Automated Volume Adjusting Bluetooth Speakers

Veronica Tu

ECE-498: Capstone Design Project

Advisor: Professor Hedrick

November 26, 2019

Acknowledgements

Thanks to Professor Hedrick for advising me on this project. My sincere gratitude would not be enough to express how thankful I am for your teaching and support during these past terms. Thank you for helping me through and guiding my research of this project, as well as providing the necessary components and equipment.

Thanks to Xavier Theo Quinn for helping me brainstorm ideas for project topics and advising me on ideas for this project throughout these past terms.

Table of Contents

List of Figures

List of Tables

Introduction

With the rise of automation and wireless technology, there are so many different possibilities to create potential future technology with these fundamentals. I want to utilize my curiosity and interest in automation and wireless signals, along with my background in audio and music, to create a device that will automatically adjust volume according to received signal strength using Bluetooth by applying my skills and knowledge from my undergraduate education in electrical engineering, as well as nontechnical skills from my liberal arts education.

Bluetooth has become a big part of modern technology, allowing wireless connection between devices. This provides a convenient connection between devices without having to plug in a cord. Anyone with a Bluetooth-enabled device can connect to another device, so long as one or both devices aren't already connected to another device. With wireless technology becoming the norm in modern technology, I have yet to explore the different uses I can have with Bluetooth and it's functionality. Some typical examples of Bluetooth include connecting a phone to a wireless headset to take a call while driving on the highway. Another example is when connecting a mobile device to wireless speakers via Bluetooth so that the user can still carry the mobile device around an area and still use the speaker to play music. Though when the user walks around into different rooms or just further away from the speakers, the sound quality and volume is subject to changes. In response, a Bluetooth speaker that improves the auditory experience by optimizing volume for indeterminate listener locations is one solution.

One problem with speakers is that it usually has a fixed volume unless you change it manually via remote or connected device. And generally, the sound is heard at a lower volume the further one is from the sound source. In the situation where the user is walking around an area while playing music on the speakers, the volume will vary depending on where the user is, and the sound quality is probably changing if the user is walking into different rooms. Bluetooth is generally used in short ranges, so if the user goes beyond the radius of the Bluetooth range, the device is no longer connected to the fixed device. Some problems that would be addressed are how to collect and/or measure the distance and position of the speakers and the connected device in real time, as well as automatically controlling the volume using that data.

The goal of this project is to build a Bluetooth-enabled speaker that will use the signal strength of the connected device to the speaker to control the volume of the speaker. The overall resulting goal is having the user listen to the audio at the same volume regardless of the distance from the speaker. Distance and location detection will not be necessary due to the usage of the RSSi value, or the signal strength. So, using the received signal strength, this will determine how far the mobile device is from the speakers. This report will be covering more in depth background information about the inner workings of the project, design requirements of the final product, design alternative for different aspects of the project, the design, some preliminary testing results found over the course of the term, and an implementation schedule for the next term.

Background

Previous Work

Researchers and engineers have been experimenting with the limits of Bluetooth technology. With much of these articles dating after 2010, these are recent developments. After deciding on my project goal, I researched how to make it happen. Here are some articles of research that has been done exploring the use of Bluetooth received signal strength indicator.

Gowda (Gowda, 2012), of the University of Utah, performed experiments measuring the RSSI values using Bluetooth interface for secret key extraction. Due to the degradation of Wi-Fi, Gowda proposes that Bluetooth would be more efficient in it's adaptive frequency hopping technique of detecting and avoiding bad signals and interferences. Gowda needed to design a protocol to send and received L2CAP packets between Bluetooth-connected devices so that the RSSI value can be measured from each packet. This thesis was published in 2012, so the methods of measuring the RSSI value were different then. The experiment was also performed in both outside and inside environments to understand the difference of the efficiency of Bluetooth, in which the outdoor setting was more efficient in the resulting secret bit rates than the indoor setting. The thesis concluded that the secret bit rates received with the Bluetooth and Wi-Fi were similar in similar settings and conditions.

Rozum and Sebesta (Rozum & Sebesta, 2019) used the Bluetooth RSSI value as an input for location estimation in Bluetooth Low Energy Indoor Positioning System. The RSSI value was used as a reference for location and the fading method was used to reduce instability in the value so that the reference remains as accurate as possible. By using four antennas to measure difference distances, the back-side reflector was used to obtain the quarter wavelength additional arrangement. They determined that, though with the reflector the antenna gain was greater, without the reflector reception was subpar. So, if the target's location had to be found or estimated through walls, the received signal could not be accurate regardless of how precise the calibration is.

Subhan, Hasbullah, Rozyyev, and Bakhsh (Subhan, Hasbullah, Rozyyev, & Bakhsh, 2011) presented a Bluetooth handover concept that measures and uses the Bluetooth RSSI, Link Quality, Transmitted Power Link, and Received Power values as parameters to estimate location, such as the previous research. Their research includes a lot of information on the date rate, range, and frequency of the Bluetooth network, as well as understanding the specifications within Bluetooth protocols. For their research, due to the issue of handover in Bluetooth, Bluetooth RSSI, Link Quality, Transmitted power Link, and Received Power were used as variables to test different techniques to using Bluetooth in positioning. They concluded that Received Power could be used for distance estimation and indoor positioning.

With the conclusions collected from these research findings, I can gather the necessary information to understand the theory behind my project and how to use these experiences to make sure my project performs as accurately as possible. For example, Gowda's thesis in using Bluetooth instead of Wi-Fi will help me understand the usage of the Bluetooth RSSI value and the fluctuations it has in an indoor setting. While Rozum and Sebesta gives proof that position estimation with RSSI is possible and will need some work into keeping the accuracy through walls. And Subhan, Hasbullah, Rozyyev, and Bakhsh provide vital information in understanding different parameters within a Bluetooth signal that can be used to measure distance.

Impacting Considerations

With these articles in mind, further research was required to understand the limitations of Bluetooth and how to manipulate the thresholds to develop the design requirements for this project. Such as the different classes of Bluetooth and the power at which the device transmits at a maximum range, as shown in Table 1. This is so that the project should be able to have enough power to transmit signals over a large enough range.

Table 1: Bluetooth Power Classes

Device Class	Transmit Power	Intended Range
Class 3	1 mW	Less than 10 meters
Class 2	2.5 mW	10 meters
Class 1	100 mW	100 meters

Another important consideration is how the Bluetooth RSSI value changes according to distance. Though the device is not actually calculating the distance value, the RSSI value is the main input parameter of the project and does change according to the distance, as shown in Figure 1. The RSSI value in relation to distance can be found using this equation:

$$RSSI = -(10*n*\log10(d) + A)$$

Where *RSSI* is the received RSSI value, *n* is the path-loss exponent, *d* is the distance, and *A* is the RSSI value at the initial distance set (Lau & Chung, 2007). In terms of this project, the setting will be indoor residential area with a path-loss exponent value of about 2.8. The reference value is generally taken as the RSSI value received when the Bluetooth receiver is 1 meter from the transmitter. This will be taken into account during theoretical calculations and testing.

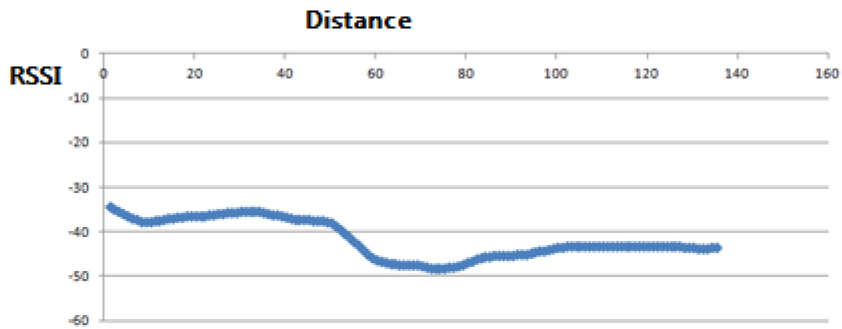

Figure 1: Change in RSSI value by Distance

Other considerations outside of the contents of the articles include usability, marketability, manufacturability, etc. This project could potentially be a marketable product in terms of the setting that it's intended to be used. The project would need to be as hands off as possible to warrant the claim that it's automatic, but accessible in the manual inputs to allow the sound to be customizable. The project should also be compliant to ethical codes and standards, as well as standard policies and regulations to ensure the safety of the user. These considerations would be applied to the range of sound that the product would allow due to the decibel maximum before sound starts to damage hearing. Using these considerations, I compiled a detailed set of design requirements that will serve as a basis for the project.

Design Requirements

The Automated Volume Adjusting Bluetooth Speaker in a broad sense, is meant to connect to a mobile device via Bluetooth and adjust the volume so that the user will listen to the audio output at a constant volume as if the user was standing next to stationary speakers. So the device will receive an input of the RSSI value with the code, which will in turn plug it into an equation that will proportionally change the volume value. This volume value will affect the audio amplifier to change the volume of the audio output of the speakers. This volume change will maintain the perceived auditory volume of the user at a constant level.

The intended setting of this device will be between a 5.5 x 7.9-meter room and a 5.9 x 12.5-meter apartment. The location where I have tested this device was in a 5.5 x 7.9-meter room, while the location where I originally intended to use this device was in a 5.9 x 12.5-meter apartment. The size of the room needed to be large enough to test the change in decibels from the stationary speakers. The apartment emulates a typical apartment with three rooms, one bathroom, and an open space with kitchen and living room. With this setting, testing can be done through multiple rooms and spaces for sound quality and volume adjustment with walls.

The Bluetooth range that the device must work within 23 to 31 meters due to the path-loss in a typical home (What is the range of Bluetooth?, 2019). With the Raspberry Pi Zero W, using BLE 4.1, which achieves an operating range of over 30 meters, this is compliant to the range specification. This ensures that the Bluetooth connection will stay strong and consistent within the intended setting, while having the potential of continuing service outside of the setting.

The initial calibration of the device must be taken 1 meter from the transmitter in order to obtain the received RSSI value at a fixed distance from the speaker. So the user must connect the mobile device to the transmitter via Bluetooth and set the volume at the desired loudness 1 meter from the transmitter. Then the device will use the RSSI value from that distance as a reference RSSI value to keep the desired loudness proportional to the change in RSSI value. This reference RSSI value is subject to reset when the device is shut down so the user will need to recalibrate each time when powering on the device.

The proportion at which the volume value changes in relation to the RSSI value will be logarithmic. During testing the change in decibel readings was logarithmic in relation to distance when the volume was constant, and the distance changed. So, in response to the data, the volume value would change logarithmically according to the change in RSSI value.

Wall detection is necessary in the code due to the intended setting and will be detected with the slight drop or increase in RSSI value when the signal passes through a wall. The typical path-loss exponent in an indoor setting with walls is 2.8, so that will be taken into account within the code. This will allow for the volume value to be boosted or diminished to accommodate the soundproof of the wall or lack of.

The device will take into consideration the sound safety limits for hearing. Due to noise danger, there is a decibel limit before the sound can cause hearing loss. Typically, people listen to music and audio from a speaker at around 70 dBA or less (Loud Noise Dangers, 2019). But hearing loss can be caused by listening at 85 dBA for long periods of time and damage accelerates as the noise level increases. So the device will stop increasing the volume once it reaches 80 dB regardless of the increase in RSSI value from that point.

The expected market price of the device is at least $25 but a manufacturing price of no more than $40. This is due to the use of the Raspberry Pi Zero W, which has a market price of $10, along with power cords and miscellaneous wires and connecters. The speakers and audio amplifiers will be connected to the Raspberry Pi Zero W with equipment resembling an L-pad. This way, the device can be fairly cheap to manufacture and depending on the quality of the speakers, can be marketed at a reasonable price.

The weight of the device itself should be less than a pound, including the casing. This is due to the number of components that would be needed to be connected to the speakers. The size of the device should be no larger than a 10 x 10 x 10 cm cube, the small size is to save space, using only needed space for the device components. The casing should also be small to add aesthetic to the entire sound system, with the use of large speakers to emphasize the amount of power needed to project the sound as needed by the user.

This device will have to comply with core Bluetooth standards. Such standards include IEEE 802.15.1 (IEEE 802.15 WPAN Task Group 1 (TG1), 2019), which includes clauses on SAPs, normative annex providing a Protocol Implementation Conformance Statement proforma, and informative high level behavioral ITU-T Z.100 specification and description language model for integrated Bluetooth MAC sublayer. Since Bluetooth is a short range RF based connectivity for portable devices, this standard provides an adaptation of the Bluetooth Specification v1.1 Foundation MAC.

Design Alternatives

This system consists of two main components that are highlighted in the block diagram in Figure 2. One such component is the input parameter used to measure the distance of the receiver from the transmitter. There are many different alternatives to measuring the physical distance that may require a sensor and others that require receiving a signal. Some examples of each that were considered will be discussed. Another component is the method of physically controlling the volume. This takes the type of speaker and control panel the device will function with into consideration. The following sections describe the purpose, multiple potential design solutions, and the advantages and disadvantages of each section.

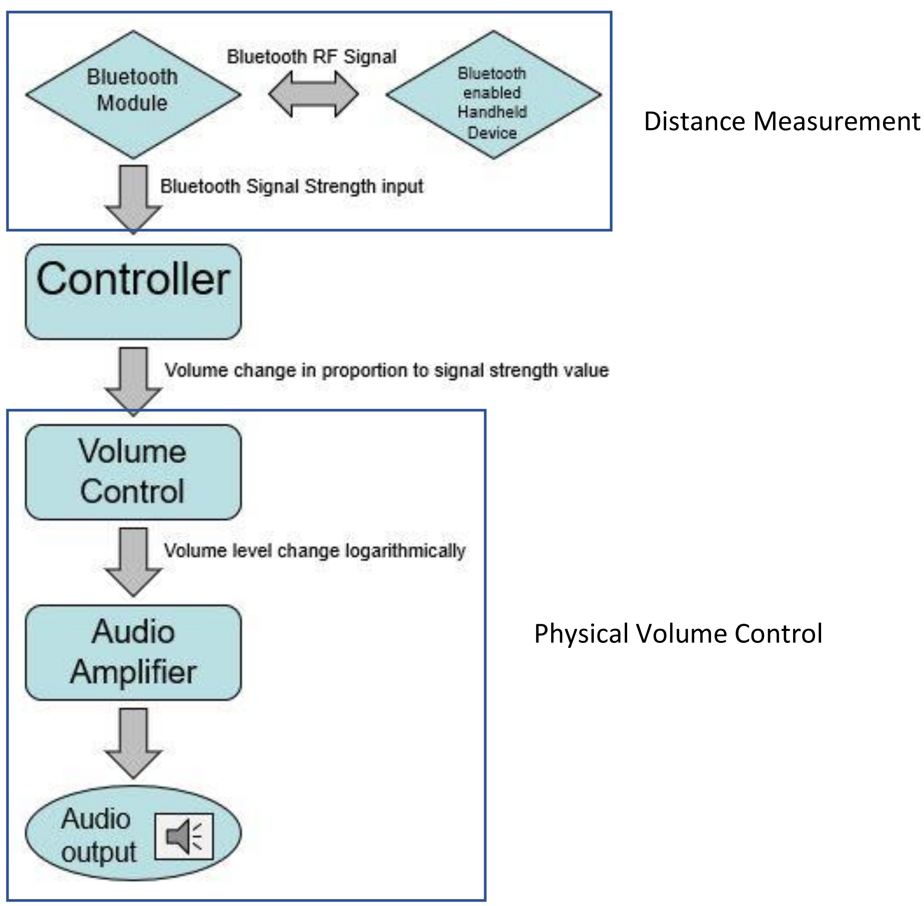

Figure 2: Overall Block Diagram

Distance Measurement

There are many ways of measuring distance, one such way is the use of sensors. Though there are many types of sensors, they measure physical attributes to measure the distance from the sensor.

Echolocation

Echolocation was the most basic form of distance detection using high frequencies, though would be ineffective with a moving target and obstacles.

LED

LED sensors are inexpensive with multiple interface options and good update rate, but it would have been power consuming with low range.

LIDAR

LIDAR sensors have a large range and fast update rate but is also very power consuming and expensive.

Ultrasonic

Ultrasonic sensors have low power consumption and multiple interface options but would have a slow update rate and low resolution.

VCSEL

VCSEL sensors have a small minimum range, large input voltage range, good resolution and is inexpensive, but there is only one interface with a low maximum range (Distance Sensing Overview, 2019).

Bluetooth

With Bluetooth connection, there a several parameters that could be taken from the Bluetooth signal alone that could estimate distance without a distance value input. In Subhan, Hasbullah, Rozyyev, and Bakhsh's research (Subhan, Hasbullah, Rozyyev, & Bakhsh, 2011), they tested different signal parameters: Received Signal Strength Indicator (RSSI), Link Quality (LQ), Transmitted Power Link (TPL) and Received Power (RX). The Link Quality and Transmitted Power Link parameters are device and manufacturer specific so the input value would have to be specific to the components used in the project. To allow for component flexibility, these parameters will not be used as input. Received Power is indirectly related to the received RSSI value and would need to be converted using the radio propagation model. But the RSSI value is directly proportional to the distance of the receiver from the transmitter, and with a low update rate, the RSSI value can be received as close to real time as possible.

Physical Volume Control

Methods for volume control is dependent on the type of speaker used during the project. One considered method of volume control is another Bluetooth connection from the device to the speakers, but multiple Bluetooth connections to a single device may result in complications. There is also no physical manual override system for the user in a case of need. The desired form of volume control would be a direct connection between the controller to the audio amplifiers and speakers by internal wires or hardware. This can potentially be done with an L-pad, as shown in Figure 3. But the L-Pad has a physical dial, much like a potentiometer, so that device may have to include a motor controller and motor to attach to the dial. Adding a motor would require research in how much torque is needed to accurate turn the dial on the L-Pad. This would also be the case if the speakers had physical dials built into the system to control the volume. But seeing as a manual control override system is necessary for usability, having a physical control panel system would be needed.

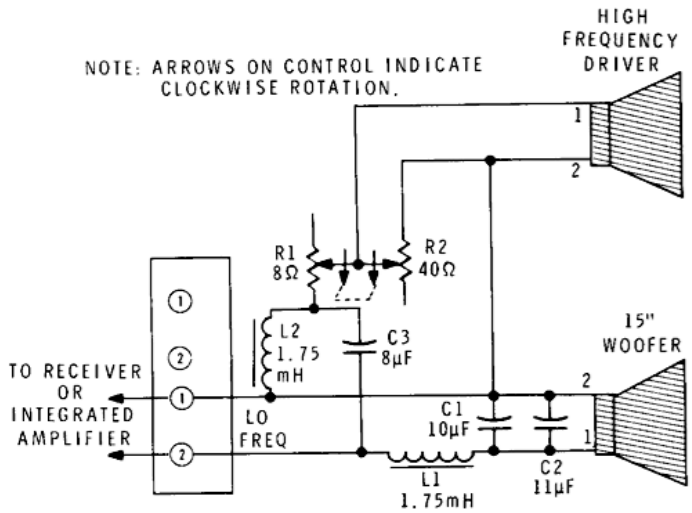

Figure 3: Schematic of Wiring for L-Pad

Design

First Level – Bluetooth input

According to the general block diagram in Figure 2, the first level of the Automatic Volume Adjusting Bluetooth Speaker is the Bluetooth aspect of the system. This is the connection between the controller, the Raspberry Pi Zero W, and a mobile device, most likely a phone with Bluetooth functionality. The Raspberry Pi Zero W uses Bluetooth 4.1/BLE and follows standard Bluetooth protocols, such as using a 2.4 GHz band for Bluetooth and Wi-Fi communication. It also includes the IEEE 802.11n specification, which entails that the Raspberry Pi Zero W can operate at a maximum throughput of at least 100 Mb/s, measured at the MAC data service access point (IEEE 802.11n-2009, 2009). Once the Raspberry Pi Zero W has a Bluetooth connection to the phone, the Pi Zero can use the signal to obtain the RSSI value.

Second Level – Controller and Code

The second level of the device is the software aspect of the system, which relies on functions within the code to calculate the volume value from the RSSI value. Though the full detailed calculation for the volume value in proportion to the RSSI is in progress, below in figure 4 is the generalized pseudocode for the system. There are packages that can be downloaded to Linux to enable Bluetooth capabilities in the Raspberry Pi Zero W. Once the Pi Zero is connected to the phone. There are functions to obtain the RSSI value from the Bluetooth signal. So starting the main program, to calibrate the initial system settings, one specification is for the user to connect to the system from a distance of 1 meter from the device. Once connected, the program will assume that the device is connected with a distance of 1 meter and use the RSSI function to obtain and store the RSSI value from that distance. The user will set the volume of the phone manually according to the volume the user would like as the constant perceived volume. The control function will perform real time calculation and output the volume value according the RSSI input. The control function consists of a loop that will take the input RSSI value, taking into account if there is significant jump or drop in input stream as wall detection and updating the loss value accordingly. Within the control function is the volume function, which includes the actual calculation of the volume value using the real time RSSI value and loss value adjustment. There will be a maximum volume limit for safety which will contain the output decibel level of the system within safety auditory range to prevent hearing damage. So if the calculated volume value is greater than the maximum volume limit, the maximum volume limit will be the output value to the volume control, otherwise the calculated volume value is outputted. Bluetooth signal disconnection will also be taken into account. If the connection to the mobile device is disconnected, the system will set a delay timer where if the phone is reconnected within the delay time range, the program will continue the control function loop with the same calibrations. Otherwise, if the phone is not reconnected by the end of the delay time range, the system will restart the calibration and remain idle and set another delay timer. If the phone is reconnected during that delay time range, the system has already reset and will need to be recalibrated, but if the phone remained unconnected, the system will shut down after the delay time range.

```
Main() - main program
    Connect to mobile device with Bluetooth
    Calibrate() - initial calibration of RSSI value and volume setting
        RSSI() - Function to obtain RSSI value
        Initial RSSI value = RSSI value at 1 meter from transmitter
        Distance = 1 meter

    Control() - real time calculation of volume value using the RSSI value for volume control
        Loop
            RSSI()
            If: RSSI value has a significant jump/drop
                Then: Add/subtract loss value
            Else: nothing
            Volume()
                Function that takes the RSSI value and loss value to calculate the volume value
            If: volume value is greater than max volume limit
                Then: Output max volume limit
            Else: Output volume value to volume control
            Delay
        End

    If: connection to mobile device is disconnected
        Delay
        If: reconnected
            Then: Control()
        Else: restart system
            Delay
            If: reconnected
                Then: Main()
            Else: System shut down
End
```

Figure 4: General Pseudocode

Third Level – Physical System and Volume Control

The third level of the project is the physical components of the system, which is shown in Figure 5. The Raspberry Pi Zero W is connected to the phone wirelessly using Bluetooth signal and is connected to the volume control system. The pin connections of the Pi Zero is found in Figure 6, which includes power connections and output pins. There are input ports for a micro SD card, which this system will be using a 32 GB SD card, and power supply, with a 5.1 V micro USB switching AC power supply. The volume control system will consist of a motor controller connected to a motor that will attach to the L-Pad, which is connected to the amplifier, which is also yet to be decided, and the speaker. This volume control system would either physically change the volume dial or send signals directly to the amplifier to change the audio output of the speakers. The full components list is found in Table 2, this list will be updated over time as official components are selected.

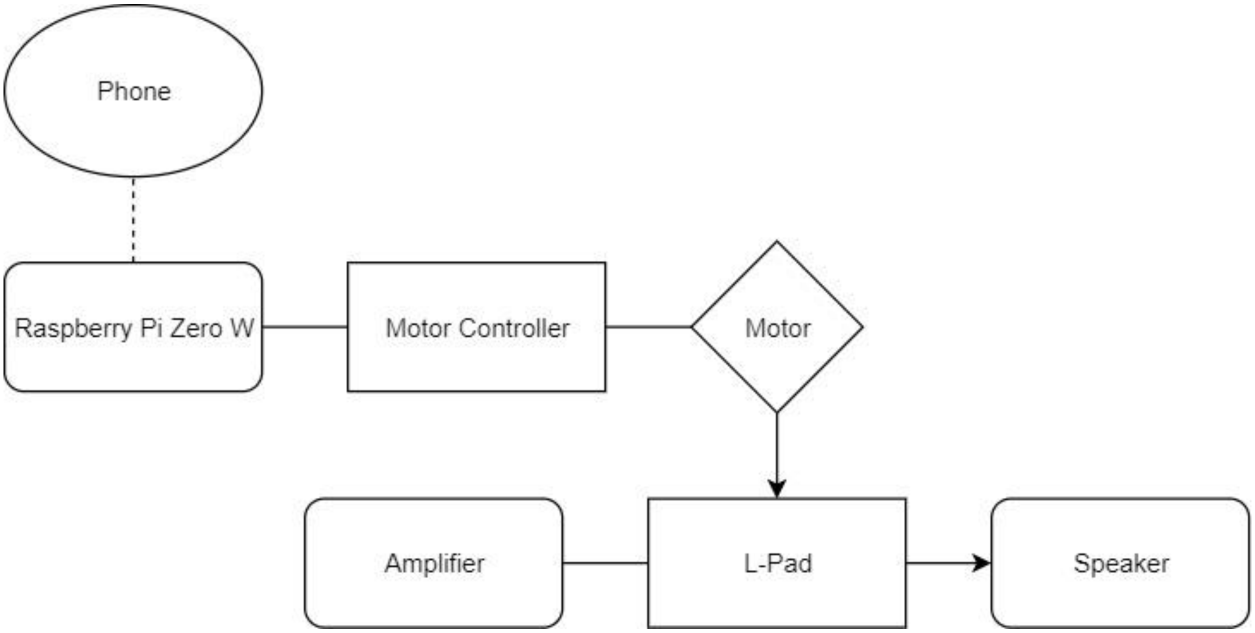

Figure 5: General Physical Block Diagram

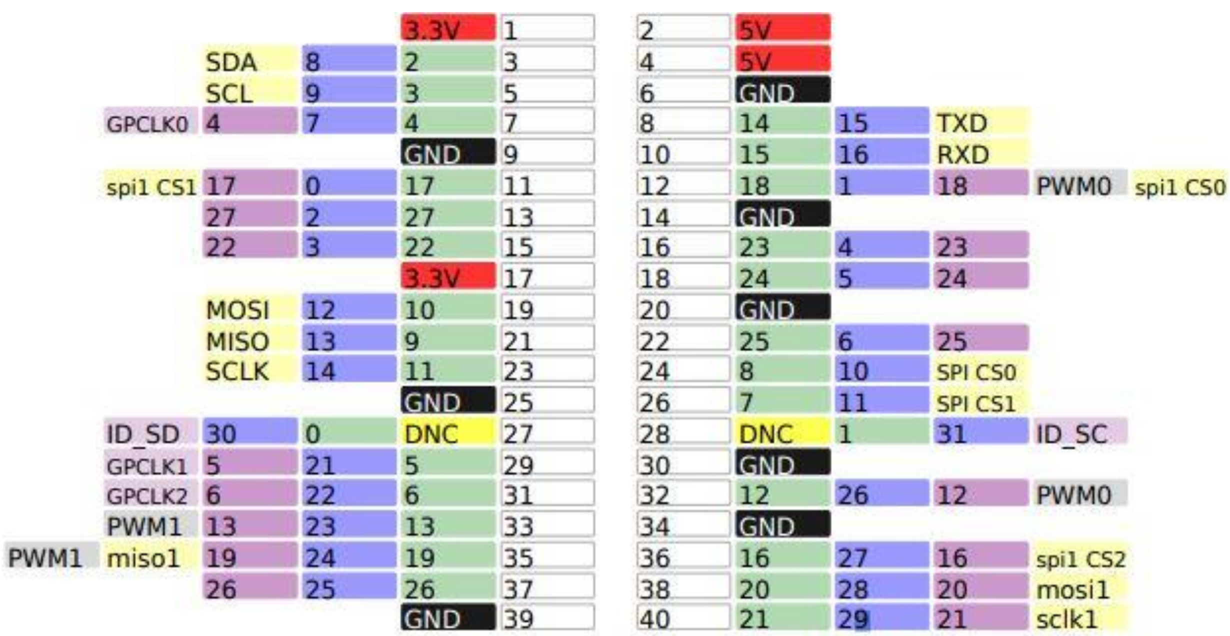

Figure 6: Raspberry Pi Zero W Pin List

Table 2: Parts List

Component	Quantity
Raspberry Pi Zero W	1
Amplifier	2
Speaker	2
Micro SD Card – 32 GB	1
Pro-Elec 5.1 V/2.5 A Micro USB Switching AC Power Supply	1
L-Pad	1
Motor	1
Motor Controller	1
Miscellaneous wires	≥ 8

Final Level – System Output Goal

The final level of the system is the audio output of the speakers or the resulting volume level output from the RSSI input value in real time. The perceived auditory output of the system should be constant to the volume set by the user during calibration. With the detection of a wall, the volume should boost slightly to adjust to the sound proofing of the wall. Then detection of straight line space would decrease the volume slightly with the lack of wall.

Preliminary Testing Results

In the process of understanding how sound level changes in respect to distance to create an equation which is accurately proportionate to the volume output with the RSSI value, I did some tests using a decibel meter and speakers. When using the decibel meter, there are two different rating options, A and C. The A rating is used for auditory sound measurements while the C rating is used for sound pressure. So for these tests I used the A rating on the decibel meter because the decibel levels change logarithmically and peak above the auditory threshold, or 0 dB. I tested the physical sound levels of the speakers by changing the volume using the volume control dial, with low, medium, and high on the dial. With each volume setting, I checked the decibel reading starting one meter away from the speakers and moving back 2.5 meters away from the speakers at a time. The results showed that the change in decibels with respect to distance in the low volume setting is greater than the high volume setting, but overall the change is logarithmic. In terms of comfort levels, at 8 feet from the speakers, the sound is uncomfortable at 75 dB and greater, which is outside of the sound safety limit of 0 – 75 dB. This will be essential to creating the volume function, shown in the pseudocode in Figure 4.

Implementation Schedule

Table 3 below shows a general comprehensive schedule for the completion and testing of the project.

Table 3: Implementation Schedule

Goal	Due Date
Learn Linux for Raspberry Pi Zero W	12/13/19
Write pseudocode for system	12/27/19
Create schematic of physical system	1/3/20
Order/obtain remaining components/equipment	1/10/20
Build and set up physical system	1/17/20
Write and implement code in the Raspberry Pi Zero W	1/24/20
Test system for Bluetooth RSSI value and adjustment of code as needed	1/31/20
Test system for volume value change according to input RSSI value and adjustment of code as needed	2/7/20
Test system for volume control accuracy according to the volume value	2/14/20
Test system with user for perceived hearing	2/21/20
Final test in an apartment setting	2/28/20

References

Distance Sensing Overview. (2019). Retrieved from Sparkfun: https://www.sparkfun.com/distance_sensing

Gowda, P. L. (2012). *Exploring Bluetooth for received signal strength indicator-based secret key extraction.* School of Computing. Salt Lake City: Univ. Utah. Retrieved from https://pdfs.semanticscholar.org/4a85/489e4f075f2155864177a54b630060850b91.pdf

IEEE 802.11n-2009. (2009). Retrieved from IEEE Standards Association: https://standards.ieee.org/standard/802_11n-2009.html

IEEE 802.15 WPAN Task Group 1 (TG1). (2019). Retrieved from IEEE802.15: http://www.ieee802.org/15/pub/TG1.html

Lau, E.-E.-L., & Chung, W.-Y. (2007). Enhanced RSSI-Based Real-Time User Location Tracking System for Indoor and Outdoor Environment. *2007 International Conference on Convergence Information Technology*, (pp. 1213-1218). Gyeongju. doi:10.1109/ICCIT.2007.253

Loud Noise Dangers. (2019). Retrieved from ASHA American Speech-Language-Hearing Association: https://www.asha.org/public/hearing/Loud-Noise-Dangers/

Rozum, S., & Sebesta, J. (2019). Bluetooth Low Power Portable Indoor Positioning System Using SIMO Approach. *Telecommunications and Signal Processing (TSP) 2019 42nd International Conference*, (pp. 228-231). doi:10.1109/RADIOELEK.2018.8376391

Subhan, F., Hasbullah, H., Rozyyev, A., & Bakhsh, S. T. (2011). Handover in Bluetooth Networks using Signal Parameters. *Information Technology Journal, 10*, 965-973. doi:http://dx.doi.org/10.3923/itj.2011.965.973

What is the range of Bluetooth? (2019). Retrieved from Bluetooth: https://www.bluetooth.com/bluetooth-technology/range/