

Guitar Tuner implementation on multi-platform

Yiming Zhao, Anzhi Li, Yang Lu, Zhiyao Duan, University of Rochester, Rochester, NY 14627

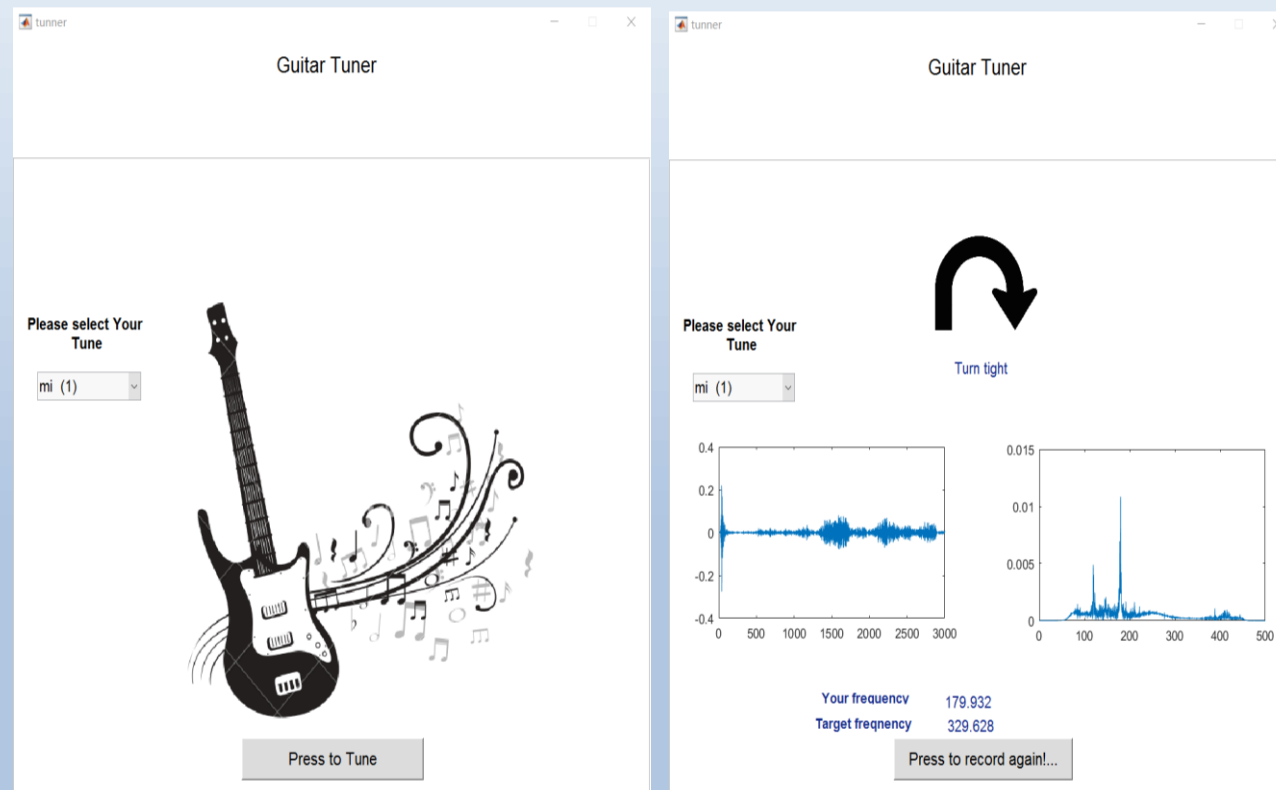
Introduction

This project is mainly designed and conducted in order to realize the function of guitar tuner on three different platforms: MATLAB, Arduino and IOS. Two approaches for frequency detection are described as follows

Frequency domain detection:
Use Fourier transform transfer the input signal to frequency domain, then detect the largest magnitude of one specific frequency range to determine the frequency of input signal.

Time domain frequency detection:
Through counting the characteristic parts of the input signal, find the period of the input in time domain directly. Then calculate the frequency of the input signal according to the period.

Guitar Tuner Using MATLAB

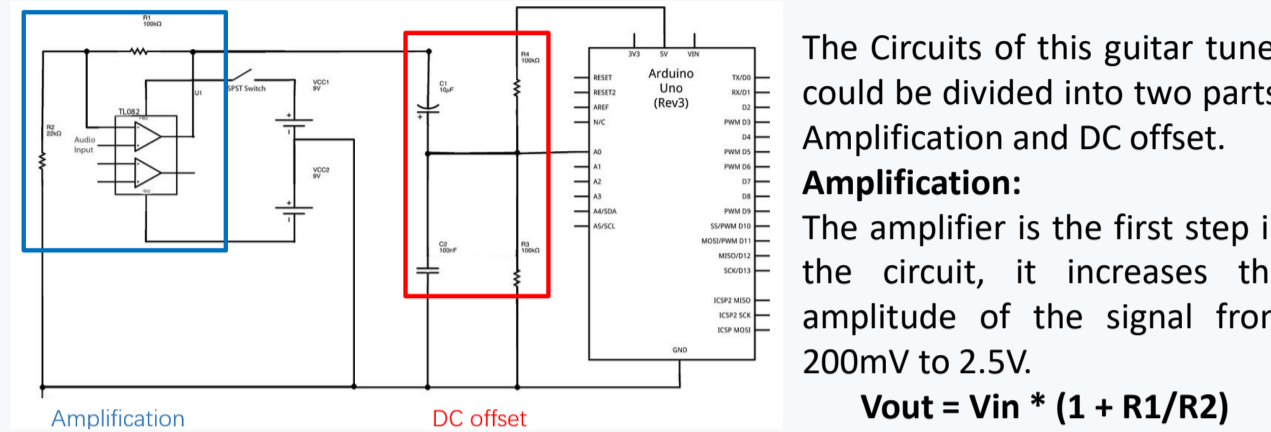


The core algorithms of this MATLAB project is based on FFT(Fast Fourier Transform). By letting user input the sounds of guitar string and record them through computer's speakers, we apply FFT directly to the recording and then search for the fundamental frequency to represent the general tune of the recording. After we obtain user's frequency, we are able to compare the frequency with the standard string's frequency listed previously in the MATLAB code. If user's frequency minus standard frequency is less than 2Hz, we can consider user's guitar is on the right tune. If error between two frequencies is larger or less than zero, we consider user's guitar string as inaccurate and will inform the user to adjust the string accordingly.

User interface

Users will be able to select which string they would like to tune. After users select the tune, they have to press the button and then play the string. The recording process will be lasted for 3 seconds. Users will be able to see the instruction on the top of the interface about the condition of their tunes. The diagrams of Signal in time domain, frequency domain and values of users' and target frequencies are also provided.

Guitar Tuner Using Arduino



The Circuits of this guitar tuner could be divided into two parts, Amplification and DC offset.

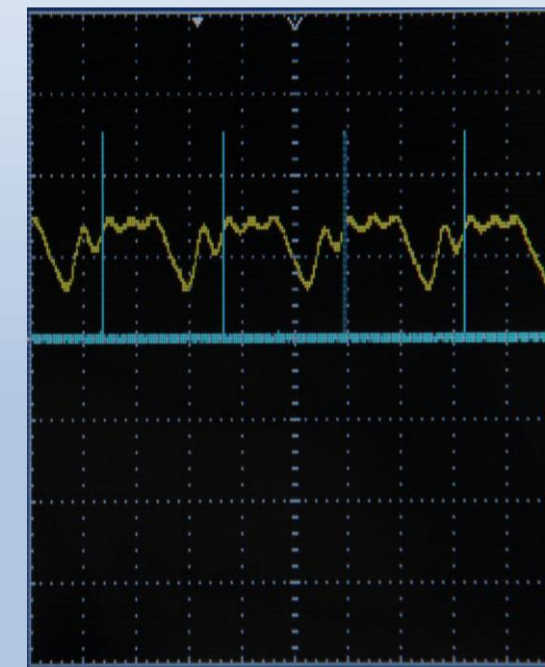
Amplification:
The amplifier is the first step in the circuit, it increases the amplitude of the signal from 200mV to 2.5V.

$$V_{out} = V_{in} * (1 + R1/R2)$$

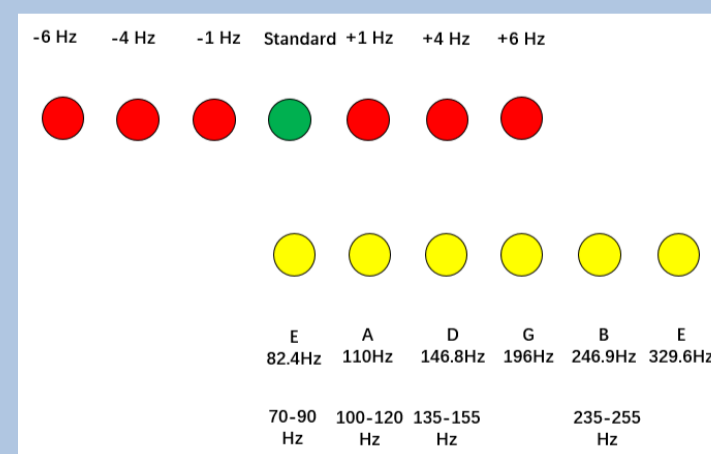
DC offset:
The purpose of the DC offset is to move the center of the input signal from 0 volts to 2.5 volts.

Methodology

1. Positive slope + Crossed 2.5v as a threshold event.
2. Choose largest slope as the beginning of one cycle.
3. Store the time between every threshold event in array1.
4. Store the slope of threshold event in array2.
5. Compared the elements of array1 and array2 to figure out where there was a match.
6. Once the match was found it means one cycle.
7. Count the cycle and calculate the frequency.



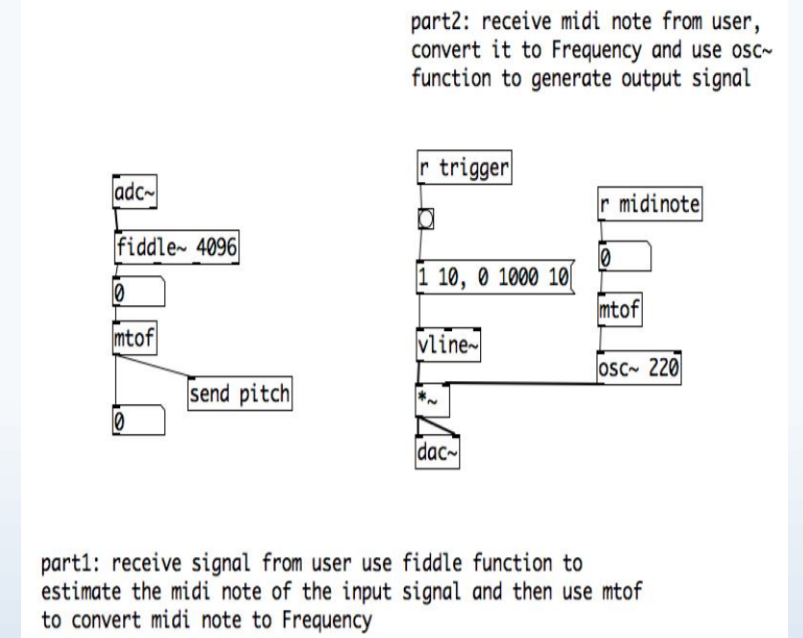
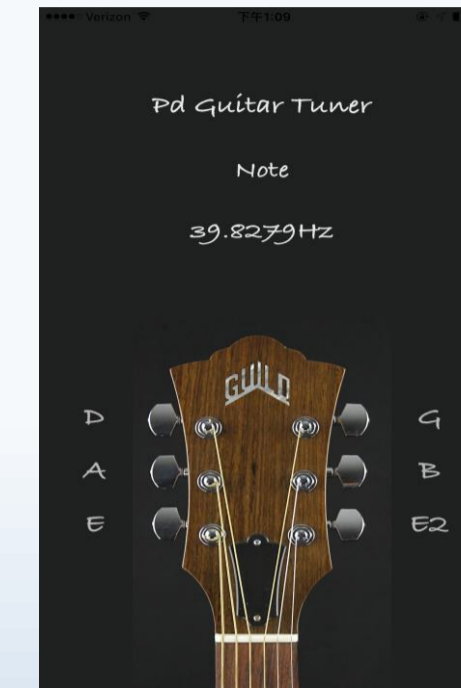
Setting



There are thirteen LED lights are connected to the Arduino board. Six yellow lights represent the six strings of guitar. And six red lights are set on the two side of the green lights, indicating the accuracy of pitch. Once the green light is turned on, it means the pitch of this string is correct.

The Character below the yellow lights are the standard pitch of each strings, and the numbers below these characters are corresponding standard frequency with unit hertz. The range in the bottom is turn on range of each light, which means the input frequency belongs to one of the range, the corresponding light will be turned on to show which string are tuning. The upper values are the error of the frequency.

Guitar Tuner Using Swift



part1: receive signal from user use fiddle function to estimate the midi note of the input signal and then use mtof to convert midi note to Frequency

part2: receive midi note from user, convert it to Frequency and use osc~ function to generate output signal

We combined Pure Data with Swift to develop a App on iOS platform Method:

Use Pure Data patch to generate six reference notes for users to adjust their string. At same time, detect input signal, recognize the frequency of it and show the frequency on the table View.

Conclusion

On MATLAB platform, we did FFT to input signal and located the frequency with maximal value. This frequency should be the fundamental frequency of input signal. On Hardware platform, we chose the other one algorithm, which is finding frequency by detecting how many period in one second. Then, Arduino board can calculate the frequency. We tried to implement one algorithm on these to platform. However, the results are far from satisfactory.

For MATLAB platform, because we used microphone to receive signal, noise cannot be ignored. If we implement period detecting algorithm in MATLAB, noise would affect detecting result on a vast scale so that detecting frequency should not be accurate.

For Hardware platform, we tried to implement FFT method. However, although FFT method is fast, it need a relatively huge power to supply its work. Thus, the requirement for power or power source is hard to satisfy.

All in all, FFT algorithm is better for MATLAB platform and period detecting algorithm is adapted to Hardware platform more.

Reference

- [1]FFT Guitar Tuner, <http://www.codeproject.com/Articles/32172/FFT-Guitar-Tuner>
- [2]Discrete-Time Signal Processing, Alan V. Oppenheim BJORG,
- [3]<http://blog.bjornroche.com/2012/07/frequency-detection-using-fft-aka-pitch.html>