Zelda Master Sword
Created by Ruiz Brothers

https://learn.adafruit.com/zelda-master-sword

Last updated on 2023-08-29 04:13:06 PM EDT
<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
</tr>
<tr>
<td>• We even made a smaller blade for children!</td>
</tr>
<tr>
<td>• Parts</td>
</tr>
<tr>
<td><strong>Software</strong></td>
</tr>
<tr>
<td>• Install CircuitPython</td>
</tr>
<tr>
<td>• Adafruit Circuit Python Libraries</td>
</tr>
<tr>
<td>• Upload Files</td>
</tr>
<tr>
<td>• Mu Python Editor</td>
</tr>
<tr>
<td>• Audio Files</td>
</tr>
<tr>
<td><strong>Circuit Diagram</strong></td>
</tr>
<tr>
<td><strong>3D Printing</strong></td>
</tr>
<tr>
<td>• 3D Printed Parts</td>
</tr>
<tr>
<td>• CAD Files</td>
</tr>
<tr>
<td>• Share, Make, Remix</td>
</tr>
<tr>
<td>• Settings</td>
</tr>
<tr>
<td>• Translucent PLA Filament</td>
</tr>
<tr>
<td>• De-Brim</td>
</tr>
<tr>
<td>• DeBurr</td>
</tr>
<tr>
<td>• Sanding</td>
</tr>
<tr>
<td>• Super Glue the Edges</td>
</tr>
<tr>
<td>• Hot Glue</td>
</tr>
<tr>
<td>• Painting</td>
</tr>
<tr>
<td><strong>Assemble</strong></td>
</tr>
<tr>
<td>• JST Battery</td>
</tr>
<tr>
<td>• Solder the PropMaker Wing</td>
</tr>
<tr>
<td>• Measure wires</td>
</tr>
<tr>
<td>• USB wire</td>
</tr>
<tr>
<td>• Reset cable</td>
</tr>
<tr>
<td>• Slide Switch</td>
</tr>
<tr>
<td>• Check measurements</td>
</tr>
<tr>
<td>• Mount the Slide Switch</td>
</tr>
<tr>
<td>• NeoPixel Strip</td>
</tr>
<tr>
<td>• NeoPixel Splitter</td>
</tr>
<tr>
<td>• Hot Glue Components</td>
</tr>
<tr>
<td>• Air Duster</td>
</tr>
<tr>
<td>• Hot Glue Strips</td>
</tr>
<tr>
<td>• Mount USB jack</td>
</tr>
<tr>
<td>• Buttons</td>
</tr>
<tr>
<td>• Speaker holes</td>
</tr>
<tr>
<td>• Paracord Rope</td>
</tr>
<tr>
<td>• Hot Glue Blade</td>
</tr>
<tr>
<td>• Complete!</td>
</tr>
</tbody>
</table>
Overview

In this project we'll build the legendary Zelda Master Sword! This light up sword is motion activated, so you can shoot blasts with lights and sound effects.

You can battle to hear random clash sounds or aim to blast enemies with ease!

This Master Sword is fully 3D Printed! NeoPixel LED Strips and the Adafruit PropMaker Feather Wing animate colors and plays sound effects!

You can recharge the battery or even add new sounds with the USB port. The sword mounts on a computer, just like a USB Drive!

Watch 3D Hangout for 1+hr of live demo, guide walkthrough and more!
Add new LED animations or program new motions to create custom actions. All the code and libraries live right on the device, so no need to install any programs to edit your code!

The PropMaker and Feather M4 Express make it easy to add more buttons, sensors or even a 3 watt LED!

The sword prints in parts and is adhered together. Components are installed in the handle, LED strips line the middle of the blade to create a beautifully diffused blade!

Designed by Garrett Kearney of CHAOS CORE TECH, its open to edit!

We modified the blade to allow it to swing easier and to cut down to the length of the LEDs.

We even made a smaller blade for children!
Copy and paste this friendly list of products.

Adafruit Feather M4 Express
Adafruit Prop-Maker FeatherWing (Assembled Headers)
Mini Skinny NeoPixel LED Strip
2200mAh Lipo Battery
Mini oval speaker
DIY USB micro B Female Plug
DIY USB micro B Male Plug
10 wire silicon cover stranded ribbon cable
3-pin PH JST cable
6mm Slim Tactile Buttons
JST Extension

Adafruit Feather M4 Express - Featuring ATSAMD51
It's what you've been waiting for, the Feather M4 Express featuring ATSAMD51. This Feather is fast like a swift, smart like an owl, strong like a ox-bird (it's half ox,...
https://www.adafruit.com/product/3857

Assembled Adafruit Prop-Maker FeatherWing
The Adafruit Feather series gives you lots of options for a small, portable, rechargeable microcontroller board. Perfect for fitting into your next prop build! This FeatherWing will...
https://www.adafruit.com/product/4145
Adafruit Mini Skinny NeoPixel Digital RGB LED Strip - 60 LED/m
So thin. So mini. So teeeeeeeny-tiny. It's the 'skinny' version of our classic NeoPixel strips! These NeoPixel strips have 60 digitally-addressable pixel Mini LEDs per...
https://www.adafruit.com/product/2959

Lithium Ion Cylindrical Battery - 3.7v 2200mAh
Need a big battery for your project? This lithium-ion battery contains a 2200mAh and a protection circuit that provides over-voltage, under-voltage, and over-current protection. Yet,...
https://www.adafruit.com/product/1781

Mini Oval Speaker - 8 Ohm 1 Watt
Hear the good news! This wee speaker is a great addition to any audio project where you need 8 ohm impedance and 1W or less of power. We particularly like...
https://www.adafruit.com/product/3923

Tactile Switch Buttons (6mm slim) x 20 pack
Slim clicky momentary switches are standard input "buttons" on electronic projects. These are half the width of classic 6mm tactile switches so they line up better on a...
https://www.adafruit.com/product/1489
USB DIY Connector - MicroB Female Plug
As an addition to our other USB DIY shells, we have a 'micro B female' in a panel-mount-like connector. This isn't true panel mount but you could glue the black plastic...
https://www.adafruit.com/product/1829

USB DIY Slim Connector Shell - MicroB Plug
Make your own USB connections without slicing apart a USB cable and soldering those thin wires inside. This is the 'slim' version of the DIY MicroB Male connector. It's a...
https://www.adafruit.com/product/1826

Silicone Cover Stranded-Core Ribbon Cable - 10 Wire 1 Meter Long
For those who are fans of our silicone-covered wires, but are always looking to up their wiring game. We now have Silicone Cover Ribbon cables! These may look...
https://www.adafruit.com/product/3890

STEMMA JST PH 2mm 3-Pin to Male Header Cable - 200mm
This cable will let you turn a JST PH 3-pin cable port into 3 individual wires with high-quality 0.1" male header plugs on the end. We're carrying these to match up with our...
https://www.adafruit.com/product/3893
Software

Install CircuitPython

The Adafruit Feather M4 ships with CircuitPython, but let’s go ahead and update it to the latest version. It’s super easy with the circuitpython.org website, just click the link below to launch the page. There you can choose to install the latest stable release or the latest beta.

Quick Start

- Connect board to computer via a known good USB and double press the reset button.
- Download the CircuitPython UF2 and upload to the FEATHERBOOT drive.
- Open CIRCUITPY drive and upload the required libraries (listed below) and code .py

Adafruit Circuit Python Libraries

Download the CircuitPython library bundle and unzip the folder. Create a new folder on the CIRCUITPY drive and name it lib. The following libraries are required to run the code properly. Double check to ensure all of the files and folders are inside the lib folder on the CIRCUITPY drive.

- adafruit_bus_device (directory)
- adafruit_lis3dh.mpy
- neopixel.mpy
Upload Files
Click the link below to download the project zip – This contains the code and audio files. Upload the code.py file to the CIRCUITPY drive root (main) folder.

Create a new folder on the CIRCUITPY drive and name it sounds. Upload the audio files to that folder. The code will run properly when all of the files have been uploaded.

```python
import time
import random
import digitalio
import audiocore
import board
import neopixel
import adafruit_lis3dh

# CUSTOMISE COLORS HERE:
COLOR = (0, 120, 120)      # Default idle is light blue
ALT_COLOR = (255, 50, 0)    # hit color is orange
```
# CUSTOMISE IDLE PULSE SPEED HERE: 0 is fast, above 0 slows down
IDLE_PULSE_SPEED = 0  # Default is 0 seconds
SWING_BLAST_SPEED = 0.007

# CUSTOMISE BRIGHTNESS HERE: must be a number between 0 and 1
IDLE_PULSE_BRIGHTNESS_MIN = 0.2  # Default minimum idle pulse brightness
IDLE_PULSE_BRIGHTNESS_MAX = 1  # Default maximum idle pulse brightness

# CUSTOMISE SENSITIVITY HERE: smaller numbers = more sensitive to motion
HIT_THRESHOLD = 250
SWING_THRESHOLD = 150

# Set to the length in seconds of the "on.wav" file
POWER_ON_SOUND_DURATION = 1.7

NUM_PIXELS = 83  # Number of pixels used in project
NEOPIXEL_PIN = board.D5
POWER_PIN = board.D10

enable = digitalio.DigitalInOut(POWER_PIN)
enable.direction = digitalio.Direction.OUTPUT
enable.value = False

strip = neopixel.NeoPixel(NEOPIXEL_PIN, NUM_PIXELS, brightness=1, auto_write=False)
strip.fill(0)  # NeoPixels off ASAP on startup
strip.show()

audio = audioio.AudioOut(board.A0)  # Speaker
wave_file = None

# Set up accelerometer on I2C bus, 4G range:
i2c = board.I2C()  # uses board.SCL and board.SDA
i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
accel = adafruit_lis3dh.LIS3DH_I2C(i2c)
accel.range = adafruit_lis3dh.RANGE_4_G

COLOR_IDLE = COLOR # 'idle' color is the default
COLOR_HIT = ALT_COLOR  # "hit" color is ALT_COLOR set above
COLOR_SWING = ALT_COLOR  # "swing" color is ALT_COLOR set above

def play_wav(name, loop=False):
    """
    Play a WAV file in the 'sounds' directory.
    :param name: partial file name string, complete name will be built around
      this, e.g. passing 'foo' will play file 'sounds/foo.wav'.
    :param loop: if True, sound will repeat indefinitely (until interrupted
      by another sound).
    """
    global wave_file  # pylint: disable=global-statement
    print("playing", name)
    if wave_file:
        wave_file.close()
    try:
        wave_file = open('sounds/' + name + '.wav', 'rb')
        wave = audiocore.WaveFile(wave_file)
        audio.play(wave, loop=loop)
    except OSError:
        pass # we'll just skip playing then

def power_on(sound, duration):
    """
    Animate NeoPixels with accompanying sound effect for power on.
    :param sound: sound name (similar format to play_wav() above)
    :param duration: estimated duration of sound, in seconds (>0.0)
    """
prev = 0
start_time = time.monotonic()  # Save audio start time
play_wav(sound)
while True:
    elapsed = time.monotonic() - start_time  # Time spent playing sound
    if elapsed > duration:  # Past sound duration?
        break  # Stop animating
    animation_time = elapsed / duration  # Animation time, 0.0 to 1.0
    threshold = int(NUM_PIXELS * animation_time + 0.5)
    num = threshold - prev  # Number of pixels to light on this pass
    if num != 0:
        strip[prev:threshold] = [ALT_COLOR] * num
        strip.show()
        prev = threshold

def mix(color_1, color_2, weight_2):
    """Blend between two colors with a given ratio.
    :param color_1: first color, as an (r,g,b) tuple
    :param color_2: second color, as an (r,g,b) tuple
    :param weight_2: Blend weight (ratio) of second color, 0.0 to 1.0
    :return (r,g,b) tuple, blended color
    """
    if weight_2 < 0.0:
        weight_2 = 0.0
    elif weight_2 > 1.0:
        weight_2 = 1.0
    weight_1 = 1.0 - weight_2
    return (int(color_1[0] * weight_1 + color_2[0] * weight_2),
            int(color_1[1] * weight_1 + color_2[1] * weight_2),
            int(color_1[2] * weight_1 + color_2[2] * weight_2))

swing_sounds = ['swing1',
                'swing2',
                'swing3',
                'swing4',
]

hit_sounds = ['hit1',
              'hit2',
              'hit3',
              'hit4',
]

mode = 0  # Initial mode = OFF

# Setup idle pulse
idle_brightness = IDLE_PULSE_BRIGHTNESS_MIN  # current brightness of idle pulse
idle_increment = 0.01  # Initial idle pulse direction

# Main loop
while True:
    if mode == 0:  # If currently off...
        enable.value = True
        power_on('on', POWER_ON_SOUND_DURATION)  # Power up!
        play_wav('idle', loop=True)  # Play idle sound now
        mode = 1  # Idle mode

        # Setup for idle pulse
        idle_brightness = IDLE_PULSE_BRIGHTNESS_MIN
        idle_increment = 0.01
        strip.fill([int(c*idle_brightness) for c in COLOR])
elif mode >= 1:  # If not OFF mode...
x, y, z = accel.acceleration  # Read accelerometer
accel_total = x * x + z * z
# (Y axis isn't needed, due to the orientation that the Prop-Maker
# Wing is mounted. Also, square root isn't needed, since we're
# comparing thresholds...use squared values instead.)
if accel_total > HIT_THRESHOLD:  # Large acceleration = HIT
    TRIGGER_TIME = time.monotonic()  # Save initial time of hit
    play_wav(random.choice(hit_sounds))  # Start playing 'hit' sound
    COLOR_ACTIVE = COLOR_HIT  # Set color to fade from
    mode = 3  # HIT mode
elif mode == 1 and accel_total > SWING_THRESHOLD:  # Mild = SWING
    TRIGGER_TIME = time.monotonic()  # Save initial time of swing
    play_wav(random.choice(swing_sounds))  # Randomly choose from available
    swing sounds

    # make a larson scanner animation_time
    strip_backup = strip[0:-1]
    for p in range(-1, len(strip)):
        for i in range(p-1, p+2): # shoot a 'ray' of 3 pixels
            if 0 <= i < len(strip):
                strip[i] = COLOR_SWING
        strip.show()
        time.sleep(SWING_BLAST_SPEED)
    if 0 <= (p-1) < len(strip):
        strip[p-1] = strip_backup[p-1]  # restore previous color at the
tail
        strip.show()
while audio.playing:
    pass # wait till we're done
mode = 2  # we'll go back to idle mode
elif mode == 1:
    # Idle pulse
    idle_brightness += idle_increment  # Pulse up
    if idle_brightness > IDLE_PULSE_BRIGHTNESS_MAX or \
        idle_brightness < IDLE_PULSE_BRIGHTNESS_MIN:  # Then...
        idle_increment *= -1  # Pulse direction flip
    strip.fill([int(c*idle_brightness) for c in COLOR_IDLE])
    strip.show()
    time.sleep(IDLE_PULSE_SPEED)  # Idle pulse speed set above
elif mode > 1:  # If in SWING or HIT mode...
    if audio.playing:  # And sound currently playing...
        blend = time.monotonic() - TRIGGER_TIME  # Time since triggered
        if mode == 2:  # If SWING,
            blend = abs(0.5 - blend) * 2.0  # ramp up, down
        strip.fill(mix(COLOR_ACTIVE, COLOR, blend))  # Fade from hit/swing
to base color
        strip.show()
    else:  # No sound now, but still SWING or HIT modes
        play_wav('idle', loop=True)  # Resume idle sound
        mode = 1  # Return to idle mode

Mu Python Editor

Check out Mu, it's a simple Python editor that works with Adafruit CircuitPython
hardware. It's written in Python and works on Windows, MacOS, Linux and Raspberry
Pi. The serial console is built right in so you get immediate feedback from your
board's serial output!
Audio Files

The sounds used in this project were created with samples from Video CoPilots Motion Pulse sound pack. Hits and swings were mixed with captured sounds in-game from Zelda: BOTW.

This pack contains sounds that are already in the supported audio format.

Adafruit CircuitPython supports 16-bit, Mono, 22.050kHz .wav audio format. See this guide to help format any audio files you might want to use in this project besides the files provided.

In the main loop, the swing and hit modes randomly choose from a list of sounds. For example, swing1.wav, swing2.wav, swing3.wav, etc. This makes the motion effects feel much more varied and less repetitive.

- Power on – on.wav
- Idle loop – idle.wav
- Swing 1 – swing1.wav
- Swing 2 – swing2.wav
- Swing 3 – swing3.wav
- Swing 4 – swing4.wav
- Hit 1 – hit1.wav
- Hit 2 – hit2.wav
- Hit 3 – hit3.wav
- Hit 4 – hit4.wav

Circuit Diagram

The wiring diagram below provides a visual reference for connecting the components. It is not true to scale, it is just meant to be used as reference. This diagrams was created using the Fritzing software package.

Take a moment to review the components in the circuit diagram. This illustration is meant for referencing wired connections - the length of wire, position and size of components are not exact.
The Prop Maker Wing should have male headers soldered onto the underside. The Feather M4 Express should have female headers soldered on. This when everything is done, the Prop Maker Wing will snap onto the Feather M4 Express below it.

We measured and cut wires to have enough slack to reach each component.

We used silicone ribbon wire to make them easier to coil and manage each wire inside the tight spacing. The wire lengths used are listed below:

- JST extender for Battery: 54mm
- Reset: 45mm
- USB Extender: 20mm
- LED Strip: 20mm
- Slide Switch: 190mm
- Button: 7.5mm
- Paracord 813mm
3D Printing

3D Printed Parts

The parts in this kit are designed to be 3D printed with FDM based machines. STL files are oriented to print "as is". Parts require tight tolerances that might need adjustment of slice settings. Reference the suggested settings below.

CAD Files

The Fusion 360 source file is included and features original sketches and feature timeline along with easily editable user parameters. The parts can further be separated into small pieces for fitting on printers with smaller build volumes. Note: a STEP file is included for other 3D surface modeling programs such as Onshape, Solidworks and Rhino.

Share, Make, Remix

This master sword was originally designed by Garrett Kearney (from Chaos Core Tech (Youtube Channel)) – The files are open to remix and feature nice details. The parts have been modified to fit all of the electronics and available to download. You can check out Garrett's project build video on YouTube. (.)

MasterSword.zip

Download Remixed STLs
Garrett's original design was modeled in Autodesk Fusion 360. It contains solid bodies, sketches and parametric feature timeline. It's a great resource to check out! Download link below.

Download Fusion 360 File

Original STLs

Settings
Use these settings as reference. Values listed were used in Cura slicing software.

0.2mm Layer Height / 0.4mm nozzle
0.38mm Line Width (inner & outer widths)
40mm/s printing speed
10% infill
Supports: No
Brim: 7mm

Translucent PLA Filament
The parts in this project were 3D printed using translucent PLA filament. Check out the stuff by 3D Solutech Printer Filament.

2.85mm Translucent PLA – 3D Solutech Printer Filament ()
De-Brim
The parts are printed so they lay flush when sandwiched together. We used a 7mm brim to ensure the parts are as flat as they can be to better align the two halves. Use flush cutters to help remove the brim.

DeBurr
Next use a deburring tool or a hobby knife to clean up and smooth out the sharp edges.
Sanding
After the edges are smooth, you'll need to sand down the ends of the blades to make it is easy for the super glue to adhere parts together.

We used a rough 120 grit sandpaper to quickly smooth out the ends.

Make sure to that test ends parts are flat and fit together without any gaps.
Super Glue the Edges
Apply a generous amount of super glue to the edges and then carefully align the ends. We used StarBound Medium Dry super glue with a narrow tip to precisely add to the edges that meet together. Apply pressure to help parts adhere.

Hot Glue
Allow the super glue to dry. Next reinforce parts by adding hot glue to the inside of the blade sections. Apply a small amount of hot glue to each section of the blade and allow it to dry.

You can also speed up dry times by spraying each section with an Air Duster. This technique is used extensively later in the build.
Painting
After all sections are dry and adhered, you can move on to painting!

We've seen the sword in purple or dark blue with gold trim, but you can mix it up with any colors you like!

Acrylic paint with a thin brush helped to fill in the small details around the curvy scripts, but you can mask out sections and use an airbrush.

We needed about six coats of paint to evenly coat the handle.

Assemble

Be careful using hot glue and cutting tools to avoid burns and cuts.
JST Battery
The JST battery connector is located on the side of the Feather M4 Express. To fit the boards inside the handle, you will need to reroute the connection so the battery connects from the end of the Feather M4 Express.

Measure and cut a JST Extension cable so it's 54mm long. Tin and solder the wires to the back of the JST connector. Make sure to verify the power and ground are connected in the correct pins with a multimeter. Check that the wires have a solid connection and curve the wire towards the prototype area on the Feather M4.

Solder the PropMaker Wing
Now we can place the PropMaker Feather wing over the Feather M4 Express.

We used the Assembled Prop-Maker FeatherWing to make it easy to solder to the Feather M4 Express.

Align the headers so the PropMaker sits flush on top of the JST battery connector. We can use a Panavise to securely hold the boards while we solder each header pin.
Measure wires
Now to dry fit the boards in the handle. Here, measure wires for the USB jack, Reset Button, Slide switch, LED strips and the optional mode switch button.

Place the USB connector and jack over the end of the handle to visualize how the rest button will sit on top of the USB jack.

Make sure the board is flush against end of the handle to get accurate measurements of the wires.

USB wire
We love silicone ribbon cables. They don't melt when soldering and help to keep wires neat and organized.

Measure 20mm of ribbon cable. Reference the circuit diagram for the USB connects.

First, apply a bit of solder to each pin on each USB connector and the carefully solder each wire.
Reset cable
Apply solder to the Reset and Ground pins on the PropMaker.

We utilized the strain relief hole to help route the Reset wires before soldering to the button.

Solder the wire to each of the two pins on the button. We'll insulate the connects with hot glue later to allow it to fit in the handle.
Slide Switch
Next solder the On and Off slide switch to the G and EN pins on the PropMaker. Tin the through holes first and then reheat while press the wires into each through hole.

We'll need to route the wire between the center of the two boards to the opposite side of the board as shown. Pass the wires through and then solder the slide switch.

Check measurements
Once all of the wires are soldered, go ahead a double check that the lengths of all wires are correct and have enough slack to reach.

Mount the Slide Switch
Maneuver the wires for the On /Off slide switch so they are flush against the handle walls. The slide switch press fits into the slot on the handle.

Gently pull the two metal tabs on the slide switch metal housing to increase the slot tolerance. Hot glue in place to help secure it.
NeoPixel Strip
Now to preparing the NeoPixel strips. Identify the start and end of the strip by looking at the direction of the small arrows printed on the strip.

Cut off the large connector and remove a small bit of silicone so you can solder to the pads on the strip. Repeat with the second NeoPixel Strip.

NeoPixel Splitter
Both strips will connect via a Y splitter. We used a JST PH 3-Pin to Male Header Cable to build our splitter.

Cut six silicone wires 20mm long and then use tweezers to hold them together while soldering to the jumper ends of the JST PH 3-Pin Header Cables.

Check that the connections are solid and then slip on heat shrink to insulate the connections.

Next is to test that the LED strips turn on.
Hot Glue Components
After verifying that all wires and components power up without any loose connections, you can start to mount the board and wires.

Air Duster
Apply hot glue to the corners of the board and then use Air Dust to quickly dry the hot glue. You can turn the Air Duster upside down and carefully spray quick bursts over each glue spot.

Some condensation will appear but quickly fades away as the temperature stabilizes.

Hot Glue Strips
Lay the strips so they have a small amount of spacing between the edge and center of the blade walls. This will help to diffuse the blade. Apply hot glue and Air Duster to hold the strips in place. Add glue to every 80mm or so to straighten out the strip.

Mount USB jack
Apply a small amount of hot glue to the end of the handle where the USB jack sits, but not at the tip end. Press the USB jack over the glue.

Check that it's properly aligned and then quickly spray Air Duster to hold it in place. Add another small drop of hot glue over the USB pins to insulate the connections. Allow the hot glue to drip down and adhere to the handle before using the Air Duster.
Buttons
Once the hot glue dries on the USB jack, place the Reset button over the USB jack plastic housing and then apply a small amount of hot glue on the sides of the Reset button. Don’t apply hot glue to the top. This will make it too thick for both halves to join.

Insulate the pins of the Rest button with a small amount of hot glue.

The optional Mode switch button can sit above the Reset button. You can add a small drop of hot glue to the sides and the pins to insulate the connections.

Speaker holes
Once the speaker is mounted in the handle, it will sound muffled, so a couple of holes should be put in the handle. We heated a poking tool (usually included with 3D Printers) and made a pattern of holes over the area around the speaker.

Add holes to both sides of the handle.

Poking each hole will leave raised edges around the holes, but they easily clean up with a diagonal Flush Cutter tool.
Paracord Rope
Now you can start to attach both halves of the sword. The rope detail on the handle will actually help to hold both halves together.

Cut a piece of green paracord 320.5mm long and then tightly criss cross the rope down the length of the handle.

Tie the ends and then apply a small amount of super glue to the area were the rope criss crosses to hold the shape of the paracord.

Hot Glue Blade
With the handle tightly attached, you can move on to adhering both halves of the blades together.

Start at the base of the blade. Apply a generous amount of hot glue as shown in the diagram. Lay the blade on a flat surface and apply pressure as you add hot glue to each section. Use Air Duster to quickly dry the hot glue.

Allow the hot glue that spreads out of the sides of the blade to dry before using a hobby knife or flush cutters to trim them off.
Complete!

Take your fully assembled sword on your next quest! Don't forget to make a second smaller one for those younger Adventurers! Just remember to fully charge the battery before going on long missions!