MOBILE DEPLOYABLE SMART DOORBELL

ECE 445 DESIGN DOCUMENT

Team # 16

Members: Charles Lai (jiayeyl2), Ricky Chen (pohsuhc2), Victor Lu (vclu2)

TA: Gong, Rui

Professor: Prof. Kejie Fang, Prof. Arne Fliflet

Oct. 3, 2024

Contents

1 Introduction

1.1 Problem & Solution

As a college student living in a dorm/apartment complex, the absence of a doorbell poses inconvenience for both myself and my visitors, such as my friends, neighbors, or anyone who would come to my house. My room is located far from the entrance; therefore, every time they knock on the door, I can't respond promptly.

Moreover, regular doorbells will fail to notify me if I am either too far from the door or there are barriers between. In scenarios where I am taking a shower, on the balcony, taking phone calls, and playing video games, I am unable to hear anything outside of my enclosed environment.

To solve this problem, our team is designing the mobile deplorable smart doorbell. As the name suggests, our project is a small, smart doorbell that can be easily deployed and notify the resident via their phone. The doorbell will be connected to the internet and to the resident's phone. When a visitor presses the doorbell, the resident will be notified via phone, which is almost always with the resident in this technological society. Therefore, the resident can be notified in real-time regardless of where they are. This solves the issue of not having the need to hear "physical" sound from a regular doorbell located outside the door. With phones being present at almost all times during our everyday's lives, the resident would be notified via real time, bypassing anything in between.

Furthermore, our doorbell will support a variety of features such as voicemail and video recording. The resident can respond to their visitors despite not being at home and in many different circumstances.

1.2 Visual aid

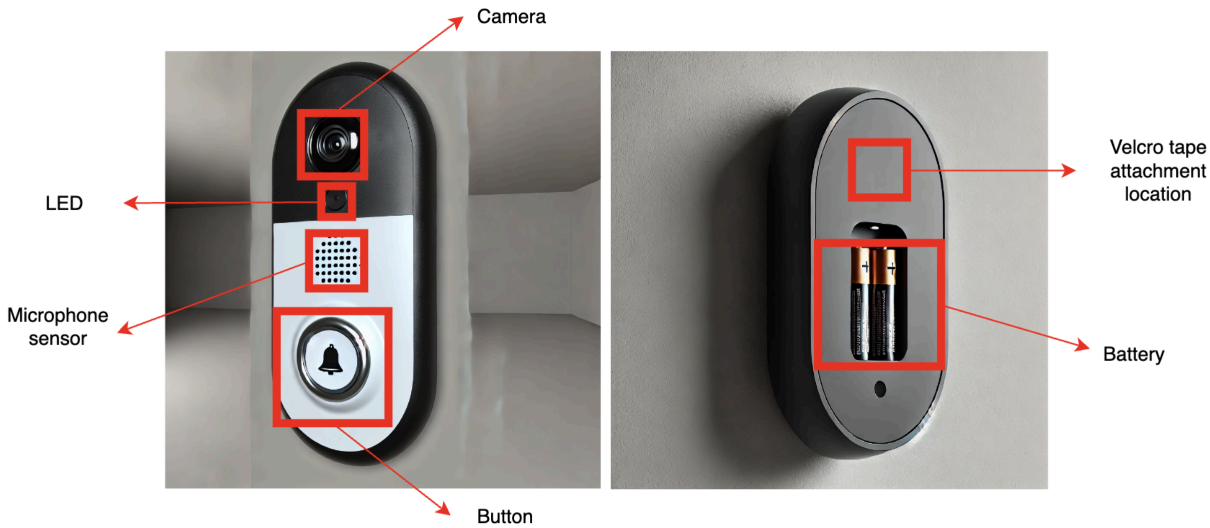

Figure 1: Doorbell Appearance

The final product of the doorbell would look similar to the visual aid above. The doorbell's interface would be easy and intuitive to use, consisting of a camera, LED, microphone sensor, and button. Upon arrival, the visitor would press the button on the doorbell. The button would then trigger all of the other features on the interface. The camera would be turned on and start recording for 5 seconds. Additionally, the voice sensor would also be triggered and start recording voice mail. Finally, the LED would be lit up to orange, signifying that the button is pressed, camera is recording, and the voice sensor is on. Additionally, one of the LEDs is used to ensure internet connection. If there is stable internet connection, one of the LEDs would be lit up as green; otherwise, it would be red. On the back side of the doorbell, there will be an open battery slot for 2 AA batteries, with a plastic lid. Moreover, there would be an empty space for velcro tape that is used for the doorbell's attachment on the wall. This allows easy replacement of batteries as well as painless deployment of the doorbell. A side note to velcro tape: it is a double sided tape that can be stripped on and off very easily while retaining a very strong tension. A single pair of velcro tape can support a maximum of 16 pounds, which is sufficient for a doorbell. Moreover, it is relatively inexpensive and can be very flexible when deploying.

1.3 High Level Requirements

1.3.1 Audio and Video Transmission Quality

Internet transmission will be one of the important features of our device. We hope to achieve grayscale video streaming with the resolution of 1280*720p and with 12 frames per second. As for audio, we aim to have the capability of transmitting audio with an 8kHz sampling rate and 8-bit bit depth.

1.3.2 Phone App Latency

In order to visually show the video data and play audio in real time, our design aims to achieve a data latency of less than 1 second.

1.3.3 Weight

To enable convenient deployment and carrying of our project, the weight of the whole module needs to be light. The project, including enclosure and batteries, will not weigh over 2 kilograms.

2 Design

2.1 Block Diagram

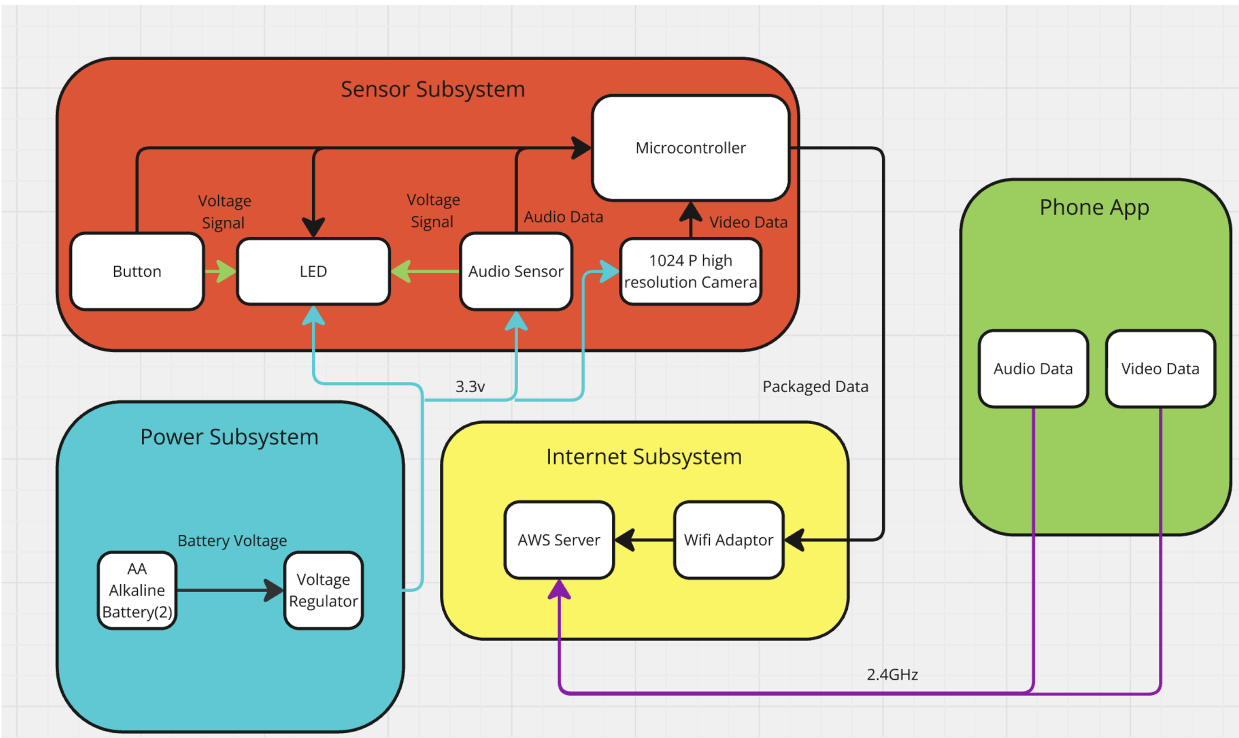

Figure 2: Block Diagram

2.2 Physical Design

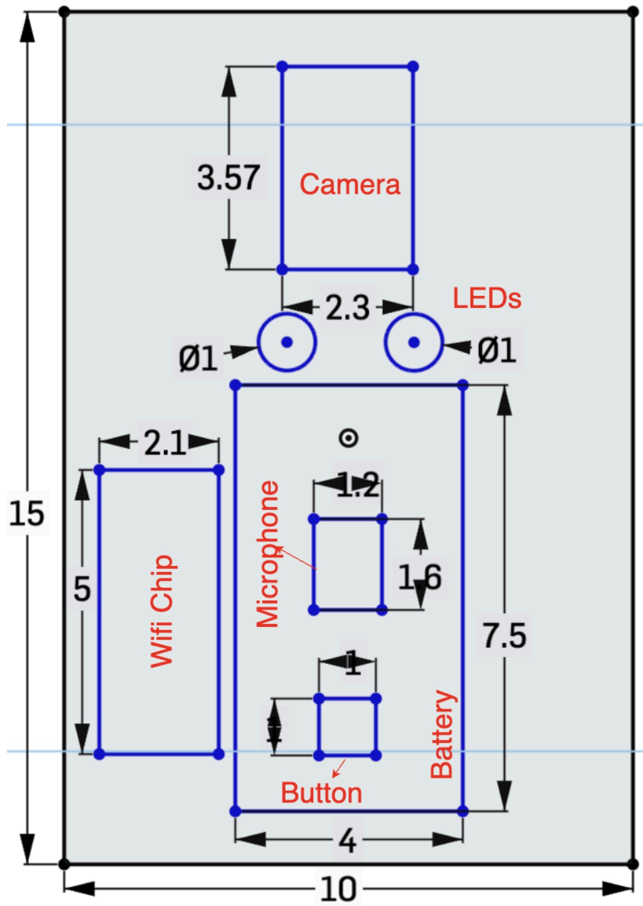

Figure 3: Physical dimensions of doorbell and its components

The figure showcases the mechanical dimensions of the doorbell components. The overall dimensions of the doorbell will be 15 cm in height, 10 cm in width, and 3 cm in depth. For the components, the battery will take up the largest space, measured at 7.5 cm in height, 4 cm in width, and 1.9 cm in depth. Therefore, the battery will be located at the center. Additionally, the centered location of the battery also allows easier replacement and access to batteries as well as better heat dissipation. Next, the wifi chip is the second largest component, measured at 5 cm in height, 2.1 cm in width, and 0.5 cm in depth. The chip will be located next to the battery since both components will be located at the back of the doorbell. On the other hand, more user-accessible components are button, microphone, LEDs, and camera are located in the front.

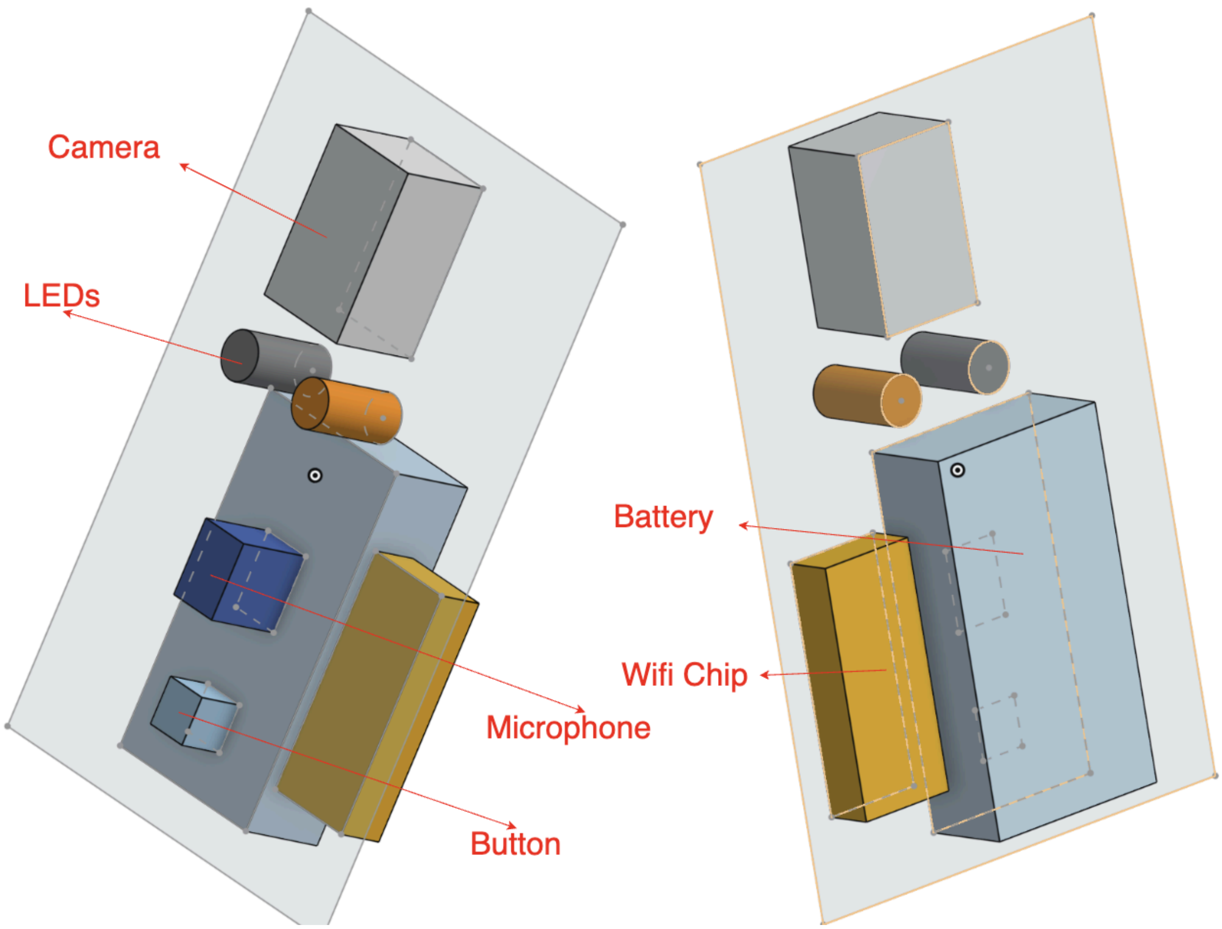

Figure 4: 3D model of Doorbell

The 3D model above is a simple illustration of the doorbell, where the camera, LEDs, microphone, and button are on one side. On the other side, there is a battery and wifi chip. For additional components, such as microcontrollers, it will go to the back side where the battery is on. This design allows a clear separation of components that are user accessible and those that shouldn't, avoiding potential interference of the user with components that shouldn't be interacted with. Moreover, this design allows easy management, where each component could be easily inspected and replaced if necessary.

2.3 Subsystem Overview and Requirements

We divided our product into four subsystems: Sensor subsystem, Power Subsystem, Internet Subsystem, and a Phone App. Power subsystem provides a direct current power to the sensor and the WiFi adapter in the internet subsystem. The sensor subsystem accepts inputs from a person, including pressing the button and recording video and audio. These data are passed to the internet subsystem and the adaptor sends these data to the AWS server. Meanwhile, the Phone App side will constantly retrieve data from the server to check if any new signals are coming, so that it can visualize the video data and play the audio data.

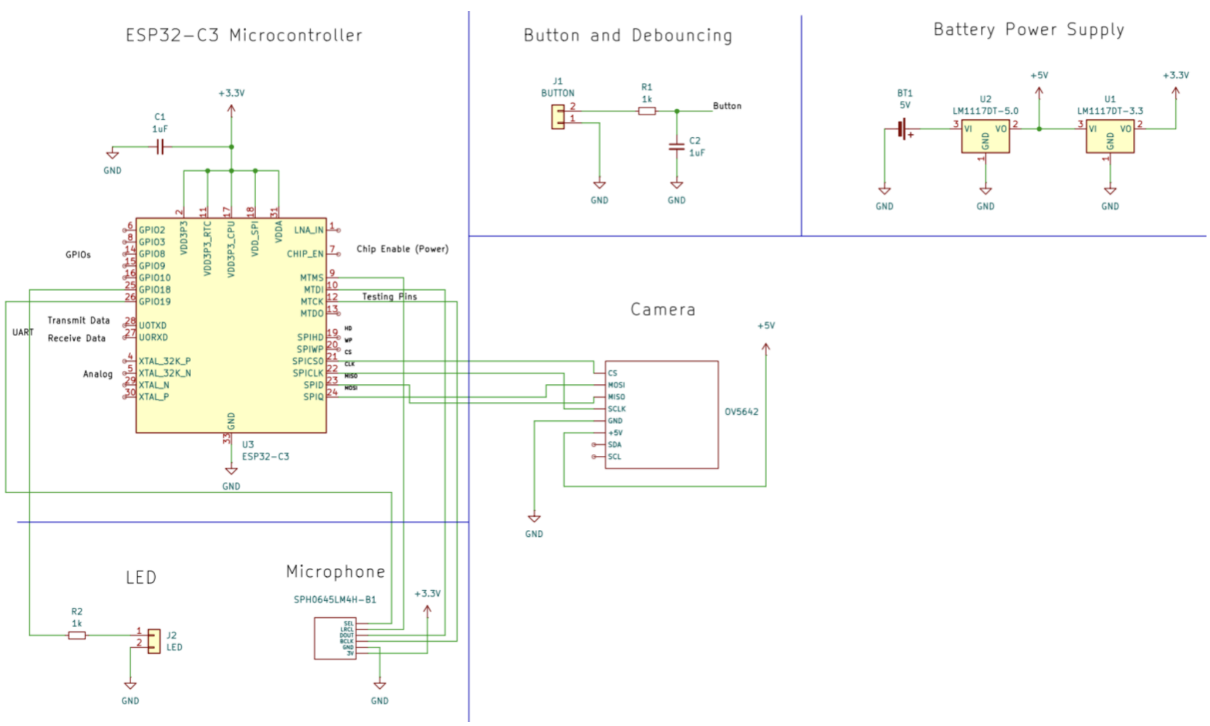

Figure 5: Circuit Schematics for all subsystems

We have drafted a schematic for our device. We have an ESP32-C3 [1] chip acting as a WiFi interface and microcontroller. It supports both SPI and I2S communication protocols so that we can transmit video and audio data respectively. We are also using both LM1117DT-5.0 [2] and LM1117DT-3.3 to regulate our voltage source since some of our components require a 5-volt input and some of them need a 3.3 voltage input. The OV5642 [4], camera module, for example, requires a 5-volt input, so we wire the 5-volt source to its power pin. We also wire the rest of the pins to the SPI interface of ESP32-C3 microcontroller since the camera is using SPI protocol to transmit data. On the other hand, the microphone [3] uses I2S protocol to transmit audio data, and we have to set up special modes in the microcontroller to make it accept I2S inputs. In fact, we have to change some bits in the registers of the ESP32-C3 chip through software, so that we can use ordinary general-purpose input/output (GPI/O) pins as interfaces of I2S protocol transmissions. Lastly, we have a button connected to one of the GPI/O pins to make the microcontroller a hint to interact with the user, and then we can do our programs to interact with the signal of this GPI/O pin, such as turning on the transmission of the camera when the GPI/O pin connected to the button is HIGH. Instead of making some of the components idle and wasting power, we can make real-time interaction to let the components active after the user has provided input.

2.3.1 Sensor Subsystem

We have several sensors in our doorbell: a microphone, a camera, a button, an LED, and a microcontroller. We want our camera to have enough resolution to tell the face of the person answering the door and the audio legible enough to recognize words. Aside from those, the haptics of the button should also be clear enough to hint to the user. Additionally, we want an LED to indicate the microphone is currently recording. Finally, we will need a microcontroller to manage all the components and organize the package. Overall, we strive to look for low-cost, low-power, but effective components.

Requirements	Verification
● When the button is pressed, video streaming will begin and the user will be able to monitor the situation outside the door in real time. The video will also need to be clear enough for the user to identify the person standing outside of his or her door.	● Ensure both the mobile device and the doorbell are connected to Wifi. ● Then, press the button and show the camera numerous pictures of people, who the tester may or may not know. ● Ensure real time video appears on the tester's device. ● Ask the tester to identify the person in each picture on the mobile device ● Then, ask the tester to identify the person in each picture in real world ● Compare the results and verify the error percentage is below 10 percent.

• When the button is pressed, audio recording will begin and the LED will light up. The audio will need to be clear enough for the user to identify the words the person is speaking.	• Ensure both the mobile device of the tester and the doorbell are connected to Wifi. • Then, press the button and ensure the LED is on. • Then, say some sentences into the microphone and ensure audio data appears on the tester's device. • Ask the tester to identify the words received on the mobile device • Then, compare the results with the correct words and verify the error percentage is under 10 percent.
• When the sensor subsystem detects the button has been pressed for an abnormal time (for more than 10 minutes), the microcontroller will send a signal through wifi to notify the user that the button might be stuck.	• Ensure both the mobile device of the tester and the doorbell are connected to Wifi. • Then, tape down the button and start a 10 minutes timer • Once the timer is reached, verify notification from the tester's mobile device.

Table 1: Requirements & Verification of Sensor Subsystem

2.3.2 Internet Subsystem

Internet transmission will be one of the important features of our device. Instead of setting up a server by ourselves, we will have to send our packet through a WiFi chip to a cloud service platform (e.g. AWS, Azure) so that users' phones have access to the audio and video data from the doorbell. We at least have to make the doorbell connect to WiFi and send data to a server with a given IP address. The transmission rate is further discussed in the high-level requirement.

Requirements	Verification
● When the doorbell is not connected to the wifi yet, the red LED will stay on until wifi connection with the server is established.	● Power up the device with inserting batteries in place. ● Verify the red LED is on as the module is not connected to wifi yet. ● Setup the device and connect it to wifi ● Verify the red LED is off ● Verify the connection is valid by pressing the button and check whether our backend server receive visual and audio data from the module
● In the case when a fatal communication error occurs or wifi connection is severed, the microcontroller will actively try to	● Ensure both the mobile device and the doorbell are connected to the internet. ● Ensure the doorbell is linked to the tester's account.

reestablish internet connection.	• Detach the doorbell from the door and bring it outside the valid range of the wifi it is connecting to. • Reattach the doorbell to the door and wait for five minutes for the doorbell to reconnect. • Verify on the tester's mobile application that the doorbell is reconnected with the server.

Table 2: Requirements & Verification of Internet Subsystem

2.3.3 Power Subsystem

We need sufficient power and low-power-consuming components in our doorbell since we don't want the users to change batteries frequently. More specifically, we will try to implement a AA battery and regulate the voltage with a voltage regulator. Our goal is to let it sustain one month without replacing new batteries.

Requirements	Verification
• When the button is released, ensure the camera and microphone is not actively sampling data to minimize power usage and prevent privacy invasion.	• Code the microcontroller such that it automatically records the time it detects microphone and camera module connection. • Then, press the button for 1 minute and release the button for 1 minute. Repeat this process for 10 times. • Stop the program and verify the result with the log given by the microcontroller.
• Ensure a stable current and voltage can be produced from our battery pack and voltage regulator to ensure device safety.	• Assemble the battery pack and the voltage regulator. • Then, insert brand new batteries into the battery pack and power up the

	doorbell ● Measure the voltage and current drawn from our power subsystem for five minutes with a multimeter. ● Within this five minute, periodically press and release the button to enable and disable the camera and microphone. ● Verify the current and voltage are stable throughout this period.

Table 3: Requirements & Verification of Power Subsystem

2.3.4 Phone App

There are several requirements for our phone application. We are aiming to develop an iOS application since iPhone users take a majority portion of people and all team members have an iPhone. Hence, we will design the application in XCode. Firstly, we need to be able to send a notification (a pop-up notification) whenever the button of the doorbell is pressed. To achieve this, we will have to connect the phone with the internet to get a real-time signal from another server. On top of that, we will also have to design a user interface that includes a panel playing the video from the camera and a hint about audio playing.

Requirements	Verification
● In the case when a fatal communication error occurs or wifi connection is severed, the mobile application must notify the user about the situation.	● Ensure both the mobile device and the doorbell are connected to the internet. ● Ensure the doorbell is linked to the tester's account. ● Detach the doorbell from the door and bring it outside the valid range of the wifi it is connecting to. ● Verify on the tester's mobile application that a notification appears regarding this connection error.
● When the button is pressed, a	● Ensure both the mobile device and the

notification is sent to the user's mobile device. Video and audio streaming will also begin on the user's phone application.	doorbell are connected to the internet. ● Ensure the doorbell is linked to the tester's account. ● Press the button on the doorbell ● Verify on the tester's mobile application that a notification appears. Video and audio streaming should also begin on the tester's mobile application.

Table 4: Requirements & Verification of Phone Application Subsystem

2.4 Tolerance Analysis

2.4.1 Required Data Transfer Rate

The major issue of our project falls into internet connection. Streaming video from a microcontroller to a phone takes a lot of bandwidth. Thus, achieving enough bandwidth and stability is our core task of realizing our project.

There are two major types of data we will be transferring extensively through the internet: visual data and audio data. Due to the fact that these two data need to be transferred simultaneously, our data transfer bandwidth has to be large enough to accommodate both live streaming of visual and audio data.

The visual data our module planning to send has the resolution of 720p, which is 1080 * 720 pixels per frame. Moreover, the frames per second value we are aiming to achieve is at least 12. From these data we can calculate the required pixels per second as shown below.

$$pixel\ per\ second = resolution * frames\ per\ second$$

$$= 1080 * 720 * 12 = 9331200$$

For each pixel in a grayscale picture, we will be using 8 bits to store the visual data to ensure the clarity of our video. With this information and the calculation result above, we can yield the final required data transfer rate for image data in bits per second by the following equation.

$$Transfer\ rate\ (bits\ per\ second) = pixel\ per\ second * bits\ per\ pixel$$

$$= 9331200 * 8 = 74649600$$

Now we convert this number into megabytes per second (MBps) for better readability. The conversion is as shown below.

$$MBps \;=\; Kilobytes\ per\ second\ (KBps)\ /\ 1024$$
$$=\; Bytes\ per\ second\ (BPS)\ /\ (1024\ *\ 1024)$$
$$=\; Bits\ per\ second\ /\ (1024\ *\ 1024\ *\ 8)$$
$$=\; Bits\ per\ second\ /\ (8388608)$$

Our data transfer rate for streaming grayscale video would be calculated as shown.

$$9331200\ /\ 8388608 \;=\; 1.1124\ MBps \quad \text{————————————} \ (1)$$

In terms of our audio interface, we want to have our resolution and sampling rate to be: 8-bit depth and an 8KHz sampling rate. We aim to have 8-bit depth data because this data format suppresses unnecessary noises. In fact, higher resolution of audio data doesn't guarantee a clear sound since the data contains too many unnecessary details. These inaudible details will take up redundant bandwidth during data transmission. Therefore, for efficiency reasons, we strive to achieve a 8-bit data format. On the other hand, we set our initial goal of the sampling rate as 8KHz so that the audio frequency covers the frequency of sound an ordinary human can perceive. According to the Nyquist theorem, the highest frequency of an 8kHz sampler is 4kHz of sound, which is sufficient for a lot of conditions, since most people's voices fall in the range of 90Hz to 255Hz [5], and although humans can sense sound ranges from 20Hz to 20,000Hz, the range where most people can perceive the most is 30Hz to 3,400Hz. Thus, sampling in 8kHz strikes the balance between human perception and transmission of data.

From these data we can calculate the required bit per second as shown below.

$$bit\ per\ second\ =\ data\ per\ sample\ *\ sampling\ rate$$
$$=\ 8\ *\ 8000\ =\ 64000$$

Now we convert this number into megabytes per second (MBps) for better readability. The conversion is already shown above during the visual data conversion. Thus, our data transfer rate for audio can be calculated by the equation below.

$$64000/\ 8388608\ =\ 0.0076\ MBps \quad \text{————————————} (2)$$

Combining the result yielded from (1) and (2), we can finally obtain the required maximum data transfer rate for the whole module.

$$1.1124\ +\ 0.0076\ =\ 1.1200\ MBps$$

From the datasheet, our ESP32-C3 chip has a maximum theoretical bandwidth of 150 MBps [1]. Although it is almost impossible to achieve the theoretical maximum and the actual bandwidth will be way lower due to internet connection stability, we can still safely assume our internet bandwidth is sufficient to handle the data transfer from our sensor module.

3 Cost & Schedule

3.1 Cost Analysis

The total cost of components purchased is $55.25 as shown in Table 5. The estimated hours it takes to complete the project is 100 hours for each team member. From the salary data gathered on the website of the Grainger College of Engineering, the hourly pay for ECE graduates is approximately $35/hour. Thus, the labor cost of the whole project is calculated as shown below:

$$\$35/hour \ * \ 100\, hour \ * \ 3\, team\, members \ = \$10500$$

Combining this number with the cost for components, we can obtain the total cost of $10555.25 for this project.

Component Name	Quantity	Price	Link
Arducam 5MP Plus OV5642 Mini Camera Module	1	$42.95	Link
Adafruit I2S MEMS Microphone Breakout - SPH0645LM4H	1	$6.95	Link
ESP32-C3	1	$1	Link
Voltage Regulator - 3.3V	1	$2.1	Link
Battery holder	1	$1.36	Link
Single Velcro Tape	1	$0.89	Link
Total		$55.25	

Table 5: List of Component Purchased

3.2 Schedule

Date	Task	Responsible Member
October 7th - October 11th	Order parts for project assembling	everyone
	Start PCB design	Ricky
	Setup amazon server	Victor
	Begin mobile app development	Charles
October 14th - October 18th	**PCB 1st ORDER OCTOBER 15**	everyone
	Revise PCB design	Ricky
	Achieve stabilized power output	Ricky
October 21th - October 25th	**PCB 2nd ORDER OCTOBER 22**	everyone
	Revise PCB design	Ricky
	Finalize enclosure design	Victor
	Establish data transmission with on board chip via wifi	Charles
October 28th - November 1st	**PCB 3rd ORDER OCTOBER 29**	everyone
	Revise PCB design	Ricky
	Debug mobile application	Charles
	Debug microcontroller logic	Victor
November 4th - November 8th	**PCB 4th ORDER NOVEMBER 5**	everyone
	Revise PCB design	Victor
	Finish mobile application development	Charles
November 11th - November 15th	**PCB FINAL ORDER NOVEMBER 12**	everyone
	Finish assembling every module	everyone
	Extensively test individual module functionality	everyone

	Integrating all modules	everyone
November 18th - November 22th	Finalize project	everyone
	Mock demo	everyone
December 2nd - December 6th	Final Demo	everyone
	Mock Presentation	everyone
December 9th - December 13th	Final presentation	everyone
	Final paper submission	everyone

Table 6: Schedule and Deadline for Project

4 Ethic & Safety

Our group's action complies with IEEE Code of Ethics adopted by the IEEE Board of Directors and incorporating revisions through June 2020. As the advancement of technology can greatly alter one's life, all teammates agreed to hold themselves to the highest ethical standard during the development of the project. These includes but not limited to the following points:

4.1 Treating all persons with fairness and respect

As specified by the code of Ethic, it is important "[t]o treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others."(*IEEE*) In order to achieve this, our group will hold a weekly meeting reflecting on our collaboration and communication method to ensure that all teammates are treated with fairness and respect. Moreover, we will also make sure all opinions from all people are evaluated without bias.

4.2 Ensure IEEE Code is upheld

To ensure the IEEE Code of Ethics is upheld and complied by all team members (*IEEE*), we have a special channel in our discord group dedicated to reporting any violations against the code anonymously. A special meeting will be held once any violations are reported, and our team will take immediate action to amend the violation.

4.3 Safety

Here are some safety concerns of utilizing our doorbell both in the aspect of hardware and software. Each aspect of safety concern will be addressed in detail in the following paragraphs.

4.3.1 Hardware Safety

To prevent other people from installing malicious programs onto our microcontroller, we are enclosing the entire structure with a 3D-printed case. We will also let the users know to not use a damaged doorbell to minimize the risk of using a doorbell with different settings. In terms of battery safety, we are using dry batteries with a voltage regulator to minimize the risk of fire and damage to both the module and other property.

4.3.2 User Security and Privacy

In order to protect against malicious attacks from other people and prevent data leak of the customers, we choose to send data from the doorbell to a server and read data in the server at users' ends. This method doesn't expose the user's phone IP address and our data encoding method will prevent data leakage.

5 References

IEEE - IEEE Code of Ethics, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 19 Sep. 2024.

[1] *ESP32-C3 Series*, Espressif, 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf

[2] *LM1117 800mA, Low-Dropout Linear Regulator*, Texas Instruments, 2023. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm1117.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1721700739746

[3] SPH0645LM4H-B, I2S Output Digital Microphone, Adafruit, 2015. [Online]. Available: https://cdn-shop.adafruit.com/product-files/3421/i2S+Datasheet.PDF

[4] *OV5642 Camera Shield for Arduino*, ArduCam, 2016. [Online]. Available: https://www.uctronics.com/download/Amazon/B0068.pdf

[5] R. J. Baken, *Clinical Measurement of Speech and Voice*, 2nd ed. London, U.K.: Taylor and Francis Ltd., 2000, p. 177, ISBN: 1-5659-3869-0, citing J. L. Fitch and A. Holbrook, "Modal fundamental frequency of young adults," *Arch. Otolaryngol.*, vol. 92, pp. 379-382, 1970.