IQaudIO

This document covers the installation, software configuration and usage of Pi-DAC+, Pi-AMP+ and Pi-CASE+ for the Raspberry Pi A+, B+ and Raspberry Pi v2.

If there is additional information required please email us at info@iqaudio.com

Yours sincerely,

Gordon Garrity
IQaudIO Limited

Pi-DAC+ / Pi-AMP+ / Pi-CASE+ instructions
(For Raspberry Pi A+ / B+ / RPi2)

Contents

1) The Pi-DAC+

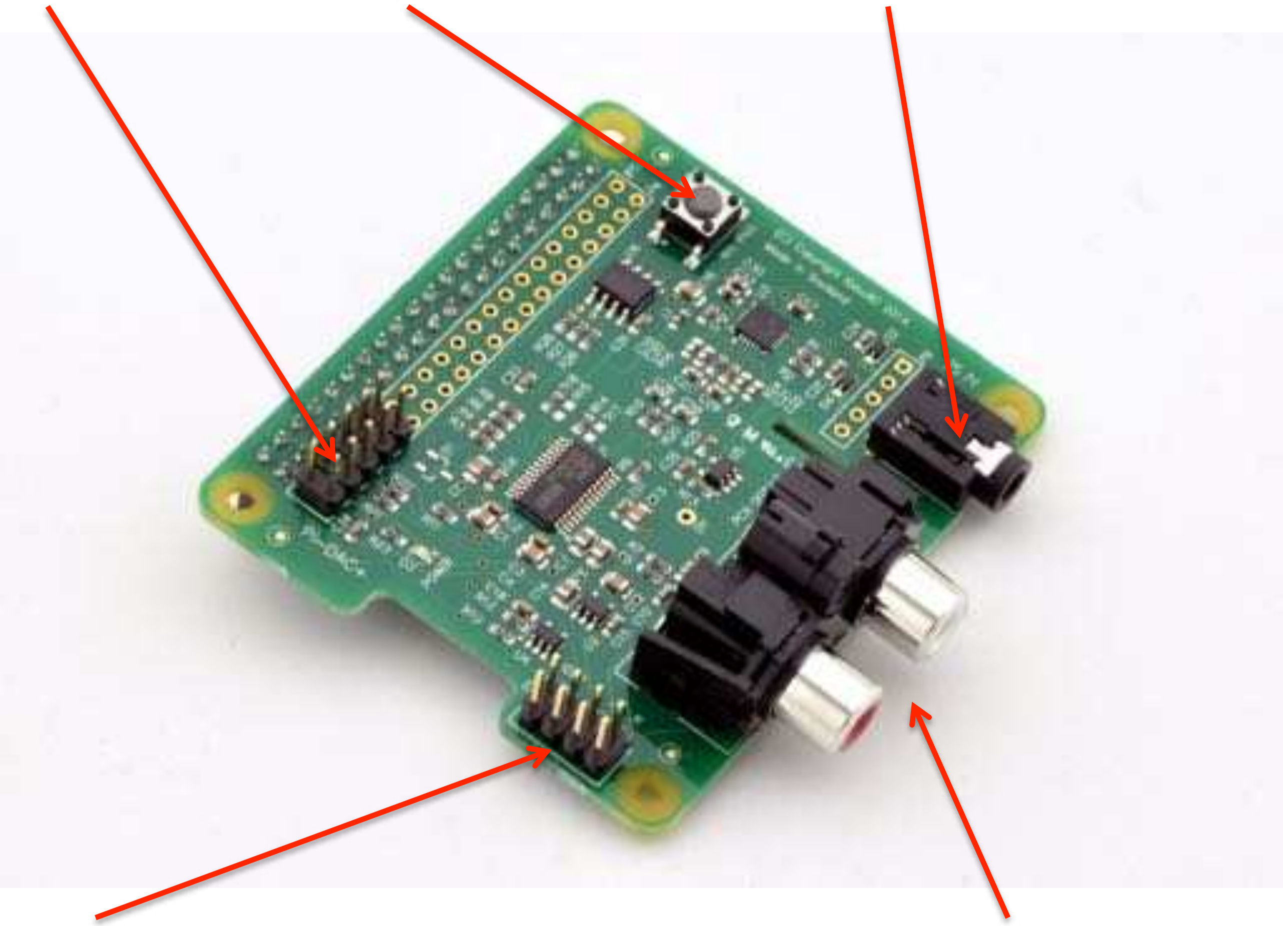
The Pi-DAC+ is a fully HAT compliant add-on accessory for the Raspberry Pi A+/B+.

The Pi-DAC+ takes the digital audio signals (I2S) from the Raspberry Pi and through the onboard Texas Instruments PCM5122 DAC delivers variable output (hardware volume control) analog audio to the Pi-DAC+'s Phono connectors. The PI-DAC+ also, via the Texas Instruments TPA6133A headphone amp, supports the direct use of headphones via the Pi-DAC+'s 3.5mm audio jack.

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

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+.

Pi-AMP+ Power delivery HAT EEPROM Write Enable Headphone Socket



Pi-DAC+ Audio out (for Pi-AMP+) Phono Connectors

2) The Pi-AMP+

The Pi-AMP+ is an add-on accessory for the Pi-DAC+ and Raspberry Pi B+ / RPi2.

The Pi-AMP+ takes the variable output analog audio from the Pi-DAC+ and through the onboard Texas Instruments TPA3118D2 Class D amplifier delivers audio to the Pi-AMP+'s speaker terminals delivering 2x20w stereo audio.

The Pi-AMP+ requires a 15volt power source to operate (we strongly recommend the XPPower VEH60US15). Power should be applied via the Pi-AMP+'s standard barrel connector or the two pin JST socket provided.

The Pi-AMP+, when powered, has been designed to provide power to the Raspberry Pi / Pi-DAC+ and Pi-AMP+ combination safely. You can continue to use your Pi devices without risk of damage.

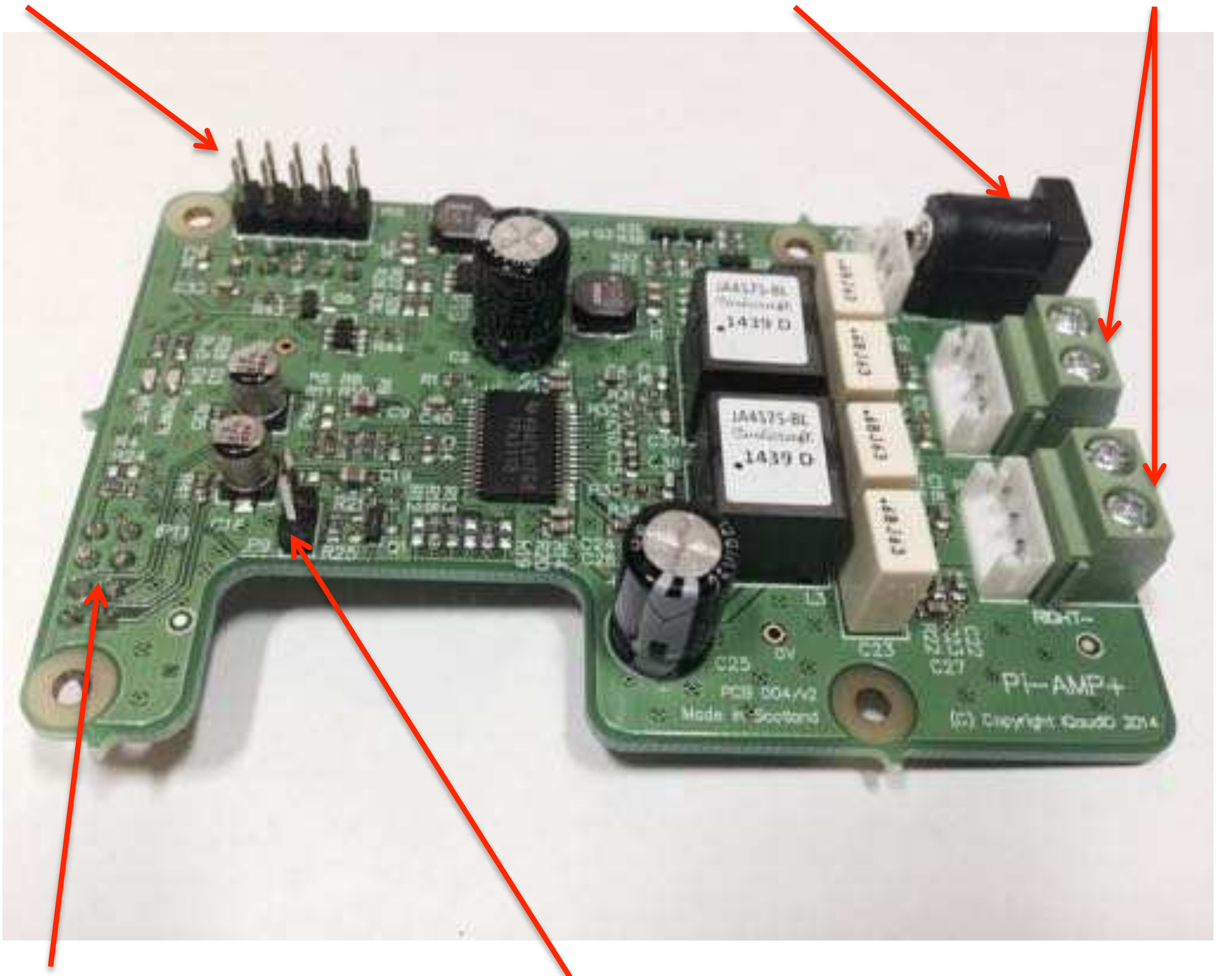
As with the Pi-DAC+, we've focused on the components selected and the attention to audio performance during PCB layout when developing the Pi-AMP+.

The Pi-AMP+ is by default muted at power on (mute LED lit). Mute state / LED is under software control (Raspberry Pi GPIO22) but mute may be overridden by shorting the Pi-AMP+'s P9 pins with a suitable jumper – note the mute LED will remain lit if P9 is shorted.

Pi Power out

15v Power In

Speaker Terminals



Audio In (from Pi-DAC+)

Mute Override (P9)

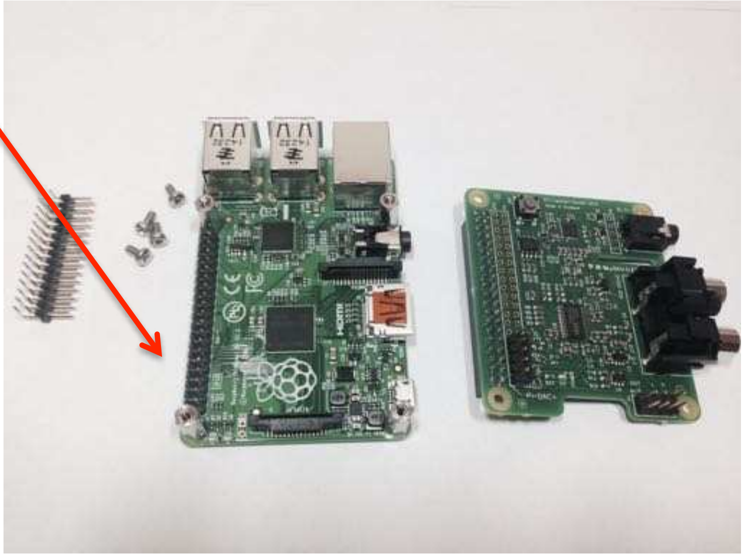
3) Connecting the Pi-DAC+ to your Raspberry Pi

WARNING: PLEASE ENSURE ALL CABLES ARE DISCONNECTED FROM THE Pi-DAC+ AND THE RASPBERRY PI BEFORE ATTEMPTING THIS PROCEDURE.

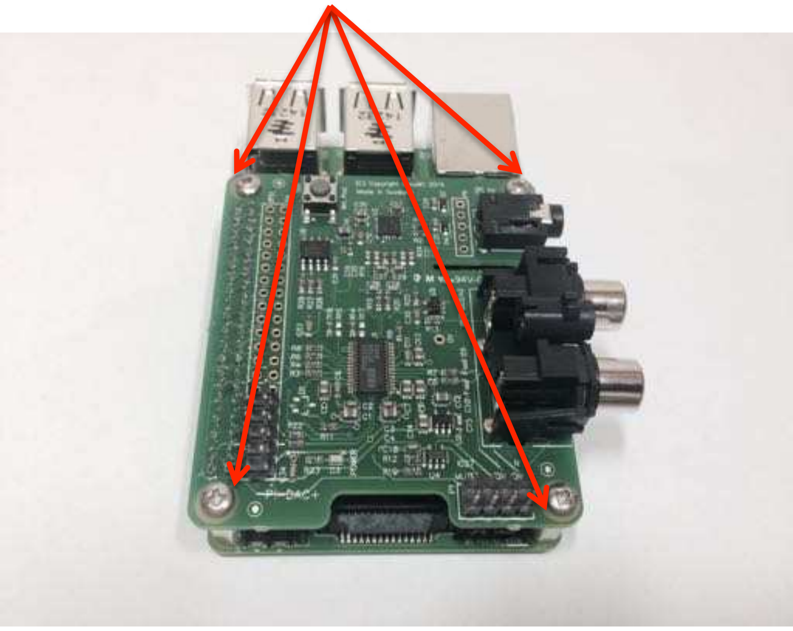
The Pi-DAC+ can be connected to the Raspberry Pi's 40way pin header without any additional soldering.

The Pi-DAC+ is supplied along with 4x 12mm PCB spacers, 8x M2.5 screws and a 30way right-angled I/O pin header (optional and only needed if you wish to use the exposed Raspberry Pi GPIO).

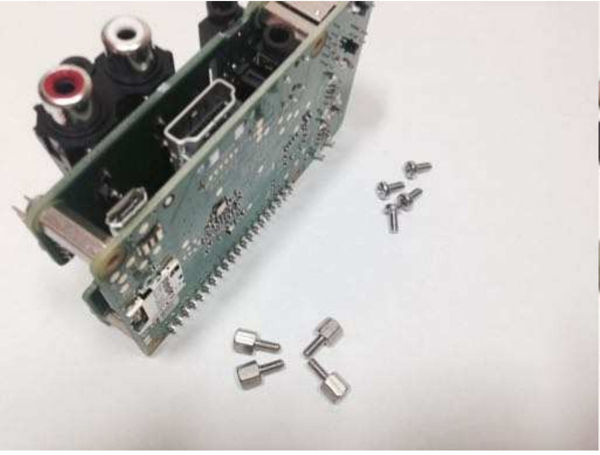
The spacers should be screwed (finger tight only) to the Raspberry Pi before adding the Pi-DAC+.

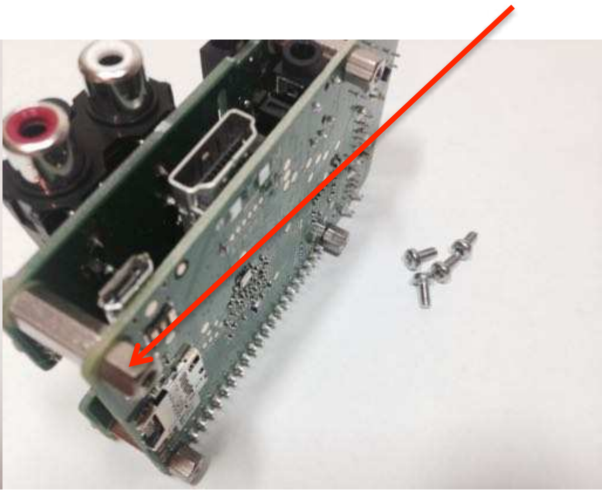


Once the Pi-DAC+ has been fitted the remaining screws can be tightened (finger tight please) as required.

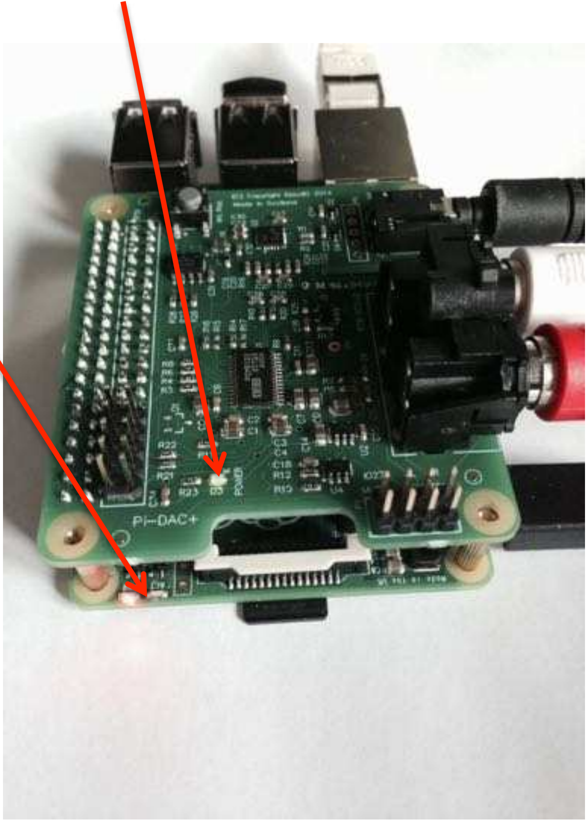


If you are using the Pi-CASE+ then replace the 4x screws beneath the Pi with the 4x 5mm PCB standoffs provided with the Pi-CASE+





Apply power to your Raspberry Pi in the normal way. You should see that the Raspberry Pi and the Pi-DAC+ both have power LEDs – this indicates that the power is successfully being applied to the Pi-DAC+.



You can now remove power and reconnect any necessary peripherals.

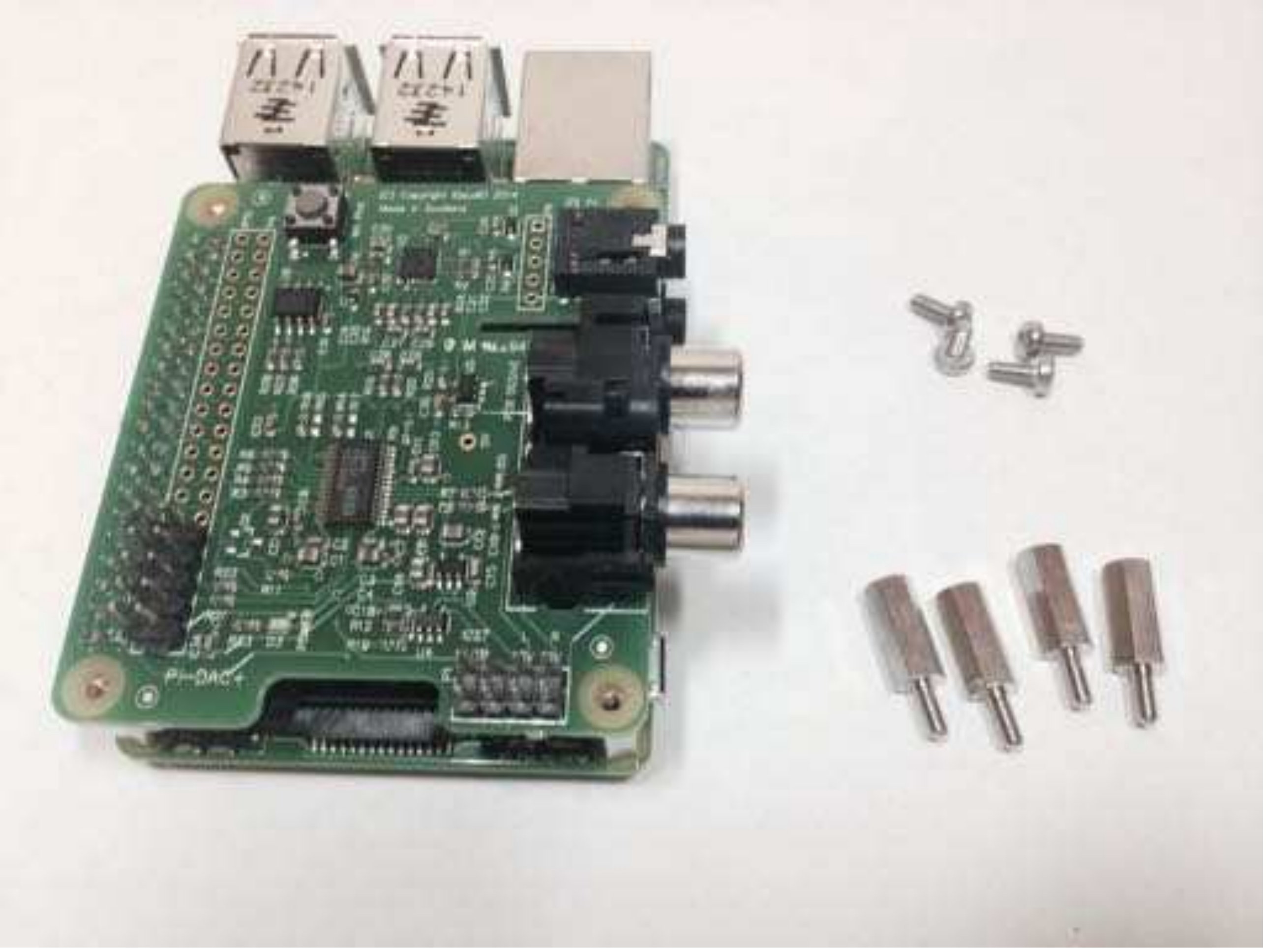
4) Connecting the Pi-AMP+ to your Pi-DAC+

WARNING: ENSURE YOU HAVE DISCONNECTED ALL POWER BEFORE ATTEMPTING THIS PROCEDURE

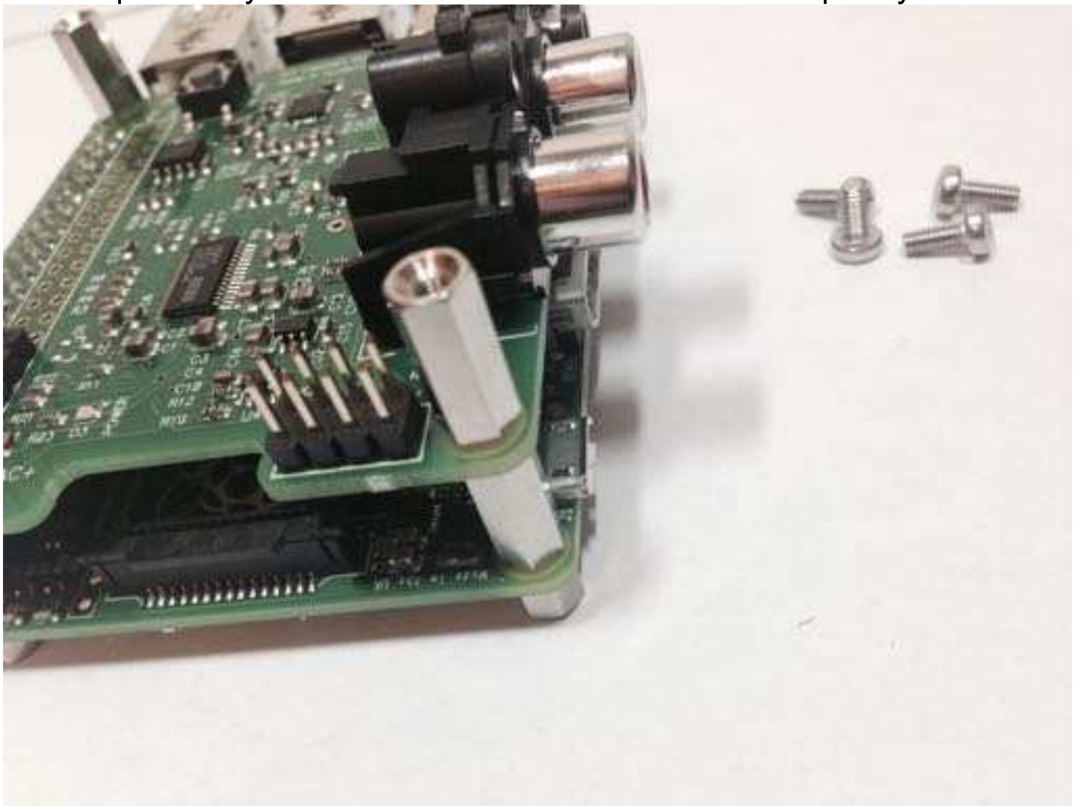
The Pi-AMP+ connects to the Pi-DAC+ through both Pi-DAC+'s two pin headers (the 2x4 audio pin header, and the 2x5 power header).

NOTE: Older Pi-DAC+ may not have the 2x5 power header fitted. Please see Section 10 for details on how to solder this to the Pi-DAC+ before continuing.

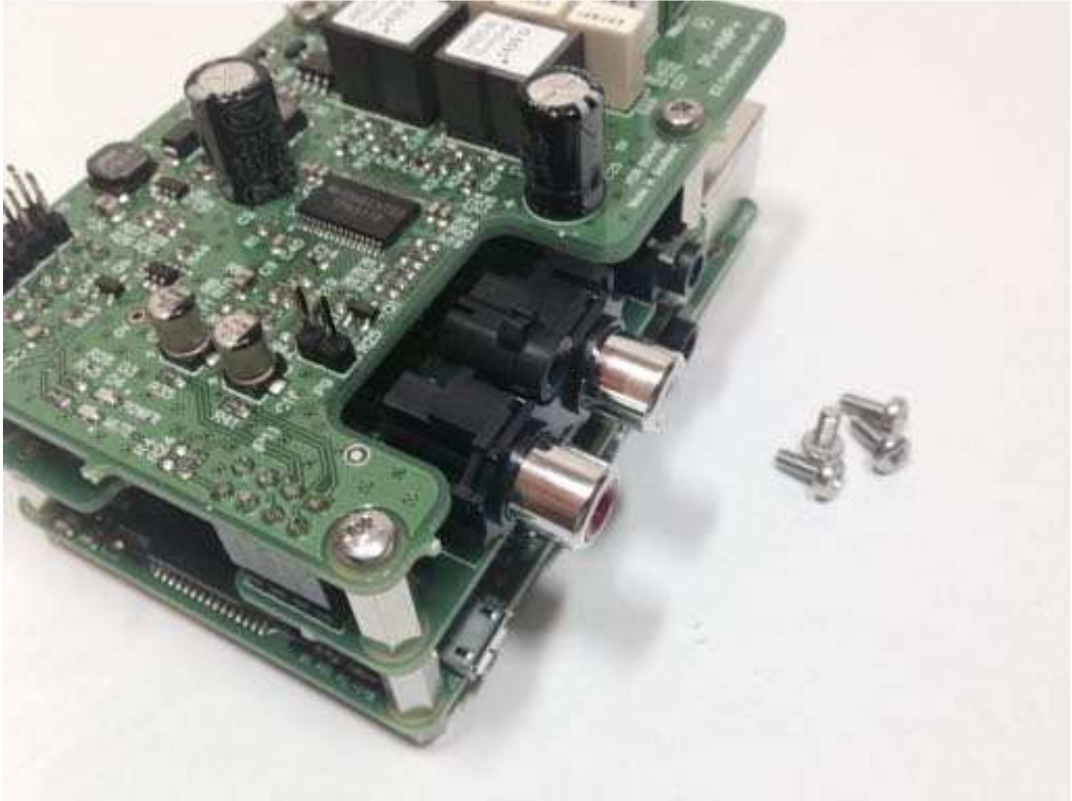
The Pi-AMP+ is supplied with 4x male/female 12mm M2.5 PCB spacers. Use these spacers to attach the Pi-DAC to the Raspberry Pi.



The male/female spacers should replace the 4 screws (these should be retained) previously used to secure the Pi-DAC+ to the Raspberry Pi.



Once the Pi-AMP+ has been fitted the remaining screws (from the Pi-DAC+) can be tightened (finger tight is fine) as required to secure the full PCB stack.

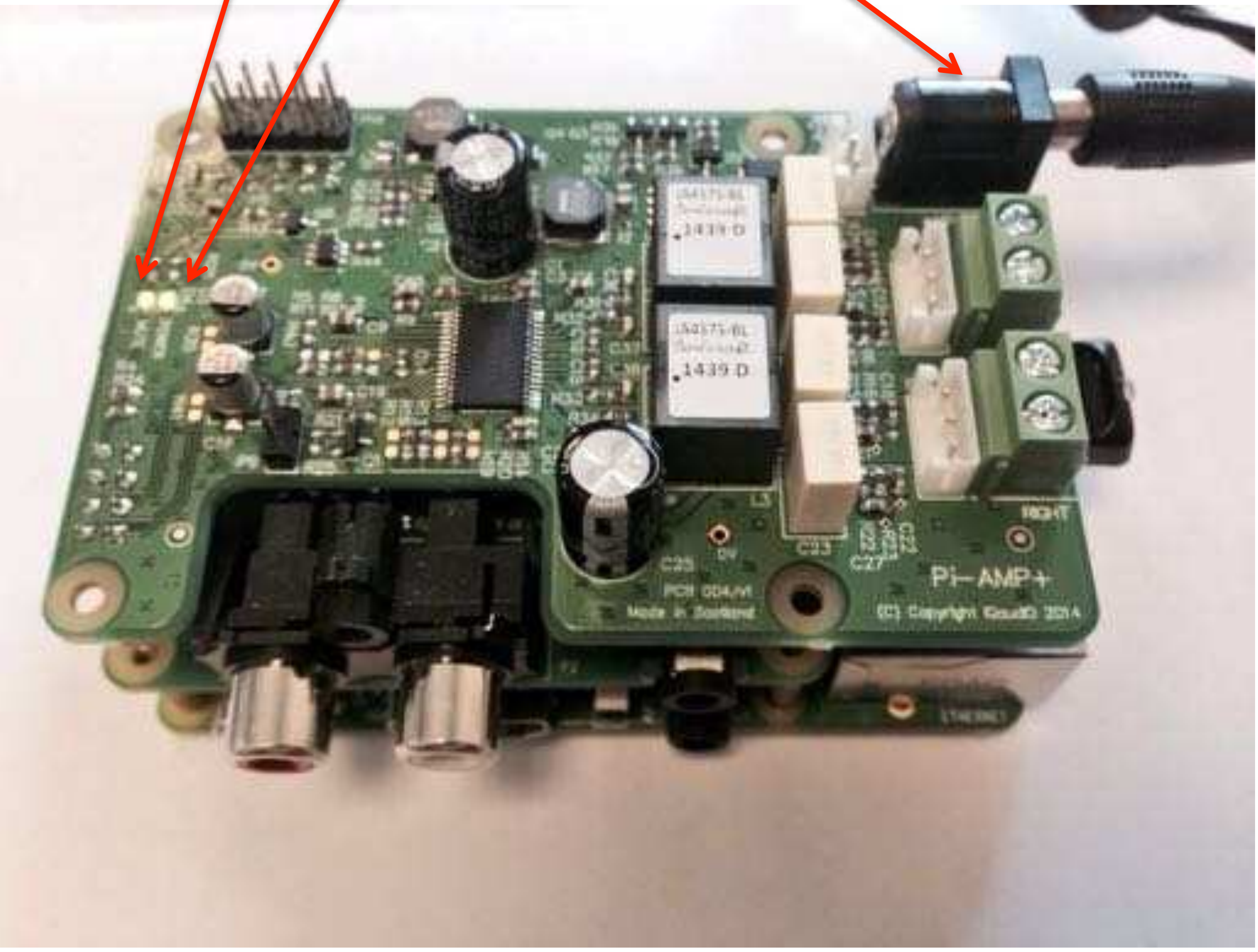


DO NOT APPLY POWER TO YOUR RASPBERRY PI VIA THE PI's MICRO USB SOCKET.

The Pi/DAC+/AMP+ combination MUST be powered via the Pi-AMP+, not the Raspberry Pi.
Once you are happy with that the Raspberry Pi / DAC+ and AMP+ are connected correctly you are ready to apply power.

The Pi-AMP+ combination is designed to be powered by the recommended XP Power VEH60US15 15v DC supply. The Power supply's output connector should be insterted into the Pi-AMP+'s DC input terminal (marked "+15V power"). Only then should the Power supply's AC input be plugged into the mains.

Once power is applied via the Pi-AMP+ you should see that the Raspberry Pi the Pi-DAC+ and the Pi-AMP+ have their power LEDs illuminated – this indicates that the power is successfully being applied to each board as expected.
The Pi-AMP+'s Mute LED will also be lit.



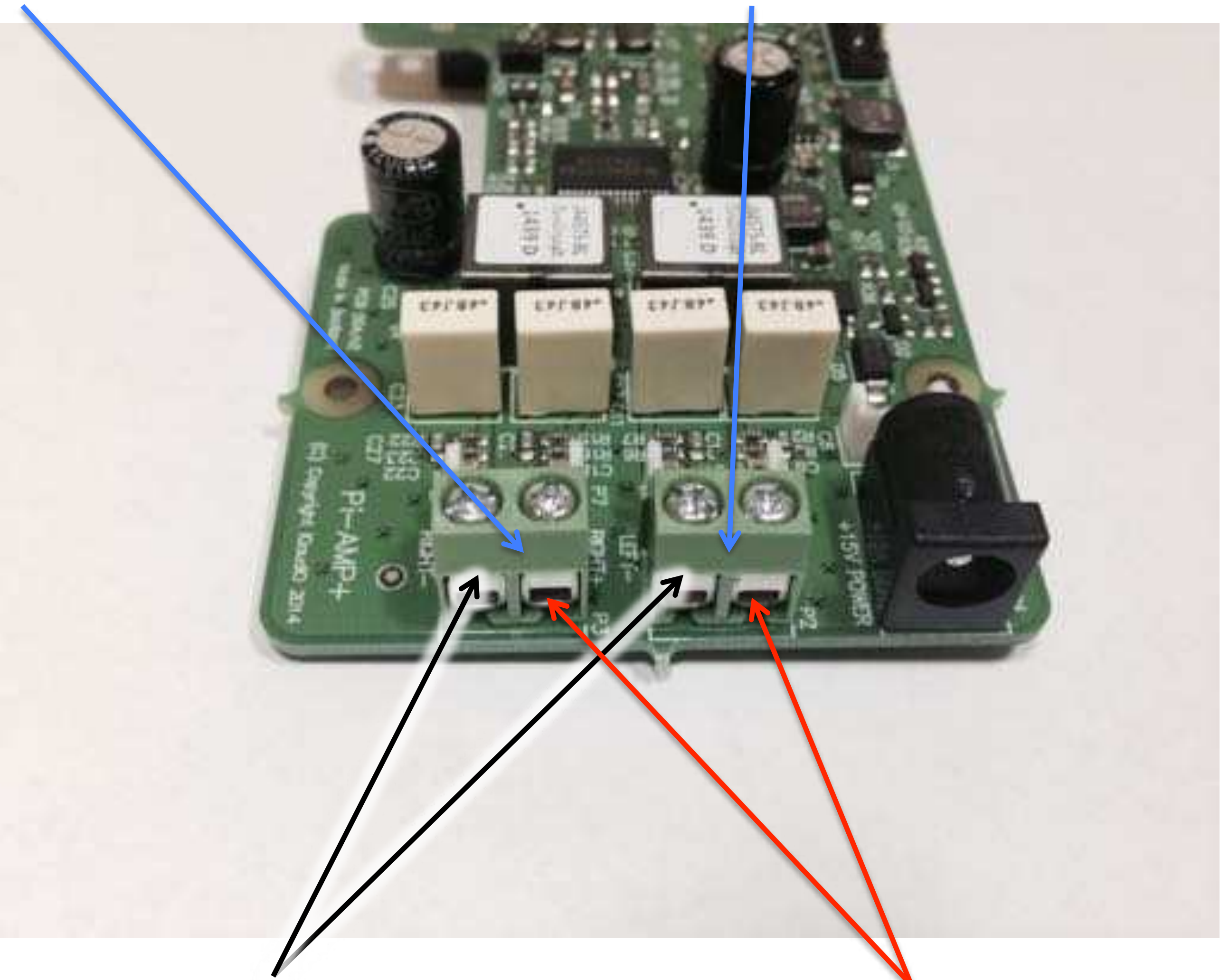
You can now remove power and reconnect any necessary peripherals you may need, including speakers.

NOTE: ALWAYS remove power BEFORE connecting speaker cable to the Pi-AMP+

For accurate performance, your loudspeakers must be connected in phase. For both left and right loudspeakers.

P3 is for connecting the RIGHT speaker speaker

P2 is for connecting the LEFT



Attach the speaker -ve terminal (often black) to the - on the amplifier output

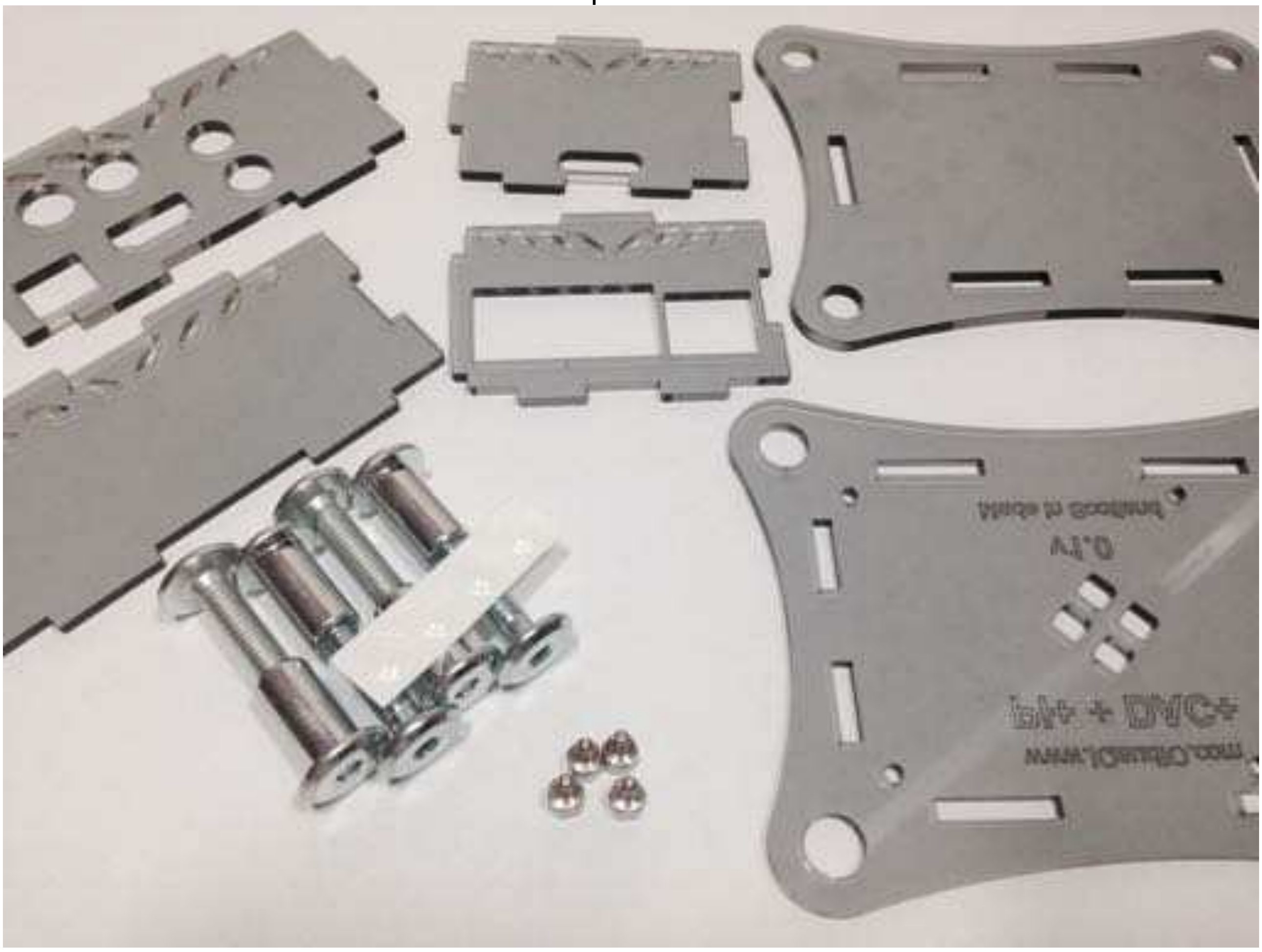
Attach the speaker +ve terminal (often red) to + on the amplifier output

5) Building the Raspberry Pi / Pi-DAC+ / Pi-AMP+ into your Pi-CASE+

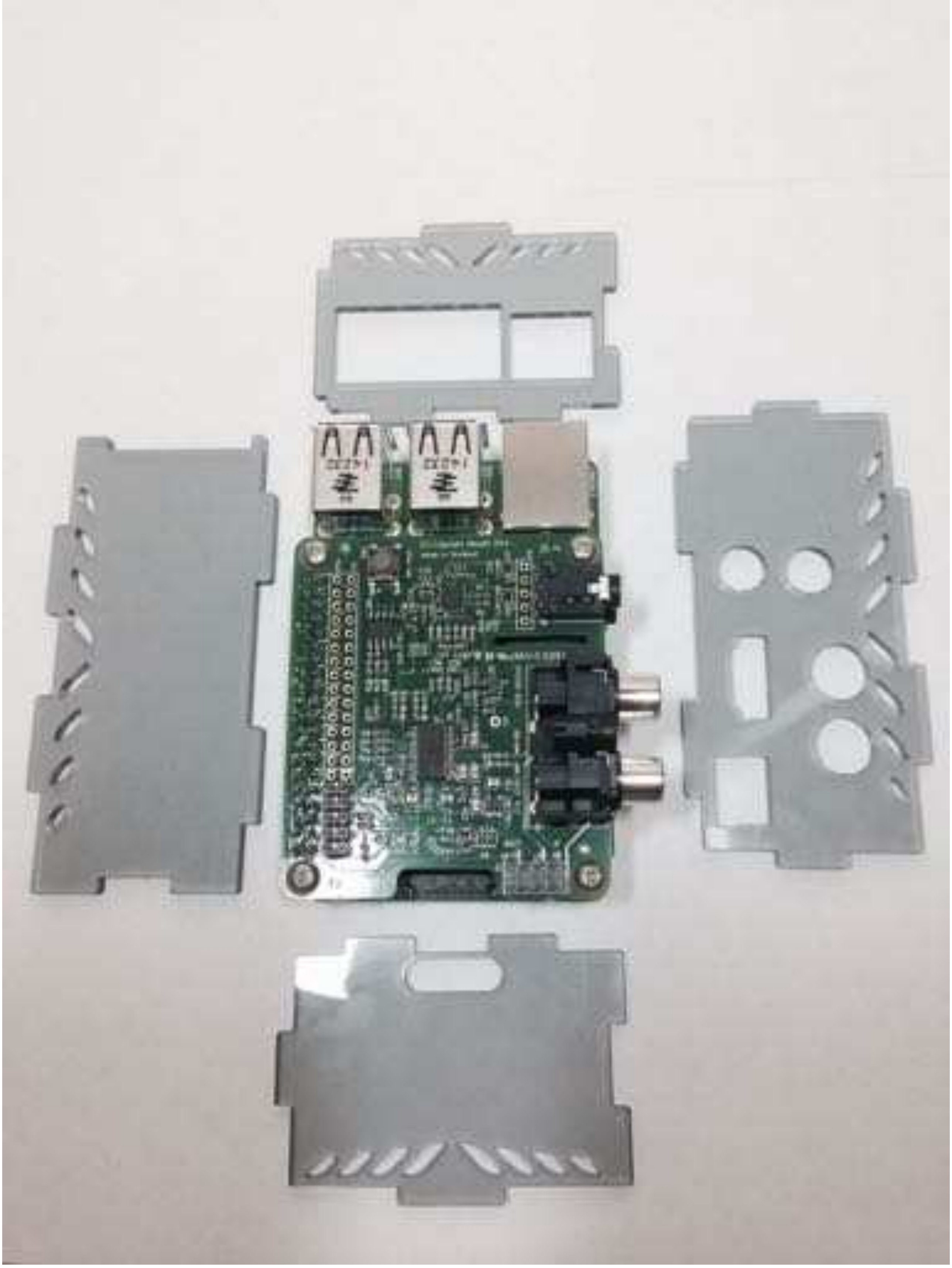
The following may prove helpful when building your Raspberry Pi / Pi-DAC+ / Pi-AMP+ etc into the appropriate Pi-CASE+.

All acrylic CASE+ installations follow the same process:

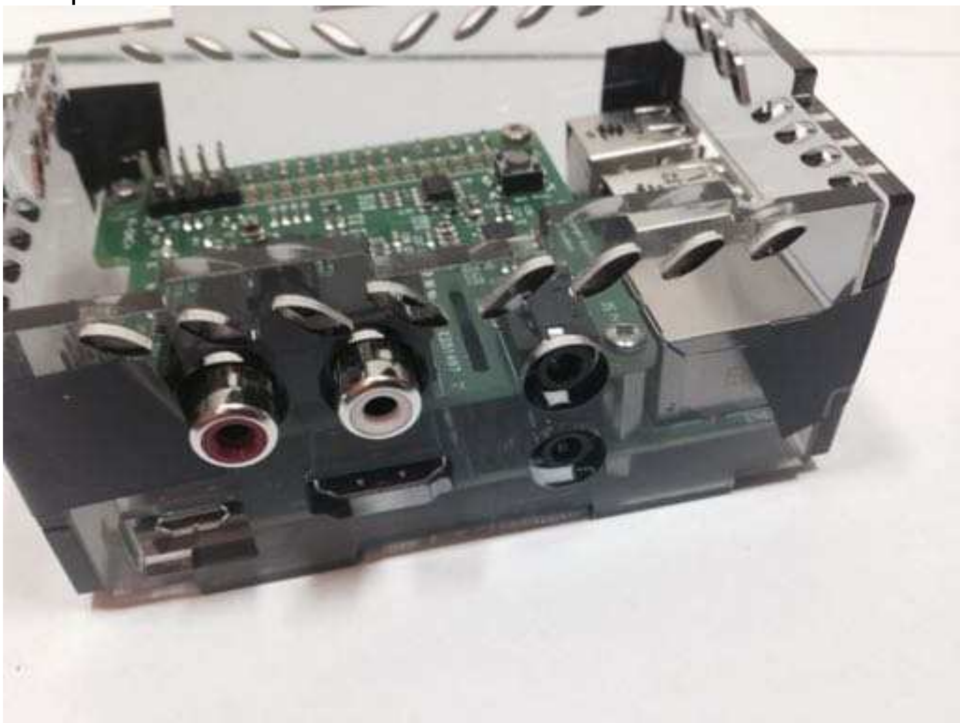
- If there is a protective film applied to either side of your acrylic, remove it now. If you build the CASE+ whilst retaining the protective film you may find that the finished look is spoilt or for transparent cases, the result may be opaque.
- Bolt the PCBs together before attempting to install the 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



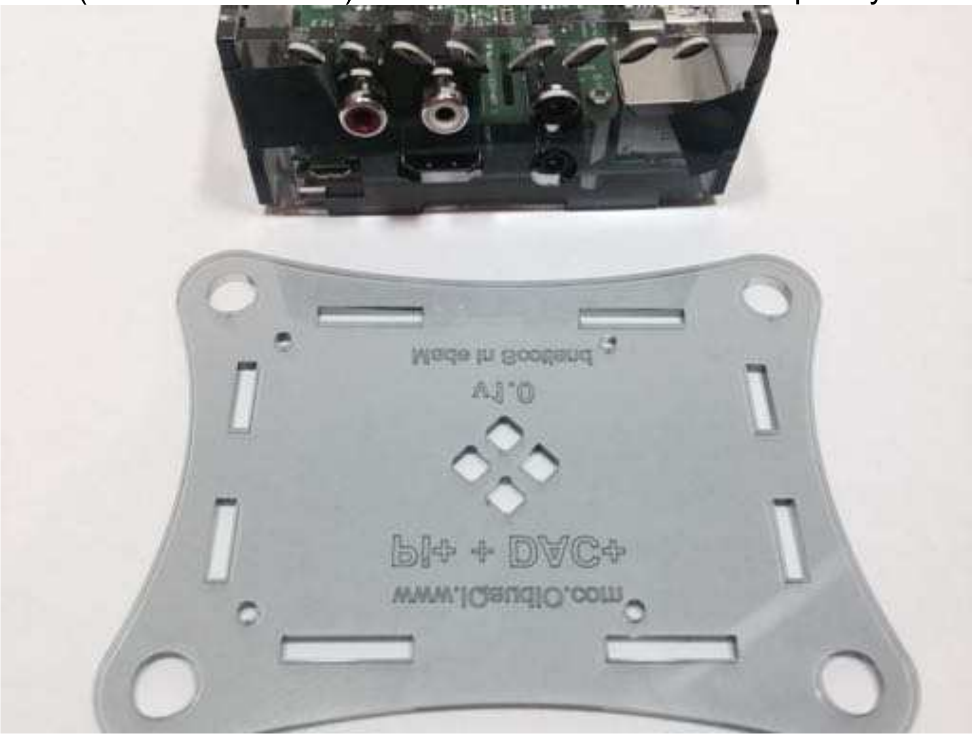
a) Lay out the CASE+ sides corresponding to the Raspberry Pi / Pi-DAC+ / Pi-AMP+



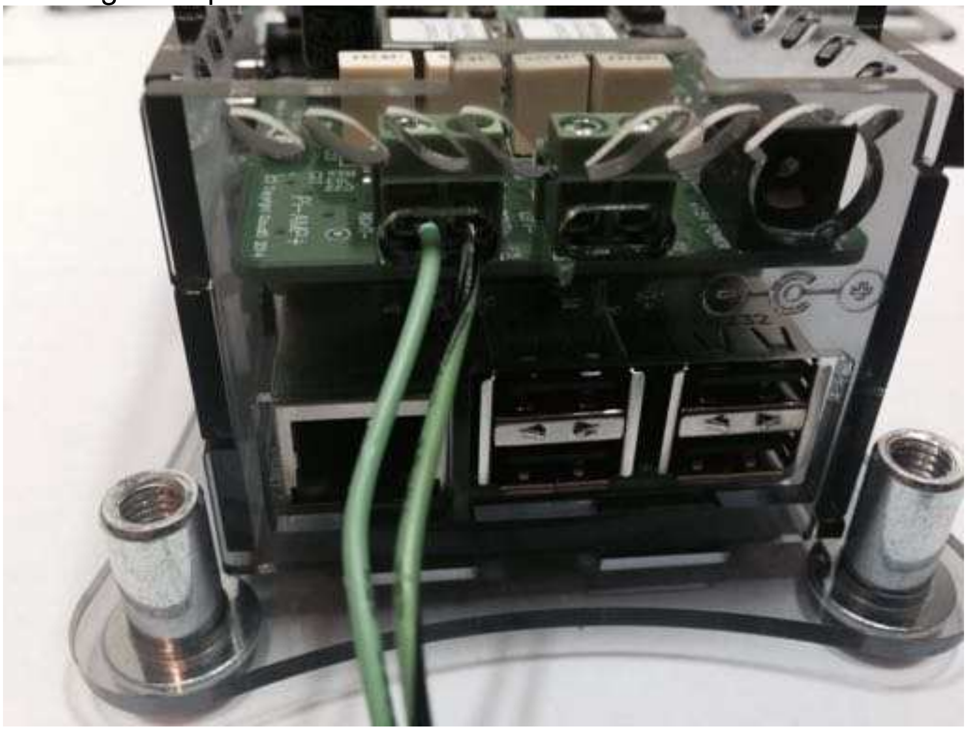
b) Then, build the sides up around the Raspberry Pi. Use sticky tape at each corner to make things simpler to assemble.



c) Add the base (with text outwards) and screw the case to the Raspberry Pi



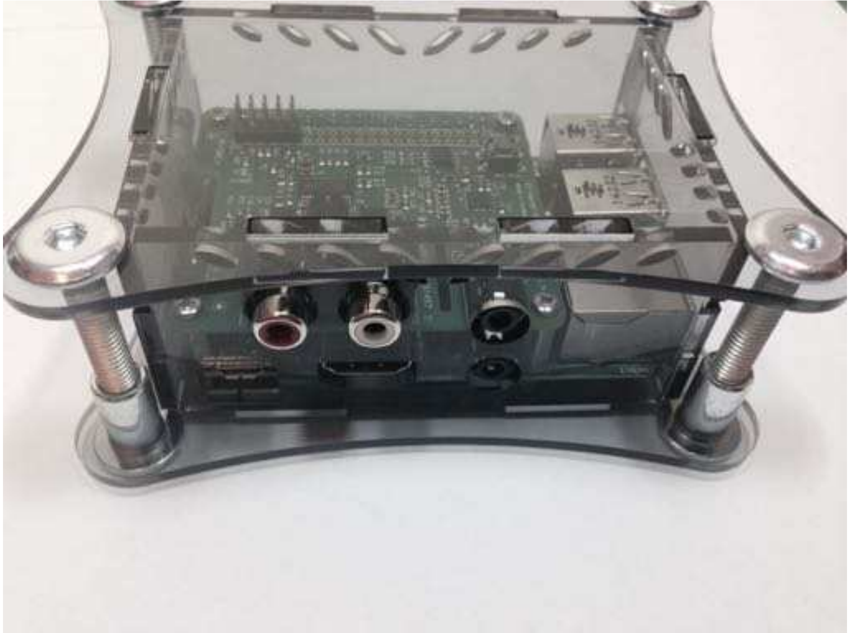
d) If you are building the CASE+ for AMP+ then now is the time to secure the speaker cables before fitting the top.



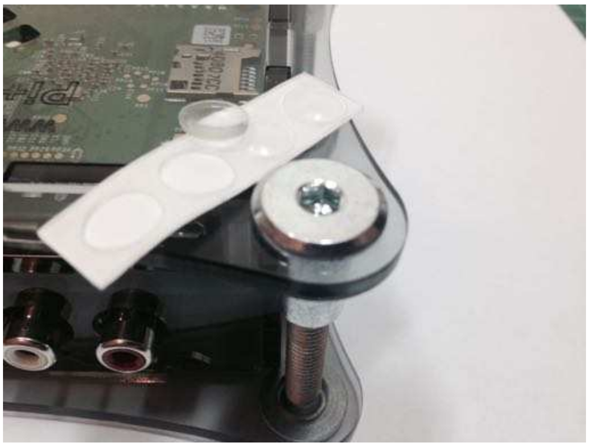
e) Pi-CASE+ top in place

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

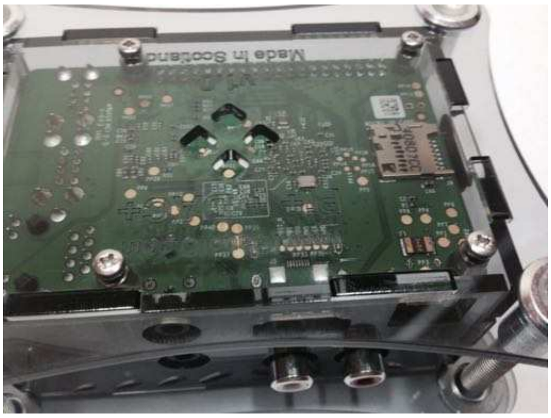
f) Completed Pi-CASE+ with Raspberry Pi and Pi-DAC+ inside



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

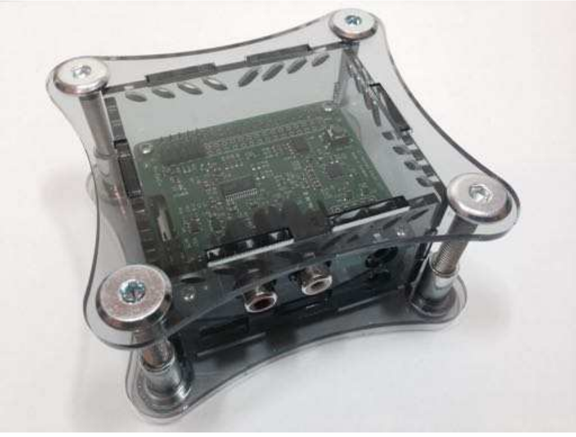


h) Now to complete screw the remaining 4 screws from the bottom of the case into the 5mm spacers inside.



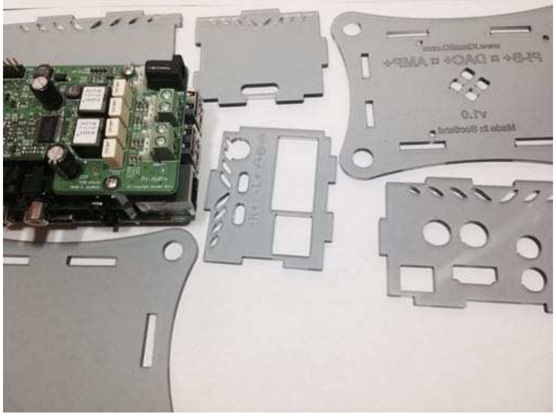
NOTE: Access to the Raspberry Pi micro SD card can be tricky due to the width of the Perspex. We suggest the use of tweezers.

The CASE+ for DAC+ and Raspberry Pi A+



The CASE+ for DAC+ and Raspberry Pi B+

The CASE+ for DAC+ / AMP+ and Raspberry Pi B+



6) Configuring Raspbian to include Pi-DAC+ support

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

- Device tree configuration should be used if you are using a 2015 Raspbian release
- You have recently updated your SD card via sudo rpi-update
- You are using a Raspberry Pi v2

A VERY different configuration is needed if you are using the older Raspbian releases from 2014 (and are not planning on updating the Linux to the latest release through sudo rpi-update) – See section 12 below.

Note - There are pre-configured image with Pi-DAC+ / Pi-AMP+ support available from www.iqaudio.com/downloads that can be used without modification.

If you would rather configure your already downloaded OS then the following stems will help…

This guide assumes you are using the standard Raspbian (Debian Wheezy) 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
```

- To ensure that the necessary kernel modules are loaded at boot, ensure your /boot/config.txt file has the entry below …

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

                              dtoverlay=iqaudio-dacplus
```

- 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 Pi-DAC+ and drivers have been installed correctly you should see something similar to:

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

7) Checking audio is working through the Pi-DAC+

Once Pi-DAC+ support is active using devicetree you should now be able to listen to music.

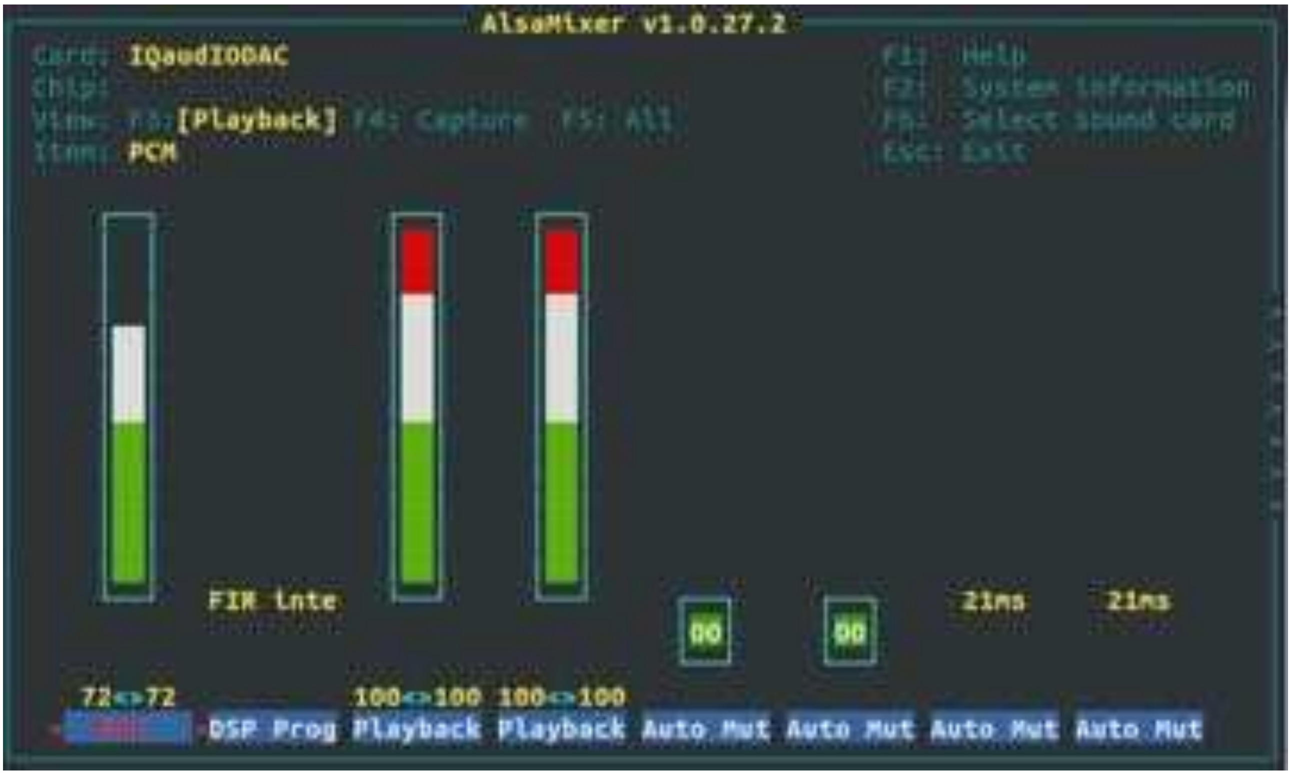
- Once the Raspberry Pi boots, log in and check that the Pi-DAC+ is visible as an ALSA audio card by running aplay –l, the output should be similar to that below. Note the device number and subdevice number – these may be useful to you when configuring media applications to access the Pi-DAC+ in your system.

```
pi@raspberrypi ~ $ aplay –l

**** List of PLAYBACK Hardware Devices ****
card 0: sndrpiiqaudioda [snd_rpi_iqaudio_dac], device 0: IQaudIO DAC HiFi pcm512x-hifi-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

- 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
```



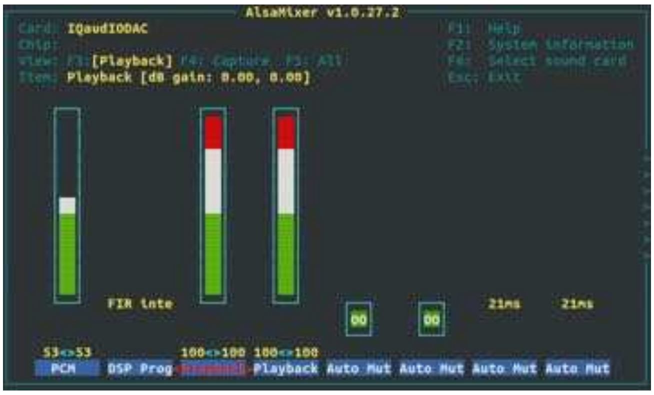
```
AlsaMixer v1.0.27.2
Card: IQaudIODAC                                  F1:  Help
Chip:                                             F2:  System information
View: F3:[Playback] F4: Capture  F5: All          F6:  Select sound card
Item: PCM                                         Esc: Exit
```

Now adjust the volume by highlighting "PCM" (Press your cursor keys left and right to select) and increase / decrease the volume by pressing your up and down cursor keys. Remember 100% = +4db, 0% is mute! (Note 0db ~ 86% on the volume range)

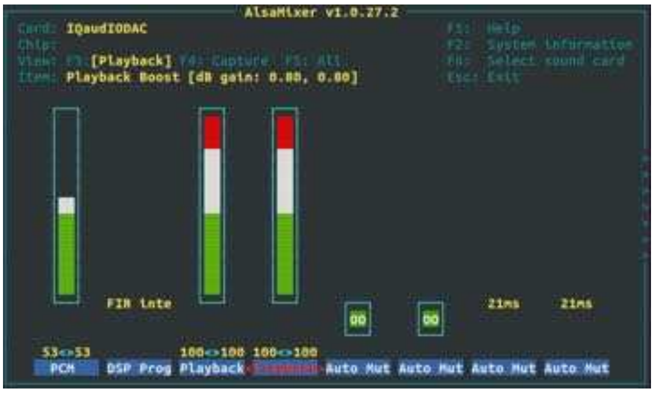
Set the volume at around 30% and from another terminal (or exit from alsamixer first)

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

- “PCM” or “Playback Digital” is the master volume control 0..100%
- “Playback” sets the maximum output to be 1vrms (0% = +0db) OR 2vrms (100% = +6db gain)
- “Playback Boost” gives an additional gain of 0.8db and should similarly be set to 100%



```
AlsaMixer v1.0.27.2
Card: IQaudIODAC                                      F1:  Help
Chip:                                                 F2:  System information
View: F3:[Playback] F4: Capture  F5: All              F6:  Select sound card
Item: Playback [dB gain: 0.00, 0.00]                  Esc: Exit
```



```
AlsaMixer v1.0.27.2
Card: IQaudIODAC                                      F1:  Help
Chip:                                                 F2:  System information
View: F3:[Playback] F4: Capture  F5: All              F6:  Select sound card
Item: Playback Boost [dB gain: 0.80, 0.80]            Esc: Exit
```

Test that audio can be heard through the Pi-DAC+. 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
http://a.files.bbci.co.uk/media/live/manifesto/audio/simulcast/hls/uk/high/ak
/bbc_radio_two.m3u8
```

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
```

If you are unable to detect the Pi-DAC+ (using aplay –l) or alsamixer doesn’t show any IQaudIO card then please re-check section 6 above and ensure that the configuration is correct.

8) Controlling the Pi-AMP+'s Mute / Unmute

MUTE state is toggled by GPIO22 on the Raspberry Pi - We'll place suitable code on out Github.com/iqaudio/tools site soon. If you want to script your own then 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
```

MUTE may also be disabled at startup by using a jumper across the P9 pins. You may experience a loud crack / pop through attached speakers on power up if this is done.

For automatic unmute on boot, add the following lines to /etc/rc.local (before the exit(0))

```
#Unmute the Pi-AMP+
echo "22" > /sys/class/gpio/export
echo "out" >/sys/class/gpio/gpio22/direction
echo "1" >/sys/class/gpio/gpio22/value
```

Another option is to create a script based on the above commands and to add this to run via cron at startup.

a. Put the commands to unmute the Pi-AMP+ into a file

e.g. `/home/pi/unmutepiamp`

b. Set execute permissions:

```
chmod a+x unmutepiamp
```

c. Run this at power up using cron.

Use the command:

```
crontab –e
```

Add the following line to crontab:

```
@reboot   /home/pi/unmuteamp
```

d. Reboot

9) Common Linux distribution configurations

Some Raspberry Pi audio/media distributions and their Pi-DAC support status are listed below. If you have other information you would like to see added to this documentation then please email info@iqaudio.com with the specific details.

a. Volumio 1.55 Pi

Volumio 1.55 contains IQaudIO Pi-DAC support. A pre-configured Raspberry Pi image is also available from www.iqaudio.com/downloads that also unmutes the Pi-AMP on startup.

b. RuneAudio

RuneAudio 0.3Beta contains IQaudIO Pi-DAC support. A pre-configured Raspberry Pi image is also available from www.iqaudio.com/downloads

c. PiCorePlayer

IQaudIO Pi-DAC+ support is built in.

d. PiMusicBox

IQaudIO Pi-DAC+ support is built in.

e. Shairport-Sync

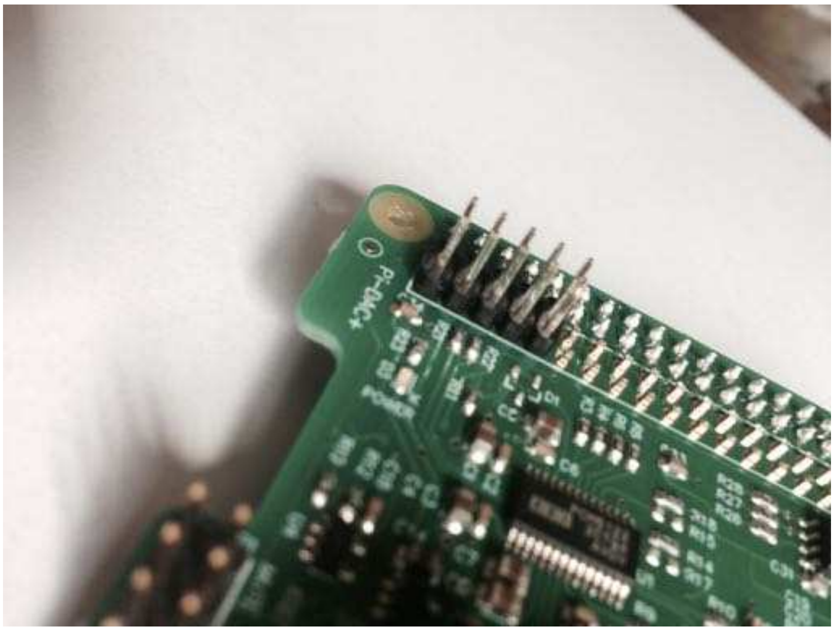
IQaudIO Pi-DAC+ support is built in. A pre-configured Raspberry Pi image is also available from www.iqaudio.com/downloads that also unmutes the Pi-AMP on startup and has built in support for a Hardware volume dial (using a rotary encoder) if needed.

10) Adding the optional PIN headers to the Pi-DAC+

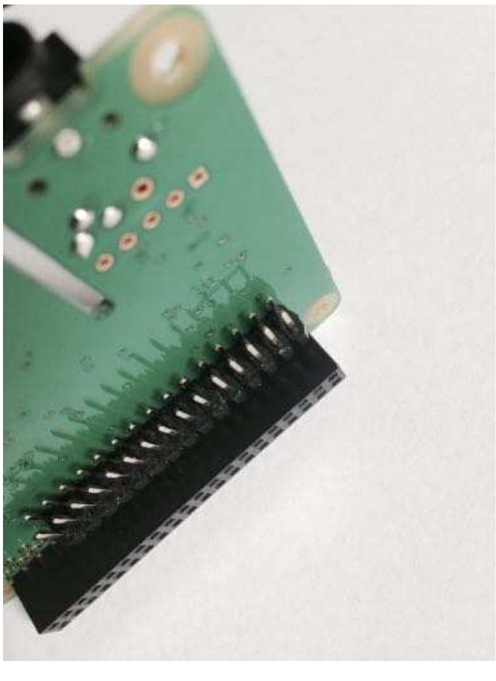
The Pi-DAC+ will work without adding any additional pin headers. The 30 way pin header supplied is for those who may want to access the B+'s GPIO and add things like rotary encoders, LCD screens or IR sensors.

It is recommended that the 30 way pin header should be added to the underside of the Pi-DAC+. This allows usage of the Pi-AMP+ whilst retaining access to the Pi's GPIO signals.

For those with very early Pi-DAC+, the 2x5 way straight pin header is designed for pins 1-10 of the 40way. These should be soldered to the TOP of the Pi-DAC+ and allow the Pi-AMP+ to power the Raspberry Pi/Pi-DAC+ in parallel.



The RIGHT ANGLED Pin headers are designed for pins 11-40 of the 40way. These should be soldered UNDERNEATH the Pi-DAC+



11) Configuring pre-2015 Raspbian images.

This section covers the configuration necessary to enable Pi-DAC+ support in pre-device tree based Linux releases. If you are using a 2015 (device tree enabled) Linux then please refer to section 6 above.

Note - There are pre-configured image with Pi-DAC+ / Pi-AMP+ support available from www.iqaudio.com/downloads that can be used without modification.

If you would rather configure your already downloaded OS then the following will help…

This guide assumes you are using the standard Raspbian (Debian Wheezy) 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.

DO NOT Update Raspbian on the SD card to the latest available (through sudo rpi-update). If you do you will need to use the devicetree configuration described in section 6 above.

- You must enable i2C support for the Pi-DAC+ to operate. On the Raspberry Pi check if you have the following file and if so delete the line containing i2c-bcm2708 and snd-soc-pcm512x.

```
pi@raspberrypi ~ $ sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

  Delete or comment out the lines containing the text below if it exists

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

- To ensure that the necessary kernel modules are loaded at boot, ensure your /etc/modules file looks the same as below…

```
pi@raspberrypi ~ $ sudo nano /etc/modules
                  snd_soc_bcm2708
                  snd_soc_bcm2708_i2s
                  bcm2708_dmaengine
                  snd_soc_pcm512x
                  snd_soc_iqaudio_dac
```

- 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 Pi-DAC+ and drivers have been installed correctly you should see something similar to:

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

12) Checking audio is working via pre-2015 Raspian and the Pi-DAC+

Once Pi-DAC+ support is active you should now be able to listen to some music.

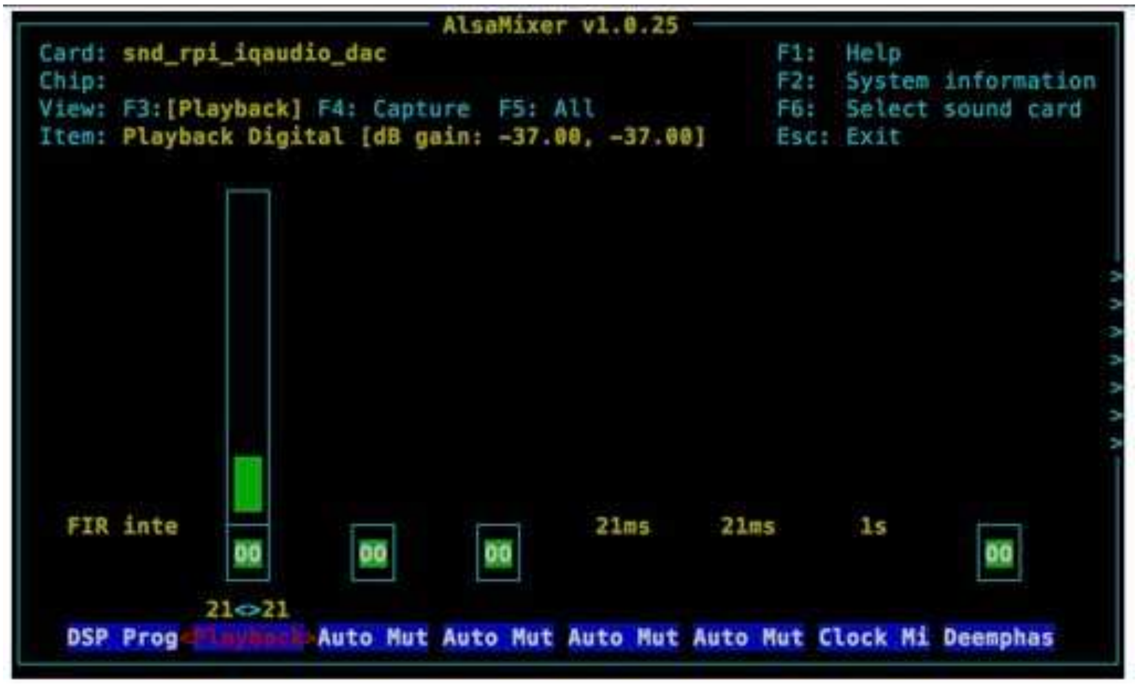
- Once the Raspberry Pi boots, log in and check that the Pi-DAC+ is visible as an ALSA audio card by running aplay –l, the output should be similar to that below. Note the device number and subdevice number – these may be useful to you when configuring media applications to access the Pi-DAC+ in your system.

```
pi@raspberrypi ~ $ aplay –l

**** List of PLAYBACK Hardware Devices ****
card 0: sndrpiiqaudioda [snd_rpi_iqaudio_dac], device 0: IQaudIO DAC HiFi pcm512x-hifi-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

- 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
```



```
AlsaMixer v1.0.25
Card: snd_rpi_iqaudio_dac                                   F1:  Help
Chip:                                                       F2:  System information
View: F3:[Playback] F4: Capture  F5: All                    F6:  Select sound card
Item: Playback Digital [dB gain: -37.00, -37.00]            Esc: Exit
```

Now adjust the volume by highlighting "Playback Digital" (Press your cursor keys left and right to select) and increase / decrease the volume by pressing your up and down cursor keys. Remember +0dB is full volume, -103.5dB is mute! (Note 0db ~ 86% on the volume range)

Set the volume at around 30% and from another terminal (or exit from alsamixer first)

Test that audio can be heard through the Pi-DAC+. 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
http://www.bbc.co.uk/radio/listen/live/r2_aaclca.pls

MPlayer svn r34540 (Debian), built with gcc-4.6 (C) 2000-2012 MPlayer Team

Playing http://www.bbc.co.uk/radio/listen/live/r2_aaclca.pls.
Resolving www.bbc.co.uk for AF_INET...
Connecting to server www.bbc.co.uk[212.58.244.66]: 80...
Resolving open.live.bbc.co.uk for AF_INET...
Connecting to server open.live.bbc.co.uk[212.58.244.80]: 80...

Cache size set to 320 KBytes


Playing http://bbcmedia.ic.llnwd.net/stream/bbcmedia_lc1_radio2.
Resolving bbcmedia.ic.llnwd.net for AF_INET...
Connecting to server bbcmedia.ic.llnwd.net[178.79.224.66]: 80...

Name   : BBC Radio 2
Genre  : BBC Radio Live
Public : yes
Bitrate: 128kbit/s
Cache size set to 320 KBytes
Cache fill:  0.00% (0 bytes)
ICY Info: StreamTitle='';
Cache fill: 15.00% (49152 bytes)

AAC file format detected.
==========================================================================
Opening audio decoder: [ffmpeg] FFmpeg/libavcodec audio decoders
libavcodec version 53.35.0 (external)
Mismatching header version 53.32.2
AUDIO: 44100 Hz, 2 ch, s16le, 0.0 kbit/0.00% (ratio: 0->176400)
Selected audio codec: [ffaac] afm: ffmpeg (FFmpeg AAC (MPEG-2/MPEG-4 Audio))
==========================================================================
AO: [alsa] 44100Hz 2ch s16le (2 bytes per sample)
Video: no video
Starting playback...
A:   0.8 (00.8) of 0.0 (unknown)  5.8% 20%
```

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
```

If you are unable to detect the Pi-DAC+ (using aplay –l) or alsamixer doesn't show any IQaudio card then please re-check section 6 above and ensure that the configuration is correct. The following commands may help you during debug:

```
pi@raspberrypi ~ $ dmesg
    [    6.549255] pcm512x 1-004c: No SCLK, using BCLK: -2
    [    6.556344] sd 0:0:0:0: Attached scsi generic sg0 type 0
    [    6.570452] pcm512x 1-004c: Completed initialisation - pcm512x_probe
    …
    [   13.302381] snd-rpi-iqaudio-dac snd-rpi-iqaudio-dac.0:  pcm512x-hifi <-> bcm2708-i2s.0 mapping ok
```

```
pi@raspberrypi ~ $ lsmod
    pi@volumio:~$ lsmod
    Module                  Size  Used by
    nfsd                  243497  2
    ipv6                  304050  36
    snd_soc_iqaudio_dac     2493  0
    bcm2708_dmaengine       5108  1
    virt_dma                2273  1 bcm2708_dmaengine
    snd_soc_pcm512x         8913  1
    sg                     18996  0
    snd_soc_bcm2708_i2s     5474  2
    regmap_mmio             2806  1 snd_soc_bcm2708_i2s
    snd_soc_core          131292  3 snd_soc_pcm512x,snd_soc_iqaudio_dac,snd_soc_bcm2708_i2s
    regmap_spi              1897  2 snd_soc_pcm512x,snd_soc_core
    snd_pcm                81593  2 snd_soc_core,snd_soc_iqaudio_dac
    snd_page_alloc          5156  1 snd_pcm
    regmap_i2c              1645  2 snd_soc_pcm512x,snd_soc_core
    snd_compress            8060  1 snd_soc_core
    snd_seq                53769  0
    snd_timer              20133  2 snd_pcm,snd_seq
    snd_seq_device          6473  1 snd_seq
    leds_gpio               2059  0
    led_class               3688  1 leds_gpio
    snd                    61291  6 snd_soc_core,snd_timer,snd_pcm,snd_seq,snd_seq_device,snd_compress
    spi_bcm2708             4728  0
    i2c_bcm2708             3997  0
```

13) Advanced details

The following section is for more advanced users.

Software:

- Some Linux kernel versions will report the Hardware volume mixer as “Playback Digital”. On upstream/more recent versions of the kernel this Hardware volume mixer control is known as “PCM”. Please see alsamixer for the name used in your particular kernel version. See also the section 7 above about volume control names and default settings.
- Not all features of the PCM512x device driver have been fully tested. Changing settings other than the standard Mixer may NOT provide the expected result.
- We expect the following Raspberry Pi GPIO to be used by IQaudIO products - GPIO22 (AMP+ mute / unmute support).
- We have used GPIO23/24 for our Rotary Encoder (volume control) software.

Hardware:

Pi-DAC+ SIGNALS USED

	Pin#	NAME			NAME	Pin#	
	1	**3.3v** (DC Power)	□	¤	(DC Power) **5v**	2	
Used by IQaudIO Pi-DAC+ (I2C) ------>	3	**GPIO2** (I2C SDA1)	¤	¤	(DC Power) **5v**	4	
Used by IQaudIO Pi-DAC+ (I2C) ------>	5	**GPIO3** (I2C SCL1)	¤	¤	(GROUND) **0v**	6	
(Pi-DAC uses I2C address 0x4C)	7	**GPIO4** (GCLK)	¤	¤	(TXD0) **GPIO14**	8	
	9	**0v** (GROUND)	¤	¤	(RXD0) **GPIO15**	10	
	11	**GPIO17** (GEN0)	¤	¤	(I2S CLK) **GPIO18**	12	<----- Used by IQaudIO Pi-DAC+ (I2S)
	13	**GPIO27** (GEN2)	¤	¤	(GROUND) **0v**	14	
Used by IQaudIO Pi-AMP+ (Mute) ------>	15	**GPIO22** (GEN3)	¤	¤	(GEN4) **GPIO23**	16	<----- Used by IQaudIO Rotary Encoder
	17	**3.3v** (DC Power)	¤	¤	(GEN5) **GPIO24**	18	<----- Used by IQaudIO Rotary Encoder
	19	**GPIO10** (SPI_MOSI)	¤	¤	(GROUND) **0v**	20	
	21	**GPIO9** (SPI_MISO)	¤	¤	(GEN6) **GPIO25**	22	<----- Used by IQaudIO IR Receiver
	23	**GPIO11** (SPI_CLK)	¤	¤	(SPI CE0) **GPIO8**	24	
	25	**0v** (GROUND)	¤	¤	(SPI CE1) **GPIO7**	26	
Used by IQaudIO Pi-DAC+ EEPROM ------>	27	**ID_SD** (HAT EEPROM)	¤	¤	(HAT EEPROM) **ID_SC**	28	<----- Used by IQaudIO Pi-DAC+ EEPROM
	29	**GPIO5**	¤	¤	(GROUND) **0v**	30	
	31	**GPIO6**	¤	¤	**GPIO12**	32	
	33	**GPIO13**	¤	¤	(GROUND) **0v**	34	
Used by IQaudIO Pi-DAC+ (I2S) ------>	35	**GPIO19** (I2S LRCLK)	¤	¤	**GPIO16**	36	
	37	**GPIO26**	¤	¤	(I2S DIN) **GPIO20**	38	<----- Used by IQaudIO Pi-DAC+ (I2S)
	39	**0v** (GROUND)	¤	¤	(I2S DOUT) **GPIO21**	40	<----- Used by IQaudIO Pi-DAC+ (I2S)

The Pi-DAC+ re-exposes the majority of the Raspberry Pi signals:

Pi-DAC+ SIGNALS NOT PASSED THROUGH / NOT TO BE USED

Pin#	NAME			NAME	Pin#
1	**3.3v** (DC Power)	□	¤	(DC Power) **5v**	2
3	**GPIO2** (I2C SDA1)	¤	¤	(DC Power) **5v**	4
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
13	**GPIO27** (GEN2)	¤	¤	(GROUND) **0v**	14
15	**GPIO22** (GEN3)	¤	¤	(GEN4) **GPIO23**	16
17	**3.3v** (DC Power)	¤	¤	(GEN5) **GPIO24**	18
19	**GPIO10** (SPI_MOSI)	¤	¤	(GROUND) **0v**	20
21	**GPIO9** (SPI_MISO)	¤	¤	(GEN6) **GPIO25**	22
23	**GPIO11** (SPI_CLK)	¤	¤	~~(SPI CE0) **GPIO8**~~	~~24~~
25	**0v** (GROUND)	¤	¤	~~(SPI CE1) **GPIO7**~~	~~26~~
~~27~~	~~**ID_SD** (HAT EEPROM)~~	¤	¤	~~(HAT EEPROM) **ID_SC**~~	~~28~~
29	**GPIO5**	¤	¤	(GROUND) **0v**	30
31	**GPIO6**	¤	¤	**GPIO12**	32
33	**GPIO13**	¤	¤	(GROUND) **0v**	34
35	**GPIO19** (I2S LRCLK)	¤	¤	**GPIO16**	36
37	**GPIO26**	¤	¤	(I2S DIN) **GPIO20**	38
39	**0v** (GROUND)	¤	¤	(I2S DOUT) **GPIO21**	40

Signals used by Pi-AMP+ (To back power Pi and Pi-DAC+)

Pin#	NAME			NAME	Pin#
1	**3.3v** (DC Power)	□	¤	(DC Power) **5v**	2
3	**GPIO2** (I2C SDA1)	¤	¤	(DC Power) **5v**	4
5	**GPIO3** (I2C SCL1)	¤	¤	(GROUND) **0v**	6
7	**GPIO4** (GCLK)	¤	¤	~~(TXD0) **GPIO14**~~	~~8~~
9	**0v** (GROUND)	¤	¤	~~(RXD0) **GPIO15**~~	~~10~~

Signals passed through by Pi-AMP+

Pin#	NAME			NAME	Pin#
1	**3.3v** (DC Power)	□	¤	(DC Power) **5v**	2
3	**GPIO2** (I2C SDA1)	¤	¤	(DC Power) **5v**	4
5	**GPIO3** (I2C SCL1)	¤	¤	(GROUND) **0v**	6
7	**GPIO4** (GCLK)	¤	¤	~~(TXD0) **GPIO14**~~	~~8~~
9	**0v** (GROUND)	¤	¤	~~(RXD0) **GPIO15**~~	~~10~~

14) Using RPi2 and external USB devices

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

```
    max_usb_current=1
```

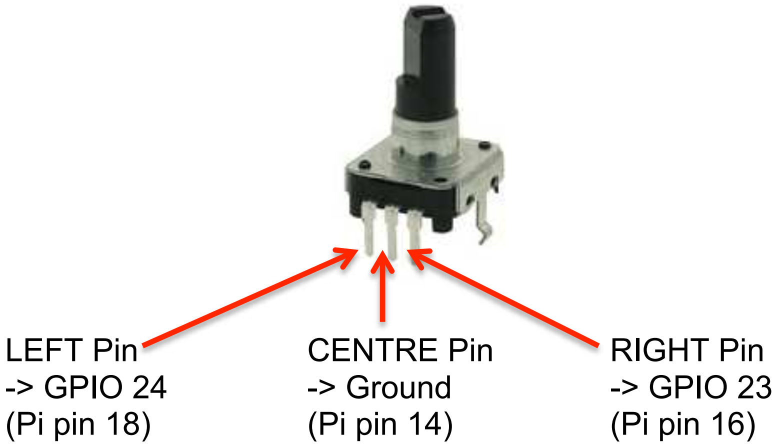
Please ensure your Pi power supply is able to deliver ~2amp if you are powering via mini USB and enable max_usb_current

15) Adding a Rotary Encoder

The following section shows the recommended pins / GPIO to use when adding a Rotary Encoder (physical volume control) and using the IQ_rot utility (from our github site: www.github.com/iqaudio/tools) to control the ALSA audio volume level.

IQ_rot utility assumes that a 3 pin rotary encoder is connected to the Raspberry Pi as follows.

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



LEFT Pin
-> GPIO 24
(Pi pin 18)

CENTRE Pin
-> Ground
(Pi pin 14)

RIGHT Pin
-> GPIO 23
(Pi pin 16)

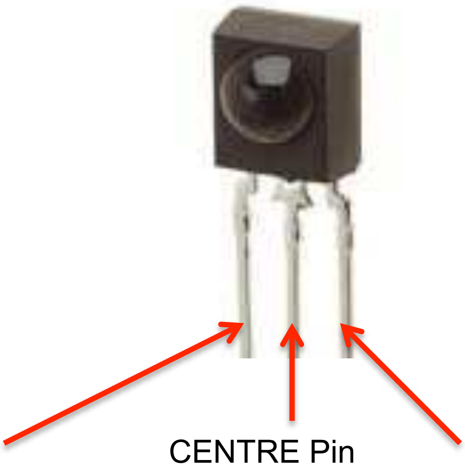
IQ_rot needs to be run with root privileges (sudo IQ_rot) and needs the WiringPi library to be pre-installed. If being run at startup (in /etc/rc.local) make sure that the call is done as a separate process (for example as: /usr/pi/iqaudio/tools/IQ_rot&)

By default IQ_rot uses the mixer name PCM – if the driver's volume control mixer is not "PCM" then the IQ_rot source code will need to be modified to access the correct mixer.

16) Adding an IR (Remote Control) Sensor

Similar to the Rotary Encoder example above, it is possible to control the Pi-DAC+ volume control through the use of a remote control and IR sensor (a Vishay tsop4838 for example). This requires a little more setup in terms of software than the Rotary Encoder, please refer to the LIRC setup guides but you should ensure that LIRC is set to use GPIO25 as an input. Example code / utility is available from our github site: www.github.com/iqaudio/tools

For H/W setup we recommend that the IR sensor should be connected as below.



LEFT Pin
-> GPIO 25
(Pi pin 22)

CENTRE Pin
-> Ground
(Pi pin 20)

RIGHT Pin
-> 3v3
(Pi pin 17)

IQ_ir needs to be run with root privileges (sudo IQ_ir) and needs the WiringPi library to be pre-installed. If being run at startup (in /etc/rc.local) make sure that the call is done as a separate process (for example as: /usr/pi/iqaudio/tools/IQ_ir&)

By default IQ_ir uses the mixer name PCM – if the driver’s volume control mixer is not “PCM” then the IQ_ir source code will need to be modified to access the correct mixer.