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

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2020

Light Luthiers
Laser Harp

Sohaib Siddiqui
Michael Morrow
Zackary Gualandi
Mitchell Lux
Sidharth Banerjee

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	2.20.2020	SS	document creation
0.2	2.22.2016	SS	completed section 1
0.3	1.12.2016	AT, GH	release candidate 1
1.0	1.20.2016	AT, GH, CB	official release
1.1	1.31.2016	AL	added design review requests

CONTENTS

1 Introduction **5**

2 System Overview **5**
- 2.1 Input Layer Description . . . 5
- 2.2 Control Layer Description . . . 5
- 2.3 Output Layer Description . . . 7

3 Input Layer Subsystems **8**
- 3.1 Layer Hardware . . . 8
- 3.2 Layer Operating System . . . 8
- 3.3 Physical Switches . . . 8
- 3.4 Instrument Playing . . . 9

4 Control Layer Subsystems **10**
- 4.1 Layer Hardware . . . 11
- 4.2 Layer Operating System . . . 11
- 4.3 Layer Software Dependencies . . . 11
- 4.4 Stepper Motor Subsystem . . . 11
- 4.5 Laser Subsystem . . . 12
- 4.6 Sonar Sensor Subsystem . . . 13

5 Output Layer Subsystems **14**
- 5.1 Layer Hardware . . . 14
- 5.2 Layer Operating System . . . 14
- 5.3 Layer Software Dependencies . . . 14
- 5.4 Display Processor . . . 14
- 5.5 Audio Processor . . . 15

List of Figures

1 Layer abstraction of software 6
2 Example subsystem description diagram 8
3 Example subsystem description diagram 9
4 Control Layer Overview 10
5 Stepper Motor Subsystem 11
6 Laser Subsystem 12
7 Sonar Subsystem 13
8 Example subsystem description diagram 14
9 Example subsystem description diagram 15

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.

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.

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.

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.

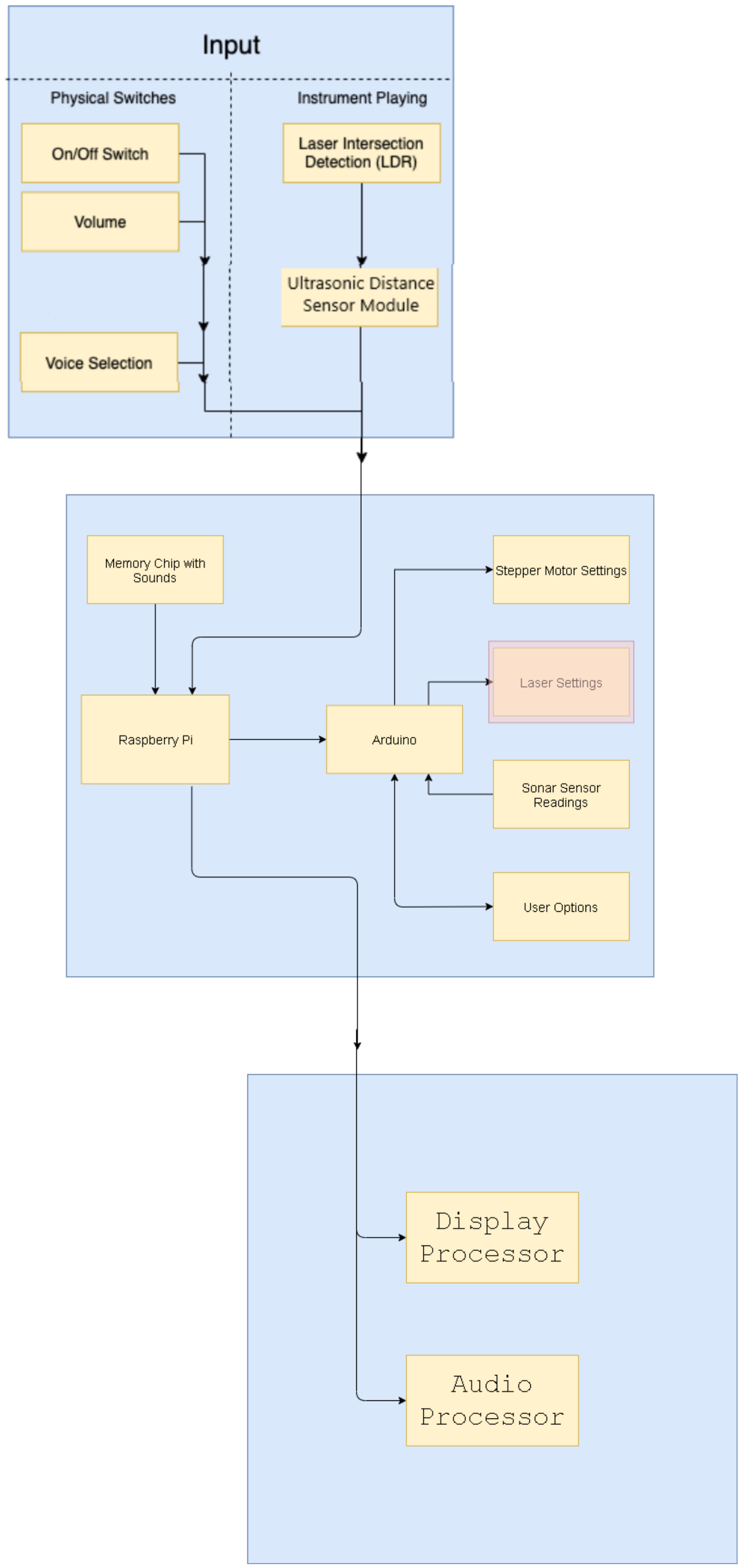

Figure 1: Layer abstraction of software

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 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. The input from this subsystem will feed information into an arduino board controlled by a raspberry pi computer.

3.1 Layer Hardware

The input layer consists of three physical switches: a simple on/off switch, a radio style knob, and a rotating knob that locks into set positions. The instrument playing consists of two sensors: an LDR (light dependent resistor), and an ultrasonic distance finder.

3.2 Layer Operating System

This layer does not directly use an operating system. Instead, it sends signals to the control subsystem.

3.3 Physical Switches

These switches are used to control the baseline state of the laser harp.

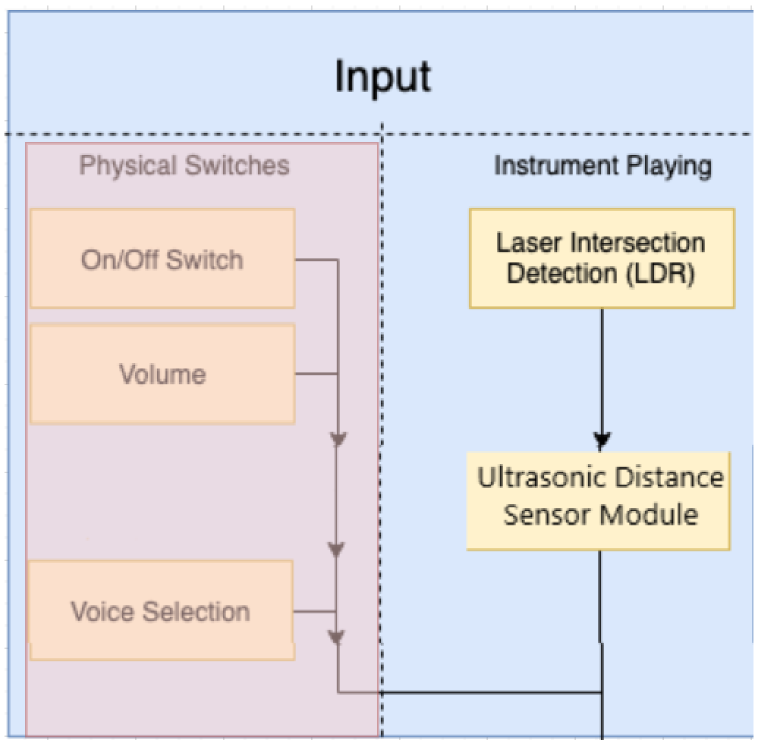

Figure 2: Example subsystem description diagram

3.3.1 Subsystem Hardware

The input layer consists of three physical switches: a simple on/off switch to control the power, a radio style knob that controls the volume, and a rotating knob that locks into set positions that is used to select which instrument to play.

3.3.2 Subsystem Data Structures

The physical switches will send a stream of simple numerical digits to the control system. The control system will then use these numbers to determine which settings to load up.

3.4 Instrument Playing

These two sensors are used to determine what notes to play and what octave to

3.4.1 Subsystem Hardware

The instrument playing consists of two sensors: an LDR (light dependent resistor) to determine what laser beam has been intersected. This is done by using the LDR and timing the reflections from the person touching the laser with the position of the stepper motor. It also includes an ultrasonic distance finder to determine what octave to play. This is done by the person playing the harp holding their hand at a certain distance above the sensor.

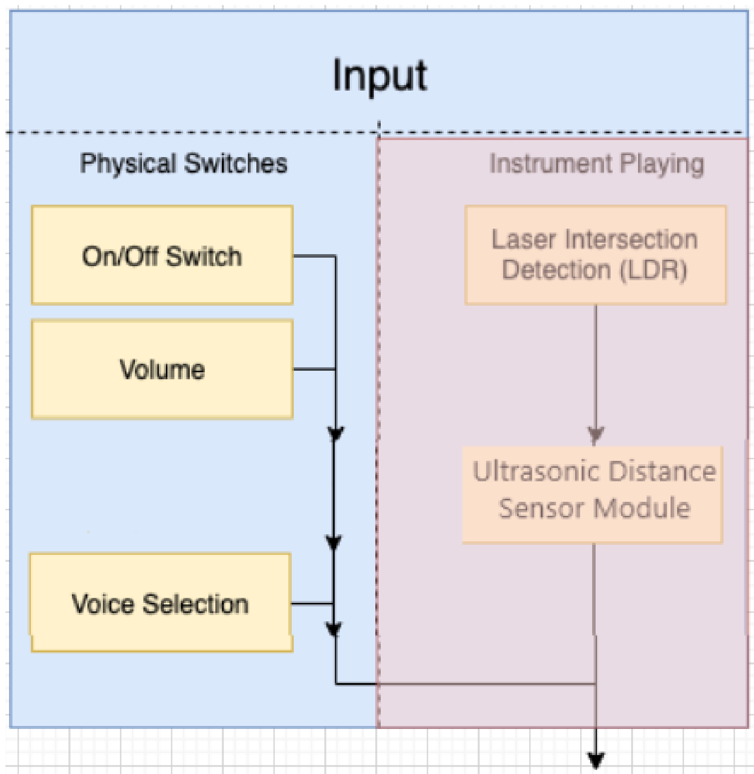

Figure 3: Example subsystem description diagram

3.4.2 Subsystem Software Dependencies

These sensors will interface with an Arduino board.

3.4.3 Subsystem Data Processing

The LDR will send a consistent stream of positive sensor reading based on what lasers are being intersected. The ultrasonic range finder sends a pair of numbers between 0 and 255 to the arduino board.

4 CONTROL LAYER SUBSYSTEMS

The control layer is the primary interface management between input and output of the system. It will be controlled by a Raspberry Pi. The Raspberry Pi will interface with the an arduino that controls a stepper motor, lasers, 6 sonar distance sensors and a set of user options to change volume, voice, octave and chord. The programming language used will be C++ and the Operating System will be the latest Raspbian Operating System. The Arduino program modules will be compiled on the Arduino IDE.

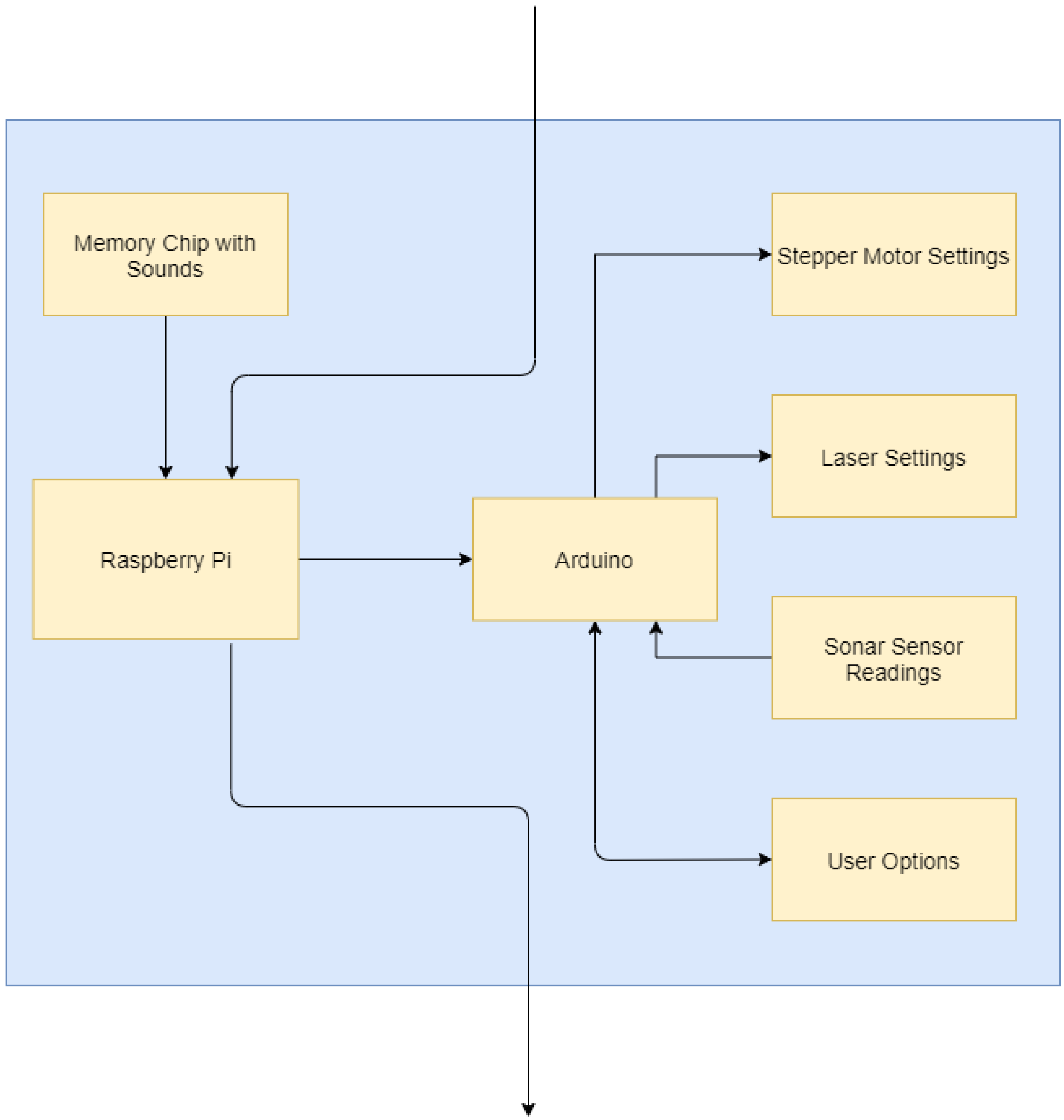

Figure 4: Control Layer Overview

4.1 Layer Hardware

The logic component will be controlled by a Raspberry Pi. The Raspberry pi will control the subsystems using an Arduino. Stepper Motor sub-system will control the pace and angle of the stepper motor. Laser Settings sub-system will manage the the laser emitter. The stepper motor settings will influence the operational speed of the laser. The sonar sensor sub-system will consist of 6 sonar depth distance sensors which will be separated by an angle of 5 degrees. This subsystem will decide which sound to play based on the sonar sensor triggered and the distance and angle at which it was triggered. Finally, User Options sub system will include digital hardware readings such as volume set by the user.

4.2 Layer Operating System

The Raspberry Pi will run on the latest LTS version of Raspbian OS.

4.3 Layer Software Dependencies

Arduino IDE

4.4 Stepper Motor Subsystem

The Stepper Motor subsystem will control the operation of a stepper motor. There will be a class which sets the stepper motor speed and angle. This will create the "harp" effect for our system.

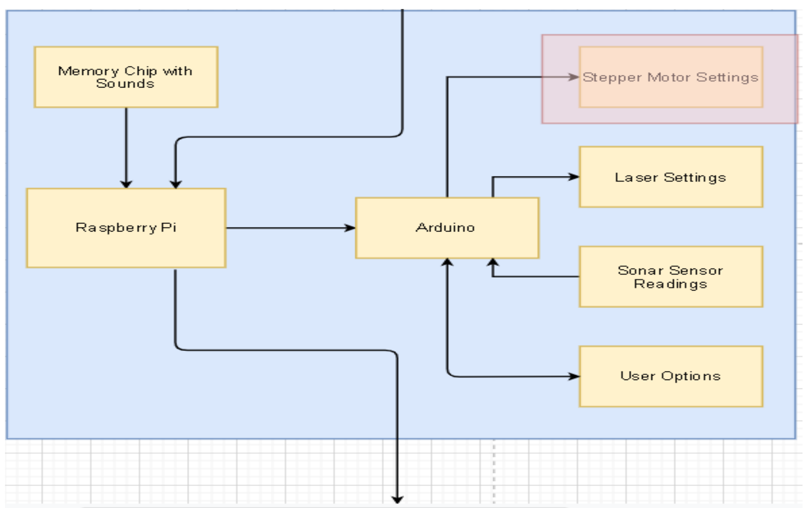

Figure 5: Stepper Motor Subsystem

4.4.1 Subsystem Hardware

Stepper Motor NEMA17

4.4.2 Subsystem Operating System

No Operating System required.

4.4.3 Subsystem Software Dependencies

None

4.4.4 Subsystem Programming Languages

C++

4.4.5 Subsystem Data Structures

A C++ class initializes the stepper motor and angle when the device starts up. This value will be constant and will not be user accessible.

4.4.6 Subsystem Data Processing

None

4.5 Laser Subsystem

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. A C++ class will be used to instantiate the laser intensity value. The ability to turn on and off the laser will be dependent on the input from the user.

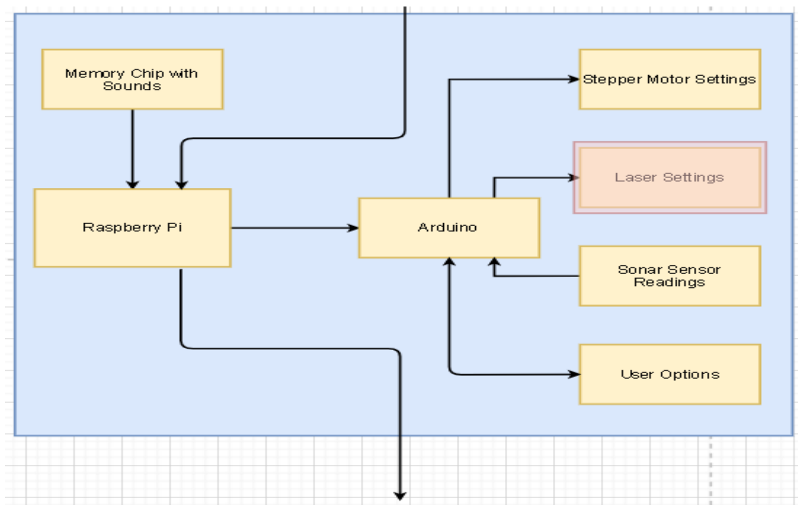

Figure 6: Laser Subsystem

4.5.1 Subsystem Hardware

Laser emitter

4.5.2 Subsystem Operating System

No Operating System required.

4.5.3 Subsystem Software Dependencies

None

4.5.4 Subsystem Programming Languages

C++

4.5.5 Subsystem Data Structures

A C++ class initializes the laser emitter intensity when the device starts up. This value will be constant and will not be user accessible.

4.5.6 Subsystem Data Processing

None

4.6 Sonar Sensor Subsystem

The sonar sensor subsystem will consist of 6 sonar sensors connected to the arduino. The song selection algorithm will choose the song based on the sonar sensor reading. Each sonar sensor will be assigned two 'virtual strings' that represent the strings of the harp. These will be separated at an angle of 5 degrees.

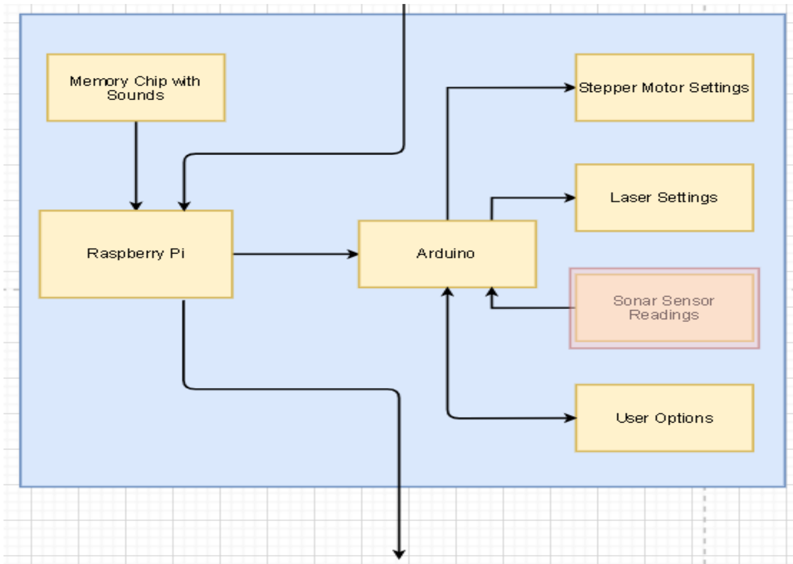

Figure 7: Sonar Subsystem

4.6.1 Subsystem Hardware

Ultrasonic Sensor HC-SR04

4.6.2 Subsystem Operating System

No Operating System required.

4.6.3 Subsystem Software Dependencies

None

4.6.4 Subsystem Programming Languages

C++

4.6.5 Subsystem Data Structures

This subsystem will contain the sound selection algorithm. Each of the 6 sonar sensors will be assigned an ID (1-6). Each ID will contain two sections. So the 12 strings will be represented from 1A - 6B. Depending on the angle of intersection between the user's hand and the sonar, the system will select the sound to played.

4.6.6 Subsystem Data Processing

All data processing is being internally handled by the Ultrasonic Sensor HC-SR04. The sensor returns an integer value of the distance of the object from the sonar.

5 Output Layer Subsystems

The laser harp output layer subsystems are tasked with communicating the device settings as well as the sound produced from the players interaction with the harp. There are two main subsystems; the display processor, and the audio processor. It will also output midi for controlling external synthesizers.

5.1 Layer Hardware

This is one of the most hardware intensive layers in the device and uses LCD screens as well as USB ports, 1/4" instrument cable ports, and speakers to output the information. It also uses an additional PCB, the music maker shield, to construct the audio which will be broadcast to the speakers and instrument cable port.

5.2 Layer Operating System

There is no Operating System at this level.

5.3 Layer Software Dependencies

LiquidCrystal_I2C.h

5.4 Display Processor

The Display Processor is in charge of powering and keeping the LCD screen up-to-date. The LCD screen will display which voice is currently selected. This will be a routine that runs on the Arduino Mega. This routine will monitor changes in the voice index global value and will reflect that state on the screen.

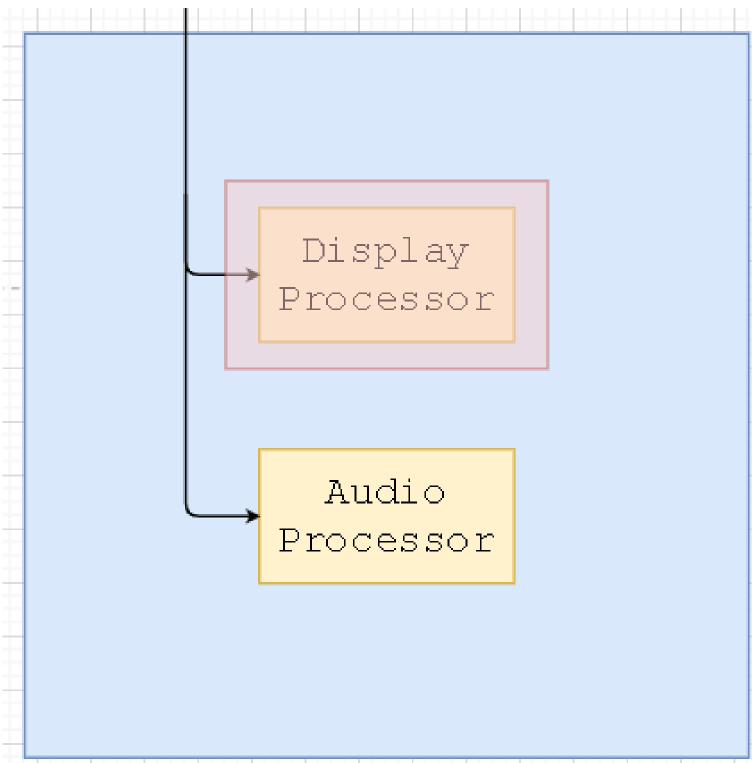

Figure 8: Example subsystem description diagram

5.4.1 Subsystem Hardware

I2C 1602 Serial LCD screen

5.4.2 Subsystem Operating System

N/A

5.4.3 Subsystem Software Dependencies

LiquidCrystal_I2C.h

5.4.4 Subsystem Programming Languages

C/C++

5.4.5 Subsystem Data Structures

This system will have a lookup table for the names of the available voices and a pointer to that voice on the music shield. Separately there will be a voice index pointer to the selected voice in the lookup table.

5.4.6 Subsystem Data Processing

N/A

5.5 Audio Processor

The Audio Processor will be tasked with producing the audio signal and midi signal. This system will require dedicated hardware for packaging the midi signal to send over USB and for generating and amplifying the audio signal.

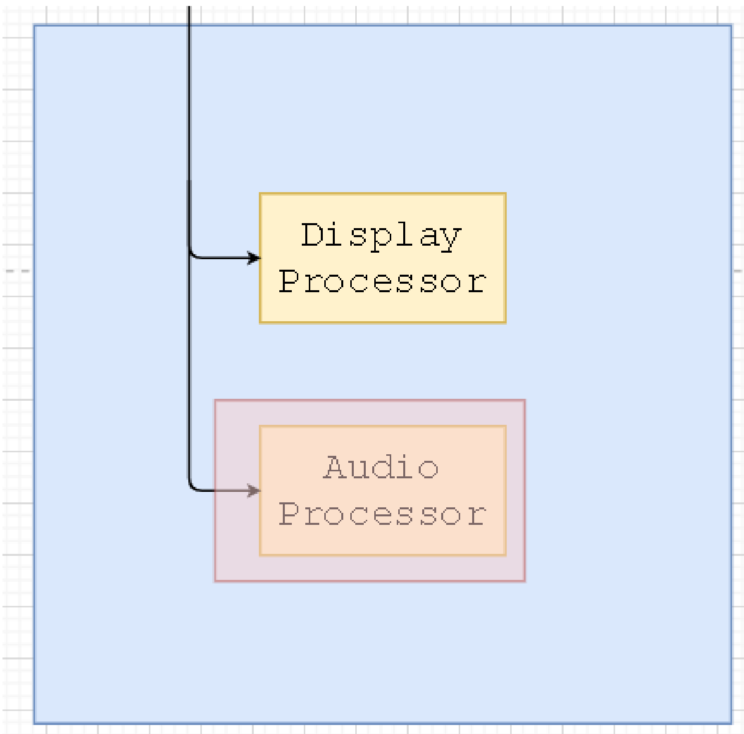

Figure 9: Example subsystem description diagram

5.5.1 Subsystem Hardware

MIDI to USB Cable Converter Adafruit Music Maker Shield MAX9744 20W Amplifier 2 x 20W 4 Ohm Speakers

5.5.2 Subsystem Operating System

N/A

5.5.3 Subsystem Software Dependencies

Adafruit_VS1053.h

5.5.4 Subsystem Programming Languages

C/C++

5.5.5 Subsystem Data Structures

The audio processor will be passed a packet. This packet will encapsulate: the index of the broken string, the index of the octave the string was broken at, and the volume.

5.5.6 Subsystem Data Processing

The packet will be used to construct a MIDI packet that will be sent simultaneously to the MIDI to USB cable converter and the Arduino Music Maker Shield.