
AdafruitServoKit Library Documentation

Release 1.0

Kattni Rembor

Jul 14, 2020

Contents

1	Dependencies	3
1.1	Installing from PyPI	3
2	Usage Example	5
3	Contributing	7
4	Documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_servokit	11
5.2.1	Implementation Notes	12
6	Indices and tables	15
	Python Module Index	17
	Index	19

CircuitPython helper library for the PWM/Servo FeatherWing, Shield and Pi HAT and Bonnet kits.

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)
- [Register](#)

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

1.1 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-servokit
```

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

```
sudo pip3 install adafruit-circuitpython-servokit
```

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-servokit
```


CHAPTER 2

Usage Example

```
import time
from adafruit_servokit import ServoKit

# Set channels to the number of servo channels on your kit.
# 8 for FeatherWing, 16 for Shield/HAT/Bonnet.
kit = ServoKit(channels=8)

kit.servo[0].angle = 180
kit.continuous_servo[1].throttle = 1
time.sleep(1)
kit.continuous_servo[1].throttle = -1
time.sleep(1)
kit.servo[0].angle = 0
kit.continuous_servo[1].throttle = 0
```


CHAPTER 3

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

CHAPTER 4

Documentation

For information on building library documentation, please check out [this guide](#).

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/servokit_simpletest.py

```
1  """Simple test for a standard servo on channel 0 and a continuous rotation servo on_
   ↪channel 1."""
2  import time
3  from adafruit_servokit import ServoKit
4
5  # Set channels to the number of servo channels on your kit.
6  # 8 for FeatherWing, 16 for Shield/HAT/Bonnet.
7  kit = ServoKit(channels=8)
8
9  kit.servo[0].angle = 180
10 kit.continuous_servo[1].throttle = 1
11 time.sleep(1)
12 kit.continuous_servo[1].throttle = -1
13 time.sleep(1)
14 kit.servo[0].angle = 0
15 kit.continuous_servo[1].throttle = 0
```

5.2 adafruit_servokit

CircuitPython helper library for the PWM/Servo FeatherWing, Shield and Pi HAT and Bonnet kits.

- Author(s): Kattni Rembor

5.2.1 Implementation Notes

Hardware:

- 8-Channel PWM or Servo FeatherWing
- Adafruit 16-Channel 12-bit PWM/Servo Shield
- Adafruit 16-Channel PWM/Servo HAT for Raspberry Pi
- Adafruit 16-Channel PWM/Servo Bonnet for Raspberry Pi

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
- Adafruit's Register library: https://github.com/adafruit/Adafruit_CircuitPython_Register
- Adafruit's PCA9685 library: https://github.com/adafruit/Adafruit_CircuitPython_PCA9685
- Adafruit's Motor library: https://github.com/adafruit/Adafruit_CircuitPython_Motor

```
class adafruit_servokit.ServoKit (*, channels, i2c=None, address=64, reference_clock_speed=25000000, frequency=50)
```

Class representing an Adafruit PWM/Servo FeatherWing, Shield or Pi HAT and Bonnet kits.

Automatically uses the I2C bus on a Feather, Metro or Raspberry Pi.

Initialise the PCA9685 chip at address.

The internal reference clock is 25MHz but may vary slightly with environmental conditions and manufacturing variances. Providing a more precise `reference_clock_speed` can improve the accuracy of the frequency and `duty_cycle` computations. See the `calibration.py` example in the [PCA9685 GitHub repo](#) for how to derive this value by measuring the resulting pulse widths.

Parameters

- **channels** (*int*) – The number of servo channels available. Must be 8 or 16. The FeatherWing has 8 channels. The Shield, HAT, and Bonnet have 16 channels.
- **address** (*int*) – The I2C address of the PCA9685. Default address is 0x40.
- **reference_clock_speed** (*int*) – The frequency of the internal reference clock in Hertz. Default reference clock speed is 25000000.
- **frequency** (*int*) – The overall PWM frequency of the PCA9685 in Hertz. Default frequency is 50.

continuous_servo

:py:class:~adafruit_motor.servo.ContinuousServo controls for continuous rotation servos.

This FeatherWing example rotates a continuous rotation servo on channel 0 forward for one second, then backward for one second, and then stops the rotation.

```
import time
from adafruit_servokit import ServoKit

kit = ServoKit(channels=8)

kit.continuous_servo[0].throttle = 1
time.sleep(1)
kit.continuous_servo[0].throttle = -1
```

(continues on next page)

(continued from previous page)

```
time.sleep(1)
kit.continuous_servo[0].throttle = 0
```

servo

:py:class:~adafruit_motor.servo.Servo controls for standard servos.

This FeatherWing example rotates a servo on channel 0 to 180 degrees for one second, and then returns it to 0 degrees.

```
import time
from adafruit_servokit import ServoKit

kit = ServoKit(channels=8)

kit.servo[0].angle = 180
time.sleep(1)
kit.servo[0].angle = 0
```


CHAPTER 6

Indices and tables

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

a

adafruit_servokit, 11

A

adafruit_servokit (*module*), 11

C

continuous_servo (*adafruit_servokit.ServoKit attribute*), 12

S

servo (*adafruit_servokit.ServoKit attribute*), 13

ServoKit (*class in adafruit_servokit*), 12