

Build custom LED snowboarding goggles

Shine bright this winter with LED snowboarding goggles



Sophy Wong

@sophywong

Sophy Wong is a designer, maker, and avid creator. Her projects range from period costumes to Arduino-driven wearable tech. She can be found on her YouTube channel and at sophywong.com, chronicling her adventures in making.

W

inter is here, and that means snow-blanketed days and long, dark winter nights. In other words, it's the perfect season for flashy wearable projects! In this tutorial, we'll put a strip of programmable LEDs into a pair of snowboarding goggles for a futuristic look. Sport them in the snow or at the lodge, and bring a blinking rainbow of joy to your winter festivities.

If you're a generalist, rejoice! We'll do some soldering, sewing, hot gluing, and even a little programming. This is a great project for beginners with a little bit of soldering experience. For advanced folks, this is a fun, one-day build.

Wearable projects are a great way to combine different types of making and design, and there are plenty of ways to customise your build.

Your first design choice is: what goggles to use? Snowboarding goggles are a good base to start with, as most will have ample room for the LEDs, and foam sides that are easy to pass wires through. If you've already invested in an expensive pair of snowboarding goggles, you may want to purchase a less expensive pair for this. Your author purchased the pair used here for about \$40 (£30).

A strip of Adafruit NeoPixels works perfectly in this project, and can be cut to size for a custom fit. If this is your first project using NeoPixel strips, prepare to be hooked! These are flexible metal strips that are packed with individually addressable LEDs. They come in several varieties and, in this project, we're using the most densely packed option: a whopping 144 LEDs per metre. We'll remove the protective sleeve the NeoPixels come in but, for projects where the strip is exposed to the elements, you'll want to leave it on. You can choose between a black or white base strip, the black disappears nicely behind darkly tinted goggles.

An Adafruit Gemma microcontroller is another good choice for this build: it's small, sewable, -



TUTORIAL



YOU'LL NEED

- ◆ **Snowboarding goggles**
- ◆ **Adafruit NeoPixel digital RGB LED strip**
- ◆ **Adafruit Gemma microcontroller**
- ◆ **30 AWG silicone-coated wire**
- ◆ **3.7V LiPo battery**
1600 mAh
- ◆ **4-way stretch Spandex**
to match your goggles, about 20 cm square
- ◆ **Black gaffer tape**

Above ◆
Fabric tape measures are easier to bend around the goggles

Right ◆
A single snip and our LED strip is the correct length

and has a built-in on/off switch. Instead of sewing the circuit, we'll solder our electrical connections and use the remaining holes to secure the Gemma in place. Programming the Gemma is easy with the Arduino IDE and, even if you're new to coding, we'll get up-and-running quickly by modifying an example sketch included with the NeoPixel library. For experienced coders, this is a great place to make this project your own with a unique animation.

A word about safety: you should be able to wear these modified goggles comfortably over your eyes. We'll be backing the LED strip with tape, but a small amount of light may still come through. Remove the

goggles if you experience discomfort. You should have good visibility out of the goggles, however some peripheral vision may be blocked, and you should not wear them while actually snowboarding, skiing, piloting a spaceship, or doing any other activity where impeded vision would be dangerous.

The circuit in this project is simple and versatile, it can be used in many other wearable projects where a strand of programmable LEDs would work. You could use a NeoPixel ring instead of a strand for an arc-reactor-inspired glove. Or, swap the snowboarding goggles for a dog collar, and make your furry companion festive and visible during winter walks.

PREPARE THE NEOPIXEL STRIP

Start by measuring the inside of your goggles. Remove any protective film on the inside of the visor, and measure the length of the visor on the inside, at the very top of your goggles. Our goggles measured about 18cm.

Next, cut your neopixel strip to size using your goggle measurement. Note the cutting lines on the strip. Cut through the middle of the contacts, so that each piece has usable contacts to solder to. The contacts are very tiny, so aim carefully before you cut. Wire cutters are fine for this, but trauma scissors are useful. Cut through the silicone sleeve and the NeoPixel strip at the same time, then remove the strip





Above ♦ The NeoPixels should leave enough space to see properly

from the sleeve. The cut corners will be sharp, it's a good idea to round them, but be careful not to cut off the contacts or any circuitry.

Test the fit by placing the strip into your goggles. It should fit comfortably at the very top of the visor, with a little bit of space at each end. If it buckles or flexes, it's too long. Cut off one LED and check again.

INSTALL THE NEOPIXEL STRIP

Cut three lengths of 30 AWG silicone-coated wire, about 15 cm long each. Use different colours of wire to keep things organised.

Strip and tin one end of each wire. Note the directional arrows that show which way data flows through the NeoPixel strip. Tin the contacts on the input end of the NeoPixel strip, and solder the wires to each contact. We used red for power, black for ground, and white for data. Add strain relief to the wires by applying a small amount of hot glue on each connection.



Above ♦ Measure twice, cut once and your strip should fit perfectly

Place the NeoPixel strip into your goggles with the LEDs face down against the visor and the wires pointing to the right. Thread a few centimetres of the wires through the eye of the yarn needle. Use the needle to pass the wires through the foam barrier at the side of your goggles to the inside of the goggle strap. Remove the needle and keep it handy.

Position the NeoPixel strip at the very top of the visor and apply a generous bead of hot glue to the top edge. Do one short section at a time, and be careful while holding the strip in place, as the hot glue will heat up the metal. Our strip felt secure with glue just along the top edge, but feel free to add some glue to the bottom edge if you think you need it. For strain relief, also apply a bit of hot glue to the wires on the right, just before they exit through the foam barrier.

Cover the back of the NeoPixel strip with black gaffer tape. Overlap the bottom edge slightly to secure the strip to the visor and block most of the light from the LEDs. We found it easiest to apply the tape in pieces, rather than using one long piece.

Thread the wires through the yarn needle again, and this time pass them through to the outside of the elastic strap. Give the wires a little extra slack at this point so they're not strained when putting on and removing the goggles.

CONNECT THE GEMMA

The Gemma microcontroller will be mounted where the

wires come through the strap. Hold the Gemma in place, with the JST battery connector to the left. Find the ideal length for your wires by holding them up to their respective pins. Your power wire goes to Vout, ground to GND, and data to D0. Cut the wires accordingly, then strip and tin the ends.

Feed the wires to their pins on the Gemma from the back to the front, and solder the connections on the front of the Gemma. Dab a bit of hot glue onto each connection on the back of the Gemma and let cool. Gently pull the slack of the wires to the inside of the strap, so the Gemma sits flat on the outside of the strap. Then, glue the Gemma in place and stitch to the elastic band through the unused GPIO holes (D1, A1, and 3Vo) with a needle and thread.

Download the Arduino IDE at [hsmag.cc/fKwgoe](https://www.arduino.cc/). Learn more about using Gemma and NeoPixels at hsmag.cc/mcjcLa.

TOOLS

- ♦ Measuring tape (the soft type)
- ♦ Soldering iron and solder
- ♦ Wire strippers
- ♦ Helping hand tool with clips
- ♦ Wire cutters or trauma scissors
- ♦ Hot glue gun and glue sticks
- ♦ Sewing machine (optional)
- ♦ Needle and thread
- ♦ Computer and micro USB cable



Above ♦ It takes just three solder joints to connect the LED strip

TUTORIAL

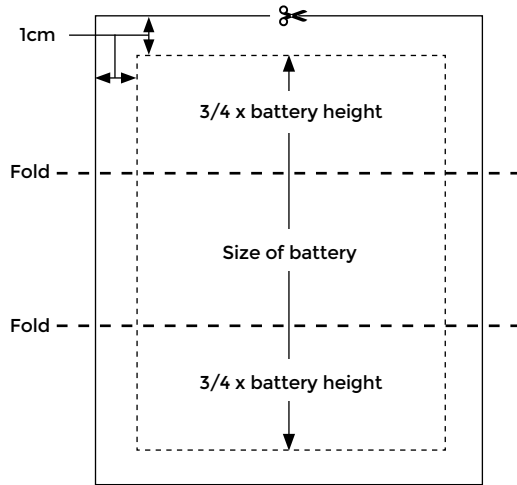


Figure 1 The pattern for the battery holder

PROGRAM THE GEMMA

Use a micro USB cable to connect the Gemma to your computer, and start the Arduino IDE. If this is your first time using NeoPixels, you'll have to download and install the Adafruit NeoPixel library. Find the Library Manager in the Sketch menu: Sketch > Include Library > Manage Libraries. Search for 'Adafruit NeoPixel by Adafruit' to find the library, and click install.

The library comes with some example code sketches that we can modify for use in this project. Go to File > Examples > Adafruit NeoPixel and choose Strandtest. This will pull up a sketch that cycles through several animations for testing your NeoPixel strip. First we'll test our strip, and then we'll modify the code to only show one of the animations the whole time.

Before uploading the code to the Gemma, we need to update two values in it to match our project: the pin we've connected our strip to, and the number of pixels in our strip. This sketch makes it easy to change these values by putting them at the top. Find this line near the beginning of the code:

// A strip of Adafruit NeoPixels works perfectly in this project, and can be cut to size for a custom fit

a few seconds, your LEDs should turn on! Watch the pretty lights cycle through the test animations, and decide which one is your favourite.

Now let's go back into the code, and choose

one animation for the Gemma to play. Scroll down to this line in the code:

```
void loop() {
```

This is the beginning of the section that cycles through the example animations. We'll simply tell the program to ignore the animations we don't want by adding two slashes at the beginning of their lines. Look through the list and find the animation you want to keep (we chose rainbowCycle). Type // at the beginning of the other lines in the list. Here's how ours looked:

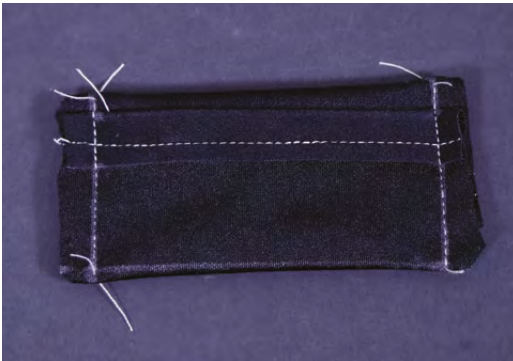
```
void loop() {
  // Some example procedures showing how to
  display to the pixels:
  // colorWipe(strip.Color(255, 0, 0), 50); // Red
  // colorWipe(strip.Color(0, 255, 0), 50); //
  Green
  // colorWipe(strip.Color(0, 0, 255), 50); //
  Blue
  // Send a theater pixel chase in...
  // theaterChase(strip.Color(127, 127, 127), 50);
```

QUICK TIP

If your project needs more I/O pins than the Gemma has, Adafruit also makes the Flora which is similar but bigger. The Lilypad is an official Arduino board similar in size to the Flora, and makes a great option for wearables projects.



Right The large pads on the Gemma are easy to solder to



```
// White
// theaterChase(strip.Color(127, 0, 0), 50); //
Red
// theaterChase(strip.Color(0, 0, 127), 50); //
Blue

// rainbow(20);
rainbowCycle(20);
// theaterChaseRainbow(50);
}
```

Leave the rest of the code as it is. Press the black button on the Gemma to start the bootloader again, and upload your new code. If your animation plays as expected, hooray! Unplug the USB cable, and let's move on to the battery.

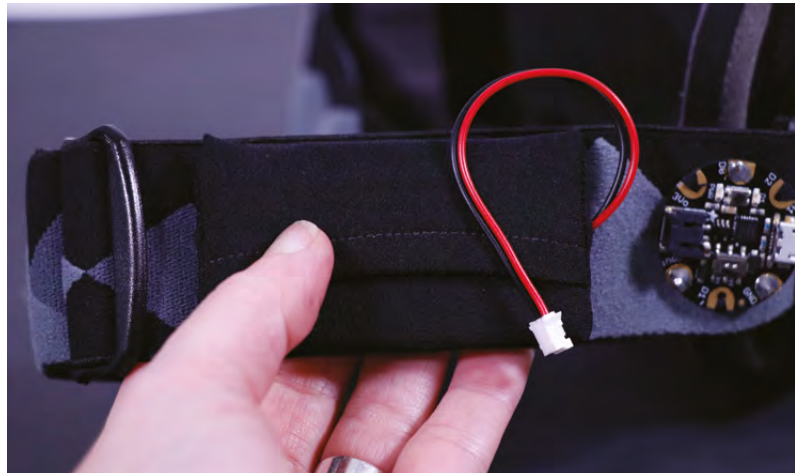
MAKE THE BATTERY HOLDER

We suggest you cover the LiPo batteries with gaffer tape for extra protection, and mark their mAh rating on the tape. Draw up a pattern for the battery holder. Start by marking the size of your battery on a piece of four-way stretch Spandex. Then use the diagram (Figure 1) to mark out the rest of your pattern. Hang on to this diagram for future projects, just scale it to the size of your battery.

Cut out the holder, then fold and sew the top and bottom edges as shown. Stretch the fabric slightly as you sew – this will build some stretch into the seams so they don't break when in use. Fold along the fold lines, right sides facing in, and sew the side seams. Trim threads and corners, and flip the holder right side out.

Note where the wires come out of the battery: at the centre of one side. Snip a small hole into one side of the battery holder for the wires to feed through. Do not cut through the seam.

Now, insert the battery into its holder: feed the JST connector wire into the opening of the holder and arefully out through the hole you made in the side, then simply slide the battery into the holder. It should be a snug fit, but you shouldn't have to wrestle too desperately with the battery to get it in the right place. And you should always handle LiPo batteries with care.



INSTALL THE BATTERY

Hold the battery and holder in place on the elastic band next to the Gemma, and mark the placement of the corners on the band. The wires of the battery should reach the JST battery connector on the Gemma comfortably. If your wires are a bit too long, like ours were, move the battery back or gently form them into a loop.

Remove the battery from its holder, and sew the holder into place on the band, stretching it to reach the corner points you marked. Hand-sew around all four sides with a whip stitch or back stitch. Insert your battery back into the holder, and plug the battery into the Gemma. If you formed your battery's wires into a loop, sew the loop down so they won't catch on anything.

LIGHT IT UP!

Flip the tiny switch on the Gemma to ON, and your LEDs should fire up! Take your new flashy goggles for a spin as they are, or turn them into a sci-fi costume piece by adding painted foam pieces or scavenged greebles. Show us how you customised this project and made it your own! ■

Above ◆
LiPo batteries are easy to connect and can power a wide range of projects

Left ■
A modified strap hides the circuit

Below ◆
The coloured LEDs add a futuristic feel to the goggles

