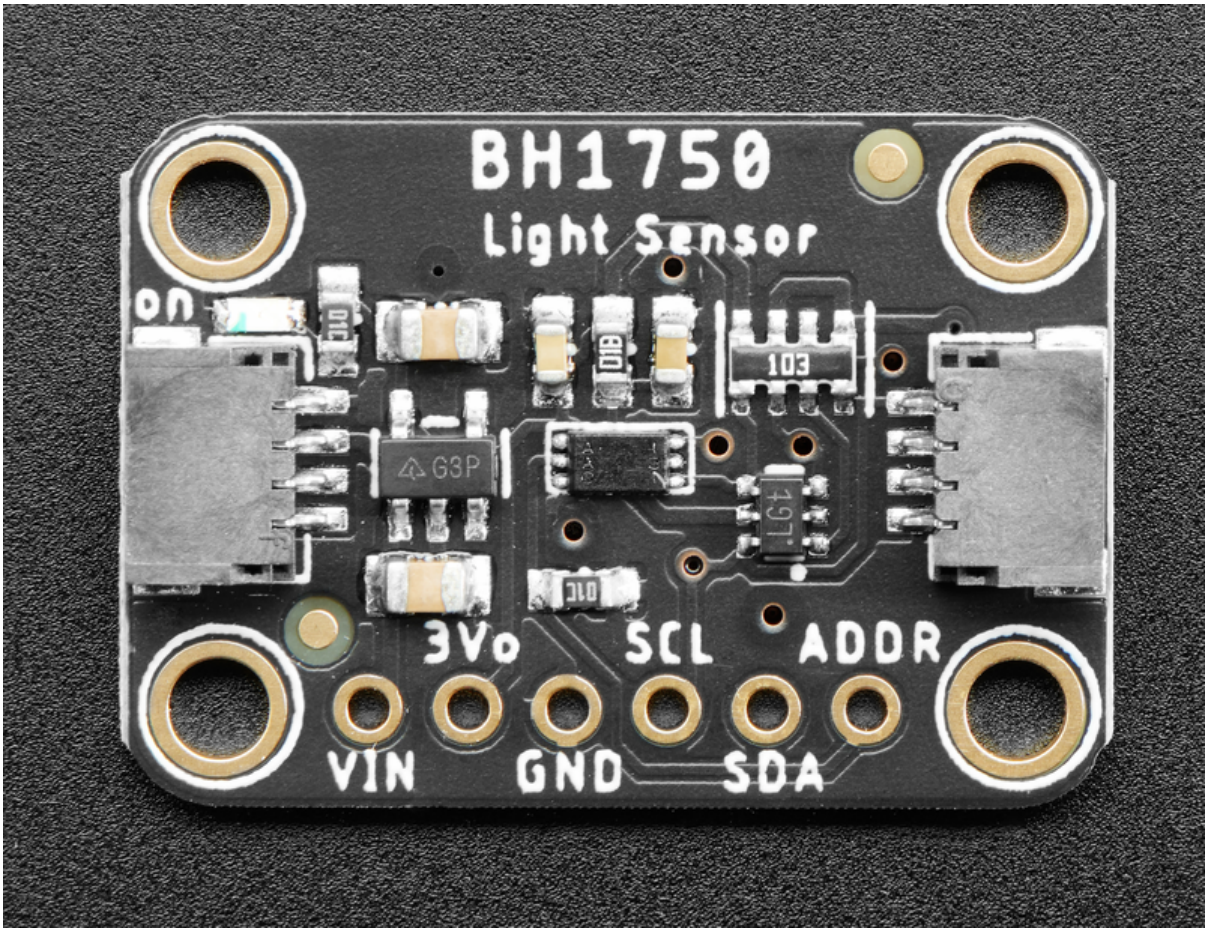
# Adafruit BH1750 Ambient Light Sensor

Created by Bryan Siepert

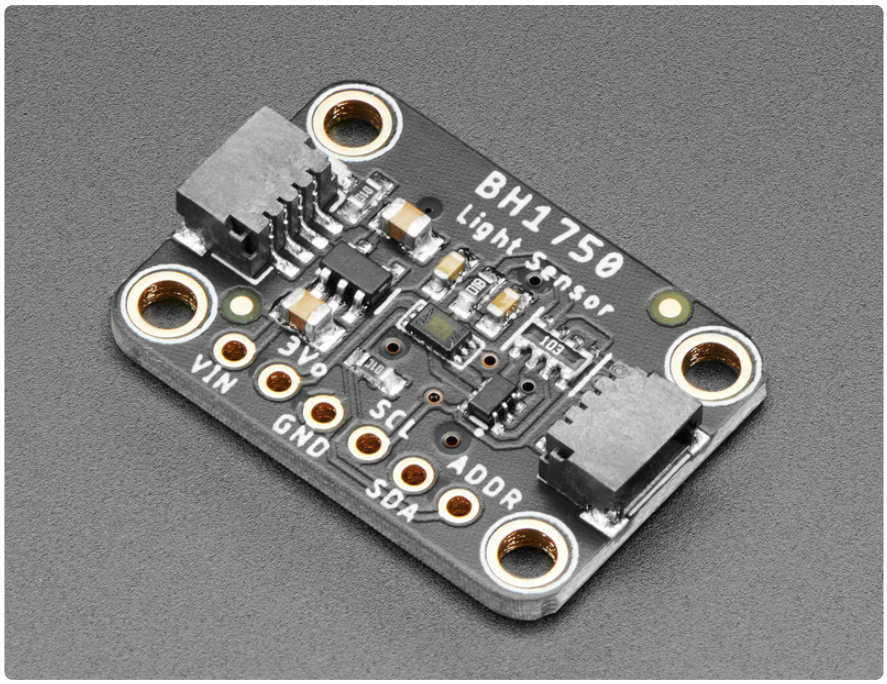



https://learn.adafruit.com/adafruit-bh1750-ambient-light-sensor

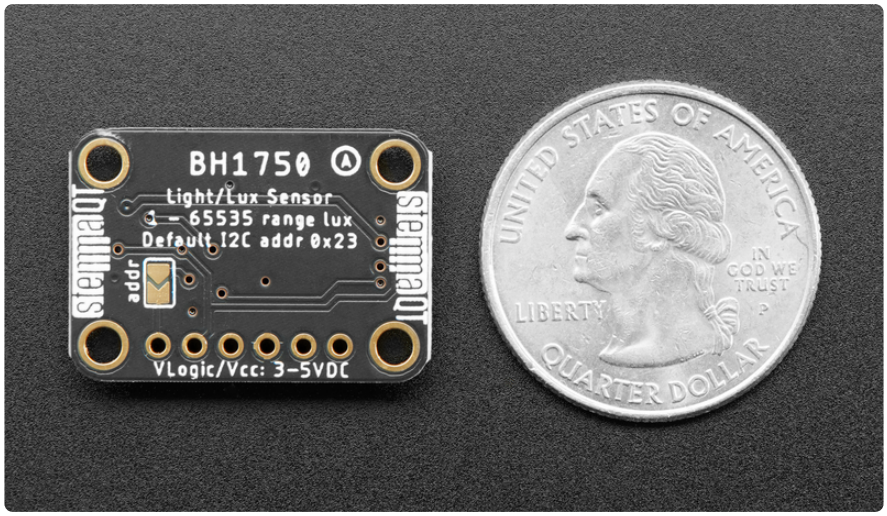
Last updated on 2024-06-03 03:09:29 PM EDT

# Table of Contents

# Overview

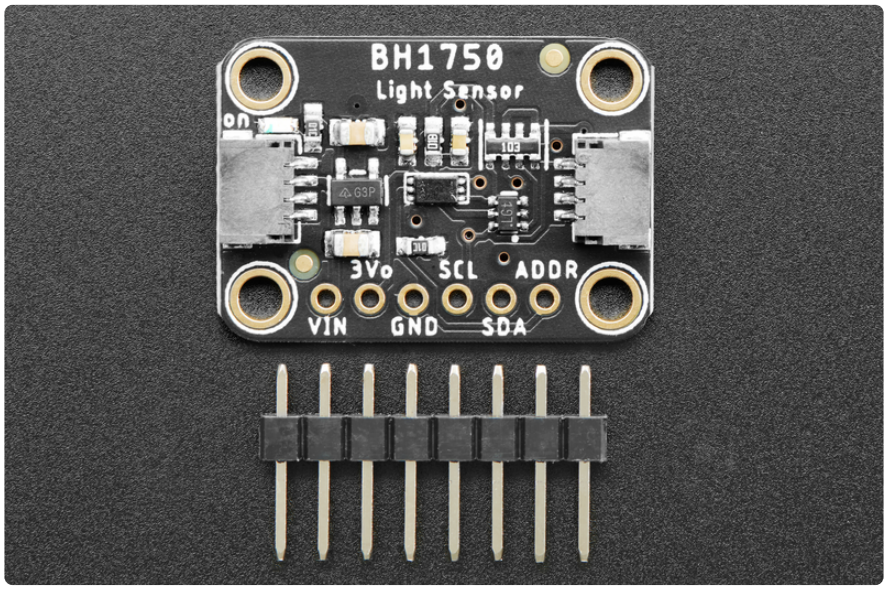



This is the BH1750 16-bit Ambient Light sensor from Rohm. Because of how important it is to humans and most other living things, sensing the amount of light in an environment is a common place to get started when learning to work with microcontrollers and sensors. Should we turn up the brightness of our display or dim it to save power? Which direction should your robot move to stay in an area with the most light? Is it day or night? All of these questions can be answered with the help of the BH1750 (https://adafru.it/MfK). It's a small, capable and inexpensive light sensor that you can include into your next project to add the detection and measurement of light.

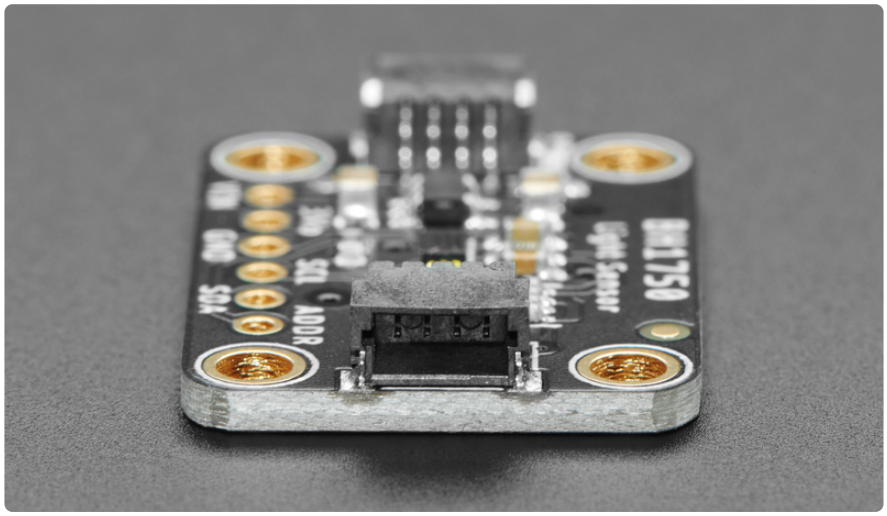



The BH1750 provides 16-bit  light measurements in lux, the SI unit for measuring light making it easy to compare against other values like references and measurements

from other sensors. Able to measure from 0 to 65K+ lux, the BH1750. With some calibration and advanced adjustment of the measurement time, it can even be convinced to measure as much as 100,000 lux!




Sensors tend to come in small packages and the BH1750 is no different. Not much bigger than a grain of rice, this handy light sensing friend needs some help to be used by folks experimenting and without the desire or tools to work with surface mount parts. We're here to help! Packaged on a PCB in our **Stemma QT** form factor, the BH1750 comes integrated with a voltage regulator and level shifting circuitry to allow it to be used with 3.3V devices like a **Feather M4** or **Raspberry Pi**, or **5V** devices such as an **Arduino**. Rather than working with the itty bitty little contacts on the sensor, the PCB it's packaged on breaks out all the pins to a standard 0.1 inch / 2.54mm pitch header.

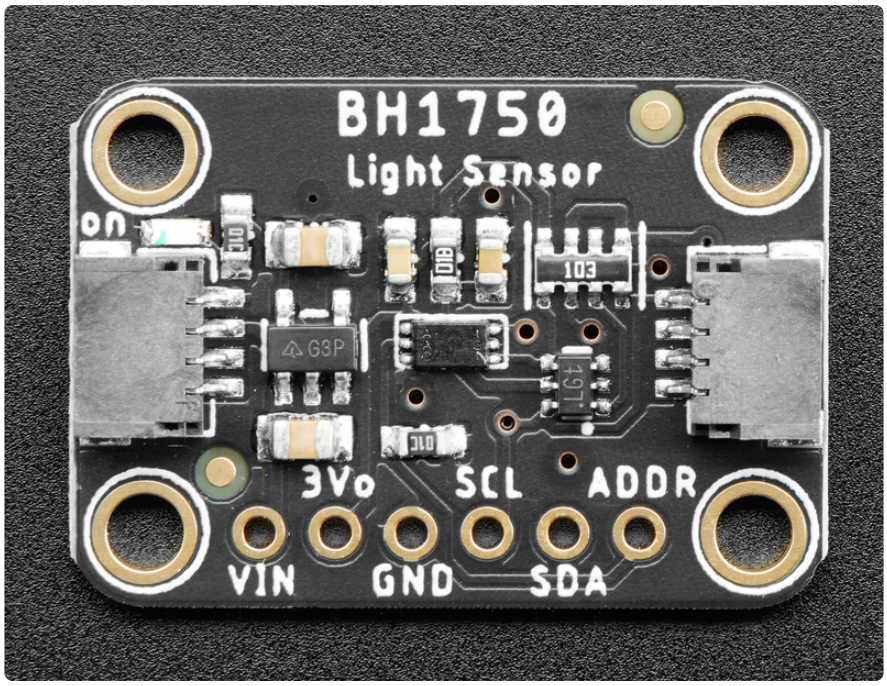


To make things easier and a bit more flexible, we've also included SparkFun Qwiic (https://adafru.it/Fpw) compatible STEMMA QT (https://adafru.it/Ft4) connectors for the I2C bus so you don't even need to solder!

Just plug in a compatible cable and attach it to your MCU of choice and you're ready to load up some software and measure some light. To help with the software part we've written a library that you can use with **CircuitPython** compatible devices as well as computers like the Raspberry Pi by installing it using PyPi. You can even use it on a full sized computer by using a MCP2221 breakout. (http://adafru.it/4471) **Arduino** users can use the well-crafted hp_BH1750 (https://adafru.it/MfD) library by Stefan Armborst ( https://adafru.it/MfE) and our installation instructions and wiring diagrams.

---

# Pinouts



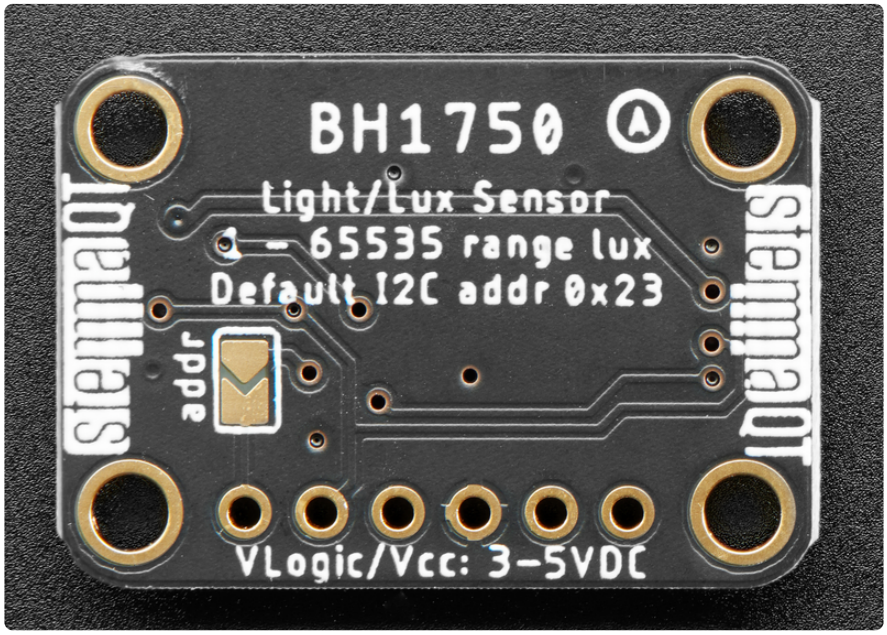

## Power Pins

- **VIN** - this is the power pin. Since the sensor chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V microcontroller like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

## I2C Logic Pins

- **SCL** - I2C clock pin, connect to your microcontroller I2C clock line. This pin is level shifted so you can use 3-5V logic, and there's a **10K pullup** on this pin.

- **SDA** - I2C data pin, connect to your microcontroller I2C data line. This pin is level shifted so you can use 3-5V logic, and there's a **10K pullup** on this pin.
- **STEMMA QT** (https://adafru.it/Ft4) **-** These connectors allow you to connectors to dev boards with **STEMMA QT** connectors or to other things with various associated accessories (https://adafru.it/Ft6)
- **ADDR/AD0 Jumper** - **I2C Address pin**. Pulling this pin high or bridging the solder jumper on the back will change the I2C address from **0x23** to **0x5C**



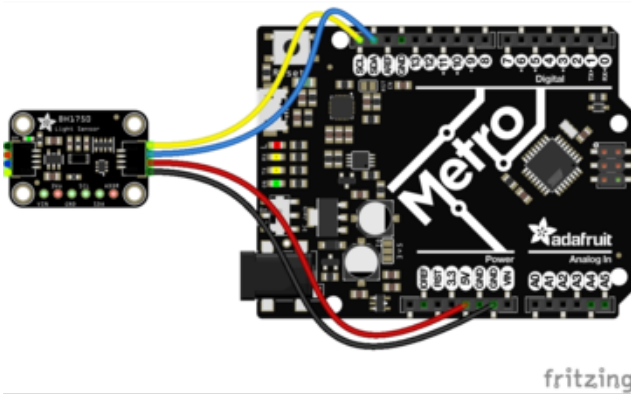

# Arduino

Using the BH1750 with Arduino is a simple matter of wiring up the sensor to your Arduino-compatible microcontroller, installing the hp_BH1750 (https://adafru.it/MfD) library written by Stefan Armborst (https://adafru.it/MfE), and running one of many very well written examples. Usually we write our own library but we were so impressed by Stefan's that we didn't think we could possibly improve on it, so use it!

## I2C Wiring

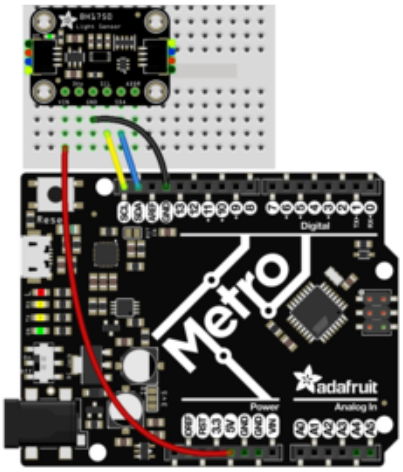
Use this wiring if you want to connect via I2C interface. The I2C address address for the BH1750 is **0x23** and can be switched to **0x5C** by pulling the address pin high to VCC

Here is how to wire up the sensor using one of the **STEMMA QT** (https://adafru.it/Ft4) connectors. The examples show a Metro but wiring will work the same for an Arduino or other compatible board.




Connect **board VIN (red wire)** to **Arduino 5V** if you are running a **5V** board Arduino (Uno, etc.). If your board is **3V,** connect to that instead.
Connect **board GND (black wire)** to **Arduino GND**
Connect **board SCL (yellow wire)** to **Arduino SCL**
Connect **board SDA (blue wire)** to **Arduino SDA**

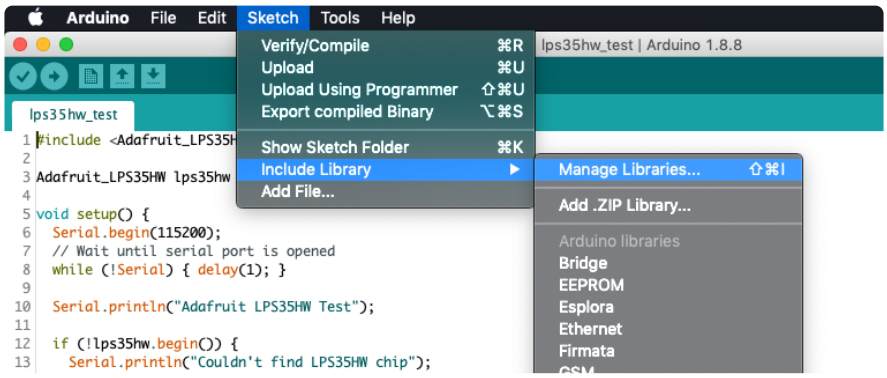
Here is how to wire the sensor to a board using a solderless breadboard:



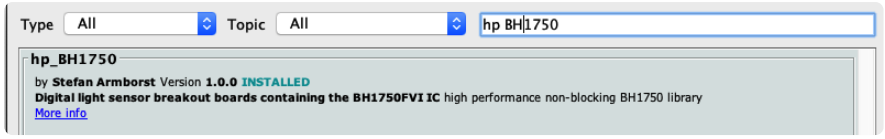
Connect **board VIN (red wire)** to **Arduino 5V** if you are running a **5V** board Arduino (Uno, etc.). If your board is **3V,** connect to that instead.
Connect **board GND (black wire)** to **Arduino GND**
Connect **board SCL (yellow wire)** to **Arduino SCL**
Connect **board SDA (blue wire)** to **Arduino SDA**

## Library Installation

You can install the hp_BH1750 (https://adafru.it/MfD) library for Arduino using the Library Manager in the Arduino IDE.




Click the **Manage Libraries ...** menu item, search for **hp BH1750**, and select the **hp_BH1750** library:




## Load Example

To show the fundamental feature of the sensor, measuring the ambient light in lux, you can use the modestly titled "**BareMinimum**" example:

Open **File -> Examples -> hp_BH1750 -> BareMinimum**

After opening the demo file, upload to your Arduino wired up to the sensor. Once you upload the code, you will see the **Lux** values being printed when you open the Serial Monitor (**Tools->Serial Monitor**) at **9600 baud**, similar these measurements I got from waving a flashlight at the sensor

```
331.67
331.67
332.92
338.75
442.50
1443.33
1790.00
515.42
1530.00
1312.92
1257.50
1280.00
345.42
343.75
344.17
```

## Example Code

A full listing (https://adafru.it/MfF) of the above example can be found in the hp_BH1750 repo's example directory (https://adafru.it/MfG) along with several other examples that show the many features of the library.

---

# Arduino Docs

Arduino Docs (https://adafru.it/M0A)

---

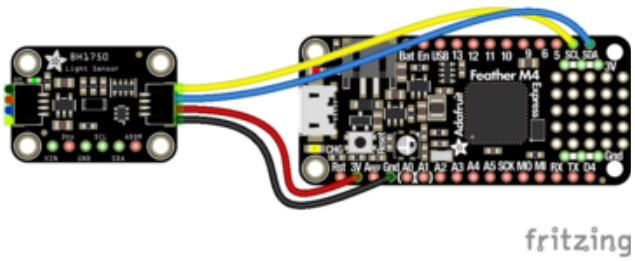
# Python & CircuitPython

It's easy to use the **BH1750** with Python or CircuitPython, and the Adafruit CircuitPython BH1750 (https://adafru.it/MfH) module. This module allows you to easily write Python code that reads lux measurements from the **BH1750**.

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).

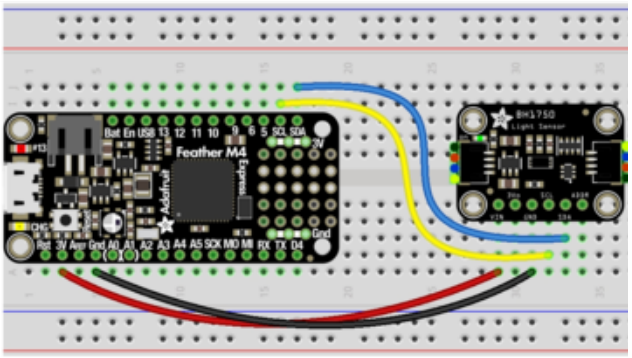
## CircuitPython Microcontroller Wiring

First wire up a BH1750 to your board exactly as shown below. Here's an example of wiring a Feather M4 to the sensor with I2C using one of the handy **STEMMA QT** (https://adafru.it/Ft4) connectors:




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

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

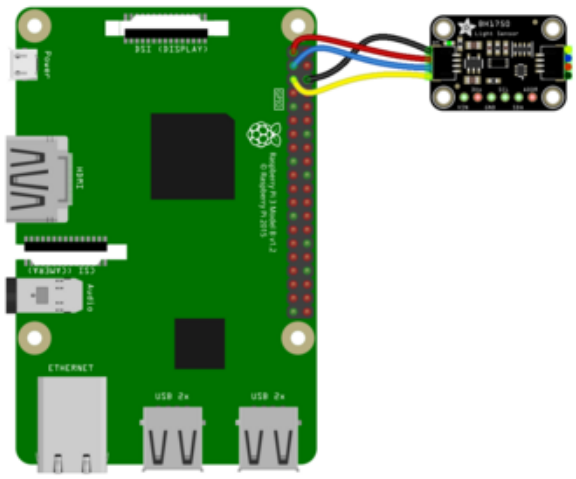


**Board 3V** to **sensor VIN (red wire)**
**Board GND** to **sensor GND (black wire)**
**Board SCL** to **sensor SCL (yellow wire)**
**Board SDA** to **sensor SDA (blue wire)**

## Python Computer Wiring

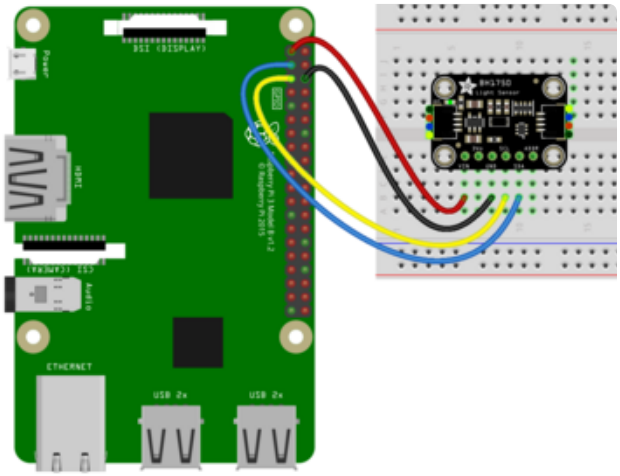
Since there's dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

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



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

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



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

## CircuitPython Installation of BH1750 Library

You'll need to install the Adafruit CircuitPython BH1750 (https://adafru.it/MfH) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafru.it/Amd) for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (https://adafru.it/ENC). Our CircuitPython starter guide has a great page on how to install the library bundle (https://adafru.it/ABU).

Before continuing make sure your board's **lib** folder or root filesystem has the **adafruit_bh1750.mpy** file **and adafruit_bus_device** folder copied over.

Next connect to the board's serial REPL (https://adafru.it/Awz)so you are at the CircuitPython >>> prompt.

## Python Installation of BH1750 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:

- sudo pip3 install adafruit-circuitpython-bh1750

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

## CircuitPython & Python Usage

To demonstrate the usage of the sensor we'll initialize it and read the temperature and humidity measurements from the board's Python REPL.

Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
>>> import board
>>> import adafruit_bh1750
>>> i2c = board.I2C()
>>> sensor = adafruit_bh1750.BH1750(i2c)
```

Now you're ready to read values from the sensor using the **lux** property to return the ambient light level in lux, the SI derived unit (https://adafru.it/MfI) for measuring illuminance (https://adafru.it/MfJ)

```
>>> print("%.2f Lux" % sensor.lux)
333.33 Lux
```

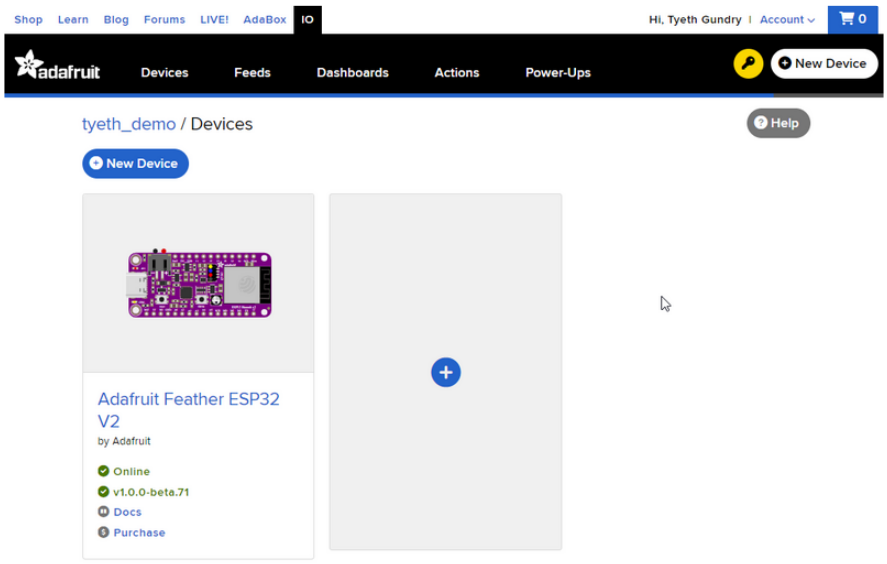
## Example Code

```
# SPDX-FileCopyrightText: 2020 Bryan Siepert, written for Adafruit Industries

# SPDX-License-Identifier: Unlicense
import time
import board
import adafruit_bh1750

i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller
sensor = adafruit_bh1750.BH1750(i2c)

while True:
    print("%.2f Lux" % sensor.lux)
    time.sleep(1)
```

# Python Docs

Python Docs (https://adafru.it/M0B)

# 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.

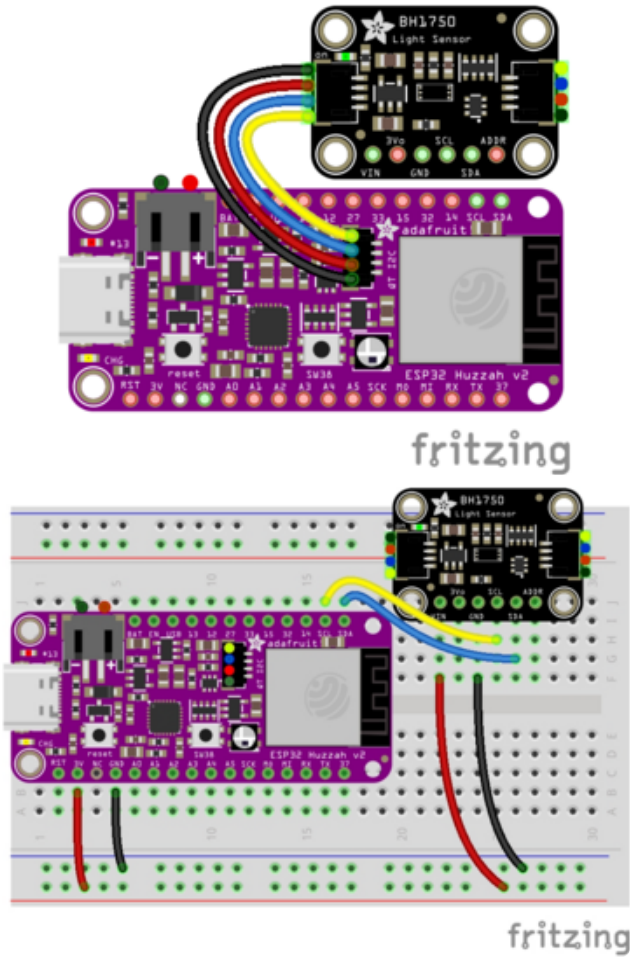
If you've never used WipperSnapper, click below to read through the quick start guide before continuing.

**Quickstart: Adafruit IO WipperSnapper**

https://adafru.it/Vfd

## Wiring

First, wire up a BH1750 to your board exactly as follows. Here is an example of the BH1750 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)

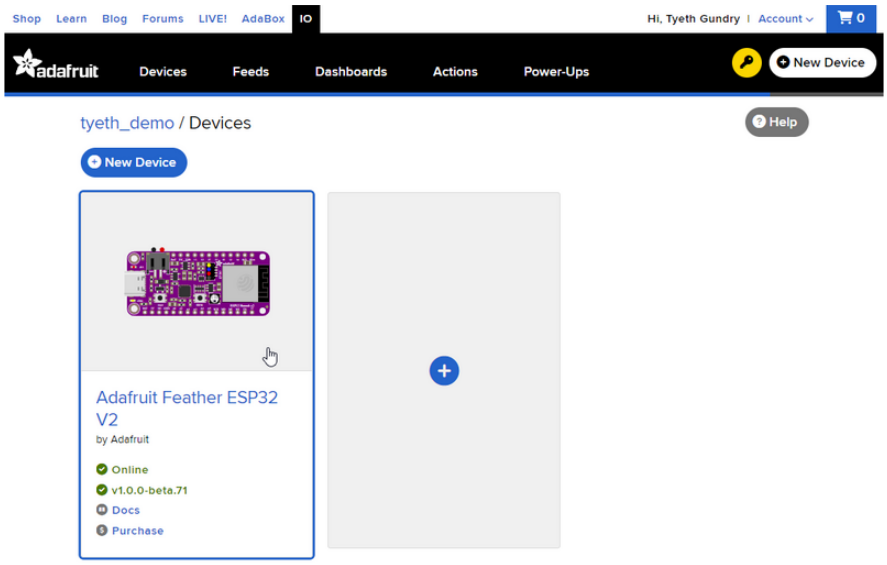



**Board 3V** to **sensor VIN (red wire on STEMMA QT)**
**Board GND** to **sensor GND (black wire on STEMMA QT)**
**Board SCL** to **sensor SCL (yellow wire on STEMMA QT)**
**Board SDA** to **sensor SDA (blue wire on STEMMA QT)**

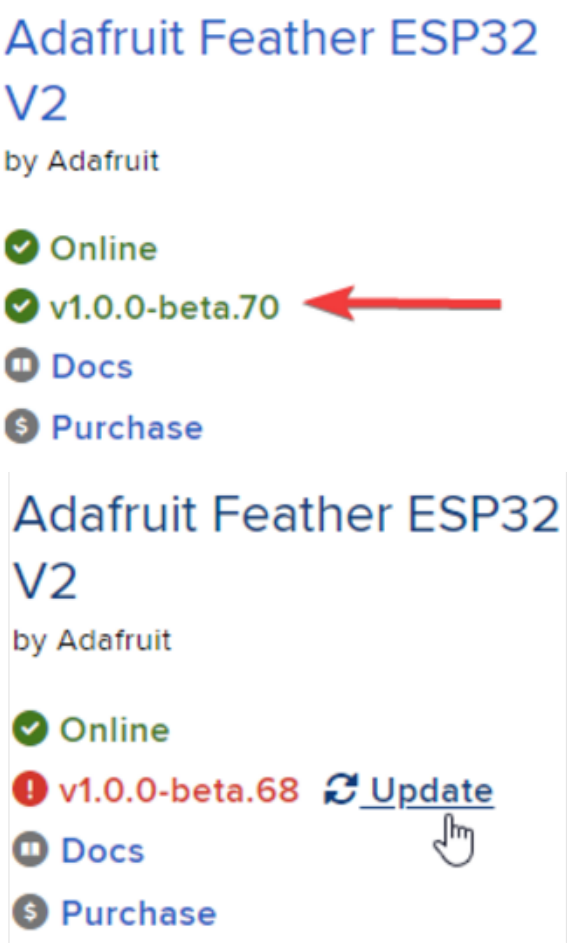
## Usage

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

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.

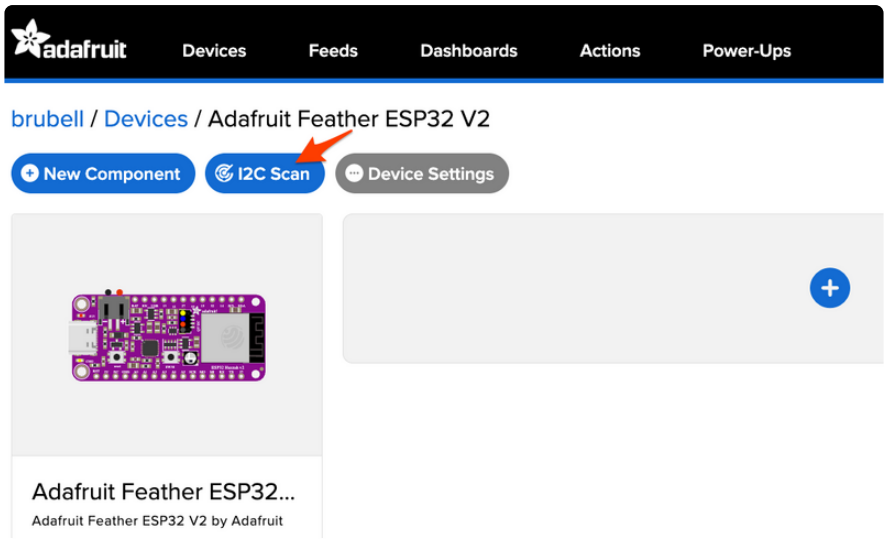



On the device page, quickly **check that you're running the latest version of the WipperSnapper firmware**.

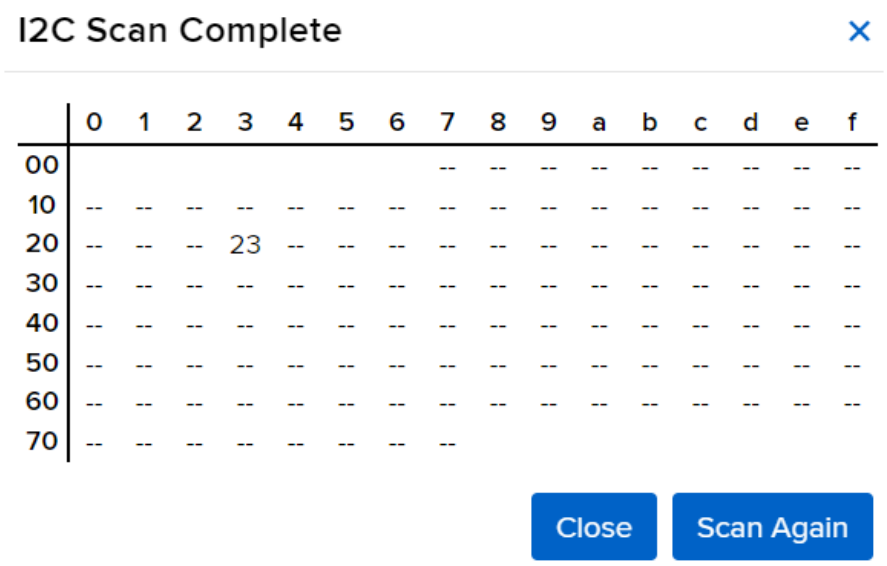
The device tile on the left indicates the version number of the firmware running on the connected board.

**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 BH1750's default I2C address of `0x23` pop-up in the I2C scan list. If the jumper on the underside of the board is soldered, this address will be changed to `0x5c`.




---

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

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

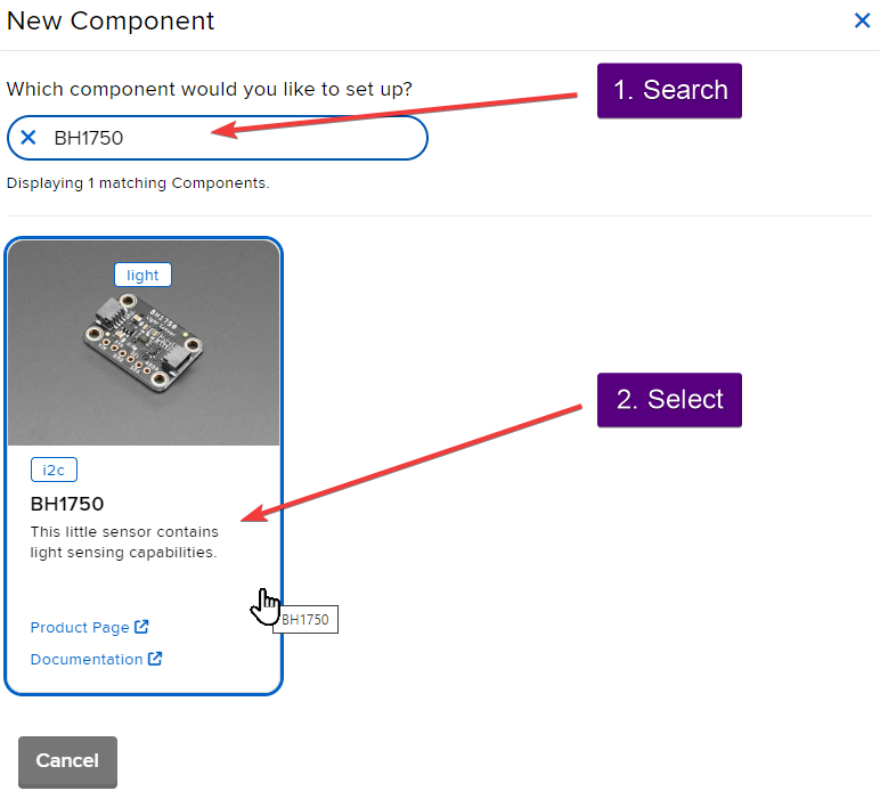
Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

---

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.

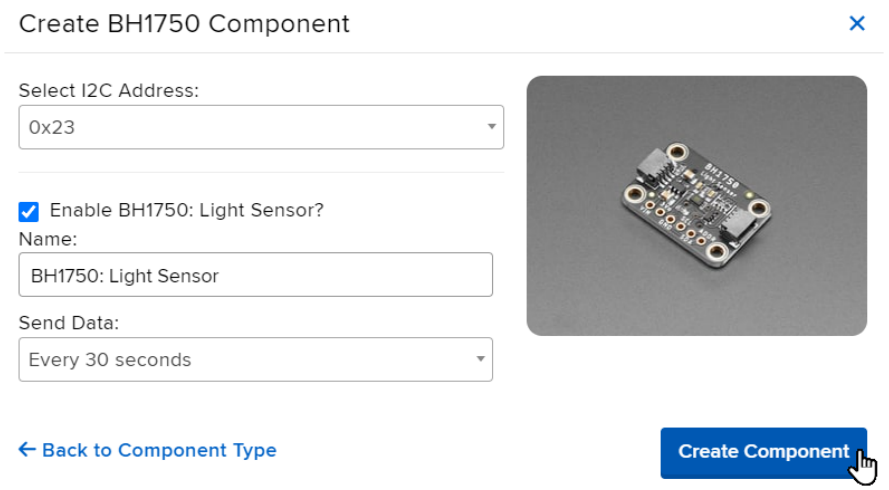

Adafruit IO supports a large amount of components. To quickly find your sensor, type `BH1750` into the search bar, then select the **BH1750** component.



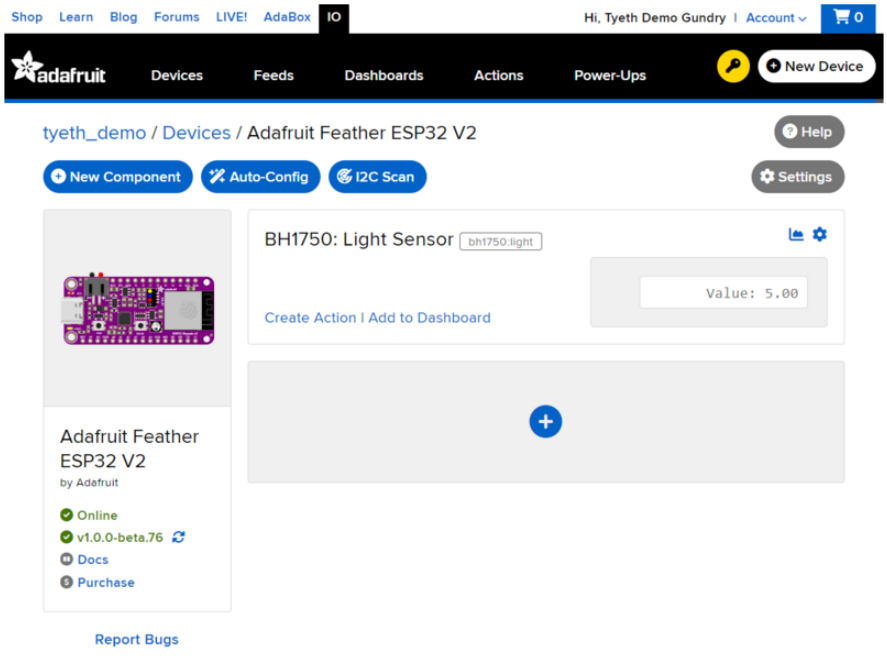

On the component configuration page, the BH1750's sensor address should be listed along with the sensor's settings.

The **Send Every** option is specific to each sensor's measurements. This option will tell the Feather how often it should read from the BH1750 sensor and send the data to Adafruit IO. Measurements can range from every 30 seconds to every 24 hours.

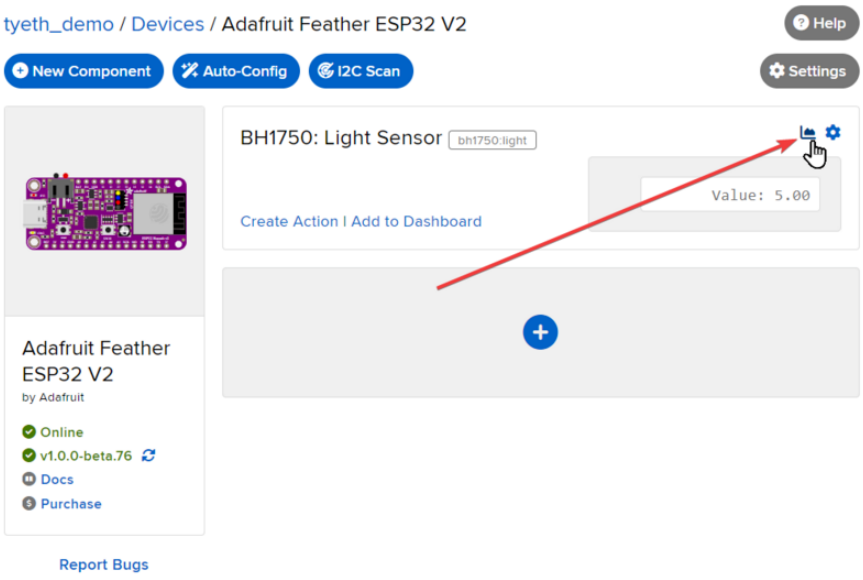
For this example, set the **Send Every** interval to every 30 seconds.



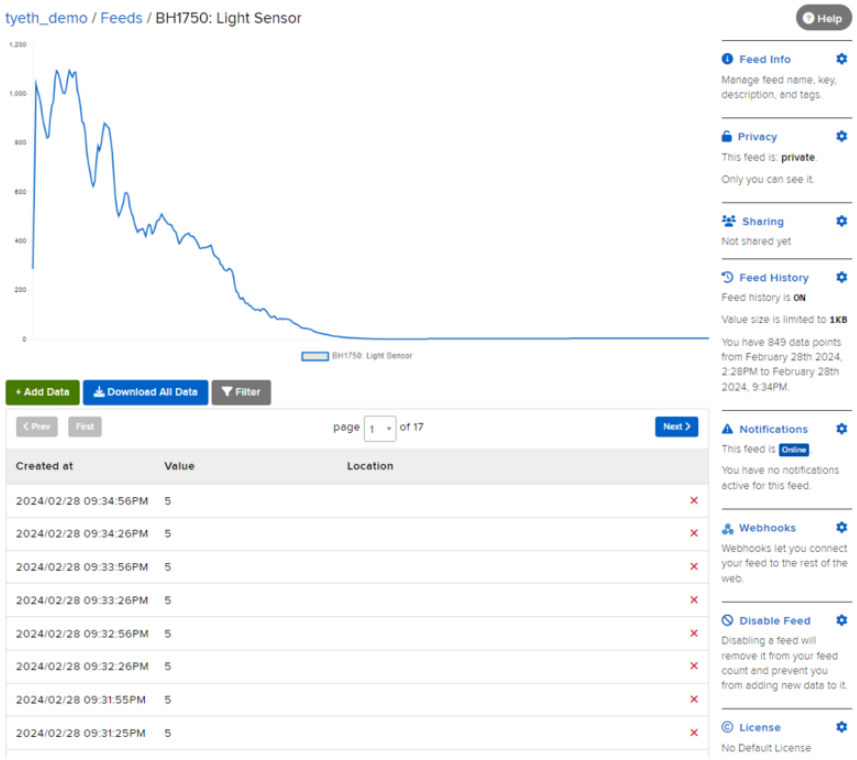

Your device interface should now show the sensor components you created. After the interval you configured elapses, WipperSnapper will automatically read values from the sensor(s) and send them to Adafruit IO.




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).



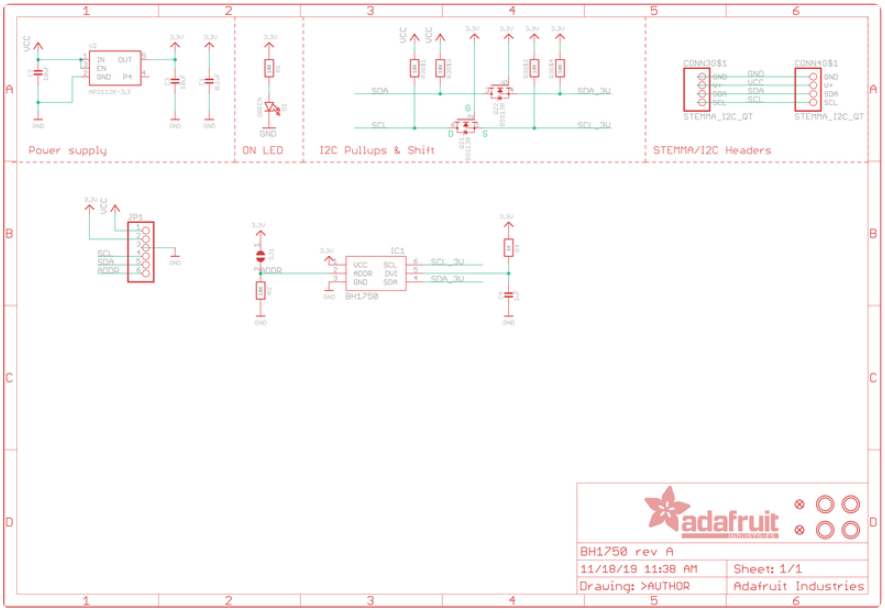

# Downloads

## Files

- BH1750 Datasheet (https://adafru.it/MfK)
- EagleCAD files on GitHub (https://adafru.it/MfL)

- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/T9E)

## Schematic




## Fab Print