

# SOX audio tool Quick Reference Guide

## generate file

Generate 1 second of white noise.

```
sox -n output.wav synth 1 noise
```

Generate a 1-second sine tone.

```
sox -n output.wav synth 1 sine 440
```

Generate a 10-second sine sweep file.

```
sox -n output.wav synth 10 sine 0:20000
```

Exponential sine sweep

```
sox -n -r 44100 sine-sweep.wav synth 10 sine 5/22050
```

## play

Play synthesized audio.

```
play -n synth sine 440 trim 0 1 gain -12
```

Play sine verbose 24 bits /=exponential

```
play -V -n -b 24 -r 48000 synth 10 sine 20/20000
```

```
play -V -n -b 24 -r 48000 synth 10 sine 20/20000 vol -50dB
```

And then ARTA:

```
Fs (Hz): 192000
```

```
FFT: 4096
```

```
Wnd: FlatTop
```

Avg: Peak Hold

# calibration tones

Play 30 tones at once, for frequency adjustment of equipment etc.

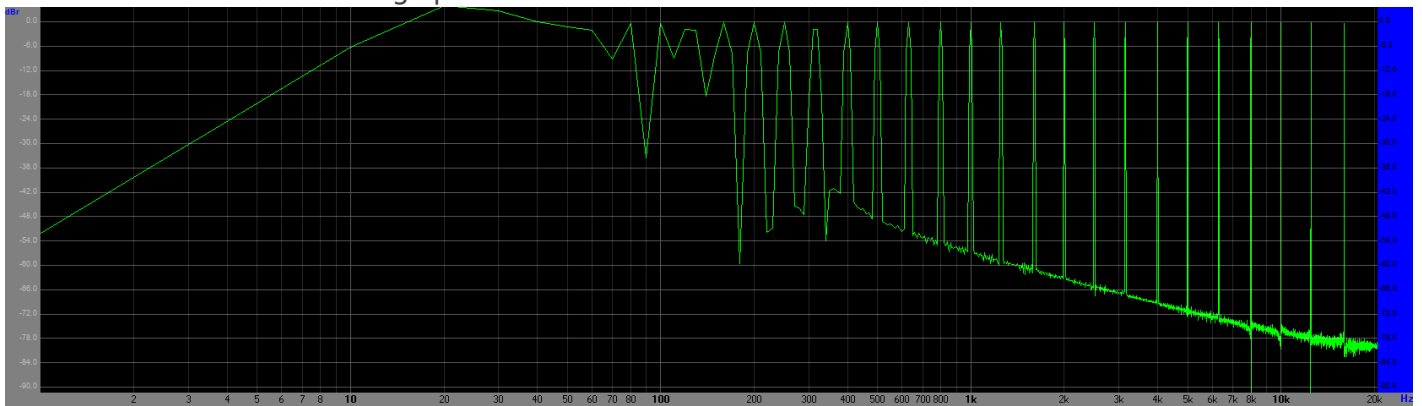
```
play -r 192000 -V -  
n synth 300 sin 20 sin 25 sin 31.5 sin 40 sin 50 sin 63 sin 80 sin 100 sin 125 sin 160 sin 200 sin  
250 sin 315 sin 400 sin 500 sin 630 sin 800 sin 1000 sin 1250 sin 1600 sin 2000 sin 2500 sin 3150 s  
in 4000 sin 5000 sin 6300 sin 8000 sin 10000 sin 12500 sin 16000 sin 20000 remix 1-31 1-31
```

where:

192000 is the samplerate, if your card supports it

300 is number of seconds

- This results in the following spectrum:



If you want to generate the file yourself, use:

```
sox -r 192000 -V -  
n "31 frequencies.flac" synth 300 sin 20 sin 25 sin 31.5 sin 40 sin 50 sin 63 sin 80 sin 100 sin 12  
5 sin 160 sin 200 sin 250 sin 315 sin 400 sin 500 sin 630 sin 800 sin 1000 sin 1250 sin 1600 sin 20  
00 sin 2500 sin 3150 sin 4000 sin 5000 sin 6300 sin 8000 sine 10000 sin 12500 sin 16000 sin 20000 r  
emix 1-31 1-31
```

It will create a 300-seconds FLAC file with the 31 tones called "31 frequencies.flac"

## modify

Reduce level by 12dB

```
sox speech.wav output.wav gain -12
```

Crop to the first 1 second of the file.

```
sox speech.wav output.wav trim 0 1
```

Reverse the contents.

```
sox speech.wav output.wav reverse
```

Normalise the contents to 0dBFS.

```
sox speech.wav output.wav norm
```

Equaliser (-6dB @ 100Hz, -24dB @ 8000Hz)

```
sox speech.wav output.wav bass -6 100 treble -24 8000
```

Add room modelling reverb.

```
sox speech.wav output.wav reverb 50 50 100
```

Trim digital silence from start and end.

```
sox input.wav trimmed/output.wav silence 1 0.1 0 1 0.1 0
```

-

```
sox infile outfile gain -n
```

normalises to 0dB, and

```
sox infile outfile gain -n -3
```

normalises to -3dB.

## effects

chorus effect: A typical delay is around 40ms to 60ms; the modulation speed is best near 0.25Hz and the modulation depth around 2ms. For example, a single delay:

```
play guitar1.wav chorus 0.7 0.9 55 0.4 0.25 2 -t
```

Two delays of the original samples:

```
play guitar1.wav chorus 0.6 0.9 50 0.4 0.25 2 -t 60 0.32 0.4 1.3 -s
```

A fuller sounding chorus (with three additional delays):

```
play guitar1.wav chorus 0.5 0.9 50 0.4 0.25 2 -t 60 0.32 0.4 2.3 -t 40 0.3 0.3 1.3 -s
```

# compand

The following example might be used to make a piece of music with both quiet and loud passages suitable for listening to in a noisy environment such as a moving vehicle:

```
sox asz.wav asz-car.wav compand 0.3,1 6:-70,-60,-20 -5 -90 0.2
```

The transfer function (``6:-70,...'`) says that very soft sounds (below -70dB) will remain unchanged. This will stop the compander from boosting the volume on 'silent' passages such as between movements.

However, sounds in the range -60dB to 0dB (maximum volume) will be boosted so that the 60dB dynamic range of the original music will be compressed 3-to-1 into a 20dB range, which is wide enough to enjoy the music but narrow enough to get around the road noise.

The ``6:'` selects 6dB soft-knee companding. The -5 (dB) output gain is needed to avoid clipping (the number is inexact, and was derived by experimentation).

The -90 (dB) for the initial volume will work fine for a clip that starts with near silence, and the delay of 0.2 (seconds) has the effect of causing the compander to react a bit more quickly to sudden volume changes.

# general

`--multi-threaded` | `--single-threaded`

By default, SoX is 'single threaded'. If the `--multi-threaded` option is given however then SoX will process audio channels for most multi-channel effects in parallel on hyper-threading/multi-core architectures. This may reduce processing time, though sometimes it may be necessary to use this option in conjunction with a larger buffer size than is the default to gain any benefit from multi-threaded processing (e.g. 131072; see `--buffer` above).

---

Revision #3

Created 2026-04-01 17:15:15 CEST by Philip

Updated 2026-04-13 19:23:43 CEST by Philip