

Audio Configuration for the CSB503HTA

The pcm5102a DAC on the csb503hta is well supported in the Raspberry Pi Linux kernel. The minimal configuration required to get things running is described below.

Raspbian Setup:

Setting up the Proper DAC Devicetree Overlay

The file `/boot/config.txt` must be edited to add the following line (use `sudo nano /boot/config.txt` or your favorite editor):

```
dtoverlay=hifiberry-dac
```

This properly configures the devicetree to load the PCM5102a codec. Reset after making this change.

Checking the System

On startup, or after running “`dmesg`”, the following lines are normal and represent proper functionality:

```
[ 3.385151] snd-hifiberry-dac sound: ASoC: CPU DAI (null) not registered
[ 3.392990] snd-hifiberry-dac sound: snd_soc_register_card() failed: -517
[ 3.789173] snd-hifiberry-dac sound: ASoC: CPU DAI (null) not registered
[ 3.809391] snd-hifiberry-dac sound: snd_soc_register_card() failed: -517
[ 3.818401] snd-hifiberry-dac sound: pcm5102a-hifi <-> 3f203000.i2s mapping ok
```

The important line to look for is “`pcm5102a-hifi <-> 3f203000.i2s mapping ok`”.

It is also useful to run the following:

```
pi@raspberrypi ~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: sndrpihifiberry [snd_rpi_hifiberry_dac], device 0: HifiBerry DAC HiFi
pcm5102a-hifi-0 []
  Subdevices: 1/1
    Subdevice #0: subdevice #0
card 1: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 8/8
    Subdevice #0: subdevice #0
    Subdevice #1: subdevice #1
    Subdevice #2: subdevice #2
    Subdevice #3: subdevice #3
    Subdevice #4: subdevice #4
    Subdevice #5: subdevice #5
    Subdevice #6: subdevice #6
    Subdevice #7: subdevice #7
card 1: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
```

You will see “`sndrpihifiberry`” at card 0, indicating that the csb503hta DAC is working properly.

Testing the Audio

There is a built in utility for testing speakers. By default it plays white noise, but can be made to play a sin wave of specified frequency at a given sample rate.

Test a sin wave of default frequency at 192K sample rate with the following command:

```
speaker-test -t sin -c 2 -r 192000
```

Test a sin wave of default frequency at 176.4K sample rate with the following command:

```
speaker-test -t sin -c 2 -r 176400
```

This will play sound on all ports: RCA analog, optical, and coax spdif. Currently the driver only supports stereo audio. Trying to play mono audio will result in an error:unsupported.

Setup For Volumio

Setup for Volumio is simple.

Download and flash SD card with Volumio.

Insert SD Card into pi and boot.

On another machine or phone with Bonjour installed, type <http://volumio.local> into a browser.

Click the gear in the upper-right corner (settings)

Choose "Playback Options"

You will see a screen similar to that below:



Select I2S "ON".

Select DAC Model "Hifiberry DAC".

Select Audio Output "Hifiberry DAC". **(Correction - Not Hifiberry Digi as shown in the picture)**

Click "Save".

You will be prompted for reset, say yes.

Enjoy!

Setup For RuneAudio

Setup for RuneAudio requires either keyboard and monitor or ssh.

Download and flash SD card with PiCorePlayer.

Insert SD Card into pi and boot.

Via ssh (root@runeaudio.local, pass=rune) or attached keyboard and monitor, edit /boot/config.txt.

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

Uncomment the line "dtoverlay=hifiberry-dac".

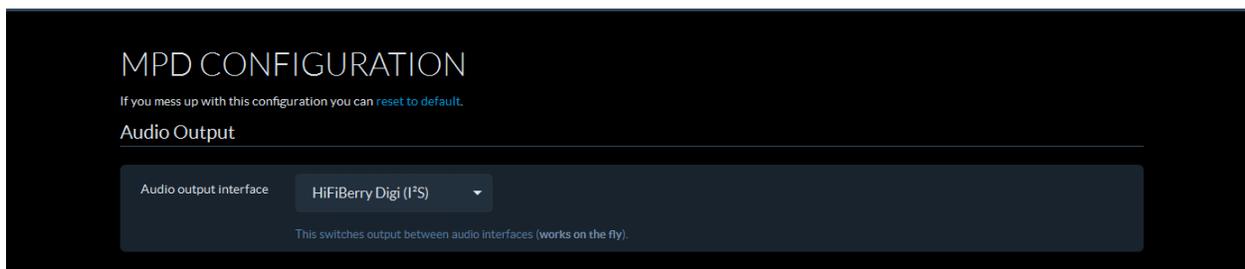
Press Ctrl-X to save and exit.

On another machine or phone with bonjour installed, type <http://runeaudio.local> into a browser.

Click "Menu" in the upper right corner and then select "Turn off" and then "Reboot".

After reboot, click "Menu" in the upper-right corner and then select "MPD"

You will see a screen similar to the one below.



Select "HiFiBerry DAC (I2S)". **(Correction - Not Hifiberry Digi as shown in the picture)**

Enjoy!

Setup For PiCorePlayer

Setup for PiCorePlayer requires a keyboard and monitor to get the proper IP address.

Download and flash SD card with PiCorePlayer.

Connect keyboard and monitor.

Insert SD Card into pi and boot.

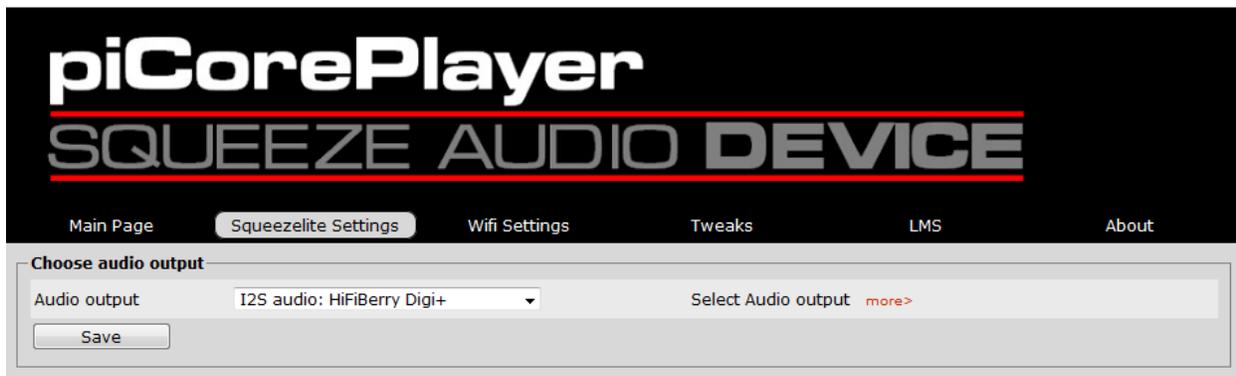
When the box is done booting, hit Enter for a prompt.

Note the IP address from the `ifconfig` command.

On another machine or phone, type the IP address of the pi into a browser.

Click "Squeezelite Settings".

You will see a screen similar to below:



Select Audio output "I2S audio: HifiBerry Dac", then click save.

You will be prompted for reboot. Click yes.

Enjoy!

Note: PiCorePlayer is usually accessed by a host machine running some sort of LMS (LogitechMediaServer)

Setup For PiMusicBox

(currently PiMusicBox does not run on Pi3)

Setup for PiMusicBox requires either keyboard and monitor or ssh.

Download and flash SD card with PiCorePlayer.

Insert SD Card into pi and boot.

Via ssh (root@musicbox.local, pass=musicbox) or attached keyboard and monitor, edit /boot/config.txt.

Note: ssh is disabled by default. If you don't have a monitor handy, enable via browser musicbox.local settings->network->ssh enable, click "update settings (reboot)"

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

comment the line "# devicetree="

uncomment the line "dtoverlay=hifiberry-dac"

Press Ctrl+X to save.

```
sudo reboot
```

On another machine or phone with bonjour installed, type <http://musicbox.local> into a browser.

From the left Pane click "Settings", then expand "Audio". You will see a screen similar to that below:



Select Audio output "HifiBerry Dac."

Scroll to the bottom of the page and click "update settings (reboot)"

Enjoy!

Power Knob Configuration for the CSB503HTA

For Volumio:

Please see instructions at <https://github.com/Pi2Design/PowerCtl>

to install python's RPi.GPIO on volumio: (Pi2 only)

```
sudo apt-get install python-dev python-pip
```

```
sudo bash
```

```
pip install RPi.GPIO
```

For RuneAudio:

It is likely that you will need to expand the filesystem to fit the ASDCard if you have not done so already.

Here are some instructions to do so, from

<http://www.runeaudio.com/documentation/troubleshooting/extend-partition-sd/>

```
fdisk /dev/mmcblk0
```

```
p (show partitions)
```

```
note start block of Linux partition
```

```
d (delete main linux partition)
```

```
enter number for main Linux partition: if mmcblk0p2, enter 2
```

```
n (create new partition)
```

```
p for primary partition
```

```
enter the start block from earlier (should be default),
```

```
use default end block (end of sdcard)
```

```
t (partition type)
```

```
enter partiton number
```

```
enter 83 (for 'Linux')
```

```
w (write changes)
```

an error reloading may appear.

reboot

now, you need to resize the filesystem to fill the partition

resize2fs /dev/mmcblk0p2 (the 'Linux' partition number from before)

After the partition is resized, run the following:

```
pacman -Syy
```

```
pacman -S python
```

```
pacman -S python-pip
```

```
pacman -S base-devel
```

```
select gcc (mine was number 10)
```

Then you can install RPi.GPIO like so:

```
pip install RPi.GPIO
```

Now we get our python script from git:

```
git clone --depth=1 https://github.com/Pi2Design/PowerCtl.git
```

move service file to /usr/lib/systemd/system/PowerCtl.service

then test with:

```
systemctl start PowerCtl.service
```

enable with:

```
systemctl enable PowerCtl.service
```

NOTE: NEED TO CREATE BRACH JUST FOR RUNEAUDIO! Also there is an image file on the fireball machine.

For PiMusicBox:

Install git:

```
sudo apt-get update
```

```
sudo apt-get install git-core
```

Clone python script:

```
git clone git://github.com/Pi2Design/PowerCtl.git
```

```
sync
```

Test python script:

```
cd PowerCtl
```

```
sudo python PowerCtl
```

Turn off power switch, power should stay on, powerdown will be initiated, power will shut off.

Turn power back on

Edit /etc/rc.local:

```
sudo nano /etc/rc.local
```

add the line '/root/PowerCtl/PowerCtl.py &' before musicbox startup.

Enjoy!

For PiCorePlayer:

Resizing SD card Partition/filesystem.

navigate to piCorePlayer.local or IP address of system

click the beta tab at the bottom.

under main, click resize FS

click Resize to auto resize partition

Install required packages:

```
tce-load -wi python-dev compile-essentials python-RPi.GPIO git
```

```
git clone https://github.com/Pi2Design/PowerCtl.git
```

Make files persistent:

```
sudo filetool -b
```

test script:

```
sudo PowerCtl/PowerCtl.py
```

Turn off power switch, power should stay on while powerdown is initiated, power will shut off when powerdown is complete.

```
sudo vi /opt/bootlocal.sh
```

add the following line above picoreplayer startup:

```
/home/tc/PowerCtl/PowerCtl.py &
```

```
sudo filetool -b
```

```
sudo reboot
```

Enjoy!