)
    while True:
        print((
            device.ambient_temperature,
            device.temperature
```

(continues on next page)

(continued from previous page)

```
    ))  
    time.sleep(1)  
except ValueError:  
    print("MCP9600 sensor not detected")
```

## 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/mcp9600\_simpletest.py

```
1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 import time
5 import board
6 import busio
7 import adafruit_mcp9600
8
9 # frequency must be set for the MCP9600 to function.
10 # If you experience I/O errors, try changing the frequency.
11 i2c = busio.I2C(board.SCL, board.SDA, frequency=100000)
12 mcp = adafruit_mcp9600.MCP9600(i2c)
13
14 while True:
15     print((mcp.ambient_temperature, mcp.temperature, mcp.delta_temperature))
16     time.sleep(1)
```

## 6.2 adafruit\_mcp9600

CircuitPython driver for the MCP9600 thermocouple I2C amplifier

- Author(s): Dan Cogliano, Kattni Rembor

## 6.2.1 Implementation Notes

### Hardware:

- Adafruit MCP9600 I2C Thermocouple Amplifier: (Product ID: 4101)

### 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)

**class** `adafruit_mcp9600.MCP9600` (*i2c*, *address=103*, *tctype='K'*, *tcfilter=0*)  
 Interface to the MCP9600 thermocouple amplifier breakout

#### Parameters

- **i2c** (*I2C*) – The I2C bus the MCP9600 is connected to.
- **address** (*int*) – The I2C address of the device. Defaults to 0x67
- **tctype** (*str*) – Thermocouple type. Defaults to "K"
- **tcfilter** (*int*) – Value for the temperature filter. Can limit spikes in temperature readings. Defaults to 0

### Quickstart: Importing and using the MCP9600 temperature sensor

Here is one way of importing the `MCP9600` class so you can use it with the name `mcp`. First you will need to import the libraries to use the sensor

```
import busio
import board
import adafruit_mcp9600
```

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

```
i2c = busio.I2C(board.SCL, board.SDA, frequency=100000)
mcp = adafruit_mcp9600.MCP9600(i2c)
```

Now you have access to the change in temperature using the `delta_temperature` attribute, the thermocouple or hot junction temperature in degrees Celsius using the `temperature` attribute and the ambient or cold-junction temperature in degrees Celsius using the `ambient_temperature` attribute

```
delta_temperature = mcp.delta_temperature
temperature = mcp.temperature
ambient_temperature = mcp.ambient_temperature
```

**alert\_1**  
Alert 1 status.

**alert\_2**  
Alert 2 status.

**alert\_3**  
Alert 3 status.

**alert\_4**  
Alert 4 status.

**alert\_config**(\*, *alert\_number*, *alert\_temp\_source*, *alert\_temp\_limit*, *alert\_hysteresis*,  
*alert\_temp\_direction*, *alert\_mode*, *alert\_state*)

Configure a specified alert pin. Alert is enabled by default when alert is configured. To disable an alert pin, use `alert_disable()`.

#### Parameters

- **alert\_number** (*int*) – The alert pin number. Must be 1-4.
- **alert\_temp\_source** – The temperature source to monitor for the alert. Options are: THERMOCOUPLE (hot-junction) or AMBIENT (cold-junction). Temperatures are in Celsius.
- **alert\_temp\_limit** (*float*) – The temperature in degrees Celsius at which the alert should trigger. For rising temperatures, the alert will trigger when the temperature rises above this limit. For falling temperatures, the alert will trigger when the temperature falls below this limit.
- **alert\_hysteresis** (*float*) – The alert hysteresis range. Must be 0-255 degrees Celsius. For rising temperatures, the hysteresis is below alert limit. For falling temperatures, the hysteresis is above alert limit. See data-sheet for further information.
- **alert\_temp\_direction** – The direction the temperature must change to trigger the alert. Options are RISING (heating up) or FALLING (cooling down).
- **alert\_mode** – The alert mode. Options are COMPARATOR or INTERRUPT. In comparator mode, the pin will follow the alert, so if the temperature drops, for example, the alert pin will go back low. In interrupt mode, by comparison, once the alert goes off, you must manually clear it. If setting mode to INTERRUPT, use `alert_interrupt_clear()` to clear the interrupt flag.
- **alert\_state** – Alert pin output state. Options are ACTIVE\_HIGH or ACTIVE\_LOW.

For example, to configure alert 1:

```
import board
import busio
import digitalio
import adafruit_mcp9600

i2c = busio.I2C(board.SCL, board.SDA, frequency=100000)
mcp = adafruit_mcp9600.MCP9600(i2c)
alert_1 = digitalio.DigitalInOut(board.D5)
alert_1.switch_to_input()

mcp.alert_config(alert_number=1, alert_temp_source=mcp.THERMOCOUPLE,
                 alert_temp_limit=25, alert_hysteresis=0,
                 alert_temp_direction=mcp.RISING, alert_mode=mcp.COMPARATOR,
                 alert_state=mcp.ACTIVE_LOW)
```

**alert\_disable** (*alert\_number*)

Configuring an alert using `alert_config()` enables the specified alert by default. Use `alert_disable()` to disable an alert pin.

**Parameters** **alert\_number** (*int*) – The alert pin number. Must be 1-4.

**alert\_interrupt\_clear** (*alert\_number*, *interrupt\_clear=True*)

Turns off the alert flag in the MCP9600, and clears the pin state (not used if the alert is in comparator mode). Required when `alert_mode` is INTERRUPT.

#### Parameters

- **alert\_number** (*int*) – The alert pin number. Must be 1-4.
- **interrupt\_clear** (*bool*) – The bit to write the interrupt state flag

**ambient\_resolution**

Ambient (cold-junction) temperature resolution. Options are `AMBIENT_RESOLUTION_0_0625` (0.0625 degrees Celsius) or `AMBIENT_RESOLUTION_0_25` (0.25 degrees Celsius).

**ambient\_temperature**

Cold junction/ambient/room temperature in Celsius

**burst\_complete**

Burst complete.

**burst\_mode\_samples**

The number of samples taken during a burst in burst mode. Options are `BURST_SAMPLES_1`, `BURST_SAMPLES_2`, `BURST_SAMPLES_4`, `BURST_SAMPLES_8`, `BURST_SAMPLES_16`, `BURST_SAMPLES_32`, `BURST_SAMPLES_64`, `BURST_SAMPLES_128`.

**delta\_temperature**

Delta temperature in Celsius

**input\_range**

Input range.

**shutdown\_mode**

Shutdown modes. Options are `NORMAL`, `SHUTDOWN`, and `BURST`.

**temperature**

Hot junction temperature in Celsius

**temperature\_update**

Temperature update.

**version**

MCP9600 chip version

# CHAPTER 7

---

## Indices and tables

---

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



**a**

`adafruit_mcp9600`, 13



## A

adafruit\_mcp9600 (*module*), 13  
alert\_1 (*adafruit\_mcp9600.MCP9600 attribute*), 14  
alert\_2 (*adafruit\_mcp9600.MCP9600 attribute*), 14  
alert\_3 (*adafruit\_mcp9600.MCP9600 attribute*), 14  
alert\_4 (*adafruit\_mcp9600.MCP9600 attribute*), 14  
alert\_config() (*adafruit\_mcp9600.MCP9600 method*), 14  
alert\_disable() (*adafruit\_mcp9600.MCP9600 method*), 15  
alert\_interrupt\_clear() (*adafruit\_mcp9600.MCP9600 method*), 15  
ambient\_resolution (*adafruit\_mcp9600.MCP9600 attribute*), 16  
ambient\_temperature (*adafruit\_mcp9600.MCP9600 attribute*), 16

## B

burst\_complete (*adafruit\_mcp9600.MCP9600 attribute*), 16  
burst\_mode\_samples (*adafruit\_mcp9600.MCP9600 attribute*), 16

## D

delta\_temperature (*adafruit\_mcp9600.MCP9600 attribute*), 16

## I

input\_range (*adafruit\_mcp9600.MCP9600 attribute*), 16

## M

MCP9600 (*class in adafruit\_mcp9600*), 14

## S

shutdown\_mode (*adafruit\_mcp9600.MCP9600 attribute*), 16

## T

temperature (*adafruit\_mcp9600.MCP9600 attribute*), 16  
temperature\_update (*adafruit\_mcp9600.MCP9600 attribute*), 16

## V

version (*adafruit\_mcp9600.MCP9600 attribute*), 16