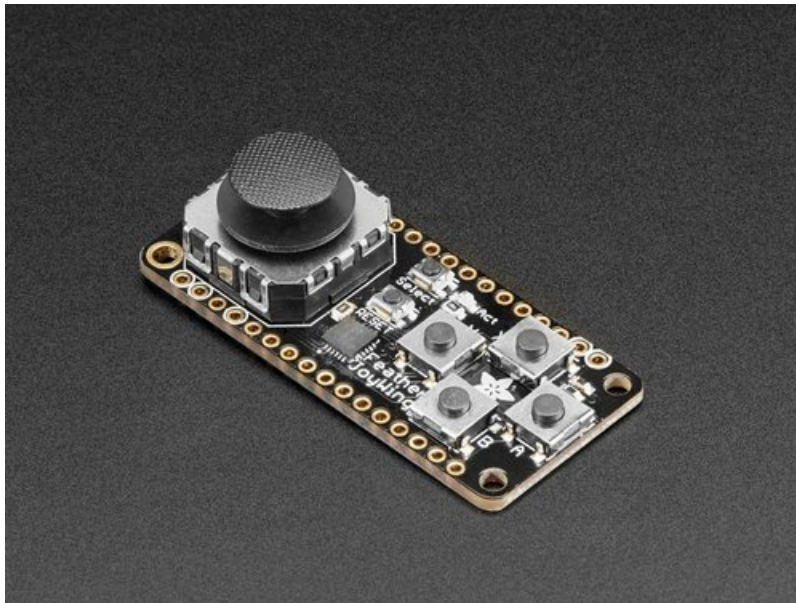




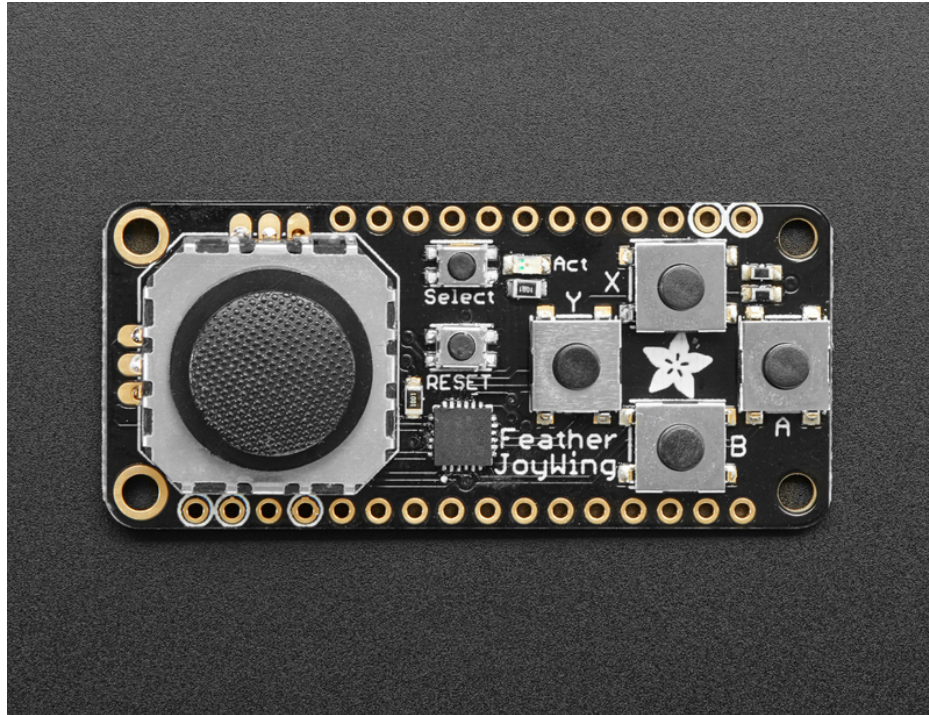
Joy Featherwing

Created by Dean Miller

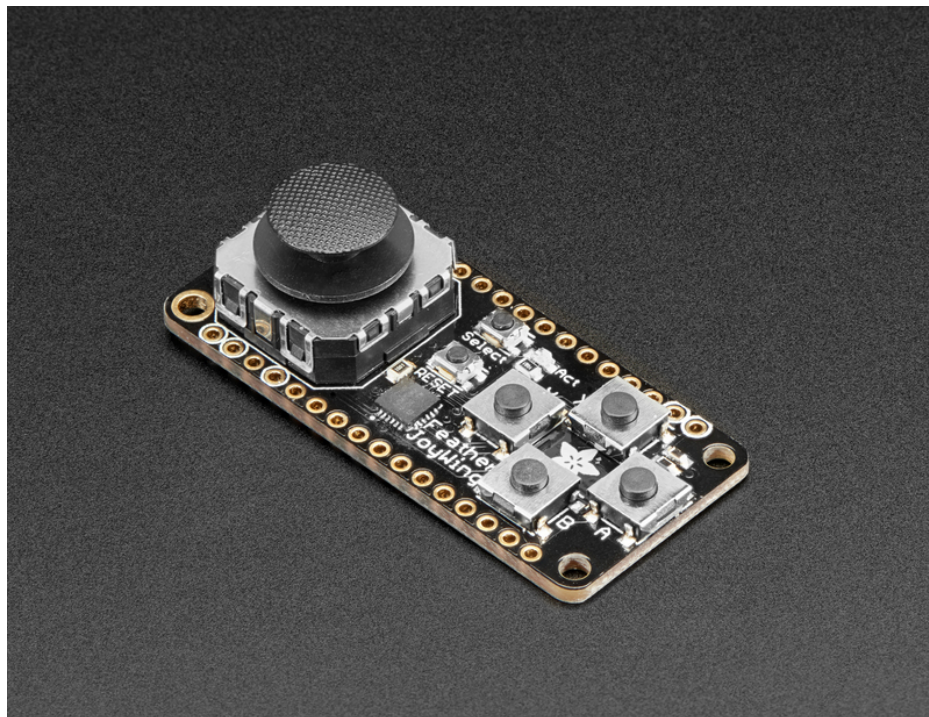


Last updated on 2020-04-15 10:31:48 AM EDT

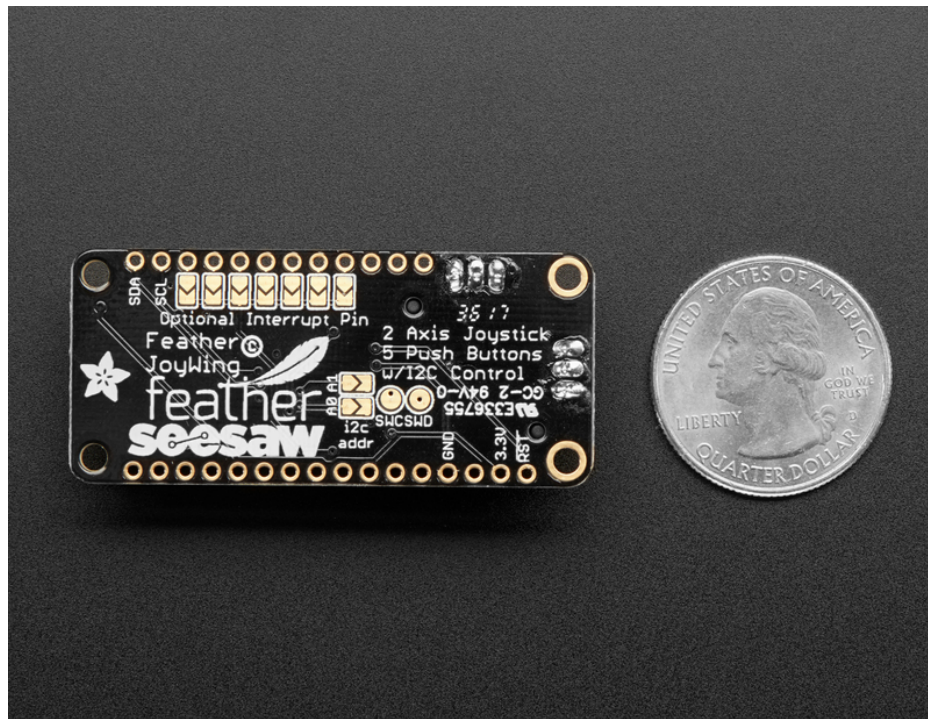
Overview



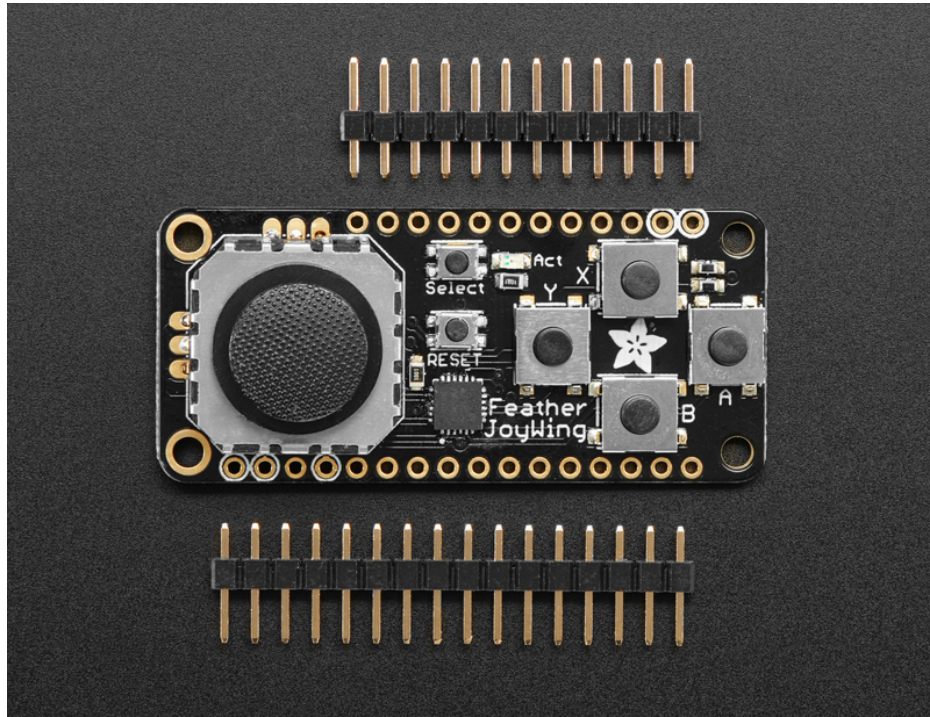
Make a game or robotic controller with this Joy-ful **FeatherWing**. This FeatherWing has a 2-axis joystick and 5 momentary buttons (4 large and 1 small) so you can turn your feather board into a tiny game controller. This wing communicates with your host microcontroller over I2C so it's easy to use and doesn't take up any of your precious analog or digital pins. There is also an optional **interrupt pin** that can alert your feather when a button has been pressed or released to free up processor time for other tasks.



This FeatherWing features Adafruit **Seesaw** technology - a custom programmed little helper microcontroller that takes the two analog inputs from the joystick, and 5 button inputs, and converts it into a pretty I2C interface. This I2C interface means you don't 'lose' any GPIO or analog inputs when using this 'Wing, **and it works with any and all Feathers!** You can easily stack this with any other FeatherWing because I2C is a shared bus. If you have an I2C address conflict, or you want to connect more than one of these to a Feather, there are two address-select jumpers so you have 4 options of I2C addresses



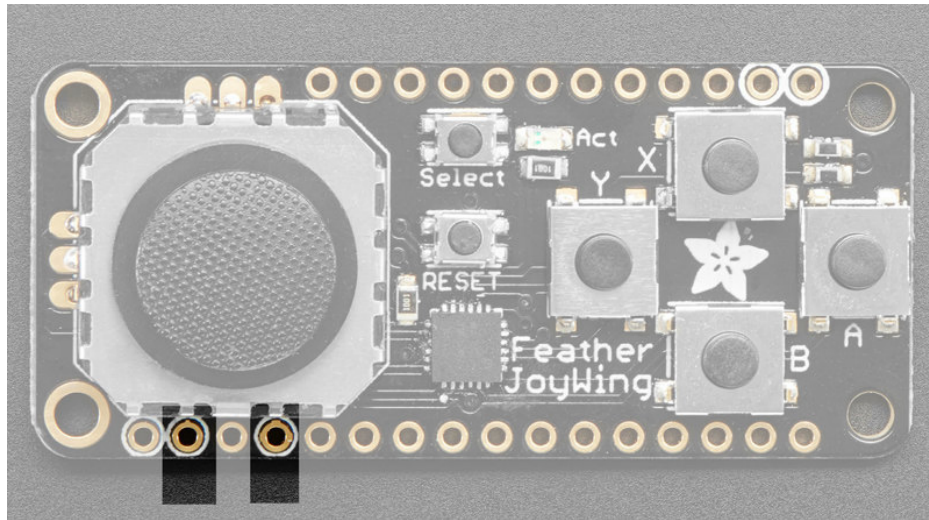
There's an optional IRQ (interrupt) line that you can use if you'd like the Wing to let you know when a button has been pressed. Since its optional (and most Feathers are perfectly happy polling the Wing for data) we left a bunch of solder jumpers so you can select what pin has the IRQ on it. Or just leave it disconnected!



We have an Adafruit Seesaw library for Arduino, Python and CircuitPython so you can get started with the Joy FeatherWing in a jiffy.

Pinouts

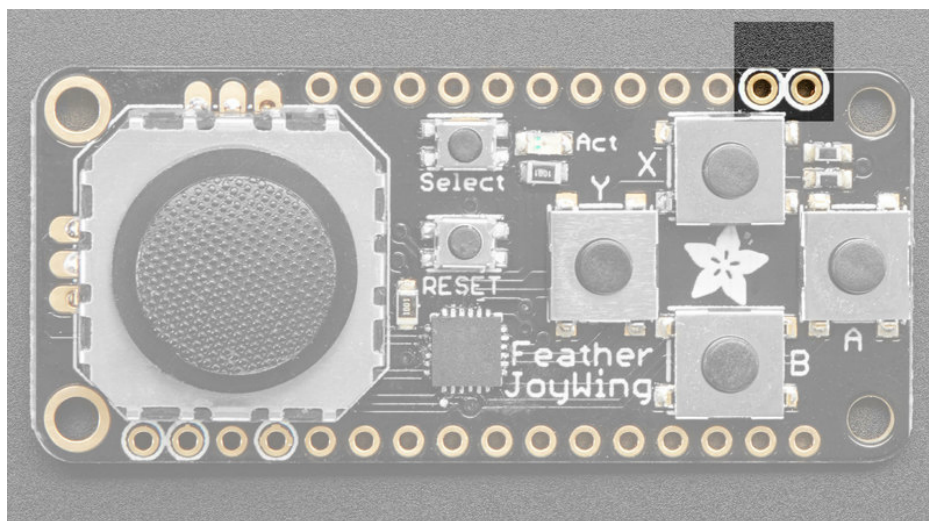
Power and Reset Pins



The power to the Joy Featherwing comes from the host Feather's 3.3V regulator, 3.3V and ground are both required and are highlighted above

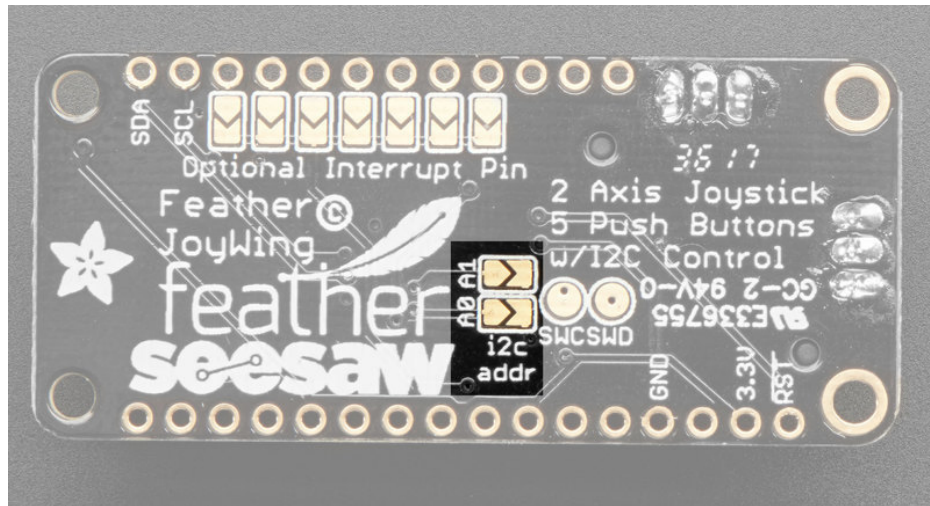
The Feather's Reset pin ends up coming from the left-most circled pin and connects to the on-board reset button. This is great for resetting when you have the Wing on top

I2C Data Pins



All data and control is done over I2C so it works with any and all Feathers! SDA and SCL are highlighted above. There are 10K pullups to 3.3V on each.

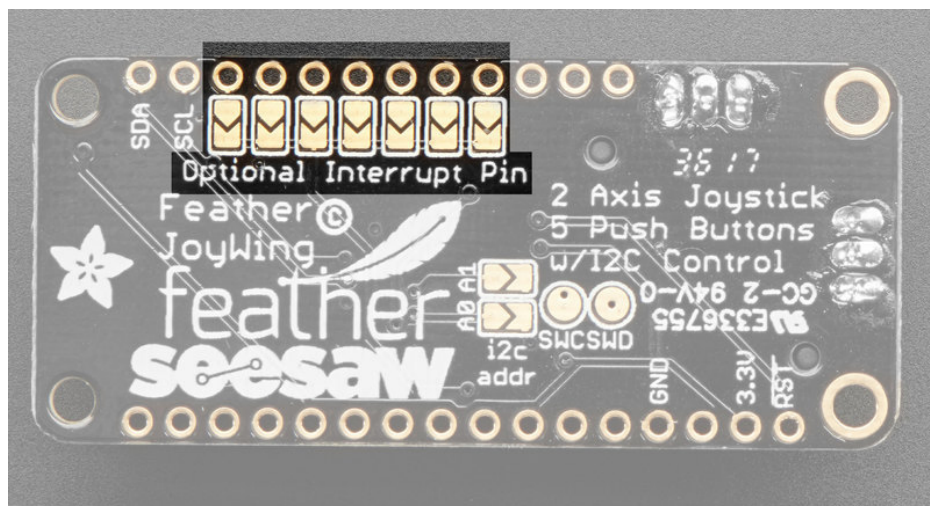
I2C Addressing



Because multiple featherwings of all kinds can be stacked, the I2C address can be changed if necessary. The Joy Featherwing's default I2C address is **0x49**, but soldering jumpers **A0** and/or **A1** can change this:

A0	A1	Addr
open	open	0x49 (default)
closed	open	0x4A
open	closed	0x4B
closed	closed	0x4C

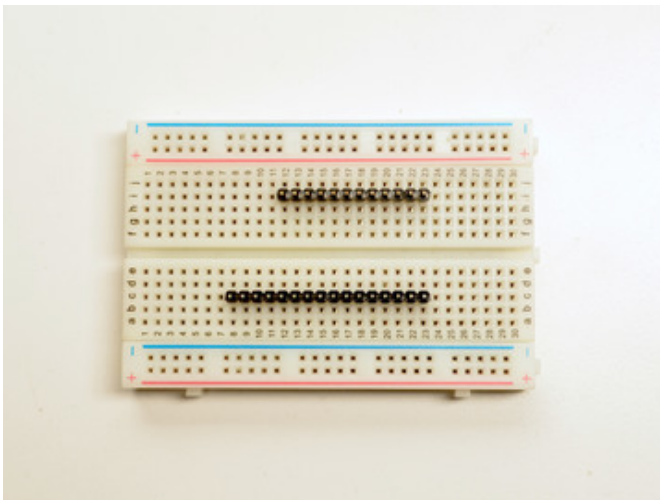
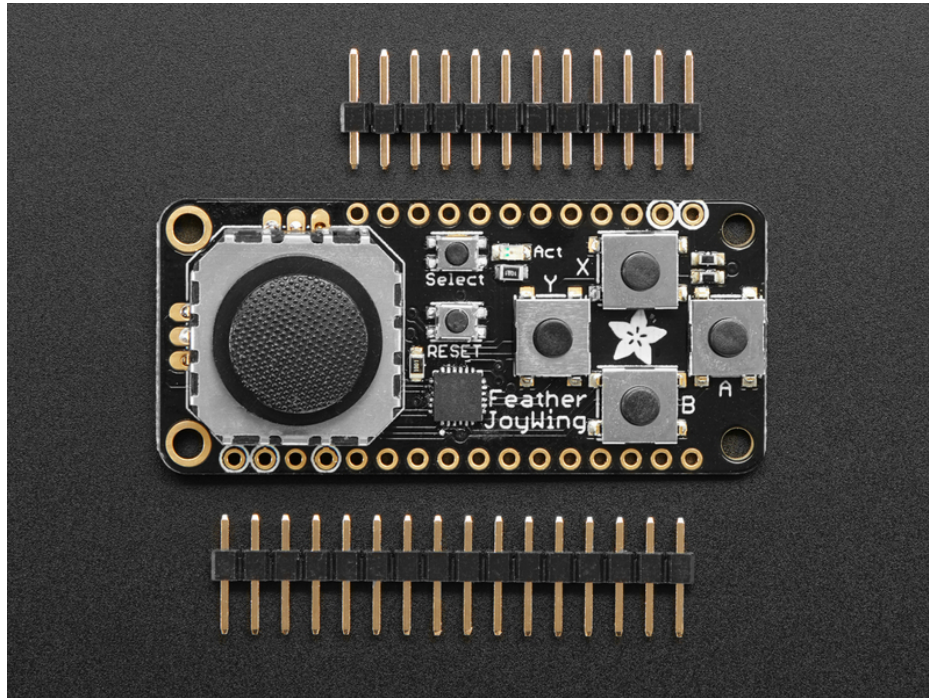
Optional Interrupt Pin



Soldering one of the jumpers for the optional interrupt will connect that pin of your Feather to the interrupt line on the Joy Featherwing. This will allow your Feather to know whether a button has been pressed or released without having to read each button over and over in a loop.

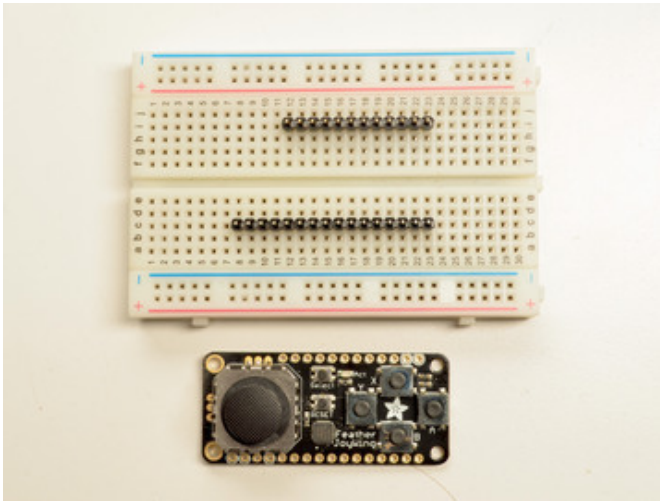
Each Feather has different pin names/numbers so check your mainboard for which pin you want to use then solder that jumper closed

Assembly

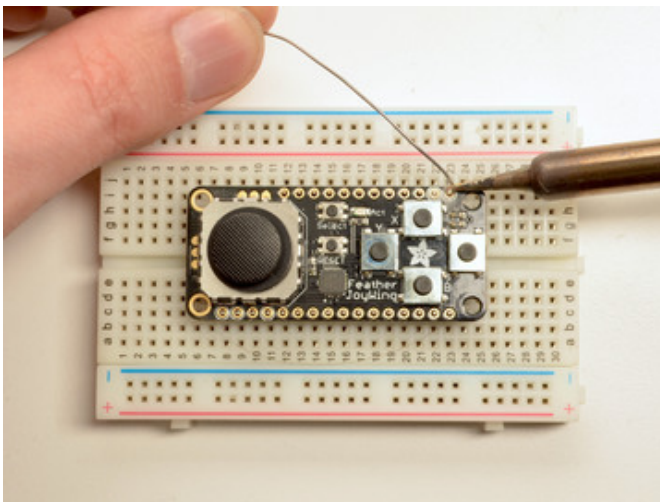
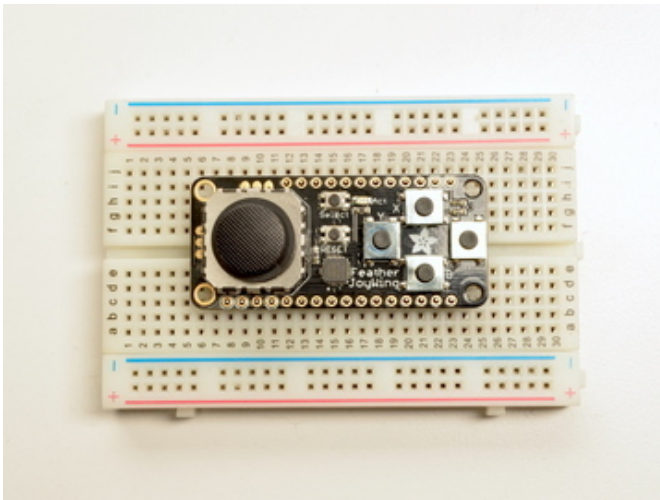


Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



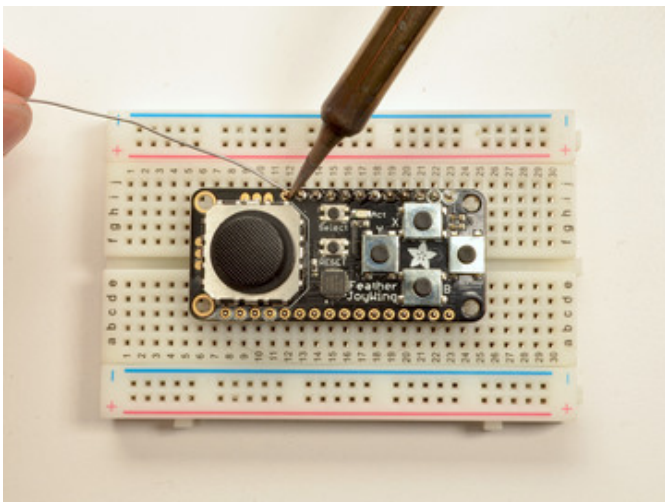
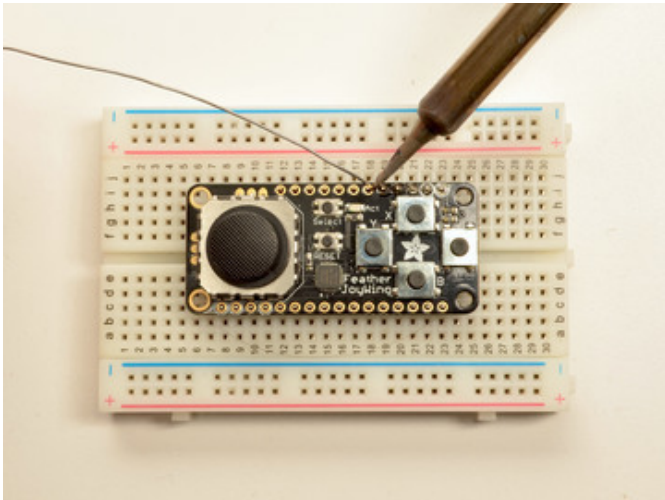
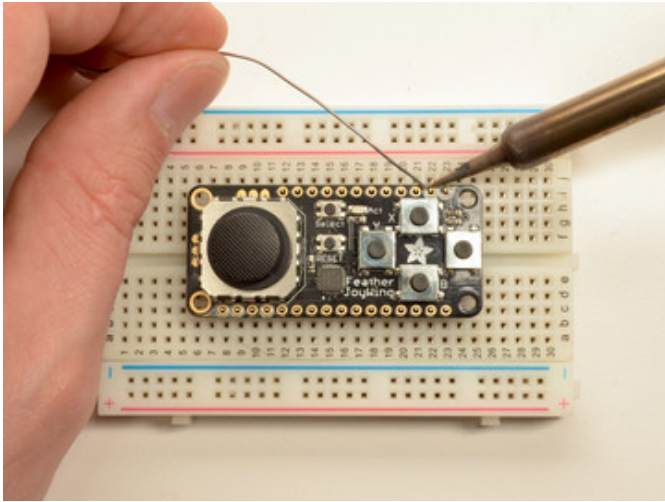
Add the FeatherWing:
Place the featherwing over the pins so that the short pins poke through the two rows of breakout pads

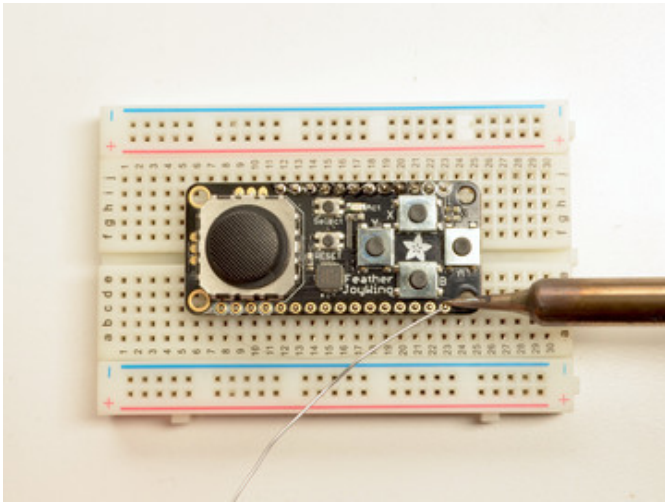


And Solder!
Be sure to solder all pins for reliable electrical contact.

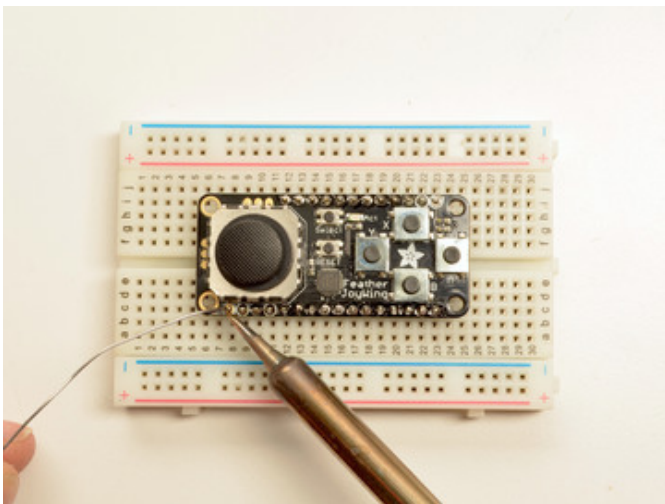
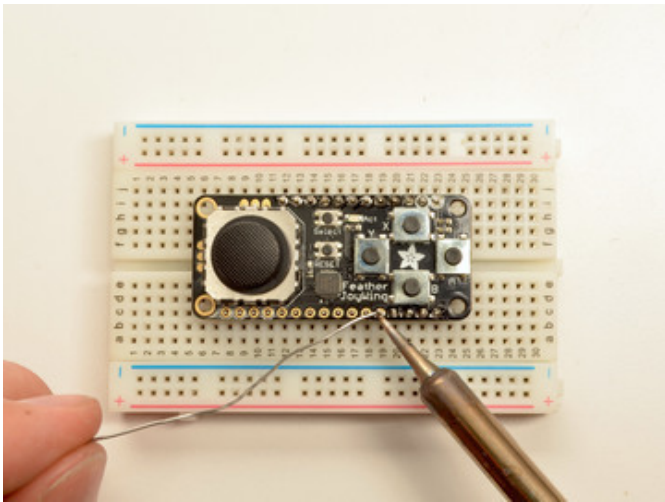
(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](https://adafruit.it/aTk) (<https://adafruit.it/aTk>)).

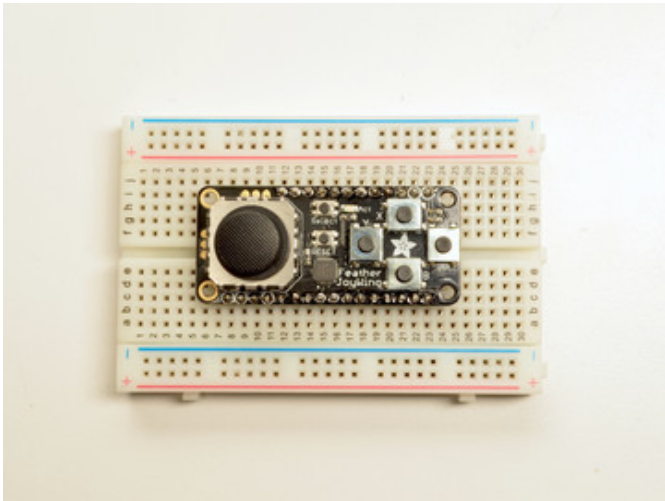
Start by soldering the first row of header





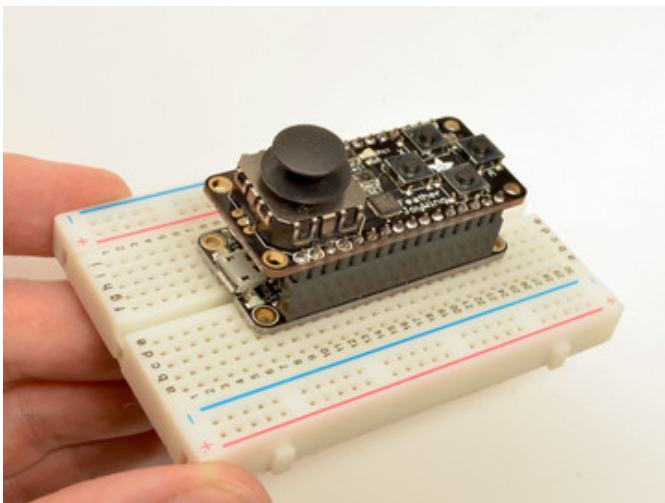
Now flip around and solder the other row completely





You're done with the two header strips.

Check your solder joints visually and continue onto the next steps



OK You're done! You can now plug your FeatherWing into your Feather and get your JOY on!

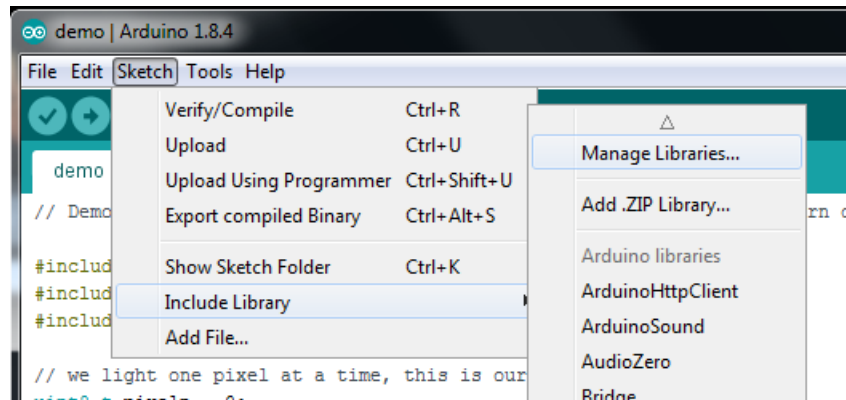
Arduino Wiring & Test

Adafruit seesaw Library

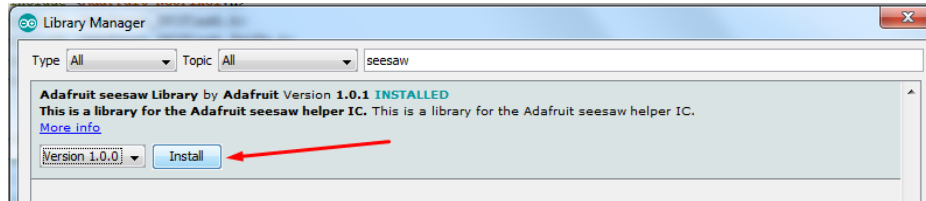
To use with the Arduino IDE, you will need to install the **Adafruit_seesaw** library (here's [the Github repository \(https://adafru.it/BrV\)](https://adafru.it/BrV).)

It is available from the Arduino library manager so we recommend using that.

From the IDE open up the library manager...



And type in **adafruit seesaw** to locate the library. Click **Install**



For more details on how to install Arduino libraries, check out our detailed tutorial! (<https://adafru.it/aYM>)

Running the Example Code

The library comes with a Joy Featherwing example to get you started fast. Plug the Wing into the Feather and connect the Feather to your computer with the USB cable.

Open the Arduino IDE and load **File->Examples->Adafruit_seesaw->joy_featherwing_example**

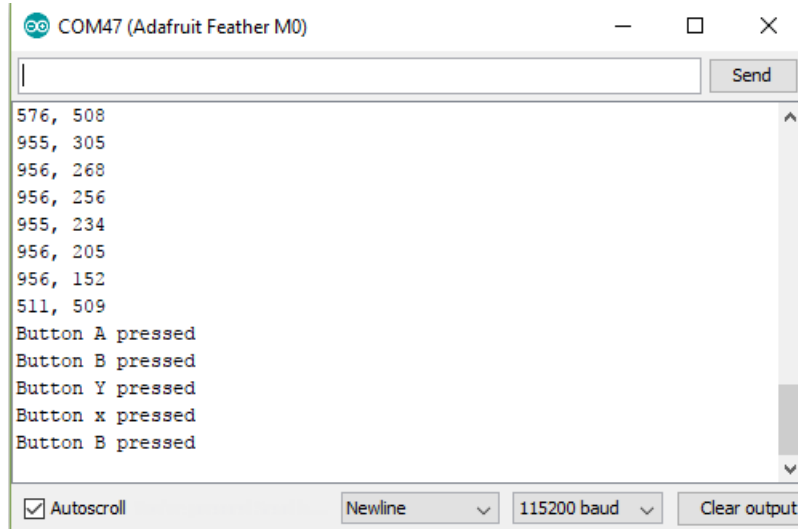


The example assumes the interrupt pin is connected to pin 5 on the Feather via the first solder jumper on the far left. If you have not soldered this jumper the example code will need to be modified.

Click the Upload button and open the Serial Monitor at 115200 baud speed.

When you move the joystick you should see data appear in the format **x,y**

When you press a button, a notification should appear in the monitor.



FeatherJoyWing library!

If you'd like an all-in-one library that abstracts some of the FeatherWing for you, check out

Leonid (<https://adafru.it/ven>)'s FeatherJoyWing library

<https://github.com/leonidlezner/FeatherJoyWing> (<https://adafru.it/Bsi>)

Circuit Python Wiring & Test

Download Adafruit_CircuitPython_seesaw Library

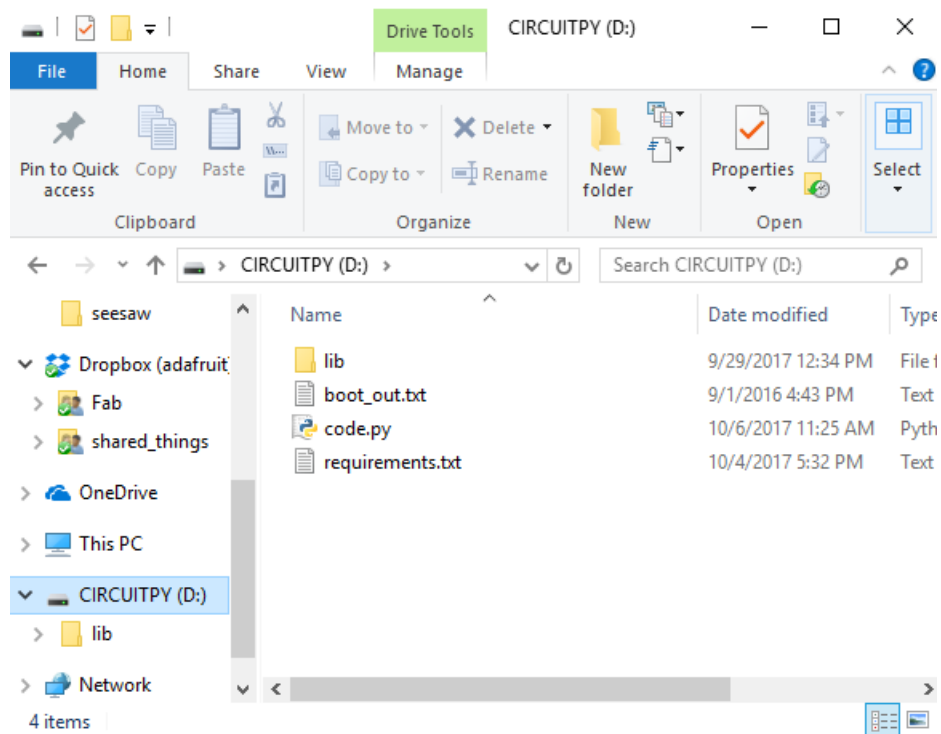
To begin reading data from your Joy Featherwing, you will need to download `Adafruit_CircuitPython_seesaw` from our github repository. You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

<https://adafru.it/A0u>

<https://adafru.it/A0u>

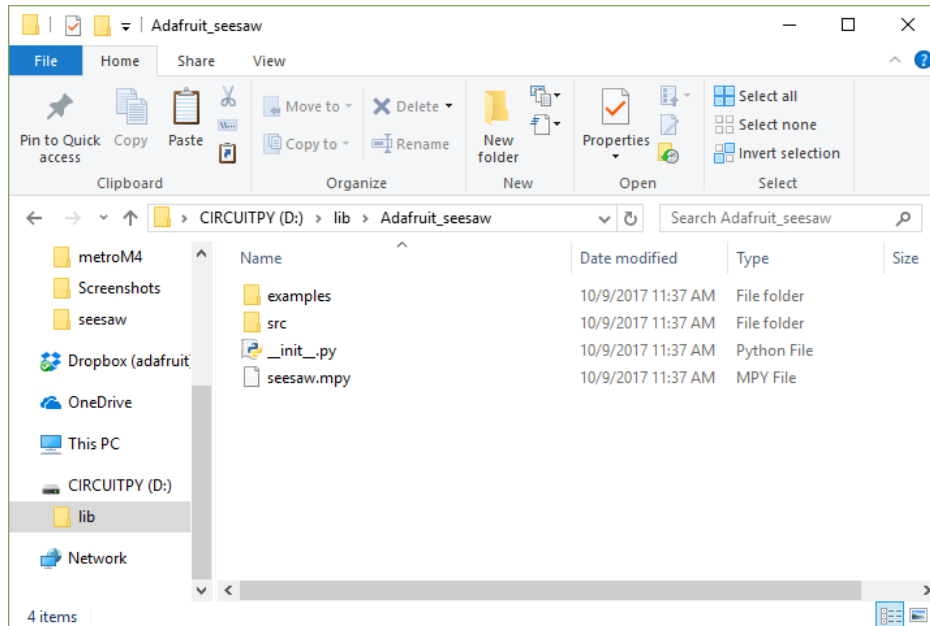
Extract the zipped folder. And rename the **folder it contains** to **Adafruit_seesaw**.

Plug your CircuitPython device into your computer via the USB cable, and you should see the CIRCUITPY drive appear.



you should have the **lib** folder already on the drive. If you do not have the **lib** folder, or you encounter errors due to missing libraries, you can download the latest library package [here](https://adafru.it/Bsl) (<https://adafru.it/Bsl>).

Drag the `Adafruit_seesaw` folder you just renamed to the **CIRCUITPY/lib** folder on your CircuitPython drive. The folder should look like this:



open the **code.py** file in the root of the **CIRCUITPY** drive and copy and paste the following code:

```

import time

from board import SCL, SDA
import busio
from micropython import const

from adafruit_seesaw.seesaw import Seesaw

# pylint: disable=bad-whitespace
BUTTON_RIGHT = const(6)
BUTTON_DOWN = const(7)
BUTTON_LEFT = const(9)
BUTTON_UP = const(10)
BUTTON_SEL = const(14)
# pylint: enable=bad-whitespace
button_mask = const(
    (1 << BUTTON_RIGHT)
    | (1 << BUTTON_DOWN)
    | (1 << BUTTON_LEFT)
    | (1 << BUTTON_UP)
    | (1 << BUTTON_SEL)
)

i2c_bus = busio.I2C(SCL, SDA)

ss = Seesaw(i2c_bus)

ss.pin_mode_bulk(button_mask, ss.INPUT_PULLUP)

last_x = 0
last_y = 0

while True:
    x = ss.analog_read(2)
    y = ss.analog_read(3)

    if (abs(x - last_x) > 3) or (abs(y - last_y) > 3):
        print(x, y)
        last_x = x
        last_y = y

    buttons = ss.digital_read_bulk(button_mask)
    if not buttons & (1 << BUTTON_RIGHT):
        print("Button A pressed")

    if not buttons & (1 << BUTTON_DOWN):
        print("Button B pressed")

    if not buttons & (1 << BUTTON_LEFT):
        print("Button Y pressed")

    if not buttons & (1 << BUTTON_UP):
        print("Button x pressed")

    if not buttons & (1 << BUTTON_SEL):
        print("Button SEL pressed")

    time.sleep(0.01)

```

Plug your Joy Featherwing onto the top of your CircuitPython Feather board.

When you open the REPL you should see output when you press buttons or move the joystick.

