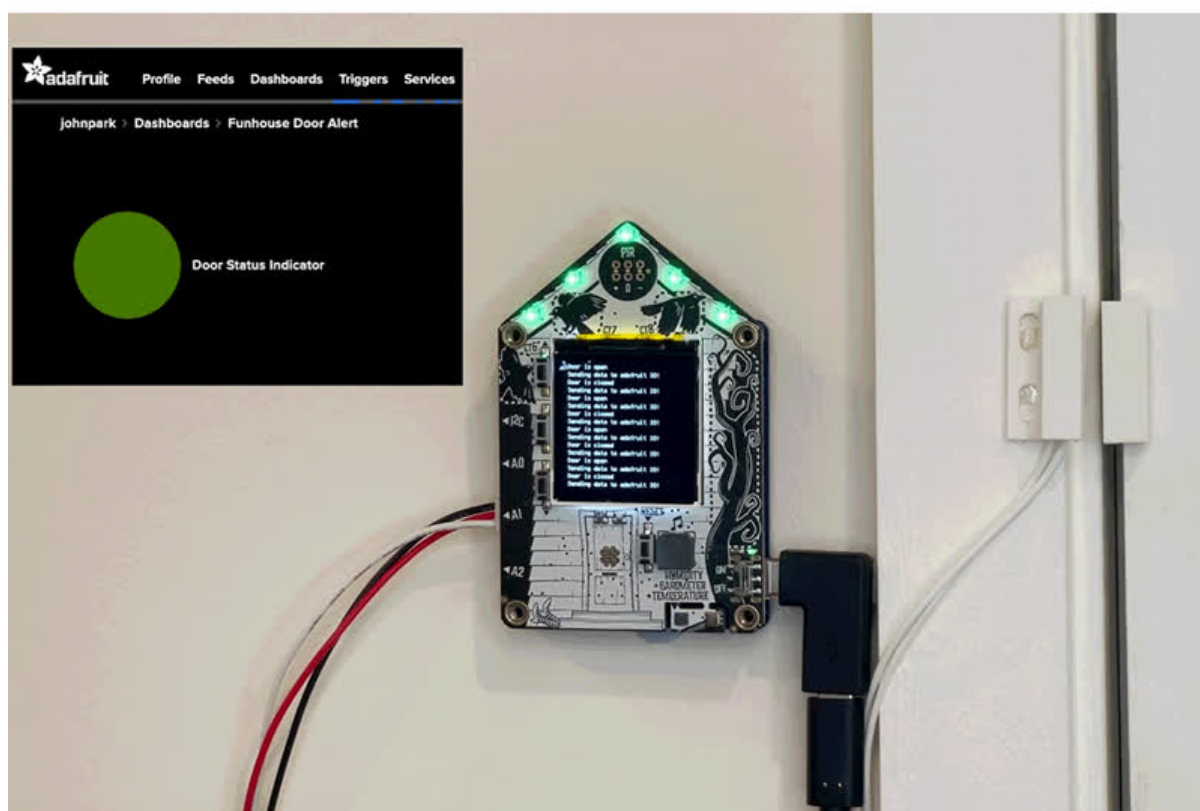




# Funhouse Door Alert with Email Notification

Created by John Park



<https://learn.adafruit.com/funhouse-door-alert-email-notification>

Last updated on 2024-06-03 03:22:41 PM EDT

# Table of Contents

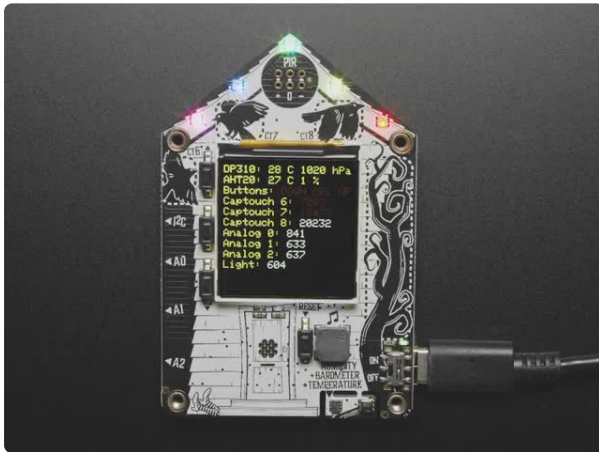
Overview	3
<hr/>	
• Parts	
Adafruit IO Setup	4
<hr/>	
• Setting up Adafruit IO Feeds	
• Creating the Adafruit IO Dashboard	
Email Alerts	9
<hr/>	
• Email Alert with AIO and IFTTT	
• Create Applet	
• Add a Condition	
• Choose Adafruit Service	
• Choose Feed Trigger	
• Configure the Trigger	
• Add an Action	
• Choose Gmail Service	
• Choose Gmail Action	
• Email Fields	
• Applet Summary	
• Review and Finish	
• Alert Received	
• Adafruit IO+ Actions	
• Actions	
• New Action	
• Reactive Action	
Code the Funhouse Door Alert	15
<hr/>	
• Install CircuitPython	
• Shhhh... Secrets	
• Text Editor	
• Download the Project Bundle	
• How it Works	
Build the Funhouse Door Alert	19
<hr/>	
• Door Sensor	
• Wire the Sensor	
• Mount the Funhouse	
• Mount the Sensor and Magnet	

---

# Overview

Create a simple door alert system with the Funhouse and Adafruit IO. It'll update your dashboard and send an email whenever a certain door is opened or closed.

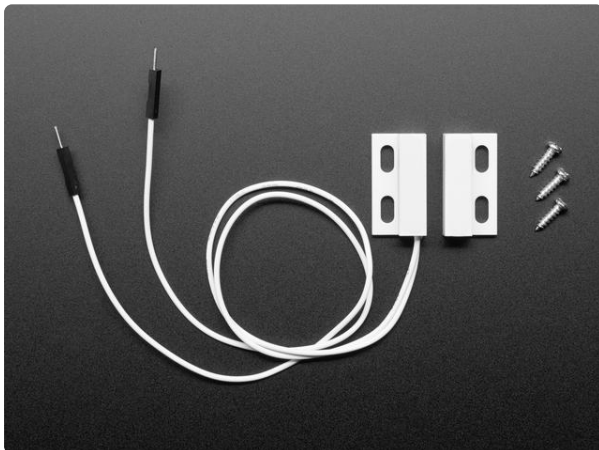
## Parts



### [Adafruit FunHouse - WiFi Home Automation Development Board](https://www.adafruit.com/product/4985)

Home is where the heart is...it's also where we keep all our electronic bits. So why not wire it up with sensors and actuators to turn our house into an electronic wonderland....

<https://www.adafruit.com/product/4985>



### [Magnetic contact switch \(door sensor\)](https://www.adafruit.com/product/375)

This sensor is essentially a reed switch, encased in an ABS plastic shell. Normally the reed is 'open' (no connection between the two wires). The other half is a magnet. When...

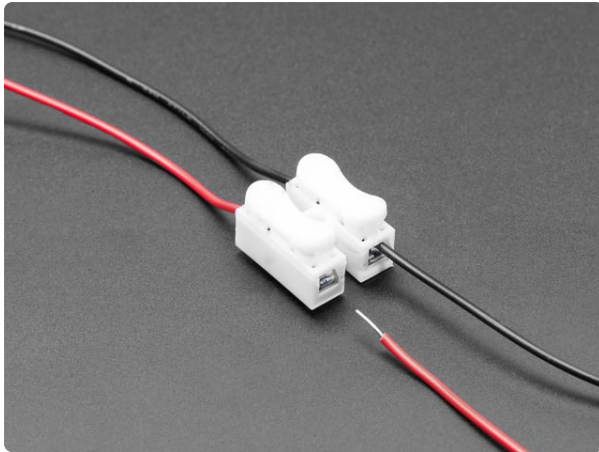
<https://www.adafruit.com/product/375>



### [STEMMA JST PH 2mm 3-Pin to Male Header Cable - 200mm](https://www.adafruit.com/product/3893)

This cable will let you turn a JST PH 3-pin cable port into 3 individual wires with high-quality 0.1" male header plugs on the end. We're carrying these to match up with our...

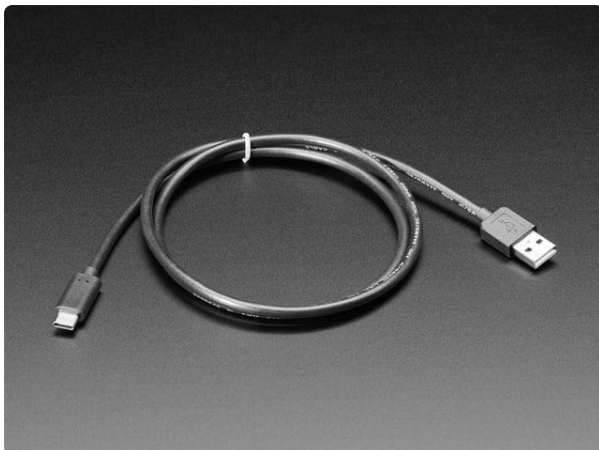
<https://www.adafruit.com/product/3893>



### 2-Pin Wire Joints (3-pack)

These are the easiest way possible to connect two wires - no if's, and's, or butt's! Well, actually, these are often referred to as wire butts 'cause you...

<https://www.adafruit.com/product/3786>



### USB Type A to Type C Cable - approx 1 meter / 3 ft long

As technology changes and adapts, so does Adafruit. This USB Type A to Type C cable will help you with the transition to USB C, even if you're still...

<https://www.adafruit.com/product/4474>



### 5V 2A Switching Power Supply w/ USB-A Connector

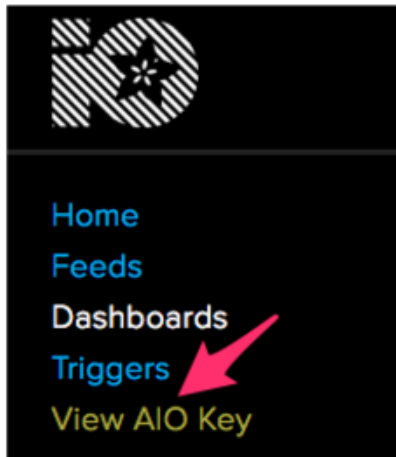
Our 5V 2A USB power adapter is the perfect choice for powering single-board computers like Raspberry Pi, BeagleBone, or anything else that's power-hungry! This adapter was...

<https://www.adafruit.com/product/1994>

---

## Adafruit IO Setup

New to Adafruit IO? [You can read all about it in this guide \(https://adafru.it/Co4\)](https://adafru.it/Co4). Go ahead and get your account set up, and come back to this page when you're ready.



You'll need to obtain our Adafruit IO Key and Username. Visit your [Adafruit IO Profile page \(https://adafru.it/BmD\)](https://adafru.it/BmD) and click the **VIEW AIO KEY** button on the left-sidebar.

A window will pop up with your Adafruit IO key and username. Keep a copy of them in a safe place, you'll need them later.

### YOUR AIO KEY



Your Adafruit IO key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your AIO key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new AIO key, all of your existing programs and scripts will need to be manually changed to the new key.



Username

Active Key

REGENERATE AIO KEY

[Show Code Samples](#)

**Your AIO Username**

**Your AIO Key**

## Setting up Adafruit IO Feeds

You'll create a feed called **door** to receive the door switch status data sent by the Funhouse.

To create the feed for the door, navigate to the [Adafruit IO Feeds Page \(https://adafru.it/mxC\)](https://adafru.it/mxC) and click **Actions->Create a New Feed**. Name the new feed **door** and then click **Create**.

## Create a new Feed



Name

Maximum length: 128 characters. Used: 4

Description

Add to groups

Cancel

Create

- If you do not know how to create feeds, [head over to the Adafruit IO Basics: Feeds for a quick overview \(https://adafru.it/ioA\)](https://adafru.it/ioA) of this process.

## Creating the Adafruit IO Dashboard

Next, you'll create an Adafruit IO Dashboard to display and control our feeds.

[Navigate to the Adafruit IO Dashboard page \(https://adafru.it/eIS\)](https://adafru.it/eIS) and click **Actions -> Create a New Dashboard**.

Name this dashboard **Funhouse Door Alert** and click **Create**. You'll be re-directed to the new Dashboard.

## Create a new Dashboard



Name

Funhouse Door Alert

Description

☐ Show Header Image

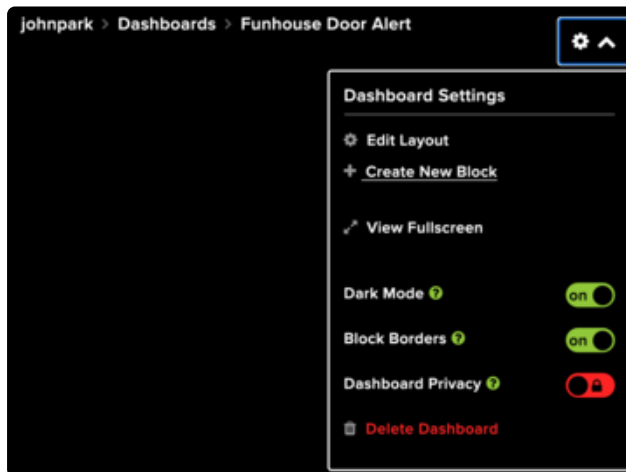
Header Image

No file chosen

[Sample header image with breakpoints marked.](#)

Cancel

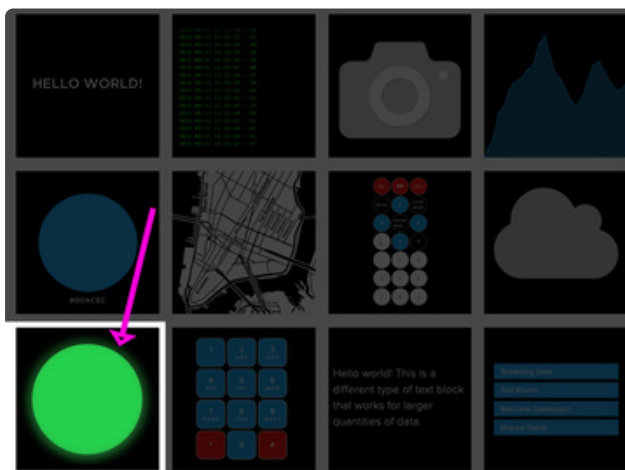
Create



### Add a Block

You'll create an **Indicator Block** to monitor the status of the door sensor.

From the IO Home dashboard, click the **gear icon** to see the **Dashboard Settings** and then click on **Create New Block** to add a new block to the dashboard.



Click the **Indicator block**. This will take you to the **Connect a Feed** window.

**Connect a Feed** ✕

A simple on/off indicator lamp. Feed values are compared using the given conditions. If the conditions are true, then "On Color" is used, if false, "Off Color". All values are assumed to be numeric for comparison. If the current feed value can't be converted to a number, it will be treated as a string.

Choose a single feed you would like to connect to this indicator. You can also create a new feed within a group.

Feed Name	Last value	Recorded	
<input type="checkbox"/> buzzer	0	1 day	🔒
<input checked="" type="checkbox"/> door	1	22 minutes	🔒
<input type="checkbox"/> humidity	21	3 days	🔒

1 of 1 feeds selected

## Pick Block Feed

Select the **door feed** and then click **Next step**.

## Configure Block

Configure its settings like so:

**Block settings** ✕

Block Title (optional)

On Color

Off Color

Conditions

Test Value



Indicator A simple on/off indicator lamp. Feed values are compared using the given conditions. If the conditions are true, then "On Color" is used, if false, "Off Color". All values are assumed to be numeric for comparison. If the current feed value can't be converted to a number, it will be treated as a string.

Set the **Block Title** to **Door Status**

**On Color** is green

**Off Color** is red

Conditions is set to **= 1**

This means the block will be **on (green)** when the Funhouse sends a **1** to the **door feed** and **off (red)** when the Funhouse sends a **0** to the **door feed**.

When you're done, click **Create block**.





---

# Email Alerts

Using Adafruit IO, you can create automated email alerts so you receive an email whenever a feed condition is met.

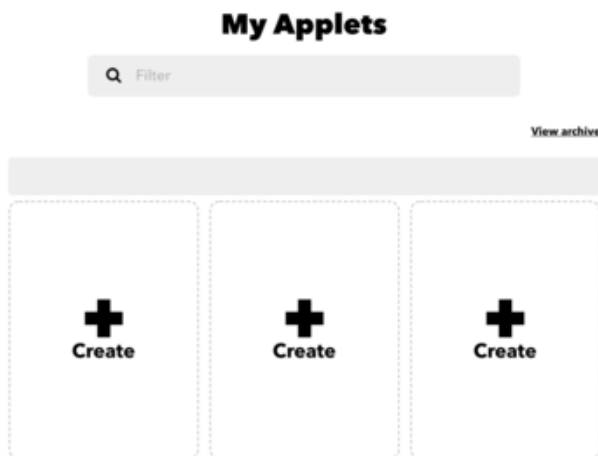
Here are two different ways to do it, one uses Adafruit IO and IFTTT, the other uses Adafruit IO+ (this is a paid service) with its built-in Actions.

Email alerts can take up to 15 minutes to be sent, so don't use this for time critical applications!

## Email Alert with AIO and IFTTT

The IFTTT (If This Then That) service can be integrated with AIO quite easily. You'll create an IFTTT applet that watches one of your AIO feeds, and when a certain value appears in the feed, IFTTT will send an email.

First, head to <https://ifttt.com/> (<https://adafru.it/iOe>) and create an account or log in if you already have one.



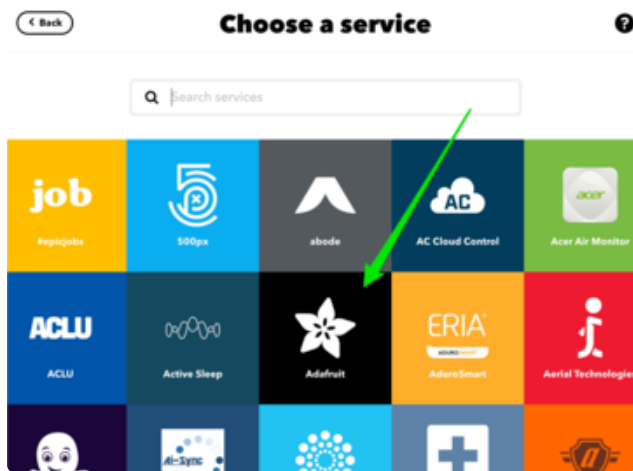
### Create Applet

On the IFTTT **My Applets** page, click the + **Create** button.



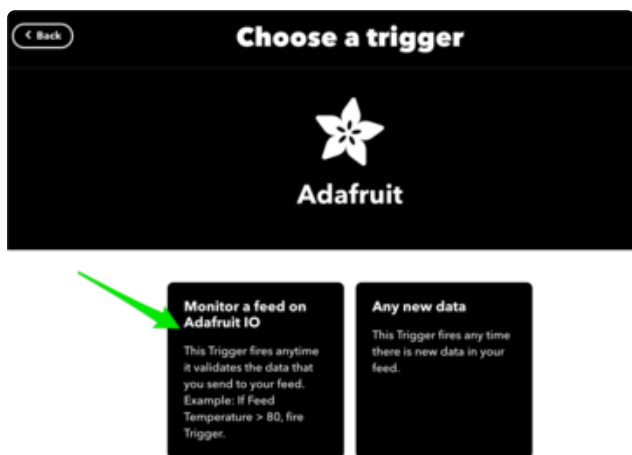
## Add a Condition

In the **If This** box, click the **Add** button.



## Choose Adafruit Service

From the **Choose a service** page, pick the **Adafruit** service.



## Choose Feed Trigger

On the **Choose a trigger** page, click on the **Monitor a feed on Adafruit IO** option. This option allows you to filter the data so only some conditions will trigger the applet.

**Complete trigger fields**

**Monitor a feed on Adafruit IO**

This trigger fires anytime it validates the data that you send to your feed. Example: If Feed Temperature > 80, fire trigger.

**Feed**  
  
The name of the feed to check.

**Relationship**  
  
Relationship between two values.

**Value**  
  
The value to compare against.

**Create trigger**

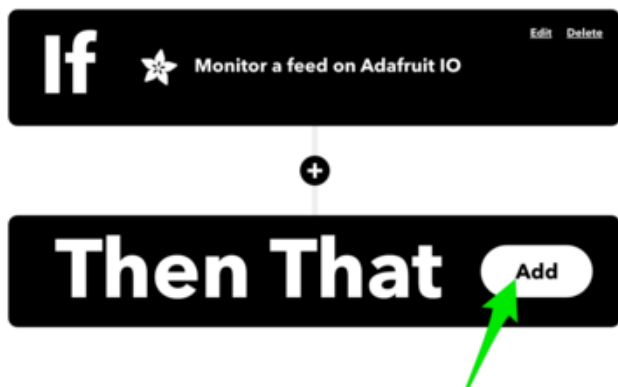
## Configure the Trigger

Fill out the fields of the **Monitor a feed on Adafruit IO** trigger.

In the example shown here, the **Feed** being monitored is named **door**, the **Relationship** is **equal to**, and the **Value** is **0**.

This means the trigger will run whenever the door feed has a new piece of data with a value of 0.


Once you've filled this out, click the **Create Trigger** button.



## Add an Action

Now that you have a trigger read, you will pick what the resulting action will be. Click on the **Then That** box's **Add** button.

**Choose a service**

  
Gmail

## Choose Gmail Service

Filter the services of the **Choose a service** page by typing **"gmail"** into the search bar.

Click on the **Gmail** icon to select it.



## Choose Gmail Action

In the Choose an action page choose to either **Send an email**.

## Email Fields

Here you can fill in the details of the email that'll be sent when the applet runs.

You can send the email to any address you like (just please don't spam people!)

You can even get fancy and have the email sent as a text message! For many carriers you can use your cellular number at txt . carrier . net pattern, e.g.

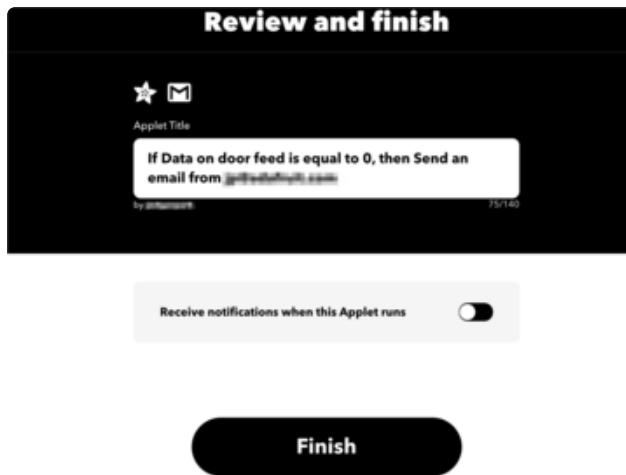
[2128675309@txt.att.net](mailto:2128675309@txt.att.net) and you'll receive a text message when the email is triggered.

Once filled in, click the **Create action** button.



## Applet Summary

This page shows a summary of the IFTTT applet flow. You can add elements here if needed, but in this case, click **Continue**.

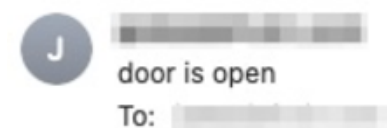
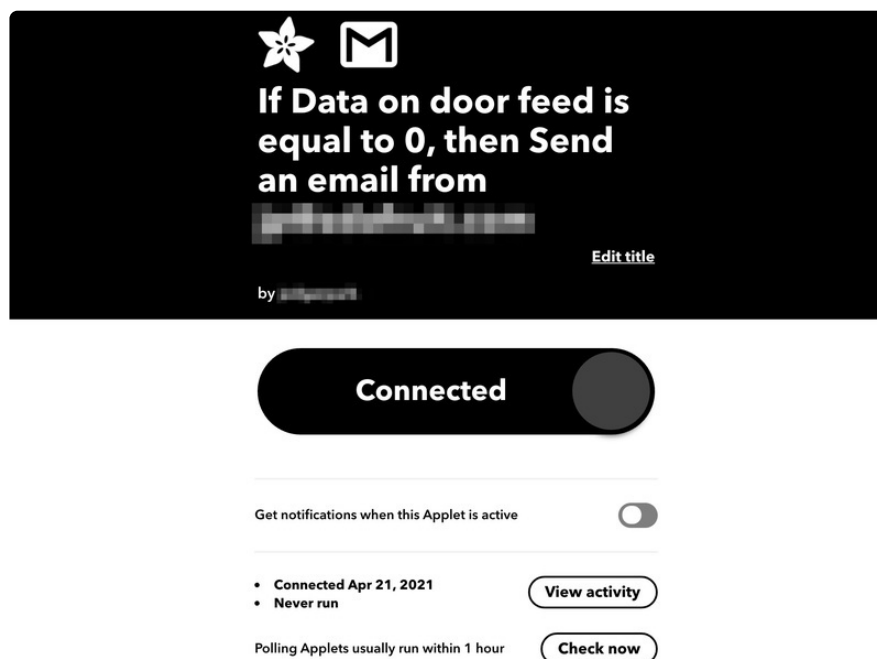


## Review and Finish

Here you'll see a review of the Applet. If everything looks good, click **Finish**.

If not, you can click the < **Back** button in the IFTTT window (not the browser back button).

Email Alert with AIO+



The 0 is equal to 0 at April 21, 2021 at 09:01AM!

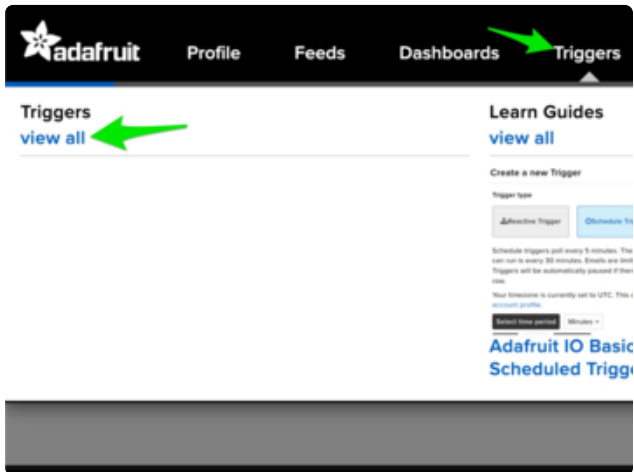
## Alert Received

Here's what a typical alert email looks like when the feed triggers the IFTTT applet.

# Adafruit IO+ Actions

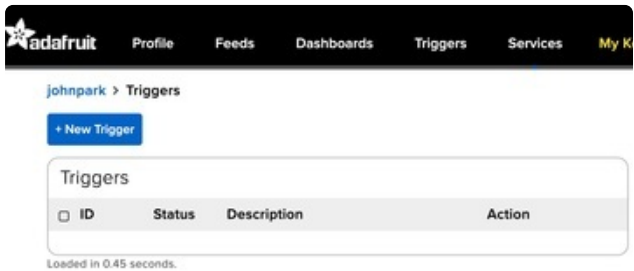
With a paid Adafruit IO+ subscription, you can use the Actions to send an email to your account email address.

Triggers were renamed to Actions in Adafruit IO.



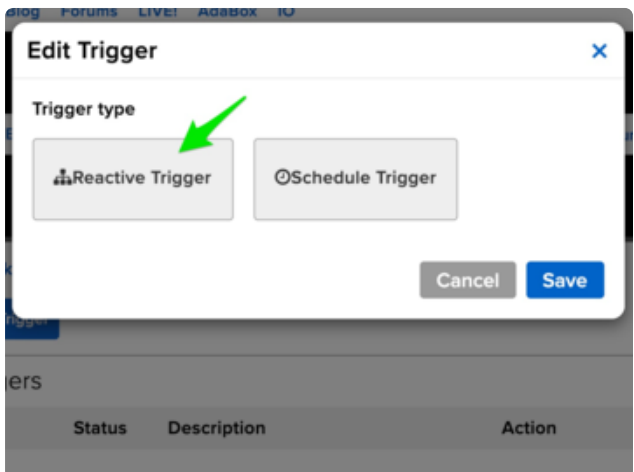
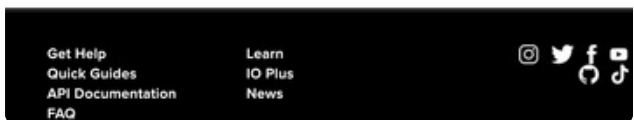
## Actions

On Adafruit IO, click the **Actions** header, and then click the **View all** link.



## New Action

Create a new action by clicking the + **New Action** button.



## Reactive Action


Click on **Reactive Action**.


Here, you'll fill out the conditions and action to take. In this example below:

**If door Is equal to 0 Then email me door value and time.**

**Create a new Trigger** ✕

**Trigger type**

 Reactive Trigger

 Schedule Trigger

Reactive triggers run after a feed is updated. They run, at most, every 5 seconds for io+ accounts, otherwise every 15 minutes. Reactive triggers that update a feed are limited to 1 update every 10 seconds. Emails are limited to io+ accounts. Triggers will be automatically paused if there are five failures in a row.

Your timezone is currently set to America/Los\_Angeles. This can be modified in [your account profile](#).

**If**

door

**Is**

equal to

Comparison Value or F...

0

**Then**

email me (IO+ only)

door

value and time.

Emailing the value of the feed will send the value at the point in time of the trigger running.

Cancel

Create

Click create and you're done! The next time your feed value equals the conditional number, you'll get an email.

---

## Code the Funhouse Door Alert

### Install CircuitPython

The first think to do is install CircuitPython on your Funhouse. [Follow this guide \(https://adafru.it/RMB\)](https://adafru.it/RMB) to get it set up with the latest release version.

### Shhhh... Secrets

In order for the Funhouse to connect to the internet, you'll need to include a **secrets.py** file on the board that contains your WiFi access point ssid and password, as well as your AIO key.

[Follow this guide page \(https://adafru.it/RMC\)](https://adafru.it/RMC) to get your **secrets.py** file set up.

## Text Editor

Adafruit recommends using the Mu editor for using your CircuitPython code with the Funhouse. You can get more info in [this guide](https://adafru.it/ANO) (<https://adafru.it/ANO>).

## Download the Project Bundle

Your project will use a specific set of CircuitPython libraries and the `code.py` file. In order to get the libraries you need, click on the **Download Project Bundle** link below, and uncompress the .zip file.



Next, drag the contents of the uncompressed bundle directory onto your microcontroller board's CIRCUITPY drive, replacing any existing files or directories with the same names, and adding any new ones that are necessary.

```
# SPDX-FileCopyrightText: 2017 Scott Shawcroft, written for Adafruit Industries
# SPDX-FileCopyrightText: Copyright (c) 2021 Melissa LeBlanc-Williams & John Park
for Adafruit
#
# SPDX-License-Identifier: MIT

import board
from digitalio import DigitalInOut, Direction, Pull
from adafruit_funhouse import FunHouse
from adafruit_debouncer import Debouncer

RED = 0x200000
GREEN = 0x002000

funhouse = FunHouse(default_bg=None)
funhouse.peripherals.dotstars.fill(RED)

switch_pin = DigitalInOut(board.A1)
switch_pin.direction = Direction.INPUT
switch_pin.pull = Pull.UP
switch = Debouncer(switch_pin)

def send_io_data(door_value):
    funhouse.peripherals.led = True
    print("Sending data to adafruit IO!")
    funhouse.network.push_to_io("door", door_value)
    funhouse.peripherals.led = False
```



```

send_io_data(0)

while True:

    switch.update()
    if switch.rose:
        print("Door is open")
        funhouse.peripherals.play_tone(2000, 0.25)
        funhouse.peripherals.dotstars.fill(RED)
        send_io_data(0)

    if switch.fell:
        print("Door is closed")
        funhouse.peripherals.play_tone(800, 0.25)
        funhouse.peripherals.dotstars.fill(GREEN)
        send_io_data(1)

```

## How it Works

The Funhouse Door Alert acts a bit like a simple switch that sends out a REST message when its position is changed. To do this, you first import some libraries.

### Libraries

```

import board
from digitalio import DigitalInOut, Direction, Pull
from adafruit_funhouse import FunHouse
from adafruit_debouncer import Debouncer

```

The `board` library provides pin definitions. `digitalio` is used to check the switch.

`adafruit_funhouse` allows simple access to the Dotstar LEDs, via the `peripherals` layer, as well as the `network` layer used to connect to the internet and send REST or MQTT messages (in the case of this project you'll use REST, which is simple and robust).

`adafruit_debouncer` make it easy to check for switch open and switch close events.

### Setup

Next you'll do some setup including setting LED color variables, creating the funhouse object, setting up the door sensor switch on the A1 pin, and creating the debouncer object on that pin.

```

RED = 0x200000
GREEN = 0x002000

funhouse = FunHouse(default_bg=None)
funhouse.peripherals.dotstars.fill(RED)

```

```
switch_pin = DigitalInOut(board.A1)
switch_pin.direction = Direction.INPUT
switch_pin.pull = Pull.UP
switch = Debouncer(switch_pin)
```

## Send IO Data Function

You'll create a function to send either a 0 or a 1 to the AIO door feed when the switch is opened or closed.

```
def send_io_data(door_value):
    funhouse.peripherals.led = True
    print("Sending data to adafruit IO!")
    funhouse.network.push_to_io("door", door_value)
    funhouse.peripherals.led = False
```

## Main Loop

The main loop of the program checks the switch pin with the debouncer's `switch.update()` function.

Two `if` statements do the rest. If the switch state rises, the door has just opened and the feed is sent a message with the data value `0`.

If the switch state falls, the door has just closed and the feed is sent a message with the data value `1`.

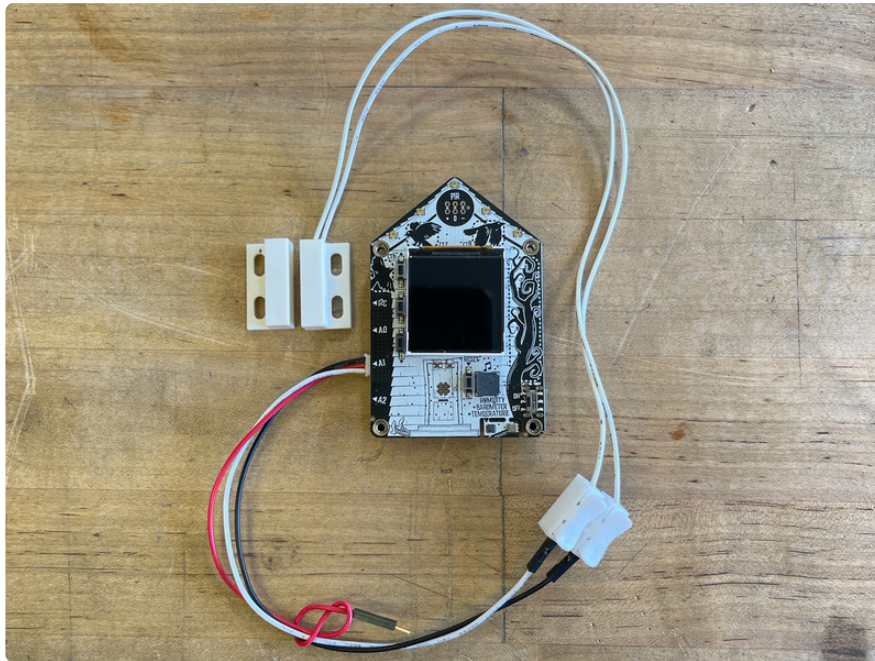
This configuration of triggering only on a state change is helpful, because it means the board won't be constantly spamming the AIO feed with data!

```
switch.update()
if switch.rose:
    print("Door is open")
    funhouse.peripherals.play_tone(2000, 0.25)
    funhouse.peripherals.dotstars.fill(RED)
    send_io_data(0)

if switch.fell:
    print("Door is closed")
    funhouse.peripherals.play_tone(800, 0.25)
    funhouse.peripherals.dotstars.fill(GREEN)
    send_io_data(1)
```

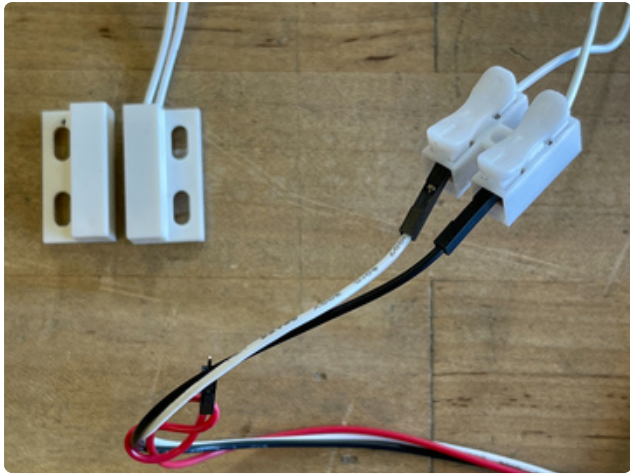
---

# Build the Funhouse Door Alert



## Door Sensor

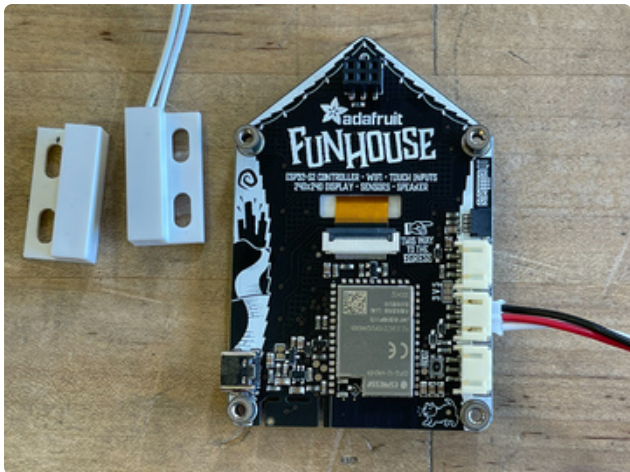
The door sensor is a type of switch called a reed switch. It consists of a wired switch and a separate magnet. When the magnet is within about 1/2" of the sensor, the internal switch mechanism is pulled open. When the magnet is further away the switch closes.



## Wire the Sensor

If your door sensor has a pair of bare wires, you'll need to join it to a JST-PH 3-pin cable.

Plug each wire of the door switch into the wire joint connector as shown. It doesn't matter which door switch wire is in which side.



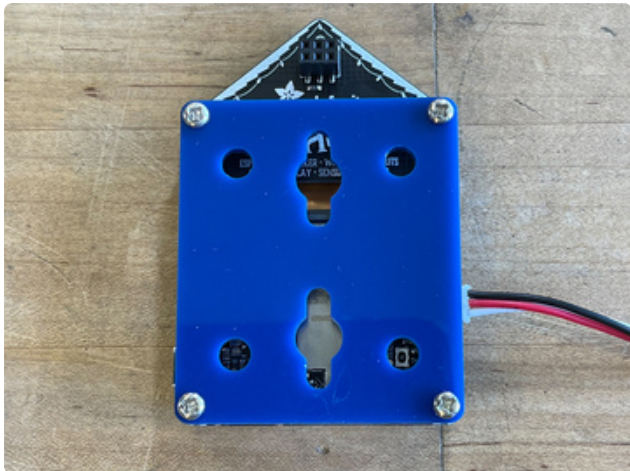
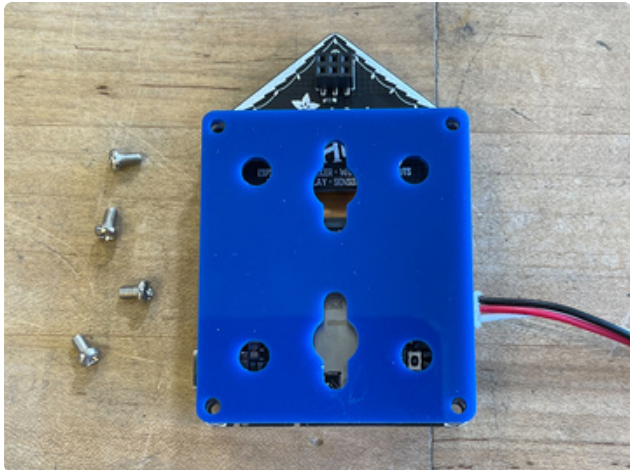
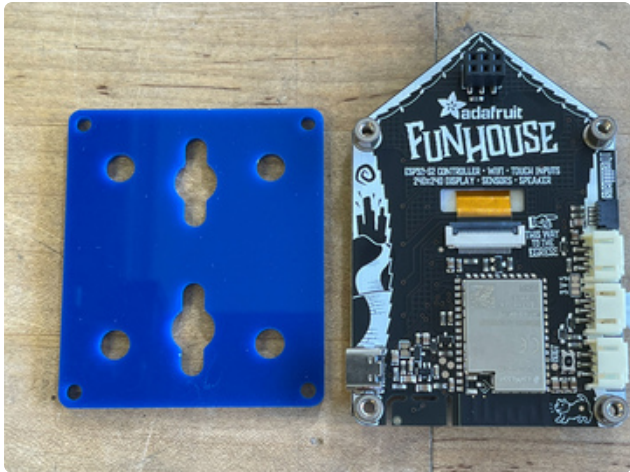
Then, plug the black (ground) and white (signal) wires into the other side of the joint connector as shown. You can leave the red wire alone or snip it off, it won't be needed. If you leave it connected, you can secure it to the wiring with a knot to avoid accidental shorts.

Plug the STEMMA connector into the A1 port of the Funhouse.

Here's a design file for a mounting bracket -- you can use it as a template for cutting your own by hand, with a laser cutter or mill, or as a jumping off point for modeling one for 3D printing.

[FunHouse-wall-mount.svg](https://adafru.it/Sfw)

<https://adafru.it/Sfw>



## Mount the Funhouse

Use the 3mm screw inserts and brackets to mount the Funhouse on the wall near the door, close enough for the door sensor wiring.





## Mount the Sensor and Magnet

Use the included screws or double-stick foam tape to mount the sensor on the door frame and the magnet to the door.

Plug the USB power into a wall outlet and power up the Funhouse.

Now, when the door is opened, you'll get an alert!

