

Combo RaspberryPi+Wolfson+PureData

Procédure détaillée

1. Install on sd-card raspbian distribution

(<http://downloads.raspberrypi.org/raspbian/images/raspbian-2014-01-09/>)
using <http://sourceforge.net/projects/win32diskimager/files/latest/download>

```
$ sudo rpi-update
$ sudo raspi-config (overclock = medium, expand sd-card, enable camera)
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get dist-upgrade
```

2. Recompile the kernel with the real-time/wolfson driver (<http://www.talkunafraid.co.uk/2014/04/real-time-kernels-and-audio-on-the-raspberry-pi/>)

First we need to add dependencies, patchsets and kernel sources

```
$ sudo apt-get install bc libncurses5-dev libusb-dev
```

There's two patchsets we're going to apply – the Wolfson kernel patches, and the -rt patches, so let's grab those first

```
$ wget http://downloads.element14.com/wolfson/wolfson_drivers.tar.gz
$ tar xzf wolfson_drivers.tar.gz
$ wget https://www.kernel.org/pub/linux/kernel/projects/rt/3.10/older/patch-3.10.25-rt23.patch.gz
$ gunzip patch*
```

Then we check out the sources

```
$ mkdir kernel_sources
$ cd kernel_sources
```

This will take a while and download plenty of data

```
$ git init
$ git fetch git://github.com/raspberrypi/linux.git rpi-3.10.y:refs/remotes/origin/rpi-3.10.y
$ git checkout rpi-3.10.y
```

Go to the specific commit the patches are for

```
$ git reset --hard c43739885d512c92c0aa443b5895b96df5141da0
```

Tell git who we are for our changes

```
$ git config --global user.name "your.user.name" && git config --global user.email "your.email"
```

If you're prompted by the following to skip any empty patches, do so as directed

```
$ git am -3 /home/pi/rpi_wlf_3.10_beta/*
$ cat /home/pi/patch-3.10.25-rt23.patch | patch -p1
```

Copy in the default config

```
$ cp arch/arm/configs/rpi_wolfson_sound_pi_defconfig .config
```

3. Configuring and building your kernel

Your Pi should have a think and then give you a menu. Navigating in here is a bit fiddly so have a play around to get your bearings – if you mess up, just hit escape until prompted to save, don't do so, and run make menuconfig again. Worst case if you save the config by accident, run the cp line above again to copy the default config back in for a clean slate.

```
#start the configuration menu
$ make menuconfig
==
Kernel config -> CPU power management -> CPU frequency scaling -> Default governor -> performance
Kernel config -> CPU power management -> CPU frequency scaling : Disable the non-performance
governor modules, we don't want them!
Kernel config -> Kernel Features > Preemption Model -> Fully preemptible kernel (RT)
==
```

What we've done there is forced the CPU to run at maximum power at all times (note that this is not something you should do in combination with a 1000MHz overclock – 950MHz seems to be stable, but lower frequencies are better if you can live with them), and enabled a fully preemptible kernel. Now we can build it and install the modules and the kernel

```
$ make
$ make modules
$ sudo make modules_install
$ sudo cp arch/arm/boot/Image /boot/kernel-rt.img
```

We copy the kernel to a new location in the boot partition rather than overwriting the existing one – this makes it easy to switch back, and OS upgrades to the kernel package won't wipe your changes out.

4. Configuring the bootloader

We are now telling the pi to use the kernel we just compiled.

```
$ sudo nano /boot/config.txt
==
disable_splash=1
force_turbo=1
kernel=kernel-rt.img
==
```

And we also need to modify /boot/cmdline.txt to work around a bug which would mean the kernel couldn't mount the SD card root partition

```
$ sudo nano /boot/cmdline.txt
==
sdhci_bcm2708.enable_llm=0
==
```

5. install the alsa extras script

```
$ wget http://www.workinprogress.ca/wp-content/uploads/wolfson\_alsa.zip
$ unzip wolfson_alsa.zip
```

6. Check if everything is in place :

```
$ uname -a
Linux raspberrypi 3.10.25-rt23+ #1 PREEMPT RT Wed Oct 15 15:34:20 UTC 2014 armv6l GNU/Linux
```

```
$ cat /proc/asound/cards
0 [sndrpiwsp    ]: snd_rpi_wsp - snd_rpi_wsp
                        snd_rpi_wsp
1 [ALSA        ]: BRCM bcm2835 ALSbcm2835 ALSA - bcm2835 ALSA
                        bcm2835 ALSA
```

```
$ cat .asoundrc
pcm.wolfson_pi_soundcard {
    type hw
    card sndrpiwsp
    device 0
}

pcm.softvol {
    type      softvol
    slave {
        pcm    "wolfson_pi_soundcard"
    }
    control {
        name    "Master"
        card    0
    }
}

pcm.!default softvol
```

7. \$ sudo su -c 'echo @audio - rtprio 99 >> /etc/security/limits.conf'
\$ sudo su -c 'echo @audio - memlock unlimited >> /etc/security/limits.conf'
\$ sudo su -c 'echo @audio - nice -10 >> /etc/security/limits.conf'
\$ sudo reboot

8. Disable the other soundcard :

#Put a '#' before the sound card

```
$ sudo sed -i 's/snd\bcm2835/#snd\bcm2835/g' /etc/modules
```

9. Install puredata from miller's site (<http://msp.ucsd.edu/software.html>)

```
$ wget http://msp.ucsd.edu/Software/pd-0.46-0.rpi.tar.gz
$ tar xzpf pd-0.46-0.rpi.tar.gz
```

Create a PureData patch and then try to open it with that command (connected via ssh)

```
$ cd /home/pi/pd-0.46-0/bin
$ sudo ./pd -rt -alsa -audiodev 1 -r 48000 -audiobuf 8 -channels 2 -nogui -stderr -open <your.patch>
```

Be careful about the DSP activation: You need to delayed it when opening the patch as I did on the fig.1

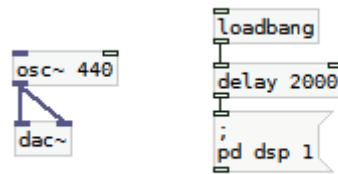


Figure 1 : My test patch osc.pd

You should hear a 440Hz sin wave after 2000ms from the headset or Line Out

10. Install VNC server (http://elinux.org/RPi_VNC_Server)

```
$ sudo apt-get install tightvncserver && vncserver :0 -geometry 1920x1080 -depth 24
```

```
$ sudo nano .vnc/svnc.sh
```

```
==
```

```
#!/bin/sh
```

```
vncserver :0 -geometry 1920x1080 -depth 24 -dpi 96
```

```
==
```

```
Save and exit
```

```
$ chmod +x svnc.sh
```

To run the vnc server just type (in /home/pi)

```
$ cd .vnc
```

```
$ ./svnc.sh
```