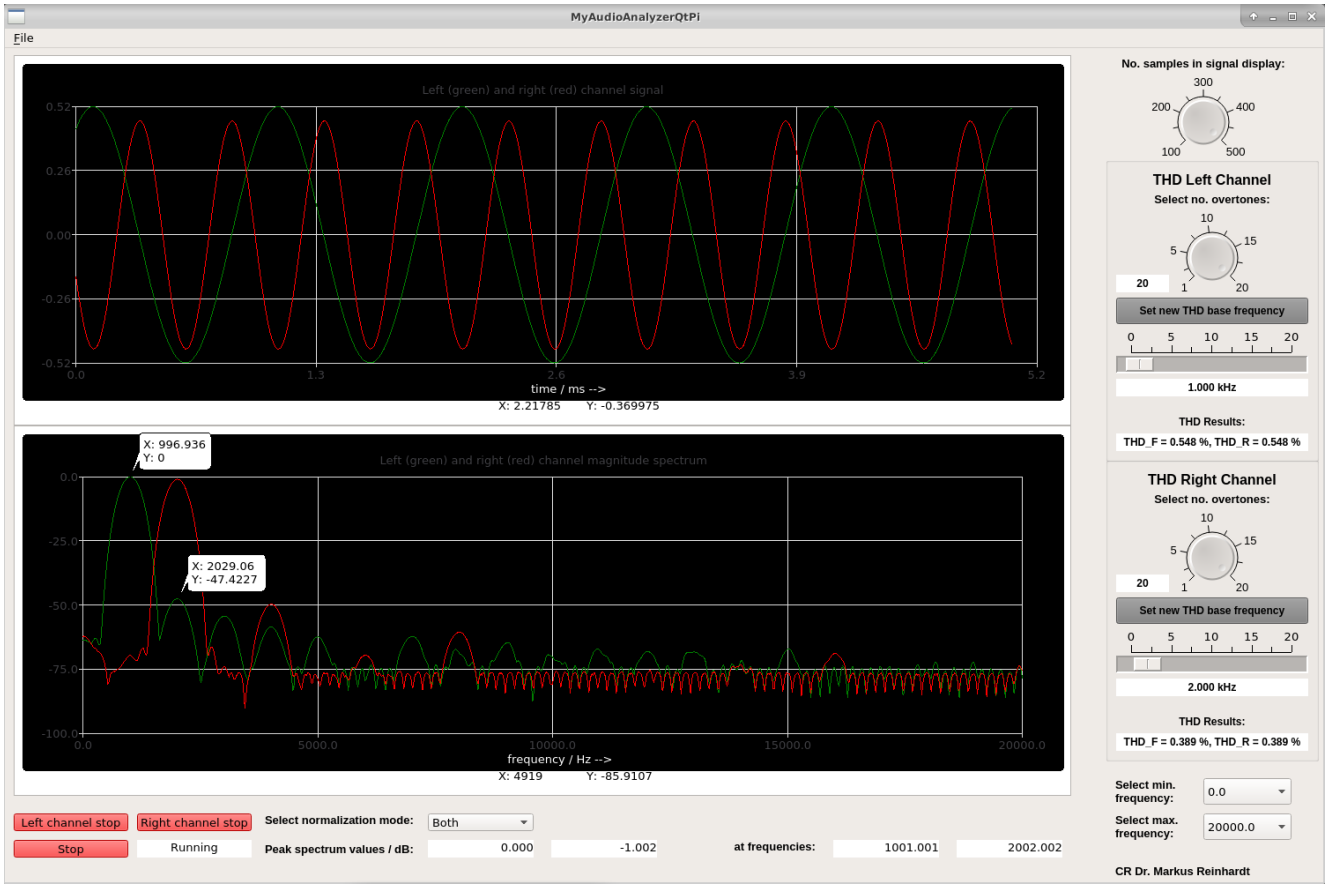
Audio Spectrum Analyzer for the Raspberry Pi with Pisound board realized in C++ with Qt/Qwt libraries

Documentation and User Guide

by Dr. Markus Reinhardt

January 27, 2021



Contents

List of Figures

1 Overall Concept Overview

The block diagram of the Pisound (a Raspberry Pi sound card) based spectrum analyzer is shown in figure 1:

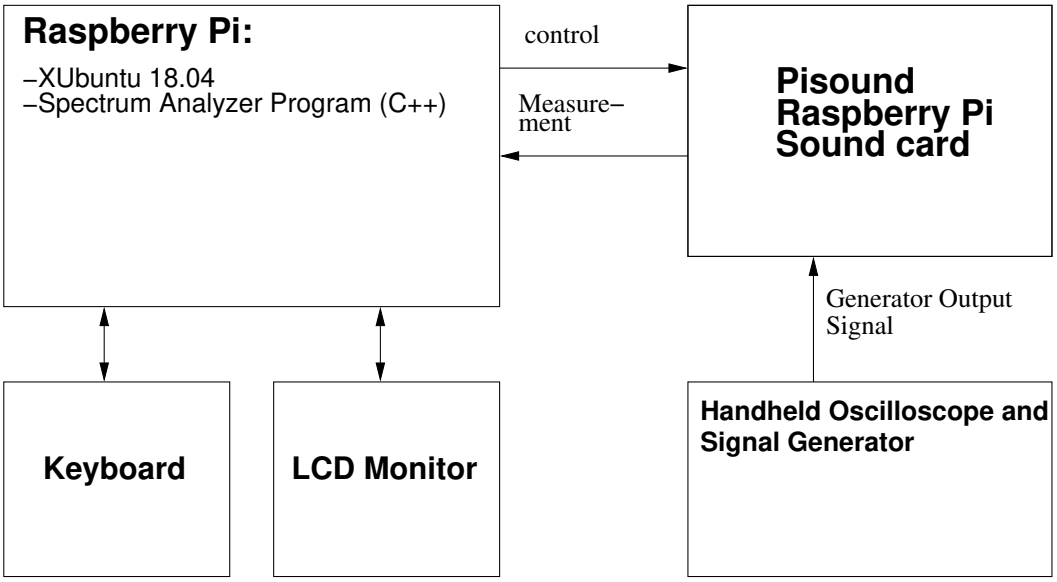


Figure 1: Spectrum analyzer top level block diagram

2 Hardware Description

2.1 Pisound Board

Figure 2 shows the top view of the Pisound board mounted on the Raspberry Pi.

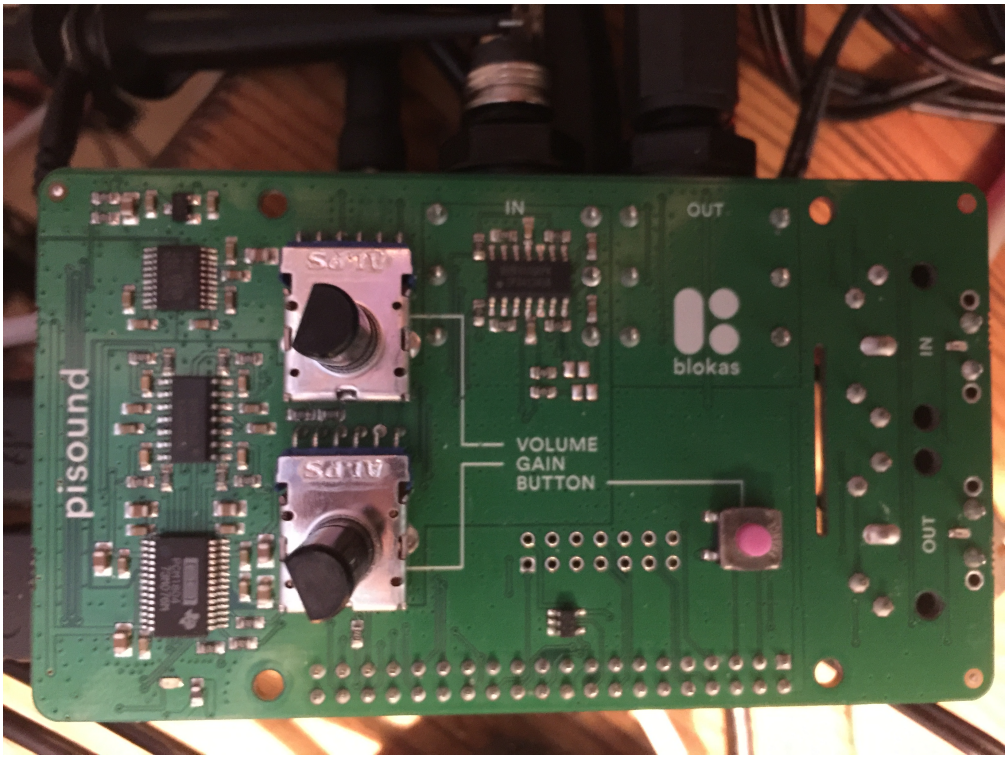


Figure 2: Pisound board (top view)

The Pisound board has a PCM1804 full differential analog input 24-BIT, 192-kHz stereo A/D converter and a PCM5102 112dB Audio Stereo DAC with 32-bit, 384kHz PCM interface. The ADC input is fed from a OPA2134 OpAmp. The output of the DAC is also buffered by a OPA2134 OpAmp. The stereo input and output signals are fed to the board via ” (6.35mm) sockets.
The input amplitude can be controlled by a potentiometer as well as the output volume. Both potentiometers are shown on figure 2.

2.2 Casing

The audio spectrum analyzer is housed in the dedicated Pisound case.

3 Software Description

3.1 Analyzer GUI

Figure 3 shows the GUI for the audio spectrum analyzer that is realized as a C++ program using the Qt/Qwt libraries. In this case both channels are activated.

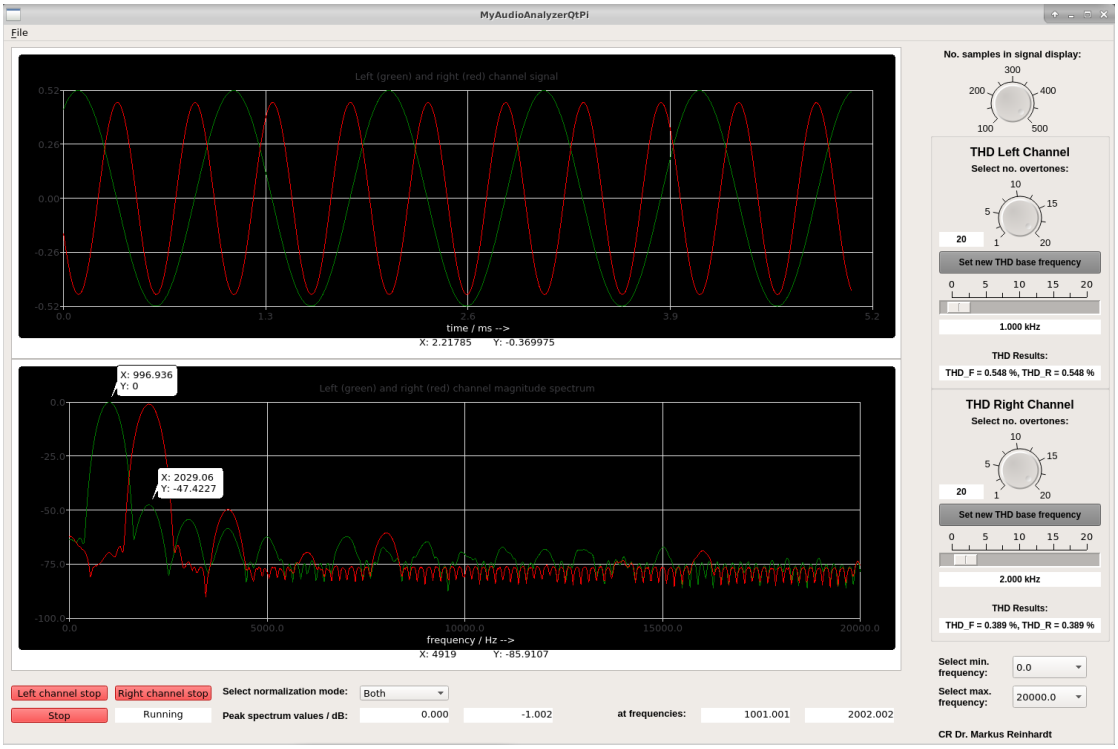


Figure 3: Spectrum analyzer GUI (both channels)

The software is realized with three threads based on the QThread class. The first thread realizes the acquisition of the input signals from the left and right channel via the Portaudio library. The second thread realizes the calculation of the spectra for both channels via a DTFT routine. The third thread realizes the handling of the GUI with several widgets and the display of the signals, spectra and calculated signal and spectrum characteristics.
The upper left part of the GUI shows the sampled signals for the left and right channel. A dial right to the signal display allows to change the no. of samples shown in the signal display.
The lower left part shows the corresponding calculated spectra of the left and right channel. In the lower right corner of the GUI there are two combo boxes that allow to change the upper and lower frequency range limits of the spectrum display.
In the middle of the right part of the GUI the control and display for the calculation of the THD of the left and right channel are shown. For each channel the number of over-

tones used in THD calculations are controlled by a knob and a related display. The base frequency of THD calculations can be controlled with a slider and the related display of the base frequency. The continuously calculated THD_F and THD_R values are shown below the controls.
The lower part of the GUI shows the control buttons and measurements.

The program is started with the command MyAudioAnalyzerQtPi.

The start/stop-button labeled either "Start" or "Stop" (depending on the processing status) at the left bottom corner of the GUI is used to start or stop the signal capturing, the spectrum calculations and the display of signals and spectra. When the analyzer is stopped the button's background color is green and the label is "Start". When the analyzer is running the button's background color is red and the label is "Stop".

The label next to the start/stop-button displayes the status, either "Running" or "Stopped". Hitting the button "Stop" will stop the spectrum calculation process.

With the buttons "Left channel start/stop" or "Right channel start/stop" the processing and display of the relevant channel(s) is activated or deactivated.

With the selector right to the text "Select normalization mode" the mode how to do the normalization of the calculated signal spectra can be chosen. There are four modes:

- "Both": Both spectra are normalized with the maximum magnitude value determined of both channels.
- "Left": Both spectra are normalized with the maximum magnitude value of the left channel.
- "Right": Both spectra are normalized with the maximum magnitude value of the right channel.
- "Separate": Each spectrum is normalized with the maximum magnitude value of the corresponding channel separately.

With "Ctrl-Q" the program is quit.

With "Ctrl-S" the signals and spectra are saved (in file SignalAndSpectra.csv).

Also the peak values of the calculated spectra (in dB) and the corresponding location in frequency domain of the peaks are displayed on the status line for both channels (if they are activated).

Figure 4 shows the GUI for the audio spectrum analyzer in the case that only the left channel is activated.

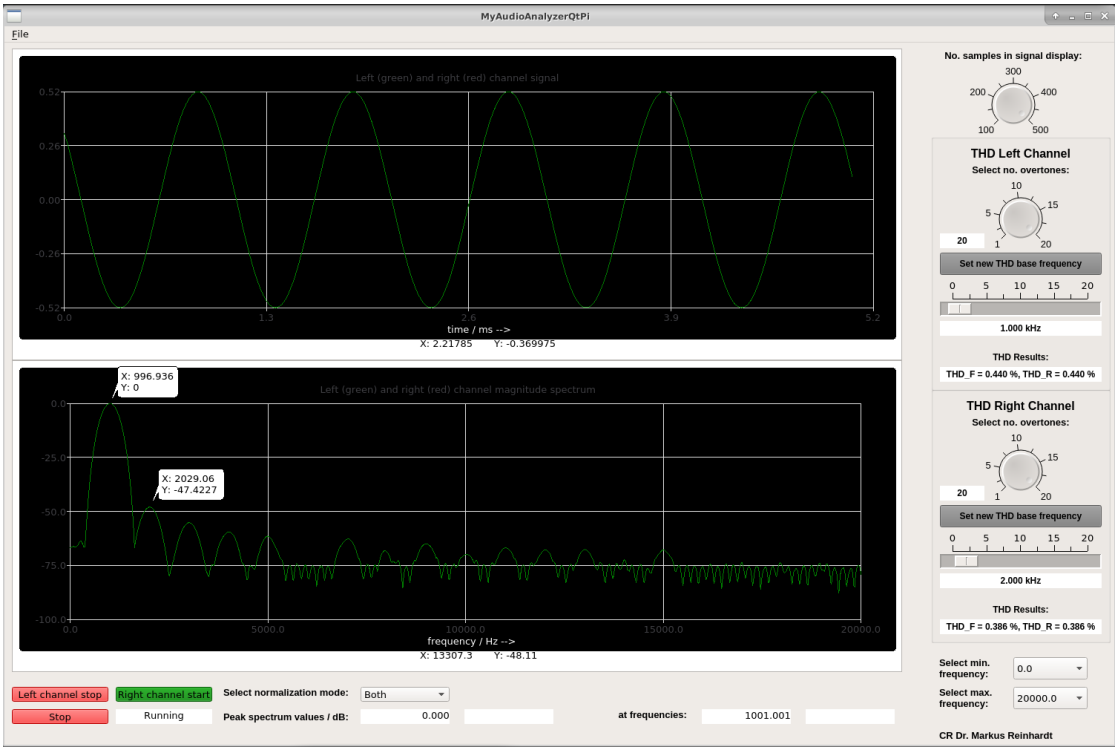


Figure 4: Spectrum analyzer GUI (single channel)

3.2 Postprocessing Tools

A python script (TestLoadSpectrumDataCSV.py) can be used to load and display the stored signals and spectra from file SignalAndSpectra.csv. Figure 5 shows the figures that are created by the program based on the stored spectra generated by the audio spectrum analyzer.

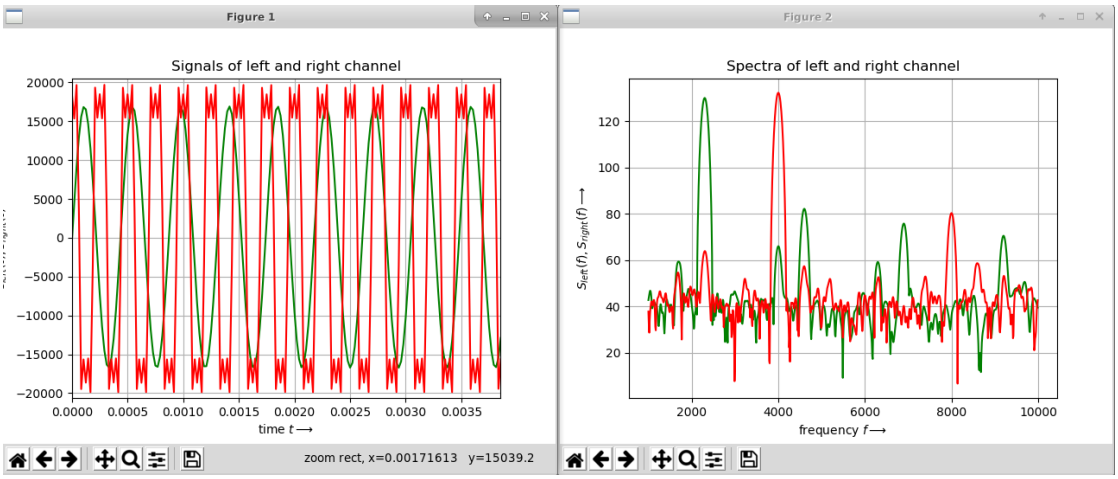


Figure 5: Figures of loaded and postprocessed signals and spectra

ToDo

- Improve the postprocessing tools.
- Add a button to remove all existing callouts.
- Remove all existing callouts at the restart of the processing.