Software Controlled Sound Sources

ECE 445 Design Document

Won Woo Lyu - Wlyu2
Micki Rentauskas - Mar3
Ben Sisserman - Bens3

Group 26
TA: AJ Schroeder
9/29/20

Table of Contents

1 Introduction

1.1 Objective

We will build three unique Software Controlled Physical Sound Source (SCPSS) devices for our client, postdoctoral student Ryan Corey, who conducts research on hearing aids at the *Illinois Augmented Listening Laboratory* (IALL). Testing hearing aids has always been a task to audiologists since soundfield speakers with the introduction of background noise through a soundfield speaker cannot replicate real-world experiences. Improvements of technologies have made a smallear leap between lab testing of hearing aids and actual real-world listening, but these systems are very complex and still not fully capture the real-world sounds [1]. They are complex, because replication of real-world environments requires a variety of different types of sounds from different distances and angles. The best way to produce a sound from a speaker that yields accurate real-world results is to put eight speakers around the person for every 45 degrees [2], which tells that it will be harder to produce accurate real-world sound if the lab environment gets complex.

Currently, Ryan uses speakers placed throughout the lab to demo prototypes. The speakers use pre-recorded sounds that must be originally recorded in anechoic chambers, which can be expensive. Speakers also direct their audio output, unlike their physical counterparts. Our physical sound sources will create a sound wave that would go in all directions and reflect off of the walls of the room creating a new sound that a speaker would not be able to mimic since the sound wave of the speaker will be directly away from the speaker along its axis [3]. Our product solves these problems by allowing various physical and electronic devices to be used and controlled by our own Python Module, which can be easily utilized by the IALL.

1.2 Background

Ryan Corey, a postdoctoral student at University of Illinois at Urbana-Champaign, does research on audio-processing for noise cancelling hearing aids using speakers placed throughout a room to emulate various sound sources for testing. He has requested that we create software controlled real sound sources for his lab. In his lab, most of the sounds are outputted from a series of speakers, and sounds are produced by using specialized software. The below picture is from the article [4] of his team working on Cooperative Listening Devices, and they used 12 speakers to test Cooperative Listening Devices. These speakers can be replaced by our project, which will produce various real sounds while not using a speaker.

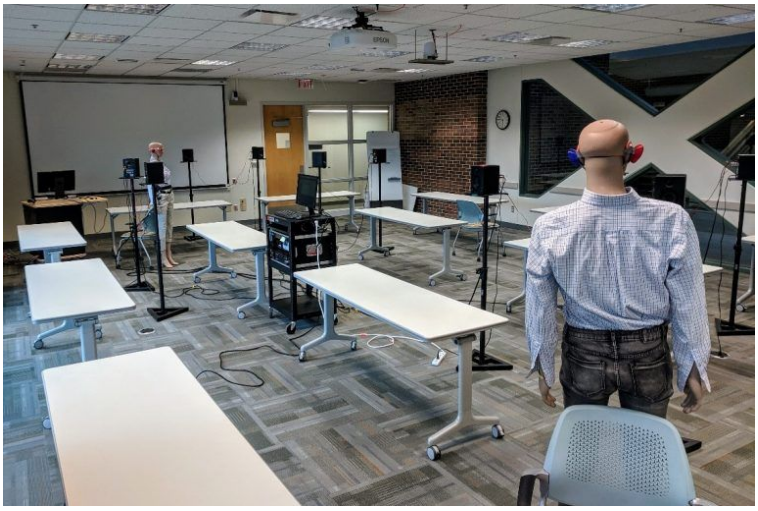

Figure 1. The conference room used for the massive distributed array dataset

1.3 High-Level Requirements

- The devices must be able to control at least three different physical sound sources.
- Latency variation must be kept to a minimum. Latency itself is not a concern, but latency must vary by more than five milliseconds.
- Devices must function with at least six feet of distance between devices for simulating real-world environments and circumstances.

2 Design

2.1 Block Diagram

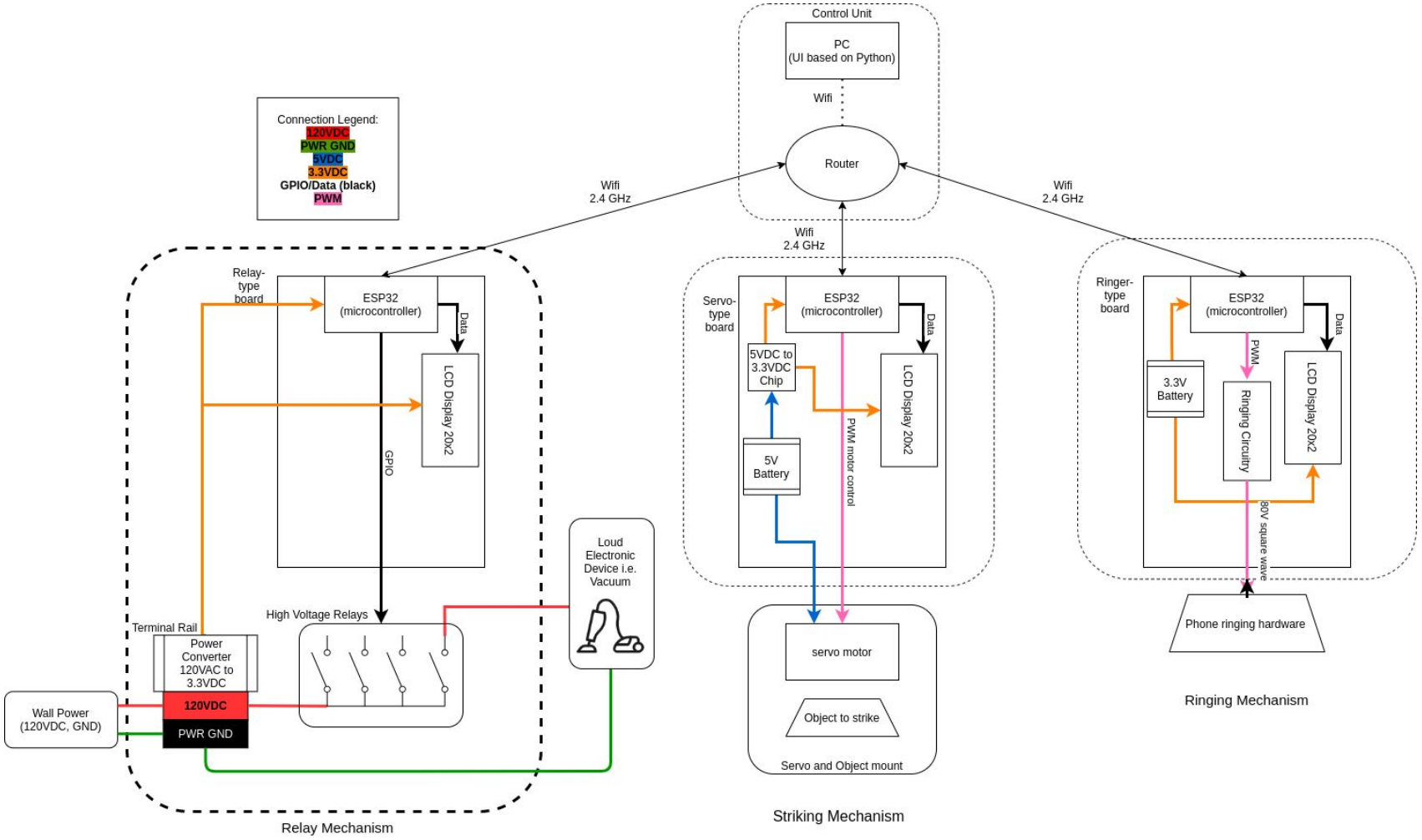

Figure 2: Block Diagram

2.2 Physical Design

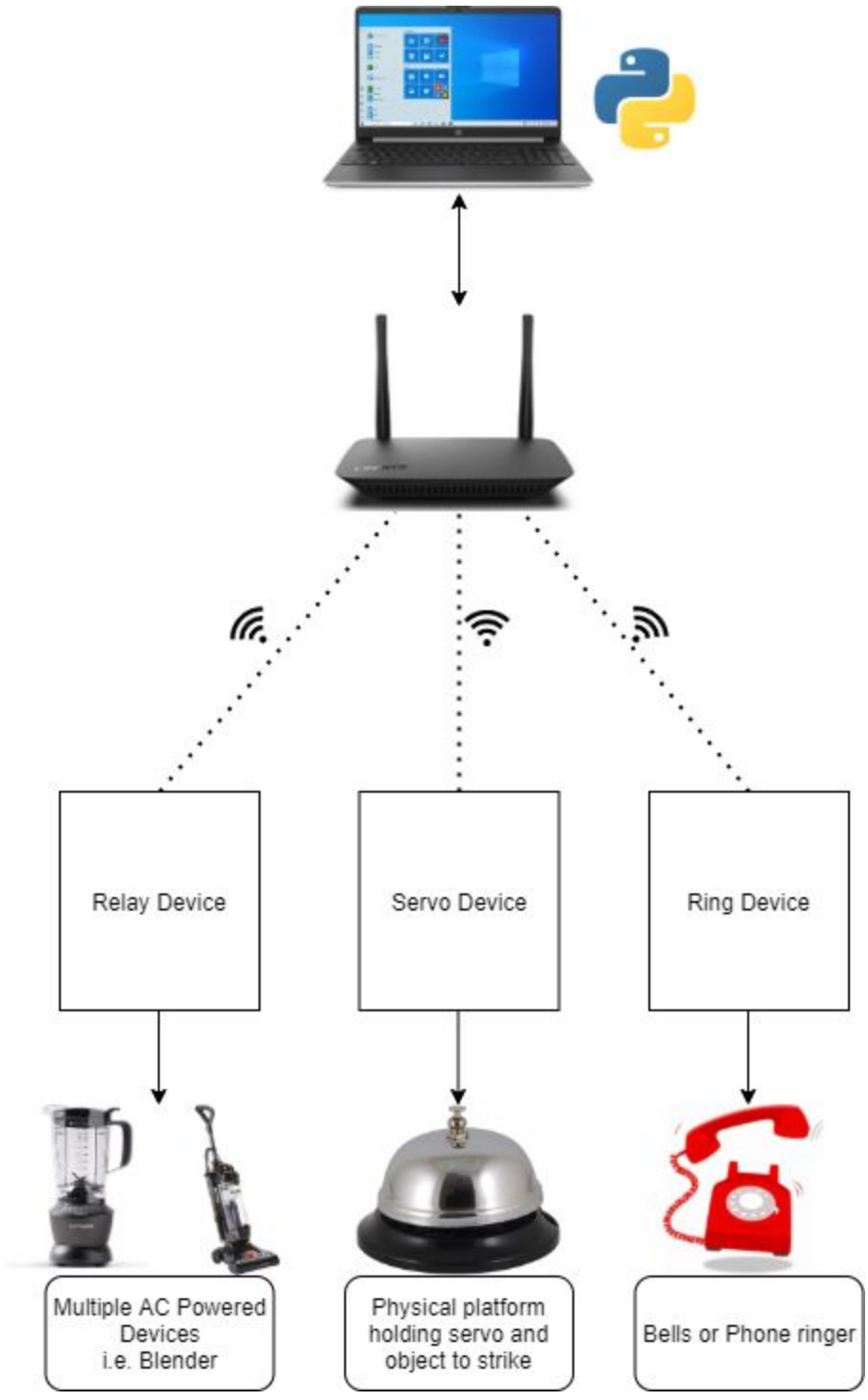

Figure 3: Physical Overview

2.3 Requirements and Verification Tables

2.3.1 Control Unit

Requirements	Verification
1. The Python module must be able to identify and establish TCP communication with all the boards. 2. Another requirement for this system is to ensure in software that all messages are consistent in their timing. We do not care about latency, but about latency consistency. Latency should not vary by more than 5 ms. 3. The Control Unit must be OS agnostic. The user PC should not be limited by the syste*m*	1. Use the Python module on the host PC to initialize the boards. Use the API to print out the number of boards and their type. All available SCPSS devices should be listed. 2. Measure time at right before issuing command in Python script and record timing of sound generation using a stopwatch. Repeat five times and check that results are consistent to within +/- 2.5 ms (5 ms). 3. Repeat verifications 1 and 2 on Linux, Windows, and Mac.

2.3.2 120VAC to 3.3VDC Power Converter

Requirements	Verification
1. Capable of supplying at least 1A for powering ESP32 and LCD display	A. Connect to 120VAC supply, and check output pins for 3.3VDC +/- .5VDC with a multimeter. B. Load test with $R = \frac{V}{I} = \frac{3.3}{1} = 3.3\Omega$ resistor rated at minimum power $P = IV = (1)(3.7) = 3.7W$

2.3.3 ESP32

Requirements	Verification
1. Receive and decode commands over 2.4GHz Wifi 2. Receive and decode commands via USB 3. Output 3.3VDC +/- .5VDC GPIO for given duration 4. Convert input from sense circuitry and translate into battery status 5. Output data to LCD display	1. Test receiving and sending a string over UDP and TCP. 2. Connect ESP32 to Host via USB and test sending and receiving strings. 3. Measure output of GPIO using multimeter. 4. Display battery status on LCD monitor and measure with multimeter to verify result. 5. Test outputting a few strings to the LCD display.

2.3.4 LCD Display

Requirements	Verification
1. Must be able to receive 3.3V signals from GPIO 2. Must be large enough to display strings of at least 20 characters	1. Send data to the display using the ESP32 microcontroller, and check that sent data is displayed. 2. Send data like in (1), using a string with 20 characters and checking they are all shown.

2.3.5 Off-the-Shelf Quad Relay Board

Requirements	Verification
1. Toggle positive line voltage for four separate devices 2.Tolerate up to 12A for each relay at 120VAC 3. <10ms latency between received GPIO and switch action	1. Apply high signal to relay coil for each relay, and check for short across NO and COMMON 2. Use Electronic Load to test each relay separately. 3. A. Apply a square wave with 50ms period and 50% duty ratio B. Put oscilloscope probes on square wave and across NO and COMMON C. Set scope trigger at 0V so it catches when the relay closes D. Use horizontal measuring tool on scope to check delay between the PWM going high and the relay shorting

2.3.6 Terminal Rail

Requirements	Verification
1. Hold terminal blocks at 120VAC and PWR GND and provide isolation from each other 2.Support up to 48A of current (12A per switch-type device)	A. Apply 120VAC across terminal nodes using a power supply and use an electronic load to apply 48A current.

2.3.7 Battery Sense Circuit

Requirements	Verification
1. Convert voltage of the battery to a level that the GPIO of the ESP32 can handle and translate 2. Draw negligible current, at most 5 microAmps	1. Verify by displaying value on LCD display and compare it to reading from a multimeter. 2. Use a multimeter to measure current through the circuit and check that it is within bounds.

2.3.8 5V Battery

Requirements	Verification
1. Must provide at least 1.5A for three hours of frequent use of both the microcontroller and the servo motor (once or twice a minute)	1. Run a Python script to turn on and off one of the SCPSS devices with a delay of 10 seconds between commands, and measure time until battery dies or low battery is displayed on LCD.

2.3.9 Step-Down Chip

Requirements	Verification
1. Must be able to step down 5VDC input to 3.3V +/- 0.3VDC output	1. Verify inputs and outputs using multimeters.

2.3.10 Servo motor

Requirements	Verification
1. Must be able to take as input 3.3VDC PWM from ESP32 to control the motor. 2. Must provide enough torque to make the strike audible for at least 10 ft.	1. Use a function generator to create a 3.3V PWM with various PWMs and check that the motor steps with the PWM as expected 2. Attach a pen or pencil to the arm of the servo, and place next to a small piece of metal or a bell. Have the servo repeatedly strike the object and check that it is audible at above 10 ft distance.

2.3.11 Servo Mount

Requirements	Verification
1. Able to hold servo and object to strike for at least 12 strikes before manual realignment is required	1. Trigger motor to hit 12 consecutive times and listen for any noticeable changes in the sounds.

2.3.12 Ringing circuit driver

Requirements	Verification
1. Take 3.3V PWM from ESP32 and use it to toggle 55VAC across the H-bridge 2. Should be able to ring for at least 30 seconds continuously.	1. Use a function generator to output a 3.3V square wave to the circuit driver for the mechanism to activate. Listen and verify that the ring is consistent for at least 30 seconds.

2.4 Software Design Plot For Device Algorithm

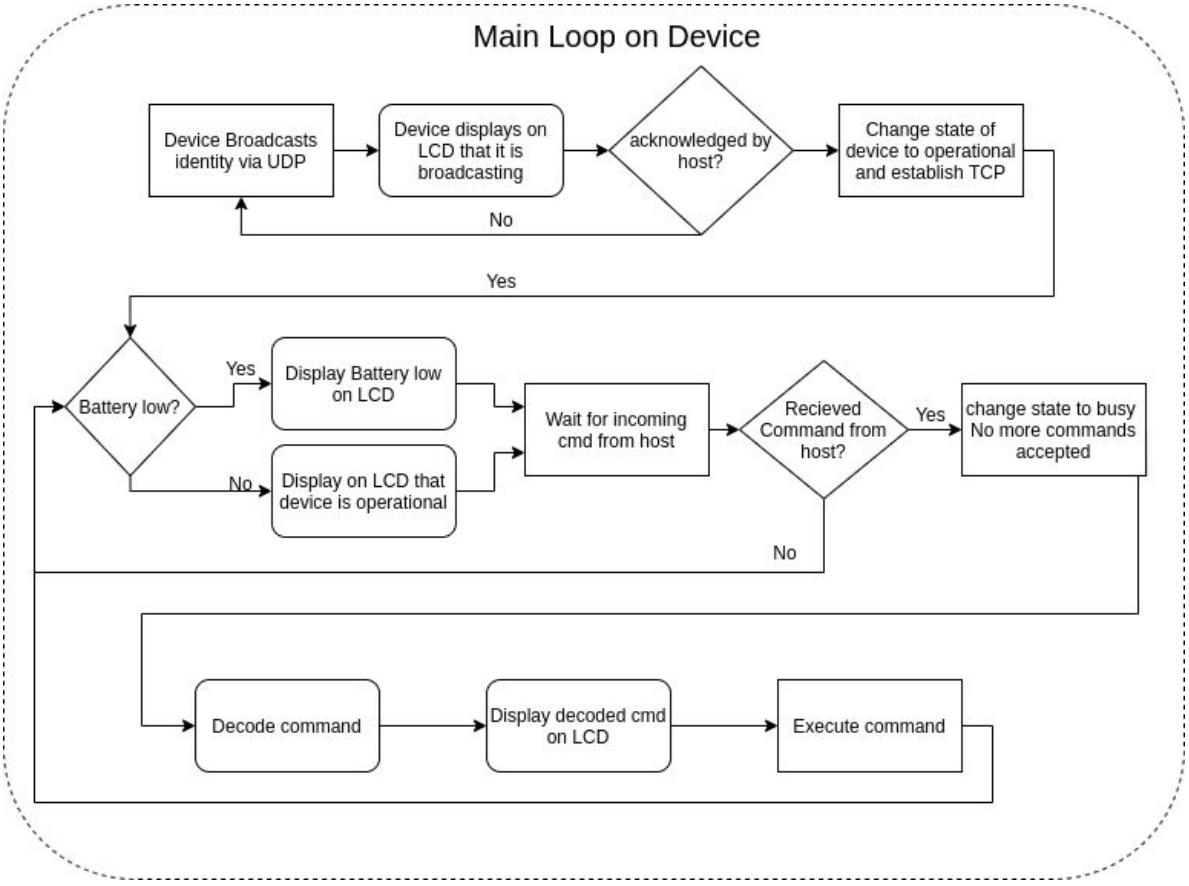

Figure 4 : Device Main Loop Algorithm

2.5 Circuit Schematics

2.5.1 Relay-type Board

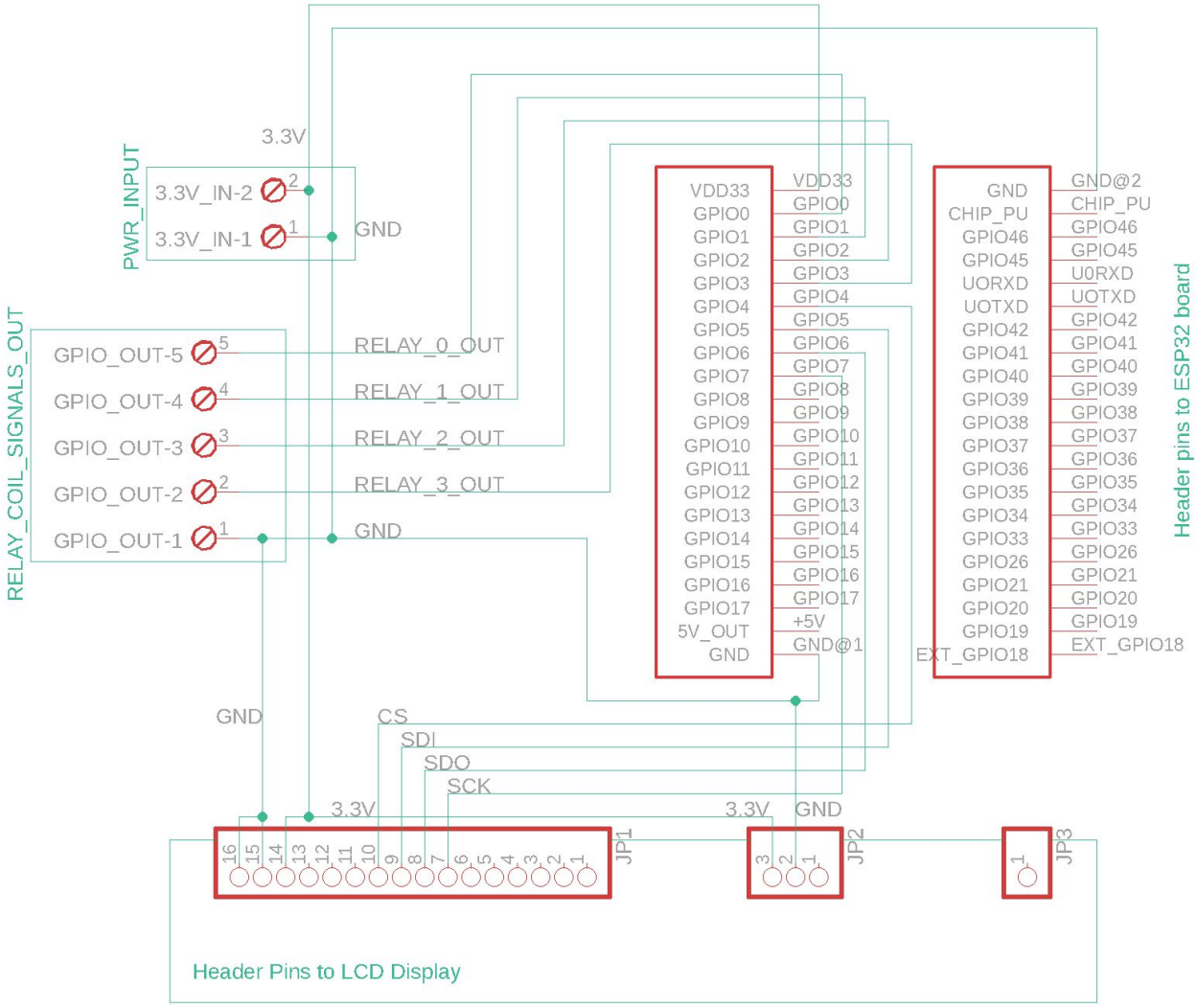

Figure 5: Circuit Schematic of the Relay Board

2.6 Tolerance Analysis

Important tolerance we must maintain is the consistency of latency between the PC and boards through using the WiFi. The average latency for 2.4 GHz WiFi is median of 6.22 ms; however, the latency itself is not the problem since it is affected equally between PC and boards. The problem we might have is that latencies over boards can be various since more than one board will be placed with different directions and angles. Distance between the board and router will affect the latency; however, it will be very small that we can consider that distance does not affect. Singal from the router transfers to the board with the speed of light, which is approximately 299,792 km/s. To have more than 5 ms of latency, the distance should be approximate of 1498.96 km according to the Eq. 1.

$$299,792\ \frac{km}{s} * 5\ ms = 1498.96\ km \quad \text{Eq. 1}$$

Next possible problem is the latency produced by the number of devices connected to the router. Theoretically, a common wireless router can hold up to 250 connected devices. If a router outputs 300 Mbps with 100 connected devices, each device will have 3 Mbps. Since a router can hold up to hundreds of devices, this will not be our problem since we are planning to use only three boards.

2.7 COVID-19 Contingency Plan

In the case that we are moved to a completely online curriculum and we lose access to the senior design lab, we intend to shift focus from the relay-based device to the servo and ring based devices. This is because we would lack the equipment to safely work on and verify the functionality of high-voltage components, thus we can shift our focus to expanding upon the devices that are safer to test and measure in our own homes. This would mean that we would have to get more creative about our servo-based device, since it is the only one left that can output a variety of sounds. One option would be to expand the functionality of the servo to strike two objects, one to the left and one to the right of the servo, rather than simply striking one object. This would require an overhaul of our API and some of the code on the ESP32, but requires no change to the hardware/electrical design of the device.

3 Ethics and Safety

Since the relay-type board will be connected to the wall power, we have to care about any components that are connected to this board since there is a possibility of getting an electric shock. In order to avoid this, we are planning to use a terminal block (aka terminal rail) to avoid any kinds of poor connected wires. Using it will lead to a convenient and safer way to distribute power from a single input source of the wall power to multiple outputs. We are responsible for our design and safety, and this safety concern is an implementation of the IEEE Code of Ethics Section I.1, “disclose promptly factors that might endanger the public or the environment” [4]. We are also considering ordering a commercial relay board to separate between sensitive components like the microcontroller and the AC power.

Since we will lead the project based on using the school's wifi during the demo, we are responsible for our design that will prevent any circumstances that violate provisions of University policy over using the school’s WiFi [5].

4 Citations

[1] Miller, A., 2020. *Extending Hearing Aid Testing Beyond The Walls Of The Sound Booth | Phonak Audiology Blog - Phonak Pro - Life Is On*. [online] Phonak Audiology Blog - Phonak Pro - life is on. Available at: <https://audiologyblog.phonakpro.com/extending-hearing-aid-testing-beyond-the-walls-of-the-sound-booth/> [Accessed 28 September 2020].

[2] Staff, H., 2007. *Developing And Testing A Laboratory Sound System That Yields...*. [online] Hearing Review. Available at: <https://www.hearingreview.com/practice-building/practice-management/developing-and-testing-a-laboratory-sound-system-that-yields-accurate-real-world-results> [Accessed 28 September 2020].

[3] Physics Lecture Demonstration Facility. 2014. *How Does A Candle Flame Respond To A Sound Wave? - Question Of The Week 2014 Summer Girls Special Part 1*. [online] Available at: <https://lecdem.physics.umd.edu/question-of-the-week-archive/154-qotw-020-with-answer.html#:~:text=Sound%20propagates%20as%20a%20longitudinal,the%20speaker%20along%20its%20axis.> [Accessed 28 September 2020].

[4] Ieee.org. n.d. *IEEE Code Of Ethics*. [online] Available at: <https://www.ieee.org/about/corporate/governance/p7-8.html> [Accessed 28 September 2020].

[5] Campus Admin. Manual. 2001. *Appropriate Use Of Computers And Network Systems*. [online] Available at: <https://cam.illinois.edu/policies/fo-07/> [Accessed 29 September 2020].