



## PiGRRL Zero

Created by Ruiz Brothers



Last updated on 2018-07-26 09:10:22 PM UTC

# Guide Contents

Guide Contents	2
Overview	5
Features	5
Project Expectations	6
3D Printed Parts	6
Prerequisite Guides	7
Parts	7
Tools and Supplies	8
Circuit Diagram	11
Wired Connections	11
GPIO Wired Connections	12
Gamepad Connections	12
Software	13
Ready-To-Go Raspberry Pi IMG	13
Software [Manual]	14
Download & Burn RetroPie	14
Ethernet Hub and USB Hub w/ Micro USB OTG Connector	14
Setup RetroPie	14
Testing Retrogame	15
Install PiTFT (fbcp) Support	15
Edit Config.txt	16
Installing Keypress (retrogame) support	17
3D Printing	19
3D Files	19
Suggested Slice Settings	20
Ninjaflex / TPE Filaments	20
3D Post Processing	20
Download, Modify, Remix	21
But I Don't Have a 3D Printer!	21
Using 3D Hubs App on Thingiverse	22
Gamepad	23
Install Buttons on Gamepad PCBs	23
Solder Buttons to Gamepad	24
Pi Zero	25
Install Headers to Pi Zero	25
Tack Headers to PCB	26
Soldered Headers	26
PiTFT	27
Cut Trace on Pin #18	27
Gamepad To PiTFT	28
Measure Gamepad Wires	28
Strip & Tin Wires	28
Inserting Wires to PiTFT	29
Solder Wire to PiTFT	29
Solder Wire to Gamepad	30

Wired Gamepad	30
Test Gamepad Connections	31
Wired Gamepads	32
<b>PiTFT To Pi Zero</b>	<b>33</b>
Install PiTFT to Pi Zero	33
Connecting PiTFT to Pi Zero	34
Solder PiTFT to Pi Zero	34
Soldered PiTFT to Pi Zero	35
Gamepad Ground Connections	35
<b>Power</b>	<b>36</b>
PowerBoost Connections	36
Prep Slide Switch	37
Solder Wires to Slide Switch	38
Connect Switch to PowerBoost 1000C	39
PowerBoost Wires	40
Connect PowerBoost to Raspberry Pi	40
<b>USB</b>	<b>41</b>
USB Adapters	41
Remove Legs from USB Female Jack	41
Prep USB Pins	42
USB Wires	42
Connect Wires to USB OTG Adapter	43
Connect Wires to Female USB Jack	43
<b>Shoulder Buttons</b>	<b>44</b>
Shoulder Buttons	44
Measure Shoulder Button Wires	44
Shoulder Button Wires	44
Connect Wires to Buttons	45
Connect Wires to PiTFT	45
Ground Shoulder Buttons	45
<b>Mounting Components</b>	<b>46</b>
Connect USB OTG Adapter	46
Test Final Circuit	46
Verify Test	46
Mount PowerBoost 1000C	47
Mount Slide Switch	47
Mount Female USB Jack	48
Install Battery	49
Screws and Washers	50
Install Actuators	51
Install PiTFT to Case	51
Fasten Screws	52
Gamepad PCB Mounting Holes	53
Mount Gamepad PCBs	54
Mount Shoulder Buttons	55
Final Circuitry Check	56
Close Case	57
Install Shoulder Actuators	58
Final Build	59
<b>Usage &amp; Extras</b>	<b>60</b>

PiTFT Buttons	60
Connecting to WiFi	60
Uploading ROMs	60
USB Port	60
Audio Adapter	60
Exiting a ROM/Game	61
<b>RetroPie: Improving Emulator Performance</b>	<b>62</b>
Overclocking	62
Choosing an Alternate Emulator	63
Installing Additional Emulators	65
Updating Emulators to Latest Versions	66
Troubleshooting RetroPie and retrogame	68
retrogame Related Troubleshooting	68
Some of my buttons/controls aren't working!	68
NONE of my buttons/controls are working!	68
My controls only work if there's also a USB keyboard plugged in!	68
Retrogame doesn't work with my optical buttons!	68
I ran Adafruit's retrogame installer script and rebooted, and now the keyboard and network are unresponsive!	68
<b>RetroPie Related Troubleshooting</b>	<b>69</b>
My controls work in the EmulationStation UI, but not in one or more specific emulators!	69
The ROM files I have worked in a different emulator before, but aren't working in RetroPie!	69
Installing RetroPie Packages	69
Accessing Alternate Emulators	71
More RetroPie Help	72

## Overview



You've seen PiGRRL, Super Game Pi, PiGRRL Pocket, PiGRRL 2 and now... it's time for PiGRRL Zero!

In this tutorial we'll turn the elusive \$5 Raspberry Pi Zero into portable game console!

### Features

- RetroPie 3.7 Emulation Station: SNES, NES, SEGA, N64, and many many more!
- 14 Buttons, including D-Pad, L & R shoulder, Start/Select, A, B, X, Y and two extras.
- 2.2" Adafruit PiTFT 320x240 Color Display
- Landscape portable console format

The USB female jack allows you to add any peripherals to the Raspberry Pi Zero, such as a WiFi or Bluetooth adapter, USB keyboard or mouse, USB Audio Adapter or USB Hub.

PLEASE BE AWARE - THIS IS NOT AN EASY/BEGINNER PROJECT! This project is for experienced makers, with successful soldering and Raspberry Pi projects. We DO NOT guarantee you will be able to build this project successfully!



## Project Expectations

The difficulty of this project is low-medium. There is wiring and soldering involved, but no PCB hacking or sawing like our previous builds. We think this is a great project for makers who have a little bit of Pi and 3D printing experience, or are first time makers with a lot of energy and willingness to learn! This tutorial will walk you through all the steps of the assembly and takes about weekend to accomplish.



## 3D Printed Parts

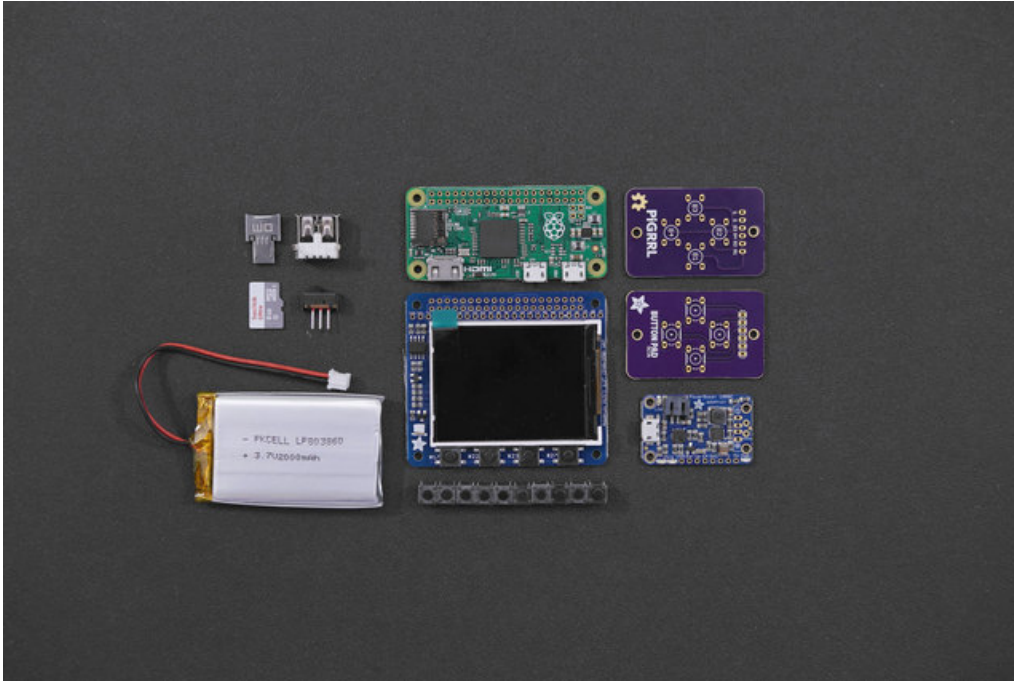
You'll need to 3D print the enclosure and buttons for this project. If you don't have access to one, 3DHubs.com is a

great option - simply download the STL files, upload them and choose a local hub to print and ship them to you!

## Prerequisite Guides

If you're new to soldering, definitely check out Collin's Lab: Soldering. You'll learn the basics of soldering and how fun it can be!

- [Learn how-to solder. Collin's Lab: Soldering \(https://adafru.it/noE\)](https://adafru.it/noE)



## Parts

You'll need the following components to make this project.

- 1x [Adafruit PiTFT 2.2"](http://adafru.it/2315) (<http://adafru.it/2315>)
- 1x [Raspberry Pi Zero](http://adafru.it/2885) (<http://adafru.it/2885>)
- 1x [PowerBoost 1000C](http://adafru.it/2465) (<http://adafru.it/2465>)
- 1x [2000mAh Battery](http://adafru.it/2011) (<http://adafru.it/2011>)
- 10x [6mm Tactile Buttons](http://adafru.it/367) (<http://adafru.it/367>)
- 1x [Slide Switch](http://adafru.it/805) (<http://adafru.it/805>)
- 1x [USB Mini Wifi Adapter](http://adafru.it/814) (<http://adafru.it/814>)
- 1x [Tiny OTG Adapter - USB Micro to USB](http://adafru.it/2910) (<http://adafru.it/2910>)
- 2x [PiGRRL Zero Gamepad PCBs](http://adafru.it/2934) (<http://adafru.it/2934>)
- 1x [2x20-pin Strip Dual Male Header](http://adafru.it/2822) (<http://adafru.it/2822>)

The following parts are optional.

- 1x [USB Audio Adapter](http://adafru.it/1475) (<http://adafru.it/1475>)



<https://adafruit.it/nyA>

<https://adafruit.it/nyA>

## Tools and Supplies

The following tools and supplies will help you build this project.

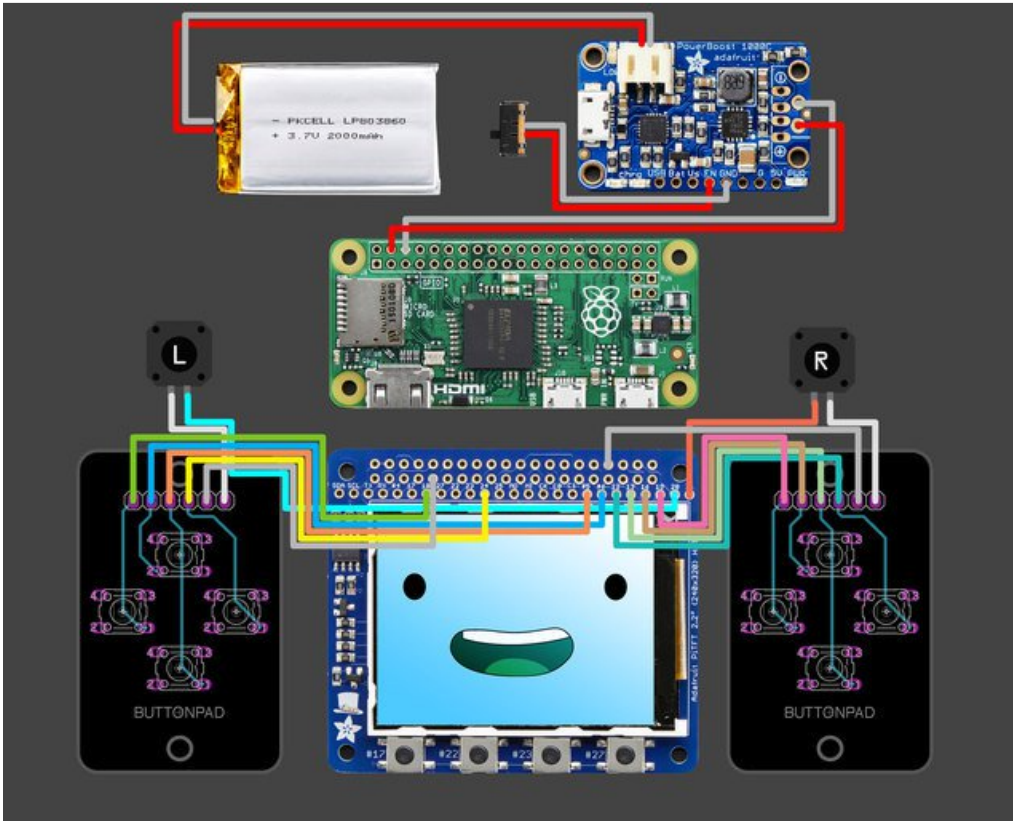
- [3D Printer \(https://adafruit.it/doT\)](https://adafruit.it/doT) + [Filament \(http://adafruit.it/2080\)](http://adafruit.it/2080)
- [Soldering Iron \(https://adafruit.it/doU\)](https://adafruit.it/doU) + [Solder \(https://adafruit.it/doU\)](https://adafruit.it/doU)
- [30AWG \(http://adafruit.it/2051\)](http://adafruit.it/2051) + [26AWG \(http://adafruit.it/1970\)](http://adafruit.it/1970) Wire
- [Helping Third Hands \(http://adafruit.it/291\)](http://adafruit.it/291) / [Panavise \(http://adafruit.it/151\)](http://adafruit.it/151)
- [Heat Shrink \(http://adafruit.it/1649\)](http://adafruit.it/1649)
- Glue / Mounting Tack
- [Wire Stripper \(http://adafruit.it/527\)](http://adafruit.it/527) / [Cutters \(http://adafruit.it/152\)](http://adafruit.it/152)
- Filing Tool / Hobby Knife
- 6x #4-40 3/8 (M3 x 9.5mm) machine screws







## Circuit Diagram



PLEASE BE AWARE - THIS IS NOT AN EASY/BEGINNER PROJECT! This project is for experienced makers, with successful soldering and Raspberry Pi projects. We DO NOT guarantee you will be able to build this project successfully!

## Wired Connections

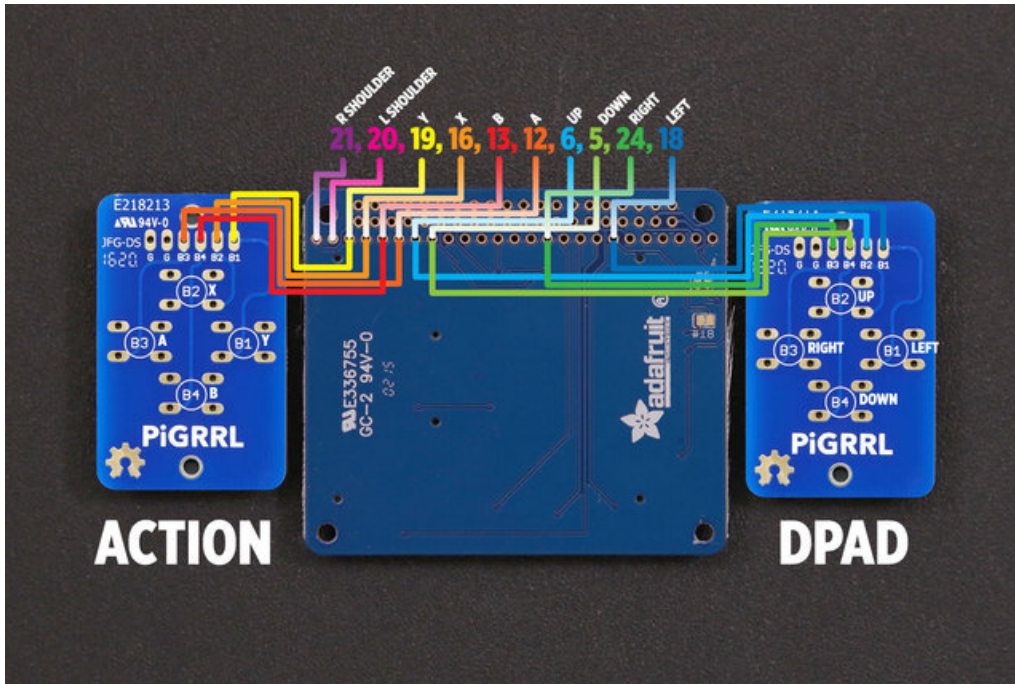
The diagram above depicts the connections for the power, audio and gamepad. Use this as a reference for wiring the components. Note, the length of wires and position of components are not exactly how the circuit will be - it's just a diagram to show connections.

```
{ 18,  KEY_LEFT }, // Left
{ 24,  KEY_RIGHT }, // Right
{ 5,   KEY_DOWN }, // Down
{ 6,   KEY_UP   }, // Up
{ 12,  KEY_Z    }, // A
{ 13,  KEY_X    }, // B
{ 16,  KEY_A    }, // X
{ 19,  KEY_S    }, // Y
{ 20,  KEY_Q    }, // L
{ 21,  KEY_W    }, // R
{ 17,  KEY_ESC }, // TFT 17
{ 22,  KEY_LEFTCTRL }, // TFT 22 SELECT
{ 23,  KEY_ENTER }, // TFT 23 START
{ 27,  KEY_4   }, // TFT 27
```

## GPIO Wired Connections

Follow the table above to reference which buttons to wire to the GPIO. The numbers on the left represent the corresponding GPIO number (NOT the pin number).

Gamepad PCB Update! Pins have been updated. Please reference the wiring diagram below.



## Gamepad Connections

When the PiTFT is flipped over, the connections are reversed. This diagram references the connections from the back of the PiTFT and Gamepad PCBs. This is most likely how you will wire the buttons to the pins on PiTFT. The colors match with each corresponding button and GPIO number.

## Software

PLEASE BE AWARE - THIS IS NOT AN EASY/BEGINNER PROJECT! This project is for experienced makers, with successful soldering and Raspberry Pi projects. We DO NOT guarantee you will be able to build this project successfully!

### Ready-To-Go Raspberry Pi IMG

Click the button below to download the PiGRRL Zero img. You'll need to burn this to an SD card (2 GB minimum) using a image burning tool, [like the ones listed in the guide \(https://adafru.it/dDL\)](https://adafru.it/dDL)

Once the img has been burnt to a blank SDCard, insert it into the Raspberry Pi Zero. The image is preloaded with tools, emulators, some games and preconfigured buttons for the gamepad controls. It will only work with the PiTFT (So plugging in an HDMI monitor might not display anything).

The image does not work with the Raspberry Pi Zero W – If you're using the Pi Zero W, you will need to manually setup the software.

<https://adafru.it/rne>

<https://adafru.it/rne>

After first boot, you'll need to expand the filesystem to fill the SD card. This requires having a keyboard attached temporarily (either through a USB hub or a USB-A-to-micro adapter).

From the EmulationStation GUI, press “F4” to exit to a Linux command prompt. Then type “sudo raspi-config”. The very first option in this menu is “Expand Filesystem.” Run that command. Afterward, tab over to the “Finish” button and reboot when prompted.

## Software [Manual]

We recommend you go with the ready-to-download image rather than trying to do this by hand, but here's details if you're interested!

### Download & Burn RetroPie

Game emulation is handled by a package called **RetroPie**. It's a complete Linux distribution designed specifically for running classic games on Raspberry Pi.

Download the current version from the [RetroPie web site \(https://adafru.it/rA3\)](https://adafru.it/rA3), then [write this to an SD card \(https://adafru.it/dDL\)](https://adafru.it/dDL) using Etcher or similar software.

We'll then make some modifications to tune this for the PiGRRL's buttons and small display.

Setup will require an **HDMI monitor**, **USB keyboard** and a **network connection** (a USB WiFi or **Ethernet** dongle, for example). This is best done **before** the Pi is enclosed in the PiGRRL case. If you have a spare Raspberry Pi board around, that's an ideal option...you could prepare the software on that system and then move the card over to the PiGRRL.

A USB/Ethernet hub + a Keyboard is ideal



Ethernet Hub and USB Hub w/ Micro USB OTG Connector

\$14.95  
IN STOCK

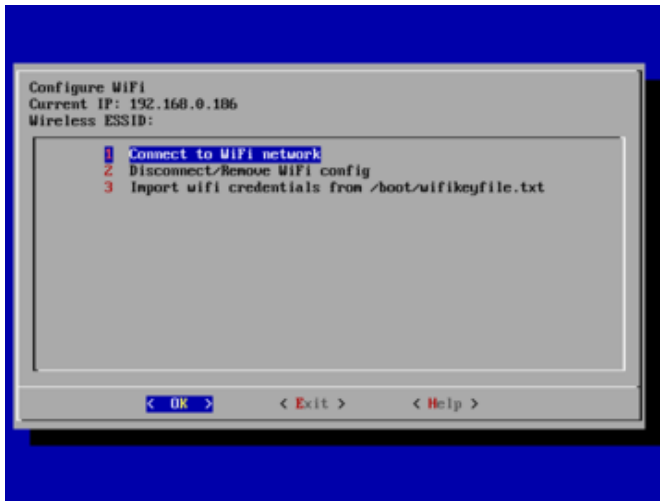
ADD TO CART

### Setup RetroPie

Insert the RetroPie card to your Pi, attach monitor and keyboard (and Ethernet, if networking that way), then power the system from a USB power source (a USB phone charger or a powered USB hub can usually work). The system will automatically reboot once (it needs this to make use of the whole SD card), then on second boot it will ask to configure the game controls...

**The PiGRRL buttons don't work yet; this is normal.** For initial setup, use the **USB keyboard** to select the D-pad directions (arrow keys), Start, Select, A and B keys. For anything else, just hold down the space bar or other key to skip that item. Don't worry, we'll re-do the keymap later once we've finished assembly! When finished, you'll see a graphical interface called *Emulation Station* where you'll select games and other options.

Let's **get this Raspberry Pi on the network** first. If WiFi, from the main EmulationStation screen, access the RetroPie settings using whatever key you've assigned as the "A" button. You'll see **WIFI** in this list:



Here you can select your WiFi network name and enter a password. It's not beautiful, but gets the job done.

Select "Exit" when done to return to the EmulationStation UI...

With networking enabled, we can now access the remaining software needed for the PiGRRL 2 experience. There are a couple ways to do this...

- **BEST:** Use an `ssh` terminal client to log into the Raspberry Pi at `retropie.local`  
This is recommended, as you can just copy-and-paste the commands that follow. The default name and password are "pi" and "raspberrypi," respectively.
- **OR:** Press "F4" to exit EmulationStation for a command-line prompt (works, but you'll need to type these commands *exactly*).

## Testing Retrogame

Once the SD card is finished, insert it into the Pi and boot it up. You should get emulation station to boot automatically. Configure your controls using a keyboard and test out the games in the ports section to make sure everything is running properly.

## Install PiTFT (fbcp) Support

This first sequence configures the system for the PiTFT display:

```
cd
curl https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/pitft-fbcp.sh >pitf
sudo bash pitft-fbcp.sh
```

Select the "PiGRRL Zero" option, which sets up various system parameters to match this project.

```
pi@retropie: ~  
This script enables basic PiTFT display  
support for portable gaming, etc. Does  
not cover X11, touchscreen or buttons  
(see adafruit-pitft-helper for those).  
HDMI output is set to PiTFT resolution,  
not all monitors support this, PiTFT  
may be only display after reboot.  
Run time ~5 minutes. Reboot required.  
  
CONTINUE? [y/N] y  
  
Select project:  
1. PiGRRL 2  
2. Pocket PiGRRL  
3. PiGRRL Zero  
4. Configure options manually  
  
SELECT 1-4: 3  
  
Device: pitft22  
HDMI framebuffer rotate: 0  
TFT MADCTL rotate: 270  
  
CONTINUE? [y/N] █
```

Answer "NO" to the reboot question...

```
pi@retropie: ~  
-- Detecting C compiler ABI info - done  
-- Detecting C compile features  
-- Detecting C compile features - done  
-- Check for working CXX compiler: /usr/bin/c++  
-- Check for working CXX compiler: /usr/bin/c++ -- works  
-- Detecting CXX compiler ABI info  
-- Detecting CXX compiler ABI info - done  
-- Detecting CXX compile features  
-- Detecting CXX compile features - done  
-- Configuring done  
-- Generating done  
-- Build files have been written to: /tmp/rpi-fbcp-master/build  
Scanning dependencies of target fbcp  
[ 50%] Building C object CMakeFiles/fbcp.dir/main.c.o  
[100%] Linking C executable fbcp  
[100%] Built target fbcp  
Configuring PiTFT...  
Done.  
  
Settings take effect on next boot.  
  
REBOOT NOW? [y/N] N  
Exiting without reboot.  
pi@retropie:~ $ █
```

## Edit Config.txt

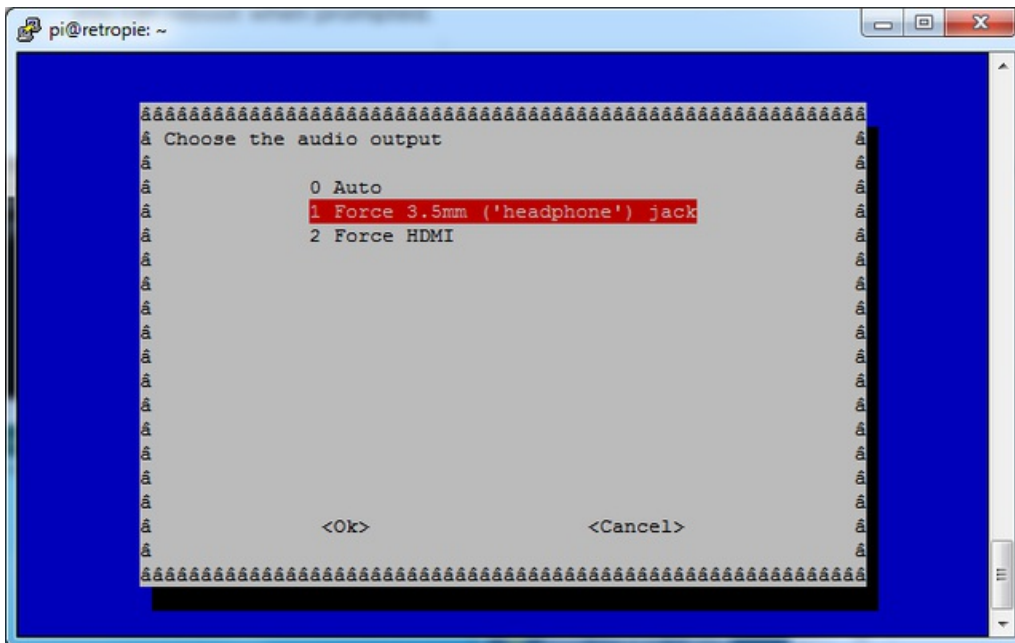
For optimized speed and decent audio from the emulators, may want to tweak some settings in the config.txt file, for overclocking/gpu configuration

run `sudo nano /boot/config.txt` and add/edit the end of the file:



```
gpu_mem=44
disable_audio_dither=1
overscan_scale=1
#gpu_mem_256=128
#gpu_mem_512=256
#gpu_mem_1024=256
dtoverlay=pitft22,rotate=270,speed=60000000,fps=40
display_rotate=0
hdmi_cvt=320 240 60 1 0 0 0
arm_freq=1000
core_freq=500
sdram_freq=450
over_voltage=6
```

Also, set the audio to 3.5mm with `sudo raspi-config` (under Advanced/Audio)



## Installing Keypress (retrogame) support

let's take care of this second script, which enables the PiGRRL buttons:

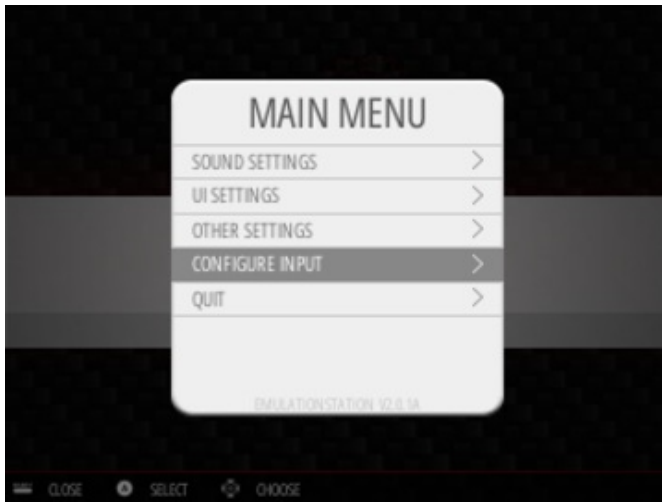
```
cd
curl https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/retrogame.sh >retro
sudo bash retrogame.sh
```

Again, select the "PiGRRL Zero" option. When finished, *now* you can reboot when prompted.

```
pi@retroPie: ~  
This script downloads and installs  
retrogame, a GPIO-to-keypress utility  
for adding buttons and joysticks, plus  
one of several configuration files.  
Run time <1 minute. Reboot recommended.  
  
Select configuration:  
1. PiGRRL 2 controls  
2. Pocket PiGRRL  
3. PiGRRL Zero  
4. Super Game Pi  
5. Two buttons + joystick  
6. Six buttons + joystick  
7. Adafruit Arcade Bonnet  
8. Quit without installing  
  
SELECT 1-8: 3  
Downloading, installing retrogame...OK  
Downloading, installing retrogame.cfg...OK  
Performing other system configuration...OK  
  
REBOOT NOW? [y/N]
```

After rebooting, the HDMI monitor may display a “no signal” message. This is normal. Not all monitors can handle the resolution setting we’re using. Once the PiTFT is wired up, that will be the primary display.

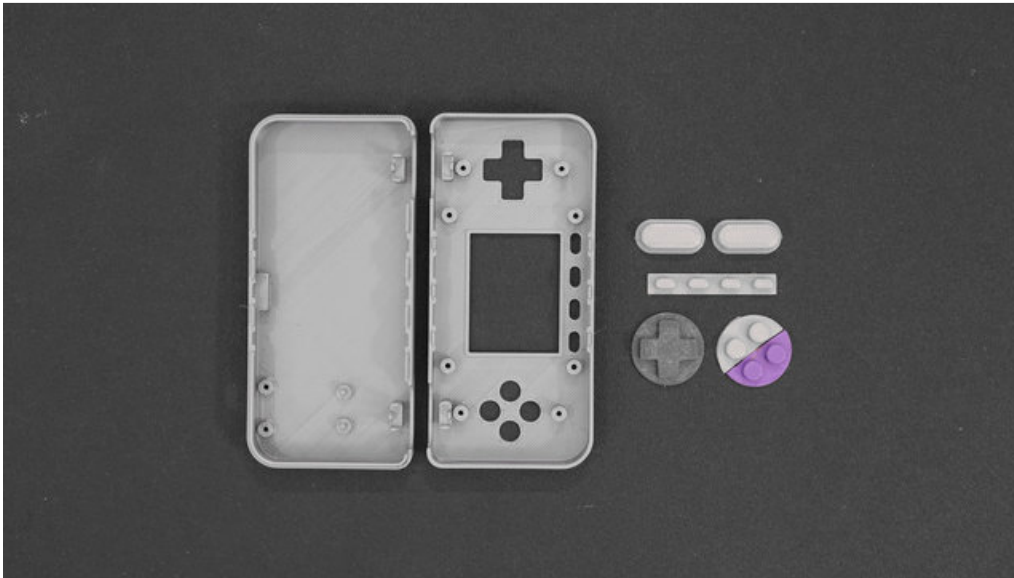
Also, after the system is assembled with the PiTFT and controls, you’ll need to re-do the controller setup. This might wait ‘til all the parts are assembled in the case.



From the main EmulationStation screen, press whatever key was assigned to the “Start” button to access the main menu. You’ll find an option here for “CONFIGURE INPUT.”

Go through the control setup process again using the PiGRRL buttons now instead of the keyboard; assign the D-pad directions, Start and Select buttons, A, B, X and Y. For anything else, hold down a key or button to skip it.

## 3D Printing



### 3D Files

Download the STLs by clicking on the button below. The table references the file names, their use and recommend material.

case-bot.stl	1x Enclosure top half	PLA
case-top.stl	1x enclosure bottom half	PLA
washer.stl	2x washers for bottom mounting holes of PiTFT	PLA
dpad.stl	1x Directional pad	Ninjaflex / TPE
buttons.stl	1x Actions buttons	Ninjaflex / TPE
pitft-buttons.stl	1x Buttons for PiTFT display	Ninjaflex / TPE
shoulder-button.stl	2x shoulder buttons	Ninjaflex / TPE

<https://adafru.it/nAe>

<https://adafru.it/nAe>

<https://adafru.it/nqB>

<https://adafru.it/nqB>

<https://adafru.it/nAf>

<https://adafru.it/nAf>

## Suggested Slice Settings

Each printer and slicing software is different. Although we can't possibly test each printer profile out, we do however have some suggested settings (these are for a 3D printer with a 0.4mm nozzle and 1.75mm filament).

- Extrusion Multiplier: 1.00
- Extrusion Width: 0.48mm
- Retraction Dist. 1mm
- Retraction Speed: 30mm/s
- Layer Height: 0.2mm
- Top / Bottom Layers: 4
- Outline/ Perimeter Shells: 2
- First Layer Height: 100%
- First Layer Width: 100%
- First Layer Speed: 50%
- Default Printing Speed: 60mm/s - 90mm/s
- Movement Speed: 90mm/s - 150mm/s
- Solid Infill Underspeed: 50%
- Interior Infill: 25%
- Outline Overlap: 25%
- Infill Width: 100%
- Extruder Temp: 220c
- Heated Bed: 0c - 60c

## Ninjabflex / TPE Filaments

Since TPE filament is a rubber based material, the slice settings will need to be slightly different. Typically you need to print hotter, slower, and completely disable retraction.

- Extruder Temp: 230c - 240c
- Default Printing Speed: 30mm/s
- Movement Speed: 40mm/s
- Retraction: OFF

## 3D Post Processing

It's a good idea to thoroughly inspect your parts after 3D printing. Check to see if there's any defects in the surfaces and walls of each part.

Test fitting the parts to see if the tolerances are balanced. Do the DPad and action buttons fit into the holes? Can the two halves snap fit together? Are the standoffs strong, or do they easily break apart? These are all dependent on your slice settings.

There's a total of 8 standoff's with mounting holes. It's a good idea to tap these in order to create the threads before mounting components. They're sized for #4-40 machine screws (or M3).

If you're interested and giving the surface a smooth finish, you can optional sand the parts down and apply [Smooth-On XTC3D](https://adafru.it/nCJ) (<https://adafru.it/nCJ>) coating.

## Download, Modify, Remix

The design is open source, so you can make modifications and remix the design. It was designed in Autodesk Fusion 360. You can download the source file in your choice of file format (IGS, STEP, SAT, etc.)

<https://adafru.it/nqC>






<https://adafru.it/nqC>

## But I Don't Have a 3D Printer!

That's totally OK, you can still buy the parts and have them shipped to you! You can use a service like 3D Hubs to 3D print all of the parts for you. Just download all of the STL files from our Thingiverse page and upload them to their website.

### Select a color

Pick the color in which you want your Thing to be printed. All models are printed in biodegradable PLA plastic.  
Visit the [3D Hubs website](#) for more colors and materials.

 Black	 White	 Red	 Blue	 Green
--	--	--	---	---

### Your address

Shipping prices may differ based on availability. Orders are shipped within one week.

Country \*

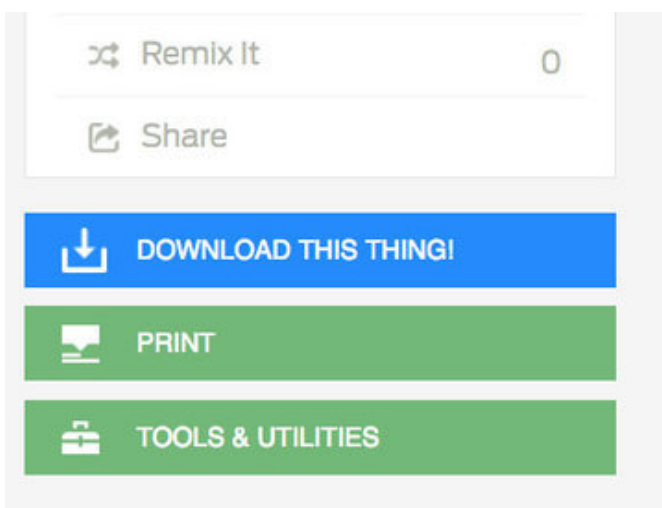
State \*

---

Startup costs: **\$5.00**  
Printing: **\$44.30**  
Shipping: Enter your address to get a shipping price.

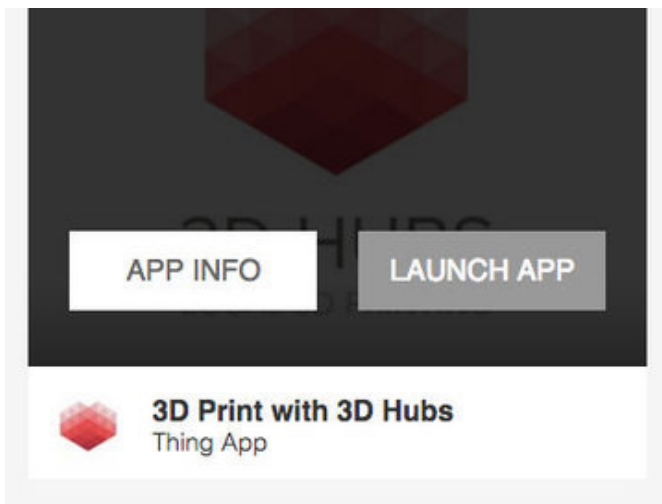
**Total price \$49.30**

No shipping address provided.



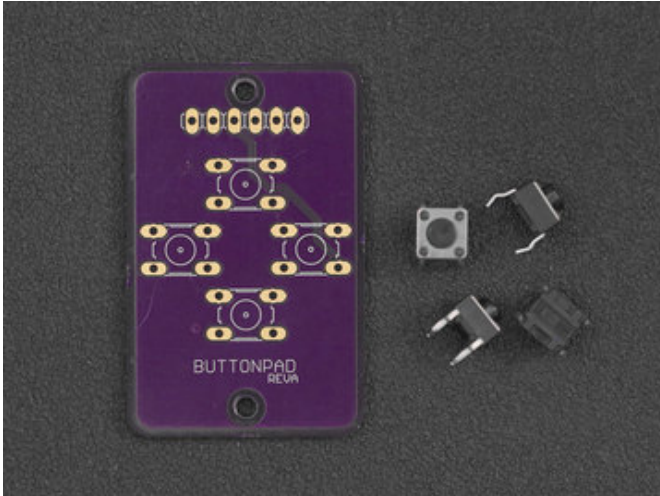
## Using 3D Hubs App on Thingiverse

The easiest way to do this is to use the "Print" button on the Thingiverse page. Then, click on the Launch App button to automatically load the STL files into 3D Hubs. From there, you can select colors, materials and enter your shipping address. A total price will let you know how much the parts will cost. A list of local hubs will appear and you can select which ever looks good to you. Hubs have different prices, ratings, reviews, and sample photos so you can narrow down your choice.



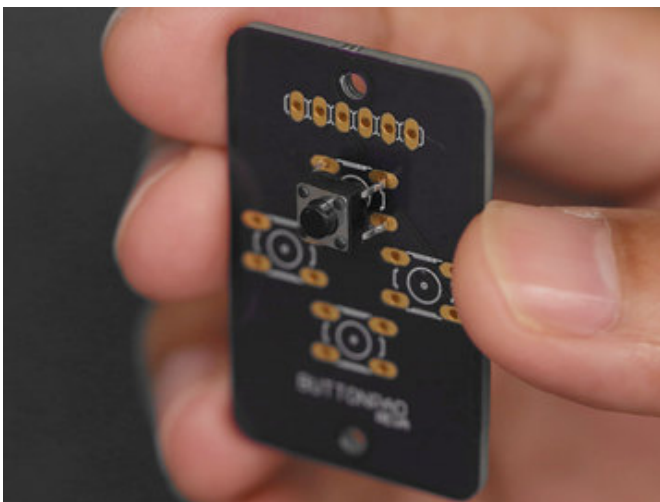
## Gamepad

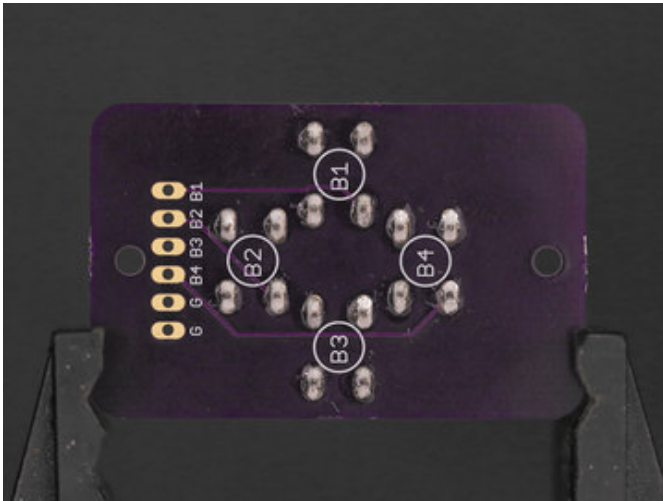
---



### Install Buttons on Gamepad PCBs

Let's start by setting up our buttons. Insert each button on the side of the PCB with the "BUTTON PAD" writing. Make sure the orientation of the legs on the buttons are positioned with the pins. Push them into the holes so they're all the way in, flush with the surface of the PCB.

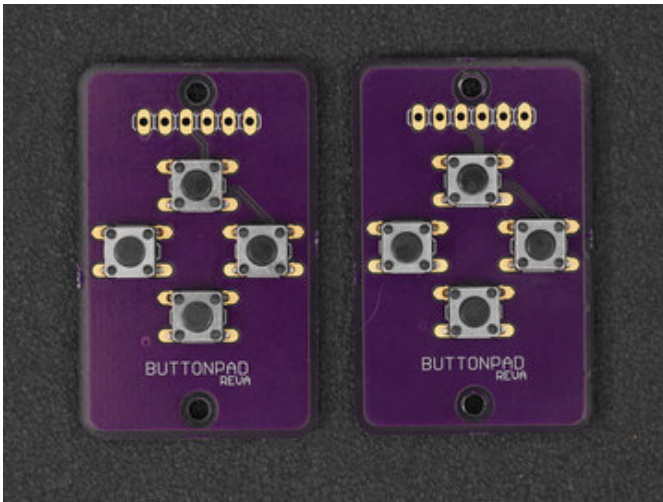




## Solder Buttons to Gamepad

Flip the PCB over and secure it to a Panavise Jr. or Helping Third Hands. Then, add solder to all of the pins.

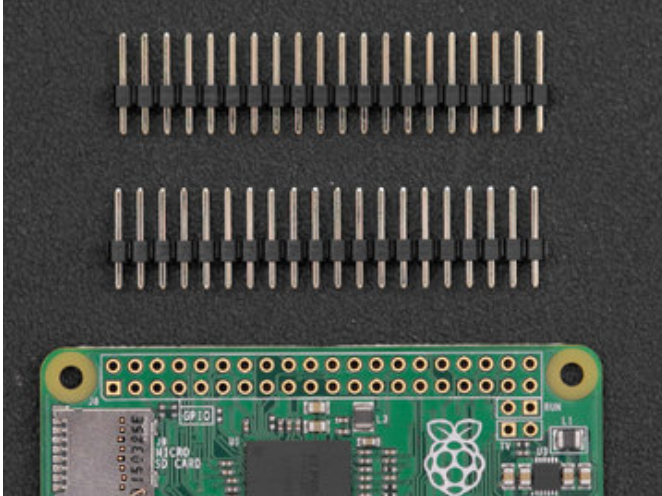
Repeat this procedure for the second gamepad PCB.





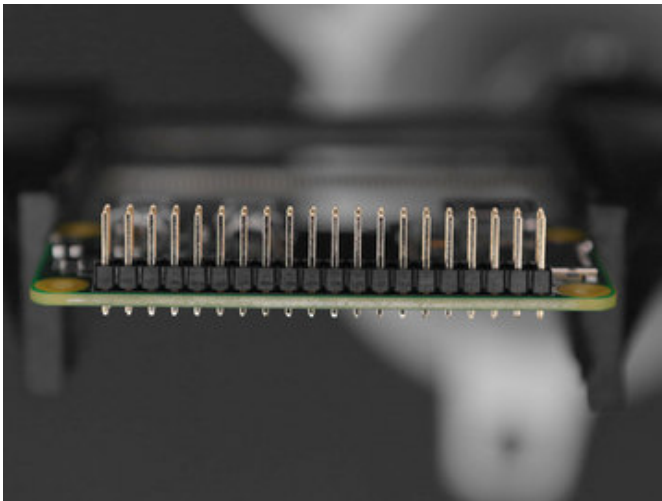
## Pi Zero

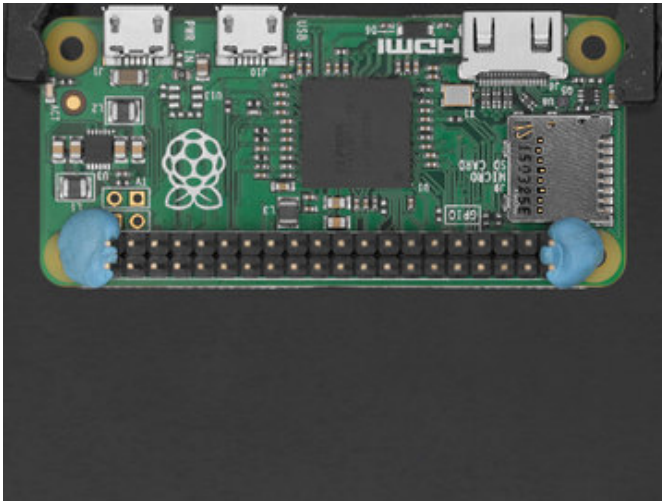
---



### Install Headers to Pi Zero

The Raspberry Pi Zero ships with no headers installed on the GPIO. Insert the 2x20 (or two 1x20) headers into the GPIO like shown the photo. The black plastic bit should be on top of the PCB.

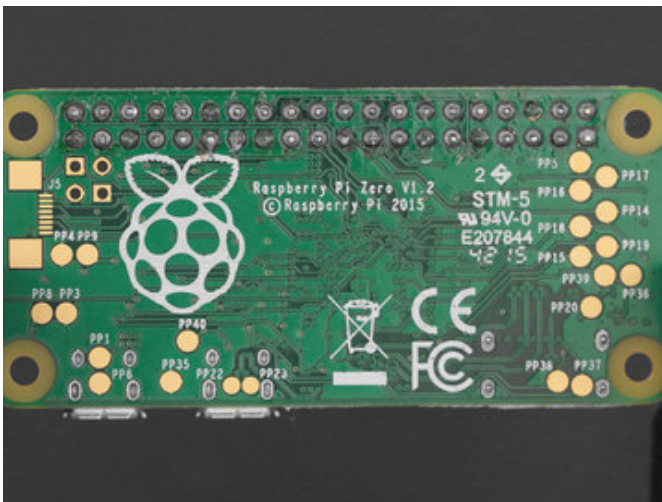
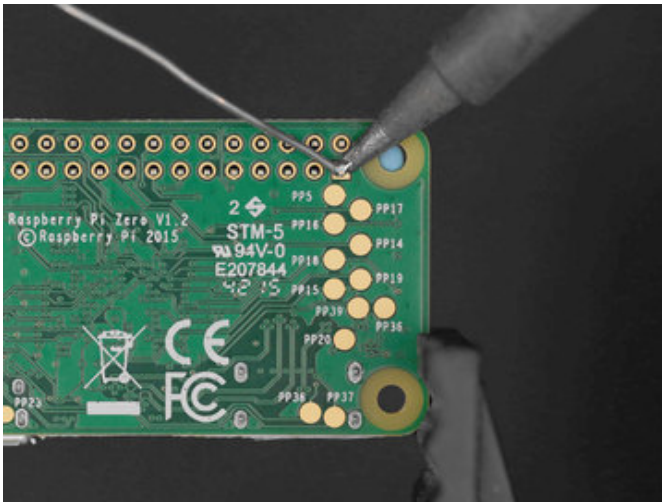




## Tack Headers to PCB

Use tape or mounting tack to hold the header onto the PCB. Then, secure the PCB to a pair of Helping Third Hands or a Panavise Jr. This will make it easier to solder.

Flip the PCB over and solder all of the pins to the GPIO.



## Soldered Headers

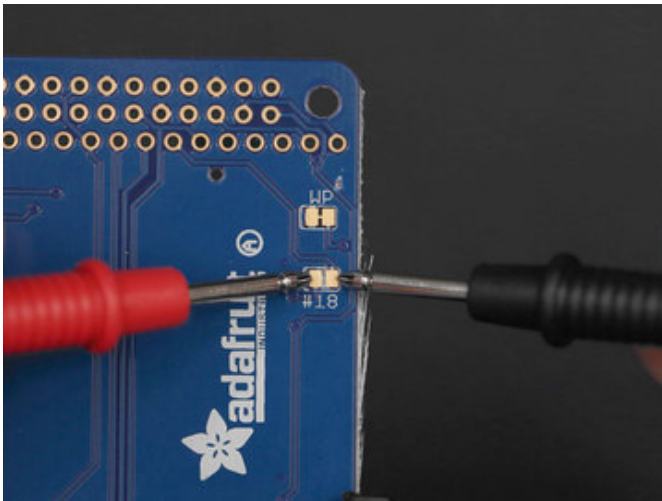
Inspect each connection and ensure there are no cold solder joints. If you find it difficult to apply solder to the pins, try tinning the tip of the soldering iron.



## Cut Trace on Pin #18

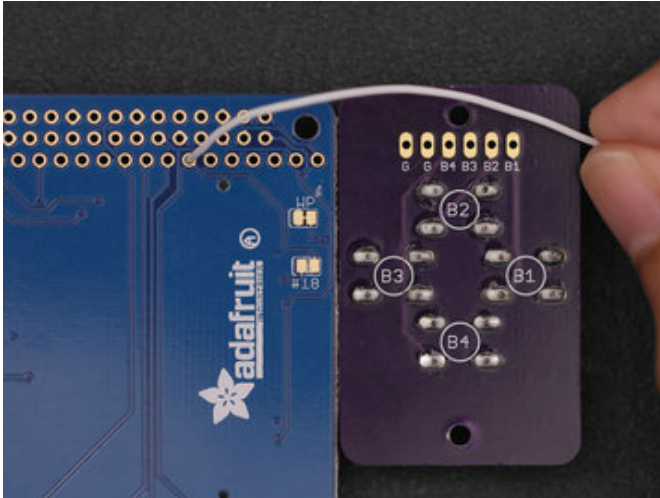
In order to properly use Pin #18 (left button) we'll need to cut the trace on the back of the PiTFT. Using a sharp hobby knife, you can score in between the pads to cut the trace.

I recommend testing the connection with a multimeter using the continuity mode. Press the probes onto the pads. If the multimeter beeps, then the trace isn't completely cut. It's best to ensure the trace is cut now then later in the build.



## Gamepad To PiTFT

Gamepad PCB Update! Please reference the wiring diagram below with blue PCB. Purple PCBs are slightly different (Pins B3 and B4 have been flipped).



### Measure Gamepad Wires

Grab one of the gamepads and position it close to the PiTFT (In the photo, we're doing the DPAD first). We need to measure a piece of wire to see how long it needs to be for the first button (The left button on the DPAD). I recommend making it slightly longer so we have some slack and wiggle room.

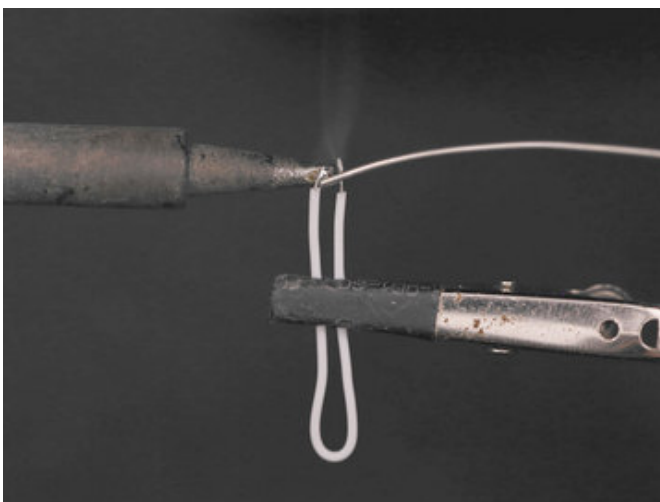
In the photo, I have inserted a long wire into Pin #18 on the back of the PiTFT. By flipping the PiTFT over, I can see which pin is labeled.

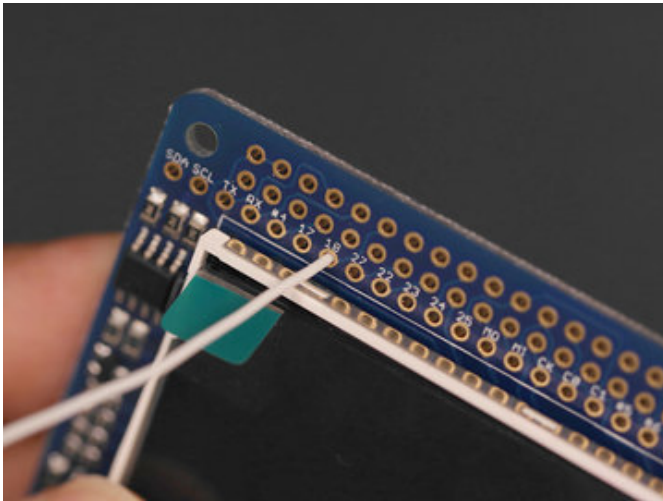
Once we've determined a good length for our first connection, cut the piece of wire.



### Strip & Tin Wires

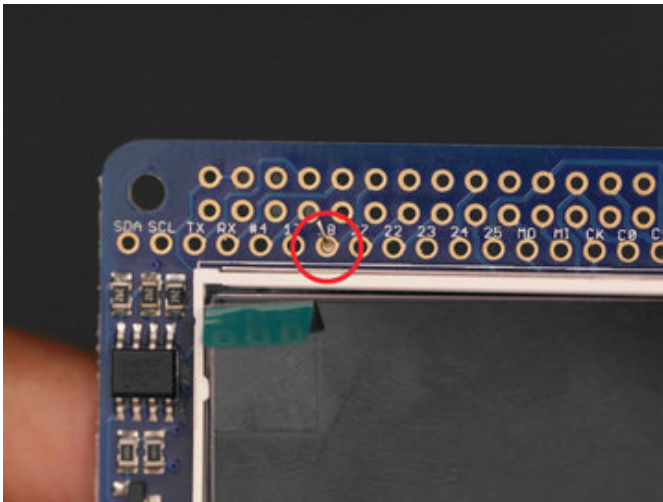
Use wire strippers to remove 5mm of insulation from each tip. Then, apply a bit of solder to the tips to tin them. This will prevent the stranded wires from fraying and make it easier to solder to the pads.





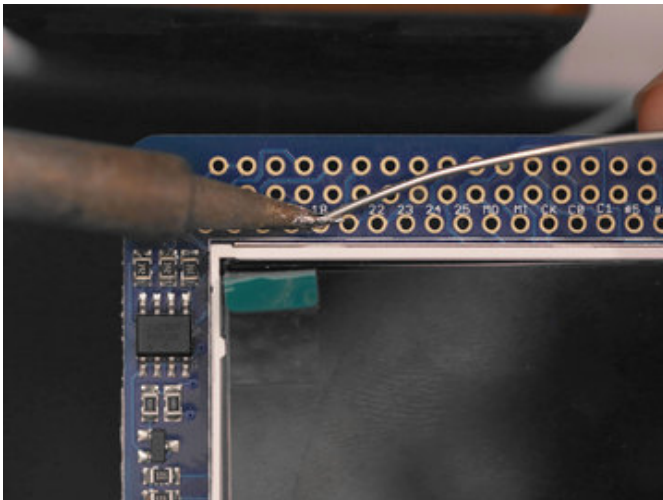
## Inserting Wires to PiTFT

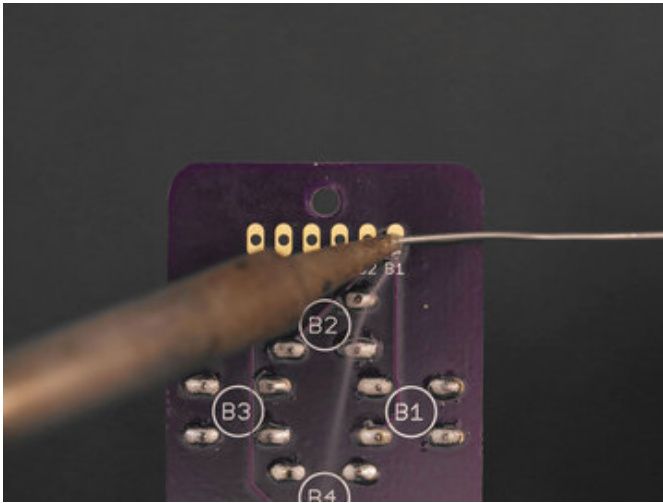
To make wiring a bit easier, I suggest inserting the wire into the corresponding pin (#18 for the left button in this case) and threading it through until the tip from the other end is touching the hole. Then, bend the tip at a right angle so it can be held in place.



## Solder Wire to PiTFT

Now that the wire is in place, you can solder the connection to secure it.

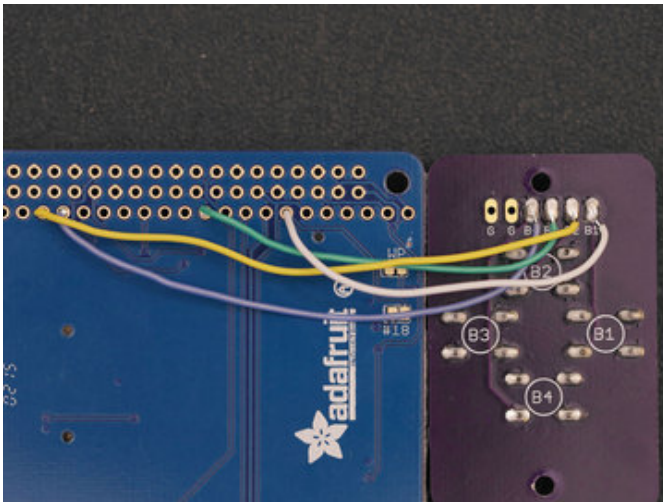
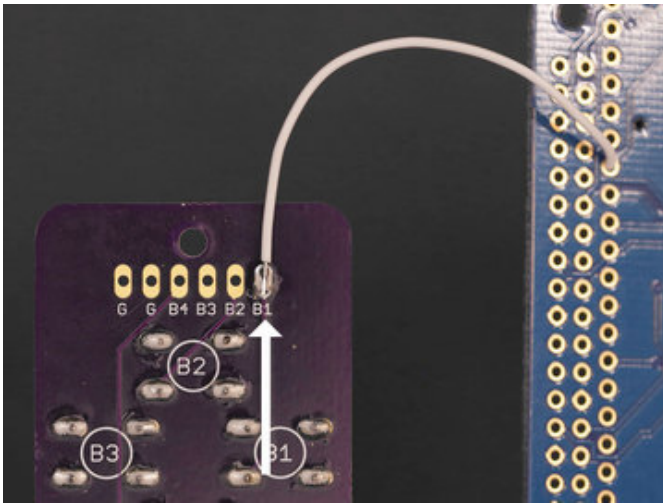




## Solder Wire to Gamepad

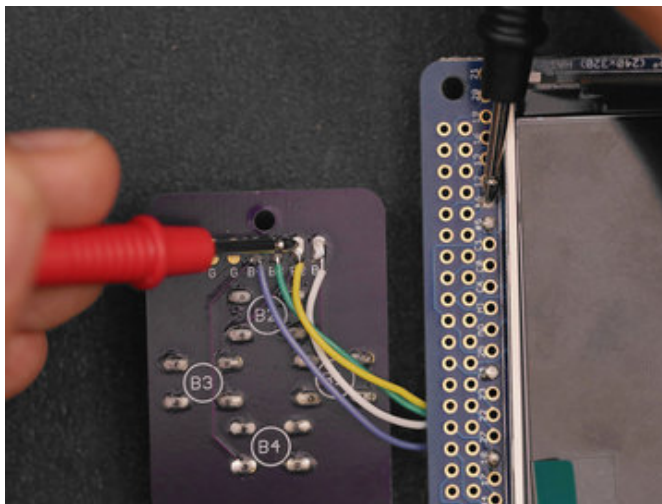
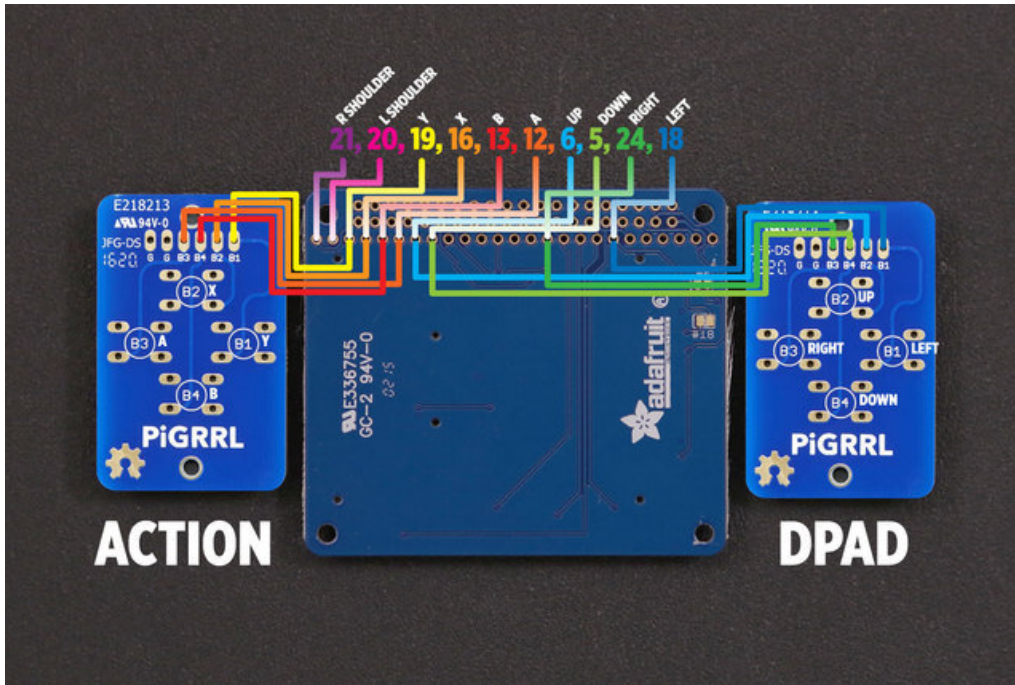
Now we can apply some solder to the header pins on the Gamepad PCB. I suggest tinning the pads on the header of the PCB before soldering wires to them so it's easier.

Although the photo shows the wire orientated from the top, I recommend positioning the wire from the bottom. This will make securing the PCB to the 3D print case easier.



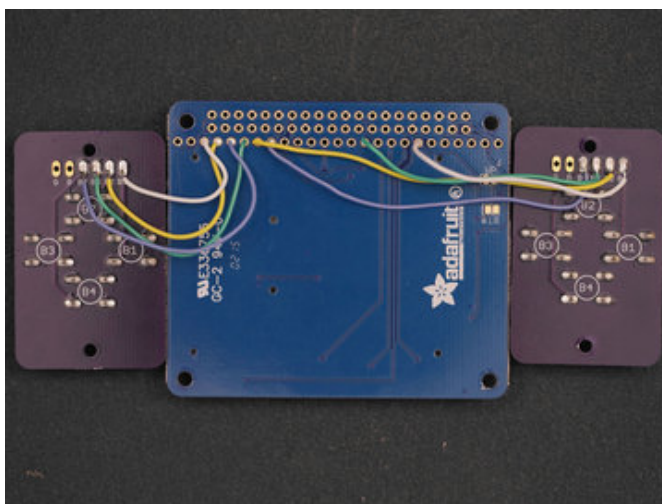
## Wired Gamepad

Repeat this process for the rest of the buttons, up, down and right. Each button requires a different length of wire. I recommend referencing the wiring diagram closely and double checking each wire is in the correct position. Also, try doing one wire at a time to prevent any confusion.



### Test Gamepad Connections

It's a good idea to test your connections! Using a multimeter in continuity mode, you can probe the header pins on the PiTFT and gamepad to see if the connections have continuity - You'll hear a "BEEP" sound if the connection is solid.



### Wired Gamepads

With patience and persistence, you'll end up with two gamepads wired up to the PiTFT.

This part of the build is the most time consuming, so try not to rush it. Once complete, feel free to take a break :-)

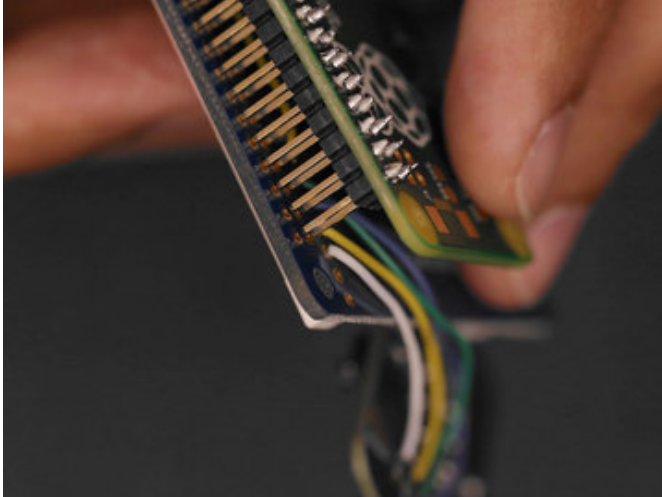
In the next page, we'll solder the Pi Zero to the PiTFT and sort out the other components.





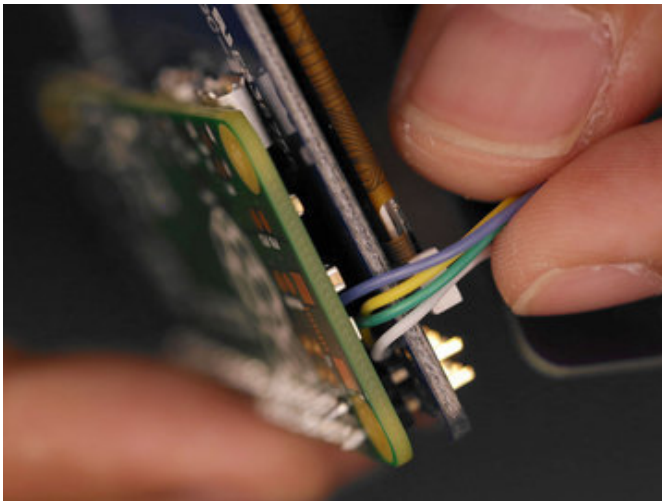
## PiTFT To Pi Zero

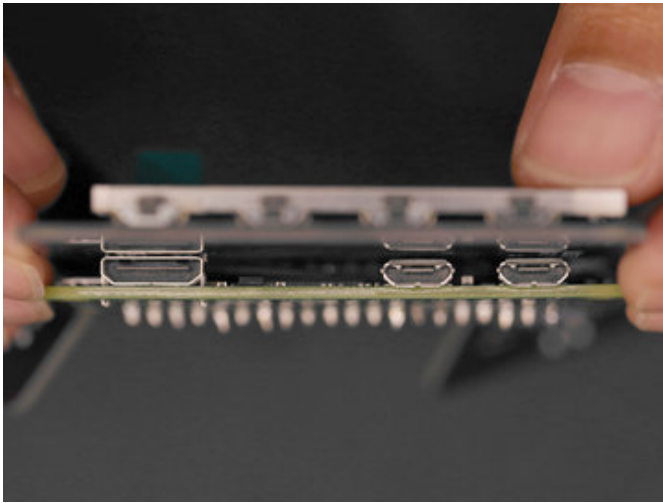
PLEASE BE AWARE - THIS IS NOT AN EASY/BEGINNER PROJECT! This project is for experienced makers, with successful soldering and Raspberry Pi projects. We DO NOT guarantee you will be able to build this project successfully!



### Install PiTFT to Pi Zero

Now we can insert the PiTFT to the Pi Zero. You'll want to group the wires from the gamepad so they aren't kinked. Slowly guide the pins from the GPIO header on the Pi Zero and seat them through the pins on the PiTFT.

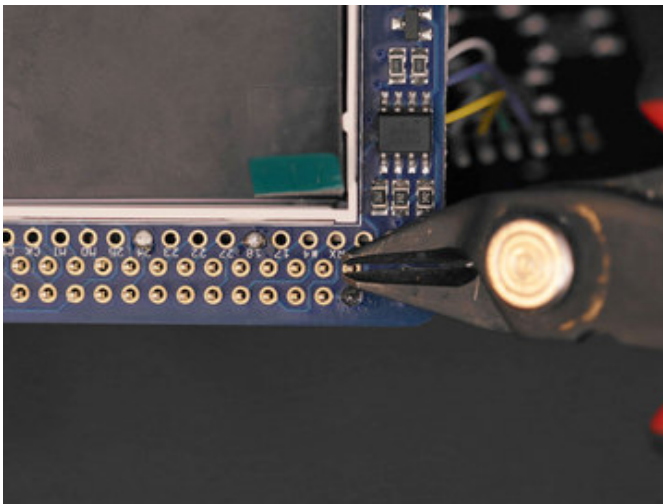




## Connecting PiTFT to Pi Zero

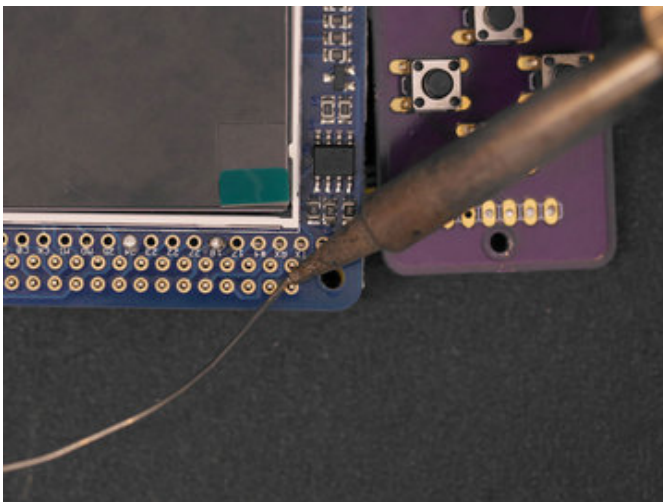
Look at the HDMI and USB ports on the Pi and make sure they're flush with the back of the PiTFT. The wires from the gamepad should be behind the ports.

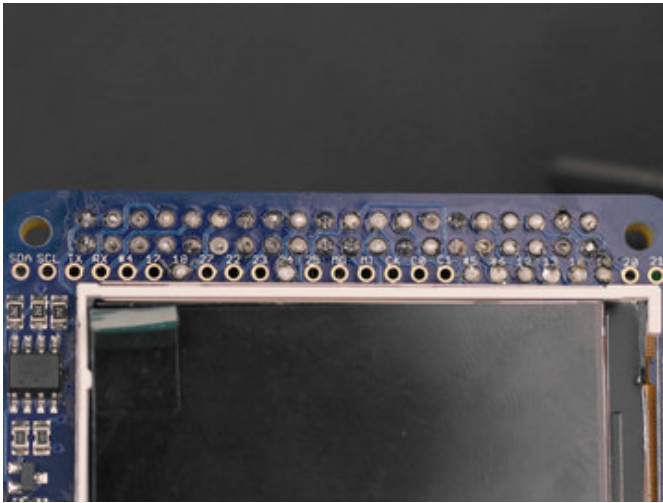
When you seat the GPIO header from the Pi to the PiTFT, try not to fully press them together. If you do, you'll see the Pi Zero PCB will be slightly angled. Try to keep the two PCBs as straight as possible.



## Solder PiTFT to Pi Zero

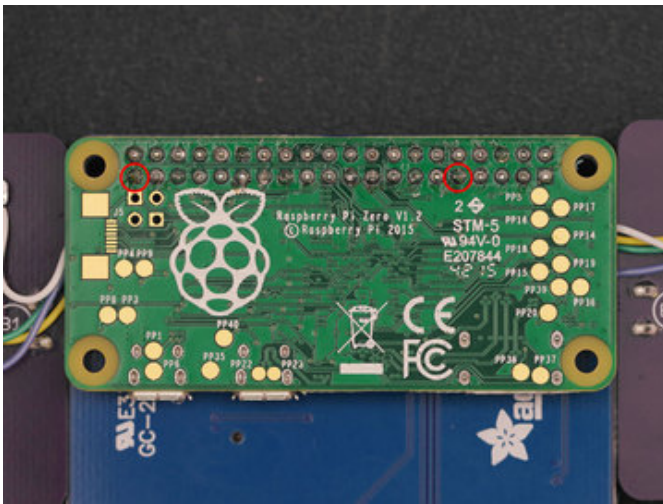
Trim the pins from the header of the Pi Zero so they're slightly taller than the pins of the PiTFT. This will make it easier to apply solder. Once they're all trimmed down, go ahead and solder all of the pins! Take your time and be sure the solder joints are solid.





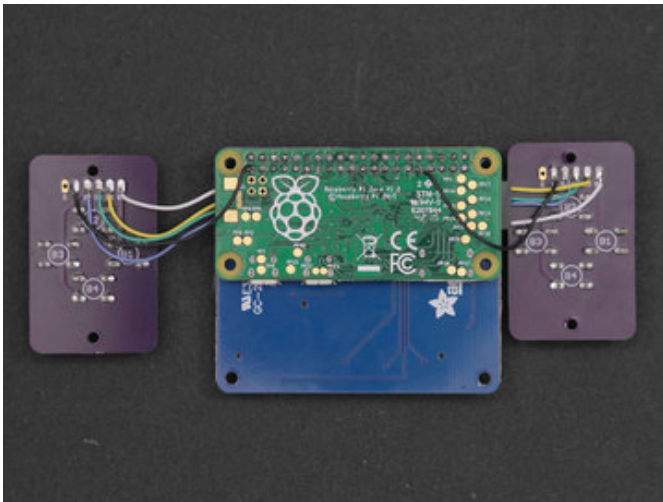
## Soldered PiTFT to Pi Zero

Double check each pin and ensure they're nicely soldered.



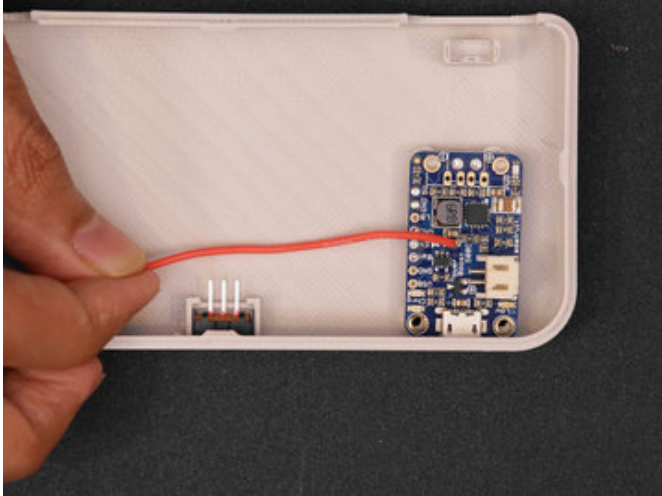
## Gamepad Ground Connections

Now is a good time to add ground connections for the two gamepad PCBs. We can add these to the back of the Pi Zero. Measure and cut two pieces of wires. Solder them to the available ground pads on the gamepad PCB. Then, connect those to the back of the Pi.



## Power

---

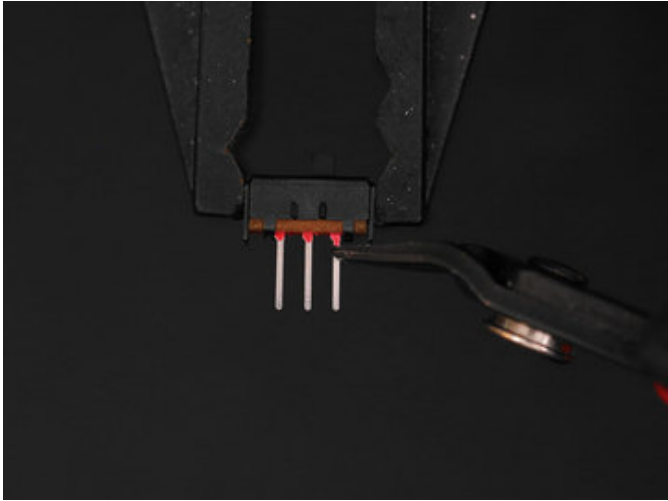


### PowerBoost Connections

Grab the bottom half of the enclosure and sit the PowerBoost 1000C on top of the four standoffs near the bottom left. Then, insert the slide switch into the bottom center. These two components should just snap fit into the case.

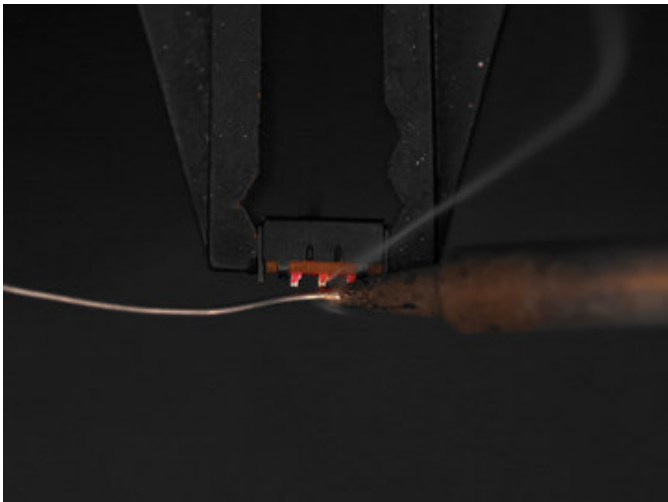
Next, we need to measure two 26AWG wires for connecting the slide switch. They'll be connected to the **EN** and **GND** pins on the PowerBoost 1000C. Once you've sorted out a good length for them, strip and tin the ends of each wire.

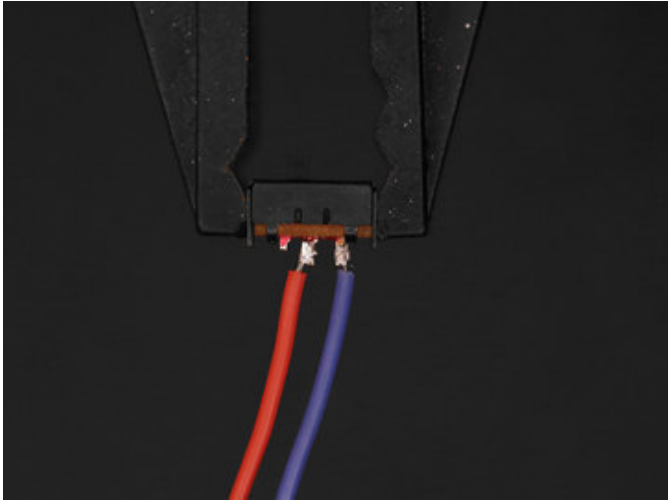
Don't use a PowerBoost 500C! The photo shows the incorrect component. You really want the power of the 1000C



## Prep Slide Switch

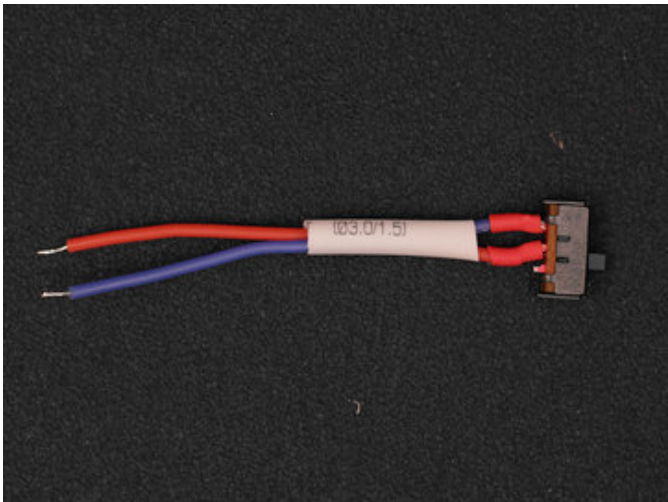
Now that we have our wires sorted out, we can prep the slide switch. First, we don't need three leads, so remove one using a pair of flush snips (it can be any of the leads on the far left or right, but not the middle). Then, trim the remaining two leads to about half their length. Now we can tin them by applying some solder.

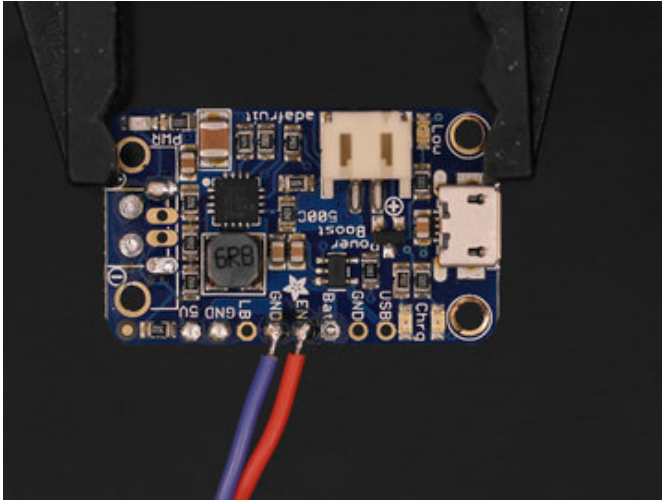




## Solder Wires to Slide Switch

With the leads tinned, solder both wires to the two leads. Pieces of shrink tubing will insulate the exposed connections. Simply apply heat to "shrink" the tubing.

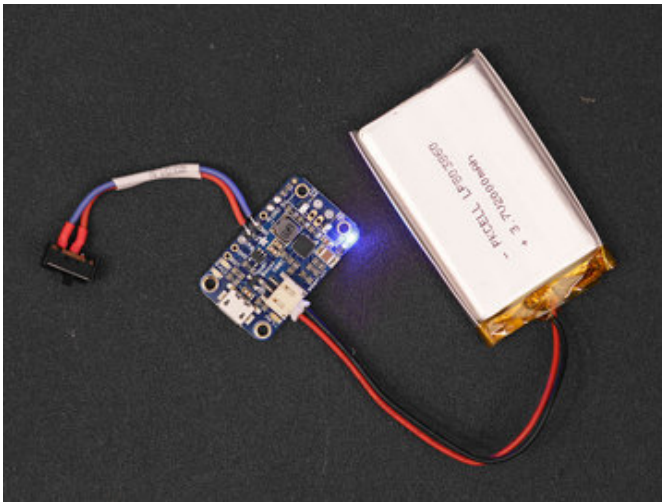


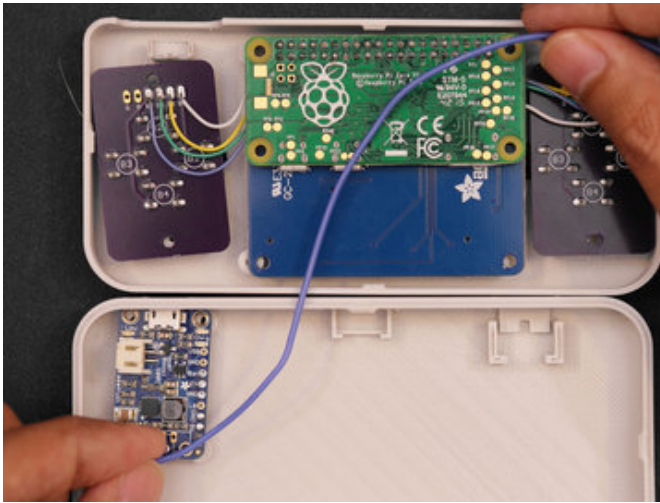


### Connect Switch to PowerBoost 1000C

Solder one wire to the **EN** pin on the PowerBoost 1000C and the other to **GND**.

Now you can plug in the JST connector from the 2000mAh battery into the JST port on the PowerBoost 1000C. The switch will allow you to power it on and off. The blue LED will light up indicating it's powered on.

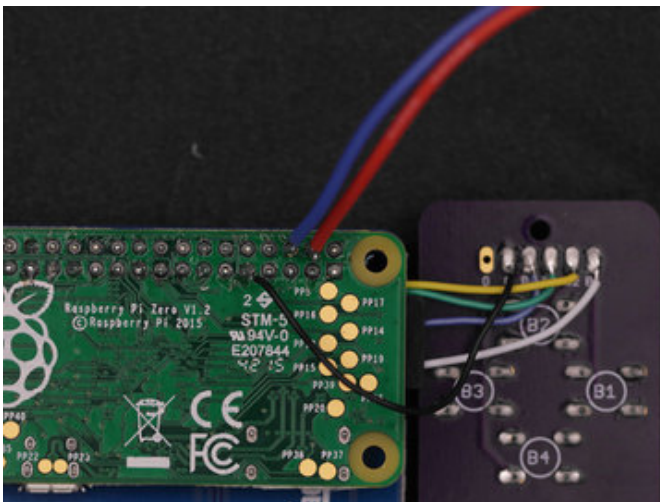
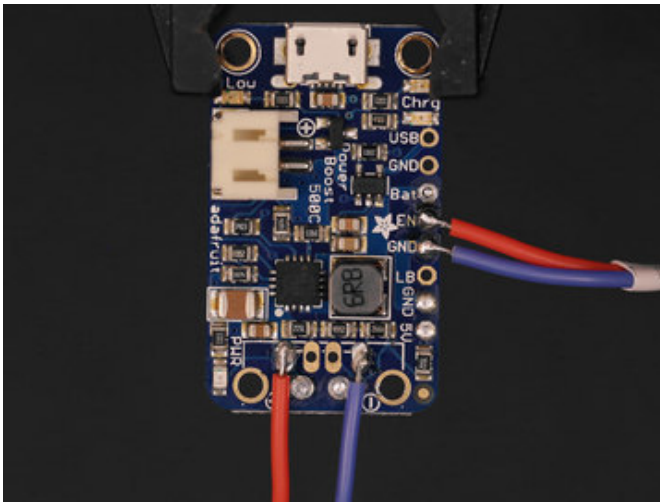




## PowerBoost Wires

Next, we need to sort out a set of two 26AWG wires for connecting the PowerBoost 1000C to the Raspberry Pi Zero. We can do this like we did for the slide switch.

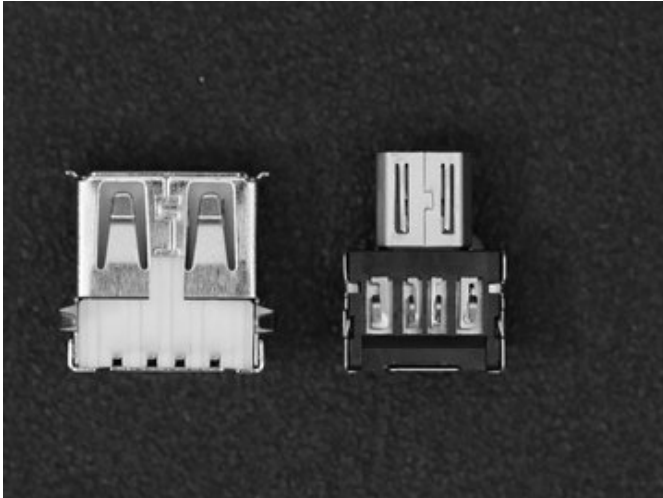
Once we have a good length, cut, strip and tin the wires. Then, solder them to the positive+ and negative+ pads on the PowerBoost 1000C.



## Connect PowerBoost to Raspberry Pi

Now we can solder the positive and negative connections from the PowerBoost 1000C to the GPIO pins on the back of the Raspberry Pi Zero. Reference the photo for the right pins. Red is positive, and blue is negative.

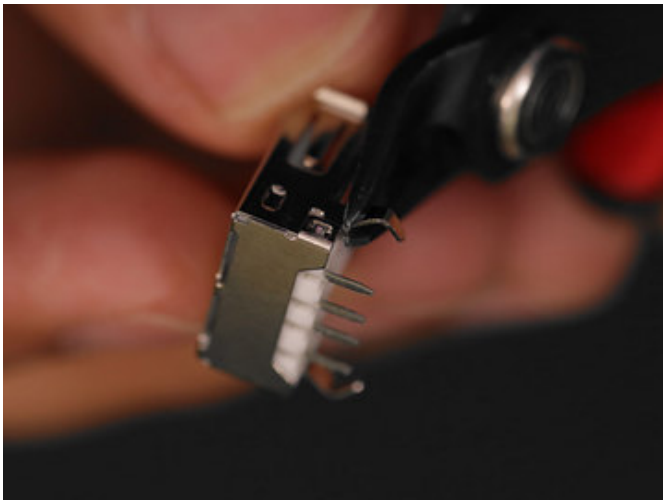




### USB Adapters

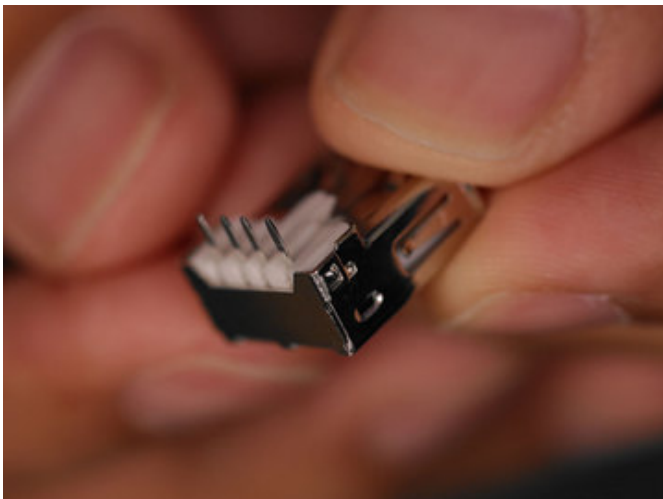
OK, now it's time to work on the USB adapters. Grab the USB OTG adapter and the USB female port (this came with the PowerBoost 1000C). We need to create a set of 4 wires that will connect these two components together.

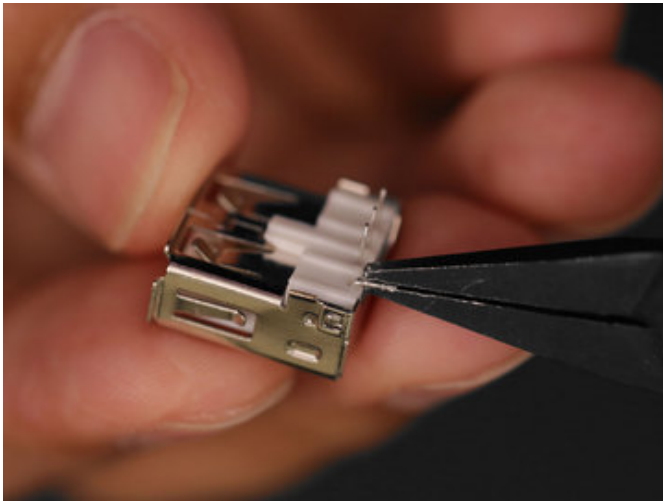
This is necessary for connecting peripherals to the Raspberry Pi such as WiFi adapter, USB Audio adapter and more.



### Remove Legs from USB Female Jack

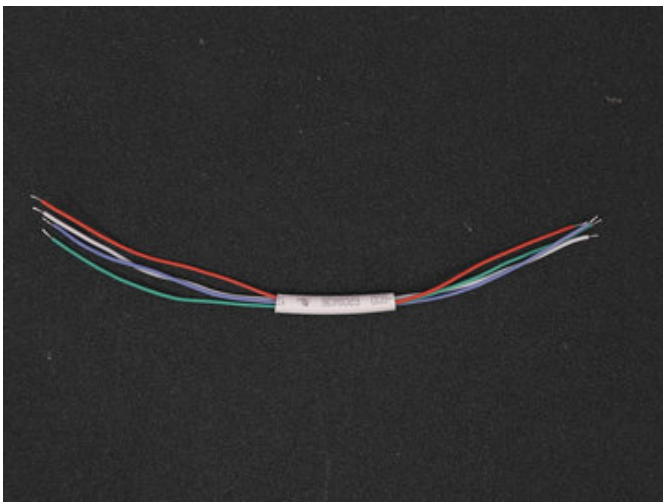
Use a pair of flush snips to cut off the legs from the female USB jack. These prevent the port from being able to fit into the case. They're normally used to secure the jack to pins on a PCB, but since we're mounting to it the case, we don't need them.





### Prep USB Pins

Next, we need to bend the four connectors on the USB female jack so they're right-angled. As they are now, it wouldn't fit in the case. Use a pair of flush pliers to carefully bend each connector. You'll want to be cautious while doing this. The connectors **MUST NOT** touch the metal housing - if it does, it'll short out the connection. You can optional remove the metal back piece from the USB jack.



### USB Wires

We'll need to make a set of four wires for connecting our USB adapters together. I found 13cm to be a good length. Like we've done previously, strip and tin the ends of each wire. A piece of heat shrink tubing will keep these four wires together.

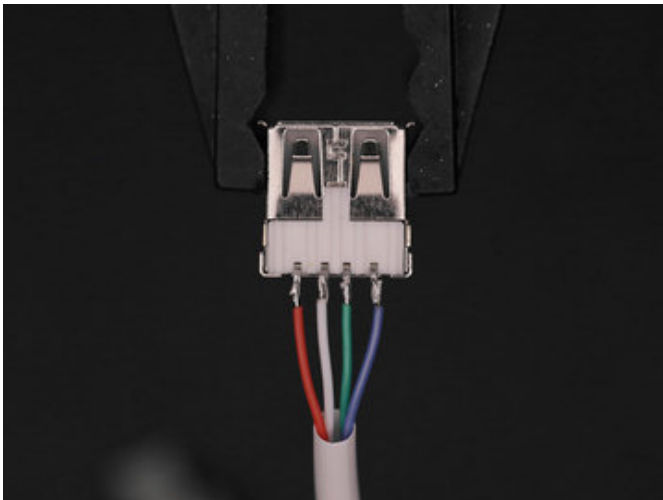


### Connect Wires to USB OTG Adapter

Most microUSB cables follow the same color coded wiring.

- Black / Blue – GND
- Red – VCC 5V
- Green – USB Data -
- White – USB Data +

Follow the photo to reference which wire to connect to the correct pin. Tinning the four pins on the USB OTG adapter will make it easier to solder the wires to them.



### Connect Wires to Female USB Jack

Following a similar method, connect the four wires to the pins on the Female USB jack. Note the colors are reversed. That's because we have the USB jack flipped. Make sure you familiarize yourself with the top and bottom of the USB jack. The bottom has the white plastic bit.

## Shoulder Buttons

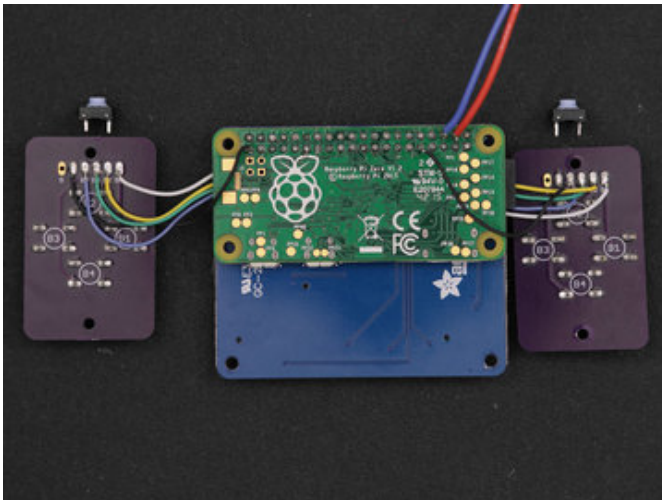
---



### Shoulder Buttons

Now it's time to work on the shoulder buttons. The two buttons I used in this project are 2-terminal versions that come with extra long legs. The ones included in the kit have four legs. You only need to use two of them.

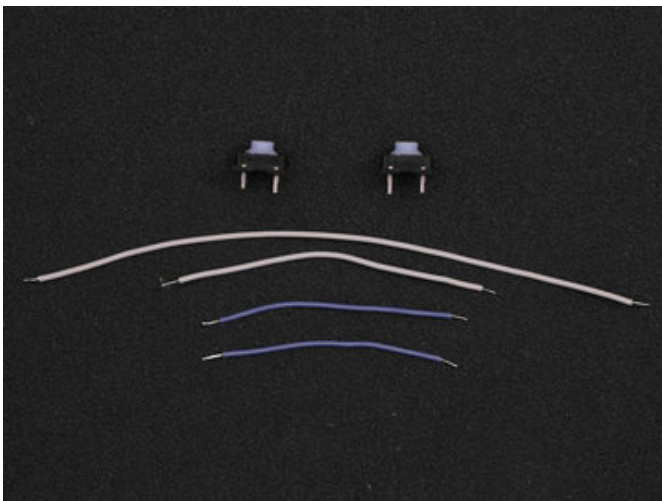
---



### Measure Shoulder Button Wires

Layout the two buttons and plan out the lengths of each wire. The L and R shoulder buttons will need to be connected to the last two remaining pins on PiTFT (#20 and #21) and the remaining ground pins on the Gamepad PCBs.

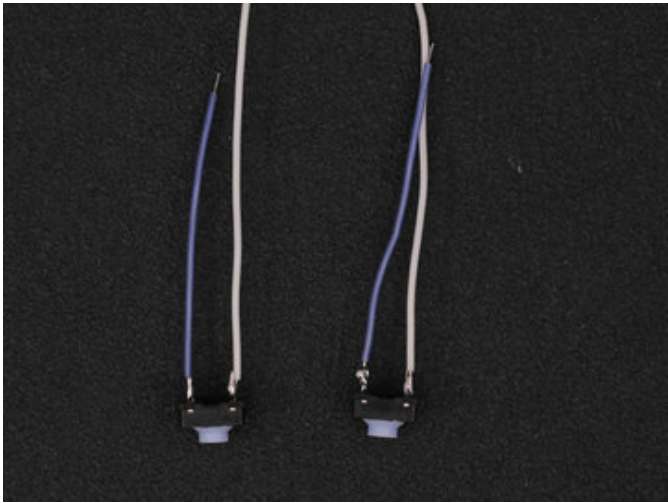
---



### Shoulder Button Wires

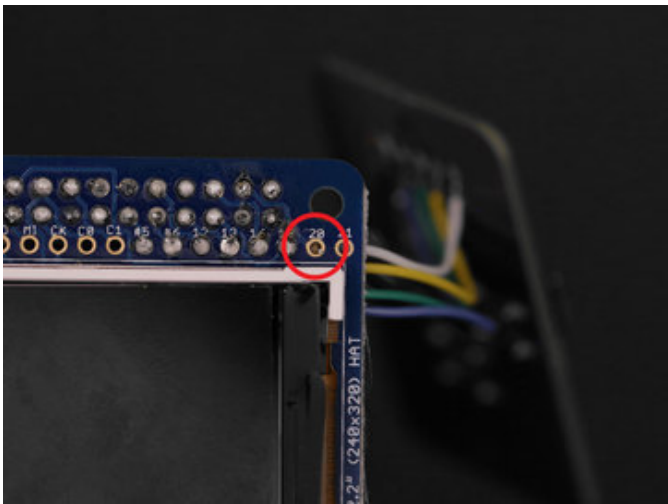
Once you've measured each wire, we'll prep them by stripping and tinning the tips. You'll also want to tin the leads on each button.

---



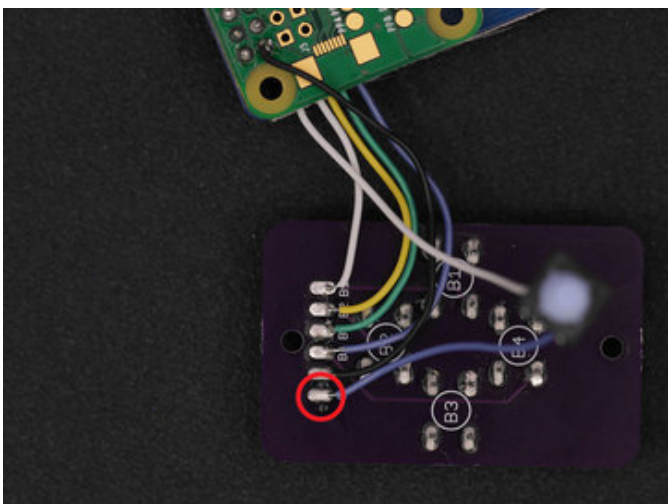
### Connect Wires to Buttons

Solder the wires to the terminals on each button.



### Connect Wires to PiTFT

Connecting the left shoulder button to the PiTFT is a little tricky because it needs to be threaded through the PiTFT and Raspberry Pi. Using the similar method we did when connecting the gamepad PCB wires to the PiTFT, the tinned tip of the wire should be able to poke through pin #20. Once it is, solder it in place. The right shoulder button is much easier to connect since pin #21 is close to the edge.

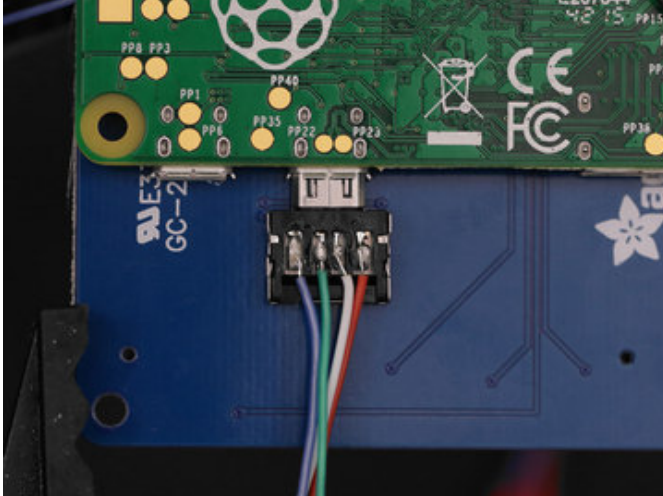


### Ground Shoulder Buttons

With the two buttons connected to the PiTFT, solder the remaining connection to the ground pins on the gamepad PCB.

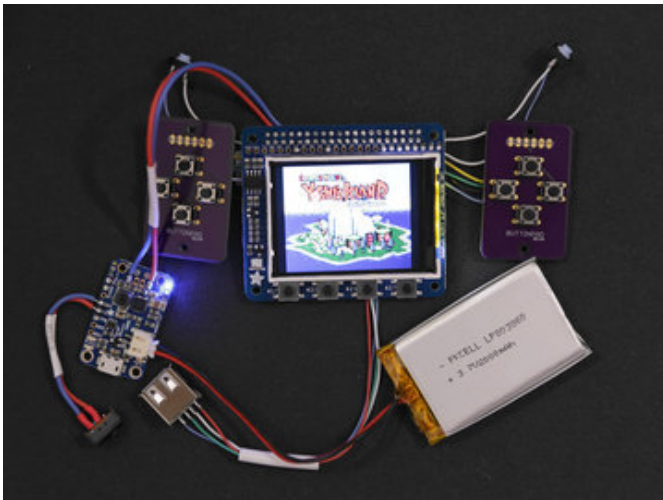
## Mounting Components

---



### Connect USB OTG Adapter

Before mounting the components, go ahead and plug in the micro USB OTG adapter to the Raspberry Pi.



### Test Final Circuit

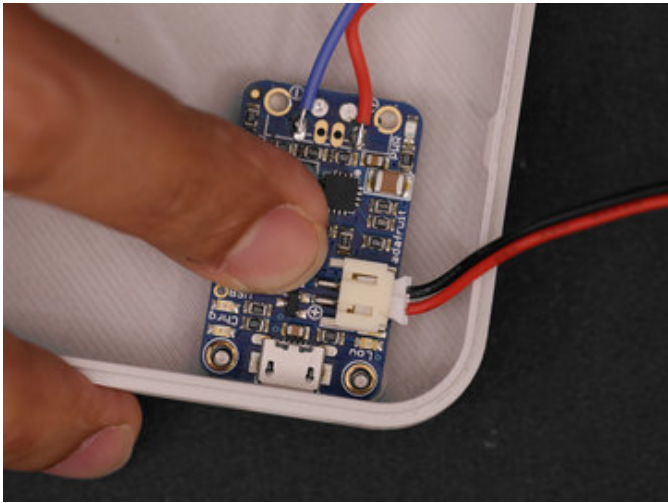
OK, now is a great time to test the circuit. Make sure the PiGRRL Zero IMG has been properly burnt to the SD card and inserted into the Raspberry Pi. "Flip" the switch and the PiTFT should turn on. The Pi will take about a minute to boot up.

The controls are premapped so you can test the gamepad PCBs. Try a play test with one of the games in the PORTS section. Use the "A" button to select. To exit a game, hold down the two middle buttons (select and start) on the PiTFT.

## Verify Test

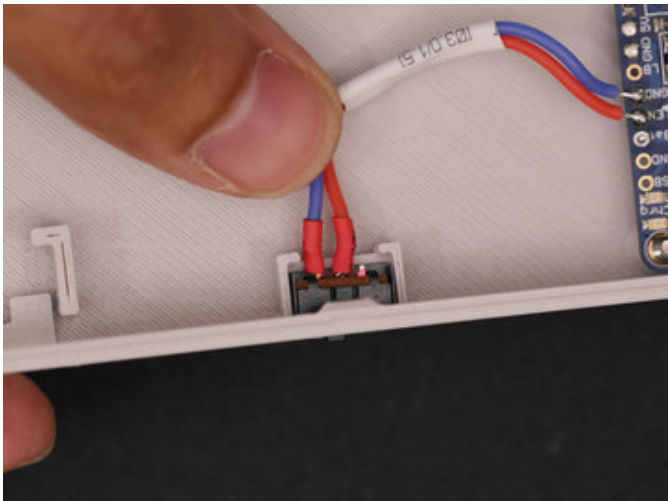
If you find something off or doesn't work as expected, feel free to post your questions or concerns in the [Adafruit Forums \(https://adafruit.com/forums\)](https://adafruit.com/forums).

If everything works as expected, feel free to move onto the final steps!



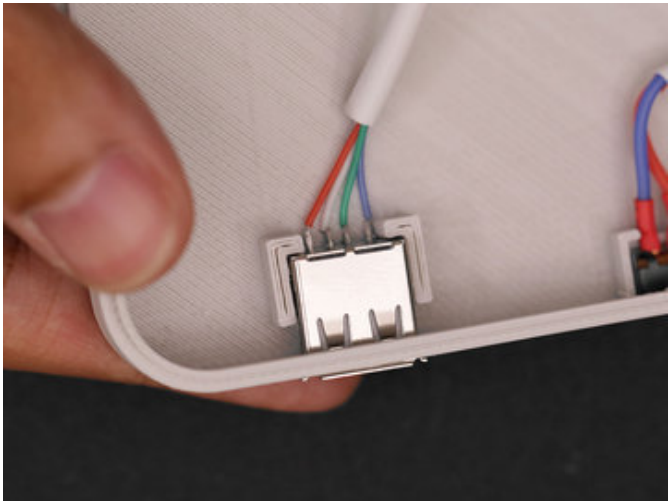
### Mount PowerBoost 1000C

Let's start with the PowerBoost. This is mounted to the bottom half of the enclosure, near the bottom right. The microUSB port should be pointing towards the opening. Simply lay the PCB over the standoffs and press it down so the pegs are seated into the mounting holes.



### Mount Slide Switch

The slide switch can be installed into the bottom half of the case by inserting it at an angle.

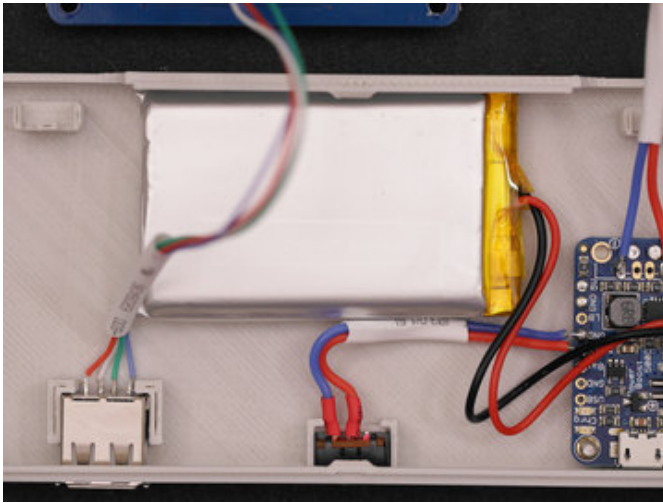


## Mount Female USB Jack

Similar to the slide switch, you'll need to insert the female USB jack at an angle. A slight amount of force is required to snap fit the housing of the USB jack into the opening.

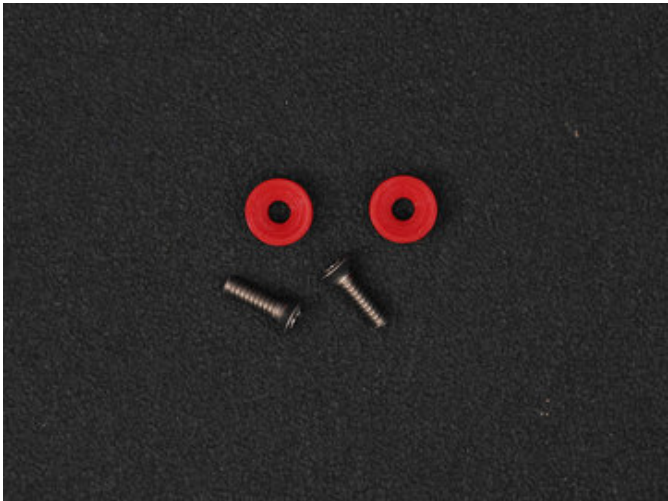






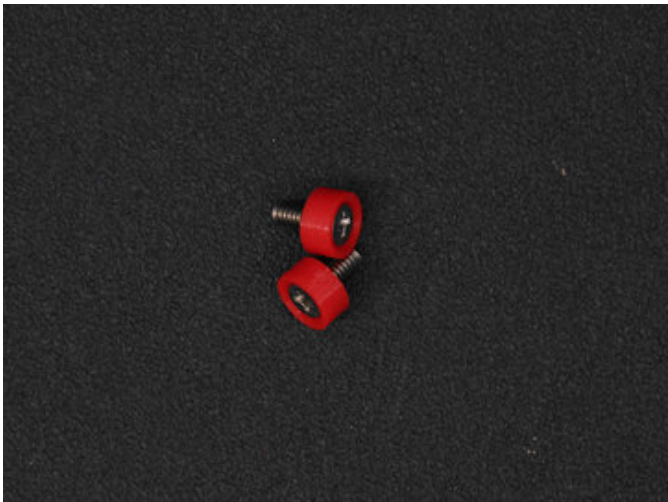
## Install Battery

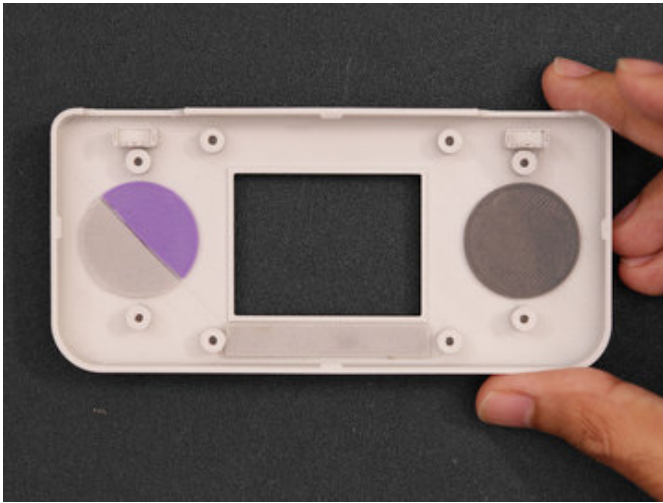
There isn't a dedicated "holder" for the battery, but a good location for the battery is near the top center. I suggest using either double-sided tape or mounting tack to keep the battery in place.



## Screws and Washers

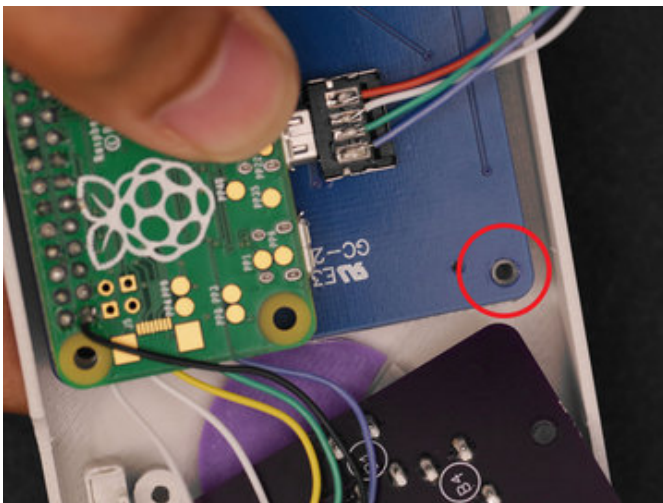
We'll need two #4-40 3/8 flat Phillips machine screws to install the PiTFT to the case. 3D printed washers will help prevent the screws from poking through the enclosure. Fasten each screw into these washers.





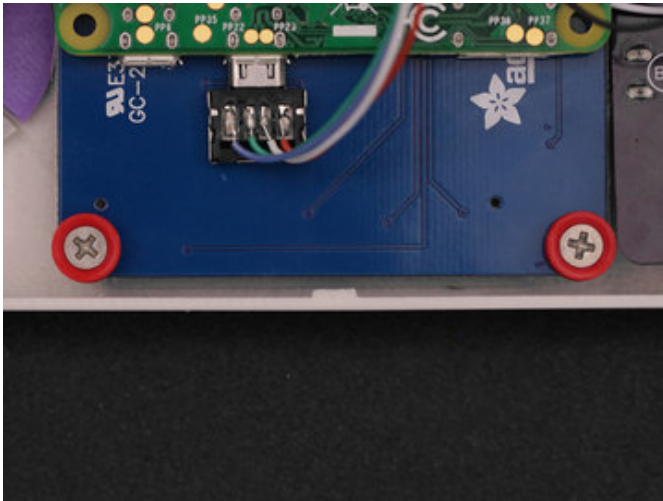
### Install Actuators

Grab the top half of the enclosure and insert the DPad, action buttons and 4-grouped menu buttons.



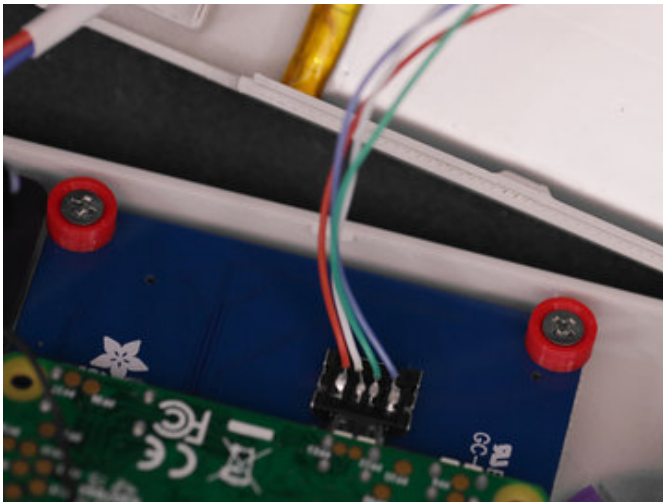
### Install PiTFT to Case

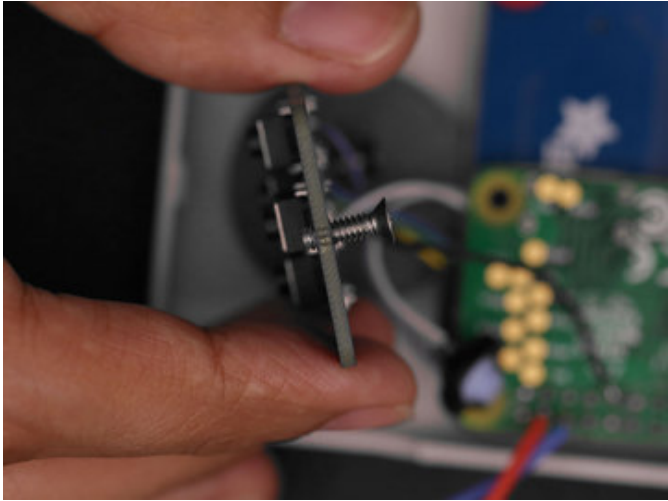
Position the full circuit into the top half of the case. Line up the mounting holes of the PiTFT with the standoffs in the center of the case.



## Fasten Screws

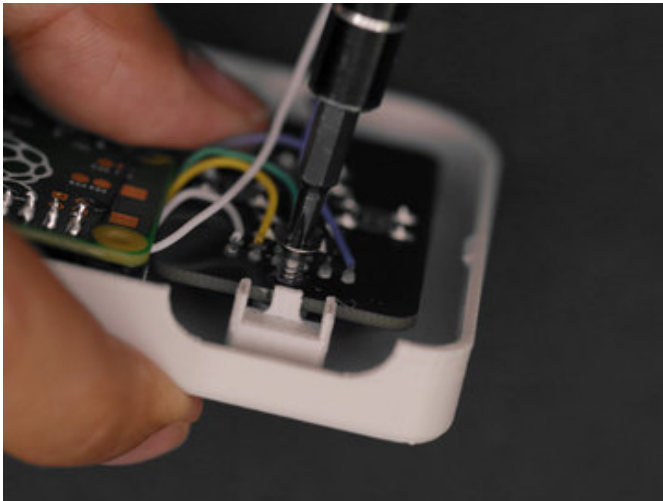
Insert one machine screw into one of the bottom mounting holes of the PiTFT. Hold the PCB down to the case while fastening the screw through the PCB and standoff until the washer is flush with the surface. Repeat for the second machine screw.





### Gamepad PCB Mounting Holes

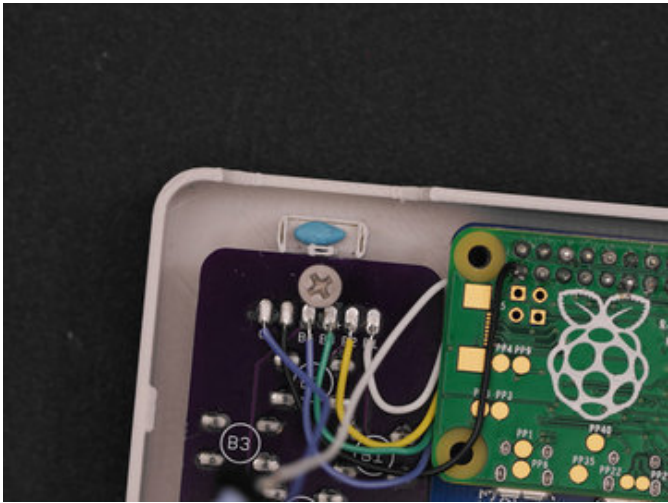
There's two mounting holes on each gamepad PCB needs to be tapped. Using a #4-40 machine screw, you can create the threads by slowly fastening through the hole. I suggest doing this before mounting the PCB to the stand offs on the case - It'll make it easier.



### Mount Gamepad PCBs

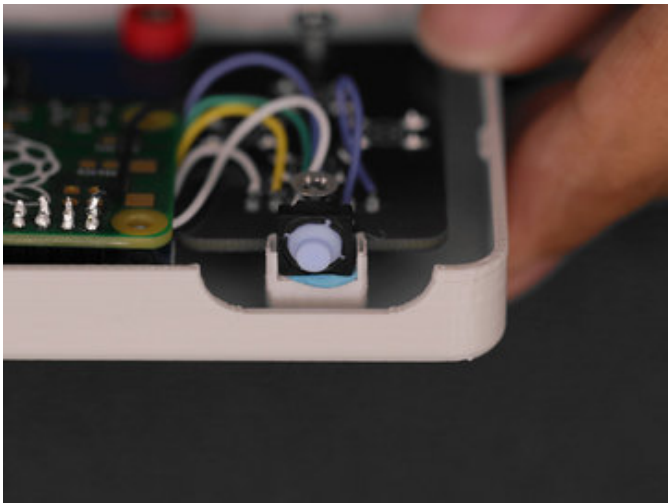
With the mounting holes tapped, position the PCB over the standoffs on the case and hold it in place. Then, fasten screws through the mounting holes. Do not fully fasten the screws into the standoffs – They only need to be fastened half way. The screw may poke through the case if it's fully fasten through.

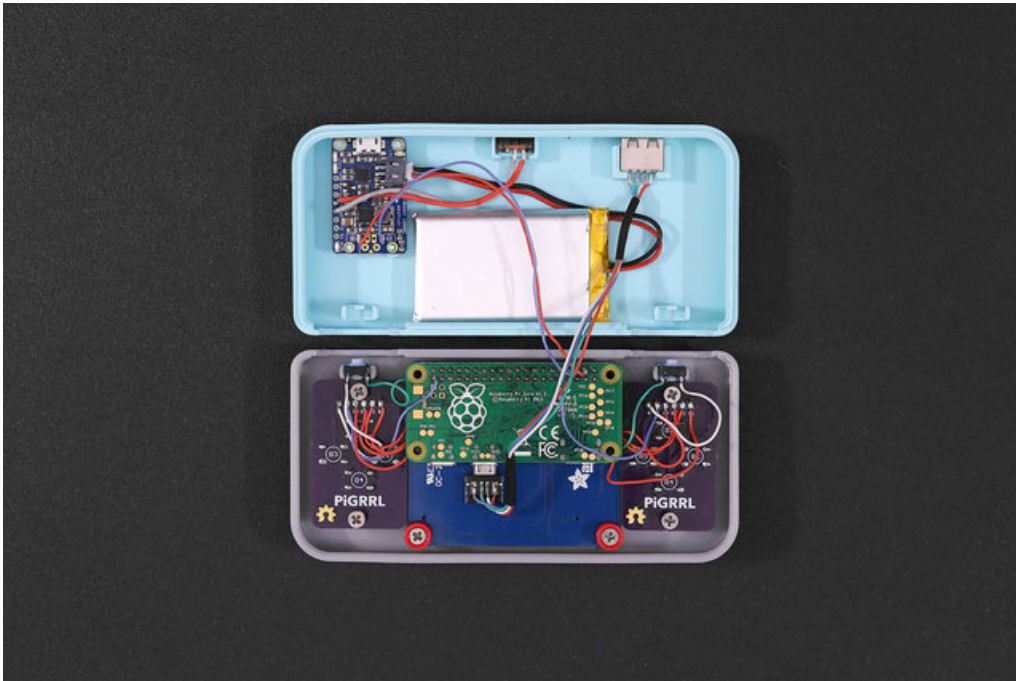




## Mount Shoulder Buttons

A small piece of mounting tack will keep the shoulder buttons in place. Near the top of the gamepad standoffs is a holder for the shoulder buttons. Insert the shoulder button into the holder with the actuator pointing towards the top.

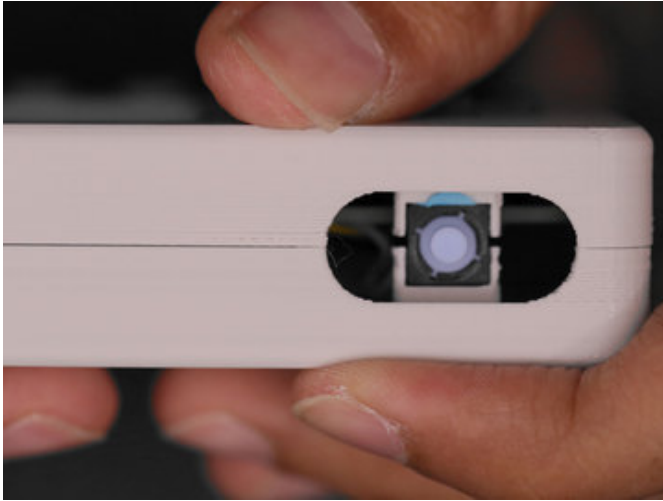




## Final Circuitry Check

Now is a good time to check all of your work. Are all of the wires connected? Are all the components secured? If you find some of the wires are too long, it's fine to trim them short.





### Close Case

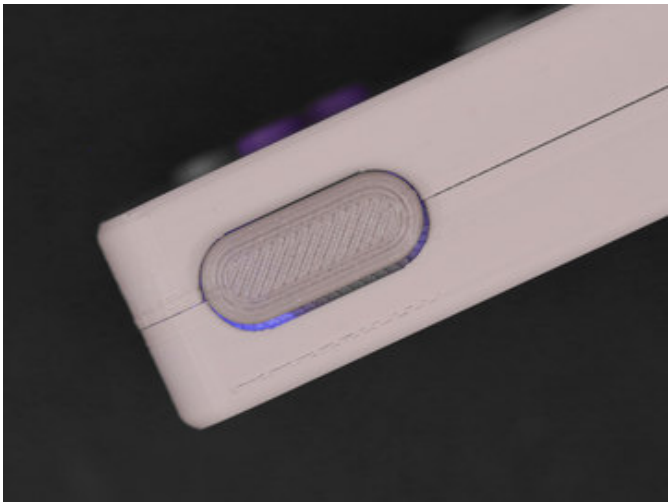
Carefully join the two halves of the case and press them firmly together. Ensure all of the wires are inside the enclosure to prevent kinking any of the wires.





### Install Shoulder Actuators

The shoulder buttons printed in TPE filament is easy to install. Squeeze the actuator and press it into the openings. Hold the two halves together while installing the actuators to prevent the case from coming apart.





## Final Build

Congratulations! You've finished building the PiGRRL Zero! Take a moment to test out all of the button acutators with a play test. If eveything works as expect, great! If not, please feel free to post any questions or concerns in the [Adafruit Forums \(https://adafru.it/dYq\)](https://adafru.it/dYq).



## Usage & Extras

Please see notes on the “Software” page about expanding the filesystem to utilize the whole SD card. This is not done automatically.

### PiTFT Buttons

The four buttons the PiTFT display are mapped to the following (start from left to right):

1. Exit - Escape (GPIO 17)
2. Select - Left Control (GPIO 22)
3. Start - Enter/Return (GPIO 23)
4. Not Mapped - 4 (GPIO 27)

### Connecting to WiFi

Before connecting a WiFi adapter, you'll want to use a USB hub so you can connect a USB keyboard necessary for typing your WiFi password. Then, connect a WiFi adapter into the USB hub.

Go to the RetroPie section in Emulation Station and choose "Configure WiFi". Select "1 Connect to WiFi network" with the Enter key. Use the up and down buttons of the DPad to highlight your network and the Enter key to choose. Type in your WiFi password and use the Enter key to submit. Your Pi's IP address will display in the top of the WiFi Menu. Choose "Cancel" using the Enter key to exit the "Configure WiFi" menu.

### Uploading ROMs

You'll need to get the Pi on your WiFi network in order to connect to it. Use your desktop computer and your preferred SFTP client to connect.

- IP Address - example: 10.1.10.XXX
- Username: pi
- Password: raspberry

Go to the following directory to upload ROMs

`/home/pi/RetroPie/roms`

Upload your ROMs to the proper emulator folder (ie. nes, snes, gb, etc.)

### USB Port

The USB port can be used to connect peripheral such as a USB hub, keyboard, WiFi, Bluetooth or audio adapter.

Please note, USB 3.0 hubs can be problematic the Raspberry Pi Zero.

### Audio Adapter

A USB Audio Adapter can be used to output audio. We suggest using the one linked in the BOM, as it's proven to work with the Raspberry Pi. There's no need to configure the audio since the ready-to-go PiGRRL Zero image has it preinstalled.

To change the audio output level, go to the "RetroPie" section in emulation station and select "Configure Audio Settings", then choose "4 Mixer - adjust output volume" using the "enter / return" key (3rd button from left) on the PiTFT. From here, you can use the DPAD up or down buttons to adjust the audio levels. Press the "exit" button (far left button) on the PiTFT to save and exit "ALSA Mixer".

## Exiting a ROM/Game

To quit a game and go back to the Emulation Station screen, press and hold the "Start" and "Select" buttons at the same time.

# RetroPie: Improving Emulator Performance

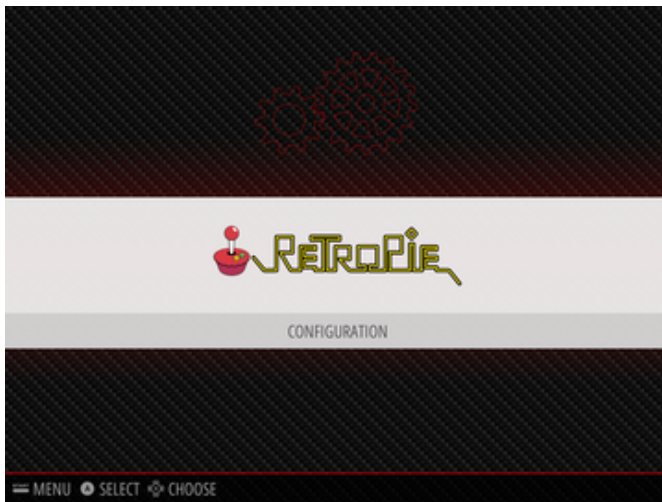
On **single-core** Raspberry Pi boards like the **Model A+** and **Pi Zero**, emulation speed is sometimes less than stellar...the screen may update like molasses, or sound may be choppy. Let's look at some ideas to fight back...

All of these methods require a **USB keyboard** attached.

The directions shown here are for **RetroPie 4.1**, but the same general ideas apply to other versions and possibly even other emulation packages, the steps will just be different.

## Overclocking

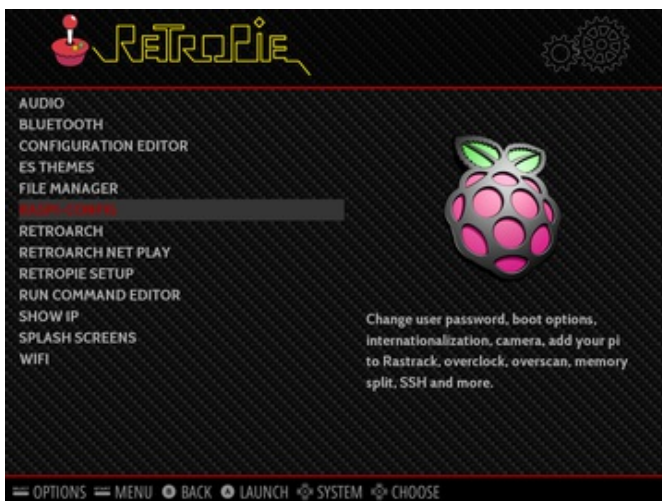
Most Raspberry Pi boards can handle some *overclocking* — running the processor slightly faster than its “official” speed. Too fast though, and the system can become unstable and crash. Due to manufacturing variances, every single board is slightly different in this regard...you'll need to do some experimentation to find the fastest *reliable* setting for your specific board. Restart after making any changes and **try running some games for several minutes** to really put the system through its paces.

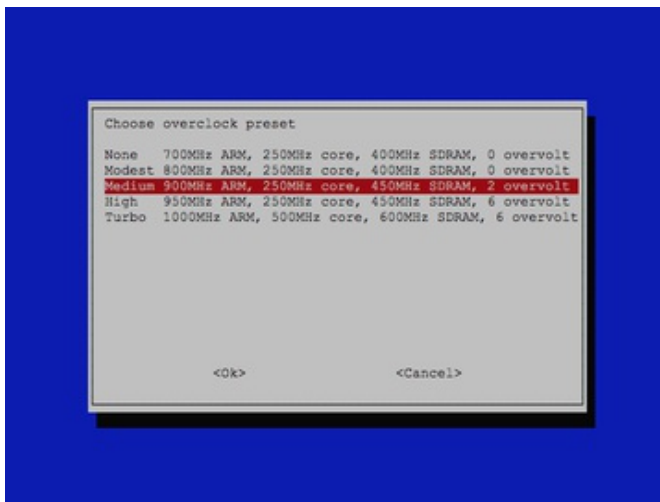
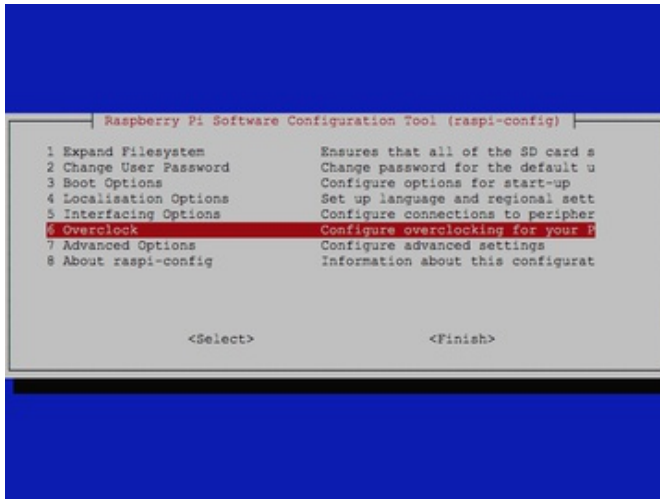


You can access the **raspi-config** utility from the RetroPie configuration screen, or press “F4” to exit to a command line and run “sudo raspi-config” manually.

raspi-config is keyboard-based, mostly using the **arrows** and **enter** keys; arcade controls, if connected, probably won't have the intended effect.

Different Pi models offer different overclock settings. On the Model B+ being tested here, the “Medium” setting seems quite reliable. Other board types may overclock automatically and there's nothing to adjust here.





If you're using a PiTFT display, the "core" frequency setting should **not exceed 300 MHz**; the display will glitch and may not work at all. So avoid the higher overclock settings if using one of these screens, or manually edit the file `/boot/config.txt` to fine-tune various system frequencies individually.

## Choosing an Alternate Emulator

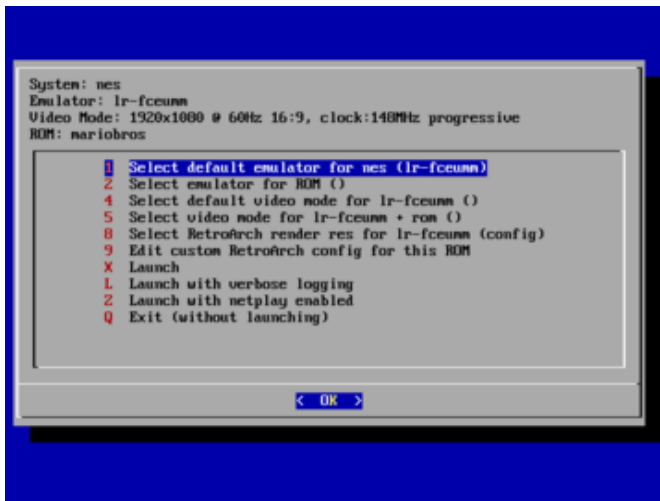


When launching a game, there's a brief moment when it's possible to access some launch options...

When you see this "launching" screen, press the space bar (or any other key that generates an actual keycode... not a "meta" key like shift or control).

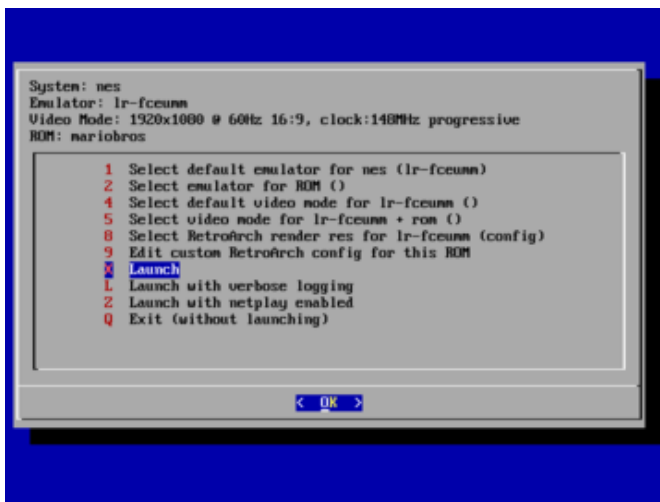
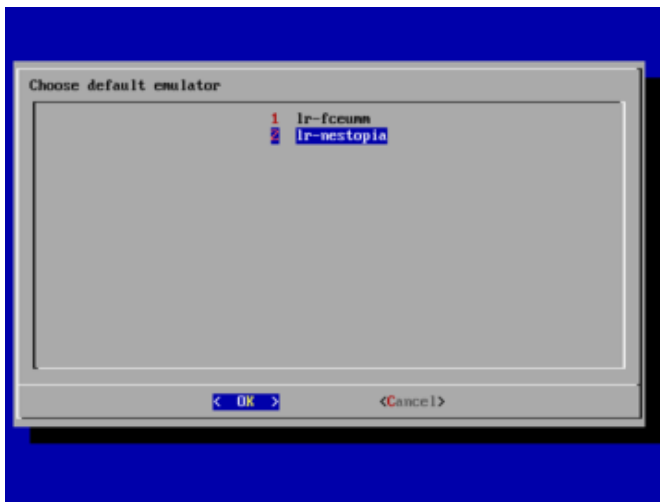
This brings up a menu...similar to raspi-config, it operates with the keyboard, not arcade buttons.

The first option lets you select an alternate emulator program (if available). For example, RetroPie 4.1 includes two different NES emulators: lr-fceumm (the default) and lr-nestopia. Different emulators make tradeoffs with regard to performance and "accuracy" of the emulation.



Try an alternative and see how it performs (use the “Launch” option).

The selection you make will “stick” — it is **not necessary to use this menu every time**, unless you want to switch back to the original emulator choice. There’s also an option to use an alternate emulator only for specific ROMs, if you have different needs for different games.



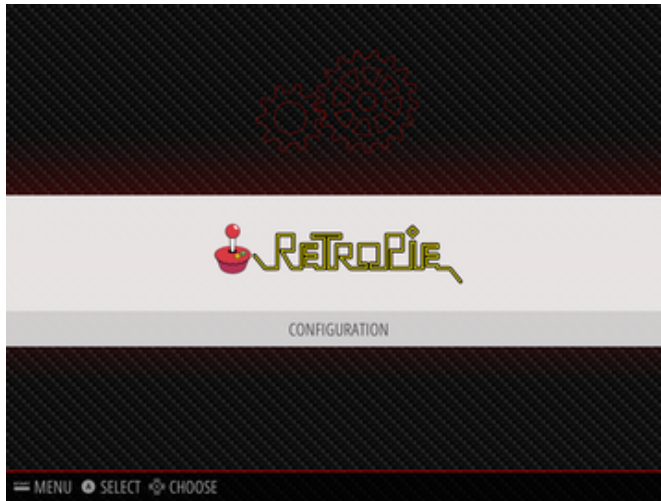
The above steps are fairly quick and easy. If they meet your needs, consider the job done! If it’s still not quite the performance you need, the situation gets more involved...

These next steps *require* a network connection. On a Model A+ or Pi Zero, that usually requires a USB hub and an Ethernet or WiFi adapter, along with some configuration (the RetroPie 4.1 configuration menu includes WiFi setup). Alternately, if you have a spare Model B+ around, that’s easy to get on an Ethernet network...move the SD card over



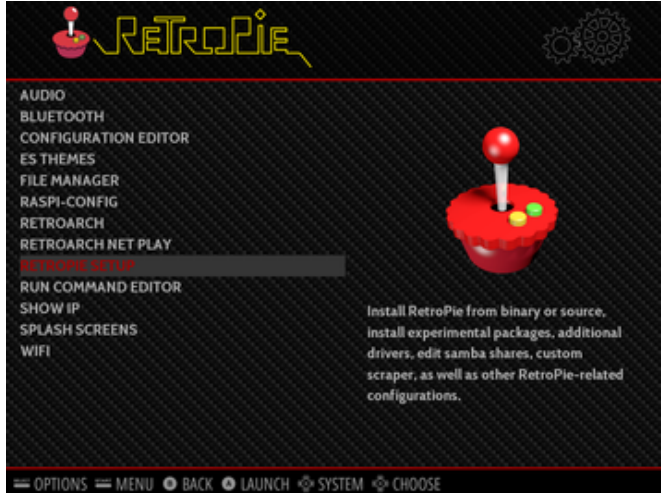
there for setup, then move it back to the target system when done.

## Installing Additional Emulators



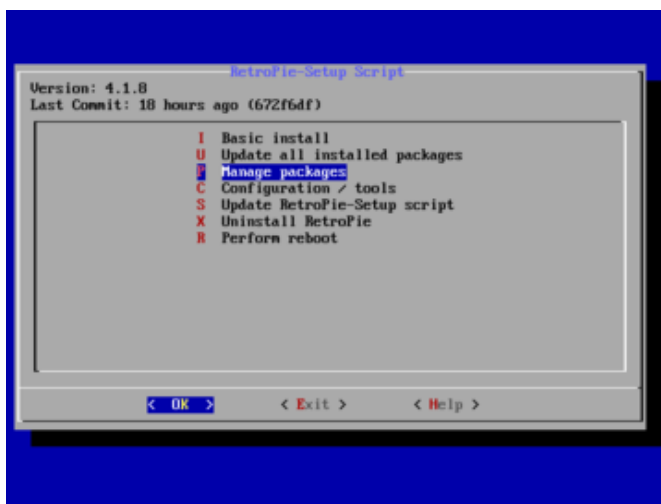
Some “bleeding edge” emulators might offer better performance than the stock options provided by RetroPie. As with the “Alternate Emulators” above, it sometimes comes with a tradeoff in accuracy (hence their non-inclusion), so you’ll need to experiment and decide if it’s worth it.

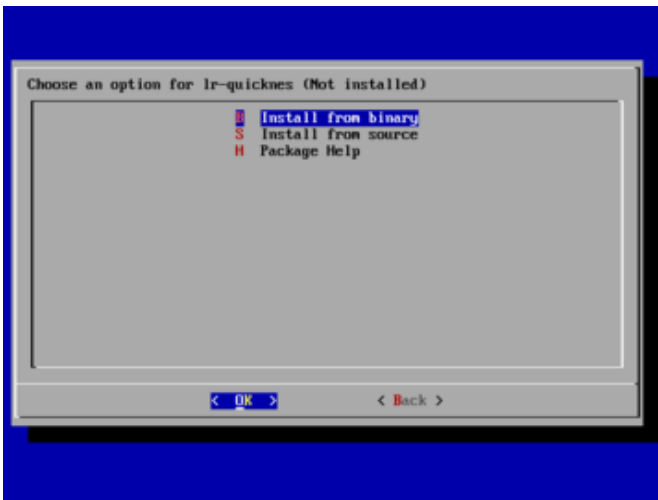
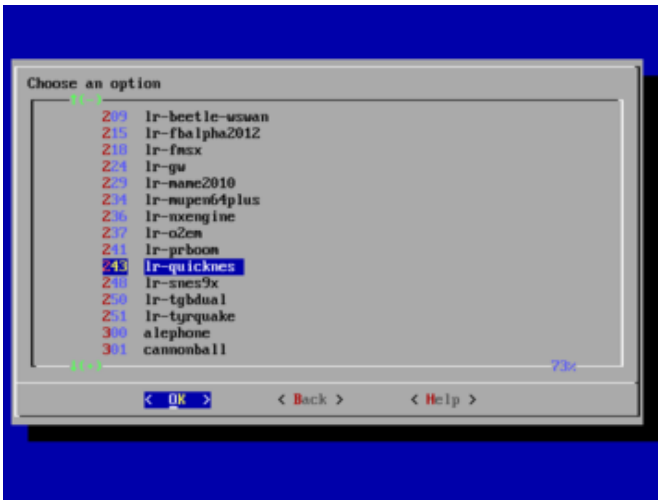
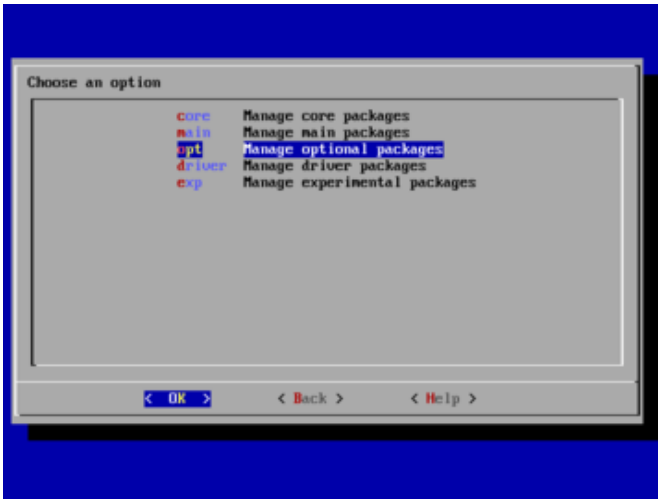
From the RetroPie configuration screen, select “RetroPie Setup,” which switches to another of these keyboard-based menus.



Select “Manage packages,” then either “Manage optional packages” or “Manage experimental packages.” You’ll get a list of various emulators and/or games that aren’t normally installed by default. *lr-quicknes* was an optional package for NES emulation... it’s faster than the others...*too* fast, in fact, so I ended up not using it...but that’s all part of the experimentation process.

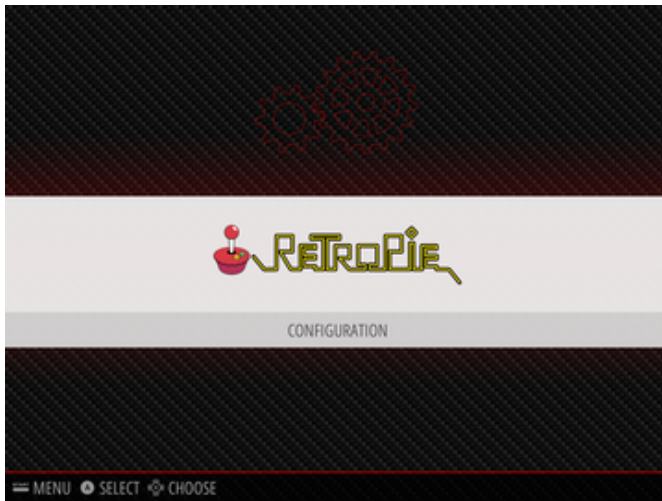
Choose the “Install from binary” option to install the selected package. Don’t bother with the “source” option...it’s *very* time consuming and usually provides no benefit.





## Updating Emulators to Latest Versions

Sometimes you just need the newest-and-shiniest version of an installed emulator to provide the best available performance...

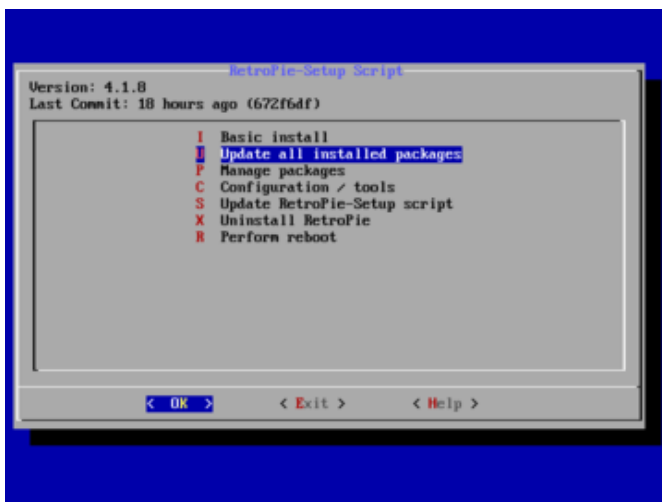
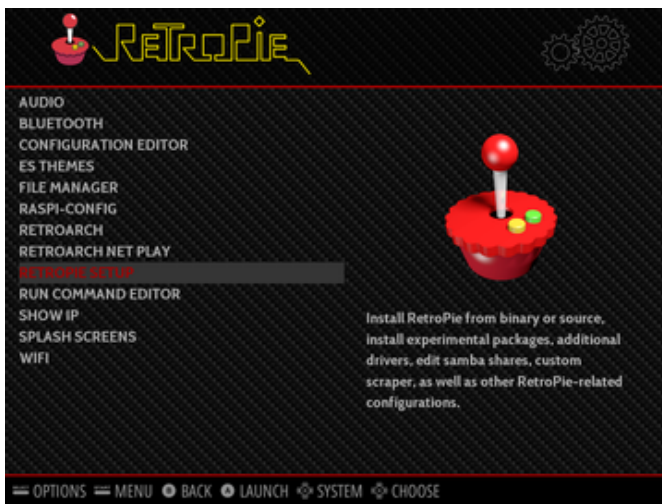


From the RetroPie configuration screen, select “RetroPie Setup.” This brings up the keyboard-driven menu again. Choose the “Update all installed packages” option to make a sweep through the system that’ll update *everything* to the latest release, including the kernel.

This system-wide update can be extremely time-consuming...possibly an hour or so, depending on processor and network speeds.

Even the default Ir-fceumm on a non-overclocked Pi ran great after updating!

If using a PiTFT display, think twice before trying this option. Perhaps make a **backup** of your SD card first. Kernel updates have been known to break PiTFT support in some circumstances, and you’d have to start all over.



# Troubleshooting RetroPie and retrogame

---

This page documents the most common pitfalls encountered when making retro gaming projects, offering some solutions and where to turn for further help.

There are some things we can fix and some we can't...we're not involved in RetroPie's development, for example...but we'll try to point you in the right direction.

Most of the following troubleshooting steps will require a **USB keyboard attached**. Some require a **network connection**.

## *retrogame* Related Troubleshooting

---

*retrogame* is our software that converts GPIO button actions into keyboard events. **These are the problems we're best equipped to fix.**

Some common things to check for any *retrogame* installation:

- Confirm button/joystick wires go to the correct pins and to ground. A multimeter with a continuity beep function is helpful for testing.
- The retrogame configuration file (`/boot/retrogame.cfg`) uses *Broadcom pin numbers*...these are *not* sequential along the GPIO header pins. [This site has a nice reference chart \(https://adafru.it/uFH\)](https://adafru.it/uFH) (use the "BCM" numbers).
- Earlier versions of *retrogame* didn't use a configuration file...you had to edit and compile the source code. That's just horrible. If you're running an early version like that, this would be a good time to upgrade. See the [Installing Retrogame \(https://adafru.it/sct\)](https://adafru.it/sct) page.

### **Some of my buttons/controls aren't working!**

- Check connections and configuration file pin numbers as explained above. (If some controls are responding, that's an encouraging start...it at least means the code is running.)
- The key codes generated by retrogame might not be assigned to EmulationStation's keyboard inputs; one or the other will need to be changed. Either edit `/boot/retrogame.cfg`, or, from the EmulationStation main screen, press Start to access the main menu, then select "Configure Input" and proceed through each of the controls.

### **NONE of my buttons/controls are working!**

- Confirm that retrogame is actually running...either exit to the command line (F4) or log in using ssh, then use `ps -ef | grep retrogame` to check. If you used our installer script or one of our ready-made SD card images, it should be started automatically on boot (added to `/etc/rc.local`).
- Confirm that the file `"/etc/udev/rules.d/10-retrogame.rules"` exists. Our installer script creates this file, but if you installed retrogame manually or from source, it may have been overlooked.

### **My controls only work if there's also a USB keyboard plugged in!**

Confirm that the file `"/etc/udev/rules.d/10-retrogame.rules"` exists. Our installer script creates this file, but if you installed retrogame manually or from source, it may have been overlooked.

### **Retrogame doesn't work with my optical buttons!**

Unfortunately, yes. retrogame only works with "passive" switches between a GPIO pin and ground (logic low=pressed). It won't work with switches that have the opposite logic level (high=pressed).

### **I ran Adafruit's retrogame installer script and rebooted, and now the keyboard and network are unresponsive!**

This can happen if you're running an early Raspberry Pi (Model A or B) with the 26-pin GPIO header and select the

“Six buttons + joystick” option in the retrogame installer. That particular configuration is set up for our Arcade Pack and newer (40 pin) Raspberry Pi boards. Some of the pin numbers referenced *don't exist* on the older 26-pin header and lead to trouble.

If this happens to you, not to worry. Power off the Pi and insert the SD card in a reader on a PC or Mac. Look for a file called retrogame.cfg...edit this file and change the pin numbers to match your specific controller wiring.

## RetroPie Related Troubleshooting

---

RetroPie provides the actual emulation software and a nice user interface. As this is third-party code, the “depth” of problems we can troubleshoot is more limited, but here are some of the common issues we’ve seen and how to resolve them...

### **My controls work in the EmulationStation UI, but not in one or more specific emulators!**

This can happen with certain emulators (usually older or more esoteric ones) that don't use the *libretro* library. Among other things, libretro allows the global controller configuration to be used everywhere. There are a couple of workarounds that *might* help, but no guarantees...

- You can rummage around in `/opt/retroPie/configs` and look for a configuration file specific to the problem emulator, then edit its keyboard layout to match your controls. The format of this file, if one even exists, is likely specific to that one emulator, so you'll need to do some research (Google search, etc.), it's not something we can help out with.
- You can try hunting for an alternate emulator based on libretro, if there's one available (see “Installing RetroPie Packages” below).

### **The ROM files I have worked in a different emulator before, but aren't working in RetroPie!**

This can happen if the ROM file format changes between versions of an emulator, or if two emulators for the same system use different formats.

- Do some research (Google search, etc.) to see if this is the case. It's possible there may be utilities to convert among different ROM formats.
- Try installing an alternate emulator, if there's one available (see “Installing RetroPie Packages” below).

A few emulators may require a “BIOS file” in order to function, but it's not included with the software for legal reasons. This is something you'll have to research and track down.

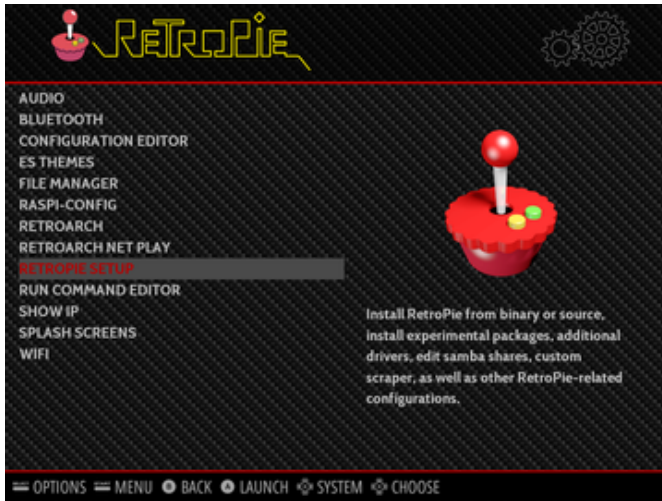
## Installing RetroPie Packages

---

To add support for a system not present in RetroPie by default, or to add an alternate emulator program for an existing system, select “RetroPie Setup” from the RetroPie menu. **This brings up a text-menu-based interface and will require a USB keyboard to navigate.**

Select “Manage packages” and then one of the core, main, optional or experimental selections...you'll probably want to navigate through each of them to see what's available, keeping in mind that each successive category might be a little rougher around the edges.

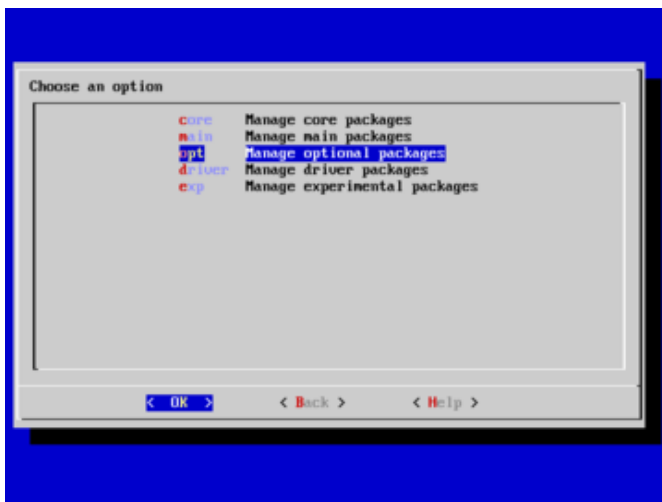
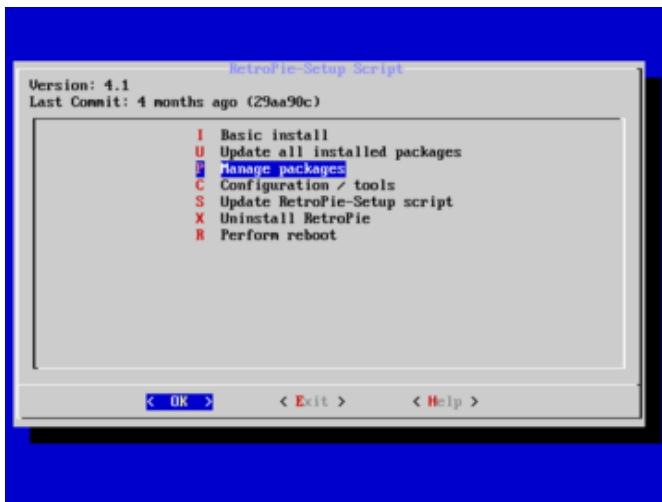
**Your best bet are packages whose names begin with “lr-”.** This means they're built using *libretro* and the

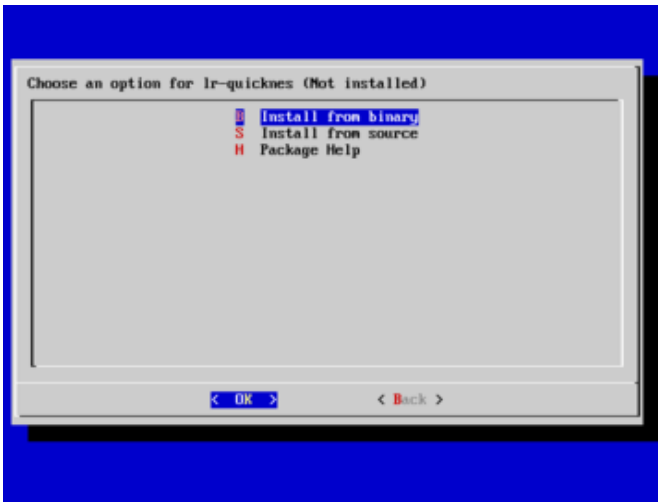
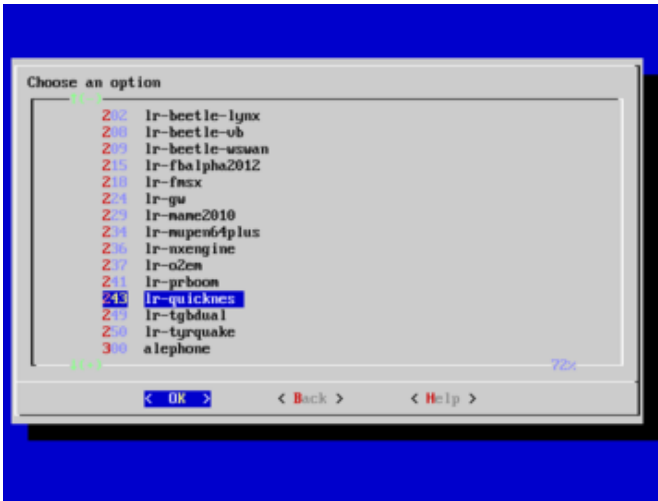


control inputs should already work with what you have! Other packages may require their own manual controller setup, which can be a real nuisance.

When asked, select “**Install from binary.**” The *source* option takes *much* longer and won’t provide any benefit for the average user...only attempt that if you *know* you need absolutely bleeding-edge code.

It’s totally valid to install multiple emulator packages that handle the same type of system. Each might have different performance or compatibility benefits, so it’s worth testing your options. See “Accessing Alternate Emulators” below.





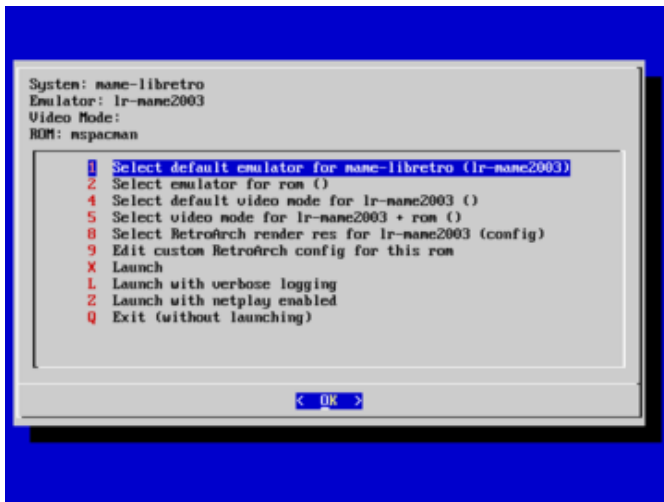
## Accessing Alternate Emulators



When you launch a game, you'll briefly see this nondescript launching message. **You have a couple of seconds to hit a button or key...**

The first option in this configuration menu lets you select a different emulator package for a given system type...or even on an individual ROM-by-ROM basis, if different games benefit from different emulation software.

Test it out with. If you don't like the results, next time you launch that game you can access the configuration menu again and restore the original selection.



## More RetroPie Help

For issues not covered above, the best sources for RetroPie-specific help are the official documentation and forum on the RetroPie web site:

[RetroPie Documentation \(https://adafru.it/uFI\)](https://adafru.it/uFI)

[RetroPie Forum \(https://adafru.it/uFJ\)](https://adafru.it/uFJ)