

IQaudio Product Guide

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

IQaudio user instructions

Contents

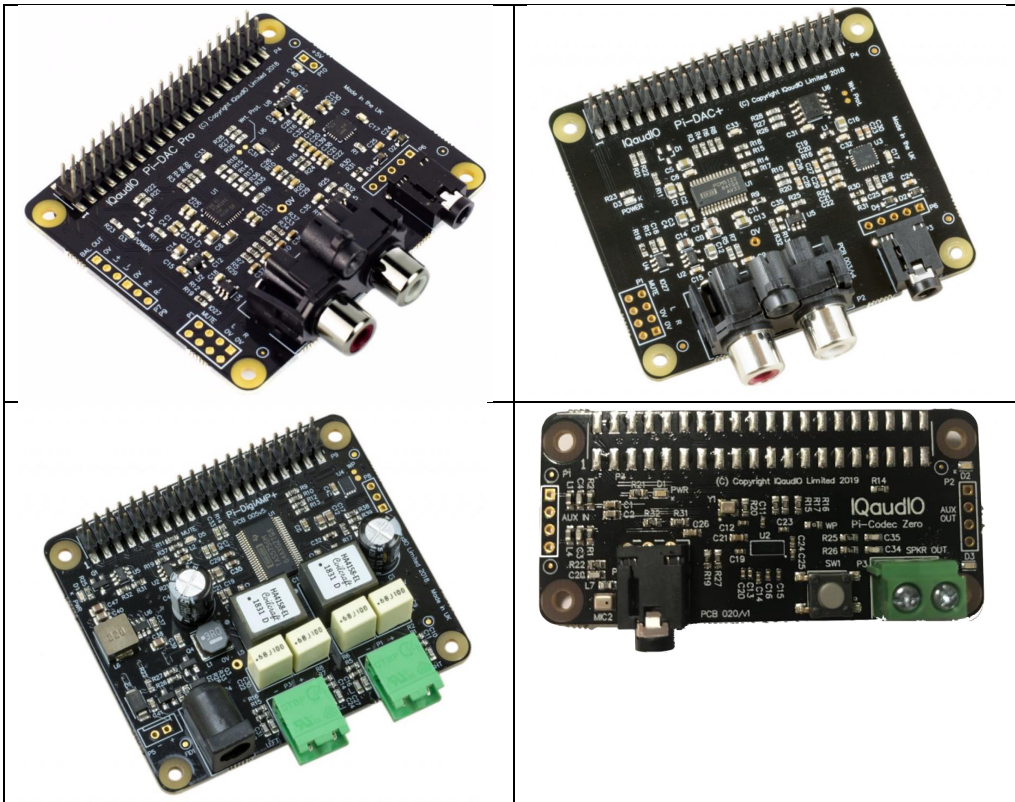
1.	GETTING STARTED	3
2.	WHICH IQAUDIO BOARD IS BEST FOR ME?	4
3.	CONFIGURING LINUX FOR THE IQAUDIO SOUND CARD	5
4.	HEADLESS SOFTWARE	6
5.	HOW DO I INSTALL SOFTWARE ON THE PI?	7
6.	DAC PRO	8
7.	DAC+	9
8.	DIGIAMP+	10
9.	CODEC ZERO	11
10.	ADDING THE IQAUDIO SOUND CARD TO YOUR RASPBERRY PI	12
11.	CODEC ZERO CONFIGURATION	14
12.	CONTROLLING THE DIGIAMP+ MUTE / UN-MUTE	16
13.	RASPBERRY PI AUDIO APPLICATIONS	17
14.	ADVANCED INFORMATION	21
15.	ADDING A ROTARY ENCODER	23
16.	FAQ (FREQUENTLY ASKED QUESTIONS) WITH ANSWERS!	24

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 and software package you want to use.

The IQaudio product family

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 Raspberry Pi 4B, 3B+, 3A+, 3, Zero etc. Some of our products are full-size HAT boards for the full-size Raspberry Pi, and others are more suited to the Raspberry Pi Zero form factor.



High-Resolution Audio Playback

	DAC Pro	DAC+	DigiAMP+		Codec Zero	
24bit Depth	★	★	★		★	
Hi-Def Audio (96kHz)	★	★	★		★	
Full HD Audio (192kHz)	★	★	★			

2. Which IQaudio board is best for me?

Each IQaudio sound card 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 DAC PRO, DAC+ and DigiAMP+ boards. All support 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
DAC PRO	★	★ (XLR board)			★				
DAC+	★				★				
DigiAMP+			★						
Codec Zero				★ (MONO)		★ (No Fit)	★ (No Fit)	★ (Electret)	★

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

All our latest full size audio boards include a pre-programmed EEPROM. This allows Raspberry Pi OS to auto detect and auto configure itself allowing the IQaudio boards to be plug and play.

If you need to configure Raspberry Pi OS yourself, you can edit the boot configuration file (found at /boot/config.txt) to manually add the appropriate device-tree overlay:

IQaudio Card	/boot/config.txt
DAC+	dtoverlay=iqaudio-dacplus
DAC PRO	dtoverlay=iqaudio-dacplus
DigiAMP+	dtoverlay=iqaudio-dacplus,unmute_amp or dtoverlay=iqaudio-dacplus,auto_mute_amp
Codec Zero	dtoverlay=iqaudio-codec

Disabling the standard Raspberry Pi On-Board Audio

To make your IQaudio sound card the primary audio device in Raspberry Pi OS you will need to disable the Raspberry Pi's on-board 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

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 Raspberry Pi into a Roon Certified end-point. The Pi is then controlled from 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](#)

Max2Play - IQAudio Edition



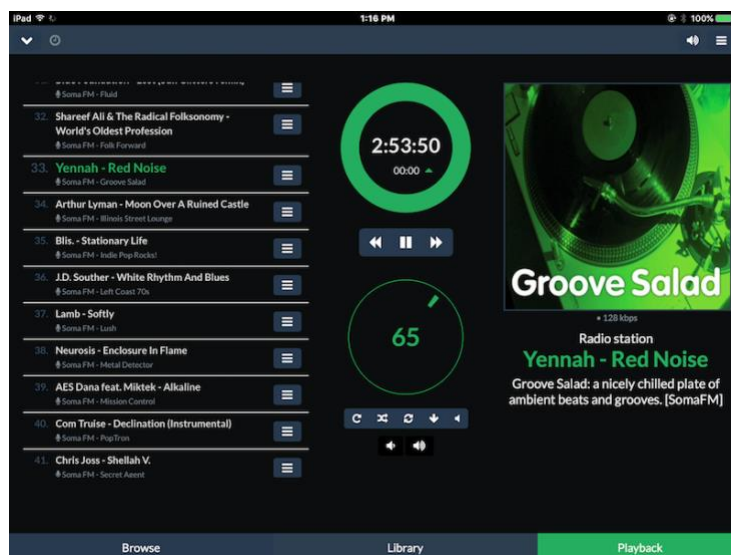
Your installed IQAudio Card: IQaudIO Pi-AMP+/DigiAMP+ (with unmute) save

Settings for Master Volume (Master) up volume and post to suit your needs. Set to optimum

Settings for Analogue Playback IQaudIO Pi-AMP+/DigiAMP+ (with unmute) ue for best performance and click save.

Settings for Analogue Boost (Optimum 100%) **100**

Volumio:

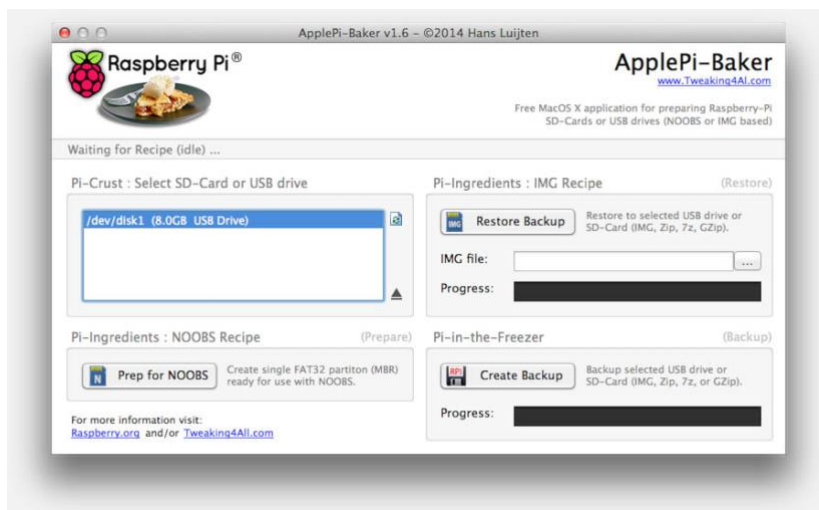


5. How do I install software on the Pi?

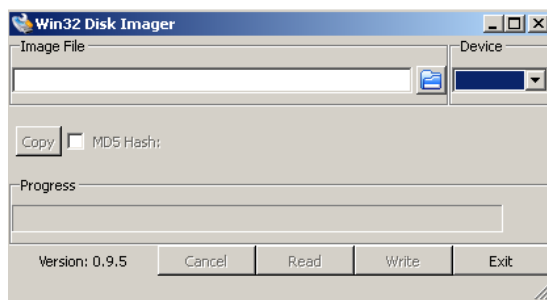
The Raspberry Pi normally loads its software from the micro-SD card. Most audio applications assume they are the only thing running on the Pi so most require a dedicated SD card. The applications are normally distributed as a compressed .img file (likely a .zip or .gz file) and this .img file will need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

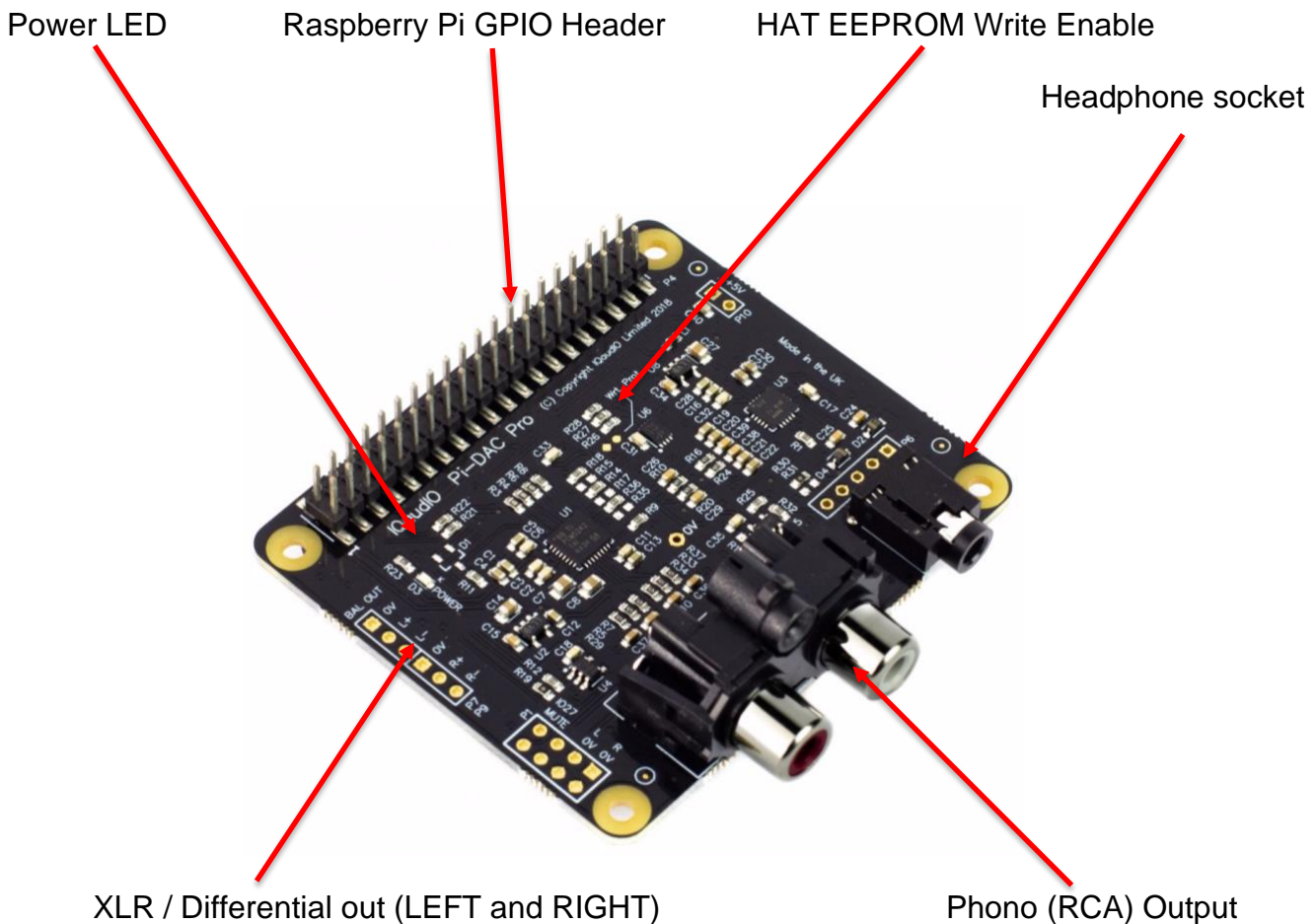
6. DAC PRO

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

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

The DAC PRO can be connected to the Raspberry Pi A+/B+/RPi2/RPi3 or RPi4's 40way pin header without any additional soldering. The 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 DAC PRO.



The 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 Raspberry 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)

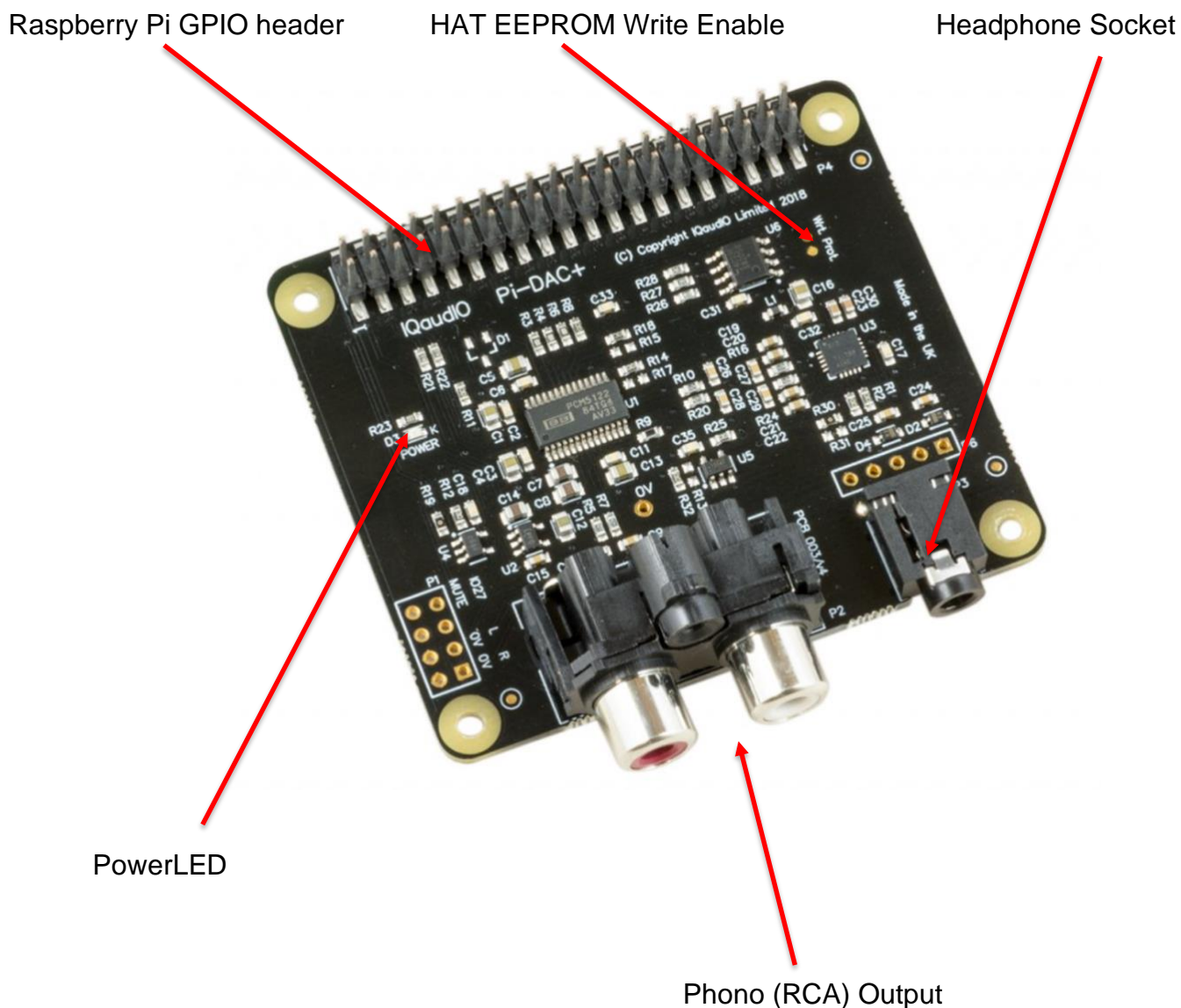
P10: Alternative 5v input, powering the Raspberry Pi in parallel

7. DAC+

The DAC+ is a HAT compliant add-on accessory for the Raspberry Pi A+/B+/RPi2/RPi3 or Pi4.

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

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



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. DigiAMP+

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

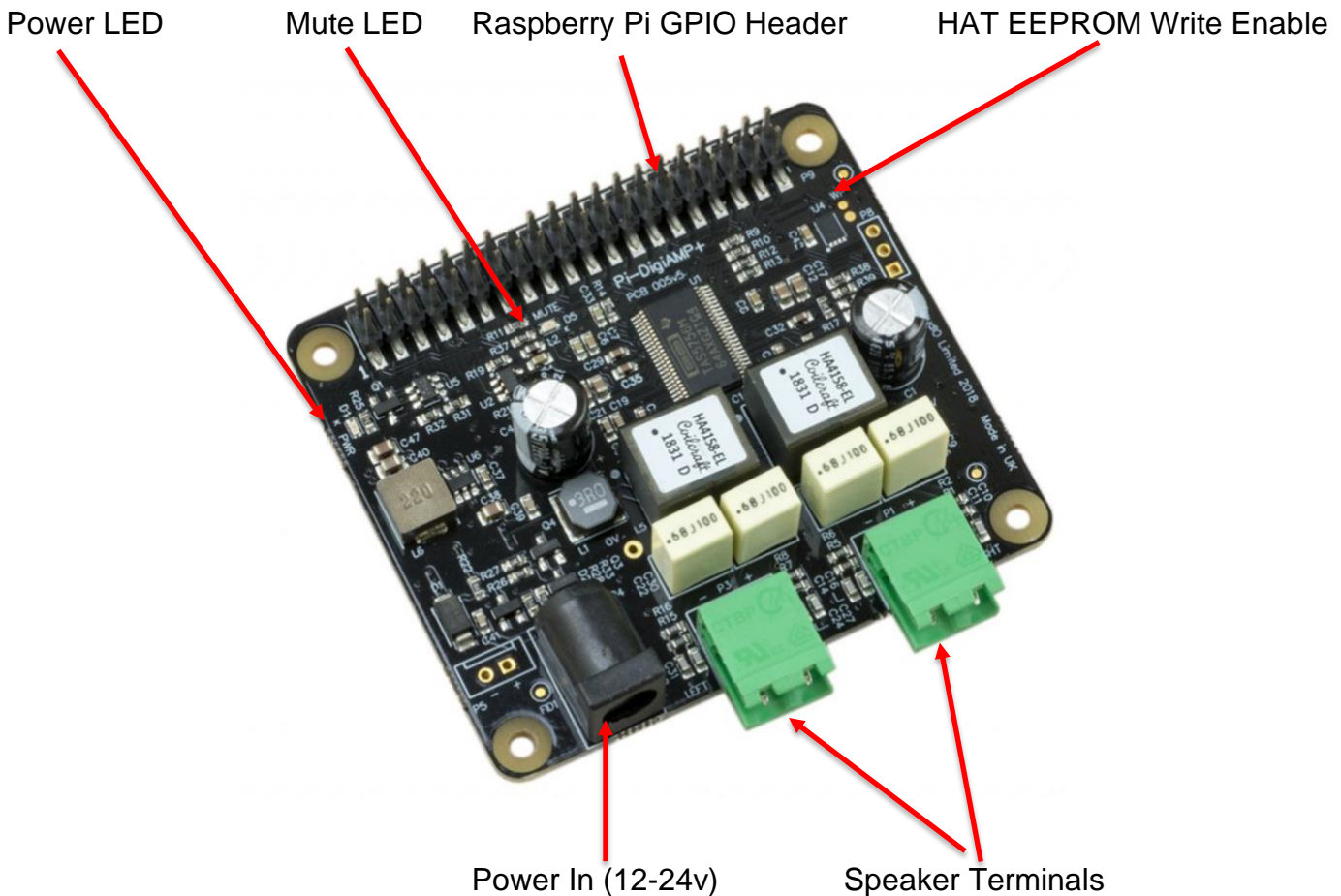
DigiAMP+ uses the Texas Instruments TAS5756M PowerDAC to deliver direct connection to PASSIVE stereo speakers at up to 2x35wpc with variable output. Ideal for a Pi based HiFi.

DigiAMP+ can be connected to the Raspberry Pi A+/B+/RPi2/RPi3 or RPi4's 40way pin header without any additional soldering.

DigiAMP+ must be powered from an external supply and requires a 12 to 24DC power source to operate. DigiAMP+'s barrel connector is 5.5mmx2.5mm. DigiAMP+ has been designed to provide power to the Raspberry Pi and 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 DIGIAMP+.

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



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

P8: TAS5756m Internal GPIO1/2/3

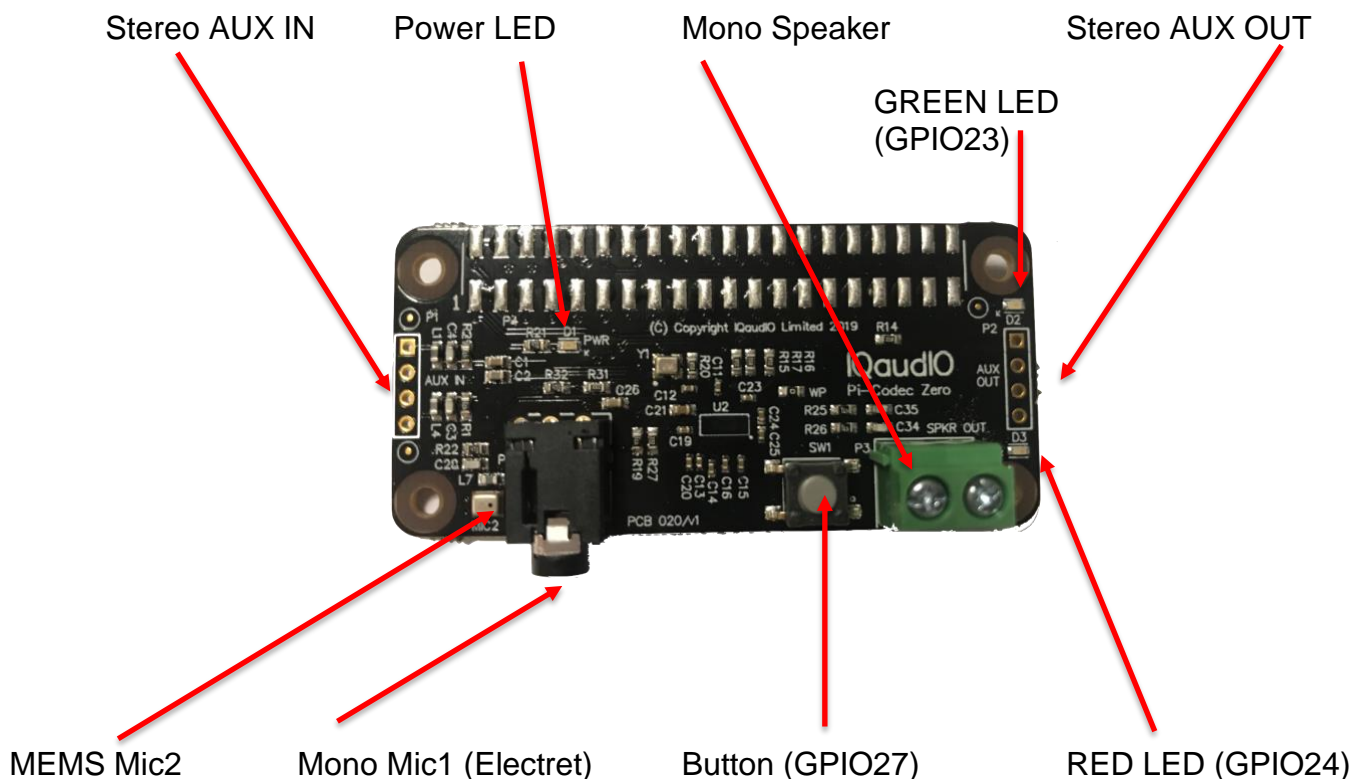
9. Codec Zero

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

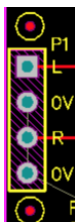
The Codec Zero delivers bi-directional digital audio signals (I2S) between the Raspberry Pi and onboard Dialog Semiconductor DA7212. Supporting a range of input and output devices from built in MEMS microphone, mono Electret microphone or mono speaker (1.2W / 8Ω).

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 Codec Zero 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:



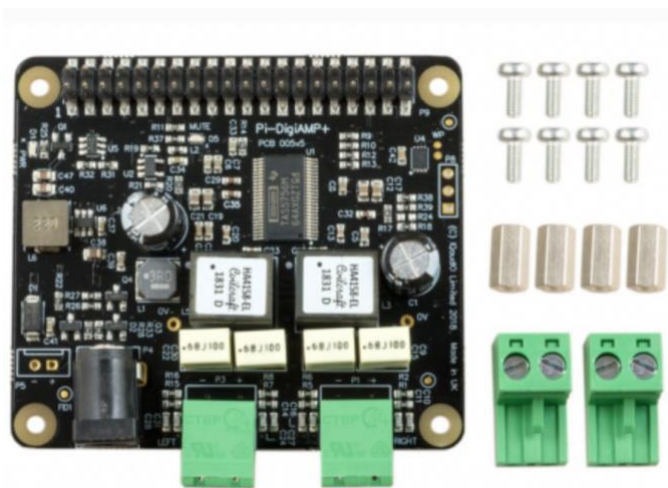
- 1: Left
- 2: Ground
- 3: Right
- 4: Ground

10. Adding the IQaudio sound card to your Raspberry Pi

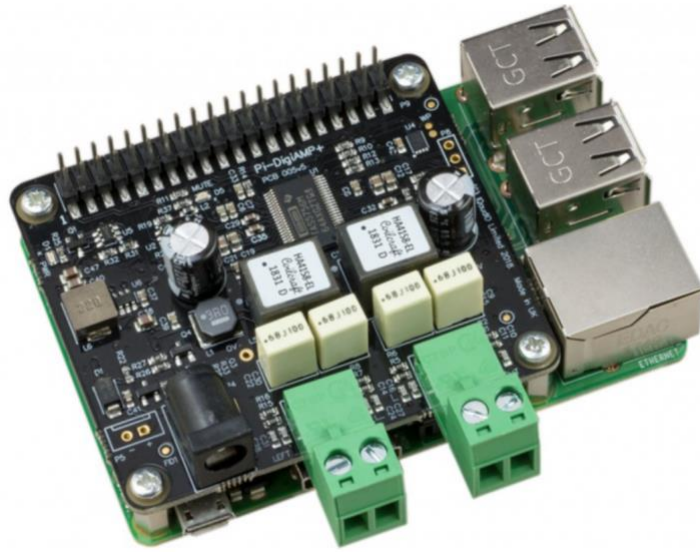
WARNING: PLEASE ENSURE ALL POWER CABLES ARE DISCONNECTED FROM THE RASPBERRY PI BEFORE ATTEMPTING THIS PROCEDURE.

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 PCB spacers should be screwed (finger tight only) to the Raspberry Pi before adding the IQaudio board.



The remaining screws are then screwed into the spacers from above.

11. Codec Zero Configuration

The IQaudio Codec Zero board 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 its 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 be adjusted to get the best possible output.

Codec Zero 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.

12. Controlling the DigiAMP+ Mute / Un-mute

The DigiAMP+ 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
```

13. 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).

Max2Play IQaudio Special Edition

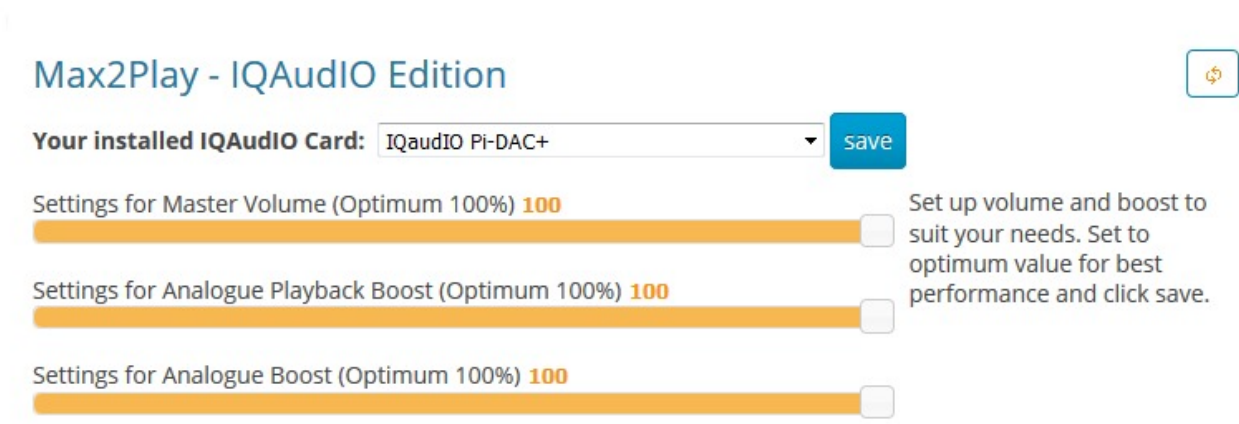
IQaudio Special Edition - (<https://www.max2play.com/en/max2play-image/>) support is built in - you can install all IQaudio cards with just one click.

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.

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.

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.

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:

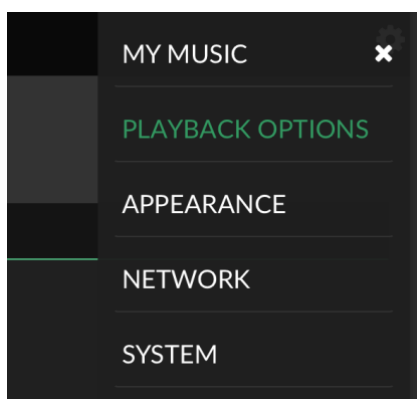
```
-o hw:CARD=IQaudIODAC -V Digital -Q
```

Install Rotary Encoder Software

Autostart Rotary Encoder Software:

save

For more guidance and to follow along the settings, you can also check out [the YouTube video](#) on the IQaudio Plugin by Max2Play.



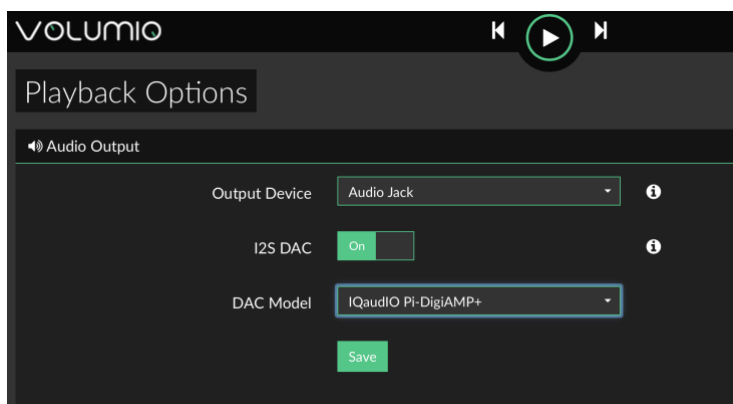
Volumio (<https://volumio.org/get-started/>)

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

Write SD card image and boot Raspberry Pi.
Connect from web browser to `volumio.local`

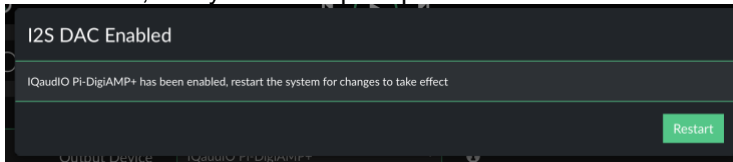
Select the top right hand side (burger) menu and select **PLAYBACK OPTIONS**.

Here change the audio interface to I2S, and select `Iqaudio DigiAMP+` for "DAC Model".



This will work with DAC/DAC PRO/DigiAMP.

Press Save, and you will be prompted to reboot.

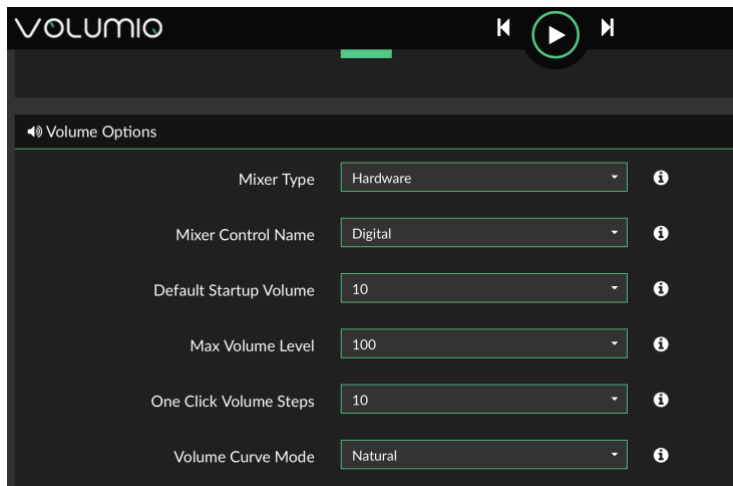


Restart Volumio for this change to persist.

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 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 IQaudio 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 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.

PiCorePlayer (<https://www.picoreplayer.org/>)

IQaudio support is built in.

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

IQaudio 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 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 DAC+ support is configurable.

14. 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)

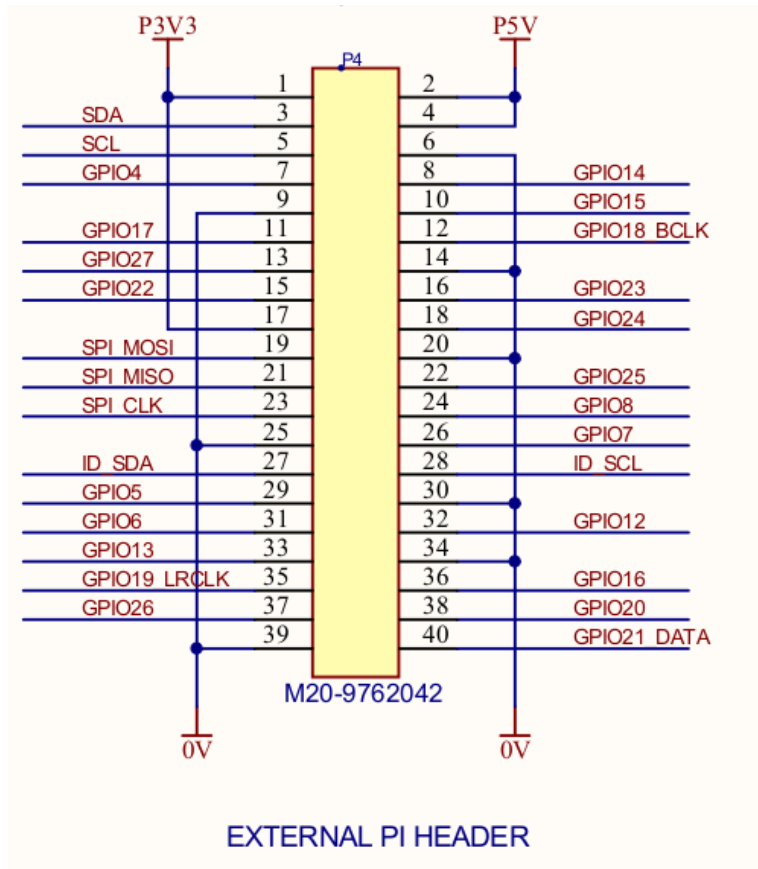
If appropriate then the following are also used

- GPIO 22 (DigiAMP+ mute / unmute support)
- GPIO 23/24 for Rotary Encoder (physical volume control) or status LED (Codec Zero)
- GPIO 25 for the IR Sensor.
- GPIO 27 for the Rotary Encoder push switch / Code Zero switch

DAC PRO, DAC+, DigiAMP+, Codec Zero

		SIGNALS					
Pin	Name			Name	Pin		
	1	3V3 (DC Power)	<input type="checkbox"/>	<input type="checkbox"/>	(DC Power) 5V	2	
Shared Use	3	GPIO2 (I2C SDA1)	<input type="checkbox"/>	<input type="checkbox"/>	(DC Power) 5V	4	
Shared Use	5	GPIO3 (I2C SCL1)	<input type="checkbox"/>	<input type="checkbox"/>	(Ground) 0V	6	
	7	GPIO4 (GCLK)	<input type="checkbox"/>	<input type="checkbox"/>	(TXD0) GPIO14	8	
	9	0V (Ground)	<input type="checkbox"/>	<input type="checkbox"/>	(RXD0) GPIO15	10	
	11	GPIO17 (GEN0)	<input type="checkbox"/>	<input type="checkbox"/>	(I2S CLK) GPIO18	12	I2S Audio
Encoder / Button	13	GPIO27 (GEN2)	<input type="checkbox"/>	<input type="checkbox"/>	(Ground) 0V	14	
DigiAMP+ (Mute)	15	GPIO22 (GEN3)	<input type="checkbox"/>	<input type="checkbox"/>	(GEN4) GPIO23	16	Rotary Encoder / LED
	17	3V3 (DC Power)	<input type="checkbox"/>	<input type="checkbox"/>	(GEN5) GPIO24	18	Rotary Encoder / LED
	19	GPIO10 (SPI_MOSI)	<input type="checkbox"/>	<input type="checkbox"/>	(Ground) 0V	20	
	21	GPIO9 (SPI_MISO)	<input type="checkbox"/>	<input type="checkbox"/>	(GEN6) GPIO25	22	IR Receiver
	23	GPIO11 (SPI_CLK)	<input type="checkbox"/>	<input type="checkbox"/>	(SPI_CE0_N) GPIO8	24	
	25	0V (Ground)	<input type="checkbox"/>	<input type="checkbox"/>	(SPI_CE1_N) GPIO7	26	
EEPROM	27	ID_SD (HAT)	<input type="checkbox"/>	<input type="checkbox"/>	(HAT) ID_SC	28	EEPROM
	29	GPIO5	<input type="checkbox"/>	<input type="checkbox"/>	(Ground) 0V	30	
	31	GPIO6	<input type="checkbox"/>	<input type="checkbox"/>	GPIO12	32	
	33	GPIO13	<input type="checkbox"/>	<input type="checkbox"/>	(Ground) 0V	34	
I2S Audio	35	GPIO19 (I2S)	<input type="checkbox"/>	<input type="checkbox"/>	GPIO16	36	
	37	GPIO26	<input type="checkbox"/>	<input type="checkbox"/>	(I2S DIN) GPIO20	38	I2S Audio
	39	0V (Ground)	<input type="checkbox"/>	<input type="checkbox"/>	(I2S DOUT) GPIO21	40	I2S Audio

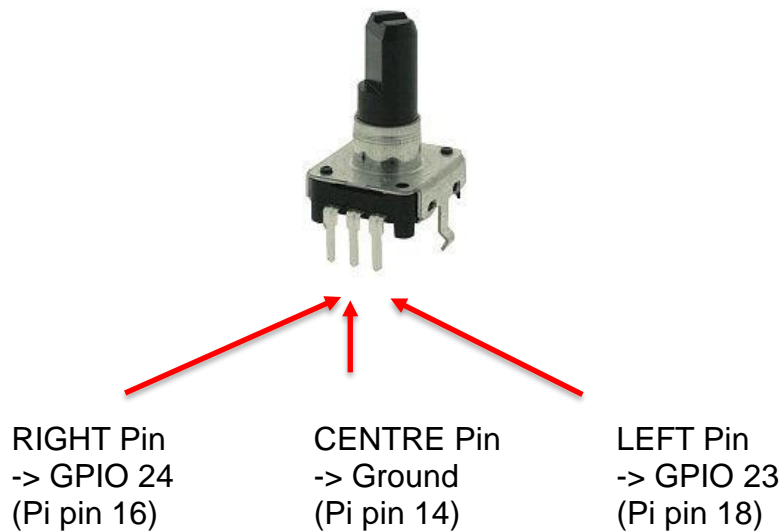
The DAC PRO, DAC+, DigiAMP+ re-expose the Raspberry Pi signals allowing additional sensors etc to be added easily. Please note that some signals are for exclusive use (I2S and EEPROM) for some of our boards, others such as I2C can be shared across multiple boards.



15. 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:



16. 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 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 (DAC PRO/DAC+/DigiAMP+)

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