



DRAFT PUNK

Created by Anthony Sherwood



Last updated on 2018-08-22 03:44:44 PM UTC

Guide Contents

Guide Contents	2
Construction	3
The Code	10

Construction

Since I don't have a 3D printer big enough to print an [entire Daft Punk helmet](https://adafru.it/Ce8) (<https://adafru.it/Ce8>), and I don't have a lot of cash to spend on Halloween this year, I elected to make the poor man's version.

You'll need:

- a [Pro Trinket](http://adafru.it/2010) (<http://adafru.it/2010>) with [battery backpack](https://adafru.it/e0w) (<https://adafru.it/e0w>)
- a [battery](https://adafru.it/drL) (<https://adafru.it/drL>)
- a [switch](https://adafru.it/drN) (<https://adafru.it/drN>)
- and of course, a [1 meter strip of NeoPixels](https://adafru.it/e7W) (<https://adafru.it/e7W>).

We're going to shove all of that inside of this \$5 cobra commander mask from Party City...



Remove the cardboard packaging at the top of the mask, and the sticker from the inside. We will also need to remove the foam padding, so pull as much of it as you can out in a single block, and then you'll have some scraping to do to get the rest out. Just get as much as you can... it's ok if you don't get it all.





Next up, we're going to start assembling the electronics. The first thing you should do is get your Trinket Pro up and running, so head on over to the tutorial for that and come back when you've got your battery backpack soldered on and charging your battery. On the bottom of the trinket you can clip off the header pins, since we won't be needing them, but leave the pins on the top. You will need to solder the power leads from the neopixel strip to two of those pins.

[Adafruit Pro Trinket LiPoly/Lilon Backpack Tutorial \(https://adafru.it/iPd\)](https://adafru.it/iPd)

Then solder the power wires from your NeoPixel strip to the BAT+ and GND pins coming out of the top of the backpack. There will be one wire left, for signal... solder it to pin 4 of the Trinket Pro.

After you've done that, open up your Arduino IDE and load the example sketch for a NeoPixel strip. Change the pin to 4, and push it to your Trinket. Remember to push the bootloader button BEFORE you try to upload, or it won't work.

If your strip starts lighting up and doing color wipes, you're good to go!

We're going to cut the strip up next...



This is the longest and hardest part of any NeoPixel strip project... use an exacto knife to cut the strips exactly in the middle of two adjacent neopixels (you'll see a dotted line on the PCB), and then solder wires to the pads from each strip to the next one. Make sure all the data arrows are pointing in the same direction, for a constant flow throughout the strip.

It's a good idea to reinforce those soldered joints, and I did so by wrapping each end of the strip with electrical tape.

To fit inside the mask, I divided my strip into 5 sections, the three in the middle being 12 pixels long, and the two on the ends being 11 pixels long, for a total of 58 pixels. After taping the ends of the strips, I scored the tape between each adjacent piece of strip so that I could spread the lights out in a triangular pattern.



Next, it's time to glue this into the mask. But first, turn on the Pro Trinket again, and make sure everything is working! If it isn't, you'll have to go back and fix some broken solder joints.

Also, when holding this inside of the mask, I noticed that you could see straight through the mask to the writing on the PCB, so let's fix that. I did so by cutting strips of office printer paper and gluing them to the inside of the mask where I wanted the strips to go.



I tried E6000 and hot glue, but neither worked all that well for holding in the strips, so eventually I used superglue to affix the strips to the paper. It worked well, but your mileage may vary. If you can find it, Permatex 66B silicone adhesive works great on the NeoPixel strips.



To make it easier to see out of the mask, you can also line the back of the strips with electrical tape...



And now, let's glue back in that foam so we have some padding between all of this and our foreheads...



Now on to the Trinket end of things... if you wrap up all the wires from your trinket to the mask with electrical tape, then the trinket can sit above your head while wearing the mask. The switch can go along the side of the mask, down near your ear, and sit on the inside of the mask. The battery is easily taped right along your forehead, where it will stay put.



Finally, you should do what you can to keep your Pro Trinket from getting wet. I just put it inside the little baggie that my switch came in.



Congratulations! You have now made a very quick and dirty approximation of a Daft Punk Helmet! I wore mine with my blue hoodie, to complete the \$5 cosplay Cobra Commander theme.

You can either use it with the color wipe test sketch that it has already, or head on over to the code page, where we will create a larsen scanner type animation for it.



The Code

Load up this animation to your Pro Trinket! You'll need the [NeoPixel library \(https://adafru.it/dhv\)](https://adafru.it/dhv) if you don't have it already.

```
#include <Adafruit_NeoPixel.h>

#define PIN 4

Adafruit_NeoPixel strip = Adafruit_NeoPixel(58, PIN, NEO_GRB + NEO_KHZ800);

int snakeLength=10; // How long of trails do you want?
int frameDelay = 30; // Lower numbers = faster animation
int waveDirection = 0; // 0 = right, 1 = left

int rgb[3] = {255,0,0}; // Color for the main pixel
int trailingRGB[3] = {0,0,0}; // Color for the trailing pixels

int hue = 60; // We're going to start off with red

int currentLED = 0; // Current LED.
int trailingLED = 0; // LED that is trailing the current LED

uint16_t i, j;

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

void loop() {
  if (currentLED > 0 && currentLED <= strip.numPixels()) {
    strip.setPixelColor(currentLED, rgb[0], rgb[1], rgb[2]);
  }
  if (waveDirection == 0) { // Wave travelling right
    for (j = 0; j < snakeLength; j++) {
      trailingLED = currentLED - j;
      if (trailingLED > 0) {
        hsi2rgb(hue,1,1/(j+0.1),trailingRGB);
        if(trailingLED > 0 && trailingLED <=strip.numPixels()) {
          strip.setPixelColor(trailingLED, trailingRGB[0], trailingRGB[1], trailingRGB[2]);
        }
      }
    }
    trailingLED--;
    if(trailingLED > 0) {
      strip.setPixelColor(trailingLED,0,0,0);
    }
  } else { // Wave travelling left
    for (j = 0; j < snakeLength; j++) {
      trailingLED = currentLED + j;
      if (trailingLED < strip.numPixels()) {
        hsi2rgb(hue,1,1/(j+0.1),trailingRGB);
        if(trailingLED > 0 && trailingLED <=strip.numPixels()) {
          strip.setPixelColor(trailingLED, trailingRGB[0], trailingRGB[1], trailingRGB[2]);
        }
      }
    }
  }
  trailingLED++;
}
```

```

    trailingLED++;
    if(trailingLED < strip.numPixels()) {
        strip.setPixelColor(trailingLED,0,0,0);
    }
}

strip.show();

if (waveDirection == 0) {
    currentLED++;
} else {
    currentLED--;
}
if (currentLED > strip.numPixels()) {
    waveDirection = 1;
    currentLED-=2;
}
if (currentLED < 0) {
    waveDirection = 0;
    currentLED+=2;
}
hue+=4;
if(hue > 360) {
    hue = 0;
}
delay(frameDelay);
}

// (FUNCTION STOLEN FROM http://blog.saikoled.com/post/43693602826/why-every-led-light-should-be-using-hs)
// This function will make it easier to create lighter and darker versions of the same color.
//
// Function example takes H, S, I, and a pointer to the
// returned RGB colorspace converted vector. It should
// be initialized with:
//
//
// int rgb[3];
//
// in the calling function. After calling hsi2rgb
// the vector rgb will contain red, green, and blue
// calculated values.

void hsi2rgb(float H, float S, float I, int* rgb) {
    int r, g, b;
    H = fmod(H,360); // cycle H around to 0-360 degrees
    H = 3.14159*H/(float)180; // Convert to radians.
    S = S>0?(S<1?S:1):0; // clamp S and I to interval [0,1]
    I = I>0?(I<1?I:1):0;

    // Math! Thanks in part to Kyle Miller.
    if(H < 2.09439) {
        r = 255*I/3*(1+S*cos(H)/cos(1.047196667-H));
        g = 255*I/3*(1+S*(1-cos(H)/cos(1.047196667-H)));
        b = 255*I/3*(1-S);
    } else if(H < 4.188787) {
        H = H - 2.09439;
        g = 255*I/3*(1+S*cos(H)/cos(1.047196667-H));
        b = 255*I/3*(1+S*(1-cos(H)/cos(1.047196667-H)));
        r = 255*I/3*(1-S);
    } else {
        H = H - 4.188787;

```

```
b = 255*I/3*(1+S*cos(H)/cos(1.047196667-H));  
r = 255*I/3*(1+S*(1-cos(H)/cos(1.047196667-H)));  
g = 255*I/3*(1-S);  
}  
rgb[0]=r;  
rgb[1]=g;  
rgb[2]=b;  
}
```