Final Project Proposal by Huyongqing Chen: Theremin with Ultrasonic Distance Sensor

Abstract:

Theremin is an instrument that one alters its volume and pitch through moving one's hand closer or further away from the two antennas, it pitch and volume depends on the capacitance of the metal antenna which can be changed by changing the distance between one's hand to the antenna. The goal of this project is to create a Theremin like instrument using a Ultrasonic distance sensor to measure the distance between one's hand and the instrument to determine the frequency, and using a passive buzzer to generate the sound on the desired frequency depending on the distance. The Theremin is able to change the frequency of the sound wave electronically by changing the capacitance, but in the adaptation of this project, we will convert distance to frequency with Arduino, and modify the pitch through PWM.

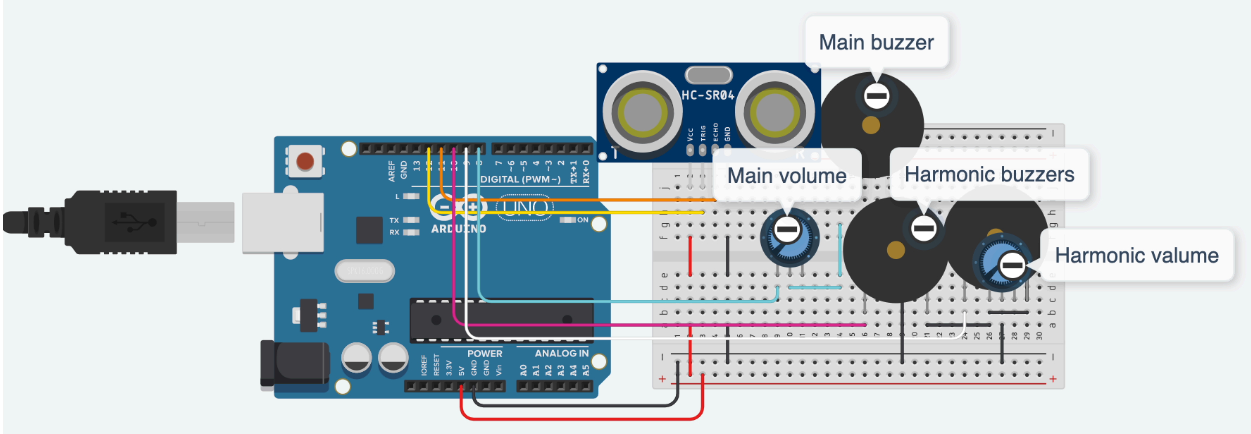
Complications:

Originally, the project planned to use an IR distance sensor to measure the distance. But the given Arduino kit only has an IR receiver paired with an IR remote controller, while this project requires an IR transmitter for measuring the distance. Although one solution to this problem is to require an IR transmitter, but it was not clear how well the IR sensors work when the player is reflecting the light with one's palm. In the end, the ultrasonic distance sensor seems to be the better solution here. One complication this brought is the theoretical delay in response time in using the ultrasonic sensor. Because light travels way faster than sound, the time needed to measure the distance by an IR distance sensor is negligible; yet suppose the player is operating at 1 meter away from the ultrasonic sensor, the time needed to measure the distance by an ultrasonic sensor is ideally 5.88ms. This response time does not affect much on simple operations using the Theremin, but it's certainly more desirable to minimize the response time as much as possible.

Stages of Project:

1. Basic Theremin setup: actively respond to ultrasonic sensor distance with frequency change on passive buzzer.
2. Volume control: with added ultrasonic distance sensor or simply with other control methods. (such as potentiometer or tilt sensors)
3. Explore advanced music technique controls with electronics first, such as damping or harmony. Alternatively use coding to achieve the same.

Plan of Implementation:



In actual implementation the ultrasonic distance sensor will be placed at a more convenient location for measuring distance with comfortable hand position. The code on implementing the basic Theremin setup would primarily evolve converting distance measured to reasonable frequency and deal with cutting off unattainable distance measurements.

Volume control is achieved by adding a potentiometer to adjust the amplitude of the input wave to the passive buzzer. Alternatively, adding a second active IR sensor allows one to use similar performance techniques like an actual Theremin. But since there are many other interesting sensors provided in the kit, it's worth exploring other combinations such as change in tilt vs. volume.

Third Stage Implementations:

1. Cord/String/Harmonies

   Creating harmonies or cords requires generating sounds that are at frequency of $f \cdot 2^{N/12}$, where N represents the number of notes that the main note is in harmonic with[1], where it can be any positive or negative integers. Originally, this project aimed at using a circuit that finds the harmonic frequency waves from the wave of the main note, but I failed to find a frequency changing circuit that has a lot of freedom in getting frequencies of a specific ratio. Full-wave rectifier have a rough frequency doubling effect, but I'm not so sure how that would play out in practice. So, the simpler idea here would be just calculating the harmonic frequencies then output it from the Arduino.

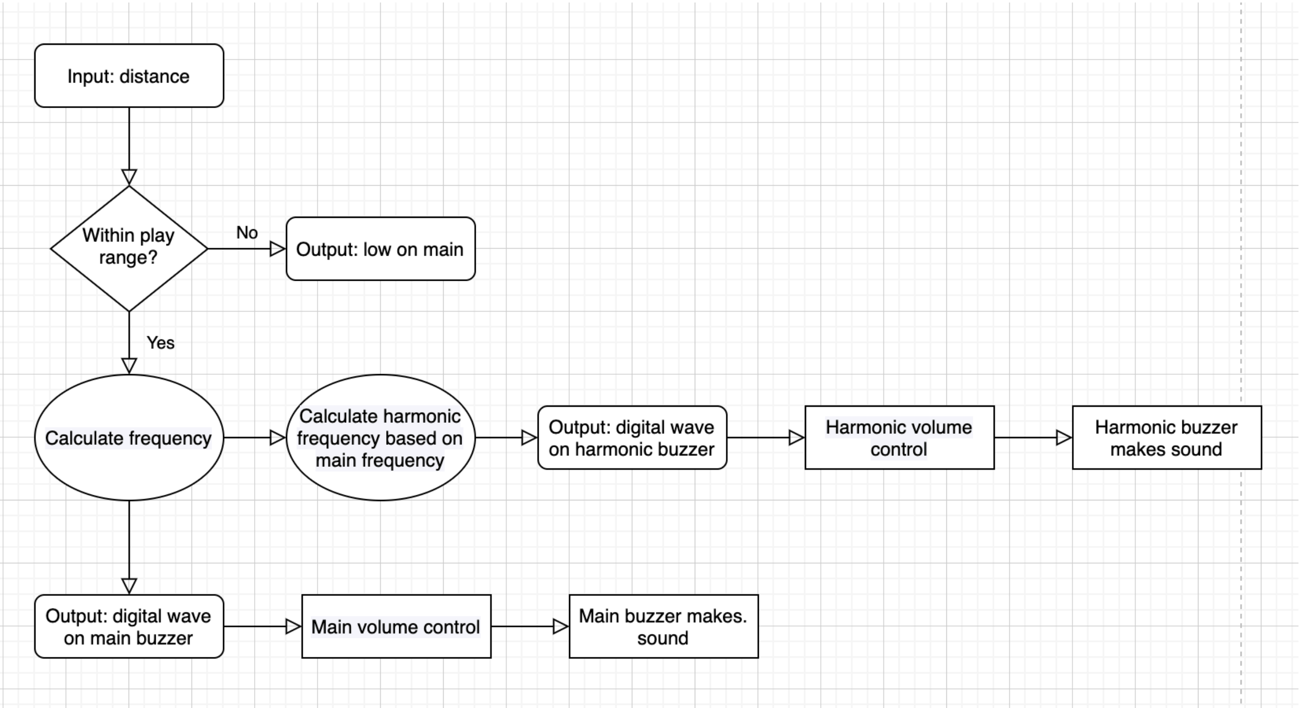
2. Sustain/Damping

   Although damping, or an exponential decay in the outputted sound wave, is not hard to calculate, but so far in my implementation, the output to the passive buzzers are fixed to be a square wave at a frequency correlated to the input distance. Then I realized that I have more freedom in selecting the outputted wave, so that it may be waves that stack waves of multiple frequencies but still make a meaningful sound.

   Another idea is to make a feedback loop which attenuates the output from the buzzer and sends the signal back to the buzzer. This aims for creating the similar effect as the sustain pedal in a piano. Safety issues need to be considered so the buzzer is not overloaded, which requires selecting certain resistance and not saturate the input.

[1] Musical Notes. (n.d.). Retrieved November 30, 2020, from http://techlib.com/reference/musical_note_frequencies.htm

Structure of code and implementation:


Input: distance
Within play range?
No
Output: low on main
Yes
Calculate frequency
Calculate harmonic frequency based on main frequency
Output: digital wave on harmonic buzzer
Harmonic volume control
Harmonic buzzer makes sound
Output: digital wave on main buzzer
Main volume control
Main buzzer makes. sound

(Up to the second and partially the third stage)

Parts needed outside of my Arduino kit:
2 additional passive buzzes (already acquired)