



Build your own Borg (BYOB)

Created by Jason Lemieux



<https://learn.adafruit.com/build-your-own-borg-byob>

Last updated on 2024-06-03 01:37:45 PM EDT

Table of Contents

Overview	3
Tools & Supplies	3
Costume Build	4
Prototyping and Coding	6
Assembly	10

Overview

The Borg Collective is one of Star Trek's most recognizable and iconic villains. After seeing them my daughter instantly knew what her next Halloween costume would be. Adapting and "assimilating" some other guides, we worked together to assemble a Gemma/Trinket & NeoPixel powered costume.

The main features include:

- some NeoPixels located throughout cycling a random pattern of colours
- a NeoPixel ring eyepiece blinking a random pattern
- a laser diode mounted to a micro servo, based off the [Laser Dog Goggles \(https://adafru.it/doggles\)](https://adafru.it/doggles) project

With this guide you can put together one of your own for your next Halloween or Sci-Fi con.

Tools & Supplies

For the costume itself here's what you'll need:

- A set of shoulder pads - we used hockey pads but a BMX chest guard looks like it would work really well too. Try your local thrift shop for some cheap ones.
- An assortment of elbow, knee, arm, shin pads. Whatever you can scrounge. Again the thrift shop is your friend here.
- A mask. We found a Phantom of the Opera style half-mask that worked out great. We did some trimming until we got the right look.
- Lots of old electronic bits. Whatever you have that looks interesting. We used mainly a old dismantled printer and a couple busted hard drives. Anything goes!
- Matte Black spray paint. To coat anything that isn't black.
- Optional: some metallic paints. I used rust and steel acrylic model paint. You can give metal bits a great weathered look with them.
- Tape: electrical & and gaffer tape. You can use hockey tape too. All black.
- Hot glue gun. For the bits of course.
- Black clothing. Pants, long sleeve shirt. We also used a balaclava, but you could shave your head if you like.

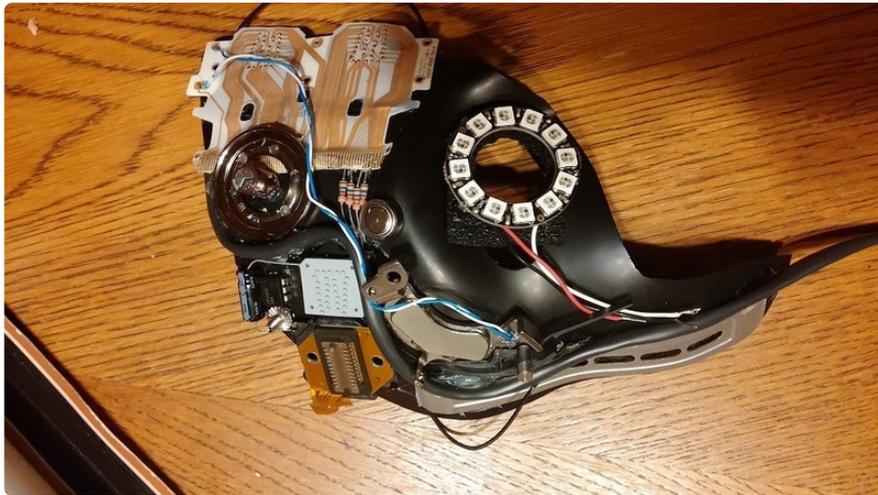
For the lighting & servo rig here's what you need:

- 2x Gemma and/or 3V Trinket. Either will work fine. I had one of each on hand so that's what I used. One controls the lighting, the other the servo.
- NeoPixels! This design uses three. Very easy to use more, or less.
- A 12 NeoPixel ring. This will form the eyepiece.
- a micro servo
- 5mW Laser diode or super bright 5mm red LED. The laser is best for full effect but if safety is a concern then use a LED. Or do what I did and mount both with interchangeable connectors.
- LiPo battery. We used a 850mAh battery and got all the charge we needed, at least 5 hours. You could use a 3xAA battery pack here too if you want.
- [Corrugated split wire loom \(https://adafru.it/edc\)](https://adafru.it/edc). Borg love hoses. Lots of them. Split wire loom is perfect to run all your wiring. And its pretty easy tell it's exactly what they [used on the show \(https://adafru.it/edd\)](https://adafru.it/edd). Get it at your local automotive or hardware store.
- Connectors. If you wire everything in one piece it will be hard to put on and remove. Break it down by sections using some [2-pin JST \(http://adafru.it/261\)](http://adafru.it/261) connectors for power and some [3-pin JST SM plugs \(http://adafru.it/1663\)](http://adafru.it/1663) for the Neopixels, laser and servo.
- 2 switches, for power to the two Gemma/Trinket devices.
- Your standard supplies: wire, heatshrink, electrical tape.
- Tools of course! Cutters & soldering iron being the essential ones.

Costume Build

First start going to town whatever electronic devices you decided to assimilate into the collective. Use anything you want: Stuff lying around, spare components, there are no rules here! Most of our bits came from a busted hp inkjet printer and two dead hard drives; along with a few other assorted components.

Grab that glue gun now and start gluing stuff to the chest piece of the shoulder pads and mask. Just stick on whatever you think will look good. Add a few wires. I also glued a few pieces to the shoulders themselves. You can add stuff to your assorted arm and leg pads too.



Above: the unpainted painted mask. The pixel ring is only attached for fitting and was removed for painting. I used some foam packaging behind the ring to get it to sit flush.

Once you've got all your bits glued you can give everything generous coat of black paint. May need more than one depending on how well it sticks to the materials you've used. Give those other pads you may be using a paint job too.



Now you can leave everything just a flat black but you can great weathered look with a couple layers of dry brushing. First I drybrushed the components with a mix of bronze and dark brown acrylics to create a rust colour. Then another layer of metallic steel acrylic. Experiment with some spare pieces to get the look you want.

If your not familiar with dry brushing techniques you can [check out some basics here \(https://adafru.it/ede\)](https://adafru.it/ede). You can see the effect below.



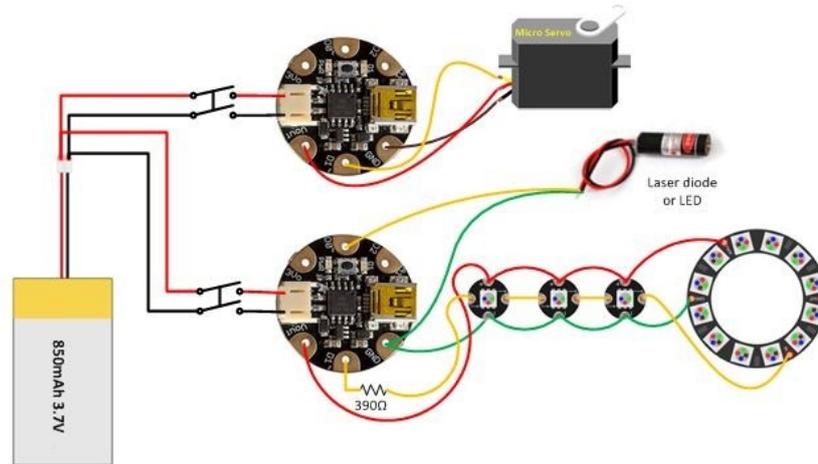
The armour pieces are pretty much built. Now onward to the electronics.

Prototyping and Coding

Here's our circuit diagram. You can easily switch in a 3V Trinket in place of one or both Gemmas.

Do not shine the laser into anyone's eyes! Especially for a prolonged period.

If you have any safety concern about the laser, then substitute in a super bright red 5mm LED and a 50 Ohm resistor. I installed both and put connectors on them to switch back and forth. The LED still looks pretty good.



First up, the servo control. You may want to refer to the [Laser Dog Goggles \(https://adafru.it/Ceg\)](https://adafru.it/Ceg) guide as it is based off that project but with a few tweaks:

- The laser has been moved to the other Gemma device. This because we wanted one switch to activate the motor and the other to activate lights.
- I wanted a slower sweep speed. To do this I changed the moveAmount and servoPos variables from int to float, which opens up decimal increments. So I can use a moveAmount of 0.5
- There is some code that makes the servo jump +/- 15 degrees at random. Why? Because we thought it looked creepier.

Download the Adafruit SoftServo library here.

<https://adafru.it/cFs>

Here's the sketch:

```

/*****
SoftServo sketch for Adafruit Trinket. Increments values to change position on
the servo
(0 = zero degrees, full = 180 degrees)

Required library is the Adafruit_SoftServo library
available at https://github.com/adafruit/Adafruit_SoftServo
The standard Arduino IDE servo library will not work with 8 bit
AVR microcontrollers like Trinket and Gemma due to differences
in available timer hardware and programming. We simply refresh
by piggy-backing on the timer0 millis() counter

Required hardware includes an Adafruit Trinket microcontroller
a servo motor, and a potentiometer (nominally 1Kohm to 100Kohm

As written, this is specifically for the Trinket although it should
be Gemma or other boards (Arduino Uno, etc.) with proper pin mappings

```

```

Trinket:      USB+  Gnd  Pin #0
Connection:  Servo+ -   Servo1

*****/

#include <Adafruit_SoftServo.h>; // SoftwareServo (works on non PWM pins)

#define SERV01PIN 0 // Servo control line (orange) on Trinket Pin #0

int skipCount = 100; //These are variables for the random servo skip
int skip = 0;

float moveAmount = 0.5; // change this value to change speed of servo
float servoPos = 0; // variable for servo position

Adafruit_SoftServo myServo1; //create servo object

void setup() {
  // Set up the interrupt that will refresh the servo for us automatically
  OCR0A = 0xAF; // any number is OK
  TIMSK |= _BV(OCIE0A); // Turn on the compare interrupt (below!)

  myServo1.attach(SERV01PIN); // Attach the servo to pin 0 on Trinket
  myServo1.write(90); // Tell servo to go to position per quirk
  delay(15); // Wait 15ms for the servo to reach the position
}

void loop() {

  skip = skip + 1; //counter for when the servo will a random
  jump
  if (skip == skipCount) {
    servoPos = servoPos + random (-15,15);
    if (servoPos < 1) servoPos=1; //this stops the servo from jumping
  outside its range
    if (servoPos > 119) servoPos=119; //the servo goes crazy if this happens
    skip=0;
    skipCount=random (100,300); //set the next interval for the servo to
  skip
  }

  myServo1.write(servoPos); // tell servo to go to position
  servoPos = servoPos + moveAmount; // increment servo position (value between
  0 and 180)
  if (servoPos == 0 || servoPos == 120){
    moveAmount = -moveAmount; //reverse incrementer at bounds
  }
  delay(15); // waits 15ms for the servo to reach the
  position
}

// We'll take advantage of the built in millis() timer that goes off
// to keep track of time, and refresh the servo every 20 milliseconds
// The SIGNAL(TIMER0_COMPA_vect) function is the interrupt that will be
// Called by the microcontroller every 2 milliseconds
volatile uint8_t counter = 0;
SIGNAL(TIMER0_COMPA_vect) {
  // this gets called every 2 milliseconds
  counter += 2;
  // every 20 milliseconds, refresh the servos!
  if (counter >= 20) {
    counter = 0;
    myServo1.refresh();
  }
}
}

```

Now the Neopixels. There's a are three things happening in the sketch:

- D0 is driving the laser/LED which is usually on. But at random intervals will flicker.
- At a set interval (in this case 3 seconds) one of the three pixels is selected at random and set to a random colour within an array. This uses the millis() timer.
- The pixel ring blinks a randomly selected pixel a random duration, picking a random colour from an array.

Yes that's a lot of use of the random() function.

You'll need the NeoPixel library of course.

**Download the Adafruit NeoPixel
library here.**

<https://adafru.it/aZU>

Here's the sketch to upload to the second Gemma or Trinket:

```
#include <Adafruit_NeoPixel.h>

#define PIN 1

// Parameter 1 = number of pixels in strip
// Parameter 2 = 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 pixels = Adafruit_NeoPixel(15, PIN, NEO_GRB);

uint32_t colorArray[3] = {0x10002C, 0x001000, 0x000010}; //the colours to use on
the pixel ring
uint32_t colorArray2[4] = {0x500000, 0x005000, 0x000050, 0x505000}; //the colours
to use for the neopixels

int howLongToWait = 3000;           // Wait this many millis()
int lastTimeItHappened = 0;        // The clock time in millis()
int howLongItsBeen;                // A calculated value

int led = 0;
int flkr = 0;

void setup() {
  pixels.begin();
  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH);
}

void loop() {
  uint8_t i;
  uint16_t d;

  //pixel ring is on pixels 4-15
  i = random(3,15); //select a random pixel on the ring
```

```

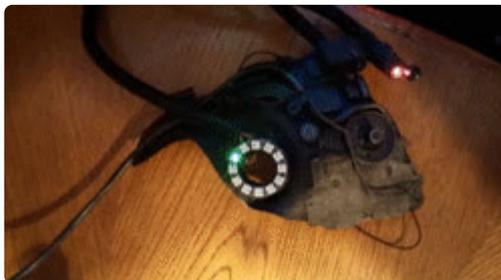
d = random(5, 250); //light it for a random duration
pixels.setPixelColor(i, colorArray[random(3)]);
pixels.show();
delay(d);
pixels.setPixelColor(i, 0);

//add some random blinks to the Laser Diode
flkr = flkr + 1;
if ( flkr >= 80) { //arbitrary number raise/lower to adjust the frquency of
blinks
  digitalWrite(led, LOW);
  delay (50);
  digitalWrite(led, HIGH);
  delay (25);
  digitalWrite(led, LOW);
  delay (50);
  digitalWrite(led, HIGH);
  flkr = 0;
}

//millis timer code to cycle random colours to pixels 1-3
howLongItsBeen = millis() - lastTimeItHappened;
if ( howLongItsBeen >= howLongToWait ) {
  pixels.setPixelColor(random(0, 3), colorArray2[random(4)]);
  pixels.show();
  // do it (again)
  lastTimeItHappened = millis();
}
}

```

Test everything out, modify code if you want to alter the timing or behaviour. If everything checks out you're ready to put it all together.



Assembly

Time to put it all together!

The wiring for the costume was assembled in three sections that were hooked together with [3-pin JST SM Plugs \(http://adafru.it/1663\)](http://adafru.it/1663). This makes the costume easier to put on or remove and for easier storage. The three sections were assigned as follows:

- Left arm and wrist guards tied together to for a one large gauntlet. On the forarm is the battery pack and power switches along with the lighting control Gemma. The upper arm has the first neopixel.
- Chest piece with the next two neopixels. Also the servo control Trinket/Gemma.

- The mask, with the 12-pixel ring over the eye, to complete the neopixel chain. At the side of the head goes the servo and laser/LED.

Start at the gauntlet and work your way up. Run all your wiring through corrugated split loom to conceal it all and at the same time give you all your Borg-like hoses. Add more empty split loom if you think you need more hoses.

Using a small project box or container of choice, install the battery pack and power switches. The battery's JST connect is brought outboard before looped back to a receptacle to allow easy connection to a charger (see below). Use and hook and loop tape to mount it to the forearm.



The lighting control Gemma can be mounted on the arm using either some hot glue or hook and loop to the arm. Solder and run some wiring to the first pixel and mount it as well. Terminate the output of the pixel with a 3-pin plug. Glue down the leads around the pixels for strain relief.

Run the the power for the other Gemma along with the wiring for the laser/LED along some split loom as well.



Mount your next two pixels on the chest piece and solder the wiring. Also find a good spot for the servo control Gemma/Trinket and mount it as well. Good strain relief is

important or you will have breaks so don't ignore it. Solder the servo wiring out to connectors, as well as the neopixel chain.

Finally run the rest of your leads to the mask. Make sure they're long enough to allow the wearer a full range of motion. Connect the neopixel ring to finish the chain, solder and glue into place. Connect the laser/LED and tape it to the servo arm. Find a spot for the servo to mount and tape, glue and or epoxy it into place.

Plug everything and try it out!





As for wearing it: black boots, leggings and turtle-neck. Then put white makeup on any areas that will be exposed. More skilled makeup artists could get into more detail but white alone will still get good results.

Armour up starting with the chest piece, then gauntlet and finally mask. Hook up all your connectors and switch on. You are one with the collective. Start assimilating.



A couple other ideas to try:

- Add a Waveshield voice changer or to play other sound effects.
- I did attempt to run it off a single Trinket Pro and rely more on the millis() timer rather than delay(). Because of time constraints I stuck with the two controllers as I wasn't getting the results I wanted, having difficulty getting everything to play nice together.