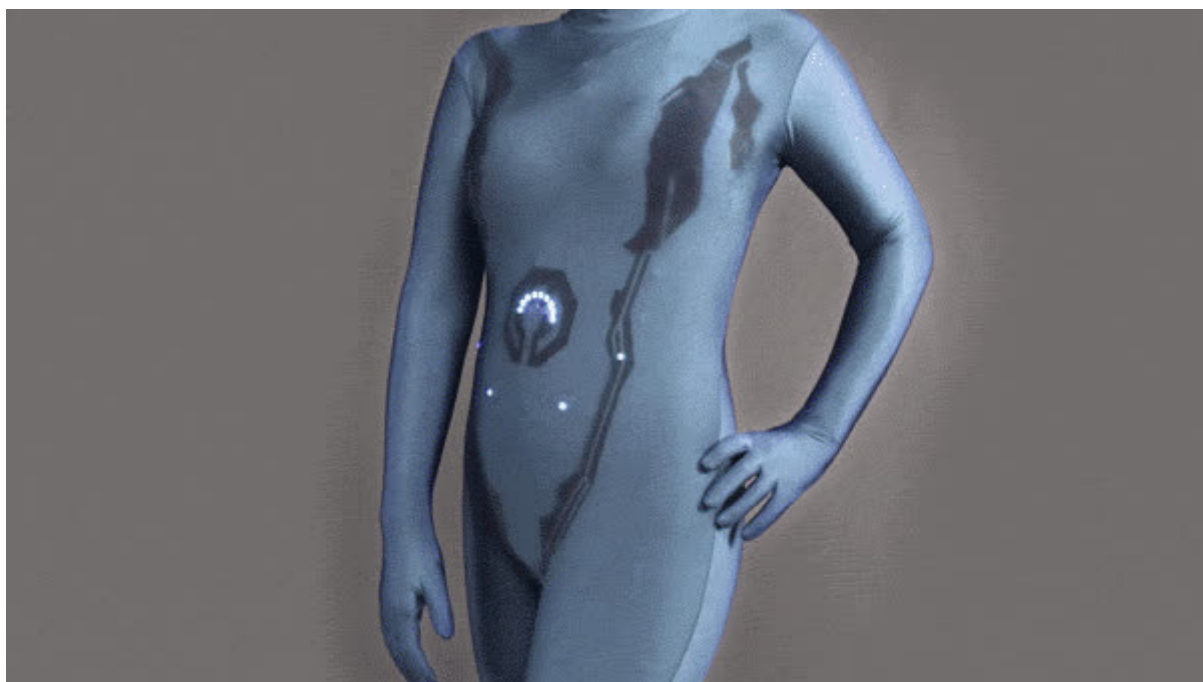




NeoPixel LED Cortana Costume

Created by Ruiz Brothers



<https://learn.adafruit.com/neopixel-led-cortana-costume>

Last updated on 2024-06-03 01:35:01 PM EDT

Table of Contents

Overview	3
Circuit Diagram	5
Code	6
Assembly	8
3D Printing	14
<ul style="list-style-type: none">• Trace Reference• Measure body length• Extrude Paths• Print Stencil• Mark LEDs on unitard• Paint Stencil	

Overview

In this project we're adding sewable NeoPixels and GEMMA to a bodysuit to make a light up Cortana costume.

To get extra details, we'll trace artwork from reference photos, format the artwork and make a stencil that we can 3d print.

How it Works

This sewable project uses a Gemma, NeoPixel Rings and sewable NeoPixels use our conductive thread to connect data, power and ground.

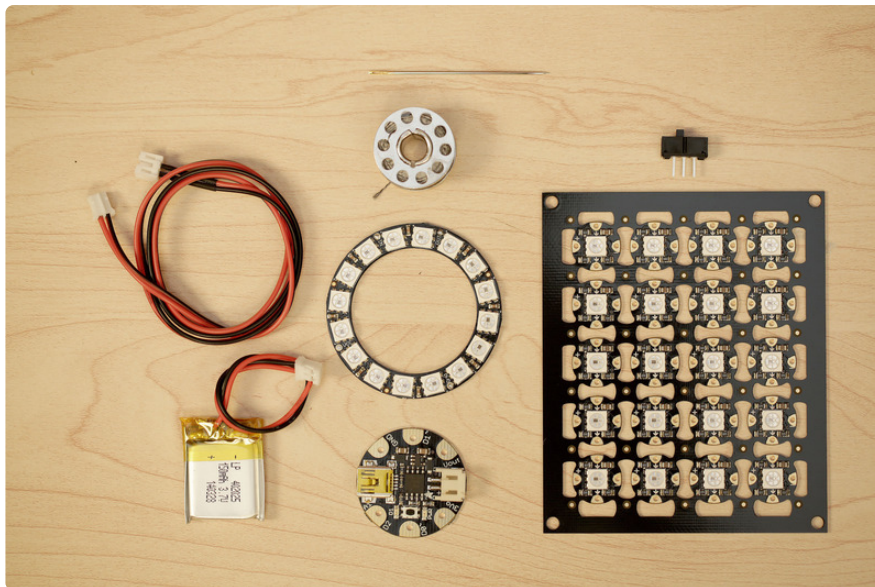


Project Advisory

Stitching on stretchy fabric can be very challenging. Be sure to stretch the fabric while you sew so the thread doesn't end up too tight, or the strong stainless steel thread might tear the fabric. If you're a beginner, stick to soldering or try an easier conductive thread project first!

Prerequisite Guides

- [Introducing GEMMA \(https://adafru.it/duA\)](https://adafru.it/duA)
- [NeoPixel Uberguide \(https://adafru.it/dhw\)](https://adafru.it/dhw)
- [Conductive Thread \(https://adafru.it/dn3\)](https://adafru.it/dn3)



Parts & Components

Most of the tools parts and supplies are available in our shop.

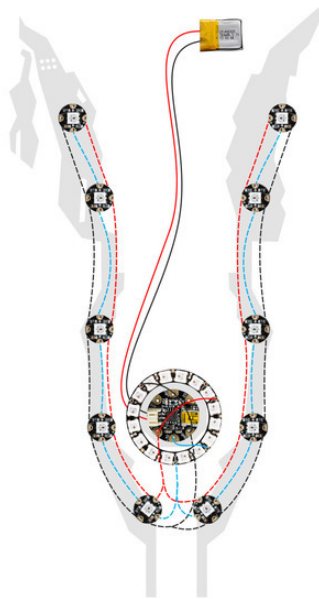
- [Gemma](http://adafru.it/1222) (<http://adafru.it/1222>)
- [NeoPixels](http://adafru.it/1559) (<http://adafru.it/1559>)
- [NeoPixel Ring 16x](http://adafru.it/1463) (<http://adafru.it/1463>)
- [150mAh Lipo Battery](http://adafru.it/1317) (<http://adafru.it/1317>)

Tools & Supplies

- [Conductive Thread](https://adafru.it/dYZ) (<https://adafru.it/dYZ>)
- [Needles](https://adafru.it/dZ0) (<https://adafru.it/dZ0>)
- [3D Printer](https://adafru.it/dTm) (<https://adafru.it/dTm>)
- [PLA Filament](https://adafru.it/dTn) (<https://adafru.it/dTn>)
- [Wire Stripper](http://adafru.it/527) (<http://adafru.it/527>)
- [Soldering Iron](https://adafru.it/dTo) (<https://adafru.it/dTo>)
- [Solder](https://adafru.it/dTp) (<https://adafru.it/dTp>)



Circuit Diagram



Connect the NeoPixel Ring to Gemma using Silicone Wire.

Vcc to **Vout**

IN to **D1**

GND to **GND**

Connect each NeoPixel into two chain along the sides. Create a Y connection where they meet in the middle.

Share the NeoPixel Ring's **Out** pin to the digital pins on both NeoPixel chains as shown in the diagram.

Connect - and + pins on the NeoPixels to **GND** and **Vout** on Gemma.

Code

Load the following code onto your GEMMA using the Adafruit Arduino IDE or Codebender:

```
//Cortana costume animating NeoPixels
//based on the Larson Scanner Shades by Phillip Burgess
//https://learn.adafruit.com/larson-scanner-shades
//modified by Becky Stern for Adafruit
#include <Adafruit_NeoPixel.h>

#define PIN 1

// Parameter 1 = number of pixels in strip
// Parameter 2 = Arduino pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(21, PIN, NEO_GRB + NEO_KHZ800);

int ringRightSide[] = {0, 1, 2, 3, 4, 5, 6, 7, 8};
int ringLeftSide[] = {0, 15, 14, 13, 12, 11, 10, 9, 8};
int singlePixels[] = {16, 17, 18, 19, 20};

// IMPORTANT: To reduce NeoPixel burnout risk, add 1000 uF capacitor across
// pixel power leads, add 300 - 500 Ohm resistor on first pixel's data input
// and minimize distance between Arduino and first pixel.  Avoid connecting
// on a live circuit...if you must, connect GND first.

void setup() {
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'
}

int pos = 0;
int pos2 = 0;

void loop() {
  // Draw 5 pixels centered on pos.  setPixelColor() will clip any
  // pixels off the ends of the strip, we don't need to watch for that.
  strip.setPixelColor(ringRightSide[abs((pos - 2)%9)], strip.Color(0, 0, 80)); //
Dark blue
  strip.setPixelColor(ringRightSide[abs((pos - 1)%9)], strip.Color(0, 0, 150)); //
Medium blue
  strip.setPixelColor(ringRightSide[abs((pos    )%9)], strip.Color(15, 15,
200)); // Center pixel is brightest
  strip.setPixelColor(ringRightSide[abs((pos + 1)%9)], strip.Color(0, 0, 150)); //
Medium blue
  strip.setPixelColor(ringRightSide[abs((pos + 2)%9)], strip.Color(0, 0, 80)); //
Dark blue

  strip.setPixelColor(ringLeftSide[abs((pos - 2)%9)], strip.Color(0, 0, 80)); //
Dark blue
  strip.setPixelColor(ringLeftSide[abs((pos - 1)%9)], strip.Color(0, 0, 150)); //
Medium blue
  strip.setPixelColor(ringLeftSide[abs((pos    )%9)], strip.Color(15, 15, 200)); //
Center pixel is brightest
  strip.setPixelColor(ringLeftSide[abs((pos + 1)%9)], strip.Color(0, 0, 150)); //
Medium blue
  strip.setPixelColor(ringLeftSide[abs((pos + 2)%9)], strip.Color(0, 0, 80)); //
Dark blue
```

```

    strip.setPixelColor(singlePixels[abs(pos2 - 1)], strip.Color(0, 0, 80)); // Dark
blue
    strip.setPixelColor(singlePixels[abs(pos2    )], strip.Color(15, 15, 200)); //
Center pixel is brightest
    strip.setPixelColor(singlePixels[abs(pos2 + 1)], strip.Color(0, 0, 80)); // Dark
blue

    strip.show();
    delay(100);

    // Rather than being sneaky and erasing just the tail pixel,
    // it's easier to erase it all and draw a new one next time.
    for(int j=-2; j<= 2; j++) {
        strip.setPixelColor(ringRightSide[(pos+j)%9], 0);
        strip.setPixelColor(ringLeftSide[(pos+j)%9], 0);
        strip.setPixelColor(singlePixels[(pos2+j)%5], 0);
    }

    pos += 1;
    if(pos < 0) {
        pos = 1;

    } else if(pos >= 9) {
        pos = 0;

    }

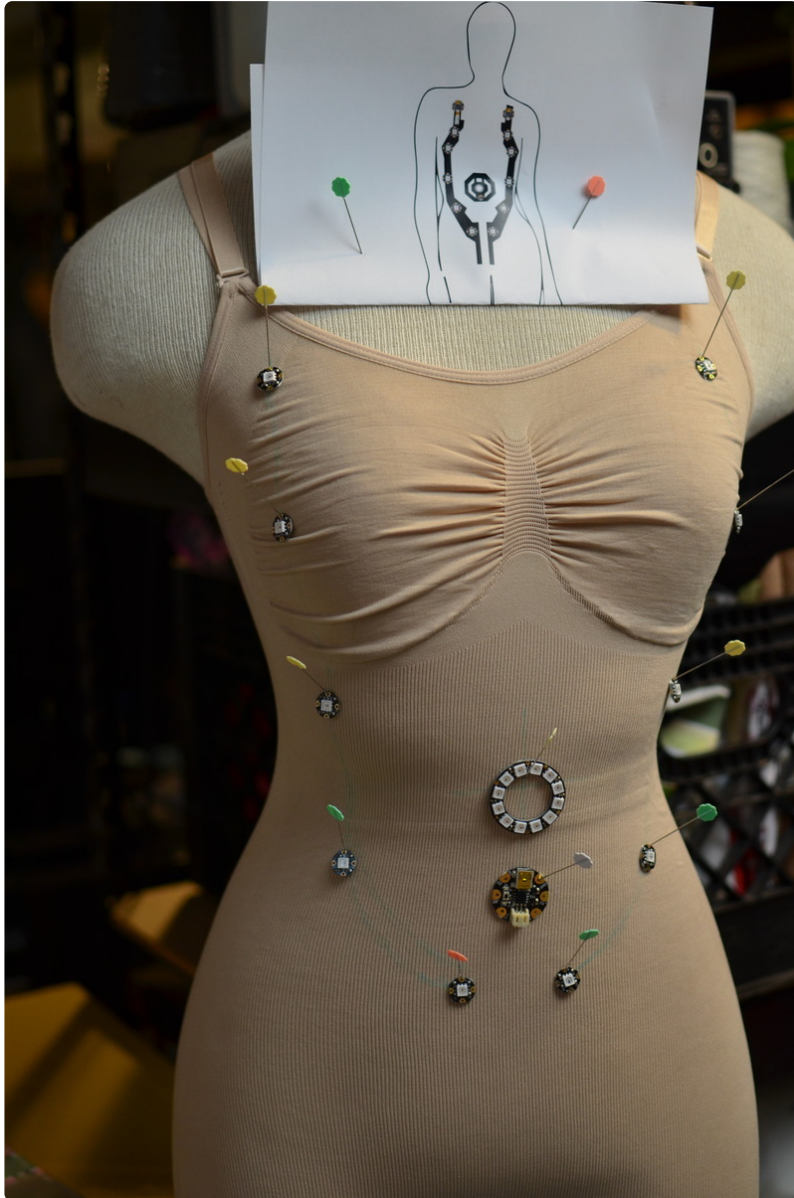
    pos2 += 1;
    if(pos2 < 0) {
        pos2 = 1;
    } else if(pos2 >= 6) {
        pos2 = 0;

    }

}

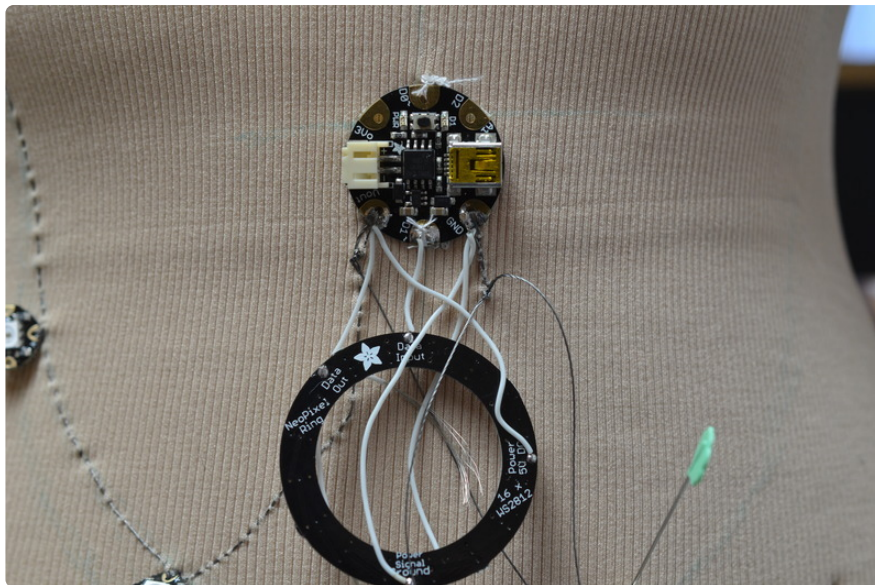
```

Assembly



A body form is a must for this project! You can make your own if necessary, but you really shouldn't attempt this project without one.

Suit up your bodyform with your shapewear undergarment and mock up the layout of circuitry using pins and a marking pen or pencil.

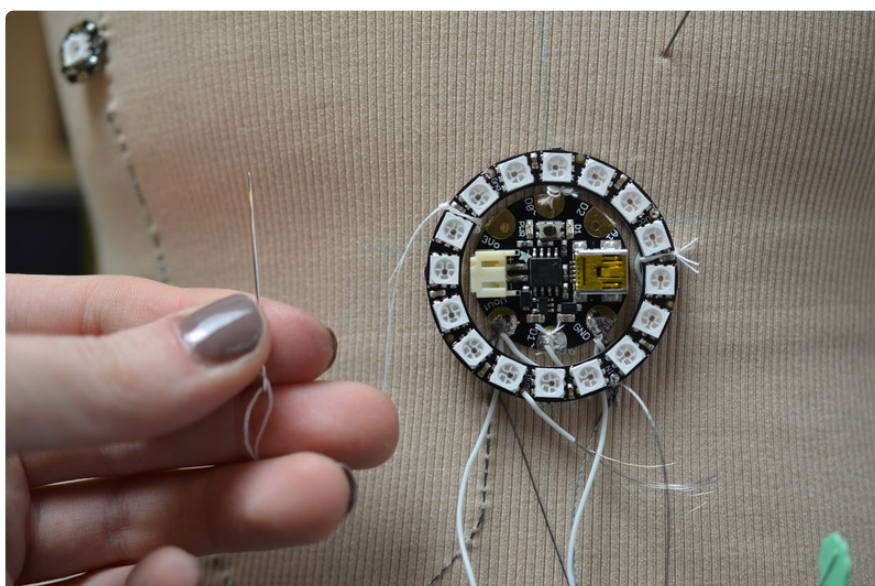


We'll be both soldering and sewing to GEMMA, so let's do the soldering first. Attach two silicone-coated wires to Vout and GND, and one to D1. Then connect one set up to Power, Ground, and Data Input on the 16-NeoPixel ring, respectively.

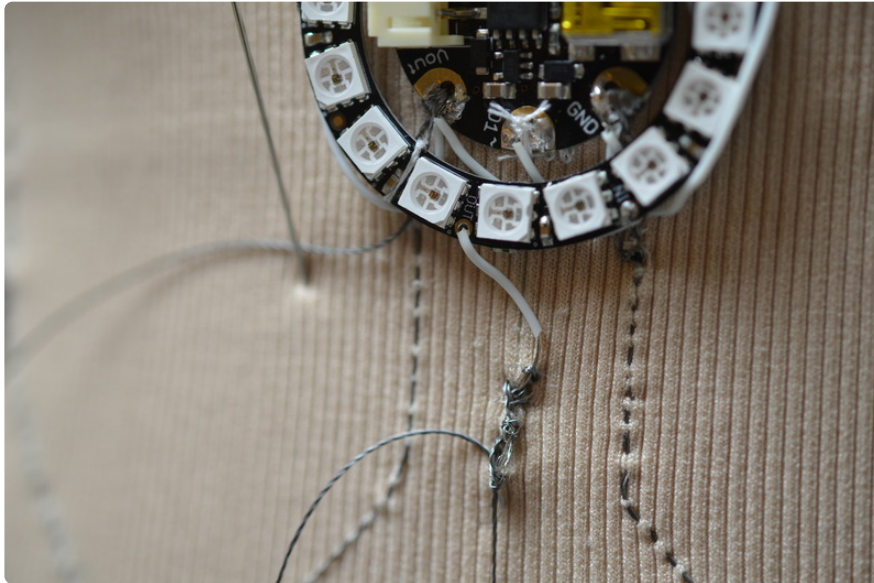
Solder another wire to the NeoPixel ring's Data Out pin.

When you're happy with the layout, stitch GEMMA to the undergarment with plain cotton thread around one of the unused pins. Be careful not to accidentally stitch it to your body form! You may need to pull the stretchy fabric away from the form to be sure the needle doesn't pierce any fabric cover that might be on your body form.

Before affixing the ring to the garment, start stitching connections from Vout and GND on GEMMA to head to your sewable pixels. You don't have to complete the buses now, this is just so you don't have to sew uncomfortably underneath the NeoPixel ring, so you can leave long thread tails hanging out for now.



Secure the NeoPixel ring to the garment with plain thread.



Attach the Data Out wire to the garment by stripping a long section of the wire and stitching over and around it with conductive thread. Double the wire over to make a loop, make sure that it can't pull loose from the thread connection!

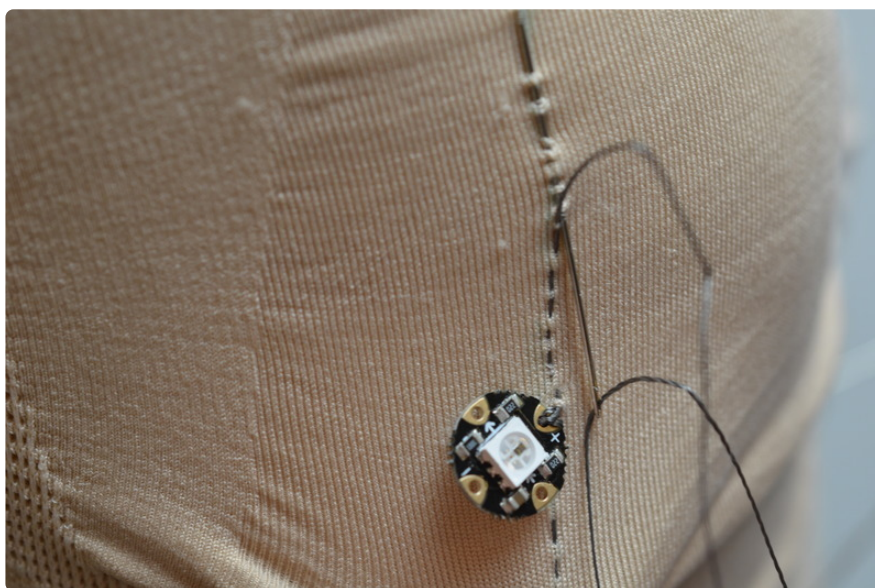


Since we want the right and left pixel chains to perform the same animation, we'll just split the data connection and send it to both at the same time-- they'll act "mirror image" this way without extra code! Stitch the data line to the inputs on the first two sewable pixels.

Use those extra power and ground wires to criss cross and transition to conductive thread for the far power and ground buses on the pixel chains (as pictured).



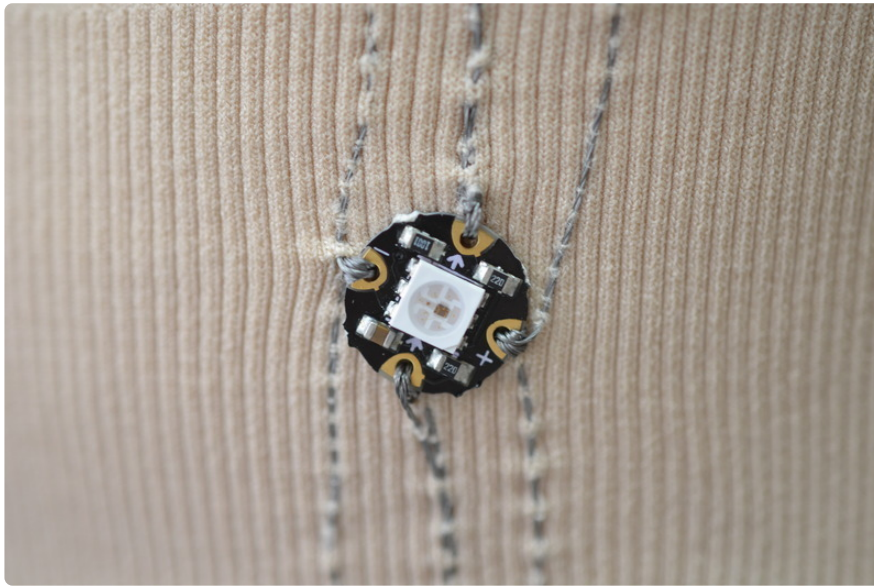
Pick up your power and ground threads and stitch on the rest of your pixels. Pull the fabric from the body form and carefully pick up bits of material to create a running stitch on your needle, then pull through. Double check you're not sewing to your form.



Gently stretch the fabric as you sew, to make sure there is enough slack in the thread to accomodate for the stretching necessary to put the garment on. Stainless steel conductive thread is very strong and will not stretch-- it will instead tear through your delicate knit fabric and could ruin your project... we've said it before but it's worth mentioning again: this project requires challenging and technical execution, if it's your first foray into conductive thread we strongly recommend you try something on woven (not stretchy) fabric first!

At each pixel, wrap tightly around the connection pad, then tie one knot and continue on without breaking the thread. Since the running stitch should be pretty loose (to

allow for more stretch), this knot prevents the loops around the pixel from opening up and possibly shorting other components on the pixel.

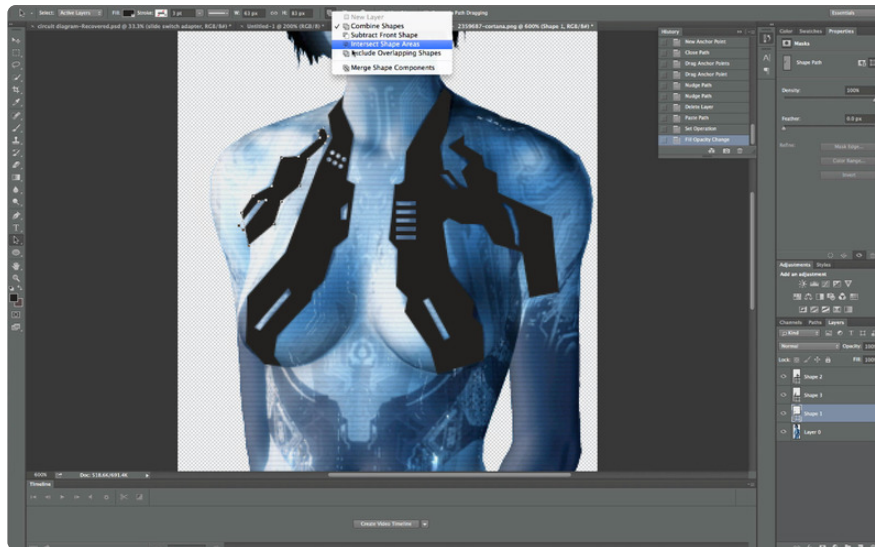


Tidy stitching is the name of the game here! Here's a sewn pixel closeup.



It's hard to test your circuit until you've finished stitching! Once you've sewn your pixel chains, you can leave the excess power and ground threads attached in case you need to double over your connections. Just coil them up and pin them to your body form, making sure they're not touching each other.

3D Printing

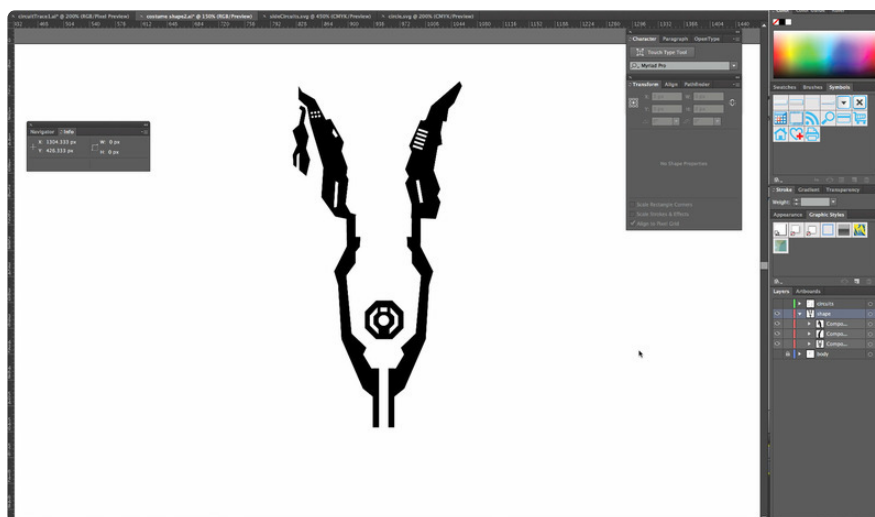


Trace Reference

Build the stencil files by tracing out the details in your favorite vector app to create the paths or download the svg vectors paths!

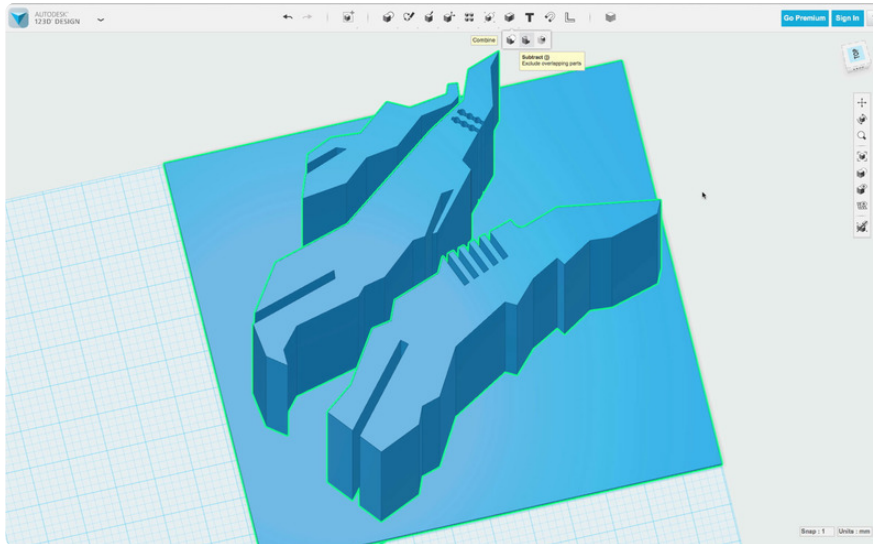
Download SVG Paths

<https://adafru.it/dZj>



Measure body length

Create a document with the correct body measurements for each part of the upper, middle and lower chest.



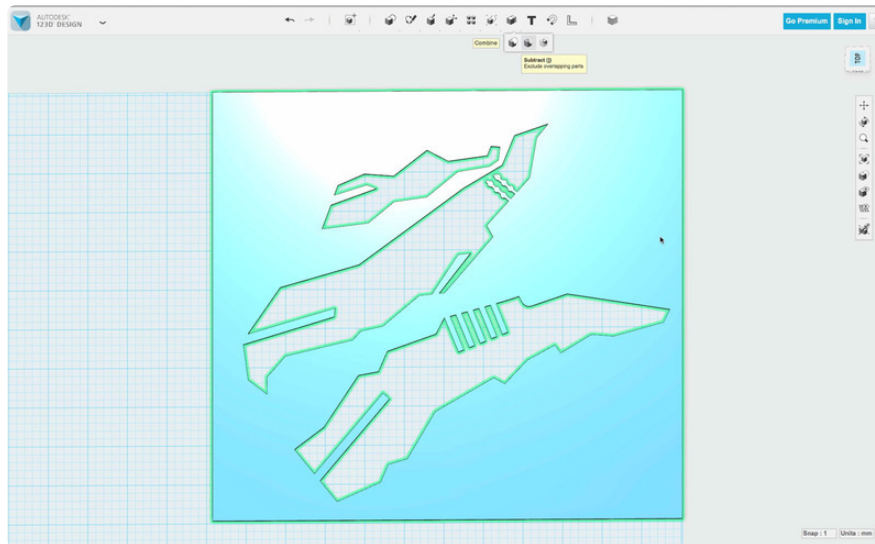
Extrude Paths

Next we'll build the solid pieces that will act as the cutouts for the stencils, or you download the STLs for the shoulders, chest and lower body.

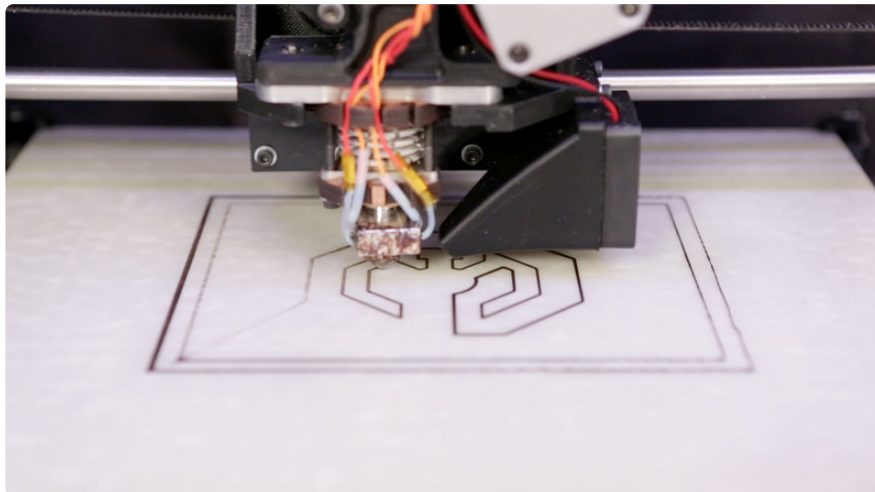
Make a square as big as the build plate you'll be printing on, here we made it 30cm x 30 cm. Import the svg file, snap the bottom faces to the top of the square, just a couple millimeters below the box.

Download Stencil STL

<https://adafruit.it/dZk>



Make the stencil about 1 to 1.5 mm thick. We found that a thinner stencil produced cleaner edges when spray painting on the unitard.



Print Stencil

Clean your build plate and preheat your extrude. We used a Flexible build plate for easy removal. Prints should take about an hour to complete.



Mark LEDs on unitard

Mark the placement of where the NeoPixel LEDs are by wearing the unitard over the spanx.



Paint Stencil

Once we have marked our NeoPixel placements we mock up where we'd like the circuit trace details. Use pieces of blue tape once you're happy with the placement of each stencil.

Keep the unitard stretch for painting. We used a box to keep the parts stretch out while spraying.



Let the paint dry for about 30mins and wear to your next halloween or cosplay event!