ACOUSTIC MOTION TRACKING

ECE 445 - DESIGN REVIEW

TEAM 35

SEAN NACHNANI

HOJIN CHUN

TA

YUCHEN HE

Table of Contents

1. Introduction

1.1 Objectives

People are always striving for a better life and are trying to make each day more comfortable than the last. As technology advances further, and AI comes more into focus, we try to satisfy the comfort needs by building autonomous cars, creating smart cities, and building robots that can eventually replace humans. Even now, we have products like Alexa, Siri, Bixbi, and many more that can understand a person's vocal commands and get data or perform tasks for the user, without them having to lift a finger.

Our objective is to build a device that use sound rather than video as a means of motion recognition. Current devices mentioned above are limited to only using natural language processing to interpret a user's need. We want to expand upon this further and allow devices to perform commands using simple gestures. Not only our device will make people's life more comfortable with this feature, but also will help people with speech impediments or certain accents, allowing them to still use these devices.

Our implementation will have 4-input microphone array that allows for at least a 48 Khz sample rate, and use a speaker that can reproduce sounds up to 24 kHz. We will use a microcontroller to act as the bridge between all the hardware components and the software running on external device. Our device will be designed to be plugged into the a regular power outlet. Our current software implementation involves pulsing a pseudo-random wave, and calculating the distance of the hand by measuring the peaks of the Channel Impulse responses received from the microphone array. If we are able to finish the project ahead of schedule, we plan to go further and implement FMCW (Frequency Modulated Continuous Waveform) radar as a basis for this approach. This will allow us to also take advantage of Doppler effects and calculate velocity, further increasing the accuracy of the device.

1.2 Background

Motion tracking has always been widely researched, but the majority of it centered around video, or radar. Now, with the increasing number of mobile devices that contain microphones and speakers, research has been drawn towards the acoustic region. A lot of this research has been been focused around VR[1] and smartphones[2] , but we intend to implement this with the smart assistants that are becoming ever more prevalent. Natural Language Processing has come very far and allowed these devices to understand human speech. However, they still struggle when dealing with speech impediments and accents. Therefore, implementing gesture control for these devices will help solve these issues. Moreover, this project will open the door to other motion recognition uses for the smart devices, such as the ability to act as a motion alarm if the user isn't home, and also as possible forms of authentication to counter acoustic DDoS attacks[3].

1.3 High-Level Requirement Lists

- The Device must be able to wirelessly transmit the signal recorded from the microphones with a Sample Rate of 48kHz to a server (or a computer) for signal processing.
- The microphone array must be able to record frequencies up to 24kHz to allow for the speaker to transmit in the inaudible range
- The Gesture Control Algorithm must be able to accurately calculate the distance of the hand performing the gestures to within 3 cm at a distance of 2 m, to allow for proper identification of the motion.

2. Design

Block Diagram

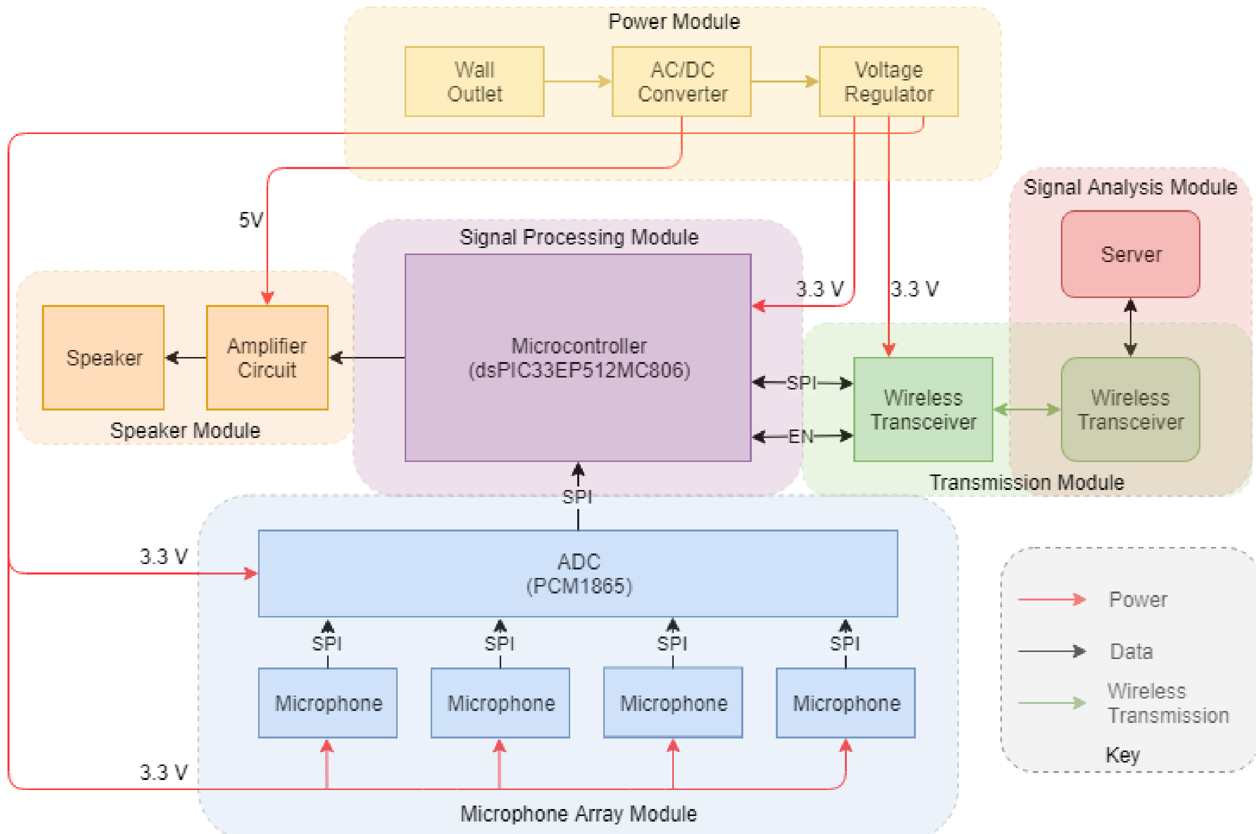

Figure 1: Component Block Diagram

Physical Design

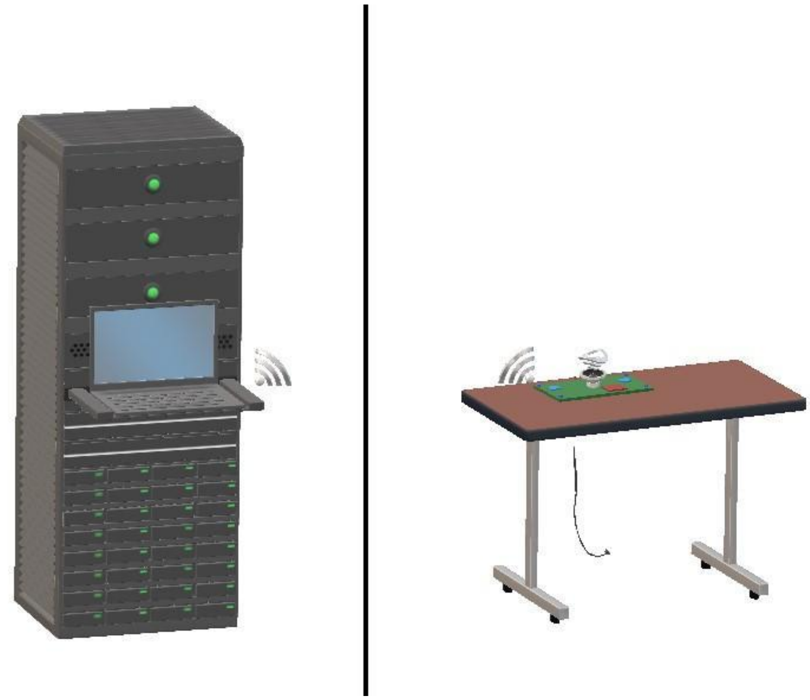

Figure 2: Physical Representation of the System

The Physical design in Figure 2 above provides a 3-D model of the system. The device itself in this model is located on the table (on the right panel) and is wirelessly transmitting to the server (on the left panel). The speaker for the device will be central to the 4-microphone array. The device will be designed such that it can be placed on any flat surface in a room and be connected to an outlet (not show). A user will be able to stand within 2m of the device to perform the gestures.

Functional Overview

2.1 Microcontroller

The microcontroller is used to sample from the microphone array module while simultaneously transmitting the sound pulses for the speaker to play. It stores the data for the full listening duration, and then transmits it to the wireless transceiver to send off to the server for signal analysis. We are using the dsPIC33EP512MC806[5] integrated DSP as our MCU. It has a low power consumption of 3.3V, 512K Bytes program memory. It has integrated DSP operations.
The link for the data sheet is: http://ww1.microchip.com/downloads/en/DeviceDoc/70616g.pdf

Microcontroller	Requirements	Verification
	1. The controller must support SPI and UART	1. Send a signal over SPI and UART and verify the output is the same as the signal transmitted
	2. The controller must have 256kb of onboard memory	Store 256kb of data and transmit it, checking to see if they are the same
	3. The controller must be able to transmit > 2Mbps	Transmit 20Mbps over SPI and verify the signal arrived in ≤10s

2.2 Microphone Array

We have chosen to use 4 microphones to create the array. We are using Knowles SPU0410LR5HQB-7[7] analog MEMS microphones. These microphones have a frequency range of 100Hz – 80kHz, and a package size of 3.76mm x 3.00mm x 1.20mm. With the addition of an ADC that has a sample rate above 48khz, we can then use sounds outside the audible range. We are therefore using one TI PCM1865[8] chip, since it has a resolution of 24bits, both SPI and I^2C, four analog inputs, and a sample rate of 48kHz. Both the microphones and ADC require an input voltage of 3.3V, the same as the MCU.
The link for the microphone is: https://www.digikey.com/product-detail/en/knowles/SPU0410LR5H-QB-7/423-1139-1-ND/2420983
The link for the ADC is: http://www.ti.com/product/PCM1864/datasheet/abstract#SLAS8312137

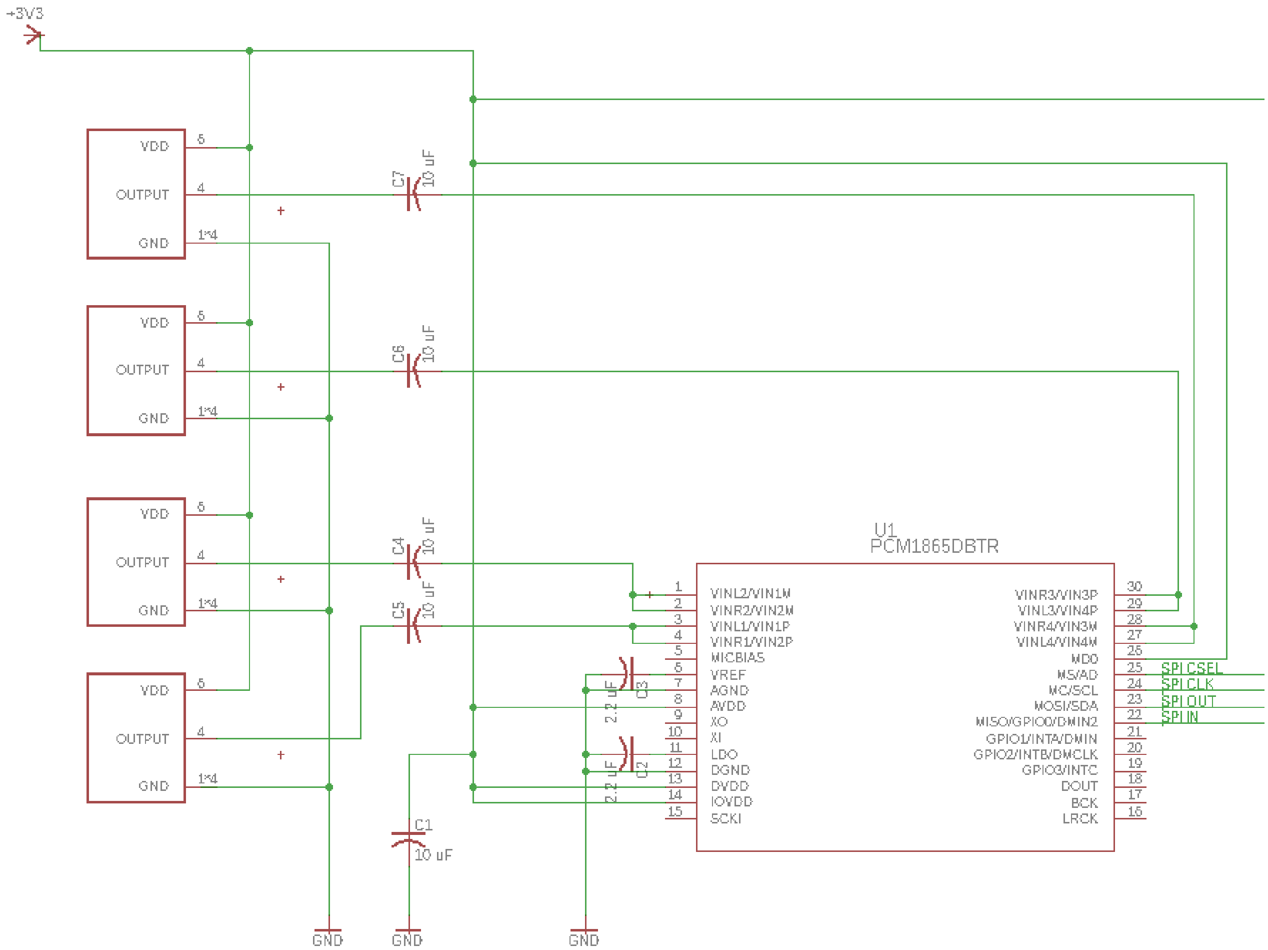

Figure 3: Microphone Array

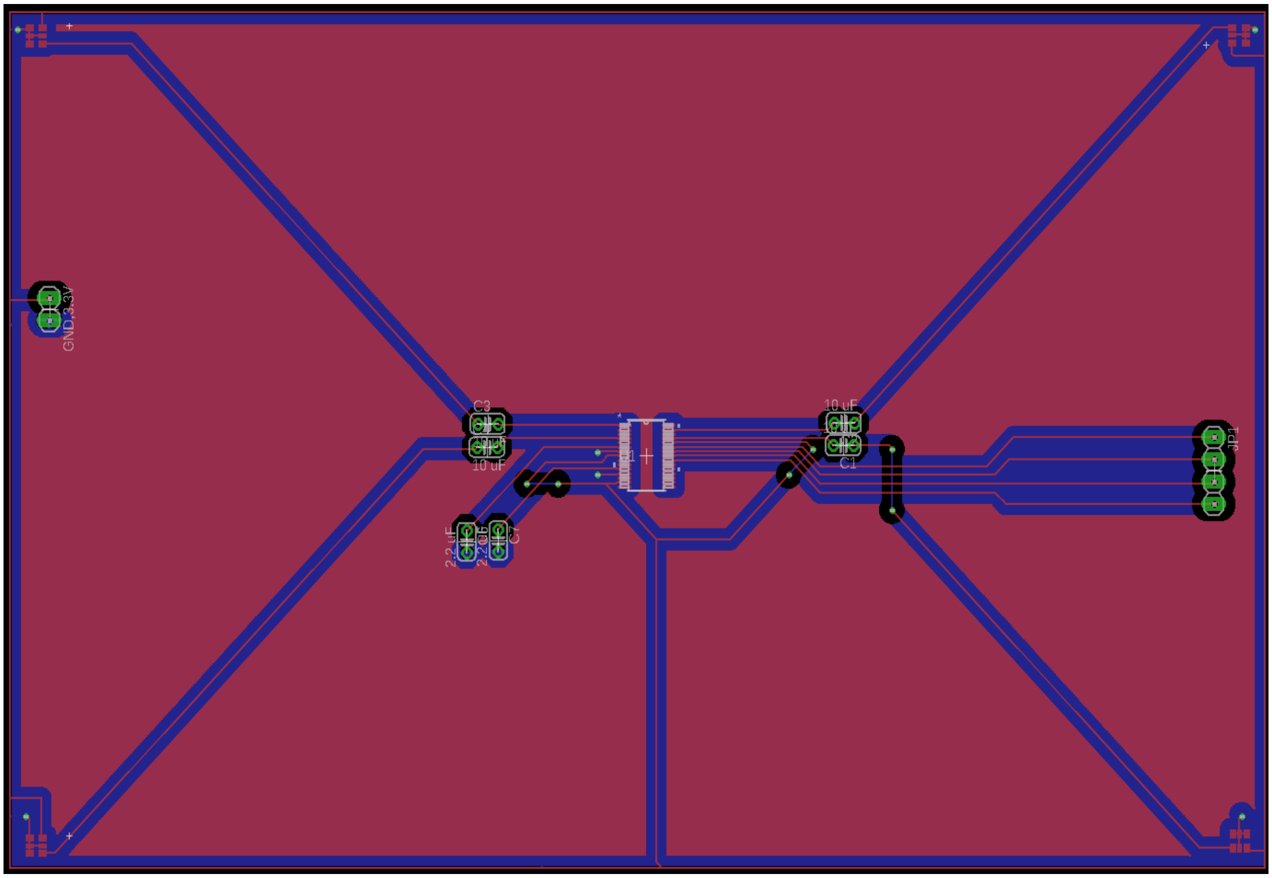

Figure 4: Preliminary Microphone Array PCB

Microphone Array	Requirements	Verification
	1. A sample rate of at least 48kHz to allow transmitting in the inaudible range	Play a sound at 24kHz and check the signal outputted from the ADC to verify it is the same
	2. The MEMS Microphones must be able to pick up frequencies of at least 24kHz	Sweep the frequency from 100Hz to 24kHz and check the output from the ADC

2.3 Speaker

The speaker needs to be able to transmit frequencies up to 24khz. For this device we are planning on using the XS-GTF1027[9] speaker by Adafruit which has a frequency response of 60Hz - 24kHz. For testing and debugging purposes, we are planning to use MAX9744[10] Audio Amplifier from Adafruit.
The link for the speaker is: https://www.adafruit.com/product/1732

Speaker	Requirements	Verification
	1. The speaker must be able to play frequencies at least up to 24kHz	Sweep the frequency up to 24kHz and record the sound played, then check if the recorded signal matches the original

2.4 Wireless Transmission

For our project the gesture recognition occurs on the server side, allowing us to implement it in software. To transmit and receive data from the microcontroller, we will be using the Microchip ATWINC1500[11]. This component has 802.11b/g/n radio protocols, allowing for standard wireless transmission. It has both SPI and UART interfaces, and supports a data rate of up to 65 Mbps on 802.11n, which is more than enough to transmit the recorded signal. The signal will have a size of no more than 2 Mb. This module also has 4Mb of flash storage, allowing us to transmit the entire sequence at once to the server. The voltage requirement ranges from 2.7 V to 3.6V, allowing it to be supplied by 3.3V just like the other components. By choosing to transmit over Wi-Fi, we will enable the device to be located anywhere, not just within a certain range of the server.
The link for the chip is: http://ww1.microchip.com/downloads/en/DeviceDoc/70005304B.pdf

Wireless Transmission	Requirements	Verification
	1. The wireless transmitter must be able to transmit over 2.4GHz channel losslessly at a rate of at least 17 Mpbs	Transmit a 2 Mb signal through the component over 2.4Ghz channel MAC protocol and verify it matches the original with a transmit time of 1 second
	2. The range must be at least 100ft	Perform the above transmission beginning next to the component then at 5ft increments until 100ft is reached

2.5 Power Supply

We will be using a wall outlet to power the device. We will need an AC to DC adapter that can convert the 120V at 60hz to DC voltage that is within the range of 3.3V – 5V for the device components. Since the different parts of the device have different voltage requirements, we will also be using a voltage regulator to allow each component to receive the correct power specifications.

2.5.1 AC/DC Converter

This converter we will be using is the RACM18-05SER[12] which will take 120 VAC from a standard wall outlet and convert it to 5 VDC. It will need to supply at least 2.4A of current to the device, and it achieves that by supplying 2.5A. This will allow our system to be plugged into any standard North American wall outlet without the need for additional voltage converters. The link is: https://www.recom-power.com/pdf/Medline_AC-DC/RACM18-ER_W.pdf

2.5.2 Voltage Regulator

For the voltage regulation, we will use LXDC55KAAA-205[13] to utilize 3.3V DC to Microcontroller, Microphones, and Wifi Transceiver a supply the correct voltages. The chips must be able to handle the peak voltage from the AC-DC converter. This chip also needs to supply enough current to all the components it is connected to and below is a table that shows the current budget calculation for this chip.
The link for the Buck Converter is:
https://www.murata.com/~/media/webrenewal/campaign/ads/america/udcdc/lxdc_product_brief.ashx

Component	Part Number	Max. Current Draw	Quantity	Subtotal
Microcontroller	dsPIC33EP512MC806	70mA	1	70mA
Wireless Transceiver	ATWINC1500	268mA	1	286mA
ADC	TI PCM1864	43mA	1	43mA
Microphones	SPU0410LR5HQB-7	5mA	4	20mA
Maximum output current requirement for DC/DC converter				419mA

Table 1: DC/DC converter current budget calculation

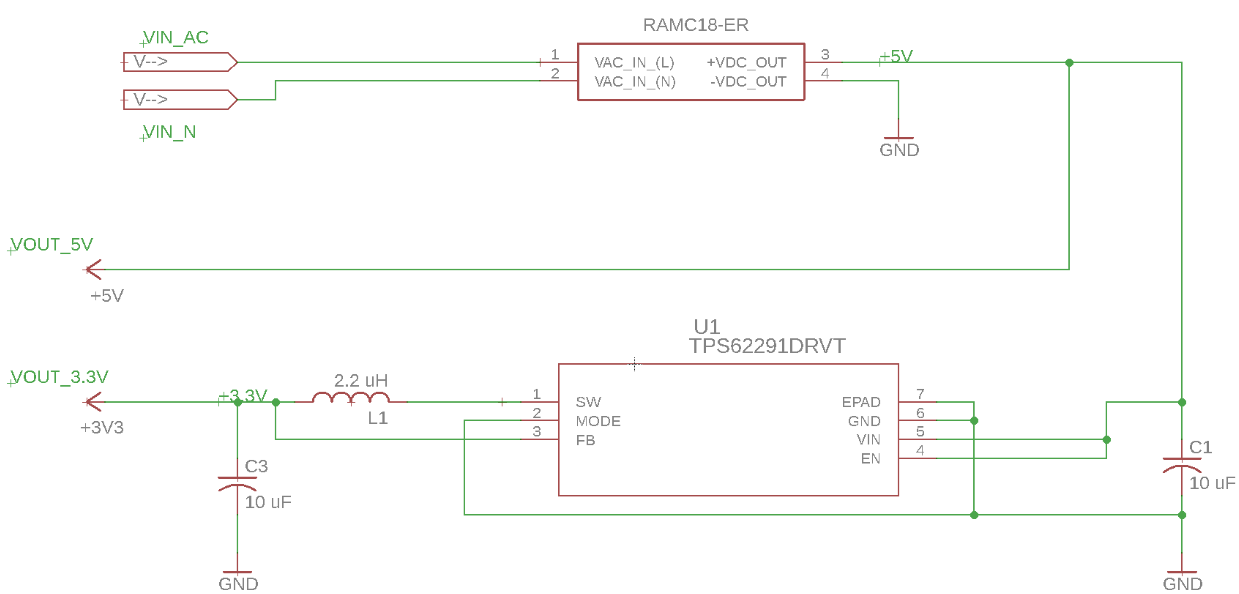

Figure 5: Power Supply Module

Power Supply	Requirements	Verification
	1. The AC/DC converter must handle converting from 120V, 60Hz to 5V DC	Supply a 120V, 60Hz AC voltage and measure the output to verify a consistent 5V DC
	2. The AC/DC converter must put out a minimum current of 2.5 A	Measure the output with an Ammeter to verify that it produces a stable 2.5A
	2. The voltage regulator must be able to supply a voltage of 3.3V with error of less than 5%	Measure the output of the Buck converter and verify the voltage stays within +/- 0.165V of 3.3V

2.6 Server

We will be using a laptop to function as the server for our project. It will be able to receive the transmission from the device over wireless transmission and compute the signal analysis to identify the gesture performed. It will then transmit the identified gesture back to the device, allowing it to perform the action.

Server	Requirements	Verification
	1. The server must be able to receive and transmit data wirelessly to the device	Receive a signal from the device, check the integrity, then transmit the same signal back and check if it is the same using the MCU

2.7 Software Signal Processing

The code will be written in Python, and take advantage of the robust signal processing libraries (i.e SciPy, NumPy, PyAudio). To calculate the time delay between the reflections of the objects in the room, the function will be calculating the CIR for each pulse and series of reflections. We first split the signal received into sections, starting from the first part of a pulse and spanning the next 5000 samples to capture the reflections.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}$$

Equation 1: Discrete Fourier Transform[15]

We then calculate the Channel Impulse Response by using Equation 1 to transform both the pulse section P_{iw} and the initial pseudo-random pulse played by the speaker S_w.

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N}, \quad n \in \mathbb{Z},$$

Equation 2: Inverse Discrete Fourier Transform[15]

By dividing P_{iw} by S_w and using Equation 2 to transform it back into the time domain, we get the CIR. To see only the reflections due to the hand, we then divide this CIR by a calibration CIR that was done prior to the hand motion. To increase the accuracy of the peaks in the CIR, a pulse signal of large frequency bandwidth with unique phase is played from the speaker at repeating intervals. The distance from the hand to each microphone is calculated by using Equation 3.

$$Distance = \frac{(Peak_1 - Peak_2)}{Sample\ Rate} * 343 \frac{m}{s}$$

Equation 3: Distance Calculation

Where, the values for $Peak_1$ and $Peak_2$ are their location in CIR array. Dividing that by the sample rate of the ADC (48 kHz) will give us the time, and multiplying by the speed of sound provides us the distance.

We calculate $Peak_1$ by finding the first location in the array that is above a certain threshold. The threshold is determined empirically over a series of test runs. After finding the first peak, the next one is calculated to be the largest magnitude peak that correlates to $Peak_1$ within range of 500 samples. We will start with this range, as it corresponds to 3.5 m distance, and then increase it as we fine tune our algorithm.

The Calibration CIR allows us to take advantage of the static reflections by the room's environment. Figure 6 below illustrates this principle. The first shows all the reflections due to the environment, while the second was taken 0.5 seconds later, correlated to then subtracted by the first. The Y axis for both graphs is the magnitude of the signal, and the x axis is time.

The software was prototyped using a Raspberry Pi 2 B and the ReSpeaker 4-input array by Seed Studio[14].

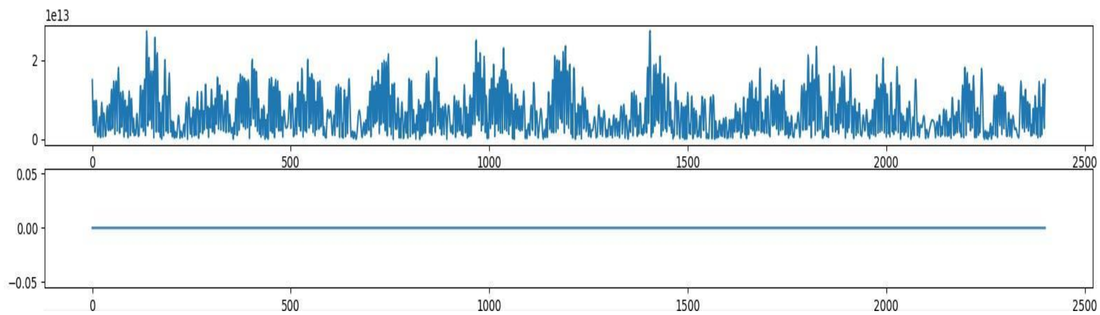

Figure 6: CIR before and after Calibration

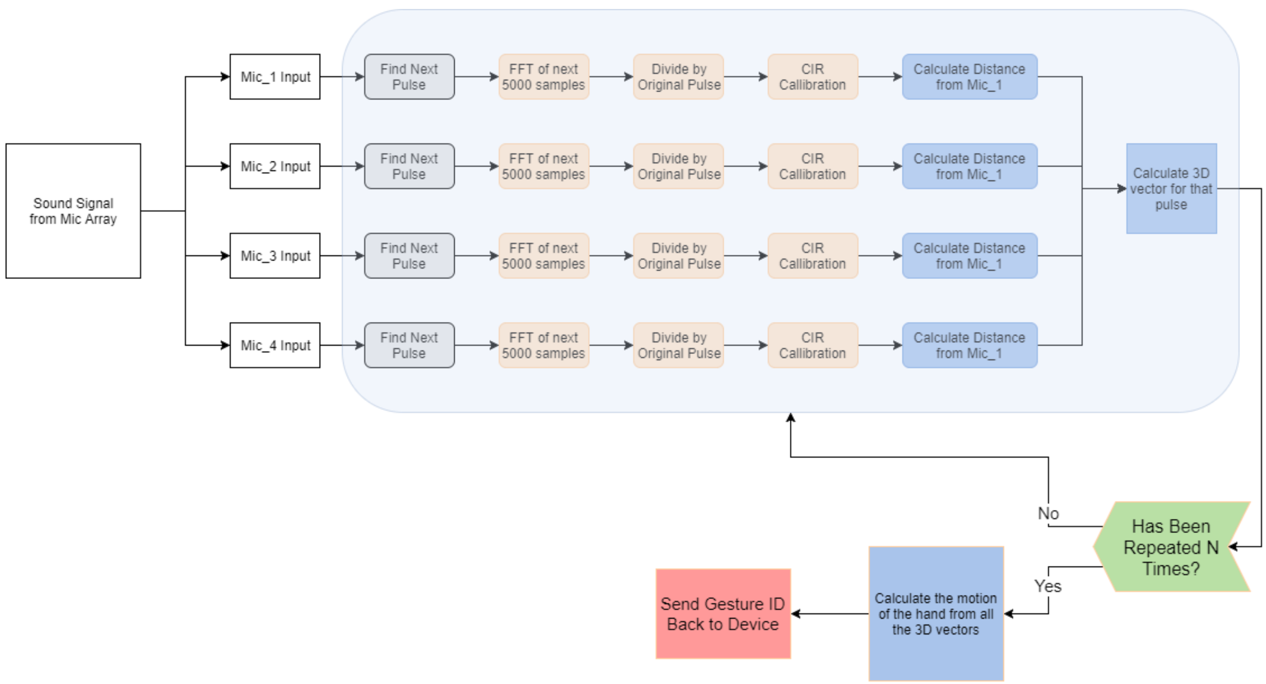

Figure 7: Gesture recognition Flowchart

Figure 7 above, illustrates how the code functions. In the flowchart, N is the number of pulses. We have currently decided that 10 pulses provide enough information to identify a gesture, however we may adjust this number through testing.

Software	Requirements	Verification
	1. The code must be able to: Perform DFTs and IDFTs	Perform DFTs and Inverse DFTs of known signals and verify the results
	2. The software code must be efficient enough to perform the calculations in under 7 seconds	Time how long it takes for the software to run with inputs of the same size as the device will send, and verify it is less than 7s
	3. The CIR Calibration must be able to remove the peaks in the CIRs due to the room environment	Measure the distance to the nearest reflective surfaces in the room and calculate their location in the CIR Array, to verify they have been removed.
	4. The 3D vector calculation must be accurate to within 3 cm at 2 m	Measure the location of the hand relative to the device, and repeat for 10 different locations, each within 2m of the device.
	5. The algorithm must be able to identify at least 3 different types of gestures with ≥ 95% accuracy	Perform each gesture at least 20 times, and record the number of times it is identified correctly.

2.8 Circuit Schematic

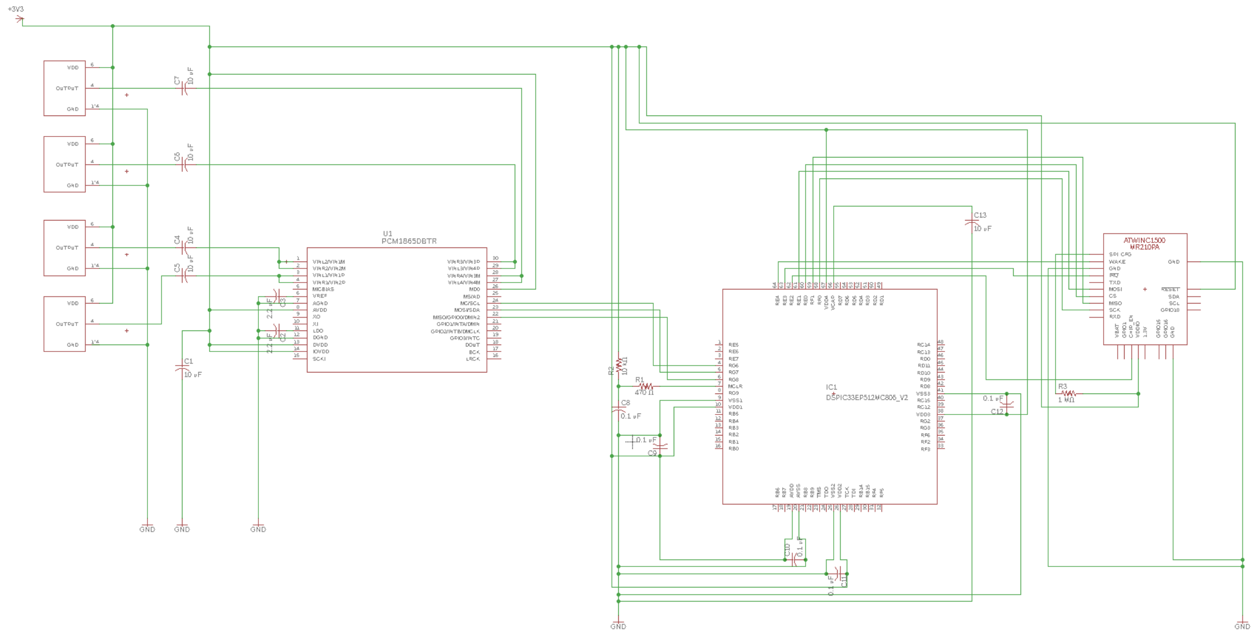

Figure 8: Microphone array, MCU, and Wi-Fi schematic

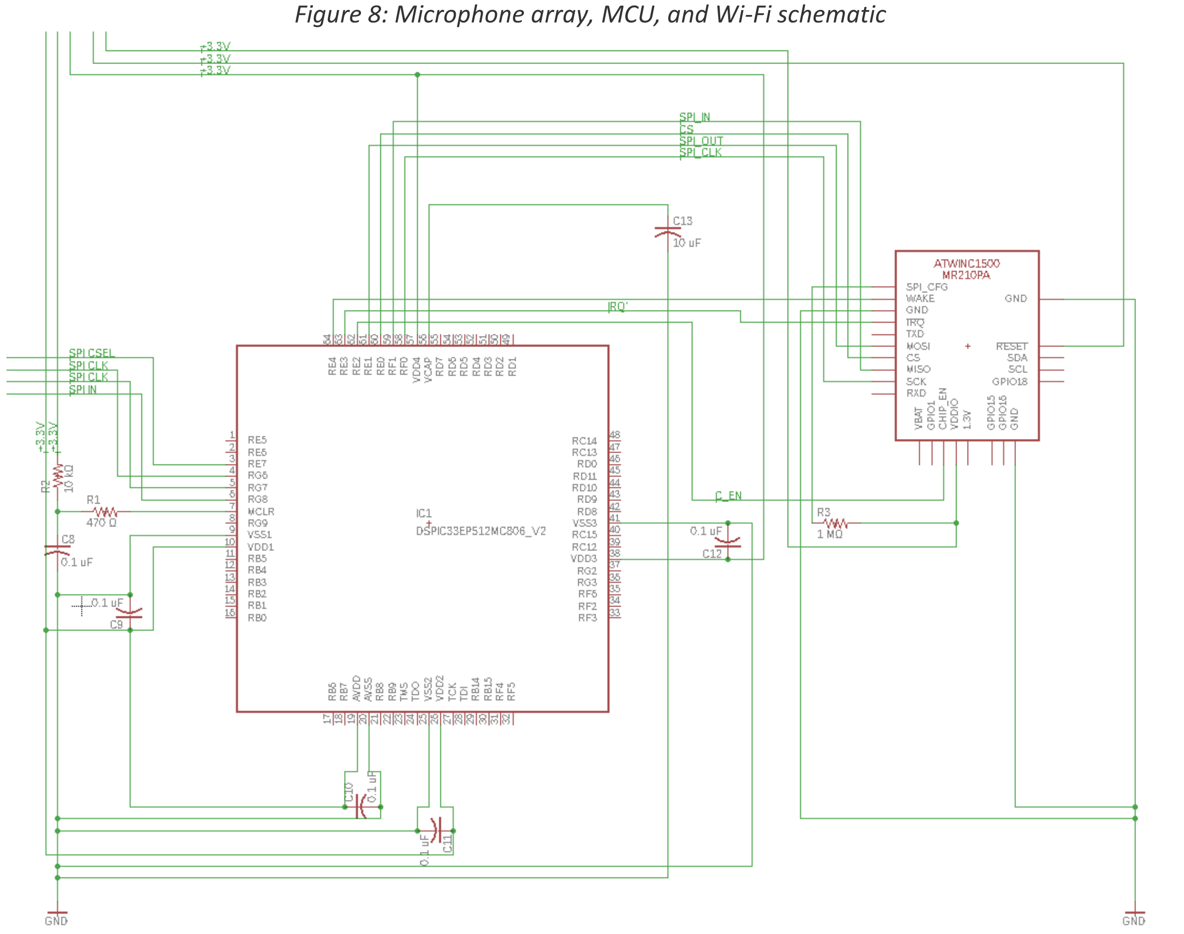

Figure 9:MCU to Wi-Fi schematic

2.9 Tolerance Analysis

The critical piece for our device that greatly affects our device performance is the sensitivity of the Microphone Array. In order to function properly, the microphones need to have a high sensitivity for our frequency range of 20kHz-24kHz. We are using short pulse signals that have a very high frequency content and random phases assigned. The microphone membranes need to be able to accurately identify those frequency changes. Moreover, the reflections, which occur within that same frequency range, are going to have a significantly lower signal strength. To properly calculate a 3-dimensional vector for the hand at each pulse, all 4 microphones need to be sensitive enough to pick up on these reflections. Our microphones have a reported signal-to-noise ratio of 94 dB SPL @ 1kHz and .15% total harmonic distortion, which indicates high levels of signal in relation to levels of noise.

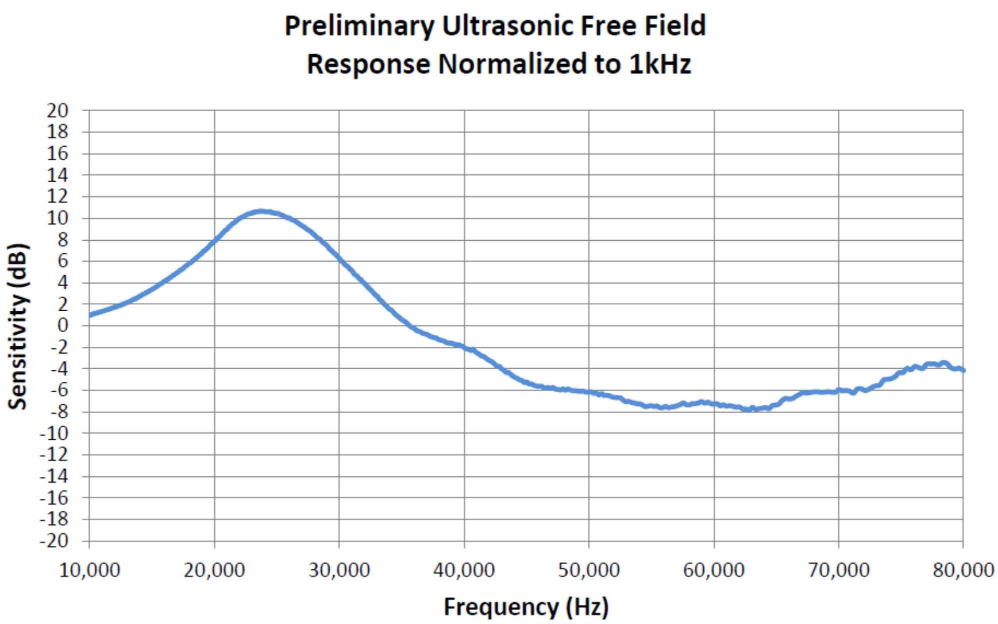

Figure 10: Sensitivity vs Frequency graph for Microphone[7]

Figure 10 shows Sensitivity response across the frequency range of 10kHz to 80kHz. The sensitivity is the highest in our targeted range of 20kHz - 24kHz, peaking at 24kHz. Therefore, our choice for Microphones aligns very well with our transmission frequency range.

On the other hand, having a high Sensitivity in this range means that not only the hand's reflection, but also all the other reflections that occur off the body and room environment will be picked up. We can calibrate the Channel Impulse Response on the software side by generating a reference CIR of just the room environment before the gesture begins. This will take care of the irrelevant reflections. However, we will need to account for the movement of the body, as it's possible that reflections can add up due to multipathing. In the first graph of Figure 6, we can see that there are multiple peaks that reach the same magnitude, which occur later than the reflections from the hand. Therefore, instead of using the calibration pulse at each CIR, we will be using the last CIR calculated as the calibration for the next CIR. This process follows the same idea as Figure 6. This will reduce the amount of irrelevant reflections, making the peak detection more manageable. Moreover, with a high sensitivity, we will need to implement better peak detection algorithms in the software code. We will need to identify the baseline noise levels of the room and accurately determine when the peaks are present in the recorded signal for each microphone, before sectioning the signal off.

3. Costs

PARTS				
Part Name	**Distributor**	**Unit Cost**	**Quantity**	**Total**
Microcontroller	Microchip	$7.42	1	$7.42
Flash Memory	Microchip	$0.64	1	$0.64
PCBs	PCB Way	$10.00	3	$30.00
			Controller Module	**$38.06**
Microphones	Knowles	$0.94	4	$3.76
ADC	TI	$3.59	1	$3.59
PCBs	PCB Way	$10.00	3	$30.00
			Microphone Module	**$37.35**
Speaker	Adafruit	$14.95	1	$14.95
Amplifier	Adafruit	$19.95	1	$19.95
			Speaker Module	**$34.90**
Wireless Transmitter	Microchip	$7.84	1	$7.84
PCBs	PCB Way	$10.00	3	$30.00
			Wireless Module	**$37.84**
AC/DC Converter	Recom	$37.00	1	$37.00
Voltage Regulator	Murata	$4.14	1	$4.14
PCBs	PCB Way	$10.00	3	$30.00
			Power Supply Module	**$71.14**
			PARTS TOTAL	**$219.29**
Labor				
Team Member	Hourly Rate	Total Hours	Expense Multiplier	Total Cost
Sean Nachnani	$50.00	160	2.5	$20,000
Hojin Chun	$50.00	160	2.5	$20,000
			LABOR TOTAL	**$40,000.00**
			GRAND TOTAL	**$40,219.29**

Table 2: Cost Table

4. Schedule

Week	Sean	Kevin	Both
2/19	Finish Power Circuit Schematics	Order Parts	Design Review
2/26	Work on Power Supply PCB	Begin laying out Power Supply on breadboard	Soldering Assignement
3/5	Begin Laying out Microcontroller on breadboard	Begin laying out ADC on breaddboard	
3/12	Begin laying out Microphones on breadboard and begin conneting all module system together on breadboard	Begin laying out Wireless Transceiver and speaker on breadboard	Breadboard prototype completed
3/19	Spring Break	Spring Break	Spring Break
3/26	Work on code in Python	Revise and Submit PCB design to Machine Shop	Individual Progress Report
4/2	ebug and fine tune Microcontroller and microphon	Debug and fine tune ADC and Wireless Transceiver. Sign up for Mock Demo	
4/9	Debug coding in python	Debug and fine tune power supply Sign up for mock presentation, demonstraion and presentation	
4/16	Mock Demo	Mock Demo	All parts soldered together
4/23	Prepare for demonstration and mock presentaion	Begin Final Paper	Demonstration
4/30	Prepare for Presenation	Prepare for Presentaion	Turn in Final Report

Table 3: Schedule

5. Ethics and Safety

For our project, there are few safety concerns. As mentioned above, our device will be plugged into a wall outlet. To be able to do that, we will have to have a AC/DC converter. The danger comes when we are dealing with wall outlet voltage at 120V AC and converting it to 24V DC. We will need to make sure the wall outlet contains a ground using one hand method and we will also need to make sure that AC/DC conversion is off limit from the user so they never have to come into contact with high voltages.

When dealing with high voltages, the concern of large current comes along with it. With large current, it can also dissipate heat. So when we are dealing with high voltages and high current, we will have to careful to the heat and we will make sure that the user will never be exposed to the excessive heat.

We are responsible for all decision we make for the design of our device and it is our responsibility to disclose any issues that might be dangerous to the user per Section 1 of the IEEE code of Ethics[4].

Lastly, since our device involves microphone and server to process the data, we need to be careful about privacy issues. As it was for the issue for Alexa, where the idea that Alexa is always listening or may somehow incriminate someone can be issue for owerner's of Alexa. To protect everyone's privacy, our device won't be on all the time, and will only take in data when you start the device, and it will mostly keep the data locally and not share or only upload anonymously or encrypted. We believed that if our device is properly and well designed, we will lessen these hazards to create an enjoyable experience for the user.

References

[1] W. Mao, J. He, and L. Qiu, “CAT: High-Precision Acoustic Motion Tracking,” Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking - MobiCom 16, 2016.

[2] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, “FingerIO,” Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI 16, 2016.

[3] N. Roy, H. Hassanieh, and R. R. Choudhury, “BackDoor,” Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys 17, 2017

[4] IEEE (2017). IEEE Code of Ethics. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 7- Feb- 2018].

[5] Microchip.com, “16-Bit Microcontrollers and Digital Signal Controllers with High-Speed PWM, USB and Advanced Analog”, 2012. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/70616g.pdf

[6] Microchip.com, “2 Mbit 2.3-3.6V SPI Serial Flash”, 2013. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/20005135B.pdf

[7] Digikey.com, “Zero-Height SiSonic Microphone”, 2013. [Online]. Available: https://www.digikey.com/product-detail/en/knowles/SPU0410LR5H-QB-7/423-1139-1-ND/2420983

[8] ti.com, “PCM186x 103 dB 2ch and 4ch Audio ADCs With Universal Front End (Rev. A)”, 2014. [Online]. Available: http://www.ti.com/product/PCM1864/datasheet/abstract#SLAS8312137

[9] adafruit.com, “20W 4 Ohm Full Range Speaker - XS-GTF1027,” [Online]. Available: https://www.adafruit.com/product/1732

[10] adafruit.com, “20W Stereo Class D Speaker Amplifier with Volume Control”, 2008. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/MAX9744.pdf

[11] microchip.com, "ATWINC15x0-MR210xB IEEE® 802.11 b/g/n SmartConnect IoT Module", 2017. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/70005304B.pdf

[12] recom-power.com, "RECOM AC/DC-Converter", 2017. [Online]. Available: https://www.recom-power.com/pdf/Medline_AC-DC/RACM18-ER_W.pdf

[13] murata.com, "LXDC Series", 2014. [Online]. Available: https://www.murata.com/~/media/webrenewal/campaign/ads/america/udcdc/lxdc_product_brief.ashx

[14] github.com, "ReSpeaker-4-Mic-Array-for-Raspberry-Pi", 2017. Available: https://github.com/SeeedDocument/ReSpeaker-4-Mic-Array-for-Raspberry-Pi/blob/master/src/ReSpeaker%204-Mic%20Array%20for%20Raspberry%20Pi%20%20v1.0.pdf

[15] wikipedia.org, "Discrete Fourier transform", 2018. Available: https://en.wikipedia.org/wiki/Discrete_Fourier_transform