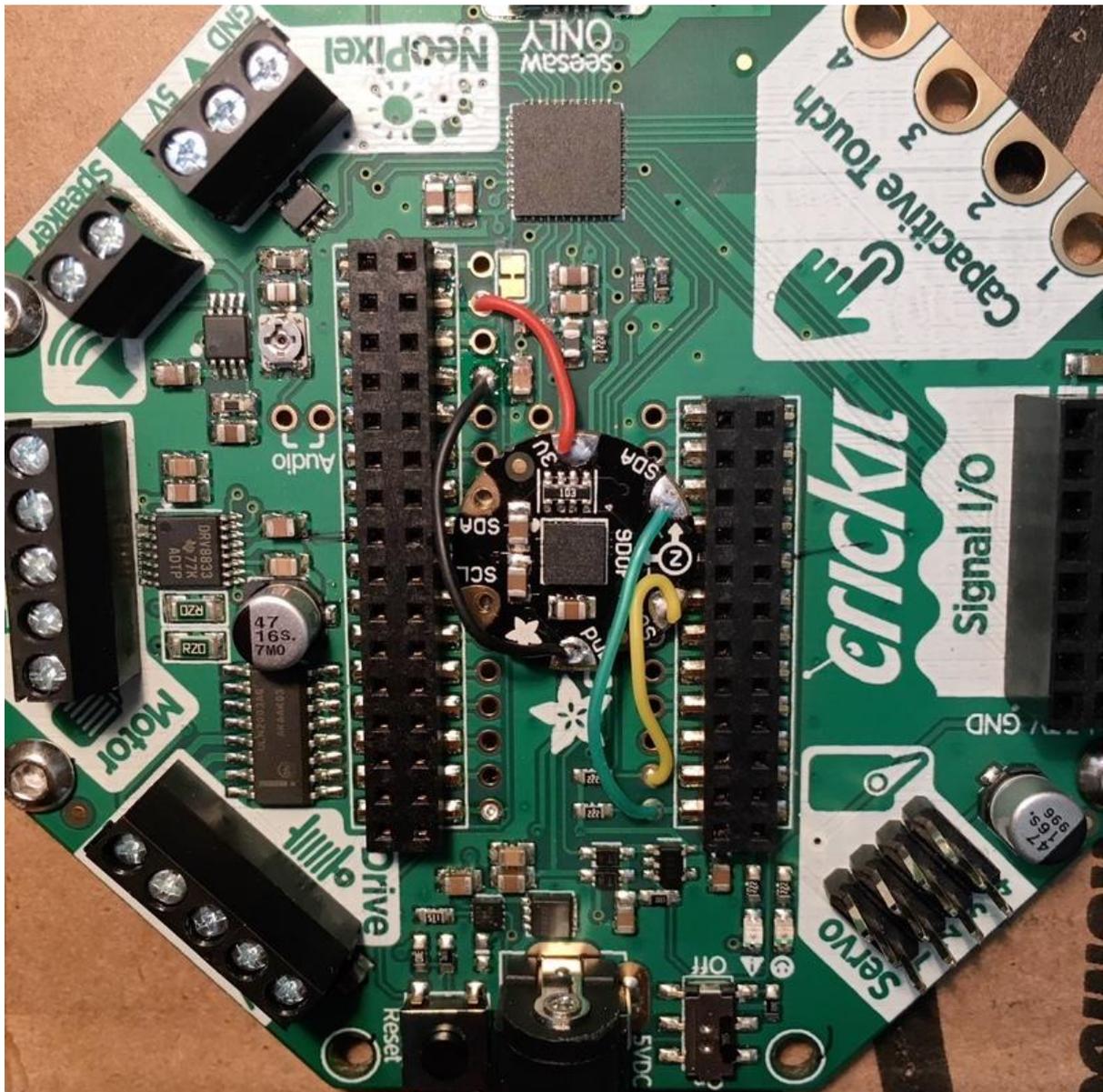




Adding an Accelerometer to your FeatherWing Cricket

Created by Dave Astels



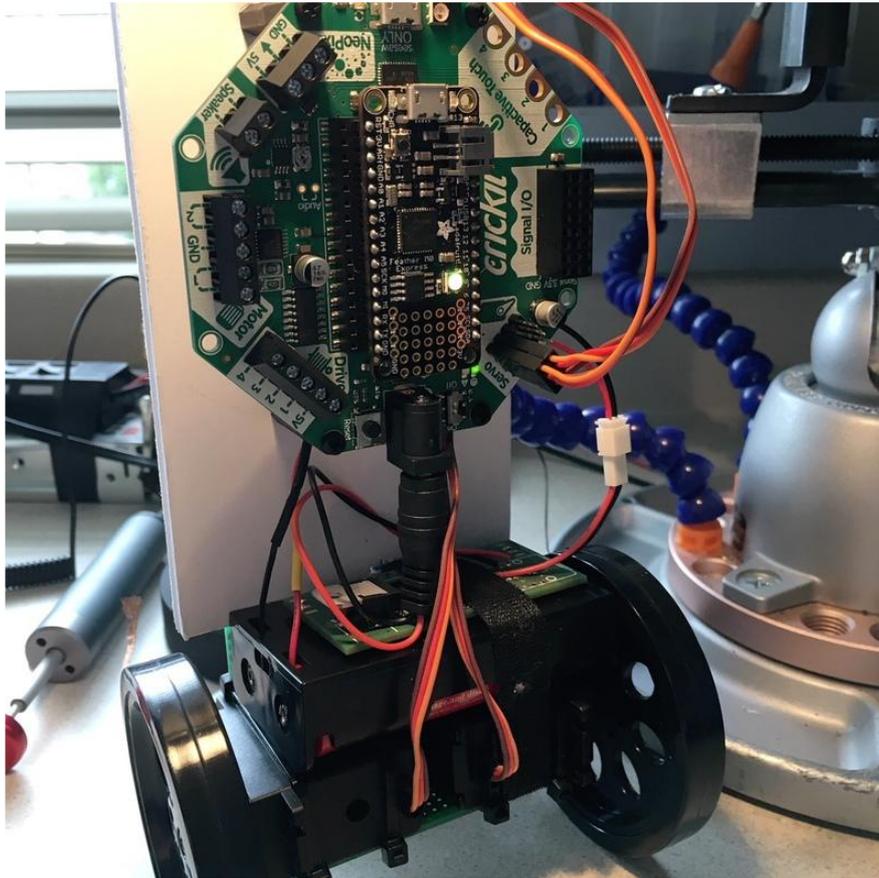
<https://learn.adafruit.com/adding-an-accelerometer-to-your-cricket-featherwing>

Last updated on 2023-08-29 03:45:03 PM EDT

Table of Contents

Overview	3
Assembly	4
Code	9

Overview



Making robots with the Crickit is awesome. When you pair it and all of its output capabilities with the Circuit Playground Express (CPX) and all of its sensors, you can really do a lot. If you are using the FeatherWing Crickit with a Feather, you have the same outputs, but none of the sensors that you just have when you use the CPX.

I am working on a project with the Crickit and wanted to use the Feather version to keep things compact and a little lighter. The thing is that I need an accelerometer. I could have used the Accelerometer FeatherWing I made in [this guide \(\)](#), But that would have made the whole thing bulkier that I wanted. Also, I figure it's a useful thing to have on a robotics board. As I found in that previous guide, the Flora breakouts for the LSM303 and LSM9DS0 fit nicely between the headers that the Feather plugs into. I had an LSM9DS0 sitting in a drawer so I used it for this project, but either will give you an accelerometer, and the pinouts of both Flora breakouts is identical. The only difference is importing the matching library.

The process was very similar to making that FeatherWing; even easier because of the connections to the Feather headers.

This project can be used with any Feather that is supported by the Crickit. I used the M0 Express since I wanted to write the code using CircuitPython.

You can read more about accelerometers in [this guide](#) ().

The Crickit is covered in detail in the [Introducing Adafruit Crickit guide](#) ().

1 x [Adafruit Feather M0 Express](#)

<https://www.adafruit.com/product/3403>

Feather board with an SAMD21 MCU and external flash.
Designed for running CircuitPython.

1 x [CRICKIT FeatherWing](#)

<https://www.adafruit.com/product/3343>

Crickit robotics board for Feather boards.

1 x [FLORA Accelerometer/Compass Sensor - LSM303](#)

<https://www.adafruit.com/product/1247>

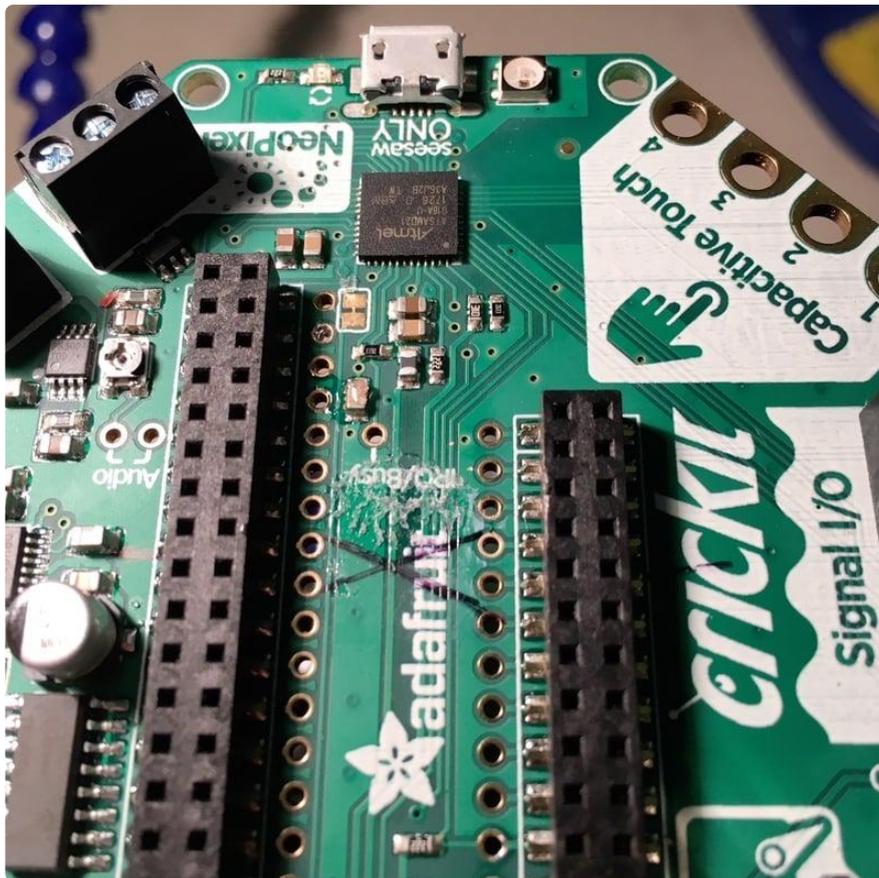
An LSM303 accelerometer and compass on a breakout designed for use with wearable Flora based projects.

Assembly

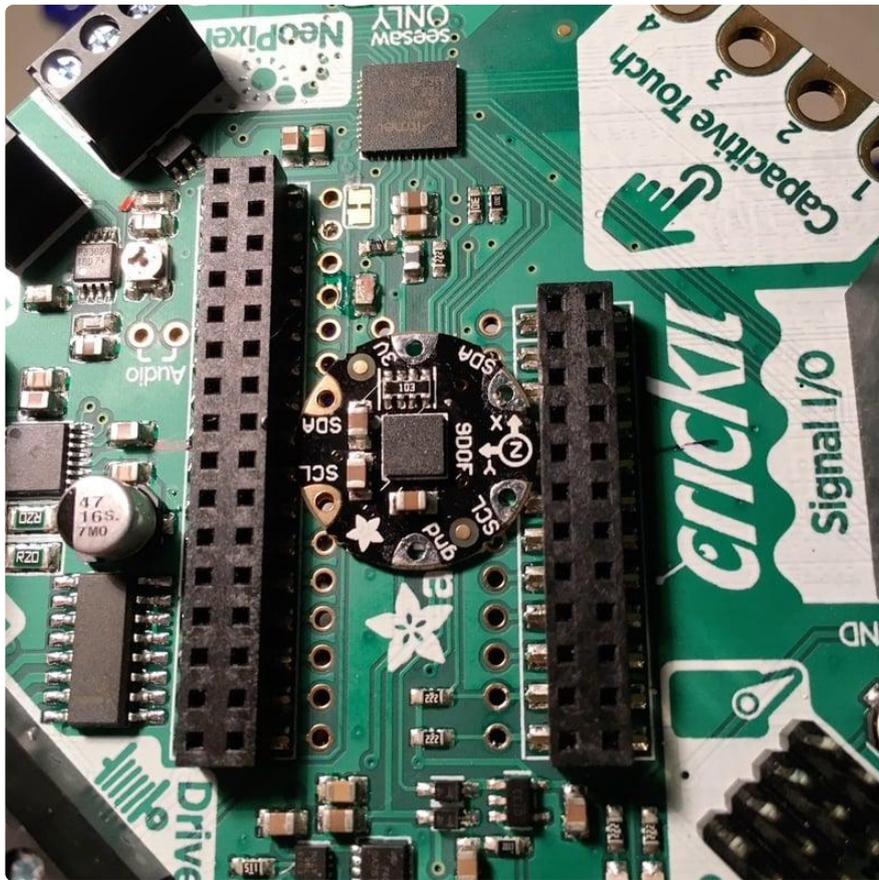
Since both Flora accelerometer breakouts have identical pinouts, these instructions work for either board.

The wiring is simple. Connect 3v, Gnd, SCL, and SDA on the accelerometer breakout to the corresponding connections on the Crickit. The FeatherWing Crickit has a hole for each Feather connection which makes this very easy to do.

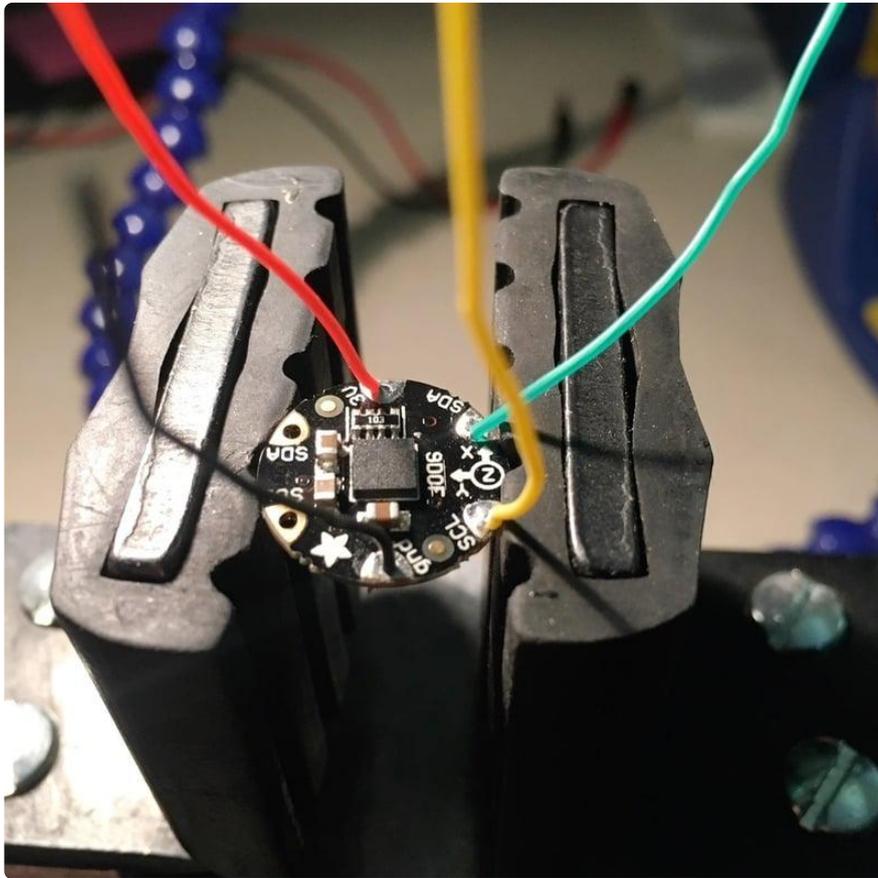
The first step is to find the center of the Crickit. We want accel chip centered so that all three axis of rotation run through the center point of the board, just like on the CPX. That's pretty easy: just find the intersection of the lines between opposite corners.



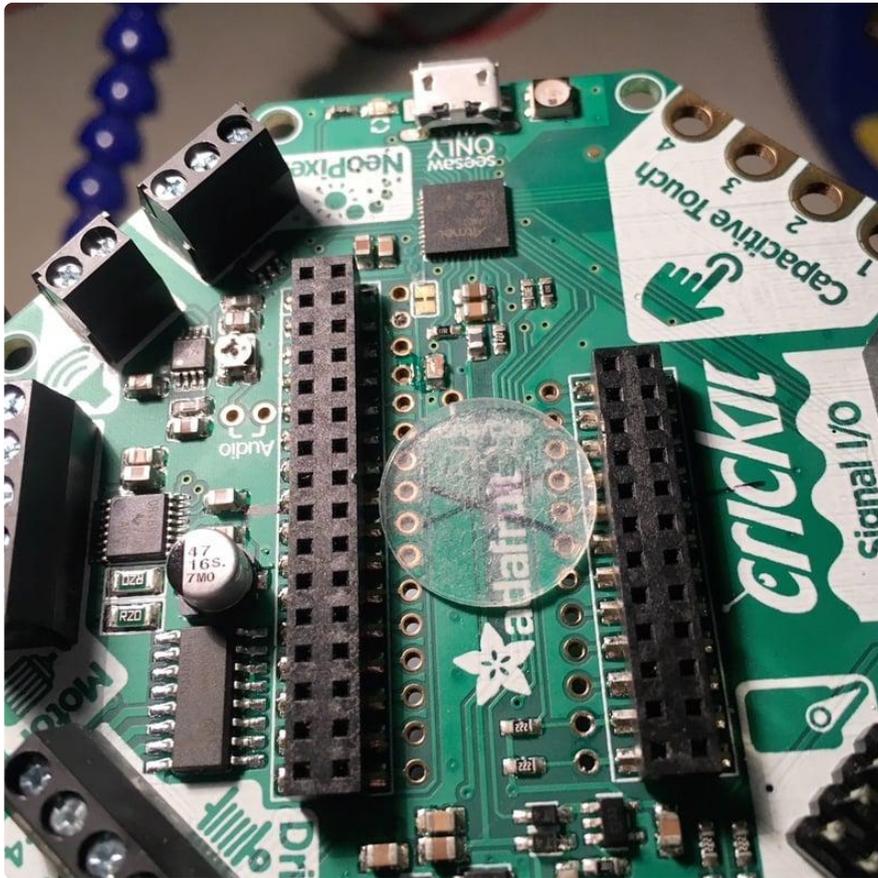
Next we need to decide how to orient the breakout. You'll want to line up the sensor axis along the natural axis of the Crickit, but also want to consider ease and neatness of the wiring. Since 3v and Gnd are on one side of the Feather (at opposite ends of the long header) and the I2C signals are at one end of the other side, it seemed reasonable to arrange it as show in the photo:



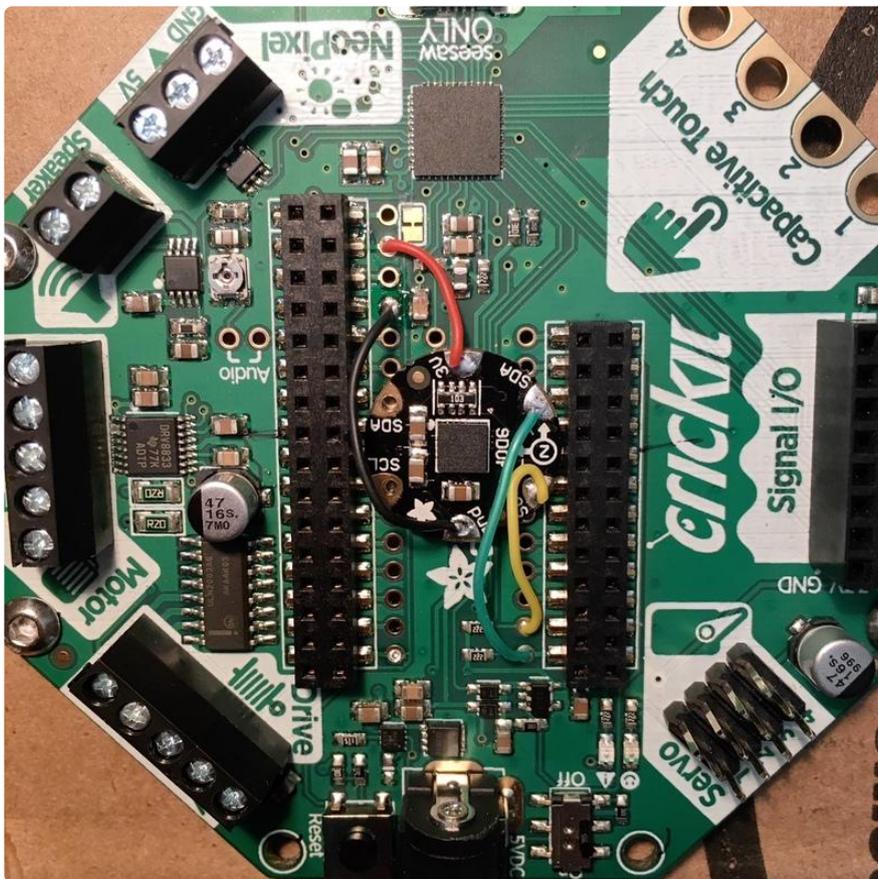
Now that we know where the sensor is going to be, let's wire it up. 30 gauge silicon coated wire is perfect for this sort of thing: it's thin and flexible and the coating doesn't melt at soldering temperatures. It's become my standard wire since I started using it.



The next step is mounting the breakout. To avoid any chance of shorting against the Crickit, I used a sticky rubber bumper, sticking it onto the Crickit where the breakout was going to be placed. Using a clear one let me get it in the right place (using the center mark).



Then it was just a matter of using a bit of superglue to attach the breakout to the rubber bumper. Once the glue sets, it's time to wire it to the Crickit.



That's it. It's ready for some code.

Code

I used CircuitPython for the example code. Are you new to using CircuitPython? No worries, [there is a full getting started guide here \(\)](#).

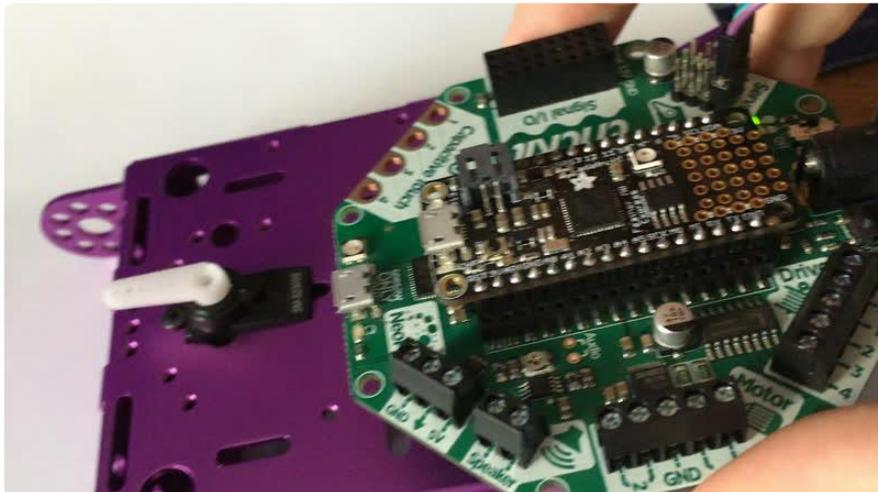
Adafruit suggests using the Mu editor to edit your code and have an interactive REPL in CircuitPython. [You can learn about Mu and installation in this tutorial \(\)](#).

We're using the `adafruit_lsm9ds0` library for CircuitPython to easily read the accelerometer.

You can learn about installing the `adafruit_lsm9ds0` library in the [CircuitPython Essentials Guide on CircuitPlayground Libraries \(\)](#). It is easiest to install the whole library package.

At this point the feather now has an accelerometer attached to it via I2C, just like in any other situation. Indeed, just like the CPX (other than that the CPX uses a LIS3DH).

As a demo, I hooked up a servo and controlled it using the accelerometer. The result is that the servo horn indicates the direction and magnitude of the Cricket's tilt.



```
# SPDX-FileCopyrightText: 2018 Dave Astels for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import busio
import board
from adafruit_seesaw.seesaw import Seesaw
from adafruit_seesaw.pwmout import PWMOut
from adafruit_motor import servo
import adafruit_lsm9ds0
```

```

# Setup hardware
i2c = busio.I2C(board.SCL, board.SDA)

sensor = adafruit_lsm9ds0.LSM9DS0_I2C(i2c)
seesaw = Seesaw(i2c)

# Create servo objects
pwm1 = PWMOut(seesaw, 17)
pwm1.frequency = 50
servo1 = servo.Servo(pwm1, min_pulse=500, max_pulse=2500)

# Center the servo
servo1.angle = 90

while True:
    # Read the accel
    x, y, z = sensor.acceleration

    # Clip the value
    if y < -10:
        y = -10
    if y > 10:
        y = 10

    # print(((y / 10) + 1) * 90)

    # Set the angle
    servo1.angle = ((-y / 10) + 1) * 90

    time.sleep(0.1)

```

The code simply reads the accelerometer and computes the servo angle based on the Y value it read. Note that I was using the Crickit build of CircuitPython 3.0.0rc0 which bundles the Crickit support libraries into the runtime to save RAM.

If you use a different Flora accelerometer breakout, change the `import` and the sensor creation line to reflect the library for that sensor.