FunHouse Mail Slot Detector

Created by John Park

https://learn.adafruit.com/funhouse-mail-slot-detector

Last updated on 2021-11-15 08:23:00 PM EST
# Table of Contents

## Overview
- Parts  
  Page 3

## Adafruit IO Setup
- Setting up Adafruit IO Feeds  
  Page 6
- Creating the Adafruit IO Dashboard  
  Page 7
- Email and Text Alerts  
  Page 9

## Code the FunHouse Mail Detector
- Install CircuitPython  
  Page 9
- Text Editor  
  Page 9
- Shhhh... Secrets  
  Page 9
- Download the Project Bundle  
  Page 10
- How it Works  
  Page 11

## Build the Mail Detector
- Break Beam  
  Page 14
- Connect to FunHouse  
  Page 15
- Sensor Adhesive  
  Page 16
- Secure to Mailslot  
  Page 17
Overview

Waiting for that very special piece of mail? This mail slot detector will alert you when the mail arrives in your mail slot. Using a break beam sensor, FunHouse running CircuitPython, and Adafruit IO, you'll get an alert when the mail has been delivered.

Parts

Adafruit FunHouse - WiFi Home Automation Development Board
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
IR Break Beam Sensors with Premium Wire Header Ends - 3mm LEDs
Infrared (IR) break-beam sensors are a simple way to detect motion. They work by having an emitter side that sends out a beam of human-invisible IR light, then a receiver across the...
https://www.adafruit.com/product/2167

2 x STEMMA JST PH 3-Pin to Female Socket Cable  https://www.adafruit.com/product/3894
200mm

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...
Clear Adhesive Squares - 6 pack
UGlu Dashes are perfect for a variety of small projects. These adhesive squares provide a stronger bond to most surfaces and are cleaner and easier to...
https://www.adafruit.com/product/4813

Pre-Cut Multi-Colored Heat Shrink Pack Kit - 280 pcs
Heat shrink is the duct tape of electronics which we guess makes this heat shrink the colorful and exciting duct tape they sell at craft stores. This pack contains heat...
https://www.adafruit.com/product/4559

Adafruit IO Setup

New to Adafruit IO? You can read all about it in this guide (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) 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.
Setting up Adafruit IO Feeds

You'll create a feed called mail 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://adafruit.io/mxC) and click Actions->Create a New Feed. Name the new feed door and then click Create.

If you do not know how to create feeds, head over to the Adafruit IO Basics: Feeds for a quick overview (https://adafruit.io/A) of this process.
Creating the Adafruit IO Dashboard

Next, you'll create an Adafruit IO Dashboard to display and control the feeds. Navigate to the Adafruit IO Dashboard page (https://adafruit.io/eIS) and click Actions -> Create a New Dashboard.

Name this dashboard Funhouse Mail Alert and click Create. You'll be redirected to the new Dashboard.

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.

Pick Block Feed
Select the mail feed and then click Next step.

Configure Block
Configure its settings like so:

- Set the Block Title to Mail Slot Status
- On Color is blue
- Off Color is amber
- Conditions is set to $= 0$

This means the block will be on (blue) when the Funhouse sends a 0 to the mail feed and off (amber) when the Funhouse sends a 1 to the mail feed.

When you're done, click Create block.
Email and Text Alerts

To set up Email and Text alerts for your feed, [check out this page](https://adafruit.it/Scc)!

---

**Code the FunHouse Mail Detector**

**Install CircuitPython**

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

**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://adafruit.it/ANO).

**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) to get your secrets.py file set up.

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.

```python
# SPDX-FileCopyrightText: Copyright (c) 2021 John Park for Adafruit
# SPDX-License-Identifier: MIT

# FunHouse Mail Slot Detector

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

beam_sense_pin = DigitalInOut(board.A0)  # defaults to input
beam_sense_pin.pull = Pull.UP  # turn on internal pull-up resistor
beam_sensor = Debouncer(beam_sense_pin)

AMBER = 0xF0D000
BLUE = 0x00D0F0
RED = 0xFF0000
WHITE = 0xFFFFFF
GRAY = 0x606060

funhouse = FunHouse(default_bg=None, scale=3)
funhouse.peripherals.dotstars.brightness = 0.05
funhouse.peripherals.dotstars.fill(AMBER)

# Create the labels
funhouse.display.show(None)
mail_label = funhouse.add_text(
    text="No Mail yet", text_position=(4, 14), text_color=AMBER
)
reset_label = funhouse.add_text(text="reset", text_position=(3, 70),
    text_color=GRAY)

funhouse.display.show(funhouse.splash)

def send_io_data(mail_value):
    funhouse.peripherals.led = True
    funhouse.network.push_to_io("mail", mail_value)
    funhouse.peripherals.led = False

send_io_data(1)

while True:
    beam_sensor.update()
    if beam_sensor.fell:
        funhouse.peripherals.set_dotstars(RED, WHITE, BLUE, WHITE, RED)
        funhouse.peripherals.play_tone(2000, 0.25)
        funhouse.set_text("Mail is here!", mail_label)
        funhouse.set_text_color(BLUE, mail_label)
        send_io_data(0)

    if funhouse.peripherals.button_down:
```

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

How it Works

The Funhouse Mail Detector acts a bit like a simple switch that sends data to Adafruit IO when it detects a change. To do this, you first import some libraries.

Libraries

```python
import board
from adafruit_debouncer import Debouncer
from digitalio import DigitalInOut, Pull
from adafruit_funhouse import FunHouse
```

The `board` library provides pin definitions. `digitalio` is used to check the switch. `adafruit_debouncer` make it easy to check for switch open and switch close events.

`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 messages to Adafruit IO.

Setup

Next you'll do some setup including setting up the break beam sensor on the A0 pin, creating the debouncer object on that pin, setting LED color variables, and creating the funhouse object.
beam_sense_pin = DigitalInOut(board.A0)  # defaults to input
beam_sense_pin.pull = Pull.UP    # turn on internal pull-up resistor
beam_sensor = Debouncer(beam_sense_pin)

AMBER = 0xF0D000
BLUE = 0x00D0F0
RED = 0xFF0000
WHITE = 0xFFFFFFFF
GRAY = 0x606060

funhouse = FunHouse(default_bg=None, scale=3)
funhouse.peripherals.dotstars.brightness = 0.05
funhouse.peripherals.dotstars.fill(AMBER)

FunHouse Display

You'll set up the FunHouse display with text labels to show the state of the mail slot,
and to label the reset button.

# Create the labels
funhouse.display.show(None)
mail_label = funhouse.add_text(
    text="No Mail yet", text_position=(4, 14), text_color=AMBER
)
reset_label = funhouse.add_text(text="reset", text_position=(3, 70),
    text_color=GRAY)
funhouse.display.show(funhouse.splash)

Send IO Data Function

You'll create a function to send either a 0 or a 1 to the AIO door feed when the
beam is broken or reset.

def send_io_data(mail_value):
    funhouse.peripherals.led = True
    funhouse.network.push_to_io("mail", mail_value)
    funhouse.peripherals.led = False

Main Loop

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

Two if statement do the rest. If the beam sensor state falls, the mail has arrived and
the feed is sent a message with the data value 0.

If the reset button is pressed, 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!

Build the Mail Detector
Break Beam
The mail slot detector uses a type of sensor called a break beam sensor. It is made of two parts, an emitter and a sensor. When the sensor has a clear line of sight to the emitter, everything is fine, when the beam is broken, it'll trigger the code to send an alert.

The emitter is an infrared (IR) LED that is always on. It has a black wire and a red wire that connect to ground and power, respectively.

The sensor has three wires for ground, power, and signal.
Connect to FunHouse
To connect the break beam sensor to the FunHouse you'll use a couple of JST-PH cables.

You can plug them in directly as shown above, simply match the colors of each wire.

For a more secure fit, a piece of heat shrink tubing is helpful. Add a bit to each bundle for wire management, and a larger piece to secure the connections.

Note one white wire will remain unused.
Sensor Adhesive

Use double-stick tape to secure the sensors inside the mailbox. Uglu Dashes (https://adafru.it/OzA) work great for this.
Secure to Mailslot
This step can vary a lot depending on the type of mail slot you have.

Use the adhesive to affix the sensor and the IR emitter so that their beam will be broken when mail enters the slot.
Mount the FunHouse with a mounting plate and screw, or strong adhesive square, plug it in to power and you're ready for business. Postal business.