ECE 445
Senior Design Laboratory - Fall 2023
Design Document

Distributed Light System using Voice Recognition

Team 24

Anshul Goswami (anshulg3@illinois.edu)

Walter Tang (waltert3@illinois.edu)

Anish Naik (anishn3@illinois.edu)

TA: Kai Chieh (Jeff) Chang

September 28, 2023

Content

1 Introduction

1.1 Problem

Traditional light switches can pose a significant problem when it comes to safety, particularly in the dark. Imagine returning home late at night, struggling to find the elusive switch in an unfamiliar room. Fumbling around in the pitch-black, you may inadvertently knock over objects, stub your toe against furniture, or collide with walls, transforming a simple task into a potential hazard. This inconvenience isn't limited to late-night scenarios; such as during emergencies, the reliance on manual switches can be precarious, leaving you in the dark when you need light the most. This problem is exasperated by the elderly or those with physical disabilities.

Even though smart switches represent a significant improvement in terms of convenience and control over traditional switches, they are not entirely immune to the challenges posed by darkness. Smart switches that lack voice recognition capabilities may still require some form of visual interaction, such as using a smartphone app. In the absence of voice commands, you might need to fumble for your phone or find it in the dark to access the app. This process, while more advanced than searching for a traditional switch, can still be cumbersome and potentially unsafe. Additionally, if your smartphone's battery is low or if the app malfunctions, you may find yourself in the same predicament as with a regular switch. So, while smart switches offer great benefits, they may not entirely eliminate the need for a backup lighting solution in situations where quick and reliable access to light is crucial.

1.2 Solution

We want to create a solution in which you can talk to a main station to turn lamps in different locations on and off. This way, one can easily access lights without needing to move around.

We want to create a base station ('main station box') containing a microphone and raspberry pi for a speech recognition system that will send signals to a circuit containing a lamp's power cable connected to a device ('lamp box') connected to the outlet in different parts of the house and use a relay to control the power from the outlet to the lamp. It will be made so that specific lights can be turned on throughout different parts in the house using voice commands. We will use Bluetooth to connect the main station box to each of the lamp boxes.

1.3 Visual Aid

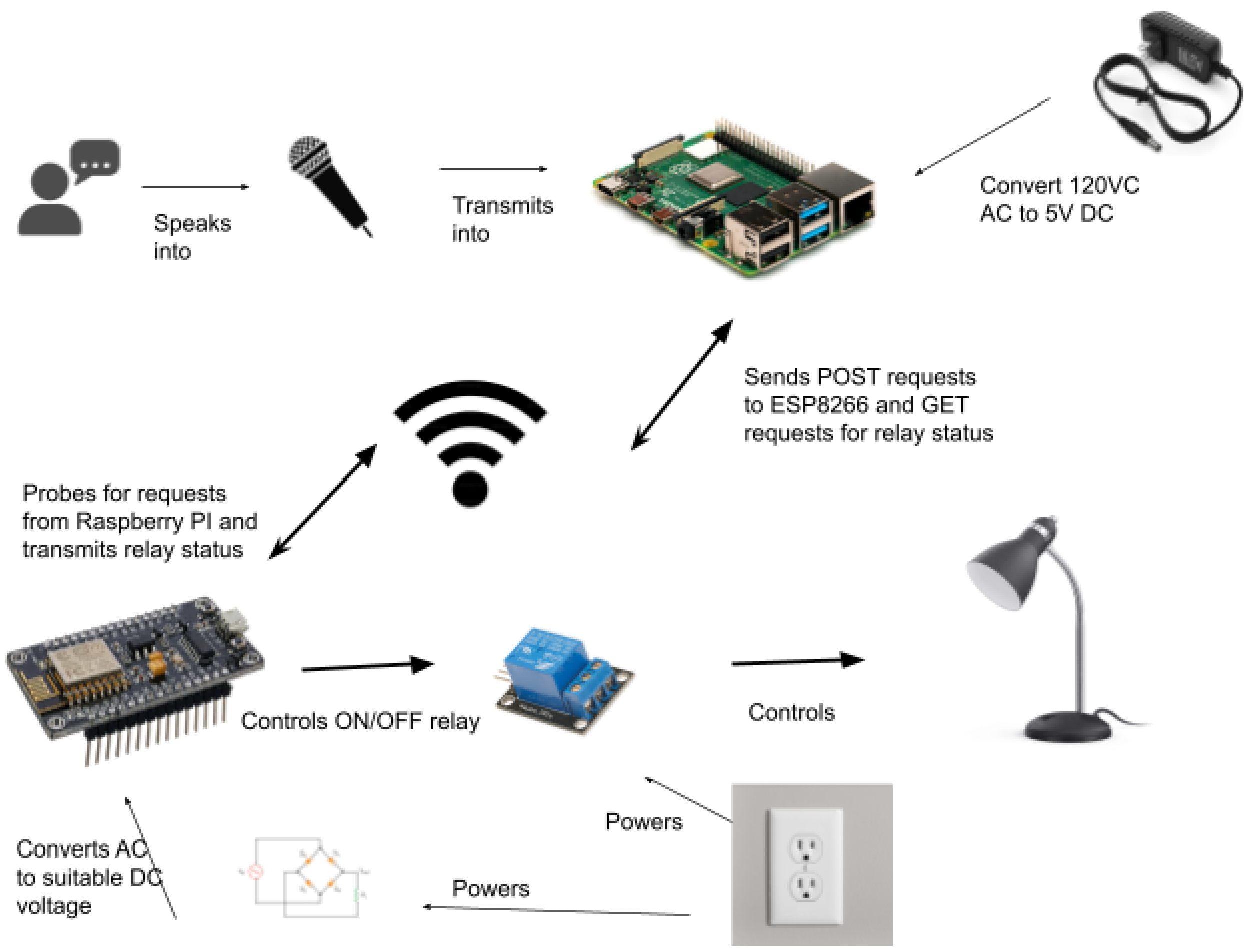

Figure 1: High Level Overview of our System

We aim to make our project as simplistic as possible so that it will be easy for people to use. A person will speak, which our main station box will try to pick up. Their voice will then be processed into a command in our Main Station box, and if the command is valid, it will send the corresponding signal to the corresponding lamp box, allowing the user to turn the specific outlets on or off using only their voice.

1.4 High Level Requirements:

To consider our project successful, we must complete the following:

1. Our system is able to recognize 6 commands with 75% accuracy on our three different voices facing the microphone from a maximum of 1 meter from the main station box. The commands will be “Outlet 1 On”, “Outlet 1 Off”, “Outlet 2 On”, “Outlet 2 Off”, “All Outlets On”, and “All Outlets Off”.
2. Supports up to two independent lamp boxes that can communicate with the main station box from a maximum of 5 meters away.
3. Visual feedback from the lamp boxes will be displayed within 5 seconds of the microphone picking up the command.

2 Design

2.1 Physical Design

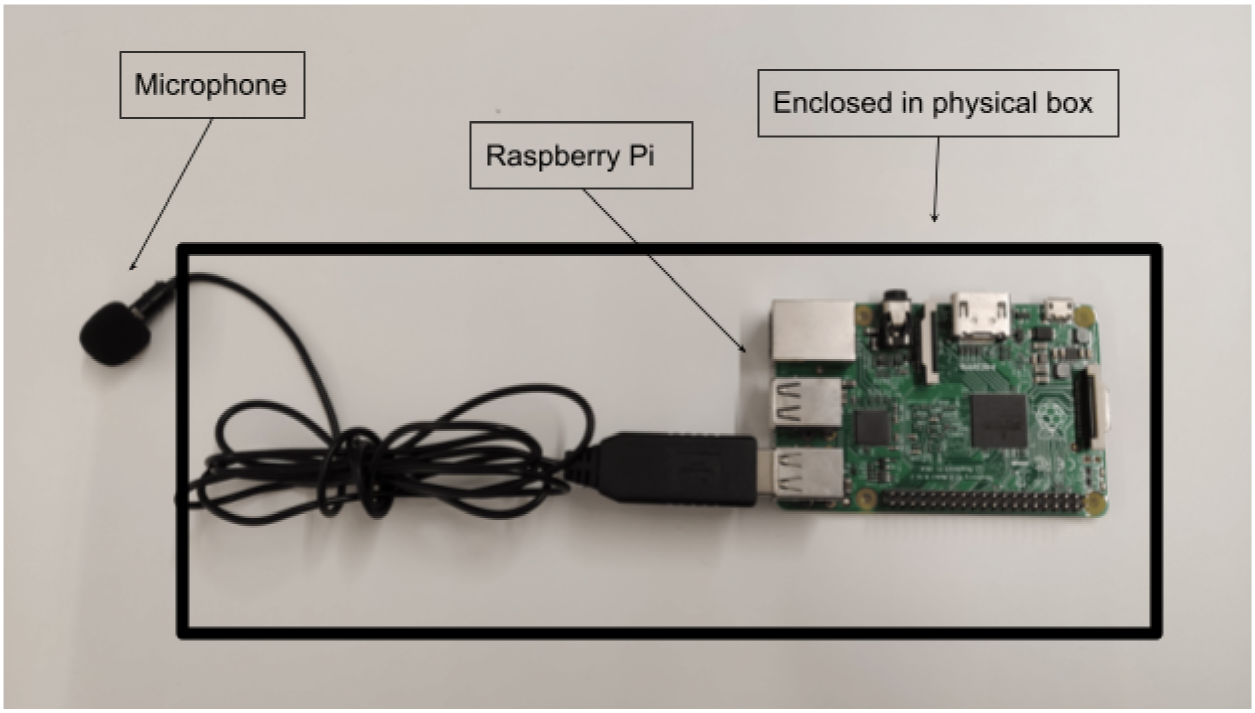

Figure 2: Physical Design of Main Station Box

The components in Figure whatever will be enclosed in a box that has an opening for the microphone to clearly hear the speaker.

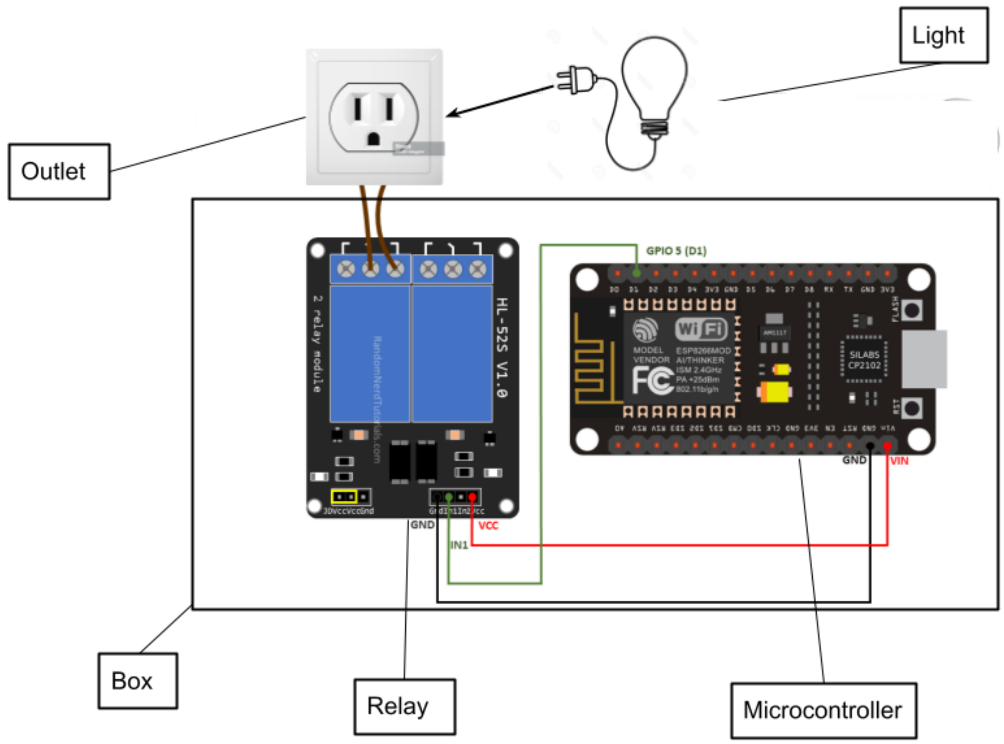

Figure 3: Physical Design of a Lamp Box

2.2 Block Diagram

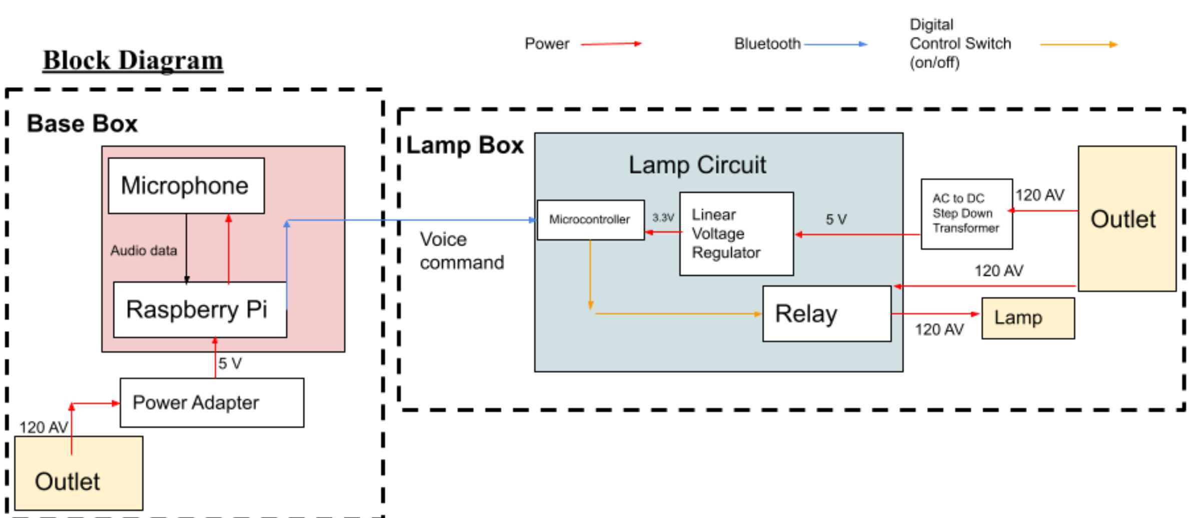

Figure 4: Block Diagram

2.2 Functional Overview & Block Diagram Requirements

2.2.1 Main Station Box

The role of the main station is to perform 2 core functions: First, it needs to communicate with the other ESP32-E modules. It will need to probe the ESP32-E modules to receive current status on the relay, whether the relay is on or off. It will also need to be able to send data to the ESP32-E in order to perform a request. Second, the Raspberry Pi will house a microphone to pick up any voice commands, and then the microphone will send data to the Raspberry Pi in order for it to perform voice recognition. This will also contain the voice recognition method in the Raspberry Pi.

Table 1: Main Station Box - Requirements & Verification

Requirement	Verification
The Raspberry Pi will be able to power and communicate the microphone via the USB Protocol.	● Plug the microphone into the Raspberry Pi ● Ensure the light on microphone that shows it is powered is on ● Check that the microphone is detected by the Raspberry Pi
There will be a cooldown of 4 ± 1 seconds between each successful voice command recognized.	● Monitor the text data in Raspberry Pi that was converted from the audio data gathered ● Repeatedly say a command at 1 second intervals ● Make sure that it is processing every 5th time the command is set
Our microphone will detect voices between 20dB to 60dB, which is the average sound intensity of speaking up to one meter away.	● Monitor the text data in Raspberry Pi that was converted from the audio data gathered ● Say a command while 1m away from the microphone ● Repeat Previous Step at .5m away and .1m away ● Ensure the Raspberry Pi picked up the 3 commands
Send commands to ESP32-E using Bluetooth signals on the LAN network using 2.4 GHZ network	● Send a Dummy Signal to ESP32-E 3 times from Raspberry Pi ● Make sure the ESP32-E receives the command 3 times using the programmer

2.2.2 Voice Recognition Method

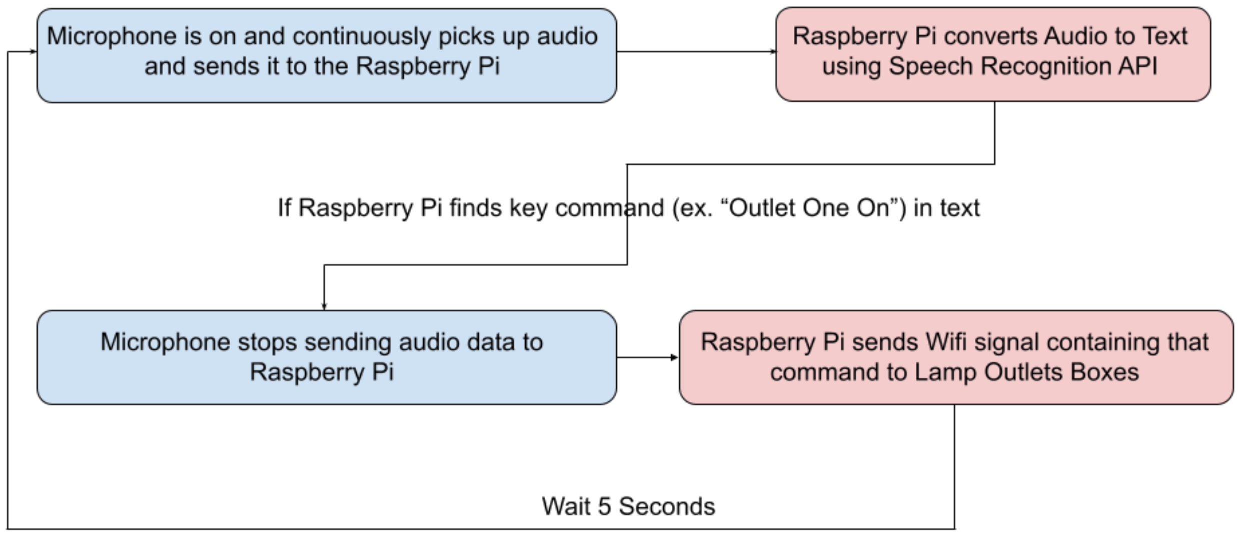

Figure 5: Speech Recognition FlowChart

We will be using the Vosk API to translate the sound data the microphone picks up into text data in the Raspberry Pi. Vosk utilizes Hidden Markov Models and Subspace Gaussian Models to build their speech recognition model on pre trained data [1], allowing us to reach up to 75% accuracy [2]. Vosk is trained on the LibriSpeech Dataset, which is a collection of 1000 hours of 16kHz read English speech prepared by Vassil Panayotov and Daniel Povey [3]. The VOSK API is available for open-source use online [4]. This software will be in the Raspberry Pi and will use the microphone from the main station box.

Table 2: Voice Recognition- Requirements & Verification

Requirement	Verification
Perform voice recognition using Raspberry Pi and the Vosk API with 75% accuracy.	• Test the voice recognition model using the same consistent commands using the same voice (“Outlet 1 On”, “Outlet 1 Off”, “Outlet 2 On”, “Outlet 2 Off”, “All Outlets On”, and “All Outlets Off”) • Ensure that each command will be recognized at least 3 out of 4 times

Can perform voice recognition of the six commands (listed in high-level requirements) using our three different voices.	● Test the voice recognition model with three different voices and the same consistent commands ("Outlet 1 On", "Outlet 1 Off", "Outlet 2 On", "Outlet 2 Off", "All Outlets On", and "All Outlets Off") ● Ensures that each command will be recognized at least 3 out of 4 times

2.2.3 Lamp Box

The system will use the ESP-32E microcontroller [5] to receive voice commands from the main station box. The lamp will be plugged into the lamp box. The lamp box is connected to an outlet which will power any necessary functions it needs to perform. It will also perform any necessary voltage conversion needed to power the ESP-32E. Upon receiving a command, the microcontroller will activate or deactivate the relay, thereby controlling the electrical flow to the lamp, effectively turning it on or off. We will be using a phone charger to take 120V from an outlet and convert it into 5V. Our linear regulator circuit will take the 5V input and convert it into 3.3V DC.

Table 3: Lamp Box Subsystem - Requirements & Verification

Requirement	Verification
Our Linear Regulator circuit will take 5V input DC and convert it into a 3.3V DC in order to power the ESP32-E	● Using a Multimeter with one end connected to ground, measure the input and the output voltages. ● Verify the input is 5V and the output is 3.3V
Receive signals from Raspberry Pi to turn Relay on/off	● Send a signal to turn the relay on and off with the Raspberry Pi ● Hear a clicking sound for confirmation
The PCB should be able to withstand 120V AC without the PCB generating too much resistance, thus generating heat	● Test to see the resistance of the 120V power lines in PCB can withstand the load of 120V Mains by measuring its resistivity with a multimeter ● Use a thermistor in order to measure the heat on the PCB and make sure that the temperature is not too high to disrupt circuit function

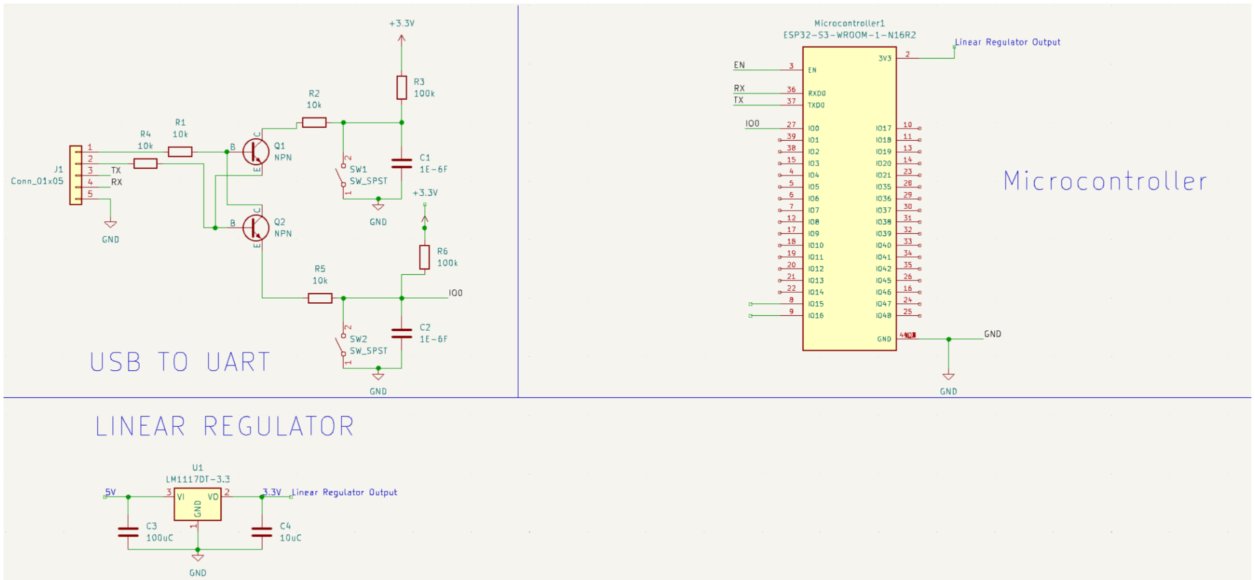

Figure 6: Lamp Box Schematic

2.4 Tolerance Analysis

We need to make sure our voice recognition will recognize someone speaking at 60 dB, the normal volume for human conversations. Our microphone should be able to pick up commands issued from a meter away.

To find the intensity of a noise a certain distance away, we can use the sound density equation (eq 1) to see how loud a sound would be a certain distance away

$$\text{(eq. 1) } I = \frac{P}{4\pi r^2}$$

We can see that the sound level will decrease exponentially as we get further away. If we are given the intensity and distance away, we can use that as a reference to calculate the intensity of that same noise at a different distance away as well. (eq. 2)

$$\text{(eq. 2) } \frac{I_1}{I_2} = \frac{r_2^2}{r_1^2}$$

Decibels are a relative measure of sound intensity, so we can't plug them directly into eq 2. Instead we will use another equation (eq 3) to find the different intensity at different distances of the same sound.

$$\text{(eq. 3) } I(db)_2 = I(db)_1 - 20log(\frac{r_2}{r_1})$$

For example, if a voice's intensity is picked up as 60 dB from 1 cm away, then that would have an intensity of 20 dB when it is 1 m away. We need to make sure to make sure our microphone can pick up a range of sound intensities from 20dB to 60dB, so we either need a microphone that has a sensitivity that can operate from those ranges or we need an amplifier to match the decibel levels at different distances.

We also need to make sure our microcontroller has a source of power. The datasheet for the ESP-32 says that the maximum voltage it can accept as for the power supply is 3.6V, which is lower than the 120V and 5V power sources our circuit is connected to. To accommodate for this, we used our linear regulator to lower the supply voltage coming into the ESP-32 down to 3.3V, which is within the tolerance for it.

3 Cost and Schedule

3.1 Cost Analysis

Labor Costs (Per Partner)

($50/hr) * 2.5 * 12 hrs = $1,500 per week

(Total for 15 weeks: $22,500)

Parts

Table 4: List of Components and Costs

Part Name	**Quantity**	**Cost**
Raspberry Pi 4 Model B (2019)	1	$68
Raspberry Pi 4 Power Adapter	1	$10

Raspberry Pi 4 MicroSD Card	1	$10
Microphone for Raspberry Pi	1	$25
ESP-32E Microcontroller	2	(Via ECE Supply Store)
Relay Modules	2	$8 (Dual Pack)
Outlet Socket	2	$4 x 2 = $8
Resistor	6	(Via ECE Supply Store)
Capacitor	4	(Via ECE Supply Store)
Transistor	2	(Via ECE Supply Store)
Switch	2	(Via ECE Supply Store)
Phone Charger	1	$10
Linear Regulator	1	(Via ECE Supply Store)
Total Cost		**$146**

3.2 Schedule

Table 5: Project Schedule

Week	Task	Person
September 24 - 30	Order Raspberry Pi	Walter Tang
	Start Designing PCB	Anshul Goswami
October 1st - 7th	Design Review	Everyone
	PCB Review	Everyone
	Speak with Machine Shop	Anish Naik
October 8th - 14th	First Round PCB Orders	Walter Tang
	Teamwork Evaluation	Everyone
	Final Machine Shop Revisions	Everyone
	PCB Revisions	Anish Naik
	Start working with Raspberry Pi	Anshul Goswami

October 15th - 21st	Second Round PCB Orders	Anish Naik
	PCB Revisions	Anshul Goswami
	Get Parts	Everyone
	Start Testing Speech Recognition API	Walter Tang
October 22nd - 28th	Pass Audit	Everyone
	Individual Team Progress Report	Everyone
	Third Round PCB Order	Anshul Goswami
	Start Assembling	Anish Naik
	Work with Microcontrollers	Walter Tang
October 29th - November 4th	PCB Revisions	Anshul Goswami
	Continue Assembling	Anish Naik
	Start Testing	Everyone
November 5th - 11th	Continue Testing	Everyone
November 12th - 18th	Mock Demo	Everyone
	Fix Bugs	Everyone
November 19th - 25th	Fall Break	N/A
November 26th - December 2nd	Final Demo	Everyone
December 3rd - 9th	Final Presentation	Everyone

4 Ethics and Safety

In terms of ethics, we followed the IEEE Code of Ethics [6]. As engineers, we know that technologies have the ability to affect someone's life and we hold ourselves to the highest ethical standard when working professionally including but not limited to:

1. **Protect the safety and privacy of the public** [6]

IEEE Code of Ethics states “to respect and protect the privacy of individuals.” Our voice recognition system will not store any data offsite to ensure data privacy of individuals.

2. **Ensure that we treat each team member with respect and ensure that each person on the team follows the code** [6]

   In order to establish good communication between the team and TA, we established a Discord server to communicate meetings and deadlines. With our GitLab repository, we store a journal detailing each member’s work to not only keep track of technical details but also attributing contributions correctly.

3. **Consistently learning and improving our abilities during the process** [6]

   The purpose of this class is to give us an opportunity to demonstrate the skills we have learned and apply them to a real world application. We will follow all required protocols, such as the CAD training, as well as consult our TAs and professors to ensure that requirements and deadlines are being met. We will also consistently ask for feedback to ensure that we are doing the best that we can.

In regards to safety regulations:

1. We will ensure that our project follows any relevant license terms of service for all software used.
2. When building any physical materials needed for our project, we will follow the required safety guidelines in the OpenLab to ensure the safety of everyone involved.
3. Since we will be working with AC power, we will ensure that each member completes the High Voltage training and adhere to these guidelines:
   a. Wearing proper safety gear - we will wear protective gear such as safety goggles and insulating gloves when working with the circuit connected to AC 120V.
   b. Before making any adjustments to the circuit, we will ensure that it is completely isolated from the power supply.

References

[1] Asma Trabelsi, Sébastien Warichet, Yassine Aajaoun, Séverine Soussilane,
Evaluation of the efficiency of state-of-the-art Speech Recognition engines,
Procedia Computer Science, Volume 207, 2022, Pages 2242-2252, ISSN 1877-0509,
https://doi.org/10.1016/j.procs.2022.09.534.
(https://www.sciencedirect.com/science/article/pii/S1877050922014338)

[2] P. Setiawan and R. Yusuf, "IoT Device Control with Offline Automatic Speech Recognition on Edge Device," 2022 12th International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 2022, pp. 111-115, doi: 10.1109/ICSET57543.2022.10010962.
https://ieeexplore.ieee.org/document/10010962

[3] Panayotov, Vassil, et al. *LIBRISPEECH: AN ASR CORPUS BASED on PUBLIC DOMAIN AUDIO BOOKS*. https://www.danielpovey.com/files/2015_icassp_librispeech.pdf

[4] “VOSK Offline Speech Recognition API,” *VOSK Offline Speech Recognition API*.
https://alphacephei.com/vosk/

[5] “ESP32-S3-WROOM-1 Datasheet v1.2.” Espressif Systems, 2023
https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[6] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).