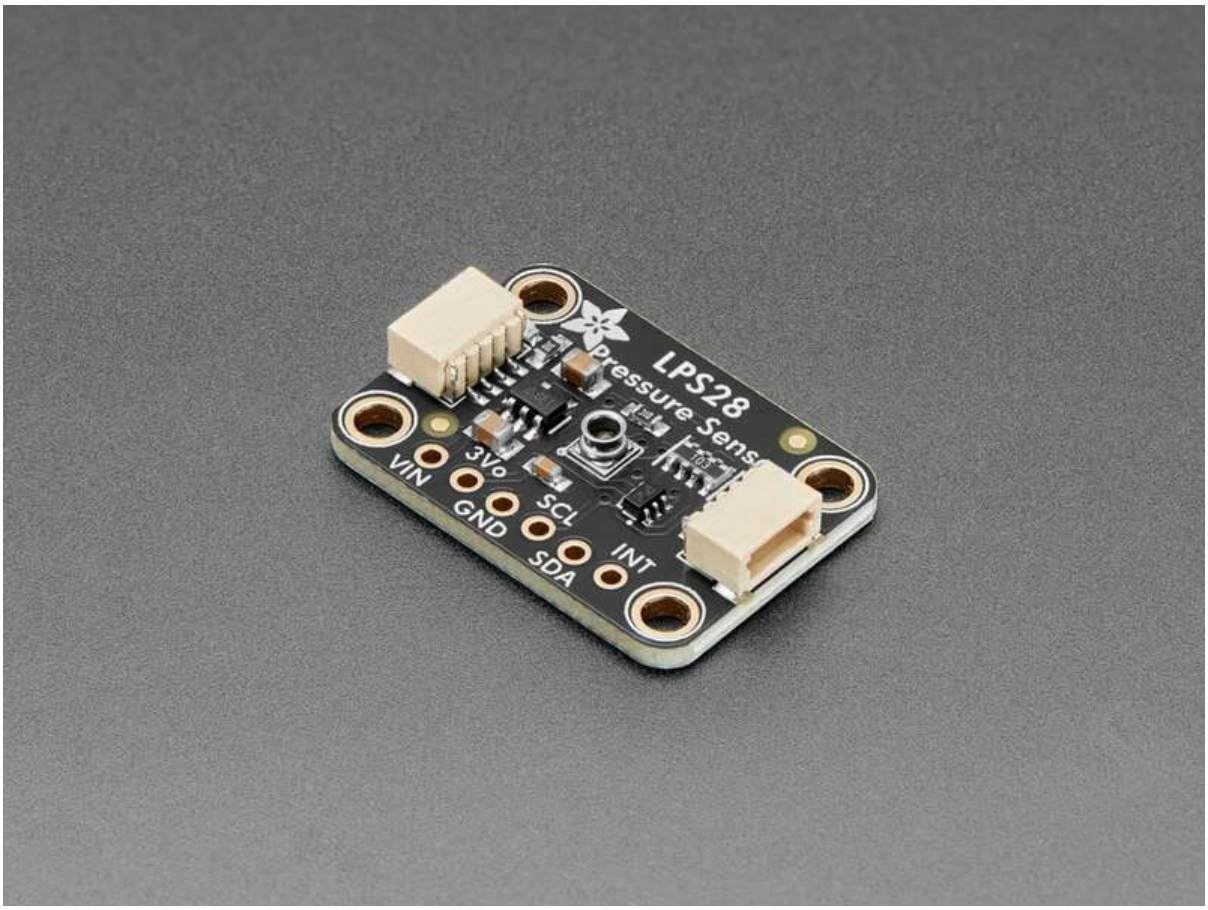
# Adafruit LPS28 Pressure Sensor

Created by Liz Clark

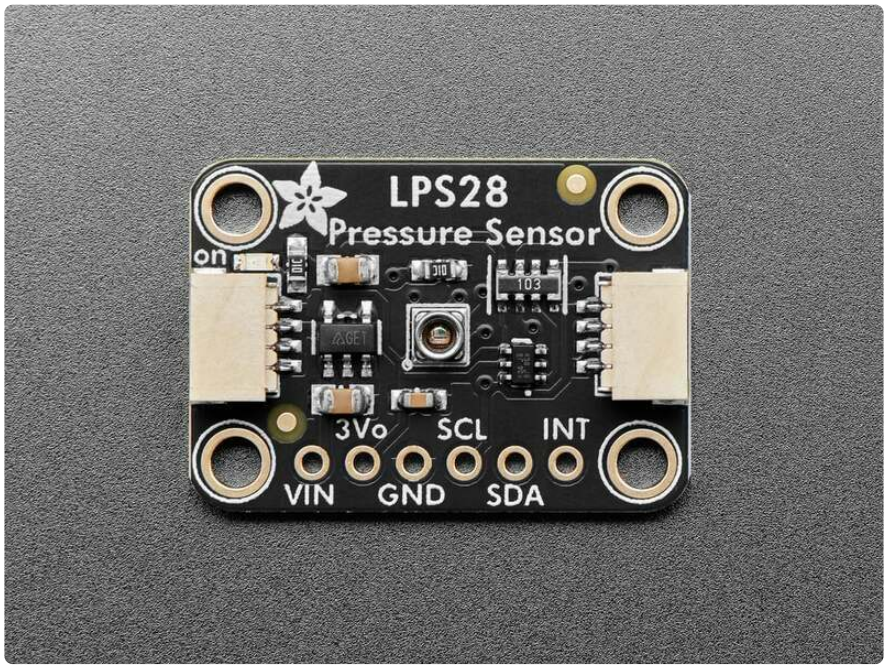



https://learn.adafruit.com/adafruit-lps28-pressure-sensor

Last updated on 2025-04-03 10:21:05 AM EDT

# Table of Contents

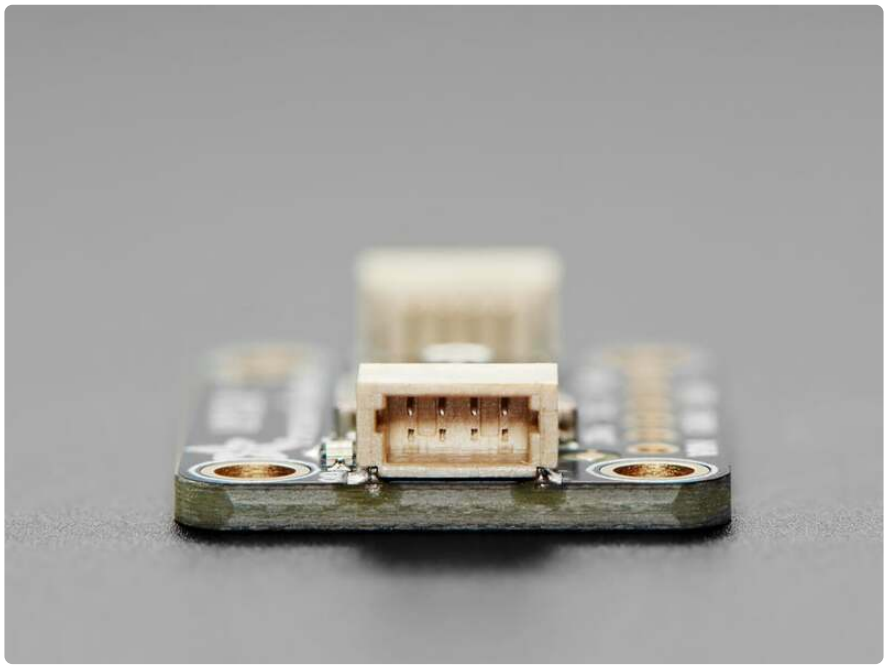
# Overview



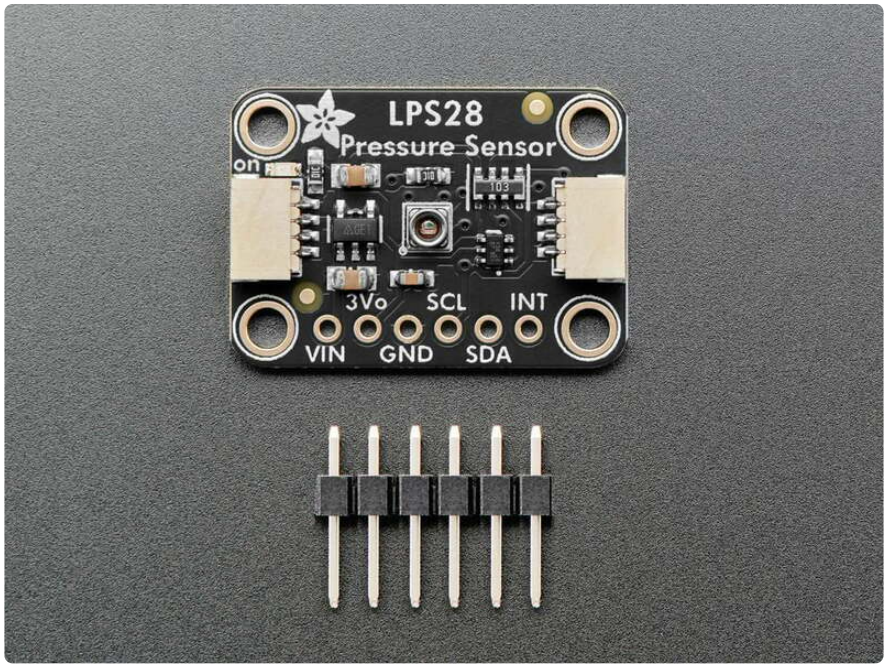

We stock a large number of barometric pressure sensors, and they all have slight differences: some are ported (http://adafru.it/3965), some are tiny (http://adafru.it/4633), some are popular and precise (http://adafru.it/2652) and some are low cost (http://adafru.it/4494). The **Adafruit LPS28 (LPS28DFW) Pressure Sensor** is unique in that it can handle a much higher range of pressures: almost every other sensor we've encountered tops out at about 1260 hPa. That means they're totally fine for measuring altitude for most human endeavors, we tend to stay above sea level. However, for some scientific or deep-water uses, 1260 will not cut it! That's why we're happy to stock a sensor that can go down to 4060 hPa (https://adafru.it/1aee) - that's as low as -43,000 feet below sea level in air or 40 meters underwater.


We covered the LPS28DFW sensor on our EYE ON NPI video segment (https://adafru.it/10na), and liked it so much we turned it into a product!

That makes this a good sensor for use with underwater devices or robots, as it can tell your depth from the pressure readings. Of course, its also still a great sensor for altitude sensing out of the water: with a 24-bit sensor and the ability to discern absolute pressure changes of ±0.5 hPa at the highest quality readings, good for less than 1 centimeter of altitude. (Like all pressure sensors, the actual altitude-above-sea-level depends on the days' pressure at sea level) Or, if you need speed, it can do up to 200 samples per second, without filtering and averaging.



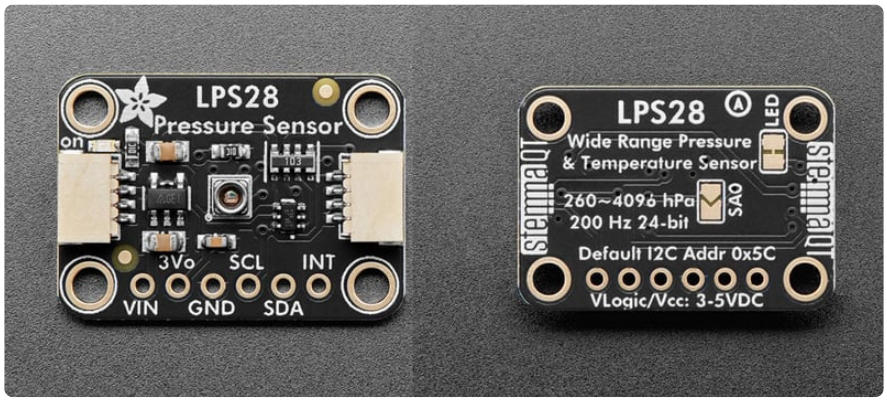
The sensor comes with a little metal port, so you could connect a thin tubing to it for measuring pressure far from where it is soldered. The sensor element within the package has potting gel good up to 10,000 hPa of pressure and protection against some harsh chemicals - [check the datasheet for more information to make sure you can use it to measure pressure of the liquid or gas you need](https://adafru.it/1aee).




To make life easier when working with this fancy sensor, we've taken the LPS28 and put it onto a breakout PCB along with support circuitry to let you use this little wonder with 3.3V (Feather/Raspberry Pi) or 5V (Arduino/ Metro328) logic levels.

Additionally, since it speaks I2C, you can easily connect it up with two wires (plus power and ground!). We've even included [SparkFun qwiic](https://adafru.it/Fpw) compatible [STEMMA QT](https://adafru.it/Ft4) **(https://adafru.it/Ft4)** connectors for the I2C bus so **you don't even need to solder!** Just wire up to your favorite micro and you can use our CircuitPython/Python or Arduino drivers to easily interface with the LPS28 and make approximate approximations of proximity in no time! **QT Cable is not included**, [but we have a variety in the shop](http://adafru.it/4210).

# Pinouts




The default I2C address is **0x5C**.

## Power Pins

- **VIN** - this is the power pin. 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
- **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's I2C clock line. This pin can use 3-5V logic, and there's a **10K pullup** on this pin.
- **SDA** - I2C data pin, connect to your microcontroller's I2C data line. This pin 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** (qwiic) connectors or to other things with various associated accessories (https://adafru.it/Ft6)

## Interrupt Pin

- **INT** - This is the interrupt pin. The default state is active high. You can change the state in software.

## Power LED and LED Jumper

- **Power LED -** In the upper left corner, above the STEMMA connector, on the front of the board, is the power LED, labeled **on**. It is a green LED.
- **LED jumper** - This jumper, labeled LED on the board silk, is located on the back of the board, in the upper right corner. Cut the trace on this jumper to cut power to the "on" LED.

## SA0 Jumper

- **SA0** - On the back of the board, towards the center, is the I2C address jumper. It is labeled **SA0** on the board silk. If you solder this jumper closed, you'll change the I2C address from the default **0x5C** to **0x5D**.

| ADDR | SA0 |
|---|---|
| 0x5C | L |
| 0x5D | H |

---

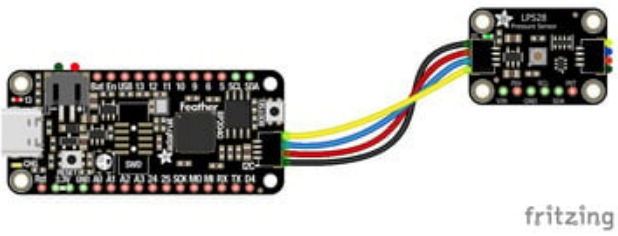
# CircuitPython and Python

It's easy to use the **LPS28** with Python or CircuitPython, and the Adafruit_CircuitPython_LPS28 (https://adafru.it/1aef) module. This module allows you to easily write Python code to read pressure and temperature data.

You can use this driver 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).

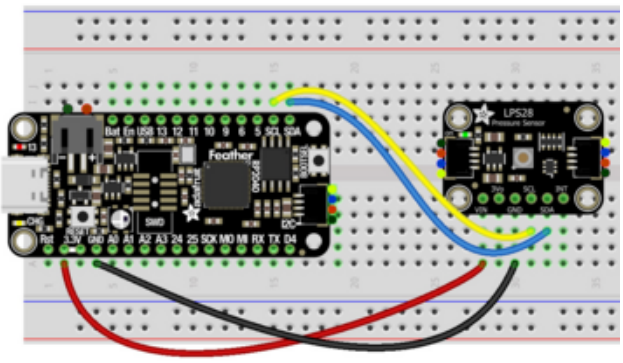
## CircuitPython Microcontroller Wiring

First wire up the sensor to your board exactly as follows. The following is the sensor wired to a Feather RP2040 using the STEMMA connector:




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

The following is the sensor wired to a Feather RP2040 using a solderless breadboard:

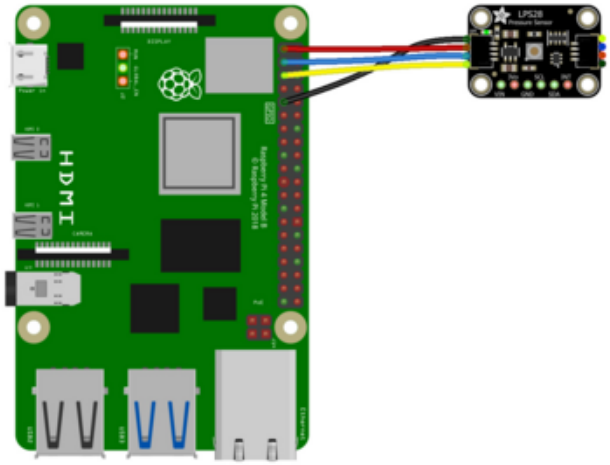


**Board 3V** to **sensor VIN (red wire)**
**Board GND** to **sensor GND (black wire)**
**Board SCL** to **sensor SCL (yellow wire)**
**Board SDA** to **sensor SDA (blue wire)**

## Python Computer Wiring

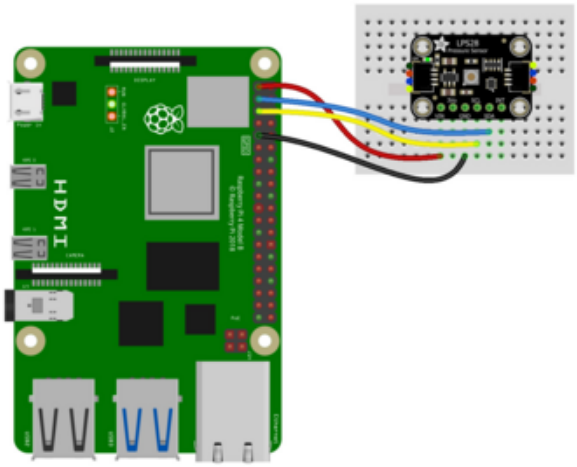
Since there are dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported](https://adafru.it/BSN) (https://adafru.it/BSN).

Here's the Raspberry Pi wired with I2C using the STEMMA connector:



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

Here's the Raspberry Pi wired with I2C using a solderless breadboard:



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

## Python Installation of LPS28 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)!

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

- `pip3 install adafruit-circuitpython-lps28`

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 Usage

To use with CircuitPython, you need to first install the **Adafruit_CircuitPython_LPS28** library, and its dependencies, into the **lib** folder on your **CIRCUITPY** drive. Then you need to update **code.py** with the example script.

Thankfully, we can do this in one go. In the example below, click the **Download Project Bundle** button below to download the necessary libraries and the **code.py** file in a zip file. Extract the contents of the zip file, and copy the **entire lib folder** and the **code.py** file to your **CIRCUITPY** drive.

Your **CIRCUITPY/lib** folder should contain the following folders and file:

- **adafruit_bus_device/**
- **adafruit_register/**
- **adafruit_lps28.mpy**

## Python Usage

Once you have the library `pip3` installed on your computer, copy or download the following example to your computer, and run the following, replacing **code.py** with whatever you named the file:

```
python3 code.py
```

## Example Code

**If running CircuitPython:** Once everything is saved to the **CIRCUITPY** drive, [connect to the serial console](https://adafru.it/Bec) (https://adafru.it/Bec) to see the data printed out!

**If running Python:** The console output will appear wherever you are running Python.

```
# SPDX-FileCopyrightText: Copyright (c) 2025 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time

import board

import adafruit_lps28

i2c = board.I2C()
sensor = adafruit_lps28.LPS28(i2c)

# Data Rate in hz
# 1, 4, 10, 25, 50, 75, 100 or 200 (default)
# sensor.data_rate = 200

# Number of samples to average per measurement
# 4 (default), 8, 16, 32, 64, 128, 512
# sensor.averaging = 4

# Full scale measurement mode for pressure
# (False = 1260 hPa, True = 4060 hPa (default))
# sensor.full_scale_mode = True

# Enable/Disable Interrupts Defaults

# sensor.data_ready_int = True # Data Ready Interrupt
# sensor.data_ready_pulse = False # Data-ready interrupt as a pulse
# sensor.fifo_full_int = False # FIFO full interrupt
# sensor.fifo_overrun_int = False # FIFO overrun interrupt
# sensor.fifo_watermark_int = False # FIFO watermark interrupt

print("LPS28 Simple Test")
print("-" * 40)

while True:
    if sensor.data_ready:
        print(f"Pressure: {sensor.pressure:.1f} hPa")
        print(f"Temperature: {sensor.temperature:.1f} °C")
        print("-" * 40)

    time.sleep(0.5)
```

First, the sensor is instantiated over I2C. Then, in the loop, the pressure and temperature readings are printed to the serial monitor.

```
Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
code.py output:
LPS28 Simple Test
----------------------------------------
Pressure: 1295.8 hPa
Temperature: 25.0 °C
----------------------------------------
Pressure: 1262.0 hPa
Temperature: 25.4 °C
----------------------------------------
Pressure: 1006.1 hPa
Temperature: 25.7 °C
----------------------------------------
Pressure: 1006.0 hPa
Temperature: 25.9 °C
----------------------------------------
Pressure: 1005.9 hPa
Temperature: 26.0 °C
----------------------------------------
Pressure: 1039.1 hPa
Temperature: 26.1 °C
----------------------------------------
```

Ln 44, Col 1 Spaces: 4 UTF-8 LF {} Python 3.11.4 Adafruit:Feather RP2040

# Python Docs

Python Docs (https://adafru.it/1aec)

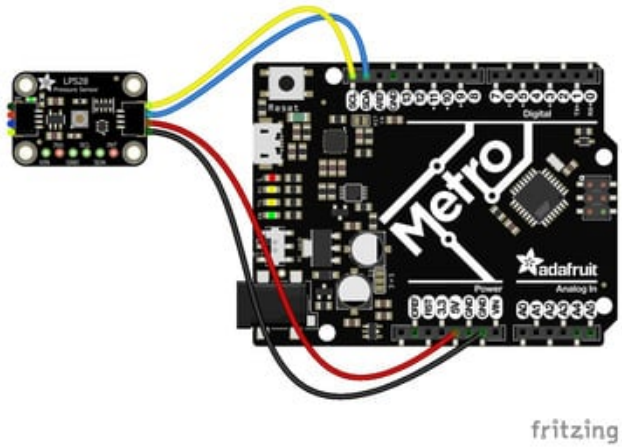
# Arduino

Using the LPS28 breakout with Arduino involves wiring up the breakout to your Arduino-compatible microcontroller, installing the Adafruit_LPS28 (https://adafru.it/1aeg) library, and running the provided example code.

## Wiring

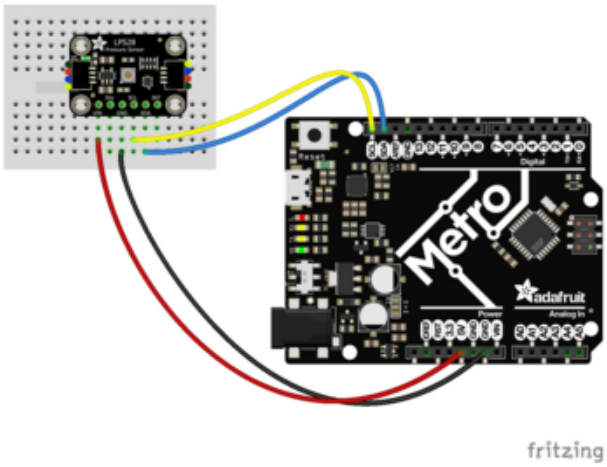
Wire as shown for a **5V** board like an Uno. If you are using a **3V** board, like an Adafruit Feather, wire the board's 3V pin to the sensor VIN.

Here is an Adafruit Metro wired up to the sensor using the STEMMA QT connector:




**Board 5V** 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)**

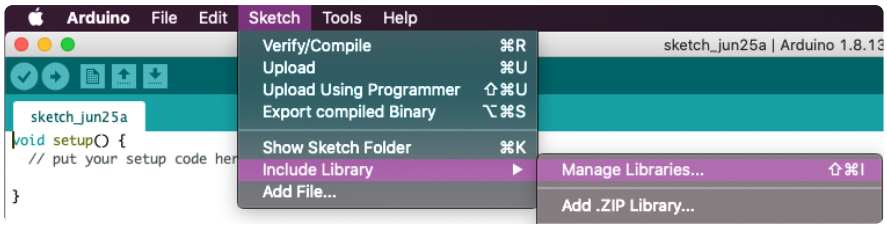
Here is an Adafruit Metro wired up using a solderless breadboard:



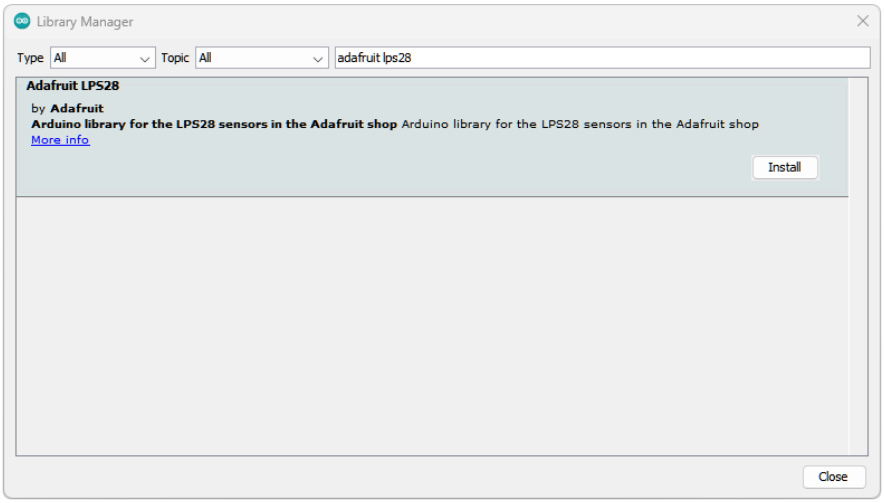

**Board 5V** to **sensor VIN (red wire)**
**Board GND** to **sensor GND (black wire)**
**Board SCL** to **sensor SCL (yellow wire)**
**Board SDA** to **sensor SDA (blue wire)**

## Library Installation

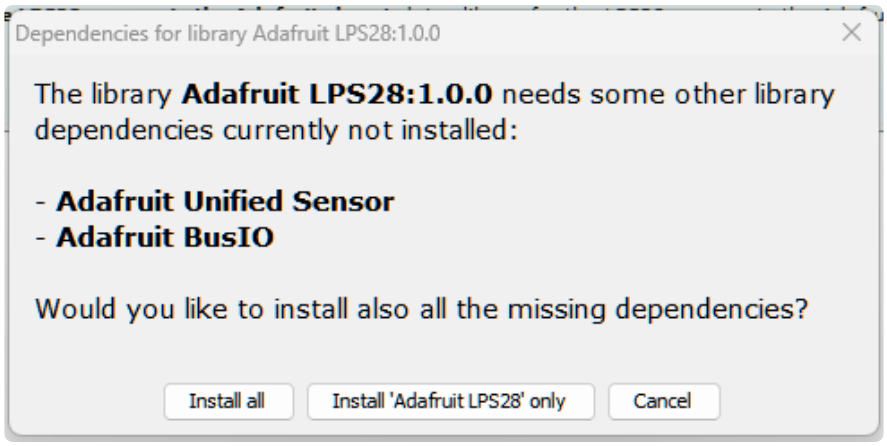
You can install the **Adafruit_LPS28** library for Arduino using the Library Manager in the Arduino IDE.




Click the **Manage Libraries ...** menu item, search for **Adafruit_LPS28**, and select the **Adafruit LPS28** library:




If asked about dependencies, click "Install all".




If the "Dependencies" window does not come up, then you already have the dependencies installed.

> If the dependencies are already installed, you must make sure you update them through the Arduino Library Manager before loading the example!

## Example Code

```
#include <Wire.h>
#include <Adafruit_LPS28.h>

// Instantiate the sensor
Adafruit_LPS28 lps28;

void setup() {
  Serial.begin(115200);
  while (!Serial) delay(10); // Wait for Serial monitor

  Serial.println("LPS28 Test");

  // Initialize the sensor
  if (!lps28.begin()) {
    Serial.println("Failed to find LPS28 sensor!");
    while (1) delay(10);
  }

  Serial.println("LPS28 sensor found!");

  // Set the highest ODR (200 Hz) and 4-sample averaging, new value every 20ms
  lps28.setDataRate(LPS28_ODR_200_HZ);
  lps28.setAveraging(LPS28_AVG_4);

  // Set range to 4060 hPa
  lps28.setFullScaleMode(true);

  // Enable DRDY interrupt on the interrupt pin
  lps28.setInterruptPin(
    true,  // Polarity: Active high
    false  // Pin mode: Push-pull
  );

  // Enable DRDY interrupt output on INT pin (we could use this with an interrupt)
```

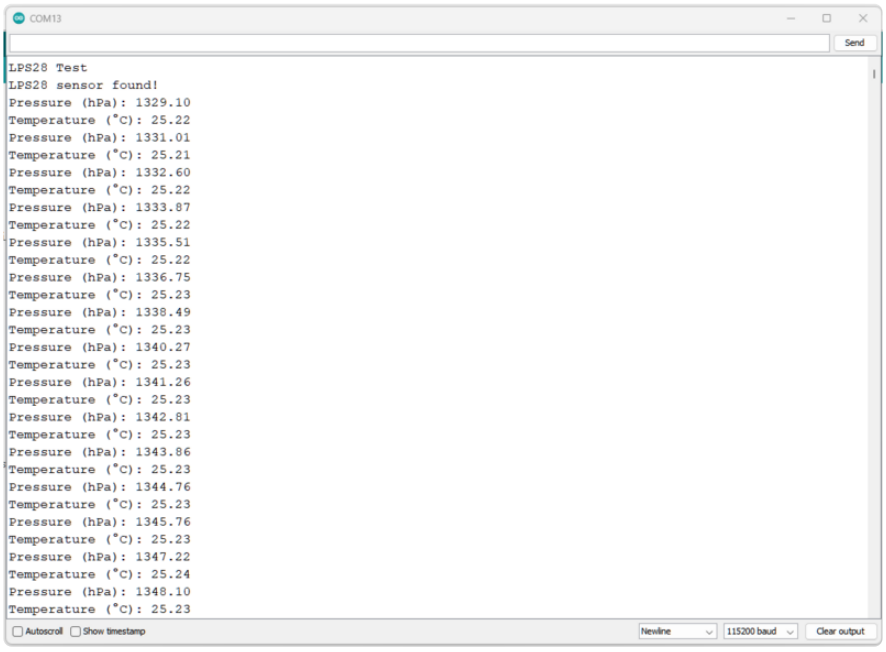
```
  lps28.setIntPinOutput(
    true,  // DRDY active
    false, // DRDY pulse not enabled
    false, // INT output not enabled
    false, // FIFO full interrupt not enabled
    false, // FIFO watermark interrupt not enabled
    false  // FIFO overrun interrupt not enabled
  );
}

void loop() {
  // Check if data is ready by reading the STATUS register
  if (lps28.getStatus() & LPS28_STATUS_PRESS_READY) { // Pressure data available
    // Read and print pressure and temperature
    float pressure = lps28.getPressure();
    float temperature = lps28.getTemperature();

    Serial.print("Pressure (hPa): ");
    Serial.println(pressure);

    Serial.print("Temperature (°C): ");
    Serial.println(temperature);
  }

  delay(10); // Polling delay
}
```



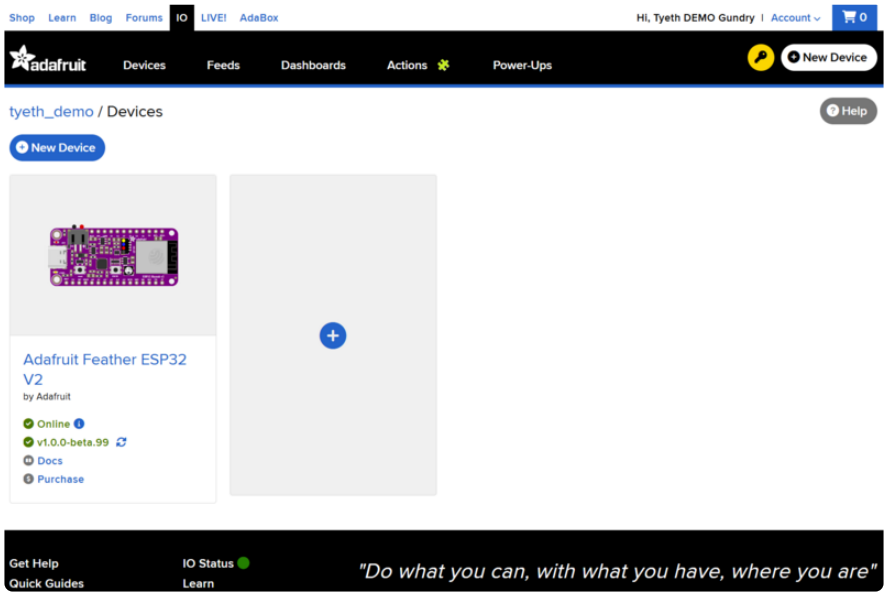

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. You'll see the LPS28 recognized over I2C. Then, pressure and temperature readings will be printed to the Serial Monitor.

---

# Arduino Docs

Arduino Docs (https://adafru.it/1aed)

# 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), a web platform designed (by Adafruit! (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.

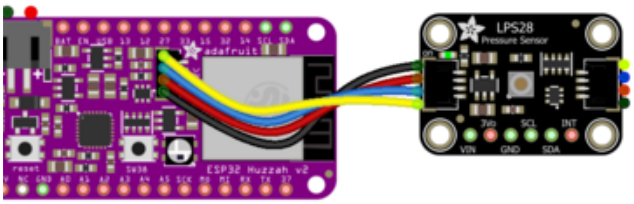
**Quickstart: Adafruit IO WipperSnapper**

https://adafru.it/Vfd

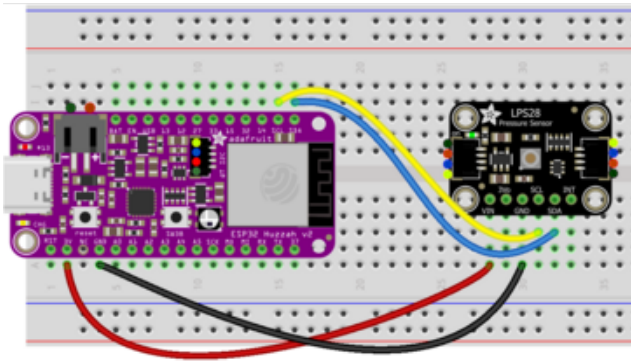
## Wiring

First, wire up an LPS28 (LPS28DFW) to your board exactly as follows. Here is an example of the LPS28 wired to an Adafruit ESP32 Feather V2 (http://adafru.it/

5400) using I2C with a STEMMA QT cable (no soldering required) (http://adafru.it/4210)



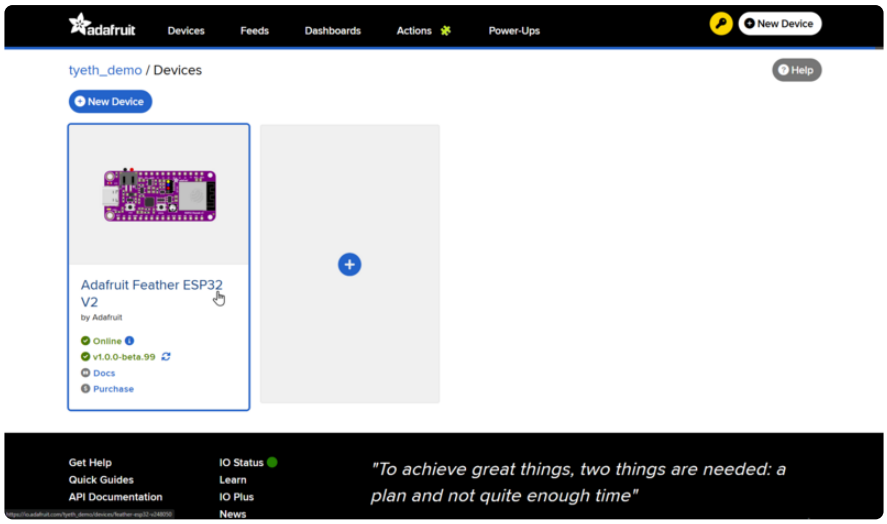

**Board 3V** to **sensor VIN (red wire on STEMMA QT)**
**Board GND** to **sensor GND (black wire on STEMMA QT)**
**Board SCL** to **sensor SCK (yellow wire on STEMMA QT)**
**Board SDA** to **sensor SDI (blue wire on STEMMA QT)**



## Usage

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

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



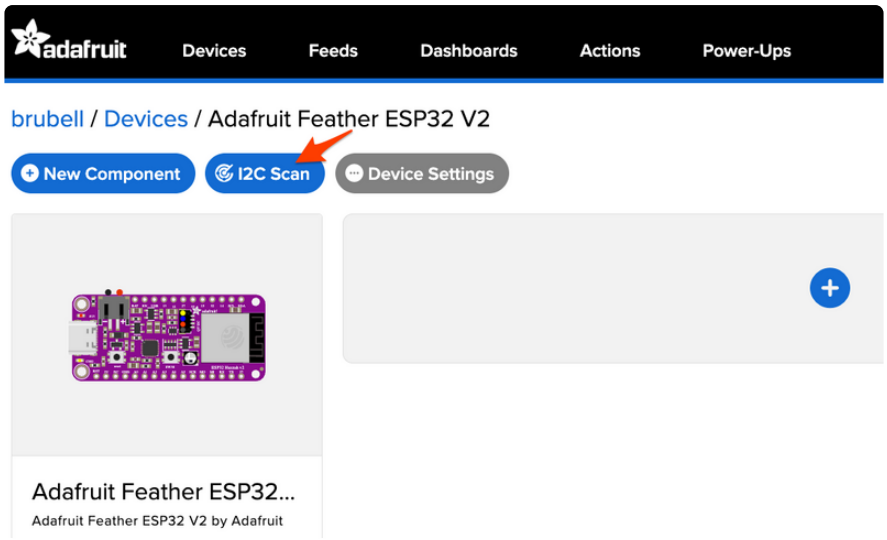

If you do not see your board listed here - you need [to connect your board to Adafruit IO (https://adafru.it/Vfd)](#) first.


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

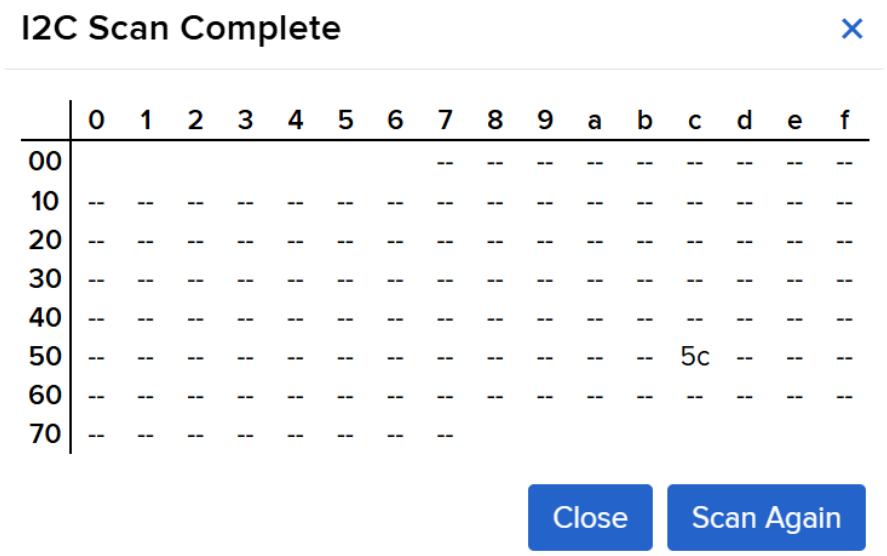
The device tile on the left indicates the version number of the firmware running on the connected board.


**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)](#) 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 LPS28's default I2C address of `0x5C` 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 | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| 40 | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| 50 | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | 5c | -- | -- | -- |
| 60 | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| 70 | -- | -- | -- | -- | -- | -- | -- | -- | | | | | | | | |

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

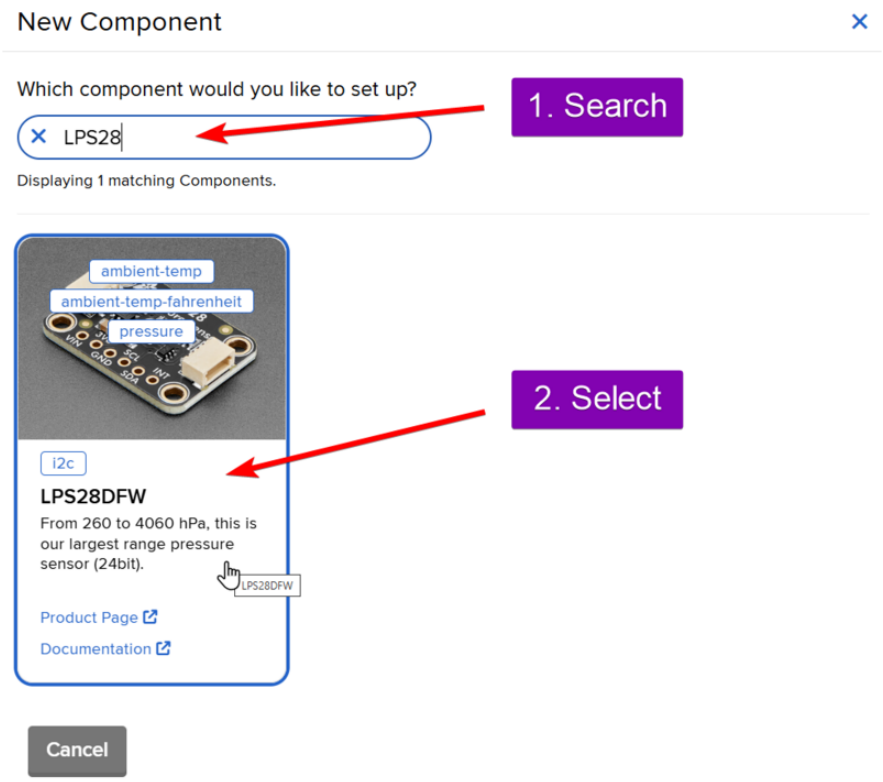
First, double-check the connection and/or wiring between the sensor and the board.

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.

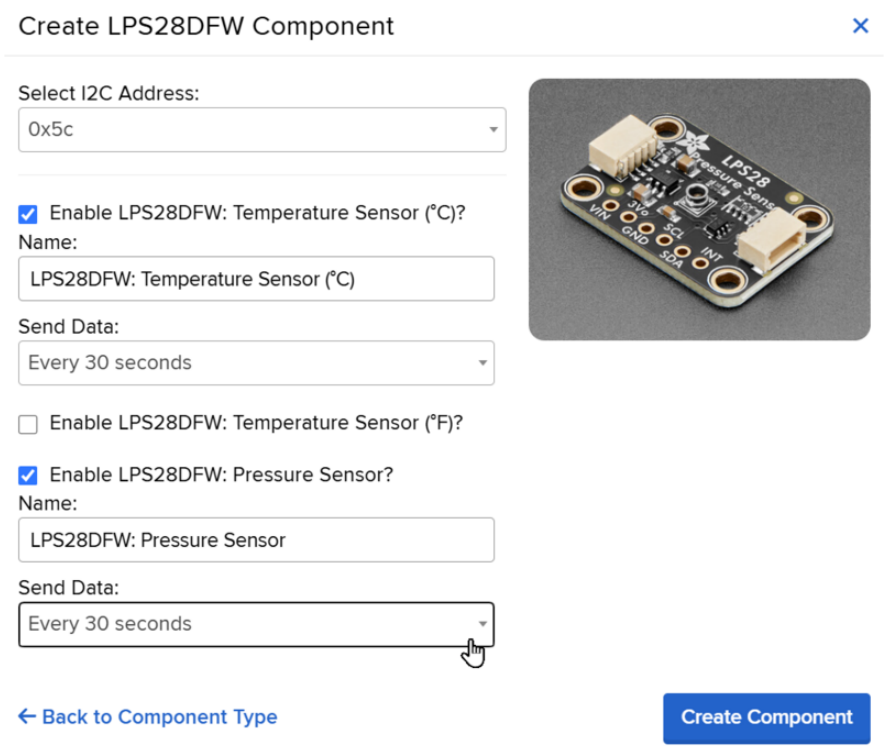

Adafruit IO supports a large amount of components. To quickly find your sensor, type `LPS28` into the search bar, then select the **LPS28** component.



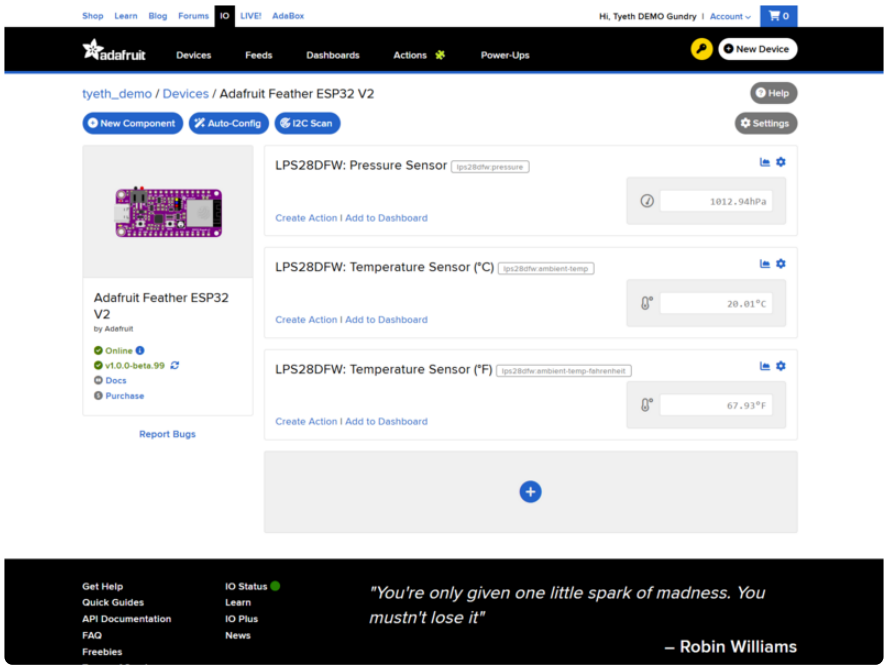

On the component configuration page, the **LPS28**'s sensor address should be listed along with the sensor's settings.

The **Send Every** option is specific to each sensor's measurements. This option will tell the Feather how often it should read from the LPS28 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.

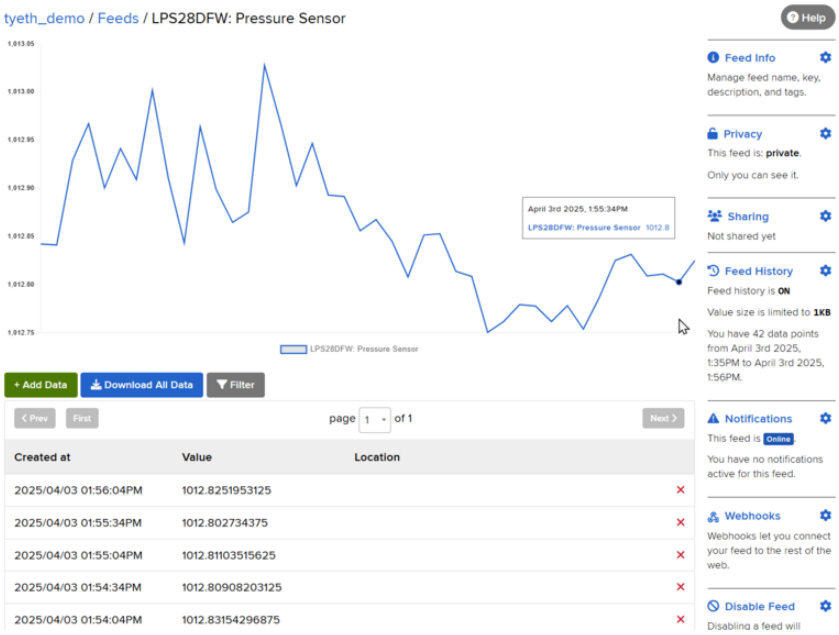



Your device interface should now show the sensor components you created. After the interval you configured elapses, WipperSnapper will automatically read values from the sensor(s) and send them to Adafruit IO.




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




# Downloads

## Files

- [LPS28DFW Datasheet](https://adafru.it/1aeh)
- [EagleCAD PCB Files on GitHub](https://adafru.it/1aei)
- [Fritzing object in the Adafruit Fritzing Library](https://adafru.it/1aej)

# Schematic and Fab Print