Adafruitfeatherwing Library Documentation

Release 1.0

Scott Shawcroft

Mar 06, 2018

Contents

1 Dependencies **3**

2 API Reference **5**

2.1 `adafruit_featherwing.motor_featherwing` 5

2.2 `adafruit_featherwing.ina219_featherwing` 5

2.3 `adafruit_featherwing.joy_featherwing` 5

3 Contributing **15**

4 Building locally **17**

Python Module Index **19**

This library provides FeatherWing specific classes for those that require a significant amount of initialization.

CHAPTER 1

Dependencies

These drivers depends on:

- Adafruit CircuitPython
- PCA9685
- Motor

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle and highly recommended over installing each one.

CHAPTER 2

API Reference

2.1 adafruit_featherwing.motor_featherwing

Helper for using motors with the Motor FeatherWing.

- Author(s): Scott Shawcroft

2.2 adafruit_featherwing.ina219_featherwing

Helper for using the INA219 FeatherWing.

- Author(s): Kattni Rembor

2.3 adafruit_featherwing.joy_featherwing

Helper for using the Joy FeatherWing.

- Author(s): Kattni Rembor

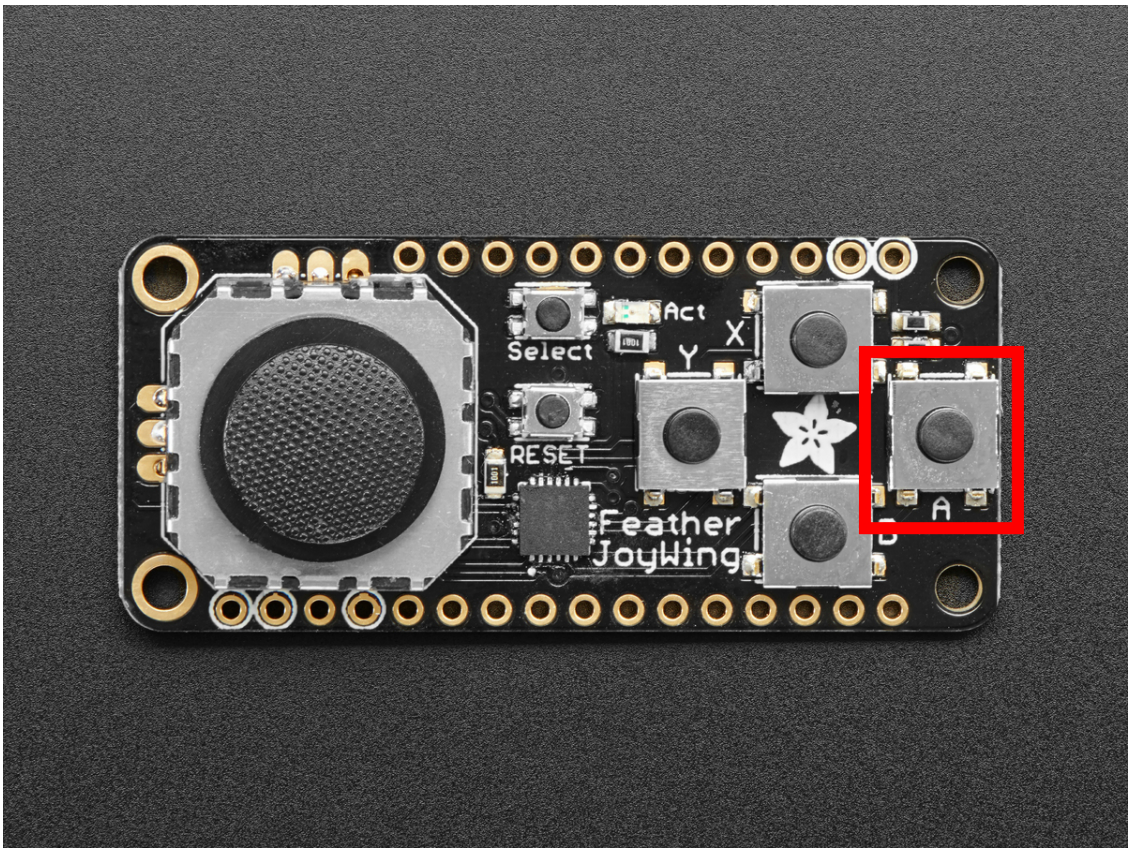
class adafruit_featherwing.joy_featherwing.**JoyFeatherWing**

Class representing an Adafruit Joy FeatherWing.

Automatically uses the feather's I2C bus.

button_a

Joy featherwing button A.



This example prints when button A is pressed.

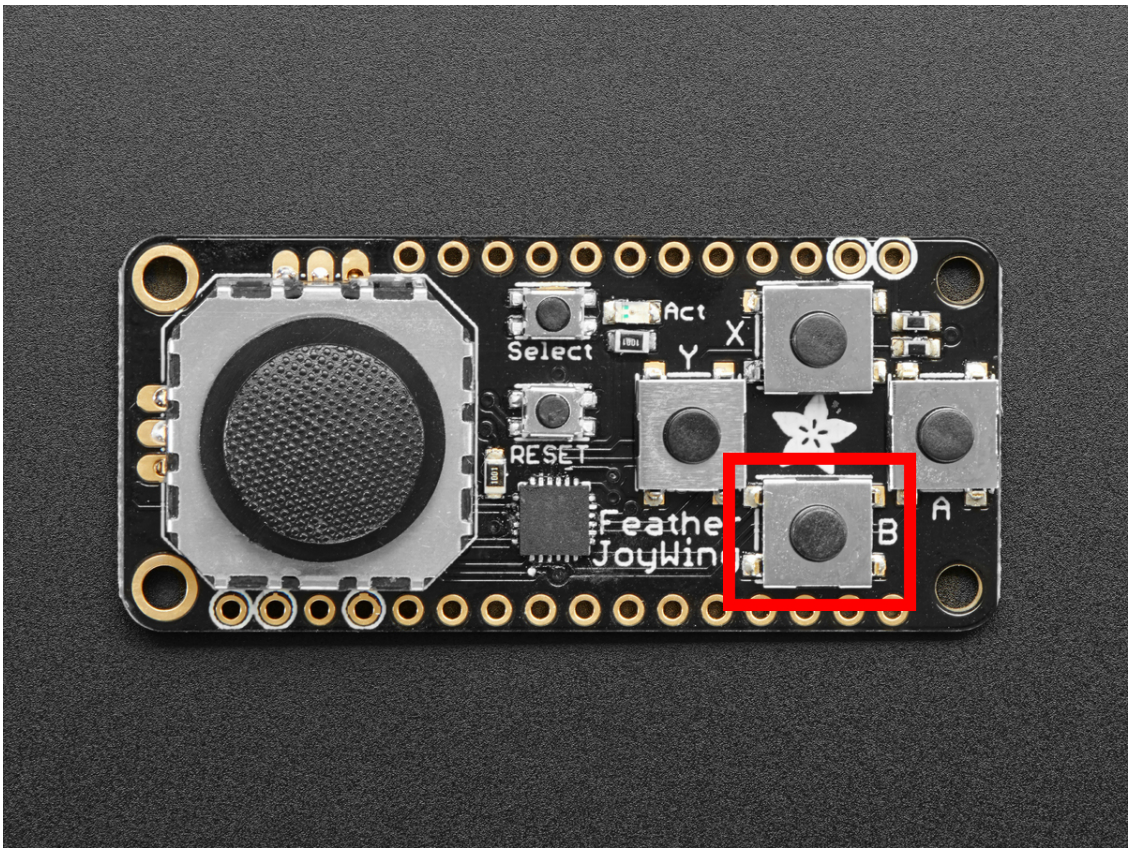
```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()

while True:
    if wing.button_a:
    print("Button A pressed!")
```

button_b

Joy featherwing button B.



This example prints when button B is pressed.

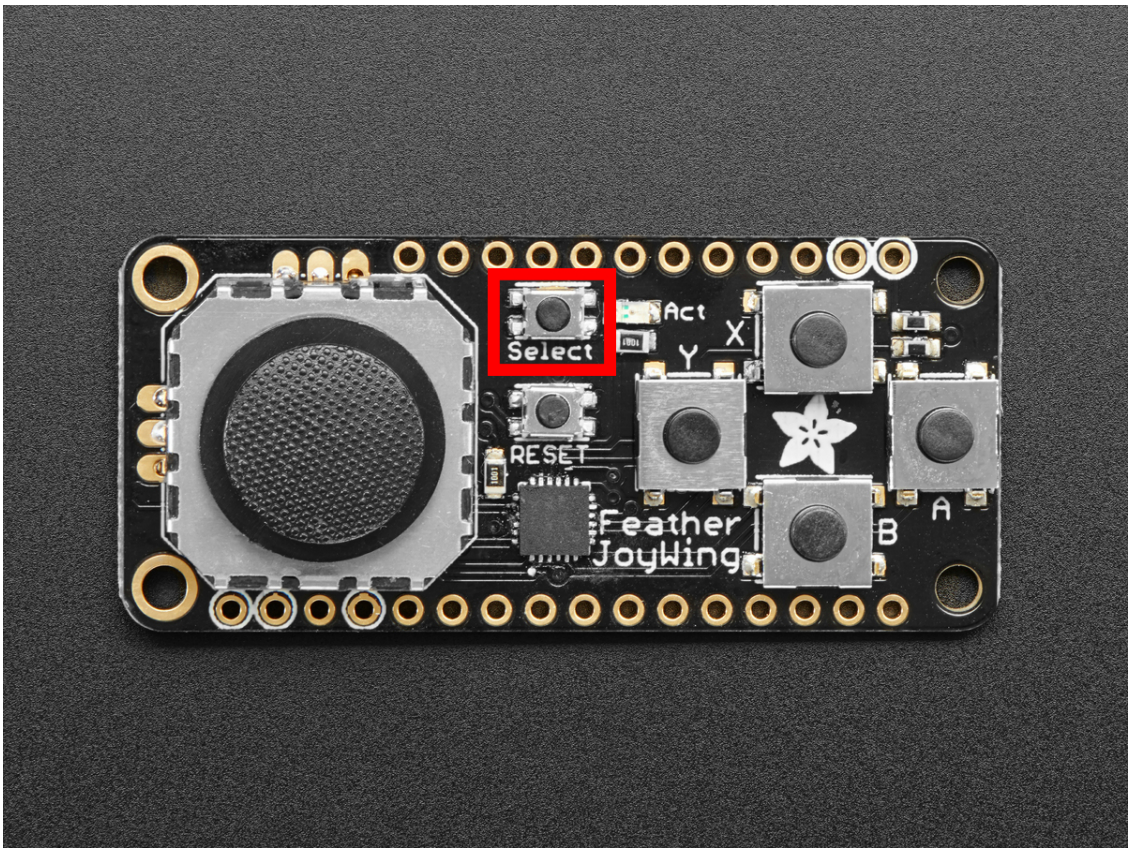
```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()

while True:
    if wing.button_b:
    print("Button B pressed!")
```

button_select

Joy featherwing button SELECT.



This example prints when button SELECT is pressed.

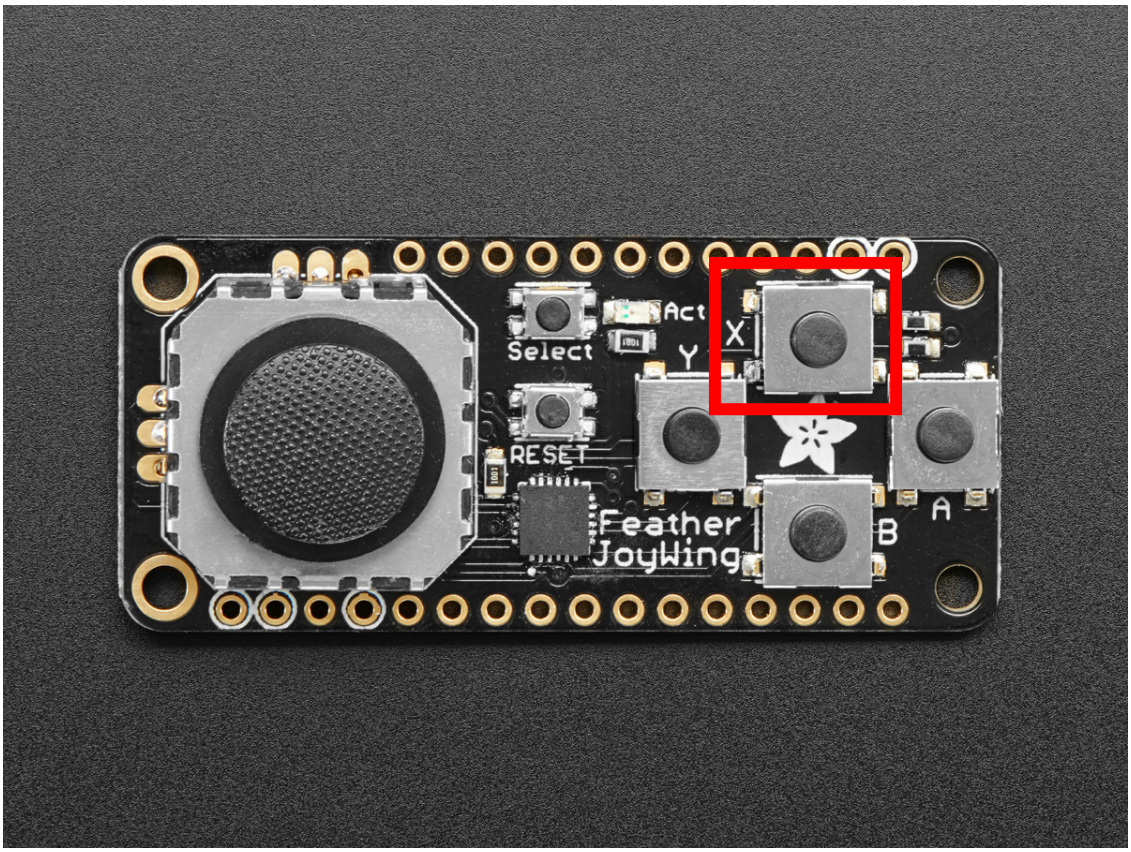
```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()

while True:
    if wing.button_select:
    print("Button SELECT pressed!")
```

button_x

Joy featherwing button X.



This example prints when button X is pressed.

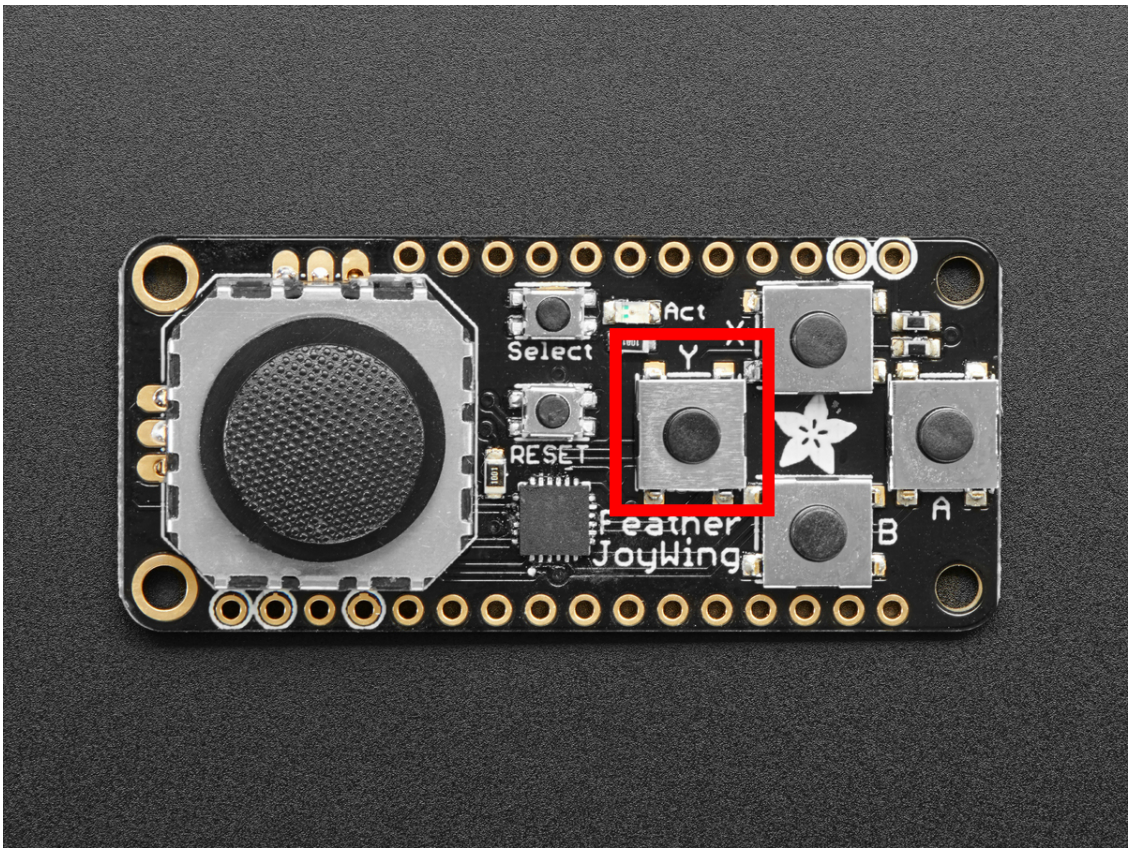
```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()

while True:
    if wing.button_x:
    print("Button X pressed!")
```

button_y

Joy featherwing button Y.



This example prints when button Y is pressed.

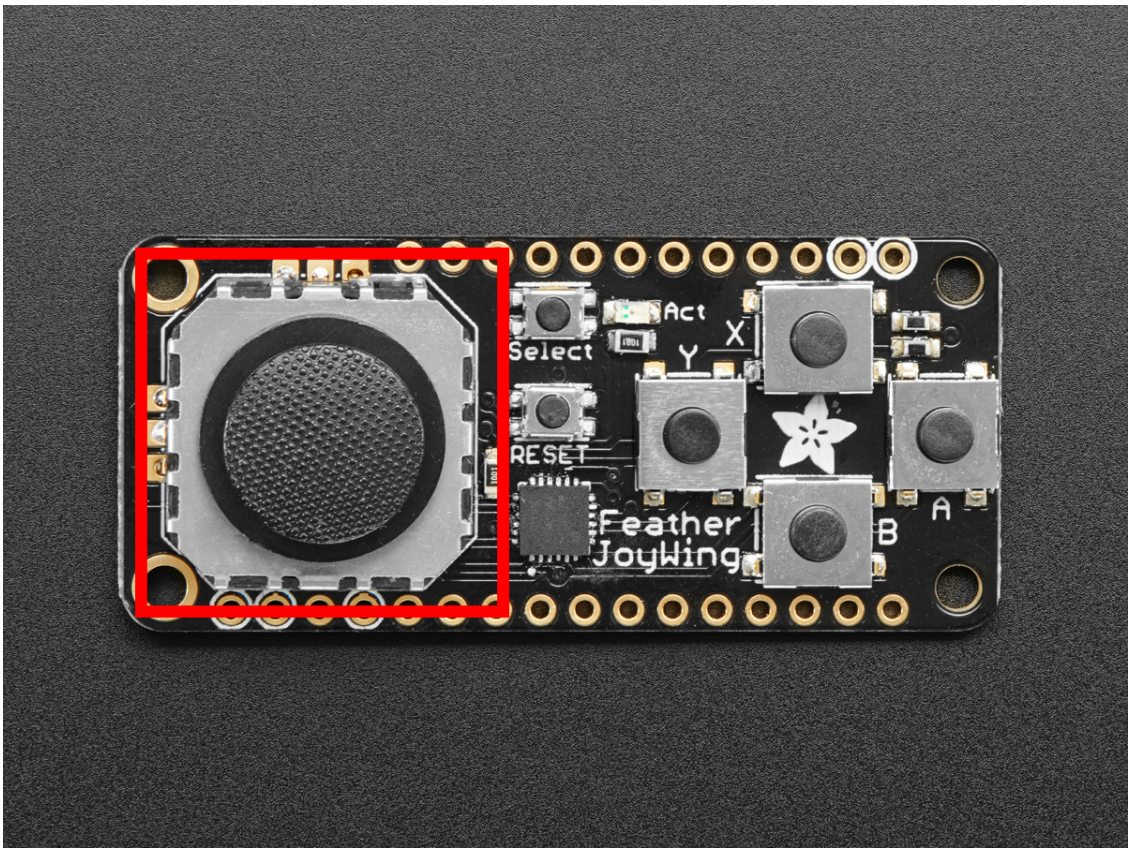
```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()

while True:
    if wing.button_y:
    print("Button Y pressed!")
```

joystick

Joy FeatherWing joystick.



This example zeros the joystick, and prints the coordinates of joystick when it is moved.

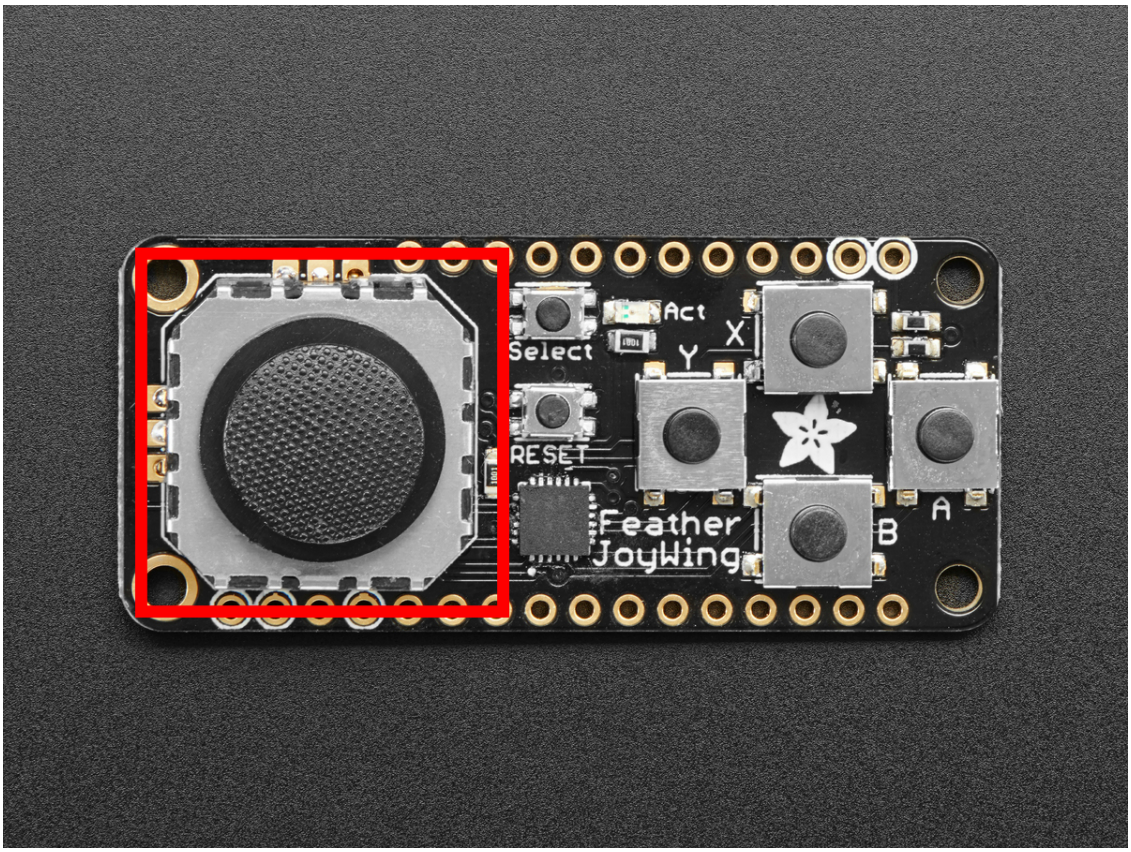
```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()
last_x = 0
last_y = 0
wing.zero_joystick()

while True:
    x, y = wing.joystick
    if (abs(x - last_x) > 3) or (abs(y - last_y) > 3):
        last_x = x
        last_y = y
        print(x, y)
    time.sleep(0.01)
```

joystick_offset

Offset used to correctly report (0, 0) when the joystick is centered.



Provide a tuple of (x, y) to set your joystick center to (0, 0). The offset you provide is subtracted from the current reading. For example, if your joystick reads as (-4, 0), you would enter (-4, 0) as the offset. The code will subtract -4 from -4, and 0 from 0, returning (0, 0).

This example supplies an offset for zeroing, and prints the coordinates of the joystick when it is moved.

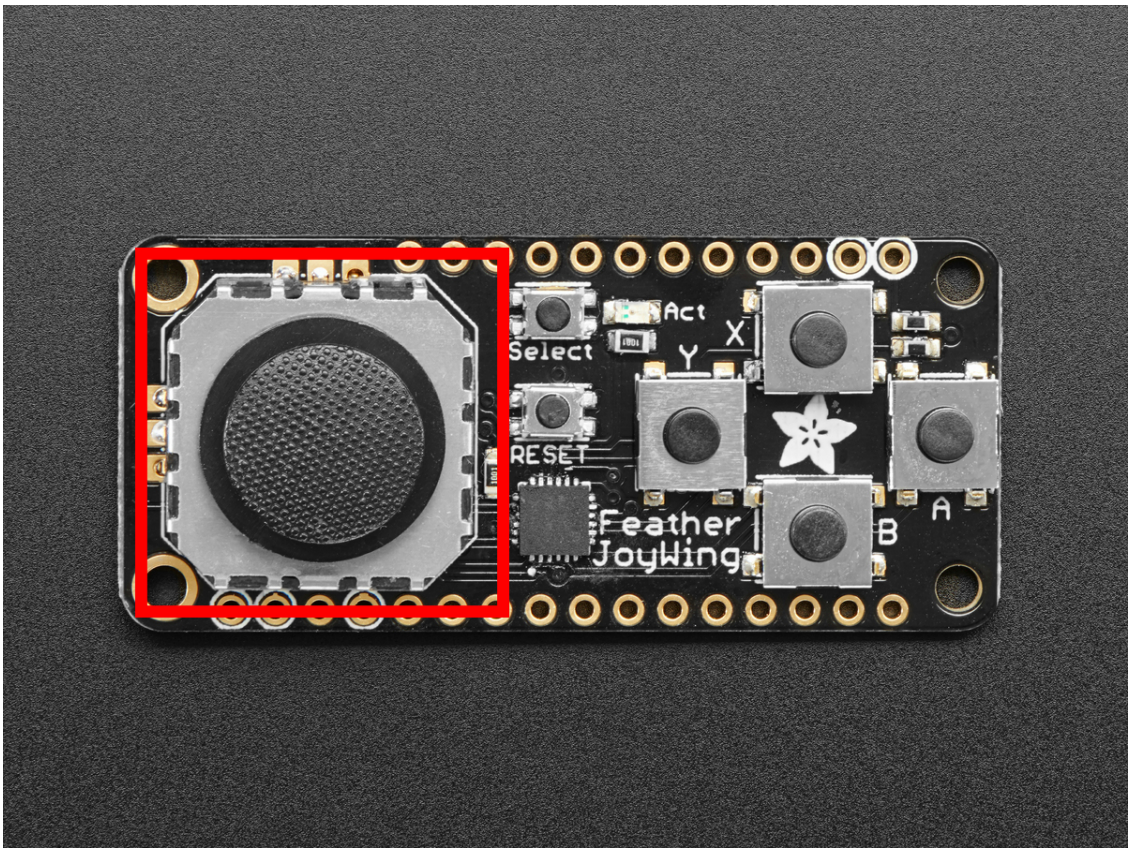
```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()
last_x = 0
last_y = 0

while True:
    wing.joystick_offset = (-4, 0)
    x, y = wing.joystick
    if (abs(x - last_x) > 3) or (abs(y - last_y) > 3):
        last_x = x
        last_y = y
        print(x, y)
    time.sleep(0.01)
```

zero_joystick()

Zeros the joystick by using current reading as (0, 0). Note: You must not be touching the joystick at the time of zeroing for it to be accurate.



This example zeros the joystick, and prints the coordinates of joystick when it is moved.

```
from adafruit_featherwing import joy_featherwing
import time

wing = joy_featherwing.JoyFeatherWing()
last_x = 0
last_y = 0
wing.zero_joystick()

while True:
    x, y = wing.joystick
    if (abs(x - last_x) > 3) or (abs(y - last_y) > 3):
        last_x = x
        last_y = y
        print(x, y)
    time.sleep(0.01)
```

CHAPTER 3

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 4

Building locally

To build this library locally you'll need to install the circuitpython-build-tools package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-featherwing --
↪library_location .
```

Python Module Index

a

adafruit_featherwing.ina219_featherwing, 5
adafruit_featherwing.joy_featherwing, 5
adafruit_featherwing.motor_featherwing, 5

Index

A

adafruit_featherwing.ina219_featherwing (module), 5
adafruit_featherwing.joy_featherwing (module), 5
adafruit_featherwing.motor_featherwing (module), 5

B

button_a (adafruit_featherwing.joy_featherwing.JoyFeatherWing attribute), 5
button_b (adafruit_featherwing.joy_featherwing.JoyFeatherWing attribute), 6
button_select (adafruit_featherwing.joy_featherwing.JoyFeatherWing attribute), 7
button_x (adafruit_featherwing.joy_featherwing.JoyFeatherWing attribute), 8
button_y (adafruit_featherwing.joy_featherwing.JoyFeatherWing attribute), 9

J

JoyFeatherWing (class in adafruit_featherwing.joy_featherwing), 5
joystick (adafruit_featherwing.joy_featherwing.JoyFeatherWing attribute), 10
joystick_offset (adafruit_featherwing.joy_featherwing.JoyFeatherWing attribute), 11

Z

zero_joystick() (adafruit_featherwing.joy_featherwing.JoyFeatherWing method), 12