# Larson Scanner Shades (Trinket-Powered NeoPixel LED Strip Glasses)

Created by Phillip Burgess

https://learn.adafruit.com/larson-scanner-shades

Last updated on 2022-12-01 02:03:07 PM EST

## Table of Contents

## Overview

We must repeat! Our “Kaleidoscope Eyes ()” goggles project was so popular that we wanted an easier option for newcomers to electronics. These “Larson Scanner” shades offer a lot of bling with fewer parts and less tricky connections. Freedom of choice is what you got!

The Larson scanner is named after Glen Larson, producer of Knight Rider and the original Battlestar Galactica television series, both of which prominently featured the effect as the “eyes” of KITT, his nemesis KARR, and the Cylon Centurions.

Larson scanners were traditionally red (or yellow in KARR's case), but thanks to the magic of NeoPixels you can change the software to use any colors you like.

This guide was written for the 'original' Gemma and Trinket boards, but can be done with either original or M0 Gemma or Trinket. We recommend the Trinket M0 or Gemma M0 as it is easier to use and is more compatible with modern computers!

## Tools Needed

This is a soldering project, albeit a small one. You will need the common soldering paraphernalia of a soldering iron, solder, wire (20 to 26 gauge, either stranded or solid) and tools for cutting and stripping wire.

You'll need some method of securing the electronics inside the glasses. Hot-melt glue (with a glue gun) works well for this. Watch your fingers! Packing tape could also be used.

## Parts needed

- Visor-style sunglasses — the 1980's style with a single "unibrow" rather than separate lenses. A well-stocked costume store may have something suitable, or you can search eBay for "robot sunglasses," "Devo sunglasses" or "alien sunglasses."
- NeoPixel RGB LED flex strip: 144 (http://adafru.it/1507), 60 (http://adafru.it/1138) or 30 (http://adafru.it/1376) LEDs/meter depending on your budget and desired look. This project only requires a short section — you can either make several pairs of glasses from the strip, or use the leftovers for other projects (it's fun stuff to play with!). The photo above is using 144 LEDs/meter NeoPixel strip.
- Adafruit's Trinket M0 () or Gemma M0 () are recommended.  The legacy versions Trinket 3.3v (), Trinket 5v () and Gemma v2  ()will also work.
- 3.7V 150mAh Lithium-Ion Polymer Battery (http://adafru.it/1317)
- LiPo battery charger (http://adafru.it/1304)
- JST Battery Extension Cable (http://adafru.it/1131) (not required if using Gemma)

There are lots of eyeglass options on eBay for $20 or less. We settled on the pair on the right because they pass more light (the LEDs will appear brighter). Ski or motocross goggles could also work!

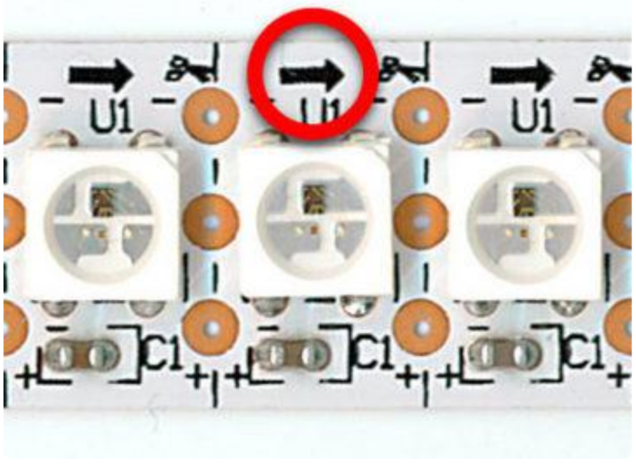
## Can I use a Gemma instead of Trinket?

Absolutely! You won't need the extra JST cable for the LiPo battery — Gemma has that plug built-in. The board is a bit wider and might be more challenging to fit, but one option is to rather than conceal it, mounting the board on the of the glasses near one temple. Geek pride!

## Wiring & Soldering

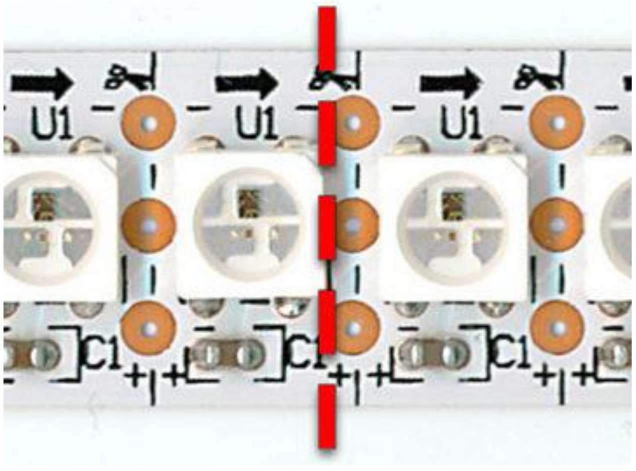
Begin by measuring how many LEDs your sunglasses will need. It's easiest just to hold the strip behind the glasses and slide it around to settle on a position, then count the LEDs. Try to place the LEDs above center, toward your forehead…this way they'll block less of your field of view.

Using the high-density NeoPixel strip (144 LEDs/meter), one of our pairs of glasses required 20 LEDs, another required 22. The 60 and 30 LEDs/meter strip would have proportionally fewer LEDs.



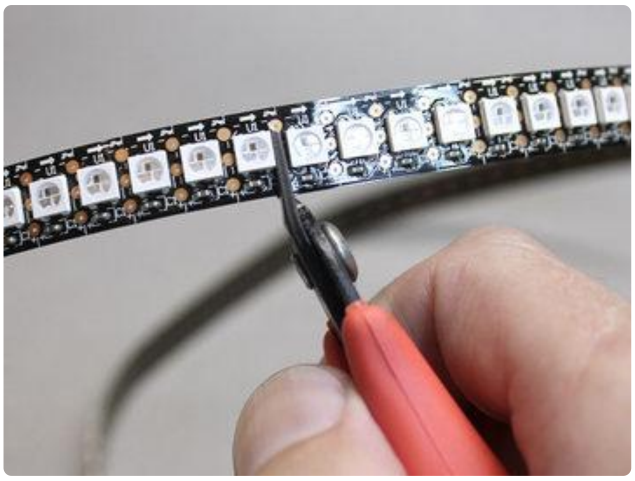
NeoPixel strip has a definite "in" and "out" end. Look for the arrows — these indicate the direction of data from the microcontroller, moving from in to out.

The Trinket/Gemma board will connect to the "in" end…that would be to the left in this view.



If using the high-density strip (144 LEDs/meter), you might choose to cut just to the left of the marked cut lines. When using the remainder of the strip in another project, this makes it easier to connect wires to the input.

This doesn't apply to 60 or 30 LEDs/m strip — these have connection points on both sides of the cut line.



Flush cutters work well for cutting the strip. You might need to make two cuts to cover the full width; one from below, one from above.

Optional: I'm not comfortable with little sharp points right next to my eyes. The strip will certainly hold in place when installed, but as an extra precaution the corners were rounded slightly at both ends of the strip. Just in case.

The vias (wire holes) on the strip are very tiny! When connecting wires, you might find it necessary to widen these holes slightly by inserting the point of a pair of tweezers and turning.

The circuit is incredibly simple, with just a few connections needed…

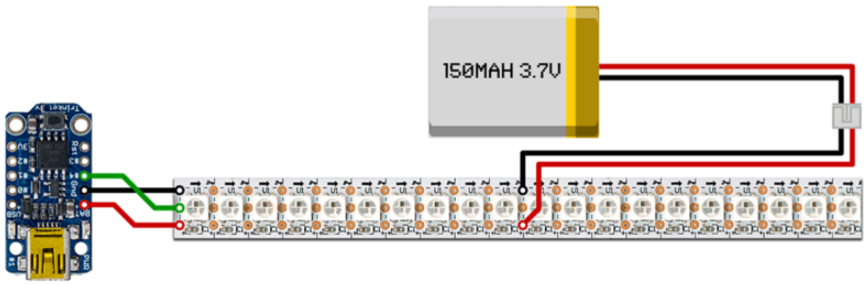
If using Trinket: Though data moves only one way through a NeoPixel strip, power transfers both ways. We use this to our advantage here, making the strip itself function as a power bus for the whole circuit. Cut the JST battery extension cable about 4" (10 cm) from the plug end (the end that connects with the battery). Strip the ends of the wires, give them a twist and then insert them anywhere along the strip; black wire to –, red wire to +, and solder in place.

The wires should be inserted back to front (soldered on the LED side); the diagram below has the wires on the front just to show how they're routed.

At the input end of the strip, three wires are then connected between the strip and Trinket board: – (minus) on the strip to Gnd on Trinket, + on the strip to 3V (or 5V) on Trinket, and the remaining connection to Pin #4 on Trinket.

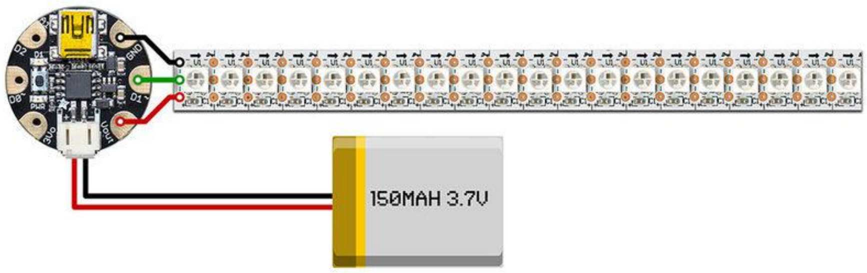
If using 60 or 30 LEDs/meter NeoPixel strip, the connections are in a different order than shown here, and are labeled GND, +5V and DIN (rather than –, + and unmarked, respectively).

This diagram uses the original Gemma and Trinket but you can also use the Gemma M0 or Trinket M0 with the exact same wiring!



Gemma is even easier. There's already a JST connector on this board, so clipping a plug off a battery extension isn't needed. Just connect – (minus) on the strip to GND on Gemma, + to Vout, and the remaining pin to D1.

If using 60 or 30 LEDs/meter strip, the note above about pin order applies to Gemma as well.



With the glasses we built upon, Trinket fit perfectly turned 90 degrees with respect to the strip (like the wiring diagram above) — the USB connector is then accessible along the top edge, for loading new code. But every pair of shades will be different. You'll probably want to use longer wires between the Trinket and strip, so you have more options when folding the pieces together; I just got carried away here with the ultra-short wires.

Reiterating a prior point: wires are run along the back side of the strip, and soldered on the front.

After soldering, all of the protruding wire points were trimmed.

## Arduino Code

The Arduino code presented below works equally well on all versions of GEMMA: v2 and M0. It also works on all versions of Trinket and Trinket M0. But if you have an M0 board, consider using the CircuitPython code on the next page of this guide, no Arduino IDE required!

If this is your first time using Trinket or Gemma, work through the guides first:

- Adafruit Trinket M0 Guide ()
- Adafruit Trinket Guide ()
- Adafruit Gemma M0 Guide ()
- Adafruit Gemma Guide ()

You need to customize some settings in the Arduino IDE. Once you have it up and running (test the 'blink' sketch), then download and install the NeoPixel library:

Click to download the NeoPixel library

Installing Arduino libraries is a frequent stumbling block. If this is your first time, or simply needing a refresher, please read the All About Arduino Libraries () tutorial.

If the library is correctly installed (and the Arduino IDE is restarted), you should be able to navigate through the “File” rollover menus as follows:

File→Sketchbook→Libraries→Adafruit_NeoPixel→strandtest

You'll need to change one line in this code. If using Trinket, change line #4 to:

```
#define PIN 4
```

And for Gemma, use:

```
#define PIN 1
```

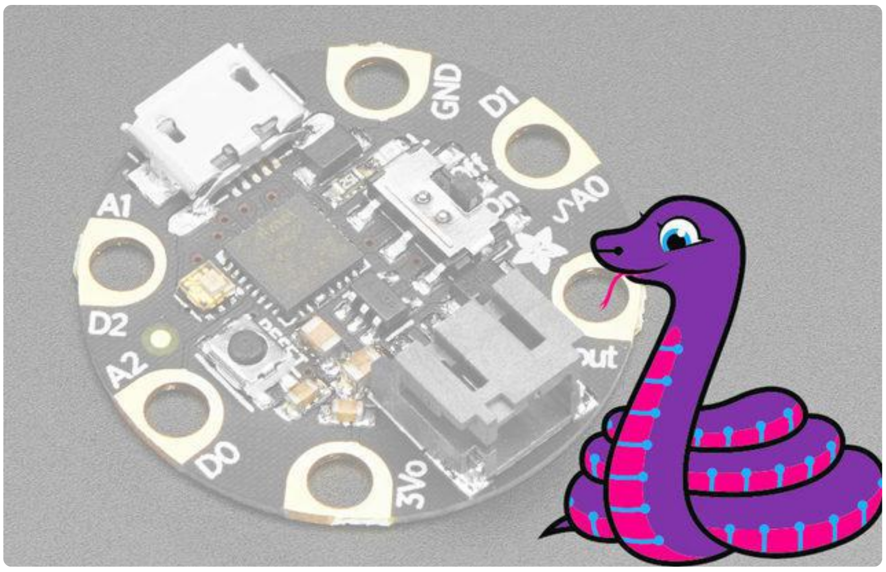
From the Tools→Board menu, select Adafruit Trinket 8 MHz or Adafruit Gemma as appropriate. Connect the USB cable between the computer and Trinket, press the reset button on the board, then click the upload button (right arrow icon) in the Arduino IDE. When the battery is connected, you should get a light show from the LEDs.

If that's working, you can then copy and paste this Larson scanner code into a new sketch. Change N_LEDS to match the number of LEDs in your strip, and if using Gemma change PIN to 1. Upload and get ready to be a Cylon!

```
// SPDX-FileCopyrightText: 2017 Mikey Sklar for Adafruit Industries
//
// SPDX-License-Identifier: MIT

#include <Adafruit_NeoPixel.h>

#define N_LEDS 22
#define PIN     4

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

void setup() {
  strip.begin();
}

int pos = 0, dir = 1; // Position, direction of "eye"

void loop() {
  int j;

  // 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(pos - 2, 0x100000); // Dark red
  strip.setPixelColor(pos - 1, 0x800000); // Medium red
  strip.setPixelColor(pos    , 0xFF3000); // Center pixel is brightest
  strip.setPixelColor(pos + 1, 0x800000); // Medium red
  strip.setPixelColor(pos + 2, 0x100000); // Dark red

  strip.show();
  delay(30);

  // 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(j=-2; j<= 2; j++) strip.setPixelColor(pos+j, 0);

  // Bounce off ends of strip
  pos += dir;
  if(pos < 0) {
    pos = 1;
    dir = -dir;
  } else if(pos >= strip.numPixels()) {
    pos = strip.numPixels() - 2;
    dir = -dir;
  }
}
```

This code works equally well on an Adafruit Hallowing board with a NeoPixel strip plugged into the NEOPIX port. Just change the number on the “N_LEDS” line to match your strip length. No need to change PIN, the default (4) already matches the NeoPixel connector on this board.

## CircuitPython Code



GEMMA M0 and Trinket M0 boards can run CircuitPython — a different approach to programming compared to Arduino sketches. In fact, CircuitPython comes factory pre-loaded on GEMMA M0 and Trinket M0. If you’ve overwritten it with an Arduino sketch, or just want to learn the basics of setting up and using CircuitPython, this is explained in the Adafruit GEMMA M0 () guide () and the Adafruit ()Trinket M0 guide ().

These directions are specific to the “M0” boards. The original GEMMA and Trinket with an 8-bit AVR microcontroller doesn’t run CircuitPython…for those boards, use the Arduino sketch on the “Arduino code” page of this guide.

Below is CircuitPython code that works similarly (though not exactly the same) as the Arduino sketch shown on a prior page. To use this, plug the GEMMA M0 or Trinket M0 into USB…it should show up on your computer as a small flash drive…then edit the file “main.py” with your text editor of choice. Select and copy the code below and paste it into that file, entirely replacing its contents (don’t mix it in with lingering bits of old code). When you save the file, the code should start running almost immediately (if not, see notes at the bottom of this page).

If GEMMA M0 or Trinket M0 doesn't show up as a drive, follow the GEMMA M0 or Trinket M0 guide link above to prepare the board for CircuitPython.

You'll need to change one line #6 which represents the output pin that controls the NeoPixels.

If using Trinket M0:

```
pixpin = board.D4
```

If using GEMMA M0:

```
pixpin = board.D1
```

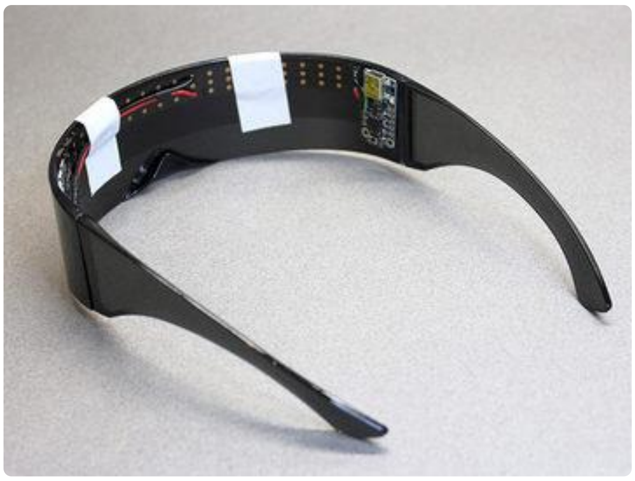
```
# SPDX-FileCopyrightText: 2017 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time

import board
import neopixel

numpix = 22  # Number of NeoPixels
pixpin = board.D1  # NeoPixels pin. For Gemma M0 = D1, Trinket M0 = D4
strip = neopixel.NeoPixel(pixpin, numpix, brightness=1, auto_write=False)
pos = 0  # position
direction = 1  # direction of "eye"

while True:
    strip[pos - 2] = ([16, 0, 0])  # Dark red
    strip[pos - 1] = ([128, 0, 0])  # Medium red
    strip[pos] = ([255, 48, 0])  # brightest
    strip[pos + 1] = ([128, 0, 0])  # Medium red

    if (pos + 2) < numpix:
        # Dark red, do not exceed number of pixels
        strip[pos + 2] = ([16, 0, 0])

    strip.write()
    time.sleep(0.03)

    # 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 j in range(-2, 2):
        strip[pos + j] = (0, 0, 0)
        if (pos + 2) < numpix:
            strip[pos + 2] = (0, 0, 0)

    # Bounce off ends of strip
    pos += direction
    if pos < 0:
        pos = 1
        direction = -direction
    elif pos >= (numpix - 1):
        pos = numpix - 2
        direction = -direction
```

This code can also work on an Adafruit Hallowing board with just a small change to lines 6 and 7:

```
numpix = 30
pixpin = board.EXTERNAL_NEOPIXEL
```

## Final Assembly and Use

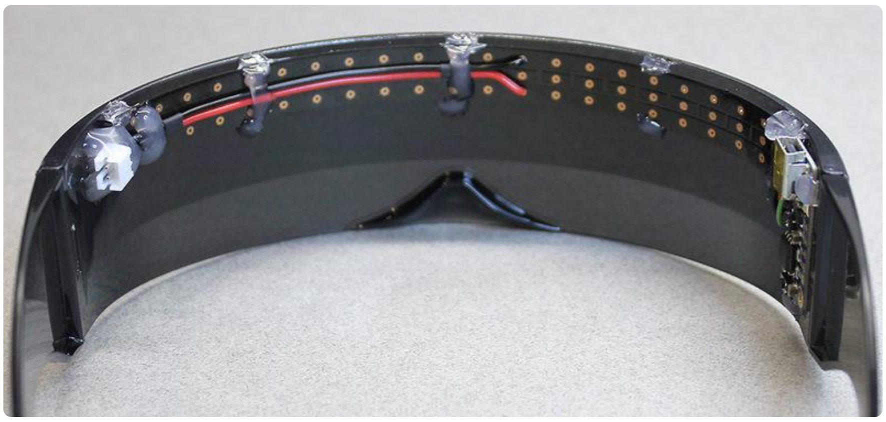
Arts & crafts time!



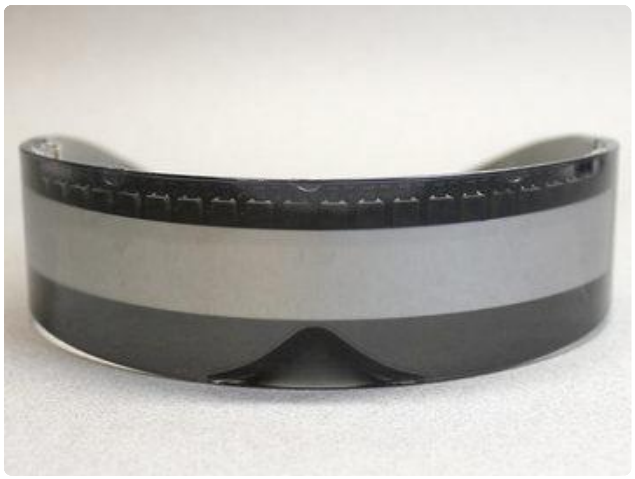
The electronics can be held in place temporarily with masking tape while a more permanent bond is arranged…

A few strategic dabs of hot-melt glue then hold the strip, Trinket, wires and JST plug in place. The latter gets a sizable glop, since the battery will be plugged and unplugged repeatedly here. Once the glue is fully set, the masking tape can be removed.

Hot melt glue works fine to hold the wires and Trinket in place, but can be disappointingly nondurable. E6000 craft adhesive (one of our favorites) dries clear and flexible with a very strong bond, but the fumes are a bit noxious and it takes 24 hours to set completely. Choose wisely.

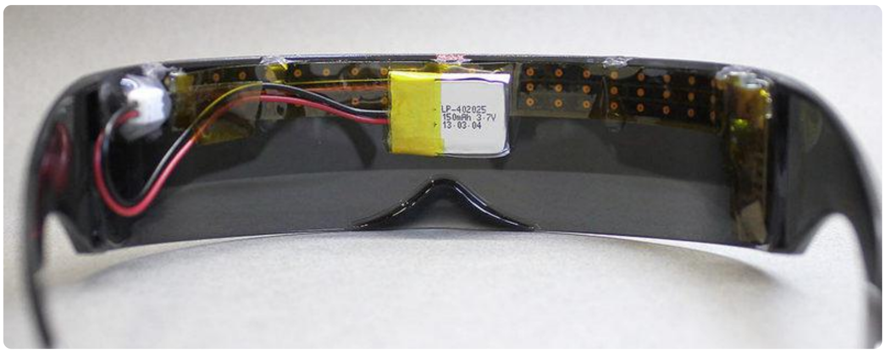


To prevent accidental contact with skin (which is slightly conductive, especially if damp with perspiration), the back side of everything is covered with tape.



Hot glue and tape are rather kludgey, but it’s all hidden on the inside. From the front everything looks pretty slick (and the blinking LEDs will provide further distraction from any imperfections).

The tiny LiPo battery can be held with tape at the bridge of the nose (between the eyes it won’t block vision), or could be affixed to one of the temples.



Bingo! You're done!

Using the code as provided, the battery should run for about two hours. Different colors and patterns will have a different current draw and run time; this is nearly impossible to predict, sometimes the best way is just to plug it in with a freshly topped-off battery and see how long it runs.

To recharge the battery, unplug it from the socket on the glasses and use the LiPo charger. The battery does not charge from the USB connection.

## SAFETY AND COMMON SENSE

Unlike the LED goggles, you can wear these shades on your eyes in an appropriate setting. Just remember that your sight is limited as the electronic parts may block peripheral vision. Also, LEDs look best in the dark, but navigating in an already-dark setting is made even more difficult by sunglasses. Use common sense; they're fun and nifty looking for indoor parties, but don't go driving at night with them.