



# Wearable Continuous Temperature Monitor with Adafruit IO

Created by Brent Rubell



<https://learn.adafruit.com/wearable-temperature-monitor>

Last updated on 2024-06-03 03:05:46 PM EDT

# Table of Contents

<b>Overview</b>	<b>5</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Wearable Continuous Temperature Monitor</a></li><li>• <a href="#">Bluetooth Low Energy (BLE) Temperature Armband</a></li><li>• <a href="#">Interactive Temperature Dashboard</a></li><li>• <a href="#">Fever Email Alerts</a></li><li>• <a href="#">Hands-Free and Rechargeable</a></li><li>• <a href="#">About your Health Data and Adafruit IO</a></li><li>• <a href="#">About this Project</a></li><li>• <a href="#">Parts</a></li><li>• <a href="#">Tools</a></li><li>• <a href="#">Materials</a></li></ul>	
<b>Circuit Diagram</b>	<b>11</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Adafruit Library for Fritzing</a></li></ul>	
<b>3D Printing</b>	<b>12</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Parts</a></li><li>• <a href="#">Case Assembly</a></li><li>• <a href="#">Sensor Assembly</a></li><li>• <a href="#">Flexible Parts</a></li><li>• <a href="#">PLA Parts</a></li><li>• <a href="#">Design Source Files</a></li></ul>	
<b>Assembly</b>	<b>14</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Wiring Circuit</a></li><li>• <a href="#">Install Feather nRF52840</a></li><li>• <a href="#">Secure Feather</a></li><li>• <a href="#">Installed PCB</a></li><li>• <a href="#">Installing Switch</a></li><li>• <a href="#">Installed Switch</a></li><li>• <a href="#">Install Parts</a></li><li>• <a href="#">Install Flexible Strap</a></li><li>• <a href="#">Install MCP9808</a></li><li>• <a href="#">Installed MCP9808</a></li><li>• <a href="#">Install Sensor Cover</a></li><li>• <a href="#">Installing MCP9808 Wiring</a></li><li>• <a href="#">Installed Wiring</a></li><li>• <a href="#">Install Case to Bottom</a></li><li>• <a href="#">Sensor Wiring Clearance</a></li><li>• <a href="#">Install Loop Band to Strap</a></li><li>• <a href="#">Tension Loop</a></li><li>• <a href="#">Installed Loop Band</a></li><li>• <a href="#">USB Access</a></li><li>• <a href="#">Power Switch</a></li><li>• <a href="#">Assembled Case</a></li></ul>	
<b>Adafruit IO Setup</b>	<b>21</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Feed Setup</a></li><li>• <a href="#">Dashboard Setup</a></li><li>• <a href="#">Add Line Chart Block</a></li><li>• <a href="#">Add Gauge Block</a></li><li>• <a href="#">Action Setup</a></li></ul>	

- [Obtain Adafruit IO Key](#)

## Arduino Code 27

---

## Usage 31

---

- [Wearing Your Monitor](#)
- [Bluefruit LE Connect Usage](#)
- [Code Usage](#)
- [Viewing Sensor Data on Adafruit IO](#)
- [Fever Notifications](#)
- [Configure the Fever Temperature Threshold](#)
- [Change the Sensor Read Delay](#)



---

# Overview



## Wearable Continuous Temperature Monitor

This armband **continuously monitors body temperature while you sleep**. You can monitor temperature **hands-free** from another device, even if you're in a different room.

You can monitor temperature for **up to 8 hours on a full charge** (tested on a 3.3V 400mAh lipoly battery, data sent to Adafruit IO once every minute). **Data is sent to your Adafruit IO account for long-term storage and visualization**. Use Adafruit IO Triggers to be **notified by email when the temperature increases past a set threshold**.



## Bluetooth Low Energy (BLE) Temperature Armband

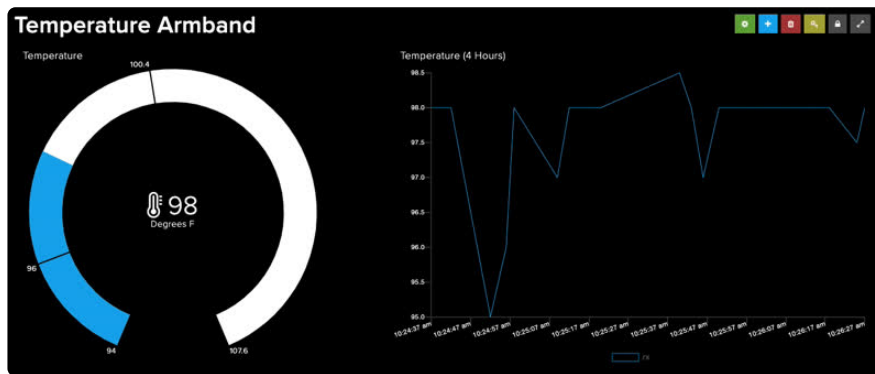
Monitor your (or someone else's) temperature, hands-free, from anywhere with the Adafruit Feather nRF52840.

Temperature readings are sent from the Feather nRF52840's Bluetooth-Low-Energy (BLE) radio to the [Adafruit BLE Connect App \(https://adafru.it/DNc\)](https://adafru.it/DNc).

## Interactive Temperature Dashboard

[Adafruit IO \(https://adafru.it/fH9\)](https://adafru.it/fH9) is the easiest way to stream, log, and interact with **your data**. This project uses Adafruit IO to log and display temperature data on an

interactive dashboard. You can check the current thermometer reading and monitor the temperature over time.



Create a new Trigger ✕

If

Is greater than or equal to

102

Then

## Fever Email Alerts

If a temperature threshold is reached, **your** armband sends an email to your phone.

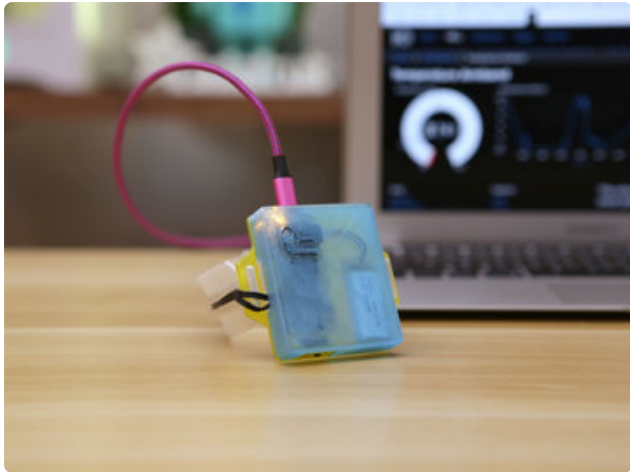
### Adafruit IO trigger 20554: Alert from rx

Adafruit IO <notify@io.adafruit.com>  
to me ▾

The value of "rx" was:

102

at 2020-04-01T14:39:09.220Z



## Hands-Free and Rechargeable

We've designed this wearable armband to hold a rechargeable 3.7V 400mAh lipo battery. The Feather NRF52840 has an on-board lipo charger - plug it into micro-USB to charge between uses.



## About your Health Data and Adafruit IO

We won't share your account information with anyone without your permission and everything you store on Adafruit IO is **private by default**, whether you're a paying customer or not.

We believe Internet of Things devices should come with a well established expectation of what they will and will not do with consumer's data.

- Read our [Internet of Things Bill of Rights](https://adafru.it/BIl) (<https://adafru.it/BIl>)
- Read the [Adafruit IO Privacy Policy](https://adafru.it/Ke0) (<https://adafru.it/Ke0>)
- Read the [Adafruit Privacy Policy](https://adafru.it/whc) (<https://adafru.it/whc>)

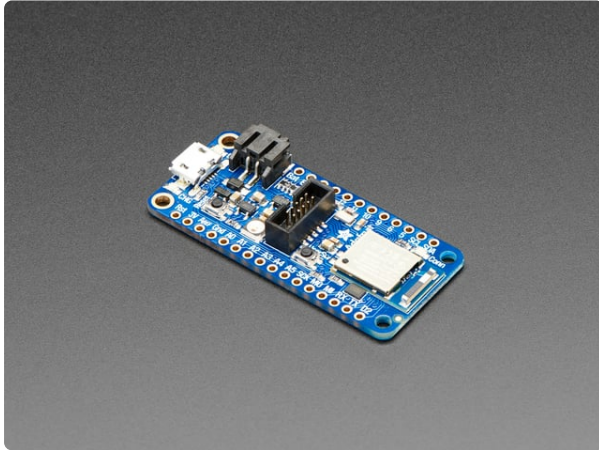
## About this Project

- This project is not a medical device.
- This project is not waterproof or water-resistant
- This project is not designed for use with children.



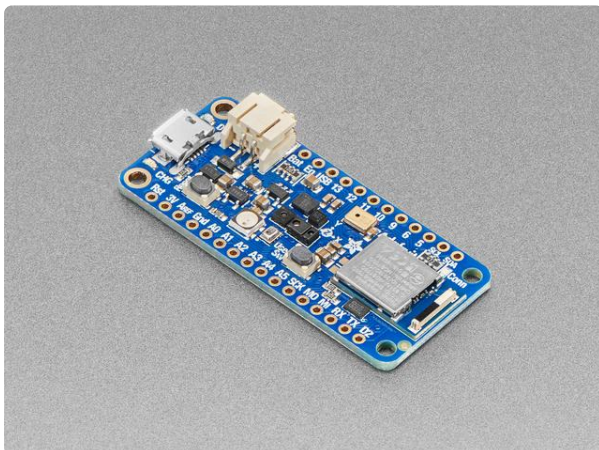
## Parts

The parts below were used in this project. You don't have to use the exact same parts but the enclosure was designed to specially fit these parts - so if you want to use different components, you can but just know it may not fit in the provided enclosure. You can of course tweak the CAD files.



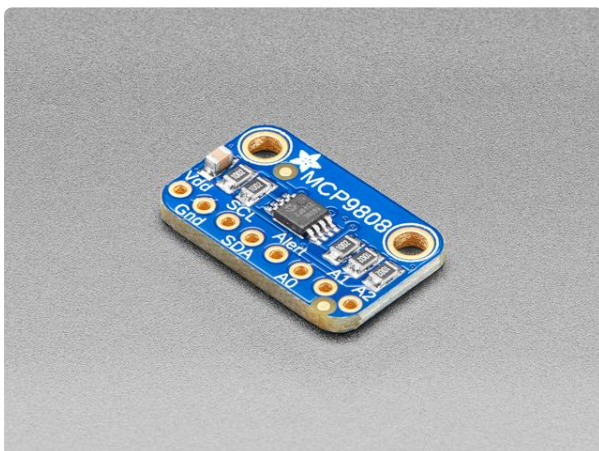
### [Adafruit Feather nRF52840 Express](https://www.adafruit.com/product/4062)

The Adafruit Feather nRF52840 Express is the new Feather family member with Bluetooth Low Energy and native USB support featuring the nRF52840! It's...  
<https://www.adafruit.com/product/4062>



### [Adafruit Feather nRF52840 Sense](https://www.adafruit.com/product/4516)

The Adafruit Feather Bluefruit Sense takes our popular Feather nRF52840 Express and adds a smorgasbord of sensors...  
<https://www.adafruit.com/product/4516>



### [MCP9808 High Accuracy I2C Temperature Sensor Breakout Board](https://www.adafruit.com/product/1782)

This I2C digital temperature sensor is one of the more accurate/precise we've ever seen, with a typical accuracy of  $\pm 0.25^{\circ}\text{C}$  over the sensor's  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ ...  
<https://www.adafruit.com/product/1782>





### Breadboard-friendly SPDT Slide Switch

These nice switches are perfect for use with breadboard and perfboard projects. They have 0.1" spacing and snap in nicely into a solderless breadboard. They're easy to switch...

<https://www.adafruit.com/product/805>



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

Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...

<https://www.adafruit.com/product/3898>



### Silicone Cover Stranded-Core Ribbon Cable - 10 Wire 1 Meter Long

For those who are fans of our silicone-covered wires, but are always looking to up their wiring game. We now have Silicone Cover Ribbon cables! These may look...

<https://www.adafruit.com/product/3890>

## Tools

Having the right tools makes this build easier and more fun. Most of these are available in the shop but use whatever tools you have on hand.

- [3D Printer](https://adafru.it/diH) (<https://adafru.it/diH>) and [Filament](http://adafru.it/2080) (<http://adafru.it/2080>) (or a 3D Printing Service)

Hakko FX-888D

**1 x Soldering Iron**

<https://www.adafruit.com/product/1204>

Hakko FX-888D

---

**1 x Flush Cutters**

<https://www.adafruit.com/product/152>

Flush diagonal cutters - CHP170

---

**1 x Wire Strippers**

<https://www.adafruit.com/product/527>

Hakko Professional Quality 20-30 AWG Wire Strippers

---

**1 x Panavise Jr**

<https://www.adafruit.com/product/151>

Panavise Jr. - PV-201

---

**1 x Helping Hands**

<https://www.adafruit.com/product/291>

Helping Third Hand Magnifier W/Magnifying Glass

---

**1 x Screwdriver**

<https://www.adafruit.com/product/829>

65 Piece Ratchet Screwdriver and Tool Bit Set

---

## Materials

Wires, screws, magnets, filament - The supplies listed below are both helpful and necessary for completing this project.

**1 x 3D Printer Filament**

<https://www.adafruit.com/product/2080>

Filament for 3D Printers in Various Colors and Types

---

**1 x 10-wire Silicone Cover Stranded-core Wire**

<https://www.adafruit.com/product/3890>

28AWG 10-wire ribbon cable

---

**1 x Heat Shrink**

<https://www.adafruit.com/product/1649>

Multi-Colored Heat Shrink Pack - 3/32" + 1/8" + 3/16" Diameters

---

**1 x Solder, Rosin-Core**

<https://www.adafruit.com/product/145>

Mini Solder spool - 60/40 lead rosin-core

---

**1 x Micro USB Cable**

<https://www.adafruit.com/product/4148>

Pink and Purple Braided USB A to Micro B Cable - 2 meter long

---

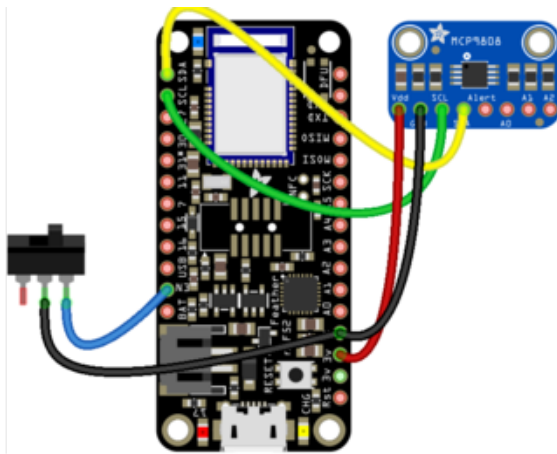
---

# Circuit Diagram

The diagram below provides a visual reference for wiring of the components. This diagram was created using the software package [Fritzing](https://adafru.it/oEP) (<https://adafru.it/oEP>).

## Adafruit Library for Fritzing

Use Adafruit's Fritzing parts library to create circuit diagrams for your projects. Download the library or just grab individual parts. Get the library and parts from [GitHub - Adafruit Fritzing Parts](https://adafru.it/AYZ) (<https://adafru.it/AYZ>).



Make the following connections between the Feather NRF52840 and the MCP9808

**Feather 3V to sensor Vdd**  
**Feather GND to sensor GND**  
**Feather SCL to sensor SCL**  
**Feather SDA to sensor SDA**

Make the following connections between the Feather NRF52840 and a SPDT slide-switch

**Feather GND to Center Switch Pin**  
**Feather EN to One of the outer pins on the switch**

Note that the Feather **GND** pad is shared between the MCP9808 and the switch.

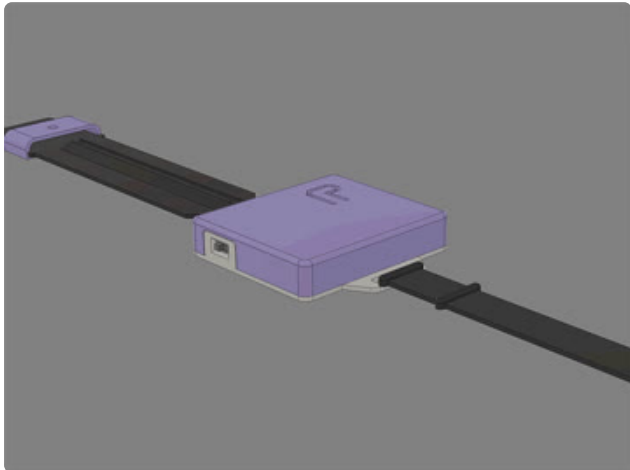
# 3D Printing



## Parts

These parts are designed for FDM 3D printing. Parts are designed for support free 3D printing. List of parts:

wctm-case.stl  
wctm-bottom.stl  
wctm-strap-sensor.stl  
wctm-loop-band.stl  
wctm-sensor-cover.stl



## Case Assembly

The case was designed in Autodesk Fusion 360 and features sketches and editable user parameters. The design source is available to download in STEP and other file formats.



## Sensor Assembly

The MCP9808 temperature sensor breakout is mounted to a flexible armband. The breakout PCB is press fitted onto built-in posts on the armband. Because the armband is printed flexible material, the posts can be forced through the mounting holes.

[Download CAD from Fusion 360](#)

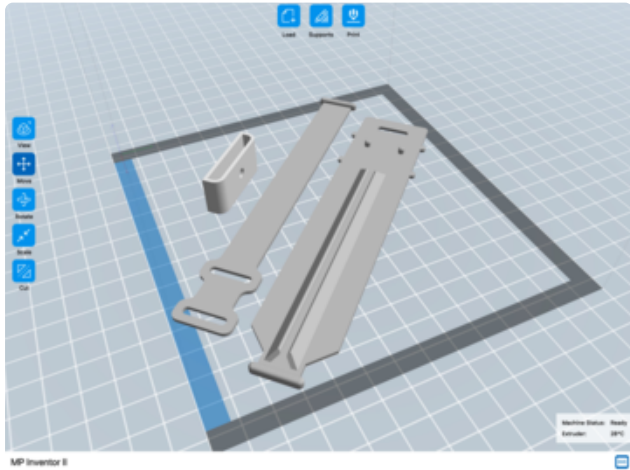
<https://adafru.it/Kdk>

[Download CAD from Prusa Printers](#)

<https://adafru.it/Kek>

Download CAD from Thingiverse

<https://adafru.it/Kel>



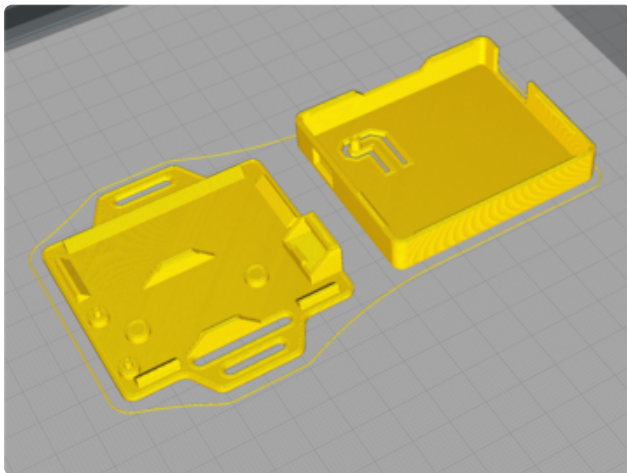
## Flexible Parts

The sensor cover and two arm strap are designed for flexible filament. These parts must be printed in TPU material in order to function properly.

wctm-strap-sensor.stl

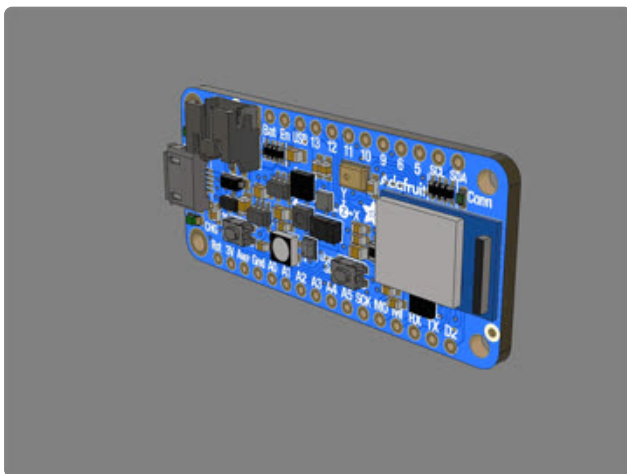
wctm-loop-band.stl

wctm-sensor-cover.stl



## PLA Parts

The bottom cover and case are designed for rigid material like PLA or PETG. These do not require any support material.



## Design Source Files

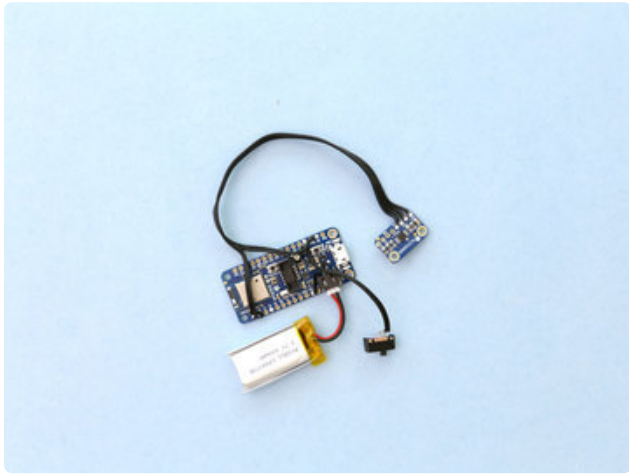
The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, STL and more. Electronic components like Adafruit's board, displays, connectors and more can be downloaded from the [Adafruit CAD parts GitHub Repo](https://adafru.it/AW8) (<https://adafru.it/AW8>).

Adafruit CAD Parts on GitHub

<https://adafru.it/AW8>



# Assembly



## Wiring Circuit

Solder the MCP9808 sensor and slide switch to the pins on the Feather nRF52840. 10-wire 28AWG silicone covered wire is suggested. Use the following wire lengths:

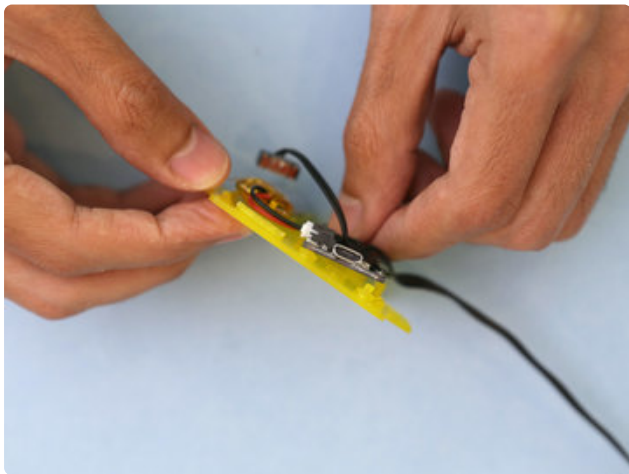
### MCP9808 4-Wire Cable

160mm (6.3in)

### Slide Switch 2-Wire Cable

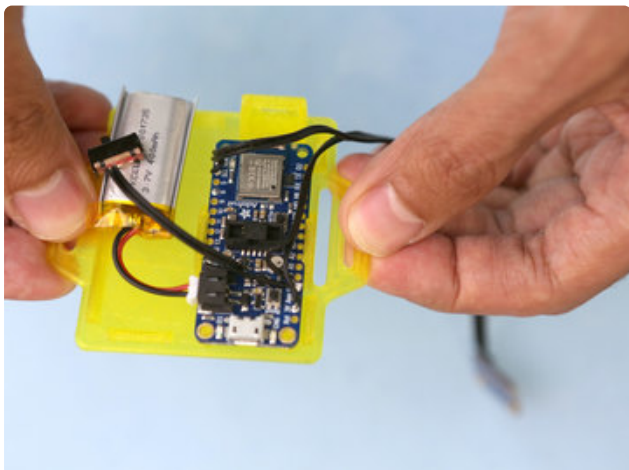
50mm (2in)

The ground pin on Feather nRF82840 is shared across the slide switch and the MCP9808



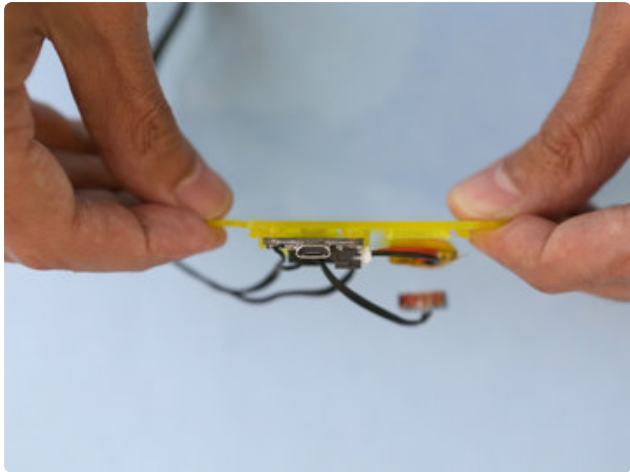
## Install Feather nRF52840

Place the Feather PCB on top of the standoffs on the bottom cover. Reference the photo for correct placement. Insert the PCB at an angle so the edge is underneath one of the tabs. Fit one of the mounting holes onto one of the standoffs.



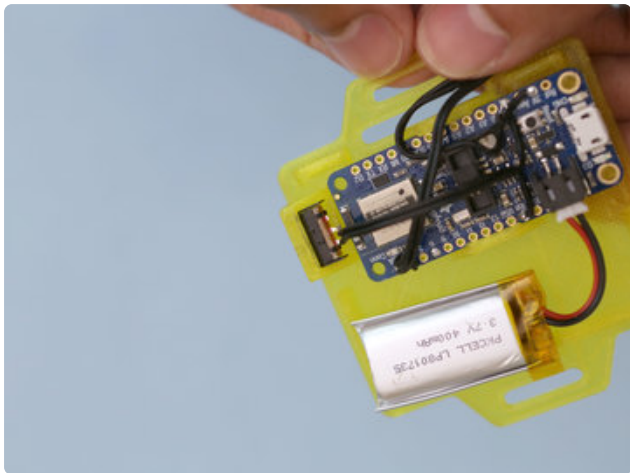
## Secure Feather

Firmly press the PCB down so it clips under the second tab. The two tabs will keep the Feather secured in place. Ensure the second mounting hole is properly fitted onto the standoff.



## Installed PCB

The Feather PCB should be flush like shown in the photo. A clearance of 2mm allows for wiring on the bottom.



## Installing Switch

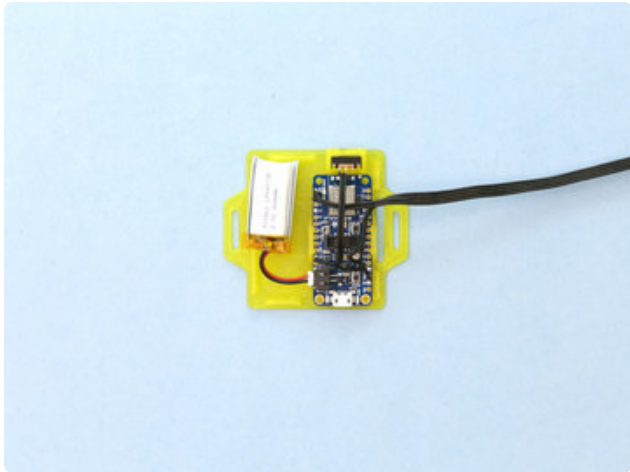
Insert the body of the slide switch to the built-in holder on the bottom cover. Firmly press the switch into the holder to secure it in place.



## Installed Switch

The actuator should be accessible on the outside of the holder.





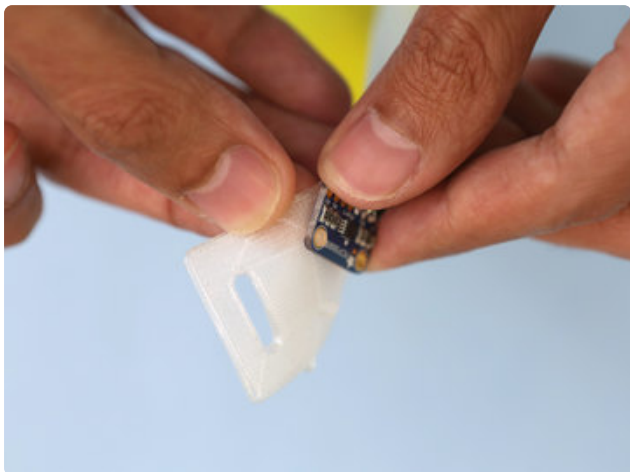
## Install Parts

The 400mah lipo battery can be secured in place with double-sided tape or mounting tack. Double check the slide switch and Feather nRF52840 are properly installed.



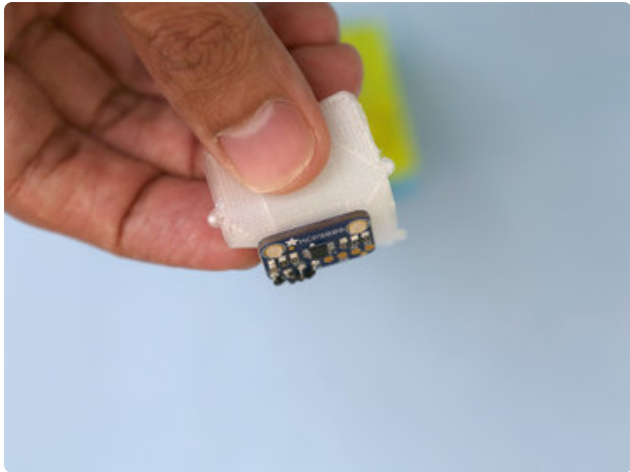
## Install Flexible Strap

Insert the tabbed end of the strap to the slotted tab on the bottom cover closest to the Feather.



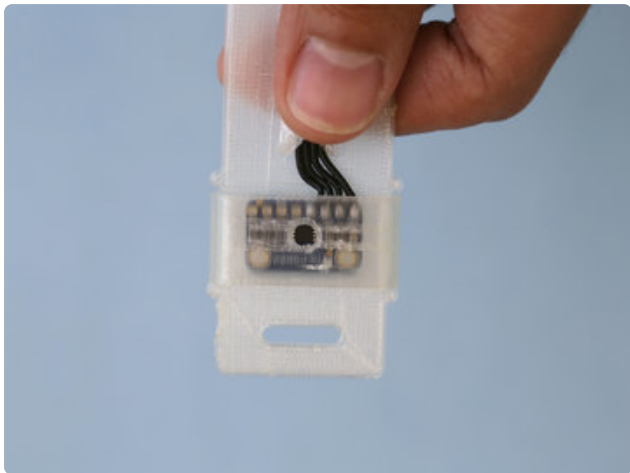
## Install MCP9808

Place the MCP9808 PCB over the two nubs on the end of the flexible strap. Firmly press the nubs into the two mounting holes on the MCP9808 PCB.



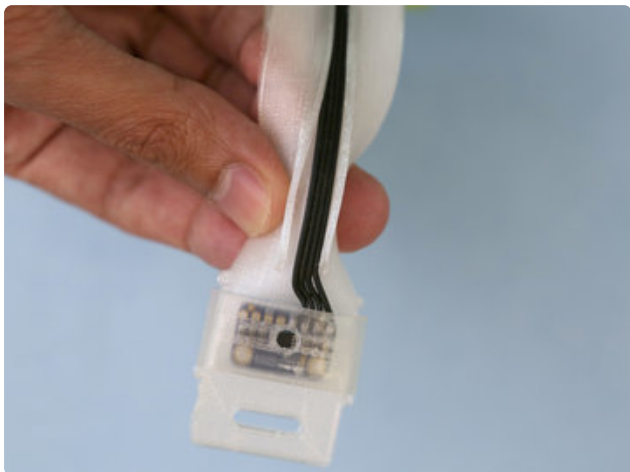
## Installed MCP9808

The nubs are essentially squeezed into the mounting holes – This is only possible with flexible material like Ninjaflex, Cheetach or TPU filament.



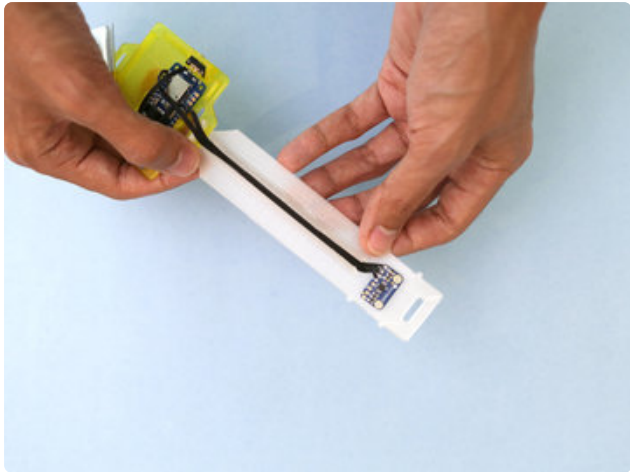
## Install Sensor Cover

Slip the cover onto the flexible strap to protect the MCP9808 sensor. The center hole in the cover should be properly lined up with the temperature sensor – if not, reverse the orientation of the cover.



## Installing MCP9808 Wiring

Pinch the flexible strap to open up the channel in the middle of the band. Begin to fit the 4-wire ribbon cable into the channel. Press wiring down so it's flattened into place.



## Installed Wiring

The silicone covered ribbon cable provides a nice grip combined with the flexible strap. By Design, the wiring will be kept in place when bending the strap inwards or outwards.



## Install Case to Bottom

The case snap fits onto the bottom cover. Line up the cutouts with the features on the bottom cover. The case features a cutaway for the sensor wiring.



## Sensor Wiring Clearance

The wiring from the sensor is fitted through the opening on the side of the case.



## Install Loop Band to Strap

Insert the loop band through the end on the flexible strap.



## Tension Loop

Grab the nub end of the loop band and fit it through the two slots. Reference the photo for correct installation.



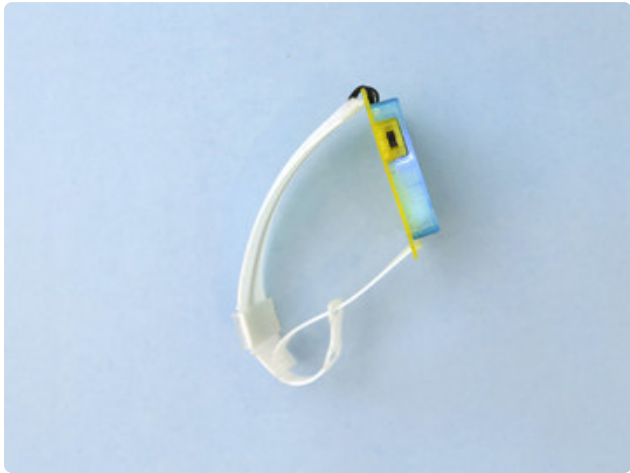
## Installed Loop Band

Fit the nub end of the loop band through the slotted tab on the bottom cover. Firmly press the nub end through.



## USB Access

The USB port is accessible for recharging the 400mah lipo battery and reprogramming.



## Power Switch

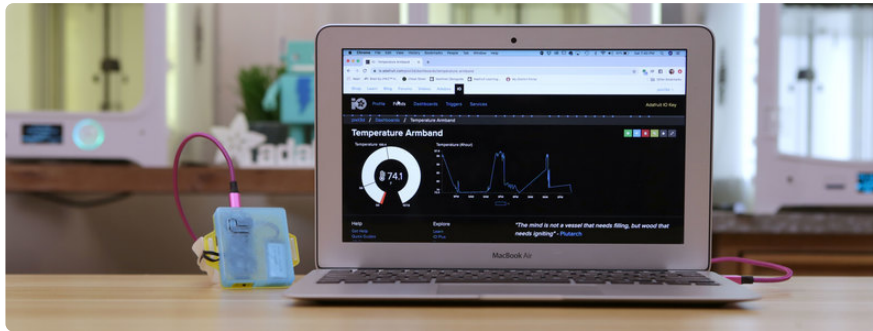
Use the slide switch to power the circuit on and off.



## Assembled Case

And there we have it! The case is assembled and ready to wear.

# 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 data from the temperature sensor. 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 **rx**.

- 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

Maximum length: 128 characters. Used: 2

Description

Add to groups

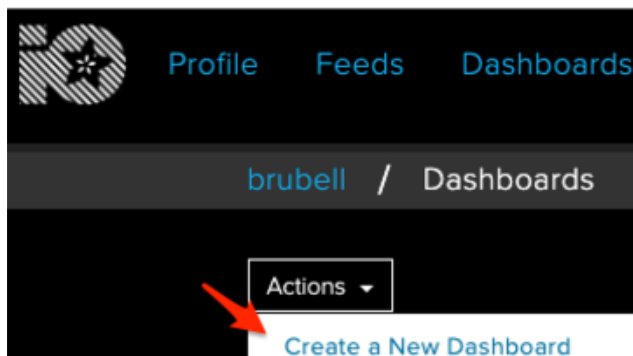
Cancel

Create

## Dashboard Setup

Now that we have feeds to hold data and display the type of data, we need a way to consolidate these feeds in one place. Dashboards are a feature of Adafruit IO which allow us to display and control feeds using widgets called Blocks.

- If you haven't used an Adafruit IO Dashboard before, [check out the Adafruit IO Basics: Dashboards learn guide \(https://adafru.it/f5m\)](https://adafru.it/f5m) for a quick primer.



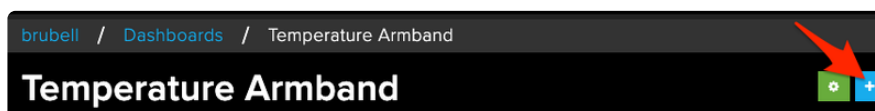
Navigate to the [Adafruit IO Dashboards page \(https://adafru.it/eIS\)](https://adafru.it/eIS).

From the Actions dropdown, click **Create a New Dashboard**.

Name the dashboard **Temperature Armband**.

## Add Line Chart Block

The MCP9808 temperature sensor sends data to Adafruit IO once per minute. We'll want a way of displaying historical data on the dashboard. From the dashboard, **click the + button to create a new block**.



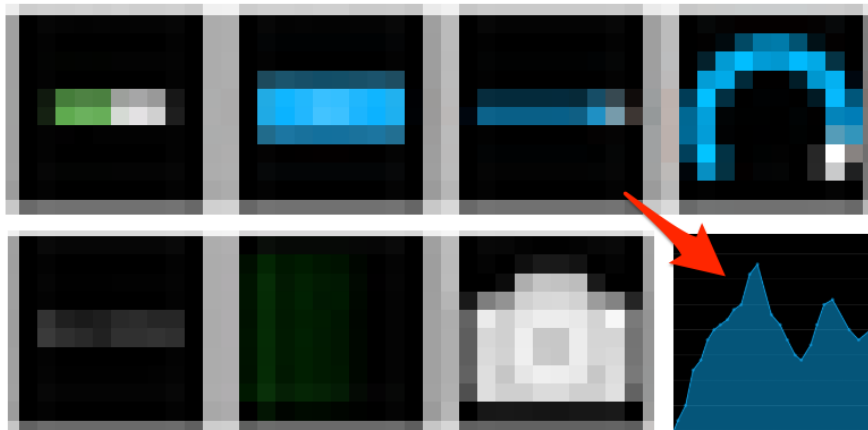
From the modal, **Click the Line Chart block**.



## Create a new block



Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



## Choose up to 5 feeds

**Line Chart:** The line chart is used to graph one or more feeds.

If you have a lot of feeds, you may want to use the search function.

### Group / Feed

☐ My Feeds

☒ rx

Search for the rx feed.

Click the checkbox next to the feed.

Click Next Step

## Block settings



In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Temperature (4 Hours)

Show History

4 hours

X-Axis Label

X

Y-Axis Label

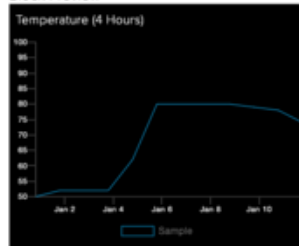
Y

Y-Axis Minimum

Leave blank to automatically detect.

1/1/2018 11:11:11 AM

Block Preview



Line Chart The line chart is used to graph one or more feeds.

Name the Block Title to Temperature (4 Hours)

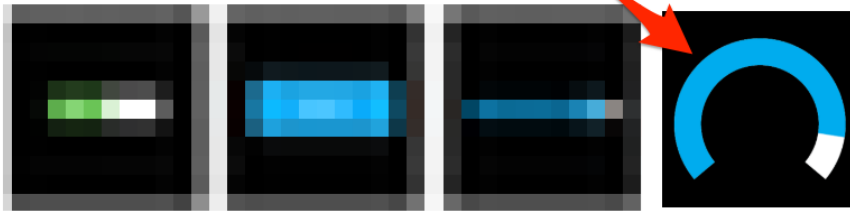
Set Show History to 4 hours

## Add Gauge Block

A gauge is a read only block type that shows a fixed range of values. We'll add a gauge block to display the real-time value from the wearable temperature monitor in real-time

## Create a new block

Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



## Choose feed

**Gauge:** A gauge is a read only block type that st

If you have lot of feeds, you may want to use the

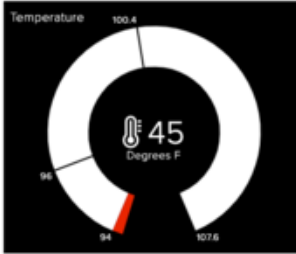
### Group / Feed

My Feeds

☒ rx

### Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)	Block Preview
<input type="text" value="Temperature"/>	
Gauge Min Value	
<input type="text" value="94"/>	
Gauge Max Value	
<input type="text" value="107.6"/>	
Gauge Width	
<input type="text" value="50px"/>	
Gauge Label	
<input type="text" value="Degrees F"/>	

Select the rx feed.

Set the **Block Title** to Temperature.

Set the **Gauge Min Value** to 94 (or 34 if you're measuring degrees Celsius).

Set the **Gauge Max Value** to 107.6 (or 42 if you're measuring degrees Celsius).

Set the **Gauge Width** to 50px.

Set the **Gauge Label** to Degrees F (or Degrees C if you're measuring degrees Celsius).

Low Warning Value  
96  
Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds.

High Warning Value  
100.4  
Optional. If no high warning value is given, the gauge will only change color when the value is out of bounds.

Decimal Places  
1  
Number of decimal places to display when value is a number. Defaults to 2.

☒ Show Icon  
When checked, show an icon with the value.

Icon  
thermometer  
Show this icon next to the value.

The gauge will change color if the value goes out of bounds.

**Set a Low Warning Value to 96.**

**Set the High Warning Value to 100.4**

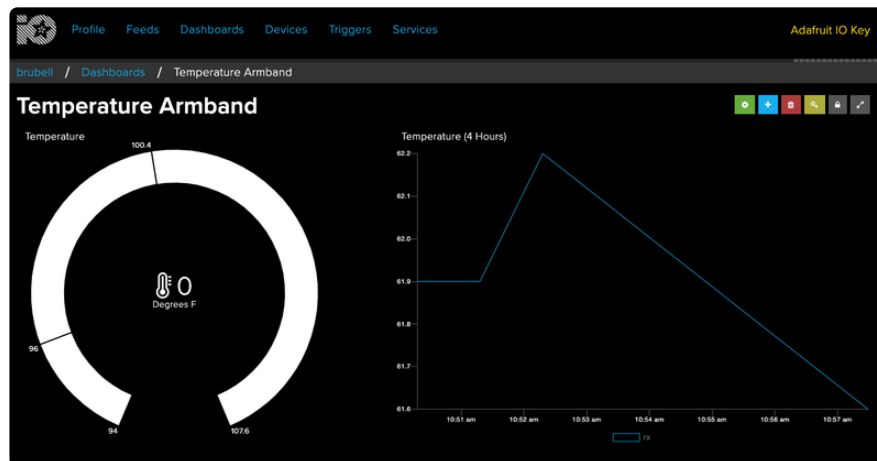
**Set Decimal Places to display to 1.**

You may also optionally show an icon next to the value. To do this:

**Tick the Show Icon checkbox**

**Set the icon to thermometer**

Your dashboard should look like the screenshot below:

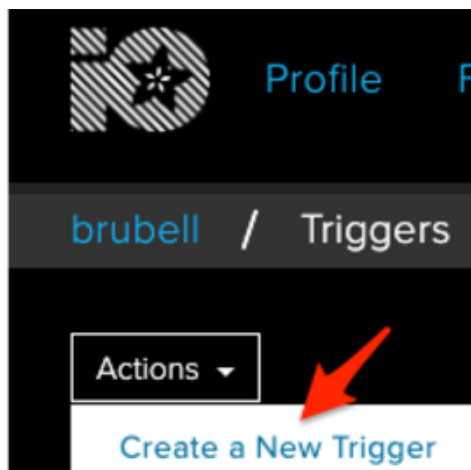


## Action Setup

Actions are a way to do something when a certain situation occurs. We'll be [using a reactive action \(https://adafru.it/lxa\)](https://adafru.it/lxa) to email us if the rx feed receives a value from the thermometer above our fever-threshold.

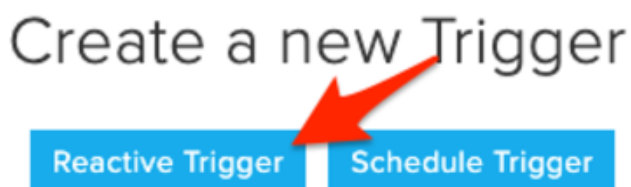
The "email me" reactive action feature is only for Adafruit IO PLUS (IO+) subscribers. To upgrade to Adafruit IO+, navigate to <https://io.adafruit.com/plus>.

Triggers were renamed to Actions in Adafruit IO.



Navigate to [your Adafruit IO account's Triggers page \(https://adafru.it/Ke4\)](https://adafru.it/Ke4). Click **Actions**, then click **Create a New Action**

Select the **Reactive Action**



Next, we'll configure the action.

Under **If**, **select the rx action feed**.

For the comparison, **select Greater Than or Equal to**.

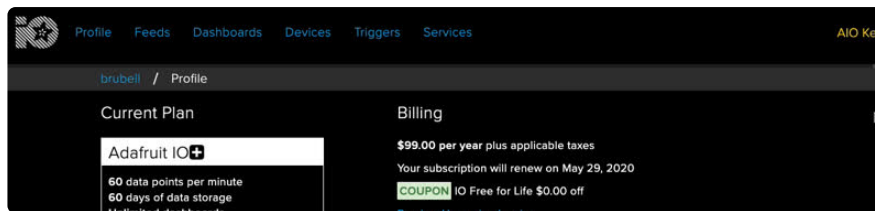
For the Comparison Value or Feed, **enter 102** (we're in the US, so our fever threshold is measured in degrees Fahrenheit. Change this value to 38.9 if you're planning to measure temperature in degrees Celsius).

**Select email me** for the action.

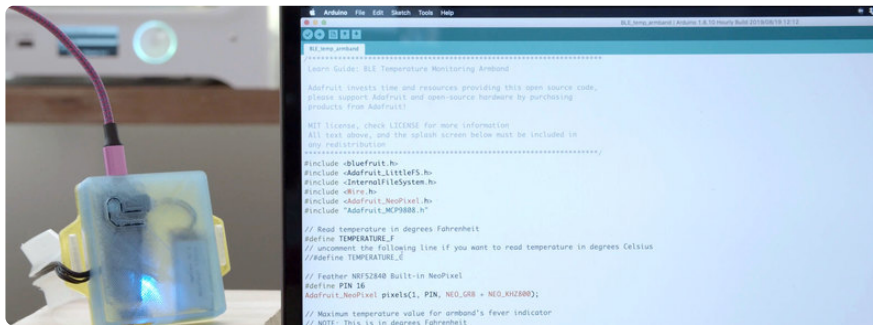
## Obtain Adafruit IO Key

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

Navigate to [your profile \(https://adafru.it/EpC\)](https://adafru.it/EpC) 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.



## Arduino Code



This guide assumes you've set up the Adafruit Bluefruit nRF52 Board Support Package.

- If you have not yet done this, [navigate to this page and come back when you've installed the package \(https://adafru.it/Ke5\)](https://adafru.it/Ke5).

Before proceeding with the code in this guide, you should be able to run the blinky sketch from the Arduino IDE.

- If you have not uploaded code to your Feather NRF52, [navigate to this page and come back when you've successfully uploaded code to the Feather NRF52 \(https://adafru.it/Ke6\)](https://adafru.it/Ke6).

Make sure your board is selected:

- Go to the **Tools** menu
- Select **Tools > Board > Adafruit Bluefruit nRF52840 Feather Express**

Copy the code below to the Arduino IDE. Click **verify**. Then, click **Upload**.

```
// SPDX-FileCopyrightText: 2020 Brent Rubell for Adafruit Industries
//
// SPDX-License-Identifier: MIT

/*****
Learn Guide: BLE Temperature Monitoring Armband
*****/
```

Adafruit invests time and resources providing this open source code,  
please support Adafruit and open-source hardware by purchasing  
products from Adafruit!

MIT license, check LICENSE for more information  
All text above, and the splash screen below must be included in  
any redistribution

```
*****/
#include <bluefruit.h>
#include <Adafruit_LittleFS.h>
#include <InternalFileSystem.h>
#include <Wire.h>
#include <Adafruit_NeoPixel.h>
#include "Adafruit_MCP9808.h"

// Read temperature in degrees Fahrenheit
#define TEMPERATURE_F
// uncomment the following line if you want to read temperature in degrees Celsius
// #define TEMPERATURE_C

// Feather NRF52840 Built-in NeoPixel
#define PIN 16
Adafruit_NeoPixel pixels(1, PIN, NEO_GRB + NEO_KHZ800);

// Maximum temperature value for armband's fever indicator
// NOTE: This is in degrees Fahrenheit
float fever_temp = 100.4;

// temperature calibration offset is +0.5 to +1.0 degree
// to make axillary temperature comparable to ear or temporal.
float temp_offset = 0.5;

// Sensor read delay, in minutes
int sensor_delay = 1;

// Measuring your armpit temperature for a minimum of 12 minutes
// is equivalent to measuring your core body temperature.
int calibration_time = 12;

// BLE transmit buffer
char temperature_buf [8];

// BLE Service
BLEDFU bledfu; // OTA DFU service
BLEDIS bledis; // device information
BLEUART bleuart; // uart over ble
BLEBAS blebas; // battery

// Create the MCP9808 temperature sensor object
Adafruit_MCP9808 tempsensor = Adafruit_MCP9808();

void setup() {
  Serial.begin(115200);
  Serial.println("Wearable BlueFruit Temperature Sensor");
  Serial.println("-----\n");

  if (!tempsensor.begin(0x18)) {
    Serial.println("Couldn't find MCP9808! Check your connections and verify the  
address is correct.");
    while (1);
  }
  Serial.println("Found MCP9808!");

  // Sets the resolution of reading
  tempsensor.setResolution(3);

  // Configure BLE
  // Setup the BLE LED to be enabled on CONNECT
```

```

// Note: This is actually the default behaviour, but provided
// here in case you want to control this LED manually via PIN 19
Bluefruit.autoConnLed(true);

// Config the peripheral connection with maximum bandwidth
Bluefruit.configPrphBandwidth(BANDWIDTH_MAX);

Bluefruit.begin();
Bluefruit.setTxPower(4);    // Check bluefruit.h for supported values
Bluefruit.setName("Bluefruit52");
Bluefruit.Periph.setConnectCallback(connect_callback);
Bluefruit.Periph.setDisconnectCallback(disconnect_callback);

// To be consistent OTA DFU should be added first if it exists
bledfu.begin();

// Configure and Start Device Information Service
bledis.setManufacturer("Adafruit Industries");
bledis.setModel("Bluefruit Feather52");
bledis.begin();

// Configure and Start BLE Uart Service
bleuart.begin();

// Start BLE Battery Service
blebas.begin();
blebas.write(100);

// Set up and start advertising
startAdv();

Serial.println("Please use Adafruit's Bluefruit LE app to connect in UART mode");

// initialize neopixel object
pixels.begin();

// set all pixel colors to 'off'
pixels.clear();
}

void loop() {

  // wakes up MCP9808 - power consumption ~200 mikro Ampere
  Serial.println("Wake up MCP9808");
  tempsensor.wake();

  // read and print the temperature
  Serial.print("Temp: ");
  #if defined(TEMPERATURE_F)
    float temp = tempsensor.readTempF();
    // add temperature offset
    temp += temp_offset;
    Serial.print(temp);
    Serial.println("*F.");
  #elif defined(TEMPERATURE_C)
    float temp = tempsensor.readTempC();
    // add temperature offset
    temp += temp_offset;
    Serial.print(temp);
    Serial.println("*C.");
  #else
    #warning "Must define TEMPERATURE_C or TEMPERATURE_F!"
  #endif

  // set NeoPixels to RED if fever_temp
  if (temp >= fever_temp) {
    pixels.setPixelColor(1, pixels.Color(255, 0, 0));
    pixels.show();
  }
}

```



```

// float to buffer
snprintf(temperature_buf, sizeof(temperature_buf) - 1, "%0.*f", 1, temp);

if (calibration_time == 0) {
    Serial.println("Writing to UART");
    // write to UART
    bleuart.write(temperature_buf);
}
else {
    Serial.print("Calibration time:");
    Serial.println(calibration_time);
    calibration_time-=1;
}

// shutdown MSP9808 - power consumption ~0.1 mikro Ampere
Serial.println("Shutting down MCP9808");
tempsensor.shutdown_wake(1);

// sleep for sensor_delay minutes
// NOTE: NRF delay() puts mcu into a low-power sleep mode
delay(1000*60*sensor_delay);
}

void startAdv(void)
{
    // Advertising packet
    Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE);
    Bluefruit.Advertising.addTxPower();

    // Include bleuart 128-bit uuid
    Bluefruit.Advertising.addService(bleuart);

    // Secondary Scan Response packet (optional)
    // Since there is no room for 'Name' in Advertising packet
    Bluefruit.ScanResponse.addName();

    /* Start Advertising
    * - Enable auto advertising if disconnected
    * - Interval: fast mode = 20 ms, slow mode = 152.5 ms
    * - Timeout for fast mode is 30 seconds
    * - Start(timeout) with timeout = 0 will advertise forever (until connected)
    *
    * For recommended advertising interval
    * https://developer.apple.com/library/content/qa/qa1931/\_index.html
    */
    Bluefruit.Advertising.restartOnDisconnect(true);
    Bluefruit.Advertising.setInterval(32, 244); // in unit of 0.625 ms
    Bluefruit.Advertising.setFastTimeout(30); // number of seconds in fast mode
    Bluefruit.Advertising.start(0); // 0 = Don't stop advertising
    after n seconds
}

// callback invoked when central connects
void connect_callback(uint16_t conn_handle)
{
    // Get the reference to current connection
    BLEConnection* connection = Bluefruit.Connection(conn_handle);

    char central_name[32] = { 0 };
    connection->getPeerName(central_name, sizeof(central_name));

    Serial.print("Connected to ");
    Serial.println(central_name);
}

/**
 * Callback invoked when a connection is dropped
 * @param conn_handle connection where this event happens

```

```
* @param reason is a BLE_HCI_STATUS_CODE which can be found in ble_hci.h
*/
void disconnect_callback(uint16_t conn_handle, uint8_t reason)
{
  (void) conn_handle;
  (void) reason;

  Serial.println();
  Serial.print("Disconnected, reason = 0x"); Serial.println(reason, HEX);
}
```

After you've successfully uploaded the code to your Feather, **toggle the switch to turn the power on**. The blue LED labeled CONN should blink.

---

## Usage



### Wearing Your Monitor

Slip your arm through the straps with the band facing out. The sensor should be located underneath your armpit when you place your arm at your side.

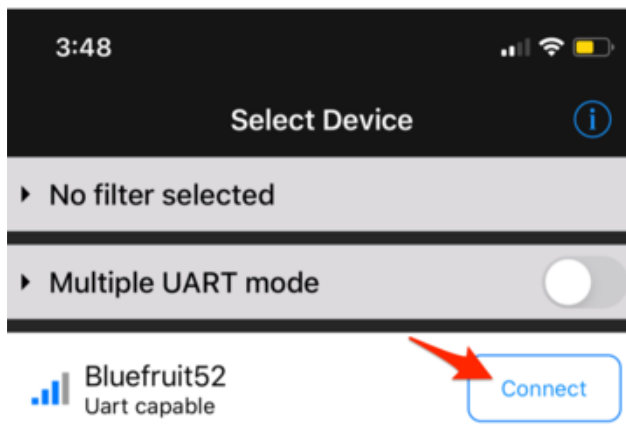
Use the loop strap to loosen or tighten the band. Adjust the armband until it no longer slides around your arm. Make sure it is not too tight, you do not want to cut off circulation.

## Bluefruit LE Connect Usage

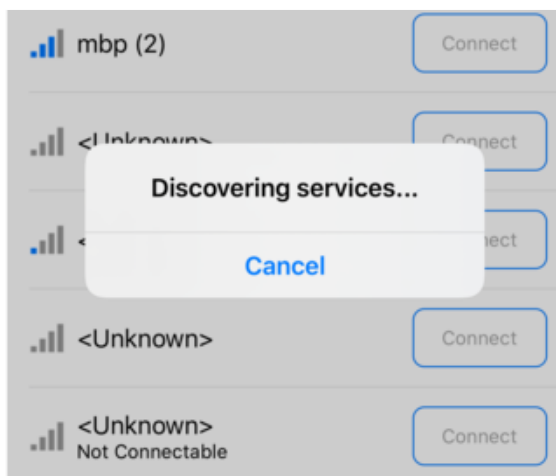
The [Adafruit Bluefruit LE Connect app](https://adafru.it/DNc) (<https://adafru.it/DNc>) provides iOS & Android devices with a variety of tools to communicate with Bluefruit LE devices such as the Feather NRF52840. We'll be using Bluefruit LE Connect App to send temperature data using the MQTT protocol to [Adafruit IO](https://adafru.it/fH9) (<https://adafru.it/fH9>).

- If you have not yet installed the Bluefruit LE Connect app and set up your phone, [follow this page and come back to this page when you're done](https://adafru.it/GLc) (<https://adafru.it/GLc>).

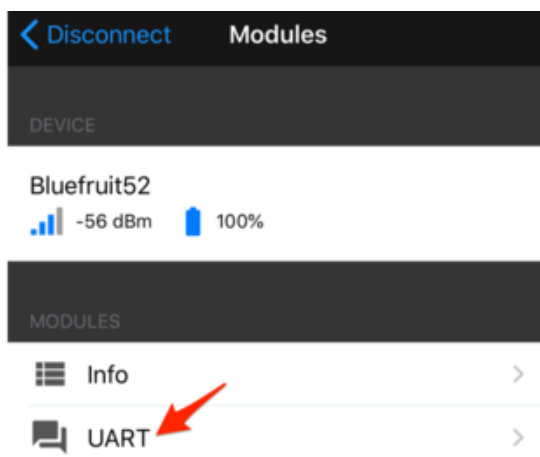
Once installed, **open the Bluefruit LE Connect app**. The app will automatically begin to scan for nearby Bluetooth LE devices.



Tap the **Connect** button next to the **Bluefruit52** device discovered by the app.



If you're having trouble finding your Bluefruit device in the scanned peripherals list, [visit this page for troubleshooting steps.](https://adafru.it/Ke7) (<https://adafru.it/Ke7>)

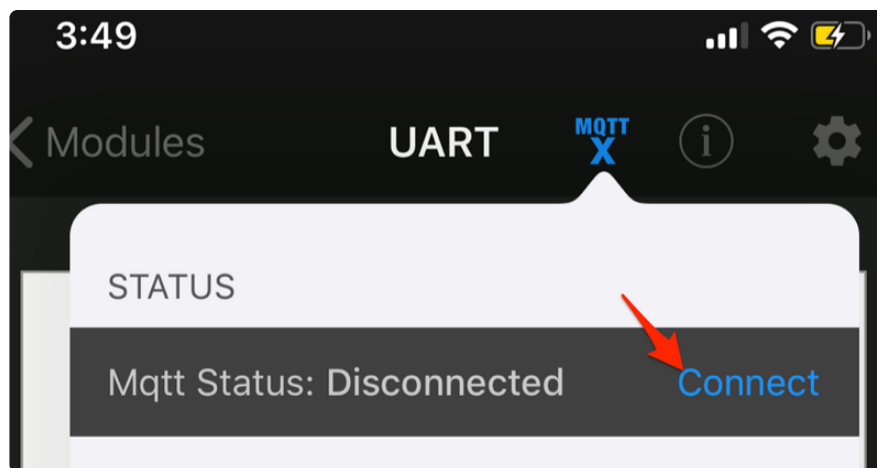


Once connected, the Bluefruit connect app will list its available modes. **Tap the UART mode.**

Next, you'll need to configure the Bluefruit LE Connect App's MQTT client with your Adafruit IO account information.

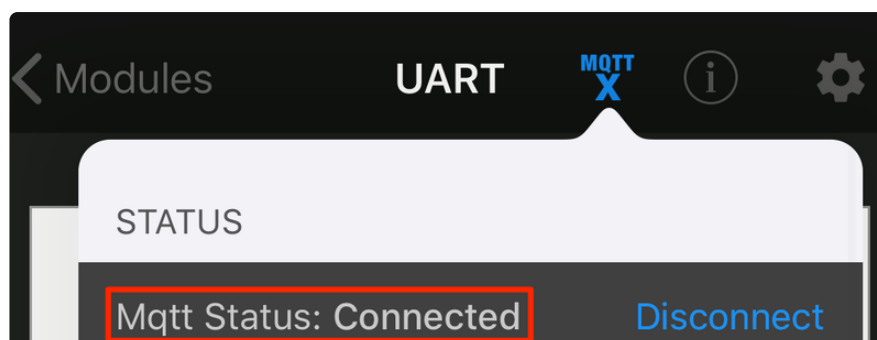
- If you have not done this yet, [follow this guide and come back to this page when you're done \(https://adafru.it/Ke8\)](https://adafru.it/Ke8).

Once configured, **tap Connect**



The MQTT Status will change to Connected, indicating that you've successfully connected to the Adafruit IO MQTT server.

- If you are not able to connect to Adafruit IO, [carefully re-read this guide and try re-connecting \(https://adafru.it/Ke8\)](https://adafru.it/Ke8).



## Code Usage



### Temperature Sensor Calibration

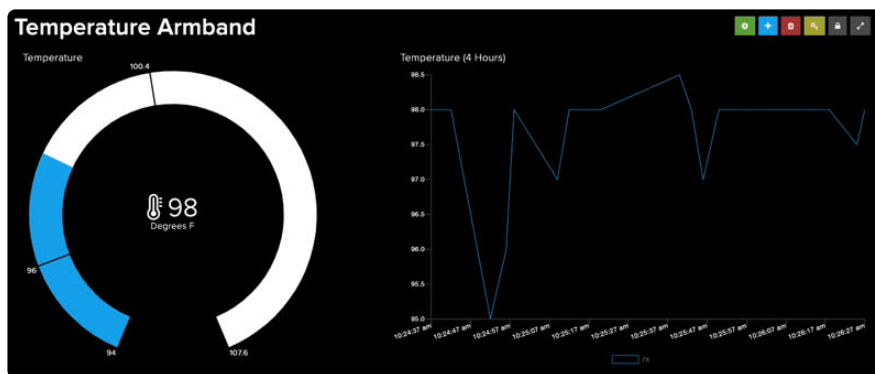
After connecting to the Bluefruit Connect App, the the armband will take twelve minutes to fully calibrate.

Your armpit provides an **axillary temperature**. Measuring your armpit temperature **for a minimum of 12 minutes is equivalent to measuring your core body temperature** (<https://adafru.it/Ke9>).

Make sure the sensor is in your armpit and that your arm is **firmly pressed against your side**. After twelve minutes have passed, temperature readings will display on the UART monitor.

## Viewing Sensor Data on Adafruit IO

Navigate to the Adafruit IO dashboard you created earlier. After 12 minutes have passed, the gauge and line block will update with the current temperature reading every minute. The line graph block updates and display the previous four hours of temperature data.



## Fever Notifications

The reactive trigger you set up earlier will email you if the temperature increases beyond a temperature threshold.

## Adafruit IO trigger 20554: Alert from rx



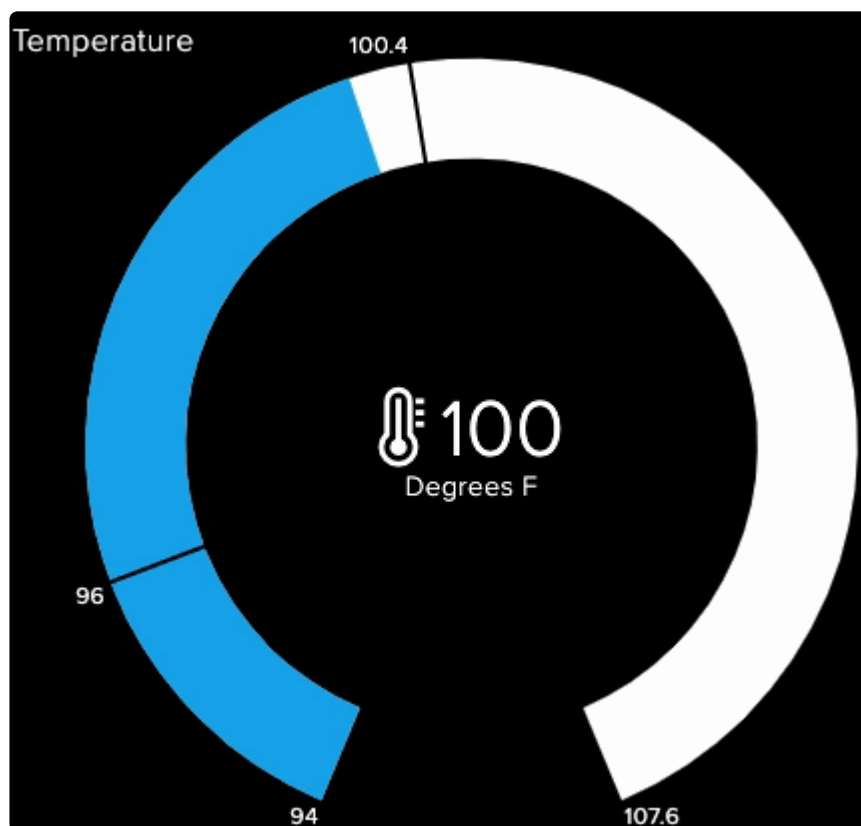
**Adafruit IO** <notify@io.adafruit.com>  
to me ▾

The value of "rx" was:

102

at 2020-04-01T14:39:09.220Z

If the temperature increases above the threshold you set, the Gauge block will change from a cool blue to bright yellow.



## Configure the Fever Temperature Threshold

The Feather NRF52840's built-in NeoPixel will glow bright red if the temperature is above a configurable fever threshold. To change this value in the code, change the following line from:

```
// Maximum temperature value for armband's fever indicator
```

```
// NOTE: This is in degrees Fahrenheit
```

```
float fever_temp = 100.4;
```

to

```
// Maximum temperature value for armband's fever indicator
```

```
// NOTE: This is in degrees Fahrenheit
```

```
float fever_temp = DESIRED_FEVER_TEMPERATURE;
```

## Change the Sensor Read Delay

By default, the code sends data to Adafruit IO every minute, then puts the temperature sensor and the NRF52840 to sleep.

If you want to change this code to read data at faster or slower intervals, change the following line from:

```
// Sensor read delay, in minutes
```

```
int sensor_delay = 1;
```

to

```
// Sensor read delay, in minutes
```

```
int sensor_delay = DESIRED_SENSOR_DELAY;
```

Keep in mind that every time the loop is executed, the microcontroller wakes up, executes code, and wirelessly sends data. You may want to increase the delay to conserve power.