Raspberry Pi Guitar Amplifier

David Silverman
July 2022
https://github.com/dsagman/guitarpedal
https://www.youtube.com/watch?v=wl24fLW1fQE
https://docs.google.com/presentation/d/1GgiD2UTEGcz-SC2rGDjdIljaK2Yn0Rs03TDFOu3duUw/edit?usp=sharing
https://hackaday.io/project/186218-raspberry-pi-guitar-amplifier

Changelog

July 2022: added list of services to consider disabling

July 2022: updated ALSA warning message disabling

July 2022: added bill of materials

Comments and Questions

This technical document contains some of the information I learned (read soldered onto my hand and then bandaged either metaphorically or actually) over several months while trying to make a guitar amplifier and effects system using a single board computer and Python.

If you are game to try to do this yourself, I hope that my learnings will help you and reduce the amount of contact burns you experience.

If you have questions or comments, please use the hackaday page to contact me. And don’t hesitate! I probably screwed something up, and if you have a question, someone else likely does also.

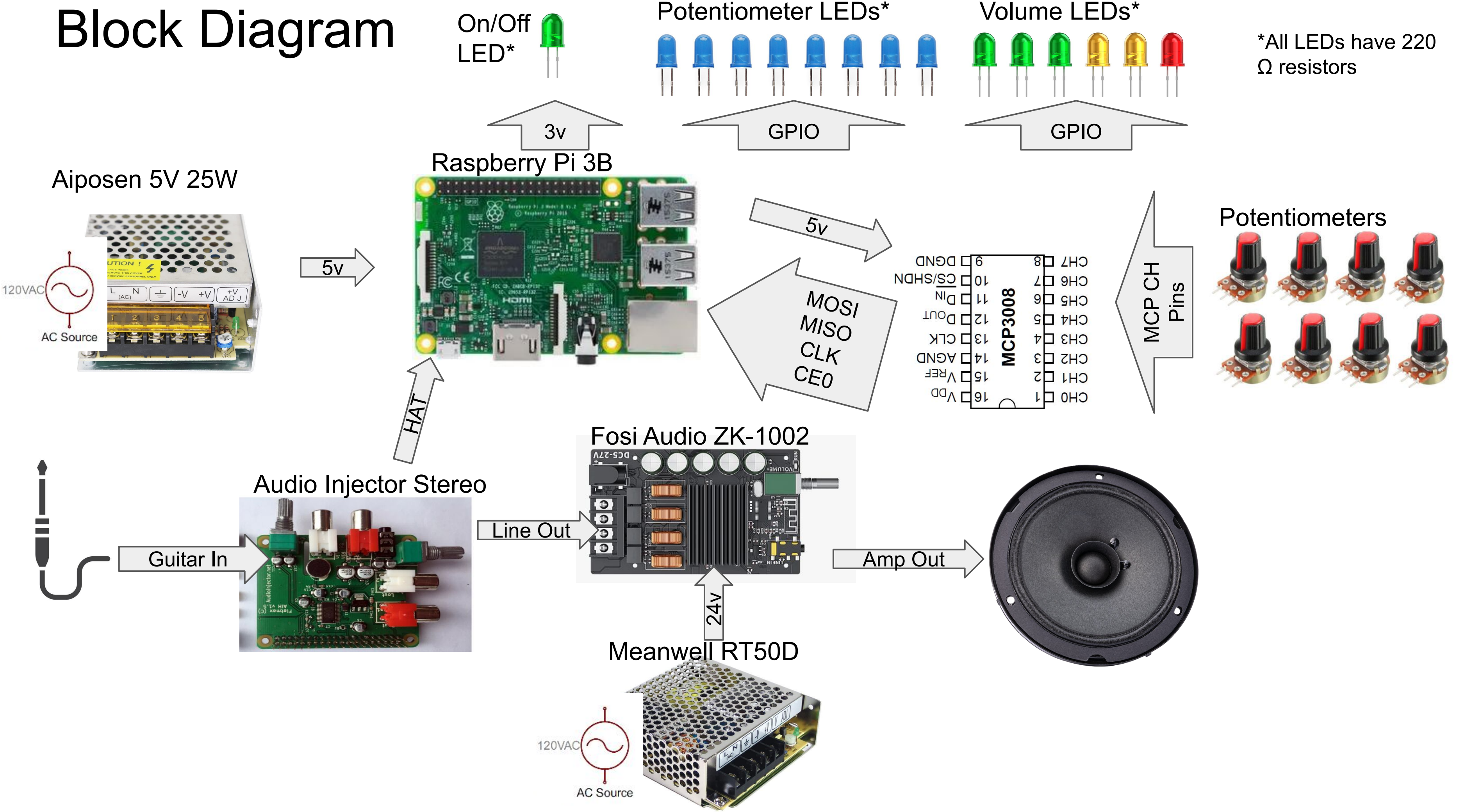
https://hackaday.io/project/186218-raspberry-pi-guitar-amplifier

Bill of Materials

I’ve listed the items I used. Feel free to substitute if you have something on hand or something you prefer.

- 1 cabinet, I used an old Vox cabinet
- 1 speaker, 8ohm or 4ohm
- 2 fanless power supplies
    - 1 24v for the amp, Aiposen 5V 25W
    - 1 5v for the PI, Meanwell RT50D
- 1 Raspberry PI 3B also works with 2 or 4
- 1 PI audio interface, Audio Injector Stereo
- 1 PI GPIO hat to make wiring easier, optional, I used one with screw terminals
- 1 Class D Amp, Fosi Audio ZK-1002
- 1 MCP3008
- 1 prototype board for mounting and connecting the MCP3008
- assorted plugboard screw terminals
- 9 50K potentiometers
- 15 Assorted LEDs
    - 8 blue
    - 4 green
    - 2 yellow
    - 1 red
- 15 220K resistors
- 1 guitar input jack
- 2 RCA jacks for connecting to and from the audio board
- 1 mini audio jack (for wiring to Fosi amp, may not need if you use different)
- 2 power switches for controlling main power and power to the PI
- Assorted wire and zip ties to keep things tidy

Block Diagram



*All LEDs have 220 Ω resistors

GPIO

On/Off LED*

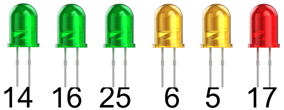
	Pin#	NAME	NAME	Pin#	
On/Off LED	01	3.3v DC Power	DC Power 5v	02	
Available	03	GPIO02 (SDA1 , I2C)	DC Power 5v	04	
Available	05	GPIO03 (SCL1 , I2C)	Ground	06	
P0	07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08	V0
	09	Ground	(RXD0) GPIO15	10	Available
V5	11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12	IJ Bit Clock
P1	13	GPIO27 (GPIO_GEN2)	Ground	14	
P5	15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16	P4
	17	3.3v DC Power	(GPIO_GEN5) GPIO24	18	P7
MCP3008 pin 11	19	GPIO10 (SPI_MOSI)	Ground	20	
MCP3008 pin 12	21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22	V2
MCP3008 pin 13	23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24	MCP3008 Pin 10
	25	Ground	(SPI_CE1_N) GPIO07	26	Not Available
	27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28	
V4	29	GPIO05	Ground	30	
V3	31	GPIO06	GPIO12	32	P6
P2	33	GPIO13	Ground	34	
IJ LR Clock	35	GPIO19	GPIO16	36	V1
P3	37	GPIO26	GPIO20	38	IJ Data In
	39	Ground	GPIO21	40	IJ Data Out

On/Off LED*

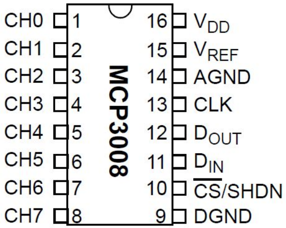
Potentiometer LEDs* (P)

4	27	13	26	23	22	12	24
P0	P1	P2	P3	P4	P5	P6	P7

Volume LEDs* (V)

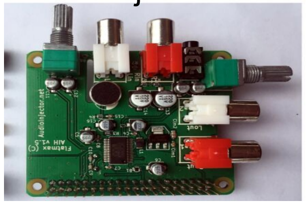


14	16	25	6	5	17
V0	V1	V2	V3	V4	V5



16: VDD -> +5V
15: VRef-> +5V
14: AGnd -> GND
13: CLK -> 11: SPI_CLK
12: Dout -> 9: SPI_MISO
11: Din -> 10: SPI_MOSI
10: CS -> 8: SPI_CE0_N
9: DGnd -> GND

Audio Injector Stereo (IJ)

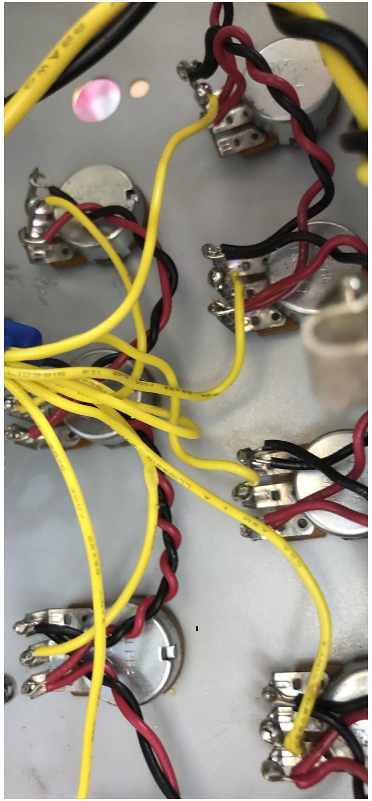


18 Bit clock
19 LR clock
20 Data In
21 Data Out

*All LEDs have 220 Ω resistors on + leg coming from the GPIO pin

Wiring Potentiometers

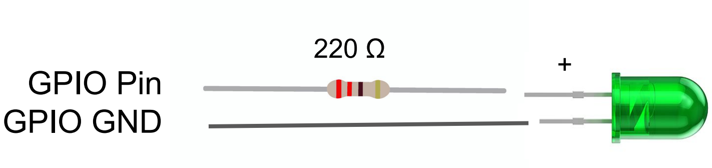
- Wire each potentiometer with one side to +5v, one side to GND, and the middle to a channel on the MCP3008. The pots can have the +5v and GND lines run in parallel across all of them.
- The MCP3008 is powered with +5v and connected to GND via the pi.
- You can use any of the +5v and GND pins on the pi for both the potentiometers and the MCP3008. You only need two pins. I just listed all for completeness.
  - +5v -> pin 2 or pin 4**
  - GND -> pin 9, 25, 39, 6, 14, 20, 30, 34**
- See photo for how to wire.
  - **Red** -> +5v
  - **Black** -> GND
  - **Yellow** -> GPIO, see previous page for pin numbers
- *NOTE: I wired all of my pots backwards. Ugh. So I changed the code to record 1.0-potvalue. If you wire the other way round, you will need to change the code.*



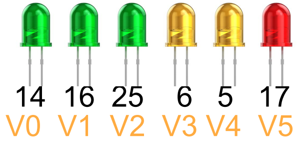
***That's pin, not GPIO number.*

LEDs and the Pi

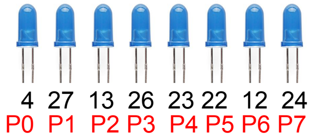
- All potentiometer indicator and VU (volume indicator) LEDs are wired with power coming a GPIO pin on the pi (see previous slide for details) and GND on the pi.
- GND -> pin 9, 25, 39, 6, 14, 20, 30, 34. (Just use one ground and connect all the LEDs negative sides together and then to one of the ground pins. I just listed all for completeness.)
- All LEDs have 220 Ω resistors on + leg coming from the GPIO pin.
- The power indicator LED is wired to a +3.3v GPIO pin on the pi, which is either pin 1 or pin 17.**
- **See page 4 for list of GPIO pins for the LEDs (v) and (P)**



Volume LEDs* (V)



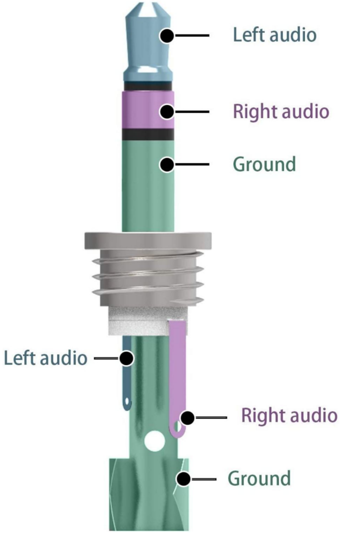
Potentiometer LEDs* (P)

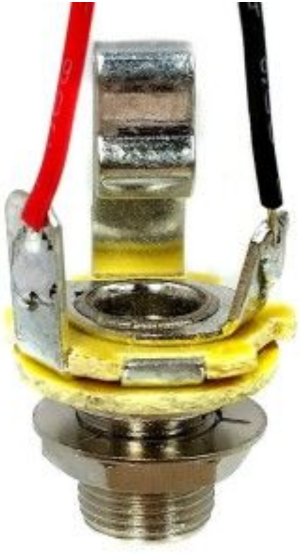


***That's pin, not GPIO number.*

Wiring Audio Jacks

References for wiring audio cables and jacks and not just because I wired everything backwards more than once.





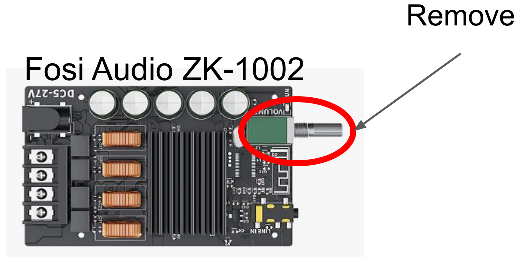
Tip
Primary
Audio

GND

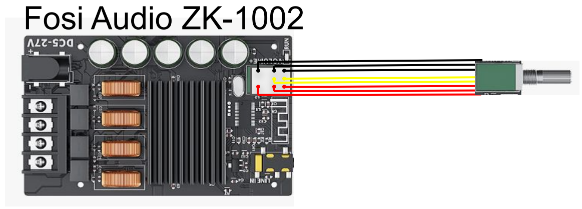
Adding Volume Potentiometer to Amp

- The class D amplifier I used, the Fosi Audio ZK-1002, has a volume potentiometer, but it is soldered to the board.
- Remove the potentiometer with hot air or soldering iron and add long wires to allow the pot to be mounted on a control panel.
- The original has two pots (left and right) and power switch. You can try to reuse the original, replace with the same, or just use a single pot, since you are only going to be controlling a mono source.
- Example new to buy: https://www.mouser.com/datasheet/2/54/PTR90-777814.pdf
- If replacing with a single pot, you will need to bridge the the two pins where power was switched by the original.

Before



After



Installing pyo Python Audio Library

- Installing pyo may require some additional Linux libraries, you can install them with:

  -
    ```
    sudo apt install libjack-jackd2-dev libportmidi-dev portaudio19-dev liblo-dev libsndfile-dev
    ```

  -
    ```
    sudo apt install python3-dev python3-tk python3-pil.imagetk python3-pip
    ```
- Then install pyo with pip:

  -
    ```
    python3 -m pip install pyo
    ```
- More information about pyo:
  - http://ajaxsoundstudio.com/pyodoc/index.html
- If you want to experiment with pyo, there are sample example files that come with the install, these are easily accessed from the `epyo` interactive development environment (IDE) application that is also installed automatically.
- Note: pyo does not work correctly with pypy compiled Python. However, in my testing pypy was less effective than regular Python.

Install Additional Python Libraries

- You need a couple of libraries, which can be installed with pip. (Instructions on the last page should have ensured pip is installed.)
- Optional: update all Python modules to latest version, but only for user:
  - `pip3 list --outdated --format=freeze | grep -v '^\-e' | cut -d = -f 1 | xargs -n1 pip3 install -U`
  - Don’t use sudo, otherwise you will be updating the installed modules directly, and that can cause problems.
  - If you are comfortable, you can set up a virtual environment for python, but since the raspberry pi is dedicated to this one application, I did not do that.
- `GPIOzero`
  - `pip3 install gpiozero`
- `Argparse`
  - `pip3 install argparse`

Potentiometers and the Pi

- The pi does not have analog inputs, only digital on the GPIO pins, therefore we need an ADC to convert analog resistance values from the potentiometers to allow variable control of the guitar effects. This is done with the MCP3008 IC.
- The GPIOzero Python module provides a software interface to the potentiometers via the MCP3008 IC. (We installed GPIOzero on the previous page.)
- More info on using GPIOzero with MCP3008 is here:
  - https://gpiozero.readthedocs.io/en/stable/recipes.html#potentiometer
- Enable the SPI interface on the pi to be able to use the MCP3008. You can enable SPI in two primary ways.
  - `sudo raspi-config`
  - Select interfacing options and then enable SPI.

  OR

  - `sudo nano /boot/config.txt`
  - Add the following line at the bottom :
  - `dtparam=spi=on`
- Detailed instructions are here https://www.raspberrypi-spy.co.uk/2014/08/enabling-the-spi-interface-on-the-raspberry-pi/

Installing Audio Injector Stereo Hat for Rasbian Bullseye **(post Jan 2022)**

- There’s lots of options for audio input and output for the pi, I chose the Audio Injector Stereo card. https://www.audioinjector.net/rpi-hat. Instructions for installing are here: https://github.com/Audio-Injector/stereo-and-zero. I’ve copied them below, but check there for the latest.
- Alter /boot/config.txt. The first thing to do is to update boot.txt using an editor, like nano :

  ```
  sudo nano /boot/config.txt
  ```

- Add this to the end of the file :

  ```
  dtoverlay=audioinjector-wm8731-audio
  ```

- Once that is done, reboot. You should be able to see your sound card :

  ```
  pi@raspberrypi:~ $ aplay -l
  **** List of PLAYBACK Hardware Devices ****
  [ SNIP ]
  card 2: audioinjectorpi [audioinjector-pi-soundcard], device 0: AudioInjector audio wm8731-hifi-0 [AudioInjector audio wm8731-hifi-0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
  ```

Continued: Installing Audio Injector Stereo Hat for Rasbian Bullseye **(post Jan 2022)**

- ****You may need to install ALSA utilities:****

  -
    ```
    sudo apt install alsa-utils
    ```
- Enable "Output Mixer HiFi"
  - If you want to do this using alsamixer, run that command from the terminal, press F6 on your keyboard and select the correct audioinjector soundcard. Next use the L+R arrow keys to navigate to the "Output Mixer HiFi" control and press "m" on the keyboard to change its state from mute ("MM") to on ("OO").
  - Another way to do this is to use the amixer command line :

    -
      ```
      amixer -D hw:CARD=audioinjectorpi set 'Output Mixer HiFi' unmute
      ```
- Setup pulseaudio to use the audio injector as output
  - Right click on the sound icon on the top right of the screen and in the outputs section, go to "Audio Outputs" and select the audio injector soundcard as the output.

Installing Audio Injector Stereo Hat for Rasbian Buster, Stretch, Jessie **(pre Jan 2022)**

- Older versions of Raspbian (including Buster, Stretch, and Jessie) and pulseaudio had issues with pulseaudio conflicting with the Audio Injector card.
- I recommend making a new SD card with the latest Raspbian rather than struggling with the older version.
- If you really don’t feel like doing that and want to use non-Bullseye Raspbian, refer to this page and follow the “old setup” procedure. https://github.com/Audio-Injector/stereo-and-zero

ALSA Warning Messages

- ALSA and JACK will issue a lot of warning messages when starting up. This is annoying, but does not cause any issues. Many of these are caused by unused audio interfaces. The remainder are caused by JACK, even though we aren't using JACK at all.
- If you are getting these errors when pyo starts up, and want to get rid of (at least some) of them, do the following.
- Edit the ALSA config file.
  - `sudo nano /usr/share/alsa/alsa.conf`
- Comment out lines that read by putting in a hash "#" at the start of lines that start with:
  - `front`
  - `rear`
  - `center_lfe`
  - `side`
  - `surround`
  - `iec958`
  - `modem`
  - `spdif`
  - `hdmi`
  - `phoneline`
- These occur at two places in the file. For me it was around line 95 and line 317 For example
  - `#defaults.pcm.front.card defaults.pcm.card`
  - `#defaults.pcm.front.device defaults.pcm.device`
  - …
  - `#pcm.front cards.pcm.front`

The Python Code

- All the code is on github, and since there’s a chance of updating the code, I didn’t copy it here: https://github.com/dsagman/guitarpedal
- However, it may help to know which effects I chose and how they are wired in.
  - Pot 0: Wet/Dry mix
  - Pot 1: Delay – The delay effect is OK. It delays the signal and repeats. This is built into pyo. You could vary the parameters to have more repeats or less.
  - Pot 2: Reverb – The reverb effect is very nice. Surprisingly nice. It’s the best one. This is also built into pyo. You could change some parameters here also. But why?
  - Pot 3: Distortion – The distortion overdrives the signal and slams the speaker cone all the way one way and then the other. It is therefore loud. It is fun. It is also built in to pyo. Amazing, right?
  - Pot 4: Wah. This is an autowah. It goes wah wah wah, automatically. I adapted code from the samples provided with pyo. So not built in, but damn close.
  - Pot 5: Bass tone control. Cuts out or enhances the bass frequencies. You could change the cutoff range.
  - Pot 6: Mid tone control. Cuts out or enhances mid frequencies. Same deal.
  - Pot 7: High tone control. Same deal.

Getting It to Start Automatically

- The method I chose was to add a line to my shell config. I use the fish shell (https://fishshell.com/) and did the following:
    - `nano ~/.config/fish/config.fish`
    - Add the following line at the bottom:
    - `python3 ~/pedalmaster.py`
    - This assume that you have the Python program in your home director (the '~' means home directory). If you have put it somewhere else, modify as appropriate.
- If you don't use fish, the standard Raspbian shell is bash.
    - `nano ~/.bashrc`
    - Add, as above: `python3 ~/pedalmaster.py`
- There are several other ways to get an application to run automatically when raspbian starts up. I tried a few, including having the application as a service that autostarted. I couldn't get it to work though. If you want to go down the rabbit hole, here's a link: https://www.simplified.guide/linux/automatically-run-program-on-startup

Shutting it Down

- To turn off the pi in headless mode you can use SSH and the `shutdown` command.
- But when using as a guitar amp, that’s not really an option. So I just turn it off. I wired the a switch between the power supply and the pi. Everytime I do that, somewhere a kernel developer winces, but in general, life moves on.

Pi Considerations and Operating System Modifications

- I have successfully gotten the application working with a pi 2B and 3B. A 4 should work great, but requires cooling, and if active cooling is used, the fan would be noisy.
- The Raspbian OS is multi-tasking and not designed for prioritizing a single application in the way that a microcontroller or realtime OS system can.
- Adjust the CPU governor to prevent frequency throttling
  - `sudo sh -c "echo performance > cpu0/cpufreq/scaling_governor"`
- Disable desktop graphical user interface. **Be sure you are comfortable in a command line only interface before doing this!** You can always undo and go back using the same `raspi-config` command. But if you were born after 1995 chances are you not natively used to not being able to use a mouse.
  - `sudo raspi-config`
  - Configure system settings to boot to console
  - Detailed instructions here: https://www.wobblycogs.co.uk/raspberry-pi/2022/01/08/disable-the-raspberry-pi-desktop/

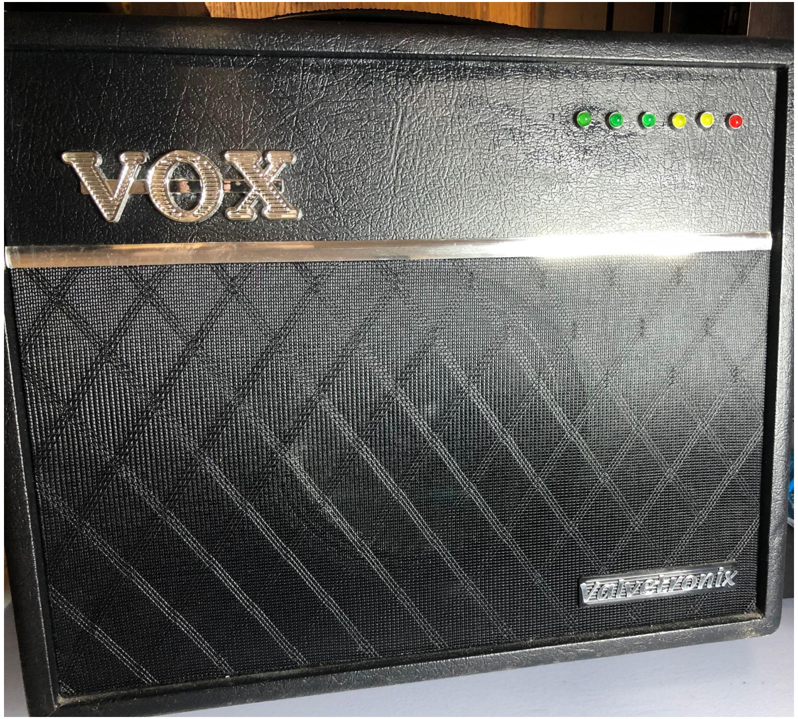
Disable Unneeded Services

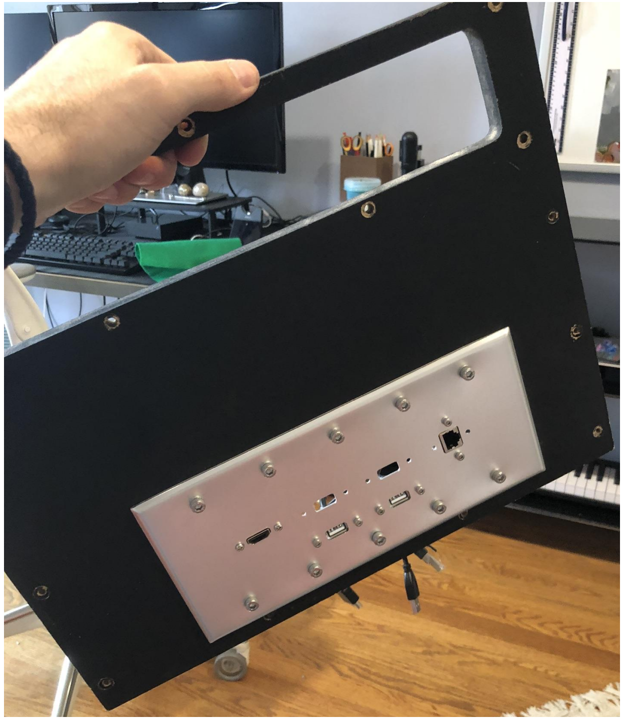
- Depending on which pi you are using, you will need to reduce competition for system resources possible more to prevent “clicking” caused by the audio processing stream dropping out due to tasks such as networking, printer subsystem, graphical user interface, etc.
- To remove unneeded daemons do the following. “xxx” is the name of the service/daemon you want to disable.
  - `sudo systemctl stop xxx.service`
  - `sudo systemctl disable xxx.service`
- For example, to stop cups, which is the printer daemon:
  - `sudo systemctl stop cups.service`
  - `sudo systemctl disable cups.service`
- To get a list of running services
  - `sudo service --status-all`
- **Alternative**: Use an application to manage services.
  - stacer, GUI based
    - `sudo apt install stacer`
  - rcconf, console based
    - `sudo apt install rcconf`

Services To Consider

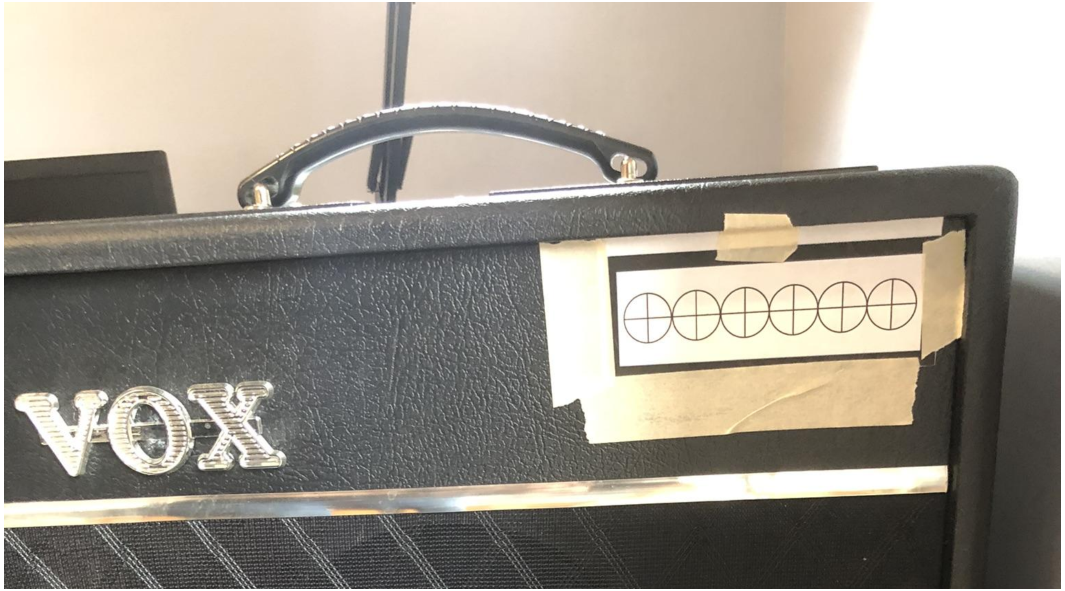
`cups`	printer
`cups-browsed`	printer
`bluetooth`	bluetooth
`raspi-config`	pi config, can still be run manually
`systemd-network-wait-online`	wait for network during boot
`dhcpd`	dhcp server
`dbus`	interapplication message bus
`packagekit`	software update
`pipewire`	audio thing you don’t need
`smbd`	network file sharing
`nmbd`	network file sharing
`colord`	color profiles
`hciuart`	bluetooth
`avahi`	network service discovery
`avahi-daemon`	network service discovery
`bluetooth`	bluetooth
`triggerhappy`	hotkeys
`xrdp`	remote desktop
`xrdp-sesman`	remote desktop
`bluealsa`	bluetooth music
`wpa_supplicant`	wifi* don’t disable if you want wifi
`vncserver-x11-serviced`	vnc remote desktop

Front and Back of the former Vox Valvetronix VT20+

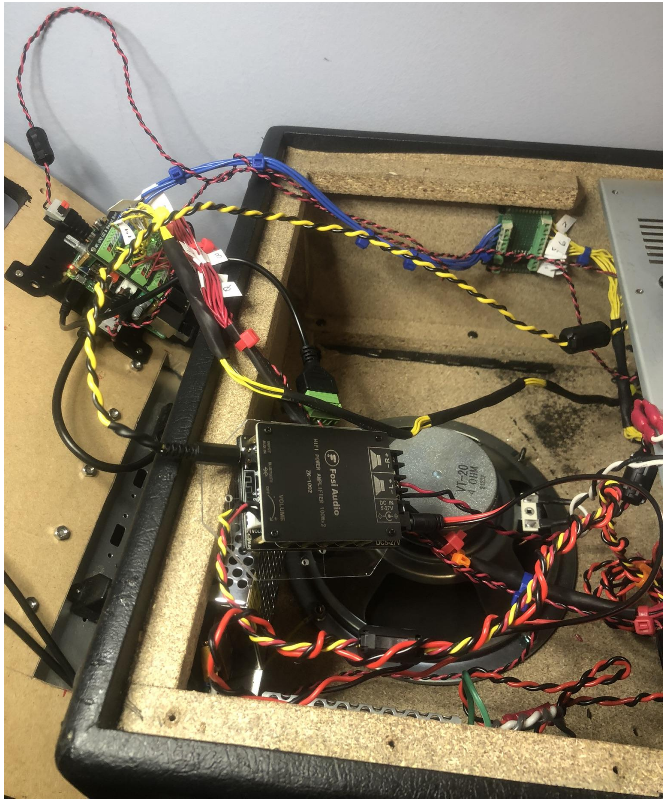


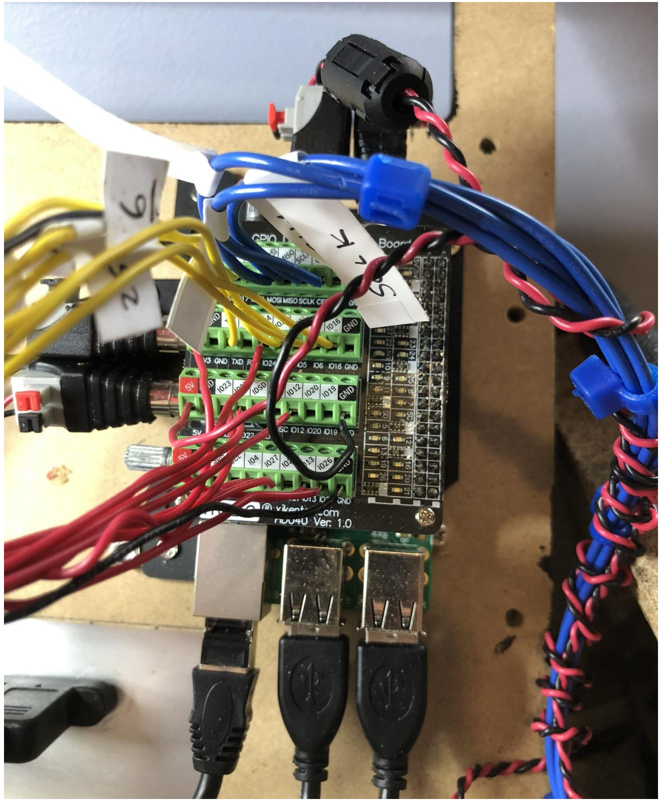


Photos: Stencil to lay out drill holes for VU LEDs

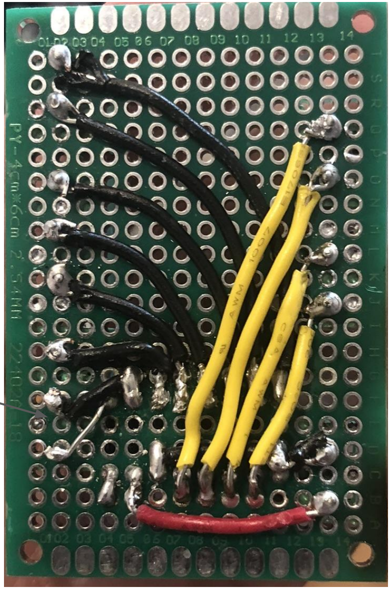


Photos: Innards

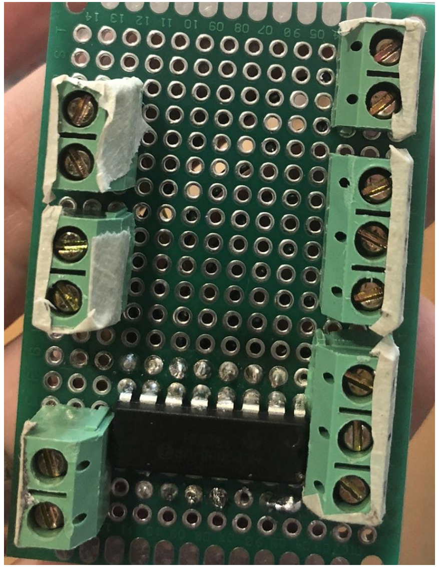




Photos: Handwired MCP3008 Board



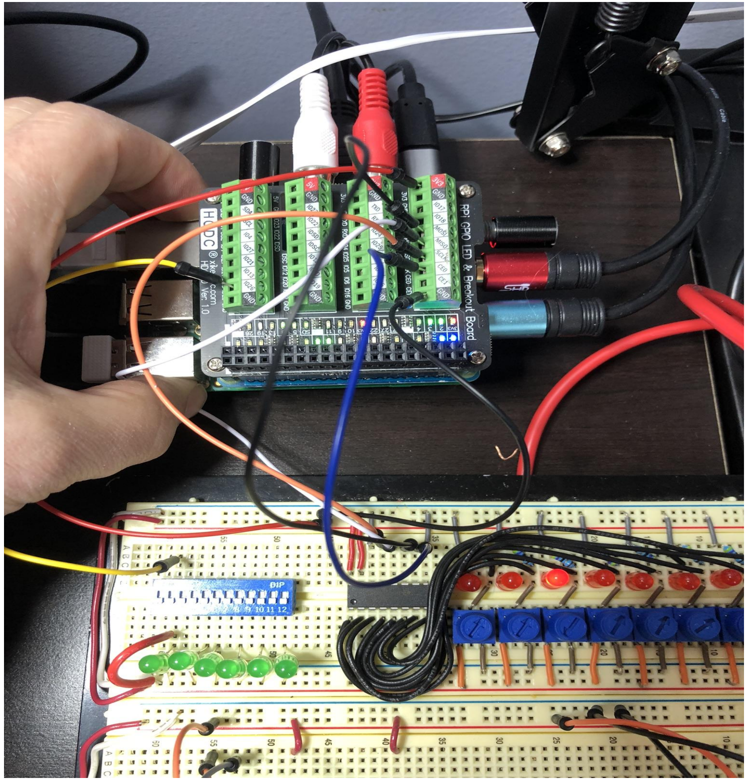
A bare wire, because I melted all the insulation….



Photos: Testing Setup

I breadboarded everything first. The blue pots and red LEDs are wired to the MCP3008.

The green LEDs and blue DIP switches were ultimately not used because I didn't use them.



Suggestions: SSH

- Use SSH to connect to the PI. Use raspi-config to set up. Detailed instructions here: https://phoenixnap.com/kb/enable-ssh-raspberry-pi#ftoc-heading-4
    - `sudo raspi-config`
- Samba/file sharing is useful for moving files back and forth, but it is a resource hog and can cause audible clicking. So I don't recommend it.
- To move files on and off the pi, instead of Samba use SCP. Instructions here: https://linuxhint.com/scp-remote-to-local/
- When using SSH, it will ask for a password every time. This is annoying. You can bypass by setting up SSH keys. Instructions here: https://danidudas.medium.com/how-to-connect-to-raspberry-pi-via-ssh-without-password-using-ssh-keys-3abd782688a
- I like the fish shell, https://fishshell.com/. The main thing I like is the `history` command to find previous things I forgot how to do. For example, `history systemctl` brings up all the commands I typed trying to figure out systemctl. You can install fish with `sudo apt install fish`.

Suggestions: Software

- I like the fish shell, https://fishshell.com/. The main thing I like is the `history` command to find previous things I forgot how to do. For example, `history systemctl` brings up all the commands I typed trying to figure out systemctl. You can install fish with `sudo apt install fish.`

Suggestions: Hardware

- Twist all pairs of wires together. This is for three reasons:.
  - Cable management: twisted wires are easier to keep together and figure out which goes with what.
  - Twisted wire reduces electromagnetic interference and noise. This is only an issue with analog signals, but because of cable management, I twisted all.
  - Looks cool.
- Heat shrink tubing to cover exposed bits of wire and solder. Less chance for short circuit.
- Don’t pick up the pointy end of the soldering iron.

Troubleshooting

- I spent a lot of hours. A lot. Many. Saying “WTFFityF? It was just working!!!”
- Diagnose component by component.
- For audio signals from the guitar and in and out of the audio injector and the class D amp a oscilloscope is handy. Since all you need to know is: is there a signal, and does it look like audio? you can get by with a cheap one. Search Amazon for “DIY Oscilloscope Kit” and get an assembled one. Prices as of today are around $50.
- For power connections a cheap multimeter is all you need.
- For other connections where you are just checking continuity that same multimeter is going to be perfect.
- If there’s no audio, first verify that you can get sound out of the pi. Connect a known working amp to the output of the audio injector. Try playing a file from the pi. You can do this with the `aplay` command. Download an mp3 file to the pi and try to play that.
- Try pulling the audio plugs halfway out or touching the middle connector of the male side to the outside of the female side. You may have wired the terminals backwards.
- Try direct connecting your phone or known good music player to the class D amp to see if that is the part that is failing.

Useful links

- **Use this to ask questions:** https://hackaday.io/project/186218-raspberry-pi-guitar-amplifier
- Github for python code: https://github.com/dsagman/guitarpedal
- Youtube demo video: https://www.youtube.com/watch?v=wI24fLW1fQE
- This document: https://docs.google.com/presentation/d/1GgiD2UTEGcz-SC2rGDjdIljaK2Yn0Rs03TDFOu3duUw/edit?usp=sharing
- Pyo python audio library: http://ajaxsoundstudio.com/pyodoc/index.html
- Audio Injector: https://www.audioinjector.net/rpi-hat. Instructions for installing are here: https://github.com/Audio-Injector/stereo-and-zero
- Datapro, custom plates for cables: https://www.datapro.net/cgi-bin/plate-designer-pro
- GPIOzero python library: https://gpiozero.readthedocs.io/en/stable/
- Enable the SPI interface on the pi to be able to use the MCP3008: https://www.raspberrypi-spy.co.uk/2014/08/enabling-the-spi-interface-on-the-raspberry-pi/
- Disable GUI: https://www.wobblycogs.co.uk/raspberry-pi/2022/01/08/disable-the-raspberry-pi-desktop/
- Setup SSH: https://phoenixnap.com/kb/enable-ssh-raspberry-pi#ftoc-heading-4
- SSH keygen: https://danidudas.medium.com/how-to-connect-to-raspberry-pi-via-ssh-without-password-using-ssh-keys-3abd782688a
- Fish shell: https://fishshell.com/.