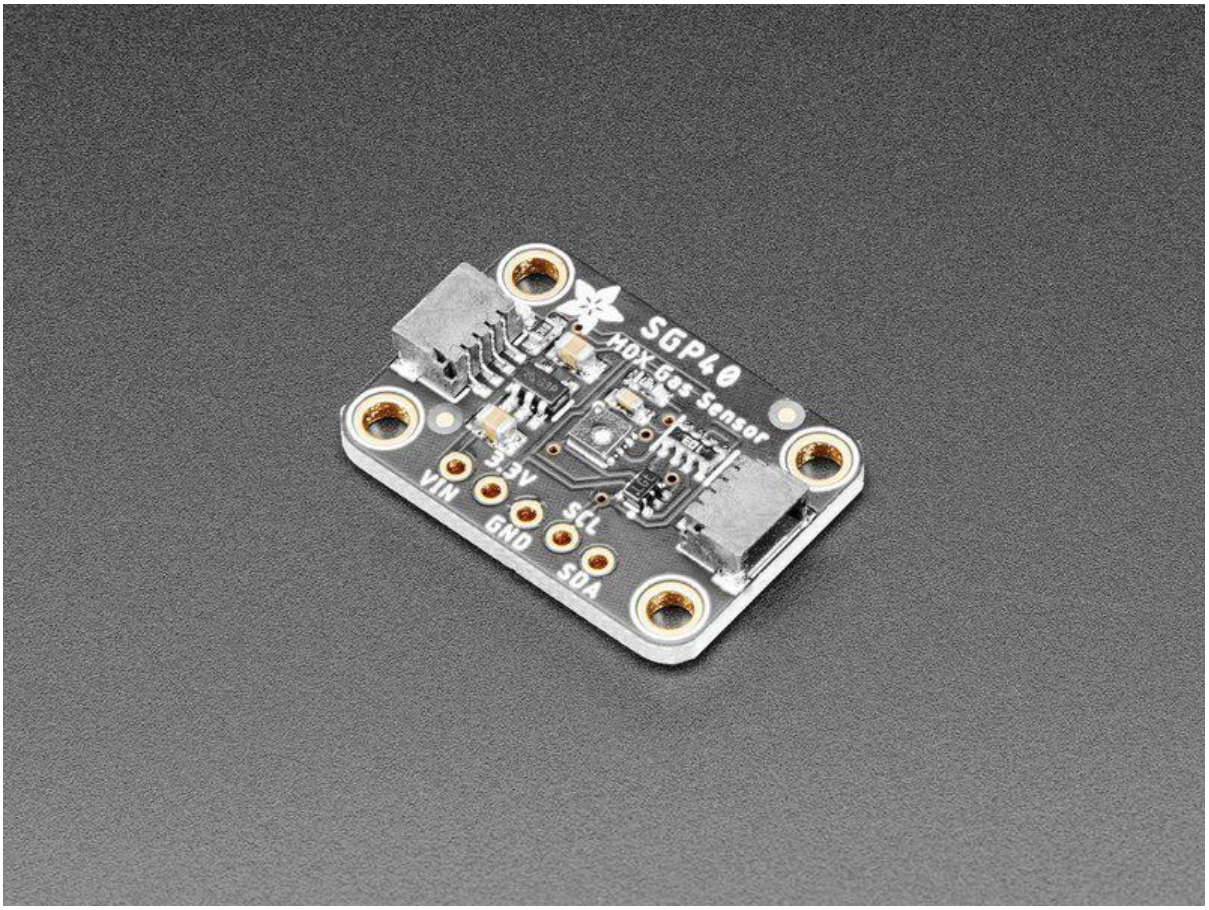
Adafruit SGP40 Air Quality Sensor

Created by Kattni Rembor

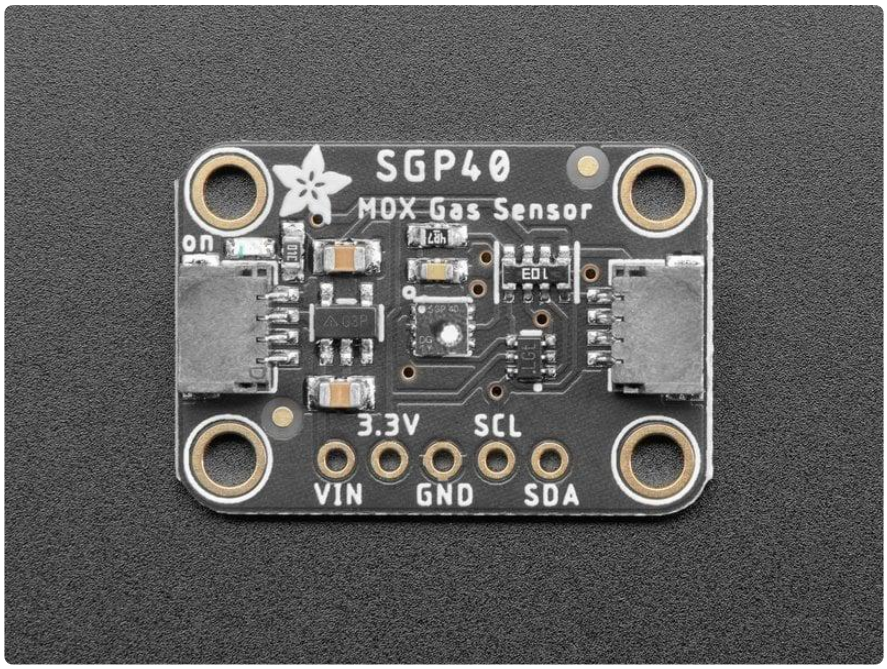


https://learn.adafruit.com/adafruit-sgp40

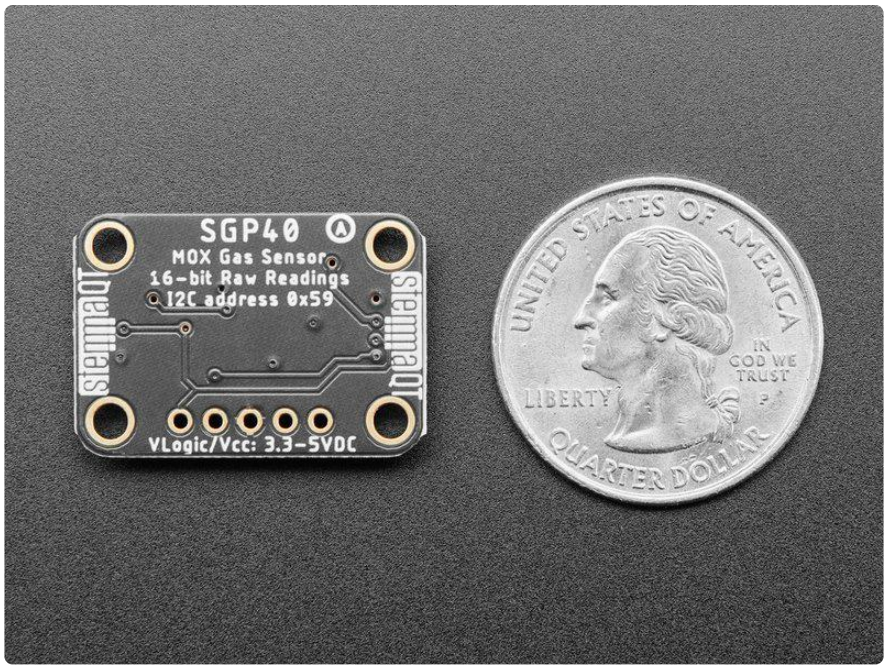
Last updated on 2023-09-21 12:39:29 PM EDT

Table of Contents

Overview



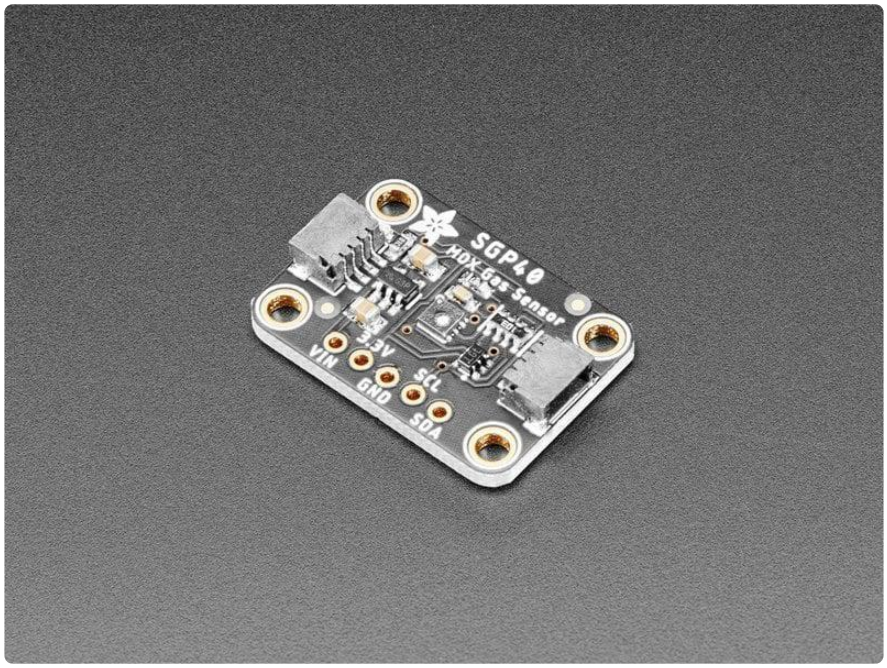
sniff *sniff* ... do you smell that? No need to stick your nose into a carton of milk anymore, you can build a digital nose with the SGP40 Multi-Pixel Gas Sensor, () a fully integrated MOX gas sensor. This is a very fine air quality sensor from the sensor experts at Sensirion, with I2C interfacing so you don't have to manage the heater and analog reading of a MOX sensor. It combines multiple metal-oxide sensing and heating elements on one chip to provide more detailed air quality signals.



The SGP40 has a 'standard' hot-plate MOX sensor, as well as a small microcontroller that controls power to the plate, reads the analog voltage and provides an I2C

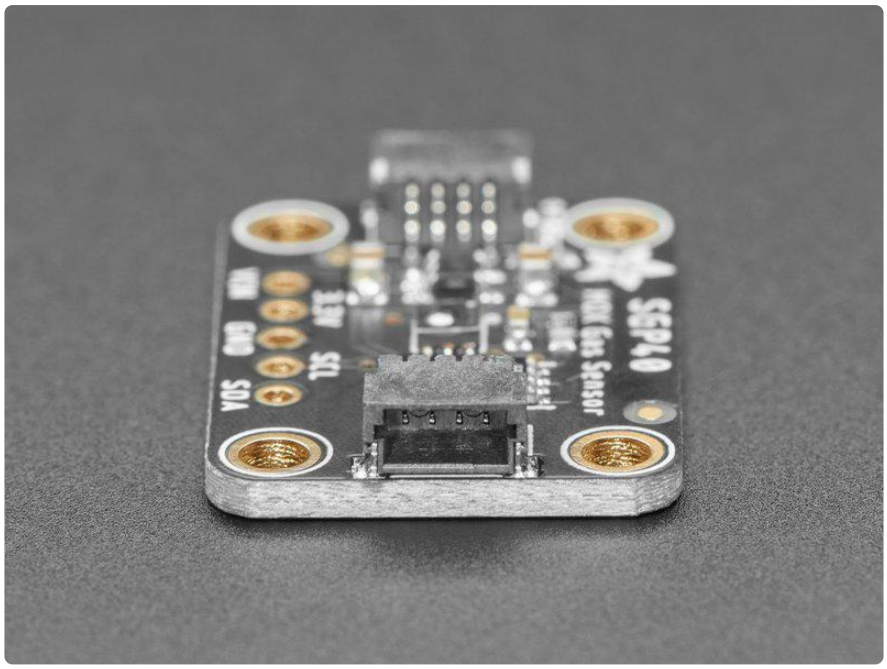
interface to read from. Unlike the CCS811, this sensor does not require I2C clock stretching. We currently have an Arduino library with examples of reading the raw value and also running the Sensirion algorithm to calculate VOC index () as well as a CircuitPython / Python library ().

This is a gas sensor that can detect a wide range of Volatile Organic Compounds (VOCs) and H2 and is intended for indoor air quality monitoring. The SGP40 is the next generation after the SGP30, but does not give TVOC/eCO2 values out like the SGP30. Instead, raw signal from the sensor is processed using their software algorithm to give an overall 'air quality' value form 0 to 500 ().



Please note, this sensor, like all VOC/gas sensors, has variability, and to get precise measurements you will want to calibrate it against known sources! That said, for general environmental sensors, it will give you a good idea of trends and comparison.

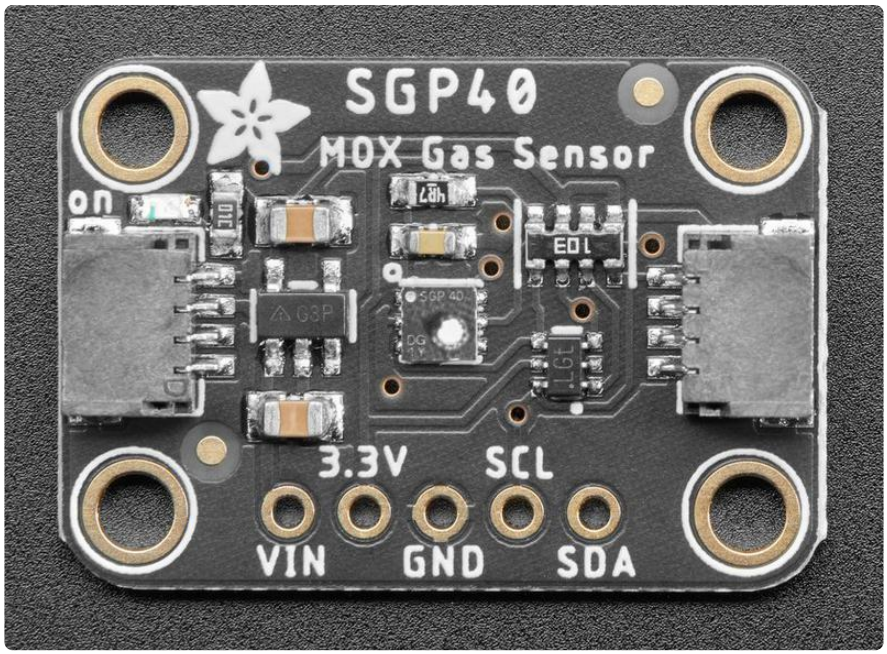
Another nice element to this sensor is the ability to set humidity compensation for better accuracy. An external humidity sensor is required and then the RH% is written over I2C to the sensor (), so it can better calibrate the MOX sensor reading and reduce humidity/temperature-based variations..



Nice sensor right? So we made it easy for you to get right into your next project. The surface-mount sensor is soldered onto a custom made PCB in the STEMMA QT form factor (), making them easy to interface with. The STEMMA QT connectors () on either side are compatible with the SparkFun Qwiic () I2C connectors. This allows you to make solderless connections between your development board and the SGP40 or to chain it with a wide range of other sensors and accessories using a compatible cable ( ). QT Cable is not included, but we have a variety in the shop ()

We've of course broken out all the pins to standard headers and added a 3.3V voltage regulator and level shifting so allow you to use it with either 3.3V or 5V systems such as the Arduino Uno, or Feather M4, or Raspberry Pi.

Pinouts



Power Pins:

- Vin - this is the power pin. Since the sensor chip uses 3 VDC for logic, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- 3.3V - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- GND - common ground for power and logic

Data Pins

- SCL - I2C clock pin, connect to your microcontrollers I2C clock line. Can use 3V or 5V logic, and has a 10K pullup to Vin
- SDA - I2C data pin, connect to your microcontrollers I2C data line. Can use 3V or 5V logic, and has a 10K pullup to Vin
- STEMMA QT () - These connectors allow you to connectors to dev boards with S TEMMA QT connectors or to other things with various associated accessories ()

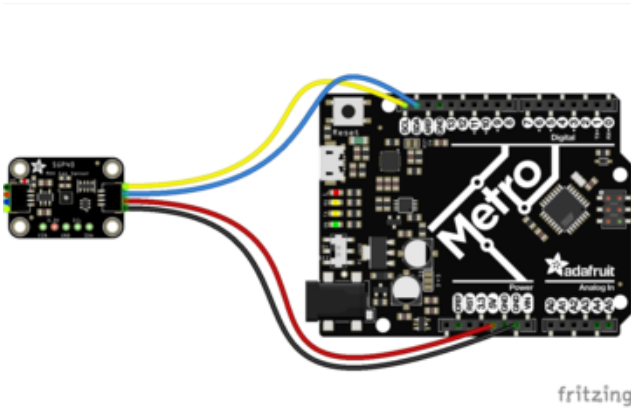
Arduino

Using the SGP40 with Arduino is a simple matter of wiring up the sensor to your Arduino-compatible microcontroller, installing the Adafruit SGP40 () library we've written, and running the provided example code.

I2C Wiring

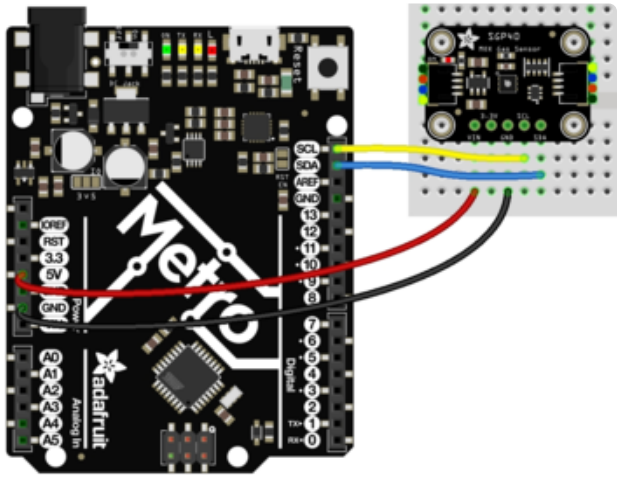
Use this wiring if you want to connect via I2C interface. The default I2C address for the SGP40 is 0x59.

Here is how to wire up the sensor using one of the STEMMA QT () connectors. The examples show a Metro but wiring will work the same for an Arduino or other compatible board.



Connect board VIN (red wire) to Arduino 5V if you are running a 5V Arduino (Uno, etc.). If your Arduino is 3V, connect to that instead.
Connect board GND (black wire) to Arduino GND
Connect board SCL (yellow wire) to Arduino SCL
Connect board SDA (blue wire) to Arduino SDA

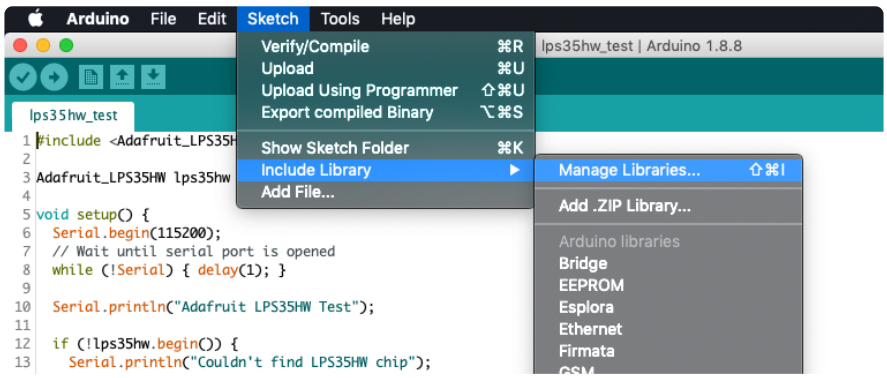
Here is how to wire the sensor to a board using a solderless breadboard:



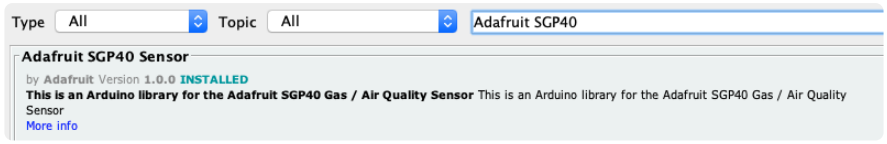
Connect board VIN (red wire) to Arduino 5V if you are running a 5V Arduino (Uno, etc.). If your Arduino is 3V, connect to that instead.
Connect board GND (black wire) to Arduino GND
Connect board SCL (yellow wire) to Arduino SCL
Connect board SDA (blue wire) to Arduino SDA

Library Installation

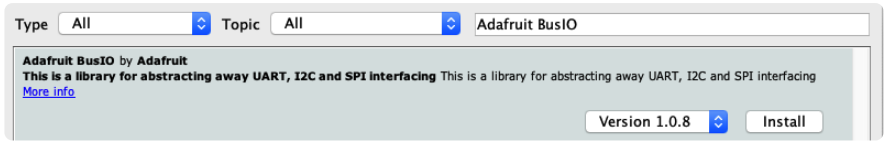
You can install the Adafruit SGP40 library for Arduino using the Library Manager in the Arduino IDE.



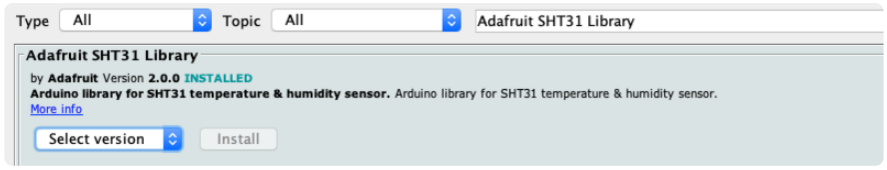
Click the Manage Libraries ... menu item, search for Adafruit SGP40 , and select the A dafruit SGP40 library:



Follow the same process for the Adafruit BusIO library.

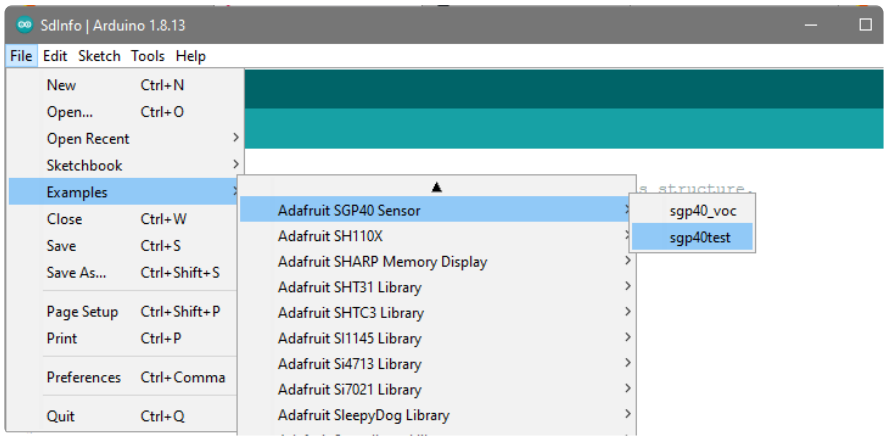


Finally follow the same process for the Adafruit SHT31 Library:



Load Basic Example

Open up File -> Examples -> Adafruit SGP40 -> sgp40test



After opening the demo file, upload to your Arduino wired up to the sensor. Once you upload the code, you will see the Raw measurement values being printed when you open the Serial Monitor (Tools->Serial Monitor) at 115200 baud, similar to this:

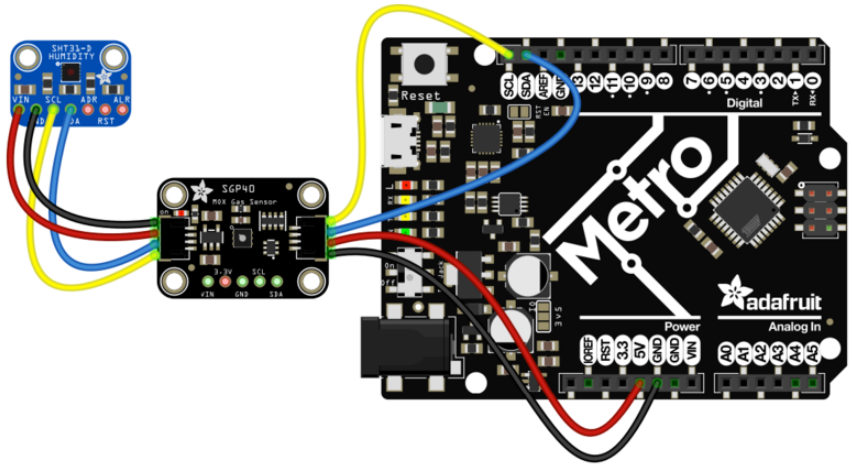
```
SGP40 test
Found SGP40 serial #023B3B4D
Measurement: 27351
Measurement: 27390
Measurement: 27439
```

These measurements rare the raw values from the VOC-sensitive resistor. They aren't quite 'resistance' but they're related. The number is affected by VOC as well as humidity.

VOC Index Example Code

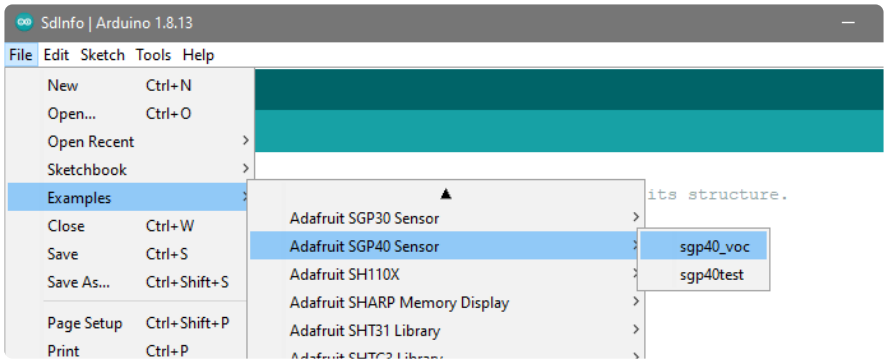
Next we'll use the SGP40 along with a SHT31 humidity sensor to calculate the VOC Index, with the SHT31 providing humidity measurements to allow the SGP40 to correct for it

For this example, you'll need to add a SHT31 () humidity sensor to the I2C bus, connected along with the SGP40 to the Metro/Arduino.



Simply add connections for VIN, GND, SCL, and SDA between the SGP40 and SHT31 as seen above. Alternately the SHT31 can be wired directly to the Metro's I2C connections.

With the wiring done, open up File -> Examples -> Adafruit SGP40 -> sgp40_voc



After opening the demo file, upload to your Arduino wired up to the sensor. Once you upload the code, you will see the raw measurement and VOC Index values being printed when you open the Serial Monitor (Tools->Serial Monitor) at 115200 baud, similar to this:

```
SGP40 test with SHT31 compensation
Found SHT3x + SGP40 serial #023B3B4D
Temp *C = 23.11         Hum. % = 39.66
Raw measurement: 27167
Voc Index: 0
Temp *C = 23.09         Hum. % = 39.79
Raw measurement: 27222
Voc Index: 0
Temp *C = 23.08         Hum. % = 39.79
Raw measurement: 27328
Voc Index: 0
Temp *C = 23.08         Hum. % = 40.04
Raw measurement: 27415
Voc Index: 0
```

It may take several minutes for the Voc index to start changing as it calibrates the baseline readings. We use the Sensirion SGP40 algorithm found here ()

Arduino Docs

Arduino Docs ()

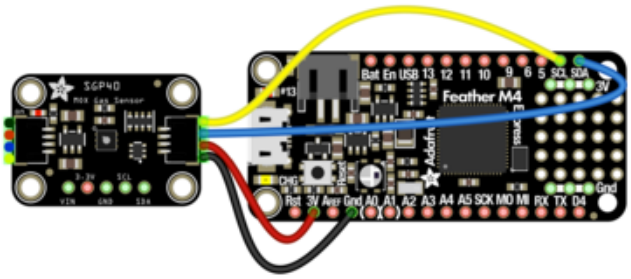
Python & CircuitPython

It's easy to use the SGP40 with Python or CircuitPython, and the Adafruit CircuitPython SGP40 () module. This module allows you to easily write Python code that reads gas measurements from the SGP40 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 ().

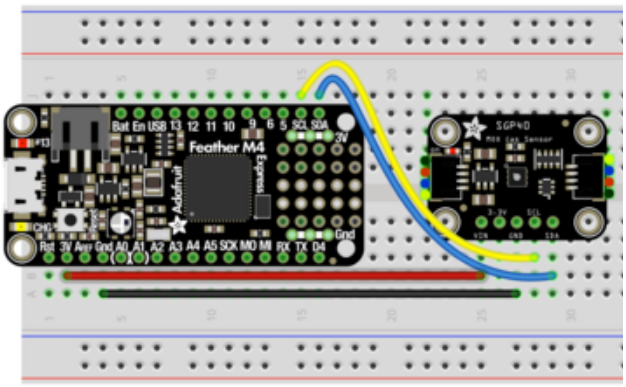
CircuitPython Microcontroller Wiring

First wire up a SGP40 to your board exactly as shown below. Here's an example of wiring a Feather M4 to the sensor with I2C using one of the handy STEMMA QT () connectors:



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

You can also use the standard 0.100" pitch headers to wire it up on a breadboard:

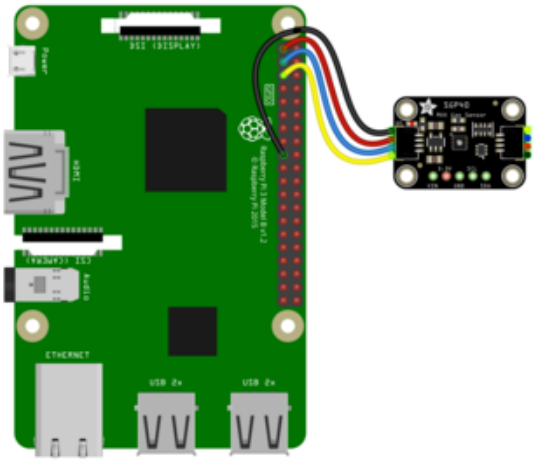


Board 3V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

Python Computer Wiring

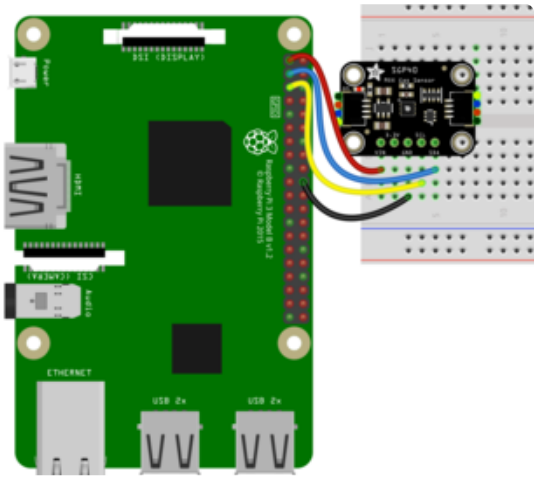
Since there's dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported ().

Here's the Raspberry Pi wired to the sensor using I2C and a STEMMA QT () connector:



Pi 3V to sensor VCC (red wire)
Pi GND to sensor GND (black wire)
Pi SCL to sensor SCL (yellow wire)
Pi SDA to sensor SDA (blue wire)

Finally here is an example of how to wire up a Raspberry Pi to the sensor using a solderless breadboard



Pi 3V to sensor VCC (red wire)
Pi GND to sensor GND (black wire)
Pi SCL to sensor SCL (yellow wire)
Pi SDA to sensor SDA (blue wire)

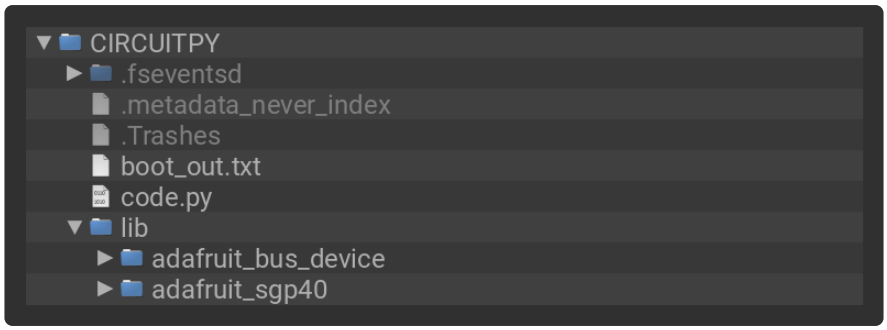
CircuitPython Installation of SGP40 Library

You'll need to install the Adafruit CircuitPython SGP40 () 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--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (). Our CircuitPython starter guide has a great page on how to install the library bundle ().

Before continuing make sure your board's lib folder or root filesystem has the adafruit_SGP40 and adafruit_bus_device folders copied over. Your CIRCUITPY drive should look like this:



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

Python Installation of SGP40 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-sgp40`

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

To demonstrate the usage of the sensor we'll initialize it and read the raw gas measurements from the board's Python REPL.

Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
import board
import busio
import adafruit_sgp40

i2c = busio.I2C(board.SCL, board.SDA)
sgp = adafruit_sgp40.SGP40(i2c)
```

Now you're ready to read values from the sensor using the raw property to read the raw gas measurements.

```
print("Raw Gas: ", sgp.raw)
```

```
>>> print("Raw Gas: ", sgp.raw)
Raw Gas:  28426
```

These measurements rare the raw values from the VOC-sensitive resistor. They aren't quite 'resistance' but they're related. The number is affected by VOC as well as humidity.

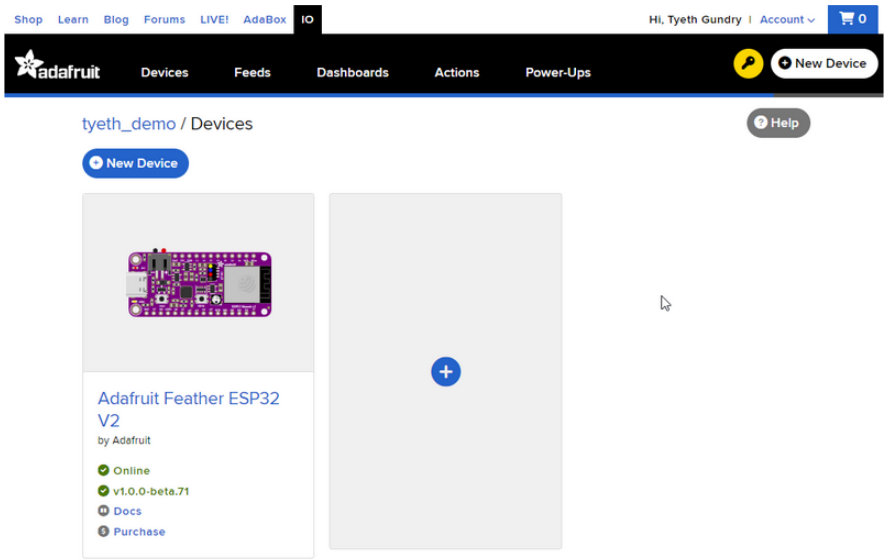
Example Code

```
# SPDX-FileCopyrightText: 2020 by Bryan Siepert for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense
import time
import board
import adafruit_sgp40

# If you have a temperature sensor, like the bme280, import that here as well
# import adafruit_bme280

i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller
sgp = adafruit_sgp40.SGP40(i2c)
# And if you have a temp/humidity sensor, define the sensor here as well
# bme280 = adafruit_bme280.Adafruit_BME280_I2C(i2c)

while True:
    print("Raw Gas: ", sgp.raw)
    # Lets quickly grab the humidity and temperature
    # temperature = bme280.temperature
    # humidity = bme280.relative_humidity
    # compensated_raw_gas = sgp.measure_raw(temperature = temperature,
relative_humidity = humidity)
    print("")
    time.sleep(1)
```

Python Docs

Python Docs ()

WipperSnapper



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.

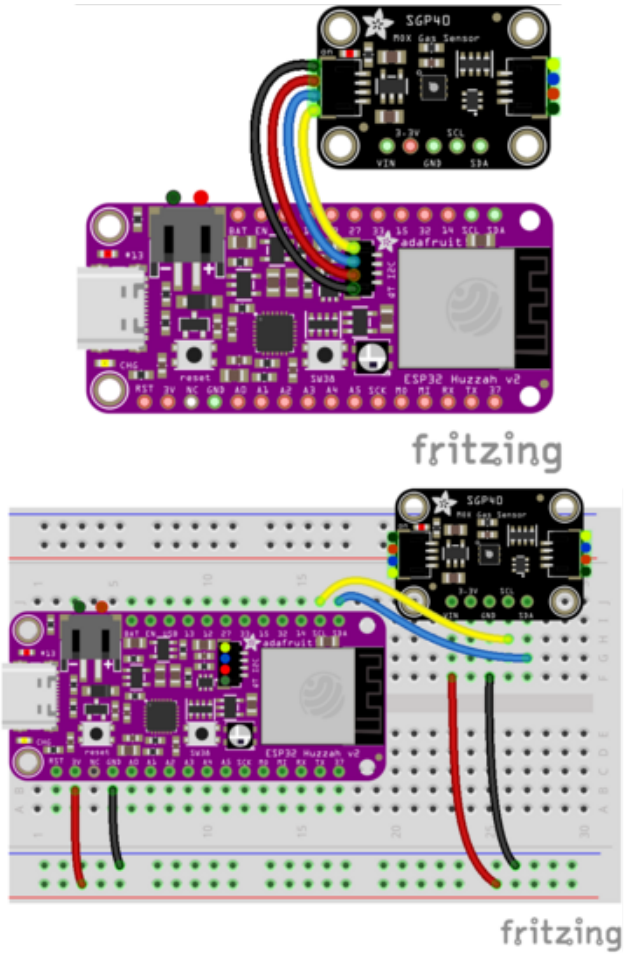
From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware, so you can immediately start interacting, logging, and streaming the data your projects produce without writing code.

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 an SGP40 to your board exactly as follows. Here is an example of the SGP40 wired to an Adafruit ESP32 Feather V2 () using I2C with a STEMMA QT cable (no soldering required) ()

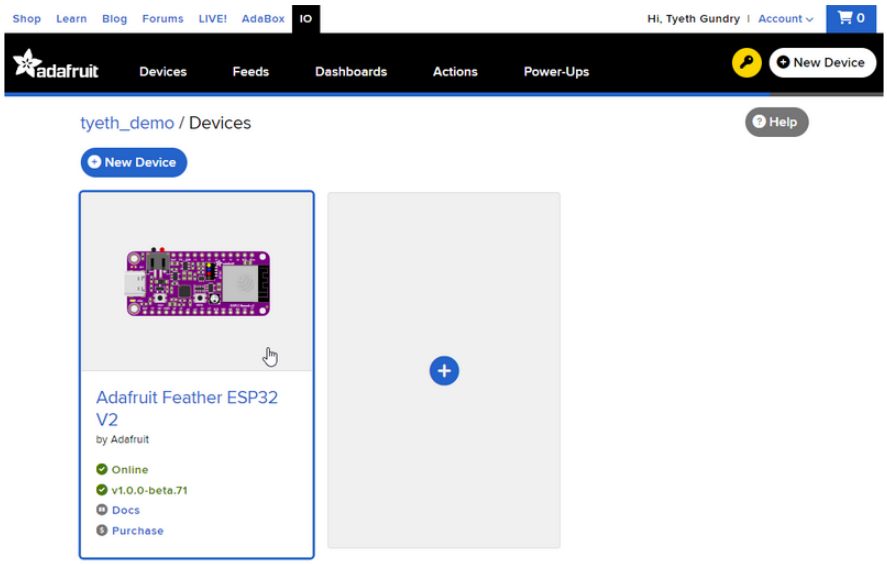


Board 3V to sensor VIN (red wire on STEMMA QT)
Board GND to sensor GND (black wire on STEMMA QT)
Board SCL to sensor SCL (yellow wire on STEMMA QT)
Board SDA to sensor SDA (blue wire on STEMMA QT)

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.



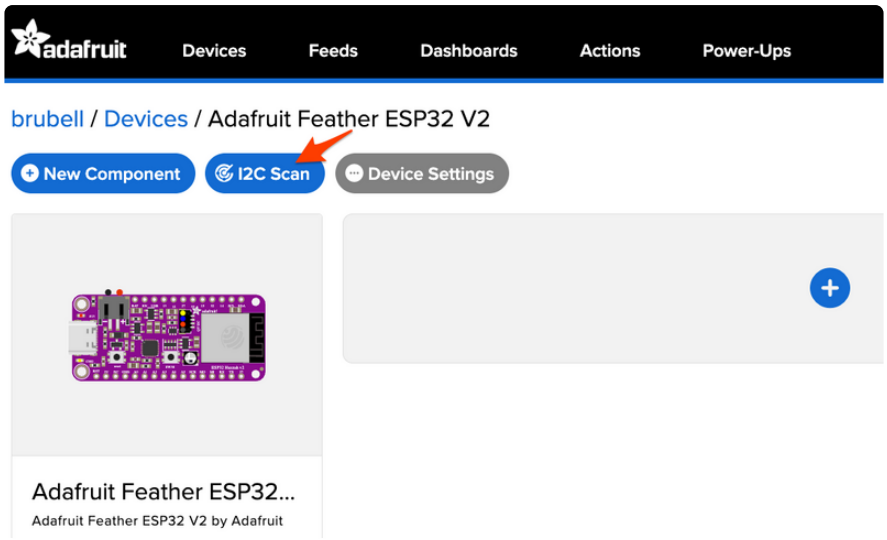
If you do not see your board listed here - you need to connect your board to Adafruit IO () first.

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

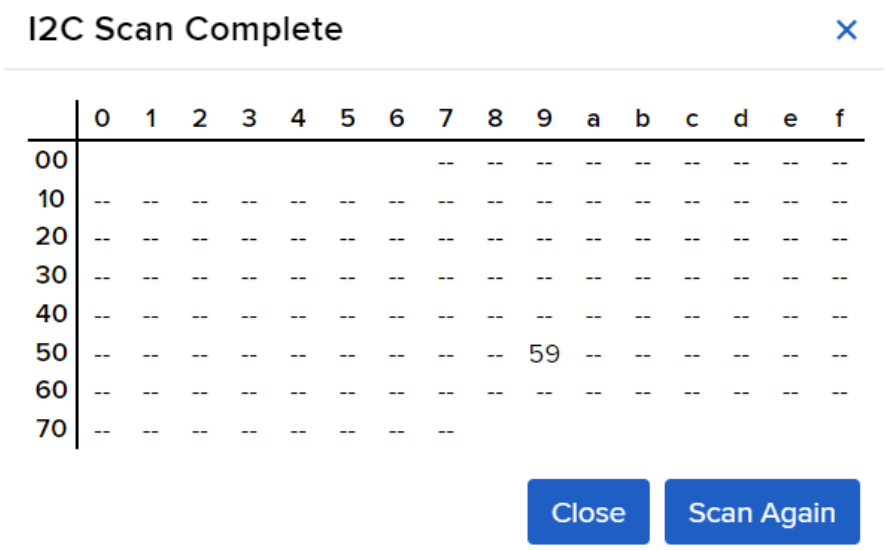
The device tile on the left indicates the version number of the firmware running on the connected board.

If the firmware version is green with a checkmark - continue with this guide.
If the firmware version is red with an exclamation mark "!" - 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 SGP40's default I2C address of `0x59` pop-up in the I2C scan list.



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

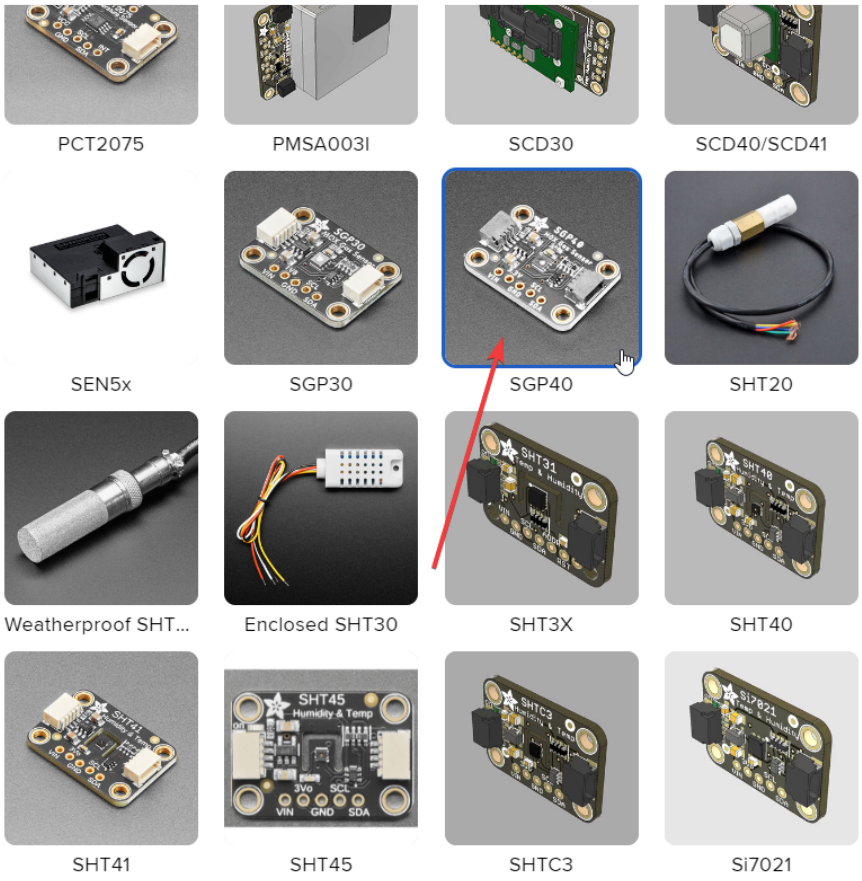
First, double-check the connection and/or wiring between the sensor and the board.

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.

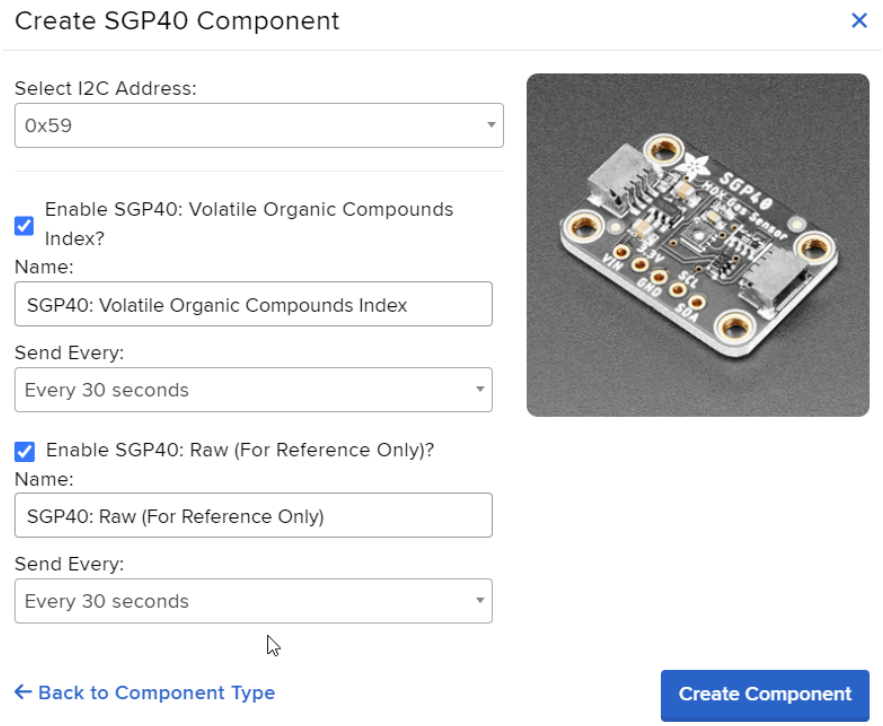
Select the SGP40 from the component picker.



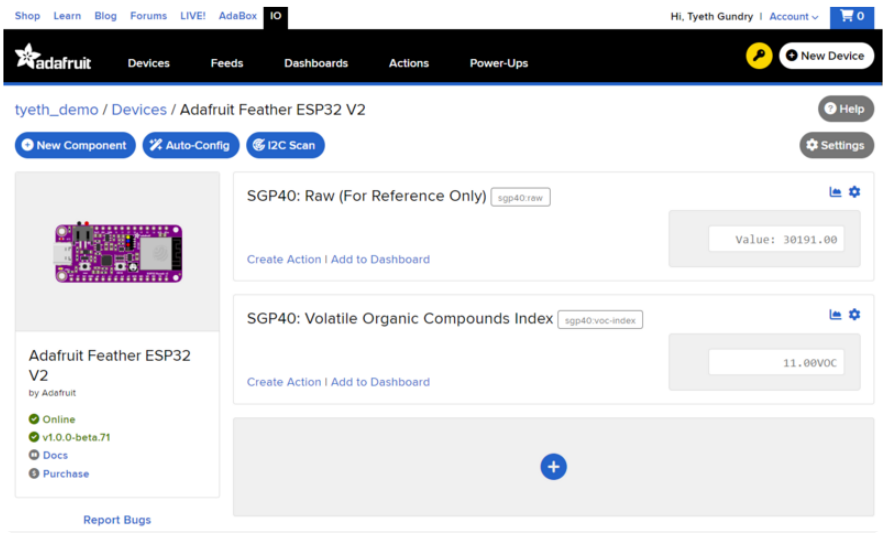
On the component configuration page, the SGP40's sensor address should be listed along with the sensor's settings.

The Send Every option is specific to each sensor's measurements. This option will tell the Feather how often it should read from the SGP40 sensor and send the data to Adafruit IO. Measurements can range from every 30 seconds to every 24 hours.

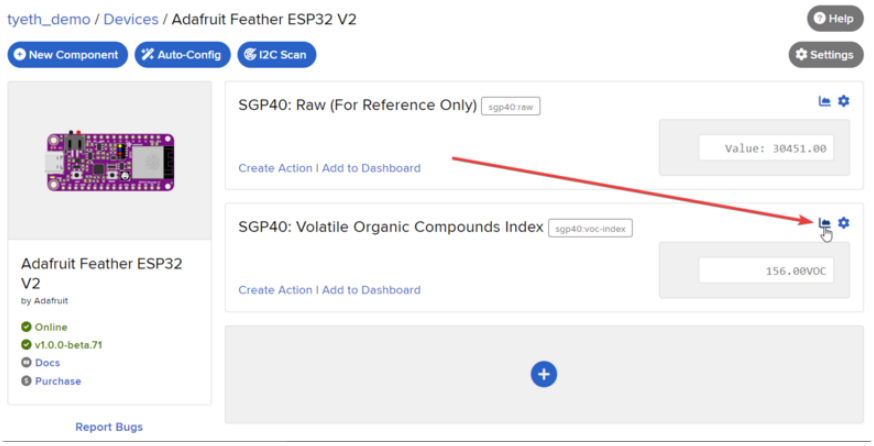
For this example, set the Send Every interval to every 30 seconds.



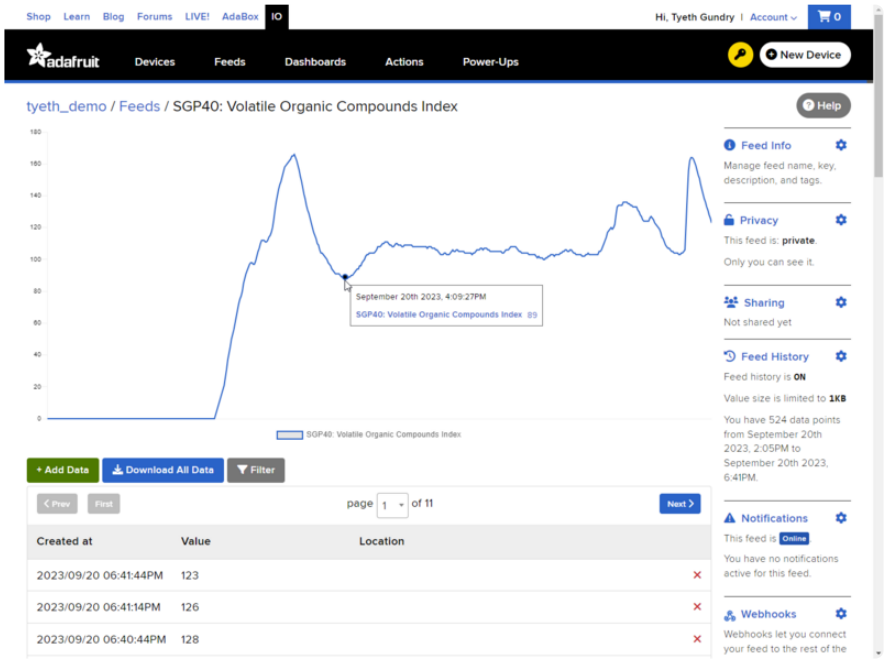
Your device interface should now show the sensor components you created. After the interval you configured elapses, WipperSnapper will automatically read values from the sensor(s) and send them to Adafruit IO.



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 ().



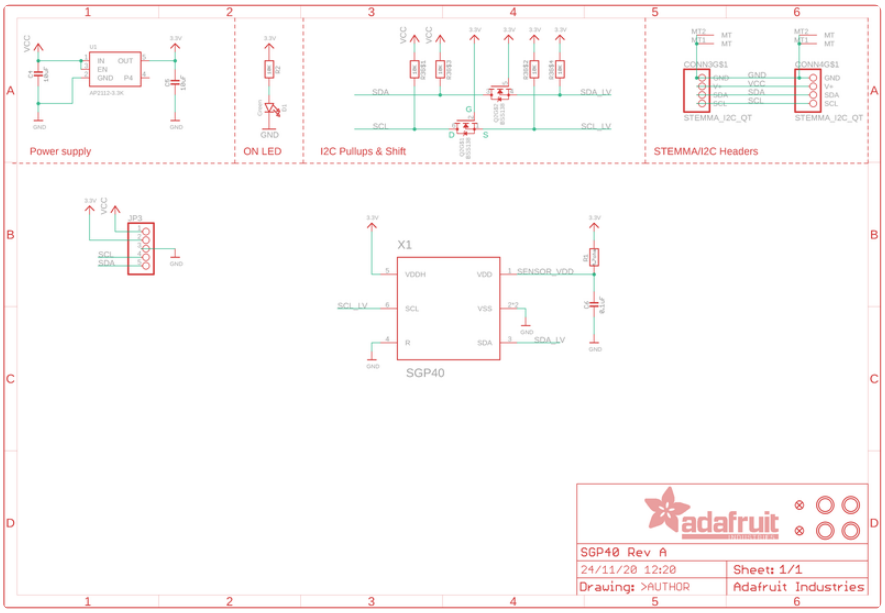
Downloads

Files:

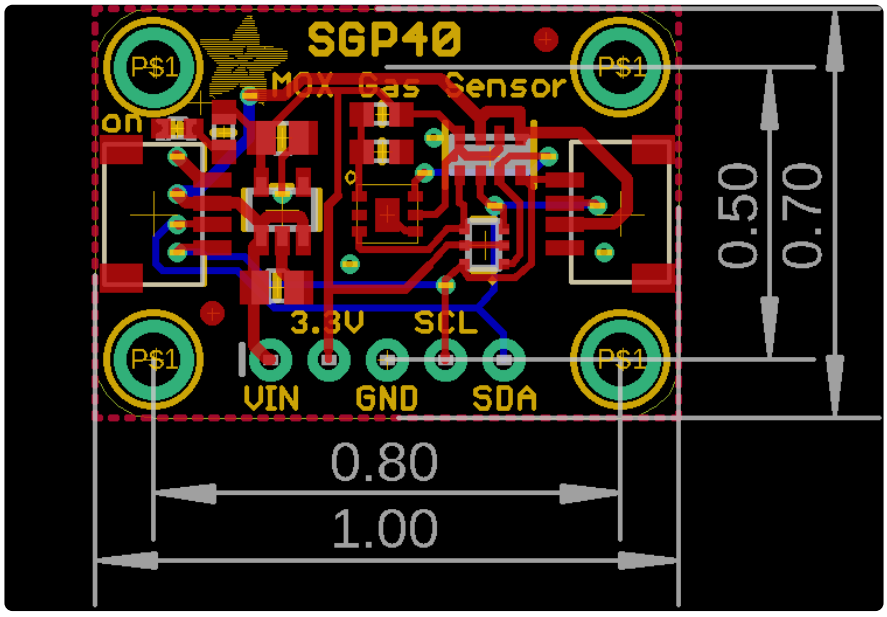
- SGP40 Datasheet ()
- Fritzing object in the Adafruit Fritzing Library ()
- EagleCAD PCB files on GitHub ()
- 3D models on GitHub ()

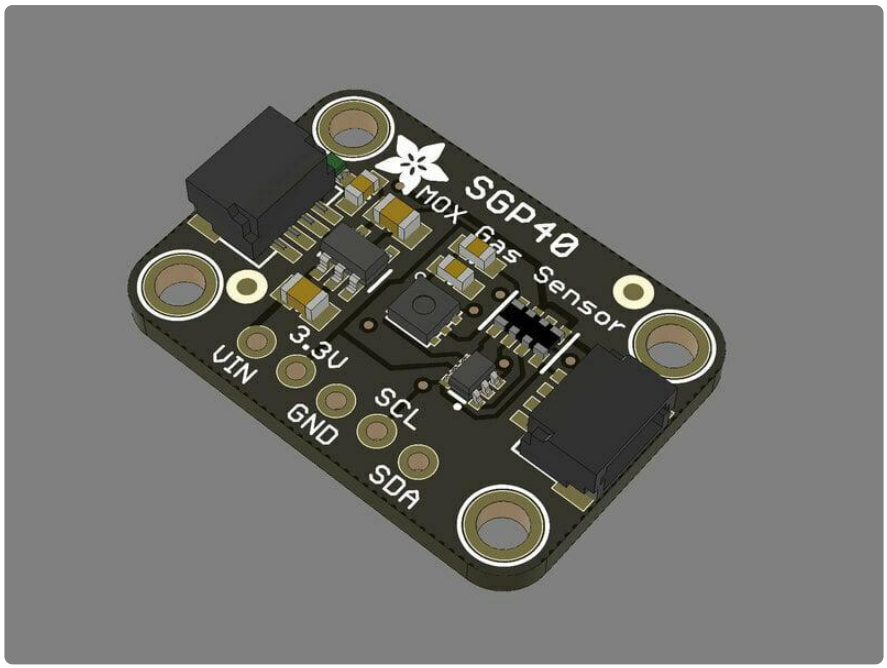
VOC Index for Experts App note

Schematic



Fab Print




SGP40
MOX Gas Sensor
3.3V
VIN
GND
SCL
SDA