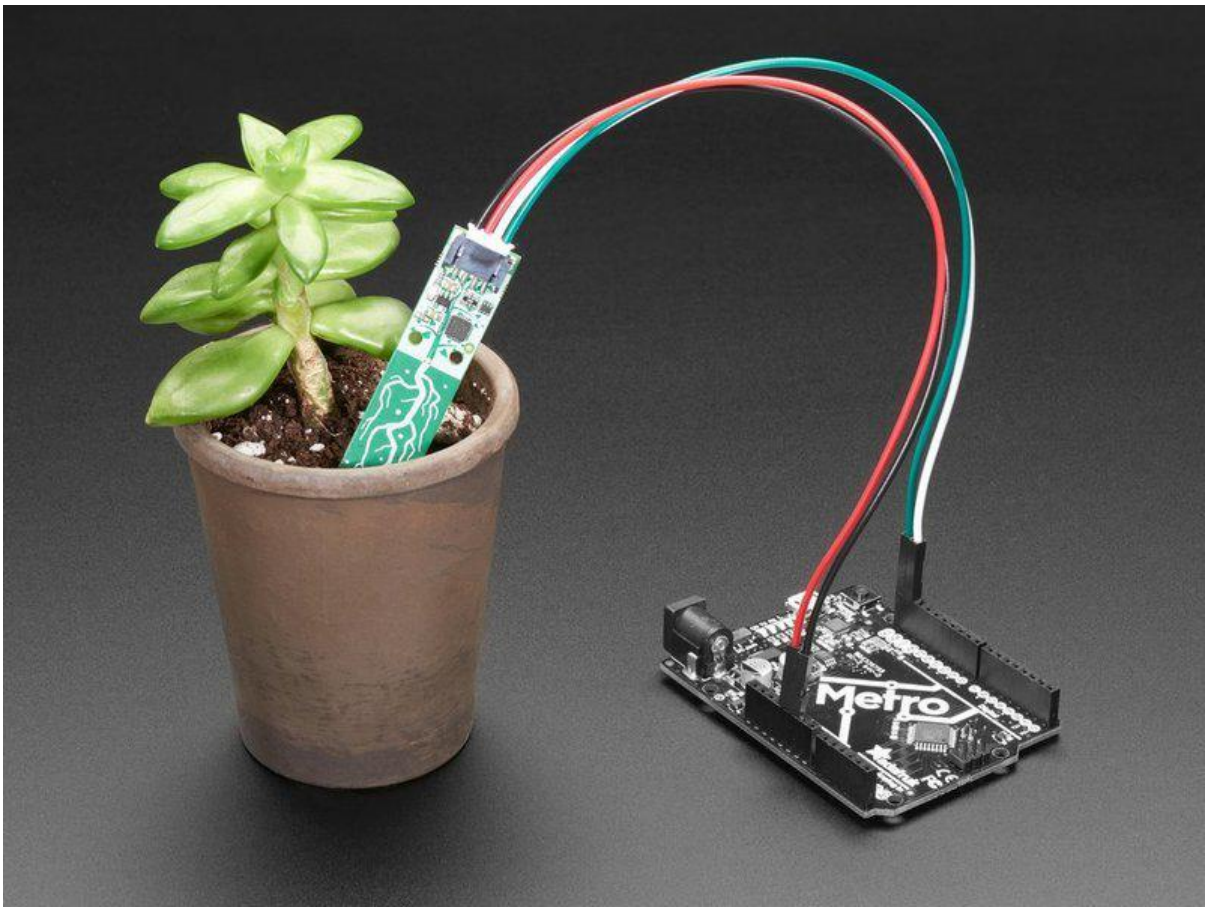
Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor

Created by lady ada



https://learn.adafruit.com/adafruit-stemma-soil-sensor-i2c-capacitive-moisture-sensor

Last updated on 2023-07-17 05:07:25 PM EDT

Table of Contents

Overview 3

Pinouts 5

Arduino Test 6

- Install Adafruit_seesaw library

Python & CircuitPython Test 8

- CircuitPython Microcontroller Wiring
- Python Computer Wiring
- CircuitPython Installation of seesaw Library
- Python Installation of seesaw Library
- CircuitPython & Python Usage

WipperSnapper Test 11

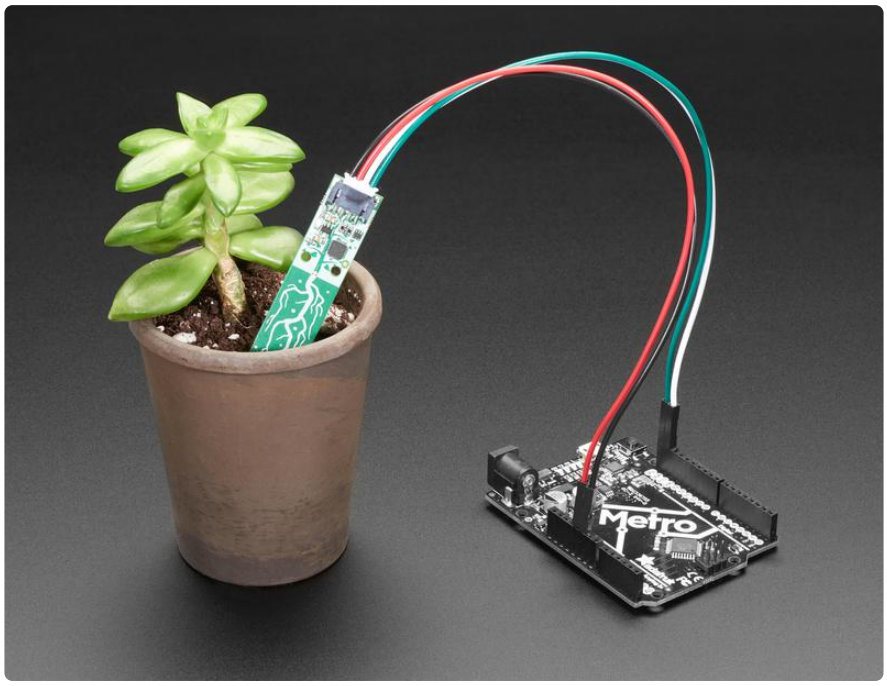
- What is WipperSnapper
- Wiring
- Usage

Downloads 18

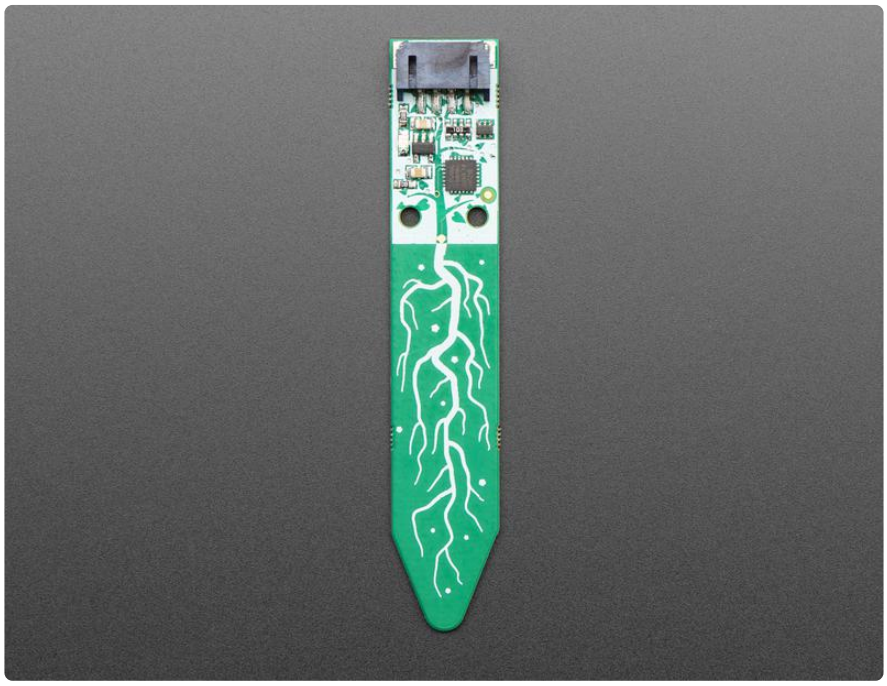
- Firmware & Files
- Schematic & Fabrication Print

FAQ 19

Overview

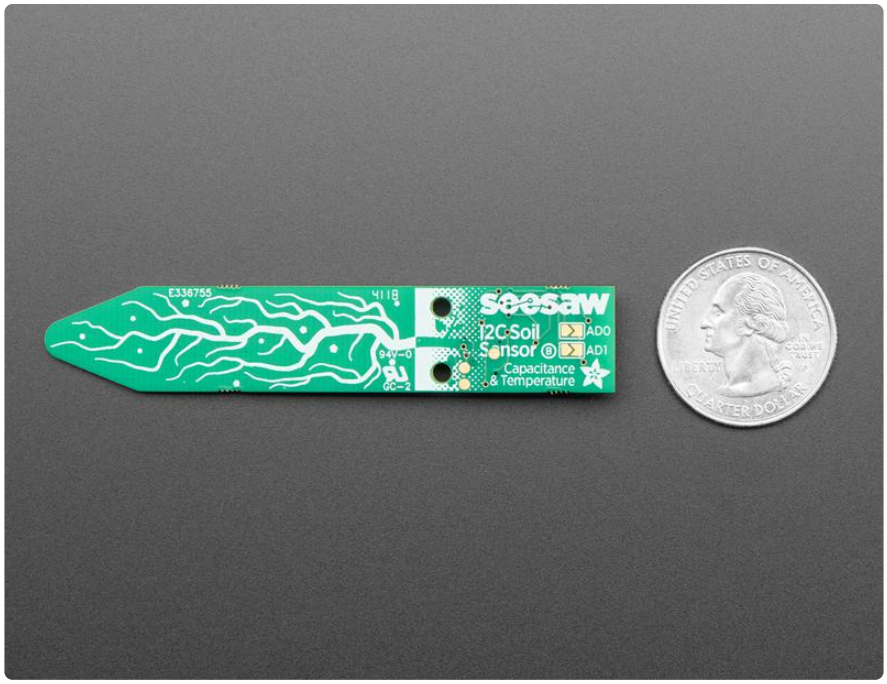


Most low cost soil sensors are resistive style, where there's two prongs and the sensor measures the conductivity between the two. These work OK at first, but eventually start to oxidize because of the exposed metal. Even if they're gold plated! The resistivity measurement goes up and up, so you constantly have to re-calibrate your code. Also, resistive measurements don't always work in loose soil.

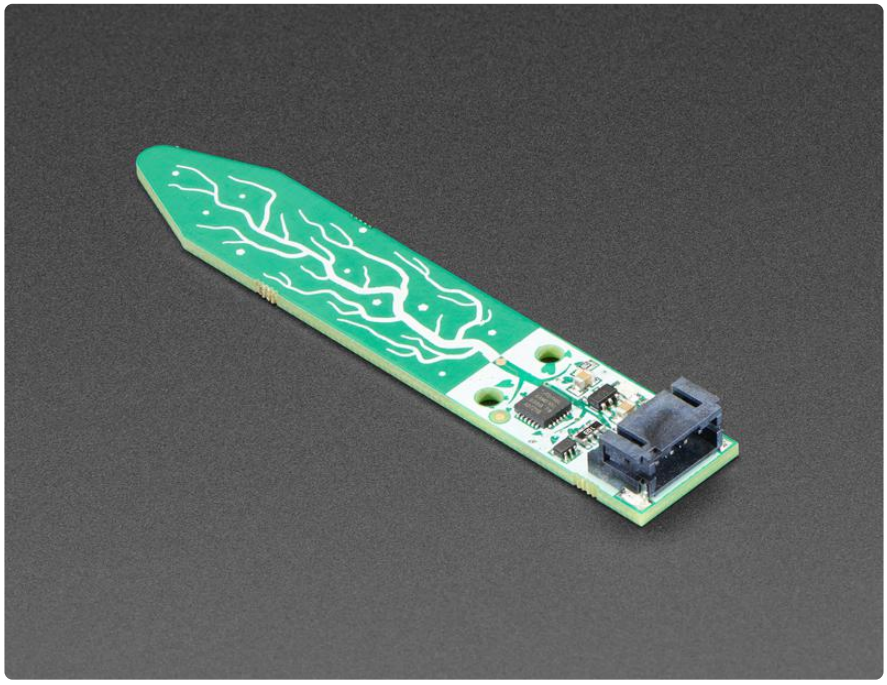


This design is superior with a capacitive measurement. Capacitive measurements use only one probe, don't have any exposed metal, and don't introduce any DC currents

into your plants. We use the built in capacitive touch measurement system built into the ATSAMD10 chip, which will give you a reading ranging from about 200 (very dry) to 2000 (very wet). As a bonus, we also give you the ambient temperature from the internal temperature sensor on the microcontroller, it's not high precision, maybe good to + or - 2 degrees Celsius.

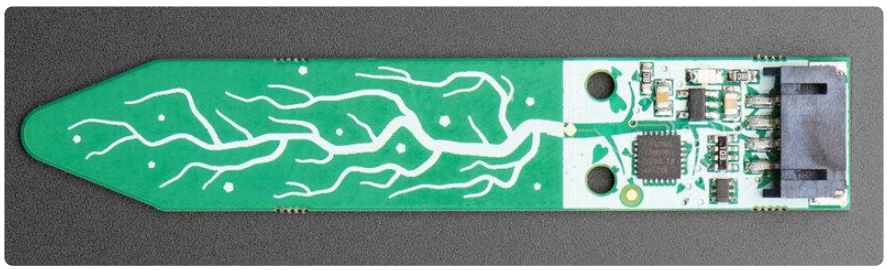


To make it so you can use the sensor with just about any microcontroller, we have an I2C interface. Connect a 4-pin JST-PH cable (we have a few stocked) to your microcontroller or single board computer to 3-5V power, Ground, I2C SDA and I2C SCL and then run our Arduino or CircuitPython code to read the temperature and capacitive measurement. No soldering required!



Please note: This is just the sensor, you'll also need a JST 4-PH cable to go along for plugging it in! We have a few options such as one with male header pins (), female header pins (), and a JST-JST cable (). You can use Seeed Grove cables and boards to connect - the Grove cable is a little different looking but will fit just fine.

Pinouts

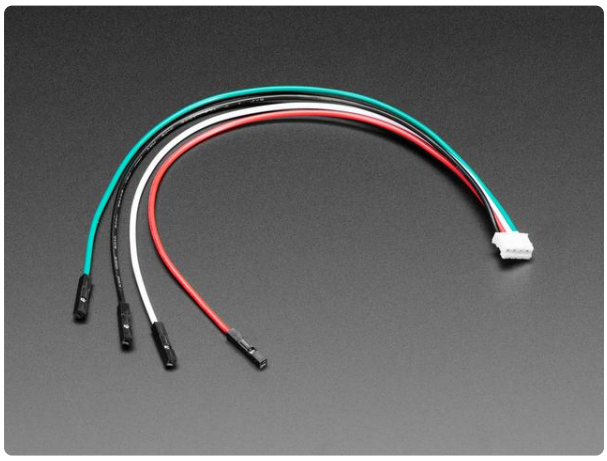


This sensor is pretty simple! Instead of headers, we have a 4-pin JST PH connector. This makes it easy to plug in a cable that goes to another board or prototyping kit.

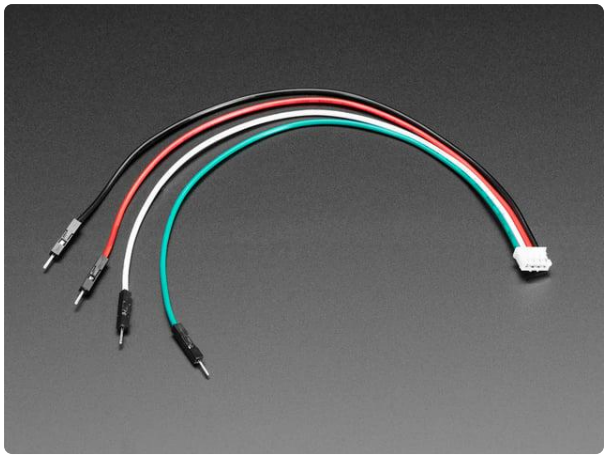
Looking at the image above, from top to bottom, the pinout is

- GND - power and logic ground
- VIN - 3-5V DC (use the same power voltage as you would for I2C logic)
- I2C SDA - there's a 10K pullup to VIN
- I2C SCL - there's a 10K pullup to VIN

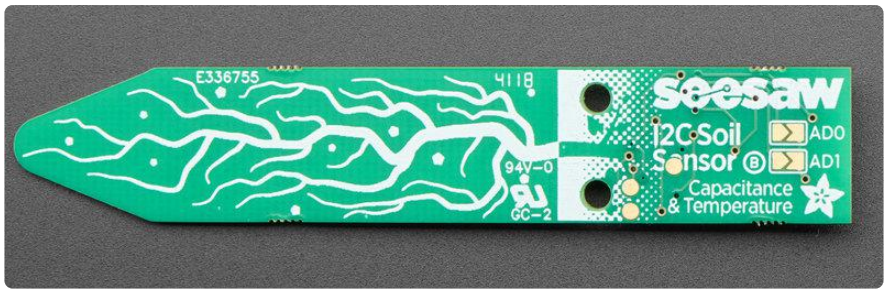
We suggest these cables, they're even color coded!



JST PH 2mm 4-Pin to Female Socket Cable - I2C STEMMA Cable
This cable will let you turn a JST PH 4-pin cable port into 4 individual wires with high-quality 0.1" female sockets on the end. We're carrying these to match up with any of...
https://www.adafruit.com/product/3950



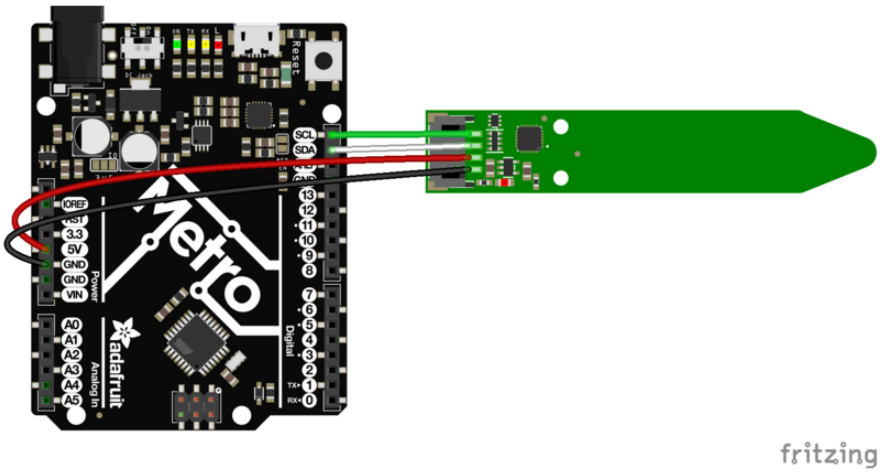
JST PH 2mm 4-Pin to Male Header Cable - I2C STEMMA Cable - 200mm
This cable will let you turn a JST PH 4-pin cable port into 4 individual wires with high-quality 0.1" male header plugs on the end. We're carrying these to match up with any...
https://www.adafruit.com/product/3955



On the back there are 3 SWD programming dots (see the schematic for the exact pinout in case you want to re-program the chip)

There's also two I2C address jumpers. The default address is 0x36, but you can select between 0x36 and 0x39 inclusive by shorting AD0 (to add 1 to the address) and/or A D1 (to add 2 to the address)

Arduino Test



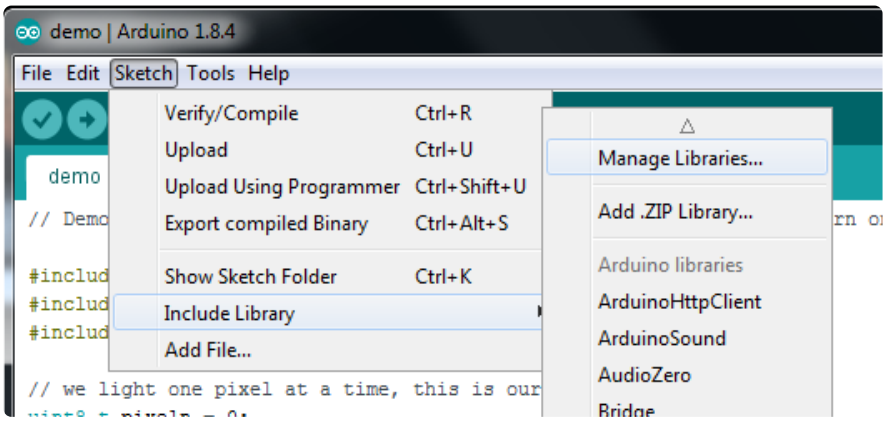
- Connect Vin Red Wire to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduino's, that is 5V. If you have 3.3V logic, use 3V

- Connect GND Black Wire to common power/data ground
- Connect the SCL Green Wire pin to the I2C clock SCL pin on your Arduino. On an UNO & '328 based Arduino, this is also known as A5, on a Mega it is also known as digital 21 and on a Leonardo/Micro, digital 3
- Connect the SDA White Wire pin to the I2C data SDA pin on your Arduino. On an UNO & '328 based Arduino, this is also known as A4, on a Mega it is also known as digital 20 and on a Leonardo/Micro, digital 2

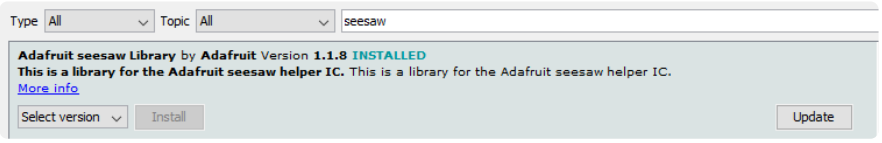
Install Adafruit_seesaw library

To begin reading sensor data, you will need to install the Adafruit_seesaw library (code on our github repository) (). It is available from the Arduino library manager so we recommend using that.

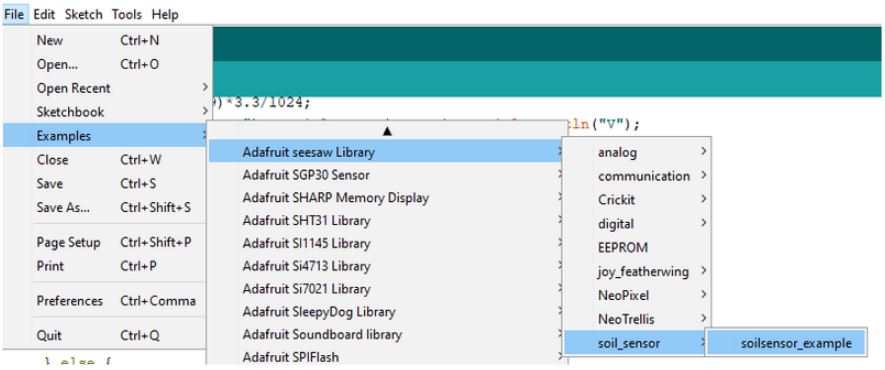
From the IDE open up the library manager...



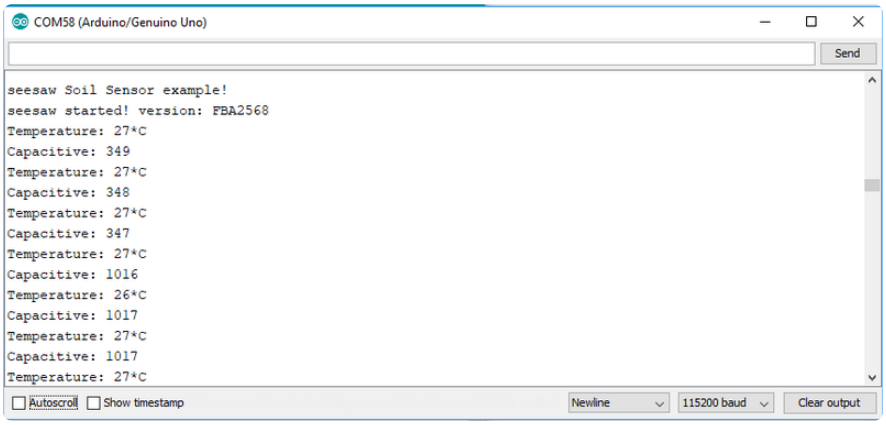
Search for Adafruit seesaw, and click Install



Now from within the IDE, open up the Adafruit seesaw Soil Sensor Example sketch:



Open up the serial console at 115200 baud to see the temperature and capacitive reading



Try touching the body of the sensor to make the 'moisture' measurement go up. In soil, you'll see this range from about 300 to 500, but make sure to test it by placing in a pot of soil and looking at the serial console to get an idea of what the range is. Note that it does depend on how packed/loose the soil is!

The temperature reading is only on the chip itself. It's not terribly precise but it is within a few degrees. Good for getting a basic sense of how the temperature is moving around the house

---

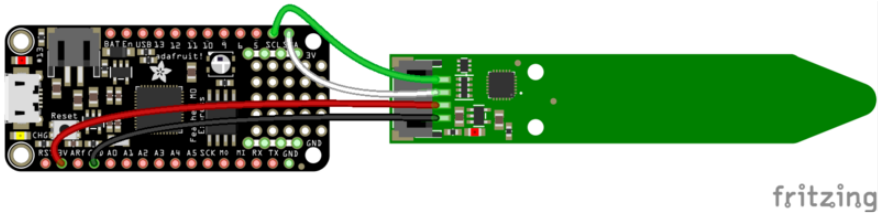
Python & CircuitPython Test

It's easy to use the seesaw soil sensor with Python or CircuitPython and the Adafruit CircuitPython seesaw () module. This module allows you to easily write Python code that reads the moisure/temperature from the sensor.

You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library ().

CircuitPython Microcontroller Wiring

First wire up the soil sensor to your board exactly as shown on the previous pages for Arduino. Here's an example of wiring a Feather M0 to the sensor with I2C:

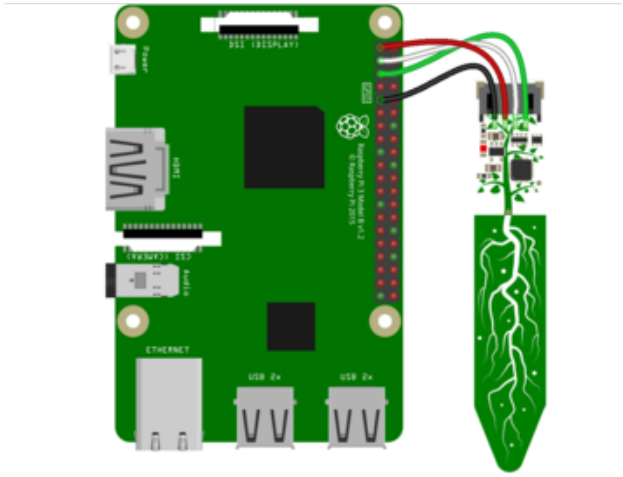


- Board 3V to sensor VIN
- Board GND to sensor GND
- Board SCL to sensor SCL
- Board SDA to sensor SDA

Python Computer Wiring

Since there's dozens of Linux computers/boards you can use we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported ().

Here's the Raspberry Pi wired with I2C:



Pi 3V3 to sensor VIN
Pi GND to sensor GND
Pi SCL to sensor SCL
Pi SDA to sensor SDA

CircuitPython Installation of seesaw Library

You'll need to install the Adafruit CircuitPython seesaw () library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython () for your board.

Next you'll need to install the necessary libraries to use the hardware. They can be downloaded via the Adafruit's CircuitPython library bundle (). Be sure to read the CircuitPython Welcome guide () for information on working with and installing libraries.

Before continuing make sure your board's lib folder has:

- adafruit_seesaw.mpy
- adafruit_bus_device

Next connect to the board's serial REPL ()so you are at the CircuitPython >>> prompt.

Python Installation of seesaw Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready ()!

Once that's done, from your command line run the following command:

- `sudo pip3 install adafruit-circuitpython-seesaw`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

This example sketch shows connecting to the sensor and reading temperature and moisture/capacitive measurement

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time

import board

from adafruit_seesaw.seesaw import Seesaw

i2c_bus = board.I2C()  # uses board.SCL and board.SDA
# i2c_bus = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller

ss = Seesaw(i2c_bus, addr=0x36)
```

```
while True:
    # read moisture level through capacitive touch pad
    touch = ss.moisture_read()

    # read temperature from the temperature sensor
    temp = ss.get_temp()

    print("temp: " + str(temp) + "  moisture: " + str(touch))
    time.sleep(1)
```

```
Adafruit CircuitPython REPL
main.py output:
temp: 23.6042  moisture: 343
temp: 23.2944  moisture: 348
temp: 23.2944  moisture: 347
temp: 23.5009  moisture: 1015
temp: 23.9204  moisture: 1015
temp: 23.7075  moisture: 335
temp: 24.1269  moisture: 332
```

Try touching the body of the sensor to make the 'moisture' measurement go up. In soil, you'll see this range from about 300 to 500, but make sure to test it by placing in a pot of soil and looking at the serial console to get an idea of what the range is. Note that it does depend on how packed/loose the soil is!

The temperature reading is only on the chip itself. It's not terribly precise but it is within a few degrees. Good for getting a basic sense of how the temperature is moving around the house

---

WipperSnapper Test

What is WipperSnapper

WipperSnapper is a firmware designed to turn any WiFi-capable board into an Internet-of-Things device without programming a single line of code. WipperSnapper connects to Adafruit IO (), a web platform designed (by Adafruit! ()) to display, respond, and interact with your project's data.

Simply load the WipperSnapper firmware onto your board, add credentials, and plug it into power. Your board will automatically register itself with your Adafruit IO account.

From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware,

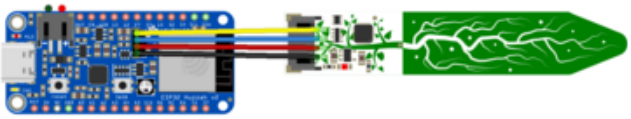
so you can immediately start interacting, logging, and streaming the data your projects produce without writing code.

If you've never used WipperSnapper, click below to read through the quick start guide before continuing.

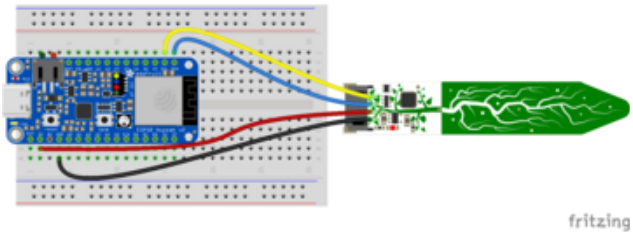
Quickstart: Adafruit IO WipperSnapper

Wiring

First, wire up the soil sensor to your board exactly as follows. Here is an example of the soil sensor wired to an Adafruit ESP32 Feather V2 () using I2C with a STEMMA QT cable (no soldering required) ()



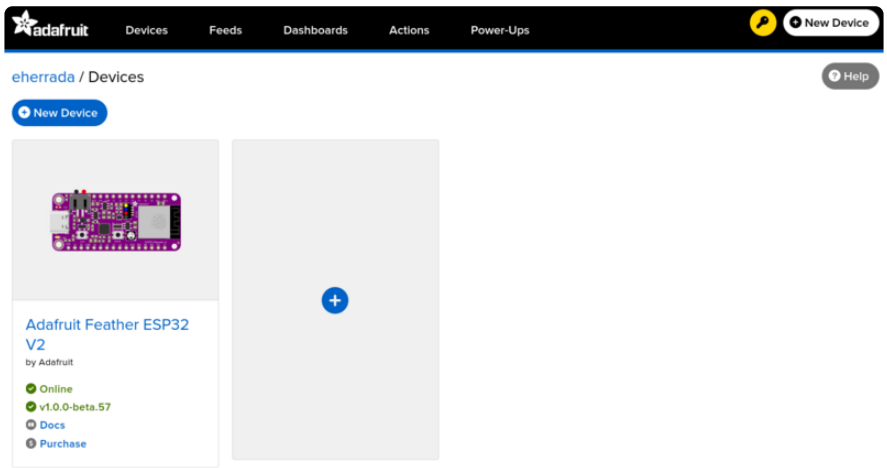
Board 3V to sensor VIN (red wire on STEMMA QT)
Board GND to sensor GND (black wire on STEMMA QT)
Board SCL to sensor SCL (yellow wire on STEMMA QT)
Board SDA to sensor SDA (blue wire on STEMMA QT)



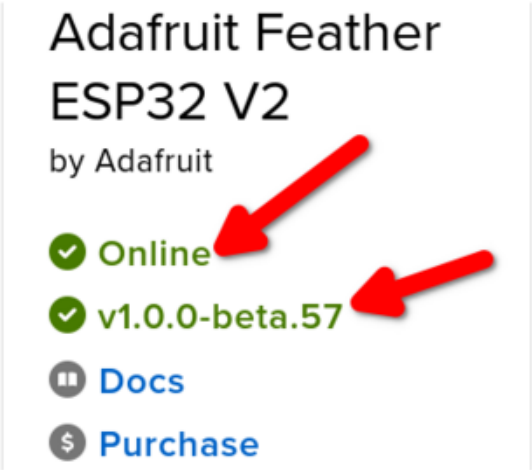
Usage

Connect your board to Adafruit IO Wippersnapper and navigate to the WipperSnapper board list ().

On this page, select the WipperSnapper board you're using to be brought to the board's interface page.



If you do not see your board listed here - you need to connect your board to Adafruit IO () first.

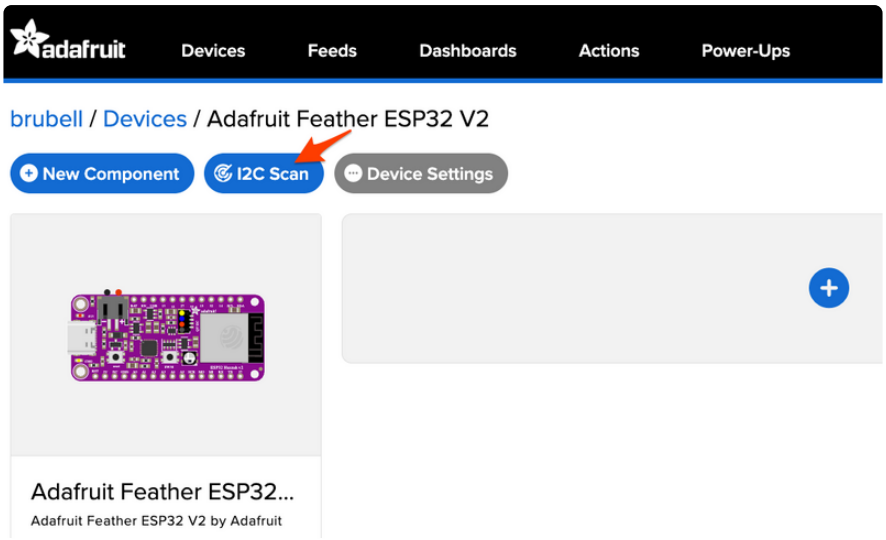


On the device page, quickly check that you're running the latest version of the WipperSnapper firmware.

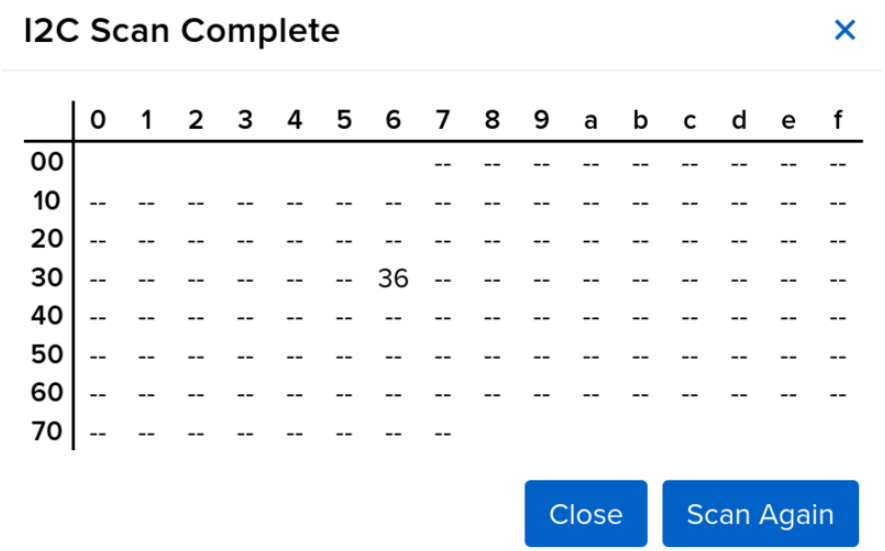
The device tile on the left indicates the version number of the firmware running on the connected board.

If the firmware version is green with a checkmark - continue with this guide.
If the firmware version is red with an "X" - update to the latest WipperSnapper firmware () on your board before continuing.

Next, make sure the sensor is plugged into your board and click the I2C Scan button.



You should see the soil sensor's default I2C address of `0x36` pop-up in the I2C scan list.



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00								--	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	36	--	--	--	--	--	--	--	--	--
40	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--								

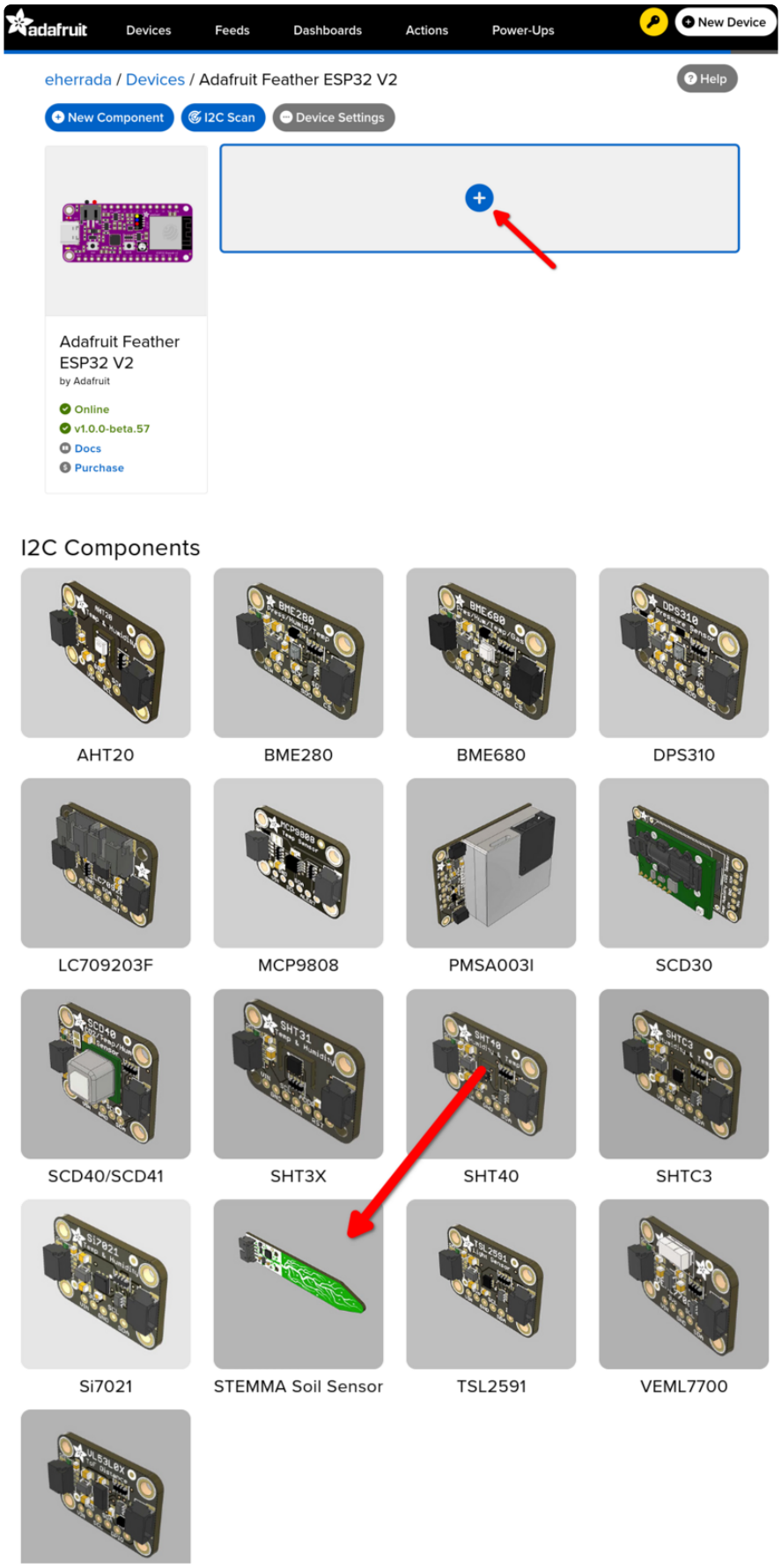
I don't see the sensor's I2C address listed!

First, double-check the connection and/or wiring between the sensor and the board.

Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

With the sensor detected in an I2C scan, you're ready to add the sensor to your board.

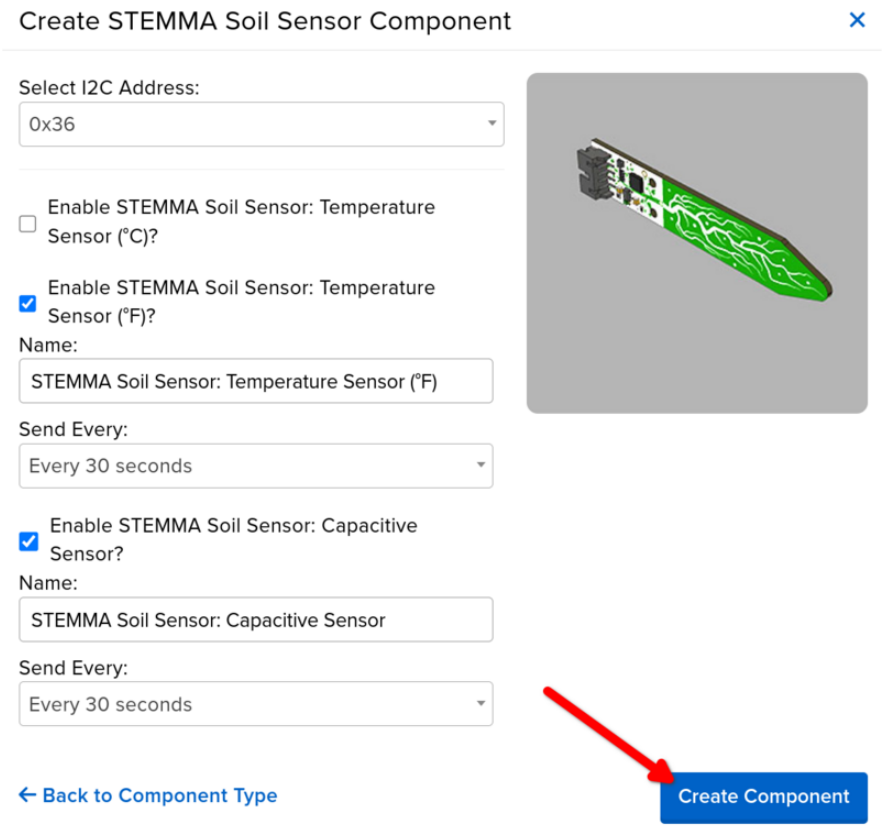
Click the New Component button or the + button to bring up the component picker.



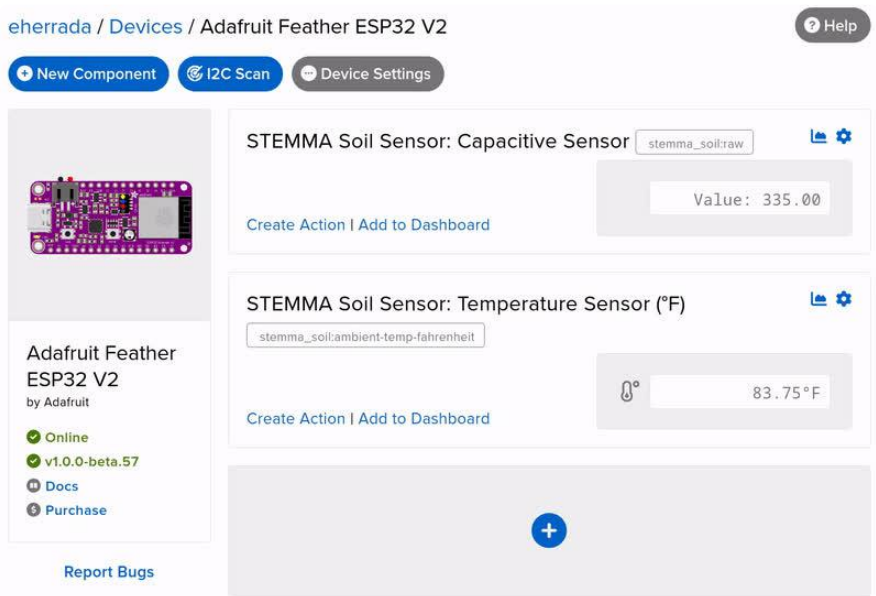
On the component configuration page, the soil sensor's address should be listed along with the sensor's settings.

The Send Every option is specific to each sensor's measurements. This option will tell the Feather how often it should read from the two sensors and send the data to Adafruit IO. Measurements can range from every 30 seconds to every 24 hours.

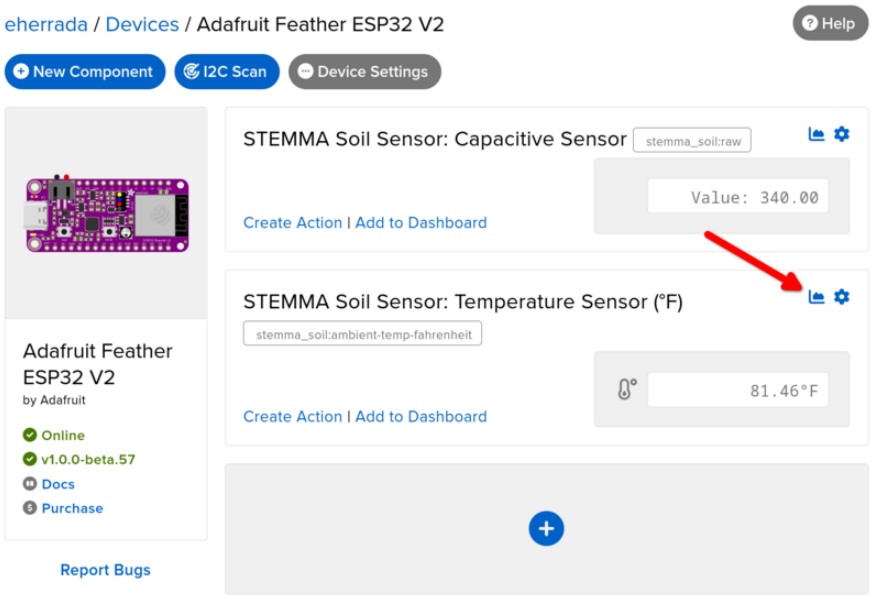
For this example, set the Send Every interval for each sensor to "Every 30 seconds".



Your device interface should now show the sensor components you created. After the interval you configured elapses, WipperSnapper will automatically read values from the sensor(s) and send them to Adafruit IO.

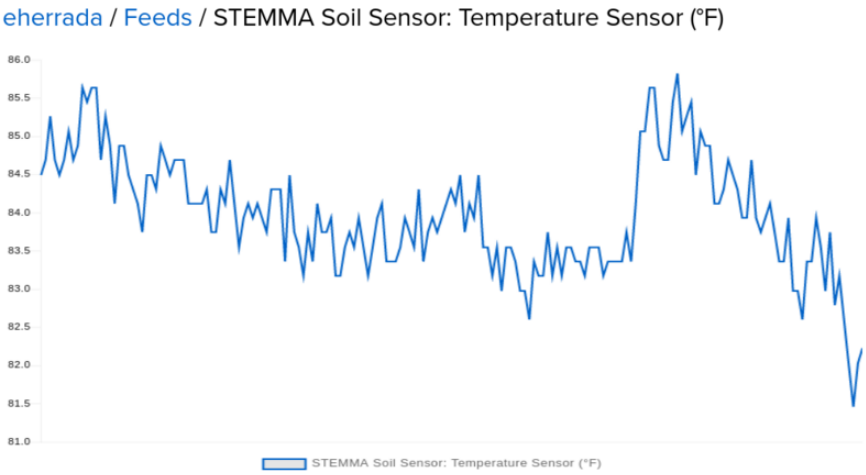


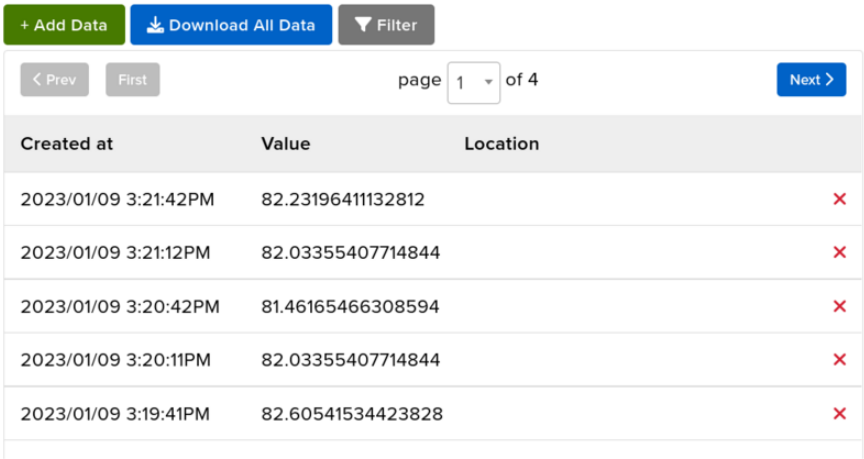
To view the data that has been logged from the sensor, click on the graph next to the sensor name.



Here you can see the feed history and edit things about the feed such as the name, privacy, webhooks associated with the feed and more. If you want to learn more about how feeds work, check out this page ().

The soil sensor has two sensors. The capacitive one has one feed and the temperature sensor has two. This picture shows the Celsius feed, but if you click on the graph icon for the different sensors, you'll see their feed history.





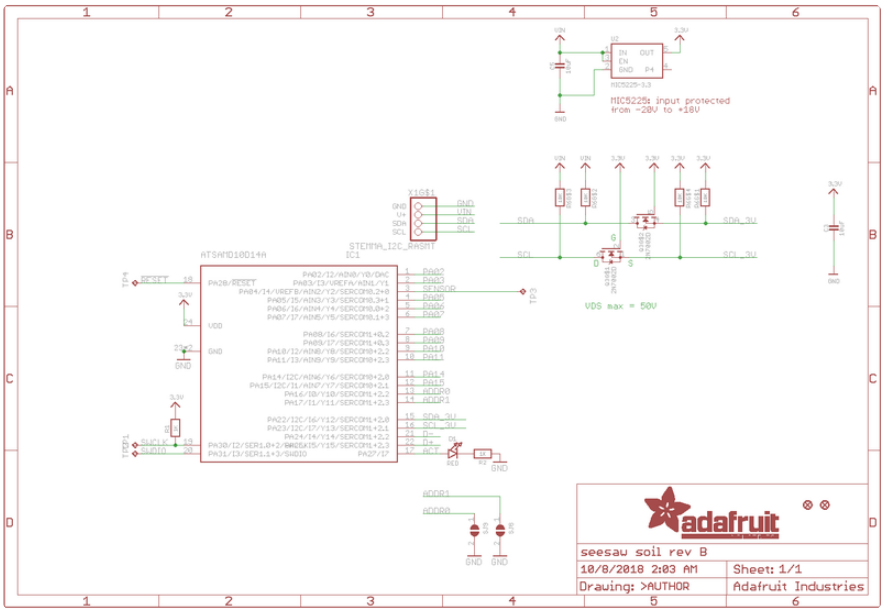
For IO Free accounts, feed data is stored for a maximum of 30 days and there's a maximum of 10 feeds. In this guide, you created three feeds. If you'd like to store data for more than 30 days, increase the number of feeds (components) you can use with WipperSnapper, or increase your data rate to send more sensor measurements to Adafruit IO - upgrade your account to Adafruit IO Plus ().

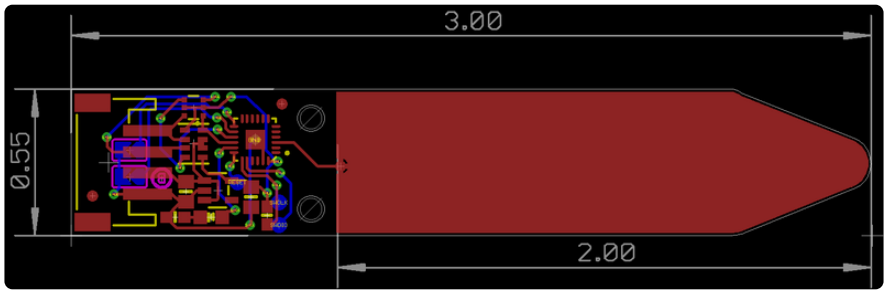
Downloads

Firmware & Files

- Adafruit seesaw firmware ()
- CircuitPython seesaw library ()
- Arduino seesaw library ()
- CAD files in GitHub ()
- 3D Models on GitHub ()

Schematic & Fabrication Print





FAQ

How do I calibrate my STEMMA Soil Sensor to return volumetric water content (VWC)?

You can't. This soil sensor provides only a capacitive measurement. You could try to calibrate it against a known source, but this is NOT guaranteed to work.

What are the seesaw registers for the soil moisture value?

This is only needed if you are not using the Arduino or CircuitPython library. See the seesaw guide for general information on reading seesaw registers. () The STEMMA Soil Sensor's reading is the capacitive measurement on channel 0. Base register = 0x0F and Function Register = 0x10.