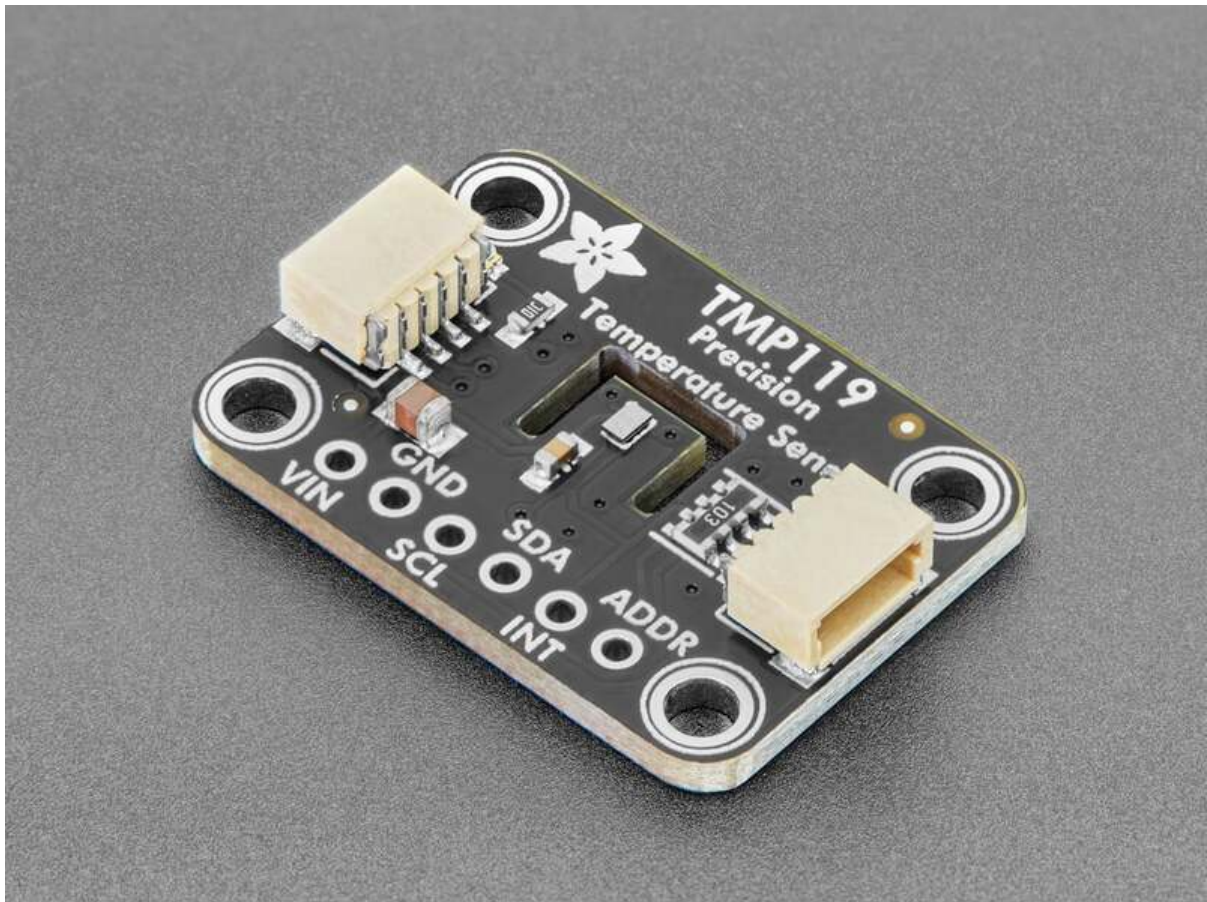




Adafruit TMP119 High-Precision Temperature Sensor

Created by Liz Clark



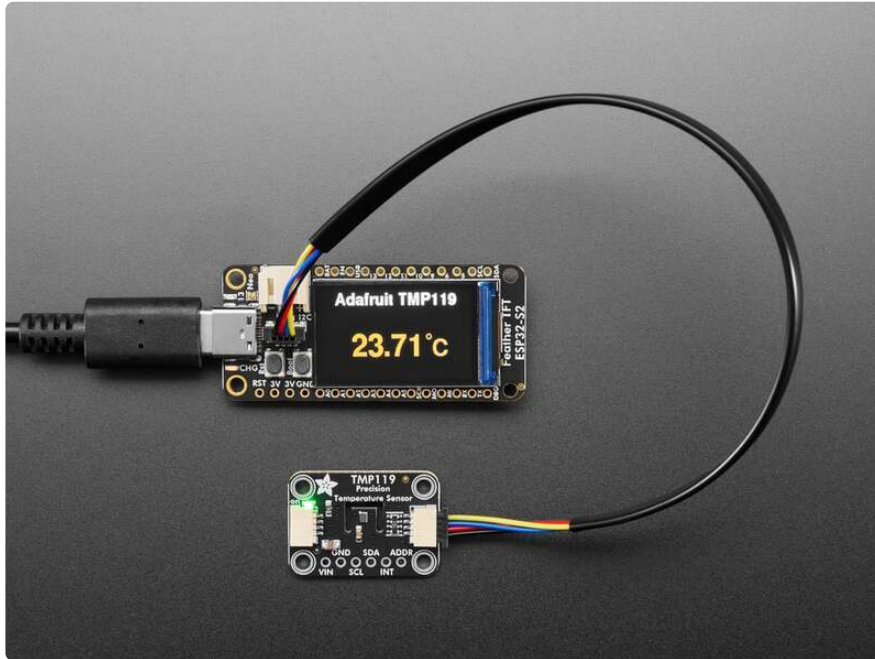
<https://learn.adafruit.com/adafruit-tmp119-high-precision-temperature-sensor>

Last updated on 2026-04-08 10:18:43 PM UTC

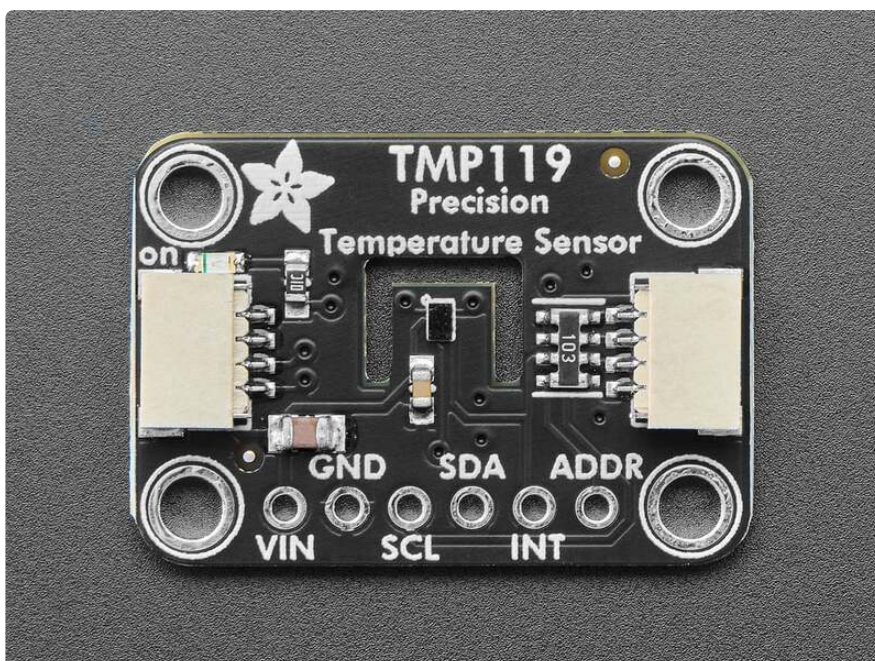
Table of Contents

Overview	3
Pinouts	5
<ul style="list-style-type: none">• Power Pins• I2C Logic Pins• Alert Pin• Address Pin and Jumper• Power LED and Jumper	
CircuitPython and Python	7
<ul style="list-style-type: none">• CircuitPython Microcontroller Wiring• Python Computer Wiring• Python Installation of TMP117 Library• CircuitPython Usage• Python Usage• Example Code	
Python Docs	11
Arduino	11
<ul style="list-style-type: none">• Wiring• Library Installation• Example Code	
Arduino Docs	15
WipperSnapper	16
<ul style="list-style-type: none">• What is WipperSnapper• Wiring• Usage	
Downloads	23
<ul style="list-style-type: none">• Files• Schematic and Fab Print	

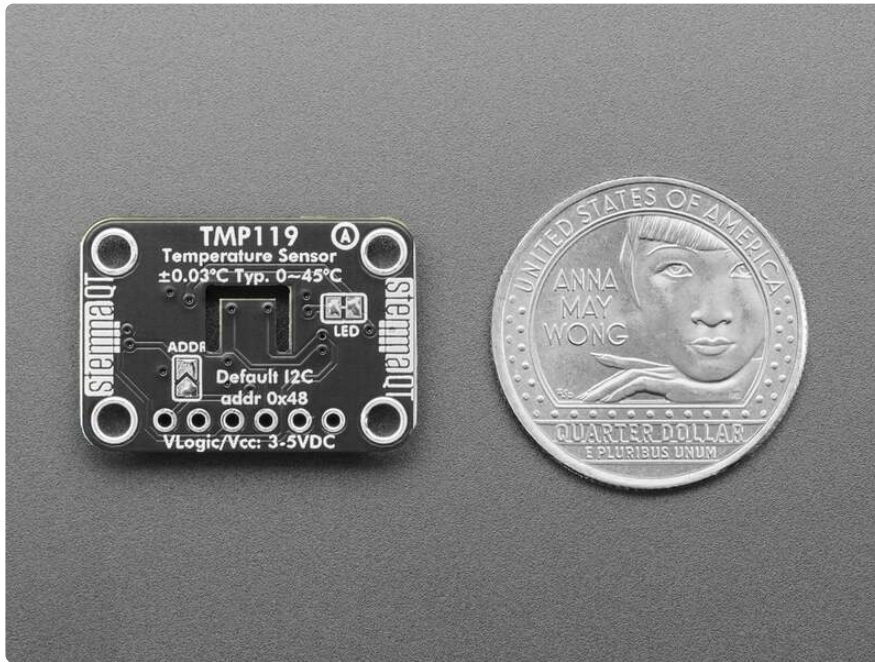
Overview



The **TMP119 High-Precision Temperature Sensor** is an I2C temperature sensor that will help you easily add super-precise temperature measurement and adjustment to your project. In addition to the obvious support for reading the temperature, the TMP119 can also monitor the temperature and alert you when corrective action needs to be taken.



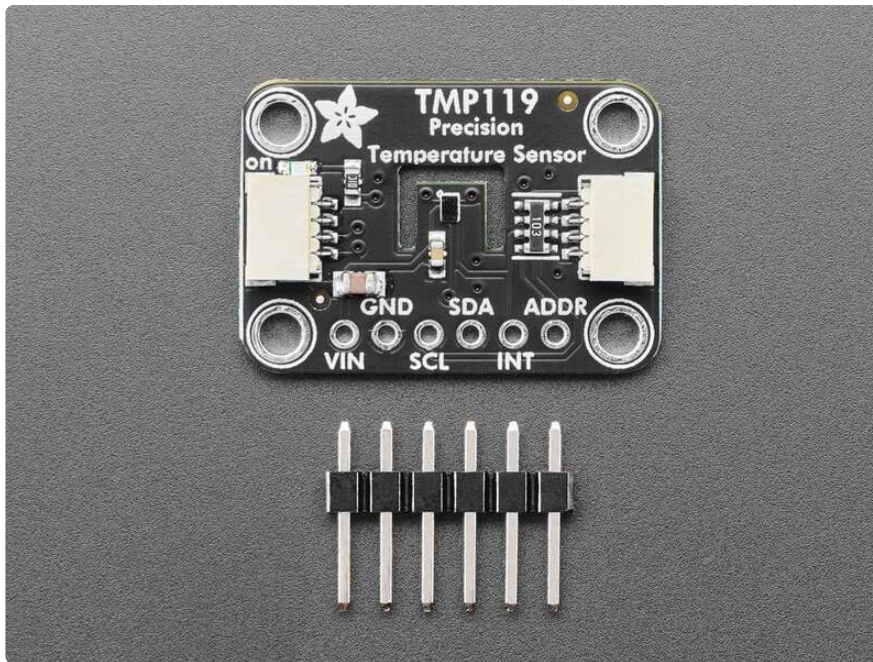
The TMP119 has **16-bit / 0.0078°C** measurement resolution, wide operating range of -55 to 155°C, and up to **±0.03°C** (typical) accuracy from 0°C to 45°C. The chip also has high and low **temperature alerts** and interrupt support, and hardware support required for NIST traceability ([contact TI for the NIST documents \(https://adafru.it/1aB2\)](https://adafru.it/1aB2)) This temperature sensor is perfect for applications where you need to keep a close eye on temperature. The manufacturer, Texas Instruments, suggests it for use in sensitive applications like thermostats and cold chain asset tracking, or even gas and heat meters!



TMP119 temperature sensor has different accuracy for different ranges, here's what you can expect as the typical & maximum variation for popular temperature ranges:

- ±0.03°C (typical) from 0°C to 45°C
- ±0.08°C (maximum) from 0°C to 45°C
- ±0.09°C (maximum) from -25°C to 85°C
- ±0.12°C (maximum) from -40°C to 85°C
- ±0.15°C (maximum) from -55°C to 125°C
- ±0.2°C (maximum) from -55°C to 150°C

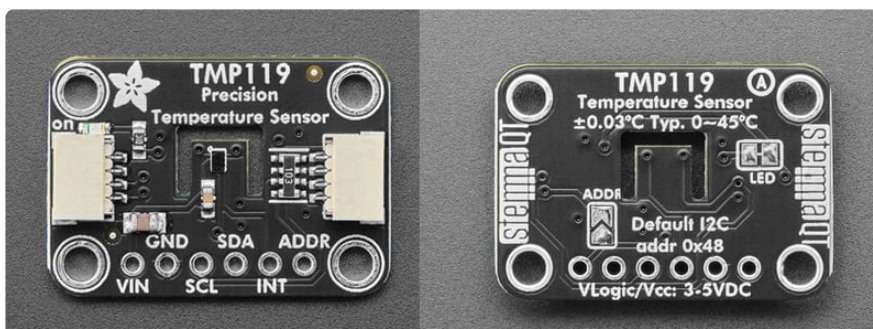
Compared to the TMP117, the TMP119 has better accuracy but has the same firmware interface (other than the Chip Identifier being different so you can tell which one you have)



To make using it as easy as possible, we've put the TMP119 on a breakout PCB in our [Stemma QT form factor](https://adafruit.it/19gF) (<https://adafruit.it/19gF>) with a sprinkle of support circuitry to give you options when testing. You can either use a breadboard or the [SparkFun gwiic](https://adafruit.it/Fpw) (<https://adafruit.it/Fpw>) compatible [STEMMA QT](https://adafruit.it/Ft4) (<https://adafruit.it/Ft4>) connectors, and compatibility with 5V voltage levels as commonly found on [Arduinos](https://adafruit.it/P4a) (<https://adafruit.it/P4a>), as well as 3.3V logic used by many other boards like the Raspberry Pi or our Feathers. **QT Cable is not included, but we have a variety in the shop** (<https://adafruit.it/17VE>) for quick plug-and-play support.

To get started, all you need to do is look over the [Pinouts](https://adafruit.it/1aB3) (<https://adafruit.it/1aB3>) page to familiarize yourself with the board, and then [use the TMP117/TMP119 Arduino or Python & CircuitPython pages for instructions on how to wire up the TMP119 to your board, as well as libraries and example code to get you started.](https://adafruit.it/1aB4) (<https://adafruit.it/1aB4>)

Pinouts



The default I2C address is **0x48**.

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 microcontroller like Arduino, use 5V. For a 3.3V microcontroller, use 3.3V.
- **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>).

Alert Pin

- **INT** - This is the alert pin. It can be used as an overtemperature alert or data-ready signal. The pin is an open-drain output that requires a pullup resistor.

Address Pin and Jumper

- **ADDR** - The address pin for setting the I2C address. You can chain up to two of these boards together on one I2C bus. Leave this pin **low** for default I2C address **0x48** or tie it **high** for I2C address **0x49**.
- **ADDR Jumper** - On the back of the board is the address jumper, labeled **ADDR**. You can leave this jumper open (low) to keep the board on the default I2C address 0x48. Solder the jumper closed (high) to change the I2C address to 0x49.

ADDR	ADDR
0x48	L
0x49	H

Power LED and Jumper

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

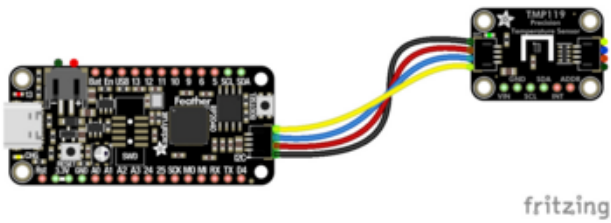
CircuitPython and Python

It's easy to use the **TMP119** with Python or CircuitPython, and the [Adafruit_CircuitPython_TMP117 \(https://adafru.it/P3B\)](https://adafru.it/P3B) module. This module allows you to easily write Python code that allows you to read **TMP117** and **TMP119** temperature sensors. 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).

The TMP119 has better accuracy and a different chip identifier from the TMP117, but is otherwise 100% compatible with the TMP117 firmware interface.

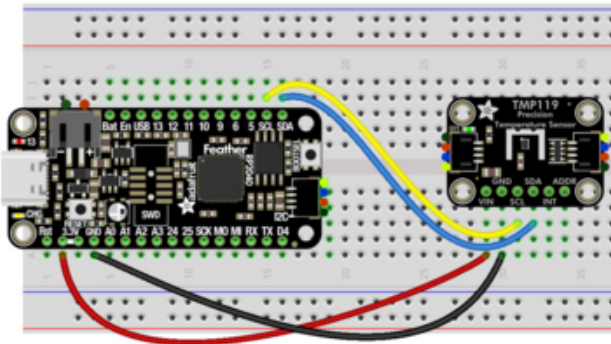
CircuitPython Microcontroller Wiring

First, wire up an TMP119 to your board exactly as shown below. Here's an example of wiring a Feather RP2040 to the TMP119 with I2C using one of the handy [STEMMA QT \(https://adafru.it/Ft4\)](https://adafru.it/Ft4) connectors:



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

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

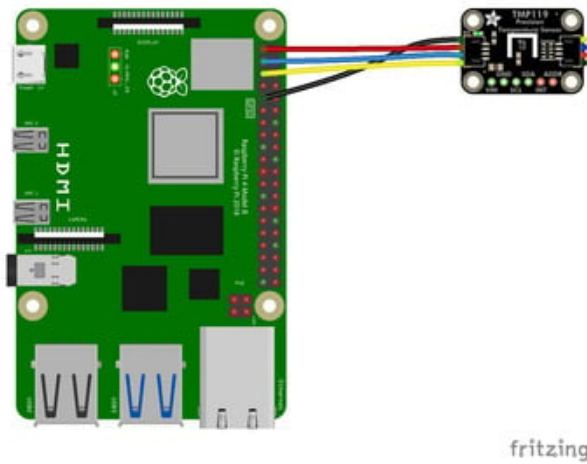


- Board 3V to breakout VIN (red wire)
- Board GND to breakout GND (black wire)
- Board SCL to breakout SCL (yellow wire)
- Board SDA to breakout SDA (blue wire)

Python Computer Wiring

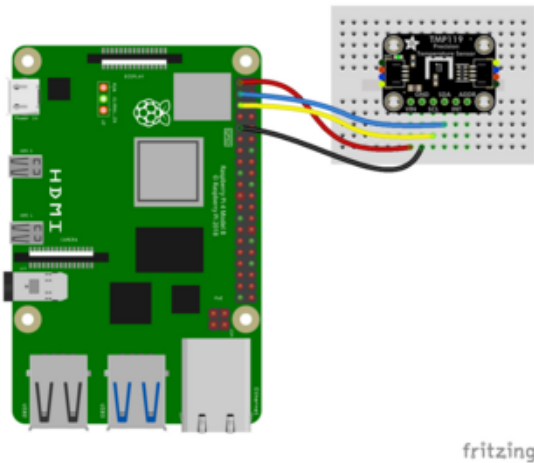
Since there's dozens of Linux computers/boards you can use, below shows wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported](https://adafru.it/BSN) (<https://adafru.it/BSN>).

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



Pi 3V to breakout STEMMA VIN (red wire)
Pi GND to breakout STEMMA GND (black wire)
Pi SCL to breakout STEMMA SCL (yellow wire)
Pi SDA to breakout STEMMA SDA (blue wire)

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



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

Python Installation of TMP117 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:

- `pip3 install adafruit-circuitpython-tmp117`

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

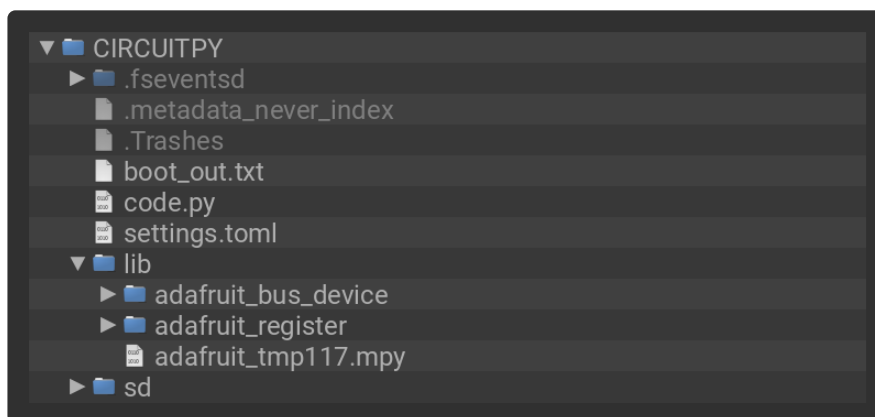
CircuitPython Usage

To use with CircuitPython, you need to first install the TMP117 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_tmp117.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: 2020 Bryan Siepert, written for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense
import time

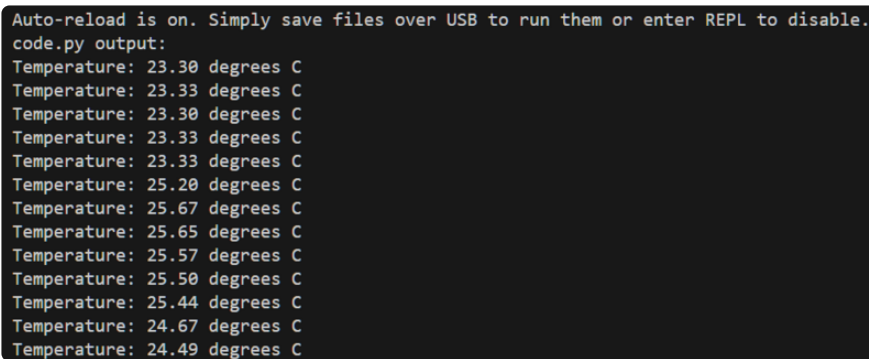
import board

import adafruit_tmp117

i2c = board.I2C() # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C() # For using the built-in STEMMA QT connector on a
microcontroller
tmp117 = adafruit_tmp117.TMP117(i2c)

while True:
    print("Temperature: %.2f degrees C" % tmp117.temperature)
    time.sleep(1)
```

In the example, the temperature sensor is instantiated on I2C. Then, in the loop, the temperature readings in Celsius are printed to the serial console every second.



Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
code.py output:
Temperature: 23.30 degrees C
Temperature: 23.33 degrees C
Temperature: 23.30 degrees C
Temperature: 23.33 degrees C
Temperature: 23.33 degrees C
Temperature: 25.20 degrees C
Temperature: 25.67 degrees C
Temperature: 25.65 degrees C
Temperature: 25.57 degrees C
Temperature: 25.50 degrees C
Temperature: 25.44 degrees C
Temperature: 24.67 degrees C
Temperature: 24.49 degrees C

Python Docs

[Python Docs \(https://adafru.it/1aAY\)](https://adafru.it/1aAY)

Arduino

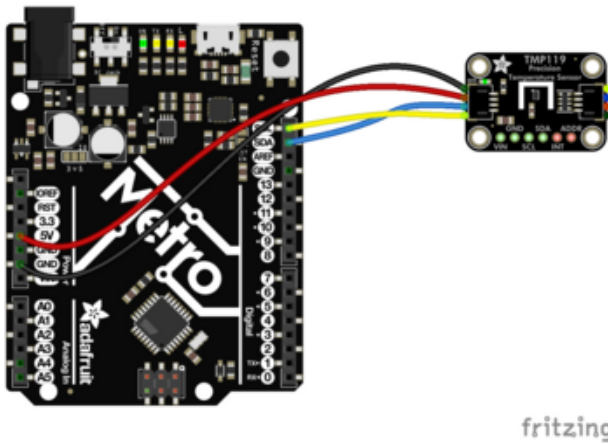
Using the TMP119 temperature sensor with Arduino involves wiring up the sensor to your Arduino-compatible microcontroller, installing the [Adafruit_TMP117 \(https://adafru.it/P3A\)](https://adafru.it/P3A) library and running the provided example code.

The TMP119 has better accuracy and a different chip identifier from the TMP117, but is otherwise 100% compatible with the TMP117 firmware interface.

Wiring

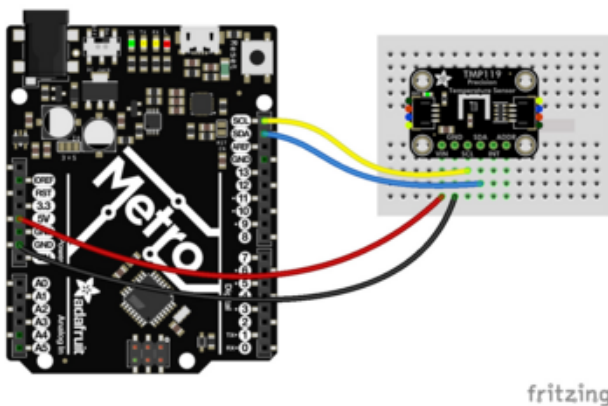
Wire as shown for a **5V** board like an Uno. If you are using a **3V** board, like an Adafruit Feather, wire the board's 3V pin to the TMP119 VIN.

Here is an Adafruit Metro wired up to the TMP119 using the STEMMA QT connector:



- Board 5V to breakout STEMMA VIN (red wire)
- Board GND to breakout STEMMA GND (black wire)
- Board SCL to breakout STEMMA SCL (yellow wire)
- Board SDA to breakout STEMMA SDA (blue wire)

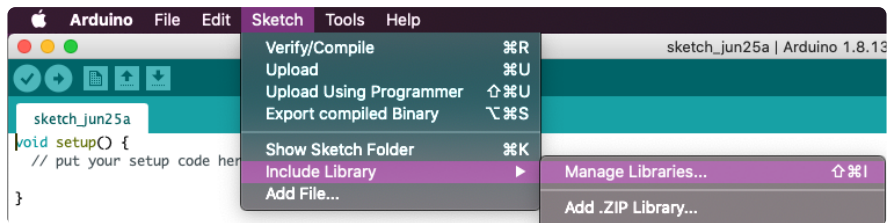
Here is an Adafruit Metro wired up using a solderless breadboard:



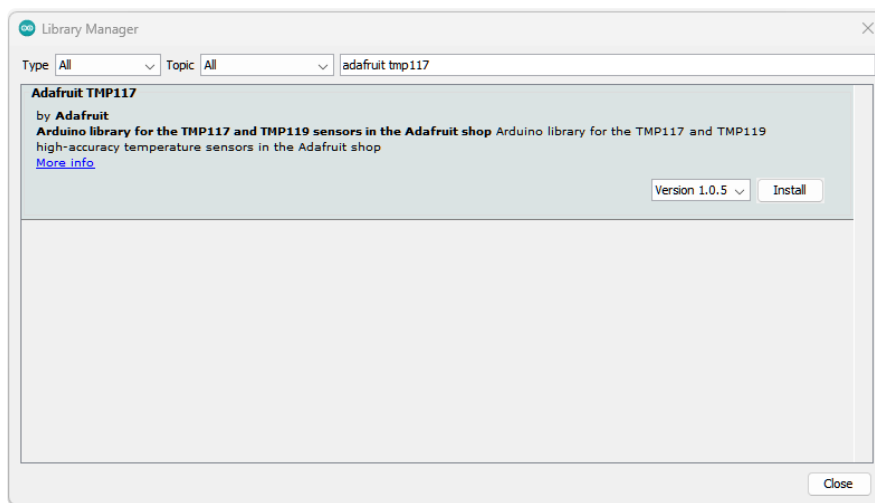
- Board 5V to breakout VIN (red wire)
- Board GND to breakout GND (black wire)
- Board SCL to breakout SCL (yellow wire)
- Board SDA to breakout SDA (blue wire)

Library Installation

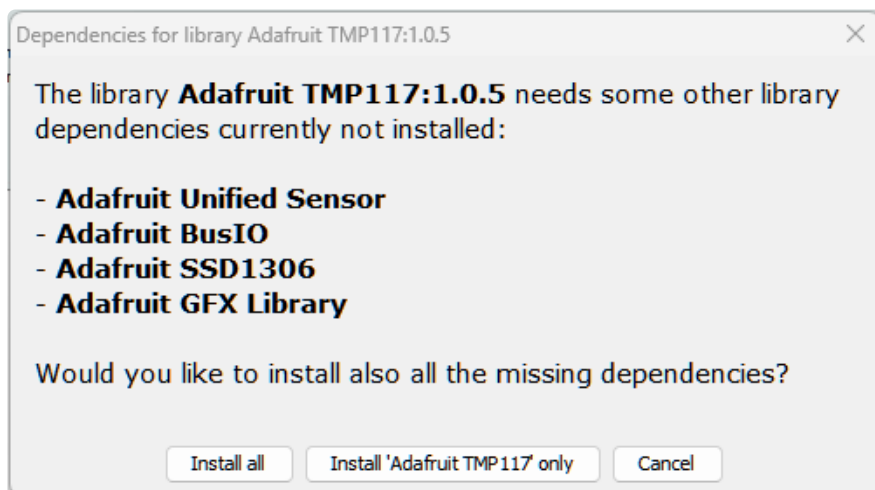
You can install the **Adafruit TMP117** library for Arduino using the Library Manager in the Arduino IDE.



Click the **Manage Libraries ...** menu item, search for **Adafruit TMP117**, and select the **Adafruit TMP117** 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

```
/**
 * @file TMP119_basic_test.ino
 * @brief Basic test sketch for the TMP117/TMP119 temperature sensor
 * @date 2025-02-25
 *
 * @copyright Copyright (c) 2025
 */
#include <Adafruit_Sensor.h>
#include <Adafruit_TMP117.h>
#include <Adafruit_TMP119.h>

// Adafruit_TMP117 tmp11x;
Adafruit_TMP119 tmp11x;

// To use with TMP117 instead, uncomment the TMP117 include/line above
// and comment out the TMP119 include/line

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens
  Serial.println(F("Adafruit TMP117/TMP119 test!"));

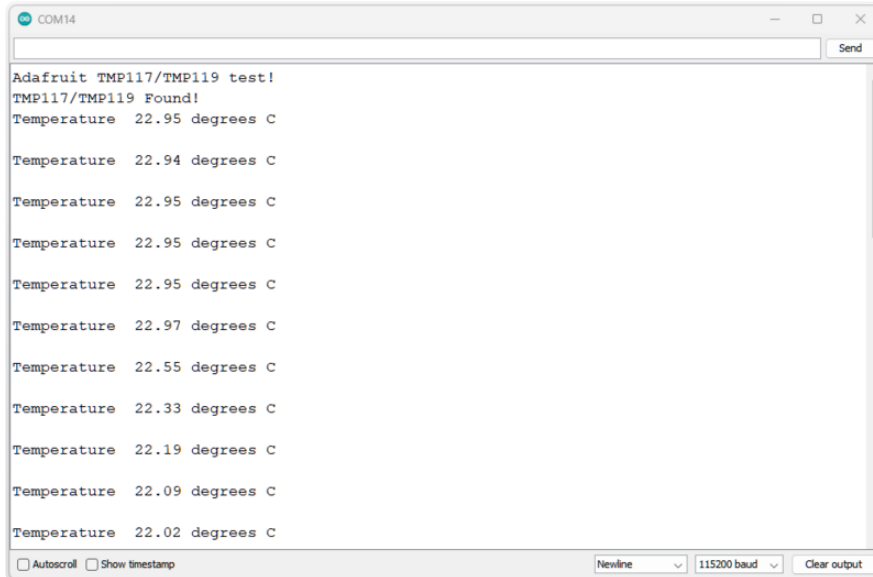
  // Try to initialize!
  if (!tmp11x.begin()) {
    Serial.println(F("Failed to find TMP117/TMP119 chip"));
    while (1) {
      delay(10);
    }
  }
  Serial.println(F("TMP117/TMP119 Found!"));
}

void loop() {
  // Wait for fresh data before reading
  while (!tmp11x.dataReady()) {
    delay(10);
  }

  sensors_event_t temp; // create an empty event to be filled
  tmp11x.getEvent(
    &temp); // fill the empty event object with the current measurements
  Serial.print(F("Temperature "));
  Serial.print(temp.temperature);
  Serial.println(F(" degrees C"));
  Serial.println();

  delay(1000);
}
```

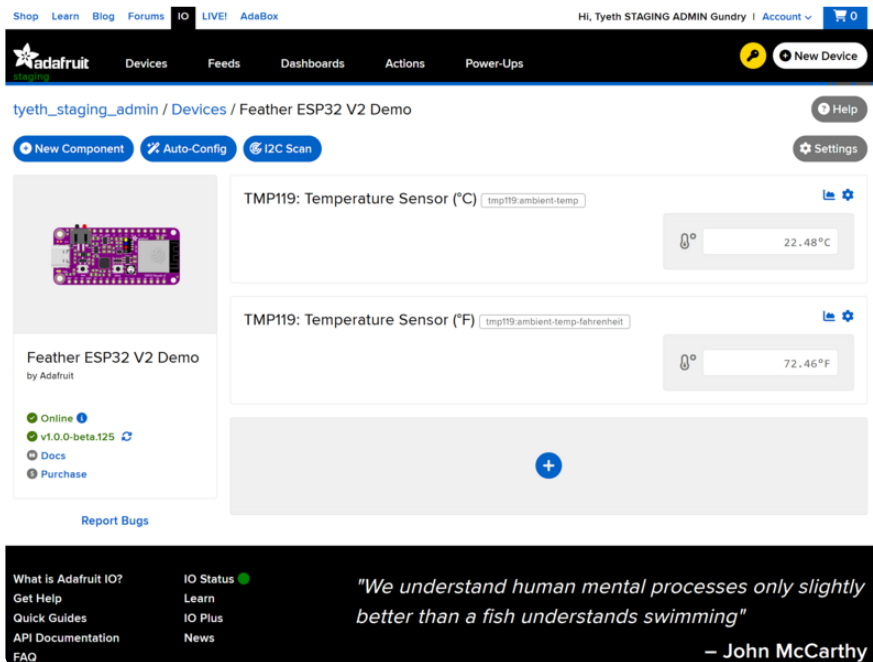
Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. You should see that the sketch has found your connected TMP119 sensor. Then, you'll see the temperature readings in Celsius printed to the Serial Monitor every second.



Arduino Docs

[Arduino Docs \(https://adafru.it/1aAZ\)](https://adafru.it/1aAZ)

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.

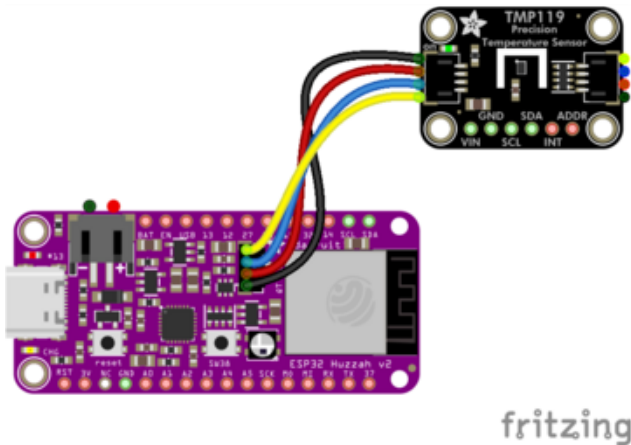
From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware, so you can immediately start interacting, logging, and streaming the data your projects produce without writing code.

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

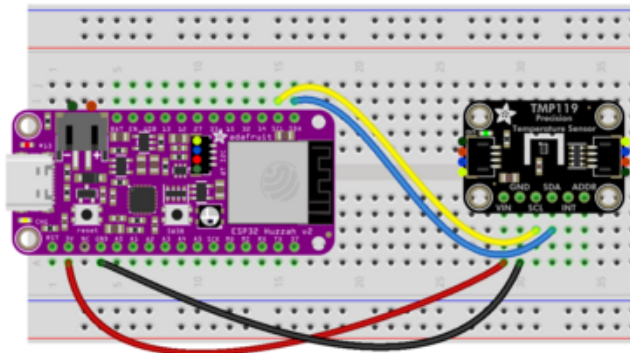
<https://adafru.it/Vfd>

Wiring

First, wire up a TMP119 to your board exactly as follows. Here is an example of the TMP119 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>)



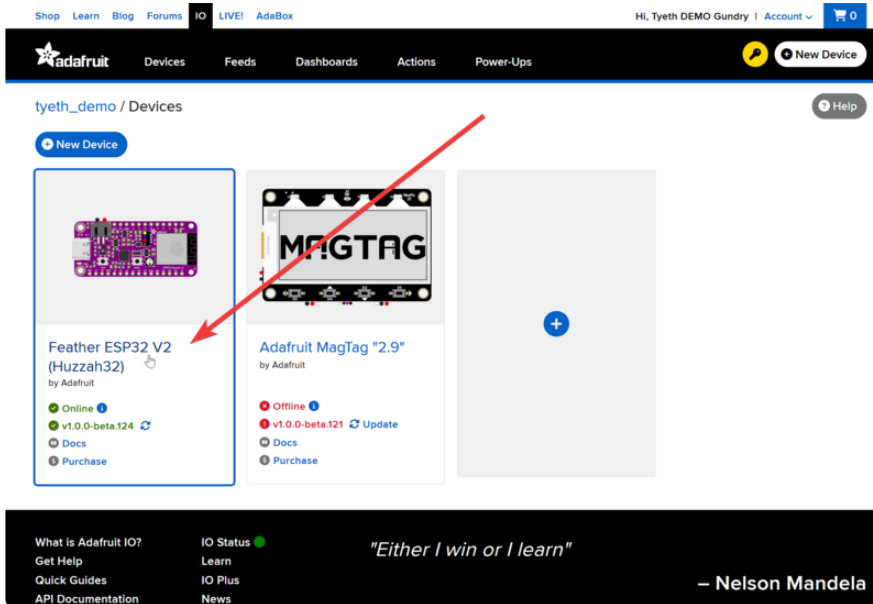
- Board 3V to breakout VIN (red wire)
- Board GND to breakout GND (black wire)
- Board SCL to breakout SCL (yellow wire)
- Board SDA to breakout SDA (blue wire)



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.

Feather ESP32 V2 (Huzzah32)

by Adafruit

- ✓ Online ⓘ
- ✓ v1.0.0-beta.124 ↻
- 📖 Docs
- 💰 Purchase

Adafruit MagTag "2.9"

by Adafruit

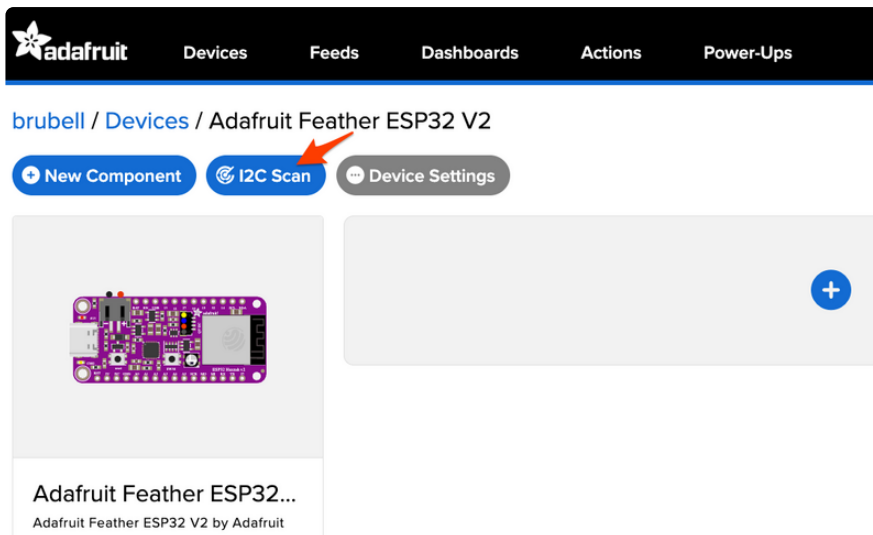
- ✓ Online ⓘ
- ! v1.0.0-beta.122 ↻ Update
- 📖 Docs
- 💰 Purchase

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.



You should see the TMP119's default I2C address of `0x48` 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	--	--	--	--	--	--	--	--	48	--	--	--	--	--	--	--
50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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 **TMP119** into the search bar, then select the **TMP119** component.

New Component



Which component would you like to set up?

X TMP119

Displaying 1 matching Components.

ambient-temp
ambient-temp-fahrenheit

i2c

TMP119
This little sensor contains temperature sensing capabilities.

[Product Page](#)

[Documentation](#)

Cancel

On the component configuration page, the TMP119'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 TMP119 sensor and send the data to Adafruit IO. Measurements can range from every second to every 24 hours.

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

Create TMP119 Component



Select I2C Address

0x48

Enable TMP119: Temperature Sensor (°C)?

Name:

TMP119: Temperature Sensor (°C)

Send Data:

Every 30 seconds

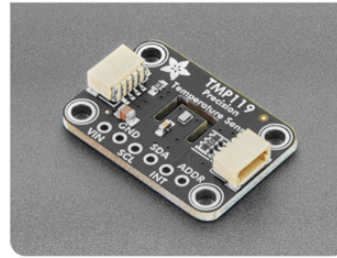
Enable TMP119: Temperature Sensor (°F)?

Name:

TMP119: Temperature Sensor (°F)

Send Data:

Every 30 seconds



[← 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.

The screenshot shows the Adafruit IO dashboard for a device named 'tyeth_demo / Devices / Adafruit Feather ESP32 V2'. Two temperature sensors have been added:

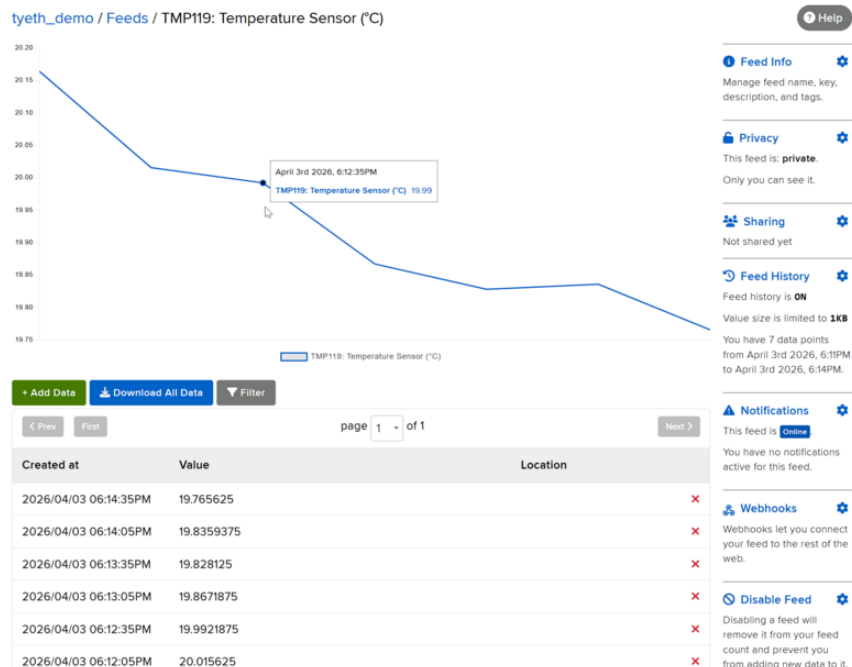
- TMP119: Temperature Sensor (°C)**: Reading 20.16°C
- TMP119: Temperature Sensor (°F)**: Reading 68.30°F

The dashboard includes a sidebar with device information (Adafruit Feather ESP32 V2, Online, v1.0.0-beta.124) and a footer with navigation links and the quote: "Forgive many things in others; nothing in yourself" – Ausonius.

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



Downloads

Files

- [TMP119 Datasheet \(https://adafru.it/1aB5\)](https://adafru.it/1aB5)
- [EagleCAD PCB files on GitHub \(https://adafru.it/1aB6\)](https://adafru.it/1aB6)
- [Fritzing object in the Adafruit Fritzing Library \(https://adafru.it/1aB7\)](https://adafru.it/1aB7)

Schematic and Fab Print

