

□

## Pocket PiGRRL

Created by Ruiz Brothers



Last updated on 2017-01-07 12:23:16 AM UTC

# Guide Contents

Guide Contents	2
Overview	7
Raspberry Pi	8
This Guide	8
Prerequisite Guides	9
Parts	10
Tools	11
Supplies	11
Circuit Diagram	12
Powerboost 1000c	12
PAM8302	13
Perma-Proto Two Button Layout	13
PiTFT Buttons	14
Perma-Proto Four Button Layout	14
Half size Perma-Proto	15
GPIO Ribbon Cable	15
Software	16
Download and Burn SD Card	16
Setup Network	16
Adding Four Button Controls	16
Configuring RetroArch Emulators Controls	17
Software [Manual]	19
Download & Burn RetroPie	19
USB 2.0 and Ethernet Hub - 3 USB Ports and 1 Ethernet	19
Setup RetroPie	20
Testing Retrogame	21
Install PiTFT (fbcp) Support	21
Edit Config.txt	23
Installing Keypress (retrogame) support	23
Configuration	26
Uploading ROMs	26
Exiting ROMs	26
Remapping Controls	26
Testing Controls	26

<b>3D Printing</b>	<b>29</b>
Two Button Version	29
Four Button Version	29
Parts	30
Support material	31
Tolerances	31
Materials	32
Warping	32
First layer + Bed Leveling	32
Ninjaflex buttons	33
Quality Prints	33
Magnets	34
<b>Wiring</b>	<b>36</b>
Prep Wiring	36
Cut, Strip, Tin, Repeat	36
The Slide Switch	36
Powerboost 1000C	37
PAM8302 & Speaker	38
2.4' PiTFT	39
Perma-proto	40
<b>Switch</b>	<b>41</b>
Prepping Slide Switch	41
Secure Slide Switch	42
Wire Slide Switch	43
Heat Shrink Slide Switch	44
Install Slide Switch	45
<b>Powerboost 1000C</b>	<b>47</b>
Prep PowerBoost 1000C	47
Wire Powerboost 1000C	48
Thread Mounting Holes	49
Install PowerBoost 1000C	50
Mount PowerBoost 1000C	51
Test PowerBoost 100C	52
<b>PAM8302</b>	<b>53</b>
Prep PAM8302	53
PAM8302 Wires	54
Thread PAM8302	55
Solder Wires to PAM8302	56
Mount PAM to enclosure	57

Solder power to PAM8302	58
<b>Pi Audio</b>	<b>59</b>
Cut Pi Audio wires	59
Strip and Tin Wires	60
Tin Audio pins on Pi	61
Solder Audio Wires to Pi	62
Solder Audio wires to PAM8302	63
Mount Pi	64
Mounted Pi	65
<b>Speaker</b>	<b>66</b>
Prep Metal Speaker	66
Prep New Wires for Speaker	67
Solder Wires to Speaker	68
Install Speaker	69
Connect Speaker wires to PAM8302	70
<b>2.4 PiTFT</b>	<b>71</b>
Install Header	71
Secure Header	73
Remove Tac	74
Add Buttons	75
Trim buttons	76
Disable GPIO 18 Lite	76
Prep Display PCB	77
Peel	78
Stick Display	79
<b>Pi Cable</b>	<b>80</b>
Connect Pi Cable	80
Remove Pi Cable Connector	81
Peel Cable	82
<b>Mount Display</b>	<b>83</b>
Install display buttons	83
Install PiTFT	84
Mount PiTFT	85
<b>Perma-Proto</b>	<b>86</b>
Using Half-size Perma-Proto	86
Mark Perma-Proto	87
Cut Perma-Proto	87
Install buttons (Two Action Buttons)	88

Install buttons (Four action buttons)	89
Solder buttons to PCB	89
Prep Ground Wires	90
Solder Ground Wires	91
Up Button Ground Wire	91
Trim legs	92
2-Button Layout	93
4-Button Layout	94
<b>Buttons</b>	<b>95</b>
Connect Pi Cable to Powerboost	95
Soldered Power	97
Cut extra wires short	98
Heat Shrink Pi Cable	98
Install NinjaFlex Buttons	99
Install Perma-Proto	100
Wire Ground	102
Wire Left Button	103
Wire Right Button	105
Wire Up Button	106
Wire Down Button	108
Wire A Button	109
Wire B button	111
Wired Perma-Proto to Pi Cable	112
Perma-Proto (Four button layout)	113
<b>Assembly</b>	<b>114</b>
Final Wire Check	114
Plug Battery	115
Tack Battery	116
Stick Battery	117
Installed Battery	118
Close GPIO headers	119
Wire Kink Check	120
Fasten Last Screw	121
Final Build	122
I Made One, YES!	123
<b>RetroPie: Improving Emulator Performance</b>	<b>124</b>
Overclocking	124
Choosing an Alternate Emulator	126

Installing Additional Emulators	127
Updating Emulators to Latest Versions	129

---

# Overview



The gameboy is an iconic portable gaming device that most adults will probably remember dedicating hours of their childhood playing games on one.

This DIY project isn't about hacking or modding an old gameboy. In this project, we're building a completely new gaming device using 3D printing, the Raspberry Pi A+ and components from Adafruit.

**Please note** this is not a professional product! It's a DIY kit, and is great fun to build, but it may not have great emulation, video, audio etc.

3D printing allows you to design custom enclosures for your projects. The design and form factor are very much inspired by the gameboy, but by no means meant to look exactly like it.



## Raspberry Pi

Our first PiGRRL project used the Raspberry Pi Model B and Adafruit 2.8' PiTFT.

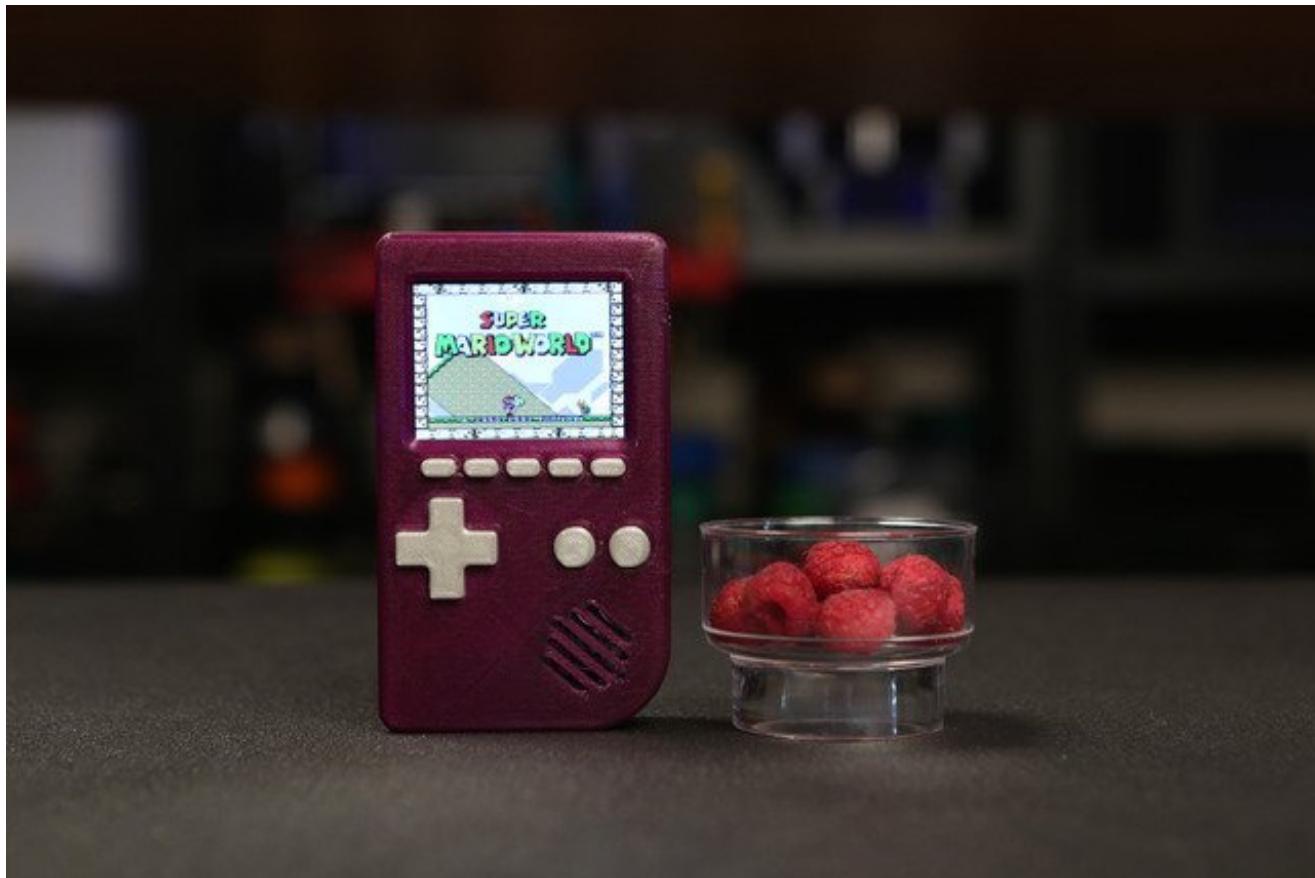
In this project, we're using the Raspberry model A+ and a 2.4" PiTFT HAT. The small size of the A+ and the compact display really allows this project to be one of the smallest builds yet.

- Raspberry Pi A+
- PiTFT 2.4" 320x240 Touch Display
- Retropie w/ Emulation Station
- Mono Audio Amp + 8 ohm ~1W Speaker
- Tactile Buttons w/ Perma-Proto
- Powerboost 1000C w/ 2000mAh lipo battery

## This Guide

This tutorial will walk you through the assembly and wiring. It's a challenging build but it's not the most hardest thing ever. Our intent was to make this project easier than our previous PiGRRL build.

So, if this is your first DIY electronics + 3D Printing project, you might want try a simpler project with less wiring and soldering. That said, you shouldn't feel discouraged to take on the project.



## Prerequisite Guides

We recommend walking through these guides to get familiar with the software and components. You don't have to, but it's a good idea to check them out if you're not sure what all these things do.

- [Running OpenGL-based Games](http://adafru.it/iOF) (<http://adafru.it/iOF>)
- [Adafruit Pi Finder](http://adafru.it/iDP) (<http://adafru.it/iDP>)
- [Powerboost 1000c intro](http://adafru.it/iPa) (<http://adafru.it/iPa>)
- [Colin's Lab: Soldering](http://adafru.it/dyT) (<http://adafru.it/dyT>)



## Parts

The parts below were used in this project. You don't have to use the exact same parts but the enclosure was designed to specially fit these parts - so if you want to use different components, you can but just know it may not fit in the provided enclosure. You can of course tweak the CAD files.

- [Raspberry Pi A+ \(http://adafru.it/2266\)](http://adafru.it/2266)
- [2.4" PiTFT touch screen \(http://adafru.it/2455\)](http://adafru.it/2455)
- [Powerboost 1000C \(http://adafru.it/2465\)](http://adafru.it/2465)
- [2.5W Class D Audio Amplifier - PAM8302 \(http://adafru.it/2130\)](http://adafru.it/2130)
- [2000mAh lithium polymer battery \(http://adafru.it/2011\)](http://adafru.it/2011)
- [Perma-proto half-sized \(http://adafru.it/eni\)](http://adafru.it/eni)
- [GPIO Pi Ribbon Cable - 26 pin \(http://adafru.it/f74\)](http://adafru.it/f74)
- [6mm tactile buttons \(square\) \(http://adafru.it/eww\)](http://adafru.it/eww)
- [6mm tactile switch buttons \(slim\) \(http://adafru.it/dSI\)](http://adafru.it/dSI)
- [12mm tactile buttons \(http://adafru.it/f75\)](http://adafru.it/f75)
- [Mini metal speaker \(http://adafru.it/dDb\)](http://adafru.it/dDb)
- [Slide switch \(http://adafru.it/drN\)](http://adafru.it/drN)

# Tools

Having the right tools makes this build easier and more fun. Most of these are available in the shop but use whatever tools you have on hand.

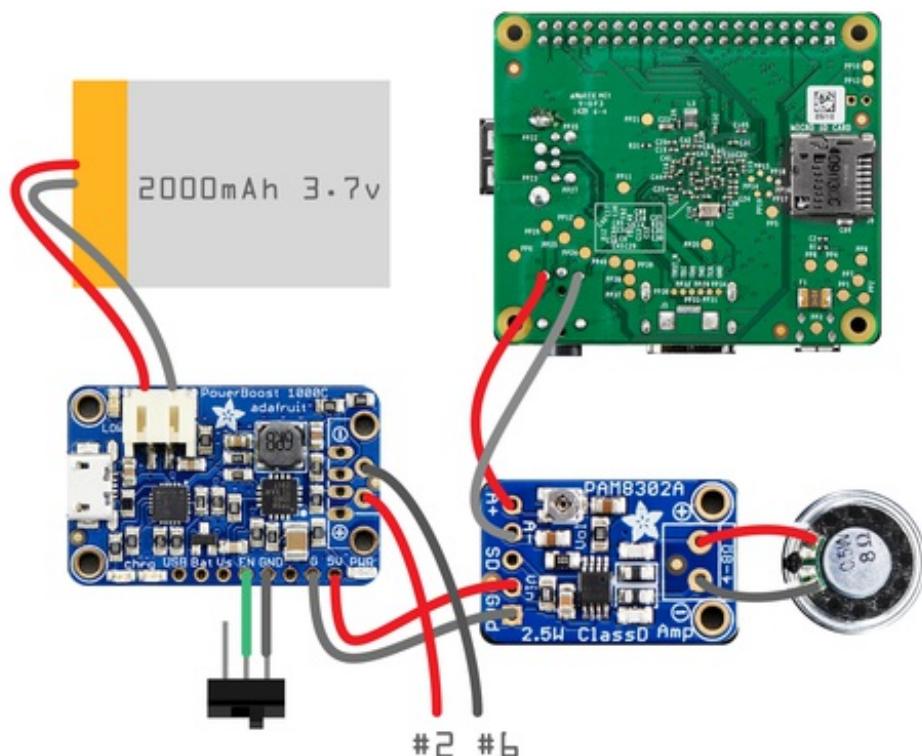
- [3D Printer](http://adafru.it/d9z) (<http://adafru.it/d9z>)
- [Soldering Iron](http://adafru.it/doU) (<http://adafru.it/doU>)
- [Flush cutters](http://adafru.it/dxQ) (<http://adafru.it/dxQ>)
- [Wire strippers](http://adafru.it/527) (<http://adafru.it/527>)
- [Panavise Jr.](http://adafru.it/151) (<http://adafru.it/151>)
- [Helping third hand](http://adafru.it/291) (<http://adafru.it/291>)
- [Screwdriver](http://adafru.it/f76) (<http://adafru.it/f76>)
- Rotary cutting power tool

# Supplies

Wires, screws, magnets, filament - The supplies listed below are both helpful and necessary for completing this project.

- [30AWG silicone cover stranded-core wire](http://adafru.it/2051) (<http://adafru.it/2051>)
- [Heat shrink tubing](http://adafru.it/1649) (<http://adafru.it/1649>)
- [3D printing filament](http://adafru.it/2080) (<http://adafru.it/2080>)
- [Blue painters tape](http://adafru.it/ftO) (<http://adafru.it/ftO>)
- Solder rosin-core
- [12 #4-40 flat phillips machine screws](http://adafru.it/fkj) (<http://adafru.it/fkj>) or [here](http://adafru.it/iPb) (<http://adafru.it/iPb>)
- 2 Neodymium Magnets 1/4 x 1/16 inch Disc N48

# Circuit Diagram



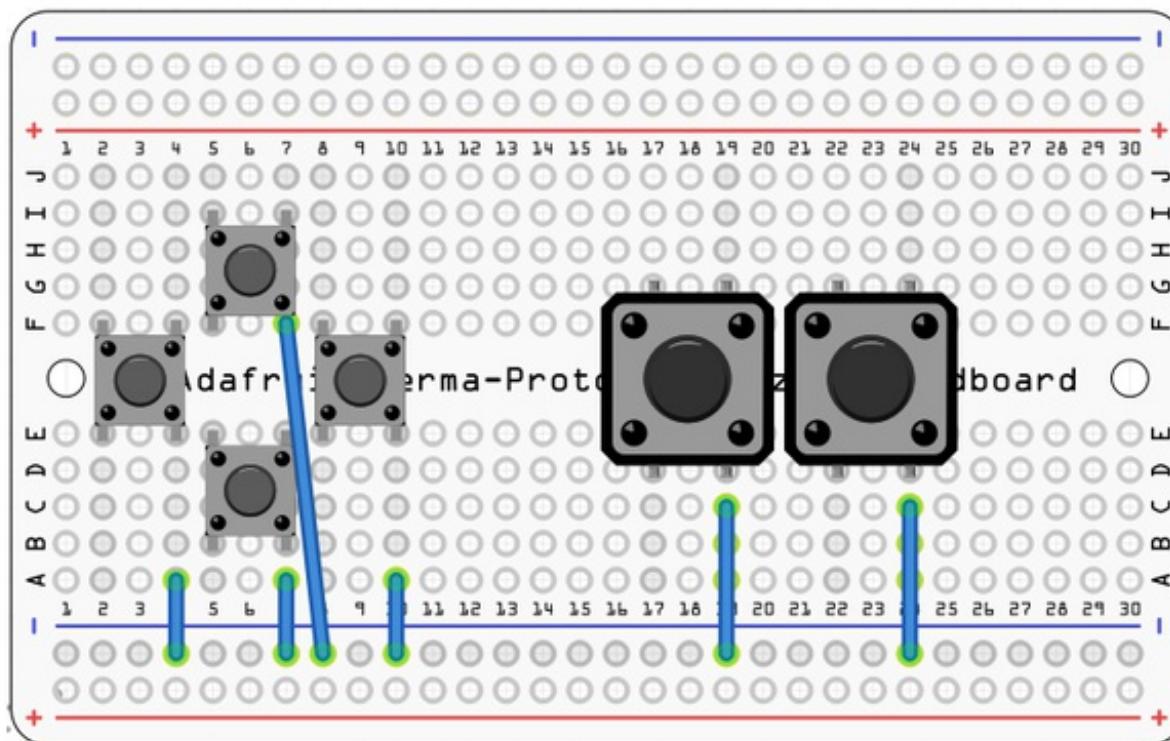
The circuit diagram above shows the connections for the power and audio. The lengths of the connections are not exact and mostly meant to visually represent which pin goes where. It's best to use this as a reference for pins.

## Powerboost 1000c

The slide switch is wired to pin **GND** and **EN**. The 2000mAh lithium polymer battery is connected to the JST port. The **#2(5V)** and **#6(GND)** connections represent the wire number from the Pi ribbon cable.

PAM8302

The **VIN** and **GND** pins are wired to the **5V** and **GND** pins on the powerboost 1000c. The **A+** and **A-** pins are wired directly to the audio pins on the bottom of the Raspberry Pi A+. The speaker is wired to the amps audio output (+ and -).



# Perma-Proto Two Button Layout

To interface the Pi with the tactile buttons, we'll use a Pi ribbon cable (26 pin). The connections below indicate the wire number, gpio name and keyboard input controller button. The White colored wire will be represented as Wire #1. The blue connections indicate the ground for each buttons. These are all wired to the bottom ground rail.

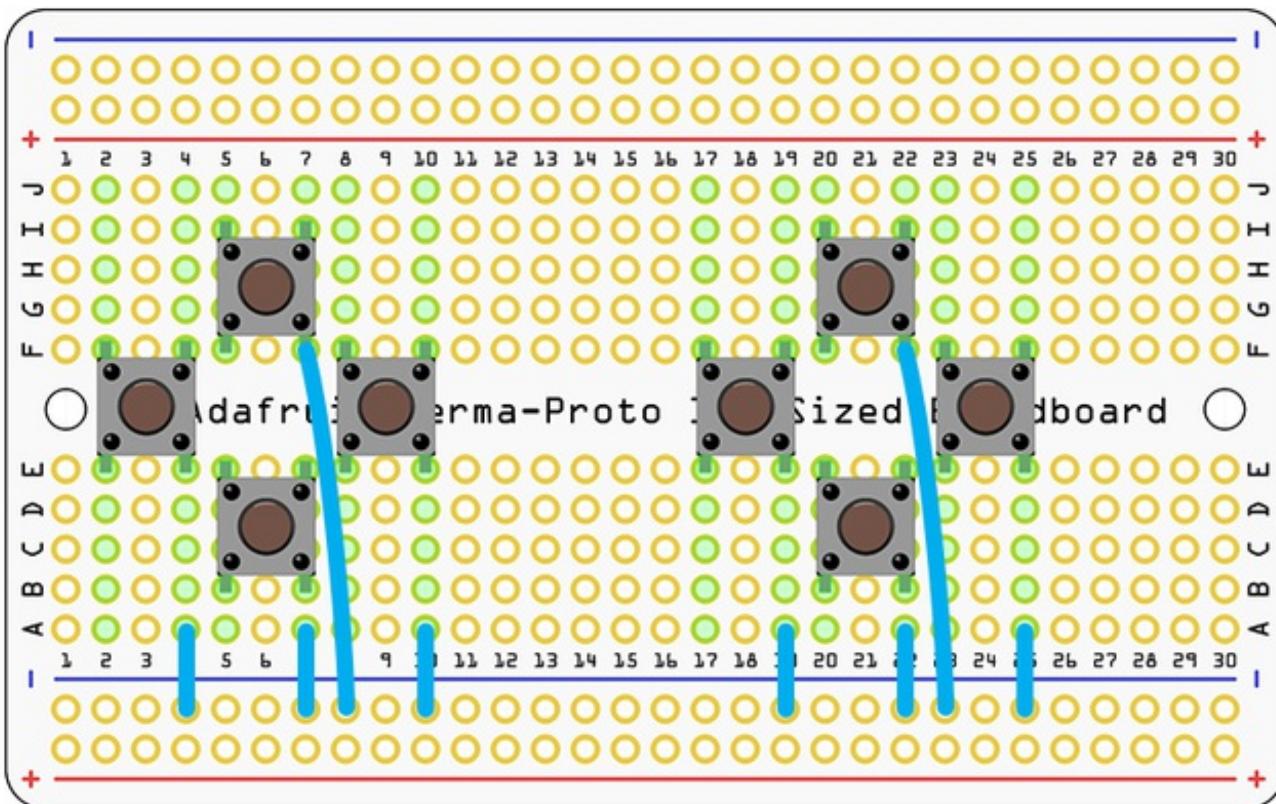
- Wire # 2 - 5V Powerboost
  - Wire # 6 - GND Powerboost
  - Wire # 7 (GPIO 4) - Key Left
  - Wire # 9 - Ground for Buttons PCB

- Wire # 11 (GPIO 17) - Key Right
- Wire # 12 (GPIO 18) - Key Up
- Wire # 13 (GPIO 27) - Key Down
- Wire # 15 (GPIO 22) - Key A
- Wire # 16 (GPIO 23) - Key B

## PiTFT Buttons

There are five buttons that are connected to the PCB near the bottom of the display. The GPIO number for these buttons are listed below and on the PCB.

- GPIO 5 - Key Z
- GPIO 6 - ESC
- GPIO 12 - ENTER
- GPIO 13 - SPACE
- GPIO 16 - Key X



## Perma-Proto Four Button Layout

The four button version uses the same wiring from the Pi Cable, except two extra wires need to be connected to the Perma-proto PCB.

- Wire #3 (GPIO 2) - Key X
- Wire #5 (GPIO 3) - Key Y

## Half size Perma-Proto

The diagram above shows which buttons are connected to the perma-proto PCB. Ideally, I want to say you can customize and change it up, but then the buttons won't fit in the cutouts on the 3D printed enclosure - so if you wanna change this layout, you'll also have to update the CAD, so keep that in mind.

## GPIO Ribbon Cable

To make the wiring a bit easier, we're using a Pi ribbon cable (the 26-pin one). This cable as a nifty connector that fits perfectly on the GPIO breakout on the 2.4" PiTFT. We'll remove the connector from the other side and wire that up directly to the buttons on the Perma-Proto.

# Software

## Download and Burn SD Card

The first step is to download the PocketPiGRRL image, which includes retrogame, fbcp tools and GPIO controls preinstalled. Once it's downloaded, you'll need to properly burn the .IMG file to a microSD card (4GB min). I personally used [RPi-SD card builder v1.2](http://adafru.it/aMW) (<http://adafru.it/aMW>).

[Download Pocket PiGRRL IMG](#)

<http://adafru.it/fOJ>

This IMG is optimized for the "two-button" version. If you're making the four button version, you'll need to make a minor edit to the retrogame.c file.

## Setup Network

Next step is to get wifi network situated on the Pi. We need the Pi to connect to your local wifi network so you can SSH and install ROMs. You're most likely going to want a USB wifi dongle. To set this up, edit the **/boot/occidentalis.txt** and add your WiFi credentials.

## Adding Four Button Controls

The PiGRRL Img is currently configured for the two button version. To get the four buttons working in Retrogame, you'll need to add them in the **retrogame.c** file.

Make sure the Raspberry Pi is configured for Wifi. Open up terminal and ssh into the Pi. (Default username: **pi** password: **raspberry**) Get into the Adafruit-Retrogame directory and edit the **retrogame.c** file.

```
ssh pi@10.1.10.XXX  
cd Adafruit-Retrogame  
sudo nano retrogame.c
```

Scroll down to the part where you see the table called ioStandard (*not* the ioTFT table — that's for other projects). Each line in brackets represents one pin on the GPIO header and a corresponding key code.

You can map the controls to any keyboard characters you'd like. The full list of available keycodes can be found in **/usr/include/linux/input.h**

```

ioStandard[] = {
// This pin/key table is used when the PiTFT isn't found
// (using HDMI or composite instead), as with our original
// retro gaming guide.
// Input Output (from /usr/include/linux/input.h)
{ 4, KEY_LEFT }, // Joystick (4 pins)
{ 17, KEY_RIGHT },
{ 18, KEY_UP },
{ 27, KEY_DOWN },
{ 22, KEY_LEFTCTRL }, // A/Fire/jump/primary/RED
{ 23, KEY_LEFTALT }, // B/Bomb/secondary/YELLOW
{ 2, KEY_X }, // X/BLUE
{ 3, KEY_Z }, // Y/GREEN
{ 5, KEY_A }, // L Shoulder
{ 16, KEY_S }, // R Shoulder
{ 6, KEY_ESC }, // EXIT ROM
{ 12, KEY_ENTER }, // START
{ 13, KEY_SPACE }, // PAUSE

```

After editing, compile, install the code and reboot with:

```

make retrogame
sudo mv retrogame /usr/local/bin
sudo reboot

```

When the Raspberry Pi boots back up, it should automatically launch Emulation Station. Your four buttons should all be working now.

The five button on the PiTFT are ordered from left to right:

1. L shoulder(a key)
2. Exit ROM (esc key)
3. Start (enter key)
4. Pause (spacebar)
5. R shoulder(s key)

## Configuring RetroArch Emulators Controls

Editing the retrogame file configures the buttons for the EmulationStation input — it does NOT transfer to each emulator, but you can configure the **Global** settings - that are settings which should apply to all emulators.

```
sudo nano /opt/retropie/configs/all/retroarch.cfg
```

Scroll down and look for the #Keyboard Input block.

```
# Keyboard input, Joypad and Joyaxis will all obey the "nul" bind, which disables
# rather than relying on a default.
```

```
input_player1_a = ctrl  
input_player1_b = alt  
input_player1_y = z  
input_player1_x = x  
input_player1_start = enter  
input_player1_select = space  
input_player1_l = a  
input_player1_r = s  
input_player1_left = left  
input_player1_right = right  
input_player1_up = up  
input_player1_down = down
```

Now most of your four button type game systems (mainly SNES) should be linked to the right controls on the Raspberry Pi.

# 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](http://adafru.it/rA3)(<http://adafru.it/rA3>), then [write this to an SD card](http://adafru.it/dDL) (<http://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



### USB 2.0 and Ethernet Hub - 3 USB Ports and 1 Ethernet

PRODUCT ID: 2909

One can never have enough socks, or USB ports. Add some more USB and

Ethernet capability to your Raspberry Pi or, really, any kind of computer with this USB 2.0 and Ethernet Hub...

<http://adafru.it/tBf>

\$14.95

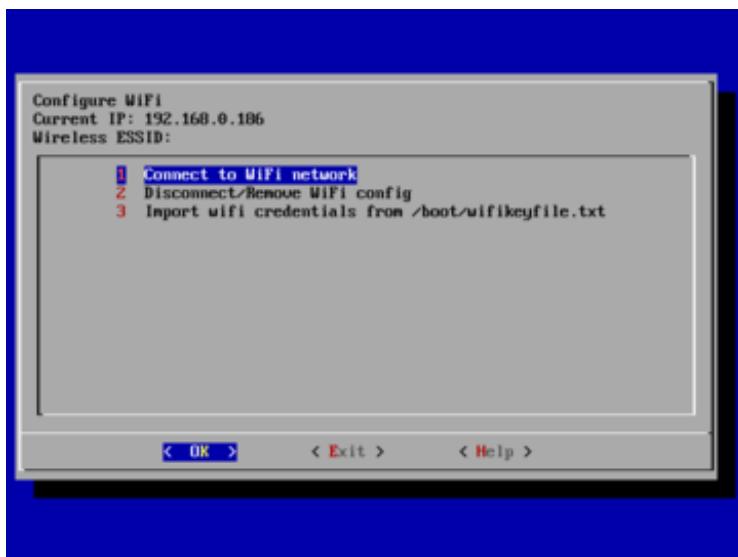
IN STOCK

## 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 “raspberry,” 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 -O https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/pitft-fbcp.sh  
sudo bash pitft-fbcp.sh
```

Select the “Pocket PiGRRL” option, which sets up various system parameters to match this project.

```
pi@retropie: ~
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: 2

Device: pitft22
HDMI framebuffer rotate: 0
TFT MADCTL rotate: 270

CONTINUE? [y/N] y

Starting installation...
Updating package index files...
Get:1 http://mirrordirector.raspbian.org jessie InRelease [14.9 kB]
0% [1 InRelease 0 B/14.9 kB 0%] [Connecting to archive.raspberrypi.org]
```

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-fbcm-master/build
Scanning dependencies of target fbcm
[ 50%] Building C object CMakeFiles/fbcm.dir/main.c.o
[100%] Linking C executable fbcm
[100%] Built target fbcm
Configuring PiTFT...
Done.

Settings take effect on next boot.

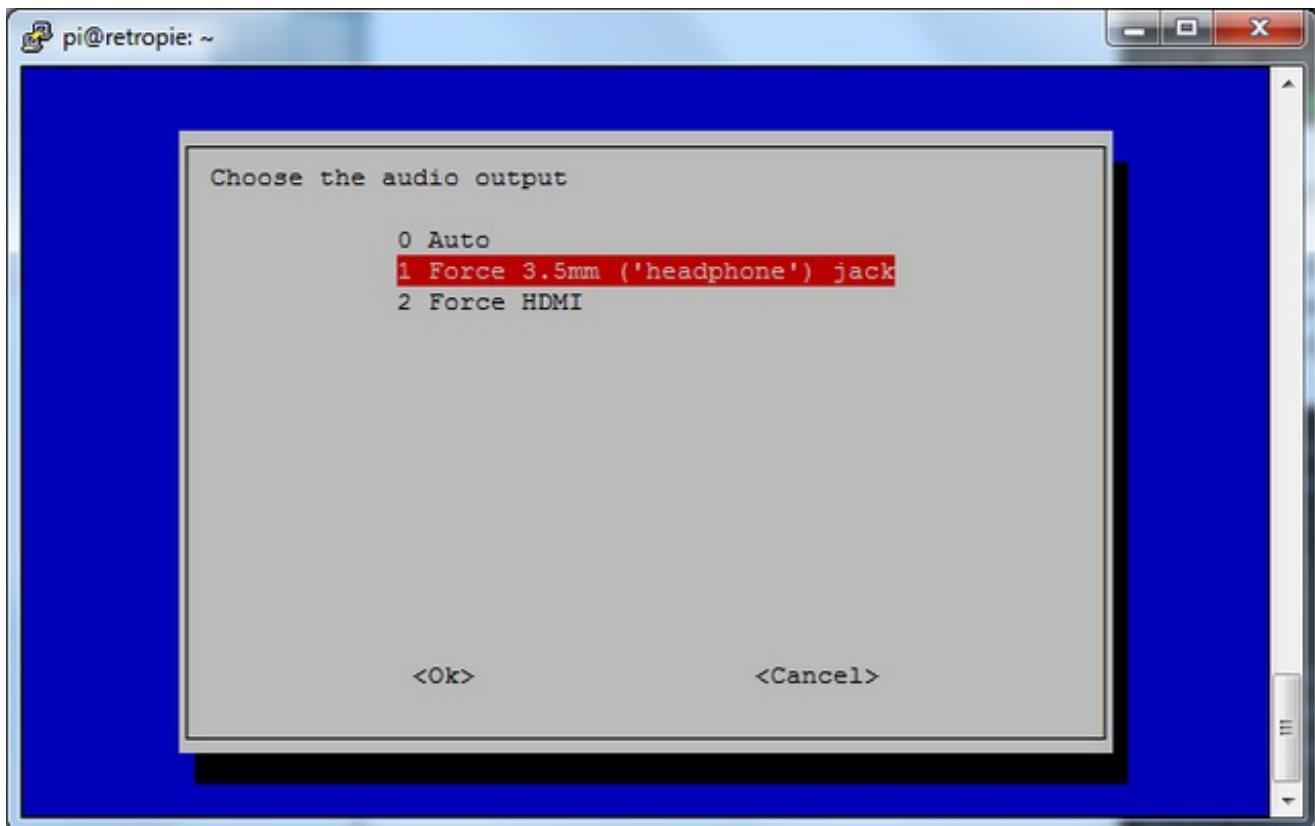
REBOOT NOW? [y/N] N
Exiting without reboot.
pi@retropie: ~ $
```

# 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

```
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)

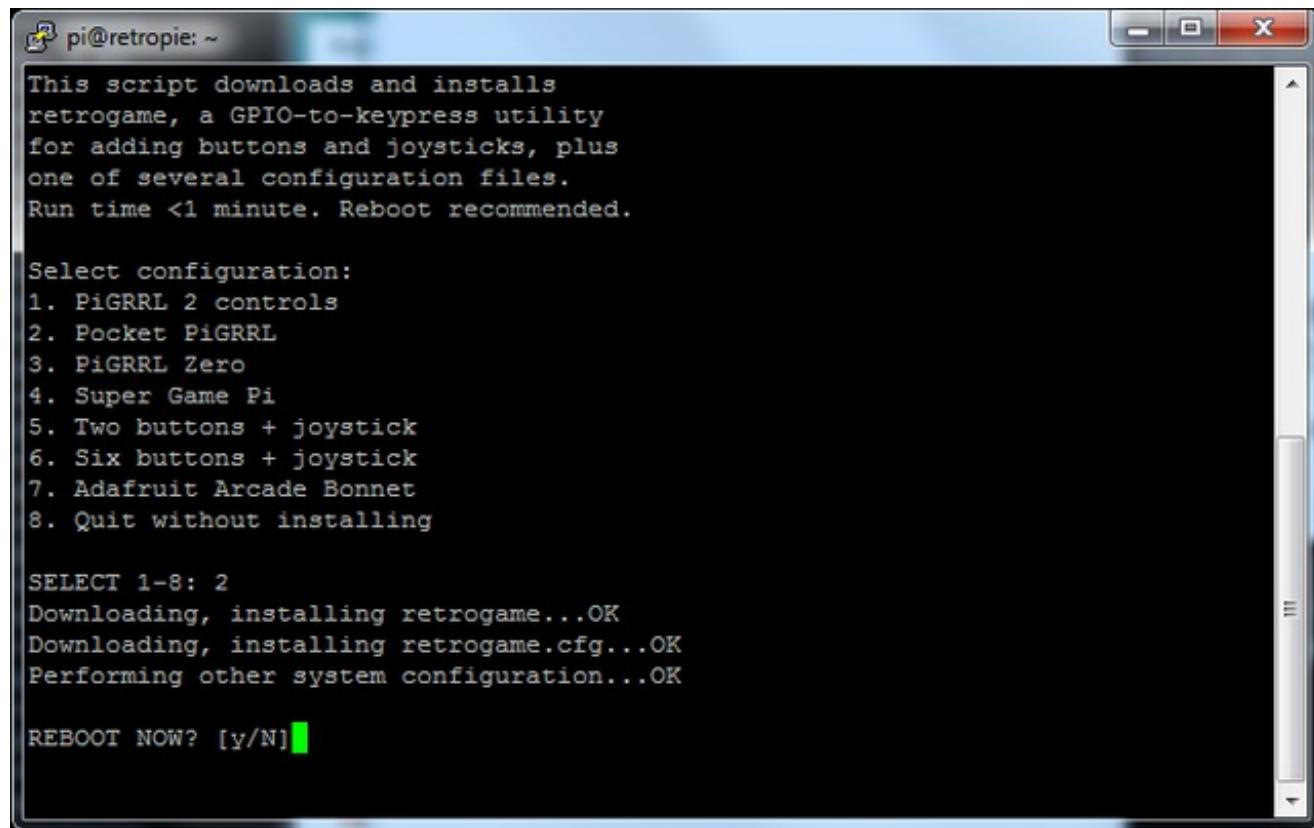


# Installing Keypress (retrogame) support

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

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

Again, select the “Pocket PiGRRL” 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: 2  
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.

# Configuration

## Uploading ROMs

Check out the Wiki on the RetroPie Setup page for how to upload ROMs:

[RetroPie - Uploading ROMs](#)

<http://adafru.it/fbb>

## Exiting ROMs

Retropie has changed how to exit out of ROMs/Emulators. Hold down the **Pause & Start** buttons at the same time to exit.

## Remapping Controls

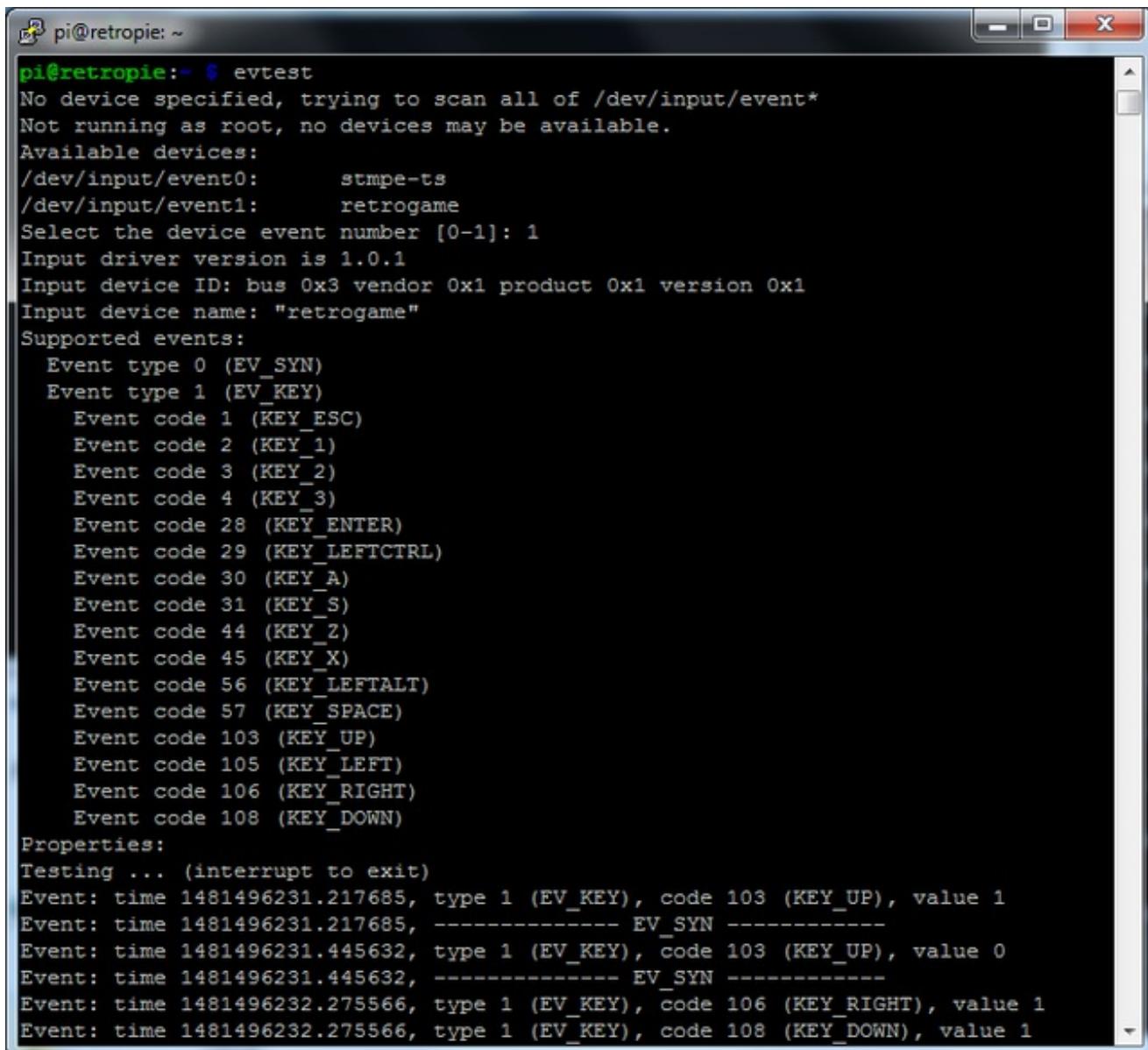
The buttons are premapped to work with NES and SNES emulators. If you'd like to remap the controls, you'll need to modify the **retrogame.cfg** file in /boot.

Make sure the Raspberry Pi is configured for Wifi. Open up terminal and ssh into the Pi. (Default username: **pi** password:**raspberry**)

[Then follow the retrogame tutorial to set it up the way you like!](#) (<http://adafru.it/sct>)

## Testing Controls

If you log in (via SSH or F4 shell) you can see exactly what keypresses are detected by running **evtest** then select 1 (or whatever **retrogame** is numbered):



```
pi@retropie: ~ $ evtest
No device specified, trying to scan all of /dev/input/event*
Not running as root, no devices may be available.
Available devices:
/dev/input/event0:      stmpe-ts
/dev/input/event1:      retrogame
Select the device event number [0-1]: 1
Input driver version is 1.0.1
Input device ID: bus 0x3 vendor 0x1 product 0x1 version 0x1
Input device name: "retrogame"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 1 (KEY_ESC)
    Event code 2 (KEY_1)
    Event code 3 (KEY_2)
    Event code 4 (KEY_3)
    Event code 28 (KEY_ENTER)
    Event code 29 (KEY_LEFTCTRL)
    Event code 30 (KEY_A)
    Event code 31 (KEY_S)
    Event code 44 (KEY_Z)
    Event code 45 (KEY_X)
    Event code 56 (KEY_LEFTALT)
    Event code 57 (KEY_SPACE)
    Event code 103 (KEY_UP)
    Event code 105 (KEY_LEFT)
    Event code 106 (KEY_RIGHT)
    Event code 108 (KEY_DOWN)
Properties:
Testing ... (interrupt to exit)
Event: time 1481496231.217685, type 1 (EV_KEY), code 103 (KEY_UP), value 1
Event: time 1481496231.217685, ----- EV_SYN -----
Event: time 1481496231.445632, type 1 (EV_KEY), code 103 (KEY_UP), value 0
Event: time 1481496231.445632, ----- EV_SYN -----
Event: time 1481496232.275566, type 1 (EV_KEY), code 106 (KEY_RIGHT), value 1
Event: time 1481496232.275566, type 1 (EV_KEY), code 108 (KEY_DOWN), value 1
```

Make sure each keypress works and matches the table below!

LEFT Pin 7 GPIO 4

UP Pin 36 GPIO 16

RIGHT Pin 35 GPIO 19

DOWN PIN 37 GPIO 26



# 3D Printing

## Two Button Version



## Four Button Version



## Parts

The parts are optimized to print as is, oriented centered on your printers build plate. The parts will fit on any printer with a minimum bed size of 120mm x 70mm.

[Download Files](#)

<http://adafru.it/f6Z>

File Name	Settings	Time to Print
pgp-top.stl	220c 2 shells 10% infill 50/60 speeds	1hr
pgp-bot.stl	-	1hr 10mins
	ninjaflex @230	

PiTFT-Buttons.stl	2 shells 20% infill 30/40 speeds	about 15 minutes
ab-buttons.stl	made for two Ninjaflex buttons	about 15 minutes
dpad.stl	Makes the DPAD (Up, Down, Left and Right) can also be printed in PLA or Ninjaflex	about 10 minutes
pgp-4-top.stl	used to make four-button version top case	about 1 hour
pgp-4-bot.stl	used to make four-button version bottom case	about 1hr 15min
flex-4btn.stl	used to make four ninjaflex button version	about 6 minutes
hard-btn.stl	used to make four hard PLA buttons	about 3 minutes each

The settings above are for reference. You're encouraged to slice these files for your printer using your preferred slicing software.

## Support material

No raft or support material necessary here. The overhangs are small enough that it doesn't present any problems with most FDM 3D printers.

## Tolerances

3D printers tend to vary from one another, so it's no surprise that one printer makes parts tighter than the other. That slight 0.1mm difference is enough to make things not fit exactly. Maybe the mounting holes are too tight, or the speaker doesn't fit. If that's the case, you can loosen them up with a filing tool or adjusting the faces in the CAD model.

# Materials

We tested the parts in PLA, ABS, BambooFill and CopperFill. You're can use whatever material and color you want to use in this project.

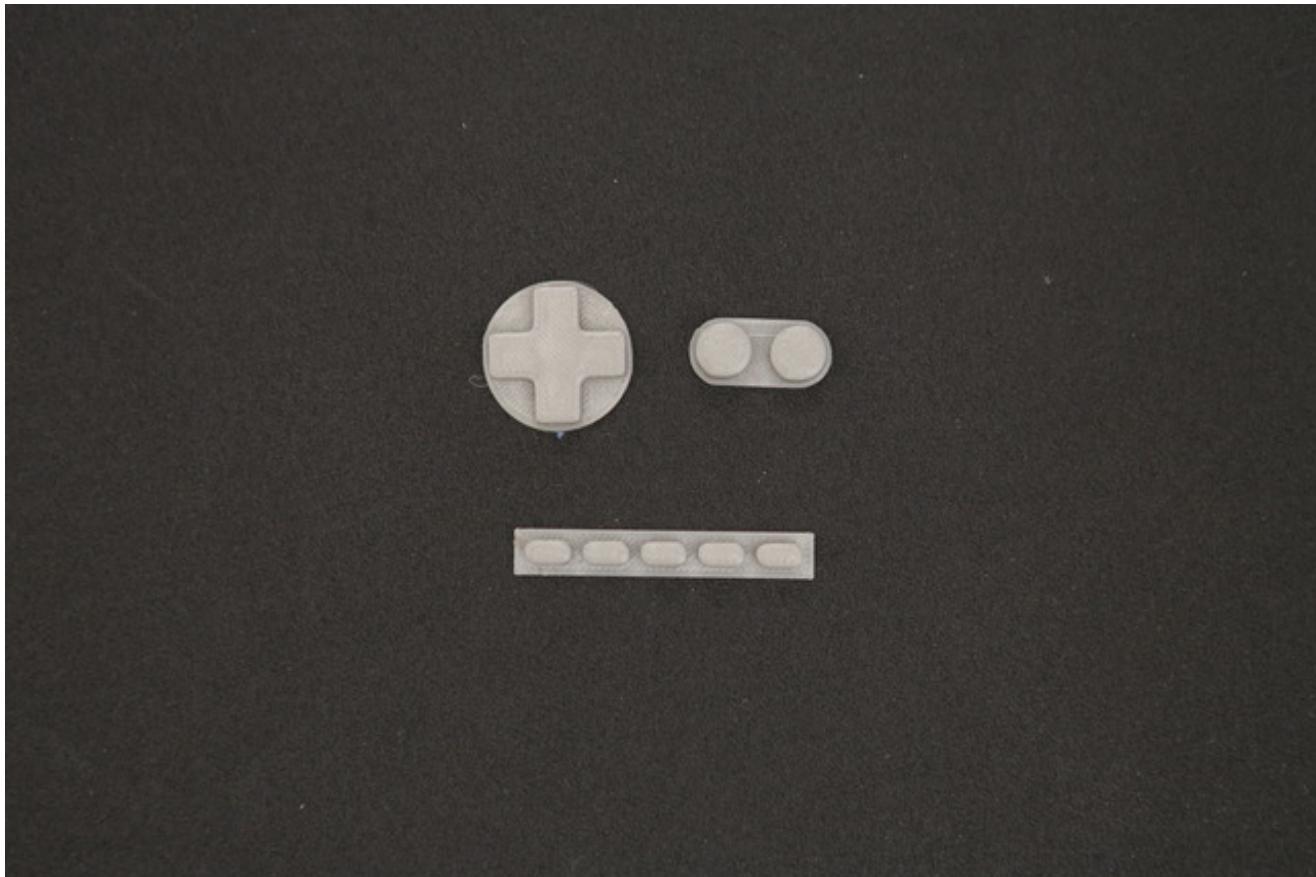
## Warping

Minimize warping by using blue painters tape. Apply gluestick or aqua net hair spray on a glass plate if applicable. If you're using a heated bed, be sure to enclose the printer to avoid air drafts.

## First layer + Bed Leveling

The first layer should be as thin as a sheet of paper. Kinda of hard to see but here's some stuff to look out for. If you see the layers are not fully bonding together, the nozzle is too far from the bed. If the layers are overlapping, the nozzle is too close to the bed.

Ideally, you want to baby sit the first layer to ensure your bed is leveled. If your printers bed can be adjusted with thumb screws, you'll want to perform leveling "live", while the print is taking place. If you're using a z-probe with auto-leveling, you'll have to manually offset your z-height in gcode.



## Ninjaflex buttons

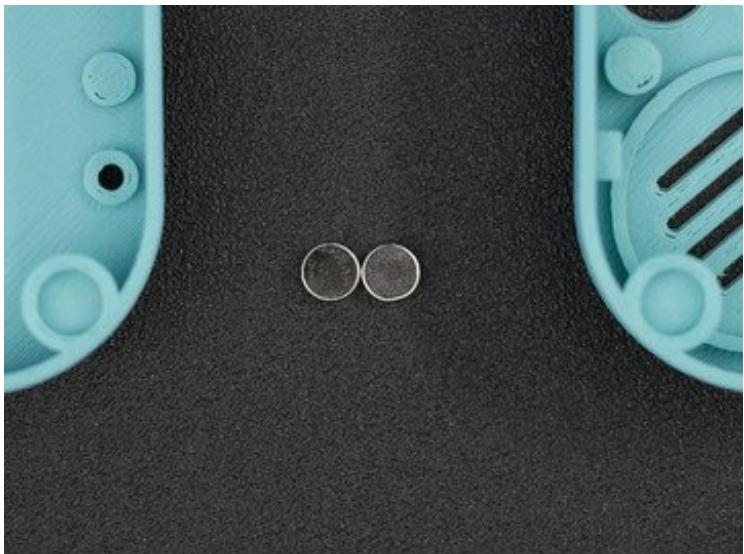
You can use TPE flexible filament for the buttons. The flexible filament works really well here because it gives you a bit of flex and grip when you press the buttons. PLA, ABS and other hard plastics will work just fine for the D-PAD but it might not work so well with the A+B and PiTFT buttons.

## Quality Prints

If your new to 3D printing and wondering: how did you get such a nice print? It's mainly because the bed was leveled really good and the slice tool path is clean.

When you're slicing parts, it's a good idea to check the toolpath and see how the nozzle is generating the walls in the part. Ideally you want to adjust your settings so that the walls are being printed with no infill and just the shells. This makes the wall appear really clean.

Don't be fool by the photos though, if you look carefully you can see we have imperfections and minor warped corners. We printed the blue one in ABS and the purple in PLA - The PLA parts even had a small amount of warping. The ABS part was printed on a makerbot replicator 1. Purple parts were printed on a Replicator 2. Both using makerware.

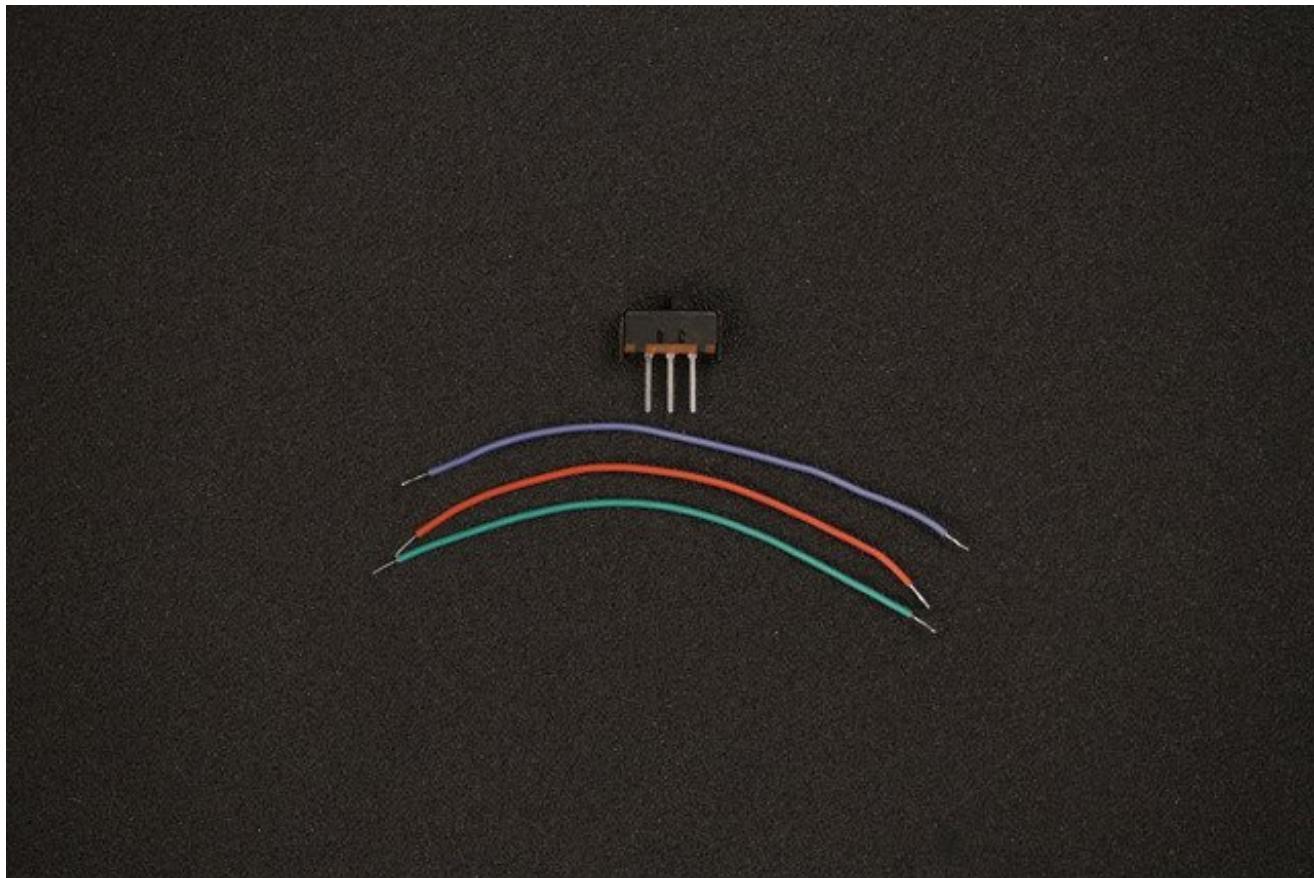


## Magnets

In order to keep these parts closed, we used two neodymium magnets to keep them shut. I recommend gluing these to the two enclosure part before starting wiring. These magnets fit inside the corner standoff located near the bottom rounded corner - you can't miss it! You can use super glue to keep them in place. Be sure to double check the polarity before gluing them!



# Wiring



## Prep Wiring

Before starting it's a good idea to get your tools and workspace suited. You'll want to have wires, tools and solder close by while you work through the sections.

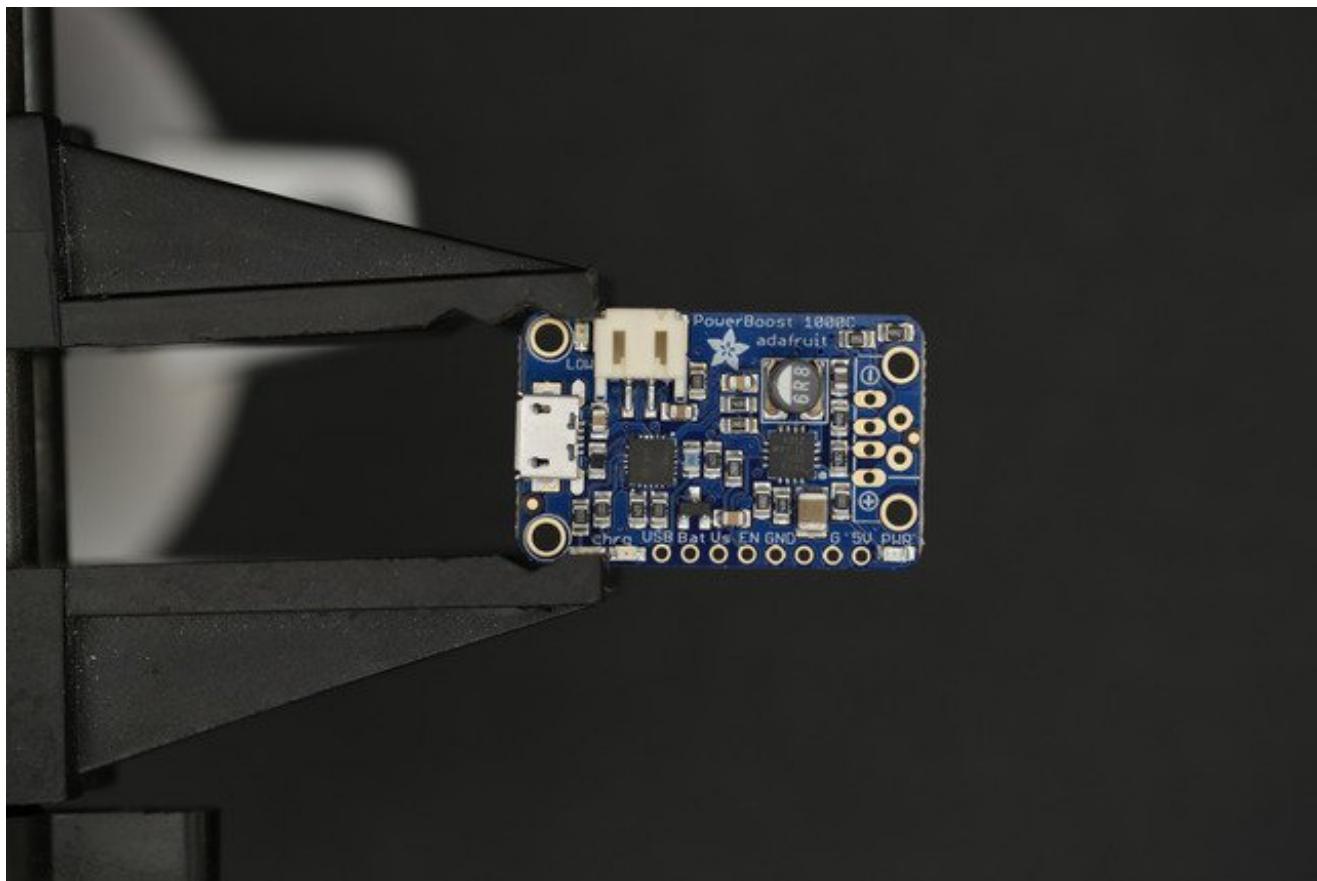
## Cut, Strip, Tin, Repeat

The wiring portion of this build is split into sections relative to the components. This hopefully makes it a good way to take breaks when completing sections.

## The Slide Switch

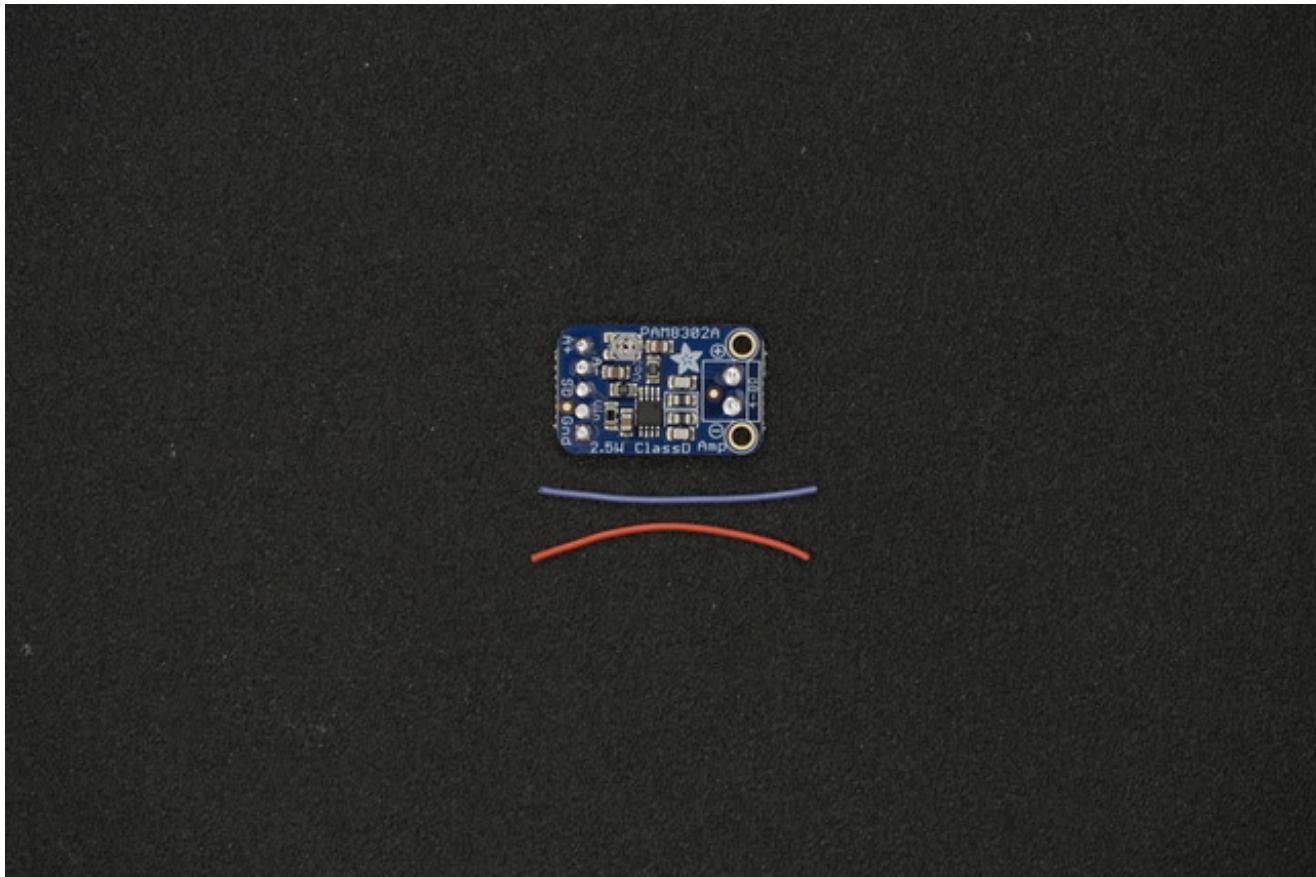
The slide switch is wired to the Powerboost 1000c with just three wires **V<sub>s</sub>, GND and EN**). It's wired to the enable pin so that it safely discharges the powerboost 1000c. This however

will not safely power off the Raspberry Pi - You'll still need to "sudo -halt p" that to do that safely.



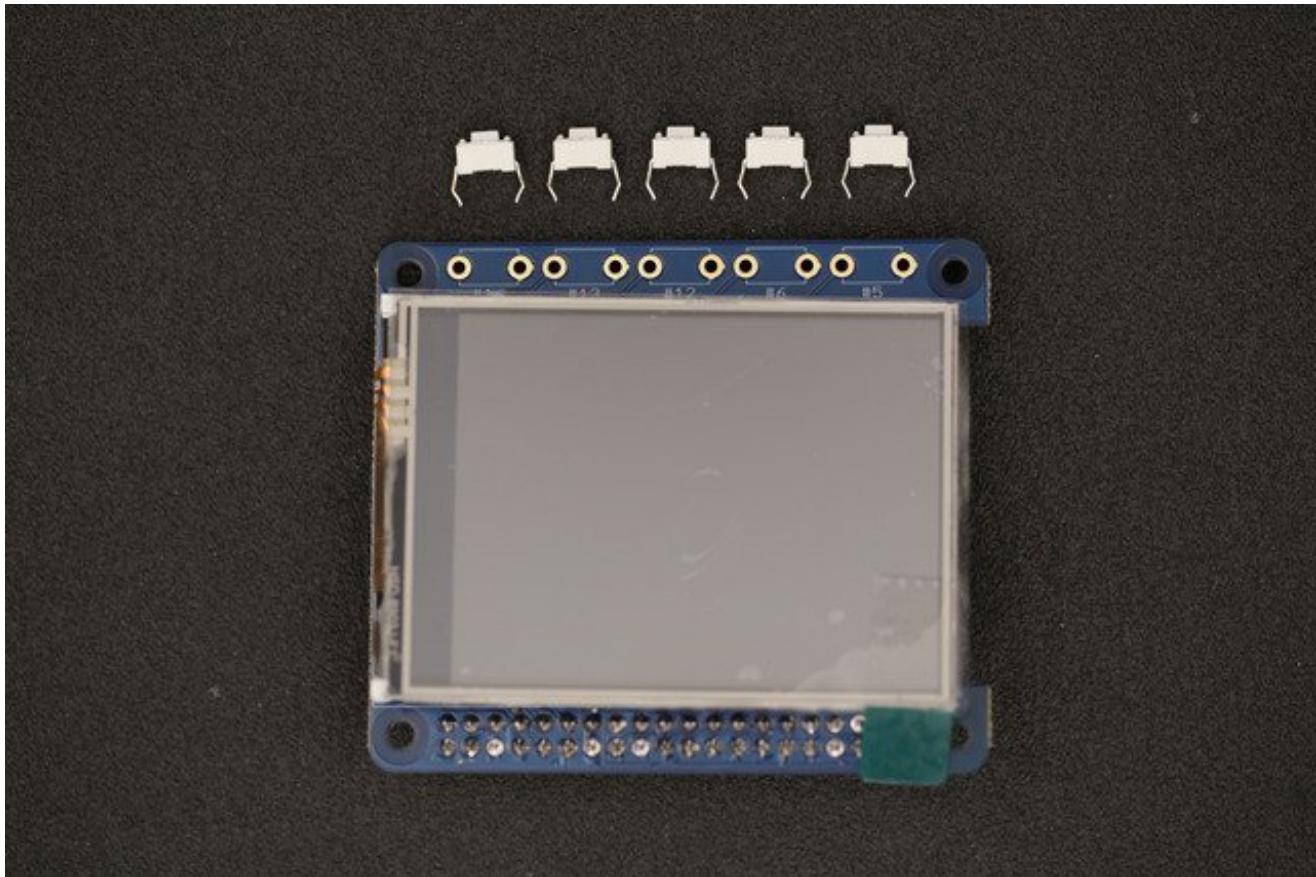
## Powerboost 1000C

The Raspberry Pi, PiTFT and PAM8302 are powered by the Powerboost 1000c and a 2000mAh lithium polymer battery. A Pi ribbon cable connects from the PiTFT to the positive and negative power pins on the powerboost breakout board (in place of the USB port). The microUSB port on the powerboost can be used to recharge the lipo battery. The lipo battery connects to the on-board JST conenctor.



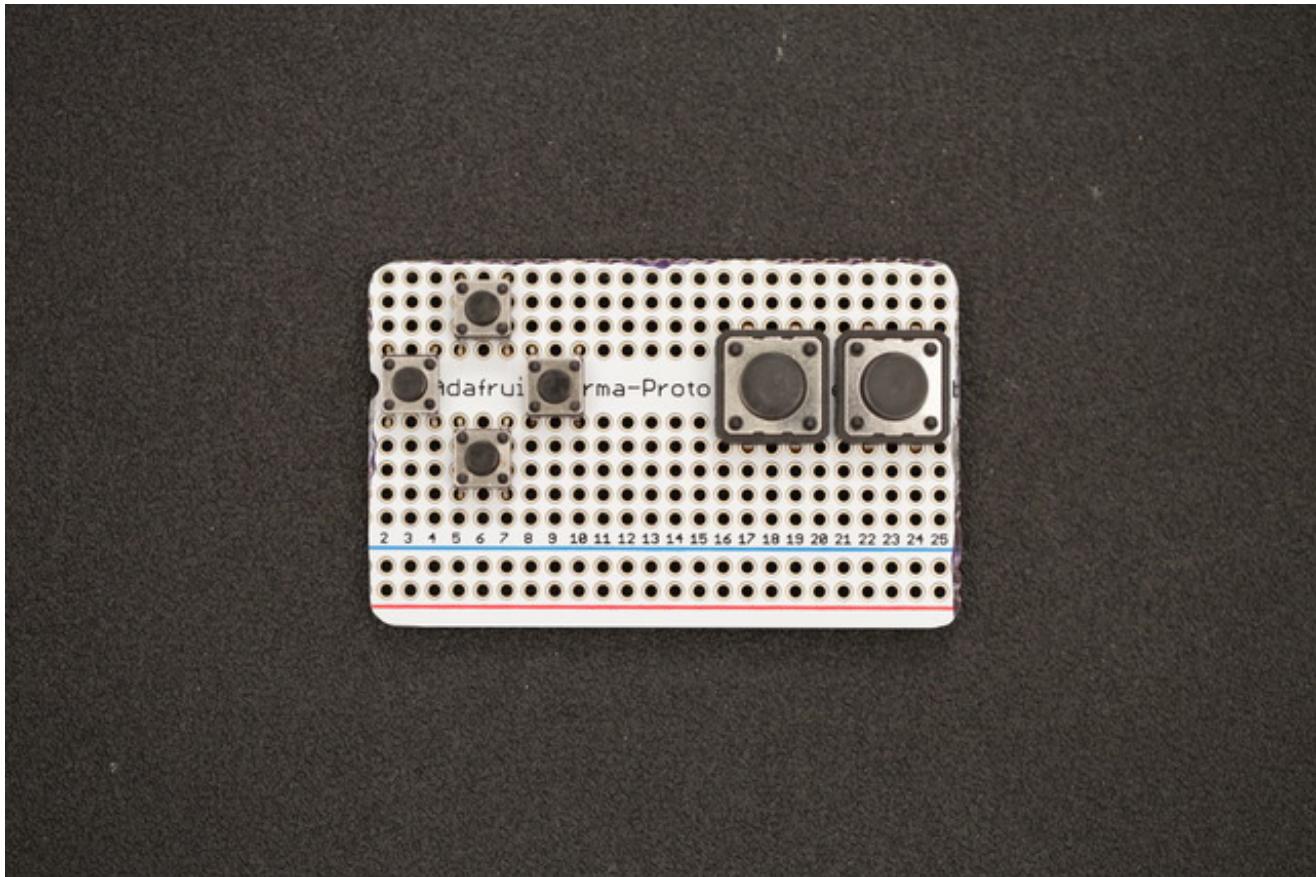
## PAM8302 & Speaker

This 2.5w mono amplifier is powered by the powerboost 1000c (5V and G). The audio input is connected to the audio jack pins on the bottom of the Raspberry Pi. A single mini metal speaker is wired to the output on the breakout.



## 2.4' PiTFT

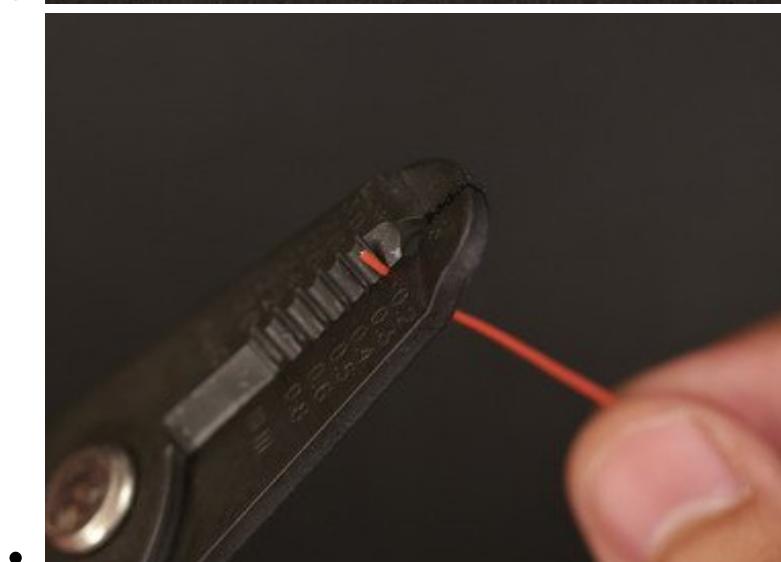
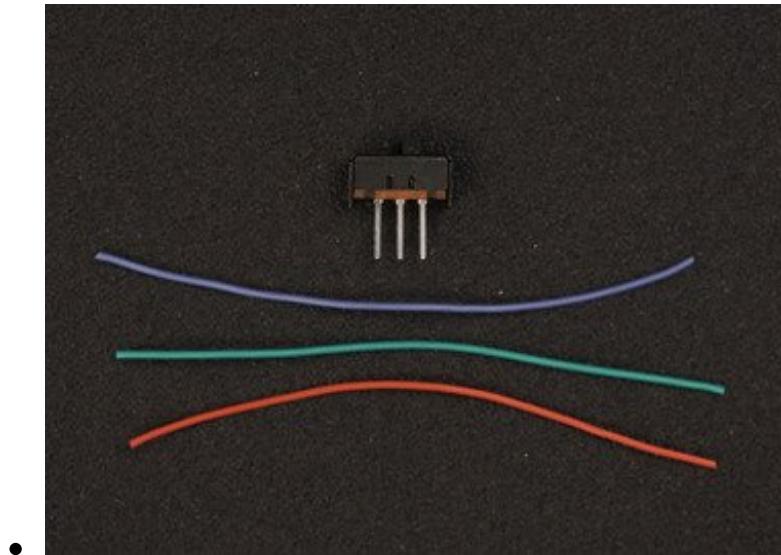
The PiTFT is designed to fit ontop of the Raspberry Pi GPIO header. The PCB on the display has a GPIO socket which can connect to a Pi ribbon cable. We'll use the Pi ribbon cable to wire up tactile buttons for the controls. The 2.4' PiTFT has five spots on the PCB for tactile buttons. Use these buttons as pause, start, L shoulder, R shoulder or even exit ROM. The display shares power with the Raspberry Pi, so that gets powered on when the Pi ribbon cable is wired up to the powerboost 1000c.



## Perma-proto

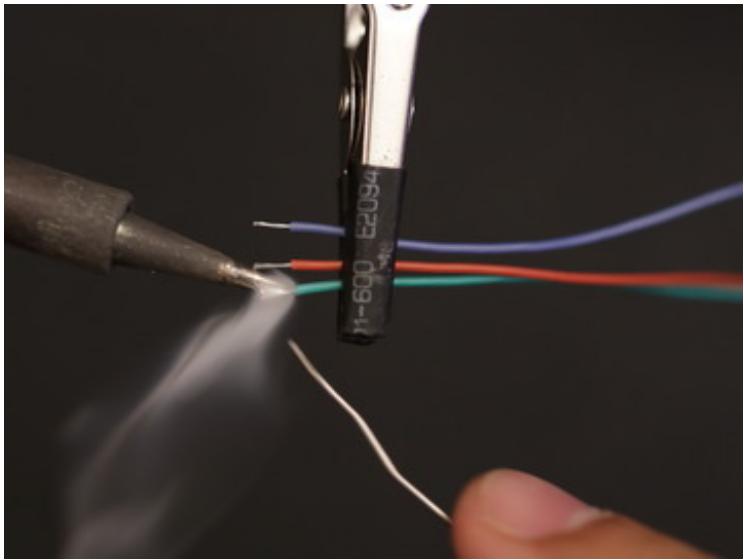
To connect our buttons we're using a perma-proto half-sized breadboard to contain them together on a PCB. The PCB will need to be cut and trimmed to fit inside the enclosure. The 6 controller buttons are soldered to the PCB. The Pi ribbon cable is wired to these buttons. The perma-proto has a power rail, so we can connect the ground together in series.

# Switch

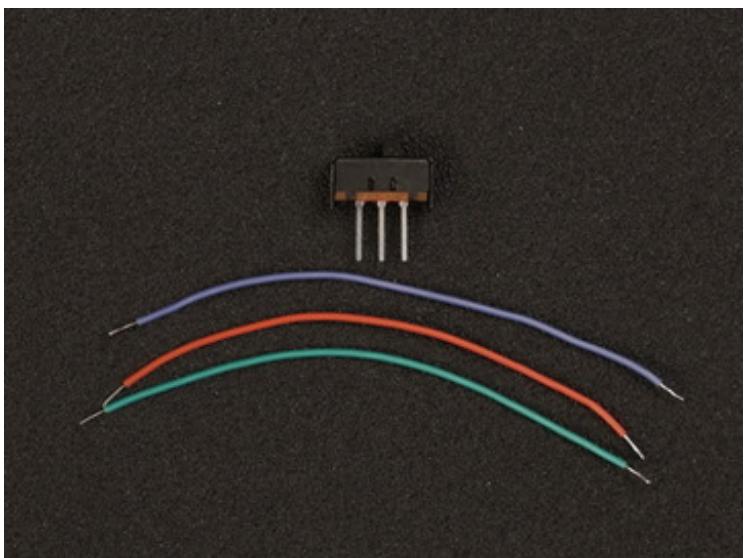


## Prepping Slide Switch

We need to prep the wires for the slide switch. You can use any colored wires. We used three different colored 30AWG silicone coated wires because it's thin, flexible and easier to indicate. Red for positive, blue for negative and green for enable. Measure three pieces of 30AWG wires and cut to approximately 6cm in length. Use wire stripper to

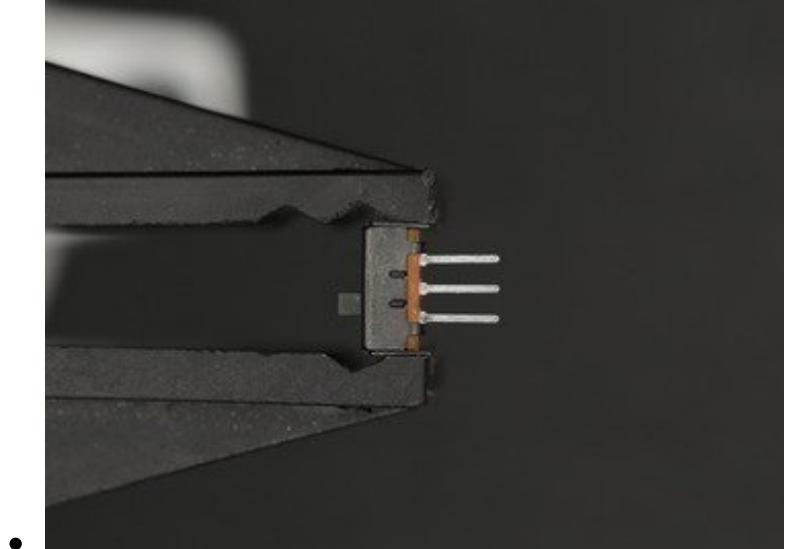


remove 3-4mm of insulation from the tips of each wire. Use a helping third hand to hold the pieces of wire while you apply solder to the tips.

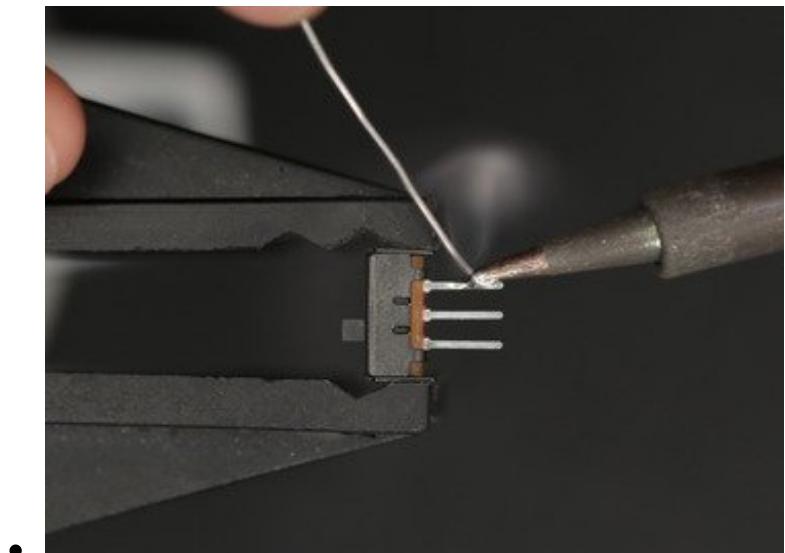


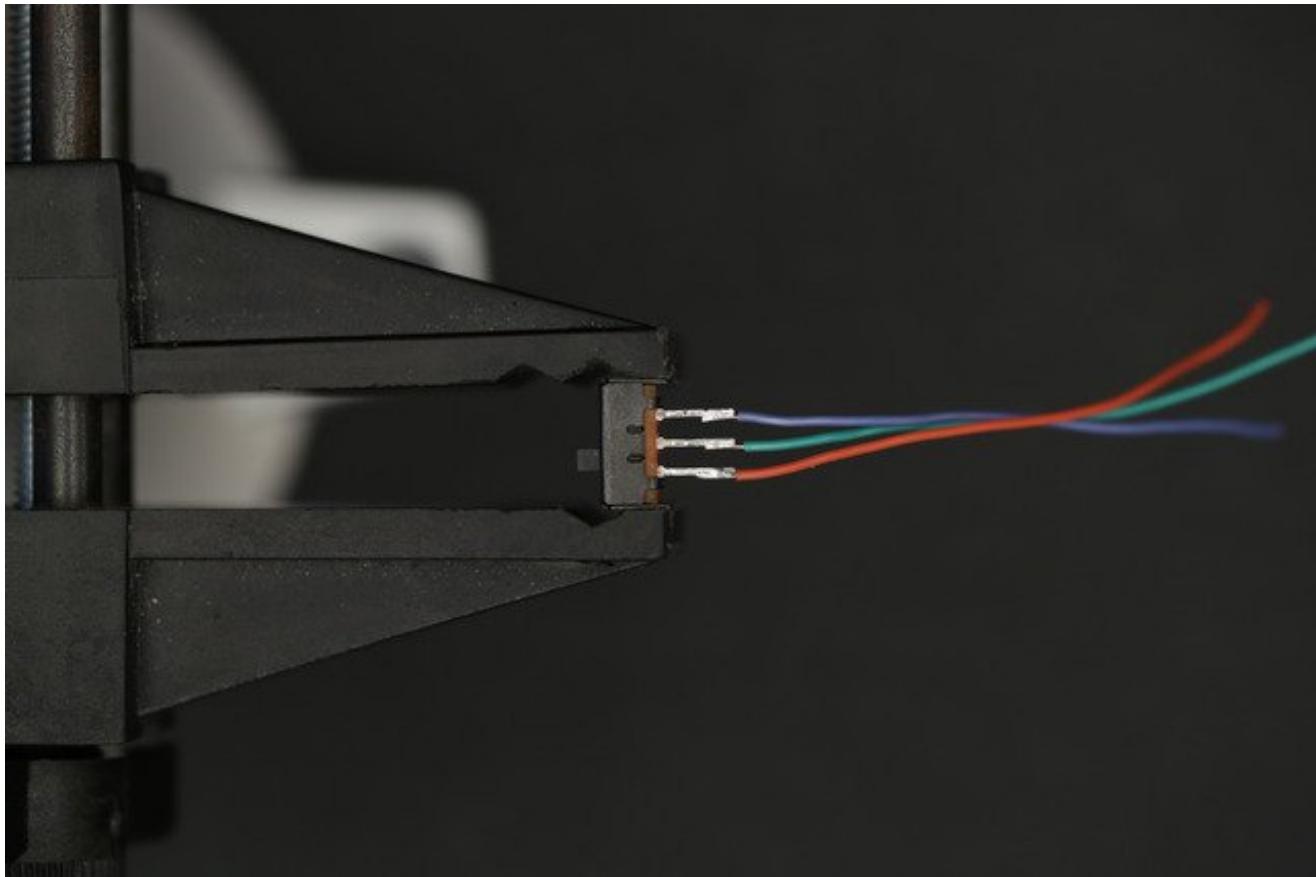
Skip the red wire, we don't need a third connection for the slide switch, only EN and GND.

## Secure Slide Switch



Secure the slide switch to a Panavise Jr. to hold it in place while you solder. Heat up the terminal lead on the slide switch with the tip of the soldering iron and lightly touch it with a strand of rosin-core solder. You'll need to tin the three terminals on the slide switch by applying a small amount of solder to them.

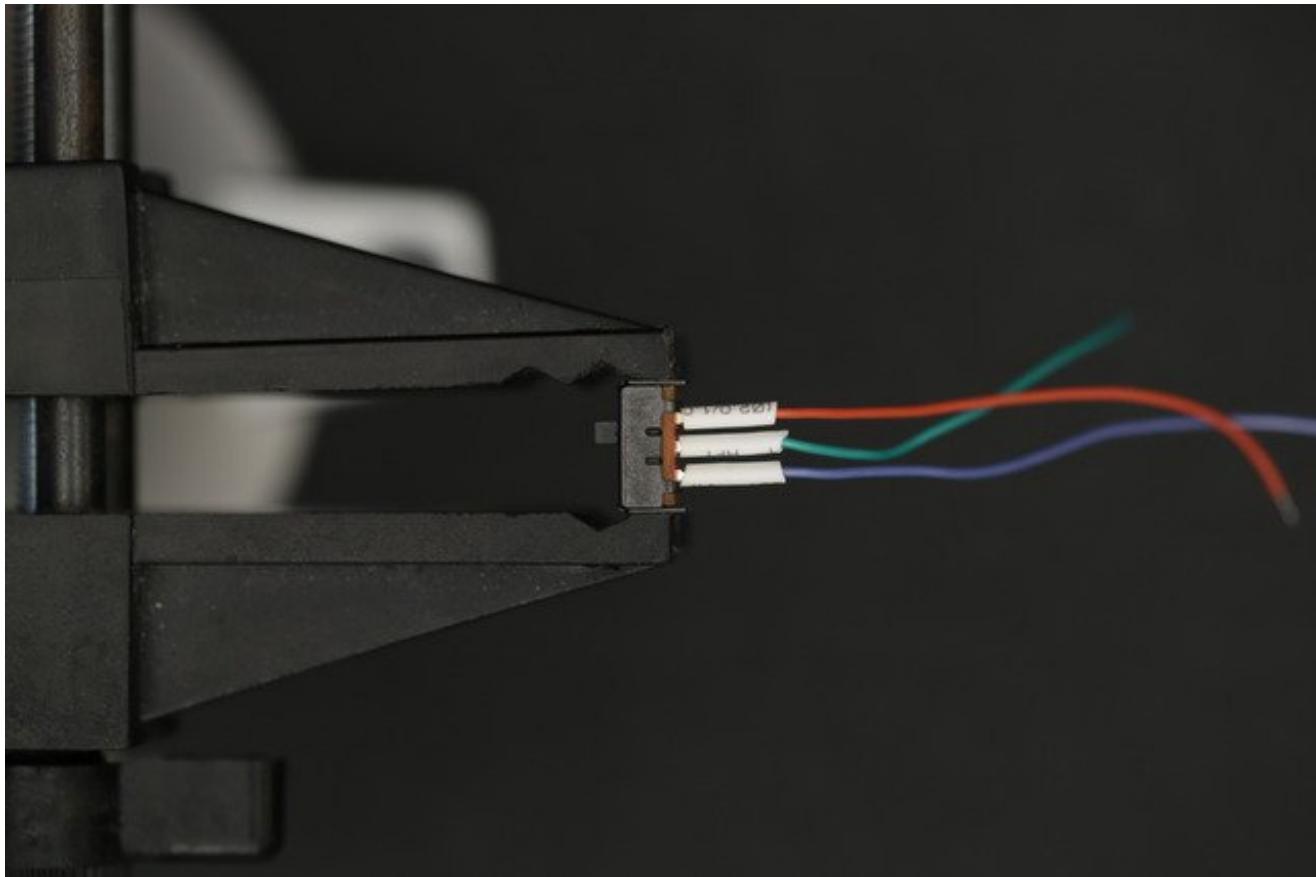




## Wire Slide Switch

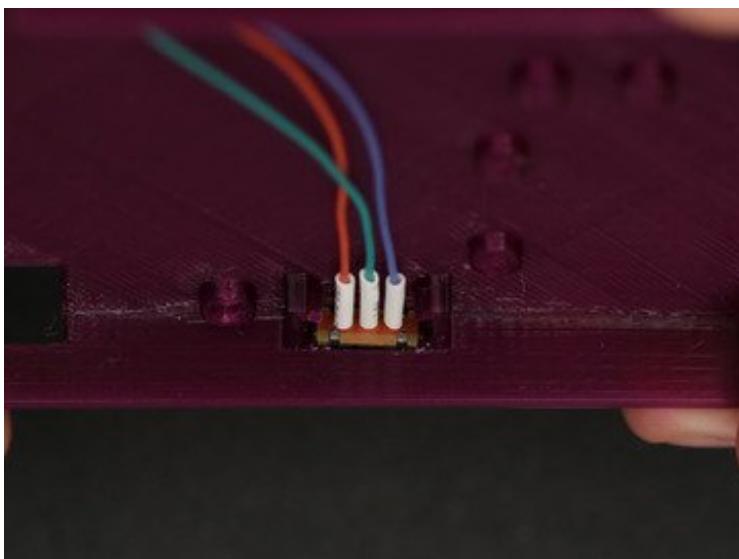
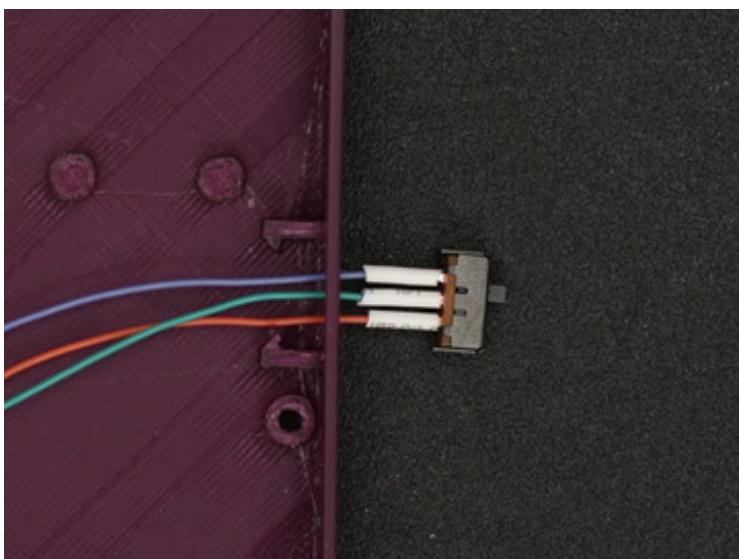
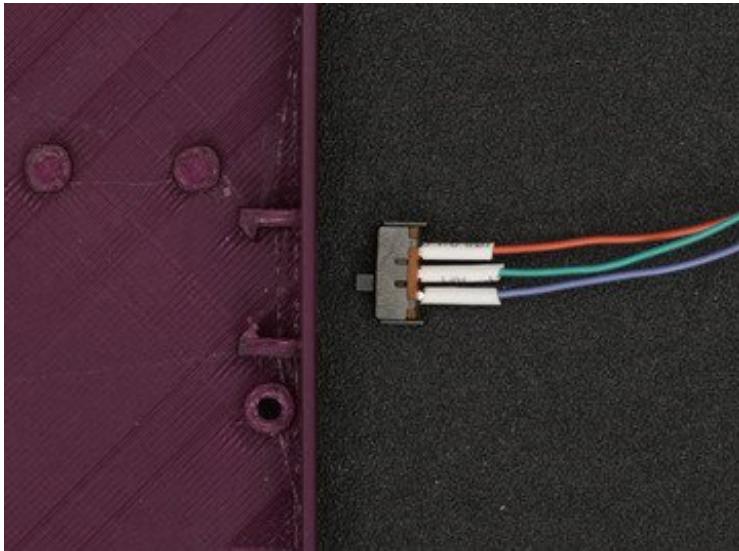
While the switch is secured, grab a prepped wire and bring it close to the first terminal on the slide switch. Position the soldered tip of the wire to the terminals and hold it there steadily. Press the soldering iron against the terminal and wire to solder them together. You'll want to do this gently and quickly. Ideally, you want the middle terminal be the "enable" while the far left/right be "ground", negative.

Again, skip the red wire, we don't need it!



## Heat Shrink Slide Switch

Once you've soldered up the three wires to the terminals, you'll want to protect the connections with some heat shrink tubing. Cut a piece of heat shrink tubing that sizes for 30AWG wire to about 5-6mm in length. Slip these onto the three wires and position them over the exposed connections. Apply some heat to the tubing to shrink! Ideally want to use hot air.



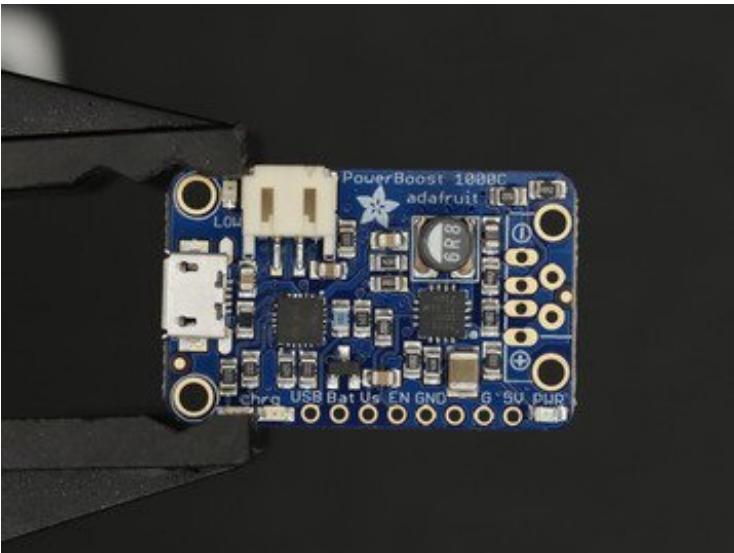
## Install Slide Switch

Now that the switch is wired up and protected, it's a good idea to install it into the enclosure before wiring it to the rest of the circuit. Carefully thread the into the slide switch cut out on the bottom enclosure part. Gently push the slide switch into the enclosure. You may want to use a hand tool or surface of a table to help force it in if the tolerances are too tight.

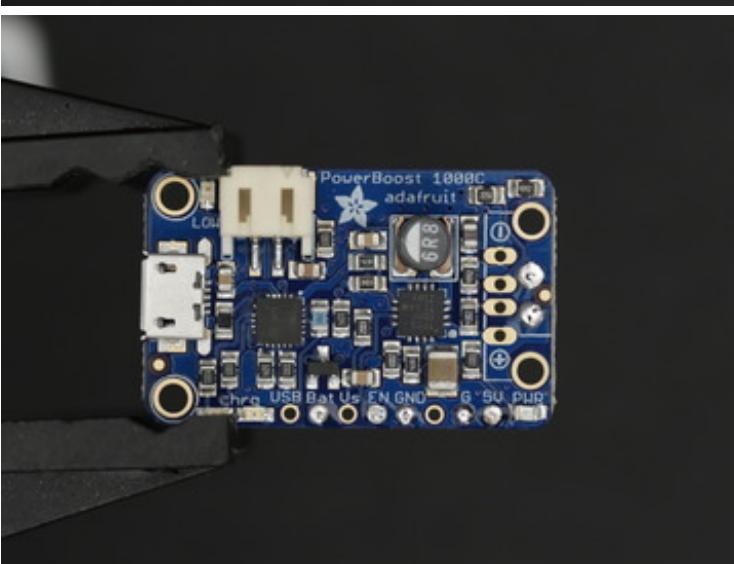


Ignore the red wire, we really don't need it for this project. We only need the Enable and Ground.

# Powerboost 1000C



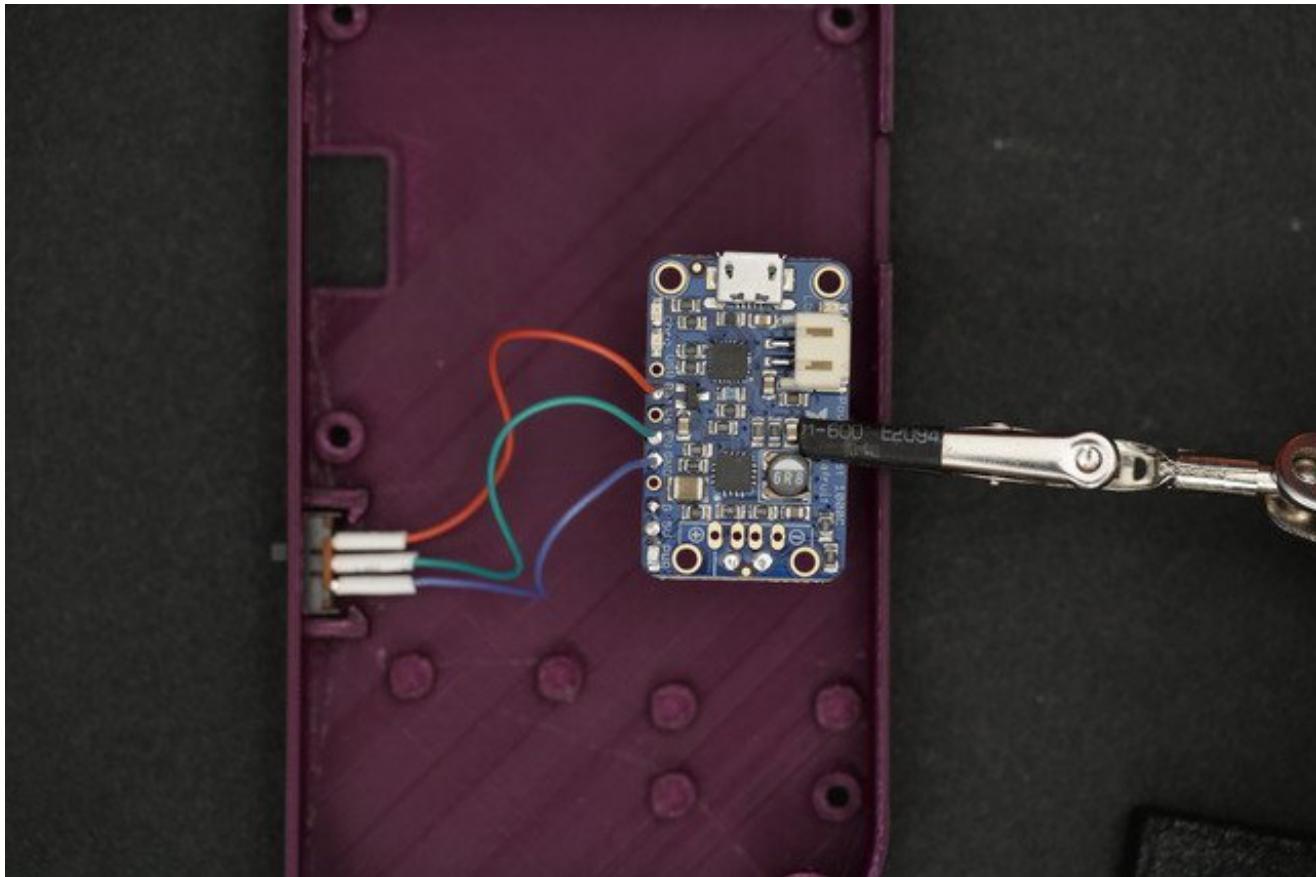
- 



- 

## Prep PowerBoost 1000C

Next up we need to wire up the slide switch to the powerboost 1000c breakout. Secure the PCB to the helping third hands. We need to tin three pins so we can insert the wires from the slide switch. Heat up a pin with the tip of the soldering iron and apply solder. Tin pins 5V, G, GND, EN (we also did VBat but it isn't required). The pins should look like they have little soldered beads.



## Wire Powerboost 1000C

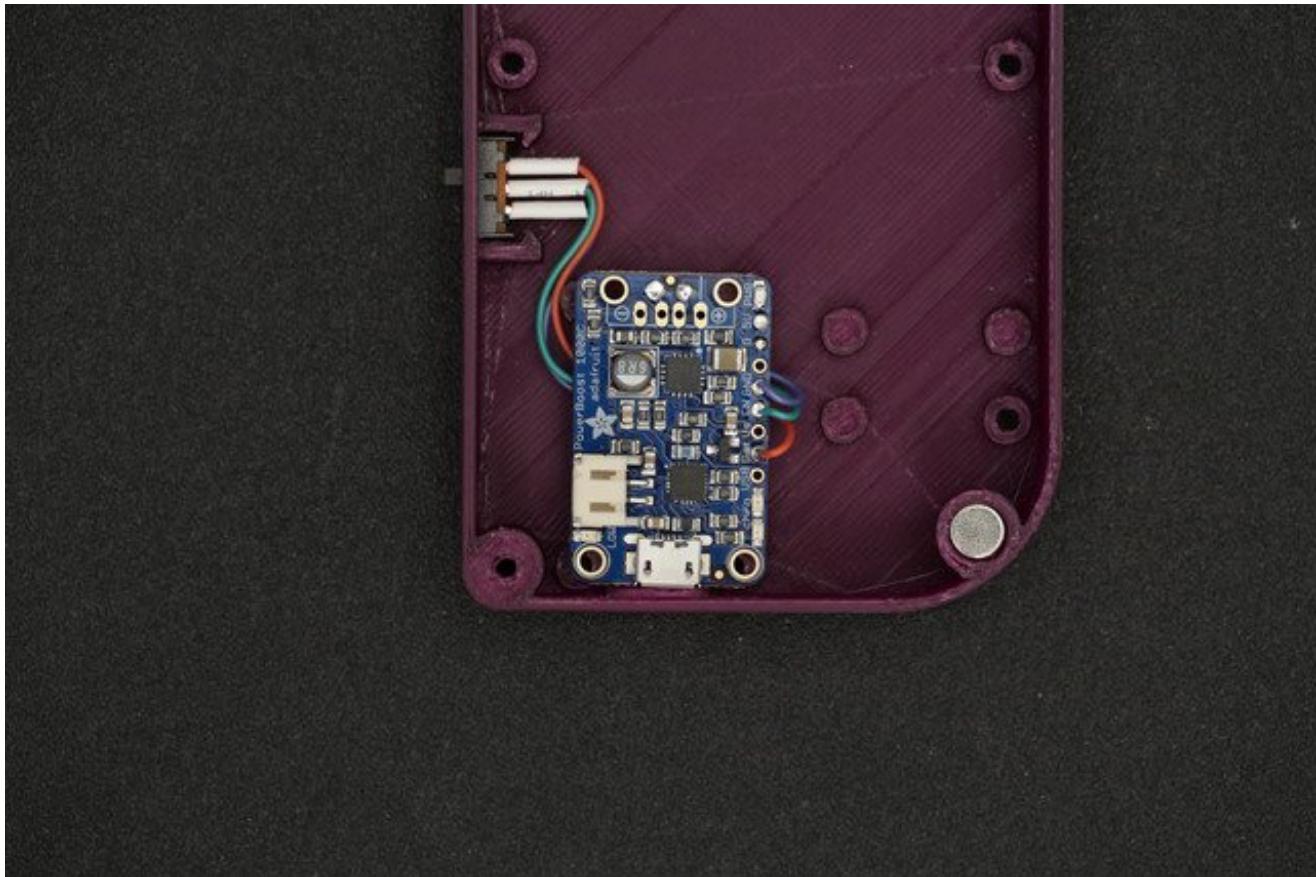
Secure the breakout to the third helping hands and position it close to enclosure. Heat up the pins on the breakout board with the soldering iron and insert the wire. You'll want to do this pretty quickly, about 5 seconds for each connection. Blue to **GND** and green to **EN**. You can **skip** the **red** wire, we don't need it.

The photo shows red wire to BAT, but you don't need to connect this wire, the circuit will function without the red wire.



## Thread Mounting Holes

It's a good idea to thread the holes on the PCB to make it easier to mount to the enclosure. #4-40 3/8 flat phillips machine screws work best.



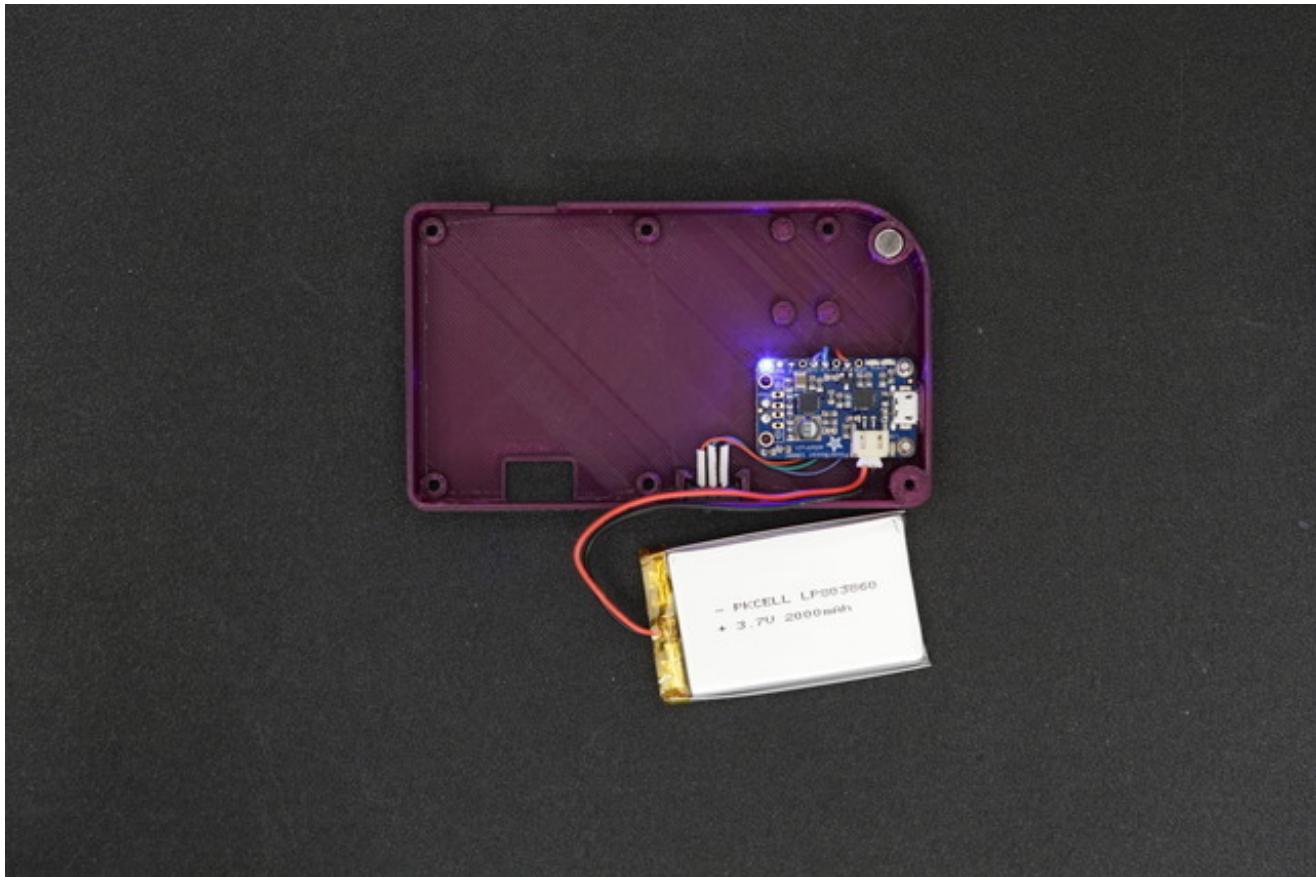
## Install PowerBoost 1000C

With the wires on the powerboost soldered, you can release it from the third helping hands. It's a good idea to tuck the wires underneath the board before mounting it. Position the breakout over the bottom enclosure with the USB port facing the cut out. You'll want to tuck the three wires from the slide switch underneath the PowerBoost 1000c. The stand-offs should give you enough clearance.



## Mount PowerBoost 1000C

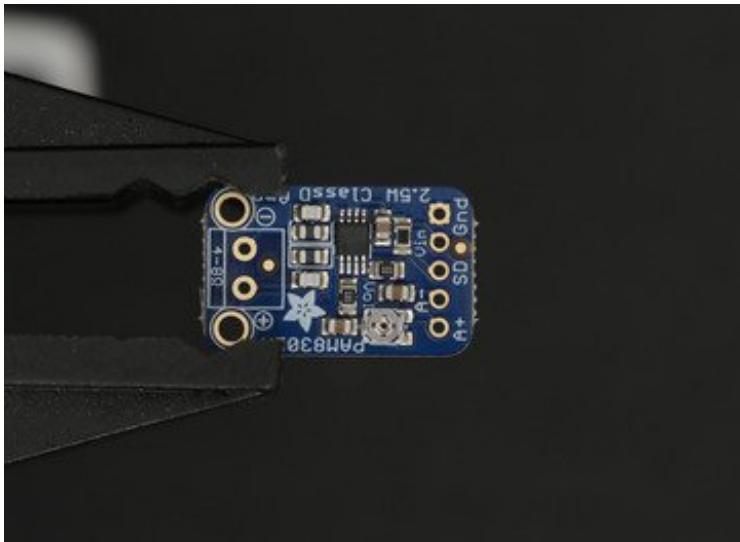
You'll need #4-40 3/8 Phillips flat machine screws to mount the board to the enclosure. Hold the breakout board in place while you insert and fasten two #4-40 machine screws into the enclosure.



## Test PowerBoost 100C

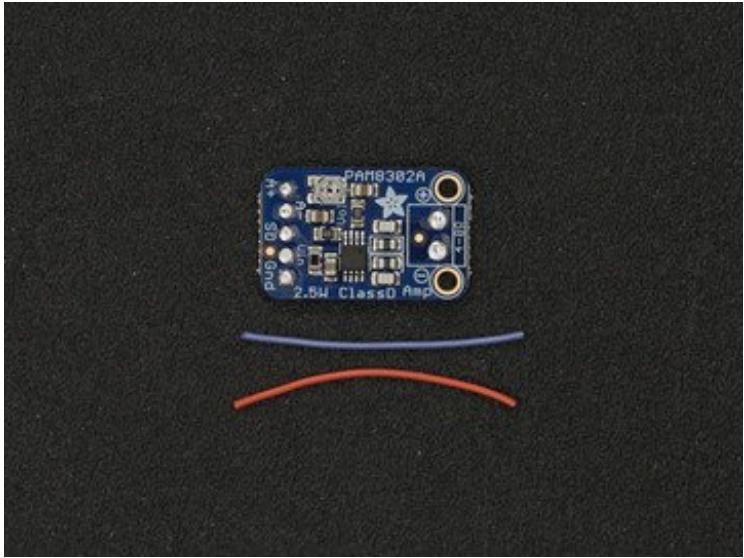
Plug in the JST cable from the rechargeable battery into the JST port. Flip the slide switch on to test if the circuit is working. The blue LED should turn on, indicating the battery is charged and ready. If everything lights up, disconnect the battery from the breakout board. If not, double check your wiring.

# PAM8302



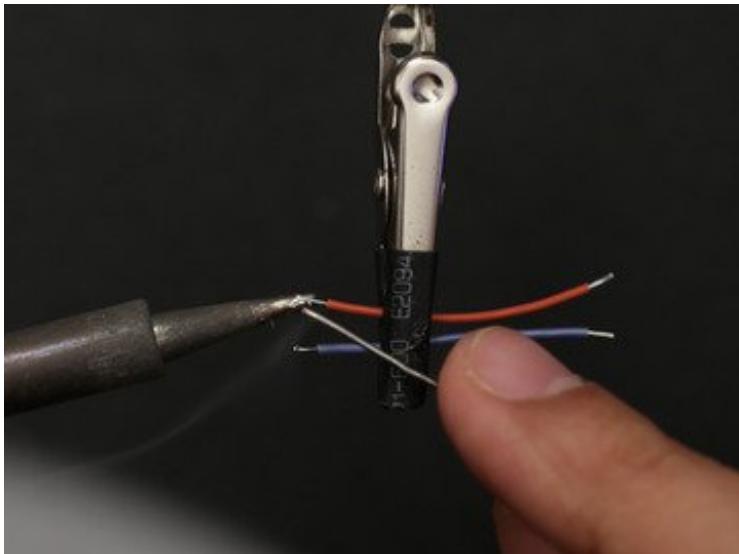
## Prep PAM8302

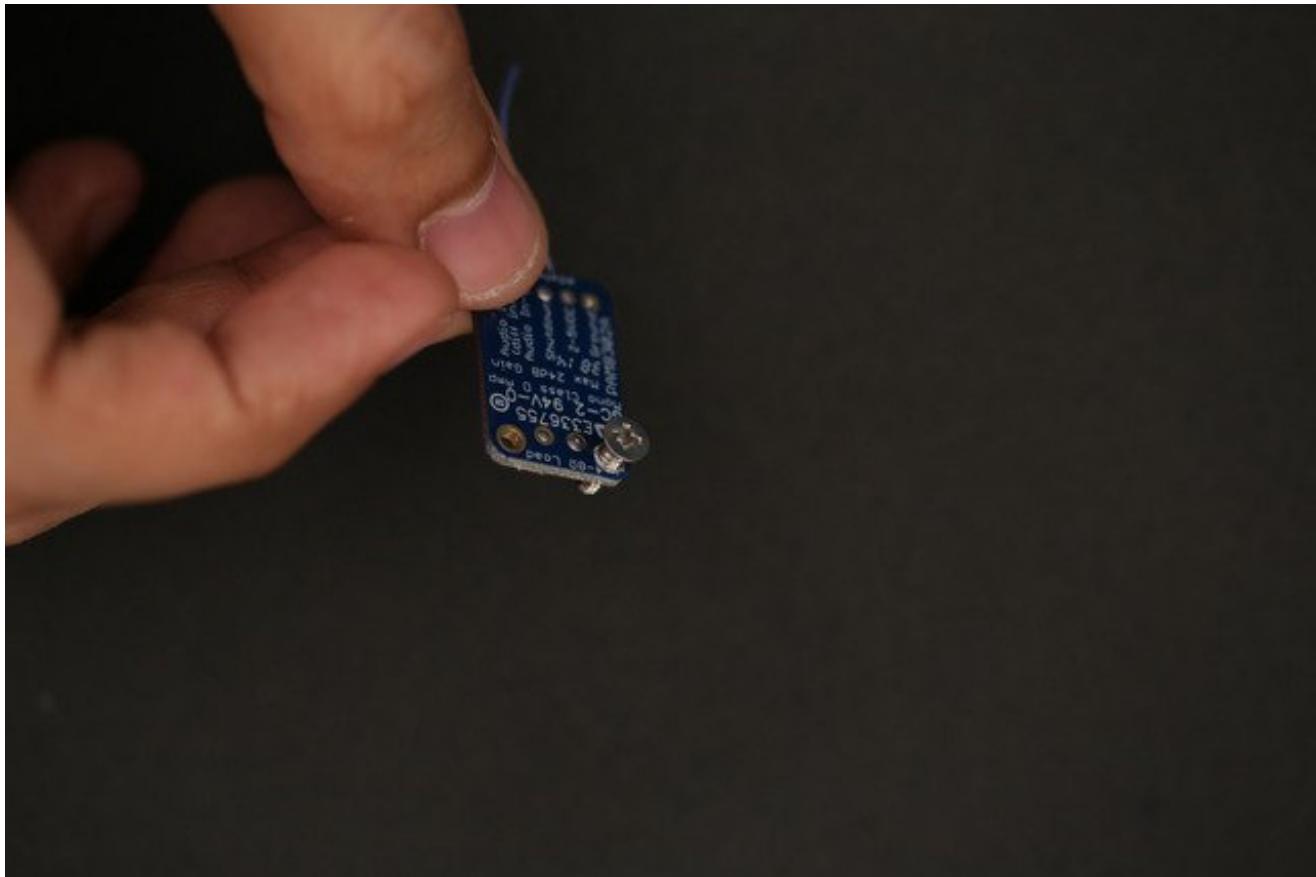
- Get the amp breakout secured to the panavise jr. Heat up the pins and apply some solder to all but the **SD** pin on the breakout.
- A close-up photograph showing a person's fingers holding a soldering iron and applying solder to the pins of the PAM8302 board. The board is held in place by a panavise Jr. clamp. The background is dark.



## PAM8302 Wires

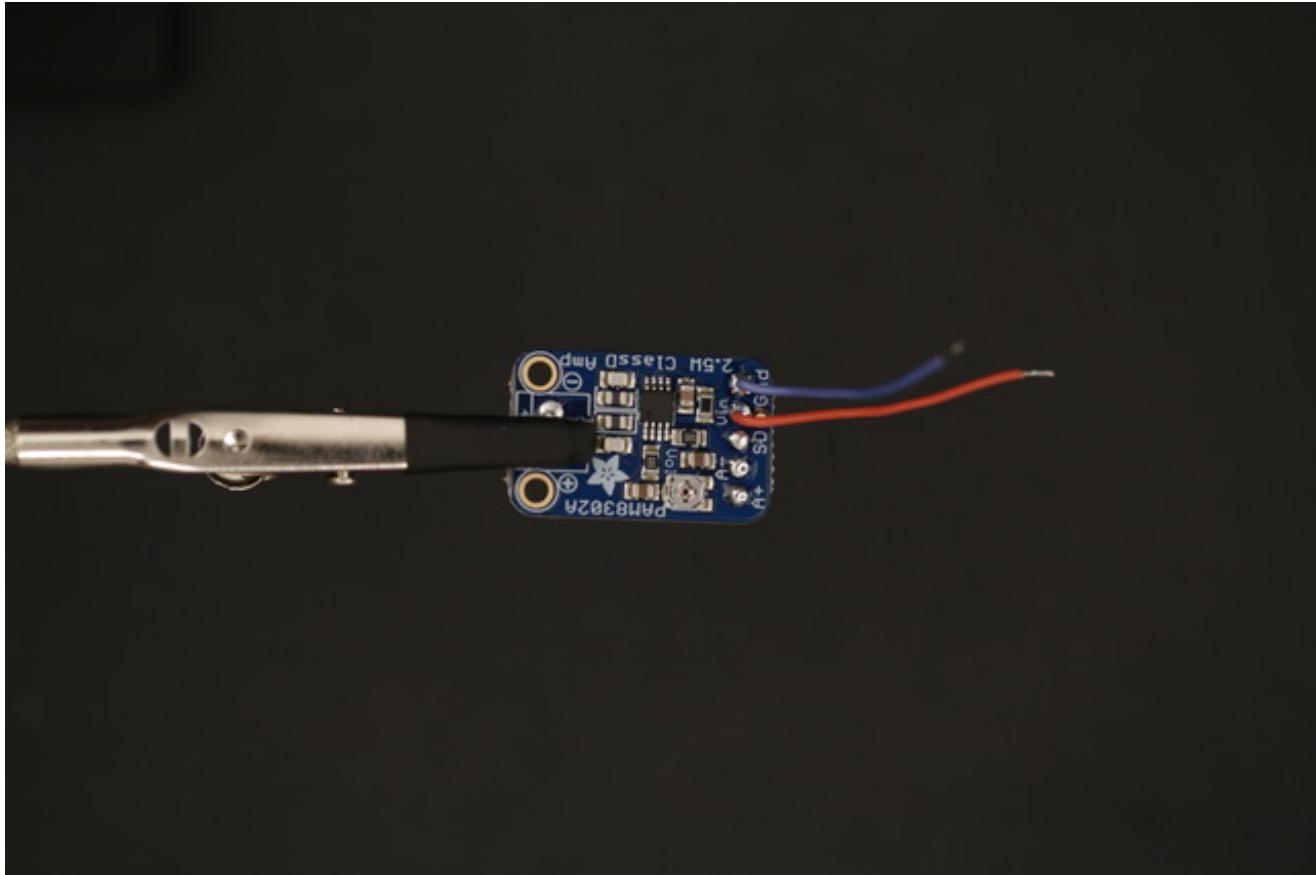
Measure and cut two 30AWG wires to about the length of the PCB. Strip the ends off each wire and tin the tips.





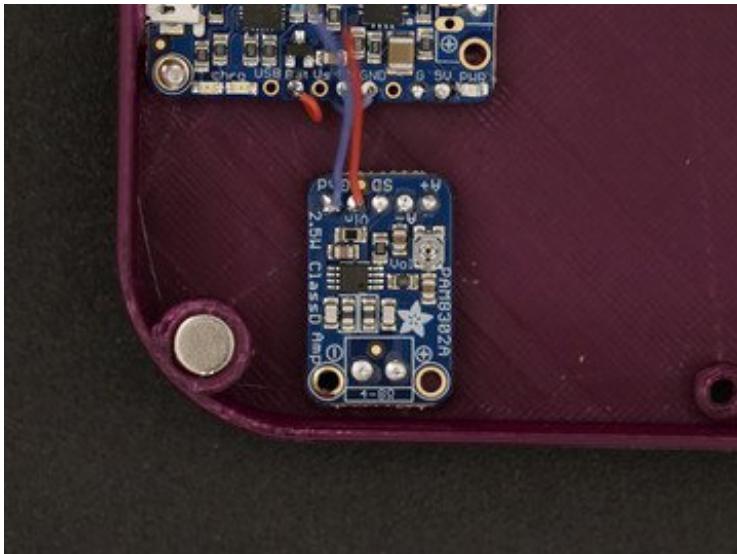
## Thread PAM8302

Just like the powerboost PCB, it's a good idea to thread the mounting hole with a machine screw. We only need one hole threaded since we're using a single screw to secure the PCB to the enclosure.



## Solder Wires to PAM8302

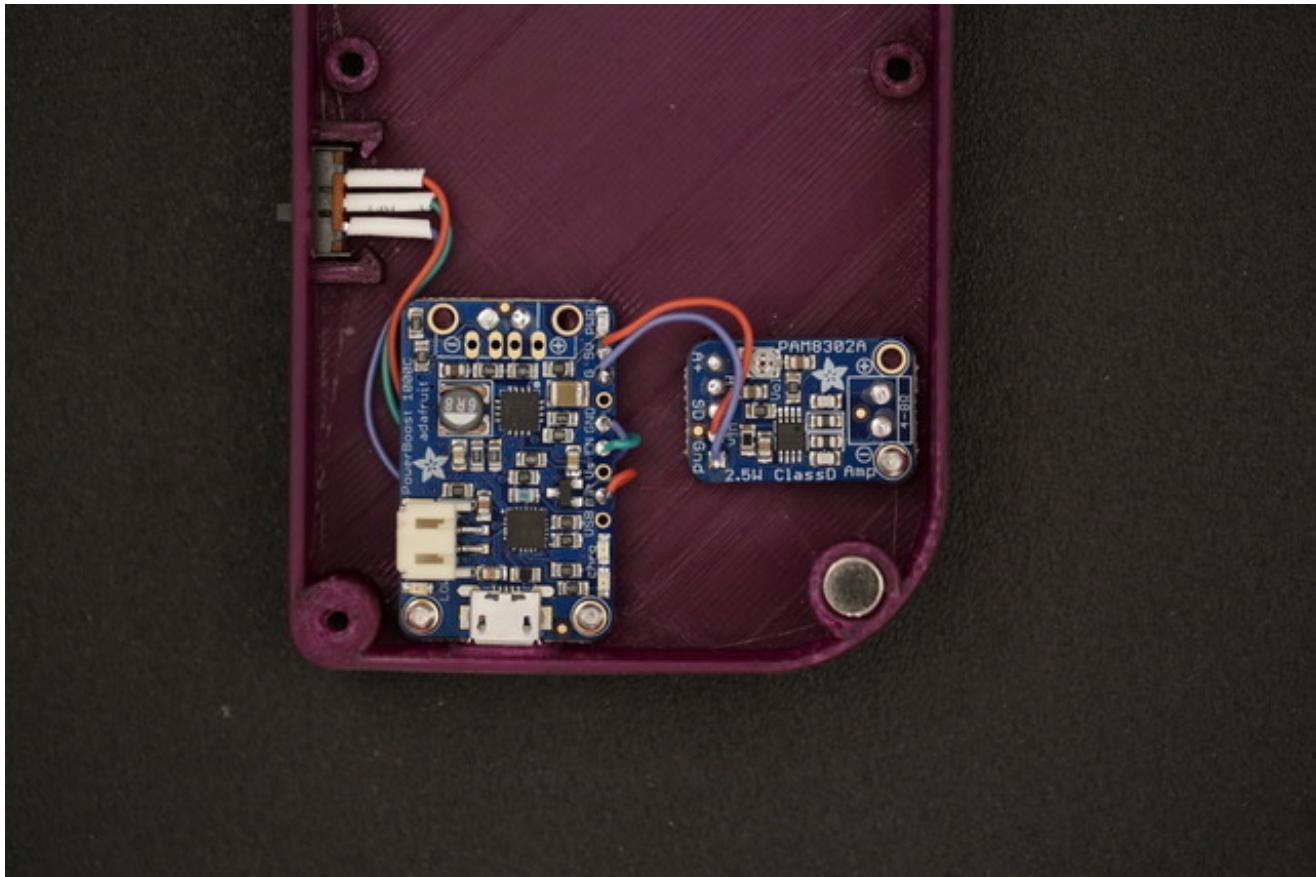
Secure the PCB and solder the red wire to **Vin** pin and the blue wire to **GND**.



## Mount PAM to enclosure

Fit the breakout over the spot near the powerboost 1000c and line up the holes. Hold the PCB in place while you fasten a single #4-40 3/8 flat phillips machine screw.

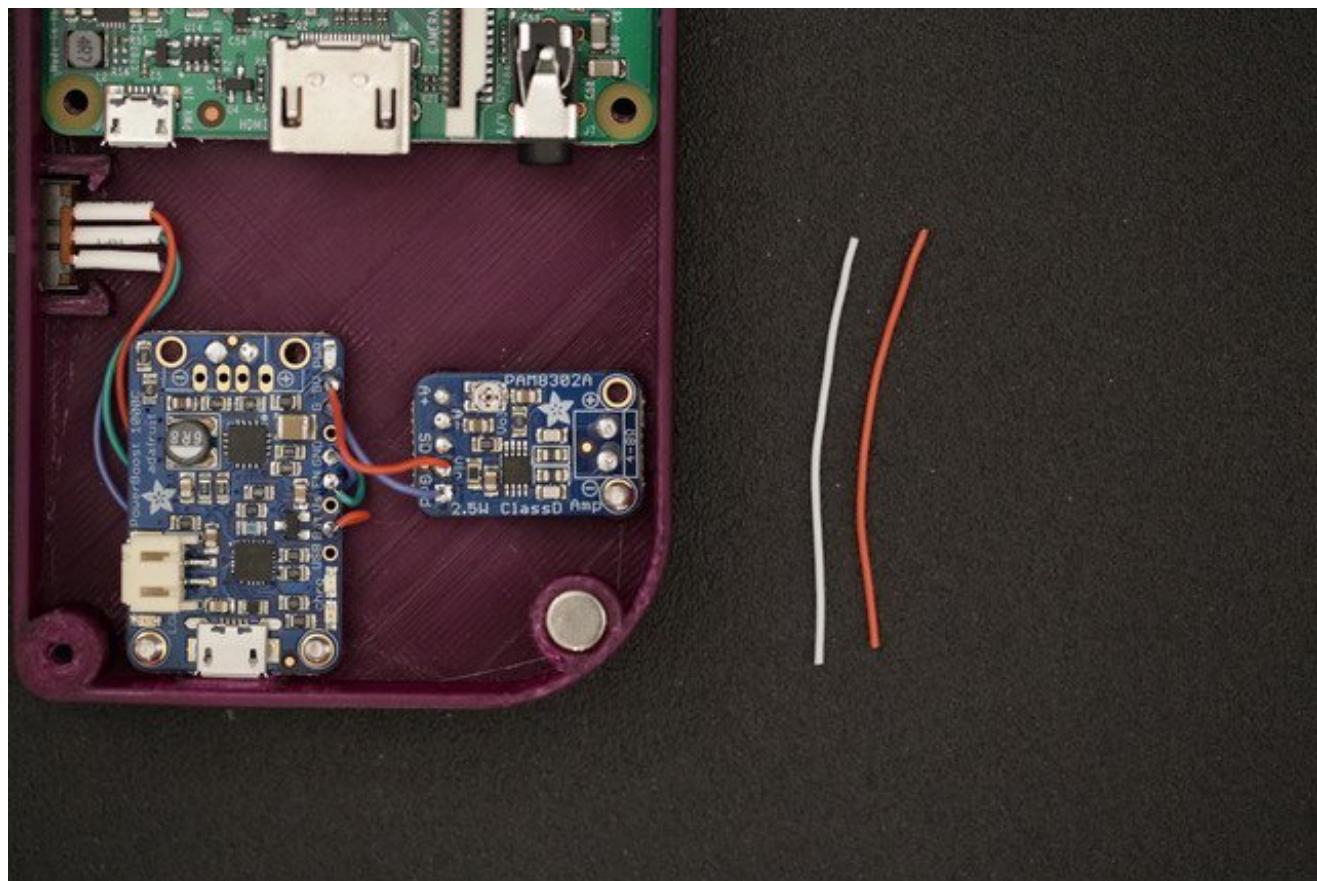




## Solder power to PAM8302

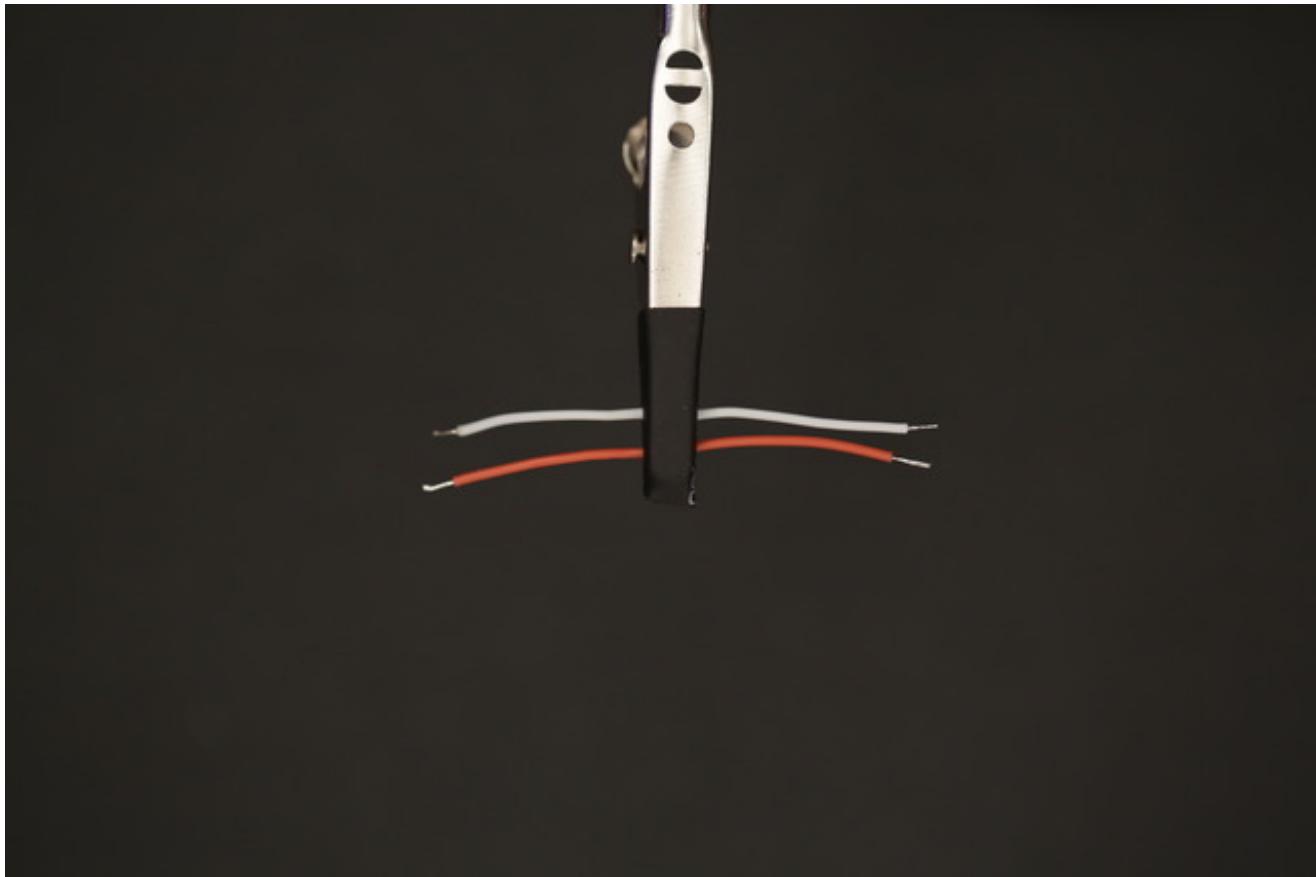
Solder up the red wire from **Vin** to **5V** on the powerboost 1000c. Then connect the blue wire from **Gnd** to **G** on the powerboost 1000c.

# Pi Audio



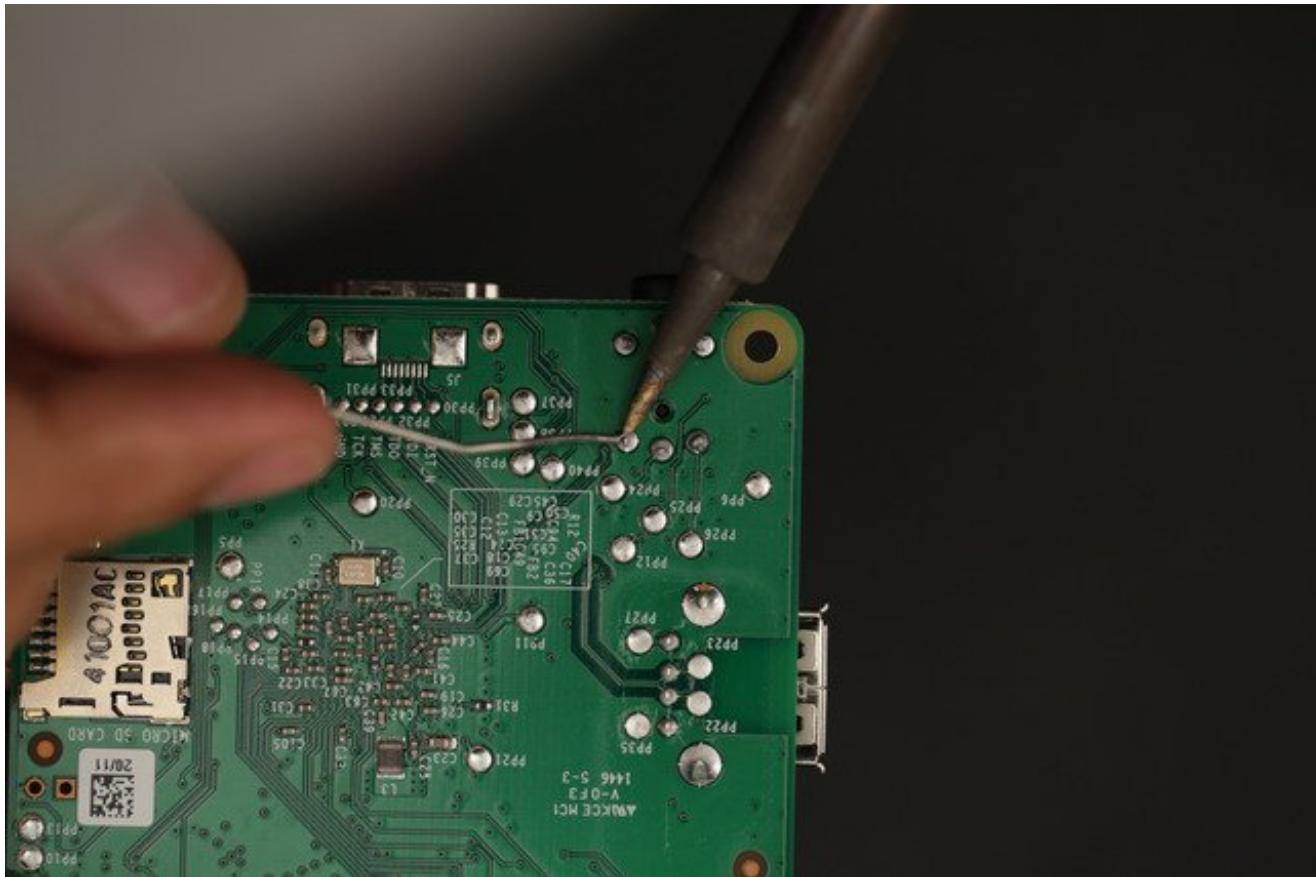
## Cut Pi Audio wires

Measure and cut up two wires for connecting to the audio jack on the Pi, to the audio input on the PAM8302.



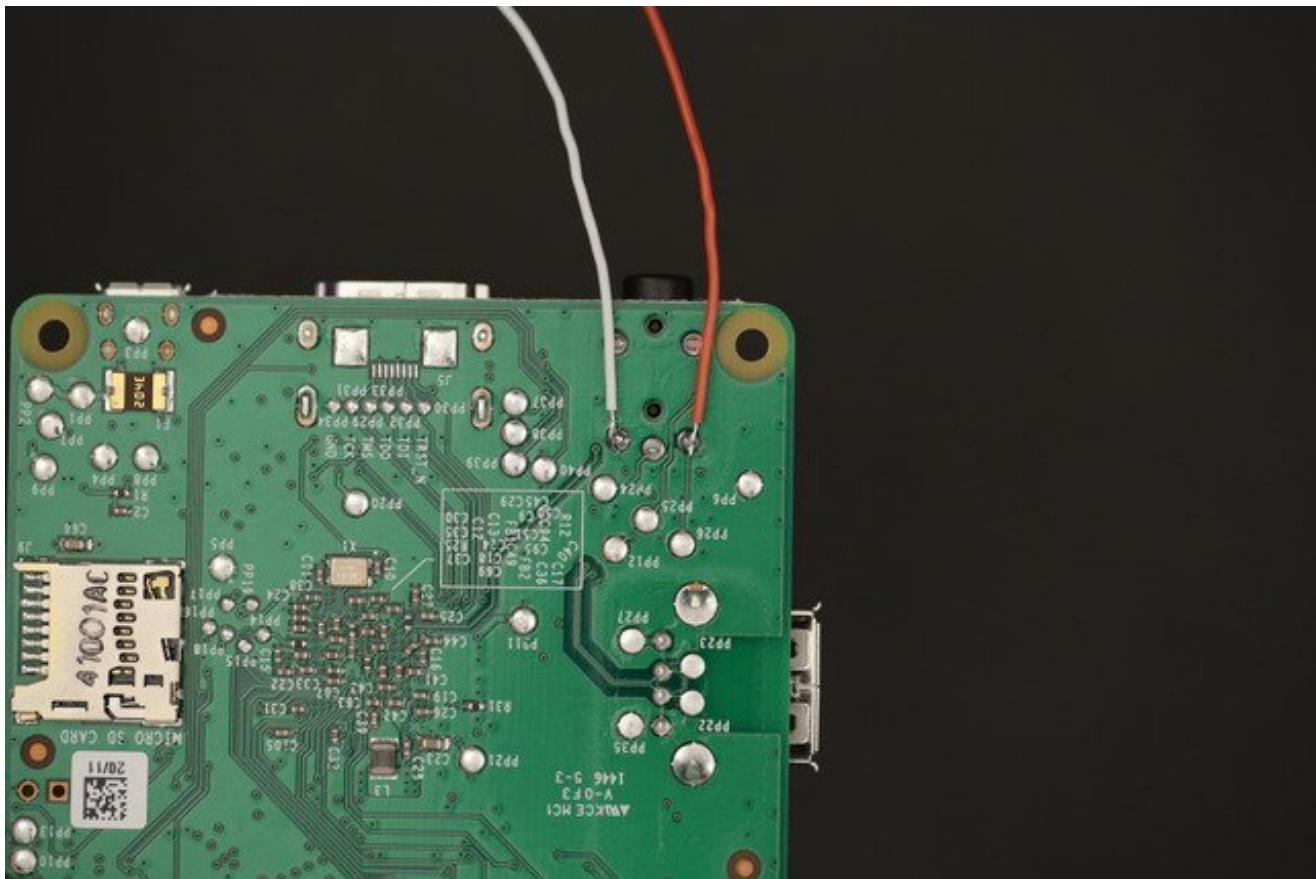
## Strip and Tin Wires

Strip off the ends of each wire and tin the tips. Quick tip, you can secure both wires to a helping third hand so you can solder them all at once.



## Tin Audio pins on Pi

Next up we need to prep the solder points on the Pi to make it easier to solder our audio wires. Start by securing the Pi to the panavise jr downside up. Locate the audio jack and find the two solder spots for the positive and ground. Add a small amount of solder to these points.



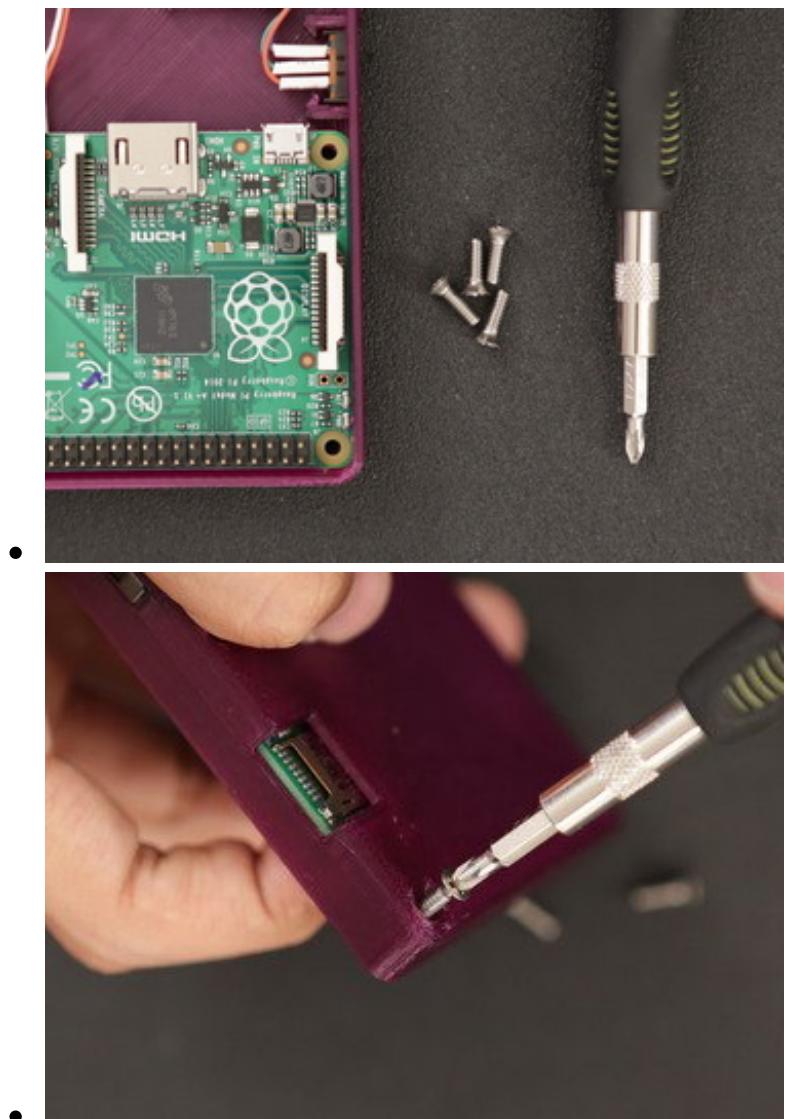
## Solder Audio Wires to Pi

Hit those points with the iron and solder the wires to positive and ground. The ground (white wire) can be indicated by seeing there's no traces connected to it. The positive (red wire) connection has visible traces.



## Solder Audio wires to PAM8302

Place the Pi into the base with the audio jack facing amp. Wire the positive (red wire) to A+ pin and the ground (white wire) to the A- pin on the PAM8302



## Mount Pi

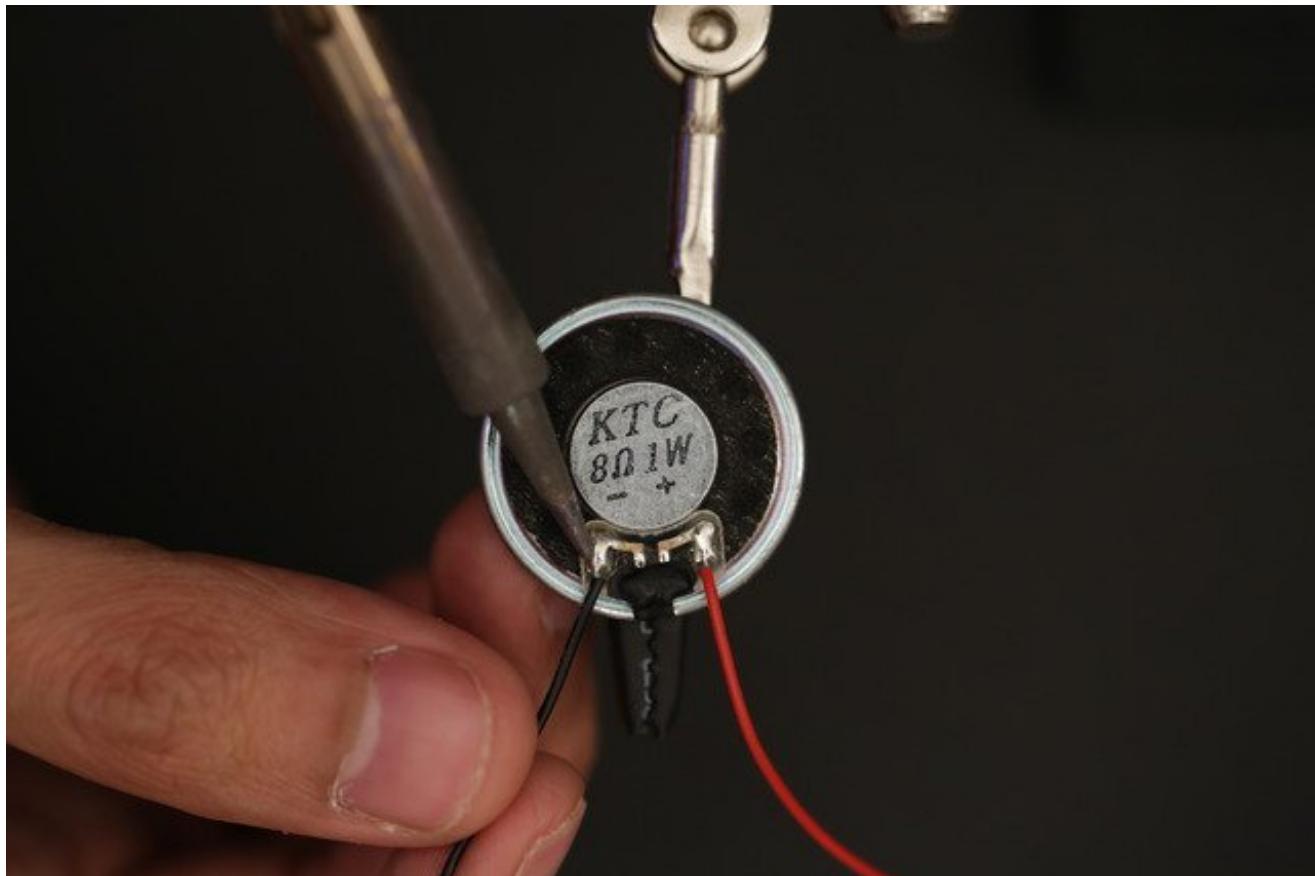
Hold the Pi in place while you insert and fasten four #4-40 3/8 flat phillips machine screws into the mounting holes.



## Mounted Pi

The screws should be fastened all the way so they're flush with the surface of the case.

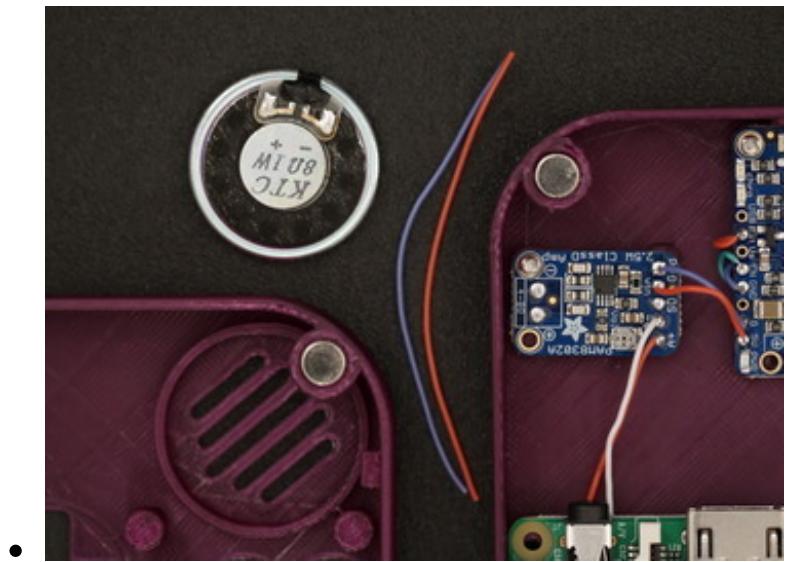
# Speaker



## Prep Metal Speaker

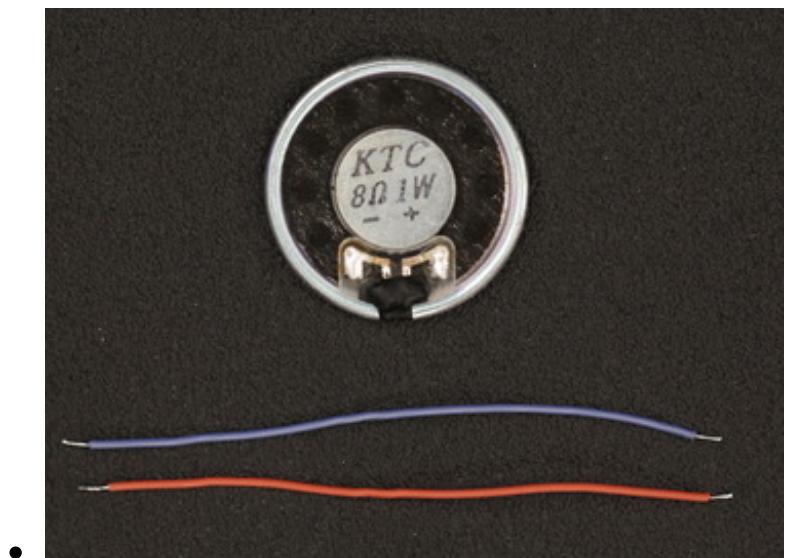
Secure the speaker on the helping third hands (it has a magnet so just let it stick to one of the grabbers). Hit the solder joints and remove the wires.

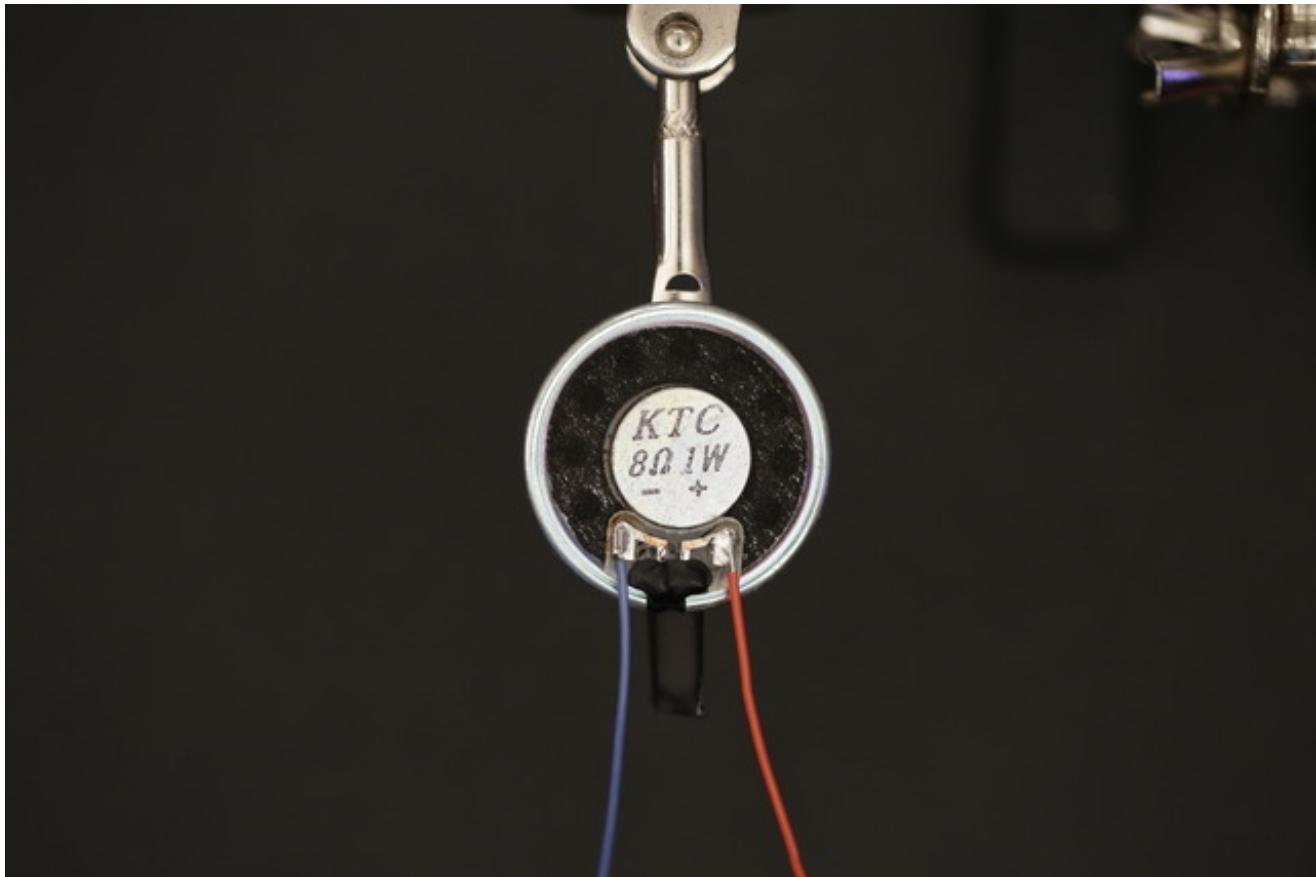
Be careful not to touch the magnet with the soldering iron tip!



## Prep New Wires for Speaker

Sit the two case parts close to each other to guage wire length and cut two pieces. Strip the tips of each wire and add solder to tin the tips.





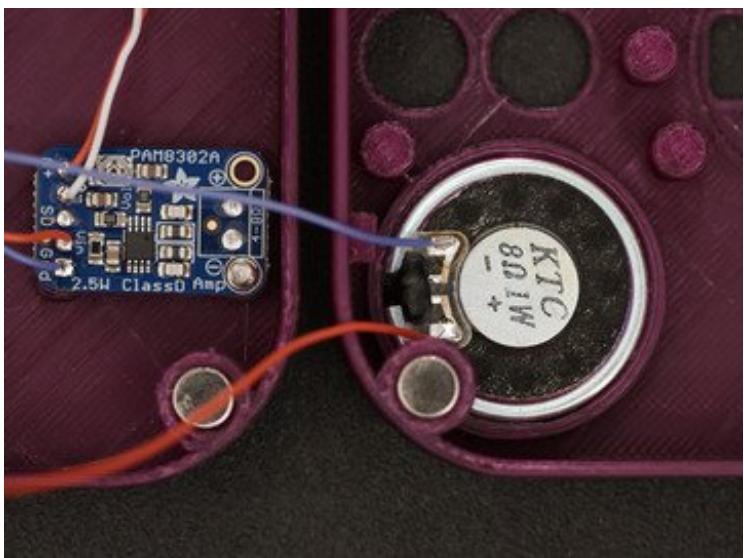
## Solder Wires to Speaker

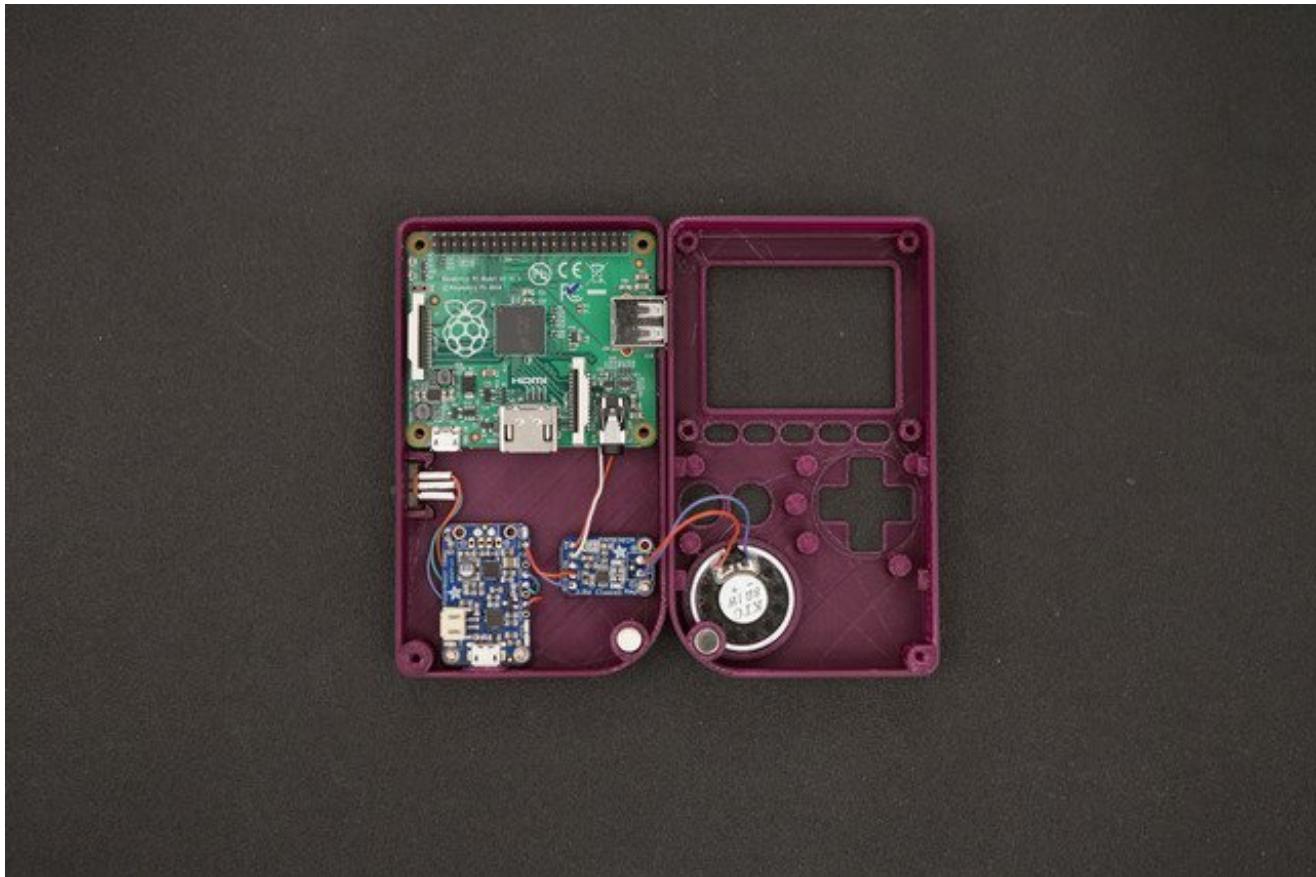
Heat up the solder points on the speaker and connect the wires. The positive and negative symbols are indicated on the speaker. If it's not, follow the photo to reference the polarity (left is negative, right is positive.)



## Install Speaker

Position the speaker into the bottom corner of the enclosure. There's a circular indent with slits near the rounded corner of the case. That's where the speaker needs to go. The tolerance should be sized to fit the speaker tightly.





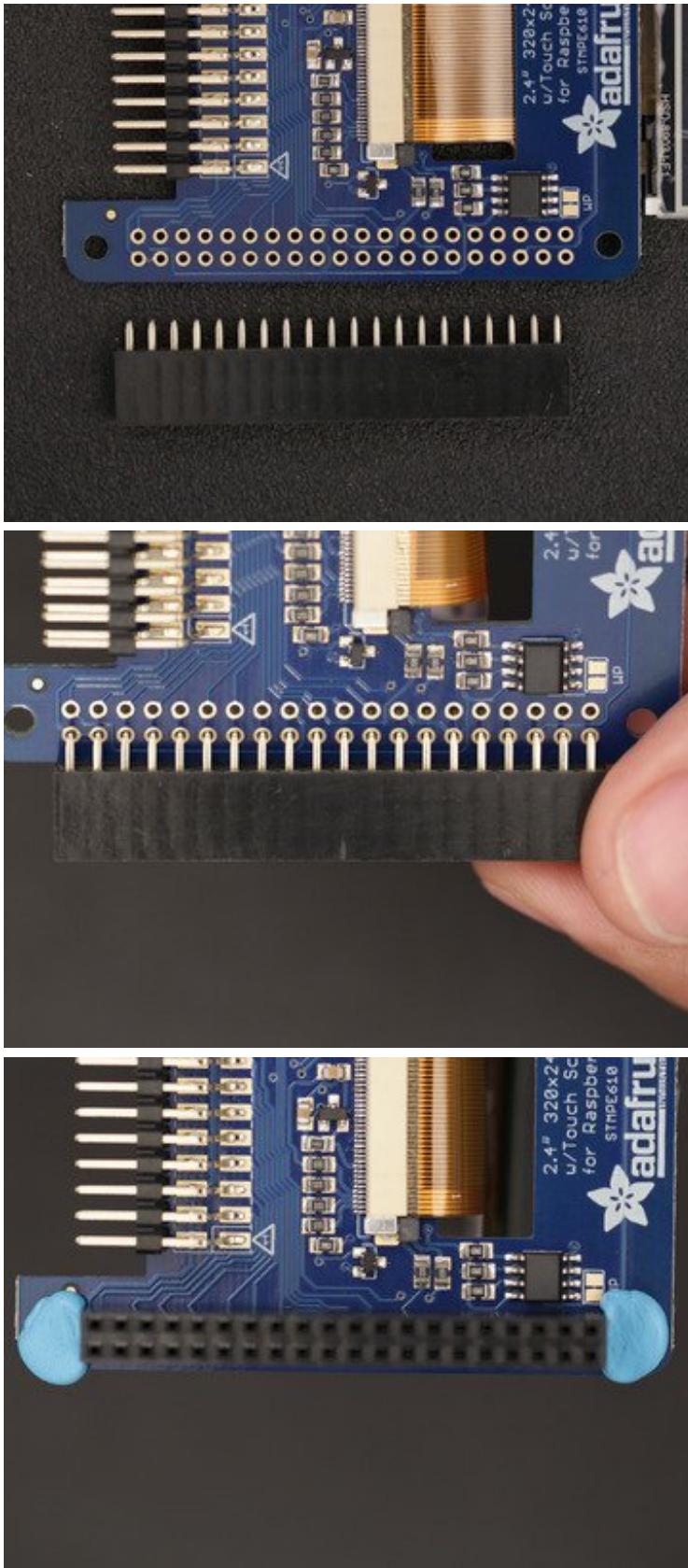
## Connect Speaker wires to PAM8302

Solder the wires from the speaker to the audio out pins on the PAM8302.

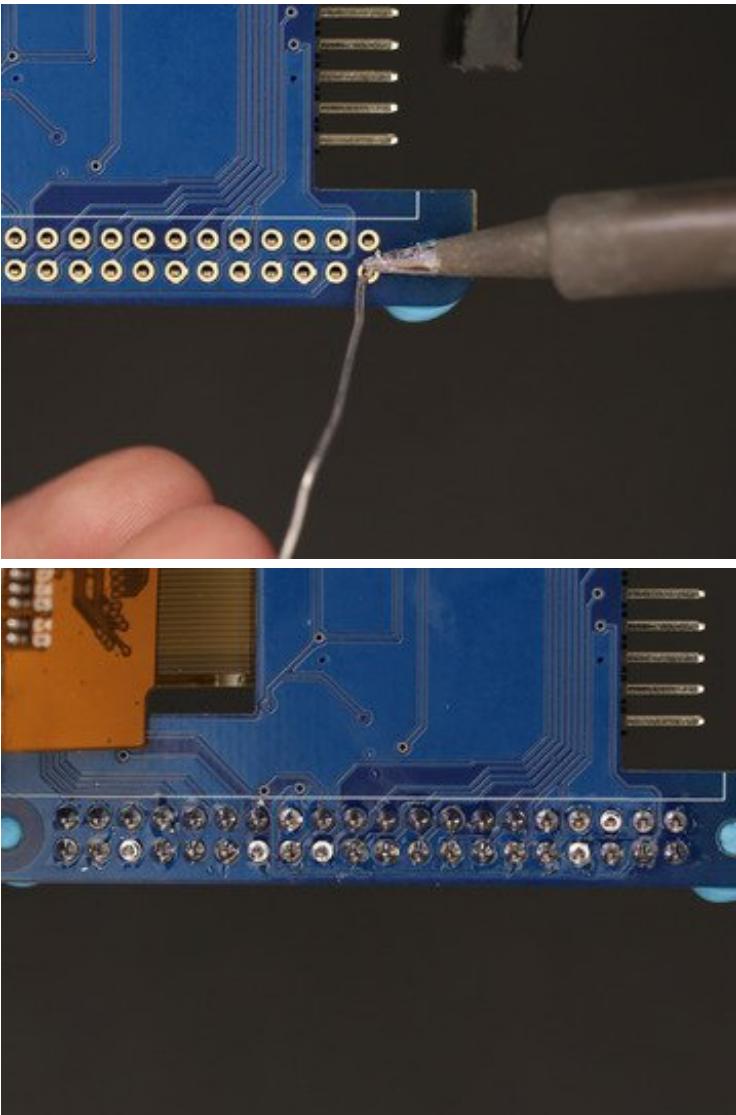


## 2.4 PiTFT

**Install Header**

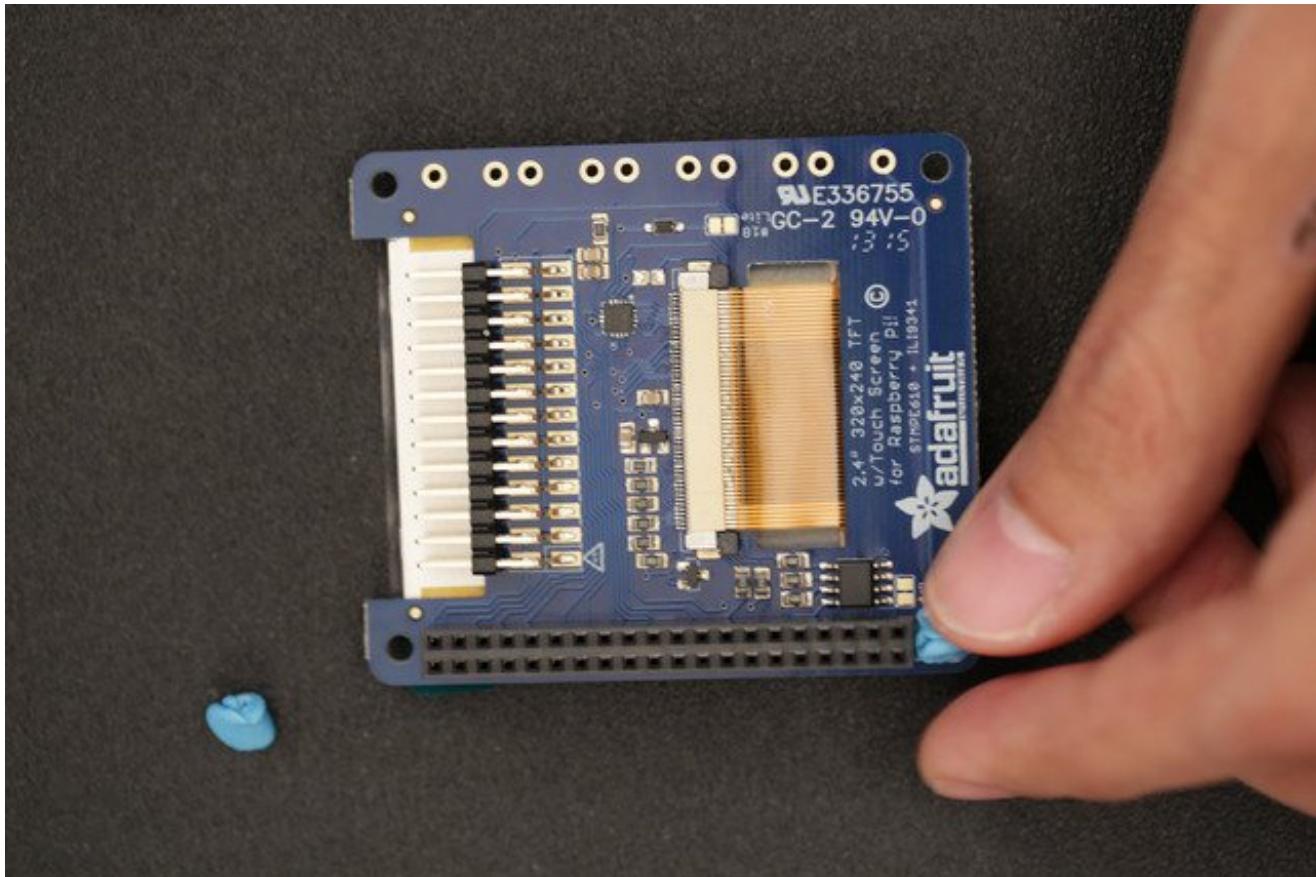


Insert the header that came with the PiTFT into the PCB. Use a piece of fun tac to hold the header in place while you solder.



## Secure Header

Solder the pins on the GPIO to secure the header to the PCB. The solder joints should look like little hersey's kisses.



## Remove Tac

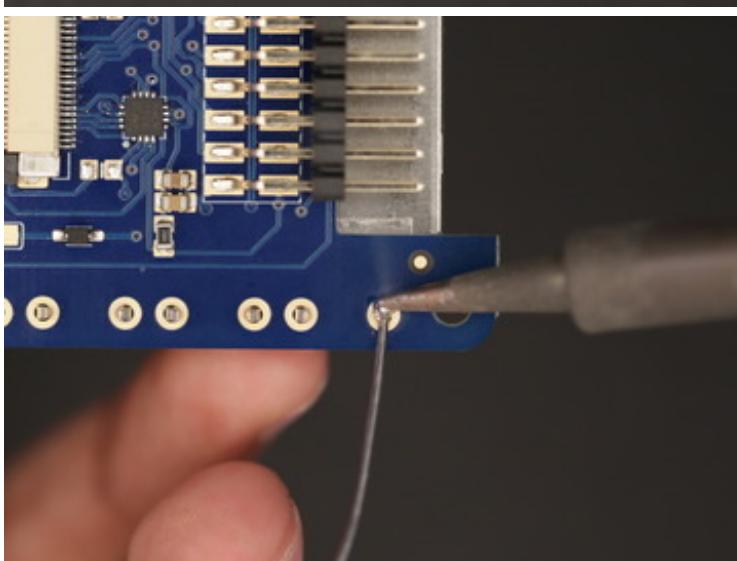
Remove the tac from the GPIO header. Ball it up and save it for the next project, it's served its purpose!

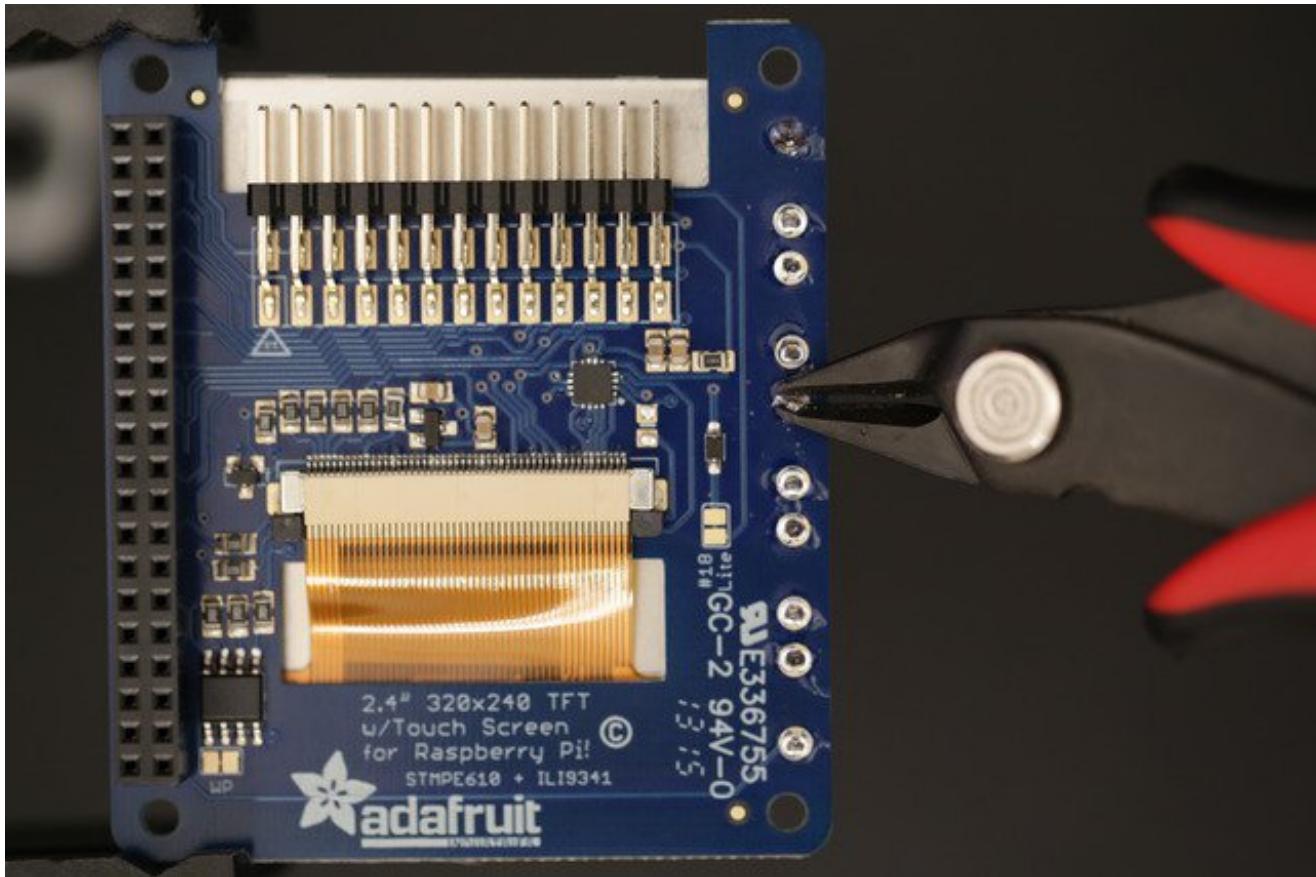




## Add Buttons

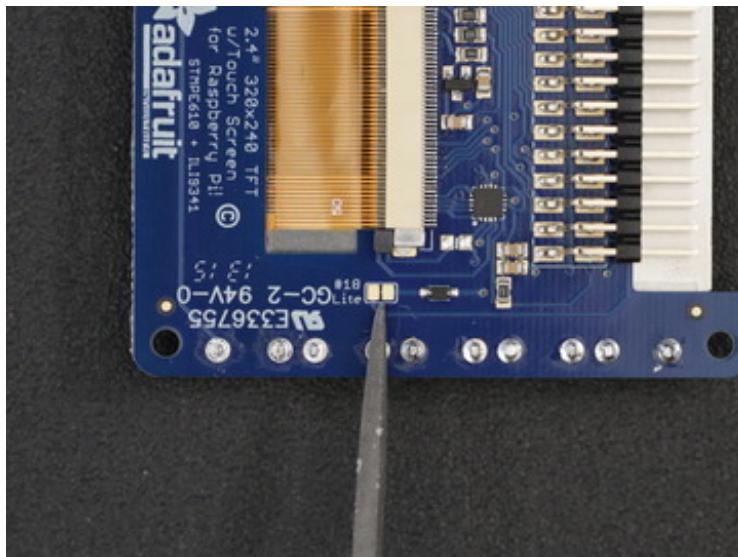
Insert five tactile switch buttons into the designated spots on the PiTFT PCB. They snap fit and will hold in place. Secure the PCB to a panavise jr and solder the buttons to secure them.





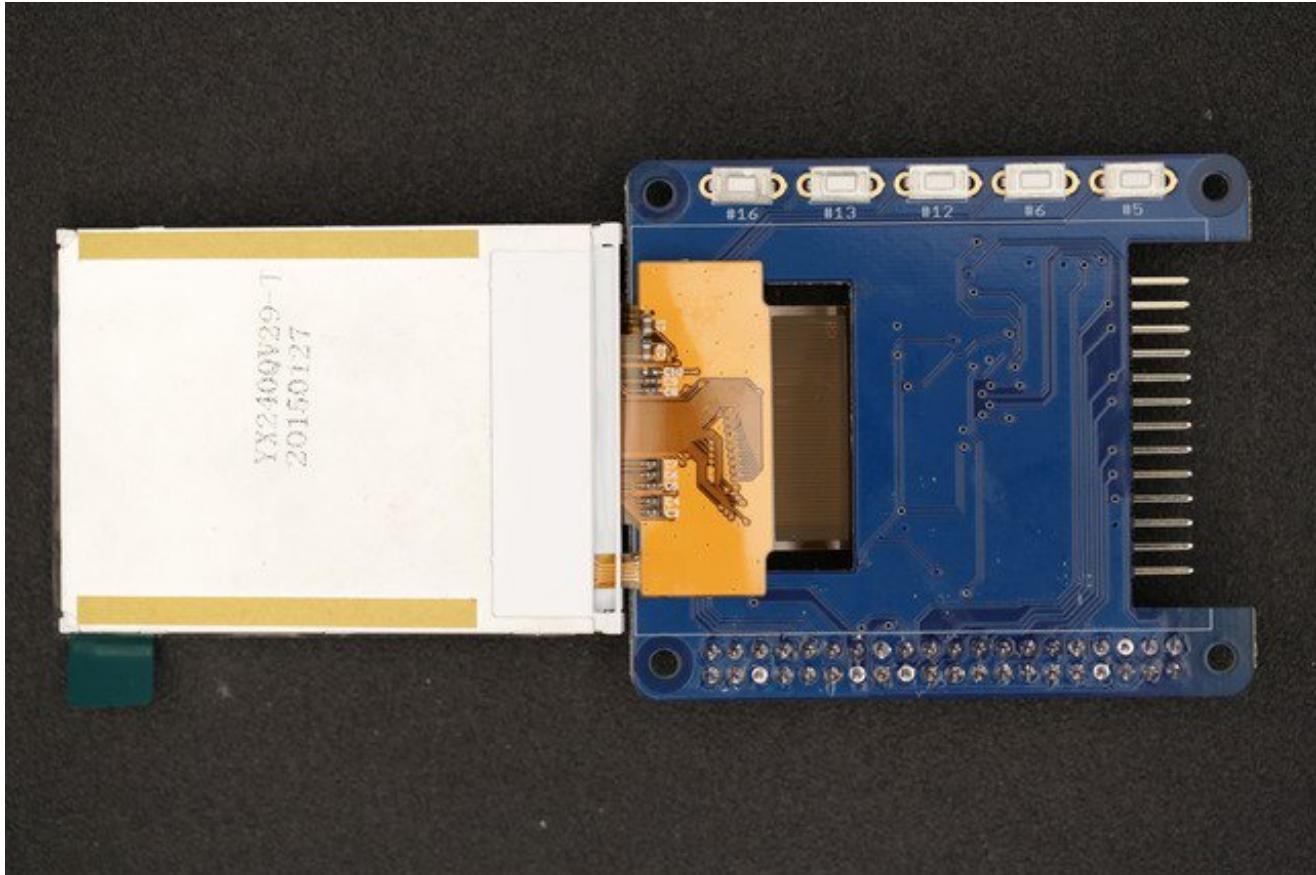
## Trim buttons

Snip the excess leads from the buttons the PiTFT PCB. These bits are sharp and could scratch the components.



## Disable GPIO 18 Lite

By default, the 2.4' PiTFT uses GPIO 18 to turn on/off the backlight on the display. In this build we're using GPIO 18 as a input button, so we need to cut the trace located on the back of the PCB, labeled **#18 Lite**. Use a hobby knife to cut the trace.



## Prep Display PCB

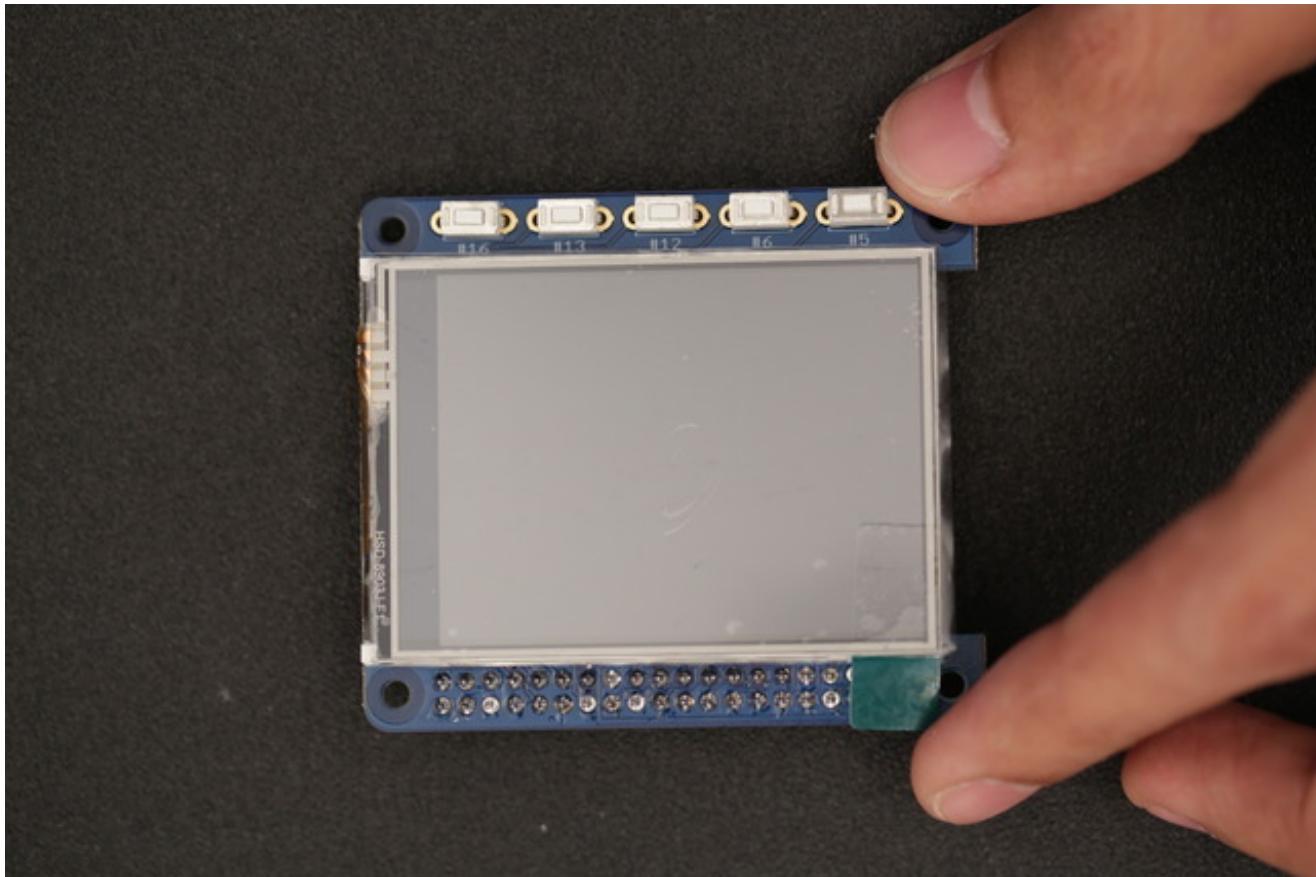
With the GPIO header and buttons soldered, go ahead and flip over the display so the two parts are side by side.



## Peel

Use your nail or hobby knife to peel the orange colored sticker backing. Remove and peel both pieces.



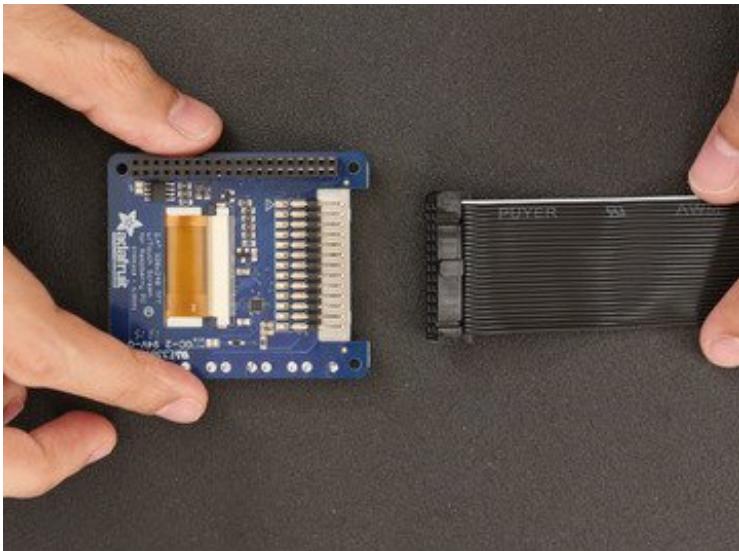


## Stick Display

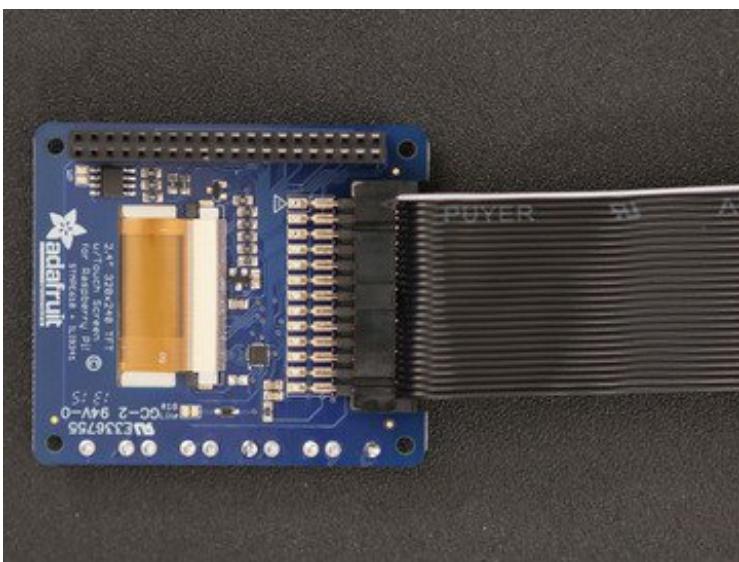
Flip over the display and line it up with the screen outline on the PCB. Press down on the screen to make the adhesive stick to the PCB.

You might also want to use some double-sided tape if the side tape pieces are not strong enough

# Pi Cable

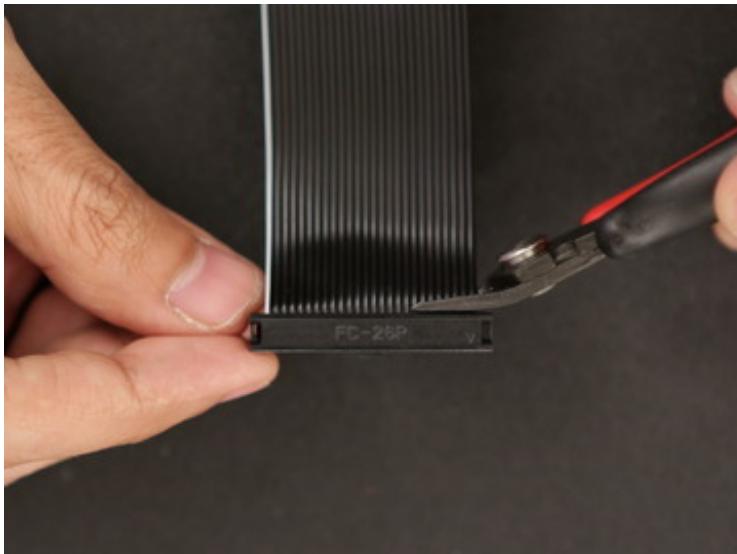


- 



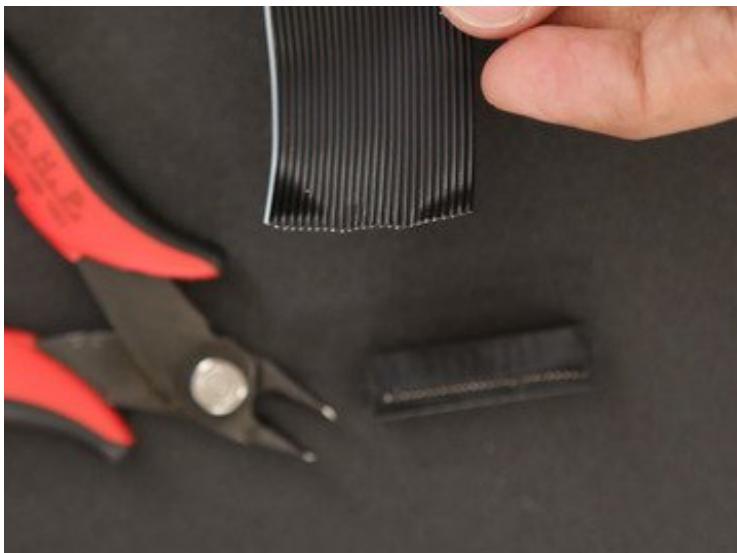
## Connect Pi Cable

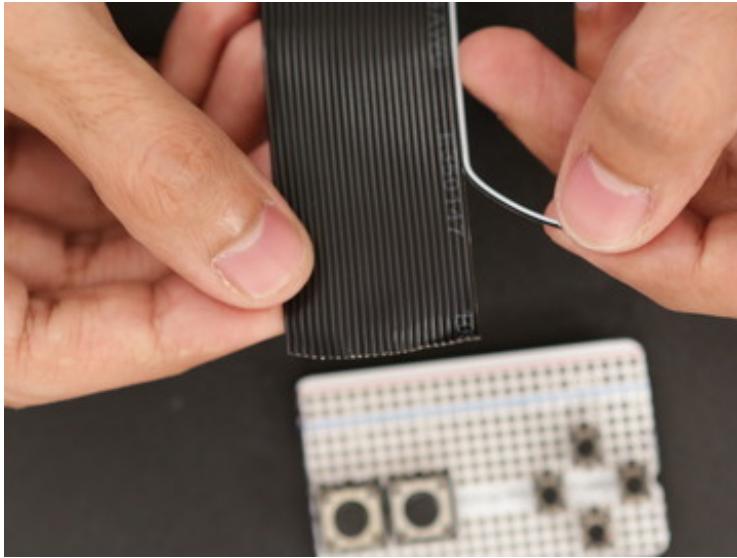
Line up the white wire on the ribbon cable with the white arrow on the PiTFT PCB. Press firmly on the connector to plug it into the pins.



## Remove Pi Cable Connector

Grab your set of flush cutters and snip off the connector the other end.





## Peel Cable

Peel apart each strand of wire from the cable like you would with stringy cheese. Peel it just enough so they're still some ribbon near the fold.

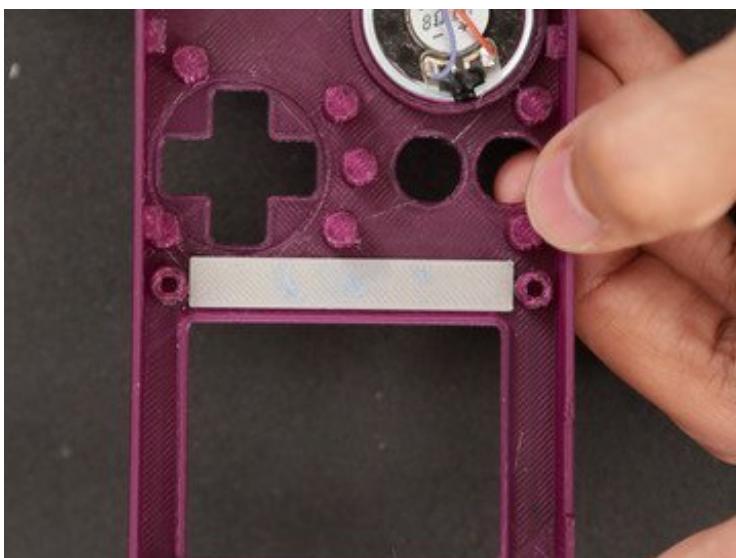


# Mount Display

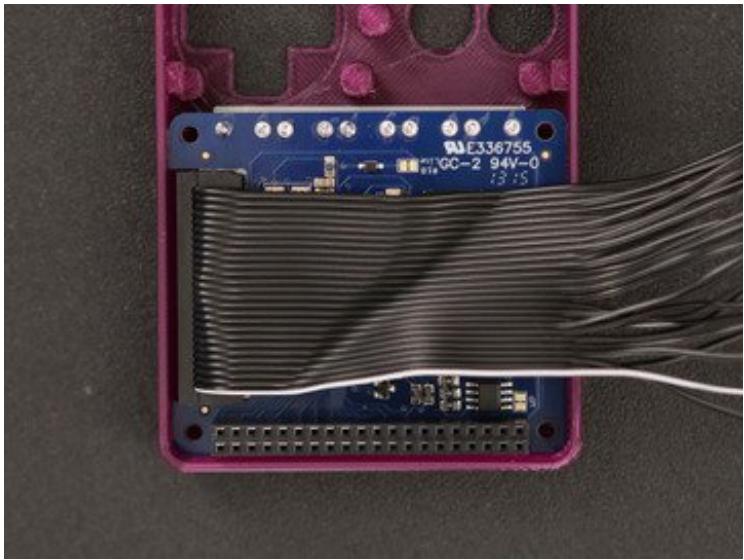


## Install display buttons

- 

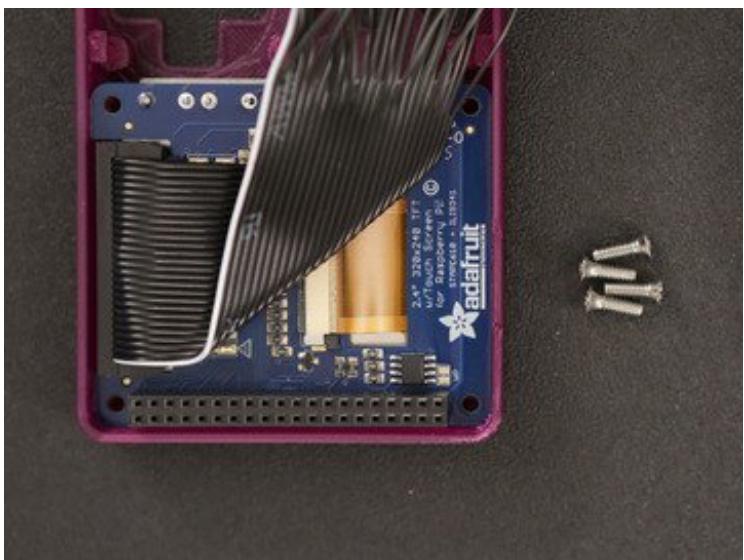


Now is a good time to insert that strip of buttons printed in Ninjaflex. The buttons should fit into the cuts on the enclosure.



## Install PiTFT

Fit the PiTFT into the enclosure and line up the buttons and mounting holes. Grab four #4-40 3/8 flat phillips machine screws.

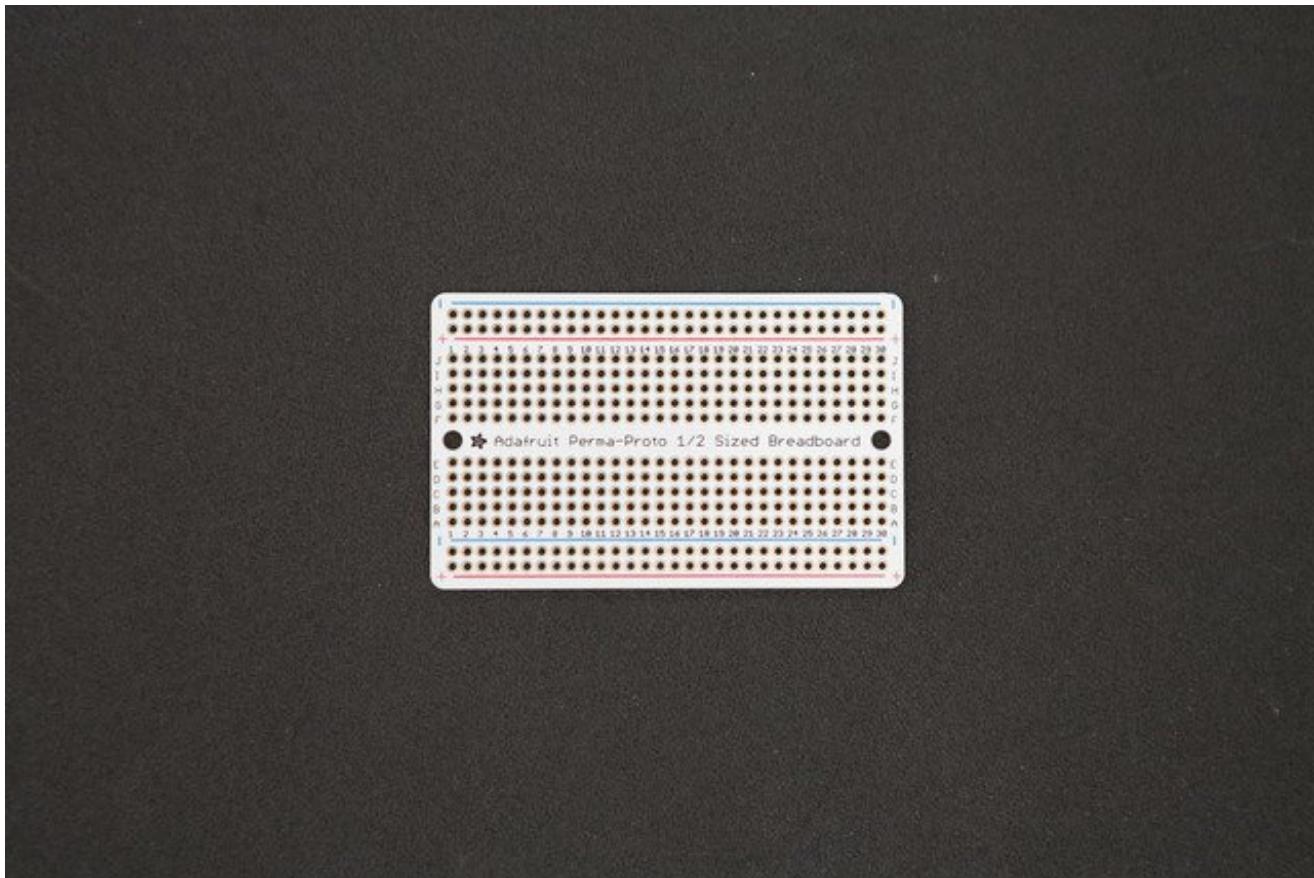




## Mount PiTFT

Hold the PiTFT in place while you insert and secure four machine screws into the PCB. These should fasten all the way through the PCB and the standoffs.

# Perma-Proto

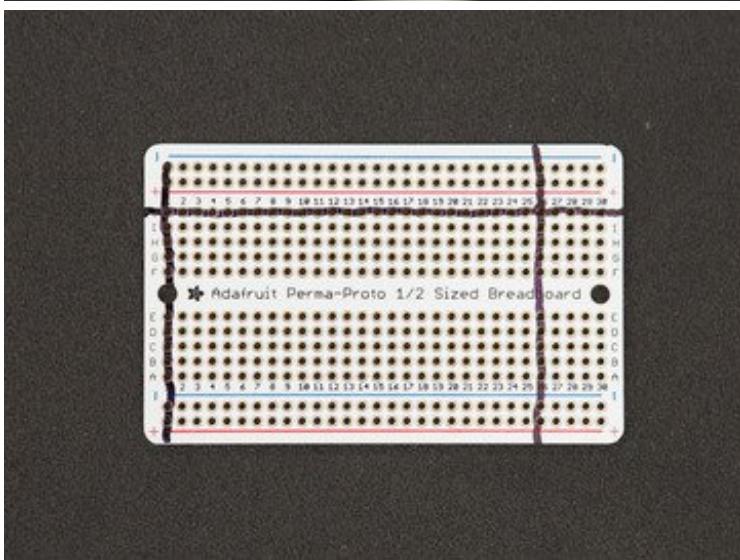
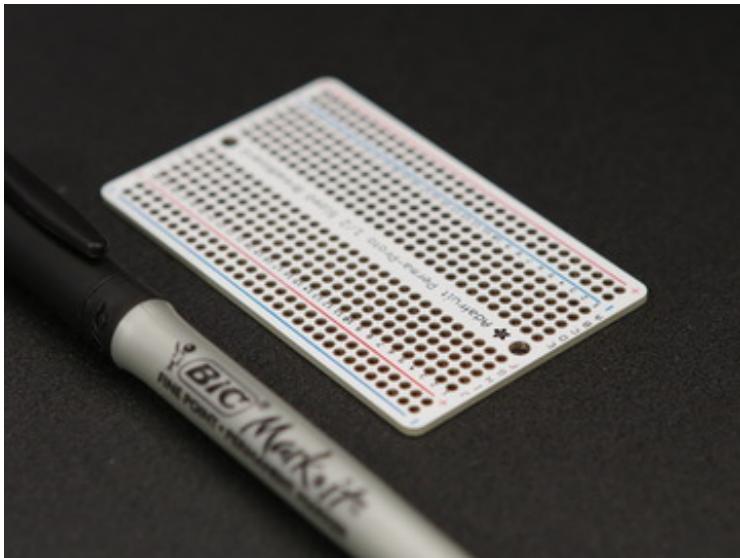


## Using Half-size Perma-Proto

The perma-proto PCB is a lot like a breadboard. The rows are all linked together, while the columns are not. There's two power rails on the top and very bottom that include ground and voltage (indicated by blue and red lines).

The columns are indicated by numbers (1-30) while the rows are indicated by letters (A-J).

The half-size PCB has two separated centers in the center (A-E and F-J). Getting familiar with these labels will help you install the buttons and manage soldering wires.



## Mark Perma-Proto

We need cut the PCB down so it can fit inside the enclosure. Before we do that, let's mark up on the PCB and draw guidelines.

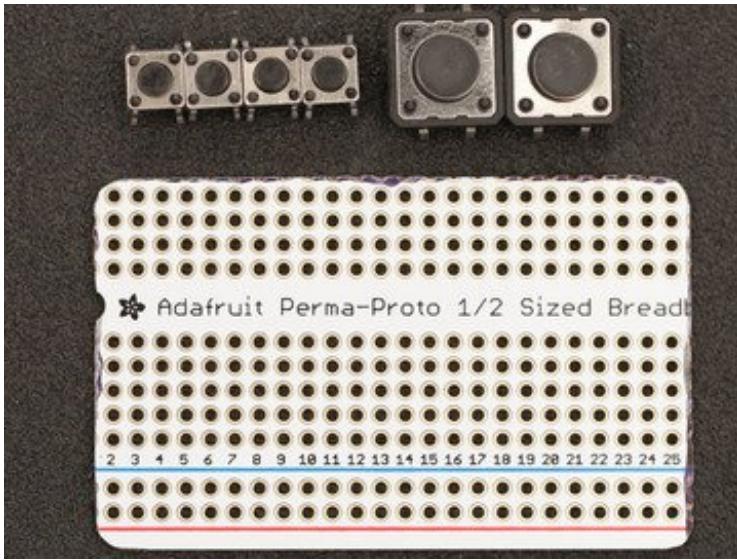
- Mark the first row down (including rail)
- Mark the first column across (including rail)
- Mark row 26 down (including rail)

## Cut Perma-Proto

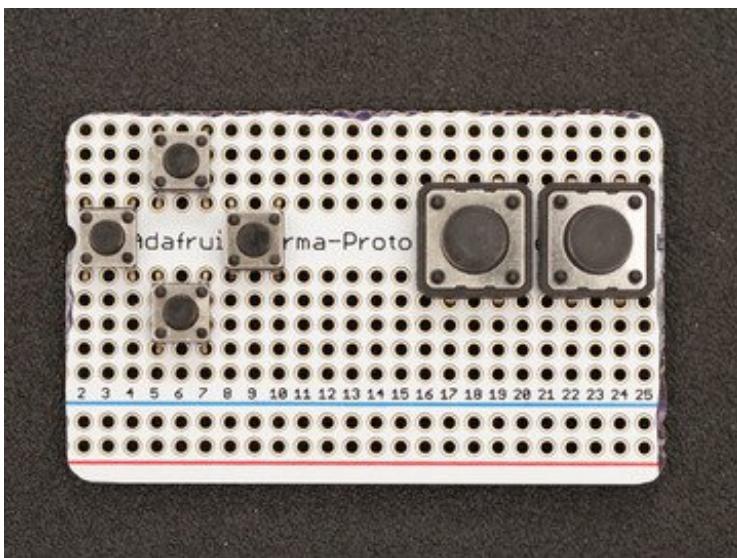
I used a Dremel with a thin cut attachment to saw the excess pieces off the PCB. I recommend taking safety precautions when doing this.

- Do it in a well ventilated area (outside).
- Use safety glasses and a breathing mask.
- Secure the PCB to panavise jr.
- Always cut away from you.
- Clean dust from work area.

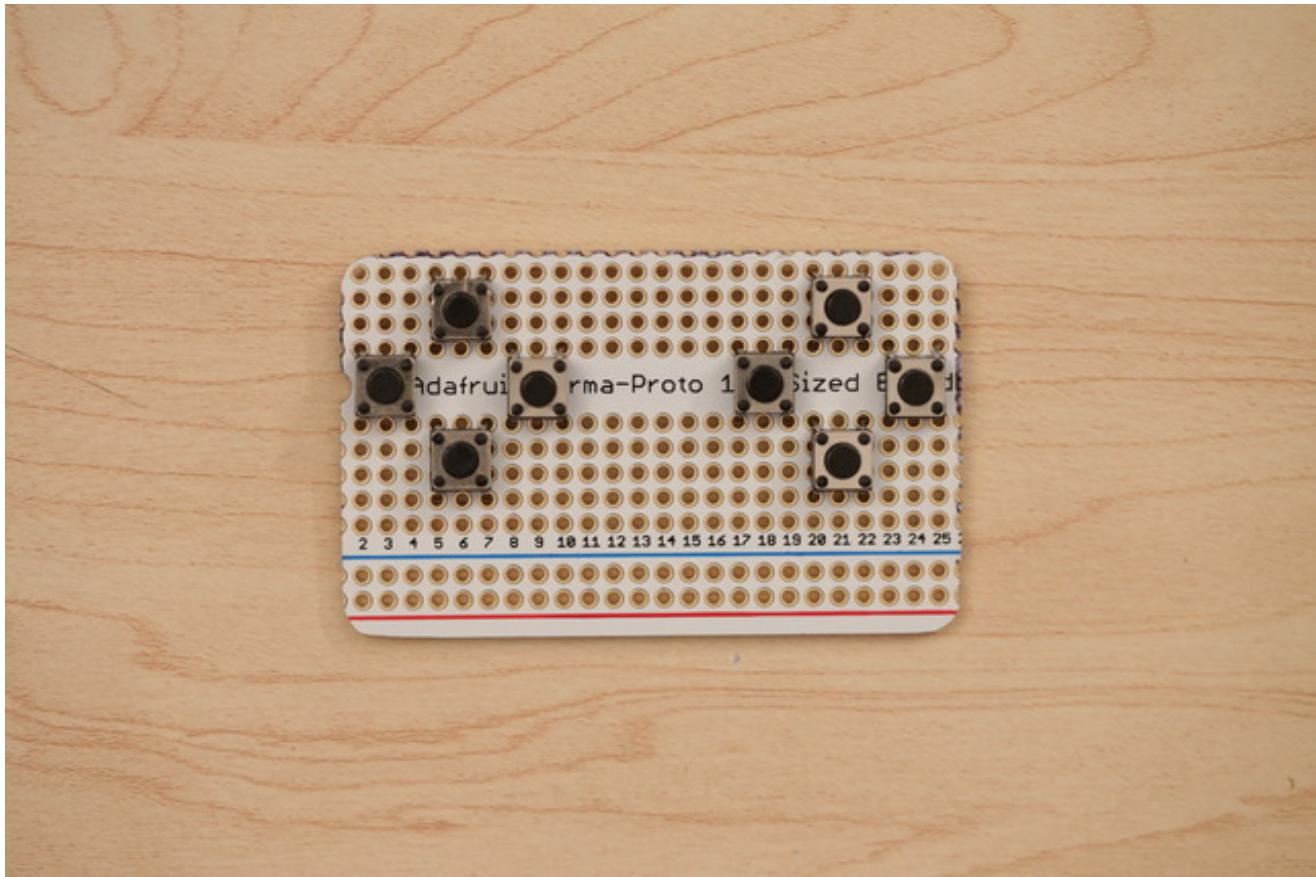
If you're not satisfied with the sharp edge, use can use a grinding bit to round off the corners.



## Install buttons (Two Action Buttons)

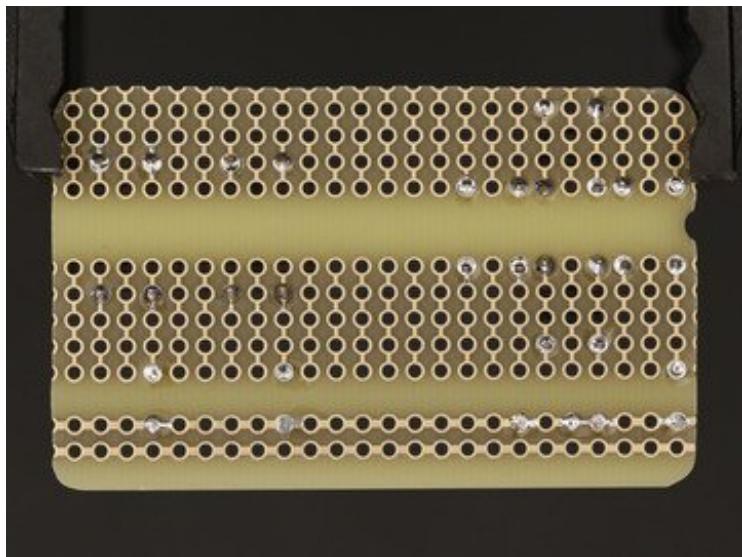


Insert the four 6mm tactile buttons into the perma-proto and form the D-pad. Two 12mm square will be the "A" and "B" buttons.



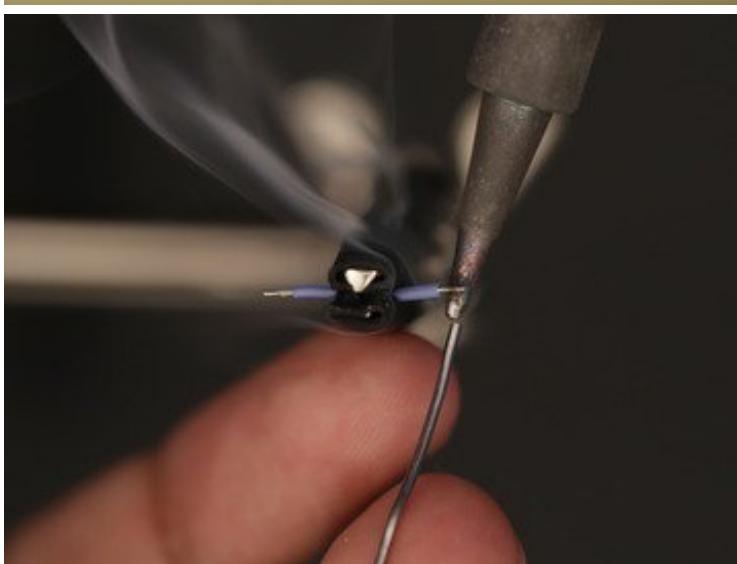
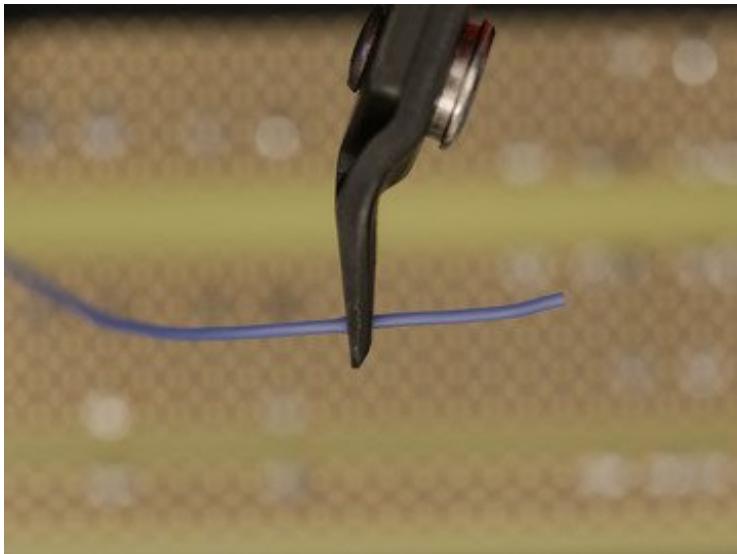
## Install buttons (Four action buttons)

If you're making the four-button layout, insert eight 6mm tactile buttons to the perma-proto PCB. Reference the photo above to get the correct spots for each button.



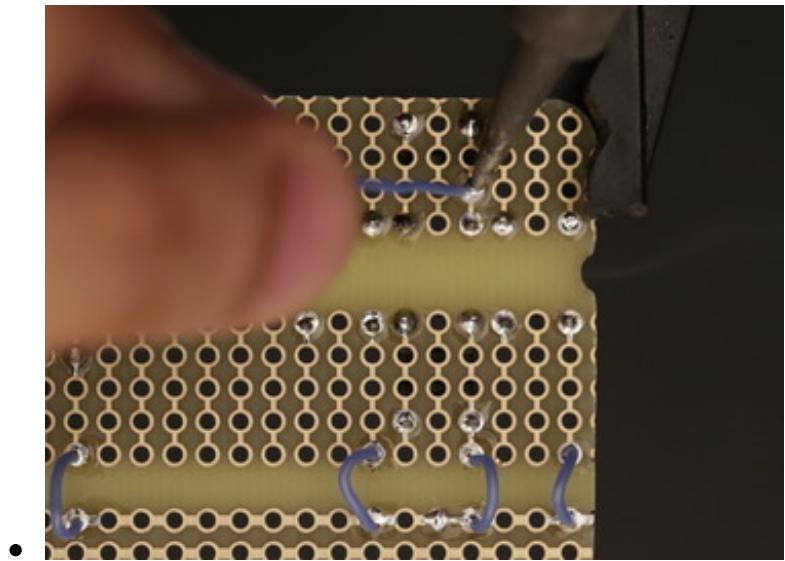
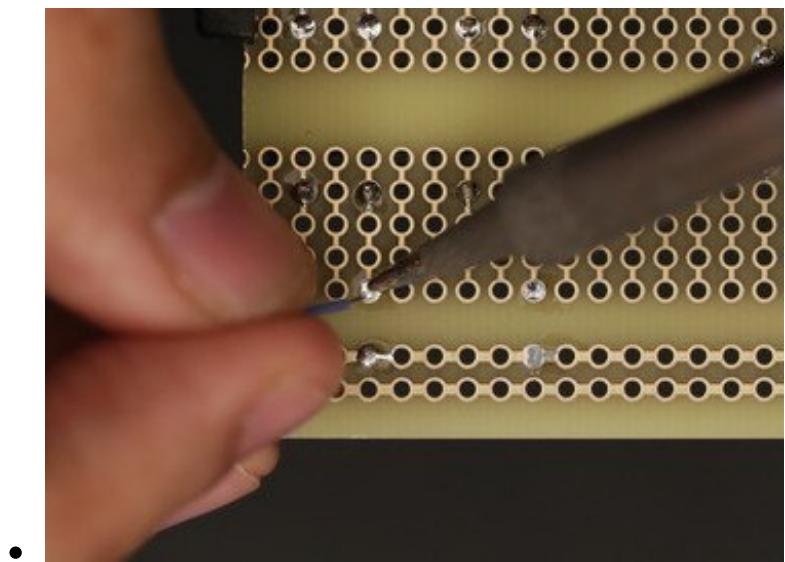
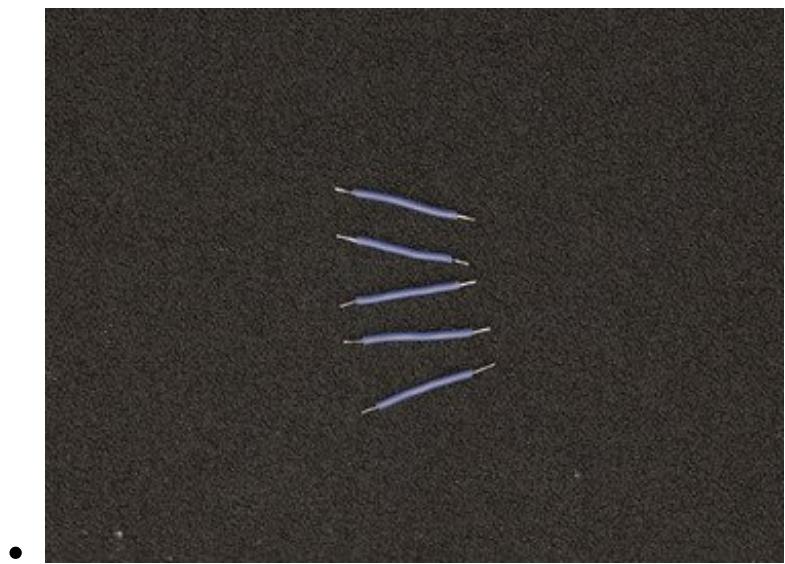
### Solder buttons to PCB

The buttons should hold in place once inserted into the PCB (if they don't, you can bend the legs to hold them in place). Flip it over so the bottom is face up and secure the PCB to the Panavise jr.



## Prep Ground Wires

We need to create short wires to connect the buttons to the ground rail. These wires should be pretty short about 10mm in length. You'll need to create 5 of them. Strip the tips off each wire and tin them.

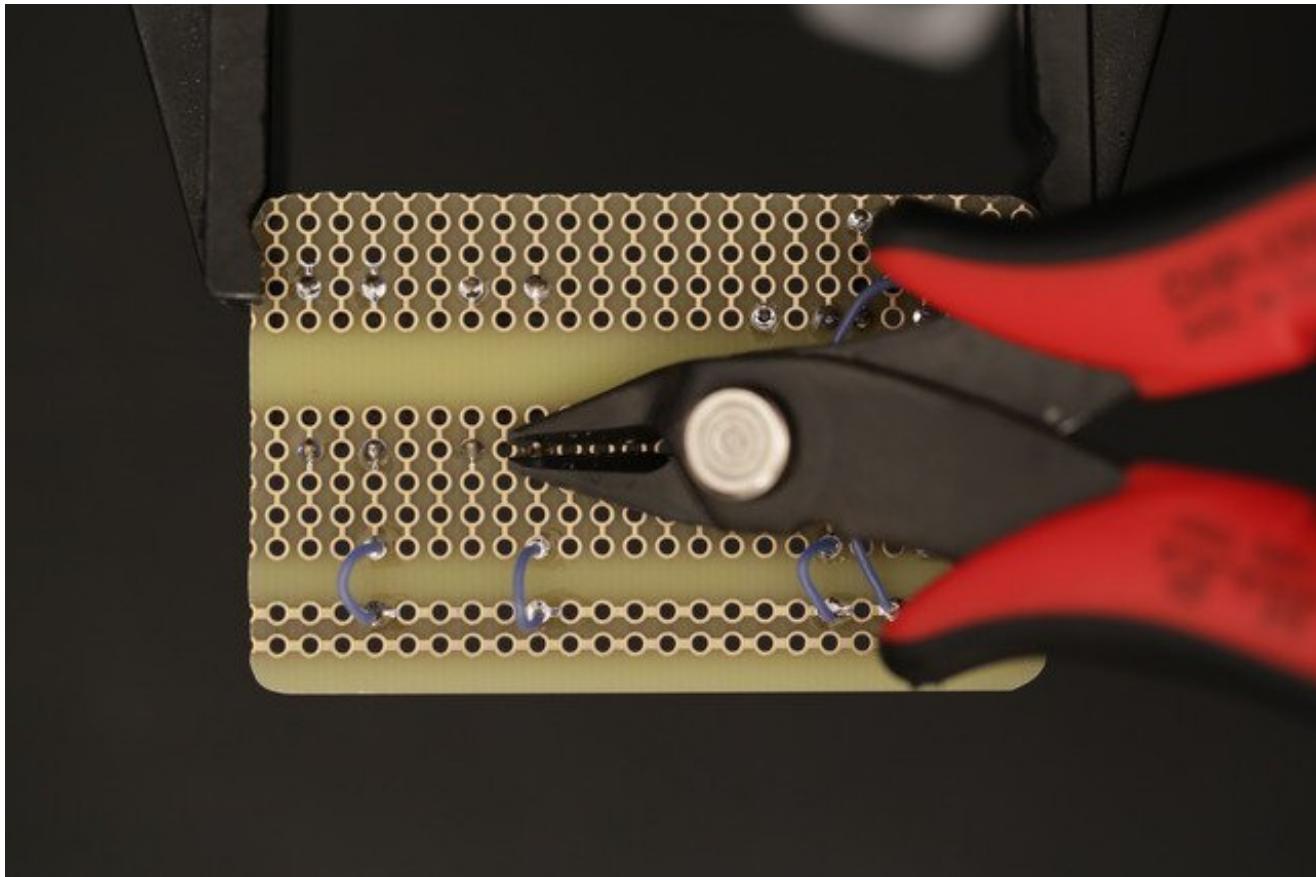


## Solder Ground Wires

For each button you'll need to tin the closest pin to the power rail and the ground pin. Since the rows are all linked together, we can just solder the pins that are closed to each other. The only button that doesn't is that up button (which we'll make a longer wire).

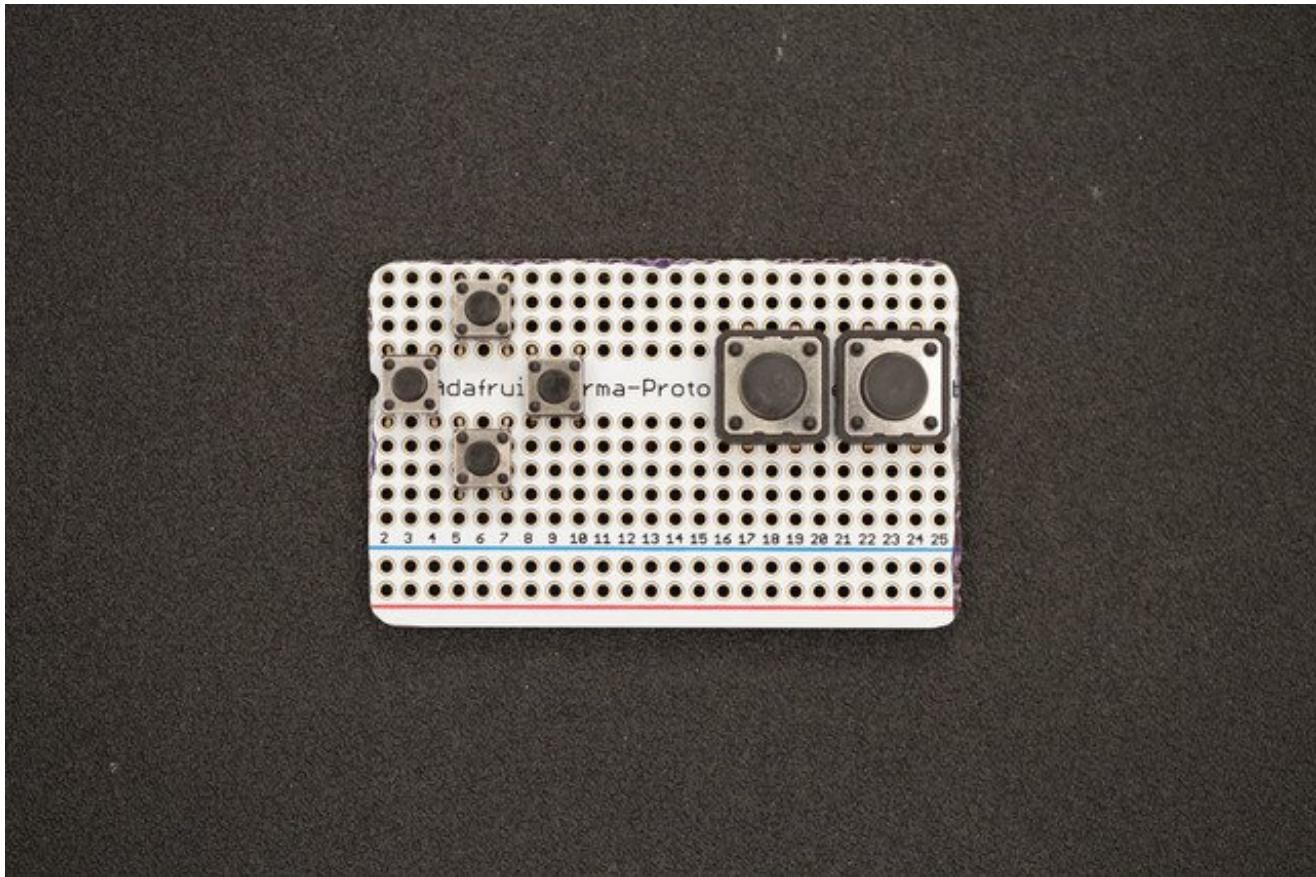
## Up Button Ground Wire

The up button for the d-pad is the only button that doesn't use a short wire because its on the top section. We'll need to create a longer wire to connect it to the ground rail.



## Trim legs

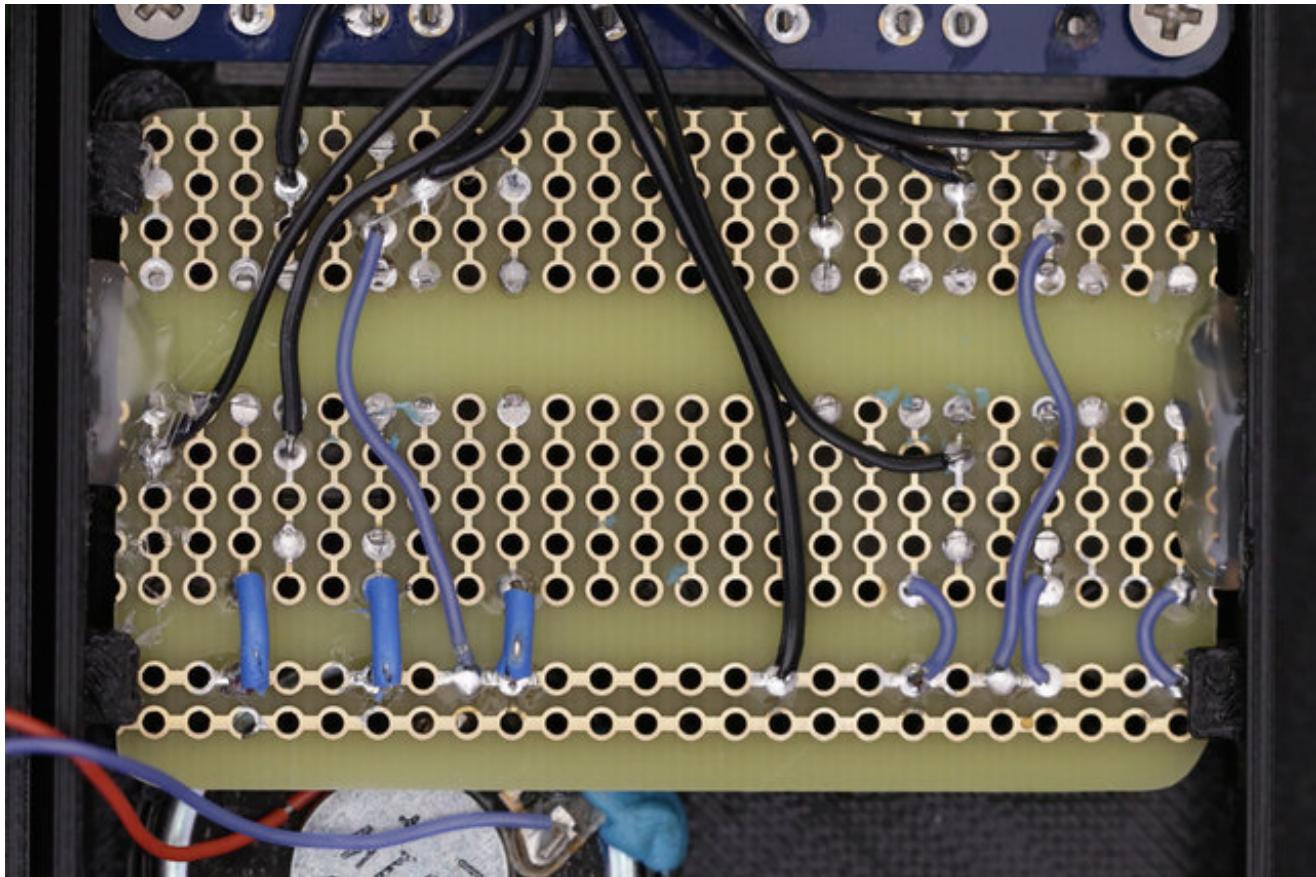
Let's go ahead and snip the points legs from the PCB. This will ensure our perma-proto doesn't puncher or scratch any of the other components.



## 2-Button Layout

OK now we have the perma-proto trimmed, the buttons solder in place and ground wires, we're ready to wire up the PiCable to the perma-proto PCB.

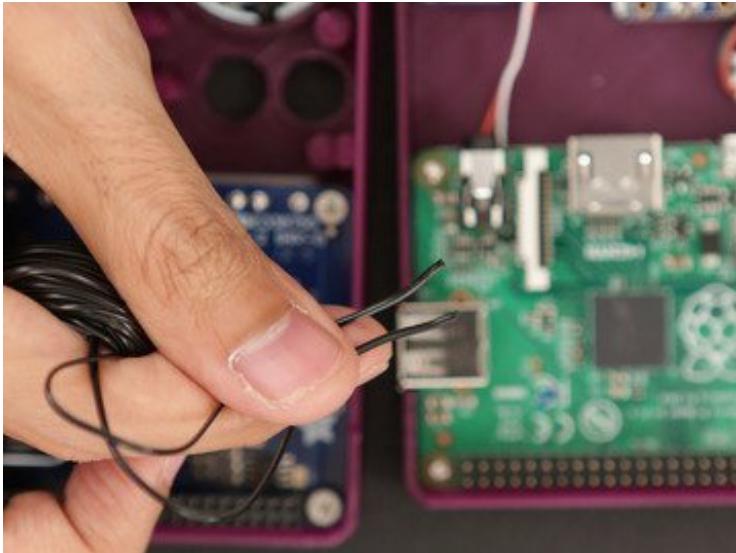
This is a good place to take a break. The next section contains lots more stripping, tinning and soldering!



## 4-Button Layout

If you're doing the 4-button version, here's a photo to reference the wired connections. Each button has a wire connected to the ground rail. Note, the PiCable is already wired to the buttons.

# Buttons



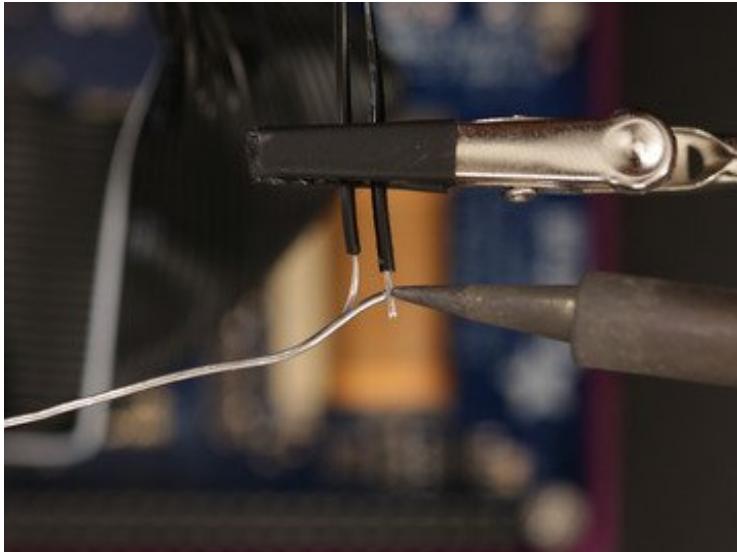
- 



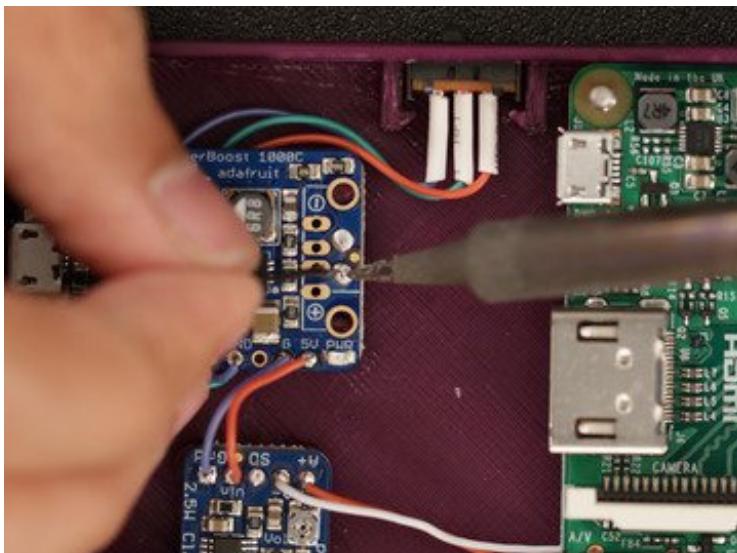
- 

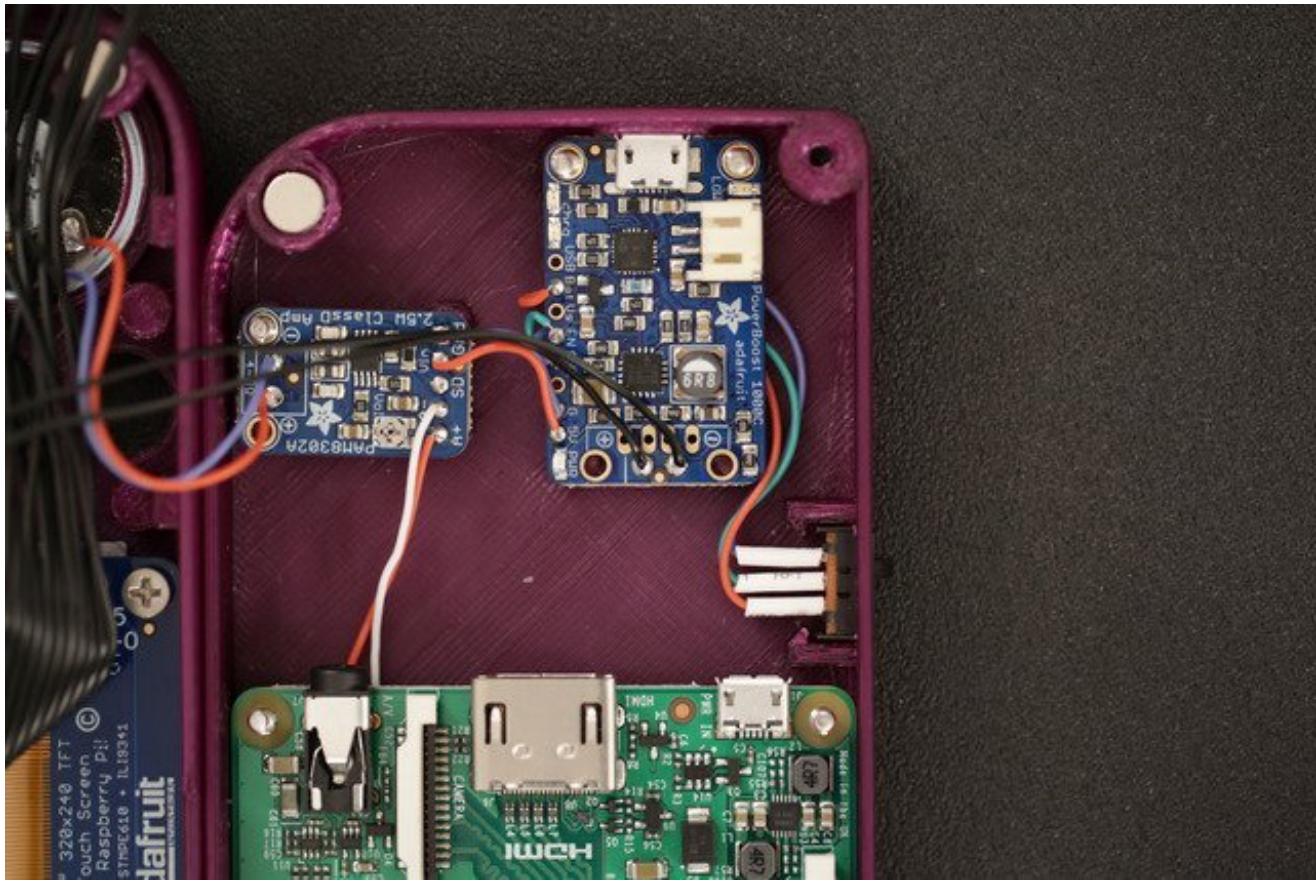
## Connect Pi Cable to Powerboost

Grab the wires from the pi cable and locate wire #2 and #6. Wire #1 is the white colored one. Isolate these two wires and strip the tips off. Secure them to one of the grabbers on the third helping hand. Apply some solder to tin them.



Solder wire #2 to the positive pin on the Powerboost. Solder wire #6 to the negative pin on the Powerboost.





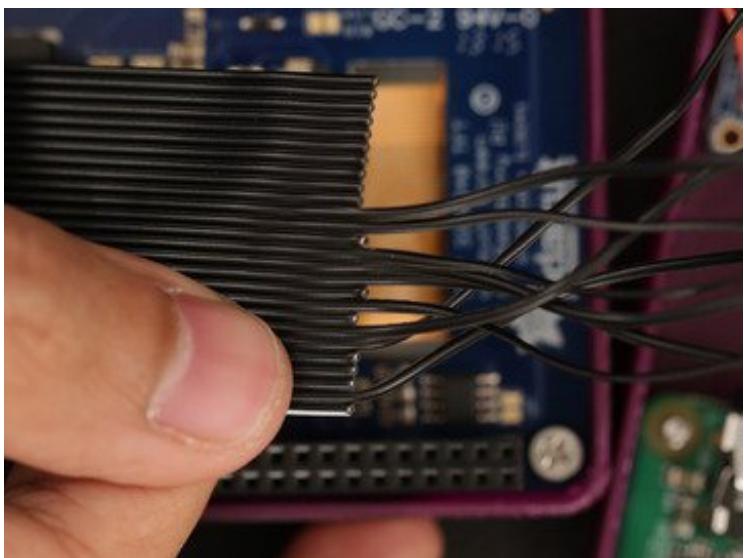
## Soldered Power

Now we should have power wired up from the Pi Cable to the powerboost 1000c. Wire #2 is actually pin 5V on the Pi, wire and #6 is ground.



## Cut extra wires short

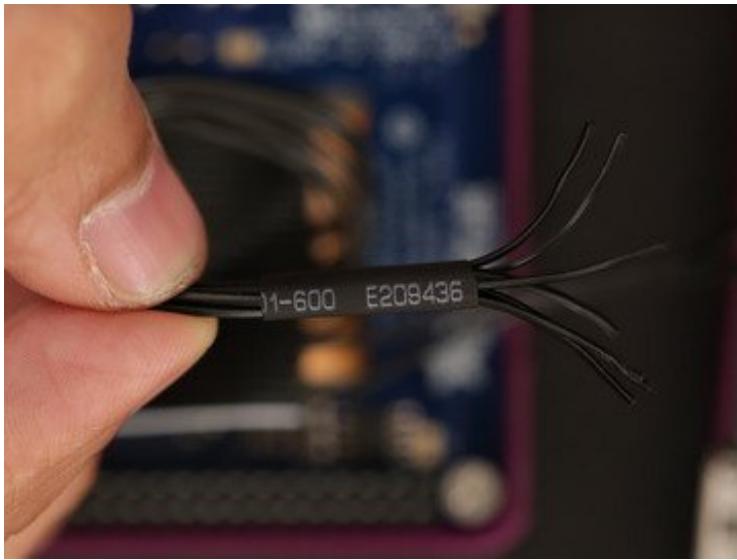
Let's go ahead and cut the wires we won't be using short. Be sure to reference the circuit diagram and double, triple check each wires. Trimming these short will helps keep everything tidy and organized.



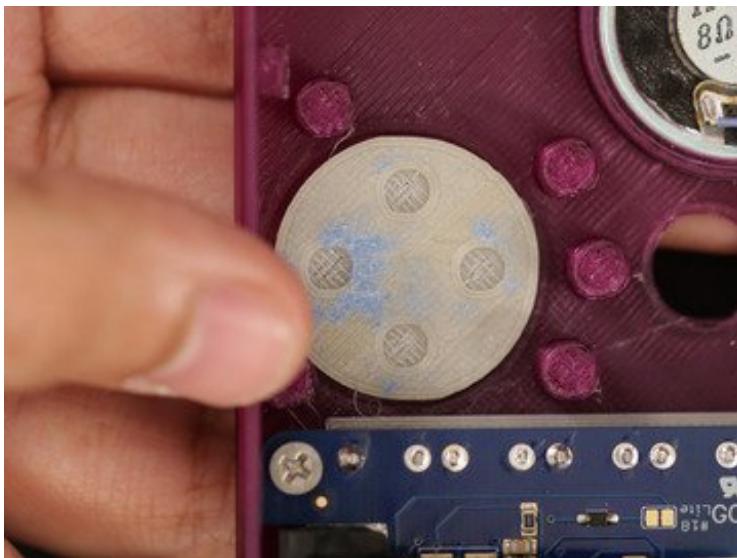
If your making the four button version, DO NOT cut wire #3 and #5!!

## Heat Shrink Pi Cable

Speaking of being organized and tidy, adding a piece of heat

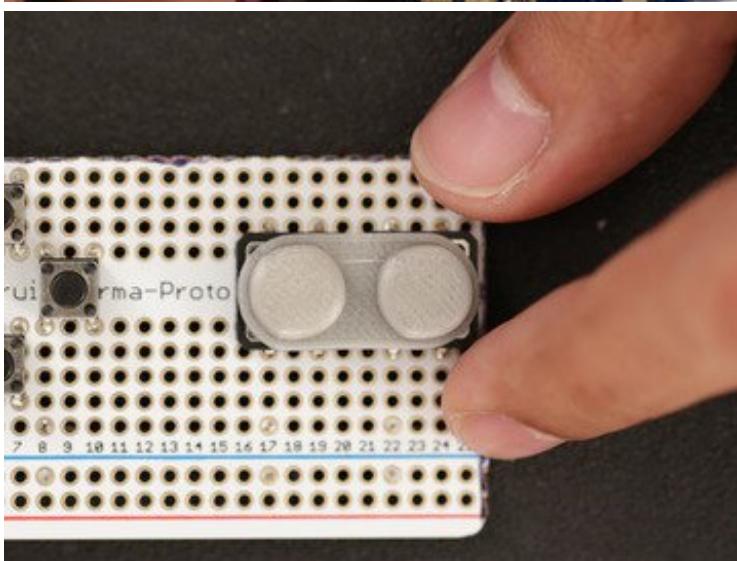


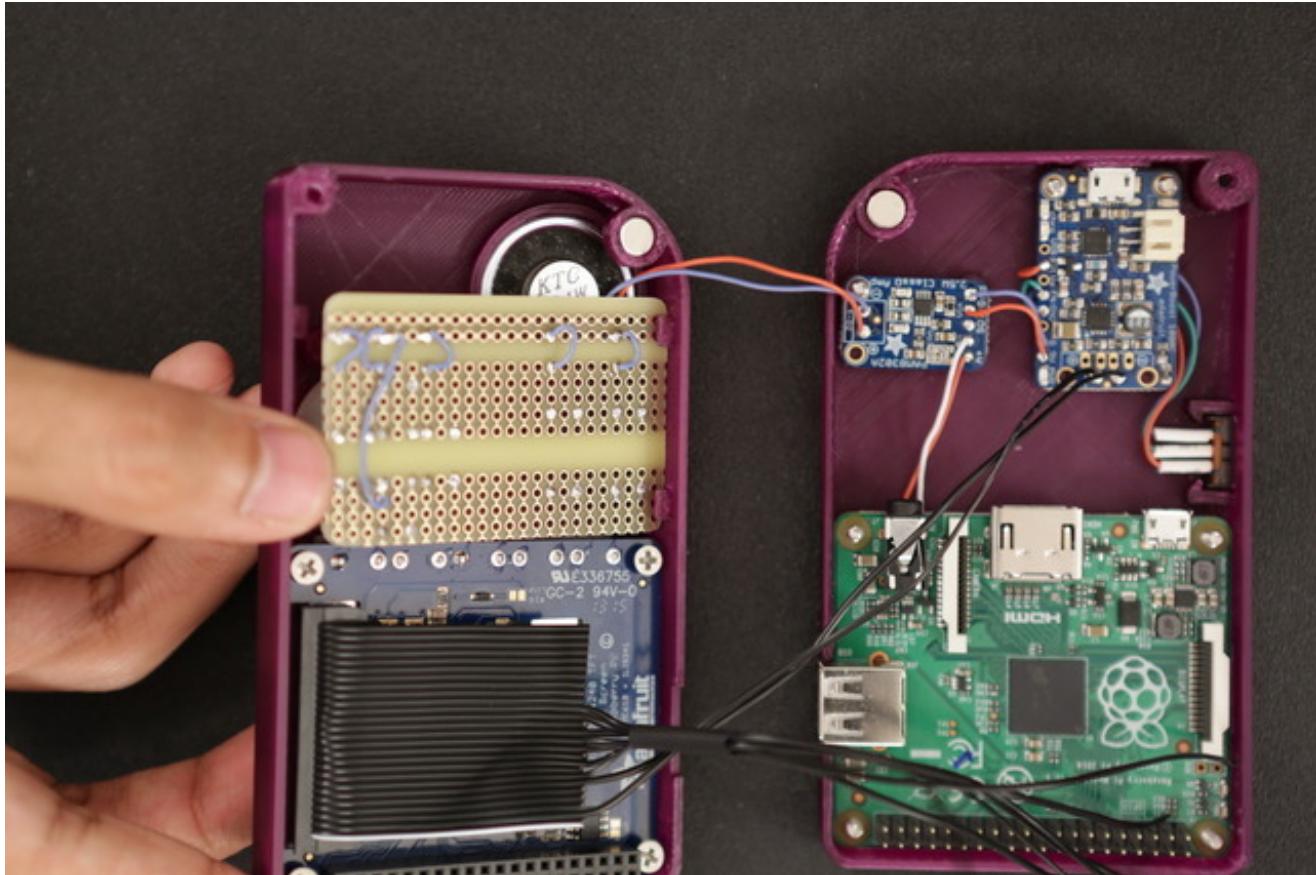
shrinking tubing to the pi cable wires keeps them together. I didn't actually heat it though (depending on the sizing you may want to apply heat to well, shrink it).



## Install Ninjaflex Buttons

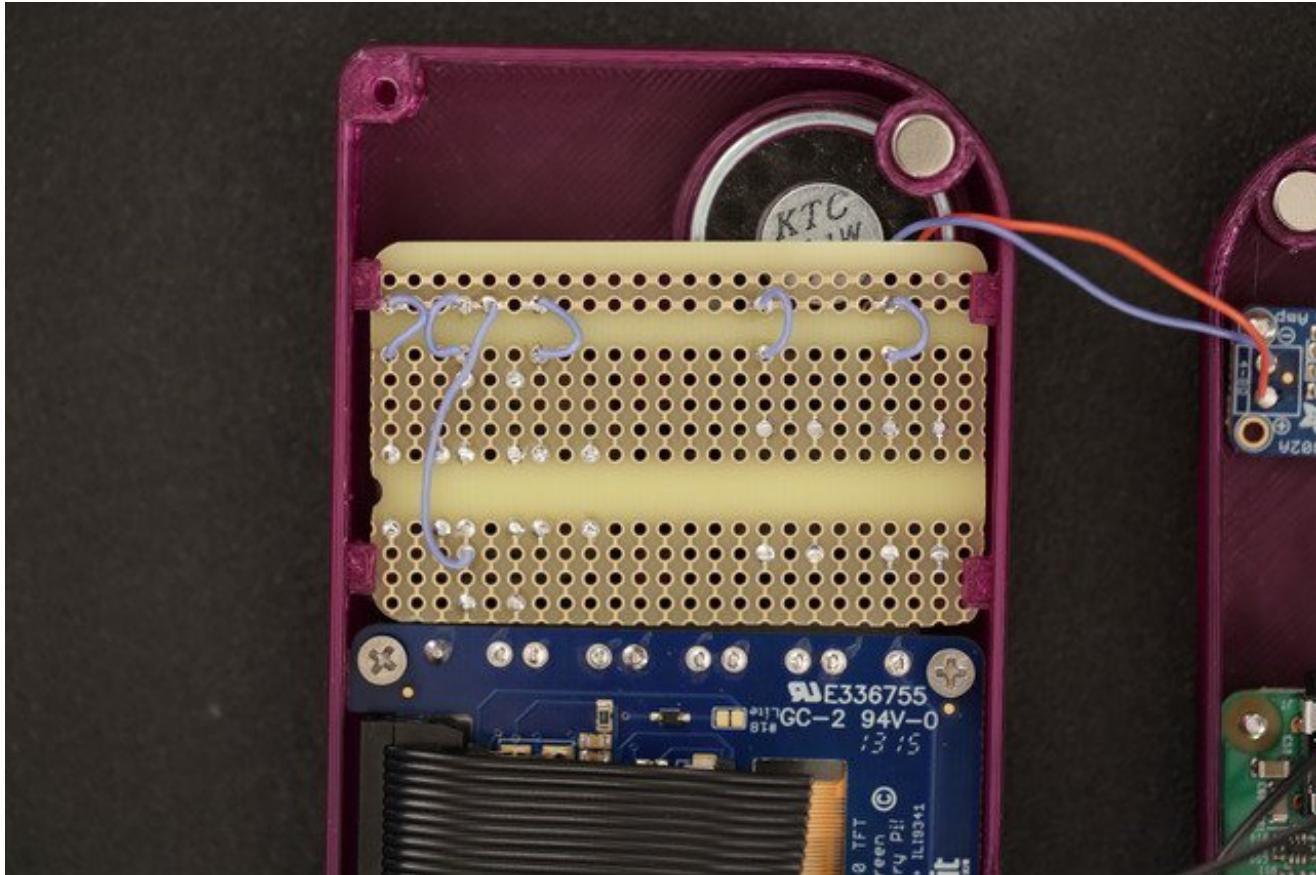
Insert the ninjaflex d-pad into the enclosure with the cut out. No special orientation here, its symmetrical so it should just fit. Place the "A" and "B" ninjaflex button set over the two 12mm tactile buttons on the perma-proto. These should have a tight fit.



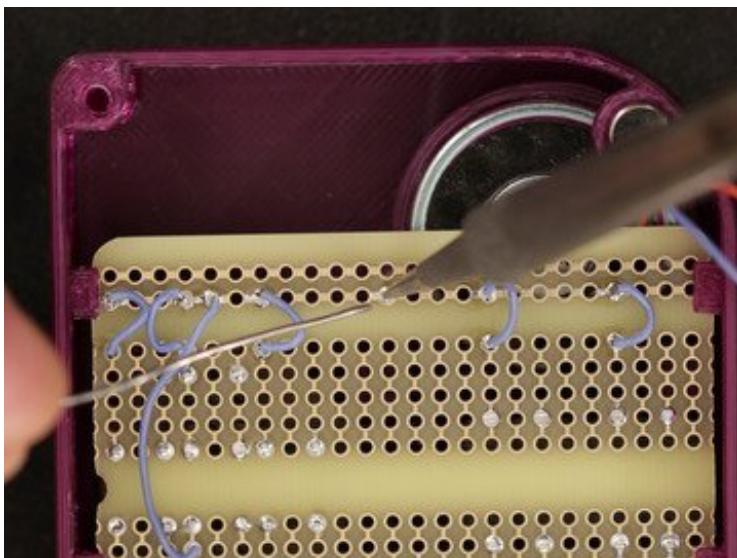


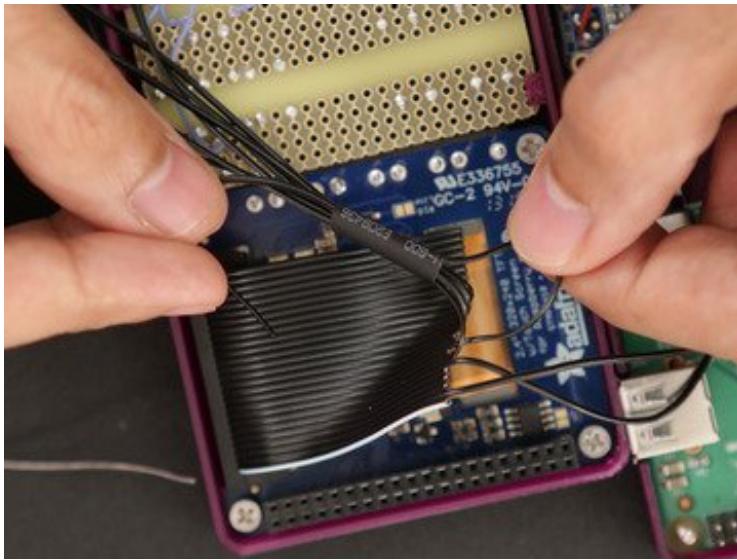
## Install Perma-Proto

OK, grab the PCB and position it over the enclosure. Notice those clips on the inner walls of the enclosure? Those are going to hold the PCB in place. So position and orient the PCB with the appropriate buttons going into the right spots. The PCB needs to be inserted at an angle so that it can fit into place. Insert the PCB with the “A” and “B” button side going in first - the PCB should go underneath the clip.



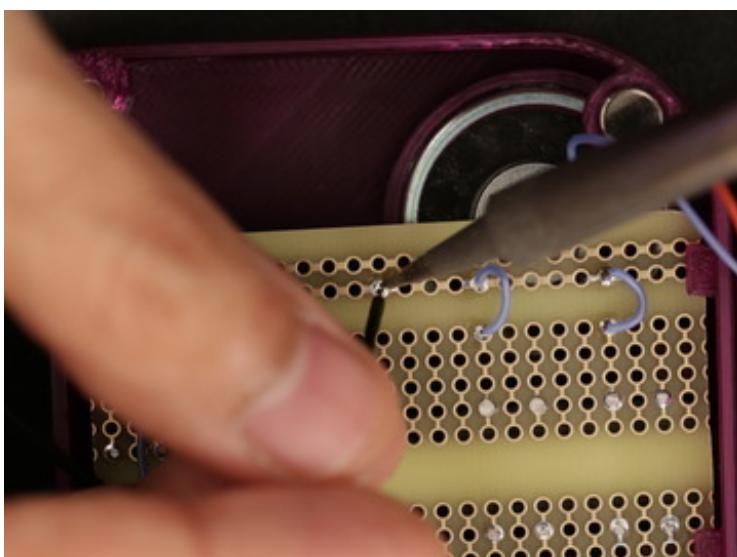
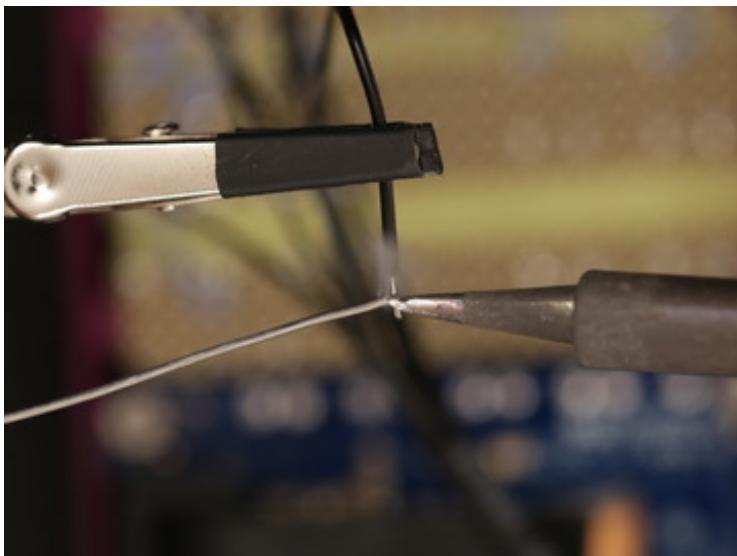
Ensure the d-pad is still in place while inserting the perma-proto PCB. Carefully angle it into place. You need to fit the ninjaflex AB buttons into the cutouts, then press down the PCB to fit into place. This takes a little force and finesse. The clips are fairly strong, so you don't need to be super gentle.

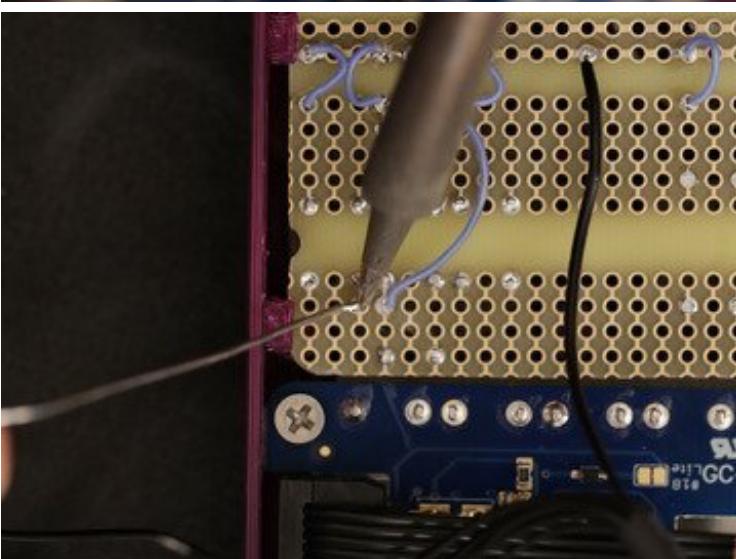
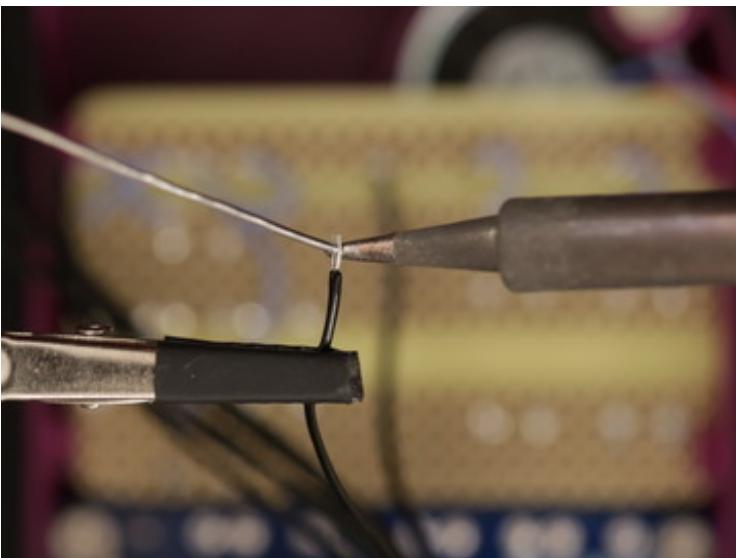
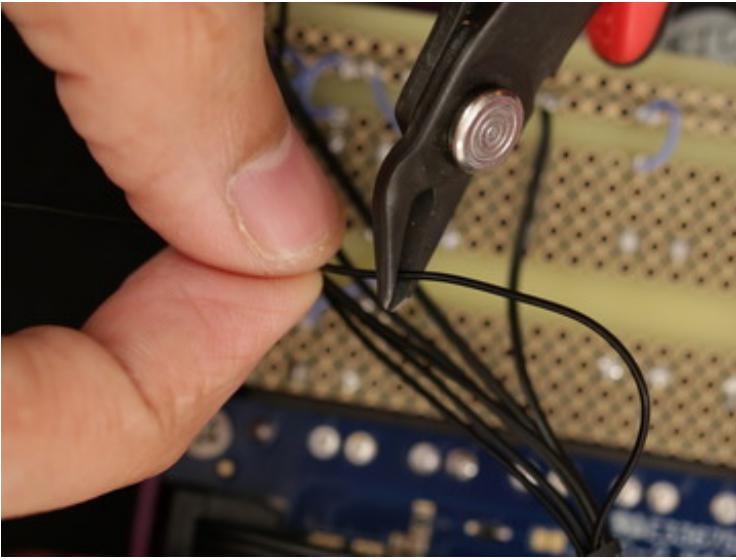




## Wire Ground

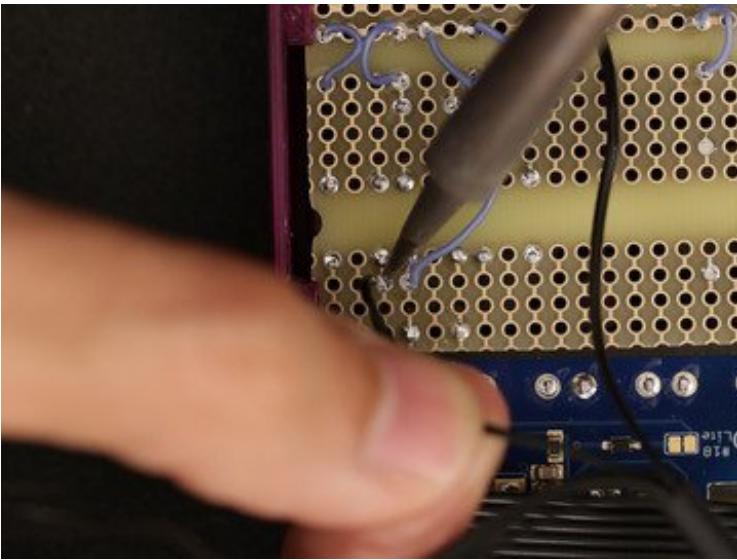
Let's connect the Pi Cable to the ground on the Perma-Proto. We can use a single ground wire from the Pi Cable and solder it to the ground rail on the Perma-Proto PCB. You can count and pull on the wires to determine which one to use. Wire #9 on the Pi Cable is associated with a ground connection.

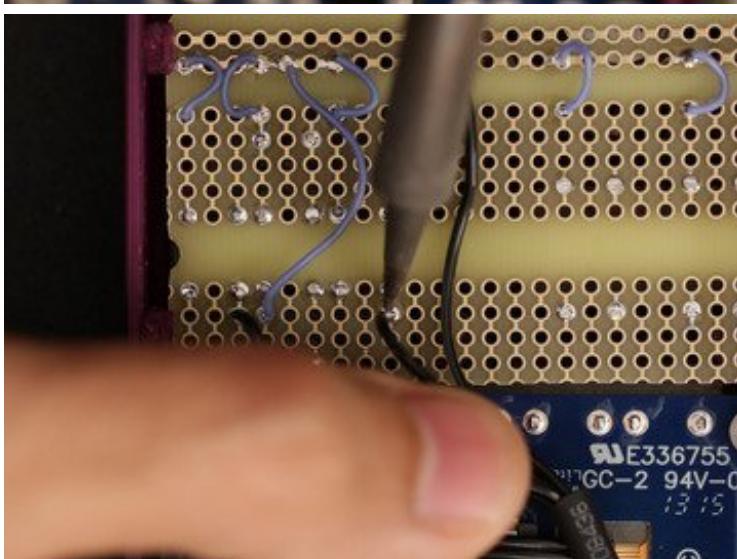
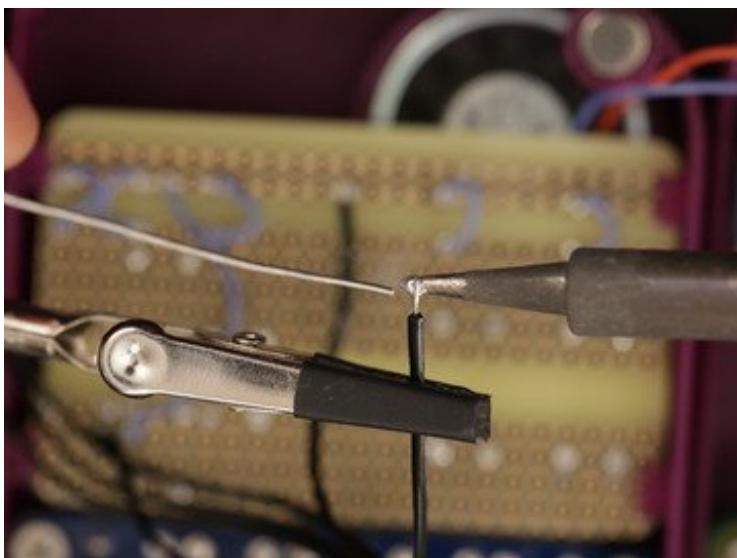
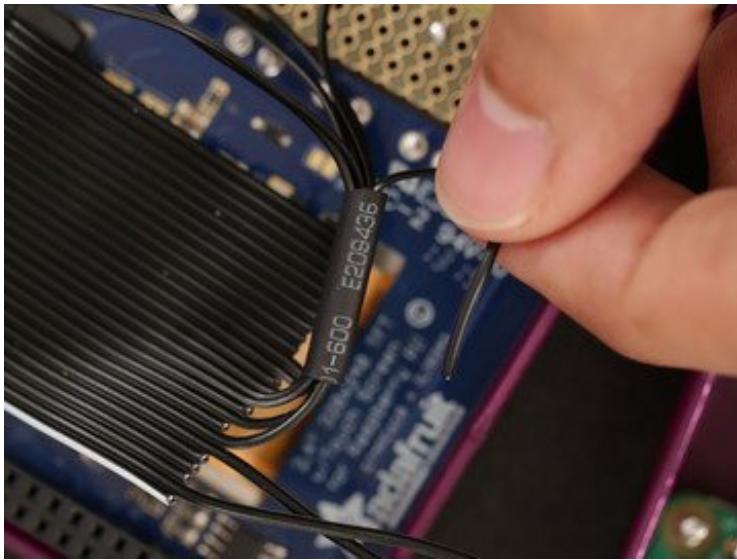




## Wire Left Button

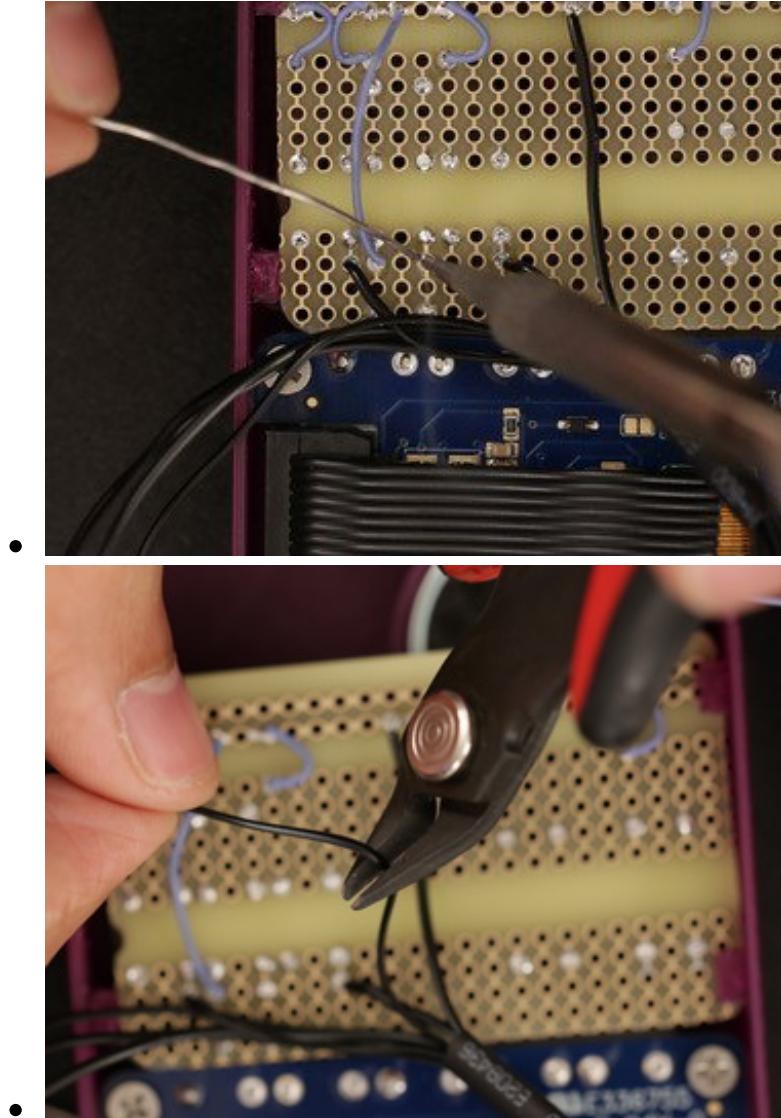
Find an available pin on the perma-proto that is linked to the left button. This should be on the opposite pin of the button where we wired ground. Locate wire #7 on the pi cable and position near it the left button. Measure how long it needs to be and cut short - leave a bit of slack so it's easier to solder. Strip and tin the wire. Then solder the wire to the perma-proto PCB.





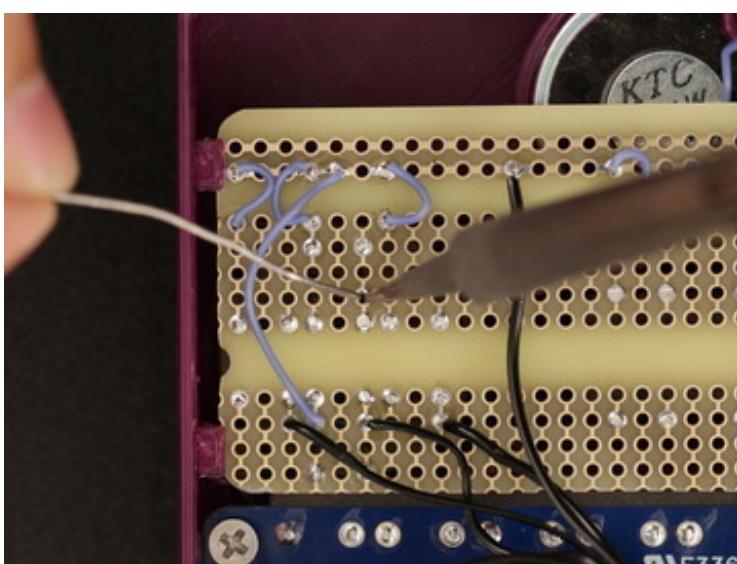
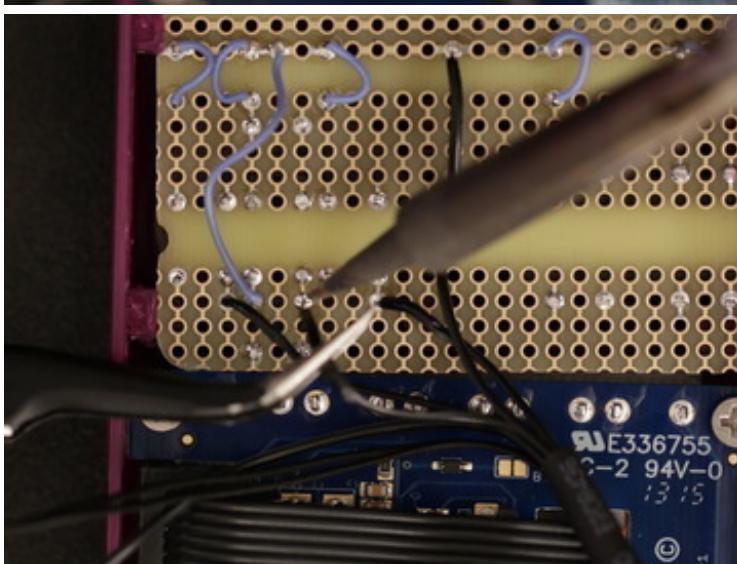
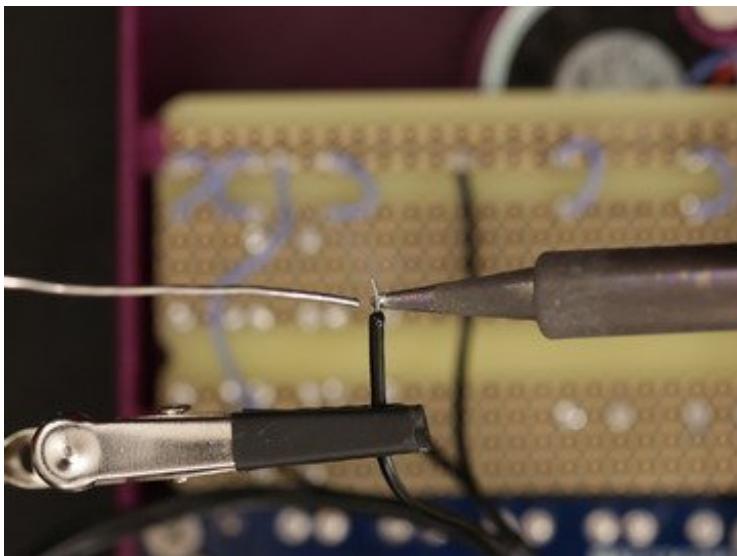
## Wire Right Button

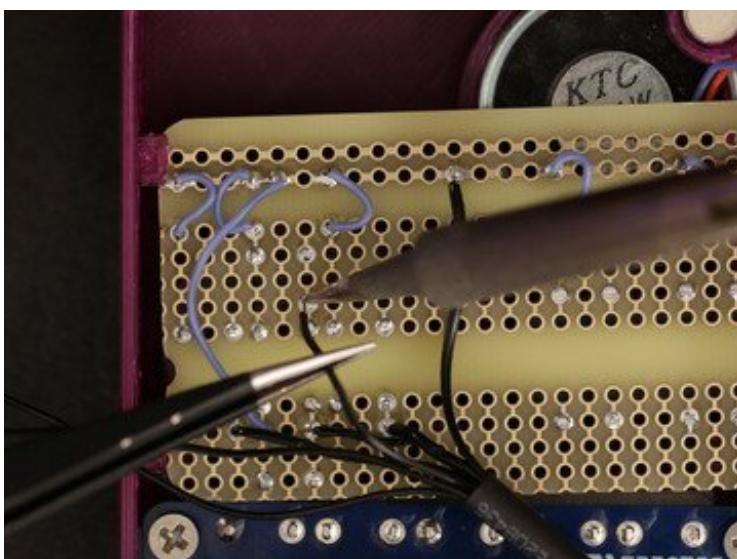
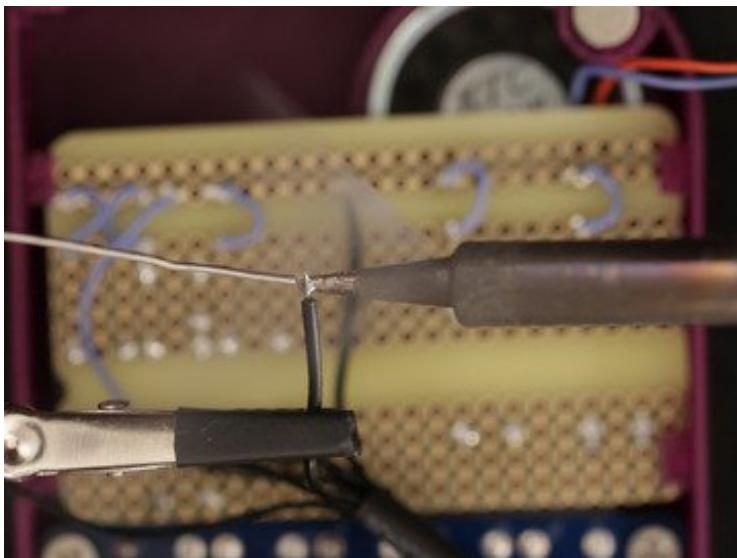
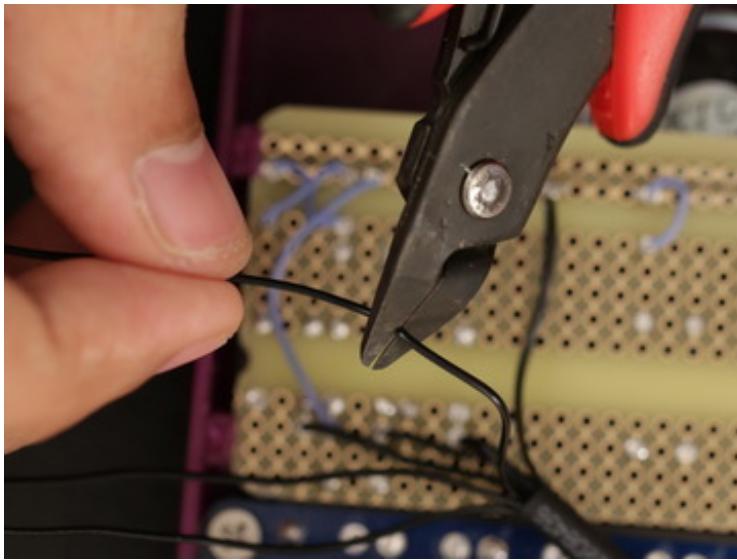
Now that you did the first button, you'll be able to do the song and dance easier. Let's locate wire #11 on the pi cable - this is going to the right button. Just like we did on the left button, measure how long it needs to be, then cut, strip and tin. Solder this one to the one of the available pins in the same row of the right button.



## Wire Up Button

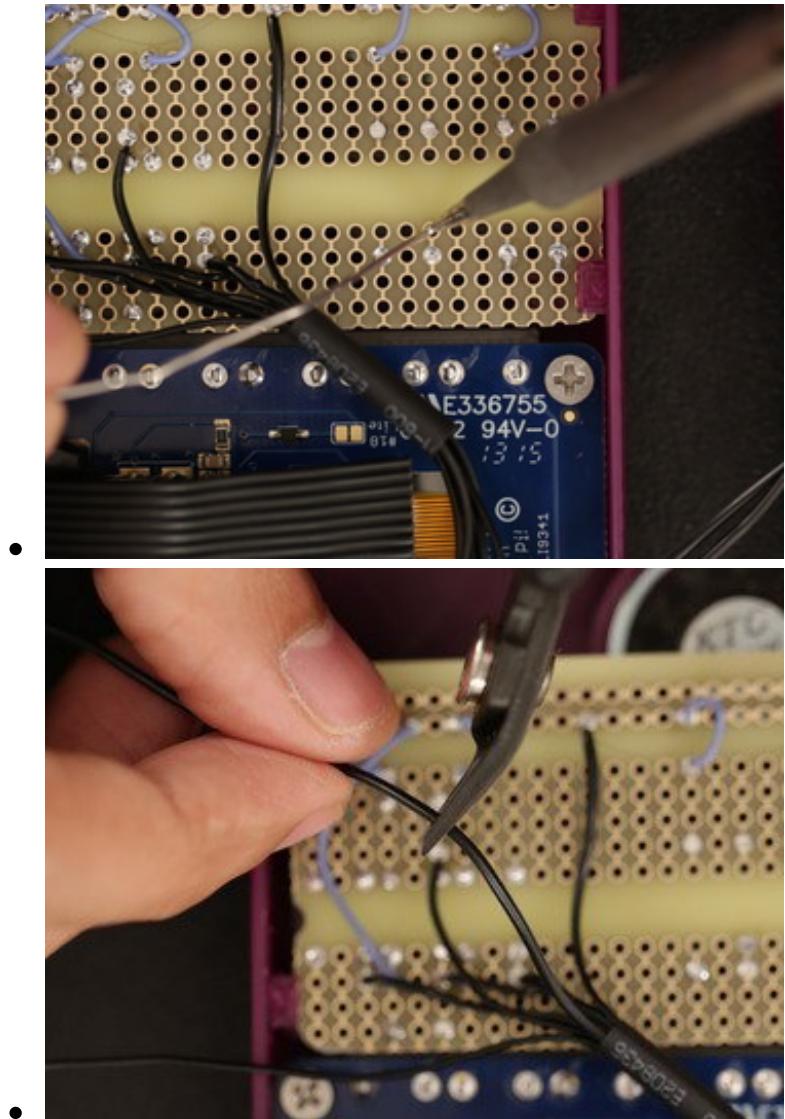
Up next, it's the up button! Locate wire #12 and do the song and dance of measuring, cutting, stripping, tipping and soldering.





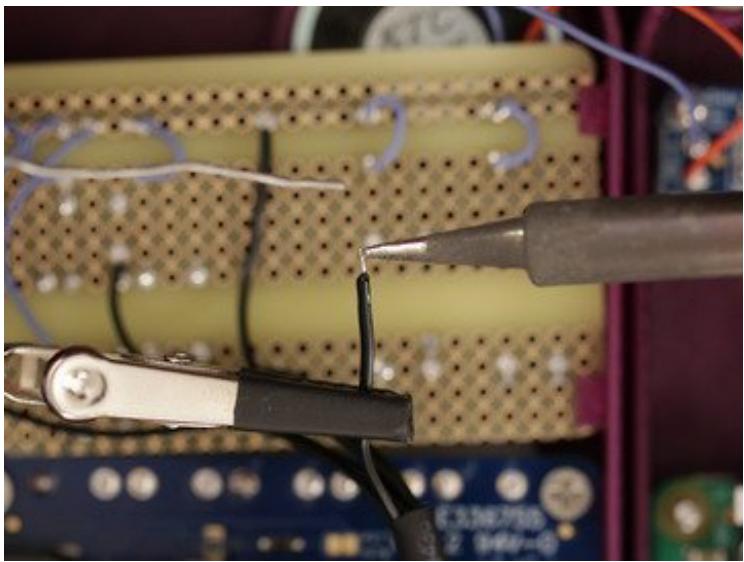
## Wire Down Button

Locate wire #13 and solder this one to the down button. You know the drill!

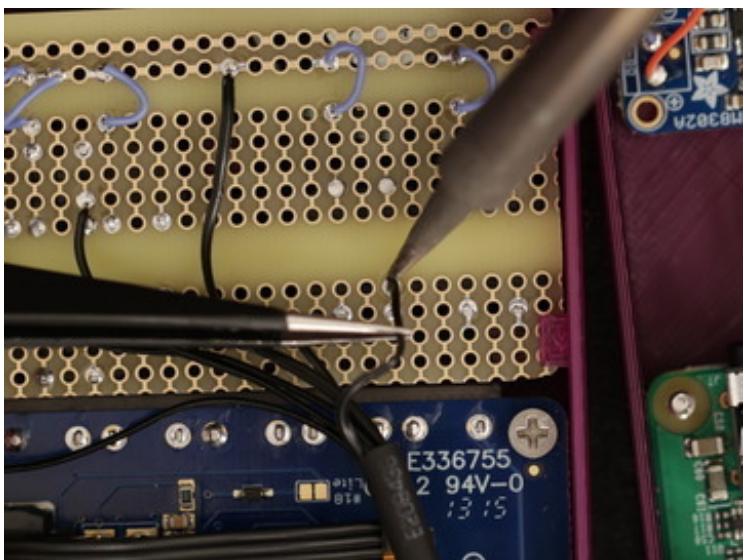


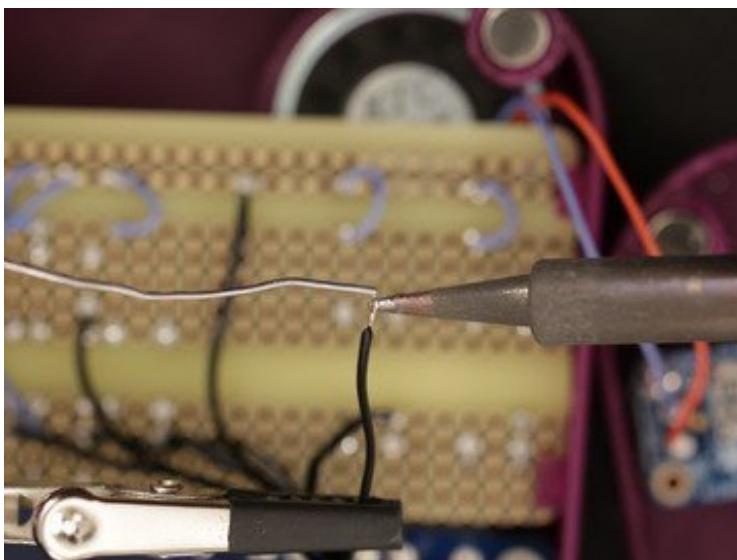
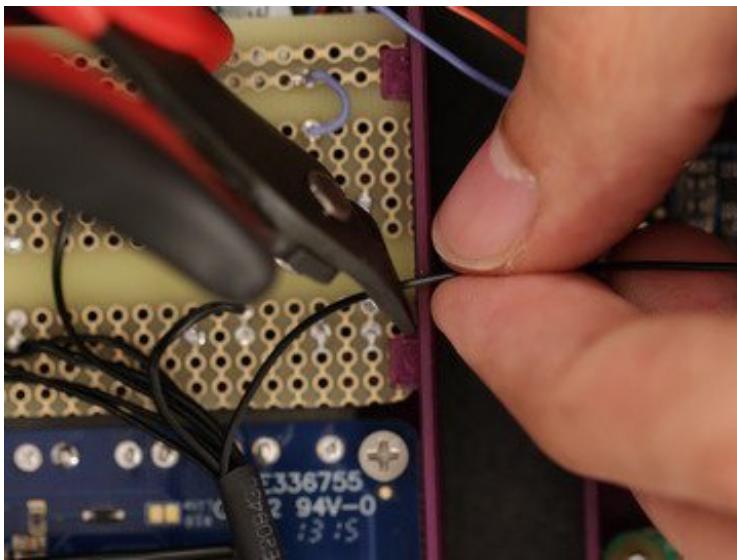
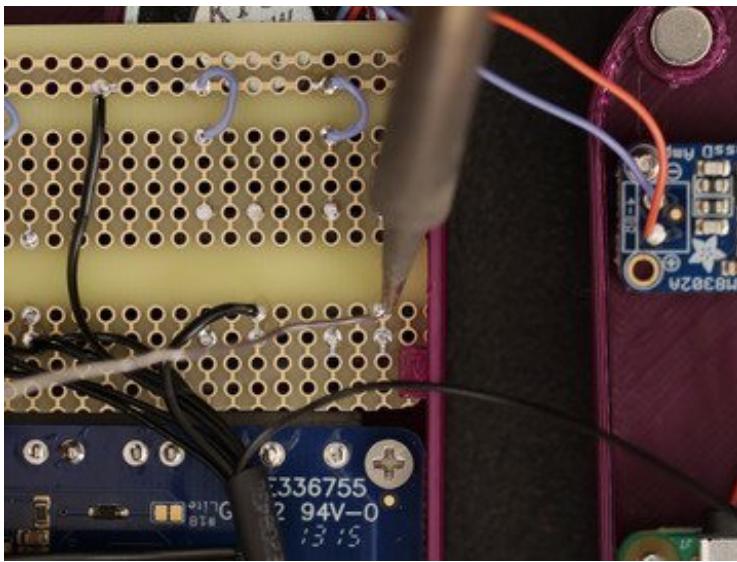
## Wire A Button

Get wire #15 and connect this one to the "A" button. Double check to see you're selecting the correct button - orientation might mess with you a little after



soldering all those wires.

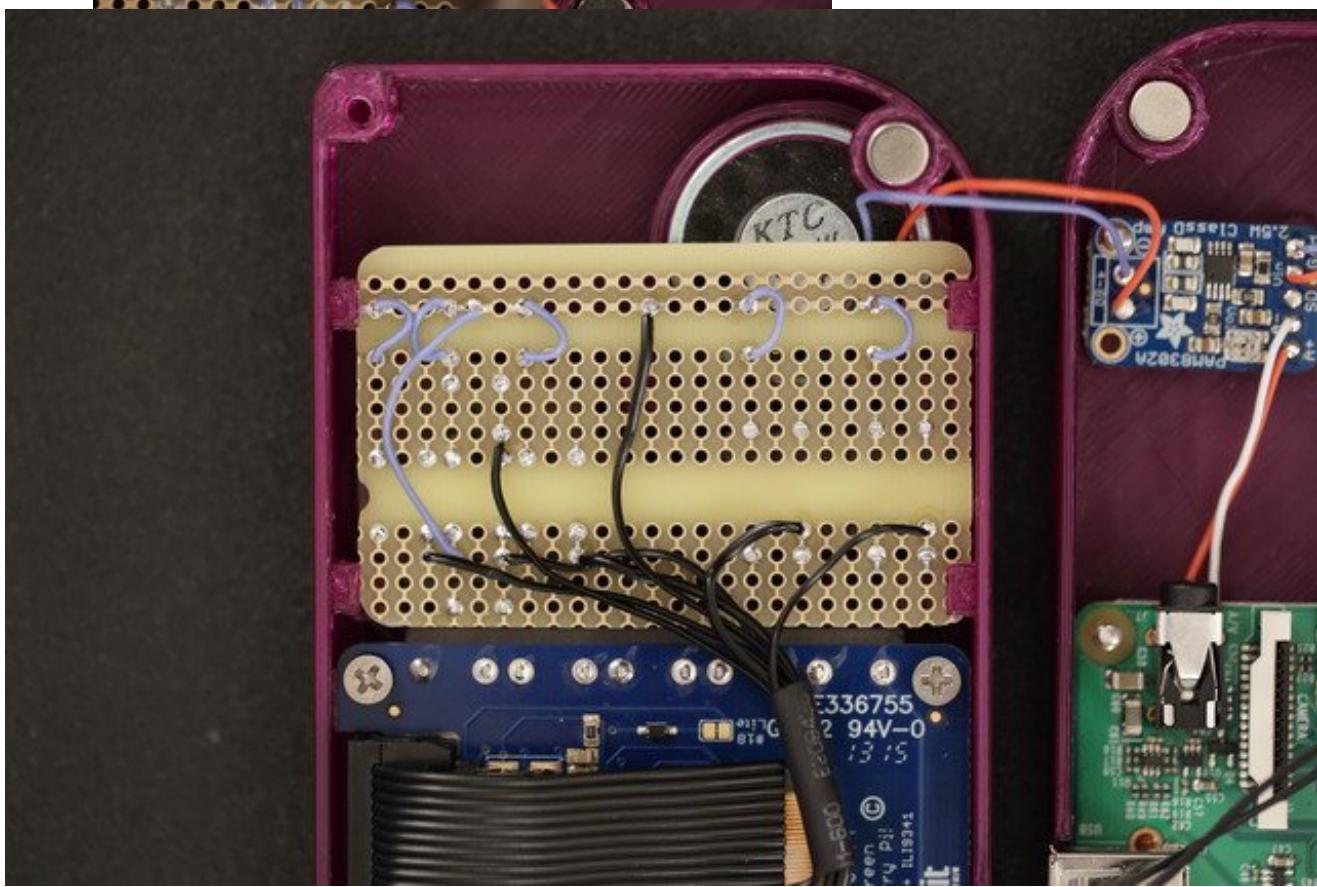




## Wire B button

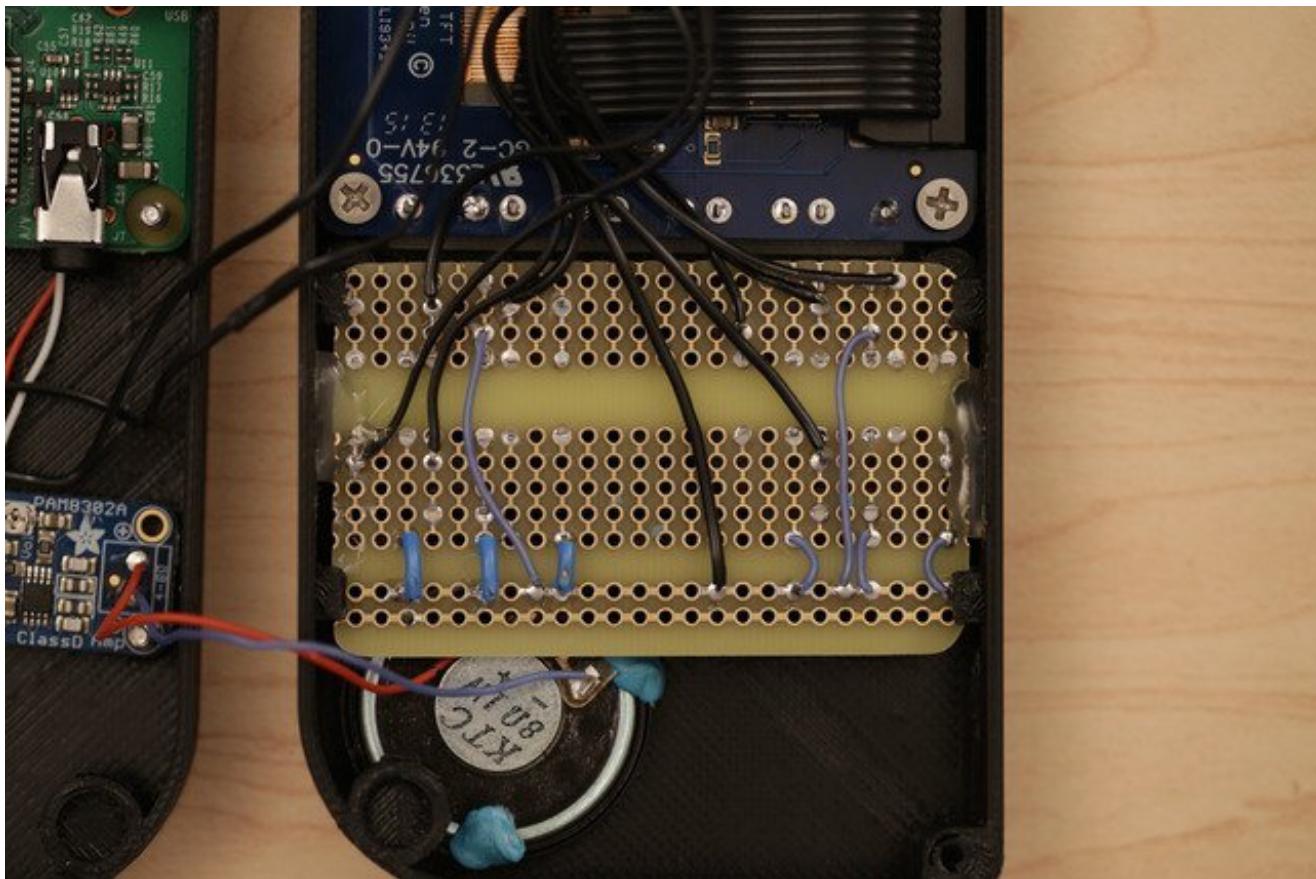
OK now we should be left with just one wire - Safe to assume its wire #16 which will connect to the "B" button

- 
- 
- 
-



## Wired Perma-Proto to Pi Cable

It's starting to look finished after soldering all those wires. It's a good idea to double check all the buttons, pins and wires to ensure everything is correct. If you got any of them mixed, it is possible to remap them in software. But make sure ground is ground and nothing is intersecting (ie, two inputs wired to a single button).

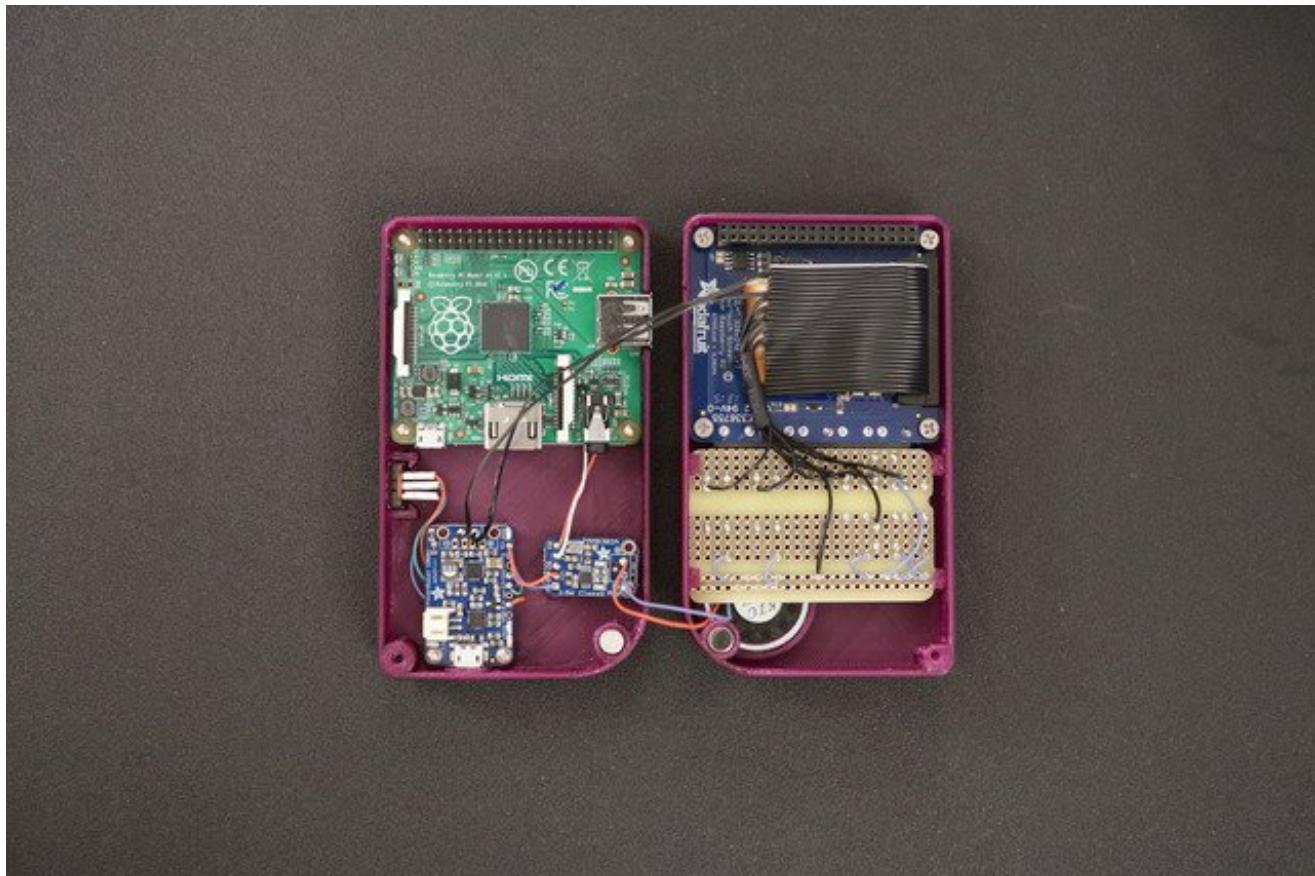


## Perma-Proto (Four button layout)

If you're making the four button version, you'll need to connect Wire #3 (GPIO 2) to the button you want to be "X". Connect wire #5 (GPIO 3) to the button you'd like to be "Y".

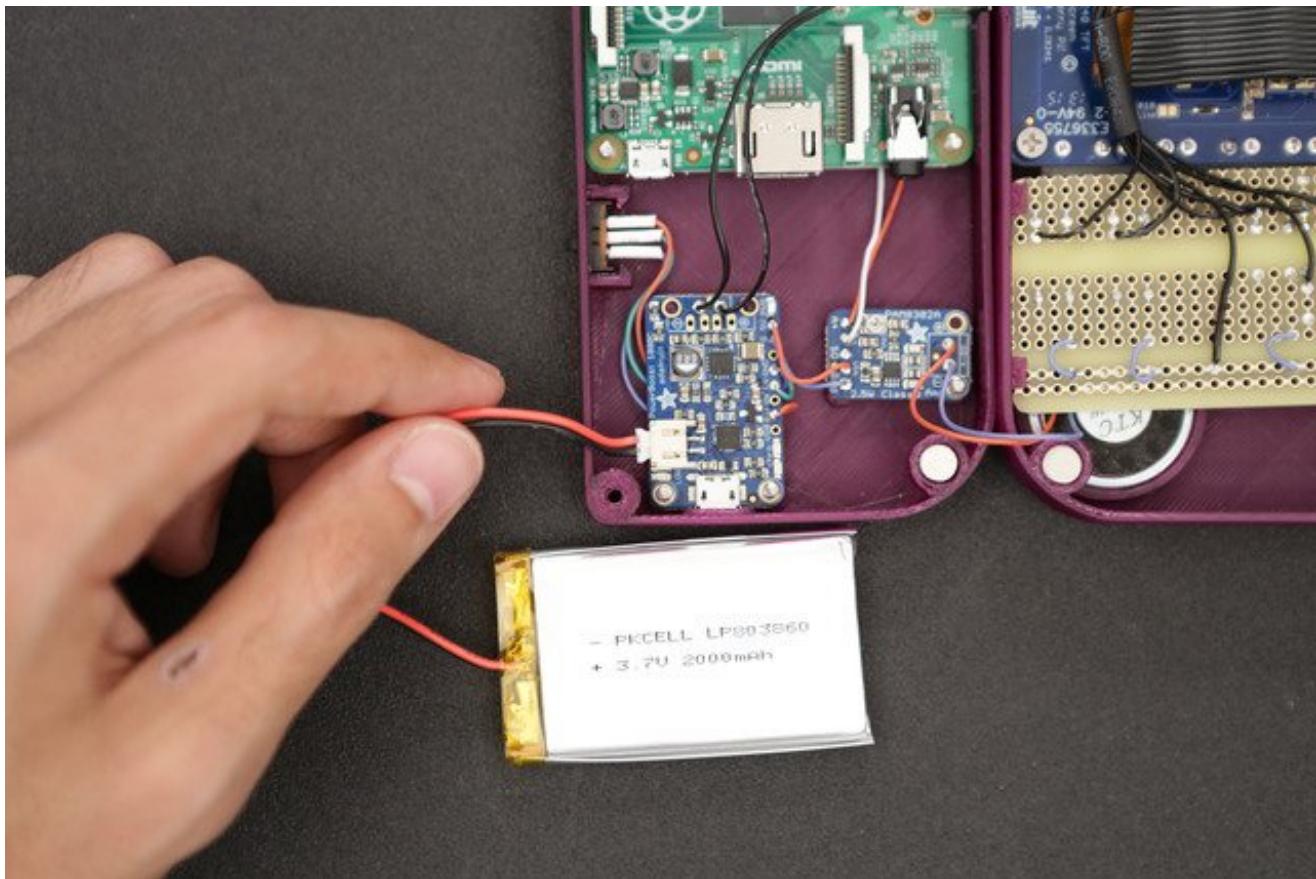
That's all the soldering for this project. Congratulations, you are now a master solderer! Next up, we need to add the battery and close the enclosure. Almost done!

# Assembly



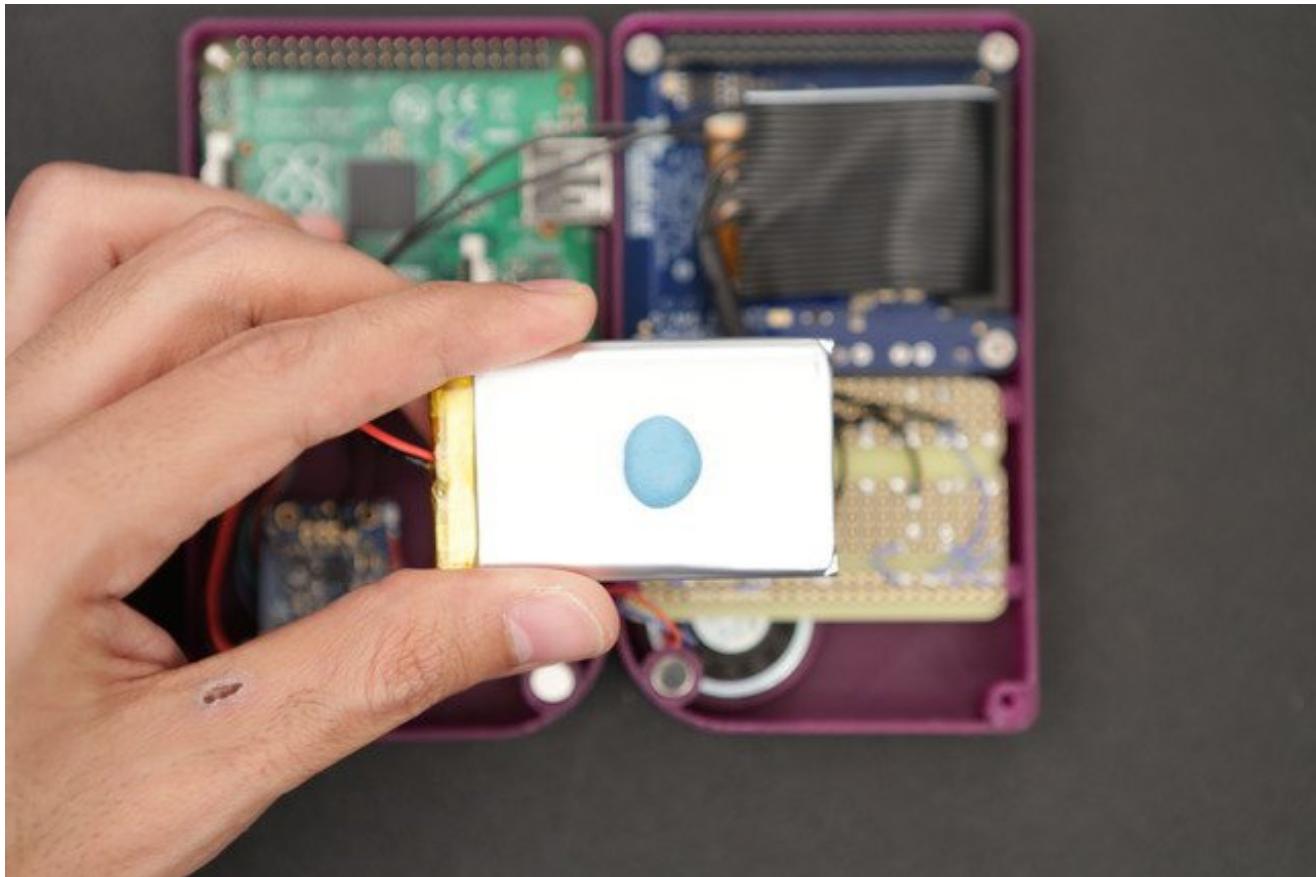
## Final Wire Check

It's a good idea to check all the wiring before closing it up. It is relatively easy to open and adjust but you should consider reviewing all the connections. If you think you have too much excess wire, now would be a good time to can trim them down.



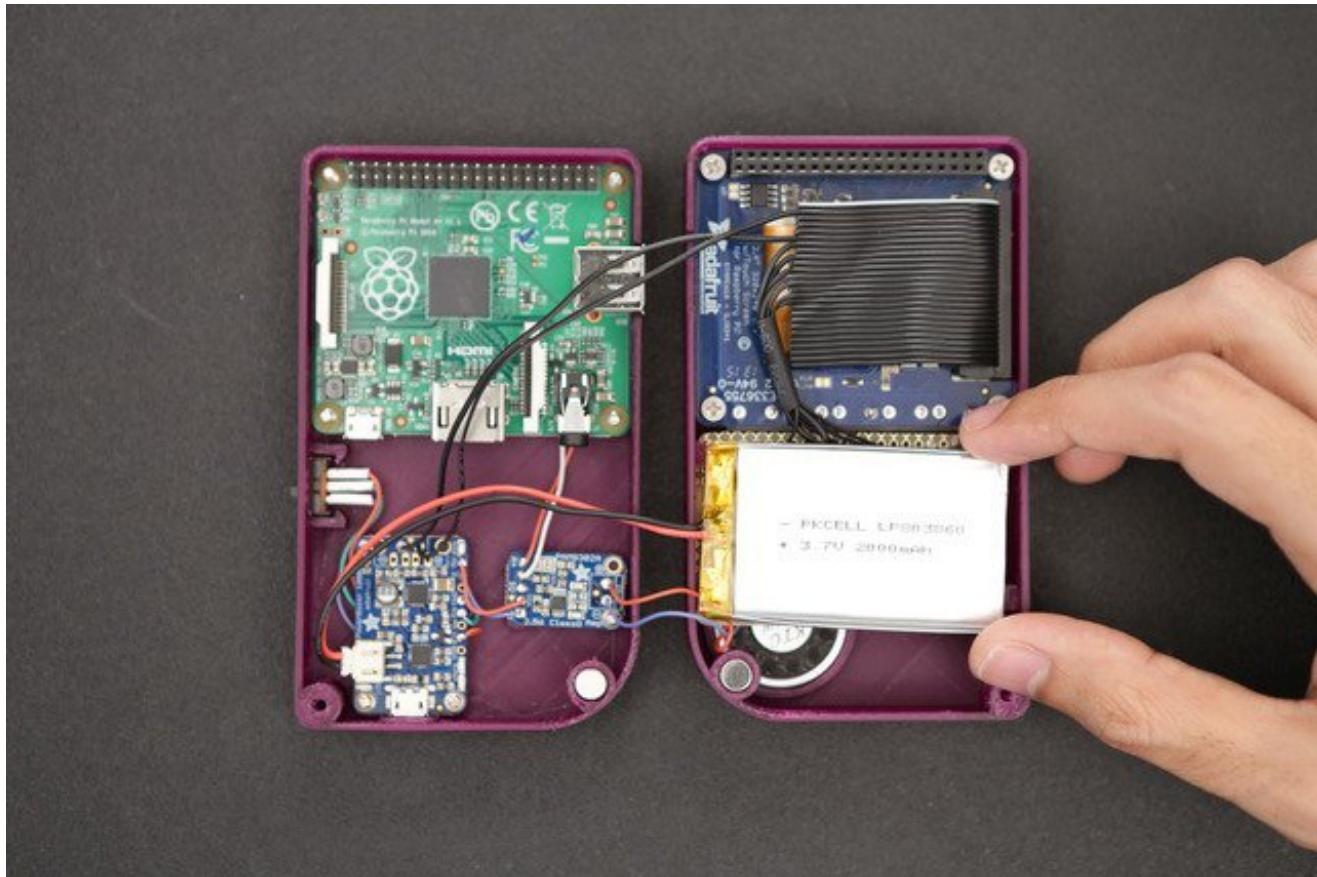
## Plug Battery

Get the 2000mAh battery and insert the cable to the JST port on the powerboost 1000c. You might want to bend it at a right-angle so you can fit it in the case.



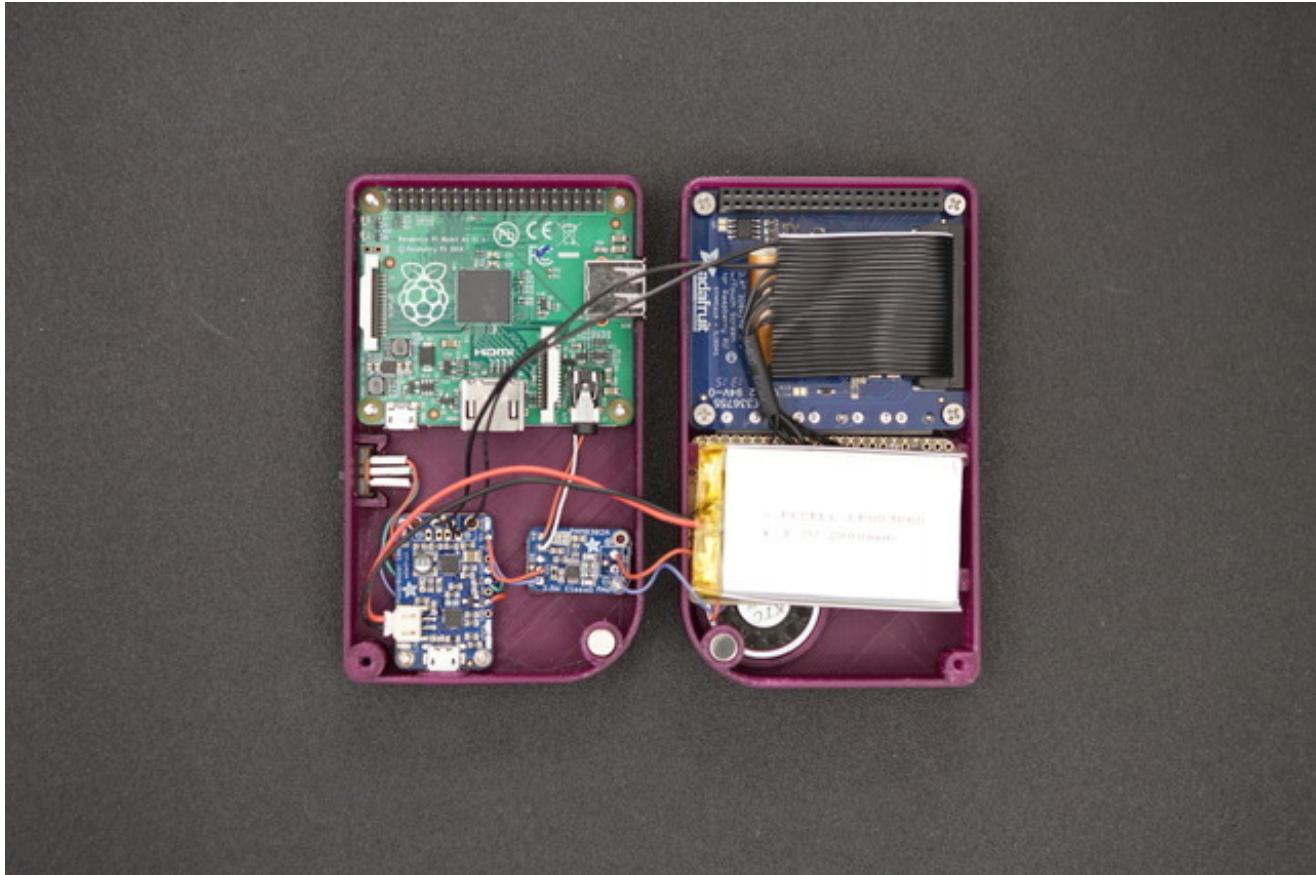
## Tack Battery

I added a small piece of fun-tak to the back of the battery so it stick the back of the Perma-Proto PCB. This fun-tak stuff from loctite works really well cause it hold it in place, but easy to remove and doesn't leave behind any sticky gunk.



## Stick Battery

Go ahead and stick the battery onto the Perma-Proto PCB.



## Installed Battery

Seems to be the best place for the battery. I would recommend wrapping this up in gaffers tape but when I did that it no longer fit, so at least you know now. We snipped off all the pointy stuff anyway, so it should be safe. Also note there's actually not too much force or pressuring being applied to the battery when it's enclosed.



## Close GPIO headers

With two hands, you want to pick up both parts of the enclosure and start closing it together. The first thing you wanna do is line up the header pins from the Pi to the socket on the PiTFT. Once their lined up, squeeze the two parts together to connect it.



## Wire Kink Check

The top section should be closed now, leaving the bottom and sides a bit open. This is where you need to make sure none of the wires are outside the enclosure - it's easy to close it up and kink a wire when doing this. You'll most likely get the two magnets to snap together here since they're pretty strong.



## Fasten Last Screw

OK we're totally almost finished. Grab a single #4-40  $\frac{3}{8}$  flat phillips machine screw and insert it into the bottom corner on the back of the enclosure. Hold the two parts together while you fasten the screw all the way through the enclosure. The whole screw will go inside the enclosure and fasten two standoffs together. This single screw keeps the two halves together. The magnet keeps the other corner closed while the top is closed with the aid of the GPIO header. And that's about it!



## Final Build

This build was a lot easier to wire and assemble than the previous PiGRRL projects. Using the perma-proto and tactile buttons in my opinion are more hassle free and ultimately feel better, too. There's a bit of extra room in the enclosure for other components and buttons, so feel encouraged to customize.

The audio out of the Pi A+ isn't the best quality so there may be some hissing. You can minimize this by increasing the volume on the Pi as much as possible using **alsamixer** and/or the emulator's sound controls and adjusting the audio amp volume down to a comfy level



## I Made One, YES!

If you made one, we want to see it!! We ask that you please share it with us. Send us a photo on twitter, better yet posit it on Thingiverse - Take photos of your build and post a **make** on the [thingiverse page](http://adafru.it/f6Z) (<http://adafru.it/f6Z>). We'll feature your build on our 3D Hangouts show! You can even share it with us on Adafruit Show & Tell show! Post it on any social channel - use hashtags #PiGRRL #Adafruit so we can find it easily.

# RetroPie: Improving Emulator Performance

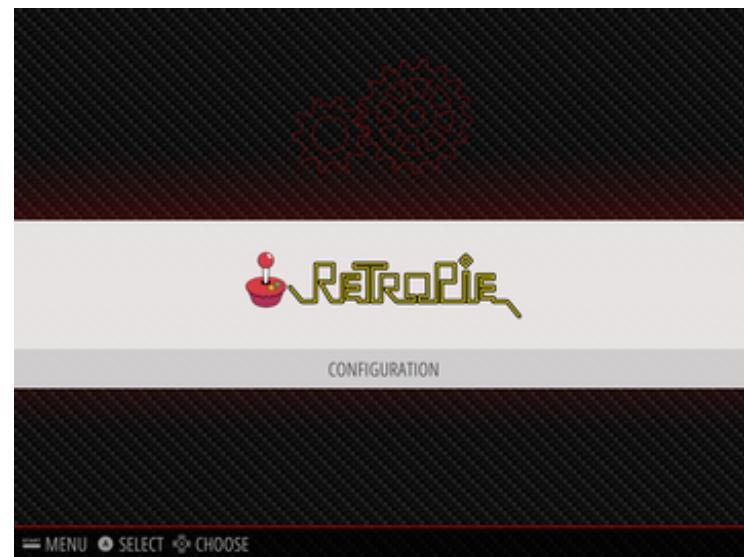
On **single-core** Raspberry Pi boards like the **Model A+** and **Pi Zero**, emulation speed is sometimes less than stellar...the screen may update like molasses, or sound may be choppy. Let's look at some ideas to fight back...

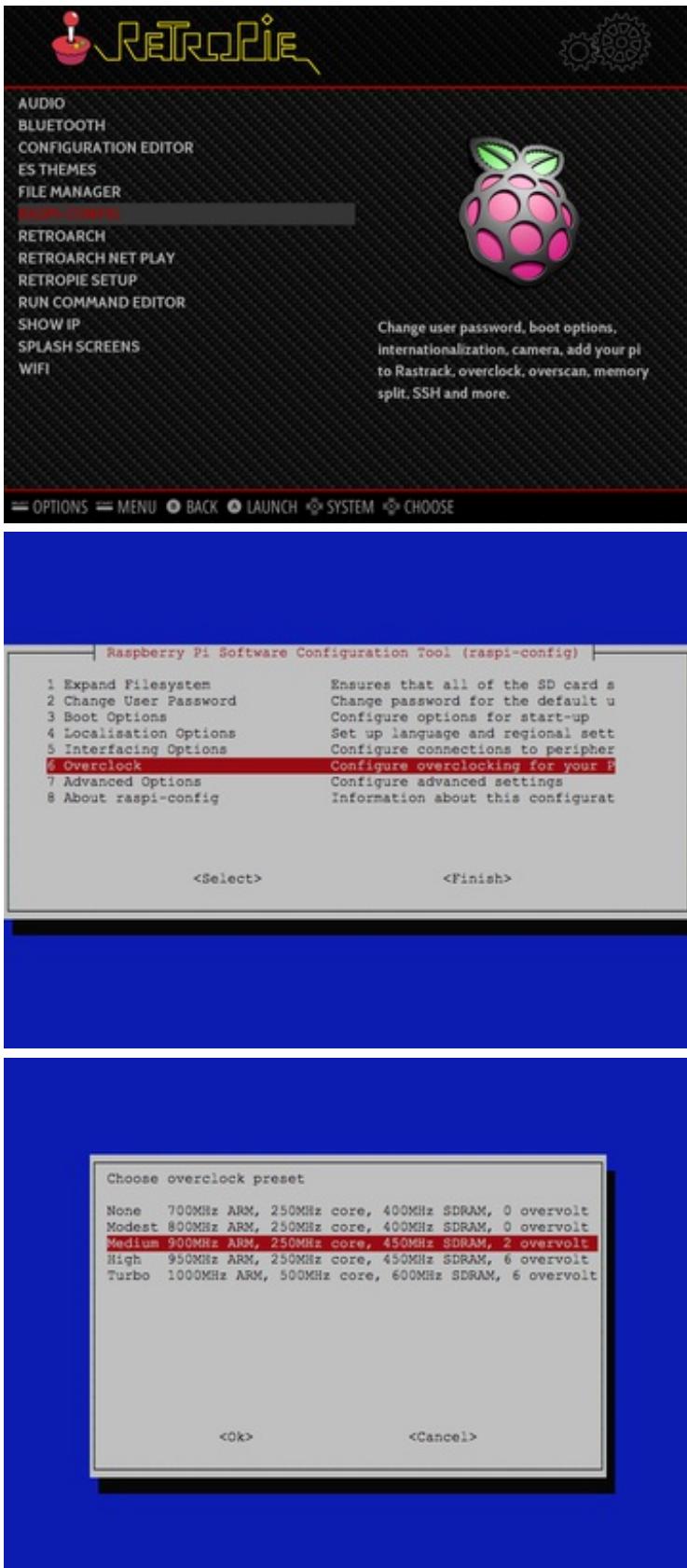
**All of these methods require a USB keyboard attached.**

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 overlock 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.

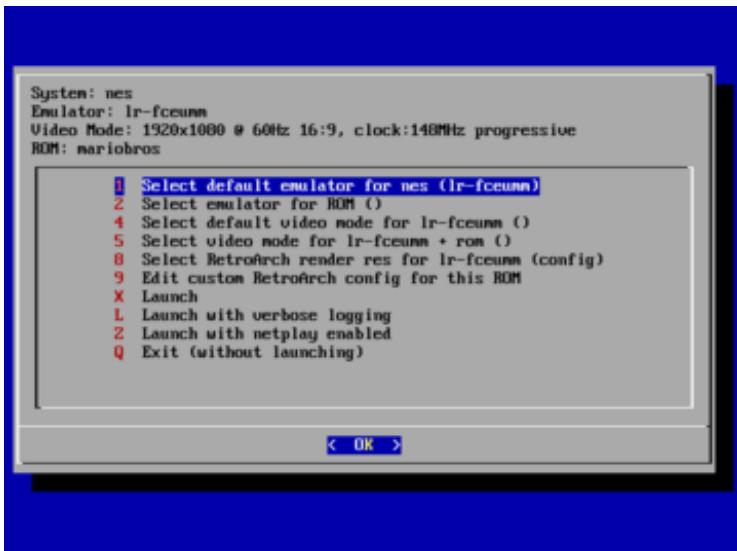
If you’re using a **PiTFT display**, the “core” frequency setting should **not exceed 300 MHz**;

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



- 



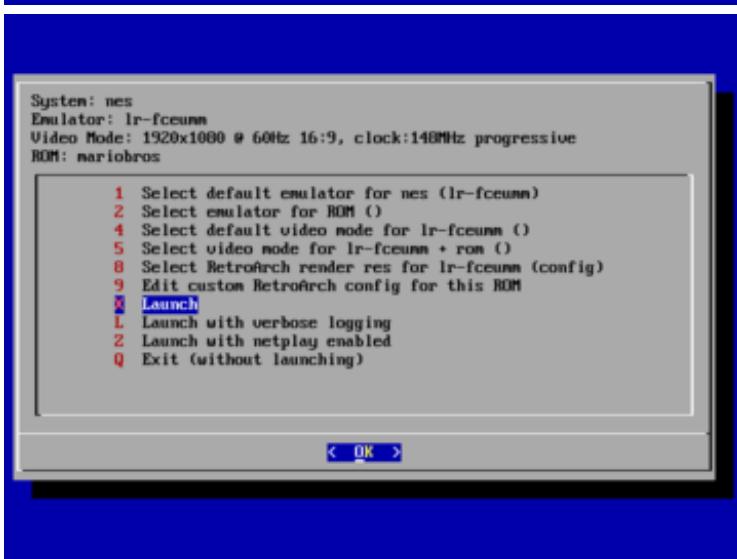
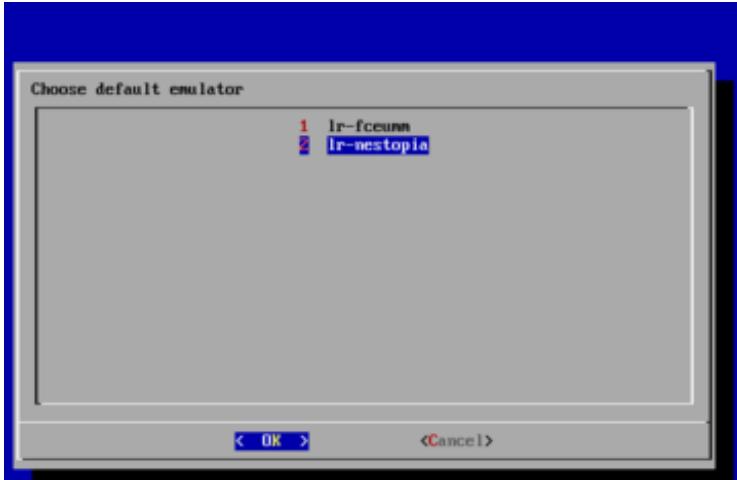
- 

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.

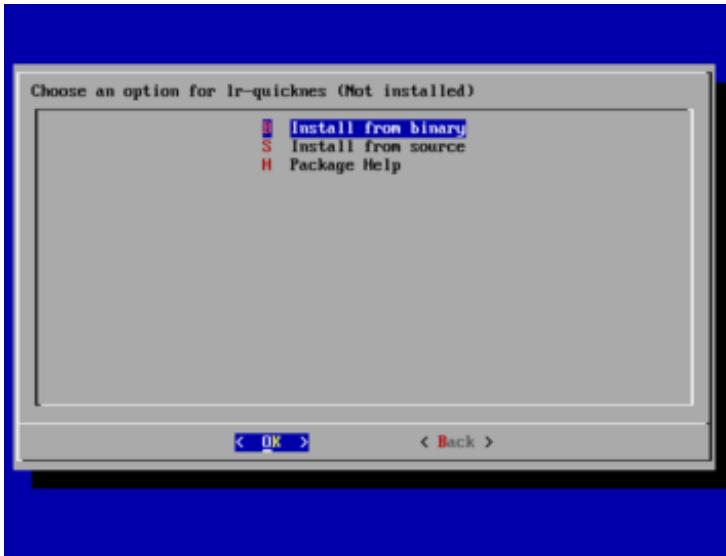
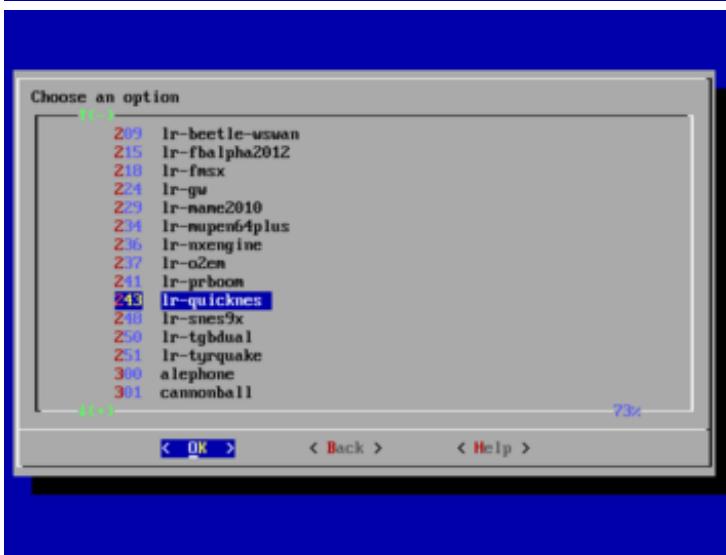
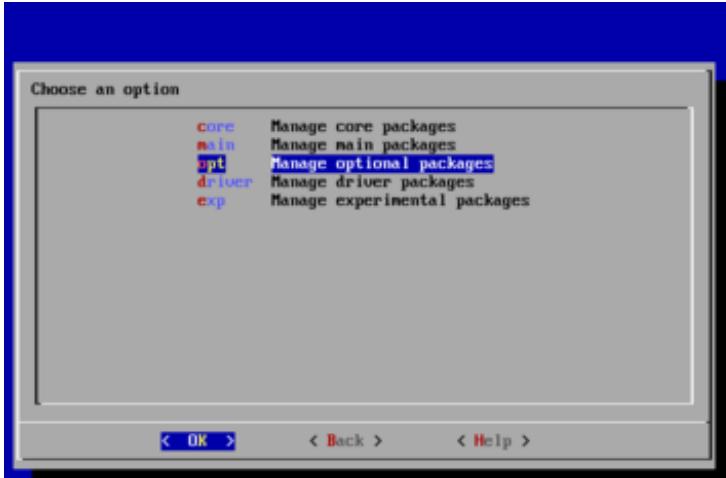
## Installing Additional Emulators



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...

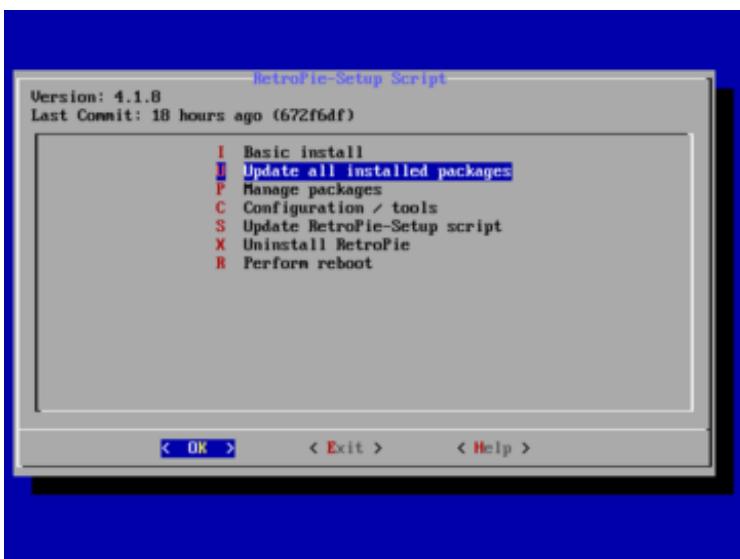
From the RetroPie configuration screen, select “RetroPie Setup.” This brings up the keyboard-driven menu again. Choose the “Update all installed packages” option to make a sweep through the system that’ll update *everything* to the latest release, including the kernel.



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



Even the default lr-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.