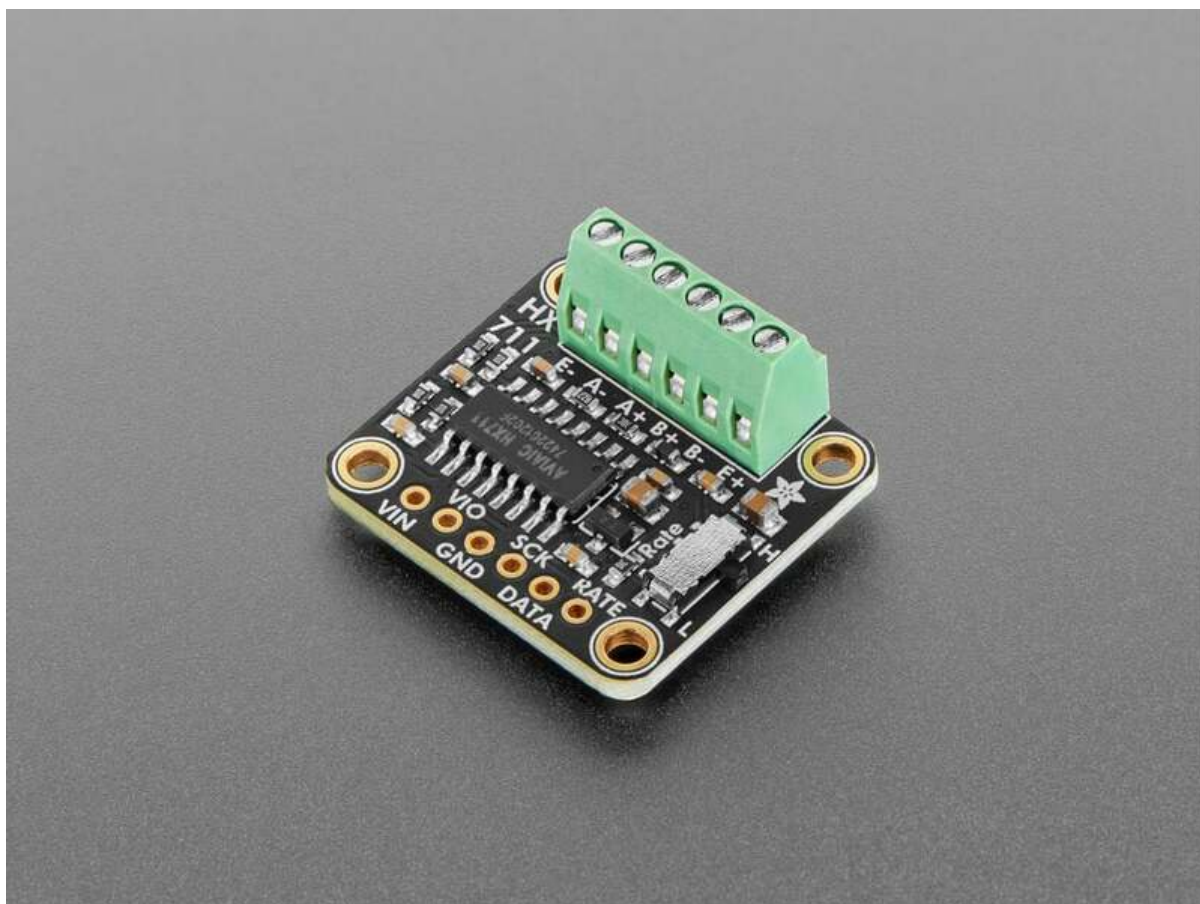




# Adafruit HX711 24-bit ADC

Created by Liz Clark



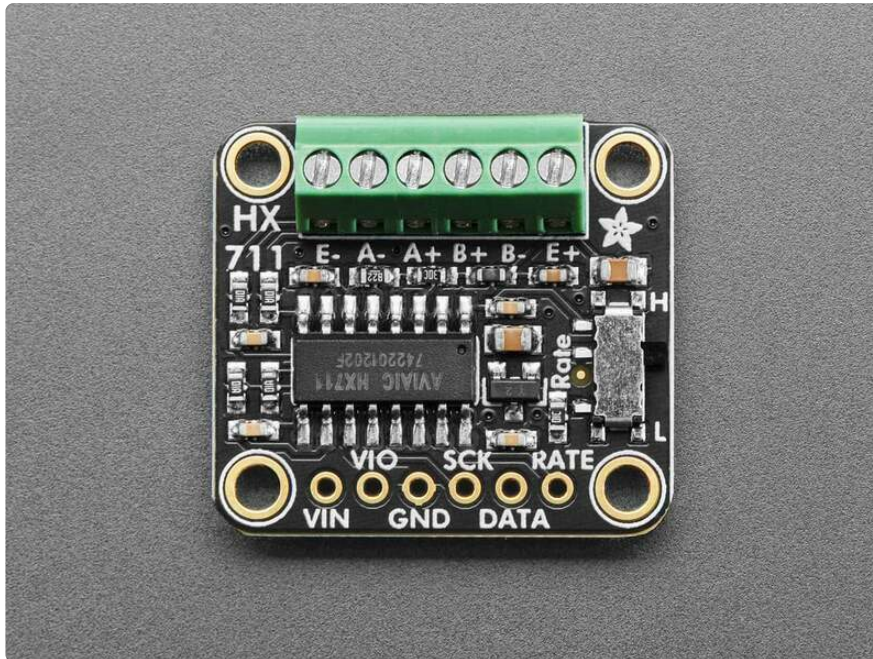
<https://learn.adafruit.com/adafruit-hx711-24-bit-adc>

Last updated on 2024-06-25 02:34:13 PM EDT

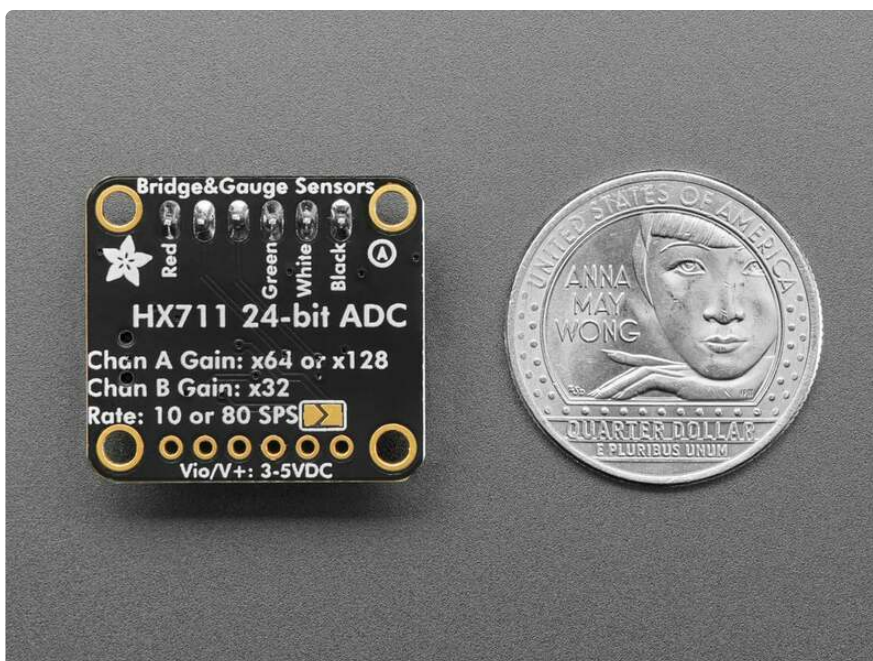
# Table of Contents

Overview	3
Pinouts	5
<ul style="list-style-type: none"><li>• Power Pins</li><li>• Terminal Block Pins</li><li>• HX711 Logic Pins</li><li>• Rate Selection Switch</li><li>• VIO Jumper</li></ul>	
CircuitPython	6
<ul style="list-style-type: none"><li>• CircuitPython Microcontroller Wiring</li><li>• CircuitPython Usage</li><li>• Example Code</li></ul>	
Python Docs	9
Arduino	9
<ul style="list-style-type: none"><li>• Wiring</li><li>• Library Installation</li><li>• Example Code</li></ul>	
Arduino Docs	12
Downloads	12
<ul style="list-style-type: none"><li>• Files</li><li>• Schematic and Fab Print</li></ul>	

# Overview



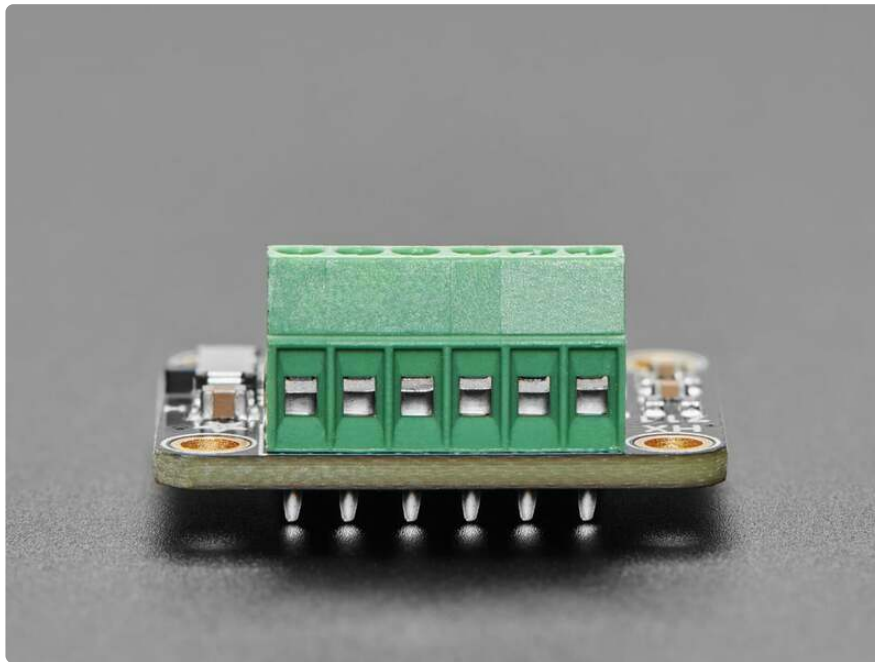
If you are feeling the stress and strain of a Wheatstone bridge and you want to quantify it, this handy breakout will do the job, no sweat! The **Adafruit HX711 Breakout** contains a super-high-resolution 24-Bit differential ADC with extra gain circuitry that makes it perfect for measuring strain gauges / load cells or other sensors that have four wires that are connected in a [Wheatstone bridge arrangement](https://adafru.it/-kd) (<https://adafru.it/-kd>).



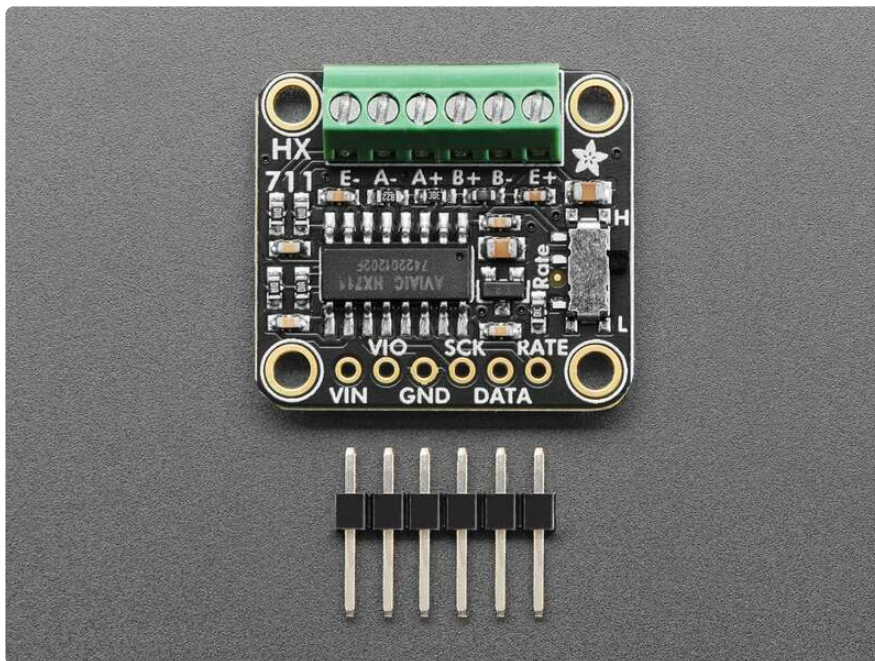
Each breakout comes with a HX711 ADC chip, plus some support circuitry, and 6 port terminal block that can be used to connect one or two 4-wire sensors. The **E-** pad



connects to ground (often a black wire), the **E+** pad connects to the power voltage supplied to the HX711 (often a red wire).



Then **A-** and **A+** pads connect to the differential outputs from the bridge. For example, connecting to a strain gauge these tend to be the white and green wires. If you have a second gauge or voltage you want to measure, you can use the **B-** and **B+** pads. On the A channel, you can select 64x or 128x gain. On the B channel, you have a fixed 32x gain.

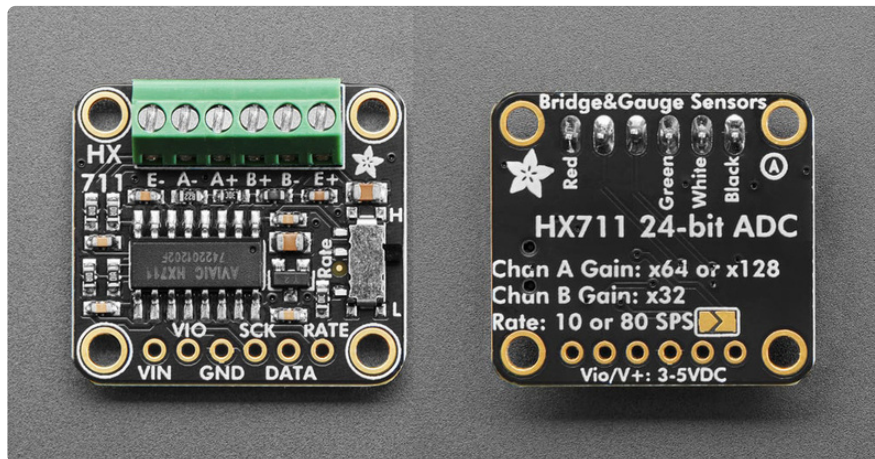


This sensor uses a two-wire protocol, similar to SPI. If you want a [similar sensor but I2C](http://adafru.it/4538), check out the [NAU7802 breakout](http://adafru.it/4538) (<http://adafru.it/4538>) we make. [We based this design on the excellent Sparkfun HX711 breakout](https://adafru.it/1a3i) (<https://adafru.it/1a3i>): incorporating

their split analog/digital supply and digital supply filtering. We then made a few tweaks: pre-soldered terminal blocks, the second bridge exposed, and a slide switch to change between 10 SPS and 80 SPS rates.

---

## Pinouts



## Power Pins

- **VIN** - 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.
- **VIO** - Output from the on-chip power supply regulator for the digital power supply.
- **GND** - Common ground for power and logic.

## Terminal Block Pins

At the top edge of the board is a 6 pin terminal block. This block has inputs for channels A and B on the HX711 (A+, A- and B+, B-) and power and ground connections for a strain gauge (E+ and E-).

- **A+** - Channel A positive input. You'll connect the green wire on your strain gauge to this block.
- **A-** - Channel A negative input. You'll connect the white wire on your strain gauge to this block.
- **B+** - Channel B positive input. This is a secondary input for the HX711.
- **B-** - Channel B negative input. This is a secondary input for the HX711.
- **E+** - Connected to the analog power output. You'll connect the red wire on your strain gauge to this block.

- **E** - Connected to common ground. You'll connect the black wire on your strain gauge to this block.

## HX711 Logic Pins

The HX711 uses "pseudo-SPI" to communicate. You can use any two digital pins on your microcontroller for clock and data.

- **SCK** - Serial clock input.
- **DATA** - Serial data output.
- **RATE** - Output data rate control. If this pin is low, the data rate is 10 SPS and if the pin is high, the data rate is 80 SPS.

## Rate Selection Switch

- **RATE Switch** - On the right side of the board is a slide switch. It is labeled **Rate** on the board silk. This switch selects the data rate for the HX711. If the switch selects **H**, then the data rate is 80 SPS. If the switch selects **L**, then the data rate is 10 SPS.

## VIO Jumper

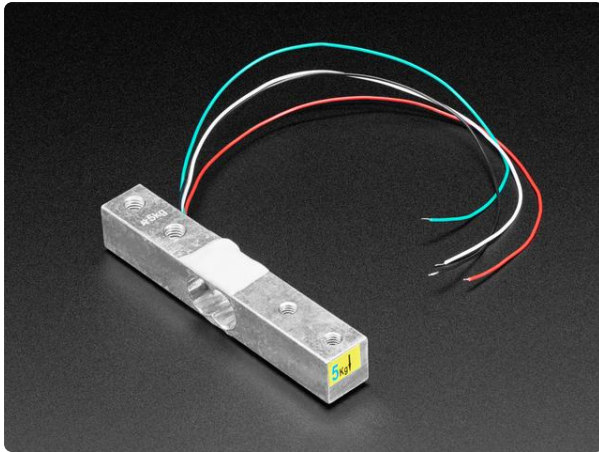
On the back of the board is an open solder jumper. It is outlined in white on the board silk. If you solder this jumper closed, it will connect **VIN** to **VIO**, which is the digital power supply (DVDD) pin on the HX711.

---

## CircuitPython

It's easy to use the **HX711** with Python or CircuitPython, and the [Adafruit\\_CircuitPython\\_HX711 \(https://adafru.it/1a3j\)](https://adafru.it/1a3j) module. This module allows you to easily write Python code to read data from the ADC.

You'll need a strain gauge to use this example with the breakout:



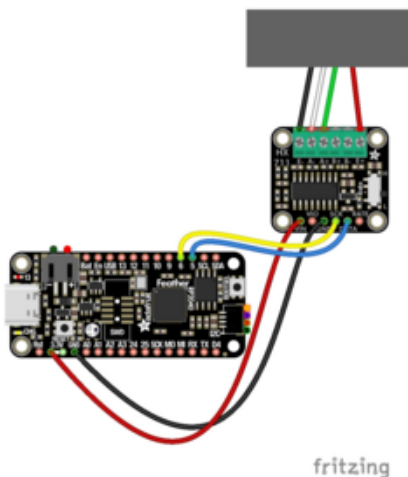
### Strain Gauge Load Cell - 4 Wires - 5Kg

A strain gauge is a type of electronic sensor used to measure force or strain (big surprise there). They are made of an insulating flexible backing with a metallic...

<https://www.adafruit.com/product/4541>

## CircuitPython Microcontroller Wiring

First, wire up a strain gauge to the breakout exactly as follows. Then, wire up the breakout to your board. The following is the breakout wired to a Feather RP2040:



- Strain gauge power to breakout E+ (red wire)
- Strain gauge ground to breakout E- (black wire)
- Strain gauge positive to breakout A+ (green wire)
- Strain gauge negative to breakout A- (white wire)
- Board 3V to breakout VIN (red wire)
- Board GND to breakout GND (black wire)
- Board D5 to breakout DATA (blue wire)
- Board D6 to breakout SCK (yellow wire)

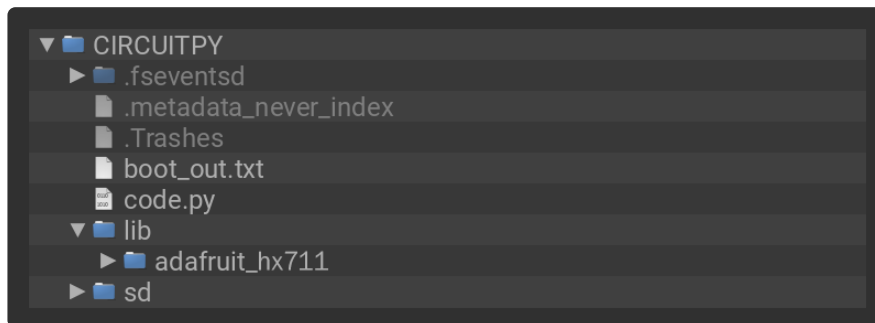
## CircuitPython Usage

To use with CircuitPython, you need to first install the **Adafruit\_CircuitPython\_HX711** library 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:

- /adafruit\_hx711



## Example Code

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!

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

import time
import board
import digitalio
from adafruit_hx711.hx711 import HX711
from adafruit_hx711.analog_in import AnalogIn

data = digitalio.DigitalInOut(board.D5)
data.direction = digitalio.Direction.INPUT
clock = digitalio.DigitalInOut(board.D6)
clock.direction = digitalio.Direction.OUTPUT

hx711 = HX711(data, clock)
channel_a = AnalogIn(hx711, HX711.CHAN_A_GAIN_128)
# channel_b = AnalogIn(hx711, HX711.CHAN_B_GAIN_32)

while True:
    print(f"Reading: {channel_a.value}")
    time.sleep(1)
```

First, the HX711 is instantiated with its clock and data pins. In the loop, the A channel on the ADC is read and printed to the serial console. As you bend, twist and put pressure on the strain gauge, you'll see the values change.



```
CircuitPython REPL
Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
code.py output:
Reading: 56231
Reading: 56685
Reading: 55878
Reading: 75170
Reading: 82382
Reading: 86865
Reading: 59695
Reading: 55499
Reading: 55151
Reading: 81666
```

---

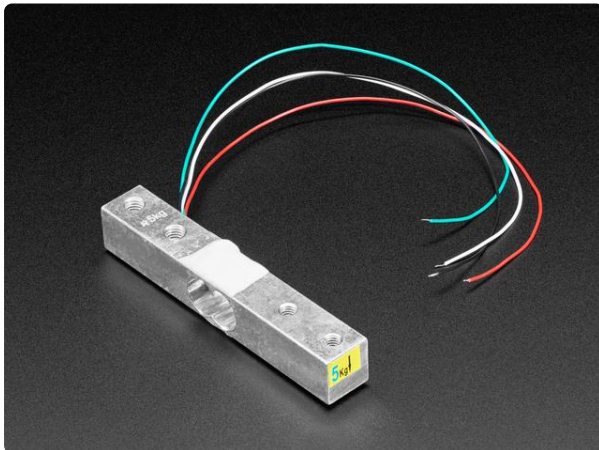
## Python Docs

[Python Docs \(https://adafru.it/1a3e\)](https://adafru.it/1a3e)

---

## Arduino

Using the HX711 breakout with Arduino involves wiring up the breakout to your Arduino-compatible microcontroller with a strain gauge, installing the [Adafruit\\_HX711 \(https://adafru.it/1a3k\)](https://adafru.it/1a3k) library, and running the provided example code.



### Strain Gauge Load Cell - 4 Wires - 5Kg

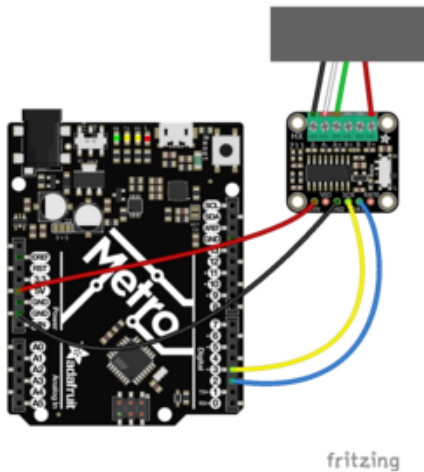
A strain gauge is a type of electronic sensor used to measure force or strain (big surprise there). They are made of an insulating flexible backing with a metallic...

<https://www.adafruit.com/product/4541>

## 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.

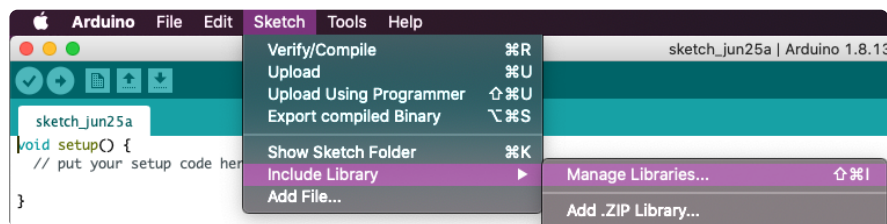
Here is an Adafruit Metro wired up to the breakout with a strain gauge:



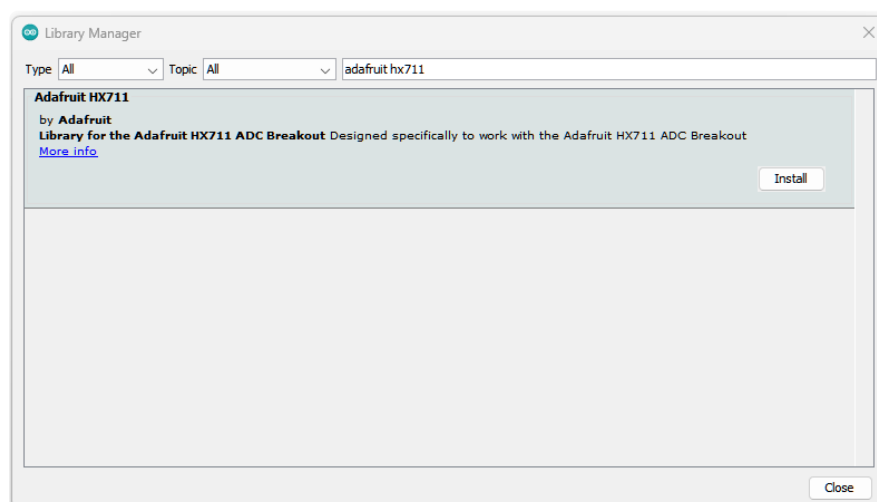
- Strain gauge power to breakout E+ (red wire)
- Strain gauge ground to breakout E- (black wire)
- Strain gauge positive to breakout A+ (green wire)
- Strain gauge negative to breakout A- (white wire)
- Board 5V to breakout VIN (red wire)
- Board GND to breakout GND (black wire)
- Board D2 to breakout DATA (blue wire)
- Board D3 to breakout SCK (yellow wire)

## Library Installation

You can install the **Adafruit\_HX711** library for Arduino using the Library Manager in the Arduino IDE.



Click the **Manage Libraries ...** menu item, search for **Adafruit\_HX711**, and select the **Adafruit HX711** library:



There are no additional dependencies for the library.

## Example Code

```
#include "Adafruit_HX711.h"

// Define the pins for the HX711 communication
const uint8_t DATA_PIN = 2; // Can use any pins!
const uint8_t CLOCK_PIN = 3; // Can use any pins!

Adafruit_HX711 hx711(DATA_PIN, CLOCK_PIN);

void setup() {
  Serial.begin(115200);

  // wait for serial port to connect. Needed for native USB port only
  while (!Serial) {
    delay(10);
  }

  Serial.println("Adafruit HX711 Test!");

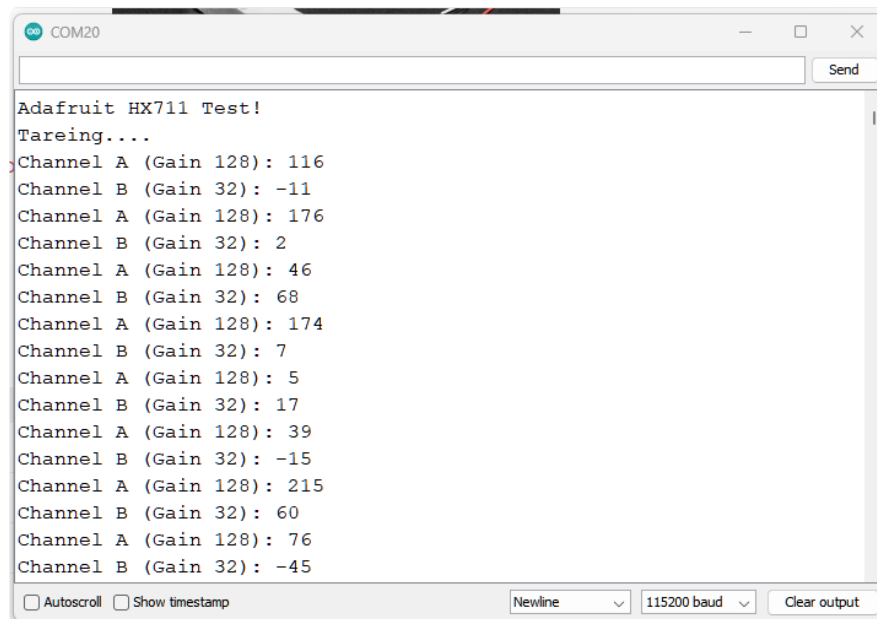
  // Initialize the HX711
  hx711.begin();

  // read and toss 3 values each
  Serial.println("Tareing...");
  for (uint8_t t=0; t<3; t++) {
    hx711.tareA(hx711.readChannelRaw(CHAN_A_GAIN_128));
    hx711.tareA(hx711.readChannelRaw(CHAN_A_GAIN_128));
    hx711.tareB(hx711.readChannelRaw(CHAN_B_GAIN_32));
    hx711.tareB(hx711.readChannelRaw(CHAN_B_GAIN_32));
  }
}

void loop() {
  // Read from Channel A with Gain 128, can also try CHAN_A_GAIN_64 or
  CHAN_B_GAIN_32
  // since the read is blocking this will not be more than 10 or 80 SPS (L or H
  switch)
  int32_t weightA128 = hx711.readChannelBlocking(CHAN_A_GAIN_128);
  Serial.print("Channel A (Gain 128): ");
  Serial.println(weightA128);

  // Read from Channel A with Gain 128, can also try CHAN_A_GAIN_64 or
  CHAN_B_GAIN_32
  int32_t weightB32 = hx711.readChannelBlocking(CHAN_B_GAIN_32);
  Serial.print("Channel B (Gain 32): ");
  Serial.println(weightB32);
}
```

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. In the loop, the readings on channel A and channel B are printed out. As you bend, twist and put pressure on the strain gauge, you'll see the values change.



```
COM20
Send

Adafruit HX711 Test!
Tareing...
Channel A (Gain 128): 116
Channel B (Gain 32): -11
Channel A (Gain 128): 176
Channel B (Gain 32): 2
Channel A (Gain 128): 46
Channel B (Gain 32): 68
Channel A (Gain 128): 174
Channel B (Gain 32): 7
Channel A (Gain 128): 5
Channel B (Gain 32): 17
Channel A (Gain 128): 39
Channel B (Gain 32): -15
Channel A (Gain 128): 215
Channel B (Gain 32): 60
Channel A (Gain 128): 76
Channel B (Gain 32): -45

☐ Autoscroll ☐ Show timestamp Newline 115200 baud Clear output
```

---

## Arduino Docs

[Arduino Docs \(https://adafru.it/1a3f\)](https://adafru.it/1a3f)

---

## Downloads

### Files

- [HX711 Datasheet \(https://adafru.it/1a3l\)](https://adafru.it/1a3l)
- [EagleCAD PCB Files on GitHub \(https://adafru.it/1a3m\)](https://adafru.it/1a3m)
- [Fritzing object in the Adafruit Fritzing Library \(https://adafru.it/1a3n\)](https://adafru.it/1a3n)



# Schematic and Fab Print

