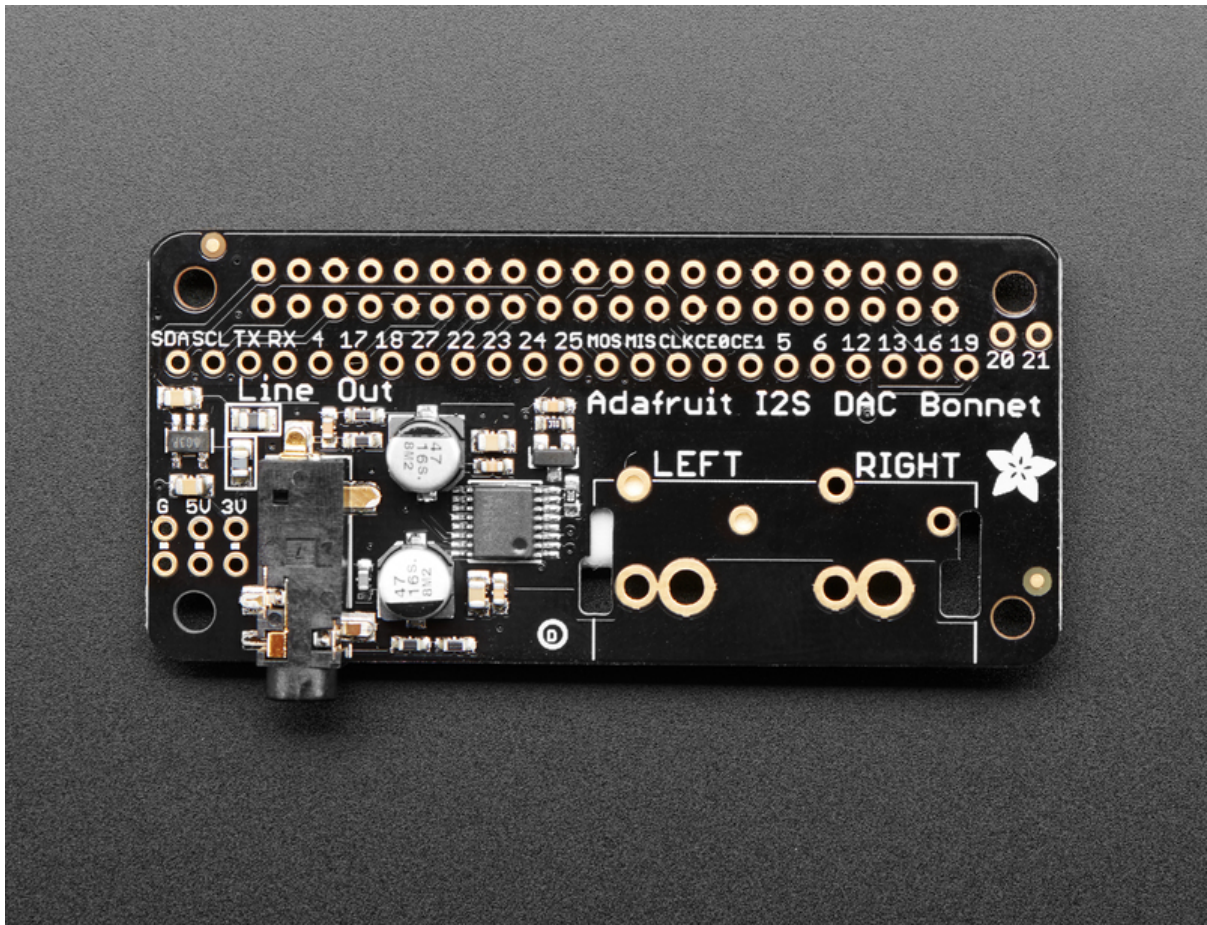




# Adafruit I2S Audio Bonnet for Raspberry Pi

Created by lady ada



<https://learn.adafruit.com/adafruit-i2s-audio-bonnet-for-raspberry-pi>

Last updated on 2024-11-19 01:11:50 PM EST

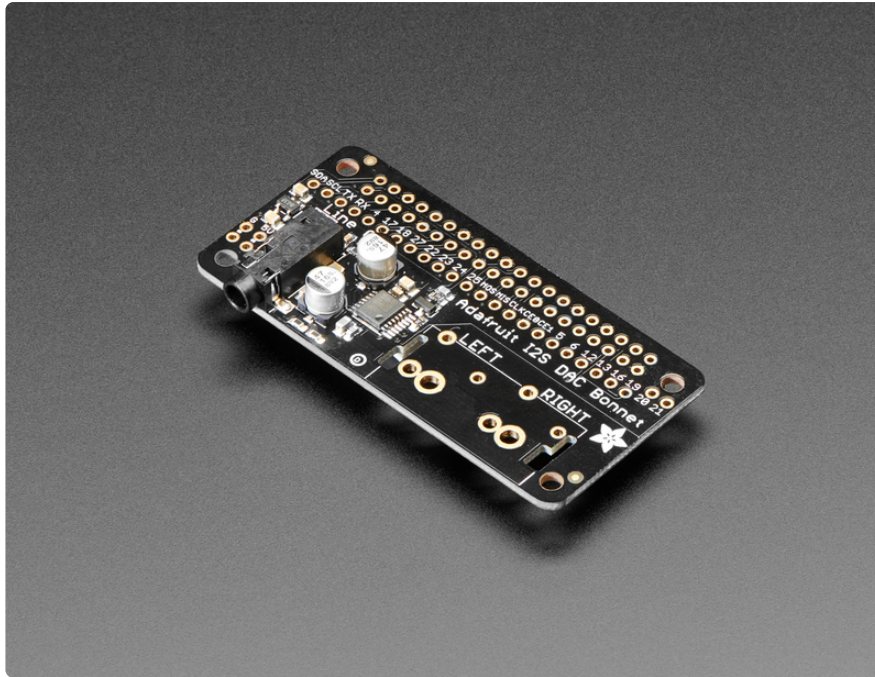
# Table of Contents

<a href="#">Overview</a>	<a href="#">3</a>
<a href="#">Pinouts</a>	<a href="#">5</a>
<ul style="list-style-type: none"><li>• <a href="#">I2S Audio Pins</a></li><li>• <a href="#">Audio Output</a></li></ul>	
<a href="#">Raspberry Pi Setup</a>	<a href="#">7</a>
<ul style="list-style-type: none"><li>• <a href="#">Setup Virtual Environment</a></li><li>• <a href="#">Installer script</a></li><li>• <a href="#">Detailed Install</a></li><li>• <a href="#">Update /etc/modprobe.d (if it exists)</a></li><li>• <a href="#">Disable headphone audio (if it's set)</a></li><li>• <a href="#">Create asound.conf file</a></li><li>• <a href="#">Add Device Tree Overlay</a></li></ul>	
<a href="#">Raspberry Pi Test</a>	<a href="#">14</a>
<ul style="list-style-type: none"><li>• <a href="#">Speaker Tests!</a></li><li>• <a href="#">Simple white noise speaker test</a></li><li>• <a href="#">Simple WAV speaker test</a></li><li>• <a href="#">Simple MP3 speaker test</a></li><li>• <a href="#">Volume adjustment</a></li></ul>	
<a href="#">Play Audio with PyGame</a>	<a href="#">16</a>
<ul style="list-style-type: none"><li>• <a href="#">Install PyGame</a></li><li>• <a href="#">Run Demo</a></li></ul>	
<a href="#">I2S Audio FAQ</a>	<a href="#">18</a>
<a href="#">Downloads</a>	<a href="#">20</a>
<ul style="list-style-type: none"><li>• <a href="#">Files</a></li><li>• <a href="#">Schematic &amp; Fabrication Print</a></li></ul>	

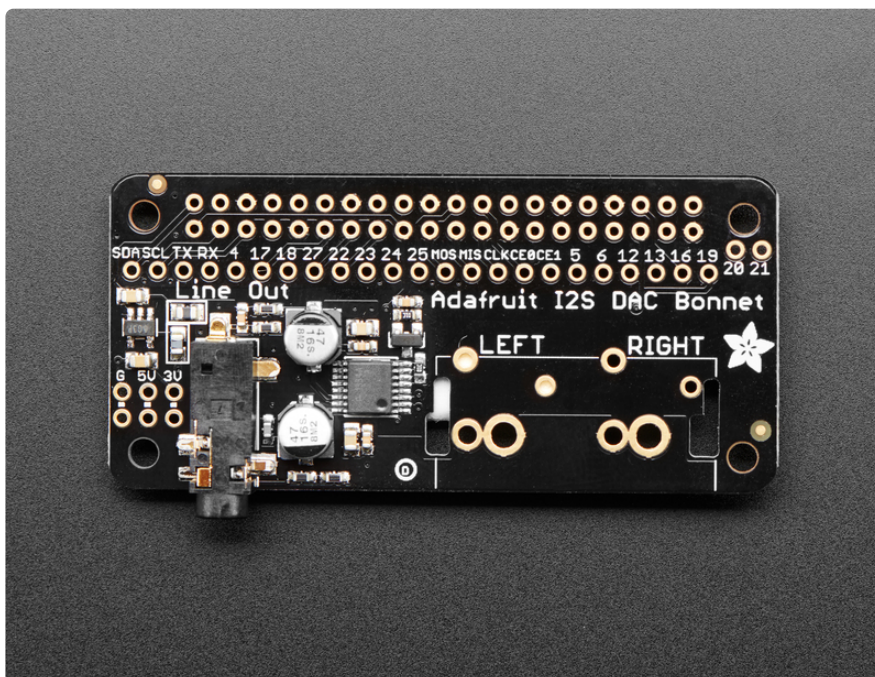
---

# Overview

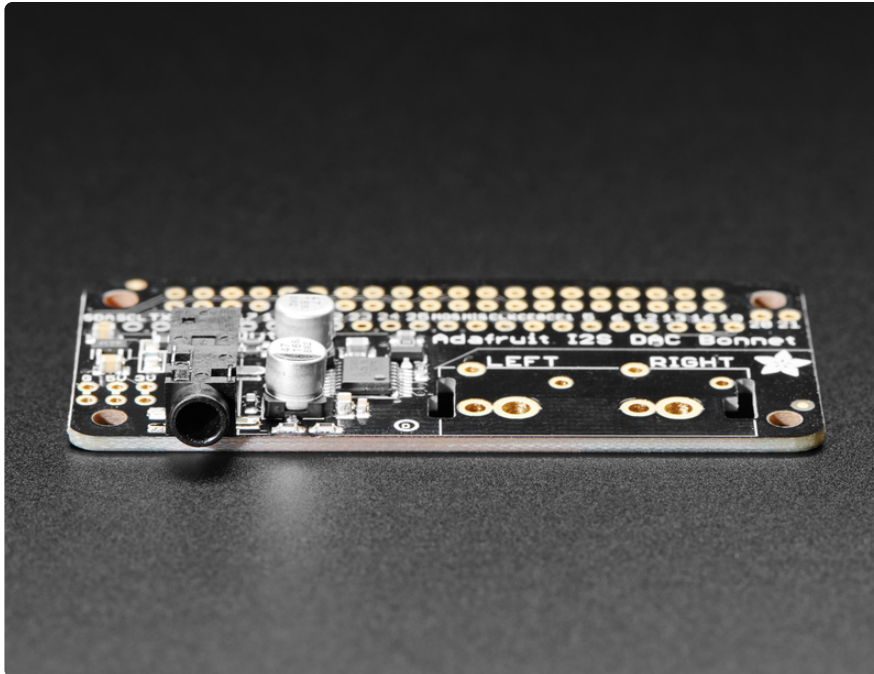
Add some easy-listenin' tunes to your Raspberry Pi using this basic audio bonnet. It'll give you stereo line out from a digital I2S converter for a good price, and sounds nice to boot! This bonnet features the UDA1334A I2S Stereo DAC, a perfect match for any I2S-output audio interface. It's affordable but sounds great - music to our ears.



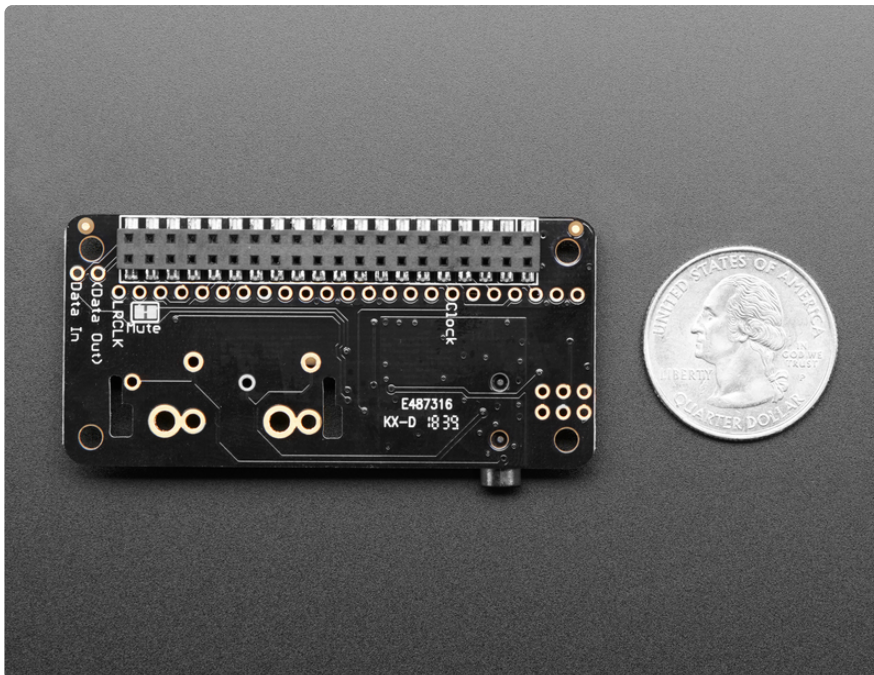
It's the exact same size as a Raspberry Pi Zero but works with **any and all** Raspberry Pi computers with a 2x20 connector - A+, B+, Zero, Pi 2, Pi 3, etc. We've tested it out with Raspbian (the official operating system) and RetroPie.



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 onboard headphone jack. And it's super easy to get started. Just plug in any line level output into powered speakers or audio system. You can plug in headphones but you'll get some distortion - so its best used into something that is expecting line level out

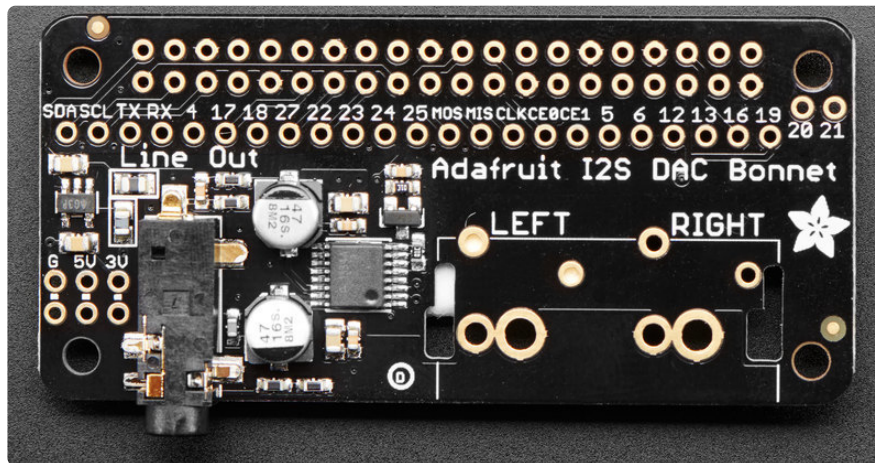


Each order comes as a fully tested and assembled PCB with a 2x20 socket connector. No soldering required! Plug it right onto your pi, install the script and reboot. Pow!



---

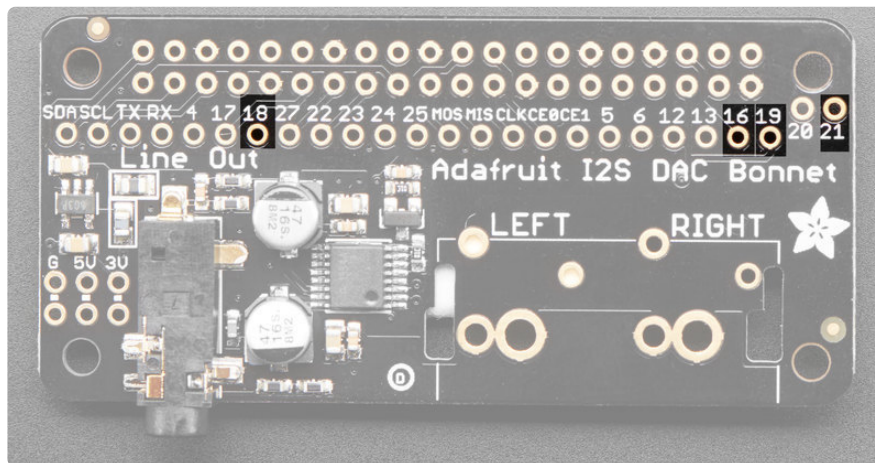
# Pinouts



## I2S Audio Pins

The UDA1334A is an **I2S** amplifier - it does not use analog inputs, it only has digital audio input support! Don't confuse I2S with I2C, I2S is a sound protocol whereas I2C is for small amounts of data.

There are 3 required data pins used on the Pi for I2S Audio, and 2 pins that are optional but good to know about.

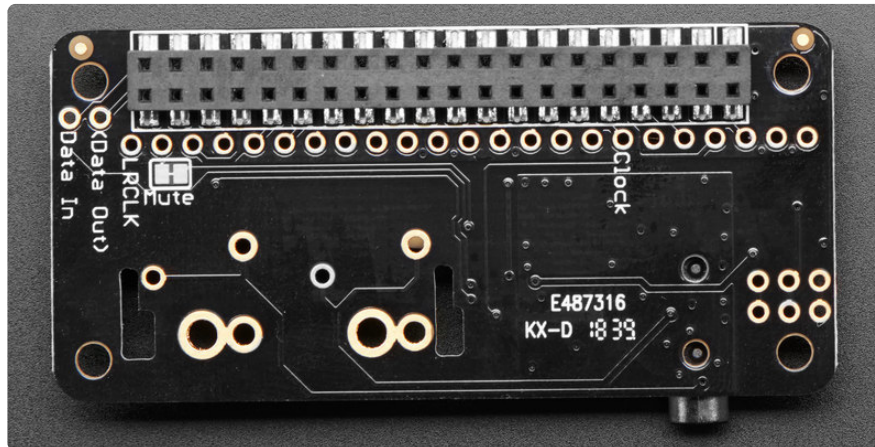


### Required pins

- **Pin #18** - I2S Clock
- **Pin #19** - I2S LR Clock (also sometimes called WS Clock)
- **Pin #21** - I2S Data Into the amplifier (out from Pi)

## Optional pins

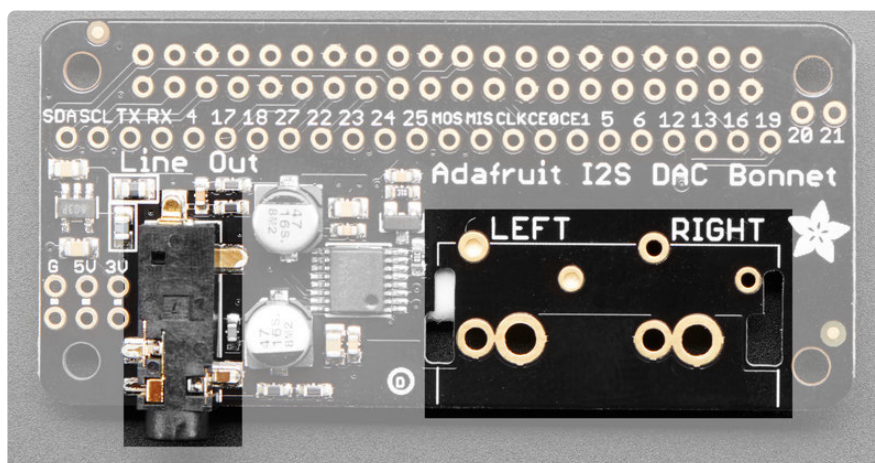
- **Pin #16** - Mute pin, you can use this to mute the output complete, sometimes helpful if you want to keep audio running and then turn it on/off with a switch or button.
- **Pin #20** - I2S Data Out from a microphone (into the Pi)



You don't have to do anything with that info other than know that pins 18, 19, 21 are not available for to use and pin #16 is by default used for muting. Pin #20 isnt used by the bonnet, but its handy if you plan to connect some other I2S microphone in.

## Audio Output

The output from the DAC is into a standard 3.5mm stereo headphone jack. Output is 'live level' (approx 1Vpp). You can plug in a set of headphones but the output isn't designed for driving low impedance so you will get some distortions at loud volume. The output is fully AC coupled with 47uF capacitors, sleeve ground is connected to the Pi ground.



There's also a breakout slot for soldering in dual/stereo RCA jacks, not included with the bonnet.

Or you can just use one of these cables:



### 3.5mm Stereo to RCA (Composite Audio) Cable 6 feet

One of the most common converter cables you'll want in your cable collection - this 6 foot long cable converts a standard 3.5mm stereo jack to two color-coded mono RCA coaxial...

<https://www.adafruit.com/product/1013>

---

## 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
```

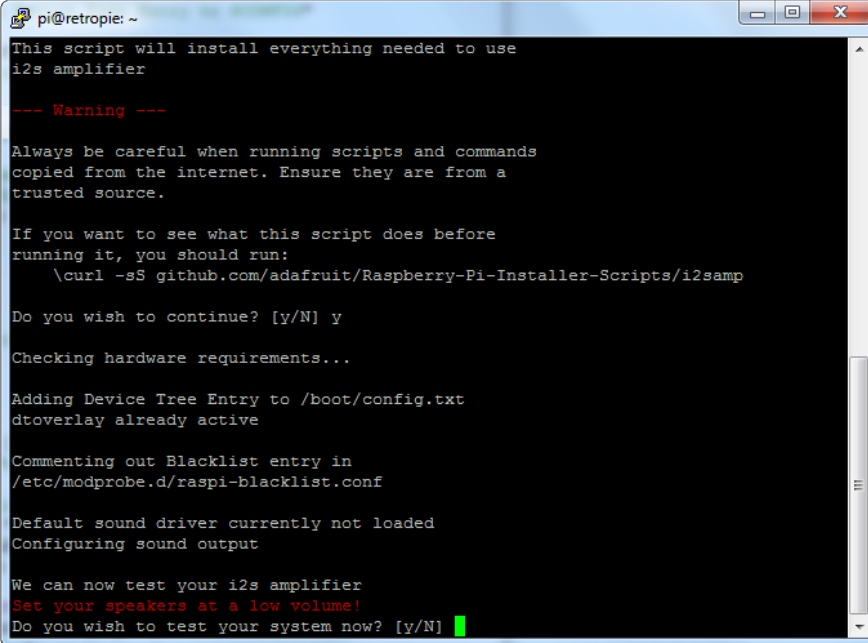
# Installer script

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! \(https://adafru.it/nFy\)](https://adafru.it/nFy)

Run the following from your Raspberry Pi with Internet connectivity:

```
sudo apt install -y wget
pip3 install adafruit-python-shell
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 github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp

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

Checking hardware requirements...

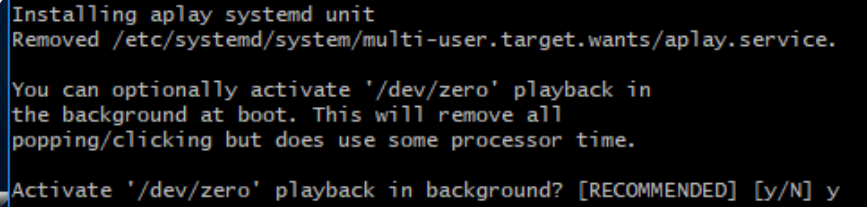
Adding Device Tree Entry to /boot/config.txt
dtoverlay already active

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] █
```

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.

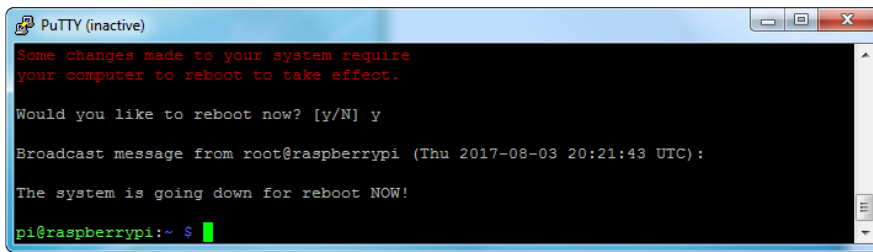


```
Installing aplay systemd unit
Removed /etc/systemd/system/multi-user.target.wants/aplay.service.

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

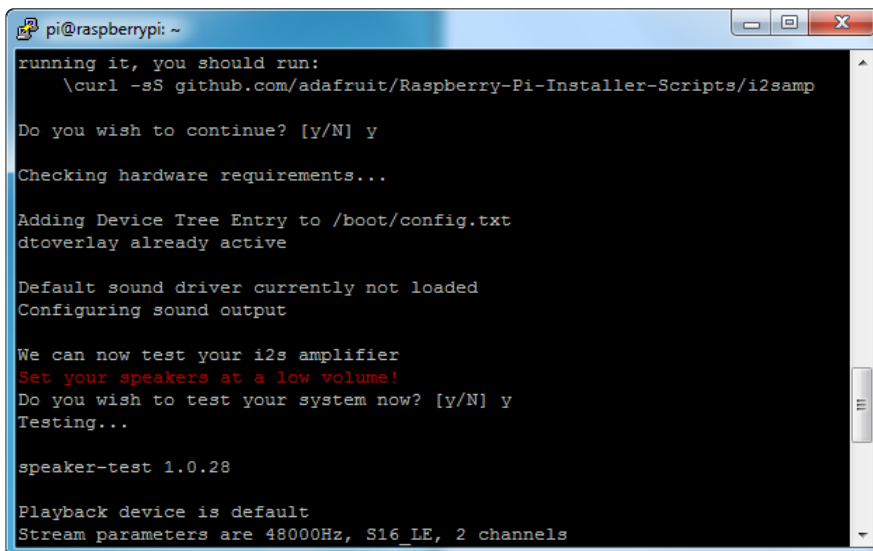
Broadcast message from root@raspberrypi (Thu 2017-08-03 20:21:43 UTC):

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

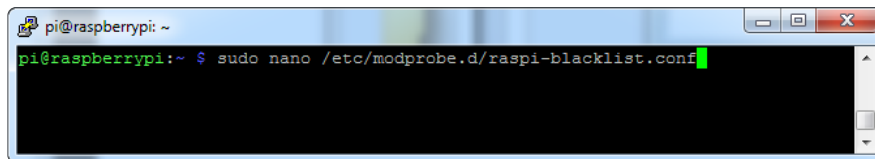
If, for some reason, you can't just run the script and you want to go through the install by hand - here's all the steps!

### 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

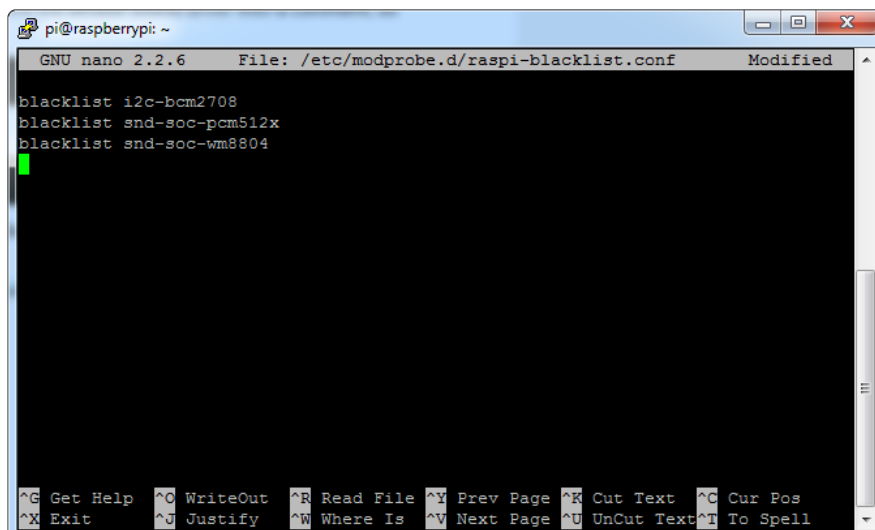
```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```



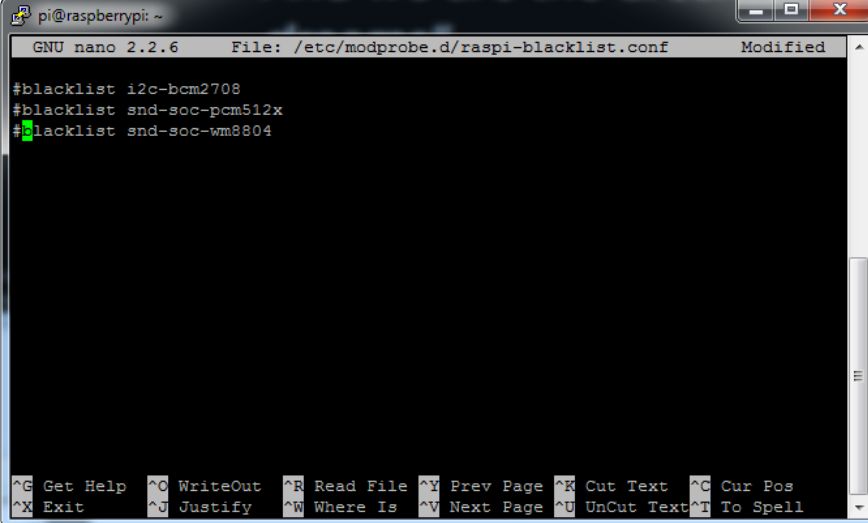
If the file is empty, just skip this step

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)

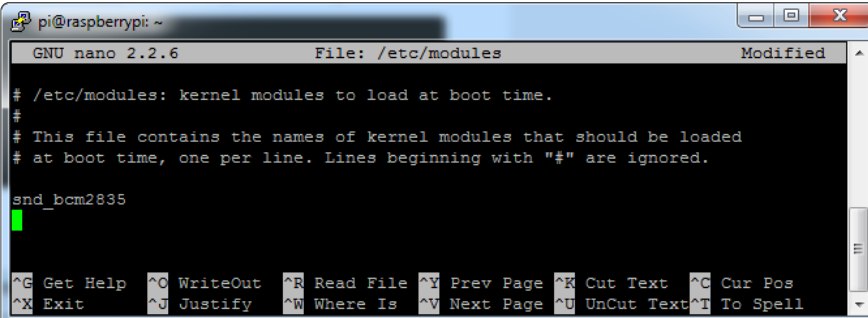
Edit the raspi modules list with

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

If the file is empty, just skip this step

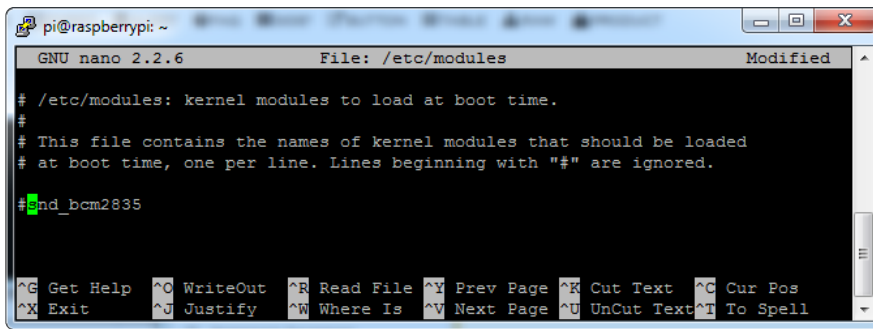
However, if you see the following line:

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

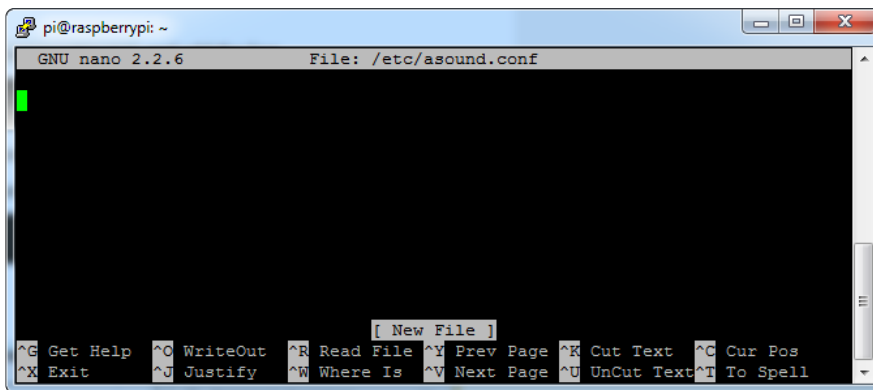
and save with **Control-X Y <return>**

## 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
[ New File ]
^G Get Help ^C WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^O 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
    }
}

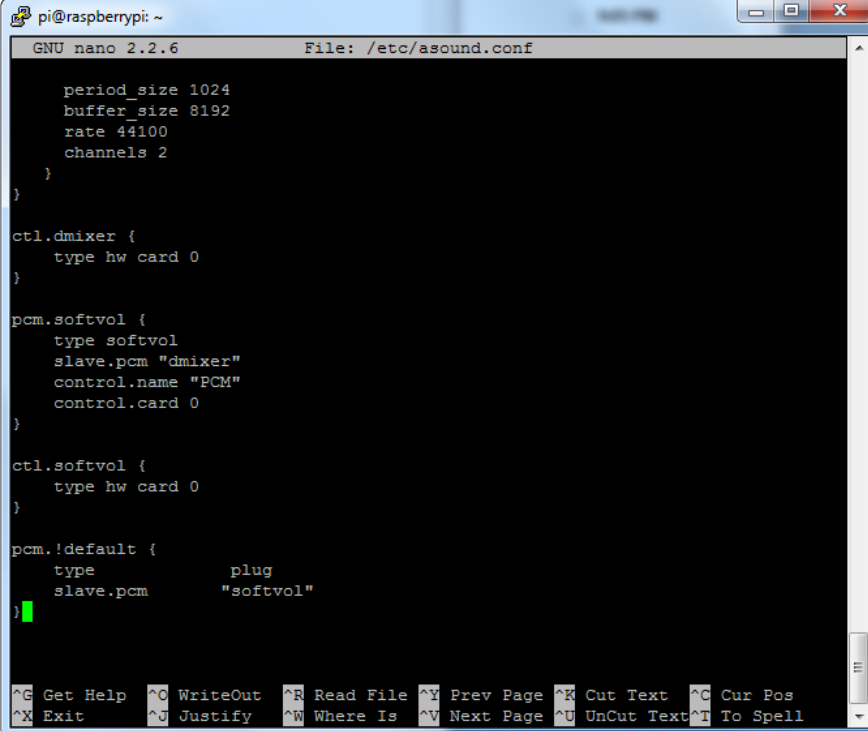
ctl.dmixer {
    type hw card 0
}

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"
}
}
```

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
```

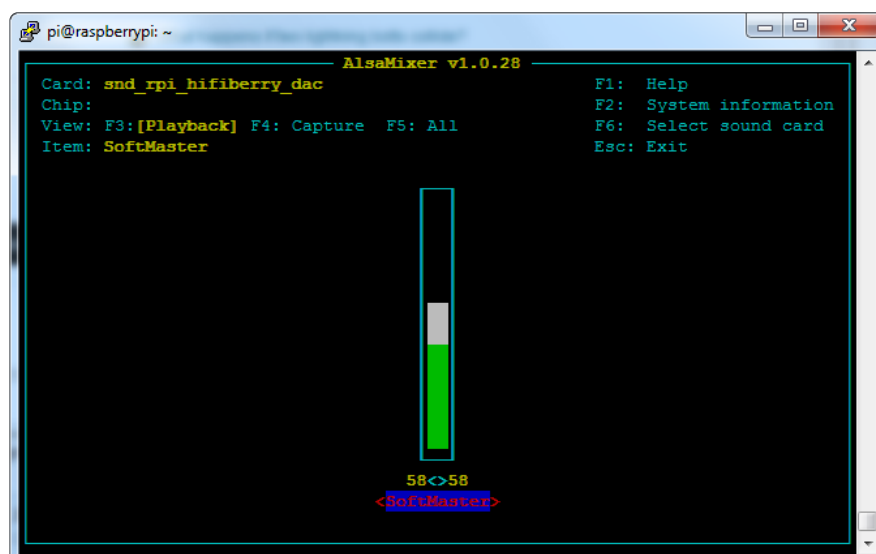
[If you want to play MP3's on command, check out this tutorial which covers how to set that up \(https://adafru.it/aTD\)](https://adafru.it/aTD)

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!

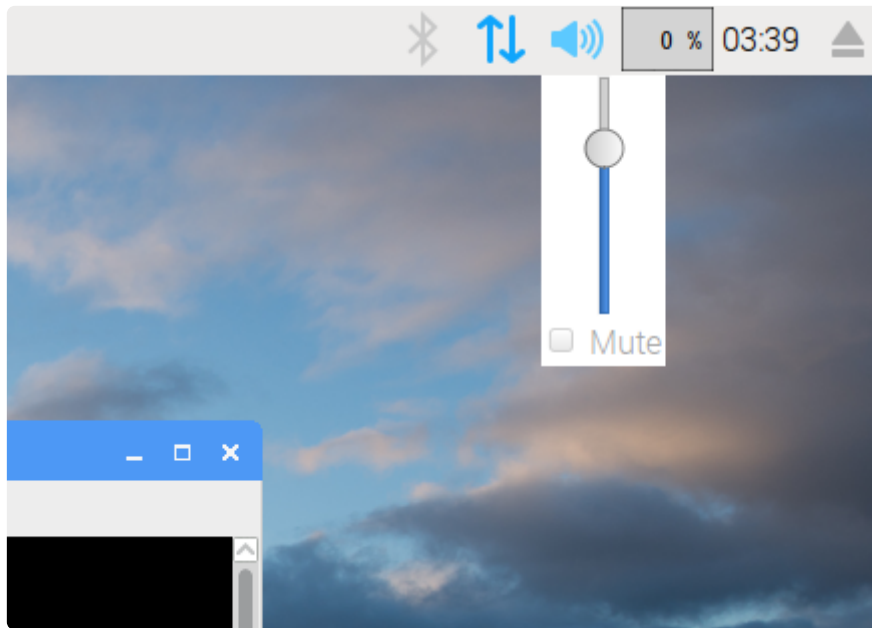
omxplayer does not seem use the I2S interface for audio - only HDMI - so you won't be able to use it

## 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)



---

## 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)

<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) (<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)")

```

## 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 alsa configuration (check the detailed install page on the Raspberry Pi Setup page for the [/etc/asound.conf](#) !

If its 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 alsa 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 \(https://adafru.it/ydS\)](https://adafru.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 also?
- **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)

## Downloads

## Files

- [EagleCAD PCB Files \(https://adafru.it/D9t\)](https://adafru.it/D9t)
- [UDA1334A Datasheet \(http://adafru.it/36781334\)](http://adafru.it/36781334)

## Schematic & Fabrication Print

Click to embiggen

