



Glowing WhoVille Hat

Created by Erin St Blaine



<https://learn.adafruit.com/glowing-whoville-hat>

Last updated on 2024-06-03 02:02:42 PM EDT

Table of Contents

Introduction	3
<ul style="list-style-type: none">• Before you Start• Materials• Tools	
Make the Hat	6
<ul style="list-style-type: none">• Cutting• Sewing• Add Pockets• Heating & Setting	
Wiring Diagram	14
Code	15
Electronics Assembly	20
<ul style="list-style-type: none">• Switch Wiring• Neopixel Strip Wiring• Flora Wiring	
Put it All Together	24
<ul style="list-style-type: none">• Place Neopixel Strips• Dangling Ball	

Introduction



Every good elf needs a hat on his shelf.
Or better that said: a hat on his head!

A long curly top with a ball at the end,
It covers his mop and it shows he's a friend

To all other elves (and to people of height).
But more than just that - this hat is alight!

With colors that swirl, shine, shimmer and glow..
A fashion "must have" for those in the know!



Before you Start

Every Who is unique, and that is what makes them so wonderful. What kind of hat does your inner Who wear? Make a [Pinterest board of ideas \(https://adafru.it/sOE\)](https://adafru.it/sOE), and use this pattern as a starting point to design and realize the silliest Who-hat of your Christmas dreams.

This is a great beginner project -- the soldering is fairly simple, and the Flora is really easy to work with. The magic is in your creativity and design. Fosshape is wonderful stuff with nearly infinite possibility*. Let your imagination run wild!

*And, maybe get twice as much Fosshape as you think you'll need. The second hat you make may be more than twice as good as your first attempt. And then you'll have one for a friend!

Materials

- 1-2 meters of [white 30/m Neopixel Strip \(http://adafru.it/1376\)](http://adafru.it/1376)
- [Flora \(http://adafru.it/659\)](http://adafru.it/659)
- [JST 2-pin on/off switch \(http://adafru.it/3064\)](http://adafru.it/3064)
- [Lithium Polymer 1200mAh battery \(http://adafru.it/258\)](http://adafru.it/258) + [charger \(http://adafru.it/1304\)](http://adafru.it/1304) (or a [3xAAA battery pack \(http://adafru.it/727\)](http://adafru.it/727) if Lithium Polymer batteries make you nervous)
- Flexible silicone 26awg wire in 3-4 colors
- [3/4" clear heatshrink tubing \(https://adafru.it/sPe\)](https://adafru.it/sPe)

- 1-2 yards of [Fosshape](https://adafru.it/sPb) (<https://adafru.it/sPb>)
- Clear plastic Christmas Tree ball ornament with at least 1/2 diameter opening (I found this at Michael's)
- Fiberfill stuffing

Tools

- [Clothing steamer](https://adafru.it/sPc) (<https://adafru.it/sPc>) or Heat Gun
- Soldering iron & accessories
- Sewing machine
- [Large Styrofoam Wig Head](https://adafru.it/sPd) (<https://adafru.it/sPd>)



Make the Hat



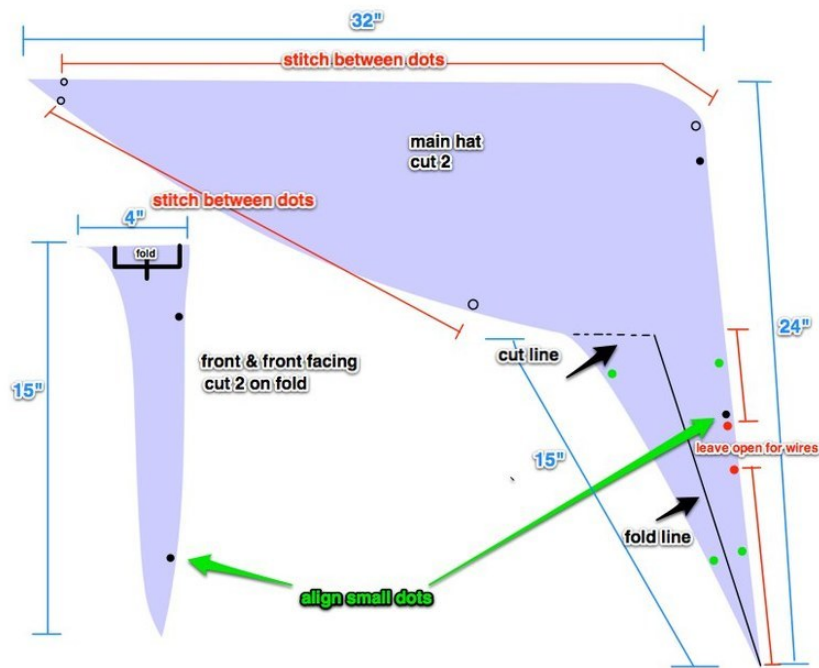
FOSSHAPE is a unique nonwoven fabric, comprised of a low melt synthetic polyester fiber that when exposed to varying degrees of heat (200 degree F range) and pressure it changes in stiffness and texture.

This means you can sculpt whatever shape you desire, hit it with a heat gun or steamer, and it will stay that way. Just Like Magic.

If you've never worked with fosshape before, start by cutting a few small scraps and play around with them. This material can shrink up to 1/3 of its size as it stiffens, so it's a great idea to make something small and get a feel for how much heat is needed, and how much shrinkage wants to happen, before diving into a larger project.

It's always easy to make the Fosshape stiffer with more heat, but if you overheat it there's no going back to soft-and-flexible. So do some experimenting first and save yourself some possible future heartache!

Much more information can be found in the [Wonderflex site's FAQ](https://adafru.it/sOc). (<https://adafru.it/sOc>)



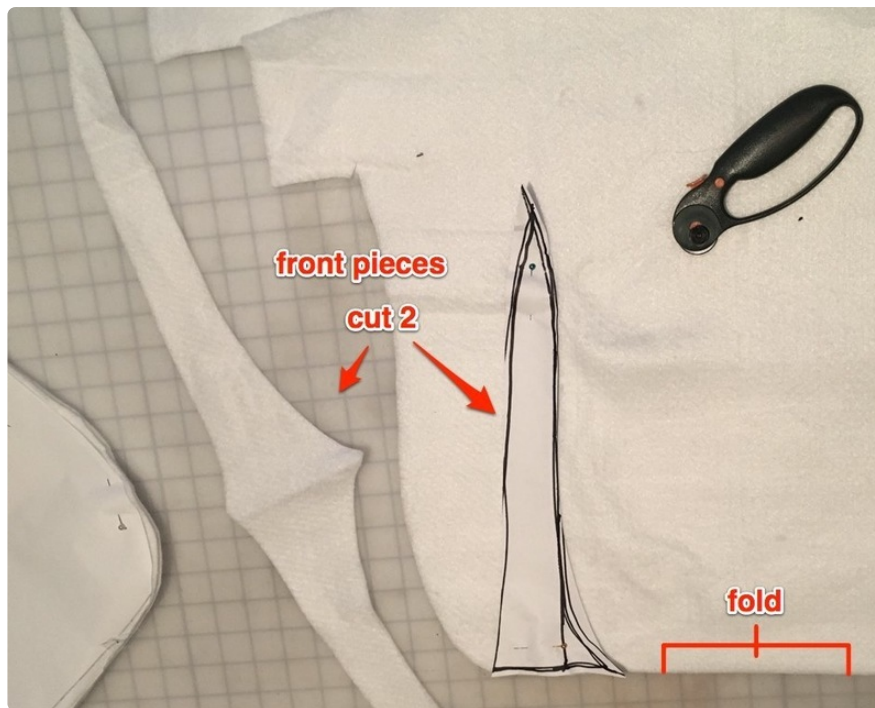
Cutting

Adjust the sizes as desired. This pattern doesn't need to be exact -- different proportions will make your hat your own, so play with the pattern to get the hat you want.

Sketch out the pattern at full size on a piece of butcher paper or wrapping paper and cut it out.

Fold your Fosshape in half so you have two layers and pin your pattern to the Fosshape. Be sure the top edge of the front piece is on the fold, so you'll end up with one double-long piece when it's cut.

Cut out two of the main hat pattern pieces, and two of the front pieces on the fold.



Sewing



For the HAT pieces, stitch the top and bottom edge together between the circles.



For the FRONT piece, stitch the pointy edges together. Trim all seams close to stitching and grade out extra fabric at the point.

Turn both pieces right-side out.



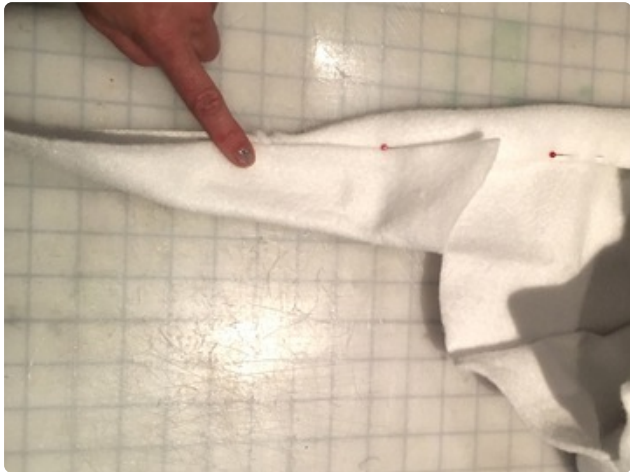


Pin one of the rough edges of the FRONT piece to the front edge of the HAT piece, matching centers. Stitch.

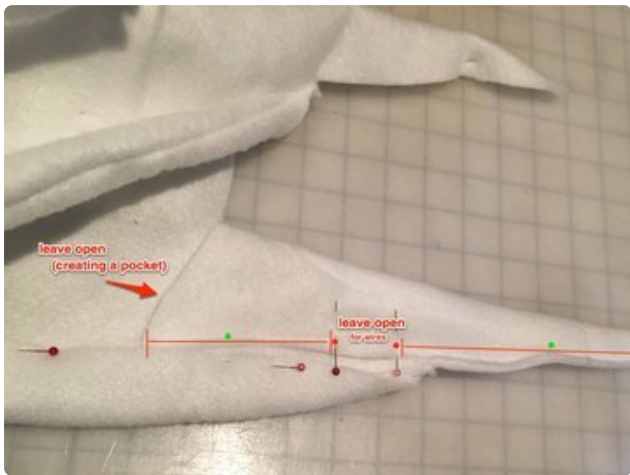


Fold the facing piece over the seam and pin it in place. **Don't stitch this yet!** We're leaving it open so we can get the lights inside.

Add Pockets



At the front of the hat, cut along the cut lines and fold along the fold lines.



Stitch as shown, leaving 1.5" opening for wires (be sure the Flora fits through) and leaving the top edge open as a pocket.

Heating & Setting



Place your hat on your wig head and pin it down. (Don't worry, it won't hurt her)



Begin to shape your hat by making pleats along the top seam. Play with the number and depth of the pleats until you get the shape you want.



If needed, temporarily stuff some fiberfill or extra fabric inside to help the hat hold its shape during heating.



Once you have your desired shape, pin down any side pleats or areas that might shift.



With a clothing steamer or heat gun, heat the fabric. Only heat the areas you want to get stiff. Stay away from the dangly wings in front. **Go easy here** -- you can always make it stiffer but once you lose the softness it's gone forever.

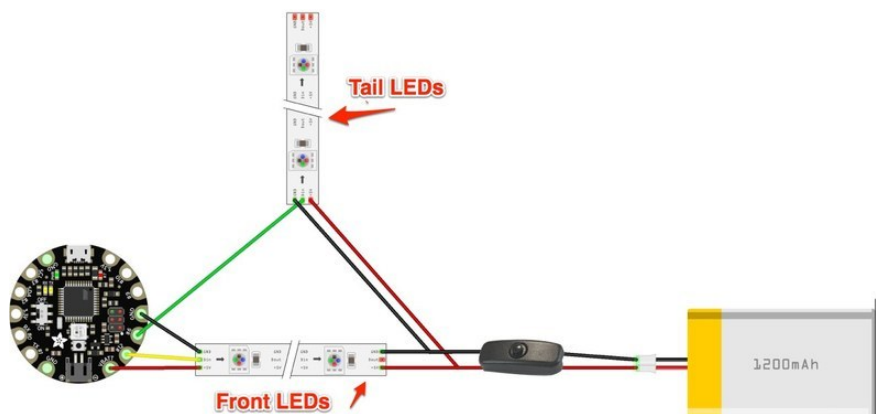
Remove all the pins except those holding the front facing in place. Try your hat on and bounce around a bit, and make any needed adjustments.

Once you're happy with the fit, get out your measuring tape and measure the front edge and the tail from the crown of your head to the tip. This measurement will tell you exactly how many LEDs to use in each spot.

Add a few more inches to the tail measurement to allow for 4-5 LEDs poking through into the dangling ball.



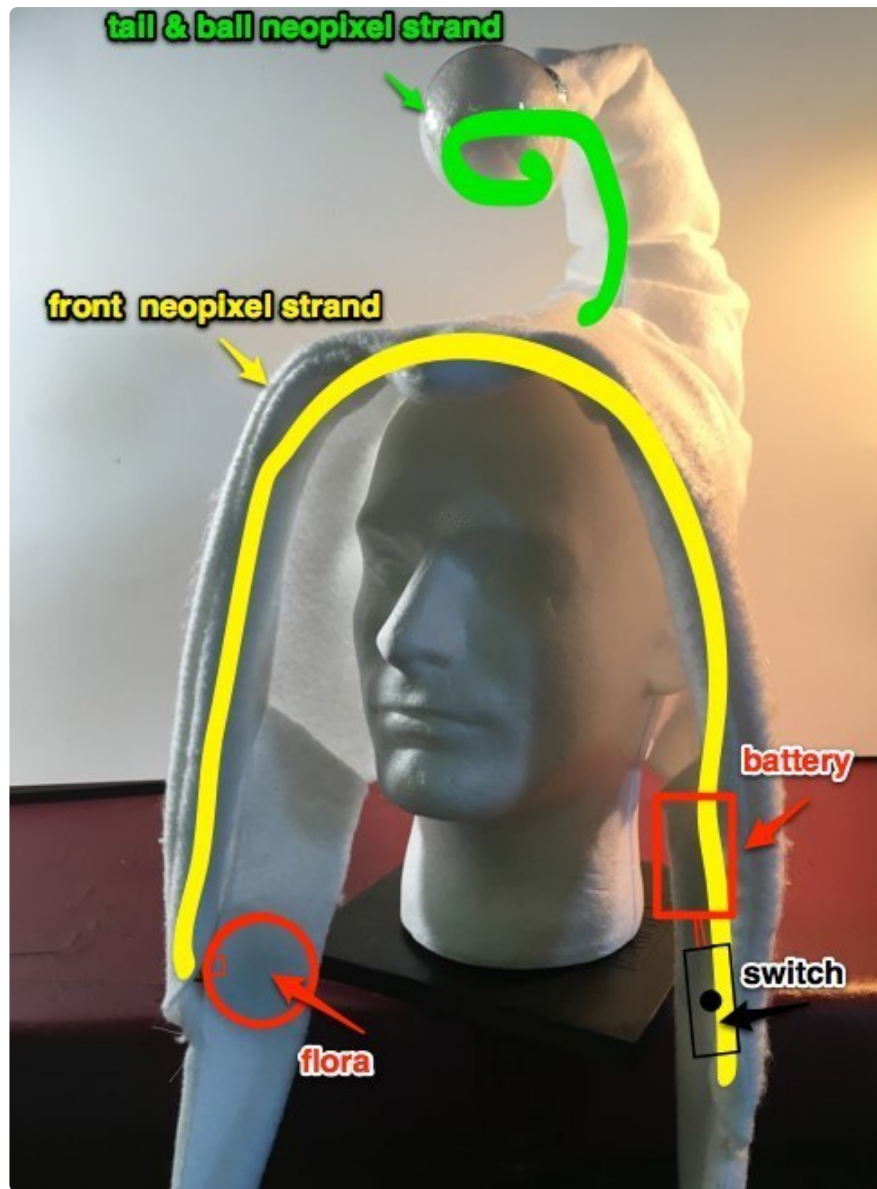
Wiring Diagram



We'll be using two separate strands of neopixels: one in the front and one in the tail and ball.

The two pockets you made in the front of the hat will hold the flora (on the left) and the battery and power switch (on the right).

The battery & switch will connect to the OUT end of the front strand and the IN end of the tail strand, to minimize long wire runs. (This is a possibility folks sometimes overlook with NeoPixels: while data **MUST** travel in a specific direction, "in" to "out," power doesn't care and can be applied at either end.)



Code

If this is your first time using Flora, check out the [Flora setup guide](https://adafruit.it/dgt) (<https://adafruit.it/dgt>) to get up and running.

You'll need to install the FastLED library to get this code working.

There is a [great tutorial on how to install Arduino libraries](https://adafruit.it/aYM) (<https://adafruit.it/aYM>) in case you haven't done this before. It's even simpler if using **Arduino version 1.6.5** or newer...from the "Sketch" menu, select "Include Library→Manage Libraries..." and use the install option for FastLED. Done!

Then copy and paste the code below into your [Arduino IDE](https://adafruit.it/jDQ). (<https://adafruit.it/jDQ>)

Update the first few lines in the code with the number of LEDs you actually have in your hat.

Plug your Flora in with a USB cable. Select Flora from your list of boards under the Tools menu, and choose your USB port under the Ports menu. Press upload.

```
#include <FastLED.h>

#define LED_PIN      12
#define TAIL_PIN     6
#define NUM_LEDS     20      //Number of LEDs in the front
#define NUM_TAIL     15      //Number of LEDs in the tail
#define NUM_BALL     4       //Number of LEDs in the ball

#define BRIGHTNESS   255
#define LED_TYPE      WS2812B
#define COLOR_ORDER   GRB
CRGB leds[NUM_LEDS];
CRGB tail[NUM_TAIL];

#define UPDATES_PER_SECOND 100

// This example shows how to cross-fade between different color palettes
// using the function nblendPaletteTowardPalette.
//
// The basic idea is that you always have a "current palette" that you're
// pulling colors from with ColorFromPalette, and you have a "target palette"
// which is the 'next' palette that you want to get to.
//
// After that, implementation is relatively simple: just periodically call
// nblendPaletteTowardPalette( currentPalette, targetPalette);
// If the current palette is not yet equal to the target palette, this
// function will make a few small changes to the current palette to make
// it slightly more like the target. Over time, the current palette will
// come to be equal to the target.
// There's no need to test the current and target for equality; it's safe
// to keep calling nblendPaletteTowardPalette even after current reaches target.
// For faster blending, call nblendPaletteTowardPalette twice per loop.

void setup() {
  delay( 3000 ); // power-up safety delay
  FastLED.addLeds<LED_TYPE, LED_PIN, COLOR_ORDER>(leds,
NUM_LEDS).setCorrection( TypicalLEDStrip );
  FastLED.addLeds<LED_TYPE, TAIL_PIN, COLOR_ORDER>(tail,
NUM_TAIL).setCorrection( TypicalLEDStrip );
  FastLED.setBrightness( BRIGHTNESS );
}

//COLOR PALETTES: Here are several different ways to set up custom color
palettes. More info can be found at http://fastled.io/docs/3.1/group\_\_colorpalettes.html
// Several palettes come pre-defined with FastLED: RainbowColors_p, ForestColors_p,
OceanColors_p, HeatColors_p, LavaColors_p, RainbowStripeColors_p
// so feel free to use those without needing to define them here.
// Much more info and a list of color definitions can be found at https://github.com/FastLED/FastLED/wiki/Pixel-reference#chsv
```

```

// A red and white striped palette -- RedWhite_p
// "CRGB::Gray" is used as white to keep the brightness more uniform.
const TProgmemRGBPalette16 RedOrange_p FL_PROGMEM =
{ CRGB::Red, CRGB::Red, CRGB::Red, CRGB::Red,
  CRGB::Red, CRGB::Red, CRGB::Red, CRGB::Orange,
  CRGB::Red, CRGB::Red, CRGB::Red, CRGB::Red,
  CRGB::Red, CRGB::Red, CRGB::Red, CRGB::Orange };

// A mostly green and dark green palette. Greens_p
const TProgmemRGBPalette16 Greens_p FL_PROGMEM =
{ CRGB::Green, CRGB::Green, CRGB::DarkGreen, CRGB::DarkGreen,
  CRGB::Green, CRGB::Green, CRGB::DarkGreen, CRGB::DarkGreen,
  CRGB::Green, CRGB::Green, CRGB::DarkGreen, CRGB::DarkGreen,
  CRGB::Green, CRGB::Green, CRGB::DarkGreen, CRGB::DarkGreen, };

// A pure "fairy light" palette with some brightness variations FairyLight_p
#define HALFFAIRY ((CRGB::FairyLight & 0xFEFEFE) / 2)
#define QUARTERFAIRY ((CRGB::FairyLight & 0xFCFCFC) / 4)
const TProgmemRGBPalette16 FairyLight_p FL_PROGMEM =
{ CRGB::FairyLight, CRGB::FairyLight, CRGB::FairyLight, CRGB::FairyLight,
  HALFFAIRY, HALFFAIRY, CRGB::FairyLight, CRGB::FairyLight,
  QUARTERFAIRY, QUARTERFAIRY, CRGB::FairyLight, CRGB::FairyLight,
  CRGB::FairyLight, CRGB::FairyLight, CRGB::FairyLight, CRGB::FairyLight };

// A palette of soft snowflakes with the occasional bright one
const TProgmemRGBPalette16 Snow_p FL_PROGMEM =
{ 0x304048, 0x304048, 0x304048, 0x304048,
  0x304048, 0x304048, 0x304048, 0x304048,
  0x304048, 0x304048, 0x304048, 0x304048,
  0x304048, 0x304048, 0x304048, 0xE0F0FF };

// A black "off" palette Black_p;
const TProgmemRGBPalette16 Black_p FL_PROGMEM =
{ CRGB::Black };

// little twinkles
const TProgmemRGBPalette16 Twinkles_p FL_PROGMEM =
{ CRGB::Black, CRGB::Black, CRGB::Black, CRGB::Black,
  CRGB::Black, CRGB::Black, CRGB::Black, CRGB::Black,
  CRGB::Gray, CRGB::Black, CRGB::Black, CRGB::Black,
  CRGB::White, CRGB::Black, CRGB::Black, CRGB::Black };

// purple blue
const TProgmemRGBPalette16 Purples_p FL_PROGMEM =
{ CRGB::Purple, CRGB::Magenta, CRGB::Blue, CRGB::DarkBlue,
  CRGB::Purple, CRGB::Magenta, CRGB::Blue, CRGB::DarkBlue,
  CRGB::Purple, CRGB::Magenta, CRGB::Blue, CRGB::DarkBlue,
  CRGB::Purple, CRGB::Magenta, CRGB::Blue, CRGB::DarkBlue };

//define starting palettes here

CRGBPalette16 currentPalette( CRGB::Black);
CRGBPalette16 tailPalette(CRGB::Black);

CRGBPalette16 targetPalette( RedOrange_p );
CRGBPalette16 tailtargetPalette( Greens_p );

void loop()
{ ball();
  ChangePalettePeriodically();

  // Crossfade current palette slowly toward the target palette
  //
  // Each time that nblendPaletteTowardPalette is called, small changes
  // are made to currentPalette to bring it closer to matching targetPalette.
  // You can control how many changes are made in each call:

```

```

// - the default of 24 is a good balance
// - meaningful values are 1-48. 1=veeeeeeeery slow, 48=quickest
// - "0" means do not change the currentPalette at all; freeze

uint8_t maxChanges = 24;
nblendPaletteTowardPalette( currentPalette, targetPalette, maxChanges);
nblendPaletteTowardPalette( tailPalette, tailtargetPalette, maxChanges);

static uint8_t startIndex = 0;
startIndex = startIndex + 1; /* motion speed */
FillLEDsFromPaletteColors( startIndex);

FastLED.show();
FastLED.delay(1000 / UPDATES_PER_SECOND);
}

void FillLEDsFromPaletteColors( uint8_t colorIndex)
{
    uint8_t brightness = 120;          //You can change the brightness of the front
    and the tail independently.
    uint8_t brightnessTail = 255;      // On my hat, having the tail brighter makes
    the hat look more balanced.

    for( int i = 0; i < NUM_LEDS; i++) {
        leds[i] = ColorFromPalette( currentPalette, colorIndex + sin8(i*16),
        brightness);
        colorIndex += 3;
    }

    for( int t = 0; t < (NUM_TAIL-NUM_BALL); t++) {
        tail[t] = ColorFromPalette( tailPalette, colorIndex + sin8(t*16),
        brightnessTail);
        colorIndex += 3;
    }
}

//Set up your playlist here

void ChangePalettePeriodically()
{
    uint8_t secondHand = (millis() / 1000) % 225;
    static uint8_t lastSecond = 250;

    if( lastSecond != secondHand) {
        lastSecond = secondHand;

        if( secondHand == 5) { targetPalette = RedOrange_p; tailtargetPalette =
        Greens_p; } // red and green Elf
        if( secondHand == 23) { targetPalette = Black_p; tailtargetPalette =
        Black_p; } //fade through black
        if( secondHand == 25) { targetPalette = RainbowColors_p; tailtargetPalette =
        RainbowColors_p; } //Rainbow
        if( secondHand == 50) { targetPalette = CloudColors_p; tailtargetPalette =
        CloudColors_p; } //Clouds
        if( secondHand == 75) { targetPalette = Twinkles_p; tailtargetPalette =
        FairyLight_p; } //Fairy lights & Twinkles
        if( secondHand == 100) { targetPalette = ForestColors_p; tailtargetPalette =
        OceanColors_p; } // Green and blue
        if( secondHand == 125) { targetPalette = FairyLight_p; tailtargetPalette =
        Greens_p; } // White and red "Santa"
        if( secondHand == 147) { targetPalette = Black_p; tailtargetPalette =
        Black_p; } //fade through black
        if( secondHand == 150) { targetPalette = Snow_p; tailtargetPalette =
        RedOrange_p; } //Green and white elf
        if( secondHand == 175) { targetPalette = Purples_p; tailtargetPalette =
        Purples_p; } //Purple palette
        if( secondHand == 200) { targetPalette = Twinkles_p; tailtargetPalette =
        Twinkles_p; } //twinkles
    }
}

```



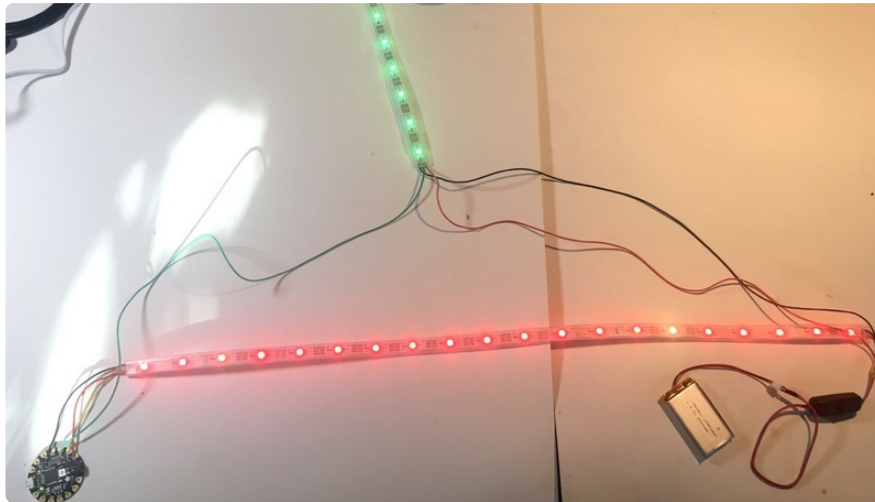
```
}  
  
void ball(){  
  for (uint8_t b = (NUM_TAIL - NUM_BALL); b < NUM_TAIL; b++)  
    tail[b] = CHSV(160, 0, 200);  
  
  FastLED.show();  
}
```

This code is fairly easy to modify. You can create your own color palettes and set up your own "playlist" of animations that will change over time.

The way it's written, the code will cycle through several different modes over 225 seconds. Play with the different palettes to come up with your own modes and your own timing.



Electronics Assembly

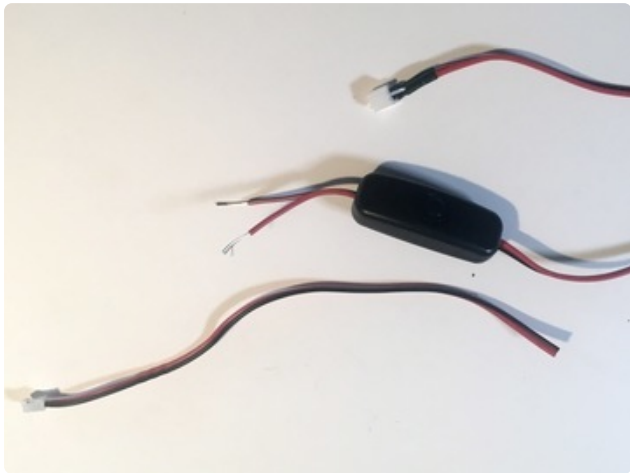


We're going to make a sort of triangle shape with your two strips, with the Flora on the left side, and the switch and battery on the right side. The FRONT strip will be wired close to the other components, and the TAIL strip will have a couple feet of lead wire so it reaches easily to its place in your final project.

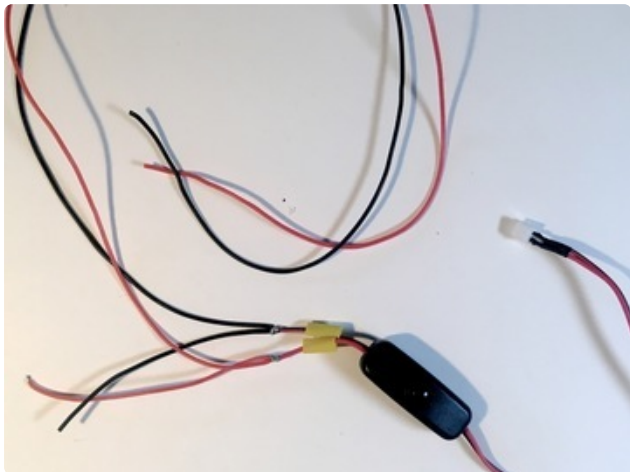
Switch Wiring



Carefully cut your Neopixel strips along the cut lines at the length needed for your hat.

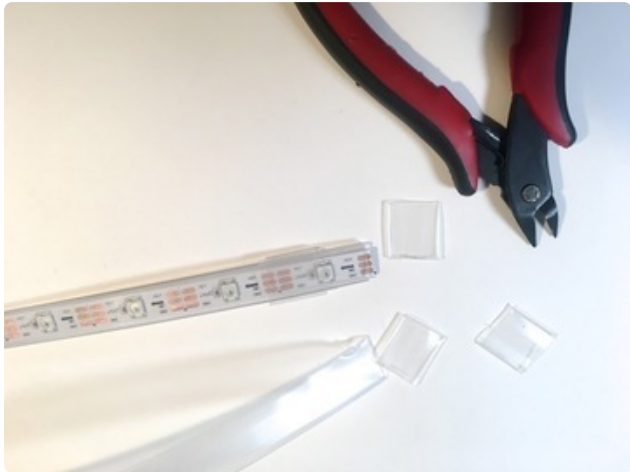


Cut the **male** lead wires (not the ones that plug into the battery, the other ones) off your switch and strip 1/4" of each wire.

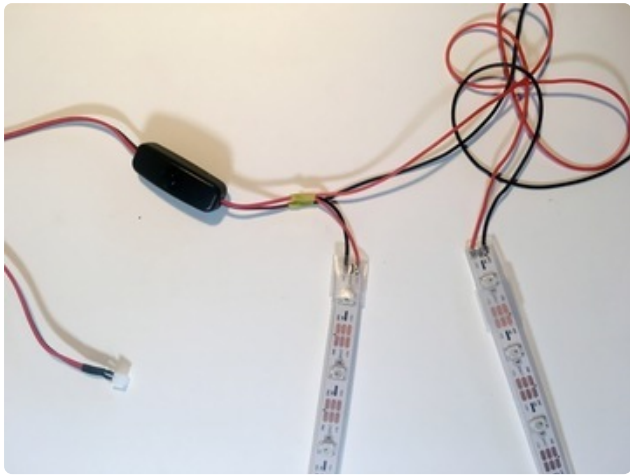


Cut a 3" piece of red wire and black wire, and a 2-3 foot piece of red wire and black wire. Splice both the long and short red wire onto the red wire coming from your switch. Repeat with the black wires.

Neopixel Strip Wiring

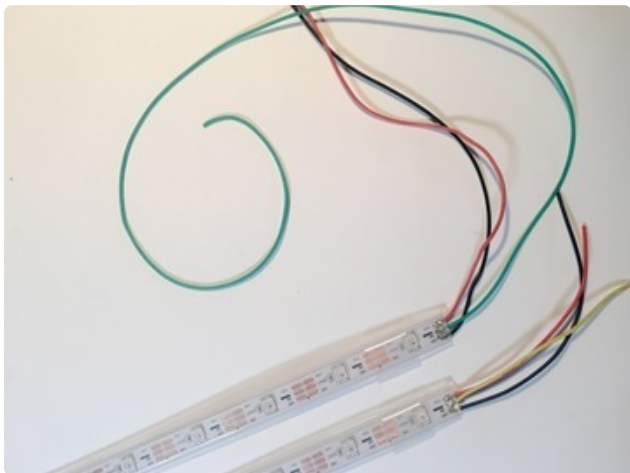


Cut a 3/4" piece of heat shrink for each end of each strip -- so, 4 pieces total. Slip them over the silicone sheath and up out of the way for now.



Solder the 3" wires coming from the switch to the **out** end of the **FRONT** strip. (Look for the arrows to indicate data direction and solder to the end the arrows are pointing **towards**.)

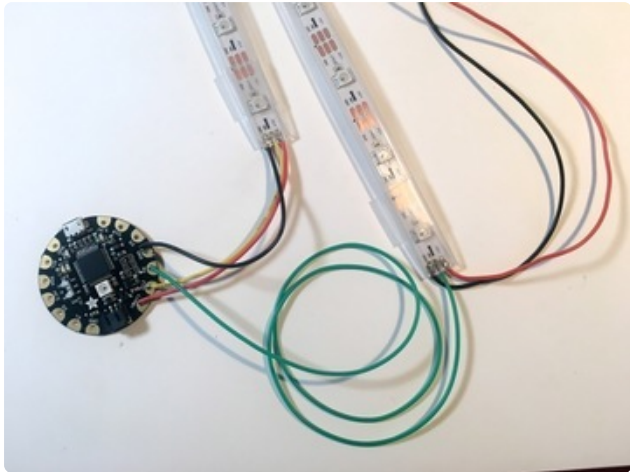
Solder the long wires to the **in** end of the **TAIL** strip (arrows pointing **away**.)



Cut another 3" red and black wire and also a 3" yellow wire. Cut a 2 foot green wire.

Solder all three 3" wires to the **in** end of the **FRONT** strip (the end with no wires yet), and the long green wire to the **in** end of the **TAIL** strip (where the other long wires are already soldered).

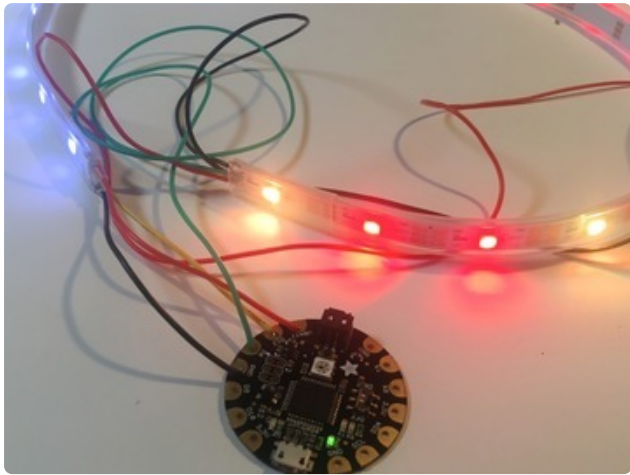
Flora Wiring



Solder the other end of the long green wire to Flora's pin 6.

Solder the 3" red wire to VBAT, the black wire to G and the yellow wire to pin 12.

Plug your battery in to the JST connector on your switch and test to be sure all the lights come on.



The power switch on the Flora board can be left in the “ON” position. The toggle button from the battery is then used to switch the circuit on or off. If your hat gets bumped and isn't responding, check that the switch is “on.”



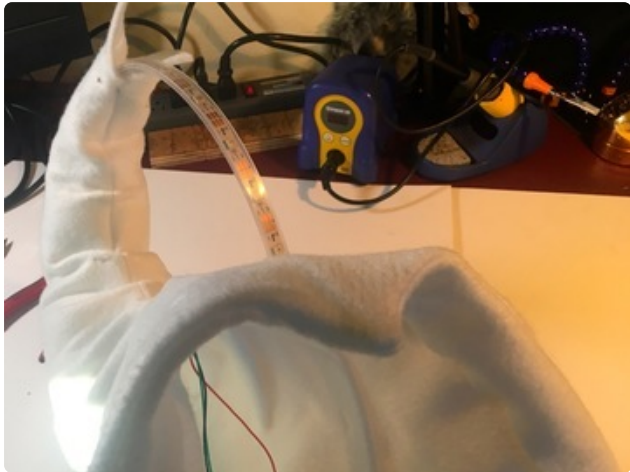
Once your sure the whole circuit is **working**, slip the heat shrink down over the ends of both strips. Fill the end of the silicone sheath with hot glue, then quickly slide the heat shrink over and shrink in place. This will keep your strips water tight and fairly break-resistant.



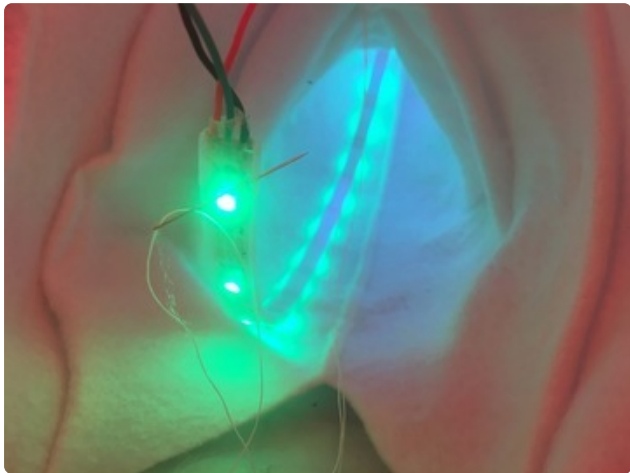
Put it All Together



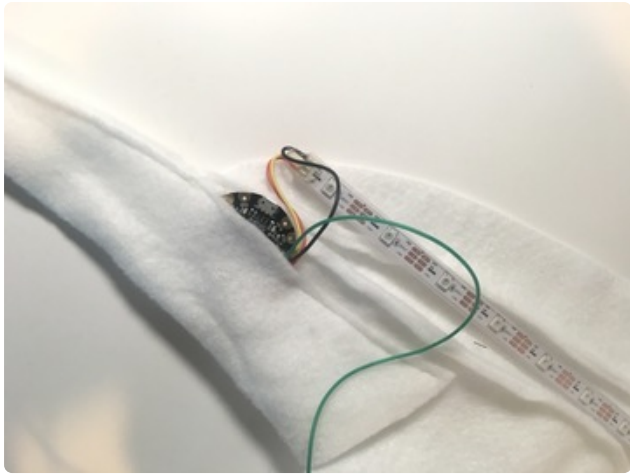
Place Neopixel Strips



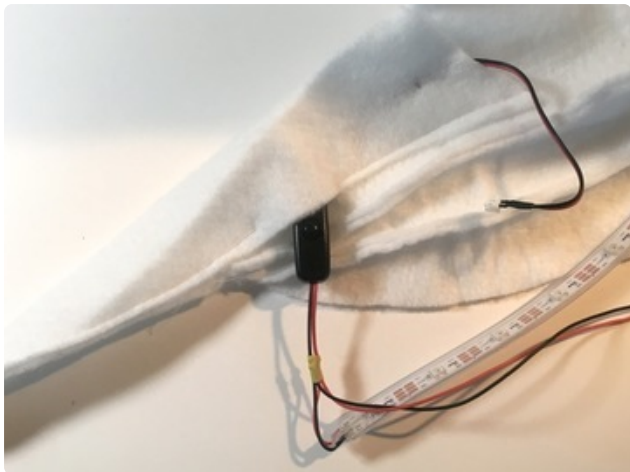
Slide the TAIL strip inside the tail, leaving 4-5 LEDs poking out -- these will go inside the ball later on.



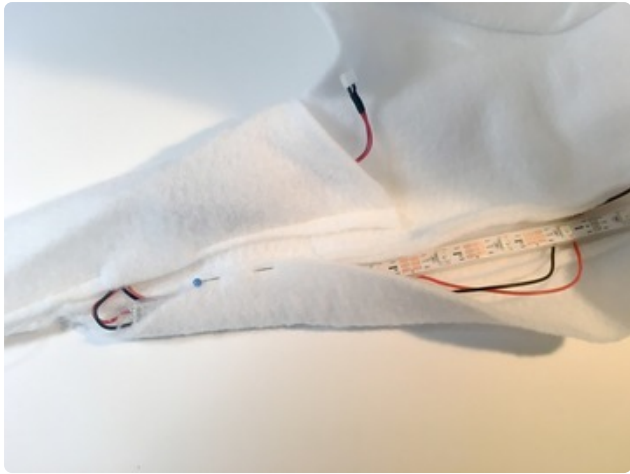
With a needle and thread, stitch carefully through the silicone sheath and tack the strip securely to the top of the hat, with the lights facing inwards.



Slip the Flora in through the opening you left in the pocket on the left front of the hat, with the USB port facing outwards so you can reach it easily for future code updates.

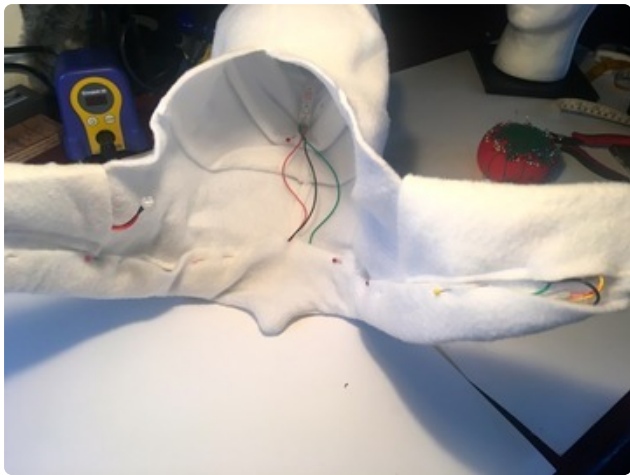


Slip the switch through the same opening on the other side and feed the battery wire up through the top of the pocket.

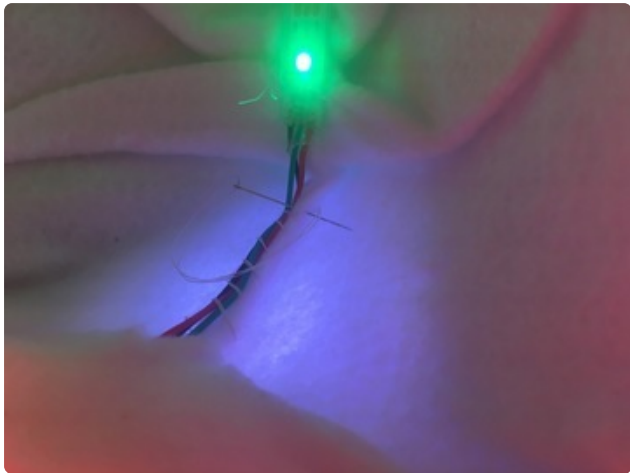


Lay the LED strip inside the front of the hat with the wires tucked behind it, and pin the front facing closed over the LEDs.

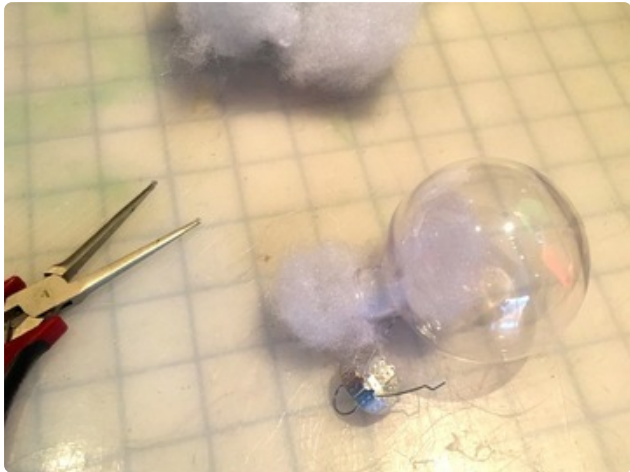
Carefully stitch the facing closed over the LED strip and wires, leaving an opening at the crown for the tail wires to get out.



Sew the tail wires neatly to the inside of the crown of the hat so they don't get pulled while you're wearing it.

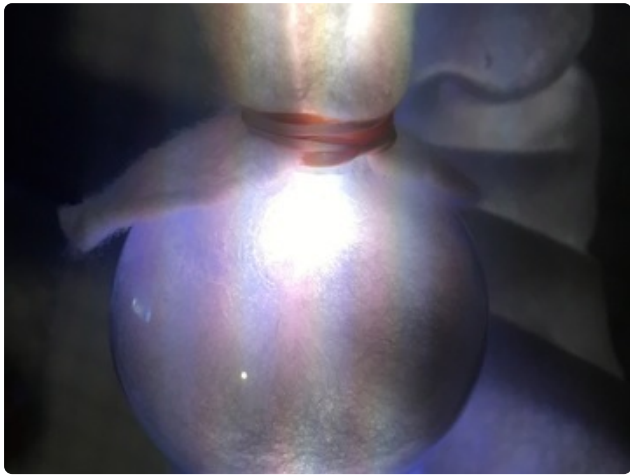


Dangling Ball



Pull the top off the ball ornament and stuff some fiberfill or other diffusion material inside.

Carefully fold your LED strip with the lights facing outward, and slip it inside the ball. Add more fiberfill until the lights are nicely placed and diffused.



With a rubber band, firmly attach the ball to the end of the tail. Cover it up with ribbon or holographic tape to make it look finished.

Note: this part is a little harder than it looks, so take your time and don't be discouraged! I found a pair of needlenose pliers and a chopstick to be extremely helpful.



Plug your battery into your switch lead and then tuck it down inside the pocket. Turn your hat on and off by squeezing the switch right through the hat.

Go light up the town!