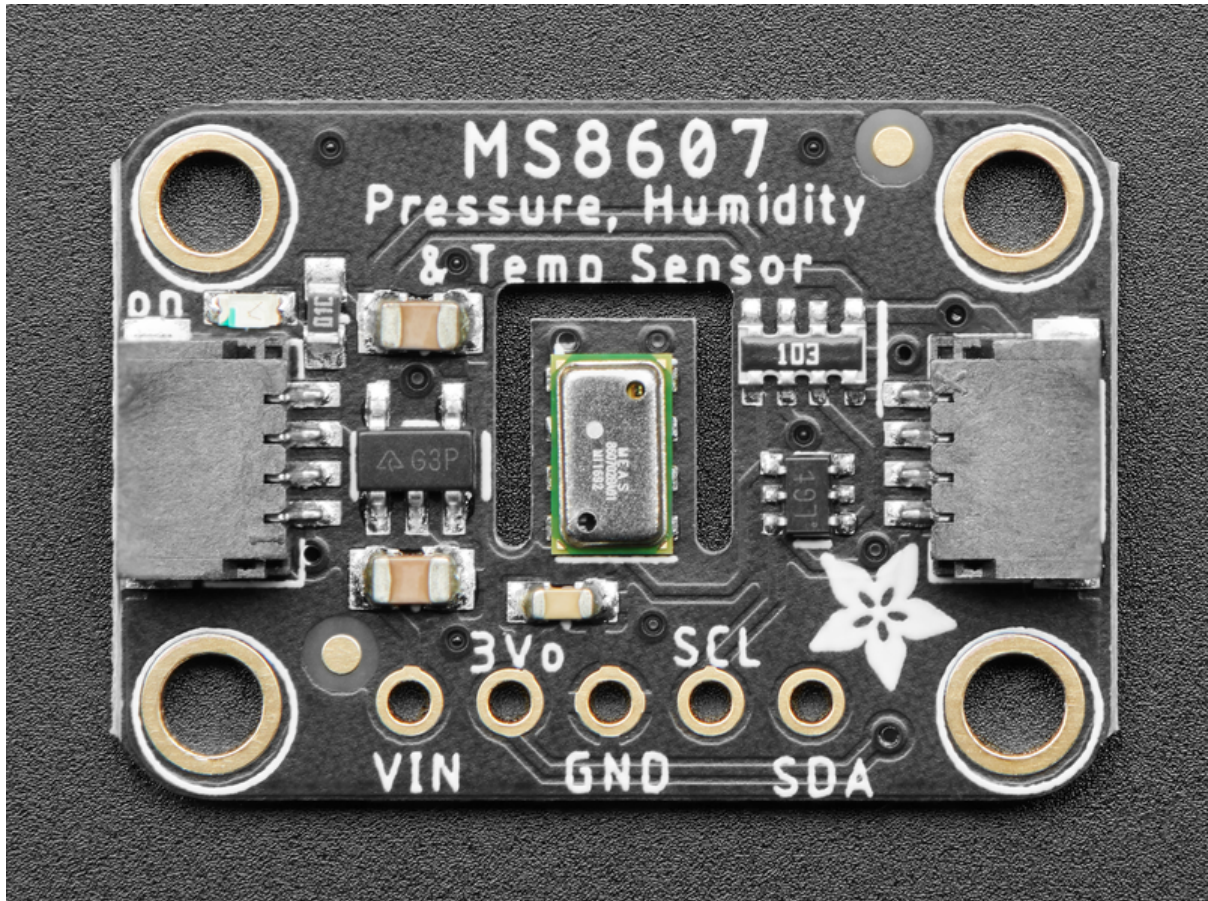




Adafruit TE MS8607 PHT Sensor

Created by Bryan Siepert



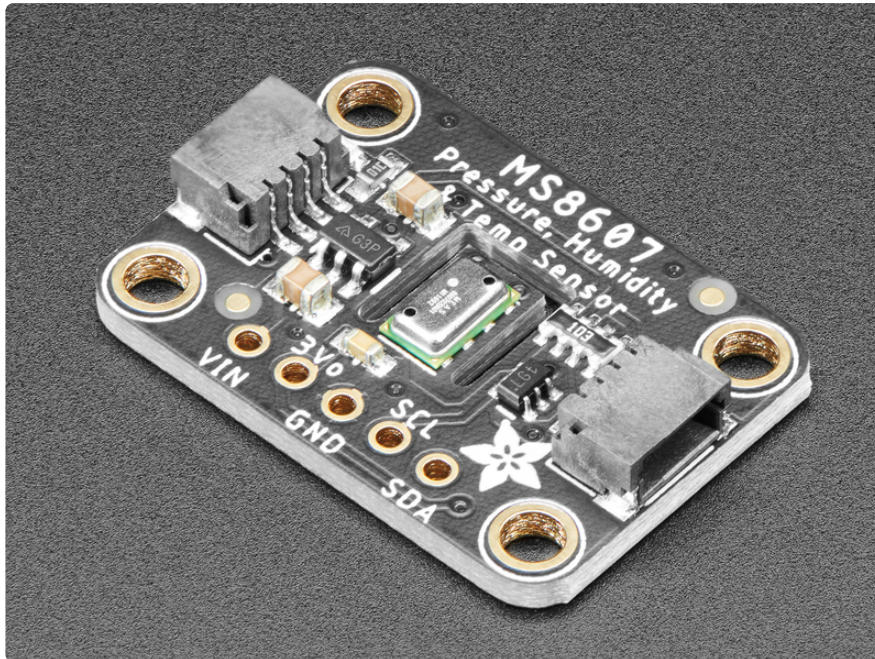
<https://learn.adafruit.com/adafruit-te-ms8607-pht-sensor>

Last updated on 2024-03-08 03:48:44 PM EST

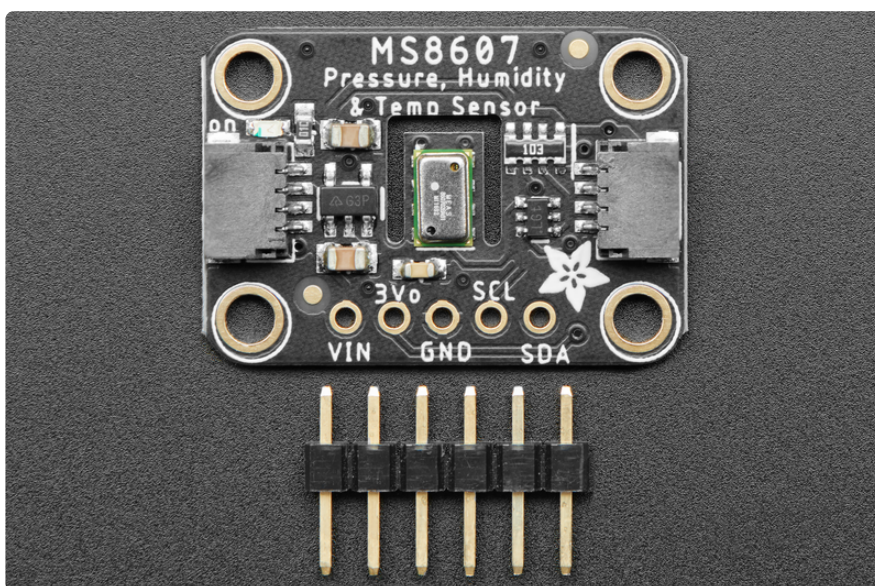
Table of Contents

Overview	3
Pinouts	5
• I2C Logic Pins	
Arduino	6
• I2C Wiring	
• Library Installation	
• Load Example	
• Example Code	
Arduino Docs	9
Python & CircuitPython	9
• CircuitPython Microcontroller Wiring	
• Python Computer Wiring	
• CircuitPython Installation of MS8607 Library	
• Python Installation of MS8607 Library	
• CircuitPython & Python Usage	
• Example Code	
Python Docs	13
WipperSnapper	13
• What is WipperSnapper	
• Wiring	
• Usage	
Downloads	21
• Files	
• Schematic	
• Fab Print	

Overview

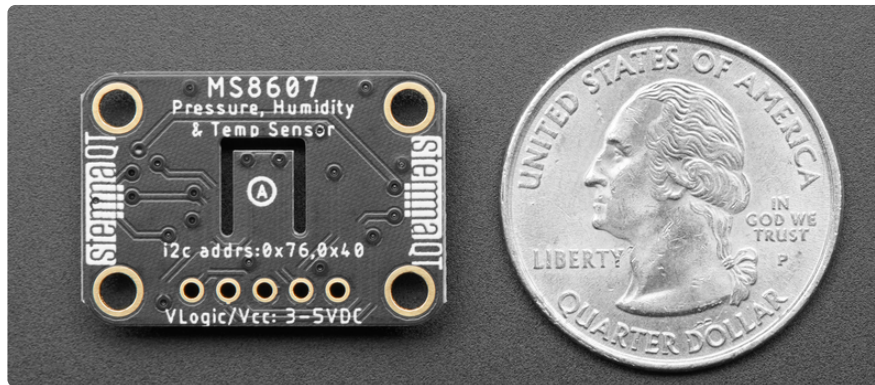


Because of their many applications, Pressure, Temperature, and Humidity sensors (PHT sensors) are a common offering from semiconductor companies. The MS8607 PHT sensor from TE Connectivity does an admirable job of measuring a wide range of all three environmental conditions. Each and every MS8607 sensor is calibrated at the factory and the calibration constants stored in the sensor itself. This sensor is a one stop shop for all your common weather and environmental related metrics. Just add a MS8607 and you'll be ready to finish your enclosure and retire as a [Dubia Cockroach Farmer](https://adafru.it/MNB) (<https://adafru.it/MNB>)

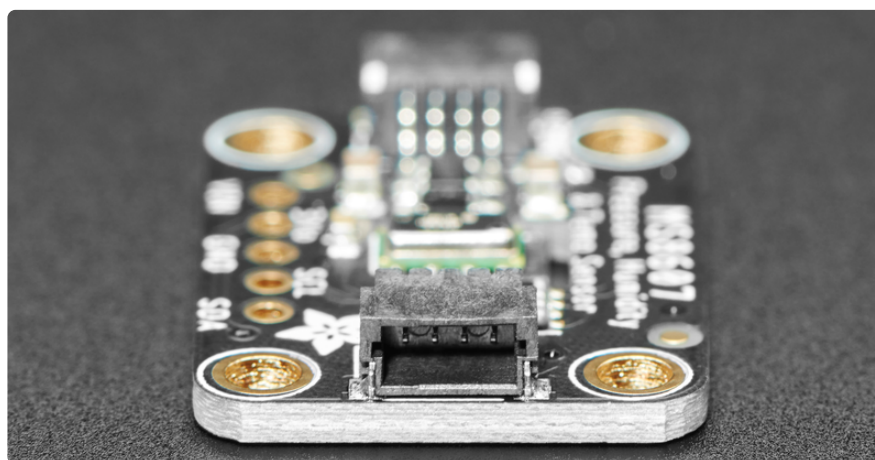


The MS8607 provides the previously mentioned factory-calibrated pressure, temperature, and humidity data are recorded as **16-bit** values and made available

over an I2C bus for simple, two wire (plus power and ground!) connection to your microcontroller. Pressure measurements accuracy comes in at +/- **2 hPa**, relative humidity at +/- **3% rH**, and temperature is good to within **1 degree Celsius**. One standout feature of the MS8607 is it's very respectable low power consumption at **as low as 0.78 μ A** ! That's hardly anything! Along with carefully managed microcontroller power usage, getting good battery life out of your sensing project will be that much easier.

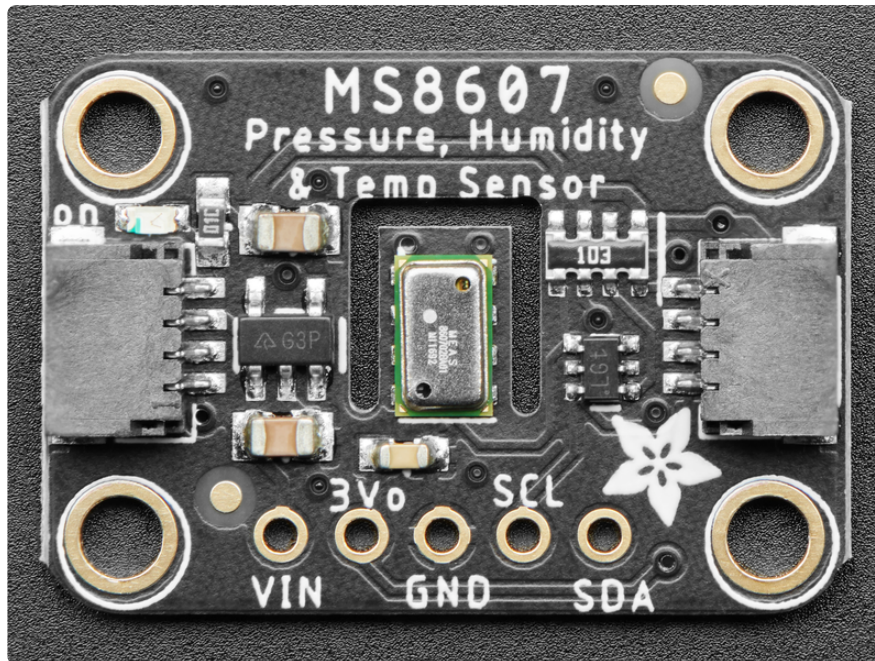


While the MS8607 sensor itself is rotund by comparison to some of the other sensors we stock, it's still a surface mount part so we've placed it on a breakout board, nestled amongst a handful components that take care of level shifting and power regulation. This lets you use it with either 5V or 3.3V devices, so no matter if you have an Arduino, Metro, Feather, or Raspberry Pi, you'll be able to interface with the MS8607 easily. The breakout has the expected header for power and I2C connections, however if you prefer to or can't solder, you can use the [SparkFun Qwiic \(https://adafruit.com/products/2666\)](https://adafruit.com/products/2666) compatible [STEMMA QT \(https://adafruit.com/products/2666\)](https://adafruit.com/products/2666) connectors to wire up to your favorite micro with a [plug-and-play cable \(https://adafruit.com/products/2666\)](https://adafruit.com/products/2666). Even if like me you don't mind soldering, STEMMA QT connectors are super convenient for wiring up projects quickly.



Lastly but certainly not least, our wiring guides, libraries, and example code for Arduino, Python, and CircuitPython will make it super easy to go from zero to hero!

Pinouts



Power Pins

- **VIN** - this is the power pin. Since the sensor chip uses 3 VDC, 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 microcontroller like Arduino, use 5V
- **3Vo** - 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

I2C Logic Pins

- **SCL** - I2C clock pin, connect to your microcontroller I2C clock line. This pin is level shifted so you can use 3-5V logic, and there's a **10K pullup** on this pin.
- **SDA** - I2C data pin, connect to your microcontroller I2C data line. This pin is level shifted so you can use 3-5V logic, and there's a **10K pullup** on this pin.
- **STEMMA QT** (<https://adafru.it/Ft4>) - These connectors allow you to connect to dev boards with **STEMMA QT** connectors or to other things with [various associated accessories](https://adafru.it/Ft6) (<https://adafru.it/Ft6>)

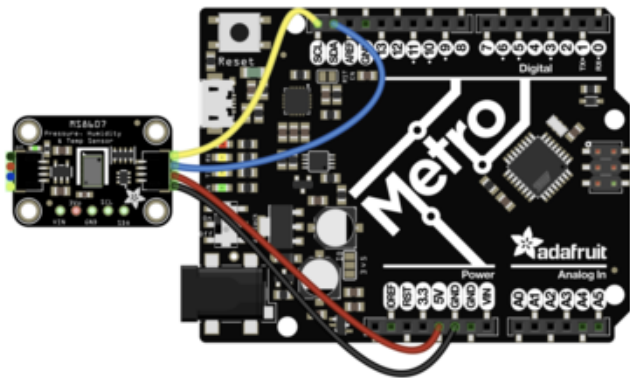
Arduino

Using the MS8607 with Arduino is a simple matter of wiring up the sensor to your Arduino-compatible microcontroller, installing the [Adafruit MS8607 \(https://adafruit.it/MNC\)](https://adafruit.it/MNC) library we've written, and running the provided example code.

I2C Wiring

Use this wiring if you want to connect via I2C interface. The MS8607 is actually two devices in the same package so the I2C address for the pressure sensor is **0x76** and the address for the humidity sensor is **0x40**.

Here is how to wire up the sensor using one of the [STEMMA QT \(https://adafruit.it/Ft4\)](https://adafruit.it/Ft4) connectors. The examples show a Metro but wiring will work the same for an Arduino or other compatible board.



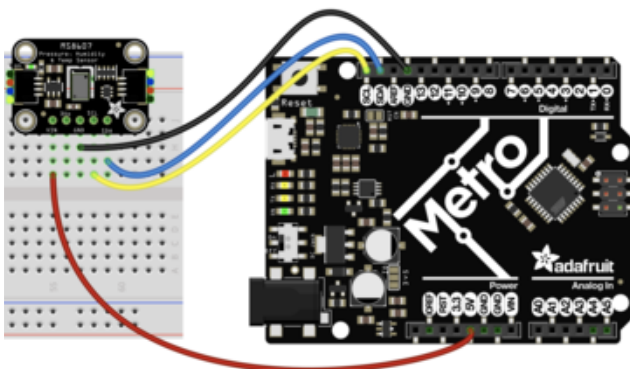
Connect **board VIN (red wire)** to **Arduino 5V** if you are running a **5V** board Arduino (Uno, etc.). If your board is **3V**, connect to that instead.

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** board Arduino (Uno, etc.). If your board is **3V**, connect to that instead.

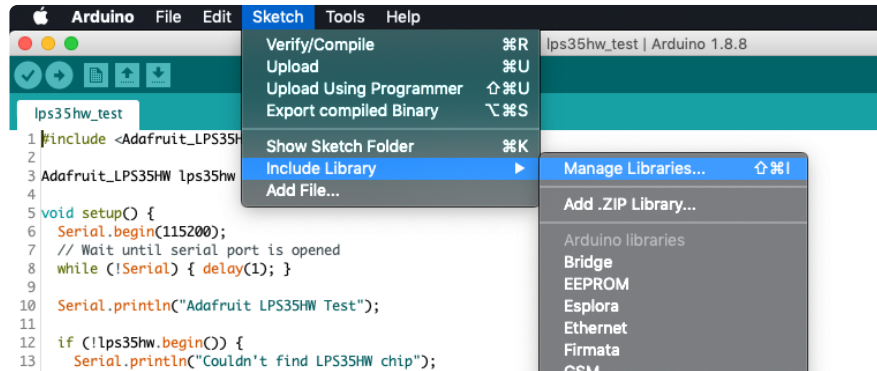
Connect **board GND (black wire)** to **Arduino GND**

Connect **board SCL (yellow wire)** to **Arduino SCL**

Connect **board SDA (blue wire)** to **Arduino SDA**

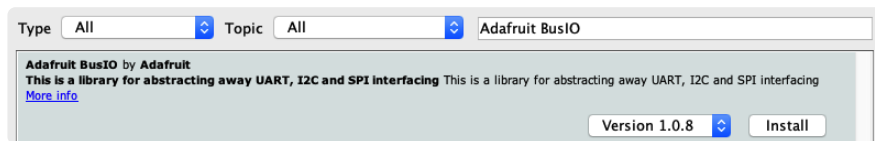
Library Installation

You can install the [Adafruit MS8607](https://adafru.it/MNC) (<https://adafru.it/MNC>) (<https://adafru.it/MND>) library for Arduino using the Library Manager in the Arduino IDE.

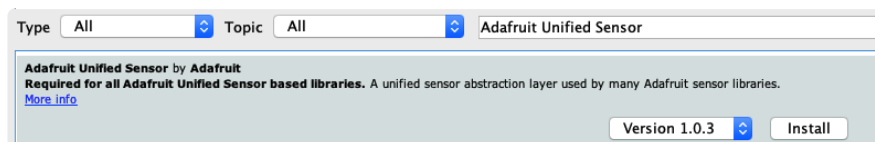


Click the **Manage Libraries ...** menu item, search for **Adafruit MS8607** , and select the **Adafruit MS8607** library:

Follow the same process for the **Adafruit BusIO** library.



Finally follow the same process for the **Adafruit Unified Sensor** library:



Load Example

Open up **File -> Examples -> Adafruit MS8607 -> sensor_test**

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

Example Code

```
// Basic demo for reading Humidity and Temperature
#include <Wire.h>
```

```

#include <Adafruit_MS8607.h>
#include <Adafruit_Sensor.h>

Adafruit_MS8607 ms8607;
void setup(void) {
  Serial.begin(115200);
  while (!Serial) delay(10);    // will pause Zero, Leonardo, etc until serial
                                // console opens

  Serial.println("Adafruit MS8607 test!");

  // Try to initialize!
  if (!ms8607.begin()) {
    Serial.println("Failed to find MS8607 chip");
    while (1) { delay(10); }
  }
  Serial.println("MS8607 Found!");

  ms8607.setHumidityResolution(MS8607_HUMIDITY_RESOLUTION_OSR_8b);
  Serial.print("Humidity resolution set to ");
  switch (ms8607.getHumidityResolution()){
    case MS8607_HUMIDITY_RESOLUTION_OSR_12b: Serial.println("12-bit"); break;
    case MS8607_HUMIDITY_RESOLUTION_OSR_11b: Serial.println("11-bit"); break;
    case MS8607_HUMIDITY_RESOLUTION_OSR_10b: Serial.println("10-bit"); break;
    case MS8607_HUMIDITY_RESOLUTION_OSR_8b: Serial.println("8-bit"); break;
  }
  // ms8607.setPressureResolution(MS8607_PRESSURE_RESOLUTION_OSR_4096);
  Serial.print("Pressure and Temperature resolution set to ");
  switch (ms8607.getPressureResolution()){
    case MS8607_PRESSURE_RESOLUTION_OSR_256: Serial.println("256"); break;
    case MS8607_PRESSURE_RESOLUTION_OSR_512: Serial.println("512"); break;
    case MS8607_PRESSURE_RESOLUTION_OSR_1024: Serial.println("1024"); break;
    case MS8607_PRESSURE_RESOLUTION_OSR_2048: Serial.println("2048"); break;
    case MS8607_PRESSURE_RESOLUTION_OSR_4096: Serial.println("4096"); break;
    case MS8607_PRESSURE_RESOLUTION_OSR_8192: Serial.println("8192"); break;
  }
  Serial.println("");
}

void loop() {
  sensors_event_t temp, pressure, humidity;
  ms8607.getEvent(&pressure, &temp, &humidity);
  Serial.print("Temperature: ");Serial.print(temp.temperature); Serial.println("
degrees C");
  Serial.print("Pressure: ");Serial.print(pressure.pressure); Serial.println("
hPa");
  Serial.print("Humidity: ");Serial.print(humidity.relative_humidity);
Serial.println(" %rH");
  Serial.println("");

  delay(500);
}

```

```

Adafruit MS8607 test!
MS8607 Found!
Humidity resolution set to 8-bit
Pressure and Temperature resolution set to 4096

```

```

Temperature: 28.79 degrees C
Pressure: 1008.00 hPa
Humidity: 39.36 %rH

```

```

Temperature: 28.79 degrees C
Pressure: 1008.01 hPa
Humidity: 39.38 %rH

```

```

Temperature: 28.79 degrees C
Pressure: 1008.01 hPa
Humidity: 39.42 %rH

```

Arduino Docs

[Arduino Docs \(https://adafru.it/MDj\)](https://adafru.it/MDj)

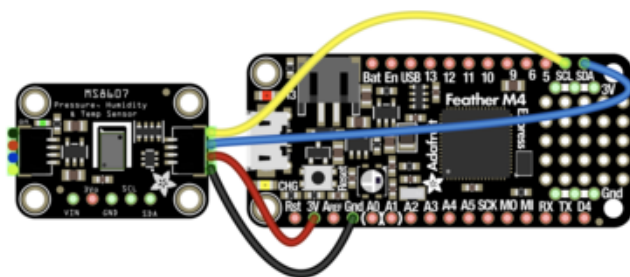
Python & CircuitPython

It's easy to use the **MS8607** with Python or CircuitPython, and the [Adafruit CircuitPython MS8607 \(https://adafru.it/MNE\)](https://adafru.it/MNE) module. This module allows you to easily write Python code that reads pressure, humidity and temperature from the **MS8607** 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 \(https://adafru.it/BSN\)](https://adafru.it/BSN).

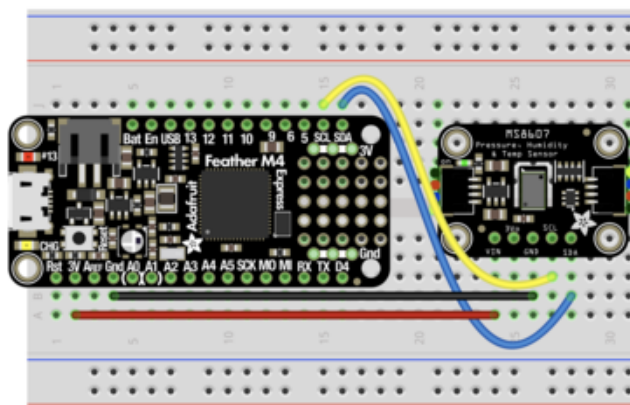
CircuitPython Microcontroller Wiring

First wire up a MS8607 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 \(https://adafru.it/Ft4\)](https://adafru.it/Ft4) 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:

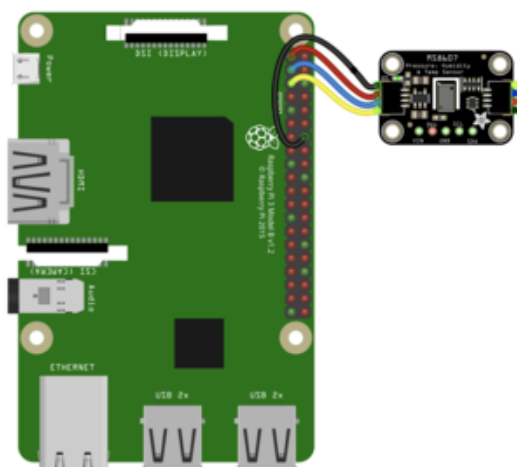


Board 3V to sensor VIN (red wire)
 Board GND to sensor GND (black wire)
 Board SCL to sensor SCL (yellow wire)
 Board SDA to sensor SDA (blue wire)

Python Computer Wiring

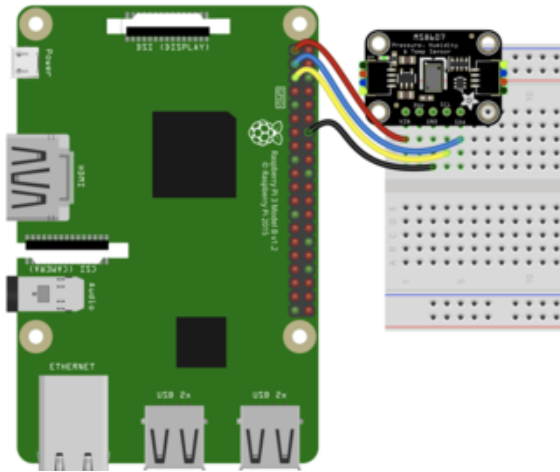
Since there's dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported](https://adafru.it/BSN) (<https://adafru.it/BSN>).

Here's the Raspberry Pi wired to the sensor using I2C and a [STEMMA QT](https://adafru.it/Ft4) (<https://adafru.it/Ft4>) 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 MS8607 Library

You'll need to install the [Adafruit CircuitPython MS8607 \(https://adafru.it/MNE\)](https://adafru.it/MNE) library on your CircuitPython board.

First make sure you are running the [latest version of Adafruit CircuitPython \(https://adafru.it/Amd\)](https://adafru.it/Amd) 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 \(https://adafru.it/ENC\)](https://adafru.it/ENC). Our CircuitPython starter guide has [a great page on how to install the library bundle \(https://adafru.it/ABU\)](https://adafru.it/ABU).

Before continuing make sure your board's `lib` folder or root filesystem has the `adafruit_ms8607.mpy` file and `adafruit_bus_device` folder copied over.

Next [connect to the board's serial REPL \(https://adafru.it/Awz\)](https://adafru.it/Awz) so you are at the CircuitPython `>>>` prompt.

Python Installation of MS8607 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 \(https://adafru.it/BSN\)](https://adafru.it/BSN)!

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

- `sudo pip3 install adafruit-circuitpython-ms8607`

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 pressure, temperature, and humidity 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
from adafruit_ms8607 import MS8607

i2c = busio.I2C(board.SCL, board.SDA)
sensor = MS8607(i2c)
```

```
>>> import board
>>> import busio
>>> from adafruit_ms8607 import MS8607
>>>
>>> i2c = busio.I2C(board.SCL, board.SDA)
>>> sensor = MS8607(i2c)
>>>
```

Now you're ready to read values from the sensor using these properties:

- **relative_humidity** - The relative humidity measured by the sensor, this is a value from 0-100%.
- **temperature** - The temperature measured by the sensor, a value in degrees Celsius.
- **pressure** - The pressure measured by the sensor, a value in hectopascals (hPa)

```
print("Pressure: %.2f hPa" % sensor.pressure)
print("Temperature: %.2f C" % sensor.temperature)
print("Humidity: %.2f %% rH" % sensor.relative_humidity)
```

```
>>> print("Pressure: %.2f hPa" % sensor.pressure)
Pressure: 1007.94 hPa
>>> print("Temperature: %.2f C" % sensor.temperature)
Temperature: 28.44 C
>>> print("Humidity: %.2f %% rH" % sensor.relative_humidity)
Humidity: 34.49 % rH
>>>
```


Example Code

```
# SPDX-FileCopyrightText: 2020 Bryan Siepert, written for Adafruit Industries
# SPDX-License-Identifier: MIT
from time import sleep
import board
from adafruit_ms8607 import MS8607

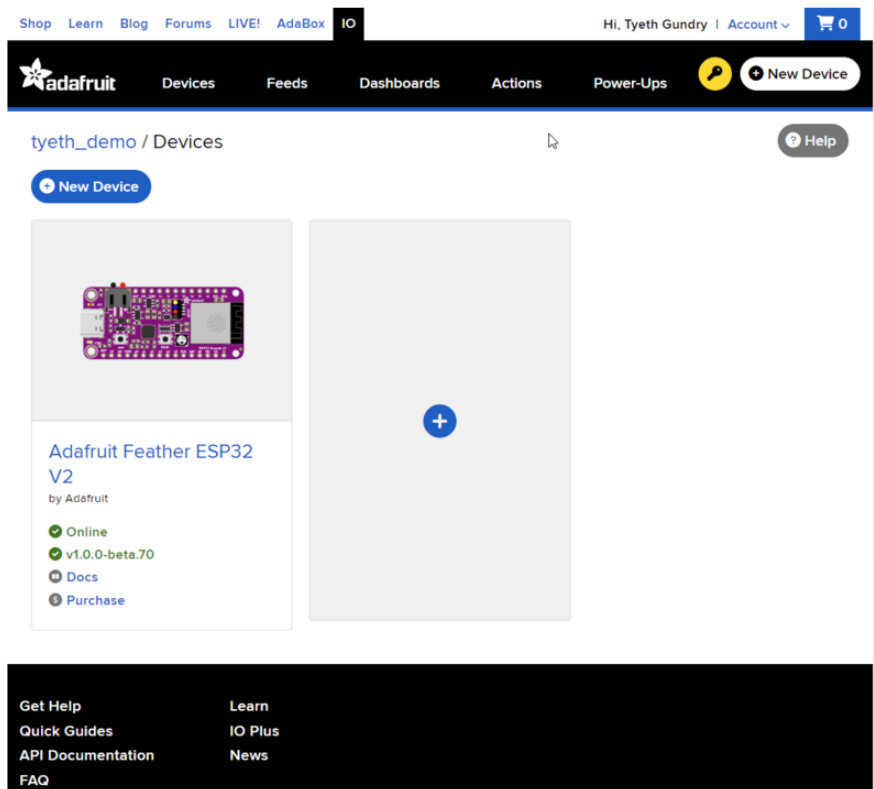
i2c = board.I2C() # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C() # For using the built-in STEMMMA QT connector on a
microcontroller
sensor = MS8607(i2c)

while True:
    print("Pressure: %.2f hPa" % sensor.pressure)
    print("Temperature: %.2f C" % sensor.temperature)
    print("Humidity: %.2f %% rH" % sensor.relative_humidity)
    print("\n-----\n")
    sleep(1)
```

Python Docs

[Python Docs \(https://adafru.it/MDk\)](https://adafru.it/MDk)

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 \(https://adafru.it/fsU\)](https://adafru.it/fsU), a web platform designed ([by Adafruit! \(https://adafru.it/Bo5\)](https://adafru.it/Bo5)) 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.

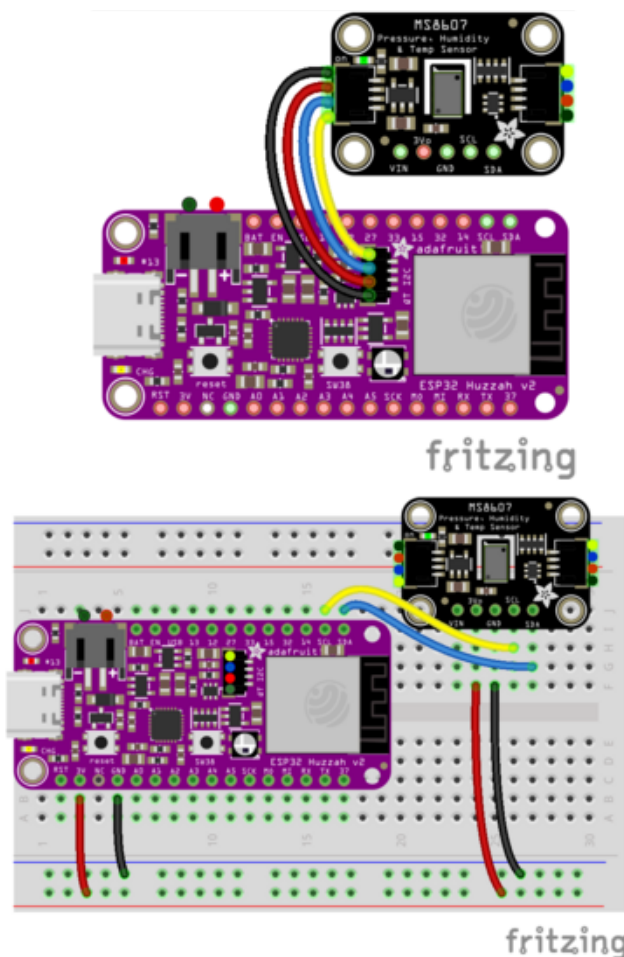
If you've never used WipperSnapper, click below to read through the quick start guide before continuing.

**Quickstart: Adafruit IO
WipperSnapper**

<https://adafru.it/Vfd>

Wiring

First, wire up an MS8607 to your board exactly as follows. Here is an example of the MS8607 wired to an [Adafruit ESP32 Feather V2 \(http://adafru.it/5400\)](http://adafru.it/5400) using I2C [with a STEMMA QT cable \(no soldering required\) \(http://adafru.it/4210\)](http://adafru.it/4210)



Board 3V to sensor VIN (red wire on STEMMA QT)

Board GND to sensor GND (black wire on STEMMA QT)

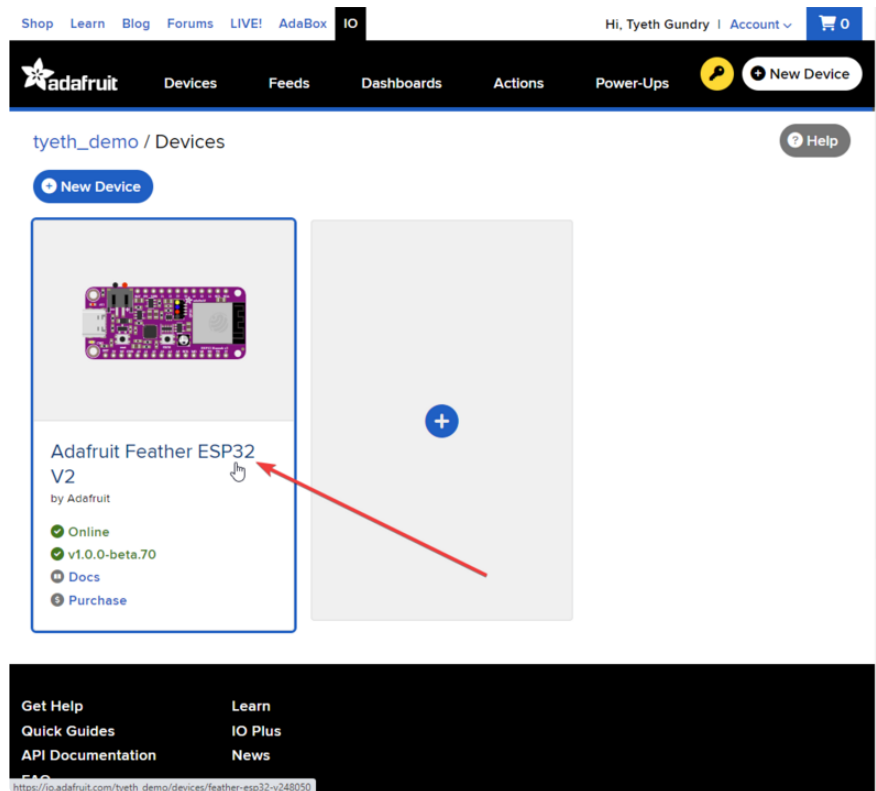
Board SCL to sensor SCL (yellow wire on STEMMA QT)

Board SDA to sensor SDA (blue wire on STEMMA QT)

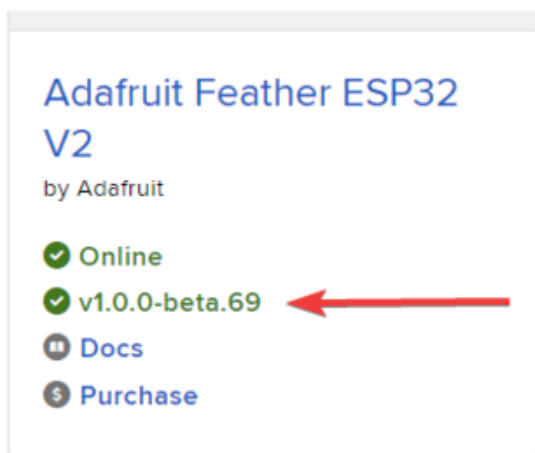
Usage

Connect your board to Adafruit IO Wippersnapper and [navigate to the WipperSnapper board list \(https://adafru.it/TAu\)](https://adafru.it/TAu).

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](https://adafru.it/Vfd) (<https://adafru.it/Vfd>) first.



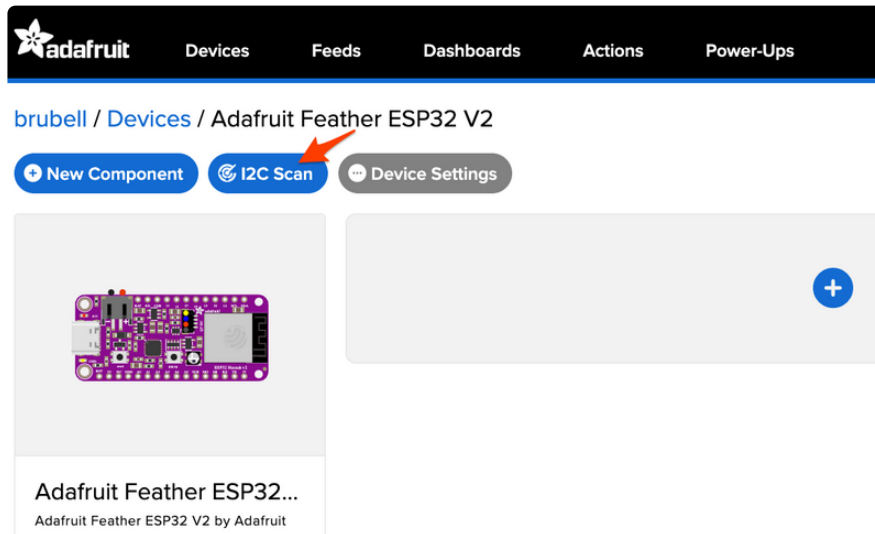
On the device page, quickly **check that you're running the latest version of the WipperSnapper firmware.**

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](https://adafru.it/Vfd) (<https://adafru.it/Vfd>) on your board before continuing.

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



The MS8607 is actually two devices in the same package so the I2C address for the pressure sensor is **0x76** and the address for the humidity sensor is **0x40**.

You should see the MS8607's I2C addresses of **0x40** and **0x76** pop-up in the I2C scan list.

I2C Scan Complete ✕

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00								--	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
40	40	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	76	--								

Close Scan Again

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

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

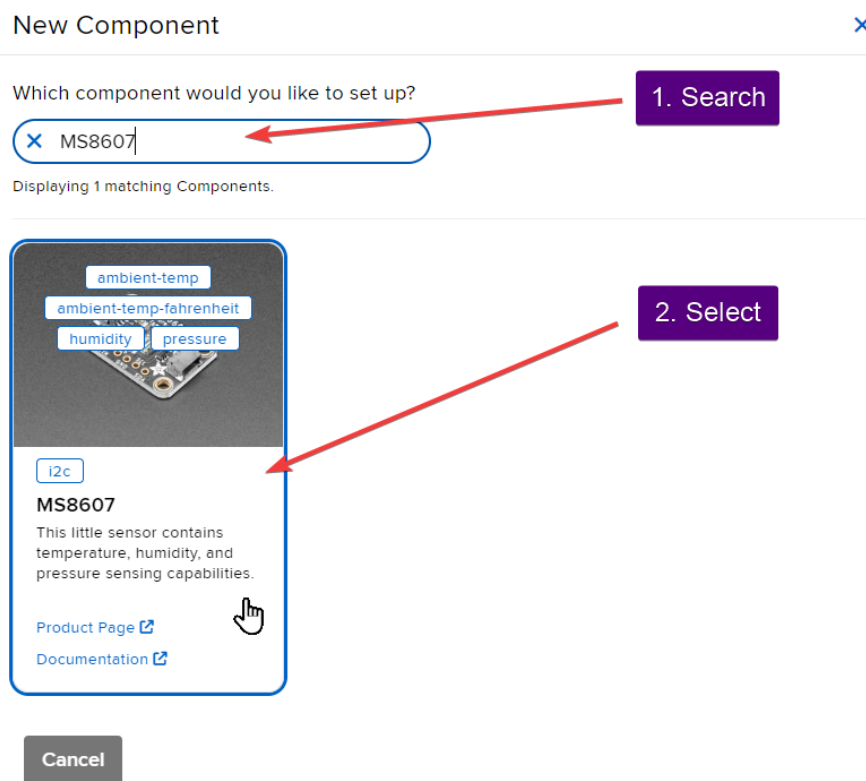
Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

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.



Adafruit IO supports a large amount of components. To quickly find your sensor, type **MS8607** into the search bar, then select the **MS8607** component.



On the component configuration page, as the MS8607 is actually two devices in the same package, the address for the humidity sensor should be listed along with the sensor's settings.

One I2C address of the MS8607 will be listed in WipperSnapper as not in use (incorrectly - this will be fixed in a future update).

Remember the MS8607 is actually two devices in the same package so the I2C address for the pressure sensor is 0x76 and the address for the humidity sensor is 0x40, and both addresses will be in use!

The **Send Every** option is specific to each sensor's measurements. This option will tell the Feather how often it should read from the MS8607 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.

Create MS8607 Component ✕

Select I2C Address:

0x40

☒ Enable MS8607: Temperature Sensor (°C)?

Name:
MS8607: Temperature Sensor (°C)

Send Every:
Every 30 seconds

☒ Enable MS8607: Temperature Sensor (°F)?

Name:
MS8607: Temperature Sensor (°F)

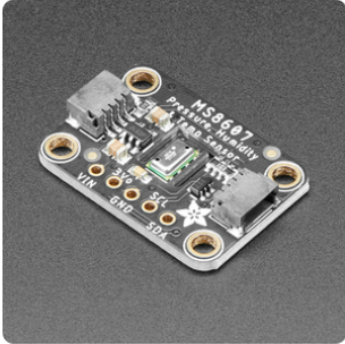
Send Every:
Every 30 seconds

☒ Enable MS8607: Humidity Sensor?

Name:
MS8607: Humidity Sensor

Send Every:
Every 30 seconds

☒ Enable MS8607: Pressure Sensor?



[← Back to Component Type](#)

Create Component

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.

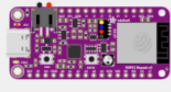
©Adafruit Industries

Page 19 of 22

tyeth_demo / Devices / Adafruit Feather ESP32 V2

Help Settings

New Component Auto-Config I2C Scan



Adafruit Feather ESP32 V2
by Adafruit

Online
v1.0.0-beta.70
Docs
Purchase

Report Bugs

MS8607: Humidity Sensor ms8607:humidity

Create Action | Add to Dashboard

58.72%

MS8607: Pressure Sensor ms8607:pressure

Create Action | Add to Dashboard

1006.35hPa

MS8607: Temperature Sensor (°C) ms8607:ambient-temp

Create Action | Add to Dashboard

20.90°C

MS8607: Temperature Sensor (°F) ms8607:ambient-temp-fahrenheit

Create Action | Add to Dashboard

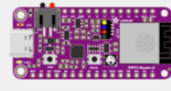
69.73°F

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

tyeth_demo / Devices / Adafruit Feather ESP32 V2

Help Settings

New Component Auto-Config I2C Scan



Adafruit Feather ESP32 V2
by Adafruit

Online
v1.0.0-beta.70
Docs
Purchase

Report Bugs

MS8607: Humidity Sensor ms8607:humidity

Create Action | Add to Dashboard

59.17%

MS8607: Pressure Sensor ms8607:pressure

Create Action | Add to Dashboard

1006.25hPa

MS8607: Temperature Sensor (°C) ms8607:ambient-temp

Create Action | Add to Dashboard

20.90°C

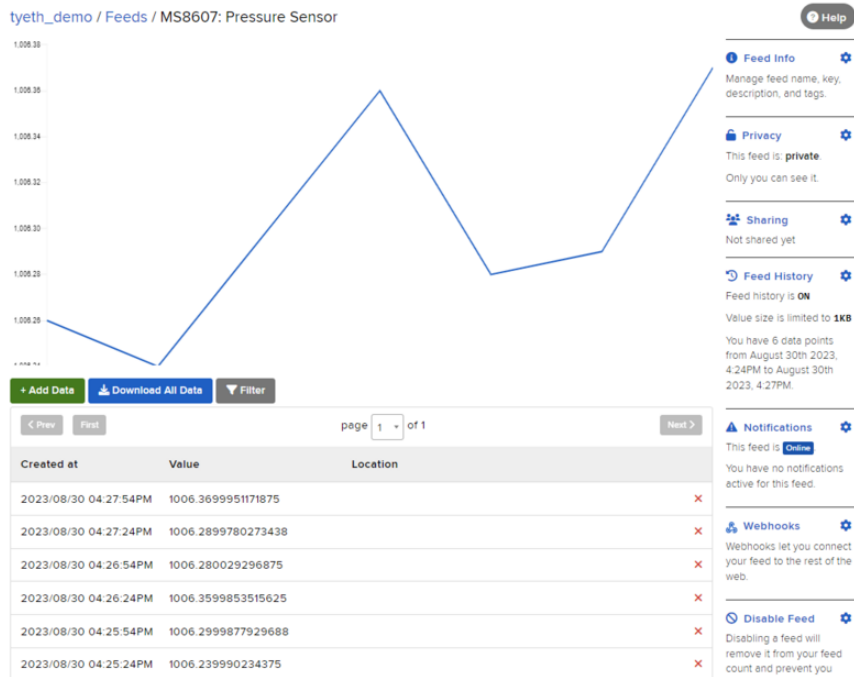
MS8607: Temperature Sensor (°F) ms8607:ambient-temp-fahrenheit

Create Action | Add to Dashboard

69.73°F

https://io.adafruit.com/tyeth_demo/feeds/feather-esp32-v248050.ws-0x40-pressure

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 \(https://adafru.it/10aZ\)](https://adafru.it/10aZ).

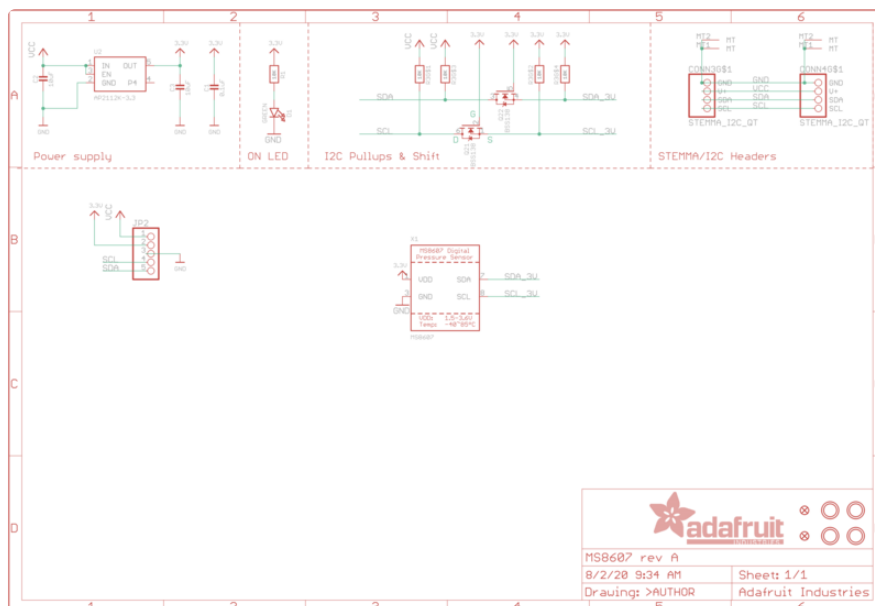


Downloads

Files

- [MS8607 Datasheet \(https://adafru.it/MNF\)](https://adafru.it/MNF)
- [EagleCAD files on GitHub \(https://adafru.it/MOa\)](https://adafru.it/MOa)
- [Fritzing object in the Adafruit Fritzing Library \(https://adafru.it/MOb\)](https://adafru.it/MOb)

Schematic



Fab Print

