Ray Gun Blaster
Created by Ruiz Brothers

https://learn.adafruit.com/ray-gun-blaster

Last updated on 2023-08-29 02:40:09 PM EDT
# Table of Contents

## Overview
- Sci-fi Inspired Alien Ray Gun - Pew Pew!
- Prerequisite Guides
- Project Expectations
- Parts
- Tools & Supplies

## Circuit Diagram
- Prototype First!
- Wire Connections

## 3D Printing
- PLA or ABS
- 3D Parts List
- Customize Design

## Finishing
- Removing Support Material
- Removal Technique
- Painting
- Weathering Techniques
- Apply Adhesives
- Half Parts Joined

## Software
- Installing Arduino
- Uploading Sketches
- Muzzle Flash Blast
- Linear Interpolation
- Trigger Spinny Animation

## Sounds
- Sound Effects
- Making FX with Samples
- Uploading Samples to Audio FX Sound Board

## Wiring
- Pro Trinket and Audio FX Sound Board
- Wire Pro Trinket and Audio FX Sound Board
- Install Pro Trinket & Audio FX sound board
- Mounting Screws
- Mount Pro Trinket & Audio FX Sound Board
- Setup Pro Trinket LiPoly Backpack
- Mount Pro Trinket LiPoly Backpack
- Wire Pro Trinket LiPoly Backpack
- Wire Slide Switch
- Wire Slide Switch to LiPoly Backpack
- Wire Amp to Audio FX Sound Board
- Wired Pro Trinket + Audio FX + Amp
- Wire Speaker to Audio FX Sound Board
- Wire Push Button
- Connect Push Button to Pro Trinket + Audio FX Sound Board
• Wire NeoPixel Ring
• Wire Laser Diode
• Mount NeoPixel Ring Diffuser to Barrel
• Connected Components
• Preassembly Test

Assembly

• Assembling Handle
• Closing Handle Enclosure
• Handle Assembly Checkpoint
• Adding Rubberband to Trigger
• Mount Trigger
• Mount Slide Switch + Push Button
• Test Trigger Spring
• Assembling Trigger box
• Secure Handle to Trigger box
• Assembled Trigger + Handle
• Assembling Body
• Add Diffusers
• Close Body
• Secure Body Parts
• Install Sights
• Install Battery
• Mount Speaker
• Install Speaker Mount to Body
• Final Check Point
Overview

Sci-fi Inspired Alien Ray Gun - Pew Pew!

In this project, we're making a 3D printed, sci-fi inspired ray gun - with LEDs, fully functioning trigger and sound effects!

This blaster was designed in CAD and split into pieces that can be 3D printed. The parts are made to house the electronics and secured together with screws and adhesives.

You'll need access to a 3D printer, a handful of components and some building experience. This project requires time, patience and passion to make an intricate build.

Prerequisite Guides

Walk through these guides before tackling this project. Check out the following guides to get familiar with the Pro Trinket and Audio FX sound board.

- [Introducing Pro Trinket ()](#)
- [Adafruit Audio FX Soundboard ()](#)
- [NeoPixel Uberguide ()](#)
Project Expectations

This project is rather challenging and intended for experienced makers. It requires a good amount of experience soldering and assembling small circuits. There’s 20 3D printed parts and about 20 different wired connections. This build requires fitting wires and components into small spaces that can be damaged if not carefully handled. This is a great project for experienced and ambitious makers.
Parts

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

- Audio FX Sound Board (http://adafru.it/2133)
- Pro Trinket 5V/3V ()
- PAM8302 Mono 2.5W Class D Audio Amplifier ()
- Pro Trinket Lilon/LiPoly Backpack Add-On ()
- NeoPixel Ring 12 x ()
- Laser Diode (http://adafru.it/1054)
- 500mAh Lithium Ion Polymer Battery ()
- Mini Metal Speaker ()
- 6mm Slim Tactile Switch Buttons ()
- Slide Switch ()

Tools & Supplies

The following tools and supplies were used in the development of this project.

- 3D Printer ()
- Filament ()
- Soldering Iron ()
- Solder ()
- Panavise Jr / Helping Third Hand (http://adafru.it/151)
- Wire Strippers ()
- Flush Diagonal Cutters ()
- Flat Pliers (http://adafru.it/1368)
• **#4-40 3/8in Phillips machine screws and nuts**
• Adhesives (E6000, superglue, etc.)
• Fun-tac (mold putty)
• Heavy gritted sandpaper (optional)
• Paint (optional)

---

**Circuit Diagram**

---

**Prototype First!**

Prototype the circuit on a breadboard before soldering components. It's good idea to test out the components and ensure they're functioning before enclosing them. Get a working prototype and then continue with the build.
Wire Connections

Follow the diagram to see how the components are connected. The Pro Trinket and Audio FX sound board are connected together through the JST pads on the bottom of the PCBs. Note the push button is connected to both the Audio fx sound board and the Pro Trinket.

Size and position of the components aren't exact but a representation of how the components will be wired.
3D Printing

<table>
<thead>
<tr>
<th>Pro Trinket</th>
<th>Audio FX Sound</th>
<th>Amp PAM8302A</th>
</tr>
</thead>
<tbody>
<tr>
<td>#4 - Push Button</td>
<td>#10 - Push button</td>
<td>+pos - speaker</td>
</tr>
<tr>
<td>GND - Push Button</td>
<td>GND - Push button</td>
<td>-neg - speaker</td>
</tr>
<tr>
<td>#3 - DIN (NeoPixel)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GND - GND (NeoPixel Ring)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bat - PWR (NeoPixel Ring)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5V - Positive(laser diode)</td>
<td>VIN - VIN (Amp)</td>
<td>+pos - speaker</td>
</tr>
<tr>
<td>GND - Negative(laser diode)</td>
<td>Gnd - Gnd (Amp)</td>
<td>-neg - speaker</td>
</tr>
<tr>
<td>BUS - 5V (lipo backpack)</td>
<td>L - A+ (Amp)</td>
<td></td>
</tr>
<tr>
<td>G - G (lipo backpack)</td>
<td>Gnd - A- (Amp)</td>
<td></td>
</tr>
<tr>
<td>Bat - Bat (lipo backpack)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JST(-) - JST(-) (audio fx)</td>
<td>JST(+) - JST(+) (audio fx)</td>
<td></td>
</tr>
</tbody>
</table>

PLA or ABS

Most of the parts can be printed in either PLA or ABS. Some of the parts require support material to print properly; these are listed in the table below. A set of parts are printed in transparent material to diffuse the NeoPixel LED rings. We've found removing support material with ABS is easier than PLA.
The parts below are grouped and separated. The first batch require support material and can be printed in any color of choice. The second batch doesn't require support and can print "as-is". The third batch should be printed in transparent material, whether that it'd be PLA, Nylon or PET+.

<table>
<thead>
<tr>
<th>Parts</th>
<th>Temperature</th>
<th>Shells</th>
<th>Infill</th>
<th>Speeds</th>
<th>Support Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>body-left, body-right, handle-bottom, handle-top</td>
<td>240c(ext) / 190c(bed)</td>
<td>2</td>
<td>10%</td>
<td>90/120</td>
<td>On</td>
</tr>
<tr>
<td>body-ammo-left, body-ammo-right, body-sight-left, body-sight-right, body-speaker-cap</td>
<td>240c(ext) / 190c(bed)</td>
<td>2</td>
<td>10%</td>
<td>90/120</td>
<td>Off</td>
</tr>
</tbody>
</table>
Customize Design

The original solids are available to download and modify using Autodesk 123D Design. The project file includes modeled electronic components for reusing in other 3D printed projects.

Modify Design
Finishing

Removing Support Material
The four parts that were printed with support material will need to be cleaned up. We recommend using a pair of flat pilers to help remove the material.
Removal Technique

A simple technique is to pinch, grip, twist and pull. PLA material is rather tough to remove so you'll have to be careful not to break the part but apply enough force to remove supports.

Painting

You can optionally paint the parts with spray paint or acyrlic based paints. We used a chrome colored spray paint for the parts in this project. Make sure to take proper precautions when using spray paint. Be sure to do it outdoors or in a well ventilated working area. Check out instructables() for some tips on properly applying spray paint.
Weathering Techniques

Check out instructables () for some weathering tips. We used heavy gritted sandpaper on the parts after letting the paint dry for a few hours. About 15 minutes of sanding achieves a weathered look.
Apply Adhesives
These parts will require adhesives to join together. A total of 6 parts are halves that will be combined to make 3 parts.
Half Parts Joined

The barrel, neopixel ring disk diffuser and eye sight are joined together with adhesives. Keep the halves held together with either rubberbands or by holding them while the adhesives bond the parts together. Ensure they're fully dried before using them.

Software

Installing Arduino

You'll need to customize some settings in the Arduino IDE to get configured to compile code onto the Adafruit Pro Trinket. Be sure to check out the [Introduction to Pro Trinket](https://learn.adafruit.com) guide for setting that up and than download and install the NeoPixel Library.
Uploading Sketches

With the Adafruit Arduino IDE and NeoPixel library installed, create a new sketch. Paste the desired code into the text editor. Select Pro Trinket from Tools > Board. Then USBtinyISP from Tool > Programmer. Proceede by plugging a micro USB cable from your computer to the Pro Trinket. Watch for the red LED to start blinking and click upload to compile the sketch onto the Pro Trinket. If everything is good, Arduino will prompt with you a successful message in the console window.

Muzzle Flash Blast

The code below was written by Tony DiCola (). The basic idea is to pass it a strip of pixels, starting RGB color, ending RGB color, and total duration in milliseconds. Then it will loop through and animate all the pixels going from the start to end color over that period of time. Just one call to go from white-ish to black works well for a simple flash, but you can get crazy and make multiple calls to flash between different colors and fade out.

Linear Interpolation

Tony put together a color demo to show linear interpolation can be used to generate color gradients. The start and end colors dynamically change as you drag the slider, it shows the intermediate colors and equations that generate them.

```cpp
//Written by Tony DiCola for Adafruit Industries.
//Adafruit invests time and resources providing this open source code,
//please support Adafruit and open-source hardware by purchasing products
//from Adafruit!
//BSD license, all text above must be included in any redistribution.

#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN  3  // Pin connected to neo pixels
#define FIREPIN    4  // Fire button
#define PIXEL_COUNT 12  // Count of neo pixels

int buttonState = 0;

Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  strip.begin();
  strip.show();
  pinMode(FIREPIN, INPUT_PULLUP);
}
```
void loop() {
    uint8_t i;
    // Button switch
    buttonState = digitalRead(FIREPIN);

    // check if the pushbutton is pressed.
    // if it is, the buttonState is LOW:
    if (buttonState == LOW) {
        // Run It:
        // Nice flash and fade out over about 3/4 of a second:
        animate_gradient_fill(strip,
                               255, 255, 255,
                               255, 0, 0,
                               150);
        // Then flash from purple to nothing over a longer period.
        animate_gradient_fill(strip,
                               255, 0, 0,
                               20, 0, 0,
                               150);
        animate_gradient_fill(strip,
                               20, 0, 0,
                               0, 0, 0,
                               150);
    }
    else {
        strip.setPixelColor(i, 0, 0, 0); // Button not pressed, turn off pixels
        strip.show(); // Show no pixels
    }

    // More complex flash between two colors and fade out using multiple calls.
    // First flash from red to purple over a short period
    animate_gradient_fill(strip,          // NeoPixel strip/loop/etc. object.
                           0, 0, 0,       // Ending R,G,B color (0,0,0 for black or off).
                           500);          // Total duration of the animation in milliseconds.
    animate_gradient_fill(strip,
                           255, 0, 0,
                           128, 0, 255,
                           200);
    animate_gradient_fill(strip,
                           128, 0, 255,
                           0, 0, 0,
                           400);
    // Delay to see different animations.
    delay(1000);
}

// Linear interpolation of y value given min/max x, min/max y, and x position.
float lerp(float x, float x0, float x1, float y0, float y1) {
    // Clamp x within x0 and x1 bounds.
    x = x > x1 ? x1 : x;
    x = x < x0 ? x0 : x;
    // Calculate linear interpolation of y value.
    return y0 + (y1-y0)*((x-x0)/(x1-x0));
}

// Get the color of a pixel within a smooth gradient of two colors.
uint32_t color_gradient(uint8_t start_r, // Starting R,G,B color
                        uint8_t start_g,
                        uint8_t start_b,
                        uint8_t end_r,
                        uint8_t end_g,
                        uint8_t end_b)

```c
uint8_t end_r, // Ending R,G,B color
uint8_t end_g,
uint8_t end_b,
float pos)       // Position along gradient, should be a value 0 to 1.0
{
// Interpolate R,G,B values and return them as a color.
uint8_t red   = (uint8_t) lerp(pos, 0.0, 1.0, start_r, end_r);
uint8_t green = (uint8_t) lerp(pos, 0.0, 1.0, start_g, end_g);
uint8_t blue  = (uint8_t) lerp(pos, 0.0, 1.0, start_b, end_b);
return Adafruit_NeoPixel::Color(red, green, blue);
}
// Set all pixels to the specified color.
void fill_pixels(Adafruit_NeoPixel& pixels, uint32_t color)
{
for (int i=0; i < pixels.numPixels(); ++i) {
    pixels.setPixelColor(i, color);
}
strip.show();
}
void animate_gradient_fill(Adafruit_NeoPixel& pixels, // NeoPixel strip/loop/etc.
uint8_t start_r,           // Starting R,G,B color
uint8_t start_g,
uint8_t start_b,
uint8_t end_r,             // Ending R,G,B color
uint8_t end_g,
uint8_t end_b,
int duration_ms)           // Total duration of animation, in milliseconds
{
unsigned long start = millis();
// Display start color.
fill_pixels(pixels, Adafruit_NeoPixel::Color(start_r, start_g, start_b));
// Main animation loop.
unsigned long delta = millis() - start;
while (delta < duration_ms) {
    // Calculate how far along we are in the duration as a position 0...1.0
    float pos = (float)delta / (float)duration_ms;
    // Get the gradient color and fill all the pixels with it.
    uint32_t color = color_gradient(start_r, start_g, start_b, end_r, end_g, end_b,
    pos);
    fill_pixels(pixels, color);
    // Update delta and repeat.
    delta = millis() - start;
}
// Display end color.
fill_pixels(pixels, Adafruit_NeoPixel::Color(end_r, end_g, end_b));
```

Trigger Spinny Animation

The code below was originally written by Phillip Burgess () and used in the Kaleidoscope () project but is slightly modified to fire the animation when a push button is pressed.

// Original Kaleidoscope eyes code from Phillip Burgess.
// Modified to play spinning animation when push button is set to low
// Written by Philip Burgess for Adafruit Industries.
// Adafruit invests time and resources providing this open source code,
// please support Adafruit and open-source hardware by purchasing products
// from Adafruit!
```c
#include <Adafruit_NeoPixel.h>

#define PIN          0  //NeoPixel Pin
#define FIREPIN      2  // Fire button

int buttonState = 0; //Default button state

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(12, PIN);

uint8_t         offset = 0;         // Position of spinny eyes
uint32_t color  = 0x00ff5a;  // HEX color here
uint32_t prevTime;

void setup() {
  pixels.begin();
  prevTime = millis();
  pinMode(FIREPIN, INPUT_PULLUP); //Button enabler
}

void loop() {
  uint8_t  i;
  uint32_t t;

  buttonState = digitalRead(FIREPIN); //Trigger button state
  if (buttonState == LOW) { // If button pressed
    for(i=0; i<16; i++) { //Run the animation
      uint32_t c = 0; //Blank color
      if(((offset + i) & 7) < 2) c = color; // 4 pixels on...
      pixels.setPixelColor(i, c); //Set the color of pixels
    }
    pixels.show();
    offset++;
  }

  t = millis();
  if((t - prevTime) &gt; 10) {      // Every 10th of a second...
    for(i=0; i&lt;32; i++) pixels.setPixelColor(i, 0);
    prevTime = t;
  } else {
    pixels.setPixelColor(i, 0,0,0); //Button not pressed, turn off pixels
    pixels.show(); //Show no pixels
    delay(25);  //duration of spinniness
  }
}
```

### Sounds

#### Sound Effects

You can upload audio files to the audio FX sound board that are in OGG or WAV format. Please check the [Adafruit Audio FX guide](#) for a full breakdown on supported sample rates and recommended settings.
Making FX with Samples

Most DAW or video editing software come packaged with royalty free sound effects. In this project, I used a handful of electronic and firearm sounds from the Final Cut Pro X sound effects library. Layering them together to make an intricate sounding laser blast and mixing the audio levels to balance out the desired effect.

Download Sound Effects

Uploading Samples to Audio FX Sound Board

Adding audio the sound board is like adding files to a USB memory stick. Connect the fx board to your computer with a micro-USB cable. The fx board will load as "Adafruit" storage drive. There, you can drag-n-drop audio samples. Audio files must be named to a specific track in order to play properly. In this project, we connected the push button to pin #10 so our audio file name must contain T10. The trigger mode is also appended to the file name so the FX board knows what mode to play the sample.

Audio FX Soundboard Guide
Wiring

Pro Trinket and Audio FX Sound Board

To power the sound board, we’ll need to connect the positive and negative pads on the bottom JST to the JST pads on the Pro Trinket. You’ll need to secure the two boards using either helping third hands or a panavise jr.

Tin the pads on the JST. Measure and cut two pieces of 30AWG silicone-coated stranded wires to about 7cm in length. Strip and tin the two wires.
Wire Pro Trinket and Audio FX Sound Board

Solder one piece of wire from the (-) JST on the Pro Trinket to the (-) JST on Audio FX sound board. Solder the second piece of wire from the (+) JST on the Pro Trinket to the (+) JST on the Audio FX sound board.

Install Pro Trinket & Audio FX sound board

Place the components into the two handle parts. You'll know which handle part goes with the component by checking if the stand-offs line up with the mounting holes. Both components should have their micro-USB connectors facing the opening cutout near the bottom.

Mounting Screws

You'll need a batch of #4-40 3/8in Phillips machine screws to secure the components to the printed parts. I recommend fastening them to the components first.
to carve the screw threads into the mounting holes. The Pro Trinket was a bit difficult (smaller) to get machine screws into the mounting holes, so I used #4 flat Phillips screws (The sharp and pointy tipped kind).

Mount Pro Trinket & Audio FX Sound Board

Fasten #4-40 Phillips machine screws to secure the components to the handle parts. You can use single or two screws for each component.

Keep handle parts close together! Avoid accidentally breaking the wires by separating parts.

Setup Pro Trinket LiPoly Backpack

We need to setup the Pro Trinket LiPoly backpack in order to use it with a slide switch. Use a filing tool or x-acto knife to break the trace between the two pins designated for the slide switch.
Mount Pro Trinket LiPoly Backpack

Insert and fasten a #4-40 Phillips machine screw into the Pro Trinket LiPoly Backpack and mount it to the handle part.

Wire Pro Trinket LiPoly Backpack

Measure and cut 30AWG silicone-coated stranded wire long enough to connect the ProTrinket to the LiPoly backpack. Wire together BAT+ to BAT, G to G, and BUS to 5V.
Wire Slide Switch
Use a helping-third hand to secure the slide switch in place while soldering a wire to the terminals. Measure and cut two pieces of 30AWG silicone-coated stranded wire and solder them to the slide switch. Use heat shrink tubing to secure the connections.

Wire Slide Switch to LiPoly Backpack
Solder the two wires from the slide switch to the designated pins on the Pro Trinket LiPoly Backpack. Connect the JST cable from a 500mAh lithium polymer battery to the JST female connector on the LiPoly Backpack. Slide the switch to power it on. If everything’s good, you should see both components green LEDs appear.
Wire Amp to Audio FX Sound Board

Measure and cut wires that will connect the PAM8302 amp to the Audio FX sound board. Wire together A+ to L or R, GND to GND, Vin to Vin, and A- to GND. Strip and tin the wires before soldering to pin outs.

If the LEDs aren't lighting up, unmount components and check your connections for any shorts.

Wired Pro Trinket + Audio FX + Amp

Check point. Double check your wiring and cross reference the connections with the circuit diagram.
Wire Speaker to Audio FX Sound Board

Remove the wires that came with the mini metal speaker and solder 30AWG silicone-coated wire. Solder the (+) and (-) wires from the speaker to the (+) and (-) audio out pins on the PAM8302.
Secure the push button in place with a helping-third hand. Use flat pilers to straighten out the terminals on the push button. Measure and cut four pieces of 30AWG silicone-coated stranded wire. Strip and tin the tips of each wire. Solder two pieces to each terminal of the push button. Add shrink tubing to secure the connections.
Connect Push Button to Pro Trinket + Audio FX Sound Board

Solder one of the wires from the push button to the #4 GPIO on the Pro Trinket. Solder wire, from the second terminal of the push button to a GND pin on Pro Trinket. Solder another wire from the push button to #10 GPIO on the Audio FX sound board (should be the wire thats shared with #4 GPIO on the Pro Trinket). Solder the remaining wire to the GND on the Audio FX sound board (shared with GND wire connected to Pro Trinket.)
Wire NeoPixel Ring

Secure the NeoPixel ring to a panavise jr. Measure and cut three pieces of 30AWG silicone-coated stranded wire. Strip and tin the tips of each wire then solder wires to GND, Data Input, and 5V Power. Thread each wire through the holes in the NeoPixel mount and press the ring down to snap it into place.
Wire Laser Diode
Double the length of the wiring from the laser diode by soldering a pair of 30AWG silicone-coated wires. Use heat shrink tubing to secure the extended connection. Insert a long piece of heat tubing into the wiring of the laser and thread it through the barrel neopixel ring part. Press into the barrel until the almost all the way through.

Mount NeoPixel Ring Diffuser to Barrel
Thread the two wires from the laser diode into the center of the NeoPixel ring mount. Rotate to orient the ring so it's 90 degrees from the the seem of the barrel. Press the mount to the barrel to snap it into place.
Connected Components

With all the components wired together, run a last flight check on all the connections. Cross reference the circuit diagram and double check all of your solder points.

Preassembly Test

Power on the circuit and wait for the Pro Trinket boat loader to kick in. Pressing the push button should trigger your desired audio effect and NeoPixel LED animation.

If the sound doesn't trigger, check the audio file name is set to the proper track. If the issue persists, open up the handle and check your wiring.

If the NeoPixel LED doesn't fire, ensure the Arduino code matches the pins of the NeoPixel ring and push button in the circuit.
Assembly

Assembling Handle
Bundle the wires from both handle parts and fit them into the indent near the top of both handle parts. Slowly join the two parts to close them together. Hold the wires together so they're fitting through the opening near the top of the handle.

Ensure wires are NOT kinking or outside of the handle. Use a pair of tweezers to help push wires into enclosure.
Closing Handle Enclosure

Whist holding parts together, line up the stand-offs near the bottom of the handles and insert #4-40 Phillips machine screws. Use screw nuts to keep the parts together tightly. Fasten with screw driver until parts are tightly enclosed.
Handle Assembly Checkpoint

Verify the circuit is still working by powering it on and pressing the push button. Preceede if everything is still working!

Adding Rubberband to Trigger

The trigger piece has a hoop near the top. The top has a curved surface to allow a rubberband to be held in place. Insert a small rubber band (ideally the kind for hair braids) through the hoop. Push it in half way and folder the end over to tie it to the hoop. Pull to tighten the knot.
Mount Trigger

Grab the trigger box part with the extended stand-off. Fit the trigger into the pin near the finger opening and place the rubberband over the extended stand-off. Press the trigger into the pin to snap it into place.
Mount Slide Switch + Push Button

Gently place the trigger box part into the top of the handle. Line up the mounting holes and insert a #4-40 Phillips machine screw to secure the two parts together.

Fit the slide switch into the designated opening. Press it into the cavity to snap it into place.

Insert the push button into the cavity near the back of the trigger. Press it into the cavity to secure it.

Gently bundle remaining wiring and fit through the top opening of the trigger box.
Test Trigger Spring

Press the trigger to verify it's able to actuate the push button and spring back. To loosen the tolerance of the trigger, try using an x-acto knife to remove material from the pin hole.

Ensure the tolerance is loose enough for the trigger to spring back. It's no fun when the trigger box is enclosed and the trigger is stuck...

Assembling Trigger box

Fit the second half of the trigger box over the trigger and line up with the holes in the handle. Carefully press the two pieces place together, ensuring the wiring is fitted through the top opening. The halves should hold the slide switch and push button in place.
Secure Handle to Trigger box
Insert a #4-40 Phillips screw into the handle to secure the part to the trigger box.

Assembled Trigger + Handle
Check point, almost there. Power on the circuit to verify components are fully operational. Proceede if everything’s working.
Assembling Body
Place a half body piece onto the neopixel/laser mounted barrel. Line up the mounting holes and insert a \#4-40 Phillips screw to secure the parts together. Ensure the orientation lines up with the trigger box and handle parts. Keep wires grouped and bundled through the opening in the trigger box. The body parts features similar opening in the bottom lined up with the trigger box.

Add Diffusers
Insert the various diffusers into the opening of the body parts. Press them into the openings (should have a tight fit). Secure the parts together with adhesives.

Let the adhesives dry before proceeding.
Close Body

Place the second half of the body over the part and line up with mounting holes. Ensure the wires are fitted through the opening and not kinking or outside of the enclosure.
Secure Body Parts

Insert and fasten #4-40 Phillips machine screws into the various stand-offs in the body parts. Use screw nuts to keep the two halves tightly secure together.
Install Sights
Insert the faux sight ornaments to the body and barrel. They should snap into place with a tight fit. Use adhesives to secure them.

Install Battery
Insert the 500mAh Lithium Polymer battery through the back of the body.
Mount Speaker

Position the mini metal speaker into the speaker cap. Use fun-tac or mold puty to secure the speaker to the part.

Install Speaker Mount to Body

Press the speaker cap into the back of the body to snap it into place. It should have a tight fit. Use fun-tac or mold puty to the edge of the cap to secure the parts.
Final Check Point

Run a last check on all the screws to ensure everything is nice and tight. Apply adhesives to the two sights (the one for the body and the other for the barrel). Allow the adhesives to dry. Power it on and try it out!