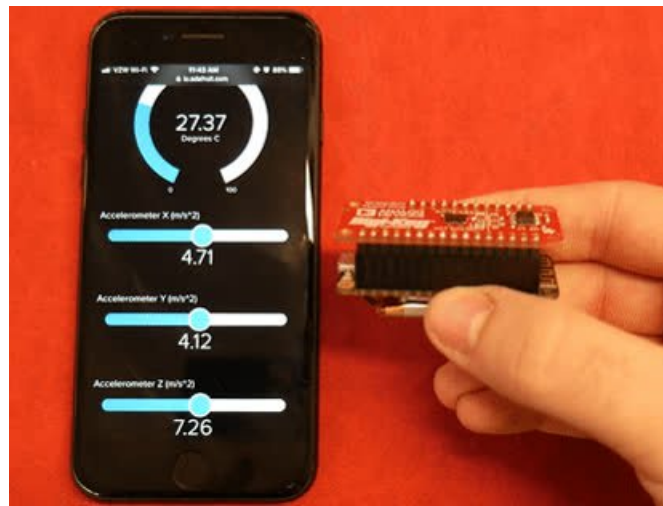


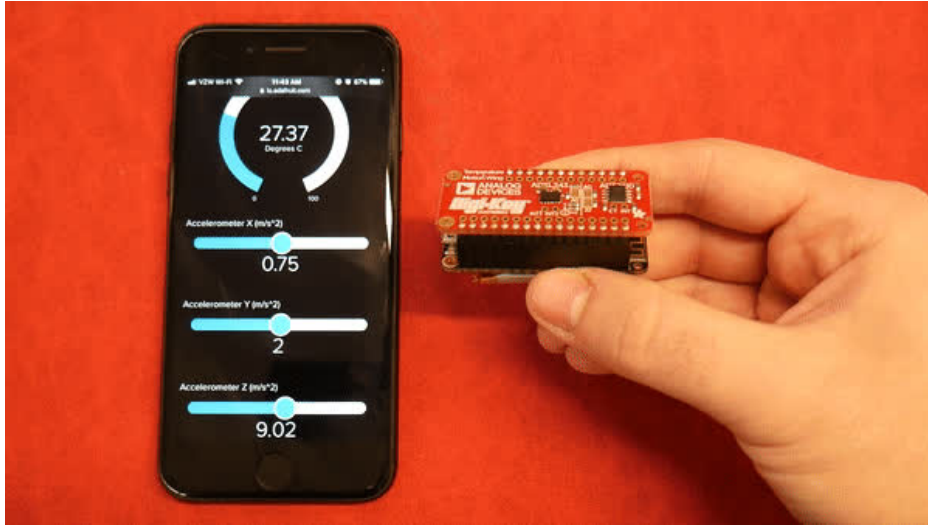
IoT Motion and Temperature Logger with the Analog Devices ADXL343 + ADT7410 Sensor FeatherWing and Adafruit IO

Created by Brent Rubell



Last updated on 2019-03-21 07:55:12 PM UTC

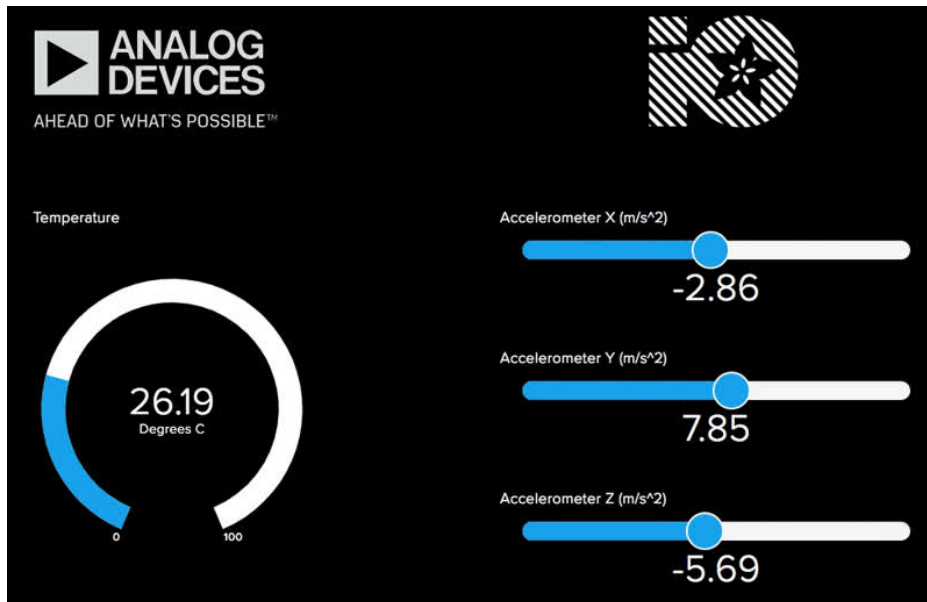
Overview



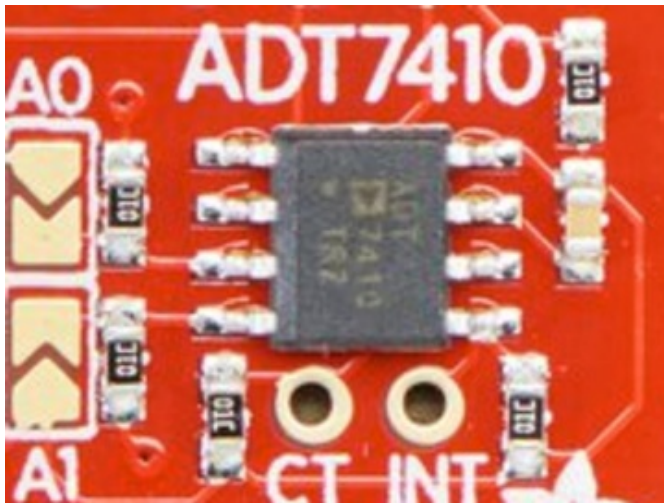
Upgrade any Feather board with motion and precision temperature sensing, with the Adafruit Sensor Featherwing. It sports two fantastic sensors from [Analog Devices \(https://adafru.it/DPF\)](https://adafru.it/DPF): an [ADXL343 triple-axis accelerometer \(https://adafru.it/Elo\)](https://adafru.it/Elo) and an [ADT7410 precision temperature sensor \(https://adafru.it/EgN\)](https://adafru.it/EgN).

This FeatherWing can easily be brought *online* using a Feather ESP8266 Huzzah and [Adafruit \(https://adafru.it/eIC\)](https://adafru.it/eIC) IO (<https://adafru.it/eIC>) - you'll be up and running *in under 15 minutes!*

If you're looking for a way to monitor and send precise motion and temperature data to the cloud, follow along!



Thanks to [Digi-Key \(https://adafru.it/BJr\)](https://adafru.it/BJr) and [Analog Devices \(https://adafru.it/DPF\)](https://adafru.it/DPF) for sponsoring the development of this breakout board - we've made the PCB "[Digi-Key red \(https://adafru.it/BJr\)](https://adafru.it/BJr)" in their honor!



You'll be using the **Analog Devices ADT7410** built into the Temperature + Motion Wing to measure the ambient temperature over I2C.



You'll also be using the **Analog Devices ADXL343** triple-axis accelerometer to monitor motion over I2C. The Analog Devices ADXL343 has three axes of measurements: X, Y, and Z. You can set the sensitivity level to either +2g, +4g, +8g or +16g. The lower range gives more resolution for slow movements, the higher range is good for high speed tracking.



Adafruit IO

Adafruit IO is the easiest way to stream, log, and interact with your data. It's built from the ground up to be easy to use - we do the hard stuff so you can focus on the fun stuff.

Data such as temperature and motion can be hard to visualize and quantify - Adafruit IO makes it simple. Send IO your data and it can store and display it using charts, graphs, gauges, and more!

```
adafruit-io-adt7410-adx343.ino | Arduino 1.8.8
adafruit-io-adt7410-adx343.ino
#define ID_KEY "YOUR_ID_KEY"

//***** WiFi Config *****/
#define WIFI_SSID "WIFI_NAME"
#define WIFI_PASS "WIFI_PASS"

// comment out the following two lines if you are using fona or ethernet
#include "AdafruitIO_Wifi.h"
AdafruitIO_Wifi io(ID_USERNAME, ID_KEY, WIFI_SSID, WIFI_PASS);
//***** Example Starts Here *****/
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADT7410.h>
#include <Adafruit_ADXL343.h>

float tempC, accelX, accelY, accelZ;

// Create the ADT7410 temperature sensor object
Adafruit_ADT7410 tempSensor = Adafruit_ADT7410(C);

// Create the ADXL343 accelerometer sensor object
Adafruit_ADXL343 accel = Adafruit_ADXL343(12345);

// set up the 'temperature' feed
AdafruitIO_Feed "huzzah_temperature" = io.feed("temperature");

// set up the 'accelX' feed
AdafruitIO_Feed "huzzah_accel_x" = io.feed("accelX");

// set up the 'accelY' feed
AdafruitIO_Feed "huzzah_accel_y" = io.feed("accelY");
```

Arduino Code and Libraries

Bring the Adafruit Sensor FeatherWing to life using Arduino libraries for both sensors on the FeatherWing - the [ADXL343 \(https://adafru.it/E5T\)](https://adafru.it/E5T) and the [ADT7410 \(https://adafru.it/DPy\)](https://adafru.it/DPy).

To rapidly connect the Feather Huzzah to the internet, you'll be using use the [Adafruit IO Arduino library \(https://adafru.it/fpd\)](https://adafru.it/fpd).

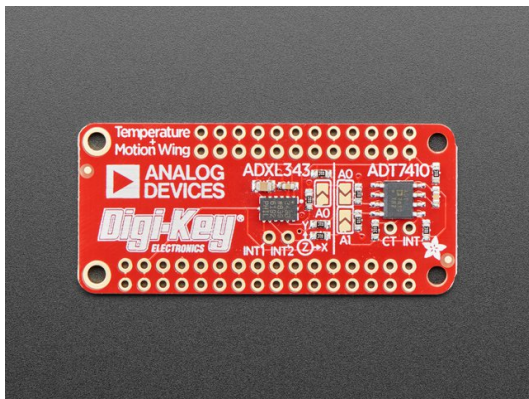
Prerequisite Guides

If you're new to Adafruit IO, take a moment to walk through the following guides to get you started and up-to-speed:

- [Welcome to Adafruit IO \(https://adafru.it/DZd\)](https://adafru.it/DZd)

Parts

You're going to be working with the Adafruit ADXL343 + ADT7410 Sensor FeatherWing. *There's no wiring required* - connect the FeatherWing to your HUZDAH and you can start working with the FeatherWing *immediately!*

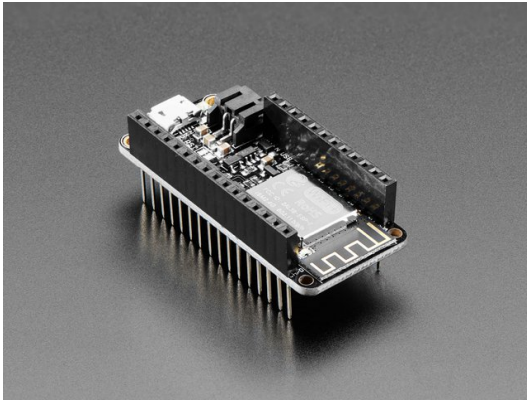


Adafruit ADXL343 + ADT7410 Sensor FeatherWing

\$11.95
IN STOCK

ADD TO CART

The HUZDAH ESP8266 is the most popular board used for Adafruit IO projects. We suggest getting one which already has [stacking headers \(https://adafru.it/oAP\)](https://adafru.it/oAP) attached to easily connect to the FeatherWing.



Assembled Feather HUZAZH w/ ESP8266 WiFi With Stacking Headers

\$19.95
IN STOCK

ADD TO CART

Materials

You'll need some extra supplies to finish this project. If you do not have them already, pick some up from Adafruit:

1x [USB Cable](#)

USB cable - USB A to Micro-B - 3 foot long

ADD TO CART

1x [Feather Stacking Headers](#)

Stacking Headers for Feather - 12-pin and 16-pin female headers

ADD TO CART

If you would like to power your IoT project wirelessly, you may want to pick up a LiPo battery, This one comes with a pre-attached 2-pin JST-PH connect so you can easily connect it to the JST on the Feather HUZAZH.

1x [Lithium Ion Battery](#)

Lithium Ion Polymer Battery Ideal For Feathers - 3.7V 400mAh

ADD TO CART

1x [Micro LiPo Charger](#)

Adafruit Micro Lipo - USB Lilon/LiPoly charger - v1

ADD TO CART

Adafruit IO Setup

Feed Setup

If you do not already have an Adafruit IO account set up, head over to [io.adafruit.com](https://adafruit.com) (<https://adafru.it/fH9>) to link your Adafruit.com account to Adafruit IO.

The first step is to create a new Adafruit IO feed to hold the AD7410's temperature. Navigate to the [feeds page](https://adafru.it/mxC) (<https://adafru.it/mxC>) on Adafruit IO. Then click **Actions** -> **Create New Feed**, and name this feed **temperature**.

- If you do not already know how to create a feed, head over to [Adafruit IO Basics: Feeds](https://adafru.it/ioA) (<https://adafru.it/ioA>).

Create a new Feed ✕

Name

Description

Add to groups

Cancel

Create

You'll also need to **create three more feeds to hold the data for the ADXL343 accelerometer - *accelX*, *accelY*, *accelZ***.

Once you have four feeds created, move to setting up a beautiful dashboard.

Dashboard Setup

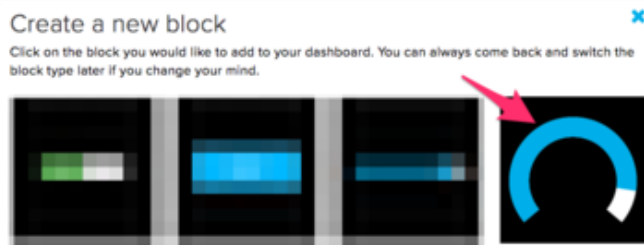
The next step is to create a dashboard to display the values read by the Motion + Temperature FeatherWing.

- If you do not know how to create or use Dashboards in Adafruit IO, head over to the [Adafruit IO Basics: Dashboards](https://adafru.it/f5m) (<https://adafru.it/f5m>) guide.

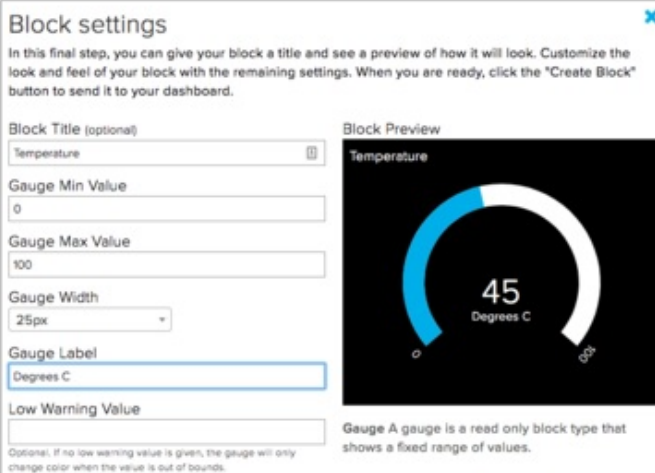
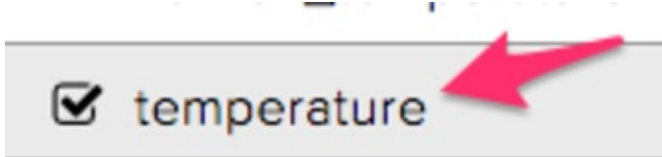
Add a Gauge Block

The gauge block allows you to quickly view the current value of a numeric feed. You can set a minimum and maximum value for the gauge, and it will automatically scale the value to a percentage and display it graphically. The gauge will update automatically whenever a new value is pushed to the feed.

Select the Gauge Block



Select the temperature feed you created earlier.



In the Block Settings step, set the **Block Title** to **Temperature**, set the **Gauge Min/Max Values** to the **upper and lower temperature thresholds** you want to measure.

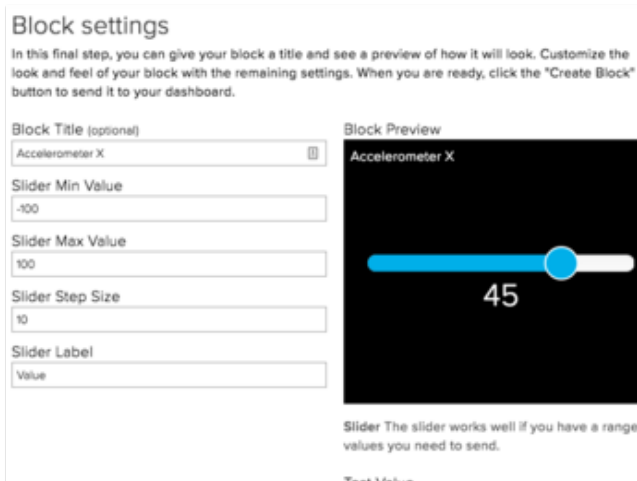
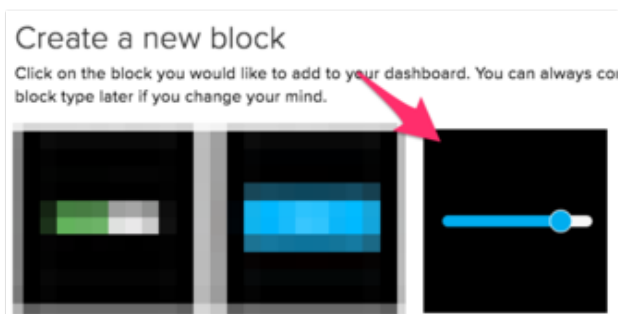
You can label the gauge by setting the Gauge Label - this example assumes temperature is to be measured in Degrees C.

Uncomfortably hot/cold? You can optionally set the gauge change color to warn you if the temperature goes above (or below) a certain value.

Add a Number Slider Block

The number slider allows you to quickly change the numeric value of a feed. You can set the minimum and maximum values for the slider, as well as change the amount the slider will increment when you drag the handle.

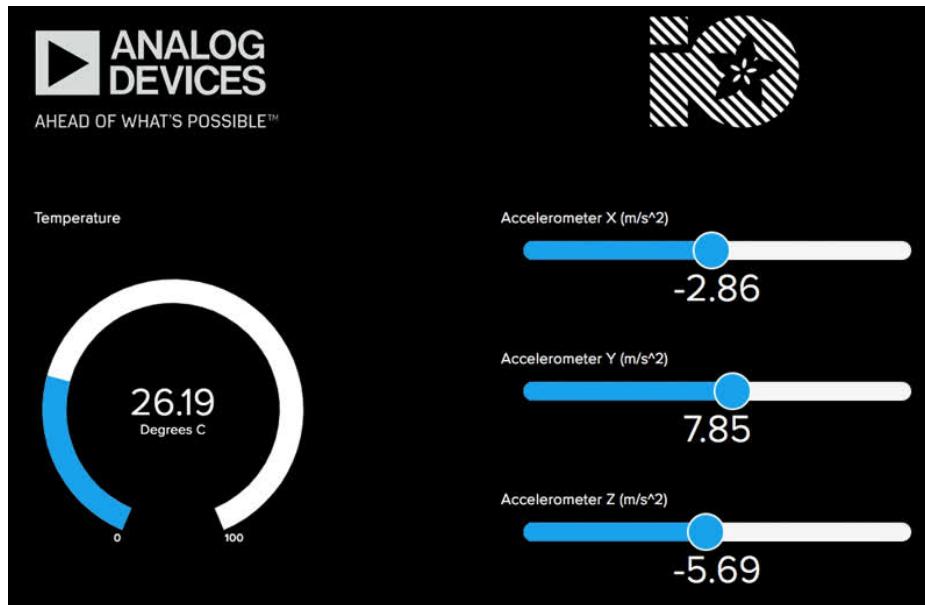
Create a new number slider block and select the *accelX* feed you created earlier.



Name the block **Accelerometer X**. From here, you can configure the minimum and the maximum values displayed by the slider.

Next, add two more sliders to the dashboard. These sliders will be linked to the *Acce/Y* and *Acce/Z* feeds you created earlier.

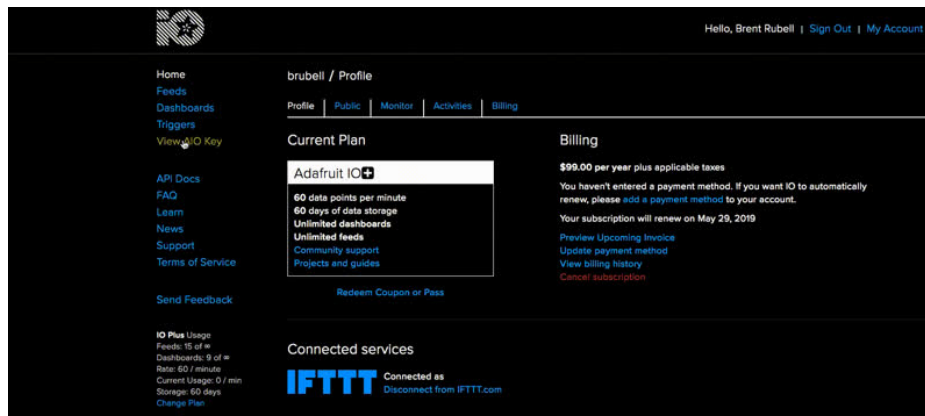
Once you're finished adding the sliders, your dashboard should look like the following screenshot:



Obtain your Adafruit IO Key

You are also going to need your **Adafruit IO** username and secret API key.

Navigate to your profile and click the View AIO Key button to retrieve them. Write them down in a safe place, you'll need them for the next step.



Next, we'll move on to programming the Feather and using it with Adafruit IO.

Arduino Code

Installation

To read data from the Motion and Temperature FeatherWing, you'll need to **install libraries for both of the ADXL343 and the ADT7410 sensors.**

To install the library for the ADXL343, visit this page (<https://adafru.it/Elp>) and come back here when you have the library installed and tested.

Next, **to install the library for the ADT7410, visit this page (<https://adafru.it/DSE>)** and come back here when you have the library installed and tested.

This guide assumes you've completed the setup required to get your Adafruit Feather Huzzah ESP8266 up and running with Arduino IDE and Adafruit IO.

- If you haven't yet set up your Adafruit Feather Huzzah ESP8266 for use with Adafruit IO and the Arduino IDE, **follow along with this guide (<https://adafru.it/DCI>)**. The setup only needs to be performed once.

Code

The code for this project is below. Copy and paste it into the Arduino IDE editor.

```
// Adafruit IO - Analog Devices ADT7410 + ADXL343 Example
//
// Adafruit invests time and resources providing this open source code.
// Please support Adafruit and open source hardware by purchasing
// products from Adafruit!
//
// Written by Brent Rubell for Adafruit Industries
// Copyright (c) 2019 Adafruit Industries
// Licensed under the MIT license.
//
// All text above must be included in any redistribution.

/***** Adafruit IO Config *****/
// visit io.adafruit.com if you need to create an account,
// or if you need your Adafruit IO key.
#define IO_USERNAME "YOUR_IO_USERNAME"
#define IO_KEY "YOUR_IO_KEY"

/***** WiFi Config *****/
#define WIFI_SSID "WIFI_NAME"
#define WIFI_PASS "WIFI_PASS"

// comment out the following two lines if you are using fona or ethernet
#include "AdafruitIO_WiFi.h"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);

/***** Configuration *****/
// time between sending data to adafruit io, in seconds.
#define IO_DELAY 5
/***** Example Starts Here *****/
#include <Wire.h>
#include <Adafruit_Sensor.h>
```

```

#include "Adafruit_ADT7410.h"
#include <Adafruit_ADXL343.h>

float tempC, accelX, accelY, accelZ;

// Create the ADT7410 temperature sensor object
Adafruit_ADT7410 tempsensor = Adafruit_ADT7410();

// Create the ADXL343 accelerometer sensor object
Adafruit_ADXL343 accel = Adafruit_ADXL343(12345);

// set up the 'temperature' feed
AdafruitIO_Feed *huzzah_temperature = io.feed("temperature");

// set up the 'accelX' feed
AdafruitIO_Feed *huzzah_accel_x = io.feed("accelX");

// set up the 'accelY' feed
AdafruitIO_Feed *huzzah_accel_y = io.feed("accelY");

// set up the 'accelZ' feed
AdafruitIO_Feed *huzzah_accel_z= io.feed("accelZ");

void setup()
{
  // start the serial connection
  Serial.begin(115200);

  // wait for serial monitor to open
  while (!Serial)
    ;

  Serial.println("Adafruit IO - ADT7410 + ADX343");

  /* Initialise the ADXL343 */
  if(!accel.begin())
  {
    /* There was a problem detecting the ADXL343 ... check your connections */
    Serial.println("Ooops, no ADXL343 detected ... Check your wiring!");
    while(1);
  }

  /* Set the range to whatever is appropriate for your project */
  accel.setRange(ADXL343_RANGE_16_G);

  /* Initialise the ADT7410 */
  if (!tempsensor.begin())
  {
    Serial.println("Couldn't find ADT7410!");
    while (1)
      ;
  }

  // sensor takes 250 ms to get first readings
  delay(250);

  // connect to io.adafruit.com
  Serial.print("Connecting to Adafruit IO");
  io.connect();
}

```

```

// wait for a connection
while (io.status() < AIO_CONNECTED)
{
  Serial.print(".");
  delay(500);
}

// we are connected
Serial.println();
Serial.println(io.statusText());
}

void loop()
{
  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

  /* Get a new accel. sensor event */
  sensors_event_t event;
  accel.getEvent(&event);

  accelX = event.acceleration.x;
  accelY = event.acceleration.y;
  accelZ = event.acceleration.z;

  /* Display the results (acceleration is measured in m/s^2) */
  Serial.print("X: "); Serial.print(accelX); Serial.print(" ");
  Serial.print("Y: "); Serial.print(accelY); Serial.print(" ");
  Serial.print("Z: "); Serial.print(accelZ); Serial.print(" ");Serial.println("m/s^2 ");

  // Read and print out the temperature
  tempC = tempsensor.readTempC();
  Serial.print("Temperature: "); Serial.print(tempC); Serial.println("C");

  Serial.println("Sending to Adafruit IO...");
  huzzah_temperature->save(tempC, 0, 0, 0, 2);
  huzzah_accel_x->save(accelX);
  huzzah_accel_y->save(accelY);
  huzzah_accel_z->save(accelZ);
  Serial.println("Data sent!");

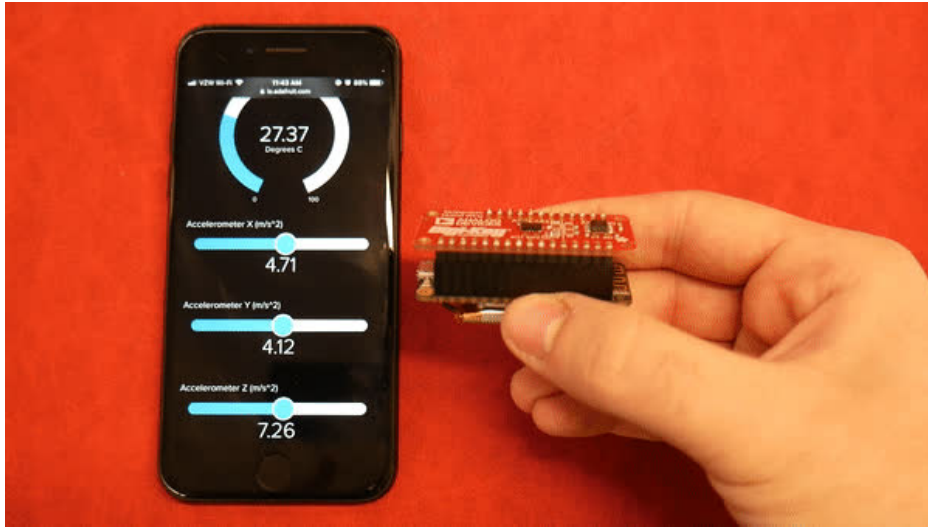
  Serial.print("Waiting ");Serial.print(IO_DELAY);Serial.println(" seconds...");
  // wait IO_DELAY seconds between sends
  for (int i = 0; i < IO_DELAY; i++)
  {
    delay(1000);
  }
}

```

Code Setup

Before you upload the sketch to the Huzzah, you'll need to configure the code for your network and Adafruit IO.

Change **IO_USERNAME** to your Adafruit IO username.



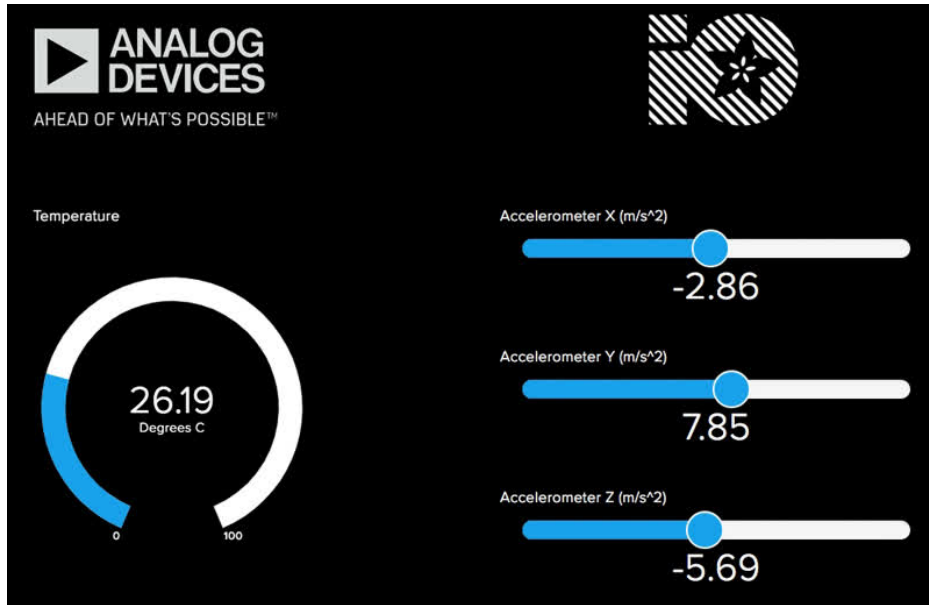
Code Usage

Let's check that the data has been received by Adafruit IO. You can do this by visiting the [Adafruit IO Monitor page \(https://adafru.it/DOK\)](https://adafru.it/DOK). Every time the Huzzah sends data to Adafruit IO, this page shows the `temperature`, `accelX`, `accelY` or `accelZ` values.

```

Live Data newest data at the top
2019/03/21 11:48am accelx -2.510502
2019/03/21 11:48am temperature 25.06
2019/03/21 11:48am accelz 8.551399
2019/03/21 11:48am accely -3.255808
2019/03/21 11:48am accelz 8.551399
2019/03/21 11:48am accelx -2.588956
2019/03/21 11:48am temperature 25.06
2019/03/21 11:48am accelz 8.472946
2019/03/21 11:48am accely -3.177355
2019/03/21 11:48am accelx -2.510502
  
```

Navigate to the dashboard you created earlier. You'll notice the Gauge block change values whenever the ADT7410 reports a new temperature to the Huzzah. As you tilt the Huzzah, you'll notice the sliders for X, Y, and Z changing their positions to reflect the ADXL343's values in m/s^2 .



Taking it Further

Data Visualization

While the sliders can provide immediate information about the ADXL343 or the ADT7410 on the FeatherWing at a glance - you may want to chart your data over a longer period of time or see a list of data with previous values on the dashboard.

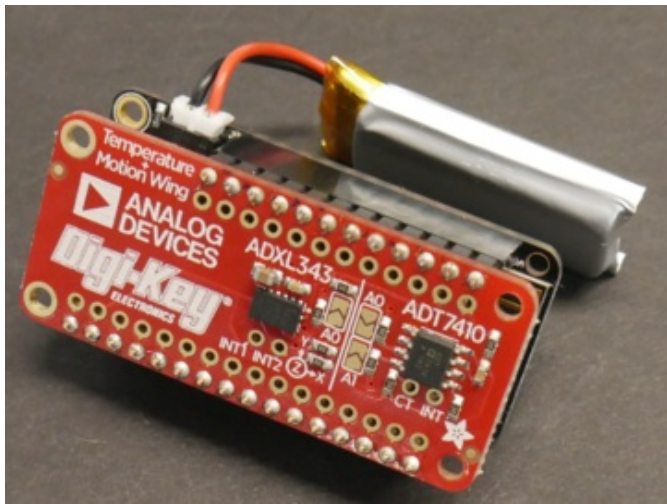


Connect a [Line Graph Block \(https://adafru.it/DZe\)](https://adafru.it/DZe) to the the four feeds to view data from the sensors over any period of time - from real-time to 30 days.


```
ADXL343
2019/03/21 11:11am AccelZ 9.296704
2019/03/21 11:11am accelX -1.019892
2019/03/21 11:11am accelY -0.039227
2019/03/21 11:11am AccelZ 9.375157
2019/03/21 11:11am accelX -0.980665
2019/03/21 11:11am accelY -0.039227
2019/03/21 11:11am AccelZ 9.296704
2019/03/21 11:12am accelX -1.019892
2019/03/21 11:12am accelY -0.039227
2019/03/21 11:12am AccelZ 9.375157
2019/03/21 11:12am accelX -1.019892
2019/03/21 11:12am accelY -0.039227
2019/03/21 11:12am AccelZ 9.335931
2019/03/21 11:12am accelX -1.019892
```

Then, add a [Stream Block \(https://adafru.it/DZe\)](https://adafru.it/DZe) and connect to any feed which you'd like to monitor. This block is a miniature version of the monitor page, but only displays feeds which you specify.

Adding a Battery



Want to run your Feather without an attached USB Cable?

For 100% wireless motion and temperature logging, pick up a Lithium Ion Polymer battery ([we recommend this one \(https://adafru.it/D7i\)](https://adafru.it/D7i) - it's *ideal* for the Feather family) and connect it to the JST-PH port on the Feather ESP8266.