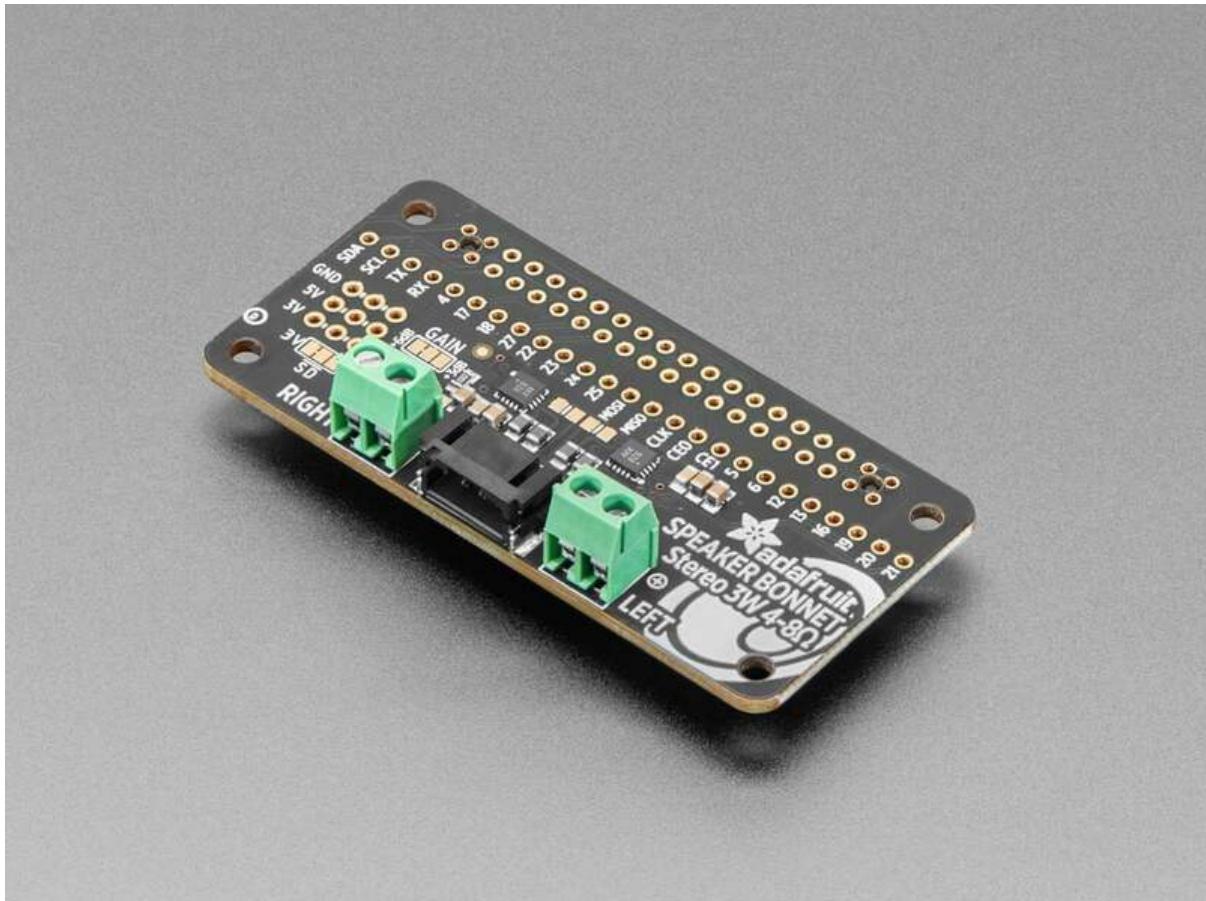




Adafruit Speaker Bonnet for Raspberry Pi

Created by lady ada



<https://learn.adafruit.com/adafruit-speaker-bonnet-for-raspberry-pi>

Last updated on 2024-06-03 02:02:57 PM EDT

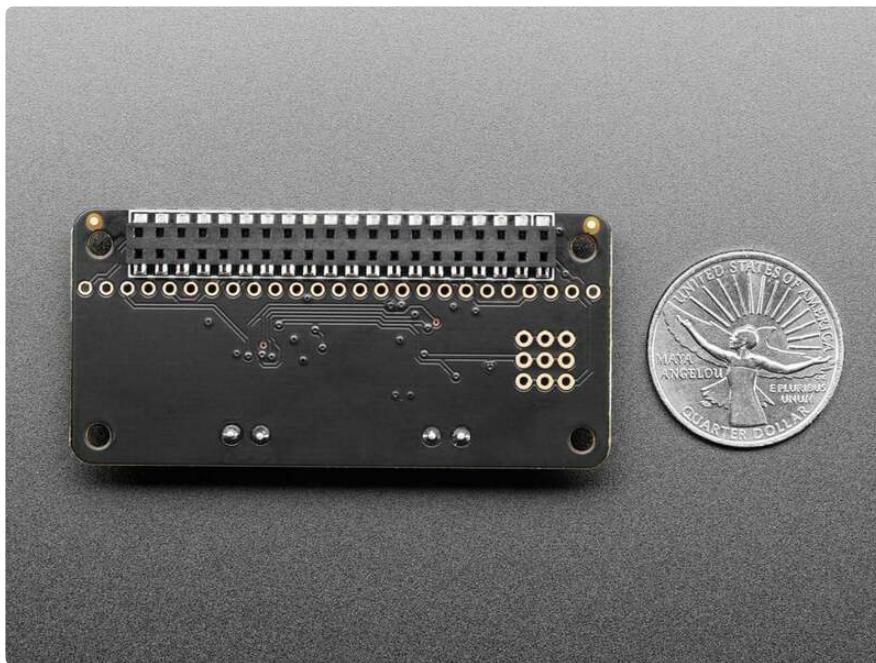
Table of Contents

Overview	3
Assembly	5
• And Solder!	
Pinouts	10
• Power Supply	
• I2S Audio Data Pins	
• Speaker Outputs	
Raspberry Pi Setup	14
• Setup Virtual Environment	
• Installer script	
• Detailed Install	
• Update /etc/modprobe.d (if it exists)	
• Disable headphone audio (if it's set)	
• Create alsound.conf file	
• Add Device Tree Overlay	
Raspberry Pi Test	21
• Speaker Tests!	
• Simple white noise speaker test	
• Simple WAV speaker test	
• Simple MP3 speaker test	
• Volume adjustment	
Pi I2S Tweaks	23
• Reducing popping	
• Step 1	
• Add software volume control	
I2S Audio FAQ	27
Play Audio with PyGame	29
• Install PyGame	
• Run Demo	
Downloads	31
• Datasheets & Files	
• Schematic	
• Fabrication Print	

Overview

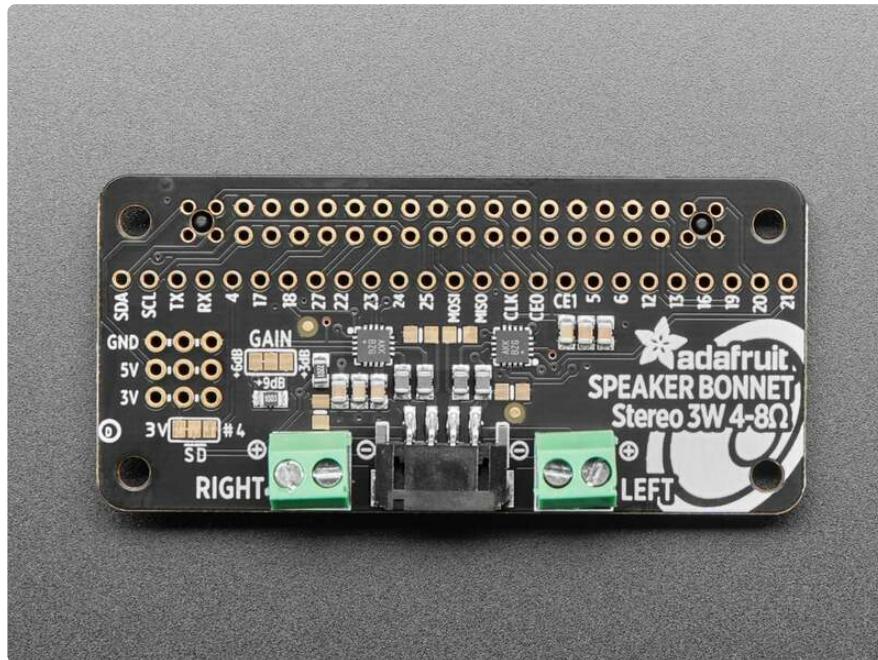


Hey Mr. DJ! Turn up that Raspberry Pi mix to the max with this cute 3W Stereo Amplifier Bonnet for Raspberry Pi. (It's not big enough to be an official HAT, so we called it a bonnet, you see?) It's the exact same size as a Raspberry Pi Zero but works with any and all Raspberry Pi computers with a 2x20 connector - A+, B+, Zero, Pi 2, Pi 3, etc. We've tested it out with Raspbian (the official operating system) and Retropie.



This Bonnet uses I2S a digital sound standard, so you get really crisp audio. The digital data goes right into the amplifier so there's no static like you hear from the headphone jack. And it's super easy to get started. Just plug in any 4 to 8 ohm

speakers, up to 3 Watts, run our installer script on any Raspberry Pi, reboot and you're ready to jam!



Each order comes as a fully assembled PCB with a 2x20 header and 2x terminal blocks. Once that's done [either plug one of our enclosed speaker sets](http://adafru.it/1669) (<http://adafru.it/1669>) right into the JST jack in the middle or you can solder the terminal blocks in and then connect any speaker you like.

Don't forget to make sure you have a good strong 5V power supply - especially if you're using the 3W 4 ohm speakers! [Our 2.4A power plug is recommended](http://adafru.it/1995) (<http://adafru.it/1995>)

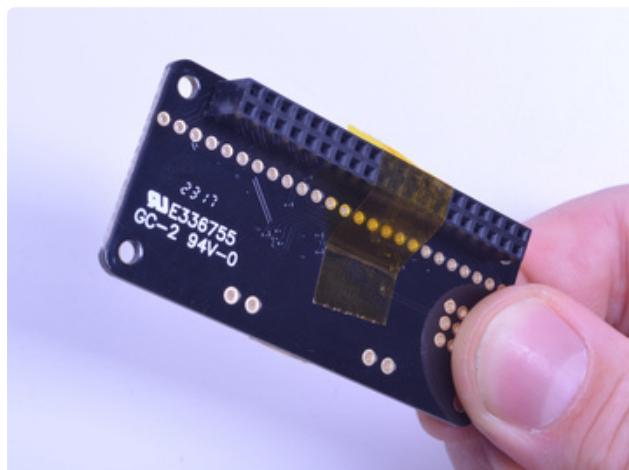
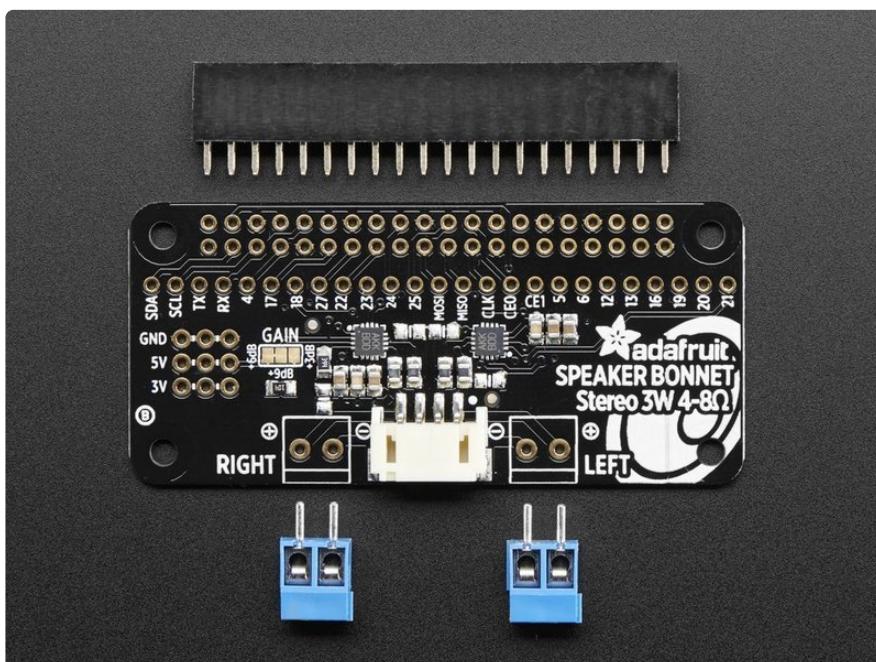


Revision History:

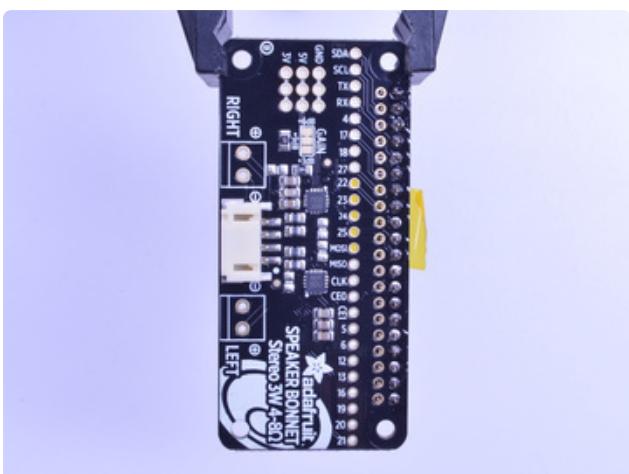
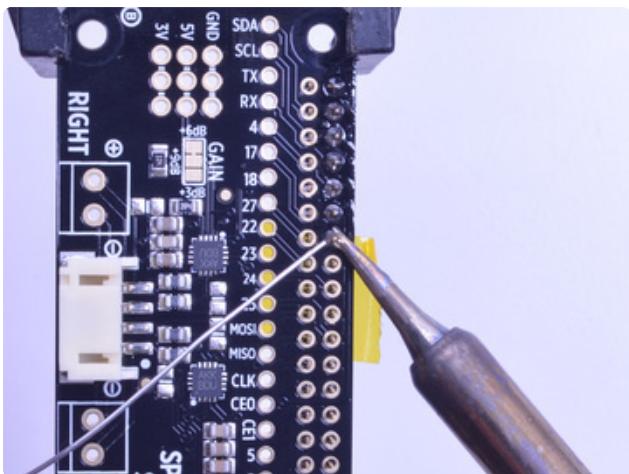
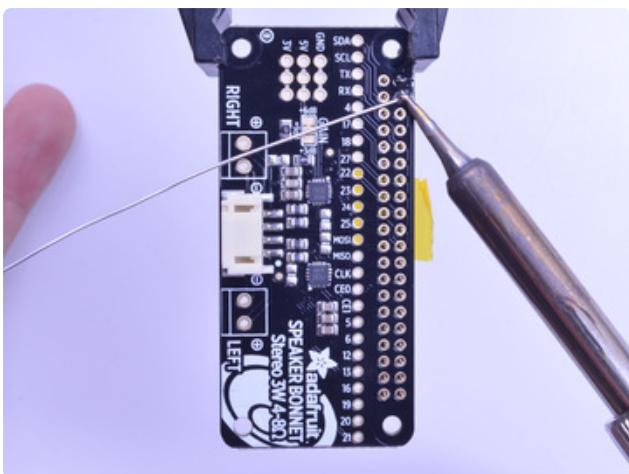
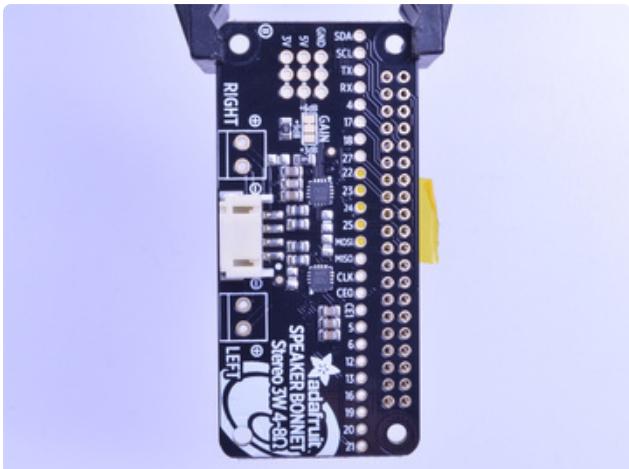
- **As of December 5, 2023** - This speaker bonnet now has the terminal blocks pre-soldered. We've also updated this PCB with [Adafruit Pinguin](https://adafru.it/11b7) (<https://adafru.it/11b7>) to make a lovely and legible silkscreen.
- **As of June 24, 2022** - We have added some breakout pads to the PCB to make configuration and muting easier for advanced users
- **As of March 16, 2018** This bonnet now comes with the 2x20 female header already soldered on

Assembly

As of December 5, 2023 - This speaker bonnet now has the terminal blocks and 40-pin header pre-soldered.



Place the 2x20 female header in so the connector is on the **bottom**. To make it easier to keep it in place, you can use some tape. Tacky clay also works, whatever you've got handy!

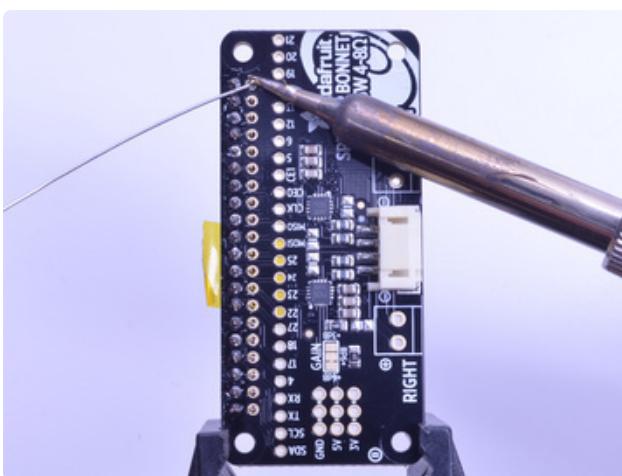


And Solder!

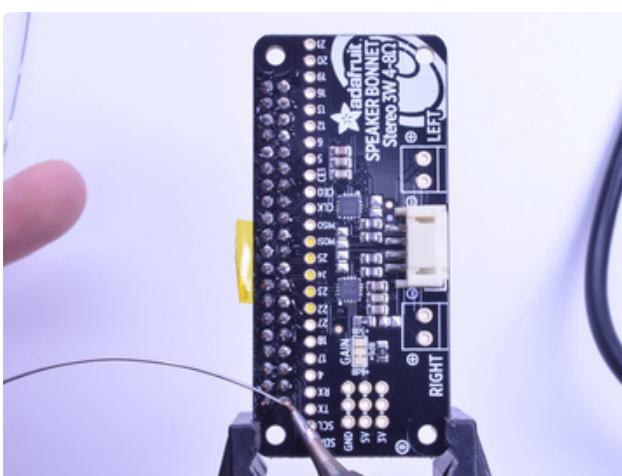
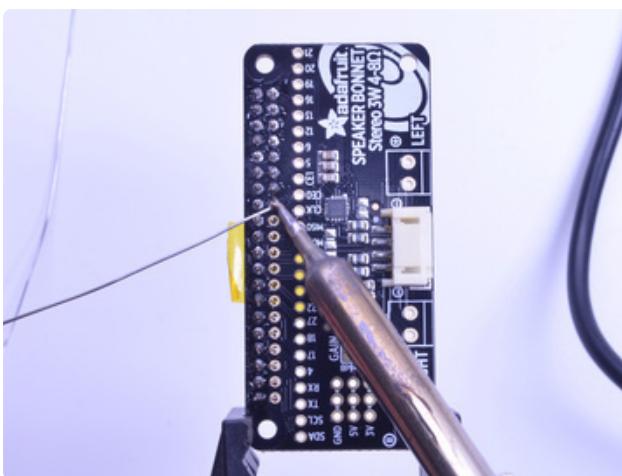
Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](https://adafru.it/aTk) (<https://adafru.it/aTk>)).

Start by soldering the first row of header



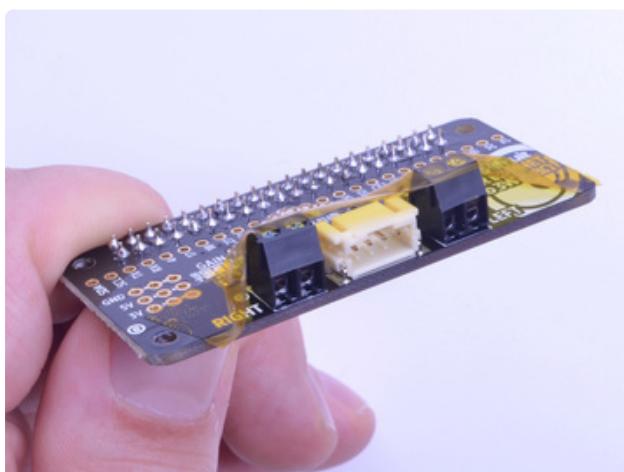
Now flip around and solder the other row completely





You're done with the header strips.

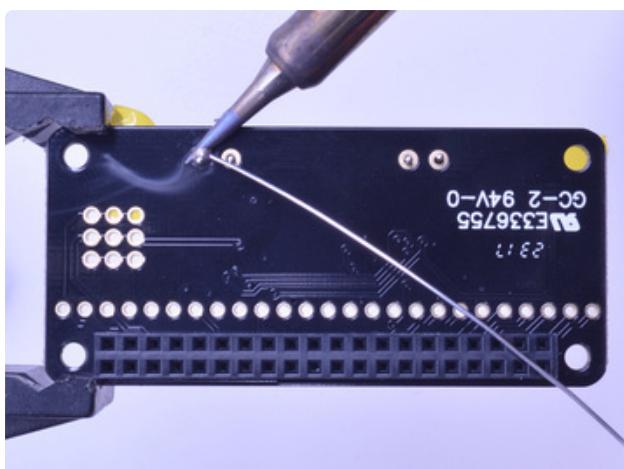
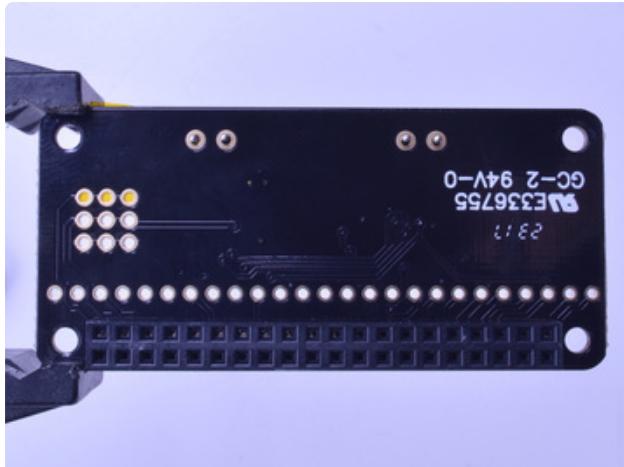
Check your solder joints visually and continue onto the next steps



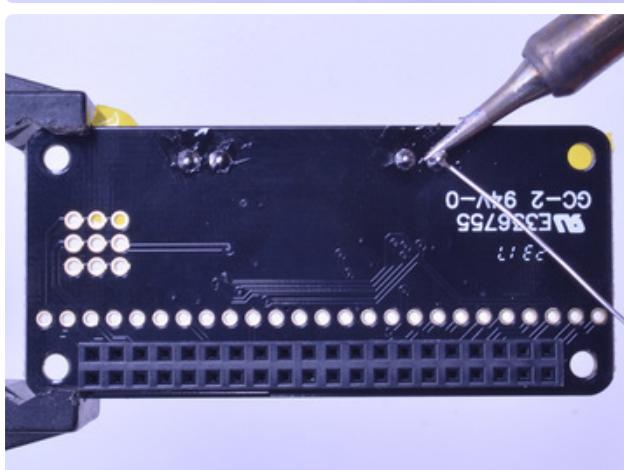
Next we will solder in the 2 3.5mm terminal blocks used to connect speakers to the Speaker Bonnet.

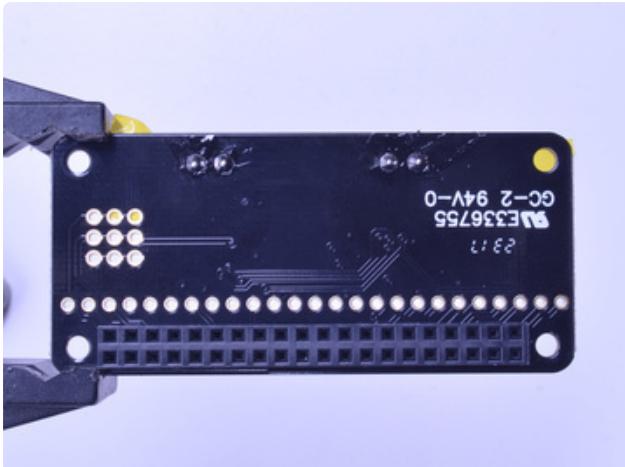
Make sure the open parts of the terminals face outwards so you can easily connect wires.

To make it easier to keep these in place, you can use some tape to hold down the two header pieces. Tacky clay also works, whatever you've got handy!

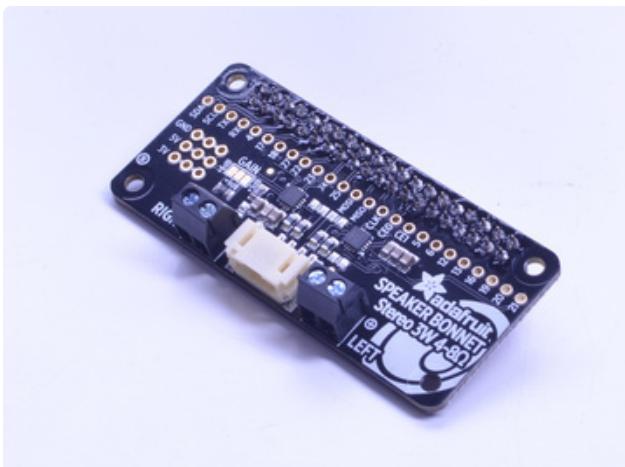


Solder in each block, make sure you get to each of the 4 pins





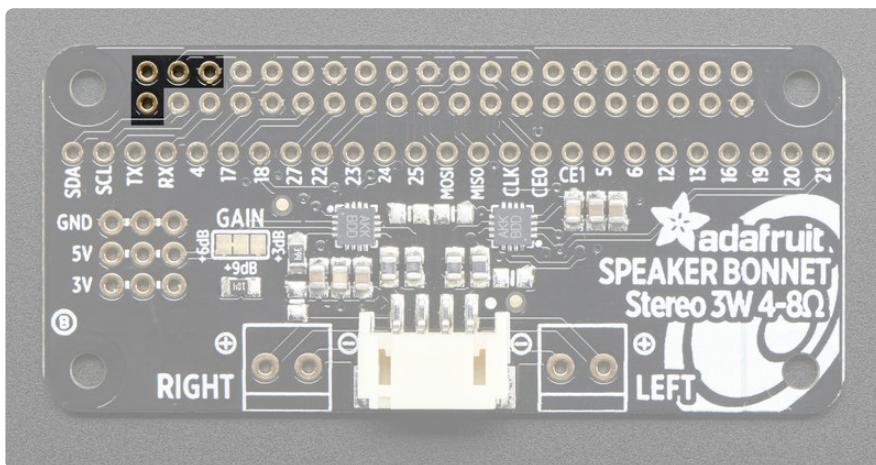
Now that you're done with the terminal blocks, check your work make sure that each solder joint is done and looks shiny



OK you're done!

Pinouts

Power Supply



The two amplifier chips use the 3V + 5V + GND power pin at the 'top' of the 2x20 header. If using 3W speakers, you can draw a significant amount of current, over 1.5

Amps! Make sure you power your Pi with a good wall adapter like our 2.4A microUSB power plug.



5V 2.5A Switching Power Supply with 20AWG MicroUSB Cable

Our all-in-one 5V 2.5 Amp + MicroUSB cable power adapter is the perfect choice for powering single-board computers like Raspberry Pi, BeagleBone, or anything else that's...

<https://www.adafruit.com/product/1995>

Alternatively, if you really need a lot of power, use a 5V 4A power adapter and then a DC to micro USB adapter



5V 4A (4000mA) switching power supply - UL Listed

Need a lot of 5V power? This switching supply gives a clean regulated 5V output at up to 4 Amps (4000mA). 110 or 240 input, so it works in any country. The plugs are "US..."

<https://www.adafruit.com/product/1466>

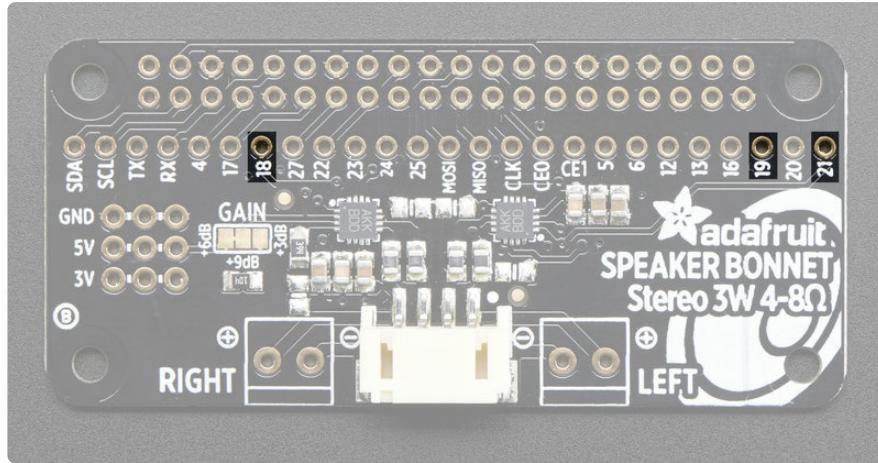


MicroUSB Plug to 5.5/2.1mm DC Barrel Jack Adapter

The premise of this MicroUSB Plug to 5.5/2.1mm Jack adapter is simple. There's a MicroUSB plug on one end and a 5.5mm/2.1mm DC barrel jack with center...

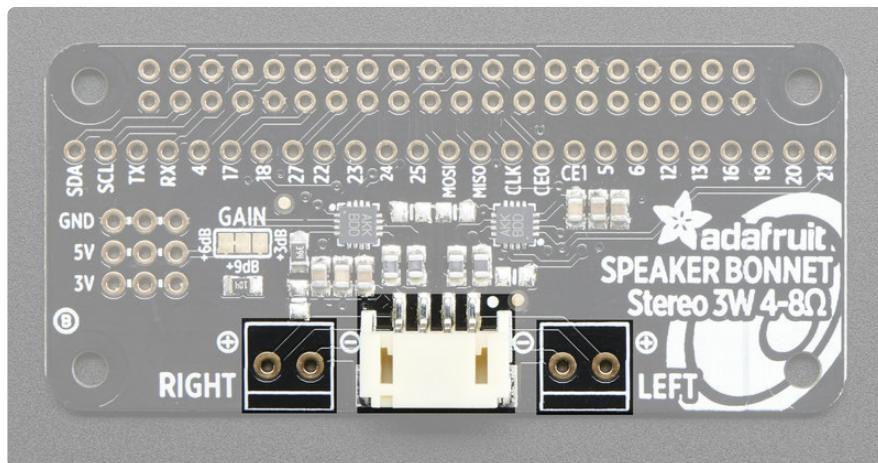
<https://www.adafruit.com/product/2727>

I2S Audio Data Pins



The Bonnet uses 3 data pins **and they cannot be changed!** Pins #18, #19 and #21 are used. All other pins are available

Speaker Outputs



This is the fun part, you get stereo output - one left and one right channel. Each one is Bridge-Tied-Load so **do not connect both outputs together** to get more volume!

You can use the JST 4-pin plug in the center if you're going to just plug in one of our enclosed speaker kits:



[Stereo Enclosed Speaker Set - 3W 4 Ohm](#)

Listen up! This set of two 2.8" x 1.2" speakers are the perfect addition to any audio project where you need 4 ohm impedance and 3W or less of power. We particularly like...

<https://www.adafruit.com/product/1669>

OR

You can use the terminal block spots on the left & right to connect regular speaker cones. You'll need to solder wires on and such but this way you can use whatever speaker you like! We suggest 8 ohm 1W or 4 ohm 3W. For louder audio, but more power usage, use the 4 ohm speakers. For quieter audio, but less power usage, use the 8 ohm speakers



[Speaker - 3" Diameter - 4 Ohm 3 Watt](#)

Listen up! This 3" diameter speaker cone is the perfect addition to any audio project where you need an 4 ohm impedance and 3W or less of power. We particularly like this cone as...

<https://www.adafruit.com/product/1314>



[Speaker - 3" Diameter - 8 Ohm 1 Watt](#)

Listen up! This 3" diameter speaker cone is the perfect addition to any audio project where you need an 8 ohm impedance and 1W or less of power. We particularly like this cone as...

<https://www.adafruit.com/product/1313>

You can also use other 'audio' devices like sonic transducers! These are devices you can put down onto a surface like a table, to make it into a speaker.



Medium Surface Transducer with Wires - 4 Ohm 3 Watt

Turn any surface/wall/table etc into a speaker with a surface transducer. This type of speaker does not have a moving cone like most speakers you've seen. Instead, a small metal...

<https://www.adafruit.com/product/1785>

Raspberry Pi Setup

At this time, Raspberry Pi linux kernel does not support mono audio out of the I2S interface, you can only play stereo, so any mono audio files may need conversion to stereo!

2017-11-2 Raspbian PIXEL ('full') has broken something in volume control. I2S works, but there's no software volume setup, if you need this, try Raspbian Lite - will try to fix as soon as we figure out why :)

Setup Virtual Environment

If you are installing on the Bookworm version of Raspberry Pi OS or later, you will need to install your python modules in a virtual environment. You can find more information in the [Python Virtual Environment Usage on Raspberry Pi \(https://adafru.it/19a5\)](https://adafru.it/19a5) guide. To Install and activate the virtual environment, use the following commands:

```
sudo apt install python3-venv
python -m venv env --system-site-packages
```

To activate the virtual environment:

```
source env/bin/activate
```

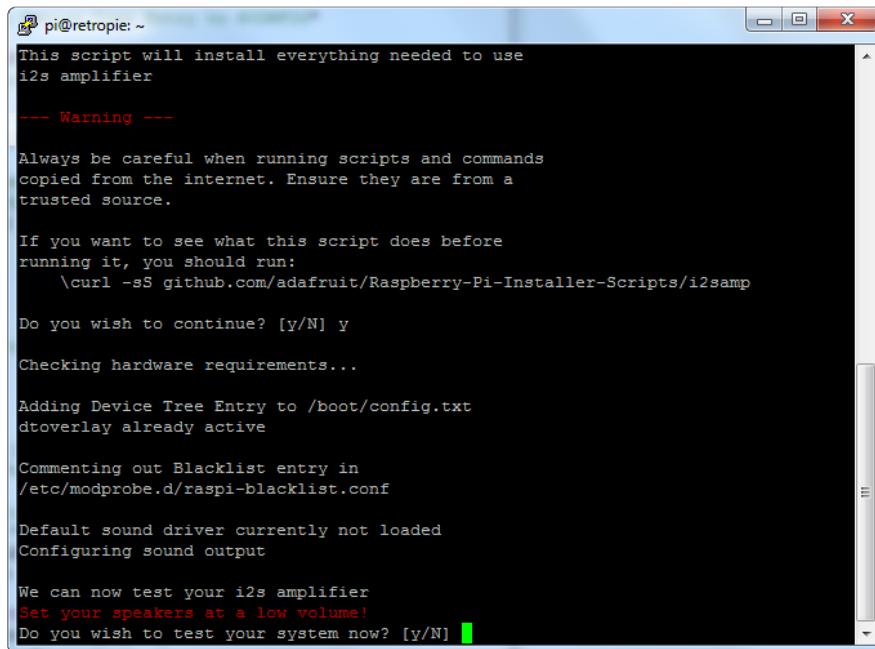
Installer script

Luckily its quite easy to install support for I2S DACs on Raspbian.

These instructions are totally cribbed from the PhatDAC instructions at the lovely folks at Pimoroni! (<https://adafruit.it/nFy>)

Run the following from your Raspberry Pi with Internet connectivity:

```
sudo apt install -y wget
wget https://github.com/adafruit/Raspberry-Pi-Installer-Scripts/raw/main/i2samp.py
sudo -E env PATH=$PATH python3 i2samp.py
```



```
pi@retropie: ~
This script will install everything needed to use
i2s amplifier

--- Warning ---

Always be careful when running scripts and commands
copied from the internet. Ensure they are from a
trusted source.

If you want to see what this script does before
running it, you should run:
  \curl -sS https://github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp

Do you wish to continue? [y/N] y
Checking hardware requirements...
Adding Device Tree Entry to /boot/config.txt
dtoverlay already active
Commenting out Blacklist entry in
/etc/modprobe.d/raspi-blacklist.conf
Default sound driver currently not loaded
Configuring sound output

We can now test your i2s amplifier
Set your speakers at a low volume!
Do you wish to test your system now? [y/N] y
```

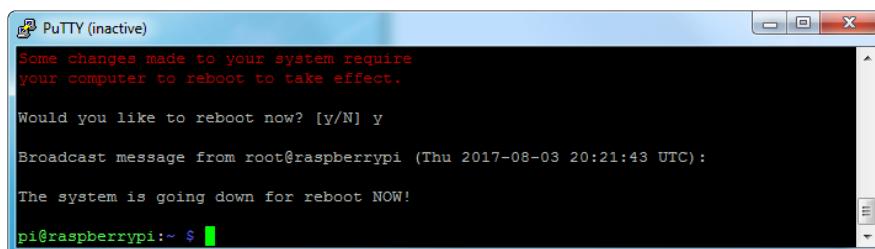
We've added an extra helper systemd script that will play quiet audio when the I2S peripheral isn't in use. This removes popping when playback starts or stops. It uses a tiny amount of CPU time (on a Pi Zero, 5%, on a Pi 2 or 3 its negligible). You don't need this on RetroPie because it never releases the I2S device, but it's great for Raspbian.

```
Installing aplay systemd unit
Removed /etc/systemd/system/multi-user.target.wants/aplay.service.

You can optionally activate '/dev/zero' playback in
the background at boot. This will remove all
popping/clicking but does use some processor time.

Activate '/dev/zero' playback in background? [RECOMMENDED] [y/N] y
```

You will need to reboot once installed.

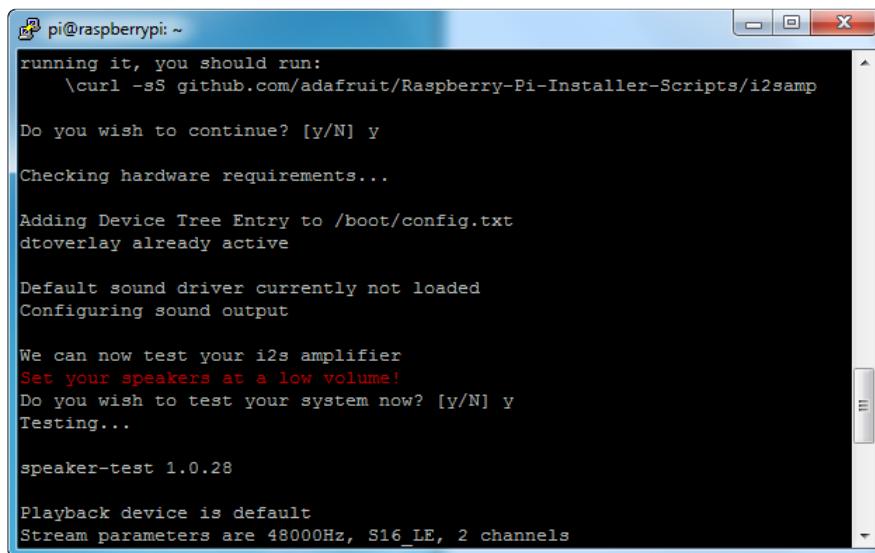


```
PuTTY (inactive)
Some changes made to your system require
your computer to reboot to take effect.

Would you like to reboot now? [y/N] y
Broadcast message from root@raspberrypi (Thu 2017-08-03 20:21:43 UTC):
The system is going down for reboot NOW!
pi@raspberrypi: ~ $
```

You must reboot to enable the speaker hardware!

After rebooting, log back in and re-run the script again...It will ask you if you want to test the speaker. Say yes and listen for audio to come out of your speakers...



```
pi@raspberrypi: ~
running it, you should run:
  curl -sS github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp

Do you wish to continue? [y/N] y

Checking hardware requirements...

Adding Device Tree Entry to /boot/config.txt
dtoverlay already active

Default sound driver currently not loaded
Configuring sound output

We can now test your i2s amplifier
Set your speakers at a low volume!
Do you wish to test your system now? [y/N] y
Testing...

speaker-test 1.0.28

Playback device is default
Stream parameters are 48000Hz, S16_LE, 2 channels
```

If it sounds really distorted, it could be the volume is too high. However, in order to have volume control appear in Raspbian desktop or Retropie you must reboot a second time after doing the speaker test, with **sudo reboot**

You must reboot *twice* to enable alsamixer volume (really!)

Once rebooted, try running **alsamixer** and use arrow keys to lower the volume, 50% is a good place to start.

If you're still having audio problems, try re-running the script and saying **N** (disable) the **/dev/zero playback service**.

You can then go to the next page on testing and optimizing your setup. Skip the rest of this page on **Detailed Installation** if the script worked for you!

Detailed Install

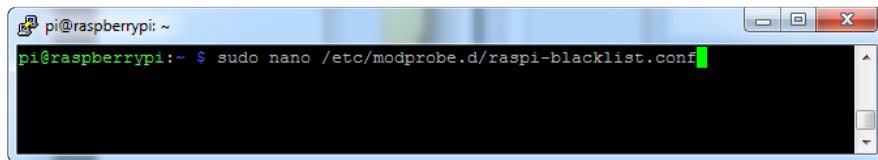
If, for some reason, you can't just run the script and you want to go through the install by hand - here's all the steps!

Update /etc/modprobe.d (if it exists)

Log into your Pi and get into a serial console (either via a console cable, the TV console, RXVT, or what have you)

Edit the raspi blacklist with

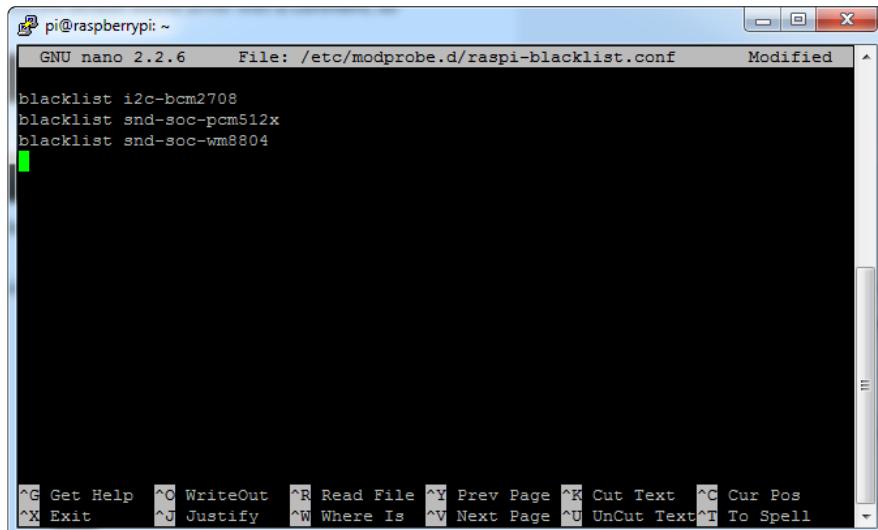
```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```



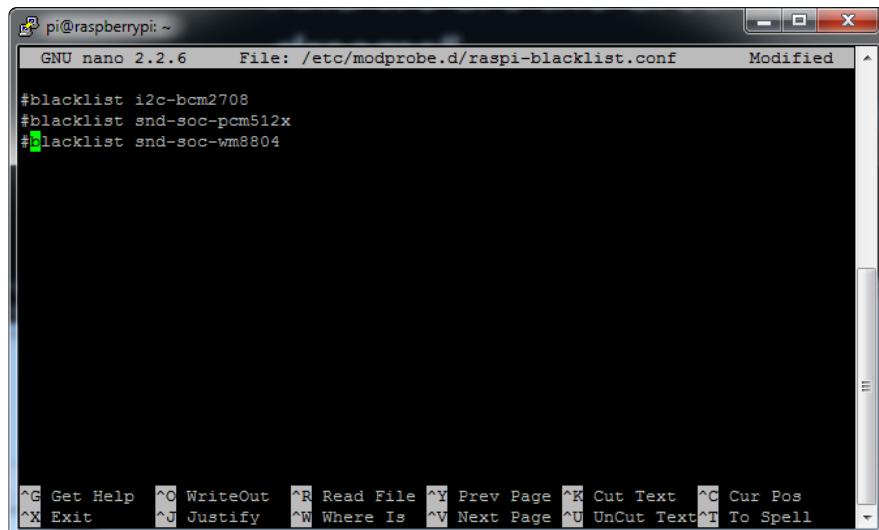
If the file is empty, just skip this step

However, if you see the following lines:

```
blacklist i2c-bcm2708
blacklist snd-soc-pcm512x
blacklist snd-soc-wm8804
```



Update the lines by putting a # before each line



```
pi@raspberrypi: ~
GNU nano 2.2.6      File: /etc/modprobe.d/raspi-blacklist.conf      Modified
#blacklist i2c-bcm2708
#blacklist snd-soc-pcm512x
#blacklist snd-soc-wm8804

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U Uncut Text  ^T To Spell
```

Save by typing **Control-X Y <return>**

Disable headphone audio (if it's set)

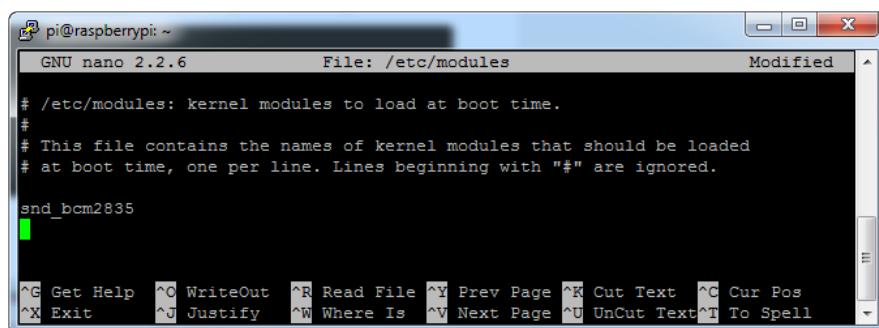
Edit the raspi modules list with

```
sudo nano /etc/modules
```

If the file is empty, just skip this step

However, if you see the following line:

snd_bcm2835

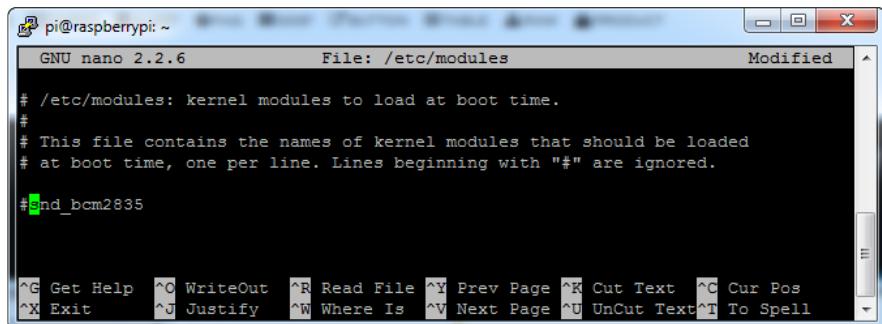


```
pi@raspberrypi: ~
GNU nano 2.2.6      File: /etc/modules      Modified
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

snd_bcm2835

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U Uncut Text  ^T To Spell
```

Put a **#** in front of it



```
pi@raspberrypi: ~
GNU nano 2.2.6          File: /etc/modules          Modified
#
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
#
#snd_bcm2835

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

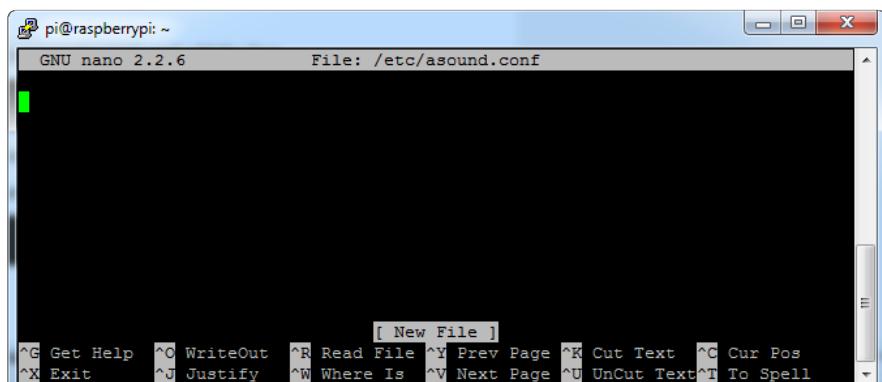
and save with **Control-X Y <return>**

Create asound.conf file

Edit the raspi modules list with

```
sudo nano /etc/asound.conf
```

This file ought to be blank!



```
pi@raspberrypi: ~
GNU nano 2.2.6          File: /etc/asound.conf          Modified
[ New File ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Copy and paste the following text into the file

```
pcm.speakerbonnet {
    type hw card 0
}

pcm.dmixer {
    type dmix
    ipc_key 1024
    ipc_perm 0666
    slave {
        pcm "speakerbonnet"
        period_time 0
        period_size 1024
        buffer_size 8192
        rate 44100
        channels 2
    }
}

ctl.dmixer {
    type hw card 0
}
```

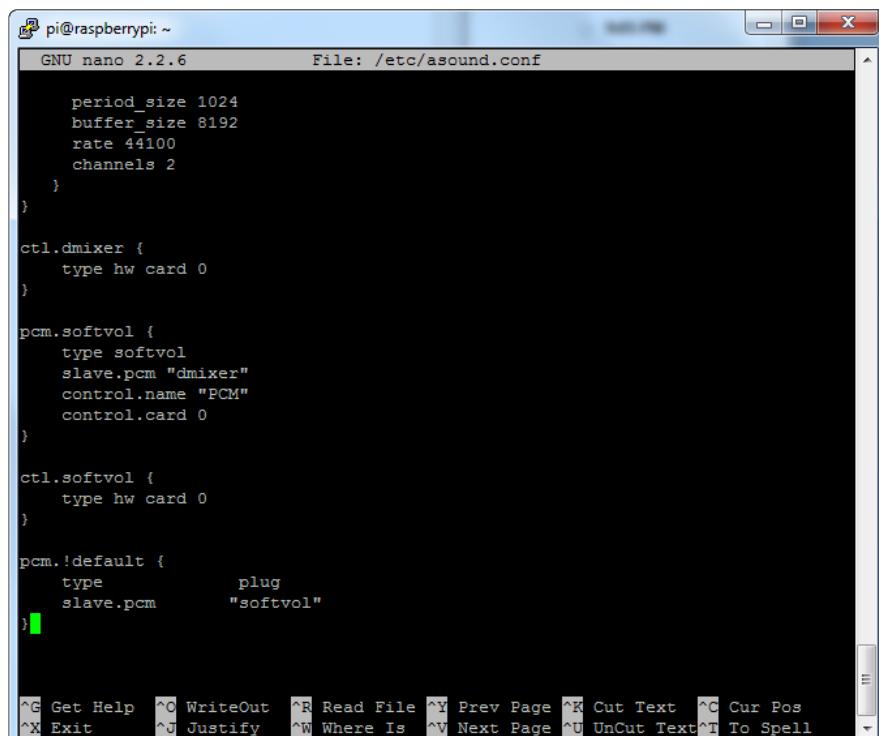
```

pcm.softvol {
    type softvol
    slave.pcm "dmixer"
    control.name "PCM"
    control.card 0
}

ctl.softvol {
    type hw card 0
}

pcm.!default {
    type plug
    slave.pcm "softvol"
}

```



```

pi@raspberrypi: ~
GNU nano 2.2.6          File: /etc/asound.conf

    period_size 1024
    buffer_size 8192
    rate 44100
    channels 2
}

ctl.dmixer {
    type hw card 0
}

pcm.softvol {
    type softvol
    slave.pcm "dmixer"
    control.name "PCM"
    control.card 0
}

ctl.softvol {
    type hw card 0
}

pcm.!default {
    type plug
    slave.pcm "softvol"
}

^G Get Help  ^Q WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell

```

Save the file as usual

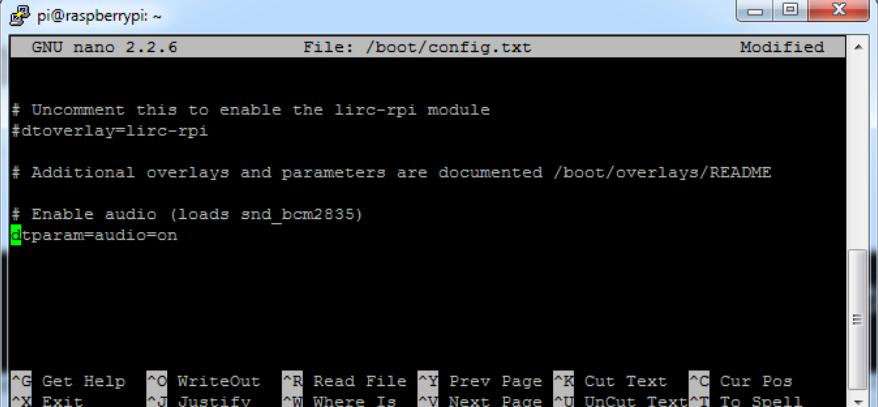
Add Device Tree Overlay

Edit your Pi configuration file with

For older versions of Raspberry Pi, edit /boot/config.txt instead.

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

And scroll down to the bottom. If you see a line that says: **dtparam=audio=on**



```
pi@raspberrypi: ~
GNU nano 2.2.6          File: /boot/config.txt          Modified

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Disable it by putting a # in front.

Then add:

```
dtoverlay=max98357a
```

on the next line. Save the file and reboot your Pi with

```
sudo reboot
```

Raspberry Pi Test

Speaker Tests!

OK you can use whatever software you like to play audio but if you'd like to test the speaker output, here's some quick commands that will let you verify your amp and speaker are working as they should!

Simple white noise speaker test

Run

```
speaker-test -c2
```

to generate white noise out of the speaker, alternating left and right.

If you have a mono output amplifier, the I2S amp merges left and right channels, so you'll hear continuous white noise

Simple WAV speaker test

Once you've got something coming out, try to play an audio file with **speaker-test** (for WAV files, not MP3)

```
speaker-test -c2 --test=wav -w /usr/share/sounds/alsa/Front_Center.wav
```

You'll hear audio coming from left and right alternating speakers

Simple MP3 speaker test

If you want to play a stream of music, you can try

```
sudo apt-get install -y mpg123
mpg123 http://ice1.somafm.com/u80s-128-mp3
```

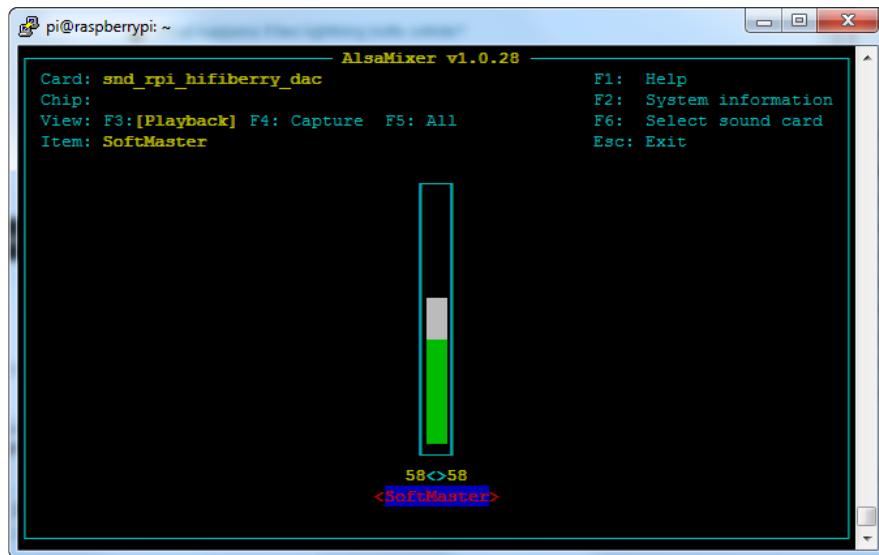
[If you want to play MP3's on command, check out this tutorial which covers how to set that up \(<https://adafruit.it/aTD>\)](#)

At this time, Jessie Raspberry Pi kernel **does not support mono audio** out of the I2S interface, **you can only play stereo**, so any mono audio files may need conversion to stereo!

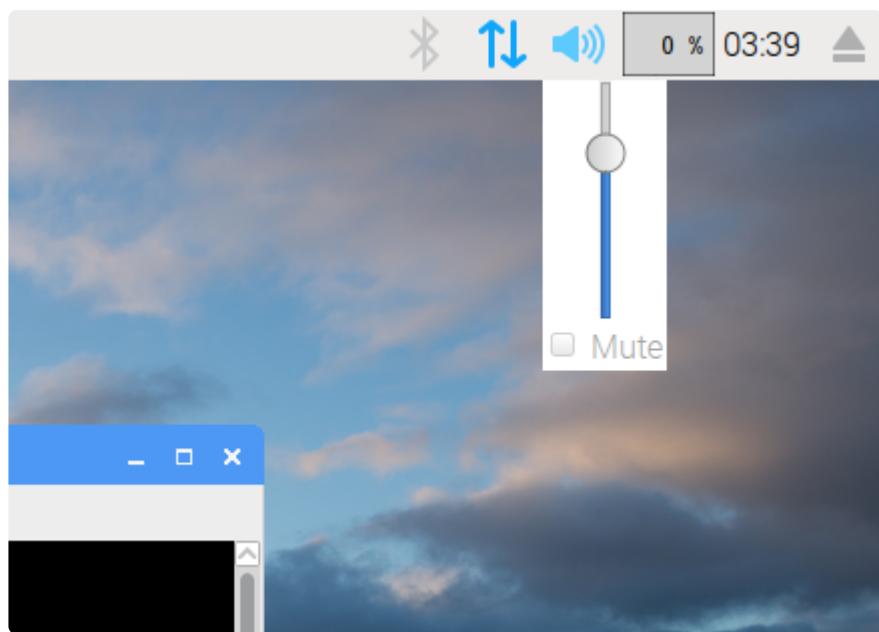
omxplayer does not seem use the I2S interface for audio - only HDMI - so you won't be able to use it

Volume adjustment

Many programs like PyGame and Sonic Pi have volume control within the application. For other programs you can set the volume using the command line tool called **alsamixer**. Just type alsamixer in and then use the up/down arrows to set the volume. Press Escape once its set



In Raspbian PIXEL you can set the volume using the menu item control. If it has an X through it, try restarting the Pi (you have to restart twice after install to get PIXEL to recognize the volume control



Pi I2S Tweaks

This page is deprecated, our installer already performs these steps for you, but we'll keep them here for archival use!

Reducing popping

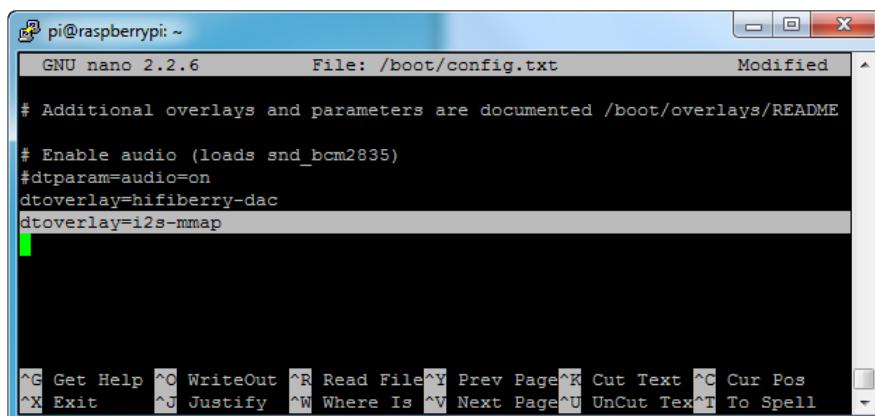
For people who followed our original installation instructions with the simple alsal config, they may find that the I2S audio pops when playing new audio.

The workaround is to use a software mixer to output a fixed sample rate to the I2S device so the bit clock does not change. I use ALSA so I configured **dmixer** and I no longer have any pops or clicks. Note that the RaspPi I2S driver does not support **dmixer** by default and you must [follow these instructions provided \(https://adafruit.it/sHF\)](https://adafruit.it/sHF) to add it. Continue on for step-by-step on how to enable it!

Step 1

Start by modify **/boot/firmware/config.txt** to add **dtoverlay=i2s-mmap**

Run **sudo nano /boot/firmware/config.txt** and add the text to the bottom like so:



```
pi@raspberrypi: ~
GNU nano 2.2.6          File: /boot/config.txt          Modified
# Additional overlays and parameters are documented /boot/overlays/README
# Enable audio (loads snd_bcm2835)
#dtparam=audio=on
dtoverlay=hifiberry-dac
dtoverlay=i2s-mmap

^G Get Help ^O WriteOut ^R Read File^Y Prev Page^K Cut Text ^C Cur Pos
^X Exit      ^J Justify ^W Where Is ^V Next Page^U UnCut Tex^I To Spell
```

If you are using an older version of Raspberry Pi OS and there is no firmware folder, you may need to edit the /boot/config.txt file.

Save and exit.

Then change **/etc/asound.conf** to:

```
pcm.speakerbonnet {
    type hw card 0
}

pcm.!default {
    type plug
    slave.pcm "dmixer"
}

pcm.dmixer {
```

```

type dmix
ipc_key 1024
ipc_perm 0666
slave {
    pcm "speakerbonnet"
    period_time 0
    period_size 1024
    buffer_size 8192
    rate 44100
    channels 2
}
}

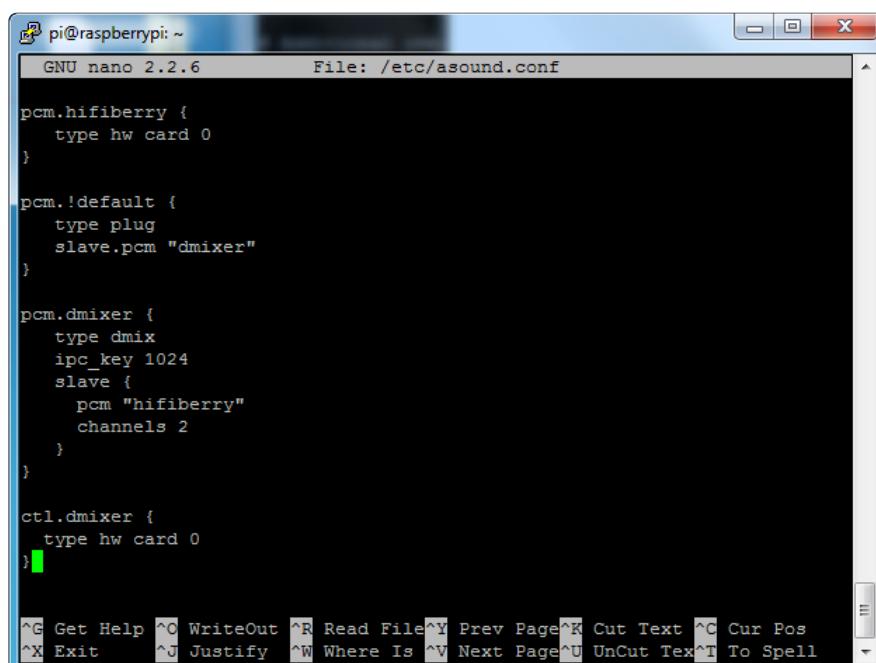
ctl.dmixer {
    type hw card 0
}

```

By running `sudo nano /etc/asound.conf`

This creates a PCM device called speakerbonnet which is connected to the hardware I2S device. Then we make a new 'dmix' device (`type dmix`) called `pcm.dmixer`. We give it a unique Inter Process Communication key (`ipc_key 1024`) and permissions that are world-read-writeable (`ipc_perm 0666`) The mixer will control the hardware pcm device speakerbonnet (pcm "speakerbonnet") and has a buffer set up so its nice and fast. The communication buffer is set up so there's no delays (`period_time 0`, `period_size 1024` and `buffer_size 8192` work well). The default mixed rate is 44.1khz stereo (`rate 44100 channels 2`)

Finally we set up a control interface but it ended up working best to just put in the hardware device here - `ctl.dmixer { type hw card 0 }`



Save and exit. Then reboot the Pi to enable the mixer. Also, while it will greatly reduce popping, you still may get one once in a while - especially when first playing audio!

Add software volume control

The basic I2S chipset used here does not have software control built in. So we have to 'trick' the Pi into creating a software volume control. [Luckily, its not hard once you know how to do it \(https://adafru.it/ydQ\)](https://adafru.it/ydQ).

Create a new audio config file in `~/.asoundrc` with `nano ~/ .asoundrc` and inside put the following text:

```
pcm.speakerbonnet {
    type hw card 0
}

pcm.dmixer {
    type dmix
    ipc_key 1024
    ipc_perm 0666
    slave {
        pcm "speakerbonnet"
        period_time 0
        period_size 1024
        buffer_size 8192
        rate 44100
        channels 2
    }
}

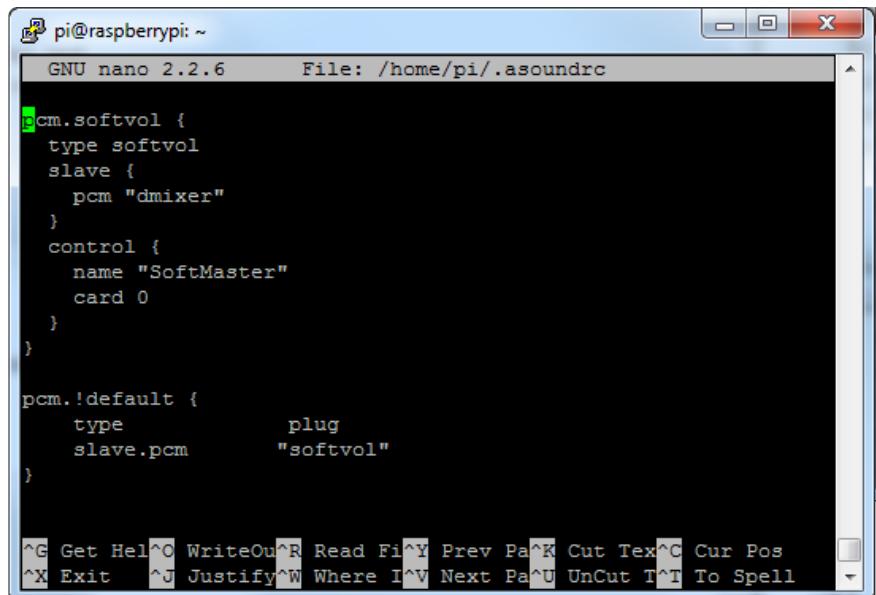
ctl.dmixer {
    type hw card 0
}

pcm.softvol {
    type softvol
    slave.pcm "dmixer"
    control.name "PCM"
    control.card 0
}

ctl.softvol {
    type hw card 0
}

pcm.!default {
    type plug
    slave.pcm "softvol"
}
```

This assumes you set up the dmixer for no-popping above!



```
pi@raspberrypi: ~
GNU nano 2.2.6      File: /home/pi/.asoundrc

pcm.softvol {
    type softvol
    slave {
        pcm "dmixer"
    }
    control {
        name "SoftMaster"
        card 0
    }
}

pcm.!default {
    type plug
    slave.pcm "softvol"
}

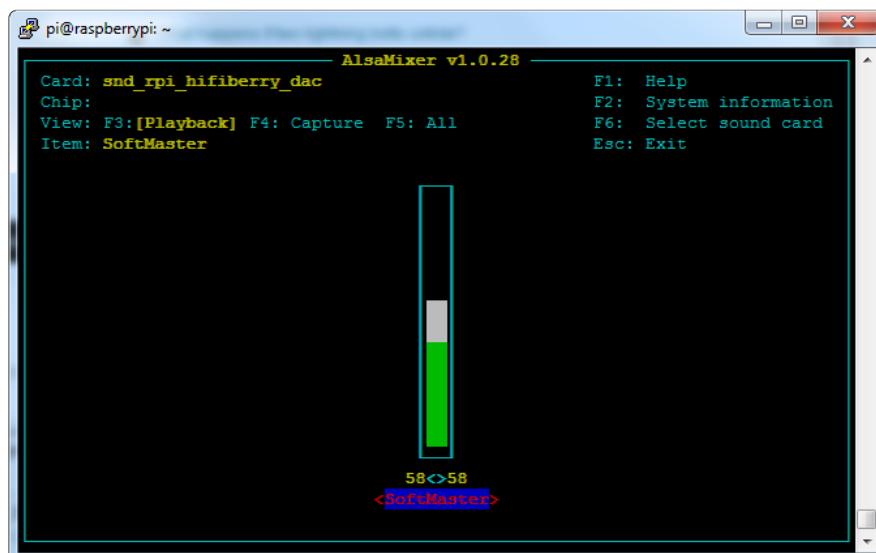
^G Get Help^O WriteOut^R Read File^Y Prev Page^K Cut Text^C Cur Pos
^X Exit ^J Justify^W Where I^V Next Page^U Uncut T^T To Spell
```

Save and exit

Now, here's the trick, you have to reboot, then play some audio through alsound, then reboot to get the alsamixer to sync up right:

```
speaker-test -c2 --test=wav -w /usr/share/sounds/alsa/
Front_Center.wav
```

Then you can type **alsamixer** to control the volume with the 'classic' alsound mixing interface



Just press the up and down arrows to set the volume, and ESC to quit

I2S Audio FAQ

Hey in Raspbian Pixel desktop, the speaker icon is X'd out!

Try rebooting once after playing some audio. Also make sure you have our latest alsound configuration (check the detailed install page on the Raspberry Pi Setup page for the `/etc/asound.conf` !

If it's still not working, you can still change the volume, just use `alsamixer` from a Terminal command prompt.

Even with dmixer enabled, I get a staticy-pop when the Pi first boots or when it first starts playing audio

Yep, this is a known Raspbian Linux thing. Yay Linux! We don't have a fix for it. If it makes you feel better, my fancy Windows development computer does the same thing with my desktop speakers.

You can reduce popping a lot with the '/dev/zero play' option in the i2s setup script. (We added it in October 2018) but you still might get that one initial pop

The audio on my DAC sounds really bad/distorted

Make sure you've lowered the volume. The default is 100% which is waaaay too high! Use `alsamixer` to reduce the volume to 50% or so

Does this work with my favorite software?

It will work with anything that has alsound audio support. There's thousands of linux programs so we can't guarantee all of them will work but here's what we found does for sure!

- **PyGame** - see our page on [playing audio with PyGame](https://adafruit.it/ydS) (<https://adafruit.it/ydS>) for example code. Volume can be controlled within pygame
- **mpg123** - command line mp3 audio playback. use alsamixer to control the volume
- **aplay** - for playing wav files on the command line
- **Sonic Pi** - tested in the Pixel Desktop. Use the Sonic Pi settings panel to change the volume - it does not seem to care about what global audio volume you set!
- **Scratch 2** - tested in the Pixel Desktop. Works fine but may have a delay and make a popping sound the first time you play audio. You can set volume with alsamixer and also via the app by using the **set volume to nn%** block
- **Scratch 1** - doesn't work, something not set up with Scratch 1 to use alsound

- **RetroPie/Emulation Station** - audio works within games (we tested NES and MAME libretro) but does not work in the 'main screen' (selecting which game to play interface)

Play Audio with PyGame

You can use **mpg123** for basic testing but it's a little clumsy for use where you want to dynamically change the volume or have an interactive program. For more powerful audio playback we suggest using PyGame to playback a variety of audio formats (MP3 included!)

Install PyGame

Start by installing pygame support, you'll need to open up a console on your Pi with network access and run:

```
sudo apt-get install python3-pygame
```

Next, download this pygame example zip to your Pi

Click to download PyGame example code & sample mp3s

<https://adafru.it/wbp>

On the command line, run

```
wget https://cdn-learn.adafruit.com/assets/assets/000/041/506/original/pygame\_example.zip (https://adafru.it/wbq)
```

```
unzip pygame_example.zip (https://adafru.it/wbq)
```

Run Demo

Inside the zip is an example called **pygameMP3.py**

This example will playback all MP3's within the script's folder. To demonstrate that you can also adjust the volume within pygame, the second argument is the volume for playback. Specify a volume to playback with a command line argument between 0.0 and 1.0

For example here is how to play at 75% volume:

```
python pygameMP3.py 0.75
```

Here's the code if you have your own mp3s!

```
''' pg_midi_sound101.py
play midi music files (also mp3 files) using pygame
tested with Python273/331 and pygame192 by vegaseat
'''

#code modified by James DeVito from here: https://www.daniweb.com/programming/software-development/code/454835/let-pygame-play-your-midi-or-mp3-files

#!/usr/bin/python

import sys
import pygame as pg
import os
import time

def play_music(music_file):
    '''
    stream music with mixer.music module in blocking manner
    this will stream the sound from disk while playing
    '''
    clock = pg.time.Clock()
    try:
        pg.mixer.music.load(music_file)
        print("Music file {} loaded!".format(music_file))
    except pygame.error:
        print("File {} not found! {}".format(music_file, pg.get_error()))
        return

    pg.mixer.music.play()

    # If you want to fade in the audio...
    # for x in range(0,100):
    #     pg.mixer.music.set_volume(float(x)/100.0)
    #     time.sleep(.0075)
    # # check if playback has finished
    while pg.mixer.music.get_busy():
        clock.tick(30)

freq = 44100    # audio CD quality
bitsize = -16    # unsigned 16 bit
channels = 2      # 1 is mono, 2 is stereo
buffer = 2048    # number of samples (experiment to get right sound)
pg.mixer.init(freq, bitsize, channels, buffer)

if len(sys.argv) > 1:
    try:
        user_volume = float(sys.argv[1])
    except ValueError:
        print "Volume argument invalid. Please use a float (0.0 - 1.0)"
        pg.mixer.music.fadeout(1000)
        pg.mixer.music.stop()
        raise SystemExit

    print("Playing at volume: " + str(user_volume)+ "\n")
    pg.mixer.music.set_volume(user_volume)
```

```

mp3s = []
for file in os.listdir("."):
    if file.endswith(".mp3"):
        mp3s.append(file)

print mp3s

for x in mp3s:
    try:
        play_music(x)
        time.sleep(.25)
    except KeyboardInterrupt:
        # if user hits Ctrl/C then exit
        # (works only in console mode)
        pg.mixer.music.fadeout(1000)
        pg.mixer.music.stop()
        raise SystemExit
else:
    print("Please specify volume as a float! (0.0 - 1.0)")

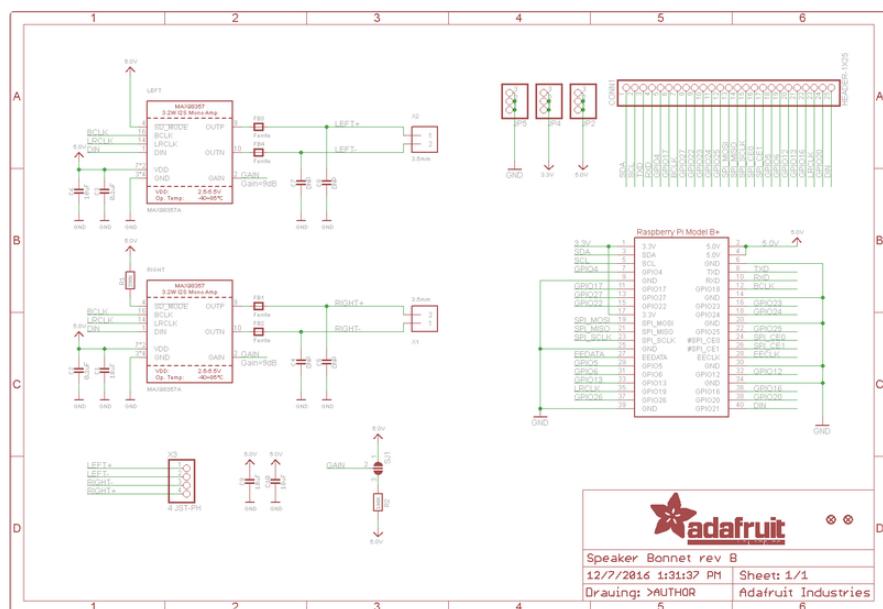
```

Downloads

Datasheets & Files

- [MAX98357 Datasheet \(http://adafru.it/30069835798357\)](http://adafru.it/30069835798357)
- [EagleCAD PCB files \(https://adafru.it/t1d\)](https://adafru.it/t1d)
- [Fritzing object in Adafruit Fritzing library \(https://adafru.it/c7M\)](https://adafru.it/c7M)

Schematic



Fabrication Print

Dims in mm

