

---

# **Adafruit-Motor-Hat Documentation**

***Release 0.2.0***

**Mark Omo**

**Apr 01, 2018**



---

## Contents

---

|          |                                                                      |           |
|----------|----------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Adafruit-Motor-Hat</b>                                            | <b>3</b>  |
| 1.1      | Features . . . . .                                                   | 3         |
| 1.2      | Credits . . . . .                                                    | 3         |
| <b>2</b> | <b>Installation</b>                                                  | <b>5</b>  |
| 2.1      | Stable release . . . . .                                             | 5         |
| 2.2      | From sources . . . . .                                               | 5         |
| <b>3</b> | <b>Usage</b>                                                         | <b>7</b>  |
| <b>4</b> | <b>Reference</b>                                                     | <b>9</b>  |
| <b>5</b> | <b>Contributing</b>                                                  | <b>13</b> |
| 5.1      | Types of Contributions . . . . .                                     | 13        |
| 5.2      | Get Started! . . . . .                                               | 14        |
| 5.3      | Pull Request Guidelines . . . . .                                    | 15        |
| 5.4      | Tips . . . . .                                                       | 15        |
| <b>6</b> | <b>Credits</b>                                                       | <b>17</b> |
| 6.1      | Development Lead . . . . .                                           | 17        |
| 6.2      | Contributors . . . . .                                               | 17        |
| 6.3      | Original Contributors to Adafruit-Motor-HAT-Python-Library . . . . . | 17        |
| <b>7</b> | <b>History</b>                                                       | <b>19</b> |
| 7.1      | 0.1.0 (2017-06-24) . . . . .                                         | 19        |
| <b>8</b> | <b>Indices and tables</b>                                            | <b>21</b> |
|          | <b>Python Module Index</b>                                           | <b>23</b> |



Contents:



An improved pip available Adafruit Motor Hat Package

- Free software: MIT license
- Documentation: <https://adafruit-motor-hat.readthedocs.io>.

## 1.1 Features

- Improve the Stepper performance for the original Adafruit-Motor-HAT-Python-Library

## 1.2 Credits

This package is based and mostly contains code from Adafruit-Motor-HAT-Python-Library

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.





### 2.1 Stable release

To install Adafruit-Motor-Hat, run this command in your terminal:

```
$ pip install adafruit_motor_hat
```

This is the preferred method to install Adafruit-Motor-Hat, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

### 2.2 From sources

The sources for Adafruit-Motor-Hat can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/ferret-guy/adafruit_motor_hat
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/ferret-guy/adafruit_motor_hat/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```



## CHAPTER 3

---

### Usage

---

To use Adafruit-Motor-Hat in a project:

```
import adafruit_motor_hat
```



**class** adafruit\_motor\_hat.MotorHat.**AdafruitStepperMotor** (*controller, num, steps=200*)

**align\_step** (*reverse=False*)

Align the current step position with a full step

**Parameters** **reverse** (*bool*) – run in reverse

**Returns** None

**double\_step** (*steps=1, reverse=False*)

Double stepping, 2 coils at once

**Parameters**

- **steps** (*int*) – The number of steps to execute
- **reverse** (*bool*) – step in reverse

**Returns** None

**interleaved\_step** (*steps=1, reverse=False*)

Interleaved stepping, single stepping and interleaved stepping

**Parameters**

- **steps** (*int*) – The number of steps to execute
- **reverse** (*bool*) – step in reverse

**Returns** None

**micro\_step** (*steps=1, reverse=False*)

Microstepping

**Parameters**

- **steps** (*int*) – The number of steps to execute
- **reverse** (*bool*) – step in reverse

**Returns** None

**set\_speed** (*rpm*)

Set the target motor speed in rpm, motor step count is used to calculate the step delay

**Parameters** *rpm* (*float*) – Target RPM

**Returns** None

**step** (*steps=1*, *reverse=False*)

Simple single step

**Parameters**

- **steps** (*int*) – The number of steps to execute
- **reverse** (*Bool*) – step in reverse

**Returns** None

**class** adafruit\_motor\_hat.MotorHat.**AdafruitDCMotor** (*controller*, *num*)

**class** adafruit\_motor\_hat.MotorHat.**AdafruitMotorHAT** (*addr=96*, *freq=1600*, *i2c=None*,  
*i2c\_bus=None*)

**get\_motor** (*num*)

Get a motor instance

**Parameters** *num* (*int*) – the stepper to get 1-4

**Returns** :class:‘AdafruitDCMotor’

**get\_stepper** (*num*)

Get a stepper instance

**Parameters** *num* (*int*) – the stepper to get 1, or 2

**Returns** :class:‘AdafruitStepperMotor’

**set\_pin** (*pin*, *value*)

Set a pin output state

**Parameters**

- **pin** (*int*) – pin number
- **value** (*int*) – pin state 0, or 1

**Returns** None

**class** adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver.**PWM** (*address=64*, *debug=False*,  
*i2c=None*, *i2c\_bus=None*)

Adafruit PCA9685 16-Channel PWM Servo Driver

**setAllPWM** (*on*, *off*)

Sets all the pwm outputs to the corresponding outputs

**Parameters**

- **on** – the portion of the PWM period in 1/4096ths of a period (0 corresponds to the start, 4096 corresponds to the end) to wait before the signal turns on
- **off** – the portion of the PWM period in 1/4096ths of a period (0 corresponds to the start, 4096 corresponds to the end) to wait before the signal turns off

**Returns** None

**setPWM** (*channel*, *on*, *off*)

Set a single pwm channel

**Parameters**

- **channel** – the channel number from 0-15 inclusive
- **on** – the portion of the PWM period in 1/4096ths of a period (0 corresponds to the start, 4096 corresponds to the end) to wait before the signal turns on
- **off** – the portion of the PWM period in 1/4096ths of a period (0 corresponds to the start, 4096 corresponds to the end) to wait before the signal turns off

**Returns** None

**setPWMFreq** (*freq*, *correctionFactor*=1.0)

Set the PWM frequency in Hz

**Parameters**

- **freq** (*int*) – frequency in Hz
- **correctionFactor** (*float*) – correction factor for drifting pwm putout

**Returns** None

**classmethod softwareReset** (*i2c*=None, *i2c\_bus*=None)

Sends a software reset (SWRST) command to all the servo drivers on the bus

`adafruit_motor_hat.Adafruit_PWM_Servo_Driver.get_i2c_device` (*address*, *i2c*,  
*i2c\_bus*)

Helper method to get a device at the specified address from the I2C bus. If no i2c bus is specified (i2c param is None) then the default I2C bus for the platform will be used.

**Parameters**

- **address** (*hex*) – i2c address to get the handle for
- **i2c** – GPIO i2c class, None to use the Adafruit\_GPIO.I2C class
- **i2c\_bus** – i2c bus number, passed to busnum, none to autodetect

**Returns** i2c Address





Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

## 5.1 Types of Contributions

### 5.1.1 Report Bugs

Report bugs at [https://github.com/ferret-guy/adafruit\\_motor\\_hat/issues](https://github.com/ferret-guy/adafruit_motor_hat/issues).

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### 5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

### 5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

### 5.1.4 Write Documentation

Adafruit-Motor-Hat could always use more documentation, whether as part of the official Adafruit-Motor-Hat docs, in docstrings, or even on the web in blog posts, articles, and such.

### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at [https://github.com/ferret-guy/adafruit\\_motor\\_hat/issues](https://github.com/ferret-guy/adafruit_motor_hat/issues).

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 5.2 Get Started!

Ready to contribute? Here's how to set up *adafruit\_motor\_hat* for local development.

1. Fork the *adafruit\_motor\_hat* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/adafruit_motor_hat.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv adafruit_motor_hat
$ cd adafruit_motor_hat/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 adafruit_motor_hat tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check [https://travis-ci.org/ferret-guy/adafruit\\_motor\\_hat/pull\\_requests](https://travis-ci.org/ferret-guy/adafruit_motor_hat/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_adafruit_motor_hat
```



### 6.1 Development Lead

- Mark Omo <[mark@markomo.me](mailto:mark@markomo.me)>

### 6.2 Contributors

None yet. Why not be the first?

### 6.3 Original Contributors to Adafruit-Motor-HAT-Python-Library

- [tdicola](#)
- [caternuson](#)
- [ladyada](#)



#### 7.1 0.1.0 (2017-06-24)

- First release on PyPI.





## CHAPTER 8

---

### Indices and tables

---

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



### a

`adafruit_motor_hat.Adafruit_PWM_Servo_Driver,`  
[10](#)



## A

adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver (module), 10

AdafruitDCMotor (class in adafruit\_motor\_hat.MotorHat), 10

AdafruitMotorHAT (class in adafruit\_motor\_hat.MotorHat), 10

AdafruitStepperMotor (class in adafruit\_motor\_hat.MotorHat), 9

align\_step() (adafruit\_motor\_hat.MotorHat.AdafruitStepperMotor method), 9

set\_speed() (adafruit\_motor\_hat.MotorHat.AdafruitStepperMotor method), 10

setAllPWM() (adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver.PWM method), 10

setPWM() (adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver.PWM method), 10

setPWMFreq() (adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver.PWM method), 11

softwareReset() (adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver.PWM class method), 11

step() (adafruit\_motor\_hat.MotorHat.AdafruitStepperMotor method), 10

## D

double\_step() (adafruit\_motor\_hat.MotorHat.AdafruitStepperMotor method), 9

## G

get\_i2c\_device() (in module adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver), 11

get\_motor() (adafruit\_motor\_hat.MotorHat.AdafruitMotorHAT method), 10

get\_stepper() (adafruit\_motor\_hat.MotorHat.AdafruitMotorHAT method), 10

## I

interleaved\_step() (adafruit\_motor\_hat.MotorHat.AdafruitStepperMotor method), 9

## M

micro\_step() (adafruit\_motor\_hat.MotorHat.AdafruitStepperMotor method), 9

## P

PWM (class in adafruit\_motor\_hat.Adafruit\_PWM\_Servo\_Driver), 10

## S

set\_pin() (adafruit\_motor\_hat.MotorHat.AdafruitMotorHAT method), 10