
AdafruitMAX9744 Library Documentation

Release 1.0

Tony DiCola

Aug 25, 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_max9744	14
6.2.1	Implementation Notes	14
7	Indices and tables	15
	Python Module Index	17
	Index	19

CircuitPython module for the MAX9744 20W class D amplifier.

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-max9744
```

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

```
sudo pip3 install adafruit-circuitpython-max9744
```

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-max9744
```


CHAPTER 3

Usage Example

See `examples/max9744_simpletest.py` for a demo of the usage.

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/mas9744_simpletest.py

```
1  # Simple demo of the MAX9744 20W class D amplifier I2C control.
2  # This show how to set the volume of the amplifier.
3  # Author: Tony DiCola
4  import board
5  import busio
6
7  import adafruit_max9744
8
9
10 # Initialize I2C bus.
11 i2c = busio.I2C(board.SCL, board.SDA)
12
13 # Initialize amplifier.
14 amp = adafruit_max9744.MAX9744(i2c)
15 # Optionally you can specify a different addres if you override the AD1, AD2
16 # pins to change the address.
17 # amp = adafruit_max9744.MAX9744(i2c, address=0x49)
18
19 # Setting the volume is as easy as writing to the volume property (note
20 # you cannot read the property so keep track of volume in your own code if
21 # you need it).
22 amp.volume = 31 # Volume is a value from 0 to 63 where 0 is muted/off and
23 # 63 is maximum volume.
24
25 # In addition you can call a function to instruct the amp to move up or down
26 # a single volume level. This is handy if you just have up/down buttons in
27 # your project for volume:
```

(continues on next page)

(continued from previous page)

```
28 amp.volume_up() # Increase volume by one level.  
29  
30 amp.volume_down() # Decrease volume by one level.
```

6.2 adafruit_max9744

CircuitPython module for the MAX9744 20W class D amplifier. See examples/simpletest.py for a demo of the usage.

- Author(s): Tony DiCola

6.2.1 Implementation Notes

Hardware:

- Adafruit [MAX9744 Stereo 20W Class D Audio Amplifier](#) (Product ID: 1752)

Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based boards: <https://github.com/adafruit/circuitpython/releases>

class adafruit_max9744.**MAX9744** (*i2c*, *, *address*=75)
MAX9744 20 watt class D amplifier.

Parameters

- **i2c** – The I2C bus for the device.
- **address** – (Optional) The address of the device if it has been overridden from the default with the AD1, AD2 pins.

volume_down ()
Decrease the volume by one level.

volume_up ()
Increase the volume by one level.

CHAPTER 7

Indices and tables

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

a

adafruit_max9744, [14](#)

A

`adafruit_max9744` (*module*), [14](#)

M

`MAX9744` (*class in adafruit_max9744*), [14](#)

V

`volume_down()` (*adafruit_max9744.MAX9744 method*), [14](#)

`volume_up()` (*adafruit_max9744.MAX9744 method*), [14](#)