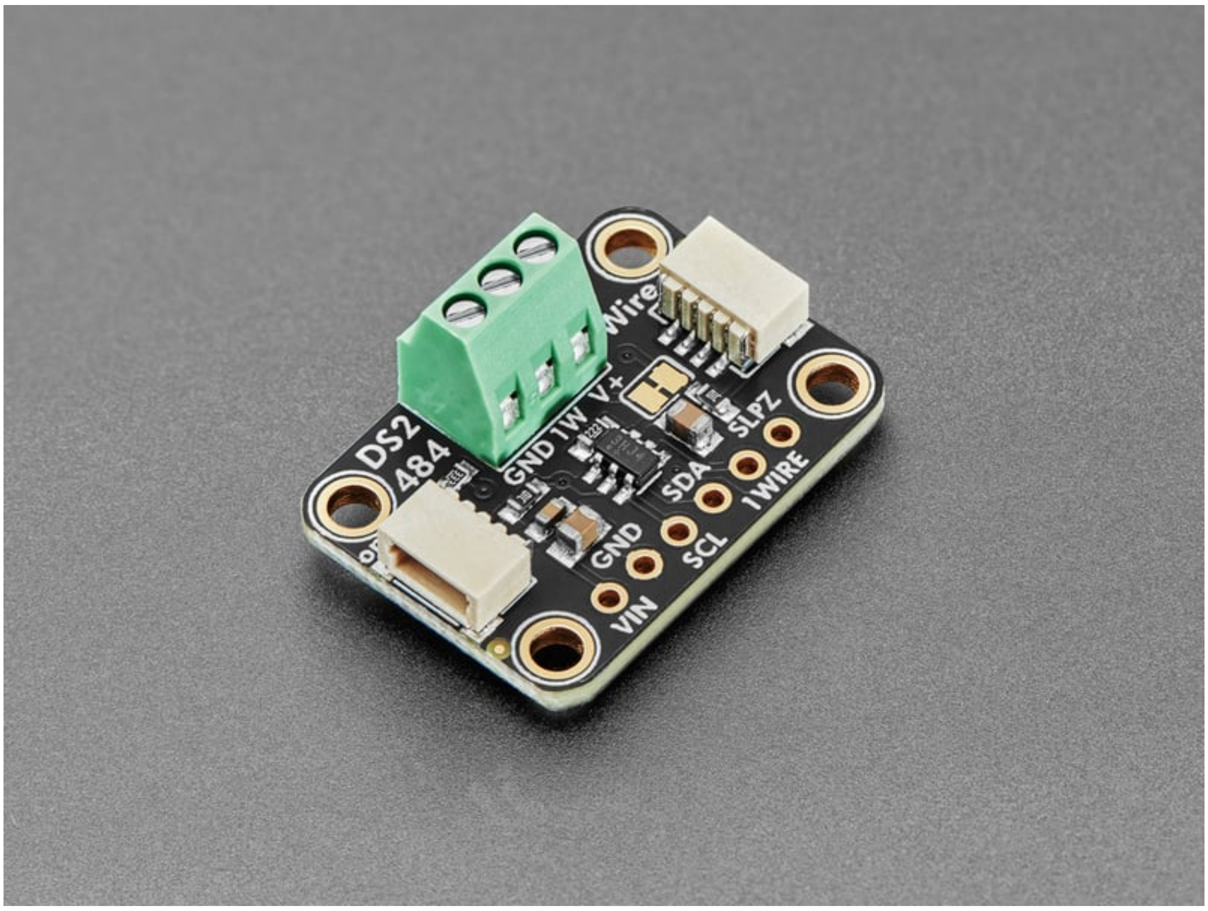
Adafruit DS2484 I2C to 1-Wire Bus Adapter Breakout

Created by Liz Clark

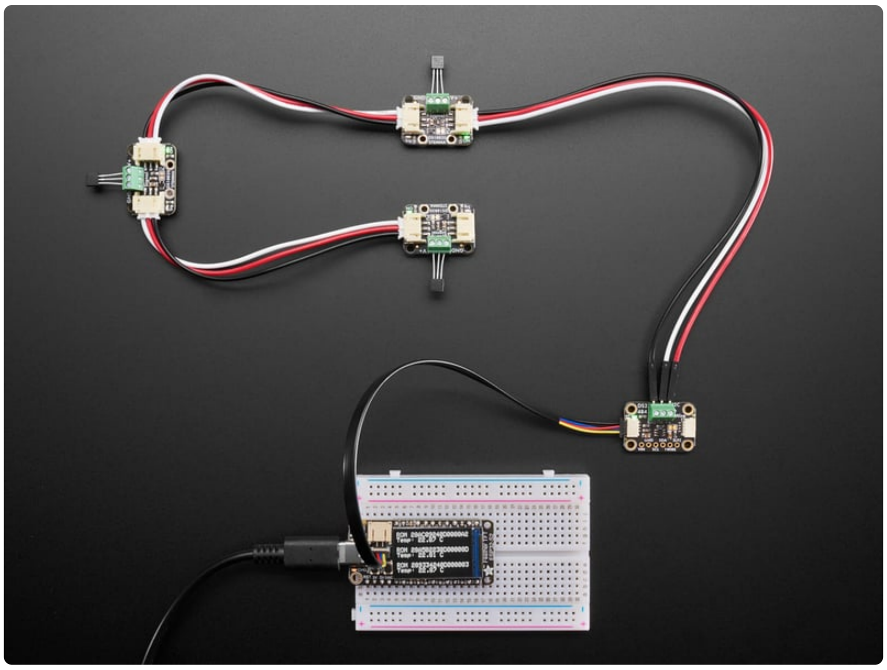


https://learn.adafruit.com/adafruit-ds2484-i2c-to-1-wire-bus-adapter-breakout

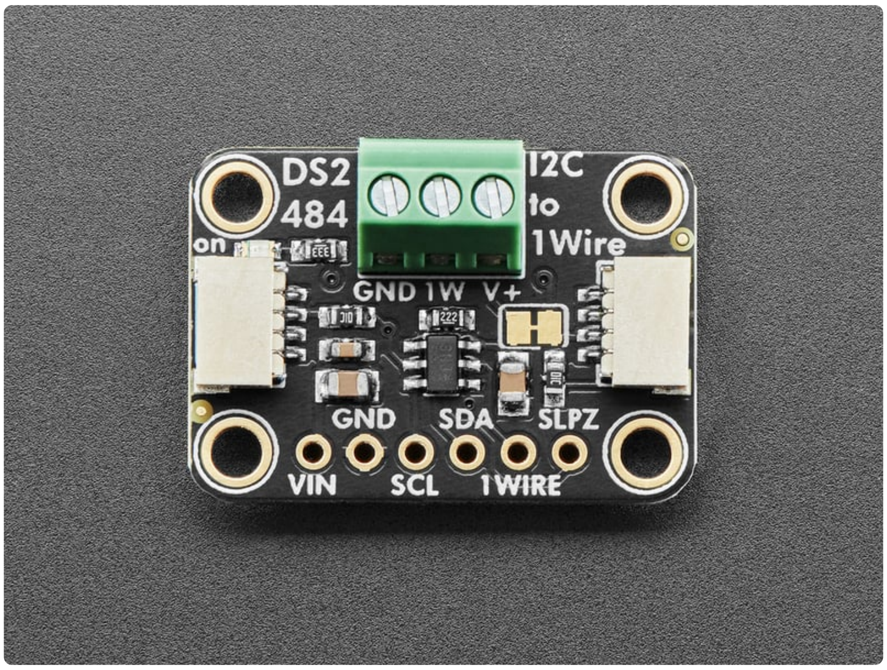
Last updated on 2024-07-23 11:32:26 AM EDT

Table of Contents

Overview

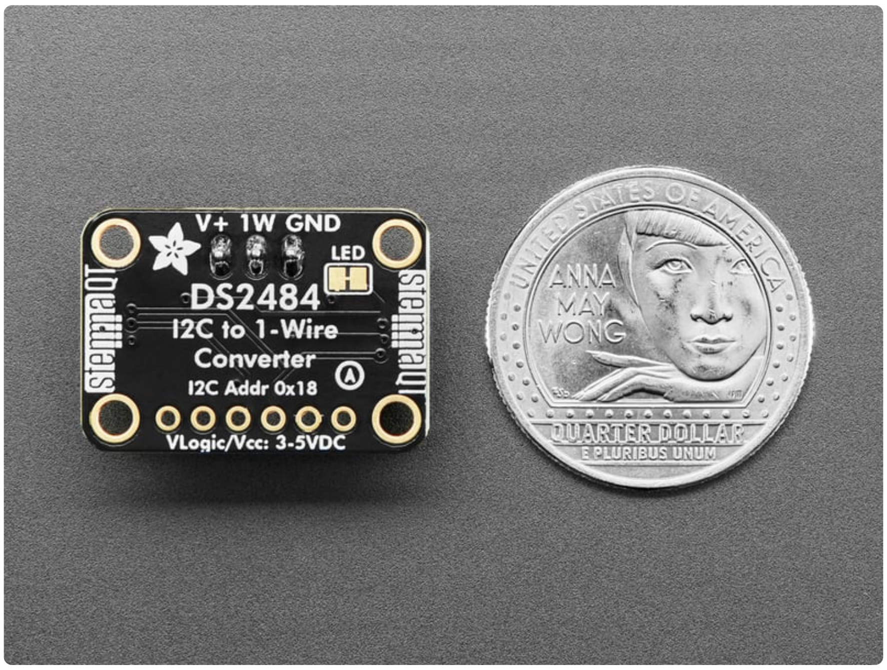


By customer request, this is a DS2484 (https://adafru.it/1a2Y) Stemma QT board that uses a I2C-to-1-Wire controller chip, with electrostatic (ESD) protection and support for split supplies. You can easily connect it to an existing I2C bus and then use the screw terminals to attach multiple DS18B20 temperature sensors, or pair it with our 1-Wire chaining breakouts (http://adafru.it/5971) for fancier experimentation.



You're probably familiar with the 'top three' electronics protocols: I2C, SPI and UART. Perhaps you're aware of a fourth one, called "1-Wire" which was invented by Dallas Semiconductor (https://adafru.it/1a2Z) (which became Maxim, which became Analog

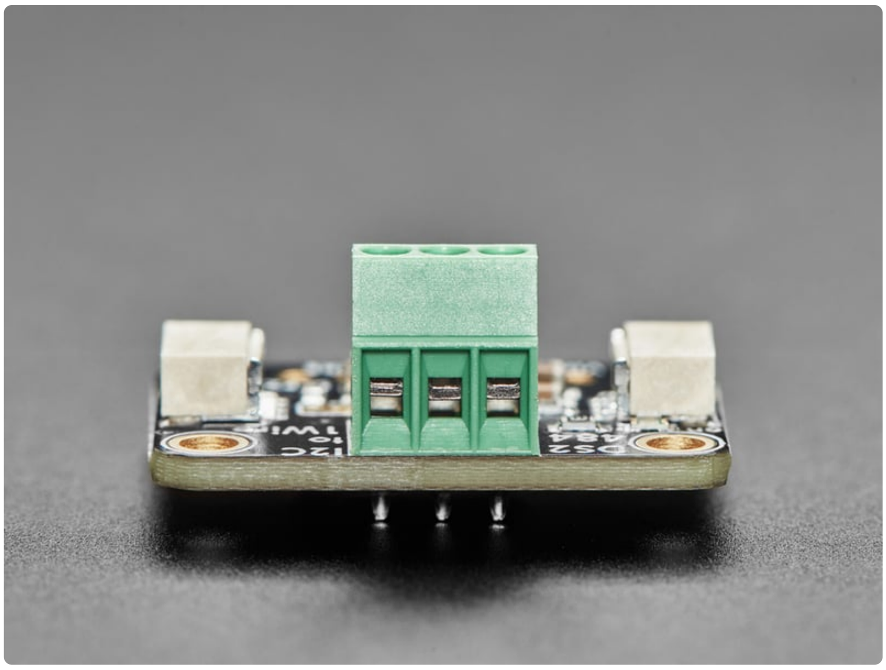
Devices). As you may expect, this protocol uses a single data wire plus a ground wire (and an optional power wire) to connect to any number of sensors or memory chips that all share the same bus.



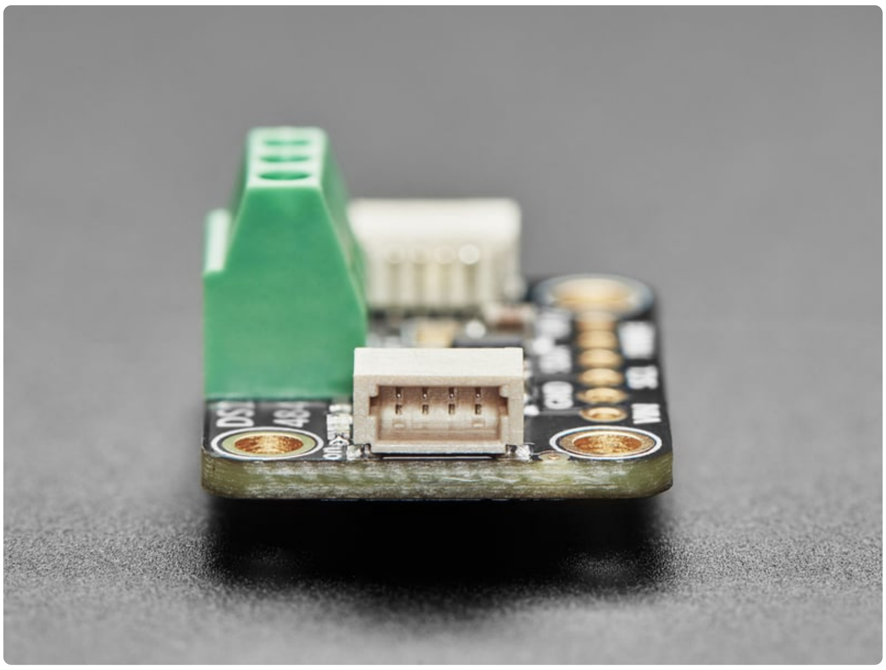
In theory, there's a lot of different 1-Wire devices out there, but in reality almost everyone uses 1-Wire for DS18B20 temperature sensors.

The long wire lengths and ease of 'chaining' by sharing a single bus wire makes it perfectly fine for this purpose.

You can bit-bang a 1-Wire interface on most microcontrollers, and some SBCs like Raspberry Pi have kernel module support (https://adafru.it/1a30). But there are lots of chips or firmware stacks without 1-Wire capability, or maybe you want to use 1-Wire devices on your desktop computer or other SBC with I2C.

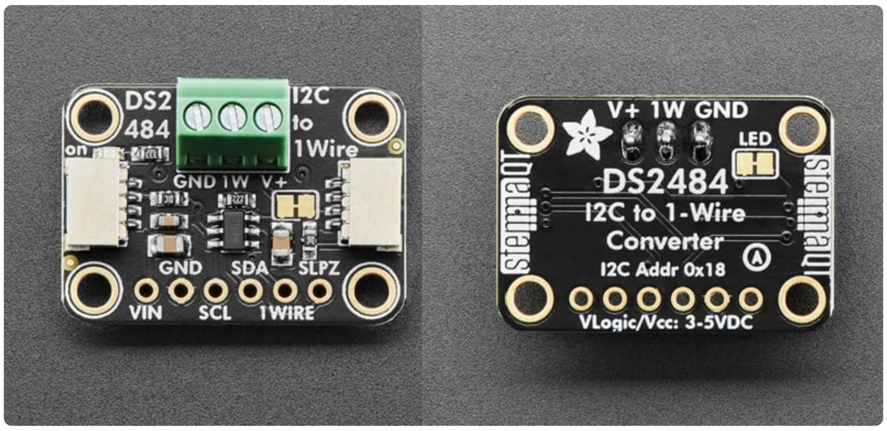


To get you going fast, we spun up a custom-made PCB in the **STEMMA QT** form factor (https://adafru.it/LBQ), making it easy to interface with. The STEMMA QT connectors (https://adafru.it/JqB) on either side are compatible with the SparkFun Qwiic (https://adafru.it/Fpw) I2C connectors. This allows you to make solderless connections between your development board and the DS2484 or to chain it with a wide range of other sensors and accessories using a **compatible cable** (https://adafru.it/JnB).



You can power the board directly from the STEMMA QT cable, whether it is 3V or 5V. If you want a different voltage on the 1-Wire power pin, cut the onboard trace and you can inject the voltage at the terminal block.

Pinouts



The default I2C address is **0x18**.

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.
- **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 to dev boards with **STEMMA QT** (Qwiic) connectors or to other things with various associated accessories (https://adafru.it/Ft6).

1-Wire Data Line

- **1WIRE** - the signal connection for the 1-wire data line

1-Wire Terminal Block

At the top edge of the board is a 1x3 terminal block. This block lets you connect a 1-wire sensor to the breakout board.

- **V+** - the power connection for the 1-wire sensor. It shares the same voltage level as **VIN**.
- **1W** - the signal connection for the 1-wire line
- **GND** - the ground connection for the 1-wire sensor

Sleep Mode Pin

- **SLPZ** - this is the sleep mode pin. This pin can be set low to put the DS2484 into power-saving sleep mode.

1-Wire Power Jumper

- **V+** - On the front of the board, below the terminal block, is the 1-Wire power pin jumper. It is outlined in white on the board silk. Cut this jumper if you want a different voltage on the 1-Wire power pin. Then, you can inject the voltage at the terminal block.

Power LED and Jumper

- **Power LED** - in the top left corner on the front of the board is the power LED, labeled **ON**. It is the green LED.
- **LED jumper** - on the back of the board is a jumper for the power LED. It is labeled **LED** on the board silk. If you want to disable the power LED, cut the trace on this jumper.

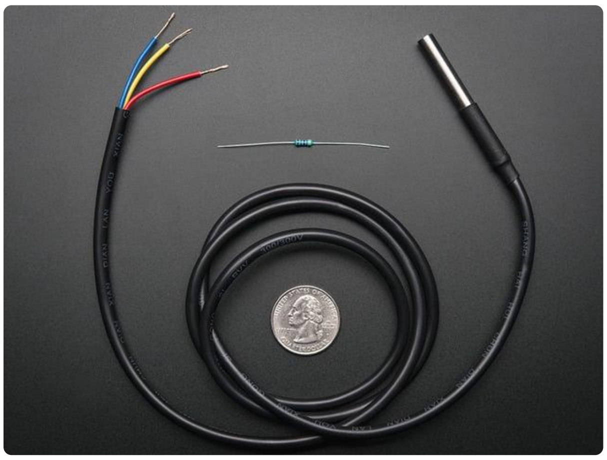
CircuitPython and Python

It's easy to use the **DS2484** with Python or CircuitPython, and the Adafruit_CircuitPython_DS248x (https://adafru.it/1a31) module. This module allows you to easily write Python code to read 1-Wire sensors over I2C.

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).

You'll need a DS18B20 sensor to use this example with the breakout:

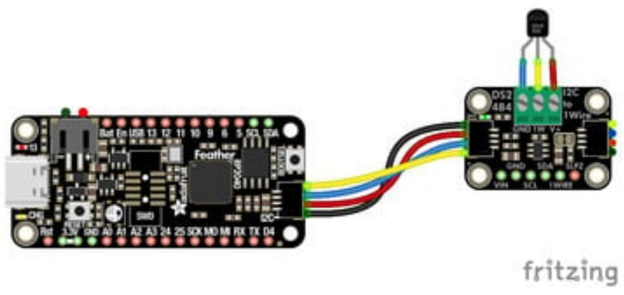
DS18B20 Digital temperature sensor + extras
These 1-wire digital temperature sensors are fairly precise (±0.5°C over much of the range) and can give up to 12 bits of precision from the onboard digital-to-analog...
https://www.adafruit.com/product/374



Waterproof 1-Wire DS18B20 Digital temperature sensor
This is a pre-wired and waterproofed (with heat shrink) version of a 1 Wire DS18B20 sensor. Handy for when you need to measure something far away, or in wet conditions. While the...
https://www.adafruit.com/product/381

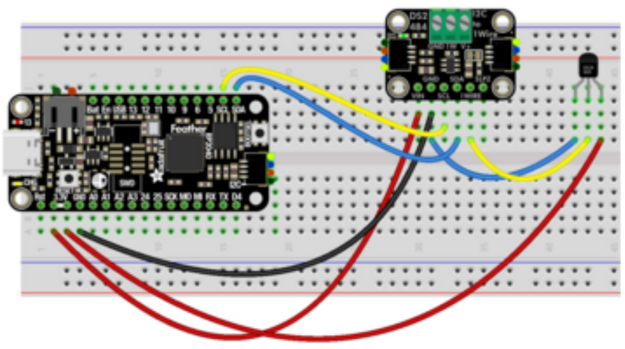
CircuitPython Microcontroller Wiring

First, wire up the DS18B20 sensor to the breakout exactly as follows. Then, wire up the breakout to your board. The following is the breakout wired to a Feather RP2040 and DS18B20 sensor using the STEMMA connector and terminal block:



Breakout terminal block GND to **DS18B20 ground (blue wire)**
Breakout terminal block 1W to **DS18B20 1-wire output (yellow wire)**
Breakout terminal block V+ to **DS18B20 VIN (red wire)**
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)**

The following is the breakout wired to a Feather RP2040 and DS18B20 using a solderless breadboard:

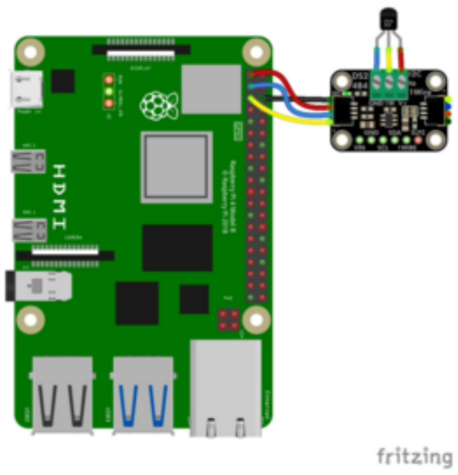


Breakout GND to **DS18B20 ground (blue wire)**
Breakout 1WIRE to **DS18B20 1-wire output (yellow wire)**
Board 3V to **DS18B20 VIN (red wire)**
Board 3V to **breakout VIN (red wire)**
Board GND to **breakout GND (black wire)**
Board SCL to **breakout SCL (yellow wire)**
Board SDA to **breakout SDA (blue wire)**

Python Computer Wiring

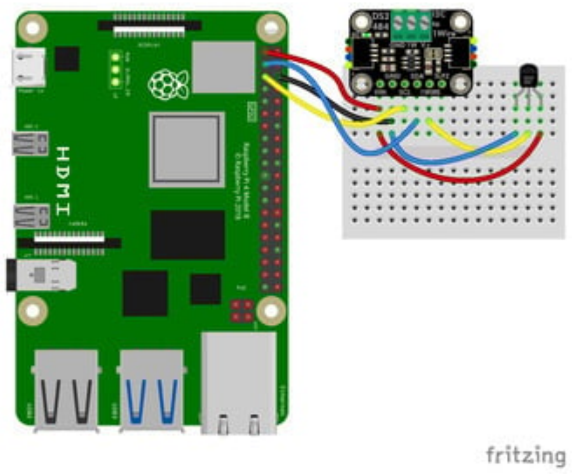
Since there are dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported](https://adafru.it/BSN) (https://adafru.it/BSN).

Here's the Raspberry Pi wired with I2C using the STEMMA connector and terminal block for the DS18B20:



Breakout terminal block GND to **DS18B20 ground (blue wire)**
Breakout terminal block 1W to **DS18B20 1-wire output (yellow wire)**
Breakout terminal block V+ to **DS18B20 VIN (red wire)**
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)**

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



Breakout GND to **DS18B20 ground (blue wire)**
Breakout 1WIRE to **DS18B20 1-wire output (yellow wire)**
Pi 3V to **DS18B20 VIN (red wire)**
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 DS248x 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-ds248x`

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!

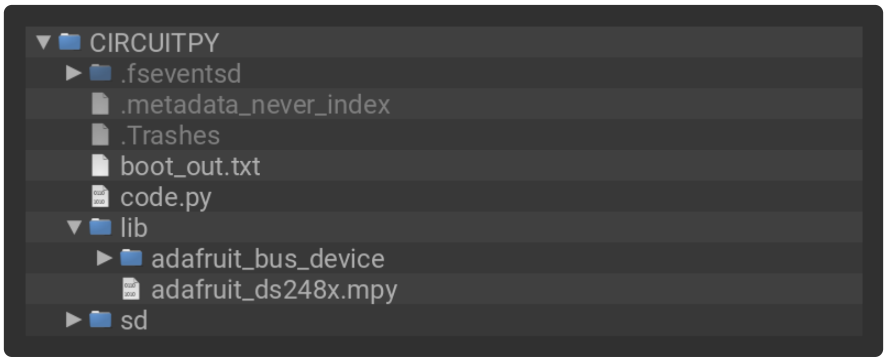
CircuitPython Usage

To use with CircuitPython, you need to first install the **Adafruit_CircuitPython_DS248x** 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 folder and file:

- **adafruit_bus_device/**
- **adafruit_ds248x.mpy**



```
▼ CIRCUITPY
  ▶ .fseventsd
    .metadata_never_index
    .Trashes
    boot_out.txt
    code.py
  ▼ lib
    ▶ adafruit_bus_device
      adafruit_ds248x.mpy
  ▶ sd
```

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`

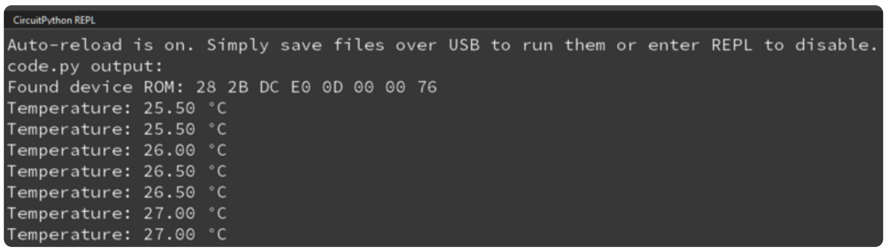
Example Code

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

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

```
# SPDX-FileCopyrightText: Copyright (c) 2024 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""Adafruit DS248x DS18B20 Example"""

import time
import board
from adafruit_ds248x import Adafruit_DS248x

# Initialize I2C bus and DS248x
i2c = board.I2C()
ds248x = Adafruit_DS248x(i2c)

rom = bytearray(8)
if not ds248x.onewire_search(rom):
    print("No more devices found\n\n")

print("Found device ROM: ", end="")
for byte in rom:
    print(f"{byte:02X} ", end="")
print()
while True:
    temperature = ds248x.ds18b20_temperature(rom)
    print(f"Temperature: {temperature:.2f} °C")

    time.sleep(1)
```

First, the DS2484 is instantiated over I2C and checks for a connected DS18B20 sensor. In the loop, the DS18B20 sensor temperature readings are printed to the serial console.



```
Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
code.py output:
Found device ROM: 28 2B DC E0 0D 00 00 76
Temperature: 25.50 °C
Temperature: 25.50 °C
Temperature: 26.00 °C
Temperature: 26.50 °C
Temperature: 26.50 °C
Temperature: 27.00 °C
Temperature: 27.00 °C
```

Python Docs

Python Docs (https://adafru.it/1a2U)

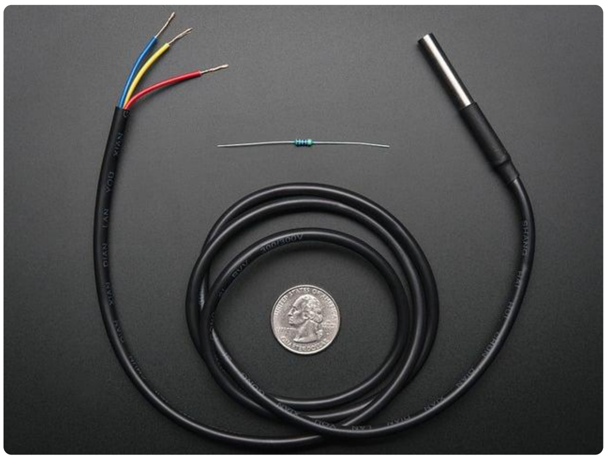
Arduino

Using the DS2484 breakout with Arduino involves wiring up the breakout to your Arduino-compatible microcontroller with a DS18B20 sensor, installing the Adafruit_DS248x (https://adafru.it/1a32) library, and running the provided example code.

DS18B20 Digital temperature sensor + extras

These 1-wire digital temperature sensors are fairly precise (±0.5°C over much of the range) and can give up to 12 bits of precision from the onboard digital-to-analog...

https://www.adafruit.com/product/374



Waterproof 1-Wire DS18B20 Digital temperature sensor

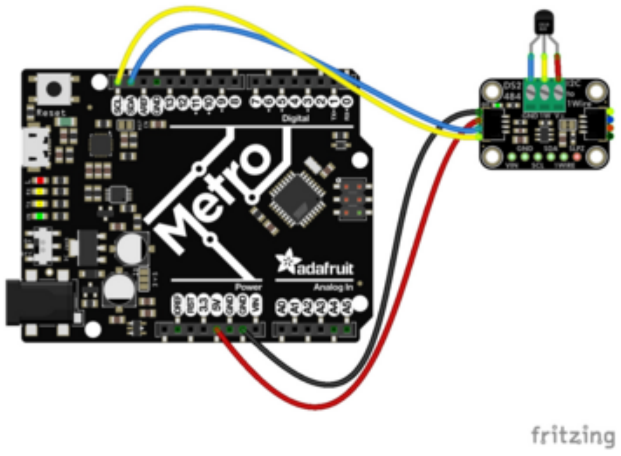
This is a pre-wired and waterproofed (with heat shrink) version of a 1 Wire DS18B20 sensor. Handy for when you need to measure something far away, or in wet conditions. While the...

https://www.adafruit.com/product/381

Wiring

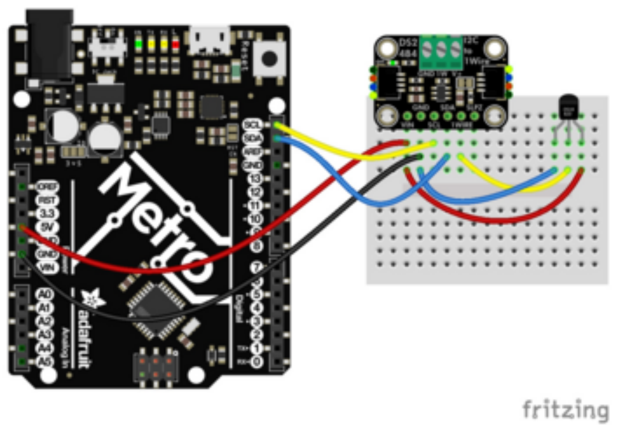
Wire as shown for a **5V** board like an Uno. If you are using a **3V** board, like an Adafruit Feather, wire the board's 3V pin to the breakout VIN.

Here is an Adafruit Metro wired up to the breakout with a DS18B20 sensor using the STEMMA QT connector:



Breakout terminal block GND to **DS18B20 ground (blue wire)**
Breakout terminal block 1W to **DS18B20 1-wire output (yellow wire)**
Breakout terminal block V+ to **DS18B20 VIN (red wire)**
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)**

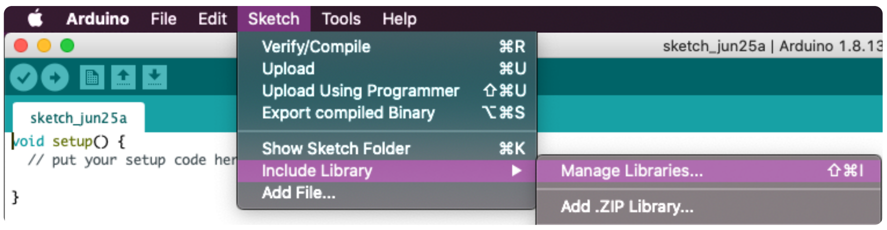
Here is an Adafruit Metro wired up using a solderless breadboard:



Breakout GND to **DS18B20 ground (blue wire)**
Breakout 1WIRE to **DS18B20 1-wire output (yellow wire)**
Board 5V to **DS18B20 VIN (red wire)**
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)**

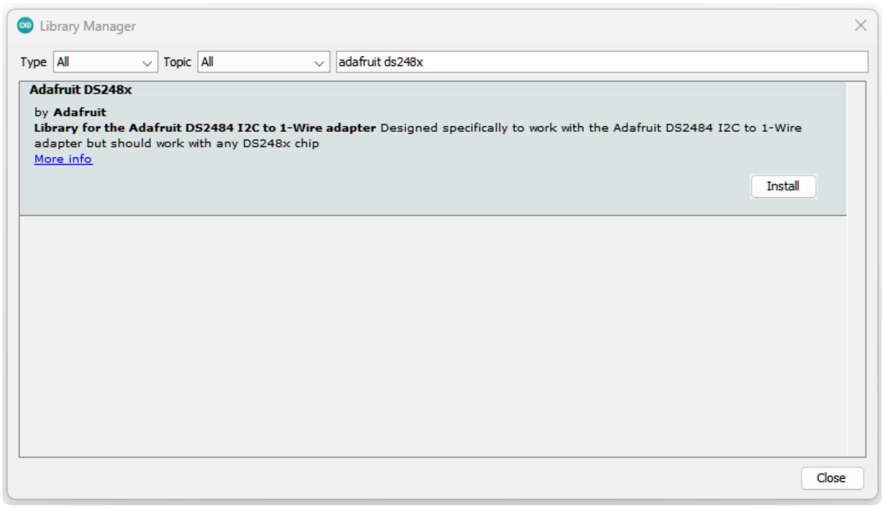
Library Installation

You can install the **Adafruit_DS248x** library for Arduino using the Library Manager in the Arduino IDE.



```
void setup() {
  // put your setup code her
}
```

Click the **Manage Libraries ...** menu item, search for **Adafruit_DS248x**, and select the **Adafruit DS248x** library:


Library Manager

Type All Topic All adafruit ds248x

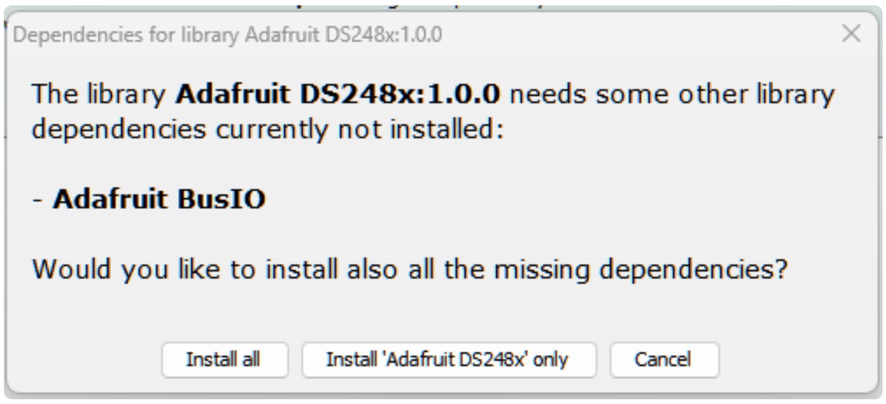
Adafruit DS248x

by **Adafruit**

Library for the Adafruit DS2484 I2C to 1-Wire adapter Designed specifically to work with the Adafruit DS2484 I2C to 1-Wire adapter but should work with any DS248x chip

More info

If asked about dependencies, click "Install all".


Dependencies for library Adafruit DS248x:1.0.0

The library **Adafruit DS248x:1.0.0** needs some other library dependencies currently not installed:

- **Adafruit BusIO**

Would you like to install also all the missing dependencies?

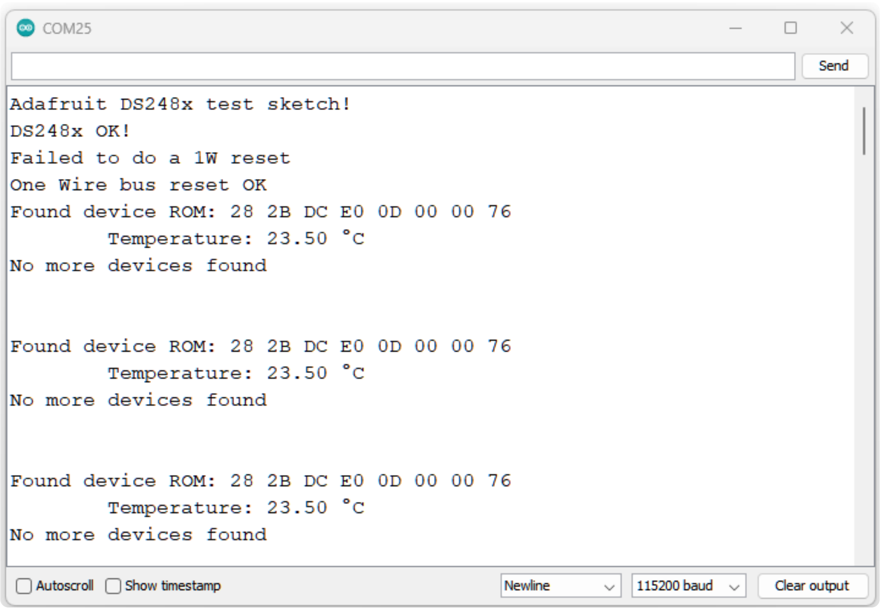
If the "Dependencies" window does not come up, then you already have the dependencies installed.

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 "Adafruit_DS248x.h"

#define DS18B20_FAMILY_CODE 0x28
#define DS18B20_CMD_CONVERT_T 0x44
#define DS18B20_CMD_MATCH_ROM 0x55
#define DS18B20_CMD_READ_SCRATCHPAD 0xBE

Adafruit_DS248x ds248x;

void setup() {
  Serial.begin(115200);
  while (!Serial) delay(10);
  Serial.println("Adafruit DS248x test sketch!");
  if (!ds248x.begin(&Wire, DS248X_ADDRESS)) {
    Serial.println(F("DS248x initialization failed."));
    while (1);
  }

  Serial.println("DS248x OK!");

  // Speed up I2C, as searching for ROMs, specifically, is slow!
  Wire.setClock(400000);

  while (! ds248x.OneWireReset()) {
    Serial.println("Failed to do a 1W reset");
    if (ds248x.shortDetected()) {
      Serial.println("\tShort detected");
    }
    if (!ds248x.presencePulseDetected()) {
      Serial.println("\tNo presense pulse");
    }
    delay(1000);
  }
  Serial.println("One Wire bus reset OK");
}


void loop() {
  uint8_t rom[8];

  if (!ds248x.OneWireSearch(rom)) {
    Serial.println("No more devices found\n\n");
    return;
  }

  Serial.print("Found device ROM: ");
  for (int i = 0; i < 8; i++) {
    if (rom[i] < 16) {
      Serial.print("0");
    }
    Serial.print(rom[i], HEX);
    Serial.print(" ");
  }
  Serial.println();

  // Check if the device is a DS18B20 (Family code 0x28)
  if (rom[0] == DS18B20_FAMILY_CODE) {
      // Read and print temperature
      float temperature = readTemperature(rom);
      Serial.print("\tTemperature: ");
      Serial.print(temperature);
      Serial.println(" °C");
  }
}

float readTemperature(uint8_t *rom) {
    // Select the DS18B20 device
    ds248x.OneWireReset();
    ds248x.OneWireWriteByte(DS18B20_CMD_MATCH_ROM); // Match ROM command
    for (int i = 0; i < 8; i++) {
        ds248x.OneWireWriteByte(rom[i]);
    }

    // Start temperature conversion
    ds248x.OneWireWriteByte(DS18B20_CMD_CONVERT_T); // Convert T command
    delay(750); // Wait for conversion (750ms for maximum precision)

    // Read scratchpad
    ds248x.OneWireReset();
    ds248x.OneWireWriteByte(DS18B20_CMD_MATCH_ROM); // Match ROM command
    for (int i = 0; i < 8; i++) {
        ds248x.OneWireWriteByte(rom[i]);
    }
    ds248x.OneWireWriteByte(DS18B20_CMD_READ_SCRATCHPAD); // Read Scratchpad command

    uint8_t data[9];
    for (int i = 0; i < 9; i++) {
        ds248x.OneWireReadByte(&data[i]);
    }

    // Calculate temperature
    int16_t raw = (data[1] << 8) | data[0];
    float celsius = (float)raw / 16.0;

    return celsius;
}
```

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. You'll see the DS2484 recognized over I2C. In the loop, a search is performed for an attached DS18B20 temperature sensor. When the sensor is found, its temperature data is printed to the Serial Monitor.

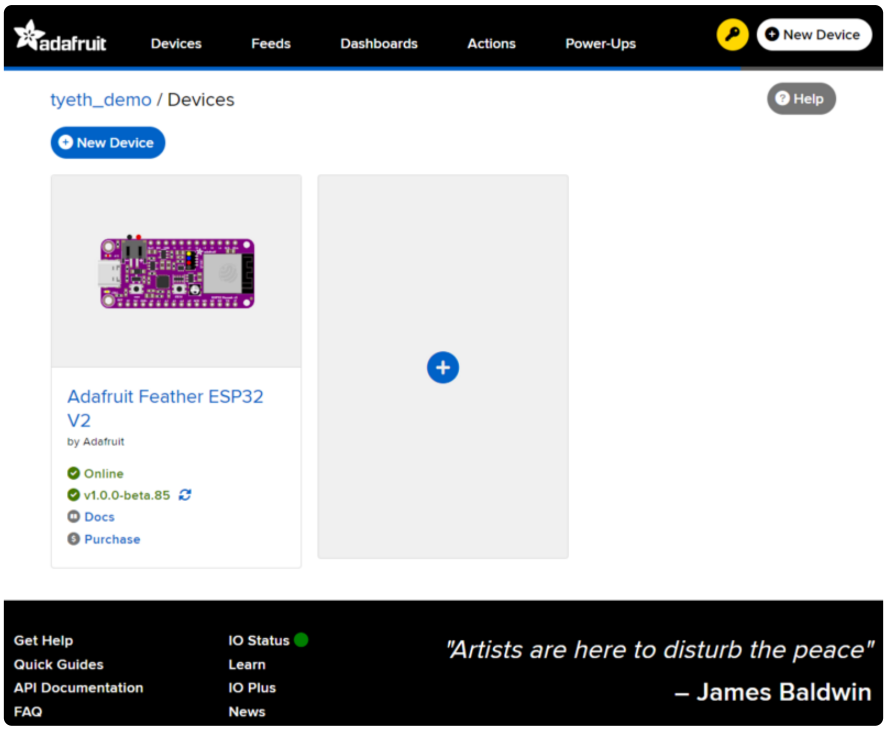


```
Adafruit DS248x test sketch!
DS248x OK!
Failed to do a 1W reset
One Wire bus reset OK
Found device ROM: 28 2B DC E0 0D 00 00 76
        Temperature: 23.50 °C
No more devices found


Found device ROM: 28 2B DC E0 0D 00 00 76
        Temperature: 23.50 °C
No more devices found


Found device ROM: 28 2B DC E0 0D 00 00 76
        Temperature: 23.50 °C
No more devices found
```

Arduino Docs

[Arduino Docs](https://adafru.it/1a2M) (https://adafru.it/1a2M)

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

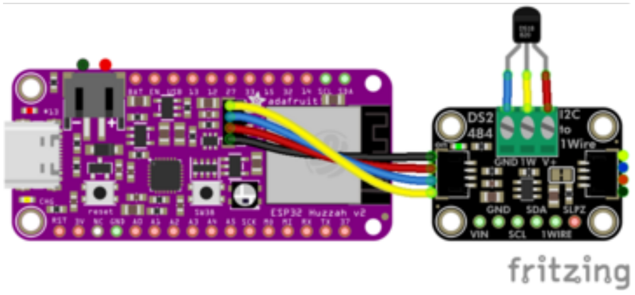
Using the DS2484 breakout with WipperSnapper involves wiring up the breakout to your WipperSnapper-compatible microcontroller using jumper cables, or a StemmaQT cable, and connecting a DS18B20 temperature sensor to the One-Wire bus (screw terminals or breakout pads).

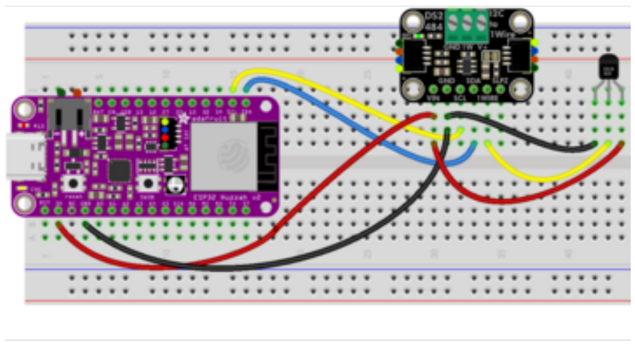
You'll need a DS18B20 sensor to use with the breakout, grab one here if you need:

1 x DS18B20 Temperature sensor (Through-hole) Either one of these	https://www.adafruit.com/product/374
1 x Wired Waterproof DS18B20 sensor or one of these	https://www.adafruit.com/product/381
1 x Wired High temp (PTFE) Waterproof DS18B20 or one of these	https://www.adafruit.com/product/642
1 x Extra-long (3m) High temp Waterproof DS18B20 or this extra long one	https://www.adafruit.com/product/3846

First, wire up a DS2484 to your board exactly as follows. Here is an example of the DS2484 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). The DS18B20 temperature sensor has been connected to the 1-Wire bus using the screw terminals.

Check the product page or datasheet (https://adafru.it/1a4x) to confirm the pin order for your DS18B20 sensor.



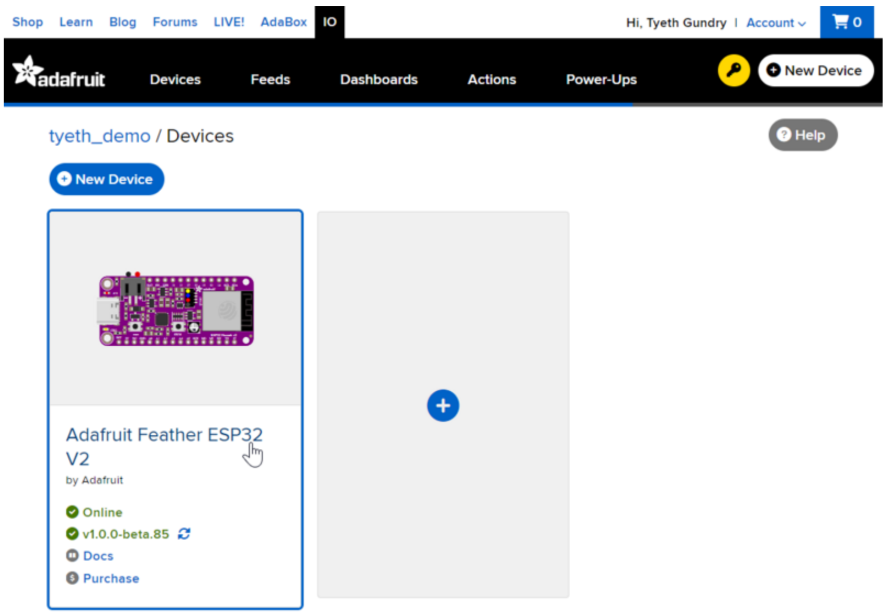


Breakout terminal block GND to **DS18B20 ground (blue wire)**
Breakout terminal block 1W to **DS18B20 1-wire output (yellow wire)**
Breakout terminal block V+ to **DS18B20 VIN (red wire)**
Board 3V to **sensor VIN (red wire on STEMMA QT)**
Board GND to **sensor GND (black wire on STEMMA QT)**
Board SCL to **sensor SCK (yellow wire on STEMMA QT)**
Board SDA to **sensor SDI (blue wire on STEMMA QT)**

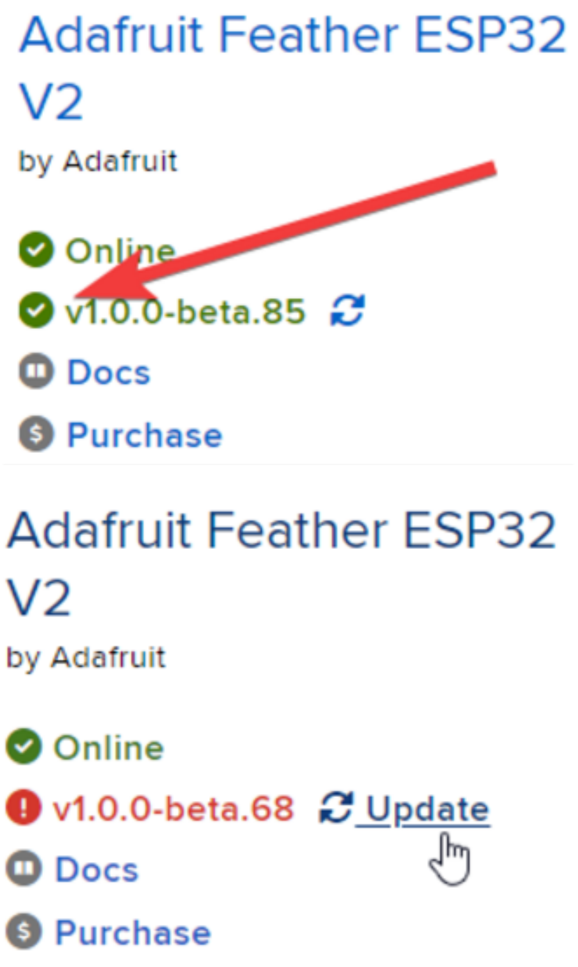
Usage

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

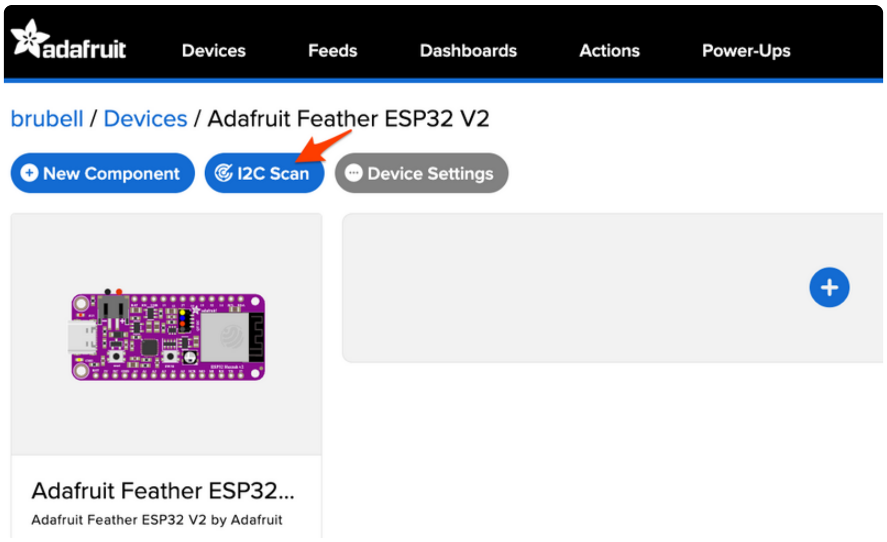


On the device page, quickly **check that you're running the latest version of the WipperSnapper firmware**.

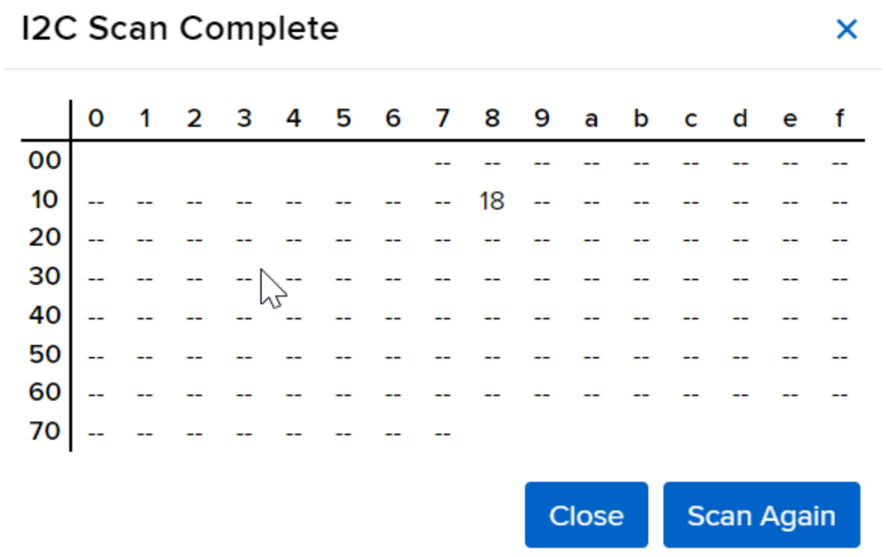
The device tile on the left indicates the version number of the firmware running on the connected board.

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.


adafruit
Devices
Feeds
Dashboards
Actions
Power-Ups
brubell / Devices / Adafruit Feather ESP32 V2
New Component
I2C Scan
Device Settings
Adafruit Feather ESP32...
Adafruit Feather ESP32 V2 by Adafruit

You should see the DS2484's default I2C address of `0x18` 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	--	--	--	--	--	--	--	--	18	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
40	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--								

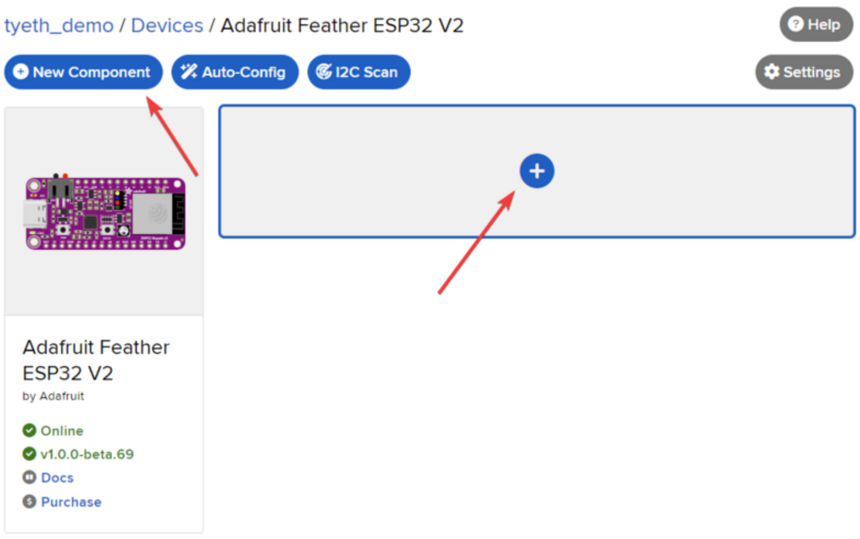
I don't see the sensor's I2C address listed!

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

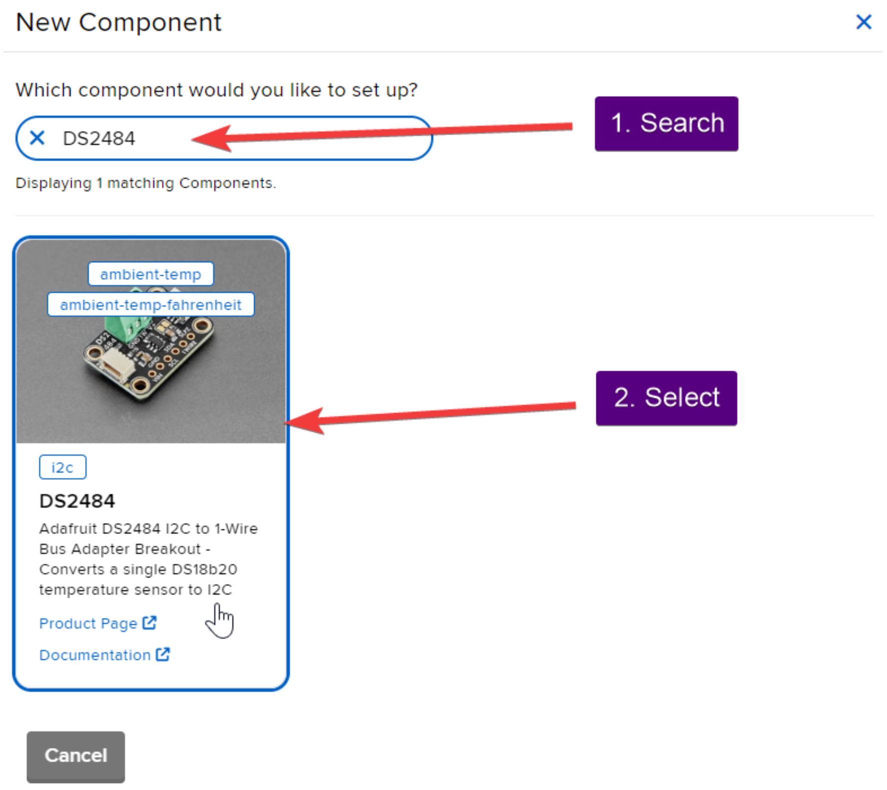
Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

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.



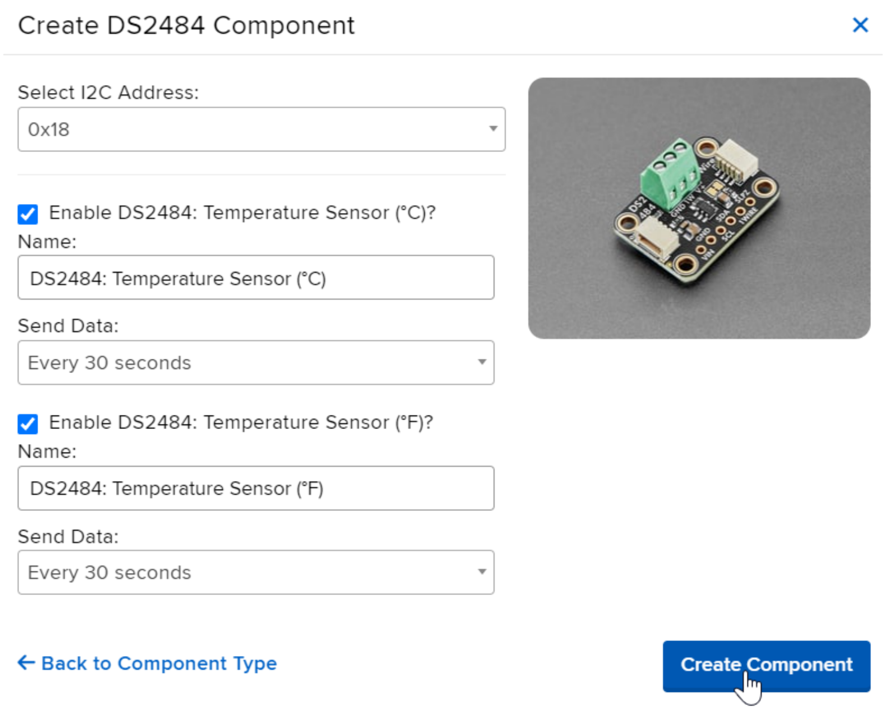
Adafruit IO supports a large amount of components. To quickly find your sensor, type `DS2484` into the search bar, then select the **DS2484** component.


New Component
Which component would you like to set up?
DS2484
1. Search
Displaying 1 matching Components.
ambient-temp
ambient-temp-fahrenheit
2. Select
i2c
DS2484
Adafruit DS2484 I2C to 1-Wire Bus Adapter Breakout - Converts a single DS18b20 temperature sensor to I2C
Product Page
Documentation
Cancel

On the component configuration page, the DS2484's I2C address should be listed along with the component'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 DS2484 component 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.

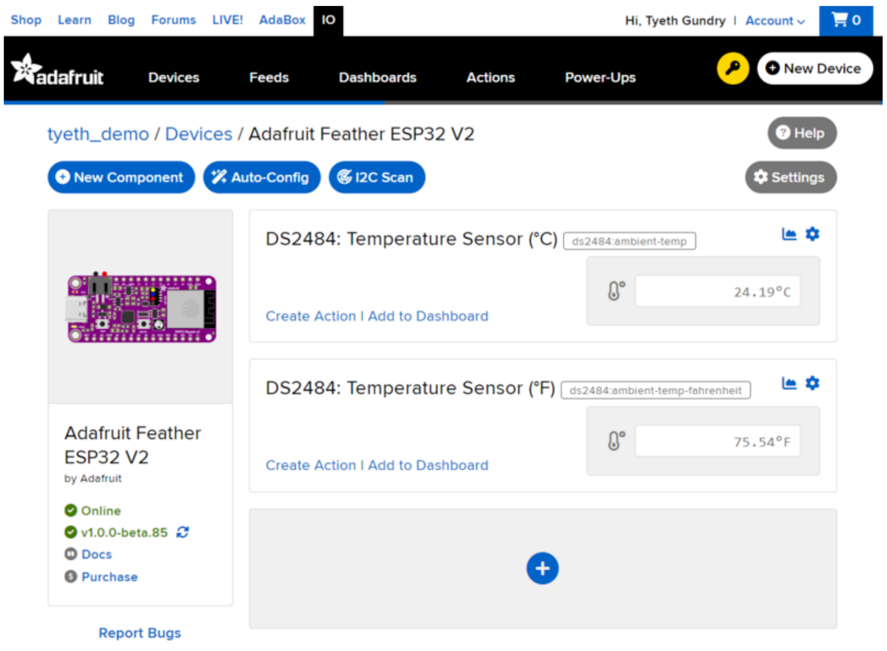

Create DS2484 Component

Select I2C Address: 0x18

[✓] Enable DS2484: Temperature Sensor (°C)?
Name: DS2484: Temperature Sensor (°C)
Send Data: Every 30 seconds

[✓] Enable DS2484: Temperature Sensor (°F)?
Name: DS2484: Temperature Sensor (°F)
Send Data: 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.



tyeth_demo / Devices / Adafruit Feather ESP32 V2

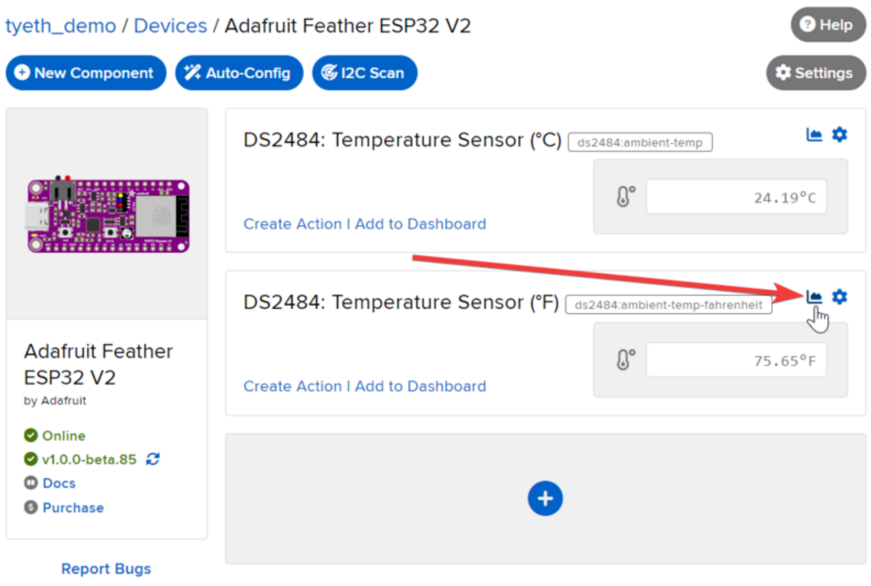
DS2484: Temperature Sensor (°C) ds2484:ambient-temp — 24.19°C
Create Action | Add to Dashboard

DS2484: Temperature Sensor (°F) ds2484:ambient-temp-fahrenheit — 75.54°F
Create Action | Add to Dashboard

Adafruit Feather ESP32 V2
by Adafruit
Online
v1.0.0-beta.85
Docs
Purchase

Report Bugs

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

DS2484: Temperature Sensor (°C) ds2484:ambient-temp

24.19°C

Create Action | Add to Dashboard

DS2484: Temperature Sensor (°F) ds2484:ambient-temp-fahrenheit

75.65°F

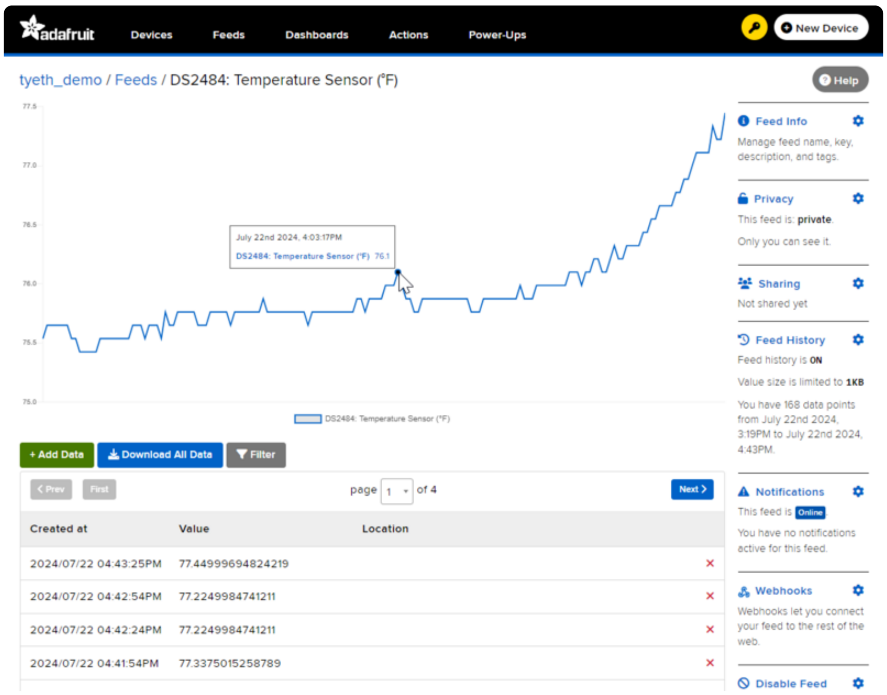
Create Action | Add to Dashboard

Adafruit Feather ESP32 V2
by Adafruit

Online
v1.0.0-beta.85
Docs
Purchase

Report Bugs

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).



tyeth_demo / Feeds / DS2484: Temperature Sensor (°F)

page 1 of 4

Created at	Value	Location
2024/07/22 04:43:25PM	77.44999694824219	
2024/07/22 04:42:54PM	77.2249984741211	
2024/07/22 04:42:24PM	77.2249984741211	
2024/07/22 04:41:54PM	77.3375015258789	

Feed Info
Manage feed name, key, description, and tags.

Privacy
This feed is: **private**.
Only you can see it.

Sharing
Not shared yet

Feed History
Feed history is **ON**
Value size is limited to **1KB**
You have 168 data points from July 22nd 2024, 3:19PM to July 22nd 2024, 4:43PM.

Notifications
This feed is Online.
You have no notifications active for this feed.

Webhooks
Webhooks let you connect your feed to the rest of the web.

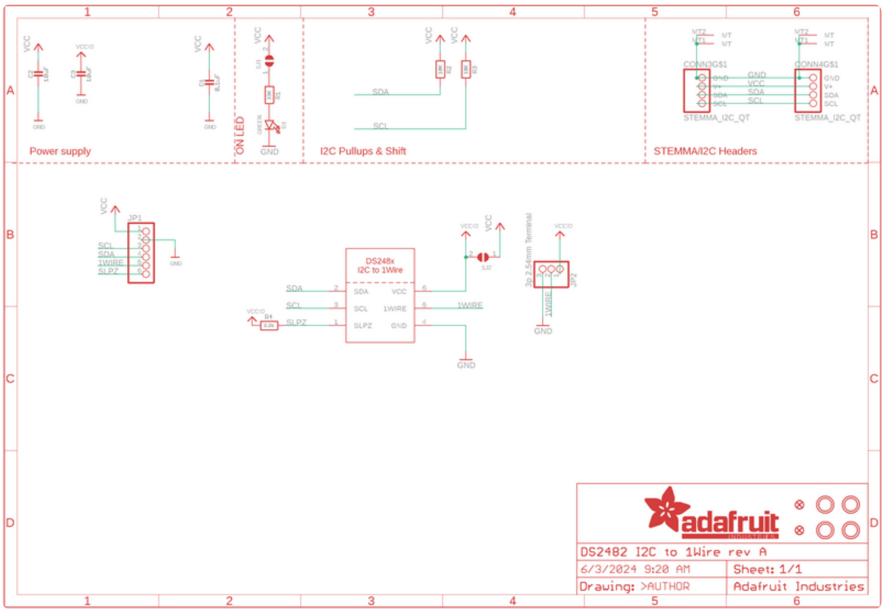
Disable Feed

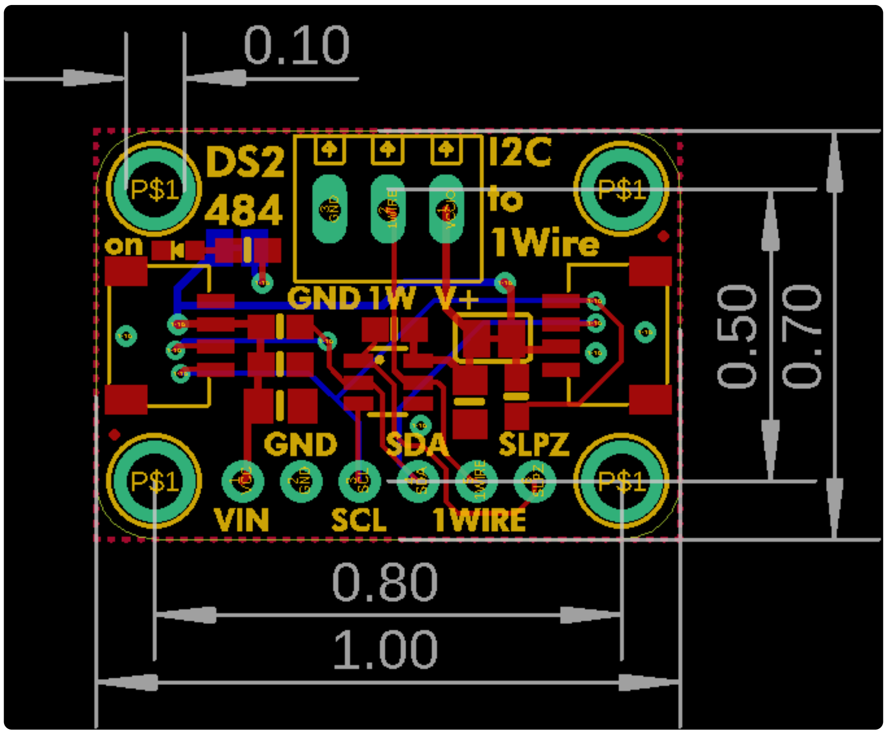
Downloads

Files

- DS2484 Datasheet (https://adafru.it/1a33)
- EagleCAD PCB Files on GitHub (https://adafru.it/1a34)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/1a35)

Schematic and Fab Print


Power supply
ON LED
I2C Pullups & Shift
STEMMA/I2C Headers
DS248x I2C to 1Wire
DS2482 I2C to 1Wire rev A
6/3/2024 9:20 AM
Sheet: 1/1
Drawing: >AUTHOR
Adafruit Industries


0.10
DS2 484
I2C to 1Wire
GND 1W V+
GND SDA SLPZ
VIN SCL 1WIRE
0.50
0.70
0.80
1.00