Dauntless Dotstar Gauntlets

Created by Erin St Blaine

https://learn.adafruit.com/dotstar-dauntless-gauntlets

Last updated on 2024-06-03 01:55:00 PM EDT

Table of Contents

Introduction

In a world where darkness runs rampant..

Where villains conspire and criminals run the streets...

One Maker has emerged to repel the dark. Gauntlets ablaze, leaping tall buildings in a single bound and making our neighborhoods safe again.

Dauntless. Dotstar. Gauntlets.

They're counting on you. What are you waiting for?

How they Work

Record a single video, a split-video or a whole video playlist to an SD card using your computer and a Processing script.

Pop an SD card into each gauntlet and you'll have hours of fascinating LED animation on your wrists, with no additional coding required. Split a video between the two gauntlets or play the same one on both.

Brightness and playlist control buttons make it easy to sync the gauntlets up or play just the right light combo for any situation.

Electronic Bits

Get 2 of everything for 2 gauntlets. Add them to your cart over on the right side of this page.

- Feather M0 Adalogger (http://adafru.it/2796)
- Dotstar 144/m strip (http://adafru.it/2242)
- 1200mAh Lithium Polymer battery (http://adafru.it/258)
- JST battery extension cable (http://adafru.it/1131)
- On/off switch (http://adafru.it/805)
- 3 x Momentary tactile buttons (http://adafru.it/1119)
- microSD card (http://adafru.it/102)

Crafty Bits

- 2-3 sheets of Craft foam
- Fabric or leather (or paint if you prefer)
- Fabric for Lining
- Craft weight fusible interfacing
- 1/4" grommets & grommet setting tools (https://adafru.it/nzE)
- Diffusion material -- I used sticky shelf liner
- Jewels or findings or rivets or other decorative schwag
- Elastic or leather lacing
- NO CAPES!

Tools & Helpful Materials

- Soldering iron & accessories
- Hot glue gun
- E6000 glue
- Spray glue
- Iron
- Heat gun
- Sewing machine (helpful but not absolutely neccessary)
- Hole punch

Design & Testing

There are three major parts to this guide:

1. **Setting up the software**
2. **Building the electronics**
3. **Building the gauntlets**

All three parts are interrelated...so take some time, do some sketches, read all the way through and do some planning before you start...

- **How many LEDs do you want to use?** More LEDs likely means a shorter battery life, but a bigger matrix will show image detail better
- **Where will the battery fit** most comfortably along your arm?
- Will the Feather board work better if it's placed **horizontally** or **vertically?**
- Do you plan to regularly add and update your SD card files? If so, **can you reach your SD card easily?**
- Do you want to be able to **swap out batteries**, or is stopping to charge your gauntlet with the battery **in-place** good enough?
- **What super power(s) will these gauntlets give you?** Will humanity as a whole benefit from your actions?

Software Setup

The technical parts of this guide build upon our Lightship LED Animation guide **(https://adafru.it/mf5)**...but rather than WiFi, these gauntlets play back animation from a micro SD card*. **Skim through that guide to familiarize yourself with all the pieces and terminology.** Though this project is built a little differently, many of the same concepts apply, and this saves us a lot of explanation here.

* Though if you'd like WiFi gauntlets instead, that's entirely 100% possible...substitute a Feather M0 WiFi rather than an Adalogger...same concepts apply, just use the original Lightship code instead.

Steps you'll need to follow:

1. Install Feather board support in your Arduino IDE (https://adafru.it/ldF) and get the "blink" sketch working.

2. Download and install the necessary libraries (https://adafru.it/nRF): you'll need the Adafruit Dotstar library, the ASF Core library and the Adafruit_ZeroDMA library.
3. Install Processing (https://adafru.it/cK1) vesrion 2.2.1
4. Download and extract the ZIP file (https://adafru.it/nRF) containing the code for this project.

Do not continue with the build until you have the “blink” sketch working on the Feather M0 board.

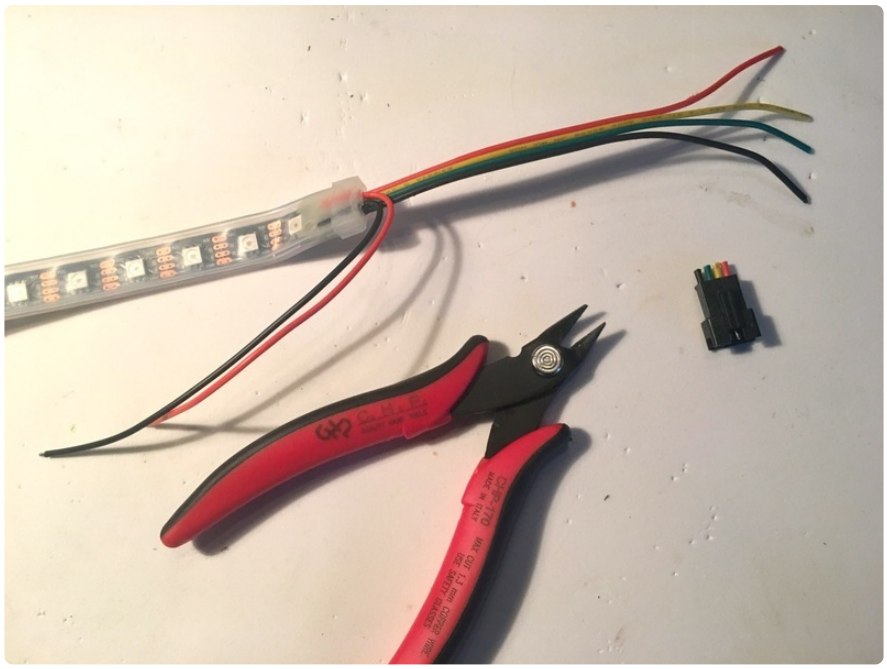
Set Up a Test Strip

Testing your LED strip first can save heartbreak later...this lets you know whether the whole strip is in good working order and that you’re making the right connections.

Dotstar strips come with a connector pre-soldered onto the ends. For testing purposes I like to just cut the connector head off and use the wires to connect directly to my project...or you can trim the plug with wires off the tail end and use that as a connector for the head end.

When connecting to a microcontroller, make certain you’re working from the **INPUT** end of the LED strip! If you look closely at the face of the strip, you’ll see a series of small arrows pointing from IN toward OUT. Connect to the “in” end...the arrows should be pointing away from the microcontroller.

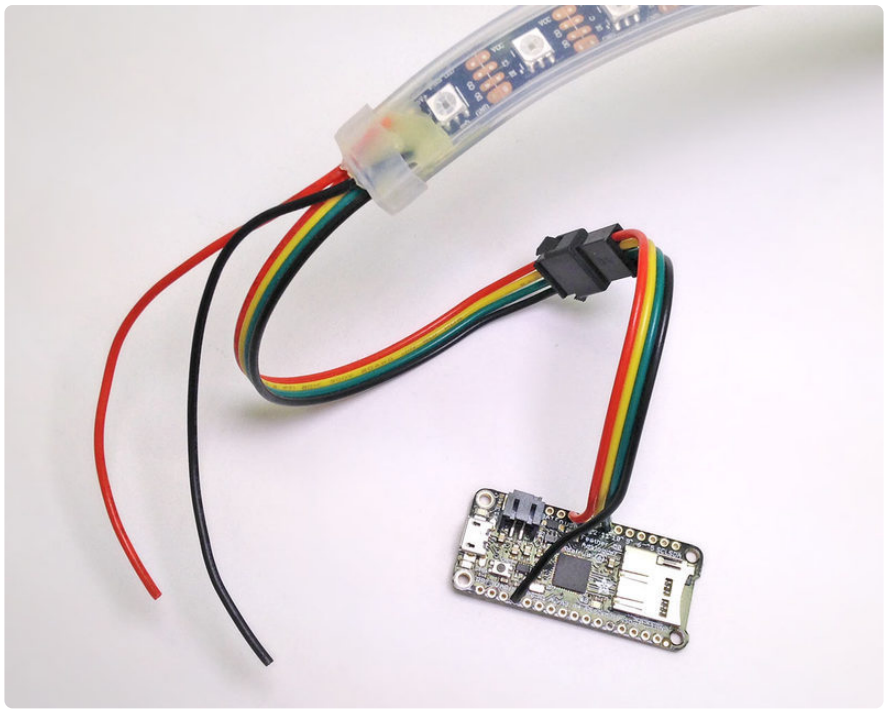
Tiny solder pads on the strip should be marked + or **5V**, **D or DI** (data in), **C or CI** (clock in) and **–** or **G** (ground) or similar. The wires on the end of the strip will be in the same order as the labels; typically a red wire for +5V, black for ground.



Temporarily solder your strand of Dotstars to the Feather M0 board:

Feather M0 Pin	DotStar
11	D or DI (data in)
13	C or CI (clock in)
USB	+ or 5V
GND	– or G

There are spare power wires at the ends of the strip. Make sure the ends of these are covered (or just clip off) so they don't accidentally contact anything and cause a short.



Later we'll power the LEDs off a battery, but for testing, use the USB and GND pins.

Use the Arduino Library Manager to install the Adafruit_DotStar library (Sketch→Include Library→Manage Libraries...), or if you're using an older version of the Arduino IDE, it can be downloaded and installed manually (https://adafru.it/dNR):

Adafruit_Dotstar Library

https://adafru.it/eio

Library installation is a frequent stumbling block...if you need assistance, our All About Arduino Libraries (https://adafru.it/dit) guide spells it out in detail.

Once the Adafruit_DotStar library is installed, you can use the "strandtest" sketch to confirm that everything works.

File→Examples→Adafruit_DotStar→strandtest

You'll need to make a few small changes in the code before uploading to the board. First, edit the number "30" on this line to reflect the actual number of LEDs in your DotStar strip (e.g. 60, 144, etc.):

```
#define NUMPIXELS 30 // Number of LEDs in strip
```

Then a couple lines down, look for these:

```
#define DATAPIN    4
#define CLOCKPIN   5
```

Change these values to:

```
#define DATAPIN    11
#define CLOCKPIN   13
```

Make sure the right board type and serial port are selected, then upload this sketch to the Feather board. After a moment, you should see a set of red pixels chase along the strip, then green, then blue.

The code uploads OK, but no LEDs light!

- Double-check all the connections between board and strip: Feather pin 11 to strip DI, 13 to CI, USB to + and GND to –.
- Make sure you're connected to the input end of the strip (arrows should point away from the microcontroller board).
- Check that the spare power wires at each end of the LED strip aren't touching each other or anything conductive around your desk.

LEDs are lit, but the order is not red-green-blue.

Occasionally from batch to batch the manufacturer changes the order in which the LEDs receive color data. You can reorder this data using the last argument to the Adafruit_DotStar() constructor, e.g. try:

```
Adafruit_DotStar strip = Adafruit_DotStar(
  NUMPIXELS, DATAPIN, CLOCKPIN, DOTSTAR_BGR);
```

Do not continue with the build until you have the "strandtest" sketch working and LEDs animating.

Download Software

With the basic blink and strandtest now tested, let's install some additional stuff...

Download and install the Adafruit Arduino Zero ASF Core **(https://adafru.it/lna)**and Adafruit_ZeroDMA **(https://adafru.it/lnb)** libraries. This is most easily done through the **Arduino Library Manager**...but if you're oldschool, here's the links:

Download ASF Core Library from Github

https://adafru.it/lnc

Download Adafruit_ZeroDMA library for Github

https://adafru.it/lnd

Then download and extract the ZIP file containing the code for this project:

Download Adafruit_Lightship from Github

https://adafru.it/ldG

In this archive are two folders:

The first, "**Arduino**," contains the **OPCstreamSD** sketch for the Feather M0 board (ignore the OPCserver sketch, it's not used in this project).

The second, "**Processing**," contains several Open Pixel Control demos for use with the Processing programming language. These will run on your main computer (desktop or laptop).

OPC clients can be written in many programming languages, but we'll use **Processing** (a derivative of Java) as it's free, cross-platform (runs on Windows, Mac and Linux) and is focused on visual arts.

Processing Downloads Page (https://adafru.it/cK1)

For now, we recommend downloading the **version 2.2.1 release** of Processing for your operating system. The 3.X series introduced some significant changes that aren't always compatible with existing Processing code.

Processing looks a lot like the Arduino IDE (in fact, the Arduino IDE derived from the same code base). This can be confusing because Arduino sketches don't work in Processing, nor vice versa. Make sure you're loading sketches into the correct IDE for each.

Upload Code

Arduino Code

Open **OPCstreamSD** from the Arduino folder and upload it to the board.

This code tells the Feather to scan the root directory of your SD card for any file ending in ".opc". These files will be generated by our Processing code (in a moment) using video files (MPEG, AVI, etc.) you provide.

This sketch won't do anything useful yet, since we haven't assembled the gauntlets, but it's handy to have it pre-loaded on the M0 board so it's ready for testing when the time comes.

Processing Code

We'll need to make a few minimal changes to the Processing code before we start creating our files.

Open the **OPCvideoSplitFiles** sketch in Processing. This code will split the video into a top half and a bottom half so you can play one video file that will be split between two gauntlets.

If you're making just one gauntlet, or if you want the same animation running on both, comment out the second OPC instance:

```
OPC    opc[]        = { new OPC(this, 30, "foldername/left/L01.opc"),
                        new OPC(this, 30, "foldername/right/R01.opc") };
```

Change the filepaths to point to some dedicated folders on your hard drive, and the filenames to something descriptive.

Find these lines of code and update them to reflect the planned width and height of your LED matrix:

```
int    arrayWidth  = 16, // Width of LED matrix
       arrayHeight =  5, // Height of LED matrix
```

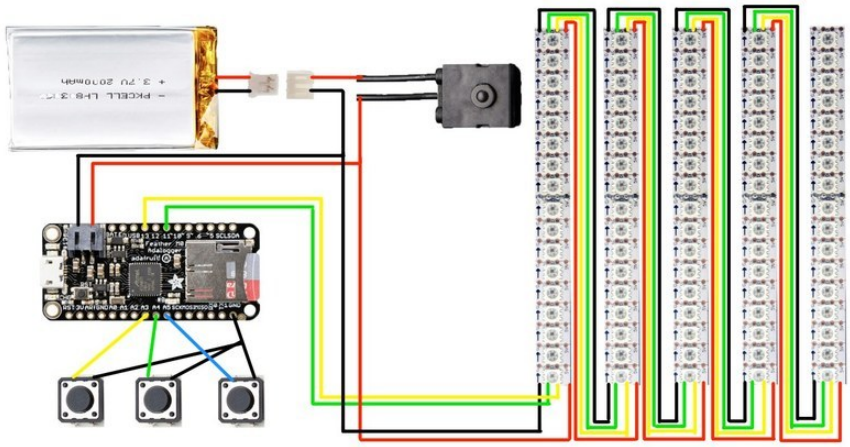
Then, click the "run" button and choose a video file. The Processing script will convert the file you choose into an .opc file and save it to the path you've specified.

Copy the files onto an SD card. Don't use subfolders -- the server code is looking right at the root of the card.

Pop the card out of your computer and into your Feather, power it up and make sure you get some blinky lights happening on your test strip.

Wiring

In schematic form, here's what the circuit will look like:



Some LED strips change the sequence of the four wires. **Always refer to the labels on the LED strip** for the pin functions, don't just blindly follow the order shown here... yours might come from a different batch.

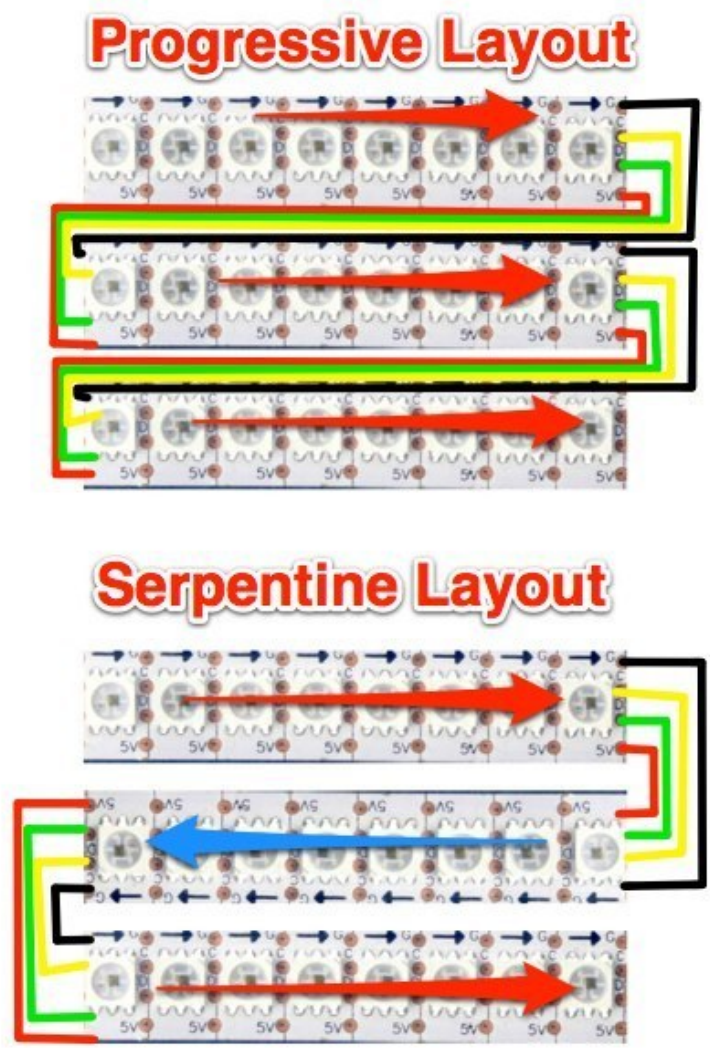
For the momentary switch buttons, connect one leg of each to **GND**, then connect the other leg to **A3**, **A4** and **A5** respectively.

Dotstar Wiring will be the same as your test strip:

- Dotstar Clock --> Feather **#13**
- Dotstar Data --> Feather **#11**
- Dotstar G --> Battery **GND**
- Dotstar + --> Battery +

A little later (on the "Matrix Build" page) the rubber sleeve will be removed from the Dotstar strip and we'll cut it into sections to create a two-dimensional matrix.

My Dotstars are wired in a "progressive" layout, rather than a "serpentine" layout, since that worked better in my design.



The code defaults to serpentine layout, so if you wire your LEDs in a serpentine fashion you don't need to make any changes to the code.

If you use a progressive layout, like me, look in the Processing **OPCvideo** script and find this section of code:

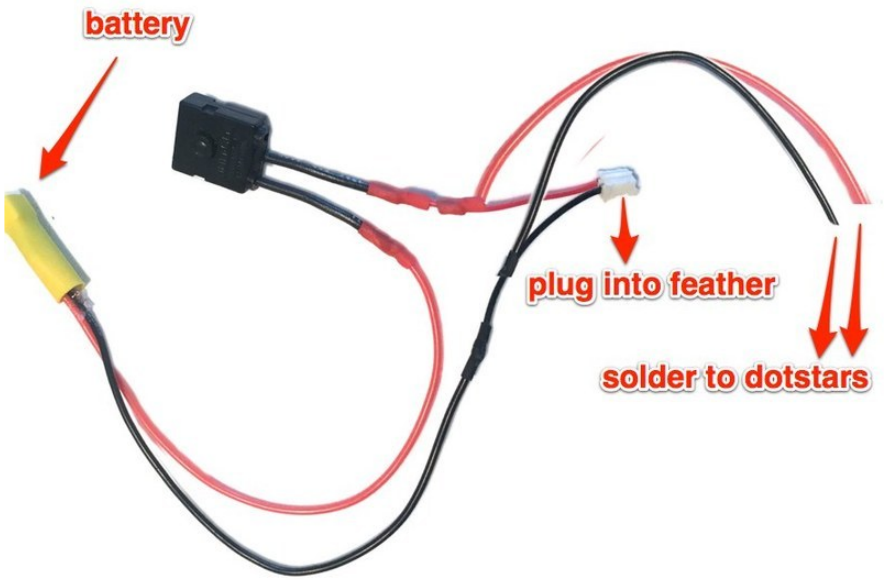
```
// Set up OPC pixel grid.  Arguments are: 1st pixel index,
  // row length, # of rows, center x, y, horizontal &amp; vertical
  // pixel spacing, angle (radians), 'zigzag' flag (true/false):
  opc.ledGrid(0, arrayWidth, arrayHeight, (width - 1) / 2,
    (height - 1) / 2, scale, scale, 0, true);
```

For a serpentine layout, leave this code alone. For a progressive layout, change the last parameter to "false".

If you find your video comes out "flipped" backwards on your matrix, add a "-" before the first scale parameter to flip it back:

```
// Set up OPC pixel grid.  Arguments are: 1st pixel index,
  // row length, # of rows, center x, y, horizontal &amp; vertical
  // pixel spacing, angle (radians), 'zigzag' flag (true/false):
  opc.ledGrid(0, arrayWidth, arrayHeight, (width - 1) / 2,
    (height - 1) / 2, -scale, scale, 0, false);
```

Buttons & Switch



On/Off Switch

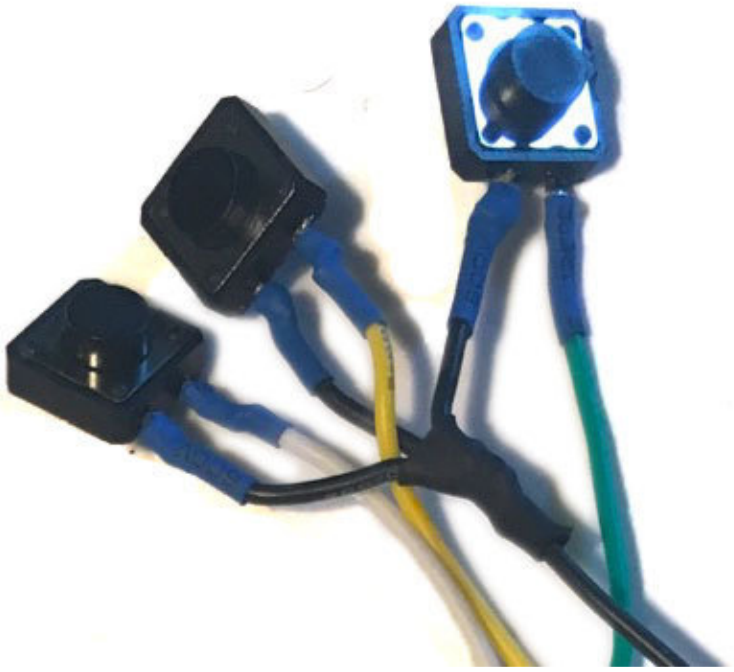
Solder your on/off switch in line with your battery extension cable. This cable will provide power to both the feather and the LEDs, so solder in a couple more wires to split the power.

This is also a good time to shorten the extension cable to an appropriate length that will plug into your Feather and your battery when they're in place.

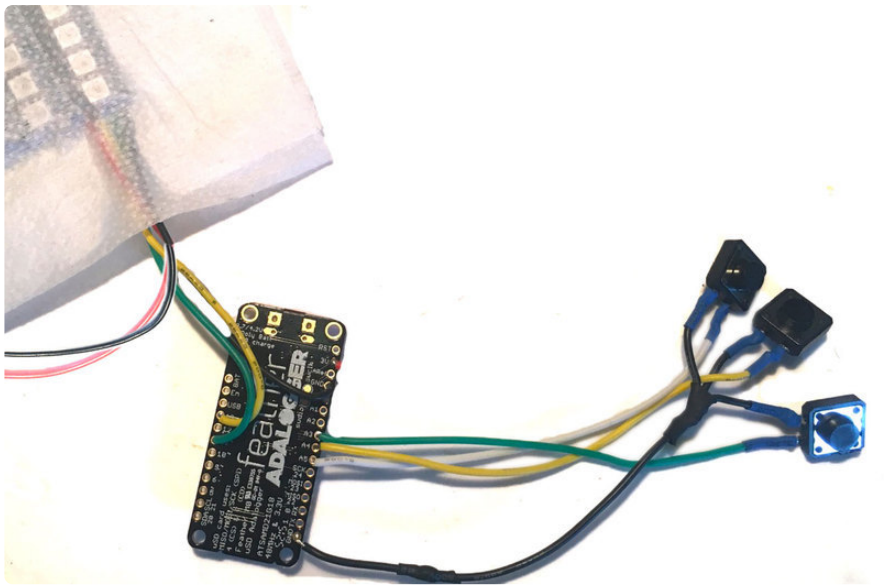
Add some heat shrink filled with glue to the female end of the extension cable, since this will get pulled when you remove your battery for charging.

Get out your 3 tactile push buttons. Gently fold down two of the legs on one side -- we won't be using these and you don't want them poking you through your gauntlet.

Securely solder a black wire to one leg of each button and a colored wire to the other leg. Gently bend these down to add stability and minimize pokiness.



Twist the 3 black wires together and splice in a 4th black wire. This will run to the Feather's remaining G pin.



Connect each colored wire to your feather's **A3**, **A4** and **A5** pins (also known as digital pins 17, 18 and 19).

Matrix Build

Spend a few minutes planning out the shape of your LED matrix. I have fairly narrow wrists and wanted a very sleek design, so my matrix ended up 16 pixels long and 5 strips wide.

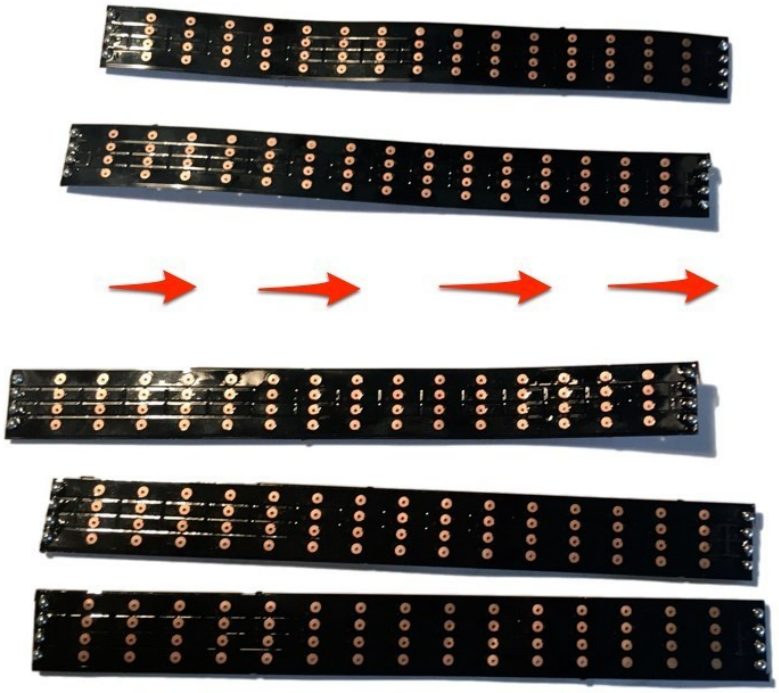
I made my matrix rectangular just to keep the project a little simpler, but this code will work with any shape you want to create, with a little tweaking in the Processing script setup.

You can also use a premade Dotstar matrix (https://adafru.it/nzB). However, a gauntlet will need to move quite a lot and these matrices are bendy but not unbreakable. I found that creating my own matrix was a little more cost-effective and flexible in terms of size and shape.

Slide your dostars out of the silicone sleeve and carefully cut each strip to length.

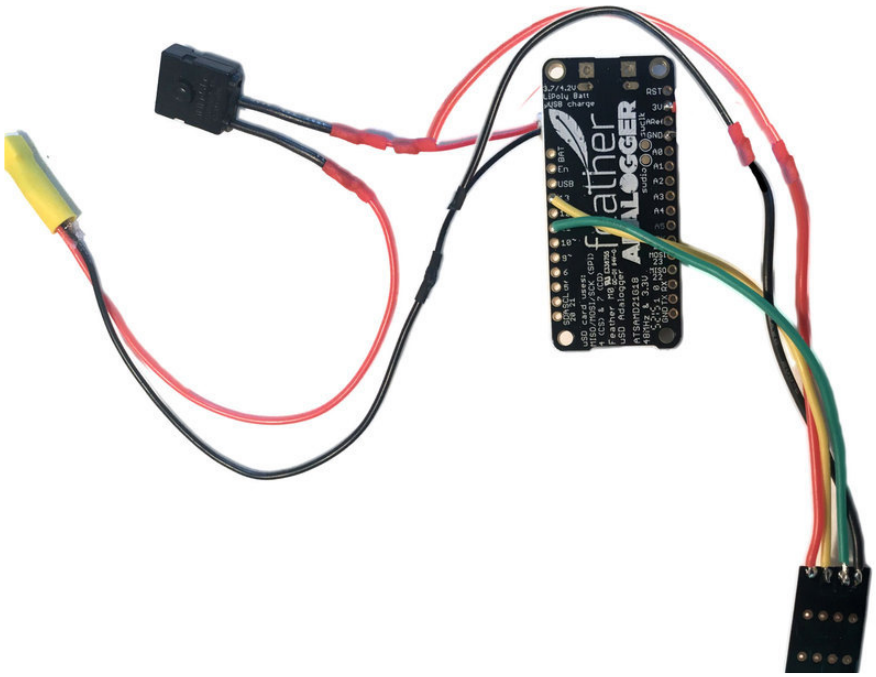
I found it easiest to use the "sacrificial pixel" method: instead of cutting through the middle of the teeny tiny solder pads, cut so you leave full pads on each end of your strip.

This will render one pixel out of each section useless but makes the whole project much easier to assemble and a lot more robust in the long run (the full pads have more contact area for solder).



Lay out your strips so that the bit-direction arrows are all pointing the same direction.

Turn the strips over and tin each pad on the back of both ends of all strips.



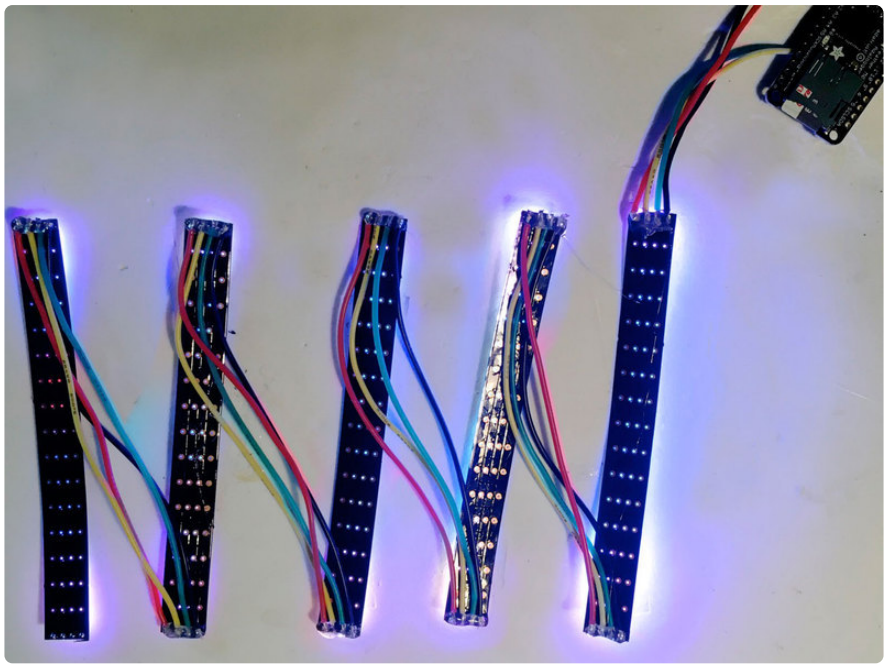
Find the "in" end of your first strip and solder a 4" wire (or whatever length your design demands) to each of the 4 tinned pads.

Connect the other end of these wires to your Feather:

- Dotstar D --> Feather **#11**
- Dotstar C --> Feather **#13**

Connect the power wires to your battery extention cable.

Turn your Feather on and make sure the LEDs in this first strand light up.



Next, solder carefully measured wires between the "out" pads of your first strip and the "in" pads of your second strip. Solder them in the direction shown to avoid any wire loops or places that could get pulled.

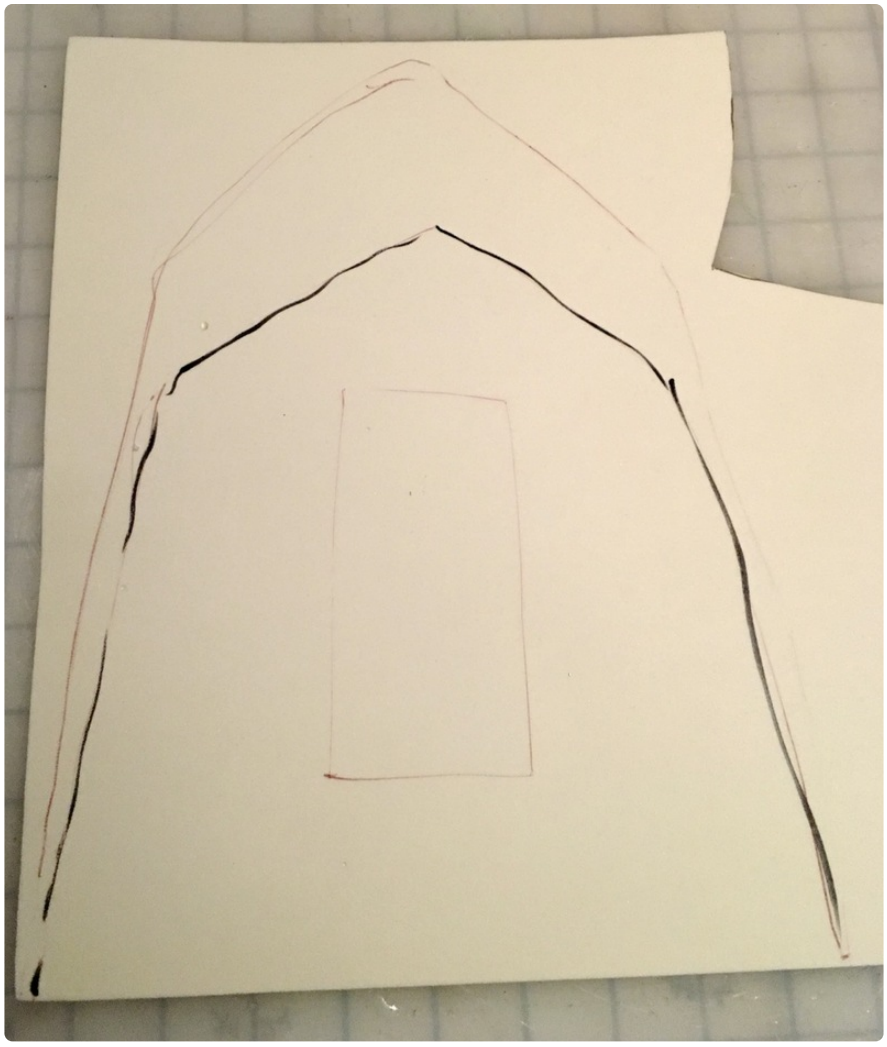
Repeat with your other strips, testing each strip as you go. Once you've got the whole matrix built, cover each wire joint with hot glue to secure it in place.

Turn your matrix on and flex it back and forth a bit to make sure all your solder joints are rock solid.

Gauntlet Build

Every superhero has a unique style and very different needs, so every set of gauntlets should uniquely reflect those needs. There's no wrong way to build your gauntlets.

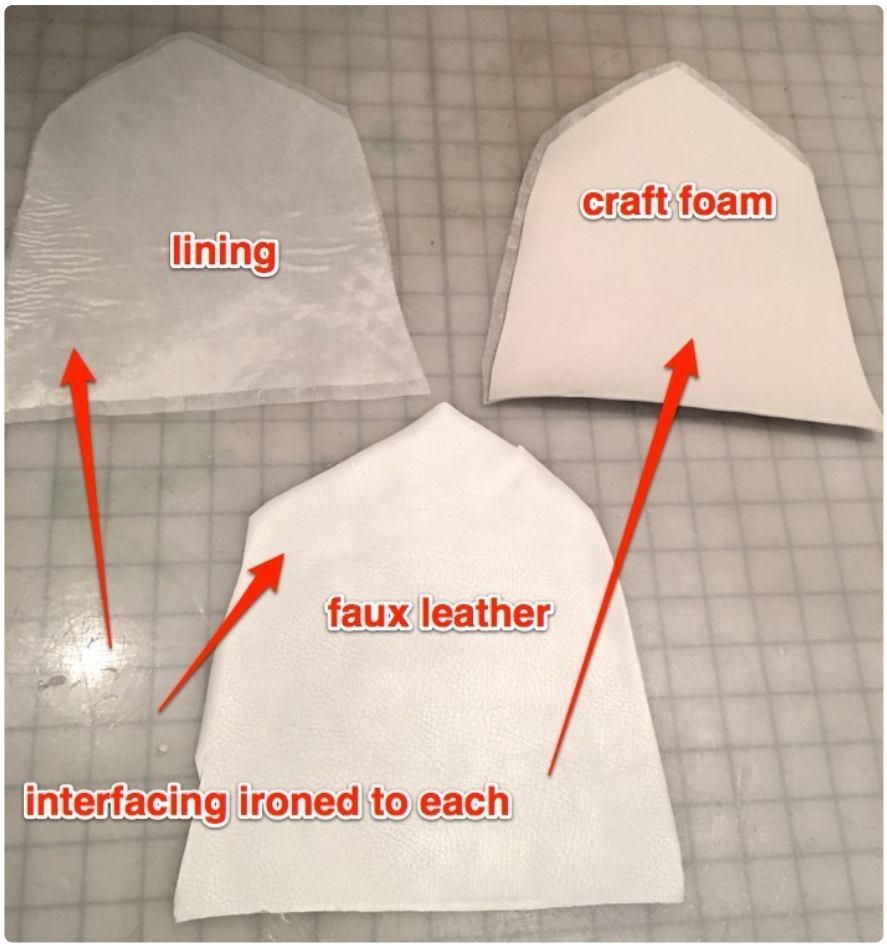
Use Leather, Worbla, craft foam, fabric, 3d printing, or whatever materials you desire. There are dozens of great gauntlet tutorials out there. I'll share some methods and tips that worked for me.



Create your Pattern

The easiest way to get a good fit is to wrap some paper or foam around your arm and hold it there while you scribble out a pattern with a sharpie.

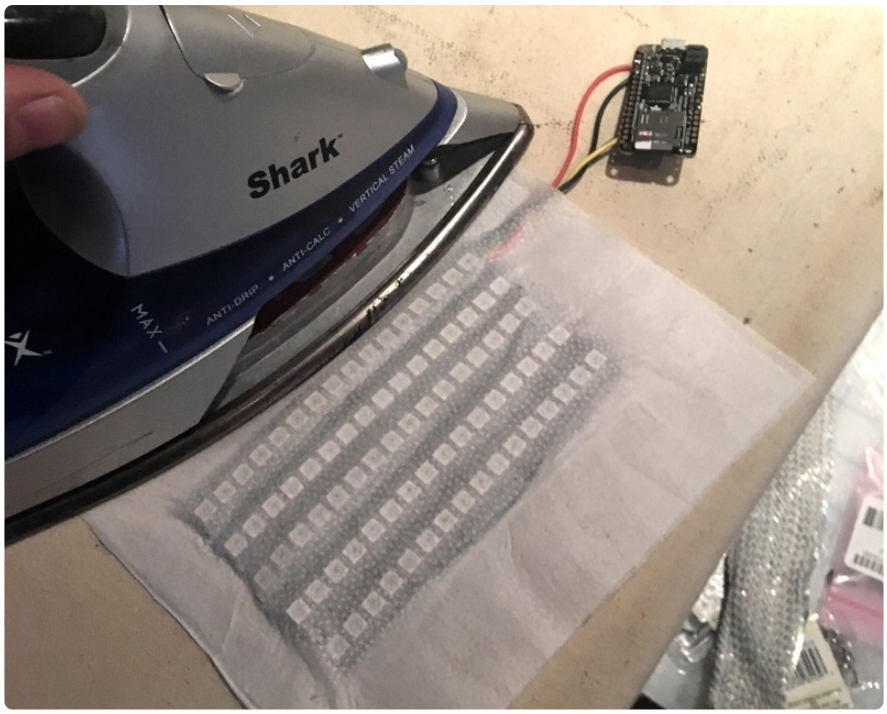
Craft foam is cheap and easy to come by, so don't be afraid to try a few different sizes and shapes until you find one you like.



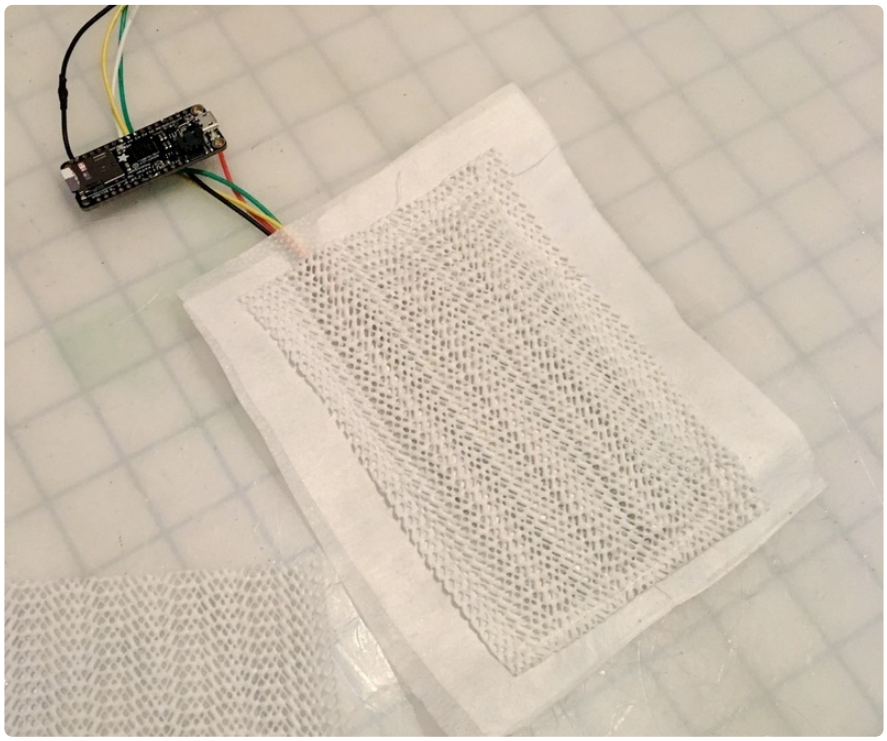
Add Interfacing

Real leather gauntlets probably don't need interfacing, but for faux-leather or craft foam gauntlets it is essential. It will keep the foam or fabric from ripping or stretching when you make holes in it.

Cut the interfacing to the same shape as your pattern and iron it on to the back of your foam and top layer, if you're using one. Cut a piece of lining for the innermost layer, and add interfacing to that too.



Use spray glue or hot glue to gently affix your LED matrix to another piece of interfacing. Add another sheet of interfacing over the top to make a sandwich. Iron around the edges to encase the matrix securely inside with just the lead wires sticking out.



Light Diffusion

Play with different materials until you find something you like. I chose sticky shelf-liner from the hardware store over the top piece of interfacing, with a hand-filigree-cut piece of faux stretch leather layered over the top.

I used my sewing machine to stitch around the matrix through the shelf liner and both layers of interfacing to keep everything tightly together.

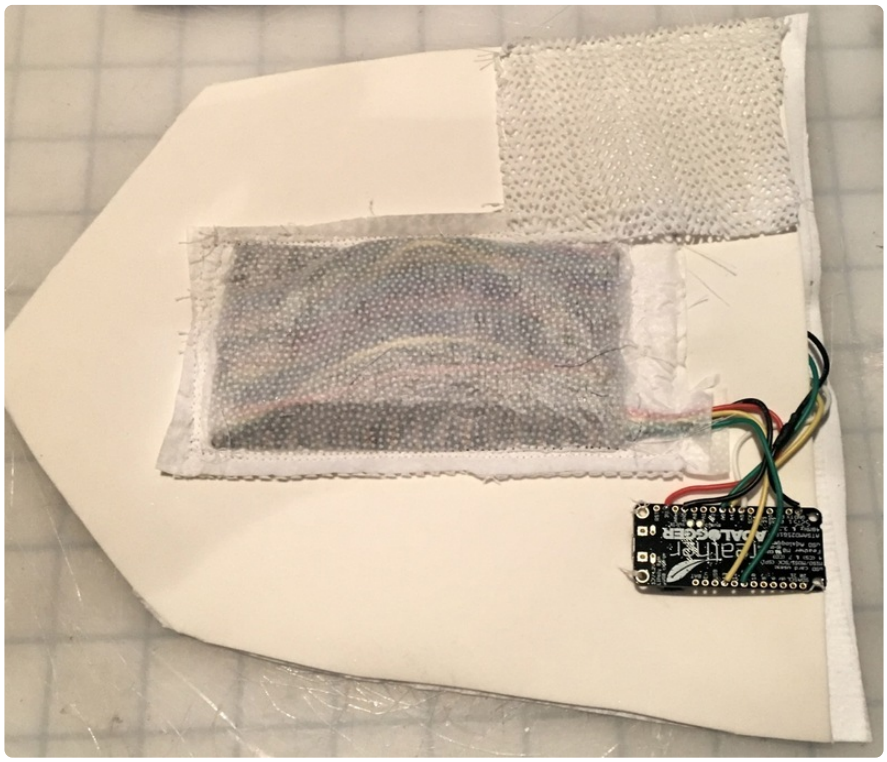
Next I cut a hole in the craft foam gauntlet the right size for my matrix to show through.

I carefully hand-cut a filigree pattern in my top layer of faux leather to create another layer of diffusion. I attached the this layer to the prepped LED matrix with strong spray glue and let it dry completely.

Then I stitched around the edges of the matrix through the leather and craft foam layers, sandwiching it in place securely.

Button Placement

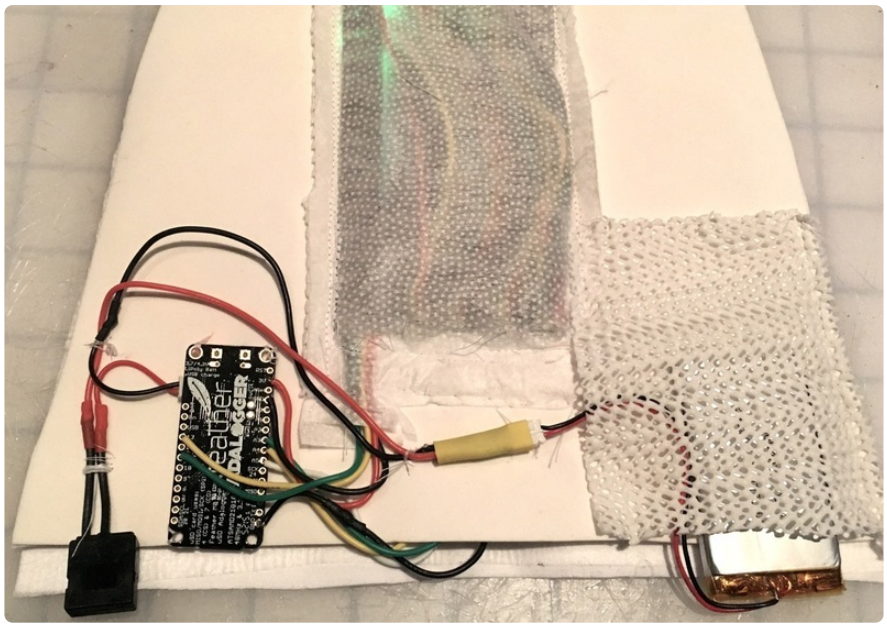
I placed my buttons at the top end of my matrix and punched small holes for the buttons to poke through the faux leather. I glued the buttons to the craft foam underneath, then stitched around the edge by hand. They are delightfully clicky.



Feather & Battery Placement

I placed my feather to the right of my matrix, length-wise so it doesn't distort the shape of the gauntlet when it's on, and beneath the layer of craft foam. I placed the SD card slot toward the top so it's easy to get to if I want to change the images.

I made a battery pocket from more shelf liner on the craft foam to the left of my LED matrix, to balance out any bulkiness.



Finishing

Before you add the bottom layer of lining, secure all your wires and components smoothly and test that everything is working.

Place the lining over the top (wrong side) of your gauntlet and stitch around the bottom and sides, leaving the top open. Turn the lining to the inside and topstitch or serge around the edges.

Set grommets along both sides. Glue on some decorations or sparkles as desired.

Finish the top edge with some more fabric or foam, leaving the inner edge open so you can reach the battery for charging and the SD card / USB port on the feather.

Use It

Adding Videos

Choose video files that have a lot of distinct color and motion. Remember that the script is going to pull the resolution down tremendously, so bright colors and bold shapes (i.e. Disney cartoons like Finding Nemo) will work better than live-action movies or romantic comedies set in Seattle.

Open the **OPCVideoSplitFiles** script that you edited earlier, and set the path and filename for your video. Click "run" and let your video play for as long as you like.

If the video file ends it will automatically loop, so be ready to press the "stop" button as soon as you've recorded enough.

Check to be sure the new files exist in the folders you requested. If they aren't there, make sure the file path is right -- the Processing script won't throw an error if it's not working, so be sure to check before going through a dozen videos.

Change the filename in the Processing script and record another video.

Then, copy all the files onto your respective SD cards.

Buttons

Press Button 1 to go to the beginning of the current video in the playlist. Double-click to go to the previous video in the playlist.

Button 2 is your brightness control. There are 5 levels of brightness available in the code as it's written.

Press button 3 to skip to the next video in the playlist. This button will also wrap around back to the beginning if you're on your last video.

Performance

Left to their own devices, the gauntlets will run through the video playlist in alphabetical order and then loop back to the beginning, ad infinitum.

It's a good idea to use a video editing program to trim off any blank space at the beginning and end of your video files so there aren’t long gaps in the animation.

Strap your gauntlets on. Go out and save the world.