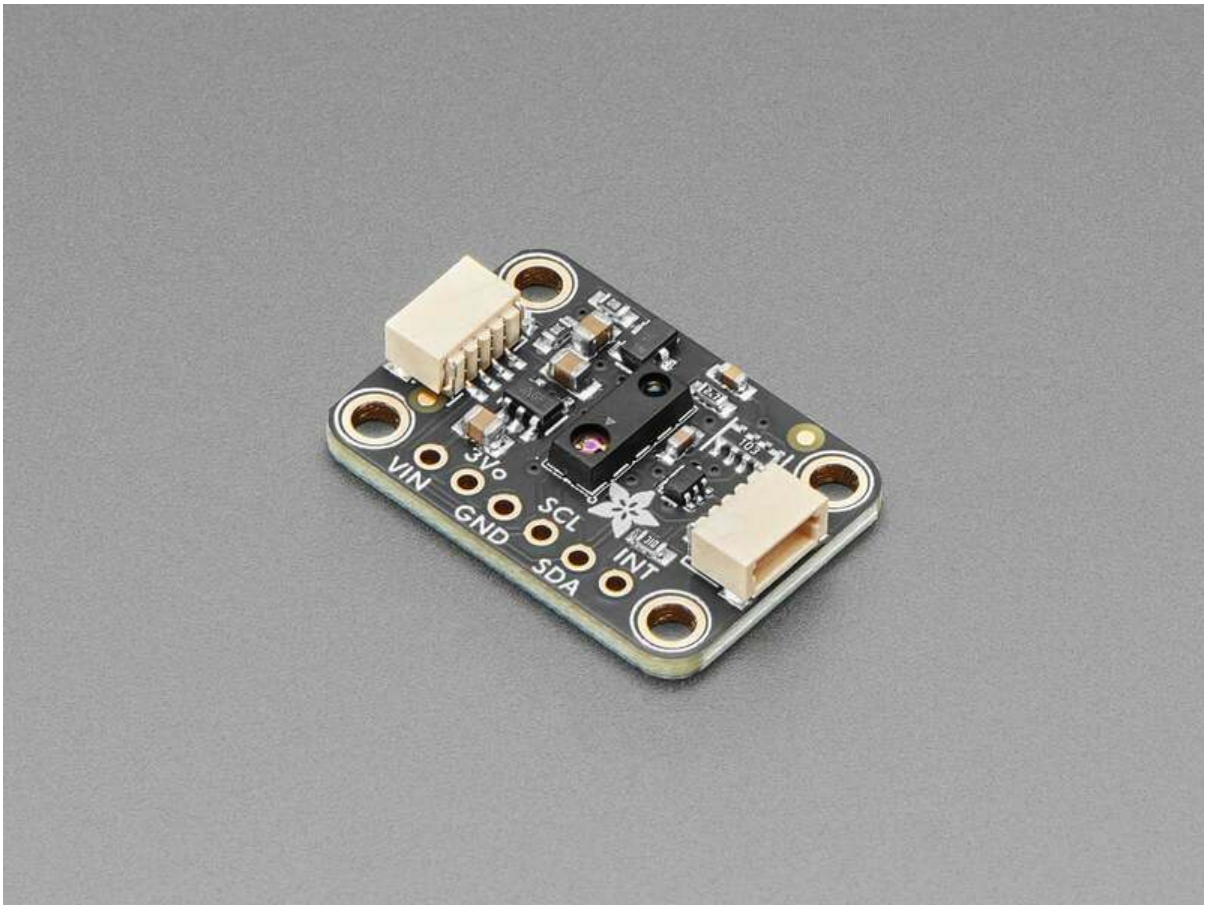
Adafruit VCNL4200 Long Distance IR Proximity and Light Sensor

Created by Liz Clark

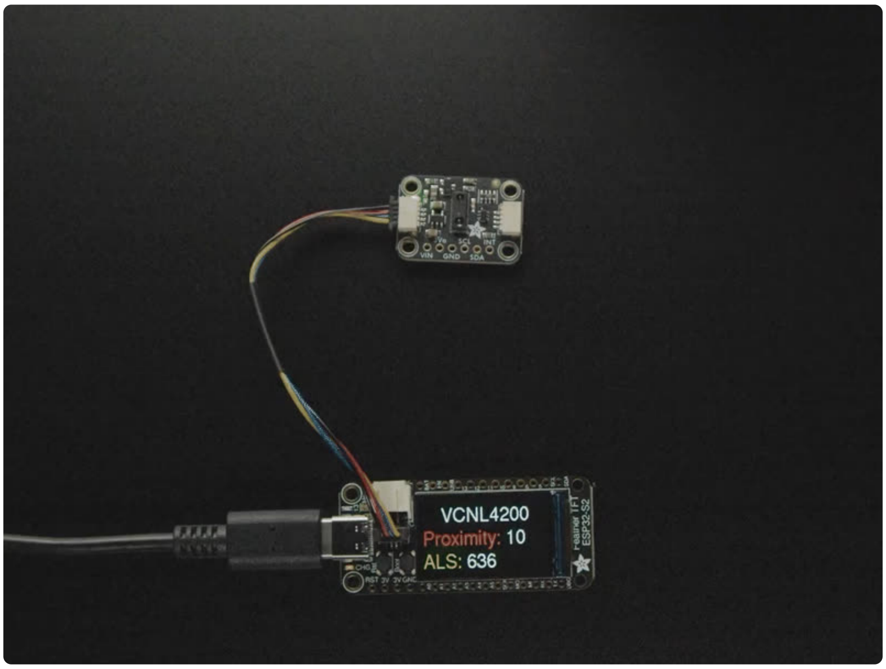


https://learn.adafruit.com/adafruit-vcnl4200-long-distance-ir-proximity-and-light-sensor

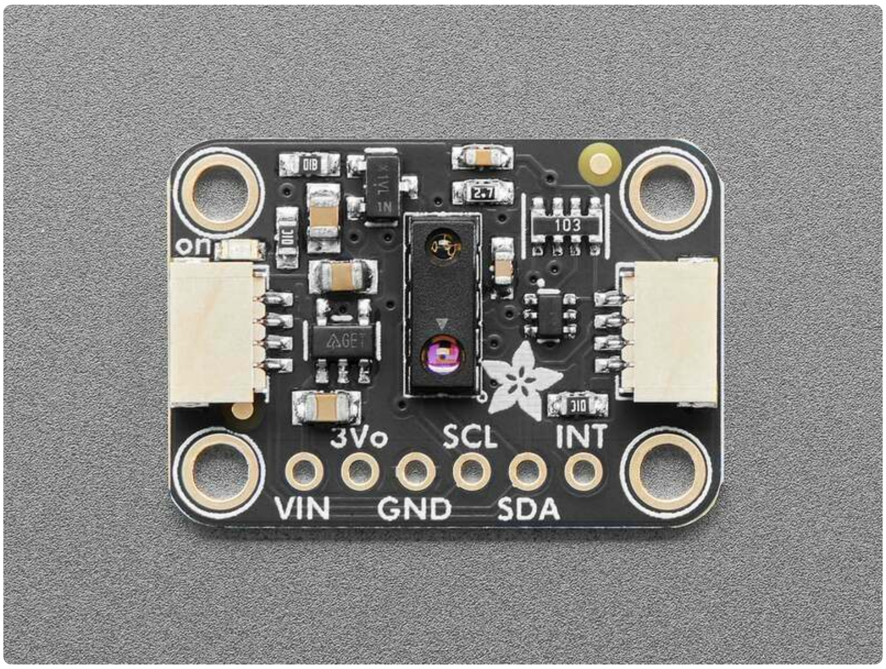
Last updated on 2025-04-18 10:06:16 AM EDT

Table of Contents

Overview

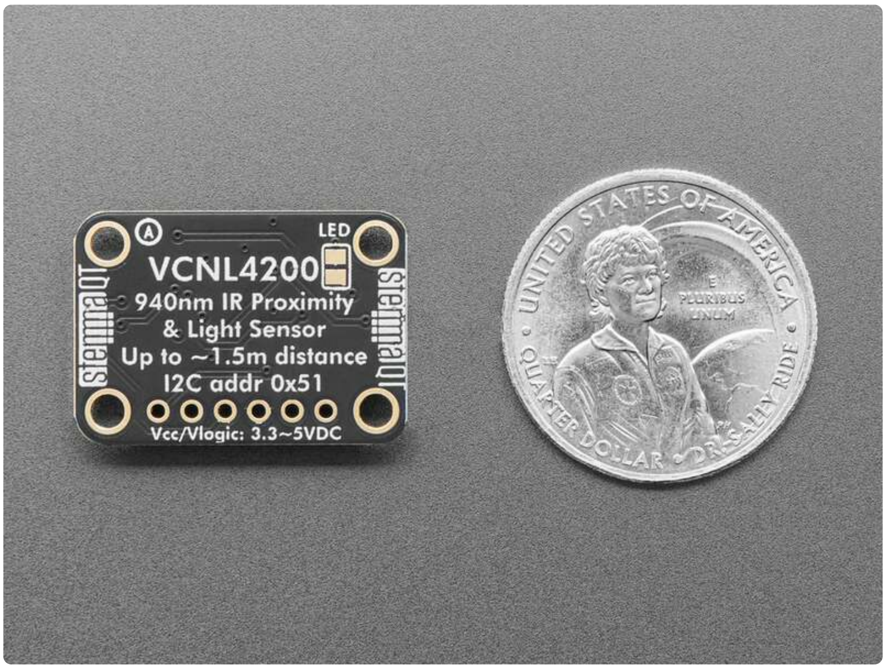


We've all been there. That thing is close but how close? When you need to measure a meter-or-so distance with reasonable accuracy, the **VCNL4200 Long Distance Proximity Sensor** from Vishay can do that for you. It's excellent for robotic or motion-detection situations. And if perchance you also needed to measure the amount of light at the same time, you're in luck! The VCNL4200 can do that too, with it's built in ambient light sensor (ALS).



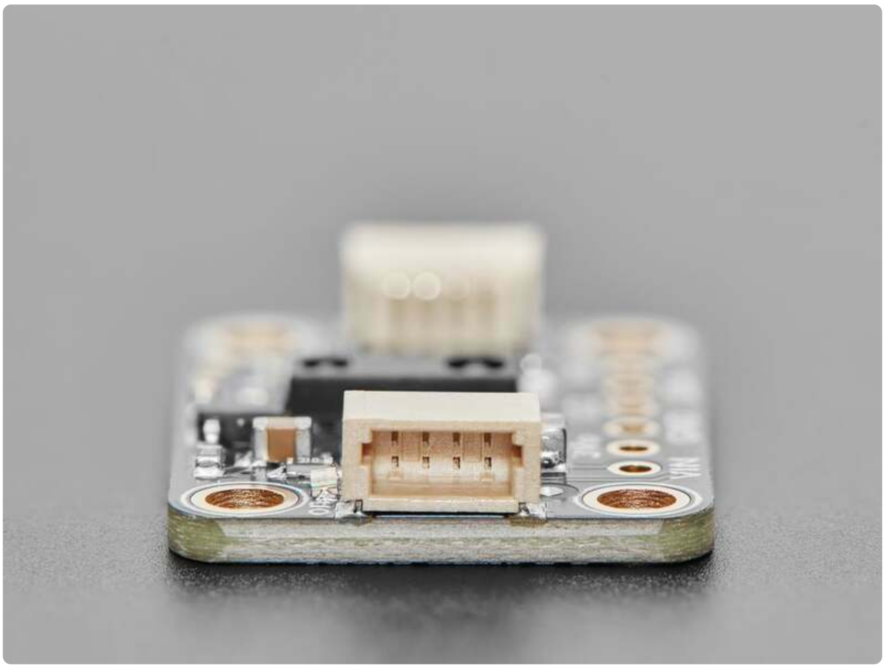
The VCNL4200 is a powerful two-in-one sensor, with a proximity sensor that works from 0 to 1.5m (59 inches) and light sensor with range of 0.003 to 1570 lux. It's the longest-range IR distance sensor we've seen other than the Sharp Analog distance sensors (http://adafru.it/1031), and at a great price with I2C interface.

Note that while the chip itself is specced for 1.5 meter distance measurements, that depends a bit on optimal lighting situations and high power configuration. We've found that 50-100cm is a good range to expect, and you won't get a lot of accuracy/ precision as the detection object gets farther away. We normally suggest ToF sensors (https://adafru.it/1aaz) for long precision measurements, but sometimes you want the low cost and simplicity of infrared. Also, you get a 'free' light sensor which ToF sensors don't contain.



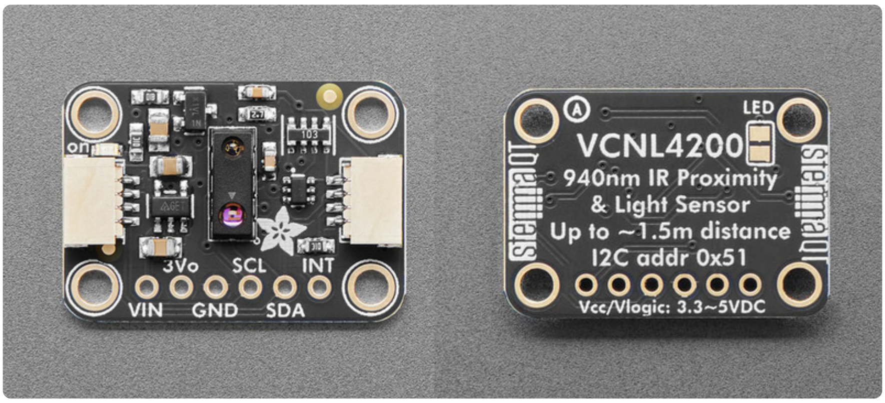
"OK, finally I can get started on my proximity and light sensing project, but how do I use it?" you say. To make life easier so you can focus on your important work, we've taken the VCNL4200 and put it onto a breakout PCB along with support circuitry to let you use this little wonder with 3.3V (Feather/Raspberry 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)** 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 VCNL4200 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)



To give you some ability to tune your measurements for your situation, the VCNL4200 lets you adjust the integration time for both light sensing and proximity, IR LED power, duty cycle, multi-pulse, and sunlight-immunity. There's also interrupt output support for light or proximity measurement thresholds.

Pinouts



Default I2C address is **0x51**.

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. It can be powered between from 3.3V to 5V.
- **3Vo** - this is the 3.3V output from the onboard regulator.
- **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 development 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. You can set up the sensor to pull this low when certain conditions are met such as proximity or lux level thresholds being crossed.

Power LED and Jumper

- **Power LED** - on the left side of the board, above the STEMMA QT connector, is the power LED, labeled **on**. It is a 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.

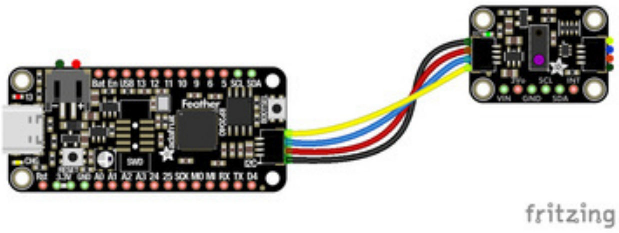
CircuitPython and Python

It's easy to use the **VCNL4200** with Python or CircuitPython, and the Adafruit_CircuitPython_VCNL4200 (https://adafru.it/1aaA) module. This module allows you to easily write Python code to read proximity and lux 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).

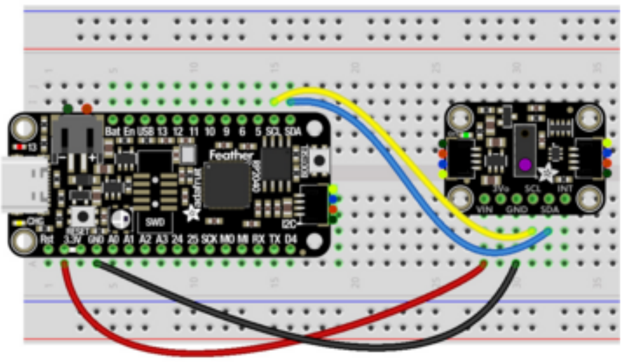
CircuitPython Microcontroller Wiring

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:

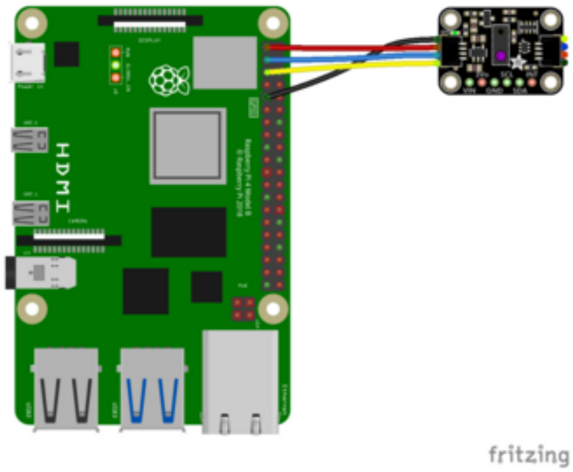


Board 3V to **sensor VIN (red wire)**
Board GND to **sensor GND (black wire)**
Board SCL to **sensor SCL (yellow wire)**
Board SDA to **sensor SDA (blue wire)**

Python Computer Wiring

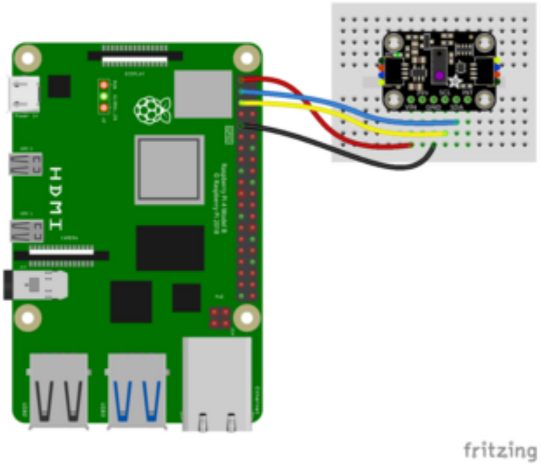
Since there are dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

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 VCNL4200 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-vcnl4200`

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

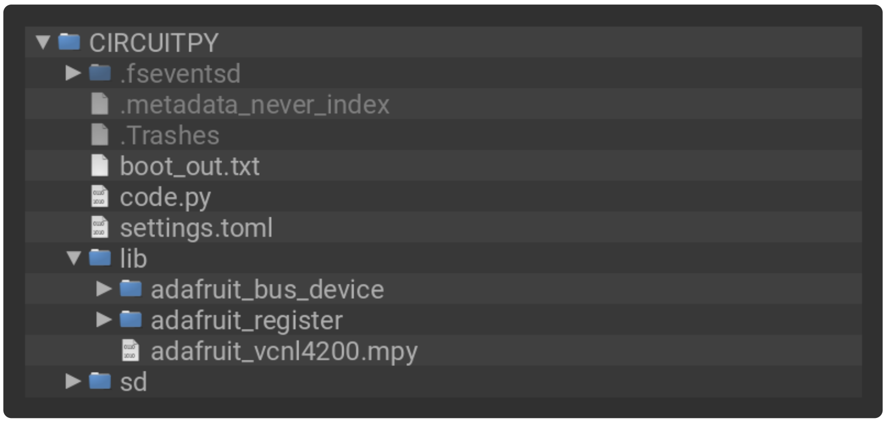
To use with CircuitPython, you need to first install the **Adafruit_CircuitPython_VCNL4200** library, and its dependencies, into the **lib** folder on your **CIRCUITPY** drive. Then you need to update **code.py** with the example script.

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_vcnl4200.mpy**



```
▼ CIRCUITPY
  ▶ .fseventsd
    .metadata_never_index
    .Trashes
    boot_out.txt
    code.py
    settings.toml
  ▼ lib
    ▶ adafruit_bus_device
    ▶ adafruit_register
      adafruit_vcnl4200.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

import time

import board

import adafruit_vcnl4200

i2c = board.I2C()

sensor = adafruit_vcnl4200.Adafruit_VCNL4200(i2c)

while True:
    print(f"Proximity is: {sensor.proximity}")
    print(f"Ambient is: {sensor.lux}")
    time.sleep(0.1)
```

First, the sensor is instantiated over I2C. Then, in the loop, the proximity and lux readings are printed to the serial monitor. You'll see these readings change depending on the proximity of objects in front of the sensor and the amount of ambient light that the sensor is exposed to.

Python Docs

[Python Docs](https://adafru.it/1aa9) (https://adafru.it/1aa9)

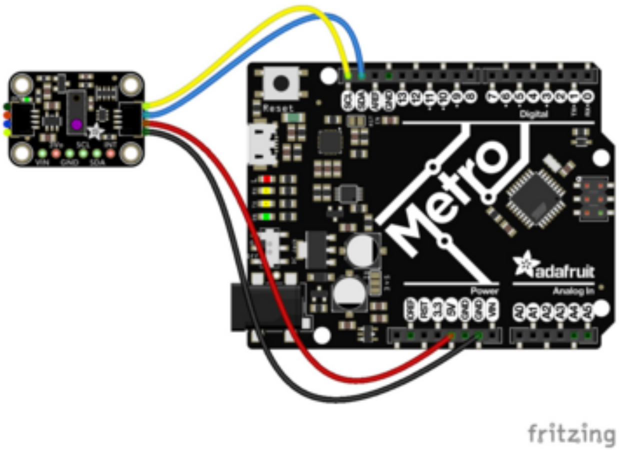
Arduino

Using the VCNL4200 breakout with Arduino involves wiring up the breakout to your Arduino-compatible microcontroller, installing the [Adafruit_VCNL4200](https://adafru.it/1aaB) (https://adafru.it/1aaB) library, and running the provided example code.

Wiring

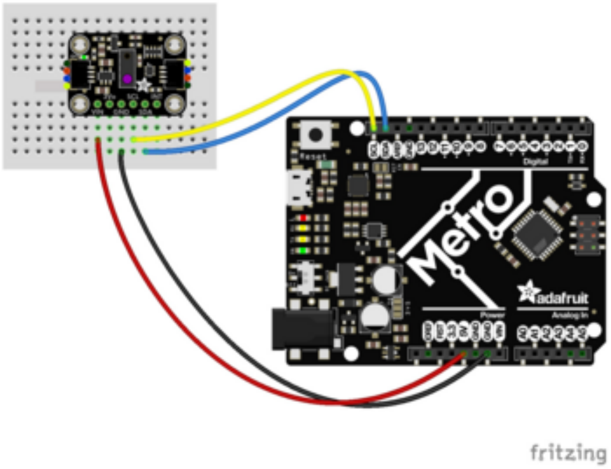
Wire as shown for a **5V** board like an Uno. If you are using a **3V** board, like an Adafruit Feather, wire the board's 3V pin to the sensor VIN.

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)**

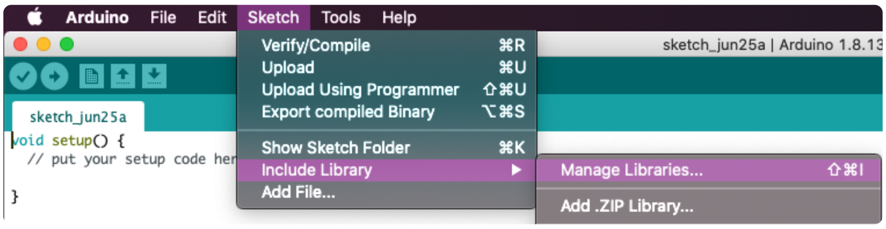
Here is an Adafruit Metro wired up using a solderless breadboard:



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)**

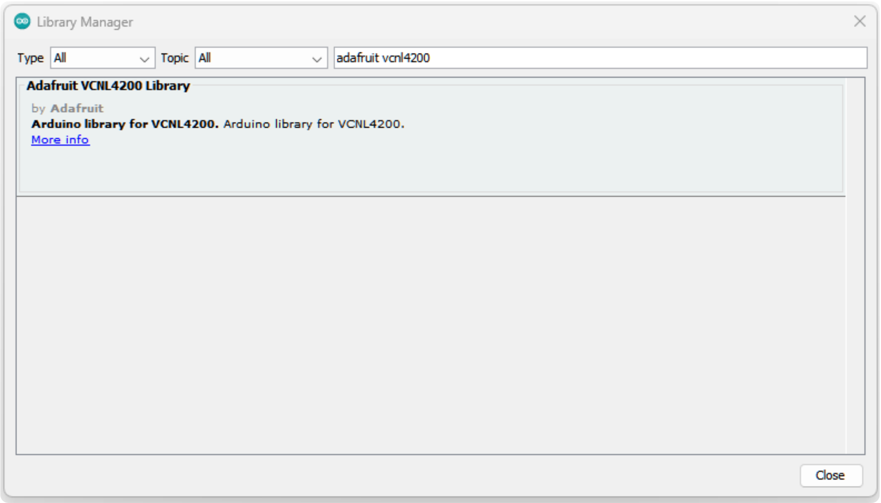
Library Installation

You can install the **Adafruit_VCNL4200** library for Arduino using the Library Manager in the Arduino IDE.



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

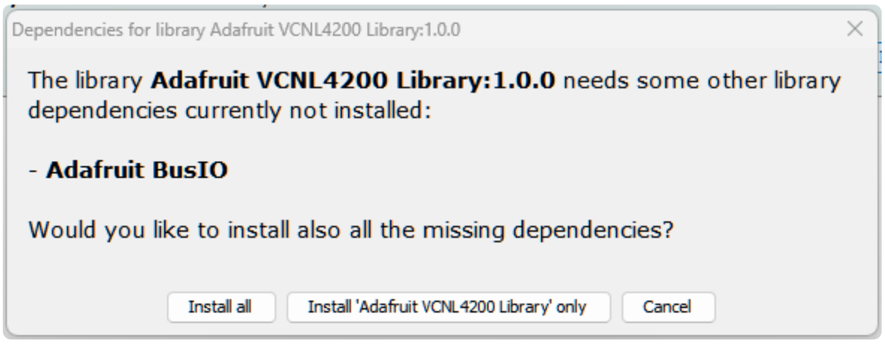
Click the **Manage Libraries ...** menu item, search for **Adafruit_VCNL4200**, and select the **Adafruit VCNL4200** library:



Type: All Topic: All adafruit vcnl4200

Adafruit VCNL4200 Library
by Adafruit
Arduino library for VCNL4200. Arduino library for VCNL4200.
More info

If asked about dependencies, click "Install all".



Dependencies for library Adafruit VCNL4200 Library:1.0.0

The library **Adafruit VCNL4200 Library:1.0.0** needs some other library dependencies currently not installed:

- **Adafruit BusIO**

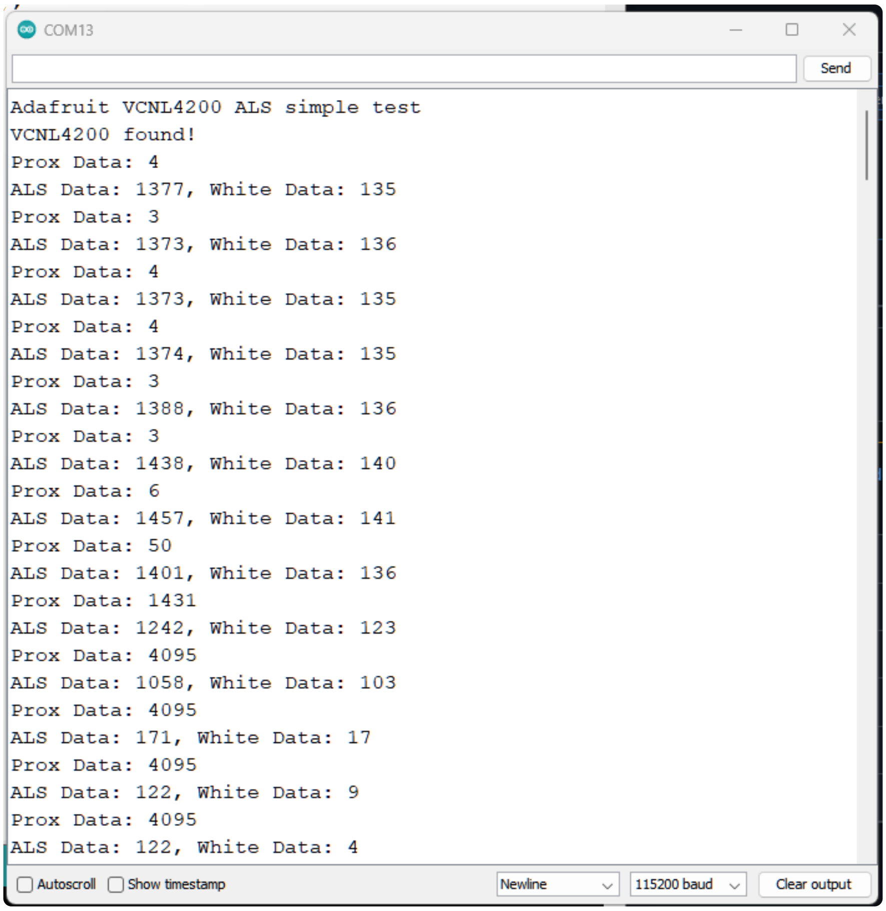
Would you like to install also all the missing dependencies?

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!

Full Test Example

```
// SPDX-FileCopyrightText: 2024 ladyada for Adafruit Industries
//
// SPDX-License-Identifier: MIT

#include "Adafruit_VCNL4200.h"

Adafruit_VCNL4200 vcnl4200;

void setup() {
  Serial.begin(115200);
  while (!Serial) {
    delay(10); // wait for native USB
  }

  Serial.println("Adafruit VCNL4200 ALS simple test");

  if (!vcnl4200.begin()) {
    Serial.println("Could not find a valid VCNL4200 sensor, check wiring!");
    while (1) {
      delay(10);
    }
  }
  Serial.println("VCNL4200 found!");

  vcnl4200.setALSshutdown(false);
  vcnl4200.setALSIntegrationTime(VCNL4200_ALS_IT_100MS);
  vcnl4200.setALSPersistence(VCNL4200_ALS_PERS_2);

  vcnl4200.setProxShutdown(false);
  vcnl4200.setProxHD(false);
  vcnl4200.setProxLEDCurrent(VCNL4200_LED_I_200MA);
  vcnl4200.setProxIntegrationTime(VCNL4200_PS_IT_8T);
}

void loop() {
  // Read the proximity sensor data
  uint16_t proxData = vcnl4200.readProxData();
  Serial.print("Prox Data: ");
  Serial.println(proxData);
  // Read the ambient light sensor (ALS) data
  uint16_t alsData = vcnl4200.readALSdata();
  Serial.print("ALS Data: ");
  Serial.print(alsData);
  // Read the raw white sensor data
  uint16_t whiteData = vcnl4200.readWhiteData();
  Serial.print(", White Data: ");
  Serial.println(whiteData);

  delay(100);
}
```



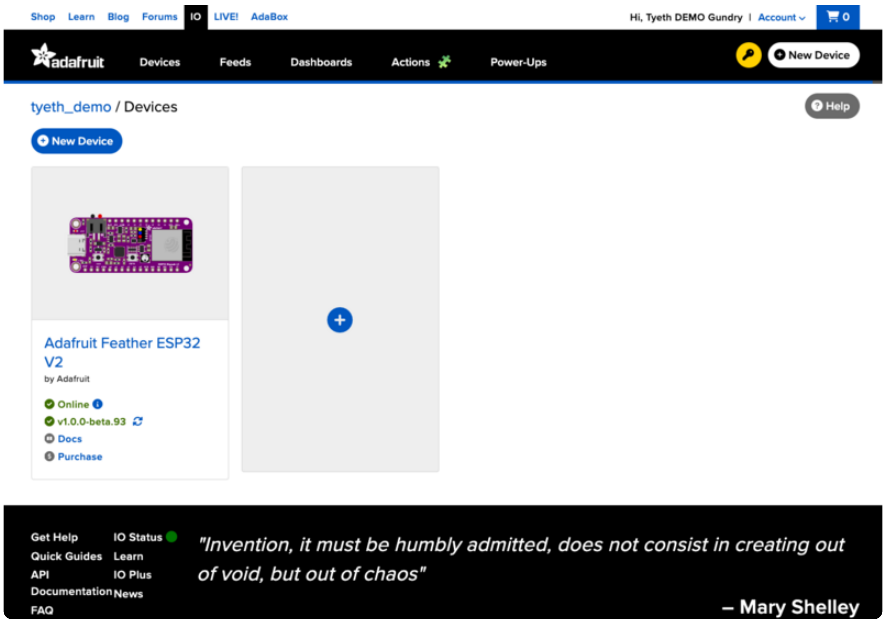
```
Adafruit VCNL4200 ALS simple test
VCNL4200 found!
Prox Data: 4
ALS Data: 1377, White Data: 135
Prox Data: 3
ALS Data: 1373, White Data: 136
Prox Data: 4
ALS Data: 1373, White Data: 135
Prox Data: 4
ALS Data: 1374, White Data: 135
Prox Data: 3
ALS Data: 1388, White Data: 136
Prox Data: 3
ALS Data: 1438, White Data: 140
Prox Data: 6
ALS Data: 1457, White Data: 141
Prox Data: 50
ALS Data: 1401, White Data: 136
Prox Data: 1431
ALS Data: 1242, White Data: 123
Prox Data: 4095
ALS Data: 1058, White Data: 103
Prox Data: 4095
ALS Data: 171, White Data: 17
Prox Data: 4095
ALS Data: 122, White Data: 9
Prox Data: 4095
ALS Data: 122, White Data: 4
```

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. You'll see the VCNL4200 recognized over I2C. Then, proximity, ambient light and white light readings will be printed to the Serial Monitor. You'll see these values changes as you affect your proximity to the sensor and the ambient light around the sensor.

Arduino Docs

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

WipperSnapper



"Invention, it must be humbly admitted, does not consist in creating out of void, but out of chaos"

– Mary Shelley

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.

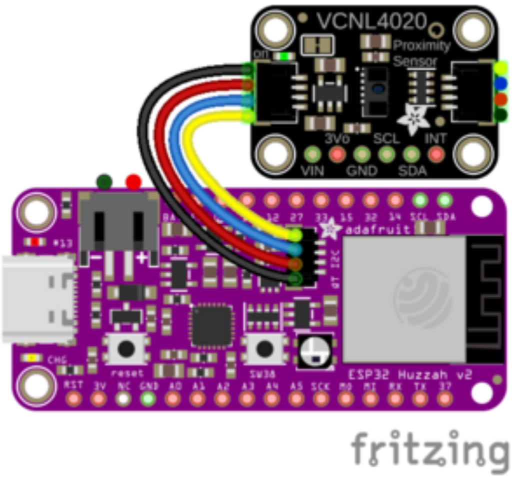
If you've never used WipperSnapper, click below to read through the quick start guide before continuing.

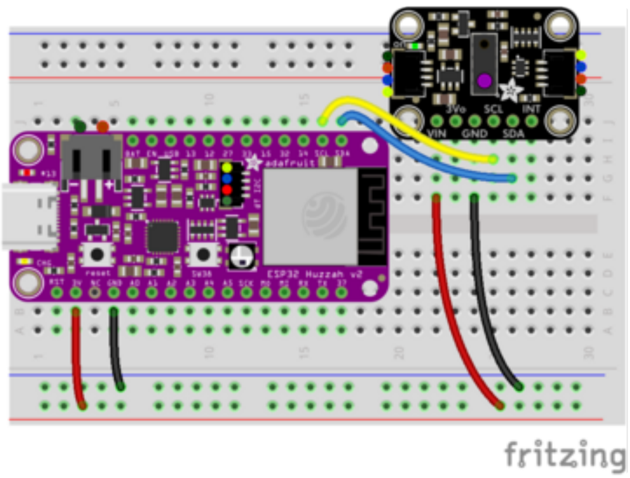
Quickstart: Adafruit IO WipperSnapper

https://adafru.it/Vfd

Wiring

First, wire up an VCNL4200 to your board exactly as follows. Here is an example of the VCNL4200 wired to an [Adafruit ESP32 Feather V2](http://adafru.it/5400) (http://adafru.it/5400) using I2C [with a STEMMA QT cable (no soldering required)](http://adafru.it/4210) (http://adafru.it/4210)



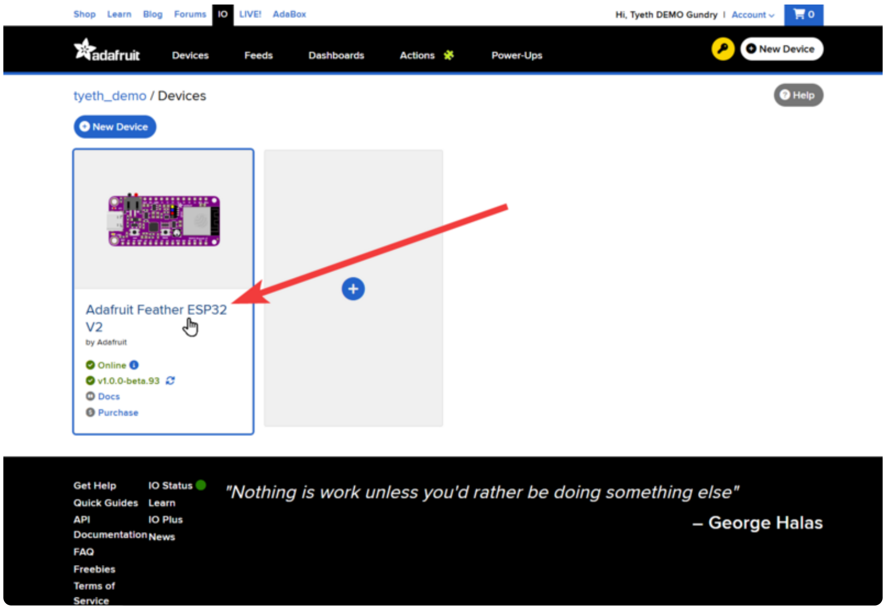


Board 3V to **sensor VIN (red wire on STEMMA QT)**
Board GND to **sensor GND (black wire on STEMMA QT)**
Board SCL to **sensor SCL (yellow wire on STEMMA QT)**
Board SDA to **sensor SDA (blue wire on STEMMA QT)**

Usage

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

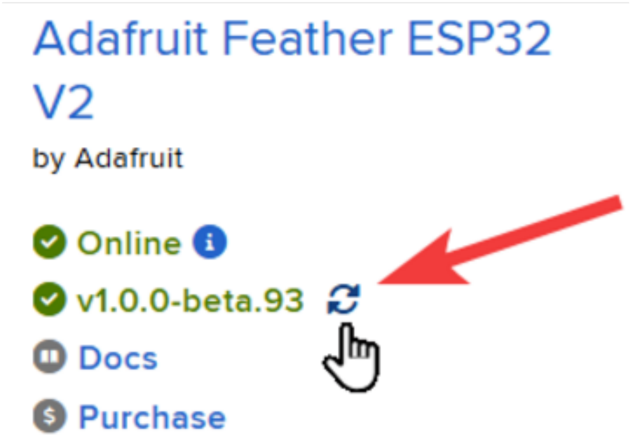
On this page, **select the WipperSnapper board you're using** to be brought to the board's interface page.

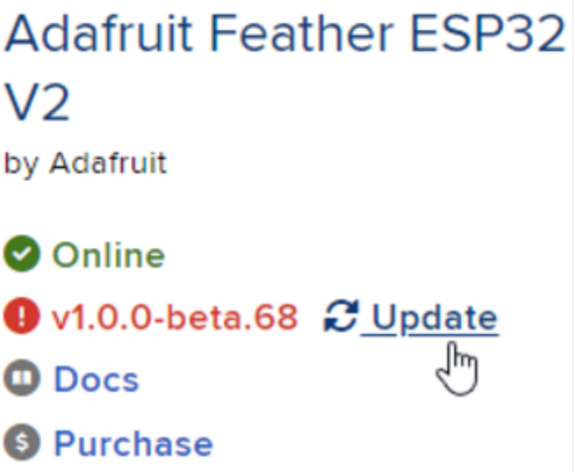


"Nothing is work unless you'd rather be doing something else"

– George Halas

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



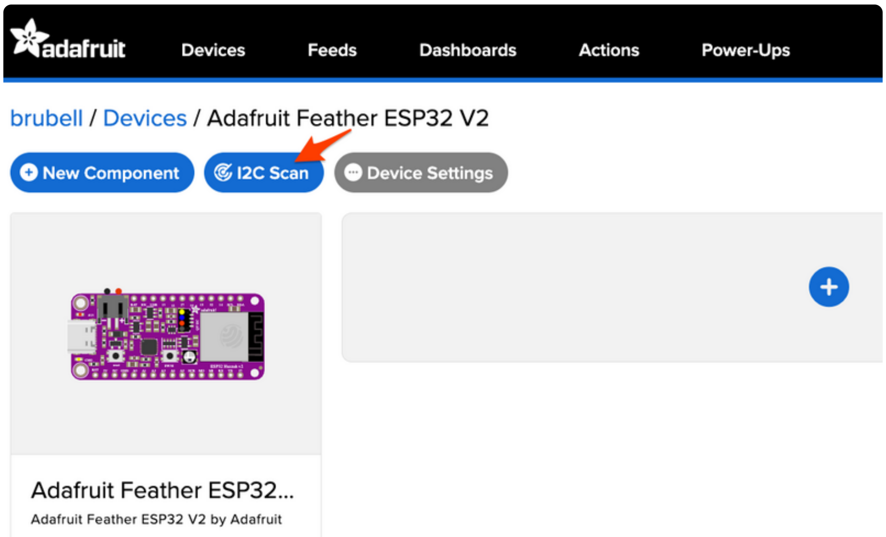


On the device page, quickly **check that you're running the latest version of the WipperSnapper firmware**.

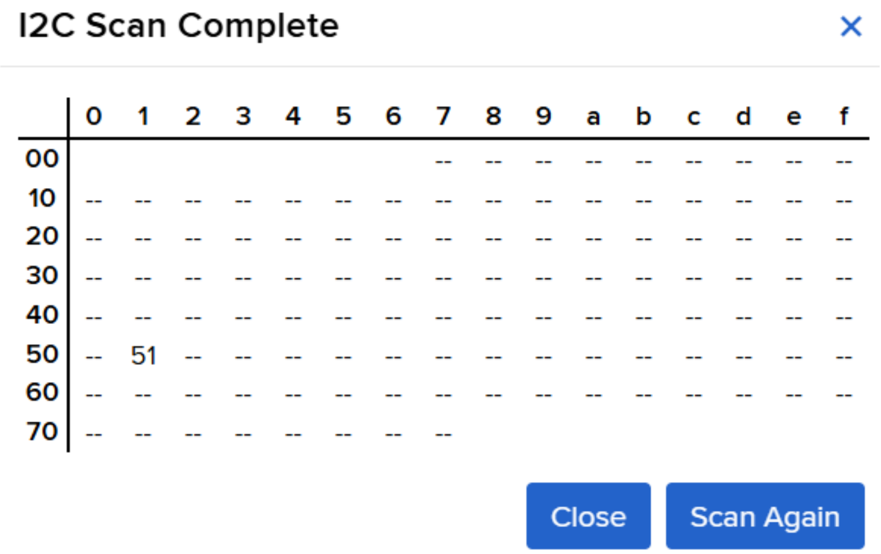
The device tile on the left indicates the version number of the firmware running on the connected board.

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.


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 VCNL4200's default I2C address of `0x51` 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	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50	--	51	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--								

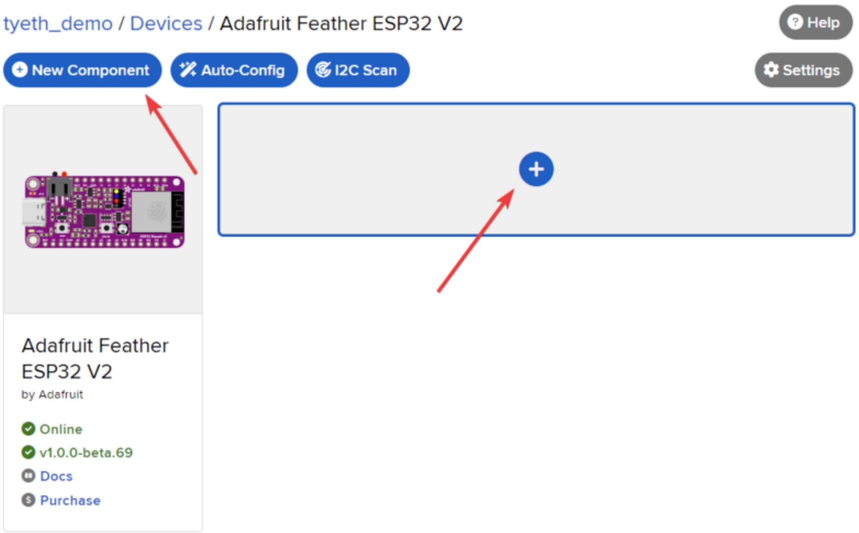
I don't see the sensor's I2C address listed!

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

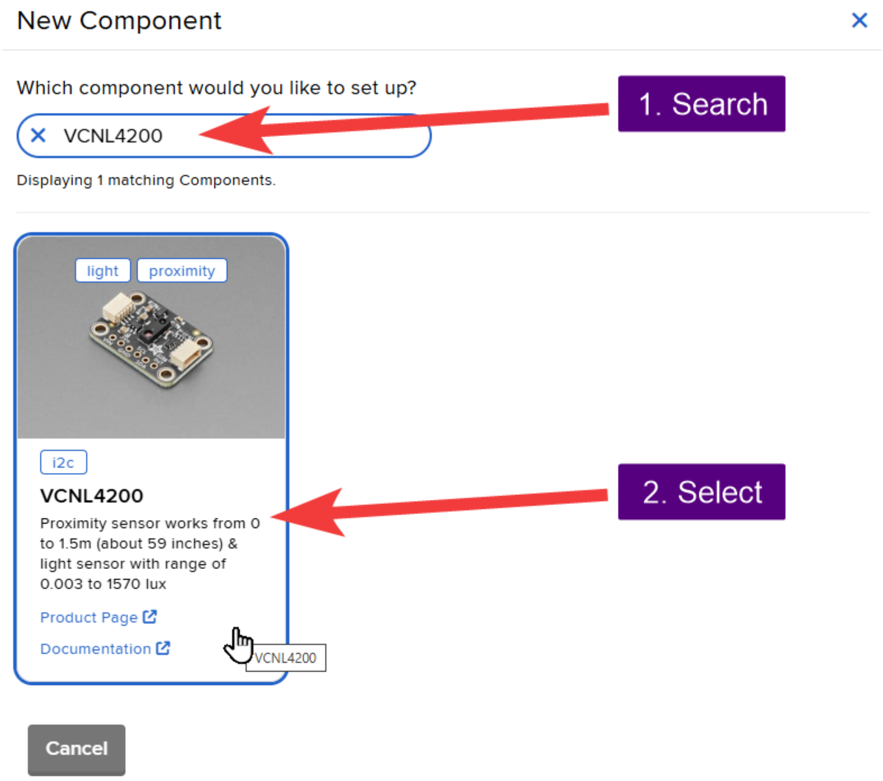
Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

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 `VCNL4200` into the search bar, then select the **VCNL4200** component.


New Component

Which component would you like to set up?

VCNL4200

Displaying 1 matching Components.

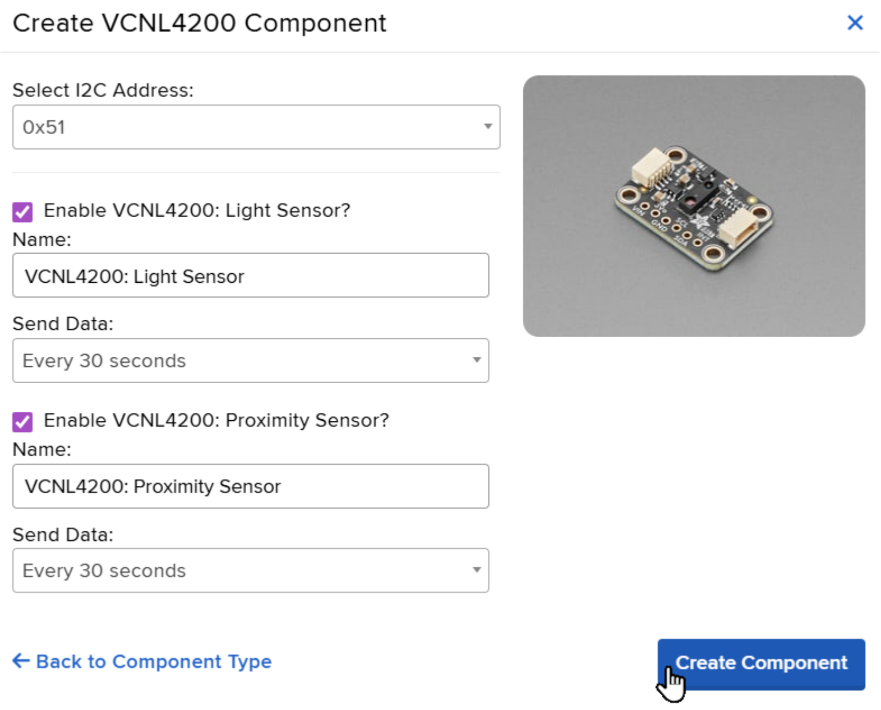
VCNL4200

Proximity sensor works from 0 to 1.5m (about 59 inches) & light sensor with range of 0.003 to 1570 lux

On the component configuration page, the VCNL4200'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 VCNL4200 sensor and send the data to Adafruit IO. Measurements can range from every 30 seconds to every 24 hours.

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



Create VCNL4200 Component

Select I2C Address: 0x51

☑ Enable VCNL4200: Light Sensor?

Name: VCNL4200: Light Sensor

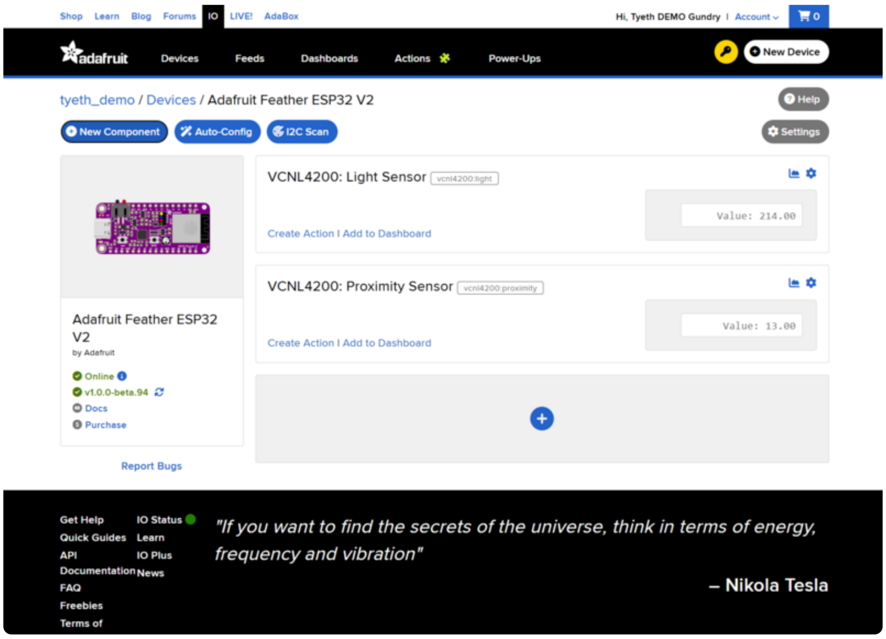
Send Data: Every 30 seconds

☑ Enable VCNL4200: Proximity Sensor?

Name: VCNL4200: Proximity Sensor

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

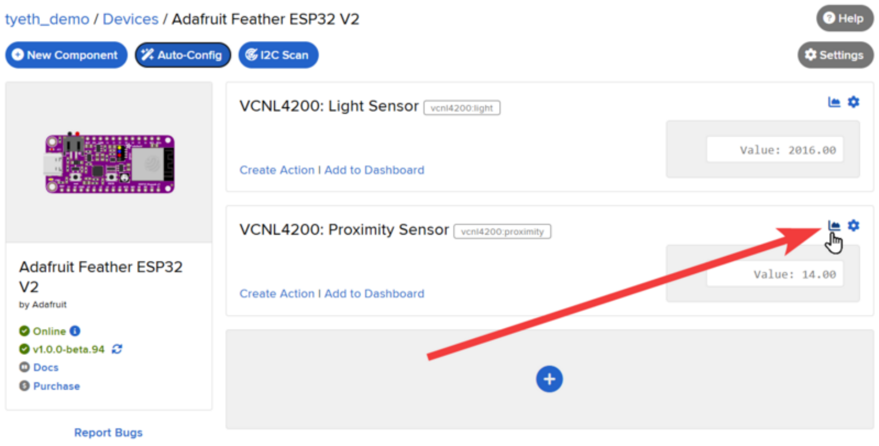
VCNL4200: Light Sensor vcnl4200:light — Value: 214.00 — Create Action I Add to Dashboard

VCNL4200: Proximity Sensor vcnl4200:proximity — Value: 13.00 — Create Action I Add to Dashboard

Adafruit Feather ESP32 V2
by Adafruit
Online
v1.0.0-beta.94
Docs
Purchase

"If you want to find the secrets of the universe, think in terms of energy, frequency and vibration"
– Nikola Tesla

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

Adafruit Feather ESP32 V2
by Adafruit

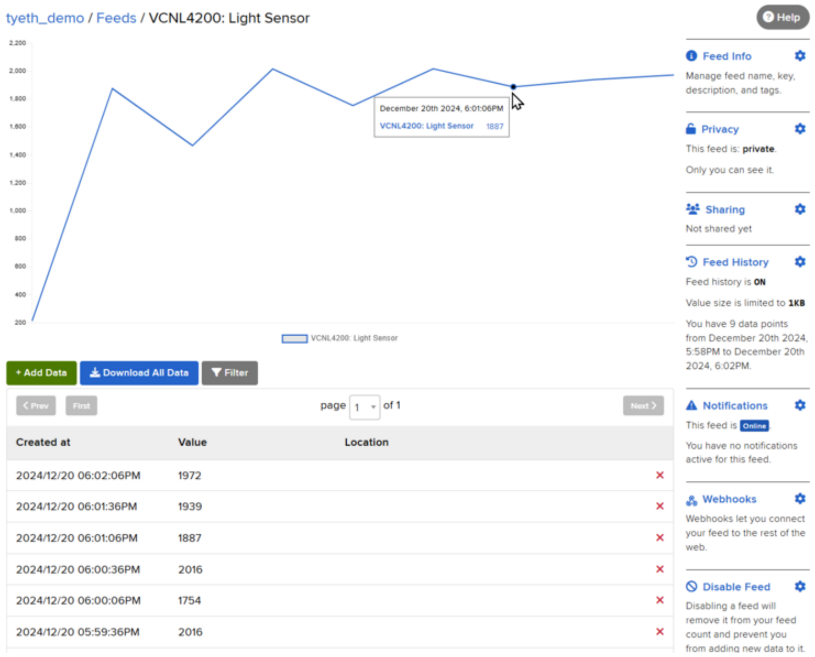
VCNL4200: Light Sensor vcnl4200:light

Value: 2016.00

VCNL4200: Proximity Sensor vcnl4200:proximity

Value: 14.00

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 / VCNL4200: Light Sensor

Created at	Value	Location
2024/12/20 06:02:06PM	1972	
2024/12/20 06:01:36PM	1939	
2024/12/20 06:01:06PM	1887	
2024/12/20 06:00:36PM	2016	
2024/12/20 06:00:06PM	1754	
2024/12/20 05:59:36PM	2016	

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 9 data points from December 20th 2024, 5:58PM to December 20th 2024, 6:02PM.

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.

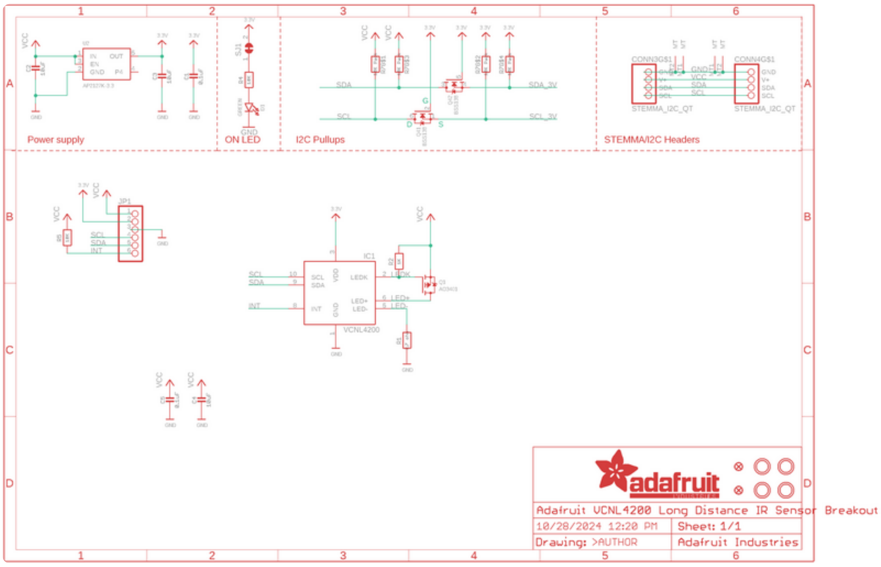
Disable Feed
Disabling a feed will remove it from your feed count and prevent you from adding new data to it.

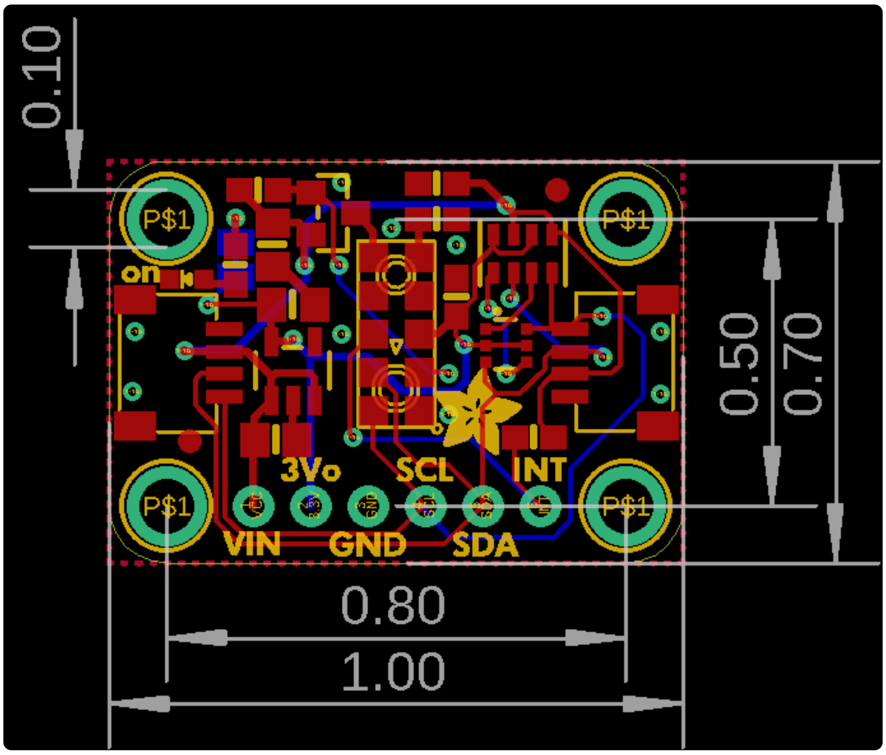
Downloads

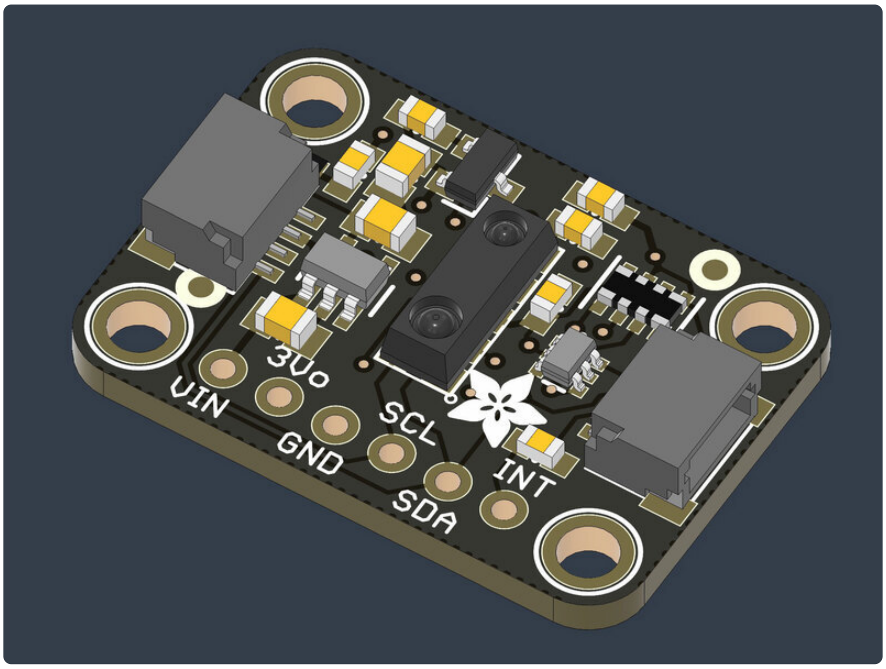
Files

- VCNL4200 Datasheet (https://adafru.it/1aaC)
- EagleCAD PCB Files on GitHub (https://adafru.it/1aaD)
- 3D models on GitHub (https://adafru.it/1ahX)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/1aaE)

Schematic and Fab Print






3Vo
VIN
GND
SCL
SDA
INT