Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2019

Light Luthiers
Laser Harp

Sohaib Siddiqui
Mitchell Lux
Michael Morrow
Zackery Gualandi
Siddharth Banerjee

Revision History

Revision	Date	Author(s)	Description
0.1	11.01.2019	SS, ML, MM, ZG, SB	document creation
0.2	11.06.2019	SS	setup and inital changes
1.0	11.11.2019	SS, ML, MM, ZG, SB	complete initial draft
2.0	12.11.2019	SS, ML, MM, ZG, SB	update Section 4, 5, and 6

CONTENTS

List of Figures

List of Tables

1 Introduction

The Laser Harp by the Light Luthiers will be a portable device that will play a full octave of musical similar to what a traditional harp would play. Instead of strings to pluck like in a traditional harp, the Laser Harp will have lasers that will play the notes when each laser beam is interrupted. It will have a built in speaker to play the notes, and may have support for MIDI/AMP output.

The Laser Harp is to be showcased to students attending school with the primary aim of encouraging students to pursue careers in engineering, or STEM in general, by giving an example of the interesting and creative things one can do in STEM fields.

NOTE: any features not indicated as tentative are part of the minimum requirements for the product. See the Software Requirement Specification for more information on the requirements of the product.

1.1 Design

This section provides an overview of how the Laser Harp functions internally, as well as how it will look externally and how it will be operated by users. The Laser Harp design will be developed based on our System Requirements Specification document to produce all needed features that work reliably and safely. The device take in multiple different inputs from the user to produce different sounds. Based on these parameters and the users interactions with the lasers the device will output sound to a user-specified output type. The physical design will be a compact portable device because of the importance of the mobility of the product. This device must be safe for use with almost all ages so characteristics such as rounded edges and safety protection are considered in the design process as well.

1.1.1 Functional Overview

There is a mirror that will be mounted (via a 3D printed mount) to a stepper motor. A laser beam is directed to the mirror, and the motor rotates in steps. With every step the motor takes, the position of the mirror changes, reflecting the beam in a different direction each time. The steps occur fast enough for the beams to appear as if they are simultaneously present.

When one or more of the beams are cut, the light sensor/light detecting resistor detects it. A signal will be sent to an Arduino, which in turn will play the note corresponding to the position of the motor when the laser beam was interrupted. This works because every beam has a corresponding motor position. When a beam is cut, it is reflected onto the sensor. the sensor then detects an increased light intensity. Then, the position of the motor is determined by the time at which a laser beam was interrupted.

1.1.2 Product Overview

The internals of the harp will be encased by a housing that will be 3D printed in the Senior Design Lab, which is located at room 240 of the Engineering Research Building of UTA. As of ADS v1.0, the design of the housing has yet to be finalized. There will be a built-in speaker, and an on/off switch. Tentatively, there may be 1/4 inch audio output as well as a MIDI jack for output to an amp or computer, respectively. There also may be ways to change the voices/tones via a switch of the proximity of the users' hands from the Laser Harp, and volume control.

2 System Overview

This section describes the overall structure of the software component of the Laser Harp. The software is divided into three layers, the input layer, control layer, and output layer. Each layer has specific subsystems that make up the entire layer. This below diagram shows the flow between the layers.

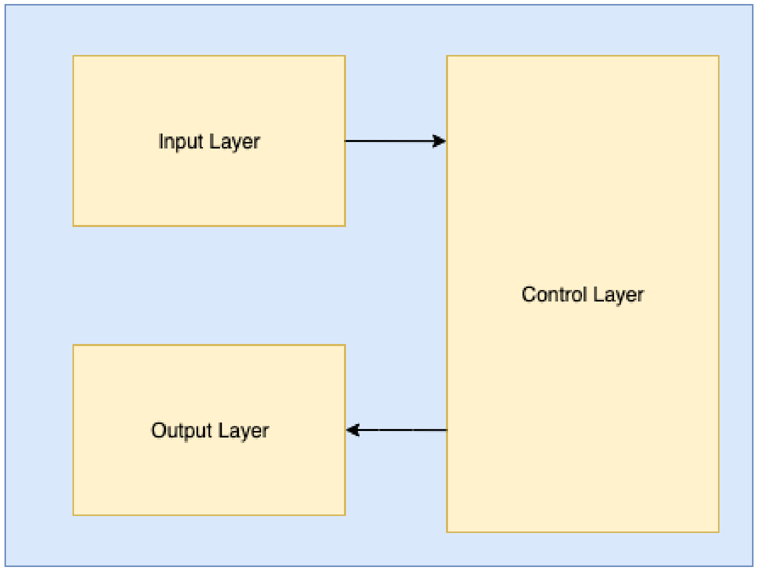

Figure 1: Layer abstraction of software

2.1 Input Layer Description

This layer communicates with the input modules such as the on/off switch, the lasers and sensors, voice, and volume. When the power is switched on, it will signal to the laser and motor to turn on. It will detect if a MIDI jack or quarter-inch audio output jack is plugged in and determine which output medias are available. These inputs are then sent to the Control Layer and are used to run and operate the device.

2.2 Control Layer Description

The Control layer encompasses everything the user interacts with and observes from the device, such as the part that controls the stepper motor, i.e. it's steps, and frequency. The layer controls the parts of the device that the user interacts with like emitting the lasers correctly, and using the different settings to change the sound. This layer also communicates with the light detecting resistor to detect when a beam was disrupted by the user. It will determine the time at which the beam was interrupted. With this, it will also determine the position of the motor when a beam was broken, and communicate it to the Output Layer so that the corresponding note may be played to the appropriate output medium.

2.3 Output Layer Description

The purpose of the Output Layer is to determine which audio output medium is appropriate to play notes to; speakers, MIDI, or amplifier/external speakers. Then, once determined it will play the audio to the proper output(s). It also is used to for displaying things to the user such as the separate beams to interact with, or setting information so they know what exactly is affecting the output sound.

3 Subsystem Definitions & Data Flow

The system overview of our product creates a 3-layer system based upon collecting input, controlling that input, and outputting sound through different output channels. The input layer gathers different settings for the voices, volume, and octave levels. Then based on these settings and the users interactions with the lasers, the device will use these inputs to control an audio output signal. This signal can be formatted and sent to different audio-outs based on user-preference.

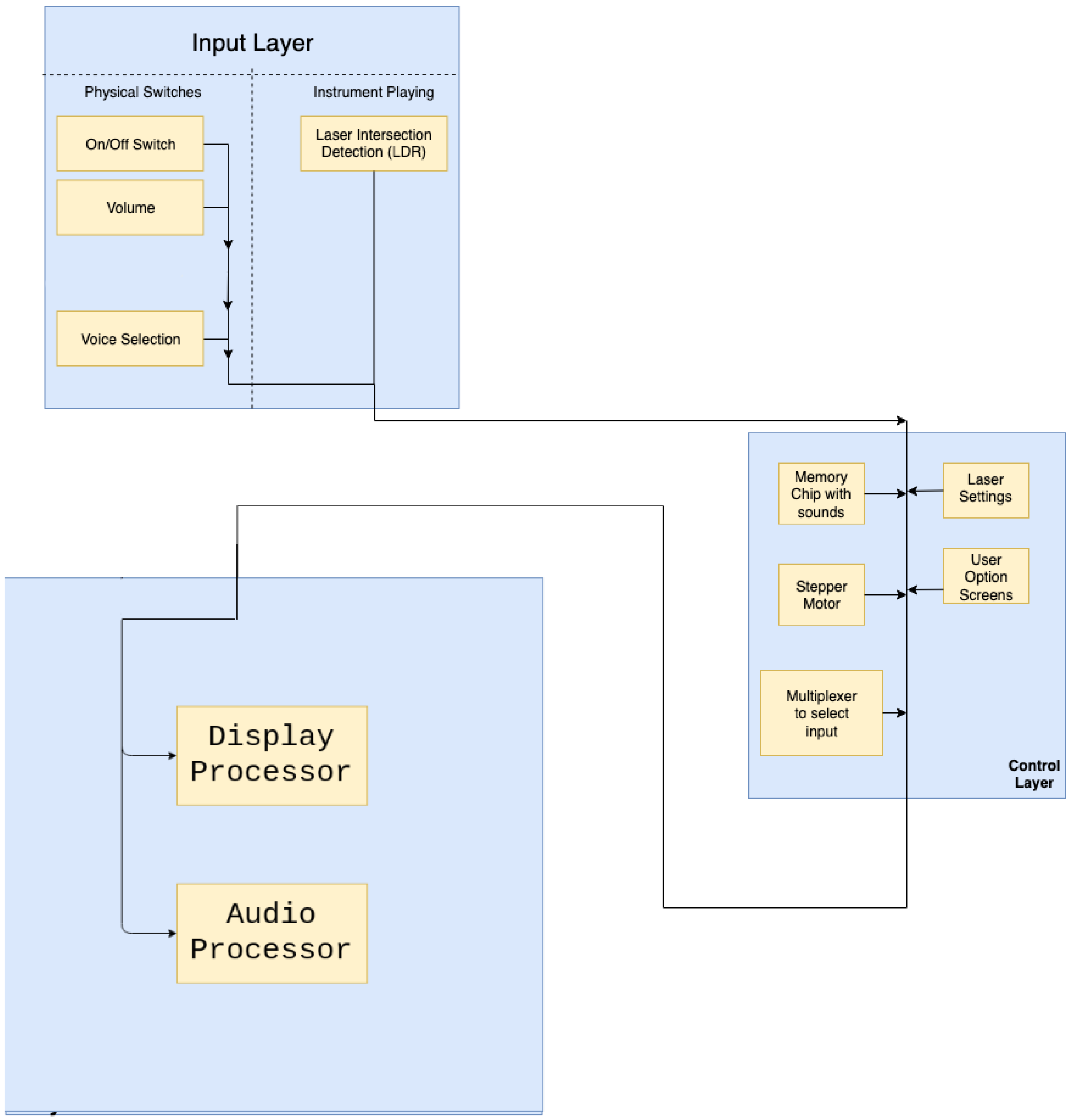

Figure 2: Data flow diagram

4 Input Layer Subsystems

This layer is divided into two parts, the physical switches located on the outside frame of the Laser Harp, and the laser detection system that indicates what notes should be played

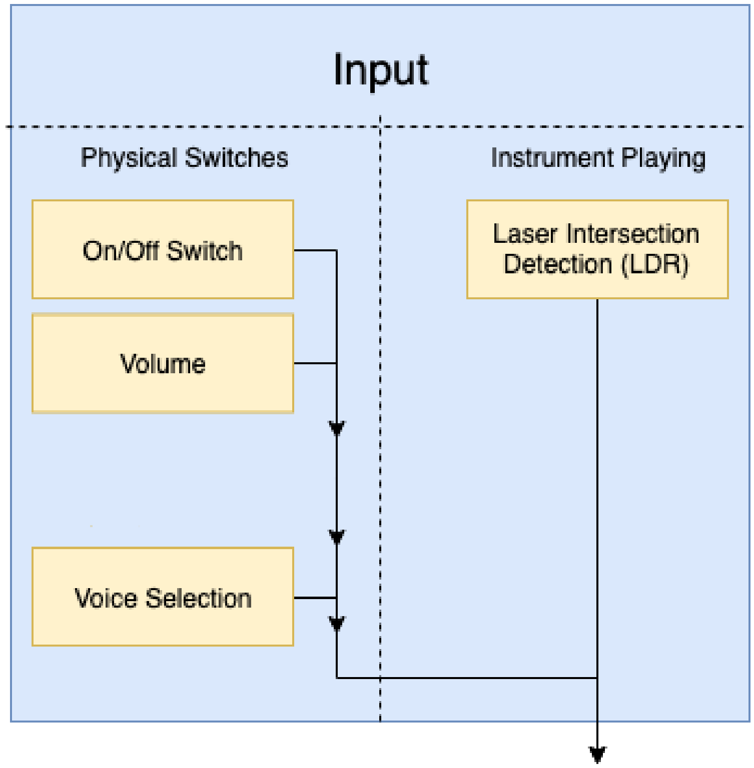

Figure 3: Input Layer Diagram

4.1 On/Off Switch

Turns the product on or off

4.1.1 Assumptions

The product is sufficiently powered and the switch is not damaged.

4.1.2 Responsibilities

The switch is to signal to all the layers and subsystems to power on and begin working or to shut down.

4.1.3 Subsystem Interfaces

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	On/Off Switch	On Off	On Off

4.2 Volume knob

Used to adjust the volume of the sound being played by the Harp.

4.2.1 Assumptions

The knob is not damaged, and the connections to input/control layer are intact.

4.2.2 Responsibilities

The knob will be used to indicate to the input layer that the volume is being adjusted, and the input layer will communicate the adjustment to the output layer.

4.2.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#02	Volume knob	0 - 100%	Volume value 0-100

4.3 Voice selection

This will be used by the user to select which voice to play notes in e.g harp, piano, guitar etc...

4.3.1 Assumptions

The knob and connections to the switch are not damaged.

4.3.2 Responsibilities

This will have a selection of instruments for the user to choose from. When chosen the coressponding sounds will be played for the selected voice.

4.3.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#02	Voice selector	TBA	Different voice outputs

4.4 Light Intersection Detection

The Intersection Detection layer takes on two roles using the LDR(Light Dependent Resistor) and sonar sensor attached to play the correct sound the user wants.

4.4.1 Assumptions

These systems assume that the device is clean, powered, and ready to operate. That the proper setting have been loaded via the logic layer. Additionally, the user must use pronated hands or flat horizontal surface to interrupt the laser beams.

4.4.2 Responsibilities

This sub-layer has the job of detection which laser is being intersected to play the correct note and also detect distance of the intersection to decide the active octave.

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	LDR sensor	Sensor that detects when 1 out of 13 laser are intersected	Code informing logic layer of which laser has been intersected
#02	Sonar Sensor	Sensor that detects distance of hand from beam	Code informing logic layer of distance

5 Control Layer Subsystems

The logic layer has multiple tasks to achieve. The logic layer will select the sound to play based on the input from the input layer. It will also control the interface for users to interact with the lasers and select the right sound output and the settings desired.

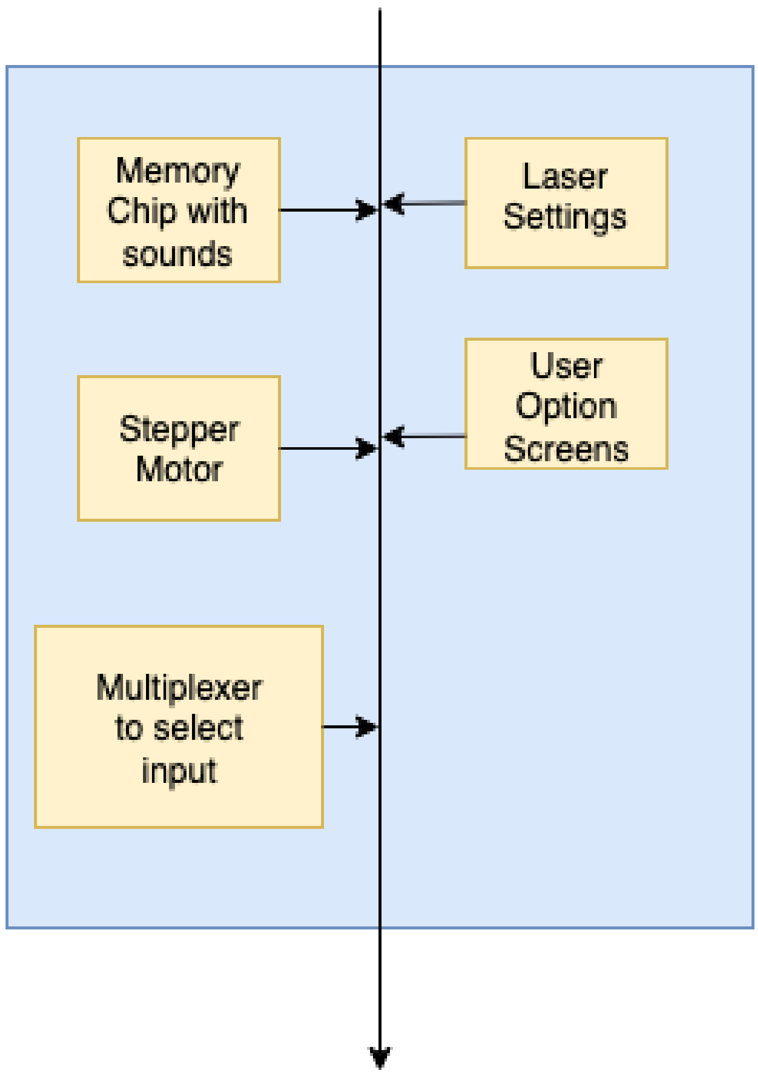

Figure 4: Control Layer Diagram

5.1 Memory Chip with sounds

The first subsystem will deal with correctly selecting the sound based on the specific laser activated and pass the digital file to the output phase.

5.1.1 Assumptions

The memory is non-volatile and will save the information in correct order. It should only have read access.

5.1.2 Responsibilities

The primary responsibility of the subsystem is to save each playable sound in memory. We also need to be able to extract information from the memory. Memory chip stores the notes that can be played. The laser is pointed at a stepper motor attached to a mirror, which moves to create the illusion of having 13 beams simultaneously. The multiplexer determines the sound that needs to be produced.

5.1.3 Subsystem Interfaces

The output of this subsystem is the correct sound to be played.

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Memory chip that stores sounds	None	List of sounds

5.2 User Option Screens

User Option Screens

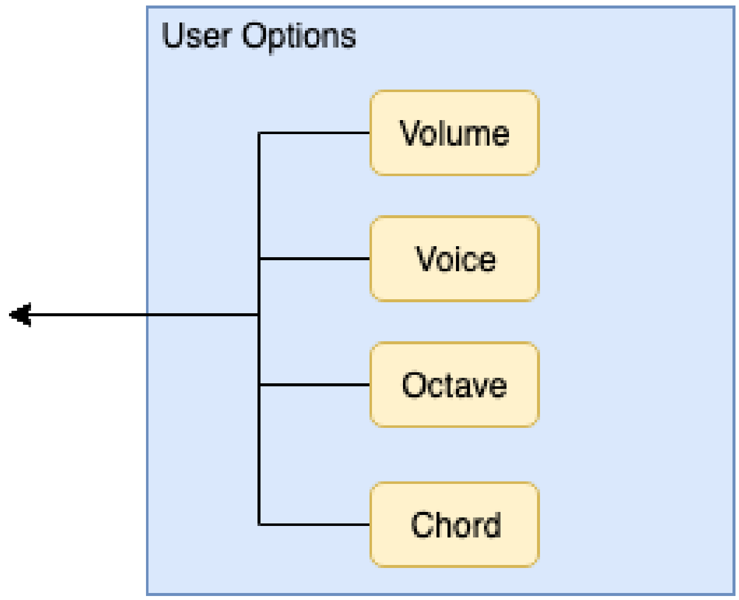

Figure 5: Control Layer Diagram

5.2.1 Assumptions

The user-options are digital, such that when the option is being read, there is not ambiguity.

5.2.2 Responsibilities

Save the user-settings such as volume and the output over which the sound has to be played.

5.2.3 Subsystem Interfaces

Interacts with input layer to receive the settings and saves it for purposes of selecting output.

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	User Option Screens	Sound and Output Settings	Select output devices

5.3 Laser Settings

Laser Settings will determine quantitative measurements about the laser. This includes the light intensity of the projected laser as well the ability to switch it off and on.

5.3.1 Assumptions

The system responds to change in laser intensity and the laser light are calibrated to the particular place.

5.3.2 Responsibilities

Change the light intensity of the laser as needed by the user. Additionally have the ability to switch on/off using a switch.

5.3.3 Subsystem Interfaces

The system takes inputs from the switches and then relays this information to the laser projector.

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Laser settings	Switches from input layer	Laser projector

5.4 Stepper Motor

This will control the mirror to create the laser "harp" effect.

5.4.1 Assumptions

The stepper motor will be run at constant rate to keep the same effect of separate beams for the user. The stepper motor will be able to attain needed speed. It will also have to be configured for the angle at which the 13 lasers will be emitted

5.4.2 Responsibilities

Will control moving the mirror to create the effect of separate beams but really coming from one laser.

5.4.3 Subsystem Interfaces

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Control of stepper motor Velocity	Constant Speed	Movement of Mirror
#2	Control of stepper motor Angle	Angle	Movement of Mirror

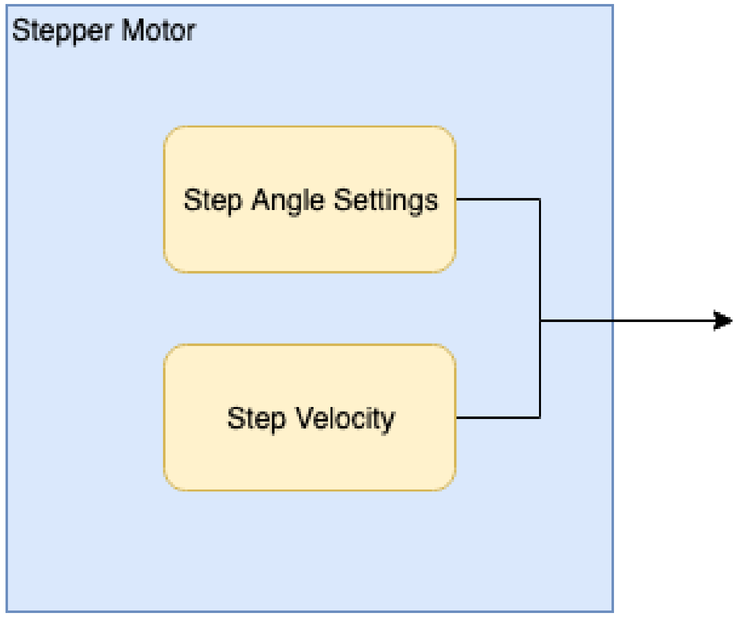

Figure 6: Control Layer Diagram

5.5 Multiplexer

Will mix all settings to create a realistic sound.

5.5.1 Assumptions

MUX selects the sound we need and does not have corrupted data bits.

5.5.2 Responsibilities

Will select the sound based on input code and and then use output code to access the particular sounds

5.5.3 Subsystem Interfaces

Table 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Multiplexer	Code to select sound	Code of the sound to be selected from memory

6 Output Layer Subsystems

This section details the structure of the output layer and its subsystems. The output layer handles the production of any info leaving the harp. It will construct audio signals, midi streams, and drive the laser array. There are two main subsystems; the display processor, and the audio processor.

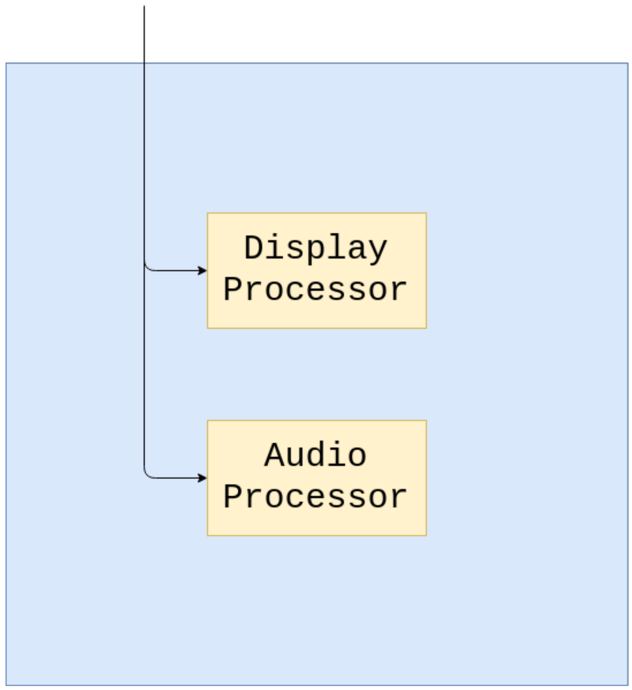

Figure 7: Output Layer Diagram

6.1 Display Processor

The display processor communicates the instrument state to the player. It will keep up-to-date the LCD screens indicating the selected voice and the octave that the musician is playing in. The processor needs to update as soon as the player changes the Octave and Voice without noticeable delay. It also will drive the laser array. A potential problem with this approach is that the array of lasers requires specific timing to create the illusion of multiple lasers.

6.1.1 Assumptions

It is assumed that the display processor will be given necessary, well formed data from the Control Layer in a timely matter. If this data is late this could lead to incorrect info in the LCD screens.

6.1.2 Responsibilities

The display processor is responsible for updating the position of the stepper motor creating the fan of lasers, laser array, and updating the LCD screens indicating the chosen voice and the octave.

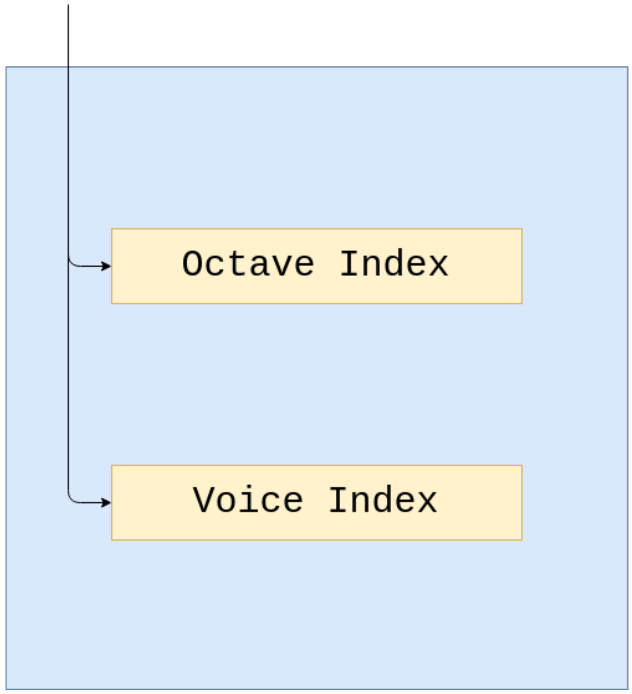

Figure 8: Display Processor Diagram

6.1.3 Subsystem Interfaces

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Octave Index	0-n octave	0-n octave
#02	Voice Index	0-n voice	0-n voice

6.2 Audio Processor

The audio processor is the unit that will generate the audio/midi signal. It will generate an analog signal for the speakers based on the string being broken, the octave chosen, the voice chosen, the volume level, and the distortion level. The audio processor will construct a midi packet to send over USB if a USB cable is plugged into the device.

6.2.1 Assumptions

It is assumed that the data being sent to is well formed and delivered quickly. If data is sent too slow this could cause issues in signal generation and lead to unwanted noise.

6.2.2 Responsibilities

The audio processor is responsible for generating the analog signal and routing it to the speakers or 1/4" jack, or constructing a midi packet to be sent over USB.

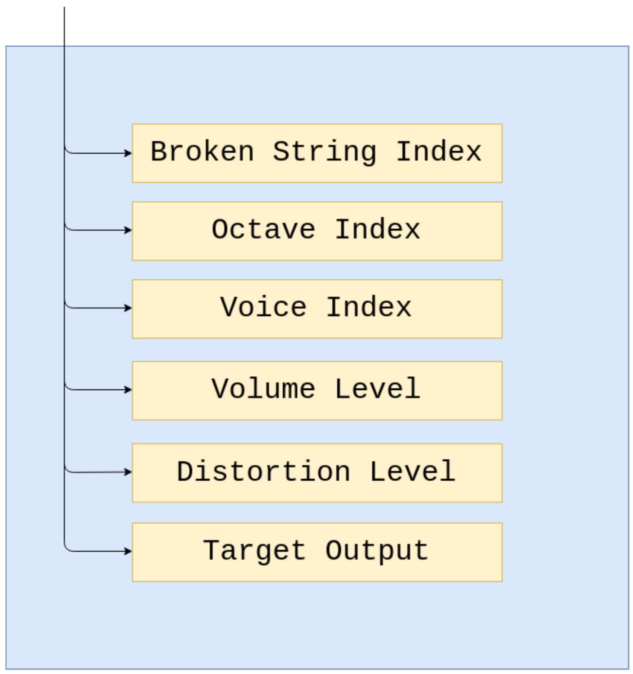

Figure 9: Audio Processor Diagram

6.2.3 Subsystem Interfaces

Table 12: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Analog Signal Generation	String Index Voice Index Octave Index Volume Level Distortion Level	Analog Signal
#02	Midi Packet Generation	String Index Octave Index Volume Level Distortion Level	Midi Packet

REFERENCES