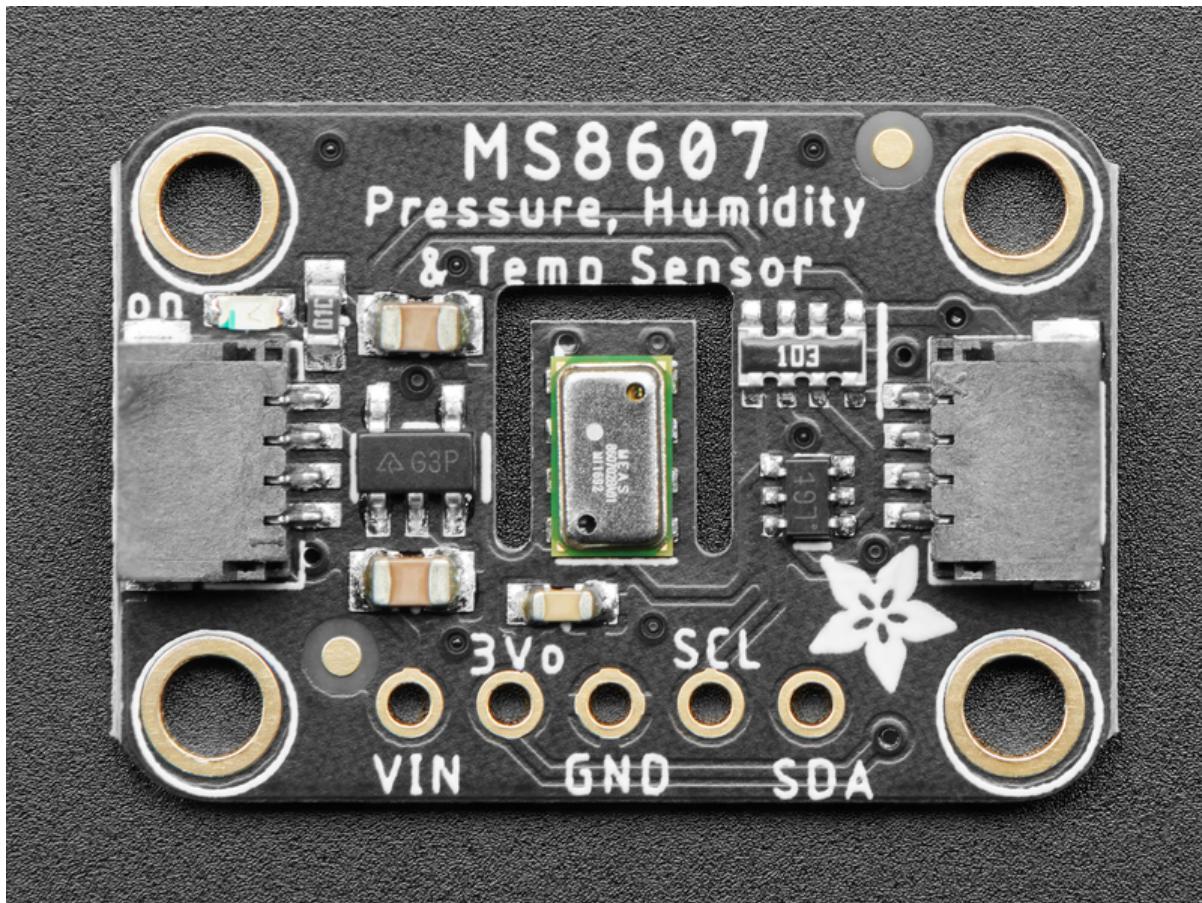


Adafruit TE MS8607 PHT Sensor

Created by Bryan Siepert



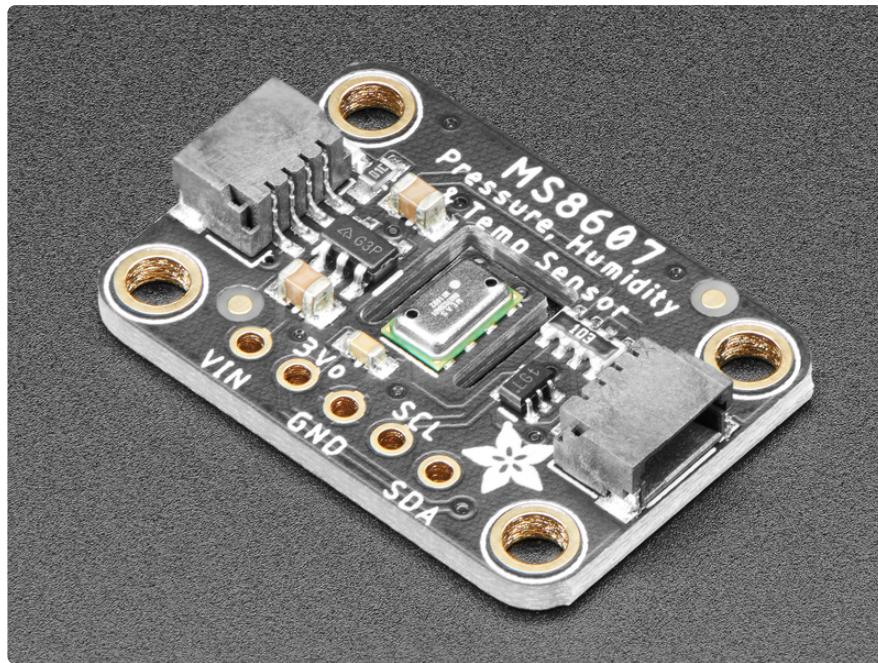
<https://learn.adafruit.com/adafruit-te-ms8607-pht-sensor>

Last updated on 2023-08-31 11:31:46 AM EDT

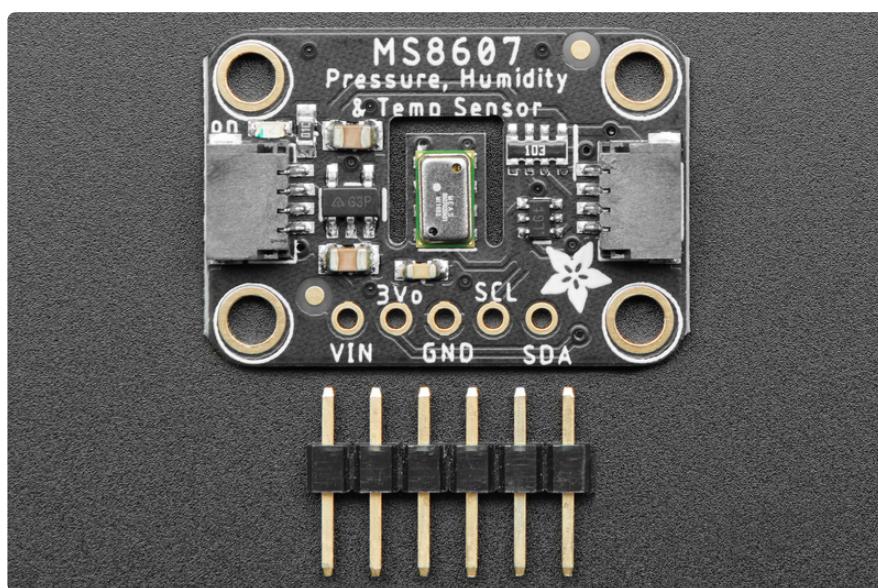
Table of Contents

| | |
|--|----|
| 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 |
| Downloads | 13 |
| • Files | |
| • Schematic | |
| • Fab Print | |

Overview

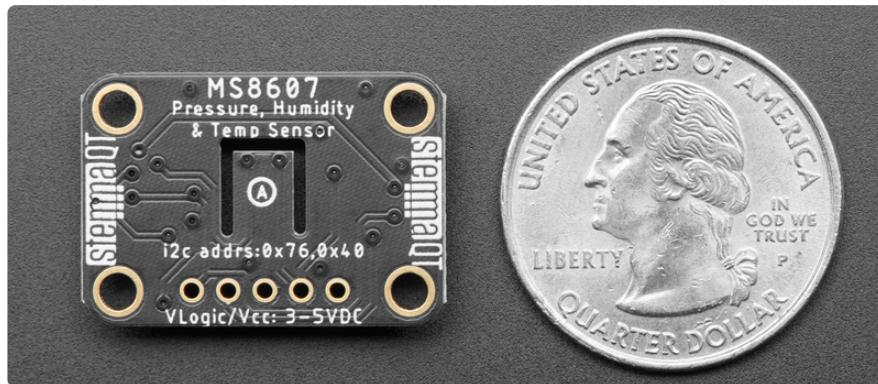


Because of their many applications, Pressure, Temperature, and Humidity sensors (PHT sensors) are a common offering from semiconductor companies. The MS8607 PHT sensor from TE Connectivity does an admirable job of measuring a wide range of all three environmental conditions. Each and every MS8607 sensor 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 \(\)](#)

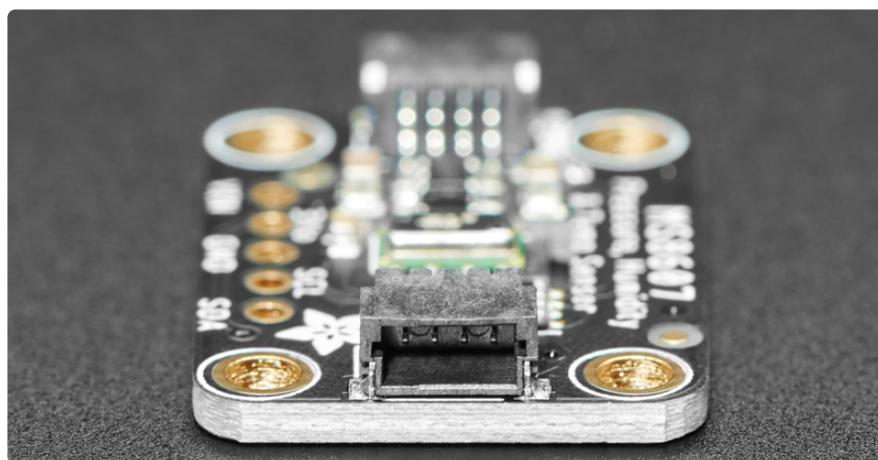


The MS8607 provides the previously mentioned factory-calibrated pressure, temperature, and humidity data are recorded as 16-bit values and made available over

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 \(\)](#) compatible [STEMMA QT \(\)](#) connectors to wire up to your favorite micro with a [plug-and-play cable \(\)](#). 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 \(\)](#) - These connectors allow you to connect to dev boards with STEMMA QT connectors or to other things with [various associated accessories \(\)](#)

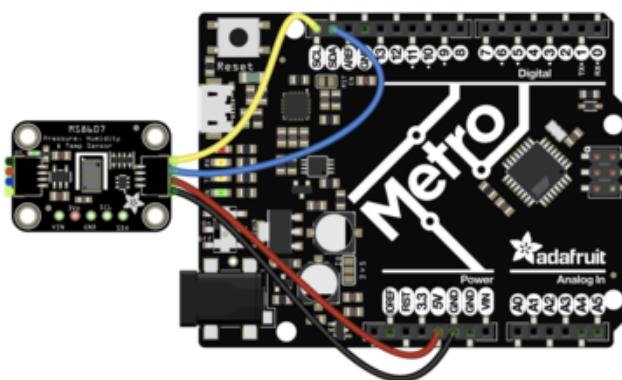
Arduino

Using the MS8607 with Arduino is a simple matter of wiring up the sensor to your Arduino-compatible microcontroller, installing the [Adafruit MS8607 \(\)](#) 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 \(\)](#) connectors. The examples show a Metro but wiring will work the same for an Arduino or other compatible board.



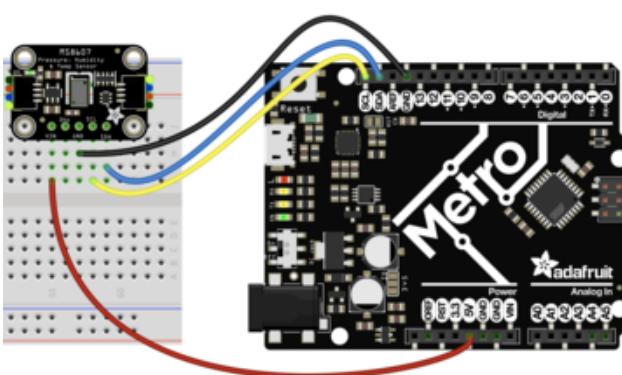
Connect board VIN (red wire) to Arduino 5V if you are running a 5V board Arduino (Uno, etc.). If your board is 3V, connect to that instead.

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.

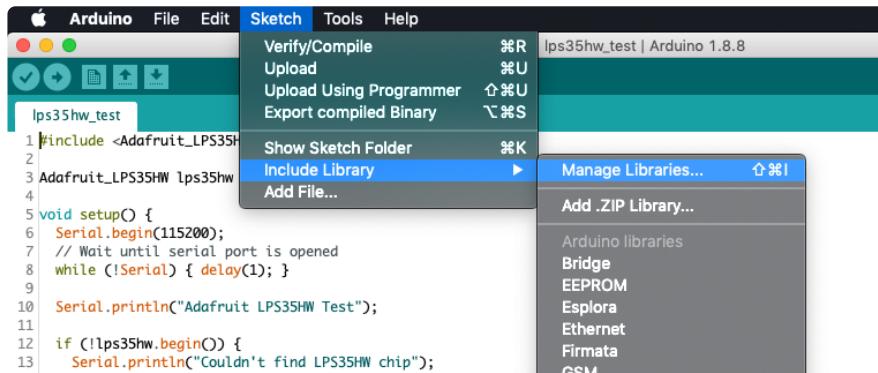
Connect board GND (black wire) to Arduino GND

Connect board SCL (yellow wire) to Arduino SCL

Connect board SDA (blue wire) to Arduino SDA

Library Installation

You can install the [Adafruit MS8607](#) 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_OS_8b);
    Serial.print("Humidity resolution set to ");
    switch (ms8607.getHumidityResolution()){
        case MS8607_HUMIDITY_RESOLUTION_OS_12b: Serial.println("12-bit"); break;
        case MS8607_HUMIDITY_RESOLUTION_OS_11b: Serial.println("11-bit"); break;
        case MS8607_HUMIDITY_RESOLUTION_OS_10b: Serial.println("10-bit"); break;
        case MS8607_HUMIDITY_RESOLUTION_OS_8b: Serial.println("8-bit"); break;
    }
    // ms8607.setPressureResolution(MS8607_PRESSURE_RESOLUTION_OS_4096);
    Serial.print("Pressure and Temperature resolution set to ");
    switch (ms8607.getPressureResolution()){
        case MS8607_PRESSURE_RESOLUTION_OS_256: Serial.println("256"); break;
        case MS8607_PRESSURE_RESOLUTION_OS_512: Serial.println("512"); break;
        case MS8607_PRESSURE_RESOLUTION_OS_1024: Serial.println("1024"); break;
        case MS8607_PRESSURE_RESOLUTION_OS_2048: Serial.println("2048"); break;
        case MS8607_PRESSURE_RESOLUTION_OS_4096: Serial.println("4096"); break;
        case MS8607_PRESSURE_RESOLUTION_OS_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 \(\)](#)

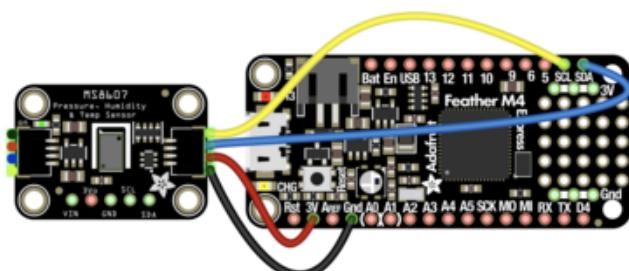
Python & CircuitPython

It's easy to use the MS8607 with Python or CircuitPython, and the [Adafruit CircuitPython MS8607 \(\)](#) 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 \(\)](#).

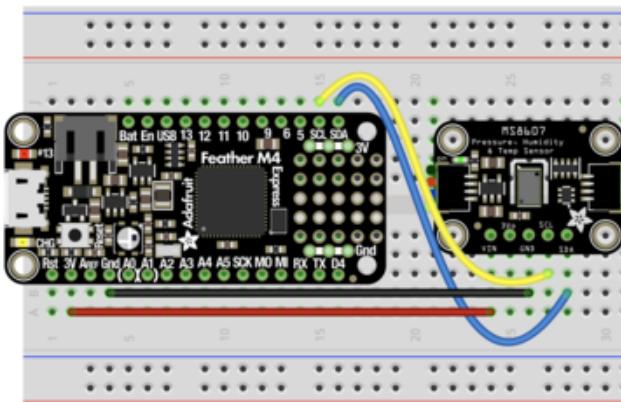
CircuitPython Microcontroller Wiring

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 \(\)](#) 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:

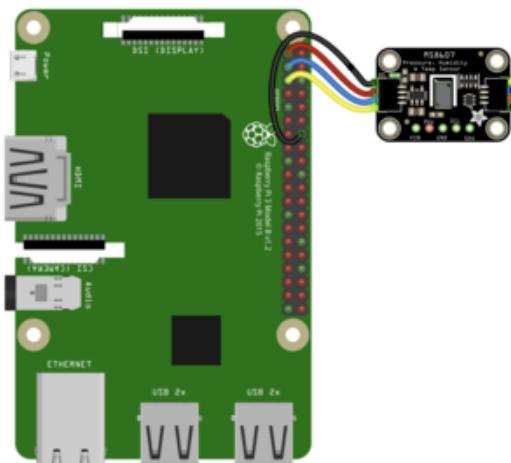


Board 3V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

Python Computer Wiring

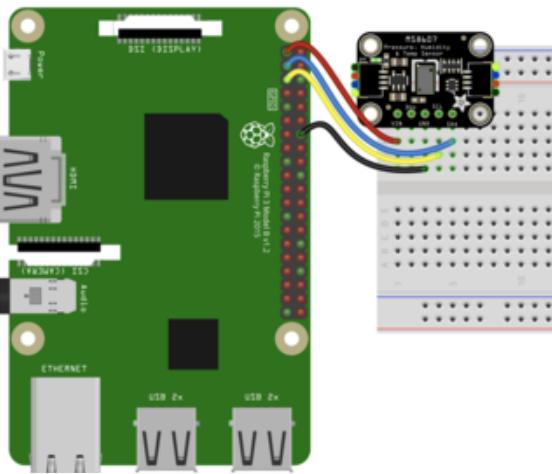
Since there's dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported \(\)](#).

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 MS8607 Library

You'll need to install the [Adafruit CircuitPython MS8607 \(\)](#) 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_ms8607.mpy file and adafruit_bus_device folder copied over.

Next [connect to the board's serial REPL \(\)](#)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 \(\)!](#)

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 STEMMA 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

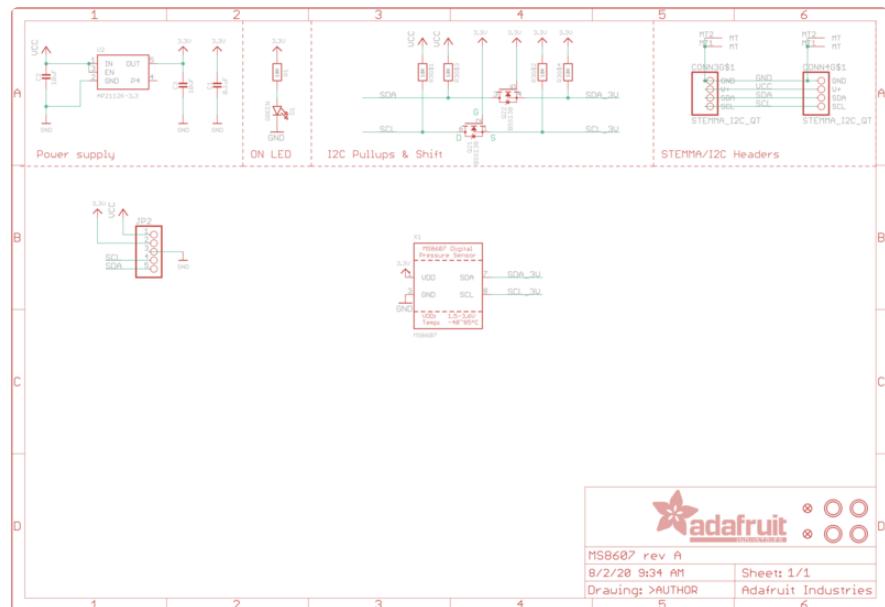
[Python Docs \(\)](#)

Downloads

Files

- [MS8607 Datasheet \(\)](#)
- [EagleCAD files on GitHub \(\)](#)
- [Fritzing object in the Adafruit Fritzing Library \(\)](#)

Schematic



Fab Print

