IQaudIO Product Guide

This document covers the installation, software configuration and usage of IQaudIO audio accessories with the Raspberry Pi.

If you would like to see additional information within this document please email us at info@iqaudio.com

> **For those who simply need the /boot/config.txt details —> Just jump to section 3**
>
> **For ROON LABS setup —-> Just jump to section 21**
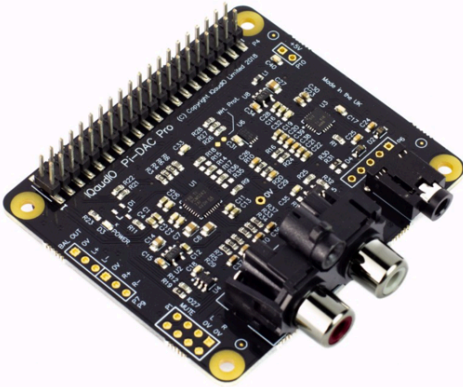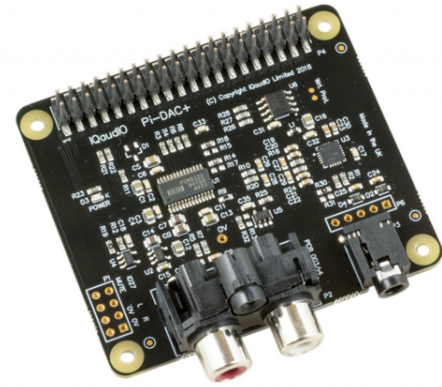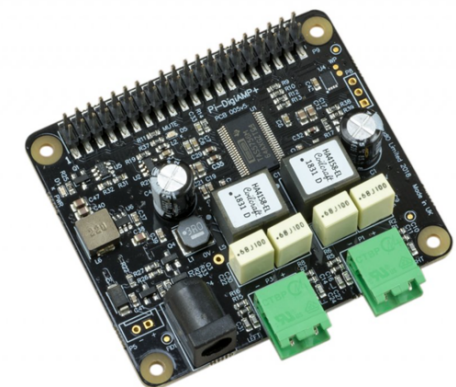
IQaudIO user instructions

Contents
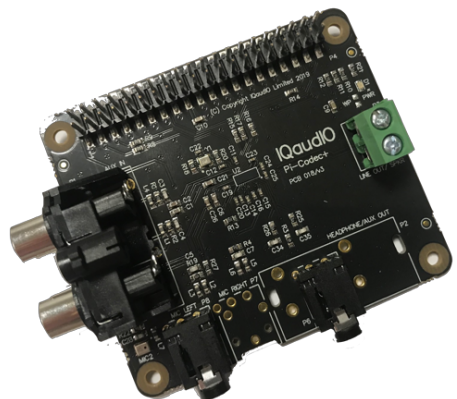
1. Getting Started

Thank you for choosing an IQaudIO sound card. Next you need to play music, but first you may need to configure the Raspberry Pi's and software package you want to use.

The IQaudIO product family.

The IQaudIO sound cards are all designed to work with the current Raspberry Pi and those that have a 40way male GPIO pin header. This includes the Pi4, Pi3B+, Pi3A+, Pi3, Pi-Zero etc. Some of our products are full size (HAT) boards for the full size Raspberry Pi, and others are more suited to the Pi-Zero form factor. We also have accessory boards that either act as accessories to our sound cards or can be used standalone (such as the CosmicController).

Pi-DAC PRO	Pi-DAC+	Pi-DigiAMP+
Phono (RCA), Balanced and headphone output	Phono (RCA) and headphone output	Passive Speaker output, 19v input

Pi-Codec+	Pi-CodecZero	Pi-DACZero+
Built in Mic, Stereo Aux In / Out, Mono speaker, built in MEMS mic and Mono ext mic support	Built in Mic, Mono speaker, built in MEMS mic, 2xLED, 1x Button and Mono ext mic support	Phono (RCA) output, supports optional Pi-DACZeroHeadphones daughter card.

High Definition Audio Playback

	Pi-DAC PRO	Pi-DAC+	Pi-DigiAMP+	Pi-Codec+	Pi-CodecZero	Pi-DACZero
24bit Depth	★	★	★	★	★	★
Hi-Def Audio (96kHz)	★	★	★	★	★	★
Full HD Audio (192kHz)	★	★	★			★

2. Which IQaudIO board is best for me?

Each IQaudIO sound cards has a specific purpose and has been designed to integrate with your existing HiFi / audio or Raspberry Pi equipment. **BEST** audio playback is from our Pi-DAC PRO, Pi-DAC+ and Pi-DigiAMP+ boards. All supporting 24bit depth and 192kHz (Full HD Audio).

	Line Out (RCA)	Balanced (XLR)	STEREO Speakers (2x35w)	MONO Speaker (1.2w)	Head-phones	Aux In 1vRMS	Aux Out 1vRMS	Ext Mic	Built in Mic
Pi-DAC PRO	★	★ (XLR board)			★				
Pi-DAC+	★				★				
Pi-DigiAMP+			★						
Pi-Codec+				★ (MONO)	★	★	★ (3.5mm)	★ (Electret)	★
Pi-CodecZero				★ (MONO)		★ (No Fit)	★ (No Fit)	★ (Electret)	★
Pi-DACZero	★				★ (with HP board)				

Line Out: A double Phono/RCA connector - Normally Red / White in colour. This output is a variable analogue signal (0..2vRMS) and connect to your existing HiFi (pre-amp or amplifier) or can be used to drive Active speakers (with their own amplifier built in).

Balanced Out: An XLR connector - Normally a 3 pin male connector, this is used in Studio and some hi-end hifi systems. It can also be used to drive ACTIVE "monitor" speakers as used at discos or on stage.

STEREO Speakers: Screw terminals - These are normal HiFi speakers (from bookshelf to floor standing), normally with 2 or 3 drive units, where the signal is filtered to each drive unit by an internal cross over within the speaker cabinet. These are known as PASSIVE speakers.

MONO Speaker: Screw terminal - Single drive unit speaker, as found in a transistor radio or similar.

Headphones: 3.5mm jack socket. Normally a tip/ring/ground (3pin headphone) delivering stereo audio to a set of normal headphones. Our headphone amplifiers on the IQaudIO DAC boards can drive up to 80/90ohm impedance headphones. The headphone output of the Codec+ is also aux out.

Aux In: A double Phono/RCA connector or 3.5mm socket. Accepts analogue audio up to 1vRMS input. Can be used to record audio from a variable analogue source such as phone / tape recorder or similar.

Aux Out: A double Phono/RCA connector or 3.5mm socket. Delivers analogue audio up to 1vRMS output. Can be used to feed audio into an amplifier (but at a reduced volume compared to Line Out)

Mic In: a 3.5mm socket for use with an external ELECTRET microphone. Some of our boards have a built in MEMS microphone that is disabled when the external microphone jack is inserted into the 3.5mm socket.

3. Configuring Linux for the IQaudIO sound card

You will find several audio applications with pre-configured Raspberry Pi SD Card images on www.iqaudio.com/downloads - we also include a quickest images which will help you check that you have the IQaudIO sound card installed correctly on your Raspberry Pi.

By default all our latest full size audio boards include a pre-programmed EEPROM. This allows the Raspberry Pi operating system to auto detect and auto configure itself allowing the IQaudIO boards to be plug and play.

If however you are using an older version of our board, or a Pi-Zero type board that does not auto-configure then you may have to edit the Raspberry Pi operating system configuration yourself. This is done by editing the Pi's boot configuration file (found at /boot/config.txt) and ensuring the correct settings for your IQaudIO is included. The onboard audio of the Pi is also disabled through the same config.txt file.

We've included below the entries needed for the IQaudIO boards and also the quickest image that can be downloaded from our website.

The following will help you check your sound card is working and it's configured correctly.

IQaudIO Card	/boot/config.txt	Quicktest img
Pi-DAC+	dtoverlay=iqaudio-dacplus	0_quicktest_buster..zip
Pi-DAC PRO	dtoverlay=iqaudio-dacplus	0_quicktest_buster..zip
Pi-DigiAMP+	dtoverlay=iqaudio-dacplus,unmute_amp or dtoverlay=iqaudio-dacplus,auto_mute_amp	0_quicktest_buster..zip
Pi-Codec+	dtoverlay=iqaudio-codec	0_quicktest_buster..zip
Pi-CodecZero	dtoverlay=iqaudio-codec	0_quicktest_buster..zip
Pi-DACZero	dtoverlay=iqaudio-dacplus	old/IQ_Pi3_quicktestDAC_20180530.zip

Disabling the standard Raspberry Pi On-Board Audio

To make your IQaudIO sound card the primary audio device in Raspbian you will need to disable the Raspberry Pi's on-board "low quality" audio card. This is done simply by commenting out the *dtparam=audio=on* device tree parameter on your sd card's /boot/config.txt file by placing a **#** at the start of the line, saving the file and rebooting your Pi.

/boot/config.txt should have the following entry to disable the Pi's on-board sound card.

    #dtparam=audio=on

4. Headless software

Many of the software packages used with the Raspberry Pi simply turn the Pi into a "Headless" end-point. That is to say that it doesn't need, or expect, a screen/display to be physically attached to the Raspberry Pi's HDMI socket. It is not something to be scared of towards the end of October.
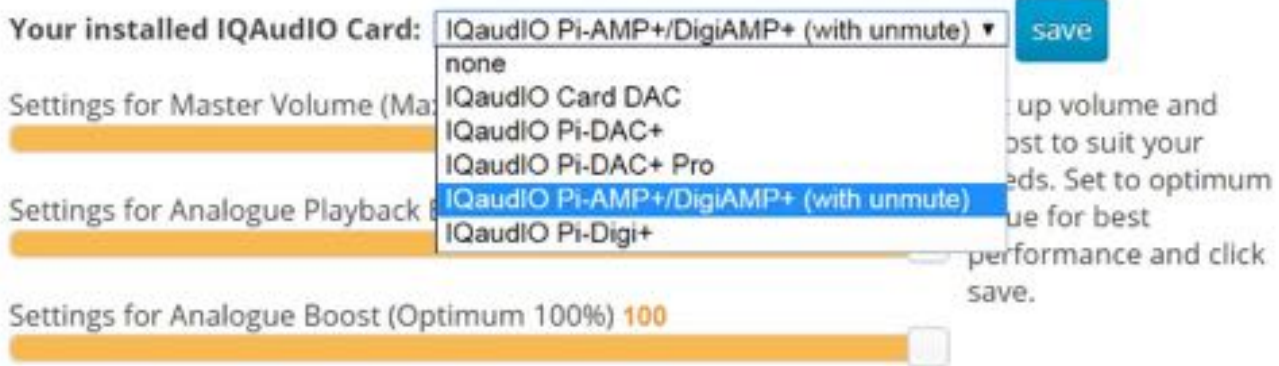
So if there's no screen how do you control it?
Well in some cases you don't need to do anything with the Pi at all. Take the RoonLabs images, that turns the Pi into a Roon Certified end-point. The Pi is then controlled form the RoonLabs PC/Mac application which connects to and plays music through the IQaudIO sound card on your Raspberry Pi. No screen needs to be attached to the Pi itself.

Other "headless" applications install a web server onto the Pi, so to control it you simply open a browser window on your PC / Mac / Tablet or similar and interact with the Pi based application through a web page. Applications such as Max2Play, MoodeAudio, Volumio and RuneAudio work this way.

Max2Play - visit their getting started page here: Max2Play

Volumio:

5. How do I install software on the Pi?

The Raspberry Pi normally loads its software from the micro-SD card. Most audio applications assume they are the only thing running on the Pi so most require a dedicated SD card. The applications are normally distributed as a compressed .img file (likely a .zip or .gz file) and this .img file will need to be written to the Pi's SD card to allow the Pi to boot.

Can't I just copy the .img file over to the Pi?
Err, no. Doing this will just store the .img file on the SD card. The Pi needs the SD card to be formatted in a particular way to allow it to boot / start up and run. Don't worry, there are several ways to do this and some are VERY simple.

Got a Mac? Get ApplePi-Baker - https://www.tweaking4all.com/

Use Windows? Use Win32DiskImager - https://sourceforge.net/projects/win32diskimager/
Top Tip: Buy a few extra micro SD cards, you'll find it much easier to try things. Also, once you have a

working Pi SD card, back it up to your PC/Mac. It's easier to re-write the SD card if things go wrong.

Some newer Raspberry Pi allow the Pi to boot from USB sticks or even over Ethernet.
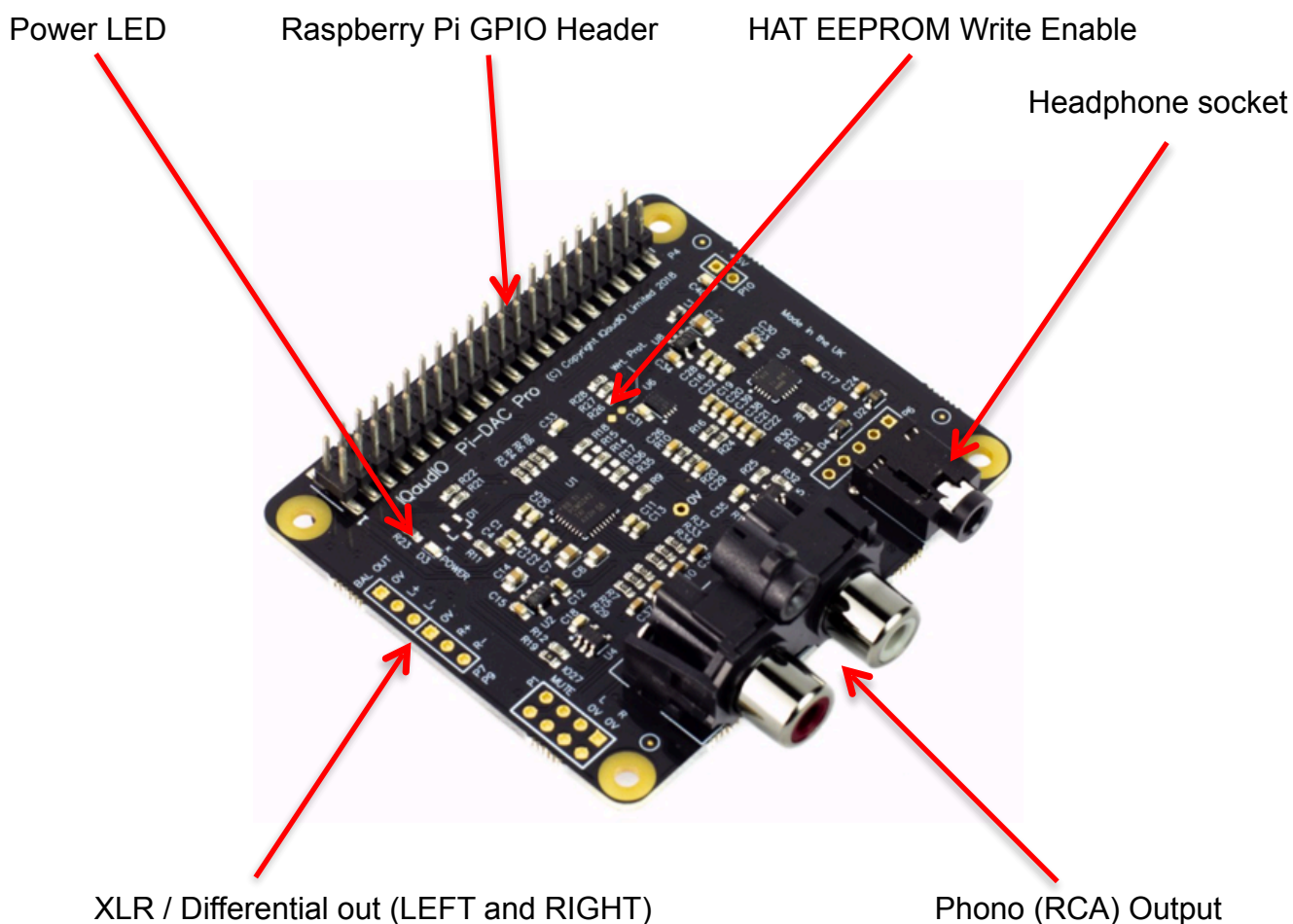
6. The Pi-DAC PRO

The Pi-DAC PRO is a HAT compliant add-on accessory for the Raspberry Pi.

The Pi-DAC PRO users the Texas Instruments PCM5242 to deliver outstanding Signal to Noise Ration and supports Balanced / Differential out in parallel to Phono / RCA line level output. It also includes a dedicated headphone amplifier.

The Pi-DAC PRO can be connected to the Raspberry Pi A+/B+/RPi2/RPi3 or PRi4's 40way pin header without any additional soldering. The Pi-DAC PRO is powered by the Raspberry Pi.

Component selection and track layout have been in the forefront of our design to ensure noise immunity and best possible audio playback with the Pi-DAC PRO.

Power LED   Raspberry Pi GPIO Header   HAT EEPROM Write Enable

Headphone socket

XLR / Differential out (LEFT and RIGHT)   Phono (RCA) Output

The Pi-DAC PRO exposes two 3 pin headers (**P7/P9)** used by the optional XLR board to provide Differential / Balanced output exposed by XLR sockets above the Pi's USB/Ethernet ports.

P1: Analogue out (0..2VRMS), carries GPIO27, MUTE signal (headphone detect), Left and Right Audio and Left and Right Ground.

P6: Headphone socket signals (pin1: LEFT, 2:GROUND, 3: RIGHT, 4:GROUND, 5:DETECT)

P7/9: Differential (0..4VRMS) output (P7:LEFT, P9: RIGHT)

P10: Alternative 5v input, powering the Pi in parallel

7. The Pi-DAC+

The Pi-DAC+ is a HAT compliant add-on accessory for the Raspberry Pi A+/B+/RPi2/RPi3 or PRi4.

The Pi-DAC+ uses the Texas Instruments PCM5122 DAC to deliver analogue audio to the Pi-DAC+'s Phono connectors. The PI-DAC+ also supports a dedicated headphone amplifier.

The Pi-DAC+ can be connected to the Raspberry Pi's 40way pin header without any additional soldering. The Pi-DAC+ is powered by the Raspberry Pi.

Raspberry Pi GPIO header   HAT EEPROM Write Enable   Headphone Socket
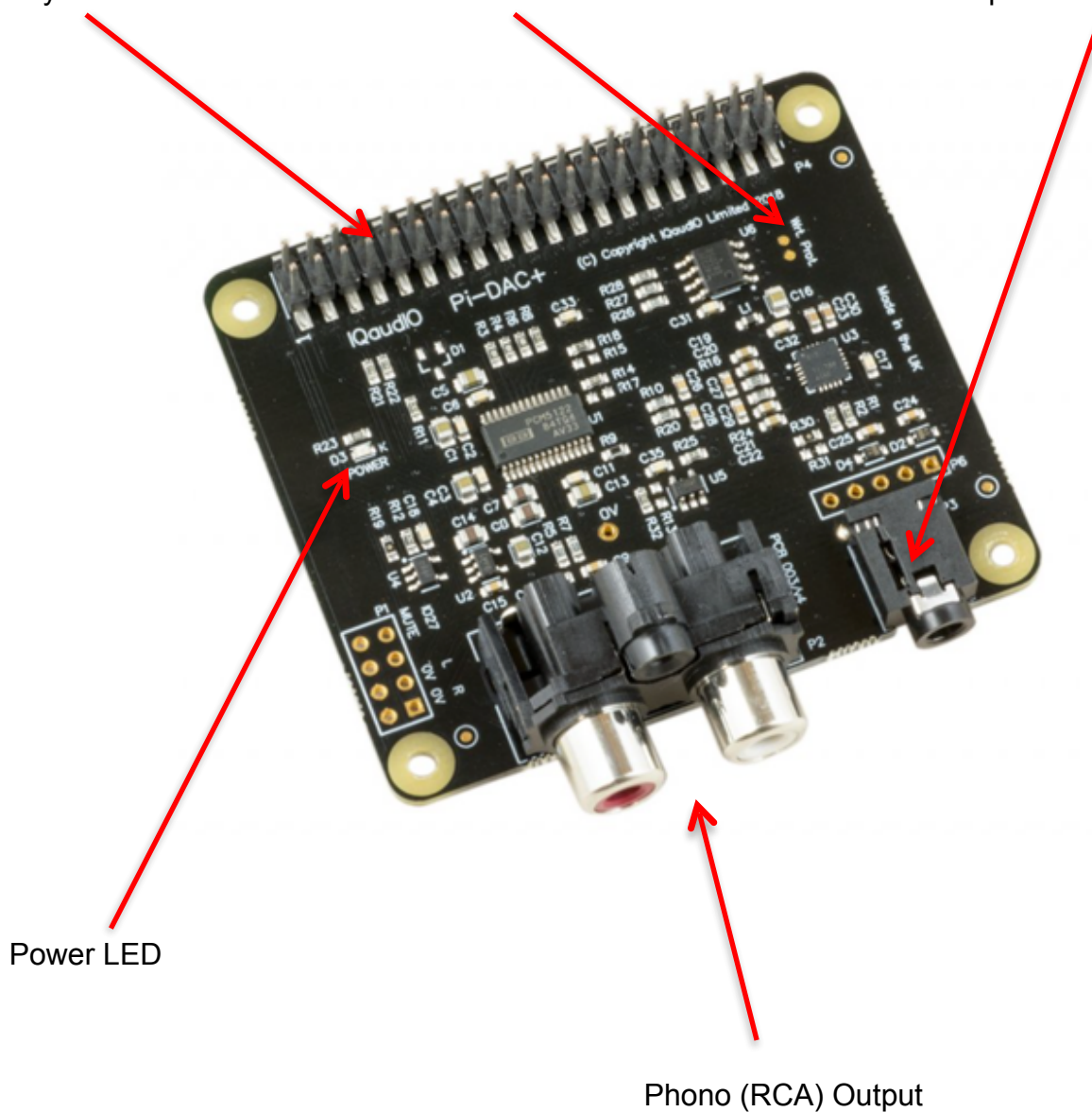
Power LED

Phono (RCA) Output

P1: Analogue out (0..2VRMS), carries GPIO27, MUTE signal (headphone detect), Left and Right Audio and Left and Right Ground.

P6: Headphone socket signals (pin1: LEFT, 2:GROUND, 3: RIGHT, 4:GROUND, 5:DETECT)

8. The Pi-DigiAMP+

The Pi-DigiAMP+ is a HAT compliant add-on accessory for the Raspberry Pi.

The Pi-DigiAMP+ uses theTexas Instruments TAS5756M PowerDAC to deliver direct connection to PASSIVE stereo speakers at up to 2x35wpc with variable output. Ideal for a Pi based HiFi.
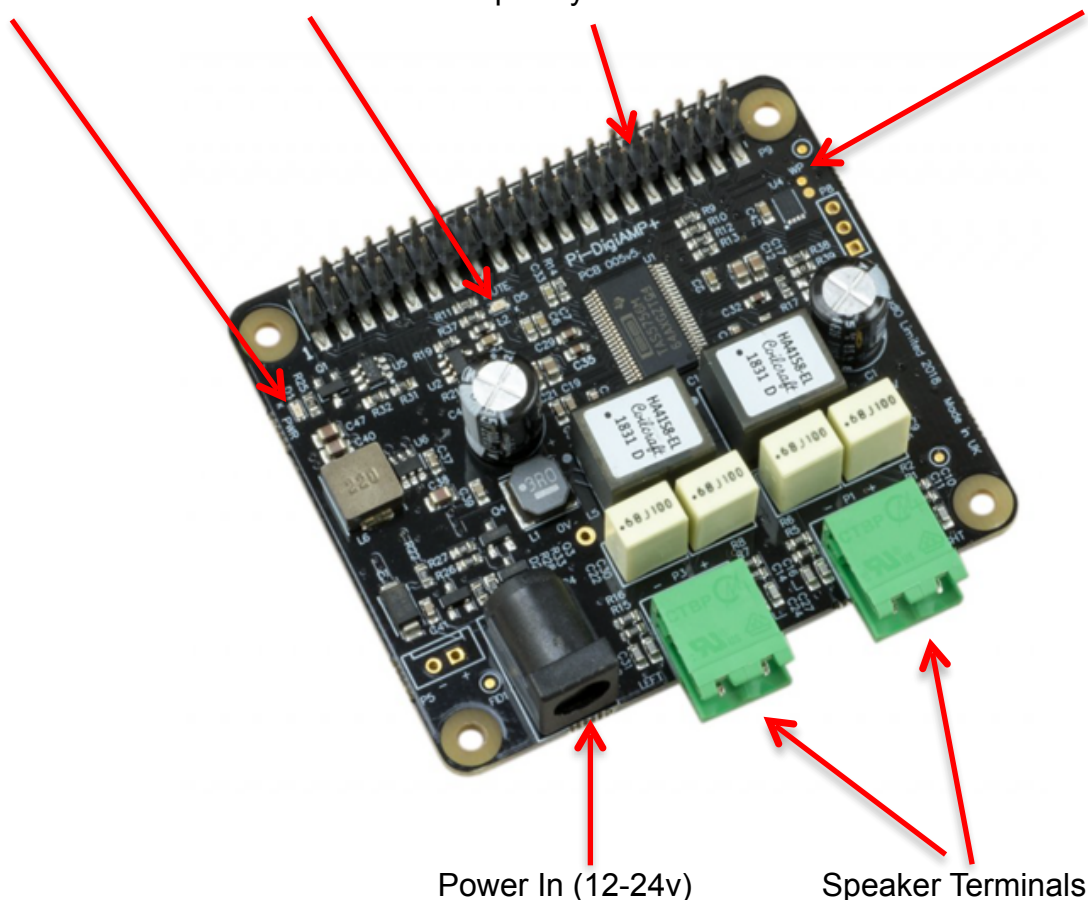
The Pi-DigiAMP+ can be connected to the Raspberry Pi A+/B+/RPi2/RPi3 or RPi4's 40way pin header without any additional soldering.

The DigiAMP+ must be powered from an external supply and requires a 12 to 24DC power source to operate. The Pi-DigiAMP+'s barrel connector is 5.5mmx2.5mm. The Pi-DigiAMP+ has been designed to provide power to the Raspberry Pi and Pi-DigiAMP+ combination in parallel, delivering 5.1v @ 2.5Amp to the Pi.

DO NOT APPLY POWER TO THE RASPBERRY PI's OWN POWER INPUT WHEN USING THE Pi-DIGIAMP+.

The amplifier is by default muted at power on (mute LED lit). The mute state / LED is under software control (Raspberry Pi GPIO22).

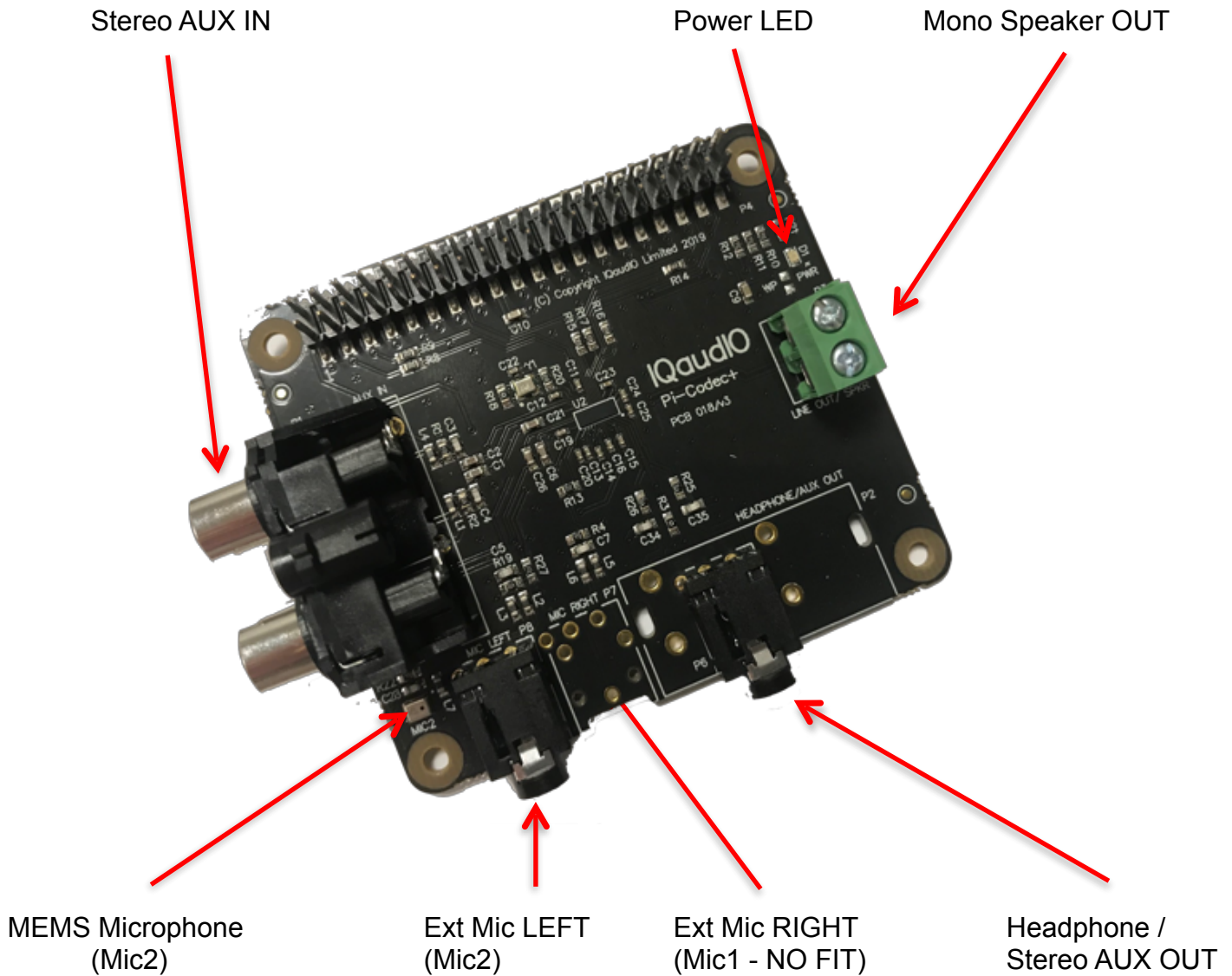Power LED   Mute LED   Raspberry Pi GPIO Header   HAT EEPROM Write Enable

Power In (12-24v)   Speaker Terminals

P5: Alternative power input for hard wired installations. **POLARITY MUST BE OBSERVED!**

P8: TAS5756m Internal GPIO1/2/3

9. The Pi-Codec+

The Pi-Codec+ is a HAT compliant add-on accessory for the Raspberry Pi and can be connected to the Raspberry Pi A+/B+/RPi2/RPi3 or RPi4's 40way pin header without any additional soldering.

The Pi-Codec+ delivers bi-directional digital audio signals (I2S) between the Raspberry Pi and onboard Dialog Semiconductor DA7212 codec. Supporting a range of input and output devices from built in MEMS microphone, AUX IN, dual mono Electret microphones to Stereo AUX OUT, or mono speaker (1.2W / 8Ω)

Stereo AUX IN   Power LED   Mono Speaker OUT

MEMS Microphone   Ext Mic LEFT   Ext Mic RIGHT   Headphone /
(Mic2)   (Mic2)   (Mic1 - NO FIT)   Stereo AUX OUT

10. The Pi-CodecZero

The Pi-CodecZero is a Raspberry Pi Zero sized board but it also includes the HAT's EEPROM allowing for auto configuration of the Linux environment if needed.

The Pi-CodecZero delivers bi-directional digital audio signals (I2S) between the Raspberry Pi and onboard Dialog Semiconductor DA7212. Supporting a range of input and output devices from built in MEMS microphone, mono Electret microphone or mono speaker (1.2W / 8Ω).

It also has GREEN (**GPIO23**) and RED (**GPIO24**) LEDs, that can be used for status and a tactile button (**GPIO27**) that can be used to start an event.

The Pi-CodecZero makes a great starting point for a walkie talkie / intelligent doorbell, vintage radio hack or Smart type speaker.
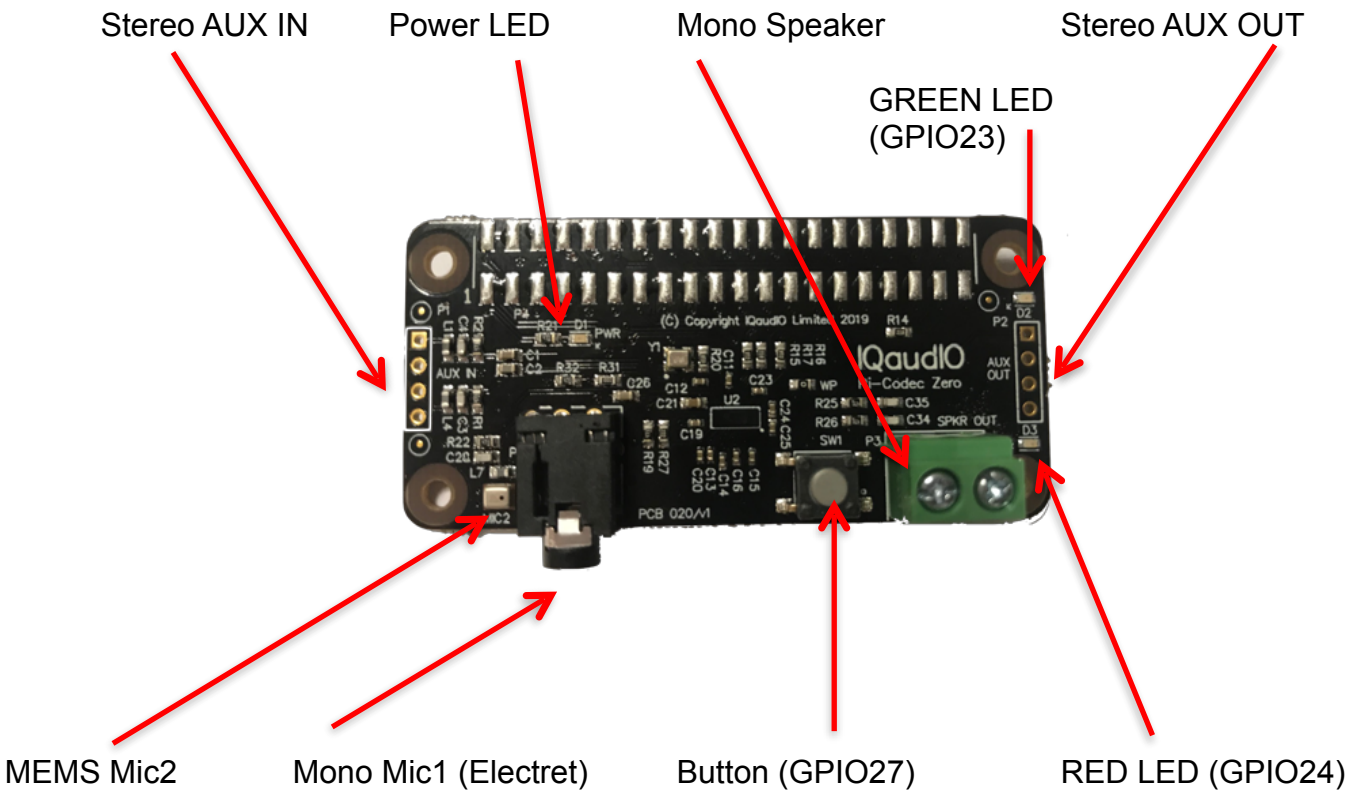
P1 and P2 will support external PHONO / RCA sockets if needed. P1: AUX IN, P2: AUX OUT.
Pin 1 is square:
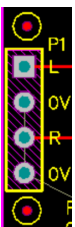

1: Left
2: Ground
3: Right
4: Ground

11. The Pi-DACZero / DACZero Headphone

The Pi-DACZero is a smaller version of the IQaudIO Pi-DAC+ which is designed to be used with the Raspberry Pi-Zero.

The Pi-DACZero delivers audio Line Out and uses the same Linux drivers as the Pi-DAC+.

There is an optional Headphone daughter card for the Pi-DACZero which can be mounted above the DACZero to give an amplified headphone output in parallel. The Pi-DACZero and Pi-DACZeroHeadphone boards are powered from the Raspberry Pi-Zero itself.

Rotary Encoder Header (P3)
Pin1:GPIO23, Pin2:GROUND, Pin3:GPIO24

Daughter card header (P1)

Power LED

Phono (RCA) Output
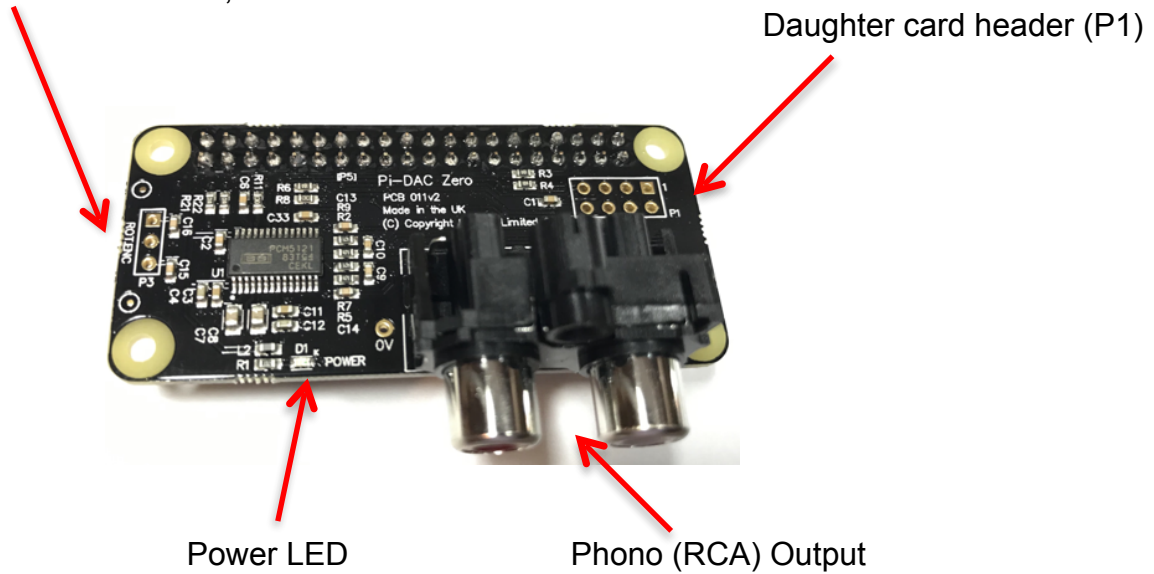
| Signals passed through by Pi-DACZero (P1)					
Pin#	NAME			NAME	Pin#
1	**RIGHT OUT**	□	¤	(GROUND) **0v**	2
3	**LEFT OUT**	¤	¤	(GROUND) **0v**	4
5	3**v3** (DC Power)	¤	¤	(DC Power) **5v**	6
7	**GPIO27**	¤	¤	GPIO22	8

DACZero Headphone

Power LED

3.5mm Stereo Headphone Jack
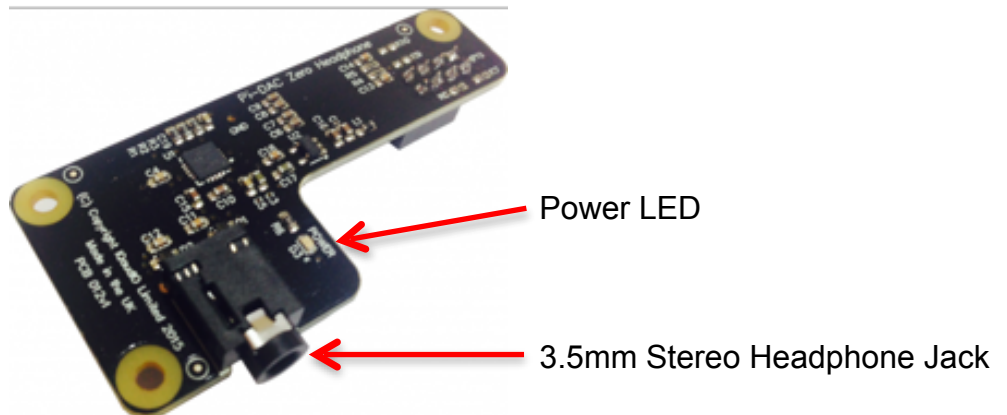
12. CosmicController

The CosmicController delivers additional functionality to the Raspberry Pi delivering 3x momentary buttons, 3x Super bright White LEDs, 1x Rotary Encoder with push button, optional IR sensor and optional I2C based OLED display. The CosmicController is supported by Bob Rathbone's radio software, and sample usage code is also available on out GitHub page (www.github.com/iqaudio)

Button1 (GPIO 4)   Button2 (GPIO 5)   Button3 (GPIO 6)   Rotary Encoder (GPIO 23/24)
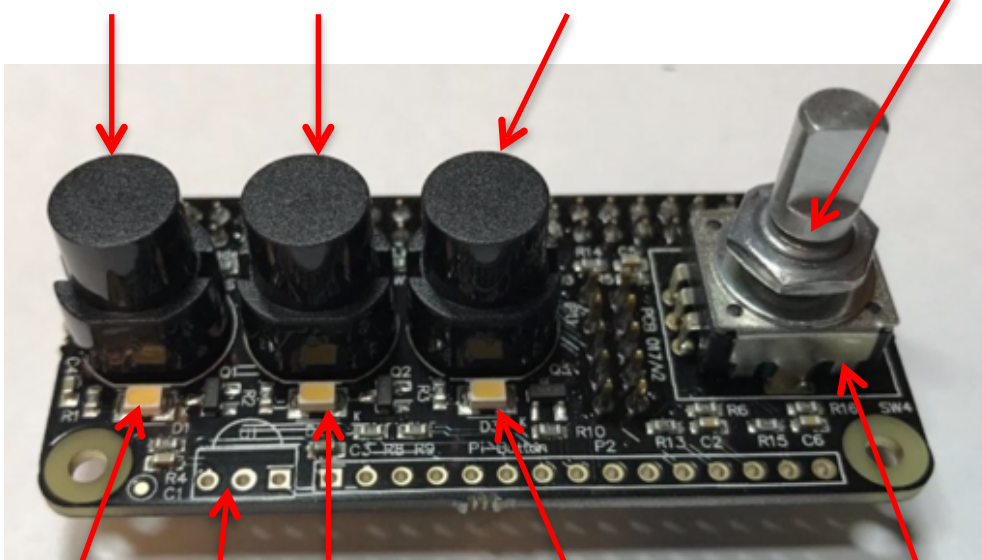
LED1 (GPIO 14)   LED2 (GPIO 15)   LED3 (GPIO 16)   Encoder Switch (GPIO 27)

OPTIONAL IR SENSOR (GPIO25)

The CosmicController also supports the optional OLED display from OLIMEX. The OLED is connected to the CosmicController via a 10way ribbon connector. The orientation of this cable **must be observed**. If should ONLY be connected as shows below:

PIN1 (MARKED RED)

13. IQaudIO KITS

Many of the IQaudIO boards are available in KIT form. These allow users to use alternative GPIO pin headers, hard wire cables to replace Phono/RCA sockets etc. Other kits (Cosmic Controller / XLR kit) add additional functionality to the Pi or IQaudIO sound cards.
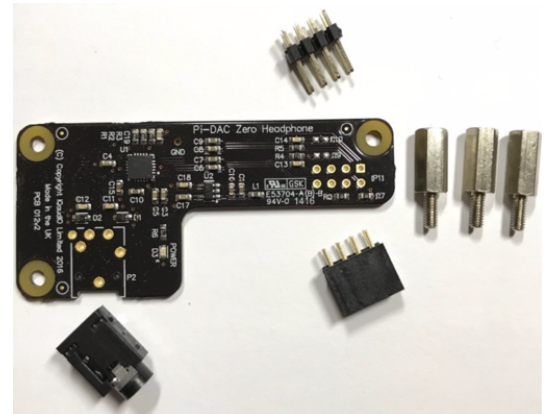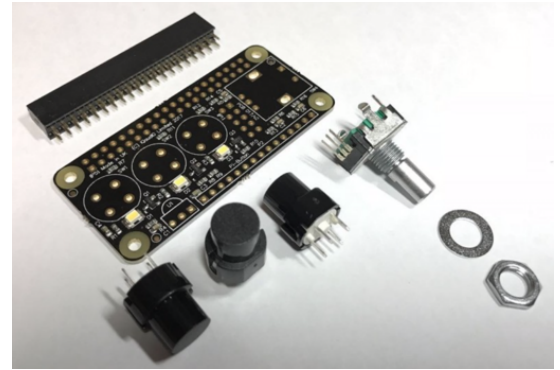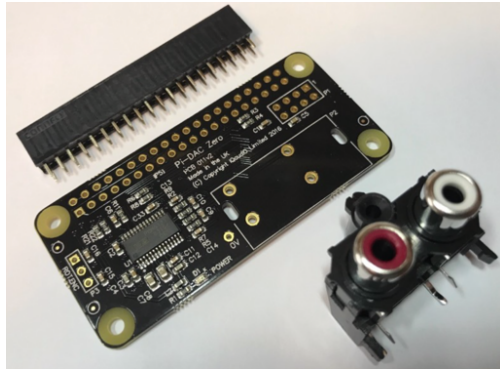
14. Adding the IQaudIO sound card to your Raspberry Pi

WARNING: PLEASE ENSURE ALL POWER CABLES ARE DISCONNECTED FROM THE RASPBERRY PI BEFORE ATTEMPTING THIS PROCEDURE.

The IQaudIO range of sound cards attach to the Raspberry Pi's 2x20 way pin header (GPIO header). They are designed to be supported on the Pi using the supplied PCB standoffs and screws. In general no soldering is required to the IQaudIO boards for normal operation.

The IQaudIO range of boards are normally supplied with the necessary mounting hardware (PCB spacers and screws), these are for our latest low profile boards 4x 9mm PCB spacers and 8x M2.5 screws. If additional parts (such as 2piece speaker connectors) are required these are normally provided too.
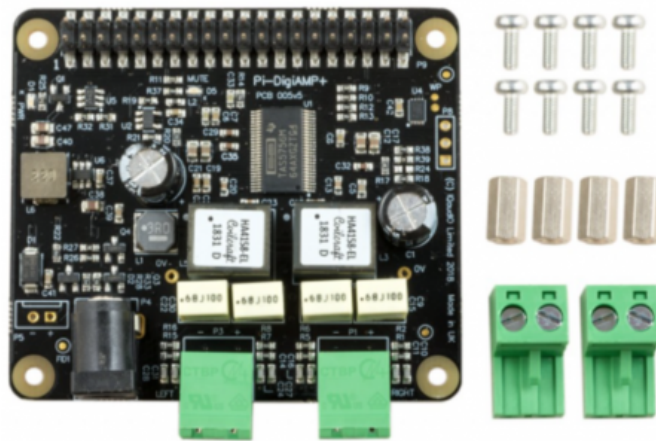
The remaining screws are then screwed into the spacers from above.
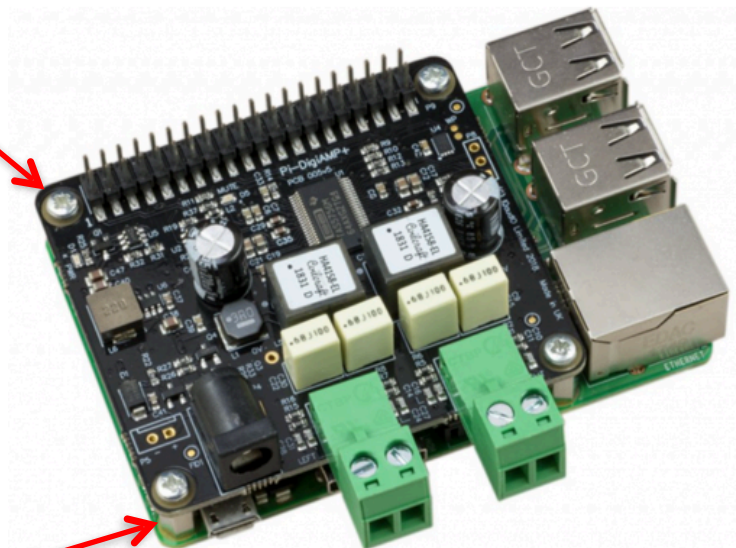
The PCB spacers should be screwed (finger tight only) to the Raspberry Pi before adding the IQaudIO board.

If you are using the Pi-CASE+ replace the 4x screws beneath the Pi with the 4x 5mm PCB standoffs provided with the Pi-CASE+, the 4 screws are then used to secure the bottom of the case to the short PCB spacers beneath the Pi.
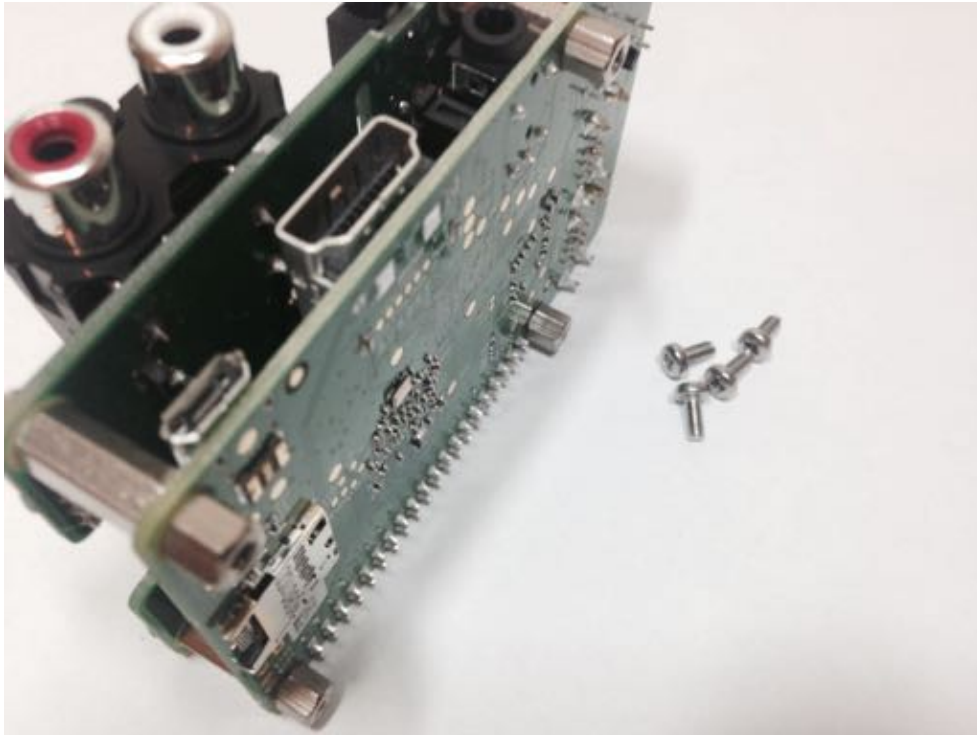
Other IQaudIO boards

Building the smaller Pi-Zero sized boards to the Raspberry Pi-Zero follows the same process, as does adding the optional Pi-DAC Zero Headphone or Pi-DAC PRO XLR daughter cards. Daughter cards are normally supplied with male / female pcbs spacers to replace the top most screws and stack the PCBs correctly.

15. Building the IQaudIO acrylic case

There are a number of project cases available for the Raspberry Pi, the addition of accessory boards, such as the IQaudIO sound cards, requires a specific case. We have the following cases available in a range of colours and finishes:

Pi3-CASE
- Supplied with a range of sides to suite our boards.
- For the Raspberry Pi3 and latest Pi-DAC+ / Pi-DAC PRO / Pi-DigiAMP+

Pi3-XLR CASE
- For the Raspberry Pi3, Pi-DAC PRO and XLR board

Pi-DACZERO CASE
- For the Raspberry Pi Zero and Pi-DAC Zero / Pi-DAC Zero Headphone daughter card

Pi "A" CASE
- ONLY suitable for the Raspberry Pi 3A+
- Supports our latest Pi-DAC+ / Pi-DAC PRO and Pi-DigiAMP+ boards

GENERAL BUILD INSTRUCTIONS

The following may prove helpful when building your Raspberry Pi and IQaudIO sounds card into the appropriate Pi-CASE+.

All acrylic CASE+ installations follow the same process:
- If there is a protective film applied to either side of your acrylic, **remove it before assembly.** If you build the CASE+ whilst retaining the protective film you may find that the finished look is spoilt or for transparent cases, the result may be opaque.
- Bolt the PCBs together before attempting to install the bundle in the case. Make sure the small 5mm female/male spacers are fitted to the bottom of the Raspberry Pi
- Build the case sides around the PCB first (tip: use tape on each corner during the build process)
- Add the bottom and top
- Add the bolts and small rubber bumper stickers

And just so we don't get more emails telling us our manufacturing process is wrong, that we've cut the material mirrored or the look and finish is spoilt with the text "perspex".... :-)
- If there is a protective film applied to either side of your acrylic, **remove it before assembly!**

NOTE: Access to the Raspberry Pi micro SD card can be tricky due to the width of the Perspex. We suggest the use of tweezers.

IQaudIO Case parts / Assembly

Each case is supplied cut out and includes:
- 4x Long bolts
- 4x Large nuts
- 4x 5mm male / female PCB spacers
- 1x Top Plate
- 1x Bottom Plate
- Several sides (There are normally 3 sides which are used for all IQaudIO boards and a set of sides specific to the IQaudIO board being used.)
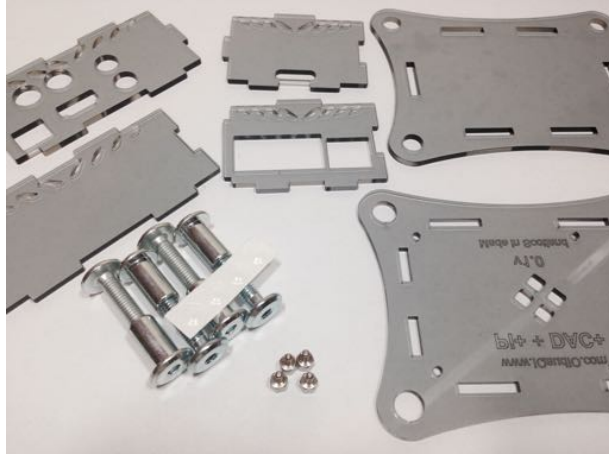
A. Lay out the CASE+ sides corresponding to the Raspberry Pi & the specific IQaudIO board
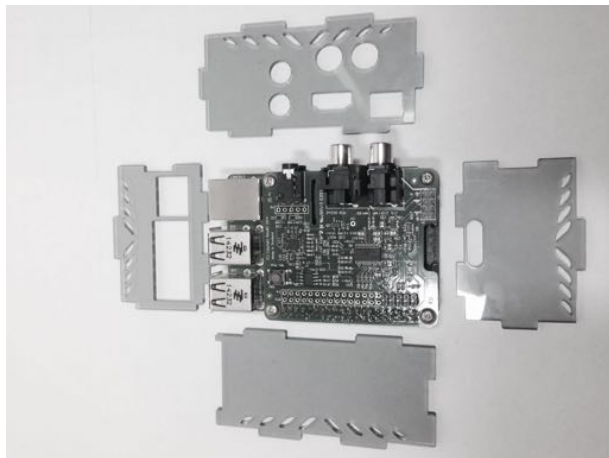
B. Then, build the sides up around the Raspberry Pi. Use sticky tape at each corner to make things simpler to assemble.
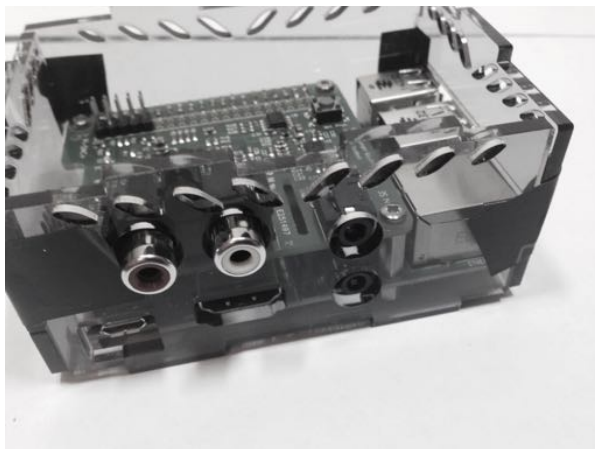
C. Add the base (with text outwards) and screw the case to the Raspberry Pi

D. Pi-CASE+ top in place

Now add the 4 nuts and bolts – Do not over tighten, finger tight is OK

E. Completed Pi-CASE+ with Raspberry Pi and Pi-DAC+ inside

F. Now apply the small transparent bumper stickers to the base of each Bolt.

G. Now to complete screw the remaining 4 screws from the bottom of the case into the 5mm spacers inside.

16. Configuring Linux to support IQaudIO sound card

Generally the full size IQaudIO boards auto configure Linux as they have the necessary EEPROM data flashed at production, therefore no user configuration is necessary.

The below information may however help with any diagnostics and also to support the configuration of legacy boards or Pi-Zero boards that don't have an on-board EEPROM (Pi-DAC Zero for example).

This section covers the use of DeviceTree based driver support – as used in the Raspbian releases from late 2015.

This guide assumes you are using the standard Raspbian distribution from here: http://www.raspberrypi.org/downloads/ and that you have an SD card flashed with Raspbian and have a login to your Raspberry Pi (either local or via ssh). You should have also run raspi-config and expanded the filesystem to occupy your full SD card.

- Update Raspbian on the SD card to be the latest available.
  ```
  pi@raspberrypi ~ $ sudo rpi-update
  pi@raspberrypi ~ $ sudo apt-get update
  pi@raspberrypi ~ $ sudo apt-get dist-upgrade
  ```

- To ensure that the IQAUDIO Device drivers are loaded make sure /boot/config.txt file has the valid entry below.

  a. For Pi-DAC+, Pi-DACZero, Pi-DAC Pro and Pi-DigiAMP+ use the iqaudio-dacplus
  b. For Pi-Codec+, use iqaudio-codec

  ```
  pi@raspberrypi ~ $ sudo nano /boot/config.txt
  ```

dtoverlay=iqaudio-dacplus

or

dtoverlay=iqaudio-codec

It is possible to control the QaudIO Pi-DigiAMP+'s "mute state" with additional optional parameters:

dtoverlay=iqaudio-dacplus,**unmute_amp**

This will give a "one-shot" unmute when kernel module loads.

dtoverlay=iqaudio-dacplus,**auto_mute_amp**

This will unmute the amp when ALSA device opened by a client. Mute, with 5 second delay when ALSA device closed. (Re-opening the device within the 5 second close window, will cancel mute.)

- To remove the standard Raspberry PI audio card, comment out the {dtparam=audio=on} device tree parameter in your /boot/config.txt and reboot.
  ```
  pi@raspberrypi ~ $ sudo nano /boot/config.txt
  # Enable audio (loads snd_bcm2835)
  #dtparam=audio=on
  ```

- Flush your filesystem changes and reboot the Raspberry Pi
  ```
  pi@raspberrypi ~ $ sync
  pi@raspberrypi ~ $ sudo reboot
  ```

Once restarted SSH into the Pi and check that the audio drivers / card is available to ALSA.

```
pi@raspberrypi ~ $ aplay -l
```

If the IQaudIO board and drivers have been installed correctly you should see something similar to:

```
pi@iqaudio ~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: IQaudIODAC [IQaudIODAC], device 0: IQaudIO DAC HiFi pcm512x-hifi-0 []
  Subdevices: 0/1
  Subdevice #0: subdevice #0
```

17. Check audio is working

Once support is active using "devicetree", and that the underlying Linux ALSA sound system is configured you should now be able to listen to music.

After the Raspberry Pi boots, log in and check that the IQaudIO board is visible as an ALSA audio card by running "aplay –l", (that's a lower case L and ignore the quotes). The output should be similar to that below. Note the **device** number and **Subdevice** numbers – these may be useful to you when configuring media applications to access the sound card in your system.

> pi@iqaudio ~ $ aplay -l
> **** List of PLAYBACK Hardware Devices ****
> card 0: IQaudIODAC [IQaudIODAC], **device 0**: IQaudIO DAC HiFi pcm512x-hifi-0 []
>   Subdevices: 0/1
>   **Subdevice #0**: subdevice #0

Texas Instruments based IQaudIO boards

For our Texas Instruments based boards (**Pi-DAC+ / Pi-DAC PRO / Pi-DigiAMP / Pi-DAC Zero**), check that you can adjust the volume. From the Raspberry Pi run alsamixer. If you have more than one sound card installed you may have to select which is being controlled by pressing the F6 key.

> pi@raspberrypi ~ $ alsamixer
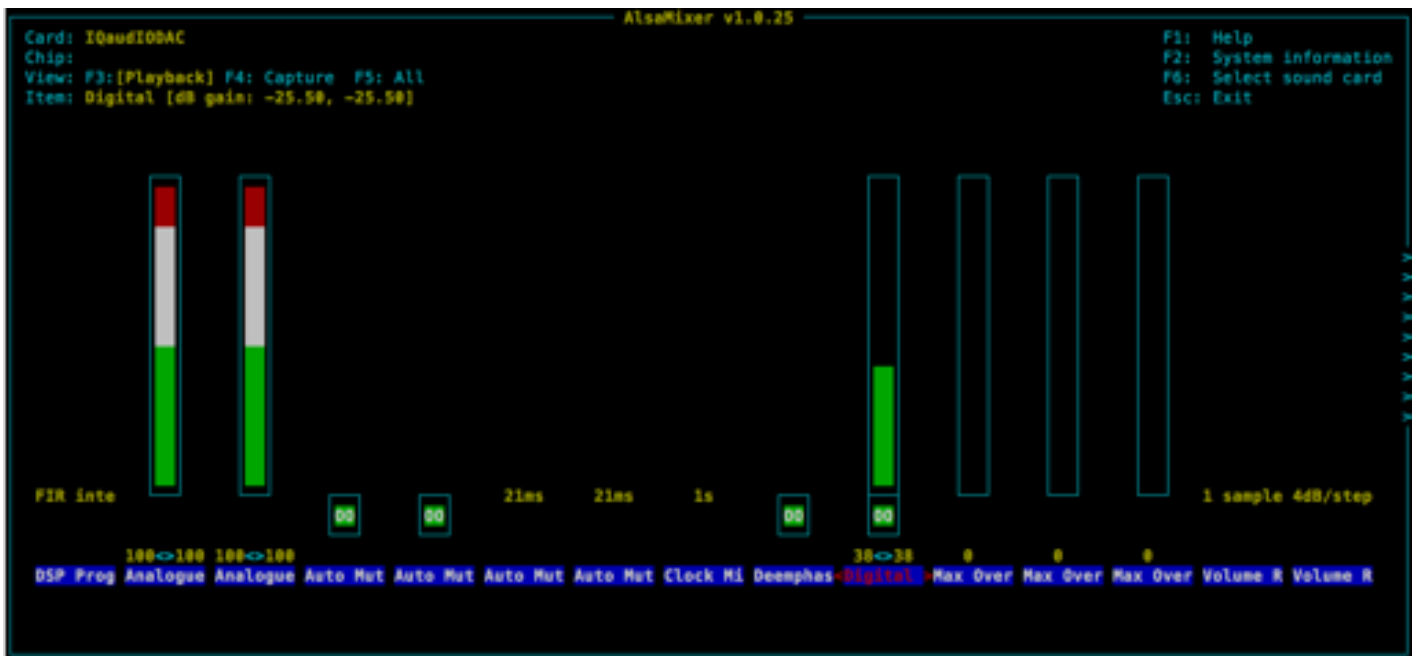

Now adjust the volume by highlighting "Digital" (Press your cursor keys left and right to select) and increase / decrease the volume by pressing up and down cursor keys.
Remember 100% = 0db, 0% is mute!

NOTE: Both mixers ANALOGUE and ANALOGUE PLAYBACK BOOST should be set to 100% to deliver the full 2.1vRMS output at the RCA socket / 4.2vRMS at the XLR socket on the Pi-DAC PRO

Please note that there are now, from the Feb 2015 Raspbian releases, 3 volume controls for the Pi-DAC+ / Pi-DigiAMP+. This can be confusing but for simplicity just ensure that the Analogue and Analoge Playback Boost volume controls are set to Maximum (100%)

- "Digital" is the master volume control 0..100%
- "Analogue" sets the maximum output to be 1vrms (0% = +0db) OR 2vrms (100% = +6db gain)
- "Analogue Playback Boost" gives an additional gain of 0.8db and should similarly be set to 100%
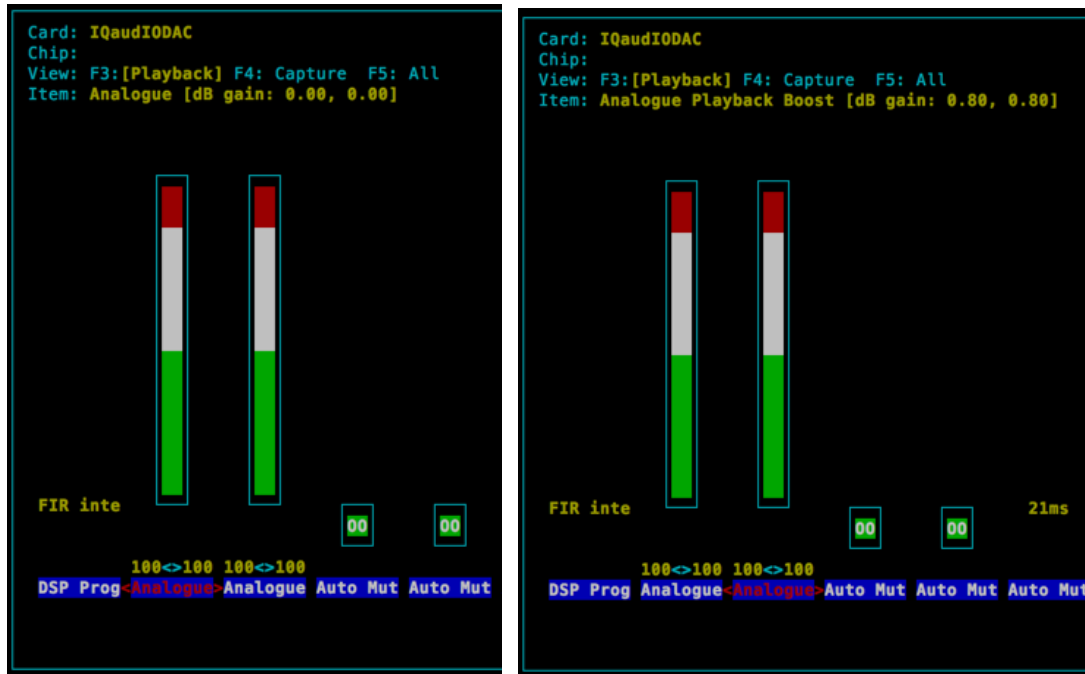
Test that audio can be heard through the Pi-DAC+. Use the ALSA speaker-test utility and listen for "Front Left / Front Right". Make sure your volume level (Digital mixer above) is not at 100% before doing this.

```
pi@raspberrypi ~ $  speaker-test –c 2 –t wav
```

Alternatively, listen to an internet based radio station…
Notice the hw parameter which points to device 0, subdevice 0 in the example below. Change these values depending on the specific output of the aplay –l utility above.

```
pi@raspberrypi ~ $  mplayer –nolirc –ao alsa:device=hw=0,0 –playlist http://
network.absoluteradio.co.uk/core/audio/mp3/live.pls?service=vrhq
```

If you don't have mplayer already installed then this can be done using:
    pi@raspberrypi ~ $ sudo apt-get update
    pi@raspberrypi ~ $ sudo apt-get install mplayer

Dialog Semiconductor based IQaudIO boards

Our Dialog Semiconductor based boards (**Pi-Codec+ / Pi-CodecZero**) are general purpose sound cards, playing back stereo audio but also recording audio too.

As it's a general purpose audio card with multiple inputs and output options and several volume controls the basic configuration is more complicated than our standard playback only products. The information below can be used to check that audio playback is working as expected.

CONFIGURE THE CODEC BOARD TO PLAYBACK AUDIO THROUGH THE AUX OUT AND MONO SPEAKER.

From the Raspberry Pi's command line, run alsamixer. If you have more than one sound card installed you may have to select which is being controlled by pressing the F6 key.

Note the following labelling differences between the software control and board silk screen.
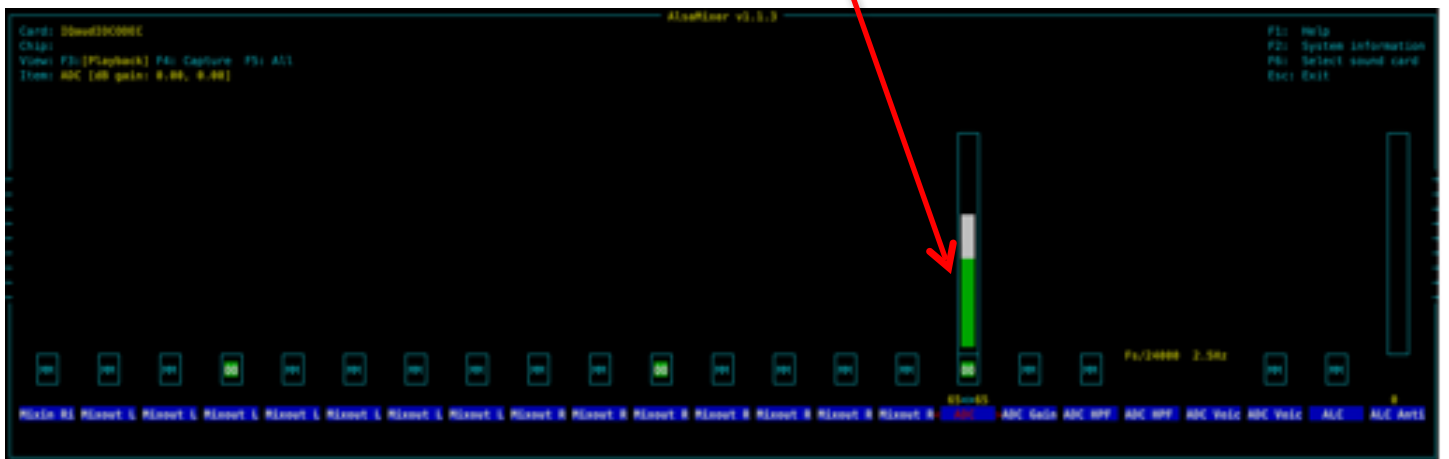
a) Headphone = AUX OUT (volume)
            b) Lineout = MONO SPEAKER (volume)
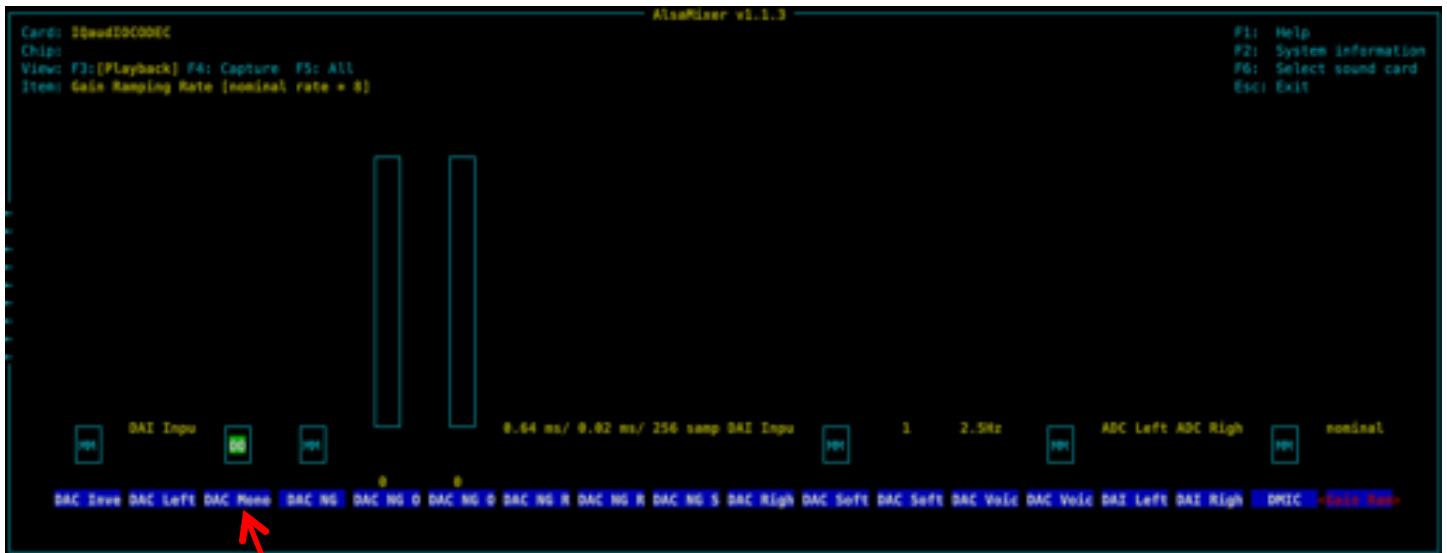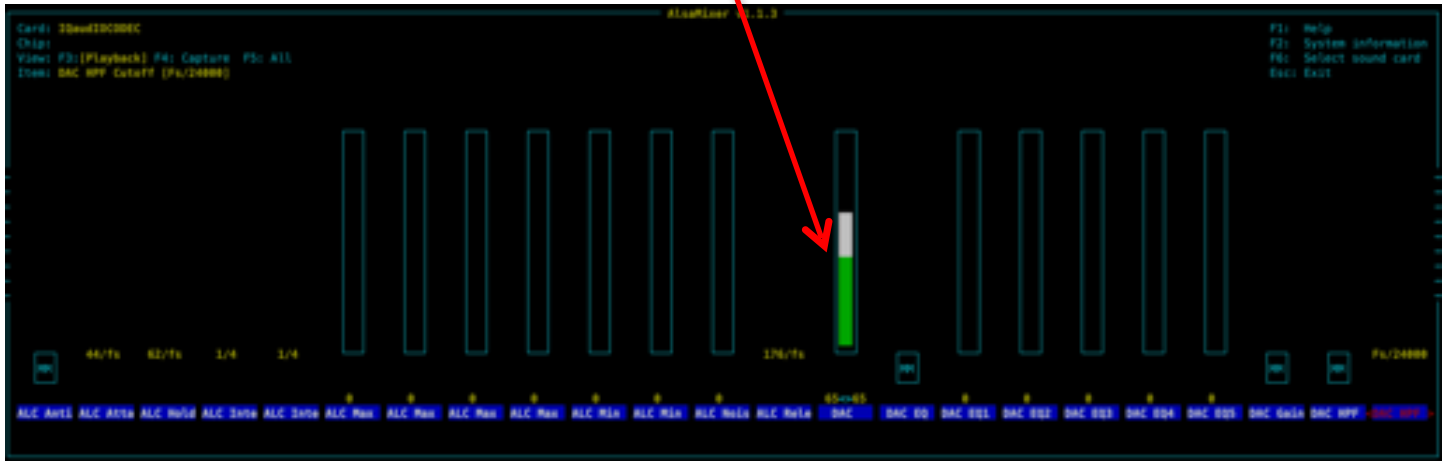   c) AUX = AUX IN (recording level)
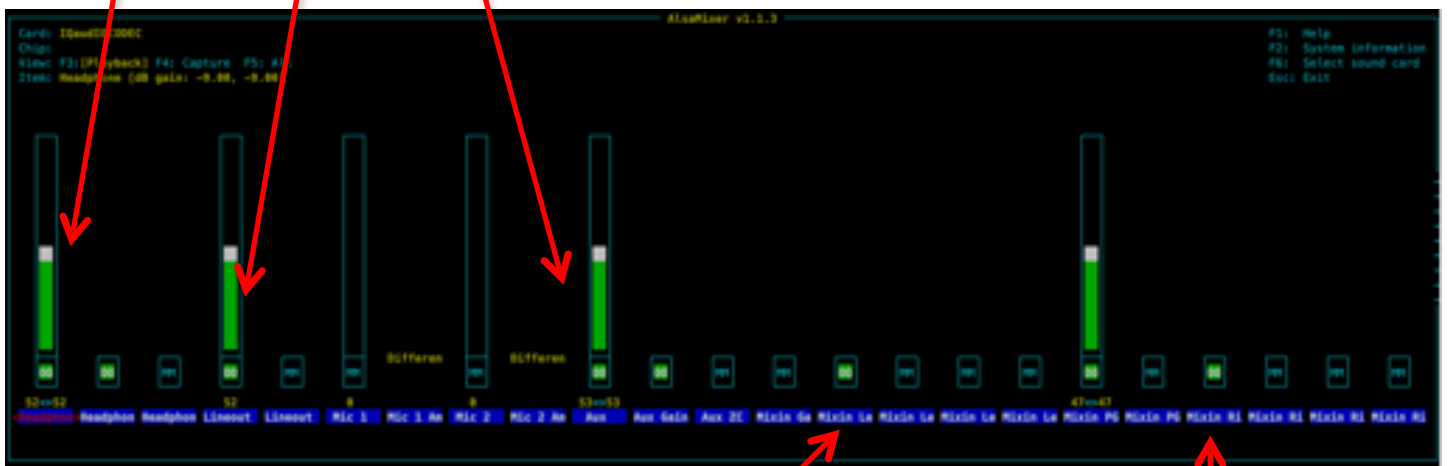
Enable AUX IN LEFT channel   Enable AUX IN RIGHT channel

Enable Analog Digital Convertor (ADC) & Mixer level

Enable Digital Analogue Convertor (DAC) & Mixer level

Set MONO output

Test that audio can be heard through the Pi-CODEC+. Use the ALSA speaker-test utility and listen for "Front Left / Front Right". Make sure your AUX OUT and SPEAKER volume levels are not at 100% before doing this.

```
pi@raspberrypi ~ $  speaker-test -c 2 -t wav
```

Alternatively, listen to an internet based radio station…
Notice the hw parameter which points to device 0, subdevice 0 in the example below. Change these values depending on the specific output of the aplay –l utility above.

```
pi@raspberrypi ~ $  mplayer -nolirc -ao alsa:device=hw=0,0 -playlist http://
network.absoluteradio.co.uk/core/audio/mp3/live.pls?service=vrhq
```

If you don't have mplayer already installed then this can be done using:
        pi@raspberrypi ~ $ sudo apt-get update
        pi@raspberrypi ~ $ sudo apt-get install mplayer

18. Pi-Codec+ / Pi-CodecZero Configuration

The IQaudIO Pi-Codec boards uses the Dialog Semiconductor DA7212 codec. The DA7212 allows the recording of audio from the board's built in MEMS microphone, from stereo Phono sockets (AUX IN), 2x mono external Electret microphones and playback through stereo Phono sockets (AUX OUT) and mono speaker connector.

Each input and output device has it's own "mixer" allowing the audio levels / volume to be adjusted independently. Within the codec itself other mixers / switches exist to allow the output to be Mono'd for single speaker output, signals may also be inverted and there is a 5 band Equaliser to adjust certain frequency bands. These settings can be controlled through alsamixer interactively or programatically.

It is important to note that the AUX IN and AUX OUT are both 1vRMS. It may be necessary to adjust the AUX IN's mixer to ensure the input signal doesn't saturate the ADCs. (Analogue to Digital Convertors). Similarly the output mixers may been to be adjusted to get the best possible output.

Pi-Codec+ features:
• High performance 24-bit audio codec
• Supports common audio sample rates between 8-96kHz
• Built in MEMS microphone (Mic2)
• Mono Electret microphone (Mic2 left)
    • Automatic MEMS disabling on Mic2 insert detect
• Supports additional (no fit) Mono Electret microphone (Mic1 right)
• Stereo auxiliary input channel (AUX IN) - PHONO/RCA connectors
• Stereo auxiliary output channel (Headphone / AUX OUT) - 3.5mm Stereo jack
• Flexible analog and digital mixing paths
• DSP for ALC (Automatic Level Control)
• 5-band EQ (Equalisation)
• Mono line-out / mini speaker driver: 1.2W @ 5V, THD<10%, R=8Ω

We've created a set of preconfigured scripts (loadable ALSA settings) on our GitHub site. https://github.com/iqaudio/Pi-Codec - These cover several use cases such as:

• Mono MEMS mic recording, mono speaker playback

• Mono MEMS mic recording, mono AUX OUT playback

• Stereo AUX IN recording, stereo AUX OUT playback

• Stereo MIC1/MIC2 recording, stereo AUX OUT playback

NOTE: THE CODEC BOARD'S DA7212 CHIP NEEDS TO KNOW WHAT IS INPUT AND WHAT IS OUTPUT AT EACH POWER CYCLE FOR IT TO OPERATE SUCCESSFULLY. We suggest your application does this at startup or you add a suitable configuration to the /etc/rc.local file. See the FAQ section of this document on how to store/save or restore ALSA settings.

19. Controlling the Pi-DigiAMP+'s Mute / Un-mute

The DigiAMP+'s MUTE state is toggled by GPIO22 on the Raspberry Pi. The latest IQaudIO device tree supports the unmute of the DigiAMP+ through additional parameters (from Raspbian 4.4.14+ onwards) …

dtoverlay=iqaudio-dacplus,unmute_amp
"one-shot" unmute when kernel module loads.

dtoverlay=iqaudio-dacplus,auto_mute_amp
Unmute amp when ALSA device opened by a client. Mute, with 5 second delay when ALSA device closed. (Re-opening the device within the 5 second close window, will cancel mute.)

If you do not want to control Mute state through device tree then you can also script your own solution. The information below may be useful.

```
# The amp will startup MUTED
# to unmute the amp.
pi@raspberrypi ~ $ sudo sh -c "echo 22 > /sys/class/gpio/export"
pi@raspberrypi ~ $ sudo sh -c "echo out >/sys/class/gpio/gpio22/direction"
pi@raspberrypi ~ $ sudo sh -c "echo 1 >/sys/class/gpio/gpio22/value"

# to mute the amp once more...
pi@raspberrypi ~ $ sudo sh -c "echo 0 >/sys/class/gpio/gpio22/value"
```

If you already use WiringPi:

```
# If using WiringPi - to unmute the amp
gpio mode 3 out
gpio write 3 1

# to mute once more
gpio write 3 0
```

20. Raspberry Pi Audio applications

We continue to work with the premier Raspberry Pi audio/media developers (both commercial and open source). Some such applications are downloadable and configurable direct from the software vendors themselves (Max2Play / MoOdeAudio / Volumio, etc) others (RoonLabs / Shairport-sync / etc.) are available pre-configured for download to sd card from our download page www.iqaudio.com/downloads

If you would like to see additional software made available then please email info@iqaudio.com with the specific details.

RoonLabs

Roon Ready Certified IQaudIO releases (for Pi-DAC+, Pi-DAC PRO, Pi-DigiAMP+ and Pi-Digi+) are available as pre-configured SD card images from our web site's download page www.iqaudio.com/downloads

Please see the dedicated RoonLabs setup (section 21) at the end of this document.

Max2Play IQaudIO Special Edition

IQaudIO Special Edition - (https://www.max2play.com/en/max2play-image/) support is built in - you can install all IQaudIO cards with just one click.

IQaudIO Max2Play Image First Steps

After connecting the Raspberry Pi to the power supply, you can choose to connect to Max2Play in three different ways:

1.   Connect an ethernet cable to the Raspberry Pi.

2.   Use the WPS feature of your router. More information

3.   Use the Access Point Autostart Mode. More Information

After connecting in one of these three ways, you are greeted by the Max2Play Start Page. You can now select IQaudIO as the sound card you are using.
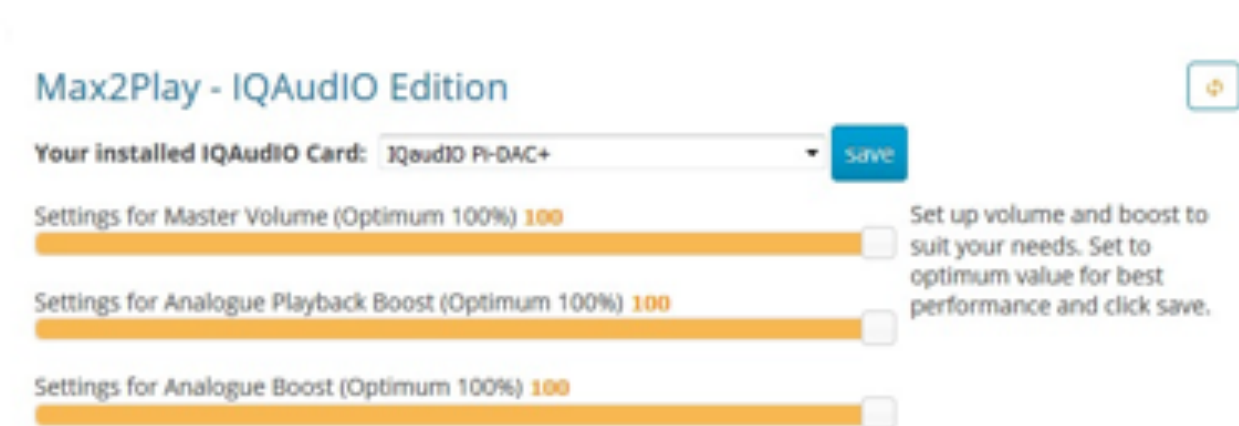
Additional Settings for IQAudIO-Cards

Rotary Encoder   **IR-Control**

Use the rotary encoder as a hardware volume control for your IQAudIO Card. The installer sets up everything and adds software to Autostart on Boot.
When using the rotary encoder with Squeezelite, make sure to use hardware volume control and additionally set the Alsa-Sync option in command line options of Squeezelite:
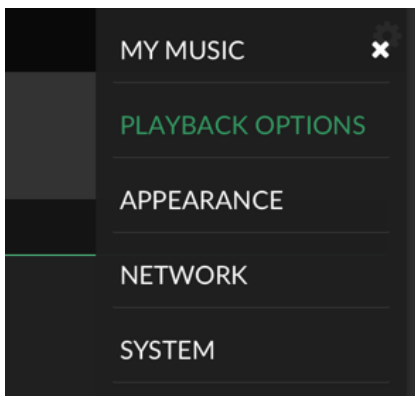
`-o hw:CARD=IQaudIODAC -V Digital -Q`

[Install Rotary Encoder Software]

Autostart Rotary Encoder Software: ☐ [save]

After selecting your sound card from the dropdown menu and rebooting the system, all settings are already made for you and you can choose the audio player you would like to use.

MY MUSIC	✖
PLAYBACK OPTIONS	
APPEARANCE	
NETWORK	
SYSTEM	

If you have a Rotary Encoder, you can also directly install it with one click in the same menu and just add the hardware volume control line to the Advanced Options' command lines of the audio player you would like to use.

For more guidance and to follow along the settings, you can also check out the YouTube video on the IQaudIO Plugin by Max2Play.
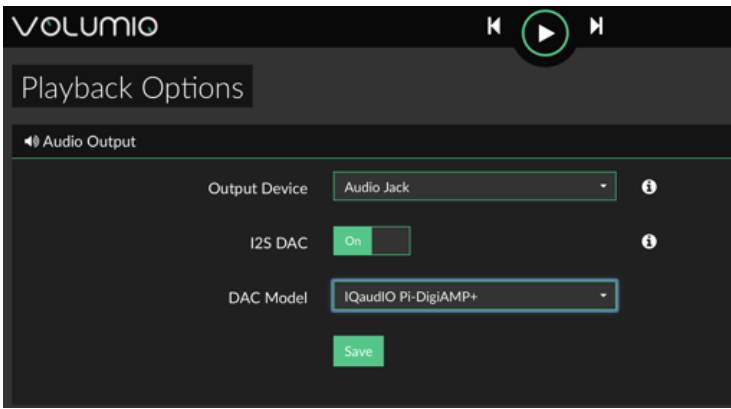

Volumio (https://volumio.org/get-started/)

From a fresh distribution of Volumio you will find that IQaudIO support is built in.

Write SD card image and boot Pi.
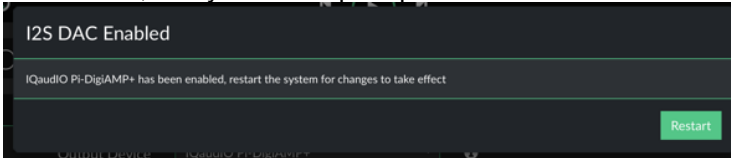connect from web browser to volumio.local

Select the top right hand side (burger) menu and select PLAYBACK OPTIONS.

Here change the audio interface to I2S, and select IQaudIO Pi-DigiAMP+ for "DAC Model".
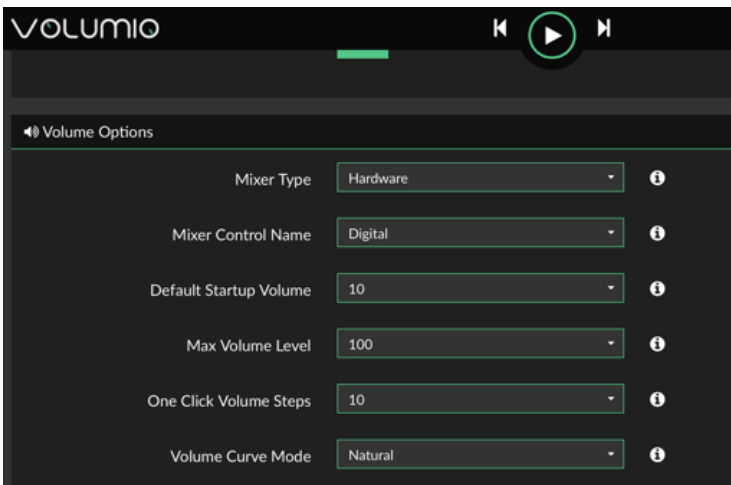
This will work with AMP/DAC/DAC PRO/DigiAMP.

Press Save, and you will be prompted to reboot.

Restart Volumio for this change to persist.

Once Volumio has restarted, once again select Menu/Playback options.

Ensure that the Volume Options have the Mixer type as Hardware, and that the Mixer Control Name is set to "Digital".
Press SAVE.

MoodeAudio (http://www.moodeaudio.org)

From a fresh distribution you will find that IQaudIO support is built in.

connect from web browser to moode.local

select configure from the pull down menu, then system.
Within the Audio section select the IQaudio device you have (such as Pi-DAC+) and press SET
Within the same selection make sure you set the volume to around 20% (not 100%) and press SET

Select reboot

Once reconnected….

Select configure from the pull down menu and then MPD
Within Setting, select volume control - hardware and then APPLY

Select reboot

RuneAudio (http://www.runeaudio.com)

From a fresh distribution you will find that QaudIO Pi-DAC+ support is built in.

Write SD card image, boot Pi.

ssh into the Pi as root (password rune) and edit /boot/config.txt to select the IQaudio Pi-DAC+ devicetree overlay. Save and reboot

connect from web browser to runeaudio.local

select the pull down settings menu and select MPD, here change audio interface to IQaudIO DAC, beneath this select Volume control: enabled-hardware

Press SAVE AND APPLY

Bob Rathbone's Internet Radio (http://www.bobrathbone.com/ raspberrypi/pi_internet_radio.html)

Bob's website has a huge amount of documentation to support the construction of purposeful internet radio's around the Raspberry Pi. He includes many details on IQaudIO board support and also includes support for the IQaudIO CosmicController buttons, volume control and mini-OLED display.

PiCorePlayer (https://sites.google.com/site/picoreplayer/home)

IQaudIO support is built in.

PiMusicBox (http://www.pimusicbox.com)

IQaudIO Pi-DAC+ support is built in.

Raspbian (https://www.raspberrypi.org/downloads/)

IQaudIO support is easily enabled as documented in this user guide

OpenElec (http://openelec.tv/get-openelec)

IQaudIO Pi-DAC+ support is configurable.

- Download Stable Openelec 6.0.3 image from http://releases.openelec.tv/OpenELEC-RPi2.arm-6.0.3.img.gz
- write image to SD card
- Boot Openelec

- SSH into the Raspberry PI and edit config.txt to include the IQaudio device tree
    - mount -o remount,rw /flash
    - nano /flash/config.txt
            # audio interfaces
                dtoverlay=iqaudio-dacplus
            # optional hardware interfaces
                dtparam=i2s=on
    - Once above changes have been made, then ctrl+x to save.
    - mount -o remount,ro /flash

- reboot

OSMC (https://osmc.tv)

IQaudIO Pi-DAC+ support is configurable.
Pre configured versions are available at www.iqaudio.com/downloads

Others

If you know of others please let us know by emailing us at info@iqaudio.com

21. RoonLabs setup

As a RoonLabs certified partner we have pre-generated sd card images for the Raspberry Pi and our IQaudIO sound cards. To date the following devices are certified:

RASPBERRY PI3 and:
- Pi-DAC+
- Pi-DAC PRO
- Pi-DigiAMP+
- Pi-Digi+

Using one of these boards and the corresponding SD card image will create a Roon RAAT based end-point from a Raspberry Pi. This is a headless device selectable and controllable from the Roon Core software, just like any other Roon Labs certified device.

We recommend Ethernet connection for RoonLabs end-points as WiFi can be susceptible to noise from DECT phones, microwaves etc. potentially causing the break up of audio playback. If you must use WiFi then this needs to be specially configured - see below.

NOTE: Although this is a RaspberryPi, it is dedicated to Roon Labs integration and as such the software running on the Pi has been "locked down", removing the need or ability to log into the Pi remotely or via connected keyboard.

Roon WiFi setup

As the RoonLabs end-point is headless, WiFi on Raspberry Pi needs to be configured through a text file on the micro-sd card itself. There are various walk though examples for this - google is your friend. One such write up is here: https://www.raspberrypi-spy.co.uk/2017/04/manually-setting-up-pi-wifi-using-wpa_supplicant-conf/

In general configuring WiFi for a headless device follows these simple steps:

1) On a PC or Apple Mac, connect the RaspberryPi sd-card
2) Create a file called wpa_supplicant.conf in the sd-card's boot partition
3) The contents of this file should be similar to below….

```
country=uk
update_config=1
ctrl_interface=/var/run/wpa_supplicant

network={
    scan_ssid=1
    ssid="MyNetworkSSID"
    psk="MyNetworkPassword"
}
```

4) Make sure you use your country code (no quotes) and your WiFi network's ssid (name) and psk (password). Both name and password should be surrounded by quotation marks.
5) Save the file back to the sd card
6) Properly eject the sd card within Windows / Mac OS
7) Place the sd card back into the RaspberryPi and boot
8) Check that the Pi has been allocated an IP address from your home network's router.
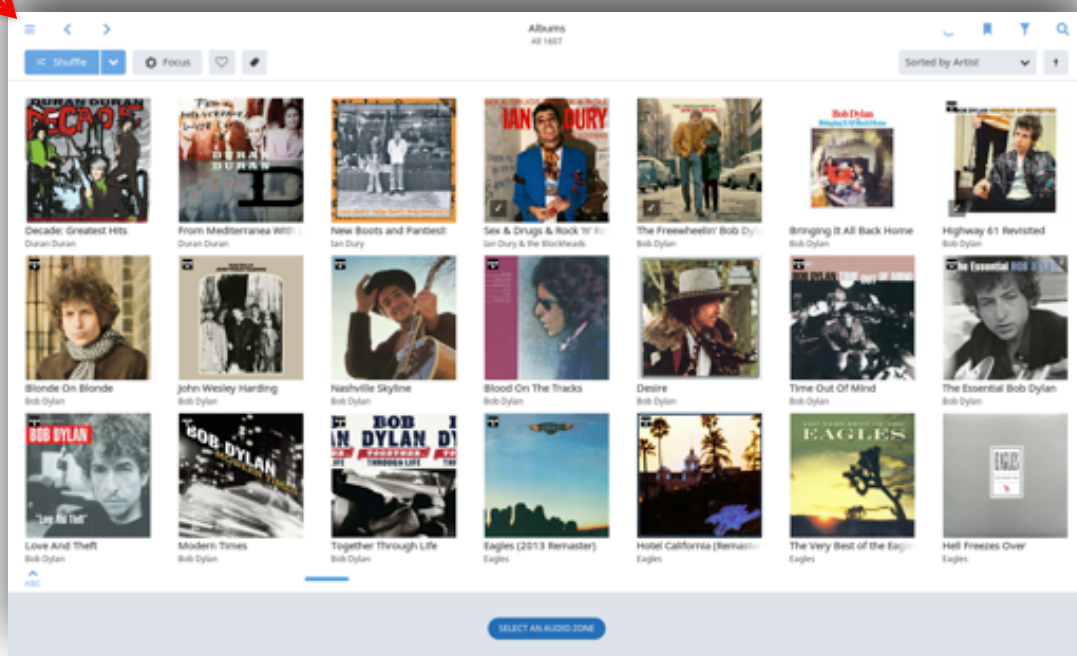
Configure Roon to use your IQaudIO device

Roon Labs software should be installed and running on your host PC or Apple Mac. To configure the music application to use the IQaudIO board please see below…
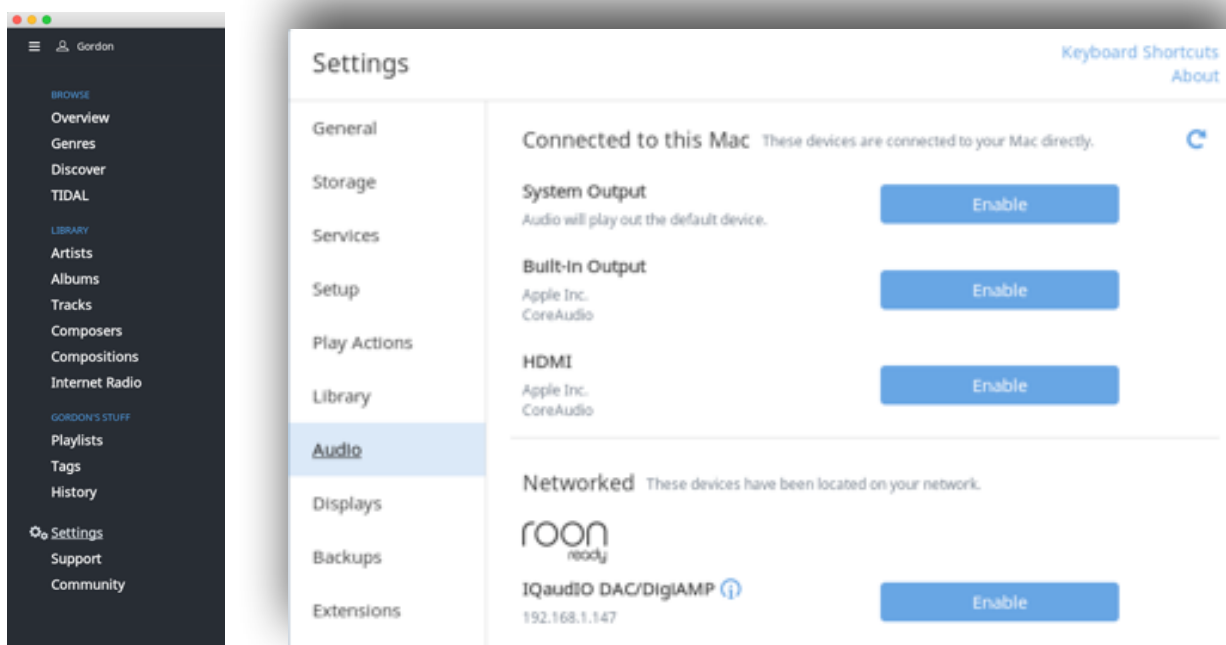
Start the Roon application and login.
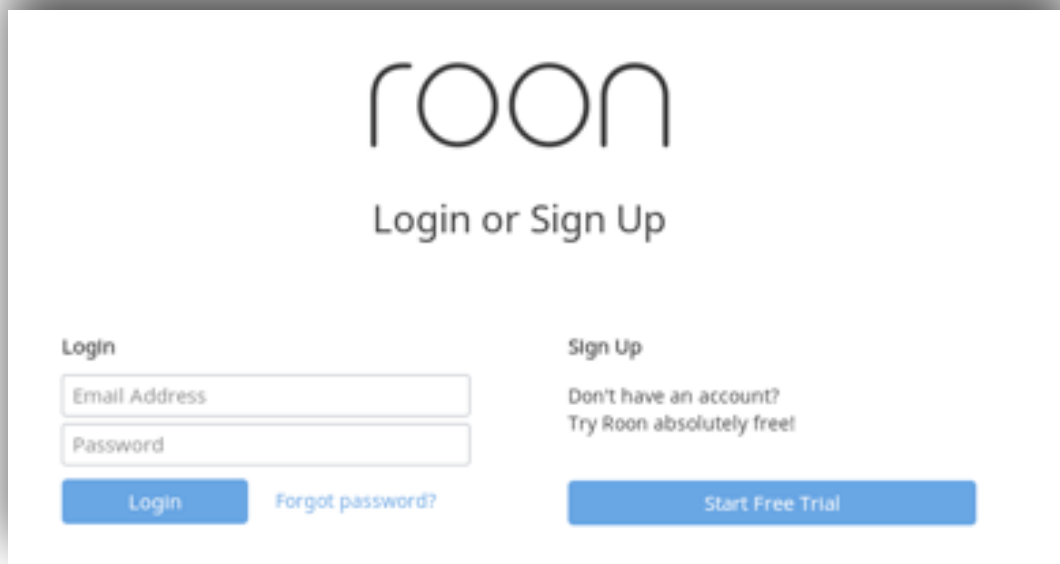
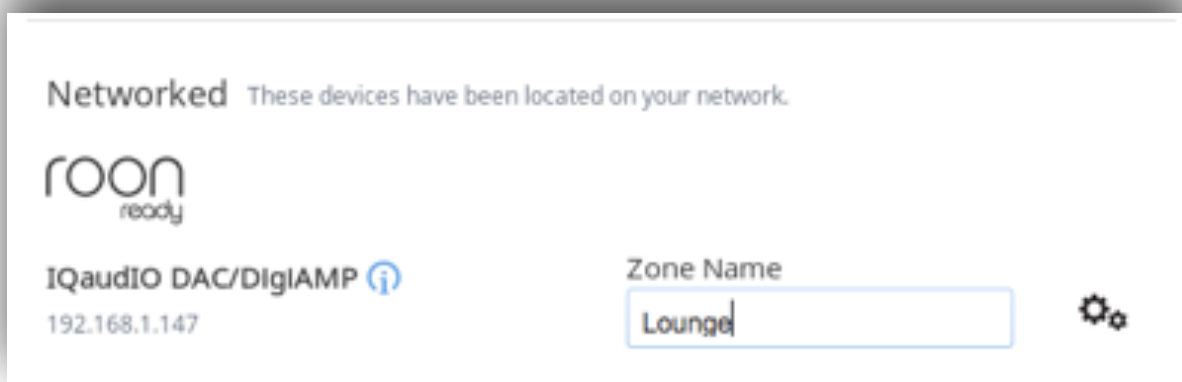
Once Roon has found your media, select the hamburger menu top left

Then select Settings (third from the bottom on this screen)

You will find any new IQaudIO Roon device listed under "Networked". You should then Enable this device and give it a descriptive Zone Name.

You can select the settings cog to adjust the IQaudIO board's internal Device settings if needed or just leave as default.
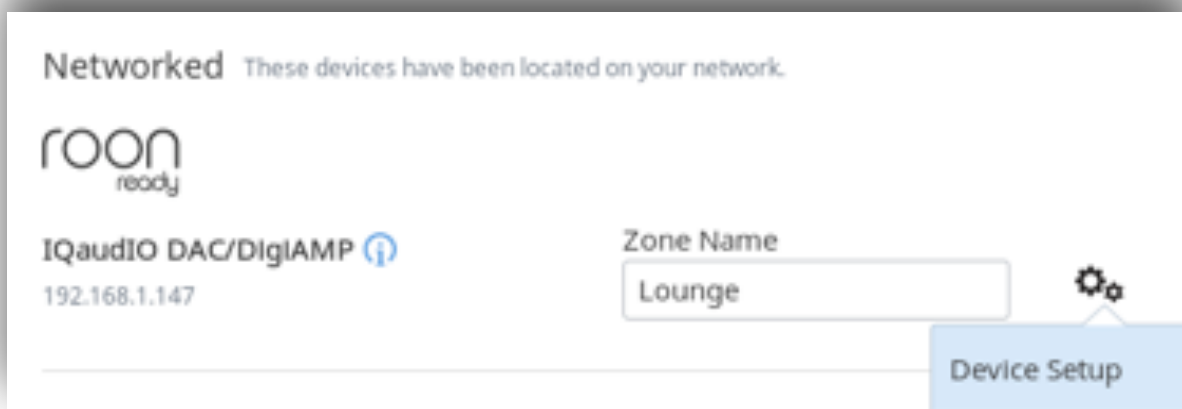
Device Volume: This allows the volume of the IQaudIO board to be adjusted from the Roon application, each zone can have a different volume as needed. Useful if the IQaudIO board is feeding directly into a Power amplifier or you are using a Pi-DigiAMP+

Fixed Volume: **DO NOT USE THIS WITH THE Pi-DigiAMP+**. This setting is for use when the Iqaudio board is fed into a HiFi pre-amp with its own volume adjustment.

Now the IQaudIO board has been configured and given a Zone Name within Roon, you can start to playback music to the device. First select the Zone you want to control (such as Lounge).
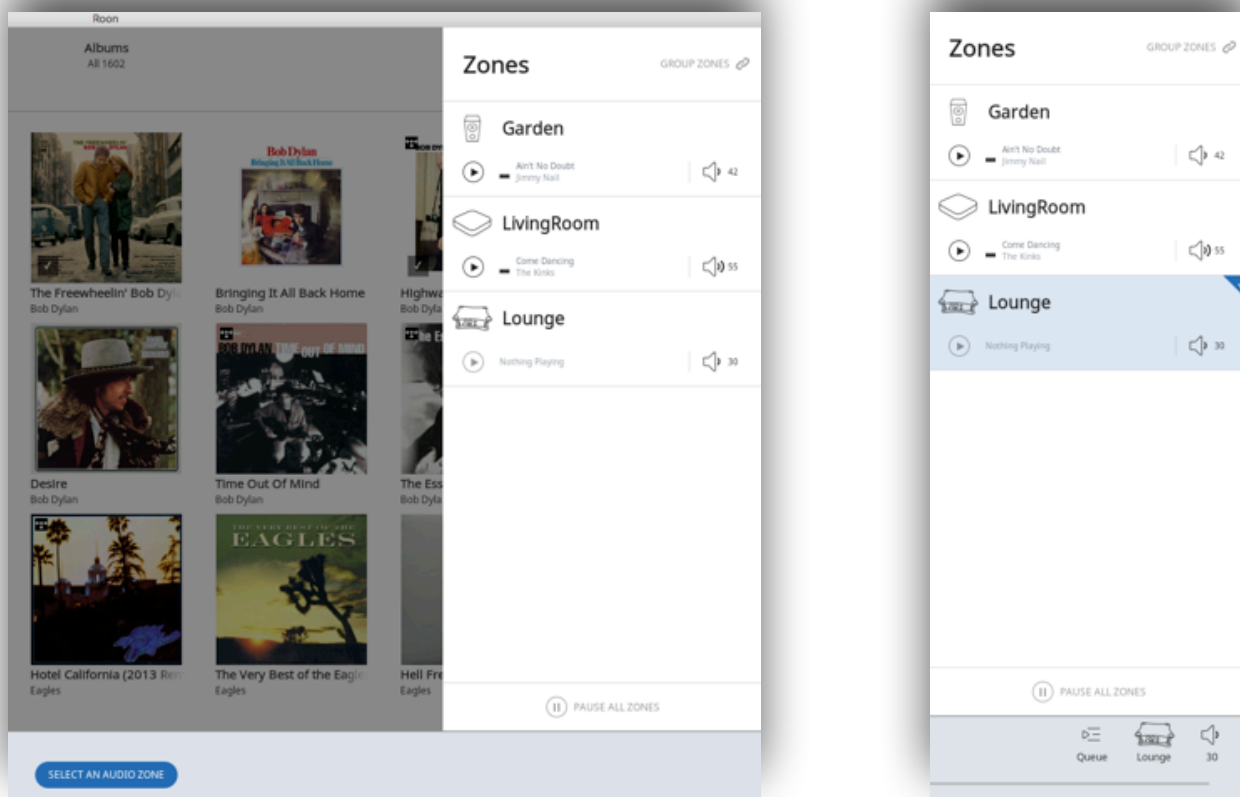
From then on you can select music to play and adjust the Zone's volume from the line-art at the bottom of the screen.
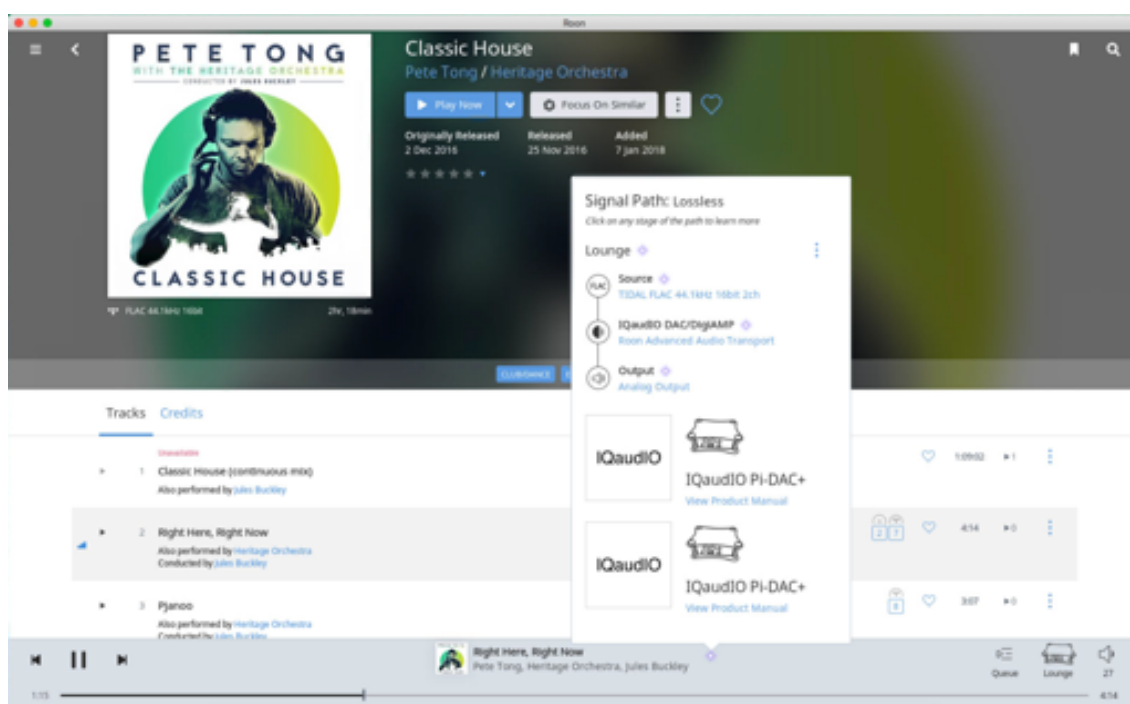
You can also group output devices into a new playback Zone allowing "party mode" playback in synchronisation if needed.

22. Advanced information

The following section is for more advanced users.

GPIO usage / Pin headers)

The IQaudIO range of sound cards for the Raspberry Pi take advantage of a number of Pi signals in order to operate successfully. Some of these signals are solely for the use of the IQaudIO board and some can be shared with other peripherals, sensors, etc. The information below covers the general cases. For more Raspberry Pi GPIO information please see the excellent site: https://pinout.xyz

We expect the following Raspberry Pi GPIO to be used by IQaudIO audio boards
- ALL POWER PINS
- ALL GROUND PINS
- GPIO 2/3 (I2C)
- GPIO 18/19/20/21 (I2S)

If appropriate then the following are also used
- GPIO 22 (DigiAMP+ mute / unmute support)
- GPIO 23/24 for Rotary Encoder (physical volume control) or status LED (Pi-CodecZero)
- GPIO 25 for the IR Sensor.
- GPIO 27 for the Rotary Encoder push switch / Pi-CodeZero switch

Pi-DAC PRO, Pi-DAC+, Pi-DigiAMP+, Pi-DACZero, Pi-Codec+, Pi-CodecZero

	Pin	NAME			NAME	Pin	
	1	**3.3v** (DC Power)			(DC Power) **5v**	2	
SHARED USE ------>	3	**GPIO2** (I2C SDA1)			(DC Power) **5v**	4	
SHARED USE ------>	5	**GPIO3** (I2C SCL1)			(GROUND) **0v**	6	
	7	**GPIO4** (GCLK)			(TXD0) **GPIO14**	8	
	9	**0v** (GROUND)			(RXD0) **GPIO15**	10	
	11	**GPIO17** (GEN0)			(I2S CLK) **GPIO18**	12	<---- I2S AUDIO
Rotary Encoder / Button ------>	13	**GPIO27** (GEN2)			(GROUND) **0v**	14	
Pi-DigiAMP+ (Mute) ------>	15	**GPIO22** (GEN3)			(GEN4) **GPIO23**	16	<---- Rotary Encoder / LED
	17	**3.3v** (DC Power)			(GEN5) **GPIO24**	18	<---- Rotary Encoder / LED
	19	**GPIO10**			(GROUND) **0v**	20	
	21	**GPIO9** (SPI_MISO)			(GEN6) **GPIO25**	22	<---- IR Receiver
	23	**GPIO11** (SPI_CLK)			(SPI CE0) **GPIO8**	24	
	25	**0v** (GROUND)			(SPI CE1) **GPIO7**	26	
Used by EEPROM ------>	27	**ID_SD** (HAT			(HAT EEPROM)	28	<---- Used by EEPROM
	29	**GPIO5**			(GROUND) **0v**	30	
	31	**GPIO6**			**GPIO12**	32	
	33	**GPIO13**			(GROUND) **0v**	34	
I2S AUDIO ------>	35	**GPIO19** (I2S			**GPIO16**	36	
	37	**GPIO26**			(I2S DIN) **GPIO20**	38	<---- I2S AUDIO
	39	**0v** (GROUND)			(I2S DOUT) **GPIO21**	40	<---- I2S AUDIO

The Pi-DAC PRO, Pi-DAC+, Pi-DigiAMP+ and Pi-Codec+ re-exposes the Raspberry Pi signals allowing additional sensors etc to be added easily. Please note that some signals are for exclusive use (I2S and EEPROM) for some of our boards, others such as I2C can be shared across multiple boards.


EXTERNAL PI HEADER
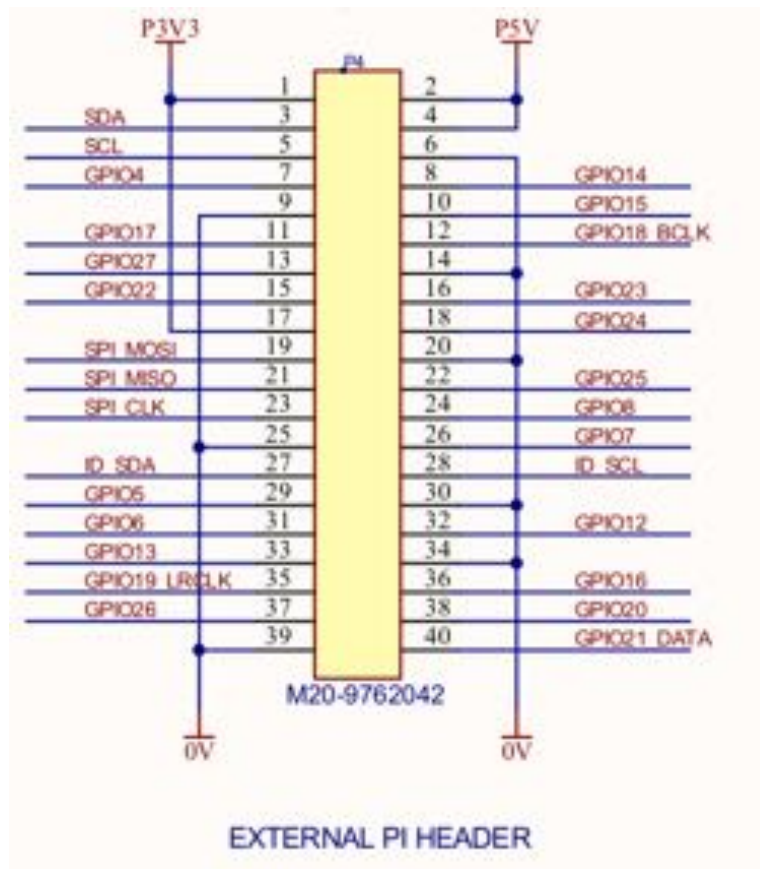
The old (Legacy) Pi-DigiAMP+ had a 2x5 pin header which exposed the following pins:
Pin1 of P6 is the closest to the main chip. Pin10 of P6 closest to the Pi's USB ports.

EXTENSION HEADER
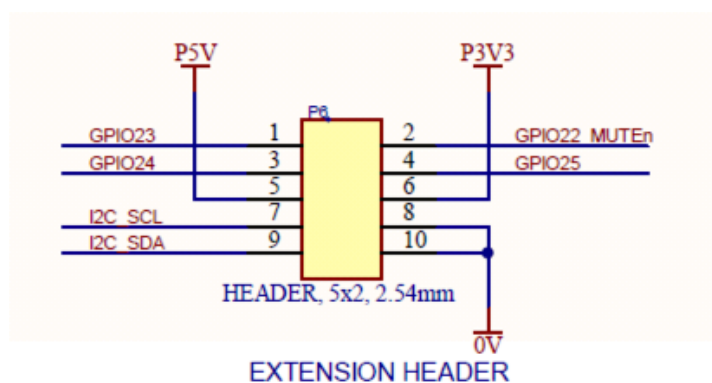
23. Adding a Rotary Encoder

The following section shows the recommended pins / GPIO to use when adding a Rotary Encoder (physical volume control). Sample software is available on our GitHub page: www.github.com/iqaudio/tools) that converts physical rotation to ALSA audio volume commands.

With the rotary encoder shaft facing you and pins facing downwards:

RIGHT Pin
-> GPIO 24
(Pi pin 16)

CENTRE Pin
-> Ground
(Pi pin 14)

LEFT Pin
-> GPIO 23
(Pi pin 18)
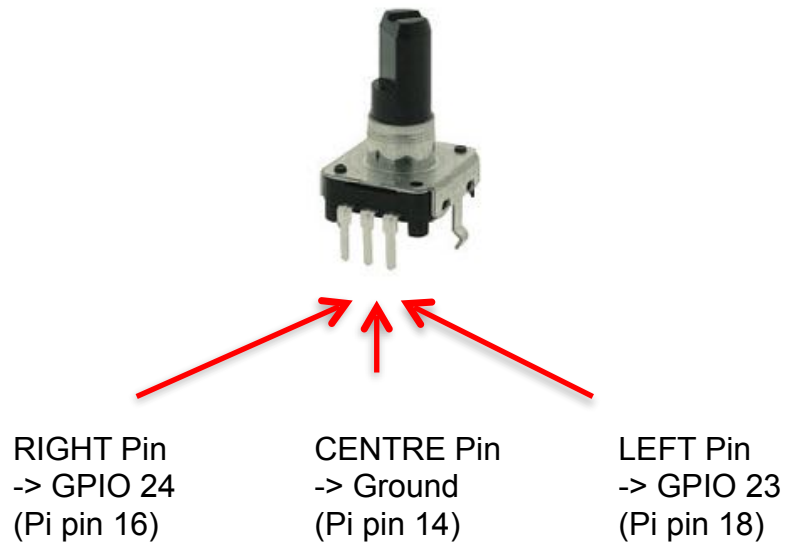
24. FAQ (Frequently Asked Questions) with answers!

1. Saving alsamixer settings

Storing the alsamixer setting can be achieved by performing the following at the command line:

> sudo alsactl store

It is possible to save the current state to a file, then reload that state at startup (very useful for the Pi-Codec boards and use cases)

To save:
> sudo alsactl store -f /home/pi/usecase.state

To restore a saved file:
> sudo alsactl restore -f /home/pi/usecase.state

2. Using external USB devices

Note that if you want to enable 1.2AMP USB support (to allow USB hard disks to power up when accessory boards are in use) then you may also want to add the following line to your /boot/config.txt file:

```
max_usb_current=1
```

Please ensure your Pi power supply is able to deliver ~2.5amp if you are powering via mini USB and enable max_usb_current

3. MPD based audio with volume control

To allow MPD based audio software to control the audio board's built in HARDWARE volume the file /etc/mpd.conf may need to be changed to support the correct ALSA mixer name.

This can be achieved by ensuring the'Audio Output' section of /etc/mpd.conf has the 'mixer_control' line. Below is an example for the Texas Instruments based IQaudIO sound cards (Pi-DAC PRO/Pi-DAC+/Pi-DigiAMP+/Pi-DACZERO)

```
audio_output {
    type         "alsa"
    name         "ALSA Device"
    mixer_control   "Digital"
}
```

4. Useful data (screw sizes / part numbers / wire gauge size etc.)

The full range of IQaudIO sound cards and accessories are available directly from IQaudIO Limited (www.iqaudio.com) or through various Worldwide distribution channels.

Useful Data:

- PCB screws are all m2.5
- PCB standoffs (for Pi-CASE) are 5mm male / female
- PCB standoffs (for Pi to IQaudIO products) are 9mm female / female
- PCB standoffs (for Pi to IQaudIO LEGACY products) are 11mm female / female
- PCB standoffs (for XLR to Pi-DAC PRO) are 8mm female / male
- PCB standoffs (for DACZero Headphones to Pi-DACZERO) are 11mm female / male
- PCB standoffs (for the official Raspberry Pi 7" display) are 5mm male / female*

- The rotary encoders we have used and tested are the ALPHA 3 pin Rotary Encoder RE160F-40E3-20A-24P, the ALPS EC12E2430804 (RS: 729-5848), the Bourns ECW0J-B24-AC0006L (RS: 263-2839)
- The CosmicController can interface to the Olimex OLED (https://www.olimex.com/Products/Modules/LCD/MOD-OLED-128x64/open-source-hardware)

- The Barrel connector used for powering the Pi-DigiAMP+ is 2.5mmID, 5.5mmOD, 11mm
- The Pi-DigiAMP+ is designed to operate with a 12v 3amp to 24v supply such as the XPPower VEC65US19 or similar
- The Pi-DigiAMP+ uses CamdenBoss two part connectors. Those fitted to the PCB are: CTBP9350/2AO

- The speaker terminal used on the Pi-Codec+ / Pi-CodecZero will accept wires of between 14 ~ 26 AWG (wire of max 1.6mm in diameter)