**DECODING AFSK**

**Now , we have to decode the audio**

We can save the audio file .. or we can make a clever use of pulseaudio  : just set the default capture device to the monitor of your output device from the fm reciever , i.e. : minimodem is capturing the sound that is played trough a specific sound card

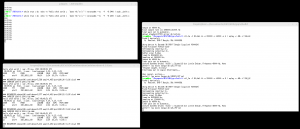
Start **minimodem** with the alsa [1] flag ‘-A’ and the same setting used on the transmitter

**receiver$** minimodem -A -8 1200

Start **pavucontrol** to connect the audio of minimodem and watch the AFSK being decoded by minimodem.

You can also use **rtl\_fm** (no gui for a lower cpu overhead) to decode the fm sample to the default alsa device.

**receiver$** rtl\_fm -f 80.0e6 -W -s 200000 -r 48000 -o 6 | aplay -r 48k -f S16\_LE

[](http://www.kprod.eu/blog/wp-content/uploads/2013/11/2013-11-05-161939_3840x1080_scrot-e1383674339417.png)

You can also do the same [RTTY](http://ukhas.org.uk/guides:linkingarduinotontx2) transmission with [AFSK and arduino](http://emmanuelgranatello.blogspot.it/2012/08/remote-temperature-sensor.html) and use the sound channel of your [fpv rig](https://www.youtube.com/watch?v=2XvINTDkPi0).

No gnuradio block was involved in this whole process.

1 : you can use the ‘default’ alsa hook with pulseaudio, no need to set a specific hdw , see

<https://wiki.debian.org/en/PulseAudio#Installing_PulseAudio>.

You may want to increase the default value when using pulesaudio and gnuradio , ex :

**#cat ~/.gnuradio/config.conf**

[audio\_alsa]

period\_time = 0.020 # in seconds

nperiods = 30 # total buffering = period\_time \* nperiods

verbose = true