Bachelor of Science in Electrical Engineering
June 2023

Wearable Emergency Alerting System With Enhanced Inclusivity with Interactive Acknowledgment and SOS Distress Signal

Jotheesh Reddy Kummathi
Savithri Venkata Tejeswar

Faculty of Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Department of Mathematics and Natural Sciences at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Bachelor of Science in Electrical Engineering. The thesis is equivalent to 10 weeks of full time studies.

Contact Information:
Author(s):
Jotheesh Reddy Kummathi
E-mail: joku22@student.bth.se

Savithri Venkata Tejeswar
E-mail: vesa22@student.bth.se

University advisor:
Irina Gertsovich
Department of Mathematics and Natural Sciences

Dept. of Mathematics and Natural Sciences
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

This abstract introduces the wearable-based alerting system for emergency situations for both disabled and non-disabled individuals. The system sends advanced technology to detect various emergencies like fire alerts, flood alerts, outside air contamination etc, for the emergency services and it gives a signal in the form of vibration for the user. AdafruitIO is a platform created to show, react to, and interact with the data from the project.

The most important problem statement revolves around the need for an inclusive and green emergency alert machine that caters to the various desires of people, which include people with disabilities. The device's primary objective is to ensure the protection and nicely-being of users in emergency scenarios.

To address this problem, a wearable device is utilized as a verbal exchange interface between the user and the emergency services. The device is prepared with signal receiving capable of detecting precise emergency occasions, together with modifications in high-quality air or strange environmental conditions. When an emergency is detected, the device promptly sends a signal to the applicable emergency services and making sure a quick reaction.

The findings of this research focus on the effectiveness and reliability of the wearable-primarily based alerting machine. Through rigorous testing and assessment, it has been validated that the machine can as it should come across a huge variety of emergency situations, making certain well-timed communication with emergency services.

In conclusion, this study provides a wearable-based totally alerting device that addresses the desires of both disabled and non-disabled people in emergency conditions. By exploiting advanced technology and the AdafruitIO platform, the device presents actual-time indicators to emergency services even as turning in spontaneous feedback to the user. This modern answer enhances basic safety and inclusivity through important activities, supplying a promising future for emergency response systems.

Keywords: AdafruitIO, Alerting system, Emergency situations, Feeds, Vibrations.

Acknowledgments

We would like to express our heartfelt gratitude to all the individuals who have played a crucial role in the completion of our bachelor's thesis. First and foremost, we are immensely grateful to our supervisor, Irina Gertsovich, for her unwavering guidance and invaluable insights throughout this journey.

Additionally, we would like to extend our sincere appreciation to our family and friends for their consistent encouragement, understanding, and motivation during challenging times. From the bottom of our hearts, we sincerely thank you for your guidance and unwavering belief in our abilities.

Kummathi Jotheesh Reddy
Venkata Tejeswar Savithri

Contents

Abstract iii

Acknowledgments iv

List of Figures viii

List of Tables ix

1 Introduction 1
- 1.1 Background . . . 2
- 1.2 Scope of the Thesis . . . 2
- 1.3 Ethical, Societal and Sustainability Aspects . . . 3

2 Related Work 4

3 Method 7
- 3.1 Problem Statement . . . 7
- 3.2 Research Questions . . . 7
- 3.3 Aim and Objectives . . . 8
- 3.4 Main Contributions . . . 8

4 System Design and Implementation 9
- 4.1 System Design . . . 9
- 4.2 Implementation . . . 11
  - 4.2.1 Hardware Implementation . . . 11
  - 4.2.2 Assembly . . . 20
  - 4.2.3 Software Implementation . . . 21
  - 4.2.4 Programming Languages and Platforms . . . 21
  - 4.2.5 Implementation Flow . . . 29

5 Validation and Evaluation 34
- 5.1 Validation . . . 34
- 5.2 Evaluation . . . 45

6 Discussion 49
- 6.1 Limitations . . . 52

7 Conclusions and Future Work **53**
7.1 Conclusion 53
7.2 Future Work 54

References **55**

List of Figures

2.1 Vibrotactile alarm display as an armlet [1]. 4
2.2 The prototype was based on a regular shoe including two force sensors, one at the tip of the shoe, one at the heel and both are connected to an IoT node [2]. 5
2.3 Situation-Aware Sensor-Based Wearable Computing Systems: A Reference Architecture-Driven Review [3]. 6

4.1 Proposed system design. 9
4.2 Proposed system design with components. 10
4.3 Circuit diagram. 11
4.4 Connection of the hardware elements in the wearable part of the system. 12
4.5 Pinout diagram of the Adafruit QT Py ESP32-S2 WiFi Dev Board [4]. 13
4.6 Adafruit DRV2605L Haptic Motor Controller [5]. 14
4.7 Vibration mini ERM motor disc [5]. 15
4.8 Adafruit LiIon or LiPoly Charger. 16
4.9 Push button. 17
4.10 Lithium Ion Polymer Battery (3.7V 400mAh) [6]. 18
4.11 Connection using STEMMA QT JST SH 4-pin [7]. 18
4.12 3D design of the wearable part of the system. 19
4.13 3D designed band straps for the watch head. 19
4.14 Assembled prototype. 20
4.15 Assembled prototype. 20
4.16 Adafruit IO Architecture [8]. 22
4.17 Arduino Settings tab. 23
4.18 AdafruitIO key. 24
4.19 Libraries used in the project. 24
4.20 Dashboard in AdafruitIO. 26
4.21 Feeds in AdafruitIO. 26
4.22 Applets in the IFTTT. 27
4.23 Flow chart of device connectivity. 29
4.24 Flow chart of SOS triggering processes. 30
4.25 Toggle button values assigned to each alert. 31
4.26 Flowchart of the functionality. 32
4.27 Flow-chart of the sending 'SAFE' message. 33

5.1 Image of the output window in Arduino IDE. 34
5.2 Device connectivity validation from the serial monitor of Arduino IDE. 35
5.3 Code snippet of the SOS. 36

5.4 Dashboard of AdfruitIO for three SOS cases. 37
5.5 Serial monitor for the three cases. 37
5.6 Dashboard of Adafruit IO when the alert is sent to the device. 38
5.7 If Else if statement conditions for the alerts. 39
5.8 Dashboard from sending Device 1 fire alert. 39
5.9 Code snippet and serial monitor after device receives the toggle value 4. 40
5.10 AdafruitIO dashboard for case 3. 41
5.11 Serial monitor ouput for the first and second alert. 41
5.12 Dashboard for the first alert not acknowledged. 42
5.13 Dashboard for a second alert not acknowledged. 42
5.14 Serial monitor shows not acknowledged alerts. 43
5.15 Image of status indicator before and after receiving the SAFE message. 44
5.16 Code snippet of the functionality long press. 44
5.17 Three different types of vibration patterns for the alerts. 47
5.18 Vibration patterns 48

List of Tables

4.1 Technical specification of Adafruit QT Py ESP32-S2 [9]. 13
4.2 Technical specification of Adafruit DRV2605L Haptic Motor Controller [5]. 14
4.3 Technical specification of Vibrating Mini ERM Motor Disc [10]. . . . 15
4.4 Technical specification of Adafruit LiIon or LiPoly Charger [11]. . . . 16
4.5 Technical specification of Large Tactile Push Button Switch [12]. . . . 17
4.6 Toggle button assigned feeds and their values. 28
4.7 Status indicator conditions to show green colour. 28

5.1 Number of attempts vs successful attempts vs unsuccessful Attempts 45
5.2 Number of attempts vs successful attempts vs unsuccessful Attempts after modification 45
5.3 Case 1 and Case 2 46
5.4 Case 3 46
5.5 Results of type 1 and 2 in case 4. 47
5.6 The average percentage of alerts recognized by different age groups . 48

Chapter 1

Introduction

Emergency situations present significant challenges that can often be amplified by traditional alert systems. These systems typically rely heavily on auditory and visual cues, which may not adequately cater to the diverse needs of all individuals within a community. For instance, sudden and loud alerts can potentially cause distress to individuals with autism, due to their heightened sensitivity to sound and light. Similarly, those with hearing impairments, including the deaf and hard-of-hearing communities, may encounter difficulties in perceiving these alerts effectively [13] [14].

These considerations underscore the need for an inclusive approach to designing emergency systems. To accommodate the varying needs, alternative options such as text alerts, vibrations, and alerts with gradual increases in volume could be beneficial. Crucially, engaging these communities in the planning and development process can help to ensure that emergency responses are more effective and accessible, thereby reducing the risk and impact of crises.

In recent years, wearable technology has emerged as a promising solution to numerous challenges, offering non-intrusive monitoring of vital signs, location data, and physical activities [15] [16]. These capabilities present an opportunity to develop an alerting system that can detect emergencies and promptly alert both emergency services and designated contacts. For instance, devices like Apple Watches and Fitbits have already proven their life-saving potential, as individuals have sought medical assistance based on alerts from these wearables. Additionally, wearables provide convenience and independence, particularly for seniors and individuals with health limitations. By integrating these features, a wearable-based alerting system can offer effective emergency detection and rapid response, ensuring enhanced safety and peace of mind [17].

This thesis states the crucial point of an emergency alert system for people with disabilities. There is a definite need for effective, accessible, and personalized alert systems that can reduce the risk of death and increase the chance of survival, given the dangers for people with disabilities during catastrophes. This thesis is going to examine the design, development, and evaluation of a wearable-based alerting system that is specially designed for people with disabilities to meet their requirements. The system is designed to detect various types of emergencies like fire, break-ins, and threats. The system is capable of communicating with the emergency services during the emergencies [18].

This thesis contributes to wearable technology development by focusing on wearable-based alerting systems for emergencies. It offers insights to enhance safety for individuals with disabilities and the broader population. The wearable proposed utilizes vibrations as a primary alert mechanism, thereby improving the user safety of individuals in critical situations.

1.1 Background

Wearable technology has made huge advances in recent years, altering several areas of our lives. Wearable devices and other wearable gadgets have become more popular because of their capacity to offer a wide variety of features in a small, convenient package. Innovative solutions that may meet the safety needs of people are increasingly needed due to the rising need for personal safety and disaster preparation [19].

Fire alarms and sirens are two examples of conventional alerting devices that have long been used to issue warnings in emergency circumstances [20]. However, when it comes to addressing the unique needs of people with disabilities, these systems frequently have limits. For those who are hard of hearing or deaf, visual or auditory warnings could not be helpful, and generic alerts might not offer individualized information to guarantee that individuals are secure in different emergency circumstances [21].

To address these limitations and provide a more inclusive and effective solution, wearable-based alerting systems have emerged as a look at the future. Personalized and real-time notifications can be sent to people, independent of their capacities, by incorporating wearable technology into the emergency response system [22]. Another layer of assurance for users and their caregivers is provided by the development of wearable gadgets that can also verify a person's safety [23].

In summary, the background of this study shows the importance of portable wearable systems for disabled people and the general population in emergency situations. By removing the limitations of traditional alarm systems and leveraging the potential of wearable technology, the proposed system aims to improve personal safety and provide a sense of security. This paper will build on existing research and literature to develop and evaluate a portable alarm system that can effectively address the diverse needs of individuals in emergency situations.

1.2 Scope of the Thesis

The purpose of this thesis is to design, develop, and evaluate portable emergency warning systems that meet the needs of both people with disabilities are individuals as well. It is a wristwatch-type wearable terminal with a unique function that allows

you to confirm your safety with a long vibration when you press a button [24]. The device aims to provide personalized, real-time alerts in emergency situations while addressing the concerns of families, caregivers, and first contact [25].

To ensure a deep and comprehensive analysis, the work will mainly focus on the following aspects:

- The technological and design components of creating the wearable gadget will be covered in depth in the thesis. Providing efficient alerting and safety confirmation involves investigating the most appropriate sensors, communication methods, and user interface designs.
- The wearable-based alerting system's performance and efficacy in various emergency situations will be evaluated in this thesis. Through user trials and comments, this evaluation will test the system's precision, dependability, responsiveness, and user satisfaction.
- The wearable gadget must be made to accommodate various disabilities, and alerts must be provided in accessible formats, including visual, tactile, and auditory signals, according to the thesis, which explicitly addresses the needs of people with disabilities. The objective is to provide an inclusive system that meets the requirements of both impaired and disabled individuals.

By limiting its scope, this thesis aims to provide a universal and detailed analysis of the design, development, and evaluation of portable warning systems for emergencies. This approach allows a targeted study of system effectiveness, usability, and comprehensiveness, providing valuable insight into the development of future wearable technologies in the emergency response space.

The thesis will be structured as follows: The introduction will provide an overview of the research topic. Related work will discuss existing works in the field. The method section will include the problem statement, research question, aim and objectives, and main contribution. System design and implementation will detail hardware and software implementations. Evaluation of the thesis, followed by a discussion. The thesis will conclude with a summary and future work.

1.3 Ethical, Societal and Sustainability Aspects

An alert system utilizing wearable technology for emergency situations raises ethical, societal, and sustainability considerations. Its primary objective is to serve the general population and cater to individuals with disabilities. The system prioritizes user privacy and data protection, ensuring that personal data is securely stored and used solely for emergency purposes. In terms of sustainability, we have developed a custom-designed prototype using 3D printing technology, and the material used for 3D printing is PLA (Polylactic Acid). PLA is a biodegradable and renewable material derived from plant-based resources such as cornstarch or sugarcane. It is considered more environmentally friendly than other plastics due to its biodegradability and renewable sourcing [26].

Chapter 2

Related Work

Fraser Young, Li Zhang, Richard Jiang, Han Liu, and Conor Wall have done research on "A Deep Learning Based Wearable Healthcare Iot Device for AI-Enabled Hearing Assistance Automation". This study proposes a wearable-based alerting system specifically designed for deaf individuals during emergency situations. The system utilizes a smartwatch that delivers visual and haptic alerts to notify the wearer about potential dangers. The authors conducted a user study involving deaf participants to evaluate the effectiveness of the system. The results indicated that the wearable-based alerting system significantly improved the participants' awareness and response time during emergency scenarios. The study highlights the importance of personalized alerts for individuals with specific sensory impairments and demonstrates the potential of wearable devices in enhancing their safety and well-being [27].

Vanessa Cobus and Wilko Heuten have done research on "To Beep or Not to Beep? Evaluating Modalities for Multimodal ICU Alarms". The research paper presents the design and evaluation of a multimodal wearable alerting system for emergency situations. The system integrates visual, auditory, and haptic alerts to effectively notify both people with disabilities are individuals as well. The authors conducted user experiments involving participants with and without disabilities to assess the system's usability and performance as shown in Figure 2.1. The results indicated that the multimodal approach significantly improved the participants' situational awareness and response times compared to traditional alerting methods. The study emphasizes the importance of considering multiple modalities in wearable-based alerting systems to accommodate various user preferences and needs [1].

Figure 2.1: Vibrotactile alarm display as an armlet [1].

Charalampos Orfanidis, Rayén Bel Haj Hassen, Armando Kwiek, Xenofon Fafoutis, and Martin Jacobsson have done research on A Discreet Wearable Long-Range Emergency System Based on Embedded Machine Learning. This study presents a wearable-based smart alert system designed to provide timely alerts and assistance during emergency situations. The system includes a device that has sensors capable of identifying abnormal environmental conditions and user conduct. Whenever an emergency is discovered, the device sends apt alarms like visual cues, vibrations, and audible sounds. The researchers performed different experiments to test the efficacy of the system in hazardous situations as shown in Figure 2.2. The results showed that individuals could be warned about any potential danger timely and helped them respond appropriately. This study emphasizes the need for integrating intelligent sensing capabilities into wearable tech to heighten emergency preparedness among disabled people as well as those without disabilities [2].

Figure 2.2: The prototype was based on a regular shoe including two force sensors, one at the tip of the shoe, one at the heel and both are connected to an IoT node [2].

Steven Goodman, Susanne Kirchner, Rose Guttman, Dhruv Jain, Jon Froehlich, and Leah Findlater have done research on "Evaluating Smartwatch-based Sound Feedback for Deaf and Hard-of-hearing Users Across Contexts". This research paper presents a smartwatch-based personal safety system designed to provide assistance during emergency situations. The system employs a device equipped with sensors to detect abnormal environmental conditions and user behaviours. When an emergency is detected, the device delivers appropriate alerts, such as visual cues, vibrations, and audible alarms. The authors conducted a series of experiments to evaluate the system's effectiveness in emergency scenarios. The results demonstrated the system's ability to promptly notify individuals about potential dangers and facilitate their response. The study highlights the importance of integrating intelligent sensing capabilities into wearable devices to enhance emergency preparedness for people with disabilities are individuals as well [28].

Giuseppe D'Aniello, Raffaele Gravina, Matteo Gaeta, and Giancarlo Fortino have done research on "Situation-Aware Sensor-Based Wearable Computing Systems: A Reference Architecture-Driven Review". This study discusses the application of wearable devices for emergency notification and situation awareness in both individual and group contexts. In reviewing a variety of wearable technology, including smartwatches, fitness trackers, and smart glasses, the authors emphasize both their advantages and disadvantages in emergency situations. The study explores the difficulties and possibilities of wearable-based alerting systems and offers details on design issues, communication protocols, and localization strategies. To improve situational awareness, communication, and coordination, the study underlines the significance of incorporating wearable technology into current emergency response systems as shown in Figure 2.3. It explores the potential advantages of alerting systems based on wearable technology for both the public and people with disabilities, eventually promoting the creation of more effective and inclusive emergency management plans [3].

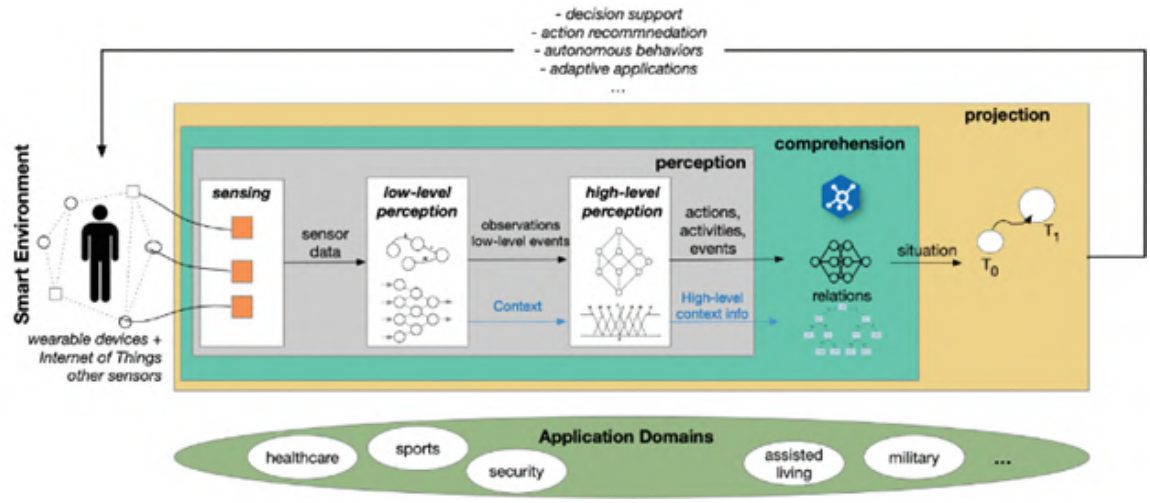

Figure 2.3: Situation-Aware Sensor-Based Wearable Computing Systems: A Reference Architecture-Driven Review [3].

Chapter 3

Method

In this chapter, we will discuss the problem that we are addressing in this thesis, along with the corresponding research questions, aims, and objectives. We will also highlight the main contributions of this thesis.

3.1 Problem Statement

Existing emergency alert systems lack inclusivity and personalization, failing to address the diverse needs of individuals, including disabled people, the elderly, and non-disability people, during emergencies [29]. Limited real-time information and insufficient consideration for individual circumstances compromise personal safety. Wearable technology, with its potential to track vitals, location, and physical activities, offers a promising solution [30]. However, there is a significant research gap in designing, developing, and evaluating wearable-based alerting systems specifically for emergency situations [31].

This thesis aims to address the stated problem by developing an inclusive and effective wearable-based alerting system for disabled people, the elderly, and the general population. The system will provide personalized alerts and collect real-time feedback, ensuring individual safety in emergencies [32]. Additionally, emergency response departments often lack information about individuals left behind during rescue operations. Addressing this critical issue, the proposed wearable alerting system will enhance emergency response and improve the safety of vulnerable populations.

3.2 Research Questions

1. How can we develop an inclusive alert system, using wearable technology, that effectively notifies individuals of varying abilities during emergency situations?
2. How do the intensity and pattern of vibration alerts impact an individual's sensory perception and response time?
3. What design principles can be employed to optimize a wearable alerting system, ensuring rapid and effective delivery of emergency alerts while minimizing the occurrence of false alarms or unnecessary alerts?

4. How can a single-button feedback system be designed and implemented to efficiently collect post-alert condition information from users?

5. What technologies can be used to ensure that the alert system reliably sends simultaneous messages to the relevant department and the user's primary contact, thereby streamlining the emergency response process?

3.3 Aim and Objectives

This research aims to develop an alarm system for individuals with diverse abilities using wearable technology. As many people demonstrate the ubiquity of touch sensitivity regardless of diverse abilities [33], our project seeks to develop a wearable device that leverages this touch sensitivity to alert a wide range of users.

Our main objective is to enhance the existing warning system by implementing our own functionality. Our system utilizes tactile feedback through vibration patterns as an alarm mechanism. This system goes beyond simply informing users; it also enables the collection of immediate post-alert status feedback from individuals with just a single push of a button.

Additionally, our project aims to establish concurrent messaging capabilities. This feature automatically sends a message in response to an alert to the appropriate department associated with the alert type, as well as to the person identified as the user's primary emergency contact. This dual communication process is expected to expedite the emergency response and assistance process, thereby improving the overall effectiveness of the warning system.

3.4 Main Contributions

- Development of an inclusive wearable-based alert system for individuals with varying abilities during emergencies.
- Investigation of the impact of vibration alert intensity and patterns on sensory perception and response time.
- Establishment of design principles to optimize the wearable alerting system, minimizing false alarms and ensuring rapid delivery of emergency alerts.
- Design and implementation of a single-button feedback system for efficient collection of post-alert condition information from users.
- Integration of technologies to enable simultaneous messaging to relevant departments and users' primary contacts, streamlining the emergency response process.

Chapter 4

System Design and Implementation

4.1 System Design

Figure 4.1 illustrates the proposed system of a wearable-based alerting system. It begins with the power supply, which powers the microcontroller for its operation, and a reachable battery is utilized. Once the connection with the AdafruitIO web server is established, two-way communication is enabled. This allows the web server to send alerts to the microcontroller, and the microcontroller to send acknowledgments and SOS signals. Furthermore, the microcontroller should have the capability to receive alerts from the web server and convey them to the user through vibrations.

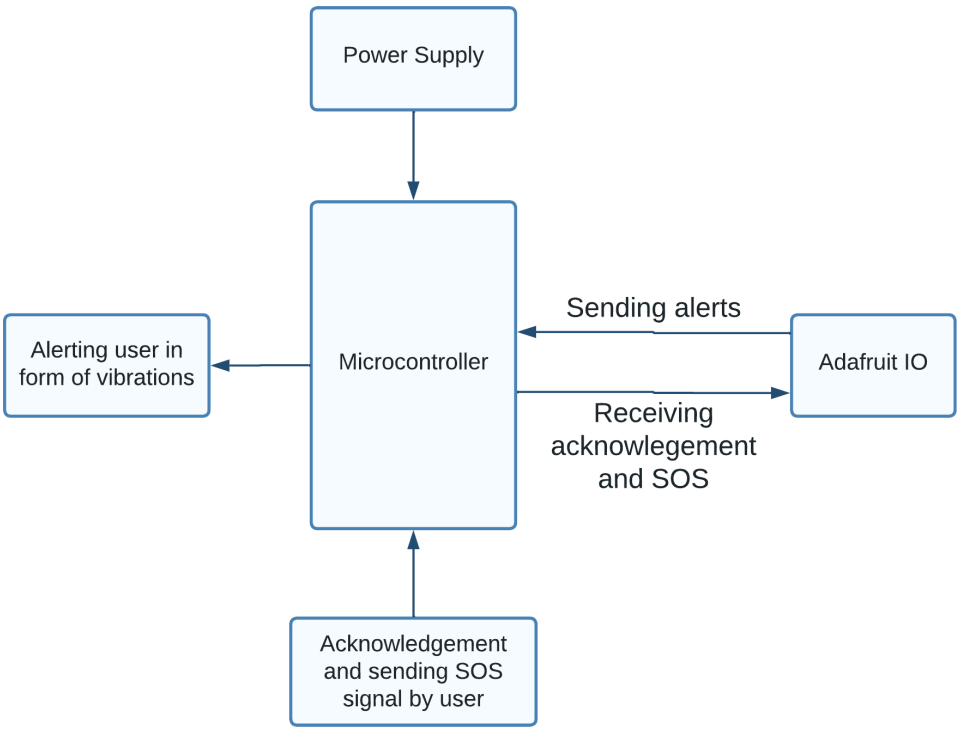

Figure 4.1: Proposed system design.

The wearable-based alerting system includes several components to ensure effective and dependable operation in emergency scenarios. The design makes use of a push button for human feedback, a vibration motor soldered to the DRV2605L haptic controller, a power source with an Adafruit LiPo charger, the Adafruit QT Py ESP32-S2 microcontroller, and an AdafruitIO platform as shown in Figure 4.2.

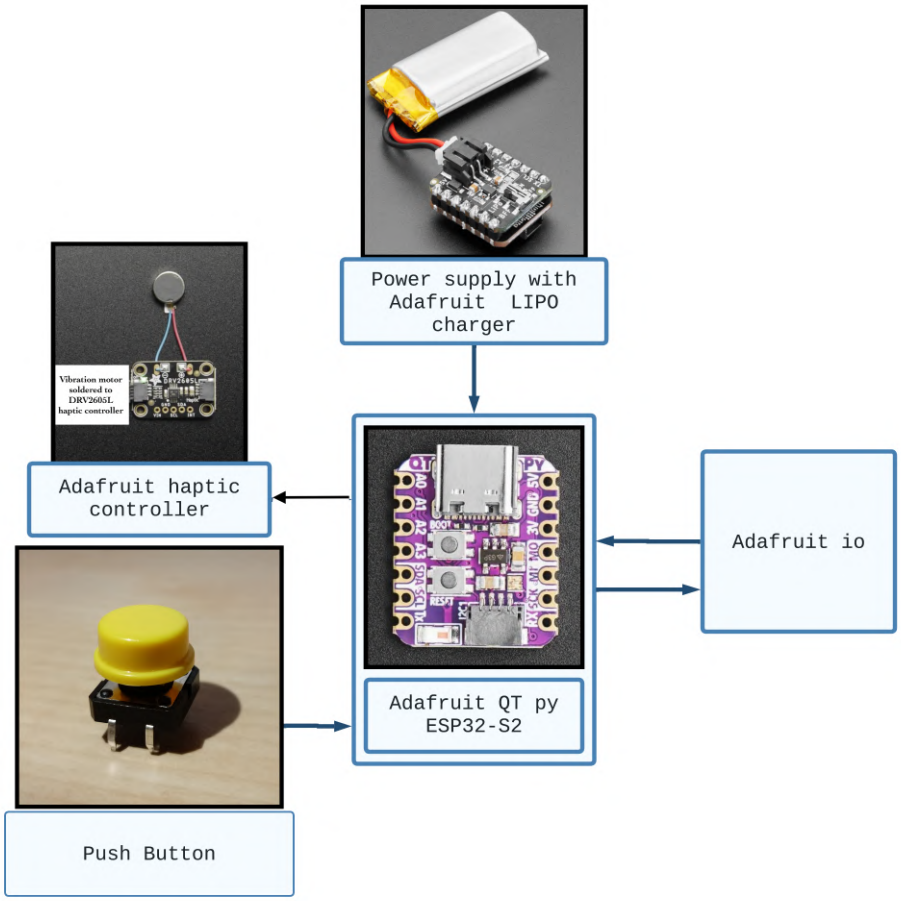

Figure 4.2: Proposed system design with components.

The wearable-based alerting system has certain requirements and limitations that ensure its effective functionality during emergency situations. The system will be able to provide a quick response time, allowing for immediate action when activated. Right now it can be capable of handling 2 number of people simultaneously and able to expand accordingly. The system's speed of operation is fast to ensure prompt communication and assistance. The device can utilize communication technologies such as Wi-Fi to establish connections for transmitting emergency signals. As for the wearable placement, can be worn on the arm, neck, or chest, providing flexibility and convenience for the user.

4.2 Implementation

This section provides details about the hardware implementation and the components utilized in our model. It is followed by a discussion of the software implementation and the flowchart illustrating the implementation process.

1. Hardware Implementation
2. Software Implementation
3. Implementation flowchart

4.2.1 Hardware Implementation

The hardware implementation of the design system involves several components that are connected together to create a functional device. The Adafruit QT Py ESP32-S2 WiFi Dev Board with STEMMA QT, serves as the central processing unit of the system, while the Adafruit DRV2605L Haptic Motor Controller with STEMMA QT, provides haptic feedback. These two devices are connected using a JST-SH female 4-pin connector and are powered by a Lithium Ion Polymer Battery (3.7V, 400mAh), which is connected to the Adafruit LiIon or LiPoly Charger. Additionally, a push button is connected to analog pin 1 of the system to enable user input. Figure 4.3 shows the circuit diagram of the designed system and the connections between the components that we have used in this system.

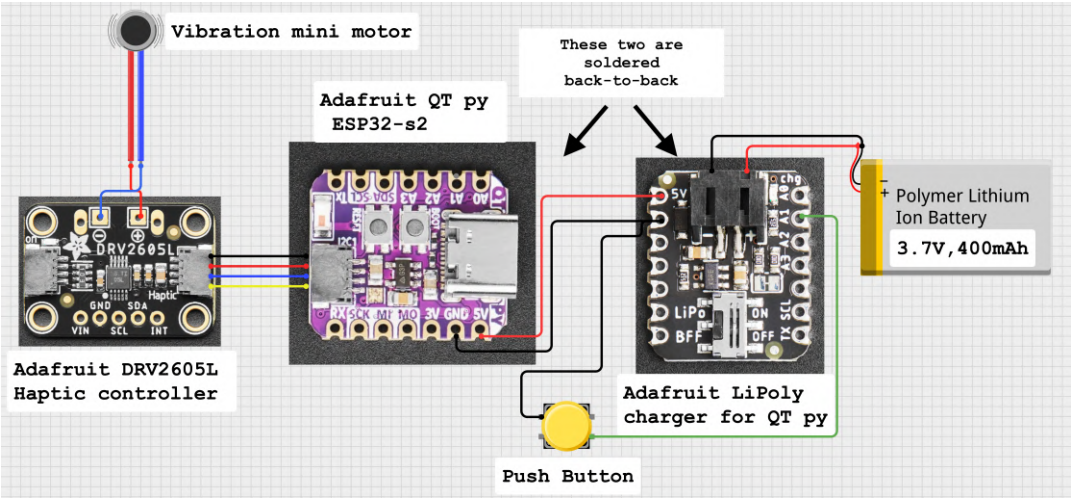

Figure 4.3: Circuit diagram.

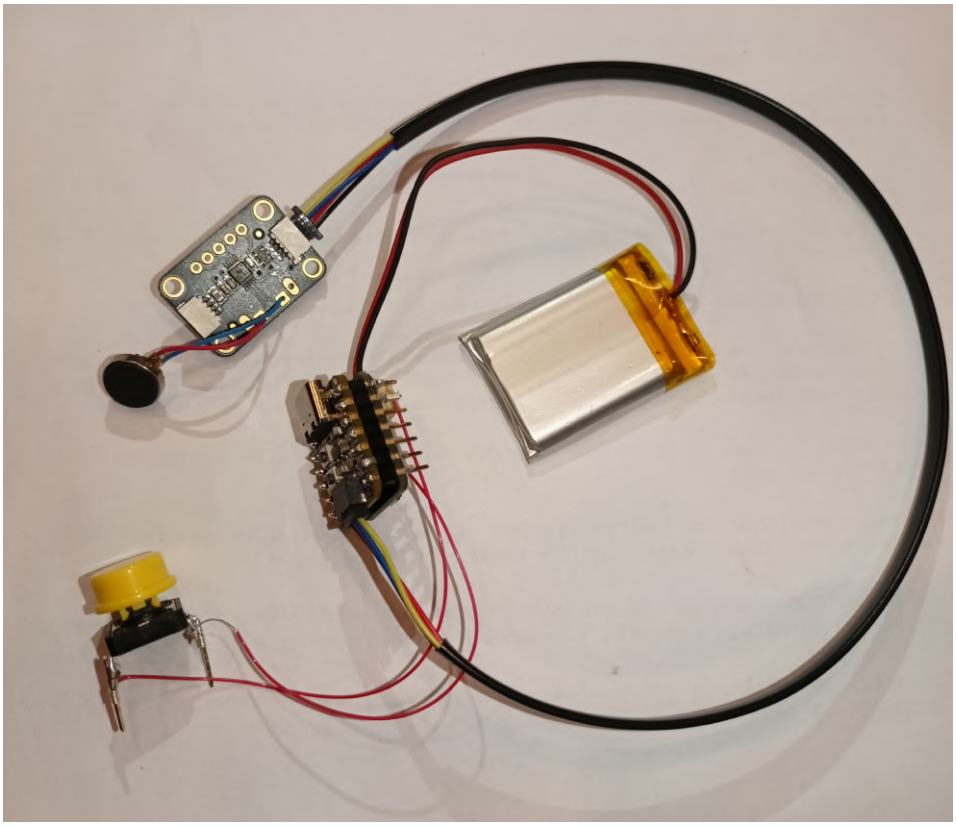

Figure 4.4: Connection of the hardware elements in the wearable part of the system.

4.2.1.1 Hardware Components

In Figure 4.4 circuit diagram shows several components used in the development of the system. In this section, we will go through the details of each component to gain a better understanding of their role in the system.

The hardware implementation consists of the following components:

- Adafruit QT Py ESP32-S2 WiFi Dev Board with STEMMA QT
- Adafruit DRV2605L Haptic Motor Controller
- Vibrating Mini Motor Disc
- Adafruit LiIon or LiPoly Charger
- Large Tact Push Button Switch
- Lithium Ion Polymer Battery (3.7V 400mAh)
- STEMMA QT JST SH 4-pin Cable
- Designed Prototype 3D model

4.2.1.2 Adafruit QT Py ESP32-S2 WiFi Dev Board with STEMMA QT

The Adafruit QT Py ESP32-S2 WiFi Dev Board with STEMMA QT was chosen as the main controller of the project due to its small size and built-in WIFI chip, which met our project requirements. The device utilizes the ESP32-S2 chip as its microcontroller and features an attached STEMMA QT port that played a crucial role in connecting the DRV2605L haptic controller to the system.Figure 4.5 shows the Adafruit QT Py ESP32-S2 WiFi Dev Board. The technical specifications of the microcontroller are presented in the Table 4.1 [9].

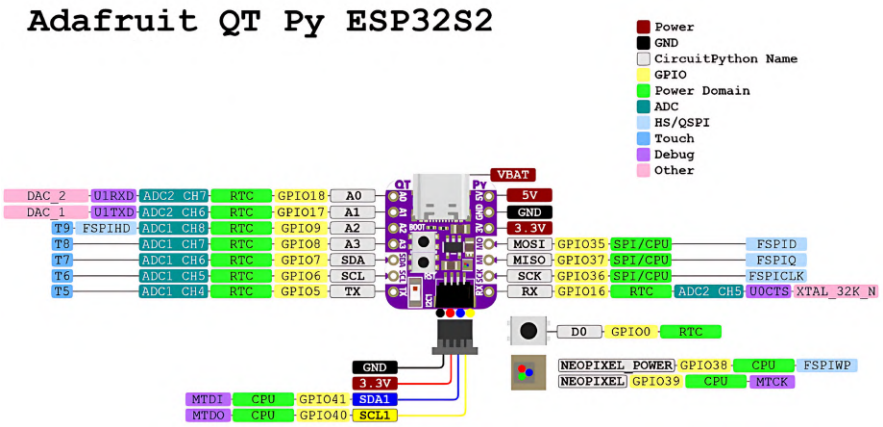

Figure 4.5: Pinout diagram of the Adafruit QT Py ESP32-S2 WiFi Dev Board [4].

Table 4.1: Technical specification of Adafruit QT Py ESP32-S2 [9].

Technical specifications		
Microcontroller	Espressif ESP32-S2	
	Flash	4 MB
	SRAM	320 KB
Clock Speed	240 MHz	
Wi-Fi	2.4 GHz 802.11b/g/n	
Power Supply	3.3V DC	
Dimensions	21.8mm x 17.9mm x 5.7mm	
Weight	2.1g / 0.1oz	

4.2.1.3 Adafruit DRV2605L Haptic Motor Controller

The Adafruit DRV2605L Haptic Motor Controller is a motor driver designed specifically for controlling haptic motors such as buzzers and vibration motors. It is used to control the vibration motor and can play 123 built-in vibration effects in different waveforms accessed through the DRV2605L Library in the Arduino IDE. The DRV2605L can control the vibration motor waveforms by adjusting the amplitude, frequency, and duration of the vibration. It is a useful tool for creating haptic feedback in our prototype [5]. Figure 4.5 shows the Adafruit DRV2605L Haptic Motor Controller we used in our system.

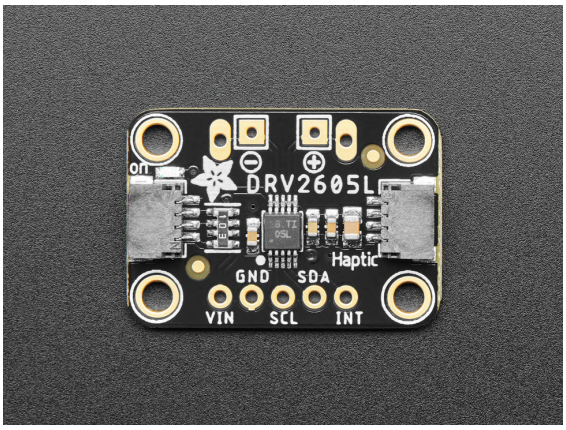

Figure 4.6: Adafruit DRV2605L Haptic Motor Controller [5].

Table 4.2: Technical specification of Adafruit DRV2605L Haptic Motor Controller [5].

Technical specifications	
Supply Voltage Range	2.0V to 5.2V
Communication Protocol	I2C
Motor Drive Output	PWM
Max Motor Drive Voltage	5.2V
Max Motor Drive Current	200mA
Program Memory Size	2KB
Operating Temperature Range	-40°C to +85°C
Supported Waveforms	LRA, ERM, SMT (synchronous motor)
Dimensions	20mm x 25mm x 3.4mm
Weight	2.5g

4.2.1.4 Vibrating Mini ERM Motor Disc

The Vibration Motor Disc was an ideal choice for our prototype due to its compact size and low power consumption. By soldering it to the Adafruit DRV2605L Haptic Motor Controller as shown in Figure 4.7, we were able to easily generate a wide range of vibration patterns and intensities. This setup offers precise control over the frequency and vibration strength, enabling us to create a variety of haptic feedback effects for our application [10].

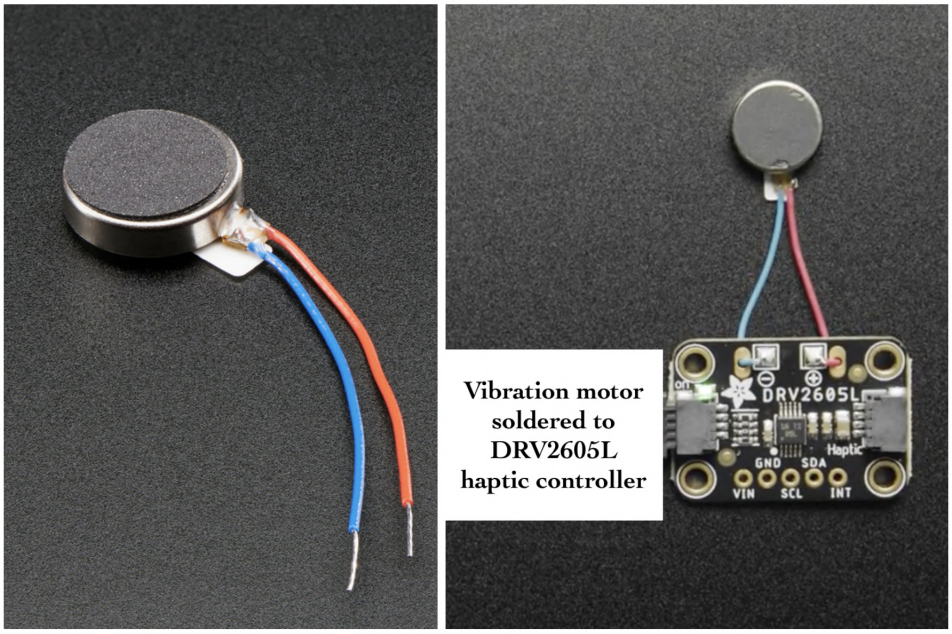

Figure 4.7: Vibration mini ERM motor disc [5].

Table 4.3: Technical specification of Vibrating Mini ERM Motor Disc [10].

Technical specifications	
Operating Voltage Range	2.0V to 5.2V
RPM (Rotations per minute)	11000 RPM at 5V
Current Draw (5V)	100mA
Current Draw (2V)	40mA
Dimensions	10mm diameter, 2.7mm thick
Weight	0.9 gram

4.2.1.5 Adafruit LiIon or LiPoly Charger BFF Add-On for QT Py

In our prototype, we utilized the Adafruit LiIon or LiPoly Charger, soldering it to the back of the Adafruit QT Py ESP32-S2 with pins 3V and GND. This charger add-on allows for seamless battery management, eliminating the need to connect the battery directly to the microcontroller as shown in Figure 4.8. The LiPoly Charger enables easy on/off control of the device and charges the battery when a USB-C cable is connected to the Adafruit QT Py ESP32-S2. This makes our wearable device rechargeable and more user-friendly, ensuring efficient power management for portable applications [11].

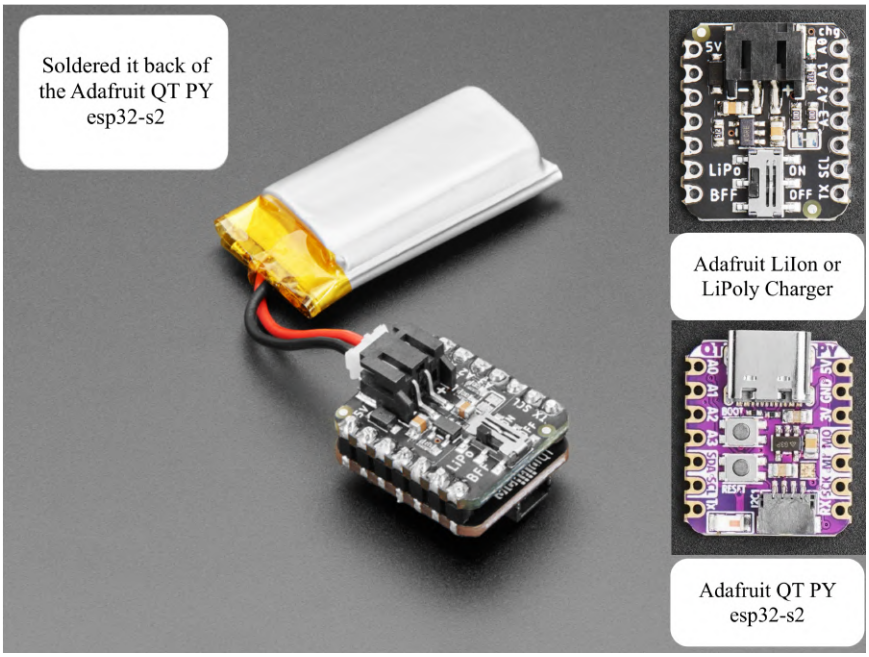

Figure 4.8: Adafruit LiIon or LiPoly Charger.

Table 4.4: Technical specification of Adafruit LiIon or LiPoly Charger [11].

Technical specifications	
Charging Voltage	4.2V
Charge Current	200mA
Input Voltage	5V via USB Type-C connector on QT Py
Battery Compatibility	Li-ion or LiPoly, single cell
Output Connector	JST-PH 2-pin
Dimensions	20.9mm x 17.9mm
Additional Features	Built-in slide switch for on/off control, Charge LED indicator

4.2.1.6 Large Tactile Push Button Switch

Large Tact Push Button Switch has been successfully integrated into the prototype and soldered to the analog A1 pin and GND of the Adafruit QT Py ESP32-S2 as demonstrated in Figure 4.9. The decision to use a single button was made to ensure easy operation for all users, regardless of their condition. The mechanical button is highly responsive to even small forces, making it ideal for acknowledging alerts. However, button bounce refers to the rapid fluctuation of the electrical signal when a button is pressed or released, leading to unintended or erratic behaviour. To address this issue, the "bounce2" library was employed, requiring the button to be stably pressed for a certain duration before registering input. This software solution effectively mitigates the effects of button bounce, ensuring that the button functions properly in the prototype system. [12].

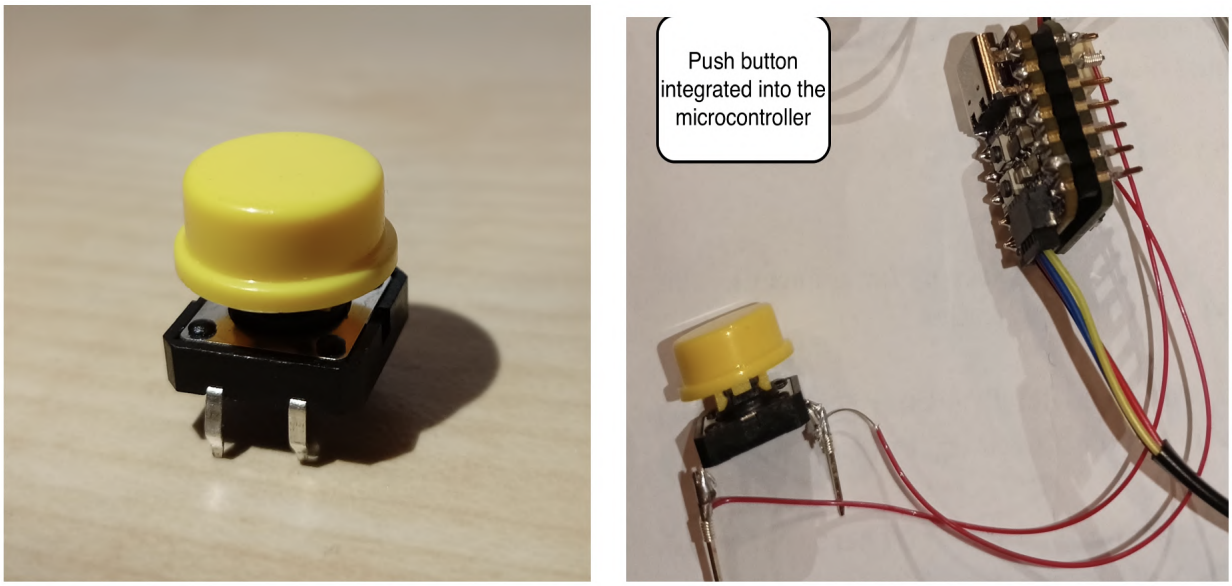

Figure 4.9: Push button.

Table 4.5: Technical specification of Large Tactile Push Button Switch [12].

Technical specifications	
Operating force (OF)	2.55N ± 0.69N
Releasing Force (RF)	0.49N minimum
Durability	1,000,000 operations minimum.
Bounce time	5ms maximum
Contact resistance (initial value)	100mΩ maximum.
Dimensions and Type	12X12mm projected

4.2.1.7 Lithium Ion Polymer Battery (3.7V 400mAh)

In Figure 4.10 Lithium Ion Polymer Battery (3.7V 400mAh) is shown, it is a lightweight and powerful battery that is ideal for our prototype. With a capacity of 400mAh, it provides approximately 1.9 Wh (Watts per hour) of energy. The output voltage ranges from 4.2V when fully charged to 3.7V. To recharge the battery safely at a rate of 400mA or less, it is essential to use a specialized LiIon/LiPoly constant-voltage/constant-current charger, such as the Adafruit LiIon or LiPoly Charger. The battery features built-in protection circuitry that prevents overcharging, over-discharging, and short circuits [6].

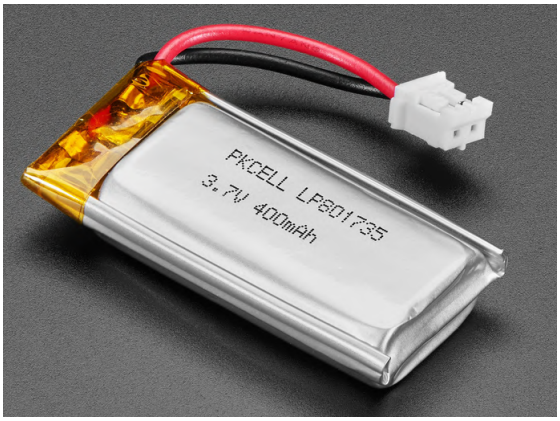

Figure 4.10: Lithium Ion Polymer Battery (3.7V 400mAh) [6].

4.2.1.8 STEMMA QT JST SH 4-pin Cable

The STEMMA QT JST SH 4-pin Cable played a significant role in connecting the Adafruit QT Py ESP32-S2 to the Adafruit DRV2605L Haptic Motor Controller in our prototype as shown in Figure 4.11. The four pins on the cable are labelled VCC, GND, SDA, and SCL, which respectively provide power, ground, and I2C communication for the devices [7].

Figure 4.11: Connection using STEMMA QT JST SH 4-pin [7].

4.2.1.9 Designed Prototype 3D model

In Figure 4.12, two models designed in Autodesk's Fusion 360 software are depicted. Both models have the same internal structure. The first model is designed to be worn around the neck, while the second model is designed to be worn on the hand. Figure 4.13 shows the band designed specifically for the second model. For the first model, a chain or thread can be passed through the hole.

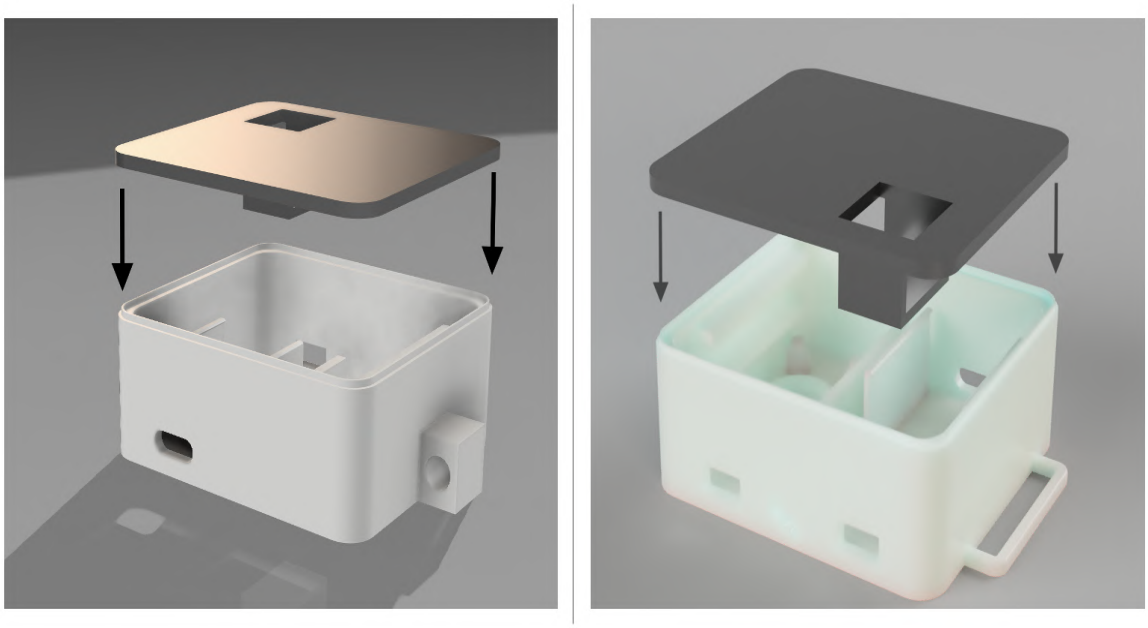

Figure 4.12: 3D design of the wearable part of the system.

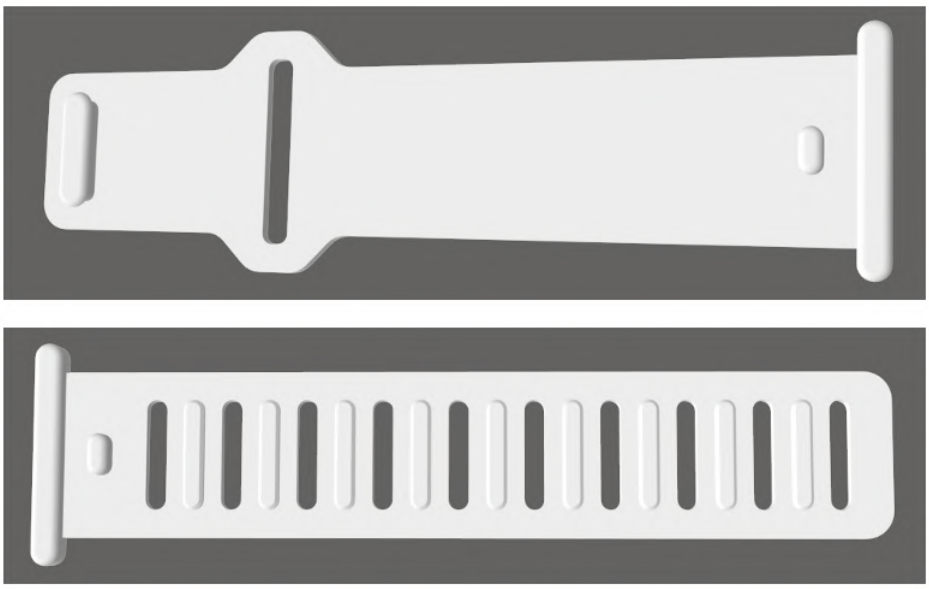

Figure 4.13: 3D designed band straps for the watch head.

4.2.2 Assembly

Figure 4.15 showcases the prototype watch head, which has been fabricated using PLA material through the process of 3D printing. The straps of the watch, on the other hand, are 3D printed using TPU 95A material, known for its flexible and elastic properties.

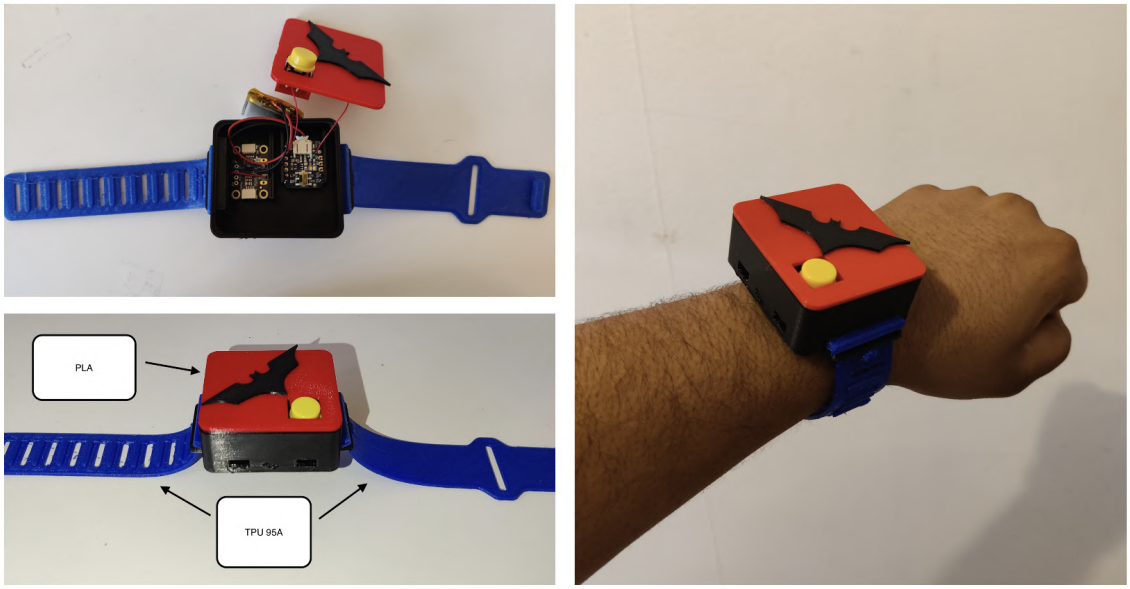

Figure 4.14: Assembled prototype.

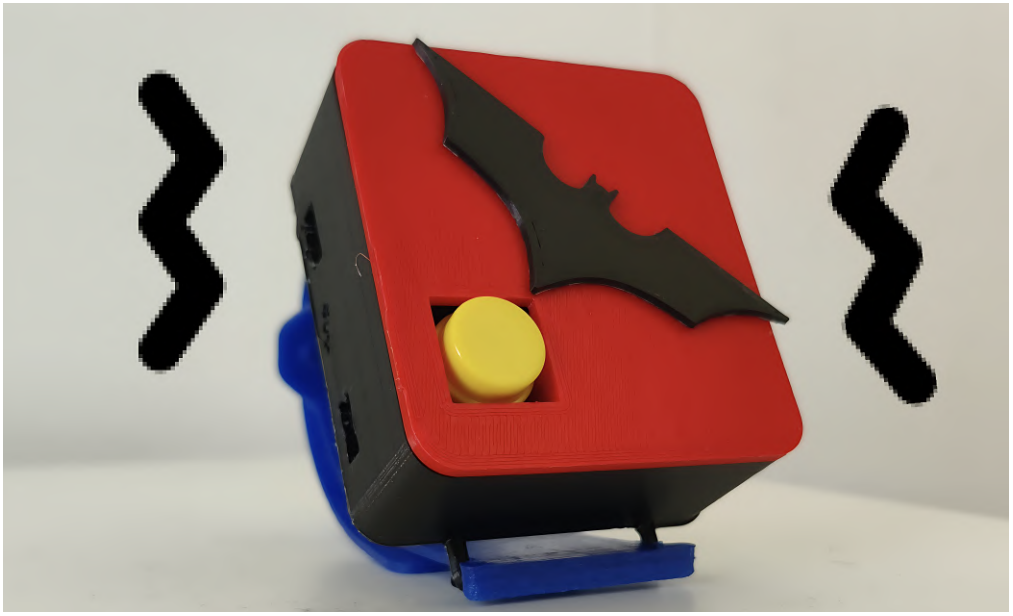

Figure 4.15: Assembled prototype.

4.2.3 Software Implementation

In the section, "Hardware Implementation (4.2.1)," we discussed the components used and the connections between them to develop a prototype model. This section addresses the software aspects, including how the model functions and the software tools used.

Our developed system is designed to cater to both our client, who manages the alerting system and sends alerts, and the client's users who receive the alerts. The prototype focuses on providing key functionalities such as enabling the client to send alerts to the users and allowing the users to provide information back to the client. In order to facilitate this communication, we have implemented a curated page on the client's end, utilizing Adafruit IO [34]. This curated page serves as the central hub for managing and monitoring the alerts and interactions between the client and the users. Now, when it comes to the user wearing the prototype, we need to program it to respond appropriately when specific messages are received from the client and user. To achieve this, we are using the Arduino IDE for programming the prototype.

In this section, we will discuss the programming languages and platforms we have employed, and explain why we chose them over the numerous available options. We will also delve into how we integrated these various technologies.

4.2.4 Programming Languages and Platforms

As mentioned earlier, we will now delve into the details of Adafruit IO and Arduino IDE, explaining their specific roles in our project.

4.2.4.1 Adafruit IO

Adafruit IO is a versatile Internet of Things (IoT) platform that enables seamless connectivity and management of devices. It's user-friendly interface and powerful features make it an ideal choice for IoT projects. Our project utilises the Adafruit QTPy ESP32-S2 microcontroller, designed by the same company behind Adafruit IO. This microcontroller integrates seamlessly with the Adafruit IO platform.

Adafruit IO is an advanced cloud-based platform designed for managing and controlling Internet of Things (IoT) devices and data. At its core, Adafruit IO operates using three key components: feeds, dashboards, and actions [34].

- **Feeds:** Adafruit IO's feed acts as a repository for data pushed to the platform and contains important metadata about the data. This metadata includes privacy settings, licenses, and detailed descriptions. In addition, feeds serve to store sensor data values sent by devices. A separate feed is required for each individual data source to properly organize and manage data within the platform [35].

- **Dashboard:** Adafruit IO's dashboard easy-to-use web interface that allows users to visualize and interact with IoT data. Dashboards allow users to create customizable views, charts, and graphs to monitor and analyze their feeds. Real-time updates are provided, making it easier to track and respond to data changes [35].
- **Actions** Actions in Adafruit IO allow users to automate certain tasks or trigger events based on certain conditions. Users can define rules and conditions. If these conditions are met, the appropriate action is taken. For example, an action could be sending a notification or controlling a connected device based on certain data. Additionally, Adafruit IO offers seamless integration with popular automation platforms [36].

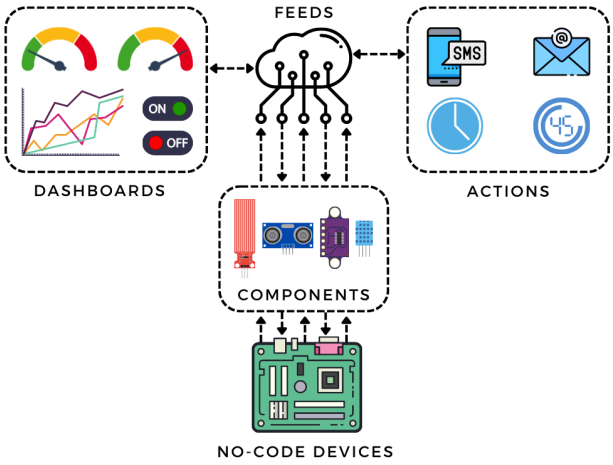

Figure 4.16: Adafruit IO Architecture [8].

Figure 4.16 shows the architecture of the Adafruit IO's cloud-based platform. Adafruit IO's compatibility with microcontrollers from different manufacturers is one of its standout characteristics. Microcontrollers may easily connect to the Adafruit IO platform by utilizing the Adafruit IO library. For establishing the connection, this library provides a standardized and user-friendly interface. Developers must enter their special ADAFRUIT IO KEY, which consists of a username and an active key, to authenticate the connection. The security and privacy of the IoT ecosystem are ensured by this authentication process, which makes sure that only authorized users may access and interact with the platform [37].

4.2.4.2 Arduino IDE

Arduino IDE is an open-source software platform that facilitates coding and uploading to microcontroller boards, such as the Adafruit QT Py ESP32-S2, using C and C++ languages. Compared to CircuitPython, Arduino IDE is often favoured due to its extensive library support, enabling the development of a more diverse range of projects, and providing users with greater flexibility and functionality [38].

4.2.4.3 IFTTT

IFTTT, short for "If This Then That," is seamlessly integrated into Adafruit IO, empowering users with advanced automation features. IFTTT allows the creation of applets that trigger specific actions based on predefined conditions. Users can automate tasks and facilitate seamless interactions by linking Adafruit IO feeds with various services like email, social media, or smart devices. This integration expands the capabilities of sending emails from Adafruit IO [39].

4.2.4.4 Integration of Adafruit IO and Arduino IDE

In the previous section, we discussed Adafruit IO and Arduino IDE, why we opted for them, and how they work. Now, in this section, we will discuss the integration of Adafruit IO with various components using the Arduino IDE.

To utilize the Adafruit QT Py ESP32-S2 with the Arduino IDE, it is necessary to configure the IDE to be compatible with the board, as this board is not manufactured by Arduino. This involves adding additional boards to the board manager URL in the settings, as shown in Figure 4.17. Since we are using the Espressif ESP32 chip in our microcontroller, we have added the URL of the Espressif ESP32 boards and downloaded the esp32 package from the board manager. This process ensures successful communication between the IDE and the microcontroller, facilitating seamless coding and uploading processes [37].

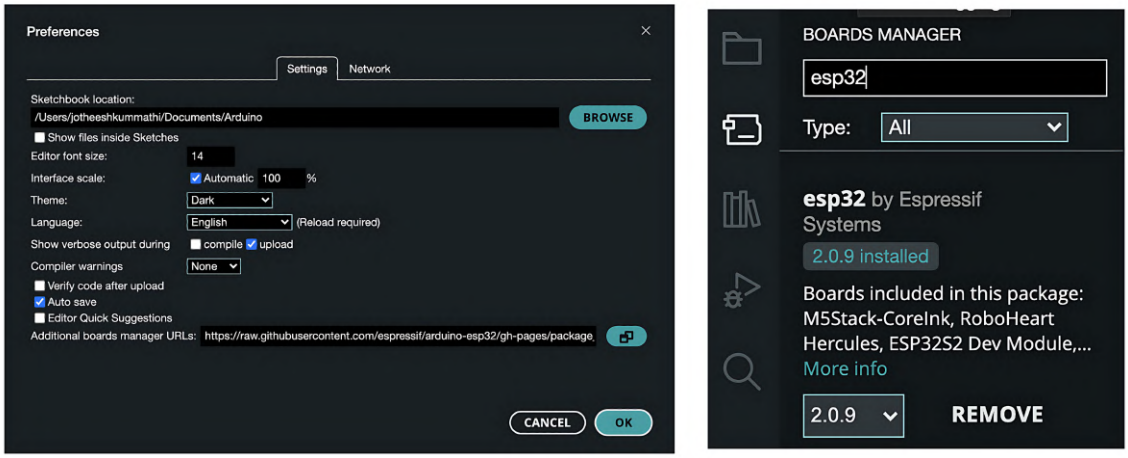

Figure 4.17: Arduino Settings tab.

After making our microcontroller compatible with the Arduino IDE, we established a connection between AdafruitIO and the microcontroller using the unique key provided by AdafruitIO, as discussed above. In Figure 4.18, the first image shows the key obtained from AdafruitIO, while the second image displays the Arduino IDE code where the AdafruitIO key is used to establish a connection with AdafruitIO.

This technical integration ensures communication between the microcontroller and the AdafruitIO platform.

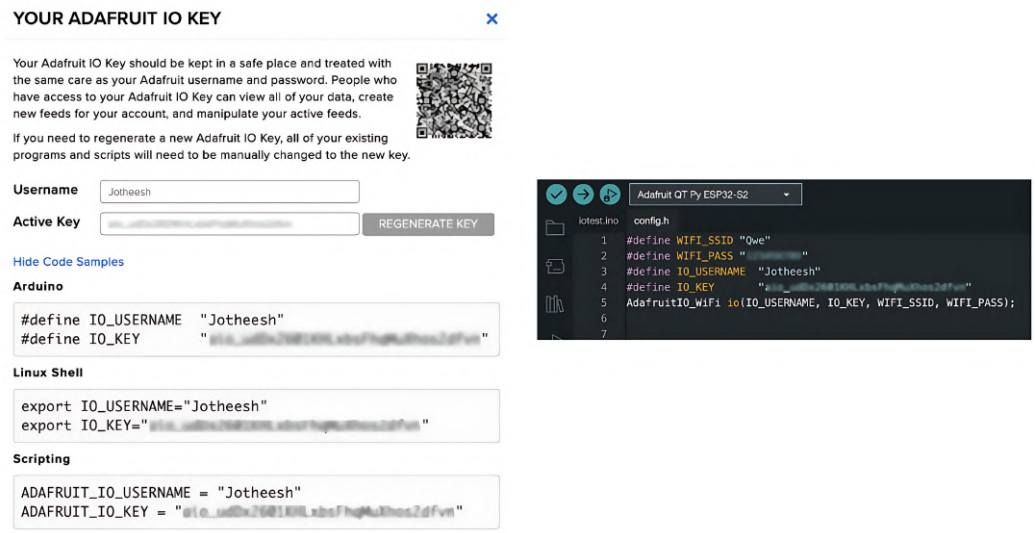

Figure 4.18: AdafruitIO key.

In Figure 4.19, a snippet from the Arduino IDE is displayed, showcasing the libraries utilized in our project.:

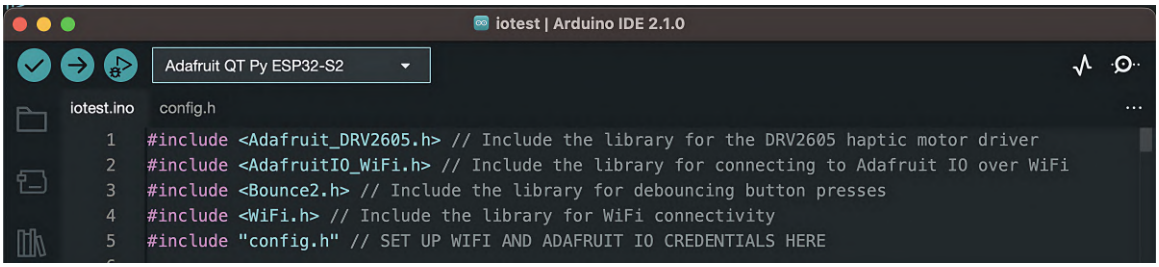

Figure 4.19: Libraries used in the project.

- **Adafruit_DRV2605.h**

  The primary actuator in our system is a vibration motor. Generally, the motor's vibration pattern can be designed using delays. However, in this case, we have connected it to the DRV2605L haptic controller. The associated Adafruit_DRV2605.h library has 123 predefined vibration patterns, offering a robust selection for customization. This approach offloads some processing demands from the microcontroller, an important consideration given that the ESP32 serves dual roles as a Wi-Fi chip and a microcontroller [5].

- **AdafruitIO_WiFi.h**

  The AdafruitIO WiFi library is utilized to establish a connection between the system and the Adafruit IO platform via WiFi, extending WiFi.h functionality offering transmitting and receiving data from Adafruit IO feeds, as well as managing Adafruit IO communication. Within the code, an AdafruitIO_WiFi object is created and used to connect to Adafruit IO by providing the username, key, and WiFi credentials as parameters [34].

- **Bounce2.h**

  The Bounce2 library is employed for button debouncing to eliminate erratic button triggers caused by mechanical inconsistencies. It ensures reliable and accurate button input, particularly for detecting specific patterns like the SOS function, by mitigating the effects of bouncing and ensuring a single button press is registered [40].

- **WiFi.h**

  The WiFi.h library is included using #include <WiFi.h> and is used to connect to WiFi networks. It provides functions and methods for connecting to WiFi networks, checking the connection status, and obtaining IP addresses. In the code, it is used to connect to WiFi networks using the given SSID and password [34].

- **config.h**

  In the configuration file, typically named "config.h", you will find all the necessary credentials for Adafruit IO and multiple WiFi networks, as well as feeds, are defined. This separation allows users to easily modify the credentials without the need to make changes to the main code. By keeping the credentials separate, the risk of corrupting the main code is minimized. This design promotes flexibility and ease of maintenance.

The above information provides a clear understanding of the Arduino IDE code, as provided in the appendix. Moving on to Adafruit IO, Figure 4.20 illustrates the dashboard within Adafruit IO, where alerts can be sent and acknowledged. Additionally, an SOS signal is displayed. This dashboard represents our prototype, which includes two devices: Device 1 and Device 2. For sample alerts, we have considered "fire," "zombie," and "both" (indicating both fire and zombie incidents). Figure 4.21 showcases the feeds that were used to develop these dashboards.

One more feature in our system is that whenever an alert is sent to the user, it simultaneously sends an alert to a certain emergency department. Additionally, it sends another alert based on the user's acknowledgement. The sent alert is through email by integrating IFTTT in Adafruit IO. IFTTT works based on a condition: if a certain feed value is attained, it sends an email. This function is called an applet in IFTTT as shown in Figure 4.22 [39].

Figure 4.20: Dashboard in AdafruitIO.

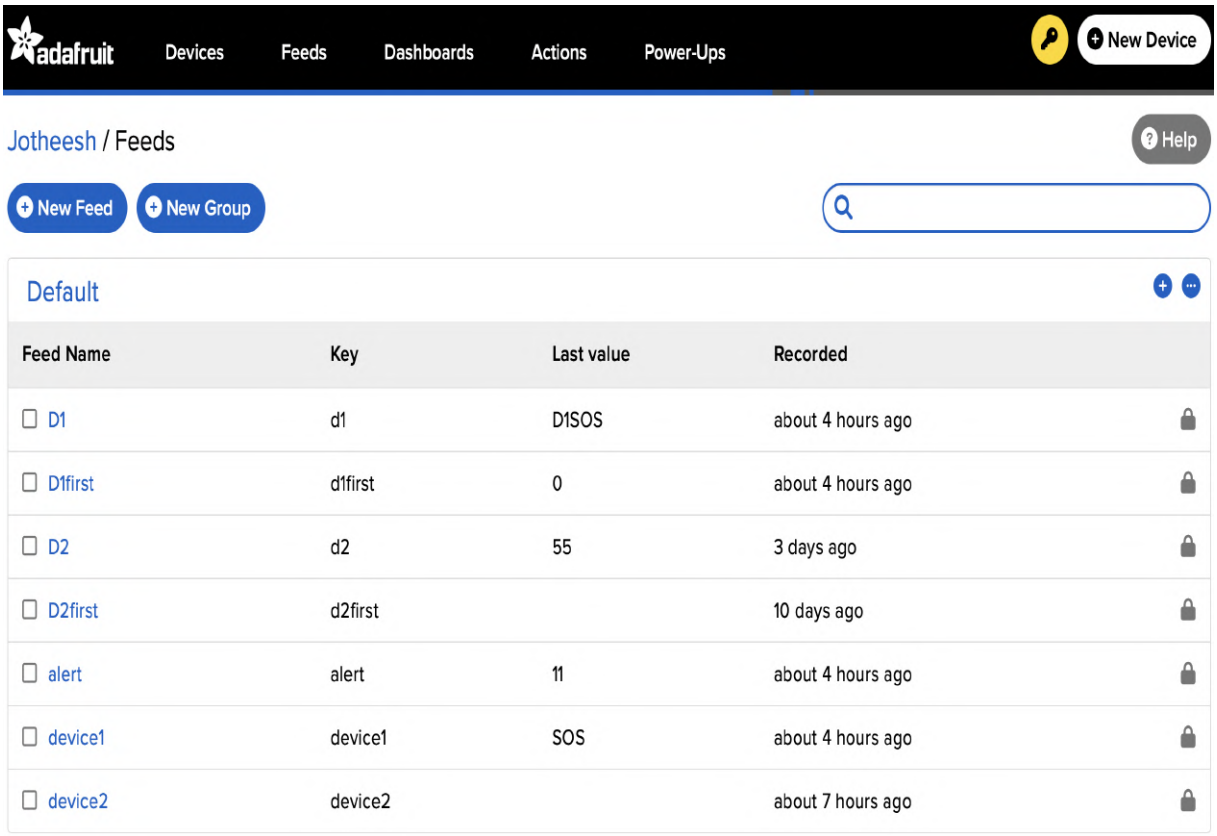

Figure 4.21: Feeds in AdafruitIO.

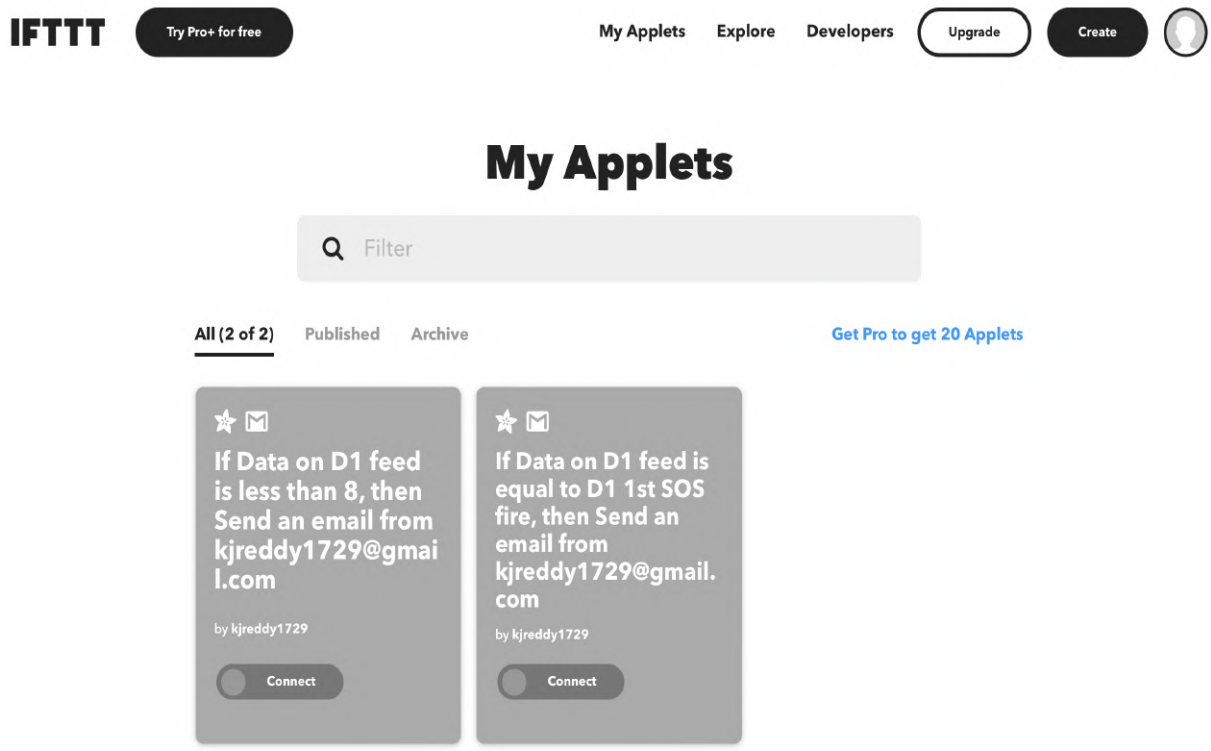

Figure 4.22: Applets in the IFTTT.

Before diving into the implementation flow, let's gain a clear understanding of the blocks used in the Adafruit IO dashboard and their respective roles. Our dashboard primarily consists of three blocks:

1. **Toggle button**

   The toggle button block provides a user interface element that allows users to switch between two states, typically on and off. It enables us to send an alert to a device, triggering a specific vibration pattern assigned to it. Additionally, data received from the feed can turn off the toggle. Values of the toggle button assigned feeds are shown in Table-4.6.

2. **Status indicator**

   The status indicator block visually represents the current status or condition of a specific device or data feed. In our system, it shows green if it receives an acknowledgement from the user after sending an alert. If the acknowledgement is not received, it shows the colour red. The colours can be customizable, but for ease of universal understanding, we have utilized red and green. Values of the status indicator conditions are shown in Table-4.7.

3. **Text block**

   The text block displays textual information or messages on the dashboard. It allows for the presentation of important information, notifications, or instructions in a text format. The acknowledgements from the user are displayed in text format for a better understanding of the condition.

Table 4.6: Toggle button assigned feeds and their values.

Toggle button assigned feeds and their values			
Alert feed	**Type of alert**	**ON**	**OFF**
Alert feed	Fire	1	11
	Zombie	2	22
	Both (fire and zombie)	3	33
D1 feed	Fire	4	44
	Zombie	6	66
	Both (fire and zombie)	8	88
D2 feed	Fire	5	55
	Zombie	7	77
	Both (fire and zombie)	9	99
Note: Any value other than the ON value in the feed will turn OFF the toggle			

Table 4.7: Status indicator conditions to show green colour.

Status indicator conditions to show green colour				
Devices	**Acknowledgement**	**Feed name**	**Alert type**	**True condition**
Device 1	1st Acknowledgement	D1first	Fire	1
			Zombie	2
			Both(fire and zombie)	3
	2nd Acknowledgement	D1	Fire	D1 safe from fire
			Zombie	D1 safe from zombie
			Both(fire and zombie)	D1 safe from both
Device 2	1st Acknowledgement	D2first	Fire	1
			Zombie	2
			Both(fire and zombie)	3
	2nd Acknowledgement	D2	Fire	D2 safe from fire
			Zombie	D2 safe from zombie
			Both(fire and zombie)	D2 safe from both

4.2.5 Implementation Flow

Our system is developed for multiple devices, and the implementation is the same for all of them. Therefore, we will discuss the implementation flow of the three main functionalities in your system, assuming it is consistent across all devices. Let us proceed to discuss the implementation flow of these four main functionalities in our system:

1. Device connectivity to WiFi and AdafruitIO.
2. User-initiated SOS transmission from the device.
3. Alert transmission from AdafruitIO and reception of an acknowledgement from the user after sending the alert.
4. Long-pressing the button communicates a 'SAFE' message to the dashboard.

1. Device connectivity to WiFi and AdafruitIO.

In Figure 4.23, the flow of device connectivity to WiFi and Adafruit IO is illustrated. When the device starts, it attempts to connect to multiple WiFi credentials in iterations. If any of the given WiFi credentials successfully connect, the device then establishes a connection to Adafruit IO. However, if it fails to connect to any WiFi network within 5 minutes, the haptic controller triggers the vibration motor to indicate that the device is not connected to WiFi. If the device loses the connection, it tries again to establish it.

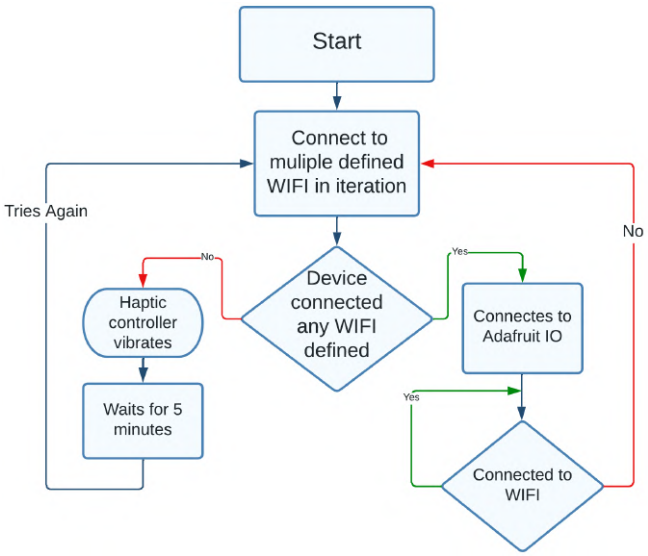

Figure 4.23: Flow chart of device connectivity.

If the device successfully connects to both WiFi and Adafruit IO, but later gets disconnected, it will initiate the process of connecting to WiFi again. This looping process allows the user to stay connected regardless of their location, whether it's in an office, house, or any other place.

2. User-initiated SOS transmission from the device.

The SOS triggering process is two conditions one triggered while connecting to WiFi and AdafruitIO and another after connected to WiFi and AdafruitIO

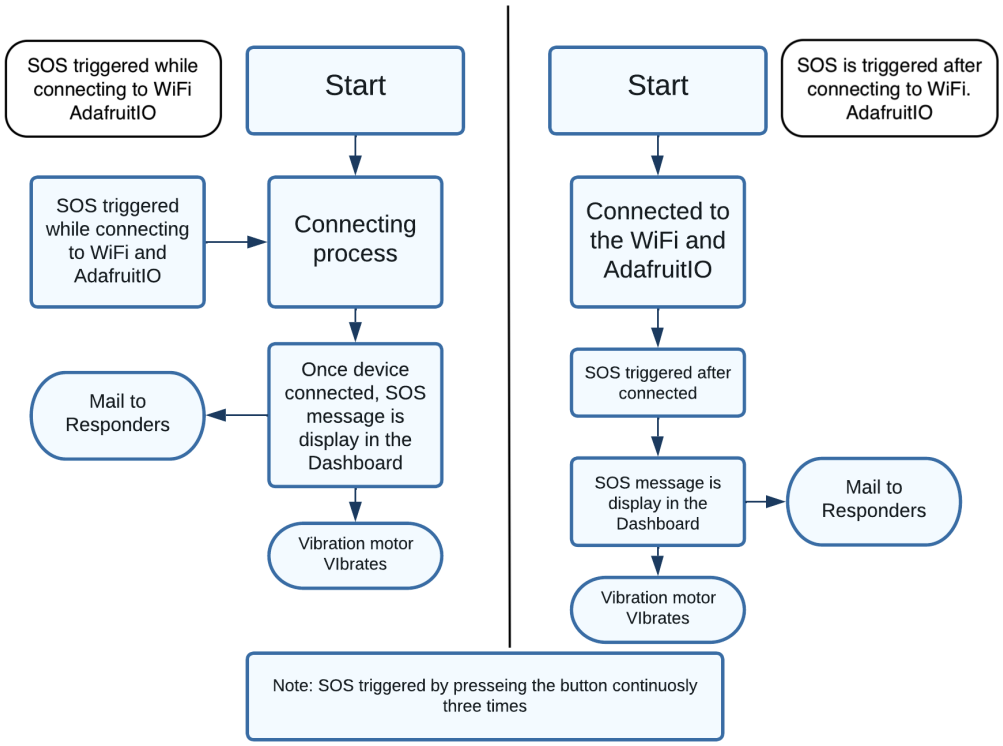

Figure 4.24: Flow chart of SOS triggering processes.

In Figure 4.24, the left side depicts the process for triggering an SOS. When the SOS button is pressed three times, the device waits for a connection to be established. Once the device successfully connects to WiFi, it proceeds with the following steps:

1. Intimate SOS to Dashboard: The device sends an SOS message to the dashboard, indicating an emergency situation.

2. Vibration Motor Confirmation: The vibration motor vibrates as confirmation that the SOS message has been successfully sent.

The process remains the same for an SOS triggered after the device has already connected to WiFi and Adafruit IO. In this case, the device directly sends the SOS message to the dashboard, and the vibration motor provides confirmation.

3. Alert transmission from AdafruitIO and reception of an acknowledgement from the user after sending the alert.

Regarding the main functionality of your system, as we discussed earlier when the alert feed is set to 1, it will activate Device 1 and Device 2 by setting their respective "on" values. Alternatively, you can directly activate Device 1 by pressing the toggle button associated with it. This process remains the same for all the alerts, and although the toggle buttons share the same feed, different numbers are assigned to each toggle button to ensure data integrity as shown in Figure 4.25.

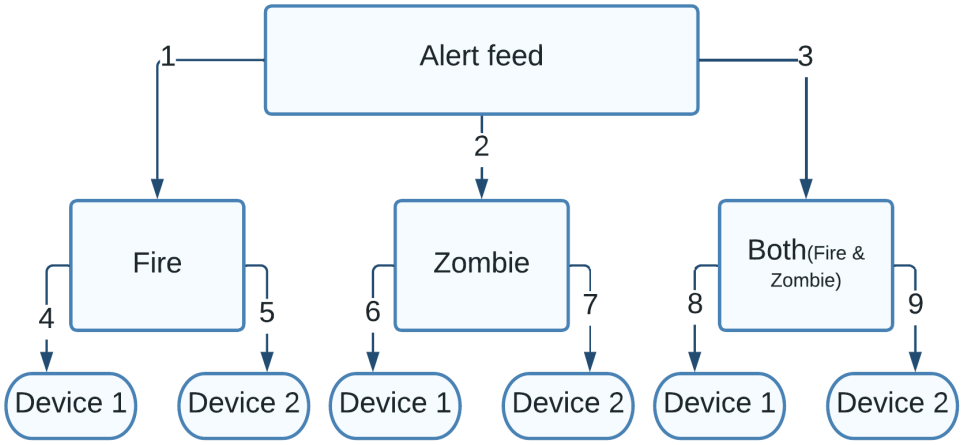

Figure 4.25: Toggle button values assigned to each alert.

The above process describes what happens when the alert is sent. Now, let us move on to what happens in the device when a message is received. Figure 4.26 depicts the flowchart of what happens in the device after receiving an alert from the Adafruit dashboard through feeds alert and D1. Since the process is the same for all the scenarios, we will consider the fire alert to describe the process.

When the feed value is received from AdafruitIO, it simultaneously sends a mail based on the feed value to specified emergency services. In the case of a fire alert (assuming the alert value is either 1 or 4), the device enters a specific alert mode. The vibration motor will vibrate for one minute if the user has not pressed the button. After one minute, if no button press is detected, the vibration stops, and an acknowledgement message "D1 1st SOS fire" is sent. The dashboard receives this message and sends an alert for immediate help.

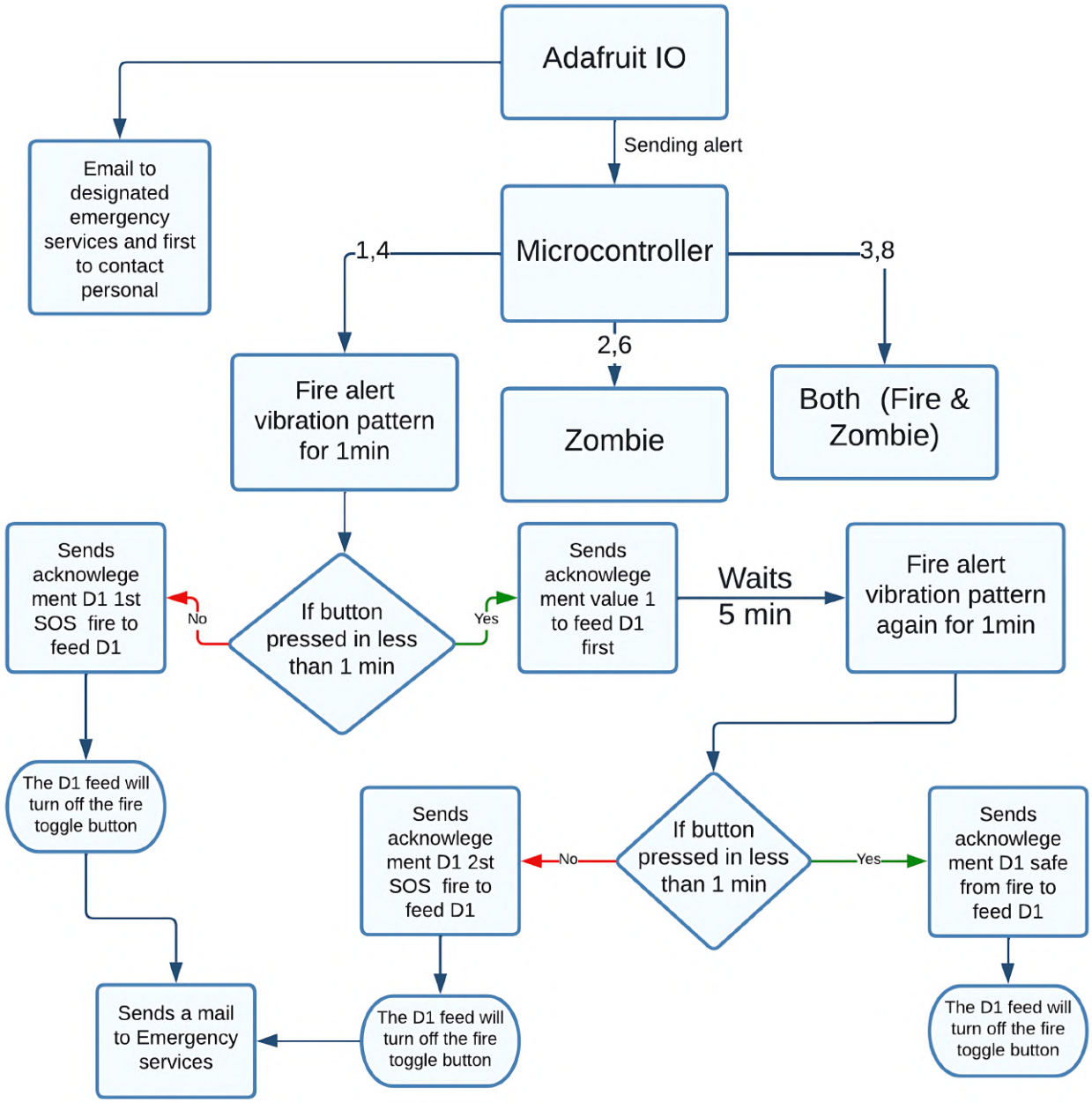

Figure 4.26: Flowchart of the functionality.

If the button is pressed, indicating user acknowledgement, the device sends a value of "1" to the "D1first" feed, and that triggers the color green on the status indicator in the dashboard indicating that the user has acknowledged the alert. The device then waits for 5 minutes and vibrates again for 1 minute. It waits for user acknowledgement through a button press. If no button press is detected, the device terminates the vibration and sends a message "D1 2nd SOS fire," along with sending a mail. If the button is pressed, indicating user safety, the device sends an acknowledgement message of "D1 safe from fire," toggles the button off, and displays a green colour in the second acknowledgement. All the text messages are displayed in the text block of the dashboard.

4. Long-pressing the button communicates a 'SAFE' message to the dashboard.

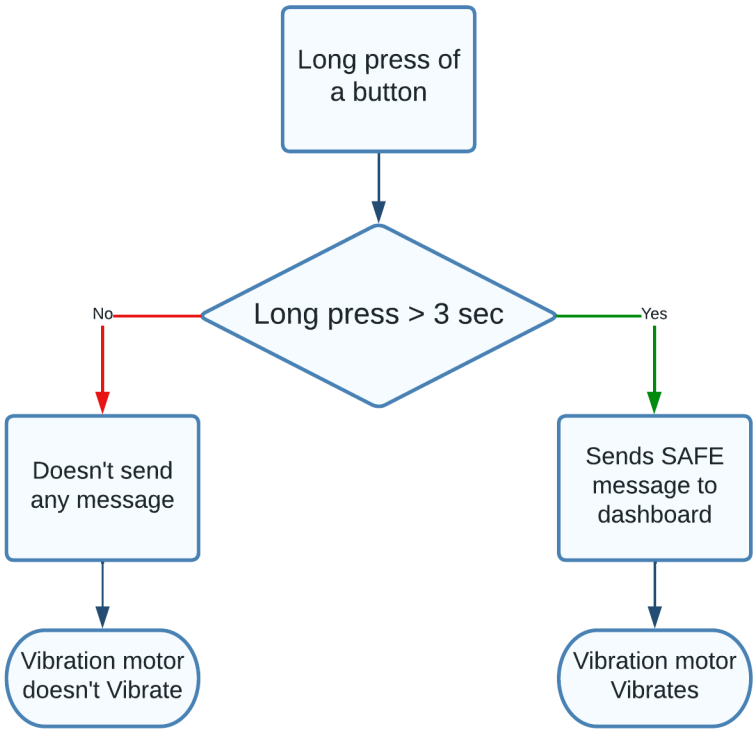

Figure 4.27: Flow-chart of the sending 'SAFE' message.

As the name suggests, this functionality is activated when a user presses and holds a button for more than 3 seconds. This action sends a 'SAFE' message to the dashboard, which subsequently sets the status indicator to a green color. Concurrently, the vibration motor activates as a form of confirmation after the message is sent. This feature aids the user at the dashboard by reassuring them that the wearer of the device is safe. The utility of this functionality will be validated in the subsequent validation section.

Chapter 5

Validation and Evaluation

In this chapter, we will first validate our project and subsequently evaluate the validation results.

5.1 Validation

As discussed in the implementation flow, our developed system comprises four prominent functionalities, and we will validate each of them:

1. Device connectivity to WiFi and AdafruitIO.
2. User-initiated SOS transmission from the device.
3. Alert transmission from AdafruitIO and reception of an acknowledgement from the user after sending the alert.
4. Long-pressing the button communicates a 'SAFE' message to the dashboard.

Before proceeding with the validation of these three functionalities, Figure 5.1 demonstrates that the code for the device has been successfully compiled and uploaded onto the device.

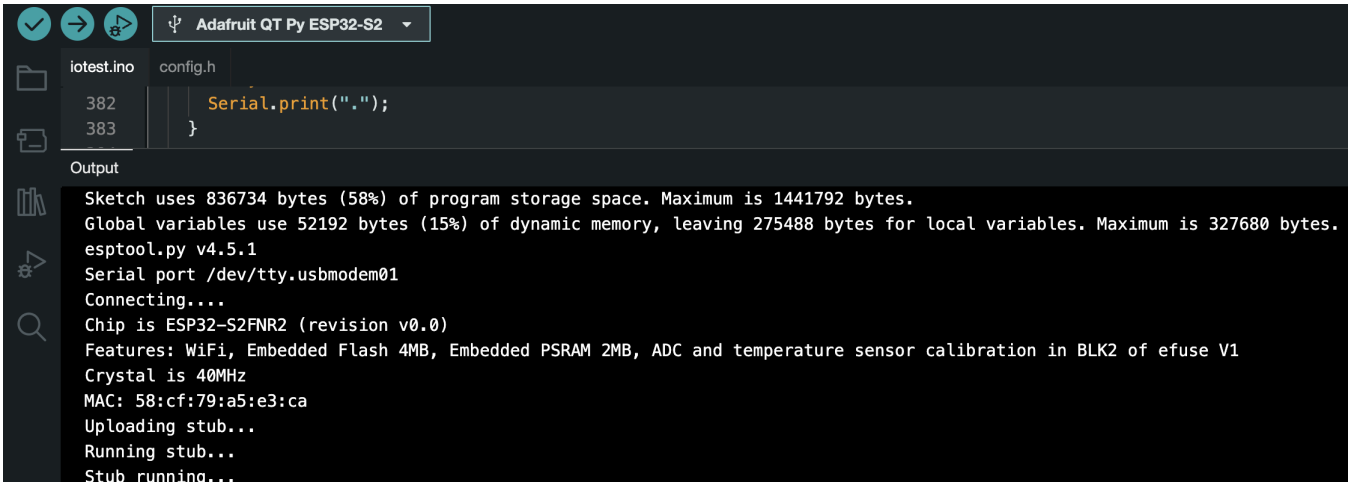

Figure 5.1: Image of the output window in Arduino IDE.

1. Device connectivity to WiFi and AdafruitIO.

As discussed earlier, our system attempts to connect to multiple WiFi networks. If no network is successfully connected within 30 seconds (for testing purposes), the haptic controller triggers the vibration motor.

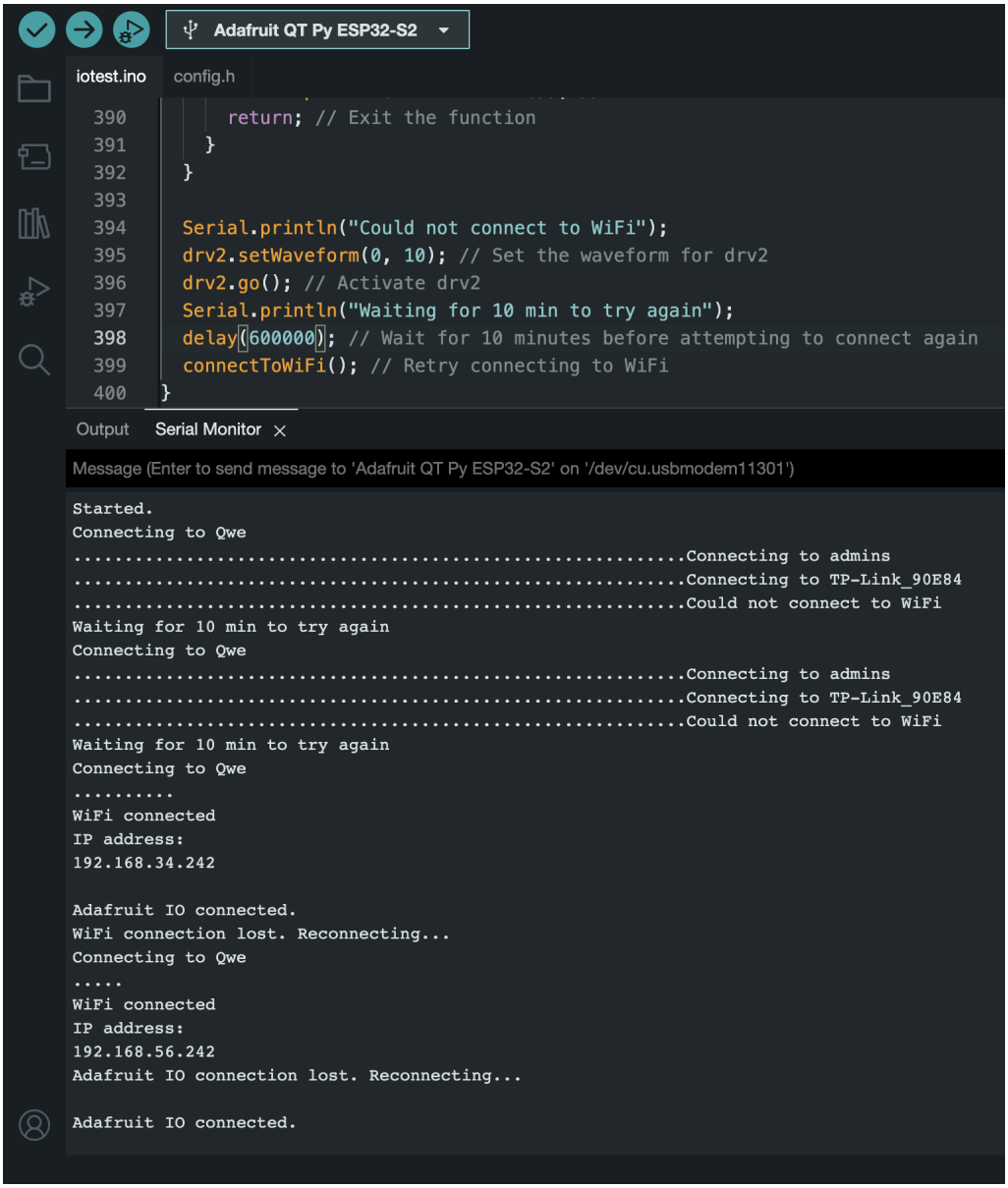

Figure 5.2: Device connectivity validation from the serial monitor of Arduino IDE.

Figure 5.2 demonstrates the feedback in a situation when the device starts, it attempts to connect to each WiFi network one by one, If none of the given credentials did not connected, as shown in the figure from code line 394, it prints "Could not connect to WiFi." The vibration motor vibrates to intimate the user, and the device will wait 10 minutes before restarting. This process continues until a successful WiFi connection is established. Once connected to WiFi, it promptly connects to Adafruit IO. In case the WiFi connection is lost, the device will make additional attempts to

reconnect.

2. User-initiated SOS transmission from the device.

To validate the SOS transmission from the device, the SOS trigger mechanism can be tested. The SOS is triggered when the button is pressed three times, and the time gap between each press is customizable. For testing purposes, it is set to 1 second, as shown in Figure 5.3.

```
81   if (button.fell()) { // Check if the button was pressed
82     unsigned long currentPressTime = millis(); // Get the current time
83     if (currentPressTime - lastPressTime < 1000) { // Check if the button was pressed within 1 second of the last press
84       pressCount++; // Increment the press count
85     } else {
86       pressCount = 1; // Reset the press count to 1
87     }
88
89     lastPressTime = currentPressTime; // Update the last press time
90
91     if (pressCount == 3) { // Check if the button was pressed three times consecutively
92       Serial.println("Help, SOS!"); // Print SOS message to Serial monitor
93       device1->save("SOS"); // Save SOS message to device1
94       sosTriggered = true; // Set the sosTriggered flag to true
95       D1->save("D1SOS"); // Save SOS message to D1
96       drv2.setWaveform(0, 10); // Set the waveform for drv2
97       drv2.go();
98       Serial.println("sos sent to dash board");
99       pressCount = 0; // Reset the press count
100    }
101  }
```

Figure 5.3: Code snippet of the SOS.

Now let us discuss the triggering process shown in Figure 5.4. There are three cases to consider:

1. The first case is when the user triggers an SOS after the device is connected to both WiFi and AdafruitIO.

2. The second case is when an SOS is triggered after the device has successfully connected to WiFi and AdafruitIO.

3. The third case is when an SOS is triggered during the device's connection process.

Figure 5.4 illustrates these three cases on the AdafruitIO dashboard.

In Figure 5.5, the serial monitor output is displayed when an SOS is triggered after the connection is established. The message "Help, SOS!" is printed. In dashboard case (2), after the SOS message is sent, the vibration motor vibrates as confirmation that the message has been successfully transmitted. For case (3), where the SOS is triggered during the device's connection process, a "sosTriggered" flag is utilized. Once the device establishes the connection, it immediately sends the SOS during the connection to the dashboard and triggers the vibration motor as confirmation.

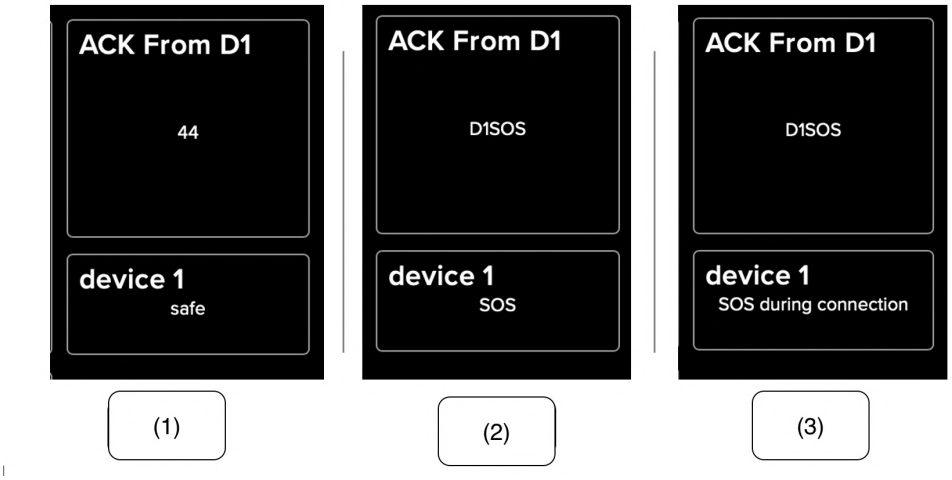

Figure 5.4: Dashboard of AdfruitIO for three SOS cases.

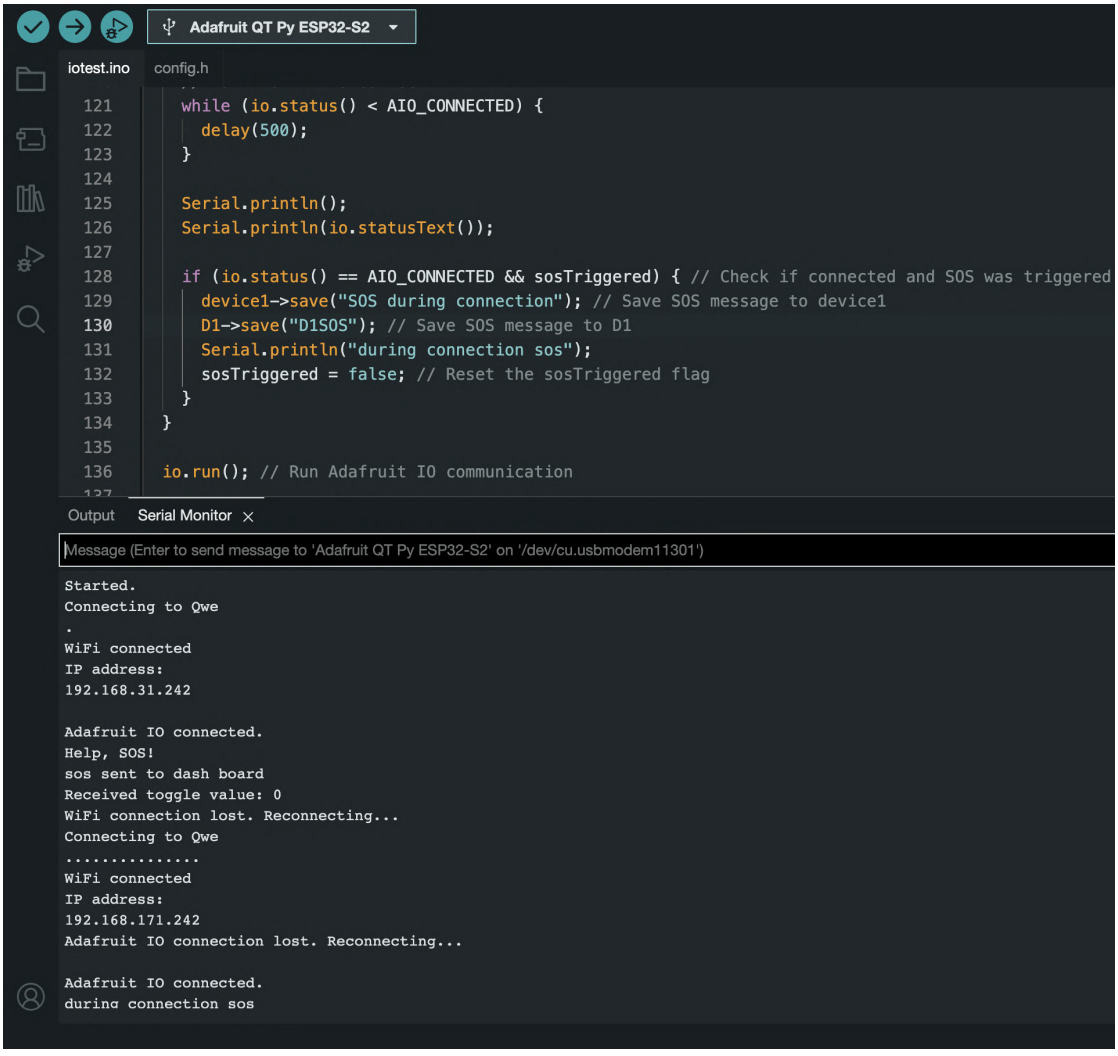

Figure 5.5: Serial monitor for the three cases.

3. Alert transmission from AdafruitIO and reception of an acknowledgement from the user after sending the alert.

To validate this functionality, we will consider one device for testing. As Figure 5.6 depicts, when a fire alert is triggered, both devices are turned on. Additionally, the fire alert can be sent to each device individually by activating the respective toggle button. For testing purposes, a temporary email address has been used to simulate the email notifications, as shown in Figure 5.6.

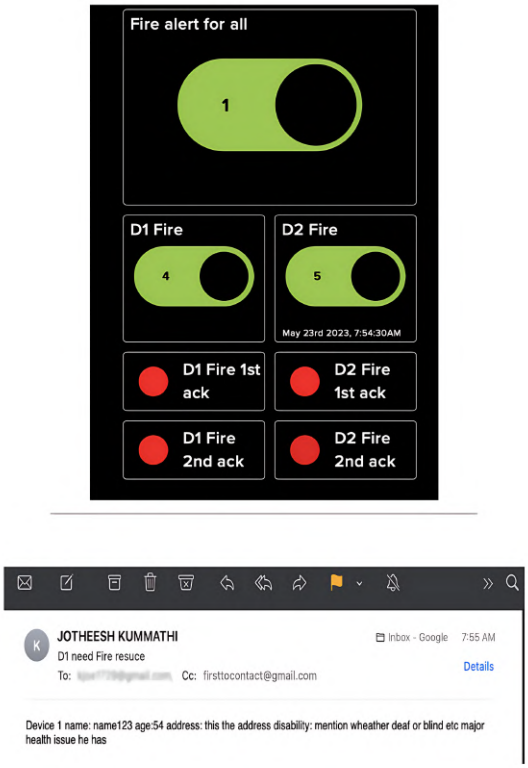

Figure 5.6: Dashboard of Adafruit IO when the alert is sent to the device.

Now that we have validated what happens in AdafruitIO when an alert is sent, the next step is to validate the device's behaviour. To achieve this, we have divided it into four cases, each of which will be validated below:

Case 1:

In the first case, as observed in the implementation flow, when any toggle button is pressed, the device receives the toggle value through the feed and enters an if statement based on the toggle value. The specific if statement to execute is determined by the toggle value, as indicated in Figure 5.7. Detailed information regarding the toggle values can be found in Table-4.6.

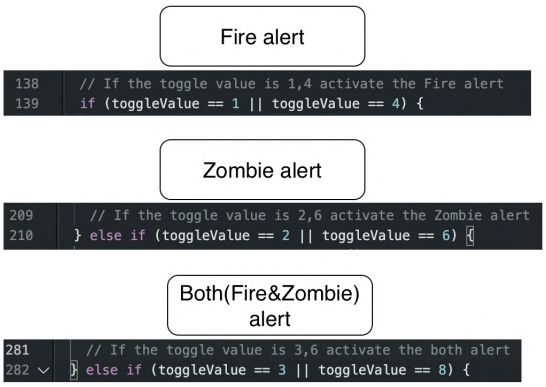

Figure 5.7: If Else if statement conditions for the alerts.

Case 2:

As discussed, the flow is the same for all alerts, with the vibration pattern changing based on the specific alert type. For validation purposes, we will focus on one alert scenario. When the device receives a toggle value of 1 or 4 as shown in Figure 5.8, it enters the corresponding if statement, activating the fire alert mode. The vibration motor will vibrate for 1 minute, but for testing purposes, a shorter duration of 5 seconds has been used.

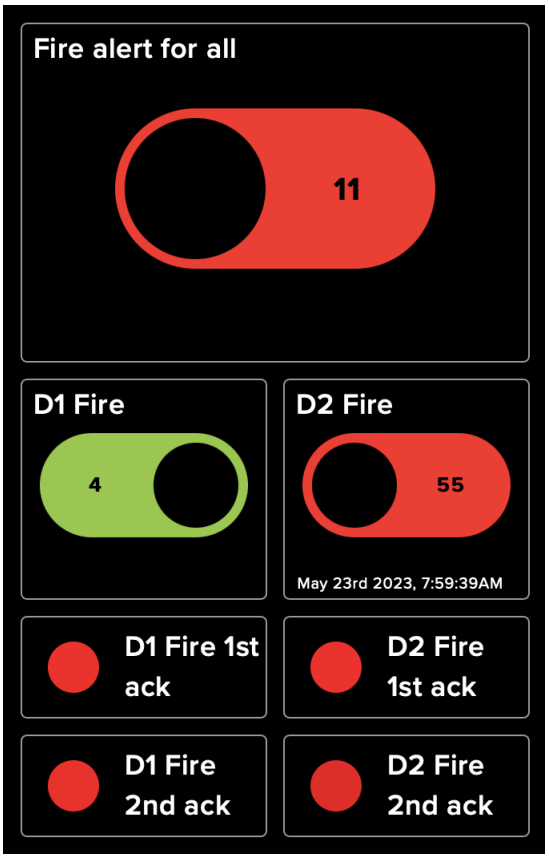

Figure 5.8: Dashboard from sending Device 1 fire alert.

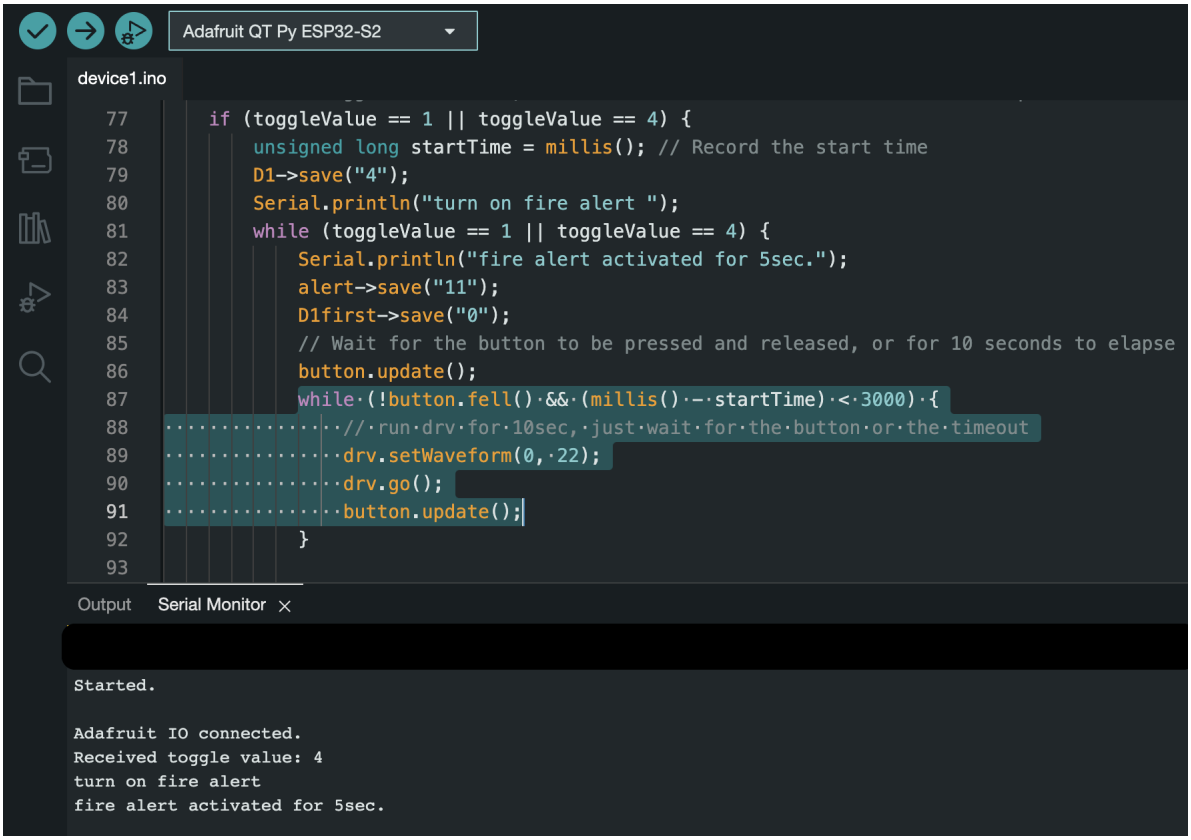

Figure 5.9: Code snippet and serial monitor after device receives the toggle value 4.

As shown in the code snippet in Figure 5.9, a while loop is implemented, which runs for 5 seconds. During this time, the device waits for a button press. If the button is pressed within the 5 seconds period, the loop breaks and further validation is performed in cases 3 and 4.

Case 3:

As observed in Figure 5.11, the fire alert is initially activated for a duration of 5 seconds. If the button is pressed to acknowledge the first alert, an acknowledgement message of "1" is sent to the AdafruitIO dashboard, which sets the first acknowledgement status indicator to green Figure 5.10. For testing purposes, the vibration motor stops for 5 seconds (rather than 5 minutes as intended), then activates again to indicate the second alert, running for 5 seconds.

If the button is pressed during the second alert, a "D1 safe from fire" message is sent to the dashboard, setting the second acknowledgement status indicator to green. Additionally, the corresponding text is displayed in the text block of the dashboard. Finally, the toggle button is turned off to signify the completion of the fire alert scenario as shown in Figure 5.10.

Figure 5.10: AdafruitIO dashboard for case 3.

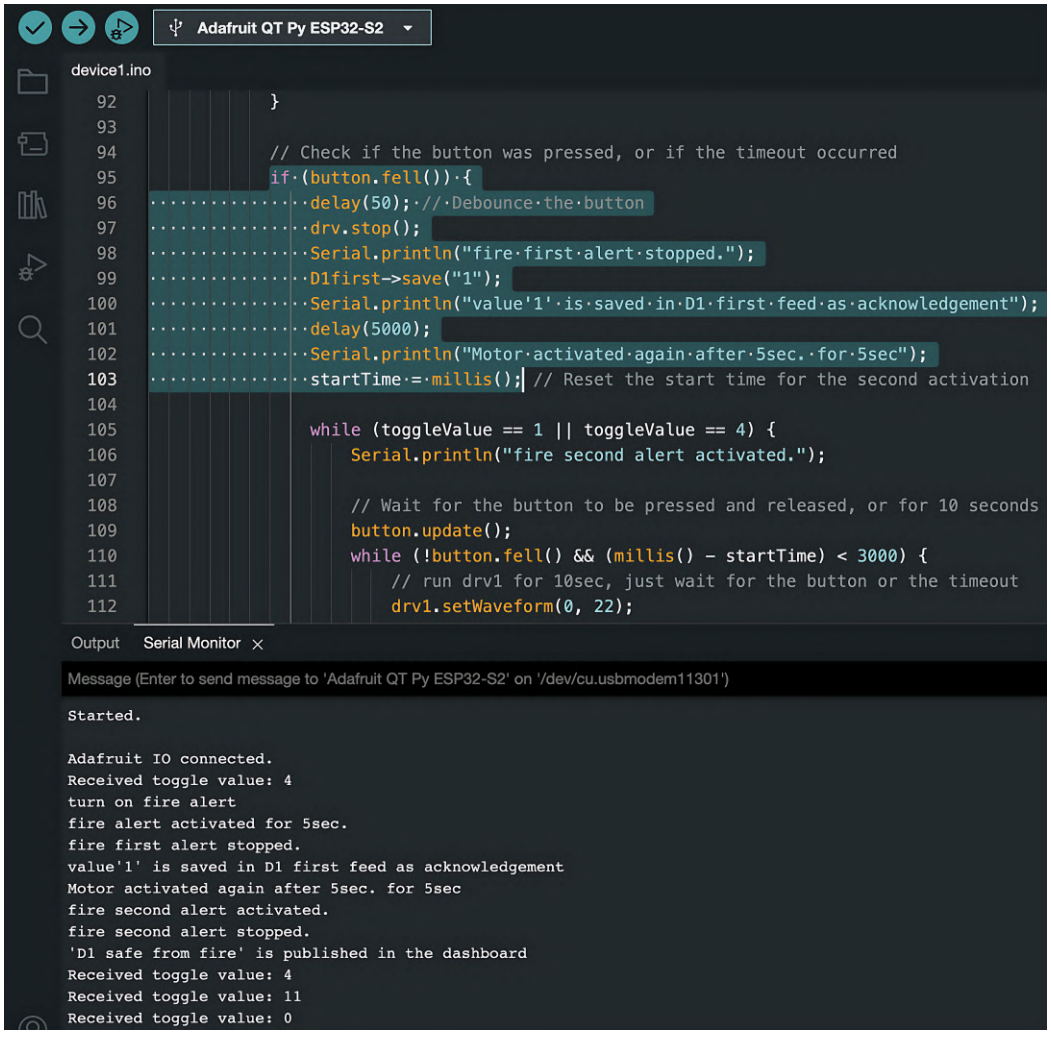

Figure 5.11: Serial monitor ouput for the first and second alert.

Case 4:

In Case 4, we validate the scenarios where the first alert is not acknowledged and where the first alert is acknowledged but the second alert is not acknowledged.

If the first alert is activated but the button is not pressed within 1 minute, the vibration motor will stop, and a message of "D1 1st SOS fire" will be sent. Additionally, the toggle button will be turned off, and this information will be displayed in the text block of the dashboard, as depicted in Figure 5.12.

Figure 5.12: Dashboard for the first alert not acknowledged.

On the other hand, if the first alert is acknowledged but the second alert is not acknowledged within a minute after being triggered, a message of "D1 2nd SOS" will be sent to the dashboard. The toggle value will be turned off, and this information will be displayed in the text block of the dashboard, as shown in Figure 5.13.

Figure 5.13: Dashboard for a second alert not acknowledged.

In Figure 5.14, the serial monitor output is displayed when the device receives a toggle value of 4 but the button is not pressed to acknowledge the first alert and second alert. It indicates that the button was not pressed within the specified time frame to acknowledge the alerts.

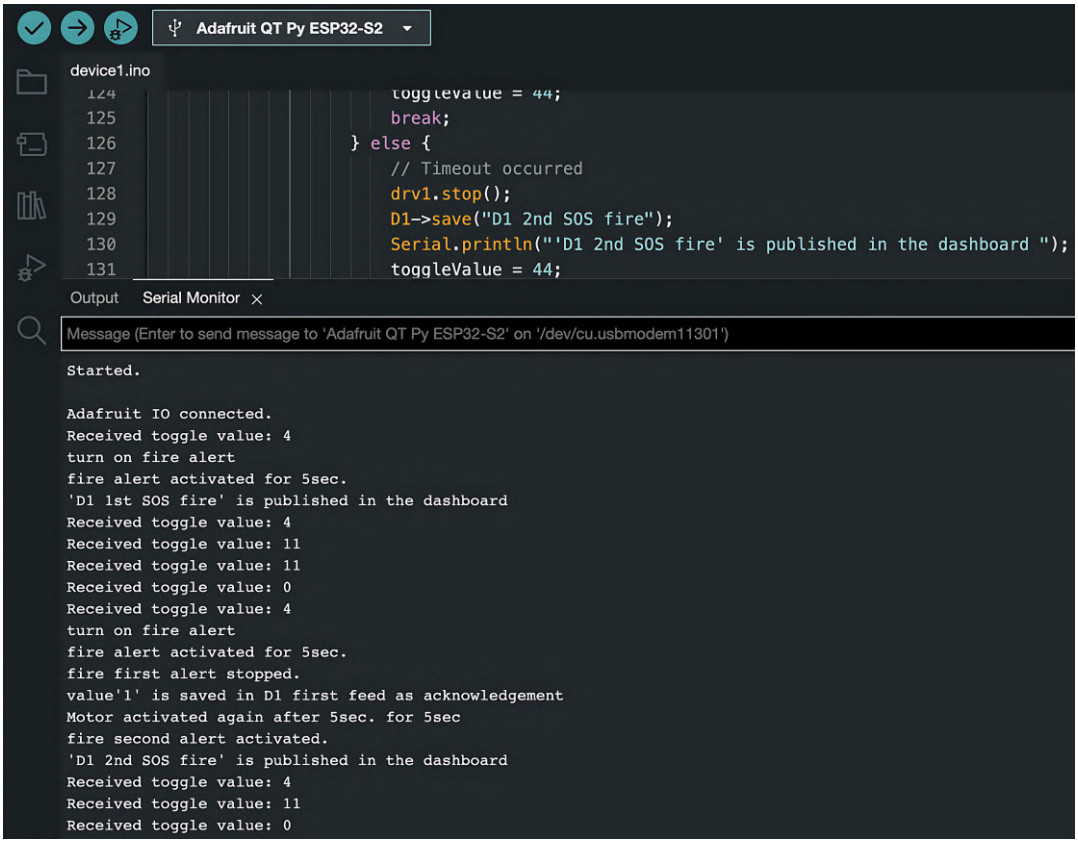

Figure 5.14: Serial monitor shows not acknowledged alerts.

4. Long-pressing the button communicates a 'SAFE' message to the dashboard.

Before validating this functionality, let us discuss its purpose. When an alert is sent from the dashboard or an SOS is triggered by the user, a status indicator will be present on the dashboard. This indicator will normally be green. However, when any of the previously mentioned events occur, it will turn red. This functionality is activated after the SOS and alert have been addressed by the user wearing the device. By holding down the button, the user signals that they are safe. This not only reassures the person monitoring the dashboard but also confirms that the device is ready for further use.

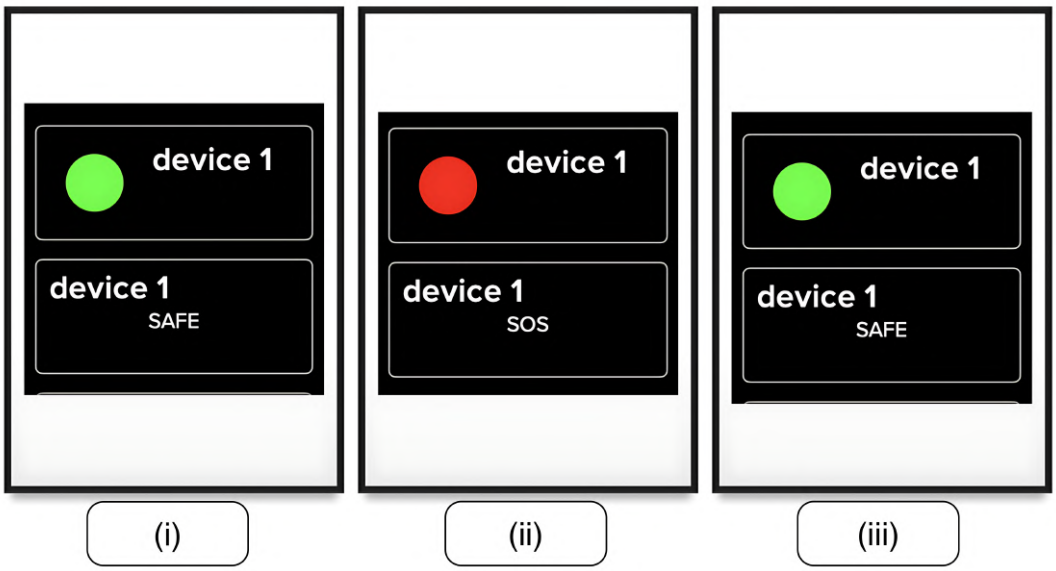

Figure 5.15: Image of status indicator before and after receiving the SAFE message.

In Figure 5.15, the status indicator illustrates three scenarios. Case (i) shows the state when the device user is safe. Case (ii) displays when the device has received an SOS, or when an SOS has been sent from the dashboard. Finally, in case (iii), the indicator reflects the situation when the user long-presses the button for more than 3 seconds, triggering a 'SAFE' message, as depicted in the code snippet in Figure 5.16.

```
141   // Check if button is released
142   if (button.fell()) { // Check if the button was pressed
143     buttonPressed = true; // Set the buttonPressed flag to true
144     buttonPressedStartTime = millis(); // Record the button pressed start time
145   }
146
147   if (button.rose()) { // Check if the button was released
148     buttonPressed = false; // Set the buttonPressed flag to false
149     if (safePrinted) {
150       safePrinted = false; // Reset the flag when the button is released
151     }
152   }
153
154   if (buttonPressed && !safePrinted && (millis() - buttonPressedStartTime) >= 3000) {
155     device1->save("SAFE");
156     drv2.setWaveform(0, 10); // Set the waveform for drv2
157     drv2.go(); // Print "Safe" to the Serial monitor
158     safePrinted = true; // Set the flag to indicate that "Safe" has been printed
159   }
```

Figure 5.16: Code snippet of the functionality long press.

5.2 Evaluation

In the above sections, 'Implementation Flow' 4.2.5 and 'Validation' 5.1, we discussed the flow of implementation and validated the functionality, respectively. Now, in this section, evaluation of each functionality is discussed.

1. **Device connectivity to WiFi and AdafruitIO.**

   We have created a separate configuration file to define the WiFi credentials and Adafruit IO credentials. We have tested the connectivity continuously 20 times. Every time the device connects to WiFi and Adafruit IO, after successfully connecting, we disconnect to check if it reconnects. All 20 times, the device successfully attempted to establish the connection. Another test involved providing incorrect WiFi credentials to see if the device continuously tries to connect to WiFi and Adafruit IO. As long as the device has battery power, it keeps trying to connect, with a 10-minute interval before attempting the connection process again.

2. **User-initiated SOS transmission from the device.**

   Initiating the SOS from the device is possible only by pressing the button three times.

Table 5.1: Number of attempts vs successful attempts vs unsuccessful Attempts

Number of Attempts	**Successful Attempts**	**Unsuccessful Attempts**
50	42	8

The reason behind the unsuccessful attempts to send the SOS is that we used a counting system to recognize three button presses. Initially, when the first button press occurs, we start a timer. If the second press happens within one second of the first press, we increment the button press value. The same rule applies to the third press. However, considering that our users might include elderly individuals who might take longer than one second to press the button after each press, we have increased the time interval to two seconds. As a result of this change, we observed two unsuccessful attempts in Table 5.2. These unsuccessful attempts were due to human error in not pressing the button correctly.

Table 5.2: Number of attempts vs successful attempts vs unsuccessful Attempts after modification

Number of Attempts	**Successful Attempts**	**Unsuccessful Attempts**
30	28	2

3. **Alert transmission from AdafruitIO and reception of an acknowledgement from the user after sending the alert.**

   In the validation of this functionality, we have segregated this functionality into four cases, and we will evaluate each case separately.

   Case 1 and Case 2:

   In the validation process, we validated Case 1 where the toggle button in the dashboard is triggered. When the toggle button is activated, the corresponding toggle value is assigned to the respective feed. In case 2 the device successfully receives this toggle value and functions accordingly. Table 5.3 shows the validation results of cases 1 and 2.

Table 5.3: Case 1 and Case 2

Devices	**Toggle button**	**No of attempts**	**Unsuccessfully attempts**
Device 1	Fire	10	0
	Zombie	10	0
	Both(Fire and Zombie)	10	0
Device 2	Fire	10	0
	Zombie	10	0
	Both(Fire and Zombie)	10	0

   For each device, we can only send one alert at a time, as it has the same feed for all alerts. If we have already sent a fire alert for device 1 and immediately toggle the fire alert button, it will save the assigned toggle value. If we trigger another alert, it will turn off the fire alert and sends the triggered new alert.

   Case 3:

   In this case, before introducing the "bounce2" library to the device, we experienced unintended presses on the mechanical button due to its spring mechanism. However, after integrating the "bounce2" library, we implemented a 5-millisecond interval for debouncing, aiming to reduce the occurrence of unintended presses. This interval allows for the elimination of false readings caused by the mechanical bouncing of the button contacts. As shown in Table-5.4If the button is not released 5 milliseconds after pressing, it won't recognize the press of the button.

Table 5.4: Case 3

No of times button pressed	**Successfully read the button press**
50	33

Case 4:

In the validation of case 4, we discussed two types. The first type is when the first alert is sent but not acknowledged within 5 seconds then it terminates vibration and sends an SOS message, and the second type is when the first alert is acknowledged but the second alert is not acknowledged within 5 seconds.Validation results for type 1 are in Table 5.5.

Table 5.5: Results of type 1 and 2 in case 4.

Type	**Number of Attempts**	**Terminating the vibration**	**Sending the message**
Type 1	30	24	24
Type 2	30	27	24

What happens with unsuccessful attempts is that after 5 seconds, sometimes the process starts again for another 5 seconds instead of terminating the vibration and sending the message. However, after 5 seconds, it eventually terminates. This might be because multiple while loops are used in the code.

4. **Long-pressing the button communicates a 'SAFE' message to the dashboard.**

   For the device to send the message, the button has to be pressed and held for more than 3 seconds. If the holding time is less than or equal to 3 seconds, the device won't send the "SAFE" message. Additionally, after sending the message, a confirmation vibration is triggered.

Now, coming to the vibration patterns for the alerts and other functionalities, we have created distinct and easily recognizable vibration patterns for each alert. We have designed five different types of vibration patterns, one for each cause.

Vibration pattern for Fire alert:

```
drv.setWaveform(0, 58); // Continuous Transition click with delay 200 ms
drv.go();
delay(200);
```

Vibration pattern for Zombie alert:

```
drv.setWaveform(0, 1); // continuous Strong click with delay 100 ms
drv.go();
delay(100);
```

Vibration pattern for Both (Fire & Zombie) alert:

```
drv.setWaveform(0,70); // Transition ramp down long smooth
drv.go();
delay(150); with delay 150 ms
```

Figure 5.17: Three different types of vibration patterns for the alerts.

The use of distinct vibration patterns for different alerts can significantly influence an individual's sensory perception and response time. Each pattern, characterized by its intensity and duration, creates a unique sensory experience that can be learned and recognized over time. For instance, continuous Transition click for fire alert, with a brief 200 ms delay, is designed to grab immediate attention, prompting a quick response. Conversely, the continuous Strong click for a zombie alert, with a 100 ms delay, might be perceived as a more sustained threat, requiring a more prolonged response. The combined fire and zombie alert uses a Transition ramp down long smooth pattern, creating a unique sensory signature that signifies a dual threat. This pattern might take slightly longer to recognize due to its complexity, but it effectively communicates the urgency of the situation. The discussed vibration pattern code is shown in Figure 5.17.

Vibration pattern for Conformation after sending Message to the dashboard:

```
drv.setWaveform(0, 37); //After sensing SOS,SAFE message
drv.go();
```

The vibration pattern for when the device couldn't establish the connection:

```
drv.setWaveform(0, 10); // couldn't connect to Wi-Fi ,AdafruitIO
drv.go();
```

Figure 5.18: Vibration patterns

After sending a message to the dashboard, the confirmation vibration provides reassuring sensory feedback, reinforcing the successful completion of an action. Conversely, the inability to establish a connection is signalled by a different waveform, alerting the user to a problem that needs attention. These vibration patterns shown in Figure 5.18 serve as a tactile language, enhancing the user's awareness and responsiveness to different situations.

To evaluate how accurately this system can convey alerts to users, we approached 17 people of different ages and conditions to try our prototype. Table 5.6 shows the results of this evaluation. Here, we took the average value of alerts recognized in each age group and calculated the percentage.

Table 5.6: The average percentage of alerts recognized by different age groups

Age group	**No of people**	**The average percentage of alerts recognized**
15 <Age <21	3	83.33 %
21 <Age <30	7	94.28 %
31 <Age <45	2	85 %
Age >50	5	72 %

Chapter 6

Discussion

This chapter focuses on the discussion of the thesis, specifically examining whether this project addresses all the research questions that were posed. Furthermore, we will compare the results with the findings from other research studies and discuss the limitations of the thesis.

1. **How can we develop an inclusive alert system, using wearable technology, that effectively notifies individuals of varying abilities during emergency situations?**

   We have developed a systematic inclusive alert system by creating a wearable that can notify each individual through vibrations. Additionally, the system collects information about their condition during emergency situations and acts accordingly based on their acknowledgements of the alerts in time intervals. [41]

2. **How do the intensity and pattern of vibration alerts impact an individual's sensory perception and response time?**

   Research has shown that touch sensitivity is ubiquitous among humans, regardless of their disabilities. In fact, individuals with disabilities often exhibit heightened touch sensitivity compared to those without disabilities, although tactile sensations may be diminished. In our inclusive alert system, we have designed vibration patterns and intensity that are easily recognizable by people with diverse conditions.

   It is important to consider the individual's sensory perception and response time when assessing the impact of vibration alerts. According to research, the average human reaction time to touch is approximately 155 milliseconds [42]. In our system, an alert is given for one minute to allow sufficient time for individuals to acknowledge it.

3. **What design principles can be employed to optimize a wearable alerting system, ensuring rapid and effective delivery of emergency alerts while minimizing the occurrence of false alarms or unnecessary**

alerts?

In our wearable alerting system, we have implemented numbering system to govern the sending and receiving of alerts, as described in Tables 4.6 and 4.7. By using unique values and variable feeds, we have minimized the occurrence of false alerts, unless there is a human error involved.

To ensure the effectiveness of the delivery system, the speed of operation depends on the strength of the connection. When the connection is strong, the delivery of alerts happens in milliseconds. However, if the connection is lost, the device waits until the connection is reestablished.

4. **How can a single-button feedback system be designed and implemented to efficiently collect post-alert condition information from users?**

   In our developed system, we have integrated a single button that serves multiple functionalities. Once an alert is received, the button is used to gather post-alert condition information efficiently within a given time interval. The system responds accordingly based on the number of button presses. For example, if the button is pressed three times continuously, it triggers the sending of an SOS distress signal. Alternatively, a long press of the button initiates the sending of a safe message for confirmation.

5. **What technologies can be used to ensure that the alert system reliably sends simultaneous messages to the relevant department and the user's primary contact, thereby streamlining the emergency response process?**

   In our system, we utilize the AdafruitIO web server for device communication, enabling reliable message transmission to the user's primary contact and the relevant emergency response departments simultaneously. Additionally, we incorporate IFTTT (If This Then That) to seamlessly transmit messages to the designated recipients. The device is programmed using Arduino IDE, facilitating the integration and synchronization of these technologies. This comprehensive approach ensures a streamlined emergency response process, optimizing efficiency and reliability.

The mentioned studies and related work focus on the development and evaluation of wearable-based alerting systems for emergency situations, particularly targeting individuals with disabilities. These studies explore the use of various modalities such as visual cues, haptic alerts, and auditory alarms to effectively notify users about potential dangers and improve their situational awareness and response times.

The first study by Fraser Young et al. proposes a deep learning-based wearable healthcare IoT device for AI-enabled hearing assistance automation. It utilizes a smartwatch to deliver visual and haptic alerts to deaf individuals during emergency situations, demonstrating significant improvements in awareness and response time [27].

Vanessa Cobus and Wilko Heuten evaluate multimodal ICU alarms for emergency situations, integrating visual, auditory, and haptic alerts to enhance situational awareness and response times for both individuals with disabilities and without disabilities [1].

Charalampos Orfanidis et al. present a discreet wearable long-range emergency system based on embedded machine learning. Their system uses sensors to identify abnormal environmental conditions and user conduct, providing timely alerts through visual cues, vibrations, and audible sounds [2].

Steven Goodman et al. develop a smartwatch-based personal safety system that detects abnormal environmental conditions and user behaviours. When an emergency is detected, the system delivers appropriate alerts, such as visual cues, vibrations, and audible alarms, facilitating prompt notifications and responses [28].

Giuseppe D'Aniello et al. discuss wearable computing systems for emergency notification and situation awareness in individual and group contexts. They explore the advantages and challenges of wearable technology, emphasizing the need for integrating it into current emergency response systems for improved situational awareness, communication, and coordination [3].

Regarding the validation and evaluation of a developed system, the provided information describes the functionalities and steps involved in validating each of them. These include device connectivity to WiFi and AdafruitIO, user-initiated SOS transmission, alert transmission from AdafruitIO, and reception of acknowledgements from the user. The validation process involves testing these functionalities and verifying their proper functioning through code snippets, serial monitor outputs, and dashboard representations.

Overall, the mentioned studies and evaluation process highlight the significance of wearable-based alerting systems in enhancing the safety and well-being of individuals, especially those with disabilities, during emergency situations. They demonstrate the potential of integrating various modalities and intelligent sensing capabilities into wearable devices to improve situational awareness, communication, and response times.

Wearable Emergency Alerting System With Enhanced Inclusivity with Interactive Acknowledgment and SOS Distress Signal. It's important to note that this system goes beyond the functionalities offered by the MiniFinder device. Our system has been designed with advanced features that aim to provide a higher level of safety and reassurance. The key aspect is the interactive acknowledgement device,

which allows users to actively confirm their safety and well-being during emergencies. This feature not only provides peace of mind to the user but also helps emergency responders quickly identify individuals who may need assistance. Additionally, we have developed a customized SOS distress signal that is specifically tailored for emergency situations. This ensures that the distress signal is easily recognizable and triggers an appropriate and prompt response from emergency services. These unique aspects of our thesis set it apart from existing solutions and emphasize our focus on user interaction and personalized distress signals [43].

6.1 Limitations

Every system has if cons even though it has been very careful and there will always be room for improvement, by addressing the limitations of the system.

- The main drawback of the system is its dependency on a WiFi connection and AdafruitIO. The device can operate and function effectively only when it is connected to WiFi. Until the connection is established, the system cannot receive alerts from the web server.

- The device is not water-resistant.

- The web server used in this thesis is a public server, which may experience increased traffic due to its usage by multiple users.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

In this thesis, we have offered the implementation and validation of a wearable-based alerting system for emergency situations. The system contains numerous functionalities inclusive of device connectivity to Wi-Fi and Adafruit IO, user-initiated SOS transmission, alert transmission from Adafruit IO, reception of acknowledgements from the user, and long-press communication of a "SAFE" message.

Through accurate checking out and evaluation, we've efficiently established the capability of every aspect of the device. The device has proven dependable connectivity to Wi-Fi and Adafruit IO, organising a constant connection in all 20 tries. Additionally, we addressed the difficulty of spotting button presses correctly, enhancing the time Arduino IDE to accommodate ability delays. This resulted in a significantly progressed success rate of 28 out of 30 tries. The validation of alert transmission from Adafruit IO and reception of acknowledgements highlighted the device's performance in delivering indicators to the consumer. The tool successfully spoke back to toggle values induced at the dashboard, taking into consideration effective communication and well-timed reaction. The implementation of vibration styles further enhanced user focus, with discernible patterns for every form of alert and affirmation vibrations imparting reassuring feedback.

In conclusion, this wearable-based alerting system gives a strong and inclusive solution for emergency situations because it demonstrates dependable connectivity, accurate person-initiated SOS transmission, efficient alert transmission, and accurate reception of acknowledgements. The extended vibration styles function as a tactile language, improving user focus and responsiveness to one-of-a-kind situations. This thesis contributes to the sphere of emergency response structures, presenting a basis for further improvements in the wearable era and inclusive design.

7.2 Future Work

While the wearable-based alerting system offered in this thesis has proven promising outcomes, there are possibilities for intention studies and improvement which could similarly enhance its capabilities and effectiveness.

One capability development for future work is to explore the mixing of additional sensors into the wearable device. By incorporating sensors including smoke detectors, air high-quality sensors, or coronary heart price monitors, the system can offer extra complete and appropriate-made emergency signals based on actual-time statistics. This would enable users to get hold of signals not accessible for predefined emergencies but also for personalized situations that need urgent attention.

Another area of future exploration is the refinement of the user back-and-forth and observation mechanisms. Currently, the device makes use of button presses and vibration patterns for communication. However, alternative modalities inclusive of audio cues, visible indicators, or even haptic observations on exceptional frame parts may be investigated to house users with unique selections or sensory impairments. Additionally, user studies and analysis collection can help inform the design of extra intuitive and user-easy interfaces for beginning SOS transmissions or acknowledging signals.

In summary, future work ought to attention to increasing the device's abilities through the integration of extra sensors, refining user interaction and mechanisms, addressing scalability and interoperability demanding situations, and engaging in complete user studies. These advancements will contribute to the development of a much better, adaptable, and user-centric wearable-based alerting device for emergency situations. [15]

References

[1] V. Cobus and W. Heuten, "To beep or not to beep? evaluating modalities for multimodal ICU alarms," vol. 3, no. 1, p. 15, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2414-4088/3/1/15

[2] C. Orfanidis, R. B. H. Hassen, A. Kwiek, X. Fafoutis, and M. Jacobsson, "A discreet wearable long-range emergency system based on embedded machine learning," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 182–187. [Online]. Available: https://ieeexplore.ieee.org/document/9430981

[3] G. D'Aniello, R. Gravina, M. Gaeta, and G. Fortino, "Situation-aware sensor-based wearable computing systems: A reference architecture-driven review," vol. 22, no. 14, pp. 13 853–13 863, conference Name: IEEE Sensors Journal. [Online]. Available: https://ieeexplore.ieee.org/document/9794918

[4] K. Rembor. Adafruit learning system. [Accessed: 2023-06-01]. [Online]. Available: https://learn.adafruit.com/assets/107493

[5] A. lady. Adafruit DRV2605l haptic controller breakout. [Accessed: 2023-05-25]. [Online]. Available: https://learn.adafruit.com/adafruit-drv2605-haptic-controller-breakout/overview

[6] A. Industries. Lithium ion polymer battery ideal for feathers - 3.7v 400mah : ID 3898 : $6.95 : Adafruit industries, unique & fun DIY electronics and kits. [Accessed: 2023-05-25]. [Online]. Available: https://www.adafruit.com/product/3898

[7] I. Adafruit. STEMMA QT / qwiic JST SH 4-pin cable - 100mm long. [Accessed: 2023-05-25]. [Online]. Available: https://www.adafruit.com/product/4210

[8] T. Cooper. Adafruit learning system. [Accessed: 2023-06-01]. [Online]. Available: https://learn.adafruit.com/assets/120841

[9] A. Industries. Adafruit QT py ESP32-s2 WiFi dev board with uFL antenna port. [Accessed: 2023-05-08]. [Online]. Available: https://www.adafruit.com/product/5348

[10] I. Adafruit. Vibrating mini motor disc. [Accessed: 2023-05-25]. [Online]. Available: https://www.adafruit.com/product/1201

[11] K. Rembor. Adafruit LiIon or LiPoly charger BFF add-on for QT py. [Accessed: 2023-05-25]. [Online]. Available: https://learn.adafruit.com/adafruit-qt-py-charger-bff/overview

[12] A. Industries. Colorful round tactile button switch assortment - 15 pack. [Accessed: 2023-05-25]. [Online]. Available: https://www.adafruit.com/product/1009

[13] Disability and health emergency preparedness | CDC. [Accessed: 2023-05-26]. [Online]. Available: https://www.cdc.gov/ncbddd/disabilityandhealth/emergencypreparedness.html

[14] P. Kashmanian, "Fire alarms and people with ASD: A literature summary," [Accessed: 2023-05-26]. [Online]. Available: https://www.nfpa.org/News-and-Research/Data-research-and-tools/Detection-and-Signaling/Fire-Alarms-and-People-with-ASD

[15] C. Selvi, R. Subramaninan, M. Yogeswari, B. Subhasini, and G. Priya, "Role of wireless wearable technologies in recent advancements," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 1167–1171. [Online]. Available: https://ieeexplore.ieee.org/document/9388572

[16] Federica Laricchia. Topic: Wearables. [Accessed: 2023-05-31]. [Online]. Available: https://www.statista.com/topics/1556/wearable-technology/

[17] How wearable emergency alert devices are saving lives. Section: Elder Care. [Online]. Available: https://www.rescusaveslives.com/blog/how-wearable-emergency-alert-devices-are-saving-lives/

[18] S. V. Khadonova, A. V. Ufimtsev, and S. S. Dymkova, "Wide application innovative monitoring system with personal smart devices," in *2020 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9166115

[19] F. Xiao, Q. Miao, X. Xie, L. Sun, and R. Wang, "Indoor anti-collision alarm system based on wearable internet of things for smart healthcare," vol. 56, no. 4, pp. 53–59, conference Name: IEEE Communications Magazine. [Online]. Available: https://ieeexplore.ieee.org/document/8337896

[20] S. Uppal, S. Raheja, and N. Ranjan Das, "Fire detection alarm system using deep learning," in *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 54–58. [Online]. Available: https://ieeexplore.ieee.org/document/10048842

[21] K. Chen, M. Huang, X. Zhu, and G. Wang, "Learning disability early warning system based on classification algorithm," in *2021 2nd International Conference on Information Science and Education (ICISE-IE)*, pp. 597–600. [Online]. Available: https://ieeexplore.ieee.org/document/9742515

[22] R. Sowah, A. R. Ofoli, S. Krakani, and S. Fiawoo, "A web-based communication module design of a real-time multi-sensor fire detection and notification system," in *2014 IEEE Industry Application Society Annual Meeting*, pp. 1–6, ISSN: 0197-2618. [Online]. Available: https://ieeexplore.ieee.org/document/6978416

[23] K. Dolui, S. Mukherjee, S. K. Datta, and V. Rajamani, "ReTiHA: Real time health advice and action using smart devices," in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 979–984. [Online]. Available: https://ieeexplore.ieee.org/document/6993101

[24] Z. Yizhe, "Design and implementation of firefighter signs monitoring system based on internet of things technology," in *2021 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, pp. 151–154. [Online]. Available: https://ieeexplore.ieee.org/document/9723757

[25] D. Dias and J. Paulo Silva Cunha, "Wearable health devices—vital sign monitoring, systems and technologies," vol. 18, no. 8, p. 2414, number: 8 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/18/8/2414

[26] E. Balla, V. Daniilidis, G. Karlioti, T. Kalamas, M. Stefanidou, N. D. Bikiaris, A. Vlachopoulos, I. Koumentakou, and D. N. Bikiaris, "Poly(lactic acid): A versatile biobased polymer for the future with multifunctional properties—from monomer synthesis, polymerization techniques and molecular weight increase to PLA applications," vol. 13, no. 11, p. 1822, number: 11 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2073-4360/13/11/1822

[27] F. YOUNG, L. ZHANG, R. JIANG, H. LIU, and C. WALL, "A deep learning based wearable healthcare iot device for AI-enabled hearing assistance automation," in *2020 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 235–240, ISSN: 2160-1348. [Online]. Available: https://ieeexplore.ieee.org/document/9469537

[28] S. Goodman, S. Kirchner, R. Guttman, D. Jain, J. Froehlich, and L. Findlater, "Evaluating smartwatch-based sound feedback for deaf and hard-of-hearing users across contexts," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. Association for Computing Machinery, pp. 1–13, [Accessed: 2023-05-24]. [Online]. Available: https://doi.org/10.1145/3313831.3376406

[29] Alerting people with disabilities and access and functional needs | FEMA.gov. [Accessed:2023-05-25]. [Online]. Available: https://www.fema.gov/emergency-managers/practitioners/integrated-public-alert-warning-system/public/alerting-people-disabilities

[30] A. Boulemtafes and N. Badache, "Design of wearable health monitoring systems: An overview of techniques and technologies," in *mHealth Ecosystems and Social Networks in Healthcare*, ser. Annals of Information Systems, A. A. Lazakidou, S. Zimeras, D. Iliopoulou, and D.-D. Koutsouris, Eds. Springer International Publishing, pp. 79–94, [Accessed: 2023-05-25]. [Online]. Available: https://doi.org/10.1007/978-3-319-23341-3_6

[31] S. David. How wearable technology could assist emergency services. [Accessed: 2023-05-25]. [Online]. Available: https://www.techradar.com/news/how-wearable-technology-could-assist-emergency-services

[32] I.-L. Wang, L.-I. Wang, Y. Liu, Y. Su, S. Yao, and C.-S. Ho, "Application of real-time visual feedback system in balance training of the center of pressure with smart wearable devices," vol. 18, no. 18, p. 9637. [Online]. Available: https://www.mdpi.com/1660-4601/18/18/9637

[33] E. Jarocka, J. A. Pruszynski, and R. S. Johansson, "Human touch receptors are sensitive to spatial details on the scale of single fingerprint ridges," vol. 41, no. 16, pp. 3622–3634, publisher: Society for Neuroscience Section: Research Articles. [Online]. Available: https://www.jneurosci.org/content/41/16/3622

[34] B. Rubell. Welcome to adafruit IO. [Accessed: 2023-05-25]. [Online]. Available: https://learn.adafruit.com/welcome-to-adafruit-io/overview

[35] T. Treece. Adafruit IO basics: Dashboards. [Accessed: 2023-05-25]. [Online]. Available: https://learn.adafruit.com/adafruit-io-basics-dashboards/overview

[36] R. Brent. Adafruit IO basics: Schedule actions. [Accessed: 2023-05-25]. [Online]. Available: https://learn.adafruit.com/adafruit-io-basics-scheduled-triggers/overview

[37] K. Rembor. Adafruit QT py ESP32-s2 and QT py ESP32-s2 with uFL antenna. [Accessed: 2023-05-25]. [Online]. Available: https://learn.adafruit.com/adafruit-qt-py-esp32-s2/overview

[38] Arduino. About arduino. [Accessed: 2023-05-25]. [Online]. Available: https://www.arduino.cc/en/about

[39] T. Treece. Using IFTTT with adafruit IO to make an IoT door detector. [Accessed: 2023-05-25]. [Online]. Available: https://learn.adafruit.com/using-ifttt-with-adafruit-io/ifttt-to-adafruit-io-setup

[40] "GitHub - thomasfredericks/bounce2: Debouncing library for arduino and wiring." [Online]. Available: https://github.com/thomasfredericks/Bounce2

[41] K. Costain, "Tactile hypersensitivity and "overwhelming subjectivity" in the touch experience of people with congenital deafblindness: Implications for a touch-based pedagogy," vol. 5, [Accessed: 2023-06-01]. [Online]. Available: https://www.frontiersin.org/articles/10.3389/feduc.2020.582808

[42] Reaction times to sound, light and touch - human homo sapiens - BNID 110800. [Accessed: 2023-06-01]. [Online]. Available: https://bionumbers.hms.harvard.edu/bionumber.aspx?id=110800

[43] Minifinder. MiniFinder nano | our personal safety alarm with GPS function. [Accessed: 2023-05-09]. [Online]. Available: https://minifinder.com/products/nano

Faculty of Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden