
AdafruitMCP9600 Library Documentation

Release 1.0

Dan Cogliano

Apr 10, 2020

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_mcp9600	13
6.2.1	Implementation Notes	13
7	Indices and tables	17
	Python Module Index	19
	Index	21

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](#).

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-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 import time
2 import board
3 import busio
4 import adafruit_mcp9600
5
6 # frequency must be set for the MCP9600 to function.
7 # If you experience I/O errors, try changing the frequency.
8 i2c = busio.I2C(board.SCL, board.SDA, frequency=100000)
9 mcp = adafruit_mcp9600.MCP9600(i2c)
10
11 while True:
12     print((mcp.ambient_temperature, mcp.temperature, mcp.delta_temperature))
13     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: <https://www.adafruit.com/product/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

class adafruit_mcp9600.**MCP9600** (*i2c*, *address=103*, *tctype='K'*, *tcfilter=0*)

Interface to the MCP9600 thermocouple amplifier breakout

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)
```

(continues on next page)

(continued from previous page)

```

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`

Python Module Index

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*), 15
ambient_temperature (*adafruit_mcp9600.MCP9600 attribute*), 15

B

burst_complete (*adafruit_mcp9600.MCP9600 attribute*), 15
burst_mode_samples (*adafruit_mcp9600.MCP9600 attribute*), 15

D

delta_temperature (*adafruit_mcp9600.MCP9600 attribute*), 15

I

input_range (*adafruit_mcp9600.MCP9600 attribute*), 15

M

MCP9600 (*class in adafruit_mcp9600*), 14

S

shutdown_mode (*adafruit_mcp9600.MCP9600 attribute*), 15

T

temperature (*adafruit_mcp9600.MCP9600 attribute*), 15
temperature_update (*adafruit_mcp9600.MCP9600 attribute*), 15

V

version (*adafruit_mcp9600.MCP9600 attribute*), 15