SquishBox Instructions

Assembly, Software Setup, and User Guide

Contents

This document will guide you through building, assembling, and setting up your SquishBox. The SquishBox is a stompbox enclosure for a Raspberry Pi 3, 4 or Pi Zero that provides a stompbutton, a rotary encoder, a high-quality sound card with 1/4" stereo, mono, and headphone outputs, and a bright and easy-to-read 16x2 character LCD display. The name means it's a stompbox that runs on software, so you can customize and change it to do whatever you need.

If you have purchased the SquishBox pre-assembled, you can skip to the “Installation” and “Software” sections for instructions on installing your Raspberry Pi computer in the SquishBox and setting up the necessary software to make it work.

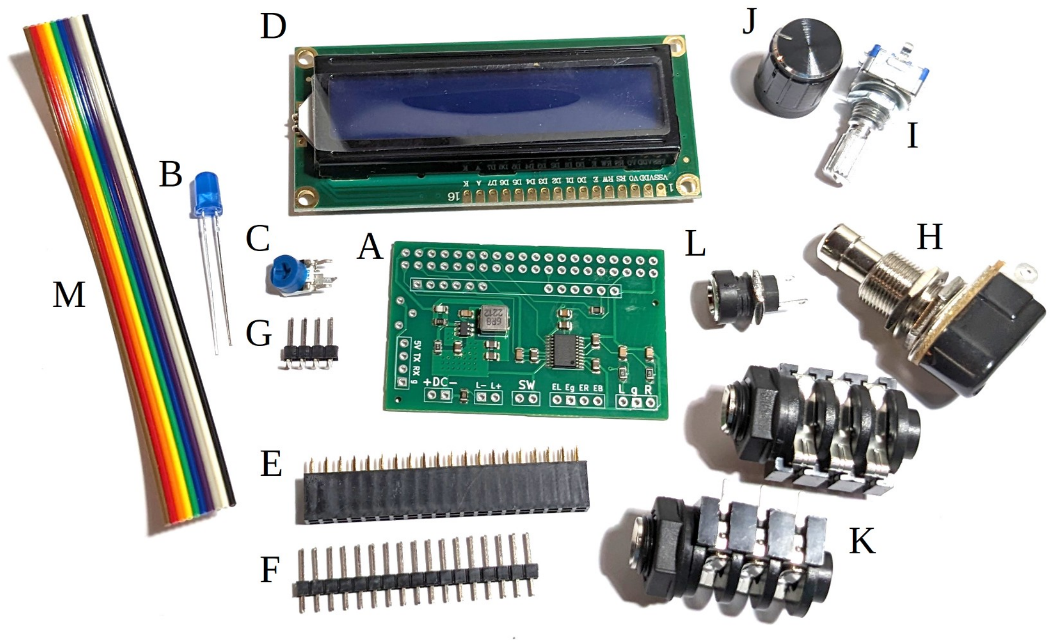
If you’ve purchased the kit, the “Components” section lists all the items you should have in your kit and “Tools” lists the tools you’ll need. The “Electronics” and “Hardware” sections describe how to solder everything together and how to mount everything in the standard 3D-printable enclosure, available from www.thingiverse.com/thing:5338973.

Once you’ve assembled your SquishBox and installed the software, you should be ready to plug in and play! If you need any further assistance visit geekfunklabs.com/support. We hope you enjoy making music with the SquishBox!

Components

The components needed to build the electronics for the SquishBox are listed below, and referenced in the image. You will also need a Raspberry Pi computer with an SD card to install software, and a power supply. The SquishBox and your Pi can be powered with a 5 Volt/2 Amp power supply via the 5.5mm × 2.1mm DC barrel jack, or you can use your existing Pi power supply plugged directly into your Pi's power port.

A. PCB with surface-mounted components
B. LED blue 5mm
C. 10K potentiometer
D. 16x2 character LCD
E. 2x20 female header
F. 1x12 male header
G. 1x4 right-angle male header
H. momentary stompswitch
I. rotary pushbutton encoder
J. decorative knob
K. 1/4" TRS audio jacks (2)
L. 5.5x2.1mm DC barrel jack
M. 26 AWG ribbon cable



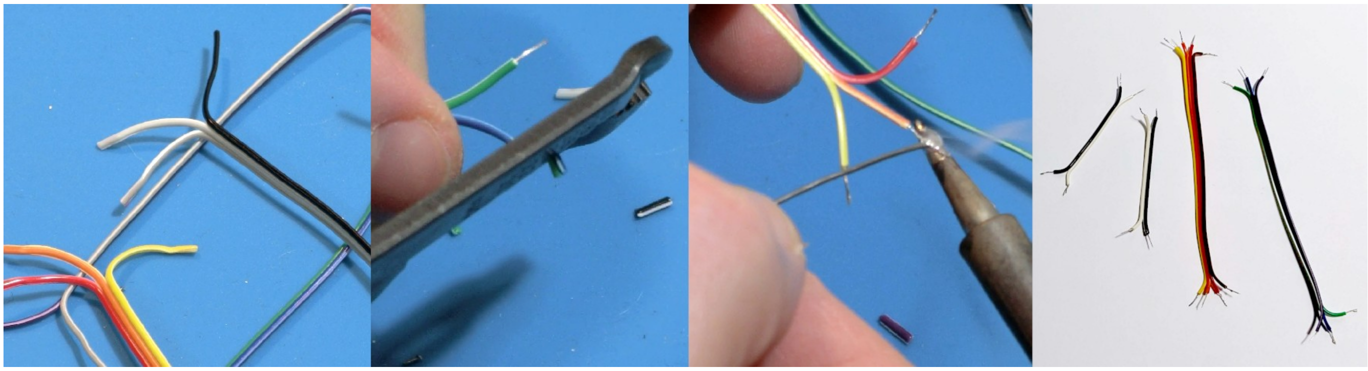
Tools

- Soldering Iron, Solder
- Flush Cutters
- Wire Strippers capable of stripping 26 AWG
- Needlenose pliers (optional but very helpful)
- A way to hold items steady while you solder:
  - vise
  - helping hands tool
  - (my favorite) a wad of poster tack/putty
- Small screwdrivers, cross- and flat-head

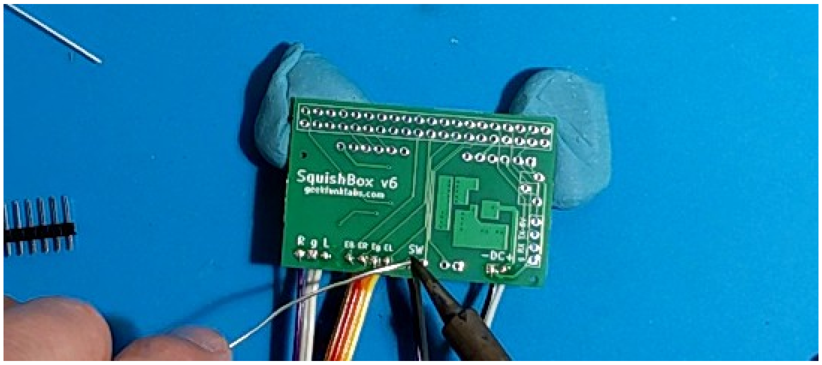
Electronics

In the instructions below, the “top” side of the PCB is considered to be the side with the surface-mounted components. Make sure to do the steps in order and solder the LCD and 2x20 header on last, otherwise you won’t be able to reach some of the components.

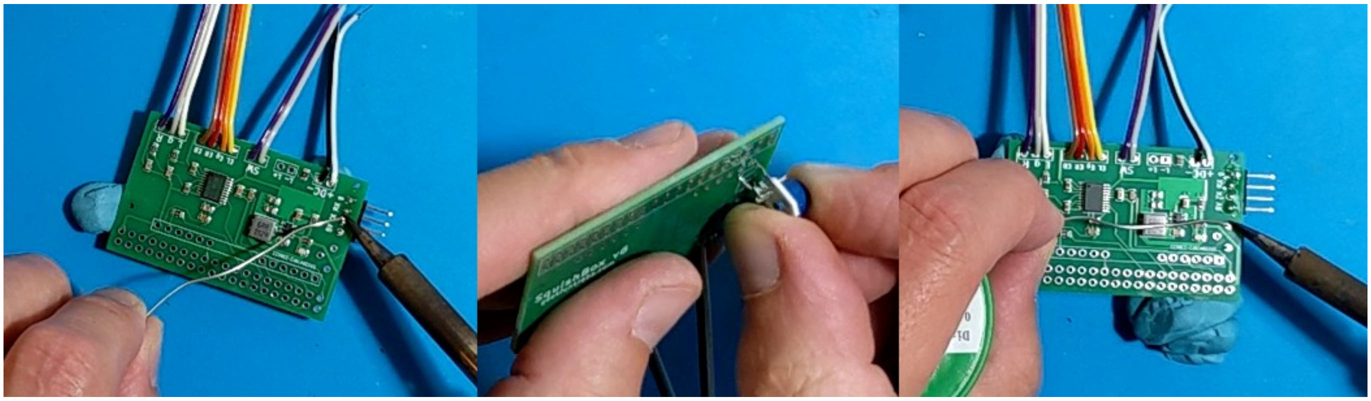
1. The button, encoder, and jacks are connected to the PCB using strips of multicolored ribbon cable, which helps to keep things organized and makes the connections easier to track. Slice off a strip of 4 wires, a strip of 3 wires, and a strip of 2 wires. Cut the strip of 2 wires halfway along its length. Separate the wires at the ends of each strip as shown, and strip, twist, and tin (add a thin coat of solder to) the ends. This makes them stronger and easier to solder in place.



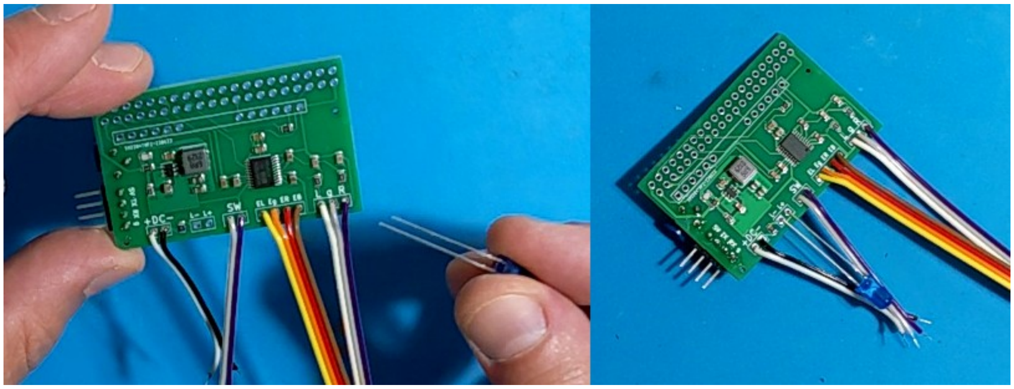
2. Insert your ribbon cable strips into the holes along the edge of the board as listed below. Bend the ends of the wires flat to hold them in place, then solder them and trim any excess away.
   - ○ A strip of two wires for the DC input (`+DC-`)
   - ○ A strip of two wires for the momentary button (`SW`)
   - ○ A strip of four wires for the rotary encoder (`EL Eg ER EB`)
   - ○ A strip of three wires for the audio output (`L g R`)



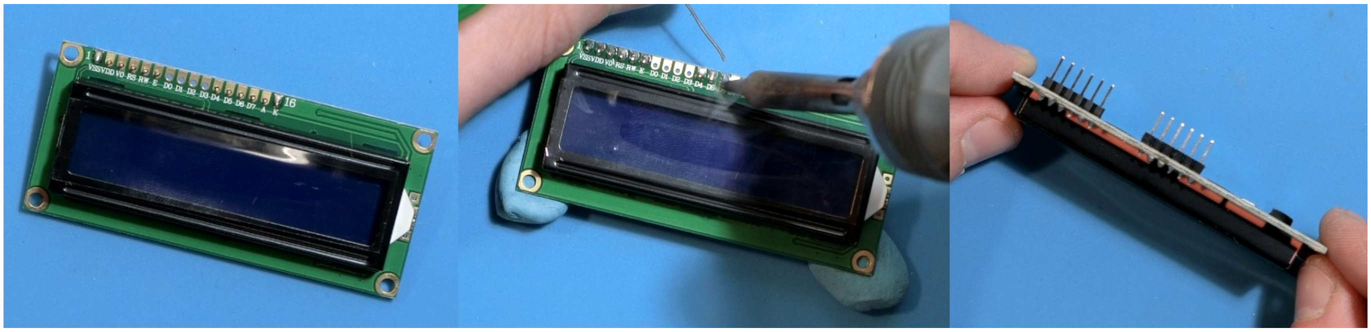
3. Insert the short ends of the right-angle header into the holes for the serial port (`5V TX RX g`) into the bottom side of the PCB as well and hold it in place with some putty. Solder one pin first and check that the long pins stick out parallel to the PCB before soldering the rest. Push the three legs of the 10K potentiometer through the PCB from the bottom side. The kinks in the legs require a bit of extra force, but once it pops through they will hold the component in place while you solder.



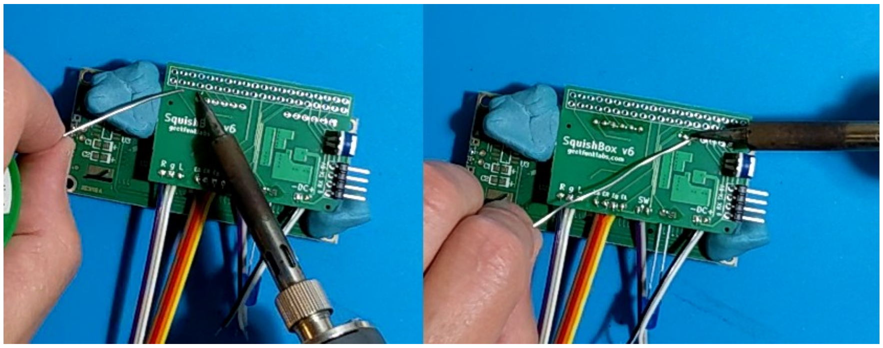
4. Insert just the ends of the LED into the holes marked `L- L+`. The longer leg is positive. Bend the legs so the LED is parallel to the plane of the PCB and solder the legs in place.



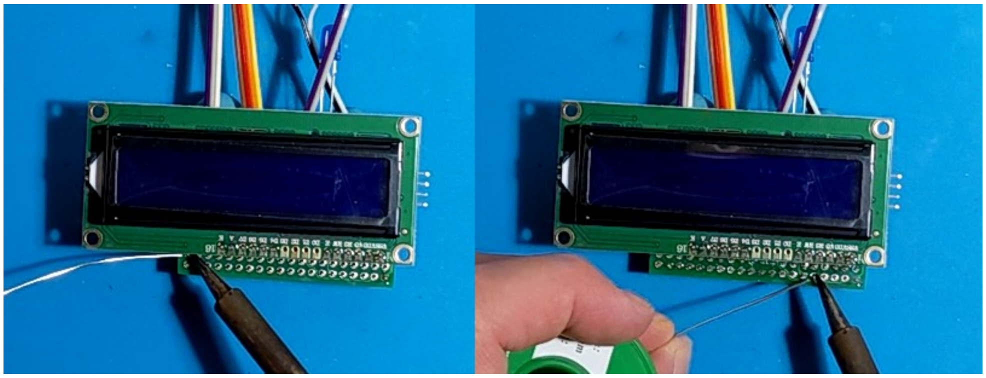
5. Break off two six-pin lengths of male header and solder them to the outside pins of the LCD as shown. The middle four pins `D0-D3` are not used. Start with one pin on each header, and make sure they are fully seated and the pins all stick up parallel to each other and perpendicular to the board before soldering the rest of the pins.



6. Insert the LCD into the top side of the PCB, but only push the pins through just enough so you can solder them to the bottom side of the PCB as shown. This allows the LCD to fit over a Raspberry Pi’s USB ports and Ethernet jack. Solder one pin on each end first and check the fit, then do the rest after re-melting and adjusting if necessary.



7. Insert the 2×20 female header into the underside of the PCB. As with the other headers, solder two pins at either end, then check that the socket is flush against the board so you can re-melt and adjust if necessary before proceeding to solder all 40 pins in place.



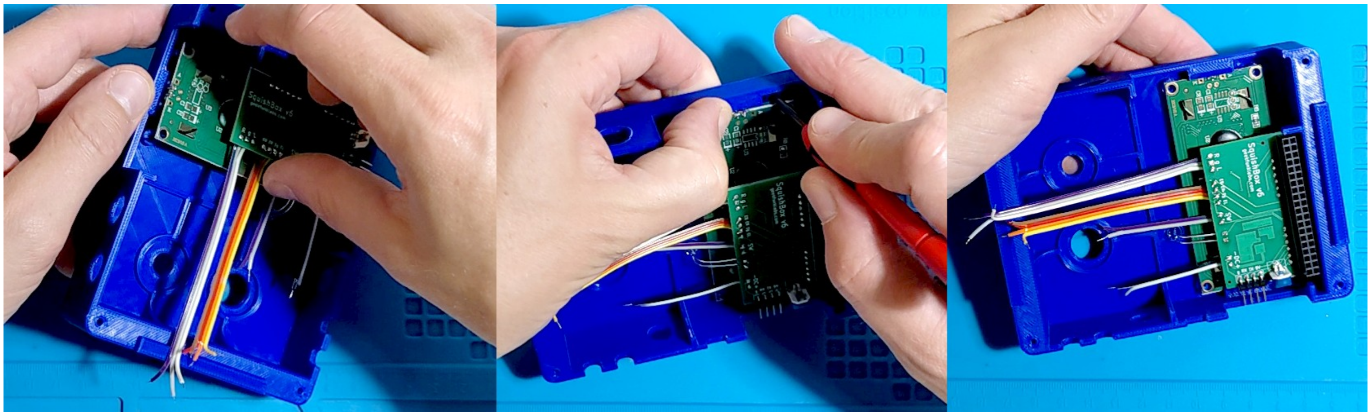
You have now soldered all the components that go into the PCB. You’ll solder all those floating wires to the button, encoder, and jacks after installing them in an enclosure. This makes it easier to keep them in one spot while you solder, and less likely you’ll rip wires when screwing things into the enclosure.

Hardware

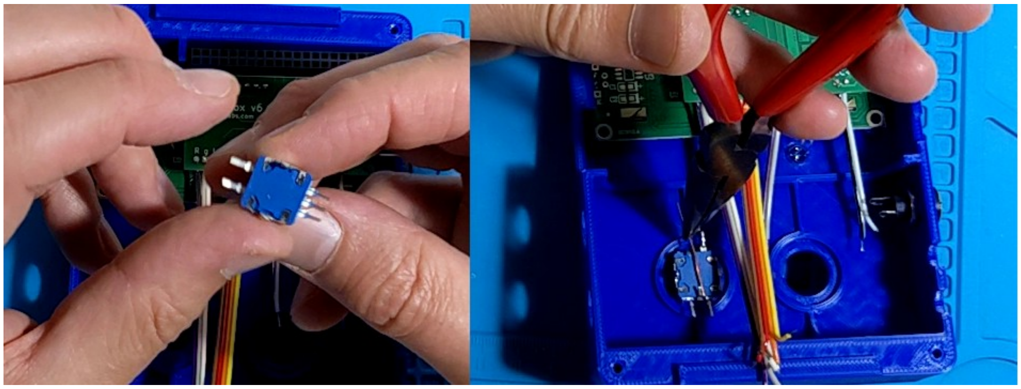
The steps in this section describe how to mount all the components in your enclosure and solder wires to the jacks, button, and encoder. You can 3D-print the official PLA case shown below by downloading the model files from thingiverse.com/thing:5338973. The lid screws on with four #4 size screws.

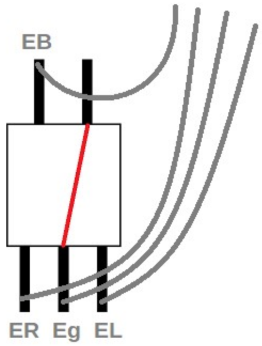
Similar to the last section, components can get in the way of your soldering if you install them in the wrong order, so you’ll have the easiest time if you do the steps as shown.

1. The enclosure has two clips that will hold the all the electronics in place. Viewing from bottom side of the PCB with the 2x20 header at the top, insert the right edge of the LCD in the opening and slide it underneath the right-hand clip. Bend the LED’s legs so it pokes through the hole in the enclosure. Now press down on the left side of the LCD with your thumb until it pops into place. It may help to use a screwdriver as a lever to flex the case a bit to get the LCD past the clip. If you mangle the clips doing this or just want some extra hold you can squirt some hot glue through the holes onto the mounting posts.

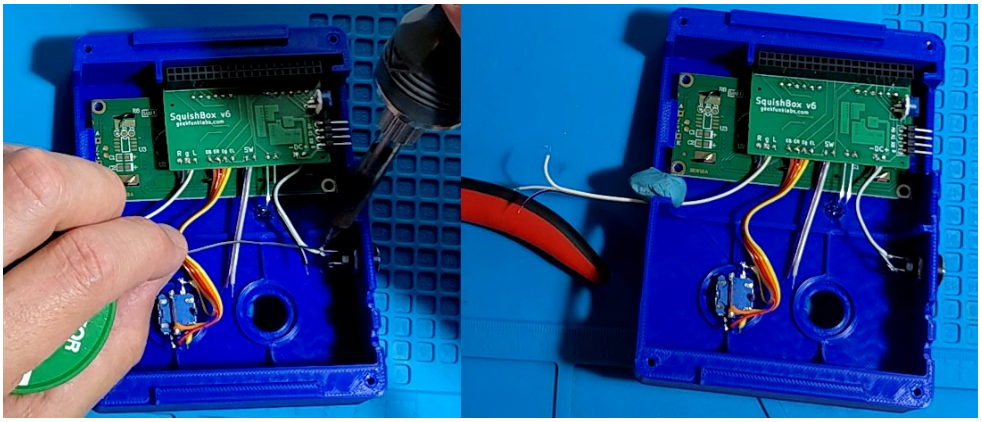


2. Bend the pins of the rotary encoder outwards and apply some solder to them. This makes it easier to solder the floating wires to them. Insert the encoder in the enclosure and secure it with the nut and washer. to them. Two of the pins need ground – solder a short wire between them as shown in the diagram. Connect the wires from the PCB to the encoder as shown in the diagram.

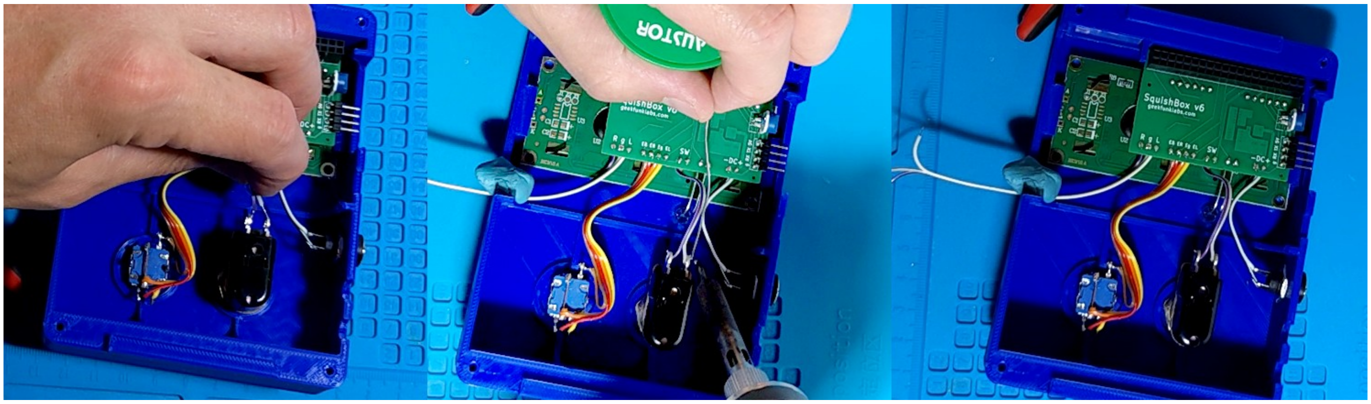




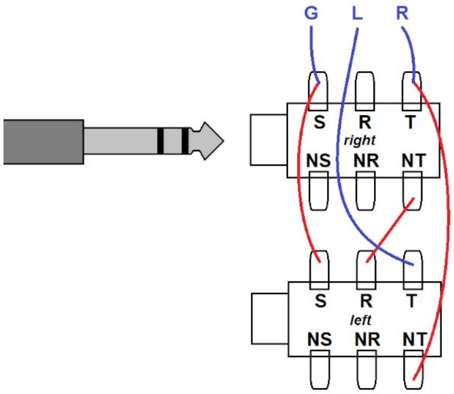
3. Next, install the DC jack. The nut for this component screws on from the inside of the enclosure, so you have to do it before you solder the wires on. The shorter lug connects to the center pin on your power supply, so if you are using a standard center-negative 9V effects pedal adapter you want to solder `DC-` to the short lug and `DC+` to the longer lug.

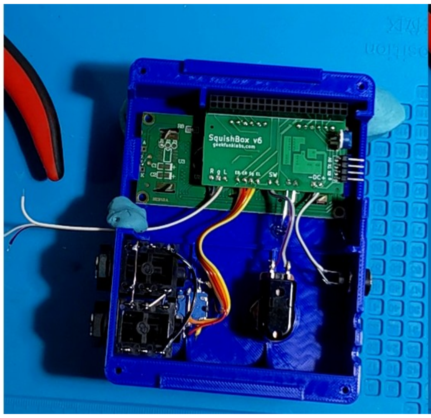


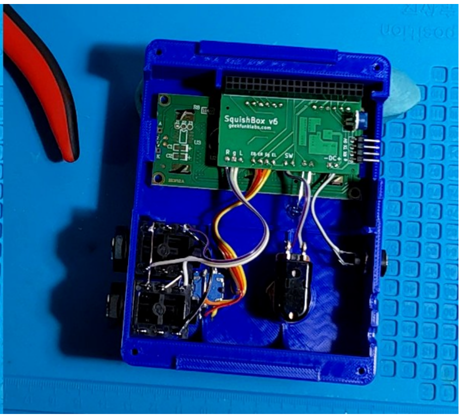
4. Screw the stompbutton into the enclosure, and solder the `SW` wires from the PCB to the stompbutton’s lugs – they can go either way. It can be helpful to bend the wire ends into hooks to get them to stay in the lugs while you solder.



5. Now wire up the audio jacks. Each jack has six pins, shown in the diagram below as they are laid out in the picture, which connect to the tip, ring, and sleeve of an audio jack. The “N” pins are normally connected, but when a plug is inserted the connections are broken, allowing audio to be routed depending on which jacks have plugs inserted. Apply some solder to the pins first. Use bits of wire or resistor legs to connect left `NT` to right `T`, left `R` to right `NT`, and left `S` to right `S` (as shown by the red lines in the diagram). Then connect the `R`, `g`, and `L` wires from the audio outputs on the PCB to the right jack’s `S` and `T` and the left jack’s `T`, respectively (shown by the blue lines in the diagram).



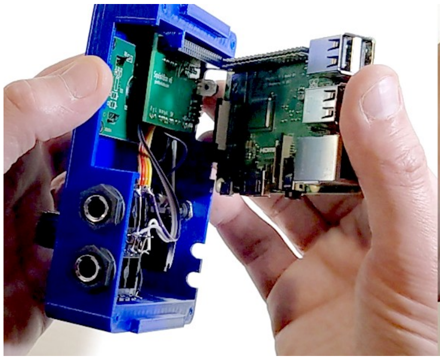
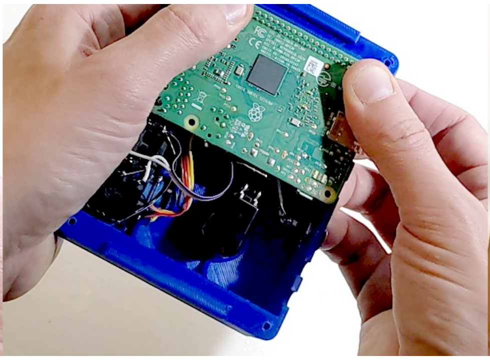
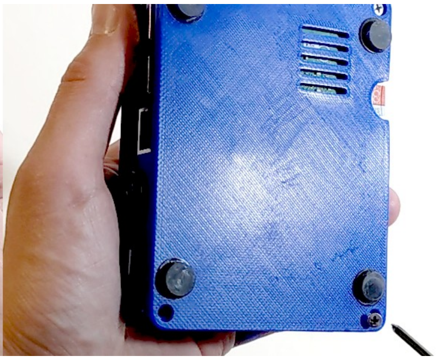




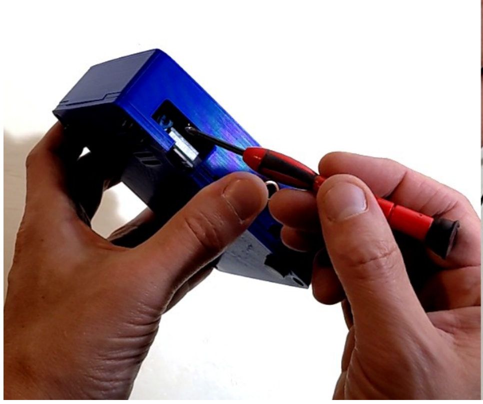
At this point you have finished assembling your SquishBox. You will install the Raspberry Pi and set up software in the next sections.

Installation

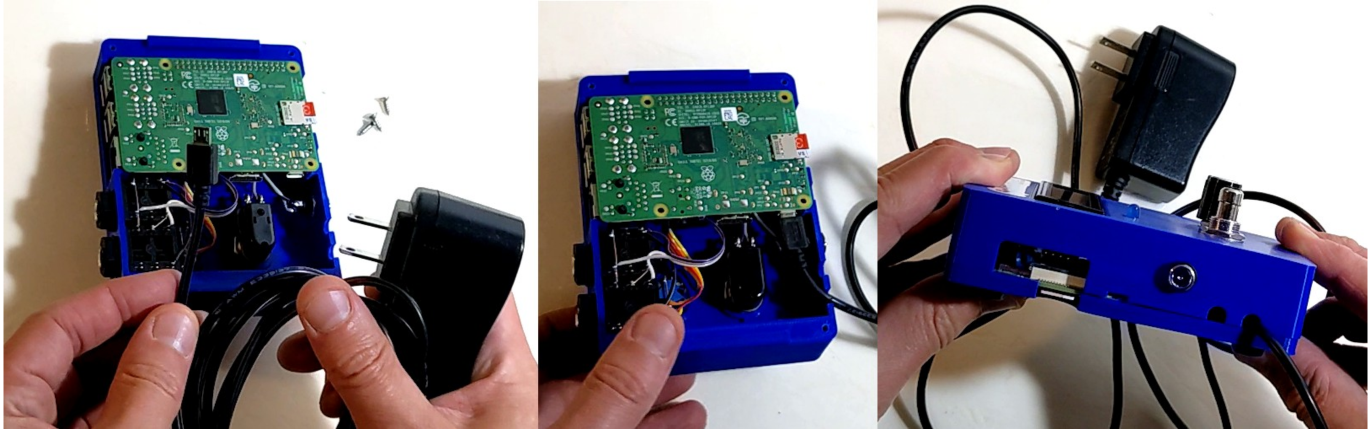
To install your Raspberry Pi into the SquishBox, just line up the pins on the Pi’s GPIO header with the SquishBox’s 2x20 female header and plug it in. Next, line up the posts on the lid with the mounting holes on the Pi and press it onto the SquishBox cover. The slots in the lid will snap on to the tabs in the cover. The lid will stay in place fairly well on its own, but a set of four #4 screws will make it tight and secure. Adhesive rubber bumpers on the base will keep the SquishBox in place when you’re using it.

You may need to adjust the LCD contrast to read the display clearly. Use a small screwdriver to adjust the 10k potentiometer until you get the desired contrast.



You may already have a 5V Raspberry Pi power supply and wish to use that to power the SquishBox rather than obtaining a new adapter. You can plug your 5V adapter into the Pi and thread its cable through one of the notches in the cover, as shown below. You can use these notches to connect other cables to the Pi as well, such as USB hubs or HDMI cables.



Software

The FluidPatcher software for the SquishBox turns it into a customizable MIDI sound module. The software is installed on Raspberry Pi OS by running a script from the command line. This downloads the code that runs the synth and a few other software dependencies, and sets up the interface to run on startup. These changes aren't drastic – you don't have to sacrifice your Raspberry Pi computer or buy a new one to use exclusively with the SquishBox! You can install the software on a working OS without wiping anything, and you can easily pop the Pi out of the enclosure and use it for something else as you need. You can view the source code for FluidPatcher at github.com/GeekFunkLabs/fluidpatcher and learn how it works if that interests you.

If you're using a brand new Raspberry Pi or just want to start fresh you can get OS images and instructions on installing at raspberrypi.org/software, and information on how to set up your Pi at raspberrypi.org/documentation.

Before running the install script, edit the */boot/config.txt* file on your Raspberry Pi, add the line below at the end to enable the sound card, and reboot the Pi.

```
dtoverlay=hifiberry-dac
```

Log in, make sure your Pi is connected to the internet, and enter the following command:

```
curl -L git.io/squishbox | bash
```

This will download and run an install script that will query you for options, then do all the configuration and software installation automatically. You can just press enter for each option to choose the defaults (enclosed in square brackets) to install everything, but here is an explanation of each of the options.

```
What are you setting up?
  1. SquishBox
  2. Headless Raspberry Pi Synth
```

This software can be installed on a bare Raspberry Pi without the SquishBox add-ons, but you should choose option 1 here. The script may need to reboot your Pi in order to set up the sound card, after which you should run the script again to continue.

```
What version of SquishBox hardware are you using?
v6 - Green PCB with SMT components
v4 - Purple PCB, has 2 resistors and LED
v3 - Purple PCB, has 1 resistor
v2 - Hackaday/perfboard build
```

Different versions of the PCB have slightly different wiring. The PCB used in these instructions is `v6`.

```
Enter Install location [/home/user]
```

The code installs in your home directory by default, but can be installed in any location where you have write privileges.

```
Install/update synthesizer software? ([y]/n)
```

This installs the base code that runs the SquishBox. You can also use this script to reconfigure some of the later optional extras, so if you want to do that without changing your code, reply no here.

```
Update/upgrade your operating system? ([y]/n)
```

Any time you install new software on the Raspberry Pi, it's a good idea to make sure your other software is all up to date, so you should probably say yes here.

```
Which audio output would you like to use?
 0. No change
 1. Default
 2. Headphones
 3. sndrpihifiberry
 4. vc4hdmi
Choose [3]
```

Choose `sndrpihifiberry` from the list.

```
Set up web-based file manager? ([y]/n)
  Please create a user name and password.
    username:
    password:
```

When the SquishBox is connected to a network, its web interface provides a convenient way of editing your patches, banks, and config files, as well as uploading soundfonts. Choose a good password to protect your SquishBox settings from other users on the same network. To log in to the file manager, connect a computer/tablet/phone to the same network as the SquishBox and point a web browser to the IP address of the SquishBox (see "WIFI Settings" below).

```
Download and install ~400MB of additional soundfonts? ([y]/n)
```

This downloads a collection of free but nice-sounding soundfonts that work in a variety of playing situations, plus an additional bank that uses them.

```
Option selection complete. Proceed with installation? ([y]/n)
This may take some time ... go make some coffee.
```

The terminal will start producing a bunch of output as it installs and configures the necessary software. If something does go wrong, this output can often be helpful in identifying the problem when seeking support, which can be found at geekfunklabs.com/support. Once finished, the software will ask if you would like to reboot. If you haven't installed the Pi in the SquishBox yet, you can reply no and then enter `sudo poweroff` to safely shut down.

If you want to use the Pi for something else, you can enter `sudo systemctl disable squishbox` at the command line to stop the synth from running on startup. If needed later, you can enter `sudo systemctl enable squishbox` to get it back again.

When you plug in the SquishBox, the Pi will boot and start the synthesizer software. The FluidPatcher version is displayed while the last-used bank loads. The current patch name, number, and total patches available are displayed on the LCD. Rotating the encoder cycles through patches. The encoder can also be tapped to advance to the next patch. The stompbutton sends MIDI messages that can be routed in banks or patches to act as a pedal, effects control, or perform other actions. The messages sent are control change 30 with a value of 127 and 0 on press and release, and control change 31 toggling between 0 and 127 with each press.

Holding down the rotary encoder for one second opens the menu. In menus the stompbutton does not send MIDI messages. Instead, rotating the encoder scrolls through options, or tapping the encoder advances to the next option and tapping the stompbutton goes back. This makes it easier to use the SquishBox with feet if it's placed on the floor. Holding the encoder for one second selects options, and holding the stompbutton for one second cancels or exits. Most menus will time out after a few seconds with no input.

Some menus have specific interaction modes:

- When asked to confirm a choice, it will be shown with a check mark or X next to it. Selecting the check mark confirms, X cancels.
- Some menus allow changing a numerical setting. Rotating the encoder adjusts the value, and holding the encoder confirms it.
- Some menus allow entering text character-by-character. The cursor appears as an underline for changing position and a blinking square for changing the current character. Holding the encoder switches between cursor modes. Holding the stompbutton exits editing, after which you will be asked to confirm or cancel your entry.

Below is a list of the menu options, with short descriptions of what they do.

- **Load Bank –** Load a bank file from the list of available banks. The current bank is displayed first.
- **Save Bank –** Save the current bank. Changing the name saves as a new bank.
- **Save Patch –** Saves the current state of the synthesizer (instrument settings, control change values) to the current patch. Modify the name to create a new patch.
- **Delete Patch –** Erases the current patch from the bank, after asking for confirmation.
- **Open Soundfont –** Opens a single soundfont and switches to playing sounds from the soundfont's presets instead of the patches in the current bank. Holding the encoder creates a new patch in the current bank that uses the selected preset on MIDI channel 1, after prompting you for a new for the new patch.
- **Effects.. –** Opens a menu that allows you to modify the settings of the chorus and reverb effects units, and the gain (maximum output volume) of the SquishBox. Changes affect all patches in the bank – save the bank to make them permanent.
- **System Menu.. –** Opens a menu with more system-related tasks:
    - **Power Down –** To protect the memory card of the SquishBox, this option should be used before unplugging. Allow 30 seconds for complete shutdown.

- **MIDI Devices –** This menu can be used to view the list of available MIDI devices, and to interconnect MIDI inputs and outputs. By default, the SquishBox automatically connects to all available MIDI devices, but this menu allows more control. It also includes a *MIDI Monitor* option that displays incoming MIDI messages on the screen. Pressing any button exits the MIDI monitor.
- **WIFI Settings –** Displays the current IP Address of the SquishBox, and provides a menu to scan for and connect to available WIFI networks. You can also enable/disable the wifi adapter here. It is useful to turn off the wifi adapter when you are out of range of any known networks, to keep the Pi from wasting CPU doing scans.
- **USB File Copy –** Allows you to copy your banks, soundfonts, and config files back and forth between the SquishBox and a USB storage device. Files are copied to/from a *SquishBox/* folder on the USB. The **Sync with USB** option will update the files to the newest available version on either device.

The SquishBox software and soundfonts collection include several banks with useful patches, and a large selection of soundfonts. However, a powerful feature of the SquishBox is the ability to configure it the way you need and create and your own patches. For information on how to edit the config and bank files for your squishbox refer to the README at:

github.com/GeekFunkLabs/fluidpatcher/blob/master/patcher/file_formats.md

There you can also find a link to a series of lesson videos on editing and creating patches, uploading new sounds, and configuring your SquishBox.

License