Zelda: Breath of the Wild – 3D Printed Guardian Sword with NeoPixel LEDs

Created by Ruiz Brothers

https://learn.adafruit.com/breath-of-the-wild-guardian-sword-led-3d-printed

Last updated on 2023-11-18 02:25:47 PM EST
# Table of Contents

## Overview
- 3D Printing & NeoPixel LEDs
- Where to find the Guardian Sword++ in the Breath of the Wild
- Parts
- Tools & Supplies
- Machine Screws

## Circuit Diagram
- Wired Connections
- Battery Power

## Arduino Code
- Guides
- Getting Code Onto Trinket
- Install Adafruit NeoPixel Library
- Uploading Code to Board
- Connect USB Data Cable to Trinket

## CircuitPython Code
- Trinket M0 Setup for CircuitPython
- Installing NeoPixel Library
- Saving CircuitPython Code

## 3D Printing
- Download and 3D Print
- Edit Design
- 3D Printed Parts
- LED Diffusion
- Slice Settings
- Thin Wall behavior
- Perfect First Layer
- Part Orientation
- Adding Supports
- Support Removal

## Assemble
- Prepare NeoPixel Strip
- Removing silicone ends
- Solder to strips together
- Prepare wires
- Slide Switch
- Prepare boards
- Ground, Data and Slide switch
- Slide switch wires
- Mounting the circuit
- Adding the NeoPixel strip
- Join Blades
- Finishing touches
- The Guardian Sword++
Overview

3D Printing & NeoPixel LEDs

This is the Guardian Sword from Zelda, Breath of Fresh Air. It’s fully 3D Printed and has a ton of NeoPixel LEDs. This thing is super bright and looks so awesome!

Pedro designed the blade in Autodesk Fusion 360 using reference images from the game. He designed it to be 3D printed in several pieces that will be connected together using screws.

The parts were 3D printed on a Type A Machines Series 1 Pro 3D printer. It’s got a massive build volume of 12 by 12 by 12 inches, so it’s great for making huge parts.

We used translucent PLA filament because it’s really easy to use and will let the NeoPixel LEDs shine through.
The sword is pretty massive, it’s about 34in in length and has over 90 NeoPixels.

Once I got it in the game, the first thing I thought was, wow this is so cool, I have to make it!

**Where to find the Guardian Sword++ in the Breath of the Wild**

So this guardian sword can be found in the Chaas Qeta shrine located in the Tenoko Island. It’s yielded by a guardian scout, so you’ll have to battle to get it. It has a decent attack rating of 40 points, but like most weapons in the game it won’t last long. So this one will last longer than the one in the game.

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

You'll need the following parts to build this project.

- Trinket M0 (or 5V Adafruit Trinket) (M0 is recommended)
- 2 Meter Adafruit skinny mini NeoPixel Strip
- Lipo Backpack
- 2200mAh 3.7v Cylindrical Lithium Ion Polymer Battery
- Slide Switch (http://adafruit.it/805)

Tools & Supplies

You'll also need the following tools and supplies to complete this project.

- 30 AWG Silicone wires
- Soldering Iron & Solder
- Tweezers Rhino SW-11 (http://adafruit.it/3096)
- Wire Strippers (http://adafruit.it/527)
- Diagonal Flush Snips
- Helping Third Hands (http://adafruit.it/291)
- Flat Pliers (http://adafruit.it/1368)
- 3D Printer (or 3D Printing Service)
- Translucent PLA filament
- NITTO Tape
Machine Screws

The parts are held together with screws to make it sturdy and easy to assemble.

Blade sides M2.5 x .5 x 8MM screws or #4-40-3/8.

After the blades are connect the two halves are combined using twenty M2 x .4 x 12MM screws or #2-56x1/2 screws.

The Trinket is mounted with M2 x 4mm screws.

The Lipo Backpack uses one M2.5 x 5mm screw.
Wired Connections

The circuit diagram above shows how the components will be wired together. This won't be 100% exact in the actual circuit but it's a very close approximation.

Lipo Backpack

- 5V on the Lipo Backpack to USB on the Trinket
- GND on the Lipo Backpack to GND Trinket
- BAT on the Lipo Backpack to BAT on the Trinket
- The two through holes next to 5V on the Lipo Backpack to the two pins on the slide switch

NeoPixel Strip

- +5V on the Strip is shared with BAT on the Trinket
- GND on the Strip is shared with GND on the Trinket
- D1 on the Strip to pin #4 on the Trinket

Battery Power

The circuit will be powered by a 3.7V 2200mAh Lithium ion battery via JST connection. The battery plugs into the JST connection on the Lipo Backpack.

Arduino Code

The Arduino code presented below works equally well on all versions of Trinket Mini 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!

Guides

If you haven’t already worked through the Adafruit Trinket M0 or Introducing Trinket guide, do that first.
Getting Code Onto Trinket

Before we start disassembling or building the circuit, it's a good idea to get code uploaded to the micro-controller first. If you don't write / understand code, don't worry! You don't need to be a programmer to be able to upload prewritten code :-)

We'll walk you through the whole process.

First, visit the Trinket tutorial pages above. Follow the instructions to download & setup the Arduino IDE and install drivers.

Make sure you are able to get sketches compiled and uploaded, especially the blink example in the tutorial. Once you are comfortable with using the Trinket, you can continue!

Install Adafruit NeoPixel Library

Next, we need to add support for NeoPixels.

Visit the Adafruit NeoPixel tutorial () to install the NeoPixel library!

Uploading Code to Board

Now that we have the Adafruit boards & NeoPixel library installed, we can get our code ready to upload onto the board. Select all of the code listed below in the black box and copy it to your clip board. Then, in Arduino IDE, paste it in the sketch window (making sure to overwrite anything currently there). Next, goto the Tools menu > Board and select either Adafruit Trinket M0 or Adafruit Trinket (if you're using the 3V Adafruit Trinket version use Trinket 8Mhz. If you're using the 5V Trinket, select Trinket 16Mhz).

Now you can click on the "check mark" icon to verify the code. If it's all good, we can continue to upload the code to the board.
Connect USB Data Cable to Trinket

Be sure to use a micro USB cable that can transfer data - A USB cable that ONLY charges devices will simply not work. Plug it into the microUSB port on the Adafruit Trinket board and the USB port on your computer (try to avoid connecting to a USB hub).

Follow the upload procedure documented in the Getting Started guides at the top of this page to upload. The Arduino IDE will notify you if the upload is successful and completed.

```cpp
// SPDX-FileCopyrightText: 2018 Mikey Sklar for Adafruit Industries
//
// SPDX-License-Identifier: MIT

#include <Adafruit_NeoPixel.h>
#if defined (__AVR__)
    #include <avr/power.h>
#endif

#define PIN        4
#define APIXELS    14 // number of first orange pixels
#define BPIXELS    84 // number of blue pixels
#define CPIXELS    93 // second orange pixels

// When we setup the NeoPixel library, we tell it how many pixels, and which pin to use to send signals.
// Note that for older NeoPixel strips you might need to change the third parameter- see the strandtest
// example for more information on possible values.
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(93, PIN, NEO_GRB + NEO_KHZ800);

int delayval = 10; // delay for half a second

void setup() {
    // This is for Trinket 5V 16MHz, you can remove these three lines if you are not using a Trinket
    #if defined (__AVR_ATtiny85__)
        if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
    #endif
    // End of trinket special code

    pixels.begin(); // This initializes the NeoPixel library.
}

void loop() {
    // For the first 14 pixels, make them orange, starting from pixel number 0.
    for(int i=0;i<APIXELS;i++){
        pixels.setPixelColor(i, pixels.Color(255,50,0)); // Set Pixels to Orange Color
        pixels.show(); // This sends the updated pixel color to the hardware.
        delay(delayval); // Delay for a period of time (in milliseconds).
    }

    // Fill up 84 pixels with blue, starting with pixel number 14.
    for(int i=14;i<BPIXELS;i++){
```
pixels.setPixelColor(i, pixels.Color(0,250,200)); // Set Pixels to Blue Color
pixels.show(); // This sends the updated pixel color to the hardware.
delay(delayval); // Delay for a period of time (in milliseconds).
}

// Fill up 9 pixels with orange, starting from pixel number 84.
for(int i=84;i<CPIXELS;i++){
  pixels.setPixelColor(i, pixels.Color(250,50,0)); // Set Pixels to Orange Color
  pixels.show(); // This sends the updated pixel color to the hardware.
  delay(delayval); // Delay for a period of time (in milliseconds).
}
}

CircuitPython Code

Trinket M0 Setup for CircuitPython

Now it's time to set up the Trinket M0 for use with CircuitPython() and our code.

If you haven't already, follow this guide() to preparing the Trinket M0, including updating it with the latest version of CircuitPython.

After prepping the Trinket M0 to run CircuitPython we'll also need to add a NeoPixel library. This guide tells you how(), as well as providing a good primer on using NeoPixels on the Trinket M0 with CircuitPython.

Installing NeoPixel Library

Download the latest adafruit-circuitpython-bundle-xxxx.zip (or newer) from the releases directory() and then unzip it somewhere easy to find, such as your desktop. Then, copy the neopixel.mpy file to your CIRCUITPY's lib directory on the Trinket M0.

Finally, if you had a neopixel.py file that was already in that same lib directory you can delete it (the mpy is a compressed version)

Saving CircuitPython Code

Once you've got things working, you can edit the code.py file on the Trinket M0 to adjust what it actually does. No need for a software IDE, complaining tools, or flashing the chip -- when you code with CircuitPython all you need is a text editor. Edit the code, save it to the Trinket M0, and it immediately runs!
import time
import board
import neopixel

pin = board.D4  # DIGITAL IO pin for NeoPixel OUTPUT from GEMMA
pixel_count = 93  # number of neopixels
delayval = .01  # 10 ms delay

APIXELS = 14  # number of first orange pixels
BPIXELS = 84  # number of blue pixels
CPIXELS = 93  # second orange pixels

# initialize neopixels
pixels = neopixel.NeoPixel(pin, pixel_count, brightness=1, auto_write=False)

while True:
    # For the first 14 pixels, make them orange, starting from pixel number 0.
    for i in range(0, APIXELS):
        pixels[i] = (255, 50, 0)  # Set Pixels to Orange Color
        pixels.write()  # This sends the updated pixel color to the hardware.
        time.sleep(delayval)  # Delay for a period of time (in milliseconds).

    # Fill up 84 pixels with blue, starting with pixel number 14.
    for i in range(APIXELS, BPIXELS):
        pixels[i] = (0, 250, 200)  # Set Pixels to Blue Color
        pixels.write()  # This sends the updated pixel color to the hardware.
        time.sleep(delayval)  # Delay for a period of time (in milliseconds).

    # Fill up 9 pixels with orange, starting from pixel number 84.
    for i in range(BPIXELS, CPIXELS):
        pixels[i] = (250, 50, 0)  # Set Pixels to Orange Color
        pixels.write()  # This sends the updated pixel color to the hardware.
        time.sleep(delayval)  # Delay for a period of time (in milliseconds).
3D Printing

Download and 3D Print

You can 3D print the parts using Translucent PLA for the blades and black for the handle details on desktop FDM style 3D printers. The parts can be downloaded using the link below. If you don't have a 3D printer, the files are free to download so can send them to a 3D printing service.

The parts were 3D printed using the Type A Machines Series 1 Pro.

- Download STLs from Thingiverse
- Download STLs from Youmagine
- Download STLs from Pinshape

Edit Design

The design is modeled in Autodesk Fusion 360 and available to edit. You can adjust the pieces to print by moving or adding construction planes to cut up the parts. You can modify the sketches or adjust features in the parametric timeline.

- Edit Design
3D Printed Parts
Blades
Our design a separated into six parts to fit on a 12 inch x 12 inch bed. The parts connect together using M2.5 x .5 x 8MM screws (or #4-40-3/8 screws).

Each half is then combined together and screwed down using twenty M2 x .4 x 12MM () screws or #2-56x1/2 screws.

Handle Details
The detail overlays for the handle are glued on top of the left and right sides with a couple drops of E6000 adhesive.

Cylindrical Gem
This goes over the hole in the handle and it just press fits into place.

LED Diffusion
We recommend using blue painters tape (), PEI or PrintinZ plates/skins () to better diffuse the blades. Printing on a glass bed will give the parts a shiny surface and doesn't diffuse the LEDs very well.
The sword on the top was printed on blue painters tape. You can see fewer hot spots. The matted surface evenly distributes the LEDs giving a much better effect.

The sword on the bottom was printed on a glass bed. We can see hot spots for each LED because it the glass bed makes the surface more translucent. You can always sand the part down, but a printing surface will allow you to diffuse the light without additional post processing.

Slice Settings

Depending on your 3D printer, you may need to adjust the slice settings. We printed all of the parts on a Type A Machines Series Pro 1. These parts were sliced with Simplify3D.

- Nozzle: 0.34mm
- Extrusion Multiplier: 1.0
- Extrusion Width: 0.38mm
- Layer Height: 0.25mm
- Infill: 10%
- Nozzle Temperature: 230c
- Heated Glass Bed: 40c
- Print Speed: 80mm/s
- Vertical Lift / Retraction: 2mm
Thin Wall behavior

Make sure to adjust the nozzle diameter and extrusion width so we have three perimeters around the standoff for the screws.

We used a .4mm nozzle but had to set our nozzle diameter to .34 and the extrusion width to .38mm in order to produce a tool path with perimeters between the walls. This is especially important for the standoffs.

Perfect First Layer

Heat the bed to around 40c (optional) and make sure that it’s absolutely leveled across the entire bed. In order to properly 3D print these large parts. It’s really important to have a perfect first layer, so make sure the bed is cleaned to ensure the filament adheres to the bed.

While leveling the bed we need to find the sweet spot so that the first layer isn’t not too squished or too loose. If the bed’s too close to the nozzle, the material may buckle. If it’s too high the bottom won’t have an even surface.

Part Orientation

The main parts actually exceed the build volume on the Type A Machines Series 1 Pro. So, you’ll have to orient them at an angle so they’re positioned diagonally across the bed. This is a great way to get more build volume from your 3D printer.
Adding Supports

Use 8mm pillars to support the four parts on the part that connects the blade to the handle. Add three pillars underneath each connection to assist the overhangs and bridging the parts.
Support Removal

Use flat pliers () to help remove the support material. We can first insert the pliers between support lines and then grip and apply a bit of force to remove.
Assemble

Prepare NeoPixel Strip

NeoPixel strips are sold by the meter. The medium density strips have 60 LEDs per meter. We recommend ordering a 2 meter strip so it's long enough to fit the entire length of the sword. The strip starts at the end of the handle and up from the base of the blade to the tip and then loops back down towards the handle.

The entire length of the strips will need to be 151mm (59in) for a total of 93 NeoPixels.

We can also solder multiple strips together to fit the whole length of the sword.
Removing silicone ends

The included wires on the NeoPixel strips are too big to fit inside the sword, so we'll need to unsolder them and replace them with flexible silicone wires.

We can use scissors or flush cutters to remove the ends of the sheathing by inserting the cutters tips between the end and the rest of the sheathing.

Solder to strips together

With the silicone ends removed we can now join multiple strips together if needed. We can add both strips to third helping hands to make soldering easier.

Make sure the arrow labels on the strips are pointing in the same direction so the data flows in a continuous direction.

Here’s how to connect multiple strips together. Tin the pads on both of the strips and carefully apply solder. Then, slide the solder onto each pad.
Prepare wires

Now we can move on to measuring and tinning the wires for the NeoPixel strips.

The NeoPixel strip wires will need to be 141mm long to reach the Trinket when its mounted to the handle, over the battery and walls around the handle.

After cutting the wires to length, tin both end of the wires. Apply a small amount of solder to the pads on the NeoPixel strips and apply heat to connect the wires to the pads.

Slide Switch
A slide switch will make it easy to power the circuit on and off. Prepare the switch by removing one of the pins on the left or right side of the switch.

Don't remove the middle pin!

Measure two wires so they're 170mm in length. These need to reach the slide switch and the Lipo Backpack.
Prepare boards
Now we can prep the Trinket and Lipo Backpack by securing them to a pair of third helping hand. This will make it easier to solder.

Lipo Backpack
We need to enable the on/off function in order to use the switch to power the Trinket. You'll need to cut the trace near the 5V pin. You can use a hobby knife to cut the trace as shown in the picture.

Tin and solder Trinket and Backpack
Now we can tin the 5V, G and BAT pins on the Lipo Backpack. Connect the 5V on the Lipo Backpack to USB pin on the Trinket.

Power wires
The NeoPixel 5V wire and Lipo Backpack wires for BAT will both need to be soldered to the BAT pin on the Trinket. The 30 gauge silicone wires are thin enough to easily insert both wires to the BAT pin on the Trinket. With them both in place, apply solder to the pin.
Ground, Data and Slide switch

Next up we'll need to share the ground pin on the Trinket with the NeoPixel strip and the Lipo Backpack. Insert both of the ground wires inside ground pin on the Trinket and apply solder.

Connect the data wire from the NeoPixel to GPIO pin #4 on the Trinket. We can tin the pin and then apply heat while inserting the wire.

Slide switch wires
Finally, we can connect the slide switch wires to the two pins on the Lipo Backpack. Polarity doesn't matter, so you can connect the wires into any of the two pins.
Mounting the circuit

Secure Trinket
Now we can mount all of the boards to the standoffs on the handle. The Trinket needs M2 x 4mm machine screws. Orientate the Trinket so the USB port is facing the cutout in handle. Make sure to trim any excess material around the port opening to easily fit a USB cable.

Lipo Backpack
The Lipo Backpack needs a single M2.5 x 5mm machine screw. Position the board with the JST connector facing the wall.

Battery
Now we can plug in the battery wire into the Lipo Backpack. Route the battery wire around the wall cut away and seat the battery in between the two walls.

Slide Switch
Route the wires for the slide switch around the wall cut away and use the flat side of a pair of tweezers to push the switch into the opening on the upper part of the handle. Make sure to remove any excess material around the opening.
Adding the NeoPixel strip

Add four pieces of Nitto tape to the middle part of the upper blade, like shown in the picture.

Next up we'll add the NeoPixel strip with the LEDs facing outward. The sides of the strips adhere well to the Nitto tape pieces. The NeoPixel strip should follow the contour of the blade.

Join Blades

With the components now mounted to the parts, we can move on to joining the two halves together. The two are held together with twenty M2 x 12mm screws. That's a lot!
Finishing touches

In the last couple of steps we'll finish up by adding the detail overlays to the handle and blade.

The "traces" on the blade are pressed into the cavities. They just snap into place. If the fittings are too loose, we can use small drops of E6000 adhesive to glue the parts in place.

The handle overlays will need E6000. Make sure to line up the screws holes on the overlays with the holes on the handle. Let the E6000 fully dry for about an hour.

While the glue dries, we can add the gem part to the handle by press fitting it into the holes.
The Guardian Sword++

And that's it! Now we have our very own guardian sword. The thing we like most about this design is that it's really durable, so you can swing it around and not worry about it falling apart. Using the screws to secure all the parts together was really great idea and it's easy to take part if we ever need to.