ECE 445

SENIOR DESIGN LABORATORY
PROJECT PROPOSAL

MOBILE DEPLOYABLE SMART DOORBELL

Team No. 16

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

TA: Gong, Rui

September 19, 2024

1 Introduction

1.1 Problem

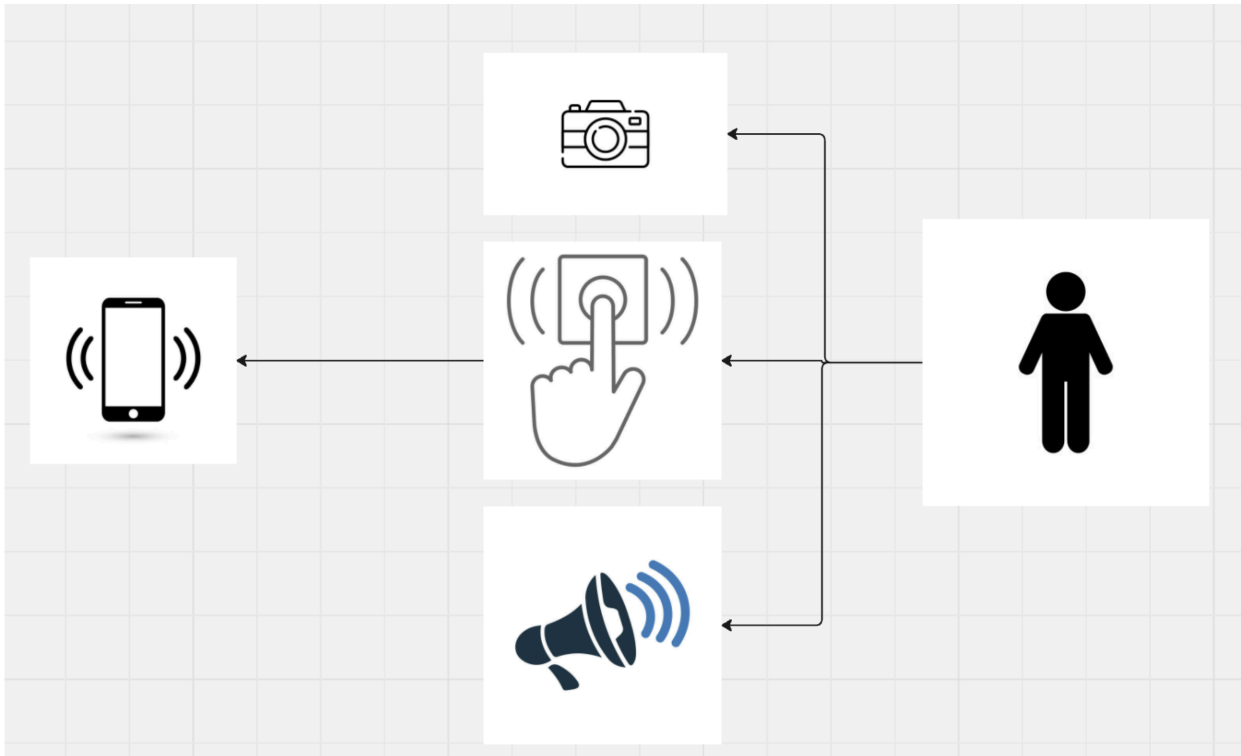
As a college student living in a dorm/apartment complex, the absence of a doorbell poses inconvenience for both myself and my visitors, such as my friends, neighbors, or anyone who would come to my house. My room is located far from the entrance; therefore, every time they knock on the door, I can't 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, at the balcony, taking phone calls, and playing video games, I am unable to hear anything outside of my enclosed environment.

1.2 Solution

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.



1.3 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

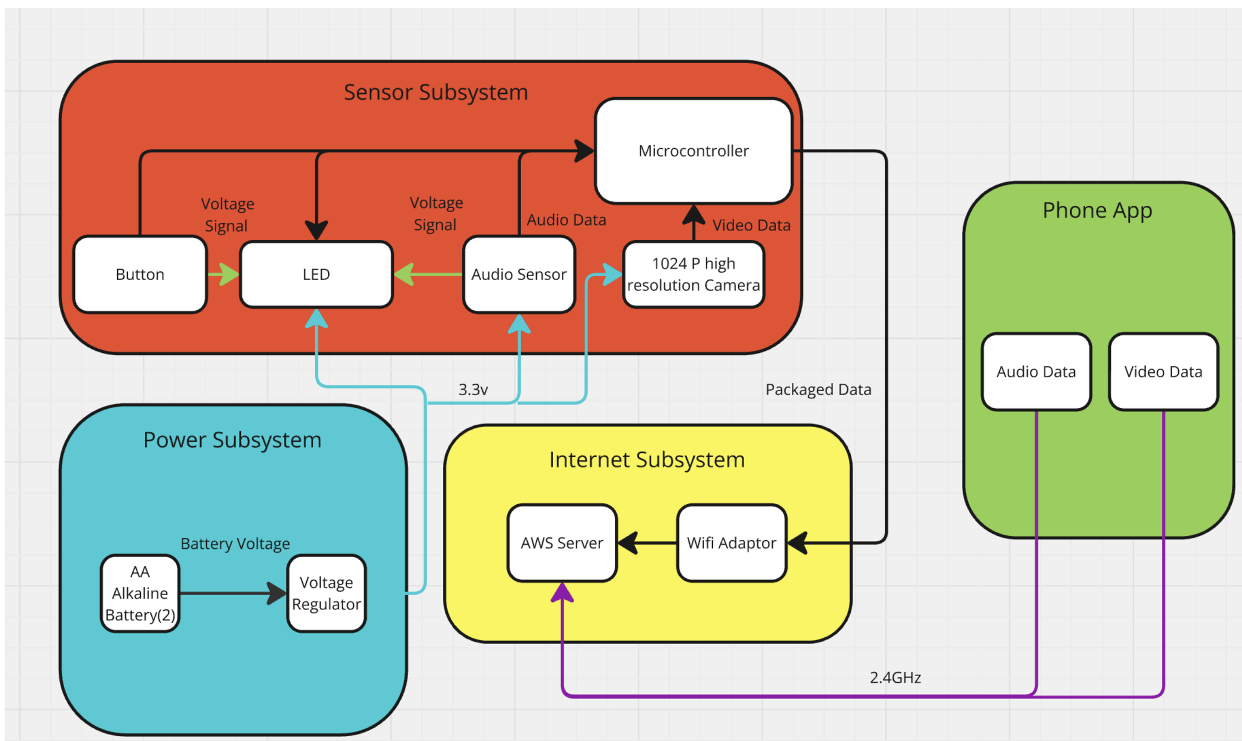
In order to visually show the video data and play audio in real time, our design aims to achieve a data latency of less than 1 second.

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



2.2 Subsystem Overview

2.2.1 Sensor Subsystem

This subsystem will serve as our main module for data collection. It contains a camera, a microphone, and a button to collect visual and audio data as well as accepting user input. When the button is pressed, the camera and the microphone will start recording and the collected data will be sent to our phone application via the internet subsystem in real time. Furthermore, an LED will also lit up when the button is pressed to inform the user that recording is in progress.

2.2.2 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 an AWS remote server so that users' phones have access to the audio and video data from the doorbell. More specially, we will use an ESP32-C3 chip to receive and transmit data via WiFi. We are also using this chip as a microcontroller and connecting it with other sensors, including the microphone, camera, button, and LED. With all the components connected to the chip with their specific communication protocol, we can program the logic around all the connected sensors. We at least have to make the ESP32-C3 chip 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.

2.2.3 Power Subsystem

The power subsystem will be the module that delivers consistent current and voltage to both the sensor subsystem and internet subsystem. It consists of a battery pack for AA batteries and a voltage regulator with a feedback circuit. The voltage regulator and circuitry of the power subsystem will ensure that the other subsystem receives the desired voltage as well as reduce voltage ripple produced by the batteries. Thus, the power efficiency and the safety of our design can be guaranteed.

2.2.4 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. The phone app will be primarily be connected to the wifi chip located in the doorbell, where it handles the data transmission. The interface will be intuitive and easy to use, where all the functions of the doorbell will be displayed on the same screen. Each of the functions will include zoom in and zoom functionalities. Moreover, the app should be available via browser download.

2.3 Subsystem Requirements

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.

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.

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.

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.

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.

$$\begin{aligned} pixel\ per\ second &= resolution * frames\ per\ second \\ &= 1080 * 720 * 12 = 9331200 \end{aligned}$$

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.

$$64000/\ 8388608\ =\ 0.0076\ MBps \qquad (2)$$

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 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:

3.1 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.

3.2 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.

3.3 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.

3.3.1 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.

3.3.2 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.

4 References

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