Tested by:Ástvaldur Hjartarson
Date: 2014-05-19

Test protocol

Audio distortion measurements

Description

The reason for doing distortion testing is to have way to measure the performance of the audio quality. The way you do this has to be consistent if you are going to get the correct results on how much improvement in audio fidelity an upgrade will make.

Note:
The way we did our final distortion testing was not good enough, hence in the future a higher quality test is needed. The measurement tools available were not sufficient for quality measurements.

Equipment:

- Nexys 4 board with implemented design and power supplies
- A phone or a similar device with a headphone jack.(HPJ)
- Computer or a similar device with a built in microphone(MIC).
- Matlab installed on a computer with the Signal Processing Toolbox.

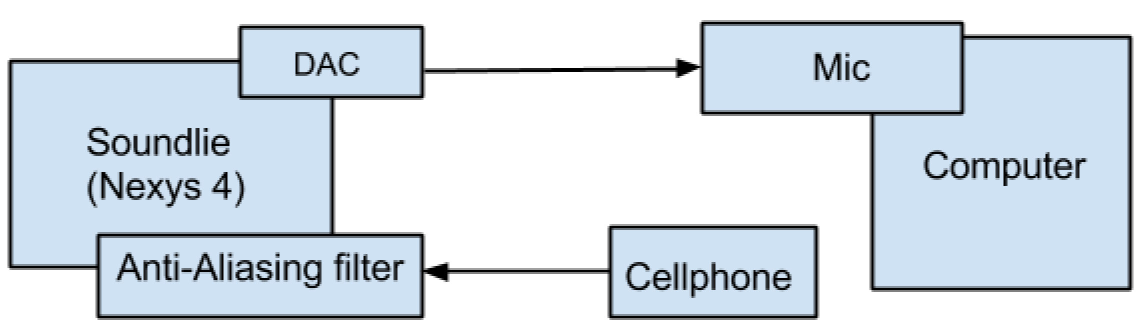
.

Preparation

To prepare the system you will have to record with the computer the sine wave played by the cell phone. Then to the snr measurements described in the matlab test part.

Setup:

The cellphone has to be loaded with a sound file that plays a 1 kHz sine wave. The cellphone HPJ is then connected to the input of the soundlie device. To the output of the Soundlie the microphone of the computer or the recorder has to be connected.



Testing

To test the device you will have to play the sound file on the input and the record the output on the computer. To test the device you will have to run four instances for the tone control:

- without tone control.
- with tone control on all bands set to 0dB.
- with tone control on all bands set to -12dB
- with tone control on all bands set to +12dB

Matlab analysis

If you are using a computer to record, you can record the sound with this code:

```
%Recording.m
recObj = audiorecorder(48000, 16, 2);
disp('Start speaking.')
recordblocking(recObj, 10);
disp('End of Recording.');
y = getaudiodata(recObj,'int16'); % get data as int16 array
audiowrite('output2000.wav',y,48000)
```

To analyse each of these recording the following code should be used to get the desired results:

```
[y,FS]=audioread('recording');

L = length(y);              % Length of signal
t = (0:L-1)*(1/FS);         % Time vector
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
f=FS/NFFT.*(0:(NFFT/2)-1);

sinesnr= snr(y(:,1),FS,6);
disp('snr for 1kHz');
disp(sinesnr)
sinethd= thd(y(:,1),FS,6);
sinethd1=100*(10^(sinethd/20));
disp('thd for 1kHz');
disp(sinethd1)

figure(1)
plot(f,20*log10(2*abs(Y(1:NFFT/2))))
title('Single-Sided decibel magnitude of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|(db)')
```

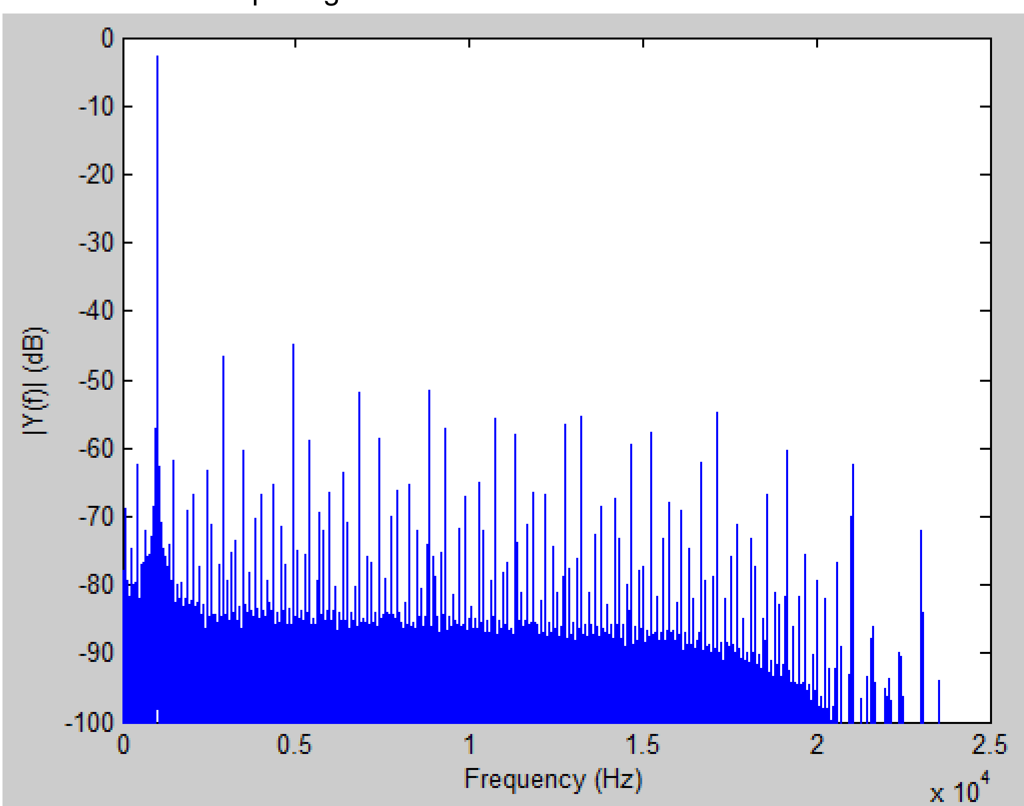
Results from test

The results from the testing done done 19 may 2014. First we will show the snr and thd and then show the fft graphs.

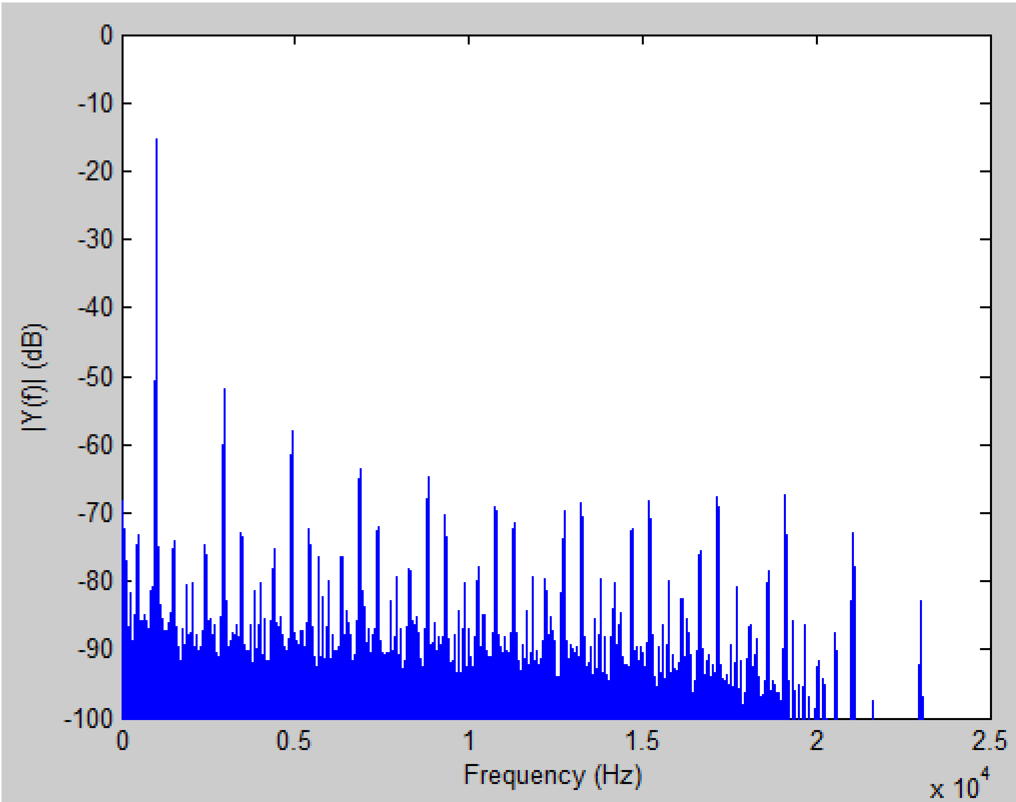
The results from the Matlab analysis of snr and thd.

Case	SBR(dB)	THD(%)
No tone control	37.5	0.02
0 dB	31.5	0.08
-12dB	14.0	2.6
+12 dB	31.5	0.05

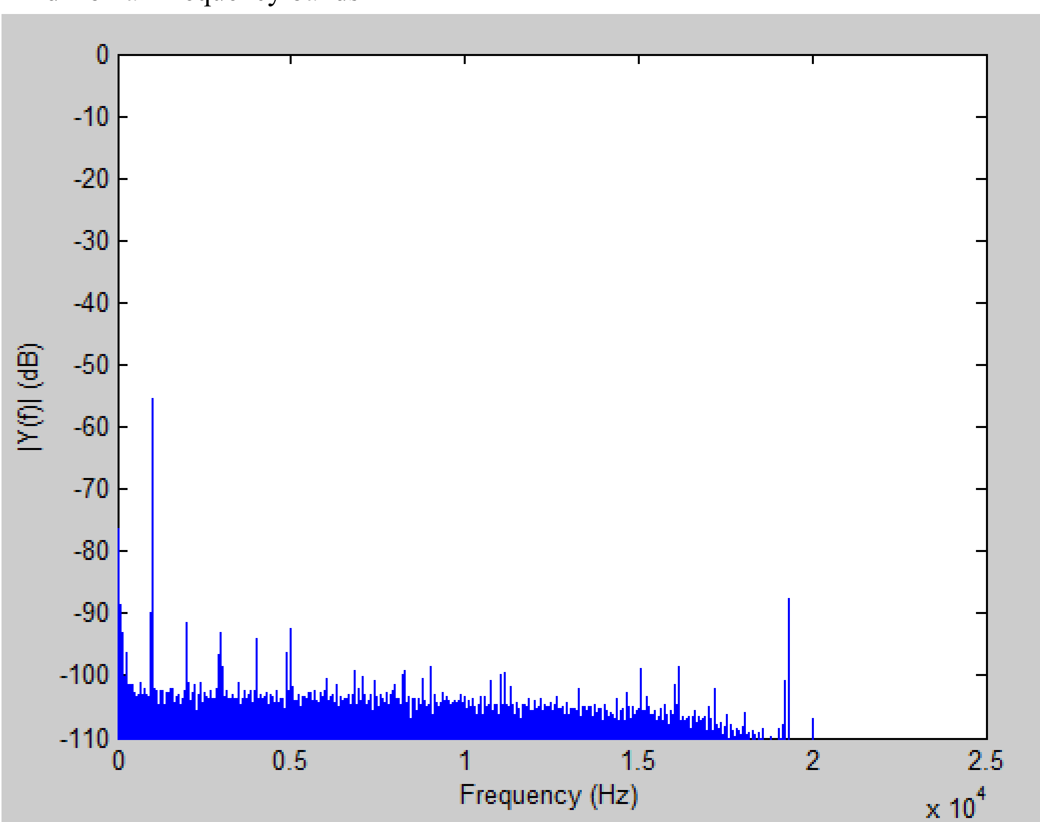
No tone control output signal.



0 dB on all frequency bands,



-12 dB on all frequency bands



Tested by:Ástvaldur Hjartarson
Date: 2014-05-19

+12 dB on all ranges