DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
FALL 2019

RESONANCE
LTUNES

AMIR DHUNGANA
ANISH YONJAN
ROBERTO TORRES
RAUL JIMENEZ
RABINSON SHRESTHA
NIKHIL PUROHIT

Revision History

Revision	Date	Author(s)	Description
0.1	02.14.2020	AD	document creation
0.2	02.20.2020	AD	complete draft
0.3	02.21.2020	AD	release candidate 1
1.0	02.21.2020	AD	official release
1.1	05.11.2020	AD	revision and update

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

This section describes the purpose, use and intended user audience for the L-Tunes laser harp. L-Tunes is a laser-based digital instrument and MIDI device that uses lasers to trigger a note when the laser is blocked or obstructed. The device can be used to play sounds like an harp, with presets and parameters to emulate other instruments and sounds.
Users of L-Tunes will be able to play the device like an instrument and even create new sounds via built in sound generators. This product is intended for anyone, but particularly musicians, harp-enthusiasts, and children.
The L-Tunes laser harp should be used to play various sounds using the sound font files that have been loaded into the teensy micro-controller. One can change the preset of the sound using the touch screen, which further allows the user to control the volume,reverberation and chorus.
This device can be used by anyone, however the device is intended to be used by musicians, harp-enthusiasts, and young children. This device is designed for anyone looking for an alternative to a real harp with strings.

2 SYSTEM OVERVIEW

Laser Harp consists of four layers. The first layer is the power system, which is going to act as a power source for the device and consists of battery or similar power source. Second layer is frames and components which consists of laser beams, receptors, mister and speaker. Receptors are going to identify the interference in the laser beam and send the signal towards MIDI converter. Mister is being used for visibility whereas speakers are for audio output. Then, we have MIDI converter which is going to receive signals from the receptors and encode those signals as MIDI signals. Finally, the last layer is the sound module which is going to decode the MIDI signals, pass the resulting outcome to fluid synth which in turn sends the audio signals to speakers via sound card.

The laser harp instrument's UI consists of a touch screen monitor which allows the users to change the sound fonts as well as change the reverberations, gain,etc. of the outgoing sound.

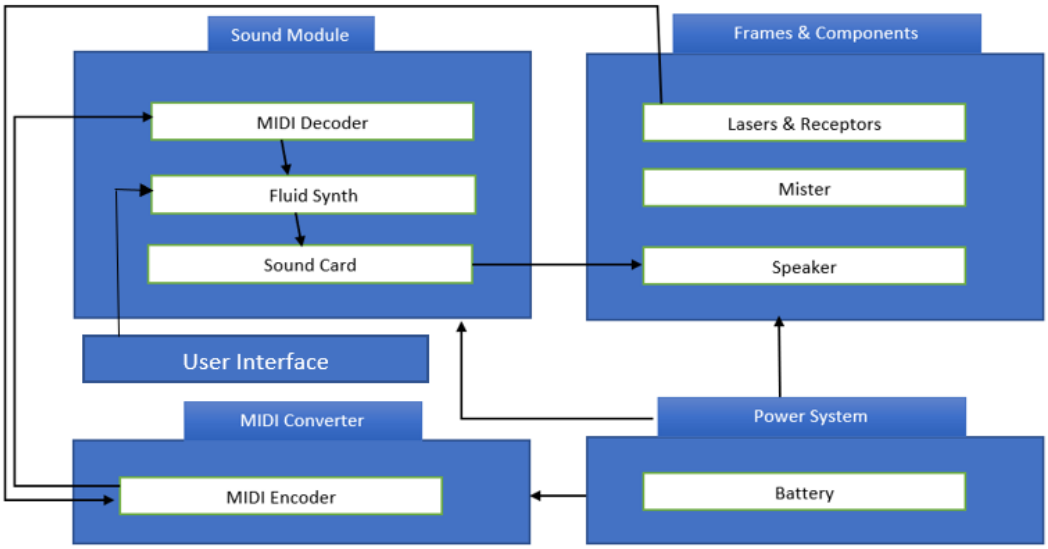

Figure 1: System architecture

3 Power Layer Subsystems

This layer is responsible for providing necessary power for the instrument. It mainly consists of 12V 2A DC batteries or regular 120V AC current can be used for this layer.

3.1 Layer Hardware

There is no specific hardware required for this layer except of combination 12V batteries and the outlet cord for AC supply . This layer is merely acting as an power source for the raspberry pi, lasers' circuit, teensy micro-controller and mister.

3.2 Battery/AC outlet

It consists of 12V 2A DC batteries' combination or AC outlet along with hookup and jumper wires for connection to the pi and the breadboard.

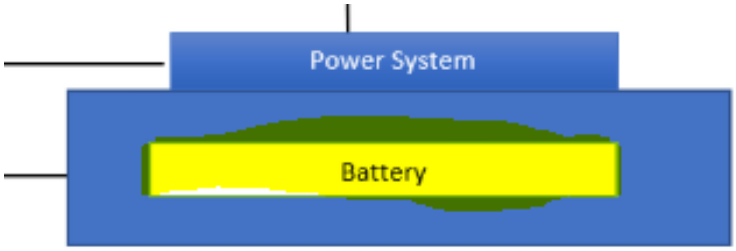

Figure 2: Power layer subsystem diagram

4 Frame and Component Layer Subsystems

This layer is going to be the interaction layer between the users and the instrument. It consists of a frame possibly made from wood or 3D printed plastic parts. Lasers and photo-resistors will be lodged in to the cavities of the frame. A mister will be there to increase visibility. Traditional 5mm RGB tri-color LED will act as indicators for which lasers were interrupted. Amazon-Basics speakers with frequency range from 103 Hz - 20 KHz will be used for the output sound.

4.1 Layer Hardware

This layer will consist of traditional speakers,USB connected misters, LEDs,laser diodes and photo-resistors.

4.2 Layer Operating System

Teensy 3.6 will detect the any breakage in the circuit which will be encoded as MIDI signals.

4.3 Layer Software Dependencies

Basic C++ arduino script will be running on the teensy to detect any circuit changes.

4.4 Lasers and receptors

This consists of laser diodes and photo-resistors connected to a breadboard which is in turn connected to teensy micro controller which detects the lasers' interruption.

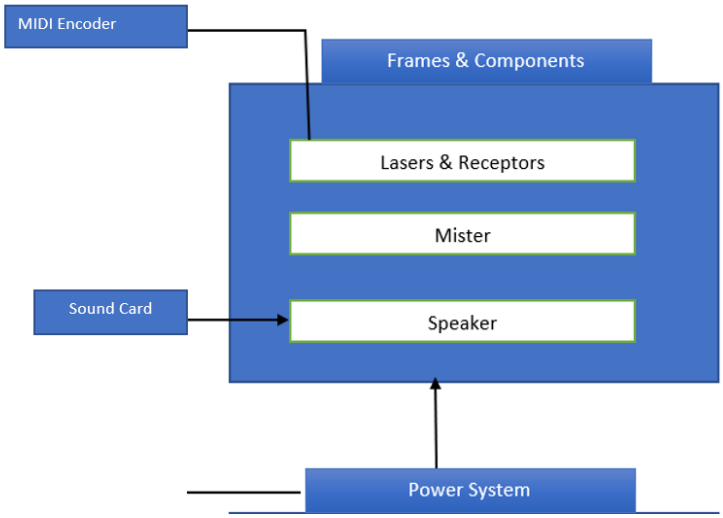

Figure 3: Lasers Receptors subsystem description diagram

4.4.1 Subsystem Hardware

Laser diodes will be 6mm 5mW red dot laser head producing 650 nm laser waves running on 5V power supply whereas the photo-resistors are 5 mm GM5539 resistors with spectral peak of 540 nm.

4.4.2 Subsystem Operating System

N/A

4.5 Mister

It consists of an AGPtck aluminium mini mist maker which will be submerged in water to produce mists which will act as reflective medium for lasers to increase the visibility.

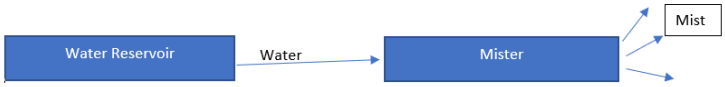

Figure 4: Mister subsystem description diagram

4.5.1 Subsystem Hardware

A homemade fog machine or a AGPtek mist maker of 1.8 inch diameter with DC 24V power source.

4.6 Speakers

They will act as the output of the whole instrument which consists of USB powered speakers for portability.

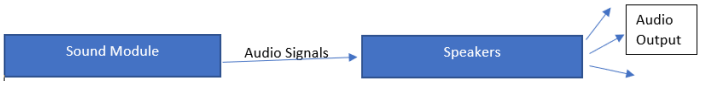

Figure 5: Speakers subsystem description diagram

4.6.1 Subsystem Hardware

USB powered speaker(5V) with simple setup having easy front-access control for power and volume.

5 MIDI Layer Subsystems

This layer is responsible for converting the analog signals coming from the layer and receptors subsystem to MIDI signals which can be analyzed by the midi decoder.

5.1 Layer Hardware

It consists of Teensy 3.6 micro-controller.

5.2 Layer Operating System

Teensy micro-controller is running a arduino script to detect the interference in the laser's circuit.

5.3 Layer Software Dependencies

MIDIUSB library is used in order to allow the micro-controller to encode the MIDI signals for the pi.

5.4 MIDI Encoder

It is C++ arduino script running on Teensy which converts the analog reading to MIDI signals.

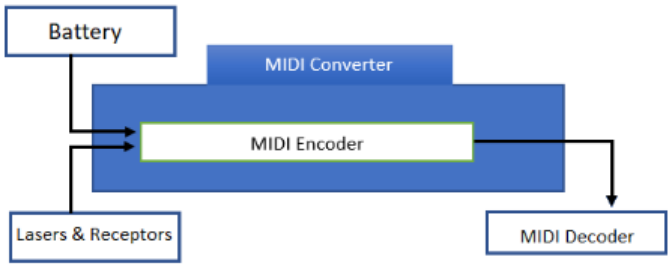

Figure 6: MIDI encoder subsystem description diagram

5.4.1 Subsystem Hardware

Teensy 3.6 180 MHz micro-controller with 32 General purpose DMA channels.

5.4.2 Subsystem Operating System

Arduino 1.8 or similar in order correctly run the script.

5.4.3 Subsystem Software Dependencies

MIDIUSB library has been used to call functions to send notes or signals to raspberry pi.

5.4.4 Subsystem Programming Languages

Basic C++ arduino language is being is used for reading the signals from the circuit.

6 Sound Modules Layer Subsystems

This layer is responsible for reading the MIDI signals and converting it into audio signals that is needed for the speakers to produce audio output. It consists of MIDI decoder, fluid synth and sound card which are integrated in a Raspberry Pi.

6.1 Layer Hardware

It consists of 1.5GHZ quad-core 64-bit raspberry pi.

6.2 Layer Operating System

It runs raspbian OS with kernel version 4.19.

6.3 MIDI Decoder

It is a driver program that acts as an bridge program between MIDI encoder and the fluidsynth.

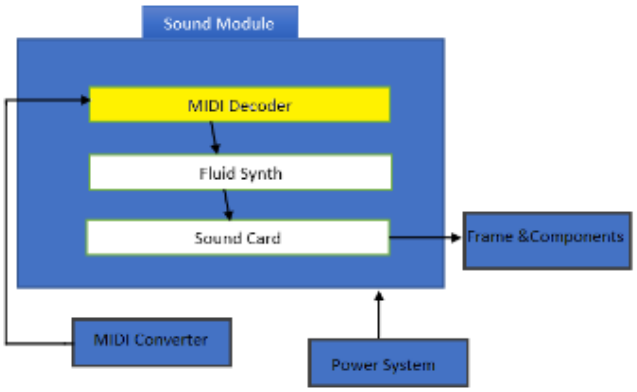

Figure 7: MIDI decoder subsystem description diagram

6.3.1 Subsystem Hardware

It is running on Raspberry pi 4.

6.3.2 Subsystem Operating System

Raspbian OS is running a python 3.6 script that reads the encoding.

6.3.3 Subsystem Programming Languages

It uses Python 3.6.0 for server processing which uses socket library for communication with fluidsynth.

6.4 Fluidsynth

Fluidsynth is a real-time MIDI synthesizer based on the SoundFont 2 specifications.

6.4.1 Subsystem Operating System

It is runs on startup on a raspbian OS.

6.4.2 Subsystem Software Dependencies

Fluidsynth is a library itself which reads in the MIDI input and plays according to the sound font files present. Moreover, socket library is called upon to connect with the fluidsynth process which in turn uses AF_INET IPV4 protocol for communication.

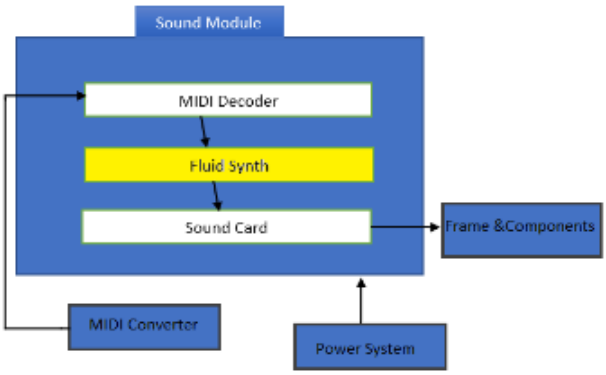

Figure 8: Fluidsynth subsystem description diagram

6.4.3 Subsystem Programming Languages

Python is used to call upon the fluidsynth functions to control the gain, reverb and chorus of the audio output.

6.5 Soundcard

Raspberry pi already comes in with a sound card integrated in its system. It is used for converting the output data from fluidsynth to audio data used by speakers for final output.

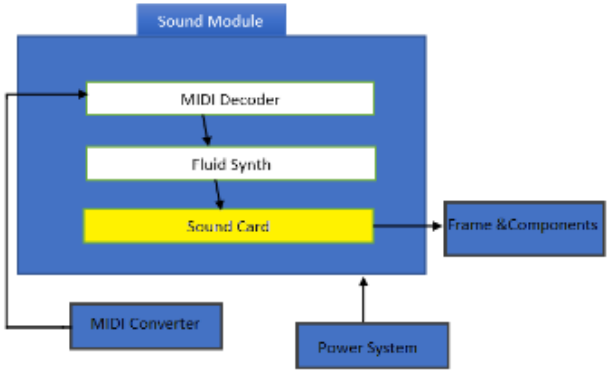

Figure 9: Sound Card subsystem description diagram

6.5.1 Subsystem Hardware

Raspberry pi 4 with compatible sound drivers for necessary sound processing.

6.5.2 Subsystem Operating System

Raspbian OS

7 User Interface

This layer allows the user to control the sound font files to use such as church,acoustic guitar,etc. Also, it allows the user to change the reverb, gain and chorus of the sound as well as switch back and forth between the presets.

7.1 Layer Hardware

Elecrow 5 inch capacitive touch screen with 800*480 TFT LCD display is used as an interactive medium with the user.

7.2 Layer Operating System

It depends on the Raspbian OS.

7.3 Layer Software Dependencies

Python 3.6 is used to create the GUI for the application which uses tkinter library to create the button and sliders to control the audio output. Moreover, in order improve the UI of the design we have decided to use kiwy python library so that the user has more pleasant experience while operating the device.

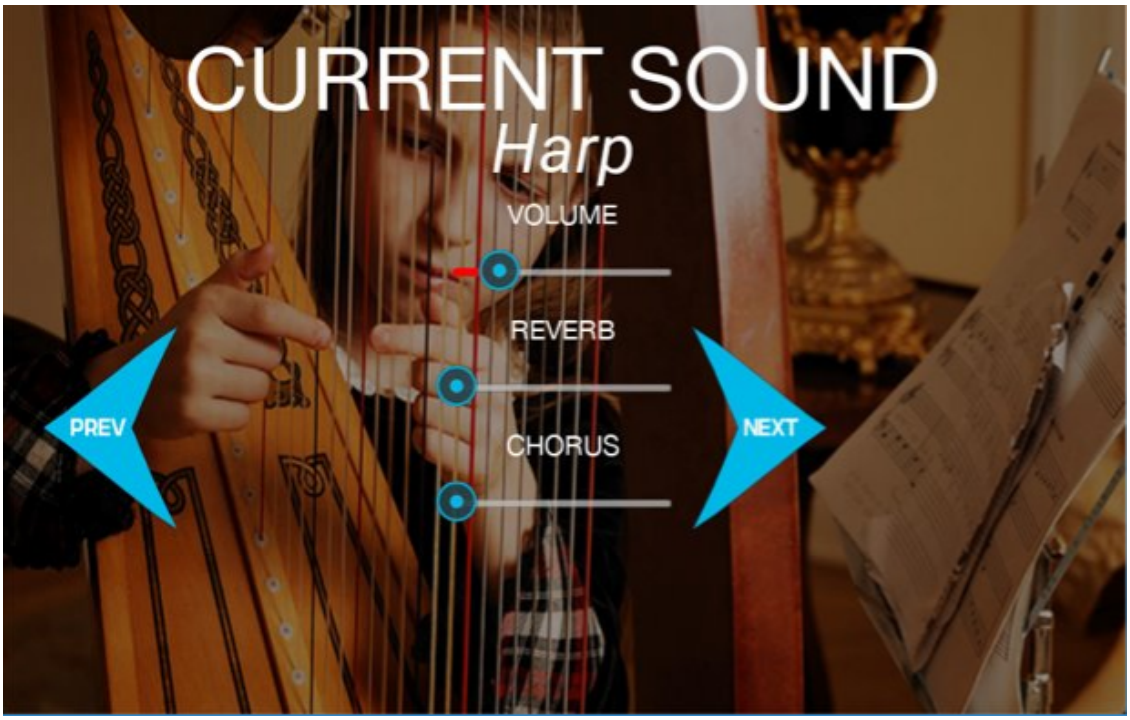

Figure 10: Sample User Interface

8 APPENDIX A

Final design model of the product,along with its views.

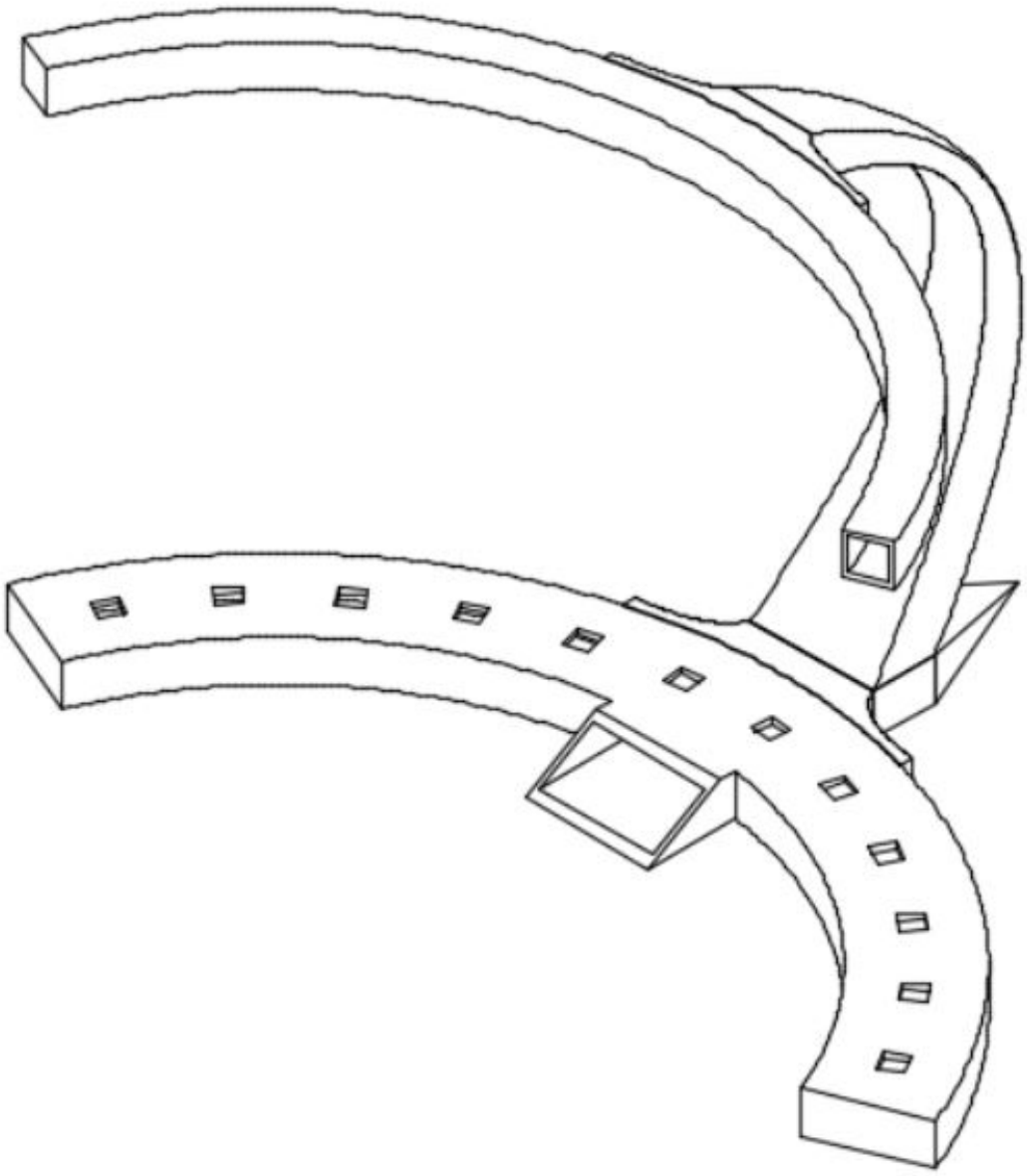

Figure 11: Final model of the final product

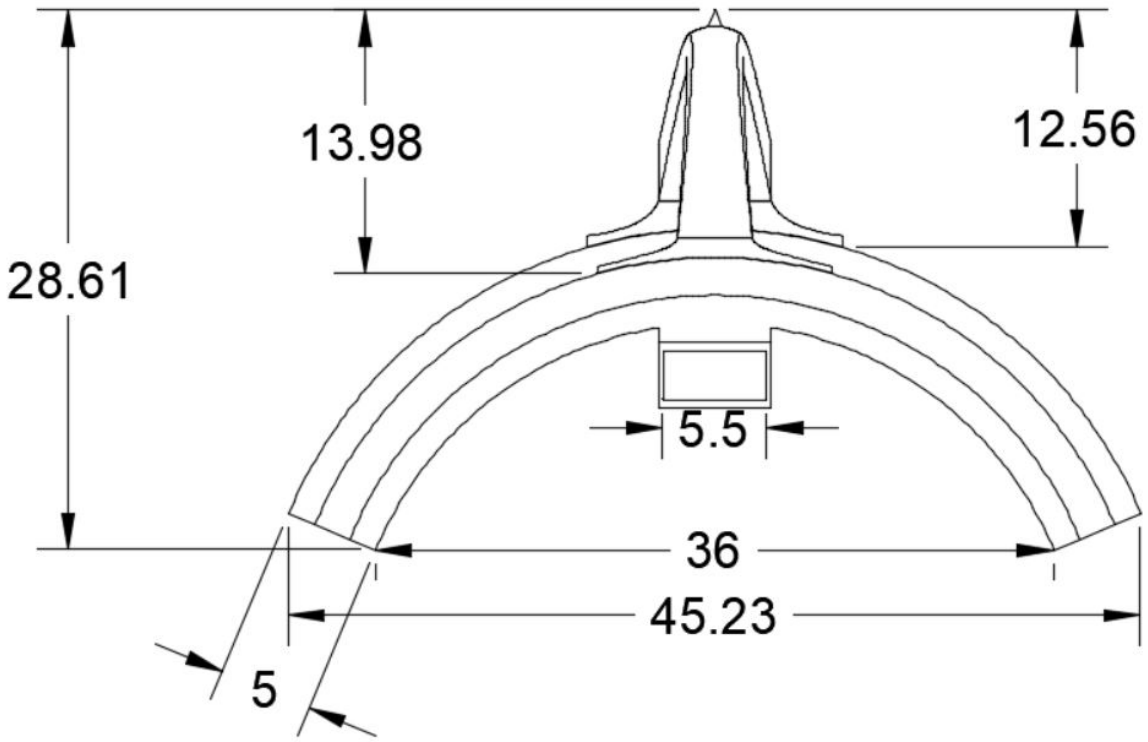

Figure 12: Top view of the model

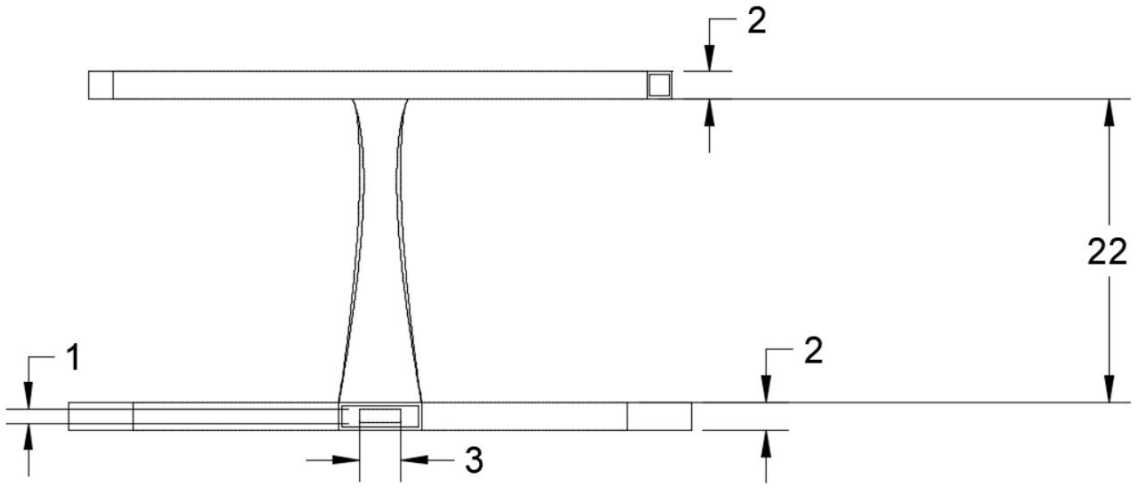

Figure 13: Front view of the model

REFERENCES

Fluidsynth manpage-https://manpages.debian.org/testing/fluidsynth/fluidsynth.1.en.html

Tkinter- http://effbot.org/tkinterbook/