DOOR KNOCKING ALARM FOR THE HEARING IMPAIRED

By

Ajay Jayaraman

Pax Kim

Jiyoon Lee

Final Report for ECE 445, Senior Design, Fall 2023

TA: Tianxiang Zheng

Professor: Victoria Shao

06 December 2023

Team 21

Abstract

This project presents a door-knocking alarm designed to help the deaf and hearing impaired with identifying when someone is knocking on their door. The device is designed to be mounted on the inside of a door and notifies the user through an LED and a WhatsApp notification when someone is knocking on the door. The report details the development process, design, part specification, and final performance of this device. Results indicate a device that can detect door knocks over a reasonable period was able to be created using a development board, piezoelectric sensor, and other peripheral components.

Contents

1. Introduction

1.1 Problem

People who are deaf or hard-of-hearing face unique challenges in their daily lives, one of which being the difficulty in discerning when someone is knocking at their door. This can lead to inconveniences like important notifications being missed and frustration for both parties, but this problem can also be an important safety concern.

While there are plenty of alarms on the market that show some visual indicator (such as a light) when a doorbell is rung, there is a gap in the market for visual alarms for people with hearing impairments who do not have doorbells. For example, people living in dorms, apartment units (especially across college campuses), and individual office rooms often don't have doorbells installed. These spaces are often rented, and therefore permanent installations are less feasible. Thus, there is a compelling need for a versatile and non-intrusive visual alert system tailored to these diverse environments, one that seamlessly bridges the communication gap and enhances the daily lives of individuals with hearing impairments.

1.2 Solution

We propose a device that can detect when someone is knocking and sends an alert. The design will use batteries for its power supply and will be easily attachable to most doors. Running off batteries will make it more accessible as not all doors have an electrical outlet readily available.

The Piezoelectric sensor will specifically be aiming to detect vibrations through the door. The differentiation between knocking vibrations and vibrations that may cause vibrations through the door (for example, lots of people running past the door in a dorm setting) will be determined by testing a range of vibrations classifiable as knocking. After the alarm is triggered, it will then send an alert to the user's phone and emit a bright light.

1.3 High Level Requirements

1. The device must be able to detect when someone is knocking on the door.
2. The device must be able to reliably communicate with the user when it detects a door knock.
3. The device should be able to last reliably for the duration of 10 hours.

1.4 Visual Aid

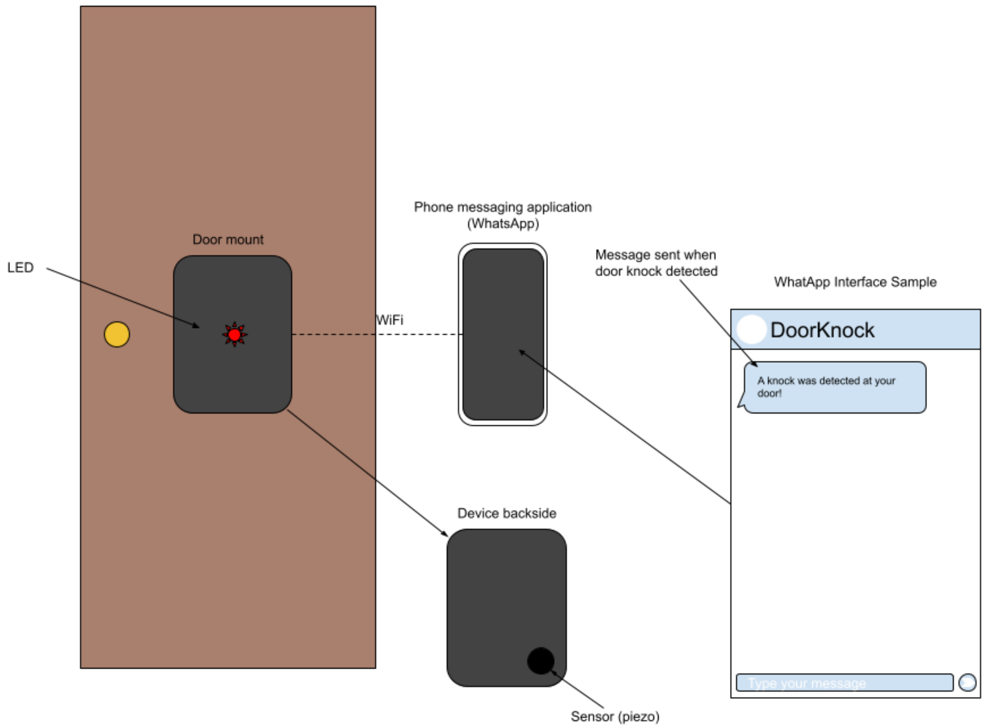

Figure 1. Visual diagram of device placement within the user's room as well as a visualization of the phone notification.

2 Design

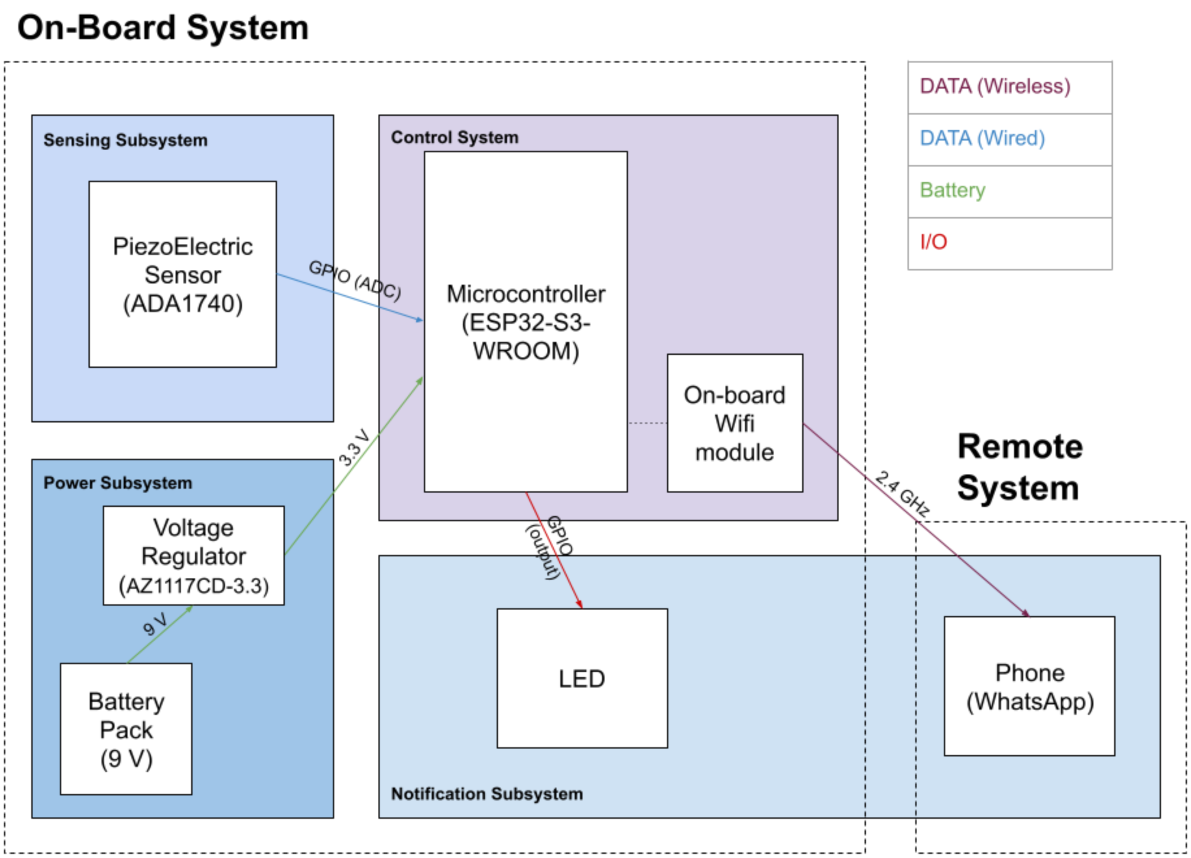

Figure 2. Block diagram of the door-knock detecting alarm. Labels indicate which components are in which subsystems and show the phone as being separate from the main device body.

2.1 Power Subsystem

The power subsystem is designed to power all the components of the device, with the goal of being able to power the device consistently for the duration of at least 10 hours on a single charge, as specified in the high-level requirements section. The main idea was to convert a 9 V battery to a 3.3 V power source using a linear voltage regulator as shown in Figure 3. The idea behind this was that the microcontroller used for this project was the ESP32-S3-WROOM, which requires a 3.3 V power supply. All peripherals operate well on the 3.3 V power source as well, with the LED specifically being powered by the microcontroller itself.

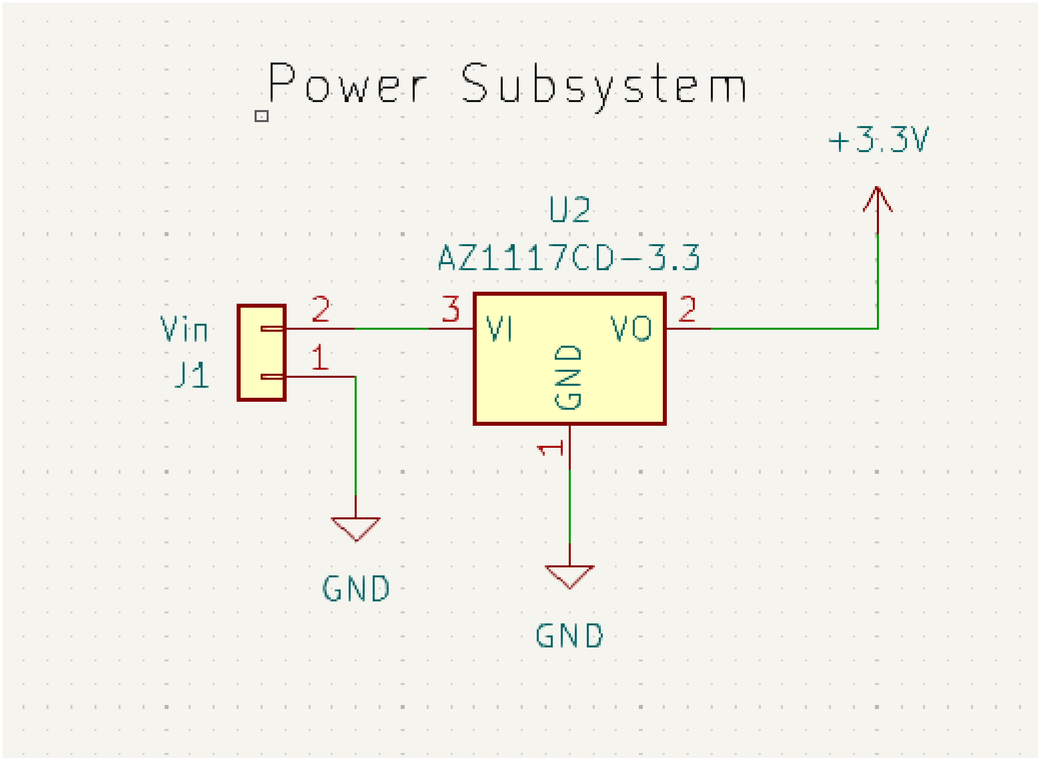

Figure 3. Circuit schematic of the power subsystem, including the battery and linear voltage regulator components.

Considerations were given to different power source options as well. Since the device is designed for mainly indoor use, rather than a battery, the team considered using a DC power cable to remove the need for battery replacement. This option was not chosen because of the logistical constraints of the room layout. Since the device is designed to be mounted on a door, having a cable of inadequate length could pose a tripping hazard if room layouts dictated that the cable be plugged in across the doorway. Considerations were also given to different battery voltages, but in the end a 9 V battery was chosen for the availability of the battery for consumers as well as the ability to power the device for the aforementioned 10-hour duration. Additionally, power consumption was further reduced by the software design. The ESP32-S3-WROOM has large power demands when using the on-board WiFi module, and as such the software design was implemented to optimize the times when the WiFi module was in use. This is described in more detail in section 2.3.2.

2.2 Sensing Subsystem

The sensing subsystem is designed to detect vibrations through a door using a piezoelectric sensor with the microcontroller reading inputs to run through the knock-detection filter. The specific connections are shown in Figure 4.

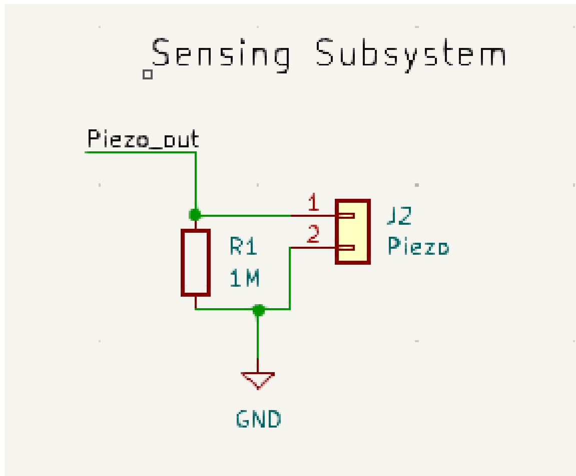

Figure 4. Circuit schematic of the sensing subsystem, including the setup of the piezoelectric sensor relative to the resistor and microcontroller input (labelled Piezo_out).

Piezoelectric sensors detect vibration frequencies using the piezoelectric effect, generating a voltage scaling on the strength of the vibration. The specific sensor used for this device was the AD1740 sensor by Adafruit, which had the specifications shown in Table 1[1].

Table 1 ADA1740 Design Specifications

Type	Threshold
Resonant Frequency	4000 ± 500 Hz
Maximum Operating Voltage	30 V
Current Consumption	1.5 mA at 12V

Other knock-sensing methods were considered were proximity sensors and audio sensors. The team found that proximity sensors would be better in detecting if someone was outside of the door rather than if someone were knocking on the door. This would solve a similar problem but was ruled out as a possible sensor because it required the device to be placed outside of the door, making it possible for it to be removed from the user's door as well as unnecessarily making it known that the user is deaf or hearing-impaired to the general community. The audio sensors were ruled out as they would be inaccurate using a simple threshold detection and would be better suited if used along a fast Fourier transform or machine learning algorithm to identify knocks. Since knocks are generally short, fast Fourier transforms would prove to be difficult, as there would be limited audio data to sample. The machine learning algorithm would require a large amount of computational power, which is not necessary using the threshold detection algorithm required by the piezoelectric sensor.

2.3 Control Subsystem

2.3.1 Microcontroller

The microcontroller used by the control subsystem for this device is the ESP32-S3-WROOM. This chip was chosen for its ability to connect to a WiFi network, which is required for the device to be able to send messages to the user's phone. The schematic of the control subsystem, shown in Figure 5, includes many different components besides just the microcontroller.

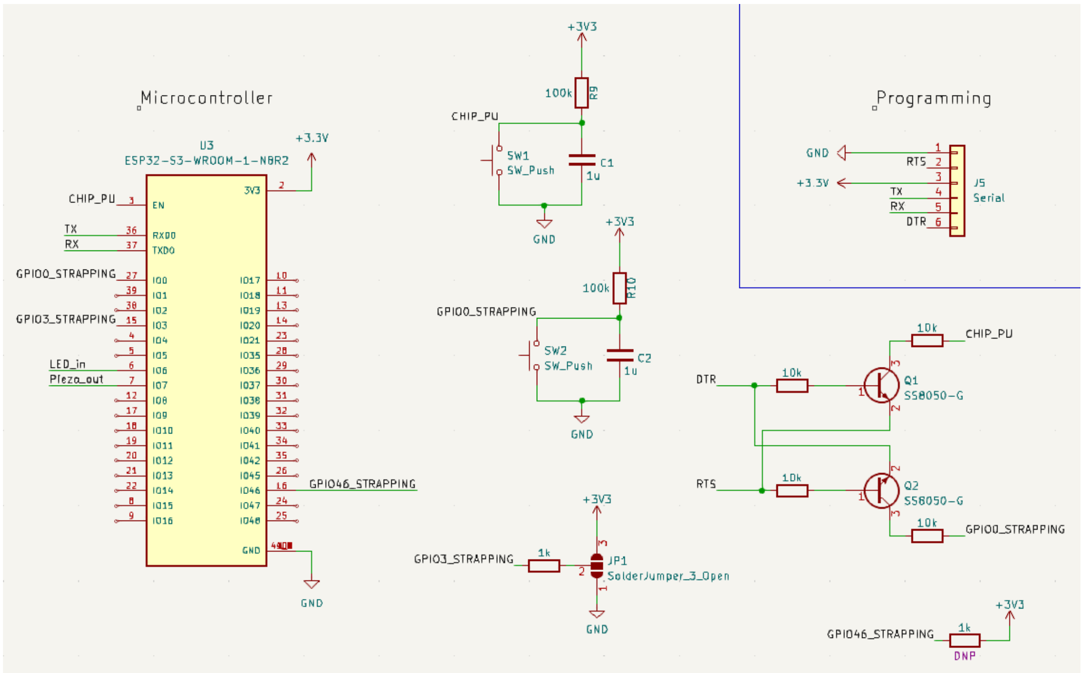

Figure 5. Circuit schematic for the control subsystem, including the ESP32-S3-WROOM, it's necessary strapping pin handling components, and programming header.

The extra components handle the behavior of strapping pins, helping control the behavior of the integrated circuits and memory on startup. The specific pins that are handled are strapping pin 0, 3, 45, 46. These pins function to put the chip into download mode (enabling flashing program memory), switch JTAG signal source, set voltage for internal serial peripheral interface memory, and set where the microcontroller ROM sends messages respectively[2].

The top right section of Figure 5 shows connection for the programming header used for the device. Specifically, the one used to program the microcontroller was the SparkFun Serial Basic Breakout[3], which is a USB-to-Serial adapter utilizing the CH340G chip. The programming header required data inputs for programming the ESP32-S3-WROOM but was unreliable. Possible reasons for this could include the reliability of the CH340G chip.

2.3.2 Microcontroller Software

The software uploaded to the microcontroller followed the algorithm shown in Figure 6. Key additions added to the original rudimentary code flow were WiFi connection optimizations, start-up messaging sending, and the addition of an idle time after knock detection. The WiFi connection was optimized, as mentioned in section 2.1, due to the high-power consumption of the ESP32-S3-WROOM's on-board WiFi module. It was optimized by connecting to WiFi, thus having the WiFi module on, only when messages needed to be sent and otherwise disconnecting from the network. The startup message was added for users to know whether device startup was successful. The idle time after initial knock detection was added to prevent an overload of notifications, as the team realized that often one series of knocks generally entailed people knocking two or three times in a row.

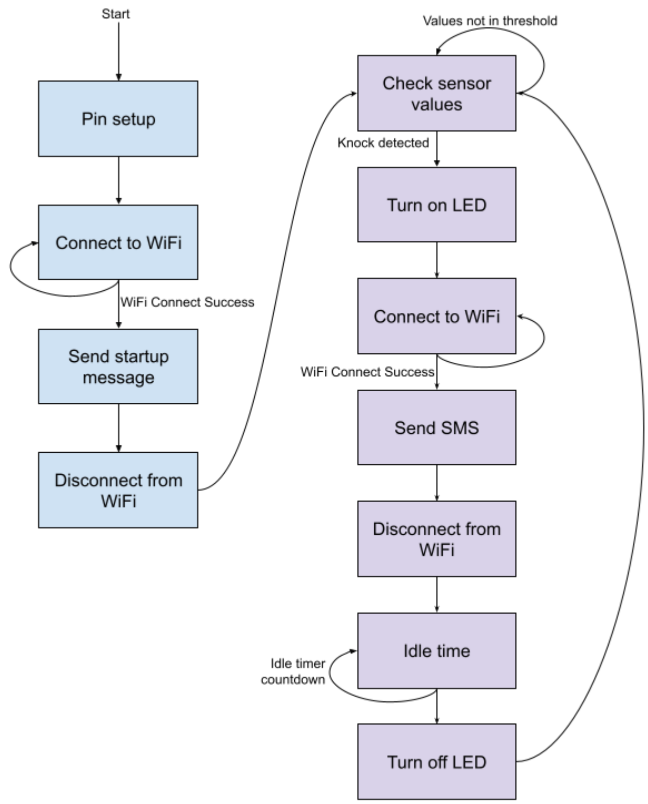

Figure 6. Code flow diagram for software of the door-knock detecting alarm. Shows the one-time setup sequence in blue on the left and the polling sequence in purple on the right.

2.4 Notification Subsystem

2.4.1 LED Notification

The LED portion of the notification subsystem is relatively straightforward, and its schema is shown in Figure 7. The powering of the LED is done through a pin on the microcontroller, with a 1000 Ω resistor used to ensure that the LED is not blown out by the microcontroller.

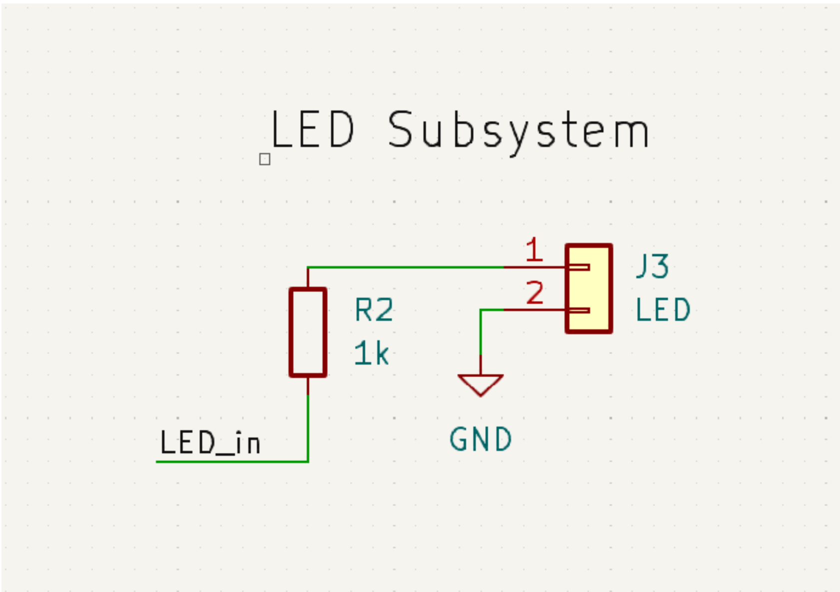

Figure 7. Circuit schematic for the LED portion of the notification subsystem, including the LED and its relation to the resistor and the microcontroller output pin.

2.4.2 Phone Notification

We decided to use the CallMeBot API to send notifications to the user through the ESP32's WiFi module. This API allows messages to be sent to phones using messaging services such as WhatsApp, Facebook Messenger, etc. While designing our own system for sending messages would have been possible, we decided to go with the CallMeBot API due to its availability and ease of use.

3. Design Verification

3.1 Power Verification

As shown in Table 3, we needed the ESP32 and the LED to be supplied with 3.3V and 1.8V, respectively. We were able to verify this using a multimeter that the voltages are within an acceptable range of these values.

It is imperative that the user can swap out the battery when it loses current, so we implemented a modular design to make this process simpler. The 9V battery is fastened with Velcro and a battery clip for ease of replacement.

3.2 Piezoelectric Sensor Verification

As indicated in Table 3, we initially proposed that the piezoelectric sensor picks up vibrations and sends the data to the microcontroller to apply signal processing techniques to it. When developing our product, we decided to modify this requirement as we decided against using signal processing methods and instead determined a numerical threshold through testing. The sensor chosen (ADA1740) generates values as a function of voltage, and our testing determined that an adequate cutoff threshold for a knock would be 60. As demonstrated in Figure 8, a knock is adequately detected when it exceeds the threshold, and that other types of ambient noise that would be present in a household environment do not display values that would result in a knock detection, all being under 40.

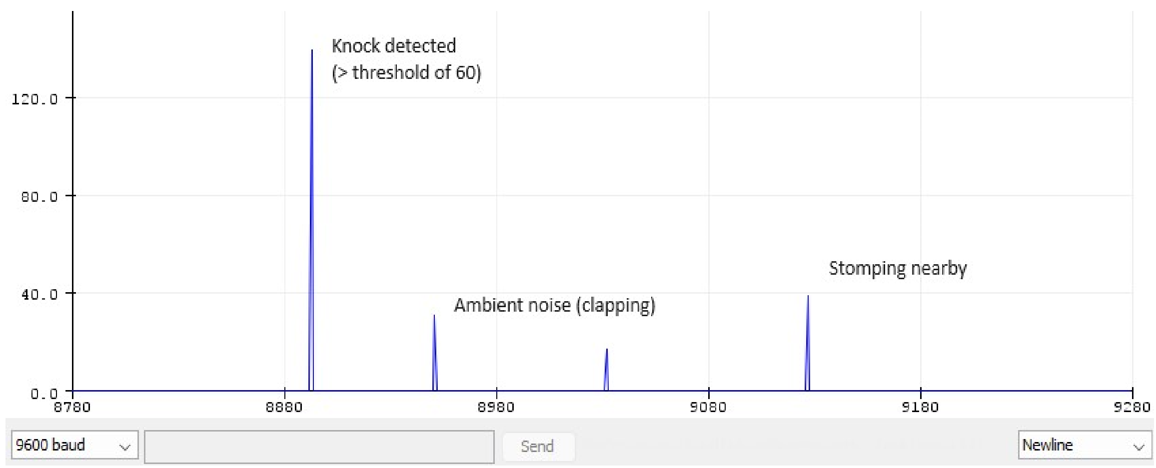

Figure 8. Screenshot of voltage-scaled sensor reading values created by different actions.

3.3 Notification Verification

We were able to verify that requirements were met by triggering the detection threshold. After this point, the LED would immediately turn on and a notification would be sent to the user within 10-15 seconds. Once the message is sent, the LED turns off. This satisfies the fourth requirement in Table 3.

While this subsystem was active, we used an ammeter to check the currents to the microcontroller and LED and found that they also satisfied the requirements listed in Table 3.

Figure 9. Screenshot of WhatsApp notification system, including start up and knock detection SMS messages.

3.4 Control Verification

An important aspect of the control system is turning the WiFi module on and off. The ESP32 has a relatively low power consumption that increases when the WiFi module is on. Therefore, the WiFi module is left off until a knock is detected. After the message is sent, it turns off again. This allows us to meet our high-level requirement of having a 10-hour life and still send timely notifications as detailed in Table 3.

4. Costs

4.1 Parts

Table 2 Parts Cost

Part	Manufacturer	Retail Cost ($)	Actual Cost ($)
ESP32	Espressif Systems	$3.20	$0
Piezoelectric Sensor	Adafruit	$3.95	$11.85 (3x)
9 Volt Battery	Duracell	$1.04	$0
PCB	PCBWay	$5	$0
Plastic Box	Machine Shop	$0	$0
Total		**$13.19**	**$11.85**

4.2 Labor

The average hourly salary of a graduate computer engineer is $40. Working for 10 hours a week for 13 weeks, the cost of labor is:

$40/hr * 10 hours/week * 13 weeks = $5,200 per engineer

A total of $15,600 for the three engineers

We will be using help from the machine shop. The cost of that is $22 for hourly labor, with work being done for 48 hours over the course of the project, the cost of labor is:

$22/hr * 48 hours = $1056 for 1 unit

5. Conclusion

5.1 Accomplishments

We were able to fulfill the high-level requirements that we had set from the start. Our device was able to detect knocking and send the user a notification in a timely manner while maintaining a low power consumption.

5.2 Uncertainties

The Adafruit piezoelectric sensor that we used was flimsy and resulted in a lot of threshold variability based on the wiring. The data that we read in from the sensor tended to have random noise.

While we were able to get our design working with the ESP32 Development Board and breadboard, we were unable to get our final design on the PCB reliably. We were able to upload our program to our PCB twice, but we were unable to recreate it. We know it worked because the ESP32 on board sent the power-on message to the user phone and was able to detect a knock and activate the notification subsystem. Around one in 20 uploads would result in our PCB activating while nothing happened the other times. We initially suspected that our data cable may be defective, but through testing we deemed that this was not the case. We also redesigned our PCB to put the MCU into bootloader mode, but this did not solve our problem. The reason that may have been the problem was regarding our programming header. We used the CH340 Serial Arduino Programmer, which, according to several TAs, was unreliable.

5.3 Ethical considerations

Our main ethical concern with this project is in Section 7.8.9 of the IEEE Code of Ethics, which pertains to the protection of property of other people[4]. One of the main purposes of our device is to allow people to have an easily installable, non-intrusive device to use on other people’s property. Therefore, our team will ensure a design that will not lead to permanent damage on whatever door surface it is mounted onto. Furthermore, it will be important to use proper battery and power safety techniques to prevent overheating or damaging the LEDs.

Regarding safety, since our device does house electrical components, we will make sure that these components are properly enclosed. In addition, our project will include an LED component, which will be used to notify the user of knocking. We will ensure a safe level of brightness and consistency to avoid photosensitive seizures.

5.4 Future work

Our primary issue was flashing the PCB, and thus, working to debug our programming issues would be our first step in continuing this project. Furthermore, there are many other types of piezoelectric sensors on the market that have greater sensitivity to vibration changes that we would like to test out.

In addition to acquiring a higher quality sensor, implementing a form of signal processing like Fast Fourier Transform rather than purely using a threshold determined through testing could help reduce noise at lower frequencies, something our sensor struggled with.

While the 9 V could last up to 14 hours, a device of this nature would be better if it could last much longer. An improvement that could be made is to replace the 9 V battery with a bigger battery pack. Another consideration is to make a version that can connect to an outlet. If the user has an outlet that is close to the door, they could choose to use that; otherwise, they can use the battery.

We also believe that creating some sort of GUI through an app or website for user setup would be crucial for bringing our product to market. This application would allow for the user to input their phone number and generate an API key to be able to communicate with the device, explaining the process to the user along the way. In addition, due to our utilization of the MultiWiFi module, we could allow for the user to add multiple WiFi connections as backups in the case that the primary WiFi is disconnected.

References

[1] Adafruit. “AD1740 Datasheet.” adafruit.com. https://cdn-shop.adafruit.com/product-files/1740/Datasheet.jpeg (accessed Oct. 4, 2023).

[2] UIUC ECE445 Staff. “ESP32-S3-WROOM EXAMPLE BOARD: MOTOR CONTROLLER.” courses.engr.illinois.edu/ece445/wiki. https://courses.engr.illinois.edu/ece445/wiki/#/esp32_example/index (accessed Nov. 7, 2023).

[3] Sparkfun. “SparkFun Serial Basic Breakout - CH340G.” sparkfun.com. https://www.sparkfun.com/products/14050 (accessed Nov. 10, 2023).

[4] IEEE code of ethics. IEEE Policies, Section 7 - Professional Activities. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Sep. 14, 2023).

Appendix A Requirement and Verification Table

Table 3 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Power Requirements 1. The ESP32 requires a constant supply of 3.3V and the LEDs require a stable voltage of 1.8V, so the voltage regulator must continuously monitor the output voltage to ensure that it remains within a range of ±5%, even when the system experiences changes in load or battery voltage. 2. The battery must be easily accessible, so that the user can replace the batteries without issue.	**Power Verifications** 1. Test with multimeter and oscilloscope to ensure that voltage being received at LED is ~1.8V and the voltage being received at the MCU is ~3.3V. 2. Demonstrate by removing battery from mount and reinserting - there should be no safety issues like loose wires and power should be immediately restored.	Y
Sensing Requirements 3. Send data to the ESP32 for signal processing at a constant rate and identify frequencies within the range of 0.2 to 1 kHz.	**Sensing Verifications** 3. Test by adjusting threshold frequency and different levels of knocking. A tone producer (tuner) will be used, holding it against the door to simulate vibrations at specific Hertz to see if the device is able to detect the required frequencies.	N
Notification Requirements 4. The LED will light up and the phone will receive a WhatsApp notification when the door-knocking alarm is triggered and only then. 5. Power supplied to the LED should not result in a loss of power and	**Notification Verifications** 4. Assemble the device with the complete sensor, microcontroller, and LED systems. Then, knock on the Piezoelectric sensor and ensure the notification is sent within 15 seconds and the LED lights up and	Y

functionality to the sensor system. The maximum current draw to the system during heavy use should be ~20 mA.	then turns back off after 5 seconds. 5. Connect an ammeter between the microcontroller/LED and power and check that the current draw to the LED does not pass ~20mA, and current to the microcontroller stays around 355mA. (High current while the WiFi module is active)	
Control Requirements 6. Correctly determine whether vibrations are background noise or knocking. 7. Communicate with WhatsApp to deliver notification and light the LED with a maximum delay of 5 seconds to light the LED and 15 seconds for the phone notification (provided good/reliable internet).	**Control Verifications** 6. This will be tested by knocking and checking detection ability as well as talking, walking, stomping past, and making various noises outside the door, further testing detection in those scenarios. 7. Initial test (without sensor detection first) to send sample messages to WhatsApp will be conducted to test ability to send messages. LED will be tested after sensing, with a timer to test how long the LED remains on after the last knock/vibration. This will be cross-checked with the message notification to ensure that both are reacting to sensor detection.	Y