ACOUSTIC MOTION TRACKING

ECE 445 - MOCK DESIGN REVIEW

TEAM 35

SEAN NACHNANI

HOJIN CHUN

TA

YUCHEN HE

High-Level Requirements List

- The Device must be able to wirelessly transmit the recorded signal to an external device for signal processing.
- The microphone array must be able to record frequencies above 20khz to allow for the speaker to transmit in the inaudible range
- The Device must be able to accurately measure the distance of the hand performing the gestures to within 3 cm, to allow for proper identification of the motion.

Block Diagram

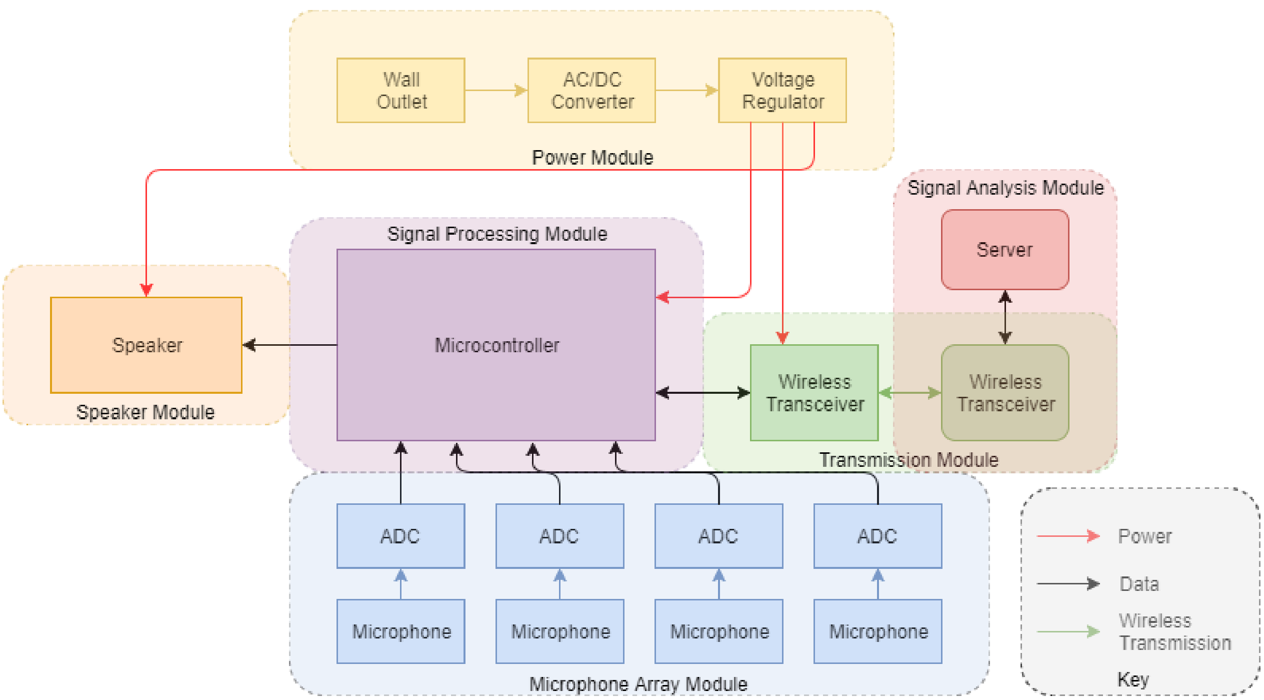

Figure 1: Component Block Diagram

Physical Design

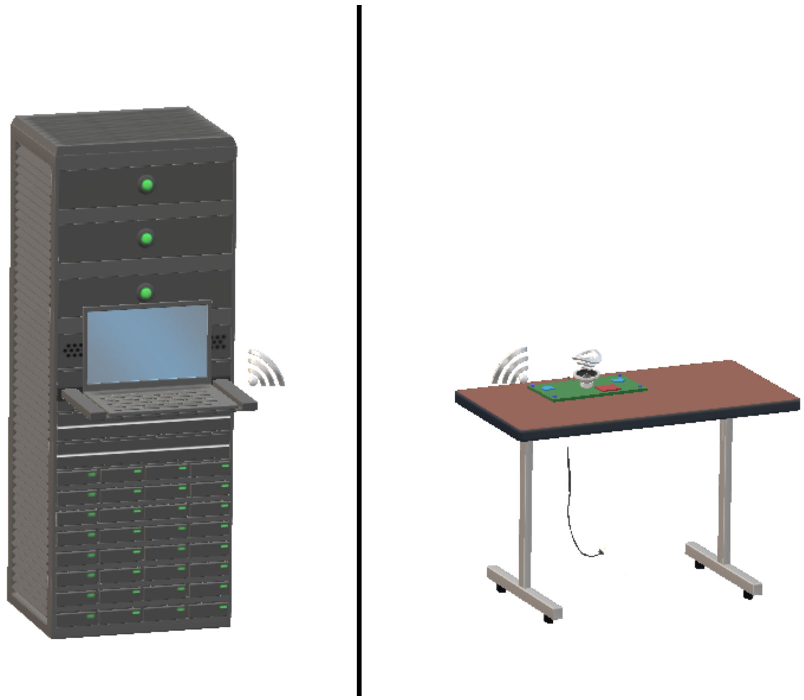

Figure 2: Physical Representation of the System

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).

Functional Overview

Microcontroller

The microcontroller is used to sample from the four microphone and ADC pair 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 TI TMS320C5515 Fixed-Point Digital Signal Processor as our MCU. It has a low power consumption of 3.3V, 320K Bytes Zero-Wait State On-Chip RAM, as well as an FFT Coprocessor to allow for initial signal processing to be done on the device before transmitting.

Microphone Array

We have chosen to use 4 microphones to create the array. We are using Knowles SPU0410LR5H-QB-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 PCM1864 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.

Speaker

The speaker needs to be able to transmit frequencies up to 24khz. For this device we are planning on using the XS-GTF1027 speaker by Adafruit which has a frequency response of 60Hz - 24kHz.

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 SimpleLink Wi-Fi CS3220. This component has 802.11b/g/n radio, baseband, and MAC protocols, allowing for standard wireless transmission. It has both SPI, I^2C, and UART interfaces, and supports rates of 16Mbps (UDP) and 13Mpbs (TDP), which is more than enough to transmit the recorded signal. The voltage requirement ranges from 2.1 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.

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 1.8V – 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.

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.

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. 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.

Due to the room itself being a static variable, as the walls and furniture are not mobile, a calibration pulse is first computed. The next series of recorded signals received will be during the motion of the user's hand. By convolving the signals with the original unique frequency pulse, and subtracting out the calibration pulse, the motion of the hand can be calculated.

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

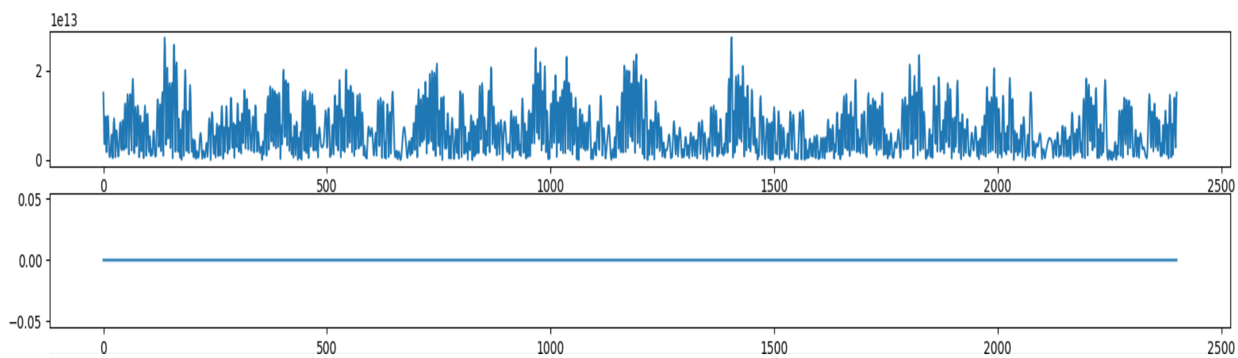

Figure 3: CIR before and after Calibration

In Figure 3 above, the top graph displays the CIR of a single microphone with magnitude on the vertical axis, and frequency bins on the horizontal axis. The bottom graph displays the subsequent pulse but with calibration, and the same axis. For this demonstration the hand was stationary.

Requirements and Verification

	Requirements	Verification
Microcontroller		
	1. The controller must support SPI and I^2C	Send a signal over SPI and I^2C 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 > 1Mbps	Transmit 10Mbps over SPI and verify the signal arrived in ≤10s
Microphone Array		
	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 ADC must have SPI or I^2C interfaces	Send a signal over SPI or I^2C and verify the output is the same as the signal transmitted
Speaker		
	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
Wireless Transmission		
	1. The wireless transmitter must be able to transmit over 2.4GHz channel and support standard Wi-Fi protocols	Transmit a signal through the component over 2.4Ghz channel MAC protocol and verify it matches the original

Wireless Transmission	2. The range must be at least 40ft	Perform the above transmission beginning next to the component then at 5ft increments until 40ft is reached
Power Supply		
	1. The adapter must handle converting from 120V, 60Hz to DC voltage	Supply a 120V, 60Hz AC voltage and measure the output to verify consistent DC voltage
	2. The Voltage regulator must be able to supply voltages in the range of 1.8V – 5V with error of less than 10%	Measure the output at 1.8V, 3.3V and 5V and verify the voltage stays within +/- 0.18V, 0.33V, and 0.5V of each voltage respectively
Server		
	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
Software		
	1. The code must be able to: Perform DFTs and Convolutions	Perform DFTs and convolutions 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

Circuit Schematics

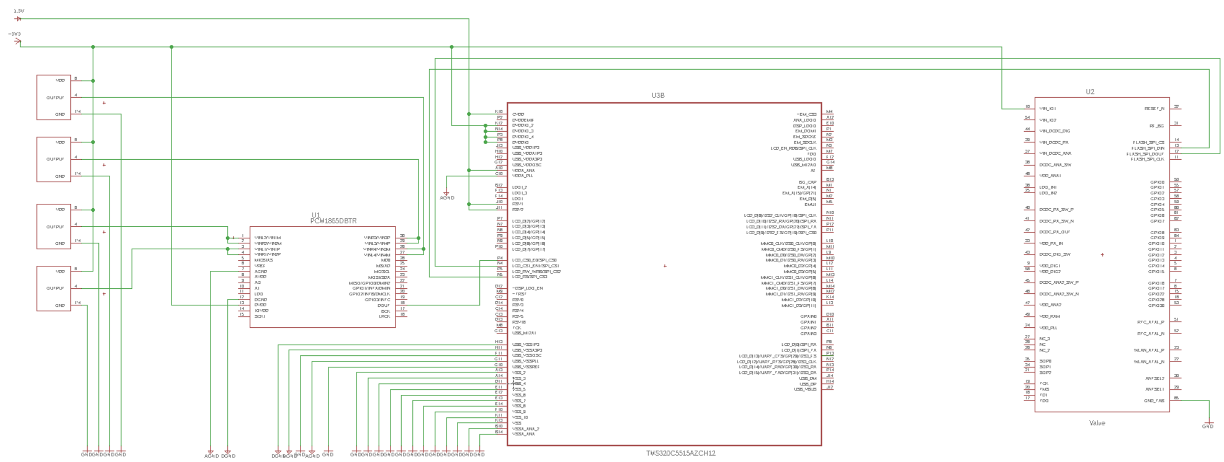

Figure 4: Microphone Array and MCU Schematic

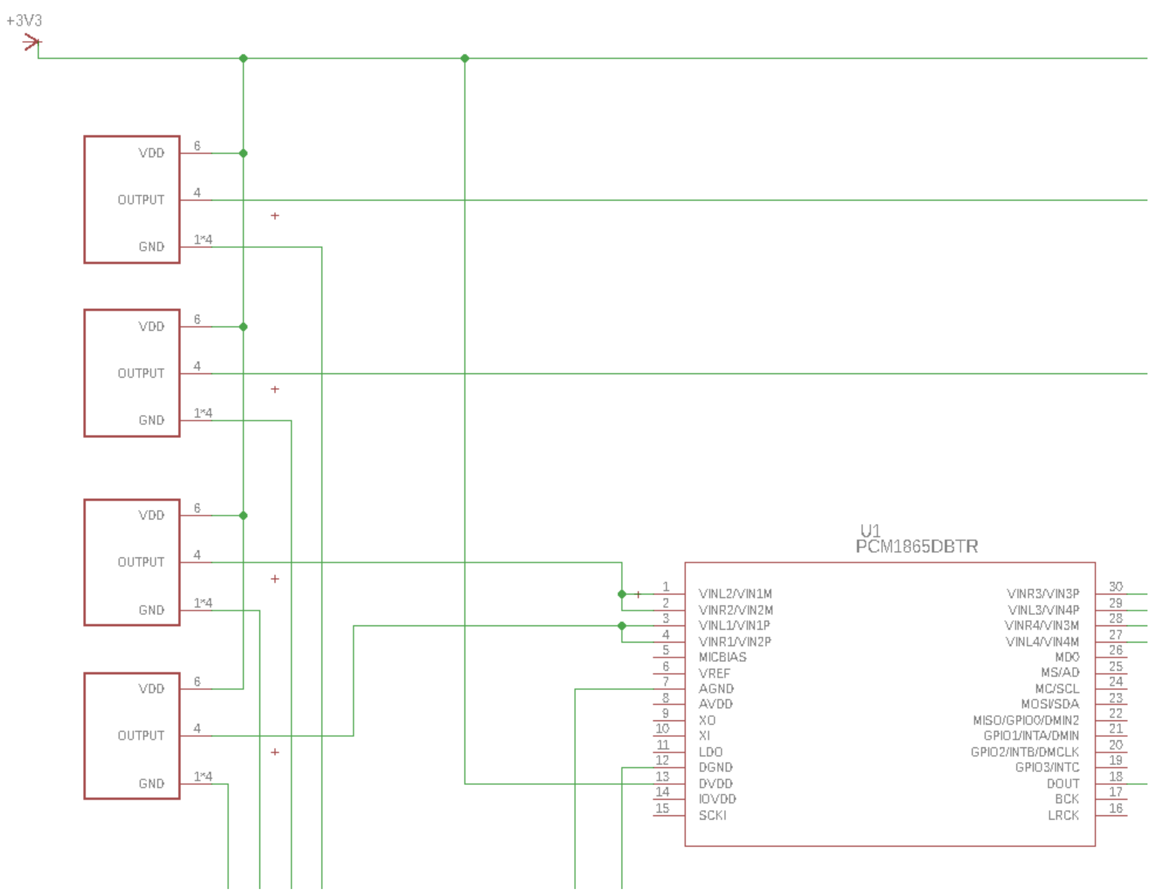

Figure 5: Microphone and ADC Closeup

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]. 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].